

PHP- PDO

(İNTERNET PROGRAMCILIĞI DERSİ İÇİN)

ADIM ADIM DİNAMİK SİTE TASARLAMA



2018

Murat SARIÖZ – Mercin KARAKAŞ

PHP- PDO PROJESİ

(Yönetici Arayüzü)

ADIM ADIM DİNAMİK SİTE TASARLAMA

Murat SARIÖZ – Mercin KARAKAŞ

BÜYÜKÇEKMECE SUDİ ÖZKAN MESLEKİ VE TEKNİK ANADOLU LİSESİ

İçindekiler

GİRİŞ	1
PROJENİN DOSYA VE DİZİN YAPISI.....	1
VERİTABANINI HAZIRLAMAK	2
VERİTABANINI OLUŞTURMAK.....	2
VERİTABANI BAĞLANTISI İÇİN KULLANILACAK PHP KOD DOSYASINI OLUŞTURMAK	5
YÖNETİM PANELİNİ OLUŞTURMAK	7
ARAYÜZ TASARIMI İÇİN GEREKLİ DOSYALARIN KURULUMU	7
WEB SAYFALARININ ANA HTML ÇATISINI OLUŞTURMAK	7
CSS STİL DOSYASINI (STYLE.CSS) OLUŞTURMAK.....	7
BAŞLIK (HEADER.PHP) SAYFASINI OLUŞTURMAK	8
ALTBİLGİ (FOOTER.PHP) SAYFASINI OLUŞTURMAK.....	9
YÖNETİM PANELİ ANASAYFASINI (INDEX.PHP) OLUŞTURMAK	9
ÜRÜN SAYFALARINI OLUŞTURMAK	11
ÜRÜN LİSTELEME SAYFASINI OLUŞTURMAK	11
ÜRÜN SİLME SAYFASINI OLUŞTURMAK.....	13
ÜRÜN DETAY SAYFASINI OLUŞTURMAK	15
ÜRÜN EKLEME SAYFASINI OLUŞTURMAK.....	17
ÜRÜN GÜNCELLEME (DÜZELTME) SAYFASINI OLUŞTURMAK	19
SAYFALANDIRMA İŞLEMİ	23
SAYFALAMA DEĞİŞKENLERİNİ TANIMLAMAK	23
SEÇME SORGUSUNU DÜZENLEMEK	23
KAYIT SAYISINI HESAPLAMAK	23
SAYFALANDIRMA DOSYASINI (PAGING.PHP) OLUŞTURMAK	24
RESİM YÜKLEMEK	28
VERİTABANINA YENİ BİR ALAN EKLEMEK	28
ÜRÜN EKLEME SAYFASINI DÜZENLEMEK	29
ÜRÜN EKLEME SORGUSUNU DÜZENLEMEK.....	29
DOSYA YÜKLEME VE KAYDETME İŞLEMİ	30
ÜRÜN DETAY SAYFASINDA YÜKLENEN RESMİ GÖRÜNTÜLEMEK	32
ÜRÜN SİLİNDİĞİNDE YÜKLENEN RESMİ SUNUCUDAN SİLMEK	33
ARAMA İŞLEMİ.....	35
ARAMA FORMUNU OLUŞTURMAK.....	35
SEÇME SORGUSUNU DÜZENLEMEK	36
ARAMA VE SAYFALANDIRMA İŞLEMLERİNİ BİRLEŞTİRMEK	36
BUTONLARIN GÖRSELLİĞİNİ ZENGİNLEŞTİRMEK	38
ÜRÜN LİSTE SAYFASINDAKİ BUTONLARA SİMGE EKLEMEK.....	38
ÜRÜN DETAY SAYFASINDAKİ BUTONLARA SİMGE EKLEMEK	38
ÜRÜN EKLEME SAYFASINDAKİ BUTONLARA SİMGE EKLEMEK	39
ÜRÜN GÜNCELLEME SAYFASINDAKİ BUTONLARA SİMGE EKLEMEK.....	40

ÇOKLU ÜRÜN SİLME İŞLEVİ.....	41
ÇOKLU ÜRÜN SEÇME KUTULARINI (CHECKBOX) OLUŞTURMAK	41
AJAX KODLARINI OLUŞTURMAK	42
ÇOKLU ÜRÜN SİLME SAYFASINI (COKLUSIL.PHP) OLUŞTURMAK	43
KATEGORİ SAYFALARINI OLUŞTURMAK	44
KATEGORİ LİSTELEME SAYFASINI OLUŞTURMAK	45
KATEGORİ SİLME SAYFASINI OLUŞTURMAK	47
KATEGORİ DETAY SAYFASINI OLUŞTURMAK	49
KATEGORİ EKLEME SAYFASINI OLUŞTURMAK	50
KATEGORİ GÜNCELLEME (DÜZELTME) SAYFASINI OLUŞTURMAK	52
ÜRÜNLER TABLOSUYLA KATEGORİLER TABLOSUNU İLİŞKİLENDİRMEK	56
ÜRÜN EKLEME SAYFASINA KATEGORİ ALANI EKLEMEK	57
ÜRÜN GÜNCELLEME SAYFASINA KATEGORİ ALANI EKLEMEK.....	58
ÜRÜN LİSTELEME SAYFASINA KATEGORİ ALANI EKLEMEK.....	61
ÜRÜN DETAY SAYFASINA KATEGORİ ALANI EKLEMEK	62
KATEGORİ SAYFASINA ÜRÜN LİSTESİ EKLEMEK.....	63
KULLANICI SAYFALARINI OLUŞTURMAK	66
ANA SAYFADA KULLANICI MENÜSÜNÜ OLUŞTURMAK	67
KULLANICI LİSTELEME SAYFASINI OLUŞTURMAK	67
KULLANICI SİLME SAYFASINI OLUŞTURMAK.....	69
KULLANICI EKLEME SAYFASINI OLUŞTURMAK	71
KULLANICI GÜNCELLEME (DÜZELTME) SAYFASINI OLUŞTURMAK	73
OTURUM YÖNETİMİ	77
SAYFALARDA OTURUM KONTROLÜNÜ YAPMAK	77
OTURUM AÇMA SAYFASINI (LOGIN.PHP) OLUŞTURMAK.....	77
KULLANICI KONTROLÜ VE OTURUM AÇMA İŞLEMİ.....	78
OTURUMU KAPATMAK.....	79
EK GÖREVLER.....	81
KULLANICI ARAYÜZÜNÜ OLUŞTURMAK	1
WEB SAYFALARININ ANA HTML ÇATISINI OLUŞTURMAK	3
CSS STİL DOSYASINI (STYLE.CSS) DÜZENLEMEK	3
BAŞLIK (HEADER.PHP) SAYFASINI OLUŞTURMAK	3
ALTBİLGİ (FOOTER.PHP) SAYFASINI OLUŞTURMAK.....	6
ANASAYFAYI (INDEX.PHP) OLUŞTURMAK	8
SLIDER SATIRINI OLUŞTURMAK	8
KARŞILAMA SATIRINI OLUŞTURMAK	10
FIRSAT SATIRINI OLUŞTURMAK	10
YENİ ÜRÜN SATIRINI OLUŞTURMAK.....	12
ÜRÜNLER SAYFASINI (URUNLER.PHP) OLUŞTURMAK	13
BANNER SATIRINI OLUŞTURMAK	15

KATEGORİ LİSTESİNİ OLUŞTURMAK	15
SIRALAMA SEÇENEKLERİNİ OLUŞTURMAK	17
FİYAT SEÇENEKLERİNİ OLUŞTURMAK	18
ÜRÜN LİSTESİNİ OLUŞTURMAK	19
ÜRÜN DETAY SAYFASINI (URUNDETAY.PHP) OLUŞTURMAK.....	21
ÜRÜN BİLGİLERİNİ LİSTELEMELİK İÇİN GEREKLİ SORGUYU OLUŞTURMAK	22
ÜRÜN RESMİNİ GÖRÜNTÜLEMELİK	22
ÜRÜN BİLGİLERİNİ GÖRÜNTÜLEMELİK	22
DİĞER ÜRÜNLERİ LİSTELEMELİK.....	23
HAKKIMIZDA SAYFASINI (HAKKIMIZDA.PHP) OLUŞTURMAK	24
TANITIM YAZISI İLE İLETİŞİM FORMUNU OLUŞTURMAK	24
E-POSTA GÖNDERME SAYFASINI OLUŞTURMAK.....	26
ALİŞVERİŞ SEPETİNİ OLUŞTURMAK.....	27
SEPET OTURUMUNU OLUŞTURMAK	27
SEPETE EKLE BUTONU KODLARINI OLUŞTURMAK	27
SEPETTEKİ ÜRÜNLERİ GÖRÜNTÜLEMELİK.....	30
ÜRÜN GÜNCELLE BUTONU KODLARINI OLUŞTURMAK.....	34
ÜRÜN SİL BUTONU KODLARINI OLUŞTURMAK	35
SATINALMA SAYFASINI OLUŞTURMAK.....	36
SİPARİŞ ONAYI SAYFASINI OLUŞTURMAK.....	39
EK GÖREVLER.....	41

GİRİŞ

Bu belge ile PHP ve MySQL kullanarak basit bir CRUD (Create, Read, Update, Delete) uygulaması geliştirme aşamaları öğretilmesi amaçlanmaktadır. Uygulama geliştirme aşamasında daha yeni bir programlama yöntemi olan PDO (PHP Data Objects) yapısı kullanılacaktır.

Geliştireceğimiz web sayfası bir şirketin ürünlerinin dinamik olarak kaydedilmesi, listelenmesi, güncellenmesi, silinmesi (CRUD) ve bir kullanıcı arayüzü ile bu ürünlerin tanıtımına yönelik olacak. Uygulamamızı aşama aşama geliştireceğiz.

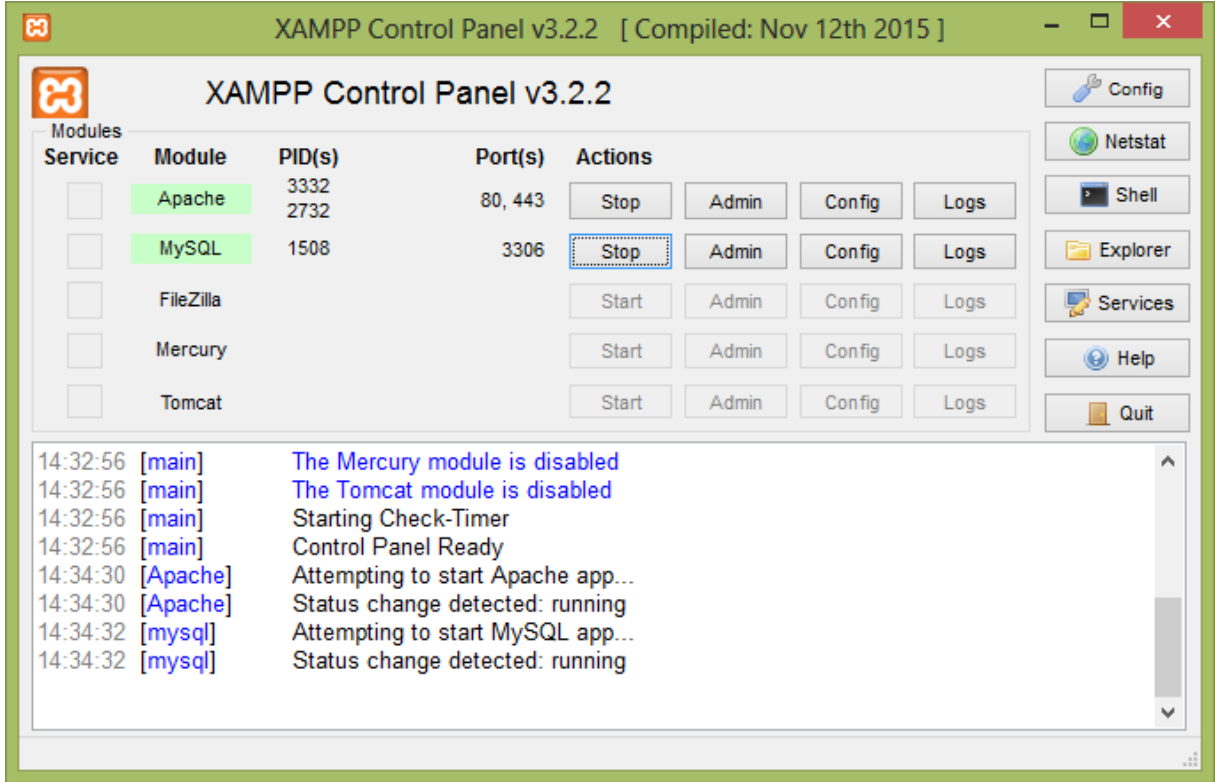
PROJENİN DOSYA VE DİZİN YAPISI

PDO CRUD uygulamamız başlıca şu dosyalardan oluşacak:

- **config/vtabani.php** – veritabanı bağlantısı ve ayarları için kullanılacak.
- **admin/header.php** – web sayfalarının başlık bilgilerini içeren sayfa.
- **admin/footer.php** – web sayfalarının alt bilgilerini içeren sayfa.
- **admin/index.php** – yönetim panelinin ana sayfası.
- **admin/login.php** – yönetim panelinin kullanıcı giriş sayfası.
- **admin/sayfalama.php** – liste sayfasındaki kayıtları sayfa sayfa görüntülemek için kullanılacak.
- **admin/xxxx/liste.php** – veritabanından kayıtları okumak için kullanılan sayfa. MySQL veritabanındaki kayıtları listelemek için HTML tablosu kullanır.
- **admin/xxxx/ekle.php** – yeni kayıt eklemek için kullanılan sayfa. Kullanıcının yeni kayıt eklemesine olanak veren HTML formu içerir.
- **admin/xxxx/detay.php** – veritabanından tek bir kayıt okumak ve listelemek için kullanılan sayfa. MySQL veritabanındaki kaydı listelemek için HTML tablosu kullanır.
- **admin/xxxx/duzelt.php** – bir kaydı güncellemek için kullanılan sayfa. Kullanıcının gelen “id” parametresine göre ilgili kaydı güncellemesine olanak veren HTML formu içerir.
- **admin/xxxx/sil.php** – bir kaydı silmek için kullanılan sayfa. Gelen “id” parametresine göre ilgili kayıt silinir ve ardından “xxxx/liste.php” sayfasına yönlendirme yapılır.
- **admin/xxxx/coklusil.php** – birden fazla kaydı aynı anda silmek için kullanılan sayfa.
- **content/css** – stil dosyalarını barındıran klasör.
- **content/js** – javascript dosyalarını barındıran klasör.
- **content/images** – resimlerin yükleneceği klasör.

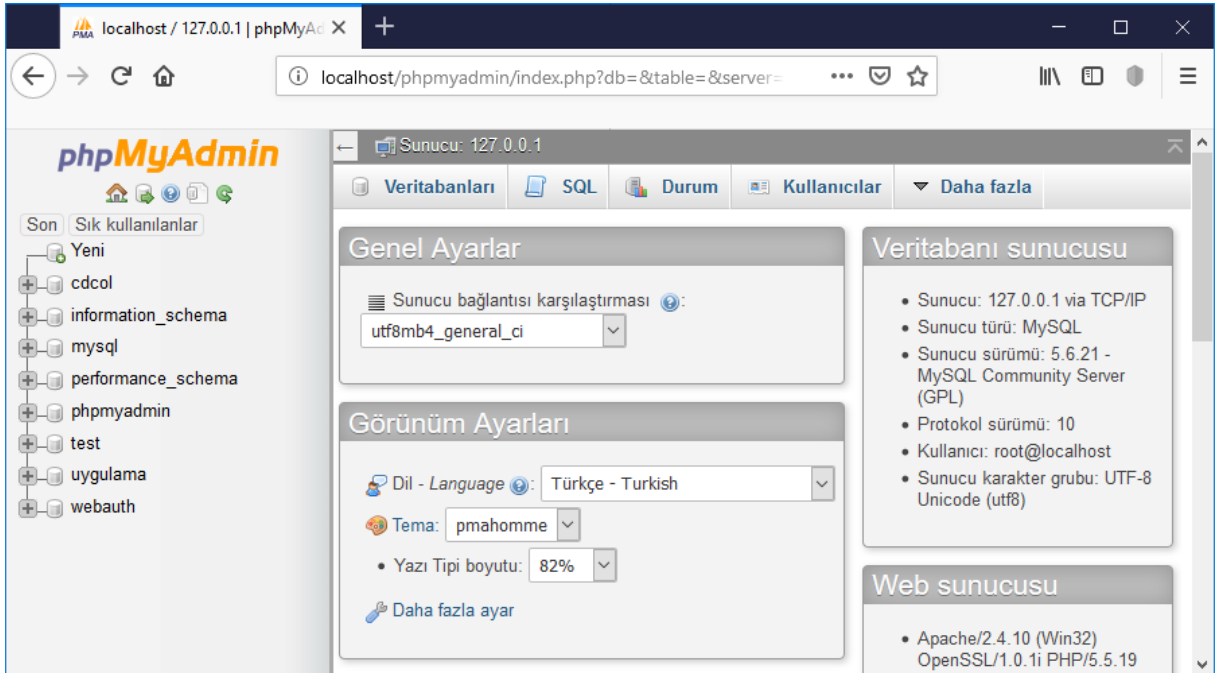
VERİTABANINI HAZIRLAMAK

Proje geliştirme aşamasında **XAMPP** açık kaynak PHP yazılım paketini kullanacağız. **XAMPP** paketini indirip kurun ve **Apache** ile **MySQL** servislerini çalıştırın.

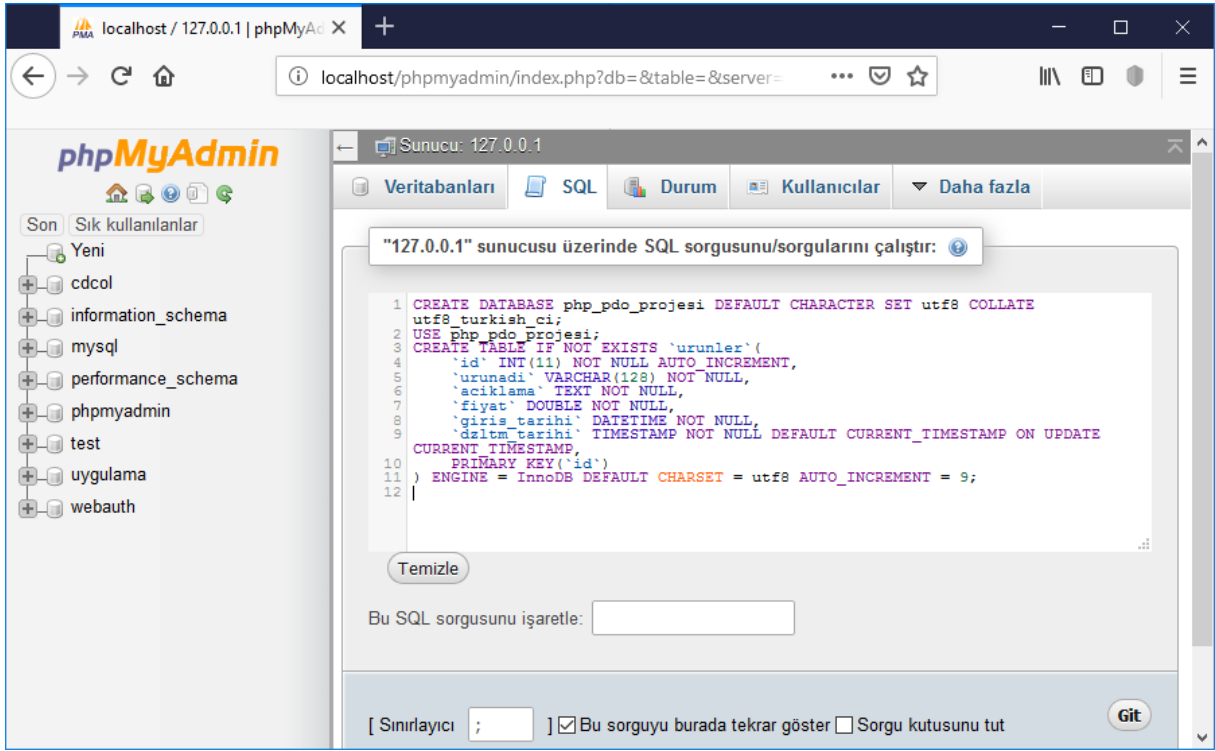


VERİTABANINI OLUŞTURMAK

Veritabanını oluşturmak için **MySQL** servisinin sağındaki **Admin** butonuna tıklayarak veya tarayıcınızın adres satırına <http://localhost/phpmyadmin/> yazarak **phpMyAdmin** veritabanı yönetim panelini açın.



Şimdi **SQL** bağlantısına tıklayın ve aşağıdaki **SQL** sorgusunu yazarak **Git** butonuyla çalıştırın.



```
CREATE DATABASE php_pdo_projesi DEFAULT CHARACTER SET utf8 COLLATE
utf8_turkish_ci;

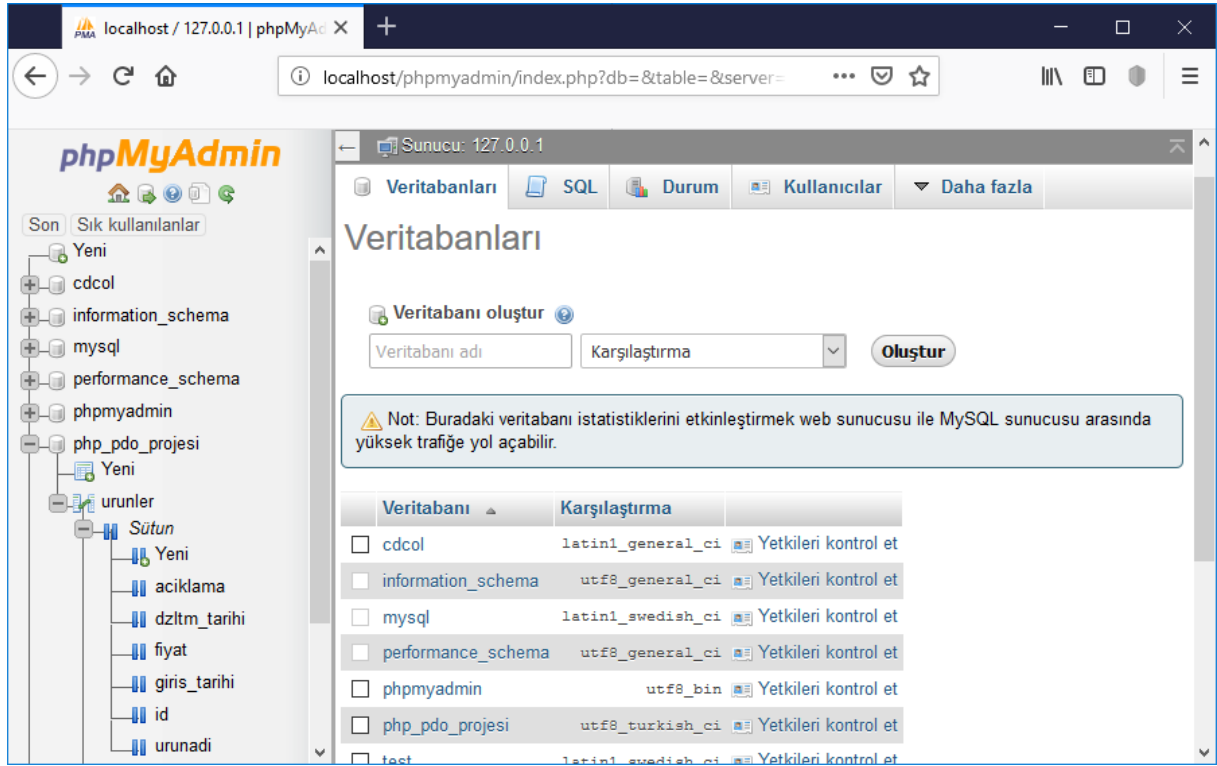
USE php_pdo_projesi;

CREATE TABLE IF NOT EXISTS `urunler` (
  `id` INT(11) NOT NULL AUTO_INCREMENT,
  `urunadi` VARCHAR(128) NOT NULL,
  `aciklama` TEXT NOT NULL,
  `fiyat` DOUBLE NOT NULL,
  `giris_tarihi` DATETIME NOT NULL,
  `dzltn_tarihi` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
  PRIMARY KEY(`id`)
) ENGINE = InnoDB DEFAULT CHARSET = utf8 AUTO_INCREMENT = 9;
```

Bilgi:

- INT() – Yalnızca integer (tamsayı) değeri tutar. Maksimum karakter genişliği parametre olarak alınır.
- VARCHAR() – Belirtilen sayı kadar karakter barındırır (en fazla 255 karakter).
- TEXT - Uzun yazılar için. Değişken bir metni en fazla 65535 karakter uzunluğu ile tutar.
- DOUBLE – Negatif ve pozitif ondalık sayıları depolar. Sayıları kesirleri ile birlikte tutar.
- DATETIME – Tarih ve zaman değerlerini tutar (yyyy-mm-dd hh:mm:ss).
- TIMESTAMP – Tarih ve zaman değerlerini tutar (yyyymmddhhmmss).
- AUTO_INCREMENT – Id alanları için her veri girişinde sıradan yeni bir değer atama işlemini yapar.
- NOT NULL – Seçili olduğu alana mutlaka veri girişi olmalıdır. Boş veri girilemez.
- PRIMARY KEY – Birincil Anahtar anlamına gelir, her tabloda olması gereken bir özelliktir. Sadece bir alana tanımlanabilir. Tablolarda verilere ulaşabilmek için bu özellik hem performans hem de kolaylık açısından önemlidir. Birincil Anahtar, o alanda aynı ikinci veriye izin vermez. O yüzden çoğu tablolarda Id alanlarına tanımlanır.

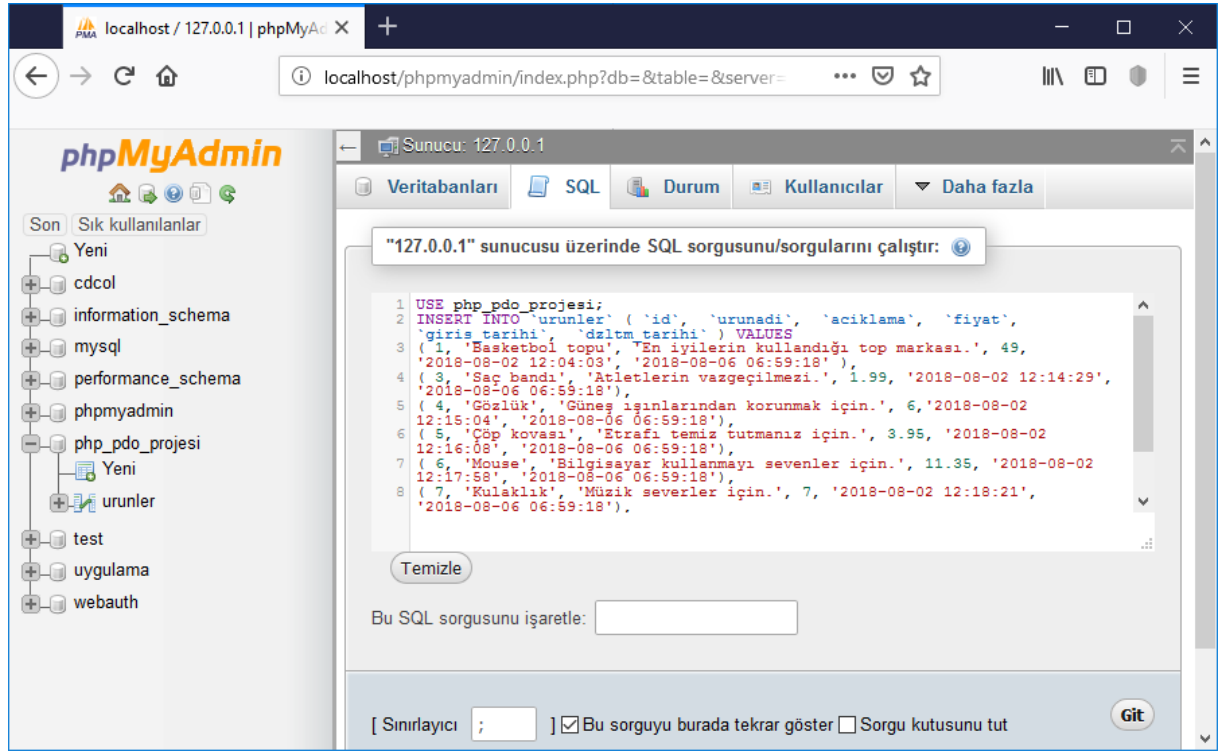
Veritabanları bağlantısına tıklayarak veritabanının, tablonun ve tablo alanlarının oluşup oluşmadığını kontrol edin.



Ardından **SQL** bağlantısına tekrar tıklayın ve aşağıdaki SQL komutunu çalıştırarak oluşturduğunuz **urunler** tablosuna örnek veriler kaydedin.

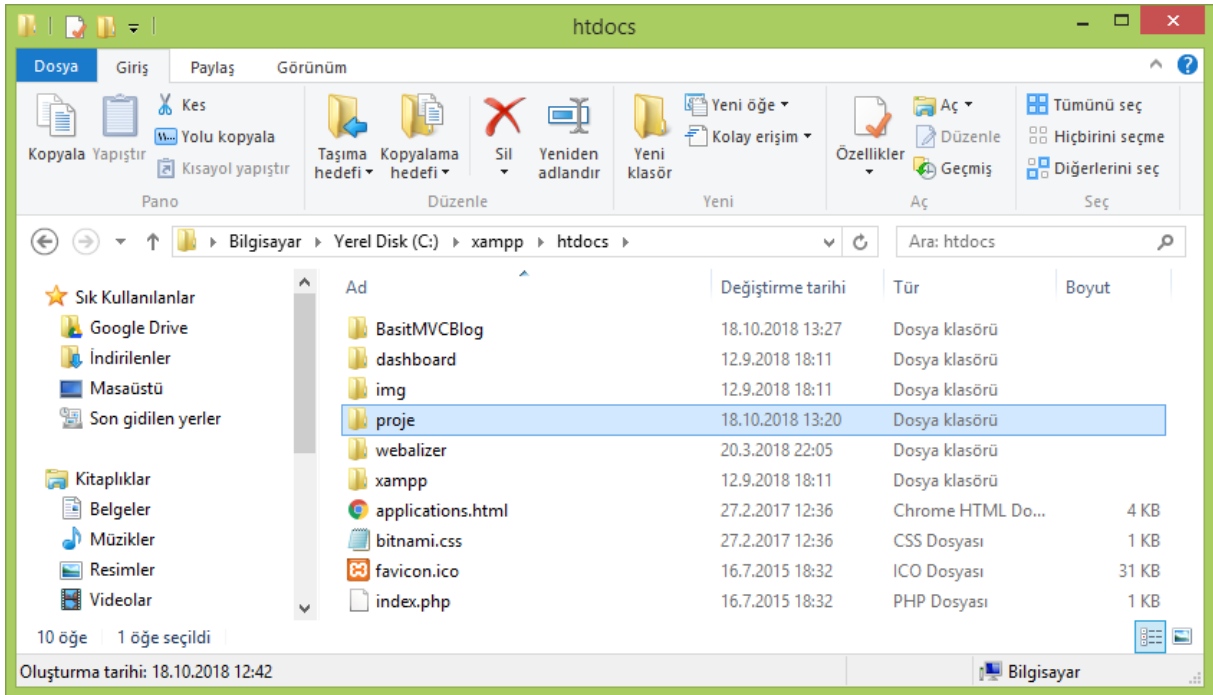
```
USE php_pdo_projesi;

INSERT INTO `urunler` ( `id`, `urunadi`, `aciklama`, `fiyat`, `giris_tarihi`,
`dzlmt_tarihi` ) VALUES
( 1, 'Basketbol topu', 'En iyilerin kullandığı top markası.', 49, '2018-08-02
12:04:03', '2018-08-06 06:59:18' ),
( 3, 'Saç bandı', 'Atletlerin vazgeçilmezi.', 1.99, '2018-08-02 12:14:29', '2018-
08-06 06:59:18'),
( 4, 'Gözlük', 'Güneş ışınlarından korunmak için.', 6, '2018-08-02 12:15:04',
'2018-08-06 06:59:18'),
( 5, 'Çöp kovası', 'Etrafı temiz tutmanız için.', 3.95, '2018-08-02 12:16:08',
'2018-08-06 06:59:18'),
( 6, 'Mouse', 'Bilgisayar kullanmayı sevenler için.', 11.35, '2018-08-02
12:17:58', '2018-08-06 06:59:18'),
( 7, 'Kulaklık', 'Müzik severler için.', 7, '2018-08-02 12:18:21', '2018-08-06
06:59:18'),
( 8, 'Yastık', 'İyi bir uyku için.', 8.99, '2018-08-02 12:18:56', '2018-08-06
06:59:18');
```



VERİTABANI BAĞLANTISI İÇİN KULLANILACAK PHP KOD DOSYASINI OLUŞTURMAK

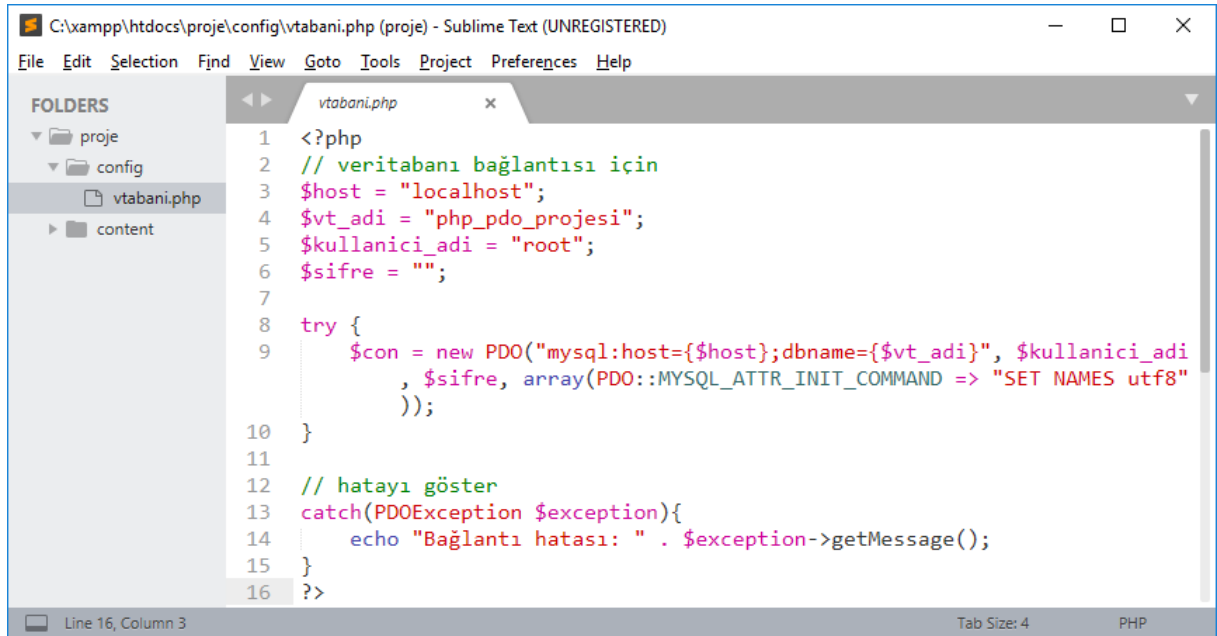
Artık PHP sayfalarını oluşturmaya hazırız. Öncelikle **XAMPP** paketinin kurulu olduğu klasörde **htdocs** klasörü içinde **proje** isimli yeni bir klasör oluşturun.



Proje klasörünün altında da **config** isimli bir klasör oluşturun ve içine aşağıdaki kodları içeren **vtabani.php** isimli dosyayı kaydedin. PHP sayfa kodlarımızı yazmak için **Sublime Text** isimli programı kullanacağız.

```
<?php
// veritabanı bağlantısı için gerekli parametreler
$host = "localhost";
$vt_adi = "php_pdo_projesi";
$kullanici_adi = "root";
$sifre = "";
```

```
try {
    $con = new PDO("mysql:host={$host};dbname={$vt_adi}", $kullanici_adi, $sifre,
array(PDO::MYSQL_ATTR_INIT_COMMAND => "SET NAMES utf8"));
}
// hatayı göster
catch(PDOException $exception){
    echo "Bağlantı hatası: " . $exception->getMessage();
}
?>
```



The screenshot shows a Sublime Text editor window titled "C:\xampp\htdocs\proje\config\vtabani.php (proje) - Sublime Text (UNREGISTERED)". The editor displays the following PHP code in a file named "vtabani.php":

```
1 <?php
2 // veritabanı bağlantısı için
3 $host = "localhost";
4 $vt_adi = "php_pdo_projesi";
5 $kullanici_adi = "root";
6 $sifre = "";
7
8 try {
9     $con = new PDO("mysql:host={$host};dbname={$vt_adi}", $kullanici_adi
, $sifre, array(PDO::MYSQL_ATTR_INIT_COMMAND => "SET NAMES utf8"
));
10 }
11
12 // hatayı göster
13 catch(PDOException $exception){
14     echo "Bağlantı hatası: " . $exception->getMessage();
15 }
16 ?>
```

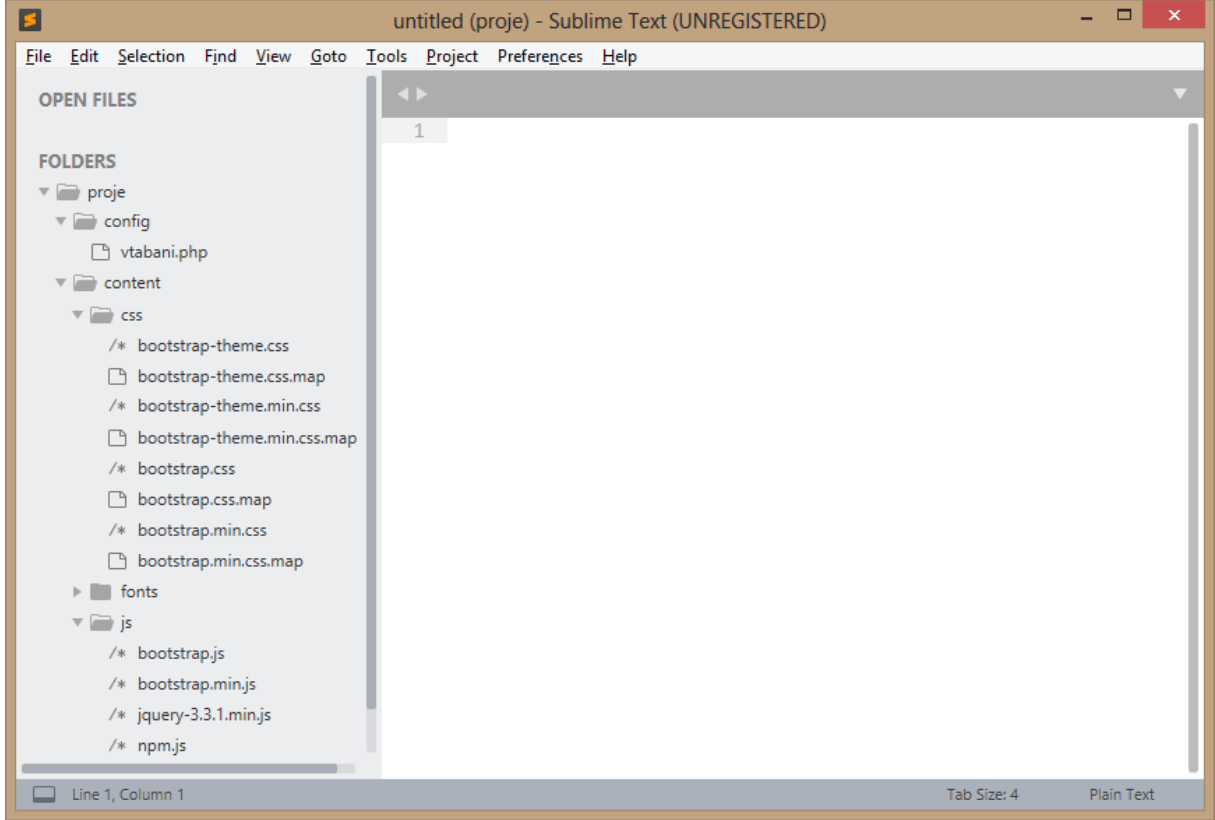
The editor interface includes a "FOLDERS" sidebar on the left showing the project structure: "proje" (expanded) containing "config" (expanded) and "vtabani.php", and "content". The status bar at the bottom indicates "Line 16, Column 3", "Tab Size: 4", and "PHP".

YÖNETİM PANELİNİ OLUŞTURMAK

ARAYÜZ TASARIMI İÇİN GEREKLİ DOSYALARIN KURULUMU

Web sayfalarının arayüzünü tasarlarken **Bootstrap** kütüphanesinden faydalanacağız. Önce proje klasörünün içine **content** isimli bir klasör oluşturun ve <https://getbootstrap.com/docs/3.3/> adresinden gereken dosyayı indirip oluşturduğunuz bu klasörün içine çıkartın.

Ardından <https://jquery.com/download/> adresinden **jQuery** kütüphanesinin dosyasını indirip **content/js** klasörüne kaydedin.



WEB SAYFALARININ ANA HTML ÇATISINI OLUŞTURMAK

CSS STİL DOSYASINI (STYLE.CSS) OLUŞTURMAK

Web sayfalarında etiketlere CSS stil sınıfları tanımlamak için kullanacağımız **style.css** dosyasını **content/css** klasörü içinde oluşturun ve içine aşağıdaki kodları yazın.

```
body.admin {
  min-height: 500px;
  padding-top: 50px;
  padding-bottom: 50px;
}
.m-t-1em { margin-top:1em; }
.m-r-1em { margin-right:1em; }
.m-b-1em { margin-bottom:1em; }
.m-l-1em { margin-left:1em; }
.mt0 { margin-top:0; }
```

Web sayfalarında tekrar eden ortak başlık (header) ve altbilgi (footer) etiketlerini ayrı sayfalar halinde oluşturacağız. Bu hem kod karmaşasını azaltacak hem de başlık ve altbilgi değişikliklerini daha esnek hale getirecektir.

BAŞLIK (HEADER.PHP) SAYFASINI OLUŞTURMAK

Sayfaların başında bir menü nesnemiz olacak. Bunun için **Bootstrap** kütüphanesinin **NavBar** bileşenini kullanacağız. Menü kodları ve üst kısımdaki HTML kodlar ortak olacağından hepsini **header.php** dosyasında toplayacağız. Önce proje klasörünün içine **admin** isimli bir klasör oluşturun. Aşağıdaki HTML kodları **header.php** dosyasına yazın ve **admin** klasörüne kaydedin.

```
<!doctype html>
<html lang="tr">
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>PHP - PDO Projesi</title>

  <!-- Bootstrap CSS dosyası -->
  <link rel="stylesheet" href="/proje/content/css/bootstrap.min.css" />
  <link rel="stylesheet" href="/proje/content/css/style.css" />
</head>
<body class="admin">
  <!-- Menü - Bootstrap Fixed Navbar -->
  <nav class="navbar navbar-default navbar-fixed-top">
    <div class="container">
      <div class="navbar-header">
        <button type="button" class="navbar-toggle collapsed" data-
toggle="collapse" data-target="#navbar" aria-expanded="false" aria-
controls="navbar">
          <span class="sr-only">Toggle navigation</span>
          <span class="icon-bar"></span>
          <span class="icon-bar"></span>
          <span class="icon-bar"></span>
        </button>
        <a class="navbar-brand" href="#">PHP - PDO Projesi</a>
      </div>
      <div id="navbar" class="navbar-collapse collapse">
        <ul class="nav navbar-nav">
          <li class="active"><a href="/proje/admin/index.php">Anasayfa</a></li>
          <li><a href="/proje/admin/urun/liste.php">Ürünler</a></li>
          <li><a href="/proje/admin/kategori/liste.php">Kategoriler</a></li>
        </ul>
        <ul class="nav navbar-nav navbar-right">
          <li><a href="#">Oturumu kapat</a></li>
        </ul>
      </div><!--/.nav-collapse -->
    </div>
  </nav>
  <!-- Menü sonu -->
```

Burada menü seçeneklerinin aktifliğini bizim program kodlarıyla kontrol etmememiz gerekiyor. Bunun için aktif adres satırını kontrol edip ilgili menüye **class="active"** parametresini verecek **navbar** kodlarını aşağıdaki gibi düzenleyin.

```
<!-- Menü - Fixed navbar -->
<nav class="navbar navbar-default navbar-fixed-top">
  <div class="container">
    <div class="navbar-header">
      <button type="button" class="navbar-toggle collapsed" data-
toggle="collapse" data-target="#navbar" aria-expanded="false" aria-
controls="navbar">
        <span class="sr-only">Toggle navigation</span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
      </button>
      <a class="navbar-brand" href="#">PHP - PDO Projesi</a>
```

```

</div>
<div id="navbar" class="navbar-collapse collapse">
  <ul class="nav navbar-nav">
    <!-- tarayıcının adres satırındaki url ifadesini okur
         ve buna göre ilgili menü seçeneğini aktifleştirir -->
    <?php $aktif_link = "http://$_SERVER[HTTP_HOST]$_SERVER[REQUEST_URI]";
?>
    <li <?php echo((strpos($aktif_link, 'index') !== false) ?
'class="active" : ''); ?>><a href="/proje/admin/index.php">Anasayfa</a></li>
    <li <?php echo((strpos($aktif_link, 'urun/') !== false) ?
'class="active" : ''); ?>><a href="/proje/admin/urun/liste.php">Ürünler</a></li>
    <li <?php echo((strpos($aktif_link, 'kategori/') !== false) ?
'class="active" : ''); ?>><a
href="/proje/admin/kategori/liste.php">Kategoriler</a></li>
  </ul>
  <ul class="nav navbar-nav navbar-right">
    <li><a href="#">Oturumu kapat</a></li>
  </ul>
</div><!--/.nav-collapse -->
</div>
</nav>
<!-- Menü sonu -->

```

ALTBİLGİ (FOOTER.PHP) SAYFASINI OLUŞTURMAK

Sayfaların sonunda bir altbilgi nesnemiz olacak. Bunun için yine **Bootstrap** kütüphanesinin **NavBar** bileşenini kullanacağız. Bu ve alt kısmındaki HTML kodlar ortak olacağından hepsini **footer.php** dosyasında toplayacağız. Aşağıdaki HTML kodları **footer.php** dosyasına yazın ve **proje/admin** klasörüne kaydedin.

```

<!-- Footer -->
<nav class="navbar navbar-default navbar-fixed-bottom">
  <div class="container m-t-1em">
    &copy; Murat SARIÖZ - Mercin KARAKAŞ
  </div>
</nav>
<!-- Footer sonu -->

<!-- jQuery ve Bootstrap için gerekli dosyalar -->
<script src="/proje/content/js/jquery-3.3.1.min.js"></script>
<script src="/proje/content/js/bootstrap.min.js"></script>
</body>
</html>

```

YÖNETİM PANELİ ANASAYFASINI (INDEX.PHP) OLUŞTURMAK

Web sayfalarımız için ortak olan başlık ve altbilgi alanlarını oluşturduk. Bundan sonra oluşturacağımız sayfalarda bu bilgileri ekleyerek işe başlayacağız. Anasayfamızı tasarlamak için **admin** klasörü içinde **index.php** isimli bir dosya oluşturun ve içine aşağıdaki satırları ekleyin.

```

<?php include "header.php"; ?>

<div class="container m-t-1em">
  <!-- Sayfa kodları bu alana eklenecek -->
</div> <!-- /container -->

<?php include "footer.php"; ?>

```

Ardından **container** sınıfı içinde **<!-- Sayfa kodları bu alana eklenecek -->** yazan açıklama satırının yerine aşağıdaki kodları yazın.

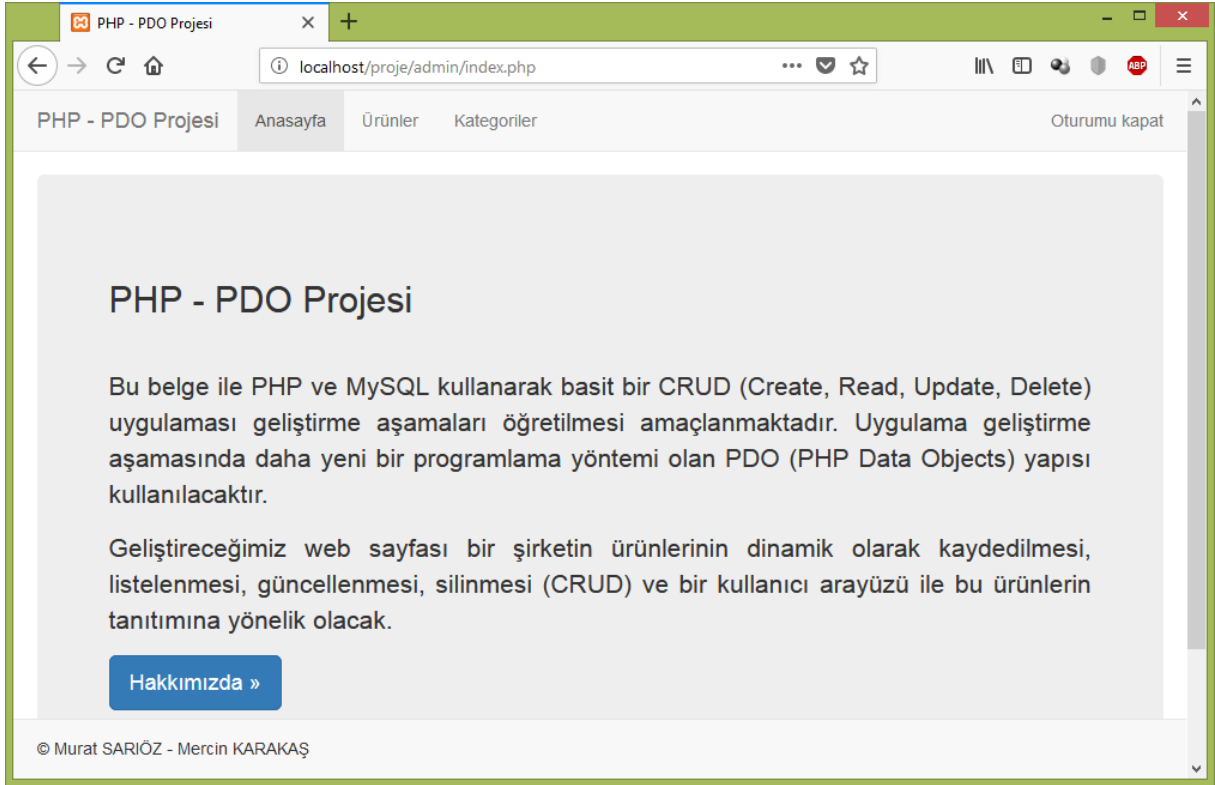
```

<!-- Proje hakkında kısa bir bilgi içeren anasayfa -->
<div class="jumbotron text-justify">

```

```
<div class="page-header"><h2>PHP - PDO Projesi</h2></div>
<p>Bu belge ile PHP ve MySQL kullanarak basit bir CRUD (Create, Read, Update, Delete) uygulaması geliştirme aşamaları öğretilmesi amaçlanmaktadır. Uygulama geliştirme aşamasında daha yeni bir programlama yöntemi olan PDO (PHP Data Objects) yapısı kullanılacaktır.</p>
<p>Geliştireceğimiz web sayfası bir şirketin ürünlerinin dinamik olarak kaydedilmesi, listelenmesi, güncellenmesi, silinmesi (CRUD) ve bir kullanıcı arayüzü ile bu ürünlerin tanıtımına yönelik olacak.</p>
<p><a class="btn btn-lg btn-primary" href="http://sudiozkanmtal.meb.k12.tr/" role="button">Hakkımızda &raquo;</a></p>
</div>
```

Tarayıcınızın adres satırına **http://localhost/proje/admin/index.php** yazarak sayfayı görüntüleyin.



ÜRÜN SAYFALARINI OLUŞTURMAK

ÜRÜN LİSTELEME SAYFASINI OLUŞTURMAK

Daha önce veritabanını oluştururken **urunler** tablosuna bazı örnek veriler de girmiştik. Ürün bilgilerini listeleyecek sayfamızı oluşturmaya başlamadan önce **admin** klasörü içinde **urun** isimli bir klasör oluşturun. Bu klasör içinde **liste.php** isimli bir dosya oluşturun ve içine aşağıdaki satırları ekleyin.

```
<?php include "../header.php"; ?>

<div class="container">

    <div class="page-header">
        <h1>Ürün Listesi</h1>
    </div>

    <!-- Kayıtları listeleyecek PHP kodları bu alana eklenecek -->

</div> <!-- /container -->

<?php include "../footer.php"; ?>

<!-- Kayıt silme onay kodları bu alana eklenecek -->
```

KAYIT LİSTELEMELİK İÇİN GEREKLİ SQL SORGUSUNU OLUŞTURMAK

Ardından **<!-- Kayıtları listeleyecek PHP kodları bu alana eklenecek -->** açıklama satırının yerine aşağıdaki PHP kodlarını yazın.

```
<?php
// veritabanı bağlantı dosyasını çağır
include '../../config/vtabani.php';

// silme mesajı burada yer alacak

// bütün kayıtları seç
$sorgu = "SELECT id, urunadi, aciklama, fiyat FROM urunler ORDER BY id DESC";
$stmt = $con->prepare($sorgu);
$stmt->execute();

// geriye dönen kayıt sayısı
$sayi = $stmt->rowCount();

// kayıt ekleme sayfasının linki
echo "<a href='ekle.php' class='btn btn-primary m-b-1em'>Yeni Ürün</a>";

//kayıt varsa listele
if($sayi>0){

    // kayıtlar burada listelenecek

}
// kayıt yoksa mesajla bildir
else{
    echo "<div class='alert alert-danger'>Listelenecek kayıt bulunamadı.</div>";
}
?>
```

KAYIT LİSTELEMELİK İÇİN GEREKLİ HTML TABLOSUNU OLUŞTURMAK

Sıra geldi veritabanından okunan kayıtların listelenmesine. Bunun için HTML tablo yapısını kullanacağız.

// kayıtlar burada listelenecek açıklama satırının yerine aşağıdaki PHP kodlarını yazın.

```
    echo "<table class='table table-hover table-responsive table-bordered'>";
//tablo başlangıcı

//tablo başlıkları
echo "<tr>";
    echo "<th>ID</th>";
    echo "<th>Ürün adı</th>";
    echo "<th>Açıklama</th>";
    echo "<th>Fiyat</th>";
    echo "<th>İşlem</th>";
echo "</tr>";

// tablo içeriği burada yer alacak

echo "</table>"; // tablo sonu
```

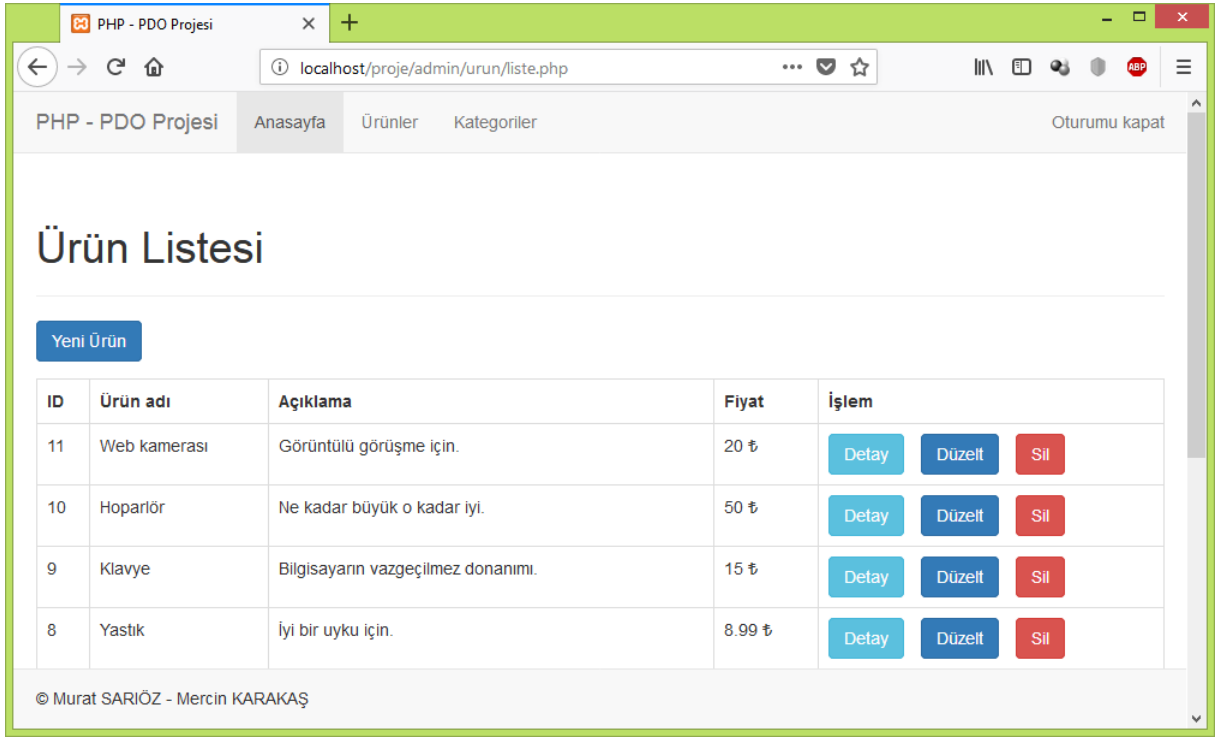
VERİTABANINDAKİ KAYITLARI LİSTELEMELİK İÇİN GEREKLİ PHP KODLARI

Tablonun ana yapısını oluşturduktan sonra, içine verileri dolduracak PHP kodlarını da // tablo içeriği burada yer alacak açıklama satırının yerine yazın.

```
// tablo verilerinin okunması
while ($kayit = $stmt->fetch(PDO::FETCH_ASSOC)){
    // tablo alanlarını değişkene dönüştürür
    // $kayit['urunadi'] => $urunadi
    extract($kayit);

    // her kayıt için yeni bir tablo satırı oluştur
    echo "<tr>";
        echo "<td>{$id}</td>";
        echo "<td>{$urunadi}</td>";
        echo "<td>{$aciklama}</td>";
        echo "<td>{$fiyat} &#8378;</td>"; // &#8378; ==> TL işareti
        echo "<td>";
            // kayıt detay sayfa bağlantısı
            echo "<a href='detay.php?id={$id}' class='btn btn-info m-r-1em'>Detay</a>";
            // kayıt güncelleme sayfa bağlantısı
            echo "<a href='duzelt.php?id={$id}' class='btn btn-primary m-r-1em'>Düzeltil</a>";
            // bu bağlantıyı dokümanın ileriki bölümlerinde işleyeceğiz
            echo "<a href='#' onclick='silme_onay({$id});' class='btn btn-danger'>Sil</a>";
        echo "</td>";
    echo "</tr>";
}
}
```

Tarayıcınızın adres satırına <http://localhost/proje/admin/urun/liste.php> yazarak ürün listeleme sayfasını görüntüleyin.



ÜRÜN SİLME SAYFASINI OLUŞTURMAK

Bir kaydı silmek için işlem butonlarından **Sil** butonuna tıklayınca önce kullanıcının onayını alacağız. Böylelikle yanlışlıkla kayıtların silinmesini önleyebiliriz. Silme işlemi onayını **liste.php** sayfasında silme işlemi ise kod karışıklığını azaltmak için ayrı bir sayfada (**sil.php**) yapacağız.

SİLME İŞLEMİNDEN ÖNCE KULLANICI ONAYINI ALMAK

Öncelikle **liste.php** dosyasını açın ve **<!-- Kayıt silme onay kodları bu alana eklenecek -->** açıklama satırının yerine aşağıdaki JavaScript kodlarını yazın.

```
<script type='text/javascript'>
// kayıt silme işlemi onayla
function silme_onay( id ){

    var cevap = confirm('Kaydı silmek istiyor musunuz?');
    if (cevap){
        // kullanıcı evet derse,
        // id bilgisini sil.php sayfasına yönlendirir
        window.location = 'sil.php?id=' + id;
    }
}
</script>
```

VERİTABANINDAN KAYIT SİLMEK İÇİN GEREKLİ PHP KODLARI

Şimdi **admin/urun** klasörü içinde **sil.php** isimli bir dosya oluşturun ve içine aşağıdaki PHP kodlarını yazın.

```
<?php
// veritabanı ayar dosyasını dahil et
include '../config/vtabani.php';

try {
    // kaydın id bilgisini al
    $id=isset($_GET['id']) ? $_GET['id'] : die('HATA: Id bilgisi bulunamadı.');
```

```

// silme sorgusu
$sorgu = "DELETE FROM urunler WHERE id = ?";
$stmt = $con->prepare($sorgu);
$stmt->bindParam(1, $id);

// sorguyu çalıştır
if($stmt->execute()){
    // kayıt listeleme sayfasına yönlendir
    // ve kullanıcıya kaydın silindiğini
    header('Location: liste.php?islem=silindi');
} // veya silinemediğini bildir
else{
    header('Location: liste.php?islem=silinemedi');
}
}
// hata varsa göster
catch(PDOException $exception){
    die('HATA: ' . $exception->getMessage());
}
?>

```

SİLME İŞLEMİNDEN SONRA KULLANICIYA BİLGİ VERMEK

Silme işleminden sonra kullanıcıyı bilgilendirmek için ürün listeleme sayfasında düzenleme yapacağız. **liste.php** dosyasını açın ve **// silme mesajı burada yer alacak** açıklama satırının yerine aşağıdaki kodları ekleyin.

```

$islem = isset($_GET['islem']) ? $_GET['islem'] : "";

// eğer silme (sil.php) sayfasından yönlendirme yapıldıysa
if($islem=='silindi'){
    echo "<div class='alert alert-success'>Kayıt silindi.</div>";
}
else if($islem=='silinemedi'){
    echo "<div class='alert alert-danger'>Kayıt silinemedi.</div>";
}

```

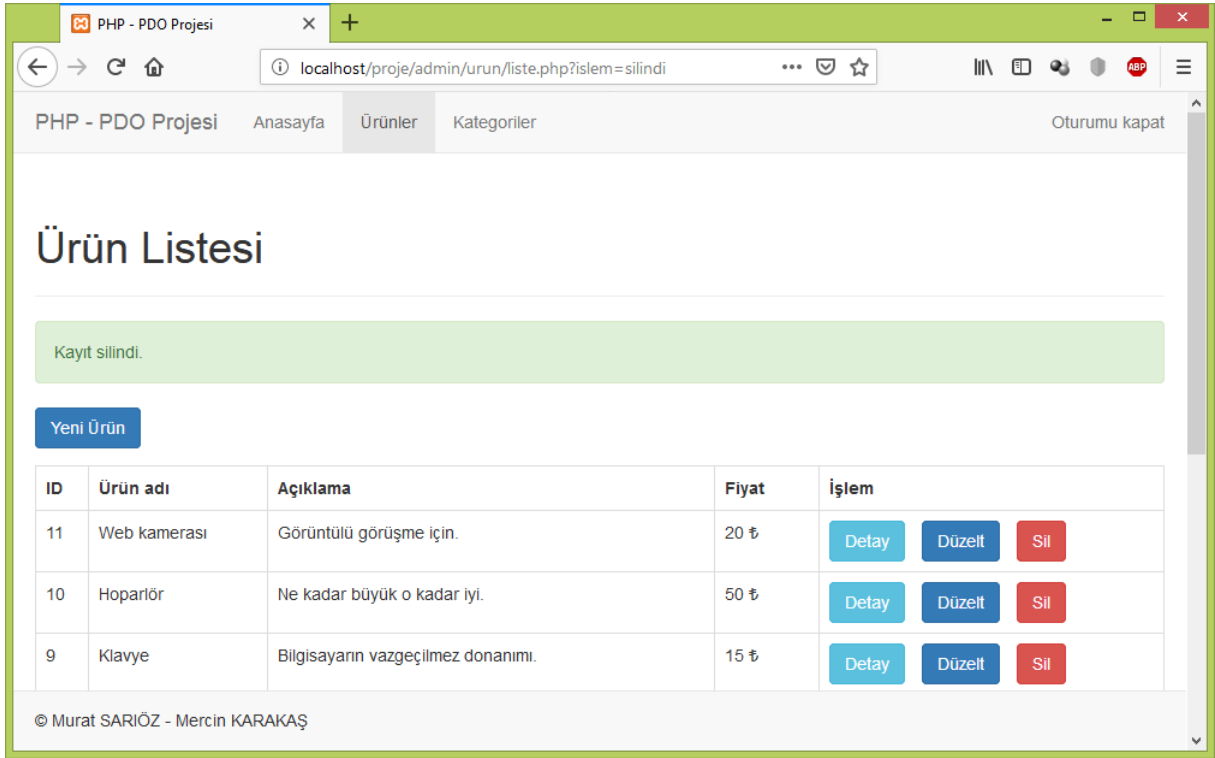
Tarayıcınızın adres satırına **http://localhost/proje/admin/urun/liste.php** yazarak ürün listeleme sayfasını görüntüleyin. Herhangi bir kaydı silmek için karşısındaki **Sil** butonuna tıklayın.

The screenshot shows a web browser window displaying a product list. A modal dialog box is open, asking for confirmation to delete a record. The dialog box contains the text "Kaydı silmek istiyor musunuz?" and two buttons: "Tamam" (Yes) and "Vazgeç" (No). The background shows a table with columns for ID, Ürün adı, Açıklama, Fiyat, and İşlem. The table contains four rows of product data.

ID	Ürün adı	Açıklama	Fiyat	İşlem
12	Bayan Şemsiyesi	Bahar yağmurlarıyla ıslanmak istemeyenler için.	25 ₺	Detay Düzelt Sil
11	Web kamerası	Görüntülü görüşme için.	20 ₺	Detay Düzelt Sil
10	Hoparlör	Ne kadar büyük o kadar iyi.	50 ₺	Detay Düzelt Sil
9	Klavye	Bilgisayarı vazgeçilmez donanımı.	15 ₺	Detay Düzelt Sil

© Murat SARIOZ - Mercin KARAKAŞ

Silme işlemini **Tamam** butonuyla onaylayın.



ÜRÜN DETAY SAYFASINI OLUŞTURMAK

Ürün listeleme sayfasında **Detay** işlem butonu tıklanınca kayıtların detay bilgisini görüntülemek için kullanıcıyı yönlendireceğimiz sayfayı **detay.php** ismiyle **admin/urun** klasöründe oluşturun ve içine aşağıdaki kodları yazın.

```
<?php include "../header.php"; ?>

<div class="container">

    <div class="page-header">
        <h1>Ürün Bilgisi</h1>
    </div>

    <!-- ürün bilgilerini getiren PHP kodu burada yer alacak -->

    <!-- ürün bilgilerini görüntüleyen HTML tablosu burada yer alacak -->

</div> <!-- container -->

<?php include "../footer.php"; ?>
```

ÜRÜN BİLGİLERİNİ VERİTABANINDAN OKUMAK

Aşağıdaki PHP kodları MySQL veritabanındaki **urunler** tablosunda kayıtlı ilgili (**Id** parametre değerine karşılık gelen) ürün bilgisini getirecektir. Bu kodları **<!-- ürün bilgilerini getiren PHP kodu burada yer alacak -->** açıklama satırının yerine yazın.

```
<?php
// gelen Id parametresini al
// isset() bir değer olup olmadığını kontrol eden PHP fonksiyonudur
$id=isset($_GET['id']) ? $_GET['id'] : die('HATA: Kayıt bulunamadı.');
```

```
// veritabanı bağlantı dosyasını çağır
include '../config/vtabani.php';
```

```

// aktif kayıt bilgilerini oku
try {
    // seçme sorgusunu hazırla
    $sorgu = "SELECT id, urunadi, aciklama, fiyat FROM urunler WHERE id = ?
LIMIT 0,1";
    $stmt = $con->prepare( $sorgu );

    // Id parametresini bağla
    $stmt->bindParam(1, $id);

    // sorguyu çalıştır
    $stmt->execute();

    // gelen kaydı bir değişkende sakla
    $kayit = $stmt->fetch(PDO::FETCH_ASSOC);

    // tabloya yazılacak bilgileri değişkenlere doldur
    $urunadi = $kayit['urunadi'];
    $aciklama = $kayit['aciklama'];
    $fiyat = $kayit['fiyat'];
}

// hatayı göster
catch(PDOException $exception){
    die('HATA: ' . $exception->getMessage());
}
?>

```

ÜRÜN BİLGİLERİNİ GÖRÜNTÜLEMEK İÇİN GEREKLİ HTML TABLOSUNU OLUŞTURMAK

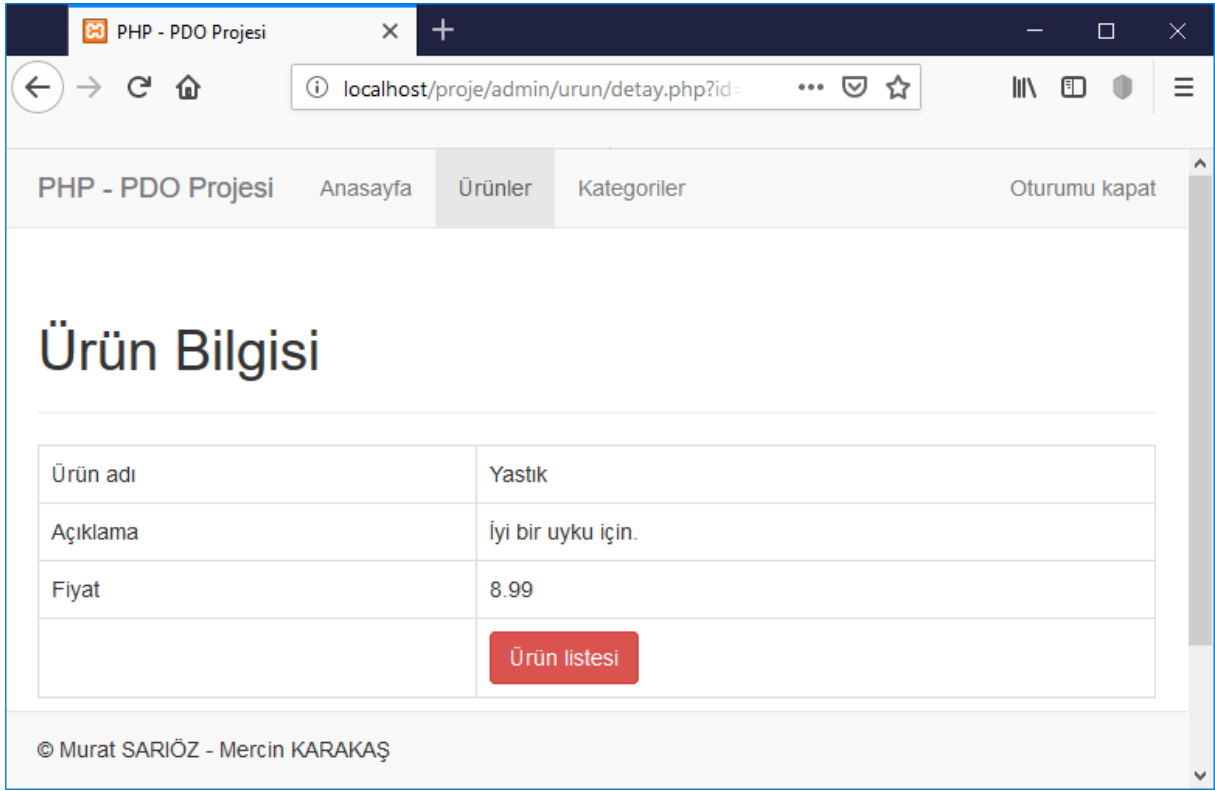
Aşağıdaki kodlar veritabanındaki ürün bilgilerinin görüntülenmesi için gerekli HTML tablosunu oluşturacaktır. Bu kodları `<!-- ürün bilgilerini görüntüleyen HTML tablosu burada yer alacak -->` açıklama satırının yerine yazın.

```

<!-- kayıt bilgilerini görüntüleyen HTML tablosu -->
<table class='table table-hover table-responsive table-bordered'>
    <tr>
        <td>Ürün adı</td>
        <td><?php echo htmlspecialchars($urunadi, ENT_QUOTES); ?></td>
    </tr>
    <tr>
        <td>Açıklama</td>
        <td><?php echo htmlspecialchars($aciklama, ENT_QUOTES); ?></td>
    </tr>
    <tr>
        <td>Fiyat</td>
        <td><?php echo htmlspecialchars($fiyat, ENT_QUOTES); ?> &#8378;</td>
    </tr>
    <tr>
        <td></td>
        <td>
            <a href='liste.php' class='btn btn-danger'>Ürün listesi</a>
        </td>
    </tr>
</table>

```

Tarayıcınızın adres satırına <http://localhost/proje/admin/urun/liste.php> yazarak ürün listeleme sayfasını görüntüleyin. Herhangi bir kaydın bilgilerini görüntülemek için karşısındaki **Detay** butonuna tıklayın.



ÜRÜN EKLEME SAYFASINI OLUŞTURMAK

Yeni kayıt eklemek için kullanılacak **ekle.php** sayfasını **admin/urun** klasöründe oluşturun ve aşağıdaki HTML kodlarını içine yazın.

```
<?php include "../header.php"; ?>

<div class="container">

    <div class="page-header">
        <h1>Ürün Ekle</h1>
    </div>

    <!-- PHP kayıt ekleme kodları burada yer alacak -->

    <!-- Ürün eklemek için kullanılacak html formu burada yer alacak -->

</div> <!-- container -->

<?php include "../footer.php"; ?>
```

YENİ KAYIT GİRİŞİ İÇİN GEREKLİ HTML FORMUNU OLUŞTURMAK

Aşağıdaki kodlar veritabanındaki alanlara karşılık gelen bilgilerin girilmesi için gerekli HTML formunu oluşturacaktır. Bu kodları **<!-- Ürün eklemek için kullanılacak html formu burada yer alacak -->** açıklama satırının yerine yazın.

```
<!-- Ürün bilgilerini girmek için kullanılacak html formu -->
<form action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>"
method="post">
    <table class='table table-hover table-responsive table-bordered'>
        <tr>
            <td>Ürün adı</td>
            <td><input type='text' name='urunadi' class='form-control' /></td>
        </tr>
        <tr>
            <td>Açıklama</td>
```

```

        <td><textarea name='aciklama' class='form-control'></textarea></td>
    </tr>
    <tr>
        <td>Fiyat</td>
        <td><input type='text' name='fiyat' class='form-control' /></td>
    </tr>
    <tr>
        <td></td>
        <td>
            <input type='submit' value='Kaydet' class='btn btn-primary' />
            <a href='liste.php' class='btn btn-danger'>Ürün listesi</a>
        </td>
    </tr>
</table>
</form>

```

VERİTABANINA YENİ KAYIT EKLEMEK İÇİN GEREKLİ PHP KODLARI

Kullanıcı **ekle.php** sayfasında bilgileri girip **Kaydet** butonuna tıklayınca aşağıdaki PHP kodları MySQL veritabanındaki **urunler** tablosuna bu bilgileri kaydedecektir. Bu kodları **<!-- PHP kayıt ekleme kodları burada yer alacak -->** açıklama satırının yerine yazın.

```

<?php
if($_POST){

    // veritabanı yapılandırma dosyasını dahil et
    include '../config/vtabani.php';

    try{
        // kayıt ekleme sorgusu
        $sorgu = "INSERT INTO urunler SET urunadi=:urunadi, aciklama=:aciklama,
fiyat=:fiyat, giris_tarihi=:giris_tarihi";

        // sorguyu hazırla
        $stmt = $con->prepare($sorgu);

        // post edilen değerler
        $urunadi=htmlspecialchars(strip_tags($_POST['urunadi']));
        $aciklama=htmlspecialchars(strip_tags($_POST['aciklama']));
        $fiyat=htmlspecialchars(strip_tags($_POST['fiyat']));

        // parametreleri bağla
        $stmt->bindParam(':urunadi', $urunadi);
        $stmt->bindParam(':aciklama', $aciklama);
        $stmt->bindParam(':fiyat', $fiyat);

        // ürünün veritabanına kaydedildiği tarihi belirt
        $giris_tarihi=date('Y-m-d H:i:s');
        $stmt->bindParam(':giris_tarihi', $giris_tarihi);

        // sorguyu çalıştır
        if($stmt->execute()){
            echo "<div class='alert alert-success'>Ürün kaydedildi.</div>";
        }else{
            echo "<div class='alert alert-danger'>Ürün kaydedilemedi.</div>";
        }
    }
    // hatayı göster
    catch(PDOException $exception){
        die('ERROR: ' . $exception->getMessage());
    }
}
?>

```


Tarayıcınızın adres satırına **http://localhost/proje/admin/urun/ekle.php** yazarak ürün ekleme sayfasını görüntüleyin.

Ürün Ekle

Ürün adı	Şemsiye
Açıklama	Bahar yağmurlarıyla ıslanmak istemeyenler için.
Fiyat	25

[Kaydet](#) [Ürün listesi](#)

© Murat SARIÖZ - Mercin KARAKAŞ

Formu doldurun ve **Kaydet** butonuyla yeni bir kayıt ekleyin.

Ürün Ekle

Ürün kaydedildi.

Ürün adı	
Açıklama	
Fiyat	

[Kaydet](#) [Ürün listesi](#)

© Murat SARIÖZ - Mercin KARAKAŞ

ÜRÜN GÜNCELLEME (DÜZELTME) SAYFASINI OLUŞTURMAK

Ürün listeleme sayfasında **Düzeltil** işlem butonu tıklanınca kayıtların bilgilerini bir form ile görüntülemek ve güncellemek için kullanıcıyı yönlendireceğimiz sayfayı **duzelt.php** ismiyle **admin/urun** klasöründe oluşturun ve içine aşağıdaki kodları yazın.

```
<?php include "../header.php"; ?>

<div class="container">

    <div class="page-header">
```

```
<h1>Ürün Güncelleme</h1>
</div>

<!-- ürün bilgilerini getiren PHP kodu burada yer alacak -->

<!-- ürün bilgilerini düzeltme HTML formu burada yer alacak -->

</div> <!-- container -->

<?php include "../footer.php"; ?>
```

GÜNCELLENECEK ÜRÜN BİLGİLERİNİ VERİTABANINDAN OKUMAK

Öncelikle ürün listeleme sayfasından gelen **Id** parametresini **\$_GET['id']** fonksiyonu ile alıp **\$id** değişkenine kaydediyoruz. Daha sonra bu parametreye karşılık gelen kayıt bilgilerini veritabanından okuyup HTML formunu bu bilgilerle dolduruyoruz.

Aşağıdaki kodları **<!-- ürün bilgilerini getiren PHP kodu burada yer alacak -->** açıklama satırının yerine yazın.

```
<?php
// gelen parametre değerini oku, bizim örneğimizde bu Id bilgisidir
$id=isset($_GET['id']) ? $_GET['id'] : die('HATA: Id bilgisi bulunamadı.');
```

```
// veritabanı bağlantı dosyasını dahil et
include '../config/vtabani.php';

// aktif kayıt bilgilerini oku
try {
// seçme sorgusunu hazırla
    $sorgu = "SELECT id, urunadi, aciklama, fiyat FROM urunler WHERE id = ?
LIMIT 0,1";
    $stmt = $con->prepare( $sorgu );

// id parametresini bağla (? işaretini id değeri ile değiştir)
    $stmt->bindParam(1, $id);

// sorguyu çalıştır
    $stmt->execute();

// okunan kayıt bilgilerini bir değişkene kaydet
    $kayit = $stmt->fetch(PDO::FETCH_ASSOC);

// formu dolduracak değişken bilgileri
    $urunadi = $kayit['urunadi'];
    $aciklama = $kayit['aciklama'];
    $fiyat = $kayit['fiyat'];
}
// hatayı göster
catch(PDOException $exception){
    die('HATA: ' . $exception->getMessage());
}
?>
```

KAYIT GÜNCELLEMEK İÇİN GEREKLİ HTML FORMUNU OLUŞTURMAK

Aşağıdaki kodlar eski kayıt bilgilerini görüntüleyip yeni bilgilerin girilmesi için gerekli HTML formunu oluşturacaktır. Bu kodları **<!-- ürün bilgilerini düzeltme HTML formu burada yer alacak -->** açıklama satırının yerine yazın.

```
<!-- kaydı güncelleyecek PHP kodu burada yer alacak -->
```

```

<!-- kayıt bilgilerini güncelleyebileceğimiz HTML formu -->
<form action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]) .
"?id={ $id}");?>" method="post">
  <table class='table table-hover table-responsive table-bordered'>
    <tr>
      <td>Ürün adı</td>
      <td><input type='text' name='urunadi' value="<?php echo
htmlspecialchars($urunadi, ENT_QUOTES); ?>" class='form-control' /></td>
    </tr>
    <tr>
      <td>Açıklama</td>
      <td><textarea name='aciklama' class='form-control'><?php echo
htmlspecialchars($aciklama, ENT_QUOTES); ?></textarea></td>
    </tr>
    <tr>
      <td>Fiyat</td>
      <td><input type='text' name='fiyat' value="<?php echo
htmlspecialchars($fiyat, ENT_QUOTES); ?>" class='form-control' /></td>
    </tr>
    <tr>
      <td></td>
      <td>
        <input type='submit' value='Kaydet' class='btn btn-primary' />
        <a href='liste.php' class='btn btn-danger'>Ürün listesi</a>
      </td>
    </tr>
  </table>
</form>

```

KAYIT GÜNCELLEME KODLARI

Kullanıcı formda bilgileri değiştirip **Kaydet** butonuna tıklarsa aşağıdaki kodlar bu değişiklikleri veritabanına kaydedecektir. Bu kodları `<!-- kaydı güncelleyecek PHP kodu burada yer alacak -->` açıklama satırının yerine yazın.

```

<?php
// Kaydet butonu tıklanmışsa
if($_POST){
    try{
        // güncelleme sorgusu
        // çok fazla parametre olduğundan karışmaması için
        // soru işaretleri yerine etiketler kullanacağız
        $sorgu = "UPDATE urunler SET urunadi=:urunadi, aciklama=:aciklama,
fiyat=:fiyat WHERE id = :id";

        // sorguyu hazırla
        $stmt = $con->prepare($sorgu);

        // gelen bilgileri değişkenlere kaydet
        $urunadi=htmlspecialchars(strip_tags($_POST['urunadi']));
        $aciklama=htmlspecialchars(strip_tags($_POST['aciklama']));
        $fiyat=htmlspecialchars(strip_tags($_POST['fiyat']));

        // parametreleri bağla
        $stmt->bindParam(':urunadi', $urunadi);
        $stmt->bindParam(':aciklama', $aciklama);
        $stmt->bindParam(':fiyat', $fiyat);
        $stmt->bindParam(':id', $id);

        // sorguyu çalıştır
        if($stmt->execute()){
            echo "<div class='alert alert-success'>Kayıt güncellendi.</div>";
        }else{

```

```
        echo "<div class='alert alert-danger'>Kayıt
güncellenemedi.</div>";
    }

}
// hata varsa göster
catch(PDOException $exception){
    die('HATA: ' . $exception->getMessage());
}
}
?>
```

Tarayıcınızın adres satırına **http://localhost/proje/admin/urun/liste.php** yazarak herhangi bir ürün için **Düzeltil** butonunu tıklayın ve ürün güncelleme sayfasını görüntüleyin. Bilgilerde değişiklik yaparak **Kaydet** butonunu tıklayın.

PHP - PDO Projesi

localhost/proje/admin/urun/duzelt.php?id=12

PHP - PDO Projesi Anasayfa Ürünler Kategoriler Oturumu kapat

Ürün Güncelleme

Kayıt güncellendi.

Ürün adı	<input type="text" value="Bayan Şemsiyesi"/>
Açıklama	<input type="text" value="Bahar yağmurlarıyla ıslanmak istemeyenler için."/>
Fiyat	<input type="text" value="25"/>

© Murat SARIÖZ - Mercin KARAKAŞ

SAYFALANDIRMA İŞLEMİ

Ürün listeleme sayfasında tüm ürünleri tek sayfada görüntülemek hem sayfa yükleme hızını etkileyecek hem de aranan ürünü bulmayı zorlaştıracaktır. Bu olumsuzlukları gidermek için listeleme işlemi sayfa sayfa yapacağız. Bunun için daha önce yazdığımız bazı kodlarda değişiklik yapmamız lazım.

SAYFALAMA DEĞİŞKENLERİNİ TANIMLAMAK

Aşağıdaki kod satırlarını **liste.php** sayfasındaki **include** '../../config/vtabani.php'; satırının üst tarafına ekleyin.

```
// SAYFALANDIRMA DEĞİŞKENLERİ
// sayfa parametresi aktif sayfa numarasını gösterir, parametre boşsa değeri
1'dir
$sayfa = isset($_GET['sayfa']) ? $_GET['sayfa'] : 1;

// bir sayfada görüntülenecek kayıt sayısı
$sayfa_kayit_sayisi = 5;

// sorgudaki LIMIT başlangıç değerini hesapla
$ilk_kayit_no = ($sayfa_kayit_sayisi * $sayfa) - $sayfa_kayit_sayisi;
```

SEÇME SORGUSUNU DÜZENLEMEK

Hala **liste.php** sayfasında yapmamız gerekenler var. Sayfadaki seçme sorgusunu aşağıdaki gibi yeniden düzenleyin.

Eski seçme sorgusu:

```
// bütün kayıtları seç
$sorgu = "SELECT id, urunadi, aciklama, fiyat FROM urunler ORDER BY id DESC";
$stmt = $con->prepare($sorgu);
$stmt->execute();
```

Yeni seçme sorgusu:

```
// sayfada görüntülenecek kayıtları seç
$sorgu = "SELECT id, urunadi, aciklama, fiyat FROM urunler ORDER BY id DESC
LIMIT :ilk_kayit_no, :sayfa_kayit_sayisi";

$stmt = $con->prepare($sorgu);
$stmt->bindParam(":ilk_kayit_no", $ilk_kayit_no, PDO::PARAM_INT);
$stmt->bindParam(":sayfa_kayit_sayisi", $sayfa_kayit_sayisi, PDO::PARAM_INT);
$stmt->execute();
```

KAYIT SAYISINI HESAPLAMAK

Toplam kayıt sayısını hesaplayarak kaç tane liste sayfası oluşturacağımızı belirleyebiliriz. Aşağıdaki kodları tablo kapatma etiketinden (**</table>**) sonra yazın.

```
// SAYFALANDIRMA
// toplam kayıt sayısını hesapla
$sorgu = "SELECT COUNT(*) as kayit_sayisi FROM urunler";
$stmt = $con->prepare($sorgu);

// sorguyu çalıştır
$stmt->execute();

// kayıt sayısını oku
$kayit = $stmt->fetch(PDO::FETCH_ASSOC);
$kayit_sayisi = $kayit['kayit_sayisi'];
```

SAYFALANDIRMA DOSYASINI (PAGING.PHP) OLUŞTURMAK

Aşağıdaki satırları az önce yazdığımız kodlardan hemen sonra yazın.

```
// kayıtları sayfalandır
$sayfa_url="liste.php";
include_once "../sayfalama.php";
```

Burada **\$sayfa_url** değişkenini kullanmamızdaki amaç **sayfalama.php** dosyasının yeniden kullanılabilirliğini sağlamaktır. Yani diğer liste sayfalarındaki sayfalendirme işlemleri için de bu dosyayı kullanabiliriz. Örneğin kategori liste sayfasında.

admin klasörü içinde **sayfalama.php** dosyasını oluşturun ve içine aşağıdaki kod satırlarını yazın.

```
<?php
echo "<ul class='pagination pull-left margin-zero mt0'>";

// önceki sayfa butonu burada yer alacak

// tıklanabilir sayfa numaraları burada yer alacak

// sonraki sayfa butonu burada yer alacak

echo "</ul>";
?>

<!-- sayfa numaralarını listeleyen drop-down menü burada yer alacak -->
```

ÖNCEKİ SAYFA BUTONUNU OLUŞTURMAK

sayfalama.php dosyasındaki **// önceki sayfa butonu burada yer alacak** satırı yerine aşağıdaki kodları yazın.

```
// önceki sayfa butonu
if($sayfa>1){

    $onceki_sayfa = $sayfa - 1;
    echo "<li>";
        echo "<a href='{ $sayfa_url }?sayfa={ $onceki_sayfa }'>";
            echo "<span style='margin:0 .5em;'>&laquo;</span>";
        echo "</a>";
    echo "</li>";
}
```

NUMARALANMIŞ SAYFA BUTONLARINI OLUŞTURMAK

sayfalama.php dosyasındaki **// tıklanabilir sayfa numaraları burada yer alacak** satırı yerine aşağıdaki kodları yazın.

```
// tıklanabilir sayfa numaraları

// sayfa sayısını hesapla
$sayfa_sayisi = ceil($kayit_sayisi / $sayfa_kayit_sayisi);

// aktif sayfanın öncesinde ve sonrasında gösterilecek sayfa numarası aralığı
$aralik = 2;

// aktif sayfanın önce ve sonrasındaki sayfa numaralarını görüntüle
$baslangic_no = $sayfa - $aralik;
$bitis_no = ($sayfa + $aralik) + 1;

for ($x=$baslangic_no; $x<$bitis_no; $x++) {
```

```

// $x deęerinin 0'dan byk VE $sayfa_sayisi'na eřit veya kk olduęundan
emin ol
if (($x > 0) && ($x <= $sayfa_sayisi)) {

    // aktif sayfa
    if ($x == $sayfa) {
        echo "<li class='active'>";
        echo "<a href='javascript::void();'>{$x}</a>";
        echo "</li>";
    }
    // aktif olmayan sayfa
    else {
        echo "<li>";
        echo " <a href='{ $sayfa_url }?sayfa={$x}'>{$x}</a> ";
        echo "</li>";
    }
}
}
}

```

SONRAKİ SAYFA BUTONUNU OLUŐTURMAK

sayfalama.php dosyasındaki **// sonraki sayfa butonu burada yer alacak** satırı yerine aŐaęıdaki kodları yazın.

```

// sonraki sayfa butonu
if($sayfa<$sayfa_sayisi){
    $sonraki_sayfa = $sayfa + 1;

    echo "<li>";
    echo "<a href='{ $sayfa_url }?sayfa={ $sonraki_sayfa }'>";
    echo "<span style='margin:0 .5em;'>&raquo;</span>";
    echo "</a>";
    echo "</li>";
}

```

Tarayıcınızın adres satırına **http://localhost/proje/admin/urun/liste.php** yazarak rn liste sayfasını grntleyin. Sayfa numaralarına tıklayarak sayfalar arasında geiŐ yapın.

ID	rn adı	Aıklama	Fiyat	İŐlem
8	Yastık	İyi bir uyku iin.	8.99 ₺	Detay Dzelt Sil
7	Kulaklık	Mzik severler iin.	7 ₺	Detay Dzelt Sil
6	Mouse	Bilgisayar kullanmayı sevenler iin.	11.35 ₺	Detay Dzelt Sil

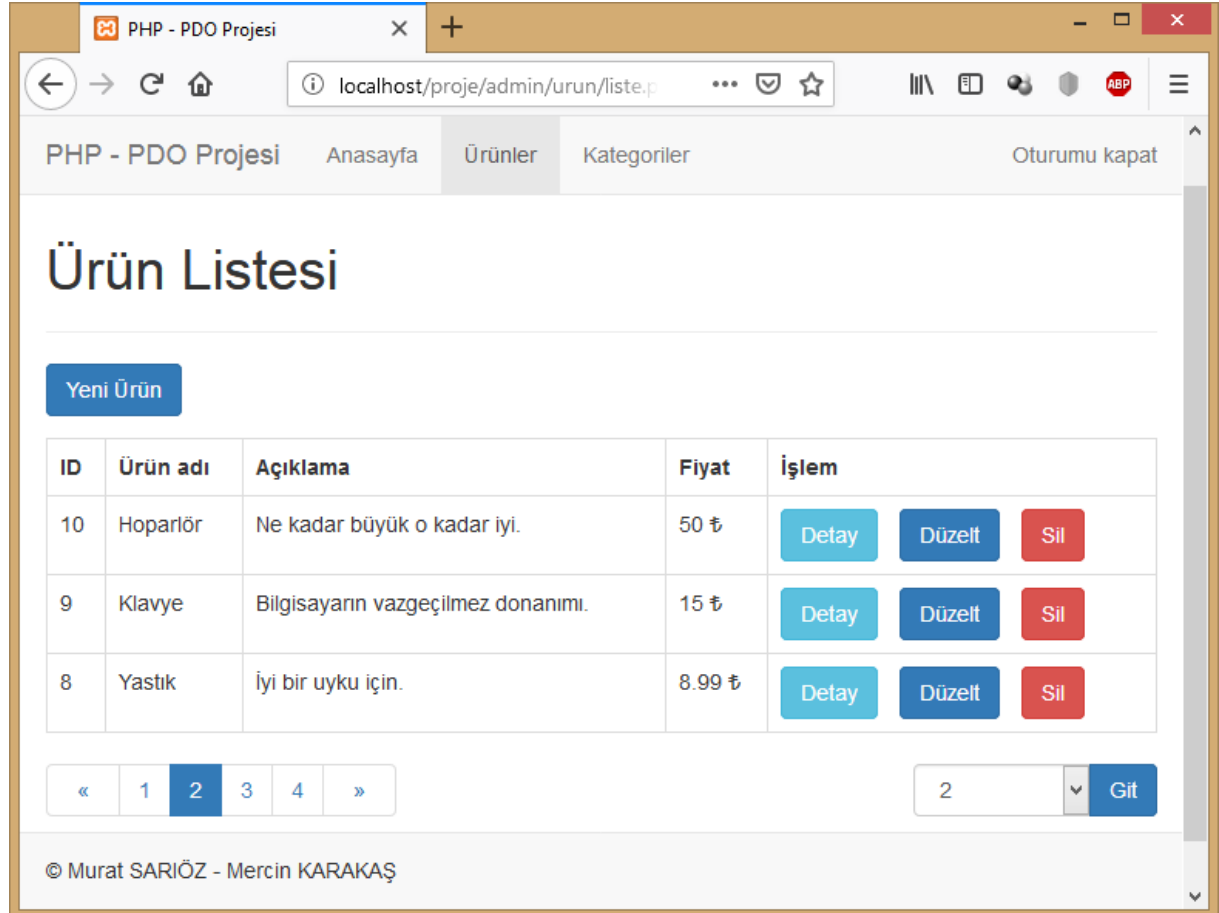
« 1 2 3 4 »

© Murat SARIZ - Mercin KARAKAŐ

sayfalama.php dosyasındaki `<!-- sayfa numaralarını listeleyen drop-down menü burada yer alacak -->` satırını yerine aşağıdaki kodları yazın.

```
<!-- sayfa yönlendirme menüsü -->
<form action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>" method="get">
  <div class="row">
    <div class="col-xs-3 col-md-2 pull-right">
      <div class="input-group">
        <select name="sayfa" class="form-control">
          <?php
            for($i=1; $i<=$sayfa_sayisi; $i++) {
              echo "<option value=" . $i . ($i == $sayfa ? " selected" :
                "")." . $i . "</option>";
            }
          <?>
        </select>
        <div class="input-group-btn">
          <button class="btn btn-primary" type="submit">
            <span>Git</span>
          </button>
        </div>
      </div>
    </div>
  </div>
</form>
```

Tarayıcınızın adres satırına <http://localhost/proje/admin/urun/liste.php> yazarak ürün liste sayfasını görüntüleyin.



Git butonunun önünde yer alan aşağı ok görsel açıdan estetik görünmediği için bunu stil uygulayarak gizleyeceğiz. Bunun için **content/css** klasöründeki **style.css** dosyasını açın ve içine aşağıdaki satırları ekleyin.

```
select.no-arrow{
```



```
-moz-appearance: none;  
-webkit-appearance: none;  
appearance: none;  
}
```

Ardından `<select>` etiketinin `class` parametresine oluşturduğunuz `no-arrow` sınıfını ekleyin.

Eski kod:

```
<select name="sayfa" class="form-control">
```

Yeni kod:

```
<select name="sayfa" class="form-control no-arrow">
```

Tarayıcınızın adres satırına `http://localhost/proje/admin/urun/liste.php` yazarak ürün liste sayfasını görüntüleyin.

PHP - PDO Projesi

localhost/proje/admin/urun/liste.p

PHP - PDO Projesi Anasayfa Ürünler Kategoriler Oturumu kapat

Ürün Listesi

Yeni Ürün

ID	Ürün adı	Açıklama	Fiyat	İşlem
10	Hoparlör	Ne kadar büyük o kadar iyi.	50 ₺	Detay Düzeltil Sil
9	Klavye	Bilgisayarın vazgeçilmez donanımı.	15 ₺	Detay Düzeltil Sil
8	Yastık	İyi bir uyku için.	8.99 ₺	Detay Düzeltil Sil

« 1 2 3 4 »

2 Göt

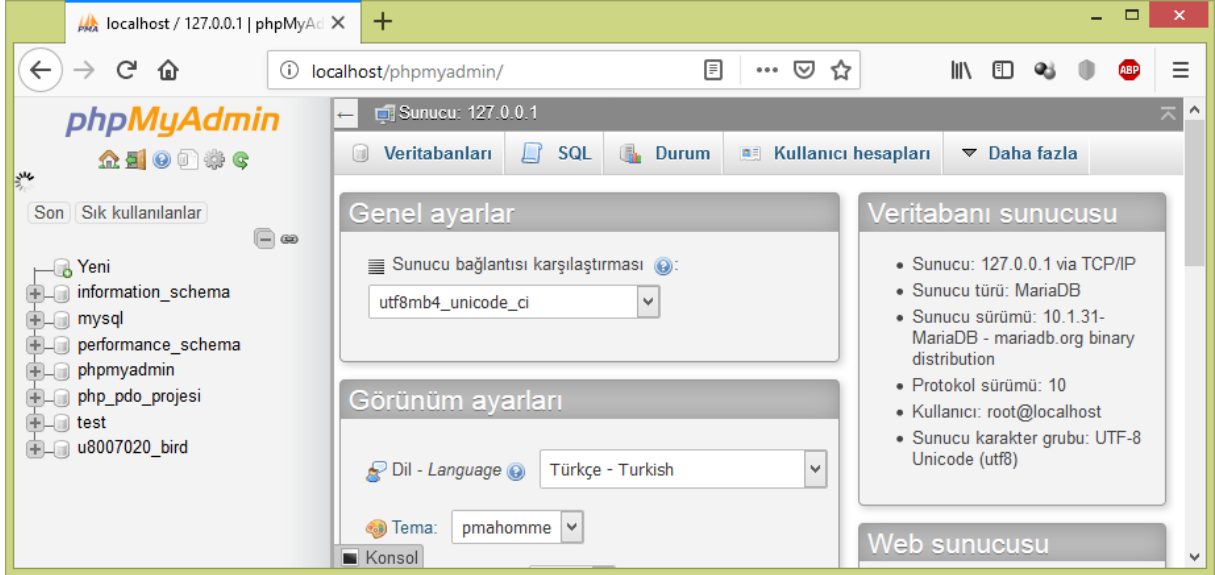
© Murat SARIÖZ - Mercin KARAKAŞ

RESİM YÜKLEMEK

Ürün ekleme sayfasında ürünle ilgili bir fotoğraf eklemek istersek öncelikle veritabanında **urunler** tablosunda yeni bir alan oluşturmamız gerekir. Bu alana yüklediğimiz fotoğrafın adını kaydedeceğiz. Yüklenen fotoğraf ise **content/images** klasörüne kaydedilecek. Proje ana klasöründeki **content** klasörü içinde **images** isminde bir klasör oluştururuz.

VERİTABANINA YENİ BİR ALAN EKLEMELİK

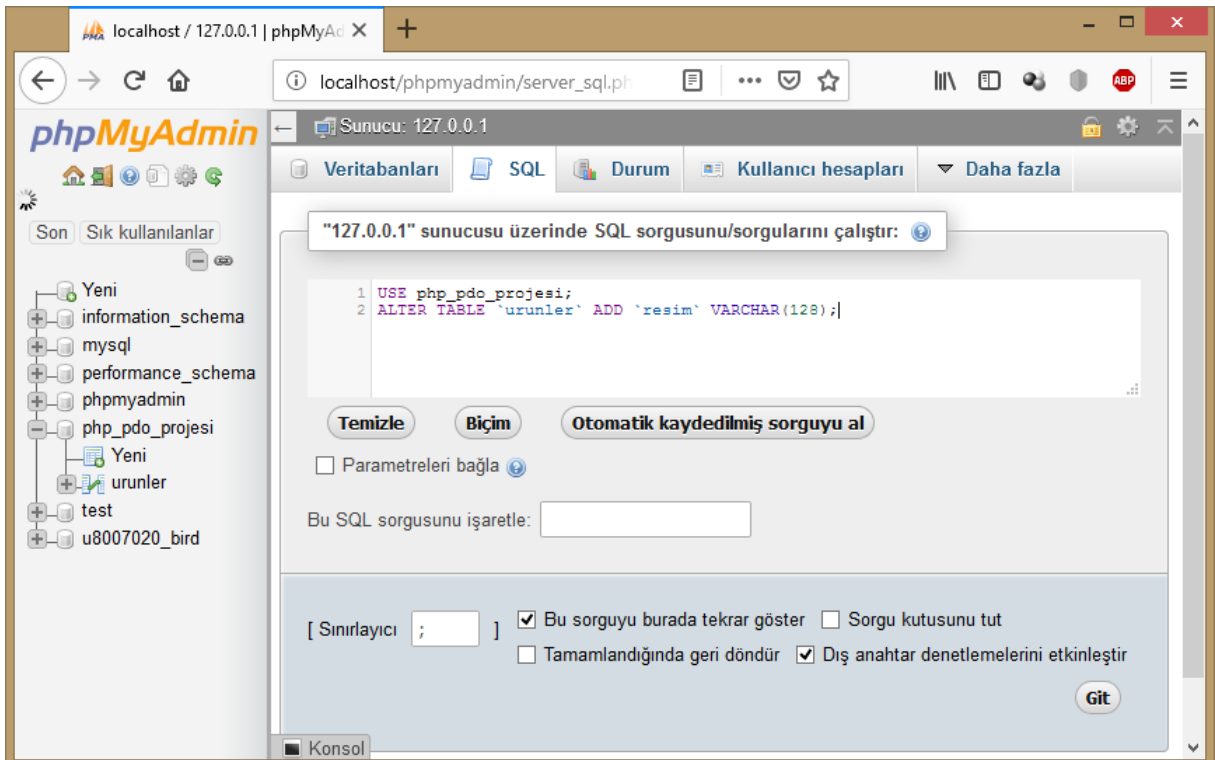
Tarayıcınızın adres satırına **http://localhost/phpmyadmin/** yazarak **phpMyAdmin** yönetim panelini açın.



SQL bağlantısına tıklayarak aşağıdaki sorguyu **Git** butonuyla çalıştırın ve **resim** isimli tablo alanını oluşturun.

```
USE php_pdo_projesi;
```

```
ALTER TABLE `urunler` ADD `resim` VARCHAR(128);
```



ÜRÜN EKLEME SAYFASINI DÜZENLEMEK

Urun klasöründeki **ekle.php** dosyasını açın ve **<form>** etiketini bularak dosya yüklemeye izin vermek için aşağıdaki şekilde değiştirin.

Eski kod:

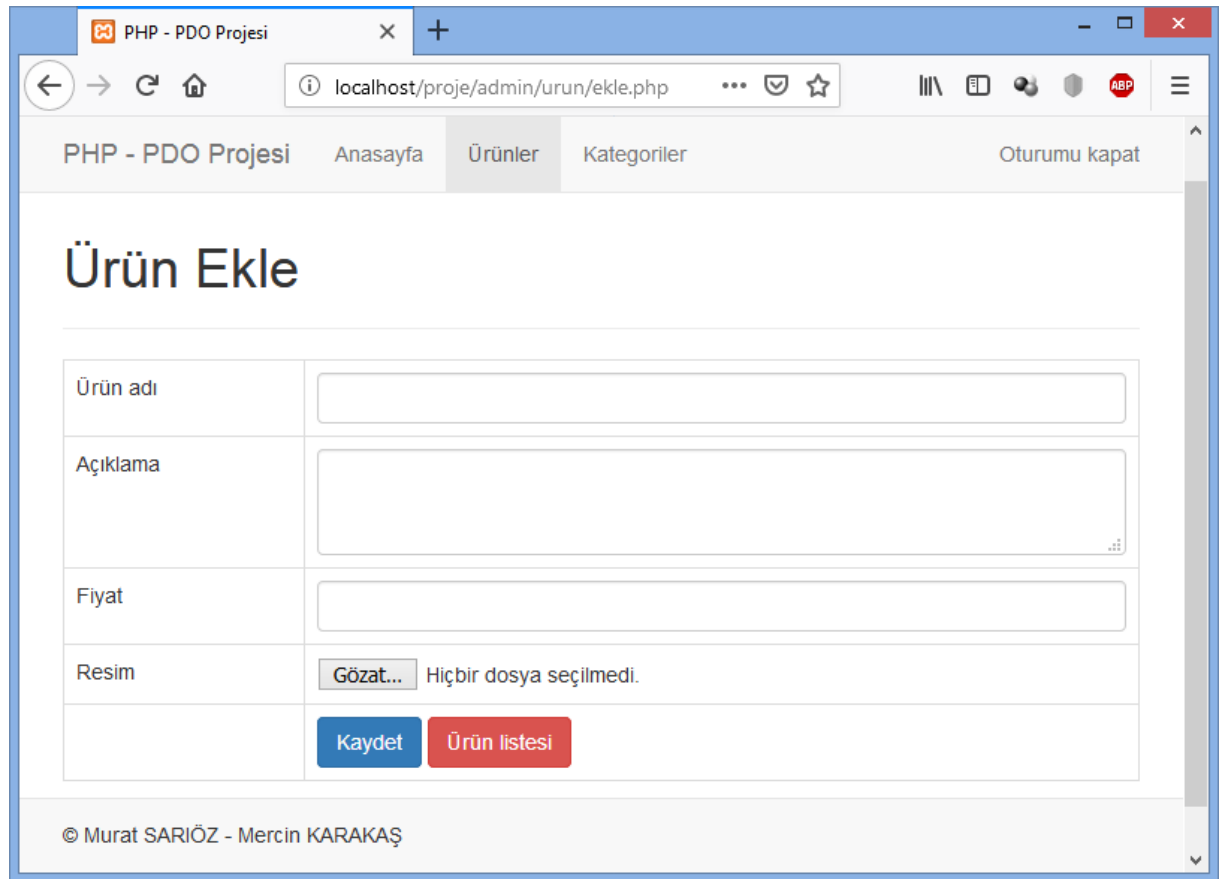
```
<form action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>" method="post">
```

Yeni kod:

```
<form action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>" method="post"
enctype="multipart/form-data">
```

Daha sonra **fiyat** alanının tablo satır sonu etiketinden (**</tr>**) sonra aşağıdaki HTML etiketlerini ekleyin.

```
<tr>
  <td>Resim</td>
  <td><input type="file" name="resim" /></td>
</tr>
```



ÜRÜN EKLEME SORGUSUNU DÜZENLEMEK

Sırada ekleme sorgusunu düzenlemek var. Yeni eklediğimiz **resim** alanında yüklenen dosyanın adını saklayacağız. Ekleme sorgusunun eski ve yeni hali aşağıdaki gibi olacak.

Eski kod:

```
// kayıt ekleme sorgusu
$sorgu = "INSERT INTO urunler SET urunadi=:urunadi, aciklama=:aciklama,
fiyat=:fiyat, giris_tarihi=:giris_tarihi";

// sorguyu hazırla
$stmt = $con->prepare($sorgu);

// post edilen değerler
```

```

$urunadi=htmlspecialchars(strip_tags($_POST['urunadi']));
$aciklama=htmlspecialchars(strip_tags($_POST['aciklama']));
$fiyat=htmlspecialchars(strip_tags($_POST['fiyat']));

// parametreleri bağla
$stmt->bindParam(':urunadi', $urunadi);
$stmt->bindParam(':aciklama', $aciklama);
$stmt->bindParam(':fiyat', $fiyat);

// ürünün veritabanına kaydedildiği tarihi belirt
$giris_tarihi=date('Y-m-d H:i:s');
$stmt->bindParam(':giris_tarihi', $giris_tarihi);

```

Yeni kod:

```

// kayıt ekleme sorgusu
$sorgu = "INSERT INTO urunler SET urunadi=:urunadi, aciklama=:aciklama,
fiyat=:fiyat, giris_tarihi=:giris_tarihi, resim=:resim";

// sorguyu hazırla
$stmt = $con->prepare($sorgu);

// post edilen değerler
$urunadi=htmlspecialchars(strip_tags($_POST['urunadi']));
$aciklama=htmlspecialchars(strip_tags($_POST['aciklama']));
$fiyat=htmlspecialchars(strip_tags($_POST['fiyat']));
// yeni 'resim' alanı
$resim=!empty($_FILES["resim"]["name"]) ? uniqid() . "-" .
basename($_FILES["resim"]["name"]) : "";
$resim=htmlspecialchars(strip_tags($resim));

// parametreleri bağla
$stmt->bindParam(':urunadi', $urunadi);
$stmt->bindParam(':aciklama', $aciklama);
$stmt->bindParam(':fiyat', $fiyat);
$stmt->bindParam(':resim', $resim);

// ürünün veritabanına kaydedildiği tarihi belirt
$giris_tarihi=date('Y-m-d H:i:s');
$stmt->bindParam(':giris_tarihi', $giris_tarihi);

```

DOSYA YÜKLEME VE KAYDETME İŞLEMİ

ekle.php sayfasında aşağıdaki kod satırını bulun

```
echo "<div class='alert alert-success'>Ürün kaydedildi.</div>";
```

ve altına şu satırları ekleyin.

```

// resim boş değilse yükle
if($resim){

    $hedef_klasor = "../content/images/";
    $hedef_dosya = $hedef_klasor . $resim;
    $dosya_turu = pathinfo($hedef_dosya, PATHINFO_EXTENSION);

    // hata mesajı
    $dosya_yukleme_hata_msj="";
}

```

DOSYA TÜRÜ KONTROLÜ

Sadece belirli türdeki resim dosyalarının yüklenmesini sınırlandırmak için aşağıdaki kodları `$dosya_yukleme_hata_msj=""`; satırının altına ekleyin.

```
// sadece belirli dosya türlerine izin ver
$izinverilen_dosya_turleri=array("jpg", "jpeg", "png", "gif");
if(!in_array($dosya_turu, $izinverilen_dosya_turleri)){
    $dosya_yukleme_hata_msj.="<div>Sadece JPG, JPEG, PNG, GIF türündeki dosyalar
yüklenbilir.</div>";
}
```

DOSYANIN İSİM KONTROLÜ

uniqid() fonksiyonunu kullanmamızdan dolayı aynı isimde bir dosyanın daha önce sunucuya yüklenme ihtimali çok düşük olsa da aynı isimde dosya olup olmadığını kontrol etmeliyiz. Dosya türü kontrol kodlarının altına aşağıdaki kodları yazın.

```
// aynı isimde başka bir resim var mı?
if(file_exists($hedef_dosya)){
    $dosya_yukleme_hata_msj.="<div>Aynı isimde başka bir resim dosyası
var.</div>";
}
```

DOSYANIN BOYUT KONTROLÜ

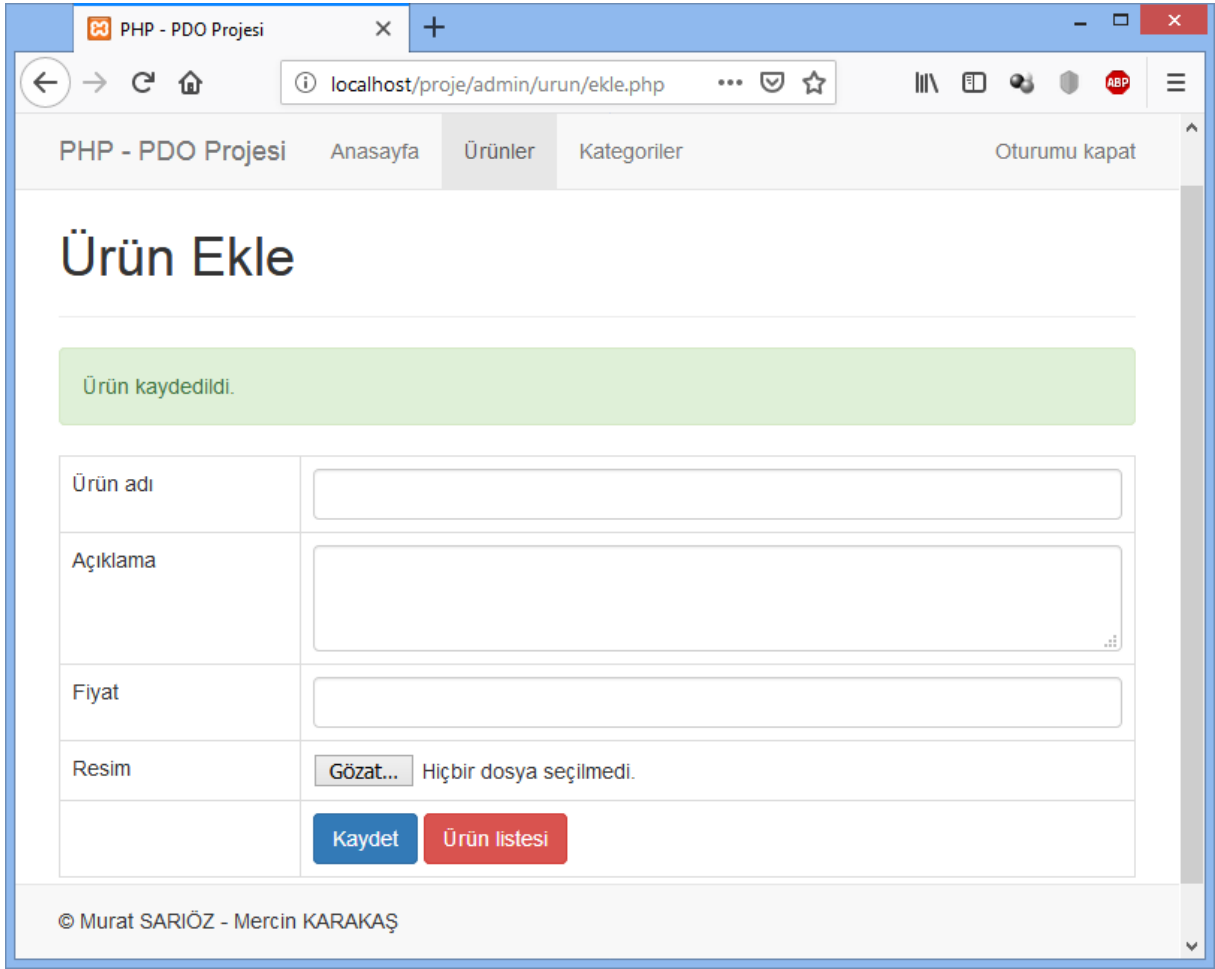
Çok büyük boyutlu dosyaların sunucuya yüklenmesi tavsiye edilmemektedir. Bu yüzden yüklenen resim dosyalarının boyutunu 1 MB ile sınırlandıracğız. Yukarıdaki kodların devamına aşağıdaki kodları yazın.

```
// yüklenen resim dosyasının boyutunun 1 mb sınırını aşmaması için
if($_FILES['resim']['size'] > (1024000)){
    $dosya_yukleme_hata_msj.="<div>Resim dosyasının boyutu 1 MB sınırını
aşamaz.</div>";
}
```

YÜKLENEN DOSYANIN SUNUCUYA KAYDEDİLMESİ

Son olarak yukarıdaki kodların hemen altına aşağıdaki kodları da ekleyerek yüklenen resim dosyasının **content/images** klasörüne kaydedilmesini sağlayın.

```
// eğer $dosya_yukleme_hata_msj boşsa
if(empty($dosya_yukleme_hata_msj)){
    // hata yok, o zaman dosya sunucuya yüklenir
    if(move_uploaded_file($_FILES["resim"]["tmp_name"], $hedef_dosya)){
        // dosya başarıyla yüklendi
    }
    else{
        echo "<div class='alert alert-danger'>";
        echo "<div>Dosya yüklenemedi.</div>";
        echo "<div>Dosyayı yüklemek için kaydı güncelleyin.</div>";
        echo "</div>";
    }
}
// eğer $dosya_yukleme_hata_msj boş değilse
else{
    // hata var, o halde kullanıcıyı bilgilendir
    echo "<div class='alert alert-danger'>";
    echo "<div>{$dosya_yukleme_hata_msj}</div>";
    echo "<div>Dosyayı yüklemek için kaydı güncelleyin.</div>";
    echo "</div>";
}
```



ÜRÜN DETAY SAYFASINDA YÜKLENEBİLİR RESMİ GÖRÜNTÜLEMEK

Yüklenen ürün resmini **detay.php** sayfasında görüntüleyeceğiz.

SEÇME SORGUSUNU DÜZENLEMEK

Urun klasöründeki **detay.php** dosyasını açın ve seçme sorgusunu aşağıdaki gibi düzenleyin.

Eski kod:

```
$sorgu = "SELECT id, urunadi, aciklama, fiyat FROM urunler WHERE id = ? LIMIT 0,1";
```

Yeni kod:

```
$sorgu = "SELECT id, urunadi, aciklama, fiyat, resim FROM urunler WHERE id = ? LIMIT 0,1";
```

\$fiyat değişkeninden sonra **\$resim** değişkenini de tanımlayın.

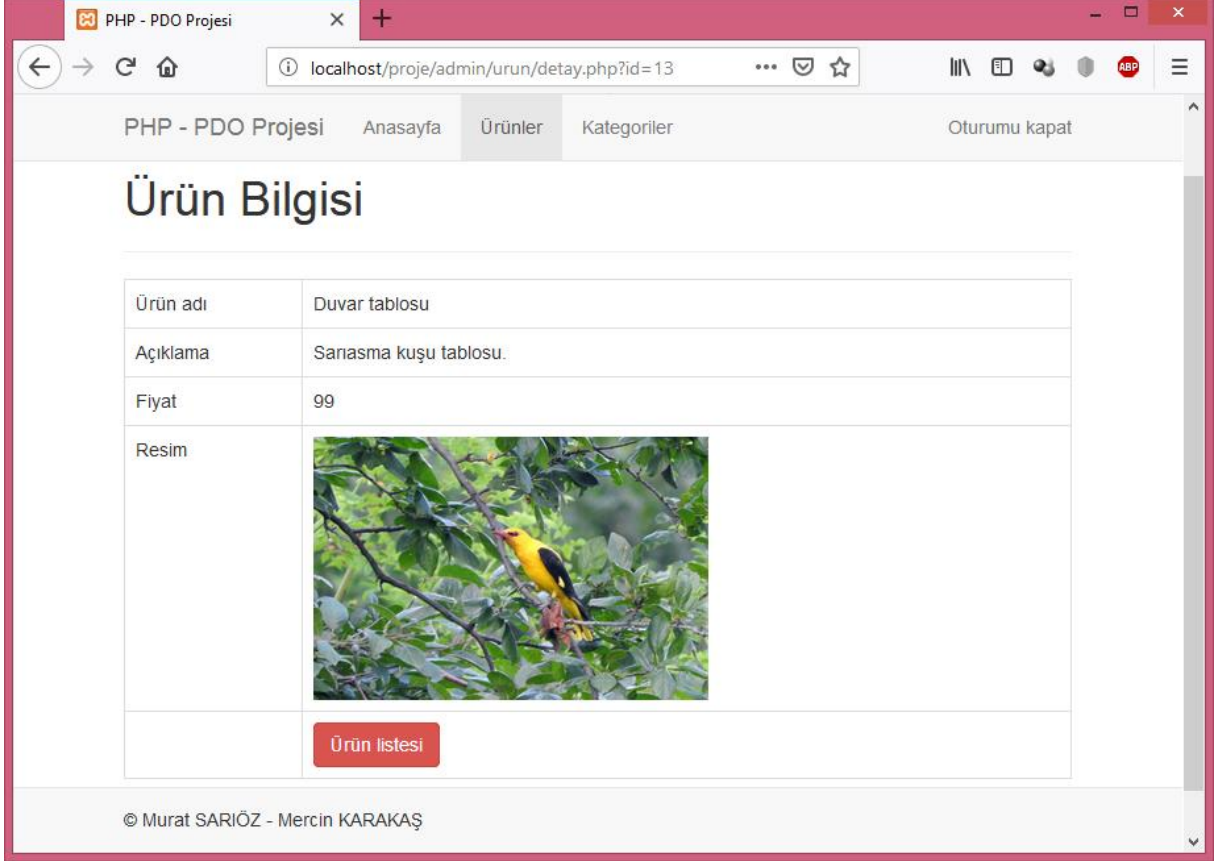
```
$resim = htmlspecialchars($kayit['resim'], ENT_QUOTES);
```

SAYFAYA IMG ETİKETİ EKLEMEK

detay.php sayfasındaki HTML tablosu içine, **fiyat** alanının satır kapatma etiketinden (`</tr>`) sonra, aşağıdaki etiketleri ekleyin. Bu kodlar yüklenmiş bir resim varsa ürün görselini, yoksa bir bilgilendirme iletisi görüntüleyecektir.

```
<tr>
  <td>Resim</td>
  <td><?php echo $resim ? "<img src='../content/images/{" . $resim . "}'
style='width:300px;' />" : "Ürün görseli yok."; ?></td>
</tr>
```

Tarayıcınızın adres satırına **http://localhost/proje/admin/urun/liste.php** yazarak ürün listeleme sayfasını görüntüleyin. Resim yükleyerek kaydettiğiniz herhangi bir kaydın bilgilerini görüntülemek için karşısındaki **Detay** butonuna tıklayın.



ÜRÜN SİLİNDİĞİNDE YÜKLENEN RESMİ SUNUCUDAN SİLMEK

Ürün bilgileri silindiğinde sunucuya yüklenen bir resim varsa bunun da silinmesini sağlamalıyız.

SİLİNECEK RESİM DOSYASININ İSMİNİ ÖĞRENMEK

Urun klasöründeki **sil.php** dosyasını açın ve **\$id=isset(\$_GET['id']) ? \$_GET['id'] : die('HATA: Id bilgisi bulunamadı.');** satırından sonra, silinecek kaydın resim alanındaki dosya adı bilgisini okuyan, aşağıdaki PHP kodlarını ekleyin.

```
// silinecek kayıt bilgilerini oku
// seçme sorgusunu hazırla
$sorgu = "SELECT id, resim FROM urunler WHERE id = ? LIMIT 0,1";
$stmt = $con->prepare( $sorgu );

// id parametresini bağla (? işaretini id değeri ile değiştir)
$stmt->bindParam(1, $id);

// sorguyu çalıştır
$stmt->execute();

$kayit = $stmt->fetch(PDO::FETCH_ASSOC);
// okunan resim bilgilerini bir değişkene kaydet
$resim = $kayit['resim'];
```

RESİM DOSYASINI SUNUCUDAN SİLMEK

Daha sonra **sil.php** sayfasındaki **header('Location: liste.php?islem=silindi');** satırından önce dosyayı sunucudan silecek aşağıdaki kodları yazın.

```
// kayda ait resim varsa sunucudan sil
if(!empty($resim)){
    $silinecek_resim = "../..content/images/".$resim;
    if (file_exists($silinecek_resim)) unlink($silinecek_resim);
}
```


ARAMA İŞLEMİ

Kayıt sayısı arttıkça aranan ürünleri bulmak zorlaşacaktır. Bu işlemi kolaylaştırmak için ürün listeleme sayfasına arama formu ekleyeceğiz.

ARAMA FORMUNU OLUŞTURMAK

Öncelikle **urun** klasöründeki **liste.php** dosyasını açın ve **Yeni Ürün** linkini aşağıdaki gibi düzenleyin.

Eski kod:

```
echo "<a href='ekle.php' class='btn btn-primary m-b-1em'>Yeni Ürün</a>";
```

Yeni kod:

```
echo "<a href='ekle.php' class='btn btn-primary m-b-1em col-xs col-md pull-left'>Yeni Ürün</a>";
```

Ardından bu satırın hemen altına aşağıdaki HTML formunu oluşturun. HTML kodlarından önce PHP etiketinin kapatılıp, sonrasında tekrar açıldığına dikkat edin.

```
?>

<!-- ürün arama formu -->
<form action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>"
method="get">
  <div class="row">
    <div class="col-xs-6 col-md-4 pull-right">
      <div class="input-group">
        <input type="text" class="form-control" placeholder="Ürün ara"
name="aranan" value="<?php echo isset($_GET['aranan']) ? $_GET['aranan'] : ''; ?>"
/>
        <div class="input-group-btn">
          <button class="btn btn-primary" type="submit">
            <span class="glyphicon glyphicon-search"></span>
          </button>
        </div>
      </div>
    </div>
  </div>
</form>

<?php
```

The screenshot shows a web browser window displaying a product list. The URL is localhost/proje/admin/urun/liste.php?sayfa=1. The page has a navigation bar with 'Anasayfa', 'Ürünler', and 'Kategoriler' tabs. The main content area is titled 'Ürün Listesi' and features a search bar with the text 'Ürün ara' and a search icon. Below the search bar is a table with the following data:

ID	Ürün adı	Açıklama	Fiyat	İşlem
13	Duvar tablosu	Sarımsıma kuşu tablosu.	99.99 ₺	Detay Düzeltil Sil
12	Yapışkan etiket	Salyangoz fotoğrafı.	2 ₺	Detay Düzeltil Sil
11	Web kamerası	Görüntülü görüşme için.	20 ₺	Detay Düzeltil Sil

At the bottom of the table, there is a pagination control showing '1 2 3 »' and a 'Git' button. The footer of the page reads '© Murat SARIÖZ - Mercin KARAKAŞ'.

SEÇME SORGUSUNU DÜZENLEMEK

Seçme sorgusuna aranan ürünü kontrol edecek şart ifadesini eklememiz lazım. Sorgunun eski ve yeni hali aşağıdaki gibi olacak.

Eski kod:

```
// sayfada görüntülenecek kayıtları seç
$sorgu = "SELECT id, urunadi, aciklama, fiyat FROM urunler ORDER BY id DESC
LIMIT :ilk_kayit_no, :sayfa_kayit_sayisi";

$stmt = $con->prepare($sorgu);
$stmt->bindParam(":ilk_kayit_no", $ilk_kayit_no, PDO::PARAM_INT);
$stmt->bindParam(":sayfa_kayit_sayisi", $sayfa_kayit_sayisi, PDO::PARAM_INT);
```

Yeni kod:

```
// sayfada görüntülenecek kayıtları seç
$aranan = isset($_GET['aranan']) ? $_GET['aranan'] : "";
$arama_sarti = isset($_GET['aranan']) ? "%".$_GET['aranan']."%" : "%";
$sorgu = "SELECT id, urunadi, aciklama, fiyat FROM urunler WHERE urunadi LIKE
:aranan ORDER BY id DESC LIMIT :ilk_kayit_no, :sayfa_kayit_sayisi";

$stmt = $con->prepare($sorgu);
$stmt->bindParam(":ilk_kayit_no", $ilk_kayit_no, PDO::PARAM_INT);
$stmt->bindParam(":sayfa_kayit_sayisi", $sayfa_kayit_sayisi, PDO::PARAM_INT);
$stmt->bindParam(":aranan", $aranan, PDO::PARAM_STR);
$stmt->bindParam(":arama_sarti", $arama_sarti, PDO::PARAM_STR);
```

ARAMA VE SAYFALANDIRMA İŞLEMLERİNİ BİRLEŞTİRMEK

liste.php sayfasındaki toplam kayıt sayısını hesaplayan sorguyu aşağıdaki gibi düzenleyin.

Eski kod:

```
$sorgu = "SELECT COUNT(*) as kayit_sayisi FROM urunler";
$stmt = $con->prepare($sorgu);
```

Yeni kod:

```
$sorgu = "SELECT COUNT(*) as kayit_sayisi FROM urunler WHERE urunadi LIKE
:aranan";
$stmt = $con->prepare($sorgu);
$stmt->bindParam(":aranan", $aranan, PDO::PARAM_STR);
$stmt->bindParam(":arama_sarti", $arama_sarti, PDO::PARAM_STR);
```

Daha sonra **sayfalama.php** dosyasını açın ve sayfa yönlendirme menü formundaki `</select>` etiketinden sonra aşağıdaki satırı ekleyin.

```
<input type="hidden" name="aranan" value="<?php echo $aranan; ?>">
```

Son olarak da sayfa numaralandırma butonlarının bağlantı etiketlerine (``) aranan parametresini ekleyin.

```
// önceki sayfa butonu
echo "<a href='{ $sayfa_url }?sayfa={ $onceki_sayfa }&aranan={ $aranan }'>";
```

```
// aktif olmayan sayfa
echo " <a href='{ $sayfa_url }?sayfa={ $x }&aranan={ $aranan }'>{ $x }</a> ";
```

```
// sonraki sayfa butonu
echo "<a href='{ $sayfa_url }?sayfa={ $sonraki_sayfa }&aranan={ $aranan }'>";
```

Tarayıcınızın adres satırına **http://localhost/proje/admin/urun/liste.php** yazarak ürün listeleme sayfasını görüntüleyin. **Ürün ara** alanına bir ürün adı yazarak arama butonuna tıklayın.

PHP - PDO Projesi

localhost/proje/admin/urun/liste.php?aranan=kulaklik

PHP - PDO Projesi Anasayfa Ürünler Kategoriler Oturumu kapat

Ürün Listesi

Yeni Ürün

kulaklik

ID	Ürün adı	Açıklama	Fiyat	İşlem
7	Kulaklık	Müzik severler için.	7 ₺	Detay Düzelt Sil

1

1 [Git](#)

© Murat SARIÖZ - Mercin KARAKAŞ

BUTONLARIN GÖRSELLİĞİNİ ZENGİNLEŞTİRMEK

Butonlara görsellik kazandırmak için işlevine uygun simgeler ekleyebilirsiniz. **Bootstrap** kütüphanesi bunun için bize **Glyphicon** simge seti sunar.

ÜRÜN LİSTE SAYFASINDAKİ BUTONLARA SİMGE EKLEMEK

Urun klasöründeki **liste.php** sayfasını açın ve buton bağlantı etiketlerini aşağıdaki gibi yeniden düzenleyin.

Yeni Ürün butonu için:

```
echo "<a href='ekle.php' class='col-xs col-md btn btn-primary m-b-1em pull-left'><span class='glyphicon glyphicon glyphicon-plus'></span> Yeni Ürün</a>";
```

Detay butonu için:

```
echo "<a href='detay.php?id={\$id}' class='btn btn-info m-r-1em'><span class='glyphicon glyphicon glyphicon-eye-open'></span> Detay</a>";
```

Düzeltil butonu için:

```
echo "<a href='duzelt.php?id={\$id}' class='btn btn-primary m-r-1em'><span class='glyphicon glyphicon glyphicon-edit'></span> Düzeltil</a>";
```

Sil butonu için:

```
echo "<a href='#' onclick='silme_onay({\$id});' class='btn btn-danger'><span class='glyphicon glyphicon glyphicon-remove-circle'></span> Sil</a>";
```

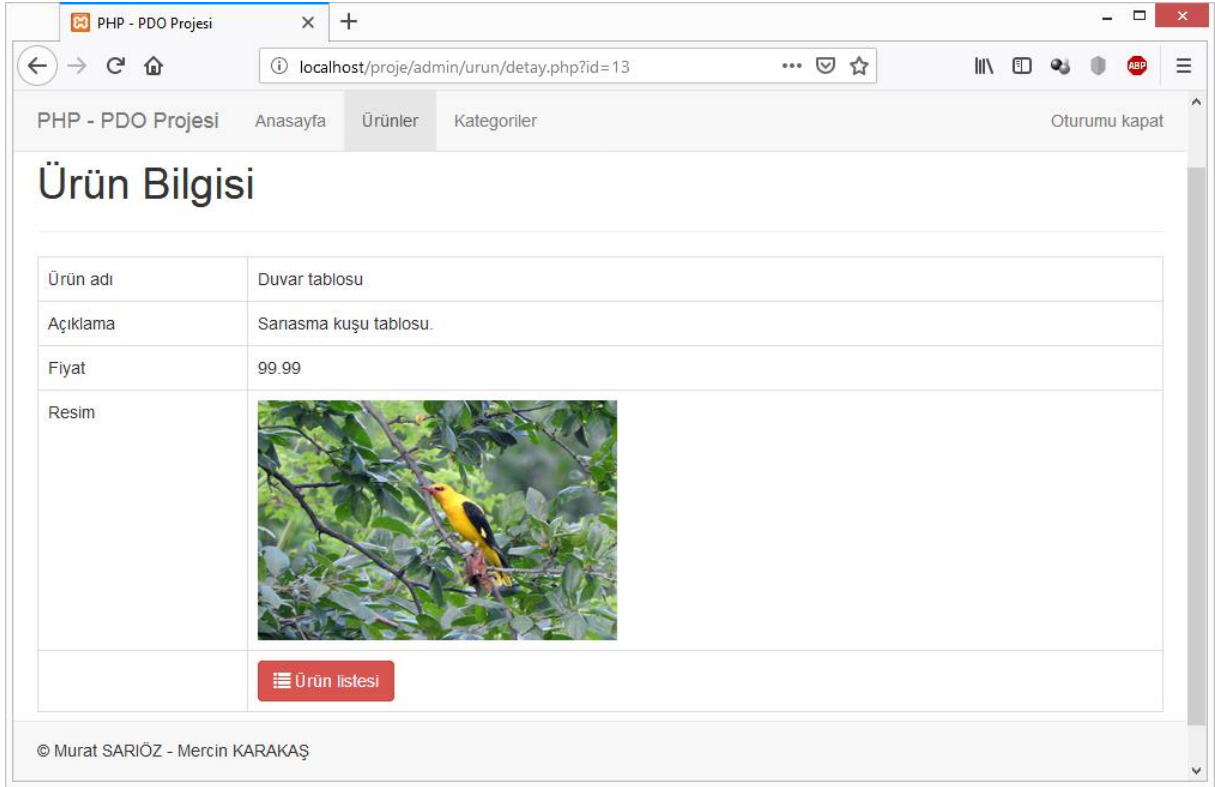
ID	Ürün adı	Açıklama	Fiyat	İşlem
13	Duvar tablosu	Sarıasma kuşu tablosu.	99.99 ₺	Detay Düzeltil Sil
12	Yapışkan etiket	Salyangoz fotoğrafı.	2 ₺	Detay Düzeltil Sil
11	Web kamerası	Görüntülü görüşme için.	20 ₺	Detay Düzeltil Sil

ÜRÜN DETAY SAYFASINDAKİ BUTONLARA SİMGE EKLEMEK

Urun klasöründeki **detay.php** sayfasını açın ve buton bağlantı etiketlerini aşağıdaki gibi yeniden düzenleyin.

Ürün Listesi butonu için:

```
<a href='liste.php' class='btn btn-danger'><span class='glyphicon glyphicon-glyphicon-list'></span> Ürün listesi</a>
```



ÜRÜN EKLEME SAYFASINDAKİ BUTONLARA SİMGE EKLEMEK

Urun klasöründeki **ekle.php** sayfasını açın ve buton bağlantı etiketlerini aşağıdaki gibi yeniden düzenleyin.

Kaydet butonu için:

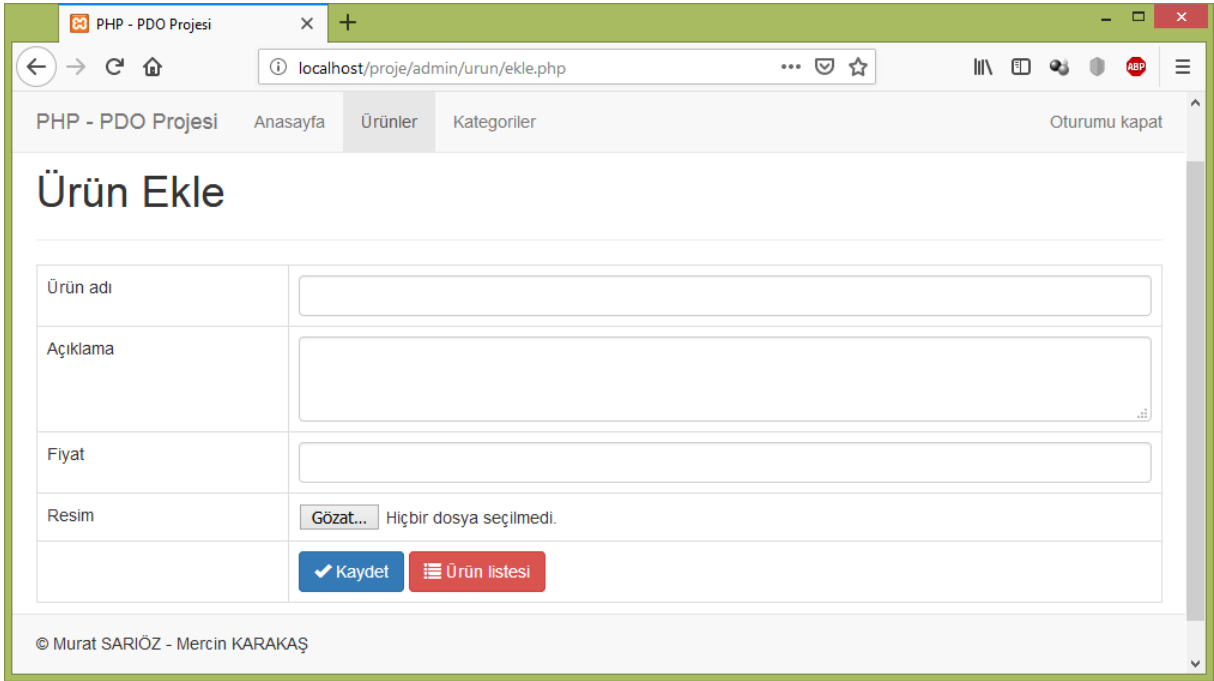
```
<input type='submit' value='Kaydet' class='btn btn-primary' />
```

Yukarıdaki kod satırını aşağıdaki satırla değiştirin.

```
<button type="submit" class='btn btn-primary'><span class="glyphicon glyphicon-ok"></span> Kaydet</button>
```

Ürün Listesi butonu için:

```
<a href='liste.php' class='btn btn-danger'><span class='glyphicon glyphicon-glyphicon-list'></span> Ürün listesi</a>
```



ÜRÜN GÜNCELLEME SAYFASINDAKİ BUTONLARA SİMGE EKLEMEK

Urun klasöründeki **duzelt.php** sayfasını açın ve buton bağlantı etiketlerini aşağıdaki gibi yeniden düzenleyin.

Kaydet butonu için:

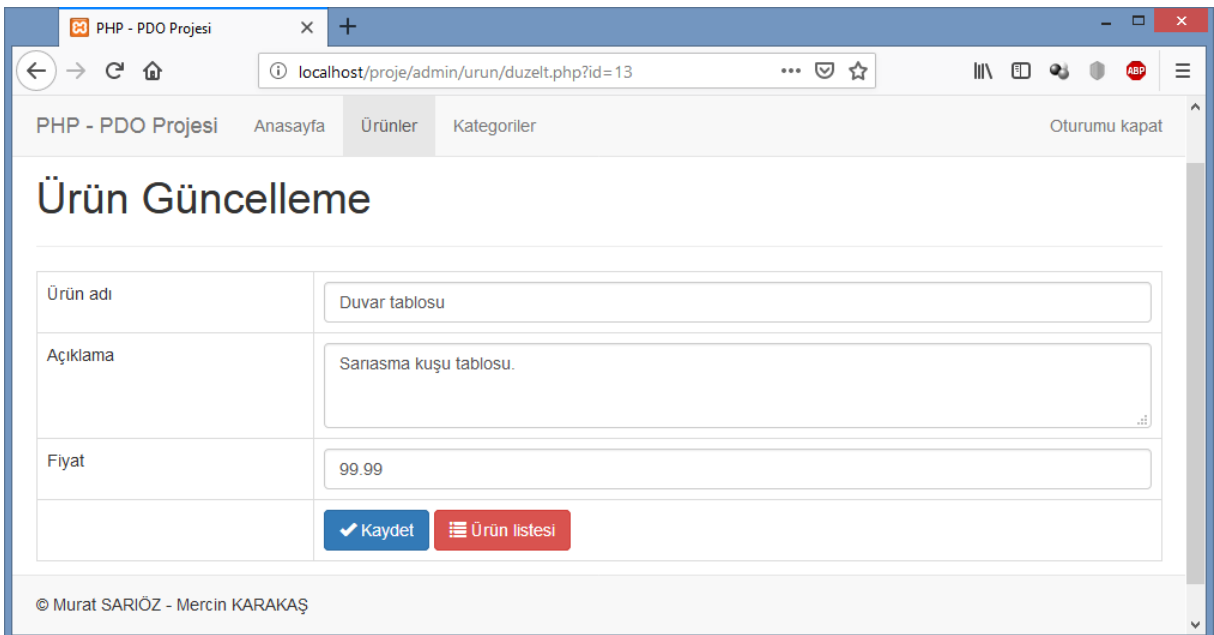
```
<input type='submit' value='Kaydet' class='btn btn-primary' />
```

Yukarıdaki kod satırını aşağıdaki satırla değiştirin.

```
<button type="submit" class='btn btn-primary'><span class="glyphicon glyphicon-ok"></span> Kaydet</button>
```

Ürün Listesi butonu için:

```
<a href='liste.php' class='btn btn-danger'><span class='glyphicon glyphicon-glyphicon-list'></span> Ürün listesi</a>
```



ÇOKLU ÜRÜN SİLME İŞLEVI

ÇOKLU ÜRÜN SEÇME KUTULARINI (CHECKBOX) OLUŞTURMAK

Silme işlevini geliştirelim ve PHP ile Ajax kullanımına bir örnek oluşturalım. Öncelikle birden fazla kaydı seçmek ve aynı anda silmek için kayıtların önüne birer işaret kutusu (checkbox) eklemelisiniz. **Admin/urun/liste.php** sayfa dosyasını açın ve ürünleri listelediğimiz HTML tablo kodlarına aşağıdaki gibi eklemeler yapalım.

Tablo başlık kodlarına aşağıdaki satırı ekleyin.

```
//tablo başlıkları
echo "<tr>";
    echo "<th></th>";
    echo "<th>ID</th>";
    echo "<th>Ürün adı</th>";
    echo "<th>Açıklama</th>";
    echo "<th>Fiyat</th>";
    echo "<th>Kategori</th>";
    echo "<th>İşlem</th>";
echo "</tr>";
```

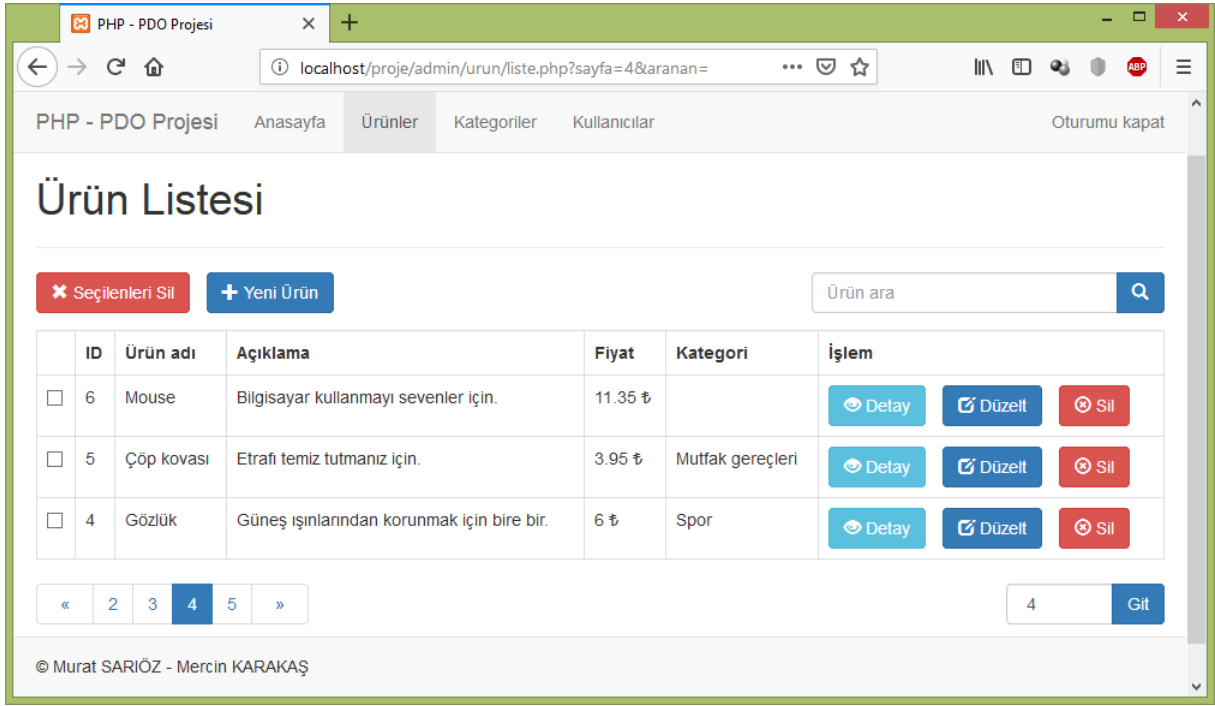
Daha sonra kayıtların listelendiği **while {}** döngüsü kodlarına aşağıdaki satırı ekleyin.

```
// her kayıt için yeni bir tablo satırı oluştur
echo "<tr>";
    echo "<td><input type='checkbox' name='sil_id[]' value='{ $id }'/></td>";
    echo "<td>{ $id }</td>";
    echo "<td>{ $urunadi }</td>";
    echo "<td>{ $aciklama }</td>";
    echo "<td>{ $fiyat } &#8378;</td>";
    echo "<td>{ $kategoriadi }</td>";
    echo "<td>";
    // kayıt detay sayfa bağlantısı
    echo "<a href='detay.php?id={ $id }' class='btn btn-info m-r-1em'><span
class='glyphicon glyphicon glyphicon-eye-open'></span> Detay</a>";
    // kayıt güncelleme sayfa bağlantısı
    echo "<a href='duzelt.php?id={ $id }' class='btn btn-primary m-r-1em'><span
class='glyphicon glyphicon glyphicon-edit'></span> Düzelt</a>";
    // bu bağlantıyı dokümanın ileriki bölümlerinde işleyeceğiz
    echo "<a href='#' onclick='silme_onay({ $id });' class='btn btn-danger'><span
class='glyphicon glyphicon glyphicon-remove-circle'></span> Sil</a>";
    echo "</td>";
echo "</tr>";
```

Ardından **// kayıt ekleme sayfasının linki** açıklama satırının üstüne aşağıdaki kod satırını ekleyerek çoklu silme butonunu oluşturun.

```
// çoklu kayıt silme butonu
echo "<a href='#' id='btn_sil' class='col-xs col-md m-r-1em m-b-1em pull-left btn
btn-danger'><span class='glyphicon glyphicon glyphicon-remove'> Seçilenleri
Sil</a>";
```

Tarayıcınızın adres satırına **http://localhost/proje/admin/urun/liste.php** yazarak ürün listeleme sayfasını görüntüleyin.



AJAX KODLARINI OLUŞTURMAK

Seçilenleri Sil butonu ile tetiklenecek aşağıdaki Ajax kodlarını **admin/urun/liste.php** kodlarının sonuna ekleyin. Uyarı mesajları için daha iyi bir görsel arayüze sahip **SweetAlert** eklentisi kullanılacak.

```
<!-- SweetAlert destekli çoklu silme -->
<script src="https://unpkg.com/sweetalert/dist/sweetalert.min.js"></script>
<script>
$(document).ready(function(){
    $('#btn_sil').click(function(){
        var id = [];

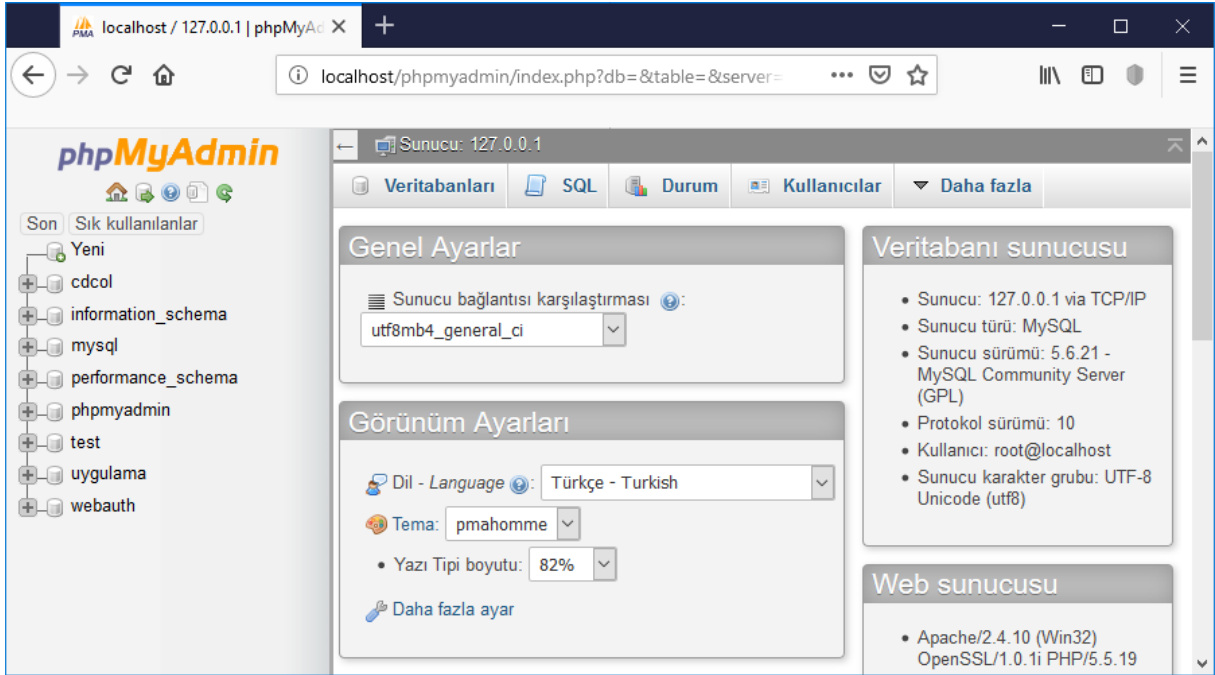
        $(':checkbox:checked').each(function(i){
            id[i] = $(this).val();
        });

        if(id.length === 0){ //dizi boşsa bilgi ver
            swal("Silme için seçilmiş ürün yok!",{
                icon: "error",
                buttons: false,
                timer: 3000,
            });
        }
        else {
            swal({ // onay al
                title: "Emin misiniz?",
                text: "Silme işlemi geri alınamaz!",
                icon: "warning",
                buttons: ["Hayır", "Evet"],
                dangerMode: true,
                closeModal: false,
            })
            .then(function(yes){
                if (yes)
                    $.ajax({
                        cache: false,
                        type: 'POST',
                        url: 'coklusil.php',
                        data: {id:id},
                        success: function(sonuc){
                            swal("Seçili ürünler silindi!", {
```

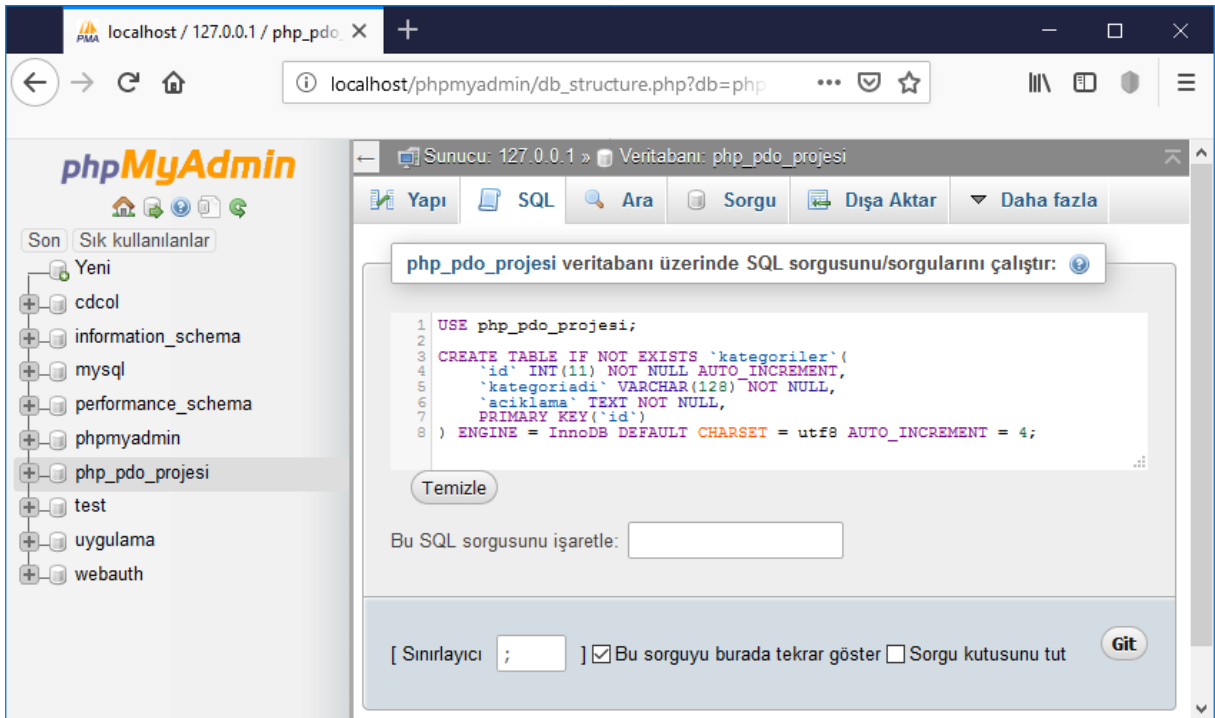

KATEGORİ SAYFALARINI OLUŞTURMAK

Ürünleri belirli kategori başlıklarıyla gruplayacağız. Bu daha sonra kullanıcıların aradıkları ürünü bulmalarını kolaylaştıracaktır.

Veritabanında **kategoriler** tablosunu oluşturmak için tarayıcınızın adres satırına **http://localhost/phpmyadmin/** yazarak **phpMyAdmin** veritabanı yönetim panelini açın.



Şimdi **SQL** bağlantısına tıklayın ve aşağıdaki SQL sorgusunu yazarak **Git** butonuyla çalıştırın.



```
USE php PDO Projesi;

CREATE TABLE IF NOT EXISTS `kategoriler` (
  `id` INT(11) NOT NULL AUTO_INCREMENT,
  `kategoriadi` VARCHAR(128) NOT NULL,
  `aciklama` TEXT NOT NULL,
  PRIMARY KEY(`id`)
) ENGINE = InnoDB DEFAULT CHARSET = utf8 AUTO_INCREMENT = 4;
```

Ardından **SQL** bağlantısına tekrar tıklayın ve aşağıdaki SQL komutunu çalıştırarak oluşturduğunuz **kategoriler** tablosuna örnek veriler kaydedin.

```
USE php_pdo_projesi;

INSERT INTO `kategoriler` ( `id`, `kategoriadi`, `aciklama` ) VALUES
( 1, 'Elektronik', 'Bilgisayar, televizyon, cep telefonu'),
( 2, 'Spor', 'Spor giyim, egzersiz aletleri'),
( 3, 'Mutfak gereçleri', 'Bıçak, kaşık, tencere, tava');
```

KATEGORİ LİSTELEME SAYFASINI OLUŞTURMAK

İlk önce **admin** klasörü içinde **kategori** isimli bir klasör oluşturun. Bu klasör içinde **liste.php** isimli bir dosya oluşturun ve içine aşağıdaki satırları ekleyin.

```
<?php include "../header.php"; ?>

<div class="container">

    <div class="page-header">
        <h1>Kategori Listesi</h1>
    </div>

    <!-- Kayıtları listeleyecek PHP kodları bu alana eklenecek -->

</div> <!-- /container -->

<?php include "../footer.php"; ?>

<!-- Kayıt silme onay kodları bu alana eklenecek -->
```

KAYIT LİSTELEMEK İÇİN GEREKLİ SQL SORGUSUNU OLUŞTURMAK

Ardından **<!-- Kayıtları listeleyecek PHP kodları bu alana eklenecek -->** açıklama satırının yerine aşağıdaki PHP kodlarını yazın.

```
<?php
// veritabanı bağlantı dosyasını çağır
include '../../config/vtabani.php';

// silme mesajı burada yer alacak

// bütün kayıtları seç
$sorgu = "SELECT id, urunadi, aciklama FROM kategoriler ORDER BY id DESC";
$stmt = $con->prepare($sorgu);
$stmt->execute();

// geriye dönen kayıt sayısı
$sayi = $stmt->rowCount();

// kayıt ekleme sayfasının linki
echo "<a href='ekle.php' class='btn btn-primary m-b-1em'>Yeni Kategori</a>";

//kayıt varsa listele
if($sayi>0){

    // kayıtlar burada listelenecek

}
// kayıt yoksa mesajla bildir
else{
    echo "<div class='alert alert-danger'>Listelenecek kayıt bulunamadı.</div>";
}
```

```
}  
?>
```

KAYIT LİSTELEMELİK İÇİN GEREKLİ HTML TABLOSUNU OLUŞTURMAK

Sıra geldi veritabanından okunan kayıtların listelenmesine. Bunun için HTML tablo yapısını kullanacağız.

// kayıtlar burada listelenecek açıklama satırının yerine aşağıdaki PHP kodlarını yazın.

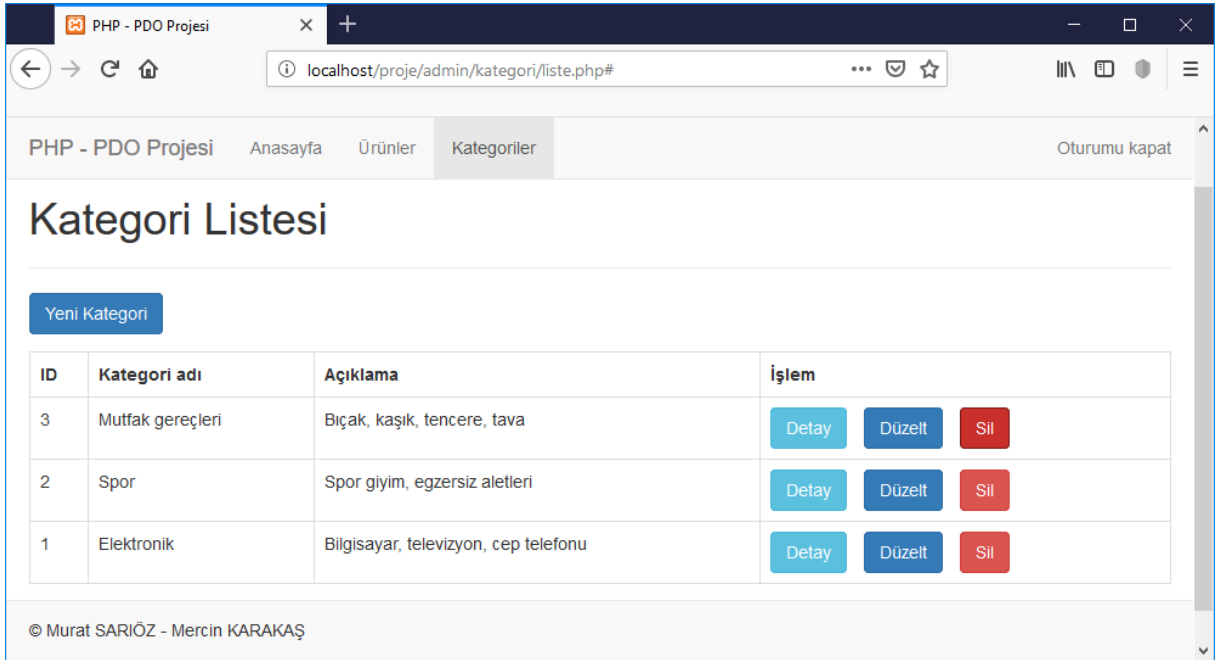
```
echo "<table class='table table-hover table-responsive table-bordered'>";  
//tablo başlangıcı  
  
//tablo başlıkları  
echo "<tr>";  
    echo "<th>ID</th>";  
    echo "<th>Kategori adı</th>";  
    echo "<th>Açıklama</th>";  
    echo "<th>İşlem</th>";  
echo "</tr>";  
  
// tablo içeriği burada yer alacak  
  
echo "</table>"; // tablo sonu
```

VERİTABANINDAKİ KAYITLARI LİSTELEMELİK İÇİN GEREKLİ PHP KODLARI

Tablonun ana yapısını oluşturduktan sonra, içine verileri dolduracak PHP kodlarını da // tablo içeriği burada yer alacak açıklama satırının yerine yazın.

```
// tablo verilerinin okunması  
while ($kayit = $stmt->fetch(PDO::FETCH_ASSOC)){  
    // tablo alanlarını değişkene dönüştürür  
    // $kayit['kategoriadi'] => $kategoriadi  
    extract($kayit);  
  
    // her kayıt için yeni bir tablo satırı oluştur  
    echo "<tr>";  
        echo "<td>{$id}</td>";  
        echo "<td>{$kategoriadi}</td>";  
        echo "<td>{$aciklama}</td>";  
        echo "<td>";  
            // kayıt detay sayfa bağlantısı  
            echo "<a href='detay.php?id={$id}' class='btn btn-info m-r-1em'>Detay</a>";  
            // kayıt güncelleme sayfa bağlantısı  
            echo "<a href='duzelt.php?id={$id}' class='btn btn-primary m-r-1em'>Düzeltil</a>";  
            // kayıt silme butonu  
            echo "<a href='#' onclick='silme_onay({$id});' class='btn btn-danger'>Sil</a>";  
            echo "</td>";  
        echo "</tr>";  
    }  
}
```

Tarayıcınızın adres satırına <http://localhost/proje/admin/kategori/liste.php> yazarak kategori listeleme sayfasını görüntüleyin.



KATEGORİ SİLME SAYFASINI OLUŞTURMAK

Bir kaydı silmek için işlem butonlarından **Sil** butonuna tıklayınca önce kullanıcının onayını alacağız. Böylelikle yanlışlıkla kayıtların silinmesini önleyebiliriz. Silme işlemi onayını **liste.php** sayfasında, silme işlemi ise kod karışıklığını azaltmak için ayrı bir sayfada (**sil.php**) yapacağız.

SİLME İŞLEMİNDEN ÖNCE KULLANICI ONAYINI ALMAK

Öncelikle **liste.php** dosyasını açın ve **<!-- Kayıt silme onay kodları bu alana eklenecek -->** açıklama satırının yerine aşağıdaki JavaScript kodlarını yazın.

```
<script type='text/javascript'>
// kayıt silme işlemi onayla
function silme_onay( id ){

    var cevap = confirm('Kaydı silmek istiyor musunuz?');
    if (cevap){
        // kullanıcı evet derse,
        // id bilgisini sil.php sayfasına yönlendirir
        window.location = 'sil.php?id=' + id;
    }
}
</script>
```

VERİTABANINDAN KAYIT SİLMEK İÇİN GEREKLİ PHP KODLARI

Şimdi **admin/kategori** klasörü içinde **sil.php** isimli bir dosya oluşturun ve içine aşağıdaki PHP kodlarını yazın.

```
<?php
// veritabanı ayar dosyasını dahil et
include '../config/vtabani.php';

try {
    // kaydın id bilgisini al
    $id=isset($_GET['id']) ? $_GET['id'] : die('HATA: Id bilgisi bulunamadı.');
```

```

// sorguyu çalıştır
if($stmt->execute()){
    // kayıt listeleme sayfasına yönlendir
    // ve kullanıcıya kaydın silindiğini
    header('Location: liste.php?islem=silindi');
} // veya silinemediğini bildir
else{
    header('Location: liste.php?islem=silinemedi');
}
}
// hata varsa göster
catch(PDOException $exception){
    die('HATA: ' . $exception->getMessage());
}
?>

```

SİLME İŞLEMİNDEN SONRA KULLANICIYA BİLGİ VERMEK

Silme işleminden sonra kullanıcıyı bilgilendirmek için ürün listeleme sayfasında düzenleme yapacağız. **liste.php** dosyasını açın ve **// silme mesajı burada yer alacak** açıklama satırının yerine aşağıdaki kodları ekleyin.

```

$islem = isset($_GET['islem']) ? $_GET['islem'] : "";

// eğer silme (sil.php) sayfasından yönlendirme yapıldıysa
if($islem=='silindi'){
    echo "<div class='alert alert-success'>Kayıt silindi.</div>";
}
else if($islem=='silinemedi'){
    echo "<div class='alert alert-danger'>Kayıt silinemedi.</div>";
}

```

Tarayıcınızın adres satırına **http://localhost/proje/admin/kategori/liste.php** yazarak kategori listeleme sayfasını görüntüleyin. Herhangi bir kaydı silmek için karşısındaki **Sil** butonuna tıklayın ve silme işlemi **Tamam** butonuyla onaylayın.

The screenshot shows a web browser window with the address bar displaying `localhost/proje/admin/kategori/liste.php#`. The page title is "PHP - PDO Projesi" and the navigation menu includes "Anasayfa", "Ürünler", "Kategoriler", and "Oturumu kapat". The main content area is titled "Kategori Listesi" and features a "Yeni Kategori" button. Below the button is a table with the following data:

ID	Kategori adı	Açıklama	Detay	Düzeltil	Sil
3	Mutfak gereçleri	Bıçak, kaşık, tencere, tava	Detay	Düzeltil	Sil
2	Spor	Spor giyim, egzersiz aletleri	Detay	Düzeltil	Sil
1	Elektronik	Bilgisayar, televizyon, cep telefonu	Detay	Düzeltil	Sil

A confirmation dialog box is displayed in the center of the screen with the text "Kaydı silmek istiyor musunuz?" and two buttons: "Tamam" and "Vazgeç".

© Murat SARIÖZ - Mercin KARAKAŞ
localhost/proje/admin/kategori/liste.php#

KATEGORİ DETAY SAYFASINI OLUŞTURMAK

Kategori listeleme sayfasında **Detay** işlem butonu tıklanınca kayıtların detay bilgisini görüntülemek için kullanıcıyı yönlendireceğimiz sayfayı **detay.php** ismiyle **admin/kategori** klasöründe oluşturun ve içine aşağıdaki kodları yazın.

```
<?php include "../header.php"; ?>

<div class="container">

    <div class="page-header">
        <h1>Kategori Bilgisi</h1>
    </div>

    <!-- kategori bilgilerini getiren PHP kodu burada yer alacak -->

    <!-- kategori bilgilerini görüntüleyen HTML tablosu burada yer alacak -->

</div> <!-- container -->

<?php include "../footer.php"; ?>
```

KATEGORİ BİLGİLERİNİ VERİTABANINDAN OKUMAK

Aşağıdaki PHP kodları MySQL veritabanındaki **kategoriiler** tablosunda kayıtlı ilgili (**Id** parametre değerine karşılık gelen) kategorinin bilgilerini getirecektir. Bu kodları **<!-- kategori bilgilerini getiren PHP kodu burada yer alacak -->** açıklama satırının yerine yazın.

```
<?php
// gelen Id parametresini al
// isset() bir değer olup olmadığını kontrol eden PHP fonksiyonudur
$id=isset($_GET['id']) ? $_GET['id'] : die('HATA: Kayıt bulunamadı.');
```

```
// veritabanı bağlantı dosyasını çağır
include '../config/vtabani.php';

// aktif kayıt bilgilerini oku
try {
    // seçme sorgusunu hazırla
    $sorgu = "SELECT id, kategoriadi, aciklama FROM kategoriiler WHERE id = ?
LIMIT 0,1";
    $stmt = $con->prepare( $sorgu );

    // Id parametresini bağla
    $stmt->bindParam(1, $id);

    // sorguyu çalıştır
    $stmt->execute();

    // gelen kaydı bir değişkende sakla
    $kayit = $stmt->fetch(PDO::FETCH_ASSOC);

    // tabloya yazılacak bilgileri değişkenlere doldur
    $kategoriadi = $kayit['kategoriadi'];
    $aciklama = $kayit['aciklama'];
}

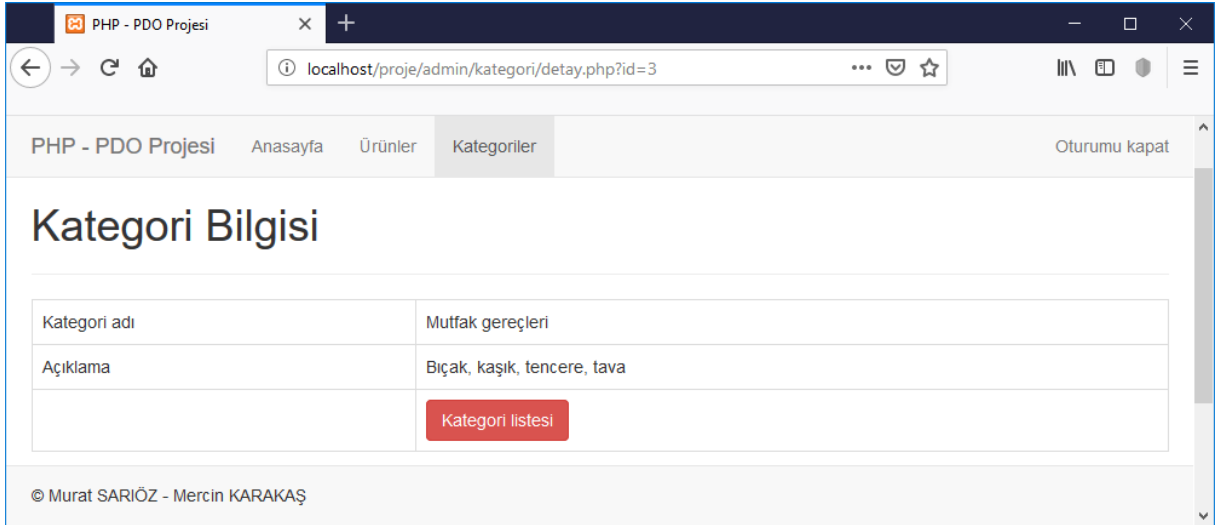
// hatayı göster
catch(PDOException $exception){
    die('HATA: ' . $exception->getMessage());
}
?>
```

KATEGORİ BİLGİLERİNİ GÖRÜNTÜLEMEK İÇİN GEREKLİ HTML TABLOSUNU OLUŞTURMAK

Aşağıdaki kodlar veritabanındaki kategori bilgilerinin görüntülenmesi için gerekli HTML tablosunu oluşturacaktır. Bu kodları `<!-- kategori bilgilerinin görüntüleyen HTML tablosu burada yer alacak -->` açıklama satırının yerine yazın.

```
<!-- kayıt bilgilerinin görüntüleyen HTML tablosu -->
<table class='table table-hover table-responsive table-bordered'>
  <tr>
    <td>Kategori adı</td>
    <td><?php echo htmlspecialchars($kategoriadi, ENT_QUOTES); ?></td>
  </tr>
  <tr>
    <td>Açıklama</td>
    <td><?php echo htmlspecialchars($aciklama, ENT_QUOTES); ?></td>
  </tr>
  <tr>
    <td></td>
    <td>
      <a href='liste.php' class='btn btn-danger'>Kategori listesi</a>
    </td>
  </tr>
</table>
```

Tarayıcınızın adres satırına <http://localhost/proje/admin/kategori/liste.php> yazarak kategori listeleme sayfasını görüntüleyin. Herhangi bir kaydın bilgilerinin görüntülemek için karşısındaki **Detay** butonuna tıklayın.



KATEGORİ EKLEME SAYFASINI OLUŞTURMAK

Yeni kayıt eklemek için kullanılacak **ekle.php** sayfasını **admin/kategori** klasöründe oluşturun ve aşağıdaki HTML kodlarını içine yazın.

```
<?php include "../header.php"; ?>
<div class="container">
  <div class="page-header">
    <h1>Kategori Ekle</h1>
  </div>
  <!-- PHP kayıt ekleme kodları burada yer alacak -->
  <!-- Kategori eklemek için kullanılacak html formu burada yer alacak -->
```



```
</div> <!-- container -->
<?php include "../footer.php"; ?>
```

YENİ KAYIT GİRİŞİ İÇİN GEREKLİ HTML FORMUNU OLUŞTURMAK

Aşağıdaki kodlar veritabanındaki alanlara karşılık gelen bilgilerin girilmesi için gerekli HTML formunu oluşturacaktır. Bu kodları <!-- Kategori eklemek için kullanılacak html formu burada yer alacak --> açıklama satırının yerine yazın.

```
<!-- Kategori bilgilerini girmek için kullanılacak html formu -->
<form action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>" method="post">
  <table class='table table-hover table-responsive table-bordered'>
    <tr>
      <td>Kategori adı</td>
      <td><input type='text' name='kategoriadi' class='form-control' /></td>
    </tr>
    <tr>
      <td>Açıklama</td>
      <td><textarea name='aciklama' class='form-control'></textarea></td>
    </tr>
    <tr>
      <td></td>
      <td>
        <input type='submit' value='Kaydet' class='btn btn-primary' />
        <a href='liste.php' class='btn btn-danger'>Kategori listesi</a>
      </td>
    </tr>
  </table>
</form>
```

VERİTABANINA YENİ KAYIT EKLEMEK İÇİN GEREKLİ PHP KODLARI

Kullanıcı **ekle.php** sayfasında bilgileri girip **Kaydet** butonuna tıklayınca aşağıdaki PHP kodları MySQL veritabanındaki **kategoriiler** tablosuna bu bilgileri kaydedecektir. Bu kodları <!-- PHP kayıt ekleme kodları burada yer alacak --> açıklama satırının yerine yazın.

```
<?php
if($_POST){

  // veritabanı yapılandırma dosyasını dahil et
  include '../config/vtabani.php';

  try{
    // kayıt ekleme sorgusu
    $sorgu = "INSERT INTO kategoriiler SET kategoriadi=:kategoriadi,
aciklama=:aciklama";

    // sorguyu hazırla
    $stmt = $con->prepare($sorgu);

    // post edilen değerler
    $kategoriadi=htmlspecialchars(strip_tags($_POST['kategoriadi']));
    $aciklama=htmlspecialchars(strip_tags($_POST['aciklama']));

    // parametreleri bağla
    $stmt->bindParam(':kategoriadi', $kategoriadi);
    $stmt->bindParam(':aciklama', $aciklama);

    // sorguyu çalıştır
    if($stmt->execute()){
      echo "<div class='alert alert-success'>Kategori kaydedildi.</div>";
    }else{
```

```
        echo "<div class='alert alert-danger'>Kategori kaydedilemedi.</div>";
    }
}
// hatayı göster
catch(PDOException $exception){
    die('ERROR: ' . $exception->getMessage());
}
}
?>
```

Tarayıcınızın adres satırına <http://localhost/proje/admin/kategori/ekle.php> yazarak kategori ekleme sayfasını görüntüleyin.

PHP - PDO Projesi Anasayfa Ürünler Kategoriler Oturumu kapat

Kategori Ekle

Kategori adı	<input type="text" value="Oyuncak"/>
Açıklama	<input type="text" value="Bebek, çocuk ve yetişkin oyuncakları"/>
	<input type="button" value="Kaydet"/> <input type="button" value="Kategori listesi"/>

© Murat SARIÖZ - Mercin KARAKAŞ

Formu doldurun ve **Kaydet** butonuyla yeni bir kayıt ekleyin.

PHP - PDO Projesi Anasayfa Ürünler Kategoriler Oturumu kapat

Kategori Ekle

Kategori kaydedildi.

Kategori adı	<input type="text"/>
Açıklama	<input type="text"/>
	<input type="button" value="Kaydet"/> <input type="button" value="Kategori listesi"/>

© Murat SARIÖZ - Mercin KARAKAŞ

KATEGORİ GÜNCELLEME (DÜZELTME) SAYFASINI OLUŞTURMAK

Kategori listeleme sayfasında **Düzel** işlem butonu tıklanınca kayıtların bilgilerini bir form ile görüntülemek ve güncellemek için kullanıcıyı yönlendireceğimiz sayfayı **duzelt.php** ismiyle **admin/kategori** klasöründe oluşturun ve içine aşağıdaki kodları yazın.

```

<?php include "../header.php"; ?>

<div class="container">

    <div class="page-header">
        <h1>Kategori Güncelleme</h1>
    </div>

    <!-- kategori bilgilerini getiren PHP kodu burada yer alacak -->

    <!-- kategori bilgilerini düzeltme HTML formu burada yer alacak -->

</div> <!-- container -->

<?php include "../footer.php"; ?>

```

GÜNCELLENECEK KATEGORİ BİLGİLERİNİ VERİTABANINDAN OKUMAK

Öncelikle kategori listeleme sayfasından gelen **Id** parametresini **\$_GET['id']** fonksiyonu ile alıp **\$id** değişkenine kaydediyoruz. Daha sonra bu parametreye karşılık gelen kayıt bilgilerini veritabanından okuyup HTML formunu bu bilgilerle dolduruyoruz.

Aşağıdaki kodları **<!-- kategori bilgilerini getiren PHP kodu burada yer alacak -->** açıklama satırının yerine yazın.

```

<?php
    // gelen parametre değerini oku, bizim örneğimizde bu Id bilgisidir
    $id=isset($_GET['id']) ? $_GET['id'] : die('HATA: Id bilgisi bulunamadı.');
```

```

    // veritabanı bağlantı dosyasını dahil et
    include '../../config/vtabani.php';

    // aktif kayıt bilgilerini oku
    try {
        // seçme sorgusunu hazırla
        $sorgu = "SELECT id, kategoriadi, aciklama FROM kategoriler WHERE id = ?
LIMIT 0,1";
        $stmt = $con->prepare( $sorgu );

        // id parametresini bağla (? işaretini id değeri ile değiştir)
        $stmt->bindParam(1, $id);

        // sorguyu çalıştır
        $stmt->execute();

        // okunan kayıt bilgilerini bir değişkene kaydet
        $kayit = $stmt->fetch(PDO::FETCH_ASSOC);

        // formu dolduracak değişken bilgileri
        $kategoriadi = $kayit['kategoriadi'];
        $aciklama = $kayit['aciklama'];
    }
    // hatayı göster
    catch(PDOException $exception){
        die('HATA: ' . $exception->getMessage());
    }
?>

```

KAYIT GÜNCELLEMEK İÇİN GEREKLİ HTML FORMUNU OLUŞTURMAK

Aşağıdaki kodlar eski kayıt bilgilerini görüntüleyip yeni bilgilerin girilmesi için gerekli HTML formunu oluşturacaktır. Bu kodları `<!-- kategori bilgilerini düzeltme HTML formu burada yer alacak -->` açıklama satırının yerine yazın.

```
<!-- kaydı güncelleyecek PHP kodu burada yer alacak -->

<!-- kayıt bilgilerini güncelleyebileceğimiz HTML formu -->
<form action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]) .
"?id={ $id}");?>" method="post">
  <table class='table table-hover table-responsive table-bordered'>
    <tr>
      <td>Kategori adı</td>
      <td><input type='text' name='kategoriadi' value="<?php echo
htmlspecialchars($kategoriadi, ENT_QUOTES); ?>" class='form-control' /></td>
    </tr>
    <tr>
      <td>Açıklama</td>
      <td><textarea name='aciklama' class='form-control'><?php echo
htmlspecialchars($aciklama, ENT_QUOTES); ?></textarea></td>
    </tr>
    <tr>
      <td></td>
      <td>
        <input type='submit' value='Kaydet' class='btn btn-primary' />
        <a href='liste.php' class='btn btn-danger'>Kategori
listesi</a>
      </td>
    </tr>
  </table>
</form>
```

KAYIT GÜNCELLEME KODLARI

Kullanıcı formda bilgileri değiştirip **Kaydet** butonuna tıklarsa aşağıdaki kodlar bu değişiklikleri veritabanına kaydedecektir. Bu kodları `<!-- kaydı güncelleyecek PHP kodu burada yer alacak -->` açıklama satırının yerine yazın.

```
<?php
// Kaydet butonu tıklanmışsa
if($_POST){
  try{
    // güncelleme sorgusu
    // çok fazla parametre olduğundan karışmaması için
    // soru işaretleri yerine etiketler kullanacağız
    $sorgu = "UPDATE kategoriler SET kategoriadi=:kategoriadi,
aciklama=:aciklama WHERE id = :id";

    // sorguyu hazırla
    $stmt = $con->prepare($sorgu);

    // gelen bilgileri değişkenlere kaydet
    $kategoriadi=htmlspecialchars(strip_tags($_POST['kategoriadi']));
    $aciklama=htmlspecialchars(strip_tags($_POST['aciklama']));

    // parametreleri bağla
    $stmt->bindParam(':kategoriadi', $kategoriadi);
    $stmt->bindParam(':aciklama', $aciklama);
    $stmt->bindParam(':id', $id);

    // sorguyu çalıştır
    if($stmt->execute()){
      echo "<div class='alert alert-success'>Kayıt güncellendi.</div>";
    }
  }
}
```

```
        }else{
            echo "<div class='alert alert-danger'>Kayıt
güncellenemedi.</div>";
        }
    }
    // hata varsa göster
    catch(PDOException $exception){
        die('HATA: ' . $exception->getMessage());
    }
}
?>
```

Tarayıcınızın adres satırına **http://localhost/proje/admin/kategori/liste.php** yazarak herhangi bir kategori için **Düzeltil** butonunu tıklayın ve kategori güncelleme sayfasını görüntüleyin. Bilgilerde değişiklik yaparak **Kaydet** butonunu tıklayın.

PHP - PDO Projesi

localhost/proje/admin/kategori/duzelt.php?id=4

PHP - PDO Projesi Anasayfa Ürünler Kategoriler Oturumu kapat

Kategori Güncelleme

Kayıt güncellendi.

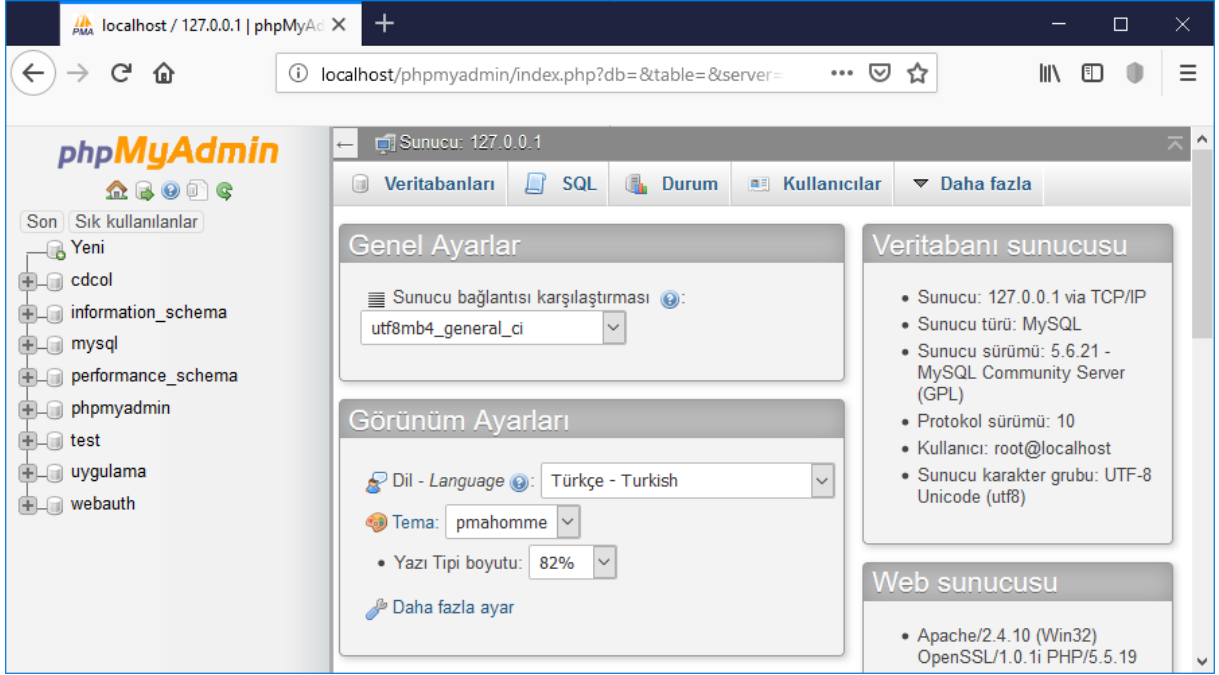
Kategori adı	<input type="text" value="Oyuncak"/>
Açıklama	<input type="text" value="Bebek, çocuk ve yetişkin oyuncaklarımız"/>
	<input type="button" value="Kaydet"/> <input type="button" value="Kategori listesi"/>

© Murat SARIÖZ - Mercin KARAKAŞ

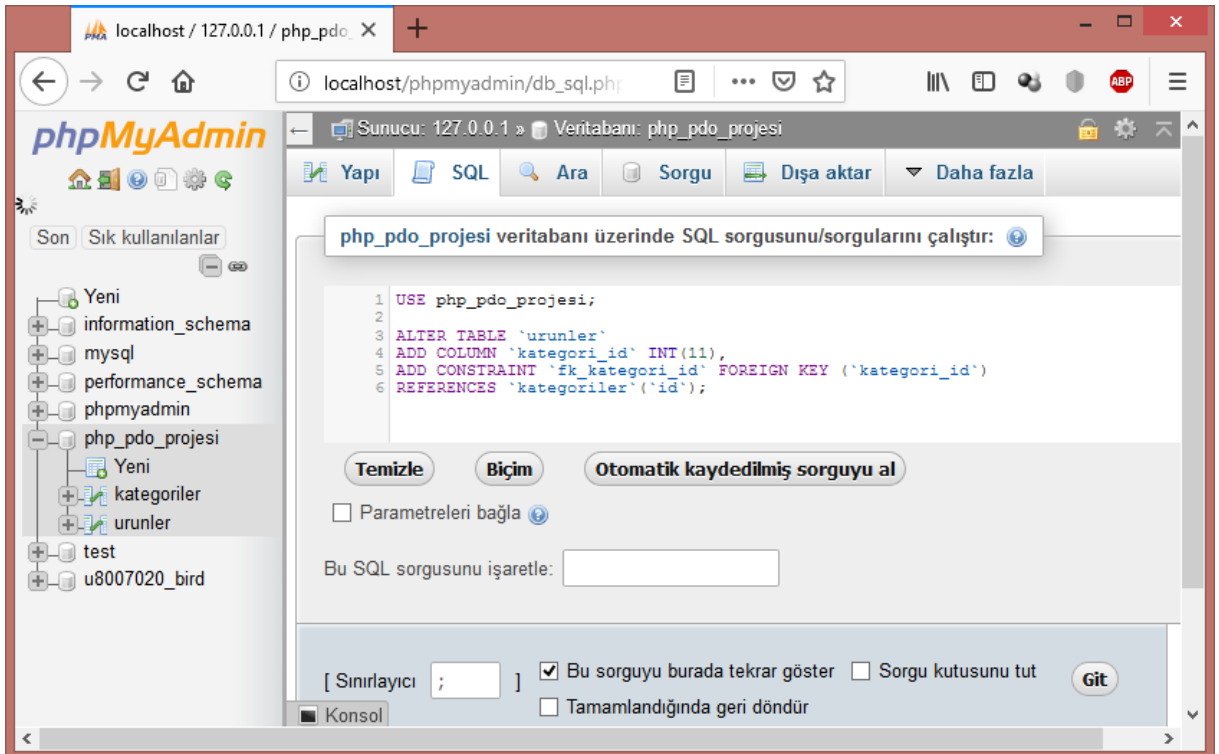
ÜRÜNLER TABLOSUYLA KATEGORİLER TABLOSUNU İLİŞKİLENDİRMEK

Artık hem ürünleri hem de kategorileri kaydedip düzenleyebileceğimiz sayfalarımız hazır. Sıra geldi ürün sayfalarında her ürünü bir kategori ile ilişkilendirmeye.

Veritabanında **urunler** tablosuyla **kategoriler** tablosunu ilişkilendirmek için tarayıcınızın adres satırına **http://localhost/phpmyadmin/** yazarak **phpMyAdmin** veritabanı yönetim panelini açın.



Şimdi **SQL** bağlantısına tıklayın ve aşağıdaki SQL sorgusunu yazarak **Git** butonuyla çalıştırın.



```
USE php_pdo_projesi;

ALTER TABLE `urunler`
ADD COLUMN `kategori_id` INT(11),
ADD CONSTRAINT `fk_kategori_id` FOREIGN KEY (`kategori_id`)
REFERENCES `kategoriler`(`id`);
```

ÜRÜN EKLEME SAYFASINA KATEGORİ ALANI EKLEMEK

Admin/urun/ekle.php dosyasını açın ve daha önce bilgi girişi için oluşturduğunuz HTML formundaki **Fiyat** alanı satır bitişi etiketiyle (</tr>) **Resim** alanı satır başlangıç etiketi (<tr>) arasına **Kategori** alanını da eklemek için aşağıdaki kodları yazın.

```
<tr>
  <td>Kategori</td>
  <td>
    <?php
      // veritabanı yapılandırma dosyasını dahil et
      include '../../config/vtabani.php';
      // kayıt listeleme sorgusu
      $sorgu='select id, kategoriadi from kategoriler';
      $stmt = $con->prepare($sorgu); // sorguyu hazırla
      $stmt->execute(); // sorguyu çalıştır
      $veri = $stmt->fetchAll(PDO::FETCH_ASSOC); // tablo verilerini oku
    ?>
    <select name='kategori_id' class='form-control'>
      <?php foreach ($veri as $kayit) { ?>
        <option value="<?php echo $kayit["id"] ?>">
          <?php echo $kayit["kategoriadi"] ?>
        </option>
      <?php } ?>
    </select>
  </td>
</tr>
```

Daha sonra PHP kayıt ekleme kodlarını düzenlemeliyiz, çünkü artık **urunler** tablosuna kaydedilecek yeni bir alanımız (**kategori_id**) daha var. Bunun için **try{** komut ifadesinden sonraki sorgu kodlarında değişiklik yapacağız. PHP kayıt ekleme kodlarını aşağıdaki gibi düzenleyin.

Eski kod:

```
// kayıt ekleme sorgusu
$sorgu = "INSERT INTO urunler SET urunadi=:urunadi, aciklama=:aciklama,
fiyat=:fiyat, giris_tarihi=:giris_tarihi, resim=:resim";

// sorguyu hazırla
$stmt = $con->prepare($sorgu);

// post edilen değerler
$urunadi=htmlspecialchars(strip_tags($_POST['urunadi']));
$aciklama=htmlspecialchars(strip_tags($_POST['aciklama']));
$fiyat=htmlspecialchars(strip_tags($_POST['fiyat']));
// yeni 'resim' alanı
$resim=!empty($_FILES["resim"]["name"]) ? uniqid() . "-" .
basename($_FILES["resim"]["name"]) : "";
$resim=htmlspecialchars(strip_tags($resim));

// parametreleri bağla
$stmt->bindParam(':urunadi', $urunadi);
$stmt->bindParam(':aciklama', $aciklama);
$stmt->bindParam(':fiyat', $fiyat);
$stmt->bindParam(':resim', $resim);
// ürünün veritabanına kaydedildiği tarihi belirt
$giris_tarihi=date('Y-m-d H:i:s');
$stmt->bindParam(':giris_tarihi', $giris_tarihi);
```

Yeni Kod:

```
// kayıt ekleme sorgusu
$sorgu = "INSERT INTO urunler SET urunadi=:urunadi, aciklama=:aciklama,
fiyat=:fiyat, giris_tarihi=:giris_tarihi, resim=:resim, kategori_id=:kategori_id";

// sorguyu hazırla
```

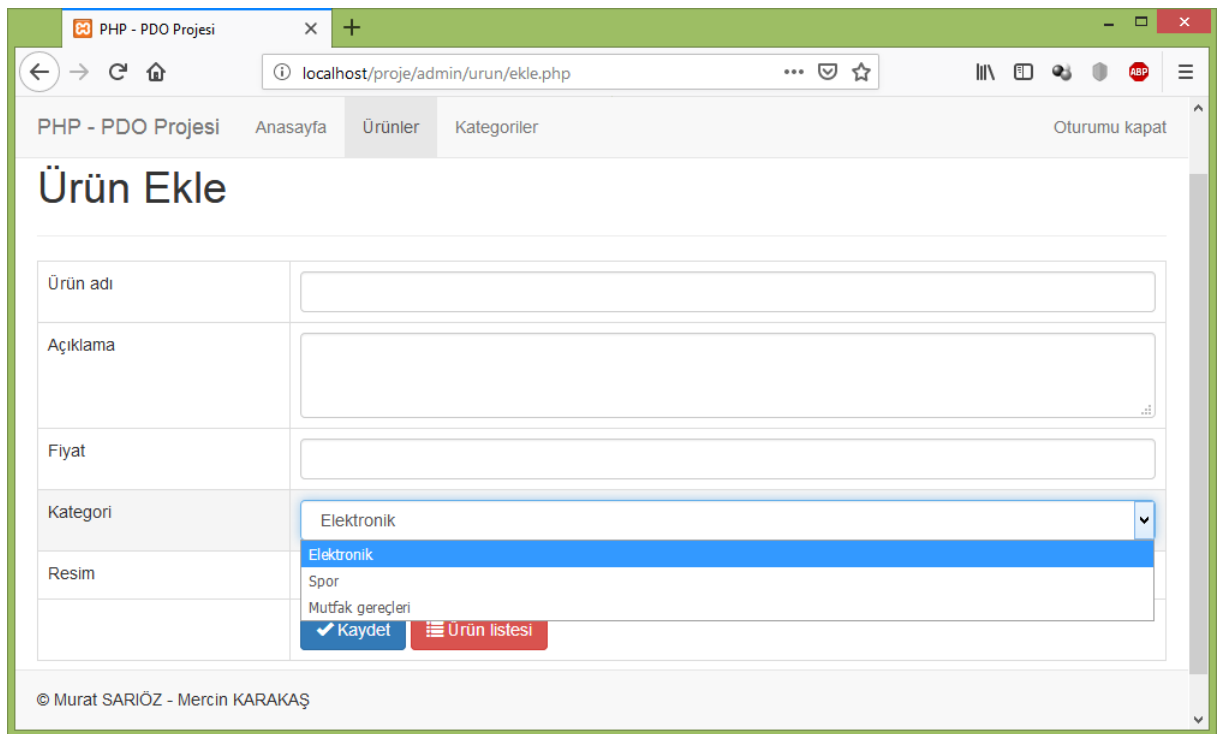
```

$stmt = $con->prepare($sorgu);

// post edilen deęerler
$urunadi=htmlspecialchars(strip_tags($_POST['urunadi']));
$aciklama=htmlspecialchars(strip_tags($_POST['aciklama']));
$fiyat=htmlspecialchars(strip_tags($_POST['fiyat']));
$resim=!empty($_FILES["resim"]["name"]) ? uniqid() . "-" .
basename($_FILES["resim"]["name"]) : "";
$resim=htmlspecialchars(strip_tags($resim));
$kategori_id=htmlspecialchars(strip_tags($_POST['kategori_id']));

// parametreleri baęla
$stmt->bindParam(':urunadi', $urunadi);
$stmt->bindParam(':aciklama', $aciklama);
$stmt->bindParam(':fiyat', $fiyat);
$stmt->bindParam(':resim', $resim);
$stmt->bindParam(':kategori_id', $kategori_id);
// ürünün veritabanına kaydedildięi tarihi belirt
$giris_tarihi=date('Y-m-d H:i:s');
$stmt->bindParam(':giris_tarihi', $giris_tarihi);

```



ÜRÜN GÜNCELLEME SAYFASINA KATEGORİ ALANI EKLEMEK

Admin/urun/duzelt.php dosyasını açın ve daha önce bilgi girişi için oluşturduğunuz HTML formundaki **Fiyat** alanı satır bitiş etiketiyle (`</tr>`) **Kaydet** butonu satır başlangıç etiketi (`<tr>`) arasına **Kategori** alanını da eklemek için aşağıdaki kodları yazın.

```

<tr>
  <td>Kategori</td>
  <td>
    <?php
      // kayıt listeleme sorgusu
      $sorgu='select id, kategoriadi from kategoriler';
      $stmt = $con->prepare($sorgu); // sorguyu hazırla
      $stmt->execute(); // sorguyu çalıştır
      $veri = $stmt->fetchAll(PDO::FETCH_ASSOC); ; // tablo verilerini oku
    ?>
    <select name='kategori_id' class='form-control'>
    <?php foreach ($veri as $kayit) { ?>
      <option value="<?php echo $kayit["id"] ?>"

```



```
<?php if($kayit["id"]==$kategori_id) echo " selected" ?>>
<?php echo $kayit["kategoriadi"] ?>
</option>
<?php } ?>
</select>
</td>
</tr>
```

Daha sonra ürün bilgilerini getiren PHP kodlarını düzenlemeliyiz. Bunun için **try{** komut ifadesinden sonraki sorgu kodlarında değişiklik yapacağız. Ürün bilgilerini getiren PHP kodlarını aşağıdaki gibi düzenleyin.

Eski kod:

```
// seçme sorgusunu hazırla
$sorgu = "SELECT id, urunadi, aciklama, fiyat FROM urunler WHERE id = ? LIMIT 0,1";
$stmt = $con->prepare( $sorgu );

// id parametresini bağla (? işaretini id değeri ile değiştir)
$stmt->bindParam(1, $id);

// sorguyu çalıştır
$stmt->execute();

// okunan kayıt bilgilerini bir değişkene kaydet
$kayit = $stmt->fetch(PDO::FETCH_ASSOC);

// formu dolduracak değişken bilgileri
$urunadi = $kayit['urunadi'];
$aciklama = $kayit['aciklama'];
$fiyat = $kayit['fiyat'];
```

Yeni Kod:

```
// seçme sorgusunu hazırla
$sorgu = "SELECT id, urunadi, aciklama, fiyat, kategori_id FROM urunler WHERE id = ? LIMIT 0,1";
$stmt = $con->prepare( $sorgu );

// id parametresini bağla (? işaretini id değeri ile değiştir)
$stmt->bindParam(1, $id);

// sorguyu çalıştır
$stmt->execute();

// okunan kayıt bilgilerini bir değişkene kaydet
$kayit = $stmt->fetch(PDO::FETCH_ASSOC);

// formu dolduracak değişken bilgileri
$urunadi = $kayit['urunadi'];
$aciklama = $kayit['aciklama'];
$fiyat = $kayit['fiyat'];
$kategori_id = $kayit['kategori_id'];
```

Son olarak kaydı güncelleyecek PHP kodlarında da düzenleme yapmalıyız. Bunun için **try{** komut ifadesinden sonraki sorgu kodlarında değişiklik yapacağız. Kaydı güncelleyecek PHP kodlarını aşağıdaki gibi düzenleyin.

Eski kod:

```
// güncelleme sorgusu
$sorgu = "UPDATE urunler
SET urunadi=:urunadi, aciklama=:aciklama, fiyat=:fiyat
WHERE id = :id";
```

```

// sorguyu hazırla
$stmt = $con->prepare($sorgu);

// gelen bilgileri değişkenlere kaydet
$urunadi=htmlspecialchars(strip_tags($_POST['urunadi']));
$aciklama=htmlspecialchars(strip_tags($_POST['aciklama']));
$fiyat=htmlspecialchars(strip_tags($_POST['fiyat']));

// parametreleri bağla
$stmt->bindParam(':urunadi', $urunadi);
$stmt->bindParam(':aciklama', $aciklama);
$stmt->bindParam(':fiyat', $fiyat);
$stmt->bindParam(':id', $id);

```

Yeni Kod:

```

// güncelleme sorgusu
$sorgu = "UPDATE urunler
        SET urunadi=:urunadi, aciklama=:aciklama, fiyat=:fiyat,
        kategori_id=:kategori_id
        WHERE id = :id";

// sorguyu hazırla
$stmt = $con->prepare($sorgu);

// gelen bilgileri değişkenlere kaydet
$urunadi=htmlspecialchars(strip_tags($_POST['urunadi']));
$aciklama=htmlspecialchars(strip_tags($_POST['aciklama']));
$fiyat=htmlspecialchars(strip_tags($_POST['fiyat']));
$kategori_id=htmlspecialchars(strip_tags($_POST['kategori_id']));

// parametreleri bağla
$stmt->bindParam(':urunadi', $urunadi);
$stmt->bindParam(':aciklama', $aciklama);
$stmt->bindParam(':fiyat', $fiyat);
$stmt->bindParam(':kategori_id', $kategori_id);
$stmt->bindParam(':id', $id);

```

PHP - PDO Projesi

localhost/proje/admin/urun/duzelt.php?id=11

PHP - PDO Projesi Anasayfa Ürünler Kategoriler Oturumu kapat

Ürün Güncelleme

Ürün adı	Web kamerası
Açıklama	Görüntülü görüşme için.
Fiyat	20
Kategori	Elektronik

© Murat SARIÖZ - Mercin KARAKAŞ

ÜRÜN LİSTELEME SAYFASINA KATEGORİ ALANI EKLEMEK

Admin/urun/liste.php dosyasını açın. Sayfada görüntülenecek kayıtları getiren seçme sorgusunda değişiklik yapacağız. Ürün bilgilerini ve ilgili kategori bilgisini getiren sorguyu aşağıdaki gibi düzenleyin.

Eski sorgu:

```
$sorgu = "SELECT id, urunadi, aciklama, fiyat FROM urunler WHERE urunadi LIKE :aranan ORDER BY id DESC LIMIT :ilk_kayit_no, :sayfa_kayit_sayisi";
```

Yeni sorgu:

```
$sorgu = "SELECT urunler.id, urunler.urunadi, urunler.aciklama, urunler.fiyat, kategoriler.kategoriadi FROM urunler LEFT JOIN kategoriler ON urunler.kategori_id = kategoriler.id WHERE urunler.urunadi LIKE :aranan ORDER BY urunler.id DESC LIMIT :ilk_kayit_no, :sayfa_kayit_sayisi";
```

Not: Tablolar arasında ilişki kurarken **LEFT JOIN** kullanmamızın sebebi kategori ilişkilendirmesi yapılmamış kayıtların da listelenmesini sağlamaktır. **INNER JOIN** kullanmış olsaydık bu kayıtlar listelenmeyecekti.

Daha sonra ürün bilgilerini listeleyen HTML tablo kodlarını da kategori bilgisini gösterecek şekilde düzenlemeliyiz. Bunun için öncelikle tablo başlıklarına **Kategori** alanını eklemeliyiz.

```
echo "<th>Fiyat</th>";
```

satırından sonra

```
echo "<th>Kategori</th>";
```

satırını ekleyin.

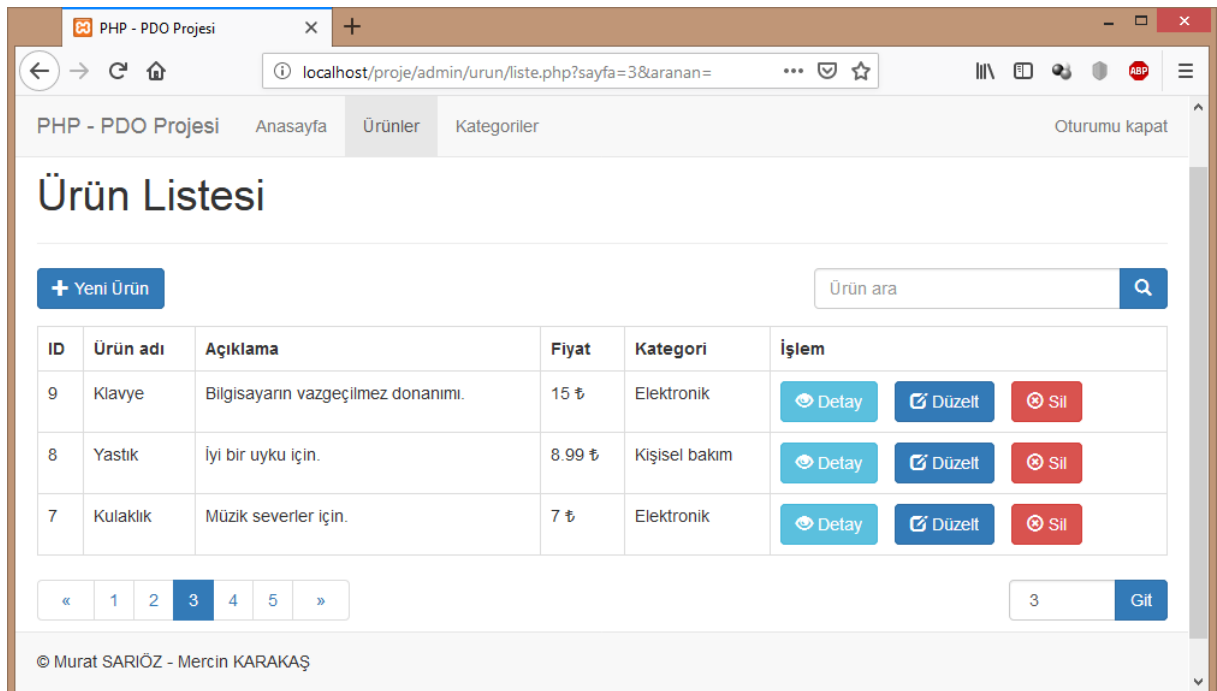
Ardından ürünlerin kategori bilgisini gösteren tablo satırlarının arasına aşağıdaki kod satırını ekleyin.

```
echo "<td>{$fiyat} &#8378;</td>";
```

kod satırından sonra

```
echo "<td>{$kategoriadi}</td>";
```

kod satırını ekleyin.



The screenshot shows a web browser window displaying a product list. The browser address bar shows the URL: localhost/proje/admin/urun/liste.php?sayfa=3&aranan=. The page title is "PHP - PDO Projesi" and the breadcrumb navigation shows "Anasayfa > Ürünler > Kategoriler". The main heading is "Ürün Listesi". There is a search bar with the text "Ürün ara" and a search icon. Below the search bar is a table with the following columns: ID, Ürün adı, Açıklama, Fiyat, Kategori, and İşlem. The table contains three rows of data:

ID	Ürün adı	Açıklama	Fiyat	Kategori	İşlem
9	Klavye	Bilgisayarın vazgeçilmez donanımı.	15 ₺	Elektronik	Detay Düzeltil Sil
8	Yastık	İyi bir uyku için.	8.99 ₺	Kişisel bakım	Detay Düzeltil Sil
7	Kulaklık	Müzik severler için.	7 ₺	Elektronik	Detay Düzeltil Sil

At the bottom of the table, there is a pagination control showing page 3 of 3, with a "Git" button.

ÜRÜN DETAY SAYFASINA KATEGORİ ALANI EKLEMEK

Admin/urun/detay.php dosyasını açın. Sayfada görüntülenecek kayıt bilgilerini getiren seçme sorgusunda değişiklik yapacağız. Ürün bilgilerini ve ilgili kategori bilgisini getiren sorguyu aşağıdaki gibi düzenleyin.

Eski sorgu:

```
$sorgu = "SELECT id, urunadi, aciklama, fiyat, resim FROM urunler WHERE id = ?
LIMIT 0,1";
$stmt = $con->prepare( $sorgu );

// Id parametresini bağla
$stmt->bindParam(1, $id);

// sorguyu çalıştır
$stmt->execute();

// gelen kaydı bir değişkende sakla
$kayit = $stmt->fetch(PDO::FETCH_ASSOC);

// tabloya yazılacak bilgileri değişkenlere doldur
$urunadi = $kayit['urunadi'];
$aciklama = $kayit['aciklama'];
$fiyat = $kayit['fiyat'];
$resim = $kayit['resim'];
```

Yeni sorgu:

```
$sorgu = "SELECT urunler.id, urunler.urunadi, urunler.aciklama, urunler.fiyat,
urunler.resim, kategoriler.kategoriadi FROM urunler LEFT JOIN kategoriler ON
urunler.kategori_id = kategoriler.id WHERE urunler.id = ? LIMIT 0,1";
$stmt = $con->prepare( $sorgu );

// Id parametresini bağla
$stmt->bindParam(1, $id);

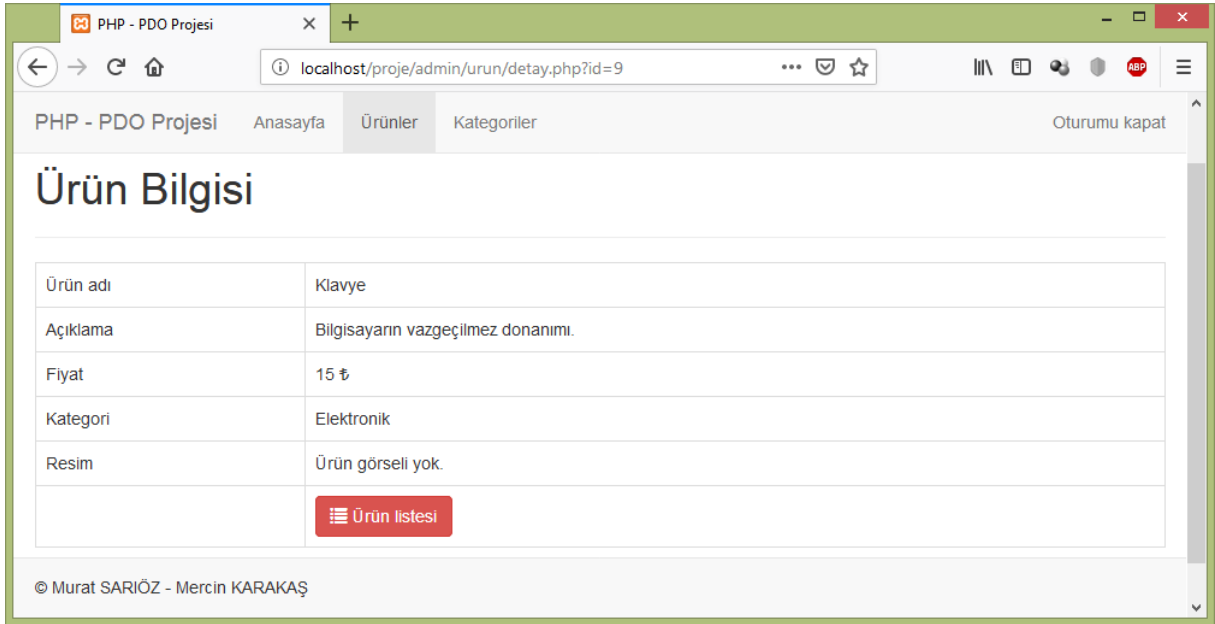
// sorguyu çalıştır
$stmt->execute();

// gelen kaydı bir değişkende sakla
$kayit = $stmt->fetch(PDO::FETCH_ASSOC);

// tabloya yazılacak bilgileri değişkenlere doldur
$urunadi = $kayit['urunadi'];
$aciklama = $kayit['aciklama'];
$fiyat = $kayit['fiyat'];
$resim = $kayit['resim'];
$kategoriadi = $kayit['kategoriadi'];
```

Daha sonra ürün bilgilerini gösteren HTML tablo kodlarını da kategori bilgisini gösterecek şekilde düzenlemeliyiz. HTML tablosundaki **Fiyat** alanı satır bitiş etiketiyle (</tr>) **Resim** alanı satır başlangıç etiketi (<tr>) arasına **Kategori** alanını da eklemek için aşağıdaki kodları yazın.

```
<tr>
  <td>Kategori</td>
  <td><?php echo htmlspecialchars($kategoriadi, ENT_QUOTES); ?></td>
</tr>
```

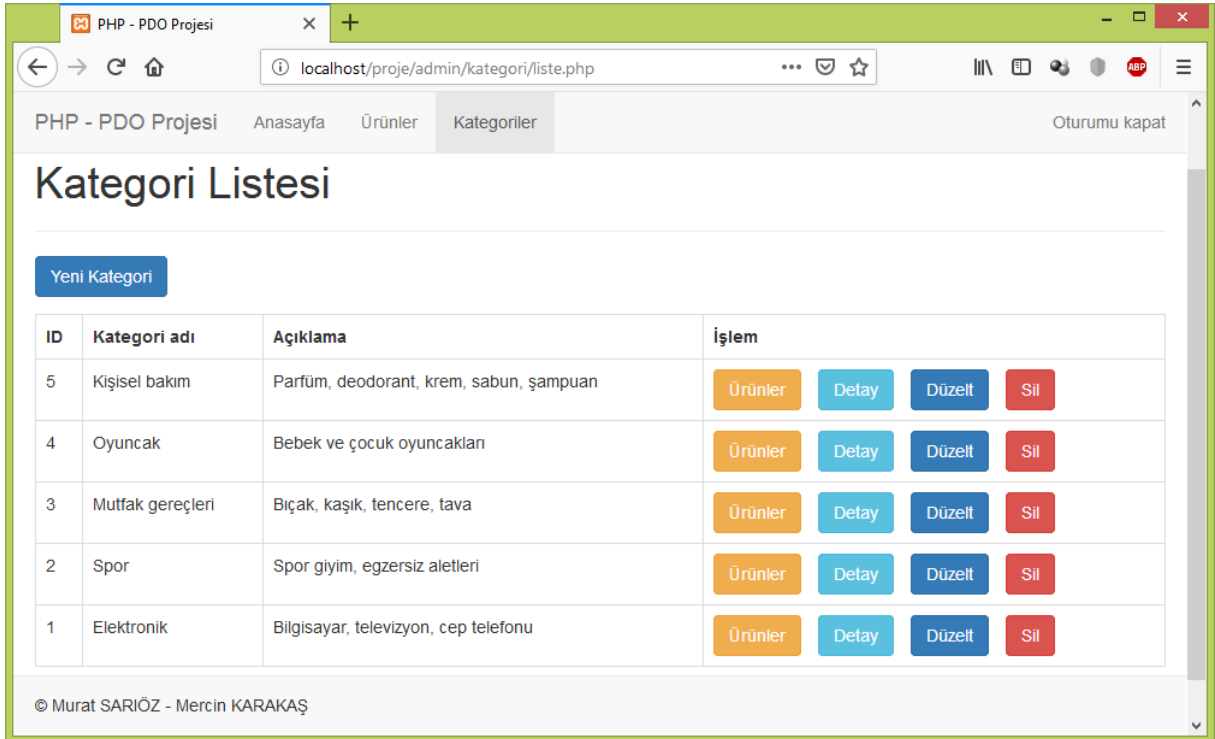


KATEGORİ SAYFASINA ÜRÜN LİSTESİ EKLEMEK

Admin/kategori/liste.php dosyasını açın ve **Detay** butonundan önce **Ürünler** isimli butonu oluşturun. Bunun için **// kayıt detay sayfa bağlantısı** açıklama satırından önce aşağıdaki kod satırını ekleyin.

```
// kategori ürünleri sayfa bağlantısı  
echo "<a href='urunler.php?id={\$id}' class='btn btn-warning m-r-1em'>Ürünler</a>";
```

Tarayıcınızın adres satırına **http://localhost/proje/admin/kategori/liste.php** yazarak kategori listeleme sayfasını görüntüleyin.



KATEGORİ ÜRÜNLERİ SAYFASINI OLUŞTURMAK

Kategori liste sayfasında **Ürünler** butonu tıklanınca ürün bilgilerini listeleyecek sayfamızı oluşturmaya başlamadan önce **admin/kategori** klasörünü açın. Bu klasör içinde **urunler.php** isimli bir dosya oluşturun ve içine aşağıdaki satırları ekleyin.

```
<?php include "../header.php"; ?>
```

```
<div class="container">
```

```
    <div class="page-header">
```

```
        <h1>Ürün Listesi</h1>
```

```
    </div>
```

```
        <!-- Kayıtları listeleyecek PHP kodları bu alana eklenecek -->
```

```
</div> <!-- /container -->
```

```
<?php include "../footer.php"; ?>
```

KAYIT LİSTELEMELİK İÇİN GEREKLİ SQL SORGUSUNU OLUŞTURMAK

Ardından <!-- Kayıtları listeleyecek PHP kodları bu alana eklenecek --> açıklama satırının yerine aşağıdaki PHP kodlarını yazın.

```
<?php
// veritabanı bağlantı dosyasını çağır
include '../config/vtabani.php';

// gelen kategori parametresini oku
$id = isset($_GET['id']) ? $_GET['id'] : "";

// kategoriye ait kayıtları seç
$sorgu = "SELECT id, urunadi, aciklama, fiyat FROM urunler WHERE kategori_id = ?
ORDER BY id DESC";

$stmt = $con->prepare($sorgu);
// Id parametresini bağla
$stmt->bindParam(1, $id);
$stmt->execute();

// geriye dönen kayıt sayısı
$sayi = $stmt->rowCount();

echo "<a href='liste.php' class='btn btn-danger m-b-1em'>Kategori listesi</a>";

//kayıt varsa listele
if($sayi>0){

    // kayıtlar burada listelenecek

}
// kayıt yoksa mesajla bildir
else{
    echo "<div class='alert alert-danger'>Listelenecek kayıt bulunamadı.</div>";
}
?>
```

KAYIT LİSTELEMELİK İÇİN GEREKLİ HTML TABLOSUNU OLUŞTURMAK

Sıra geldi veritabanından okunan kayıtların listelenmesine. Bunun için HTML tablo yapısını kullanacağız.

// kayıtlar burada listelenecek açıklama satırının yerine aşağıdaki PHP kodlarını yazın.

```
echo "<table class='table table-hover table-responsive table-bordered'>";
//tablo başlangıcı

//tablo başlıkları
echo "<tr>";
    echo "<th>ID</th>";
    echo "<th>Ürün adı</th>";
```

```
        echo "<th>Açıklama</th>";
        echo "<th>Fiyat</th>";
    echo "</tr>";

    // tablo içeriği burada yer alacak

echo "</table>"; // tablo sonu
```

VERİTABANINDAKİ KAYITLARI LİSTELEMELİK İÇİN GEREKLİ PHP KODLARI

Tablonun ana yapısını oluşturduktan sonra, içine verileri dolduracak PHP kodlarını da **// tablo içeriği burada yer alacak** açıklama satırının yerine yazın.

```
// tablo verilerinin okunması
while ($kayit = $stmt->fetch(PDO::FETCH_ASSOC)){
    // tablo alanlarını değişkene dönüştür
    // $kayit['urunadi'] => $urunadi
    extract($kayit);

    // her kayıt için yeni bir tablo satırı oluştur
    echo "<tr>";
        echo "<td>{$id}</td>";
        echo "<td>{$urunadi}</td>";
        echo "<td>{$aciklama}</td>";
        echo "<td>{$fiyat} &#8378;</td>"; // &#8378; ==> TL işareti
    echo "</tr>";
}
```

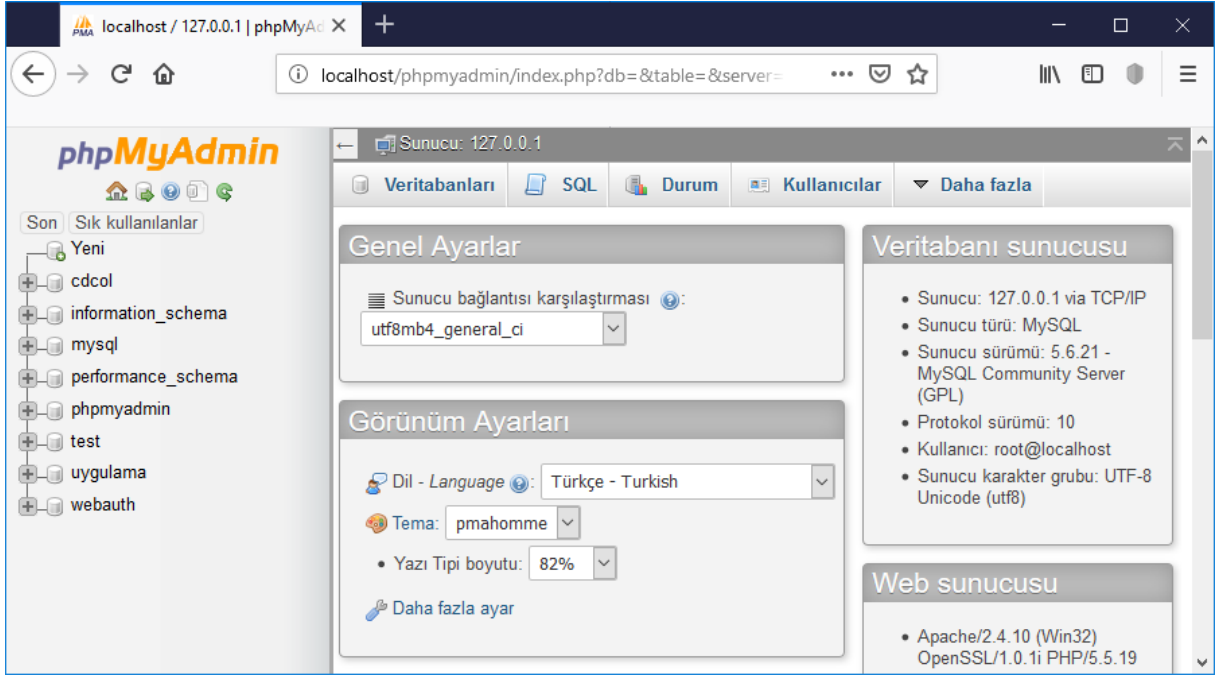
Tarayıcınızın adres satırına **http://localhost/proje/admin/kategori/liste.php** yazarak herhangi bir kategori için ürün listeleme sayfasını görüntüleyin.

ID	Ürün adı	Açıklama	Fiyat
16	dfs	sdfsdf	343 ₺
13	Duvar tablosu	Sarıasma kuşu tablosu.	99.99 ₺
12	Yapışkan etiket	Salyangoz fotoğrafı.	2 ₺
5	Çöp kovası	Etrafı temiz tutmanız için.	3.95 ₺

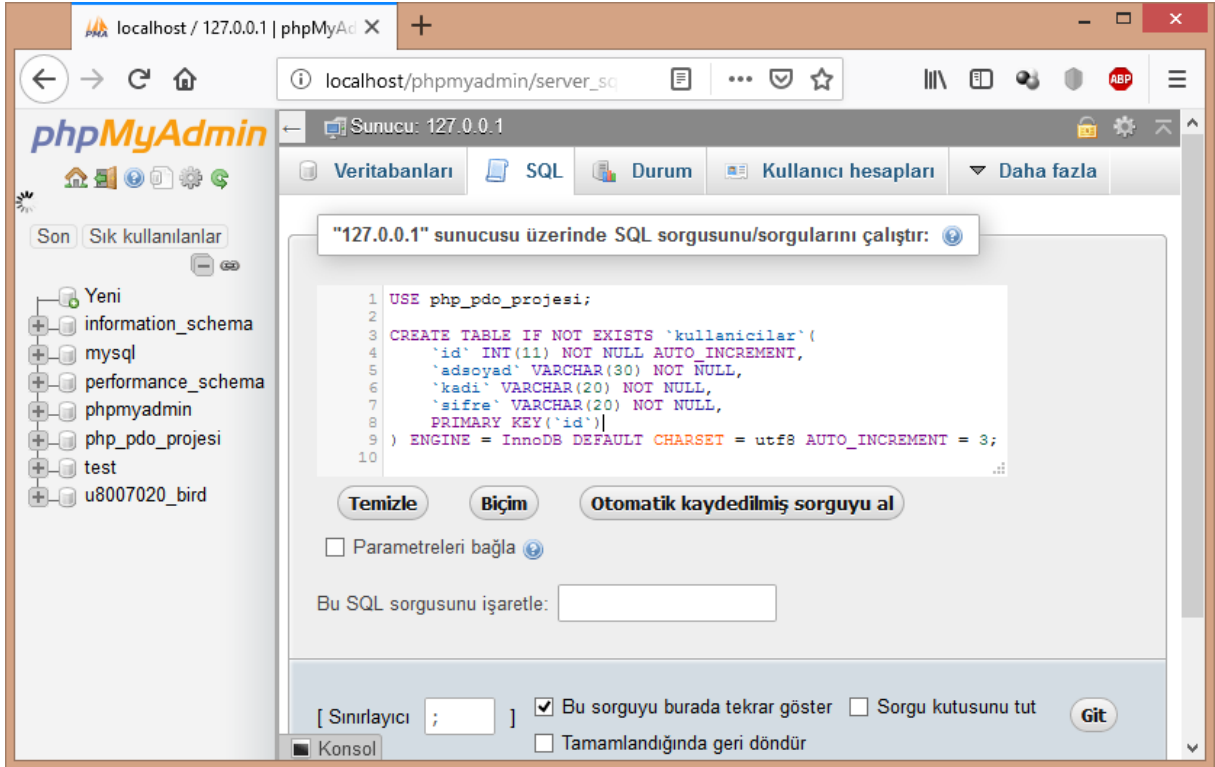
KULLANICI SAYFALARINI OLUŞTURMAK

Yönetim paneline giriş kontrolü için kullanıcı adı ve şifre bilgileri kullanılacak. Öncelikle bu bilgileri saklayacağımız tabloyu oluşturmalıyız. Daha sonra da tabloya bilgi girişi ve bilgi düzenleme işlemleri için web sayfalarını hazırlamalıyız.

Veritabanında **kullanıcılar** tablosunu oluşturmak için tarayıcınızın adres satırına <http://localhost/phpmyadmin/> yazarak **phpMyAdmin** veritabanı yönetim panelini açın.



Şimdi **SQL** bağlantısına tıklayın ve aşağıdaki SQL sorgusunu yazarak **Git** butonuyla çalıştırın.



```
USE php_pdo_projesi;

CREATE TABLE IF NOT EXISTS `kullaniciilar` (
  `id` INT(11) NOT NULL AUTO_INCREMENT,
  `adsoyad` VARCHAR(30) NOT NULL,
  `kadi` VARCHAR(20) NOT NULL,
  `sifre` VARCHAR(20) NOT NULL,
  PRIMARY KEY(`id`)
```



```
PRIMARY KEY(`id`)  
) ENGINE = InnoDB DEFAULT CHARSET = utf8 AUTO_INCREMENT = 3;
```

Ardından **SQL** bağlantısına tekrar tıklayın ve aşağıdaki SQL komutunu çalıştırarak oluşturduğunuz **kullanıcılar** tablosuna örnek veriler kaydedin.

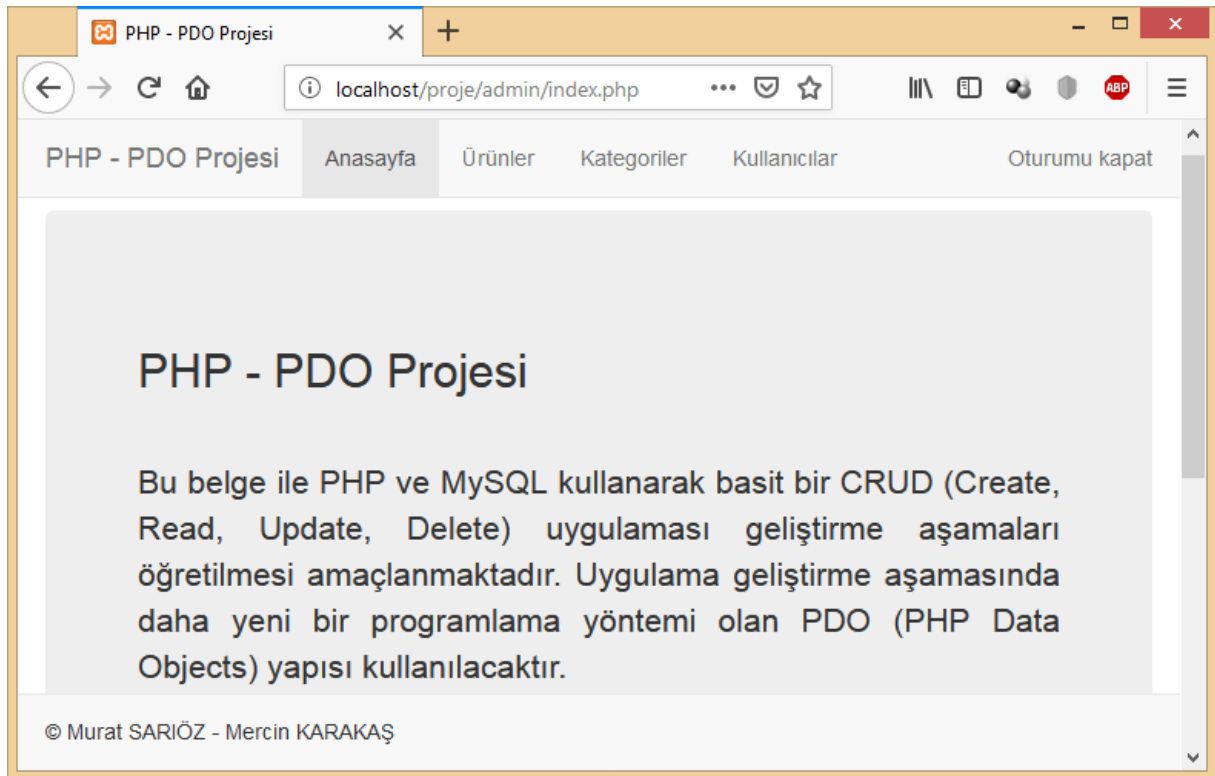
```
INSERT INTO `kullanıcılar` ( `id`, `adsoyad`, `kadi`, `sifre`) VALUES  
( 1, 'Mercin Karakaş', 'mercin', '1234'),  
( 2, 'Murat Sarıöz', 'murat', 'abc')
```

ANA SAYFADA KULLANICI MENÜSÜNÜ OLUŞTURMAK

Kullanıcı sayfalarını oluşturmadan önce ana sayfadaki menüye **Kullanıcılar** seçeneğini de eklemeliyiz. **admin/header.php** sayfasını açın ve Bootstrap Navbar menü kodlarında **Kategoriler** menü seçeneğinin liste kapama etiketinden (``) sonra aşağıdaki satırı ekleyin.

```
<li <?php echo((strpos($aktif_link, 'kullanici/') !== false) ? 'class="active"' :  
''); ?><a href="/proje/admin/kullanici/liste.php">Kullanıcılar</a></li>
```

Tarayıcınızın adres satırına **http://localhost/proje/admin/index.php** yazarak proje ana sayfasını görüntüleyin.



KULLANICI LİSTELEME SAYFASINI OLUŞTURMAK

Daha sonra **admin** klasörü içinde **kullanici** isimli bir klasör oluşturun. Bu klasör içinde **liste.php** isimli bir dosya oluşturun ve içine aşağıdaki satırları ekleyin.

```
<?php include "../header.php"; ?>  
  
<div class="container">  
    <div class="page-header">  
        <h1>Kullanıcı Listesi</h1>  
    </div>  
  
    <!-- Kayıtları listeleyecek PHP kodları bu alana eklenecek -->
```

```
</div> <!-- /container -->
<?php include "../footer.php"; ?>
<!-- Kayıt silme onay kodları bu alana eklenecek -->
```

KAYIT LİSTELEMEK İÇİN GEREKLİ SQL SORGUSUNU OLUŞTURMAK

Ardından <!-- Kayıtları listeleyecek PHP kodları bu alana eklenecek --> açıklama satırının yerine aşağıdaki PHP kodlarını yazın.

```
<?php
// veritabanı bağlantı dosyasını çağır
include '../../config/vtabani.php';

// silme mesajı burada yer alacak

// bütün kayıtları seç
$sorgu = "SELECT id, adsoyad, kadi FROM kullanicilar ORDER BY id DESC";
$stmt = $con->prepare($sorgu);
$stmt->execute();

// geriye dönen kayıt sayısı
$sayi = $stmt->rowCount();

// kayıt ekleme sayfasının linki
echo "<a href='ekle.php' class='btn btn-primary m-b-1em'>Yeni Kullanıcı</a>";

//kayıt varsa listele
if($sayi>0){

    // kayıtlar burada listelenecek

}
// kayıt yoksa mesajla bildir
else{
    echo "<div class='alert alert-danger'>Listelenecek kayıt bulunamadı.</div>";
}
?>
```

KAYIT LİSTELEMEK İÇİN GEREKLİ HTML TABLOSUNU OLUŞTURMAK

Sıra geldi veritabanından okunan kayıtların listelenmesine. Bunun için HTML tablo yapısını kullanacağız. // kayıtlar burada listelenecek açıklama satırının yerine aşağıdaki PHP kodlarını yazın.

```
echo "<table class='table table-hover table-responsive table-bordered'>";
//tablo başlangıcı

//tablo başlıkları
echo "<tr>";
    echo "<th>ID</th>";
    echo "<th>Ad ve Soyad</th>";
    echo "<th>Kullanıcı adı</th>";
    echo "<th>İşlem</th>";
echo "</tr>";

// tablo içeriği burada yer alacak

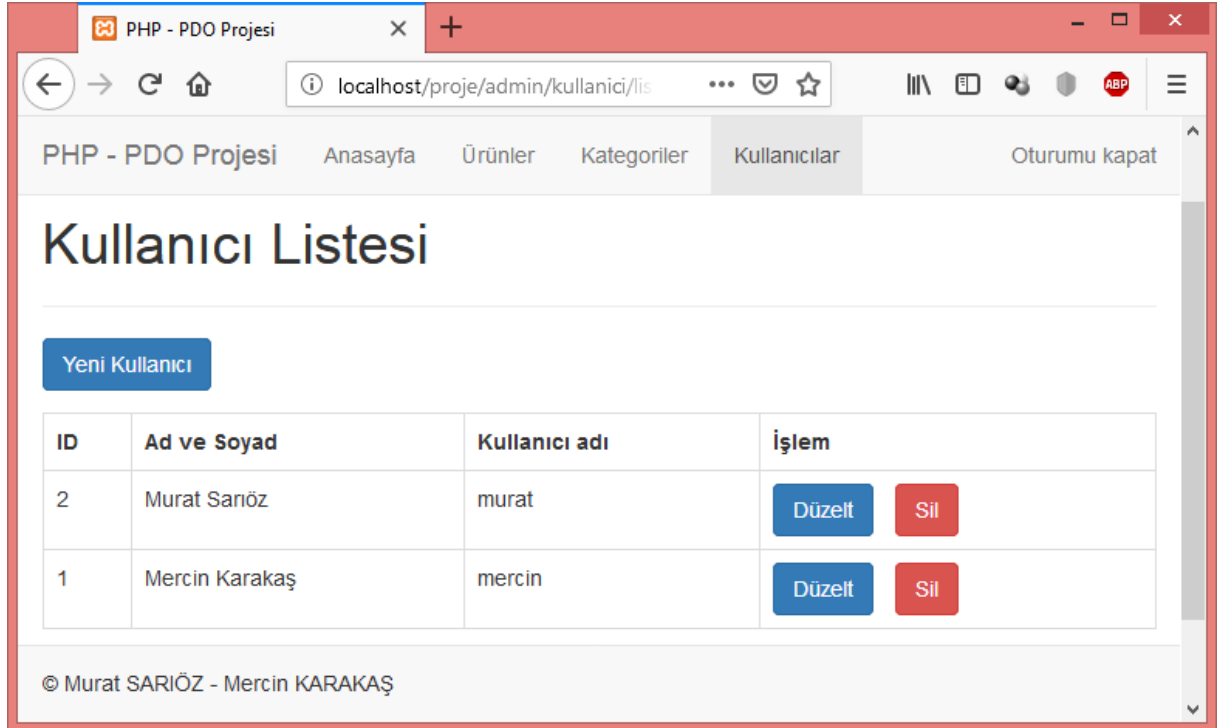
echo "</table>"; // tablo sonu
```

Tablonun ana yapısını oluşturduktan sonra, içine verileri dolduracak PHP kodlarını da // **tablo içeriği burada yer alacak** açıklama satırının yerine yazın.

```
// tablo verilerinin okunması
while ($kayit = $stmt->fetch(PDO::FETCH_ASSOC)){
    // tablo alanlarını deęişkene dönüştürür
    // $kayit['adsoyad'] => $adsoyad
    extract($kayit);

    // her kayıt için yeni bir tablo satırı oluştur
    echo "<tr>";
        echo "<td>{$id}</td>";
        echo "<td>{$adsoyad}</td>";
        echo "<td>{$kadi}</td>";
        echo "<td>";
            // kayıt güncelleme sayfa bağlantısı
            echo "<a href='duzelt.php?id={$id}' class='btn btn-primary m-r-1em'>Düzeltil</a>";
            // kayıt silme butonu
            echo "<a href='#' onclick='silme_onay({$id});' class='btn btn-danger'>Sil</a>";
        echo "</td>";
    echo "</tr>";
}
}
```

Tarayıcınızın adres satırına **http://localhost/proje/admin/kullanici/liste.php** yazarak ürün listeleme sayfasını görüntüleyin.



KULLANICI SİLME SAYFASINI OLUŞTURMAK

Bir kaydı silmek için işlem butonlarından **Sil** butonuna tıklayınca önce kullanıcının onayını alacağız. Böylelikle yanlışlıkla kayıtların silinmesini önleyebiliriz. Silme işlemi onayını **liste.php** sayfasında, silme işlemi ise kod karışıklığını azaltmak için ayrı bir sayfada (**sil.php**) yapacağız.

SİLME İŞLEMİNDEN ÖNCE KULLANICI ONAYINI ALMAK

Öncelikle **liste.php** dosyasını açın ve **<!-- Kayıt silme onay kodları bu alana eklenecek -->** açıklama satırının yerine aşağıdaki JavaScript kodlarını yazın.

```

<script type='text/javascript'>
  // kayıt silme işlemini onayla
  function silme_onay( id ){

    var cevap = confirm('Kaydı silmek istiyor musunuz?');
    if (cevap){
      // kullanıcı evet derse,
      // id bilgisini sil.php sayfasına yönlendirir
      window.location = 'sil.php?id=' + id;
    }
  }
</script>

```

VERİTABANINDAN KAYIT SİLMEK İÇİN GEREKLİ PHP KODLARI

Şimdi **admin/kullanici** klasörü içinde **sil.php** isimli bir dosya oluşturun ve içine aşağıdaki PHP kodlarını yazın.

```

<?php
// veritabanı ayar dosyasını dahil et
include '../../config/vtabani.php';

try {
  // kaydın id bilgisini al
  $id=isset($_GET['id']) ? $_GET['id'] : die('HATA: Id bilgisi bulunamadı.');
```

```

  // silme sorgusu
  $sorgu = "DELETE FROM kullanicilar WHERE id = ?";
  $stmt = $con->prepare($sorgu);
  $stmt->bindParam(1, $id);

  // sorguyu çalıştır
  if($stmt->execute()){
    // kayıt listeleme sayfasına yönlendir
    // ve kullanıcıya kaydın silindiğini
    header('Location: liste.php?islem=silindi');
  } // veya silinemediğini bildir
  else{
    header('Location: liste.php?islem=silinemedi');
  }
}
// hata varsa göster
catch(PDOException $exception){
  die('HATA: ' . $exception->getMessage());
}
?>

```

SİLME İŞLEMİNDEN SONRA KULLANICIYA BİLGİ VERMEK

Silme işleminden sonra kullanıcıyı bilgilendirmek için ürün listeleme sayfasında düzenleme yapacağız. **liste.php** dosyasını açın ve **// silme mesajı burada yer alacak** açıklama satırının yerine aşağıdaki kodları ekleyin.

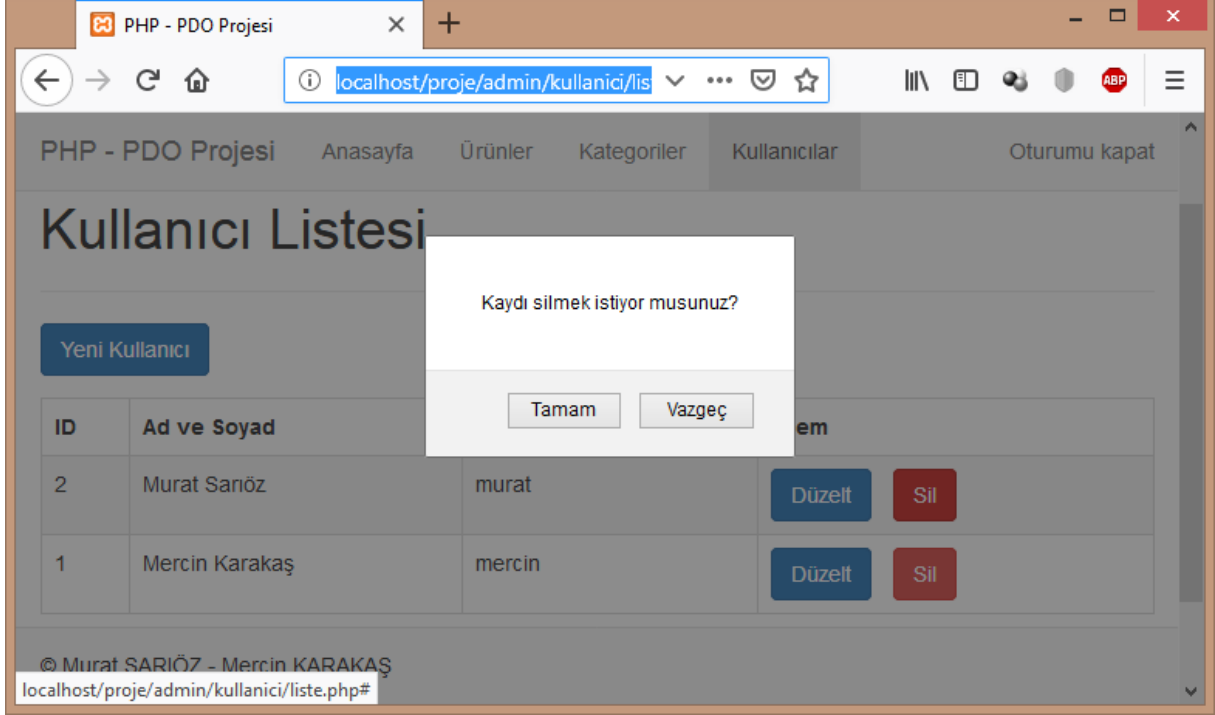
```

$islem = isset($_GET['islem']) ? $_GET['islem'] : "";

// eğer silme (sil.php) sayfasından yönlendirme yapıldıysa
if($islem=='silindi'){
  echo "<div class='alert alert-success'>Kayıt silindi.</div>";
}
else if($islem=='silinemedi'){
  echo "<div class='alert alert-danger'>Kayıt silinemedi.</div>";
}

```

Tarayıcınızın adres satırına **http://localhost/proje/admin/kullanici/liste.php** yazarak kullanıcı listeleme sayfasını görüntüleyin. Herhangi bir kaydı silmek için karşısındaki **Sil** butonuna tıklayın ve silme işlemini **Tamam** butonuyla onaylayın.



KULLANICI EKLEME SAYFASINI OLUŞTURMAK

Yeni kayıt eklemek için kullanılacak **ekle.php** sayfasını **admin/kullanici** klasöründe oluşturun ve aşağıdaki HTML kodlarını içine yazın.

```
<?php include "../header.php"; ?>

<div class="container">

  <div class="page-header">
    <h1>Kullanıcı Ekle</h1>
  </div>

  <!-- PHP kayıt ekleme kodları burada yer alacak -->

  <!-- Kullanıcı eklemek için kullanılacak html formu burada yer alacak -->

</div> <!-- container -->

<?php include "../footer.php"; ?>
```

YENİ KAYIT GİRİŞİ İÇİN GEREKLİ HTML FORMUNU OLUŞTURMAK

Aşağıdaki kodlar veritabanındaki alanlara karşılık gelen bilgilerin girilmesi için gerekli HTML formunu oluşturacaktır. Bu kodları **<!-- Kategori eklemek için kullanılacak html formu burada yer alacak -->** açıklama satırının yerine yazın.

```
<!-- Kategori bilgilerini girmek için kullanılacak html formu -->
<form action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>" method="post">
  <table class='table table-hover table-responsive table-bordered'>
    <tr>
      <td>Ad ve Soyad</td>
      <td><input type='text' name='adsoyad' class='form-control' /></td>
    </tr>
  </table>
```

```

        <td>Kullanıcı adı</td>
        <td><input type='text' name='kadi' class='form-control' /></td>
    </tr>
    <tr>
        <td>Şifre</td>
        <td><input type='password' name='sifre' class='form-control' /></td>
    </tr>
    <tr>
        <td></td>
        <td>
            <input type='submit' value='Kaydet' class='btn btn-primary' />
            <a href='liste.php' class='btn btn-danger'>Kullanıcı listesi</a>
        </td>
    </tr>
</table>
</form>

```

VERİTABANINA YENİ KAYIT EKLEMEK İÇİN GEREKLİ PHP KODLARI

Kullanıcı **ekle.php** sayfasında bilgileri girip **Kaydet** butonuna tıklayınca aşağıdaki PHP kodları MySQL veritabanındaki **kullanici** tablosuna bu bilgileri kaydedecektir. Bu kodları **<!-- PHP kayıt ekleme kodları burada yer alacak -->** açıklama satırının yerine yazın.

```

<?php
if($_POST){

    // veritabanı yapılandırma dosyasını dahil et
    include '../config/vtabani.php';

    try{
        // kayıt ekleme sorgusu
        $sorgu = "INSERT INTO kullanicilar SET adsoyad=:adsoyad, kadi=:kadi,
sifre=:sifre";

        // sorguyu hazırla
        $stmt = $con->prepare($sorgu);

        // post edilen değerler
        $adsoyad=htmlspecialchars(strip_tags($_POST['adsoyad']));
        $kadi=htmlspecialchars(strip_tags($_POST['kadi']));
        $sifre=htmlspecialchars(strip_tags($_POST['sifre']));

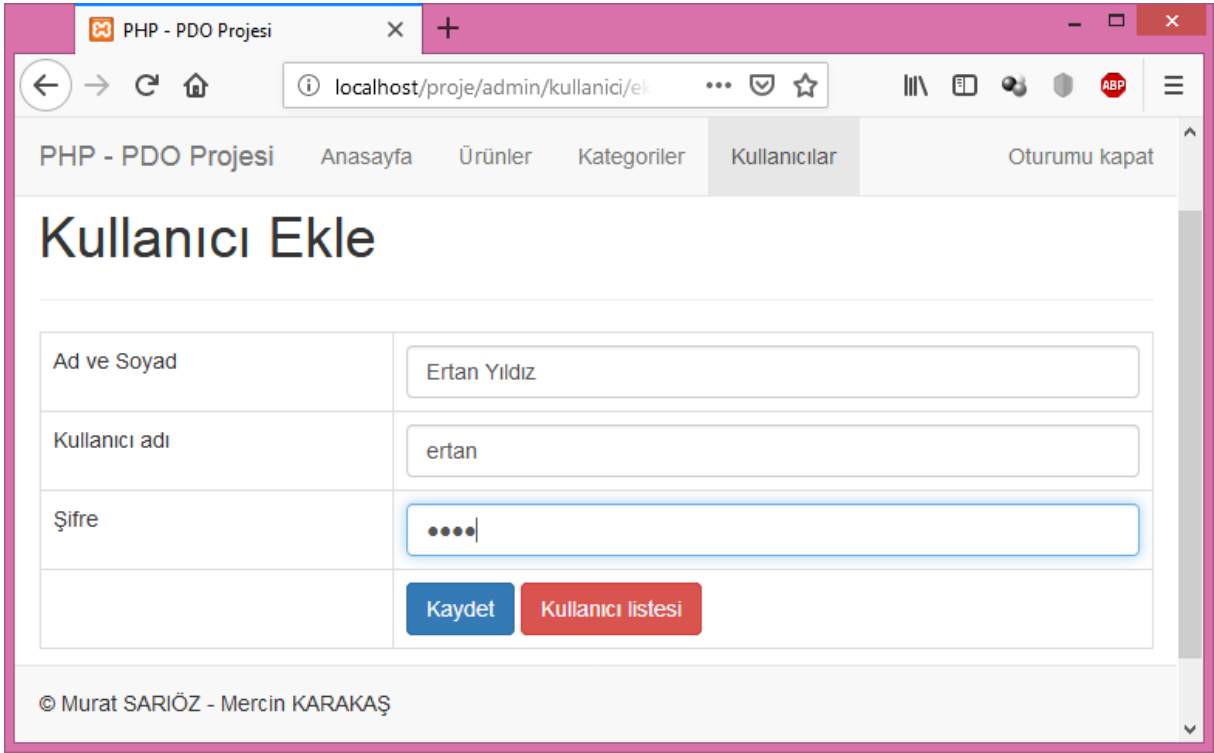
        // parametreleri bağla
        $stmt->bindParam(':adsoyad', $adsoyad);
        $stmt->bindParam(':kadi', $kadi);
        $stmt->bindParam(':sifre', $sifre);

        // sorguyu çalıştır
        if($stmt->execute()){
            echo "<div class='alert alert-success'>Kullanıcı kaydedildi.</div>";
        }else{
            echo "<div class='alert alert-danger'>Kullanıcı kaydedilemedi.</div>";
        }
    }

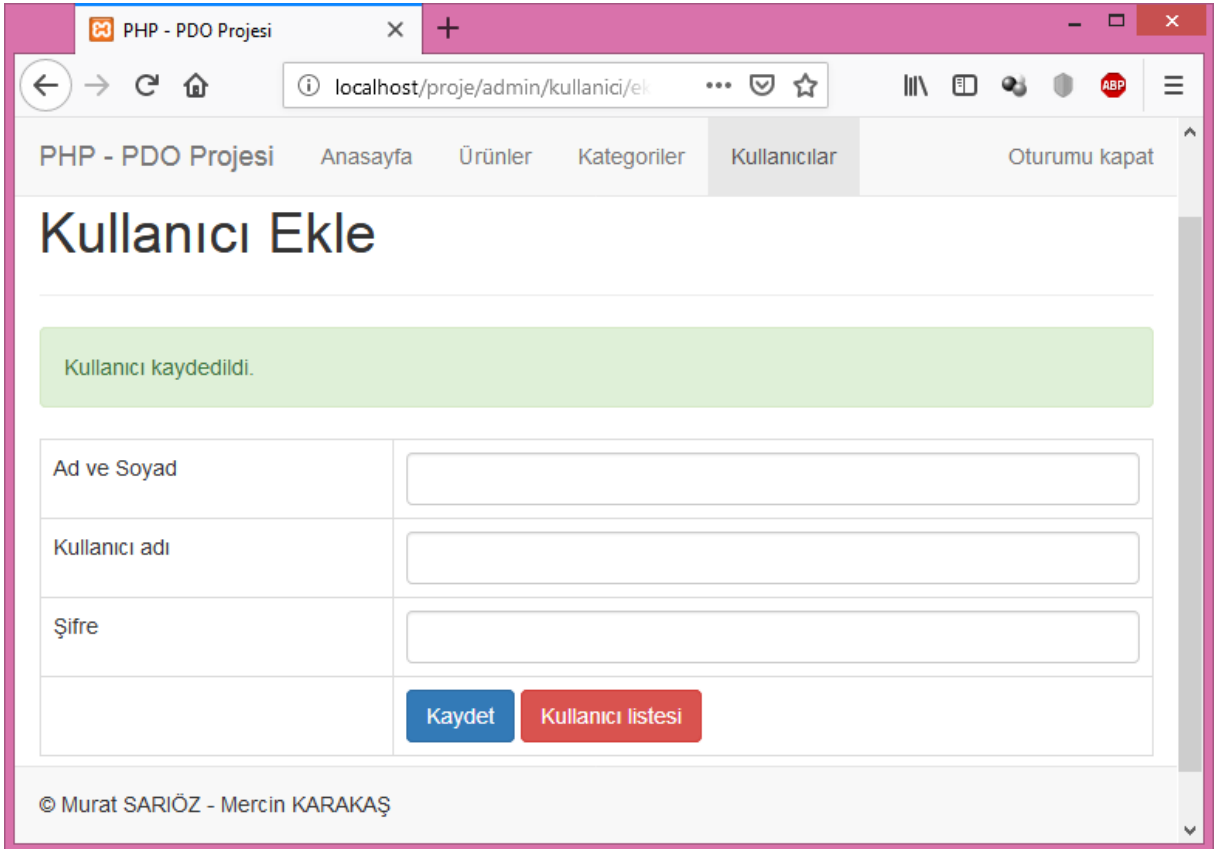
    // hatayı göster
    catch(PDOException $exception){
        die('ERROR: ' . $exception->getMessage());
    }
}
?>

```

Tarayıcınızın adres satırına **http://localhost/proje/admin/kullanici/ekle.php** yazarak kullanıcı ekleme sayfasını görüntüleyin.



Formu doldurun ve **Kaydet** butonuyla yeni bir kayıt ekleyin.



KULLANICI GÜNCELLEME (DÜZELTME) SAYFASINI OLUŞTURMAK

Kullanıcı listeleme sayfasında **Düzeltil** işlem butonu tıklanınca kayıtların bilgilerini bir form ile görüntülemek ve güncellemek için kullanıcıyı yönlendireceğimiz sayfayı **duzelt.php** ismiyle **admin/kullanici** klasöründe oluşturun ve içine aşağıdaki kodları yazın.

```
<?php include "../header.php"; ?>
<div class="container">
```

```

<div class="page-header">
    <h1>Kullanıcı Güncelleme</h1>
</div>

<!-- Kullanıcı bilgilerini getiren PHP kodu burada yer alacak -->

<!-- Kullanıcı bilgilerini düzeltme HTML formu burada yer alacak -->

</div> <!-- container -->

<?php include "../footer.php"; ?>

```

GÜNCELLENECEK KULLANICI BİLGİLERİNİ VERİTABANINDAN OKUMAK

Öncelikle kullanıcı listeleme sayfasından gelen **Id** parametresini **\$_GET['id']** fonksiyonu ile alıp **\$id** değişkenine kaydediyoruz. Daha sonra bu parametreye karşılık gelen kayıt bilgilerini veritabanından okuyup HTML formunu bu bilgilerle dolduruyoruz.

Aşağıdaki kodları **<!-- kategori bilgilerini getiren PHP kodu burada yer alacak -->** açıklama satırının yerine yazın.

```

<?php
// gelen parametre değerini oku, bizim örneğimizde bu Id bilgisidir
$id=isset($_GET['id']) ? $_GET['id'] : die('HATA: Id bilgisi bulunamadı. ');

// veritabanı bağlantı dosyasını dahil et
include '../config/vtabani.php';

// aktif kayıt bilgilerini oku
try {
    // seçme sorgusunu hazırla
    $sorgu = "SELECT id, adsoyad, kadi, sifre FROM kullanicilar WHERE id = ?
LIMIT 0,1";
    $stmt = $con->prepare( $sorgu );

    // id parametresini bağla (? işaretini id değeri ile değiştir)
    $stmt->bindParam(1, $id);

    // sorguyu çalıştır
    $stmt->execute();

    // okunan kayıt bilgilerini bir değişkene kaydet
    $kayit = $stmt->fetch(PDO::FETCH_ASSOC);

    // formu dolduracak değişken bilgileri
    $adsoyad = $kayit['adsoyad'];
    $kadi = $kayit['kadi'];
    $sifre = $kayit['sifre'];
}
// hatayı göster
catch(PDOException $exception){
    die('HATA: ' . $exception->getMessage());
}
?>

```

KAYIT GÜNCELLEMEK İÇİN GEREKLİ HTML FORMUNU OLUŞTURMAK

Aşağıdaki kodlar eski kayıt bilgilerini görüntüleyip yeni bilgilerin girilmesi için gerekli HTML formunu oluşturacaktır. Bu kodları **<!-- kategori bilgilerini düzeltme HTML formu burada yer alacak ->** açıklama satırının yerine yazın.


```

<!-- kaydı güncelleyecek PHP kodu burada yer alacak -->

<!-- kayıt bilgilerini güncelleyebileceğimiz HTML formu -->
<form action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]."?id={$id}");?>"
method="post">
  <table class='table table-hover table-responsive table-bordered'>
    <tr>
      <td>Ad ve Soyad</td>
      <td><input type='text' name='adsoyad' value="<?php echo
htmlspecialchars($adsoyad, ENT_QUOTES); ?>" class='form-control' /></td>
    </tr>
    <tr>
      <td>Kullanıcı adı</td>
      <td><input type='text' name='kadi' value="<?php echo
htmlspecialchars($kadi, ENT_QUOTES); ?>" class='form-control' /></td>
    </tr>
    <tr>
      <td>Şifre</td>
      <td><input type='password' name='sifre' value="<?php echo
htmlspecialchars($sifre, ENT_QUOTES); ?>" class='form-control' /></td>
    </tr>
    <tr>
      <td></td>
      <td>
        <input type='submit' value='Kaydet' class='btn btn-primary' />
        <a href='liste.php' class='btn btn-danger'>Kullanıcı listesi</a>
      </td>
    </tr>
  </table>
</form>

```

KAYIT GÜNCELLEME KODLARI

Kullanıcı formda bilgileri değiştirip **Kaydet** butonuna tıklarsa aşağıdaki kodlar bu değişiklikleri veritabanına kaydedecektir. Bu kodları **<!-- kaydı güncelleyecek PHP kodu burada yer alacak -->** açıklama satırının yerine yazın.

```

<?php
// Kaydet butonu tıklanmışsa
if($_POST){

  try{
    // güncelleme sorgusu
    // çok fazla parametre olduğundan karışmaması için
    // soru işaretleri yerine etiketler kullanacağız
    $sorgu = "UPDATE kullanicilar SET adsoyad=:adsoyad, kadi=:kadi,
sifre=:sifre WHERE id = :id";

    // sorguyu hazırla
    $stmt = $con->prepare($sorgu);

    // gelen bilgileri değişkenlere kaydet
    $adsoyad=htmlspecialchars(strip_tags($_POST['adsoyad']));
    $kadi=htmlspecialchars(strip_tags($_POST['kadi']));
    $sifre=htmlspecialchars(strip_tags($_POST['sifre']));

    // parametreleri bağla
    $stmt->bindParam(':adsoyad', $adsoyad);
    $stmt->bindParam(':kadi', $kadi);
    $stmt->bindParam(':sifre', $sifre);
    $stmt->bindParam(':id', $id);

    // sorguyu çalıştır
    if($stmt->execute()){
      echo "<div class='alert alert-success'>Kayıt güncellendi.</div>";
    }
  }
}

```

```
    }else{
        echo "<div class='alert alert-danger'>Kayıt güncellenemedi.</div>";
    }
}
// hata varsa göster
catch(PDOException $exception){
    die('HATA: ' . $exception->getMessage());
}
}
?>
```

Tarayıcınızın adres satırına **http://localhost/proje/admin/kullanici/liste.php** yazarak herhangi bir kullanıcı için **Düzeltil** butonunu tıklayın ve kullanıcı güncelleme sayfasını görüntüleyin. Bilgilerde değişiklik yaparak **Kaydet** butonunu tıklayın.

PHP - PDO Projesi

localhost/proje/admin/kullanici/di

PHP - PDO Projesi Anasayfa Ürünler Kategoriler **Kullanıcılar** Oturumu kapat

Kullanıcı Güncelleme

Kayıt güncellendi.

Ad ve Soyad	Mercin Karakaş
Ad ve Soyad	mercin
Ad ve Soyad	••••

Kaydet **Kullanıcı listesi**

© Murat SARIÖZ - Mercin KARAKAŞ

OTURUM YÖNETİMİ

Yönetim panelindeki sayfalara erişim sadece yetkili kişilere verilmelidir. Güvenli bir oturum yönetimi için girişte kullanıcıların veritabanından kimlik bilgilerini doğrulamalıyız.

SAYFALARDA OTURUM KONTROLÜNÜ YAPMAK

Yönetim panelindeki sayfalara erişim sağlandığında daha önce oturum açılıp açılmadığını kontrol etmeliyiz. Bu işlemi her sayfada ortak olan başlık sayfasında (**header.php**) yapacağız. **admin** klasöründeki **header.php** dosyasını açın ve HTML kodlarının başına aşağıdaki PHP kodlarını ekleyin.

```
<?php
    session_start();

    if ($_SESSION["loginkey"] == "") {
        // oturum açılmamışsa login.php sayfasına git
        header("Location: /proje/admin/login.php");
    }
?>
```

Not: **admin/urun/sil.php** ve **admin/kategori/sil.php** sayfalarının başına da bu kod bloğunu eklemeyi unutmayın. Aksi halde yetkisiz kişiler adres satırında **id** parametresi belirterek kayıtlarınızı silebilir.

OTURUM AÇMA SAYFASINI (LOGIN.PHP) OLUŞTURMAK

Kullanıcı adı ve şifresini doğrulayacağımız sayfayı **admin** klasörü içinde **login.php** ismiyle oluşturun. Dosyayı açıp içine aşağıdaki HTML kodlarını ekleyin.

```
<!-- Kullanıcı kontrolü için gerekli PHP kodları burada yer alacak -->
<!doctype html>
<html lang="tr">
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <title>PHP - PDO Projesi</title>

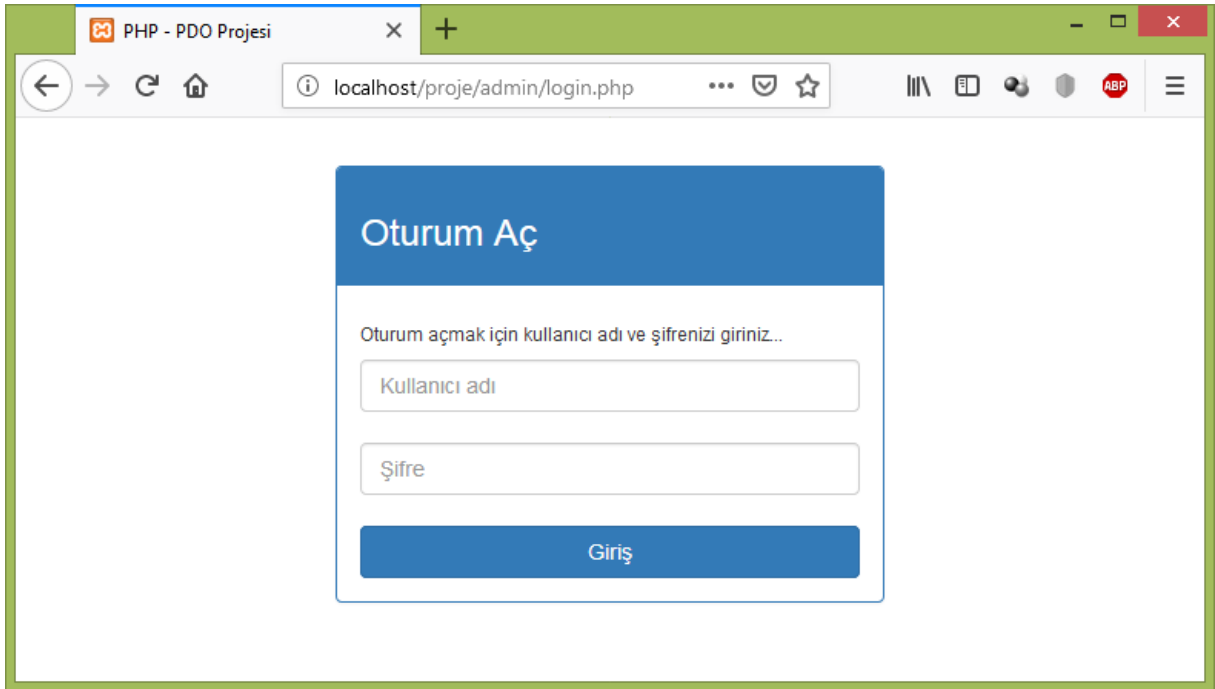
    <!-- Bootstrap CSS dosyası -->
    <link rel="stylesheet" href="/proje/content/css/bootstrap.min.css" />
    <link rel="stylesheet" href="/proje/content/css/style.css" />
</head>
<body>
    <div class="vertical-center">
        <div class="container col-sm-6 col-md-5">
            <form action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>"
method="post" name="loginform">
                <div class="panel panel-primary">
                    <div class="panel-heading"><h3>Oturum Aç</h3></div>
                    <div class="panel-body">
                        <div class="form-horizontal">
                            <h6>Oturum açmak için kullanıcı adı ve şifrenizi
giriniz...</h6>
                            <input type="text" class="form-control"
placeholder="Kullanıcı adı" name="kadi"><br />
                            <input type="password" class="form-control"
placeholder="Şifre" name="ksifre"><br />
                            <button type="submit" class="btn btn-primary btn-
block">Giriş</button>
                        </div>
                    </div>
                </div>
            </form>
        </div>
    </div>
</body>
```

```
</html>
```

Oturum Aç formunun sayfada ortalanması için ilk **<div>** etiketine stil uygulayacağız. Bunun için **content/css** klasöründeki **style.css** dosyasını açın ve içine aşağıdaki satırları ekleyin.

```
.vertical-center {  
    min-height: 100%;  
    min-height: 100vh;  
    display: flex;  
    align-items: center;  
}
```

Tarayıcınızın adres satırına **http://localhost/proje/admin/index.php** yazarak oturum açma sayfasının açılmasını sağlayın. Daha önce oturum açılmadığı için **index.php** sayfası yüklenirken **header.php** içindeki oturum kontrolü çalışacak ve kullanıcıyı **login.php** sayfasına yönlendirecektir.



KULLANICI KONTROLÜ VE OTURUM AÇMA İŞLEMİ

login.php sayfasındaki `<!-- Kullanıcı kontrolü için gerekli PHP kodları burada yer alacak -->` açıklama satırını aşağıdaki PHP kodları ile değiştirin.

```
<?php  
    session_start();  
  
    // oturumu sonlandırmak için gerekli PHP kodları burada yer alacak  
  
    // kullanıcı kontrolü  
    if ($_POST){  
        $kadi = $_POST["kadi"];  
        $ksifre = $_POST["ksifre"];  
  
        if (isset($kadi) && isset($ksifre)) {  
            include "../config/vtabani.php";  
            try {  
                $sorgu = "SELECT kadi,sifre FROM kullanicilar WHERE kadi=:kadi AND  
sifre=:ksifre";  
  
                $stmt = $con->prepare($sorgu);  
  
                // parametreleri bağla  
                $stmt->bindParam(":kadi", $kadi);
```

```

        $stmt->bindParam(":ksifre", $ksifre);

        // sorguyu çalıştır
        $stmt->execute();
        $sayi = $stmt->rowCount();

        if ($sayi == 1) {
            $_SESSION["loginkey"] = $kadi; // oturum değişkenini oluştur
            header("Location: index.php");
        }
    }
    // hatayı göster
    catch(PDOException $exception){
        die('HATA: ' . $exception->getMessage());
    }
}
}
?>

```

OTURUMU KAPATMAK

login.php sayfasındaki // oturumu sonlandırmak için gerekli PHP kodları burada yer alacak açıklama satırını aşağıdaki PHP kodları ile değiştirin.

```

// oturum sonlandırma kontrolü
if ($_GET) {
    $cikis = $_GET["cikis"];
    if ($cikis == 1) {
        session_destroy(); // oturumu sonlandır
        unset($_SESSION["loginkey"]); // oturum değişkenini sıfırla
        header("Location: login.php");
    }
}
}

```

Ardından **header.php** dosyasını açın ve Bootstrap Navbar menü kodlarında **Oturumu kapat** menü seçeneğini aşağıdaki şekilde düzenleyin.

Eski kod:

```
<li><a href="#">Oturumu kapat</a></li>
```

Yeni kod:

```
<li><a href="/proje/admin/login.php?cikis=1">Oturumu kapat</a></li>
```

PHP - PDO Projesi

Anasayfa Ürünler Kategoriler Kullanıcılar Oturumu kapat

PHP - PDO Projesi

Bu belge ile PHP ve MySQL kullanarak basit bir CRUD (Create, Read, Update, Delete) uygulaması geliştirme aşamaları öğretilmesi amaçlanmaktadır. Uygulama geliştirme aşamasında daha yeni bir programlama yöntemi olan PDO (PHP Data Objects) yapısı kullanılacaktır.

© Murat SARIÖZ - Mercin KARAKAŞ
localhost/proje/admin/login.php?cikis=1

EK GÖREVLER

- Kategori sayfalarındaki butonlara **Glyphicon** ekleyin.
- Ürün düzenleme sayfasında ürün resminin de değıştirilmesini sağlayın.
- Kategori listeleme sayfasında kayıtlar için sayfalandırma (pagination) yapın.
- Kategori listeleme sayfasında kayıtlar için arama işlevi ekleyin.
- Ürün detay sayfasında ürünün giriş ve düzeltme tarihlerinin de görüntülenmesini sağlayın.
- Kategori ürünlerini listeleme sayfasında kayıtlar için sayfalandırma (pagination) yapın.
- Ürün ekleme ve düzeltme sayfalarında **Açıklama (acıklama)** alanı için **CKEditor** desteęi ekleyin.
- Bütün uyarı ve onay iletilerini **SweetAlert** eklentisini kullanarak yeniden düzenleyin.

PHP- PDO PROJESİ

(Kullanıcı Arayüzü)

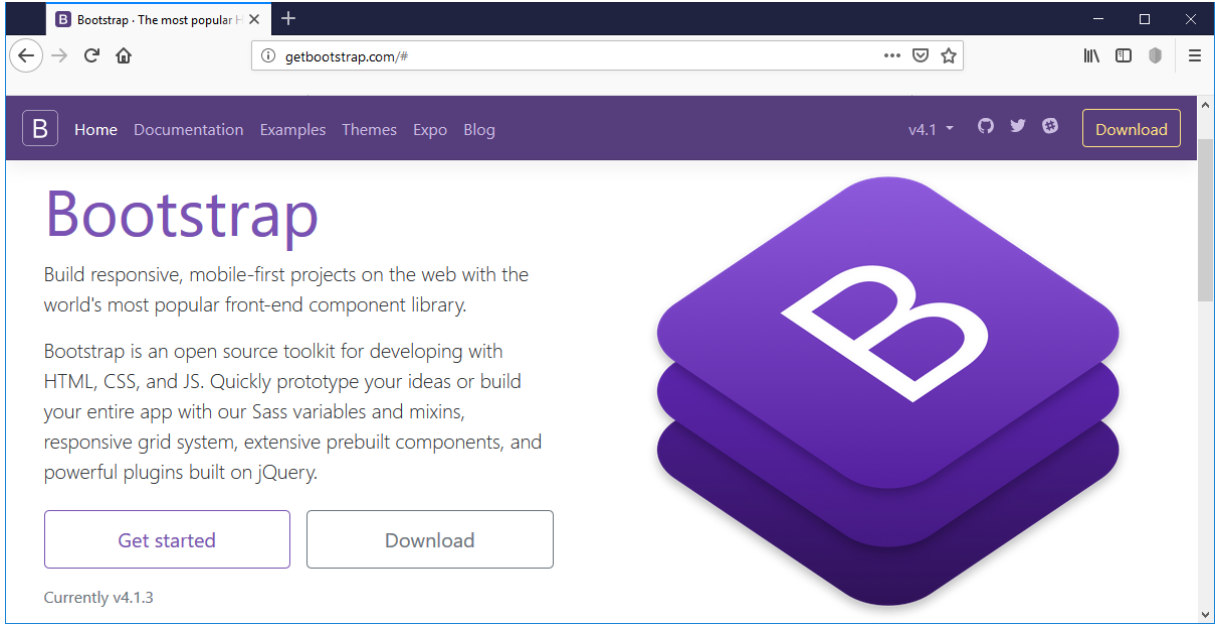
ADIM ADIM DİNAMİK SİTE TASARLAMA

Murat SARIÖZ – Mercin KARAKAŞ

BÜYÜKÇEKMECE SUDİ ÖZKAN MESLEKİ VE TEKNİK ANADOLU LİSESİ

KULLANICI ARAYÜZÜNÜ OLUŞTURMAK

Yönetici arayüzünü tasarlarken **Bootstrap** kütüphanesinin **3.3.7** sürümünü kullanmıştık. Web sitemizin kullanıcı arayüzünde **Bootstrap 4.1.x** sürümünü kullanacağız. Bunun için Bootstrap sitesinden gerekli dosyaların bağlantılarını kopyalamalıyız. <http://getbootstrap.com/> adresine bağlanın.



Get started butonunu tıklayıp **Starter Template** başlığı altındaki HTML kodları kopyalayın. **Proje** klasörü içinde **index.php** isminde bir dosya oluşturun ve kopyaladığınız kodları bu dosyanın içine yapıştırın. Kodları aşağıdaki gibi yeniden düzenleyin.

```
<!doctype html>
<html lang="tr">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <title>PHP - PDO Projesi</title>

    <!-- Bootstrap CSS -->
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css">
    <!-- Simge kütüphanesi -->
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
    <!-- Bizim stil dosyamız -->
    <link rel="stylesheet" type="text/css" href="content/css/style.css">

    <!-- İlk önce jQuery, sonra Popper.js, sonra da Bootstrap JS -->
    <script type="text/javascript" src="content/js/jquery-3.3.1.min.js"></script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"></script>
    <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"></script>
  </head>
  <body>
    <!-- Sayfa içeriği burada yer alacak -->
  </body>
</html>
```

Bu aşamadan sonra tasarlayacağımız web sitesinin anasayfa taslağını belirlememiz lazım. Sitemizin anasayfa taslağı aşağıdaki gibi olacaktır.

Duyuru satırı

Kampanya: 100% üzeri alışverişlerinizde ücretsiz kargo Sepetiniz 4

İstediğin Burada [Anasayfa](#) [Ürünler](#) [Kategoriler](#) [Hakkımızda](#)

Logo, arama ve menü satırı

Slider satırı

Bayan Çanta Koleksiyonumuz
Yeni koleksiyon bayan çantalarında %50'ye varan indirimler

Karşılama satırı **İstediğin Burada'ya Hoş Geldiniz**

İstediğin Burada olarak ilk önceliğimiz olan "müşteri memnuniyeti" ve ardından gelen kalite anlayışımızla, satışlarımızı hız kesmeden sürdürmeye devam ederken, sektörün ihtiyaçları doğrultusunda, kaliteli ürün tedarik etmek ve ürünlerimizden daha uzun yıllar faydalanmanız için çalışıyoruz. Türkiye'de değer kazanan bir marka olarak, uluslararası bir marka olma yolunda hızla ilerliyoruz. Kuruluşumuzun 10. yılını kutladığımız bu günlerde, heyecanı ve mutluluğu yaşıyoruz.



Sorunsuz İade
Bütün ürünlerde 7 gün içinde ücretsiz iade



Güvenli Ödeme
Ödemelerinizi tüm banka ve kredi kartlarıyla yapabilirsiniz



Ücretsiz Kargo
100% üzeri alışverişlerinizde yurt içi ücretsiz kargo

Yeni ürünler satırı

Yeni Ürünler

Fırsat satırı

Card title
Some quick example text to build on the card title and make up the bulk of the card's content.

55.00 ₺

Card title
Some quick example text to build on the card title and make up the bulk of the card's content.

55.00 ₺

Card title
Some quick example text to build on the card title and make up the bulk of the card's content.

55.00 ₺

Yardım

Ödeme
İade ve Değişim
Mağazalarımız
Kariyer Fırsatları
Sık Sorulan Sorular
Kampanyalar

Altbilgi satırı

Fırsat Ürünleri



Ürün Adı
55.00 ₺



Ürün Adı
55.00 ₺



Ürün Adı
55.00 ₺

İletişim

Adres:
Celaliye Mahallesi, Cumhuriyet Cd. No:26
Büyükkçekmece/İstanbul
Telefon: (0212) 885 98 82
Faks: (0212) 885 40 30
Web: <http://sudiozkanmtal.meb.k12.tr>
Eposta: 7-24@sudiozkanmtal.meb.k12.tr

WEB SAYFALARININ ANA HTML ÇATISINI OLUŞTURMAK

CSS STİL DOSYASINI (STYLE.CSS) DÜZENLEMEK

Web sayfalarında etiketlere CSS stil sınıfları tanımlamak için oluşturduğunuz **style.css** dosyasını açın ve içine aşağıdaki kodları ekleyin.

```
@import url('https://fonts.googleapis.com/css?family=Quicksand');

body.admin{
  min-height: 500px;
  padding-top: 50px;
  padding-bottom: 50px;
}

.m-t-1em { margin-top:1em; }
.m-r-1em { margin-right:1em; }
.m-b-1em { margin-bottom:1em; }
.m-l-1em { margin-left:1em; }
.mt0 { margin-top:0; }

select.no-arrow{
  -moz-appearance: none;
  -webkit-appearance: none;
  appearance: none;
}

.vertical-center {
  min-height: 100%;
  min-height: 100vh;
  display: flex;
  align-items: center;
}

.link1{
  text-decoration: none;
  color: white;
}

.link1:hover{
  text-decoration: none;
  color: yellow;
}

p.firsat{
  margin: 0;
  padding: 0;
}

.baslik{
  font-family: 'Quicksand', sans-serif;
}
```

Web sayfalarında tekrar eden ortak başlık (header) ve altbilgi (footer) alanlarını ayrı sayfalar halinde oluşturacağız. Bu hem kod karmaşasını azaltacak hem de başlık ve altbilgi değişikliklerini daha esnek hale getirecektir.

BAŞLIK (HEADER.PHP) SAYFASINI OLUŞTURMAK

Sayfaların başında duyuru satırı ve bir menü nesnemiz olacak. Menü için **Bootstrap** kütüphanesinin **NavBar** bileşenini kullanacağız. Menü kodları ve üst kısmındaki HTML kodlar ortak olacağından hepsini **header.php** dosyasında toplayacağız. Öncelikle **index.php** dosyasında **<body>** etiketine kadar olan kodları alıp (**<body>** etiketi dahil) **proje** klasöründe oluşturduğunuz **header.php** dosyasına taşıyın.

```

<!doctype html>
<html lang="tr">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <title>PHP - PDO Projesi</title>

    <!-- Bootstrap CSS -->
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css">
    <!-- Simge kütüphanesi -->
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
    <!-- Bizim stil dosyamız -->
    <link rel="stylesheet" type="text/css" href="content/css/style.css">

    <!-- İlk önce jQuery, sonra Popper.js, sonra da Bootstrap JS -->
    <script type="text/javascript" src="content/js/jquery-3.3.1.min.js"></script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"></script>
    <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"></script>
  </head>
  <body>
    <!-- Duyuru satırı kodları burada yer alacak -->

    <!-- Logo, arama ve menü satırı kodları burada yer alacak -->

```

DUYURU SATIRINI OLUŞTURMAK

header.php dosyasını açın ve `<!-- Duyuru satırı kodları burada yer alacak -->` yazan açıklama satırının yerine aşağıdaki kodları yazın.

```

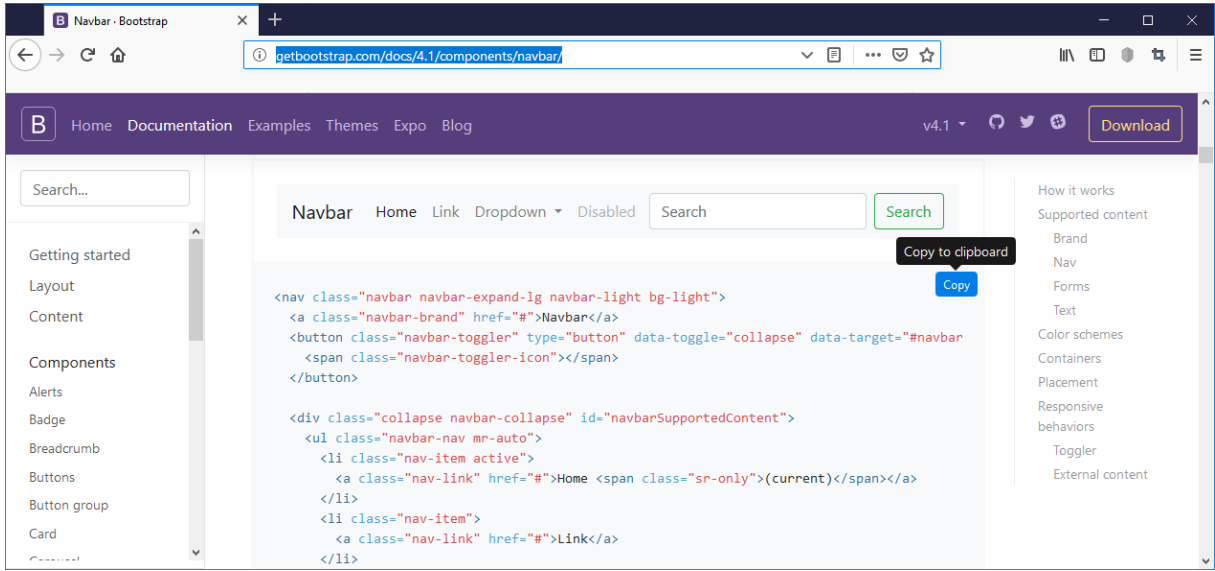
<div class="container-fluid bg-success">
  <div class="container">
    <div class="row p-2">
      <div class="col-md-8">
        <span class="text-white">Kampanya: 100&#8378; üzeri alışverişlerinizde ücretsiz kargo</span>
      </div>
      <div class="col-md-4 text-right text-white">
        <a class="link1" href="sepetim.php"> <!-- Sepet içeriği sayfası linki -->
          <span class="fa-stack fa-1x">
            <i class="fa fa-circle-thin fa-stack-2x"></i>
            <i class="fa fa-shopping-cart fa-stack-1x"></i>
          </span> Sepetiniz
          <span class="badge badge-light">4 <!-- Sepetteki ürün sayısı --> </span>
        </a>
      </div>
    </div>
  </div>
</div>

```

Not: Sepet işlemleriyle ilgili sayfanın tasarımı ve kodları daha sonra ayrı bir başlık altında anlatılacak.

MENÜ SATIRINI OLUŞTURMAK

<http://getbootstrap.com/docs/4.1/components/navbar/> adresinden **NavBar** menü bileşenine ait örnek kodları kopyalayın ve `<!-- Logo, arama ve menü satırı kodları burada yer alacak -->` yazan açıklama satırının yerine yapıştırıp aşağıdaki gibi düzenleyin.



```

<div class="container">
  <nav class="navbar navbar-expand-lg navbar-light bg-white">
    <a class="navbar-brand" href="index.php">
      <!-- Site logosu -->
      
    </a>
    <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-
expanded="false" aria-label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarSupportedContent">
      <!-- Ürün arama formu -->
      <form class="form-inline my-2 my-lg-0" action="urunler.php" method="get"
name="form_ara">
        <input class="form-control mr-sm-2" type="search" placeholder="Ürün
ara..." aria-label="Ara" name="aranan">
        <button class="btn btn-outline-success my-2 my-sm-0"
type="submit">Ara</button>
      </form>
      <ul class="navbar-nav ml-auto">
        <li class="nav-item active">
          <a class="nav-link" href="index.php">Anasayfa <span class="sr-
only">(current)</span></a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="urunler.php">Ürünler</a>
        </li>
        <li class="nav-item dropdown">
          <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown"
role="button" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
            Kategoriler
          </a>
          <div class="dropdown-menu" aria-labelledby="navbarDropdown">
            <!-- Burada katogorileri listeleyen PHP kodları yer alacak -->
          </div>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="hakkimizda.php">Hakkımızda</a>
        </li>
      </ul>
    </div>
  </nav>
</div>

```

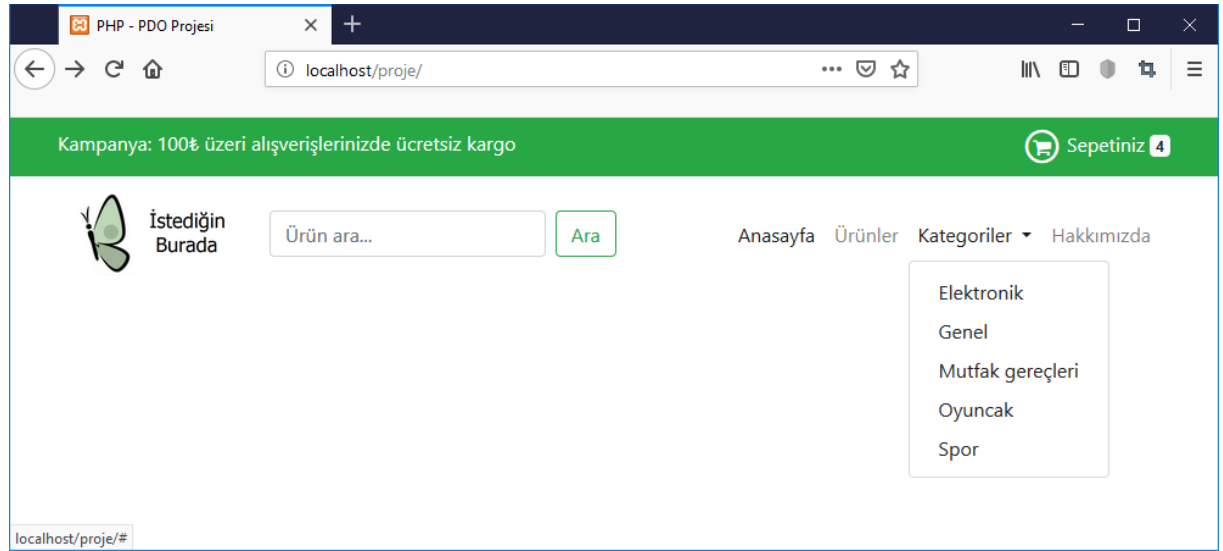
Not: Siteniz için oluşturduğunuz logo dosyasını **content/images** klasörüne **logo.gif** ismiyle kaydedin.

KATEGORİ MENÜSÜNDEKİ ALT SEÇENEKLERİ OLUŞTURMAK

Ardından `<!-- Burada kategorileri listeleyen PHP kodları yer alacak -->` açıklama satırının yerine aşağıdaki PHP kodlarını yazın.

```
<?php
// veritabanı yapılandırma dosyasını dahil et
include 'config/vtabani.php';
// kayıt listeleme sorgusu
$sorgu='select id, kategoriadi from kategoriler order by kategoriadi';
$stmt = $con->prepare($sorgu); // sorguyu hazırla
$stmt->execute(); // sorguyu çalıştır
$veri = $stmt->fetchAll(PDO::FETCH_ASSOC); // tablo verilerini oku
foreach ($veri as $kayit) { // her kategori için bir menü seçeneği oluştur
    echo "<a class='dropdown-item'
href='urunler.php?id={$kayit[\"id\"]}'>{$kayit[\"kategoriadi\"]}</a>";
}
?>
```

Tarayıcınızın adres satırına <http://localhost/proje/index.php> yazarak sayfayı görüntüleyin.



ALTBİLGİ (FOOTER.PHP) SAYFASINI OLUŞTURMAK

Sayfaların sonunda bir altbilgi alanımız olacak. Bunun için index.html dosyasında `</body>` etiketinden itibaren olan HTML kodlarını **footer.php** dosyasına taşıyın ve dosyayı **proje** klasörüne kaydedin.

```
<!-- Altbilgi satırı kodları burada yer alacak -->
</body>
</html>
```

Daha sonra `<!-- Altbilgi satırı kodları burada yer alacak -->` açıklama satırının yerine aşağıdaki HTML kodlarını yazın.

```
<div class="container-fluid bg-dark">
  <div class="container pt-4">
    <div class="row">
      <div class="col-md-4">
        <h4 class="text-white">Yardım</h4>
        <a href="#" class="link1">Ödeme</a><br>
        <a href="#" class="link1">İade ve Değişim</a><br>
        <a href="#" class="link1">Mağazalarımız</a><br>
        <a href="#" class="link1">Kariyer Fırsatları</a><br>
        <a href="#" class="link1">Sık Sorulan Sorular</a><br>
        <a href="#" class="link1">Kampanyalar</a>
      </div>
      <div class="col-md-4">
        <h4 class="text-white">Fırsat Ürünleri</h4>
        <!-- Fırsat ürünlerini listeleyen PHP kodları burada yer alacak -->
      </div>
    </div>
  </div>
</div>
```

```

    </div>
    <div class="col-md-4">
      <h4 class="text-white">İletişim</h4>
      <p class="text-light">Adres: <br>Celaliye Mahallesi, Cumhuriyet Cd. No:26
<br>Büyükcemece/İstanbul <br>Telefon: (0212) 885 98 82 <br>Faks: (0212) 885 40 30
<br>Web: http://sudiozkanmtal.meb.k12.tr <br>Eposta: 7-
24@sudiozkanmtal.meb.k12.tr</p>
    </div>
  </div>
</div>
</div>
<!-- Kullanıcı tanımlı JavaScript kodları burada yer alacak -->
</body>
</html>

```

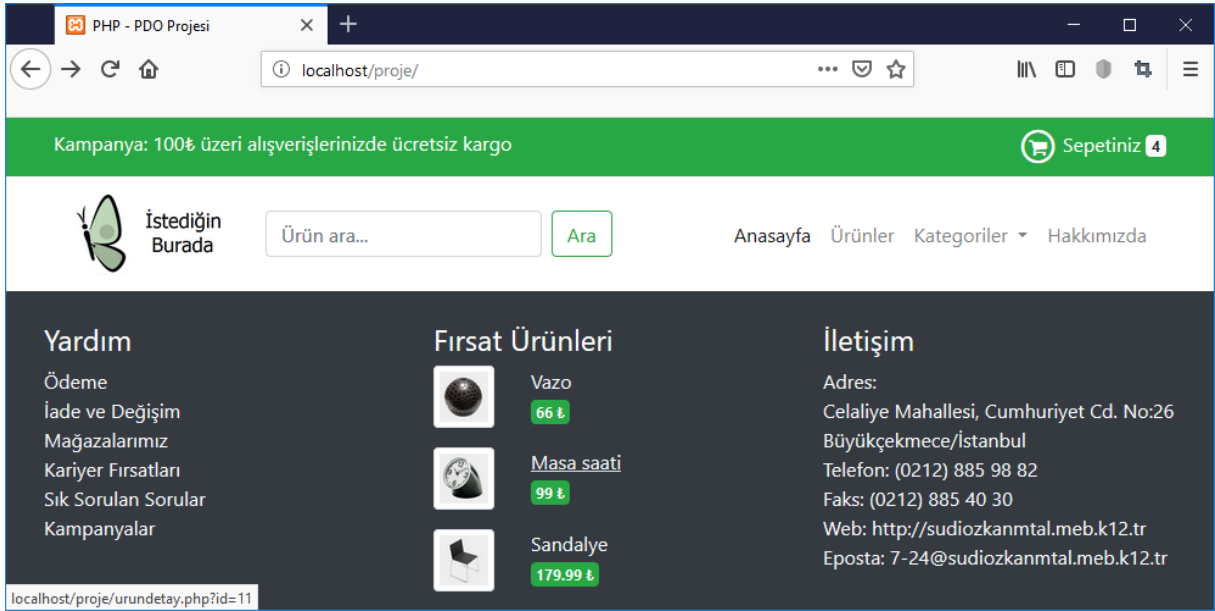
Görüldüğü gibi altbilgi alanı üç bölümden oluşmaktadır. Orta bölümde en düşük fiyatlı üç fırsat ürünü listelenecek. Bunun için <!-- Fırsat ürünlerini listeleyen PHP kodları burada yer alacak --> açıklama satırının yerine aşağıdaki PHP kodlarını yazın.

```

<?php
// veritabanı yapılandırma dosyasını dahil et
include 'config/vtabani.php';
// kayıt listeleme sorgusu
$sorgu='select id, urunadi, aciklama, fiyat, resim from urunler order by fiyat
limit 0,3';
$stmt = $con->prepare($sorgu); // sorguyu hazırla
$stmt->execute(); // sorguyu çalıştır
$veri = $stmt->fetchAll(PDO::FETCH_ASSOC); // tablo verilerini oku
foreach ($veri as $kayit) { ?>
  <div class="row mb-3">
    <div class="col-md-3">
      
    </div>
    <div class="col-md-9">
      <a href="urundetay.php?id=<?php echo $kayit['id']?>" class="link1">
        <p class="firsat"><?php echo $kayit['urunadi']?></p>
      </a>
      <span class="badge badge-success p-1"><?php echo $kayit['fiyat']?>
&#8378;</span>
    </div>
  </div>
<?php }
?>

```

Tarayıcınızın adres satırına **http://localhost/proje/index.php** yazarak anasayfayı görüntüleyin.



ANASAYFAYI (INDEX.PHP) OLUŞTURMAK

Web sayfalarımız için ortak olan başlık ve altbilgi alanlarını oluşturduk. Bundan sonra oluşturacağımız sayfalarda bu bilgileri ekleyerek işe başlayacağız. **proje** klasörü içindeki **index.php** isimli dosyayı açın ve `<!-- Sayfa içeriği burada yer alacak -->` yazan açıklama satırının yerine aşağıdaki kodları yazın.

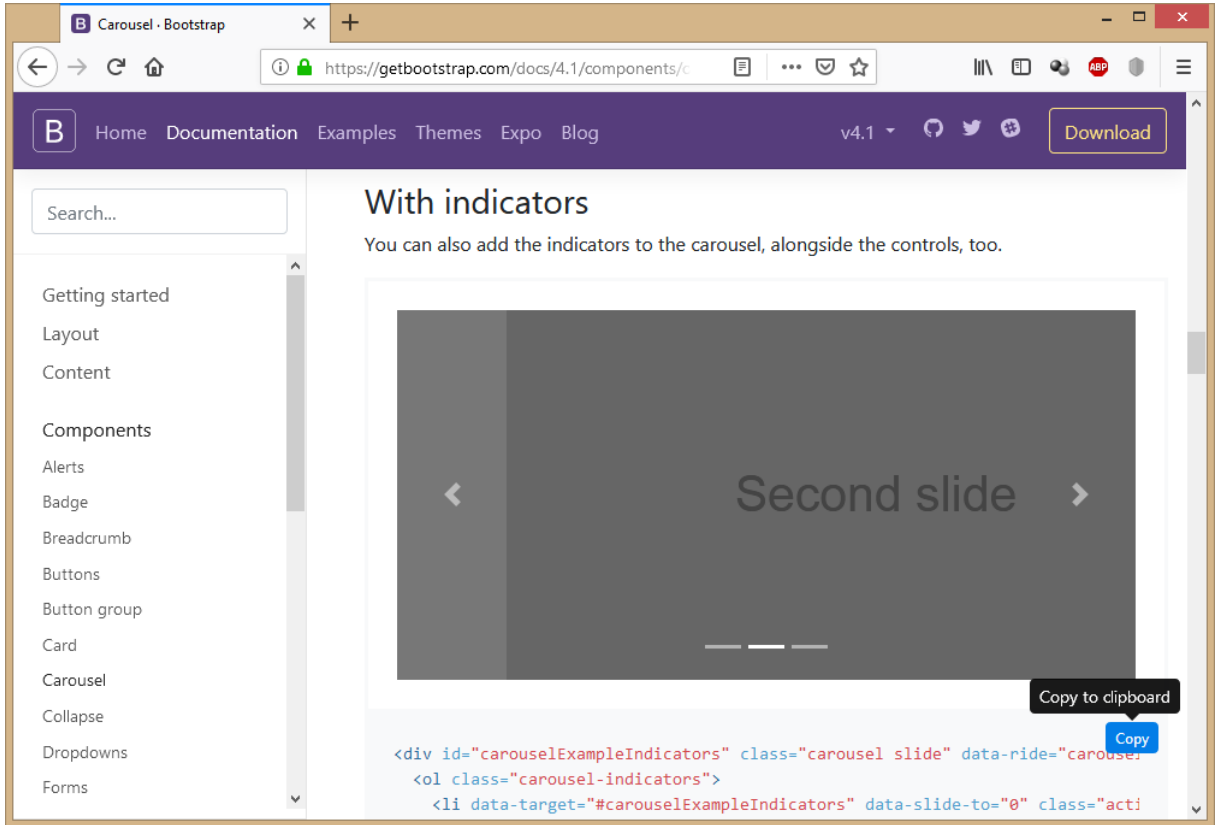
```
<?php include "header.php"; ?>

<!-- Slider satırı kodları burada yer alacak -->
<div class="container pt-4">
  <!-- Karşılama satırı kodları burada yer alacak -->
  <!-- Fırsat satırı kodları burada yer alacak -->
  <!-- Yeni ürün satırı kodları burada yer alacak -->
</div>

<?php include "footer.php"; ?>
```

SLIDER SATIRINI OLUŞTURMAK

Slider olarak **Bootstrap** kütüphanesinden **Carousel** bileşenini kullanacağız. <https://getbootstrap.com/docs/4.1/components/carousel/> adresinden örnek kodları kopyalayın ve **index.php** dosyasında `<!-- Slider satırı kodları burada yer alacak -->` açıklama satırının yerine yapıştırın.



Carousel kodlarını aşağıdaki gibi düzenleyin.

```

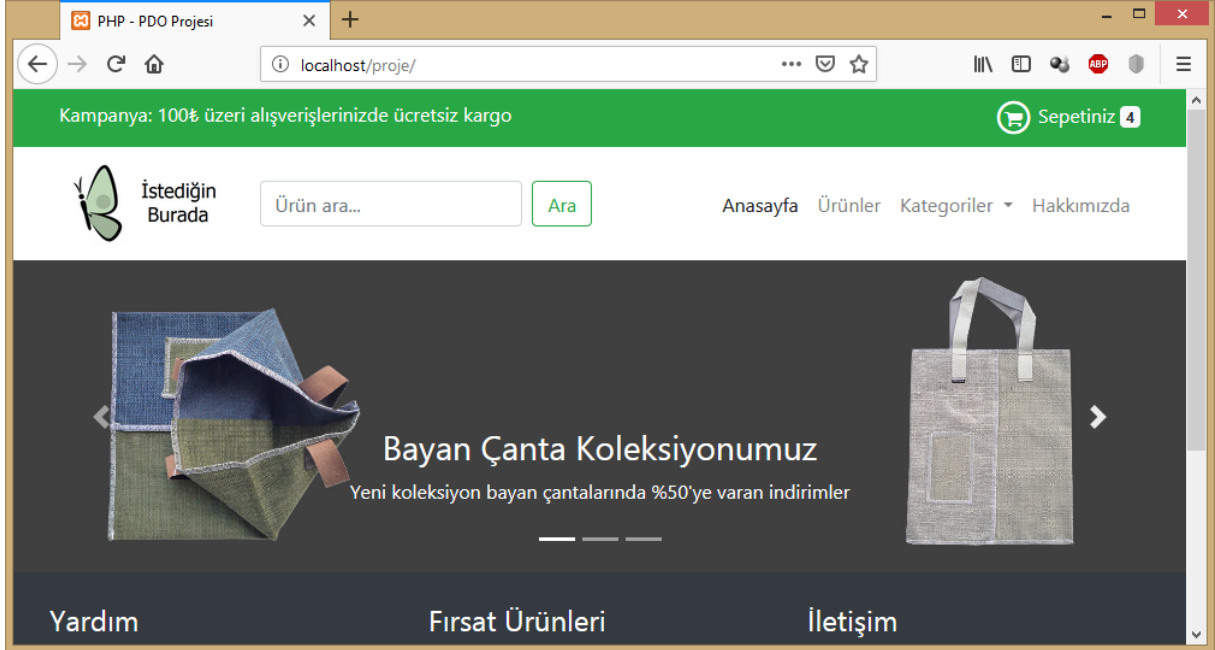
<div id="carouselExampleIndicators" class="carousel slide d-none d-sm-block" data-ride="carousel">
  <ol class="carousel-indicators">
    <li data-target="#carouselExampleIndicators" data-slide-to="0"
class="active"></li>
    <li data-target="#carouselExampleIndicators" data-slide-to="1"></li>
    <li data-target="#carouselExampleIndicators" data-slide-to="2"></li>
  </ol>
  <div class="carousel-inner">
    <div class="carousel-item active">
      
      <div class="carousel-caption d-none d-md-block">
        <h3>Bayan Çanta Koleksiyonumuz</h3>
        <p>Yeni koleksiyon bayan çantalarında %50'ye varan indirimler</p>
      </div>
    </div>
    <div class="carousel-item">
      
      <div class="carousel-caption d-none d-md-block text-warning text-right">
        <h3>Saat Koleksiyonumuz</h3>
        <p>Yeni ve modern duvar saatlerinde %20'ye varan indirimler</p>
      </div>
    </div>
    <div class="carousel-item">
      
      <div class="carousel-caption d-none d-md-block text-dark text-left">
        <h3>Lamba Koleksiyonumuz</h3>
        <p>En iyi lambalar en iyi fiyata</p>
      </div>
    </div>
  </div>
  <a class="carousel-control-prev" href="#carouselExampleIndicators" role="button"
data-slide="prev">
    <span class="carousel-control-prev-icon" aria-hidden="true"></span>
    <span class="sr-only">Önceki</span>
  </a>

```

```
<a class="carousel-control-next" href="#carouselExampleIndicators"
role="button" data-slide="next">
  <span class="carousel-control-next-icon" aria-hidden="true"></span>
  <span class="sr-only">Sonraki</span>
</a>
</div>
```

Not: Siteniz için oluşturduğunuz slayt dosyalarını **content/images** klasörüne **slide1.jpg**, **slide2.jpg**, **slide3.jpg** isimleriyle kaydedin.

Tarayıcınızın adres satırına **http://localhost/proje/index.php** yazarak anasayfayı görüntüleyin.



KARŞILAMA SATIRINI OLUŞTURMAK

index.php dosyasını açın ve `<!-- Karşılama satırı kodları burada yer alacak -->` açıklama satırının yerine aşağıdaki kodları yazın.

```
<h1 class="text-center baslik">İstediğin Burada'ya Hoş Geldiniz</h1>
<p class="text-center pt-4">İstediğin Burada olarak ilk önceliğimiz olan "müşteri memnuniyeti" ve ardından gelen kalite anlayışımızla, satışlarımızı hız kesmeden sürdürmeye devam ederken, sektörün ihtiyaçları doğrultusunda, kaliteli ürün tedarik etmek ve ürünlerimizden daha uzun yıllar faydalanmanız için çalışıyoruz. Türkiye'de değer kazanan bir marka olarak, uluslararası bir marka olma yolunda hızla ilerliyoruz. Kuruluşumuzun 10. yılını kutladığımız bu günlerde, heyecanı ve mutluluğu yaşıyoruz.</p>
```

FIRSAT SATIRINI OLUŞTURMAK

index.php dosyasını açın ve `<!-- Fırsat satırı kodları burada yer alacak -->` açıklama satırının yerine aşağıdaki kodları yazın.

```
<div class="row pt-4 bg-light">
  <div class="col-md-4">
    <div class="row">
      <div class="col-md-4">
        <span class="fa-stack fa-3x">
          <i class="fa fa-circle-thin fa-stack-2x"></i>
          <i class="fa fa-refresh fa-stack-1x"></i>
        </span>
      </div>
    </div>
  </div>
```

```

<div class="col-md-8">
  <h5>Sorunsuz İade</h5>
  <p>Bütün ürünlerde 7 gün içinde ücretsiz iade</p>
</div>
</div>
<div class="col-md-4">
  <div class="row">
    <div class="col-md-4">
      <span class="fa-stack fa-3x">
        <i class="fa fa-circle-thin fa-stack-2x"></i>
        <i class="fa fa-credit-card fa-stack-1x"></i>
      </span>
    </div>
    <div class="col-md-8">
      <h5>Güvenli Ödeme</h5>
      <p>Ödemelerinizi tüm banka ve kredi kartlarıyla yapabilirsiniz</p>
    </div>
  </div>
</div>
<div class="col-md-4">
  <div class="row">
    <div class="col-md-4">
      <span class="fa-stack fa-3x">
        <i class="fa fa-circle-thin fa-stack-2x"></i>
        <i class="fa fa-truck fa-stack-1x"></i>
      </span>
    </div>
    <div class="col-md-8">
      <h5>Ücretsiz Kargo</h5>
      <p>100₺#8378; üzeri alışverişlerinizde yurt içi ücretsiz kargo</p>
    </div>
  </div>
</div>
</div>
</div>

```

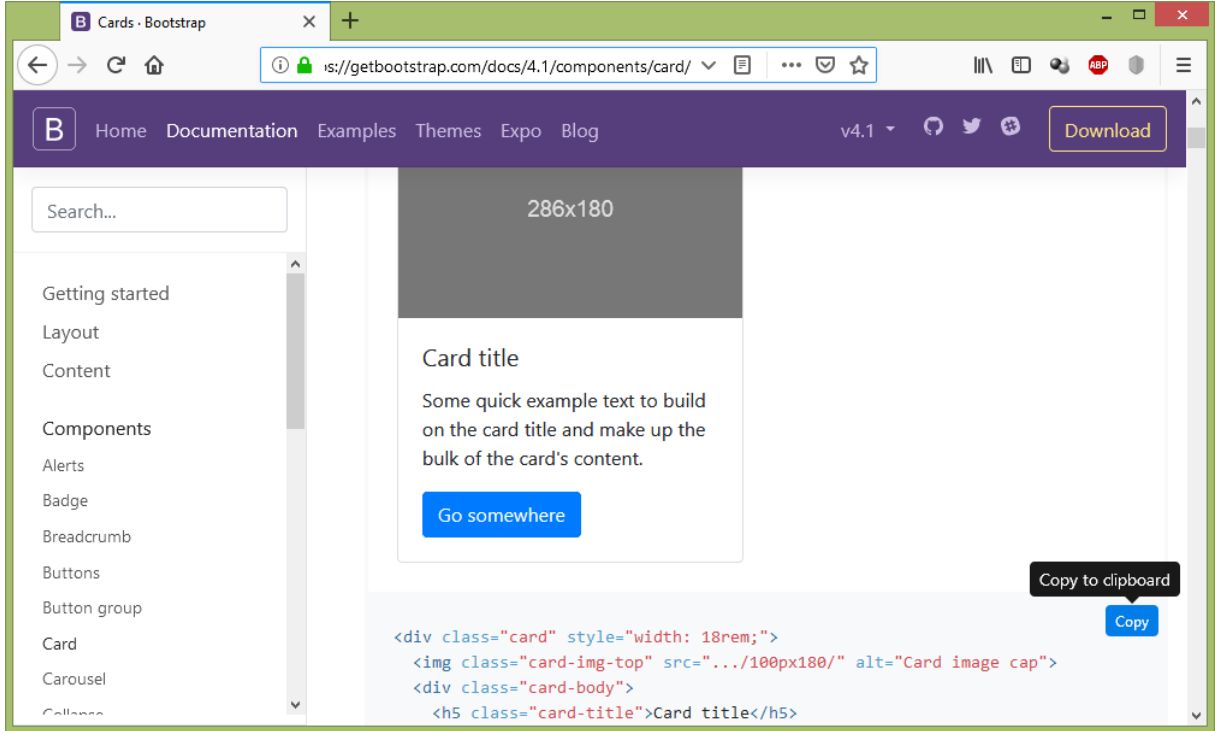
Not: Font Awesome kütüphanesindeki simgeleri kullanabilmek için **header.php** dosyasında simge bağlantısını önceden tanımladımıza dikkat edin (`<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">`).

Tarayıcınızın adres satırına **http://localhost/proje/index.php** yazarak anasayfayı görüntüleyin.



YENİ ÜRÜN SATIRINI OLUŞTURMAK

Ürün listesi için **Bootstrap** kütüphanesinden **Card** bileşenini kullanacağız. <https://getbootstrap.com/docs/4.1/components/card/> adresinden örnek kodları kopyalayın ve **index.php** dosyasında `<!-- Yeni ürün satırını kodları burada yer alacak -->` açıklama satırının yerine yapıştırın.

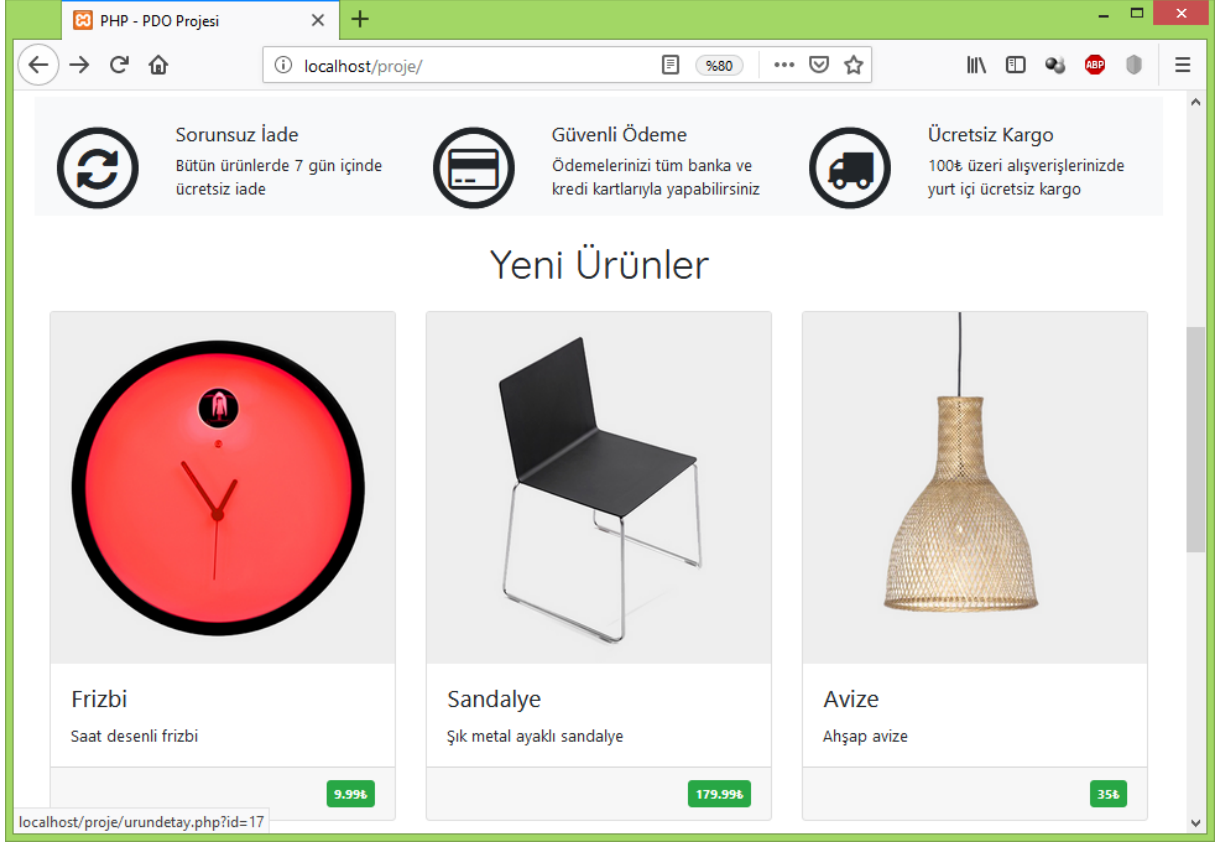


Card kodlarını aşağıdaki gibi veritabanına en son eklenen altı ürün kaydını listeleyen PHP döngüsü içinde olacak şekilde düzenleyin.

```
<h1 class="text-center baslik pt-4 pb-3">Yeni Ürünler</h1>
<div class="row">
  <?php
    // veritabanı yapılandırma dosyasını dahil et
    include 'config/vtabani.php';
    // kayıt listeleme sorgusu
    $sorgu='select id, urunadi, aciklama, fiyat, resim from urunler order
by giris_tarihi desc limit 0,6';
    $stmt = $con->prepare($sorgu); // sorguyu hazırla
    $stmt->execute(); // sorguyu çalıştır
    $veri = $stmt->fetchAll(PDO::FETCH_ASSOC); // tablo verilerini oku
    foreach ($veri as $kayit) { ?>
      <div class="col-md-4 mb-4">
        <div class="card">
          <a href="urundetay.php?id=<?php echo $kayit['id']?>">
            ">
          </a>
          <div class="card-body">
            <h4 class="card-title"><?php echo $kayit['urunadi']?></h4>
            <p class="card-text"><?php echo $kayit['aciklama']?></p>
          </div>
          <div class="card-footer text-muted">
            <span class="badge badge-success p-2 float-right"><?php echo
$kayit['fiyat']?>&#8378;</span>
          </div>
        </div>
      </div>
    </div>
  <?php }
```

```
</div>
```

Tarayıcınızın adres satırına <http://localhost/proje/index.php> yazarak anasayfayı görüntüleyin.



ÜRÜNLER SAYFASINI (URUNLER.PHP) OLUŞTURMAK

Öncelikle tasarlayacağımız ürünler sayfasının taslağını belirlememiz lazım. Sayfa taslağı aşağıdaki gibi olacaktır.



Ürün ara...

Ara

Banner satırı

Kategoriler ▾ Hakkımızda



Kategori listesi

Ürünler

Ürün listesi

Kategoriler

- Elektronik 4
- Genel 4
- Mutfak gereçleri 3

Sıralama seçenekleri

Sıralama

- Akıllı sıralama
- Yeni ürünler
- Artan fiyat
- Azalan fiyat

Fiyat Aralığı

- 0 - 25 ₺
- 25 - 50 ₺
- 50 - 100 ₺
- 100 - 250 ₺
- 250 - 500 ₺
- 500 - 1000 ₺

Fiyat seçenekleri



Yardım

Ödeme
İade ve Değişim
Mağazalarımız
Kariyer Fırsatları
Sık Sorulan Sorular
Kampanyalar

Fırsat Ürünleri



Saç bandı
1.99 ₺



Çöp kovası
3.95 ₺



Gözlük
6 ₺

İletişim

Adres:
Celaliye Mahallesi, Cumhuriyet Cd. No:26
Büyükkçekmece/İstanbul
Telefon: (0212) 885 98 82
Faks: (0212) 885 40 30
Web: http://sudiozkanmtal.meb.k12.tr
Eposta: 7-24@sudiozkanmtal.meb.k12.tr

Satışı yapılan ürünleri listelemek, aramak, sıralamak gibi işlemleri yapacağımız ürünler sayfasını **proje** klasöründe **urunler.php** ismiyle oluşturun ve içine aşağıdaki satırları yazın.

```
<?php include "header.php"; ?>
```

```

<!-- Banner satırı kodları burada yer alacak -->
<div class="container mt-4">
  <div class="row">
    <div class="col-md-3">
      <!-- Kategori liste kodları burada yer alacak -->
      <!-- Sıralama seçenekleri burada yer alacak -->
      <!-- Fiyat seçenekleri burada yer alacak -->
    </div>
    <div class="col-md-9">
      <!-- Ürün listeleme kodları burada yer alacak -->
    </div>
  </div>
</div>

<?php include "footer.php"; ?>

```

BANNER SATIRINI OLUŞTURMAK

urunler.php dosyasında <!-- Banner satırı kodları burada yer alacak --> açıklama satırının yerine aşağıdaki HTML kodlarını yazın.

```

<div class="row">
  <div class="col-md text-center">
    
    <div class="carousel-caption">
      <span class="fa-stack fa-3x">
        <i class="fa fa-circle-thin fa-stack-2x"></i>
        <i class="fa fa-search fa-stack-1x"></i>
      </span>
      <h1 class="baslik"><b>Ürünler</b></h1>
    </div>
  </div>
</div>

```

Not: **content/images** klasörüne **banner.jpg** ismiyle en az 1600px genişliğinde uygun bir resim kaydetmeyi unutmayın.

KATEGORİ LİSTESİNİ OLUŞTURMAK

Kategori listesi, sıralama ve fiyat seçenekleri için **Bootstrap** kütüphanesinden **List group** bileşenini kullanacağız. <https://getbootstrap.com/docs/4.1/components/list-group/> adresindeki **Links and buttons**, **With badges** ve **Flush** başlıkları altındaki örnek kodlarından faydalanacağız.

urunler.php dosyasında <!-- Kategori liste kodları burada yer alacak --> açıklama satırının yerine aşağıdaki kodları yazın.

```

<div><!--kategoriler-->
  <h4 class="baslik">Kategoriler</h4>
  <ul class="list-group list-group-flush">
    <?php
      // veritabanı yapılandırma dosyasını dahil et
      include 'config/vtabani.php';
      // kayıt listeleme sorgusu
      $sorgu='SELECT kategoriler.*, COUNT(urunler.id) AS adet FROM kategoriler LEFT
      JOIN urunler ON kategoriler.id=urunler.kategori_id GROUP BY kategoriler.id ORDER
      BY kategoriler.kategoriadi';
      $stmt = $con->prepare($sorgu); // sorguyu hazırla
      $stmt->execute(); // sorguyu çalıştır
      $veri = $stmt->fetchAll(PDO::FETCH_ASSOC); // tablo verilerini oku
      $toplamlam=0;
      foreach ($veri as $kayit) { ?>

```

```

        <a href="urunler.php?id=<?php echo $kayit["id"]; ?>" class="list-group-item
d-flex justify-content-between align-items-center list-group-item-action"><?php
echo $kayit["kategoriadi"] ?>
        <span class="badge badge-success badge-pill"><?php echo $kayit["adet"]
?></span>
    </a>
    <?php $toplam+=$kayit["adet"];
} ?>
    <a href="urunler.php" class="list-group-item d-flex justify-content-between
align-items-center list-group-item-action">Hepsi
        <span class="badge badge-success badge-pill"><?php echo $toplam ?></span>
    </a>
</ul>
</div><!--/kategoriler-->

```

Bu şekilde oluşturulan kategori listesi biraz eksik kaldı. Mesela kullanıcı bir kategoriye tıkladığında aranan ürün bilgisi, fiyat ve sıralama bilgisi linke eklenmediğinden bir sonraki tıklamada bu parametreler sıfırlanacaktır. Ayrıca aktif kategori seçeneğini de vurgulamamız gerekir. Bu eksiklikleri düzeltmek için kategori liste kodunu aşağıdaki gibi yeniden düzenleyin.

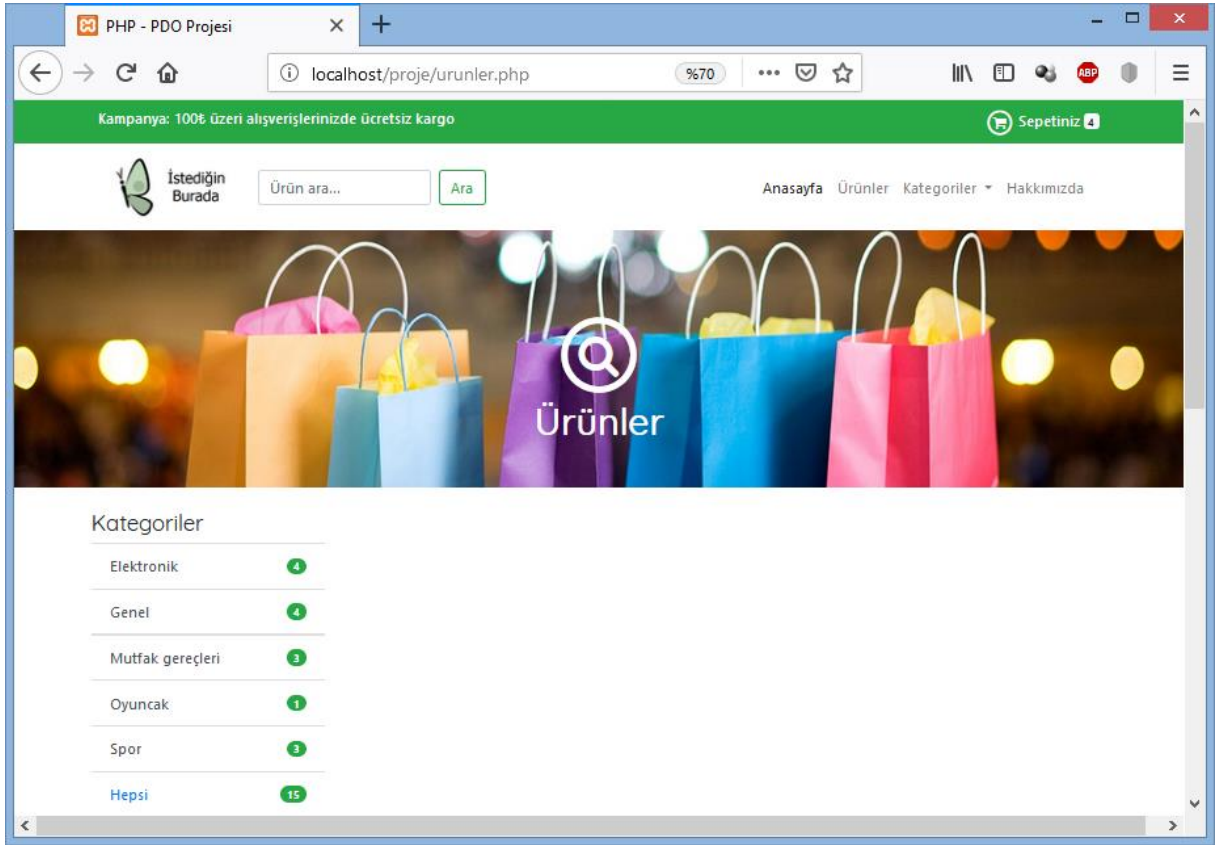
```

<div><!--kategoriler-->
    <h4 class="baslik">Kategoriler</h4>
    <ul class="list-group list-group-flush">
        <?php
            // gelen parametreleri değişkenlere kaydet
            $aranan=isset($_GET["aranan"]) ? $_GET["aranan"]:"";
            $kategori=isset($_GET["id"]) ? $_GET["id"]:"";
            $siralama=isset($_GET["siralama"]) ? $_GET["siralama"]:"akilli";
            $fiyat=isset($_GET["fiyat"]) ? $_GET["fiyat"]:"0";
            // kategori haricindeki seçenekleri de linke dahil etmek için
            $parametre="&aranan=$aranan&siralama=$siralama&fiyat=$fiyat";

            // veritabanı yapılandırma dosyasını dahil et
            include 'config/vtabani.php';
            // kayıt listeleme sorgusu
            $sorgu='SELECT kategoriler.*, COUNT(urunler.id) AS adet FROM kategoriler LEFT
JOIN urunler ON kategoriler.id=urunler.kategori_id GROUP BY kategoriler.id ORDER
BY kategoriler.kategoriadi';
            $stmt = $con->prepare($sorgu); // sorguyu hazırla
            $stmt->execute(); // sorguyu çalıştır
            $veri = $stmt->fetchAll(PDO::FETCH_ASSOC); // tablo verilerini oku
            $toplam=0;
            foreach ($veri as $kayit) { ?>
                <a href="urunler.php?id=<?php echo $kayit["id"]; echo $parametre ?>"
class="list-group-item d-flex justify-content-between align-items-center list-
group-item-action"><?php echo $kategori==$kayit["id"] ? "active":""; ?>"><?php echo
$kayit["kategoriadi"] ?>
                    <span class="badge badge-success badge-pill"><?php echo $kayit["adet"]
?></span>
                </a>
                <?php $toplam+=$kayit["adet"]; } ?>
                <a href="urunler.php?id=<?php echo $parametre ?>" class="list-group-item d-
flex justify-content-between align-items-center list-group-item-action"><?php echo
$kategori==" ? "active":""; ?>">Hepsi
                    <span class="badge badge-success badge-pill"><?php echo $toplam ?></span>
                </a>
            </ul>
        </div><!--/kategoriler-->

```

Tarayıcınızın adres satırına **http://localhost/proje/urunler.php** yazarak ürünler sayfasını görüntüleyin.



SIRALAMA SEÇENEKLERİNİ OLUŞTURMAK

urunler.php dosyasında `<!-- Sıralama seçenekleri burada yer alacak -->` açıklama satırının yerine aşağıdaki kodları yazın.

```
<div class="pt-4"><!--siralama-->
  <h4 class="baslik">Sıralama</h4>
  <div class="list-group list-group-flush">
    <a href="urunler.php?siralama=akilli" class="list-group-item list-group-item-action">Akıllı sıralama</a>
    <a href="urunler.php?siralama=yeni" class="list-group-item list-group-item-action">Yeni ürünler</a>
    <a href="urunler.php?siralama=artan" class="list-group-item list-group-item-action">Artan fiyat</a>
    <a href="urunler.php?siralama=azalan" class="list-group-item list-group-item-action">Azalan fiyat</a>
  </div>
</div><!--/siralama-->
```

Yine bu şekilde oluşturulan sıralama seçenekleri de biraz eksik kaldı. Mesela kullanıcı bir sıralama seçeneğini tıkladığında aranan ürün bilgisi, fiyat ve kategori bilgisi linke eklenmediğinden bir sonraki tıklamada bu **parametreler** sıfırlanacaktır. Ayrıca **aktif** sıralama seçeneğini de vurgulamamız gerekir. Bu eksiklikleri düzeltmek için sıralama seçenekleri kodunu aşağıdaki gibi yeniden düzenleyin.

```
<div class="pt-4"><!--siralama-->
  <h4 class="baslik">Sıralama</h4>
  <div class="list-group list-group-flush">
    <!-- sıralama haricindeki seçenekleri de linke dahil etmek için -->
    <?php $parametre="&aranan=$aranan&id=$kategori&fiyat=$fiyat"; ?>

    <a href="urunler.php?siralama=akilli<?php echo $parametre ?>" class="list-group-item list-group-item-action<?php echo $siralama=="akilli" ? "active":""; ?>">Akıllı sıralama</a>
    <a href="urunler.php?siralama=yeni<?php echo $parametre ?>" class="list-group-item list-group-item-action<?php echo $siralama=="yeni" ? "active":""; ?>">Yeni ürünler</a>
```

```

    <a href="urunler.php?siralama=artan<?php echo $parametre ?>" class="list-
group-item list-group-item-action<?php echo $siralama=="artan" ? "active":"";
?>">Artan fiyat</a>
    <a href="urunler.php?siralama=azalan<?php echo $parametre ?>" class="list-
group-item list-group-item-action<?php echo $siralama=="azalan" ? "active":"";
?>">Azalan fiyat</a>
</div>
</div><!--/siralama-->

```

FİYAT SEÇENEKLERİNİ OLUŞTURMAK

urunler.php dosyasında <!-- Fiyat seçenekleri burada yer alacak --> açıklama satırının yerine aşağıdaki kodları yazın.

```

<div class="pt-4"><!--fiyat-->
  <h4 class="baslik">Fiyat Aralığı</h4>
  <div class="list-group list-group-flush">
    <a href="urunler.php?fiyat=0" class="list-group-item list-group-item-
action">Yok</a>
    <a href="urunler.php?fiyat=1" class="list-group-item list-group-item-action">0
- 25 &#8378;</a>
    <a href="urunler.php?fiyat=2" class="list-group-item list-group-item-
action">25 - 50 &#8378;</a>
    <a href="urunler.php?fiyat=3" class="list-group-item list-group-item-
action">50 - 100 &#8378;</a>
    <a href="urunler.php?fiyat=4" class="list-group-item list-group-item-
action">100 - 250 &#8378;</a>
    <a href="urunler.php?fiyat=5" class="list-group-item list-group-item-
action">250 - 500 &#8378;</a>
    <a href="urunler.php?fiyat=6" class="list-group-item list-group-item-
action">500 - 1000 &#8378;</a>
  </div>
</div><!--/fiyat-->

```

Bu şekilde oluşturulan fiyat seçenekleri de eksik kaldı. Mesela kullanıcı bir fiyat seçeneğini tıkladığında aranan ürün bilgisi, sıralama ve kategori bilgisi linke eklenmediğinden bir sonraki tıklamada bu parametreler sıfırlanacaktır. Ayrıca aktif fiyat seçeneğini de vurgulamamız gerekir. Bu eksiklikleri düzeltmek için sıralama seçenekleri kodunu aşağıdaki gibi yeniden düzenleyin.

```

<div class="pt-4"><!--fiyat-->
  <h4 class="baslik">Fiyat Aralığı</h4>
  <div class="list-group list-group-flush">
    <!-- fiyat haricindeki seçenekleri de linke dahil etmek için -->
    <?php $parametre="&aranan=$aranan&id=$kategori&siralama=$siralama"; ?>

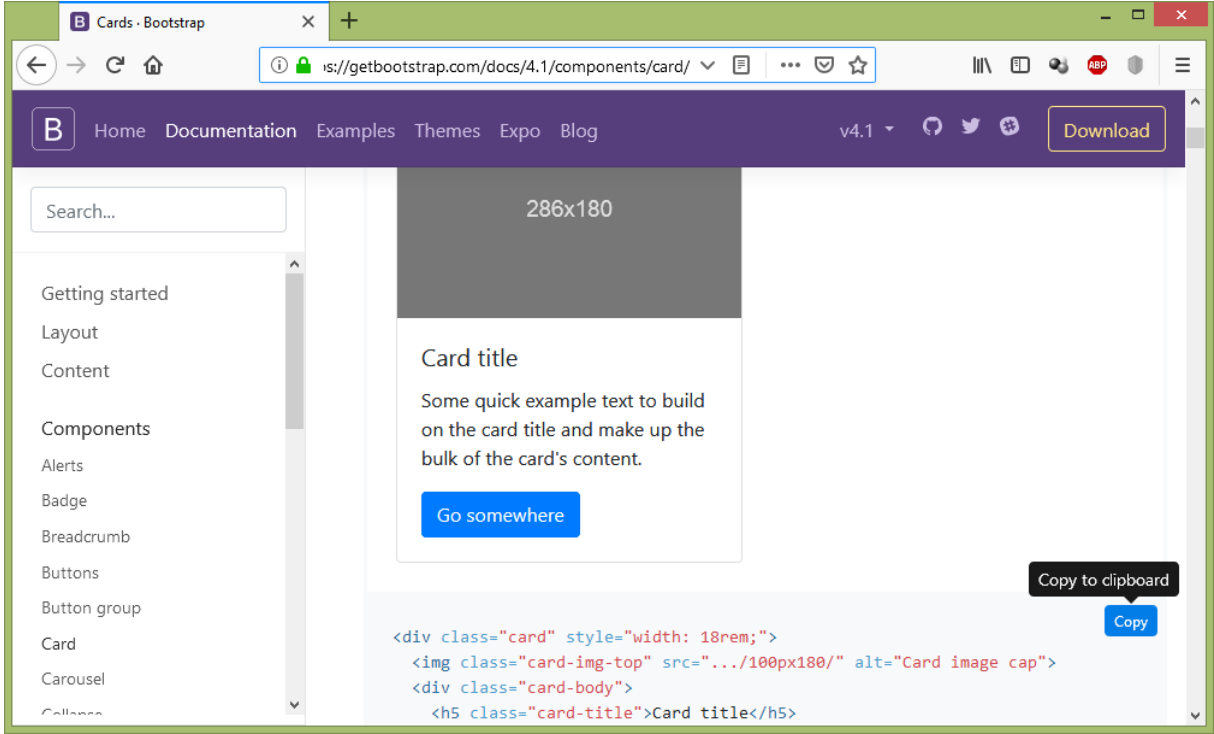
    <a href="urunler.php?fiyat=0<?php echo $parametre ?>" class="list-group-item
list-group-item-action<?php echo $fiyat=="0" ? "active":""; ?>">Yok</a>
    <a href="urunler.php?fiyat=1<?php echo $parametre ?>" class="list-group-item
list-group-item-action<?php echo $fiyat=="1" ? "active":""; ?>">0 - 25 &#8378;</a>
    <a href="urunler.php?fiyat=2<?php echo $parametre ?>" class="list-group-item
list-group-item-action<?php echo $fiyat=="2" ? "active":""; ?>">25 - 50
&#8378;</a>
    <a href="urunler.php?fiyat=3<?php echo $parametre ?>" class="list-group-item
list-group-item-action<?php echo $fiyat=="3" ? "active":""; ?>">50 - 100
&#8378;</a>
    <a href="urunler.php?fiyat=4<?php echo $parametre ?>" class="list-group-item
list-group-item-action<?php echo $fiyat=="4" ? "active":""; ?>">100 - 250
&#8378;</a>
    <a href="urunler.php?fiyat=5<?php echo $parametre ?>" class="list-group-item
list-group-item-action<?php echo $fiyat=="5" ? "active":""; ?>">250 - 500
&#8378;</a>
    <a href="urunler.php?fiyat=6<?php echo $parametre ?>" class="list-group-item
list-group-item-action<?php echo $fiyat=="6" ? "active":""; ?>">500 - 1000
&#8378;</a>
  </div>

```

```
</div><!--/fiyat-->
```

ÜRÜN LİSTESİNİ OLUŞTURMAK

Ürün listesi için **Bootstrap** kütüphanesinden **Card** bileşenini kullanacağız. <https://getbootstrap.com/docs/4.1/components/card/> adresinden örnek kodları kopyalayın ve **index.php** dosyasında `<!-- Ürün listeleme kodları burada yer alacak -->` açıklama satırının yerine yapıştırın.



Card kodlarını aşağıdaki gibi - yani veritabanındaki ürün kayıtlarını şarta bağlı olarak listeleyen ve sıralayan PHP döngüsü içinde olacak şekilde - düzenleyin.

```
<div class="row">
  <?php
  // select sorgusu için sıralama seçenekleri hazırlanıyor
  switch($siralama){
    case 'yeni': $orderby="giris_tarihi desc";break;
    case 'artan': $orderby="fiyat";break;
    case 'azalan':$orderby="fiyat desc";break;
    default:     $orderby="dzltm_tarihi desc";
  }
  // select sorgusu için fiyat aralığı hazırlanıyor
  switch($fiyat){
    case '1': $where1="fiyat between 0 and 25";break;
    case '2': $where1="fiyat between 25 and 50";break;
    case '3': $where1="fiyat between 50 and 100";break;
    case '4': $where1="fiyat between 100 and 250";break;
    case '5': $where1="fiyat between 250 and 500";break;
    case '6': $where1="fiyat between 500 and 1000";break;
  }
  // select sorgusu için kategori şartı hazırlanıyor
  if($kategori!="") $where2="kategori_id=$kategori";
  // select sorgusu için arama şartı hazırlanıyor
  if($aranan!="") $where3="urunadi like '%$aranan%'"; else $where3="urunadi like
  '% '";
  // select sorgusu için şartlar birleştiriliyor
  if(isset($where1) && isset($where2)) $where="(".$where1.") and ( ".$where2.") and
  ( ".$where3.")";
  elseif(isset($where1)) $where="(".$where1.") and ( ".$where3.")";
```

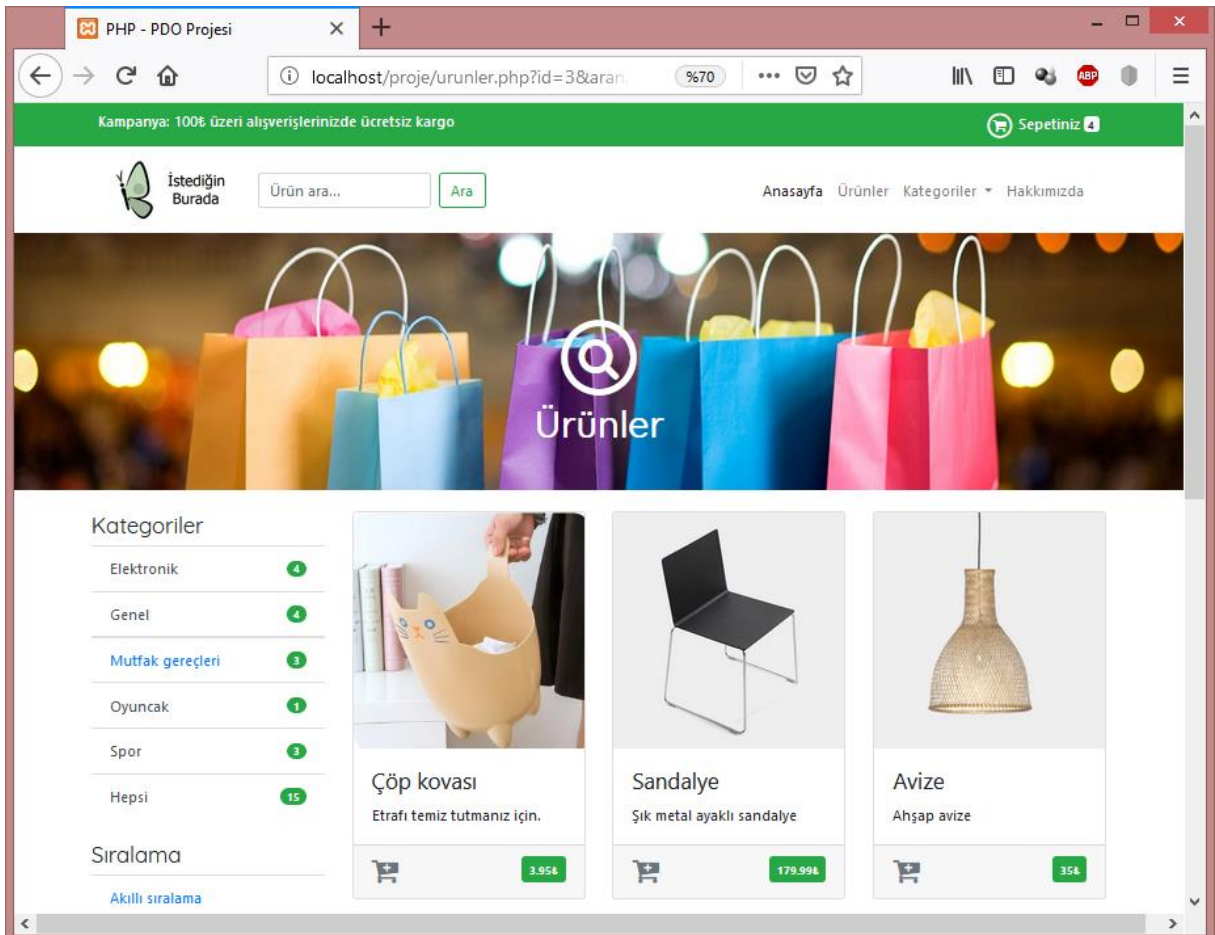
```

elseif(isset($where2)) $where="(".$where2.") and (".$where3.")";
else $where=$where3;

// veritabanı yapılandırma dosyasını dahil et
include 'config/vtabani.php';
// kayıt listeleme sorgusu
$sorgu="select id, urunadi, aciklama, fiyat, resim from urunler where $where
order by $orderby";
$stmt = $con->prepare($sorgu); // sorguyu hazırla
$stmt->execute(); // sorguyu çalıştır
$veri = $stmt->fetchAll(PDO::FETCH_ASSOC); // tablo verilerini oku
foreach ($veri as $kayit) { ?>
    <div class="col-md-4 mb-4">
        <div class="card">
            <a href="urundetay.php?id=<?php echo $kayit['id']?>">
                ">
            </a>
            <div class="card-body">
                <h4 class="card-title"><?php echo $kayit['urunadi']?></h4>
                <p class="card-text"><?php echo $kayit['aciklama']?></p>
            </div>
            <div class="card-footer text-muted">
                <a href="#" class="text-secondary float-left"><i class="fa fa-cart-plus
fa-2x"></i></a>
                <span class="badge badge-success p-2 float-right"><?php echo
$kayit['fiyat']?>&#8378;</span>
            </div>
        </div>
    </div>
    <?php } ?>
</div>

```

Tarayıcınızın adres satırına <http://localhost/proje/urunler.php> yazarak ürünler sayfasını görüntüleyin.



ÜRÜN DETAY SAYFASINI (URUNDETAY.PHP) OLUŞTURMAK


İşe tasarlayacağımız ürün detay sayfasının taslağını belirlemekle başlayalım. Sayfa taslağı aşağıdaki gibi olacaktır.

Kampanya: 100€ üzeri alışverişlerinizde ücretsiz kargo 🛒 Sepetiniz 4

İstediklerin Burada

Anasayfa Ürünler Kategoriler Hakkımızda

Ürün resmi



Ürün açıklaması

Sandalye

179.99€

Şık metal ayaklı sandalye

Ürünün stoklarımızı girişi: 30-11-2018
En son bilgi güncelleme: 30-11-2018

1

Mutfak gereçleri kategorisindeki diğer ürünler

Diğer Ürünler

Basketbol topu

En iyilerin kullandığı top markası.

49€

Saç bandı

Atletlerin vazgeçilmezi.

1.99€

Diğer ürünlerin listesi

Gözlük

Güneş ışınlarından korunmak için.

6€

Çöp kovası


Etrafı temiz tutmanız için.


3.95€


Yardım

Ödeme
İade ve Değişim
Mağazalarımız
Kariyer Fırsatları
Sık Sorulan Sorular
Kampanyalar

Fırsat Ürünleri

 Saç bandı 1.99€

 Çöp kovası 3.95€

 Gözlük 6€

İletişim

Adres:
Celaliye Mahallesi, Cumhuriyet Cd. No:26
Büyükcçekmece/İstanbul
Telefon: (0212) 885 98 82
Faks: (0212) 885 40 30
Web: <http://sudiozkanmtal.meb.k12.tr>
Eposta: 7-24@sudiozkanmtal.meb.k12.tr

Seçili ürünün ayrıntılı bilgilerini görüntüleyeceğimiz ürün detay sayfasını **proje** klasöründe **urundetay.php** ismiyle oluşturun ve içine aşağıdaki satırları yazın.

```
<?php include "header.php"; ?>

<!-- Ürün bilgilerini veritabanından getiren PHP kodları burada yer alacak -->
<div class="container mt-4 mb-4">
  <div class="row">
    <div class="col-md-6">
      <!-- Ürün resmi burada yer alacak -->
```

21

```

        </div>
        <div class="col-md-6">
            <!-- Ürün açıklaması burada yer alacak -->
        </div>
    </div>
    <!-- Diğer ürünleri listeleyen kodlar burada yer alacak -->
</div>

<?php include "footer.php"; ?>

```

ÜRÜN BİLGİLERİNİ LİSTELEMELİK İÇİN GEREKLİ SORGUYU OLUŞTURMAK

urundetay.php sayfasında <!-- Ürün bilgilerini veritabanından getiren PHP kodları burada yer alacak --> açıklama satırının yerine aşağıdaki PHP kod satırlarını yazın.

```

<?php
// gelen Id parametresini al
$id=isset($_GET['id']) ? $_GET['id'] : die('HATA: Kayıt bulunamadı.');
```

```

// veritabanı bağlantı dosyasını çağır
include 'config/vtabani.php';

// kayıt bilgilerini oku
try {
    // seçme sorgusunu hazırla
    $sorgu = "SELECT urunler.*, kategoriler.kategoriadi FROM urunler LEFT JOIN
kategoriler ON urunler.kategori_id = kategoriler.id WHERE urunler.id = ?";
    $stmt = $con->prepare( $sorgu ); // Id parametresini bağla
    $stmt->bindParam(1, $id); // sorguyu çalıştır
    $stmt->execute();
    $kayit = $stmt->fetch(PDO::FETCH_ASSOC); // gelen kaydı bir değişkende sakla

    // tablo alanlarını değişkene dönüştürür: $kayit['urunadi'] => $urunadi
    extract($kayit);
}
// hatayı göster
catch(PDOException $exception){
    die('HATA: ' . $exception->getMessage());
}
?>

```

ÜRÜN RESMİNİ GÖRÜNTÜLEMEK

Ardından **urundetay.php** sayfasında <!-- Ürün resmi burada yer alacak --> açıklama satırının yerine aşağıdaki kod satırını yazın.

```

<?php echo $resim ? " <img src='content/images/{ $resim }' alt='{ $urunadi }'
class='img-fluid rounded' />": "Ürün görseli yok."; ?>

```

ÜRÜN BİLGİLERİNİ GÖRÜNTÜLEMEK

Resmin sağında ürün bilgilerini görüntülemek için **urundetay.php** sayfasında <!-- Ürün açıklaması burada yer alacak --> açıklama satırının yerine aşağıdaki kod satırlarını yazın.

```

<h2 class="baslik"><?php echo $urunadi ?></h2>
<h5 class="text-success"><?php echo $fiyat ?>&#8378;</h5><hr>
<p class="mb-5"><?php echo $aciklama ?></p>
<h6 class="text-info">Ürünün stoklarımıza girişi: <?php echo date("d-m-Y",
strtotime($giris_tarihi)) ?></h6>
<h6 class="text-warning">En son bilgi güncelleme: <?php echo date("d-m-Y",
strtotime($dzlmt_tarihi)) ?></h6>

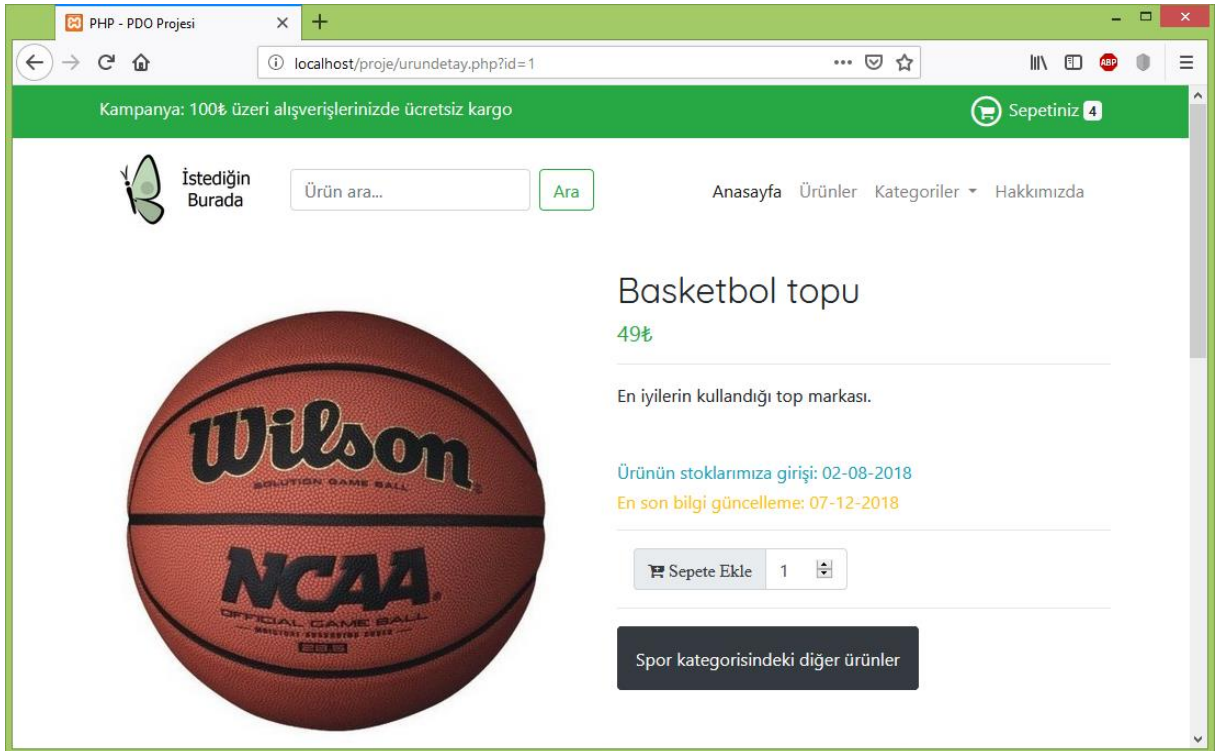
```

```

<hr><h5>
  <div class="col-md-6 input-group">
    <div class="input-group-prepend">
      <div class="input-group-text">
        <a href="#" class="text-dark"><i class="fa fa-cart-plus"> Sepete
Ekle</i></a>
      </div>
    </div>
    <input type="number" min="1" max="99" class="form-control" value="1">
  </div>
</h5><hr>
<p><a href="urunler.php?id=<?php echo $kategori_id ?>" class="btn btn-dark p-3"><?php echo $kategoriadi ?> kategorisindeki diğ er ürünler</a></p>

```

Tarayıcınızın adres satırına **http://localhost/proje/urunler.php** yazarak ürünler sayfasını görüntüleyin ve bir ürün seçerek **urundetay.php** sayfasını açın.



DİĞ ER ÜRÜNLERİ LİSTELEM EK

Ürün bilgilerinin altında diğ er ürünlerden dört tanesini listelemek için **urundetay.php** sayfasında `<!-- Diğ er ürünleri listeleyen kodlar burada yer alacak -->` açıklama satırının yerine aşağıdaki kod satırlarını yazın.

```

<h4 class="baslik mt-5">Diğ er Ürünler</h2>
<div class="row">
  <?php
  // veritabanı yapılandırma dosyasını dahil et
  include 'config/vtabani.php';
  // kayıt listeleme sorgusu
  $sorgu='select id, urunadi, aciklama, fiyat, resim from urunler limit 0,4';
  $stmt = $con->prepare($sorgu); // sorguyu hazırla
  $stmt->execute(); // sorguyu çalıştır
  $veri = $stmt->fetchAll(PDO::FETCH_ASSOC); // tablo verilerini oku
  foreach ($veri as $kayit) { ?>
    <div class="col-md-3 mb-3">
      <div class="card">
        <a href="urundetay.php?id=<?php echo $kayit['id']?>">
          ">

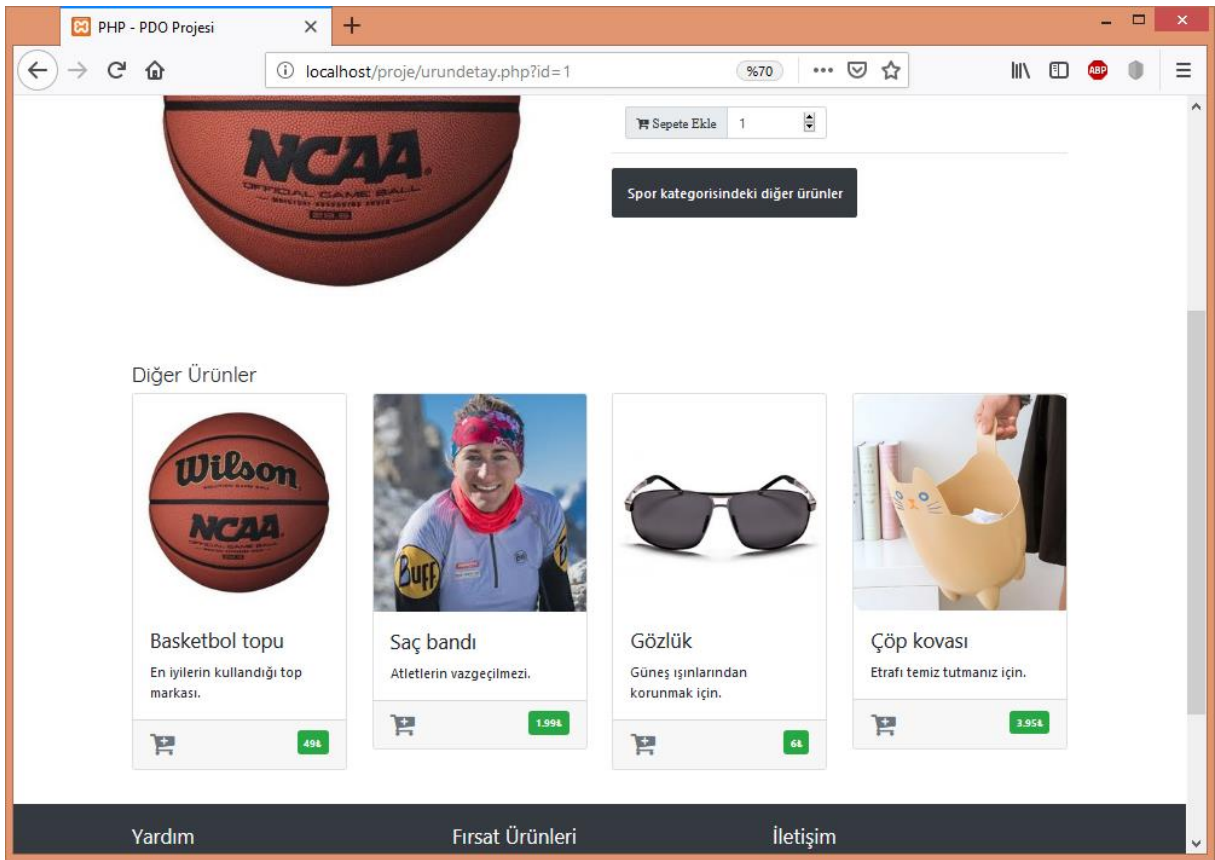
```

```

</a>
<div class="card-body">
  <h4 class="card-title"><?php echo $kayit['urunadi']?></h4>
  <p class="card-text"><?php echo $kayit['aciklama']?></p>
</div>
<div class="card-footer text-muted">
  <a href="#" class="text-secondary float-left"><i class="fa fa-cart-plus
fa-2x"></i></a>
  <span class="badge badge-success p-2 float-right"><?php echo
$kayit['fiyat']?>&#8378;</span>
</div>
</div>
</div>
<?php } ?>
</div>

```

Tarayıcınızın adres satırına **http://localhost/proje/urunler.php** yazarak ürünler sayfasını görüntüleyin ve bir ürün seçerek **urundetay.php** sayfasını açın.



HAKKIMIZDA SAYFASINI (HAKKIMIZDA.PHP) OLUŞTURMAK

Hakkımızda sayfasında sitenin kısa bir tanıtım yazısı ve altında da bir iletişim formu yer alacak. Bu iletişim formu vasıtasıyla müşteriler dilek / talep / öneri / soru / teşekkür / şikâyet e-postaları gönderebileceklerdir.

TANITIM YAZISI İLE İLETİŞİM FORMUNU OLUŞTURMAK

Hakkımızda sayfasını **proje** klasöründe **hakkimizda.php** ismiyle oluşturun ve içine aşağıdaki satırları yazın.

```

<?php include "header.php"; ?>

<div class="container p-5">
  <!-- E-posta gönderme iletisi burada yer alacak -->

  <!-- Site tanıtım yazısı burada yer alacak -->

```



```
<!-- İletişim formu burada yer alacak -->
<div>

<?php include "footer.php"; ?>
```

Site hakkında bilgi içeren yazıyı oluşturmak için **hakkimizda.php** sayfasında `<!-- Site tanıtım yazısı burada yer alacak -->` açıklama satırının yerine aşağıdaki kod satırlarını yazın.

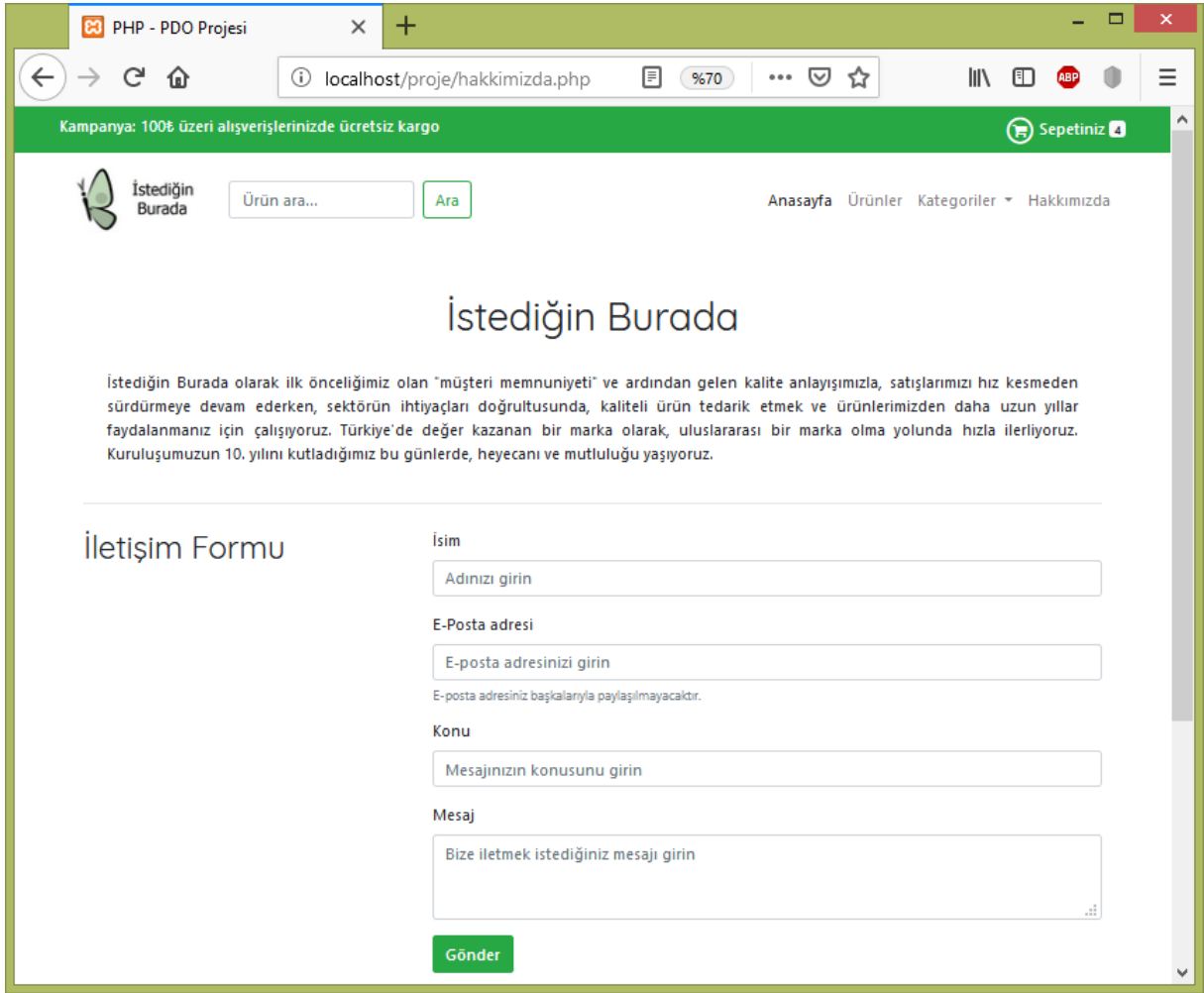
```
<h1 class="text-center baslik">İstediğin Burada</h1>
<p class="text-justify p-4">İstediğin Burada olarak ilk önceliğimiz olan "müşteri memnuniyeti" ve ardından gelen kalite anlayışımızla, satışlarımızı hız kesmeden sürdürmeye devam ederken, sektörün ihtiyaçları doğrultusunda, kaliteli ürün tedarik etmek ve ürünlerimizden daha uzun yıllar faydalanmanız için çalışıyoruz. Türkiye'de değer kazanan bir marka olarak, uluslararası bir marka olma yolunda hızla ilerliyoruz. Kuruluşumuzun 10. yılını kutladığımız bu günlerde, heyecanı ve mutluluğu yaşıyoruz.</p><hr>
```

İletişim formunu oluşturmak için **Bootstrap** kütüphanesinin **Form** örneklerinden faydalanacağız. <https://getbootstrap.com/docs/4.1/components/forms/> adresindeki örnek kodlara bakabilirsiniz.

İhtiyacınıza göre oluşturduğunuz iletişim formu kodlarını da `<!-- İletişim formu burada yer alacak -->` açıklama satırının yerine yazın.

```
<div class="row mt-4">
  <div class="col-md-4">
    <h2 class="baslik">İletişim Formu</h2>
  </div>
  <div class="col-md-8">
    <form action="eposta.php" method="post" name="iletisim">
      <div class="form-group">
        <label for="exampleInputText1">İsim</label>
        <input type="text" class="form-control" id="exampleInputText1"
placeholder="Adınızı girin" name="kimden">
      </div>
      <div class="form-group">
        <label for="exampleInputEmail1">E-Posta adresi</label>
        <input type="email" class="form-control" id="exampleInputEmail1" aria-
describedby="emailHelp" placeholder="E-posta adresinizi girin" name="eposta">
        <small id="emailHelp" class="form-text text-muted">E-posta adresiniz
başkalarıyla paylaşılmayacaktır.</small>
      </div>
      <div class="form-group">
        <label for="exampleInputText2">Konu</label>
        <input type="text" class="form-control" id="exampleInputText2"
placeholder="Mesajınızın konusunu girin" name="konu">
      </div>
      <div class="form-group">
        <label for="exampleFormControlTextarea1">Mesaj</label>
        <textarea class="form-control" id="exampleFormControlTextarea1" rows="4"
placeholder="Bize iletmek istediğiniz mesajı girin" name="mesaj"></textarea>
      </div>
      <button type="submit" class="btn btn-success">Gönder</button>
    </form>
  </div>
</div>
```

Tarayıcınızın adres satırına <http://localhost/proje/hakkimizda.php> yazarak Hakkımızda sayfasını görüntüleyin.



E-POSTA GÖNDERME SAYFASINI OLUŞTURMAK

Hakkımızda sayfasında girilen bilgileri e-posta yoluyla yollamak için **proje** klasöründe **eposta.php** ismiyle bir dosya oluşturun ve içine aşağıdaki satırları yazın.

```
<?php
if($_POST){
    $kime = 'destek@sudiozkanmtal.meb.k12.tr';
    $kimden = $_POST['kimden'];
    $eposta = $kimden.'<'.$eposta.'>';
    $konu = $_POST['konu'];
    $mesaj = $_POST['mesaj'] . " - " . $kimden;
    $tanimlar = 'MIME-Version: 1.0' . "\r\n" .
        'Content-type: text/html; charset=UTF-8' . "\r\n" .
        'From: {$eposta}' . "\r\n" .
        'Reply-To: destek@sudiozkanmtal.meb.k12.tr' . "\r\n" .
        'X-Mailer: PHP/' . phpversion();
    try {
        mail($kime, $konu, $mesaj, $tanimlar);
        header('Location: hakkimizda.php?islem=tamam');
    }
    catch(Exception $e) {
        header('Location: hakkimizda.php?islem=hata');
    }
}
?>
```

Daha sonra **hakkimizda.php** sayfasını tekrar açın ve `<!-- E-posta gönderme iletisi burada yer alacak -->` açıklama satırının yerine aşağıdaki kod satırlarını yazın.

```
<?php
```

```
$islem = isset($_GET['islem']) ? $_GET['islem'] : "";

// eğer eposta.php sayfasından yönlendirme yapıldıysa
if($islem == 'tamam') {
    echo "<div class='alert alert-success'>Mesajınız başarıyla iletildi.</div>";
}
else if($islem == 'hata') {
    echo "<div class='alert alert-danger'>Mesajınız iletilemedi!</div>";
}
?>
```

ALIŞVERİŞ SEPETİNİ OLUŞTURMAK

Alışveriş sepetindeki bilgileri saklamak için birkaç yöntem mevcut. Session (oturum) yönteminde bilgiler tarayıcı programı kapanıncaya kadar hafızada saklı kalır; sunucu taraflıdır. Böylece siz çıktıktan sonra başkası hangi ürünlerle ilgilendiğinizi göremez. Cookie (çerez) yönteminde ise bilgiler kullanıcının bilgisayarındaki geçici bir dosyada saklanır; istemci taraflıdır. Fakat sizden sonra başkası aynı siteye girdiğinde sizin sepetinizi görebilecektir. Her iki yöntemde de farklı cihazlardan girildiğinde sepetteki bilgiler kaybedileceğinden en güvenilir yöntem üyelik sistemi oluşturup kullanıcı giriş yaptığında sepet bilgilerinin veritabanında saklanması olacaktır.

Biz bir üyelik sistemi oluşturmadığımız için örneğimizde Session yöntemini uygulayacağız.

SEPET OTURUMUNU OLUŞTURMAK

Oturum işlemini başlık sayfasında (header.php) oluşturacağız. Bu sayfa diğer sayfaların başına eklendiği için burada başlattığımız oturum tüm sayfalarda geçerli olacaktır.

header.php dosyasını açın ve HTML etiketlerinin başına aşağıdaki kod satırlarını ekleyin.

```
<?php
// Oturum işlemlerini başlat
session_start();

// Sepet oluşturulmamışsa oluştur
$_SESSION['sepet']=isset($_SESSION['sepet']) ? $_SESSION['sepet'] : array();
?>
```

Ardından aynı dosyada içinde `<!-- Sepetteki ürün sayısı -->` açıklaması bulunan **** etiketini aşağıdaki gibi düzenleyin.

```
<span class="badge badge-light" id="urun-sayisi">
<!-- Sepetteki ürün sayısı -->
<?php
    $urun_sayisi=count($_SESSION['sepet']);
    echo $urun_sayisi;
?>
</span>
```

SEPETE EKLE BUTONU KODLARINI OLUŞTURMAK

urunler.php ve **urundetay.php** sayfalarında üzerinde alışveriş sepeti simgesi bulunan butonlar (Sepete Ekle) tıklanınca ilgili ürün sepete eklenecek. Butonların tetiklediği JavaScript fonksiyonlarını tüm sayfalar için ortak olan altbilgi sayfasında (footer.php) oluşturacağız. Fakat daha önce buton etiketlerinin **class** ve **id** parametrelerini tanımlamalıyız.

BUTON ETİKETLERİNİ DÜZENLEMEK

urunler.php sayfasını açın ve **Sepete Ekle** butonunun HTML etiketini aşağıdaki gibi düzenleyin.

```
<a href="#" class="text-secondary float-left sepete-ekle" id="<?php echo $kayit['id']?>"><i class="fa fa-cart-plus fa-2x"></i></a>
```

Şimdi de **urundetay.php** sayfasını açın ve benzer şekilde **Sepete Ekle** butonunun HTML etiketini aşağıdaki gibi düzenleyin.

```
<a href="#" class="text-dark sepete-ekle" id="<?php echo $kayit['id']?>"><i class="fa fa-cart-plus"> Sepete Ekle</i></a>
```

Ardından ürün sayısının girildiği metin kutusunun HTML etiketini düzenleyin.

```
<input type="number" min="1" max="99" class="form-control" value="1" id="urun_<?php echo $id; ?>">
```

Son olarak **Diğer Ürünler** başlığı altındaki ürünlerin **Sepete Ekle** buton etiketini de düzenleyin.

```
<a href="#" class="text-secondary float-left sepete-ekle" id="<?php echo $kayit['id']?>"><i class="fa fa-cart-plus fa-2x"></i></a>
```

JAVASCRIPT FONKSİYONUNU OLUŞTURMAK

footer.php sayfa kodlarını açın ve `<!-- Kullanıcı tanımlı JavaScript kodları burada yer alacak -->` açıklama satırının yerine aşağıdaki satırları yazın.

Not: Uyarı mesajları için daha iyi bir görsel arayüze sahip **SweetAlert** eklentisi kullanılacak.

```
<!-- SweetAlert eklentisi -->
<script src="https://unpkg.com/sweetalert/dist/sweetalert.min.js"></script>

<!-- Sepet işlemleri için gerekli JavaScript fonksiyonları -->
<script>
    $(document).ready(function(){

        // sepete-ekle butonu ile tetiklenen fonksiyon burada yer alacak

        // urun-guncelle butonu ile tetiklenen fonksiyon burada yer alacak

        // urun-sil butonu ile tetiklenen fonksiyon burada yer alacak
    });
</script>
```

`// sepete-ekle butonu ile tetiklenen fonksiyon burada yer alacak` açıklama satırının yerine aşağıdaki JavaScript fonksiyonunu yazın.

```
// sepete-ekle butonu ile tetiklenir
$('.sepete-ekle').on('click', function(){

    var id = $(this).attr('id');
    var sayi = parseInt(document.getElementById('urun-sayisi').innerHTML);
    if (document.getElementById('urun_'+id) !== null) {
        // buton urundetay.php sayfasında tıklanmış
        var adet=document.getElementById('urun_'+id).value;
    }
    else {
        // buton urunler.php sayfasında tıklanmış
        var adet=1;
    }

    // id ve adet parametreleri sepete_ekle.php sayfasına yönlendirilir
    $.ajax ({
        cache: false,
        type: 'POST',
        url: 'sepete_ekle.php',
        data: {id:id, adet:adet},
```

```

success: function(sonuc){
    if(sonuc=="true"){
        swal("Ürün sepete eklendi!", {
            icon: "success",
            buttons: false,
            timer: 2000,
        });
        // sepetteki ürün sayısını güncelle
        $("#urun-sayisi").text(sayi + 1);
    }
    else{
        swal("Ürün daha önce sepete eklenmiş!", {
            icon: "warning",
            buttons: false,
            timer: 2000,
        });
    }
},
error: function(jqXHR, textStatus, errorThrown){
    alert(textStatus + errorThrown);
}
});
return false;
});

```

AJAX SAYFASINI (SEPETE_EKLE.PHP) OLUŞTURMAK

Sepete Ekle butonu tıklanınca fonksiyona gelen **id** ve **adet** parametre bilgilerine göre ilgili ürünü sepete ekleyecek kodları **Ajax** metodu ile oluşturacağız. Bunun için **proje** klasöründe **sepete_ekle.php** isimli bir dosya oluşturun ve içine aşağıdaki PHP kodlarını yazın.

```

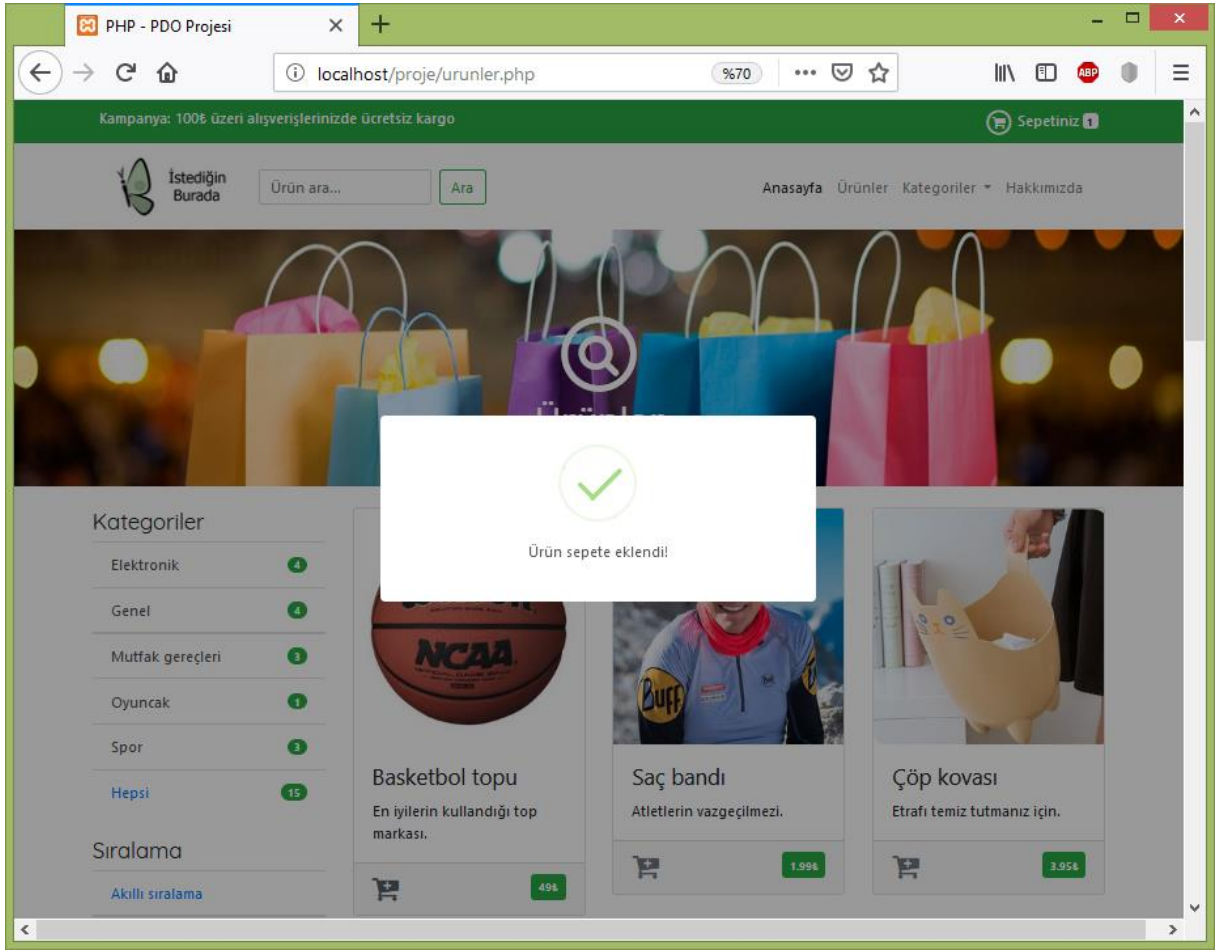
<?php
// oturum işlemlerini başlat
session_start();

// ürün id ve adet bilgilerini al
$id = isset($_POST['id']) ? $_POST['id'] : "";
$adet = isset($_POST['adet']) ? $_POST['adet'] : 1;

// ürün sepette varsa ekleme
if(array_key_exists($id, $_SESSION['sepet'])) {
    echo "false";
}
// ürün sepette yoksa ekle
else{
    $_SESSION['sepet'][$id]=array('adet'=>$adet);
    echo "true";
}
?>

```

Tarayıcınızın adres satırına **http://localhost/proje/urunler.php** yazarak listelenen bir ürünün **Sepete Ekle** butonunu tıklayın ve sepete ekleyin.



SEPETTEKİ ÜRÜNLERİ GÖRÜNTÜLEMEK

Sepetteki ürünleri görüntülemek için kullanılacak sayfayı **proje** klasöründe **sepetim.php** ismiyle oluşturun ve içine aşağıdaki kodları ekleyin.

```
<?php include "header.php";

// sepet boşsa uyar
if(count($_SESSION['sepet']) == 0) {
    echo "<div class='container mt-5 mb-5'><div class='col-md-12'><div class='alert alert-danger'>Sepetinizde henüz ürün yok!</div></div></div>";
}
// sepet boş değilse ürünleri listele
else {

    // sepetteki ürün bilgilerini veritabanından okuyan kodlar burada yer alacak
    ?>

    <div class="container mt-4 mb-5">

        <!-- Sepetteki ürünleri görüntüleyen HTML tablosu burada yer alacak -->

        <!-- Sepet özetini görüntüleyen HTML tablosu burada yer alacak -->

    </div>

<?php
}

include "footer.php"; ?>
```

SEPETTEKİ ÜRÜN KAYITLARINI VERİTABANINDAN OKUMAK

Session dizisindeki ürün id'leri kullanılıp ürün bilgileri veritabanından okunup listelenecek. Bunun için // sepetteki ürün bilgilerini veritabanından okuyan kodlar burada yer alacak açıklama satırının yerine aşağıdaki PHP kodlarını yazın.

```
// sepetteki ürün id'lerini diziye kaydet
$ids = array();
foreach($_SESSION['sepet'] as $id=>$value){
    array_push($ids, $id);
}
$ids_arr = str_repeat('?',',', count($ids) - 1) . '?';

// veritabanı yapılandırma dosyasını dahil et
include 'config/vtabani.php';

// sepetteki ürünleri getiren sorgu
$sorgu = "SELECT id, urunadi, fiyat, resim FROM urunler WHERE id IN ({$ids_arr})
ORDER BY urunadi";

// sorguyu hazırla
$stmt = $con->prepare($sorgu);

// sorguyu çalıştır
$stmt->execute($ids);
```

SEPETTEKİ ÜRÜNLERİ LİSTELEMELİK

<!-- Sepetteki ürünleri görüntüleyen HTML tablosu burada yer alacak --> açıklama satırının yerine aşağıdaki kod satırlarını yazın.

```
<div class="baslik">
    <h3>Sepet Özeti</h3>
</div>
<!-- Sepetteki ürünleri görüntüleyen HTML tablosu -->
<div class="table-responsive">
    <table class="table table-bordered sepet-tablo">
        <thead class="bg-light">
            <tr>
                <th>Ürün görseli</th>
                <th>Ürün adı</th>
                <th>Fiyat</th>
                <th>Adet</th>
                <th>Sil</th>
            </tr>
        </thead>
        <tbody>
            <!-- Ürünleri listeleyen döngü kodları burada yer alacak -->
        </tbody>
    </table>
</div><!--/Sepetteki ürünler-->
```

Ardından <!-- Ürünleri listeleyen döngü kodları burada yer alacak --> açıklama satırının yerine aşağıdaki kodları yazın.

```
<?php
$urun_toplami = 0;
$urun_sayisi = 0;
// Sepetteki ürünleri listeleyen döngü
while ($kayit = $stmt->fetch(PDO::FETCH_ASSOC)) {
    extract($kayit);

    $adet = $_SESSION['sepet'][$id]['adet'];
    $urun_sayisi += $adet;
    $urun_toplami += $fiyat * $adet;
```

```

?>
<tr>
  <td class="col-md-2">
    
  </td>
  <td class="col-md-4 text-left">
    <h6><a href="urun_detay.php?id=<?php echo $id; ?>" class="link2"><?php
echo $urunadi; ?></a></h6>
  </td>
  <td class="col-md-2">
    <h6><?php echo number_format($fiyat, 2, ',', '.'); ?>&#8378;</h6>
  </td>
  <td class="col-md-2">
    <h6>
      <div class="input-group">
        <div class="input-group-prepend">
          <div class="input-group-text">
            <a href="#" class="text-dark urun-guncelle" id="<?php echo $id;
?>"><i class="fa fa-refresh"></i></a>
          </div>
        </div>
        <input type="number" value="<?php echo $adet; ?>" id="urun_<?php echo
$id; ?>" min="1" max="99" class="form-control">
      </div>
    </h6>
  </td>
  <td class="col-md-1">
    <h6><a href="#" class="link2 urun-sil" id="<?php echo $id; ?>"><i
class="fa fa-trash"></i></a></h6>
  </td>
</tr>
<?php
} // while döngüsü sonu
$kargo_ucreti = 5.99;
$toplam = $urun_toplami + $kargo_ucreti;
?>

```

SEPET ÖZETİNİ GÖRÜNTÜLEMEK

<!-- Sepet özetini görüntüleyen HTML tablosu burada yer alacak --> açıklama satırının yerine aşağıdaki kodları yazın.

```

<div class="row">
  <div class="col-sm-8">
  </div>
  <div class="col-sm-4">
    <!--Sepet özeti-->
    <div class="text-right">
      <table class="table">
        <tbody>
          <tr>
            <td>Ürün Sayısı</td>
            <td><?php echo $urun_sayisi; ?></td>
          </tr>
          <tr>
            <td>Ürün Tutarı</td>
            <td><?php echo number_format($urun_toplami, 2, ',', '.');
?>&#8378;</td>
          </tr>
          <tr>
            <td>Kargo Ücreti</td>
            <td><?php echo number_format($kargo_ucreti, 2, ',', '.');
?>&#8378;</td>
          </tr>
        </tbody>
      </table>
    </div>
  </div>
</div>

```



```

                <tr>
                    <td><strong>Toplam</strong></td>
                    <td><strong><?php echo number_format($toplam, 2, ',', '.');
?>&#8378;</strong></td>
                </tr>
            </tbody>
        </table>
    </div><!--/Sepet özeti-->
    <form name="frm_sepet" method="post" action="satinal.php">
        <input type="hidden" name="uruntutari" value="<?php echo
$urun_toplam; ?>">
        <input type="hidden" name="kargo" value="<?php echo $kargo_ucreti;
?>">
        <input type="hidden" name="toplam" value="<?php echo $toplam; ?>">
        <div class="text-right">
            <button type="submit" class="btn btn-success btn-lg btn-block">
                <i class="fa fa-credit-card"></i> Satın al
            </button>
        </div>
    </form>
</div>
</div>

```

CSS STİL DOSYASINI (STYLE.CSS) DÜZENLEMEK

Web sayfalarında etiketlere CSS stil sınıfları tanımlamak için oluşturduğunuz **content/css/style.css** dosyasını açın ve içine aşağıdaki kodları ekleyin.

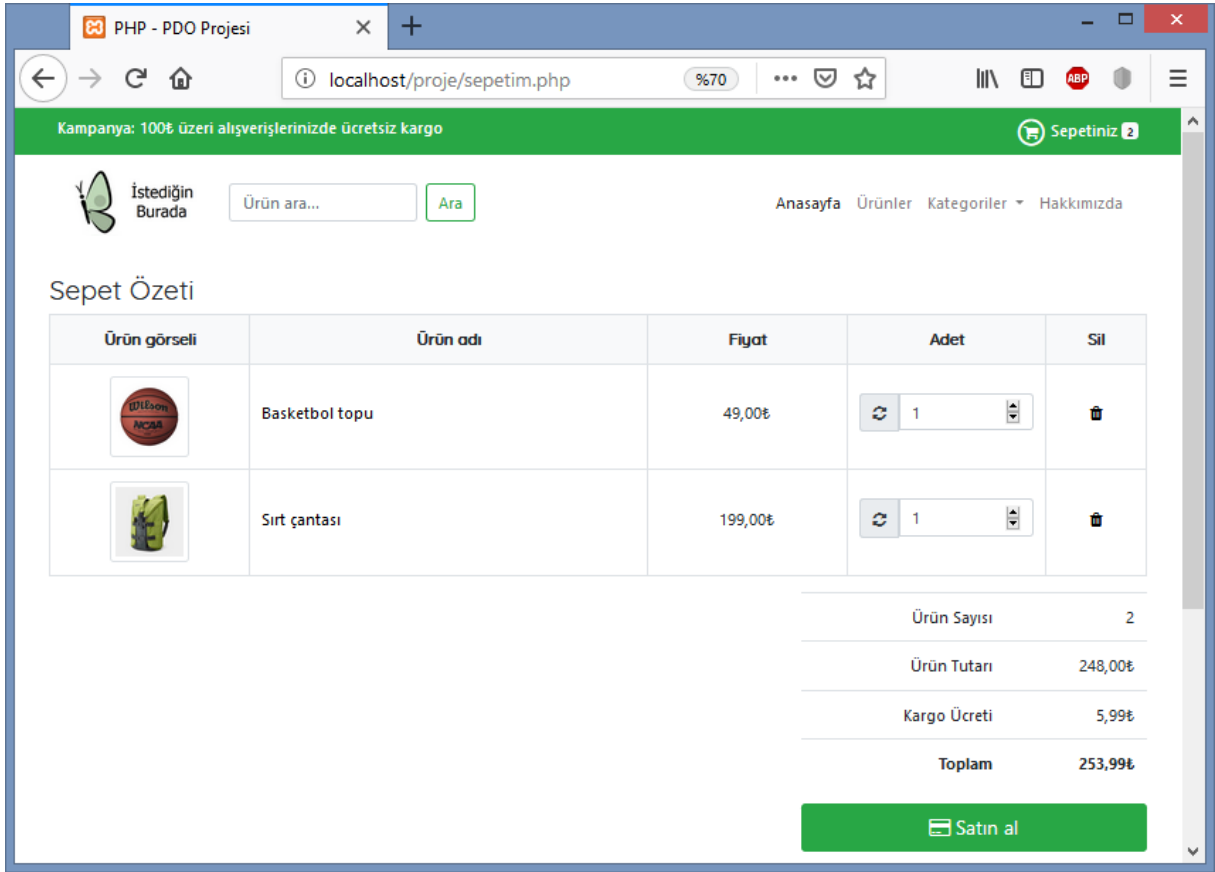
```

.link2{
    text-decoration: none;
    color: black;
}
.link2:hover{
    text-decoration: none;
    color: grey;
}

.sepet-tablo > tbody > tr > td {
    text-align: center;
    vertical-align: middle;
}
.sepet-tablo > thead > tr > th {
    text-align: center;
    font-family: 'Quicksand', sans-serif;
}

```

Tarayıcınızın adres satırına **http://localhost/proje/urunler.php** yazarak listelenen birkaç ürünün **Sepete Ekle** butonunu tıklayın ve sepete ekleyin. Ardından duyuru satırındaki **Sepetiniz** butonunu tıklayın.



ÜRÜN GÜNCELLE BUTONU KODLARINI OLUŞTURMAK

JAVASCRIPT FONKSİYONUNU OLUŞTURMAK

footer.php sayfa kodlarını açın ve // `urun-guncelle` butonu ile tetiklenen fonksiyon burada yer alacak açıklama satırının yerine aşağıdaki JavaScript fonksiyonunu yazın.

```
// urun-guncelle butonu ile tetiklenir
$('.urun-guncelle').on('click', function(){

    var id = $(this).attr('id');
    var adet = document.getElementById('urun_'+id).value;

    // id ve adet parametreleri sepeti_guncelle.php sayfasına yönlendirilir
    $.ajax ({
        cache: false,
        type: 'POST',
        url: 'sepeti_guncelle.php',
        data: {id:id, adet:adet},
        success: function(){
            swal("Ürün adedi güncellendi!", {
                icon: "success",
                buttons: false,
                timer: 2000,
            }).then(function() {
                window.location.href="sepetim.php";
            });
        },
        error: function(jqXHR, textStatus, errorThrown){
            alert(textStatus + errorThrown);
        }
    });
    return false;
});
```

AJAX SAYFASINI (SEPETI_GUNCELLE.PHP) OLUŞTURMAK

urundetay.php sayfasında **Güncelle** butonu tıklanınca fonksiyona gelen **id** ve **adet** parametre bilgilerine göre ilgili ürünün miktarını güncelleyen kodları **Ajax** metodu ile oluşturacağız. Bunun için **proje** klasöründe **sepeti_guncelle.php** isimli bir dosya oluşturun ve içine aşağıdaki PHP kodlarını yazın.

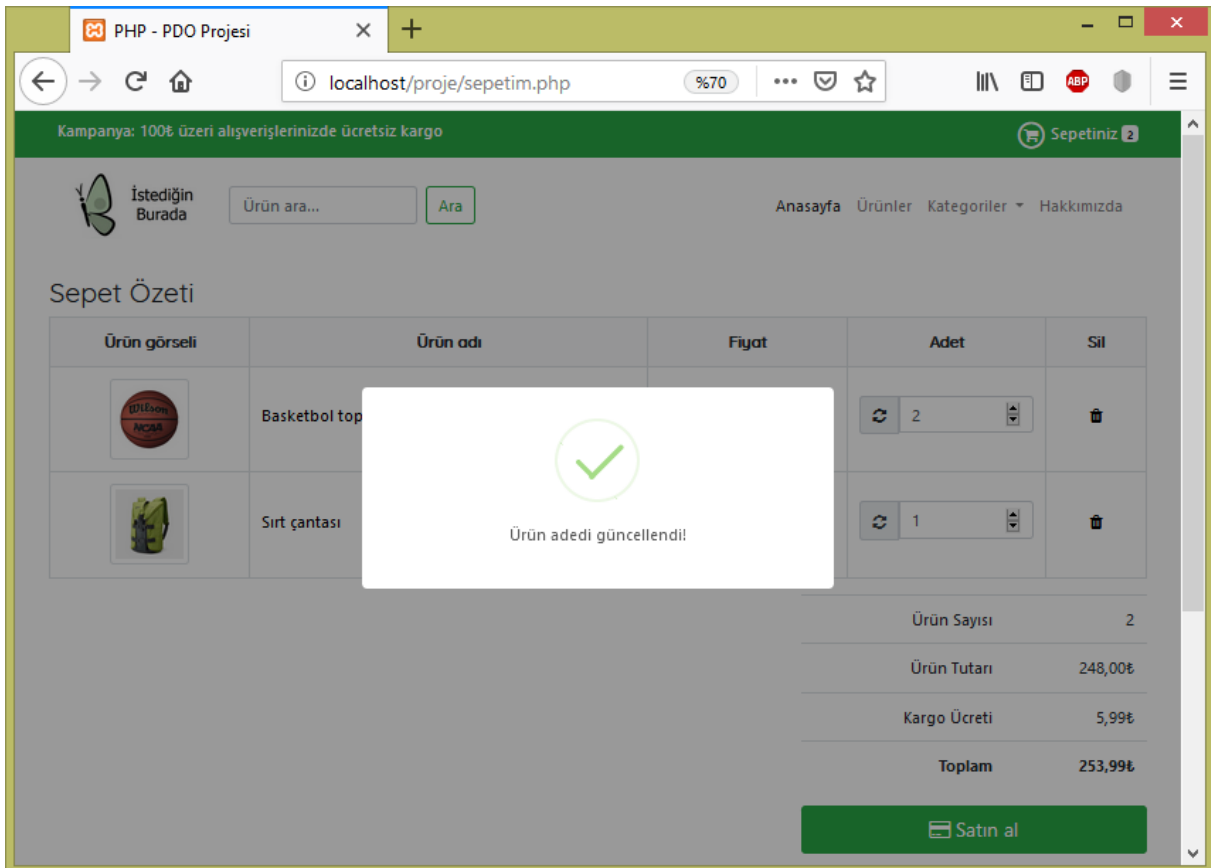
```
<?php
// oturum işlemlerini başlat
session_start();

// ürün id ve adet bilgilerini al
$id = isset($_POST['id']) ? $_POST['id'] : "";
$adet = isset($_POST['adet']) ? $_POST['adet'] : "";

// ürünü sepetten sil
unset($_SESSION['sepet'][$id]);

if($adet <> "") {
// ürünü güncel adet bilgisiyle kaydet
$_SESSION['sepet'][$id]=array('adet'=>$adet);
}
?>
```

Tarayıcınızın adres satırına **http://localhost/proje/urunler.php** yazarak listelenen bir ürünün **Sepete Ekle** butonunu tıklayın ve sepete ekleyin. Ardından duyuru satırındaki **Sepetiniz** butonunu tıklayın ve daha önce sepete eklediğiniz bir ürünün miktarını değiştirip **Güncelle** butonuna tıklayın.







Kampanya: 100% üzeri alışverişlerinizde ücretsiz kargo

İstediğin Burada

Ürün ara... Ara

Anasayfa Ürünler Kategoriler Hakkımızda

Sepet Özeti

Ürün görseli	Ürün adı	Fiyat	Adet	Sil
	Basketbol topu		2	
	Sırt çantası		1	

Ürün Sayısı 2

Ürün Tutarı 248,00₺

Kargo Ücreti 5,99₺

Toplam 253,99₺

Satın al

ÜRÜN SİL BUTONU KODLARINI OLUŞTURMAK

JAVASCRIPT FONKSİYONUNU OLUŞTURMAK

footer.php sayfa kodlarını açın ve // urun-sil butonu ile tetiklenen fonksiyon burada yer alacak açıklama satırının yerine aşağıdaki JavaScript fonksiyonunu yazın.

```
// urun-sil butonu ile tetiklenir
$('.urun-sil').on('click', function(){

    var id = $(this).attr('id');

    // id parametresi sepeti_guncelle.php sayfasına yönlendirilir
    $.ajax ({
        cache: false,
        type: 'POST',
        url: 'sepeti_guncelle.php',
        data: {id:id},
        success: function(){
            swal("Ürün sepetten çıkarıldı!", {
                icon: "success",
                buttons: false,
                timer: 2000,
            }).then(function() {
                window.location.href="sepetim.php";
            });
        },
        error: function(jqXHR, textStatus, errorThrown){
            alert(textStatus + errorThrown);
        }
    });
    return false;
});
```

AJAX SAYFASINI OLUŞTURMAK

Ürün silme butonu fonksiyonu için ayrı bir Ajax dosyası oluşturmayacağız, **sepeti_guncelle.php** dosyası kullanılacak.

SATINALMA SAYFASINI OLUŞTURMAK

Sepetteki ürünleri listeleen sayfada **Satın al** butonu tıklanınca sipariş özeti, teslimat adresi bilgileri ve ödeme seçeneklerini görüntüleyen sayfa açılır. **proje** klasöründe **satinal.php** isimli bir dosya oluşturun ve içine aşağıdaki kodları yazın.

```
<?php include "header.php";

if($_POST) {
    $urun_tutari = $_POST["uruntutari"];
    $kargo_ucreti = $_POST["kargo"];
    $toplam = $_POST["toplam"];
}
?>

<div class="container mt-4 mb-5">
    <form name="form" method="post" action="siparis_onay.php">
        <div class="row">
            <div class="col-sm-7">
                <!-- Adres bilgi formu burada yer alacak -->
            </div>
            <div class="col-sm-5">
                <div>
                    <!-- Sipariş özet bilgileri burada yer alacak -->
                </div>
                <div>
                    <!-- Ödeme seçenekleri burada yer alacak -->
                </div>
            </div>
        </div>
    </form>
</div>
```

```

        </div>
    </div>
</form>
</div>
<?php include "footer.php"; ?>

```

<!-- Adres bilgi formu burada yer alacak --> açıklama satırının yerine aşağıdaki HTML kodları yazın.

```

<div class="baslik">
    <h3>Teslimat Adresi</h3>
    <hr>
</div>
<div class="row">
    <div class="col-sm-6">
        <div class="form-group">
            <label for="isim">İsim*</label>
            <input type="text" id="isim" class="form-control"
placeholder="Adınız">
        </div>
    </div>
    <div class="col-sm-6">
        <div class="form-group">
            <label for="soyisim">Soyisim*</label>
            <input type="text" id="soyisim" class="form-control"
placeholder="Soyadınız">
        </div>
    </div>
</div>
<div class="row">
    <div class="col-sm-6">
        <div class="form-group">
            <label for="il">İl*</label>
            <select class="form-control" id="il">
                <option>Adana</option>
                <option>Ankara</option>
                <option>Bursa</option>
                <option>Edirne</option>
                <option>İstanbul</option>
                <option>Tekirdağ</option>
                <option>Zonguldak</option>
            </select>
        </div>
    </div>
    <div class="col-sm-6">
        <div class="form-group">
            <label for="ilce">İlçe</label>
            <input type="text" id="ilce" class="form-control"
placeholder="İlçe adı">
        </div>
    </div>
</div>
<div class="row">
    <div class="col-sm-12">
        <div class="form-group">
            <label for="adres">Adres*</label>
            <textarea id="adres" class="form-control" placeholder="Mahalle,
Cadde/Sokak, Kapı no..."></textarea>
        </div>
    </div>
</div>
<div class="row">
    <div class="col-sm-6">

```

```

        <div class="form-group">
            <label for="kurum">Kurum/Şirket</label>
            <input type="text" id="kurum" class="form-control"
placeholder="Kurum veya Şirket adı">
        </div>
    </div>
    <div class="col-sm-6">
        <div class="form-group">
            <label for="postakodu">Posta Kodu</label>
            <input type="text" id="postakodu" class="form-control"
placeholder="Beş haneli posta kodu">
        </div>
    </div>
</div>
<div class="row">
    <div class="col-sm-6">
        <div class="form-group">
            <label for="eposta">E-posta Adresi*</label>
            <input type="email" id="eposta" class="form-control"
placeholder="aaa@bbb.ccc">
        </div>
    </div>
    <div class="col-sm-6">
        <div class="form-group">
            <label for="telefon">Telefon Numarası*</label>
            <input type="text" id="telefon" class="form-control"
placeholder="0111 222 33 44">
        </div>
    </div>
</div>

```

Ardından `<!-- Sipariş özet bilgileri burada yer alacak -->` açıklama satırının yerine aşağıdaki HTML kodları yazın.

```

<div class="baslik mb-3">
    <h3>Sipariş Özeti</h3>
</div>
<div>
    <table class="table">
        <tbody>
            <tr>
                <td>Ürün Tutarı</td>
                <td><?php echo number_format($urun_tutari, 2, ',', '.');
?>&#8378;</td>
            </tr>
            <tr>
                <td>Kargo Ücreti</td>
                <td><?php echo number_format($kargo_ucreti, 2, ',', '.');
?>&#8378;</td>
            </tr>
            <tr>
                <td><strong>Toplam</strong></td>
                <td><strong><?php echo number_format($toplam, 2, ',', '.');
?>&#8378;</strong></td>
            </tr>
        </tbody>
    </table>
</div>

```

Son olarak da `<!-- Ödeme seçenekleri burada yer alacak -->` açıklama satırının yerine aşağıdaki HTML kodları yazın.

```

<div class="baslik">
    <h3>Ödeme Seçenekleri</h3>
    <hr>

```

```

</div>
<ul class="list-unstyled">
  <li>
    <input type="radio" name="odemesekli" id="havale" checked >
    <label for="havale"> Havale/EFT</label>
    <p class="text-secondary">Havale ve EFT ödemelerinizi anlaşmalı banka
    hesaplarına üç gün içinde yapabilirsiniz.</p>
  </li>
  <li>
    <input type="radio" name="odemesekli" id="kart">
    <label for="kart"> Banka/Kredi Kartı</label>
  </li>
  <li>
    <input type="radio" name="odemesekli" id="kapida">
    <label for="kapida"> Kapıda Ödeme</label>
  </li>
</ul>
<p>
  <input type="checkbox" name="onay" id="onay">
  <label for="onay"><a href="#">Satış Sözleşmesi</a>ni onaylıyorum.</label>
</p>
<div class="text-center">
  <hr>
  <button type="submit" class="btn btn-success btn-lg btn-block">
    <i class="fa fa-turkish-lira"></i> Ödemeyi tamamla
  </button>
</div>

```

Sepeti oluşturup **Satın al** butonuyla sayfayı görüntüleyin.

The screenshot shows a web browser window with the URL 'localhost/proje/satinal.php'. The page features a green header with a campaign message 'Kampanya: 100₺ üzeri alışverişlerinizde ücretsiz kargo' and a shopping cart icon labeled 'Sepetiniz 2'. The main content is divided into two columns. The left column is titled 'Teslimat Adresi' and contains form fields for 'İsim*' (Adınız, Soyadınız), 'İl*' (Adana), 'İlçe' (İlçe adı), 'Adres*' (Mahalle, Cadde/Sokak, Kapı no...), 'Kurum/Şirket' (Kurum veya Şirket adı), 'Posta Kodu' (Beş haneli posta kodu), 'E-posta Adresi*' (aaa@bbb.ccc), and 'Telefon Numarası*' (0111 222 33 44). The right column is titled 'Sipariş Özeti' and contains a table with the following data:

Ürün Tutarı	297,00₺
Kargo Ücreti	5,99₺
Toplam	302,99₺

Below the summary is the 'Ödeme Seçenekleri' section, which includes radio buttons for 'Havale/EFT' (selected), 'Banka/Kredi Kartı', and 'Kapıda Ödeme', and a checkbox for 'Satış Sözleşmesini onaylıyorum.'. A green button labeled 'Ödemeyi tamamla' is at the bottom right.

SİPARİŞ ONAYI SAYFASINI OLUŞTURMAK

Satın alma sayfasında **Ödemeyi tamamla** butonu tıklanınca bir uyarı mesajıyla bilgi verip sepeti boşaltacak kodları **proje** klasöründe oluşturacağınız **siparis_onay.php** dosyasına yazın.

```
<?php
```

```
// oturum işlemlerini başlat
session_start();

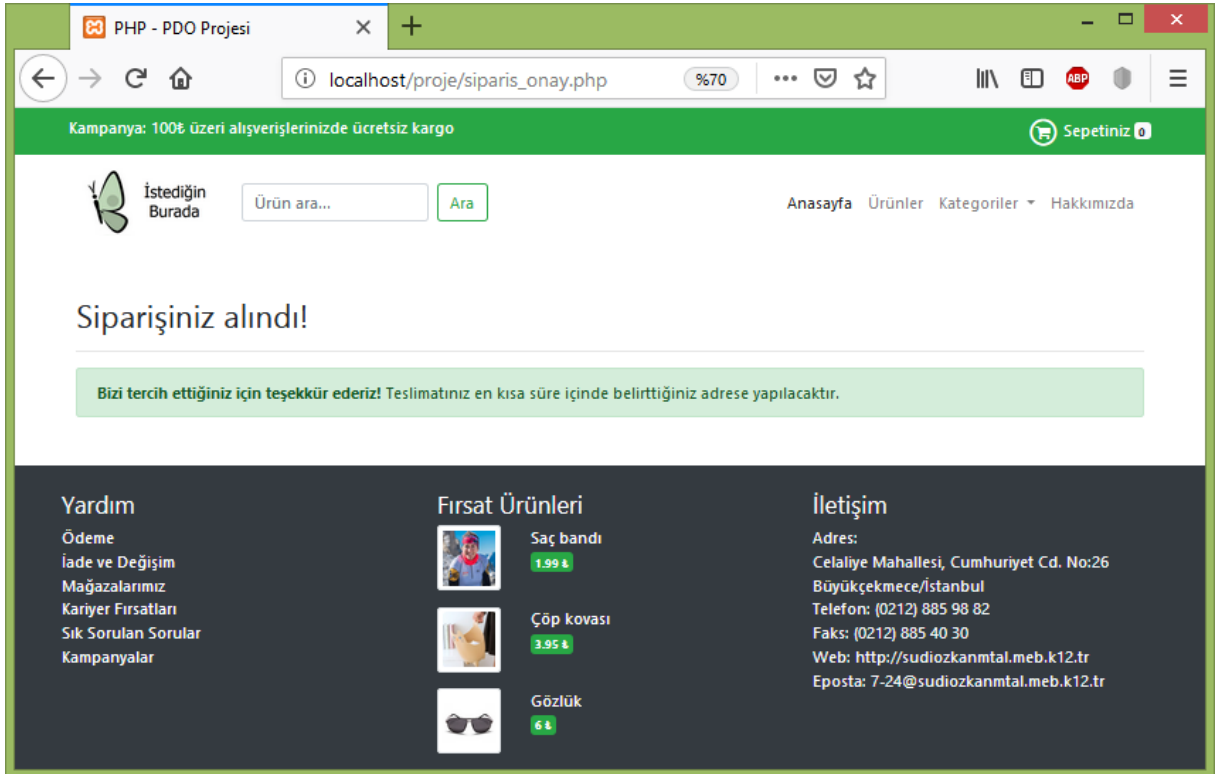
// sepeti boşalt
session_destroy();

include "header.php";
?>

<div class="container mt-5 mb-5">
  <div class="col-md-12">
    <h2>Siparişiniz alındı!</h2>
    <hr>
    <!-- tell the user order has been placed -->
    <div class="alert alert-success">
      <strong>Bizi tercih ettiğiniz için teşekkür ederiz!</strong> Teslimatınız en
      kısa süre içinde belirttiğiniz adrese yapılacaktır.
    </div>
  </div>
</div>

<?php
include "footer.php";
?>
```

Önceki aşamada **Ödemeyi tamamla** butonuna tıklayıp sayfayı görüntüleyin.



Kampanya: 100% üzeri alışverişlerinizde ücretsiz kargo

İstediklerin Burada

Ürün ara... Ara

Anasayfa Ürünler Kategoriler Hakkımızda

Siparişiniz alındı!

Bizi tercih ettiğiniz için teşekkür ederiz! Teslimatınız en kısa süre içinde belirttiğiniz adrese yapılacaktır.

Yardım

- Ödeme
- İade ve Değişim
- Mağazalarımız
- Kariyer Fırsatları
- Sık Sorulan Sorular
- Kampanyalar

Fırsat Ürünleri

- Saç bandı **1.99 ₺**
- Çöp kovası **3.95 ₺**
- Gözlük **6 ₺**

İletişim

Adres:
Celaliye Mahallesi, Cumhuriyet Cd. No:26
Büyükkçekmece/İstanbul
Telefon: (0212) 885 98 82
Faks: (0212) 885 40 30
Web: <http://sudiozkanmtal.meb.k12.tr>
Eposta: 7-24@sudiozkanmtal.meb.k12.tr

EK GÖREVLER

- Ürünler (urunler.php) sayfasında kayıtlar için sayfalandırma (pagination) yapın.
- Anasayfadaki (index.php) ürün kartlarına **Sepete Ekle** butonu ekleyin.