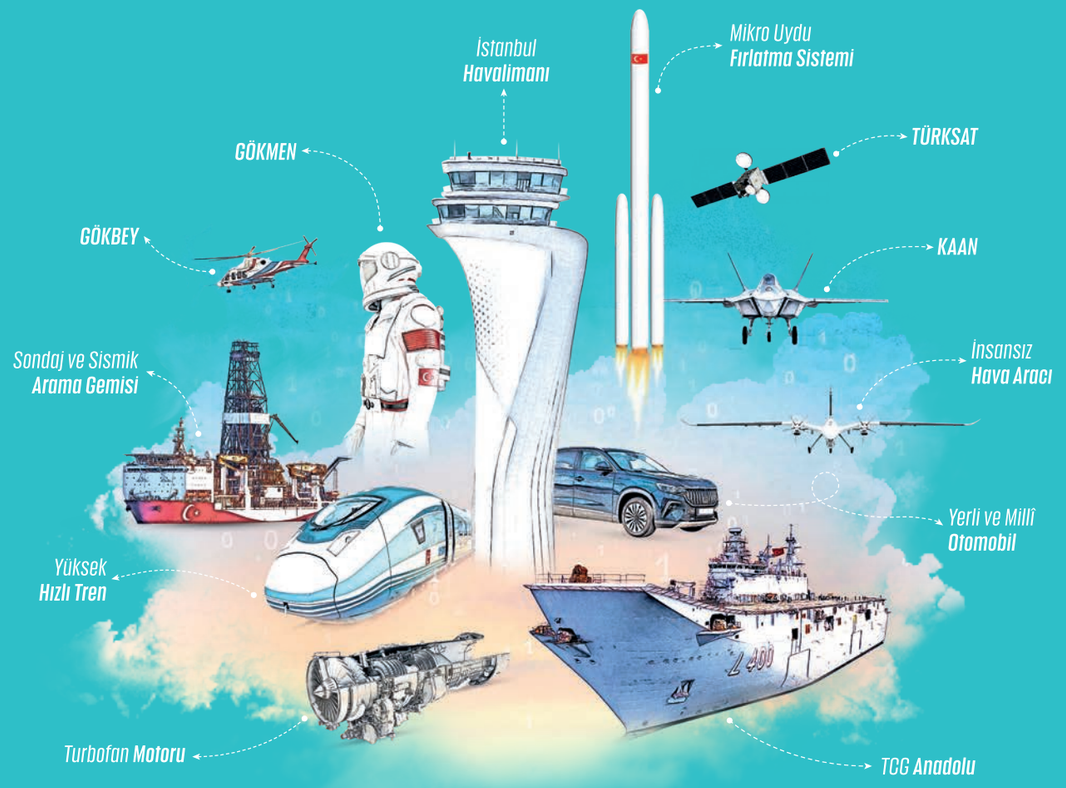
Bu kitaba sığmayan daha neler var!

eba
www.eba.gov.tr

Karekodu okut, bu kitapla ilgili EBA içeriklerine ulaş!

Teknoloji Yolculuğumuz



BU DERS KİTABI MİLLÎ EĞİTİM BAKANLIĞINCA ÜCRETSİZ OLARAK VERİLMİŞTİR.
PARA İLE SATILAMAZ.

ISBN: 978-975-11-7118-4

Bandrol Uygulamasına İlişkin Usul ve Esaslar Hakkında Yönetmeliğin Beşinci Maddesinin İkinci Fıkrası Çerçevesinde Bandrol Taşıması Zorunlu Değildir.

BİLİŞİM TEKNOLOJİLERİ ALANI

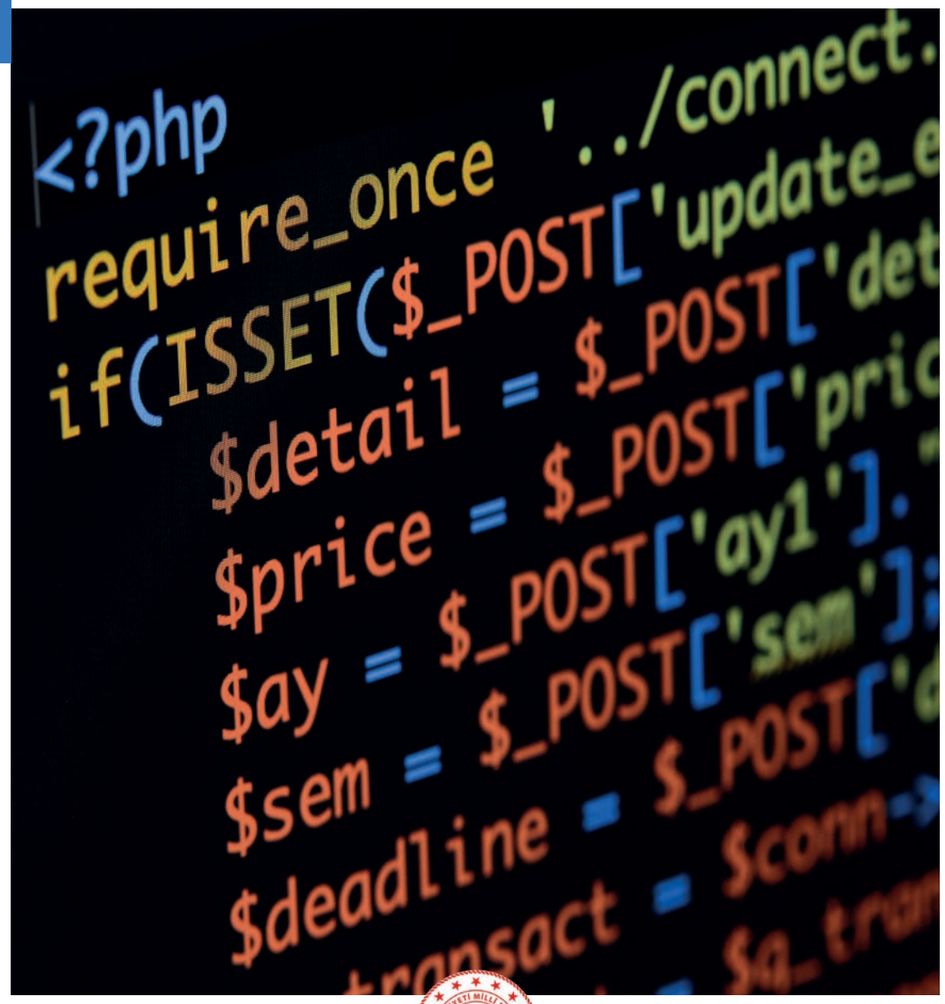
WEB PROGRAMCILIĞI

11-12 DERS MATERYALİ

MESLEKİ VE TEKNİK ANADOLU LİSESİ

BİLİŞİM TEKNOLOJİLERİ ALANI

WEB PROGRAMCILIĞI



11-12 DERS MATERYALİ

MESLEKİ VE TEKNİK ANADOLU LİSESİ

BİLİŞİM TEKNOLOJİLERİ ALANI

WEB PROGRAMCILIĞI
11-12
DERS MATERYALİ

Yazarlar

Abdullah HOCAOĞLU
Mehmet KAVAKLI
Mustafa NACAR
Tolga BOZKURT

MİLLÎ EĞİTİM BAKANLIĞI YAYINLARI..... : 8344
YARDIMCI VE KAYNAK KİTAP DİZİSİ.....: 2236

Her hakkı saklıdır ve Millî Eğitim Bakanlığına aittir. Ders materyalinin metin, soru ve şekilleri kısmen de olsa hiçbir surette alınıp yayımlanamaz.

HAZIRLAYANLAR

Dil Uzmanı
Osman Nuri GÜVEN

Program Geliştirme Uzmanı
Uğurcan TOGAY

Ölçme ve Değerlendirme Uzmanı
Hatice GÜRDİL EGE

Rehberlik Uzmanı
Fatih DÜĞENCİ

Görsel Tasarım Uzmanı
Adike Candan DOĞRUÖZ

ISBN: 978-975-11-7118-4

Millî Eğitim Bakanlığının 24.12.2020 gün ve 18433886 sayılı oluru ile Meslekî ve Teknik Eğitim Genel Müdürlüğünce ders materyali olarak hazırlanmıştır.



İSTİKLÂL MARŞI

Korkma, sönmez bu şafaklarda yüzen al sancak;
Sönmeden yurdumun üstünde tüten en son ocak.
O benim milletimin yıldızıdır, parlayacak;
O benimdir, o benim milletimindir ancak.

Çatma, kurban olayım, çehreni ey nazlı hilâl!
Kahraman ırkıma bir gül! Ne bu şiddet, bu celâl?
Sana olmaz dökülen kanlarımız sonra helâl.
Hakkıdır Hakk'a tapan milletimin istiklâl.

Ben ezelden beridir hür yaşadım, hür yaşarım.
Hangi çılgın bana zincir vuracakmış? Şaşarım!
Kükremiş sel gibiyim, bendimi çiğner, aşarım.
Yırtarım dağları, enginlere sığmam, taşarım.

Garbın âfâkını sarmışsa çelik zırhlı duvar,
Benim iman dolu göğsüm gibi serhaddim var.
Ulusun, korkma! Nasıl böyle bir imanı boğar,
Medeniyyet dediğin tek dişi kalmış canavar?

Arkadaş, yurduma alçakları uğratma sakın;
Siper et gövdeni, dursun bu hayâsızca akın.
Doğacaktır sana va'dettiği günler Hakk'ın;
Kim bilir, belki yarın, belki yarından da yakın.

Bastığın yerleri toprak diyerek geçme, tanı:
Düşün altındaki binlerce kefensiz yatanı.
Sen şehit oğlusun, incitme, yazıktır, atanı:
Verme, dünyaları alsan da bu cennet vatanı.

Kim bu cennet vatanın uğruna olmaz ki feda?
Şüheda fışkıracak toprağı sıksan, şüheda!
Cânı, cânânı, bütün varımı alsın da Huda,
Etmesin tek vatanımdan beni dünyada cüda.

Ruhumun senden İlâhî, şudur ancak emeli:
Değmesin mabedimin göğsüne nâmahrem eli.
Bu ezanlar -ki şehadetleri dinin temeli-
Ebedî yurdumun üstünde benim inlemeli.

O zaman vecd ile bin secde eder -varsa- taşım,
Her cerîhamdan İlâhî, boşanıp kanlı yaşım,
Fışkırır ruh-ı mücerret gibi yerden na'şım;
O zaman yükselerek arşa değer belki başım.

Dalgalan sen de şafaklar gibi ey şanlı hilâl!
Olsun artık dökülen kanlarımın hepsi helâl.
Ebediyyen sana yok, ırkıma yok izmihlâl;
Hakkıdır hür yaşamış bayrağımın hürriyyet;
Hakkıdır Hakk'a tapan milletimin istiklâl!

Mehmet Âkif Ersoy

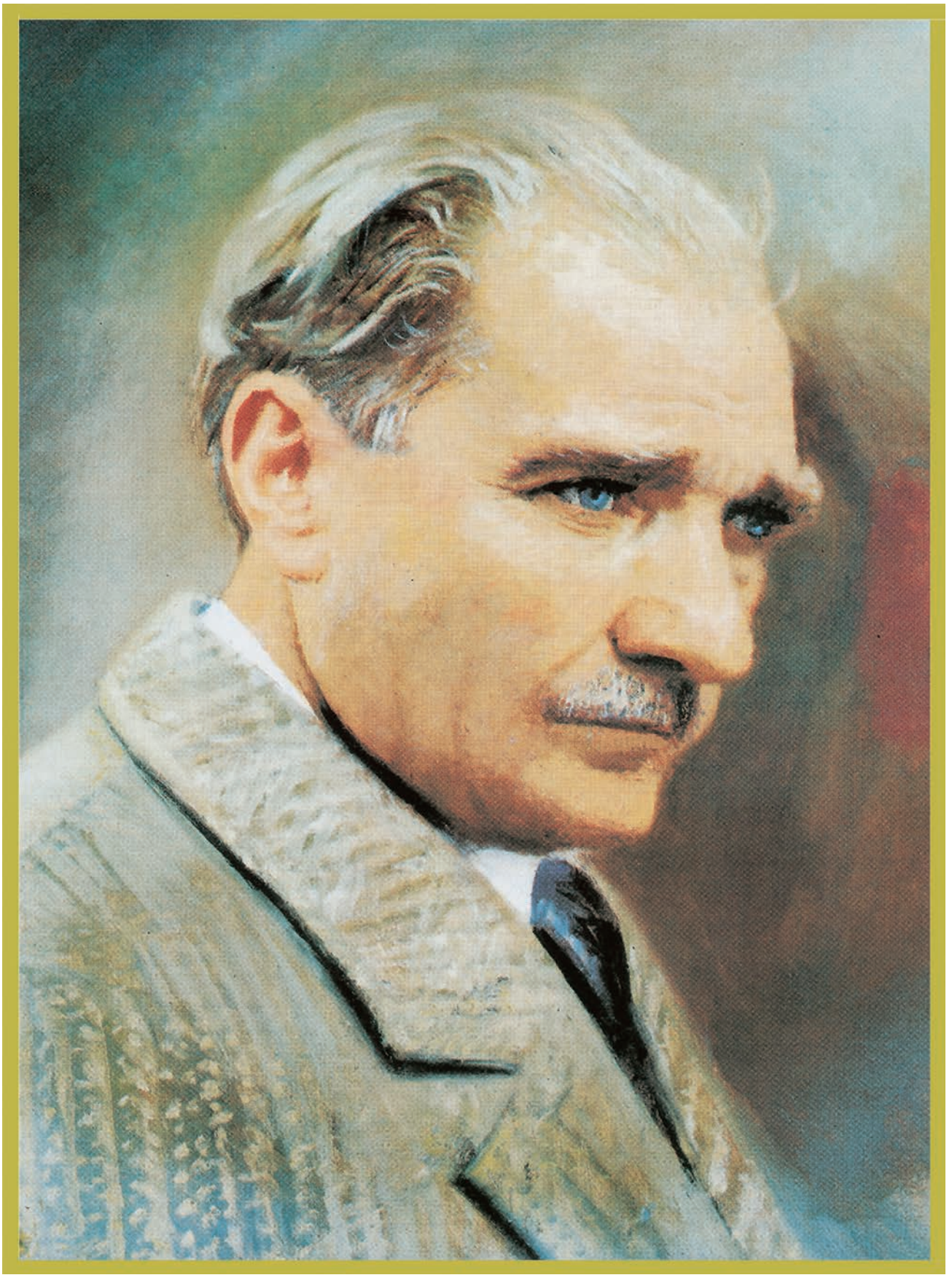
GENÇLİĞE HİTABE

Ey Türk gençliği! Birinci vazifen, Türk istiklâlini, Türk Cumhuriyetini, ilelebet muhafaza ve müdafaa etmektir.

Mevcudiyetinin ve istikbalinin yegâne temeli budur. Bu temel, senin en kıymetli hazinendir. İstikbalde dahi, seni bu hazineden mahrum etmek isteyecek dâhilî ve hâricî bedhahların olacaktır. Bir gün, istiklâl ve cumhuriyeti müdafaa mecburiyetine düşersen, vazifeye atılmak için, içinde bulunacağın vaziyetin imkân ve şeraitini düşünmeyeceksin! Bu imkân ve şerait, çok namüsait bir mahiyette tezahür edebilir. İstiklâl ve cumhuriyetine kastedecek düşmanlar, bütün dünyada emsali görülmemiş bir galibiyetin mümessili olabilirler. Cebren ve hile ile aziz vatanın bütün kaleleri zapt edilmiş, bütün tersanelerine girilmiş, bütün orduları dağıtılmış ve memleketin her köşesi bilfiil işgal edilmiş olabilir. Bütün bu şeraitten daha elîm ve daha vahim olmak üzere, memleketin dâhilinde iktidara sahip olanlar gaflet ve dalâlet ve hattâ hıyanet içinde bulunabilirler. Hattâ bu iktidar sahipleri şahsî menfaatlerini, müstevlîlerin siyasî emelleriyle tevhit edebilirler. Millet, fakr u zaruret içinde harap ve bîtap düşmüş olabilir.

Ey Türk istikbalinin evlâdı! İşte, bu ahval ve şerait içinde dahi vazifen, Türk istiklâl ve cumhuriyetini kurtarmaktır. Muhtaç olduğun kudret, damarlarındaki asil kanda mevcuttur.

Mustafa Kemal Atatürk



MUSTAFA KEMAL ATATÜRK

İÇİNDEKİLER

DERS MATERYALİNİN TANITIMI 13

1. ÖĞRENME BİRİMİ: WEB PROGRAMLAMAYA GİRİŞ 17

1.1. PHP DİLİ VE ÖZELLİKLER 18

1.2. PHP KURULUMU 19

1.3. DEĞİŞKEN VE SABİTLER 28

ÖLÇME VE DEĞERLENDİRME 33

2. ÖĞRENME BİRİMİ: KARAR VE DÖNGÜ YAPILARI 35

2.1. MANTIKSAL OPERATÖRLER 36

2.1.1. Ve (AND-&&) Operatörü 36

2.1.2. Veya (OR - ||) Operatörü 38

2.1.3. XOR Operatörü 39

2.1.4. Değil (!) Operatörü 40

2.1.5. Mantıksal Operatörlerin Doğruluk Tablosu 42

2.2. KARAR KONTROL DEYİMLERİ 42

2.2.1. if-else Kontrol Deyimi 42

2.2.2. Switch-Case Kontrol Deyimi 53

2.3. DÖNGÜ DEYİMLERİ 55

2.3.1. While Döngüsü 55

2.3.2. Do-While Döngüsü 58

2.3.3. For Döngüsü 60

2.3.4. For-Each Döngüsü 63

2.3.5. Break ve Continue Deyimleri 65

2.4. VERİLER ÜZERİNDE DÖNGÜ KONTROLLERİ 68

2.5. DÜZENLİ İFADELER 76

ÖLÇME VE DEĞERLENDİRME....87
3. ÖĞRENME BİRİMİ: DİZİ YAPILARI....89
3.1. DİZİ YAPILARI....90
3.2. DİZİLERLE İŞLEMLER....95
ÖLÇME VE DEĞERLENDİRME....110
4. ÖĞRENME BİRİMİ: FONKSİYONLAR....113
4.1. PHP'DE FONKSİYONLAR....114
4.1.1. Programcı (Kullanıcı) Tanımlı Fonksiyonlar....114
4.1.2. Hazır (Yerleşik) Fonksiyonlar....115
4.2. FONKSİYON PARAMETRELERİ....115
4.2.1. Parametresiz Fonksiyonlar....116
4.2.2. Parametreli Fonksiyonlar....117
4.3. DEĞER DÖNDÜREN FONKSİYONLAR....119
4.4. TARİH / SAAT FONKSİYONLARI....123
4.5. METİN FONKSİYONLARI....126
4.6. MATEMATİK FONKSİYONLARI....132
ÖLÇME VE DEĞERLENDİRME....136
5. ÖĞRENME BİRİMİ: FORM İŞLEMLERİ....137
5.1. HTML FORM ELEMANLARI....138
5.1.1. Form Etiketi....138
5.1.2. Input Etiketi....139
5.1.3. Textarea Etiketi....143
5.1.4. Select Etiketi....143
5.2. GET VE POST METOTLARI....150
5.3. GÜVENLİ FORM OLUŞTURMA....161
5.4. FORMDA DOSYA BİLEŞENLERİ KULLANMA....166

ÖLÇME VE DEĞERLENDİRME....174

6. ÖĞRENME BİRİMİ: HATA YAKALAMA İŞLEMLERİ 175

6.1. PHP İSTİSNA İŞLEMLERİ....176

6.2. Try-Catch BLOKU....178

6.3. Finally BLOKU....181

ÖLÇME VE DEĞERLENDİRME....183

7. ÖĞRENME BİRİMİ: DOSYA İŞLEMLERİ....185

7.1. PHP'DE KLASÖR VE DOSYA İŞLEMLERİ....186

7.1.1. Dosya İçe Aktarma Fonksiyonları....188

7.2. PHP'DE DOSYA OLUŞTURMA....189

7.3. PHP'DE DOSYA AÇMA....190

7.4. PHP'DE DOSYA İÇİNE BİLGİ EKLEME....193

7.5. PHP'DE DOSYAYI SİLME....194

ÖLÇME VE DEĞERLENDİRME....196

8. ÖĞRENME BİRİMİ: OTURUM YÖNETİMİ....197

8.1. ÇEREZ YÖNETİMİ....198

8.2. OTURUM MİMARİSİ VE $_SESSION GENEL DEĞİŞKENİ....202

ÖLÇME VE DEĞERLENDİRME....206

9. ÖĞRENME BİRİMİ: MYSQL VERİ TABANI KULLANIMI....207

9.1. MySQL TEMEL KAVRAMLARI....208

9.1.1. Sayı Veri Türleri....209

9.1.2. Metin Veri Türleri....209

9.1.3. Tarih ve Saat Veri Türleri....210

9.2. UYGULAMA SUNUCUSUNUN KURULMASI VE YÖNETİLMESİ....211

9.2.1. MySQL Komut Satırı Kullanımı....211

9.2.2. phpMyAdmin Kullanımı....214

9.2.3. Tablo İlişkileri 217
9.3. PHP İLE VERİ TABANI BAĞLANTISI YAPMA 219
9.4. PHP İLE VERİ TABANI OLUŞTURMA 221
9.5. PHP İLE TABLO OLUŞTURMA İŞLEMLERİ 223
9.6. PHP İLE MySQL VERİ TABANINDA VERİ EKLEME, SİLME, GÜNCELLEME İŞLEMLERİ 224
9.6.1. Veri Ekleme 224
9.6.2. Veri Silme 226
9.6.3. Veri Güncelleme 227
9.7. PHP İLE MySQL VERİ TABANINDA VERİ LİSTELEME İŞLEMLERİ 228
ÖLÇME VE DEĞERLENDİRME 248
CEVAP ANAHTARI 250
KAYNAKÇA 253

DERS MATERYALİNİN TANITIMI



Öğrenme biriminin adını gösterir.

Öğrenme biriminde neler öğrenileceğini gösterir.

Öğrenme biriminin kare-kodunu gösterir.

Derse başlamadan yapılacak olan hazırlıkları gösterir.

Hazırlık Çalışmaları

1. Betik diller ile programlama dilleri arasındaki farklar nelerdir?

Konu başlığını gösterir.

2.1. MANTIKSAL OPERATÖRLER

Mantıksal (logical) operatörlerin kullanıldığı ifadede ifadenin doğru mu, yanlış mı olduğuna bağlı olarak 0 (false) veya 1 değeri (true) döndürür (Görsel 2.1). Mantıksal operatörler, yazılan programlarda genellikle karar vermede yaygın olarak kullanılan özel karakterlerdir (Tablo 2.1).

Görselleri gösterir.



Görsel 2.1: Mantıksal operatörler

Alt konu başlıklarını gösterir.

2.1.1. Ve (And-&&) Operatörü

Karşılaştırılan iki veya daha fazla koşulun tümünün doğru olduğu durumlarda sonucu doğru olarak geriye döndüren operatördür.

Yapılacak örnekleri gösterir.

1. Örnek

```
<?php
   $a = true;
   $b = true;
   echo $a && $b; // Çıktı: 1 yani true (doğru)
   echo $a and $b; // && işaretinin yerine and ifadesi de kullanılabilir.
?>
```

And ile && operatörleri aynı karşılaştırmayı yapar ancak öncelikleri farklıdır.

Öğrencilerin yapacağı uygulamaları gösterir.

Sıra Sizde

Aşağıdaki kodlarla index.php oluşturup tarayıcıda görüntüleyerek sonucu arkadaşlarınız ile paylaşınız.

Yapılacak örnek uygulamaları gösterir.

1. Uygulama

PHP kodlarının yerel bilgisayarda çalıştırılması işlemini aşağıdaki adımları takip ederek gerçekleştiriniz.

1. Adım: XAMPP sunucusu ile çalışıyorsanız XAMPP'ın kurulu olduğu klasör içindeki **htdocs** klasörüne, direkt Apache sunucusu ile çalışıyorsanız Apache sunucusunun kurulu olduğu klasör içindeki **htdocs** klasörüne **WebSitem** isminde klasör oluşturunuz.

2. Adım: WebSitem klasörünü Visual Studio Code uygulamasında açarak içine **index.php** adında dosya oluşturunuz.

3. Adım: Oluşturulan index.php dosyasına aşağıdaki kodlamaları yaparak dosyayı kaydediniz.

Program kodlarını gösterir.

```
<html>
 <head>
  <title>PHP Denemesi</title>
 </head>
 <body>
```

İpuçlarını gösterir.

Önemli

MySQL User Accounts bölümünde root haricînde yeni kullanıcılar oluşturulabilir.

Ölçme ve değerlendirme sorularını gösterir.

ÖLÇME VE DEĞERLENDİRME

Aşağıdaki soruları dikkatlice okuyarak doğru seçeneği işaretleyiniz.

1. PHP için aşağıdaki ifadelerden hangisi doğrudur?

A) İstemci taraflı betik dildir.

B) Sunucu taraflı betik dildir.

C) Açılımı "Personel Home Page" dir.

D) Tüm kodlar derlendikten sonra çalışır.

E) Ara katman yazılım dilidir.

2. PHP dosya uzantısı aşağıdakilerden hangisidir?

A) .html

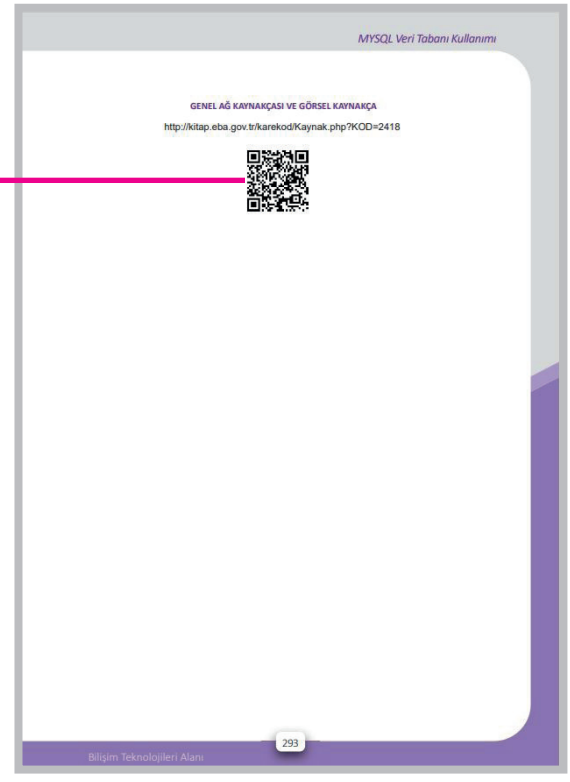
B) .xml

C) .phpx

D) .ph

E) .php

Karekod, görsel kaynakçasını gösterir.



1. ÖĞRENME BİRİMİ

ETKİLEŞİM VE VERİ YÖNETİMİ



KONULAR

1.1. PHP DİLİ VE ÖZELLİKLER

1.2. PHP KURULUMU

1.3. DEĞİŞKEN VE SABİTLER

NELER ÖĞRENECEKSİNİZ?

- PHP dilinin özellikleri
- PHP dilinin kullanım alanları
- PHP ile uygulama geliştirmek için gerekli yazılımların kurulumu
- Bir PHP dosyasını web sunucuda çalıştırma
- PHP dilinde değişkenler oluşturma
- PHP dilinde değişkenlere değer aktarma

TEMEL KAVRAMLAR

PHP, Apache, MySQL, PHP değişkenler, PHP sabitler.



Hazırlık Çalışmaları

1. **Betik diller ile programlama dilleri arasındaki farkı araştırınız.**
2. **Bir web sunucusu nasıl çalışır?**

1.1. PHP DİLİ VE ÖZELLİKLER

PHP; sunucu üzerinde çalışan, açık kaynak, ücretsiz, web uygulamaları geliştirmek için kullanılan betik programlama dilidir. 1995'te Rasmus Lerdorf tarafından geliştirilen PHP ilk olarak Personel Home Page (Kişisel Anasayfa) olarak adlandırılmaktayken günümüzde PHP Hypertext Preprocessor (PHP Üstün Yazı Ön işlemcisi) olarak adlandırılmaktadır.

PHP betik dilinin özellikleri şunlardır:

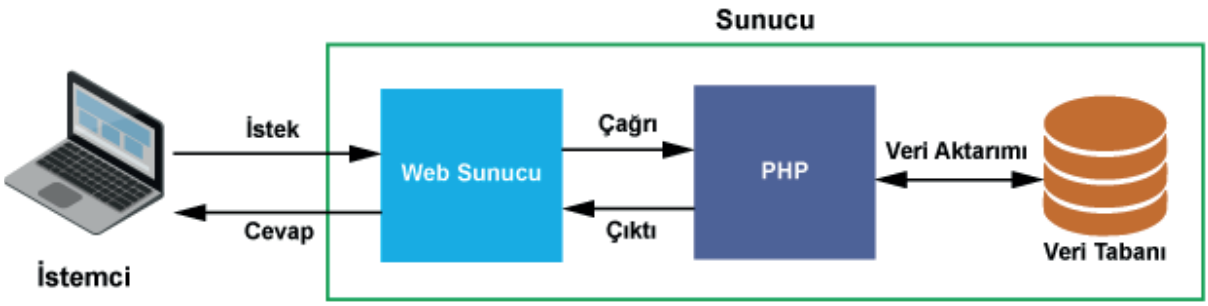
- PHP açık kaynak kodludur ve kullanımı ücretsizdir. İnternet üzerinden ücretsiz indirilebilir ve kullanılabilir. Kaynak kodlar incelenebilir ve gelişimine katkıda bulunulabilir.
- PHP kodlaması kolay ve basittir. Kodların yazım şekilleri C diline benzer. Kodların yazım kolaylığı ve basitliği programcılar tarafından tercih edilme nedenlerinin başında gelir.
- PHP; Windows, MacOS, Linux ve Unix türevleri işletim sistemlerinde çalıştırılabilmekte ve IIS, Apache gibi çeşitli web sunucuları tarafından desteklenmektedir.
- Esnek kullanımı sayesinde PHP kodları HTML, Javascript, WML, XML vb. ile birlikte kullanılabilir.
- PHP, hataları yakalama ve hataları görüntüleme özelliği sayesinde programcılara önceden tanımlanmış hataları uyarı ve mesaj olarak gösterir.
- PHP değişken tanımlamalarında küçük / büyük harflere karşı duyarlıdır fakat PHP anahtar kelimelerinde (if, else, while, echo vb.), sınıf isimlerinde, fonksiyon isimlerinde küçük / büyük harflere duyarlı değildir.
- PHP ile nesne yönelimli programlama yapılabilir.
- PHP ile MySQL, SQL Server, Oracle vb. tüm önde gelen veri tabanları ile işlem yapılabilir.
- PHP kötü niyetli saldırılara karşı birden çok güvenlik katmanından oluşan güvenli bir dildir.
- PHP geniş bir yazılımcı topluluğuna sahiptir. Bu topluluk sayesinde güncel ve öğretici dokümanlara ulaşılabilir, sorular sorulabilir ve yardım alınabilir.

PHP betik dilinin genel olarak kullanım alanları şunlardır:

- Dinamik web sayfaları oluşturmak
- Sunucu üzerinde dosya oluşturma, okuma, yazma ve silme işlemleri yapmak
- PHP veri tabanı üzerinde CRUD (Create, Read, Update, Delete - Oluştur, Oku, Güncelle, Sil) işlemleri yapmak
- PHP ile GD grafik kütüphanesi, ImageMagick ve diğer PHP temelli uygulamalar sayesinde resimler üzerinde kesme, döndürme, yeniden boyutlandırma vb. görüntü işlemek

- PHP ile GUI (Graphical User Interface – Grafiksel Kullanıcı Arayüzü) temelli uygulamalar oluşturmak
- PHP ile verileri farklı formlarda çizgi grafik, pasta grafik vb. göstermek
- PHP ile e-ticaret uygulamaları geliştirmek
- PHP ile bir web site içeriğini kolaylıkla yönetecek CMS (Content Management System - İçerik Yönetim Sistemi) oluşturmak

PHP temel çalışma şekli ilk olarak istemci tarafından sunucuya kullanıcının tarayıcısından istek gönderilmesi ile başlar. Gelen istek, sunucu içinde bulunan web sunucusu (Apache, IIS vs.) tarafından analiz edilerek ne yapılacağı belirlenir. Eğer gelen istek index.php gibi bir PHP dosyasına ait ise bu dosya PHP yorumlayıcısına gönderilir. PHP yorumlayıcısı PHP dosyasını okur, inceler ve çalıştırır. Çalıştırma işlemi sonucunda PHP yorumlayıcısı tarafından bir çıktı oluşturulur. Oluşturulan çıktı web sunucusu tarafından cevap olarak istemciye gönderilir. Görsel 1.1'de sunucu üzerinde bulunan PHP sayfalarının nasıl çalıştığı gösterilmiştir.



Görsel 1.1: PHP sayfalarının istemci ile sunucu arasındaki iletişimi

1.2. PHP KURULUMU

PHP ile uygulama geliştirmek için bilgisayara bazı yazılımların kurulması gerekir. Bu yazılımlar şunlardır:

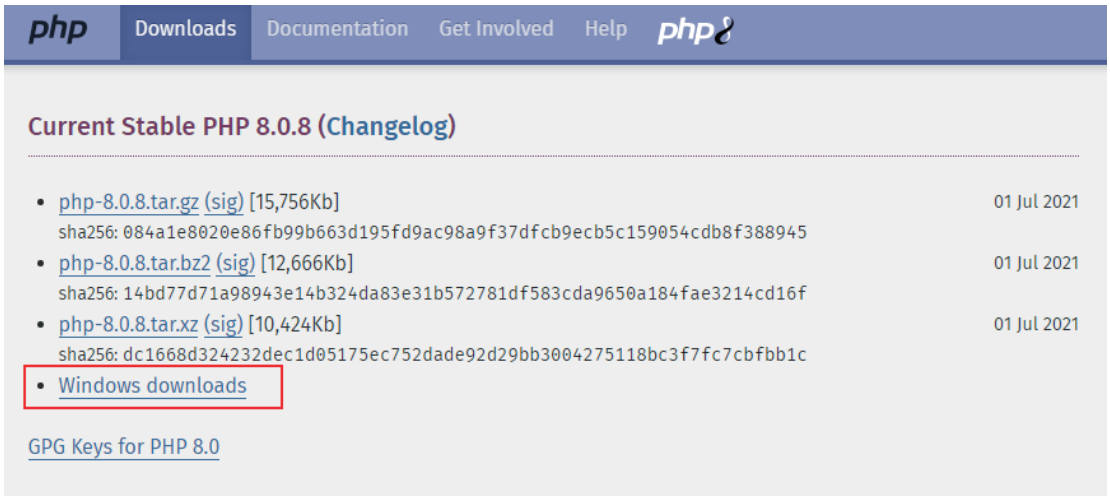
- PHP
- Web Sunucu
- Veri tabanı Sunucu

Yukarıdaki yazılımlar ayrı ayrı kurulabileceği gibi hepsi bir arada paket olarak da kurulabilir. Bu paketler XAMPP, WampServer vb. yazılımlardır.

1. Uygulama

Windows işletim sistemine PHP kurulum işlemlerini aşağıdaki adımları takip ederek gerçekleştiriniz.

1. Adım: Tarayıcınızın adres satırına **www.php.net/downloads.php** yazarak açılan sayfadan Görsel 1.2'deki **Windows downloads** linkine tıklayınız.



Görsel 1.2: PHP dosyalarını Windows için indirme sayfası

2. Adım: En son sürüm sıkıştırılmış dosyayı Görsel 1.3'teki linke tıklayarak indiriniz.

Önemli

Sunucu olarak IIS kullanılacak ise **Non Thread Safe** bölümü altındaki dosyayı indiriniz.



Görsel 1.3: PHP sıkıştırma dosyasını indirme sayfası

3. Adım: Bilgisayarınızın C:\ sürücüsüne **PHP** adında klasör oluşturarak indirilen sıkıştırılmış dosyayı

bu klasör içine çıkartınız.

4. Adım: C:\php klasörü içindeki **php.ini-development** dosyasının ismini **php.ini** olarak değiştiriniz.

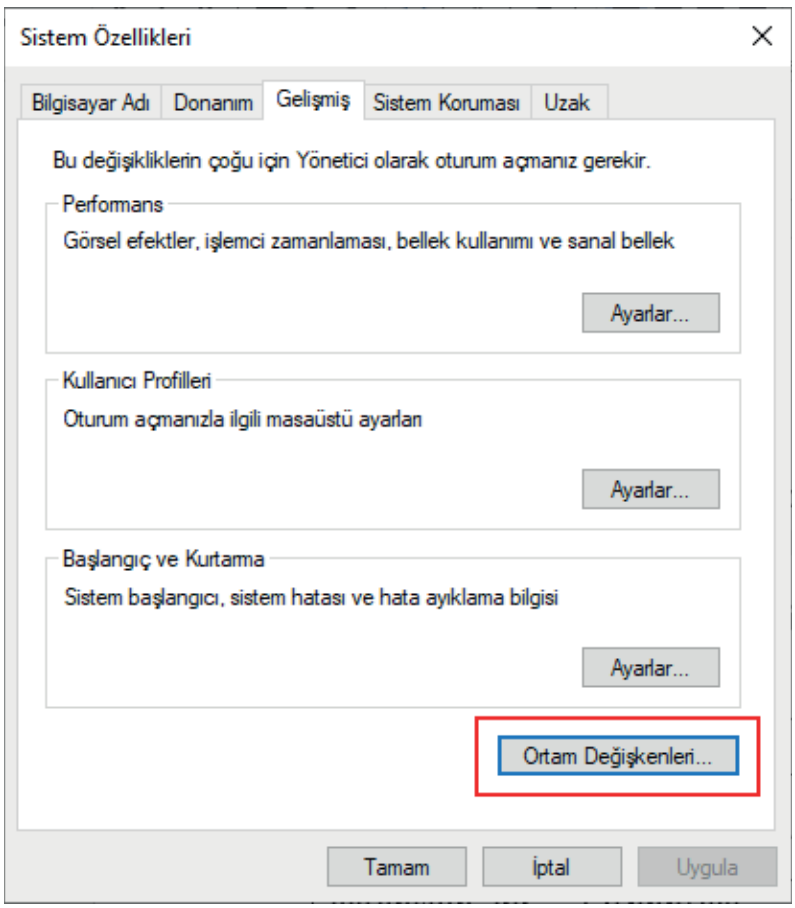
5. Adım: C:\php klasörü içindeki php.ini yapılandırma dosyasını metin düzenleme programında açarak aşağıdaki satırı bulunuz.

```
;extension_dir = “ext”
```

6. Adım: Bulduğunuz satır PHP uzantılarının dizin yoludur. Bu satırı aşağıdaki gibi değiştiriniz.

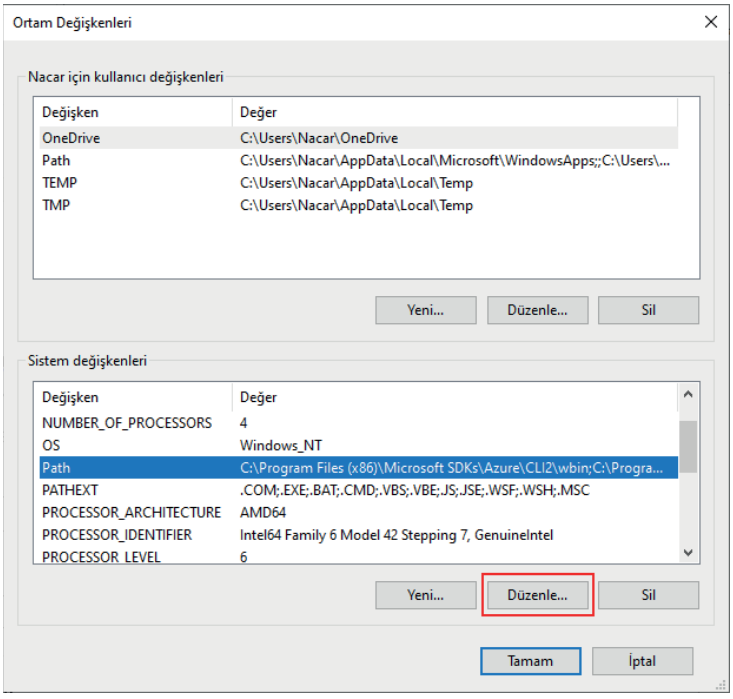
```
extension_dir = “C:\php\ext”
```

7. Adım: Windows işletim sisteminin PHP yürütülebilir dosyalarını bulmasını sağlamak için Görsel 1.4’teki gibi **Sistem Özellikleri** penceresinden **Ortam Değişkenleri** butonuna tıklayınız.



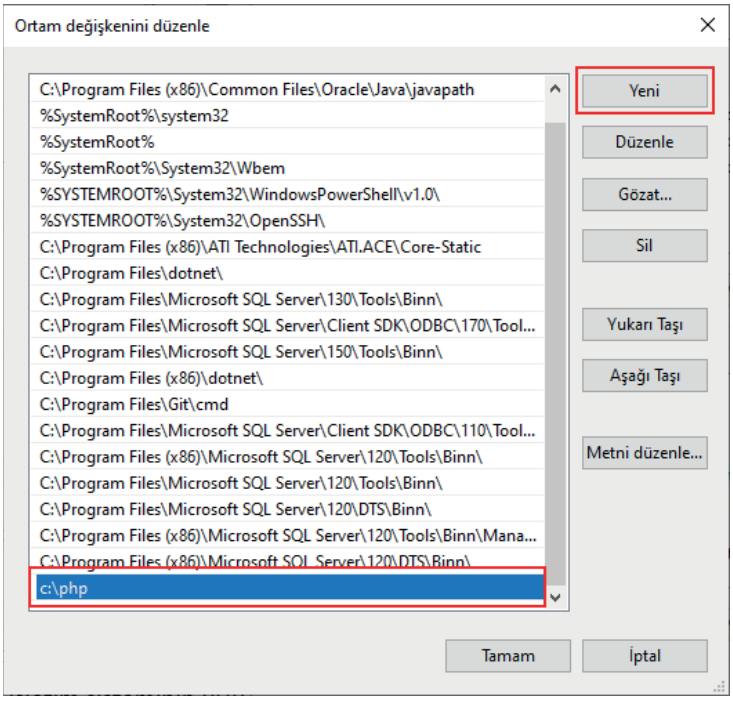
Görsel 1.4: Sistem Özellikleri penceresi

8. Adım: Görsel 1.5'teki **Ortam Değişkenleri** penceresinde **Sistem değişkenleri** bölümündeki **Path** alanını seçtikten sonra **Düzenle** butonuna tıklayınız.



Görsel 1.5: Ortam Değişkenleri penceresi

9. Adım: Ortam değişkenini düzenle penceresinde **Yeni** butonuna tıkladıktan sonra Görsel 1.6'daki gibi PHP'nin kurulu olduğu dizinin yolunu ekleyiniz.



Görsel 1.6: Ortam değişkenini düzenleme penceresi

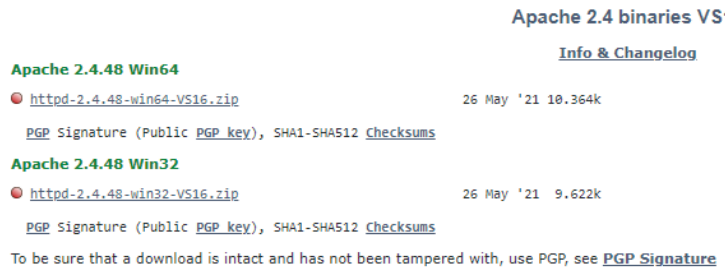
10. Adım: Komut istemi penceresini açarak PHP sürüm kontrolü için **php –v** komutunu yazınız.

Web sunucusu, web sitelerinin sunucuda veya yerel bilgisayarda yayınlanmasından sorumlu olan yazılımdır. Bu yazılım sayesinde HTTP istekleri yerine getirilerek kullanıcıların verilere erişmeleri sağlanır. Kitapta web sunucusu olarak Apache Web Server kullanılacaktır. Apache açık kaynak kodlu ve ücretsizdir.

2. Uygulama

Windows işletim sistemine Apache web sunucusunun kurulum işlemlerini aşağıdaki adımları takip ederek gerçekleştiriniz.

1. Adım: Tarayıcınızın adres satırına **www.apachelounge.com/download** yazarak açılan sayfadan Görsel 1.7'deki gibi işletim sisteminize göre 32-bit veya 64-bit sıkıştırılmış dosya olarak indiriniz.



Görsel 1.7: Apache web sunucusu indirme sayfası

2. Adım: Bilgisayarınızın C:\ sürücüsüne indirilen sıkıştırılmış dosyayı çıkartınız.

3. Adım: Bilgisayarınızın **C:\Apache24\conf** klasörü içindeki **httpd.conf** dosyasını metin düzenleme programında açarak sunucu adının bulunduğu satırı aşağıdaki gibi düzenleyiniz.

```
#ServerName www.example.com:80
ServerName localhost
```

4. Adım: httpd.conf dosyasında ana sayfa olarak index.php'nin de başlangıç sayfası olarak ayarlanması için aşağıdaki düzenlemeleri yapınız.

```
<IfModule dir_module>
      DirectoryIndex index.php index.html
</IfModule>
```

5. Adım: httpd.conf dosyasının en altına aşağıdaki satırları ekleyiniz.

```
PHPIniDir "C:\php"
LoadModule php_module "C:\php\php8apache2_4.dll"
AddType application/x-httpd-php .php
```

1. Satır: Apache'nin php.ini dosyasındaki ayarlara ulaşabilmesi için dosyanın bulunduğu dizini belirtir.

2. Satır: Apache sunucusunun PHP iletişimini sağlar.

3. Satır : **.php** uzantılı dosyaları Apache sunucusuna tanıtır.

6. Adım: httpd.conf dosyasında yapılan değişiklikleri kaydettikten sonra komut istemine aşağıdaki ko-

mutları yazarak test ve güncelleme işlemini gerçekleştiriniz.

```
> cd C:\Apache24\bin
> httpd -t
```

7. Adım: Test sonucunda **Syntax OK** mesajı görüntülendiğinde komut istemi penceresine **httpd** yazarak veya C:\Apache24\bin klasörü içindeki **httpd.exe** dosyasını çalıştırarak sunucuyu başlatınız.

8. Adım: Sunucu başlatıldıktan sonra tarayıcınızın adres satırına **localhost** yazarak açılış sayfasını görüntüleyiniz.

PHP, birçok veri tabanı yazılımı (MySQL, PostgreSQL, MsSQL, Oracle vb.) ile birlikte sorunsuz çalışabilmesine rağmen PHP'de en yüksek performans PHP+MySQL+Apache üçlüsünden elde edilmiştir. Internet üzerinde bulunabilecek hazır PHP betikleri ya da yardım belgelerinin çoğu PHP'yi MySQL ile birlikte ele alır. Bu nedenle kitapta veri tabanı olarak MySQL kullanılacaktır.

3. Uygulama

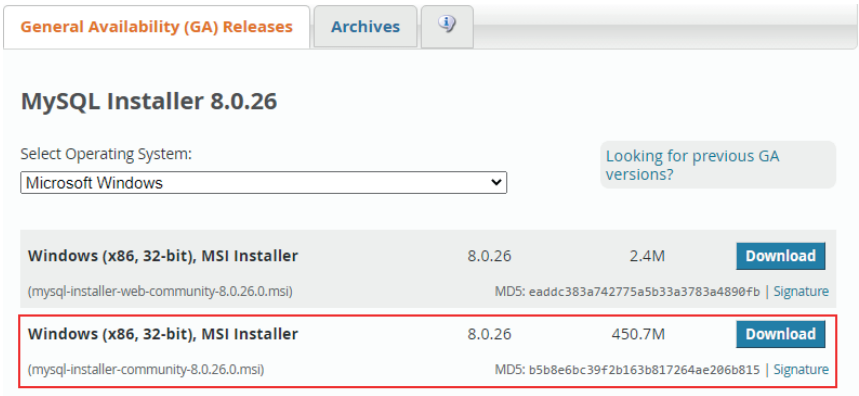
Windows işletim sistemine MySQL veri tabanı sunucusu kurulum işlemlerini aşağıdaki adımları takip ederek gerçekleştiriniz.

1. Adım: Tarayıcınızın adres satırına **https://dev.mysql.com/downloads/mysql** yazarak açılan sayfadan Görsel 1.8'deki kurulum dosyalarını indirme linkine tıklayınız.



Görsel 1.8: MySQL kurulum dosyaları indirme sayfası

2. Adım: Görsel 1.9'da gösterilen MySQL installer dosyasını indirerek çalıştırınız.



Görsel 1.9: Windows için MySQL indirme sayfası

3. Adım: MySQL kurulum dosyasını indirmek için Download butonuna tıkladıktan sonra ekrana gelen sayfada "No thanks, just start my download" linkine tıklayarak indirme işlemini başlatınız.

4. Adım: İndirilen kurulum dosyasını çalıştırınız. Kurulum ekranında ilk seçili alan olan **Developer Default** seçeneğini **Server Only** olarak değiştiriniz ve **Next** butonuna tıklayınız.

5. Adım: Installation penceresinde **Execute** butonuna tıklayarak kurulumun tamamlanmasını bekleyiniz. Kurulum tamamlandıktan sonra **Next** butonuna tıklayınız.

6. Adım: Product Configuration penceresinde **Next** butonuna tıklayınız.

7. Adım: Type and Networking penceresinde **Developer Computer** seçili iken **Next** butonuna tıklayınız.

8. Adım: Authentication Method penceresinde **Next** butonuna tıklayınız.

9. Adım: Account and Roles penceresinde **root** kullanıcısı için unutulmaması gereken bir şifre belirledikten sonra **Next** butonuna tıklayınız.

Önemli

MySQL User Accounts bölümünde root haricînde yeni kullanıcılar oluşturulabilir.

10. Adım: Windows Service penceresinde Windows işletim sistemine **MySQL80** adında bir servis oluşturulacağı belirtilmektedir. Servis ismini değiştirmek istemiyorsanız **Next** butonuna tıklayınız.

11. Adım: Apply Configuration penceresinde yapılan ayarların uygulanması için **Execute** butonuna tıklayarak kurulumun tamamlanmasını bekleyiniz. Kurulum tamamlandığında **Finish** butonuna tıklayarak kurulum işlemini tamamlayınız.

XAMPP, PHP ile uygulama geliştirirken uygulamayı yayınlamadan önce yerel bilgisayarda test etmek için kullanılan web sunucusu yazılımıdır. Yerel bilgisayara Apache, MySQL ve PHP kurulumlarını tek seferde yaparak tek bir program üzerinden yönetilmesini sağlar.

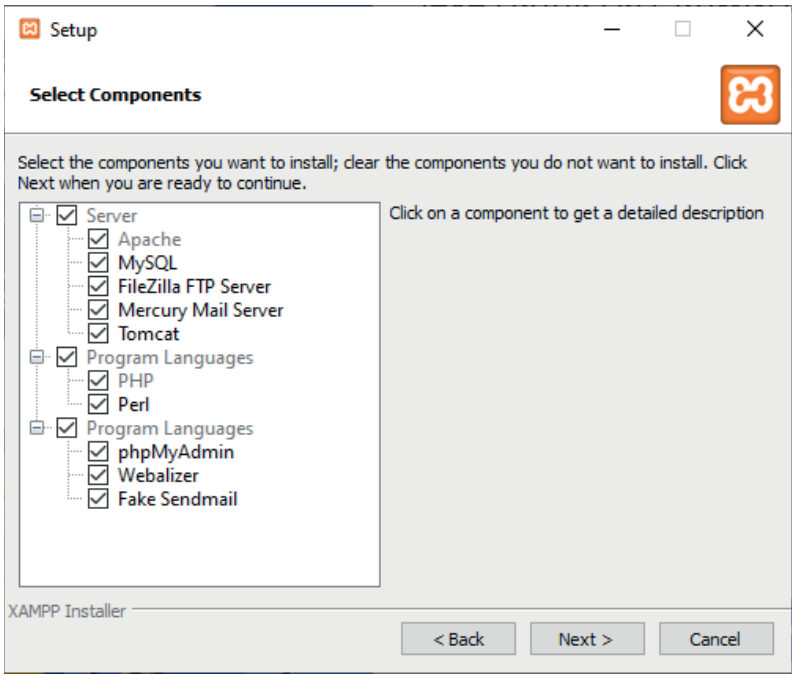
4. Uygulama

XAMPP web sunucunun kurulum işlemlerini aşağıdaki adımları takip ederek gerçekleştiriniz.

1. Adım: Tarayıcınızın adres satırına **http://www.apachefriends.org** yazarak açılan sayfada işletim sisteminize uygun indirme işlemini gerçekleştiriniz.

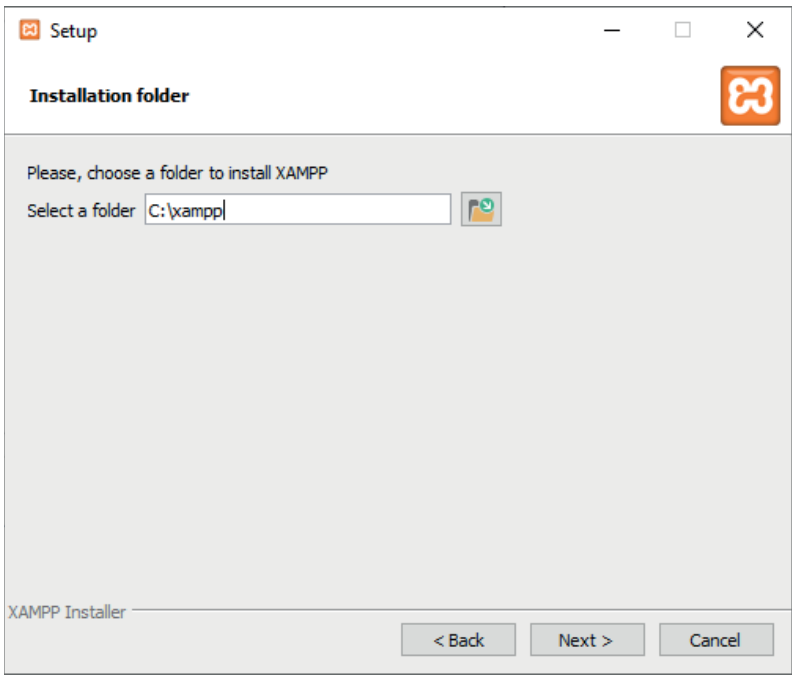
2. Adım: İndirilen kurulum dosyasını çalıştırınız. Daha sonra karşınıza gelen ekranda **Next** butonuna tıklayınız.

3. Adım: Görsel 1.10'da gösterilen kurmak istediğiniz bileşenleri seçiniz. Bu bileşenlerden PHP ve Apache zorunludur, diğerleri ise seçime bağlı olarak kaldırılabilir. PHP ile web uygulamaları geliştirmek için PHP, Apache, MySQL ve phpMyAdmin seçilmelidir. Bir sonraki adıma geçmek için **Next** butonuna tıklayınız.



Görsel 1.10: XAMPP bileşenleri seçim penceresi

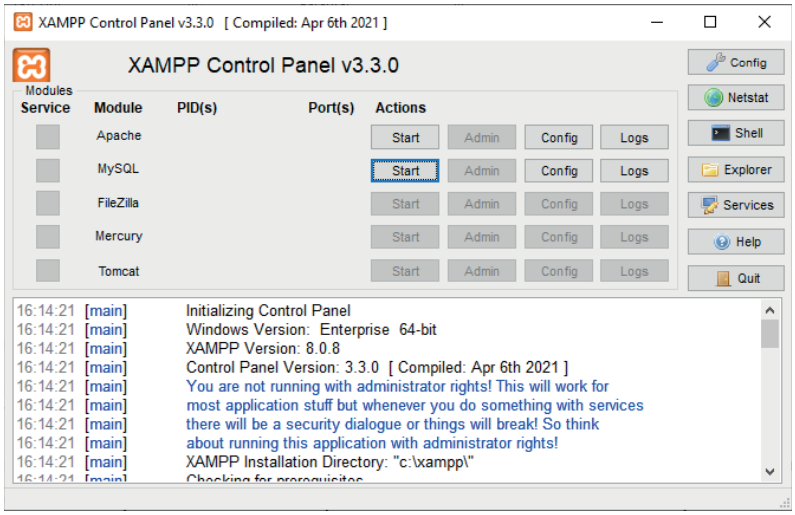
4. Adım: Görsel 1.11'de kurulumun yapılacağı alanın seçileceği ekran gelmektedir. İstenirse farklı alana kurulum yapılabilir. Kurulumun yapılacağı alanı belirledikten sonra **Next** butonuna tıklayınız.



Görsel 1.11: XAMPP kurulum için klasör seçim penceresi

5. Adım: Dil seçiminin yapılacağı pencerede İngilizce ve Almanca seçeneklerinden birini seçerek **Next** butonuna tıklayınız. Daha sonraki pencerelerde **Next** butonuna tıklayarak kurulumu tamamlayınız.

6. Adım: Kurulum tamamlandığında Görsel 1.12'de XAMPP Control Panel uygulamasını açınız.



Görsel 1.12: XAMPP kontrol paneli

7. Adım: XAMPP kontrol panelinden Apache web sunucusunu başlatmak için **Start** butonuna tıklayınız. Başlatma işlemi sırasında port numarası hatası verir ise **Config** butonuna tıklayarak, açılan menüden Apache (httpd.conf) linkine tıklayarak ayar dosyasını açınız. Ayar dosyası içinde aşağıdaki değişiklikleri yaparak dosyayı kaydediniz.

```
Listen 80
yerine
Listen 8080
```

```
ServerName localhost:80
yerine
ServerName localhost:8080
```

8. Adım: Apache web sunucusu çalıştırılması sırasında güvenlik duvarı uyarısına izin vererek tarayıcınızın adres satırına http://localhost veya http://localhost:8080 yazarak XAMPP sayfasının açıldığını gözlemleyiniz.

PHP uygulamaları geliştirirken not defteri gibi programlarda kodlama yapılabilir fakat daha gelişmiş editörler kullanmak uygulama geliştirme sürecini daha da kısaltacaktır. PHP uygulamaları geliştirmek için ücretli veya ücretsiz birçok editör mevcuttur. Bunlardan biri de Visual Studio Code editörüdür.

Bir PHP kodu sunucuda çalıştırılır. Çalıştırma sonucunda elde edilen düz HTML kodları tarayıcıya gönderilir. PHP kodları belgenin herhangi bir yerine yazılabilir. Bir PHP kodu **<?php** ile başlar, **?>** ile biter.

```
<?php
// PHP kodları buraya yazılır.
?>
```

PHP dosyalarının ön tanımlı uzantısı ".php"dir. Bir PHP dosyası normalde HTML etiketleri ve PHP kodlarından oluşur.

```
<!DOCTYPE html>
<html>
<body>
```

```
<h1>Hoş Geldiniz</h1>
<?php
echo “Merhaba Dünya”;
?>
</body>
</html>
```

5. Uygulama

PHP kodlarının yerel bilgisayarda çalıştırılması işlemini aşağıdaki adımları takip ederek gerçekleştiriniz.

1. Adım: XAMPP sunucusu ile çalışıyorsanız XAMPP'ın kurulu olduğu klasör içindeki **htdocs** klasörüne, direkt Apache sunucusu ile çalışıyorsanız Apache sunucusunun kurulu olduğu klasör içindeki **htdocs** klasörüne **WebSitem** isminde klasör oluşturunuz.

2. Adım: WebSitem klasörünü Visual Studio Code uygulamasında açarak içine **index.php** adında dosya oluşturunuz.

3. Adım: Oluşturulan index.php dosyasına aşağıdaki kodlamaları yaparak dosyayı kaydediniz.

```
<html>
 <head>
  <title>PHP Denemesi</title>
 </head>
 <body>
 <?php echo ‘<p>Merhaba Dunya</p>’; ?>
 </body>
</html>
```

4. Adım: Web sunucusu çalışıyorken tarayıcınızın adres satırına **http://localhost/websitem** veya **http://localhost:8080/websitem** yazarak oluşturulan sayfayı gözlemleyiniz.

1.3. DEĞİŞKEN VE SABİTLER

Değişkenler verileri saklamak için kullanılan kapsayıcılardır. PHP'de değişken oluşturmanın kuralları şunlardır:

- Bir değişken $ işareti ile başlar, ardından değişken ismi gelir.
- Değişken isimleri harfle veya alt çizgi "_" ile başlamalıdır.
- Değişken isimleri rakam ile başlayamaz.
- Değişken isimleri yalnızca alfa sayısal karakterler ve alt çizgiler içerebilir (A-z, 0-9,_).
- Değişken isimleri büyük / küçük harfe karşı duyarlıdır. $abc ile $ABC birbirinden farklı iki değişkendir.
- PHP dilinin kullandığı komutlar değişken ismi olarak kullanılamaz.

PHP'de kullanılan değişkenlerin bazı özellikleri şunlardır:

- Bir değişkenin değeri en son atanan değere sahiptir.
- Değişkenler "=" ile değişken sol tarafta, atanacak değer sağ tarafta olacak şekilde atama yapılır.

- PHP'de değişken oluştururken saklanacak değerler için int, string, char gibi bildirim yapılmaz.
- PHP değişken türlerinin birbirine dönüştürülmesi kolaydır.

Aşağıdaki kodlamalarda PHP betik dilinde değişken oluşturma, değer aktarma ve yazdırma işlemleri gerçekleştirilmiştir.

```
<?php
$sayi = 10;
$yazi = "Bilişim Teknolojileri";
$karakter = 'X';
echo $sayi;
echo '<br/>';
echo $yazi;
echo '<br/>';
echo $karakter;
echo '<br/>';
?>
```

PHP'de iki değişkeni birleştirmek için "." kullanılır. Aşağıdaki kodlamalarda iki değişkenin birleştirilmesi gerçekleştirilmiştir.

```
<?php
$x = "Bilişim";
$y = "Teknolojileri";
echo $x.$y;
?>
```

Değişkenlerle toplama, çıkarma, çarpma ve bölme işlemi yapılabilmesi için değişken değerlerinin sayı olması gerekir.

```
<?php
 $sayi1 = 10;
 $sayi2 = 5;
 echo $sayi1+$sayi2;
 echo '<br/>';
 echo $sayi1-$sayi2;
 echo '<br/>';
 echo $sayi1*$sayi2;
 echo '<br/>';
 echo $sayi1/$sayi2;
?>
```

Değişkenler farklı türdeki verileri depolayabilir. PHP'de kullanılan veri tipleri şunlardır:

1. String
2. Integer
3. Float
4. Boolean
5. Array
6. Object
7. NULL

Bir string değişken birden çok karakter dizinden oluşur. Değişken değerleri çift veya tek tırnak içine yazılır.

```
<?php
$x = “Merhaba Dünya”;
$y = ‘Merhaba Dünya’;
echo $x;
echo “<br>”;
echo $y;
?>
```

Bir integer değişken ondalık olmayan sayılardan oluşur. Bir integer değişken içinde en az bir basamaklı sayı bulunmalıdır.

```
<?php
$x = 10;
$y = 20;
echo $x;
echo “<br>”;
echo $y;
?>
```

Bir float değişken “.” işareti ile belirlenmiş ondalık sayılardan oluşur.

```
<?php
$x = 10.5;
echo $x;
?>
```

Bir boolean değişken True veya False olmak üzere iki değer alır.

```
<?php
$x = true;
$y = false;
?>
```

Bir array değişken birden çok değeri tek bir değişkende saklar.

```
<?php
$dersler = array(“Türkçe”, “Matematik”, “Tarih”, “Coğrafya”);
?>
```

PHP, nesne yönelimli bir dildir. Sınıflardan türetilen nesneler kullanılarak uygulamalar geliştirilir.

```
<?php
class ogrenci
{
   public $adi;
   public $bolum;
   public $yas;
```

```
    function yazdir()
    {
        echo $this->adi . " " . $this->bolum . " " . $this->yas;
    }
}
$nesne_ogrenci = new ogrenci;
$nesne_ogrenci->adi = "Ahmet";
$nesne_ogrenci->bolum = "Bilişim";
$nesne_ogrenci->yas = 16;
$nesne_ogrenci->yazdir();
?>
```

Bir NULL değişken yalnızca bir değere sahip olan özel bir değişkendir. Bir değişkenin değeri NULL ise değişkene herhangi bir değer atanmadığı anlamına gelir. Eğer bir değişken oluşturulduğunda herhangi bir atama işlemi gerçekleşmemiş ise otomatik olarak NULL değerine sahip olur.

```
<?php
$x = "Bilişim";
$x = null;
echo $x
?>
```

Sabitler, PHP ile yapılan web uygulamalarında değişken olmayan, uygulama boyunca hiç değiştirilmeden sabit kalacak değerleri tanımlamak için kullanılır. Sabitler sadece bir kez tanımlanıp bir daha değiştirilemediği için güvenlidir. Sabitler içinde string, integer, float ve boolean türünde veriler saklanır. Sabitlerin tanımlanması **define()** fonksiyonu ve **const** anahtar kelimesi ile gerçekleştirilir. Sabit isimlendirmesinde değişken isimlendirme kuralı uygulanır ve ismin başında $ işareti kullanılmaz. Geleneksel olarak sabit isimleri büyük harflerle yazılır.

Define() fonksiyonunun kullanımı şu şekildedir:

define(Sabit adı, Sabit değeri, Büyük / küçük harf duyarlı)

Üçüncü parametre isteğe bağlıdır. Sabit adının büyük / küçük harfe karşı duyarlı olup olmadığını belirler.

Define() fonksiyonu ile sabit oluştururken aşağıdaki kodlamalarda olduğu gibi biri sabit adı, diğeri ise sabit değeri olmak üzere iki parametre kullanılır.

```
<?php
define("KURUM", "Millî Eğitim Bakanlığı");
echo KURUM;
echo "<br>";
define("PI",3.14);
echo 2*PI*10;
echo "<br>";
define ("Sayi", "5");
echo Sayi*Sayi;
echo "<br>";
define('meyveler', array('elma', 'muz', 'portakal'));
echo meyveler[1]
?>
```

PHP ile sabit tanımlamanın bir diğer yolu const ifadesidir. Bu tanımlamanın kullanımı aşağıda verilmiştir.

```
const SABİT_ADI = deger;
```

Yukarıdaki tanımlamada sabit adı const anahtar kelimesinden sonra gelir. Tanımlanan sabite "=" operatörü ile değer aktarımı gerçekleştirilir.

```
<?php

const KURUM = "Millî Eğitim Bakanlığı";
const PI=3.14;

echo KURUM;
echo 2*PI*10;

?>
```



Const ile sabit tanımlaması daima bir PHP dosyasının en başında yapılmalıdır. Fonksiyon, karşılaştırma veya döngü içinde tanımlanamaz.

ÖLÇME VE DEĞERLENDİRME

Aşağıdaki soruları dikkatlice okuyarak doğru seçeneği işaretleyiniz.

1. PHP için aşağıdaki ifadelerden hangisi doğrudur?

A) İstemci taraflı betik dildir.
B) Sunucu taraflı betik dildir.
C) Açılımı "Personel Home Page"dir.
D) Tüm kodlar derlendikten sonra çalışır.
E) Ara katman yazılım dilidir.

2. PHP dosya uzantısı aşağıdakilerden hangisidir?

A) .html B) .xml C) .phpx D) .ph E) .php

3. PHP kodları aşağıdaki etiketlerden hangisinin arasına yazılır?

A) <?php>...</?>
B) <php>...</php>
C) <?php...?>
D) <script>...</script>
E) <echo>…</echo>

4. I. Apache ve PHP
II. IIS ve PHP
III. MySQL
IV. Visual Studio Code
V. XAMPP

PHP kodlarını çalıştırabilmek için yukarıdakilerden hangilerinin bilgisayara kurulması gerekir?

A) Sadece IV B) I, IV C) III, IV D) I, II, V E) I, II, III, IV

5. Aşağıdakilerden hangisi PHP'de değişken oluşturmanın doğru yolu <u>değildir</u>?

A) $xDegisken;
B) $x_Degisken = 10;
C) $XDegisken;
D) $x Degisken;
E) $_xDegisken;

6.
```php
<?php
   $sayi  = 5;
   $sayi2 = 10;
   print $sayi . "+". $sayi2;
?>
```

Verilen bu PHP kodlarının çıktısı aşağıdakilerden hangisidir?

A) 5+10 B) 15 C) 5.+10. D) 510 E) Hata verir.

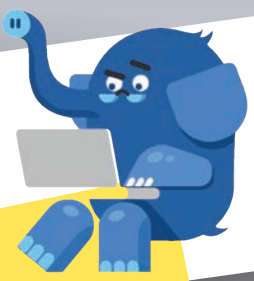
7.
```php
<?php
define("sayi","10");
$sayi=sayi+5;
$sayi=$sayi+2;
echo sayi;
?>
```

Verilen bu PHP kodlarının çıktısı aşağıdakilerden hangisidir?

A) 5 B) 10 C) 15 D) 17 E) Hata verir.

2. ÖĞRENME BİRİMİ

KARAR VE DÖNGÜ YAPILARI



KONULAR

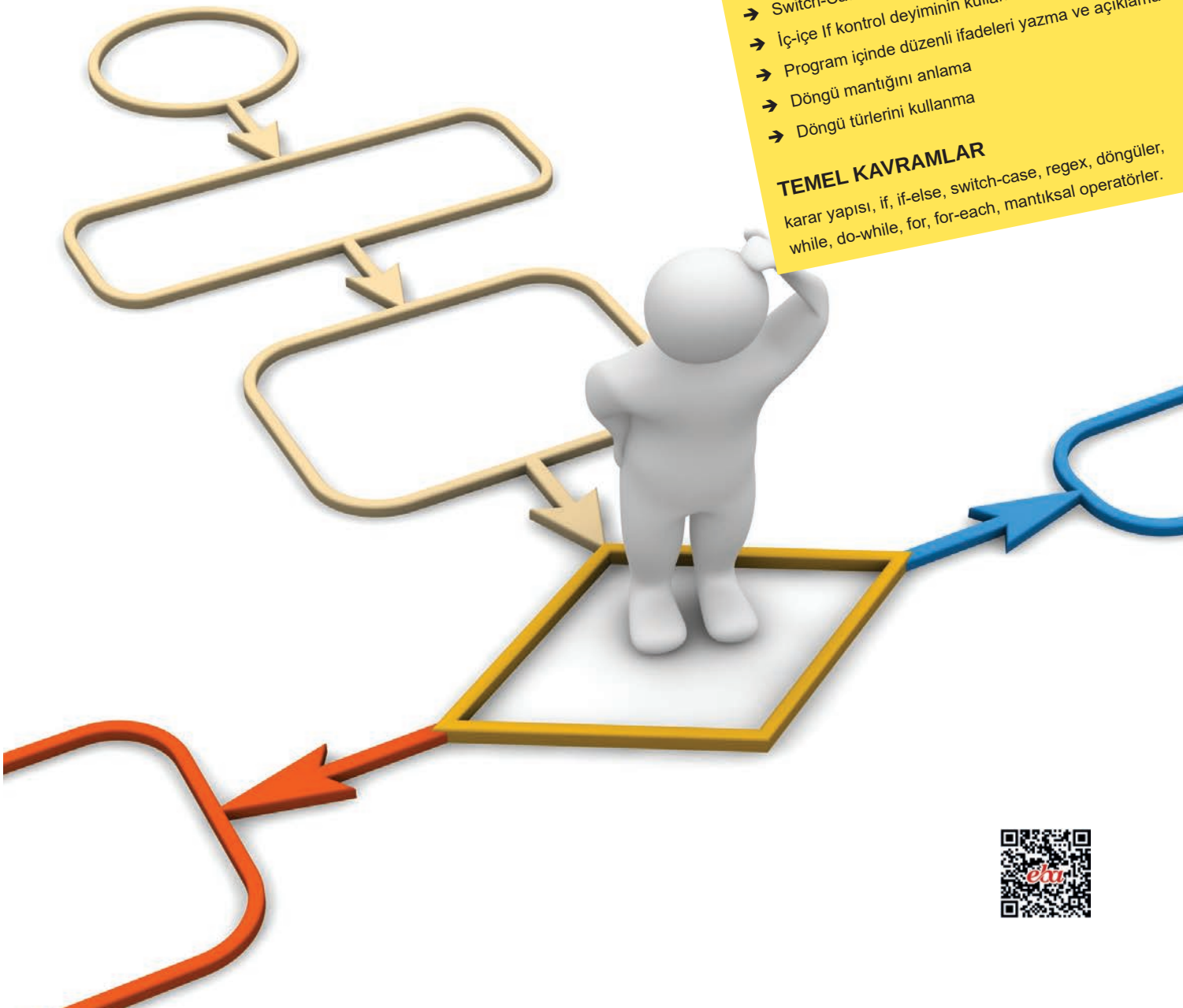
2.1. MANTIKSAL OPERATÖRLER
2.2. KARAR KONTROL DEYİMLERİ
2.3. DÖNGÜ DEYİMLERİ
2.4. VERİLER ÜZERİNDE DÖNGÜ KONTROLLERİ
2.5. DÜZENLİ İFADELER

NELER ÖĞRENECEKSİNİZ?

- Mantıksal operatörleri anlama ve kullanma
- Kontrol deyimlerinin kullanımı
- if – Else kontrol deyiminin kullanımı
- Switch-Case kontrol deyiminin kullanımı
- İç-içe If kontrol deyiminin kullanımı
- Program içinde düzenli ifadeleri yazma ve açıklama
- Döngü mantığını anlama
- Döngü türlerini kullanma

TEMEL KAVRAMLAR

karar yapısı, if, if-else, switch-case, regex, döngüler, while, do-while, for, for-each, mantıksal operatörler.



Hazırlık Çalışmaları

1. **Karar kontrol deyimlerini araştırarak günlük hayatta nasıl kullanıldığını tartışınız.**
2. **PHP'de kullanılan düzenli ifadeleri (regex) araştırarak arkadaşlarınız ile paylaşınız.**
3. **Programlama dillerinde kullanılan döngülerin sağladığı kolaylıkları araştırınız.**
4. **PHP programlama dilinde kullanılan mantıksal operatörleri matematik dersinin mantık konusunda kullanılan operatörlerle karşılaştırınız.**

2.1. MANTIKSAL OPERATÖRLER

Mantıksal (logical) operatörlerin kullanıldığı ifadede ifadenin doğru mu, yanlış mı olduğuna bağlı olarak 0 (false) veya 1 değeri (true) döndürür. Mantıksal operatörler, yazılan programlarda genellikle karar vermede yaygın olarak kullanılan özel karakterlerdir (Tablo 2.1).

***Tablo 2.1:** Mantıksal Operatörler*

Operatör	Ad	Örnek	Sonuç
&& - And	Ve	$a= 6 ve $b = 3 ; echo(($a == 6) && ($b> 4)) ;	0'a eşittir (0 (false) değerinde ekranda hiçbir şey görülmez.).
|| - Or	Veya	$a= 6 ve $b = 3 ; echo (($a ==6) || ($b> 7))	1'e eşittir.
!	Değilse	$c = 15 ; echo !($c == 15),	0'a eşittir.
xor	XOR	$a= 6 ve $b = 3 ; echo(($a == $b) XOR ($b<= $a));	1'e eşittir.

2.1.1. Ve (And-&&) Operatörü

Karşılaştırılan iki veya daha fazla koşulun tümünün doğru olduğu durumlarda sonucu doğru olarak geriye döndüren operatördür.

1. Örnek

```
<?php
    $a = true;
    $b = true;
    echo $a && $b; // Çıktı: 1 yani true (doğru)
    echo $a and $b; // && işaretinin yerine and ifadesi de kullanılabilir.
?>
```

And ile && operatörleri aynı karşılaştırmayı yapar ancak öncelikleri farklıdır.

2. Örnek

```
<?php
$a = (true and false); // 0 (false)
echo $a. " (false) <br>";
// Parantez kaldırılıp denenebilir.
$a = true and false; // 1 (true)
echo $a. " (true) ";
?>
```

Son satırda true dönmesinin nedeni işlem önceliğidir çünkü = (atama operatörü), and operatöründen daha önceliklidir.

Sıra Sizde

Aşağıdaki kodlarla index.php oluşturup tarayıcıda görüntüleyerek sonucu arkadaşlarınız ile paylaşınız.

```
<?php
    $a = 15;
    $b =15;
    $c = 16;
    echo ($a == $b) && ($c > $b);
?>
```



Değerlendirme

Çalışmanız diğer sayfada yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

KONTROL LİSTESİ

Aşağıda listelenen ölçütlerden öğrencide gözlenen davranış için Evet, gözlenemeyen davranış için Hayır kutucuğunun altına (X) işareti koyarak işaretleyiniz.

Ölçütler	Evet	Hayır
1. Server içinde index.php dosyası oluşturuldu.		
2. Ve (AND - &&) operatörünü program içinde doğru kullandı.		
3. Çalışmada iş sağlığı ve güvenliği kurallarına dikkat edildi.		
4. Zaman verimli kullanıldı.		

2.1.2. Veya (Or - ||) Operatörü

Karşılaştırılan iki veya daha fazla koşuldan en az bir tanesinin doğru olduğu durumlarda sonucu doğru olarak geriye döndüren operatördür.

3. Örnek

```
<?php
    $a = false;
    $b =true;
     echo $a || $b; // Çıktı: 1 yani true (iki değerden birinin doğru olması
                                              yeterlidir.)
     echo “<br>”
     echo $a OR $b; // || yerine Or ifadesi de yazılabilir.
  ?>
```

Or ile || operatörleri aynı karşılaştırmayı yapar ancak öncelikleri farklıdır.

4. Örnek

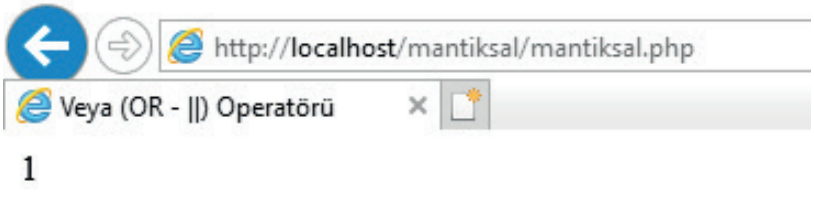
```
<?php
$a = (false or true); // 1 (true)
echo $a. “ (true) <br>”;
// aynı işlemi parantezleri kaldırarak deneyiniz.
$a = false or true; // 0 (false)
echo $a. “ (false) ”;
?>
```

Son satırda false dönmesinin nedeni de işlem önceliğidir. İşlem önceliği tablosuna bakılırsa = (atama) operatörünün Or operatöründen daha öncelikli olduğu görülür.

Sıra Sizde

Aşağıdaki kodlarla index.php oluşturup tarayıcıda görüntüleyerek sonucu arkadaşlarınız ile paylaşınız.

```
<?php
     $a = 51;
     $b = 61;
     echo ($a == $b) || ($a < $b);
   ?>
```



1

Değerlendirme

Çalışmanız aşağıda yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

KONTROL LİSTESİ

Aşağıda listelenen ölçütlerden öğrencide gözlenen davranış için Evet, gözlenemeyen davranış için Hayır kutucuğunun altına (X) işareti koyarak işaretleyiniz.

Ölçütler	Evet	Hayır
1. Server içinde index.php dosyası oluşturuldu.		
2. Veya (OR - ||) operatörünü program içinde doğru kullandı.		
3. Çalışmada iş sağlığı ve güvenliği kurallarına dikkat edildi.		
4. Zaman verimli kullanıldı.		

2.1.3. XOR Operatörü

Or operatörü gibidir. Or operatöründen farkı ise şartlardan sadece birinin doğru olması durumunda işlem yaparken iki veya daha fazla koşulun doğru olması durumunda işlem yapmayarak bunu geçersiz olarak kabul eder.

5. Örnek

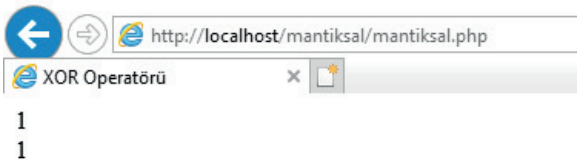
```
<?php
$a = (false xor false); // 0 (false)
echo $a. " (false) <br> ";
$a = (true xor false); // 1 (true)
echo $a. " (true) <br> ";
$a = false xor true; // 0 (false)
echo $a. " (false) ";
?>
```

Xor operatörü = (atama) operatöründen düşük öncelikli olduğu için son satırda sonuç false olmuştur.

Sıra Sizde

Aşağıdaki kodlarla index.php oluşturup tarayıcıda görüntüleyerek sonucu arkadaşlarınız ile paylaşınız.

```
<?php
    $a = 5;
    $b = 5;
echo ($a == $b) XOR ($a < $b); // Çıktı: 1 (true )
echo "<br>";
echo ($a == $b) || ($a <= $b); // Çıktı: 1 (true)
  ?>
```



Değerlendirme

Çalışmanız aşağıda yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

KONTROL LİSTESİ

Aşağıda listelenen ölçütlerden öğrencide gözlenen davranış için Evet, gözlenemeyen davranış için Hayır kutucuğunun altına (X) işareti koyarak işaretleyiniz.

Ölçütler	Evet	Hayır
1. Server içinde index.php dosyası oluşturuldu.		
2. XOR operatörünü program içinde doğru kullandı.		
3. Çalışmada iş sağlığı ve güvenliği kurallarına dikkat edildi.		
4. Zaman verimli kullanıldı.		

2.1.4. Değil (!) Operatörü

Değişkenin durumunu tersine çevirir. Bir başka ifadeyle durum doğruysa sonuç "yanlış" olarak, yanlış ise de sonucun doğru olarak değer döndürmesini sağlar.

6. Örnek

```
<?php
    $a = false;
    $b = true;
    echo !$a || !$b;
/* Çıktı: 1( true )
$a değişkenin değeri false ve değil true olur, $b değişkenin değeri ise true
ve değil false olur.
İki değişken arasında OR operatörü olduğundan bir değerin doğru olması
sonucun 1 (true) olmasını sağlar.*/
    ?>
```

Sıra Sizde

Aşağıdaki kodlar çalıştırıldığında çıktılarının 1 (true) mi yoksa 0 (false) mı olduğunu karşısındaki boşluğa yazınız.

`$a=5;`
`$b=3;`
`echo ($a>$b and $b>2);`	

`$a=6;`
`$b=6;`
`echo ($a==$b and $a<10);`	

`$a=2;`
`$b=4;`
`echo ($a==$b or $a>$b);`	

`$a=2;`
`$b=4;`
`echo ($a!=$b or $a>8);`	

`$a=9;`
`$b=7;`
`echo (!($a>$b));`	

`$a=5;`
`$b=5;`
`echo (!($a>=$b and $a<1));`	

`$a=10;`
`$b=15;`
`echo(!(($a>=$b xor $a>1) and ($a>=10)) ||(!($a>$b)));`	

`$anahtar=3>4;` `echo $anahtar;`	

Değerlendirme

Çalışmanız aşağıda yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

KONTROL LİSTESİ

Aşağıda listelenen ölçütlerden öğrencide gözlenen davranış için Evet, gözlenemeyen davranış için Hayır kutucuğunun altına (X) işareti koyarak işaretleyiniz.

Ölçütler	Evet	Hayır
1. And operatörünü program içinde doğru kullandı.		
2. Or operatörünü program içinde doğru kullandı.		
3. Xor operatörünü program içinde doğru kullandı.		
4. Değil operatörünü program içinde doğru kullandı.		
5. And, Or, Xor ve Değil operatörlerinin program içinde beraber nasıl kullanıldığı öğrenildi.		
6. Çalışmada iş sağlığı ve güvenliği kurallarına dikkat edildi.		
7. Zaman verimli kullanıldı.		

2.1.5. Mantıksal Operatörlerin Doğruluk Tablosu

Programlamada doğru ve yanlışlar, “1” ve “0” ile de ifade edilir. 1’leri “doğru” ve 0’ları da “yanlış” kabul ederek mantıksal operatörlerin alabileceği tüm değerler Tablo 2.2’de gösterilmiştir.

Tablo 2.2: *Mantıksal Operatörlerin Doğruluk Tablosunda Gösterilişi*

ve (AND - &&)			veya (OR - ||)			XOR			Değil (!)	
a	b	a&&b	a	b	a || b	a	b	a XOR b	a	! b
0	0	0	0	0	0	0	0	0	0	1
0	1	0	0	1	1	0	1	1	1	0
1	0	0	1	0	1	1	0	1		
1	1	1	1	1	1	1	1	0		

2.2. KARAR KONTROL DEYİMLERİ

Tüm programlama dillerinde uygulama geliştirilirken en sık kullanılan yapılardan bir tanesi karar kontrol deyimleridir. Verilen bir koşula veya koşullar dizisine bağlı olarak programın ne tür bir yol ile ilerleyeceğini belirleyen yapılardır. Örneğin “Bugün hava güzel olursa ve Ahmet gelirse pikniğe gideceğiz, Ahmet gelmez de Fatma gelirse sinemaya gideceğiz. Eğer ikisi de gelmezse parka gideceğim.” şeklindeki bir olayı gerçekleştirmek için karar kontrol deyimleri kullanılır.

PHP dilinde iki adet karar kontrol deyimi vardır. Bunlar;

- if–else karar kontrol deyimi,
- Switch–Case karar kontrol deyimidir.

2.2.1. if-else Kontrol Deyimi

if Yapısı

İf kontrol yapısı ile oluşturulan kod parçacığında istenen koşul veya koşullar sağlandığında if kontrol yapısı içindeki komutlar çalıştırılır.

```
if (koşul) //Koşul sağlandığında çalıştırılacak satır;
if (koşul) {
//Şart sağlandığında çalıştırılacak satırlar;
}
```

Önemli

Çalıştırılacak satır birden fazla ise satırlar küme parantezi içine yazılır.

1. Uygulama

Tanımlı olan kullanıcı adının ve şifresinin sırasıyla ***gazimtal*** ve ***123456*** olup olmadığını kontrol eden programı aşağıdaki adımları takip ederek yazınız.

1. Adım: index.php dosyasını server içinde oluşturunuz.

2. Adım: kullanici_adi ve sifre şeklinde iki adet değişken oluşturarak "gazimtal" ve "123456" değerlerini atayınız.

3. Adım: kullanici_adi ile sifre değişkenlerini if yapısı ile kontrol edip doğru ise ekrana Görsel 2.1'deki gibi "Giriş Başarılı Bir Şekilde Yapıldı" ifadesini italik ve koyu bir şekilde yazdırınız.

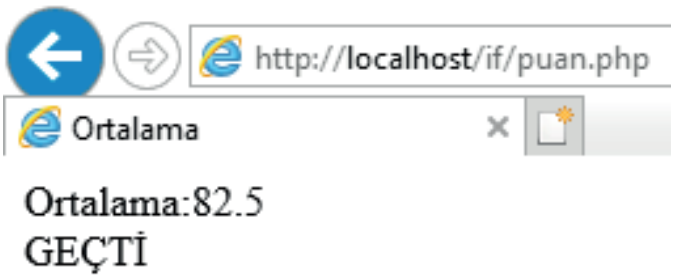
```
<?php
     $kullanici_adi="gazimtal";
     $sifre="123456";
     if  ($kullanici_adi=="gazimtal"  and   $sifre=="123456")
     echo  "<strong><em> Giriş Başarılı Bir Şekilde Yapıldı </strong></em>";
  ?>
```



Görsel 2.1: Giriş paneli

Sıra Sizde

Tanımlı olan iki adet yazılı ve performans puanının ortalamasını bulan ve ortalaması 50 ve 50'den yüksekse Görsel 2.2'deki gibi ekran çıktısı veren programı yazınız.



Görsel 2.2: Ortalama puan

Değerlendirme

Çalışmanız diğer sayfada yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

KONTROL LİSTESİ

Aşağıda listelenen ölçütlerden öğrencide gözlenen davranış için Evet, gözlenemeyen davranış için Hayır kutucuğunun altına (X) işareti koyarak işaretleyiniz.

Ölçütler	Evet	Hayır
1. Server içindeki if klasöründe index.php dosyası oluşturuldu.		
2. Kullanılacak değişkenler belirlendi.		
3. Değişkenlerin değerleri belirlendi.		
4. Verilen değerlere göre ortalama alınıp if deyimi programda gerekli yerlere yazıldı.		
5. Çalışmada iş sağlığı ve güvenliği kurallarına dikkat edildi.		
6. Zaman verimli kullanıldı.		

else Yapısı

if yapısında yer alan koşul doğru ise belirli işlem satırı veya satırları çalıştırılmakta, yanlış ise herhangi bir işlem yapılmamaktadır. if yapısındaki koşul yanlış ise belirli işlem satırlarını çalıştırmak için else deyimi kullanılır.

```
if (koşul) işlem satırı; // koşul doğru ise çalışacak satır
else işlem satırı; // koşul  yanlış ise çalışacak satır
```

2. Uygulama

if ($durum) $durum değişkeninde bir değer varsa ya da $durum değişkeni true değerini almış ise if-else yapısının hangi blokunun çalışacağını tahmin edip adımları tek tek yazarak serverda çalıştırıp karşılaştırmak için aşağıdaki adımları takip ediniz.

1. Adım: index.php dosyasını server içinde oluşturunuz.

2. Adım: PHP kodlarını yazmak için gerekli kod blokunu yazınız.

3. Adım: $durum değişkenine -5 değerini atayarak diğer sayfadaki kod blokunun hangi satırını çıktı vereceğini tahmin ediniz.

```
<?php
$durum=-5;
if  ($durum)
      echo “durum değişkeni true’dur veya sıfırdan farklı bir değere sahiptir.”;
else
      echo “durum değişkeni false’tur veya sıfırdır.”;
?>
```

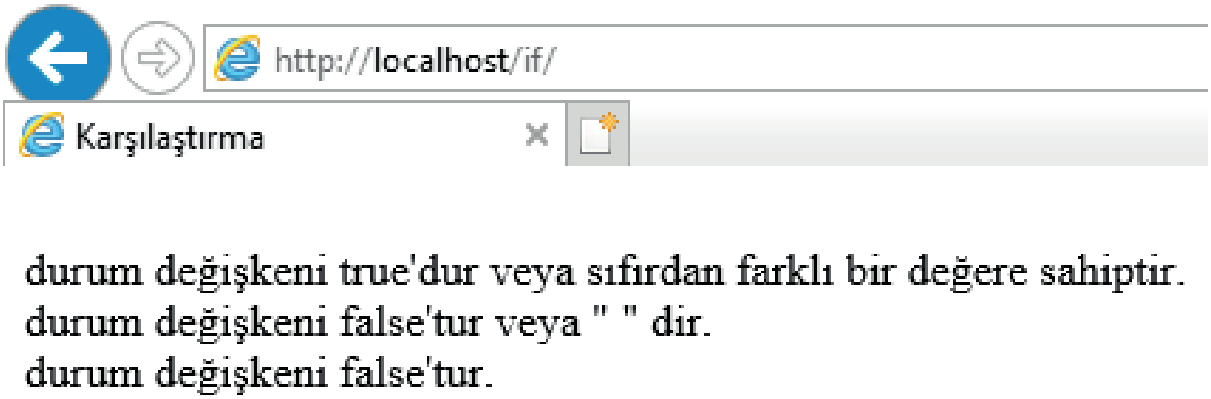
4. Adım: $durum değişkenine “ “ değerini atayarak aşağıdaki kod blokunun hangi satırını çıktı vereceğini tahmin ediniz.

```
$durum="";
   if  ($durum)
     echo "durum değişkeni true'dur veya \" \" den farklı bir değere sahiptir.";
   else
        echo  "durum  değişkeni  false'tur  veya  \"  \ "  dir."
```

5. Adım: $durum değişkenine false değerini atayarak aşağıdaki kod blokunun hangi satırını çıktı vereceğini tahmin ediniz.

```
$durum=false;
           if  ($durum)
                 echo  "durum  değişkeni  true'dur.";
           else
                 echo  "durum  değişkeni  false'tur.
```

6. Adım: 3. Adım, 4. Adım ve 5. Adımdaki kod bloklarının hepsini birlikte çalıştırarak cevaplarınızı Görsel 2.3 ile karşılaştırınız.



Görsel 2.3: Karşılaştırma

Sıra Sizde

Tanımlı olan kullanıcı adını ve şifresini kontrol ederek ifadeler doğru ise ekrana **Başarılı Bir Şekilde Giriş Yaptınız**, ifadeler doğru değilse **Hatalı Giriş Yaptınız** çıktısını veren programı yazınız.

Değerlendirme

Çalışmanız aşağıda yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

KONTROL LİSTESİ

Diğer sayfada listelenen ölçütlerden öğrencide gözlenen davranış için Evet, gözlenemeyen davranış için Hayır kutucuğunun altına (X) işareti koyarak işaretleyiniz.

Ölçütler	Evet	Hayır
1. Server içindeki if klasörü içinde sayi.php dosyaları oluşturuldu.		
2. Kullanılacak değişkenler belirlendi.		
3. Değişkenlerin değerleri belirlendi.		
4. Tanımlanan değerlere göre kullanıcının bilgilerini kontrol eden if-else deyimi programda gerekli yere yazıldı.		
5. Programların çalışıp çalışmadığı kontrol edildi.		
6. Çalışmada iş sağlığı ve güvenliği kurallarına dikkat edildi.		
7. Zaman verimli kullanıldı.		

Sıra Sizde

Tanımlanan 3 sayı değişkenin hepsi 0'dan büyük ve çift ise ekrana "Sayılar sıfırdan büyük ve hepsi çift", aksi hâlde "Sıfırdan küçük sayı veya çift olmayan sayı var" yazdıran programı yazınız.

Değerlendirme

Çalışmanız diğer sayfada yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

KONTROL LİSTESİ

Aşağıda listelenen ölçütlerden öğrencide gözlenen davranış için Evet, gözlenemeyen davranış için Hayır kutucuğunun altına (X) işareti koyarak işaretleyiniz.

Ölçütler	Evet	Hayır
1. Server içindeki if klasöründe sayi.php dosyası oluşturuldu.		
2. Kullanılacak 3 adet sayı değişkeni belirlendi.		
3. Değişkenlerin değerleri belirlendi.		
4. Tanımlanan sayıların kontrolünü yapan if-else şartı programda gerekli yere yazıldı.		
5. Programların çalışıp çalışmadığı kontrol edildi.		
6. Çalışmada iş sağlığı ve güvenliği kurallarına dikkat edildi.		
7. Zaman verimli kullanıldı.		

elseif Yapısı

Yazılan programlarda bazen birden fazla koşulun olduğu durumlar da ortaya çıkar. Bu gibi durumlarda elseif yapısı kullanılır. Bu yapıda ilk koşul if, aradaki koşullar elseif, tüm koşulların yanlış olma durumunda ise else şeklinde kullanım sırası vardır. Örneğin bir değişkenin değeri 100 iken bazı işlemler, 150 iken bazı işlemler yapılmak istenebilir. Böyle durumlarda yeni şartlar ekleyebilmek için elseif ifadesi kullanılır.

```
if (koşul) işlem satırı;
elseif (koşul) işlem satırı;// if'teki koşul yanlış ise elseif'teki koşul kontrol edilir.
else işlem satırı;// if ve elseif'teki koşullar yanlış ise else satırı çalışır.
```

3. Uygulama

Tanımlı olan kullanıcı adını ve şifresini kontrol ederek

1. Kullanıcı adı ve şifre doğru ise ekrana "Programa hoş geldiniz",
2. Kullanıcı adı doğru şifre yanlış ise ekrana "Hatalı şifre girdiniz",
3. Kullanıcı adı yanlış şifre doğru ise ekrana "Hatalı kullanıcı adı girdiniz",
4. Bunun dışında ise "Kullanıcı adı ve şifre hatalı" şeklinde ekran çıktısı verecek programı adımları takip ederek yazınız.

1. Adım: index.php dosyasını server içinde oluşturunuz.

2. Adım: kullanici_adi, kullanici_adi1, sifre ve sifre1 şeklinde dört adet değişken oluşturarak kullanici_adi ve sifre değerlerine sabit değerler atayınız.

3. Adım: kullanici_adi1 ve sifre1 değişkenlerinin dışarıdan alındığını kabul ederek değerler atayınız.

```
#Sabit  değişkenler
   $kullanici_adi="gazimtal";
   $sifre="123456";
  #Dışarıdan  alınacak  değişkenler, önceki iki değerinden doğru giriniz.
    $kullanici_adi1="gazimtal";
    $sifre1="123456";
```

4. Adım: kullanici_adi ile sifre değişkenlerini if yapısı ile kontrol edip doğru ise ekrana Görsel 2.4'teki gibi "Programa hoş geldiniz" ifadesini italik ve koyu bir şekilde yazdırınız.

```
#Sabit değişkenler
       $kullanici_adi="gazimtal";
       $sifre="123456";
       #Dışarıdan alınacak değişkenler,önceki iki değerinden doğru giriniz.
       $kullanici_adi1="gazimtal";
       $sifre1="123456";
                if ($kullanici_adi==$kullanici_adi1 and  $sifre==$sifre1)
                echo "<strong><em> Programa Hoş Geldiniz </strong></em>";
```



Görsel 2.4: Programa hoş geldiniz ekranı

5. Adım: elseif yapısı ile kullanıcı adını ve şifreyi kontrol edip hatalı değerleri ekrana yazdırınız.

```
elseif($kullanici_adi==$kullanici_adi1  and    $sifre!=$sifre1)
        echo  "<strong><em>  Hatalı  Şifre  Girdiniz    </strong></em>";
elseif($kullanici_adi!=$kullanici_adi1  and    $sifre==$sifre1)
        echo  "<strong><em>  Hatalı  Kullanıcı  Adı  Girdiniz    </strong></em>";
```

6. Adım: Kullanıcı adı ve şifre aynı, ad yanlış girilmiş ise else ifadesini kontrol edip hatalı değerleri ekrana yazdırınız.

```
else
        echo  "<strong><em>  Kullanıcı  Adı  ve  Şifre  Hatalı  </strong></em>";
```

Sıra Sizde

Tanımlanan sıcaklık bilgisini dikkate alarak aşağıda verilen şartlara göre hava durumu hakkında bilgi veren programı yazınız.

(-30) – (-10): Çok Soğuk, (-10) – (-0): Soğuk, (1 – 10): Serin, (11-30): Sıcak, (30'dan büyükse): Çok Sıcak

Değerlendirme

Çalışmanız aşağıda yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

KONTROL LİSTESİ

Aşağıda listelenen ölçütlerden öğrencide gözlenen davranış için Evet, gözlenemeyen davranış için Hayır kutucuğunun altına (X) işareti koyarak işaretleyiniz.

Ölçütler	Evet	Hayır
1. Server içindeki if klasöründe sicaklik.php dosyası oluşturuldu.		
2. Kullanılacak sıcaklık değişkeni belirlendi.		
3. Değişkenin değeri belirlendi.		
4. Tanımlanan değerlere göre sıcaklık bilgisini kontrol eden if - elseif - else deyimi programda gerekli yere yazıldı.		
5. Programların çalışıp çalışmadığı kontrol edildi.		
6. Çalışmada iş sağlığı ve güvenliği kurallarına dikkat edildi.		
7. Zaman verimli kullanıldı.		

Sıra Sizde

Tanımlanan iki yazılı ve performans puanının ortalaması bulunduktan sonra aşağıdaki değerlere göre ortalama puanının hangi nota denk geldiğini bulan programı yazınız.

0-24:Sıfır	25-44:Bir	45-54:İki	55-69:Üç	70-84:Dört	85-100:Beş

Değerlendirme

Çalışmanız aşağıda yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

KONTROL LİSTESİ

Aşağıda listelenen ölçütlerden öğrencide gözlenen davranış için Evet, gözlenemeyen davranış için Hayır kutucuğunun altına (X) işareti koyarak işaretleyiniz.

Ölçütler	Evet	Hayır
1. Server içindeki if klasöründe harfsistem.php dosyası oluşturuldu.		
2. Kullanılacak 4 adet değişken belirlendi.		
3. Değişkenlerin değerleri belirlendi.		
4. Bulunan ortalamaya göre karşılık gelen rakamı yazı ile yazacak if-else-if-else deyimi programda gerekli yere yazıldı.		
5. Programların çalışıp çalışmadığı kontrol edildi.		
6. Çalışmada iş sağlığı ve güvenliği kurallarına dikkat edildi.		
7. Zaman verimli kullanıldı.		

İç İçe İf Kullanımı

Bazı durumlarda bir şartın gerçekleşmesi hâlinde başka şartların da kontrol edilmesi gerekir. Bunun için if yapısının içinde bir veya birden fazla if yapısı kullanılabilir.

```
if  ( şart1  )
{
if  ( şartA )   //şart1 ve şartA gerçekleştiğinde yapılacak işlemler;
elseif  ( şartB )   //şart1 ve şartB gerçekleştiğinde yapılacak işlemler;
else //şart1 gerçekleşip şartA ve şartB gerçekleşmediğinde yapılacak işlemler;
}
elseif  ( şart2  )
{
if  ( şartC  ) //şart2  ve şartC  gerçekleştiğinde yapılacak işlemler;
elseif  ( şartD )   //şart2 ve şartD gerçekleştiğinde yapılacak işlemler;
else  //şart2 gerçekleşip şartC ve şartD gerçekleşmediğinde yapılacak işlemler;
}
else
{
//şart1  ve şart2  gerçekleştiğinde yapılacak işlemler;
}
```

4. Uygulama

Tanımlı iki yazılı ve performans değişkeninin ortalaması hesaplanıp ekrana "Geçti" veya "Kaldı" şeklinde sonuç yazdırılması planlanmaktadır. Ortalama hesaplaması yapılırken aşağıdaki şartlar göz önünde bulundurulacak şekilde programı adımları takip ederek yazınız.

Şartlar:

Puanlardan herhangi biri tanımlanmamışsa ***Puanlardan biri eksik girilmiş*** hata mesajı ekrana yazdırılacaktır.

Puanlardan herhangi biri 0'dan küçük veya 100'den büyük sayı girilmiş ise **Hatalı puan girilmiş** hata mesajı ekrana yazdırılacaktır.

1. Adım: index.php dosyasını server içinde oluşturunuz.

2. Adım: $yazili1, $yazili2, $performans1, $performans2 şeklinde dört adet değişken oluşturarak sabit değerler atayınız.

```
<?php
$yazili1=50;
$yazili2=50;
$performans1=60;
$performans2=60;
?>
```

3. Adım: Yazılı ve performans puanlarını kontrol ederek eksik puan var ise ekrana ***Puanlardan biri eksik*** **girilmiş** hata mesajını yazdırınız.

```
   if($yazili1== " "||$yazili2==" "||$performans1==" "||$performans2==" ")
{
        echo "<em><strong>Puanlardan  biri  eksik  girilmiş  </em></strong>";
}
```

4. Adım: Eksik puan yok ise atanan puanların 0'dan küçük veya 100'den büyük olup olmadığını kontrol ederek ekrana ***Hatalı puan girilmiş*** hata mesajını yazdırınız.

```
else
{
        if($yazili1  <  0  ||  $yazili2>  100||  $yazili2  <  0  ||  $yazili2>
100  ||  $performans1  <  0  ||  $performans1>  100    ||  $performans2  <  0
||  $performans2>  100)
        {
         echo "<em><strong>Hatalı  Puan  Girilmiş</em></strong>";
        }
}
```

5. Adım: Hatalı girilen puan yok ise ortalamayı buldurup ortalama 50'den büyük ve eşitse ekrana "Geçti", 50'den küçükse "Kaldı" mesajını yazdırınız.

```
else
 {          $ortalama=($yazili1+$yazili2+$performans1+$performans2)/4;
    if($ortalama>=50)  echo "<em><strong>Geçti</em></strong>";
    else  echo "<em><strong>Kaldı</em></strong>";
 }
```

Sıra Sizde

İç içe if'leri kullanarak sisteme kayıt yapmak için girilen birinci şifre ile ikinci şifrenin eşit olup olmadığını kontrol eden programı yazınız.

Değerlendirme

Çalışmanız aşağıda yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

KONTROL LİSTESİ

Aşağıda listelenen ölçütlerden öğrencide gözlenen davranış için Evet, gözlenemeyen davranış için Hayır kutucuğunun altına (X) işareti koyarak işaretleyiniz.

Ölçütler	Evet	Hayır
1. Server içindeki if klasöründe sifre.php dosyaları oluşturuldu.		
2. Kullanılacak birinci şifre değişkenleri ve değerleri belirlendi.		
3. Kullanılacak ikinci şifre değişkenleri ve değerleri belirlendi.		
4. Tanımlanan değerlere göre şifrelerin birbiri ile uyuşup uyuşmadığını kontrol eden iç içe if'ler programda gerekli yere yazıldı.		
5. Programların çalışıp çalışmadığı kontrol edildi.		
6. Çalışmada iş sağlığı ve güvenliği kurallarına dikkat edildi.		
7. Zaman verimli kullanıldı.		

if-elseif-else Alternatif Söz Dizimi

Alternatif söz diziminde küme parantezi yerine iki nokta üst üste (:), if-elseif-else blokunun bittiği yere ise endif (Tablo 2.3) eklenerek kontrol deyimi sonlandırılır.

***Tablo 2.3:** if-elseif-else Alternatif Söz Dizimi*

Normal Kullanım	Alternatif Kullanım

`if (koşul)`
`{işlem satırı;}`
`elseif (koşul)`
`{işlem satırı;}`
`else`
`{işlem satırı ;}`
	`if (koşul) :`
	`işlem satırı;`
	`elseif (koşul) :`
	`işlem satırı;`
	`else :`
	`işlem satırı ;`
	`endif;`

5. Uygulama

Tanımlı olan sayı değişkeninin tek mi, çift mi olduğunu bulan programı aşağıdaki adımları takip ederek yazınız.

1. Adım: index.php dosyasını server içinde oluşturunuz.

2. Adım: $sayi değişkenini tanımlayarak 125 değerini atayınız.

```
<?php
        $sayi=125;
?>
```

3. Adım: $sayi değişkeninin tek veya çift olup olmadığını if-else blokunda alternatif söz dizimi kullanarak kontrol ediniz.

```
if($sayi%2==0):
                echo  "$sayi'sı  çifttir.";
 else:
                echo  "$sayi'sı  tektir.";
 endif;
```

Sıra Sizde

Bir hastanedeki kapalı otoparka park eden personel aracının park ücreti 1 saati 1 TL, muayeneye gelen hasta aracının park ücreti ise 1 saati 2 TL'dir. Bununla birlikte birinci saatten sonraki her saat için personelin %5, hastanın ise %7 fazla ücret ödemesi gerekmektedir. Buna göre araç türü ve otoparkta kalınan saat bilgisi tanımlandıktan sonra otoparka ödenecek toplam ücreti ekrana yazan programı yazınız (if yapısında alternatif yazımı kullanınız.).

Değerlendirme

Çalışmanız aşağıda yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

KONTROL LİSTESİ

Aşağıda listelenen ölçütlerden öğrencide gözlenen davranış için Evet, gözlenemeyen davranış için Hayır kutucuğunun altına (X) işareti koyarak işaretleyiniz.

Ölçütler	Evet	Hayır
1. Server içindeki if klasöründe otopark.php dosyası oluşturuldu.		
2. Kullanılacak araç türü, saat değişkenleri ve değerleri belirlendi.		
3. İf-else deyimi yerine program içinde ödenecek toplam ücreti belirlemede alternatif yazım kuralı kullanıldı.		
4. Programların çalışıp çalışmadığı kontrol edildi.		
5. Çalışmada iş sağlığı ve güvenliği kurallarına dikkat edildi.		
6. Zaman verimli kullanıldı.		

Üçlü Operatör Kullanımı

Tek satırlık kodlarda if-else deyimleri yerine kullanılır.

İfade(koşul)?(Koşul Doğru ise):(Koşul Yanlış ise);

Bu kullanımda ifade alanı, değişkeni veya ekrana yazdırma komutunu temsil eder. Soru işareti (?) koşul doğru ise çalışacak kısımdır. İki nokta üst üste (:) koşul yanlış ise çalışacak kısımdır (Tablo 2.4).

***Tablo 2.4:** Üçlü Operatör Kullanımı*

Normal Kullanım	Üçlü Operatör Kullanımı

<?php
$sayi=120;
if($sayi%2==0)
{
echo "$sayi'sı çifttir.";
}
else
{
echo "$sayi'sı tektir.";
}
?>
	<?php
	$sayi=120;
	echo($sayi%2==0)?"$sayi'sı çifttir.":"$sayi'sı tektir.";
	?>

	<?php $sayi=120; $sonuc=($sayi%2==0)?"$sayi'sı çifttir.":"$sayi'sı tektir."; echo $sonuc; ?>

2.2.2. Switch-Case Kontrol Deyimi

Switch-case, tanımlanmış bir değişkenin program içinde belirtilen birçok durumdan hangisine uyuyorsa o durum altında bulunan kod bloklarını çalıştıran kontrol deyimidir. Switch-case ile yazılan programların hepsi if-elseif-else bloku ile yazılabilir.

```
switch ($değişken) {
case koşul1 :
//Komutlar;
break;
case  koşul2 :
//Komutlar;
break;
.....................
.....................
default:
//Komutlar;
}
```

Hiçbir durum ile eşleşme olmazsa default ifadesinde belirtilen kodlar çalışır. “break;” komutu, kodlar çalıştıktan sonra switch-case ifadesinden çıkmayı sağlar.

6. Uygulama

Tanımlı olan iki sayı ve işleme göre toplama, çıkarma, bölme ve çarpma yapan programı aşağıdaki adımları takip ederek yazınız.

1. Adım: index.php dosyasını server içinde oluşturunuz.

2. Adım: $sayi1, $sayi2 ve $islem değişkenlerini tanımlayınız.

```
<?php
   $sayi1=15;
   $sayi2=5;
   $islem= "+";//sırasıyla  işlemi  "-","*","/",  şeklinde  değiştiriniz.
?>
```

3. Adım: $islem değişkenin değerine göre toplama, çıkarma, çarpma ve bölme için gerekli kodları yazınız.

```
switch  ($islem){
              case  "+":  $sonuc=$sayi1+$sayi2;  break;
              case  "-":  $sonuc=$sayi1-$sayi2;  break;
              case  "*":  $sonuc=$sayi1*$sayi2;  break;
              default:  $sonuc=$sayi1/$sayi2;
        }
        echo  "$sayi1  $islem  $sayi2=$sonuc";
```

Sıra Sizde

Ayşe'ye babası alışveriş yapması için 2500 TL vermiştir. Ayşe A mağazasına giderek kendisine 100 TL'den 2 adet kazak, 75 TL'den 2 adet şapka, 250 TL'den 2 adet etek, 150 TL'den 1 adet çanta ve 300 TL'den 1 adet spor ayakkabısı almıştır. A mağazası alışveriş tutarına göre;

500 TL'den az alışveriş için %2	500-1000 TL arası alışveriş için %5	1001-2000 TL arası alışveriş için %8	2001 TL üzeri alışveriş için %8

indirim uygulamaktadır. Ayşe'nin yaptığı alışverişten sonra mağazanın uygulayacağı indirimi ve elinde kalan parasını ekrana yazdıran programı yazınız.

Değerlendirme

Çalışmanız aşağıda yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

KONTROL LİSTESİ

Aşağıda listelenen ölçütlerden öğrencide gözlenen davranış için Evet, gözlenemeyen davranış için Hayır kutucuğunun altına (X) işareti koyarak işaretleyiniz.

Ölçütler	Evet	Hayır
1. Server içindeki switch klasöründe harcama.php dosyası oluşturuldu.		
2. Kullanılacak gerekli değişkenler ve alınan malzemeye göre değerleri belirlendi.		
3. Toplam harcama alınıp verilen şartlar doğrultusunda istenilen Switch-case yapısı oluşturuldu.		
4. Programların çalışıp çalışmadığı kontrol edildi.		
5. Çalışmada iş sağlığı ve güvenliği kurallarına dikkat edildi.		
6. Zaman verimli kullanıldı.		

Case Alternatif Söz Dizimi

Alternatif söz diziminde küme parantezi yerine iki nokta üst üste (:), switch-case blokunun bittiği yere ise endswitch; (Tablo 2.5) eklenerek kontrol deyimi sonlandırılır.

Tablo 2.5: *Switch-case Alternatif Söz Dizimi*

Normal Kullanım	Switch-case Alternatif Söz Dizimi

switch ($değişken) {
case koşul1 :
//Komutlar;
break;
case koşul2 :
//Komutlar;
break;
.....
default:
//Komutlar;
}
	switch ($değişken) :
	case koşul1 :
	//Komutlar;
	break;
	case koşul2 :
	//Komutlar;
	break;
	.....
	default:
	//Komutlar;
	endswtich;

2.3. DÖNGÜ DEYİMLERİ

Program yazarken genellikle karşılaşılan durumlardan biri de programın içinde belirli bir işin tekrarlanmasıdır. Döngüler, program içinde verilen koşul sağlandığı sürece kod bloklarının istenen sayıda tekrar edilerek çalıştırılmasını sağlayan yapılardır.

PHP dilinde 4 adet döngü bulunur. Bunlar;

1. While döngüsü,
2. Do-While döngüsü,
3. For döngüsü,
4. Foreach döngüsüdür.

2.3.1. While Döngüsü

While döngüsü, belirlenen komut satırlarının koşul sağlandığı sürece tekrar etmesini sağlayan döngü yapısıdır. While döngüsü genellikle döngünün kaç kez döneceğinin tahmin edilemediği durumlarda kullanılır. While döngüsünün normal ({ }) ve alternatif yazım yöntemi (:) ile nasıl kullanıldığı Tablo 2.6'da gösterilmektedir.

Tablo 2.6: *While Döngüsünün Kullanım Şekilleri*

Normal Kullanım	While Döngüsünün Kullanım Şekilleri

<?php
while (koşul) {
işlem satırı
…….
…….
işlem satırı
}
?>
	<?php
	while (koşul) :
	işlem satırı
	…….
	…….
	işlem satırı
	endwhile;
	?>

7. Uygulama

1'den 10'a kadar olan sayıları ekrana yazdıran programı aşağıdaki adımları takip ederek yazınız.

1. Adım: index.php dosyasını server içinde oluşturunuz.

2. Adım: Tarayıcıda çalışacak olan index.php sayfasının title kısmını "While Döngüsü" olarak ayarlayınız.

3. Adım: $sayac şeklinde bir değişken tanımlayarak ilk değerini 1 yapınız.

```
DOCTYPE  html>
<html  lang="tr">
<head>
        <meta  charset="UTF-8">
        <title>While  Döngüsü</title>
</head>
<body>
      <?php
      $sayac=1;
      ?>
</body>
</html>
```

4. Adım: While döngüsü için gerekli şartı yazınız.

```
   <?php
      $sayac=1;
      while($sayac<=10){
        //tekrarlanacak  kod  blokları  yazılacak
      }
      ?>
```

5. Adım: While döngüsü içine ekrana 1'den 10'a kadar yazması için gerekli kod satırlarını yazınız.

```
 while($sayac<=10){
    //tekrarlanacak kod blokları yazılacak
    echo $sayac. " " ;    $sayac++;
   //Döngünün sonsuza kadar dönmemesi, şartın istenilen yerde bitmesi için sayaç
bir bir artırılır
   }
```

6. Adım: Aynı işlemleri alternatif yazım yöntemi ile yapmak için tekrarlayınız.

```
     <?php
      $sayac=1;
      while($sayac<=10):
        //tekrarlanacak  kod  blokları  yazılacak
        echo  $sayac.  "  ";
        $sayac++;
      endwhile;
      ?>
```

8. Uygulama

Alfa numerik karakterleri (Türkçe karaktersiz) yan yana aralarına iki boşluk olacak şekilde ekrana yazdıran programı aşağıdaki adımları takip ederek yazınız.

1. Adım: index.php dosyasını server içinde oluşturunuz.

2. Adım: Tarayıcıda çalışacak olan index.php sayfasının title kısmını "Harfler" olarak ayarlayınız.

3. Adım: $sayac=1; $harf="a"; ilk değerlerini alacak iki adet değişken tanımlayınız.

4. Adım: While döngüsü için gerekli koşulu yazınız. Türkçe karaktersiz harfler olacağı için 26 harf üzerinde işlemlerin yapılacağını unutmayınız.

5. Adım: While döngüsü içine ekrana yazmak için gerekli olan kodları ekleyiniz.

6. Adım: While döngüsünün belirtilen şarta kadar dönmesi için $sayac değerini bir bir artırınız.

7. Adım: $harf değerini bir bir artırarak kendisinden sonra gelen harfi ekrana yazdırmasını sağlayınız.

```
DOCTYPE  html><html>
<head>
    <title>While Döngüsü</title>
</head>
<body>
   <?php
         $sayac=1;$harf="a";
        while($sayac<=26):
            echo $harf."  ";
            $sayac++;$harf++;
        endwhile;
   ?>
</body></html>
```

Sıra Sizde

a) 0 ile 100 arasındaki çift sayıları toplayarak ekrana yazdıran programı tasarlayınız.

b) 0-100 (100 dâhil) arasındaki sayılardan 5'e tam bölünenleri yazdıran programı tasarlayınız.

c) Tanımlanan iki sayı değişkeni arasındaki sayıların ortalamasını bulup ekrana yazdıran programı tasarlayınız.

d) Tanımlı olan sayının faktöriyelini bulan programı yazınız.

Değerlendirme

Çalışmanız diğer sayfada yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

KONTROL LİSTESİ

Aşağıda listelenen ölçütlerden öğrencide gözlenen davranış için Evet, gözlenemeyen davranış için Hayır kutucuğunun altına (X) işareti koyarak işaretleyiniz.

Ölçütler	Evet	Hayır
1. Server içindeki whiledongu klasöründe whiledonguornekleri.php dosyası oluşturuldu.		
2. Kullanılacak sayac değişkenleri tanımlandı.		
3. Kullanılacak toplam değişkenleri tanımlandı.		
4. 0 ile 100 arasında işlem yapmak için while döngüsü yazıldı.		
5. Çift sayıları bulup toplamak için while döngüsü içine gerekli if kod bloku yazıldı.		
6. Kullanılan sayac değişkeni sıfırlandı.		
7. 0 ile 100 arasında işlem yapmak için yeni while döngüsü yazıldı.		
8. While döngüsü işlem yaparken sayıların 5'e tam bölünüp bölünmediğini kontrol eden if kod bloku yazıldı.		
9. Kullanılacak iki sayı değişkeni belirlendi.		
10. Kullanılacak iki sayı değerleri belirlendi.		
11. Sayac değişkeni birinci sayıya eşitlendi.		
12. Toplam değişkeni yeniden kullanılmak üzere sıfıra eşitlendi.		
13. Belirlenen aralıkta işlem yapmak için gerekli while kod bloku yazıldı.		
14. İstenilen aralıktaki sayıların toplamı bulundu.		
15. İstenilen aralıktaki sayıların ortalaması bulundu.		
16. Kullanılacak sayı değişkenine yeniden sayı atandı.		
17. Faktöriyel için gerekli while kod bloku yazılıp çarpma işlemi yapıldı.		
18. Programların çalışıp çalışmadığı kontrol edildi.		
19. Çalışmada iş sağlığı ve güvenliği kurallarına dikkat edildi.		
20. Zaman verimli kullanıldı.		

2.3.2. Do-While Döngüsü

Do-while döngüsü temel olarak while döngüsü ile aynıdır. While döngüsünden farklı olarak verilen şart döngünün sonunda kontrol edilir. Bu nedenle döngü içinde yer alan kod bloku en az bir defa çalıştırılır. İlk tekrarın sonunda şart kontrolü yapılır ve şart doğru olduğu sürece döngü çalışmasına devam eder. En az tercih edilen döngüdür. Do-while döngü yapısı Tablo 2.7'de gösterilmektedir.

Tablo 2.7: Do-While Döngüsünün Kullanımı

```
do {
    işlem satırları;
    ……..
} while (şart);
```

9. Uygulama

0 ile 20 arasında rastgele sayı üretip üretilen sayı 5 olmayana kadar bu sayıları toplayan ve ekrana yazdıran programı aşağıdaki adımları takip ederek yazınız.

Önemli

PHP'de rastgele sayı üretmek için rand() fonksiyonu kullanılır. rand(0,20) : 0 ile 20 arasında rastgele sayı üretilmesi için kullanılır.

1. Adım: index.php dosyasını server içinde oluşturunuz.

2. Adım: Tarayıcıda çalışacak olan index.php sayfasının title kısmını "Do-While" olarak ayarlayınız.

3. Adım: $toplam=0; şeklinde bir değişken tanımlayınız.

4. Adım: Do-while döngüsünü şartıyla birlikte ekleyiniz.

```
<!DOCTYPE  html><html>
<head>
                <title>Do-While  Döngüsü</title>
</head>
<body>
<?php
$toplam=0;
do  {
}while  ($sayi!=5);
?>
</body></html>
```

5. Adım: 0 ile 20 arasında rastgele sayı üretip $sayi değişkenine atayınız.

6. Adım: Ürettiğiniz $sayi değişkeninin değerini sayfada aralarında bir boşluk olacak şekilde ekrana yazdırıp aynı anda toplayınız.

7. Adım: Do-while döngüsünün bittiği yere üretilen sayıların toplamını yazdırınız.

```
<?php
$toplam=0;
do  {
$sayi=rand(0,20);
echo  $sayi.“  ”;
$toplam+=$sayi;
}while  ($sayi!=5);
echo  “<br>Toplam=$toplam”;
?>
```

Sıra Sizde

a) 0 ile 100 arasındaki tek sayıları toplayarak ekrana yazdıran programı tasarlayınız.

b) 0-100 (100 dâhil) arasındaki sayılardan 3'e ve 5'e tam bölünenleri yazdıran programı tasarlayınız.

c) Tanımlı olan sayının faktöriyelini bulan programı tasarlayınız.

Değerlendirme

Çalışmanız aşağıda yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

KONTROL LİSTESİ

Aşağıda listelenen ölçütlerden öğrencide gözlenen davranış için Evet, gözlenemeyen davranış için Hayır kutucuğunun altına (X) işareti koyarak işaretleyiniz.

Ölçütler	Evet	Hayır
1. Server içindeki dowhiledongu klasöründe dowhiledonguornekleri.php dosyası oluşturuldu.		
2. Kullanılacak sayac değişkenleri tanımlandı.		
3. Kullanılacak toplam değişkenleri tanımlandı.		
4. 0 ile 100 arasında işlem yapmak için do-while döngüsü yazıldı.		
5. Tek sayıları bulup toplamak için do-while döngüsü içine gerekli if kod bloku yazıldı.		
6. Kullanılan sayac değişkeni sıfırlandı.		
7. 0 ile 100 arasında işlem yapmak için yeni do-while döngüsü yazıldı.		
8. Do-while döngüsü işlem yaparken sayıların 3'e ve 5'e tam bölünüp bölünmediğini kontrol eden if kod bloku yazıldı.		
9. Kullanılacak sayı değişkeni ve değeri belirlendi.		
10. Faktöriyel için gerekli do-while kod bloku yazılıp çarpma işlemi yapıldı.		
11. Programların çalışıp çalışmadığı kontrol edildi.		
12. Çalışmada iş sağlığı ve güvenliği kurallarına dikkat edildi.		
13. Zaman verimli kullanıldı.		

2.3.3. For Döngüsü

Yorumlanması ve yazım şekli basit olduğu için tüm yazılım dillerinde en fazla tercih edilen döngü türüdür. For döngüsü belirlenen başlangıç şartından döngü koşuluna kadar yazılan kod bloklarını çalıştırır. For döngüsünde başlangıç değeri, döngü koşulu ve değişim miktarı (artış azalış değeri) olmak zorundadır. For döngüsünün normal ({ }) ve alternatif yazım yöntemi (:) ile kullanımı Tablo 2.8'de gösterilmektedir.

***Tablo 2.8:** For Döngüsünün Kullanım Şekilleri*

Normal Kullanım	Alternatif Kullanım

`for (başlangıç değeri; döngü koşulu; değişim miktarı) {`
`  işlem satırları;`
`  ……..`
`}`
	`for (başlangıç değeri; döngü koşulu; değişim miktarı) :`
	`  işlem satırları;`
	`  ……..`
	`endfor;`

Tablo 2.8'de gösterilen yapıda başlangıç değeri bir kere ve sadece döngünün başlangıcında çalıştırılır. Döngünün başlayacağı ilk değer atama amacıyla kullanılır. Döngü koşulu doğru olduğu sürece döngü içindeki kod bloklarının çalışmasını sağlar. Değişim miktarı ise her döngü tekrarının sonunda kullanılır ve e döngü değişken değeri artırma / azaltma işlemi için kullanılır.

10. Uygulama

1'den 10' kadar olan sayıları yan yana yazdıran programı aşağıdaki adımları takip ederek yazınız.

1. Adım: index.php dosyasını server içinde oluşturunuz.

2. Adım: Tarayıcıda çalışacak olan index.php sayfasının title kısmını "For Döngüsü" olarak ayarlayınız.

3. Adım: For döngüsünü (başlangıç değeri; döngü koşulu; değişim miktarı) şeklinde yazınız.

4. Adım: Değerleri ekrana yanında boşluk olacak şekilde yazdırınız.

```
<!DOCTYPE  html>
<head>
        <meta  charset="UTF-8">
          <title>For  Döngüsü</title>
</head>
<body>

<?php
for  ($i=1;$i<=10;$i++)
      echo  $i." ";
?>

</body></html>
```

11. Uygulama

Tanımlanan veya girilen bir kelimeyi ekrana tersten yazdıran programı aşağıdaki adımları takip ederek yazınız.

Önemli

PHP'de bir cümle veya kelimenin kaç karakterde olduğunu bulmak için strlen() fonksiyonu kullanılır. Girilen bir metindeki istenilen harfi alabilmek için metin[indis] yapısı kullanılmalıdır. İlk harf 0 indisinde başlar.

1. Adım: index.php dosyasını server içinde oluşturunuz.

2. Adım: Tarayıcıda çalışacak olan index.php sayfasının title kısmını "Kelimeyi tersten yazma" olarak ayarlayınız.

3. Adım: $kelime şeklinde bir değişken tanımlayarak ilk değerini "Gazi MTAL" olarak belirleyebilirsiniz.

4. Adım: $uzunluk şeklinde bir değişken tanımlayarak girilen kelimenin kaç karakter olduğunu bulunuz.

```
<!DOCTYPE  html>
<head>
        <meta  charset="UTF-8">
          <title>Kelimeyi  Tersten  Yazma</title>
</head>
<body>
<?php
$kelime  =  "Gazi  MTAL";
$uzunluk  =  strlen($kelime);
#buraya for döngüsü bloku gelecek
?>
</body></html>
```

5. Adım: for döngüsü için gerekli başlangıç değerini, döngü koşul ve değişim miktarını belirleyiniz ve for döngüsünü yazınız.

```
for  ($i=($uzunluk-1)  ;  $i  >=  0  ;  $i--)
{
#"Gazi MTAL" 9 karakterdir. Girilen kelimeyi harf harf yazdırmak için metin[$i]
şeklinde kullanılması gereklidir fakat indisin 0'dan başladığı unutulmamalıdır.
}
```

6. Adım: for döngüsü içine girilen kelimeyi harf harf yazdırınız.

```
for  ($i=($uzunluk-1)  ;  $i  >=  0  ;  $i--)
{
    echo  $string[$i];  }
```

Sıra Sizde

a) 20'ye kadar olan tek sayıları alt alta yazan PHP kodunu yazınız.

b) 7'den başlayarak 61'e kadar olan sayıları yedişer artırarak yazan PHP kodunu yazınız.

c) Tanımlanan bir sayı değişkeninin faktöriyelini bulan programı yazınız.

d) 1'den 10'a kadar olan sayıların karesini ve küpünü bulup ekrana yan yana yazdıran PHP kodunu yazınız.

Değerlendirme

Çalışmanız aşağıda yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

KONTROL LİSTESİ

Diğer sayfada listelenen ölçütlerden öğrencide gözlenen davranış için Evet, gözlenemeyen davranış için Hayır kutucuğunun altına (X) işareti koyarak işaretleyiniz.

Ölçütler	Evet	Hayır
1. Server içindeki fordongu klasöründe fordonguornekleri.php dosyası oluşturuldu.		
2. 0'dan 20'ye kadar tek sayıları ekrana yazdırmak için for döngüsü kod bloku yazıldı.		
3. 7'den 61'e kadar yedişer yedişer sayıları ekrana yazdırmak için for döngüsü kod bloku yazıldı.		
4. Faktöriyeli bulunacak sayı değişkeni belirlendi.		
5. Faktöriyeli bulunacak sayı değişkenin değeri belirlendi.		
6. Çarpım değişkeni belirlendi.		
7. 1'den başlayıp sayı değişkenine kadar birer birer artan şeklinde gerekli for döngüsü kod bloku yazıldı.		
8. Döngü içinde faktöriyelin bulunması için gerekli çarpım işlemi yapıldı.		
9. 1'den 10'a kadar olan sayıların karesini ve küpünü bulabilmek için gerekli for döngüsü kod bloku yazıldı.		
10. For kod bloku içine 1'den 10'a kadar giderken sayıların karesi bulundu.		
11. For kod bloku içine 1'den 10'a kadar giderken sayıların küpü bulundu.		
12. Programların çalışıp çalışmadığı kontrol edildi.		
13. Çalışmada iş sağlığı ve güvenliği kurallarına dikkat edildi.		
14. Zaman verimli kullanıldı.		

2.3.4. Foreach Döngüsü

Foreach döngüsü diziler içindeki verileri çağırmak için kullanılır. Foreach döngüsü dizi içindeki verileri sırasıyla değişkene atama işlemi yapar ve dizi içindeki elemanlar bitene kadar bu işi tekrarlar. Foreach yapısının iki farklı yazımı Tablo 2.9'da gösterilmektedir.

Tablo 2.9: *Foreach Döngüsünün Kullanım Şekilleri*

1. Kullanım	2. Kullanım

`foreach ($dizi as $deger)`
`{`
`  //  çalışacak  kod blokları;}`
	`foreach ($dizi as $indis => $değer)`
	`{`
	`  //  çalışacak  kod blokları;}`

1. Kullanımda dizide yer alan değerlerle ilgili işlemler yapılır. Döngünün her tekrarında sırası gelen dizi elemanının değeri $deger değişkenine atanır ve kod blokları çalıştırılır.

2. Kullanımda ise hem dizi elemanları hem de dizi elemanlarının indisleri ile ilgili işlemler yapılır. Döngünün her tekrarında sırası gelen dizi elemanının indisi $indis değişkenine, değeri ise $deger değişkenine atanır ve kod blokları çalıştırılır.

12. Uygulama

Ayların tanımlı olduğu bir dizide önce değerleri (ayları) daha sonra da indisleri ile birlikte ekrana yazdıran programı aşağıdaki adımları takip ederek yazınız.

1. Adım: index.php dosyasını server içinde oluşturunuz.

2. Adım: Tarayıcıda çalışacak olan index.php sayfasının title kısmını "Foreach Döngüsü" olarak ayarlayınız.

3. Adım: Aylar için kullanılacak diziyi tanımlayınız.

```
<!DOCTYPE  html>
<head>  <title>Foreach  Döngüsü</title></head>
<body>
<?php
$aylar=array("Ocak","Şubat","Mart","Nisan","Mayıs","Haziran","Temmuz",
"Ağustos","Eylül","Ekim","Kasım","Aralık");
?>
</body></html>
```

4. Adım: $aylar dizisinde bulunan ayları ekrana yazdırmak için gerekli olan foreach döngüsünü yazınız.

```
foreach($aylar  as  $deger){
      echo  $deger,"<br>";
}
```

5. Adım: $aylar dizisinde bulunan ayları indisleriyle birlikte ekrana yazdırmak için gerekli olan foreach döngüsünü yazınız.

Sıra Sizde

a) $dizi_sayilarim=array(14,8,9,4,5,8,9,9,1,6,7,1,7,4,2,2); şeklinde tanımlı olan dizideki sayıların toplamını bulan PHP kodunu yazınız.

b) 1 ile 50 arasında rastgele üretilen sayının tanımlı olan
$sayilarim=aray(42,25,18,29,36,45,27,11,7,4,2,42); dizisinde olup olmadığını bulup ekrana yazdıran PHP kodunu yazınız.

Değerlendirme

Çalışmanız aşağıda yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

KONTROL LİSTESİ

Diğer sayfada listelenen ölçütlerden öğrencide gözlenen davranış için Evet, gözlenemeyen davranış için Hayır kutucuğunun altına (X) işareti koyarak işaretleyiniz.

Ölçütler	Evet	Hayır
1. Server içindeki foreach klasöründe foreachdonguornekleri.php dosyası oluşturuldu.		
2. $dizi_sayilarim dizisi programda yazıldı.		
3. $dizi_sayilarim dizisi üzerinde tek tek dolaşmak için foreach döngüsü kod bloku yazıldı.		
4. Foreach döngüsü içine $dizi_sayilarim dizisi içinde bulunan sayıların toplamı için gerekli kod bloku yazıldı.		
5. 1 ile 50 arasında rastgele üretilen sayı değişkene atandı.		
6. $ sayilarim dizisi programda yazıldı.		
7. $ sayilarim dizisi üzerinde tek tek dolaşmak için foreach döngüsü kod bloku yazıldı.		
8. $sayı değişkeni foreach döngüsü içinde alınan her değer ile karşılaştırıp ekrana yazdıran kod bloku yazıldı.		
9. Programların çalışıp çalışmadığı kontrol edildi.		
10. Çalışmada iş sağlığı ve güvenliği kurallarına dikkat edildi.		
11. Zaman verimli kullanıldı.		

2.3.5. Break ve Continue Deyimleri

Break ve continue deyimleri döngülerle birlikte çok sık kullanılır. Döngü içinde verilen bir koşula göre döngünün çalışmasını sonlandırarak döngüden çıkmak için break deyimi, işlem yapmadan döngünün bir sonraki adıma geçmesi için ise continue deyimi kullanılır.

13. Uygulama

1 ile 100 arasında rastgele üretilen sayının asal sayı olup olmadığını bulan programı aşağıdaki adımları takip ederek yazınız.

1. Adım: index.php dosyasını server içinde oluşturunuz.

2. Adım: Tarayıcıda çalışacak olan index.php sayfasının title kısmını “Break Deyimi” olarak ayarlayınız.

3. Adım: $sayi değişkeni tanımlayarak rastgele bir sayı üretiniz.

4. Adım: $durum değişkeni tanımlayarak ilk değerini True yapınız.

```
DOCTYPE  html>
<head>
        <title>Asal  Sayı  Bulma</title>
</head>
<body>
    <?php
$sayi=rand(1,100);//1 ile 100 arasında rastgele sayı üretme
$durum=True;
    ?>
</body></html>
```

5. Adım: Asal sayı olup olmadığını kontrol için for döngüsünde gerekli koşulu yazınız.

```
for($i=2;$i<$sayi;$i++)
{
}
```

6. Adım: $sayi değişkeninin $i değerine bölümünden kalan kontrolünü yapınız.

```
        if($sayi%$i==0)
         {
                $durum=False;
         }
```

7. Adım: Döngüden sonra $durum değişkeninin değerine göre üretilen sayının asal olup olmadığını ekrana yazdırınız.

```
if($durum)
{
echo "Rastgele üretilen $sayi sayısı asaldır";
}
else
{
        echo "Rastgele üretilen $sayi sayısı asal değildir";
}
```

14. Uygulama

1'den 10'a kadar olan tek sayıları ekrana yazdıran programı aşağıdaki adımları takip ederek yazınız.

1. Adım: index.php dosyasını server içinde oluşturunuz.

2. Adım: Tarayıcıda çalışacak olan index.php sayfasının title kısmını "Continue Deyimi" olarak ayarlayınız.

3. Adım: $sayac değişkeni tanımlayarak ilk değerini sıfır olarak atayınız.

4. Adım: $sayac değişkeninin 10'a kadar dönmesi için gerekli while döngüsünü yazınız.

```
<!DOCTYPE  html>
<head><title>Continue  Deyimi</title></head>
<body>
      <?php
      $sayac=0;
      while($sayac<10)
     {
     }
      ?>
</body></html>
```

5. Adım: Döngü içinde $sayac değişkenini birer birer artırınız.

6. Adım: $sayac değişkeninin 2'ye bölümünden kalanı kontrol ediniz. Kalan sıfır ise bu sayı çift olduğundan ekrana yazdırmadan devam etmek için continue deyimini kullanınız.

7. Adım: $sayac değişkenin değerlerini yan yana yazdırınız.

```
$sayac++;
        if($sayac %2== 0){
          continue;
        }
          echo  $sayac.“  ”;
```

Sıra Sizde

Aşağıdaki kodların nasıl bir çıktı ürettiğini yazınız.

```
<?php
$sayi  =  0;
        while  (++$sayi  <=  10)  {
                if  ($sayi==5)  continue;
                echo  $sayi  .  “  ”;
        }

?>
```

```
<?php
for  ($sayi=1;  $sayi<=10;  $sayi++)  {
                if  ($sayi==7)  continue;
                echo  $sayi  .  “  ”;
?>
```

```
<?php
              for  ($sayi=1;  $sayi<=10;  $sayi++)  {
                    echo  $sayi  .  “  ”;
                    if  ($sayi==4)  break;
        }
?>
```

```
<?
        $sayi  =  1;
        do  {
                echo  $sayi++  .  “  ”;
                if  ($sayi==7)  break;
        }  while  ($sayi  <=  10);
?>
```

2.4. VERİLER ÜZERİNDE DÖNGÜ KONTROLLERİ

Web programcılığında tasarım yapılırken genellikle karşılaşılan durumlardan biri de program içinde HTML etiketlerinin isteğe göre tekrarlanmasıdır. Bu kısımda HTML etiketlerinin döngüler ile nasıl kullanıldığına yönelik örnekler verilecektir.

15. Uygulama

Ekranda alt alta 7 defa "PHP Öğreniyorum" cümlesini Görsel 2.5'teki gibi küçükten büyüğe doğru yazdıran PHP programını aşağıdaki adımları takip ederek yazınız.



Görsel 2.5: Font size kullanımı

1. Adım: index.php dosyasını server içinde oluşturunuz.

2. Adım: Tarayıcıda çalışacak olan index.php sayfasının title kısmını "Veriler üzerinde döngü kontrolleri" olarak ayarlayınız.

3. Adım: 1'den 7'ye kadar devam edecek for döngüsünün kod blokunu yazınız.

```
<!DOCTYPE  html>
<html  lang=”tr”>
<head><title>Veriler  üzerinde  döngü  kontrolleri</title>
</head><body>
        <?php
            for  ($i=1;  $i<=7;  $i++)
                {                }
        ?>
</body></html>
```

Önemli

HTML font büyüklüğü ile belirlenir.

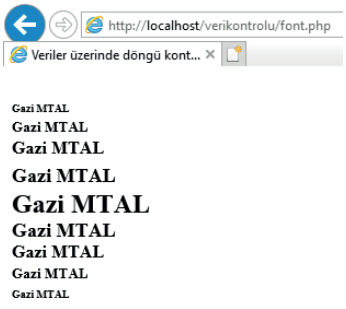
4. Adım: <font size>'ın 1'den 7'ye kadar tek tek değer alması ve ekrana "PHP Öğreniyorum" ifadesini yazması için gerekli kod blokunu yazınız.

```
echo  “<font  size=$i>”;
echo  “<br><b>PHP  Öğreniyorum</b></font>”;
```

5. Adım: Kod blokunu localhost üzerinde çalıştırınız.

Sıra Sizde

Görsel 2.6'daki ekran çıktısını verecek olan PHP kodunu yazınız.



Görsel 2.6: Alt alta yazdırma

Değerlendirme

Çalışmanız aşağıda yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

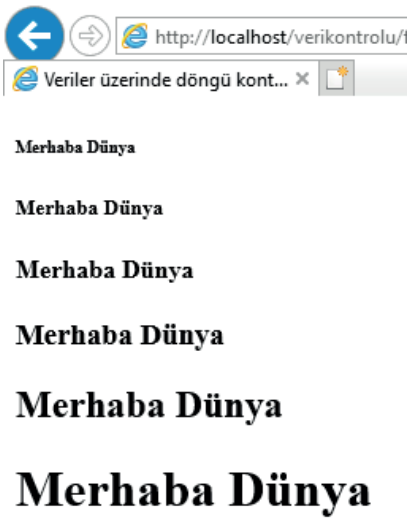
KONTROL LİSTESİ

Aşağıda listelenen ölçütlerden öğrencide gözlenen davranış için Evet, gözlenemeyen davranış için Hayır kutucuğunun altına (X) işareti koyarak işaretleyiniz.

Ölçütler	Evet	Hayır
1. Server içindeki verikontrolu klasöründe index.php dosyası oluşturuldu.		
2. Birinci for döngüsü 1'den 5'e kadar kod bloku ile yazıldı.		
3. Birinci for döngüsü içinde Gazi MTAL cümlesinin küçükten büyüğe doğru yazılması için gerekli kod bloku yazıldı.		
4. İkinci for döngüsü 4'ten 1'e doğru bir bir azalacak şekilde kod bloku ile yazıldı.		
5. İkinci for döngüsü içinde Gazi MTAL cümlesinin büyükten küçüğe doğru yazılması için gerekli kod bloku yazıldı.		
6. Programın çalışıp çalışmadığı kontrol edildi.		
7. Çalışmada iş sağlığı ve güvenliği kurallarına dikkat edildi.		
8. Zaman verimli kullanıldı.		

16. Uygulama

Heading (Başlık) etiketlerini kullanarak for ve while döngüsü ile küçükten büyüğe "Merhaba Dünya" yazdıran PHP kodunu aşağıdaki adımları takip ederek yazınız (Görsel 2.7).



Görsel 2.7: Heading etiketinin kullanımı

Önemli

Heading etiketinin kod tarafında yazım şekli h'dir. Heading etiketi, Google botlarının sayfa içeriği hakkında bilgi alabilmesi için başlıklara yerleştirilen HTML etiketleridir. Heading etiketleri 1'den başlayarak (h1) 6'ya kadar (h6) gidebilir.

1. Adım: index.php dosyasını server içinde oluşturunuz.

2. Adım: Tarayıcıda çalışacak olan index.php sayfasının title kısmını "Veriler üzerinde döngü kontrolleri" olarak ayarlayınız.

3. Adım: 6'dan 1'e kadar azalacak olan for döngüsü kod blokunu yazınız.

```
<!DOCTYPE  html>
```

```
<head><title>Veriler  üzerinde  döngü  kontrolleri</title>
</head><body>
        <?php
          for  ($i=6;$i>0;$i--)
{
}
        ?>
</body></html>
```

4. Adım: For döngüsü içine h etiketini küçükten büyüğe doğru yazdırmak için gerekli kod blokunu yazınız.

```
                echo  "<h".$i. ">". "Merhaba  Dünya". "</h".$i. ">";
```

5. Adım: 6'dan 1'e kadar azalacak olan while döngüsü kod blokunu yazınız.

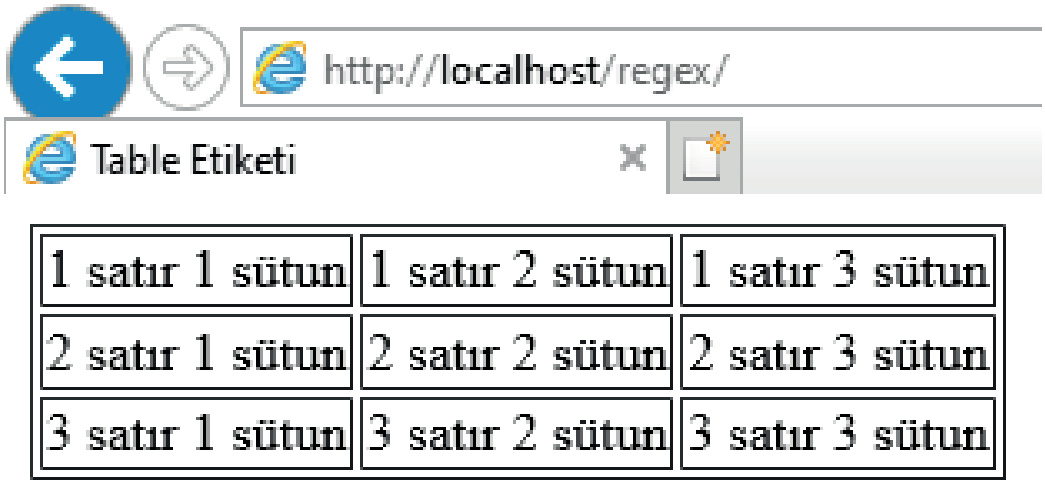
```
   $i=6;
        while($i>0){
                $i--;
        }
```

6. Adım: While döngüsü içinde h etiketini küçükten büyüğe doğru yazdırmak için gerekli kod blokunu yazınız.

```
                echo  "<h".$i. ">". "Merhaba  Dünya". "</h".$i. ">";
```

7. Adım: Kod blokunu localhost üzerinde çalıştırınız.

17. Uygulama



Görsel 2.8: Table etiketinin kullanımı

Table (Tablo) etiketlerini kullanarak Görsel 2.8'deki ekran çıktısını verecek PHP kodlarını aşağıdaki

adımları takip ederek yazınız.

Önemli

<table> </table> Web sayfasına tablo eklemek için kullanılan etikettir. Tablolar profesyonel web sayfalarının ayrılmaz ögelerindendir. <tr></tr> etiketi tablo içinde satır, <td></td> etiketinde ise açılan satıra sütunlar eklenir.

1. Adım: index.php dosyasını server içinde oluşturunuz.

2. Adım: Tarayıcıda çalışacak olan index.php sayfasının title kısmını *"Table Etiketi"* olarak ayarlayınız.

3. Adım: echo komutu ile öncelikle tablo etiketini oluşturunuz.

```
<!DOCTYPE  html>
<head><title>Table  Etiketi</title>
</head><body>
        <?php
        echo  "<table  border  =  '1px'  >";//table  açılıyor
    //arasına for döngü kodları yazılacak
        echo  "</table>";//table  kapatılıyor
        ?>
      </body></html>
```

4. Adım: Görsel 2.12'de verilen tablonun satırlarını oluşturmak için gerekli for döngüsü kod blokunu yazınız.

```
       for($satir=1;  $satir  <4;  $satir++)
        {
              echo  "<tr>";//satır  açılıyor
       //arasına sütunları eklemek için gerekli for döngü kodları yazılacak
              echo  "</tr>";//satır  kapatılıyor
        }
```

5. Adım: Görsel 2.8'de verilen tablonun satırlarının içindeki sütunları oluşturmak için for döngüsü kod blokunu yazınız.

```
          for($sutun  =  1;  $sutun<4;  $sutun++)
          {
                      echo  " <td>$satir  satır  $sütun  sütun</td>  ";
//hücreler  oluşturup  ekrana  1  satır  2  sütun  gibi  değişen  değerler  yazar
          }
```

6. Adım: Kod blokunu localhost üzerinde çalıştırınız.

18. Uygulama

1'den 5'e kadar olan çarpım tablosunu ekrana yazdıran programı aşağıdaki adımları takip ederek yazınız.

Önemli

HTML taglarında tablo yapısı kullanılacaktır. <table><tr><td></td></tr></table>

1. Adım: index.php dosyasını server içinde oluşturunuz.

2. Adım: Tarayıcıda çalışacak olan index.php sayfasının title kısmını "Çarpım tablosu" olarak ayarlayınız.

3. Adım: Tablo oluşturup satırı açınız.

```
<!DOCTYPE  html>
<head>
        <title>Çarpım  Tablosu</title>
</head>
<body>
<?php
echo  "<table  border=1><tr>";
//Döngü bloku yazılacak
?>
```

4. Adım: 1'den 5'e kadar olacak birinci döngüyü yazınız.

```
for  ($i=1;$i<=5;$i++)// Çarpım tablosu için
{
//Diğer kodların yazılacağı alan
}
```

5. Adım: Her çarpım tablosu yazılacak ifade için bir sütun açınız.

6. Adım: Her değerin 1'den 10'a kadar çarpım tablosunun yazılması için gerekli ikinci for döngüsünü yazınız.

```
for  ($i=1;$i<=5;$i++)// Çarpım tablosu için
{
echo  "<td>";//Her değer için bir sütun oluşturunuz 1'ler 2'ler …. için
        for($a=1;$a<=10;$a++)//Bu  döngü  1'den 10'a kadar çarpma yapmak için
                {
           // Ekrana yazdırma işleminin yapılacağı yer
                }
}
```

7. Adım: 1*1=1 şeklinde olacak şekilde çarpım tablosunu ekrana yazdırıp sütunu kapatınız.

```
for  ($i=1;$i<=5;$i++)// 1ler  2ler…  için
{
echo  “<td>”;//Her değer için bir sütun oluşturunuz 1’ler 2’ler …. için
        for($a=1;$a<=10;$a++)//Bu  döngü  1’den 10’a kadar çarpma yapmak için
                {
        echo  “$i*$a=”.$i*$a;//  ekranda  1*1=1    görüntüsü  için
                echo  “<br>”;
                }
echo  “</td>”;//  sütunu  kapatılıyor
}
```

8. Adım: İlk başta yazılan satır ve tablo etiketini kapatınız.

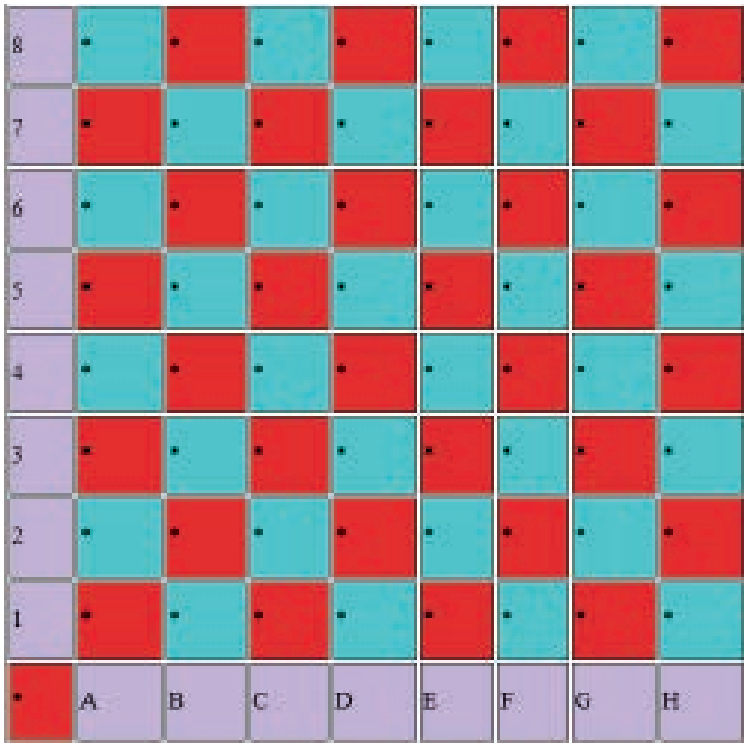
```
echo  “</tr></table>”;
```

9. Adım: Kod blokunu localhost üzerinde çalıştırınız.

Sıra Sizde

a) 7 satırlı 5 sütunlu bir tablo oluşturarak her hücresine “*” karakteri yazdıran PHP kodunu yazınız.

b) Görsel 2.9’da verilen ekran görüntüsündeki satranç tahtasını 500px*500px boyutunda oluşturan PHP kodunu yazınız.



Görsel 2.9: Satranç tahtası

Değerlendirme

Çalışmanız diğer sayfada yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

KONTROL LİSTESİ

Aşağıda listelenen ölçütlerden öğrencide gözlenen davranış için Evet, gözlenemeyen davranış için Hayır kutucuğunun altına (X) işareti koyarak işaretleyiniz.

Ölçütler	Evet	Hayır
1. Server içindeki verikontrolu klasöründe index.php dosyası oluşturuldu.		
2. <table></table> etiketi echo komutu ile birlikte yazıldı.		
3. Tablonun içine 7 satır eklemek için for döngüsü kod bloku yazıldı.		
4. Satırlar içine 5 sütun eklenmesi için gerekli for döngüsü kod bloku yazıldı.		
5. Hücrelere “*” işareti yazan kod satırları yazıldı.		
6. Satranç tahtası için gerekli tablo oluşturuldu.		
7. 9*9 satır ve sütun için gerekli iç içe for kod bloku yazıldı.		
8. Rakamlar ve harfler sütunlara yazıldı.		
9. Hücrelere “*” işareti yazan kod satırları yazıldı.		
10. Hücrelerin arka plan renk ayarları yapıldı.		
11. Programların çalışıp çalışmadığı kontrol edildi.		
12. Çalışmada iş sağlığı ve güvenliği kurallarına dikkat edildi.		
13. Zaman verimli kullanıldı.		

19. Uygulama

Masaüstünde bulunan resimler klasöründeki resimleri ekrana yazdıran programı aşağıdaki adımları takip ederek yazınız.

Önemli

Sayfaya resim yerleştirmek için <img> etiketi kullanılır. src = “resim_dosyası” kullanılacak resim yolunu göstermek için kullanılır.

1. Adım: index.php dosyasını server içinde oluşturunuz.

2. Adım: Tarayıcıda çalışacak olan index.php sayfasının title kısmını “Resimleri Gösterme” olarak ayarlayınız.

3. Adım: Resim klasörünün içinde 5 adet resim olduğunu düşünerek bu resimleri sırayla göstermek için gerekli for kod blokunu yazınız.

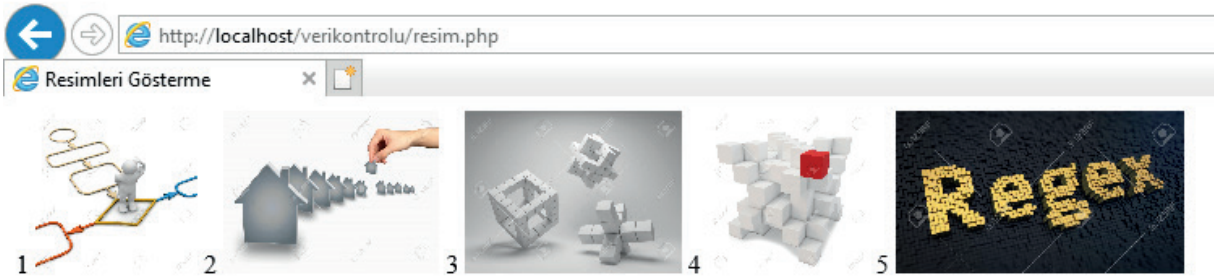
```
<!DOCTYPE  html><head>
          <title>Resimleri  Gösterme</title>
</head><body>
<?php

        for($sayi  =  1;  $sayi  <=  5;  $sayi++)
                {
                                        //<img> etiketi için gerekli satır yazılacak
                }
?>
</body></html>
```

4. Adım: For bloku içinde resimleri göstermek ve ekrana bastırmak için gerekli kodları yazınız.

```
echo  $sayi;
echo  ‘
<img src=“C:/Users/OKUL/Desktop/resimler/‘.$sayi.’.jpg” witdh=100 height=100>
’;
echo“          ”;
```

5. Adım: Kod blokunu localhost üzerinde çalıştırınız (Görsel 2.10).



Görsel 2.10: Resimleri Gösterme

Sıra Sizde

Ailenizdeki kişilerin resimlerini ve adlarını ekranda gösteren PHP kodunu yazınız.

Değerlendirme

Çalışmanız diğer sayfada yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

KONTROL LİSTESİ

Aşağıda listelenen ölçütlerden öğrencide gözlenen davranış için Evet, gözlenemeyen davranış için Hayır kutucuğunun altına (X) işareti koyarak işaretleyiniz.

Ölçütler	Evet	Hayır
1. Server içindeki verikontrolu klasöründe resim.php dosyası oluşturuldu.		
2. Masaüstünde veya server içinde resim klasörü oluşturuldu.		
3. Aile bireylerinin resimleri klasörün içine eklendi.		
4. Klasörün içindeki resimlerin alınması için gerekli for kod bloku oluşturuldu.		
5. For kod blokundan resimleri çekmek için <img> etiketi eklenerek yükseklik ve genişlik ayarı yapıldı.		
6. Programların çalışıp çalışmadığı kontrol edildi.		
7. Çalışmada iş sağlığı ve güvenliği kurallarına dikkat edildi.		
8. Zaman verimli kullanıldı.		

2.5. DÜZENLİ İFADELER

Düzenli ifadeler (Regular Expression-Regex), birçok programlama dilinde kullanılabilen sayısal ve dizgisel içeriklerde belirli kurallara uyan bölümleri bulmak için yazılan ifadelerdir. Düzenli ifadeler genellikle bir metin içinde arama yapmak veya istenilen metni değiştirmek amacıyla kullanılır (Görsel 2.11).



Görsel 2.11: Düzenli ifadeler

Program içinde düzenli ifadeleri kullanabilmek için Tablo 2.10'da verilen PHP'de bulunan hazır fonksiyonların kullanılması gerekir.

Tablo 2.10: *Regex Fonksiyonları*

Fonksiyon Adı	Açıklaması
preg_match	Bu fonksiyon, düzenli ifadenin içerik içinde olup olmadığını kontrol eder. Eğer ifade metinde mevcutsa 1, değilse 0 sonucunu döndürür.
preg_match_all	Bu fonksiyon, düzenli ifadenin metinde kaç tane olduğunu geri döndürür.
preg_replace	Bu fonksiyon, düzenli ifade aranan metin varsa onu başka bir metinle değiştirmek için kullanılır.
preg_split	Bu fonksiyon, metin içinde istenilen kısımları alarak diziye dönüştürmek için kullanılır.
preg_grep	Bu fonksiyon, düzenli ifadeyle eşleşenleri içeren bir dizi döndürür.
preg_quote	Bu fonksiyon, düzenli ifadeyle sorgulanacak metin içinde hataya sebep olacak özel karakterleri engellemek için kullanılır.

Metin içinde aranan sabit bir ifadenin olup olmadığını sorgularken kullanılacak temel eşleştirme desenlerinin (düzenli ifadeler) örnek yazımları Tablo 2.11'de gösterilmiştir.

Tablo 2.11: *Arama İfadelerinin Yazım Şekilleri*

Yazım	Açıklaması
ornek	Belirlenen metin içinde ornek kelimesinin olup olmadığını kontrol eder. Eğer ifade metinde mevcutsa 1, değilse 0 sonucunu döndürür.
^ornek	Belirlen metin içinde ornek kelimesi ile başlayan cümle olup olmadığını kontrol eder. Eğer ifade metinde mevcutsa 1, değilse 0 sonucunu döndürür.
ornek$	Belirlen metin içinde ornek kelimesi ile biten cümle olup olmadığını kontrol eder. Eğer ifade metinde mevcutsa 1, değilse 0 sonucunu döndürür.
^ornek$	Belirlenen metin içinde ornek kelimesinin olup olmadığını kontrol eder. Eğer ifade metinde mevcutsa 1, değilse 0 sonucunu döndürür.

20. Uygulama

Basit eşleştirme desenleriyle arama yapma işlemini aşağıdaki adımları takip ederek gerçekleştiriniz.

1. Adım: index.php dosyasını server içinde oluşturunuz.

2. Adım: Tarayıcıda çalışacak olan index.php sayfasının title kısmını "Basit Eşleştirme Desenleri" olarak ayarlayınız.

3. Adım: <?php ?> içine aşağıdaki kodları yazınız.

4. Adım: PHP sayfasını kaydediniz ve serverda bu sayfayı çalıştırarak görüntüleyiniz.

```
  $desenim="#gazi#";
  $desenim1="#^gazi#";//#  karakterleri  sınırlayıcı  olarak  kullanılması
gerekmektedir.
$icerik=" Düzenli ifadeler (Regular Expression-Regex), birçok programlama
dilinde kullanılabilen sayısal ve dizgisel içeriklerde belirli kurallara
uyan bölümleri bulmak için yazılan ifadelerdir.";
```

```
preg_match($desenim,$icerik,$bulunan);
preg_match($desenim1,$icerik,$bulunan1);
print_r($bulunan);
echo“<br>”;
print_r($bulunan1);
```

Önemli

Print_r() komutu bir değişken hakkında okunabilir bilgileri yazdırmak için kullanılır.

Sıra Sizde

$metin=“Düzenli ifadeler genellikle bir metin içinde arama yapmak veya istenilen metni değiştirmek amacıyla kullanılır.”; şeklinde tanımlı olan metin içinde **veya** kelimesinin geçip geçmediğini bulan PHP kodunu yazınız.

Değerlendirme

Çalışmanız aşağıda yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

KONTROL LİSTESİ

Aşağıda listelenen ölçütlerden öğrencide gözlenen davranış için Evet, gözlenemeyen davranış için Hayır kutucuğunun altına (X) işareti koyarak işaretleyiniz.

Ölçütler	Evet	Hayır
1. Server içindeki duzenliifade klasöründe duzenliifade.php dosyası oluşturuldu.		
2. $metin değişkeni programda yazıldı.		
3. $metin değeri programda yazıldı.		
4. $metin içinde veya kelimesinin aramasının yapılması için gerekli düzenli ifade ($desen) hazırlandı.		
5. preg_match fonksiyonu ile $desen ifadesinin $metin içinde olup olmadığı kontrol edildi.		
6. print_r ile ekrana yazdırıldı.		
7. Programların çalışıp çalışmadığı kontrol edildi.		
8. Çalışmada iş sağlığı ve güvenliği kurallarına dikkat edildi.		
9. Zaman verimli kullanıldı.		

Arama için hazırlanan düzenli ifadeleri detaylandırırken köşeli parantezler arasında tanımlama yapılmalıdır. Tablo 2.12’de detaylandırma işlemlerinin hangi karakterler ile yapılacağı açıklanmıştır.

Tablo 2.12: Karakter Sınıfları

RegExp İfadesi	Açıklaması
[abc]	a, b veya c karakterleri bulunur.
[a-z]	Metin içinde küçük harfler bulunur.
[A-Z]	Metin içinde büyük harfler bulunur.
[0-9]	Metin içinde rakamlar bulunur.
[a-z0-9]	Metin içinde küçük harfler ve rakamlar bulunur.
[a-z\-\.]	Metin içinde küçük harfler ile kısa çizgi ve nokta işaretleri bulunur.
[a-Z]	Küçük harf ile büyük harflerin hepsi bulunur.

Önemli

[A-Z] veya [a-z] şeklinde yazıldığında Türkçe karakterleri dâhil etmez. Bunun için [a-z] ifadesinin içine[a-zşğçıüö] şeklinde tanımlama yapılmalıdır. Aynı zamanda tanımlama aralığı programa göre değiştirilebilir. Özel karakterler tanımlanırken mutlaka kaçış operatörü (\) ile kullanılmalıdır. Arama işleminde birden fazla sonuç alınacaksa **preg_match_all()** fonksiyonu kullanılır.

21. Uygulama

Karakter sınıflarına göre arama yapma işlemini aşağıdaki adımları takip ederek gerçekleştiriniz.

1. Adım: index.php dosyasını server içinde oluşturunuz.

2. Adım: Tarayıcıda çalışacak olan index.php sayfasının title kısmını "Karakter Sınıfları" olarak ayarlayınız.

3. Adım: <?php ?> içine aşağıdaki kodları yazınız.

4. Adım: PHP sayfasını kaydediniz ve serverda bu sayfayı çalıştırarak görüntüleyiniz.

```
<!DOCTYPE  html>
<html  lang="tr">
<head>
          <meta  charset="UTF-8">
        <title>Karakter  Sınıfları</title>
</head>
<body>
        <?php
        $desenim="#[a-z  1-5  \.  ]#";
        $desenim1="#[A-Z  1-5  \.  ]#";
        $metin="10 Kasım Atatürk'ü anma günü ve Atatürk haftası.";
        preg_match_all($desenim,$metin,$bulunan);
        preg_match_all($desenim1,$metin,$bulunan1);
        echo"<pre>";
                print_r($bulunan);
                echo"<pre>";
                print_r($bulunan1);
        ?>
</body>  </html>
```

Sıra Sizde

$metin=" ***Çanakkale Zaferi, Türk askerinin ruh kudretini gösteren şayanı hayret ve tebrik bir misaldir. Emin olmalısınız ki, Çanakkale Muharebelerini kazandıran bu yüksek ruhtur." (Mustafa Kemal Atatürk)***"; şeklinde tanımlı olan metin içinde büyük harfler ile virgül(,) bulan PHP kodunu yazınız.

Değerlendirme

Çalışmanız aşağıda yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

KONTROL LİSTESİ

Aşağıda listelenen ölçütlerden öğrencide gözlenen davranış için Evet, gözlenemeyen davranış için Hayır kutucuğunun altına (X) işareti koyarak işaretleyiniz.

Ölçütler	Evet	Hayır
1. Server içindeki duzenliifade klasöründe duzenliifade1.php dosyası oluşturuldu.		
2. $metin değişkeni programda yazıldı.		
3. $metin değeri programda yazıldı.		
4. $metin içinde büyük harfler ile virgül (,) bulan gerekli düzenli ifade ($desen) hazırlandı.		
5. preg_match_all fonksiyonu ile $desen ifadesinin $metin içinde olup olmadığı kontrol edildi.		
6. print_r ile ekrana yazdırıldı.		
7. Programların çalışıp çalışmadığı kontrol edildi.		
8. Çalışmada iş sağlığı ve güvenliği kurallarına dikkat edildi.		
9. Zaman verimli kullanıldı.		

Girilen veya tanımlanan bir ifadenin istenilen formatta yazılıp yazılmadığını kontrol etmek amacıyla tekrarlayıcılar kullanılabilir. Örneğin girilen tarihin doğru formatta yazılıp yazılmadığını kontrol etme vb. için Tablo 2.13'te tekrarlayıcıların nasıl kullanılacağı açıklanmıştır.

Tablo 2.13: *Tekrarlayıcılar*

Regex İfadesi	Açıklaması
İfade {12}	İfadenin küme parantezi içinde yazılan sayı kadar tekrar etmesini sağlar.
İfade {1,5}	İfadenin küme parantezi içinde belirtilen sayı aralığında tekrar etmesini sağlar.
ifade ?	İfade değerinin önemli olmadığını ifade eder.
ifade +	İfade değerinin bir veya daha fazla tekrar etmesini sağlar.
ifade *	İfade değerinin sıfır veya daha fazla tekrar etmesini sağlar.

22. Uygulama

Tanımlanan veya girilen tarih değerinin istenilen formatta girilip girilmediğini kontrol eden programı aşağıdaki adımları takip ederek yazınız.

1. Adım: index.php dosyasını server içinde oluşturunuz.

2. Adım: Tarayıcıda çalışacak olan index.php sayfasının title kısmını "Tekrarlayıcılar" olarak ayarlayınız.

3. Adım: Girilmesi istenilen tarih formatı için gerekli düzenli ifadeyi yazınız.

```
<!DOCTYPE  html>
<html  lang=”tr”>
<head>
          <meta  charset=“UTF-8”>
        <title>Tekrarlayıcılar</title>
</head>
<body>
        <?php
        $format=“#[0-9]{2}\.[0-9]{2}\.[0-9]{4}#”;
        ?>
</body>
</html>
```

4. Adım: $girilen tarih şeklinde bir tarih değişkeni tanımlayıp değerini 03.04.1983 yapınız.

5. Adım: Oluşturulan tarih formatının girilen metin ile eşleşip eşleşmediğini kontrol ediniz ve metni ekrana yazdırınız.

```
     preg_match($format,$girilen_tarih,$bulunan);
      echo“<pre>”;
        print_r($bulunan);
```

6. Adım: Arama sonucunu ekran yerine doğrulama işlemi için kullanabilirsiniz. Bunun için $bulunan değişkeni kaldırıp aşağıdaki kod satırını yeniden yazarak serverda çalıştırınız.

```
if(preg_match($format,$girilen_tarih)):
              echo“Girilen  tarih  formatı  doğrudur”;
       else:
                echo“Girilen  tarih  formatı  yanlıştır”;
       endif;
```

23. Uygulama

Tanımlanan veya girilen web adresinin doğru formatta girilip girilmediğini kontrol eden programı aşağıdaki adımları takip ederek yazınız.

1. Adım: index.php dosyasını server içinde oluşturunuz.

2. Adım: Tarayıcıda çalışacak olan index.php sayfasının title kısmını "Gruplandırıcılar" olarak ayarlayınız.

3. Adım: Girilmesi istenilen adres formatı için gerekli düzenli ifadeyi yazınız.

```
<!DOCTYPE  html>
<html  lang="tr">
<head>
          <meta  charset="UTF-8">
        <title>Gruplandırma</title>
</head><body>
        <?php
        $format="#([a-z\.\/\:]+)#";
        ?>
</body>
</html>
```

4. Adım: $girilen_adres şeklinde bir tarih değişkeni tanımlayıp değerini https://meb.gov.tr/ yapınız.

5. Adım: Oluşturulan tarih formatının girilen metin ile eşleşip eşleşmediğini kontrol ediniz ve metni ekrana yazdırınız.

```
       $girilen_adres="https://meb.gov.tr/";
          if(preg_match($format,$girilen_adres)):
              echo"Girilen  adres  formatı  doğrudur";
        else:
                echo"Girilen  adres  formatı  yanlıştır";
        endif;
```

6. Adım: Arama sonucunu ekran yerine doğrulama işlemi için kullanabilirsiniz. Bunun için $bulunan değişkeni kaldırıp aşağıdaki kod satırını yeniden yazarak serverda çalıştırınız.

```
if(preg_match($format,$girilen_tarih)):
       echo "Girilen tarih formatı doğrudur";
    else:
        echo "Girilen tarih formatı yanlıştır";
    endif;
```

24. Uygulama

Tanımlanan veya girilen web adresinin doğru formatta girilip girilmediğini kontrol eden programı aşağıdaki adımları takip ederek yazınız.

1. Adım: index.php dosyasını server içinde oluşturunuz.

2. Adım: Tarayıcıda çalışacak olan index.php sayfasının title kısmını "Gruplandırıcılar" olarak ayarlayınız.

3. Adım: Girilmesi istenilen adres formatı için gerekli düzenli ifadeyi yazınız.

```
<!DOCTYPE html>
<html lang="tr">
<head>
     <meta charset="UTF-8">
    <title>Gruplandırma</title>
</head><body>
    <?php
    $format="#([a-z\.\/\:]+)#";
    ?>
</body>
</html>
```

4. Adım: $girilen_adres şeklinde bir tarih değişkeni tanımlayıp değerini https://meb.gov.tr/ yapınız.

5. Adım: Oluşturulan tarih formatının girilen metin ile eşleşip eşleşmediğini kontrol ediniz ve metni ekrana yazdırınız.

```
   $girilen_adres="https://meb.gov.tr/";
     if(preg_match($format,$girilen_adres)):
         echo"Girilen adres formatı doğrudur";
    else:
        echo"Girilen adres formatı yanlıştır";
    endif;
```

Sıra Sizde

Tanımlanan bir mailin doğru formatta yazılıp yazılmadığını kontrol eden PHP programını yazınız.

Değerlendirme

Çalışmanız diğer sayfada yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

KONTROL LİSTESİ

Aşağıda listelenen ölçütlerden öğrencide gözlenen davranış için Evet, gözlenemeyen davranış için Hayır kutucuğunun altına (X) işareti koyarak işaretleyiniz.

Ölçütler	Evet	Hayır
1. Server içindeki duzenliifade klasöründe duzenliifade2.php dosyası oluşturuldu.		
2. $girilen_mail değişkeni programda yazıldı.		
3. $girilen_mail değeri programda yazıldı.		
4. $girilen_mail değişkeninin doğru formatta yazılıp yazılmadığını kontrol eden gerekli düzenli ifade ($mail_kontrol) hazırlandı.		
5. $girilen_mail , $mail_kontrol değişkenleri, if deyimi ve preg_match fonksiyonuyla kontrol edildi.		
6. Doğru olup olmadığı ekrana yazdırıldı.		
7. Programların çalışıp çalışmadığı kontrol edildi.		
8. Çalışmada iş sağlığı ve güvenliği kurallarına dikkat edildi.		
9. Zaman verimli kullanıldı.		

Regex'te iki tane operatör kullanılır. Bunlar veya (|) ve değildir ([^..]) operatörleridir (Tablo 2.14).

***Tablo 2.14:** Regex Operatörleri*

Operatör Adı	Sembolü	Örnek	Açıklaması
Veya	|	#deneme(lar|ler) #	Deneme kelimesinin yanına gelen -lar ve -ler eklerine göre iki farklı arama yapılır.
Değil	[^..]	#[^0-5]*#	Metin içinde 0 ile 5 arasında rakam geçmeyecek manasında kullanılır.

Hazırlanan düzenli ifadenin (desenin) sonuna eklenecek karakter ile desenin durumu değiştirilebilir (Tablo 2.15).

***Tablo 2.15:** Desen Düzenleyiciler*

RegExp İfadesi	Açıklaması
i	Düzenli ifadenin büyük / küçük harf ayrımı yapmasını engeller.
m	Uzun metinli içeriklerin paragraf başını ve sonunu belirler.
s	Uzun metinli içeriğin tek satır şeklinde algılanmasını sağlar.
u	UTF-8 kodlaması kullanılır (Türkçe karakterler için mutlaka kullanılması gereklidir.).

Önemli

Sınırlar belirlenirken # işareti yerine / işareti de kullanılabilir.

25. Uygulama

Tanımlanan kelimeyi metin içinde büyük / küçük harf ayrımı yapmadan bulan programı aşağıdaki adımları takip ederek yazınız.

1. Adım: index.php dosyasını server içinde oluşturunuz.

2. Adım: Tarayıcıda çalışacak olan index.php sayfasının title kısmını "Düzenleyiciler" olarak ayarlayınız.

3. Adım: $kelime değişkeni büyük / küçük harf ayrımı yapmadan De hecesini bulmak için gerekli düzenli ifadeyi yazınız.

```
<!DOCTYPE  html>
<head>
        <title>Basit  Eşleştirme  Desenleri</title>
</head><body>
      <?php
      $kelime='/De/i';
      ?>
</body>
</html>
```

4. Adım: $icerik=" Düzenli ifadeler (Regular Expression-Regex), birçok programlama dilinde kullanılabilen sayısal ve dizgisel içeriklerde belirli kurallara uyan bölümleri bulmak için yazılan ifadelerdir. Düzenli ifadeler genellikle bir metin içinde arama yapmak veya istenilen metni değiştirmek amacıyla kullanılır."; değişkenini tanımlayınız.

5. Adım: $kelime değişkeninin değerinin $içerik içinde olup olmadığını kontrol eden kodları yazınız.

```
   $icerik= "Düzenli ifadeler (Regular Expression-Regex), birçok programlama
dilinde kullanılabilen sayısal ve dizgisel içeriklerde belirli kurallara
uyan bölümleri bulmak için yazılan ifadelerdir. Düzenli ifadeler genellikle
bir metin içinde arama yapmak veya istenilen metni değiştirmek amacıyla
kullanılır.";
  if(preg_match_all($kelime,$icerik)) {
      echo "İçerik var";
  }
  else{
    echo "İçerik yok";
  }
```

Sıra Sizde

İstenilen sayıyı yakalayan PHP programını yazınız.

Değerlendirme

Çalışmanız aşağıda yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

KONTROL LİSTESİ

Aşağıda listelenen ölçütlerden öğrencide gözlenen davranış için Evet, gözlenemeyen davranış için Hayır kutucuğunun altına (X) işareti koyarak işaretleyiniz.

Ölçütler	Evet	Hayır
1. Server içindeki duzenliifade klasöründe duzenliifade3.php dosyası oluşturuldu.		
2. $metin değişkeninin içinde rakam ve kelimelerden oluşan bir içerik yazıldı.		
3. $metin içinde geçen rakamları bulmak için $rakam düzenli ifadesi hazırlandı.		
4. if deyimi ve preg_match fonksiyonu kontrol edildi.		
5. $metin içindeki rakamlar ekrana yazdırıldı.		
6. Programların çalışıp çalışmadığı kontrol edildi.		
7. Çalışmada iş sağlığı ve güvenliği kurallarına dikkat edildi.		
8. Zaman verimli kullanıldı.		

ÖLÇME VE DEĞERLENDİRME

A) Aşağıdaki cümlelerde parantezlerin içine yargılar doğru ise "D", yanlış ise "Y" yazınız.

1. ( ) ,==,<=,>=,!= karakterleri mantıksal operatörlerdir.

2. ( ) Şart ifadesi sağlanmadığında çalışacak kodlar "else" blokunun içine yazılır.

3. ( ) Karar ifadelerinde ve(&&) operatörü veya(||) operatöründen öncelikli işleme alınır.

4. ( ) Döngü şart ifadesi sağlanmasa da do-while döngüsü en az bir kez çalışamaz.

5. ( ) Continue deyimi döngünün çalışmasını bitirmek ve döngüden çıkmak için kullanılır.

6. ( ) Düzenli ifadeler, programda belirtilen içeriklerdeki belli kurallara uyan bölümleri bulmak için yazılan ifadelerdir.

B) Aşağıdaki soruları dikkatlice okuyarak doğru seçeneği işaretleyiniz.

7. $a = true; $b=false; $c=true; ; $d=false; olması durumunda aşağıdaki mantıksal ifadelerin hangisi TRUE değerini geri döndürür?

A) (!$a && $c) || ($b)|| ($d);
B) ($a) && ($c || $b) || ($d);
C) ($a || $b) && ($b) && (!$d);
D) ($a || $b) && (!$d) && ($b && $c) && $a;
E) ($a || !$b) && ($d) && ($b && $c) && !$a;

8. AND operatörünün karşılığı aşağıdakilerden hangisidir?

A) && B) || C) ==
D) ?? E) !

9. Aşağıdakilerden hangisi mantıksal operatördür?

A) >= B) != C) <>
D) = = E) OR

10. Aşağıdakilerden hangisi PHP'de if kontrol yapısını gösterir?

A) if (şart) {işlemler}
B) if (şart) :{işlemler} endif
C) if {şart} (işlemler)
D) if {şart} : (işlemler) end ;
E) {şart} if (işlemler)

11.

```
<?php
$x=4;  $y=5a;  $z=3;
if($x<$z)  $y++;
if($x>=$y)  $y-=$z;
if  ($z>$x)  $x=$y++;
if($x>=$z)  $y=$x=$z++;
echo  "$x  $y  $z";
?>
```

Verilen bu PHP kodu ekrana aşağıdakilerden hangisini yazar?

A) 3 3 4 B) 4 3 3 C) 4 3 4
D) 4 3 7 E) 5 5 6

12. Aşağıdakilerden hangisi karar kontrol deyimidir?

A) do-while B) for C) foreach
D) switch E) while

13. For döngüsü için aşağıdakilerden hangisi <u>yanlıştır?</u>

A) Artış veya azalış değeri her zaman 1'dir.
B) Başlangıç değerini belirleyebilir.
C) Bitiş değerini belirleyebilir.
D) Döngünün kaç defa döneceği bilinir.
E) İstenilen yerde döngü sonlandırılabilir.

14.

```
< ?php
$i=0;
while ($i<5)
{ echo “Gazi MTAL”;
$i++; }
 ?>
```

Verilen bu program parçası aşağıdaki işlemlerden hangisini yapar?

A) Gazi MTAL'nin harf sayısını verir.
B) 5 defa yan yana Gazi MTAL yazar.
C) 5 defa tersten Gazi MTAL yazar.
D) 5 defa alt alta Gazi MTAL yazar.
E) Gazi MTAL'nin sondan 4 harfini alarak ekrana MTAL yazar.

15.

```
<?php
  $i=0;
  do{
        if($i%2==0)
        echo  “$i”;
        $i++;
  }while($i<=5);
?>
```

Aşağıdaki kodun ekran çıktısı aşağıdakilerden hangisidir?

A) 01234 B) 012345 C) 024 D) 24 E) 246

16. Aşağıdakilerden hangisi PHP'de yer alan foreach döngüsünde kullanılacak şart ifadesi için doğru bir örnektir?

A) Foreach($i=10;$i>=1;$i--)
B) Foreach($dongu<=14)
C) Foreach($deger=0;;$deger++)
D) Foreach($katalog as $urun)
E) Foreache($deger=0; $katalog as $urun)

17.

```
<?php
$j=0;
for($k=5;$k>=1;$k--)
{
if($k%2==1)  continue;
$j+=$k;
}
echo  $j;
?>
```

Verilen bu kodların çalışması durumunda ekran çıktısı aşağıdakilerden hangisidir?

A) 3 B) 5 C) 6 D) 7 E) 9

18. Hazırlanan düzenli ifadenin içerik içinde olup olmadığını kontrol edip eğer düzenli ifade içerik içerisinde mevcutsa 1, mevcut değilse 0 sonucunu döndüren fonksiyon aşağıdakilerden hangisidir?

A) preg_match
B) preg_match_all
C) preg_replace
D) preg_split
E) preg_grep

19. Belirlen metin içinde *ornek* kelimesi ile başlayan cümle olup olmadığını kontrol eden yazım şekli aşağıdakilerden hangisidir?

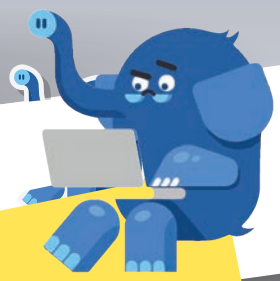
A) ornek
B) ^ornek
C) ornek$
D) ^ornek$
E) ornek^

20. Büyük / küçük harf ayrımı yapmadan metin içinde *De* hecesini bulmak için gerekli düzenli ifade aşağıdakilerden hangisidir?

A) $kelime='/De/i';
B) $kelime='/De/m';
C) $kelime='/De/s';
D) $kelime='/De/u';
E) $kelime='/De/a';

3. ÖĞRENME BİRİMİ

DİZİ YAPILARI



KONULAR

3.1. DİZİ YAPILARI

3.2. DİZİLERLE İŞLEMLER

NELER ÖĞRENECEKSİNİZ?

- Tek boyutlu, çok boyutlu ve çağrışımsal dizi kavramları
- Dizileri üzerinde işlem yapma

TEMEL KAVRAMLAR

tek boyutlu diziler, çok boyutlu diziler, çağrışımsal diziler, diziler.



Hazırlık Çalışmaları

1. **Dizileri araştırarak günlük hayatta nasıl kullanıldığını tartışınız.**
2. **Programlama dillerinde kullanılan dizilerin sağladığı kolaylıkları araştırınız.**
3. **PHP programlama dilinde kullanılan dizileri araştırarak matematik dersinde kullanılan matrisler ile karşılaştırınız.**

3.1. DİZİ YAPILARI

Aynı veya farklı türden çok sayıda değeri üzerinde tutabilen özel değişkenlere dizi (array) denir. Bu yapı hem kod yazımını kolaylaştırır hem de verileri daha düzenli bir şekilde kullanmayı sağlar.

```
$araba_tur="Sedan";
$araba_tur2="Wagon";
$araba_tur2="SUV";
$araba_tur3="Hatcback";
$araba_tur4="Sport";
$araba_tur5="Van";
…….
```

Örneğin bir bilgisayar programında araba türlerinin tanımlanması isteniyor. Girilen tüm türlerin ayrı bir değişkende tutulması gerektiği varsayıldığında her tür için ayrı bir değişken tanımlamak gerekir. Bu tür kullanımlarda hem değişkenleri tanımlamak hem de tanımlanmış değişkenlerden değer okumak zordur. Bu tür işlemleri kolaylaştırmak ve zamandan tasarruf etmek için diziler kullanılır.

Dizi Tanımlama

Dizi tanımlamak için farklı yöntemler vardır. Yazılan programdaki ihtiyaca göre farklı dizi tanımlama yöntemleri kullanılır. Genel olarak tek boyutlu en basit dizilerin kullanımı şu şekildedir:

```
<?php
$dizi_adi=array(ifade1,ifade2,ifade3,…….);
?>
```

Dizi tanımlanırken **array()** deyimi kullanılır. Array() deyiminin içine istenilen sayıda eleman aralarına virgül konularak yazılır. Array() içinde yazılan eleman herhangi bir türde olabilir.

1. Uygulama

Haftanın günleri ile mevsimleri içeren farklı tek boyutlu dizileri tanımlama işlemini aşağıdaki adımları takip ederek gerçekleştiriniz.

1. Adım: index.php dosyasını server içinde oluşturunuz.

2. Adım: Tarayıcıda çalışacak olan index.php sayfasının title kısmını "Dizi Tanımlama" olarak ayarlayınız.

3. Adım: <?php ?> içine aşağıdaki kodları yazınız.

4. Adım: PHP sayfasını kaydediniz ve serverda bu sayfayı çalıştırarak görüntüleyiniz.

```
<!DOCTYPE html>
<html lang=”tr”><head>
   <meta charset=“UTF-8”>
   <title>Dizi Tanımlama</title>
</head><body>
   <?php
   //günler için dizi tanımlanıyor
   $gunler=array(“Pazartesi”,“Salı”,“Çarşamba”,“Perşembe”,“Cuma”,“Cumartesi”,“Pazar”);
   //aylar için farklı bir dizi tanımlama yöntemi ile yapılıyor
   $mevsimler=array();
   $mevsimler[]=“İlkbahar”;
   $mevsimler[]=“Yaz”;
   $mevsimler[]=“Sonbahar”;
   $mevsimler[]=“Kış”;
   print_r($gunler);
   echo“<pre>”;
        var_dump($mevsimler);
   ?>
</body>
</html>
```

Önemli

var_dump komutu bir değişkenin içeriğini ve türünü öğrenmek için kullanılır.

Sıra Sizde

Ayları ekrana yazdıran tek boyutlu dizi için gerekli PHP kodlarını yazınız.

Değerlendirme

Çalışmanız diğer sayfada yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

KONTROL LİSTESİ

Aşağıda listelenen ölçütlerden öğrencide gözlenen davranış için Evet, gözlenemeyen davranış için Hayır kutucuğunun altına (X) işareti koyarak işaretleyiniz.

Ölçütler	Evet	Hayır
1. Server içindeki dizi klasöründe dizitanimlama.php dosyası oluşturuldu.		
2. $aylar şeklinde dizi değişkeni tanımlanarak değerleri atandı.		
3. $aylar dizisinin değerleri ekrana yazdırıldı.		
4. Programların çalışıp çalışmadığı kontrol edildi.		
5. Çalışmada iş sağlığı ve güvenliği kurallarına dikkat edildi.		
6. Zaman verimli kullanıldı.		

Dizi Elemanlarına Erişim

Program içinde tanımlı olan bir dizinin elemanlarına erişmek için köşeli parantezler içine ([4]) indis değeri belirtilmelidir. Dizi tanımlarken indis numarası verilmediği zaman indis değeri varsayılan olarak sıfırdan (0) başlar.

```
<?php
  //günler için dizi tanımlanıyor
$gunler=array("Pazartesi","Salı","Çarşamba","Perşembe","Cuma","Cumartesi","Pazar");
   echo $gunler[2]."<br>" #ekrana Çarşamba yazar, indisin sıfırdan başladığını unutmayınız
   echo $gunler[0] #ekrana Pazartesi yazar, indisin sıfırdan başladığını unutmayınız
   ?>
```

Yukarıdaki örnekte $gunler şeklinde tanımlanan dizinin elemanlarına ne şekilde ulaşılacağı gösterilmiştir.

Çağrışımsal (Associative) Diziler

Diziler tanımlanırken indis değeri belirtilmediğinde indis değeri varsayılan olarak sıfırdan (0) başlar. Bazen dizi tanımlarken özel indis değerleri vermek gerekebilir. Çağrışımsal diziler isimlendirilmiş indislere atama yapılarak tanımlanır. Çağrışımsal dizi oluşturmanın iki yolu bulunur.

Birinci yol:

```
$plaka=array("Anakara"=>"06","Elazığ"=>"23","İstanbul"=>"34");
```

Tanımlanacak dizide özel olarak indis değeri belirtmek için indis ismi yazıldıktan sonra eşit ve büyük işaretleri (=>) kullanılarak verilecek değer belirtilir.

İkinci yol:

```
<?php
 $plaka=array();
  $plaka["Ankara"]="06";
  $plaka["Elazığ"]="23";
  $plaka["İstanbul"]="34";
  ?>
```

Bu yol tanımlama biraz daha uzundur. İlk tanımlamadan farklı olarak köşeli parantezler içine indis adları yazılır.

Çağrışımsal Dizi Elemanlarına Erişim

Program içinde tanımlı olan bir dizinin elemanlarına erişmek için köşeli parantezler içine (["Elazığ"]) indis değeri belirtilmelidir.

```
<?php
    //plakalar için dizi tanımlanıyor
    $plaka=array("Ankara"=>"06","Elazığ"=>"23","İstanbul"=>"34");
    echo $plaka["Elazığ"]."<br>" #ekrana 23 yazar
    echo $plaka["İstanbul"] #ekrana 34 yazar
    ?>
```

Yukarıdaki örnekte plaka dizisinin içinde verilen illerin plaka değerleri öğrenilmek isteniyor.

Çok Boyutlu Diziler

Tek boyutlu diziler bir indis => bir değer şeklinde verileri tutar ancak bazı durumlarda bir indis ile birden fazla veri tutmak gerekir. Böyle durumlarda çok boyutlu diziler kullanılır. Kısaca dizi içinde dizi kullanmaya çok boyutlu dizi denir. Çok boyutlu dizilerde indis değerleri, çağrışımsal (=>) veya standart olarak tanımlanabilir. Programlarda erişim kolaylığı bakımında indis değerinin çağrışımsal (=>) olarak tanımlanması gerekir.

Kullanım şekli:

```
<?php
   $dizi_adi    =  array(
   indis1ifade => array(değer1,değer1....),
   indis2ifade => array(değer2,değer2....),
                     ...);
 ?>
```

2. Uygulama

Müşteri			
Ad Soyad	**Şehir**	**Yaş**	**Meslek**
X Kişisi	Elazığ	38	Doktor
Y Kişisi	İstanbul	45	Öğretmen

Yukarıdaki müşteri tablosunun çok boyutlu dizi şeklinde tanımlama işlemini aşağıdaki adımları takip ederek gerçekleştiriniz.

1. Adım: index.php dosyasını server içinde oluşturunuz.

2. Adım: Tarayıcıda çalışacak olan index.php sayfasının title kısmını "Çok Boyutlu Dizi Tanımlama" olarak ayarlayınız.

3. Adım: Öncelikle müşteri şeklinde içi boş tek boyutlu bir dizi yazınız.

```
<!DOCTYPE html>
<html lang="tr">
<head>    <meta charset="UTF-8">    <title>Çok Boyutlu Dizi</title></head>
<body>
    <?php
     //Öncelikle Musteri dizisi tanımlanıyor
     $musteri=array();
    ?>
</body></html>
```

4. Adım: Oluşturduğunuz dizi içine her müşteri için ayrı bir dizi oluşturunuz.

```php
<?php
  $musteri=array(
  0=>array(),//her müşteri için ayrı ayrı dizi oluşturuluyor
  1=>array(),
  );
  ?>
```

```php
<?php
  $musteri=array( );
  $musteri[0]=array();//her müşteri için ayrı ayrı dizi oluşturuluyor
  $musteri[1]=array();
  ?>
```

5. Adım: Oluşturduğunuz müşteri için bilgileri giriniz.

```
0=>array("adsoyad"=>"X Kişisi","sehir"=>"Elazığ","yas"=>"38","meslek"=>"Doktor"),
1=>array("adsoyad"=>"Y Kişisi","sehir"=>"İstanbul","yas"=>"43","meslek"=>"Öğretmen"),
```

veya

```
$musteri[0]=array("adsoyad"=>"X Kişisi","sehir"=>"Elazığ","yas"=>"38","meslek"=>"Doktor");
$musteri[1]=array("adsoyad"=>"Y Kişisi","sehir"=>"İstanbul","yas"=>"43","meslek"=>"Öğretmen");
```

6. Adım: Ekrana yazdırmak için gerekli kodları ekleyip serverda çalıştırınız.

```
echo"<pre>";
  var_dump($musteri);
```

Sıra Sizde

Diğer sayfada verilen tablodaki özellikleri çok boyutlu diziye çevirmek ve tabloyu yazdırmak için gerekli PHP kodlarını yazınız.

Öğrenci Tablosu				
Sıra No	**Okul No**	**Sınıf**	**Ad Soyad**	**Cinsiyet**
1	65	ATP-9A	X Kişisi	Erkek
2	125	AMP-9A	Y Kişisi	Kız
3	128	AMP-9D	Z Kişisi	Erkek

Değerlendirme

Çalışmanız aşağıda yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

KONTROL LİSTESİ

Diğer sayfada listelenen ölçütlerden öğrencide gözlenen davranış için Evet, gözlenemeyen davranış için Hayır kutucuğunun altına (X) işareti koyarak işaretleyiniz.

Ölçütler	Evet	Hayır
1. Server içindeki dizi klasöründe dizitanimlama1.php dosyası oluşturuldu.		
2. $ogrenci şeklinde dizi değişkeni tanımlandı.		
3. $ogrenci dizisi içine tabloda bulunan üç satır için ayrı ayrı öğrenci tanımlaması yapılarak ekrana yazdırıldı.		
4. Programların çalışıp çalışmadığı kontrol edildi.		
5. Çalışmada iş sağlığı ve güvenliği kurallarına dikkat edildi.		
6. Zaman verimli kullanıldı.		

Çok Boyutlu Dizilerde Elemanlara Erişim

Tek boyutlu dizilerde olduğu gibi çok boyutlu dizilerde ulaşılmak istenilen eleman için indis köşeli parantezler içinde kullanılmalıdır.

```
<?php
$musteri=array(
   array("adsoyad"=>"X Kişisi","sehir"=>"Elazığ","yas"=>"38","meslek"=>"Doktor"),
   array("adsoyad"=>"Y Kişisi","sehir"=>"İstanbul","yas"=>"43","meslek"=>"Öğretmen"),
   );
   echo $musteri[1]["adsoyad"];      ?>
```

Örnekte $musteri isimli bir boyutlu ve iki elemanlı bir dizi oluşturulmuştur. Bu dizide herhangi bir elemana ulaşmak için indis değerleri köşeli parantez içinde verilmelidir.1 No.lu müşterinin ad soyadına ulaşmak için $musteri[1]["adsoyad"]; yazılır.

3.2. DİZİLERLE İŞLEMLER

Birden fazla veri türü üzerinde işlem yapmak için farklı veri yapıları geliştirilmiştir. Genel olarak bu veri yapıları dizilerdir. Diziler üzerinde ekleme, çıkarma, arama, sıralama vb. birçok işlem yapılabilir.

Döngü ile Dizi Elemanlarına Erişim

Dinamik olarak tanımlanan dizilerde elemanlara tek tek ulaşmak hem zaman kaybına hem de program içinde karmaşıklığa yol açabilir. Bu nedenle diziyi baştan sona doğru taramak amacıyla döngüler kullanılır. Dizilerle birlikte en sık kullanılan döngü foreach'tir. Dizilerde for döngüsü de kullanılabilir. For döngüsü kullanımında öncelikle count() fonksiyonu ile dizinin boyutu tespit edilmelidir.



Sınıf listesinde bulunan öğrencilerin adlarını ekrana yazan programı aşağıdaki adımları takip ederek gerçekleştiriniz.

1. Adım: index.php dosyasını server içinde oluşturunuz.

2. Adım: Tarayıcıda çalışacak olan index.php sayfasının title kısmını "Döngü ile Dizi Elemanlarına Erişim" olarak ayarlayınız.

3. Adım: <?php ?> içine foreach ile ilgili kodları yazınız.

```
<!DOCTYPE html>
<html lang="tr"><head> <title>Döngü İle  Dizi Elemanlarına Erişim</title></head>
<body>
    <?php
   $liste=array("Ahmet","Veli","Ayşe","Vedat","Mustafa","Veysel","Derya","Hacer" ,"Selim" );
    foreach($liste as $ad){
     // döngü her dönüşte dizi içindeki eleman $ad değişkenine atanır.
     echo $ad."<br>";
     }
     ?>
</body>  </html>
```

4. Adım: PHP sayfasını kaydediniz ve serverda bu sayfayı çalıştırarak görüntüleyiniz.

5. Adım: <?php ?> içine for döngüsü ile ilgili kodları yazınız.

```
<?php
  $liste=array("Ahmet","Veli","Ayşe","Vedat","Mustafa","Veysel","Derya","Hacer","Selim" );
   for($indis=0;$indis<count($liste);$indis++){
    echo $liste[$indis]."<br>";
   //Döngü sıfırdan başlayıp dizinin boyutuna kadar tek tek artarken her
artışta dizi elemanları indise göre ekrana yazdırılacaktır.
   }
 ?>
```

6. Adım: PHP sayfasını kaydediniz ve serverda bu sayfayı çalıştırarak görüntüleyiniz.

Sıra Sizde

```
$musteri=array{
        "adsoyad"=>"Ahmet Ahmet",
        "sehir"=>"Elazığ",
        "Borç"=>20000,
    }
```

Şeklinde tanımlanan bir diziyi foreach ve for döngüsü kullanarak anahtarlarıyla birlikte ekrana yazan PHP kodlarını yazınız.

Sıra Sizde

$sayilar=array(45,3,12,17,4,24,65,85,741); şeklinde verilen dizinin ortalamasını bulan PHP programını yazınız.

Değerlendirme

Çalışmanız aşağıda yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

KONTROL LİSTESİ

Aşağıda listelenen ölçütlerden öğrencide gözlenen davranış için Evet, gözlenemeyen davranış için Hayır kutucuğunun altına (X) işareti koyarak işaretleyiniz.

Ölçütler	Evet	Hayır
1. Server içindeki dizi klasöründe foreachdizi.php dosyası oluşturuldu.		
2. $musteri dizisi programda yazıldı.		
3. $musteri dizisindeki elemanları anahtarlarıyla birlikte ekrana yazdırmak için foreach döngü bloku yazıldı.		
4. $musteri dizisindeki elemanları anahtarlarıyla birlikte ekrana yazdırmak için for döngü bloku yazıldı.		
5. Foreach döngüsü ile $sayilar dizisi üzerindeki elemanlara ulaşılıp tek tek alınarak toplanması için gerekli kod bloku yazıldı.		
6. Programların çalışıp çalışmadığı kontrol edildi.		
7. Çalışmada iş sağlığı ve güvenliği kurallarına dikkat edildi.		
8. Zaman verimli kullanıldı.		

Dizilerde Eleman Çıkartma

Oluşturulan bir dizinin başına ve sonuna değer eklenip çıkartılabilir. Bu işlemler Tablo 3.1'de verilen yardımcı fonksiyonlar yardımıyla yapılır.

Tablo 3.1: *Yardımcı Fonksiyonlar*

Fonksiyon Adı	Açıklaması
unset	Dizide verilen indisteki değeri kaldırır.
array_unshift	Dizinin başına veri eklemek için kullanılır.
array_shift	Dizinin başındaki veriyi kaldırmak için kullanılır.
array_push	Dizinin sonuna veri eklemek için kullanılır.
array_pop	Dizinin sonundaki veriyi kaldırmak için kullanılır.

4. Uygulama

$kitaplar=array(“Çalıkuşu”,“Dokuzuncu Hariciye Koğuşu”,“Sinekli Bakkal”); şeklinde tanımlanan bir dizi-ye ekleme ve diziden çıkarma işlemlerini aşağıdaki adımları takip ederek gerçekleştiriniz.

1. Adım: index.php dosyasını server içinde oluşturunuz.

2. Adım: <?php ?> içinde $kitaplar dizisini oluşturunuz.

3. Adım: $kitaplar dizisinin başına **Suç ve Ceza**, sonuna ise **Vadideki Zambak** kitaplarını eklemek için gerekli kodları yazınız.

```
<?php
   $kitaplar=array(“Çalıkuşu”,“Dokuzuncu Hariciye Koğuşu”,“Sinekli Bakkal”);
   array_unshift($kitaplar,“Suç ve Ceza”);
   array_push($kitaplar,“Vadideki Zambak”);
   echo“<pre>”;
   print_r($kitaplar);
    ?>
```

4. Adım: $kitaplar dizisinin başındaki ve sonundaki kitapları diziden çıkarmak için gerekli kodları yazınız.

```
array_shift($kitaplar);
   array_pop($kitaplar);
   echo“<pre>”;
   print_r($kitaplar);
```

5. Adım: $kitaplar dizisinin 1 No.lu indisinde bulunan elemanı silmek için gerekli kodları yazınız.

```
unset($kitaplar[1]);
   echo“<pre>”;
   print_r($kitaplar);
```

6. Adım: PHP sayfasını kaydediniz ve serverda bu sayfayı çalıştırarak görüntüleyiniz.

Sıra Sizde

Sınıfınızdaki öğrencilerin sadece adlarını içeren bir dizi tanımladıktan sonra aşağıdaki işlemleri yapınız.

a) 1, 5, 9 No.lu indise sahip olan öğrenci adlarını silen PHP kodlarını yazınız.

b) Sildiğiniz 1 ve 5 no.lu indise sahip olan öğrencileri dizin başına, 9 No.lu indise sahip olan öğrenciyi dizinin sonuna ekleyen PHP kodlarını yazınız.

c) Dizin başından ve sonundan iki tane öğrenci adını çıkaran PHP kodunu yazınız.

Değerlendirme

Çalışmanız diğer sayfada yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

KONTROL LİSTESİ

Aşağıda listelenen ölçütlerden öğrencide gözlenen davranış için Evet, gözlenemeyen davranış için Hayır kutucuğunun altına (X) işareti koyarak işaretleyiniz.

Ölçütler	Evet	Hayır
1. Server içindeki dizi klasöründe dizieklecikar.php dosyası oluşturuldu.		
2. $sinif dizi için gerekli tanımlama yapıldı.		
3. $sinif dizi için gerekli değerler atandı.		
4. $sinif dizine değerler atandı.		
5. $sinif dizisi içinde 1, 5, 9 No.lu öğrencileri silmek için gerekli kod bloku yazıldı.		
6. $sinif dizisinde silinen 1 ve 5 No.lu öğrencileri dizinin başına eklemek için gerekli kod bloku yazıldı.		
7. $sinif dizisinde silinen 9 No.lu öğrenciyi dizinin sonuna eklemek için gerekli kod bloku yazıldı.		
8. $sinif dizisinin başından ve sonundan iki öğrenci silmek için gerekli kod bloku yazıldı.		
9. Programların çalışıp çalışmadığı kontrol edildi.		
10. Çalışmada iş sağlığı ve güvenliği kurallarına dikkat edildi.		
11. Zaman verimli kullanıldı.		

Dizilerde Birleştirme

Birden fazla sayıda diziyi birleştirip yeni dizi oluşturmak için **array_merge()** fonksiyonu kullanılır. Bu fonksiyon dizilerin değerlerini değiştirmez, dizileri birbirine ekleyerek yeni dizi oluşturur.

```
<?php
   $yaz_meyveleri=array(“karpuz”,“kiraz”,“kavun”,“üzüm”);
   $kis_meyveleri=array(“portakal”,“mandalina”);
   $meyveler=array_merge($yaz_meyveleri,$kis_meyveleri);
   echo“<pre>”;
   print_r($meyveler);
   ?>
```

Yukarıdaki örnekte array_merge() fonksiyonu ile yaz ve kış meyveleri birleştirilerek meyveler adına yeni bir dizi oluşturulmuş ve bu dizilerin elemanları diziye aktarılmıştır.

Dizilerin Ortak Elemanını Bulma

Tanımlanan birden fazla dizide aynı elemanları bulmak için **array_intersect()** fonksiyonu kullanılır. Bu fonksiyon diziler içinde aynı elemanları bulup yeni dizi oluşturur. Büyük ve küçük harf duyarlılığı vardır.

```
<?php
   $renk1=array("Kırmızı","Mavi","Sarı","Siyah","Beyaz","Gri");
   $renk2=array("Yeşil","Mavi","kırmızı","siyah","Beyaz","gri");
   $renk3=array("yeşil","Mavi","Sarı","siyah","Beyaz","Gri");
   $son_renk=array_intersect($renk1,$renk2,$renk3);
   echo"<pre>";
   print_r($son_renk);
   ?>
```

Yukarıdaki örnekte $son_renk dizisinin elemanları Mavi ve Beyaz olacaktır. Büyük ve küçük harf duyarlılığı olduğundan Gri ve gri değerleri birbirinden farklı yorumlandığı için ortak eleman olarak kabul edilmez.

Dizilerin Farkını Bulma

Tanımlanan birden fazla dizide aynı olmayan elemanları bulmak için **array_diff()** fonksiyonu kullanılır. Bu fonksiyon, diziler içinde aynı elemanları bulup yeni dizi oluşturur. Büyük ve küçük harf duyarlılığı vardır.

```
<?php
   $renk1=array("Kırmızı","Mavi","Sarı","Siyah","Beyaz","Gri");
   $renk2=array("Yesil","Mavi","kırmızı","siyah","Beyaz","Gri");
   $son_renk=array_diff($renk1,$renk2);
   echo"<pre>";
   print_r($son_renk);
   ?>
```

Yukarıdaki örnekte $renk1 dizisinde olup da $renk2 dizisinde olmayan değerleri $son_renk dizisine ekler. $son_renk dizisinin elemanları Kırmızı, Sarı ve Siyah olacaktır. Büyük ve küçük harf duyarlığı olduğundan Siyah ve siyah değerleri birbirinden farklı yorumlandığı için farklı değer olarak alır.

Dizideki Tekrarlanan Değerleri Silme

Tanımlanan bir dizide tekrarlanan değerlerden sadece bir tanesini alıp diğerlerini diziden çıkartmak için **array_uniqe()** fonksiyonu kullanılır. Bu fonksiyon tekrarlanan elemanlar içinde ilk elemanı alır. Çok boyutlu dizilerde kullanılmaz.

```
<?php
   $sayilar=array(10,10,5,6,6,10,5,1,2,6,2,1);
   $son_sayilar=array_unique($sayilar);
   echo"<pre>";
   print_r($son_sayilar);
   ?>
```

Yukarıdaki örnekte $sayilar dizisinde tekrarlanan elemanlar olmasına rağmen **array_uniqe()** fonksiyonuna uygulanması sonucunda $son_sayilar dizisindeki elemanlar tekrarlanmayan benzersiz bir şekilde dizilir.

Dizi Elemanlarını Tek Seferde Ekrana Yazdırma

Tanımlanan bir dizinin yan yana tüm elemanlarını tek seferde yazdırmak için **implode()** fonksiyonu kullanılır.

```
<?php
   $sayilar=array(10,10,5,6,6,10,5,1,2,6,2,1);
   echo implode("-",$sayilar);
   ?>
```

Yukarıdaki örnekte $sayilar dizisindeki elemanlar sırası ile yan yana yazdırılırken aralarına ayraç olarak kısa çizgi işareti eklenmesi gerektiği **implode()** fonksiyonu ile belirlenir.

Dizide Rastgele Eleman Alma

Tanımlanan bir dizi içinden rastgele bir eleman almak için **array_rand()** fonksiyonu kullanılır. **Array_rand()** fonksiyonu dizi içinde alınacak elemanın indisini verir.

```
<?php
   $kitaplar=array(
   0=>array("kitap adı"=>"Çile","yazar"=>"Necip Fazıl Kısakürek"),
   1=>array("kitap adı"=>"Bu Ülke","yazar"=>"Cemil Meriç"),
   2=>array("kitap adı"=>"Anadolu Notları","yazar"=>"Reşat Nuri Güntekin"),
   3=>array("kitap adı"=>"Çankaya","yazar"=>"Falih Rıfkı Atay"),
   );
   $rastgele=array_rand($kitaplar);
   echo $kitaplar[$rastgele]["kitap adı"]." --->";
   echo $kitaplar[$rastgele]["yazar"];
   ?>
```

Yukarıdaki örnekte $kitaplar dizisinde rastgele bir kitabın indisi alınıyor. Rastgele seçilen kitabın indisine göre ekrana kitabın adı ile yazarı yazdırılıyor.

Dizilerde Sıralama Yapma

Dizilere girilen verileri sıralama yapmak için Tablo 3.2'deki fonksiyonlar kullanılır.

***Tablo 3.2:** Sıralama Fonksiyonları*

Fonksiyon Adı	Açıklama
asort()	Bir dizinin elemanlarını ilişkilerini bozmadan küçükten büyüğe doğru sıralar.
arsort()	Bir dizinin elemanlarını ilişkilerini bozmadan büyükten küçüğe doğru sıralar.
sort()	Bir dizinin elemanlarını ilişkilerini korumadan büyükten küçüğe doğru sıralar.
rsort()	Bir dizinin elemanlarını ilişkilerini korumadan küçükten büyüğe doğru sıralar.
shuffle()	Bir dizinin elemanlarını ilişkilerini korumadan rastgele sıralar.
natcasesort()	Bir diziyi harf büyüklüğüne göre duyarlı olacak şekilde "doğal sıralama" algoritması ile sıralar.
natsort()	Bir diziyi "doğal sıralama" algoritması ile sıralar.
ksort()	Bir diziyi anahtarlarına göre küçükten büyüğe doğru sıralar.
krsort()	Bir diziyi anahtarlarına göre büyükten küçüğe doğru sıralar.
uasort()	Kullanıcı tanımlı bir girdi kullanılarak indislerine göre ilişkileri bozmadan sıralama yapar.
uksort()	Kullanıcı tanımlı bir girdi kullanılarak anahtarlara göre sıralama yapar.
usort()	Kullanıcı tanımlı bir girdi kullanılarak değerlere göre sıralama yapar.
array_multisort()	Çok boyutlu fonksiyonlarda sıralama yapmak için kullanılır.

5. Uygulama

```
$notlarim=array(
       “Bilişim Teknolojilerinin Temelleri”=>75,
       “Programlama Temelleri”=>100,
       “Bilgisayarlı Tasarım Uygulamaları”=>85,
       “Mesleki Gelişim Atölyesi”=>95);
```

$notlarim şeklinde tanımlanan bir dizinin sıralama işlemlerini aşağıdaki adımları takip ederek gerçekleştiriniz.

1. Adım: index.php dosyasını server içinde oluşturunuz.

2. Adım: <?php ?> içinde $notlarim dizisini oluşturunuz.

3. Adım: $notlarim dizisinin anahtar indislerini bozmadan küçükten büyüğe ve büyükten küçüğe doğru sıralayan kodları yazınız.

```
<?php
   $notlarim=array(
       “Bilişim Teknolojilerinin Temelleri”=>75,
       “Programlama Temelleri”=>100,
       “Bilgisayarlı Tasarım Uygulamaları”=>85,
       “Mesleki Gelişim Atölyesi”=>95);
    asort($notlarim);
    echo“<pre>”;
    print_r($notlarim);
  arsort($notlarim);
    echo“<pre>”;
    print_r($notlarim);
?>
```

4. Adım: $notlarim dizisinin anahtar indislerini değerlerini korumadan küçükten büyüğe ve büyükten küçüğe doğru sıralayan kodları yazınız.

```
sort($notlarim);
    echo“<pre>”;
    print_r($notlarim);
    rsort($notlarim);
    echo“<pre>”;
    print_r($notlarim);
```

5. Adım: $notlarim dizisini büyük / küçük harf ayrımı yapmadan küçükten büyüğe ve büyükten küçüğe doğru sıralayan kodları yazınız.

```
   natcasesort($notlarim);
    echo“<pre>”;
    print_r($notlarim);
    natsort($notlarim);
    echo“<pre>”;
    print_r($notlarim);
```

6. Adım: PHP sayfasını kaydediniz ve serverda bu sayfayı çalıştırarak görüntüleyiniz.

Sıra Sizde

Rastgele üretilmiş birbirinden farklı en az 10 sayıdan oluşan üç diziyi tek dizi hâline getirip bu dizi üzerinde sıralama fonksiyonlarını uygulayarak tek tek ekrana yazdıran PHP programını yazınız.

Değerlendirme

Çalışmanız aşağıda yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

KONTROL LİSTESİ

Aşağıda listelenen ölçütlerden öğrencide gözlenen davranış için Evet, gözlenemeyen davranış için Hayır kutucuğunun altına (X) işareti koyarak işaretleyiniz.

Ölçütler	Evet	Hayır
1. Server içindeki dizi klasöründe dizisıralama.php dosyası oluşturuldu.		
2. Üç tane birbirinden farklı, 10 tane rastgele sayı içeren dizi tanımlamaları yapıldı.		
3. Üç dizi tek dizi hâline getirildi.		
4. Oluşturulan dizi üzerinde sıralama fonksiyonları tek tek uygulanıp ekrana yazdırmak için gerekli kod blokları yazıldı.		
5. Programların çalışıp çalışmadığı kontrol edildi.		
6. Çalışmada iş sağlığı ve güvenliği kurallarına dikkat edildi.		
7. Zaman verimli kullanıldı.		

Dizileri Tersine Çevirme

Dizideki elemanları sondan başa doğru yazdırmak için **array_reverse()** fonksiyonu kullanılır.

```
<?php
   //Rastgele 1 ile 50 arasında 10 sayı diziye ekleniyor.
   $sayilar=array();
   for($i=1;$i<=10;$i++){
       array_unshift($sayilar,rand(1,50));
   }
   echo“<pre>”;
   print_r($sayilar);
   $sonuc=array_reverse($sayilar);
   echo“<pre>”;
   print_r($sonuc);
   ?>
```

Yukarıdaki örnekte 1 ile 50 arasında rastgele 10 tane sayıyı $sayilar dizisine ekledikten sonra **array_reverse()** fonksiyonu ile ters çevirerek $sonuc dizisine ekliyor.

Dizi İşaretçileri

Dizideki verilere döngü yardımıyla baştan sona doğru tek tek ulaşılabilir. Bazı programlarda dizi içinde istenilen elemanın öncesindeki, sonrasındaki veya dizinin en başındaki, en sonundaki elemanları almak için gerekli olan hazır fonksiyonlar Tablo 3.3'te gösterilmiştir.

***Tablo 3.3:** Dizi İşaretçileri*

Fonksiyon Adı	Açıklama
current()	Dizi göstericisinin üzerinde bulunduğu elemanını gösterir.
pos()	Dizi göstericisinin üzerinde bulunduğu elemanını gösterir.
prev()	Dizi göstericisini bir önceki konuma taşır.
next()	Dizi göstericisini bir sonraki konuma taşır.
reset()	Dizi göstericisini başlangıç konumuna taşır.
end()	Dizi göstericisini dizinin sonuna taşır.
list()	Değişkenlere bir dizi gibi atama yapar.
each()	Dizi göstericisinin üzerinde bulunan elemanı verir ve göstericiyi bir ilerletir.

```php
<?php
   //Rastgele 1 ile 20 arasında 10 sayı diziye ekleniyor.
   $sayilar=array();
   for($i=1;$i<=10;$i++){
       array_unshift($sayilar,rand(1,20));
   }
   echo“<pre>”;
   print_r($sayilar);
   echo current($sayilar).“ ”;
   echo next($sayilar).“ ”;
   echo pos($sayilar).“ ”;
   echo next($sayilar).“ ”;
   echo next($sayilar).“ ”;
   echo prev($sayilar).“ ”;
   echo end($sayilar).“ ”;
   echo current($sayilar).“ ”;
   echo reset($sayilar).“ ”;
   echo pos($sayilar).“ ”;
   ?>
```

Yukarıdaki örnekte diziler üzerinde işaretçi (gösterici) işlemleri yapılmıştır. Dizi işaretçileri ile dizi içinde ileri, geri, başa git, sona git işlemleri oldukça kolay yapılır. Eğer dizide yeterli sayıda eleman olmadığı durumlarda işaretçi işlemleri yapılmak istenirse boş (null) değer döndürebilir.

Dizi Sayıcıları

Dizinin kaç elemanlı olduğunu, sayısal veri içeren dizilerin elemanlarının toplamını, çarpımını bulmak

için kullanılan hazır fonksiyonlar Tablo 3.4'te gösterilmiştir.

Tablo 3.4: *Dizi Sayıcıları*

Fonksiyon Adı	Açıklama
count()	Dizideki toplam eleman sayısını bulur.
sizeof()	Dizideki toplam eleman sayısını bulur.
array_count_values()	Dizideki değerlere rastlanma sıklığını hesaplar.
array_product()	Dizideki sayısal değerlerin çarpımını bulur.
array_sum()	Dizideki sayısal değerlerin toplamını bulur.

```
<?php
   //Rastgele 1 ile 10 arasında 5 sayı diziye ekleniyor.
   $sayilar=array();
   for($i=1;$i<=5;$i++){
       array_unshift($sayilar,rand(1,10));
   }
   echo“<pre>”;
   print_r($sayilar);
   echo“Dizideki Eleman Sayısı: ”. count($sayilar). “<br>”;
   echo“Dizideki Sayısal Değerlerin Toplamı: ” . array_sum($sayilar).“<br>”;
   echo“Dizideki Sayısal Değerlerin çarpımı: ” . array_product($sayilar).“<br>”;
   echo“<pre>”;
   print_r(array_count_values($sayilar));
   ?>
```

Yukarıdaki örnekte 1 ile 10 arasında rastgele üretilen 5 sayı $sayilar dizine eklenip bu dizi üzerinde Tablo 3.4'teki hazır fonksiyonlarının kullanımı gösterilmiştir.

Dizilerde Arama Yapma

Dizi içinde bir değerin olup olmadığını kontrol etmenin farklı yolları vardır. Dizideki arama işlemleri için Tablo 3.5'te PHP tarafından sunulan fonksiyonlar kullanılır.

Tablo 3.5: *Arama Fonksiyonları*

Fonksiyon Adı	Açıklama
array_search()	Dizide belirtilen değeri arar, bulursa ilgili anahtarı geri döndürür.
in_array()	Dizide bir verinin olup olmadığını kontrol eder.
array_key_exists()	Belirtilen anahtar veya indisin dizide olup olmadığını kontrol eder.
array_key ()	Dizide belirtilen elemanı arar, bulursa ilgili anahtarları geri döndürür.

```
<?php
   //Rastgele 1 ile 10 arasında 10 sayı diziye ekleniyor.
   $sayilar=array();
   for($i=1;$i<=10;$i++){
       array_unshift($sayilar,rand(1,10));
   }
   echo“<pre>”;
   print_r($sayilar);
   $kontrol=array_search(2,$sayilar);
   if($kontrol!=FALSE){
       echo $sayilar[$kontrol];
   }
   ?>
```

Yukarıdaki örnekte $sayilar dizisinde 2 değeri **array_search()** fonksiyonu ile aranmaktadır. Rastgele üretilen dizide birden fazla 2 olursa dahi arama işleminde ilk bulduğu değerinin indisini geri döndürecek ve bunu ekrana yazacaktır. Metinsel ifadelerde büyük / küçük harf duyarlılığına göre işlem yapıldığı unutulmamalıdır.

```
  <?php
   //Rastgele 1 ile 10 arasında 10 sayı diziye ekleniyor.
   $sayilar=array();
   for($i=1;$i<=10;$i++){
       array_unshift($sayilar,rand(1,10));
   }
   echo“<pre>”;
   print_r($sayilar);
   if(in_array(2,$sayilar)){
       echo “2 sayısı dizide var.”;
   }
   else{
    echo “2 sayısı dizide yok.”;
   }
   ?>
```

Arama işlemi **in_array()** fonksiyonuyla da yapılabilir. Bu fonksiyon ile arama yapıldığında aranan eleman var ise geriye true, eleman yok ise geriye false değeri döndürür. Metinsel ifadelerde arama işlemleri büyük / küçük harflere duyarlı bir şekilde yapılır.

```
<?php
  $notalarim=array(“Matematik”=>85,“Fizik”=>100,“Kimya”=>85,“Tarih”=>95);
   if(array_key_exists(“Matematik”,$notalarim)){
       echo “Matematik dersinin notu var”;
   }
   else{
    echo “Matematik dersinin notu yok”;
   }  ?>
```

Değer yerine indis ismi üzerinde arama yapılacaksa **array_key_exists()** fonksiyonu kullanılır. Aranan indis var ise geriye true değeri döndürür. Yukarıdaki örnekte ise $notlarim dizisinin içinde Matematik notunun olup olmadığı **array_key_exists()** fonksiyonu ile kontrol edilmiştir.

```
<?php
   //Rastgele 1 ile 10 arasında 10 sayı diziye ekleniyor.
   $sayilar=array();
   for($i=1;$i<=10;$i++){
       array_unshift($sayilar,rand(1,10));
   }
   echo“<pre>”;
   print_r($sayilar);
   $kontrol=array_keys($sayilar,2);
   echo“<pre>”;
   print_r($kontrol);
   ?>
```

En gelişmiş arama yöntemi **array_keys()** fonksiyonu ile yapılır. Bu fonksiyon yardımı ile aranan elemanın dizi içinde bulunduğu tüm indisleri geri döndürür. Metinsel ifadelerin arama işlemleri büyük / küçük harflere duyarlı bir şekilde yapılır. Yukarıdaki örnekte 2 değerinin rastgele üretilen dizide olup olmadığını kontrol eder ve var ise hepsini indisleriyle birlikte ekrana yazar.

6. Uygulama

Alfabeden seçilecek rastgele harflerin oluşturacağı 29 harflik dizi üzerindeki işlemleri aşağıdaki adımları takip ederek gerçekleştiriniz.

1. Adım: index.php dosyasını server içinde oluşturunuz.

2. Adım: <?php ?> içinde rastgele harflerin olacağı $harf dizisi için gerekli kodları yazınız.

```
 <?php
   $harf=array();
$alfabe = ‘AaBbCcÇçDdEeFfGgĞğHhIıİiJjKkLlMmNnOoÖöPpRrSsŞşTtUuÜüVvYyZz’;
$harf_sayisi = mb_strlen($alfabe); //mb_strlen alfabede değişkenin karakter sayısını verir.
  for($i=1;$i<=58;$i++){
   $secilen_harf_konumu = rand(0, $harf_sayisi - 1); //0 ile 58 arasında rastgele sayı üret.
   $harfx=mb_substr($alfabe, $secilen_harf_konumu, 1); // mb_substr $alfabe
içinde istenilen yerden istenilen kadar harf almasını sağlar.
   array_unshift($harf,$harfx);
}
   ?>
```

3. Adım: A ve b harflerinin olup olmadığını kontrol eden kodları ekleyiniz.

```
$kontrol=array_search(“A”,$harf);
$kontrol1=array_search(“b”,$harf);
if($kontrol==TRUE){
    echo “A harfi dizi içinde var <br>”;
```

```
}
else{
    echo “A harfi dizi içinde yok<br>”;
}
if($kontrol1==TRUE){
    echo “b harfi dizi içinde var”;
}
else{
    echo “b harfi dizi içinde yok”;
}
```

4. Adım: ş ve Ş harflerinin olup olmadığını **in_array()** fonksiyonuyla kontrol eden kodları ekleyiniz.

```
if(in_array(“ş”,$harf)){
    echo “ş harfi dizi içinde var<br>”;
}
else{
    echo “ş harfi dizi içinde yok<br>”;
}
if(in_array(“Ş”,$harf)){
    echo “Ş harfi dizi içinde var<br>”;
}
else{
    echo “Ş harfi dizi içinde yok<br>”;
}
```

5. Adım: j ve K harfleri dizi içinde var ise hangi indislerde olduğunu bulan kodları ekleyiniz.

```
$arananharf=array_keys($harf,“j”);
echo“<pre>”;
print_r($arananharf);
$arananharf1=array_keys($harf,“K”);
echo“<pre>”;
print_r($arananharf1);
```

6. Adım: PHP sayfasını kaydediniz ve serverda bu sayfayı çalıştırarak görüntüleyiniz.

Sıra Sizde

Aşağıda verilen tabloyu dizi olarak tanımlayıp dizi içinde “Mehmet” ve “Ali”ye ait notların olup olmadığını kontrol eden kodları yazınız.

Öğrenciler					
Mehmet			Hasan		
Fizik=60	Matematik=70	Kimya=75	Fizik=90	Matematik=90	Kimya=75

Değerlendirme

Çalışmanız aşağıda yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

KONTROL LİSTESİ

Aşağıda listelenen ölçütlerden öğrencide gözlenen davranış için Evet, gözlenemeyen davranış için Hayır kutucuğunun altına (X) işareti koyarak işaretleyiniz.

Ölçütler	Evet	Hayır
1. Server içindeki dizi klasöründe diziarama.php dosyası oluşturuldu.		
2. $ogrenciler dizisi oluşturuldu.		
3. $ogrenciler dizisinde Mehmet için derslerin atamaları yapıldı.		
4. $ogrenciler dizisinde Mehmet için notların atamaları yapıldı.		
5. $ogrenciler dizisinde Hasan için derslerin ve notların atamaları yapıldı.		
6. $ogrenciler dizisinin içinde Mehmet'e ait notların olup olmadığını bulmak için gerekli kod blokları yazıldı.		
7. $ogrenciler dizisinin içinde Ali'ye ait notların olup olmadığını bulmak için gerekli kod blokları yazıldı.		
8. Programların çalışıp çalışmadığı kontrol edildi.		
9. Çalışmada iş sağlığı ve güvenliği kurallarına dikkat edildi.		
10. Zaman verimli kullanıldı.		

ÖLÇME VE DEĞERLENDİRME

A) Aşağıdaki cümlelerde parantezlerin içine yargılar doğru ise "D", yanlış ise "Y" yazınız.

1. ( ) Dizi tanımlanırken array() deyimi kullanılır.
2. ( ) Dizinin elemanlarına ulaşmak için köşeli parantez içine ({0}) indis değeri yazılmalıdır.
3. ( ) Dizinin elemanlarına ulaşmak için köşeli parantez içine (["Elazığ"]) indis değeri yazılmalıdır.
4. ( ) Dizinin boyutu **count()** fonksiyonuyla bulunur.
5. ( ) En gelişmiş arama yöntemi **in_array()** fonksiyonu ile yapılır.

B) Aşağıdaki soruları dikkatlice okuyarak doğru seçeneği işaretleyiniz.

6. Aşağıdaki tanımlamalardan hangisi doğrudur?

A) $sayim=$array(1,5,4,4);
B) $sayim=array(1;4;5;6);
C) $sayim=array(1,4,8,2);
D) $sayim=aray(1,5,2,4);
E) $sayim=array[1,5,2,4];

7. Aşağıdakilerden hangisi doğru bir dizi tanımlamasıdır?

A) $yol=array["Elazığ","Malatya","Kayseri"];
B) $yol=array("Elazığ"=>23,"Malatya"=>44,"Kayseri"=>38);
C) $yol=array(0="Elazığ",1="Malatya",2="Kayseri");
D) $yol=$array("Elazığ","Malatya","Kayseri");
E) $yol=aray("Elazığ","Malatya","Kayseri");

8. $liste=array("Ahmet","Veli","Ayşe","Vedat","Mustafa","Veysel","Derya","Hacer","Selim"); dizisinde count($liste) komutunun değeri aşağıdakilerden hangisidir?

A) 6 B) 7 C) 8 D) 9 E) Hata verir.

9.

```
<?php
$sayilar = array(1, 4, 8, 2 );
$toplam = 0;
for ( $index=0; $index < count($sayilar); $index++ )
{
 $toplam = $toplam + $sayilar[$index] ;
}
echo $toplam;
?>
```

Verilen bu kodlar çalıştırıldığında toplam değişkenin değeri aşağıdakilerden hangisidir?

A) 15 B) 13 C) 5 D) 1 4 8 2 E) 0

10.

```
<?php
$musteri=array(
      array(“adsoyad”=>“X Kişisi”,“sehir”=>“Elazığ”,“yas”=>“38”,
“meslek”=>“Doktor”),
      array(“adsoyad”=>“X Kişisi”,“sehir”=>“Elazığ”,“yas”=>“38”,
“meslek”=>“Öğretmen”),
);
    echo $musteri[1][“adsoyad”];
    ?>
```

Verilen bu kodlar çalıştırıldığında tarayıcıda görüntülenecek değer aşağıdakilerden hangisidir?

A) X Kişisi B) Elazığ C) Y Kişisi D) İstanbul E) Öğretmen

11.

```
<?php
$dizi = [  [1, 2],    [3, 4], ];
foreach ($dizi as list($a))
 {        echo “$a”;   }
?>
```

Verilen bu kodlar çalıştırıldığında tarayıcıda görüntülenecek değer aşağıdakilerden hangisidir?

A) 12 B) 13 C) 14 D) 23 E) 34

12. Aşağıdakilerden hangisi dizilere eleman ekleme fonksiyonudur?

A) array_pop() B) array_push() C) array_shift()
D) foreach() E) implode()

13. Aşağıdakilerden hangisi dizi sıralama fonksiyonudur?

A) array() B) array_shift() C) array_slice () D) sort () E) uniqe()

14. Aşağıdakilerden hangisi dizileri birleştirme fonksiyonudur?

A) array_merge () B) array_push() C) array_slice()
D) array_shift () E) current()

15. Aşağıdakilerden hangisi tanımlanan birden fazla dizide aynı <u>olmayan</u> elemanları bulmak için kullanılan fonksiyondur?

A) array_diff() B) array_intereset() C) array_rand()
D) array_reverse() E) array_unshift()

16. Aşağıdakilerden hangisi dizi göstericisinin üzerinde bulunan elemanı verip göstericiyi bir ilerleten fonksiyondur?

A) each() B) list() C) next() D) prev() E) reset()

17. Aşağıdakilerden hangisi bir değerin dizide olup olmadığını kontrol eder?

A) array() B) array_search() C) in_array() D) sort() E) unset()

4. ÖĞRENME BİRİMİ

FONKSİYONLAR
(Functions)



KONULAR

4.1. PHP'DE FONKSİYONLAR
4.2. FONKSİYON PARAMETRELERİ
4.3. DEĞER DÖNDÜREN FONKSİYONLAR
4.4. TARİH / SAAT FONKSİYONLARI
4.5. METİN FONKSİYONLARI
4.6. MATEMATİK FONKSİYONLARI

NELER ÖĞRENECEKSİNİZ?

- PHP programlama dilinde fonksiyon tanımlama
- PHP programlama dilinde fonksiyon çeşitlerini sıralama
- Fonksiyon parametrelerini kullanma
- Fonksiyon parametrelerini kendi içinde sıralama
- Değer döndüren fonksiyonları kavrama
- Değer döndüren fonksiyonları örneklendirme
- Tarih / saat fonksiyonlarını tanımlama
- Tarih / saat fonksiyonlarının arasındaki fark
- Metin fonksiyonlarının çeşitleri
- Metin fonksiyonlarının kullanımı

TEMEL KAVRAMLAR

fonksiyon, return, tarih / saat, sözel ifadeler, parametre, matematiksel işlemler.

Hazırlık Çalışmaları

1. **Programlama dillerinde fonksiyonlar nerelerde kullanılır?**
2. **Günlük hayatta kullanılan elektronik cihazların içindeki fonksiyonlar nelerdir?**

4.1. PHP'DE FONKSİYONLAR

PHP'de programların içindeki komutların ihtiyaç hâlinde birden fazla tekrarlanması gerekebilir. Bu gibi durumlarda komutların her defasında tekrar tekrar yazılmasına gerek yoktur. Bunun yerine tek bir program parçacığında (fonksiyon) yazılarak ana programda fonksiyon ismi yazılarak çağrılması beklenir. Diğer bir ifade ile fonksiyonlar bir defaya mahsus tanımlanır. Birçok yerde kullanılır. Bunun sonucunda hem bilgisayar derlemesi hızlanır hem de uzun uzuna yazılan komutların görsel olarak kısaltılması sağlanır. Aynı zamanda tekrarlanan komutlarda oluşabilecek hataların düzeltilme süresi azaltılır.



Görsel 4.1: PHP

Fonksiyonlar PHP'de programcı (kullanıcı) tanımlı ve hazır (yerleşik) fonksiyonlar olmak üzere ikiye ayrılır (Görsel 4.1).

4.1.1. Programcı (Kullanıcı) Tanımlı Fonksiyonlar

Programcı tanımlı fonksiyonlar, programcının kendisinin belirlediği öznel komutları barındıran fonksiyonlardır. Bu fonksiyonu tanımlamak için belirli bir dizilim kullanılması söz konusudur. Görsel 4.2'deki gibi fonksiyonlar tanımlanabilir.

```
function fonksiyon_ismi(){
    //Komutlar..
    //Komutlar..
    //Komutlar..
}
```

Görsel 4.2: Fonksiyon blok dizimi

4.1.2. Hazır (Yerleşik) Fonksiyonlar

Hazır fonksiyonlar PHP'de tanımlanmış olup ihtiyaç hâlinde kullanılır. Hazır fonksiyonlar sayısal, metin, tarih / saat, veri tabanı vs. gibi fonksiyonlardır. Bu fonksiyonlar değiştirilmeden kullanılır (Görsel 4.3).



Görsel 4.3: Hazır fonksiyonlar

4.2. FONKSİYON PARAMETRELERİ

Fonksiyonların içinde ihtiyaç olan değerin veya değerlerin ana programdan fonksiyona gönderilmesini sağlayan yapıya parametre denir. Fonksiyonlar parametre alıp almaması durumuna göre ikiye ayrılır (Görsel 4.4).



Görsel 4.4: Fonksiyon parametre örneği

4.2.1. Parametresiz Fonksiyonlar

Ana programdan gelebilecek herhangi bir değere ihtiyacı olmayan fonksiyonlardır. Fonksiyon ana programda her çağrıldığında aynı sonucu verir.

1. Uygulama

HTML body etiketinin içine parametresiz fonksiyon yöntemi kullanarak fonksiyon oluşturma işlemini aşağıdaki adımları takip ederek yapınız.

1. Adım: Ana programda fonksiyonun ismini tanımlayınız.

```
<?php
    ders_bilgileri();
```

2. Adım: Fonksiyonu ana programda çağırıp ders bilgilerini yazınız.

```
     function ders_bilgileri(){
       echo “Dersimizin Adı:Web Geliştirme<br>”;
       echo “Dersin Anlatıldığı Sınıf:12 BLŞ A<br>”;
       echo “Ders Öğretmenimiz:Özgür Çelik”;
    }
?>
```

Sıra Sizde

Sanal kart bilgilerini içinde barındıran tabloyu fonksiyon kullanarak yapınız (Görselin CSS kodlarını haricî olarak sayfanıza ekleyebilirsiniz.).

Sanal Kart Numarası	1111 1111 1111 0000
Son Kullanım Tarihi	09/27
CVV	2363

Görsel 4.5: Parametresiz fonksiyon

Değerlendirme

Çalışmalarınız diğer sayfada yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.

KONTROL LİSTESİ

Diğer sayfada listelenen ölçütlerden öğrencide gözlenen davranış için Evet, gözlenemeyen davranış için Hayır kutucuğunun altına (X) işareti koyarak işaretleyiniz.

Ölçütler	Evet	Hayır
1. Görsel 4.5 için gerekli olan HTML kodları yazıldı.		
2. CSS kodları haricî olarak oluşturuldu.		
3. PHP sayfasını çalıştırmak için gerekli kontroller yapıldı.		
4. Ana programda sanal kart bilgi fonksiyonu oluşturuldu.		
5. Oluşturulan fonksiyon ana programdan çağrıldı.		
6. Görseldeki sanal kart bilgileri fonksiyonda yazıldı.		

4.2.2. Parametreli Fonksiyonlar

Ana programdan gönderilen parametreler ile oluşturulan fonksiyonlara denir. Bu parametreler ile fonksiyonda işlemler yapılabilir. İşlemlerin sonuçları fonksiyondan ana programa gönderilerek değer dönüşü sağlanır.

2. Uygulama

Parametreli fonksiyon yöntemi kullanarak satranç tahtası oluşturma işlemini aşağıdaki adımları takip ederek yapınız.

1. Adım: Ana programda fonksiyonun ismini tanımlayınız ve parametrelerini giriniz.

```
<?php
    satrancTahtasiOlustur(8,8);
```

2. Adım: satrancTahtasiOlustur fonksiyonunu parametreleriyle beraber yazınız.

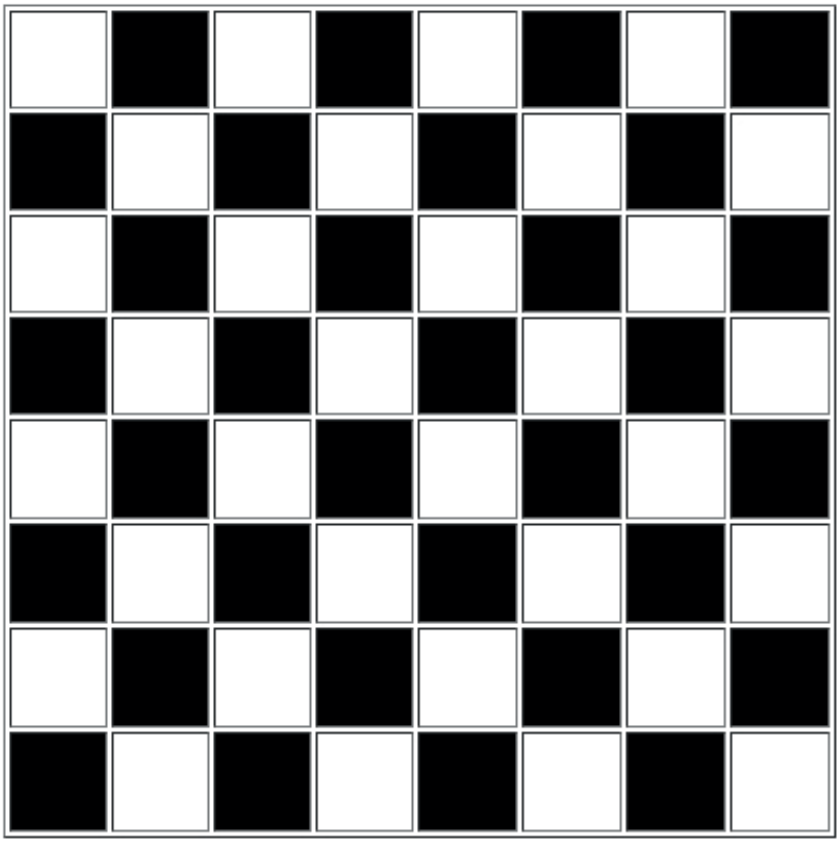
```
function satrancTahtasiOlustur($satir,$sutun)
        {
```

3. Adım: Oluşturulan fonksiyonun içine aşağıdaki kodları ekleyiniz.

```
echo "<table width=400 height=400 border=1>";
 for($i=1;$i<=$satir;$i++) //satır
     {
         echo '<tr>';
         for($j=1;$j<=$sutun;$j++) //sütun
             {
                 $kutu=$i+$j;
                 if($kutu%2==1)
                     { echo '<td bgcolor=black></td>'; }
                else
                     { echo '<td></td>';}
             }
         echo '</tr>';
     }
 echo '</table>';
}
?>
```

Bu uygulamada fonksiyon yapısı kullanılarak bir satranç tahtası elde edilir. Ana programda tanımlanan satır ve sütun sayıları fonksiyon parametresi olarak fonksiyona gönderilir.

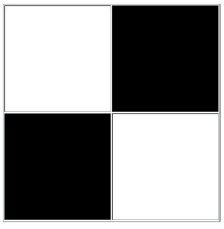
4. Adım: Gerekli HTML ve CSS kodlarını kullanarak Görsel 4.6'daki gibi satranç tahtası elde ettiğinizden emin olunuz.



Görsel 4.6: Satranç uygulaması

Önemli

Ana programdan fonksiyona değer gönderilirken bazen parametre bilgisini eksik göndermeler olabilir. Bunun önüne geçmek için aşağıdaki kodda olduğu gibi fonksiyonun parametrelerine varsayılan değerler atayınız. Böylelikle Görsel 4.7'deki gibi en az 2x2'lik bir tablo elde edeceksiniz (Görsel 4.7).



Görsel 4.7:Varsayılan değer

```
satrancTahtasiOlustur();
function satrancTahtasiOlustur($sat=2,$sut=2)
```

3. Uygulama

Döviz (para birimi) çevirme fonksiyonu oluşturmak için aşağıdaki adımları takip ediniz.

1. Adım: Ana programdan fonksiyonlara gönderilecek olan parametreyi değeriyle beraber tanımlayınız.

```
<?php
    $turkLirasi=3600;
```

2. Adım: Fonksiyonları ana programda tanımlayınız.

```
    dolarCevir($turkLirasi);
    euroCevir($turkLirasi);
    sterlinCevir($turkLirasi);
```

3. Adım: Parametre kullanarak ilgili çevirme işlemini gerçekleştiriniz.

```
    function dolarCevir($tutar){
       $dCeviri=$tutar/8;
       echo "Dolar Miktarı:$dCeviri $ <br>";
    }
    function euroCevir($tutar){
        $eCeviri=$tutar/10;
        echo "Euro Miktarı:$eCeviri € <br>";
     }
     function sterlinCevir($tutar){
        $sCeviri=$tutar/12;
        echo "Sterlin Miktarı:$sCeviri £  ";
     }
?>
```

4. Adım: Örnek kodun çıktısı olarak Görsel 4.8'de verilen çıktıyı elde ettiğinizden emin olunuz.

Dolar Miktarı:450 $
Euro Miktarı:360 €
Sterlin Miktarı:300 £

Görsel 4.8: Para birimi uygulaması

Önemli

Programda sonuç işlemleri fonksiyonda yapılmaktadır. Ana programa değer dönüşü olmamıştır.

4.3. DEĞER DÖNDÜREN FONKSİYONLAR

Ana programdan alınan parametreleri kullanarak çıkan sonuçları tekrar ana programa gönderen fonksiyonlara denir.

4. Uygulama

Döviz (para birimi) çevirme fonksiyonunda değerleri ana programa göndermek için aşağıdaki adımları takip ediniz.

1. Adım: Ana programdan fonksiyonlara gönderilecek olan parametreyi değeriyle beraber tanımlayınız.

```
<?php
    $turkLirasi=3600;
```

2. Adım: Fonksiyonları ana programda tanımlayınız.

```
    $dCeviri=dolarCevir($turkLirasi);
    $eCeviri=euroCevir($turkLirasi);
    $sCeviri=sterlinCevir($turkLirasi);
```

3. Adım: Parametre kullanarak ilgili çevirme işlemini gerçekleştiriniz. Sonucu return komutuyla ana programa gönderiniz.

```
    function dolarCevir($tutar){
        return ($tutar/8);
    }
    function euroCevir($tutar){
        return ($tutar/10);
    }
    function sterlinCevir($tutar){
        return ($tutar/12);
    }
```

4. Adım: Fonksiyondan dönen değerleri yazdırınız.

```
    echo "Dolar Miktarı:$dCeviri $ <br>";
    echo "Euro Miktarı:$eCeviri € <br>";
    echo "Sterlin Miktarı:$sCeviri £  ";          ?>
```

5. Adım: Uygulama çıktısı olarak sayfa 132 - Görsel 4.8'de verilen çıktıyı elde ettiğinizden emin olunuz.

Önemli

Her fonksiyonda return komutu bir kere kullanılır. Eğer fonksiyondan ana programa birden fazla değer gönderilecekse bu değer bir dizi yardımıyla taşınabilir.

5. Uygulama

Döviz (para birimi) çevirme fonksiyonunda değerleri ana programa dizi(array) kullanarak göndermek için aşağıdaki adımları takip ediniz.

1. Adım: Ana programdan fonksiyonlara gönderilecek olan parametreyi değeriyle beraber tanımlayınız.

```
<?php
    $turkLirasi=3600;
```

2. Adım: Fonksiyonu ana programda tanımlayınız. Fonksiyondan dönecek olan diziyi tutacak bir değişken tanımlayınız.

```
$dovizler=cevirIslemi($turkLirasi);
```

3. Adım: Parametre kullanarak ilgili çevirme işlemini gerçekleştiriniz. İşlem sonuçlarını dizinin elemanlarına atınız.

```
function cevirIslemi($tutar){
    $ceviri=array(3);
    $ceviri[0]=$tutar/8;  //Dolar
    $ceviri[1]=$tutar/10; //Euro
    $ceviri[2]=$tutar/12; //Sterlin
```

4. Adım: Return komutuyla diziyi ana programa gönderiniz.

```
    return $ceviri;         }
```

5. Adım: Fonksiyondan gönderilen diziyi yazdırınız.

```
echo "Dolar Miktarı:$dovizler[0] $ <br>";
echo "Euro Miktarı:$dovizler[1] € <br>";
echo "Sterlin Miktarı:$dovizler[2] £  ";
?>
```

6. Adım: Uygulama çıktısı olarak sayfa 132 - Görsel 4.8'de verilen çıktıyı elde ettiğinizden emin olunuz.

Önemli

Fonksiyondan ana programa üç değer gönderilmiştir. Bu değerler dizinin içine alınıp tek bir return komutuyla ana programa taşınmıştır. Böylelikle return komutuyla tek bir dizi içinde üç tane değer gönderilmiştir. Üç uygulamada da sayfa 132'de yer alan Görsel 4.8'deki çıktıyı elde ettiğinize dikkat ediniz.

6. Uygulama

Tablo 4.1: *Vücut Kütle İndeksi*

VKİ (kg/m²)	KİLO KATEGORİSİ
<18.5	Zayıf
18.5-24.9	Sağlıklı
25-29.9	Fazla Kilolu
30-34.9	I. Derece Obezite
35-39.9	II. Derece Obezite
≥40	III. Derece Obezite

Tablo 4.1'deki bilgileri kullanarak vücut kütle indeksi hesabını yapmak için aşağıdaki işlem adımlarını takip ediniz.

1. Adım: Ana programdan fonksiyonlara gönderilecek olan parametreyi değeriyle beraber tanımlayınız.

```
<?php
    $Boyunuz=180;
    $Kilonuz=80;
```

2. Adım: Fonksiyonu ana programda tanımlayınız.

```
    $Sonuc=vkiHesapla($Boyunuz,$Kilonuz);
```

3. Adım: Parametreler kullanarak vücut kütle indeksi hesabını yapınız. Sonucu diğer fonksiyona parametre olarak gönderiniz.

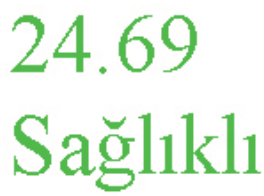
```
   function vkiHesapla($Boy,$Kilo){
        $Boy= $Boy /100;
        $BoyHesapla = $Boy * $Boy;
        $vki=round(($Kilo / $BoyHesapla),2);
        ekranayaz($vki);
    }
```

4. Adım: vkiHesapla fonksiyonundan gelen parametreyi kullanarak kilogram kategorisini yapınız. Sonucu web sayfasına basınız.

```
    function ekranayaz($Hesapla){
        if($Hesapla < “18.5”){
            echo “<div  style=’color:lime’>”
            .$Hesapla.“<span><br>Zayıf</span></div>”;
        }else if($Hesapla >= “18.5” and $Hesapla < “24.9”){
            echo “<div  style=‘color:limegreen’>”
            .$Hesapla.“<span><br>Sağlıklı</span></div>”;
        }else if($Hesapla >= “25” and $Hesapla < “29.9”){
            echo “<div  style=’color:orange’>”
            .$Hesapla.“<span><br>Fazla Kilolu</span></div>”;
        }else if($Hesapla >= “30” and $Hesapla < “34.9”){
            echo “<div  style=‘color:orange’>”
            .$Hesapla.“<span><br>I. Derece Obezite</span></div>”;
        }else if($Hesapla >= “35” and $Hesapla < “39.9”){
            echo “<div  style=‘color:red’>”
            .$Hesapla.“<span><br>II. Derece Obezite</span></div>”;
        }else if($Hesapla >= “40”){
            echo “<div  style=’color:maroon’>”
            .$Hesapla.“<span><br>III. Derece Obezite</span></div>”;
        }
    }

?>
```

5. Adım: Kodların çıktısı olarak Görsel 4.9'da verilen çıktıyı elde ettiğinizden emin olunuz.



Görsel 4.9: Vücut kütle indeksi

Önemli

Bu uygulamada iç içe fonksiyon yapısı kullanılmıştır. Bir fonksiyondan diğer fonksiyona bilgi aktarımı yapılmıştır.

4.4. TARİH / SAAT FONKSİYONLARI

PHP programlama dilinde hazır hâlde gelen zaman fonksiyonlarına tarih / saat fonksiyonları denir. PHP'de zaman 1 Ocak 1970 00:00:00 tarihinden başlar. Bu tarihten itibaren zaman saniye cinsinden tutulur. Bu duruma PHP'de **unix timestamp (zaman damgası)** denir.

date_default_timezone_set (): Bu fonksiyon ile yerel tarih / saat bilgisi güncellenebilir. PHP sayfalarında doğru tarih / saat bilgilerini alabilmek için ilk yapılması gereken sistem tarih / saat bilgisini Türkiye olarak ayarlamaktır. Bunun için aşağıdaki kodlardan herhangi biri sayfaya eklenmelidir.

```
date_default_timezone_set('Etc/GMT-3');
date_default_timezone_set('Europe/Istanbul');
```

time(): Sistemin tarih / saat bilgisine saniye cinsinden ulaşılmasını sağlar.

```
echo time()."<br>"; //çıktı: 1627028207
```

date(): date() fonksiyonu ile Unix timestamp (saniye) olarak verilen bilginin anlamlı hâle gelmesini sağlar. Tablo 4.2'deki parametreler kullanılarak istenilen biçimde tarih / saat bilgisine ulaşılabilir.

Tablo 4.2: Date() Parametreleri

Parametre	Açıklama	Çıktı
s	İki haneli saniye değeri	11
i	İki haneli dakika değeri	26
H	İki haneli 24 saat değeri	00 ile 23 arası
h	İki haneli 12 saat değeri	01 ile 12 arası
d	İki haneli gün değeri	01 ile 31 arası
D	Üç haneli gün kısaltması	Sun, Mon…
m	İki haneli ay değeri	01 ile 12 arası
M	Üç haneli ay kısaltması	Jan, Feb…

Parametre	Açıklama	Çıktı
Y	4 haneli yıl değeri	2021
j	Tek haneli gün değeri	4
n	Tek haneli ay değeri	1 ile 12 arası
y	İki haneli yılın son iki değeri	21
l	İngilizce haftanın günleri	Friday
F	İngilizce ay isimleri	July
z	Sayısal şekilde yılın günleri	0 ile 365 arası

Time() fonksiyonunu anlamlı hâle getirebilmek için yukarıdaki date() fonksiyonu ve parametreleri kullanılır.

7. Uygulama

time(), date() fonksiyonlarını ve parametrelerini kullanarak sistem tarih ve saat bilgilerini alma işleminde aşağıdaki adımları takip ediniz.

1. Adım: Sistem tarih / saatini Türkiye olarak ayarlayınız.

```
<?php
    date_default_timezone_set('Europe/Istanbul');
```

2. Adım: 1 Ocak 1970'ten günümüze kadar geçen saniye bilgisine ulaşabilmek için kodu yazınız.

```
    echo "Geçen Saniyeler:".time()."<br>";
```

3. Adım: Gün, ay, yıl, saat, dakika, saniye olarak almak için kodu yazınız.

```
    echo date('d.m.Y H:i:s',time())."<br>";
```

4. Adım: Sadece saat ve dakika bilgisini almak için kodu yazınız.

```
    echo 'Saat: ' . date('H.i')."<br>";
```

5. Adım: Ay ve gün bilgilerini metin olarak alabilmek için kodu yazınız.

```
        echo date('d F y l')."<br>";
?>
```

6. Adım: Kodların çıktısı olarak Görsel 4.10'da verilen çıktıyı elde ettiğinizden emin olunuz.

Geçen Saniyeler:1627041092
23.07.2021 14:51:32
Saat: 14.51
23 July 21 Friday

Görsel 4.10: Tarih / Saat uygulaması

Önemli

3. Adımda date() fonksiyonunun içinde yazılan time() fonksiyonu yazılmasa bile varsayılan olarak sistem tarih / saatini baz alarak biçimlendirme yapar. Diğer adımlarda olduğu gibi kodlar çalışır.

mktime(): Var olan tarih / saat bilgisini Unix timestamp (saniye) olarak almayı sağlar. Fonksiyonun kullanımı mktime (saat, dakika, saniye, ay, gün, yıl) şeklindedir.

```
mktime(23, 01, 04, 8, 5, 1990);//çıktı: 649886464
```

Önemli

Fonksiyon parametrelerinin kullanım sırasına dikkat edilmelidir. Tek haneli saat bilgilerinde önünde 0 (sıfır) kullanıldığında hata vermezken tek haneli ay ve gün bilgilerinin önünde 0 (sıfır) kullanıldığında hata verir.

8. Uygulama

time(), date() ve mktime() fonksiyonlarını ve parametrelerini kullanarak geçmiş zamandan bu zamana kadar geçen tarih / saat bilgilerini bulmak için işlem adımları takip ediniz.

1. Adım: Sistem tarih / saatini Türkiye olarak ayarlayınız.

```
<?php
    date_default_timezone_set('Europe/Istanbul');
```

2. Adım: Doğum tarihini saniyeye çeviriniz ve şimdiki zamanı tanımlayınız.

```
    $dogum_tarihi = mktime(0, 0, 0, 8, 5, 1990);
    $suan=time();
```

3. Adım: Şimdiki zaman ile doğum tarihi arasındaki farkı bulunuz.

```
   $gecen_zaman=$suan-$dogum_tarihi;
```

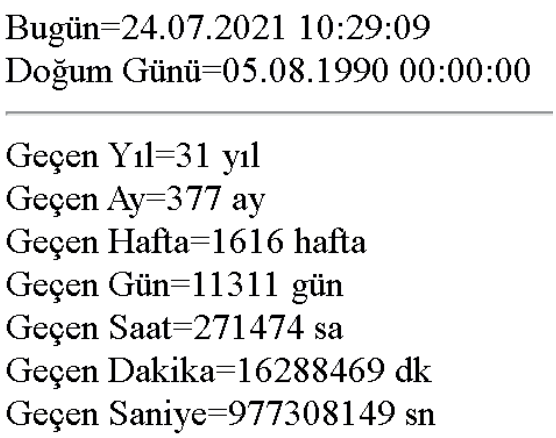
4. Adım: Şimdiki zaman ile doğum tarihini biçimli bir şekilde yazınız.

```
    echo "Bugün=".date('d.m.Y H:i:s',$suan)."<br>";
    echo "Doğum günü=".date('d.m.Y H:i:s',$dogum_tarihi)."<br><hr>";
```

5. Adım: Geçen zamanı saniye, dakika, saat, gün, ay, yıl şeklinde yazdırınız.

```
    echo "Geçen Yıl=".round($gecen_zaman/(60*60*24*365))." yıl<br>";
    echo "Geçen Ay=".round($gecen_zaman/(60*60*24*30))." ay<br>";
    echo "Geçen Hafta=".round($gecen_zaman/(60*60*24*7))." hafta<br>";
    echo "Geçen Gün=".round($gecen_zaman/(60*60*24))." gün<br>";
    echo "Geçen Saat=".round($gecen_zaman/(60*60))." sa<br>";
    echo "Geçen Dakika=".round(($gecen_zaman/60))." dk<br>";
    echo "Geçen Saniye=".$gecen_zaman." sn<br>";
?>
```

6. Adım: Yukarıdaki uygulama ile doğum tarihinden şimdiki zamana kadar geçen sürenin hesaplandığı Görsel 4.11'de verilen çıktıyı elde ettiğinizden emin olunuz.



Görsel 4.11: Tarih / Saat hesaplama uygulaması

Önemli

Round() fonksiyonu sayısal işlemlerde yuvarlama işlemi yapar.

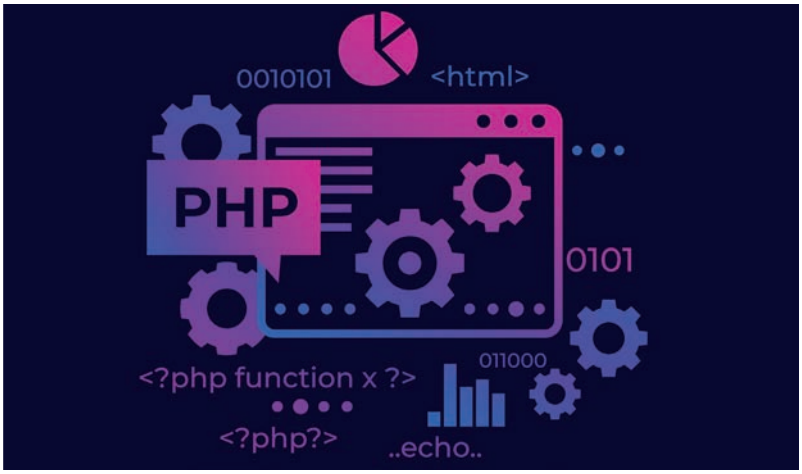
getdate(): Bu fonksiyon saniye (Unix) olarak verilen bilgiyi diziye çevirir. Parametre verilmez ise şimdiki zamanı diziye çevirir.

```
<?php
    $zaman=getdate(mktime(22, 04, 12, 8, 5, 1990));
    print_r($zaman);
     /* Çıktı:
            Array (
                [seconds] => 12
                [minutes] => 4
                [hours] => 22
                [mday] => 5
                [wday] => 0
                [mon] => 8
                [year] => 1990
                [yday] => 216
                [weekday] => Sunday
                [month] => August
                [0] => 649883052
            )
      */
?>
```

4.5. METİN FONKSİYONLARI

PHP'de metin (string), ifadelerde işlem yapmak için kullanılan hazır fonksiyonlara denir. Bu fonksiyonlar

ile metinlerin uzunluğu, metinler arası değer değiştirme, metin içindeki karakterin sayısı vs. gibi işlemler yapılabilir (Görsel 4.12).



Görsel 4.12: Metin fonksiyonları

mb_strlen($ifade): String ifadenin karakter uzunluğunu bulur.

```
<?php
    $cumle="12. Sınıf Web Programcılığı Dersi";
    $uzunluk=strlen($cumle);
    echo "$cumle ==> karakter uzunluğu = $uzunluk";
    /* Çıktı:
        12. Sınıf Web Programcılığı Dersi ==> karakter uzunluğu = 33
    */
?>
```

mb_strtolower($ifade): String ifadeyi küçük harfe çevirir.

```
<?php
    $cumle="12. Sınıf Web Programcılığı Dersi";
    $kucuk=mb_strtolower($cumle);
    echo "$cumle <br><br> $kucuk";
    /* Çıktı:
        12. sınıf web programcılığı dersi
    */
?>
```

mb_strtoupper($ifade): String ifadeyi büyük harfe çevirir.

```
<?php
    $cumle="12. Sınıf Web Programcılığı Dersi";
    $kucuk=mb_strtoupper($cumle);
    echo "$cumle <br><br> $kucuk";
    /* Çıktı:
        12. SINIF WEB PROGRAMCILIĞI DERSI
    */
?>
```

strrev($ifade): String ifadeyi tersinden yazar.

```
<?php
    $cumle="12.Sinif Web Programciligi Dersi";
    $ters=strrev($cumle);
    echo "$cumle <br><br> $ters";
    /* Çıktı:
        isreD igilicmargorP beW finiS .21
    */
?>
```

mb_str_split($ifade,$karakter): String ifadeyi girilen değer kadar bölüp diziye aktarır.

```
<?php
    $iban_numarasi="TR110001100000111110011101";
    $bol=mb_str_split($iban_numarasi,4);
    foreach($bol as $eleman)
    echo $eleman," ";
    /* Çıktı:
       TR11 0001 1000 0011 1110 0111 01
    */
?>
```

strpos($ifade,$aranan): String ifadenin içinde aranan değerin ilk başlangıç değerini döndürür.

```
<?php
    $tekerleme = 'Bir berber bir berbere:
    "Bre berber gel beraber,
    Bir berber dükkanı açalım" demiş.';
    echo strpos($tekerleme,'berber');
    /* Çıktı:
       4
    */
?>
```

strrpos($ifade,$aranan): String ifadenin içinde aranan değerin son değerini döndürür.

```
<?php
    $tekerleme = 'Bir berber bir berbere:
    "Bre berber gel beraber,
    Bir berber dükkanı açalım" demiş.';
    echo strpos($tekerleme,'berber');
    /* Çıktı:
       65
    */
?>
```

str_replace($kaynak,$hedef,$ifade): String ifadenin içindeki kaynak metnin hedef metin ile değişmesini sağlar.

```
<?php
   $ifade  = “Sınıfın en çalışkan öğrencisi Emir.”;
   $ifade = str_replace(“Emir”, “Melih”, $ifade);
   echo $ifade;
    /* Çıktı:
       Sınıfın en çalışkan öğrencisi Melih.
    */
?>
```

mb_substr($ifade,$baslangicDegeri,$karakterSayisi): String ifadenin belirlenen yerinden itibaren istenilen uzunlukta karakter alır.

```
<?php
    $hitabe = “Ey Türk Gençliği”;
    echo mb_substr($hitabe,0,2).“<br>”;
    echo mb_substr($hitabe,3,4).“<br>”;
    echo mb_substr($hitabe,8,strlen($hitabe)).“<br>”;
    /* Çıktı:
       Ey
       Türk
       Gençliği
    */
?>
```

nl2br($ifade): String ifadeleri alt alta yazmak için kullanılır. Satır sonlarına <br> tagı ekler.

```
<?php
    $mars = “İSTİKLÂL MARŞI

    Korkma, sönmez bu şafaklarda yüzen al sancak;
    Sönmeden yurdumun üstünde tüten en son ocak.
    O benim milletimin yıldızıdır, parlayacak;
    O benimdir, o benim milletimindir ancak.”;

    echo nl2br($mars);
    /* Çıktı:
      İSTİKLÂL MARŞI

      Korkma, sönmez bu şafaklarda yüzen al sancak;
      Sönmeden yurdumun üstünde tüten en son ocak.
      O benim milletimin yıldızıdır, parlayacak;
      O benimdir, o benim milletimindir ancak.
    */
?>
```

explode($karakter,$ifade): String ifadenin içinde belirlenen karaktere göre ifadeyi bölüp diziye aktaran fonksiyondur.

```
<?php
   $ifade="Teknoloji sayesinde herkesin kapısı, kocaman bir pencereye
dönüşebiliyor. Daniel Palmer";
   $dizi=explode(" ",$ifade);
   print_r($dizi);
   /* Array (
                [0] => Teknoloji
                [1] => sayesinde
                [2] => herkesin
                [3] => kapısı,
                [4] => kocaman
                [5] => bir
                [6] => pencereye
                [7] => dönüşebiliyor.
                [8] => Daniel
                [9] => Palmer
             )
   */
?>
```

implode($karakter,$ifade): String bir dizinin elemanlarını belirlenen karakter ile birleştiren fonksiyondur.

```
<?php
   $oyunlar=array("Saklambaç","Körebe","Köşe Kapmaca","Yağ Satarım Bal
                  Satarım");
   $ifade=implode("<br>",$oyunlar);
   echo "En çok sevilen sokak oyunları:<br>".$ifade;
   /* Çıktı:
      En çok sevilen sokak oyunları:
      Saklambaç
      Körebe
      Köşe Kapmaca
      Yağ Satarım Bal Satarım
   */
?>
```

9. Uygulama

Bir paragrafın içinde kaç tane kelime, cümle geçtiğini bulup cümleleri alt alta yazdıracak programda aşağıdaki işlem adımlarını takip ediniz.

1. Adım: Kelime ve cümle sayıları bulunacak paragrafı tanımlayınız.

```
<?php
    $paragraf="Önce bilgisayarın bir hesap makinesi olduğunu düşündük.
Sonra sayıları ASCII ile harflere nasıl çevireceğimizi öğrendik.
Bunun bir daktilo olduğunu düşündük. Sonra grafikleri keşfettik.
Bir televizyon olduğunu düşündük. World Wide Web(www) ile bunun bir broşür
olduğunu anladık.
Douglas Adams";
```

2. Adım: Paragrafın içinde geçen kelimelerin ve cümlelerin sayısını bulunuz.

```
    $kelime=explode(" ",$paragraf);
    $kelime_say=count($kelime)+1;
    $cumle=explode(".",$paragraf);
    $cumle_say=count($cumle);
```

3. Adım: Sayılarını bulduğunuz kelimeleri ve cümleleri ekrana yazdırınız.

```
    echo "Paragrafta $kelime_say kelime ve $cumle_say cümle vardır.";
    echo "<br><hr>";
```

4. Adım: Cümleleri alt alta yazdırınız.

```
    foreach($cumle as $sayac=>$deger)
      echo $sayac+1,"=>",$deger,"<br>";
?>
```

5. Adım: Programın çıktısı olarak Görsel 4.13'te verilen çıktıyı elde ettiğinizden emin olunuz.

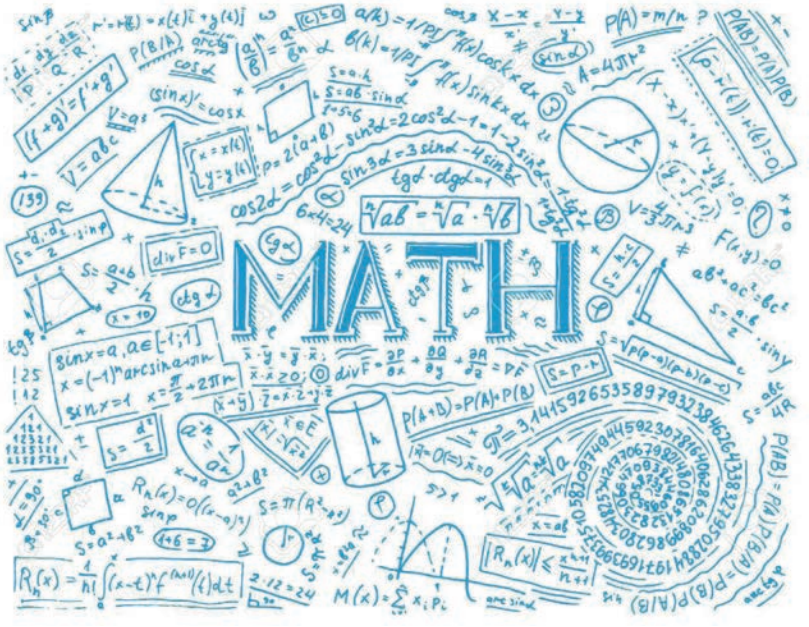
Paragrafta 38 kelime ve 7 cümle vardır.

1=>Önce bilgisayarın bir hesap makinesi olduğunu düşündük
2=> Sonra sayıları ASCII ile harflere nasıl çevireceğimizi öğrendik
3=> Bunun bir daktilo olduğunu düşündük
4=> Sonra grafikleri keşfettik
5=> Bir televizyon olduğunu düşündük
6=> World Wide Web(www) ile bunun bir broşür olduğunu anladık
7=> Douglas Adams

Görsel 4.13: Metin fonksiyonları

4.6. MATEMATİK FONKSİYONLARI

PHP'de hazır fonksiyonlar kullanılarak string(sözel) işlemler yapılabildiği gibi matematiksel fonksiyonlar kullanılarak matematiksel işlemler de kolaylıkla yapılabilir (Görsel 4.14).



Görsel 4.14: Matematiksel fonksiyonlar

abs($sayi): Parametre olarak gönderilen sayının mutlak değerini döndürür.

```
<?php
    echo abs(-6); //çıktı: 6
    echo abs(-15); //çıktı: 15
    echo abs(2); //çıktı: 2
?>
```

pow($taban,$kuvvet): Matematikteki üs alma işlemini yapan fonksiyondur.

```
<?php
   echo pow(2,5); // çıktı:32
   echo pow(3,-1); // çıktı:0.3
   echo pow(4,2); // çıktı:16
?>
```

sqrt($sayi): Gönderilen değerin karekökünü döndüren fonksiyondur.

```
<?php
   echo sqrt(16); // çıktı:4
   echo sqrt(25); // çıktı:5
   echo sqrt(144); // çıktı:12
?>
```

floor($sayi): Ondalık olarak gönderilen sayının noktadan sonrasını siler. Tam sayı kısmını döndürür.

```
<?php
   echo floor(2.5326); // çıktı:2
   echo floor(3.14562); // çıktı:3
   echo floor(5.895); // çıktı:5
?>
```

round($sayi,$ondalik): Noktalı olan sayıyı 5'ten küçükse aşağıya 5 ve 5'ten büyükse yukarıya yuvarlar. Ondalık değeri girilmezse en yakın tam sayıya yuvarlar.

Önemli

Noktadan sonra ilk rakamın indisi 0'dan (sıfır) başlamaktadır.

```
<?php
   echo round(15.7); //çıktı:16
   echo round(2.5326,3); // çıktı:2.533
   echo round(3.14562,1); // çıktı:3.1
   echo round(5.895,2); // çıktı:5.9
?>
```

max/min($sayi1, $sayi2, $sayi3, $sayi4): Parametre olarak gelen sayıların içinden en büyük olanı döndüren fonksiyondur. Parametre dizi olarak da gönderilebilir. Bu durumda dizinin içindeki en büyük değeri döndürebilir.

```
<?php
    echo max(30,58,103,2,-96); //çıktı:103
    echo min(30,58,103,2,-96); //çıktı:-96

    $dizi=array(30,58,103,2,-96);
    echo max($dizi); //çıktı:103
    echo min($dizi); //çıktı:-96
?>
```

rand($min,$max): Belirtilen aralıkta rastgele sayı üreten fonksiyondur.

```
<?php
    echo rand(30,58); //çıktı:43
    echo rand(-96,-1); //çıktı:-38
?>
```

10. Uygulama

Hazır matematiksel fonksiyonları kullanarak karmaşık bir matematik sorusunun çözümünü bulan programda aşağıdaki işlem adımlarını takip ediniz.

1. Adım: x ve y değişkenlerini tanımlayınız.

```
<?php
     // |x⁵-x⁴y²+7√y-35|   // 2|x+√xy+y|
    $x=2;
    $y=4;
```

2. Adım: Matematiksel fonksiyonlar kullanarak işlemi yapınız.

```
    $islem=abs(pow($x,5)-(pow($x,4)*pow($y,2))+7*sqrt($y)-35);
```

3. Adım: İşlem sonucunu yazdırınız.

```
      echo "Matematiksel işlemin sonucu:".$islem;
    // çıktı: Matematiksel işlemin sonucu:245
?>
```

11. Uygulama

Rastgele 10 sayı içinden sayı doğrusunda 0'a (sıfıra) en yakın ve en uzak değerleri PHP'de matematiksel fonksiyonlar kullanarak bulan programı aşağıdaki işlem adımlarını takip ederek gerçekleştiriniz.

1. Adım: -500 ile +500 arasında üretilen sayıları diziye aktarınız.

```
<?php

   for ($i=0; $i <10 ; $i++) {
     $dizi[$i]=rand(-500,500);
   }
```

2. Adım: Üretilen değerleri görmek için yazdırınız.

```
 foreach($dizi as $sayac=>$deger)
   echo $sayac+1,"=>",$deger,"<br>";
   echo "<hr>";
```

3. Adım: Dizinin içinde bulunan negatif değerleri pozitife çevirip başka bir diziye atınız.

```
for ($i=0; $i <10 ; $i++) {
      $mutlakdizi[$i]=abs($dizi[$i]);
    }
```

4. Adım: Pozitif dizi ile gerçek dizi değerlerini karşılaştırıp sıfıra yakın ve uzak olan değerleri bulunuz.

```
 for ($i=0; $i <10 ; $i++)
    {
      if(min($mutlakdizi)==abs($dizi[$i]))
      {
         echo "Sıfıra En Yakın Değer:".$dizi[$i]."<br>";
      }
      if(max($mutlakdizi)==abs($dizi[$i]))
      {
         echo "Sıfıra En Uzak Değer:".$dizi[$i]."<br>";
      }
    }
/*
      çıktı:
         1=>-315
         2=>-38
         3=>-402
         4=>207
```

```
        5=>53
        6=>301
        7=>437
        8=>-145
        9=>443
        10=>193
        Sıfıra En Yakın Değer:-38
        Sıfıra En Uzak Değer:443
    */
?>
```

Sıra Sizde

$2|x^3+\sqrt{xy}-y^2|$ denkleminin kodunu matematik fonksiyonları kullanarak oluşturunuz. X ve Y değişkenlerinin değerlerini Tablo 4.3'te belirtilen aralık içinde rastgele sayı üreterek veriniz.

Tablo 4.3: Rastgele Sayı Üretme

Rastgele Sayı Üretme Aralığı	X Değeri	Y Değeri	İşlem Sonuçları
-100 ile -20			
-20 ile 20			
20 ile 100			

Değerlendirme

Çalışmalarınız aşağıda yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.

KONTROL LİSTESİ

Diğer sayfada listelenen ölçütlerden öğrencide gözlenen davranış için Evet, gözlenemeyen davranış için Hayır kutucuğunun altına (X) işareti koyarak işaretleyiniz.

Ölçütler	Evet	Hayır
1. Kodların yazılacağı PHP sayfası oluşturuldu.		
2. X değerine Tablo 4.3'te belirtilen aralıkta sayı üretildi.		
3. Y değerine Tablo 4.3'te belirtilen aralıkta sayı üretildi.		
4. Denkleme göre PHP matematik fonksiyonları kullanıldı.		
5. İşlem sonuçlarının çıktısı ekrana yazıldı.		
6. Rastgele üretilen x ve y değerlerine göre Tablo 4.3 dolduruldu.		

ÖLÇME VE DEĞERLENDİRME

A) Aşağıdaki cümlelerde parantezlerin içine yargılar doğru ise (D), yanlış ise (Y) yazınız.

1. ( ) Fonksiyonun ana program tarafından her çağrıldığında tanımlanması gerekir.

2. ( ) Hazır fonksiyonlar, kullanıcı tarafından tasarlanan fonksiyonlardır.

3. ( ) Parametre, ana programdan fonksiyona gönderilen değerdir.

4. ( ) Return komutuyla aynı anda birden fazla değer ana programa döndürülebilir.

5. ( ) PHP programlama dilinde zaman 1 Ocak 1970 00:00:00 tarihinden itibaren saniye cinsinden tutulur.

B) Aşağıdaki soruları dikkatlice okuyarak doğru seçeneği işaretleyiniz.

6. Aşağıdakilerden hangisi Date() fonksiyonunda günleri üç haneli (Sun, Mon…) olarak gösterir?

A) s B) D C) m D) M E) d

7. 7.13:28:51 15.08.2021 şeklinde verilen tarihi aşağıdaki fonksiyonlardan hangisi saniyeye çevirir?

A) mktime() B) date() C) time()
D) getdate() E) date_default_timezone_set()

8. Aşağıdakilerden hangisi string ifadeyi tersine çeviren fonksiyondur?

A) mb_strtolower() B) mb_strlen() C) strpos()
D) strrev() E) str_replace()

9. Aşağıdakilerden hangisi string ifadenin içinde belirlenen karaktere göre ifadeyi bölüp diziye aktaran fonksiyondur?

A) implode() B) explode() C) strpos() D) strrev() E) mb_strlen()

10. Aşağıdakilerden hangisi ondalık sayıları tam sayılara çeviren programdır?

A) pow() B) sqrt() C) round() D) floor() E) abs()

5. ÖĞRENME BİRİMİ

FORM İŞLEMLERİ



KONULAR

5.1. HTML FORM ELEMANLARI

5.2. GET VE POST METOTLARI

5.3. GÜVENLİ FORM OLUŞTURMA

5.4. FORMDA DOSYA BİLEŞENLERİ KULLANMA

NELER ÖĞRENECEKSİNİZ?

- HTML form elemanları
- HTML form elemanları arasındaki kullanım farkları
- Form elemanlarını aktif olarak kullanma
- GET ve POST metotları arasındaki fark
- GET ve POST metotları ile filtreleme
- GET ve POST metotları ile etkileşimli uygulamalar
- Güvenli formu tanımlama
- Formlarda dosya bileşenini kullanma

TEMEL KAVRAMLAR

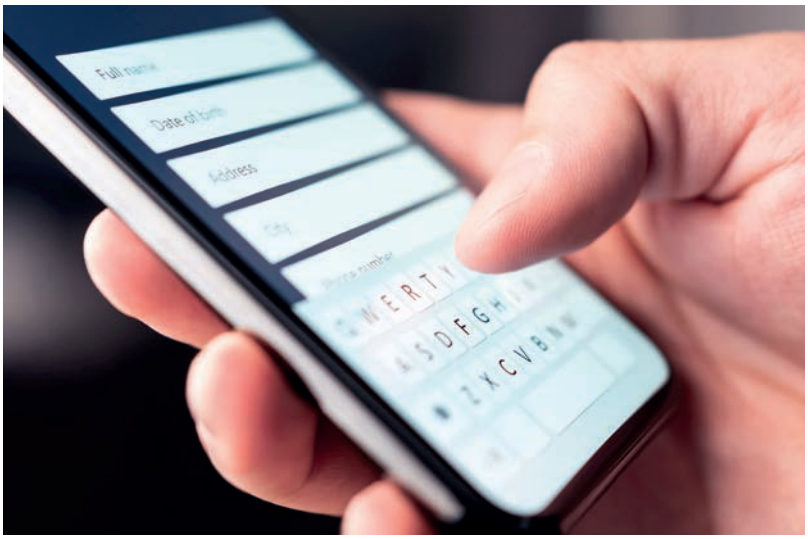
form, form elemanları, GET, POST, güvenli form oluşturma, dosya bileşenleri.

Hazırlık Çalışmaları

1. **Web sitelerinde anket ve iletişim formu gibi sayfalarda hangi form elemanları kullanılır?**
2. **Güvenli form oluşturmak için nelere dikkat edilmelidir?**

5.1. HTML FORM ELEMANLARI

PHP'de form elemanları kullanılarak kullanıcı ile etkileşimli (interaktif) web siteleri yapılabilir. Formlar sayesinde kullanıcıdan bilgi alınabilir, kullanıcıya bilgi verilebilir. Örneğin web sitelerindeki giriş sayfalarında, e-mail gönderimlerinde, bankacılık işlemlerinde, e-ticaret sektöründe, veri tabanı kayıt ve veri güncellemeleri gibi yerlerde formlar kullanılabilir (Görsel 5.1).



Görsel 5.1: Form elemanları

5.1.1. Form Etiketi

<form></form> etiketi ile form oluşturulur. Bu etiketin içinde metin, parola, onay kutusu, radyo butonu, seçim işlemi ve dosya seçimi gibi form elemanları form blokunun içine eklenir. Form etiketinde kullanılabilecek üç özellik vardır. Bu özellikler Tablo 5.1'de gösterilmiştir.

***Tablo 5.1:** Form Etiketi*

ÖZELLİK	GÖREVİ
NAME	Form ve form elemanlarına ulaşılırken kullanılan isimdir.
ACTION	Formdaki bilgilerin gönderileceği web sayfası dosya adı yazılır.
METHOD	Formdan bilgi gönderme yöntemidir.

```
<form name="iletisimFormu" action="…\iletisim.php" method="post" >
    /*  Form Elemanları */
</form>
```

Önemli

Bütün form elemanlarının isim (name) bilgisi bulunmalıdır. Formdan bilgi alışverişi olabilmesi için isim (name) özelliğinin değerleri bulunmalıdır.

Önemli

Form elemanlarının özelliklerini kullanırken daha görsel sayfalar oluşturmak için bootstrap kütüphanesinden yararlanılır. Sayfalarınızdaki <head></head> etiketinin içine aşağıdaki kodlar eklenmelidir.

```
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.0/
dist/css/bootstrap.min.css">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.
js"> </script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/
popper.min.js"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.
min.js"></script>
```

5.1.2. Input Etiketi

Genel olarak metin veri girişi için kullanılan form elemanıdır. Input etiketinin kapatma özelliği yoktur. Type özelliği kullanılarak aldığı değerler üzerinden farklı veri girişleri yapılabilir. Çok kullanılan type özellikleri şunlardır:

Text: Bir satır bilgi girişi için kullanılır (Görsel 5.2).

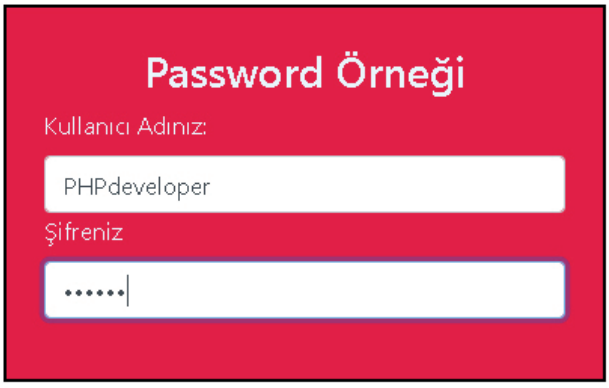
```
<body>
    <div style="height:250px;
    width:400px;
    color:crimson;
    border:1px solid crimson;
    padding:25px;
    background:aliceblue;">
        <form>
                <h3 style="text-align:center;">İnput Text Örneği</h3>
                <label>Adınız:</label>
                <input type="text" class="form-control" name="adi">
                <label>Soyadınız:</label>
                <input type="text" class="form-control" name="soyadi">
        </form>
    </div>
</body>
```



Görsel 5.2: Text özelliği

Password: Bilgi girişini gizler. Genelde yıldız ya da dolu daire karakterleri kullanılır (Görsel 5.3).

```
<body>
    <div style="height:250px;
    width:400px;
    color:aliceblue;
    border:2px solid black;
    padding:25px;
    background:crimson;">
        <form>
           <h3 style="text-align:center;">Password Örneği</h3>
            <label>Kullanıcı Adınız:</label>
            <input type="text" class="form-control" name="kAdi">
            <label>Şifreniz</label>
            <input type="password" class="form-control" name="sifre">
        </form>
    </div>
</body>
```



Görsel 5.3: Password özelliği

Radiobutton: Çoklu seçimlerde tek bir bilgiyi seçmek için kullanılır. Radio butonlarda isim (name) özelliğinin değerinin aynı olması gerekir (Görsel 5.4).

```
<div class="container p-3 my-3 bg-warning text-black">
     <h3> Askerlik Durumu </h3>
     <p>
        <input type="radio" name="onay" checked> Yapıldı<br>
        <input type="radio" name="onay"> Yapılmadı
     </p>
</div>
```



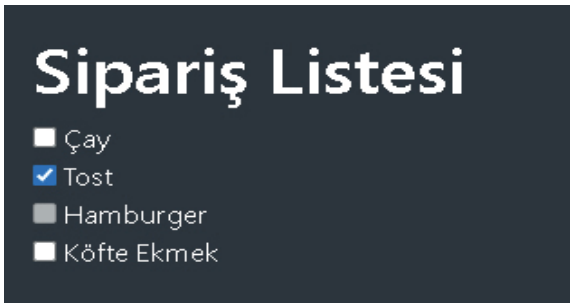
Görsel 5.4: Radio özelliği

Checkbox: Çoklu seçimlerde birden fazla bilgiyi seçmek için kullanılır. Checkbox'ta çoklu seçim için isim (name) değerlerinin farklı olması gerekir.

```
 <div class="container p-3 my-3 bg-dark text-white">
     <h1>Sipariş Listesi </h1>
     <p>
     <form>
        <input type="checkbox" name="onay1" >  Çay <br>
        <input type="checkbox" name="onay2" checked >  Tost <br>
        <input type="checkbox" name="onay3" disabled>  Hamburger <br>
        <input type="checkbox" name="onay4" >  Köfte Ekmek
     </form>
     </p>
 </div>
```

Önemli

Radio ve checkbox'ta checked özelliği seçili duruma getirir. Disabled özelliği de seçimi engellemek için kullanılabilir.

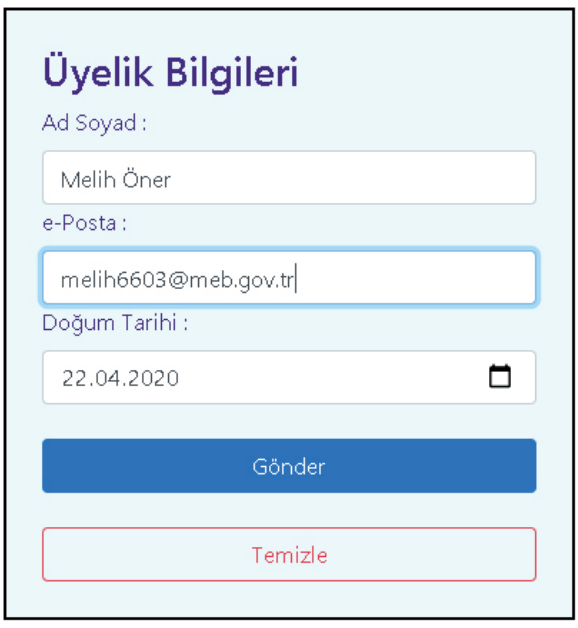


Görsel 5.4: Checkbox özelliği

Submit: Form etiketinin içine girilen bilgiyi action değerine gönderir (Görsel 5.5).

Reset: Ait olduğu formun içindeki bilgileri temizler (Görsel 5.5).

```
<body>
<div style="height:auto;
width:400px;
color:indigo;
border:2px solid black;
padding:25px;
background:azure;">
    <h3>Üyelik Bilgileri</h3>
    <form action="kayit.php">
        <label>Ad Soyad : </label>
        <input type="text" name="adSoyad" class="form-control">
        <label>e-Posta : </label>
        <input type="email" name="ePosta" class="form-control">
        <label>Doğum Tarihi : </label>
        <input type="date" name="dogumTarihi" class="form-control"><br>
       <input type="submit" value="Gönder" class="btn btn-primary btn-block"/><br>
        <input type="reset" value="Temizle" class="btn btn-danger btn-block"/>
    </form>
</div>
</body>
```



Görsel 5.5: Web sitelerinde form kullanımı

Yukarıdaki input özelliklerinin yanı sıra az bilinen ama ihtiyaç hâlinde kullanılabilecek diğer özellikler de Tablo 5.2'de gösterilmiştir.

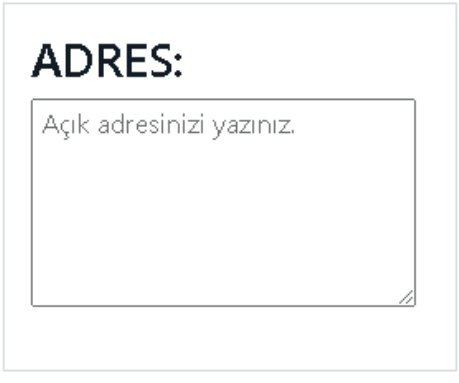
Tablo 5.2: Diğer İnput Etiketleri

TYPE	İŞLEVİ
COLOR	Renk seçimi yapılır.
DATE	Gün, ay, yıl seçimi yapılır.
MONTH	Ay ve yıl seçimi yapılır.
WEEK	Hafta ve yıl seçimi yapılır.
TIME	Saat ve dakika seçimi yapılır.
DATETIME-LOCAL	Tarih ve saat seçimi yapılır.
E-MAIL	E-mail bilgi girişini kontrol eder.
URL	URL bilgi girişini kontrol eder.
NUMBER	Belirlenen sayı aralığında seçim yaptırır.
RANGE	Bir bar üzerinde seçim yaptırır.
SEARCH	Arama çubuğu ekler.
FILE	Dosya seçimini isim olarak yapar. Dosya yükleme işlemi olmaz.

5.1.3. Textarea Etiketi

<textarea></textarea> etiketi bir metin elemanıdır. Sütun (cols) ve satır (rows) özellikleri kullanılarak yüksekliği ve genişliği ayarlanabilir. Genelde yorum veya adres bilgilerinde kullanılır (Görsel 5.6).

```
ADRES:<br>
<textarea name="metin"  cols="30" rows="10" placeholder="Açık adresinizi
yazınız."></textarea>
```

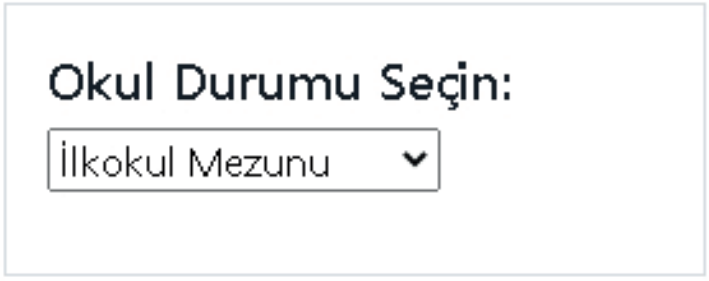


Görsel 5.6: Textarea özelliği

5.1.4. Select Etiketi

Birden fazla seçenekli durumlarda kullanılan etikettir. **<select></select>** şeklinde kullanılır. Seçenekler ise **<option></option>** etiketinin içine yazılır (Görsel 5.7).

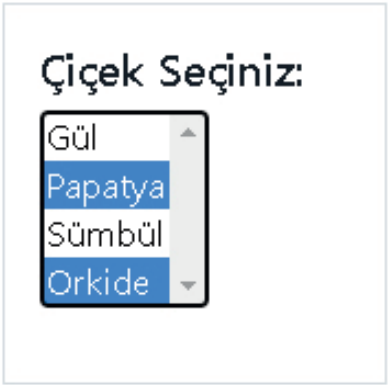
```
<div class="container p-3 my-3  border " style="width:275px">
     <h5>Okul Durumu Seçin:</h5>
         <p>
            <select name="okul" >
               <option value="ilkokul">İlkokul Mezunu</option>
               <option value="ortaokul">Ortaokul Mezunu</option>
               <option value="lise" >Lise Mezunu</option>
               <option value="üniversite">Üniversite Mezunu</option>
             </select>
           </p>
</div>
```



Görsel 5.7: Select özelliği

Seçimler içinde birden fazla seçim yapabilmek için multiple özelliği kullanılır (Görsel 5.8).

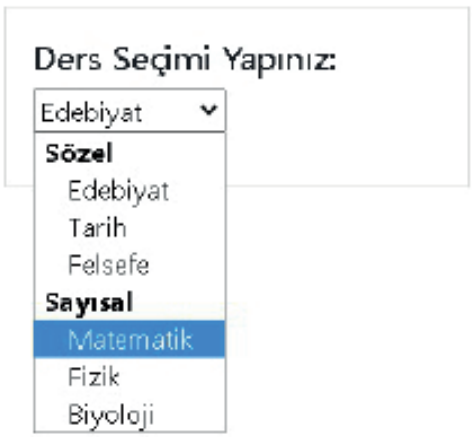
```
<div class="container p-3 my-3  border" style="width:175px">
    <h5>Çiçek Seçiniz:</h5>
    <p>
        <select name="cicekler" multiple >
                <option value="gul">Gül</option>
                <option value="pap">Papatya</option>
                <option value="sum">Sümbül</option>
                <option value="ork">Orkide</option>
        </select>
    </p>
</div>
```



Görsel 5.8: Multiple özelliği

Seçeneklerin grup hâlinde görünmesi için <optgroup></optgroup> blokunun içine yazılır (Görsel 5.9).

```
<div class="container p-3 my-3  border " style="width:275px">
     <h5>Ders Seçimi Yapınız:</h5>
     <p>
        <select name="dersler"  >
            <optgroup label="Sözel">
                <option value="edb">Edebiyat</option>
                <option value="tar">Tarih</option>
                <option value="fel">Felsefe</option>
            </optgroup>
            <optgroup label="Sayısal">
                <option value="mat">Matematik</option>
                <option value="fiz">Fizik</option>
                <option value="biy">Biyoloji</option>
            </optgroup>
          </select>
      </p>
 </div>
```



Görsel 5.9: Optgroup özelliği

1. Uygulama

Form elemanlarının özelliklerini kullanarak iş başvuru formunu aşağıdaki adımları takip ederek oluşturunuz.

1. Adım: Web sayfasının <head></head> kısmına bootstrap kütüphanelerini ekleyiniz.

HTML <head>

```
<head>
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.0/
dist/css/bootstrap.min.css">
```

```
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/
jquery.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/
umd/popper.min.js"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/
bootstrap.min.js"></script>
</head>
```

2. Adım: Web sayfasına CSS kodlarını haricî veya global olarak ekleyiniz.

CSS

```
.arkaplan{
            margin:auto;
            margin-top:20px;
            height:auto;
            width:500px;
            color:indigo;
            border:2px solid black;
            padding:25px;
            background:azure;
     }
```

3. Adım: Form elemanlarını <tablo> etiketinin içine uygun şekilde yazınız.

```
<html>
<body>
   <div class="arkaplan">
      <form action="basvuru.php" name="basvuru">
         <h2>Yazılım Uzmanı Bilgi Formu</h2>
            <h3>Kişisel Bilgiler</h3>
            <table cellpadding="5px">
               <tr>
                  <td>T.C. Kimlik</td>
                  <td><input type="text" name="tc" class="form-control">
                  </td>
               </tr>
               <tr>
                  <td>Ad Soyad</td>
                  <td><input type="text" name="adSoyad" class="form-
control"></td>
               </tr>
                <tr>
                   <td>Yaş</td>
                   <td><input type="number" name="yas" min="20" max="40"
class="form-control" ></td>
                </tr>
```

```
                <tr>
                   <td>Mail Adresi</td>
                   <td><input type="email" name="ePosta" class="form-
control" ></td>
                </tr>
                <tr>
                   <td>Fotoğraf Ekle</td>
                   <td><input type="file" name="profil"></td>
                </tr>
                </table>
                <h3>Mesleki Deneyim</h3>
             <table cellpadding="5px">
                <tr>
                   <td>Yazılıma Giriş Tarihi</td>
                   <td>
                      <input type="date" name="tarih" value="2015-01-01"
class="form-control">
                   </td>
                </tr>
                <tr>
                   <td>Yazılım Deyimi</td>
                   <td>
                      <label><input type="radio" name="deneyim"> 1 - 5 yıl
                      </label><br>
                      <label><input type="radio" name="deneyim"> 6 - 10 yıl
                      </label><br>
                   </td>
                </tr>
                <tr>
                   <td>Programlama Dilleri</td>
                   <td>
                      <select name="pDil"  multiple size="3" class="form-control">
                         <option>C#</option>
                         <option selected>PHP</option>
                         <option>Java</option>
                         <option>Python</option>
                         <option>JavaScript</option>
                         <option>C++</option>
                         <option>C</option>
                         <option>Pascal</option>
                      </select>
                   </td>
                </tr>
             </table>
```

```
            <h3>Form Onay</h3>
               <label><input type="checkbox" name="onay"> Yukarıdaki bilgilerin
doğruluğunu kabul ediyorum ve sözleşmeyi onaylıyorum.
              </label><br>
          <input type="submit" value="Kaydet"  class="btn btn-primary btn-block">
          <input type="reset" value="Temizle"  class="btn btn-danger btn-block">
       </form>
    </div>
</body>
</html>
```

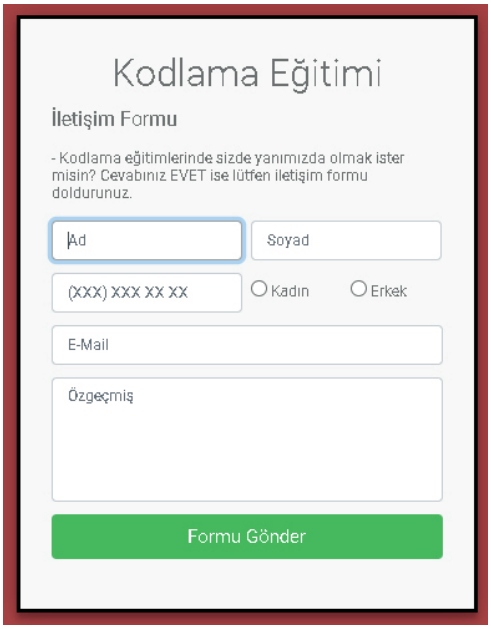
Yukarıdaki uygulamada yer alan kodlar eksiksiz bir şekilde yazıldığı zaman Görsel 5.10'daki gibi bir görsel ile karşılaşılır.



Görsel 5.10: Form uygulaması

Sıra Sizde

Form elemanlarını kullanarak Görsel 5.11'deki form örneğini oluşturunuz. Görseli oluşturmak için bootstrap kütüphanesinden faydalanabilirsiniz.



Görsel 5.11: Form örneği

Değerlendirme

Çalışmalarınız aşağıda yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.

KONTROL LİSTESİ

Aşağıda listelenen ölçütlerden öğrencide gözlenen davranış için Evet, gözlenemeyen davranış için Hayır kutucuğunun altına (X) işareti koyarak işaretleyiniz.

Ölçütler	Evet	Hayır
1. Görsel 5.11 için gerekli olan HTML kodları yazıldı.		
2. Sayfada bootstrap kütüphaneleri kullanıldı.		
3. CSS kodları haricî olarak oluşturuldu.		
4. Form etiketinde kullanılan özellikler eklendi.		
5. Formun gönderilme yöntemi POST olarak ayarlandı.		
6. Ad, soyad alanları için type="text" yazıldı.		
7. Cinsiyet alanında kullanılan radio butonlarının name özelliklerine aynı değerler verildi.		
8. Formu yönlendirmek için butona submit değeri yazıldı.		

5.2. GET VE POST METOTLARI

Form elemanları kullanılarak forma girilen bilgilerin kaybolmaması için bir sunucuya aktarılması gerekir. Aktarma işlemi yapılırken GET ve POST metotları kullanılır.

GET metodunda veriler URL'de (adres çubuğunda) görünür şekilde gönderilir. Gönderilen bilgiler adres çubuğunda gözüktüğü için güvenli değildir. POST metodu ise verileri adres çubuğunda göstermez. Bu sebeple form işlemlerinde kullanıcı adı ve şifre ya da daha önemli bilgiler sunuculara aktarılırken POST metodu kullanılmalıdır (Görsel 5.12).





Görsel 5.12: GET ve POST adres çubuğu

2. Uygulama

Form elemanları kullanarak dört işlem uygulamasını aşağıdaki adımları takip ederek yapınız. Form elemanlarından bilgileri GET metodunu kullanarak alınız. İşlem sonucunu bootstrap kütüphanesini kullanarak hesapla.php sayfasında yazdırınız.

1. Adım: Localhost'unuzda hesapMakinesi.php sayfasını oluşturunuz.

2. Adım: hesapMakine.php sayfasının <head></head> etiketin içine bootstrap kütüphanelerini ekleyiniz.

HTML <head>

```
<html>
<head>
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.0
/dist/css/bootstrap.min.css">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/
jquery.min.js"></script>
    <script  src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/
umd/popper.min.js"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/
bootstrap.min.js"></script>
```

3. Adım: hesapMakine.php sayfasına CSS kodlarını global olarak ekleyiniz.

CSS

```
 <style>
        div{
            margin:auto;
            margin-top:20px;
            height:auto;
            width:500px;
            color:forestgreen;
            border:2px solid black;
            padding:25px;
            background:honeydew;
        }
 </style>
</head>
```

4. Adım: hesapMakine.php sayfasının <body> etiketinin içine form elemanlarını ekleyiniz.

HTML <body>

```
<body>
<div>
        <form action="hesapla.php" name="hesapla" method="get">
                <h3 align="center">Hesap Makinesi Uygulaması</h3>
                <table cellpadding="5px" width="450px">
                    <tr>
                        <td>Birinci Sayı</td>
                        <td><input type="text" name="sayi1" class="form-
control" placeholder="1. Sayıyı Giriniz"></td>
                    </tr>
                    <tr>
                        <td>İkinci Sayı</td>
                        <td><input type="text"
  name="sayi2"  class="form-control" placeholder="2.Sayıyı Giriniz"></td>
```

```
                    </tr>
                    <tr>
                        <td>Yapılacak İşlem</td>
                        <td> <select name="secim" class="form-control">
                                <option value="+">TOPLA</option>
                                <option value="-">FARK</option>
                                <option value="*">ÇARPIM</option>
                                <option value="/">BÖLÜM</option>
                            </select>
                        </td>
                    </tr>
                    <tr>
                        <td colspan="2"><input
 type="submit" value="HESAPLA" style="width:215px" class="btn btn-
outline-primary">
                        <input type="reset"
 value="Temizle"  style="width:220px" class="btn btn-outline-danger"></td>
                    </tr>
                </table>

        </form>
    </div>
</body>
</html>
```

5. Adım: Localhost'unuzda hesapla.php sayfasını oluşturunuz.

6. Adım: hesapla.php sayfasının <head></head> etiketin içine bootstrap kütüphanelerini ekleyiniz.

HTML <head>

```
<html>
<head>
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.0/
dist/css/bootstrap.min.css">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/
jquery.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/
umd/popper.min.js"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/
bootstrap.min.js"></script>
```

7. Adım: hesapla.php sayfasına CSS kodlarını global olarak ekleyiniz.

CSS

```
 <style>
        div{
            margin:auto;
```

```
            margin-top:20px;
            height:auto;
            width:500px;
            color:white;
            border:2px solid black;
            padding:25px;
            background:red;
            text-align:center;
            margin-left:50px;
        }
        a{
            color:white;
        }
        a:hover{
            color:white;
            text-decoration:none;

        }
    </style>
</head>
```

8. Adım: hesapla.php sayfasının <body> etiketinin içine hesapMakinesi.php sayfasından gelen bilgileri GET metoduyla alıp işlem sonucunu yazdırınız.

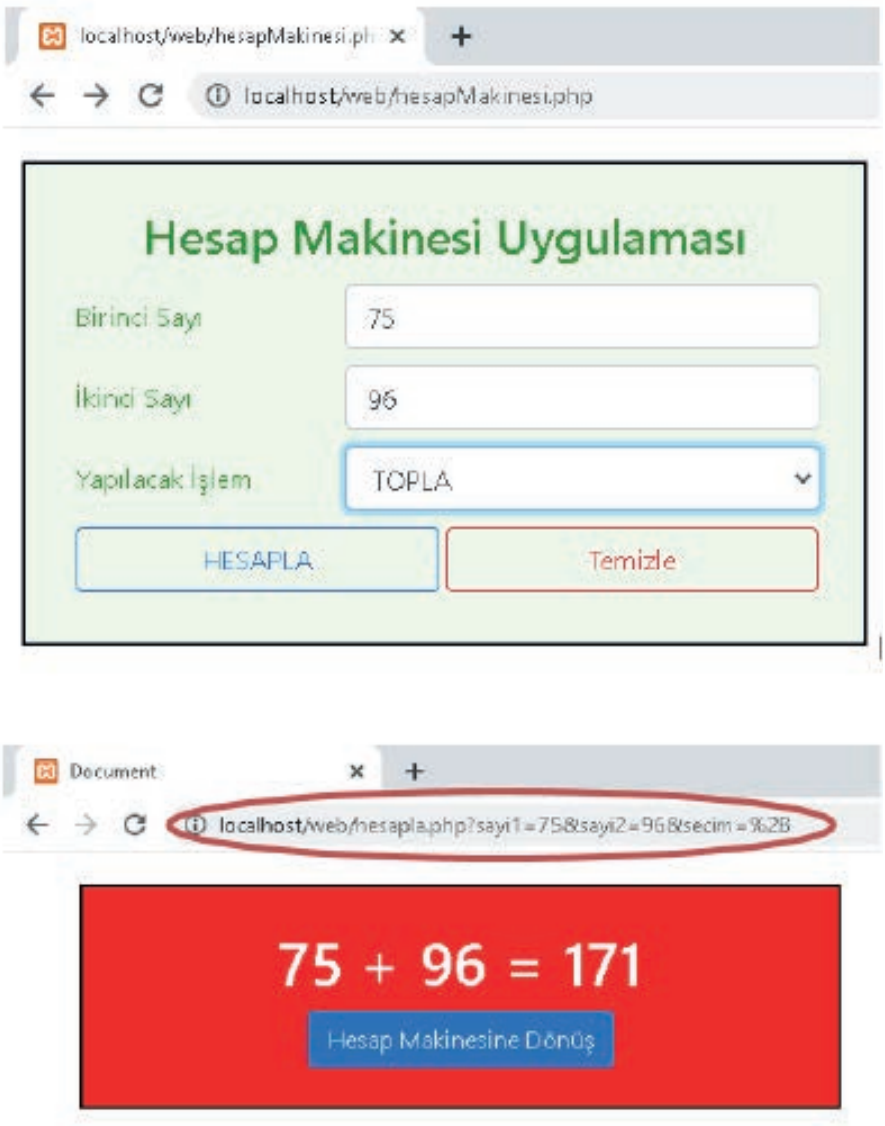
HTML <body>

```
<body>
<?php
    $sayi1=$_GET[“sayi1”];
    $sayi2=$_GET[“sayi2”];
    $secim=$_GET[“secim”];
    $sonuc=0;

    if(isset($secim))
    {
        if($secim == ‘+’)
        {
            $sonuc=$sayi1+$sayi2;
        }
        elseif($secim == ‘-’)
        {
            $sonuc=$sayi1-$sayi2;
        }
        elseif($secim == ‘*’)
        {
            $sonuc=$sayi1*$sayi2;
        }
```

```
        elseif($secim == '/')
        {
                $sonuc=$sayi1/$sayi2;
        }
    }
    echo "<div><h1 align='center'>$sayi1 $secim $sayi2 = $sonuc</h1>
    <button class='btn btn-primary'><a href='hesapMakinesi.php'>
Hesap.Makinesine.Dönüş</a></button></div>";
?>
</body>
</html>
```

Uygulamada kullanıcının bilgi girişini doğru yaptığı düşünüldüğünde programın çıktısı Görsel 5.13'teki gibi olur.



Görsel 5.13: Hesap makinesi

Önemli

Uygulamada matematiksel veya sayısal ifade yerine sözel ifadelerin girilmesi gibi kullanıcı kaynaklı hataların kontrolü yapılmamıştır. Bir sonraki hata yakalama işlemleri öğrenme biriminde bu gibi hataların önüne nasıl geçileceğiyle ilgili bilgilendirme yapılacaktır.

isset($değişken): Parametre değişkeninin tanımlı olup olmadığını kontrol eden fonksiyondur. Boolean değerlerini döndürür.

empty($değişken): Parametresindeki değişkenin içindeki değerinin varlığını kontrol eden fonksiyondur. Boolean değerlerini döndürür.

header(): Bir sayfadan başka sayfaya formun action özelliği kullanmadan belirli bir zamandan sonra geçişini sağlayan fonksiyondur.

3. Uygulama

Form elemanları yardımıyla basit bir yönetici giriş sayfasını aşağıdaki adımları takip ederek oluşturunuz. Bilgi girişi olacağı için bilgi gönderim yönetimini POST metodu olacak şekilde ayarlayınız.

1. Adım: Localhost'unuzda login.php sayfasını oluşturunuz.

2. Adım: login.php sayfasının <head></head> etiketinin içine bootstrap kütüphanelerini ekleyiniz.

HTML <head>

```
<html>
<head>
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.0/
dist/css/bootstrap.min.css">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/
jquery.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/
umd/popper.min.js"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/
bootstrap.min.js"></script>
```

3. Adım: login.php sayfasına CSS kodlarını global olarak ekleyiniz.

CSS

```
<style>
       div{
           margin:auto;
           margin-top:20px;
           height:auto;
           width:500px;
           color:white;
           border:2px solid black;
           padding:25px;
           background:lightskyblue;
       }   </style>   </head>
```

4. Adım: login.php sayfasının <body> etiketinin içine form elemanlarını ekleyiniz.

HTML <body>

```
<body>
<div>
    <form action="girisKontrol.php" name="Kontrol" method="post">
            <h3 align="center">Yönetim Sisteme Giriş</h3>
            <table cellpadding="5px" width="450px">
                <tr>
                    <td>Kullanıcı Adınız:</td>
                    <td><input type="text" name="kAdi" maxlength="11"
class="form-control" placeholder="Kullanıcı Adınızı Giriniz"></td>
                </tr>
                <tr>
                    <td>Şifreniz:</td>
                    <td><input type="password" name="sifre" class="form-
control" placeholder="Şifrenizi Giriniz"></td>
                </tr>
                <tr>
                    <td colspan="2"><input type="submit"
value="Giriş için tıklayınız"
style="width:450px;color:white;" class="btn btn-outline-primary"></td>
                </tr>
            </table>
    </form>
</div>
</body>
</html>
```

5. Adım: Localhost'unuzda girisKontrol.php sayfasını oluşturunuz.

6. Adım: girisKontrol.php sayfasına CSS kodlarını global olarak ekleyiniz.

CSS

```
<head>
 <style>
        div{
            margin:auto;
            margin-top:20px;
            height:auto;
            width:500px;
            color:red;
            border:2px solid red;
            padding:25px;
            text-align:center;
            margin-left:10px;
        }
```

```
        .bosGiris{
            border:2px solid orange;
            color:orange;
        }
        .dogruGiris{
            border:2px solid green;
            color:green;
        }
    </style>
</head>
```

7. Adım: girisKontrol.php sayfasının <body> etiketinin içine hesapMakinesi.php sayfasından gelen bilgileri GET metoduyla alıp işlem sonucunu yazdırınız.

HTML <body>

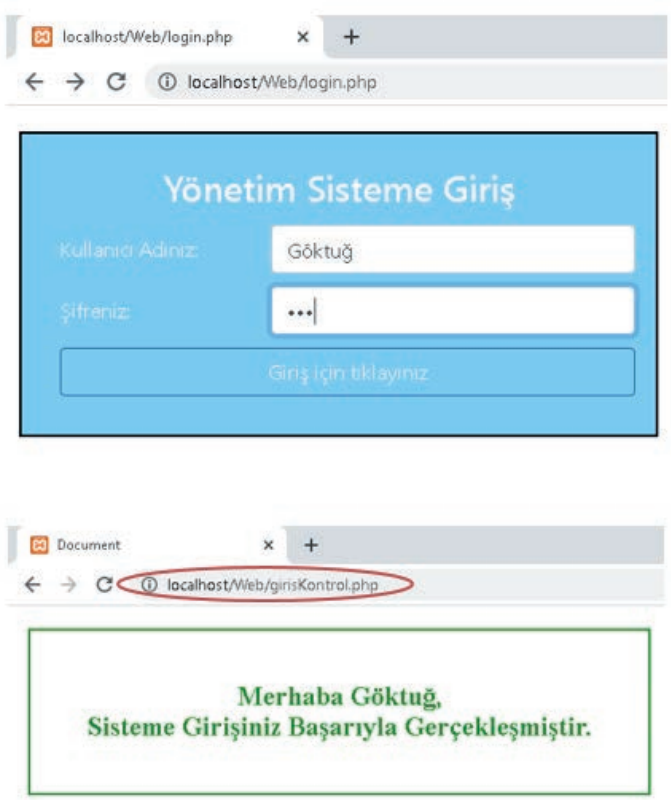
```
<body>
<?php
    $yonetim = array(
        array( “ad” => “Melih”, “sifre” => “123”),
        array( “ad” => “Hüma”, “sifre” => “456”),
        array( “ad” => “Afra”, “sifre” => “789”),
        array( “ad” => “Göktuğ”, “sifre” => “741”),
       );
    $kullanici=$_POST[‘kAdi’];
    $sifre=$_POST[‘sifre’];
    if(empty($kullanici)||empty($sifre))
    {
        echo “<div class=’bosGiris’><h2>Bilgileriniz eksiktir. <br> Giriş
sayfasına yönlendiriliyorsunuz.</h2></div>”;
        header(“Refresh: 5;url=login.php”);
    }
    else{
        for ($i=0; $i <count($yonetim); $i++)
         {
            if($yonetim[$i][“ad”]==$kullanici && $yonetim[$i][“sifre”]==$sifre )
            {
                $indis=$i;
                break;
            }
            else
            {
                $indis=-1;
            }
         }
         if($indis==-1)
         {
```

```
            echo “<div><h2>Sistemde Kaydınız bulunmamaktadır.<br>Giriş
sayfasına yönlendiriliyorsunuz.</h2> </div>”;
            header(“Refresh: 5; url=login.php”);
        }
        if($indis>-1)
        {
            echo “<div class=‘dogruGiris’><h2>Merhaba”.
 $yonetim[$indis][“ad”].“,<br>Sisteme Girişiniz Başarıyla Gerçekleşmiştir.
</h2></div>”;
        }
    }
?>
</body>
</html>
```

Uygulamada bilgi aktarımının gizli olması gerektiği için bilgi gönderimi POST metoduyla sağlanmıştır. Yönetici bilgileri çift boyutlu bir dizi yardımıyla tutulmuştur (Görsel 5.14).

Önemli

Statik olarak dizilerden alınan kullanıcı bilgilerinin MySQL Veri Tabanı Kullanımı konusu işlendikten sonra dinamik hâlde veri tabanından alınması daha uygun olacaktır.



Görsel 5.14: Yönetim sistemine giriş

4. Uygulama

Bir resim klasöründen sayfanızda görüntülemek istediğiniz resmin isminin form elemanları yardımıyla alınıp getirilmesini sağlayan uygulamayı aşağıdaki adımları takip ederek yazınız.

1. Adım: Localhost'unuzda resimGetir.php sayfasını oluşturunuz.

2. Adım: resimGetir.php sayfasının <head></head> etiketin içine bootstrap kütüphanelerini ekleyiniz.

HTML <head>

```
<head>
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.0/
dist/css/bootstrap.min.css">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/
jquery.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/
popper.min.js"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/
bootstrap.min.js"></script>
```

3. Adım: resimGetir.php sayfasına CSS kodlarını global olarak ekleyiniz.

CSS

```
<style>
        .arkaplan{
            margin:auto;
            margin-top:20px;
            height:500px;
            width:500px;
            color:honeydew;
            border:2px solid black;
            padding:25px;
            margin-left:10px;
            background:goldenrod
        }
        .formgrup{
            height:75px;
            width: auto;
        }
        #resimText{
            width:250px;
            float:left;
            margin-right:10px;
        }
        #buton{
            width:180px;
        }
```

```
        .resim{
            height:325px;
            width: auto;
            padding-left:150px;
            padding-top:150px;
            border:2px dotted honeydew;
            background-image:url('img/resim1.jpg');
            background-size:cover;
        }
    </style>
</head>
```

4. Adım: resimGetir.php sayfasının <body> etiketinin içine form elemanlarını ekleyip aynı sayfaya GET metoduyla gönderim sağlayınız.

HTML <body>

```
<body>
    <div class="arkaplan">
        <form action="resimGetir.php" name="resimGetir" method="get">
            <h1 align='center'>Resmi Getirelim</h1>
            <div class="formgrup">
               <input type="text" class="form-control" id="resimText"
name="resimAdi" placeholder="Resim adı giriniz">
                <button class="btn btn-outline-primary" id="buton">
Getir</button>
            </div>
```

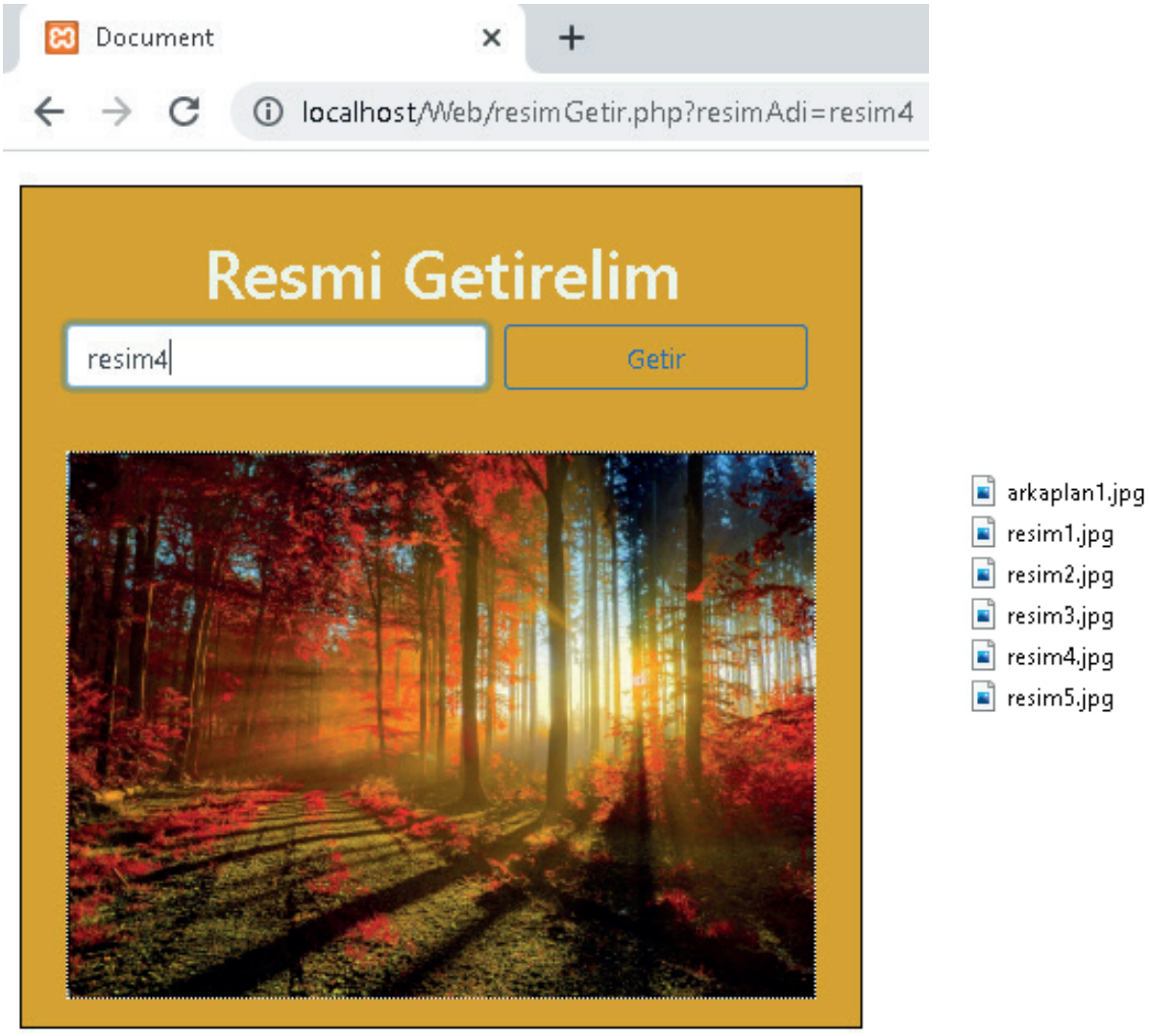
5. Adım: resimGetir.php sayfasının üzerinden GET metoduyla gelen resim bilgisini alıp ekrana yazdırınız.

HTML <body>

```
 <?php
                if(!empty($_GET['resimAdi']))
                {
                    $text=$_GET['resimAdi'];
                    $text="img/".$text.".jpg";
                    echo "<div class='resim' style='background-image:
url($text)'></div>";
                }
                else
                    echo "<div class='resim' style='background-image:
url(img/resim1.jpg)'></div>";

            ?>
        </form>
    </div>
</body>
</html>
```

Bu uygulama ile aynı sayfaya bilgi gönderimi yapıldığı görülür (Görsel 5.15).



Görsel 5.15: Resim getirme

5.3. GÜVENLİ FORM OLUŞTURMA

Kullanıcıdan gelen bilgileri filtrelemeden veri tabanına kaydetmek her zaman için bir tehdit unsurudur. Bir web sitesinin güvenliğini tehdit eden en büyük unsurlardan biri de web sitesinde bulunan formlardır. Kötü niyetli kişiler form elemanları üzerinden istenmeyen kodlar ve karakterler gönderebilir, veri tabanını bozabilir, bilgileri çalabilirler. Kullanıcıların veri tabanına erişimlerinin açık olduğu yerlerde kötü niyetli kişilerin var olacağı göz önünde bulundurularak birtakım önlemler alınmalıdır.

Kullanıcıdan gelen bilgilerin güvenliğini sağlamak için birkaç yöntem bulunur. PHP programlama dilinin içinde gelen hazır fonksiyonlar sayesinde bilgi güvenliği sağlanır.

addslashes(): Bu fonksiyon, kötü niyetli kişilerin zararlı içerik göndermesi için eklediği tırnak işaretlerini ters eğik çizgi işareti ile engeller. Bu sayede tırnak işaretlerinden doğabilecek olan açıklıklar kapanır.

```
echo addslashes(“ Hasan:Yolda Yücel’i gördüm, ‘Yarın sizin sınıfla maç
yapalım.’ dedi.”);
```

```
// çıktı: Hasan:Yolda Yücel\'i gördüm, \'Yarın sizin sınıfla maç yapalım.
\' dedi.
```

Kod çıktısında da görüldüğü üzere tek tırnak karakteri engellemiştir. Bilgi veri tabanına kaydedilirken de bu şekilde kaydedilmelidir.

htmlentities() ve htmlspecialchars(): HTML etiketlerinin çalışmasını engelleyen fonksiyonlardır. Form elemanına bir bilgi girilirken önüne ya da arkasına HTML etiketleri eklenmiş ise onları sözel ifade olarak algılar ve aynen yazılmasını sağlar.

```
echo htmlentities("<u><i><b>Bozkırın Tezenesi Neşet Ertaş</b></i></u>");

// çıktı: <u><i><b> Bozkırın Tezenesi Neşet Ertaş </b></i></u>

echo htmlspecialchars("<u><i><b> Bozkırın Tezenesi Neşet Ertaş </b></i></u>");

// çıktı: <u><i><b> Bozkırın Tezenesi Neşet Ertaş </b></i></u>
```

Eğer ki bu fonksiyonlar kullanılmadan form elemanına tırnak içindeki değer yazılsaydı içeriğin çıktısı ***Bozkırın Tezenesi Neşet Ertaş*** olacaktı. Fonksiyonlar sayesinde içerik bozulmadan veri tabanına aktarım sağlanır.

strip_tags(): HTML etiketlerini engellemek yerine silmek gerekirse bu fonksiyon kullanılır.

```
echo strip_tags("<u><i><b> Bozkırın Tezenesi Neşet Ertaş </b></i></u>");

// çıktı:  Bozkırın Tezenesi Neşet Ertaş
```

Bu fonksiyon yardımıyla veri tabanına bilgiler HTML etiketlerinden arınmış hâlde kaydedilir.

5. Uygulama

Güvenli form oluşturmak için PHP programlama dilinde hazır hâlde bulunan fonksiyonları kullanarak talep formu oluşturmak için aşağıdaki adımları takip ediniz.

1. Adım: talep.php sayfasını Localhost'ta oluşturunuz.

2. Adım: talep.php sayfasının <head></head> etiketin içine bootstrap kütüphanelerini ekleyiniz.

HTML <head>

```
<head>
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.0/
dist/css/bootstrap.min.css">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/
jquery.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/
popper.min.js"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/
bootstrap.min.js"></script>
```

3. Adım: talep.php sayfasına CSS kodlarını global olarak ekleyiniz.

CSS

```
<style>
        div{
            margin:auto;
            margin-top:5px;
            margin-left:5px;
            height:auto;
            width:410px;
            color:white;
            border:2px solid black;
            padding:25px;
            background:lightskyblue;
        }
        .btn{
          width:253px;
        }
    </style>
  </head>
```

4. Adım: talep.php sayfasının <body> etiketinin içine form elemanlarını ekleyip talepKontrol.php sayfasına POST metoduyla gönderim sağlayınız.

HTML <body>

```
<body>
  <div>
    <h3 align="center">Talep Formu</h3>
    <form action="talepKontrol.php" method="POST">
      <table align="center" cellpadding="5px">
        <tr>
          <td>Adı Soyadı</td>
          <td><input type="text" name="adisoyadi" class="form-control"
placeholder="Ad Soyad Giriniz" require></td>
        </tr>
        <tr>
          <td>Mesaj</td>
          <td><textarea name="mesaj" cols="30" rows="5" class="form-
control" placeholder="Talebinizi yazınız"></textarea></td>
        </tr>
        <tr>
          <td colspan="2" align="right"><input
 type="submit" value="Gönder" class="btn btn-primary"></td>
        </tr>
      </table>
    </form>
```

```
  </div>
</body>
</html>
```

5. Adım: Localhost'unuzda talepKontrol.php sayfasını oluşturunuz.

6. Adım: talepKontrol.php sayfasına CSS kodlarını global olarak ekleyiniz.

CSS

```
<head>
<style>
        div{
            margin:auto;
            margin-top:20px;
            height:auto;
            width:300px;
            color:white;
            border:2px solid black;
            padding:2px;
            text-align:center;
            margin-left:10px;
        }
        .guvensiz{
            background:red;
        }
        .guvenli{
            background:green;
        }
        table{
            color:white;
        }
    </style>
</head>
```

7. Adım: talep.php sayfasından POST metoduyla gelen bilgileri tablo içinde yazdırınız.

HTML <body>

```
<?php

$adisoyadi=$_POST['adisoyadi'];
$mesaj= $_POST['mesaj'];

echo "<div class='guvensiz'>
       <h4> Güvensiz Form</h4>

        <table>
            <tr>
```

```
                <td>Gönderen:</td>
                <td>$adisoyadi</td>
            </tr>
            <tr>
                <td>Mesaj:</td>
                <td>$mesaj</td>
            </tr>
        </table>
        </div>";
```

8. Adım: POST edilen bilgileri fonksiyon kullanarak güvenli hâle getiriniz.

HTML <body>

```
$gadisoyadi=htmlengelle($adisoyadi);
$gmesaj= guvenli_icerik($mesaj);

function htmlengelle($gelen){
    $giden = htmlspecialchars($gelen);
    $giden = htmlentities($giden);
    return $giden;
}

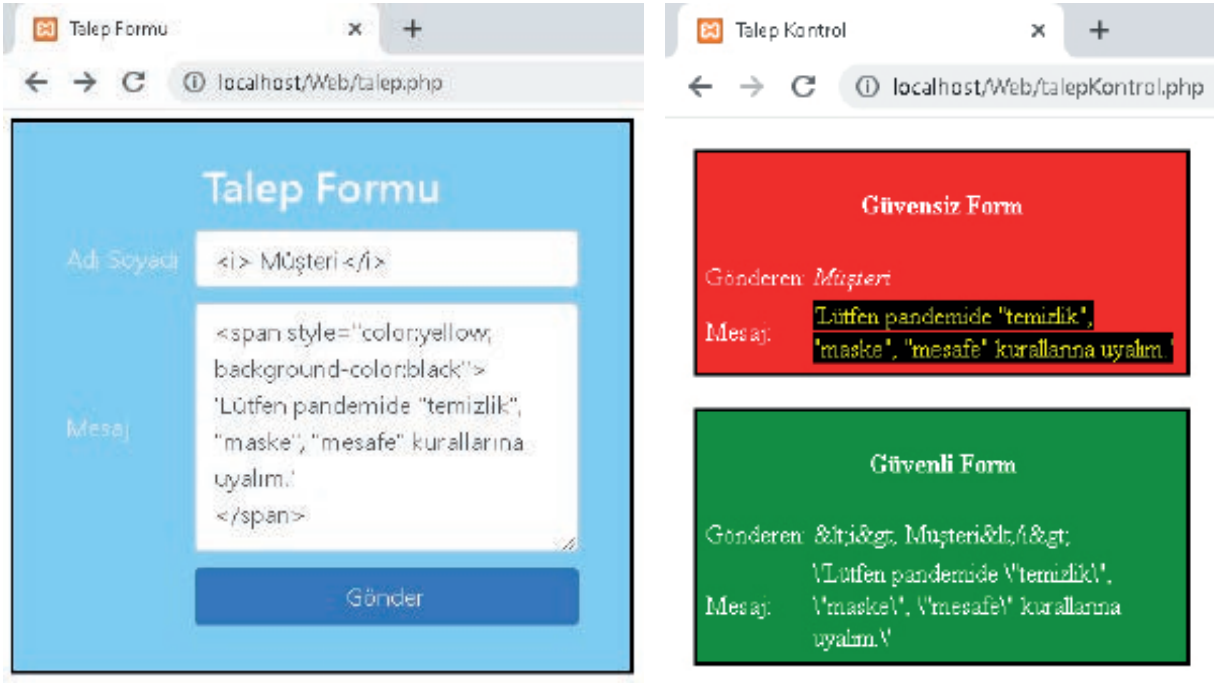
function guvenli_icerik($gelen){
    $giden = strip_tags($gelen);
    $giden = addslashes($giden);
    return $giden;
}
```

9. Adım: Güvenli hâle gelen bilgilerin çıktısını web sitesinden tablo şeklinde alınız.

HTML <body>

```
echo "<div class='guvenli'>
       <h4>Güvenli Form</h4>
        <table>
            <tr>
                <td>Gönderen:</td>
                <td>$gadisoyadi</td>
            </tr>
            <tr>
                <td>Mesaj:</td>
                <td>$gmesaj</td>
            </tr>
        </table>
        </div>";
?>
```

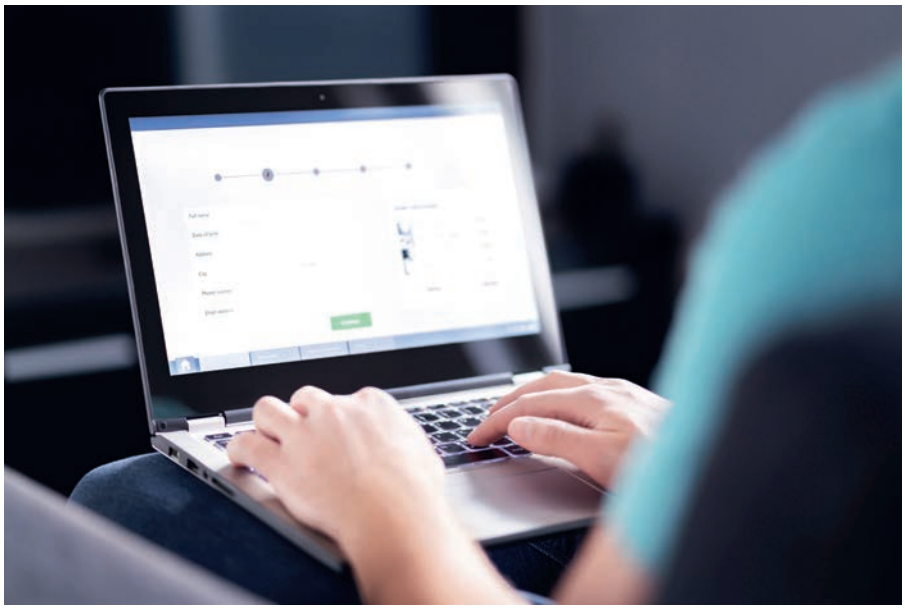
Uygulamada hem kullanıcı tanımlı hem de PHP hazır fonksiyonları kullanılarak forma girilen bilgiler güvenli hâle getirilmiştir (Görsel 5.16).



Görsel 5.16: Talep formu

5.4. FORMDA DOSYA BİLEŞENLERİNİ KULLANMA

PHP programlama dili kullanılarak oluşturulan web sitelerinde kullanıcıların en çok yaptığı işlerin başında dosya ve profil resmi yükleme (upload) gelir. PHP ile yükleme işlemleri kolaylıkla yapılır (Görsel 5.17).



Görsel 5.17: Dosya yükleme işlemi örneği

Dosya yüklemesi yapabilmek için aşağıdaki kodlar kullanılarak basit bir form oluşturulur.

```
<form action="dosyayukle.php" method="post" enctype="multipart/form-data">
    Yüklenecek dosyayı seçin:
    <input type="file" name="dosya" />
    <input type="submit" value="Yükle" />
</form>
```

Oluşturulan formun metodunun **POST** enctype özelliğinde **multipart/form-data** olmasına dikkat edilmelidir.

PHP tarafında dosya yükleme kodlarını içinde barındıracak bir **dosyayukle.php** adında localhostta bir sayfa oluşturulur. Bir sonraki sayfadaki kodlar yazılır.

```
<?php
     if ($_FILES["dosya"]) {
       echo "<pre>";
       print_r($_FILES);
       echo "</pre>";
     } else {
        echo "Lütfen bir dosya
seçin";
     }
?>
```

```
/* çıktı:
Array
(
[dosya] => Array
    (
 [name] => cuvallar.jpg
 [type] => image/jpeg
 [tmp_name] => C:\xampp\tmp\
php957.tmp
 [error] => 0
 [size] => 775920
    )
) */
```

$_FILES[] global değişken dizisi, formdan dosya yükleme işleminde kullanılır. Bu dizinin içinde yüklenen dosya ile ilgili bütün bilgiler tutulur. Dizi içindeki veriler kullanılarak dosya boyutu sınırlandırılabilir, dosya yüklenirken oluşabilecek hatalar bulunabilir ve yüklenmek istenilen dosya türü belirlenebilir.

```
<?php

if ($_FILES["dosya"]) {
  $islem = move_uploaded_file($_FILES["dosya"]["tmp_name"], $_FILES["dosya"]
["name"]);
  cho $islem ? "Dosya başarıyla yüklendi" : "Hata oluştu";
} else {
  echo "Lütfen bir dosya seçin";
}

?>
```

Yukarıdaki kod bloku ile oluşturulan PHP dosyasının **(dosyayukle.php)** bulduğu klasöre hiçbir tür, boyut ve hata kontrolü yapılmadan dosya yükleme işlemi yapılır.

PHP sayfasıyla yüklenen dosyaların genellikle farklı bir klasörde saklanması istenir. Bu durumda yuka-

rıdaki kod güncellenerek aşağıdaki kod yazılır.

```
<?php
  if ($_FILES[“dosya”]) {
  $dosyaAdi = “YüklenenBelgeler”;
  $dosyaYeri = __DIR__ . DIRECTORY_SEPARATOR . $dosyaAdi . DIRECTORY_SEPARATOR .
$_FILES[“dosya”][“name”];
  $islem = move_uploaded_file($_FILES[“dosya”][“tmp_name”], $dosyaYeri);
  echo $islem ? “Dosya başarıyla yüklendi” : “Hata oluştu”;
} else {
  echo “Lütfen bir dosya seçin”;
}
?>
```

$dosyaAdi değişkenine girilen değer ile yüklenecek dosyaların yeri belirlenir. $dosyaAdi = “YüklenenBelgeler”; Bu şekildeki bir tanımlamada **YüklenenBelgeler** dosyası PHP sayfasının bulunduğu klasör içinde aranır.

Önemli

Dosya yüklemesi yapılırken dosya adının yanlış yazılması hataya sebep olabilir. Aynı isimli dosyalar yüklendiğinde önceki dosya silinebilir ve yeni dosya eklenebilir. Bu durumları engellemek için dosyanın daha önceden yüklenip yüklenmediğinin kontrolü yapılmalıdır. Yüklü dosya varsa yeni yüklemeye izin verilmemelidir.

```
<?php
if ($_FILES[“dosya”]) {
  $dosyaAdi = “YüklenenBelgeler”;
  $dosyaYeri = __DIR__ . DIRECTORY_SEPARATOR . $dosyaAdi . DIRECTORY_SEPARATOR .
$_FILES[“dosya”][“name”];
  if ( file_exists($dosyaYeri) ) {
      echo “ Bu dosya daha önceden yüklenmişti”;
  } else {
      $islem = move_uploaded_file($_FILES[“dosya”][“tmp_name”], $dosyaYeri);
      echo $islem ? “Dosya başarıyla yüklendi” : “Hata oluştu”;
  }
} else {
  echo “Lütfen bir dosya seçin”;
}
?>
```

Bu kod bloku ile var olan bir dosyanın üstüne yükleme yapılmasına izin verilmemektedir.

Dosya yükleme işlemi yapılırken boyut sınırlaması ya da belirli uzantıya sahip dosyaların yüklenmesi istenir. Aşağıdaki kodlar yardımıyla istenilen kısıtlamalar dosya yükleme sayfasına eklenir.

```
<?php
if ($_FILES[“dosya”]) {
   $dosyaAdi = “YüklenenBelgeler”;
   $dosyaYeri = __DIR__ . DIRECTORY_SEPARATOR . $dosyaAdi . DIRECTORY_SEPARATOR .
   $_FILES[“dosya”][“name”];
   if ( file_exists($dosyaYeri) )
```

```
   {echo " Bu dosya daha önceden yüklenmişti";}
   else {
      if ($_FILES["dosya"]["size"]  > 1000000) {// 1MB üstü dosyalar engellenir.
        echo " Dosya boyutunuz 1MB'dan fazladır.";}
        else {
           $dosyaUzantisi = pathinfo($_FILES["dosya"]["name"],
           PATHINFO_EXTENSION);
           if ($dosyaUzantisi != "jpg" && $dosyaUzantisi != "png")
           { // Dosyanın uzantısı kontrol edilir.
              echo "Sadece jpg ve png uzantılı dosyalar yüklenebilir.";
           }
              else { $islem = move_uploaded_file($_FILES["dosya"]["tmp_name"],
                    $dosyaYeri);
                 echo $islem ? "Dosya başarıyla yüklendi" : "Hata oluştu"; }
        }
   }
}
else { echo "Lütfen bir dosya seçin"; }
?>
```

Dosya yükleme işleminde dosyanın boyutu sınırlandırılıp belirli uzantılara izin verilerek yükleme işleminin daha hatasız olması sağlanır. Böylelikle PHP sayfalarında yükleme işlemleri daha güvenli bir hâle gelir.

6. Uygulama

Dosya bileşeni kullanılarak basit bir profil resmi yükleme uygulamasını aşağıdaki adımları takip ederek yapınız.

1. Adım: dosyaYukleme.php sayfasını Localhost'ta oluşturunuz.

2. Adım: dosyaYukleme.php sayfasının <head></head> etiketin içine bootstrap kütüphanesini ekleyiniz.

HTML <head>

```
<head>
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.0/
dist/css/bootstrap.min.css">
```

3. Adım: dosyaYukleme.php sayfasına CSS kodlarını global olarak ekleyiniz.

CSS

```
<style>
        div{
            margin:auto;
            margin-top:20px;
            height:auto;
            width:400px;
            color:black;
            border:2px solid black;
            margin-left:10px;
            padding:5px;
            background:lime;
```

```
        }
    </style>
</head>
```

4. Adım: dosyaYukleme.php sayfasının <body> etiketinin içine form elemanlarını ekleyip dosyaKaydet.php sayfasına POST metoduyla gönderim sağlayınız.

HTML <body>

```
<body>
    <div>
        <form action="dosyaKaydet.php" method="post" enctype="multipart/
form-data">
            <h3 align="center">Dosya Yükleme Uygulaması</h3><hr>
            <p><input type="file" name="dosya"></p><hr>
            <p><input type="submit" value="Dosyayı Gönder" class="form-
control btn btn-light"></p>
        </form>
    </div>
</body>
</html>
```

5. Adım: Localhost'unuzda dosyaKaydet.php sayfasını oluşturunuz.

6. Adım: dosyaKaydet.php sayfasına CSS kodlarını global olarak ekleyiniz.

CSS

```
<head>
 <style>
        .zemin{
          width:250px;
          height:auto;
          border:2px solid black;
          background:lightcyan;
          padding:25px;
          text-align:center;
          color:black;
        }
        .uyari{
          background:gold;
        }
        .resim{
            width:200px;
            height:200px;
            background-size:cover;
        }
        img{
            width:200px;
            height:200px;
```

```
            box-shadow:5px 5px 5px gray;
        }
    </style>
</head>
```

7. Adım: Yüklenecek olan resim dosyalarının saklanacağı **yüklenenResimler** klasörünü dosyaYukleme.php ve dosyaKaydet.php sayfalarının bulunduğu dizinde oluşturunuz.

8. Adım: dosyaYukle.php sayfasından POST metoduyla gelen resim yolunu, boyutunu ve uzantısını kontrol ederek **yüklenenResimler** klasörüne yüklemesini yapan kodları yazınız.

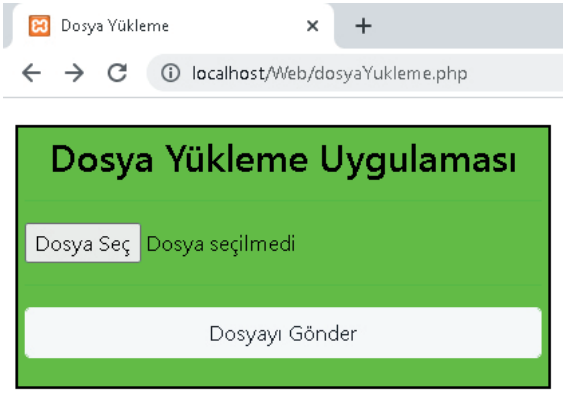
HTML <body>

```
<body>
<?php
if ($_FILES[“dosya”]) {
   $dosyaAdi = “YüklenenBelgeler”;
   $resimAdi=$dosyaAdi . DIRECTORY_SEPARATOR .$_FILES[“dosya”][“name”];
   $dosyaYeri = __DIR__ . DIRECTORY_SEPARATOR . $dosyaAdi . DIRECTORY_SEPARATOR .
 $_FILES[“dosya”][“name”];
   if ( file_exists($dosyaYeri) ) {
      echo “<div class=’zemin uyari’><h3>Bu dosya daha önceden yüklenmişti
      </h3></div>”;}
   else {
      if ($_FILES[“dosya”][“size”]  > 1000000) { // 1MB üstü dosyalar engellenir.
         echo “<div class=’zemin uyari’><h3>Dosya boyutunuz 1MB’dan fazladır.
         </h3></div>”;}
      else {
          $dosyaUzantisi = pathinfo($_FILES[“dosya”][“name”], PATHINFO_EXTENSION);
         if ($dosyaUzantisi != “jpg” && $dosyaUzantisi != “png”) { // Dosyanın
         uzantısı kontrol edilir.
            echo “<div class=‘zemin uyari’><h3>Sadece jpg ve png uzantılı
            dosyalar yüklenebilir.</h3></div>”;}
          else {
             $islem = move_uploaded_file($_FILES[“dosya”][“tmp_name”], $dosyaYeri);

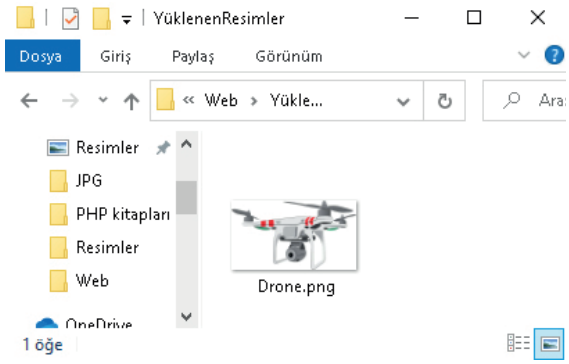
              if($islem){
                 echo “<div class=‘zemin’><h3> Dosya başarıyla yüklendi</h3><img
                src=$resimAdi><h3>Dosyanızın Yeri:<br>$resimAdi</h3></div>”;}
              else{
                 echo “Hata oluştu.”;
              }
          }
      }
   }
}
else { echo “Lütfen bir dosya seçin”; }
?>
</body>
</html>
```

Uygulama ile seçilen dosyanın belirlenen klasöre yükleme işlemi tamamlanır (Görsel 5.18).

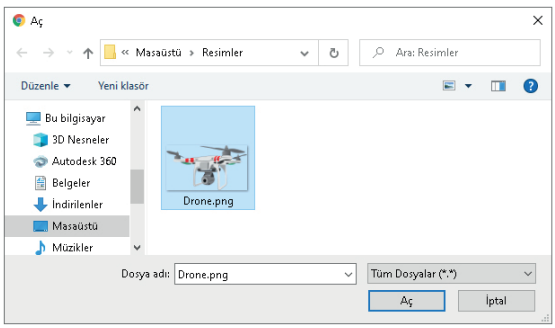
1. ADIM



4. ADIM



2. ADIM



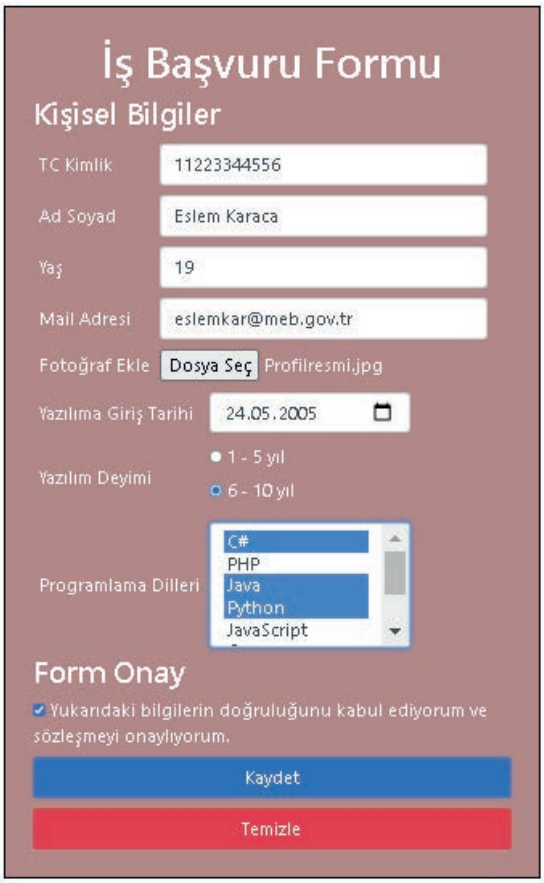
3. ADIM



Görsel 5.18: Dosya yükleme uygulaması

Sıra Sizde

HTML form elemanları kullanılarak oluşturulan iş başvuru formunun bilgilerini PHP kodları yardımıyla alınız (Görsel 5.19). Bilgileri alırken güvenli form oluşturma adımlarına dikkat ediniz. Dosya bileşenini kullanarak kişinin yükleyeceği fotoğrafı belirleyeceğiniz dosyaya kaydediniz.



Görsel 5.19: Başvuru formu uygulaması

Değerlendirme

Çalışmalarınız diğer sayfada yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.

KONTROL LİSTESİ

Aşağıda listelenen ölçütlerden öğrencide gözlenen davranış için EVET, gözlenemeyen davranış için HAYIR kutucuğunun altına (X) işareti koyarak işaretleyiniz.

Ölçütler	Evet	Hayır
1. Görsel 5.21 için gerekli HTML kodları yazıldı.		
2. Form elemanlarının görselliği için bootstrap kütüphaneleri sayfaya eklendi.		
3. CSS ile dosya oluşturulup ilgili CSS özellikleri yazıldı.		
4. Form etiketindeki name, method ve action özellikleri kontrol edildi.		
5. Form etiketine dosya yüklemede kullanılan enctype özelliği eklendi.		
6. Formdan POST metoduyla gönderilen bilgiler PHP sayfasında değişkenlere aktarıldı.		
7. Dosya bileşeni kullanılarak alınan fotoğraf bilgisi belirlenen klasöre kopyalandı.		
8. Formdan alınan bütün bilgiler tablo hâlinde sayfaya yazdırıldı.		

ÖLÇME VE DEĞERLENDİRME

A) Aşağıdaki cümlelerde parantezlerin içine yargılar doğru ise "D", yanlış ise "Y"yazınız.

1. ( ) POST metodunda formdaki veriler adres çubuğu üzerinden taşınır.

2. ( ) $_POST ve $_GET, PHP'de veri taşımada kullanılan global dizi değişkenleridir.

3. ( ) <input> etiketinin kapatma özelliği vardır.

4. ( ) Radio butonların name özellikleri aynı olmalıdır.

5. ( ) Multiple özelliği çoklu seçimlerde kullanılır.

B) Aşağıdaki soruları dikkatlice okuyarak doğru seçeneği işaretleyiniz.

6. Aşağıdakilerden hangisi HTML etiketlerini içerikten siler?

A) htmlspecialchars() B) mb_strlen() C) htmlentites()
D) addslashes() E) select

7. Aşağıdakilerden hangisi güvenli form oluşturmada kullanılan fonksiyonlardan biri değildir?

A) addslashes() B) htmlentites() C) mb_strlen()
D) htmlspecialchars() E) strip_tags()

8. Aşağıdakilerden hangisi <input type=""> etiketinin içine yazılan özellik değildir?

A) color B) time C) URL D) search E) strip_tags()

6. ÖĞRENME BİRİMİ

HATA YAKALAMA İŞLEMLERİ



KONULAR

6.1. PHP İSTİSNA İŞLEMLERİ
6.2. Try-Catch BLOKU
6.3. Finally BLOKU

NELER ÖĞRENECEKSİNİZ?

- PHP sayfalarında oluşabilecek hata çeşitleri
- İstisna (exception) kavramı
- İstisna çeşitleri
- Hazır hata yakalama fonksiyonları
- Try-catch blokunun kullanımı
- Try-catch bloku ile hatalı kodları yakalama
- Finally blokunun kullanımı
- Try-catch ile Finally blokları arasındaki farklar
- PHP sayfasındaki kodlarda hata olsa dahi çalışma yapma

TEMEL KAVRAMLAR

istisna, hata yakalama, Try-catch, Finally, kullanıcı hataları, yazılım hataları.



Hazırlık Çalışmaları

1. **PHP sayfalarında kullanıcıdan kaynaklı oluşabilecek hatalar neler olabilir?**
2. **Hata yakalama yöntemleri yazılım dünyasına ne gibi katkılar sağlar?**

6.1. PHP İSTİSNA İŞLEMLERİ

PHP dilinde veya herhangi bir başka programlama dilinde yazılan kodlar birçok sebepten hata verebilir. Bu hatalar genellikle mantıksal (**logical**), ölümcül (**fatal error**) veya uyarı (**warning**) hatalarıdır.

Programın kodlarından kaynaklı bir hata yoksa ama program yanlış sonuç üretiyorsa programda mantıksal (**logical)** düzeyde hata var demektir. Mantıksal hataları çözmek için programın amacına uygun bir algoritma geliştirilmesi gerekir. Aksi durumda program görünürde bir hata mesajı olmasa da sürekli yanlış sonuçlar üretecektir.

Programın kodlarında oluşan hatalar ölümcül (**fatal error**) düzeyde ise hata oluştuktan sonraki kodlar artık çalışmaz. Bu da görüntülenmek istenilen web sayfasının bir kısmının görüntüleneceği bir kısmının da hatadan kaynaklı görüntülenemeyeceği anlamına gelir. Ölümcül düzeydeki hatalar için istisna tanımlanmamaktadır. Ölümcül düzeydeki hatalar genellikle programın kodlarından kaynaklı oldukları için hatalı kodlar üzerinde düzenleme yapmadan PHP sayfası düzgün çalışmaz.

Programın kodlarında oluşan hatalar uyarı (**warning**) düzeyinde ise hata oluştuğunda PHP derleyicisi sayfaya hata mesajı verir ve derleyici diğer kodları çalıştırmaya devam eder. Uyarı düzeyindeki hatalar için istisnalar (**exception**) oluşturulabilir. Böylece hata mesajı kullanıcılara kontrollü bir şekilde gösterilir.

Aşağıda farklı düzeylerde hata veren örnek kodlar verilmiştir.

Mantıksal Hata

```
<?php
function cevreHesapla($kenar1,$kenar2)
{
     $sonuc= $kenar1*$kenar2;
     echo “Dikdörtgenin Çevresi=”.$sonuc;
}
 cevreHesapla(4,5);
?>
```

Yukarıdaki örnek kod çalıştırıldığında sayfada mantıksal (**logical**) düzeyde hata oluşur.

Ekran Görüntüsü

Dikdörtgenin Çevresi=20

Yapılan hesaplama yukarıdaki ekran görüntüsündeki gibi yanlış olacaktır çünkü `$sonuc` değişkeninin değeri “18” olması gerekirken uygulanan yanlış formül nedeniyle `$sonuc` değişkeninin değeri “20” çıkmıştır. Böylece mantıksal bir hata oluşmuştur. Bu hatayı ortadan kaldırmak için istisna oluşturmaya gerek yoktur. Kod üzerinde yanlış sonuca sebep olan formülün değiştirilmesi yeterlidir.

Mantık hatası ortadan kaldırılınca oluşacak kodlar ve ekran görüntüsü aşağıdaki gibidir.

```
<?php
function cevreHesapla($kenar1,$kenar2)
{
     $sonuc=2*($kenar1 + $kenar2);
     echo “Dikdörtgenin Çevresi=”.$sonuc;
}
```

```
cevreHesapla(4,5);   ?>
```

Ekran Görüntüsü

Dikdörtgenin Çevresi=18

Ölümcül Hata

```
<?php
echo “<u style=‘color:red’>Hata Yakalama İşlemleri (Fatal Error)</u>”;
function toplamaIslemi($sayi1, $sayi2)
 {
     $sonuc=$sayi1+$sayi2;
  return $sonuc;
}
echo “<br>İşlem Sonucu=”.topla(16,34);
?>
```

Yukarıdaki örnek kod çalıştırıldığında sayfa ölümcül (**fatal error**) düzeyde hata verir.

Ekran Görüntüsü

Hata Yakalama İşlemleri (Fatal Error)
Fatal error: **Uncaught Error: Call to undefined function topla()**
in C:\xampp\htdocs\HataAyiklama\ornek1.php:7
Stack trace: #0 {main} thrown in C:\xampp\htdocs\ornek\index.php **on line** 7

Sayfa yukarıdaki ekran görüntüsündeki gibi **tanımlanmamış fonksiyon çağrılması** (Call to undefined function) hatası verir. Ayrıca hata alınan yerden itibaren kodlar çalıştırılmaz. Bu hatayı ortadan kaldırmak için `topla(16,34)` fonksiyonu toplamaIslemi(16,34) şeklinde yazılmalıdır. Bu hata istisna oluşturularak ortadan kaldırılamaz.

Ölümcül hata ortadan kaldırılınca oluşacak kodlar ve ekran görüntüsü aşağıdaki gibidir.

```
<?php
echo “<u style=‘color:red’>Hata Yakalama İşlemleri (Fatal Error)</u>”;
function toplamaIslemi($sayi1, $sayi2)
    $sonuc=$sayi1+$sayi2;
  return $sonuc;
}
echo “<br>İşlem Sonucu=”.toplamaIslemi(16,34);    ?>
```

Ekran Görüntüsü

Hata Yakalama İşlemleri (Fatal Error)
İşlem Sonucu=50

Uyarı Düzeyinde Hata

```
<?php
function bolme($sayi1, $sayi2) {
     $sonuc=$sayi1/$sayi2;
  return $sonuc;
}
echo “İşlem Sonucu=”.bolme(52,0);
?>
```

Yukarıdaki örnek kod çalıştırıldığında sayfa uyarı (warning) düzeyinde bir hata verir.

Ekran Görüntüsü

```
Warning: Division by zero in C:\xampp\htdocs\ornek\index.php on line 3
İşlem Sonucu=INF
```

Sayfa yukarıdaki ekran görüntüsündeki gibi sıfıra bölünme (Division by zero) hatası verir. Ayrıca yapılan işlemin sonucu olarak ekranda tanımsız, sonsuz anlamına gelen "INF" ifadesi ile karşılaşılır. Bu hata istisna (excepition) oluşturularak kontrol altına alınır.

6.2. Try-Catch BLOKU

Birçok programlama dilinde olduğu gibi PHP dilinde de oluşabilecek hatalara karşı önlemler almak mümkündür. Hata oluşturabilecek kodlar `throw new Exception` kalıbı ile birlikte `Try` bloku içine yazılarak tanımlanır. `Catch` bloku ile de oluşan istisnalar tespit edilir.

Aşağıda istisna (exception) kavramı için kullanılan en genel kodlar ve exception kavramının program işleyişindeki kullanım şekline örnek verilmiştir.

```
<?php
  function islem($a,$b){
    try{
        if($b==0){
           throw new Exception("Hata ile ilgili verilecek mesaj" ,1);
    //Yukarıda bulunan kod satırındaki "1" sayısı istisnanın sıra numarasıdır.
       }
          return $a/$b;
       }catch(Exception $e) {
        echo $e->getMessage()."<br>";
        echo $e->getLine()."<br>";
        echo $e->getCode()."<br>";
        echo $e->getFile()."<br>";
      }
  }
 echo islem(52,0);?>
```

Try Bloku: Program içinde hata verme olasılığı olan kodların yazıldığı bölümdür. Try blokunda bir veya birden fazla `throw new Exception` kalıbı kullanılarak istisnalar tanımlanır.

```
try{
       if($b==0){
           throw new Exception("Hata ile ilgili verilecek mesaj" ,1);
         }
          return $a/$b;         }
```

Catch Bloku: Try blokunda oluşan istisnaların yakalandığı bölümdür. `Catch` blokunda `Exception` türünde bir sınıf tanımlanır ve bu sınıf yardımıyla oluşan istisnalar hakkında detaylı bilgi elde edilir. Eğer Try blokunda istisna meydana gelmez ise Catch blokunda bulunan kodlar çalışmaz.

```
catch(Exception $e) {
        echo $e->getMessage()."<br>";
        echo $e->getLine()."<br>";
        echo $e->getCode()."<br>";
        echo $e->getFile()."<br>";
      }
```

`Catch` blokunda `Exception` sınıfından türetilerek oluşturulan $e değişkeni ile oluşan istisna ile ilgili

birçok bilgi getirilebilir.

getMessage() Fonksiyonu: `throw new Exception` kalıbında belirtilen mesajı verir. `$e->getMessage()` fonksiyonu ile "Hata ile ilgili verilecek mesaj" şeklinde bir bilgi getirilir.

getLine() Fonksiyonu: Tanımlanan istisnanın kaçıncı kod satırında olduğunun bilgisini verir. Böylece kalabalık bir kod sayfası içinde istisnanın oluştuğu yer rahatlıkla bulunur. `$e->getLine()` fonksiyonu ile istisnanın oluştuğu kodun satır numarası yani "5" sayısı getirilir.

getCode() Fonksiyonu: Tanımlanan istisnanın kaçıncı istisna olduğunun bilgisini verir. Bu özellik aynı Try blokunda birden fazla istisna tanımlandığında işe yarar.

getFile() Fonksiyonu: Tanımlanan istisnanın hangi sayfada olduğu bilgisini verir. `$e->getFile()` fonksiyonu ile "C:\xampp\htdocs\ornek\index.php" şeklinde istisnanın oluştuğu sayfanın dosya yolu bilgisini getirir.

Önemli

Geniş kapsamlı web siteleri geliştirilirken kullanıcıların yaşadığı problemleri belirlemek için veri tabanında hata kayıtları kaydedilmelidir. getMessage(), getLine(), getCode(), getFile() fonksiyonları bu hata kayıtlarını tutmak için kullanılır.

1. Uygulama

İki sayıyı bölme işlemini hata yakalama yöntemlerini kullanarak Görsel 6.1'de görüldüğü gibi aşağıdaki adımları takip ederek yapınız.

4
Hatalı İşlem Yapıldı
2
Hatalı İşlem Yapıldı

Görsel 6.1: Sıfıra bölünme hatası

1. Adım: Proje klasörünüze "hataYakalama1.php" adında bir sayfa ekleyiniz.

2. Adım: Bölme işlemini yapmak için "bolme" adında iki parametreli fonksiyon oluşturunuz.

3. Adım: Sıfıra bölünme hatasını kontrol altına almak için Try bloku içinde istisna tanımlayınız ve Catch bloku içinde istisnaya ait mesajı verdiriniz.

4. Adım: Oluşturduğunuz "bolme" fonksiyonunu birçok kez çağırarak sonucu ekrana yazdırınız.

```
<?php
function bolme($sayi1, $sayi2) {
    $sonuc=0;
    try{

     if($sayi2==0){
      throw new Exception("<i style='color:red'>Hatalı İşlem Yapıldı</i>");
     //Yukarıda bulunan kod satırında istisna tanımlandı.
```

```
        }
            $sonuc=$sayi1/$sayi2;
            return $sonuc;
    }
     catch(Exception $e)
       {  //istisna tespit edildi.
        echo  $e->getMessage();
       }
}
echo "<i style='color:darkblue'>".bolme(16,4)."</i><br>";
echo "<i style='color:darkblue'>".bolme(34,0)."</i><br>";
echo "<i style='color:darkblue'>".bolme(6,3)."</i><br>";
echo "<i style='color:darkblue'>".bolme(25,0)."</i><br>";
?>
```

Önemli

Yukarıdaki örnek kod çalıştırıldığında sayfa uyarı (**warning**) düzeyinde iki adet hata verir fakat hata verecek kodlar Try bloku içine yazıldığı için hatalar kontrollü bir şekilde görüntülenir.

Önemli

Kodlar için exception tanımlanmasaydı sayfa sıfıra bölünme (Division by zero) hatası verecekti. Ayrıca yapılan işlemin sonucu ekranda tanımsız, sonsuz anlamına gelen "INF" ifadesi yazılacaktır. Bu hata istisna (excepition) oluşturularak kontrol altına alınır.

2. Uygulama

Fonksiyona satır ve sütun sayıları girerek tablo oluşturma işlemlerini hata yakalama yöntemleri kullanarak Görsel 6.2'de görüldüğü gibi aşağıdaki adımları takip ederek yapınız.

0,0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9
1,0	1,1	1,2	1,3	1,4	1,5	1,6	1,7	1,8	1,9
2,0	2,1	2,2	2,3	2,4	2,5	2,6	2,7	2,8	2,9
3,0	3,1	3,2	3,3	3,4	3,5	3,6	3,7	3,8	3,9
4,0	4,1	4,2	4,3	4,4	4,5	4,6	4,7	4,8	4,9

Lütfen satır ve sütun bilgisi için sadece sayı giriniz

Görsel 6.2: Yanlış veri türü

1. Adım: Proje klasörünüze "hataYakalama2.php" adında bir sayfa ekleyiniz.

2. Adım: Tablo oluşturma işlemini yapmak için "tablo" adında iki parametreli fonksiyon oluşturunuz.

3. Adım: Yanlış veri türünde giriş yapılmasını kontrol altına almak için Try bloku içinde istisna tanımlayınız ve Catch bloku içinde istisnaya ait mesajı verdiriniz.

4. Adım: Oluşturduğunuz "tablo" fonksiyonunu sırasıyla `tablo(5,10);` ve `tablo("dort", "beş");` şeklinde çağırarak sonucu ekrana yazdırınız.

```
<?php
function tablo($satir, $sutun) {
   try{
      if((is_numeric($satir) && is_numeric($sutun))) //değerler sayı ise
        {
           echo "<table style='border: solid 5px orange'>";
           for($i=0; $i<$satir; $i++) {
              echo "<tr>";
              for($j=0; $j<$sutun; $j++){
                echo "<td style='border: solid 5px gray'>".$i.",".$j."</td>";
                 }
                echo "</tr>";
               }
           echo "</table>";
        }
        else{
        throw new Exception("<i style='color:red; background-color:gray'>
          Lütfen satır ve sütun bilgisi için sadece sayı giriniz.</i>");
        }
    }catch(Exception $e) {
echo  $e->getMessage();
      }
}
tablo(5,10);
tablo("dort","beş");
?>
```

6.3. Finally BLOKU

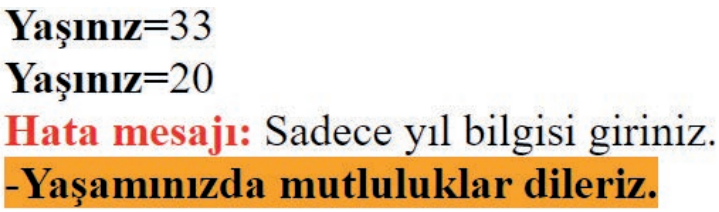
Try blokunda istisna meydana gelse de gelmese de finally blokundaki kodlar çalışır. Finally blokunu kullanmak zorunlu değildir, tercihe bağlıdır.

```
<?php
   try{
       //kodlar
   }catch(Exception $e){
       //kodlar
   }finally{
    echo "Buradaki kodlar her zaman çalışır.";
   }
?>
```

Finally blokundaki "Kodlar çalışmayı tamamladı." mesajı mutlaka ekranda görüntülenir.

3. Uygulama

Fonksiyona yıl bilgisi girerek yaş hesaplama işlemini hata yakalama yöntemleri kullanarak Görsel 6.3'te görüldüğü gibi aşağıdaki adımları takip ederek yapınız.



Görsel 6.3: Yaş hesaplama

1. Adım: Proje klasörünüze "hataYakalama3.php" adında bir sayfa ekleyiniz.

2. Adım: Yaş hesaplama işlemlerini yapmak için "yasHesapla" adında tek parametreli fonksiyon oluşturunuz.

3. Adım: Yaş bilgisinin yanlış girilmesini kontrol altına almak için Try bloku içinde istisna tanımlayınız ve Catch bloku içinde istisnaya ait mesajı verdiriniz.

4. Adım: Try blokuna ek olarak finally bloku oluşturarak bu blokta "Yaşamınızda mutluluklar dileriz." şeklinde mesaj verdiriniz.

5. Adım: Oluşturduğunuz "tablo" fonksiyonunu sırasıyla `tablo(5,10);` ve `tablo("dort", "beş");` şeklinde çağırarak sonucu ekrana yazdırınız.

```
<?php
    function yasHesapla($yil) {
      if(is_string($yil)) {
        throw new Exception("Sadece yıl bilgisi giriniz.");
      }
      $sonuc=2021-$yil;
     return $sonuc;
    }
    try {
      $yas=yasHesapla(1988);
      echo "<b>Yaşınız=</b>".$yas."<br>";
      $yas=yasHesapla(2001);
      echo "<b>Yaşınız=</b>".$yas."<br>";
      $yas=yasHesapla("ikibin");
      echo "<b>Yaşınız=</b>".$yas;
    }
    catch(Exception $e) {
      echo "<b style='color:red'>Hata mesajı:</b> " .$e->getMessage();
    }
    finally
    {
        echo "<br>";
        echo "<b style='background-color:orange'>
                    -Yaşamınızda mutluluklar dileriz.</b>";
    }
?>
```

ÖLÇME VE DEĞERLENDİRME

A) Aşağıdaki cümlelerde boş bırakılan yerlere doğru sözcükleri yazınız.

1. Programın kodlarından kaynaklı bir hata yoksa ama program yanlış sonuç üretiyorsa programda ……..…….... düzeyde hata var demektir.
2. `throw new Exception` kalıbı ……..…….....bloku içinde tanımlanır.
3. `throw new Exception` kalıbında belirtilen mesaj .....……..…...……..fonksiyonu ile getirilir.
4. Try blokunda istisna meydana gelse de gelmese de …….... blokundaki kodlar çalışır.

B) Fatal (ölümcül) düzeydeki hata ile warning (uyarı) düzeyindeki hata arasındaki iki farkı yazınız.

1)……
……
.....
.....
2)……
……
.....
.....

C) `catch(Exception $e)` **blokundaki** `$e` **değişkeni ile** `Exception` **sınıfından ulaşılabilecek dört fonksiyonu yazınız.**

1) `$e->`……
2) `$e->`……
3) `$e->`……
4) `$e->`…… …

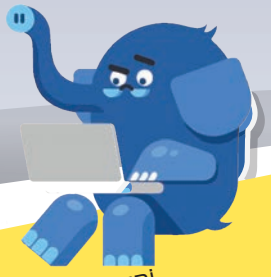
Ç) Aşağıdaki kodlarda bırakılan boşlukları verilen ekran görüntüsüne uygun olacak şekilde tamamlayınız.

Ekran Görüntüsü
Hata ile ilgili verilecek mesaj 1923

```
<?php
 function topla($a,$b){
   try{
      if(is_string($a) && ……){
         throw new ……(“Hata ile ilgili verilecek mesaj”);
      }
       $toplam=$a + $b;
       return $toplam;
   }catch(Exception ……) {
      echo $e->……”<br>”;
       }
 }
 echo topla(“üç”,“beş”);
 echo topla(……,23);
?>
```

7. ÖĞRENME BİRİMİ

DOSYA İŞLEMLERİ



KONULAR

7.1. PHP'DE KLASÖR VE DOSYA İŞLEMLERİ
7.2. PHP'DE DOSYA OLUŞTURMA
7.3. PHP'DE DOSYA AÇMA
7.4. PHP'DE DOSYA İÇİNE BİLGİ EKLEME
7.5. PHP'DE DOSYAYI SİLME

NELER ÖĞRENECEKSİNİZ?

- Klasör ekleme, silme, listeleme işlemleri
- Dosyaların isimlerini değiştirme
- PHP sayfasına başka bir dosyadaki kodları dâhil etme komutları
- PHP sayfalarında dosya işlemlerini sıralama
- Dosya oluşturma işlemi
- Dosya açma komutlarını kullanma
- Dosya açma komutları arasındaki farklar
- Dosyanın içine bilgi ekleme işlemi
- Dosya silme komutunu kullanma

TEMEL KAVRAMLAR

klasör işlemleri, dosya işlemleri, dosya oluşturma, dosya açma, dosyaya bilgi ekleme, dosya silme.



Hazırlık Çalışmaları

1. **PHP sayfalarında hangi tür dosyalar üzerinde işlem yapılır?**
2. **Web sayfası ile metin dosyalarının içeriğine müdahale edilerek ne gibi işlemler yapılabilir?**

7.1. PHP'DE KLASÖR VE DOSYA İŞLEMLERİ

PHP, klasör ve dosyalara ulaşmak ve onları yönetmekte oldukça yetenekli bir dildir. Anında onlarca klasör oluşturabilir ve oluşturulan klasörlerin içine dosyalar aktarabilir. Var olan klasörlerin içindeki dosyalar görüntülenebilir, dosyalara müdahale edilebilir, dosyalarla ilgili detaylara ulaşılabilir.

Mkdir() Fonksiyonu: PHP'de yeni bir klasör oluşturmak için kullanılan fonksiyondur. Fonksiyona parametre olarak klasörün konumu, klasör adıyla birlikte yazılır. Eğer aynı isimde bir klasör varsa uyarı (warning) düzeyinde hata mesajı alınır.

Kişiye özel profil resimlerinin saklandığı bir web sayfasında yeni kaydolan kullanıcıya özel bir resim klasörü oluşturmak gerekir. Bu durumda **mkdir() fonksiyonu** kulanılarak kullanıcının profil52 isminde bir klasör oluşturulabilir.

```
<?php
mkdir("Resimler/profil52");
 //PHP proje klasörünün altındaki "Resimler" klasörünün içinde "profil52" adında
bir klasör oluşturulur.
mkdir("profil52");
//PHP proje klasörünün içinde "profil52" adında bir klasör oluşturulur.
mkdir("../profil52");
//PHP proje klasörünün bir alt klasörünün içinde "profil52" adında bir klasör
oluşturulur.
?>
```

Rmdir() Fonksiyonu: PHP'de istenilen klasörü silmek için kullanılan fonksiyondur. Fonksiyona parametre olarak silinmek istenilen klasörün konumu, klasör adıyla birlikte yazılır. Eğer verilen isimde bir klasör yoksa uyarı (warning) düzeyinde hata mesajı alınır.

```
<?php
mkdir("Resimler/profil52");
 //PHP proje klasörünün altındaki "Resimler" klasörünün içinde "profil52"
adındaki klasörü siler.
mkdir("profil52");
//PHP proje klasörünün içinde "profil52" adındaki klasörü siler.
mkdir("../profil52");
//PHP proje klasörünün bir alt klasörünün içinde "profil52" adındaki klasörü
siler.
?>
```

Opendir() Fonksiyonu: Parametresine girilen konumdaki klasörü açan ve klasörü temsil eden değeri geriye döndürür. Genellikle readdir() fonksiyonu ile birlikte klasör içindeki dosyaları listelemek için kullanılır.

```
$klasor = opendir("Resimler");
```

Readdir() Fonksiyonu: Klasör içindeki dosyaları listelemek için kullanılır. Parametre olarak opendir() fonksiyonundan dönen klasörü temsil eden değeri alır.

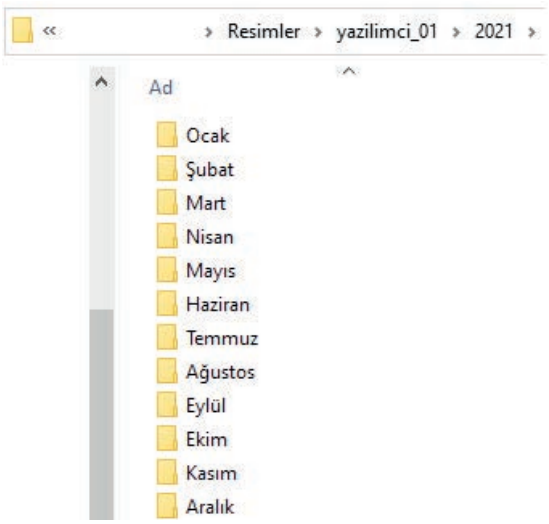
```
<?php
$klasor = opendir("Resimler");
while($dosya = readdir($klasor)){ //Klasördeki tüm dosyalara tek tek
ulaşılıyor.
    if (!is_dir($dosya)){
        echo $dosya.'<br>';
         //$dosya değişkenindeki değer klasör değilse alt alta yazdır.
    }
}
?>
```

Önemli

`is_dir`() fonksiyonu, parametre olarak yazılan değerin klasör olup olmadığını kontrol eder. Eğer klasörse geriye "true" değeri döndürür.

1. Uygulama

Görsel 7.1'de verilen şekilde bir klasör oluşturmak için aşağıdaki adımları takip ediniz.



Görsel 7.1: Klasör oluşturma

1. Adım: Proje klasörünüze "Resimler" adında bir klasör ekleyiniz.

2. Adım: Profil adı, yıl bilgisi için değişken oluşturunuz ve değişkenlere değer atayınız.

3. Adım: Resimler klasörü içinde $profilAdi adında bir klasör oluşturunuz.

4. Adım: Profil adında oluşturulan klasörün içinde $yil adında bir klasör oluşturunuz.

5. Adım: Yıl adında oluşturulan klasörün içine yılın tüm aylarının adında klasörler oluşturunuz.

```
<?php
$profilAdi="yazilimci_01";
$yil="2021";
mkdir("Resimler/".$profilAdi);
mkdir("Resimler/".$profilAdi."/".$yil);
$aylar=["Ocak","Şubat","Mart","Nisan","Mayıs","Haziran","Temmuz" ,
"Ağutos", "Eylül","Ekim","Kasım","Aralık"];

for($i=0;$i<12;$i++){
    $url="Resimler/".$profilAdi."/".$yil."/".$aylar[$i];
    mkdir($url);
}
?>
```

7.1.1. Dosya İçe Aktarma Fonksiyonları

PHP dosyası içine başka bir PHP dosyası aktarma işlemine içe aktarma denir. PHP sayfalarında kod karışıklığını ortadan kaldırmak için bazı fonksiyonlar, değişkenler vs. yapılar farklı PHP sayfalarından aktarılabilir. İçe aktarma işlemini yapmak için işlevleri çok benzer dört farklı fonksiyon kullanılır. Bütün fonksiyonlar aktarılacak dosya yolunun yazılacağı bir parametreden oluşur.

include() Fonksiyonu: İçe aktarılmak istenen PHP sayfası aynı sayfada birden çok yerde çağrılabilir. Eğer aktarım sırasında hata oluşursa uyarı düzeyinde bir hata verir.

```
include(icerik.php');
```

include_once() Fonksiyonu: İçe aktarılmak istenen PHP sayfası aynı sayfada sadece bir kez çağrılabilir. Eğer aktarım sırasında hata oluşursa uyarı düzeyinde bir hata verir.

```
include_once(ayarlar.php');
```

require() Fonksiyonu: Kullanımı include() fonksiyonunda olduğu gibidir. İçe aktarılmak istenen PHP sayfası aynı sayfada birden çok yerde çağrılabilir. Eğer aktarım sırasında hata oluşursa ölümcül (fatal) düzeyde bir hata verir.

```
require('resimler.php');
```

require_once() Fonksiyonu: Kullanımı include_once() fonksiyonunda olduğu gibidir. İçe aktarılmak istenen PHP sayfası aynı sayfada birden çok yerde çağrılabilir. Eğer aktarım sırasında hata oluşursa ölümcül (fatal) düzeyde bir hata verir.

```
require_once(veritabani_ayarlari.php');
```

2. Uygulama

Dosya içe aktarma işlemini aşağıdaki adımları takip ederek yapınız.

1. Adım: Proje klasörünüze "ekle_sil.php", "listele.php", "index.php" adında sayfalar oluşturunuz.

2. Adım: "ekle_sil.php" sayfasında klasör ekleme ve klasör silme işlemleri için fonksiyonlar oluşturunuz.

```
<?php
function klasor_ekle($klasor_adi){
    if(!is_dir($klasor_adi)){ // Klasör değil ise
       mkdir($klasor_adi);
```

```
echo “Klasör eklendi.<br>”;
}
}
function klasor_sil($klasor_adi){
    if(is_dir($klasor_adi)){ // Klasör ise
rmdir($klasor_adi);
echo “Klasör silindi.<br>”;
}
}
?>
```

3. Adım: “listele.php” sayfasında klasör listeleme işlemleri için fonksiyon oluşturunuz.

```
<?php
function klasor_listele($konum){
$klasor = opendir($konum);
echo “<b>Klasörler Listeleniyor...</b><br>”;
while($dosya = readdir($klasor)){
    if (!is_dir($dosya)){
        echo $dosya.‘<br>’;
    }
  }
}
?>
```

4. Adım: “index.php” sayfasında ekle_sil.php ve listele.php sayfalarını içe aktarınız.

```
<?php
require_once(“ekle_sil.php”); // Sayfa içeri aktarıldı.
require_once(“listele.php”);

klasor_ekle(“Kayitlar”);
klasor_ekle(“Kayitlar/Yasir”);
klasor_ekle(“Kayitlar/Selman”);

klasor_listele(“Kayitlar”);

klasor_sil(“Kayitlar/Yasir”);
klasor_sil(“Kayitlar/Selman”);
?>
```

7.2. PHP’DE DOSYA OLUŞTURMA

PHP; dili, belirli türlerdeki dosyaları oluşturma, okuma, içeriğini değiştirme ve silme işlemlerini kolaylıkla yapabilir. Bu yönüyle oldukça güçlü bir programlama dilidir. Bir döngü kurarak saniyeler içinde binlerce dosya oluşturabilir, düzenleyebilir ve silebilir (Görsel 7.2).



Görsel 7.2: Dosya işlemleri

PHP ile txt, xml, php, css, html, sql vb. farklı uzantılarda dosyalar oluşturulur. Oluşturulan dosyalar üzerinde işlemler yapılabilir.

Touch() Fonksiyonu: PHP’de yeni bir dosya oluşturmak için kullanılan fonksiyondur. Fonksiyona parametre olarak dosyanın konumu, dosya adıyla birlikte yazılır.

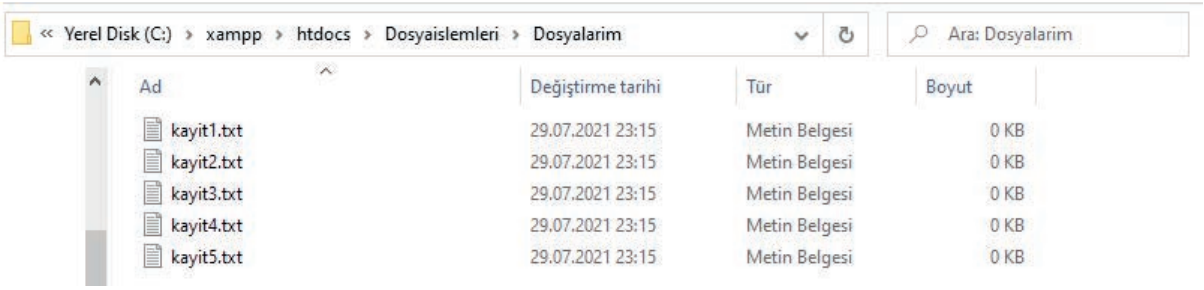
Örnek

```
<?php
touch("Dosyalarim/dosya.txt");
?>
```

Yukarıdaki örnek kod çalıştırıldığında "Dosyalarim" adlı klasörün içine "dosya.txt" metin dosyası oluşturulur.

3. Uygulama

Dizi elemanı içindeki eleman sayısına ve eleman değerlerine göre Görsel 7.3'teki gibi "txt" uzantılı dosyalar oluşturma işlemini aşağıdaki adımları takip ederek yapınız.



Görsel 7.3: Dosya oluşturma

1. Adım: Proje klasörünüze "Dosyalarim" adında bir klasör ekleyiniz.

2. Adım: Kendi belirleyeceğiniz isimde ve kendi belirlediğiniz değerlere sahip en az beş elemanlı bir dizi değişkeni oluşturunuz.

3. Adım: For döngüsü ile dizideki elemanların hepsi için yeni "txt" dosyası oluşturunuz.

```
<?php
$dosya_isimleri = array("kayit1","kayit2","kayit3","kayit4","kayit5");
for($i=0;$i<count($dosya_isimleri);$i++){
  touch("Dosyalarim/".$dosya_isimleri[$i].".txt");
}
?>
```

7.3. PHP'DE DOSYA AÇMA

Dosyaların içeriklerine ulaşmak, içeriklerini değiştirmek veya dosyaları silmek için öncelikle dosyanın açılması gerekir. Ayrıca açılan tüm dosyalar kapatılmalıdır.

fopen() Fonksiyonu: PHP'de var olan bir dosyayı açmak için kullanılır. Fonksiyona iki parametre girilir. İlk parametre olarak açılmak istenen dosyanın yolu yazılır. İkinci parametre olarak dosya açma modu yazılır. Altı tane dosya açma modu bulunmaktadır. Dosya açma modları Tablo 7.1'de listelenmiştir.

Tablo 7.1: *Dosya Açma Modları*

Dosya Modu	Açıklaması
r	Okuma (Read) modudur. Dosyanın sadece okunmak için açılacağını belirtir.
w	Yazma (Write) modudur. Dosyanın sadece yazılmak için açılacağını belirtir. Yeni bilgiler eklenmeden önce dosyadaki tüm içerik silinir.
a	Ekleme (Append) modudur. Dosyanın sadece yazılmak için açılacağını belirtir. Yeni bilgileri, eski bilgilerin üzerine ekler.
r+	Hem okuma hem yazma modudur. Dosyanın okuma ve yazma için açılacağını belirtir. Yazma işleminde önce dosyadaki tüm içerik silinir.
w+	Hem okuma hem yazma modudur. Dosyanın okuma ve yazma için açılacağını belirtir. Okuma ve ekleme işlemlerinden önce dosyadaki tüm içerik silinir.
a+	Hem okuma hem yazma modudur. Dosyanın okuma ve yazma için açılacağını belirtir. Yeni bilgileri, eski bilgilerin üzerine ekler.

```
$dosya = fopen('Dosyalarim/dosya.txt','r');
```

fread() Fonksiyonu: Dosyadaki verileri okumak için kullanılır. Fonksiyona iki parametre girilir. İlk parametresine $dosya(dosya değişeni) yazılır. İkinci parametreye ise açılan dosyadan kaç bayt veri okunacağı yazılır.

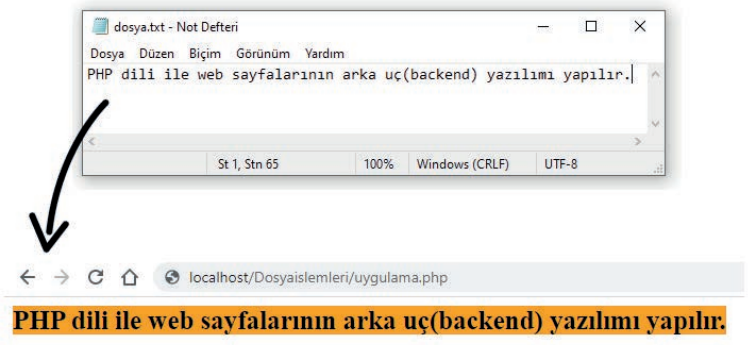
Önemli

Okunacak dosyanın boyutunu bayt cinsinden hesaplamak için **filesize('dosya yolu');** fonksiyonu kullanılır.

```
$dosya_boyutu = filesize('Dosyalarim/dosya.txt');
```

4. Uygulama

dosya.txt isimli metin dosyasındaki bilgileri okuyup Görsel 7.4'te görüldüğü gibi ekrana yazdırma işlemini aşağıdaki adımları takip ederek yapınız.



Görsel 7.4: Dosya okuma

1. Adım: Proje klasörünüze "Dosyalarim" adında bir klasör ekleyiniz.

2. Adım: Dosyalarim klasörünün içine dosya.txt isimli bir metin dosyası oluşturunuz.

3. Adım: dosya.txt dosyasının içine "PHP dili ile web sayfalarının arka uç (backend) yazılımı yapılır." metnini yazınız ve dosyayı kaydediniz.

4. Adım: dosya.txt dosyasına kaydedilmiş bilgileri yazı boyutu 25 piksel, arka plan rengi turuncu, yazı rengi siyah olacak şekilde web sayfasına yazdırınız.

```
<?php
$dosya = fopen('Dosyalarim/dosya.txt', 'r+');
$dosya_boyutu = filesize(' Dosyalarim/dosya.txt');
$dosya_icerigi=fread($dosya,$dosya_boyutu);
echo "<b style='font-size:25px; background-color:orange'>
$dosya_icerigi  </b>";
fclose($dosya);
?>
```

Önemli

fopen() fonksiyonu kullanılırken istenilirse sadece 'r' dosya modu da kullanılabilir.

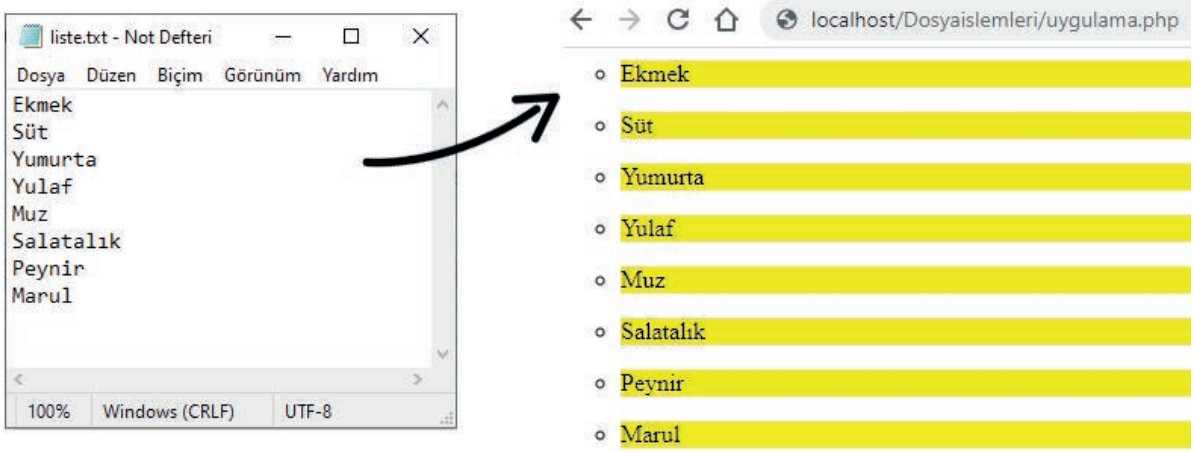
fgets() Fonksiyonu: Okumak için açılan dosyadaki verilere satır satır ulaşmak için kullanılan fonksiyondur. Fonksiyon her çağrılmasında bir sonraki satırdaki verilere ulaşır.

Önemli

Açılan dosyanın verileri okunurken dosyanın sonuna gelinip gelinmediği bilgisi **feof(**`$dosya`**)** fonksiyonu ile elde edilir.

5. Uygulama

liste.txt isimli metin dosyasındaki bilgileri satır satır okuyup Görsel 7.5'te görüldüğü gibi ekrana yazdırma işlemini aşağıdaki adımları takip ederek yapınız.



Görsel 7.5: Satır satır okuma

1. Adım: Proje klasörünüze "Dosyalarim" adında bir klasör ekleyiniz.

2. Adım: Dosyalarim klasörünün içine liste.txt isimli bir metin dosyası oluşturunuz.

3. Adım: liste.txt dosyasının içine "Ekmek, Süt, Yumurta, Yulaf, Muz, Salatalık, Peynir, Marul" değerlerini alt alta yazınız ve dosyayı kaydediniz.

4. Adım: liste.txt dosyasına kaydedilmiş bilgileri arka plan rengi sarı, yazı rengi siyah olacak şekilde web sayfasına alt alta yazdırınız.

```
<?php
$dosya = fopen('Dosyalarim/liste.txt', 'r');
while(!feof($dosya)){
   $okunan_satir=fgets($dosya);
   echo "<ul type='circle'>
            <li style='background-color:yellow'> $okunan_satir </li>
         </ul>";
}
fclose($dosya);
?>
```

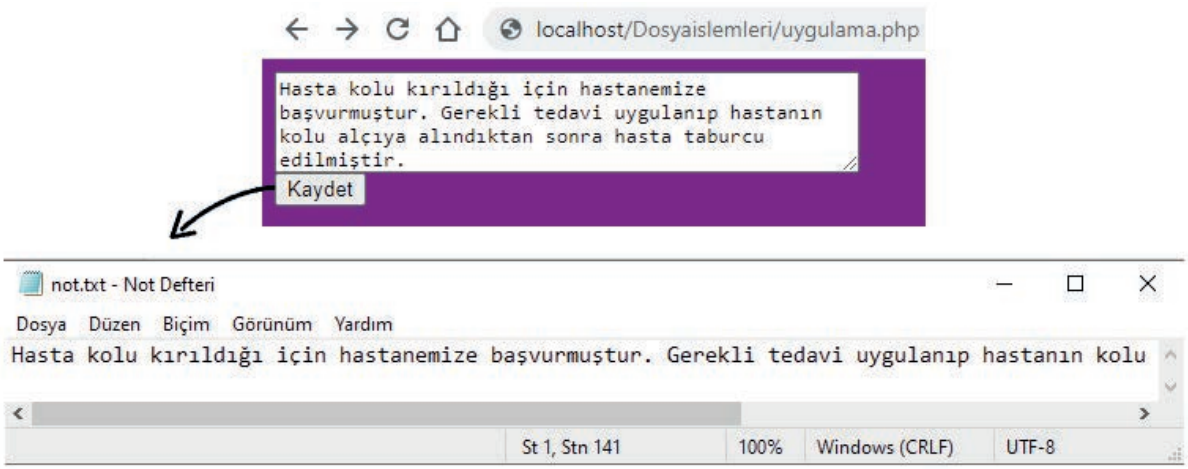
7.4.PHP'DE DOSYA İÇİNE BİLGİ EKLEME

fwrite() Fonksiyonu: Dosyanın içine bilgi eklemek için fwrite() fonksiyonu kullanılır. Fonksiyona iki parametre girilir. İlk parametresine $dosya(dosya değişeni) yazılır. İkinci parametreye ise dosyaya eklenecek bilgiler yazılır.

```
$dosya = fopen('Dosyalarim/dosya.txt', 'w');
fwrite($dosya,'Dosyaya eklenecek bilgiler');
```

6. Uygulama

Hastaya ait tedavi geçmişi bilgilerini not.txt dosyasına yazdırma işlemini Görsel 7.6'da görüldüğü gibi aşağıdaki adımları takip ederek yapınız.



Görsel 7.6: Bilgi ekleme

1. Adım: Proje klasörünüze uygulama.php ve kayit.php adında web sayfası ekleyiniz.

2. Adım: Proje klasörünüze "Dosyalarim" adında bir klasör ekleyiniz ve "Dosyalarim" klasörünün içine de "not.txt" metin dosyası ekleyiniz.

3. Adım: uygulama.php sayfasının arka plan rengini turuncu yapınız.

4. Adım: uygulama.php sayfasına form elemanı ekleyiniz ve form elemanı aracılığı ile kayit.php sayfasına textarea metin kutusundaki verileri gönderiniz.

```
<html>
<body style="background-color:purple">
    <div>
    <form action="kayit.php" method="get">
    <textarea  name="notlar" rows="4" cols="50">
        Notunuzu buraya yazınız.
    </textarea>
      <br>
      <input type="submit" value="Kaydet">
    </form>
    <div>
</body>
</html>
```

5. Adım: kayit.php sayfasına GET metodu ile gönderilen verileri "not.txt" metin dosyasına yazdırınız.

```
<?php
if(isset($_GET['notlar'])){
         $dosya = fopen('Dosyalarim/not.txt', 'w+');
         $gelen=$_GET['notlar'];
         fwrite($dosya,$gelen);
         fclose($dosya);
}
else{  echo "Dosya bulunamadı!";}
?>
```

7.5.PHP'DE DOSYAYI SİLME

unlink() Fonksiyonu: Var olan bir dosyayı silmek için unlink() fonksiyonu kullanılır.

```
unlink('Dosyalarim/dosya.txt');
```

Önemli

Bir dosyanın var olup olmadığı bilgisi **file_exists(($dosya)** fonksiyonu ile elde edilir.

```
if(file_exists('dosya.txt')){
    echo "Böyle bir dosya var";
}
else{ echo "Böyle bir dosya yok"; }
```

7. Uygulama

03, 06 ve 16 numaralı öğrencilerin "aktif/12.Sınıf" klasöründeki dosya bilgilerini silip "mezun" klasöründe mezun olan öğrenciler için yeni dosya oluşturma işlemini Görsel 7.7'de görüldüğü gibi aşağıdaki adımları takip ederek yapınız.



Görsel 7.7: Dosya silme

1. Adım: Proje klasörünüze "aktif" ve "mezun" isimlerinde yeni klasörler ve "aktif" klasörünün içine "12. Sınıf" adında bir alt klasör oluşturunuz.

2. Adım: "12.Sınıf" klasörünün içine Görsel 7.7'de görüldüğü gibi 03, 06, 16, 35, 52, 66 numaralı öğrenciler için "txt" uzantılı dosyalar oluşturunuz.

3. Adım: "12.Sınıf" klasöründe bulunan 03, 06 ve 16 numaralı öğrencilerin dosyalarını siliniz.

4. Adım: Dosyaları silinen öğrenciler için "mezun" klasöründe dosya adı mezuniyet yılıyla başlayan yeni dosyalar oluşturunuz.

```
<?php
unlink("aktif/12.Sınıf/03numaraliOgrenci.txt");
unlink("aktif/12.Sınıf/06numaraliOgrenci.txt");
unlink("aktif/12.Sınıf/16numaraliOgrenci.txt");
echo "3 Öğrencinin dosya bilgileri silindi.";
touch("mezun/2021_03numaraliOgrenci.txt");
touch("mezun/2021_06numaraliOgrenci.txt");
touch("mezun/2021_16numaraliOgrenci.txt");
echo "3 Öğrenci için yeni dosyalar oluşturuldu.";
?>
```

ÖLÇME VE DEĞERLENDİRME

A) Aşağıdaki cümlelerde boş bırakılan yerlere doğru sözcükleri yazınız.

1. Klasör içindeki dosyaları listelemek için …….. fonksiyonu kullanılır.
2. …….. fonksiyonu, parametre olarak yazılan değerin klasör olup olmadığını kontrol eder.
3. PHP'de yeni bir dosya oluşturmak için …….. fonksiyonu kullanılır.

B) require() fonksiyonu ve require_once() fonksiyonu arasındaki farkı yazınız.

……
……
……
……

C) fopen() fonksiyonu ile dosya açılırken kullanılabilecek okuma modlarından üç tanesini yazınız.

1. ……..
2. ……..
3. ……..

Ç) Aşağıdaki fonksiyonların web sayfası çalıştırıldığında hangi işlevi yaptıklarını karşılarındaki boşluklara yazınız.

1	`touch("dosya.txt");`	
2	`fopen("makale.php","r+");`	
3	`mkdir("Resimler");`	
4	`is_dir($dosya);`	
5	`rmdir("Resimler");`	
6	`unlink("profil.txt");`	
7	`file_exists("evrak.txt");`	
8	`fclose($dosya);`	

8. ÖĞRENME BİRİMİ

OTURUM YÖNETİMİ



KONULAR

8.1. ÇEREZ YÖNETİMİ

8.2. OTURUM MİMARİSİ VE Ş_SESSION DEĞİŞKENİ

NELER ÖĞRENECEKSİNİZ?

- Çerez kavramı
- PHP sayfalarında çerez (cookie) yapısı
- $_COOKIE genel değişkeninin kullanım yerleri
- Oturum kavramı
- PHP sayfalarında oturum (session) yapısı
- $_SESSION genel değişkeninin kullanım yerleri
- Çerez ve oturum kavramları arasındaki farklar
- Oturum mimari yapısını kullanma
- PHP sayfalarında veri saklama

TEMEL KAVRAMLAR

oturum yönetimi, çerezler yönetimi, $_ SESSION genel değişkeni, $_ COOKIE genel değişkeni.



Hazırlık Çalışmaları

1. **Çerez yapısı kullanılarak web sayfalarında kullanıcıların genellikle hangi bilgileri alınır?**
2. **Web sayfalarında açılan oturumların kullanıcılara katkıları nelerdir?**

8.1. ÇEREZ YÖNETİMİ

Çerez yönetimi ile kullanıcıya ait bilgiler kayıt altına alınır. Bu bilgiler reklam, veri toplama, kullanıcı deneyimini iyileştirme gibi çeşitli amaçlarla kullanılır (Görsel 8.1). Çerez dosyaları kullanıcının kendi bilgisayarında saklanır. Çerez dosyalarına ulaşmak ve üzerinde değişiklik yapmak kolaydır. Bu sebeple güvenlik problemi oluşturabilecek bilgiler çerez yapıları ile kaydedilmemelidir. Örneğin web sayfasına girilen kullanıcı adı ve şifresi beni hatırla seçeneği ile çerez yapısında saklanmamalıdır. Bu tarz işlemler için oturum (session) yapısını kullanmak çok daha güvenlidir.



Görsel 8.1: Çerez yönetimi

Kişisel hakların korunması için dünya genelinde çerez kullanımı ile toplanacak verilere her geçen gün kısıtlamalar getirilmektedir. Yasal web sayfalarında çerez yapısı kullanılmadan önce kullanıcıdan izin istenir. Kullanıcı, çerez kullanımına izin vermiş olsa dahi dilerse internet tarayıcısı ayarlarından çerez kullanımını iptal edebilir.

Kullanıcının bilgisayarında çerez oluşturmak için **setcookie() fonksiyonu** kullanılır. Bu fonksiyon genel olarak üç parametre ile kullanılır.

```
setcookie(çerez adı, çerez içeriği, çerez geçerlilik zamanı);
$cookie_adi = “Selman_YILDIRIM”;
$cookie_degeri = “Siteye Yalova’dan giriş yapti”;
setcookie($cookie_adi, $cookie_degeri,time() + (60*60*24));
//60*60*24 = 86400 = 1 gün geçerli çerez
setcookie($cookie_adi, $cookie_degeri);
// Sadece çerez değerini güncellemek için 2 parametre kullanılabilir.
```

1. Uygulama

Web sayfasına giriş sayısını çerez ile tutma ve ekrana yazdırma işlemlerini aşağıdaki adımları takip ederek yapınız.

1. Adım: Çerez adını vermek için bir değişkene değer atayınız.

2. Adım: Giriş sayısını tutan çerez var mı diye kontrol ediniz.

3. Adım: Eğer çerez var ise değerini bir artırınız ve setcookie() fonksiyonu ile çerezi kaydediniz.

4. Adım: Çerezin değerini ekrana yazdırınız.

5. Adım: Eğer çerez yok ise çerezin değerine sıfır veriniz ve çerezi kaydediniz.

```
<?php
$cookie_adi="kullanici";
if(isset($_COOKIE[$cookie_adi])){
    $cookie_degeri=$_COOKIE[$cookie_adi]+1;
   setcookie($cookie_adi,$cookie_degeri,time()+(60*60*24));
    echo $_COOKIE[$cookie_adi];
  }
  else {
    $cookie_degeri=0;
    setcookie($cookie_adi, $cookie_degeri,time() + (60*60*24));
  }
?>
```

2. Uygulama

Kullanıcının seçtiği kategori bilgisini çerez olarak saklama işlemini aşağıdaki adımları takip ederek yapınız (Görsel 8.2).



Görsel 8.2: Çerez oluşturma

1. Adım: Proje klasörünüze "Resimler" adında bir klasör ekleyiniz.

2. Adım: Proje klasörünüze "uygulama.php" adında bir sayfa ekleyiniz.

3. Adım: uygulama.php sayfasında farklı kategoriler için <a> elemanı ile link verilmiş resimler ekleyiniz.

```
<html>
 <body>
 <div style="text-align:center;background-color:tomato">
  <h3 style="margin:0px">Kategoriler</h3>
```

```
  <a href="uygulama1.php?kategori=cicek">
    <img src="Resimler/cicek.jpg" style="height:100px">
  </a>
  <a href="uygulama1.php?kategori =sus_esyasi">
    <img src="Resimler/esya.jpg" style="height:100px">
  </a>
  <a href="uygulama1.php?kategori =hayvan">
    <img src="Resimler/hayvan.jpg" style="height:100px">
  </a>
  <a href="uygulama1.php?kategori =spor">
    <img src="Resimler/spor.jpg" style="height:100px">
  </a>
 </div>
 </body>
</html>
```

4. Adım: Çerez adını verirken kullanmak için kullanıcının IP numarasını değişkene atayınız.

5. Adım: $_GET genel değişkeni ile adres çubuğu üzerinden gelen "kategori" değerini değişkene atayınız.

6. Adım: Setcookie fonksiyonu ile bir gün boyunca kullanılacak bir çerez oluşturunuz.

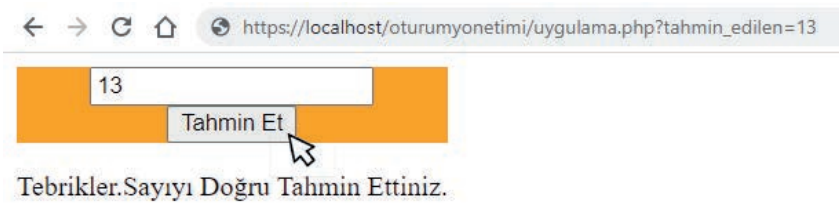
```
<?php
$kullanici_ip=$_SERVER["REMOTE_ADDR"];
$kategori=$_GET['kategori'];
$cookie_adi = "ip_".$kullanici_ip;
$cookie_deger = $kategori;
setcookie($cookie_adi, $cookie_deger,time() + (60*60*24));
?>
```

Önemli

Tüm kodlar "uygulama.php" sayfasına yazılacaktır.

3. Uygulama

Tutulan bir adet rastgele sayının çerez dosyasında saklandığı ve kullanıcının bu sayıyı tahmin etmeye çalıştığı sayı tahmin oyunu işlemlerini Görsel 8.3'te görüldüğü gibi aşağıdaki adımları takip ederek yapınız.



Görsel 8.3: Sayı tahmin oyunu

1. Adım: Proje klasörünüze "uygulama.php" adında bir sayfa ekleyiniz.

2. Adım: içine action özelliği uygulama.php olan bir form elemanı ekleyiniz.

3. Adım: Form elemanı içine bir adet metin kutusu ve bir adet buton ekleyiniz.

```
<html>
    <body>
        <div style="background-color:orange; width:250px;text-align:center">
            <form action="uygulama.php" method="get" >
              <input type="text" name="tahmin_edilen"><br>
              <input type="submit" value="Tahmin Et">
            </form>
        </div>
    </body>
</html>
```

4. Adım: 0-10 arasında rastgele bir sayı oluşturup bir saatlik kullanım süresi olan çerez dosyasına kaydediniz.

5. Adım: Eğer $_GET genel değişkeni ile adres çubuğu üzerinden gelen "tahmin_edilen" değeri çerezdeki değere eşitse ekrana "Tebrikler. Sayıyı Doğru Tahmin Ettiniz." değilse ekrana "Tekrar Tahminde Bulununuz." mesajını yazdırınız.

```
<?php
if(!isset($_COOKIE['tutulan_sayi'])) {
    $cookie_deger = rand(0,10);
    setcookie("tutulan_sayi", $cookie_deger,time() + 3600); // 1 saat
  } else {
    if(isset($_GET['tahmin_edilen']))
    {
       if($_GET['tahmin_edilen']==$_COOKIE['tutulan_sayi']){
           echo "Tebrikler. Sayıyı Doğru Tahmin Ettiniz.";
       }else{
           echo "Tekrar Tahminde Bulununuz.";
       }
    }
  }
?>
```

Önemli

Tüm kodlar "uygulama.php" sayfasına yazılacaktır.

Çerez Kaldırma: Çerezin geçerlilik zamanı dolduğunda çerez otomatik silinir.

```
setcookie($cookie_adi, $cookie_degeri,time() -1);
```

Yukarıdaki örnek kodda görüldüğü gibi "time() – 1" ifadesi çerezin silinmesi için yeterlidir.

Aşağıdaki örnek kodda çerez onayı için sunulan form elemanından gelen cevaba göre çerez oluşturulur veya kaldırılır.

```php
<?php
if(isset($_GET[“cerez_onay”]))
{
   $cerez_onay=$_GET[“cerez_onay”];
   if($cerez_onay==true){
      setcookie(“Cerez_adi”, $cerez_icerigi,time() + (60*60*24*365)); // 1 yıl
   }
   else{
     setcookie(“Cerez_adi”, $cerez_icerigi,time() - 1); // Çerez siliniyor
   }
 }
?>
```

8.2. OTURUM MİMARİSİ VE $_SESSION GENEL DEĞİŞKENİ

Oturum Mimarisi: Çerez kullanımında olduğu gibi oturum mimarisi ile de kullanıcıya ait bilgiler kayıt altına alınır fakat bu bilgiler kullanıcının bilgisayarında değil PHP sayfasının yüklü olduğu server tarafında saklanır. Oturum dosyalarına ulaşmak ve üzerinde değişiklik yapmak çok zordur. Bu sebeple güvenlik problemi oluşturabilecek bilgiler genellikle oturum (session) yapısı ile saklanır. Oluşturulan oturumlar web sayfasının bulunduğu sekme kapatıldığında veya tarayıcı kapatıldığında sonlanır.

$_SESSION Genel Değişkeni: Oturum mimarisinde verilerin saklanması, değiştirilmesi, güncellenmesi için kullanılan genel değişken $_SESSION değişkenidir.

Kullanımı $_SESSION[‘değişken_adı’] şeklindedir.

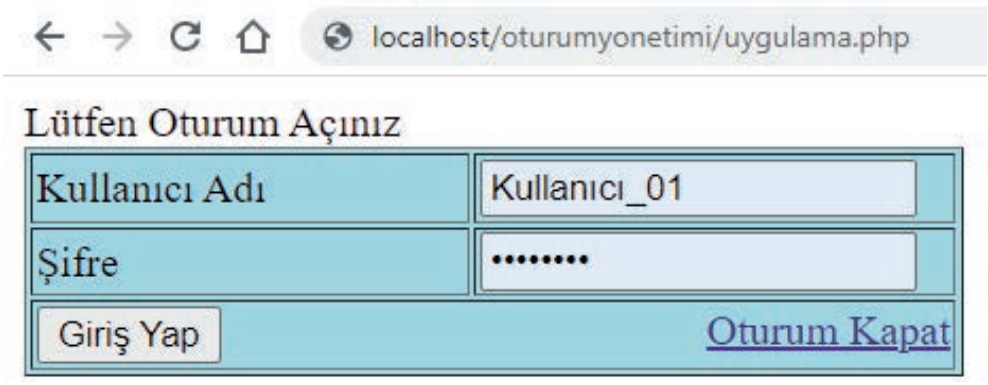
$_SESSION değişkeni kullanmak için her sayfada PHP kodlarının başına oturum başlatma komutu `session_start();` yazılmalıdır.

Tüm Oturumları Silmek: Oluşturulan tüm oturumları silmek için `session_destroy();` komutu kullanılır.

Oluşturulan oturumu özel olarak silmek için `unset($_SESSION[“session_adı”]);` komutu kullanılır.

4. Uygulama

Kullanıcı adı ve şifre bilgilerinin kontrol edilip eğer doğru giriş yapıldıysa oturum bilgilerini saklama ve “Oturum Kapat” bağlantısı tıklandıysa oturum bilgilerini kaldırma işlemlerini aşağıdaki adımları takip ederek yapınız (Görsel 8.4).



Görsel 8.4: Oturum açma ve kapatma

1. Adım: Proje klasörünüze “uygulama.php”, “sifreKontrol.php”, oturum “Kapat.php” adında sayfalar ekleyiniz.

2. Adım: “uygulama.php” sayfasında kullanıcı adı ve şifre kontrolü için form elemanları ekleyiniz.

uygulama.php sayfası HTML kodları

```
<form name=“kullanicigirisi” method=“post” action=“sifreKontrol.php”>
<table bgcolor=“lightblue” width=“354” border=“1” >
  <tr>
    <td width=“150px” height=“25px”>Kullanıcı Adı</td>
    <td width=“150px”>
      <input type=“text” name=“k_adi” >
    </td>
  </tr>
  <tr>
    <td width=“150px” height=“25px”>Şifre</td>
    <td width=“150px”>
      <input type=“password” name=“sifre” >
    </td>
  </tr>
  <tr>
    <td colspan=“3”>
      <input type=“submit” value=“Giriş Yap”>
      <a href=“oturumKapat.php” style=“float:right”> Oturum Kapat </a>
   </td>
  </tr>
</table>
</form>
```

3. Adım: “uygulama.php” sayfasında daha önceden açılmış oturum olup olmadığını kontrol ediniz.

uygulama.php sayfası PHP kodları

```
<?php
session_start();
if(isset($_SESSION[“oturum_sahibi”]))
{
  echo “<b style=‘color:green’>Sayfamıza Hoşgeldin”
   .$_SESSION[“oturum_sahibi”].“</b>”;
}
else
{
  echo “<style=‘color:danger’>Lütfen Oturum Açınız </b>”;
}
?>
```

4. Adım: “sifreKontrol.php” sayfasında “uygulama.php” sayfasından gönderilen kullanıcı adı ve şifre bilgilerini kontrol ediniz ve tekrar “uygulama.php” sayfasına yönlendirme yapınız.

sifreKontol.php sayfası PHP kodları

```
<?php
session_start();
if($_POST[“k_adi”] == “Kullanıcı_01” && $_POST[“sifre”] == “php123”)
{
    $_SESSION[“oturum_sahibi”] = $_POST[“k_adi”];
    header(“location:uygulama4.php”);
}
else
{
    header(“location:uygulama4.php”);
}
?>
```

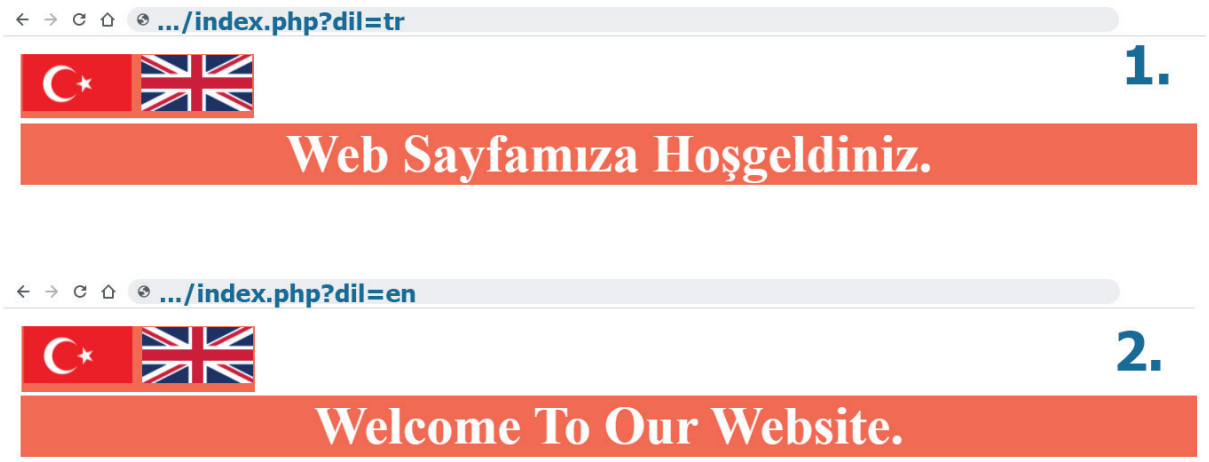
5. Adım: “oturumKapat.php” sayfasında oturumu kaldırınız ve tekrar “uygulama.php” sayfasına yönlendirme yapınız.

oturumKapat.php sayfası PHP kodları

```
<?php
    session_start();
    unset($_SESSION[“oturum_sahibi”]);
    header(“location:uygulama4.php”);
?>
```

5. Uygulama

Web sitesinin oturumu sürdürdüğü sürece seçtiği dile göre sayfanın içerik dilini değiştirme işlemini Görsel 8.5’te görüldüğü gibi aşağıdaki adımları takip ederek yapınız.



Görsel 8.5: Dil seçimi

1. Adım: Proje klasörünüze “index.php” adında sayfa ekleyiniz.

2. Adım: Proje klasörünüze "Resimler" adında bir klasör ve klasörün içine eşit oranlarda Türkiye ve İngiltere bayrakları ekleyiniz.

3. Adım: "index.php" sayfasına elemanı içinde genişlikleri 50 piksel olan Türkiye ve İngiltere bayraklarını ekleyiniz.

4. Adım: Eklenen bayraklara bağlantı veriniz. Bağlantı ile "dil" değişkeninin değerini gönderiniz.

```
<html>
<body>
<div style="background-color:tomato; width:105px; text-align:center;">
  <a href="index.php?dil=tr">   <img src="Resimler/tr.png" style="width:50px;"></a>
  <a href="index.php?dil=en">   <img src="Resimler/en.png" style="width:50px;"></a>
</div>
```

5. Adım: Bayraklardaki bağlantıdan gönderilen `$_GET["dil"]` değişkeninin değerini `$_SESSION["dil"]` genel değişkenine atayınız.

6. Adım: Eğer `$_SESSION["dil"]` değişkeninin değerine göre içerik bilgisini Türkçe veya İngilizceye çeviriniz.

```
<?php
session_start();
if(isset($_GET["dil"])){
    $_SESSION["dil"]=$_GET["dil"];
}

if(isset($_SESSION["dil"]))
{
    if($_SESSION["dil"]=="tr"){
        $icerik="Web Sayfamıza Hoşgeldiniz.";
    }
    if($_SESSION["dil"]=="en"){
    $icerik="Welcome To Our Website.";
    }
}
else{
    $icerik="Web Sayfamıza Hoşgeldiniz.";
}
?>
```

7. Adım: Elde edilen içerik bilgisini sayfaya yazdırınız.

```
<div style="background-color:tomato;color:white; text-align:center;">
    <h2> <?php echo $icerik; ?>    </h2>
</div>
</body>
</html>
```

ÖLÇME VE DEĞERLENDİRME

Aşağıdaki soruları cevaplandırınız.

1. **Çerez yönetiminde bilgiler nerede saklanır?**

……
……

2. **Oturum mimarisinde bilgiler nerede saklanır?**

……
……

3. **setcookie(); fonksiyonunun parametreleri hangi değerleri alır?**

……
……

4. **Çerezlerin ve oturumların geçerlilik süreleri nasıl belirlenir?**

……
……

5. **Sayfadaki tüm oturum değişkenlerini kaldırmak için kullanılan fonksiyon hangisidir?**

……
……

6. **$_SESSION değişkeni kullanmak için her sayfada PHP kodlarının başına yazılan komut hangisidir?**

……
……

7. **Oluşturulan bir çerezi kaldırmak için gerekli kodu yazınız?**

……
……

8. **Çerez adı "sitedili", çerez içeriği "Kullanıcı Almanca biliyor", çerez kullanım ömrü 90 gün olacak bir çerezi oluşturma komutunu yazınız.**

……
……

9. **Bir kullanıcının web sayfasına giriş saati ve IP bilgilerini oturum değişkenine kaydeden PHP kodlarını yazınız.**

……
……

9. ÖĞRENME BİRİMİ

MySQL VERİ TABANI KULLANIMI



KONULAR

9.1. MySQL TEMEL KAVRAMLARI
9.2. UYGULAMA SUNUCUSUNUN KURULMASI VE YÖNETİLMESİ
9.3. PHP İLE VERİ TABANI BAĞLANTISI YAPMA
9.4. PHP İLE VERİ TABANI OLUŞTIRMA
9.5. PHP İLE TABLO OLUŞTURMA İŞLEMLERİ
9.6. PHP İLE MySQL VERİ TABANINDA VERİ EKLEME, SİLME, GÜNCELLEME İŞLEMLERİ
9.7. PHP İLE MySQL VERİ TABANINDA VERİ LİSTELEME İŞLEMLERİ

NELER ÖĞRENECEKSİNİZ?

- Veri tabanı temel kavramları
- MySQL veri tabanını yönetme
- MySQL sunucusuna bağlanma
- PHP ile MySQL veri tabanı oluşturma
- PHP ile veri tabanına veri ekleme, silme ve güncelleme yapma
- PHP ile verileri listeleme

TEMEL KAVRAMLAR

PHP, MySQL, SQL, phpMyAdmin, select, insert, delete, update, MySQLi, PDO.

MySQL

Hazırlık Çalışmaları

1. **MySQL veri tabanı farklı veri tabanları ile kıyaslandığında karşılaşılan avantajlar ve dezavantajlar nelerdir?**
2. **PHP ile en çok MySQL veri tabanının kullanılmasının sebepleri nelerdir?**

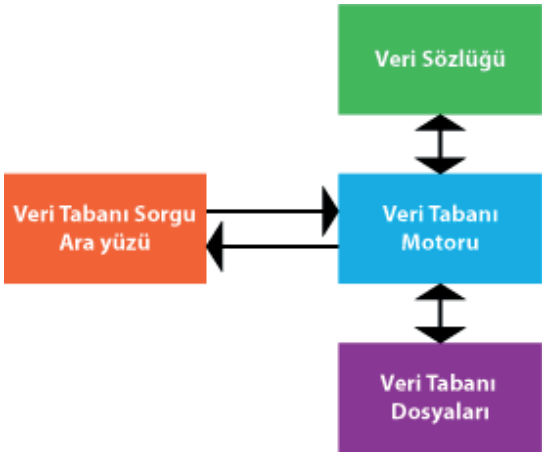
9.1. MySQL TEMEL KAVRAMLARI

Veri tabanı, bir bilgisayar sisteminde depolanan yapılandırılmış kayıtlar veya veriler topluluğudur. Verileri kullanıcılara göstermek veya kullanıcıdan alınan bilgileri depolamak için birçok dinamik web sitesi arka uç tarafında veri tabanı kullanılır. MySQL hızlı ve kolay bir şekilde web siteleri geliştirmek için kullanılan popüler bir veri tabanıdır. PHP ve MySQL veri tabanının uyum içinde çalışması en çok tercih edilme nedenlerindendir.

Bir veri tabanında bir grup veri dosyasından daha fazlası vardır. Birçok veri tabanında programlar, araçlar ve dosyalardan oluşan katmanlı bir yapı bulunur. Tüm bu yapıya DBMS (Database Management System – Veri Tabanı Yönetim Sistemi) denir. Veri tabanları temel olarak şu bölümlerden oluşur:

- Veri tabanı motoru
- Bir veya birden fazla veri tabanı dosyası
- Veri sözlüğü
- Sorgu dili arayüzü

Veri tabanı motoru bir veri tabanı yönetim sisteminin merkezidir. Veri tabanı dosyalarında saklanan verilerin erişim kontrolünü sağlar. Görsel 9.1'de bir veri tabanı yönetim sisteminin çalışması gösterilmiştir.



Görsel 9.1: Veri tabanı yönetim sistemi çalışma şeması

Veri tabanının nasıl çalıştığı, dosyalarda saklanacak verilerin türü ve veri tabanınının yapısı gibi bilgiler **Veri Sözlüğü** içinde bulunur.

MySQL içindeki SQL (Structured Query Language) ifadesi yapılandırılmış sorgu dilinden gelir. Bu dil birçok veri tabanının kullandığı İngilizceye dayalı bir dildir. Aşağıda örneği verilen ifade gibi veri tabanında işlem yapılmasını sağlar.

```
SELECT Baslik FROM Yazilar WHERE Yazar = 'AAAA BBBB';
```

Bir MySQL veri tabanı dosyası bir veya birden çok tablodan oluşur. Bir tablo sütunlar ve satırlardan oluşan ilişkili veriler topluluğudur. Bir sütun, tablodaki her kaydın belirli bilgilerini tutar. Satır ise tablo içinde birden fazla alan içeren veri kaydıdır.

Veri tabanında saklanacak veriler için tablo oluşturulduktan sonra tablo içindeki her bir sütunun veri türü belirlenmelidir. Veri tabanlarının çoğunda kullanılan üç temel tür vardır. Bu türler şunlardır:

- Sayı
- Metin
- Tarih ve saat

Bu temel türlerden MySQL içinde kullanılabilecek birçok çeşit bulunur. Doğru veri türünü seçmek sadece hangi verinin saklanacağını belirlemekle kalmaz, veri tabanının performansını da önemli ölçüde etkiler.

9.1.1. Sayı Veri Türleri

Bu türler sadece rakamdan oluşan sayısal değerleri saklar. İşaretli veya işaretsiz gibi özellik alabilirler.

INT: İşaretli veya işaretsiz tam sayıları saklamak için kullanılır. Eğer işaretli ise -2.147.483.648 ile 2.147.483.647 arasındaki sayıları, işaretsiz ise 0 ile 4.294.967.295 arasındaki tam sayıları saklar. En fazla 11 basamağa kadar saklanacak verinin basamak değeri belirlenebilir.

TINYINT: Çok küçük tam sayılar için kullanılır. İşaretli olarak -128 ile 127 arasında, işaretsiz olarak 0 ile 255 arasındaki tam sayıları saklar. En fazla 4 basamağa kadar saklanacak verinin basamak değeri belirlenebilir.

SMALLINT: Küçük tam sayılar için kullanılır. İşaretli olarak -32.768 ile 32.767 arasında, işaretsiz olarak 0 ile 65535 arasındaki tam sayıları saklar. En fazla 5 basamağa kadar saklanacak verinin basamak değeri belirlenebilir.

MEDIUMINT: Orta büyüklükteki tam sayılar için kullanılır. İşaretli olarak -8.388.608 ile 8.388.607 arasında, işaretsiz olarak 0 ile 16.777.215 arasındaki tam sayıları saklar. En fazla 9 basamağa kadar saklanacak verinin basamak değeri belirlenebilir.

BIGINT: Büyük boyutlardaki tam sayılar için kullanılır. İşaretli olarak -9.223.372.036.854.775.808 ile 9.223.372.036.854.775.807 arasında, işaretsiz olarak 0 ile 18.446.744.073.709.551.615 arasındaki tam sayıları saklar. En fazla 20 basamağa kadar saklanacak verinin basamak değeri belirlenebilir.

FLOAT(Toplam,Ondalık): Ondalık sayılar için kullanılır. İşaretsiz olarak ayarlanamaz. Tam ve ondalık sayıların basamak toplamı "Toplam" ile ondalık kısmın basamak sayısı ise "Ondalık" ile belirtilir. Örneğin 123.45 sayısı için float (5,2) olarak ayarlanmalıdır. Ön tanımlı olarak (10,2) ayarlıdır.

DOUBLE(Toplam,Ondalık): Ondalık sayılar için kullanılır. İşaretsiz olarak ayarlanamaz. Tam ve ondalık sayıların basamak toplamı "Toplam" ile ondalık kısmın basamak sayısı ise "Ondalık" ile belirtilir. Ön tanımlı olarak (16,4) ayarlıdır.

DECIMAL (Toplam,Ondalık): Paketlenmemiş ondalık sayılar için kullanılır. İşaretsiz olarak ayarlanamaz. Paketlenmemiş ondalık sayılarda her ondalık bir byte karşılık gelir.

9.1.2. Metin Veri Türleri

Metinsel verilerin saklandığı veri türleridir. Sadece Latin harflerinden oluşan metinler değil sayılar, özel karakterler ve farklı dillerin harfleri de bu veri türlerinde saklanır.

CHAR (Uzunluk): 1 ile 255 arasında karaktere sahip metinler için kullanılır. Ön tanımlı olarak 1 karakterdir. Belirlenen uzunluk kadar veri kaydeder. Eğer belirlenen uzunluktan daha az veri girilirse kalan karakterler boşluklar ile tamamlanır.

VARCHAR (Uzunluk): 0 ile 65.535 arasında karaktere sahip metinler için kullanılır. Ön tanımlı olarak 1 karakterdir. Belirlenen uzunluk kadar veri kaydeder. Eğer belirlenen uzunluktan daha az veri girilirse girilen karakter sayısının bir fazlası olarak hafızada yer kaplar.

BLOB veya TEXT: Maksimum uzunluğu 65.535 olan alanlardır. Uzunluk değerleri belirtilmez. Resim veya dosya gibi verileri saklamak için kullanılır. İkisi arasındaki fark BLOB'da küçük / büyük harfe karşı duyarlı, TEXT'de ise duyarlı değildir.

TINYBLOB veya TINYTEXT: Maksimum uzunluğu 255 karakter olan alanlardır. Uzunluk değerleri belirtilmez.

MEDIUMBLOB veya MEDIUMTEXT: Maksimum uzunluğu 16.777.215 karakter olan alanlardır. Uzunluk değerleri belirtilmez.

LONGBLOB veya LONGTEXT: Maksimum uzunluğu 4.294.967.295 karakter olan alanlardır. Uzunluk değerleri belirtilmez.

ENUM: Önceden tanımlanmış listeden veri eklemek için kullanılır. Örneğin cinsiyet bilgisinin ekleneceği bir alana "KADIN", "ERKEK" tanımlaması yapılır.

9.1.3. Tarih ve Saat Veri Türleri

DATE: YYYY-MM-DD formatında 1000-01-01 ile 9999-12-31 arasındaki tarihleri saklayan veri türüdür. Örneğin 19 Mayıs 1881 tarihi veri tabanına 1881-05-19 olarak kaydedilir.

DATETIME: YYYY-MM-DD HH:MM:SS formatında 1000-01-01 00:00:00 ile 9999-12-31 23:59:59 arasındaki tarih ve saatleri saklayan veri türüdür. Örneğin 10 Kasım 1938 tarihi 09:05:00 saati veri tabanına 1938-11-10 09:05:00 olarak kaydedilir.

TIMESTAMP: 1970-01-01 00:00:01 ile 2038-01-19 03:14:07 arasında her bir biti 1 saniyeye denk gelen 32 bit değerinde veri türüdür.

TIME: Saat:Dakika:Saniye cinsinden verileri kaydetmek için kullanılır.

Sıra Sizde

Aşağıda verilen bilgiler veri tabanına kaydedilmek istendiğinde hangi veri tipleri kullanılmalıdır?

"Kimlik Numarası, Adı Soyadı, Telefon Numarası, Elektronik Posta, Yaşı, Cinsiyeti, Doğum Tarihi, İş Başlama Saati, Maaşı, Adresi"

Değerlendirme

Çalışmalarınız aşağıda yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.

KONTROL LİSTESİ

Diğer sayfada listelenen ölçütlerden, öğrencide gözlenen davranış için Evet, gözlenemeyen davranış için Hayır kutucuğunun altına (X) işareti koyarak işaretleyiniz.

Ölçütler	Evet	Hayır
1. Tam sayı veri türü kullanıldı.		
2. Ondalık sayı veri türü kullanıldı.		
3. Tarih ve saat veri türleri kullanıldı.		
4. Metin veri türü kullanıldı.		

9.2. UYGULAMA SUNUCUSUNUN KURULMASI VE YÖNETİLMESİ

Kitabın birinci bölümünde MySQL veri tabanı sunucusunun kurulum işlemi gerçekleştirildi. Veri tabanı ile işlem yapmak için veri tabanı sunucuyu kurmak yeterli değildir. Veri tabanı oluşturmak, kayıt eklemek ve verileri almak için MySQL sunucusu ile iletişime geçmeye ihtiyaç duyulur. İletişime geçme işlemi için genelde mysql client, phpMyAdmin veya MySQL Workbench gibi uygulamalar kullanılır. Bu uygulamalar istemci olarak adlandırılır.

9.2.1. MySQL Komut Satırı Kullanımı

Kullanılan istemcilerden mysql client, MySQL ile birlikte kurulur. Grafiksel bir arayüze sahip olmasa da kullanımı kolaydır ve birçok işletim sistemleri ile birlikte çalışabilir. Komut istemi uygulamasını kullanarak erişim sağlanır. Komut istemi penceresinde XAMPP kullanılıyor ise **c:\xampp\mysql\bin**, MySQL sunucu yazılımı kullanılıyor ise yazılımın kurulu olduğu **c:\Program Files\MySQL\MySQL Server 8.0\bin** klasörüne geçiş yapılarak aşağıdaki örnek verilen komut dizesi yazılarak mysql client başlatılır.

```
mysql -u username -h hostname -p
```

- -u parametresi ile kullanıcı adı belirtilir.
- -h parametresi ile MySQL sunucusunun bulunduğu bilgisayarın adresi belirtilir. Yerel bilgisayarda çalışılıyor ise kullanılmayabilir.
- -p parametresi ile MySQL sunucusuna bağlanmak için kullanılan şifre belirtilir.

1. Uygulama

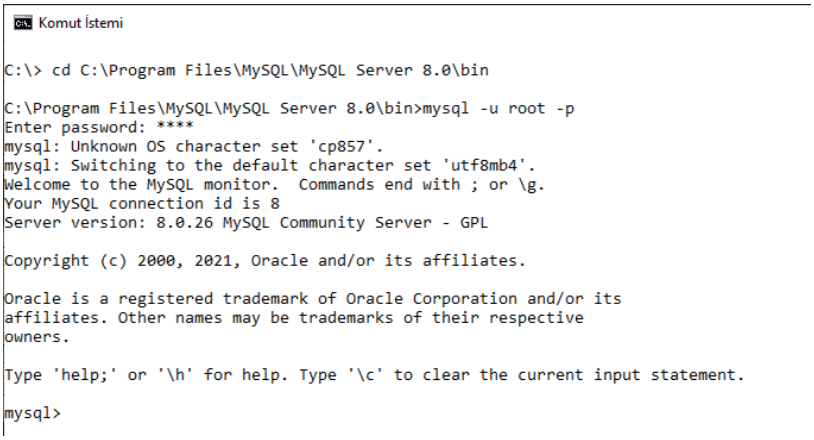
MySQL sunucusu ile komut satırı kullanıp iletişime geçme işlemini aşağıdaki adımları takip ederek gerçekleştiriniz.

1. Adım: Komut istemi uygulamasını açınız.

2. Adım: Görsel 9.2'deki gibi MySQL sunucusunun kurulu olduğu klasöre geçiş yaparak mysql istemcisini başlatınız.

Önemli

Kullanıcı adı ön tanımlı root olarak kullanılmıştır. Şifre kısmı için kurulum sırasında belirlenen şifreyi kullanınız.



Görsel 9.2: İstemci ile MySQL sunucusuna bağlanma

3. Adım: MySQL sunucunun durum bilgilerini almak için aşağıdaki komutu yazınız.

```
mysql> status
```

4. Adım: Komut satırından çıkmak için aşağıdaki komutu yazınız.

```
mysql> exit
```

2. Uygulama

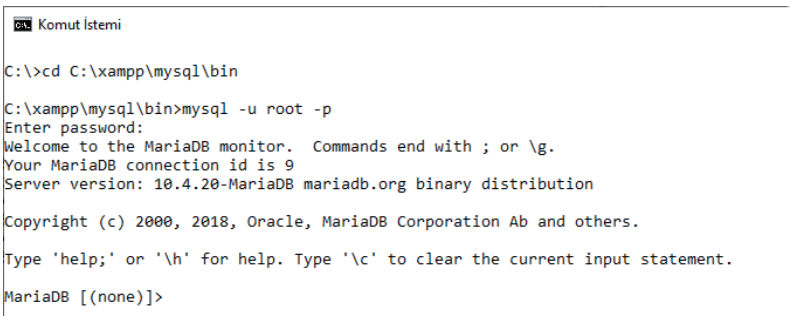
XAMPP paketi ile kurulu olan MySQL sunucusuna komut satırı kullanıp iletişime geçme işlemini aşağıdaki adımları takip ederek gerçekleştiriniz.

1. Adım: Komut istemi uygulamasını açınız.

2. Adım: Görsel 9.3'teki gibi MySQL sunucusunun kurulu olduğu klasöre geçiş yaparak mysql istemcisini başlatınız.

Önemli

Kullanıcı adı ön tanımlı root olarak kullanılmıştır. Şifre kısmı için kontrol panelinden belirlenen şifreyi kullanınız. Eğer şifre belirtilmemiş ise boş geçiniz.



Görsel 9.3: XAMPP paketi kurulu istemci ile MySQL sunucusuna bağlanma

3. Adım: Sunucunun durum bilgilerini almak için aşağıdaki komutu komut istemi penceresine yazınız.

```
MariaDB [(none)]> status
```

Önemli

XAMPP uygulamasında MySQL sunucu yerine MariaDB sunucusu kullanılmaktadır. MariaDB, kaynak kodu MYSQL geliştiricileri tarafından yazılan açık kaynak kodlu, ücretsiz veri tabanı sunucusudur. MariaDB ile MySQL komutları aynıdır.

3. Uygulama

MySQL client ile veri tabanı oluşturma işlemini aşağıdaki adımları takip ederek gerçekleştiriniz.

1. Adım: Komut istemi uygulamasını açınız.

2. Adım: MySQL istemcisini başlatmak için aşağıdaki komutu yazınız.

```
> mysql -u root -p
```

3. Adım: İstemciyi başlattıktan sonra aşağıdaki komutu kullanarak mevcut tüm veri tabanlarını listeleyiniz.

```
> SHOW DATABASES;
```

4. Adım: MySQL sunucusuna yeni bir veri tabanı eklemek için aşağıdaki komutu yazınız.

```
> CREATE DATABASE test_db;
```

5. Adım: Veri tabanlarını tekrar listeleyerek oluşturulan veri tabanının eklendiğini gözlemleyiniz.

6. Adım: Veri tabanını kaldırmak için aşağıdaki komutu yazınız.

```
> DROP DATABASE test_db;
```

4. Uygulama

Veri tabanına tablo ve tablonun içine alan oluşturma işlemini aşağıdaki adımları takip ederek gerçekleştiriniz.

1. Adım: Komut istemi uygulamasını açınız.

2. Adım: MySQL istemcisini başlatınız.

3. Adım: Aşağıdaki komutu kullanarak **demo_db** adında bir veri tabanı oluşturunuz.

4. Adım: Aşağıdaki komutu kullanarak işlem yapılacak veri tabanını seçiniz.

```
> USE demo_db;
```

5. Adım: Seçilen veri tabanına tablo ve alanları eklemek için aşağıdaki komutları yazınız.

```
> CREATE TABLE kisiler
  -> (
  -> id INT UNSIGNED NOT NULL AUTO_INCREMENT,
  -> ad VARCHAR(50) NOT NULL,
```

```
-> soyad VARCHAR(50) NOT NULL,
-> PRIMARY KEY (id)
-> );
```

6. Adım: Tablonun oluşturulup oluşturulmadığını kontrol etmek için aşağıdaki komutu yazınız.

```
> SHOW TABLES;
```

7. Adım: Oluşturulan tablonun bütün sütunlarının bilgilerini göstermek için aşağıdaki komutu yazınız.

```
> DESCRIBE kisiler;
```

9.2.2. phpMyAdmin Kullanımı

MySQL sunucusuyla çalışmanın bir diğer yolu web tabanlı phpMyAdmin aracıdır. phpMyAdmin aracı, tarayıcı üzerinden erişim sağlanan MySQL sunucusunun grafiksel bir arayüz ile yönetilmesini sağlayan web uygulamasıdır.

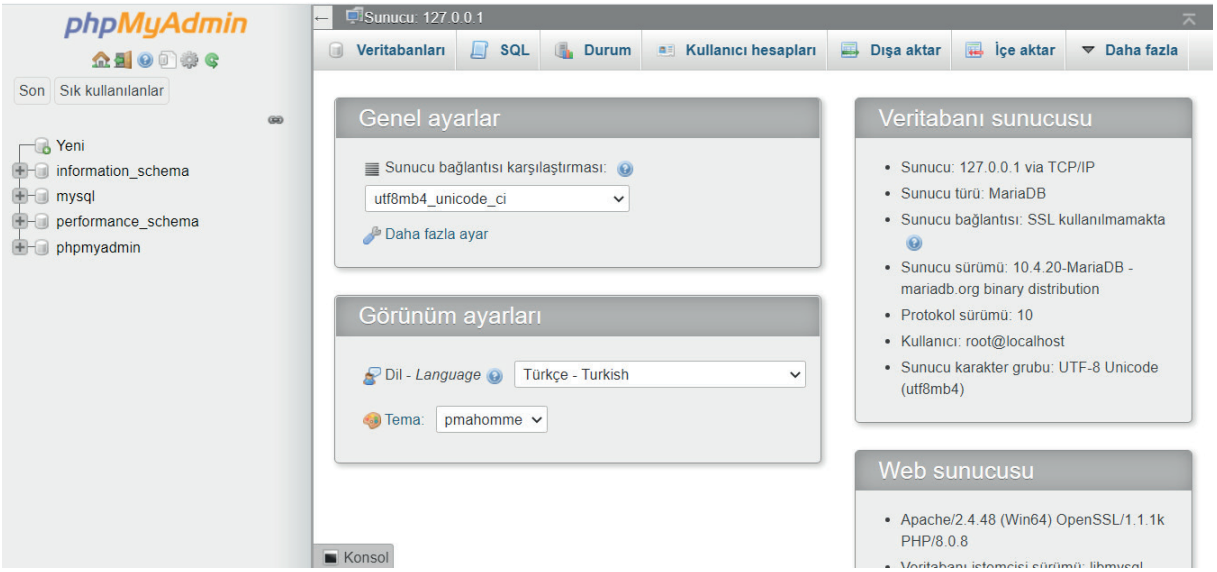
phpMyAdmin temel özellikleri şunlardır:

- Sezgisel web uygulamasıdır.
- MySQL veri tabanının çoğu özelliğini destekler.
- Veri tabanı, tablo, alan ve indeks üzerinde ekleme, silme, güncelleme, yeniden adlandırma veya silme işlemlerini yapar.
- Veri tabanı sunucusu, veri tabanı ve tabloların bakımını yapar.
- SQL ifadelerini veya toplu sorguları çalıştırır ve düzenler.
- MySQL kullanıcı hesabını ve haklarını yönetir.
- Saklı yordam (stored procedure) ve tetikleyicileri (trigger) yönetir.
- SQL ve CVS verilerini içe aktarır.
- Verileri SQL, CVS, XML, PDF, OpenDocument, Excel ve Word olarak dışa aktarır.

phpMyAdmin aracı XAMPP ile birlikte kurulur. Eğer XAMPP yerine Apache kurulumu yapıldı ise aşağıdaki adımları kullanarak phpMyAdmin uygulamasını kullanabilirsiniz.

1. Tarayıcının adres satırına http://www.phpmyadmin.net yazarak açılan sayfadan phpMyAdmin kurulum dosyasını indiriniz.
2. İndirilen sıkıştırılmış dosyayı Apache sunucusunun kurulu olduğu dizindeki htdocs klasörü içine çıkartınız.

phpMyAdmin uygulamasını başlatmak için tarayıcının adres satırına **http://localhost/phpmyadmin** adresi yazılır. Eğer Apache sunucusunun port numarası değiştirilmiş ise **http://localhost:8080/phpmyadmin** adresinde olduğu gibi port numarası URL adresine eklenir. Görsel 9.4'te phpMyAdmin arayüz sayfası gösterilmektedir.



Görsel 9.4: phpMyAdmin arayüz sayfası

phpMyAdmin ana sayfasının sol tarafında sunucudaki mevcut veri tabanları bulunur. En üstte bulunan menüde ise on iki farklı seçenek arasından işlem yapılmasına olanak sağlar. Bu seçenekler şunlardır:

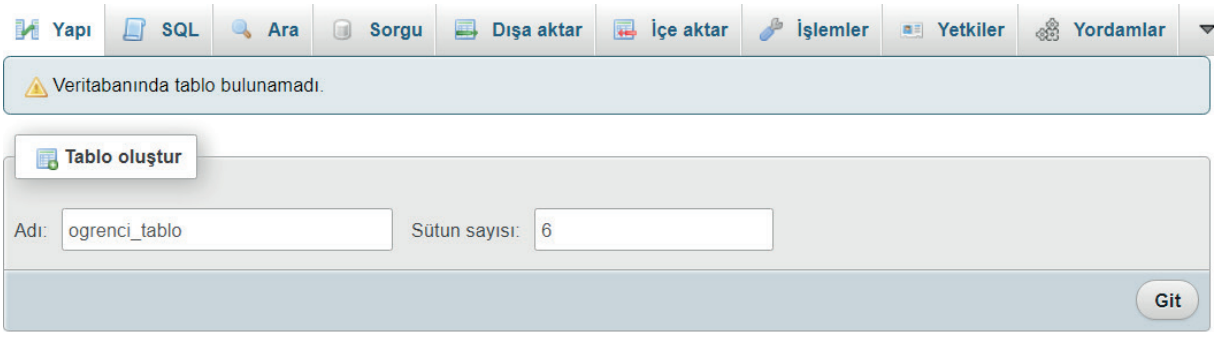
- **Veritabanları:** Veri tabanları oluşturur ve veri tabanlarını yönetir.
- **SQL:** SQL sorgularını sunucuda çalıştırır.
- **Durum:** Bağlantıların, sunucu işlemlerinin ve veri tabanı sorgularının durumunu görüntüler.
- **Kullanıcı hesapları:** Kullanıcı hesaplarının oluşturulmasını ve yönetilmesini sağlar.
- **Dışa Aktar:** Bir veya daha fazla veri tabanının yedeğinin oluşturulmasını sağlar.
- **İçe Aktar:** Veri tabanının geri yüklenmesini sağlar.
- **Ayarlar:** phpMyAdmin'in ayarlarını yönetir.
- **Kopya etme:** Asıl veya bağımlı sunucuyu kopyalamayı sağlar.
- **Değişkenler:** MySQL sunucusu yapılandırma ayarlarını yönetir.
- **Karakter Grupları:** Sunucu için kullanılabilen karakter gruplarını görüntüler.
- **Motorlar:** Sunucu için kullanılabilen depolama motorlarını görüntüler.
- **Eklentiler:** phpMyAdmin için kurulu olan eklentileri görüntüler.

phpMyAdmin arayüzü ile veri tabanı oluşturma işlemi üst tarafta bulunan **Veritabanları** sekmesine tıklanarak açılan sayfada gerçekleştirilir. Görsel 9.5'te gösterildiği gibi veri tabanı adı ve kullanılacak sıralama seti (collation) belirlenerek **Oluştur** butonu ile veri tabanı oluşturulur.



Görsel 9.5: phpMyAdmin ile veri tabanı oluşturma sayfası

phpMyAdmin arayüzü ile veri tabanı içine verilerin kaydedileceği tablo oluşturmak için işlem yapılacak veri tabanı seçilir. Görsel 9.6'daki gibi tablo adı ve tabloda bulunacak sütunlar belirlendikten sonra Git butonuna tıklanarak tablo oluşturulur.



Görsel 9.6: phpMyAdmin ile tablo oluşturma

Tablo oluşturulduktan sonra tablo içinde verilerin kaydedileceği alanlar oluşturulur. Görsel 9.7'de kaydedilecek verilere göre alan adları ve özellikleri belirlendikten sonra **Kaydet** butonu ile alanlar oluşturulur.



Görsel 9.7: phpMyAdmin ile tablo alanlarını oluşturma

Önemli

Tablo içindeki her kaydın benzersiz bir kayıt numarası olması için ogrenci_id alanına **PRIMARY** özelliği verilmelidir. Bu benzersiz alanın değerlerinin otomatik olarak verilmesi için **A_I** (Auto Increment – Otomatik Artırım) kutucuğu işaretlenmelidir.

Alanlar oluşturulduktan sonra **Yapı** sekmesine tıklanarak Görsel 9.8'deki gibi alanlar üzerinde düzenleme işlemi yapılabilir.



Görsel 9.8: phpMyAdmin ile alanları düzenleme

Bir veri tabanını silmek için Veritabanları sekmesine tıklanır. Açılan sayfada Görsel 9.9'daki gibi silinecek olan veri tabanının yanındaki kutucuk seçildikten sonra **Kaldır** butonuna tıklanarak silme işlemi gerçekleştirilir.



Görsel 9.9: phpMyAdmin ile veri tabanı silme

9.2.3. Tablo İlişkileri

İlişkisel bir veri tabanında, farklı tablolara ait satırların ilişkilendirilmesiyle bir ilişki oluşturulur. Tablo ilişkileri bir tablonun birincil anahtar (primary key) alanı ile diğer tablonun yabancı anahtar (foreign key) alanı arasında oluşturulur. Tablo ilişkilendirmede MySQL komut satırı ve phpMyAdmin SQL sekmesi kullanılabilir.

Tablolar arasında üç tür ilişki bulunur. Bu ilişkiler şunlardır:

- Bire-Bir (One to One)
- Bire-Çok (One to Many)
- Çoka-Çok (Many to Many)

Bire-Bir İlişki: A tablodaki her bir kayıt ile B tablodaki sadece bir kayıt eşleşiyor ise bu ilişki türü bire-bir ilişkidir. Aşağıdaki sql kodlamalarında öğrenci ve öğrencilere ait adres bilgilerinin saklandığı tablolar oluşturulmuştur. Her iki tablonun da birincil anahtar isimleri aynıdır. Adres tablosunun yabancı anahtarı öğrenci tablosunun birincil anahtarı olarak ayarlanarak eşleştirme işlemi gerçekleşmiştir.

```
CREATE TABLE ogrenci(
   ogrenci_id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
   adsoyad VARCHAR(50)
   );
CREATE TABLE adres(
   ogrenci_id INT NOT NULL PRIMARY KEY ,
  adres VARCHAR(50),
   FOREIGN KEY (ogrenci_id) REFERENCES ogrenci (ogrenci_id)
   );
```

Bire-Çok İlişki: A tablodaki her bir veri B tabloda birden fazla kayıt ile eşleşiyor ise bu ilişki türü bire-çok ilişkidir. Aşağıdaki sql kodlamalarında öğrenci ve sınıf bilgilerinin saklandığı tablolar oluşturulmuştur. Öğrenci tablosundaki sinif_id alanı yabancı anahtar olarak ayarlanarak eşleştirme işlemi gerçekleşmiştir.

```
CREATE TABLE sinif (
 sinif_id int AUTO_INCREMENT PRIMARY KEY,
 sinif_ad varchar(10) NOT NULL
);
```

```
CREATE TABLE ogrenci(
   ogrenci_id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
   adsoyad VARCHAR(50),
   sinif_id int NOT NULL,
   FOREIGN KEY (sinif_id) REFERENCES sinif(sinif_id)
   );
```

Çoka-Çok İlişki: A tablodaki her bir veri B tabloda birden fazla kayıt ile eşleşiyor ve B tablosundaki her bir kayıt A tablosundaki birden fazla kayıt ile eşleşiyor ise bu ilişki türü çoka-çok ilişkidir. Çoka-çok ilişki türünde araya bir bağlantıyı sağlayacak bir tablo oluşturulması gerekir.

```
CREATE TABLE ogrenci(
    ogrenci_id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    adsoyad VARCHAR(50)
);
```

```
CREATE TABLE ogretmen(
    ogretmen_id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
```

```
    adsoyad VARCHAR(50)
);
```

```
CREATE TABLE ogretmen_ogrenci(
   ogretmen_id INT NOT NULL,
   ogrenci_id INT NOT NULL,
   FOREIGN KEY (ogretmen_id) REFERENCES ogretmen (ogretmen_id),
   FOREIGN KEY (ogrenci_id) REFERENCES ogrenci (ogrenci_id)
   );
```

9.3. PHP İLE VERİ TABANI BAĞLANTISI YAPMA

MySQL veri tabanındaki verileri depolamak veya verilere erişmek için ilk önce MySQL sunucusuna bağlanmak gerekir. PHP ile MySQL sunucusuna bağlanmak için üç temel yöntem bulunur. Bu yöntemler şunlardır:

- MySQL arabirimi
- MySQLi arabirimi
- PHP Data Object (PDO) uzantısı

MySQLi (MYSQL Improved- Geliştirilmiş MySQL), PHP 5 sürümünden önce kullanılan MySQL arabiriminin geliştirilmiş hâlidir. MySQL ve MySQLi arabirimleri kullanım olarak benzerlik gösterir. MySQLi adından da anlaşılacağı gibi sadece MySQL veri tabanını destekler. MySQLi veri tabanı kolaylıkla bağlanmaya ve sorgu çalıştırmaya olanak sağlar.

MySQLi arabiriminin özellikleri şunlardır:

- Nesne Tabanlı Arabirim (Object-Oriented Interface)
- Ön Hazırlıklı Sorgular (Prepared Statements)
- Çoklu Sorgular (Multi Query)

MySQL arabirimi ile veri tabanına bağlanmak için **mysqli_connect()** fonksiyonu kullanılır. PHP ile MySQL arasındaki bütün iletişim bu bağlantı üzerinden gerçekleşir. Prosedürel ve nesne yönelimli olmak üzere iki farklı kullanımı vardır. Prosedürel kullanımı aşağıdaki gibidir.

```
mysqli_connect("sunucu adı", "kullanıcı adı", "kullanıcı şifre", "veri tabanı adı");
```

MySQLi arabirimi ile veri tabanına bağlanmak için aşağıdaki PHP kodlamaları kullanılır.

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$database = "veritabanı";
$baglanti= mysqli_connect($servername, $username, $password, $database);
if ($baglanti === false)
{
   die("Bağlantı Hatası:" . mysqli_connect_error());
}
echo "Bağlantı Başarılı";
?>
```

MySQLi arabirimi nesne yönelimli programlama ile veri tabanına bağlanmak için aşağıdaki kodlamalar kullanılır.

```
<?php
$servername = “localhost”;
$username = “root”;
$password = “”;
$database = “veritabanı”;
$baglanti= new mysqli ($servername, $username, $password, $database);

if($baglanti === false){
    die(“Bağlantı Hatası:” . $ baglanti ->connect_error);
}
echo “Bağlantı Başarılı”;
?>
```

MySQLi arabirimi kullanılarak bağlantı işlemlerinde kullanılan ifadeler şunlardır:

- **servername:** Veri tabanı sunucusunun adıdır. Yerel bilgisayarda çalışılıyor ise localhost, uzak bilgisayarda çalışılıyor ise IP adresi veya alan adı girilmelidir.
- **username:** Veri tabanı sunucusu oluşturulurken kullanılan kullanıcı adıdır.
- **password:** Veri tabanı sunucusu oluşturulurken kullanılan şifredir.
- **database:** Kullanılacak olan veri tabanının adıdır.

PDO (PHP Data Object – PHP Veri Nesneleri), MySQLi arabiriminden farklı olarak birden çok veri tabanını destekleyen nesne yönelimli bir arabirimdir. PDO komutları birkaç değişiklik ile farklı veri tabanlarında da kullanılabilir.

PDO ile veri tabanına bağlanmak için aşağıdaki kodlarda gösterildiği gibi DSN (Data Source Name – Veri Kaynağı Adı) adı verilen bağlantı yapısı kullanılır.

```
$servername = “localhost”;
$username = “root”;
$password = “”;
$database = “veritabanı”;
try
{
  $baglanti= new PDO(“mysql:host=$servername;dbname=$database”,$username,
$password);
  $baglanti->setAttribute(PDO::ATTR_ERRMODE,PDO::ERRMODE_EXCEPTION);
  echo “Bağlantı Başarılı”;
}
catch (PDOException $e)
{
    die(“Bağlantı Hatası:” . $e->getMessage());
}
?>
```

Önemli

setAttribute özniteliği bağlantı sırasında hata meydana geldiğinde hatayı yakalamak için kullanılır. Yukarıdaki tanımlanan özniteliklerde bağlantı sırasında hata meydana geldiğinde hatayı yakalamak için kullanılır.

PHP kodlarının yürütülmesi bittiğinde MySQL sunucusu ile bağlantısı otomatik olarak kapatılır. Bununla birlikte yazılımcı tarafından kapatılmak istenir ise aşağıdaki kodlar kullanılır.

```
// MySQLi prosedürel
mysqli_close($baglanti);
// MySQLi nesnel
$mysqli->close();
// PDO
unset($pdo);
```

9.4. PHP İLE VERİ TABANI OLUŞTURMA

PHP ile veri tabanı sunucusuna bağlantı yapıldıktan sonra bir veri tabanı oluşturmak için MySQLi veya PDO kullanılır.

Aşağıdaki kodlamalarda MySQLi prosedürel yöntem ile veri tabanı oluşturma işlemi gösterilmiştir.

```
<?php

$baglanti = mysqli_connect("localhost", "root", "");

if($baglanti === false)
{
    die("Bağlantı Hatası:" . mysqli_connect_error());
}

$sorgu = "CREATE DATABASE test";
if(mysqli_query($baglanti, $sorgu))
{
    echo "Veri tabanı oluşturuldu.";
}
Else
{
    echo "Hata:" . mysqli_error($baglanti);
}

mysqli_close($baglanti);
?>
```

Aşağıdaki kodlamalarda MySQLi nesne yönelimli yöntem ile veri tabanı oluşturma işlemi gösterilmiştir.

```php
<?php
$baglanti = new mysqli("localhost", "root", "");
if($baglanti === false){
    die("Bağlantı Hatası:" . $baglanti->connect_error);
}
$sorgu = "CREATE DATABASE test";
if($baglanti->query($sorgu) === true)
{
    echo "Veri tabanı oluşturuldu.";
}
else
{
    echo "Hata:" . $baglanti->error;
}
$baglanti->close();
?>
```

Aşağıdaki kodlamalarda PDO yöntemi ile veri tabanı oluşturma işlemi gösterilmiştir.

```php
<?php
try
{
  $baglanti = new PDO("mysql:host=localhost;", "root", "");
}
catch(PDOException $e)
{
  die("Bağlantı Hatası:" . $e->getMessage());
}
Try
{
    $sorgu = "CREATE DATABASE test";
    $baglanti->exec($sorgu);
    echo "Veri tabanı oluşturuldu.";
}
catch(PDOException $e)
{
    die("Hata:" . $e->getMessage());
}
unset($baglanti);
?>
```

9.5. PHP İLE TABLO OLUŞTURMA İŞLEMLERİ

İlişkisel veri tabanlarında tablolar verileri saklayan kısımlardır. Bir tablo, bilgileri satırlar ve sütunlar hâlinde düzenler.

Aşağıdaki PHP kodlarında MySQLi prosedürel yöntem ile tablo oluşturma işlemi gerçekleştirilmiştir.

```
<?php
$baglanti = mysqli_connect("localhost", "root", "", "test");

if($baglanti === false){
    die("Bağlantı Hatası:" . mysqli_connect_error());
}

$sorgu = "CREATE TABLE test_tablo(
    id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
    ad VARCHAR(20) NOT NULL,
    soyad VARCHAR(20) NOT NULL,
    email VARCHAR(60) NOT NULL UNIQUE
)";
if(mysqli_query($baglanti, $sorgu)){
    echo "Tablo oluşturuldu.";
} else{
    echo "Hata :" . mysqli_error($baglanti);
}
mysqli_close($baglanti);
?>
```

Aşağıdaki PHP kodlarında MySQLi nesne yönelimli yöntem ile tablo oluşturma işlemi gerçekleştirilmiştir.

```
<?php
$baglanti = new mysqli("localhost", "root", "", "test");
if($baglanti === false){
    die("Bağlantı Hatası:" . $baglanti->connect_error);
}
$sorgu = "CREATE TABLE test_tablo(
  id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
  ad VARCHAR(20) NOT NULL,
  soyad VARCHAR(20) NOT NULL,
  email VARCHAR(60) NOT NULL UNIQUE
)";
if($baglanti->query($sorgu) === true){
    echo "Tablo oluşturuldu.";
} else{
    echo "Hata :" . $baglanti->error;
}
$baglanti->close();
?>
```

Aşağıdaki PHP kodlarında PDO yöntemi ile tablo oluşturma işlemi gerçekleştirilmiştir.

```
<?php
try
{
    $baglanti = new PDO("mysql:host=localhost;dbname=test", "root", "");
}
catch(PDOException $e)
{
    die("Bağlantı Hatası:" . $e->getMessage());
}

try
{
  $sorgu = "CREATE TABLE test_tablo(
    id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
    ad VARCHAR(20) NOT NULL,
    soyad VARCHAR(20) NOT NULL,
    email VARCHAR(60) NOT NULL UNIQUE
  )";

  $baglanti->exec($sorgu);
  echo "Tablo oluşturuldu.";
} catch(PDOException $e)
{
    die("Hata :" . $e->getMessage());
}
unset($baglanti);
?>
```

9.6. PHP İLE MySQL VERİ TABANINDA VERİ EKLEME, SİLME VE GÜNCELLEME İŞLEMLERİ

Veri tabanındaki bir tabloya veri ekleme, silme ve değiştirme işlemi için SQL ifadeleri kullanılır. Kitabın bu bölümünde MySQLi arabirimi kullanılarak tablolar üzerinde işlem yapılacaktır.

9.6.1. Veri Ekleme

Veri tabanında bulunan bir tabloya kayıt eklemek için **INSERT INTO** sorgu ifadesi kullanılır. Tabloya veriyi eklemek için PHP'nin mysqli_query() fonksiyonu kullanılarak sorgu ifadesi çalıştırılır. Temel kullanımı aşağıdaki gibidir.

```
INSERT INTO tabloadı (sütun1, sütun2, sütun3) VALUES (değer1, değer2, değer3);
```

Aşağıdaki PHP kodlarında MySQLi prosedürel yöntem kullanılarak tabloya veri ekleme işlemi gerçekleştirilmiştir.

```
<?php
$baglanti = mysqli_connect("localhost", "root", "", "test");
if($baglanti === false){
    die("Bağlantı Hatası:" . mysqli_connect_error());
}

$sorgu = "INSERT INTO test_tablo (ad, soyad, email) VALUES
          ('aaaa', 'bbbb', 'cccc@mail.com')";
if(mysqli_query($baglanti, $sorgu)){
    echo "Kayıt başarıyla eklendi.";
} else{
    echo "Hata:" . mysqli_error($baglanti);
}
mysqli_close($baglanti);
?>
```

Aşağıdaki PHP kodlarında MySQLi prosedürel yöntem kullanılarak tabloya çoklu veri ekleme işlemi gerçekleştirilmiştir.

```
<?php

$baglanti = mysqli_connect("localhost", "root", "", "test");

if($baglanti === false){
    die("Bağlantı Hatası:" . mysqli_connect_error());
}

$sorgu = "INSERT INTO test_tablo (ad, soyad, email) VALUES
            ('AAAA', 'BBBB', 'aaaa@mail.com'),
            ('CCCC', 'DDDD', 'cccc@mail.com'),
            ('EEEE', 'FFFF', 'eeee@mail.com'),
            ('GGGG', 'HHHH', 'gggg@mail.com')";

if(mysqli_query($baglanti, $sorgu)){
    echo "Kayıtlar başarıyla eklendi.";
} else{
    echo "Hata:" . mysqli_error($baglanti);
}
mysqli_close($baglanti);
?>
```

Veri ekleme sorgularında direkt veri değerlerini yazmak yerine parametreler kullanılabilir. Bu parametreler sorgu ifadelerindeki yer tutuculardır. Sorgu çalıştırıldığında parametreler gerçek değer ile yer değiştirir. Kullanımı aşağıda gösterilmiştir.

```
INSERT INTO tabloadı (sütun1, sütun2, sütun3) VALUES (?, ?, ?);
```

Sorgu ifadelerinde parametre kullanımı belirli bir sorgu ifadesinin birden çok çalıştırılmasında ve SQL injection saldırılarına karşı önlem alınmasında kullanılır. Aşağıdaki kodlamalarda parametre kullanılarak veri ekleme işlemi gerçekleştirilmiştir.

```
<?php
$baglanti = mysqli_connect("localhost", "root", "", "test");
if($baglanti === false){
    die("Bağlantı Hatası:" . mysqli_connect_error());
}
// Veri ekleme ifadesinin hazırlanması
$sorgu = "INSERT INTO test_tablo (ad, soyad, email) VALUES (?, ?, ?)";

//Sorguda hata olup olmadığının denetimi
if($stmt = mysqli_prepare($baglanti, $sorgu))
{
    // Değişkenlerin sorgu parametrelerine bağlanması
    mysqli_stmt_bind_param($stmt, "sss", $ad, $soyad, $email);

    $ad = "AAAA";
    $soyad = "BBBB";
    $email = "AAAA@mail.com";
    mysqli_stmt_execute($stmt);

    $ad = "CCCC";
    $soyad = "DDDD";
    $email = "CCCC@mail.com";
    mysqli_stmt_execute($stmt);

    echo "Kayıtlar başarıyla eklendi";
} else{
    echo "Hata:" . mysqli_error($baglanti);
}

mysqli_stmt_close($stmt);
mysqli_close($baglanti);
?>
```

9.6.2. Veri Silme

Veri tabanından kayıt silmek için **DELETE** sorgu ifadesi kullanılır. Genellikle DELETE ifadesi hangi kaydın silineceğinin belirlenmesi için WHERE yan cümlesi ile birlikte kullanılır. Aksi takdirde veri tabanındaki bütün kayıtlar silinir.

DELETE ifadesinin temel kullanımı aşağıda verilmiştir.

```
DELETE FROM tabloadı WHERE sütunadı=değer
```

Aşağıdaki PHP kodlarında kayıt silme işlemi gerçekleştirilmiştir.

```
<?php

$baglanti = mysqli_connect("localhost", "root", "", "test");

if($baglanti === false){
    die("Bağlantı Hatası:" . mysqli_connect_error());
}

$sorgu = "DELETE FROM test_tablo WHERE id=1";

if(mysqli_query($baglanti, $sorgu)){
    echo "Kayıt başarıyla silindi.";
} else{
    echo "Hata:" . mysqli_error($baglanti);
}
mysqli_close($baglanti);
?>
```

9.6.3. Veri Güncelleme

Bir veri tabanı tablosundaki mevcut kayıtları değiştirmek için **UPDATE** ifadesi kullanılır. Genellikle UPDATE ifadesi hangi kaydın güncelleneceğinin belirlenmesi için WHERE yan cümlesi ile birlikte kullanılır. Aksi takdirde veri tabanındaki bütün kayıtlar güncellenir.

UPDATE ifadesinin temel kullanımı aşağıda verilmiştir.

```
UPDATE tabloadı SET sütun1=değer1, sütun2=değer2 WHERE sütunadı=değer
```

Aşağıdaki PHP kodlarında kayıt güncelleme işlemi gerçekleştirilmiştir.

```
<?php
$baglanti = mysqli_connect("localhost", "root", "", "test");

if($baglanti === false){
    die("Bağlantı Hatası:" . mysqli_connect_error());
}
$sorgu = "UPDATE test_tablo SET email='ZZZZ@mail.com' WHERE id=1";

if(mysqli_query($baglanti, $sorgu)){
    echo "Kayıt başarıyla güncellendi.";
} else {
    echo "Hata:" . mysqli_error($baglanti);
}
mysqli_close($baglanti);
?>
```

9.7. PHP İLE MySQL VERİ TABANINDA VERİ LİSTELEME İŞLEMLERİ

PHP ile veri tabanı tablosundaki verileri almak için **SELECT** ifadesi kullanılır.

Tablo içindeki belirli alanların verilerini almak için aşağıdaki sorgu yapısı kullanılır.

```
SELECT sütun1, sütun2, sütunN FROM tabloadı;
```

Tablo içindeki tüm alanların verilerini almak için aşağıdaki sorgu yapısı kullanılır.

```
SELECT * FROM tabloadı;
```

Aşağıdaki PHP kodlarında kayıt listeleme işlemi gerçekleştirilmiştir.

```
<?php
$baglanti = mysqli_connect("localhost", "root", "", "test");
if($baglanti === false){
    die("Bağlantı Hatası:" . mysqli_connect_error());
}
$sorgu = "SELECT * FROM test_tablo";
if($veri = mysqli_query($baglanti, $sorgu)){
    if(mysqli_num_rows($veri) > 0){
        echo "<table>";
            echo "<tr>";
                echo "<th>id</th>";
                echo "<th>Ad</th>";
                echo "<th>Soyad</th>";
                echo "<th>Email</th>";
            echo "</tr>";
        while($kayit = mysqli_fetch_array($veri)){
            echo "<tr>";
                echo "<td>" . $kayit['id'] . "</td>";
                echo "<td>" . $kayit['ad'] . "</td>";
                echo "<td>" . $kayit['soyad'] . "</td>";
                echo "<td>" . $kayit['email'] . "</td>";
            echo "</tr>";
        }
        echo "</table>";
        //veri seti temizleniyor.
        mysqli_free_result($veri);
    } else{
        echo "Kayıt bulunamadı";
    }
} else{
    echo "Hata:" . mysqli_error($baglanti);
}
mysqli_close($baglanti);
?>
```

Önemli

Kullanılan mysqli_fetch_array fonksiyonu veri tabanından alınan kayıtları bir dizi hâline dönüştürmek için kullanılır.

5. Uygulama

Temel bir haber sitesinin yönetim işlemlerinde kullanılacak olan veri tabanı, tablo ve tablo içindeki alanların oluşturulması işlemini aşağıdaki adımları takip ederek gerçekleştiriniz.

1. Adım: MySQL sunucusuna aşağıdaki sql kodlarını MySQL komut satırı veya phpMyAdmin arayüzünü kullanarak **haberler_db** adında bir veri tabanı oluşturunuz.

```
CREATE DATABASE IF NOT EXISTS haberler_db DEFAULT CHARACTER SET utf8 DEFAULT
COLLATE utf8_turkish_ci;
```

2. Adım: Oluşturulan veri tabanı içine aşağıdaki sql kodlarını kullanarak **yazar_tablo** adında tablo oluşturunuz.

```
CREATE TABLE yazar_tablo(
    yazar_id int PRIMARY KEY NOT NULL AUTO_INCREMENT,
    yazar_eposta VARCHAR(50) NOT NULL,
    yazar_sifre VARCHAR(50) NOT NULL,
    yazar_adsoyad VARCHAR(50) NOT NULL
    );
```

3. Adım: Oluşturulan veri tabanı içine aşağıdaki sql kodlarını kullanarak **kategori_tablo** adında tablo oluşturunuz.

```
CREATE TABLE kategori_tablo(
    kategori_id INT PRIMARY KEY NOT NULL AUTO_INCREMENT,
    kategori_adi VARCHAR(20) NOT NULL
    );
```

4. Adım: Oluşturulan veri tabanı içine aşağıdaki sql kodlarını kullanarak **haber_tablo** adında tablo oluşturunuz.

```
CREATE TABLE haber_tablo(
    haber_id INT PRIMARY KEY NOT NULL AUTO_INCREMENT,
    kategori_id INT NOT NULL,
    haber_baslik VARCHAR(100) NOT NULL,
    haber_icerik TEXT,
    yazar_id INT NOT NULL,
    haber_tarih DATETIME DEFAULT CURRENT_TIMESTAMP(),
    FOREIGN KEY (kategori_id) REFERENCES kategori_tablo (kategori_id),
    FOREIGN KEY (yazar_id) REFERENCES yazar_tablo (yazar_id)
    );
```

Önemli

Kategori ve yazarlar tablosu yabancı anahtarlar üzerinden haberler tablosu ile ilişkilendirilmiştir.

6. Uygulama

Haber sitesinin yönetimine dosya ve dizinler oluşturmak için aşağıdaki adımları takip ediniz.

1. Adım: Web sunucusunun kurulu olduğu htdocs klasörü içine **habersite** adında bir klasör oluşturunuz.

2. Adım: Habersite klasörünün içine **admin** adında bir klasör oluşturunuz. Admin klasörü içine aşağıdaki listede belirtilen dosyaları oluşturunuz.

- **index.php**, haber sitesinin başlangıç sayfasıdır.
- **baglanti.php**, veri tabanına bağlantı işlemlerini MySQLi prosedürel yöntem ile gerçekleştirir.
- **baglanti2.php**, veri tabanına bağlantı işlemlerini MySQLi nesne yönelimli yöntem ile gerçekleştirir.
- **baglanti3.php**, veri tabanına bağlantı işlemlerini PDO yöntemi ile gerçekleştirir.
- **haberleroop.php**, bir sınıf oluşturarak sınıf içindeki özellik ve fonksiyonlarla veri tabanı işlemlerini gerçekleştirir.
- **kategoriler.php**, kategorileri listeleyerek kategori ekleme, silme ve güncelleme sayfalarına yönlendirir.
- **kategoriekle.php**, kategori ekleme işlemini gerçekleştirir.
- **kategorisil.php**, kategori silme işlemini gerçekleştirir.
- **kategoriduzenle.php**, kategori güncelleme işlemini gerçekleştirir.
- **yazarlar.php**, yazarları listeleyerek yazar ekleme, silme ve güncelleme sayfalarına yönlendirir.
- **yazarekle.php**, yazar bilgilerini ekleme işlemini gerçekleştirir.
- **yazarsil.php**, yazar bilgilerini silme işlemini gerçekleştirir.
- **yazarduzenle.php**, yazar bilgilerini güncelleme işlemini gerçekleştirir.
- **haberler.php,** haberleri listeleyerek haber ekleme, silme ve güncelleme sayfalarına yönlendirir.
- **haberekle.php**, haber bilgilerini ekleme işlemini gerçekleştirir.
- **habersil.php**, haber bilgilerini silme işlemini gerçekleştirir.
- **haberduzenle.php**, haber bilgilerini düzenleme işlemini gerçekleştirir.
- **login.php**, oturum açma işlemlerini gerçekleştirir.
- **logout.php**, oturum sonlandırma işlemlerini gerçekleştirir.

3. Adım: habersite klasörünün içine includes adında klasör oluşturunuz. includes klasörü içine aşağıdaki listede belirtilen dosyaları oluşturunuz.

- **header.php**, HTML sayfasının header kodlarıdır.

- **footer.php**, HTML sayfasının footer kodlarıdır.
- **navbaradmin.php**, admin sayfasının gezinti menüsüdür.

7. Uygulama

Sitenin genelinde tekrarlanan HTML kodlarını yazarak ihtiyaç duyulan yerlerde include fonksiyonu ile sayfalara dâhil etme işlemini aşağıdaki adımları takip ederek gerçekleştiriniz.

1. Adım: includes klasörü içindeki header.php dosyasını açarak aşağıdaki kodlamaları yapınız.

```
<html>
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width" ,initial-scale="1">
    <title>Haber Sitesi</title>
    <link rel="stylesheet" type="text/css" href="https://cdn.jsdelivr.net/npm/
bootstrap@5.1.0/dist/css/bootstrap.min.css" rel="stylesheet">
 </head>
<body class="d-flex flex-column h-100">
```

2. Adım: includes klasörü içindeki footer.php dosyasını açarak aşağıdaki kodlamaları yapınız.

```
          </div>
        </div>
    </div>
    <footer class="footer mt-auto py-3 bg-light">
      <div class="container">
        <span class="text-muted">Bilişim Teknolojileri</span>
      </div>
    </footer>
  </body>
</html>
```

3. Adım: includes klasörü içindeki navbaradmin.php dosyasını açarak aşağıdaki kodlamaları yapınız.

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <div class="container-fluid">
    <a class="navbar-brand" href="#">Haberler Yönetim</a>
    <button class="navbar-toggler" type="button"
data-bs-toggle="collapse" data-bs-target="#navbarNav"
aria-controls="navbarNav" aria-expanded="false"
aria-label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNav">
      <ul class="navbar-nav">
        <li class="nav-item">
```

```
          <a class="nav-link" href="kategoriler.php">Kategoriler</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="haberler.php">Haberler</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="yazarlar.php">Yazarlar</a>
        </li>
      </ul>
    </div>
  </div>
</nav>
<div class="container mt-4">
    <div class="row">
        <div class="col-lg-8">
```

4. Adım: Admin klasörü içindeki index.php dosyasına aşağıdaki kodlamaları yapınız.

```
<?php
include ('baglanti.php');
include ('../includes/header.php');
include ('../includes/navbaradmin.php');
?>
<h1> Haber Sitesi Yönetim Paneli<h1>
<h2> Hoş Geldiniz</h2>
<?php include ('../includes/footer.php'); ?>
```

8. Uygulama

Kategori ekleme, silme, güncelleme ve listeleme işlemlerini MySQLi prosedürel yöntem ile gerçekleştirmek için aşağıdaki adımları takip ediniz.

1. Adım: Veri tabanına bağlantı için aşağıdaki PHP kodlarını admin klasörü içindeki baglanti.php dosyasına yazınız.

```
<?php
$servername = "localhost";
$username   = "root";
$password   = "";
$database   = "haberler_db";
$baglanti= mysqli_connect($servername, $username, $password, $database);
if ($baglanti === false)
{
    die("Bağlantı Hatası:" . mysqli_connect_error());
}
?>
```

2. Adım: Kategori bilgilerini eklemek için aşağıdaki PHP kodlarını admin klasörü içindeki kategoriekle.php dosyasına yazınız.

```
<?php
include (‘baglanti.php’);
include (‘../includes/header.php’);
include (‘../includes/navbaradmin.php’);
if ($_SERVER[“REQUEST_METHOD”] == “POST”)
{
  $kategori_adi = $_POST[‘kategoriadi’];
  $sorgu=“INSERT INTO kategori_tablo (kategori_adi) VALUES (‘$kategori_adi’)”;
  if(mysqli_query($baglanti, $sorgu)){
    header( “Location: kategoriler.php” );
  } else{
     echo “Hata:” . mysqli_error($baglanti);
  }
}
?>
<form action=“kategoriekle.php” method=“post”>
    <div class=“form-group mb-3”>
        <label>Kategori Adı</label>
        <input type=“text” class=“form-control” name=“kategoriadi”>
    </div>
    <button type=“submit” class=“btn btn-primary”>Kaydet</button>
</form>
<?php include (‘../includes/footer.php’); ?>
```

3. Adım: Veri tabanına eklenen kategori bilgileri listeleme ve ekleme, silme, güncelleme linkleri oluşturmak için aşağıdaki kodları kategoriler.php dosyasına yazınız.

```
<?php
include (‘baglanti.php’);
include (‘../includes/header.php’);
include (‘../includes/navbaradmin.php’);
?>

<a href=“kategoriekle.php” class=‘btn btn-primary’>Kategori Ekle</a>
<table class=“table”>
  <thead class=”thead-dark”>
    <tr>
      <th scope=”col”>ID</th>
      <th scope=“col”>Kategori Adı</th>
      <th scope=“col”>Sil</th>
      <th scope=“col”>Güncelle</th>
    </tr>
  </thead>
```

```
  <tbody>
    <?php
        $sorgu = "SELECT * FROM kategori_tablo";
        if($veri = mysqli_query($baglanti, $sorgu))
        {
         if(mysqli_num_rows($veri) > 0)
            {
                while($kayit = mysqli_fetch_array($veri))
                {
                echo "<tr>";
                echo "<td>".$kayit["kategori_id"]."</td>";
                echo "<td>".$kayit["kategori_adi"]."</td>";
                echo "<td><a class='btn btn-danger'
         href='kategorisil.php?id=".$kayit["kategori_id"]."'>Sil</a></td>";
                echo "<td><a class='btn btn-success'
href='kategoriduzenle.php?id=".$kayit["kategori_id"]."'>Güncelle</a></td>";
                echo "</tr>";
                }
            }
        }
    ?>
  </tbody>
</table>
<?php include ('../includes/footer.php'); ?>
```

4. Adım: Veri tabanına eklenen kategori bilgilerini silmek için aşağıdaki kodları kategorisil.php dosyasına yazınız.

```
<?php
include ('baglanti.php');
$sorgu = "DELETE FROM kategori_tablo WHERE kategori_id=' " . $_GET["id"] . "'";
if (mysqli_query($baglanti, $sorgu)) {
header("location: kategoriler.php");
exit();
} else {
echo "Kayıt silinemedi :" . mysqli_error($baglanti);
}
mysqli_close($baglanti);
?>
```

5. Adım: Veri tabanına eklenen kategori bilgilerini güncellemek için aşağıdaki kodları kategoriduzenle.php dosyasına yazınız.

```
<?php
include ('baglanti.php');
include ('../includes/header.php');
include ('../includes/navbaradmin.php');
if (isset($_GET['id']))
{
```

```
    $getid=$_GET['id'];
    $sorgu="SELECT * FROM kategori_tablo WHERE kategori_id='$getid'";
    $veri = mysqli_query($baglanti,$sorgu);
    $kayit= mysqli_fetch_array($veri);
    if(!$kayit)
    {
        die("Kayıt bulunamadı");
    }
}
if ($_SERVER["REQUEST_METHOD"] == "POST")
{
    $kategori_id=$_POST['kategoriid'];
    $kategori_adi = $_POST['kategoriadi'];
    $sorgu = "UPDATE kategori_tablo SET kategori_adi='$kategori_adi' WHERE
kategori_id='$kategori_id'";
    if(mysqli_query($baglanti, $sorgu)){
        header( "Location: kategoriler.php" );
    } else{
        echo "Hata:" . mysqli_error($baglanti);
    }
}
?>
<form action="kategoriduzenle.php" method="post">
    <div class="form-group mb-3">
        <label>Kategori Adı</label>
        <input type="text" class="form-control" name="kategoriadi"
value="<?php echo $kayit['kategori_adi'];?>">
    </div>
    <input type="hidden" name="kategoriid" value="<?php echo
       $kayit['kategori_id'];?>" />
    <button type="submit" class="btn btn-primary">Kaydet</button>
</form>
```

9. Uygulama

Yazar ekleme, silme, güncelleme ve listeleme işlemlerini MySQLi nesne yönelimli yöntem ile gerçekleştirmek için diğer sayfadaki adımları takip ediniz.

1. Adım: Admin klasörü içindeki baglanti2.php dosyasını açarak aşağıdaki PHP kodlamalarını yapınız.

```
<?php
$servername = "localhost";
$username   = "root";
$password   = "";
$database   = "haberler_db";
$baglanti2 = new mysqli($servername, $username, $password, $database);
if ($baglanti2 == false)
{
    die("Bağlantı Hatası:" . $baglanti2->connect_error);
}
?>
```

2. Adım: Yazar bilgilerini eklemek için aşağıdaki PHP kodlarını admin klasörü içindeki yazarekle.php dosyasına yazınız.

```
<?php
include ('baglanti2.php');
include ('../includes/header.php');
include ('../includes/navbaradmin.php');

if ($_SERVER["REQUEST_METHOD"] == "POST")
{
   $yazar_eposta = $_POST['yazareposta'];
   $yazar_sifre = $_POST['yazarsifre'];
   $yazar_adsoyad = $_POST['yazaradsoyad'];
   $sorgu="INSERT INTO yazar_tablo (yazar_eposta, yazar_sifre, yazar_adsoyad )
VALUES ('$yazar_eposta', '$yazar_sifre', '$yazar_adsoyad')";

   if($baglanti2->query($sorgu)=== true)
   {
      header( "Location: yazarlar.php" );
   } else{
     echo "Hata:" . $baglanti2->error;
   }
   $baglanti2->close();
}

?>

<form action="yazarekle.php" method="post">
    <div class="form-group mb-3">
        <label>Yazar Eposta</label>
        <input type="text" class="form-control" name="yazareposta">
    </div>
    <div class="form-group mb-3">
        <label>Yazar Şifre</label>
        <input type="password" class="form-control" name="yazarsifre">
    </div>
    <div class="form-group mb-3">
        <label>Yazar Ad Soyad</label>
        <input type="text" class="form-control" name="yazaradsoyad">
    </div>

    <button type="submit" class="btn btn-primary">Kaydet</button>
</form>

<?php include ('../includes/footer.php'); ?>
```

3. Adım: Veri tabanına eklenen yazar bilgilerini listeleme ve ekleme, silme, güncelleme linkleri oluşturmak için aşağıdaki kodları yazarlar.php dosyasına yazınız.

```
<?php
include (‘baglanti2.php’);
include (‘../includes/header.php’);
include (‘../includes/navbaradmin.php’);
?>

<a href=“yazarekle.php” class=‘btn btn-primary’>Yazar Ekle</a>
<table class=“table”>
  <thead class=“thead-dark”>
    <tr>
      <th scope=“col”>ID</th>
      <th scope=“col”>Yazar Adı Soyadı</th>
      <th scope=“col”>Sil</th>
      <th scope=“col”>Güncelle</th>
    </tr>
  </thead>

  <tbody>
    <?php
        $sorgu = “SELECT * FROM yazar_tablo”;
        if($veri = $baglanti2->query($sorgu))
        {
         if($veri->num_rows > 0)
            {
                while($kayit = $veri->fetch_array())
                {
                echo “<tr>”;
                echo “<td>”.$kayit[“yazar_id”].“</td>”;
                echo “<td>”.$kayit[“yazar_adsoyad”].“</td>”;
                echo “<td><a class=‘btn btn-danger’
href=‘yazarsil.php?id=“.$kayit[“yazar_id”].”’>Sil</a></td>”;
                echo “<td><a class=‘btn btn-success’
href=‘yazarduzenle.php?id=“.$kayit[“yazar_id”].”’>Güncelle</a></td>”;
                echo “</tr>”;
                }
            }
        }
    ?>
   </tbody>
</table>

<?php include (‘../includes/footer.php’); ?>
```

4. Adım: Veri tabanına eklenen yazar bilgilerini silmek için aşağıdaki kodları yazarsil.php dosyasına yazınız.

```
<?php
include ('baglanti2.php');
$sorgu = "DELETE FROM yazar_tablo WHERE yazar_id='" . $_GET["id"] . "'";

if ($baglanti2->query($sorgu)===true) {
header("location: yazarlar.php");
exit();
} else {
echo "Kayıt silinemedi :" . $baglanti2->error;
}
$baglanti2->close();
?>
```

5. Adım: Veri tabanına eklenen yazar bilgilerini güncellemek için aşağıdaki kodları yazarduzenle.php dosyasına yazınız.

```
<?php
include ('baglanti2.php');
include ('../includes/header.php');
include ('../includes/navbaradmin.php');
if (isset($_GET['id']))
{
    $getid=$_GET['id'];
    $sorgu="SELECT * FROM yazar_tablo WHERE yazar_id='$getid'";
    $veri = $baglanti2->query($sorgu);
    $kayit= $veri->fetch_array();
    if(!$kayit)
    {
        die("Kayıt bulunamadı");
    }
}
if ($_SERVER["REQUEST_METHOD"] == "POST")
{
    $yazar_id=$_POST['yazarid'];
    $yazar_eposta = $_POST['yazareposta'];
    $yazar_sifre = $_POST['yazarsifre'];
    $yazar_adsoyad = $_POST['yazaradsoyad'];
    $sorgu = "UPDATE yazar_tablo SET yazar_eposta='$yazar_eposta',
              yazar_sifre ='$yazar_sifre', yazar_adsoyad='$yazar_adsoyad'
              WHERE yazar_id='$yazar_id'";
    if($baglanti2->query($sorgu)){
        header( "Location: yazarlar.php" );
    } else{
```

```
        echo "Hata:" . $baglanti2->error;
    }
}
?>
<form action="yazarduzenle.php" method="post">
    <div class="form-group mb-3">
        <label>Yazar Eposta</label>
        <input type="text" class="form-control" name="yazareposta"
        value="<?php echo $kayit['yazar_eposta'];?>">
    </div>
    <div class="form-group mb-3">
        <label>Yazar Şifre</label>
        <input type="password" class="form-control" name="yazarsifre"
        value="<?php echo $kayit['yazar_sifre'];?>">
    </div>
    <div class="form-group mb-3">
        <label>Yazar Adı Soyadı</label>
        <input type="text" class="form-control" name="yazaradsoyad"
         value="<?php echo $kayit['yazar_adsoyad'];?>">
    </div>
    <input type="hidden" name="yazarid"
      value="<?php echo $kayit['yazar_id'];?>" />
    <button type="submit" class="btn btn-primary">Kaydet</button>
</form>
```

10. Uygulama

PHP Data Objects (PDO) ile yazarlar için oturum açma işlemlerini aşağıdaki adımları takip ederek gerçekleştiriniz.

1. Adım: Admin klasörü içine baglanti3.php adında dosya oluşturarak aşağıdaki PHP kodlamalarını yapınız.

```
<?php
$servername = "localhost";
$username   = "root";
$password   = "";
$database   = "haberler_db";

try {
   $baglanti3 = new PDO("mysql:host=$servername;dbname =$database",$username
                        ,$password);
   $baglanti3->setAttribute(PDO::ATTR_ERRMODE,PDO::ERRMODE_EXCEPTION);
} catch (PDOEXCEPTION $e) {
  die("Bağlantı Hatası:" . $e->getMessage());
}
?>
```

2. Adım: Oturum oluşturma işlemleri için login.php dosyasına aşağıdaki PHP kodlarını yazınız.

```
<?php
include ('baglanti3.php');
session_start();

if($_SERVER["REQUEST_METHOD"] == "POST"){
    $yazar_eposta = $_POST['yazareposta'];
    $yazar_sifre = $_POST['yazarsifre'];
    $sorgu = "SELECT * FROM yazar_tablo WHERE yazar_eposta='$yazar_eposta' AND
yazar_sifre='$yazar_sifre'";
    $veri = $baglanti3->query($sorgu);
    $kayit = $veri->fetch();
    if($kayit) {
        $_SESSION['user'] = $kayit['yazar_eposta'];
        $_SESSION['yazar_id'] = $kayit['yazar_id'];
        $_SESSION['yazaradsoyad'] = $kayit['yazar_adsoyad'];
        header("location: index.php");
    } else{
        echo "Kullanıcı adı veya şifre hatalı";
    }
}
?>

<html lang="tr">
<head>
    <title>Yazar Girişi</title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" type="text/css" href="https://cdn.jsdelivr.net/npm/
bootstrap@5.1.0/dist/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>
 <div class="container">
  <div class="row">
   <div class="col-md-4 offset-md-4">
     <div class="bg-light mt-4 p-4">
       <form action="login.php" method="post" class="row g-3">
          <div class="col-12">
            <label>Yazar Eposta</label>
           <input type="text" name="yazareposta" class="form-control">
          </div>
          <div class="col-12">
            <label>Şifre</label>
            <input type="password" name="yazarsifre" class="form-control" >
```

```
          </div>
          <div class="col-12">
           <button type="submit" class="btn btn-dark float-end">Giriş</button>
          </div>
      </form>
     </div>
   </div>
 </div>
</div>
</body>
</html>
```

3. Adım: Oturum sonlandırma işlemleri için logout.php sayfasına aşağıdaki kodlamaları yapınız.

```
<?php
   session_start();
   if(session_destroy()) {
      header("Location: login.php");
   }
?>
```

4. Adım: İzinsiz kullanıcıların erişimini engellemek için oturum kontrolünün yapılacağı sayfalara aşağıdaki kodlamaları ekleyiniz.

```
<?php
session_start();
if(isset($_SESSION['user'])===false){
    header('location:login.php');
}
?>
```

11. Uygulama

Bir sınıf oluşturarak veri tabanına bağlanma, haber ekleme, silme, güncelleme ve listeleme işlemlerini aşağıdaki adımları takip ederek gerçekleştiriniz.

1. Adım: haberleroop.php dosyasına Haberler adında bir sınıf ve içine veri tabanına bağlantısı için yapıcı (constructor) metot oluşturan aşağıdaki kodlamaları yapınız.

Önemli

Yapıcı metot bir sınıftan nesne oluşturulduğunda ilk çalışan metottur.

```
<?php
class Haberler
    {
        private $servername = "localhost";
```

```
        private $username   = "root";
        private $password   = "";
        private $database   = "haberler_db";
        public  $baglanti;
        public function __construct()
        {
            $this->baglanti = new mysqli($this->servername,
                              $this->username, $this->password,
                              $this->database);
            if($this->baglanti === false){
                die("Hata:" . $this->baglanti->connect_error);
            }else{
            return $this->baglanti;
            }
        }
    }
?>
```

2. Adım: Haberler sınıfı içine haber bilgilerini kaydetmek için kullanılacak fonksiyonun kodlamalarını yapınız.

```
public function haberEkle($post)
{
  session_start();
  $kategori_id = $_POST['kategoriid'];
  $haber_baslik =$_POST['haberbaslik'];
  $haber_icerik =$_POST['habericerik'];
  $yazar_id = $_SESSION['yazar_id'];
  $sorgu="INSERT INTO haber_tablo(kategori_id,haber_baslik,haber_icerik,ya
zar_id) VALUES('$kategori_id','$haber_baslik','$haber_icerik','$yazar_id')";
  $sonuc = $this->baglanti->query($sorgu);
  if ($sonuc==true) {
     header("Location:haberler.php");
  }else{
     echo "Kayıt işlemi başarısız!". $this->baglanti->error;
  }
}
```

3. Adım: Haberler sınıfındaki haberEkle fonksiyonunu kullanarak veri eklemek için haberekle.php sayfasına aşağıdaki kodlamaları yapınız.

```
<?php

  include ('haberleroop.php');
  include ('../includes/header.php');
  include ('../includes/navbaradmin.php');
  $haberObj = new Haberler();
  if($_SERVER["REQUEST_METHOD"] == "POST"){
```

```
    $haberObj->haberEkle($_POST);
  }
?>
<form action="haberekle.php" method="POST">
    <div class="form-group mb-3">
        <label>Kategori</label>
        <select class="form-select" name="kategoriid">
        <?php
            $sorgu = "SELECT * FROM kategori_tablo";
            $veri = $haberObj->baglanti->query($sorgu);
           while($kayit = $veri->fetch_assoc()) {
             echo "<option value='" . $kayit['kategori_id'] . "'>"
                   .$kayit['kategori_adi'] . "</option>";
           }
      ?>
        </select>
    </div>
    <div class="form-group mb-3">
        <label>Haber Başlık</label>
        <input type="text" class="form-control" name="haberbaslik">
    </div>
    <div class="form-group mb-3">
        <label>Haber İçerik</label>
        <textarea class="form-control" name="habericerik" rows="10">
        </textarea>
    </div>
    <input type="submit"  class="btn btn-primary" value="Kaydet">
  </form>
</div>
```

4. Adım: Haberler sınıfı içine haber bilgilerini listelemek için kullanılacak fonksiyonun kodlamalarını yapınız.

```
public function haberListe()
 {
    $sorgu = "SELECT * FROM haber_tablo";
    $veri = $this->baglanti->query($sorgu);
    if ($veri->num_rows > 0) {
       $haberler = array();
       while ($kayit = $veri->fetch_assoc())
       {
          $haberler[] = $kayit;
       }
       return $haberler;
    }
    else{
       echo "Kayıt bulunamadı.";
    }
 }
```

5. Adım: Haberler sınıfındaki haberListe fonksiyonunu kullanarak verileri listelemek için haberler.php sayfasına aşağıdaki kodlamaları yapınız.

```
<?php
 include (‘haberleroop.php’);
include (‘../includes/header.php’);
include (‘../includes/navbaradmin.php’);
$haberObj = new Haberler();
?>
<a href=“haberekle.php” class=‘btn btn-primary’>Haber Ekle</a>
<table class=“table”>
  <thead class=“thead-dark”>
    <tr>
      <th scope=“col”>Haber Id</th>
      <th scope=“col”>Haber Başlık</th>
      <th scope=“col”>Haber Tarihi</th>
      <th scope=“col”>Sil</th>
      <th scope=“col”>Güncelle</th>
    </tr>
  </thead>
  <tbody>
        <?php
          $haberler = $haberObj->haberListe();
          foreach ($haberler as $haber)
          {
          echo “<tr>”;
          echo “<td>”.$haber[‘haber_id’].“</td>”;
          echo “<td>”.$haber[‘haber_baslik’].“</td>”;
          echo “<td>”.$haber[‘haber_tarih’].“</td>”;
          echo “<td><a class=‘btn btn-danger’
         href=‘habersil.php?id=“.$haber[‘haber_id’].”’>Sil</a></td>”;
          echo “<td><a class=‘btn btn-success’
   href=‘haberduzenle.php?id=“.$haber[‘haber_id’].”’>Güncelle</a></td>”;
          echo “</td>”;
          echo “</tr>”;
         }
         ?>
    </tbody>
  </table>
```

6. Adım: Haberler sınıfı içine haber bilgilerini silmek için kullanılacak fonksiyonun kodlamalarını yapınız.

```
public function haberSil($id)
{
    $sorgu = “DELETE FROM haber_tablo WHERE haber_id = ‘$id’”;
    $sonuc = $this->baglanti->query($sorgu);
    if ($sonuc==true) {
       header(“Location:haberler.php”);
     }else{
        echo “Haber silinemedi.”;
     }
}
```

7. Adım: Haberler sınıfındaki haberSil fonksiyonunu kullanarak verileri listelemek için habersil.php sayfasına aşağıdaki kodlamaları yapınız.

```
<?php
 include (‘haberleroop.php’);
 $haberObj = new Haberler();
if(isset($_GET[‘id’]) && !empty($_GET[‘id’])) {
      $haber_id = $_GET[‘id’];
      $haberObj->haberSil($haber_id);
  }
?>
```

8. Adım: Haberler sınıfı içine veri tabanındaki kayıtlardan birini seçmek için kullanılacak fonksiyonun kodlamalarını yapınız.

```
public function haberSec($id)
{
    $sorgu = “SELECT * FROM haber_tablo WHERE haber_id = ‘$id’”;
    $sonuc = $this->baglanti->query($sorgu);
    if ($sonuc->num_rows > 0) {
    $haber = $sonuc->fetch_assoc();
    return $haber;
    }else{
    echo “Kayıt bulunamadı.”;
    }
}
```

9. Adım: Haberler sınıfı içine haber bilgilerini güncellemek için kullanılacak fonksiyonun kodlamalarını yapınız.

```
public function haberDuzenle($post)
{
    session_start();
    $haber_id = $_POST[‘id’];
    $kategori_id = $_POST[‘kategoriid’];
    $haber_baslik =$_POST[‘haberbaslik’];
    $haber_icerik =$_POST[‘habericerik’];
    $yazar_id = $_SESSION[‘yazar_id’];

    if (!empty($haber_id) && !empty($post)) {
        $sorgu = “UPDATE haber_tablo SET kategori_id = ‘$kategori_id’,
                    haber_baslik = ‘$haber_baslik’,
                    haber_icerik = ‘$haber_icerik’,
                    yazar_id = ‘$yazar_id’
                    WHERE haber_id = $haber_id”;
        $sonuc = $this->baglanti->query($sorgu);
        if ($sonuc==true) {
            header(“Location:haberler.php”);
        }else{
            echo “Güncelleme yapılamadı!.”. $this->baglanti->error;
        }
        }
}
```

10. Adım: Haberler sınıfındaki haberSec ve haberDuzenle fonksiyonlarını kullanarak verileri güncellemek için haberduzenle.php sayfasına aşağıdaki kodlamaları yapınız.

```
<?php

include ('haberleroop.php');
include ('../includes/header.php');
include ('../includes/navbaradmin.php');

$haberObj = new Haberler();

  // Güncellenecek kayıtları getirir.
  if(isset($_GET['id']) && !empty($_GET['id'])) {
    $haber_id = $_GET['id'];
    $haber = $haberObj->haberSec($haber_id);
  }

  // Güncellenen kayıtları kaydeder.
  if($_SERVER["REQUEST_METHOD"] == "POST") {
    $haberObj->haberDuzenle($_POST);
  }

?>
<form action="haberduzenle.php" method="POST">
    <div class="form-group mb-3">
        <label>Kategori</label>
        <select class="form-select" name="kategoriid">
        <?php
            $sorgu = "SELECT * FROM kategori_tablo";
            $veri = $haberObj->baglanti->query($sorgu);
           while($kayit = $veri->fetch_assoc()) {
               if($haber['kategori_id']==$kayit['kategori_id'])
               {
                echo "<option selected value='" . $kayit['kategori_id'] . "'>" .
$kayit['kategori_adi'] . "</option>";
               }
               else
               {
                echo "<option value='" . $kayit['kategori_id'] . "'>" .
$kayit['kategori_adi'] . "</option>";
               }

           }
      ?>
        </select>
```

```
    </div>
    <div class="form-group mb-3">
        <label>Haber Başlık</label>
        <input type="text" class="form-control" name="haberbaslik"
value="<?php echo $haber['haber_baslik']; ?>">
    </div>
    <div class="form-group mb-3">
        <label>Haber İçerik</label>
        <textarea class="form-control" name="habericerik" rows="10" >
          <?php echo $haber['haber_icerik']; ?>
        </textarea>
    </div>
    <input type="hidden" name="id" value="<?php echo $haber['haber_id']; ?>">
    <input type="submit" class="btn btn-primary" value="Güncelle">
  </form>
</div>
```

ÖLÇME VE DEĞERLENDİRME

Aşağıdaki soruları dikkatlice okuyarak doğru seçeneği işaretleyiniz.

1. MySQL komut satırından veri tabanı sunucusuna bağlanmak için aşağıdaki komutlardan hangisi kullanılır?

A) > mysql –r root –p

B) >mysql –h localhost –s root

C) >mysqli –u root –p

D) >mysql –u root –p

E) > mysqli –r root –p

2. Aşağıdakilerden hangisi MySQL sayı veri türlerinden değildir?

A) CHAR

B) DECIMAL

C) DOUBLE

D) FLOAT

E) INT

3. Aşağıdakilerden hangisi YYYY-MM-DD formatında veri saklayan MySQL veri türüdür?

A) BIGDATE

B) DATE

C) DATETIME

D) TIME

E) TIMESTAMP

4. Veri tabanı yönetim sisteminde aşağıdakilerden hangisi bulunmaz?

A) Programlama dili derleyicisi

B) Veri sözlüğü

C) Veri tabanı motoru

D) SQL arayüzü

E) Veri tabanı dosyaları

5. Aşağıdakilerden hangisi veri tabanı oluşturma ve silme ifadeleridir?

A) CREATE DATABASE – DELETE DATABASE

B) CREATE DATABASE – DROP DATABASE

C) ADD DATABASE – DROP DATABASE

D) INSERT DATABASE – DELETE DATABASE

E) CREATE DATABASE – REMOVE DATABASE

6. phpMyAdmin arayüzü ile aşağıdakilerden hangisi yapılamaz?

A) SQL komutları çalıştırılır.

B) Veri tabanı, tablo ve alan oluşturulur.

C) Web sunucusu olarak çalışır.

D) Veri tabanı, tablo ve alanlar düzenlenir.

E) Verileri metin belgesi olarak dışa aktarır.

7. MySQL veri tabanında sayısal türdeki alanların değerlerini artırmak için aşağıdaki özelliklerden hangisi kullanılır?

A) Auto_Increment

B) Auto_Insert

C) Increment_Auto

D) Primary_Increment

E) Auto_Rise

8. mysqli_connect(?, ?, ?, ?)
Verilen bu bağlantı işleminde soru işareti ile belirtilen yerlere sırasıyla aşağıdakilerden hangisi gelmelidir?

A) kullanıcı adı-şifre- sunucu adı- veri tabanı

B) veri tabanı-sunucu adı-kullanıcı adı-şifre

C) sunucu adı-veri tabanı - kullanıcı adı-şifre

D) sunucu adı- kullanıcı adı-şifre-veri tabanı

E) veri tabanı- kullanıcı adı-şifre- sunucu adı

9. PDO("?:host=?;dbname=?", ? , ?)
Verilen bu bağlantı işleminde MYSQL veri tabanı için soru işareti ile belirtilen yerlere sırasıyla aşağıdakilerden hangisi gelmelidir?

A) mysql- veritabanı adı- sunucu adı- kullanıcı adı-şifre

B) sunucu adı-veritabanı adı- kullanıcı adı-şifre- mysql

C) mssql-sunucu adı-veritabanı adı- kullanıcı adı-şifre

D) sunucu adı- mysql-veritabanı adı- kullanıcı adı-şifre

E) mysql-sunucu adı-veritabanı adı- kullanıcı adı-şifre

10. Aşağıdakilerden hangisi SQL sorgularını çalıştıran PHP fonksiyonudur?

A) mysqli_run()

B) mysqli_query()

C) mysqli_connect()

D) mysqli_exec()

E) mysqli_prepare()

CEVAP ANAHTARLARI

1. ÖĞRENME BİRİMİ

1	2	3	4	5	6	7
B	E	C	E	D	A	D

2. ÖĞRENME BİRİMİ

A.	1	2	3	4	5	6
	Y	D	D	Y	Y	D

B.	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	B	A	E	A	C	D	C	B	D	D	E	A	B	A

3. ÖĞRENME BİRİMİ

A.	1	2	3	4	5
	D	Y	D	D	Y

B.	6	7	8	9	10	11	12	13	14	15	16	17
	C	B	D	A	C	B	B	D	A	A	A	C

4. ÖĞRENME BİRİMİ

A.	1	2	3	4	5
	Y	Y	D	Y	D

B.	6	7	8	9	10
	B	A	D	B	C

5. ÖĞRENME BİRİMİ

A.	1	2	3	4	5
	Y	D	Y	D	D

B.	6	7	8
	E	C	E

6. ÖĞRENME BİRİMİ

A.	1	2	3	4
	mantıksal	try	getMessage()	finally

B.		
	1.	Ölümcül düzeydeki hatanın meydana geldiği yerden sonraki PHP kodları çalışmazken uyarı düzeyindeki hatanın meydana geldiği yerden sonraki PHP kodları çalışır.
	2.	Ölümcül düzeydeki hatalar exception ile kontrol edilemez. Uyarı düzeyindeki hatalar exception ile kontrol edilebilir.

C.	1	2	3	4
	getMessage()	getLine()	getCode()	getFile()

D.					
	is_string($b)	Exception	$e	getMessage()	1900

7. ÖĞRENME BİRİMİ

A.	1	2	3
	readdir()	is_dir()	touch()

B.	
	require_once() fonksiyonu ile içe aktarılmak istenen PHP sayfası aynı sayfada birden çok yerde çağrılabilirken require() fonksiyonu ile sadece bir yerde çağrılabilir.

C.	1	2	3
	r -Okuma(Read) modudur.	w -Write(Yazma) modudur.	a - Append(Ekleme) modudur.

D.		
	1.	“dosya.txt” adında yeni bir dosya oluşturur.
	2.	“makale.php” dosyasını hem okuma hem yazma modunda açar.
	3.	“Resimler” adında yeni bir klasör oluşturur.
	4.	$dosya değişkende bulunan değerin klasör olup olmadığını kontrol eder.
	5.	“Resimler” adındaki klasörü siler.
	6.	“profil.txt” adındaki dosyayı siler.
	7.	“evrak.txt” adında bir dosyanın var olup olmadığını kontrol eder.
	8.	Belirtilen dosyayı kapatır.

8. ÖĞRENME BİRİMİ

1.	Kullanıcının bilgisayarında saklanır.
2.	Sunucu da saklanır.
3.	1.parametre: Çerezin adı 2.parametre: Çerezin alacağı değer 3.parametre: Çerezin ne kadar kullanılacağı
4.	Oluşturulan oturumlar web sayfasının bulunduğu sekme kapatıldığında veya tarayıcı kapatıldığında sonlanır. Çerez yapılarında ise saniye cinsinden girilen değere göre çerez kullanılır.
5.	session_destroy();
6.	session_start();
7.	setcookie($cookie_adi, $cookie_degeri,time() -1);
8.	`setcookie("sitedili","Kullanıcı Almanca biliyor.",60*60*24*90);`
9.	`<?php` `session_start();` `$ip = $_SERVER['REMOTE_ADDR'];` `$_SESSION["kullanici_ip"]=$ip;` `$saat=date('H:i');` `$_SESSION["giris_saat"]=$saat;` `echo $ip." ".$saat;` `?>`

9. ÖĞRENME BİRİMİ

1	2	3	4	5	6	7	8	9	10
D	A	B	A	B	C	A	D	E	B

KAYNAKÇA

- Bilişim Teknolojileri Alanı Çerçeve Öğretim Programı
- Türk Dil Kurumu, sözlükleri. (t.y.). https://sozluk.gov.tr/
- Türk Dil Kurumu (t.y.). https://www.tdk.gov.tr/

GENEL AĞ KAYNAKÇASI

- php.net (anasayfa). (2021). Downloads. https://www.php.net/downloads.php, Haziran 25, 2021, kaynağından alınmıştır.
- apachelounge.com (anasayfa). (2021). Downloads. https://www.apachelounge.com/download/, Haziran 26, 2021, kaynağından alınmıştır.
- MySQL (anasayfa). (2021). Downloads. https://dev.mysql.com/downloads/, Temmuz 16, 2021, kaynağından alınmıştır.
- apachefriends.org (anasayfa). (2021). https://www.apachefriends.org/tr/index.html, Temmuz 29, 2021, kaynağından alınmıştır.

GÖRSEL KAYNAKÇASI

http://kitap.eba.gov.tr/karekod/Kaynak.php?KOD=2418

* Kaynakça kısmı APA6 referanslama sistemi kullanılarak oluşturulmuştur.