WEB TABANLI PROGRAMLAMA

BÖLÜM 15 PHP-MYSQL İLE ÖRNEK PROJE KULLANICI YETKİLENDİRMELİ PERSONEL OTOMASYONU

Prof. Dr. Turgay Tugay Bilgin turgay.bilgin@btu.edu.tr



GENEL BAKIŞ...

15.1) MySQL Veritabanı Oluşturulması

15.2) MySQL Veritabanına Bağlantı

15.3) Kullanıcı Kayıt Uygulaması

15.4) Kullanıcı Girişi (Login) Uygulaması

15.5) Yetkilendirme Sistemi ile Personel Otomasyonu Entegrasyonu15.6) Özet

- Bu ünitede PHP ve MySQL kullanarak kullanıcı yetkilendirme özelliğine sahip bir Personel Otomasyonu oluşturulacaktır. Uygulamanın kodlaması sırasında aşağıdaki adımlar izlenecektir:
 - 1) Kullanıcı adı, Şifre ve E-posta bilgisini tutacak bir MySQL veritabanı oluşturulacaktır.
 - 2) MySQL veritabanına bağlantı gerçekleştirilecektir.
 - 3) PHP ile bir kullanıcı kayıt uygulaması oluşturulacaktır. Bu uygulama ile üye kaydı yapılacaktır. Diğer bir ifade ile, kullanıcı kendini veritabanına kaydedecektir.
 - 4) PHP ile bir login uygulaması geliştirilecektir. Bu uygulama ile kullanıcıdan alınan kullanıcı adı ve şifre MySQL veritabanında kontrol edilecek, eğer bilgiler doğru ise kullanıcı kendi özel ana sayfasına yönlendirilecektir.
 - 5) İlk üç maddede geliştirilen yetkilendirme sistemi 14. Ünitede geliştirilen Personel Otomasyonuna entegre edilecektir.

(15.1) MYSQL VERİTABANI OLUŞTURULMASI

• Yetkilendirme sistemi için gereken veritabanı ve tabloyu oluşturacak SQL komutları aşağıda verilmiştir.

```
CREATE DATABASE `sifreligiris` DEFAULT CHARACTER
SET utf8 COLLATE utf8_turkish_ci;
USE `sifreligiris`;
CREATE TABLE `kullanicilar` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `kullaniciadi` varchar(255) NOT NULL,
  `eposta` varchar(255) NOT NULL,
  `sifre` varchar(255) NOT NULL,
  PRIMARY KEY (`id`)
);
```

Kod 15.1. Yetkilendirme sistemi için SQL komutları

 Örnekteki SQL kodlarını Bölüm 14.5'de anlatıldığı şekilde MySQL konsoluna giriş yaparak veya phpMyAdmin ekranında SQL sekmesini tıklayarak ekrana gelen metin kutusuna yazabilir ve en alt sağ köşedeki "git" butonunu tıklayarak çalıştırabilirsiniz.

(15.2) MYSQL VERİTABANINA BAĞLANTI

• Aşağıdaki kodları "_mysqlbaglan.php" adıyla "c:\wamp\www" dizinine kaydediniz.

```
Dikkat G Önceki bölümlerde oluşturduğunuz diğer dosyalar ile karışmaması için bu bölümde oluşturulan tüm dosyaların başına alt çizgi (_) işareti eklenmiştir. Siz de bu şekilde kaydediniz.
```

```
<?php
$server = 'localhost';
$user = 'root';
$password = '';
$database = 'sifreligiris';
$baglanti = mysqli_connect($server,$user,$password,$database);

if (!$baglanti) {
    echo "MySQL sunucu ile baglanti kurulamadi! </br>';
    echo "HATA: " . mysqli_connect_error();
    exit;
}
```

(15.3) KULLANICI KAYIT UYGULAMASI

• Aşağıdaki kodları "_**register.php**" adıyla "c:\wamp\www" dizinine kaydediniz.

```
<?php
require (' mysqlbaglan.php');
if (isset($ POST['username']) && isset($ POST['password'])){
extract($ POST);
// sifre metni SHA256 ile şifreleniyor.
$password = hash('sha256', $password);
$sql="INSERT INTO `kullanicilar` (kullaniciadi, sifre,
eposta)";
$sql = $sql . "VALUES ('$username', '$password',
'$email')";
    $cevap = mysqli query($baglanti, $sql);
    if ($cevap) {
        $mesaj = "<h1>Kullanıcı oluşturuldu.</h1>";
    }else{
        $mesaj = "<h1>Kullanıcı oluşturulamadı!</h1>";
}
?>
```

```
→ <html>
```

</body> </html>

```
<!-- türkçe karakter desteği ayarı -->
<meta http-equiv="Content-Type" content="text/html;</pre>
charset=UTF-8" />
<body>
<?php
//mesaj varsa ekrana yazdır.
if (isset($mesaj)) echo $mesaj; ?>
<h2>Kayıt Formu</h2>
<form action="<?php $ PHP SELF ?>" method="POST">
    Kullanıcı Adı:
    <input type="text" name="username"><br />
    E-posta:
    <input type="email" name="email"><br />
    Sifre:
    <input type="password" name="password"><br />
    <input type="submit" value="Kaydet"/><br /><br />
    <a href="login.php">[Kullanıcı Girişi]</a>
</form>
```

• Önceki slaytta verilen kodu, web tarayıcınızda http://localhost/_register.php yazarak çalıştırdığınızda karşınıza üye kayıt formu gelecektir. Bu formu doldurarak bir kullanıcı oluşturunuz. Formu gönderdikten sonra phpMyAdmin uygulaması üzerinden "*sifreligiris*" adlı veritabanındaki "*kullanıcılar*" tablosuna oluşturduğunuz kaydın eklendiğini kontrol ediniz.

Bilgilendirme & phpMyAdmin ile veritabanına baktığınızda kullanıcının parolasının açık olarak okunabilir durumda olmadığını göreceksiniz. Parolaları açık olarak veritabanına kaydetmek güvenlik açığı doğmasına sebep olur. Çünkü, böyle bir durumda herhangi bir veritabanı yöneticisi tüm kullanıcıların parolalarını görebilir. Bu yüzden, bu örnekte kullanıcıdan alınan paşsword bilgisi SHA256 algoritması ile şifrelenerek veritabanına kaydedilmektedir. Bu sayede parolalar şifrelenmiş olarak depolandığından veritabanı yüneticisi tarafından öğrenilemeyecektir.

```
(15.4) KULLANICI GIRIŞI (LOGIN)
UYGULAMASI
• Kod 15.4'ü "_login.php" adıyla "c:\wamp\www" dizinine kaydediniz.
<?php
session_start();
require('_mysqlbaglan.php');
```

```
if (isset($_POST['username']) and
isset($_POST['password'])) {
  extract($_POST);
```

```
// sifre metni SHA256 ile şifreleniyor.
$password = hash('sha256', $password);
$sql = "SELECT * FROM `kullanicilar` WHERE ";
$sql= $sql . "kullaniciadi='$username' and
sifre='$password'";
```

```
$cevap = mysqli_query($baglanti, $sql);
//eger cevap FALSE ise hata yazdiriyoruz.
if(!$cevap){
    echo '<br>Hata:' . mysqli_error($baglanti);
}
```

```
//veritabanindan dönen satır sayısını bul
$say = mysqli num rows($cevap);
if ($say == 1) {
    $ SESSION['username'] = $username;
 }else{
$mesaj = "<h1> Hatalı Kullanıcı adı veya Şifre!</h1>";
if (isset($ SESSION['username'])) {
header("Location: uyesayfasi.php");
}else{
//oturum yok ise login formu görüntüle
?>
<html>
<meta http-equiv="Content-Type" content="text/html;</pre>
charset=UTF-8" />
<body>
<form action="<?php $ PHP SELF ?>" method="POST">
<?php
if(isset($mesaj)) { echo $mesaj;}
?>
Kullanıcı Adı:
<input type="text" name="username"><br />
Sifre:
<input type="password" name="password" ><br /><br />
<input type="submit" value="GiRis"/><br /><br />
<a href=" register.php">[Kayıt Ol]</a>
</form>
</body>
                                                8
</html>
<?php } ?>
```

Kod 15.4. Yetkilendirme sistemi

Bilgilendirme *Geometrica Bu formda da kullanıcıdan alınan parola metni SHA256 ile şifrelenmektedir. Çünkü veritabanında şifrelenmiş parola tutulduğundan kullanıcıdan alına parola bilgisi de şifrelenerek veritabanındaki parola ile karşılaştırılmalıdır.*

• Kullanıcı adı ve şifre doğru ise kullanıcıyı kendi özel sayfasına yönlendireceğiz. Bu işlem yukarıdaki örnek projede

header("Location: _uyesayfasi.php");

satırı ile gerçekleştirilmiştir. Şimdi, "**_uyesayfasi.php**" dosyasının içeriğini oluşturalım. Aşağıdaki kodları "**_uyesayfasi.php**" adıyla "c:\wamp\www" dizinine kaydediniz.

Kod 15.5. Üye sayfası

```
<?php
//oturumu başlat
session start();
//eğer username adlı oturum değişkeni yok ise
//login sayfasına yönlendir
if ( !isset($ SESSION['username']) ) {
 header("Location: login.php");
  exit();
?>
<html>
<meta http-equiv="Content-Type"
content="text/html; charset=UTF-8" />
 Merhaba <? echo $ SESSION['username'] ?>, <br />
Özel sayfanıza hoş geldiniz.<br /><br />
<a href=' logout.php'>[Oturumu Kapat]</a>
</html>
```

10

• Kullanıcı giriş uygulamasını web tarayıcınızda http://localhost/_login.php yazarak çalıştırdığınızda, kullanıcı kayıt formunda oluşturduğunuz kullanıcı adı ve şifre ile giriş yapabilirsiniz. Eğer "Turgay" adlı bir kullanıcı ile giriş yaparsanız Şekil 15.1'deki gibi bir üye sayfası görüntülenecektir.



Şekil 15.1 Üye sayfası

• Yetkili kullanıcı işini bitirdiğinde oturumu sonlandırması gerekir. Bunun için Kod 15.5'de verilen örnekte "**logout.php**" dosyasına bir link oluşturulmuştur. Bu dosyanın içinde sadece oturumu sonlandırmayı ve kullanıcıyı login sayfasına yönlendirmeyi sağlayan komutlar bulunur. Dosya içeriği aşağıda verilmiştir. Bunu da "c:\wamp\www" dizinine kaydediniz.

Kod 15.6. Oturumu sonlandırma



• Bu aşamadan itibaren kullanıcı yetkilendirmesi tamamlanmış durumdadır. Yetkili kullanıcının gerçekleştirebileceği tüm işlemleri "**_uyesayfasi.php**" dosyası içinde oluşturabilirsiniz.

(15.5) YETKİLENDİRME SİSTEMİ İLE PERSONEL OTOMASYONU **ENTEGRASYONU**

Yetkilendirme sistemi modüler olarak tasarlandığından, personel otomasyonu ile entegrasyon son derece basittir. Tek yapmanız gereken, • Personel otomasyonunda hangi PHP dosyasına parola ile erişim isteniyorsa o PHP dosyasının *ilk satırına* aşağıdaki kod parçasını eklemektir.

```
<?php
//oturumu başlat
session start();
//eğer username adlı oturum değişkeni yok ise
//login sayfasına yönlendir
if ( !isset($ SESSION['username']) ) {
  header("Location: login.php");
 exit();
?>
```

Kod 15.7. Yetkilendirme işlemi için kod parçası

Kod 15.7'de öncelikle oturum başlatılıyor, eğer kullanıcı daha önce login olmuş ise username adlı bir oturum değişkeni olmalıdır. Eğer bu değişken yok ise kullanıcı "login.php" dosyasına yönlendiriliyor. Bu sayede kullanıcı yetkilendirme yapmadan ilgili PHP dosyasını çalıştıramıyor.

Kod 15.7'yi Bölüm 14.7'de oluşturduğunuz "kayitformu.php" dosyasının ilk satırına ekleyiniz. Personel otomasyonudaki diğer PHP dosyaların da hepsinin başına bu kod parçasını ekleyebilirsiniz. Böylece, yetkisiz kullanıcı hiçbirini çalıştıramayacaktır. 13

• Son olarak Şekil 15.1'de görülen üye sayfasına "Personel Otomasyonu" linki koyarsanız kullanıcıların otomasyona erişimi çok daha kolay olacaktır. Bunun için Kod 15.5'de verilen üye sayfasını aşağıdaki gibi tekrar düzenleyebilirsiniz.

```
<?php
//oturumu baslat
session start();
//eğer username adlı oturum değişkeni yok ise
//login sayfasına yönlendir
if ( !isset($ SESSION['username']) ) {
 header("Location: login.php");
 exit();
?>
<html>
<meta http-equiv="Content-Type"
content="text/html; charset=UTF-8" />
Merhaba <? echo $ SESSION['username'] ?>, <br />
Özel sayfanıza hoş geldiniz.<br /><br />
<a href="kayitformu.php">Personel Otomasyonu</a>
<br /><br />
<a href=' logout.php'>[Oturumu Kapat]</a>
</html>
```

```
Kod 15.8. Personel otomasyonu linki bulunan üye sayfası
```

• Üye sayfasını bu değişiklik sonrası çalıştırdığınızda Şekil 15.2'deki gibi bir görüntü oluşacaktır.



Şekil 15.2 Personel otomasyonu linki bulunan üye sayfası

• Uygulama projesi kodlarını kısa tutabilmek adına tasarımda CSS öğeleri kullanılmamıştır. Siz, kendi zevkinize göre CSS dosyaları oluşturarak bu kodlara ekleyebilirsiniz. Böylece estetik açıdan daha gösterişli ve ergonomik tasarımlar elde edebilirsiniz.

(15.6) ÖZET

• Bu ünitede okuyucuların kitapta öğrendikleri kavramları pekiştirmeleri amacıyla örnek bir proje senaryosu oluşturulmuş ve veritabanı oluşturma adımından başlayarak kodlama işlemleri sırasıyla açıklanmıştır. Bu uygulama projesinin kaynak kodları yazarın internet sayfasından indirilebilir.