

# WEB PROGRAMLAMA II

Öğr. Gör. M. Mutlu YAPICI

Ankara Üniversitesi  
Elmadağ Meslek Yüksekokulu

# Ders İzlencesi

Hafta	Modüller/İçerik/Konular
1. Hafta	Oturum yönetimi
2. Hafta	Cookies kullanımı ve oturum yönetimi
3. Hafta	Session kullanımı ve oturum yönetimi
4. Hafta	Sayfalama ve Arama İşlemleri
5. Hafta	JavaScript, JQuery ve PHP
6. Hafta	AJAX ve PHP
7. Hafta	AJAX ve PHP
8. Hafta	ARA SINAV
9. Hafta	PHP'de Nesne Yönelimli Programlamaya Giriş(347)
10. Hafta	PHP de PDO
11. Hafta	
12. Hafta	
13. Hafta	
14. Hafta	

# Bu Ünite de Ele Alınan Konular

- Oturum yönetimi
- Cookies kullanımı ve oturum yönetimi
- Session kullanımı ve oturum yönetimi
- Sayfalama ve Arama İşlemleri

# Ders Kazanımları

Bu bölümü Bitirdiğimizde,

- **Oturum yönetimi**
- Cookies kullanımı ve oturum yönetimi
- Session kullanımı ve oturum yönetimi
- Sayfalama ve Arama İşlemleri

öğrenmiş olacaksınız.

# Sayfalama Ve Arama İşlemleri

Evet sayfalarımızı oluşturduk ancak tıkladığımızda verilerin değişmediğini görüyorsunuz. Şimdi işlevsellik kazandırmak için her sayfaya tıkladığımızda GET ile bir ofset numarası göndermeli ve ona göre SQL cümlesi düzenlenmelidir.

Sayfalama.php sayfamızın son halini tekrara yazalım değişiklik yaptığımız yerler kırmızı çerçeve içerisindedir.

# Sayfalama Ve Arama İşlemleri

## Sayfalama.php içeriği;

```
<?php
require_once("fonksiyonlar.php");
//öncelikle verilerimizi veritabanından çekelim
$vt=new mysqli( "localhost","root","usbw","emyopazar");
//eğer bağlantı başarısızsa hata verdirelim
if($vt->connect_error)
    die("Veritabanı Bağlantı Hatası. ".$vt->connect_error);

$offsetim=0;
if(!empty($_GET['offsetno']))
    $offsetim=$_GET['offsetno']+0;

//bağlantı başarılı ise
//$$SQL="SELECT * FROM `urunler`";
$$SQL="SELECT * FROM `urunler` LIMIT 2 OFFSET ?";
$vt->set_charset("UTF8");
$sorgu=$vt->prepare($$SQL);
$sorgu->bind_param('i',$offsetim);
$sorgu->execute();
```

# Sayfalama Ve Arama İşlemleri

Sayfalama.php içeriği (DEVAMI);

```
0 //eğer Sorgu başarısızsa hata verdirelim
1 if($vt->error)
2     die("SQL Sorgu Hatası. ".$vt->error);
3
4 $sonuc=$sorgu->get_result();
5
6 $veriDizi=Array();
7 $i=0;
8
9 while($veri=$sonuc->fetch_assoc())
10 {
11     $veriDizi[$i]=$veri;
12     $i++;
13 }
```

# Sayfalama Ve Arama İşlemleri

Sayfalama.php içeriği (DEVAMI);

```
$sorgu->close();
$vt->close();

///başlıkları oluşturalım ve Tablo yapma fonksiyonumuza gönderelim
$basliklar=Array("Ürün İd","Ürün Adı","Ürün Açıklaması","Stok","Fiyatı","Resim");

echo TabloYapBakalim($veriDizi, $basliklar);
///tablo altına sayfa numaraları oluşturalım
$listelenecekSatirSayisi=2;
for($i=1;$i<5;$i++)
{
    echo"<a href='sayfalama.php?ofsetno=".((($i-1)*$listelenecekSatirSayisi)." '>$i</a> <b>|</b> "
}
```

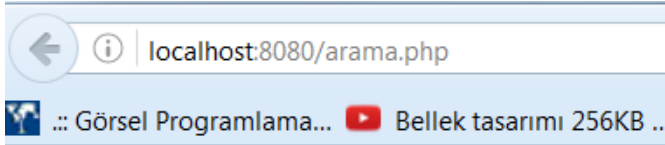


# Sayfalama Ve Arama İşlemleri

Sayfalama içeriğini yaptık şimdi sıra geldi arama yapmaya veritabanındaki veriler içerisinde arama yapmak için aslında şimdiye kadar yaptığımız listeleme işlemlerinden farklı bir işlem yapmayacağız sadece SQL sorgusu değişecek.

Aslında bu arama işlemini biraz önce yaptığımız sayfalama tablosunun üstüne bir arama kutusu ekleyerek yapmak istiyorum ama kodlar daha fazla karmaşık olmasın diye öncelikle ayrı bir arama sayfası yapacağım.

Öncelikle arayüzü aşağıdaki gibi tasarlayalım. Sayfamızın adı 'arama.php' olacak.



## Ürün Arama Sayfası

Ürünün Adı :	<input type="text"/>
<input type="button" value="Ürün Ara"/>	

# Sayfalama Ve Arama İşlemleri

'arama.php' arayüz kodları;

```
<h1 style="color:red;">Ürün Arama Sayfası</h1>
<form action="arama.php" method="POST">
  <table border="1" cellpadding="10px">
    <tr>
      <th> Ürünün Adı : </th><td><input type="text" value="" name="uadi"/></td>
    </tr>
    <tr>
      <td colspan="2" align="right"><input type="submit" value="Ürün Ara" name="btn"/></td>
    </tr>
  </table>
</form>
```

localhost:8080/arama.php

:: Görsel Programlama... Bellek tasarımı 256KB ...

## Ürün Arama Sayfası

Ürünün Adı :

Ürün Ara

# Sayfalama Ve Arama İşlemleri

'arama.php' sayfasında Ürün Ara butonuna tıkladığımızda veri tabanından ürünü bulup aynı sayfada tablo halinde listeleyelim. Öncelikle birebir arama yapalım yani ürün adını tam olarak arasın. Daha sonrada yazdığımız kelime herhangi bir ürünün adında veya bir parçasında geçiyorsa da listeleyebilelim.

Kodlarımızı çalıştırdığımızda eğer ürün varsa aşağıdaki gibi listemizi oluşturuyor.



localhost:8080/arama.php

.. Görsel Programlama... Bellek tasarımı 256KB ... Multiplexer example ... Implementation of ... U UDEMY KURS

## Ürün Arama Sayfası

Ürünün Adı :

Ürün İd	Ürün Adı	Ürün Açıklaması	Stok	Fiyatı	Resim
1	Defter	Be? ortal? harita metod kareli ders defteri MOPAK	158	16.50	

# Sayfalama Ve Arama İşlemleri

'arama.php' sayfasına ait kodlar; Bu kodları aynı sayfanın en altına ekliyoruz.

```
3 <?php
4 //eğer aranacak kelime geldiyse arayıp listeyi oluşturalım
5 if(!empty($_POST['uadi']))
6 {
7     require_once("fonksiyonlar.php");
8     $aranacakKelime=$_POST['uadi'];//aranacak kelimeyi aldık
9     $vt=new mysqli("localhost","root","usbw","emyopazar");
10
11     if($vt->connect_error)
12         die("Veritabanı Bağlantı Hatası. ".$vt->connect_error);
13
14     $sql="SELECT * FROM `urunler` WHERE `uadi`=?";
15     $sorgu=$vt->prepare($sql);
16     $sorgu->bind_param('s',$aranacakKelime);
17     $sorgu->execute();
```

# Sayfalama Ve Arama İşlemleri

'arama.php' sayfasına ait kodlar (DEVAMI);

```
if($sorgu->error)
    die("SQL Sorgu Hatası. ".$svt->error);

$sonuc=$sorgu->get_result();

$veriDizi=Array();
$i=0;
while($veri=$sonuc->fetch_assoc())
{
    $veriDizi[$i]=$veri;
    $i++;
}

$sorgu->close();
$svt->close();

///başlıkları oluşturalım ve Tablo yapma fonksiyonumuza gönderelim
$basliklar=Array("Ürün İd", "Ürün Adı", "Ürün Açıklaması", "Stok", "Fiyatı", "Resim");

echo TabloYapBakalim($veriDizi, $basliklar);
}
```

# Sayfalama Ve Arama İşlemleri

Arama sayfamızda hazır ve çalışıyor ancak fark ettiyseniz sadece aradığımız kelime bire bir tutuyorsa liste oluşuyor. D ile başlayan tüm verileri listele diyemiyoruz. Aynı şekilde 'a' ile bitenleri listele diyemiyoruz peki bunu nasıl yapacağız.

Aslında bu işlem oldukça basit sadece biraz önceki proje üzerinden SQL sorgumuzu değiştireceğiz. Şimdi bunu yapalım.

SQL cümlesi içerisinde LIKE anahtar kelimesi aranacak değerler benzer değerlerinin de bulunmasına olanak sağlar.

# Sayfalama Ve Arama İşlemleri

Aşağıda önceki sorgu ve değiştirilmiş hali görülmektedir.

```
$sql="SELECT * FROM `urunler` WHERE `uadi`=?";  
$sorgu=$vt->prepare($sql);
```

Son hali ;

```
$aranacakKelime=("%" . $aranacakKelime . "%");  
$sql="SELECT * FROM `urunler` WHERE `uadi` LIKE ?";  
$sorgu=$vt->prepare($sql);
```

Sonuç olarak SQL cümleciğini

```
$sql="SELECT * FROM `urunler` WHERE `uadi` LIKE '%$aranacakKelime%'";
```

Haline dönüştürmüş olduk. Böylece istediğimiz gibi esnek bir arama yapabiliyoruz.

# KAYNAKLAR

- İnternet ortamı
- PHP ve AJAX Haydar TUNA
- A'dan Z'ye PHP Rıza ÇELİK

