

PHP – PDO

(İNTERNET PROGRAMCILIĞI DERSİ İÇİN)

ADIM ADIM DİNAMİK SİTE TASARLAMA



2018

Mercin KARAKAŞ – Murat SARIÖZ

İçindekiler

GİRİŞ	1
PROJENİN DOSYA VE DİZİN YAPISI.....	2
VERİTABANINI HAZIRLAMAK	3
VERİTABANINI OLUŞTURMAK.....	3
VERİTABANI BAĞLANTISI İÇİN KULLANILACAK PHP KOD DOSYASINI OLUŞTURMAK	6
YÖNETİM PANELİNİ OLUŞTURMAK	8
ARAYÜZ TASARIMI İÇİN GEREKLİ DOSYALARIN KURULUMU.....	8
WEB SAYFALARININ ANA HTML ÇATISINI OLUŞTURMAK	9
CSS STİL DOSYASINI (STYLE.CSS) OLUŞTURMAK.....	9
BAŞLIK (HEADER.PHP) SAYFASINI OLUŞTURMAK.....	9
ALTBİLGİ (FOOTER.PHP) SAYFASINI OLUŞTURMAK.....	10
YÖNETİM PANELİ ANA SAYFASINI (INDEX.PHP) OLUŞTURMAK	11
ÜRÜN SAYFALARINI OLUŞTURMAK	12
ÜRÜN LİSTELEME SAYFASINI OLUŞTURMAK	12
ÜRÜN SİLME SAYFASINI OLUŞTURMAK.....	14
ÜRÜN DETAY SAYFASINI OLUŞTURMAK	16
ÜRÜN EKLEME SAYFASINI OLUŞTURMAK.....	18
ÜRÜN GÜNCELLEME (DÜZELTME) SAYFASINI OLUŞTURMAK	20
SAYFALARI NUMARALAMA İŞLEMİ.....	24
SAYFALAMA DEĞİŞKENLERİNİ TANIMLAMAK	24
SEÇME SORGUSUNU DÜZENLEMEK	24
KAYIT SAYISINI HESAPLAMAK	24
SAYFALAMA DOSYASINI (SAYFALAMA.PHP) OLUŞTURMAK.....	25
RESİM YÜKLEMESİ.....	29
VERİTABANINA YENİ BİR ALAN EKLEMESİ	29
ÜRÜN EKLEME SAYFASINI DÜZENLEMESİ	30
ÜRÜN EKLEME SORGUSUNU DÜZENLEMESİ	30
DOSYA YÜKLEME VE KAYDETME İŞLEMİ	31
ÜRÜN DETAY SAYFASINDA YÜKLENEN RESMİ GÖRÜNTÜLEMESİ	33
ÜRÜN SİLİNDİĞİNDE YÜKLENEN RESMİ SUNUCUDAN SİLMEK	34
ARAMA İŞLEMİ.....	36
ARAMA FORMUNU OLUŞTURMAK.....	36
SEÇME SORGUSUNU DÜZENLEMEK	37
ARAMA VE SAYFALANDIRMA İŞLEMLERİNİ BİRLEŞTİRMESİ	37
BUTONLARIN GÖRSELLİĞİNİ ZENGİNLEŞTİRMEK	39
ÜRÜN LİSTE SAYFASINDAKİ BUTONLARA SİMGE EKLEMESİ.....	39
ÜRÜN DETAY SAYFASINDAKİ BUTONLARA SİMGE EKLEMESİ	39
ÜRÜN EKLEME SAYFASINDAKİ BUTONLARA SİMGE EKLEMESİ	40

ÜRÜN GÜNCELLEME SAYFASINDAKİ BUTONLARA SİMGE EKLEMELİ	41
ÇOKLU ÜRÜN SİLME İŞLEVİ	42
ÇOKLU ÜRÜN SEÇME KUTULARINI (CHECKBOX) OLUŞTURMAK	42
AJAX KODLARINI OLUŞTURMAK	43
ÇOKLU ÜRÜN SİLME SAYFASINI (COKLUSIL.PHP) OLUŞTURMAK	44
KATEGORİ SAYFALARINI OLUŞTURMAK	46
KATEGORİ LİSTELEME SAYFASINI OLUŞTURMAK	47
KATEGORİ SİLME SAYFASINI OLUŞTURMAK	49
KATEGORİ DETAY SAYFASINI OLUŞTURMAK	51
KATEGORİ EKLEME SAYFASINI OLUŞTURMAK	52
KATEGORİ GÜNCELLEME (DÜZELTME) SAYFASINI OLUŞTURMAK	54
ÜRÜNLER TABLOSUYLA KATEGORİLER TABLOSUNU İLİŞKİLENDIRMEK	58
ÜRÜN EKLEME SAYFASINA KATEGORİ ALANI EKLEMELİ	59
ÜRÜN GÜNCELLEME SAYFASINA KATEGORİ ALANI EKLEMELİ	60
ÜRÜN LİSTELEME SAYFASINA KATEGORİ ALANI EKLEMELİ	63
ÜRÜN DETAY SAYFASINA KATEGORİ ALANI EKLEMELİ	64
KATEGORİ SAYFASINA ÜRÜN LİSTESİ EKLEMELİ	65
KULLANICI SAYFALARINI OLUŞTURMAK	68
ANA SAYFADA KULLANICI MENÜSÜNÜ OLUŞTURMAK	69
KULLANICI LİSTELEME SAYFASINI OLUŞTURMAK	69
KULLANICI SİLME SAYFASINI OLUŞTURMAK	71
KULLANICI EKLEME SAYFASINI OLUŞTURMAK	73
KULLANICI GÜNCELLEME (DÜZELTME) SAYFASINI OLUŞTURMAK	75
OTURUM YÖNETİMİ	79
SAYFALARDA OTURUM KONTROLÜNÜ YAPMAK	79
OTURUM AÇMA SAYFASINI (LOGIN.PHP) OLUŞTURMAK	79
KULLANICI KONTROLÜ VE OTURUM AÇMA İŞLEMİ	80
OTURUMU KAPATMAK	81
EK GÖREVLER	83
KULLANICI ARAYÜZÜNÜ OLUŞTURMAK	1
WEB SAYFALARININ ANA HTML ÇATISINI OLUŞTURMAK	2
CSS STİL DOSYASINI (STYLE.CSS) DÜZENLEMELİ	3
BAŞLIK (HEADER.PHP) SAYFASINI OLUŞTURMAK	4
ALTBİLGİ (FOOTER.PHP) SAYFASINI OLUŞTURMAK	7
ANA SAYFAYI (INDEX.PHP) OLUŞTURMAK	9
SLIDER SATIRINI OLUŞTURMAK	9
KARŞILAMA SATIRINI OLUŞTURMAK	11
FIRSAT SATIRINI OLUŞTURMAK	11
YENİ ÜRÜN SATIRINI OLUŞTURMAK	12
ÜRÜNLER SAYFASINI (URUNLER.PHP) OLUŞTURMAK	14

BANNER SATIRINI OLUŞTURMAK	15
KATEGORİ LISTESİNİ OLUŞTURMAK	15
SIRALAMA SEÇENEKLERİNİ OLUŞTURMAK	17
FİYAT SEÇENEKLERİNİ OLUŞTURMAK	18
ÜRÜN LİSTESİSİ OLUŞTURMAK	19
ÜRÜN DETAY SAYFASINI (URUNDETAY.PHP) OLUŞTURMAK.....	21
ÜRÜN BİLGİLERİ LISTELEMEN İÇİN GEREKLİ SORGUYU OLUŞTURMAK	22
ÜRÜN RESMİNİ GÖRÜNTÜLEMEK	22
ÜRÜN BİLGİLERİ GÖRÜNTÜLEMEK	22
DİĞER ÜRÜNLERİ LISTELEMEN	23
HAKKIMIZDA SAYFASINI (HAKKIMIZDA.PHP) OLUŞTURMAK	25
TANITIM YAZISI İLE İLETİŞİM FORMUNU OLUŞTURMAK	25
E-POSTA GÖNDERME SAYFASINI OLUŞTURMAK.....	26
ALIŞVERİŞ SEPETİNİ OLUŞTURMAK.....	28
SEPET OTURUMUNU OLUŞTURMAK	28
SEPETE EKLE BUTONU KODLARINI OLUŞTURMAK	28
SEPETTEKİ ÜRÜNLERİ GÖRÜNTÜLEMEK.....	31
ÜRÜN GÜNCELLE BUTONU KODLARINI OLUŞTURMAK.....	35
ÜRÜN SİL BUTONU KODLARINI OLUŞTURMAK	36
SATINALMA SAYFASINI (SATINAL.PHP) OLUŞTURMAK	37
SİPARİŞ ONAYI SAYFASINI (SIPARIS_ONAY.PHP) OLUŞTURMAK	40
EK GÖREVLER.....	42
WEB SİTESİNİ YAYIMLAMAK.....	43
LINKLERİ DÜZENLEMEN	43
VERİTABANI BAĞLANTI PARAMETRELERİNİ AYARLAMAK.....	43
RESİM KLASÖRÜNE YAZMA İZNİ VERMEK	43
VERİTABANI İŞLEMLERİNDE SINIF KULLANIMI (OOP).....	1
VERİTABANI BAĞLANTISI İÇİN KULLANILACAK SINIF YAPISINI OLUŞTURMAK	2
OTURUM AÇMA İŞLEMİNİ NESNEYE YÖNELİK DÜZENLEMEN	2
KULLANICILAR TABLOSUNUN SINIF YAPISINI OLUŞTURMAK	2
OTURUM AÇMA SAYFASINI DÜZENLEMEN.....	3
ÜRÜN SAYFALARINDAKİ VERİTABANI İŞLEMLERİNİ NESNEYE YÖNELİK DÜZENLEMEN.....	4
URUNLER TABLOSUNUN SINIF YAPISINI OLUŞTURMAK.....	4
KATEGORİLER TABLOSUNUN SINIF YAPISINI OLUŞTURMAK	7
SAYFALAR İÇİN ORTAK SINIF DOSYALARINI EKLEMEN	7
ÜRÜN LİSTELEMEN SAYFASINI DÜZENLEMEN	8
ÜRÜN SİLME SAYFASINI DÜZENLEMEN	9
ÇOKLU ÜRÜN SİLME SAYFASINI DÜZENLEMEN	10
ÜRÜN DETAY SAYFASINI DÜZENLEMEN.....	11
ÜRÜN EKLEMEN SAYFASINI DÜZENLEMEN	12

ÜRÜN GÜNCELLEME (DÜZELTME) SAYFASINI DÜZENLEMEK.....	14
EK GÖREVLER.....	17

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. Bir e-ticaret sitesi olacak uygulamamızı aşama aşama geliştireceğiz.

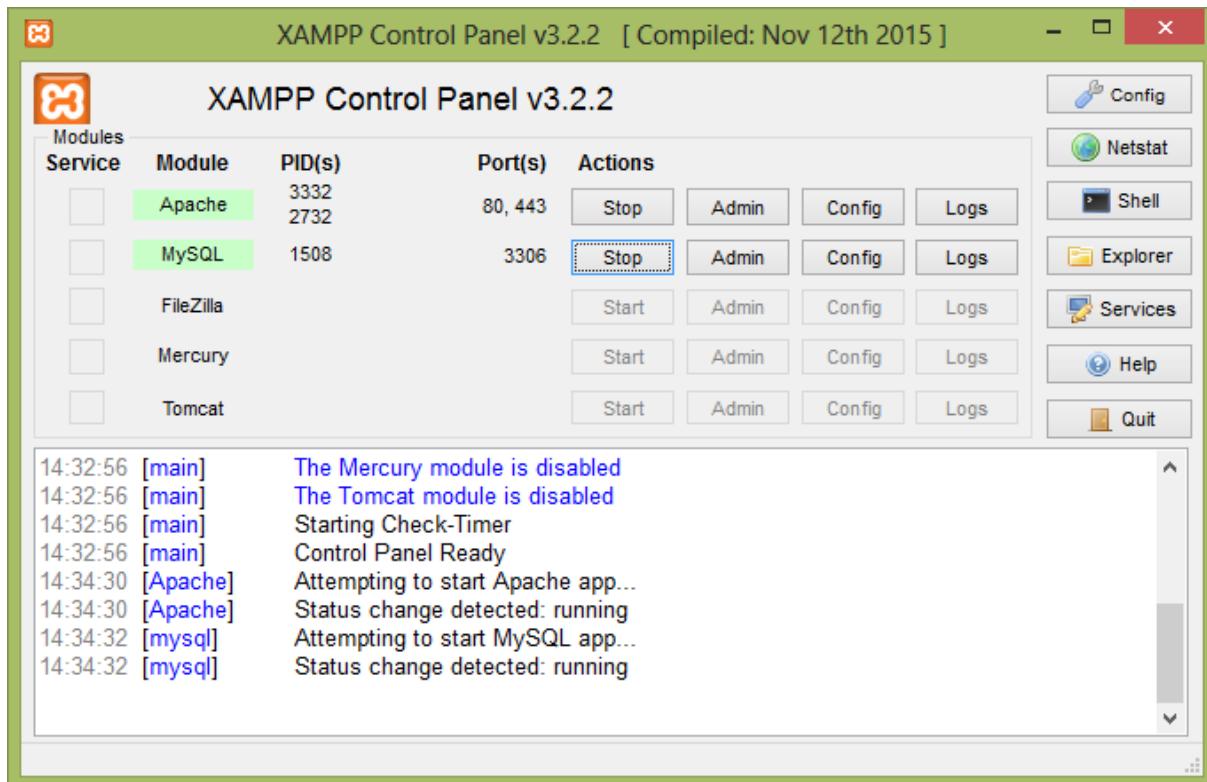
PROJENİN DOSYA VE DİZİN YAPISI

PDO CRUD uygulamamız başlıca şu dosyalardan oluşacak:

- **config/vtabani.php** – veritabanı bağlantısı ve ayarları için kullanılacak.
- **admin/header.php** – yönetici sayfalarının başlık bilgilerini içeren sayfa.
- **admin/footer.php** – yönetici 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** – resimleri barındıran klasör.
- **header.php** – web sayfalarının başlık bilgilerini içeren sayfa.
- **footer.php** – web sayfalarının alt bilgilerini içeren sayfa.
- **index.php** – web sitesinin ana sayfası.
- **urunler.php** – ürün bilgilerinin listelendiği, filtrelentiği, sıralandığı, sepete eklendiği sayfa.
- **urundetay.php** – seçili bir ürünün ayrıntılı bilgileri görüntüleyen sayfa. Ürün sepete istenen miktarda eklenebilir.
- **hakkimizda.php** – site tanıtıcı yazının ve iletişim formunun yer aldığı sayfa. Kullanıcının eposta göndermesine izin verir.
- **eposta.php** – eposta göndermek için kullanılacak yardımcı sayfa.
- **sepetim.php** – sepete eklenen ürünler gösteren sayfa. Burada sepetteki ürün miktarı değiştirilebilir, sepetteki ürünler silinebilir. Ayrıca bu sayfada sepet tutarı da görüntülenir.
- **sepete_ekle.php** – sepete ürün eklemek için kullanılacak Ajax sayfası.
- **sepeti_guncelle.php** – sepetteki ürün miktarını değiştirmek ve ürün silmek için kullanılacak Ajax sayfası.
- **satinal.php** – satın alma sayfasında sepet özeti gösterilip müşteriden adres bilgisi ve ödeme seçeneklerini girmesi istenir.
- **siparis_onay.php** – müşteriye siparişin alındığı mesajı verilip sepet boşaltılır.
- **config/vt_oop.php** – veritabanı bağlantısı ve ayarları için kullanılacak sınıf.
- **config/urun.php** – urunler tablosu sınıfı.
- **config/kategori.php** – kategoriler tablosu sınıfı.
- **config/kullanici.php** – kullanıcılar tablosu sınıfı.

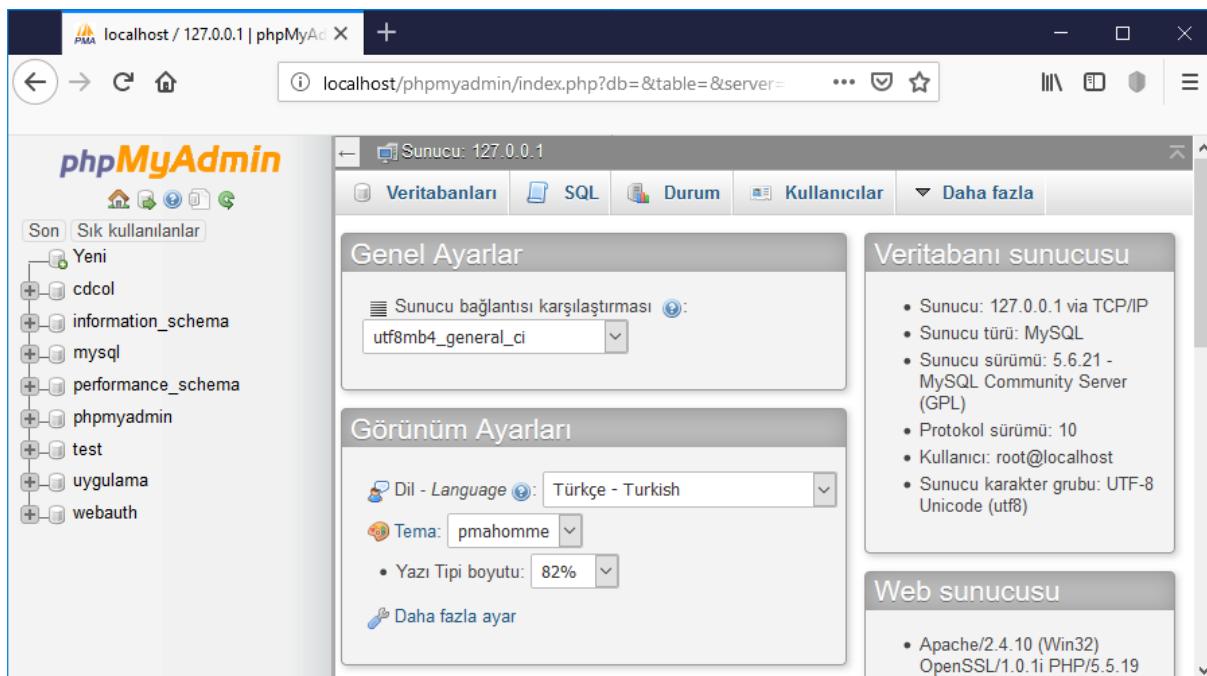
VERİTABANINI HAZIRLAMAK

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.

The screenshot shows the phpMyAdmin interface on a browser window. The left sidebar lists databases: Yeni, cdcol, information_schema, mysql, performance_schema, phpmyadmin, test, uygulama, and webauth. The main area has tabs for Veritabanları, SQL, Durum, Kullanıcılar, and Daha fazla. The SQL tab is active, displaying the following SQL code:

```

1 CREATE DATABASE php_pdo_projesi DEFAULT CHARACTER SET utf8 COLLATE
utf8_turkish_ci;
2 USE php_pdo_Projesi;
3 CREATE TABLE IF NOT EXISTS `urunler`(
4     `id` INT(11) NOT NULL AUTO_INCREMENT,
5     `urunadi` VARCHAR(128) NOT NULL,
6     `aciklama` TEXT NOT NULL,
7     `fiyat` DOUBLE NOT NULL,
8     `giris_tarihi` DATETIME NOT NULL,
9     `dzltm_tarihi` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
10    PRIMARY KEY(`id`)
11 ) ENGINE = InnoDB DEFAULT CHARSET = utf8 AUTO_INCREMENT = 9;
12

```

Below the code, there's a 'Temizle' (Clear) button and a note: 'Bu SQL sorgusunu işaretle:'. At the bottom, there are buttons for 'Sınırlayıcı' (Limits), a search bar, and checkboxes for 'Bu sorguyu burada tekrar göster' (Repeat this query here) and 'Sorgu kutusunu tut' (Keep query in box). A 'Git' button is also present.

```

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,
    `dzltm_tarihi` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
    PRIMARY KEY(`id`)
) ENGINE = InnoDB DEFAULT CHARSET = utf8 AUTO_INCREMENT = 9;

```

Bilgi:

- INT() – Yanlızca tamsayı değerleri tutar. Görüntülenecek 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 verİYE izin vermez. O yüzden çoğu tablolarda Id alanlarına tanımlanır.

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

The screenshot shows the phpMyAdmin interface. On the left, the database structure is displayed under 'Veritabanları'. It includes the 'phpmyadmin' schema and a user-defined schema 'php_pdo_projesi'. Within 'php_pdo_projesi', there is a table named 'urunler'. A 'Create Database' dialog is open in the center, titled 'Veritabanı oluştur'. It has a dropdown menu 'Veritabanı adı' set to 'Kаршилаштма' (Comparison) and a 'Create' button. Below the dialog, a note says: 'Not: Buradaki veritabanı istatistiklerini etkinleştirmek web sunucusu ile MySQL sunucusu arasında yüksek trafiğe yol açabilir.' (Note: Enabling statistics for this database may cause high traffic between the web server and the MySQL server). A table titled 'Veritabanı' lists various databases with their collation and character set.

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

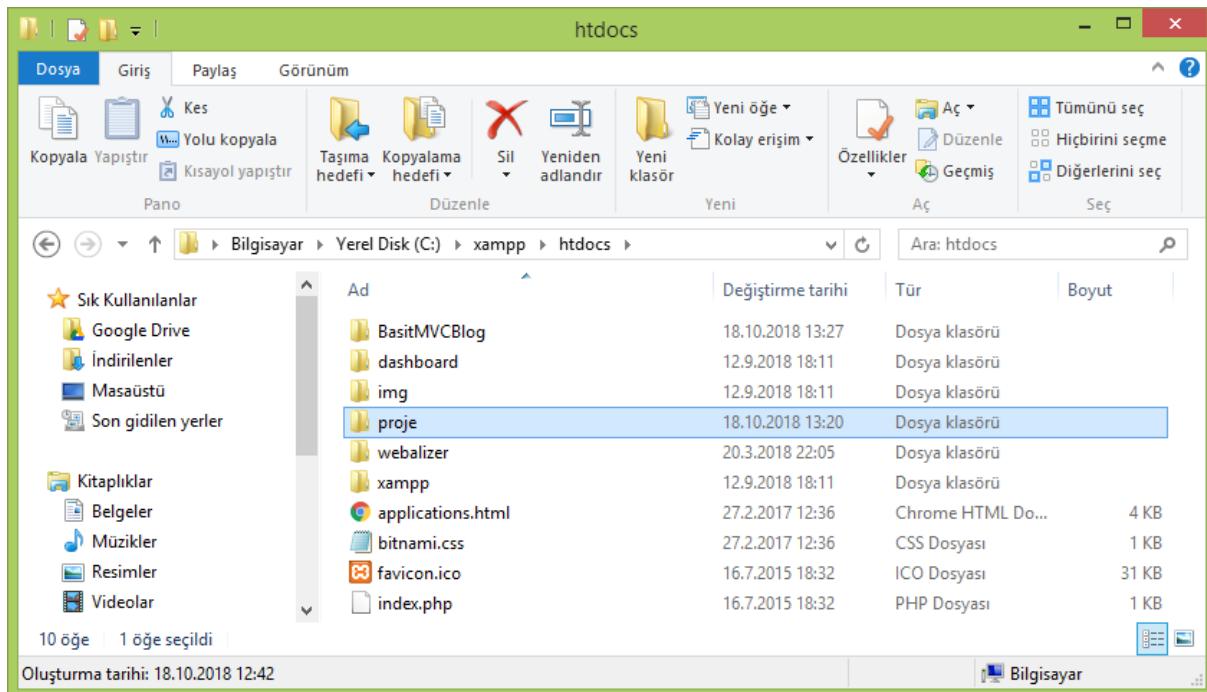
```
USE php_pdo_projesi;

INSERT INTO `urunler` ( `id`, `urunadi`, `aciklama`, `fiyat`, `giris_tarihi`, `dzltm_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' );
```

The screenshot shows the phpMyAdmin interface with the SQL tab selected. A query window is open with the title '127.0.0.1 sunucusu üzerinde SQL sorgusunu/sorgularını çalıştır:'. The query text is the same as the one provided above. Below the query window, there is a 'Temizle' (Clear) button and a checkbox 'Bu SQL sorgusunu işaretle:' (Mark this SQL query as executed). At the bottom, there are buttons for 'Sınırlayıcı' (Limit), 'Git' (Go), and checkboxes for 'Bu sorguyu burada tekrar göster' (Show this query again here) and 'Sorgu kutusunu tut' (Keep the query box open).

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 kodlarını 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"));
}
// hata varsa göster
catch(PDOException $exception){
    echo "Bağlantı hatası: " . $exception->getMessage();
}
?>
```

The screenshot shows a Sublime Text window with the following details:

- Title Bar:** C:\xampp\htdocs\proje\config\vtabani.php (proje) - Sublime Text (UNREGISTERED)
- Menu Bar:** File Edit Selection Find View Goto Tools Project Preferences Help
- Folders List:** FOLDERS
proje
config
vtabani.php
content
- Code Editor:** vtabani.php
The code is a PHP script for database connection configuration. It includes variables for host, database name, user, and password, and a try-catch block for PDO exceptions.
- Status Bar:** Line 16, Column 3 | Tab Size: 4 | PHP

```
<?php
// veritabanı bağlantısı için
$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();
}
?>
```

PHP - PDO PROJESİ

(Yönetici Arayüzü)

ADIM ADIM DİNAMİK SİTE TASARLAMA

Mercin KARAKAŞ – Murat SARIÖZ

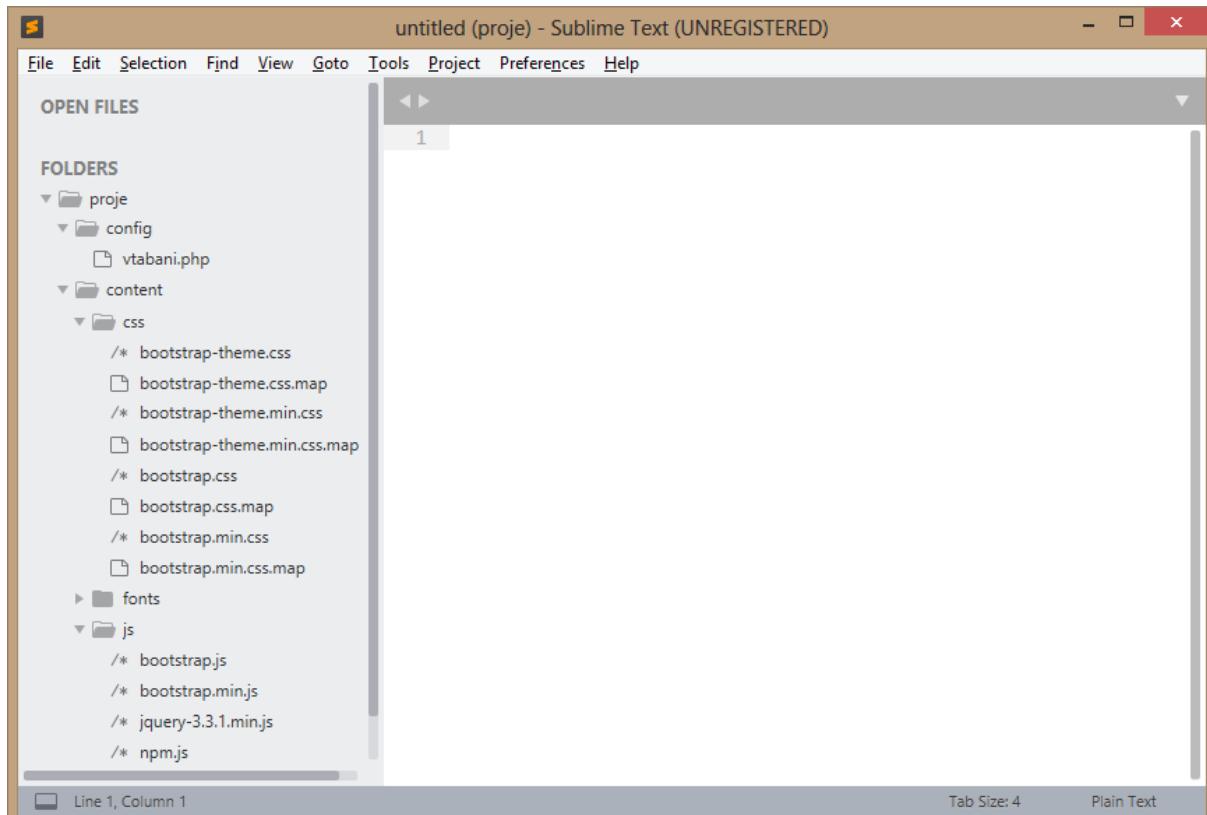
BÜYÜKÇEKMECE SUDİ ÖZKAN MESLEKİ VE TEKNİK ANADOLU LİSESİ

YÖNETİM PANELİNİ OLUŞTURMAK

ARAYÜZ TASARIMI İÇİN GEREKLİ DOSYALARIN KURULUMU

Web sayfalarının arayüzüne tasarlarken **Bootstrap** kütüphanelerinden faydalanağınız. 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: 70px;  
    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 kısmındaki HTML kodlar ortak olduğundan 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">
                <!-- tarayıcıının 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 albilgi nesnemiz olacak. Bunun için yine **Bootstrap** kütüphanesinin **NavBar** bileşenini kullanacağız. Bu ve alt kısmındaki HTML kodlar ortak olduğundan 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">
        © 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 ANA SAYFASINI (INDEX.PHP) OLUŞTURMAK

Web sayfalarımız için ortak olan başlık ve altbilgi alanlarını oluşturduk. Bundan sonra oluşturacağımız sayfalarda bu bilgileri ekleyerek işe başlayacağız. Ana sayfamızı 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"; ?>

```

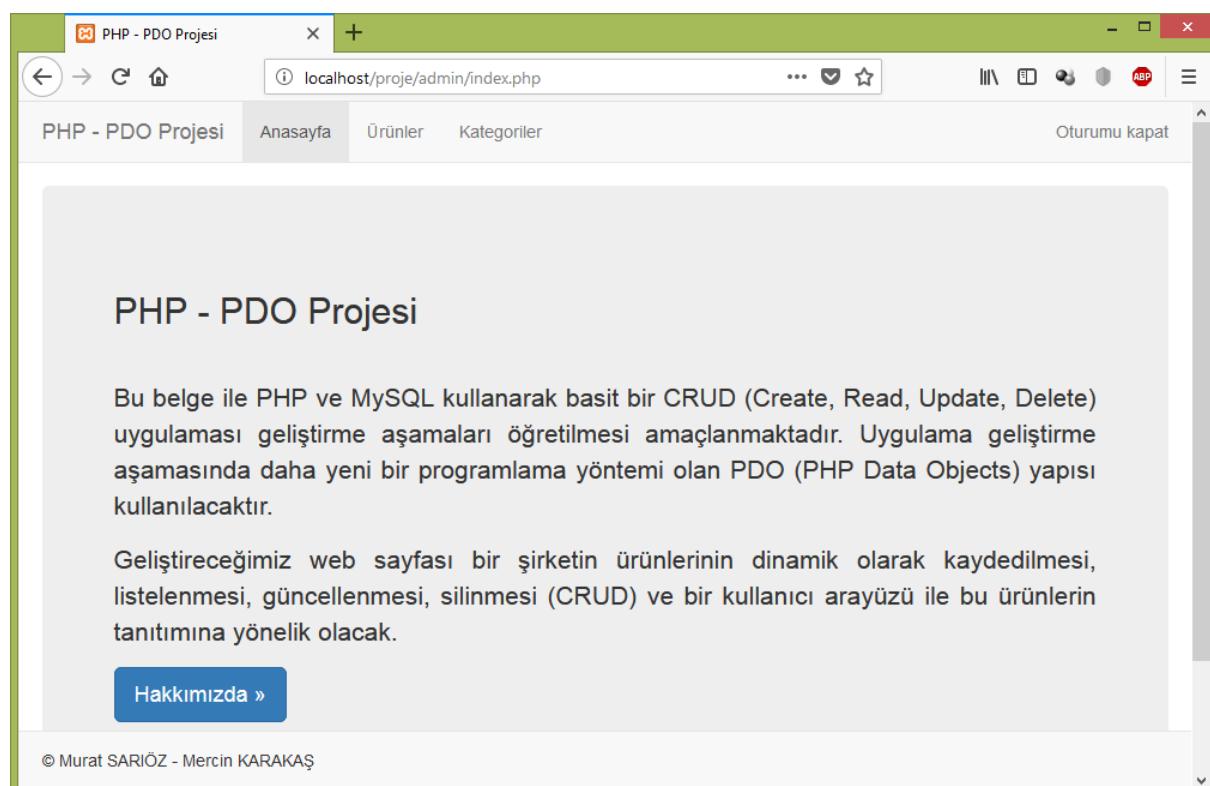
Ardından **container** sınıfı içinde **<!-- Sayfa kodları bu alana eklenecek -->** yazan açıklama satırının yerine aşağıdaki kodları yazın.

```

<!-- Proje hakkında kısa bir bilgi içeren ana sayfa -->
<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 LİSTELEMİR İÇİN GEREKLİ SQL SORGUSUNU OLUŞTURMAK

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

```
<?php
    // silme mesajı burada yer alacak

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

    // 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 liste
    if($sayi>0){

        // kayıtlar burada listelenecek

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

KAYIT LİSTELEMİR 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.

```

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

//tablo başlıklar
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 LİSTELEMEK 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 işaretü
        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'>Düzelt</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 listelemeye sayfasını görüntüleyin.

ID	Ürün adı	Açıklama	Fiyat	İşlem
11	Web kamerası	Görüntülü görüşme için.	20 ₺	<button>Detay</button> <button>Düzenle</button> <button>Sil</button>
10	Hoparlör	Ne kadar büyük o kadar iyi.	50 ₺	<button>Detay</button> <button>Düzenle</button> <button>Sil</button>
9	Klavye	Bilgisayarın vazgeçilmez donanımı.	15 ₺	<button>Detay</button> <button>Düzenle</button> <button>Sil</button>
8	Yastık	İyi bir uyku için.	8.99 ₺	<button>Detay</button> <button>Düzenle</button> <button>Sil</button>

© Murat SARIÖZ - Mercin KARAKAŞ

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

```
<?php

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

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

```

// silme sorusu
$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.

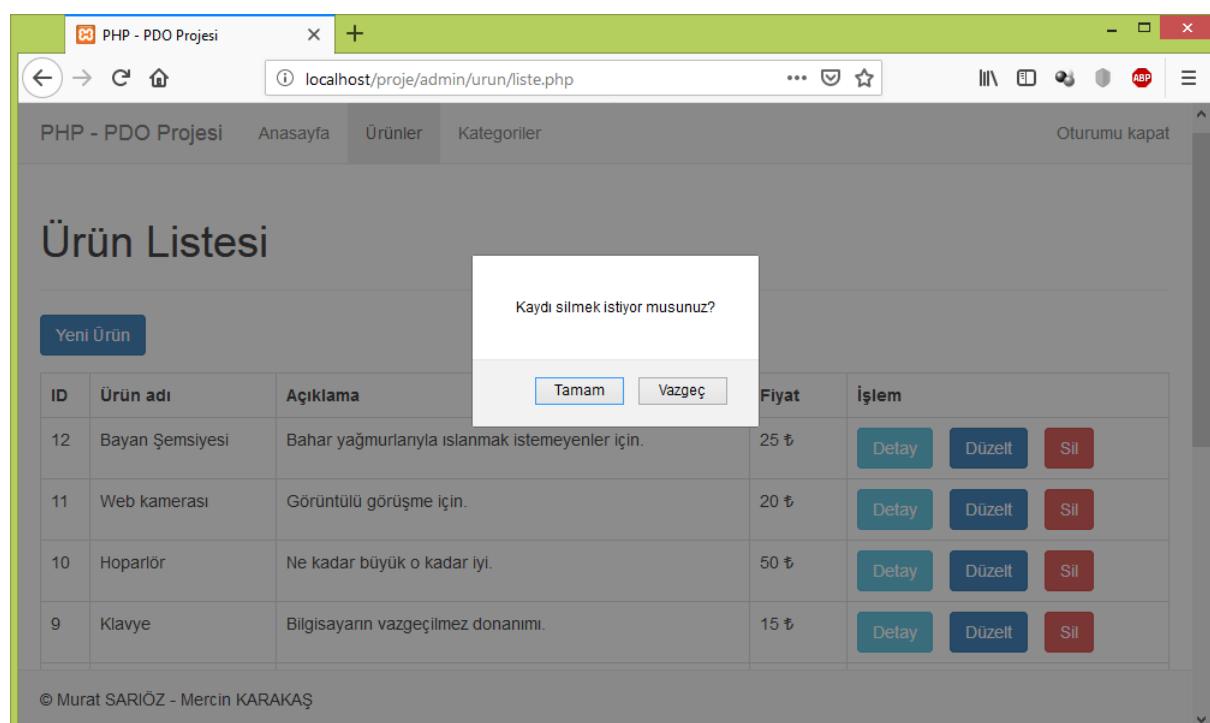
```

$islem = isset($_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 silinmedi.</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.



Silme işlemini **Tamam** butonuyla onaylayın.

Kayıt silindi.

Yeni Ürün

ID	Ürün adı	Açıklama	Fiyat	İşlem
11	Web kamerası	Görüntülü görüşme için.	20 ₺	<button>Detay</button> <button>Düzel</button> <button>Sil</button>
10	Hoparlör	Ne kadar büyük o kadar iyi.	50 ₺	<button>Detay</button> <button>Düzel</button> <button>Sil</button>
9	Klavye	Bilgisayann vazgeçilmez donanımı.	15 ₺	<button>Detay</button> <button>Düzel</button> <button>Sil</button>

© Murat SARIÖZ - Mercin KARAKAŞ

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 bilgilerini 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. ');

// aktif kayıt bilgilerini oku
try {
```

```

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

// 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
$urunadi = htmlspecialchars($kayit['urunadi'], ENT_QUOTES);
$aciklama = htmlspecialchars($kayit['aciklama'], ENT_QUOTES);
$fiyat = htmlspecialchars($kayit['fiyat'], ENT_QUOTES);
}

// hata varsa 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 -->





```

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>
    </table>
</form>
```

```

</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 kaydedecektr. Bu kodları **<!-- PHP kayıt ekleme kodları burada yer alacak -->** açıklama satırının yerine yazın.

```

<?php
if($_POST){

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

        // 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);

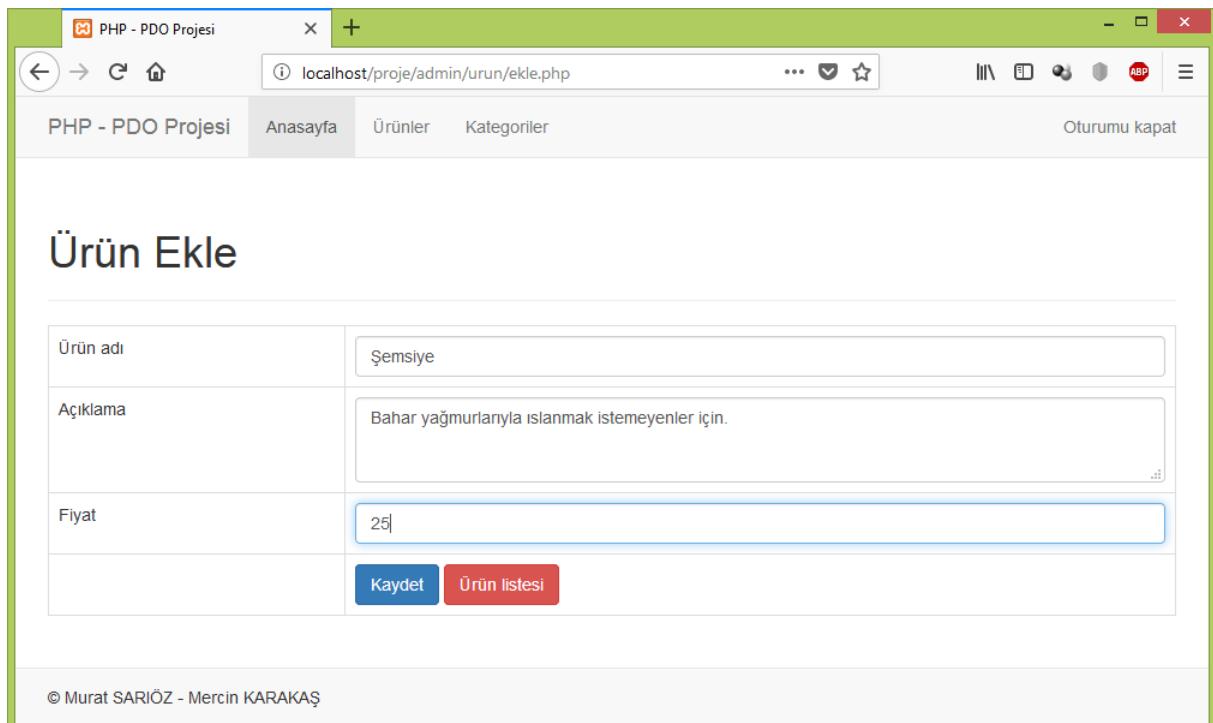
        // ürünün veritabanına kaydedildiği tarihi belirt
        $giris_tarihi=date('Y-m-d H:i:s');
        $stmt->bindParam(':giris_tarihi', $giris_tarihi);

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

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

```

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



PHP - PDO Projesi

localhost/proje/admin/urun/ekle.php

Anasayfa Ürünler Kategoriler

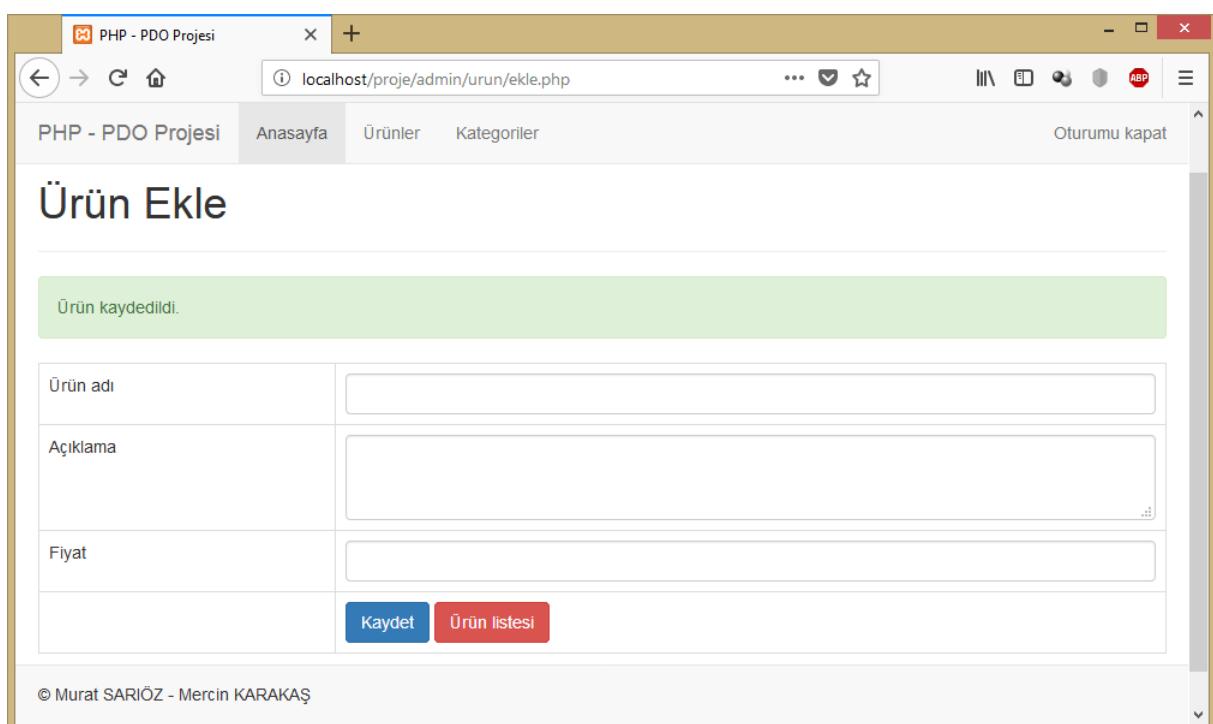
Oturumu kapat

Ürün Ekle

Ürün adı	Şemsiye
Açıklama	Bahar yağmurlarıyla ıslanmak istemeyenler için.
Fiyat	25
<button>Kaydet</button> <button>Ürün listesi</button>	

© Murat SARIÖZ - Mercin KARAKAŞ

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



PHP - PDO Projesi

localhost/proje/admin/urun/ekle.php

Anasayfa Ürünler Kategoriler

Oturumu kapat

Ürün Ekle

Ürün kaydedildi.

Ürün adı	
Açıklama	
Fiyat	
<button>Kaydet</button> <button>Ürün listesi</button>	

© Murat SARIÖZ - Mercin KARAKAŞ

ÜRÜN GÜNCELLEME (DÜZELTME) SAYFASINI OLUŞTURMAK

Ürün listeleme sayfasında **Düzel**t 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 -->

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

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

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

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

```

GÜNCELLENECEK ÜRÜN BİLGİLERİ VERİTABANINDAN OKUMAK

Öncelikle ürün listelemeye sayfasından gelen **Id** parametresini **\$_GET['id']** fonksiyonu ile alıp **\$id** değişkenine kaydediyoruz. Daha sonra bu parametreyle 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. ');

    // aktif kayıt bilgilerini oku
    try {
        // veritabanı bağlantı dosyasını dahil et
        include '../../config/vtabani.php';

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

```

KAYIT GÜNCELLEMESİ 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.

```

<!-- 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 değiştirdip **Kaydet** butonuna tıklarsa aşağıdaki kodlar bu değişiklikleri veritabanına kaydedecekktir. 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>";
    }

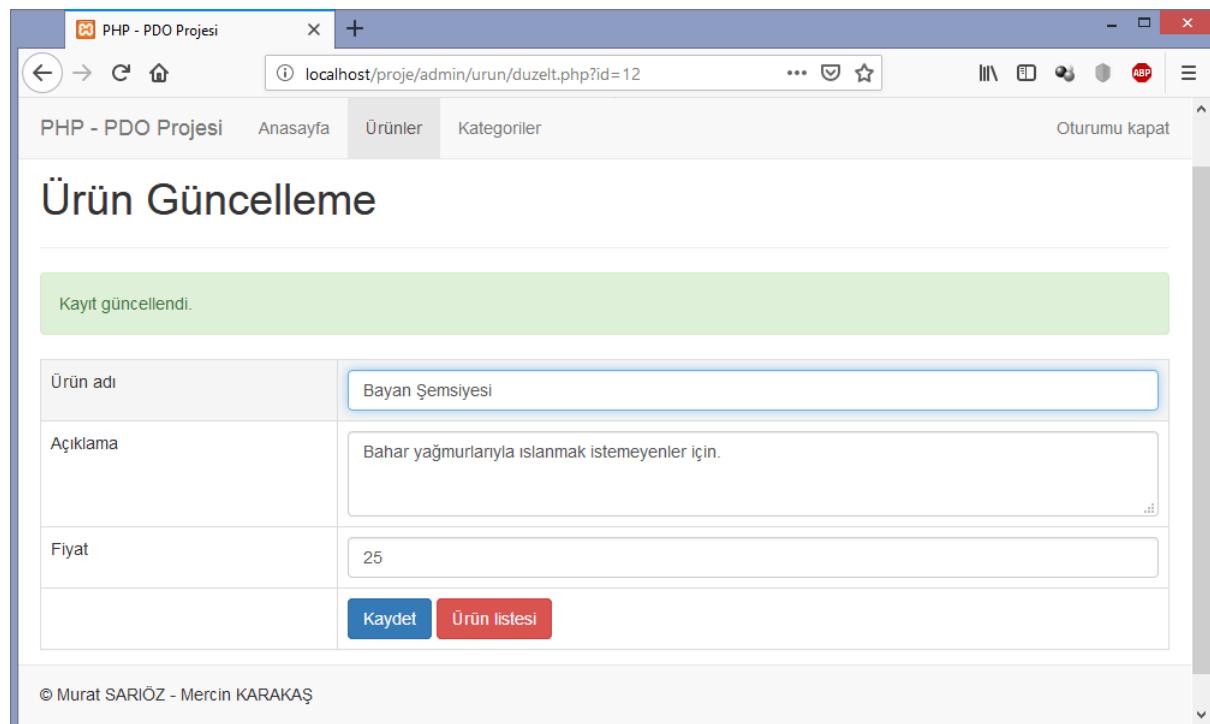
}

// 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 **Düzel**t butonunu tıklayın ve ürün güncelleme sayfasını görüntüleyin. Bilgilerde değişiklik yaparak **Kaydet** butonunu tıklayın.



SAYFALARI NUMARALAMA İŞLEMİ

Ürün listeleme sayfasında tüm ürünleri tek sayfada görüntülemek hem sayfa yükleme hızını etkileyecik hem de aranan ürünü bulmayı zorlaştıracaktır. Bu olumsuzlukları gidermek için liste işlemini 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.

```
// SAYFALAMA 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.

```
// SAYFALARI NUMARALAMA
// 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'];
```

SAYFALAMA DOSYASINI (SAYFALAMA.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 kullanmadığımız amaç **sayfalama.php** dosyasının yeniden kullanılabilirliğini sağlamak için. Yani diğer liste sayfalarındaki sayfalandırma 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 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 --&gt;</pre>
```

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  
$sayfa_sayisi = ceil($sayfa_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 değerinin 0'dan büyük VE $sayfa_sayisi'na eşit veya küçük olduğundan
emin ol
if (($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 OLUŞTURMAK

sayfalama.php dosyasındaki **// sonraki sayfa butonu burada yer alacak** satırı yerine aşağıdaki 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 ürün liste sayfasını görüntüleyin. Sayfa numaralarına tıklayarak sayfalar arasında geçiş yapın.

ID	Ürün adı	Açıklama	Fiyat	İşlem
8	Yastık	İyi bir uykú için.	8.99 ₺	Detay Düzenle Sil
7	Kulaklık	Müzik severler için.	7 ₺	Detay Düzenle Sil
6	Mouse	Bilgisayar kullanmayı sevenler için.	11.35 ₺	Detay Düzenle Sil

« 1 2 3 4 »

© Murat SARIÖZ - Mercin KARAKAŞ

SAYFA NUMARALARINI LİSTELEYEN DROP-DOWN MENÜYÜ OLUŞTURMAK

sayfalama.php dosyasındaki `<!-- sayfa numaralarını listeleyen drop-down menü burada yer alacak -->` satırı 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.

The screenshot shows a web application titled "PHP - PDO Projesi" with a sub-page "Ürün Listesi". The main content area displays a table of products:

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

Below the table is a navigation bar with page numbers («, 1, 2, 3, 4, ») and a search bar. The footer contains the text "© Murat SARIÖZ - Mercin KARAKAŞ".

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.php

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 ₺	<button>Detay</button> <button>Düzel</button> <button>Sil</button>
9	Klavye	Bilgisayarın vazgeçilmez donanımı.	15 ₺	<button>Detay</button> <button>Düzel</button> <button>Sil</button>
8	Yastık	İyi bir uyku için.	8.99 ₺	<button>Detay</button> <button>Düzel</button> <button>Sil</button>

« 1 2 3 4 »

2 Git

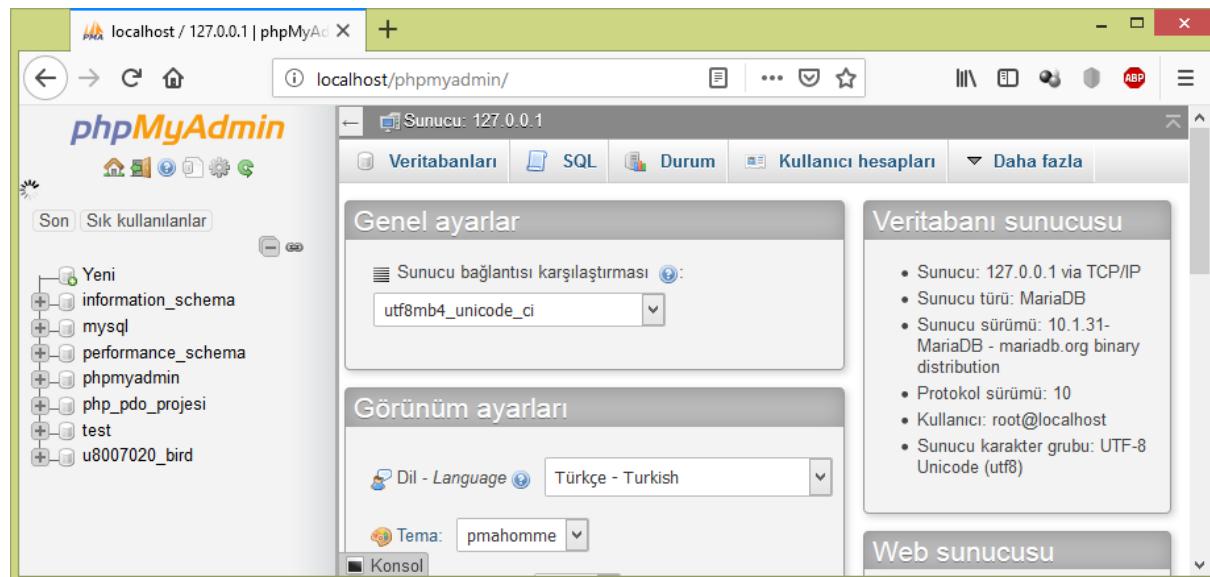
© Murat SARIÖZ - Mercin KARAKAŞ

RESİM YÜKLEMEK

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

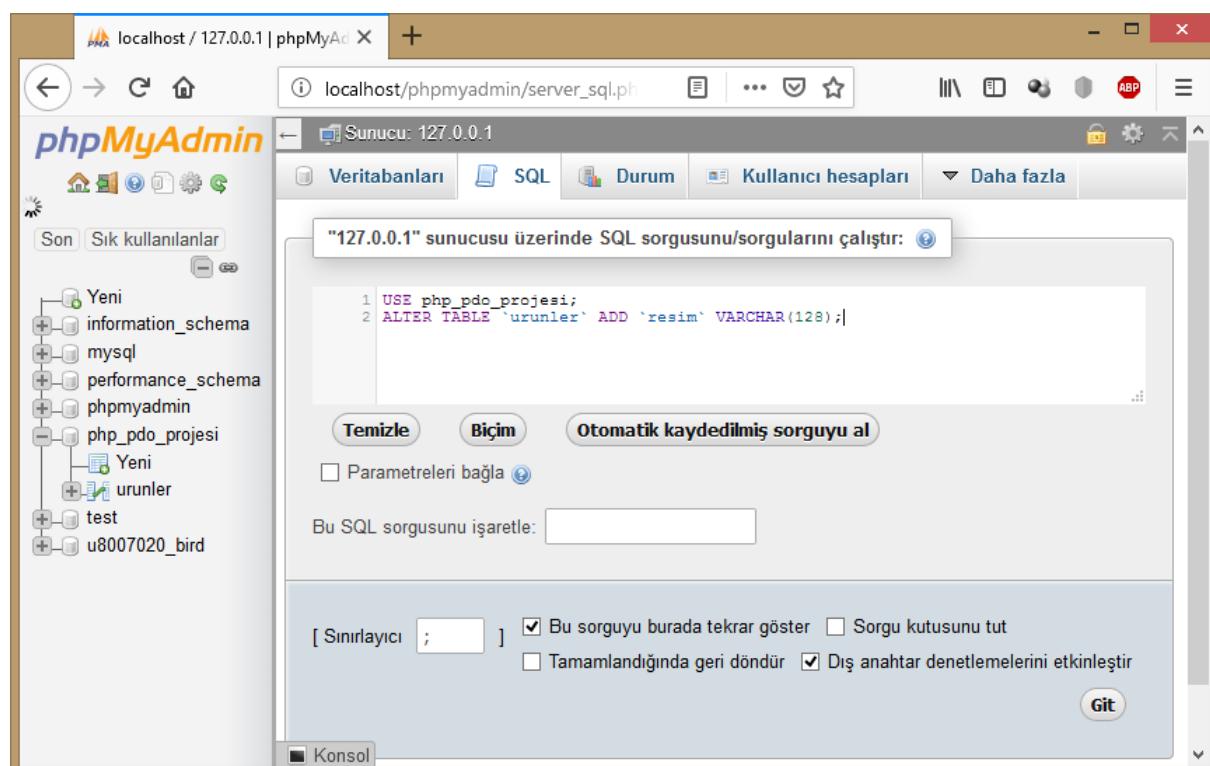
VERİTABANINA YENİ BİR ALAN EKLEMEK

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 Ekle

Ürün adı	
Açıklama	
Fiyat	
Resim	Gözat... Hiçbir dosya seçilmedi.
	Kaydet Ürün listesi

© Murat SARIÖZ - Mercin KARAKAŞ

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 sınırlamak 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 yüklenebilir.</div>";
}
```

DOSYANIN İSİM KONTROLÜ

uniqid() fonksiyonunu kullanmadan 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 sınırlıracagız. 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>";
}
```

PHP - PDO Projesi

localhost/proje/admin/urun/ekle.php

PHP - PDO Projesi Anasayfa Ürünler Kategoriler Oturumu kapat

Ürün Ekle

Ürün kaydedildi.

Ürün adı	<input type="text"/>
Açıklama	<input type="text"/>
Fiyat	<input type="text"/>
Resim	Gözat... <input type="button" value="Gözat..."/> Hiçbir dosya seçilmedi.
	<input type="button" value="Kaydet"/> <input type="button" value="Ürün listesi"/>

© Murat SARIÖZ - Mercin KARAKAŞ

ÜRÜN DETAY SAYFASINDA YÜKLENEN RESMİ GÖRÜNTÜLEMEK
Yüklenen ürün resmini **detay.php** sayfasında görüntüleyeceğiz.

SEÇME SORGUSUNU DÜZENLEMEK

Ürün 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 EKLEMELİ

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}'>" : "Ürün görseli yok."; ?></td>
</tr>

```

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

Ürün adı	Duvar tablosu
Açıklama	Sırasıma kuşu tablosu.
Fiyat	99
Resim	
	Ürün listesi

ÜRÜN SİLİNDİĞİNDE YÜKLENEN RESMİ SUNUCUDAN SİLMEK

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

SİLİNEN RESİM DOSYASININ İSMİNİ ÖĞRENMEK

Urun klasöründeki **sil.php** dosyasını açın ve **\$id=isset(\$_GET['id']) ? \$_GET['id'] : die('HATA: Id bilgi 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 ürünler bulmak zorlaşacaktır. Bu işlemi kolaylaştırmak için ürün listelemeye 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
```

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

SEÇME SORGUSUNU DÜZENLEMEK

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

Eski kod:

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

// 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:

```
$aranan = isset($_GET['aranan']) ? $_GET['aranan'] : "";
$arama_sarti = isset($_GET['aranan']) ? "%" . $_GET['aranan'] . "%" : "%";

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

// sayfada görüntülenecek kayıtları seç
$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", $arama_sarti);
```

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", $arama_sarti);
```

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 listelemeyi sayfasını görüntüleyin. **Ürün ara** alanına bir ürün adı yazarak arama butonuna tıklayın.

The screenshot shows a browser window titled "PHP - PDO Projesi". The address bar displays "localhost/proje/admin/urun/liste.php?aranan=kulaklik". The main content area is titled "Ürün Listesi". At the top right, there is a search input field containing "kulaklik" and a magnifying glass icon. Below the search bar, there is a table with the following data:

ID	Ürün adı	Açıklama	Fiyat	İşlem
7	Kulaklik	Müzik severler için.	7 ₺	Detay Düzenle Sil

At the bottom left, there is a page number "1" and a "Git" button. At the bottom center, the copyright notice "© Murat SARIÖZ - Mercin KARAKAŞ" is visible.

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 EKLEMELİ

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-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-eye-open'></span> Detay</a>";
```

Düzelte butonu için:

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

Sil butonu için:

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

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

ÜRÜN DETAY SAYFASINDAKİ BUTONLARA SİMGE EKLEMELİ

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-list'></span> Ürün listesi</a>
```

The screenshot shows a web browser window titled "PHP - PDO Projesi". The address bar displays "localhost/proje/admin/urun/detay.php?id=13". The page content is titled "Ürün Bilgisi". It contains a table with the following data:

Ürün adı	Duvar tablosu
Açıklama	Sarıasma kuşu tablosu.
Fiyat	99.99
Resim	

Below the table is a red button labeled "Ürün listesi". At the bottom of the page, there is a copyright notice: "© Murat SARIÖZ - Mercin KARAKAŞ".

ÜRÜN EKLEME SAYFASINDAKİ BUTONLARA SİMGЕ EKLEMЕK

Ürün 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-list'></span> Ürün listesi</a>
```

PHP - PDO Projesi

localhost/proje/admin/urun/ekle.php

PHP - PDO Projesi Anasayfa Ürünler Kategoriler

Oturumu kapat

Ürün Ekle

Ürün adı	<input type="text"/>
Açıklama	<input type="text"/>
Fiyat	<input type="text"/>
Resim	Gözat... <small>Hiçbir dosya seçilmedi.</small>

© Murat SARIÖZ - Mervin KARAKAŞ

ÜRÜN GÜNCELLEME SAYFASINDAKİ BUTONLARA SİMGE EKLEMELİK

Ürün 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-list'></span> Ürün listesi</a>
```

PHP - PDO Projesi

localhost/proje/admin/urun/duzelt.php?id=13

PHP - PDO Projesi Anasayfa Ürünler Kategoriler

Oturumu kapat

Ürün Güncelleme

Ürün adı	Duvar tablosu
Açıklama	Sarıasma kuşu tablosu.
Fiyat	99.99

© Murat SARIÖZ - Mervin KARAKAŞ

ÇOKLU ÜRÜN SİLME İŞLEVİ

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 ürünlerini listelediğimiz HTML tablo kodlarına aşağıdaki gibi eklemeler yapalım.

Tablo başlık kodlarına aşağıdaki satırı ekleyin.

```
//tablo başlıklar
echo "<tr>";
    echo "<th></th>";
    echo "<th>ID</th>";
    echo "<th>Ürün adı</th>";
    echo "<th>Açıklama</th>";
    echo "<th>Fiyat</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>{$fiyat} &#8378;</td>";
    echo "<td>";
        // kayıt detay sayfa bağlantısı
        echo "<a href='detay.php?id={$id}' class='btn btn-info m-r-1em'><span
class='glyphicon glyphicon-eye-open'></span> Detay</a>";
        // kayıt güncelleme sayfa bağlantısı
        echo "<a href='duzelt.php?id={$id}' class='btn btn-primary m-r-1em'><span
class='glyphicon glyphicon-edit'></span> Düzelt</a>";
        // bu bağlantıyı dokümanın ileriki bölümlerinde işleyeceğiz
        echo "<a href='#' onclick='silme_onay({$id});' class='btn btn-danger'><span
class='glyphicon glyphicon-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-remove'> Seçilenleri
Sil</a>";
```

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

ID	Ürün adı	Açıklama	Fiyat	İşlem
7	Kulaklık	Müzik sevenler için	7 ₺	<button>Detay</button> <button>Düzenle</button> <button>Sil</button>
6	Mouse	Bilgisayar kullanmayı sevenler için.	11.35 ₺	<button>Detay</button> <button>Düzenle</button> <button>Sil</button>
5	Çöp kovası	Etrafi temiz tutmanız için.	3.95 ₺	<button>Detay</button> <button>Düzenle</button> <button>Sil</button>

« 2 3 4 5 » 4 Gt

© Murat SARIÖZ - Mercin KARAKAŞ

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("Silmek 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!", {
                icon: "success",
                buttons: false,
                timer: 3000,
            });
            // silinen kayıtları html tablosundan da sil
            jQuery('input:checkbox:checked').parents("tr").remove();
        },
        error: function(jqXHR, textStatus, errorThrown){
            alert(textStatus + errorThrown);
        }
    });
});

```

ÇOKLU ÜRÜN SİLME SAYFASINI (COKLUSIL.PHP) OLUŞTURMAK

Son olarak **admin/urun** klasöründe **coklusil.php** isminde bir dosya oluşturun ve içine aşağıdaki kodları yazın.

```

<?php
$id_dizisi = $_POST['id'];
$ids = implode(',', $id_dizisi);

include '../config/vtabani.php';

// önce ürün resimleri sil
foreach ($id_dizisi as $id) {
    $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);
    $resim = $kayit['resim'];

    if(!empty($resim)){
        $silincek_resim = "../content/images/".$resim;
        if (file_exists($silincek_resim)) unlink($silincek_resim);
    }
}
// daha sonra ürünleri tablodan sil
$con->query("DELETE FROM urunler WHERE id IN ($ids)");
?>

```

Tarayıcınızın adres satırına <http://localhost/proje/admin/urun/liste.php> yazarak ürün listelemeyi sayfasını görüntüleyin. Silmek istediğiniz ürün kayıtlarını seçin ve **Seçilenleri Sil** butonuna tıklayın.

PHP - PDO Projesi

localhost/proje/admin/urun/liste.php?sayfa=4&aranan=#

PHP - PDO Projesi Anasayfa Ürünler Kategoriler Oturumu kapat

Ürün Listesi

	ID	Ürün adı	Açıklama
<input checked="" type="checkbox"/>	7	Kulaklık	Müzik dinleme cihazı
<input type="checkbox"/>	6	Mouse	Bilgisayar faresi
<input checked="" type="checkbox"/>	5	Çöp kovası	Etraf milleti koruma

Silme işlemi geri alınamaz!

© Murat SARIÖZ - Mercin KARAKAŞ

4

KATEGORİ SAYFALARINI OLUŞTURMAK

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

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

The screenshot shows the phpMyAdmin interface with the following details:

- General Settings (Genel Ayarlar):** Server encoding is set to utf8mb4_general_ci.
- Display Settings (Görünüm Ayarları):** Language is set to Türkçe - Turkish, theme is pmahomme, and font size is 82%.
- Database Information (Veritabanı sunucusu):** The server is 127.0.0.1 via TCP/IP, MySQL version is 5.6.21, and character set is UTF-8 Unicode (utf8).
- Web Server Information (Web sunucusu):** Apache/2.4.10 (Win32) is running.

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

The screenshot shows the SQL query editor in phpMyAdmin with the following query:

```
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;
```

```
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 LİSTELEMİ 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 LİSTELEMİK İÇİN GEREKLİ SQL SORGUSUNU OLUŞTURMAK

Ardından **<!-- Kayıtları listeleyecek PHP kodları bu alana eklenecek -->** açıklamasının 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, kategoriadi, 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 liste
    if($sayi>0){
        // kayıtlar burada listelenecek
    }
    // kayıt yoksa mesajla bildir
    else{
        echo "<div class='alert alert-danger'>Listelenecek kayıt bulunamadı.</div>";
    }
?>
```

KAYIT LİSTELEMЕK 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 LİSTELEMЕK 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['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={$id}' class='btn btn-primary m-r-1em'>Düzelt</a>";  
        // kayıt silme butonu  
        echo "<a href='#' onclick='silme_onay({$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 onayı **liste.php** sayfasında, silme işlemini 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/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.');

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

```

// soruyu çalıştır
if($stmt->execute()){
    // kayıt listeleme sayfasına yönlendir ve kullanıcıya kaydın silindiğini
    header('Location: liste.php?islem=silindi');
} // veya silinmediğini 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 işlemini **Tamam** butonuyla onaylayın.

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

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 BİLGİLERİNI VERİTABANINDAN OKUMAK

Aşağıdaki PHP kodları MySQL veritabanındaki **kategoriler** 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 kategoriler 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'];
}

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

KATEGORİ BİLGİLERİНИ GÖRÜNTÜLEMEK İÇİN GEREKLİ HTML TABLOSUNU OLUŞTURMAK

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

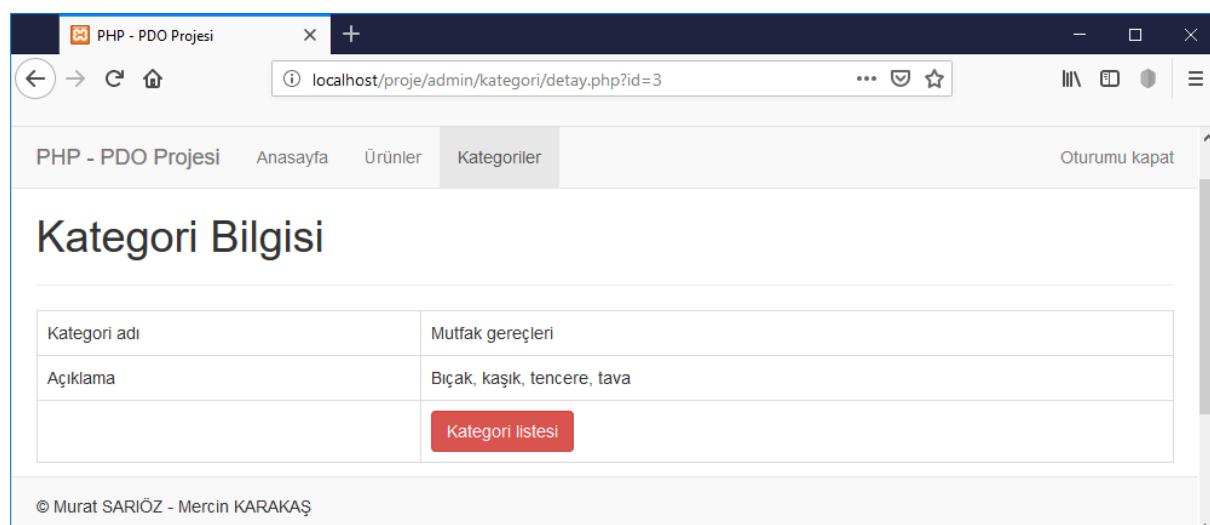
```
<!--kayıt bilgilerini görüntüleyen HTML tablosu -->





```

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



Kategori adı	Mutfak gereçleri
Açıklama	Bıçak, kaşık, tencere, tava

Kategori listesi

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 **kategoriler** 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 eklemeye sorgusu
        $sorgu = "INSERT INTO kategoriler 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>";
        }

    }
    // hata varsa göster
    catch(PDOException $exception){
}
```

```

        die('HATA: ' . $exception->getMessage());
    }
}
?>

```

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

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

KATEGORİ GÜNCELLEME (DÜZELTME) SAYFASINI OLUŞTURMAK

Kategori liste sayfasında **Düzel**t 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 -->
<!-- kaydı güncelleyecek PHP kodu burada yer alacak -->
<!-- kategori bilgilerini düzeltme HTML formu burada yer alacak -->
</div> <!-- container -->
<?php include "../footer.php"; ?>

```

GÜNCELLENECEK KATEGORİ BİLGİLERİNI VERİTABANINDAN OKUMAK

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

```

KAYIT GÜNCELLEMEMEK 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.

```

<!-- 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 değiştirdip **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 **Düzel**t butonunu tıklayın ve kategori güncelleme sayfasını görüntüleyin. Bilgilerde değişiklik yaparak **Kaydet** butonunu tıklayın.

Kategori Güncellemeye hoş geldiniz.

Kayıt güncellendi.

Kategori adı	Oyuncak
Açıklama	Bebek, çocuk ve yetişkin oyuncaklarımız

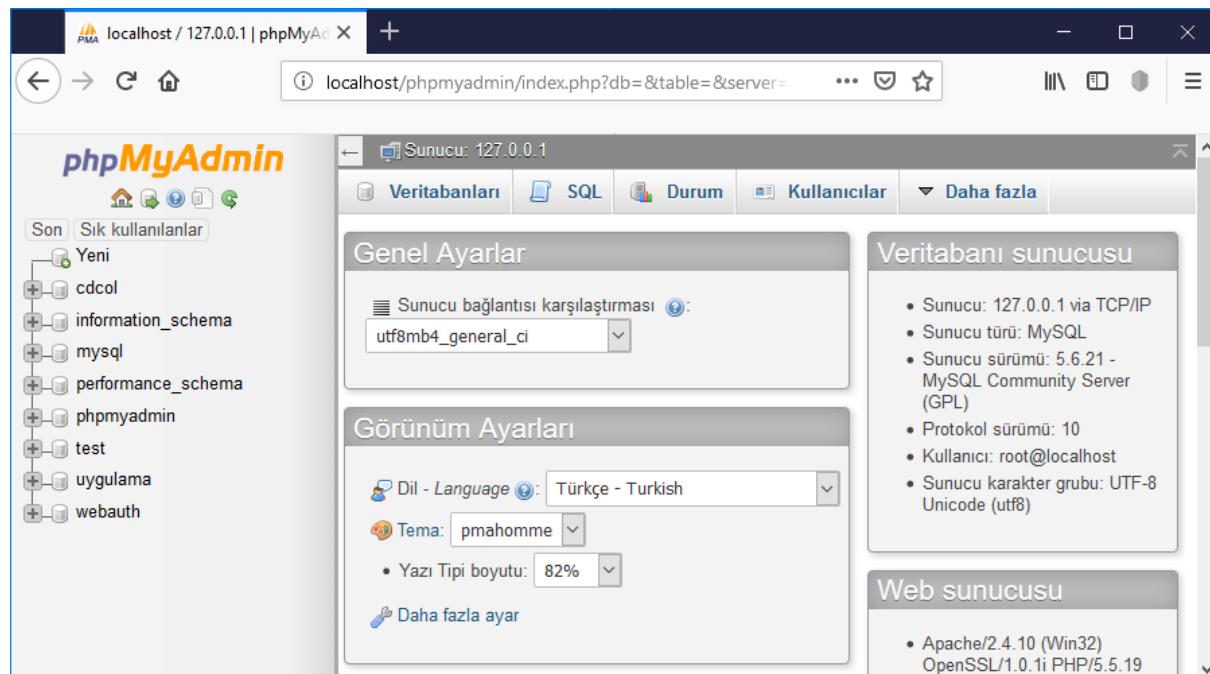
[Kategori listesi](#)

© 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.

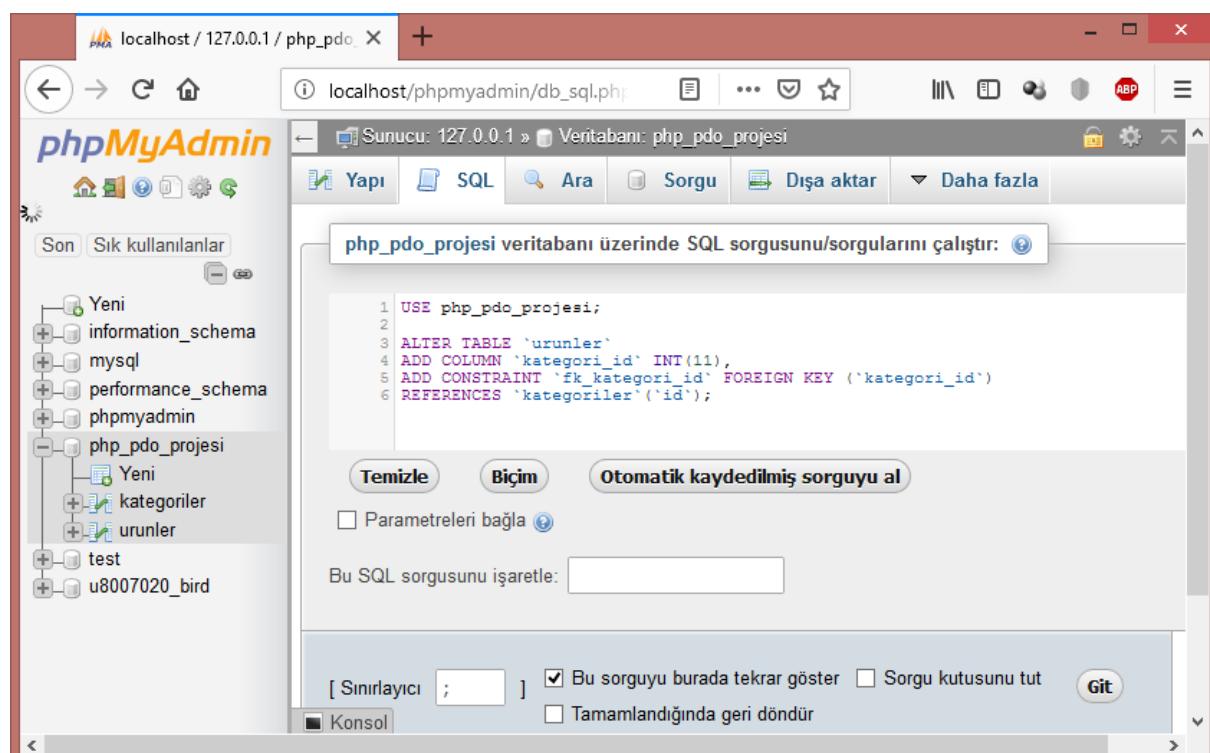
Veritabanında **urunler** tablosuya **kategoriler** tablosunu ilişkilendirmek için tarayıcınızın adres satırına <http://localhost/phpmyadmin/> yazarak **phpMyAdmin** veritabanı yönetim panelini açın.



The screenshot shows the phpMyAdmin interface with the following details:

- General Settings (Genel Ayarlar):** Character set for server connection is set to utf8mb4_general_ci.
- Display Settings (Görünüm Ayarları):** Language is set to Türkçe - Turkish, theme is pmahomme, and font size is 82%.
- Server Information (Veritabanı sunucusu):** Server details include: Sunucu: 127.0.0.1 via TCP/IP, Sunucu türü: MySQL, Sunucu sürümü: 5.6.21 - MySQL Community Server (GPL), Protokol sürümü: 10, Kullanıcı: root@localhost, and Sunucu karakter grubu: UTF-8 Unicode (utf8).
- Web Server (Web sunucusu):** Apache/2.4.10 (Win32) OpenSSL/1.0.1i PHP/5.5.19

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



The screenshot shows the phpMyAdmin interface with the following details:

- Database Structure (Yapı):** The database structure for the **php_pdo_projesi** database is shown, including tables **kategoriler** and **urunler**.
- SQL Editor:** A SQL query is entered in the editor:

```
1 USE php_pdo_projesi;
2
3 ALTER TABLE `urunler`
4 ADD COLUMN `kategori_id` INT(11),
5 ADD CONSTRAINT `fk_kategori_id` FOREIGN KEY (`kategori_id`)
6 REFERENCES `kategoriler`(`id`);
```
- Execution Options:** Buttons for **Temizle**, **Biçim**, and **Otomatik kaydedilmiş sorguyu al** are present. There is also a checkbox for **Parametreleri bağla**.
- Execution Buttons:** **Git** button, **Sınırlayıcı** dropdown, and checkboxes for **Bu sorguyu burada tekrar göster** and **Sorgu kutusunu tut**.

```
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 EKLEMEM

Admin/urun/ekle.php dosyasını açın ve daha önce bilgi girişi için oluşturduğunuz HTML formundaki **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
            // veritabanı yapılandırma dosyasını dahil et
            include '../config/vtabani.php';
            // kayıt listeleme sorusu
            $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 sorusu
$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 sorusu
$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 değerler
$urunadi=htmlspecialchars(strip_tags($_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 bağla
$stmt->bindParam(':urunadi', $urunadi);
$stmt->bindParam(':aciklama', $aciklama);
$stmt->bindParam(':fiyat', $fiyat);
$stmt->bindParam(':resim', $resim);
$stmt->bindParam(':kategori_id', $kategori_id);
// ürünün veritabanına kaydedildiği tarihi belirt
$giris_tarihi=date('Y-m-d H:i:s');
$stmt->bindParam(':giris_tarihi', $giris_tarihi);

```

The screenshot shows a web application interface for adding a product. The title bar says "PHP - PDO Projesi". The URL in the address bar is "localhost/proje/admin/urun/ekle.php". The page title is "Ürün Ekle". Below it, there are tabs for "Anasayfa", "Ürünler" (selected), and "Kategoriler". The main content area contains a form with the following fields:

- Ürün adı (Product Name): An empty input field.
- Açıklama (Description): An empty input field.
- Fiyat (Price): An empty input field.
- Kategori (Category): A dropdown menu with options "Elektronik", "Elektronik" (highlighted in blue), "Spor", and "Mutfak gereğleri".
- Resim (Image): An empty input field.

At the bottom of the form, there are two buttons: "Kaydet" (Save) and "Ürün listesi" (Product List). The footer of the page says "© Murat SARIÖZ - Mercin KARAKAŞ".

ÜRÜN GÜNCELLEME SAYFASINA KATEGORİ ALANI EKLEMELİ

Admin/urun/duzelt.php dosyasını açın ve daha önce bilgi girişi için oluşturduğunuz HTML formundaki **Fiyat** alanı satır bitiş etiketileyi (**</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);

```

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

© Murat SARIÖZ - Mercin KARAKAŞ

ÜRÜN LISTELEME 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.

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

« 1 2 3 4 5 » 3 Git

© Murat SARIÖZ - Mercin KARAKAŞ

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>
```

The screenshot shows a web browser window titled "PHP - PDO Projesi". The URL is "localhost/proje/admin/urun/detay.php?id=9". The page has a header with tabs: "PHP - PDO Projesi", "Anasayfa", "Ürünler" (selected), and "Kategoriler". On the right, there's a "Logout" link. The main content area is titled "Ürün Bilgisi" and contains a table with the following data:

Ürün adı	Klavye
Açıklama	Bilgisayarın vazgeçilmez donanımı.
Fiyat	15 ₺
Kategori	Elektronik
Resim	Ürün görseli yok.

Below the table is a red button labeled "Ürün listesi". At the bottom left, there's a copyright notice: "© Murat SARIÖZ - Mercin KARAKAŞ".

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** açıklaması 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.

The screenshot shows a web browser window titled "PHP - PDO Projesi". The URL is "localhost/proje/admin/kategori/liste.php". The page has a header with tabs: "PHP - PDO Projesi", "Anasayfa", "Ürünler" (selected), and "Kategoriler". On the right, there's a "Logout" link. The main content area is titled "Kategori Listesi" and features a blue button labeled "Yeni Kategori". Below it is a table with the following data:

ID	Kategori adı	Açıklama	İşlem
5	Kişisel bakım	Parfüm, deodorant, krem, sabun, şampuan	Ürünler Detay Düzenle Sil
4	Oyuncak	Bebek ve çocuk oyuncakları	Ürünler Detay Düzenle Sil
3	Mutfak gereçleri	Bıçak, kaşık, tencere, tava	Ürünler Detay Düzenle Sil
2	Spor	Spor giyim, egzersiz aletleri	Ürünler Detay Düzenle Sil
1	Elektronik	Bilgisayar, televizyon, cep telefonu	Ürünler Detay Düzenle Sil

At the bottom left, there's a copyright notice: "© Murat SARIÖZ - Mercin KARAKAŞ".

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 LİSTELEMİR 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 liste
if($sayi>0){

    // kayıtlar burada listelenecek

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

```

KAYIT LİSTELEMİR 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.

```

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

//tablo başlıklar
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 LİSTELEMEK İÇİN GEREKLİ PHP KODLARI

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

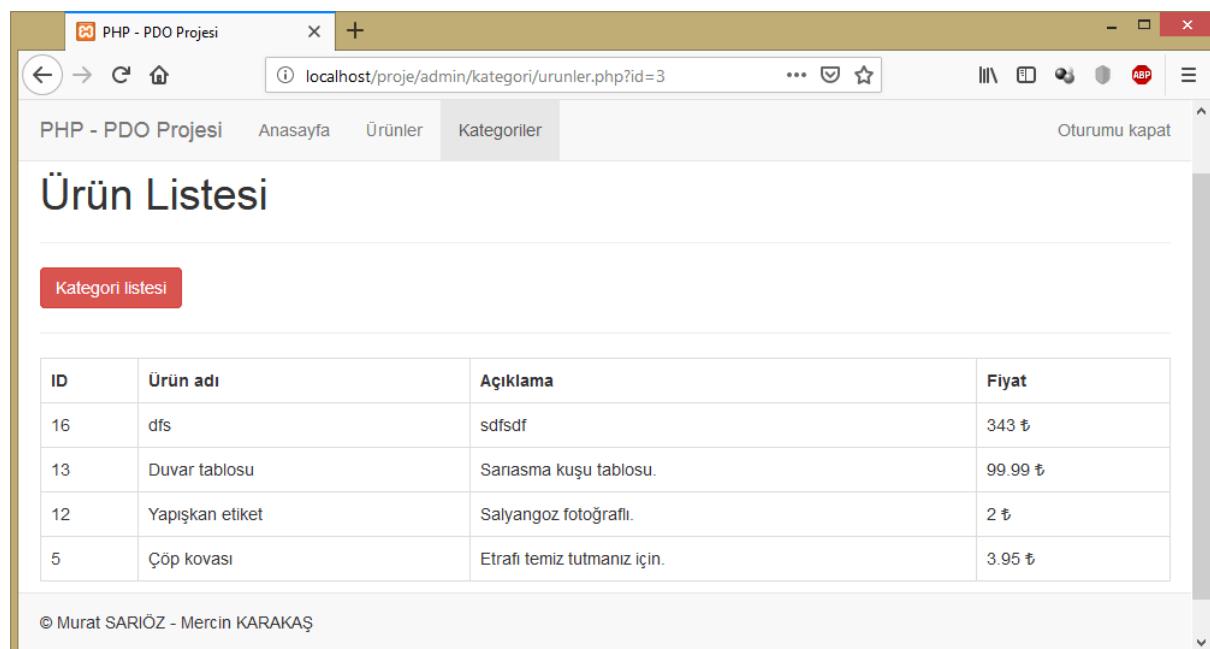
```

// 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 işaretini
    echo "</tr>";
}

```

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



The screenshot shows a web browser window titled "PHP - PDO Projesi". The address bar contains the URL "localhost/proje/admin/kategori/urunler.php?id=3". The page title is "Ürün Listesi". A red button labeled "Kategori listesi" is visible. Below it is a table with the following data:

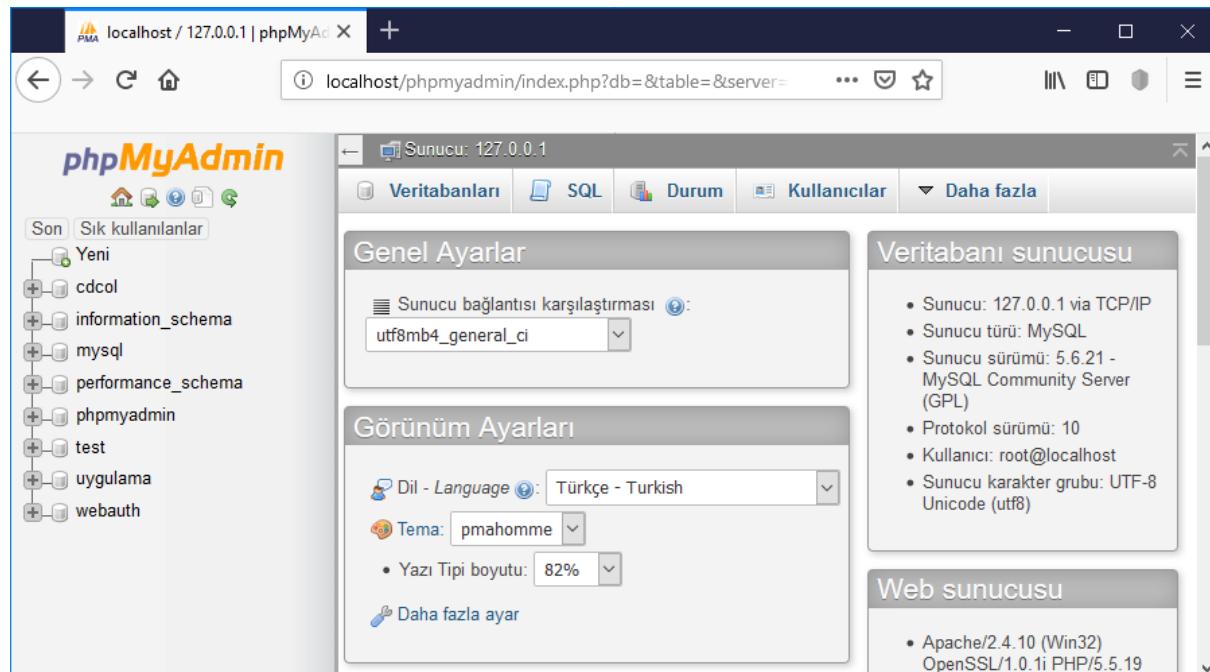
ID	Ürün adı	Açıklama	Fiyat
16	dfs	sdfsdf	343 ₺
13	Duvar tablosu	Sırasma kuşu tablosu.	99.99 ₺
12	Yapışkan etiket	Salyangoz fotoğrafı.	2 ₺
5	Çöp kovası	Etrafı temiz tutmanız için.	3.95 ₺

At the bottom of the page, there is a copyright notice: "© Murat SARIÖZ - Mercin KARAKAŞ".

KULLANICI SAYFALARINI OLUŞTURMAK

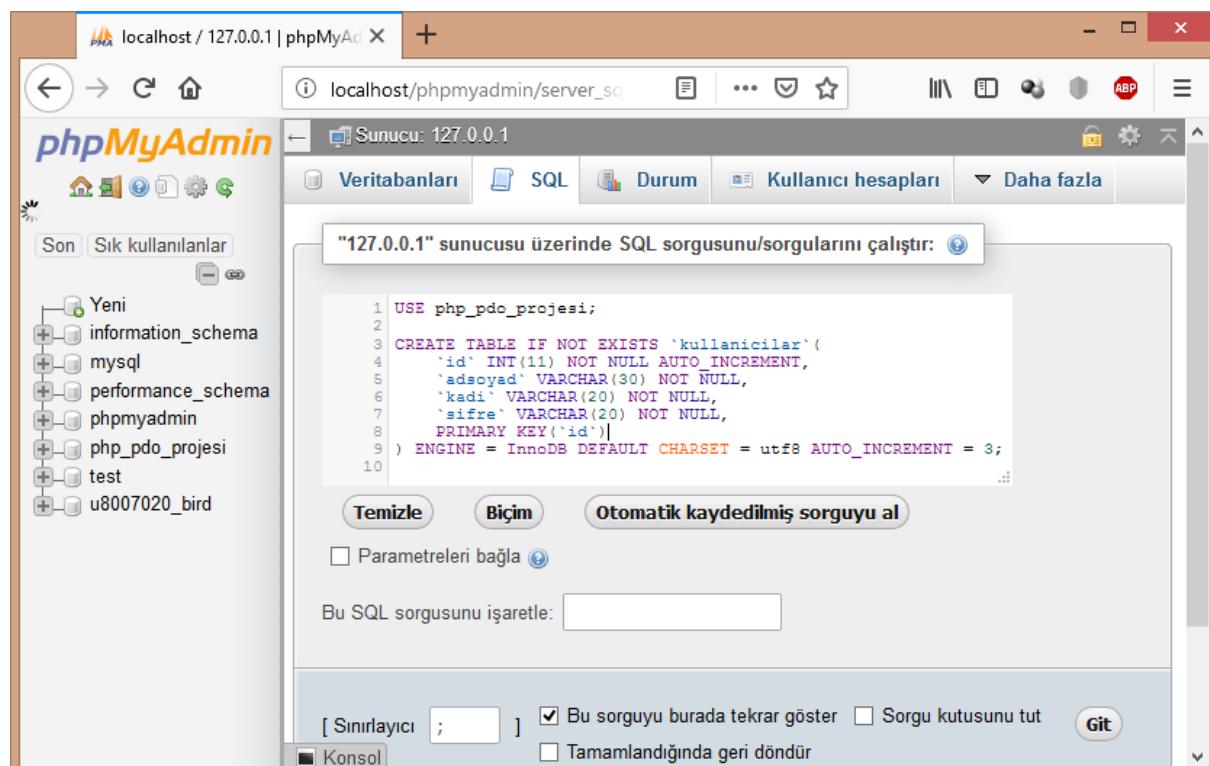
Yönetim paneline giriş kontrolü için kullanıcı adı ve şifre bilgileri kullanılacak. Öncelikle bu bilgileri saklayacağımız tabloyu oluşturmalıyız. Daha sonra da tabloya bilgi girişi ve bilgi düzenleme işlemleri için web sayfalarını hazırlamalıyız.

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



The screenshot shows the phpMyAdmin configuration page. On the left, there's a sidebar with a tree view of databases: Yeni, cdcol, information_schema, mysql, performance_schema, phpmyadmin, test, uygulama, and webauth. The main area has two panels: 'Genel Ayarlar' (General Settings) and 'Görünüm Ayarları' (View Settings). In 'Genel Ayarlar', the character set for the connection is set to 'utf8mb4_general_ci'. In 'Görünüm Ayarları', the language is set to 'Türkçe - Turkish' and the font size is set to 82%. To the right, there are two boxes: 'Veritabanı sunucusu' (Database Server) which lists the server details, and 'Web sunucusu' (Web Server) which lists the Apache, OpenSSL, and PHP versions.

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



The screenshot shows the SQL query editor in phpMyAdmin. The URL in the address bar is 'localhost/phpmyadmin/server_sql'. The main area contains a large text box with the following SQL code:

```
USE php_pdo_projesi;
CREATE TABLE IF NOT EXISTS `kullanicilar`(
  `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`)
) ENGINE = InnoDB DEFAULT CHARSET = utf8 AUTO_INCREMENT = 3;
```

Below the code, there are several buttons: 'Temizle' (Clear), 'Biçim' (Format), and 'Otomatik kaydedilmiş sorguyu al' (Get stored query). There are also checkboxes for 'Parametreleri bağla' (Bind parameters) and 'Bu SQL sorgusunu işaretle:' (Mark this SQL query). At the bottom, there are options for 'Sınırlayıcı' (Limit), 'Bu sorguyu burada tekrar göster' (Show this query again here), 'Sorgu kutusunu tut' (Keep query in box), 'Konsol' (Console), and 'Tamamlandıında geri döndür' (Return after completion). A 'Git' button is also present.

```
USE php_pdo_projesi;

CREATE TABLE IF NOT EXISTS `kullanicilar`(
  `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`)
) ENGINE = InnoDB DEFAULT CHARSET = utf8 AUTO_INCREMENT = 3;
```

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

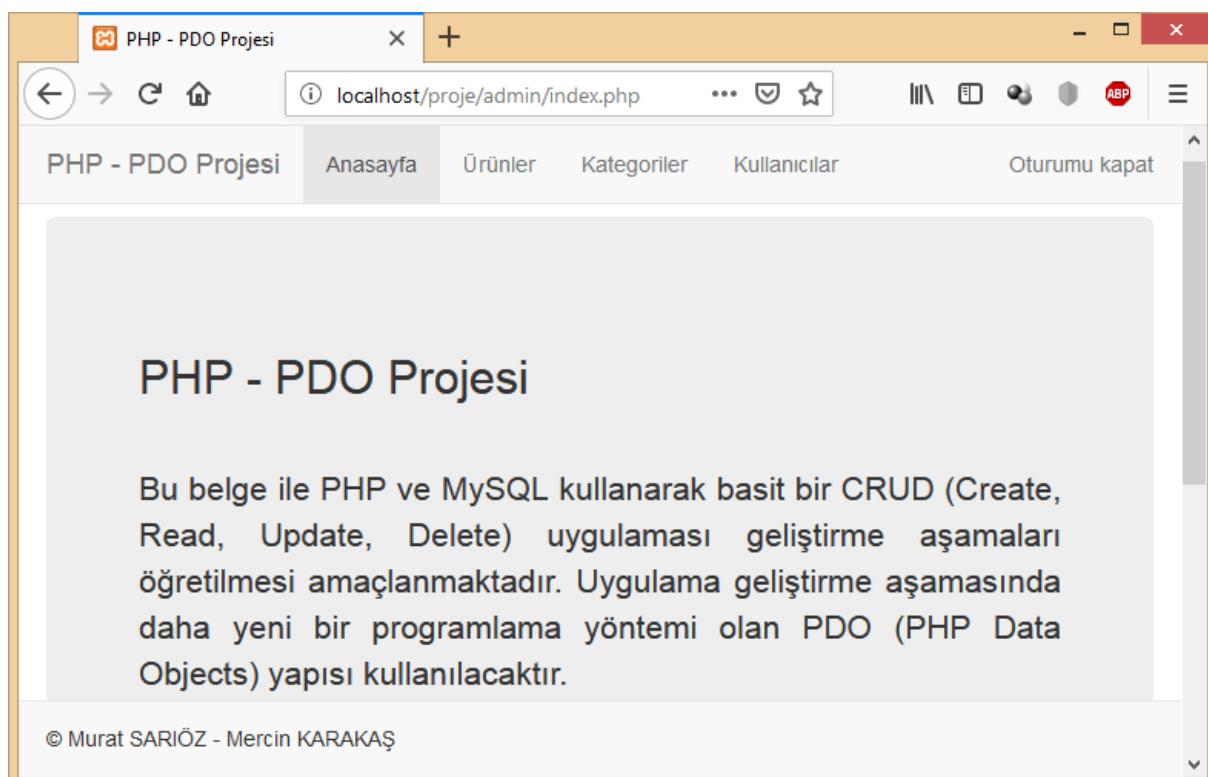
```
INSERT INTO `kullaniciilar` ( `id` , `adsoyad` , `kadi` , `sifre` ) VALUES
( 1 , 'Mercin Karakaş' , 'mercin' , '1234' ),
( 2 , 'Murat Sarıöz' , 'murat' , 'abc' )
```

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 sayfاسını 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 LİSTELEMİR 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 liste
if($sayi>0){

    // kayıtlar burada listelenecek

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

```

KAYIT LİSTELEMİR 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 başlıklar
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

```

VERİTABANINDAKİ KAYITLARI LİSTELEMEK 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['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'>Düzelt</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.

The screenshot shows a browser window titled "PHP - PDO Projesi". The address bar contains "localhost/proje/admin/kullanici/lis". The page header includes links for "Anasayfa", "Ürünler", "Kategoriler", "Kullanıcılar" (which is the active tab), and "Oturumu kapat". The main content area is titled "Kullanıcı Listesi". Below the title is a button labeled "Yeni Kullanıcı". A table lists users with the following data:

ID	Ad ve Soyad	Kullanıcı adı	İşlem
2	Murat Sanöz	murat	<button>Düzelt</button> <button>Sil</button>
1	Mercin Karakaş	mercin	<button>Düzelt</button> <button>Sil</button>

At the bottom of the page, there is a footer note: "© Murat SARIOZ - Mercin KARAKAŞ".

KULLANICI SILME 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 işlemini 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/kullanıcı** 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 silinmediğini 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 silinmedi.</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.

Kullanıcı Listesi

ID	Ad ve Soyad		
2	Murat Sarıöz	murat	Düzel Sil
1	Mercin Karakaş	mercin	Düzel Sil

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

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>
```

```

<tr>
    <td>Kullanıcı adı</td>
    <td><input type='text' name='kadi' class='form-control' /></td>
</tr>
<tr>
    <td>$ifre</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 EKLEMELİ PHP KODLARI

Kullanıcı **ekle.php** sayfasında bilgileri girip **Kaydet** butonuna tıklayınca aşağıdaki PHP kodları MySQL veritabanındaki **kullanicilar** tablosuna bu bilgileri kaydedecekdir. 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>";
        }
    }
    // hata varsa göster
    catch(PDOException $exception){
        die('HATA: ' . $exception->getMessage());
    }
}
?>

```

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

PHP - PDO Projesi

Kullanıcı Ekle

Ad ve Soyad	Ertan Yıldız
Kullanıcı adı	ertan
Şifre	*****

Kaydet **Kullanıcı listesi**

© Murat SARIÖZ - Mercin KARAKAŞ

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

PHP - PDO Projesi

Kullanıcı Ekle

Kullanıcı kaydedildi.

Ad ve Soyad	
Kullanıcı adı	
Şifre	

Kaydet **Kullanıcı listesi**

© Murat SARIÖZ - Mercin KARAKAŞ

KULLANICI GÜNCELLEME (DÜZELTME) SAYFASINI OLUŞTURMAK

Kullanıcı listeleme sayfasında **Düzzelt** işlem butonu tıklanınca kayıtların bilgilerini bir form ile görüntülemek ve güncellemek için kullanıcıyı yönlendireceğimiz sayfayı **duzzelt.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 -->

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

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

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

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

```

GÜNCELLENECEK KULLANICI BİLGİLERİNI VERİTABANINDAN OKUMAK

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

Aşağıdaki kodları **<!-- Kullanıcı 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'];
    }
    // hata varsa 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 **<!-- Kullanıcı bilgilerini düzeltme HTML formu burada yer alacak -->** açıklama satırının yerine yazın.

```
<!-- kayıt bilgilerini güncelleyebileceğimiz HTML formu -->
<form action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>" method="post">
    <table class='table table-hover table-responsive table-bordered'>
        <tr>
            <td>Ad ve Soyad</td>
            <td><input type='text' name='adsoyad' value="<?php echo htmlspecialchars($adsoyad, ENT_QUOTES); ?>" class='form-control' /></td>
        </tr>
        <tr>
            <td>Kullanıcı adı</td>
            <td><input type='text' name='kadi' value="<?php echo htmlspecialchars($kadi, ENT_QUOTES); ?>" class='form-control' /></td>
        </tr>
        <tr>
            <td>$sifre</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 değiştirdiğinde **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);
```

```

// soruyu 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 **Düzelt** butonunu tıklayın ve kullanıcı güncelleme sayfasını görüntüleyin. Bilgilerde değişiklik yaparak **Kaydet** butonunu tıklayın.

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");
        exit;
    }

    if (isset($_SESSION['son_islem']) && (time() - $_SESSION['son_islem'] > 600)) {
        // son işleminden beri geçen süreyi kontrol et. Oturum süresi: 10 dakika
        (10*60sn=600sn)
        session_unset();      // $_SESSION değişkenlerini sıfırla
        session_destroy();   // oturumu sonlandır
        header("Location: /proje/admin/login.php");
        exit;
    }
    $_SESSION['son_islem'] = time(); // son işlem zamanını kaydet
?>
```

Not: **admin/urun/sil.php**, **admin/urun/coklusil.php**, **admin/kategori/sil.php** ve **admin/kullanici/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>
</body>
</html>

```

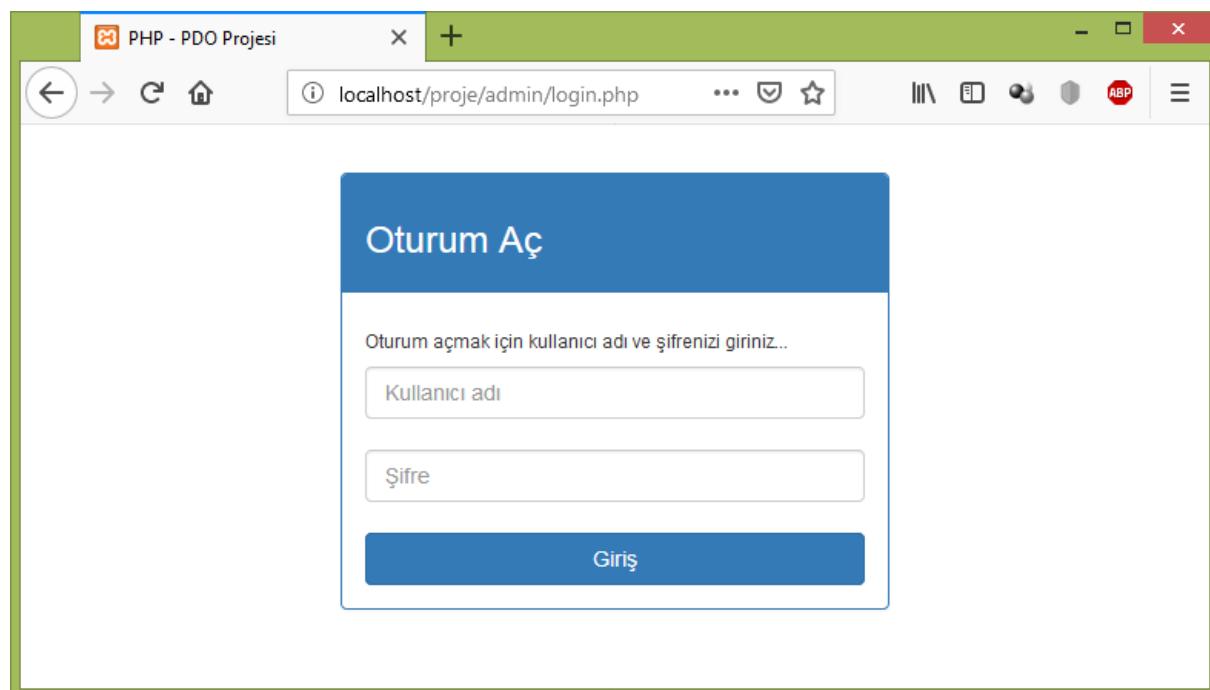
Oturum Açı formunun sayfada ortalanması için ilk **<div>** etiketine stil uygulayacağız. Bunun için **content/css** klasöründeki **style.css** dosyasını açın ve içine aşağıdaki satırları ekleyin.

```

.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 değiştirelim.

```

<?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");
                exit;
            }
        }
        // hata varsa 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");
        exit;
    }
}
```

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>
```

The screenshot shows a web browser window with the following details:

- Title Bar:** PHP - PDO Projesi
- Address Bar:** localhost/proje/admin/index.php
- Header:** PHP - PDO Projesi (highlighted), Anasayfa, Ürünler, Kategoriler, Kullanıcılar, Oturumu kapat
- Main Content:**
 - ## 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.
- Page Footer:** © Murat SARIÖZ - Mervin KARAKAŞ
localhost/proje/admin/login.php?cikis=1

EK GÖREVLER

- Kategori sayfalarındaki butonlara **Glyphicon** ekleyin.
- Ürün düzenleme sayfasında ürün resminin de değiştirilmesini 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 (aciklama)** alanı için **CKEditor** desteği ekleyin.
- Bütün uyarı ve onay iletilerini **SweetAlert** eklentisini kullanarak yeniden düzenleyin.
- Ürünler için birden fazla resim eklemeyi gerçekleştirsin. (Bunun için yeni bir tablo oluşturmanız gerekecek.)

PHP- PDO PROJESİ

(Kullanıcı Arayüzü)

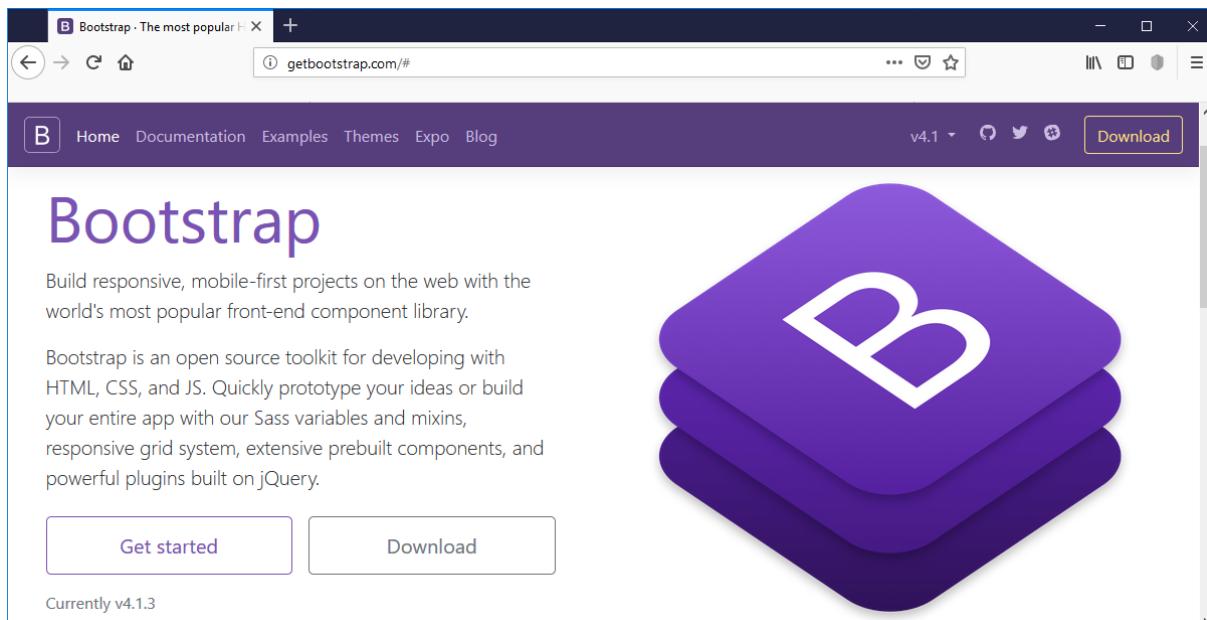
ADIM ADIM DİNAMİK SİTE TASARLAMA

Mercin KARAKAŞ – Murat SARIÖZ

BÜYÜKÇEKMECE SUDİ ÖZKAN MESLEKİ VE TEKNİK ANADOLU LİSESİ

KULLANICI ARAYÜZÜNÜ OLUŞTURMAK

Yönetici arayüzüne 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 kopyaladığını 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>
```

WEB SAYFALARININ ANA HTML ÇATISINI OLUŞTURMAK

İşe başlamadan önce tasarlayacağımız web sitesinin ana sayfa taslağını belirlememiz lazım. Sitemizin ana sayfa taslağı aşağıdaki gibi olacaktır.

Duyuru satırı

Kampanya: 100₺ üzeri alışverişlerinizde ücretsiz kargo

Sepetiniz 4

İstediğin Burada

Ürün ara... Ara

Anasayfa Ürünler Kategoriler Hakkımızda

Logo, arama ve menü satırı

Slider satırı

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

Altbilgi satırı

İstediğin Burada'ya Hoş Geldiniz

İstediğin Burada olarak ilk öncelikimiz 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 faydalananız için çalışıyoruz. Türkiye'de değer kazanan bir marka olarak, uluslararası bir marka olma yolunda hızla ilerliyoruz. Kurulumuzun 10. yılını kutladığımız bu günlerde, heyecanı ve mutluluğu yaşıyoruz.



Sorunsuz İade
Bütün ürünlerde 7 gün içinde
Ücretsiz İade



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
Büyücekmece/Istanbul
Telefon: (0212) 885 98 82
Faks: (0212) 885 40 30
Web: <http://sudiozkanmtal.meb.k12.tr>
Eposta: 7-24@sudiozkanmtal.meb.k12.tr

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: 70px;
    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.

proje klasöründe **header.php** ve **footer.php** isimli iki yeni dosya oluşturun.

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 olduğundan hepsini **proje** klasöründe yeni oluşturduğumuz **header.php** dosyasında toplayacağız. Öncelikle **index.php** dosyasında **<body>** etiketine kadar olan kodları alıp (**<body>** etiketi dahil) **header.php** dosyasına taşıyın.

```
<!-- header.php sayfa kodları -->
<!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">
      <!-- Simgə 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 -->
```

Ardından **index.php** sayfasını aşağıdaki şekilde yeniden düzenleyin.

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

<!-- Sayfa içeriği burada yer alacak -->

</body>
</html>
```

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% 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>
```

```

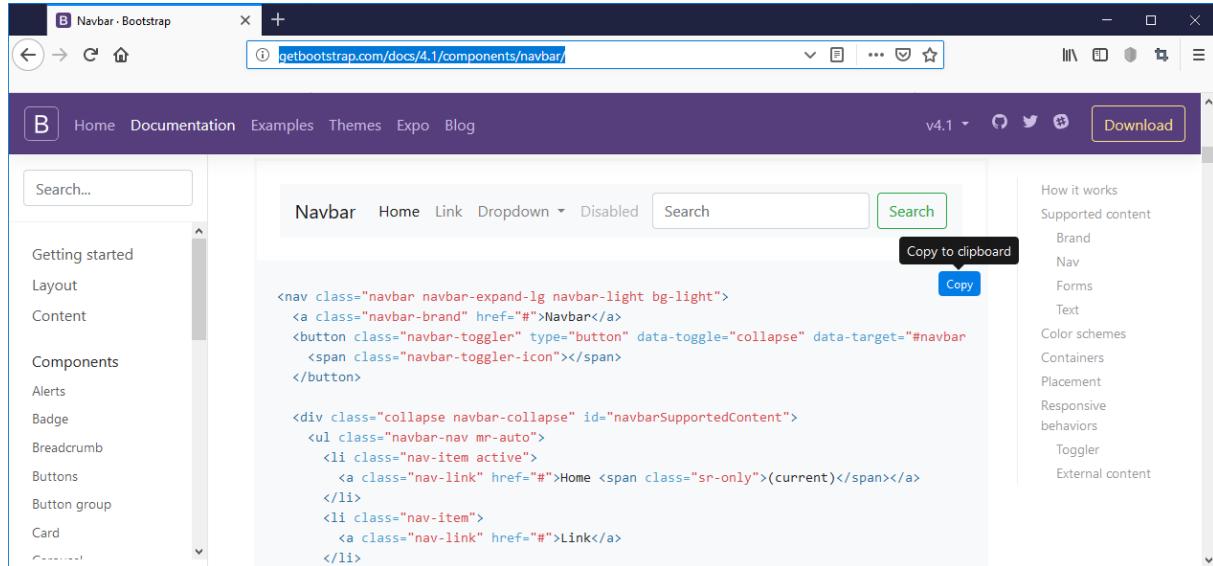
        </span> Sepetiniz
        <span class="badge badge-light">4 <!-- Sepetteki ürün sayısı --> </span>
    </a>
</div>
</div>
</div>

```

Not: Sepet işlemleriyle ilgili sayfanın tasarımları 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

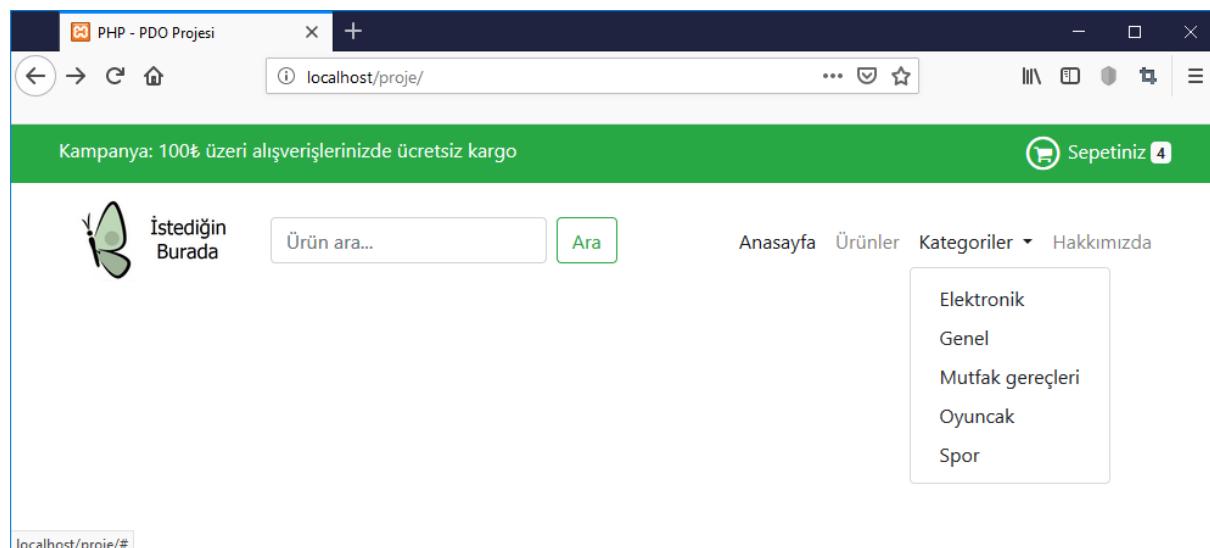
Ardından `<!-- Burada katogorileri listeleyen PHP kodları yer alacak -->` açıklama satırının yerine aşağıdaki PHP kodlarını yazın.

```

<?php
// veritabanı yapılandırma dosyasını dahil et
include 'config/vtabani.php';
// kayıt listeleme sorusu
$sorgu='select id, kategoriadi from kategoriler order by kategoriadi';
$stmt = $con->prepare($sorgu); // soruyu hazırla
$stmt->execute(); // soruyu 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']}";
}
?>

```

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



ALTBİLGİ (FOOTER.PHP) SAYFASINI OLUŞTURMAK

Sayfaların sonunda bir altbilgi alanımız olacak. Bunun için **index.php** dosyasında **</body>** etiketinden itibaren olan HTML kodlarını **proje** klasöründe yeni oluşturduğunuz **footer.php** dosyasına taşıyın.

```
<!-- footer.php sayfa kodları -->

    <!-- 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 mb-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 mb-4">
            <h4 class="text-white">Fırsat Ürünleri</h4>

                <!-- Fırsat ürünlerini listeleyen PHP kodları burada yer alacak -->

        </div>
        <div class="col-md-4 mb-4">
            <h4 class="text-white">İletişim</h4>
            <p class="text-light">Adres: <br>Celaliye Mahallesi, Cumhuriyet Cd. No:26
<br>Büyükköy/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>

<!-- Kullanıcı tanımlı JavaScript kodları burada yer alacak -->

</body>
</html>
```

Gördüğü gibi altbilgi alanı üç bölümden oluşmaktadır. Orta bölümde en düşük fiyatlı üç fırsat ürünü listelenenecek. 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-3">
```

```


</div>
<div class="col-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 }>
?>

```

Ardından **index.php** sayfasını aşağıdaki şekilde yeniden düzenleyin.

```

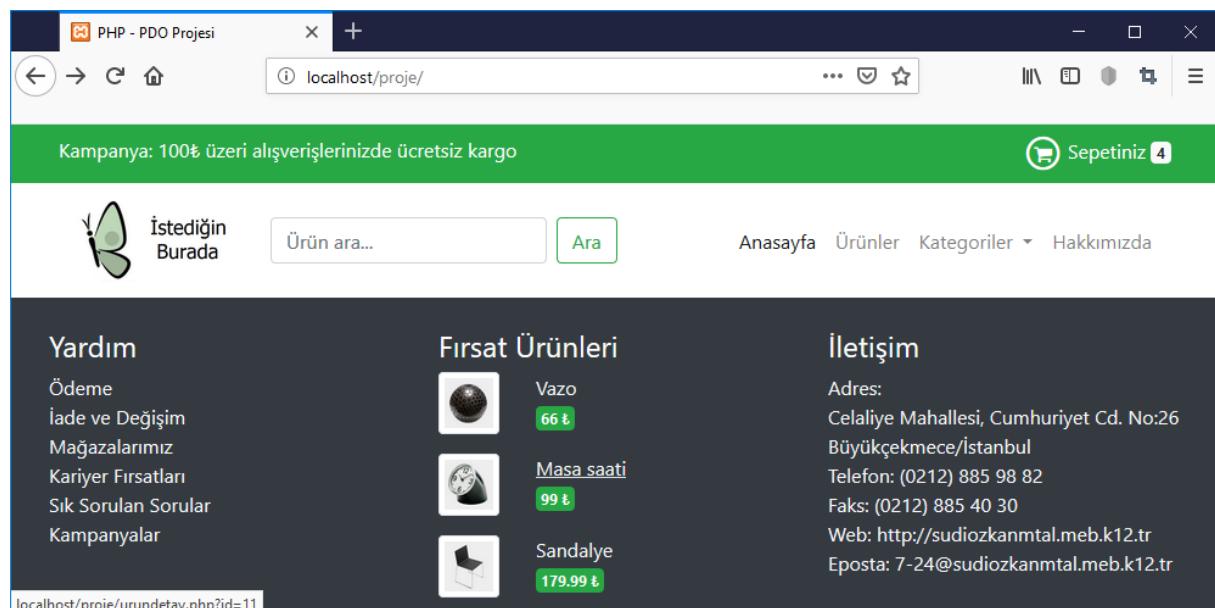
<?php include "header.php"; ?>

<!-- Sayfa içeriği burada yer alacak -->

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

```

Tarayıcınızın adres satırına **http://localhost/proje** yazarak ana sayfayı görüntüleyin.



ANA SAYFAYI (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 -->

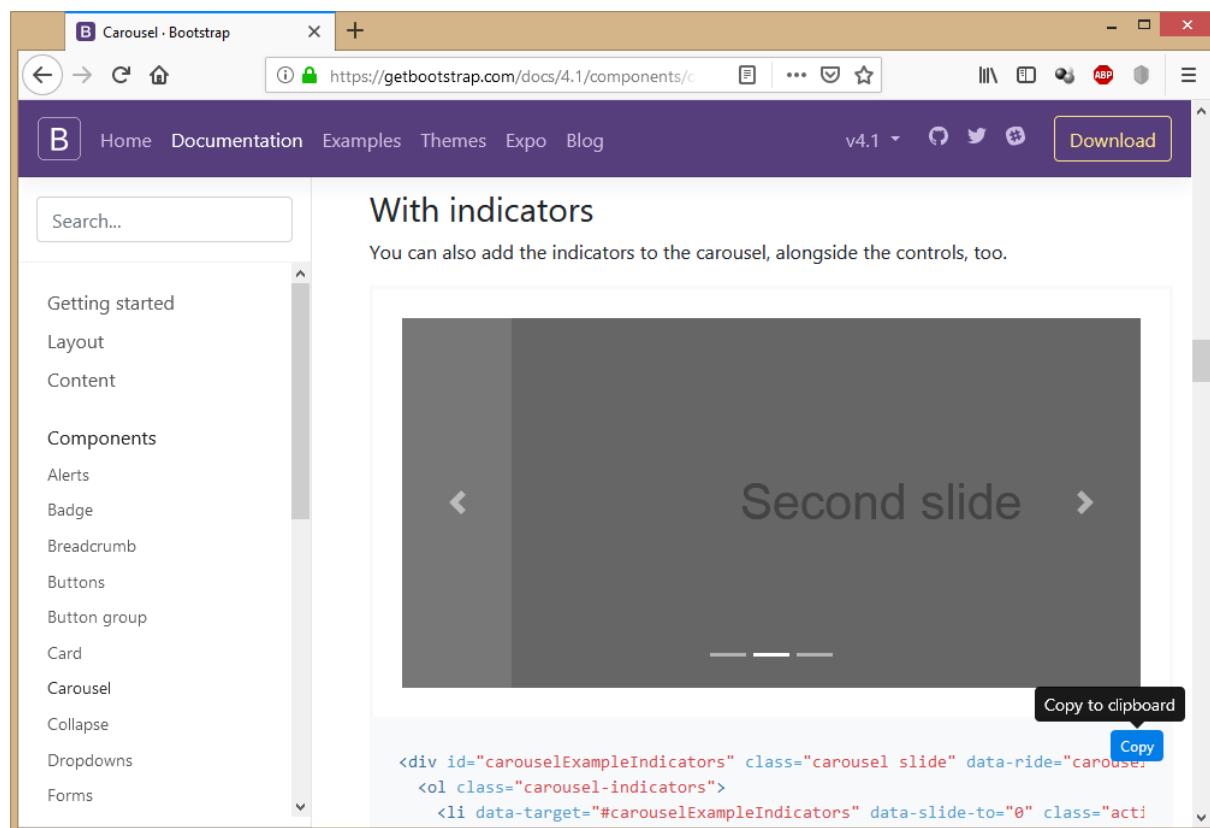
    <!-- Fırsat satırı kodları burada yer alacak -->

    <!-- 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>
```

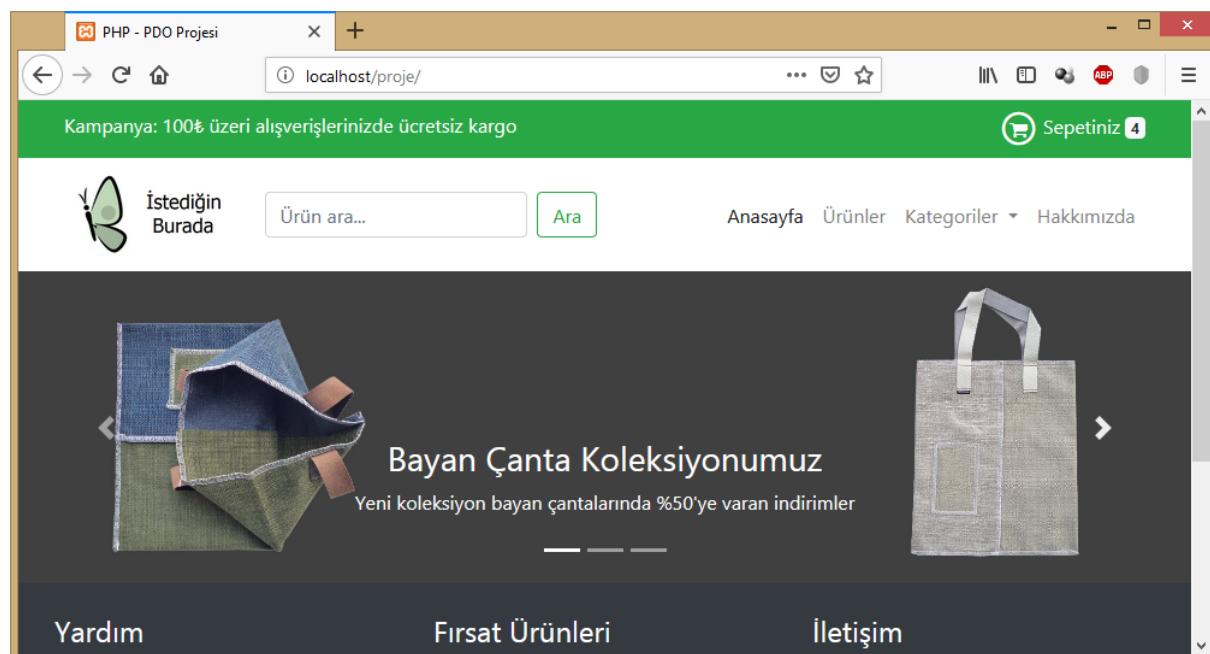
```

<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 fiyatlar</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** yazarak ana sayfayı görüntüleyin.



KARŞILAŞMA SATIRINI OLUŞTURMAK

index.php dosyasını açın ve `<!-- Karşılıkla 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, satışlarımıza hız kesmeden sürdürmeye devam ederken, sektörün ihtiyaçları doğrultusunda, kaliteli ürün tedarik etmek ve ürünlerimizden daha uzun yıllar faydalamanız 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 class="col-md-8">
        <h5>Sorunsuz İade</h5>
        <p>Bütün ürünlerde 7 gün içinde ücretsiz iade</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-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₺ üzeri alışverişlerinizde yurt içi ücretsiz kargo</p>
      </div>
    </div>
  </div>
</div>
```

Not: Font Awesome kütüphanesindeki simgeleri kullanabilmek için **header.php** dosyasında simge bağlantısını önceden tanımladığımıza 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** yazarak ana sayfayı görüntüleyin.

The screenshot shows a web browser window titled "PHP - PDO Projesi". The address bar displays "localhost/proje/". The main content area features three dark-colored wall clocks arranged horizontally. To the right of the clocks, the text "Saat Koleksiyonumuz" is displayed in orange. Below the clocks, a yellow banner contains the text "Yeni ve modern duvar saatlerinde %20'ye varan indirimler". Below this, a large heading "İstediğin Burada'ya Hoş Geldiniz" is centered. A paragraph follows, stating: "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 faydalamanız 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." At the bottom, there are three service icons: a circular icon with a refresh symbol for "Sorunsuz İade" (Return policy), a card icon for "Güvenli Ödeme" (Safe payment), and a truck icon for "Ücretsiz Kargo" (Free shipping). The background of the page has a light green gradient.

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 satırı kodları burada yer alacak -->` açıklama satırının yerine yapıştırın.

The screenshot shows a web browser window titled "Cards · Bootstrap". The address bar displays "https://getbootstrap.com/docs/4.1/components/card/". The page has a purple header with navigation links: Home, Documentation, Examples, Themes, Expo, Blog, and a Download button. On the left, there is a sidebar with a search bar and a list of components: Getting started, Layout, Content, Components, Alerts, Badge, Breadcrumb, Buttons, Button group, Card, Carousel, and Collapse. The main content area shows a card component with a placeholder image (286x180), a title "Card title", and some sample text. At the bottom, there is a code snippet for the card component. A tooltip "Copy to clipboard" is shown over the "Copy" button in the bottom right corner of the code block.

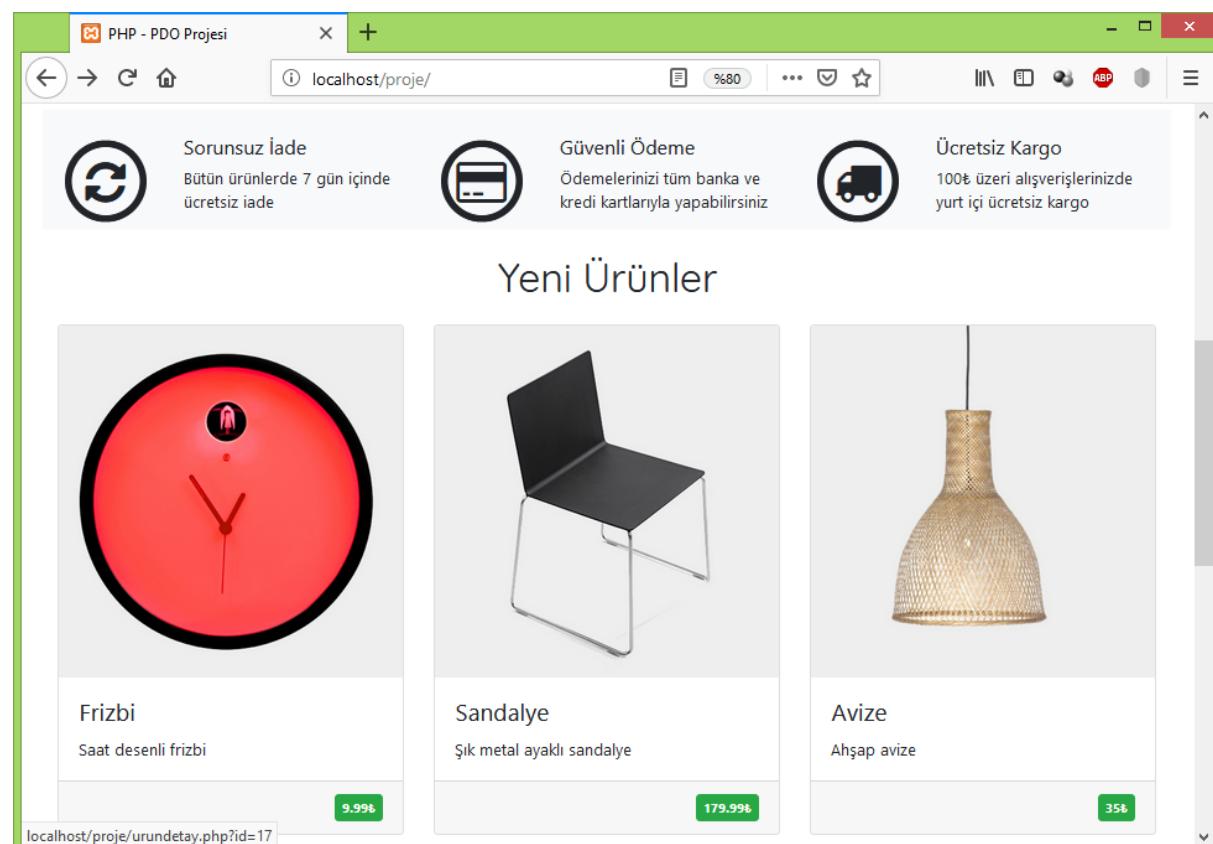
Card kodlarını aşağıdaki gibi veritabanına en son eklenen altı ürün kaydını listeleyen PHP döngüsü içinde olacak şekilde düzenleyin.

```

<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 h-100">
                    <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>
        <?php }
    ?>
</div>

```

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



ÜRÜNLER SAYFASINI (URUNLER.PHP) OLUŞTURMAK

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

Kampanya: 100₺ üzeri alışverişlerinizde ücretsiz kargo

Sepetiniz 4

İstediğin Burada

Ürün ara... Ara

Banner satırı

Kategoriler Hakkımızda

Kategori listesi

Ürün listesi

Sıralama seçenekleri

Fiyat seçenekleri

Kategoriler

- Elektronik 4
- Genel 4
- Mutfak gereçleri 3

Sıralama

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

Fiyat Aralığı

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

Banner satırı

Ürünler

Frizbi
Saat desenli frizbi
9.99₺

Sandalye
Şık metal ayaklı sandalye
179.99₺

Avize
Ahşap avize
35₺

Bayan çantası
Siyah omuz çantası
349₺

Vazo
Küre şekilli vazo
66₺

Sırt çantası
Spor çantası
199₺

Kol saati
Erkek kol saati
249₺

Masa saati
Analog masa saati
99₺

Yastık
İyi bir uyu için.
25.56₺

Fırsat Ürünleri

Suç bandı
1.99₺

Çöp kovası
3.99₺

Gözlük
6₺

Yardım

Ödeme
İade ve Değişim
Mağazalarımız
Kariyer Fırsatları
Sık Sorulan Sorular
Kampanyalar

Fırsat Ürünləri

İletişim

Adres:
Celalîye Mahallesi, Cumhuriyet Cd. No:26
Büyükkemce/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 LİSTESİ 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** başlıklarını altındaki örnek kodlarından faydalanağınız.

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 sorusu
```

```

$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"]; ?>" 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 gereklidir. Bu eksiklikleri düzeltmek için kategori liste kodunu aşağıdaki gibi yeniden düzenleyin.

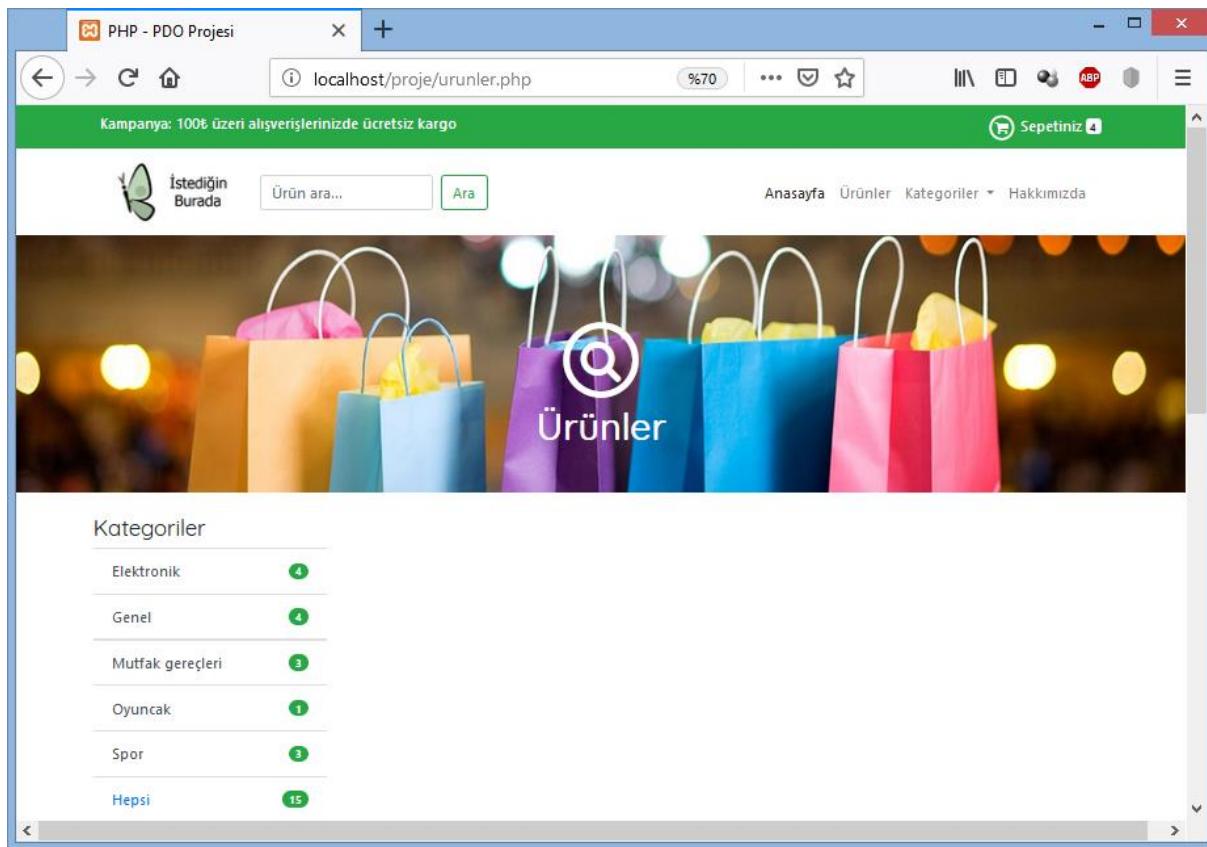
```

<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"><!--sıralama-->
  <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><!--/sıralama-->
```

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 gereklidir. Bu eksiklikleri düzeltmek için sıralama seçenekleri kodunu aşağıdaki gibi yeniden düzenleyin.

```
<div class="pt-4"><!--sıralama-->
  <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 gereklidir. Bu eksiklikleri düzeltmek için sıralama seçenekleri kodunu aşağıdaki gibi yeniden düzenleyin.

```

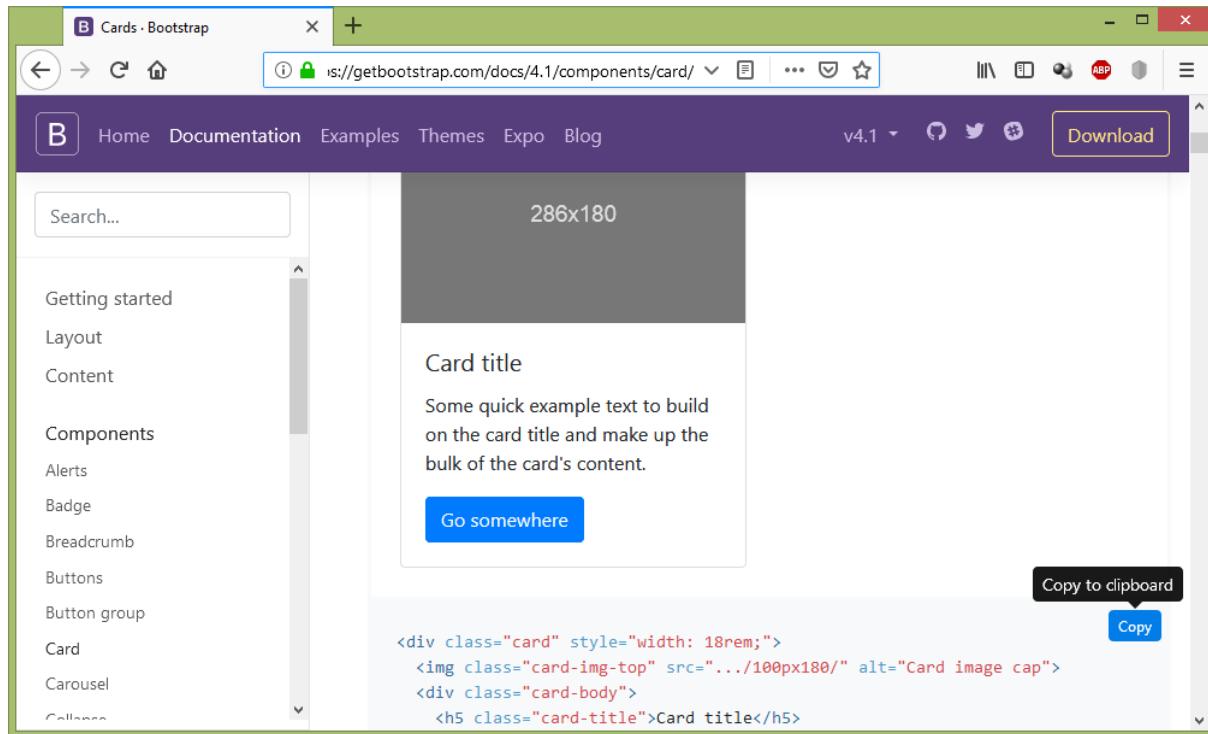
<div class="pt-4"><!--fiyat-->
  <h4 class="baslik">Fiyat Aralığı</h4>
  <div class="list-group list-group-flush">
    <!-- fiyat haricindeki seçenekleri de linke dahil etmek için -->
    <?php $parametre = "&aranan=$aranan&id=$kategori&siralama=$siralama"; ?>

    <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 LİSTESİ 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.\")";
```

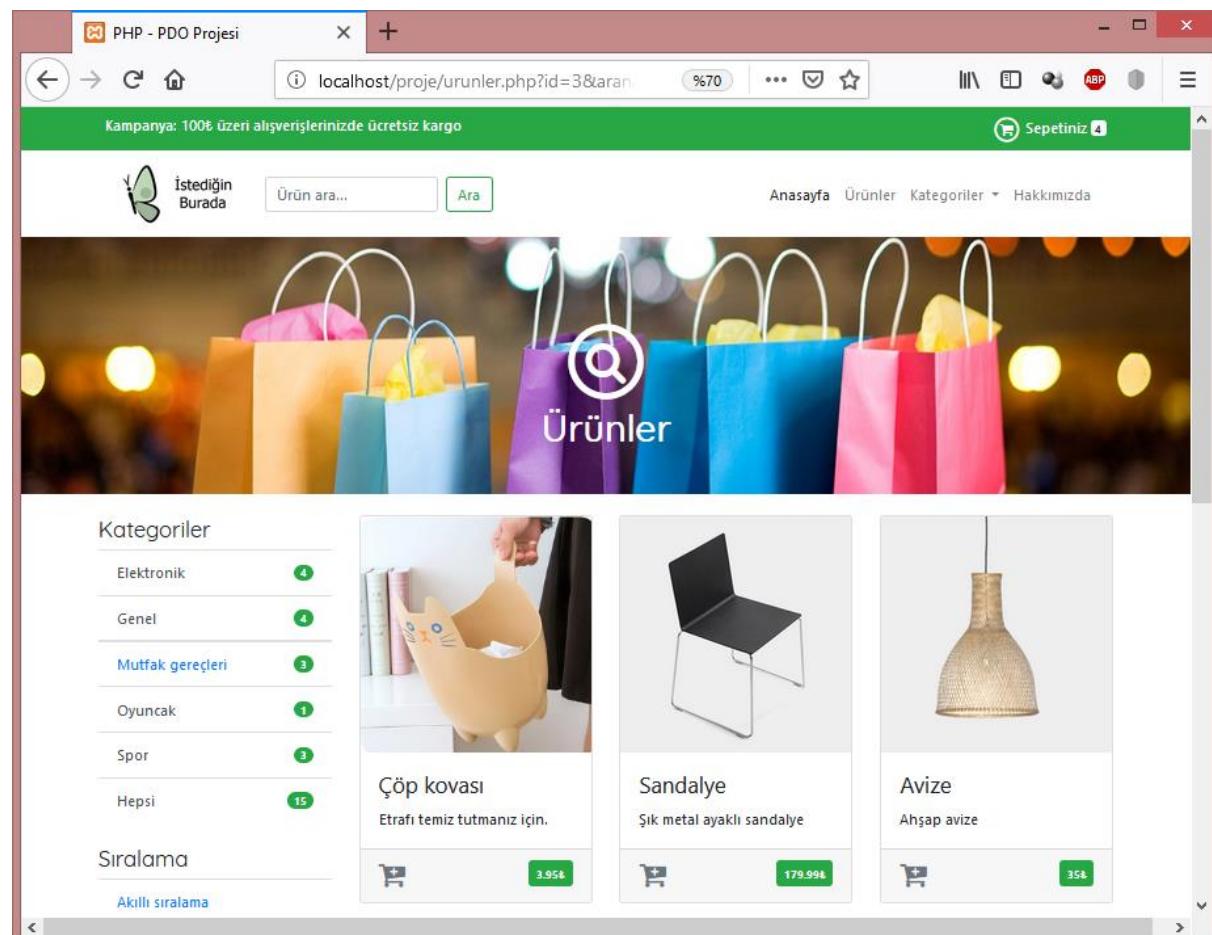
```

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 h-100">
            <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.

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 -->
        </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 LİSTELEMEK 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);
    }
    // hata varsa 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}'>" : "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 ?>₺</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($dzltm_tarihi)) ?></h6>

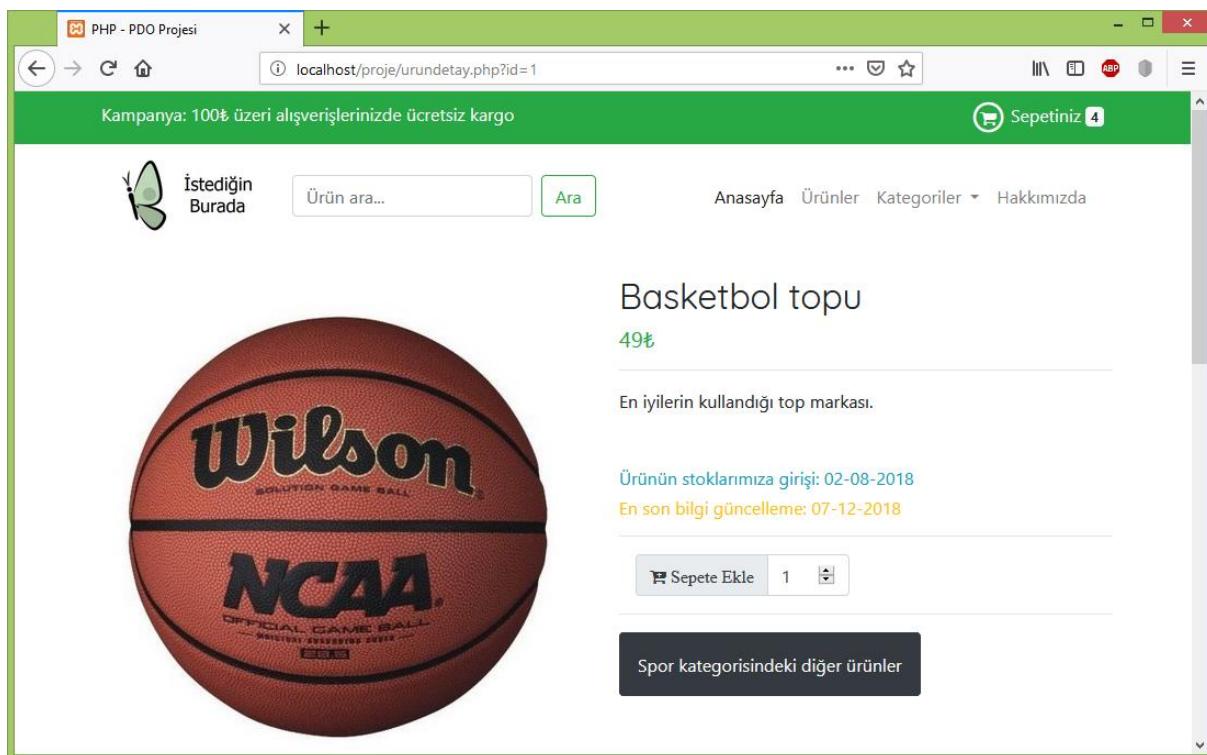
```

```

<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 diğer ürünler</a></p>

```

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



DİĞER ÜRÜNLERİ LISTELEMEMK

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

```

<h4 class="baslik mt-5">Diğer 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 h-100">
          <a href="urundetay.php?id=<?php echo $kayit['id']?>">
            ">

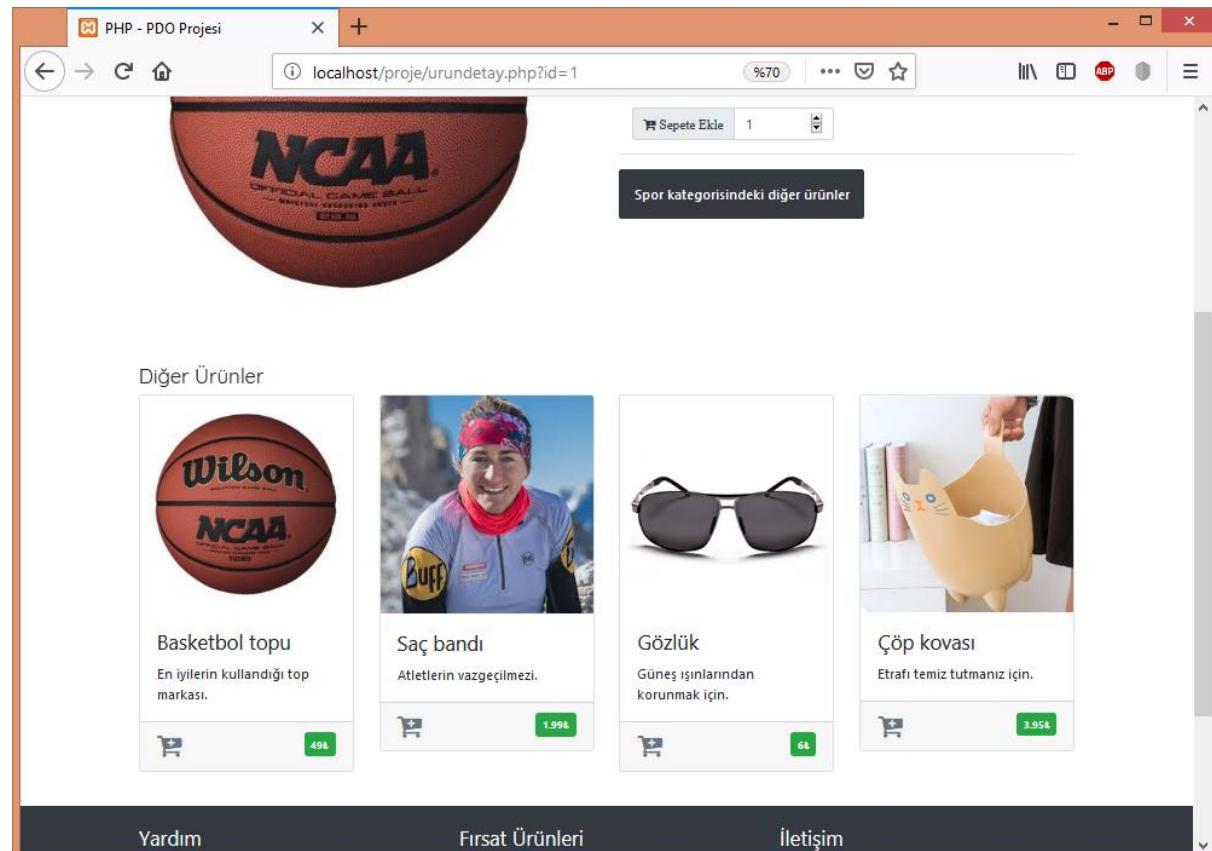
```

```

</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>
<?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 / şıkâyet 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 faydalananız 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 faydalanaçagız. <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 = $kimden.'<'.$_POST['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
$ismle = isset($_GET['islem']) ? $_GET['islem'] : "";

// eğer eposta.php sayfasından yönlendirme yapıldıysa
if($ismle == 'tamam') {
    echo "<div class='alert alert-success'>Mesajınız başarıyla iletildi.</div>";
}
else if($ismle == '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 ilgilendiğini 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">
<?php
// Sepetteki ürün sayısı
$urun_sayisi=count($_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 sayı = 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 ürünler.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(sayı + 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.

```

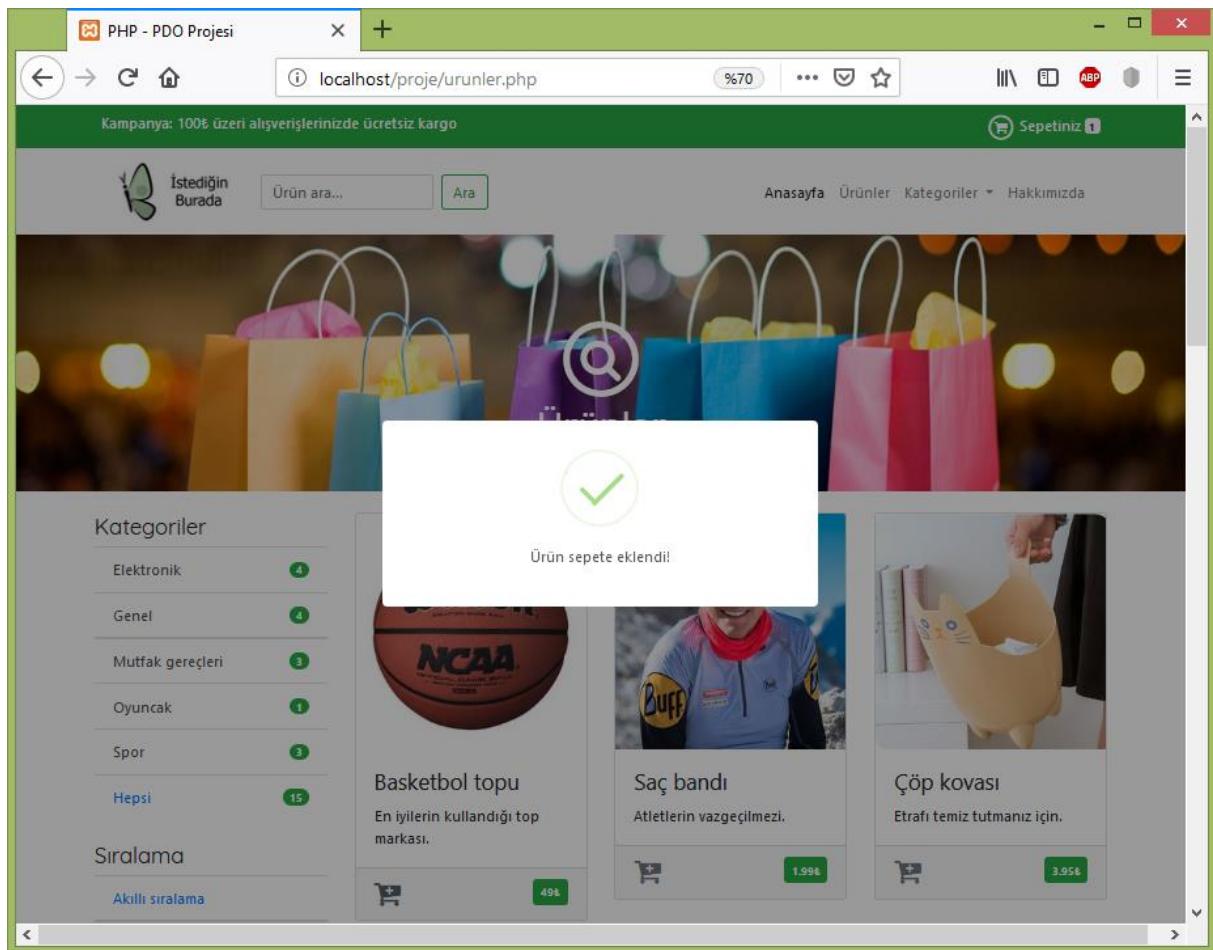
<?php
// oturum işlemlerini başlat
session_start();

// 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 GÖRÜNTÜLEMEMEK

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 ürünlerini 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 --&gt;
        &lt;!-- Sepet özeti görüntüleyen HTML tablosu burada yer alacak --&gt;

    &lt;/div&gt;

&lt;?php
}

include "footer.php"; ?&gt;</pre>
```

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 ürünler 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 LISTELEMEK

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

```
<!-- Sepetteki ürünlerini görüntüleyen HTML tablosu -->
<div class="container table-responsive">
    <div class="baslik">
        <h3>Sepet Özeti</h3>
    </div>
    <table class="table table-bordered sepet-tablo">
        <thead class="thead-dark">
            <tr class="row">
                <th class="col-md-2">Ürün görseli</th>
                <th class="col-md-5">Ürün adı</th>
                <th class="col-md-2">Fiyat</th>
                <th class="col-md-2">Adet</th>
                <th class="col-md-1">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 ürünlerini listeleyen döngü
while ($kayit = $stmt->fetch(PDO::FETCH_ASSOC)) {
    extract($kayit);
```

```
$adet = $_SESSION['sepet'][$id]['adet'];
$urun_sayisi += $adet;
$urun_toplami += $fiyat * $adet;
?>
<tr class="row">
    <td class="col-md-2">
        
    </td>
    <td class="col-md-5 text-left">
        <span><a href="urundetay.php?id=<?php echo $id; ?>" class="link2"><?php
echo $urunadi; ?></a></span>
    </td>
    <td class="col-md-2">
        <span><?php echo number_format($fiyat, 2, ',', '.'); ?>&#8378;</span>
    </td>
    <td class="col-md-2">
        <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>
    </td>
    <td class="col-md-1">
        <span><a href="#" class="link2 urun-sil" id="<?php echo $id; ?>"><i
class="fa fa-trash"></i></a></span>
    </td>
</tr>
<?php
} // döngüsü sonu
$kargo_ucreti = 5.99;
$toplam = $urun_toplami + $kargo_ucreti;
?>
```

SEPET ÖZETİNİ GÖRÜNTÜLEMEK

<!-- Sepet özettini görüntüleyen HTML tablosu burada yer alacak --> açıklama satırının yerine aşağıdaki kodları yazın.

```

        <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><!--/Sepet özeti-->
<form name="frm_sepet" method="post" action="satinal.php">
    <input type="hidden" name="uruntutari" value=<?php echo
$urun_toplami; ?>">
    <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 ürünü **Sepete Ekle** butonu ile sepete ekleyin. Ardından sayfanın üst kısmında yer alan duyuru satırındaki **Sepetiniz** 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	49,00₺	<input type="button" value="1"/>	
	Sırt çantası	199,00₺	<input type="button" value="1"/>	

Ürün Sayısı 2

Ürün Tutarı 248,00₺

Kargo Ücreti 5,99₺

Toplam 253,99₺

Satın al

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.

The screenshot shows a web browser window with a green header bar. The address bar says "localhost/proje/sepetim.php". The main content area is titled "Sepet Özeti" (Cart Summary). It lists two items: "Basketbol top" (Basketball) and "Sırt çantası" (Backpack). For the basketball, there is a green checkmark and the message "Ürün adedi güncellendi!" (Product quantity updated!). Below the table, there are summary details: "Ürün Sayısı" (Product Count) 2, "Ürün Tutarı" (Product Price) 248,00₺, "Kargo Ücreti" (Shipping Cost) 5,99₺, and "Toplam" (Total) 253,99₺. At the bottom is a green button labeled "Satin al" (Buy).

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.

```
// ürün-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 (SATINAL.PHP) OLUŞTURMAK

Sepetteki ürünleri listeleyen 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_uretici = $_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>  
  
<?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 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="odemeselekli" 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 **Satin al** butonuyla sayfayı görüntüleyin.

SİPARİŞ ONAYI SAYFASINI (SIPARIS_ONAY.PHP) OLUŞTURMAK

Satin 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

İstediğin 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	Fırsat Ürünleri	İletişim
Ödeme İade ve Değişim Mağazalarımız Kariyer Fırsatları Sık Sorulan Sorular Kampanyalar	 Saç bandı 1.99 ₺  Çöp kovası 3.95 ₺  Gözlük 6 ₺	Adres: Celaliye Mahallesi, Cumhuriyet Cd. No:26 Büyükçekmece/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.
- Ana sayfadaki (index.php) ürün kartlarına **Sepete Ekle** butonu ekleyin.
- Ürün Detay sayfasında ürünün birden fazla resminin görüntülenmesini sağlayın.
- Ürün (urunler) tablosuna **Ürün Detayı (urundetay)** alanı ekleyin ve Ürün Detay sayfasında bu alanın bilgilerini de görüntüleyin.
- Müşterilerin hesap oluşturabilmesini sağlayın. (Bunun için yeni bir tablo oluşturmanız gerekecek.)
- Müşterilerin siparişlerini takip edebilmelerini sağlayın. (Bunun için yeni bir tablo / tablolar oluşturmanız gerekecek.)

WEB SİTESİNİ YAYIMLAMAK

LİNKLERİ DÜZENLEMEK

Web sayfalarını yerel bilgisayarda oluştururken **proje** klasörünü kullandık. Web sitesini İnternette yayayılmayınca bu klasörün içindeki tüm dosyaları sunucunun bize sunduğu klasöre (htdocs, www vb.) kopyalayacağız. Bu yüzden **Yönetim Panelindeki** linkleri yeniden düzenlemeliyiz.

admin klasörü içindeki tüm web sayfalarında linkleri kontrol edin ve linklerin başındaki **/proje** ifadesini silin.

Eski link:

```
<link rel="stylesheet" href="/proje/content/css/style.css" />
```

Yeni link:

```
<link rel="stylesheet" href="/content/css/style.css" />
```

VERİTABANI BAĞLANTI PARAMETRELERİNİ AYARLAMAK

Sunucuda web sayfanızı yönetmenizi sağlayan yazılımla (genelde cPanel) yeni bir MySQL veritabanı oluşturun. Sunucudaki phpMyAdmin arayüzüyle de veritabanı tablolarınızı oluşturun.

Ardından **config** klasöründeki **vtabani.php** dosyasını açın ve parametre değişkenlerini sunucuda oluşturduğunuz MySQL veritabanına bağlanmak için uygun şekilde ayarlayın.

```
// veritabanı bağlantısı için gerekli parametreler
$host = "sunucu adresiniz";
$vt_adi = "veritabanınızın adı";
$kullanici_adi = "kullanıcı adınız";
$sifre = "şifreniz";
```

RESİM KLASÖRÜNE YAZMA İZNI VERMEK

Son olarak sunucuya tüm dosya ve klasörleri kopyaladıktan sonra **content/images** klasörüne yazma izni verin (chmod 777).

PHP - PDO PROJESİ

(Nesne Yönelimli Programlama – OOP)

ADIM ADIM DİNAMİK SİTE TASARLAMA

Mercin KARAKAŞ – Murat SARIÖZ

BÜYÜKÇEKMECE SUDİ ÖZKAN MESLEKİ VE TEKNİK ANADOLU LİSESİ

VERİTABANI İŞLEMLERİNDE SINIF KULLANIMI (OOP)

Nesneye yönelik programlamada (OOP) sınıf, bir nesnenin tüm özelliklerini ve eylemlerini tanımlayan şablondur. Sınıf oluşturmak için **class** anahtar kelimesi kullanılır.

Bir nesne oluşturulacağı zaman sınıfın verdiği bilgilere bakılır ve bu bilgiler doğrultusunda nesne oluşturulur. Sınıflar nesnelerin ilk oluşumu sırasında gerekli olan başlangıç durumlarını da belirtir. Sınıf içerisinde tanımlanan nesnelere sınıf nesnesi diyoruz. Sınıf nesneleri değişken, sabit veya metot olabilir. Metotlar sınıfların temel taşıdır. Metotlar eylem gerçekleştiren nesnelerdir.

Nesneler ancak sınıfların belirttiği özelliklere sahip olabilir. Sınıfın içerisindeki her şey bir nesnedir, örneğin değişken veya sabit tanımlamakla bir nesne oluşturmuş oluyoruz. Bir sınıfın yeni bir nesne türetmek için **new** anahtar kelimesi kullanılır.

Bir sınıfın türetilen nesneler türetildiği sınıfın tüm özelliklerine erişemezler. Nesnelerin hangi sınıf elemanlarına erişip erişemeyeceğini belirlemek için “**erişim belirleyici**” ler kullanılır. PHP’de erişim belirleyici olarak **public**, **private**, **extends** ve **protected** deyimleri kullanılmaktadır.

public sözcüğü bir nesneyi herkesin kullanımına açar. Yani bu nesneye her yerden ulaşılabilir. Bir değişkeni veya fonksiyonu **public** olarak tanımladığınızda o nesneye sınıf dışından ulaşabilirsiniz. Bu şekilde tanımlanan sınıfların özellikleri kalıtımla diğer nesnelere aktarılırlar.

private kelimesi, özel, gizli, şahsi anlamına gelir. **private** olarak tanımladığınız bir nesneye sadece tanımladığınız sınıf içerisinde ulaşabilirsınız.

protect ise korumalı anlamına gelir. **protect** olarak tanımladığınız nesneye bulunduğu sınıfın veya alt sınıfın ulaşabilirsınız.

Oluşturulacak bir sınıf bir başka sınıfın alt sınıfı olacaksa **extends** kullanılır.

\$this anahtar sözcüğü sınıf içerisindeki bütün nesnelere sınıf içerisinde (metot, değişken) ulaşmak için kullanılır.

-> işaretinin sınıf nesnelerine ulaşmak için kullanılır.

Bir nesne oluşturulduğu anda otomatik olarak çalıştırılan metotlara yapıcı metot diyoruz. Bu metotların kullanım amacı, sınıf elemanlarına ilk değer atamak veya sınıf için birtakım düzenlemeler yapmaktır. Yapıcı metotların geri dönüş değeri yoktur ve **public** olarak tanımlanmalıdır. Yapıcı metot oluşturmak için **_construct** deyimi kullanılır.

Nesneye yönelik programlamada da bir sınıfın özelliklerinin bir başka sınıf'a aktarılmasına kalıtım denir. **Kalıtım (inheritance)**, yeni sınıflar yazarken elimizdeki eski sınıfları kullanmamızı sağlar. Kalıtım, yazılımın yeniden kullanılmasına büyük ölçüde katkıda bulunur. Zaten kalıtım kavramının ortaya çıkmasının sebebi, özellikle büyük çaplı yazılım geliştirme çalışmalarında aynı kod parçalarının tekrar tekrar kullanılmasını önlemek ve yeni sınıfları var olanların üstüne inşa etmektir. Kalıtım yoluyla yeni bir sınıf oluşturmak için **extends** anahtar sözcüğü kullanılır.

Çok biçimlilik (Polymorphism) ise bir metodun farklı nesnelerde farklı sonuçlar üretmesidir. Bir temel sınıfın özellik ve metotlarını, temel sınıfın türetilen bir sınıf'a aktardığımızda temel sınıf'a ait metotları türemiş sınıf içerisinde de kullanabiliriz. Her iki sınıf da aynı metot bulunduğuanda, temel sınıf'a ait metotları türemiş sınıf'ta farklı biçimlerde tanımlayarak kullanmak çok biçimlilik olarak adlandırılır.

VERİTABANI BAĞLANTISI İÇİN KULLANILACAK SINIF YAPISINI OLUŞTURMAK

Proje klasörünün altındaki **config** klasöründe **vt_oop.php** isimli yeni bir dosya oluşturun ve aşağıdaki kodları içine yazarak kaydedin.

```
<?php
class Veritabani{
    // veritabanı bağlantısı için gerekli parametreler
    private $host = "localhost";
    private $vt_adi = "php_pdo_projesi";
    private $kullanici_adi = "root";
    private $sifre = "";
    public $baglanti;

    public function Baglan(){
        $this->baglanti = null;

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

        return $this->baglanti;
    }
}
?>
```

OTURUM AÇMA İŞLEMİNİ NESNEYE YÖNELİK DÜZENLEMEK

KULLANICILAR TABLOSUNUN SINIF YAPISINI OLUŞTURMAK

Oturum açma işleminde kullanıcı adı ve şifre kontrolünü yapmak için öncelikle **kullanicilar** tablosunun nesne sınıfını oluşturmalıyız. Bunun için projemizin **config** klasöründe **kullanici.php** isimli bir dosya oluşturalım. Dosya içine aşağıdaki kodları yazalım.

```
<?php
class Kullanici{
    // veritabanı bağlantısı ve tablo adı
    private $baglanti;
    private $tablo_adi = "kullanicilar";
    // nesne özellikleri (tablo alanları)
    public $id;
    public $adsoyad;
    public $kadi;
    public $sifre;
    public $sonay;

    public function __construct($vt){
        $this->baglanti = $vt;
    }

    public function kullanici_oku($kullanici, $parola){
        $sorgu = "SELECT kadi, sifre FROM ".$this->tablo_adi." WHERE kadi=:kadi
AND sifre=:sifre";

        $komut = $this->baglanti->prepare($sorgu);
```

```

    // parametreleri bağla
    $komut->bindParam(":kadi", $kullanici);
    $komut->bindParam(":sifre", $parola);

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

    return $komut;
}

```

OTURUM AÇMA SAYFASINI DÜZENLEMEK

login.php dosyasını açın ve veritabanı işlem kodlarını aşağıdaki gibi değiştirin.

Eski kod:

```

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

```

Yeni kod:

```

// kullanıcı kontrolü
if ($_POST){
    $kadi = $_POST["kadi"];
    $ksifre = $_POST["ksifre"];

    if (isset($kadi) && isset($ksifre)) {
        include_once '../config/vt_oop.php';
        include_once '../config/kullanici.php';
        try {
            // veritabanı ve nesneleri oluştur
            $veritabani = new Veritabani();

```

```

$vt = $veritabani->Baglan();
$kullanici = new Kullanici($vt);
// kullanıcıyı sorula
$stmt = $kullanici->kullanici_oku($kadi, $ksifre);
$sayi = $stmt->rowCount();

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

```

ÜRÜN SAYFALARINDAKİ VERİTABANI İŞLEMLERİNİ NESNEYE YÖNELİK DÜZENLEMEMEK

URUNLER TABLOSUNUN SINIF YAPISINI OLUŞTURMAK

Ürün sayfalarındaki veritabanı işlemlerini nesneye yönelik düzenlemeden önce **urunler** tablosunun nesne sınıfını oluşturacağız. Bunun için projemizin **config** klasöründe **urun.php** isimli bir dosya oluşturalım. Dosya içine aşağıdaki kodları yazalım.

```

<?php
class Urun{
    // veritabanı bağlantısı ve tablo adı
    private $baglanti;
    private $tablo_adi = "urunler";
    // nesne özellikleri
    // tablo alanları
    public $id;
    public $urunadi;
    public $fiyat;
    public $aciklama;
    public $resim;
    public $kategori_id;
    public $giris_tarihi;
    public $dzltm_tarihi;
    // yardımcı özellikler
    public $kategoriadi;

    public function __construct($vt){
        $this->baglanti = $vt;
    }

    public function urun_oku(){
        // sorgu
        $sorgu = "SELECT k.kategoriadi, u.id, u.urunadi, u.aciklama, u.fiyat,
u.kategori_id, u.giris_tarihi, u.dzltm_tarihi, u.resim FROM ".$this->tablo_adi." u
LEFT JOIN Kategoriler k ON u.kategori_id = k.id WHERE u.id = ? LIMIT 0,1";
        $komut = $this->baglanti->prepare($sorgu);
        $komut->bindParam(1, $this->id);
        // sorguyu çalıştır
        $komut->execute();
        $kayit = $komut->fetch(PDO::FETCH_ASSOC);
        // veritabanı kayıtlarını döndür
        //return $kayit;
        $this->urunadi = $kayit['urunadi'];
        $this->fiyat = $kayit['fiyat'];
        $this->aciklama = $kayit['aciklama'];
    }
}

```

```

        $this->resim = $kayit['resim'];
        $this->kategori_id = $kayit['kategori_id'];
        $this->giris_tarihi = $kayit['giris_tarihi'];
        $this->dzltm_tarihi = $kayit['dzltm_tarihi'];
        $this->kategoriadi = $kayit['kategoriadi'];
    }

    public function urunleri_oku($arama_sarti, $ilk_kayit_no, $sayfa_kayit_sayisi){
        // sorgu
        $sorgu = "SELECT k.kategoriadi, u.id, u.urunadi, u.aciklama, u.fiyat,
u.kategori_id, u.giris_tarihi FROM ".$this->tablo_adi." u LEFT JOIN kategoriler k
ON u.kategori_id = k.id WHERE u.urunadi LIKE ? OR u.aciklama LIKE ? ORDER BY
u.urunadi ASC LIMIT ?, ?";

        $komut = $this->baglanti->prepare($sorgu);
        // parametre değerlerini ata
        //{$arama_sarti}%" ;
        $komut->bindParam(1, $arama_sarti);
        $komut->bindParam(2, $arama_sarti);
        $komut->bindParam(3, $ilk_kayit_no, PDO::PARAM_INT);
        $komut->bindParam(4, $sayfa_kayit_sayisi, PDO::PARAM_INT);
        // sorguyu çalıştır
        $komut->execute();
        // veritabanı kayıtlarını döndür
        return $komut;
    }

    public function urun_ekle(){
        // sorgu
        $sorgu = "INSERT INTO ".$this->tablo_adi." SET urunadi = :urunadi, fiyat =
:fiyat, aciklama = :aciklama, resim = :resim, kategori_id = :kategori_id,
giris_tarihi = :giris_tarihi";

        $komut = $this->baglanti->prepare($sorgu);
        // parametre değerlerini ata
        $komut->bindParam(":urunadi", $this->urunadi);
        $komut->bindParam(":fiyat", $this->fiyat);
        $komut->bindParam(":aciklama", $this->aciklama);
        $komut->bindParam(":resim", $this->resim);
        $komut->bindParam(":kategori_id", $this->kategori_id);
        $komut->bindParam(":giris_tarihi", $this->giris_tarihi);
        // sorguyu çalıştır
        if($komut->execute()){
            return true;
        }
        else{
            return false;
        }
    }

    public function urun_duzelt(){
        $sorgu = "UPDATE ".$this->tablo_adi." SET urunadi = :urunadi, fiyat = :fiyat,
aciklama = :aciklama, kategori_id = :kategori_id WHERE id = :id";
        $komut = $this->baglanti->prepare($sorgu);

        // parametre değerlerini ata
        $komut->bindParam(':urunadi', $this->urunadi);
        $komut->bindParam(':fiyat', $this->fiyat);
        $komut->bindParam(':aciklama', $this->aciklama);
        $komut->bindParam(':kategori_id', $this->kategori_id);
        $komut->bindParam(':id', $this->id);
        // sorguyu çalıştır
        if($komut->execute()){
            return true;
        }
    }
}

```

```

        else{
            return false;
        }
    }

    public function urunu_sil(){
        $sorgu = "DELETE FROM ".$this->tablo_adi." WHERE id = ?";
        $komut = $this->baglanti->prepare($sorgu);
        $komut->bindParam(1, $this->id);
        $sonuc = $komut->execute();

        if($sonuc){
            return true;
        }
        else{
            return false;
        }
    }

    public function urunleri_sil($ids){
        $sonuc = $this->baglanti->query("DELETE FROM ".$this->tablo_adi." WHERE id IN ($ids)");

        if($sonuc){
            return true;
        }
        else{
            return false;
        }
    }

    public function urun_say_ara($arama_sarti){
        // sorgu
        $sorgu = "SELECT COUNT(*) as kayit_sayisi FROM ".$this->tablo_adi." u LEFT
JOIN kategoriler k ON u.kategori_id = k.id WHERE u.urunadi LIKE ?";

        $komut = $this->baglanti->prepare( $sorgu );
        // parametre değerlerini ata
        // $arama_sarti = "%{$arama_sarti}%";
        $komut->bindParam(1, $arama_sarti);
        $komut->execute();
        $kayit = $komut->fetch(PDO::FETCH_ASSOC);

        // kayıt sayısını döndür
        return $kayit['kayit_sayisi'];
    }

    public function urun_say(){
        $sorgu = "SELECT id FROM ".$this->tablo_adi."";

        $komut = $this->baglanti->prepare( $sorgu );
        $komut->execute();

        $sayi = $komut->rowCount();
        // kayıt sayısını döndür
        return $sayi;
    }

    //php'de fonksiyonları 'overLoad' etmek
    public function __call($name, $args) {
        switch ($name) {
            case 'urun_sayisi':
                switch (count($args)) {
                    case 0:
                        return $this->urun_say();

```

```

        case 1:
            return call_user_func_array(array($this, 'urun_say_ara'),
$args);
    }
    case 'urun_sil':
        switch (count($args)) {
            case 0:
                return $this->urunu_sil();
            case 1:
                return call_user_func_array(array($this, 'urunleri_sil'),
$args);
        }
    }
}

```

KATEGORILER TABLOSUNUN SINIF YAPISINI OLUŞTURMAK

Son olarak **kategoriler** tablosunun nesne sınıfını oluşturacağız. Bunun için projemizin **config** klasöründe **kategori.php** isimli bir dosya oluşturalım. Dosya içine aşağıdaki kodları yazalım.

```

<?php
class Kategori{
    // veritabanı bağlantısı ve tablo adı
    private $baglanti;
    private $tablo_adi = "kategoriler";
    // nesne özellikleri (tablo alanları)
    public $id;
    public $kategoriadi;
    public $aciklama;

    public function __construct($vt){
        $this->baglanti = $vt;
    }

    // drop-down Listesi için
    function kategorileri_oku(){
        //select all data
        $sorgu = "SELECT id, kategoriadi, aciklama FROM ".$this->tablo_adi." ORDER BY
kategoriadi";
        $komut = $this->baglanti->prepare($sorgu);
        $komut->execute();

        return $komut;
    }
}
?>

```

SAYFALAR İÇİN ORTAK SINIF DOSYALARINI EKLEMEMEK

Nesneye yönelik programlama işlemlerinde kullanacağımız ortak sınıf dosyalarını sayfalarımızda ayrı ayrı tanımlamak yerine, sayfalar için ortak olan **header.php** dosyasında tanımlamak daha pratik olacaktır. **header.php** dosyasını açın ve kodların sonuna aşağıdaki satırları ekleyin.

```

<?php
function absolute_include($file) {
    $folder_depth = substr_count($_SERVER["PHP_SELF"] , "/");
    if($folder_depth == false) $folder_depth = 1;

    if($_SERVER['SERVER_NAME']=='localhost') {
        $path = str_repeat("../", $folder_depth - 2) . $file;
    }
    else {

```

```

    $path = str_repeat("../", $folder_depth - 1) . $file;
}

return $path;
}

include('config/vt_oop.php');
include('config/urun.php');
include('config/kategori.php');

// veritabanı ve nesneleri oluştur
$veritabani = new Veritabani();
$vt = $veritabani->Baglan();
$urun = new Urun($vt);
$kategori = new Kategori($vt);
?>

```

ÜRÜN LISTELEME SAYFASINI DÜZENLEMEK

admin klasörü altındaki **urun** klasörü içinde yer alan **liste.php** isimli dosyayı aşağıdaki şekilde düzenleyin.

Önce veritabanından kayıtları okuyan kodları değiştirin:

Eski kod:

```

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

// sayfada görüntülenecek kayıtları seç
$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";

$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", $arama_sarti);
$stmt->execute();

```

Yeni kod:

```

// sayfada görüntülenecek kayıtları seç
$stmt = $urun->urunleri_oku($arama_sarti, $ilk_kayit_no, $sayfa_kayit_sayisi);

```

Ardından sayfa numaralandırma işlemi için gerekli olan kayıt sayısını hesaplayan kodları değiştirin:

Eski kod:

```

// SAYFALANDIRMA
// toplam kayıt sayısını hesapla

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

// sorguyu çalıştır
$stmt->execute();

// kayıt sayısını oku
$kayit = $stmt->fetch(PDO::FETCH_ASSOC);

```

```
$kayit_sayisi = $kayit['kayit_sayisi'];
```

Yeni kod:

```
// SAYFALANDIRMA
// toplam kayıt sayısını hesapla

$kayit_sayisi=$urun->urun_sayisi($arama_sarti);
```

ÜRÜN SİLME SAYFASINI DÜZENLEMEK

admin klasörü altındaki **urun** klasörü içinde yer alan **sil.php** isimli dosya kodlarındaki **try** kontrol bloğunu aşağıdaki şekilde düzenleyin.

Eski kod:

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

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

    // 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'];

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

    // sorguyu çalıştır
    if($stmt->execute()){

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

        // kayıt Listeleme sayfasına yönlendir
        // ve kullanıcıya kaydın silindiğini
        header('Location: liste.php?islem=silindi');
    } // veya silinemediğini bildir
    else{
        header('Location: liste.php?islem=silinemedi');
    }
}
```

Yeni kod:

```

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

    include('../config/vt_oop.php');
    include('../config/urun.php');

    // veritabanı ve nesneleri oluştur
    $veritabani = new Veritabani();
    $vt = $veritabani->Baglan();
    $urun = new Urun($vt);

    $urun->id = $id;
    $urun->urun_oku();

    // okunan resim bilgilerini bir değişkene kaydet
    $resim = $urun->resim;

    if($urun->urun_sil()){
        // kayda ait resim varsa sunucudan sil
        if(!empty($resim)){
            $silincek_resim = "../content/images/".$resim;
            if (file_exists($silincek_resim)) unlink($silincek_resim);
        }
        // kayıt listeleme sayfasına yönlendir
        // ve kullanıcıya kaydın silindiğini
        header('Location: liste.php?islem=silindi');
    } // veya silinmediğini bildir
    else{
        header('Location: liste.php?islem=silinemedi');
    }
}

```

ÇOKLU ÜRÜN SİLME SAYFASINI DÜZENLEMEK

admin klasörü altındaki **urun** klasörü içinde yer alan **coklusil.php** isimli dosya kodlarını aşağıdaki şekilde düzenleyin.

Eski kod:

```

<?php

...

$id_dizisi = $_POST['id'];
$ids = implode(',', $id_dizisi);

include '../config/vtabani.php';

foreach ($id_dizisi as $id) {
    $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);

    $resim = $kayit['resim'];
    if(!empty($resim)){
        $silincek_resim = "../content/images/".$resim;
        if (file_exists($silincek_resim)) unlink($silincek_resim);
    }
}

```

```
$con->query("DELETE FROM urunler WHERE id IN ($ids)");
?>
```

Yeni kod:

```
<?php

...
$id_dizisi = $_POST['id'];
$ids = implode(',', $id_dizisi);

include('.../.../config/vt_oop.php');
include('.../.../config/urun.php');

// veritabanı ve nesneleri oluştur
$veritabani = new Veritabani();
$vt = $veritabani->Baglan();
$urun = new Urun($vt);

foreach ($id_dizisi as $id) {
    $urun->id = $id;
    $urun->urun_oku();

    $resim = $urun->resim;
    if(!empty($resim)){
        $silincek_resim = ".../.../content/images/".$resim;
        if (file_exists($silincek_resim)) unlink($silincek_resim);
    }
}

$sonuc = $urun->urun_sil($ids);
?>
```

ÜRÜN DETAY SAYFASINI DÜZENLEMEK

admin klasörü altındaki **urun** klasörü içinde yer alan **detay.php** isimli dosya kodlarını aşağıdaki şekilde düzenleyin.

Eski kod:

```
// aktif kayıt bilgilerini oku
try {
    // veritabanı bağlantı dosyasını çağır
    include '.../.../config/vtabani.php';

    // seçme sorgusunu hazırla
    $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'];
}

```

Yeni kod:

```

// aktif kayıt bilgilerini oku
try {
    $urun->id = $id;
    $urun->urun_oku();

    // tabloya yazılacak bilgileri değişkenlere doldur
    $urunadi = $urun->urunadi;
    $aciklama = $urun->aciklama;
    $fiyat = $urun->fiyat;
    $resim = $urun->resim;
    $kategoriadi = $urun->kategoriadi;
}

```

ÜRÜN EKLEME SAYFASINI DÜZENLEMEK

admin klasörü altındaki **urun** klasörü içinde yer alan **ekle.php** isimli dosyayı açın ve ilk önce kategorileri açılır liste kutusunda listeleyen kodları aşağıdaki şekilde düzenleyin.

Eski kod:

```

<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>

```

Yeni kod:

```

<tr>
    <td>Kategori</td>
    <td>
        <?php
            $stmt = $kategori->kategorileri_oku();

            $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>

```

Ardından bilgileri tabloya kaydeden kodları düzenlemeliyiz. Bunun için kayıt ekleme kodlarının **try** bloğunda değişiklik yapacağız:

Eski kod:

```

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

// 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 değerler
$urunadi=htmlspecialchars(strip_tags($_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 bağla
$stmt->bindParam(':urunadi', $urunadi);
$stmt->bindParam(':aciklama', $aciklama);
$stmt->bindParam(':fiyat', $fiyat);
$stmt->bindParam(':resim', $resim);
$stmt->bindParam(':kategori_id', $kategori_id);

// 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()){

```

Yeni kod:

```

$urun->urunadi=htmlspecialchars(strip_tags($_POST['urunadi']));
$urun->aciklama=htmlspecialchars(strip_tags($_POST['aciklama']));
$urun->fiyat=htmlspecialchars(strip_tags($_POST['fiyat']));
$resim!=empty($_FILES["resim"]["name"]) ? uniqid() . "-" .
basename($_FILES["resim"]["name"]) : "";
$urun->resim=htmlspecialchars(strip_tags($resim));
$urun->kategori_id=htmlspecialchars(strip_tags($_POST['kategori_id']));
// ürünün veritabanına kaydedildiği tarihi belirt
$urun->giris_tarihi=date('Y-m-d H:i:s');

// sorguyu çalıştır
if($urun->urun_ekle()){

```

ÜRÜN GÜNCELLEME (DÜZELTME) SAYFASINI DÜZENLEMEMEK

admin klasörü altındaki **urun** klasörü içinde yer alan **duzelt.php** isimli dosyayı açın ve ilk önce kategorileri açılır liste kutusunda listeleyen kodları aşağıdaki şekilde düzenleyin.

Eski kod:

```
<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>
```

Yeni kod:

```
<tr>
    <td>Kategori</td>
    <td>
        <?php
            $stmt = $kategori->kategorileri_oku();

            $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>
```

Ardından düzenlenecek kaydın bilgilerini tablodan okuyan kodları düzenlemeliyiz. Bunun için ilgili **try** bloğunda değişiklik yapacağız:

Eski kod:

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

    // 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'];
}

```

Yeni kod:

```

try {
    $urun->id = $id;
    $urun->urun_oku();

    // formu dolduracak değişken bilgileri
    $urunadi = $urun->urunadi;
    $aciklama = $urun->aciklama;
    $fiyat = $urun->fiyat;
    $kategori_id = $urun->kategori_id;
}

```

Son olarak bilgileri tabloya kaydeden kodları düzenlemeliyiz. Bunun için ilgili **try** bloğunda değişiklik yapacağız:

Eski kod:

```

try{
    // güncelleme sorgusu
    // çok fazla parametre olduğundan karışmaması için
    // soru işaretleri yerine etiketler kullanacağız
    $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);

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

```

```
    }  
}
```

Yeni kod:

```
try{  
    $urun->urunadi=htmlspecialchars(strip_tags($_POST['urunadi']));  
    $urun->aciklama=htmlspecialchars(strip_tags($_POST['aciklama']));  
    $urun->fiyat=htmlspecialchars(strip_tags($_POST['fiyat']));  
    $urun->kategori_id=htmlspecialchars(strip_tags($_POST['kategori_id']));  
    $urun->id = $id;  
  
    // sorguyu çalıştır  
    if($urun->urun_duzelt()){  
        echo "<div class='alert alert-success'>Kayıt güncellendi.</div>";  
    }else{  
        echo "<div class='alert alert-danger'>Kayıt güncellenemedi.</div>";  
    }  
}
```

EK GÖREVLER

- Bütün veritabanı işlemlerini sınıf yapısını kullanarak gerçekleştirin.