



- **Proje Başlığı:** Kullanıma Hazır E-Ticaret Sitesi
- **Konu:** PHP teknolojisini kullanarak kullanıma hazır websitesi
- **Projeyi Yapan Kişi:** Samet Samyeli - 1415010030
- **Öğrenci Numarası:** 1415010030
- **Öğrenim Türü:** 1. Öğretim
- **Proje Öğretmeni:** MUZAFFER TATLI
- **Kullanılacak Teknolojiler:** HTML, CSS, JAVASCRIPT, PHP, MYSQL
- **Projeye Başlama Tarihi:** 1 Ekim 2017 - Pazar

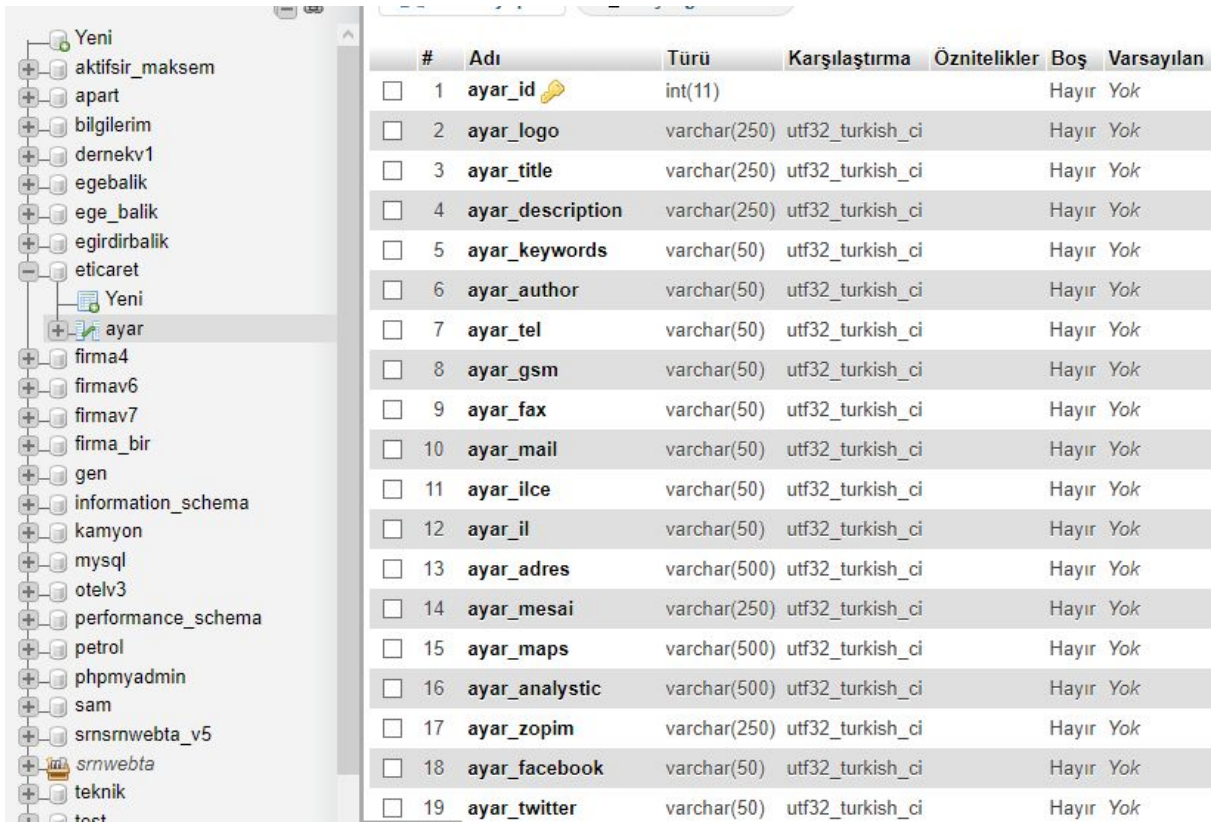
## > İlk Ayar Yapımı

E-Ticaret scripti yazmaya başlıyorum. Tamamen dinamik ve kullanıma hazır bir script olacak. İçinde **Web Programlama** hakkında bütün materyaller kullanılmış olacak.

İlk ayar hakkında bahsetmem gerekirse herşeyden önce veritabanı oluşturmamız gerekiyor. Ben Veritabanımı **PhpMyAdmin**'den **MySQL** teknolojisini kullanarak yaptım.

**eticaret** adını verdiğim Veritabanı sayesinde sitemizin veri bilgilerini burada tutabileceğiz. İlk oluşturduğum tablo ise **ayar tablosu**.

Bu tablo yazacak olduğumuz sitemizin yönetim panelinde olması gereken bilgileri barındırmaktadır.



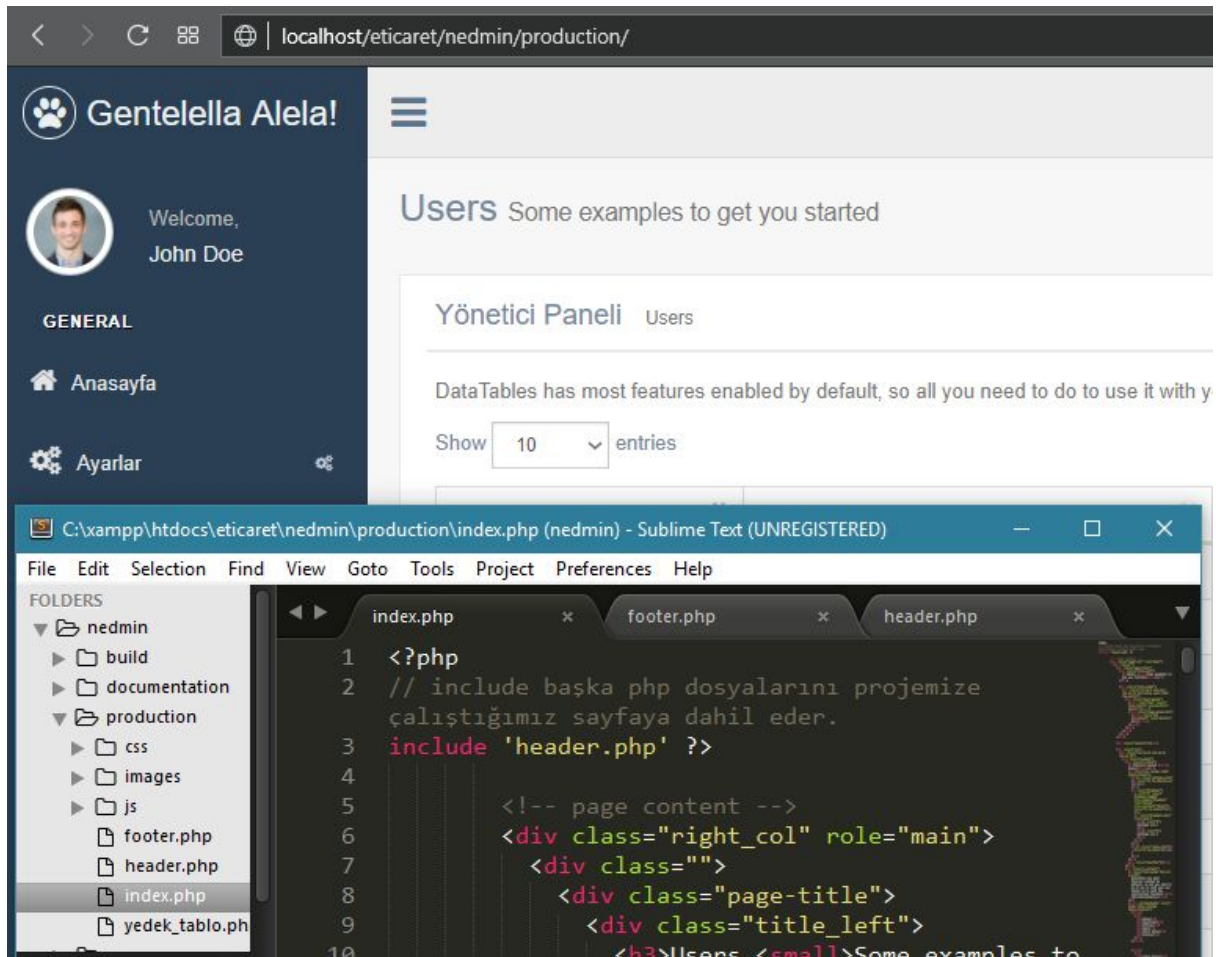
The screenshot shows the phpMyAdmin interface. On the left, the database 'eticaret' is selected, and the 'ayar' table is highlighted. On the right, the table structure is displayed as a table with columns: #, Adı, Türü, Karşılaştırma, Öznitelikler, Boş, and Varsayılan. The table contains 19 rows of field definitions.

#	Adı	Türü	Karşılaştırma	Öznitelikler	Boş	Varsayılan
1	ayar_id	int(11)			Hayır	Yok
2	ayar_logo	varchar(250)	utf32_turkish_ci		Hayır	Yok
3	ayar_title	varchar(250)	utf32_turkish_ci		Hayır	Yok
4	ayar_description	varchar(250)	utf32_turkish_ci		Hayır	Yok
5	ayar_keywords	varchar(50)	utf32_turkish_ci		Hayır	Yok
6	ayar_author	varchar(50)	utf32_turkish_ci		Hayır	Yok
7	ayar_tel	varchar(50)	utf32_turkish_ci		Hayır	Yok
8	ayar_gsm	varchar(50)	utf32_turkish_ci		Hayır	Yok
9	ayar_fax	varchar(50)	utf32_turkish_ci		Hayır	Yok
10	ayar_mail	varchar(50)	utf32_turkish_ci		Hayır	Yok
11	ayar_ilce	varchar(50)	utf32_turkish_ci		Hayır	Yok
12	ayar_il	varchar(50)	utf32_turkish_ci		Hayır	Yok
13	ayar_adres	varchar(500)	utf32_turkish_ci		Hayır	Yok
14	ayar_mesai	varchar(250)	utf32_turkish_ci		Hayır	Yok
15	ayar_maps	varchar(500)	utf32_turkish_ci		Hayır	Yok
16	ayar_analytic	varchar(500)	utf32_turkish_ci		Hayır	Yok
17	ayar_zopim	varchar(250)	utf32_turkish_ci		Hayır	Yok
18	ayar_facebook	varchar(50)	utf32_turkish_ci		Hayır	Yok
19	ayar_twitter	varchar(50)	utf32_turkish_ci		Hayır	Yok

## > Admin Template Parçalama İşlemleri

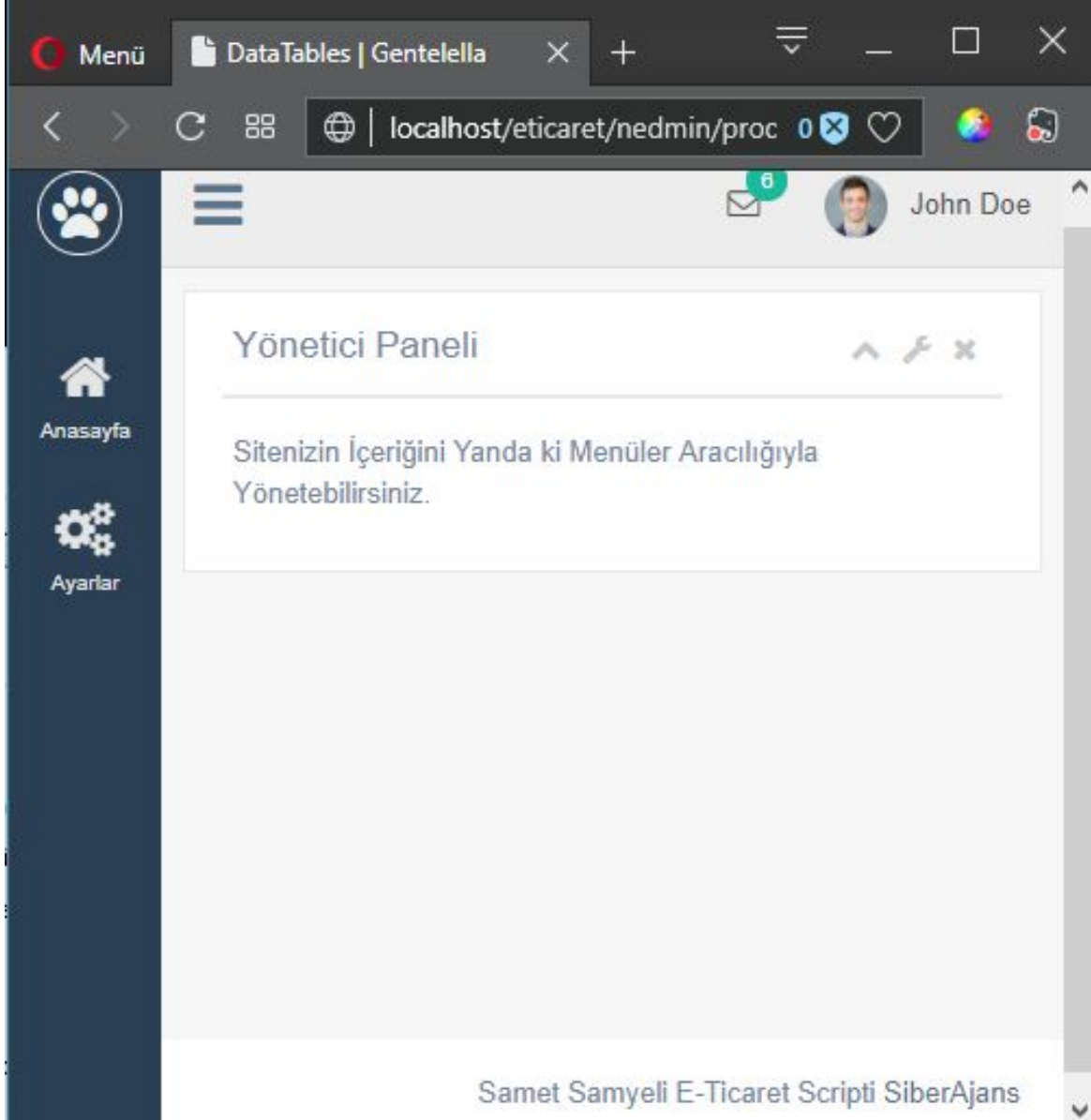
(Header, Footer, Index)

İnternet üzerinden bulduğumuz template'i dinamik olarak kullanabilmemiz için parçalama işlemini yapmak gerekmektedir. Sitenin arka yüzünden başlayarak **Gentelella Alela!** isimli yönetici temasını parçalama işlemine geçiyorum. Bunun için bir tane benim kopyaladığım, bir tane de sistemde yapacağım çalışma dosyasını bırakıyorum. **tables\_dynamic.html** sayfası kalacak şekilde diğer tüm dosyaları siliyorum ve ismini **index.php** olarak değiştiriyorum. Bu benim yönetim panelinde ilk sayfam olacak.



index.php dosyasını oluşturduktan sonra **Asp.Net**'te ki **MasterPage** mantığıyla **header** kısmını komple kopyalayıp **header.php** dosyasını oluşturup içine atıyorum ve index.php dosyamızın başına **<?php include 'header.php' ?>** yazıyoruz.

Bunun sebebi ise ne kadar deęişiklik yaparsak yapalım header kısmı aynı olacağı için farklı farklı deęişiklikler yapmamamızı sağlayacak. Aynı işlemi footer içinde yapıyorum ve içerisinde boş bir alan olmasını istediğim için <div>'leri dikkat ederek tüm sayfayı temizliyorum.



Bu bana yedeğim olduğu için daha sonradan yedeğimden istediğim tasarımı çekmek için rahat kullanmamı sağlayacak.

## > Genel Ayar Modül Sayfasının Oluşturulması

(Dosya oluşturma, Veritabanı Bağlantısı ve Form İşlemleri)

Genel Ayar Sayfasının oluşturulması için yedeğim de bulunan **form.html** sayfasını kullanıp temizleme yaptım. Veritabanının da yapmış olduğum “ana” bilgilerin düzenlemesini bu sayfa üzerinden yapacağım.

Temizleme işlemim bittikten sonra bilgilerin geri dönebilmesi için **netting** isminde bir dosya açıyor ve içine **baglan.php** ve **islem.php** dosyalarını yazıyorum. **baglan.php** sayesinde veritabanı bağlantısı yapabilecek **islem.php** sayesinde ise post ve get methodlarının geri dönebilmesini sağlayacağım.

```
<?php

try {

    $db=new PDO("mysql:host=localhost;dbname=eticaret;charset=utf8",'root','');
    //echo "veritabanı bağlantısı başarılı";
}

catch (PDOException $e) {

    echo $e->getMessage();
}

?>
```

Bunun için veritabanı bağlantısını yapmak zorundayım. **baglan.php**'ye girerek **PDO** teknolojisinin yardımıyla **\$db** isminde değişken oluşturup **try/catch**'e sokuyorum. Local'de çalıştığım için bilgilerimi de ona göre giriyorum.

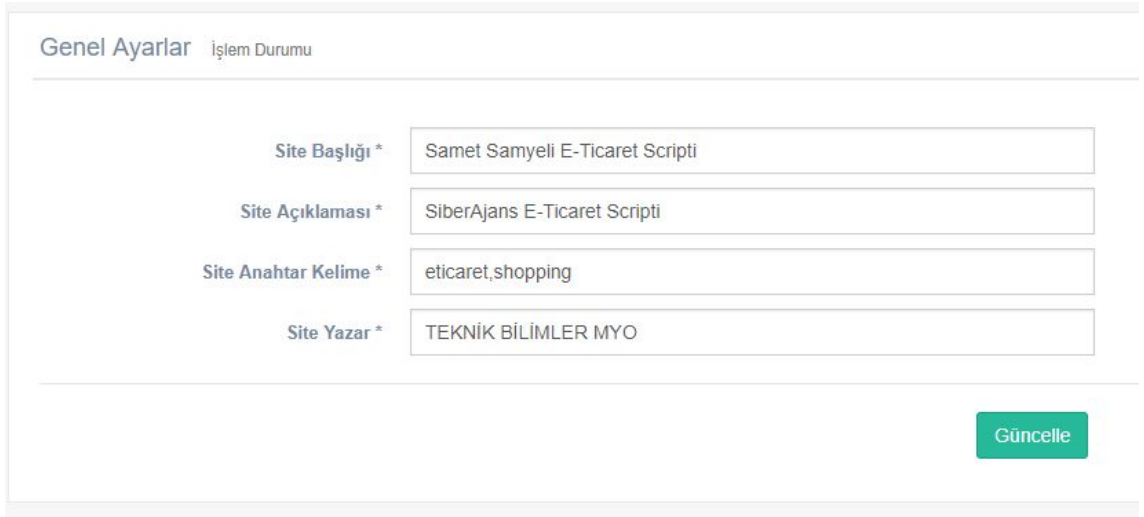
Ardından header.php'ye geri gelerek **baglan.php** dosyamızı **include** yapıyor ve verimizi **SQL** çekiyoruz. Başlangıçta “../” olmasının sebebi ise dosyamız bir üst dizinde olduğu için onu kullanıyoruz. Direkt **WHERE** sorgusundan sonra **ayar\_id** diye bırakmayıp devamını **=id** diye getirip Array sorgusunun içinde çekmemizin sebebi ise farklı girdiler verip saldırılara karşı güvenli olmak.

```
<?php

include '../netting/baglan.php';

//Belirli veriyi seçme işlemi
$ayarsor=$db->prepare("SELECT * FROM ayar WHERE ayar_id=:id");
$ayarsor->execute(array(
    'id'=>0
));
$ayarcek=$ayarsor->fetch(PDO::FETCH_ASSOC);
?>
```

Ne olup, ne bittiğini öğrenmemiz için tekrardan sitemize giriyoruz ve durumun ne olduğunu teyit ediyoruz.



Genel Ayarlar	İşlem Durumu
Site Başlığı *	Samet Samyeli E-Ticaret Scripti
Site Açıklaması *	SiberAjans E-Ticaret Scripti
Site Anahtar Kelime *	eticaret,shopping
Site Yazar *	TEKNİK BİLİMLER MYO

Güncelle

Verilerimizi yazdığımızımıza göre artık “Güncelle” butonuna aktiflik kazandırıp veritabanı işlemlerimizi kayıt altına tutmakta. Bunun için **islem.php** dosyamıza gidiyoruz ve kodlarımızı yazmaya başlıyoruz.



**islem.php**'de **isset** fonksiyonunu açıyoruz ve içine formumuzun **Güncelle** butonunun ismini yazıyoruz. Bu sayede direkt işlem **isset**'in içine girecek ve **if sorgusu** okumaya başlayacak. Öncelikle veritabanına bağlantı yapıyoruz ve ardından **UPDATE** işlemini gerçekleştiriyoruz.

```
if (isset($_POST['genelayarkaydet'])) {  
    //Tablo İşlem Kodları  
  
    $ayarkaydet=$db->prepare("UPDATE ayar SET  
        ayar_title=:ayar_title,  
        ayar_description=:ayar_description,  
        ayar_keywords=:ayar_keywords,  
        ayar_author=:ayar_author  
        WHERE ayar_id=0");  
  
    $update=$ayarkaydet->execute(array(  
        'ayar_title' => $_POST['ayar_title'],  
        'ayar_description' => $_POST['ayar_description'],  
        'ayar_keywords' => $_POST['ayar_keywords'],  
        'ayar_author' => $_POST['ayar_author']  
    ));  
}
```

İşlemimiz normalde bitti fakat ben sayfanın tekrardan işlem sayfasına geri dönmesini ve yazı yazdırmak istiyorum bunun içinde yeniden **genel-ayar.php** sayfasına gidiyor ve aşağıda ki sorguyu yazıyorum. PDO'nun güzel kısmı burada başlıyor. Çünkü tablo isimlerini farklı girdiğimiz için herhangi bir XSS, SQL Inj. saldırısında farklı bir yere göndermesini sağlayacak.

```
<h2>Genel Ayarlar <small>,<br>  
    <?php  
    if ($_GET['durum']=="ok") {?>  
        <b style="color:green;">İşlem Başarılı...</b>  
    <?php } elseif ($_GET['durum']=="no") {?>  
        <b style="color:red;">İşlem Başarısız...</b>  
    <?php }  
    ?>  
</small></h2>
```

ve **islem.php**'de geri döndürme kodlarımızı yazıyoruz.

```
if ($update) {  
    header("Location:../production/genel-ayar.php?durum=ok");  
}else{  
    header("Location:../production/genel-ayar.php?durum=no");  
}
```

## > İletişim, Sosyal, API ve Mail Modülü Sayfaları (Form Tasarımı ve İşlem.php Ayarları)

```
if (isset($_POST['iletisimayarkaydet'])) {  
    //Tablo İşlem Kodları  
    $ayarkaydet=$db->prepare("UPDATE ayar SET  
        ayar_tel=:ayar_tel,  
        ayar_gsm=:ayar_gsm,  
        ayar_fax=:ayar_fax,  
        ayar_mail=:ayar_mail,  
        ayar_ilce=:ayar_ilce,  
        ayar_il=:ayar_il,  
        ayar_adres=:ayar_adres,  
        ayar_mesai=:ayar_mesai  
        WHERE ayar_id=0");
```

Genel ayarlar sayfamız ile aynı mantıkta olacağı için **genel-ayar.php** dosyasını kopyalayıp ismini **iletisim-ayarlar.php** ve **api-ayarlar** yapıyorum. Bir önce ki işlemin aynı kopyalarını çıkartıyorum ve fazladan form ekliyorum. Sadece veritabanı işlem isimleri, form action ve method isimlerini değiştiriyorum ve sayfamız hazır oluyor.

*Sosyal Ağlar ve Mail Ayarlar sayfalarını da bu mantıkla yapıyoruz.*

localhost/eticaret/nedmin/production/iletisim-ayarlar.php

Hoşgeldiniz,  
Samet Samyeli

GENERAL

Anasayfa

Site Ayarlar

Genel Ayarlar

İletişim Ayarları

API Ayarları

İletişim Ayarlar

Telefon Numarası\* 3114848

Telefon Numarası (GSM)\* 5051563083

FAX Numarası\* 5051563083

Mail Adresi\* ruzgarsamet@gmail.com

İl\* Antalya

İlçe\* Konyaaltı

Adres\* Liman Mahallesi

Mesai\* 7/24 Açık



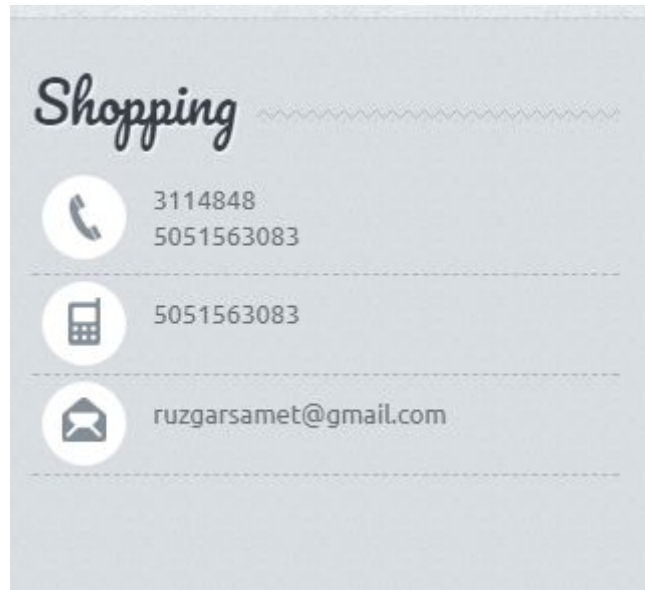
## > Ön Template Parçalama İşlemleri

(Header, Footer, index.php)

Yönetim panelinde yeteri kadar bilgi girdikten sonra ön template parçalama işlemine giriyorum. Bu sayede ne değişiklik yaptığımı, düzenlemelerimi canlı olarak görmemi sağlayacak. Hemen daha önceden de yapmış olduğum gibi dosyalarımın birer yedeğini alarak **index.html** sayfasını çağırıyorum ve **header.php** , **footer.php** ve **index.php** olarak üçe bölüyorum. Daha sonrasında header'a gelip veritabanı bağlantı ayarımı yapıyorum. Sadece header'a bağlantı ayarı yapmamın sebebi ise sırayla hepsini okuyacak (header/index/footer) ve **header.php** en tepede olduğu için diğerlerine de bağlantı imkanı tanıyacak. Daha sonra **include** yapıyorum ve sayfam tek sayfa olarak görünmekte.

```
</div><!-- footer newsletter widget-->
<div class="col-md-4"><!-- footer contact widget-->
  <div class="title-widget-bg">
    <div class="title-widget-cursive">Shopping</div>
  </div>
  <ul class="contact-widget">
    <li class="fphone"><?php echo $ayarcek['ayar_tel'] ?><br><?php echo $ayarcek['ayar_fax'] ?></li>
    <li class="fmobile"><?php echo $ayarcek['ayar_gsm'] ?></li>
    <li class="fmail lastone"><?php echo $ayarcek['ayar_mail'] ?></li>
  </ul>
</div><!-- footer contact widget-->
```

Başlık, sosyal ayarlar, telefon numaraları gibi ayrıntıları da çekerek istediğim görüntünün yavaş yavaş ortaya çıkmasını sağlıyorum.



## > Hakkımızda Modülü

(MySQL, Form Tasarımı ve Islem.php Ayarları)

Hakkımızda(About Us) modülünü yapmak istiyorum bunun için veritabanı oluşturun. Aynı ayar tablosundan devam edebilirdim fakat bu iyi bir yöntem olmayacaktır. Sırasıyla aşağıda ki tabloları oluşturdum.

The screenshot shows the phpMyAdmin interface for a database named 'eticaret'. The table 'hakkimizda' is selected, and its structure is displayed. The table has the following columns:

Adı	Türü	Uzunluk/Değerler
hakkimizda_id	INT	11
hakkimizda_baslik	VARCHAR	250
hakkimizda_icerik	TEXT	
hakkimizda_vizyon	VARCHAR	500
hakkimizda_misyon	VARCHAR	500
hakkimizda_video	VARCHAR	50

Bu sayfamızda Başlık, İçerik, Vizyon, Misyon ve Video konulmasını istiyorum. İleri de daha farklı düşüncelerimiz olduğunda değiştirmek bize kalmış. **hakkimizda\_icerik**'in **TEXT** girmemizin sebebi, kullanıcının istediği yazı tutarı kadar girilmesini sağlamak içindir.

İşlemlerin yapıldığında boş dönmemesi için doluluk olması açısından birkaç yazı yazıyorum ve kayıt ediyorum.

Sütun	Türü	İşlev	Boş Değer
hakkimizda_id	int(11)		0
hakkimizda_baslik	varchar(250)		SDÜ TBMYO SAMET SAMİYELİ
hakkimizda_icerik	text		Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. Lorem Ipsum is simply dummy text of the printing and typesetting industry.
hakkimizda_vizyon	varchar(500)		SDÜ TBMYO ile ilgili hakkımızda vizyon içeriği alacağından buraya vizyon içeriği girilecektir.
hakkimizda_misyon	varchar(500)		SDÜ TBMYO ile ilgili hakkımızda misyon içeriği alacağından buraya misyon içeriği girilecektir.
hakkimizda_video	varchar(50)		video_kodu

Git

Bunu oluşturduğumuza göre yönetim panelinden içerik değiştirme ve eklemeleri yapmak için yönetim panelinde değişiklikler yapıyorum. **genel\_ayar.php**'ye giderek **hakkımızda** bölümünü güncellemek için kullanacağım.

Genel Ayar Modülünü yaptığımız mantıkla aynı kodların sadece isimlerini değiştirerek işleme sokuyorum. Aşağıda **hakkimizda.php**'nin en üstünde veritabanı bağlantısı bulunuyor. Daha önceden \$ayarsor gibi değişken isimlerinin hepsini kafa karışıklılığı olmaması için **\$hakkimizdasor** olarak değiştiriyorum.

```
<?php

include 'header.php';

//Belirli veriyi seçme işlemi
$hakkimizdasor=$db->prepare("SELECT * FROM hakkimizda where hakkimizda_id=:id");
$hakkimizdasor->execute(array(
    'id' => 0
));
$hakkimizdacek=$hakkimizdasor->fetch(PDO::FETCH_ASSOC);

?>
```

En sonunda da **islem.php**'ye giderek işlem yaptırmamızı sağlayacak kodlara bakıyor ve gerekli değişiklikleri yapıyorum.

```
if (isset($_POST['hakkimizdakaydet'])) {

    //Tablo İşlem Kodları

    $hakkimizdakaydet=$db->prepare("UPDATE ayar SET

        hakkimizda_baslik=:hakkimizda_baslik,
        hakkimizda_icerik=:hakkimizda_icerik,
        hakkimizda_video=:hakkimizda_video,
        hakkimizda_misyon=:hakkimizda_misyon
        WHERE ayar_id=0");

    $update=$hakkimizdakaydet->execute(array(
        'hakkimizda_baslik' => $_POST['hakkimizda_baslik'],
        'hakkimizda_icerik' => $_POST['hakkimizda_icerik'],
        'hakkimizda_vizyon' => $_POST['hakkimizda_vizyon'],
        'hakkimizda_misyon' => $_POST['hakkimizda_misyon']
    ));

    if ($update) {
        header("Location:../production/hakkimizda.php?durum=ok");
    }else{
        header("Location:../production/hakkimizda.php?durum=no");
    }

}
```

İşlemlerin ne durumda olduğunu görmek için yönetim paneline giriyorum.

Hoşgeldiniz,  
Samet Samyeli

GENERAL

Anasayfa

Hakkımızda

Site Ayarlar

### Hakkımızda Ayarları

Hakkımızda Başlığı \*

SDÜ TBMYO SAMET SAMYELİ

Hakkımızda İçerik \*

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lc

Hakkımızda Video \*

video\_kodu

Vizyon \*

SDÜ TBMYO ile ilgili hakkımızda vizyon içeriği alacağından buraya vizyon iç

Misyon \*

SDÜ TBMYO ile ilgili hakkımızda misyon içeriği alacağından buraya misyon iç

Güncelle

Verilerim gelmiş durumda. Fakat **Hakkımızda İçerik** kısmı **textbox** olduğu için bir düzenleme yapmam gerekmekte. Bunun içinde **CK Editor** kurmam gerekiyor. Hemen kuruluma başlıyorum.

```
<!--CK Editor-->
<script src="https://cdn.ckeditor.com/4.7.3/standard/ckeditor.js"></script>
```

Öncelikle **CK Editor**'un kendi sayfasına girerek **javascript** kodunu alıyorum ve **header.php**'de yani sitemizin en tepesinde **<head>** kod blogumun içerisine yapııştırıyorum. Ardından düzenleme sayfamıza geliyorum ve formumuzun **input** kısmını komple değiştirip **textarea** yapıyor ve **div**'in bittiği yere script kodlarını yapııştırıyorum.

```
<!-- Ck Editor Başlangıç -->

<div class="form-group">
  <label class="control-label col-md-3 col-sm-3 col-xs-12" for="first-name">İçerik <span class="required">*</span>
</label>
<div class="col-md-6 col-sm-6 col-xs-12">
  <textarea class="ckeditor" id="editor1" name="hakkimizda_icerik"><?php echo $hakkimizdacek['hakkimizda_icerik']; ?></textarea>
</div>
</div>

<script type="text/javascript">
  CKEDITOR.replace( 'editor1',
  {
    filebrowserBrowseUrl : 'ckfinder/ckfinder.html',
    filebrowserImageBrowseUrl : 'ckfinder/ckfinder.html?type=Images',
    filebrowserFlashBrowseUrl : 'ckfinder/ckfinder.html?type=Flash',
    filebrowserUploadUrl : 'ckfinder/core/connector/php/connector.php?command=QuickUpload&type=Files',
    filebrowserImageUploadUrl : 'ckfinder/core/connector/php/connector.php?command=QuickUpload&type=Images',
    filebrowserFlashUploadUrl : 'ckfinder/core/connector/php/connector.php?command=QuickUpload&type=Flash',
    forcePasteAsPlainText: true
  }
  )
</script>
```



**CK Editor Kurulumu** da bu şekilde sayfamı yeniledikten sonra değişikliği görüyoruz. Artık içerik düzenlemelerinde yazı fontundan, kalınlığına, resim ekleme gibi işlemleri basitce yapabileceğiz.

[illegible]

Güncelle

Yönetim paneli ayarları bu kadar. Şimdi ön tarafa gidip sayfamızı parçalayıp, bilgileri çekelim.

Parçalama işlemi daha önceden de yaptığımız gibi **header/index** ve **footer.php** olarak üçe ayrılıyor. İlk başta header ve footer oluşturduğumuz içinde bu sayfada bunları siliyoruz. Sayfanın başına ve sonuna **include** yapıyoruz.



Sayfamız bir bütünlük kazandıktan sonra komutlarımızı sırasıyla yazmaya başlıyoruz. **title** kısmına **Hakkımızda Sayfası** yazıyorum ve aşağıya inerek sırasıyla *hakkimizda\_baslik*, *hakkimizda\_video*, *hakkimizda\_icerik*, *hakkimizda\_misyon* ve *hakkimizda\_vizyon* kısımlarını yazdırma komutuyla çekiyorum.

Video kısmı gözüme takılıyor ve sayfamın boyutunu bozmadan **703x343** ayarında kalacak şekilde sabitliyor ve **Youtube** embed özelliğini kullanmadan sadece kod kısmını alabilsin diye linkin sonundan itibaren php kodlarımı yazıyorum. Artık bu sayede uzun uzun embed kodları yazmamıza gerek kalmayacak.

```
<iframe width="703" height="343" src="https://www.youtube.com/embed/<?php echo $hakkimizdacek['hakkimizda_video']; ?>" frameborder="0" allowfullscreen></iframe>
```

Sitemize genel bir bakış atıyorum ve istediğimiz gibi ayarlarımızın ekrana yazdırıldığını görüyoruz.

### Hakkımızda Sayfası

Hakkımızda Başlığı Burada Yazar



SDÜ Tanıtım Filmi 2017 "Hepimizin Hikayesi" (4K)

Kurulma aşamasında adı Cöller Bölgesi Üniversitesi olarak düşünülmüş daha sonra İspartalı siyasetçi ve devlet adamı Süleyman Demirel'in adı verilmiştir. Süleyman Demirel'in cumhurbaşkanlığı döneminde tanınırlığını artıran Üniversitenin Doğu Kampüsünde yer alan Tıp Fakültesi'ni İş adamı Dr. Özeyir Canlı yapmıştır. Kampüs, İstanbul yolu üzerindeki Çiğirde yolun iki tarafında Doğu Kampüsü ve Batı Kampüsü olarak ikiye ayrılmış durumdadır. Üniversitede 19 Fakülte, 3 Yüksekokul, 20 meslek yüksek okulu, 6 enstitü, 40 araştırma ve uygulama merkezi bulunmaktadır.

#### Misyonumuz

SDÜ; toplumla etkileşim içinde, insanı öncelleyerek, uluslararası standartlarda eğitim vermeyi, araştırma yapmayı, hizmet sunmayı ve çözümler üretmeyi görev edinmiştir.

#### Vizyonumuz

SDÜ; evrensel ölçekte bilgi, teknoloji ve hizmet üreterek, ilham ve yön veren bir Üniversite olmayı hedeflemektedir.

#### Categories

- Women Accessories
- Men Shoes
- Gift Specials
- Electronics
  - iPhone 4S
  - Samsung Galaxy
  - MacBook Pro 17"
- On sale
- Summer Specials
- Electronics
- Unique Stuff

#### Best Seller

-  Panasonic MB Price : \$122
-  Panasonic MB Price : \$122
-  Panasonic MB Price : \$122

## > .Htaccess Dosyası Oluşturma İşlemler

(404, SEO, Author)

Sitemizin genel ayarlarına girmeden önce **.htaccess** dosyası oluşturmayı düşündüm. Bu ileride bize çok kolaylık sağlayacak. **.htaccess** sayesinde SEO linkleri oluşturabilir, 404 hata sayfalarını sunabiliriz.

Böyle ayarları normalde kod ortamında uzun uzun yapabiliriz fakat biz kolaylık istediğimizden dolayı dosyalarımızın olduğu yere hiçbir uzantı eklemekten **.htaccess** dosyasını oluşturuyoruz ve aşağıda ki kod bloklarını yapıştırıyoruz.

```
RewriteEngine on

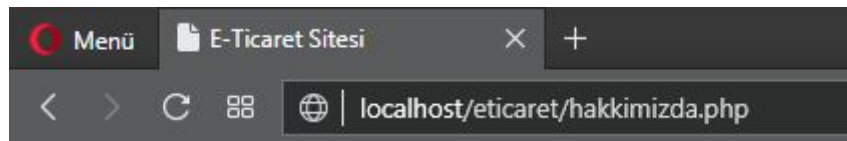
#RewriteCond %{SERVER_PORT} 80
#RewriteRule ^(.*)$ http://www.site_adi.com/$1 [R,L]

#RewriteCond %{HTTP_HOST} ^site_adi.com. [NC]
#RewriteRule ^(.*)$ http://www.site_adi.com/$1 [L,R=301,NC]

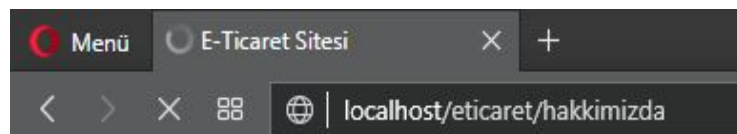
#Seo url oluşturma
RewriteCond %{REQUEST_FILENAME} !-d
RewriteCond %{REQUEST_FILENAME}\.php -f
RewriteRule ^(.*)$ $1.php

#404 hata sayfası yönlendirme kodu
#ErrorDocument 404 /404.html
```

Örnek vermek gerekirse artık sitemizin url kısmında **.php** olarak görmeyeceğiz ve bu bize **SEO** konusunda yardımcı olacaktır.



Bu kullanım yerine aşağıda ki kullanım yer alacaktır.



## > Yönetici Giriş İşlemleri

(Login Sayfası, Kullanıcı Tabloları, MD5 Şifreleme ve \$\_SESSION Kullanımı)

Sıra yönetici giriş işlemlerine geldi. Artık yönetici panelimize girerken kullanıcı adı ve şifre yazmamız gerekli. Bunun için kullanıcı bilgilerinin kayıt olacağı bir tane veritabanı oluşturdum ve ismini **kullanici** olarak verdim.

#	Adı	Türü	Karşılaştırma	Öznitelikler	Boş	Varsayılan
<input type="checkbox"/>	1	kullanici_id	int(11)		Hayır	Yok
<input type="checkbox"/>	2	kullanici_zaman	datetime		Hayır	CURRENT_TIMESTAMP
<input type="checkbox"/>	3	kullanici_resim	varchar(250)	utf8_turkish_ci	Hayır	Yok
<input type="checkbox"/>	4	kullanici_tc	varchar(50)	utf8_turkish_ci	Hayır	Yok
<input type="checkbox"/>	5	kullanici_ad	varchar(50)	utf8_turkish_ci	Hayır	Yok
<input type="checkbox"/>	6	kullanici_mail	varchar(100)	utf8_turkish_ci	Hayır	Yok
<input type="checkbox"/>	7	kullanici_gsm	varchar(50)	utf8_turkish_ci	Hayır	Yok
<input type="checkbox"/>	8	kullanici_password	varchar(50)	utf8_turkish_ci	Hayır	Yok
<input type="checkbox"/>	9	kullanici_adsoyad	varchar(50)	utf8_turkish_ci	Hayır	Yok
<input type="checkbox"/>	10	kullanici_adres	varchar(250)	utf8_turkish_ci	Hayır	Yok
<input type="checkbox"/>	11	kullanici_il	varchar(100)	utf8_turkish_ci	Hayır	Yok
<input type="checkbox"/>	12	kullanici_ilce	varchar(100)	utf8_turkish_ci	Hayır	Yok

Veritabanım hazır olduğuna göre artık bunu kullanmak kaldı.

Yönetim panelinin giriş ekranını sevmediğim için basitce Google'a **login template** yazıp arama yaptım ve aşağıda ki linkten hazırlanmış template buldum ve hemen koda dökmeye başladım.

**link:** <https://colorlib.com/wp/html5-and-css3-login-forms/>

Template'i Türkçeleştirdikten sonra **form input** kısımlarında ki **name** alanlarını veritabanında koyduğumuz isimleri verdim. Form action kısmını ise her zaman ki gibi **islem.php**'ye **\$\_POST** yaptım.

Daha sonra **islem.php**'ye gelerek aşağıda ki kodları yazdım.

```
if (isset($_POST['admingiris'])) {  
    //Kullanıcı Giriş İşlemleri  
  
    $kullanici_mail=$_POST['kullanici_mail'];  
    $kullanici_password=md5($_POST['kullanici_password']);  
  
    $kullanicisor=$db->prepare("SELECT * FROM kullanici WHERE kullanici_mail=:mail and  
        kullanici_password=:sifre and kullanici_yetki=:yetki");  
    $kullanicisor->execute(array(  
        'mail'=>$kullanici_mail,  
        'sifre'=>$kullanici_password,  
        'yetki'=>5  
    ));  
    $say = $kullanicisor->rowCount();  
    if ($say==1) {  
        $_SESSION['kullanici_mail']=$kullanici_mail;  
        $_SESSION['kullanici_password']=$kullanici_password;  
        header("Location:../production/index.php");  
    }else{  
        header("Location:../production/login.php?durum=no");  
    }  
}
```

Diğer kodlardan farklı olarak **\$kullanici\_password=md5()**; olarak başladım. **MD5** yazmamın sebebi veritabanına şifrelenmiş bir şekilde göndermesidir. **\$\_SESSION**'lar ise de bilgileri site içinde tutmasını sağlamaktadır. **Yetki'nin 5** olması da yönetici olduğu anlamına gelmektedir.

Bu şekilde yazılan kodda her hangi bir sıkıntı yok gibi gözüküyor. Yönetici paneline kullanıcı adı ve şifreyle girebiliyoruz fakat link'i kopyalayıp da girebiliyoruz yani dışardan birisi bizim panelimize istediği gibi girebilir.

Bunu önlemek içinde **header.php**'ye geliyoruz ve **islem.php**'de kullandığımız **rowCount()**'u kullanıyoruz. Bu fonksiyon işlemin 1 veya 0 olduğunu tutuyor.

```
//Kullanici
$kullaniciisor=$db->prepare("SELECT * FROM kullanici WHERE kullanici_mail=:mail");
$kullaniciisor->execute(array(
    'mail'=>$_SESSION['kullanici_mail']
));
$say=$kullaniciisor->rowCount();
$kullanicicek=$kullaniciisor->fetch(PDO::FETCH_ASSOC);

if ($say==0) {
    header("Location:login.php?durum=izinsiz");
    exit;
}
```

**\$say** diye bir değişken açıp **\$kullaniciisor**'u **rowCount()**'un içine aldım. daha sonra **if koşulunu kullanarak** sonucun 1 veya 0 olduğunu anlamasına sağlayarak işlem tutmasını söyledim. Eğer SESSION bilgilerimiz tam değil, giriş yapılmamışsa işlem sıfır oluyor ve direkt giriş sayfasına yönlendiriliyor. Eğer işlemimizde herhangi bir problem yoksa **yönetici paneline girmiş oluyoruz**.

## > Kullanıcı İşlemleri

(Kayıtlı Kullanıcıları Listeleme, Düzenleme ve Silme İşlemleri)

Kullanıcı işlemlerine giriş yapıyorum. Burada öncelikle yönetim panelimizde ki veritabanımıza kayıt ettiğimiz kullanıcıları listelemek ile başlayacağım.

Yönetim panelimizde daha önceden oluşturduğumuz **hakkımızda** sayfasının bir kopyasını alıyor ve içeriğini siliyorum. Daha sonra **tablo** yardımı ile görüntülenmesini istediğim veritabanı isimlerine bakıyor ve listeliyorum.

kullanici_zaman	datetime	<input type="text"/>	2017-07-08 15:21:45
kullanici_resim	varchar(250)	<input type="text"/>	
kullanici_tc	varchar(50)	<input type="text"/>	
kullanici_ad	varchar(50)	<input type="text"/>	
kullanici_mail	varchar(100)	<input type="text"/>	ruzgarsamet@gmail.com
kullanici_gsm	varchar(50)	<input type="text"/>	08508408076
kullanici_password	varchar(50)	<input type="text"/>	2583197bf317d53b18c582ba2826d6ed
kullanici_adsoyad	varchar(50)	<input type="text"/>	Samet Samyeli
kullanici_adres	varchar(250)	<input type="text"/>	
kullanici_il	varchar(100)	<input type="text"/>	Isparta
kullanici_ilce	varchar(100)	<input type="text"/>	Eğirdir

Burada benim amacım ilk sırada **kullanici\_zaman**, **kullanici\_adsoyad**, **kullanici\_mail** ve **kullanici\_mail** işlemlerinin ekranda görüntülenmesini sağlamak. Sayfaman en yukarisına istediğim sorguların çekilmesi için gerekli olan bağlantı kodlarını yazıyorum.



```
include 'header.php';

//Belirli veriyi seçme işlemi
$kullaniciisor=$db->prepare("SELECT * FROM kullanıcı ");
$kullaniciisor->execute(array());

?>
```

**SELECT\*FROM** diyip tüm kullanıcıların ve işlemlerin listelenmesini istiyorum.

```
</thead>

<?php

while ($kullanicicek=$kullaniciisor->fetch(PDO::FETCH_ASSOC)) {?>

<tr>
<td><?php echo $kullanicicek['kullanici_zaman']; ?></td>
<td><?php echo $kullanicicek['kullanici_adsoyad']; ?></td>
<td><?php echo $kullanicicek['kullanici_mail']; ?></td>
<td><?php echo $kullanicicek['kullanici_gsm']; ?></td>
```

Daha sonra ise aşağıya inerek tablo kodlarımı içine alarak bir tane **while döngüsü** oluştuyorum. Sebebi ise **while döngüsü işlem doğru olduğu sürece çalışmasıdır yani yeni işlem geldikçe aşağıya doğru uzayacak bir liste yapıyorum.**

Kullanıcı Ayarları					
Show	10	entries	Search:		
Kayıt Tarihi	Ad Soyad	Mail Adresi	Telefon		
2017-07-08 15:21:45	Samet Samyeli	ruzgarsamet@gmail.com	08508408076	Düzenle	Sil
Showing 1 to 1 of 1 entries					
Previous 1 Next					

Tabloların içine de **Düzenle** ve **Sil** butonlarımı koyuyorum. Bunları koymak için de **CSS Framework'u Bootstrap** kullanıyorum ve daha sonradan sayfama geldi/gelmedi diye kontrol ediyorum. İstediğim gibi yazdırıldığını görüyorum. Daha sonra **genel-ayar.php**'ye gidip sayfama link veriyorum.

Listeleme işlemlerini **while döngüsü** ile basitçe hallettik. Şuanda ise **Kullanıcı Düzenleme İşlemlerine** giriş yapıyorum.

Kullanıcı düzenleme işlemlerinde yapacağım olay gerçekten benim için şuanda çok önemli. Sebebi ise daha önceden yazdığım kodları referans olarak gittiğimden dolayı, bu düzenleme işlemini daha sonra **Menü İşlemleri, Ürün İşlemleri, Kategori İşlemleri, Sepet İşlemleri, Sipariş ve Slider Modülünde** aynı mantıkta kullanmaya devam edeceğim.

Az önce ki işlemlerimiz çalışmaya devam ediyor. Bunu bir sonra ki sayfaya aktarmak istiyorum bunun içinde en mantıklı sayfa olarak **Genel Ayar** sayfasını düzenliyorum ve sadece 4 tane alan bırakıyorum. Bu sayfa da **TC ve Telefon Numarası, Mail Adresi ve Kullanıcı Durumu(Aktif/Pasif)** yazılarını giriyorum.

```
<td><center><a href="kullanici-duzenle.php?kullanici_id=<?php echo $kullanicicek['kullanici_id']; ?>"><button class="btn btn-primary btn-xs">Düzenle</button></a></center></td>
```

İşlemlerimiz veritabanında kayıt edilsin diye sayfanın en başına veri seçme işlemini giriyoruz. **kullanici\_id**'den düzenleme işlemini yapacağımız için sorgumuza **WHERE** kullanici\_id=:id olarak değişiklik yapıyoruz. İşlemlerimizi **\$\_GET** ile alacağım. Bunun içinde bir önce ki sayfamızda olan Düzenle butonuna gerekli kodları yazıyorum ve bu sayfamıza yönlendiriyorum. Bu sayede kullanıcımızın **id'si** kaybolmayacak ve istediğimiz işlemi yapmamıza olanak sağlayacak.

```
<div class="form-group">
  <label class="control-label col-md-3 col-sm-3 col-xs-12" for="first-name">Kullanıcı Durumu <span class="required">*</span>
</label>
<div class="col-md-6 col-sm-6 col-xs-12">
  <select id="heard" class="form-control" name="kullanici_durum" required">
    <option value="1" <?php echo $kullanicicek['kullanici_durum'] == '1' ? 'selected=""' : '';>
      >Aktif</option>
    <option value="0" <?php echo $kullanicicek['kullanici_durum'] == '0' ? 'selected=""' : '';>
      >Pasif</option>
  </select>
</div>
</div>
```

Aktif/Pasif ayarlarını yapmam için **if döngüsünü** “**kısaltılmış**” halde kullanıyorum. Bu **PHP**'de **son sürümlerinde** gelen bir özellik. Bunun içinde kopyalamış olduğum textbox'u komple silip **<select><option>** tagları arasına kodlarımı yazıyorum.

Kod blogunun açıklaması olarak diyor ki, eğer veritabanından gelen değer 1 ise **aktif** 0 ise **pasif** yazdırmasını söylüyorum.

Kullanıcının giriş yapmasını sağlayan E-Posta adresinin de değiştirilemeyeceğini düşünüp **disable** özelliği veriyorum. Bu sayede yönetim panelinden E-Posta ayarlarını görüp, üzerinde değişiklik yapmayacağız.

---

TC Numarası *	<input type="text" value="123456798"/>
Telefon Numarası *	<input type="text" value="08508408076"/>
Mail Adresi *	<input type="text" value="ruzgarsamet@gmail.com"/>
Kullanıcı Durumu *	<input type="text" value="Aktif"/>

Güncelle

---

Artık kullanıcı bilgilerimi gösterebiliyorum. Fakat güncelle butonuna bastığımda işlemler kayboluyor ve farklı bir sayfaya gidiyor. Bunun için öncelikle **kullanici-duzenle.php** sayfama gidip formlarımızın olduğu yerine bir tane **input** açıyoruz. İsmi **kullanici\_id** verdim ve **value**'sine echo vererek `$kullanici_id` diyorum. Bu sayede bilgilerin hangi id'ye göre gidileceği belli olacak. Butonun ismine ise **kullanici-duzenle** diyorum ve **islem.php**'ye giderek altta ki kodları yazıyorum.

```
if (isset($_POST['kullanici-duzenle'])) {  
    $kullanici_id=$_POST['kullanici_id'];  
    $ayarkaydet=$db->prepare("UPDATE kullanici SET  
        kullanici_tc=:kullanici_tc,  
        kullanici_adsoyad=:kullanici_adsoyad,  
        kullanici_gsm=:kullanici_gsm,  
        kullanici_durum=:kullanici_durum  
        WHERE kullanici_id={$_POST['kullanici_id']}");  
    $update=$ayarkaydet->execute(array(  
        'kullanici_tc' => $_POST['kullanici_tc'],  
        'kullanici_adsoyad' => $_POST['kullanici_adsoyad'],  
        'kullanici_gsm' => $_POST['kullanici_gsm'],  
        'kullanici_durum' => $_POST['kullanici_durum']  
    ));  
    if ($update) {  
        header("Location:../production/kullanici-duzenle.php?kullanici_id=$kullanici_id&durum=ok");  
    }else{  
        header("Location:../production/kullanici-duzenle.php?kullanici_id=$kullanici_id&durum=no");  
    }  
}
```

İşlemimi kayıt ettikten sonra, sayfama gidip kontrol ettiğimde istediğim gibi kayıt olduğunu görüyorum. Artık web sitemizde **admin** panelinden istediğimiz gibi kullanıcı bilgilerini görebilir, artırabilir ve düzenleyebiliriz fakat şuan için tek eksik **silmek**. Bunun için hemen **silme** işlemlerine geçiyorum.

```
<td><center><a href="../netting/islem.php?kullanici_id=?php echo $kullanicicek['  
kullanici_id'] ?>&kullanicisil=ok"><button class="btn btn-danger btn-xs">Sil</
```

Bunu fark edebilmek için veritabanımdan bir kaç tane kullanıcı girdisi yapıyorum. Ardından kodlarıma geçiyorum ve düzenle butonuna netting/**islem.php**'ye gönderilecek link vererek **GET** işlemi yapıyorum. Ardından **islem.php** sayfama gidiyorum ve **if koşulu** ile işlemimi yayına geçiriyorum.

```
if ($_GET['kullanicisil']=="ok"){  
  
    $sil=$db->prepare("DELETE FROM kullanıcı WHERE kullanıcı_id=:id");  
    $kontrol=$sil->execute(array(  
        'id' => $_GET['kullanici_id']  
    ));  
  
    if ($kontrol) {  
        header("Location:../production/kullanici.php?sil=ok");  
    }else{  
        header("Location:../production/kullanici.php?sil=no");  
    }  
}
```

## > Menü İşlemleri

(Tablo Tasarımı, Düzenleme, Silme, Listeleme, Aktif/Pasif ve SEO URL)

Menü işlemlerine geçiş yapıyorum. Burada sitemizin anasayfasında ki menüleri dinamik bir hale getireceğim. Bunun için hemen **PHPMyAdmin'e** giderek **MySQL** veritabanı ayarlarımı yapıyorum.

✓ menu tablosu başarılı olarak değiştirildi.

```
ALTER TABLE `menu` CHANGE `menu_id` `menu_id` INT(11) NOT NULL AUTO_INCREMENT;
```

#	Adı	Türü	Karşılaştırma	Öznitelikler	Boş	Varsayılan	Açıklamalar	Ekstra
<input type="checkbox"/>	1	menu_id	int(11)		Hayır	Yok		AUTO_INCREMENT
<input type="checkbox"/>	2	menu_ust	varchar(50)	utf32_turkish_ci	Hayır	Yok		
<input type="checkbox"/>	3	menu_ad	varchar(100)	utf32_turkish_ci	Hayır	Yok		
<input type="checkbox"/>	4	menu_detay	text	utf32_turkish_ci	Hayır	Yok		
<input type="checkbox"/>	5	menu_url	varchar(250)	utf32_turkish_ci	Hayır	Yok		
<input type="checkbox"/>	6	menu_sira	varchar(50)	utf32_turkish_ci	Hayır	Yok		
<input type="checkbox"/>	7	menu_durum	enum('0', '1')	utf32_turkish_ci	Hayır	1		
<input type="checkbox"/>	8	menu_seourl	varchar(250)	utf32_turkish_ci	Hayır	Yok		

Aynı işlemleri sürekli yapmak istemediğimden dolayı **Kullanıcı İşlemleri Modülünde** yapmış olduğum ve kullanımda sıkıntı çıkmadığı için sayfa içerisinde **CTRL-H** tuşuna basarak “**kullanici**” olan metinlerin hepsini “**menu**” olarak değiştiriyorum. Artık **\$kullanicicek['kullanici\_id']** yerine **\$mencek['menu\_id']** yazabilirim ve ardından veritabanımda ki bilgileri giriyorum.

Menu Adı	Menu URL	Menu Sıra	Menu Durum		
Hakkımızda	hakkimizda.php	0	Aktif	Düzenle	Sil

Aktif/Pasif işleminin yazdırılması için basit bir **if koşulu** tutuyorum. Ardından kaç tane menüm olduğunu bulabilmek için de **\$say=0;** diye bir değişken oluşturup, **while döngümün sonuna \$say++** ekliyorum. Artık menü sayısı arttıkça kaç tane menüm olduğunu bana otomatik olarak gösterecek.







Ardından gerekli kodlarımızı yazıyoruz.

```
if ($_GET['menusil']=="ok"){  
  
    $sil=$db->prepare("DELETE FROM menu WHERE menu_id=:id");  
    $kontrol=$sil->execute(array(  
        'id' => $_GET['menu_id']  
    ));  
  
    if ($kontrol) {  
        header("Location:../production/menu.php?sil=ok");  
    }else{  
        header("Location:../production/menu.php?sil=no");  
    }  
}
```

Deneme yaptığımızda bakıyoruz ki, işlemimiz başarılı. Zaman kaybetmeden hemen yeni eklemiş olduğumuz **Yeni Ekle** butonuna dinamiklik getirmekte.

Az önce oluşturmuş olduğum **menu-duzenle.php** sayfasını tamamen kopyalayıp, **menu-ekle.php** olarak değiştiriyorum. İşlemlerimizde yazdırma işlemi olmadığı için value ve php kod parçalarını siliyorum ve isim ayarlamaları yapıyorum. En sona gelerek butona **menukaydet** ismi yazıyorum ve **islem.php** ye gidiyorum. Bu zamana kadar bütün işlemlerimizi **SQL** ortamında **UPDATE** veya **DELETE** yapıyorduk. Şuanda **INSERT** işlemi yaptırarak istediğim butonu işlev kazandırmış oluyorum.

```
if (isset($_POST['menukaydet'])) {  
  
    $menu_seourl=seo($_POST['menu_ad']);  
  
    $ayarekle=$db->prepare("INSERT INTO menu SET  
        menu_ad=:menu_ad,  
        menu_detay=:menu_detay,  
        menu_url=:menu_url,  
        menu_sira=:menu_sira,  
        menu_seourl=:menu_seourl,  
        menu_durum=:menu_durum  
    ");  
  
    $insert=$ayarekle->execute(array(  
        'menu_ad' => $_POST['menu_ad'],  
        'menu_detay' => $_POST['menu_detay'],  
        'menu_url' => $_POST['menu_url'],  
        'menu_sira' => $_POST['menu_sira'],  
        'menu_seourl' => $menu_seourl,  
        'menu_durum' => $_POST['menu_durum']  
    ));  
}
```

Menü işlemlerimizin bazılarını yaptık. Şimdi sıra anasayfada yazdırılmasında bunun için **productions** yani anasayfamızın olduğu dosyalardan **header.php**'ye gidiyoruz ve menü'nün html ile yazılmış kısmını buluyoruz. Tüm kodları silip sadece bir tane **li** bırakıyoruz. Onun ismini de **index.php**'ye gönderip **Anasayfa** yazıyoruz ve altına **php taglarımıza açıp veritabanı bağlantımızı** yaptırılmasını istediğimiz kodları yazıyor ve **while döngüsü** oluşturuyoruz.

```
<div class="navbar-collapse collapse">
  <ul class="nav navbar-nav">
    <li><a href="index.php" class="active">Anasayfa</a><div class="
    curve"></div></li>

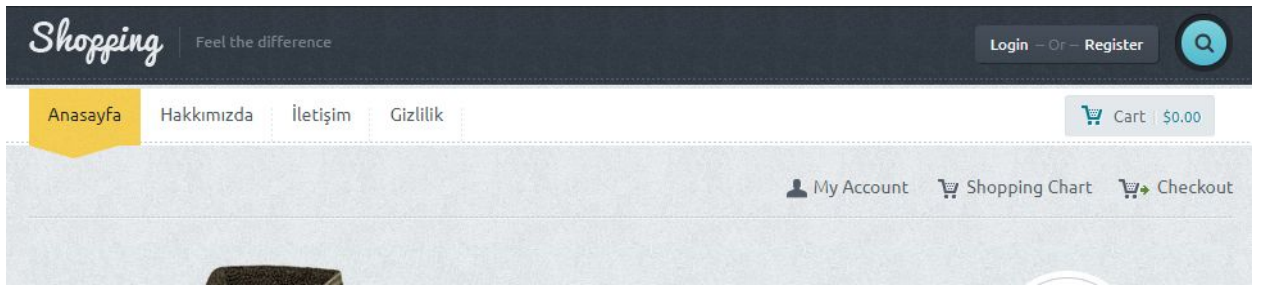
    <?php
    $menusor=$db->prepare("SELECT * FROM menu ORDER BY menu_sira ASC
        limit 5");
    $menusor->execute();
    while ($mencek=$menusor->fetch(PDO::FETCH_ASSOC)) {?>

    <li><a href="<?php echo $mencek['menu_url']; ?>"><?php echo $
        mencek['menu_ad']; ?></a></li>
    <?php } ?>
  </ul>
</div>
```

**SQL komutlarımızı** yazarken **ORDER BY** yapmamın sebebi sıralı bir şekilde hizalanmasını istememden dolayıydı. **limit 5** olmasının sebebi de daha fazla değer baktığımda nav kısmının görüntüsünü bozuyordu. Bir de beraber bakalım sitemiz üstünde ki değişikliğe.

Menü Ayarları							<a href="#">Yeni Ekle</a>
Show	10	entries	Search:				
Sıra Numarası	Menu Adı	Menu URL	Menu Sıra	Menu Durum			
1	Hakkımızda	hakkimizda.php	0	Aktif	<a href="#">Düzenle</a>	<a href="#">Sil</a>	
2	İletişim	iletisim.php	1	Aktif	<a href="#">Düzenle</a>	<a href="#">Sil</a>	
3	Gizlilik	deneme.php	2	Aktif	<a href="#">Düzenle</a>	<a href="#">Sil</a>	

### Admin Paneli Görüntüsü



### Menü kısmı site görüntüsü

İşlem başarılı gibi görünüyor fakat tabiki daha bitmedi. **Aktif/Pasif** ayar sayfamı yapmam gerekiyor. Bu işlem gayet basit az önce ki yazmış olduğum **SQL**'de **WHERE sorgusu** yapıp **menu\_durum=:durum** bir altına inip **arraya** da 'durum'=>1 diye yazdırdığımız da artık sadece aktif olanlar listelenecek.

Şimdi **Menü Detay** sayfasını oluşturacağım. Aslında normal de menü detay sayfası olmaz direkt olarak işlemleri listeler. Fakat benim yapmak istediğim eğer listeleme durumu yapılmak istenilmediğinde aynı yerden basitçe **sayfa** oluşturmak. Bunun için **menü detay** modülünü yazdım. İşlemlerimize geçelim. Daha önceden **hakkimizda.php** oluşturmuştuk onu tamamen kopyalayıp, ismini **menu-detay.php** yapıyorum. Ardından **CTRL-H** kombinasyonu ile hakkımızda olan kelimelerin hepsini menu olarak değiştiriyorum. Bu kısım bu şekilde dursun **SEO URL** yapımına geçiyorum. Bunun için **.htaccess** sayfama gidiyorum. Altta ki kodu yazıyorum

```
RewriteRule ^sayfa-([0-9a-zA-Z-]+) menu-detay.php?sef=$1 [L,QSA]
```

Daha sonra **header**'a giderek a href kısmına **if koşulu** tutuyorum ve altta ki kodlarımı yazıyorum.

```
<?php
$menusor=$db->prepare("SELECT * FROM menu WHERE menu_durum=:durum
ORDER BY menu_sira ASC limit 5");
$menusor->execute(array(
    'durum'=>1
));
while ($menucek=$menusor->fetch(PDO::FETCH_ASSOC)) {?>

    <li><a href="
        <?php
        if (!empty($menucek['menu_url'])) {
            echo $menucek['menu_url'];
        }else{
            echo "sayfa-".seo($menucek['menu_ad']);
        }
        ?>
        "><?php echo $menucek['menu_ad']; ?></a></li>
    <?php } ?>
```

Burada **htaccess** kullanıp fonksiyon yazmadan ben bu işi yapabilirdim. Fakat ileri de, kategoriler gibi ayar yerleri geleceği için bu işlemler beni daha fazla zorlayacaktı. Artık linklerim sitenin sonuna tam, uzantısız olarak **bu-sekilde-gidiyor**.

Sanırım menü ile ilgili olarak sadece sayfalama özelliği kaldı. **\$\_GET** verisini **.htaccess** olarak kullanacağım. Yukarı da yazdığım kod parçaları bu işlem için geçerliydi. **menu-detay** sayfamı oluşturmuştum içine girerek parçalama işlemleri yapıyorum.

```
$menusor=$db->prepare("SELECT * FROM menu where menu_seourl=:sef");
$menusor->execute(array(
    'sef' => $_GET['sef']
));
$menucek=$menusor->fetch(PDO::FETCH_ASSOC);
```

**.htaccess** yapısında olan artık sayfalarımın hangi id ile geldiğini bildiğim için **\$\_GET** döndürüyorum.

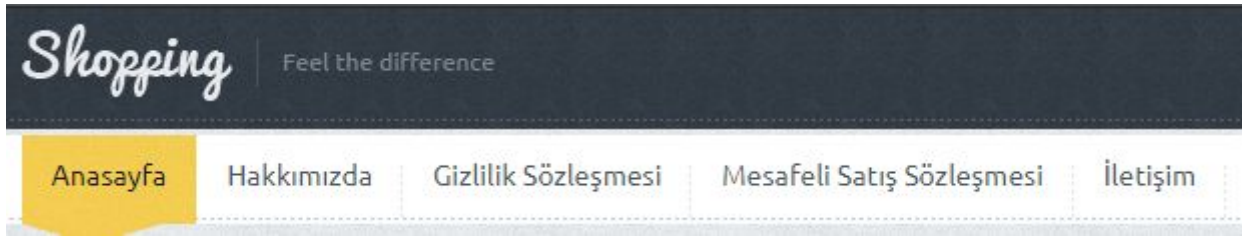


Biraz daha aşağıya gelerek metnin yazdırılmasını istediğim yerlere **echo** komutumu kullanıyorum.

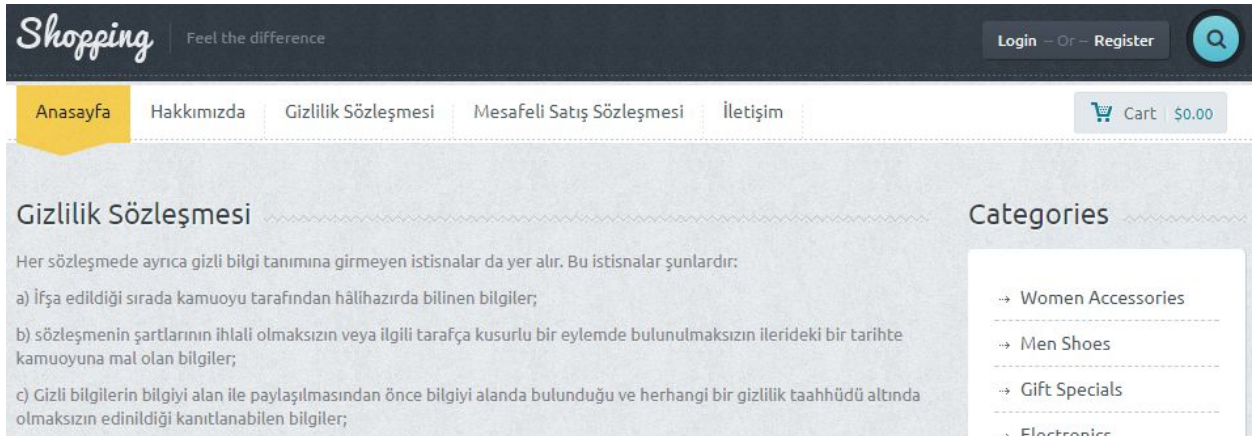
```
<div class="title-bg">
    <div class="title"><?php echo $menucek['menu_ad'] ?></div>
</div>

<div class="page-content">
    <p>
        <?php echo $menucek['menu_detay']; ?>
    </p>
</div>
```

İşlemlerimin hepsi tamam gibi görünüyor. Siteme bakıyorum.



Artık **yönetim panelinden** eklediğim tüm menüler eklenmiş durumda. Herhangi birinin içine giriyorum.



İstediğim gibi görünüyor. Artık menü sayfasından istersem kategorileme, boş döndürüp içerik girersem de sayfa özelliği taşıyor.

## > Logo İşlemleri

(Logo Yükleme Alanı, Kodlaması, Unlink, Çekme ve Listeme İşlemi)

Logo işlemlerine geçiş yapıyorum. Bu işlem biraz zaman ayrılabilir. Yönetim panelinden değişiklik yapılacağı için yönetim panelime giderek **genel-ayar.php** dosyamı buluyorum ve kodlarımı yazmaya başlıyorum.

```
<form action="../../../netting/islem.php" method="POST" enctype="multipart/form-data" data-parsley-validate class="form-horizontal form-label-left">

  <div class="form-group">
    <label class="control-label col-md-3 col-sm-3 col-xs-12" for="first-name">Yüklü Logo<br><span class="required">*</span></label>
    <div class="col-md-6 col-sm-6 col-xs-12">

      <?php
      if (strlen($ayarcek['ayar_logo'])>0) {?>

        <?php } else {?>

        <?php } ?>

    </div>
  </div>
```

**if** koşulunda **strlen** kullanıyorum. **strlen** bir değişkenin içerisinde ne kadar karakter olduğunu hesaplar. Bu kullanım da, diyorum ki eğer **ayarcek**'te ki **ayar\_logo** sıfırdan büyükse yani böyle bir yükleme işlemi varsa yüklenilecek olan resimin yolunu yazıyorum, yoksa da **dimg** dosyasının içerisine atmış olduğum **logo-yok.png** isimli dosyayı ekrana yazdır diyorum.

```
<div class="form-group">
  <label class="control-label col-md-3 col-sm-3 col-xs-12" for="first-name">Resim Seç<span class="required">*</span></label>
  <div class="col-md-6 col-sm-6 col-xs-12">
    <input type="file" id="first-name" name="ayar_logo" class="form-control col-md-7 col-xs-12">
  </div>
</div>

<input type="hidden" name="eski_yol" value="../../../?php echo $ayarcek['ayar_logo']; ?>">

<div align="right" class="col-md-6 col-sm-6 col-xs-12 col-md-offset-3">
  <button type="submit" name="logoduzenle" class="btn btn-primary">Güncelle</button>
</div>
```



İşlemlerimin hepsi diğerleri gibi **islem.php**'ye gönderiyorum. Bu gönderim de **Güncelle** butonumun ismini **logoduzenle** olarak belirliyor ve **islem.php**'ye geçiyorum. Burada uzun bir kod parçası yazıyorum.

```
$uploads_dir = '../../dimg';

$tmp_name = $_FILES['ayar_logo']['tmp_name'];
$name = $_FILES['ayar_logo']['name'];

$benzersizsayi4=rand(20000,32000);
$refimgyol=substr($uploads_dir, 6)."/". $benzersizsayi4.$name;

move_uploaded_file($tmp_name, "$uploads_dir/$benzersizsayi4$name");

$duzenle=$db->prepare("UPDATE ayar SET
    ayar_logo=:logo
    WHERE ayar_id=0");
$update=$duzenle->execute(array(
    'logo' => $refimgyol
));

if ($update) {

    $resimsilunlink=$_POST['eski_yol'];
    unlink("../../$resimsilunlink");

    Header("Location:../production/genel-ayar.php?durum=ok");

} else {

    Header("Location:../production/genel-ayar.php?durum=no");

}
```

Burada aşağıya doğru ok çizdiğim kırmızı'dan sonrası ki olaylar defalarca yaptığımız olaylar. Fakat yukarıda kileri açıklamak istersek **\$uploads\_dir** diye bir değişken oluşturuyorum ve logonun yükleneceği alanı yazıyorum. Ardından **tmp\_name** isminde ki değişken oluşturup **\$\_FILES** tmp\_name diyorum. Onun ardından **benzersizsayi4** diye bir değişken oluşturuyorum ve sallamasyon rakam aralığı giriyorum. **substr** yaparak ta 6 karakterden sonrasını ele alması gerektiğini söylüyorum. Bunun sebebi resim yolu ../../dimg olmasından dolayıdır. Direkt olarak **dimg** dosyasını seçiyor. **benzersizsayi** değişkeni ile birlikte de resim yüklenmeden önce dosyanın

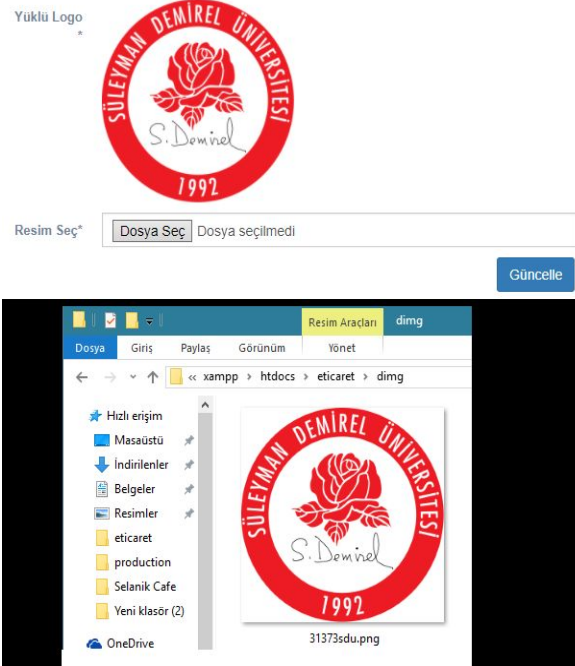
ismini rastgele rakamlar vererek dosya yüklenildiğinde aynı isim varsa karşımıza problem olarak çıkmamasını sağlaması için. Durumu kontrol ediyorum.



Ekranda **sdu.png** isminde bir dosyam var ve sitemin logo yükleme modülüne bakıyorum. Resim yüklenmediği için yazdığım komut doğrulanmış ve **logo-yok.png** dosyasını yazdırmış.



Dosya yüklemesi yaptığım da istediğim gibi logo karşıma çıkıyor.



Benzersizisim diye yazmış olduğum kod parçası da gayet güzel rastgele sayı üreterek karşıma çıktı sağlıyor.

Fakat şöyle bir olay dönüyor. Ne kadar resim yüklersem orayı galeri gibi kullanıyor ve dosyalarım orada tutuluyor. Bu bana ileri ki zamanlar da fazla dosya yüklememden dolayı orayı şişirecek ve boşa alan kaybı yapacak. Bunun için **unlink** fonksiyonunu kullanıyorum. **unlink fonksiyonu** php'de fazla dosyaları silmeyi yapıyor. Daha önce kodlarımı oluşturken **genel-ayar.php**'de eskiyol diye gizli bir input yazdım. İşlem olarak logo'yu çeviriyordu. Şimdi ise **islem.php**'de aşağıda ki kodları yazıyorum. Artık bir logo yüklenirken diğer logoyu silecek.

```
if ($update) {  
    $resimsilunlink=$_POST['eski_yol'];  
    unlink("../$resimsilunlink");  
}
```

Logo işlemlerim her ne kadar **tamamlandı gibi gözükse bile** tamamlanmadı. Çünkü **güvenlik olarak çok zayıf**. Fakat bunları tüm işlemlerim bittikten sonra geri dönerek yapacağım. Şimdi genel olarak iyi duruyor. Logoyu ekrana yazdıralım. Bunun için **header.php**'ye gidiyor ve logomun olduğu html kodunu buluyorum ve **php echo** yazdırma komutumu giriyorum.



Her ne kadar orantısız olsa bile yazdırıldığını görmek mutluluk verici. :) Width/height ayarlarını yapıyoruz ve tertemiz bir logoya sahip oluyoruz.

## > Slider İşlemleri

(MySQL Tablo Oluşturma, Admin Listeleme, Yeni Slider Ekleme, Link Verme)

Her modüle başladığımız gibi bu modüle de başlarken tablo yaparak başlıyoruz. **PhpMyAdmin**'e giderek **E-Ticaret** veritabanımdan yeni bir tablo oluştur diyerek başlıyorum.



The screenshot shows the phpMyAdmin interface for a database named 'eticaret'. The 'slider' table is selected, and the 'Tablo yapısı' (Table structure) tab is active. The table has 6 columns:

#	Adı	Türü	Karşılaştırma	Öznitelikler	Boş	Varsayılan	Açıklamalar	Ekstra	Eylem
1	slider_id	int(11)			Hayır	Yok		AUTO_INCREMENT	Değiştir Kaldır
2	slider_ad	varchar(100)	utf32_turkish_ci		Hayır	Yok			Değiştir Kaldır
3	slider_resimyol	varchar(250)	utf32_turkish_ci		Hayır	Yok			Değiştir Kaldır
4	slider_sira	int(2)			Hayır	Yok			Değiştir Kaldır
5	slider_link	varchar(250)	utf32_turkish_ci		Hayır	Yok			Değiştir Kaldır
6	slider_durum	enum('0', '1')	utf32_turkish_ci		Hayır	1			Değiştir Kaldır

Böyle bir tablo oluşturdum. Buna en yakın örnek olarak daha önceden yapmış olduğum menü sayfam var. Bunun için bunu kullanabilirim diye düşünüyorum.

Bana en yakın sayfanın **menü sayfası** olduğu dikkatimi çekiyor ve onu kopyalayarak üzerinde değişiklikler yapıyorum. Ardından ekleme oldukça aşağıya ineceği için **while döngüsüne** sokuyorum. İşlemlerim **slider-duzenle.php** sayfamı gidecek. Slider düzenle sayfamı da **menü ekle** sayfasının yardımı ile hızlıca yapıyorum ve **islem.php**'ye giderek asıl olaya giriş yapıyorum. **sliderkaydet** işlemini gerçekleştiriyorum. **Slider kaydet**'in asıl mantığı **logo**'da ki gibi benzersiz resim çekmesini sağlıyorum. Ana olarak 4 tane olduğu için 4 adet **benzersizsayı** üreterek ismine yazdırıyorum.

```
$uploads_dir = '../..//dimg/slider';
$tmp_name = $_FILES['slider_resimyol']['tmp_name'];
$name = $_FILES['slider_resimyol']['name'];
//resmin isminin benzersiz olması
$benzersizsayi1=rand(20000,32000);
$benzersizsayi2=rand(20000,32000);
$benzersizsayi3=rand(20000,32000);
$benzersizsayi4=rand(20000,32000);
$benzersizad=$benzersizsayi1.$benzersizsayi2.$benzersizsayi3.$benzersizsayi4;
$refimgyol=substr($uploads_dir, 6)."/".$benzersizad.$name;
@move_uploaded_file($tmp_name, "$uploads_dir/$benzersizad$name");
```



Daha sonra **sliderduzenle** işlemine geçiyorum ve altta ki kodları yazıyorum.

```
$uploads_dir = '.././dimg/slider';
$tmp_name = $FILES['slider_resimyel']['tmp_name'];
$name = $_FILES['slider_resimyel']['name'];
$benzersizsayi1=rand(20000,32000);
$benzersizsayi2=rand(20000,32000);
$benzersizsayi3=rand(20000,32000);
$benzersizsayi4=rand(20000,32000);
$benzersizad=$benzersizsayi1.$benzersizsayi2.$benzersizsayi3.$benzersizsayi4;
$refimgyol=substr($uploads_dir, 6)."/".$benzersizad.$name;
@move_uploaded_file($tmp_name, "$uploads_dir/$benzersizad$name");

$duzenle=$db->prepare("UPDATE slider SET
    slider_ad=:ad,
    slider_link=:link,
    slider_sira=:sira,
    slider_durum=:durum,
    slider_resimyel=:resimyel
    WHERE slider_id={$_POST['slider_id']}");
$update=$duzenle->execute(array(
    'ad' => $_POST['slider_ad'],
    'link' => $_POST['slider_link'],
    'sira' => $_POST['slider_sira'],
    'durum' => $_POST['slider_durum'],
    'resimyel' => $refimgyol,
));

$slider_id=$_POST['slider_id'];

if ($update) {

    $resimsilunlink=$_POST['slider_resimyel'];
    unlink(".././$resimsilunlink");

    Header("Location:../production/slider-duzenle.php?slider_id=$slider_id&durum=ok");

} else {

    Header("Location:../production/slider-duzenle.php?durum=no");

}
```

```
} else {

    $duzenle=$db->prepare("UPDATE slider SET
        slider_ad=:ad,
        slider_link=:link,
        slider_sira=:sira,
        slider_durum=:durum
        WHERE slider_id={$_POST['slider_id']}");
    $update=$duzenle->execute(array(
        'ad' => $_POST['slider_ad'],
        'link' => $_POST['slider_link'],
        'sira' => $_POST['slider_sira'],
        'durum' => $_POST['slider_durum']
    ));

    $slider_id=$_POST['slider_id'];

    if ($update) {

        Header("Location:../production/slider-duzenle.php?slider_id=$slider_id&durum=ok");

    } else {

        Header("Location:../production/slider-duzenle.php?durum=no");





    }

}

}
```

Burası işin can alıcı kısmı oluyor. Biraz uğraştırıyor fakat projeme baktığımda değiştiğini hissediyorum.




S.No	Resim	Ad	Url	Sıra	Durum		
1		Slider 1 Başlığı		1	Aktif	Düzenle	Sil
2		Slider 2 Başlığı	2	2	Aktif	Düzenle	Sil
3		Slider 3 Başlığı	3	3	Aktif	Düzenle	Sil
4		Slider 4 Başlığı	4	4	Aktif	Düzenle	Sil

### Slider menü sayfası

Resim Seç*	<div>Dosya Seç Dosya seçilmedi</div>
Slider Ad *	<input type="text" value="Slider adını giriniz"/>
Slider Url *	<input type="text" value="Slider Link giriniz"/>
Slider Sıra *	<input type="text" value="Slider sıra giriniz"/>
Slider Durum*	<div>Aktif</div>

Kaydet

### Slider ekleme sayfası

Slider Adı*	<input type="text" value="Slider 1 Başlığı"/>
Yüklü Slider *	
Slider Resmi*	<div>Dosya Seç Dosya seçilmedi</div>
Slider Url *	<input type="text" value="Slider Link giriniz"/>
Slider Sıra *	<input type="text" value="1"/>
Slider Durumu *	<div>Aktif</div>

Güncelle

### Slider düzenleme sayfası

Şimdi sıra bunu ön tarafta göstermeye geldi. İşlemlerim **index.php**'deydi ve slider ile ilgili tüm html kodlarını kesiyorum ve yeni bir sayfaya yapııştırıyorum bu sayfamın adı **slider.php** olacak. Sonra tekrar index.php'ye gidip **include** işlemi yapıyorum. **slider.php**'de içerisinde sadece 1 tane içerik olmasını sağlıyor ve **veritabanı bağlantımı yapıyorum**. Ardından **while döngüsü** döndürüyorum. İçerik kısmında daha önceden yazılmış tüm metinleri silerek veritabanımdan çektiğim değişkenlerin isimlerini yazıyorum.

```
<?php

$slidersor=$db->prepare("SELECT * FROM slider");
$slidersor->execute();

while($slidercek=$slidersor->fetch(PDO::FETCH_ASSOC)) { ?>

<div class="item">
  <div class="slide-desc">
    <div class="inner">
      <h1><?php echo $slidercek['slider_ad']; ?></h1>
      <p>
        <?php echo $slidercek['slider_icerik']; ?>
      </p>
      <button class="btn btn-default btn-red btn-lg">Add to cart</button>
    </div>
    <div class="inner">
      <div class="pro-pricetag big-deal">
        <div class="inner">
          <span class="oldprice">$314</span>
          <span>$199</span>
          <span class="ondeal">Best Deal</span>
        </div>
      </div>
    </div>
  </div>
  <div class="slide-type-1">
    <a href="<?php echo $slidercek['slider_link']; ?>"></a>
  </div>
</div>
<?php } ?>
```

Ekranda **slider\_ad**, **slider\_icerik**, **slider\_link** ve **slider\_resim yol** isimli verilerin çıktı olarak yazdırılmasını istiyor ve işlemlerimi kayıt ediyorum. Kayıt işleminden sonra **localhost**'uma giriyor ve sayfayı güncelliyorum. İstediğim gibi işlemlerimin tamamı kayıt edilmiş durumda.



## > Login İşlemleri

(MySQL Tablo Oluşturma, Register ve Login İşlemleri)

Login işlemlerine giriş yapıyorum. Bu modül sayesinde kullanıcılar sitemizin içerisinden giriş yapabilecek veya kayıt olabilecek. Bunun için **header.php**'ye gidip öncelikle Türkçeleştirme yapıyorum.

Daha sonra kayıt ol butonuna **register.php** ismini veriyor ve temamın kayıt olma sayfasını kendime göre düzenliyorum.

Diğer işlemler de olduğu gibi bunda da **parçalama** işlemi yapıp dosyalarımı **include** yaptım.

Daha sonra **islem.php**'ye gelerek `if(isset($_POST['kullanici kaydet']))` diyerek kodlarımı yazmaya başlıyorum.

```
if (isset($_POST['kullanici kaydet'])) {  
  
    echo $kullanici_adsoyad=htmlspecialchars($_POST['kullanici_adsoyad']); echo "<br>";  
    echo $kullanici_mail=htmlspecialchars($_POST['kullanici_mail']); echo "<br>";  
  
    echo $kullanici_passwordone=$_POST['kullanici_passwordone']; echo "<br>";  
    echo $kullanici_passwordtwo=$_POST['kullanici_passwordtwo']; echo "<br>";  
}
```

Burada ilk defa kullandığım bir fonksiyon var. Bu fonksiyonun adı ise **htmlspecialchars** kullanıcı kayıt olurken isim soyisim ve mail girişinin ismini bunu veriyorum. Bu fonksiyonun anlamı zararlı kod parçacıklarını mesela **javascript script** veya **meta gibi** taglar kullanarak zarar vermeye çalışan kişilerin kodlarını bozmaya yarıyor. **passwordone** ve **passwordtwo** değişkenlerim, form'da ki **1. şifre** ve **2. şifre** inputların isimleri. Onları görebilmek için post ediyorum. Daha sonra aşağıya gelerek onlara işlev kazandırıyorum.

```
if ($kullanici_passwordone==$kullanici_passwordtwo) {  
  
    if ($kullanici_passwordone>=6) {  
  
        $kullanicisor=$db->prepare("select * from kullanıcı where kullanıcı_mail=:mail");  
        $kullanicisor->execute(array(  
            'mail' => $kullanici_mail  
        ));  
  
        //dönen satır sayısını belirtir  
        $say=$kullanicisor->rowCount();  
  
        if ($say==0) {  
  
            //md5 fonksiyonu şifreyi md5 şifreli hale getirir.  
            $password=md5($kullanici_passwordone);  
  
            $kullanici_yetki=1;  
  
            //Kullanıcı kayıt işlemi yapılıyor...  
            $kullanici kaydet=$db->prepare("INSERT INTO kullanıcı SET  
                kullanıcı_adsoyad=:kullanici_adsoyad,  
                kullanıcı_mail=:kullanici_mail,  
                kullanıcı_password=:kullanici_password,  
                kullanıcı_yetki=:kullanici_yetki  
            ");  
            $kullanici kaydet->execute(array(  
                'kullanici_adsoyad' => $kullanici_adsoyad,  
                'kullanici_mail' => $kullanici_mail,  
                'kullanici_password' => $password,  
                'kullanici_yetki' => $kullanici_yetki  
            ));  
        }  
    }  
}
```



Eğer kullanıcı şifresi1 ve şifresi2 değişkenleri doğruysa kod parçası aşağıya iniyor ve bir tane daha **if koşulunun** içine giriyor burda da **6'dan büyük veya 6'ya eşit olduğunu** tespit ediyor. O da doğruysa içine girerek bilgileri doğruluyor. Bizim girişimiz **e-posta** ile olacağı için klasik veritabanı bağlantı işlemlerimizi yapıyoruz. İşlemlerimiz tamamlandıktan sonra ise **rowCount()** kullanıyorum. **rowCount()** işlemin içinde bilgi varsa devam eder ve boş geçilmesini önler. Bu **mükerrer kayıt** olmasını önüyor yani **aynı mail ile tekrar tekrar kayıt olunmasını önler**. \$say diye bir değişken oluşturup içine devam ediyorum ve eğer say değişkeni **sıfıra eşit ise** password değerini **md5 olarak şifrelenmesini sağlıyor** ve kullanıcı yetkisine **1** veriyor. **Daha önceden yaptığım gibi yetki 5 olursa** yönetim paneline giriş yapılabilirdi. Fakat **1 default** olarak vereceğim yetki olduğu için sadece kullanıcı işlemlerinde kullanılacak. Devamı da bildiğimiz veritabanı bağlantısı. Deneme işlemi yapıyorum ve işlem gayet başarılı :)

kullanici_password	kullanici_adsoyad	kullanici_adres	kullanici_il	kullanici_ilce	kullanici_unvan	kullanici_yetki
2583197bf317d53b18c582ba2826d6ed	Samet Samyeli	YENİ MAHALLE YAŞAM SOKAK	Isparta	Eğirdir	Yonetici	5
827ccb0eea8a706c4c34a16891f84e7b	mustafa samyeli					1

MD5 ile şifrelenmiş şifre      deneme için isim      kullanıcı yetkisi 1

Register işlemlerim bu kadardı. Şimdi ise **Login** işlemlerine giriş yapıyorum.

Login işlemi ayrı sayfa olmayacak sadece **header**'da bulunduğu için **header.php**'ye gidiyorum. Buradan gerekli ayarlamaları yapacağım.

[Giriş Yap](#) — yada — [Kayıt Ol](#)

### Kullanıcı Giriş

[Giriş Yap](#)

### Kayıt Ol

Yeni Kullanıcı mısınız? Alışverişe başlamak için hemen kayıt olmalısınız!

[Kayıt Ol](#)



**form action="islem.php" method="post"** diyerek form üstünde oynama yapıyorum. Daha sonra aşağıda bulunan **input** alanlarının isimlerini değiştiriyorum. **kullanici\_mail** ve **kullanici\_password** olacak ve butonun ismini **kullanicigiris** diyerek **islem.php**'ye geçiş yapıyorum.

```
if (isset($_POST['kullanicigiris'])) {  
    echo $kullanici_mail=htmlspecialchars($_POST['kullanici_mail']);  
    echo $kullanici_password=md5($_POST['kullanici_password']);  
  
    $kullanicisor=$db->prepare("select * from kullanici where kullanici_mail=:mail and kullanici_yetki=:yetki and  
        kullanici_password=:password and kullanici_durum=:durum");  
    $kullanicisor->execute(array(  
        'mail' => $kullanici_mail,  
        'yetki' => 1,  
        'password' => $kullanici_password,  
        'durum' => 1  
    ));  
  
    $say=$kullanicisor->rowCount();  
  
    if ($say==1) {  
        echo $_SESSION['userkullanici_mail']=$kullanici_mail;  
        header("Location:../../");  
    } else {  
        header("Location:../../?durum=basarisizgiris");  
    }  
}
```

Burada açıklayacak olursam daha önce ki işlemlerin bir kaçını burada tekrar kullanıyorum. Sebebi ise kullanıcı register olup giriş yaparken işlemlerin karışmaması için. **kullanici\_password**'u md5 olarak çağdırmamın sebebi ise kayıt olunurken **md5** formatında kaydedilmişti. Direkt olarak çağdıracak olursak veritabanında ki farklı olacağı için işlem kayıt edilemeyecekti. Sonra aşağısında **mail, yetki ve durum** çağırıyorum array sokarak devam ediyorum. Durum 1 olacak eğer banlamak/engellemek istersek yönetim panelinde yapmış olduğumuz değişiklikleri işleme sunacak. Yetki her zaman ki gibi 1'de. Sonra aşağıya gelerek rowCount döngüsü yapıyoruz. Eğer işlem doğru bir şekilde girildiyse 1 oluyor ve **\$\_SESSION** işlemlerimize giriyor. **userkullanici\_mail** ismiyle işlem yapıyorum artık **session** işlemimizin adı bu olarak değiştirdim ve anasayfa yönlendiriyor. Değilse de başarısız giriş diyerek giriş sayfasında hata veriyor. **İşlem sayfası bu kadardı**. Header'e giderek üstünde değişiklikler yapıyorum. Header işlemlerinde **session işlemi** yapacağım için sayfanın en tepesine veritabanı bağlantımızı yazıyorum.

```
$kullanicisor=$db->prepare("SELECT * FROM kullanici where kullanici_mail=:mail");  
$kullanicisor->execute(array(  
    'mail' => @$_SESSION['userkullanici_mail']  
));  
$say=$kullanicisor->rowCount();  
$kullanicicek=$kullanicisor->fetch(PDO::FETCH_ASSOC);
```

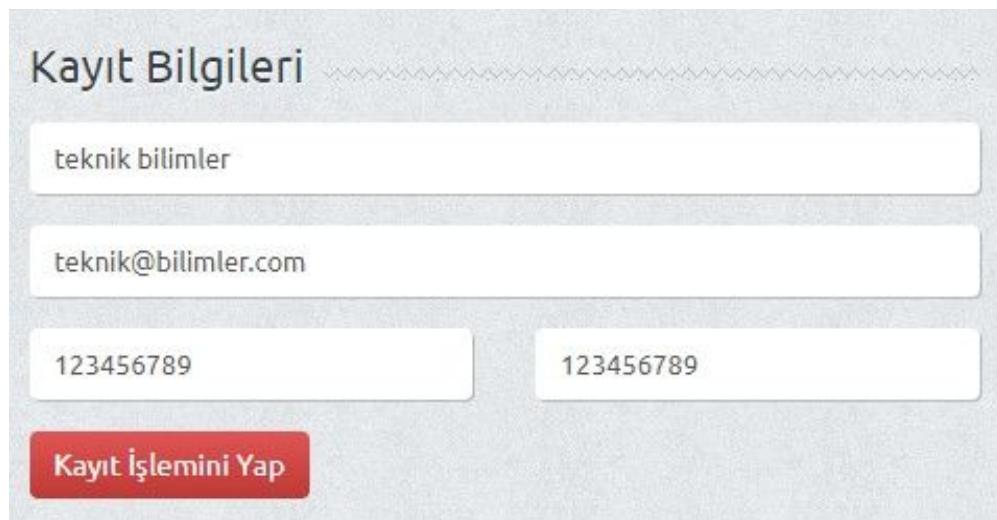
Veritabanı giriş işlemlerimizi yaptıktan sonra aşağıya doğru inip iki tane işlem yapıyorum.

```
<?php
if (!isset($_SESSION['userkullanici_mail'])) {?>
<a href="#" id="reg" class="btn btn-default btn-dark">Giriş Yap<span>-- yada --</span>Kayıt Ol</a>
<?php } else { ?>
<a href="#" class="btn btn-default btn-dark">Hoşgeldin<span>--</span><?php echo $kullanici['kullanici_adsoyad']
?></a>
<?php } ?>
```

Artık giriş yaptıktan sonra kart bilgileri, hesap bilgileri vb. gösterilecekken aynı zamanda giriş yap ve kayıt ol kısımları değil üye olan kullanıcının adı ve soyadı yazacak.

```
<?php
if (isset($_SESSION['userkullanici_mail'])) {?>
<ul class="small-menu">
<li><a href="hesabim" class="myacc">Hesap Bilgilerim</a></li>
<li><a href="siparislerim" class="myshop">Siparişlerim</a></li>
<li><a href="logout" class="mycheck">Güvenli Çıkış</a></li>
</ul>
<?php }
?>
```

Komple yaptığımızın deneme görüntüsünü buraya aktaracak olursam;



Kayıt oluyoruz. İsim soyisim: teknik bilimler

Giriş Yap -- yada -- Kayıt Ol

### Kullanıcı Giriş

teknik@bilimler.com

\*\*\*\*\*

Giriş Yap

### Kayıt Ol

Yeni Kullanıcımısın? Alışverişe başlamak için hemen kayıt olmalısın!

Kayıt Ol

Kullanıcı giriş olarak E-Postayı kabul etmiştik bunun için kayıt olduğumuz E-Posta adresini ve şifreyi yazıyoruz. (123456789)

---

Hoşgeldin -- teknik bilimler

etişim

Cart | \$0.00

Hesap Bilgilerim Siparişlerim Güvenli Çıkış

Giriş yaptıktan sonra bizi karşılayan ekran

---

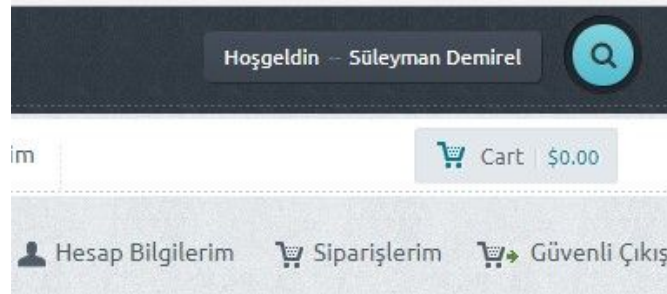
Giriş Yap -- yada -- Kayıt Ol

Giriş yapmadan önce ki görünen ekran

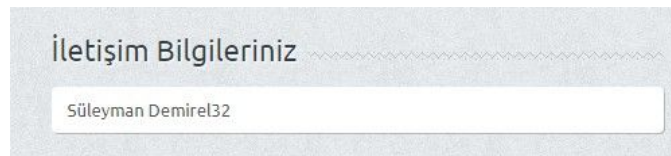
Şimdi ise yaptığımız olayın update işlemini yapmakta. Mantık aynı, değişiklik yok. Butonuma “**kullanicibilgiguncelle**” ismini veriyorum ve her zaman ki gibi **islem.php**’ye gönderiyorum. SQL sorguma **UPDATE** işlemi tanımlıyorum.

```
if (isset($_POST['kullanicibilgiguncelle'])) {  
    $kullanici_id=$_POST['kullanici_id'];  
  
    $ayarkaydet=$db->prepare("UPDATE kullanıcı SET  
        kullanıcı_adsoyad=:kullanici_adsoyad,  
        kullanıcı_il=:kullanici_il,  
        kullanıcı_ilce=:kullanici_ilce  
        WHERE kullanıcı_id={$_POST['kullanici_id']}");  
  
    $update=$ayarkaydet->execute(array(  
        'kullanici_adsoyad' => $_POST['kullanici_adsoyad'],  
        'kullanici_il' => $_POST['kullanici_il'],  
        'kullanici_ilce' => $_POST['kullanici_ilce']  
    ));  
  
    if ($update) {  
        Header("Location:../../hesabim?durum=ok");  
    } else {  
        Header("Location:../../hesabim?durum=no");  
    }  
}
```

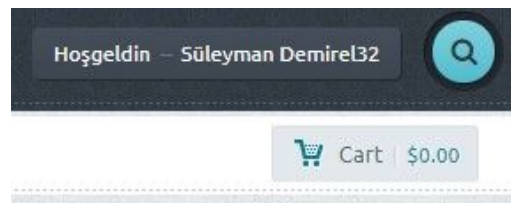
İşlemlerimi kontrol ediyorum.



Son işlemimizi yapmadan önce



Değişiklik yapıyorum ve sonuna **32** ekliyorum.



İşlemimiz istediğimiz gibi kayıt edildi.

## > Kategori İşlemleri

(MySQL Tablo Oluşturma, Ürünler ve Listelenmesi)

Login işlemleri de bittiğine göre kategori işlemlerine geçiş yapıyoruz. Kategori işlemlerinde ürünlerin sıralanacağı ürün listeleme mantığı olarak düşünebiliriz. Bunun için en önce **phpmyadmin**'e gidip **mysql tablomu** oluşturun.

#	Adı	Türü	Karşılaştırma	Öznitelikler	Boş	Varsayılan	Açıklamalar	Ekstra
<input type="checkbox"/>	1	kategori_id	int(11)		Hayır	Yok		AUTO_INCREMENT
<input type="checkbox"/>	2	kategori_ad	varchar(100)	utf8_general_ci	Hayır	Yok		
<input type="checkbox"/>	3	kategori_ust	int(2)		Hayır	Yok		
<input type="checkbox"/>	4	kategori_icon	varchar(50)	utf8_general_ci	Hayır	Yok		
<input type="checkbox"/>	5	kategori_seourl	varchar(250)	utf8_general_ci	Hayır	Yok		
<input type="checkbox"/>	6	kategori_sira	int(2)		Hayır	Yok		

Şimdilik bunlar yazılı devamı için eklemeler yapabiliriz. Kategori icon oluşturmamın sebebi bu tema da yok fakat ilerleyen zamanlar da kendim yazacağım bir sistem de kategorilere icon vermeyi düşünüyorum. Bunun içinde direkt olarak **fontawesome** kullanıp ismini vereceğim.

MySQL işlemlerinde tablo olayı bu kadardı. Şuanda devam edip, yönetim paneline geçiş yapıyorum ve **header**'a ekleme yapıyor ve kategori.php'ye gitmesine söylüyorum.

```
<li><a href="menu.php"><i class="fa fa-list"></i>Menüler</a></li>
<li><a href="kategori.php"><i class="fa fa-bars"></i>Kategoriler</a></li>
```

Daha sonra yönetim panelinde en yakın özellikleri menüler taşıdığı için menu.php'yi kopyalayıp ismini kategori.php olarak değiştiriyorum. İsimleri de aratıp menü yazılarını kategori olarak değiştiriyor ve ekleme yapıyorum. İşlemlerime tekrardan baktığımda veritabanının da ekleme yaptığım bilgiler listelendiğini görüyorum.

Düzenle diyince **menu-duzenle.php**'ye gidiyor. Aynı kodları tekrar tekrar yazmak istemediğimden dolayı da yine aynı mantıkla menu-duzenle.php'yi kopyalıyor ve ismini **kategori-duzenle.php** yapıyorum ve aynı isme dair ne var ise kategori olarak değiştiriyorum.



Düzenle butonuna basıldığında aşağıda ki gibi görünmektedir.

Kategori Adı*	<input type="text" value="Ayakkabılar"/>
Kategori Sıra*	<input type="text" value="0"/>
Kategori Durumu *	<input type="text" value="Aktif"/>
<div>Güncelle</div>	

İşlemler tamam gibi gözükse de güncelle butonuna değer ataması yapmadık. Bunlar kategoriduzenle ismi ile **islem.php**'ye **post** ediliyor fakat orada herhangi bir komut satırı olmadığı için işlemlerimiz boşa gidecektir. Bunun için hemen islem.php'ye gidip kodlarımızı yazmaya başlıyorum.

Formumuzun action kısmında post ettiği ismi if koşulunun içinde isset oluşturarak yazmaya başlıyorum ve aşağıda ki kod blogunu yazıyorum.

```
if (isset($_POST['kategoriduzenle'])) {  
  
    $kategori_id=$_POST['kategori_id'];  
    $kategori_seourl=seo($_POST['kategori_ad']);  
  
    $kaydet=$db->prepare("UPDATE kategori SET  
        kategori_ad=:ad,  
        kategori_durum=:kategori_durum,  
        kategori_seourl=:seourl,  
        kategori_sira=:sira  
        WHERE kategori_id={$_POST['kategori_id']}");  
    $update=$kaydet->execute(array(  
        'ad' => $_POST['kategori_ad'],  
        'kategori_durum' => $_POST['kategori_durum'],  
        'seourl' => $kategori_seourl,  
        'sira' => $_POST['kategori_sira']  
    ));  
  
    if ($update) {  
        Header("Location:../production/kategori-duzenle.php?durum=ok&kategori_id=$kategori_id");  
    } else {  
        Header("Location:../production/kategori-duzenle.php?durum=no&kategori_id=$kategori_id");  
    }  
}
```

Hemen bir deneme yapıyorum ve başarıyla güncelleme işlemleri kayıt edildiğini görüyorum.

```
//Kategori Sil

if ($_GET['kategorisil']=="ok"){

    $sil=$db->prepare("DELETE FROM kategori WHERE kategori_id=:id");
    $kontrol=$sil->execute(array(
        'id' => $_GET['kategori_id']
    ));

    if ($kontrol) {
        header("Location:../production/kategori.php?sil=ok");
    }else{
        header("Location:../production/kategori.php?sil=no");
    }
}
}
```

Güncelleme butonun hemen yanında Silme butonu var. Basit bir kod yardımı ile bu işi kolaylıkla halledebilirim. Bir alt satıra geçip kodlamaya başlıyorum. Bu işlemi daha önceden de yaptığım gibi ekstra bir sayfa oluşturmaktan ziyade **\$\_GET** sayesinde yaptım. Güncelleme işlemi de bittiğine göre artık ekleme olayına geçiş yapabilirim.

```
if (isset($_POST['kategorikaydet'])) {

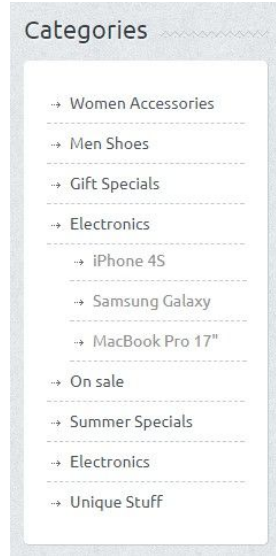
    $kategori_seourl=seo($_POST['kategori_ad']);

    $ayarekle=$db->prepare("INSERT INTO kategori SET
        kategori_ad=:kategori_ad,
        kategori_sira=:kategori_sira,
        kategori_seourl=:kategori_seourl,
        kategori_icon=:kategori_icon,
        kategori_durum=:kategori_durum
    ");

    $insert=$ayarekle->execute(array(
        'kategori_ad' => $_POST['kategori_ad'],
        'kategori_sira' => $_POST['kategori_sira'],
        'kategori_seourl' => $kategori_seourl,
        'kategori_icon' => $_POST['kategori_icon'],
        'kategori_durum' => $_POST['kategori_durum']
    ));
}
```

Kaydet düzenle ismini bir kopya daha oluşturarak ismini kaydet-ekle.php yapıyorum ve üstünde ki kodlarını değiştiriyorum. Value olarak isim verdiğim alanları placeholder yapıp buraya yazı giriniz gibi bilgiler yazdıktan sonra **islem.php**'ye giderek yukarıda ki resimde olan **INSERT INTO SQL** ve **PDO** kodlarımızı yazıyorum. Admin panelimde ki işlemler bu kadardı artık girdilerimizi anasayfada listelemek istiyorum.

Ürünlerimi listelemek istediğim anasayfa da bulunan kategori kısmı aşağıda ki gibi görünüyor.



Bunları öncelikle parçalama işlemi yapmam gerekmekte **index.php**'de bulduğum kodları kesiyorum ve **sidebar.php** diye bir dosya oluşturup içine kayıt ediyorum. Daha sonra tekrar **index.php**'ye gelip **include** işlemi yapıyorum. Kod parçalarım dağıldıktan sonra bunları kodlara bölme işlemi kaldı.

```
<div class="categorybox">
  <ul>
    <li><a href="category.htm">Women Accessories</a></li>
    <li><a href="category.htm">Men Shoes</a></li>
    <li><a href="category.htm">Gift Specials</a></li>
    <li><a href="category.htm">Electronics</a>
      <ul>
        <li><a href="#">iPhone 4S</a></li>
        <li><a href="#">Samsung Galaxy</a></li>
        <li><a href="#">MacBook Pro 17"</a></li>
      </ul>
    </li>
    <li><a href="category.htm">On sale</a></li>
    <li><a href="category.htm">Summer Specials</a></li>
    <li><a href="category.htm">Electronics</a></li>
    <li class="lastone"><a href="category.htm">Unique Stuff</a></li>
  </ul>
</div>
```

Sadece en yukarıda ki **Women Accesories** yazısının kalacak şekilde diğerlerini siliyorum ve üst kısmına veritabanı bağlantımızı yapıyorum. Altına da veriler sırasıyla gelmesi için gerekli olan **while** koşulumu döndürüyorum.

```

<div class="categorybox">
  <ul>
    <?php
      $kategorisor=$db->prepare("SELECT * FROM kategori order by kategori_sira asc ");
      $kategorisor->execute(array());
      while ($kategoricek=$kategorisor->fetch(PDO::FETCH_ASSOC)){?>
        <li><a href="category.htm">Women Accessories</a></li>
      <?php }?>
    </ul>
  </div>

```

**PHP** yazdırma komutu olan **echo** ile beraber de ekrana kolaylıkla yazdırıyorum. (**<?php echo \$kategoricek['kategori\_ad']; ?>**)



Ürün kategorilerini ekleme, düzenleme, silme ve listeleme bu şekildeydi. Şimdi sıra bu ürün kategorilerimizin işlevlik kazanabilmesi için işe yarayacak ürünlerimizi ekleme kısmına geçiş yapıyorum. Artık bu kategorilerin içini yönetici panelinden ürün ekleyerek doldurabiliriz.

## > Ürünler İşlemleri

(MySQL Tablo Oluşturma, Ürünler ve Listelenmesi)

Belki de bu zamana kadar en fazla üzerinde duracağımız konuya giriş yapıyorum. Mantık hep aynı fakat birazcık uğraştıracak tabi hâliyle. Her zaman olduğu gibi işlemime **PHPMYAdmin**'den **MySQL** veritabanıma geçiş yaparak başlıyorum. **urun** diye bir tablo oluşturdum ve aşağıda ki sütunları içine karşısında ki bilgiler doğrultusunda doldurup kayıt ettim.

#	Adı	Türü	Karşılaştırma	Öznitelikler	Boş	Varsayılan	Açıklamalar	Ekstra
1	urun_id	int(11)			Hayır	Yok		AUTO_INCREMENT
2	kategori_id	int(11)			Hayır	Yok		
3	urun_zaman	timestamp			Hayır	CURRENT_TIMESTAMP		
4	urun_ad	varchar(250)	utf8_general_ci		Hayır	Yok		
5	urun_seourl	varchar(250)	utf8_general_ci		Hayır	Yok		
6	urun_detay	text	utf8_general_ci		Hayır	Yok		
7	urun_fiyat	float(9,2)			Hayır	Yok		
8	urun_video	varchar(50)	utf8_general_ci		Hayır	Yok		
9	urun_keyword	varchar(250)	utf8_general_ci		Hayır	Yok		
10	urun_stok	int(11)			Hayır	Yok		
11	urun_durum	enum('0','1')	utf8_general_ci		Hayır	Yok		

Veritabanı tablomuzu oluşturduğumuza göre artık işlemlere başlayabiliriz. Bunun için de **menu.php**'yi kopyalarak ismini **urun.php** yapıyorum. Burada isimleri **urun** olarak değiştiriyorum ve gerekli olan kısımları yazıyorum. Yönetim panelinde ürün adı, ürün zaman, ürün fiyat ve ürün stok göstermek istiyorum ve gerekli kodları yazıyorum. İstiyorum ki fiyat kısmının yanında Türk Lirası simgesini kullanmak. Bunun için **font awesome** kullanarak o kısma ekleme yapıyorum.

```
<?php
$say=0;
while ($suruncak=$surunsor->fetch(PDO::FETCH_ASSOC)) { $say++>

<tr>
<td><?php echo $say; ?></td>
<td><?php echo $suruncak['urun_ad']; ?></td>
<td><?php echo $suruncak['urun_zaman']; ?></td>
<td align="center"><?php echo $suruncak['urun_fiyat'] ?> <i class="fa fa-try" style="color: orange"
aria-hidden="true"></i></td>
<td align="center"><?php echo $suruncak['urun_stok'] ?></td>

<td><center><?php if ($suruncak['urun_durum']==1) {>
<button class="btn btn-success btn-xs">Aktif</button>
<?php }else{>
<button class="btn btn-warning btn-xs">Pasif</button>
<?php } ?></td></center>
<td><center><a href="urun-duzenle.php?urun_id=<?php echo $suruncak['urun_id']; ?>"><button class
="btn btn-primary btn-xs">Düzenle</button></a></center></td>
<td><center><a href="netting/islem.php?urun_id=<?php echo $suruncak['urun_id']; ?>&urunsil=ok">
<button class="btn btn-danger btn-xs">Sil</button></a></center></td>
```



***İşlemleri aynı daha önceden yaptığım gibi olduğundan dolayı teker teker aynı işlemleri tekrar tekrar yapıyorum. Ekle, sil ve düzenle komutları aynı şekilde sadece isimleri değişiyor herhangi bir farklı bir olay yok.***

Kategorileri çekebiliyorduk. Artık bu kategorileri özelleştirmek gerekiyor. Yani tıklandığında ayrıntılı bir sayfaya gönderimi. Tamamdan ilgili sayfayı buluyorum ve Türkçeleştirmeye başlıyorum. Daha sonra gerekli yerleri **include** yapıyor ve işlemlerime başlıyorum.

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

if (isset($_GET['sef'])) {

    $kategorisor=$db->prepare("SELECT * FROM kategori where kategori_seourl=:seourl");
    $kategorisor->execute(array(
        'seourl' => $_GET['sef']
    ));

    $kategoricek=$kategorisor->fetch(PDO::FETCH_ASSOC);

    $kategori_id=$kategoricek['kategori_id'];

    $urunsonor=$db->prepare("SELECT * FROM urun where kategori_id=:kategori_id order by urun_id DESC");
    $urunsonor->execute(array(
        'kategori_id' => $kategori_id
    ));

    $say=$urunsonor->rowCount();

} else {

    $urunsonor=$db->prepare("SELECT * FROM urun order by urun_id DESC");
    $urunsonor->execute();

}
```

Burada ki kodları açıklayacak olursam **\$\_GET** ile çağırdığım **sef** değişkeni **.htaccess**'ten yararlanarak link gitmemi sağlayan seourl kısmı daha önceden fonksiyonlar dosyasına oluşturduğumuz seourl kısmı da ismini **ad**'dan alıyor ve parçalanmış bir isimde tutuyor.

Altında ki ise klasik, ktegori çekirme işlemleri. Eğer **\$\_GET** dönmüyor ise de ürünlerin hepsinin sıralanmasını istiyorum. Ayrıca orada **rowCount()** kullandığım **\$say** değişkeni var. O da isminden anlaşıldığı üzere işlemi saydırmaya yarıyor.

```

<div class="row prdct"><!--Products-->

<?php

if ($say==0) {
    echo "Bu kategoride ürün bulunamadı";
}

while($uruncek=$urunson->fetch(PDO::FETCH_ASSOC)) {
    ?>

    <div class="col-md-4">
        <div class="productwrap">
            <div class="pr-img">
                <div class="hot"></div>
                <a href="urun-<?seo($uruncek["urun_ad"]).'-'.'$uruncek["urun_id"]?>"></a>
                <div class="pricetag on-sale"><div class="inner on-sale"><span class="onsale"><span class="oldprice"><?php echo $uruncek['urun_fiyat']*1.50 ?> TL</span><?php echo $uruncek['urun_fiyat'] ?> TL</span></div></div>
            </div>
            <span class="smalltitle"><a href="product.htm"><?php echo $uruncek['urun_ad'] ?></a></span>
            <span class="smalldesc">Ürün Kodu.: <?php echo $uruncek['urun_id'] ?></span>
        </div>
    </div>
}
</div>

```

Kategoriler sayfamda say değişkenimde eğer durum sıfır ise **Bu kategoride ürün bulunamadı** hatasını veriyor. Eğer daha fazla varsa da **while döngüsü** ile veritabanında ki işlemleri çekmeye yarıyor. Şimdi burada önemli bir kısım var.

```

<a href="urun-<?seo($uruncek["urun_ad"]).'-'.'$uruncek["urun_id"]?>">
</a>

```

İşlemleri çağırıyor ama nasıl geliyor? O da yukarda eklemiş olduğumuz **sef** değişkeninden geliyor. **.htaccess** yapısı devreye geliyor ve aşağıdaki resimden daha yakından gördüğümüz kısım resim linkine tıkladığımızda bize doğru yere gitmemizi sağlıyor.

```

RewriteRule ^sayfa-([0-9a-zA-Z-_-]+) menu-detay.php?sef=$1 [L,QSA]
RewriteRule ^kategori-([0-9a-zA-Z-_-]+) kategoriler.php?sef=$1 [L,QSA]
RewriteRule ^urun-([0-9a-zA-Z-_-]+)-([0-9]+)$ urun-detay.php?sef=$1&urun_id=$2 [L,QSA]

```

Yukarıda da **.htaccess** yapım bulunmakta daha önceden aynı olayları **menu-detay.php** de menüleri listeletirken yapmıştık. Şimdi kategorileri çekerken yaptık. En alttaki de ürünleri çekerken kullanıyoruz. İlerleyen kısımlarda o kısımları da anlattım.



İşlemlerimizin genel görüntüsü bu şekildedir. Yukarıdaki resimde Bilgisayar Parçaları kategorisine tıklandıktan sonra yönetim panelinden o kategori ile ilgili ürünler girildiğinde karşımıza çıkmamızı sağlayan ürünler ekrana yazdırılmıştır.

Ürün Detay sayfasına geçiyorum. Bunun için temamda ki **productions.html** sayfasını buluyorum ve Türkçeleştirme gerçekleştiriyorum. Bu Türkçeleştirmeden daha sonra gerekli kod parçalarını yerleştiriyorum. 4'e bölüp diğer yerlerini daha önceden yaptığımız **include** işlemini yapıyorum. Daha sonra çekeceğimiz ürün bilgileri için gerekli işlemleri yapıyorum.

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

$urunson=$db->prepare("SELECT * FROM urun where urun_id=:urun_id");
$urunson->execute(array(
    'urun_id' => $_GET['urun_id']
));

$uruncek=$urunson->fetch(PDO::FETCH_ASSOC);

$say=$urunson->rowCount();

if ($say==0) {
    header("Location:index.php?durum=yavsama");
    exit;
}
?>
```

Burada veritabanından ürün ile ilgili bilgileri id yardımıyla çekiyoruz ve **PDO::FETCH** sayesinde de yazdırıyoruz. Onun altında say değişkeni var. Bu say değişkeni bizim sitemizden veri çekmesinler veya uğraşmasınlar için eklediğim basit bir method. Aşağıda da o kodun açıklaması bulunmaktadır.

```

<div class="col-md-6 det-desc">
  <div class="productdata">
    <div class="infospan">Ürün Adı: <span><?php echo $uruncek['urun_ad']; ?></span></div>
    <div class="infospan">Markası: <span><?php echo $uruncek['urun_marka']; ?></span></div>
    <div class="infospan">Ürün Kodu: <span><?php echo $uruncek['urun_id']; ?></span></div>
    <div class="infospan">Ürün Fiyatı: <span><?php echo $uruncek['urun_fiyat']; ?></span></div>
  </div>
</div>

```

Klasik, isim-marka-kod-fiyat gibi bilgilerin karşıdan çekilme işlemlerini gerçekleştiren kodlar.

```

<div align="center" class="avatoek"><span>
  <?php
  if ($uruncek['urun_stok']==0) {
    echo "Stoklarımızda bu ürün kalmamıştır.";
  }elseif($uruncek['urun_stok']<5){
    echo "Stokta sadece <b>". $uruncek['urun_stok']."</b> tane kaldı. <br> Bitmek üzere, hemen alın!";
  }else{
    echo "Stok adeti: ". $uruncek['urun_stok'];
  }
  ?></span></div>
</div>

```

Veritabanımızda stok kısmını yapmıştık. Yönetici panelinde de olduğuna göre şöyle bir şey yapayım dedim. Yönetici panelinden girilen ürünün stok değeri sıfır ise **Stoklarımızda bu ürün kalmamıştır.** yazısını **5'ten az var ise** bitmek üzere uyarısını ve 5'ten fazla var ise de normal stok adeti yazmasını sağladım.

```

<div id="myTabContent" class="tab-content shop-tab-ct">
  <div align="center" class="tab-pane fade active in" id="desc">
    <p>
      <?php echo $uruncek['urun_detay']; ?>
      <hr>
      <?php
      $say=strlen($uruncek['urun_video']);
      if ($say>0) {?>
        <iframe width="560" height="315" src="https://www.youtube.com/embed/<?php echo $uruncek['urun_video'] ?>"
        frameborder="0" allowfullscreen></iframe>
      <?php } else {
        echo "Bu ürüne video eklenmemiştir.";
      }?>
    </p>
  </div>
</div>

```

Burada ise Ürün açıklama kısmı hakkında birkaç değer döndürdüm. Bunlar örnek vermek gerekirse yönetim paneline sadece **YouTube** kodunu koyarak koyabileceğimiz ürün video kısmı. Eğer video eklenmiş ise video gösterilecek, eklenmemişse bu ürüne video eklenmemiştir uyarısını verecek.



```

<h4>Yorum Yazın</h4>

<?php if (isset($_SESSION['userkullanici_mail'])) {?>
<form role="form">
    <div class="form-group">
        <textarea class="form-control" placeholder="Lütfen yorumunuzu buraya yazınız..." id="text"></textarea>
    </div>
    <button type="submit" class="btn btn-default btn-red btn-sm">Yorumu Gönder</button>
</form>

<?php } else {?>
Yorum yazabilmek için <a style="color:red" href="register">kayıt</a> olmalı yada giriş yapmalısınız...
<?php } ?>

```

Yorumlar kısmı olacak. Orayı şuanda yapıyorum. Yaptığım kısma kadar bahsetmek gerekirse **isset** sayesinde **\$\_SESSION** kontrolü yapıyorum ve eğer kullanıcı varsa Lütfen yorumunuzu yazınız diye bir **textarea** karşısına çıkıyor eğer kullanıcı girişi session ile geri dönmemişse kayıt sayfsına yönlendiriyor. Burası tabi daha olmadığı için bu şekilde kalacak. İlerleyen kısımlar da bu konu hakkında açıklamayı da anlatıyor olacağım.

```

<?php
$kategori_id=$uruncek['kategori_id'];

$urunaltisor=$db->prepare("SELECT * FROM urun where kategori_id=:kategori_id order by rand() limit 3");
$urunaltisor->execute(array(
    'kategori_id' => $kategori_id
));

while($urunaltcek=$urunaltisor->fetch(PDO::FETCH_ASSOC)) {
    ?>

```

En sonunda da **Dikkatinizi Çekebilir** veya benim kullanmış olduğum başlıkla **Benzer Ürünler** kısmı var. Burada öncelikle uruncek ile kategori id döndürdüm ve **klasik** veritabanı bağlantımız ile işlemleri sokuyorum. Veritabanı bağlantısında **rand() ve limit 3** var. Rand() rastgele içeri getirmesi limit 3'de üç tane içerik gelmesini sağlayacak. Daha sonra **while ile işlemlerimi çeviriyorum**. Yani burada yaptığım olay aynı kategoriye sahip ürünlerin 3 tanesini ekrana getirmek.



### Bil.Programcılığı



45.00 ₺

Ürün Adı:	Bil.Programcılığı
Markası:	Microsoft
Ürün Kodu:	10
Ürün Fiyatı:	45.00 ₺

Stokta sadece 1 tane kaldı.  
Bitmek üzere, hemen alın!

Adet  [Sepete Ekle](#)

Ürün Açıklaması

Yorumlar (0)

Lorem Ipsum, dizgi ve baskı endüstrisinde kullanılan mıgır metinlerdir. Lorem Ipsum, adı bilinmeyen bir matbaacının bir hurufat numune kitabı oluşturmak üzere bir yazı galerisini alarak karıştırdığı 1500'lerden beri endüstri standardı sahte metinler olarak kullanılmıştır. Başvüriz ul hovinca vadiğini cüürdürmekle kalmamış, aynı

İşlemlerimiz, genel olarak bu şekilde görünüyor. Çoğunluk olarak bitmiş durumda. Tabi eksik olan alanlar da var. Yönetim paneline resim yükleme ve ekrana yazdırma, yorum kısmının aktif olması ve sepete ekle gibi kısımlar şuan yapmadığım kısımlar. Yavaş yavaş ilerleyen sayfalarda oralarında anlatımını anlatmış olacağım.

### Benzer Ürünler



HOT

99 TL  
60.00 TL

Maliye  
Ürün Kodu.: 11



HOT

67.5 TL  
45.00 TL

Bil.Programcılığı  
Ürün Kodu.: 10



HOT

69 TL  
40.00 TL

İşletme  
Ürün Kodu.: 9

Böyle bir alan yaptım. Açıklamanın tam alt kısmına gelecek şekilde. Burada **Benzer Ürünler** adı altında aynı kategoriye ait 3 tane içeriğin **rasgele** gelmesini işledim ve **MySQL sorgumda LIMIT 3** olarak belirlemem yeterli oldu.

```

<?php

$kategori_id=$uruncek['kategori_id'];

$urunaltisor=$db->prepare("SELECT * FROM urun where kategori_id=:kategori_id order by rand() limit 3");
$urunaltisor->execute(array(
    'kategori_id' => $kategori_id
));

while($urunaltcek=$urunaltisor->fetch(PDO::FETCH_ASSOC)) {

    ?>

```

Bu modülümüzün kodları ise bu şekilde. Ürün tablomuzu oluştururken kategori\_id'yi de çektiğim için ilişkisel veritabanı oluşturmuş oldum. Burada da kategori id'den random 3 tane veri getirmesini istemiş oluyorum.

Daha sonra **index sayfamda** dolaşırken öne çıkarılan ürünler kısmının olduğunu dikkatimi çekiyor ve hemen ona el atıyorum. **ürün** tablomun içine **urun\_onecikar** isminde bir tane sütun oluşturuyorum ve **enum** değeri giriyorum. Ürün öne çıkar değeri **1 olduğuna ekrana yazdıracak 0 olduğunda pasif olacak**. Ürün aktif kısmını yaptığım mantığı aynı olduğundan dolayı sırasıyla yönetim panelinde ki ürün sayfasını, düzenleme sayfasını ve ekleme sayfasına bu girdileri eklemiş oluyorum. Daha sonra islem.php'ye giderek sadece ismini ekliyorum. Ana sayfamda o ilgili kod kısmını buluyorum ve alttaki kod blogunu yazıyorum.

```

<?php

$urunsonor=$db->prepare("SELECT * FROM urun where urun_durum=:urun_durum and urun_onecikar=:urun_onecikar");
$urunsonor->execute(array(
    'urun_durum' => 1,
    'urun_onecikar' => 1
));

while ($uruncek=$urunsonor->fetch(PDO::FETCH_ASSOC)){ ?>

```

Burada klasik veri tabanı bağlantımı yazıyorum ve ürün\_durum ve ürün\_onecikar değişkenimizin değeri 1 ise while döngüsü ile yazdırmaya başlıyorum. İşlem bu kadar basit. Şimdi ana sayfama bakıyorum.

## Öne Çıkan Ürünler



İşletme  
Ürün Kodu: 9

40.00 TL



Bil.Programcılığı  
Ürün Kodu: 10

45.00 TL



Maliye  
Ürün Kodu: 11

60.00 TL

Yaptığım işlemler istediğim gibi sıralanmaya başlamış. Gayet hoş bir görüntüsü oldu. Bu modülümüzün yönetici panelinde ki görüntüsü ise aşağıda ki gibi.

Ürün Ad	Ürün Marka	Ürün Stok	Ürün Fiyat	Ürün Durum	Ürün Öne Çıkar
Maliye	Apple	23	60.00 ₺	Aktif	Evet
Bil.Programcılığı	Microsoft	1	45.00 ₺	Aktif	Evet
İşletme	Hasan	50	40.00 ₺	Aktif	Hayır
Fare	Logitech	42	70.00 ₺	Aktif	Hayır

--tabi bi' tanesi hayır yaptığım zaman çektiğim için iki tane evet butonu var gözükmekte--

## > Yorum İşlemleri

(MySQL Tablo Oluşturma, Yorum Yazma ve Düzenleme)