



T.C.

CUMHURİYET ÜNİVERSİTESİ

MÜHENDİSLİK FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

# FULL-STACK WEB DEVELOPER UYGULAMA GELİŞTİRME

RECEP KARADEMİR

2015141003

## Özet:

Hazırladığım web uygulaması front-end ve back-end geliştirme süreçlerini içerir.

**Geliştirdiğim uygulamayı yaklaşık ~70 saatte hazırladım.**

Uygulama geliştirmeye başlamadan önce CSS, PHP ve HTML bilgim sıfırdı fakat proje bittikten sonra temel bilgiler ve plan doğrultusunda proje tamamlayabilme yetisi kazandım.

Uygulamamda Front-End olarak CSS tercih ettim. Database kısmında MySql

Back-End olarak PHP(Html işaretleme dili de kullanıldı) ve server tarafında Apache kullandım.

Projede kullanılan araçlar :GIT, Atom, Notepad++,Gitedesktop,XAMPP,WAMPSEVER

[D]URUM [A]DET [K]ONUM Projesi

Site; giriş yapma, kayıt olma, eşya ekleme, eşya silme, eşya güncelleme ve eşya listeleme yapabilmektedir.

Bu proje evde veya iş yerinde var olan fakat yeri, adedi ve durumu bilinmeyen eşyaların kontrol altında tutulabilmesi için geliştirilmiştir. Kullanıcı siteyle, kayıt olarak veya giriş yaparak etkileşime geçecektir.

Kullanıcı sahip olduğu eşyaların durumlarını, siteye giriş yaparak veri tabanına kaydedebilmektedir.

Kayıtlı kullanıcı eşya kontrol panelinden eşyalarla ilgili istediği işlemi yapabilir.

## PROJE GITHUB LINK:

**<https://github.com/recepkarademir/Computer-Engineering-Project-2>**

**Anahtar Kelimeler:** Full-stack web developer, front-end, back-end, web uygulaması, veritabanı etkileşimi,

CRUD, servis tabanlı mimari, DAK

**Keywords:** Full-stack web developer, front-end, back-end, create-readupdate-delete , web application, service-oriented-architecture, DAK

# GİRİŞ

Bu proje evde veya iş yerinde var olan fakat yeri, adedi ve durumu bilinmeyen eşyaların kontrol altında tutulması için geliştirilmiştir.

Bazen, ihtiyacımız olan eşyaya sahip olduğumuzu biliriz fakat eşyanın yerini, durumunu ve adedini unuturuz.

Bu ve buna benzer durumlar eşyanın yeri, durumu ve adedi güncel olarak tutularak çözülebilir.

Örneğin, bir depoda eşyaların üzerlerine yerleştirilen etiket ile eşyanın bilgileri ve anlık hareketi etiket okuyucuları tarafından algılanarak bu sisteme otomatik girilirse, eşyaların durumu sistemde güncel olarak kontrol edilebilir.

Hatta, eşya temin edildiği an bir kez sisteme kaydedilmesi bile bazen yeterli olacaktır.

Yani kullanılan eşya alındığı yere bırakıldığında veya kullanılıp atık olduğunda, ikinci kez kullanılmak istenilirse sistemdeki son konumuna bakıp eşyanın bittiği veya kullanılmış olduğu sistem güncellenmeden ve olası diğer yerlere bakılmadan bile bilinebilir.

Sistemin tam olarak sorunsuz çalışması için etkileşime geçilen eşyayla ilgili güncelleme yapılmalıdır.

Güncellemeler yapılmasa dahi, eşya temin edildiğinde sisteme girilmişse, eşyanın son konumuna bakarak eşyanın akıbeti öğrenilebilir.

# UYGULAMA

Php, html ve css kullanılarak hazırlanan site, eşya durum kontrolü yetkisini giriş yapan kullanıcıya verir.

Giriş yapan kullanıcı etkileşime geçtiği eşyanın son durumunu sisteme girmelidir.

Eşyaların durumu ve konumu böylelikle kontrol altında tutulmuş olur.

## Proje hazırlama aşaması:

Genel hatlarla ne yapılacak ve nasıl bir güzergahla yapılacak belirlenmeli.

Aşağıdaki programların bazıları kullanılarak istenilen proje hazırlanacaktır.



1)AtomSetup-x64.exe



2)npp.7.5.3.Installer.x64.exe



3)Sublime Text Build 3143 x64 Setup.exe



4)GitHubDesktop Setup.exe



5)wampserver3.1.0\_x64.exe



6)xampp-win32-7.1.10-0-VC14-installer.exe

Atom, Notepad++, Sublime Text veya başka kod editörü(Dreamweaver vs.) ile yazacağınız kodları düzenleyebilirsiniz.

Ben aktif olarak Notepad++ kullanacağım.

Projemizde sunucu ve veritabanı kullanılacağından WampServer veya Xampp gereklidir.

WampServer ve Xampp programları Apache, MySQL ve diğer bileşenleri içerdiğinden WampServer veya Xampp kurmanız işinizi kolaylaştıracaktır.

Siteye giriş yapabilmek için Wampserver veya Xampp gibi Apache ve MySQL barındıran bir program kullanmanız gerekir.

İsterseniz Apache'yle MySQL'i ayrı olarak da kurabilirsiniz. Projemde WampServer kullanacağım.

İsterseniz Xampp da kullanabilirsiniz(WampServer ve Xampp aynı anda kullanamazsınız).

Xampp için C:\wamp64\www\ yerine C:\xampp\htdocs\ konumunu proje klasörü olarak kullanmalısınız.

İleride kullanacağımız phpmyadmin erişimi için ise XAMPP Control Panel'inden Apache ve MySQL **start** konuma getirilmelidir.

(Görev çubuğundan Xampp'e sağ tıklayarak da Apache ve MySQL start konuma getirebilirsiniz.)

Adres kısmına localhost/phpmyadmin/ yazdığınızda phpmyadmin çalışacaktır.

# UYGULAMA(DEVAM)

Github'ı versiyon kontrol sistemi aracı olarak tercih edebilir ve kodunuzda yaptığınız her değişikliği görebilirsiniz. Ben projemde GithubDesktop aracını kullanıyorum.

## //GithubDesktop Temel Kullanımı

GithubDesktop'ı bilgisayarınıza kurun ve github.com'dan Github hesabı açıp yeni bir repository oluşturun.

GithubDesktop uygulamasında File sekmesinden Clone repository'e tıklayın. Github hesabınızla giriş yaptıktan sonra oluşturduğunuz repository i seçin.

Local Path kısmından C:\wamp64\www\repositoryadı olacak şekilde seçin ve klonlayın.

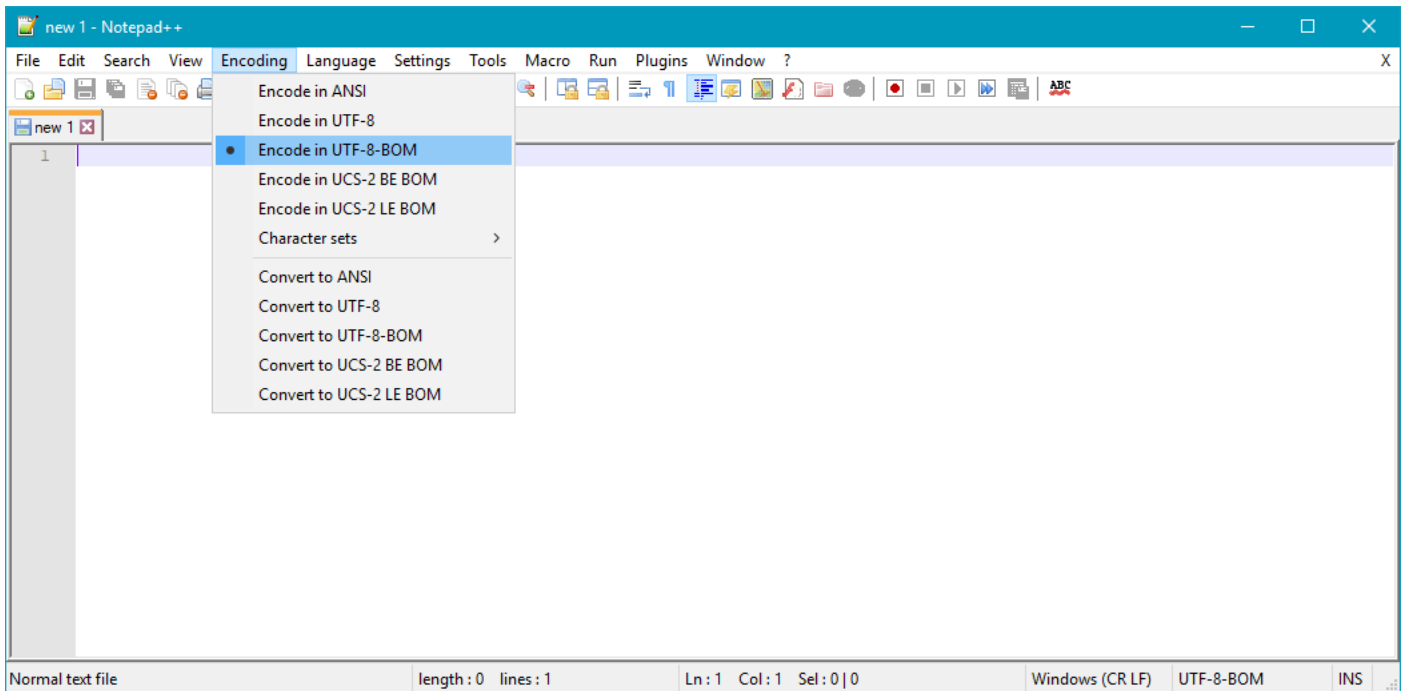
Artık www klasörü içerisindeki klasörde yapılacak her değişiklik GithubDesktop tarafından algılanacak ve changes kısmından commit ettiğinizde kaydedilecektir. Bazen değişiklikleri Push yapmayı unutmayın.

Push origin yazan yere tıkladığınızda değişiklikler Github hesabınıza yüklenecektir.

Hazırlayacağımız siteyi test etmek için bir internet tarayıcınızın olması gerekmektedir.

Google Chrome, Opera, Iexplorer, Safari, Mozilla, Vivaldi, Edge, YandexBrowser vs.

Türkçe karakter sorunu aşağıdaki ayarı Notepad++ a uygulansınlar.

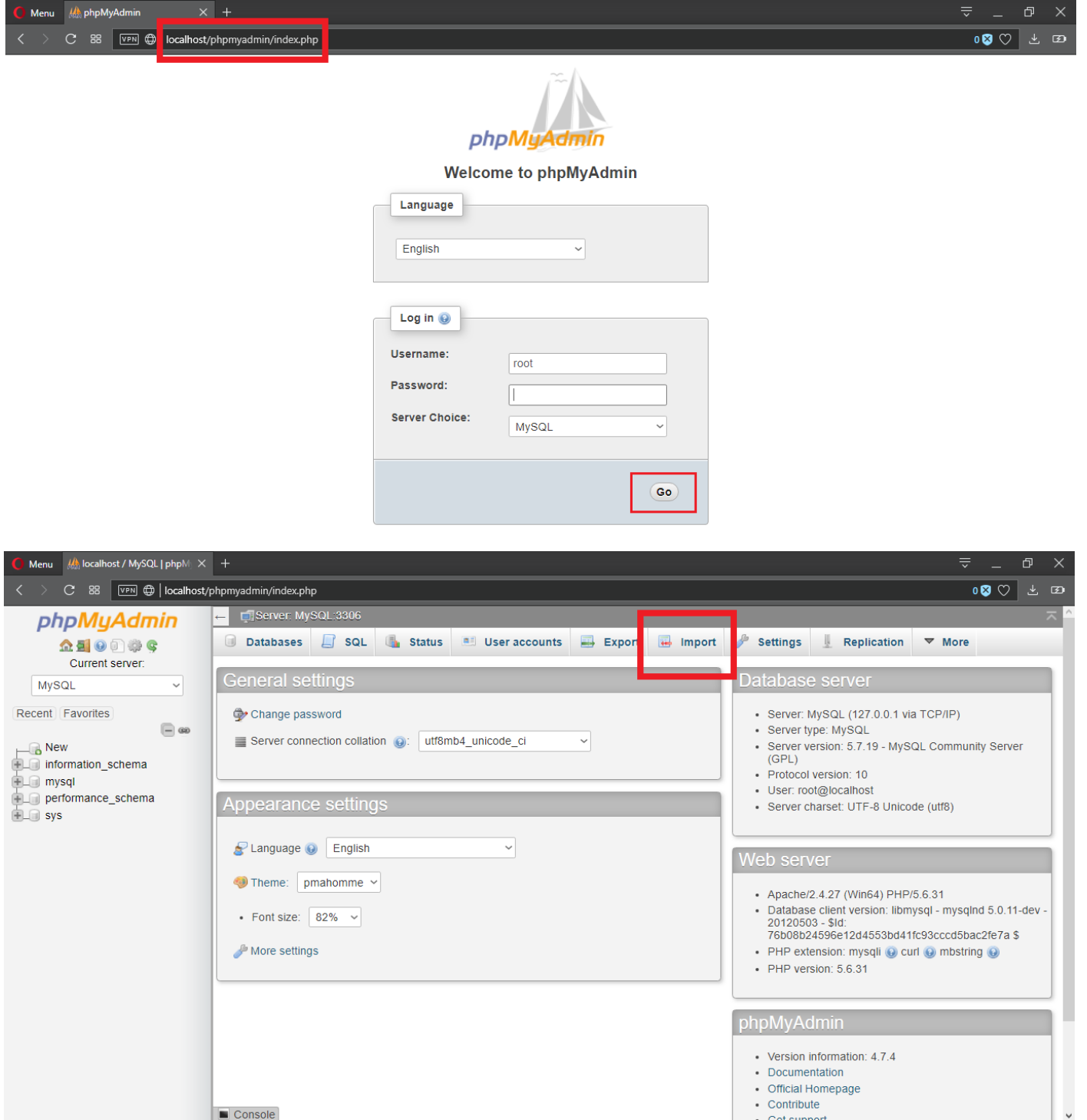


# UYGULAMA(DEVAM)

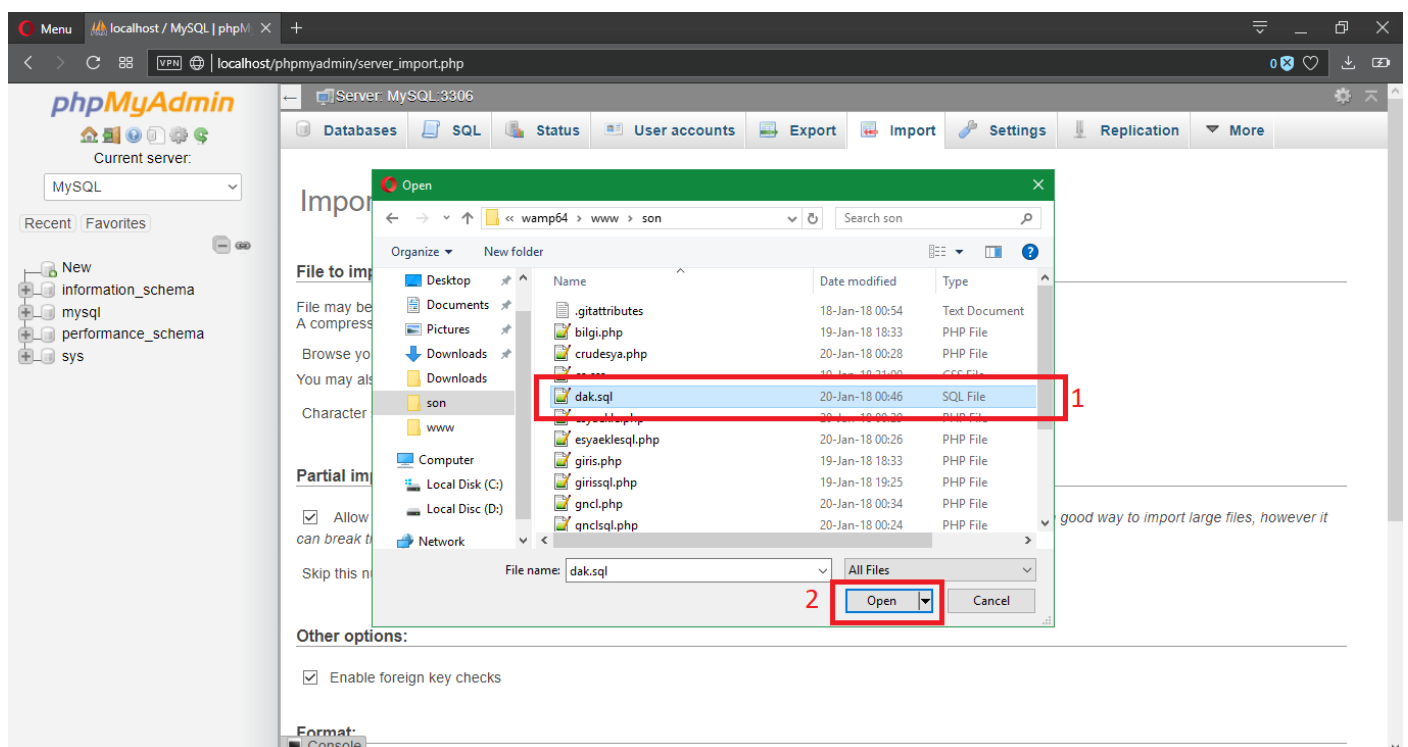
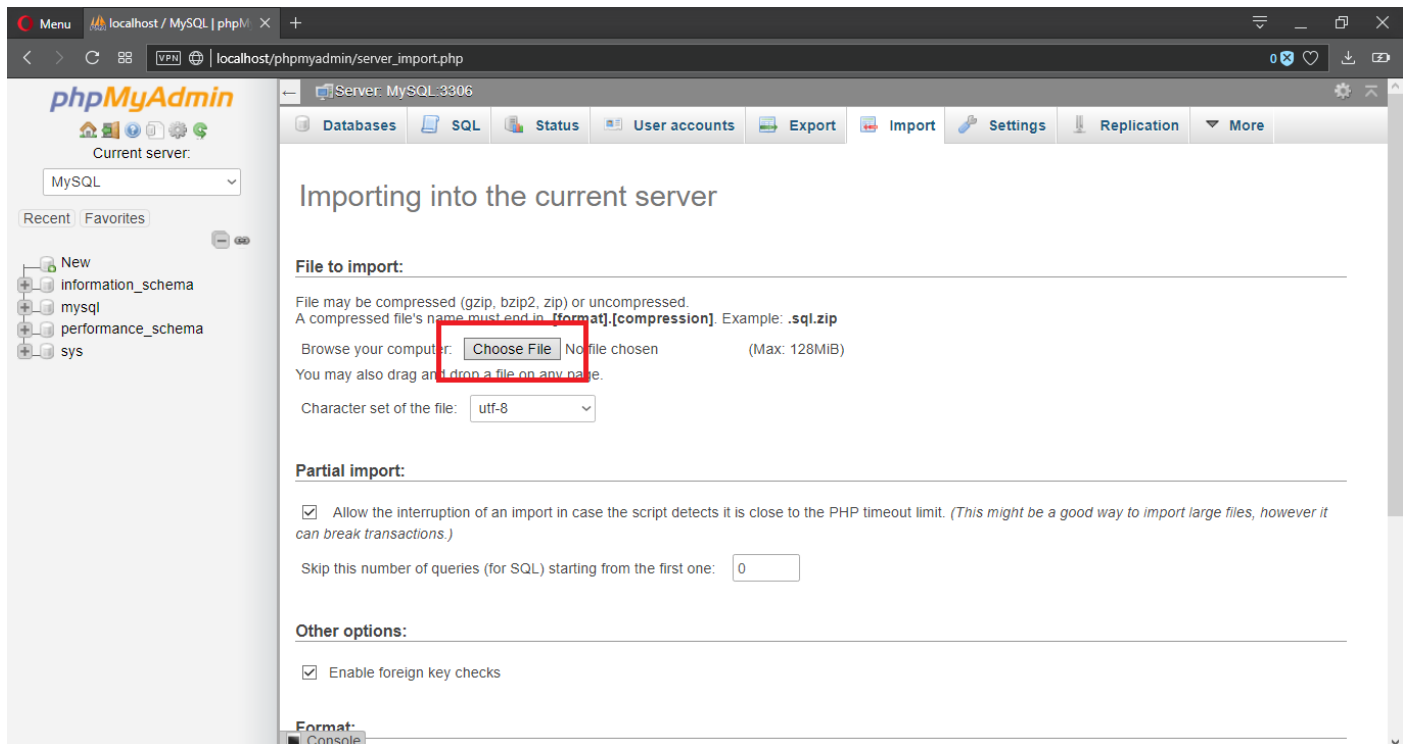
1.adım: Gereksinim analizindeki verilere uygun veritabanı ve tablolar MySQL'de oluşturulur.

Veya hazır olan veritabanı `duk.sql` , `http://localhost/phpmyadmin/` 'den ekran alıntılarındaki işlem sırası takip edilerek eklenmelidir.

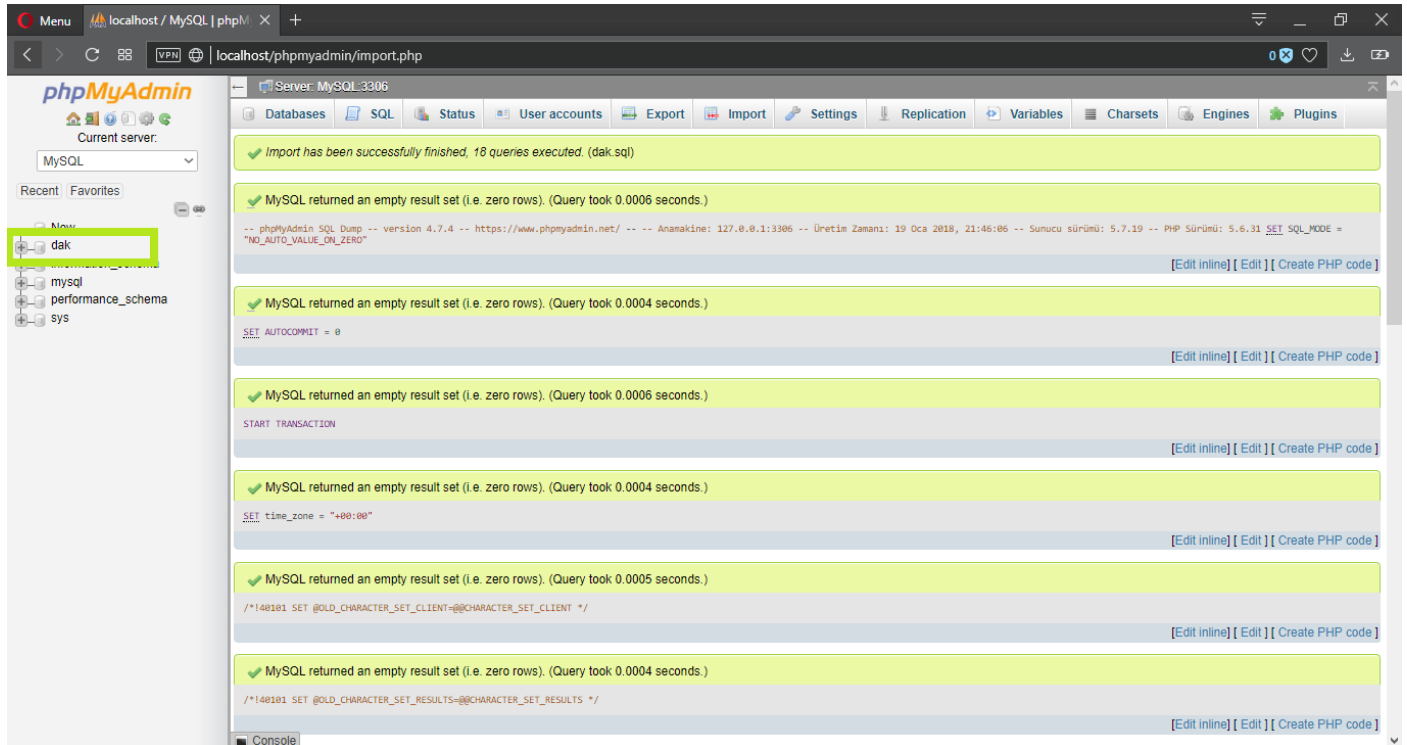
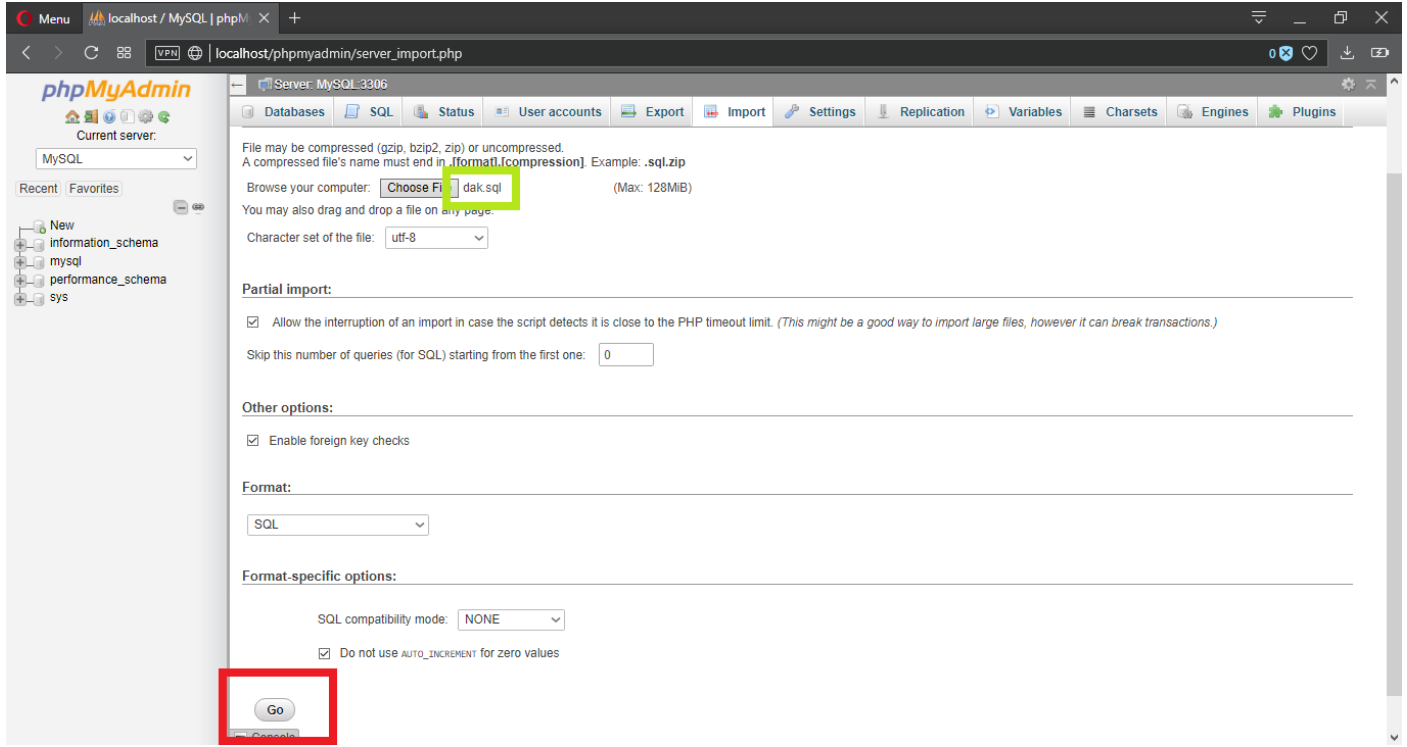
Ekran alıntıları:



# UYGULAMA(DEVAM)

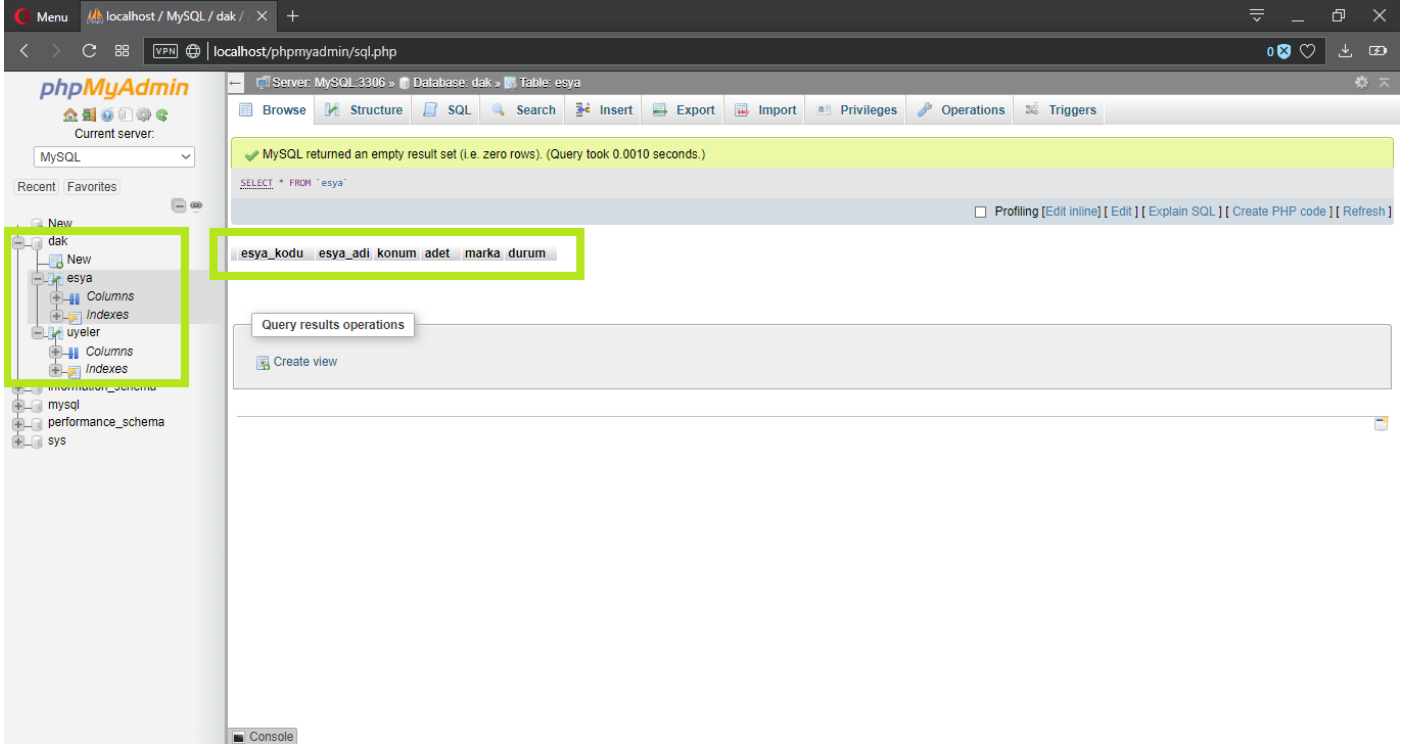


# UYGULAMA(DEVAM)





## UYGULAMA(DEVAM)



Veritabanı eklenmesi tamamlandı.

Xampp ve WampServer için phpmyadmin paneli farklıdır. Hangisi kullanılacaksa veri tabanı onun phpmyadmin panelinden kurulması gerekir.

C:\wamp64\www klasörünün içine veya C:\xampp\htdocs klasörünün içine, proje klasörünüzü atın.

Benim projem DAK bu klasörde. Php ile kodlanan sitenin çalışabilirliği internet tarayıcısıyla doğrulanır.

Tarayıcının adres kısmına <http://localhost/projenizinadi/> yazarak test işlemlerinizi yapabilirsiniz.

Benim projemde herhangi bir ad değişikliği vs. yapmayın. Proje DAK adlı proje klasörüne ve dak.sql adlı veritabanına göre hazırlanmıştır.

Proje işlevsel olarak çalıştığında CSS ile site görselleştirilmeleri yapılır.

Benim görselleştirme stil dosyam cs.css 'dir.

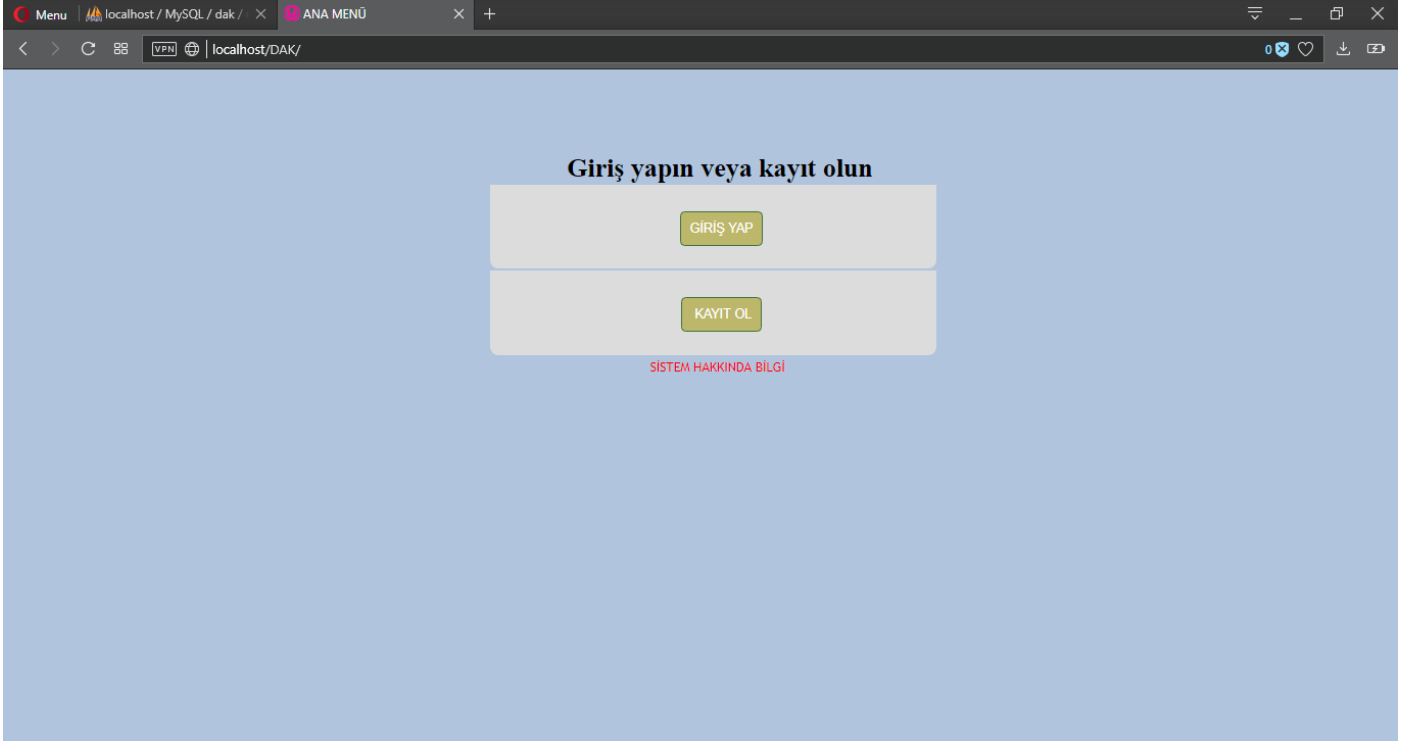
Sitenin planlandığı şekilde çalışıp çalışmadığı kontrol edilir.

Hatalar eksiklikler giderilir. Site için gerekli açıklamalar ve sadeleştirmeler yapılır.

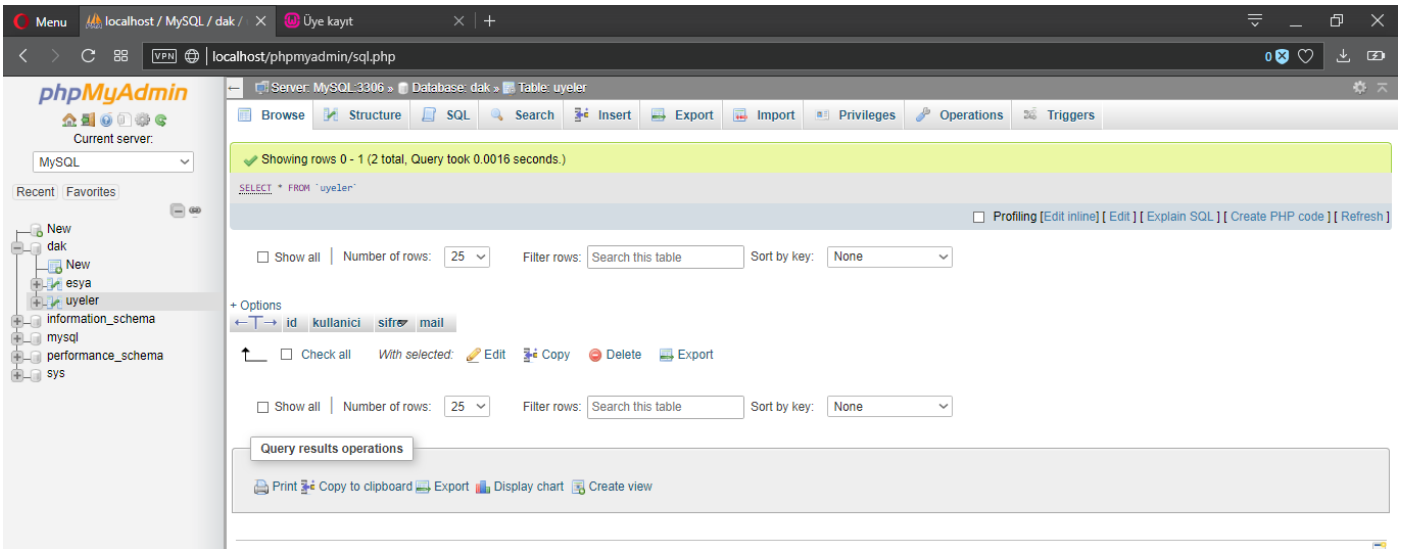
# SONUÇLAR

Güvenlik açıklarının en aza indirilmeye çalışıldığı ve istenilen işlemleri yerine getirebilen bir site hazırlandı.

Apache ve MySQL aktif olduğunda siteye bir bakalım:

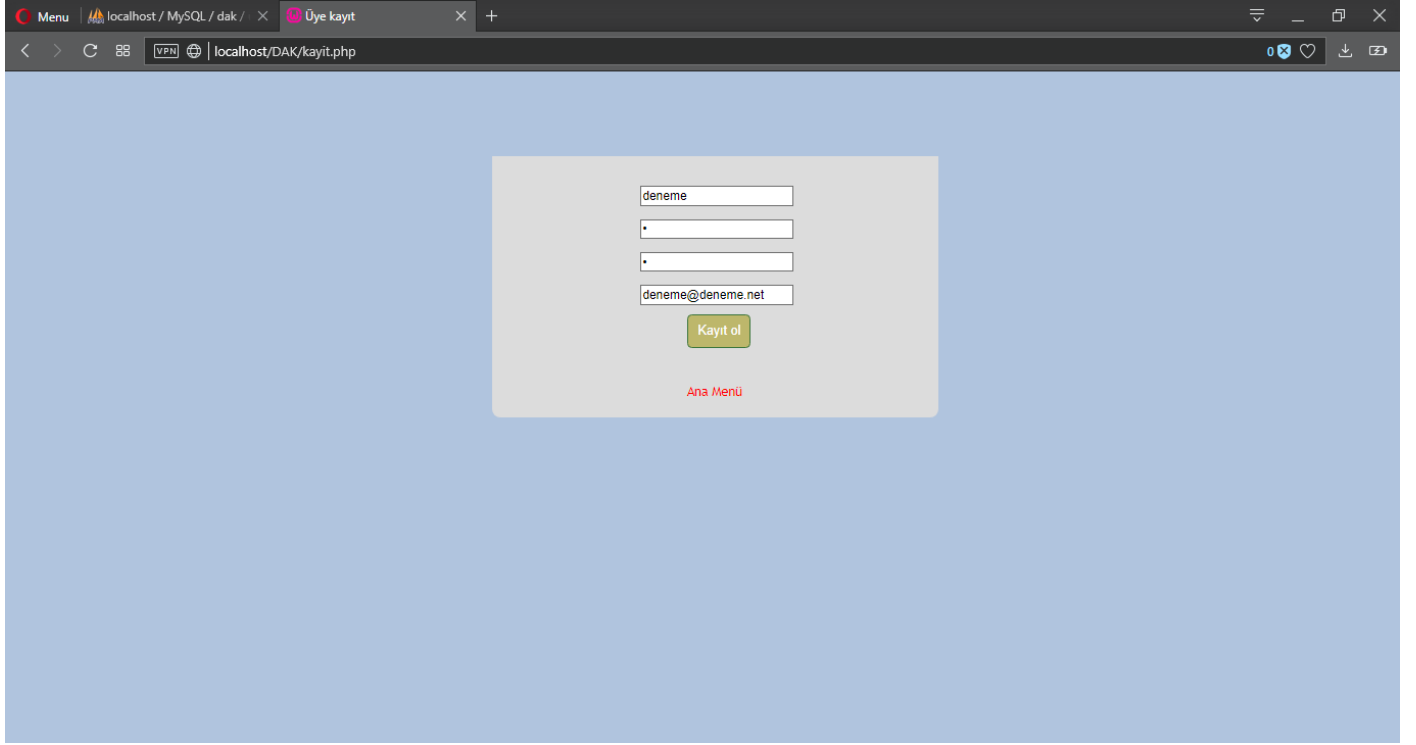


Veri tabanına bakalım: Kayıtlı üye yok.



## SONUÇLAR(DEVAM)

Yeni üye oluşturalım.



Menu localhost / MySQL / dak / Üye kayıt

localhost/DAK/kayit.php

deneme

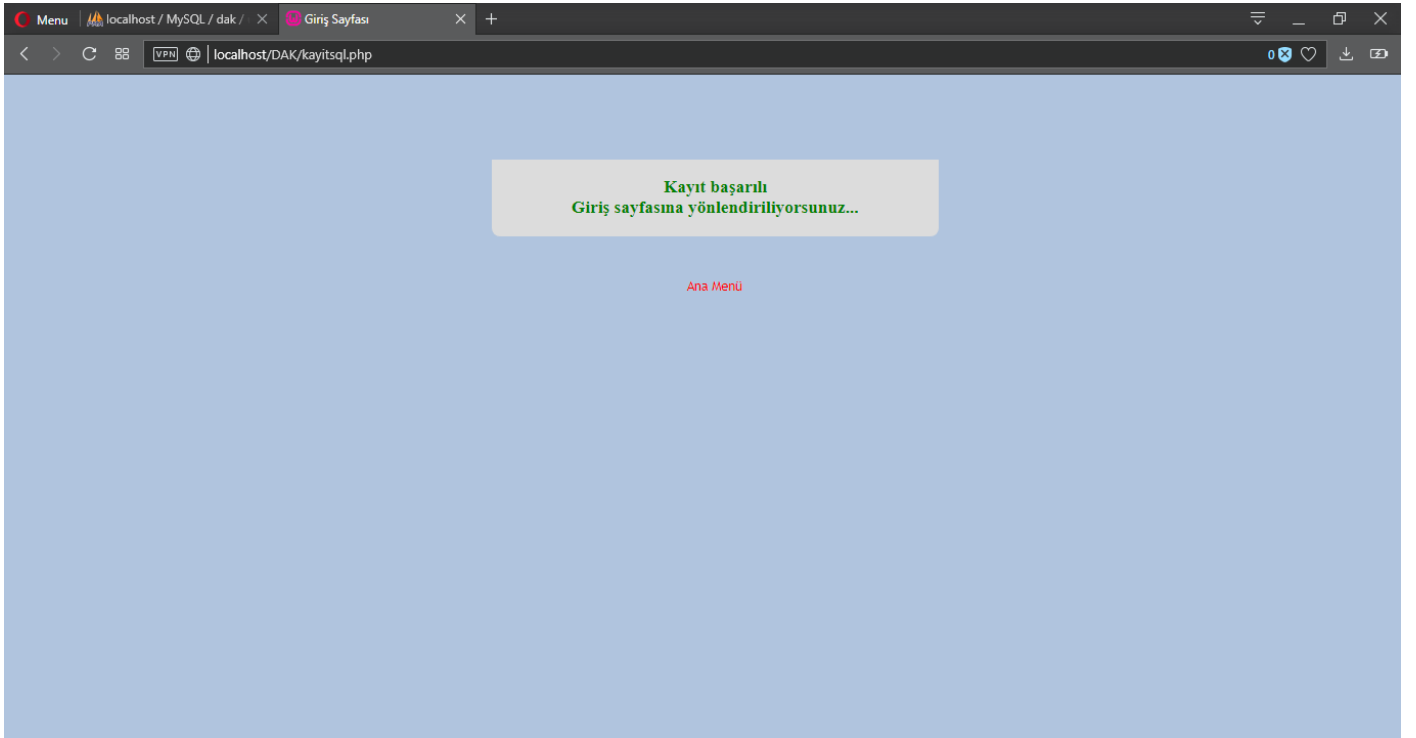
\*

\*

deneme@deneme.net

Kayıt ol

[Ana Menü](#)



Menu localhost / MySQL / dak / Giriş Sayfası

localhost/DAK/kayitsql.php

Kayıt başarılı  
Giriş sayfasına yönlendiriliyorsunuz...

[Ana Menü](#)

Kayıt olduk mu? Veri tabanına bakalım:

## SONUÇLAR(DEVAM)

The screenshot shows the phpMyAdmin interface. The left sidebar displays the database structure with 'dak' selected. The main area shows the 'uyeler' table in the 'dak' database. The table has one row with the following data:

id	kullanici	sifre	mail
1	deneme	1	deneme@deneme.net

Evet kayıt başarılı. id eşsiz(benzersizdir)

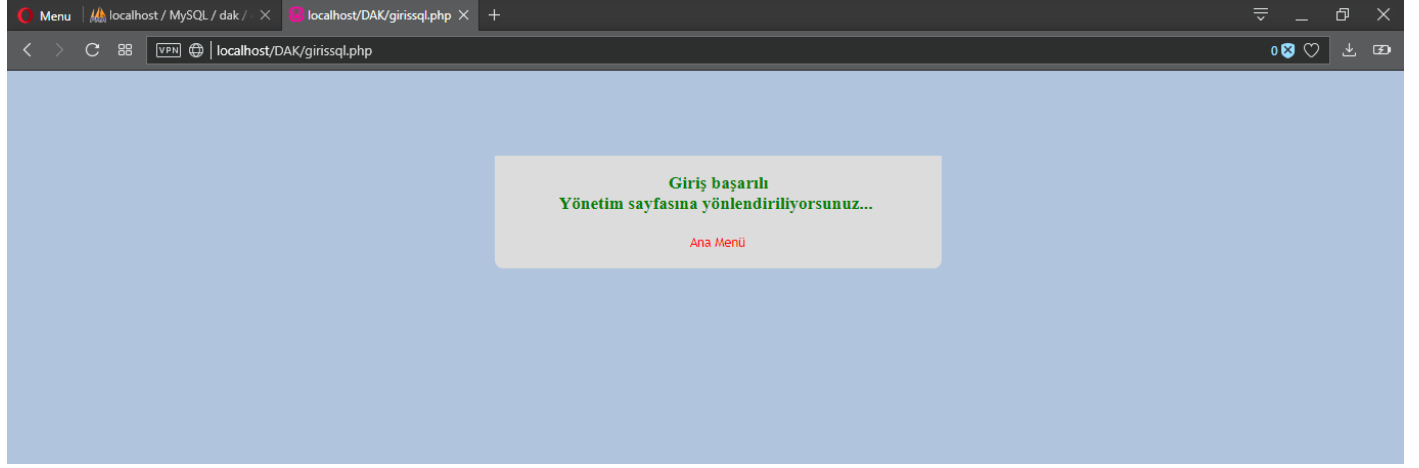
Şimdi kayıtlı hesapla giriş yapalım.

The screenshot shows the login page (giris.php) with a light blue background. A white box in the center contains the login form:

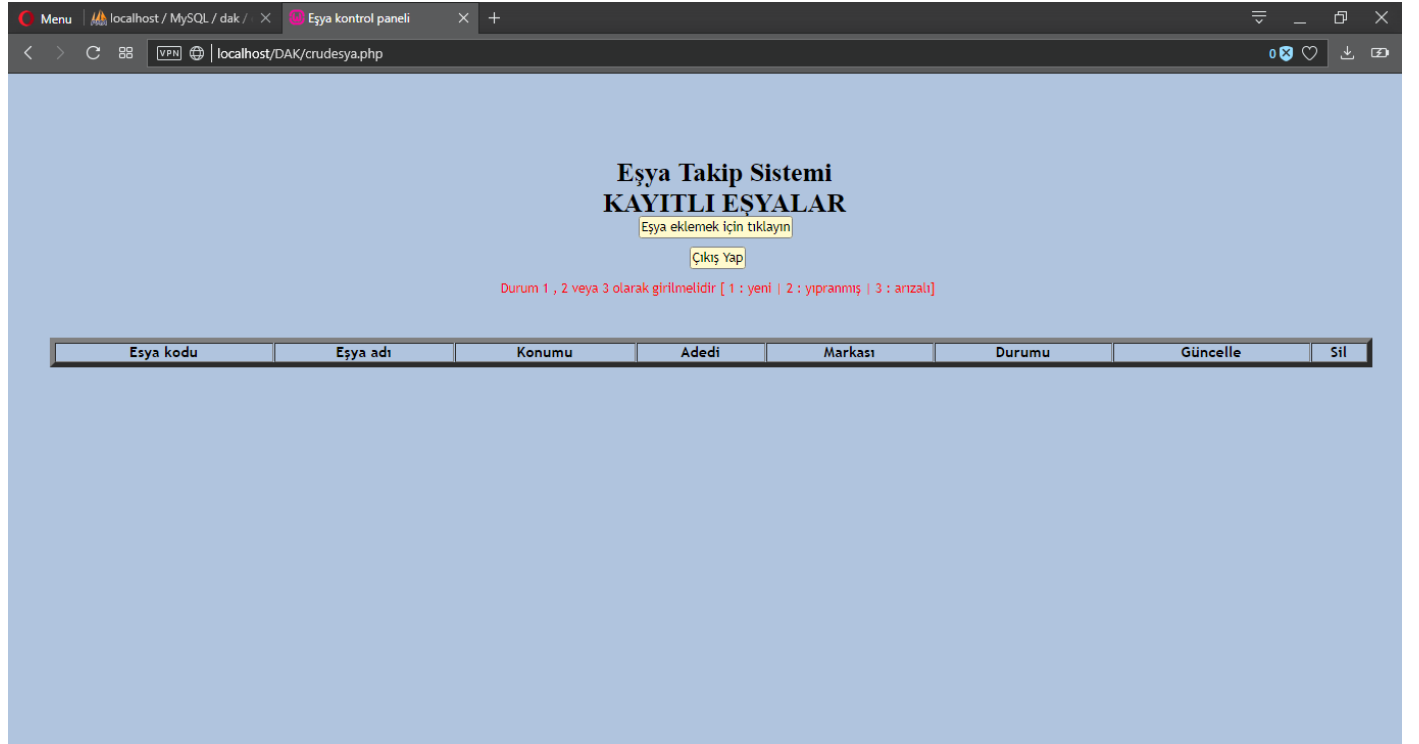
- A text input field containing 'deneme'.
- A password input field with a dot as a placeholder.
- A green 'Giriş Yap' button.
- A red link labeled 'Ana Menü'.

# SONUÇLAR(DEVAM)

Giriş başarılı.



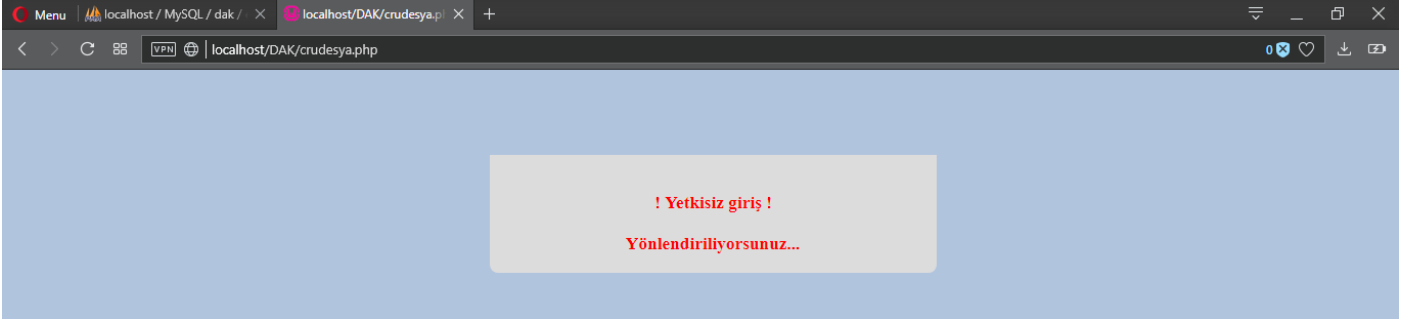
Yönetici sayfası: Şimdilik eşya yok.



## SONUÇLAR(DEVAM)

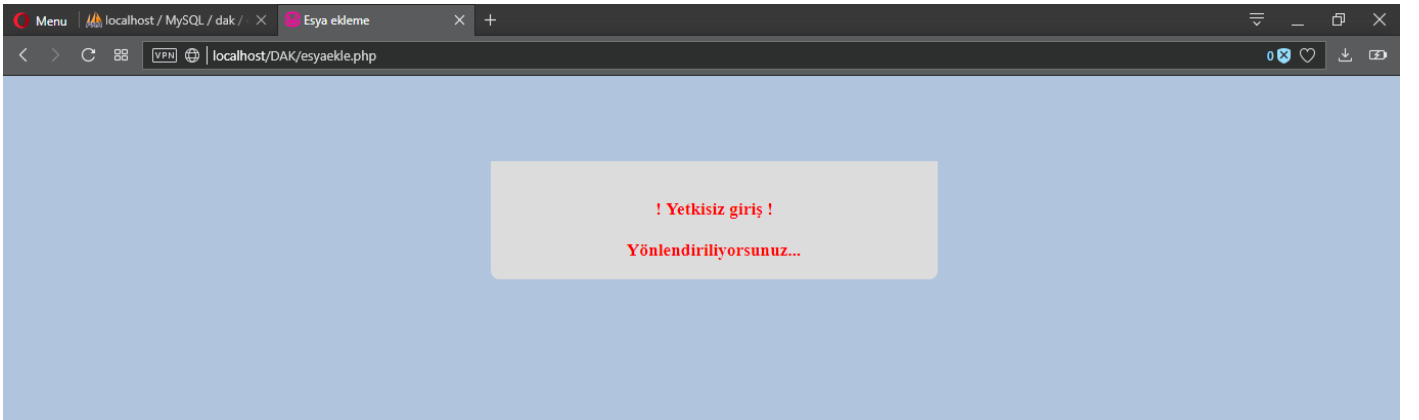
Peki giriş yapmadan yani kullanıcı kaydı olmaksızın herhangi bir yönetim sayfasına erişmeye çalışırsak:

Örn: adres kısmına giriş yapmadan önce yönetici sayfa adresini localhost/DAK/crudesya.php yazarsak.



Giriş yapan üye sistemden çıkış yapıp tekrar giriş yapmadan yetkisiz girişi denerse aynı sonuçları alır.

Adres kısmına giriş yapmadan önce eşya ekleme sayfa adresini localhost/DAK/esyaekle.php yazarsak.

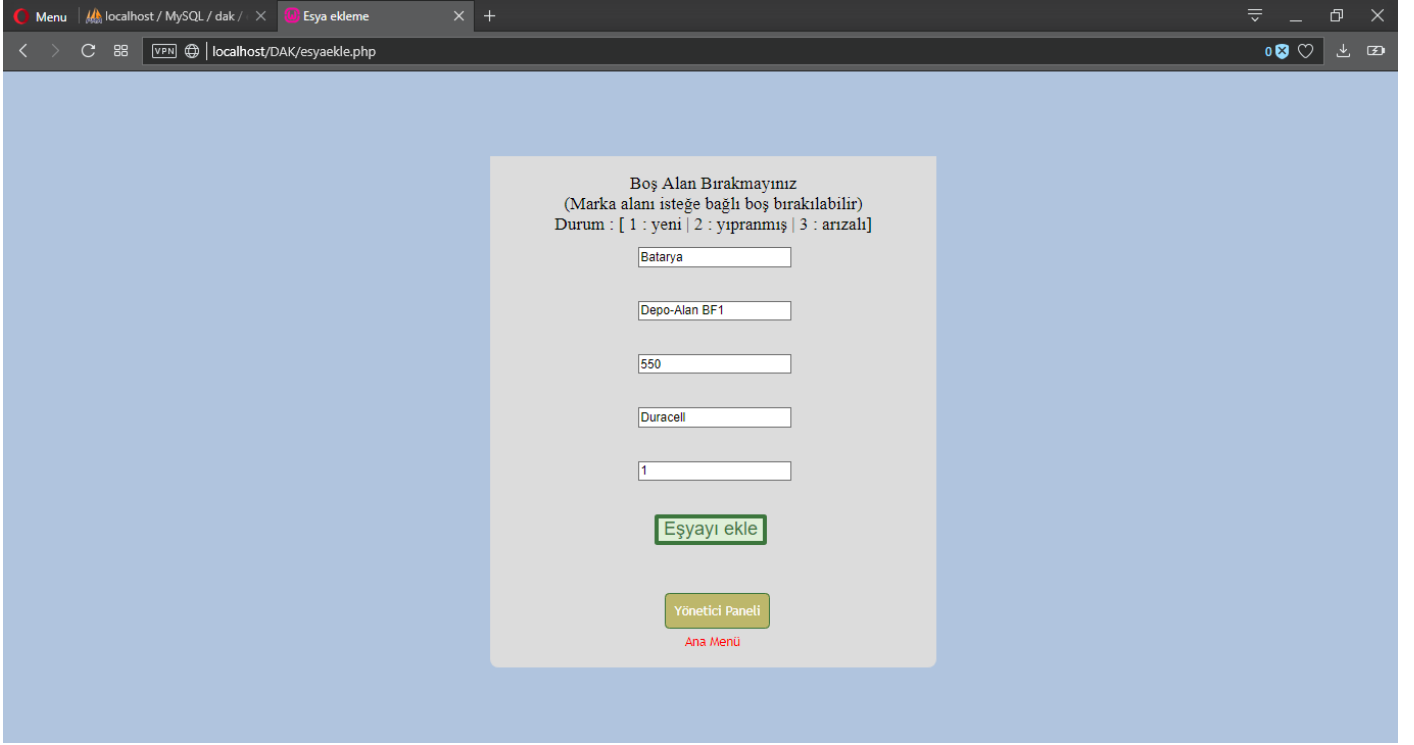


Diğer bütün kritik önem taşıyan sayfalar giriş yapmadan kendilerine ulaşmaya çalışan kişiye aynı çıktıyı verir.

Boş bırakılan form alanları, önceden kullanılmış kullanıcı adı ve e-postalar, sınırı aşan eşya özellik girişi vb durumlar ilgili sayfalarda engellendi.

# SONUÇLAR(DEVAM)

Giriş yapıp yönetim panelinden eşya ekleyelim.



Boş Alan Bırakmayınız  
(Marka alanı isteğe bağlı boş bırakılabilir)  
Durum : [ 1 : yeni | 2 : yıpranmış | 3 : arızalı]

Batarya

Depo-Alan BF1

550

Duracell

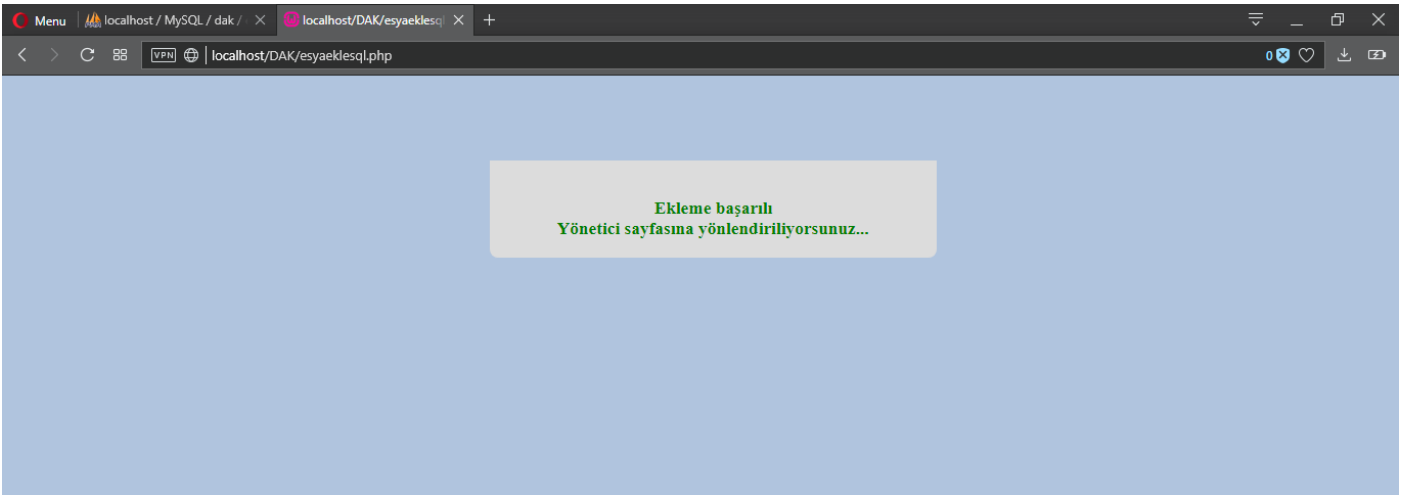
1

Eşyayı ekle

Yönetici Paneli

Ana Menü

Eşya ekleye tıklayalım.



Ekleme başarılı  
Yönetici sayfasına yönlendiriliyorsunuz...

Başarılı şekilde eşya eklenmiş.

# SONUÇLAR(DEVAM)

Yönetim paneline yönlendirildik. Eşyalarımız listelendi.

**Eşya Takip Sistemi**  
**KAYITLI EŞYALAR**  
Eşya eklemek için tıklayın  
Çıkış Yap

Durum 1 , 2 veya 3 olarak girilmelidir [ 1 : yeni | 2 : yıpranmış | 3 : arızalı]

Eşya kodu	Eşya adı	Konumu	Adedi	Markası	Durumu	Güncelle	Sil
1	Batarya	Depo-Alan BF1	550	Duracell	1	Güncelle	Sil

Veri tabanında eşyaya bakalım.

Server: MySQL:3306 Database: dak Table: esya

Showing rows 0 - 0 (1 total, Query took 0.0018 seconds.)

SELECT \* FROM `esya`

Options: esya\_kodu esya\_adi konum adet marka durum

esya_kodu	esya_adi	konum	adet	marka	durum
1	Batarya	Depo-Alan BF1	550	Duracell	1

Query results operations: Print, Copy to clipboard, Export, Display chart, Create view



# SONUÇLAR(DEVAM)

Bir başka eşya ekleyelim.

Menu Eşya ekleme localhost / MySQL / dak / localhost/DAK/esyaekle.php

Boş Alan Bırakmayınız  
(Marka alanı isteğe bağlı boş bırakılabilir)  
Durum : [ 1 : yeni | 2 : yıpranmış | 3 : arızalı]

LCD Panel

Depo-Alan BG1

360

Panasonic

1

Eşyayı ekle

Yönetici Paneli

Ana Menü

Eşyayı güncelleyelim.

Menu UPDATE localhost / MySQL / dak / localhost/DAK/gnci.php

GÜNCELLEMEK İSTEDİĞİNİZ EŞYANIN ESKİ BİLGİLERİ

Eşya kodu	Eşya adı	Konumu	Adedi	Markası	Durumu
2	LCD Panel	Depo-Alan BG1	360	Panasonic	1

Durum 1 , 2 ve 3 olarak girilmelidir  
[ 1 : yeni | 2 : yıpranmış | 3 : arızalı]

Boş\* Alan Bırakmayınız  
(\*Marka alanı isteğe bağlı boş bırakılabilir)

LCD Panel

Depo-Alan BF1

10

Panasonic

2

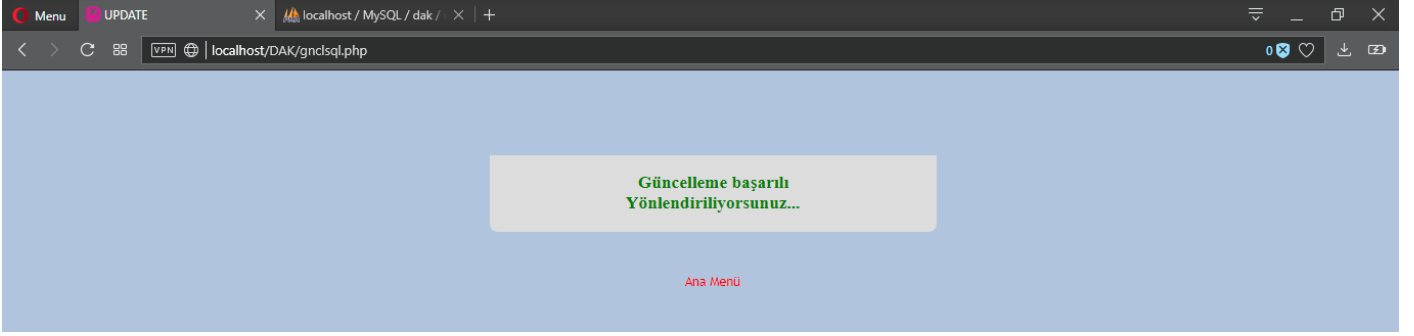
Eşyayı Güncelle

Yönetici Paneli

Ana Menü

# SONUÇLAR(DEVAM)

Eşya düzgün şekilde güncellendi.



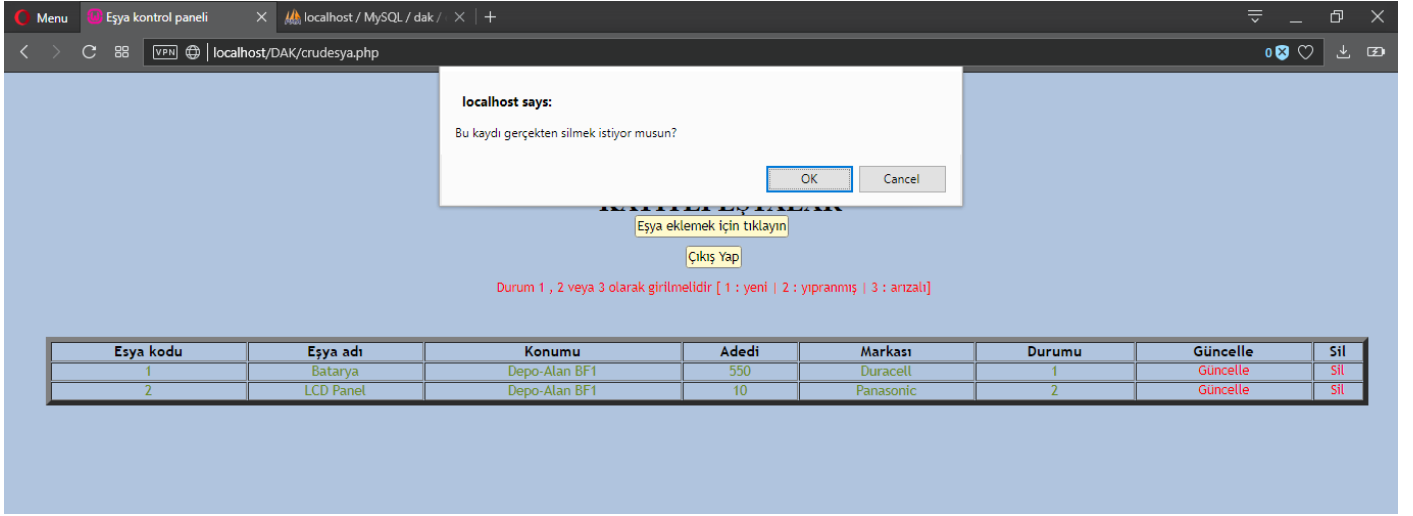
Eşya listesinde eşya güncellenmiş olarak görülmektedir.



# SONUÇLAR(DEVAM)

Eşya silmeye çalışalım.

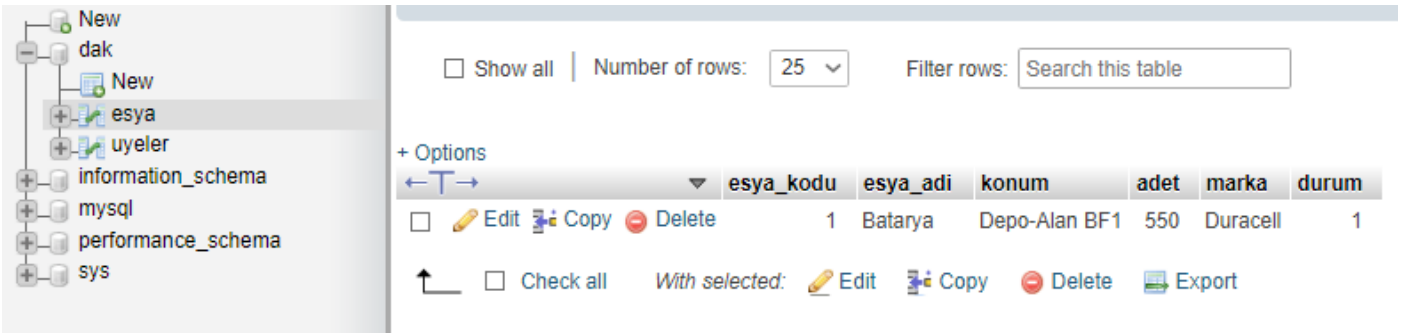
2.esya kodlu eşyanın sil butonuna basıyoruz.



Sonuç başarılı eşya silindi.



Çıkış yapalım ve siteyi kapatıp veri tabanına bakalım.



Veri tabanında sildiğimiz veri satırı yok. Fonksiyonlar düzgün çalışıyor.

# KAYNAK KODLAR

Kodlar kod yapısı bozulmaması için küçük boyuttadır.

## PROJE GITHUB LINK:

<https://github.com/recepkarademir/Computer-Engineering-Project-2>

bilgi.php

```
<link rel="stylesheet" href="cs.css"> <!-- CSS kullanabilmek için. -->
<?php // Hiçbir yerden direkt olarak esya menüsüne izinsiz erişememesi için.
    session_start(); // SESSION ları kullanabilmek için.
    $izin=0;// izin 0 yapılıyor yani giriş izni şimdilik yok.
    $_SESSION["test"]=$izin;// SESSION kullanarak başka sayfalara izin durumu taşınıyor.
?>
<?php ## Bu sayfa sistem hakkında bilgiler içerir.
echo "<br> [D]URUM<br>[A]DET<br>[K]ONUM<br>PROJESİ<br>Bu proje evde veya iş yerinde var olan fakat yeri, adedi
ve durumu bilinmeyen eşyaların kontrol altında tutulması için geliştirilmiştir.<br>
Örneğin, bir depoda eşyaların üzerlerine yerleştirilen etiket ile eşyanın bilgileri ve anlık hareketi etiket okuyucuları tarafından <br>
algılanarak bu sisteme otomatik girilirse, eşyaların durumu sistemde güncel olarak kontrol edilebilir.<br>
Bazen, ihtiyacımız olan eşyanın evde var olduğunu biliriz fakat eşyanın yerini,durumunu ve adedini unuturuz.<br>
Bu ve buna benzer durumlar eşyanın yeri, durumu ve adedi güncel olarak tutularak çözülebilir.<br>
Hatta, eşya temin edildiği an bir kez sisteme kaydedilmesi bile bazen yeterli olacaktır.<br>
Kullanılan eşya alındığı yere bırakıldığında veya kullanılıp atık olduğunda, ikinci kez kullanılmak istenilirse sistemdeki son konumuna bakıp eşyanın<br>
bittiği veya kullanılmış olduğu sistem güncellenmeden ve olası diğer yerlere bakılmadan bile bilinebilir.<br>
Sistemin tam olarak sorunsuz çalışması için etkileşime geçilen eşyayla ilgili güncelleme yapılmalıdır.<br>
Güncellemeler yapılmasa dahi, eşya temin edildiğinde sisteme girilmişse, eşyanın son konumuna bakarak eşyanın akıbeti öğrenilebilir.<br>
dolayısı İbrahim Varol'a teşekkür ederim.<br>";
?>
<br>
<a href="index.php">Ana menüye dönmek için tıkla</a> <br/> <!-- Anasayfaya geri dönüş. -->
```

# KAYNAK KODLAR

## Crudesya.php

```
<link rel="stylesheet" href="cs.css"> <!-- CSS kullanabilmek için. -->
<?php
    session_start();

    $test=$_SESSION["test"];

    if($test ==0)
    {
        ?><form id="form" actin="" method="">
            <?php
                echo '<b style="color:red"><br> ! Yetkisiz giriş !<br><br>Yönlendiriliyorsunuz... </b>';
                echo '<meta http-equiv="refresh" content ="1;URL=http://localhost/DAK/giris.php">';
                session_destroy();
                die; ## Eğer giriş yapmadan gelmeye çalışırsa burdan sonrasına erişemeyecek. girissql.php deyken denerse giremeyecek.
                ## Eğer giriş yapmadan gelmeye çalışırsa burdan sonrasına erişemeyecek. giris.php deyken denerse giremeyecek.
            ?>
        </form>
    <?php

    }
    require("sqlconnect.php"); // include yerine require kullanıldı çünkü hatayla karşılaşıldığında işleme devam edilemesin.
?>
<html>

<head>

    <meta charset="UTF-8" />
    <title>Eşya kontrol paneli</title>

</head>

<body>

<h2 align="center">Eşya Takip Sistemi<br>

KAYITLI EŞYALAR</h2>

<a class="bos" align="center" href="http://localhost/DAK/esyaeikle.php">Eşya eklemek için tıklayın<a/> <br><br>
<a class="bos" href="http://localhost/DAK/index.php">Çıkış Yap<a/><br><br>

<table width="97%" border="5px">
    <tr>
        <th>Eşya kodu</th>
        <th>Eşya adı</th>
        <th>Konumu</th>
        <th>Adedi</th>
        <th>Markası</th>
        <th>Durumu</th>
        <th>Güncelle</th>
        <th>Sil</th>
    </tr>
    <?php
        echo " Durum 1 , 2 veya 3 olarak girilmelidir [ 1 : yeni | 2 : yıpranmış | 3 : arızalı]";
        $izin=1; // Crud yönetim panelinde sayfayı sorunsuz yenileyebilmek için.
        $_SESSION["test"]=$izin; // Crud yönetim panelinde sayfayı sorunsuz yenileyebilmek için.

        $sorgula = $dak -> query("SELECT *FROM esya "); // Kullanıcı adı veya sifre kontrol.

        while($row=$sorgula ->fetch(PDO::FETCH_ASSOC))
        {
            Print "<tr>";
            Print '<td align="center">'. $row['esya_kodu'] . "</td>";
            Print '<td align="center">'. $row['esya_adi'] . "</td>";
            Print '<td align="center">'. $row['konum']. "</td>";
            Print '<td align="center">'. $row['adet']. "</td>";
            Print '<td align="center">'. $row['marka']. "</td>";
            Print '<td align="center">'. $row['durum']. "</td>";
            Print '<td align="center"><a href="gncl.php?esya_kodu='. $row['esya_kodu'] .' ">Güncelle<a/> </td>';
            Print '<td align="center"><a href="#" onclick="myFunction('.$row['esya_kodu'].') ">Sil<a/> </td>';
            Print "</tr>";
        }
    ?>

</table>
<script>
function myFunction(esya_kodu)
{
    var r=confirm("Bu kaydı gerçekten silmek istiyor musun?");
    if (r==true)
    {
        window.location.assign("sil.php?esya_kodu=" + esya_kodu);
    }
}
</script>
</body>

</html>
```

# KAYNAK KODLAR

## Cs.css

```
* {
    margin: 1px;
    margin-left:1%;
    margin-top:3%;
}
a{
    text-decoration:none;
    text-align: center;
    font-family: 'Trebuchet MS';
    color: red;
    font-size:14px;
}
td{
    text-decoration:none;
    text-align: center;
    font-family: 'Trebuchet MS';
    color: OliveDrab;
    font-size:15px;
}
body {
    font-size: 120%;
    text-align: center;
    background: #B0C4DE;
}

.header {
    width: 30%;
    margin: 50px auto 0px;
    color: white;
    background: #5F9EA0;
    text-align: center;
    border: 1px solid #B0C4DE;
    font-family: 'Trebuchet MS';
    border-bottom: none;
    border-radius: 10px 10px 0px 0px;
    padding: 20px;
}
form, .content {
    width: 30%;
    margin: 0px auto;
    padding: 20px;
    border: 1px solid #B0C4DE;
    background: Gainsboro;
    text-align: center;
    border-radius: 0px 0px 10px 10px;
}
.input-group {
    margin: 10px 0px 10px 0px;
    text-align: center;
}
.input-group label {
    display: block;
    text-align: center;
    margin: 3px;
}
.input-group input {
    height: 20px;
    width: 90%;
    padding: 5px 5px;
    font-size: 16px;
    text-align: center;
    border-radius: 5px;
    border: 1px solid gray;
}
.btn {
    padding: 10px;
    font-size: 14px;
    color: white;
    background: #BDB76B;
    text-align:center;
    margin: 5px;
    margin-left:5%;
    margin-top:2%;
    border:1px solid #3c763d;
    border-radius: 5px;
}
.bos {
    padding: 3px;
    font-size: 14px;
    color: black;
    background: #FFFACD;
    text-align:center;
    border: 1px solid gray;
    border-radius: 3px;
}
.err {
    width: 92%;
    padding: 10px;
    border: 1px solid #a94442;
    color: #a94442;
    margin: 5px;
    margin-left:5%;
    margin-top:2%;
    text-align: center;
    background: #f2dede;
    text-decoration:none;
    border-radius: 50px;
}
.success {
    color: #3c763d;
    text-decoration:none;
    background: #dff0d8;
    border: 4px solid #3c763d;
    border-radius: 3px;
    margin-bottom: 30px;
    text-align: center;
    font-size:20px;
}
```

**Her etiket özgün ve kullanıldığı kısımla alakalı adlandırdım. Css kodlarını site ile uyumlu olacak şekilde yazdım.**

# KAYNAK KODLAR

dak.sql

```
* {
    margin: 1px;
    margin-left:1%;
    margin-top:3%;
}
a{
    text-decoration:none;
    text-align: center;
    font-family: 'Trebuchet MS';
    color: red;
    font-size:14px;
}
td{
    text-decoration:none;
    text-align: center;
    font-family: 'Trebuchet MS';
    color: OliveDrab;
    font-size:15px;
}
body {
    font-size: 120%;
    text-align: center;
    background: #B0C4DE;
}

.header {
    width: 30%;
    margin: 50px auto 0px;
    color: white;
    background: #5F9EA0;
    text-align: center;
    border: 1px solid #B0C4DE;
    font-family: 'Trebuchet MS';
    border-bottom: none;
    border-radius: 10px 10px 0px 0px;
    padding: 20px;
}
form, .content {
    width: 30%;
    margin: 0px auto;
    padding: 20px;
    border: 1px solid #B0C4DE;
    background: Gainsboro;
    text-align: center;
    border-radius: 0px 0px 10px 10px;
}
.input-group {
    margin: 10px 0px 10px 0px;
    text-align: center;
}
.input-group label {
    display: block;
    text-align: center;
    margin: 3px;
}
.input-group input {
    height: 20px;
    width: 90%;
    padding: 5px 5px;
    font-size: 16px;
    text-align: center;
    border-radius: 5px;
    border: 1px solid gray;
}
.btn {
    padding: 10px;
    font-size: 14px;
    color: white;
    background: #BDB76B;
    text-align:center;
    margin: 5px;
    margin-left:5%;
    margin-top:2%;
    border:1px solid #3c763d;
    border-radius: 5px;
}
.bos {
    padding: 3px;
    font-size: 14px;
    color: black;
    background: #FFFACD;
    text-align:center;
    border: 1px solid gray;
    border-radius: 3px;
}
.err {
    width: 92%;
    padding: 10px;
    border: 1px solid #a94442;
    color: #a94442;
    margin: 5px;
    margin-left:5%;
    margin-top:2%;
    text-align: center;
    background: #f2dede;
    text-decoration:none;
    border-radius: 50px;
}
.success {
    color: #3c763d;
    text-decoration:none;
    background: #dff0d8;
    border: 4px solid #3c763d;
    border-radius: 3px;
    margin-bottom: 30px;
    text-align: center;
    font-size:20px;
}
```

# KAYNAK KODLAR

## esyaekle.php

```
<html>

<head>

<meta charset="UTF-8" /> <!-- Türkçe karakter için. -->
<title>Eşya ekleme</title>
<link rel="stylesheet" href="cs.css"> <!-- CSS kullanabilmek için. -->

</head>

<body>

<form id="body" method="POST" action="esyaeklesql.php"> <!-- Eşya ekle butonuna basınca. -->
<?php

    session_start();
    $test=$_SESSION["test"]; // yetkisiz giriş engeli.

    if($test ==0)
    {
        echo '<b style="color:red"><br> ! Yetkisiz giriş !<br><br>Yönlendiriliyorsunuz... </b>';
        echo '<meta http-equiv="refresh" content ="1;URL=http://localhost/DAK/giris.php">';
        session_destroy();
        die; ## Eğer giriş yapmadan gelmeye çalışırsa burdan sonrasına erişemeyecek. girissql.php deyken denerse giremeyecek.
        ## Eğer giriş yapmadan gelmeye çalışırsa burdan sonrasına erişemeyecek. giris.php deyken denerse giremeyecek.
    }

    require("sqlconnect.php"); // include yerine require kullanıldı çünkü hatayla karşılaşıldığında işleme devam edilemesin.
    echo "Boş Alan Bırakmayınız<br>(Marka alanı isteğe bağlı boş bırakılabilir)<br>";
    echo "    Durum : [ 1 : yeni | 2 : yıpranmış | 3 : arızalı]";

    ?>

    <input type="text" name="esya_adi" placeholder    ="Eşya adı" /><br><br>
    <input type="text" name="konum" placeholder    ="Konumu" /><br><br>
    <input type="text" name="adet" placeholder    ="Eşya adedi" /><br><br>
    <input type="text" name="marka" placeholder    ="Eşyanın markası" /><br><br>
    <input type="text" name="durum" placeholder    ="Eşyanın durumu" /><br><br>
    <input type="submit" class="success" value="Eşyayı ekle"/> <a>

    <br><br><br>    <a class="btn" href="http://localhost/DAK/crudesya.php">Yönetici Paneli<a/>
    <br><br>    <a class="bosl" href="http://localhost/DAK/index.php">Ana Menü<a/>

</form>

</body>

</html>
```



# KAYNAK KODLAR

esyaeklesql.php

```
<link rel="stylesheet" href="cs.css"><!-- CSS kullanabilmek için. -->
<form action="crudesya.php" method="post"> <!-- İşlemler bittiğinde yönetim paneline dönecek. -->
    <?php
        session_start();
        $test=$_SESSION["test"]; // yetkisiz giriş engeli.

        if($test ==0)
        {
            ?><form id="form" actin="" method="">
            <?php
                echo '<b style="color:red"><br> ! Yetkisiz giriş !<br><br>Yönlendiriliyorsunuz... </b>';
                echo '<meta http-equiv="refresh" content ="1;URL=http://localhost/DAK/giris.php">';
                session_destroy();
                die; ## Eğer giriş yapmadan gelmeye çalışırsa burdan sonrasına erişemeyecek. girissql.php deyken denerse giremeyecek.
                ## Eğer giriş yapmadan gelmeye çalışırsa burdan sonrasına erişemeyecek. giris.php deyken denerse giremeyecek.
            ?>
            </form>
            <?php

        }
        require("sqlconnect.php");
        @$Esysaadi = $_POST["esya_adi"];
        @$Konum = $_POST["konum"];
        @$Adet = $_POST["adet"];
        @$Marka = $_POST["marka"]; // Marka ekleme veya güncelleme yapılırken boş bırakılabilir.
        @$Durum = $_POST["durum"];
        @$esya_kodu= $_GET['esya_kodu'];
        ?>

    <?php

        if($Esysaadi == "" || $Konum == "" || $Adet == "" || $Durum == "") // Boş alan olmamalı. Marka alanı hariç. Marksız esya
        {
            echo '<b style="color:red">Boş alan var <br> Yönlendiriliyorsunuz... !</b>';
            echo '<meta http-equiv="refresh" content ="3;URL=http://localhost/DAK/esyaekle.php">';
        }
        else if($Durum>3 || $Durum<1)
        {
            echo '<b style="color:red">Durum 1 , 2 veya 3 olarak girilmelidir. <br> Yönlendiriliyorsunuz... !</b>';
            echo '<meta http-equiv="refresh" content ="3;URL=http://localhost/DAK/esyaekle.php">';
        }
        else if($Adet<0 || $Adet>99999)
        {
            echo '<b style="color:red">Eşya adeti [1,99999] aralığında girilmelidir. <br> Yönlendiriliyorsunuz... !</b>';
            echo '<meta http-equiv="refresh" content ="3;URL=http://localhost/DAK/esyaekle.php">';
        }
        else
        {
            echo "<br>";
            $ekle = $dak -> prepare("INSERT INTO esya SET esya_adi = ?, konum = ?, adet = ?, marka= ?, durum = ?"); // SQL
            $ekle ->execute(array($Esysaadi,$Konum,$Adet,$Marka,$Durum)); // Gidecek veriler bir dizi yardımıyla gönderiliyor.

            if($ekle)
            {
                echo '<b style="color:green">Ekleme başarılı <br>Yönetici sayfasına yönlendiriliyorsunuz...</b>'; //
                echo '<meta http-equiv="refresh" content ="3;URL=http://localhost/DAK/crudesya.php">';
            }
            else
            {
                echo '<b style="color:red">Ekleme başarılı olmadı !</b>'; // Ekleme başarılı bir şekilde
                echo '<meta http-equiv="refresh" content ="3;URL=http://localhost/DAK/crudesya.php">';
            }
        }
    ?>

</form>
```

# KAYNAK KODLAR

## giris.php

```
<?php // Hiçbir yerden direkt olarak esya menüsüne izinsiz erişememesi için.
    session_start();// SESSION ları kullanabilmek için.
    $izin=0; // izin 0 yapılıyor yani giriş izni şimdilik yok.
    $_SESSION["test"]=$izin; // SESSION kullanarak başka sayfalara izin durumu taşıyor.
?>
<html>

    <head>
        <meta charset="UTF-8" /> <!-- Türkçe karakter için. -->
        <title>Üye giriş</title>
        <link rel="stylesheet" href="cs.css"> <!-- CSS kullanabilmek için. -->
    </head>

    <body>
        <form id="form" method="POST" action="girissql.php"> <!-- Giriş yap butonuna tıklanınca girissql.php ye gidecek. -->
            alanlarındaki silik bilgi yazıları için -->
            alınıyor. -->
            ye gidiyor.-->
            <input type="text" class="bos" name="kullanici" placeholder="Kullanıcı adınız" /><br> <!-- placeholder form
            <input type="password" class="bos" name="sifre" placeholder="Şifreniz" /><br> <!-- Kullanıcı ve şifre bilgileri
            <input type="submit" class="btn" value="Giriş Yap"/> <a> <!-- Bilgiler veri tabanında kontrol edilmek üzere girissql.php
            <br><br><br><a href="http://localhost/DAR/index.php">Ana Menü<a/> <!-- Anasayfaya geri dönüş. -->
        </form>
    </body>
</html>
```

# KAYNAK KODLAR

## girissql.php

```
<link rel="stylesheet" href="cs.css"><!-- CSS kullanabilmek için. -->
<?php // Hiçbir yerden direkt olarak esya menüsüne izinsiz erişememesi için.
    session_start(); // SESSION ları kullanabilmek için.
    $izin=0; // izin 0 yapılıyor yani giriş izni şimdilik yok.
    $_SESSION["test"]=$izin; // SESSION kullanarak başka sayfalara izin durumu taşıyor.
?>
<?php

    require("sqlconnect.php"); // include yerine require kullanıldı çünkü hatayla karşılaşıldığında işleme devam edilemesin.

    // POST Metodu ile giris.php deki veriler bu php dosyasına çağrılıyor.
    @$Kullanici = $_POST["kullanici"]; // POST metodu yardımıyla formdan gelen veriler alınıyor.
    @$Sifre = $_POST["sifre"]; // Küçük büyük duyarlı olduğundan böyle tanımlamalar yapıldı.

    $sorgula = $dak -> prepare("SELECT *FROM uyeler WHERE kullanici = ? AND sifre = ?"); // Veri tabanından kullanıcı adı ve sifre kontrolü için veriler çekiliyor..
    $sorgula ->execute(array($Kullanici,$Sifre)); // Veriler array(dizi) kullanarak çekiliyor.
    $sqloku = $sorgula -> fetch(PDO::FETCH_ASSOC);

?>

    <form action="giris.php" method="post"> <!-- Eğer giriş ekranındaki alan(lar) boş bırakılırsa giris.php gidecek.-->
        <?php
            if($Kullanici == "" || $Sifre == "") // Eğer gerekli alanlardan bazıları boşsa.
            {
                echo '<b style="color:red"><br>Boş Alan Bırakmayınız</b>';?>
                <br><button class="err" type="submit">Tamam</button><?php // err etiketli css stilini kullan.
            }
            else if($sorgula -> rowcount()<=0) // Eğer kullanıcı kayıtlı değilse giriş sayfasına gidecek. Kullanıcıya kayıtlı veri varsa rowcount() 1
            olur.
            {
                echo '<b style="color:red"><br>Kullanıcı kayıtlı değil !<br> Kullanıcı veya şifre hatalı
            girildi !<br><br>Yönlendiriliyorsunuzuz... </b>';
                echo '<meta http-equiv="refresh" content ="5;URL=http://localhost/DAK/giris.php">'; // giriş başarılı bir şekilde
            gerçektirilemedi. giris.php git.
            }
            else if($sorgula -> rowcount()>=0)// Kullanıcı kayıtlıysa, eşyalar üzerinde işlem yapacağı sayfaya gidecek.
            {
                $izin=1; // Kullanıcı esya menüsüne ulaşabilir oldu.
                $_SESSION["test"]=$izin;
                echo '<b style="color:green">Giriş başarılı <br>Yönetim sayfasına yönlendiriliyorsunuzuz...</b>'; // kayıt başarılı bir şekilde
            gerçektirildi.
                echo '<meta http-equiv="refresh" content ="3;URL=http://localhost/DAK/crudesya.php">'; // Eşya kontrol paneline gidiyor. 3sn
            içinde
            }
        ?>
        <br><br>
        <a class="" href="http://localhost/DAK/index.php">Ana Menü<a/> <!-- Anasayfaya geri dönüş. -->
    </form>
```

# KAYNAK KODLAR

## gncl.php

```
<link rel="stylesheet" href="cs.css"> <!-- CSS kullanabilmek için. -->
<?php
    session_start();
    $test=$_SESSION["test"];

    if($test ==0)
    {
        ?>
        <form id="form" actin="" method=""> <?php
        echo '<b style="color:red"><br> ! Yetkisiz giriş !<br><br>Yönlendiriliyorsunuz... </b>';
        echo '<meta http-equiv="refresh" content ="1;URL=http://localhost/DAK/giris.php">';
        session_destroy();
        die; ## Eğer giriş yapmadan gelmeye çalışırsa burdan sonrasına erişemeyecek. girissql.php deyken denerse giremeyecek.
        ## Eğer giriş yapmadan gelmeye çalışırsa burdan sonrasına erişemeyecek. giris.php deyken denerse giremeyecek.
        ?></from><?php
    }
    require("sqlconnect.php");
    $esya_kodu=$_GET['esya_kodu'];
    $_SESSION["esya_kodu"]=$esya_kodu;

?>

<html>

    <head>

        <meta charset="UTF-8" />
        <title>UPDATE</title>

    </head>

    <body>
    GÜNCELLEMEK İSTEDİĞİNİZ EŞYANIN ESKİ BİLGİLERİ</h2>
    <table width="90%" border="10px">
        <tr>
            <th>Esya kodu</th>
            <th>Esya adı</th>
            <th>Konumu</th>
            <th>Adedi</th>
            <th>Markası</th>
            <th>Durumu</th>
        </tr>
        <?php

        $sorgula = $dak -> query("SELECT *FROM esya Where esya_kodu=$esya_kodu");

        while($row=$sorgula ->fetch(PDO::FETCH_ASSOC))
        {
            Print "<tr>";
            Print '<td align="center">'. $row['esya_kodu'] . "</td>";
            Print '<td align="center">'. $row['esya_adi'] . "</td>";
            Print '<td align="center">'. $row['konum'] . "</td>";
            Print '<td align="center">'. $row['adet'] . "</td>";
            Print '<td align="center">'. $row['marka'] . "</td>";
            Print '<td align="center">'. $row['durum'] . "</td>";
            Print "</tr>";
        }

        ?>

    </table>

    <form name="form" method="POST" action="gnclsql.php">
        Durum 1 , 2 ve 3 olarak girilmelidir <br>[ 1 : yeni | 2 : yıpranmış | 3 : arızalı]</br>
        <br>Boş Alan Bırakmayınız<br>("Marka alanı isteğe bağlı boş bırakılabilir)</br>
        <input type="text" name="esya_adi" placeholder = "Esya adı" /><br>
        <input type="text" name="konum" placeholder = "Konumu" /><br>
        <input type="text" name="adet" placeholder = "Esya adedi" /><br>
        <input type="text" name="marka" placeholder = "Eşyanın markası" /><br>
        <input type="text" name="durum" placeholder = "Eşyanın durumu" /><br>
        <input type="submit" value="Eşyayı Güncelle"/> <a>

        <br><br><br> <a class="" href="http://localhost/DAK/crudesya.php">Yönetici Paneli<a/>
        <br><br> <a class="" href="http://localhost/DAK/index.php">Ana Menü<a/>

    </form>

</body>

</html>
```

# KAYNAK KODLAR

## gnclsql.php

```
<?php
    session_start();
    @$esya_kodu=$_SESSION["esya_kodu"]; // esya kodu burda lazım.
?>

<html>

    <head>

        <meta charset="UTF-8" />    <!-- Türkçe karakter için. -->
        <title>UPDATE</title>
        <link rel="stylesheet" href="cs.css">    <!-- CSS kullanabilmek için. -->

    </head>

    <body>

        <?php

            require("sqlconnect.php"); // veri tabanına bağlanılıyor.
            @$Esysaadi = $_POST["esya_adi"];
            @$Konum = $_POST["konum"];
            @$Adet = $_POST["adet"];
            @$Marka = $_POST["marka"];
            @$Durum = $_POST["durum"];
            if($Esysaadi == "" || $Konum == "" || $Adet == "" || $Durum == "") // Eğer gerekli alan(lar)dan bazıısı boşsa.
            {
                <form id="form" action="" method="post"><?php
                    echo '<b style="color:red"><br> ! Boş Alan Bırakmayınız !<br><br>Yönlendiriliyorsunuz... </b>';
                    echo '<meta http-equiv="refresh" content ="3;URL=http://localhost/DAK/crudesya.php">';
                    die;?>
                </form>

                <?php
                }
                $Guncelle = $dak ->prepare("UPDATE esya SET esya_adi=?, konum=?, adet=?, marka=?, durum=? where esya_kodu=?");
                $Guncelle ->execute(array($Esysaadi,"$Konum","$Adet","$Marka","$Durum","$esya_kodu"));
                ?>

                <form id="form" action="" method="post"><?php
                    echo '<b style="color:green">Güncelleme başarılı <br>Yönlendiriliyorsunuz...</b>';
                    echo '<meta http-equiv="refresh" content ="2;URL=http://localhost/DAK/crudesya.php">';?>

                </form>

            <br><br>    <a class="" href="http://localhost/DAK/index.php">Ana Menü<a/>

        </body>

    </html>
```

# KAYNAK KODLAR

## index.php

```
<?php // Hiçbir yerden direkt olarak eşya menüsüne izinsiz erişememesi için.
session_start();// SESSION ları kullanabilmek için.
$izin=0; // izin 0 yapılıyor yani giriş izni şimdilik yok.
$_SESSION["test"]=$izin; // SESSION kullanarak başka sayfalara izin durumu taşıyor.
?>
<html>

    <head>
        <title>ANA MENÜ</title>
        <link rel="stylesheet" href="cs.css"> <!-- CSS kullanabilmek için. -->
    </head>

    <body>

        <h2>Giriş yapın veya kayıt olun</h2>

        <form id="form" action="giris.php" method="post"> <!-- form yardımıyla post metodunu kullanarak butona basıldığında giris.php ye git -->
            <button type="submit" class="btn" >GİRİŞ YAP</button> <!-- Giriş yap butonu. btn etiketli css stilini kullan -->
        </form>

        <form id="form" action="kayit.php" method="post"> <!-- form yardımıyla post metodunu kullanarak butona basıldığında kayit.php ye git -->
            <button type="submit" class="btn" >&nbsp;&nbsp;&nbsp;KAYIT OL</button> <!-- Kayıt ol butonu. btn etiketli css stilini kullan. &nbsp;&nbsp;& boşluk ekler.-->
        </form>

        <a href="bilgi.php">SİSTEM HAKKINDA BİLGİ</a> <br/>

    </body>

</html>
```

# KAYNAK KODLAR

## kayit.php

```
<?php // Hiçbir yerden direkt olarak esya menüsüne izinsiz erişememesi için.
session_start();// SESSION ları kullanabilmek için.
$izin=0; // izin 0 yapılıyor yani giriş izni şimdilik yok.
$_SESSION["test"]=$izin;// SESSION kullanarak başka sayfalara izin durumu taşıyor.
?>
<html>

    <head>
        <meta charset="UTF-8" /> <!-- Türkçe karakter için. -->
        <title>Üye Kayıt</title>
        <link rel="stylesheet" href="cs.css"> <!-- CSS kullanabilmek için. -->
    </head>

    <body>
        <form id="form" method="POST" action="kayitsql.php"> <!-- Kayıt ol butonuna tıklanınca kayitsql.php ye gidecek. -->
            <input type="text" name="kullanici" placeholder="Kullanıcı adınız" /><br> <!-- placeholder form alanlarındaki silik
yazılar için -->
            <input type="password" name="sifre" placeholder="Şifreniz" /><br> <!-- sifre password formatında -->
            <input type="password" name="sifre2" placeholder="Şifrenizi tekrar girin" /><br> <!-- sifre password formatında -->
            <input type="text" name="mail" placeholder="Mailiniz" /><br> <!-- mail text formatında -->
            <input type="submit" class="btn" value="Kayıt ol"/> <a> <!-- Bilgiler veri tabanına kayıt olmak üzere kayitsql.php ye
gidiyor.-->

        </form>
        <br><br><br> <a class="" href="http://localhost/DAK/index.php">Ana Menü<a/> <!-- Anasayfaya geri dönüş. -->
    </body>

</html>
```

# KAYNAK KODLAR

## kayitsql.php

```
<?php // Hiçbir yerden direkt olarak esya menüsüne izinsiz erişememesi için.
    session_start(); // SESSION ları kullanabilmek için.
    $izin=0; // izin 0 yapıyor yani giriş izni şimdilik yok.
    $_SESSION["test"]=$izin; // SESSION kullanarak başka sayfalara izin durumu taşıyor.
?>
<html>

    <head>

        <meta charset="UTF-8" /> <!-- Türkçe karakter için. -->
        <title>Giriş Sayfası</title> <!-- Sayfa başlığı -->
        <link rel="stylesheet" href="cs.css"> <!-- CSS kullanabilmek için. -->

    </head>

    <body>

        <?php

            require("sqlconnect.php"); // include yerine require kullanıldı çünkü hatayla karşılaşıldığında işleme devam edilmesin. Veri tabanına

bağlantı kuruluyor.

            // POST Metodu ile kayit.php deki veriler kayitsql.php dosyasına çağrılıyor.
            @$Kullanici = $_POST["kullanici"];
            @$Sifre = $_POST["sifre"]; // Küçük büyük harf duyarlı olduğundan böyle tanımlamalar yapıldı.
            @$Sifre2 = $_POST["sifre2"];
            @$Mail = $_POST["mail"];
            // Veri çağırma bitti.

        ?>

        <form action="kayit.php" method="post"> <!-- Hata olduğunda kullanıcı tamam butonuna basarsa kayıt ekranına yönlendirilecek. kayit.php ye gidecek. --
        >

            <?php
            if($Kullanici == "" || $Sifre == "" || $Sifre2 == "" || $Mail == "") // Eğer gerekli alanlardan bazıları boşsa.
            {
                echo '<b style="color:red">Boş Alan Bırakmayınız ! </b>';?>
                <br><button class="err" type="submit">Tamam</button><?php // err etiketli css stilini kullan.

            }
            else
            {

                if((!filter_var($Mail,FILTER_VALIDATE_EMAIL))) // ...@...com şeklinde girilmesi için.
                {
                    echo '<b style="color:red">Mail Formatı Hatalı ! </b>';?>
                    <br><button class="err" type="submit">Tamam</button><?php // err etiketli css stilini kullan.

                }
                else
                {

                    if($Sifre != $Sifre2)// Eğer şifreler uyuşmazsa buradan tekrar kayit.php ye gidecek.
                    {
                        echo '<b style="color:red">Girdiğiniz şifreler uyuşmuyor ! </b>';?>
                        <br><button class="err" type="submit">Tamam</button><?php // err etiketli css stilini kullan.

                    }
                    else
                    {

                        $sorgula = $dak -> prepare("SELECT *FROM uyeler WHERE kullanici = ? OR mail = ?"); // Kullanıcı adı

                        $sorgula ->execute(array($Kullanici,$Mail)); // kullanici ler ve mail ler veri tanından kontrol için

                        if($sorgula -> rowCount() > 0)// Kullanıcı veya mail önceden kayıtlıysa,kayıt yapacağı kayit.php

                        {
                            echo '<b style="color:red">Kayıt başarılı olmadı !<br>E-posta veya kullanıcı adı önceden

kullanılmış!<br><br>Yönlendiriliyorsunuz... </b>'; // kayıt başarılı bir şekilde gerçekleştirilemedi.
                            echo '<meta http-equiv="refresh" content ="5;URL=http://localhost/DAK/kayit.php">';

                        }
                        else
                        {

                            $kayityap = $dak -> prepare("INSERT INTO uyeler SET kullanici = ?, sifre = ?, mail = ?");

                            $sekle = $kayityap ->execute(array($Kullanici,$Sifre,$Mail)); // Kaydedilecek veriler

                            if($sekle)
                            {
                                echo '<b style="color:green">Kayıt başarılı <br>Giriş sayfasına

                                echo '<meta http-equiv="refresh" content

                            }
                            else
                            {

                                echo '<b style="color:red">Kayıt başarılı olmadı !</b>'; // kayıt başarılı bir

                                echo '<meta http-equiv="refresh" content

                            }

                        }

                    }

                }

            }

        </form>

        <br><br>
        <a class="" href="http://localhost/DAK/index.php">Ana Menü<a>

    </body>

</html>
```



# KAYNAK KODLAR

## sil.php

```
<?php
    require("sqlconnect.php");
    $esya_kodu= $_GET['esya_kodu']; // silinecek veri alınıyor.
    $sil = $dak ->prepare("DELETE FROM esya WHERE esya_kodu=?"); // istenilen sütundaki veriler siliniyor.
    $temizle = $sil ->execute(array($esya_kodu));
    header("location: crudesya.php"); // silinme bitti.
?>
```

## sqlconnect.php

```
<html>

    <head>

        <meta charset="UTF-8" />    <!-- Türkçe karakter için. -->

        <link rel="stylesheet" href="style.css"><!-- CSS kullanabilmek için. -->

    </head>

</html>

<?php

    try

    {

        // Veri tabanına bağlanırken karşılaşılan hataların tespit edilmesi için.

        $dak = new PDO("mysql:host=localhost; dbname=dak; charset=utf8", "root", ""); //

        $dak -> setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

        $dak -> exec("set names utf8");

    }

    catch(PDOException $hata)

    {

        print $hata -> getMessage(); // Hata yakalandığında hata mesajı yazdırılacak.

    }

?>
```