



Burdur
Mehmet Akif Ersoy
Üniversitesi

MAKÜ|GMYO
GÖLHİSAR MESLEK YÜKSEKOKULU

İnternet Programcılığı I

Öğr. Gör. Hüseyin Şengün
hsengun@mehmetakif.edu.tr

12. Hafta
VERİ TABANI İŞLEMLERİ-4





Kullanıcıdan gelen verilerin veri tabanında kullanılması

- Kullanıcıdan veri girişi istenip bunları veri tabanına kaydetmek önemlidir.
- Bir form oluşturarak kullanıcının bu form sayesinde veri tabanına kayıt işlemi yapmasını istiyorsak gelen verinin filtreden geçirilmeden SQL kodu içerisine eklenmesi sakıncalıdır.
- Çünkü SQL kodlarında metinsel bir ifade “ ’ ” tırnak işaretleri arasında yer almak zorundadır eğer kullanıcının gönderdiği veri içerisinde “ ’ ” tırnak işareti varsa ise SQL sunucumuz metinsel ifadenin bittiğini düşünecek ve “ ’ ” tırnak işaretinden sonraki yazıları SQL kodu olarak çalıştırmayı deneyecek bu da hataya neden olacaktır.



mysql_real_escape_string() fonksiyonu kullanılmadan önce veri tabanı bağlantısı yapılması zorunludur.





Kullanıcıdan gelen verilerin veri tabanında kullanılması

- Hatta art niyetli kullanıcılar bu yazılım açığını kullanarak gönderilen metin içerisine SQL kodları yazıp çalıştırabilirler.
- Bu bir çeşit hack yöntemidir adı ise **injection** dır.
- Bunu engellemek için “**GET**” veya “**POST**” yöntemi ile gelen veri **mysqli_real_escape_string()** fonksiyonunun süzgecinden geçirilmesi gerekir.
- Bu fonksiyonun kullanılması için veri tabanı bağlantısı yapılması zorunludur.

- **Kullanımı:** `mysqli_real_escape_string(deger);`





Kullanıcıdan gelen verilerin veri tabanında kullanılması

- `$temiz_veri=mysqli_real_escape_string($_POST[" veri "]);`
- Şimdiye kadar yaptığımız işlemlerin hepsinde veri tabanına gönderdiğimiz bütün değerler sabit değerlerdir.
- Şimdi kısa bir örnekle form oluşturup formdan gelen verileri kaydedelim.





Kullanıcıdan gelen verilerin veri tabanında kullanılması

- İlk önce html kodları ile kullanıcının duyuru başlığı ve içeriği girebileceği basit bir form oluşturalım.

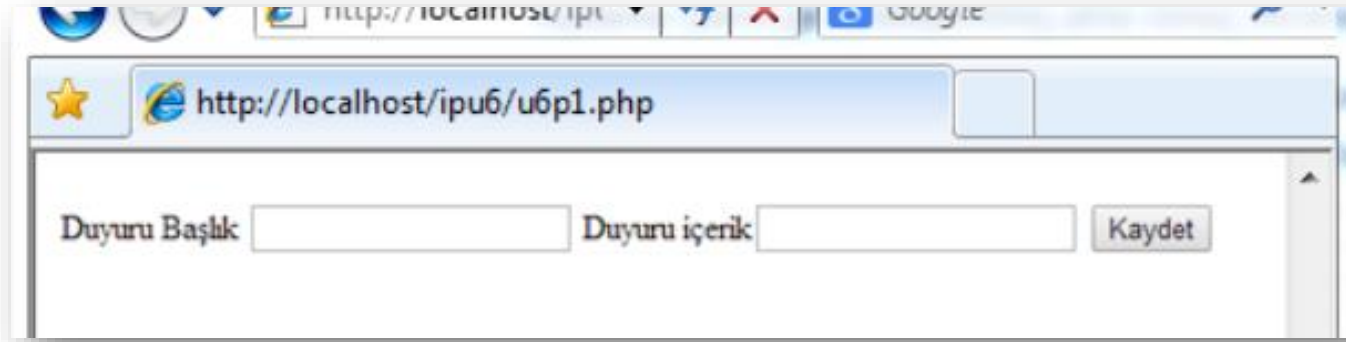
```
9 <form action="kaydet.php" method="post"/>
10     Duyuru Başlık <input name="baslik" type="text">
11     Duyuru içerik <input name="icerik" type="text">
12     <input value="Kaydet" type="submit"/>
13 </form>
```

- Yukardaki örnekte basit bir kayıt formu oluşturduk. Bu dosyamıza **“form.html”** adını vererek kaydettik. Formda “Kaydet” butonuna tıklandığında girilen “baslik” ve “icerik” değerlerini “post” metodu ile **“kaydet.php”** dosyasına göndermesi için gerekli ayarlamaları yaptık.





Kullanıcıdan gelen verilerin veri tabanında kullanılması



- Daha önceden html kodları ile oluşturduğumuz formun ekran görüntüsü yukarıdaki gibidir.
- Kaydet butonuna bastığımızda tarayıcı “kaydet.php” sayfasına gitmeye çalışacak ancak henüz o sayfayı oluşturmadığımız için hata verecektir.
- Şimdi “[kaydet.php](#)” sayfasını oluşturalım.





Kullanıcıdan gelen verilerin veri tabanında kullanılması

```
<?php
$baglanti = mysqli_connect("Sunucu Adresi","Kullanıcı Adı"," Şifre","Veritabanı");
$sorgu=mysqli_query($baglanti, "INSERT INTO duyurular (baslik,icerik) VALUES
('".$_$_POST["baslik"]."', '".$_$_POST["icerik"]."'");
```

```
if($sorgu ==true)
{
echo("Kayıt eklendi!");
}
else
{
echo("Kayıt Eklenemedi!");
}
?>
```

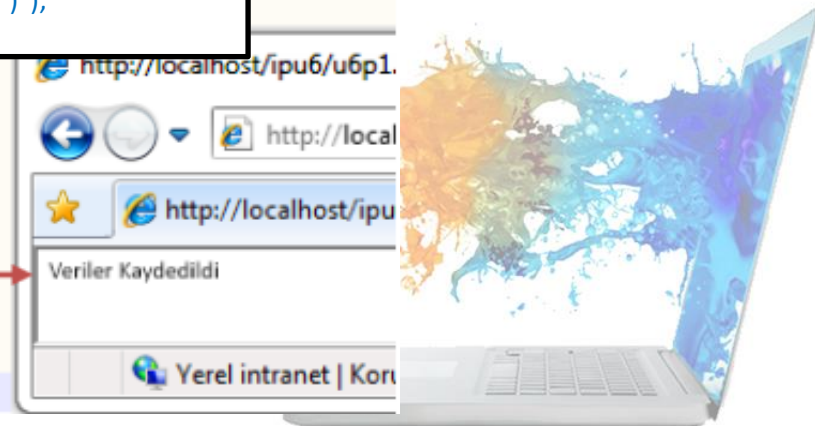
('".\$_\$_POST["baslik"]."', '".\$_\$_POST["icerik"]."');



Kullanıcıdan gelen verilerin veri tabanında kullanılması

- Gelen verilerin yani “**post**” veya “**get**” verileri eğer sorgumuzda yer alacaksa önce `mysqli_real_escape_string()` fonksiyonundan geçirilmesi gerekir.

```
2 <?php
3
4 $baglanti = mysqli_connect("Sunucu Adresi","Kullanıcı Adı","Şifre","Veritabanı");
5
6 $baslik = mysqli_real_escape_string($_POST["baslik"]);
7 $icerik = mysqli_real_escape_string($_POST["icerik"]);
8
9
10 $sorgu=mysqli_query($baglanti, "INSERT INTO duyurular (baslik,icerik) VALUES
11 (" .$_POST["baslik"]."," .$_POST["icerik"].")");
12
13 if($sql == true)
14 {
15     echo "Veriler Kaydedildi";
16 }
17 else
18 {
19     echo "Veriler kaydedilemedi";
20 }
21 ?>
```





Kullanıcıdan gelen verilerin veri tabanında kullanılması

- Yukardaki iki örnekte de aynı işlemler gerçekleştirilmiştir.
- Ancak ilk örnekte “post” metodu ile gelen veri direk olarak sorgu içerisine yazılmış, ikinci örnekte ise `mysqli_real_escape_string()` fonksiyonundan geçirilmiştir.



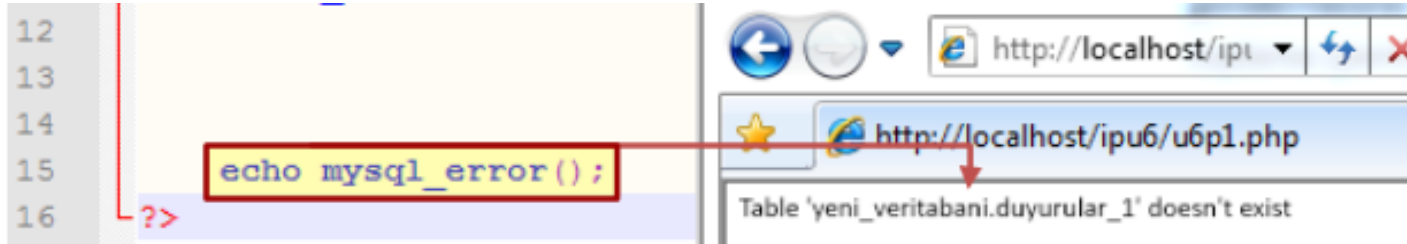
HATA KONTROLÜ

- Yazdığımız sorguların tek bir harfi yanlış olması durumunda bile sorgumuz çalışmayabilir. Bizim için bu hatayı bulmak çok zor olabilir. Eğer sorgumuzda bir hata varsa bunu bulmak için **mysqli_error()** veya **mysqli_errno()** fonksiyonlarını kullanılır.
- Ancak mysqli_error() fonksiyonunun cevabı İngilizcedir.

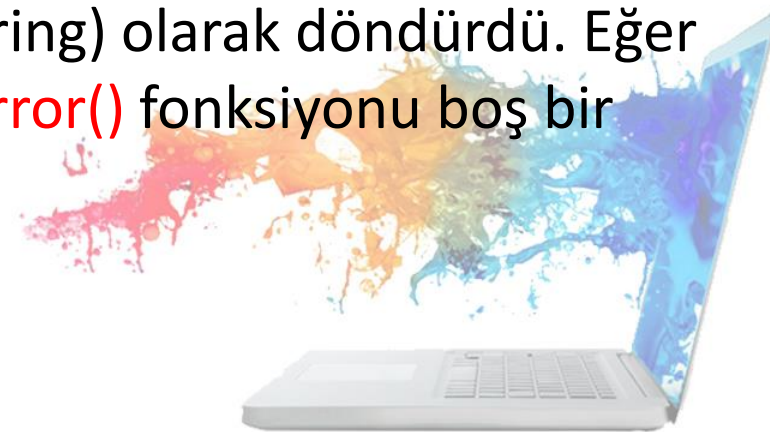


HATA KONTROLÜ

```
12  
13  
14  
15 echo mysql_error();  
16 ?>
```

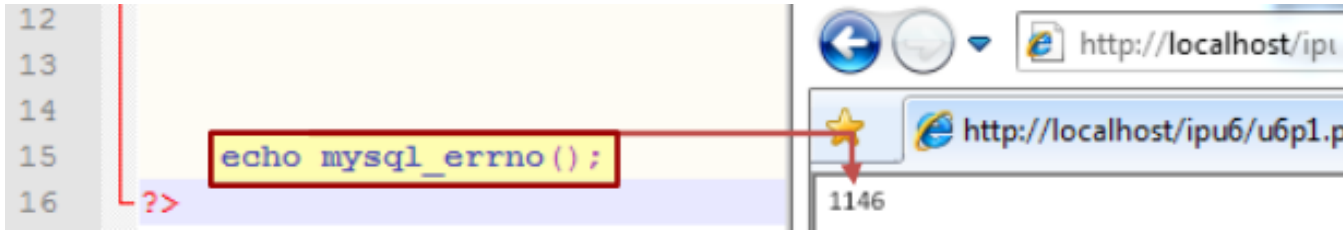


- Yukardaki örnekte veri tabanımızda olmayan “duyurular_1” adlı tablodan veri silmek istedik. Ancak **mysql_error()** fonksiyonu bize bu tablonun olmadığını metin (string) olarak döndürdü. Eğer sorgumuzda hata olmasaydı **mysql_error()** fonksiyonu boş bir metin (string) döndürecekti.

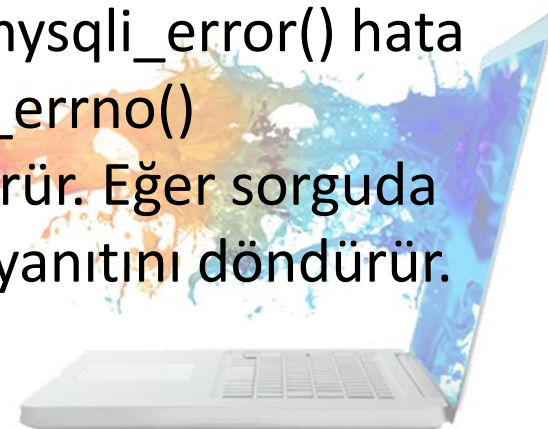


HATA KONTROLÜ

```
12  
13  
14  
15 echo mysql_errno();  
16 ?>
```



- Yukardaki örnekte de işlemler gerçekleştirilmiş ancak **mysqli_error()** fonksiyonu yerine **mysqli_errno()** fonksiyonu kullanılmıştır. Bu iki fonksiyon arasındaki fark **mysqli_error()** hata varsa metin (string) bir ifade döndürür, **mysqli_errno()** fonksiyonu ise sayısal (integer) bir ifade döndürür. Eğer sorguda hata yoksa **mysqli_errno()** fonksiyonu sıfır "0" yanıtını döndürür.



Bağlantının kapatılması

- Veri tabanı bağlantısının kapatılma işlemi **mysqli_close()** fonksiyonu ile yapılmaktadır.
- İşlemimiz bittikten sonra bu fonksiyon kullanılır.
- Bu fonksiyonun kullanımı PHP'de zorunlu bir işlem değildir çünkü eğer bağlantıyı kapatmazsan belirli bir süre işlem yapmadığımızı anlayan mysql bağlantıyı otomatik olarak kapatacaktır.
- Ancak yapılan site çok sık ziyaret ediliyorsa veri tabanı bağlantıları çok fazla olacağı için mysqli_close() fonksiyonu kullanarak bağlantıların kapatılması tavsiye edilir.





Bağlantının kapatılması

```
7 | mysql_close();  
8 | ?>
```

- Yukardaki örnekte önce “yeni_veritabanı” adındaki veri tabanına bağlantı yapılmış ardından duyurular tablosundaki bütün veriler silinmiş ardından veri tabanı bağlantısı kesilmiştir.



Uygulama Ödevi

- Kullanıcı kayıt sistemi için VeriTabanı ve tablo oluşturun.
- Kullanıcılar Web sayfasından üye olması için formu tasarlayıp, kayıt ol butonuna tıkladığında, üye bilgilerini veri tabanına kaydeden php kodlarını yazınız?
- Üye girişi için form tasarlayın ve kullanıcı_adi, şifre ile giriş sağlansın.



Kaynaklar

- PHP Kılavuzu - <http://php.net/manual/tr/index.php>
- PHP Tutorials - <https://www.w3schools.com/php/default.asp>
- PHP Videolu Temel Dersler İçin: Youtube/Php Türkiye
- <https://php-mysql.org/>
- Atatürk Üniversitesi Açık Öğretim Fakültesi

