



**Bayındır Sağlık Grubu
Teknik Mülakat Raporu**

Ayşegül Yanık

Web Servis Dosya İşlemleri

**ANKARA, Haziran
2020**

Ayşegül YANIK
Teknik Mülakat Raporu

Proje Taslağı

Projede üç temel görevin gerçekleştirilmesi hedeflenmiştir. İlk olarak, üye girişiyle oturum açma ve kimlik bilgileriyle kullanıcıya özgü bağımsız işlemlerin yürütülmesine olanak veren bir giriş/oturum açma sayfası tanımlanacaktır. İkinci ve üçüncü aşamalarda ise kullanıcıya, sunucunun kendine ayrılmış dosya sistemi kısmına veri yükleme özelliğinin sunulduğu ve yüklediği dosyaları görüntüleyebileceği veya seçtiği dosyayı indirebileceği bir arayüz sunulacaktır.

Proje detayları raporda mevcuttur.

ANKARA, Haziran
2020

*“Kod, espri gibidir.
Açıklamak zorundaysanız kötüdür.”
(Cory House)*

ÖZET

Projede; C# ve ASP.Net kullanarak oturum açma, ardından sunucuya dosya yükleme ve yüklenen dosyalardan seçerek indirme operasyonları gerçekleştirilmiştir.

Projenin çözüm dosyası Visual Studio üzerinde, veri tabanı yapısı ise SQL Server Management Studio üzerinde geliştirilmiştir, kendi bilgisayarınızda çalıştırabilmeniz için muadil program kullanmanızı öneririm.

Kurulumları gerçekleştirdikten sonra raporda anlatıldığı şekilde veritabanında kullanıcı anahtarı ve şifrelerini tutan bir tablo oluşturulmalı ve proje ile veri tabanı arasında bağlantı konfigürasyonu yapılandırılmalıdır. Raporda nasıl yapılacağına dair anahtar değişiklikler mevcuttur.

Oturum açma arayüzünde başarılı eşleşen bilgileri giren kullanıcıya dosya yükleme ve yüklü dosyalardan seçip, seçtiği dopsayı kendi bilgisayarına indirme olanağı sunulur.

Anahtar Kelimeler: Web Servis, Dosya İşlemleri, Oturum Açma, MS Visual Studio, MS SQL Server Management Studio, SQL Express, *C#, ASP.Net*

1 KURULUM VE GEREKSİNİMLER

İlk olarak kullanacağınız veritabanı sistemini çalışma alanınıza indirin. Bu projede SQL Express kullanıldı. Linkten ulaşabilirsiniz: <https://www.microsoft.com/en-us/download/details.aspx?id=55994>

Projenin çalışması için ön koşul veritabanında (en azından) kullanıcıya ait kullanıcı adı ve şifre bilgileri girilmiş olmalı. Daha sonra veritabanı ile kendi ASP.Net çözüm alanınızı birlikte çalışır hale getirmelisiniz.

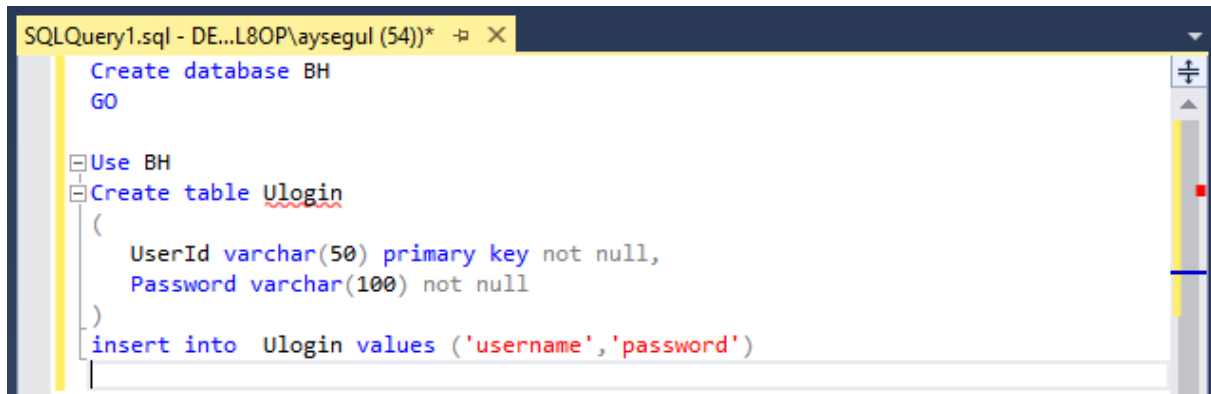
Kullanılan entegre geliştirme ortamı Visual Studio 2013. Daha yeni versiyonları da kullanabilirsiniz.

2 VERİ TABANI İŞLEMLERİ

2.1 Kullanıcının Oturum Açma Bilgilerinin Tutulması

İlk olarak bir veritabanı oluşturup, içinde kullanıcıların giriş yapmada kullanacakları parametreyi (kullanıcı adı veya mail adresi) ve kullanıcının şifre bilgilerinin ilişkilendirilerek tutulduğu bir tablo oluşturulmalıdır.

Figura 1: Veri Tabanı Oluşturma



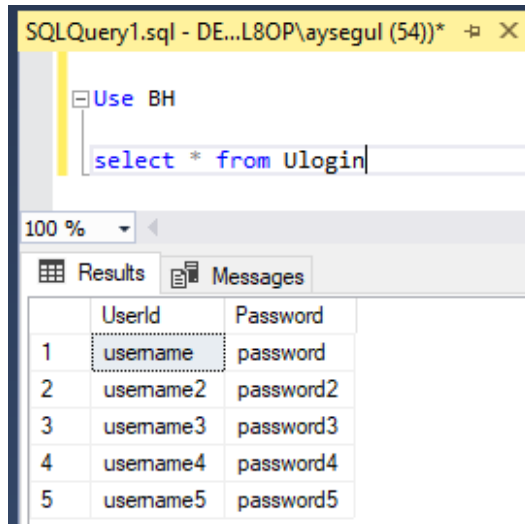
```

SQLQuery1.sql - DE...L8OP\aysegul (54))* -> X
Create database BH
GO

Use BH
Create table Ulogin
(
    UserId varchar(50) primary key not null,
    Password varchar(100) not null
)
insert into Ulogin values ('username', 'password')
  
```

Oluşturulan kullanıcı girdisinin değerleri UserId birincil anahtarı ile özgüllendirilmiştir. Tablo bir SQL sorgusuyla çağrılacak olursa, elde edilecek tablo aşağıdaki gibi görünecektir.

Figura 2: Kullanıcı Giriş Bilgileri Tablosu Sorgulama



```

SQLQuery1.sql - DE...L8OP\aysegul (54))* -> X

Use BH

select * from Ulogin
  
```

	UserId	Password
1	username	password
2	username2	password2
3	username3	password3
4	username4	password4
5	username5	password5

2.2 Web Uygulaması ile SQL Sunucusunun Bağlanması

Konfigürasyon dosyasına yapacağımız küçük eklemeyle veri tabanını kullanılabılır hale getirdikten sonra sayfanın c# yansıması üzerine bağlantı verisi tanımlamamız gerekiyor. Bağlantı verisini tanıtmak için önce iki kapsam ekliyoruz. Son olarak da, butona çift tıklayıp oluşan metodun içinde sunucu-alıcı bağlantısını kuruyoruz.

Figura 3: Web.config dosyasında konfigürasyon etiketinin içine eklenen kısım

```
<!-- CONFIGURATION REQUIREMENTS! -->
<connectionStrings>
  <add name="mycon" connectionString="server=DESKTOP-8LHL80P\SQLEXPRESS;database=BH;integrated security=true;" />
</connectionStrings>
```

Figura 4: Bağlantı verisi tanımlanacak olan .aspx.cs dosyasına eklenecek kapsam

```
using System.Data.SqlClient;
using System.Configuration;
```

Figura 5: Bağlantı nesnesi yaratma

```
SqlConnection con = new SqlConnection(ConfigurationManager.ConnectionStrings["mycon"].ToString());
```

Figura 6: Butona eklenen aksiyon kodu

```
protected void Button1_Click(object sender, EventArgs e)
{
    SqlConnection con = new SqlConnection(ConfigurationManager.ConnectionStrings["mycon"].ToString());

    try{
        string uid = TextBox1.Text;
        string pass = TextBox2.Text;
        con.Open();
        string qry = "select * from Ulogin where UserId='" + uid + "' and Password='" + pass + "'";
        SqlCommand cmd = new SqlCommand(qry, con);
        SqlDataReader sdr = cmd.ExecuteReader();
        if(sdr.Read())
        {
            Label4.Text = "Login Sucess";
        }
        else
        {
            Label4.Text = "UserId and Password pair does not match, try again please.";
        }
        con.Close();
    }
    catch(Exception ex)
    {
        Response.Write(ex.Message);
    }
}
```

3 OTURUM AÇMA İŞLEMLERİ

Oturum açma işlemi sayesinde her kullanıcının yaptığı işlemin kendine has ve özel/gizli yürütülebilmesi hedeflenir. Tanımladığımız bağlantı nesnesi sayesinde, sql sunucusu ile uygulamamız ilişki içinde çalışacaktır. Tanımlı kişi adı ve şifre ikilisi veri tabanında mevcutsa, tarayıcıda kullanıcıya oturum açma işleminin başarıyla gerçekleştiği bilgisi verilir ve kullanıcı kendi yüklediği dosyaları görebildiği, seçtiği dosyayı indirebildiği veya yeni dosyalar yükleyebildiği bir arayüze yönlendirilir.

Figura 7: Giriş işlemi başarılı geri bildirimi

Log In

UserId :

Password :

Login Sucess

Eğer kullanıcı; veri tabanındaki mevcut eşleşen kullanıcı adı ve şifre ikilisini hiç girmeden veya yanlış girerek formu ibraz ederse, kullanıcı bir sonraki sayfaya yönlendirilmeyecek, kendisinden yeniden kullanıcı adı şifre ikilisini girmesi istenecektir.

Figura 8: Giriş işlemi başarısız geri bildirimi

Log In

UserId :

Password :

UserId and Password pair does not match, try again please.

4 DOSYA İŞLEMLERİ

Başarıyla oturum açan kullanıcı, dosyalarını yönetebileceği bir arayüze yönlendirilecektir. Bu arayüz kullanıcıya özgüdür, kullanıcının biri diğerinin verilerine erişememeli. Bu motivasyonla sunucuda tutulan dosyalar ayrılmalı veya etiketlenmelidir.

Eğer kullanıcılarım arasında bir kısıtlamaya yol açan roller veya izinler hiyerarşisi mevcutsa, bu ilişkiyi kurmada ayrıca bir modelleme aşaması gerekecekti. Ancak şu durumda, kullanıcı oturum açabildiği sürece ortak bir sayfaya yönlendirilecek.

Kullanıcıyı sayfalar arasında yönlendirmenin iki kullanışlı yolu var. İlki `Response.Redirect()` ve diğeri `Server.Transfer()` metotlarını kullanarak yönlendirme gerçekleştirmek. İkisi arasındaki temel fark, sayfanın aynı bağlantı üzerinden mi çalışacağı yoksa yeni bir bağlantıya mı yönlendirileceği ile ilgilidir. Bu uygulamada, `Response.Redirect()` kullandım çünkü kullanıcıyı yeni bir bağlantı adresine yönlendirerek işlemlerin sürmesini istedim.

4.1 DOSYA YÜKLEME

Dosyalarını yönetebileceği arayüze yönlendirilen kullanıcı, yüklemek istediği dosyayı seçtikten sonra dosyanın birtakım koşulları sağlayıp sağlamadığı incelenmelidir. Örneğin; bu projede, dosya boyutunun 2 MB ve daha küçük boyutta olması şartı aranmaktadır. Ancak örnekler çoğaltılabilir. Kullanıcının hangi tipte veya ne tür uzantılı bir dosya yükleyeceğini tartarak bu yükleme işlemine müdahale edilebilir. Sunucunun ve uygulamanın güvenliği/güvenilirliği için yükleme ve indirme işlemlerinin mümkün olan her yönden denetlenmesi elzemdir. Bu denetimi sağlamak adına incelenecek verinin, kullanıcı mahremiyetini ihlal etmemesi önemli. Verinin boyutu veya dosya türüne bakmak gereklidir ancak dosyanın içeriğini ve semantiğini incelemek kullanıcı gizliliğine yapılacak bir saldırı olacağından bu tür işlemler içeren yetkiler, izinsiz eklenmemesi gereken özelliklerdir.

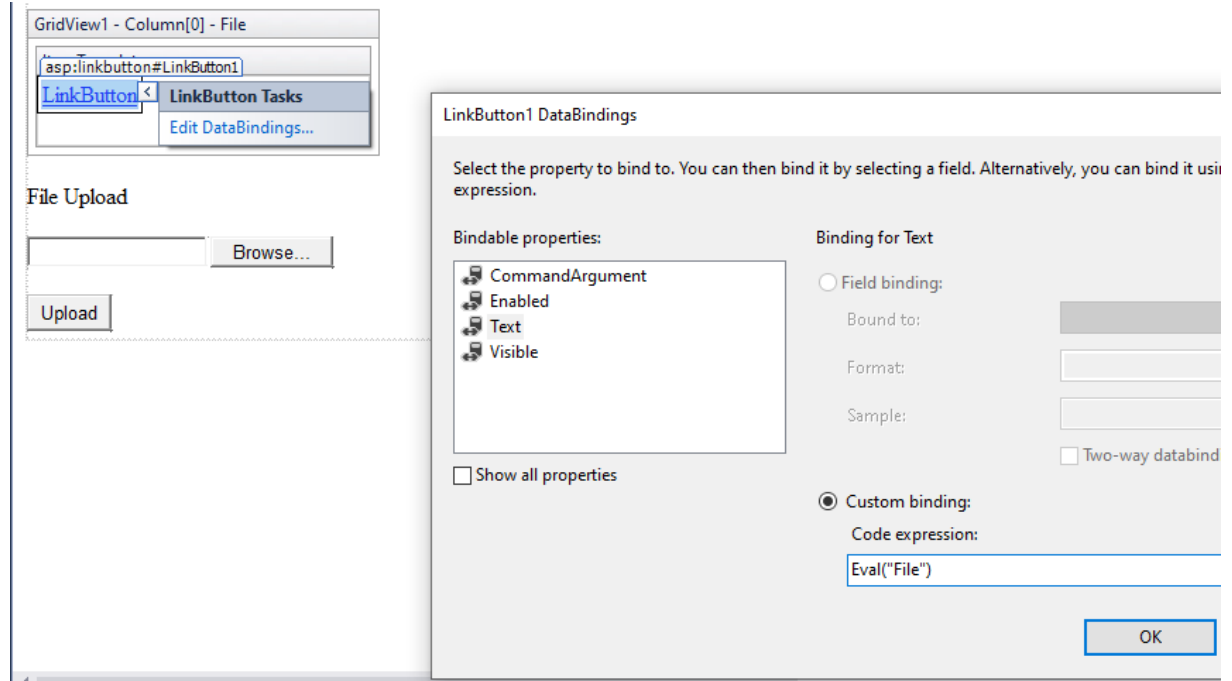
Figura 9: Dosya yükleme

```
//FIND SERVER PATH TO DECIDE WHERE TO SAVE
string path = Server.MapPath("/images/");
FileUpload1.SaveAs(path + FileUpload1.FileName);
```

4.2 DOSYA İNDİRME

Kullanıcının indirebileceği dosyaları görüntülediği arayüzü oluştururken öyle bir araç kullanılmalı ki indirilebilecek bir veri mevcut değilse, tablo oluşturulmamalı. Ancak kullanıcı bir dosya yüklediği anda, tabloya bu yüklenen veri indirilebilir bir eleman olarak eklenmelidir. Bu isteri sağlaması için tabloya *DataBinder.Eval* yapısını ekledim.

Figura 10: Tabloyu veriye bağlama aracı tanımlama



Her yeni yüklemeden sonra ve sayfa ilk yüklenirken çağırmak üzere bir tablo yaratma metodu oluşturdum.

Figura 11: Tabloyu yükleme ve güncelleme metodu

```
public void Create_Table()
{
    DataTable dt = new DataTable();
    dt.Columns.Add("File", typeof(string));
    dt.Columns.Add("Size", typeof(string));
    dt.Columns.Add("Type", typeof(string));

    //DO FOR EACH FILE IN THE DIRECTORY
    foreach (string strFile in Directory.GetFiles(Server.MapPath("/images/")))
    {
        FileInfo fi = new FileInfo(strFile);

        //add overloaded method
        dt.Rows.Add(fi.Name, fi.Length, fi.Extension);
    }

    GridView1.DataSource = dt;
    GridView1.DataBind();
}
```

YARARLANILAN KAYNAK VE REFERANSLAR

How to create login page in ASP.Net Web Application Using C# And SQL Server :
<https://www.c-sharpcorner.com>

Asp.Net File Upload Kullanımı :
<https://www.youtube.com/watch?v=reP8fpCWs7E>

How to upload and download files using asp net and c# :
<https://www.youtube.com/watch?v=-fkdrGFiael>

Proje kaynak kodunu bu bağlantı üzerinden indirebilirsiniz :
<https://github.com/AysegulDemirtas/WebFileManagement>