

ANKARA ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ



BLM4531 Ağ Tabanlı Teknolojiler ve Uygulamalar

Rapor

Özel Ders Rezervasyon Sistemi

Demir Mesut ÖZTAŞ

19290760

Ocak, 2023

BLM4531 Ağ Tabanlı Teknolojiler ve Uygulamalar dersi kapsamında Microsoft tarafından geliştirilen Asp.Net teknolojisi ile “Özel Ders Rezervasyon Sistemi” isminde veritabanı içeren bir web uygulaması geliştirilmiştir. Proje konusu olan “Özel Ders Rezervasyon Sistemi”; ders almak isteyen öğrenciler ile ders vermek isteyen öğretmenler için rezervasyon programı sağlayan bir web uygulamasıdır. Sistemin kullanılabilmesi için her iki tipteki (öğrenci, öğretmen) kullanıcının sisteme kayıt olması gerekmektedir. Özel ders almak isteyen öğrenci sisteme kayıt olduktan sonra sisteme kayıtlı öğretmenlerin bilgilerin görüntülediği sayfadan istediği dersi ve zamanı seçerek bir rezervasyon kaydı oluşturmaktadır. Oluşturulan rezervasyon kaydı, hem öğretmenin hem de öğrencinin ilgili kontrol panelinde görüntülenmektedir. Böylelikle öğretmen, vereceği özel dersin programını kolaylıkla takip edebilirken öğrenci ise almak istediği özel derslerle ilgili kurs listesinde uygun öğretmeni seçerek kendine uygun bir özel ders programı oluşturabilmektedir.

Proje; .net framework ve Microsoft Visual Studio geliştirme ortamı kullanılarak geliştirilmiştir. Uygulamanın geliştirme süreçleri; **önyüz (fronted)**, **arkayüz (backend)**, **veritabanı** ve **api** başlıkları altında gruplandırılarak anlatılmıştır.

1. Önyüz (Frontend)

Geliştirilen web uygulamasının önyüz tasarımları; uygulamayı geliştirmeden önce tasarlanan prototipe göre tasarlanmıştır. Bu kapsamda; tasarlanan uygulamanın ana sayfası, Resim 1.1’de gösterilmekte olup bu sayfada en çok tercih edilen ders ve öğretmen bilgilerinin istatistiksel olarak gösterildiği bir tablo (api vasıtasıyla) ile öğrencinin giriş yapacağı veya kayıt olacağı sayfaya yönlendirilecek giriş/kayıt paneli yer almaktadır. Eğer öğrenci kayıt olmak istiyorsa bu giriş panelinde yer alan “Hesap aç” sekmesine kullanılarak öğrenci kayıt ekranına ulaşabilmektedir. Bu sayfanın ekran görüntüsü Resim 1.2’de gösterilmiştir.

Resim 1.1 Anasayfa

Özel Ders

Eğitmen

En Çok Tercih Edilenler

Ders Adı	Öğretmen Adı	Öğretmen Soyadı	Ders Tipi	Toplam
Fizik	Murat	Sel	Online	5
Felsefe	Gülsah	Savsas	Online	4
Tarih	Sinasi	Odabasi	Online	2
Biyoloji	Emine	Alptekin	Online	2
Biyoloji	Emine	Alptekin	Yüz yüze	1
İngilizce	Vecdi	Arslansoyu	Online	1

Oturum aç

Yeni kullanıcı ? [Hesap aç](#)

E-posta

Şifre

Giriş

Resim 1.2 Öğrenci Kayıt Ekranı

[Özel Ders](#) [Eğitmen](#)

Öğrenci Kayıt Ekranı

Adı

Soyadı

E-mail

Şehir

Şifre

Şifre Tekrar

Kaydet

Eğitmenin bu uygulamaya giriş yapabilmesi veya yeni bir eğitmenin bu uygulamaya kayıt olabilmesi için öğrenci giriş/kayıt panelinden ayrı olarak ana sayfanın üst menü çubuğunda “Eğitmen” adında bir alan oluşturulmuştur. Eğitmen sayfasının görüntüsü Resim 1.3’te gösterilmekte olup bu sayfa vasıtasıyla eğitmen kullanıcısı gerekli bilgiler girerek uygulamaya giriş yapabilmektedir.

Resim 1.3 Eğitmen Giriş Sayfası

[Özel Ders](#) [Eğitmen](#)

Eğitmen Giriş

Oturum aç ? [Hesap aç](#)

E-posta

Şifre

Giriş

Eğer kullanıcı kayıtlı bir eğitmen değilse ve kayıt olmak istiyorsa “Hesap aç” sekmesini kullanarak eğitmen kayıt ekranına ulaşabilmektedir. Bu sayfanın ekran görüntüsü Resim 1.4’te yer almaktadır.

Resim 1.4 Eđitmen Kayıt Ekranı

[Özel Ders](#) [Eđitmen](#)

Eđitmen Kayıt Ekranı

Adı	<input type="text"/>
Soyadı	<input type="text"/>
E-mail	<input type="text"/>
Yaş	<input type="text"/>
Şehir	<input type="text"/>
Eđitim	<input type="text"/>
Branş	<input type="text"/>
Açıklama	<input type="text"/>
Şifre	<input type="password"/>
Şifre Tekrar	<input type="password"/>

Dosya Seç





Dosya seçilmedi

Kaydet

Sisteme kayıtlı olan bir öğrenci uygulamaya giriş yaptığıında, sistemde listelenen derslerin görüntülendiđi sayfaya yönlendirilmektedir. Resim 1.5'te derslerin listelendiđi sayfa yer almakta olup bu sayfada, kayıtlı tüm eđitmenlerin adının, soyadının, fotoğrafının, branş ve şehir bilgilerinin yer aldığı bir veri tablosu yer almaktadır. Aynı zamanda bu ders tablosundan seçeceđi özel ders için sağ tarafta rezervasyon oluşturabileceđi bir panel bulunmaktadır.

Resim 1.5 Ders Listesi Sayfası

[Özel Ders](#) [Kontrol Paneli](#)

	Fotoğraf	Ders	Ad	Soyad	Şehir
Seçiniz		Fizik	Murat	Sel	Gaziantep
Seçiniz		Biyoloji	Emine	Alptekin	Malatya
Seçiniz		Felsefe	Gülsah	Savsa	Ankara
Seçiniz		Tarih	Sinasi	Odabasi	Kayseri

Rezervasyon Yap

Ders Tipi [Seçiniz](#)

Tarih gg . aa . yyyy

Saat [Seçiniz](#)

Rezervasyon Yap

Kullanıcı, bu sayfanın (Resim 1.5) menü çubuğunda yer alan “Kontrol Paneli” sekmesi ile oluşturduğu rezervasyonların bulunduğu sayfaya geçiş yapabilmektedir. Resim 1.6'daki görsel, uygulamanın veritabanında kayıtlı bir kullanıcının kontrol panel sayfasını göstermektedir. Bu sayfa rezervasyon bilgilerinin yer aldığı bir tablo, tabloda yer alan rezervasyonları silebilmesi için “buton” ve bu tabloda arama yapabilmek için oluşturulmuş bir “text” alanı mevcuttur. Ayrıca üst menüde yer alan “Ders Listesi” sekmesi vasıtasıyla tekrar derslerin listelendiği sayfaya geçiş yapabilmektedir.

Resim 1.6 Öğrenci Kontrol Paneli

[Özel Ders](#) [Ders Listesi](#) [Elif Duru Savsa](#)

Ders Adı	Ders Tipi	Tarih	Saat	Öğretmen Adı	Öğretmen Soyadı	E-posta
Fizik	Online	2023-01-03	12.00	Murat	Sel	muratsel@yandex.com Sil
Tarih	Online	2023-01-05	15.00	Sinasi	Odabasi	sinasi@gmail.com Sil
Felsefe	Online	2023-01-09	14.00	Gülsah	Savsa	gulsah@gmail.com Sil
Biyoloji	Online	2023-01-09	13.00	Emine	Alptekin	emine@gmail.com Sil
Felsefe	Online	2023-01-25	14.00	Gülsah	Savsa	gulsah@gmail.com Sil

Diğer taraftan uygulamaya bir eğitmen giriş yaptığında otomatik olarak eğitmen kontrol paneline yönlendirilmektedir. Resim 1.7'de Eğitmen kontrol paneli sayfası yer almaktadır. Bu sayfa, öğrenci kontrol paneline benzer bir şekilde oluşturulmuş olup bu sayfada, özel ders için rezervasyon kaydı oluşturan öğrencilere ait bilgilerin tutulduğu bir tablo yer almaktadır.

Resim 1.7 Eđitmen Kontrol Paneli

Özel Ders		Eđitmen Kontrol Paneli				Murat Sel	
		Ara Temizle					
Ders Tipi	Tarih	Saat	Öđrenci Adı	Öđrenci Soyadı	E-posta		
Online	2023-01-03	12.00	Elif Duru	Savsa	duru@gmail.com	Sil	
Online	2023-01-08	11.00	Toygar	Kaan	toygar@gmail.com	Sil	
Online	2023-01-12	14.00	Aleyna	Demir	aleyna@gmail.com	Sil	
Online	2023-01-03	10.00	Mesut	Demir	mesut@gmail.com	Sil	
Online	2023-01-25	15.00	Aleyna	Demir	aleyna@gmail.com	Sil	

2. Veritabanı

Uygulamada kayıt altında tutulan veriler için gerekli veritabanı işlemleri MSSQL kullanılarak yapılmıştır. “course” isminde bir veritabanı oluşturulmuş ve bu veritabanı altında uygulamada ihtiyaç duyulan tablolar oluşturulmuştur. Bu doğrultuda; uygulamada iki farklı kullanıcı tipi olduğu için “student” ve “teacher” isminde tablolar ile rezervasyon kayıtlarının tutulduğu “reservation” isimli bir tablo ve ders saat bilgilerinin tutulduğu “hour” isimli tablo oluşturulmuştur.

Resim 2.1 Veritabanı Tablosu

student	reservation
studentId	reservationId
name	studentId
surname	teacherId
email	date_choose
city	hour
password	lessonname
password_again	lessontype
teacher	hour
teacherId	hourId
name	saat
surname	
email	
age	
city	
education	
branch	
password	
password_again	
photo	
information	

3. Arkayüz (Backend)

Arkayüzde yapılan kodlama sürecinde ilk başta uygulamanın veritabanı ile bağlantısını sağlayabilmek için gerekli kodlamalar yapılmıştır.

Resim 3.1 Veritabanı Bağlantısı

```
30
31 <connectionStrings>
32 <add name="Veritabani" connectionString="Data Source=(localdb)\MSSQLLocalDB;Initial Catalog=course;Integrated Security=True" />
33 </connectionStrings>
34
```

Veritabanı bağlantısı kurulduktan sonra kullanıcıların uygulamaya kayıt olurken girdiği bilgilerinin veritabanındaki ilgili tabloya kayıt olmasına sağlamak için kodlamalar yapılmıştır. Resim 3.2 ve 3.3'te öğrenci kayıt ekranı için kodlar yer almakta olup bu kod parçacığında kayıt ekranındaki bilgilerin veritabanına kayıt edilmesini sağlayan sql komutları yer almaktadır. Bu kod bloğunda kullanıcı bilgileri veritabanına kayıt edilmeden önce if else bloğu kullanılarak ilk önce veritabanında aynı bilgilere sahip kullanıcı olup olmadığı kontrolü yapılmaktadır. Eğer aynı bilgilere sahip bir kullanıcı yoksa "student" tablosuna kayıt edilmektedir.

Resim 3.2 Öğrenci Kayıt Ekranı Kodları-1

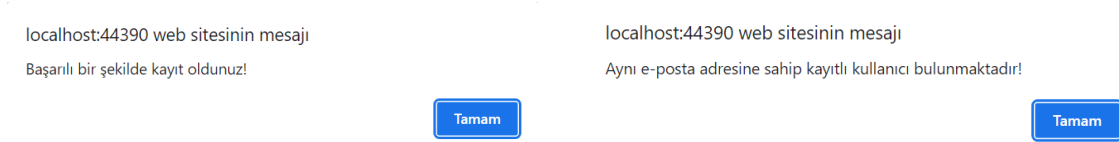
```
12 namespace CourseProject
13 {
14     public partial class trainee : System.Web.UI.Page
15     {
16         SqlConnection baglanti = new SqlConnection(WebConfigurationManager.ConnectionStrings["Veritabani"].ConnectionString);
17
18         protected void Page_Load(object sender, EventArgs e)
19         {
20         }
21
22         protected void user_kaydet_Click(object sender, EventArgs e)
23         {
24             SqlCommand sorgu3 = new SqlCommand("select studentId from student where email=@posta", baglanti);
25             sorgu3.Parameters.AddWithValue("@posta", t_txt_email.Text);
26             baglanti.Open();
27             sorgu3.Connection = baglanti;
28             SqlDataReader dr = sorgu3.ExecuteReader();
29
30             string message = "";
31             string script = "";
32             if (dr.Read())
33             {
34                 message = "Aynı e-posta adresine sahip kayıtlı kullanıcı bulunmaktadır!";
35                 script = "window.onload = function(){ alert('";
36                 script += message;
37                 script += "');}";
38                 ClientScript.RegisterStartupScript(this.GetType(), "SuccessMessage", script, true);
39                 baglanti.Close();
40                 baglanti.Dispose();
41                 dr.Close();
42                 ScriptManager.RegisterClientScriptBlock(this, typeof(Page), "redirectJS",
43                 "setTimeout(function() { window.location.replace('home3.aspx') }, 2000);", true);
44             }
45         }
46     }
47 }
```

Resim 3.3 Öğrenci Kayıt Ekranı Kodları-2

```
47         else
48         {
49             dr.Close();
50
51             SqlCommand gonder = new SqlCommand("insert into student (name,surname,email, password, password_again ) values('" +
52                 + "','" + t_txt_email.Text + "','" +
53                 + t_txt_password.Text + "','" + t_txt_password2.Text + "')", baglanti);
54
55             gonder.ExecuteNonQuery();
56             baglanti.Dispose();
57             baglanti.Close();
58
59             message = "Başarılı bir şekilde kayıt oldunuz!";
60             script = "window.onload = function(){ alert('";
61             script += message;
62             script += "')});";
63             ClientScript.RegisterStartupScript(this.GetType(), "SuccessMessage", script, true);
64
65             ScriptManager.RegisterClientScriptBlock(this, typeof(Page), "redirectJS",
66                 "setTimeout(function() { window.location.replace('home3.aspx') }, 2000);", true);
67         }
68     }
69 }
70 }
71 }
72 }
```

Ayrıca javascript kodları kullanılarak kullanıcı başarılı bir şekilde kayıt olabilmişse ekrana mesaj gösterilmesi ya da aynı bilgilere sahip kullanıcı var ise kayıt işlemi yapmayıp kullanıcıya uyarı mesajı gösterilmesi sağlanmıştır.

Resim 3.4 Uyarı Mesajları



Öğrenci kayıt ekranına benzer şekilde eğitimci kayıt ekranları oluşturulmuştur. Bu sayfaya ilişkin kodlar Resim 3.5 ve 3.6'da gösterilmiştir.

Resim 3.5 Öğitmen Kayıt Ekranı Kodları-1

```
13 public partial class teacher : System.Web.UI.Page
14 {
15     SqlConnection baglanti2 = new SqlConnection(WebConfigurationManager.ConnectionStrings["Veritabani"].ConnectionString);
16
17     protected void Page_Load(object sender, EventArgs e)
18     {
19     }
20
21
22     protected void teacher_kaydet_Click1(object sender, EventArgs e)
23     {
24
25         SqlCommand sorgu3 = new SqlCommand("select teacherId from teacher where email=@posta", baglanti2);
26         sorgu3.Parameters.AddWithValue("@posta", tr_txt_email.Text);
27         baglanti2.Open();
28
29         sorgu3.Connection = baglanti2;
30         SqlDataReader dr = sorgu3.ExecuteReader();
31
32         string message = "";
33         string script = "";
34
35         if (dr.Read())
36         {
37             message = "Aynı e-posta adresine sahip kayıtlı kullanıcı bulunmaktadır!";
38             script = "window.onload = function(){ alert('";
39             script += message;
40             script += "')}";
41             ClientScript.RegisterStartupScript(this.GetType(), "SuccessMessage", script, true);
42             baglanti2.Close();
43             baglanti2.Dispose();
44             dr.Close();
45             ScriptManager.RegisterClientScriptBlock(this, typeof(Page), "redirectJS",
46             "setTimeout(function() { window.location.replace('home3.aspx') }, 2000);", true);
47         }
48     }
```

Resim 3.6 Öğitmen Kayıt Ekranı Kodları-2

```
49     else
50     {
51         dr.Close();
52
53         SqlCommand gonder2 = new SqlCommand();
54         gonder2.Connection = baglanti2;
55
56         string fn = "~/image/" + System.IO.Path.GetFileName(fileUpload1.PostedFile.FileName);
57         string saveLocation = Server.MapPath("~/image/") + System.IO.Path.GetFileName(fileUpload1.PostedFile.FileName);
58
59         fileUpload1.PostedFile.SaveAs(saveLocation);
60
61         gonder2.CommandText = "INSERT INTO teacher (name, surname, email, age, city, education, branch, password, password_agai
62         "VALUES('" + tr_txt_adi.Text + "','" + tr_txt_surname.Text + "','" + tr_txt_email.Text +
63         "','" + tr_txt_age.Text + "','" + tr_txt_city.Text + "','" + tr_txt_education.Text + "','" +
64         tr_txt_branch.Text + "','" + tr_txt_password.Text + "','" +
65         tr_txt_password2.Text + "','" + fn + "','" + tr_txt_information.Text + "')";
66
67
68         gonder2.ExecuteNonQuery();
69
70         baglanti2.Dispose();
71         baglanti2.Close();
72
73         message = "Başarılı bir şekilde kayıt oldunuz!";
74         script = "window.onload = function(){ alert('";
75         script += message;
76         script += "')}";
77         ClientScript.RegisterStartupScript(this.GetType(), "SuccessMessage", script, true);
78
79         ScriptManager.RegisterClientScriptBlock(this, typeof(Page), "redirectJS",
80         "setTimeout(function() { window.location.replace('teacherLogin.aspx') }, 3000);", true);
81     }
82 }
83
84 }
85 }
```

Kayıt ekranları için yapılan kodlamadan sonra kullanıcıların sisteme giriş yapabilmesi için gerekli kodlar yazılmıştır. Bu kısımda giriş ekranına girilen bilgilerin veritabanında

bulunup bulunmadığı sorgulaması yapılarak kullanıcının sisteme giriş yapabilmesi sağlanmıştır. Öğrenci giriş ekranı arkaplan kodları Resim 3.7’de ve öğretmen giriş ekranı arkaplan kodları Resim 3.8’de gösterilmiştir. Ayrıca kullanıcı sisteme giriş yaptığıında, “session” özelliği ile oturum bilgilerinin yönetimine yönelik kodlama yapılmıştır.

Resim 3.7 Öğrenci Giriş Ekranı Kodları

```
21 protected void giris_Click(object sender, EventArgs e)
22 {
23     string kullanıcı = kullaniciAdi.Text;
24     string sifre = kullaniciSifre.Text;
25
26     SqlCommand sorgula = new SqlCommand("SELECT * FROM student WHERE email=@email AND password=@password", baglan);
27
28     sorgula.Parameters.AddWithValue("@email", kullanıcı);
29     sorgula.Parameters.AddWithValue("@password", sifre);
30     baglan.Open();
31     SqlDataReader oku = sorgula.ExecuteReader();
32
33     if (oku.Read())
34     {
35         Session["studentId"] = oku["studentId"].ToString();
36         lblBilgi.Text = oku["studentId"].ToString();
37
38         string message = "Başarılı bir şekilde oturumunuzu açtınız!";
39         string script = "window.onload = function(){ alert('";
40         script += message;
41         script += "')});";
42         ClientScript.RegisterStartupScript(this.GetType(), "SuccessMessage", script, true);
43
44         ScriptManager.RegisterClientScriptBlock(this, typeof(Page), "redirectJS",
45         "setTimeout(function() { window.location.replace('course.aspx') }, 3000);", true);
46
47         oku.Close();
48         baglan.Close();
49         baglan.Dispose();
50     }
51     else
52     {
53         lblBilgi.Text = "Kullanıcı adınızı ya da şifrenizi hatalı girdiniz!";
54
55         oku.Close();
56         baglan.Close();
57         baglan.Dispose();
58     }
59 }
```

Resim 3.8 Eđitmen Giriř Ekranı Kodları

```
21 protected void giriř_Click(object sender, EventArgs e)
22 {
23     string email = tremail.Text;
24     string sifre = trpassword.Text;
25
26     SqlCommand sorgula2 = new SqlCommand("SELECT * FROM teacher WHERE email=@email AND password=@password", baglan);
27
28     sorgula2.Parameters.AddWithValue("@email", email);
29     sorgula2.Parameters.AddWithValue("@password", sifre);
30     baglan.Open();
31     SqlDataReader oku = sorgula2.ExecuteReader();
32     if (oku.Read())
33     {
34         Session["teacherId"] = oku["teacherId"].ToString();
35
36         string message = "Bařarılı bir řekilde oturumunuzu aėtınız!";
37         string script = "window.onload = function(){ alert('";
38         script += message;
39         script += "')});";
40         ClientScript.RegisterStartupScript(this.GetType(), "SuccessMessage", script, true);
41
42         ScriptManager.RegisterClientScriptBlock(this, typeof(Page), "redirectJS",
43         "setTimeout(function() { window.location.replace('teacherControl.aspx') }, 2000);", true);
44         oku.Close();
45         baglan.Close();
46         baglan.Dispose();
47     }
48     else
49     {
50         trlblBilgi.Text = "Kullanıcı adınızı ya da řifrenizi hatalı girdiniz!";
51         oku.Close();
52         baglan.Close();
53         baglan.Dispose();
54     }
55 }
```

Rezervasyon kayıtlarının veritabanına kayıt edilmesine ilişkin kodlama Resim 3.9’da gösterilmiştir. Bu sayfada kullanıcıların seçimleri oluşturulan değişkenler ile takip edilmiş ve sql komutları kullanılarak “reservation” veritabanı tablosuna kayıt edilmesi sağlanmıştır. Kayıt edilme sırasında seçilen tarih aralığında eđitmenin başka bir rezervasyon kaydının olup olmadığını sql komutlarıyla sorgulanarak rezervasyon kaydı oluşturan öğrenciye sadece uygun saat dilimlerinin gösterilmesi sağlanmıştır. Bu kontrol; kullanıcı tarafından tarih seçildiđi anda tarih bilgilerin gösterildi text bileřenin “event” ı kullanılarak yapılmış olup bu kodlamaya ilişkin görsel Resim 3.10’da gösterilmiştir.

Resim 3.9 Rezervasyon Kaydına İlişkin Kodlar

```
37 protected void Button1_Click(object sender, EventArgs e)
38 {
39     SqlConnection baglan = new SqlConnection(WebConfigurationManager.ConnectionStrings["Veritabani"].ConnectionString);
40
41     SqlCommand sorgula2 = new SqlCommand("INSERT INTO reservation (teacherId,studentId, date_choose, hour, lessonname, lessontype) " +
42     "VALUES ( @teacher, @student, @tarih, @saat, @ders, @derstipi)", baglan);
43
44     sorgula2.Connection = baglan;
45     string secilenTarih2 = tarih.Text;
46     sorgula2.Parameters.AddWithValue("@trh", secilenTarih2);
47     string sa = saatDrop.SelectedValue;
48     var id = GridView1.SelectedRow.FindControl("trid") as Label;
49     string id2 = id.Text;
50     int id3 = int.Parse(id2);
51     string sessionId = Session["studentId"].ToString();
52     int stdId = int.Parse(sessionId);
53     string derstip = dersTipi.SelectedValue;
54
55     sorgula2.Parameters.AddWithValue("@ders", GridView1.SelectedRow.Cells[2].Text);
56     sorgula2.Parameters.AddWithValue("@tarih", secilenTarih2);
57     sorgula2.Parameters.AddWithValue("@saat", sa);
58     sorgula2.Parameters.AddWithValue("@teacher", id3);
59     sorgula2.Parameters.AddWithValue("@student", stdId);
60     sorgula2.Parameters.AddWithValue("@derstipi", derstip);
61     baglan.Open();
62
63     sorgula2.ExecuteNonQuery();
64     baglan.Close();
65
66     string message = "Rezervasyonunuz başarılı bir şekilde oluşturuldu!";
67     string script = "window.onload = function(){ alert('";
68     script += message;
69     script += "')}";
70     ClientScript.RegisterStartupScript(this.GetType(), "SuccessMessage", script, true);
71
72     ScriptManager.RegisterClientScriptBlock(this, typeof(Page), "redirectJS",
73     "setTimeout(function() { window.location.replace('course.aspx') }, 2000);", true);
74
75     saatDrop.ClearSelection();
76     dersTipi.ClearSelection();
77 }
```

Resim 3.10 Rezervasyon Kayıt Kontrol Kodları

```
78 protected void tarih_TextChanged(object sender, EventArgs e)
79 {
80     SqlConnection baglan2 = new SqlConnection(WebConfigurationManager.ConnectionStrings["Veritabani"].ConnectionString);
81
82     SqlCommand sorgu = new SqlCommand("select saat from hour where saat not in (select hour from reservation where teacherId=@ogr and date_choose=@trh)", baglan2);
83
84     sorgu.Connection = baglan2;
85     string secilenTarih = tarih.Text;
86     sorgu.Parameters.AddWithValue("@trh", secilenTarih);
87
88     var id = GridView1.SelectedRow.FindControl("trid") as Label;
89     string id2 = id.Text;
90     int id3 = int.Parse(id2);
91     sorgu.Parameters.AddWithValue("@ogr", id3);
92
93     saatDrop.Items.Clear();
94     baglan2.Open();
95     SqlDataReader dr = sorgu.ExecuteReader();
96
97     while (dr.Read())
98     {
99         string x = dr["saat"].ToString();
100         saatDrop.Items.Add(x);
101     }
102
103     baglan2.Close();
104 }
105
106
107 }
```

Arkaplan kodlamasında son aşama olarak öğrenci ve öğretmen kullanıcı tipleri için kontrol panel sayfaları için kodlama yapılmıştır. Öğrenci kontrol panel sayfası kodlamasında, öğrencinin rezervasyon yaptığı ders bilgilerin listelendiği tablonun gösterilmesini sağlamak için yine sql kodları kullanılmıştır. Bu tablodaki bilgiler sadece sayfaya giriş yapmış kullanıcıya ait olduğundan session ile kullanıcı bilgileri temin edilmiş ve bu bilgi sql kodlarında koşul parametresi olarak kullanılmıştır. Öğrenci kontrol paneli sayfasına ilişkin kodlar Resim 3.11’de gösterilmiştir.

Resim 3.11 Öğrenci Kontrol Paneli Kodları

```
11 namespace CourseProject
12 {
13     public partial class studentControl : System.Web.UI.Page
14     {
15         SqlConnection baglan = new SqlConnection(WebConfigurationManager.ConnectionStrings["Veritabani"].ConnectionString);
16
17         protected void Page_Load(object sender, EventArgs e)
18         {
19
20             baglan.Open();
21
22             string ogrenciNo = Session["studentId"].ToString();
23             int ogrenci = int.Parse(ogrenciNo);
24
25             SqlCommand sorgula2 = new SqlCommand("SELECT * FROM student WHERE studentId=@ogrenciId", baglan);
26             sorgula2.Connection = baglan;
27
28             sorgula2.Parameters.AddWithValue("@ogrenciId", ogrenci);
29
30
31             SqlDataReader oku = sorgula2.ExecuteReader();
32
33             oku.Read();
34
35             icon.Text = oku["name"].ToString() + " " + oku["surname"].ToString();
36
37
38             oku.Close();
39             baglan.Close();
40
41             SqlDataSource22.SelectParameters["studentId"].DefaultValue = ogrenciNo;
42         }
43     }
```

Benzer şekilde eğitmen için de kendisine adına oluşturulan rezervasyon kayıtlarını bilgilerinin gösterildiği tablo için sql komutları kullanılmıştır. Eğitmen hesabına özel rezervasyon bilgilerinin görüntülenebilmesi için yine session ile kullanıcı bilgileri temin edilmiş ve bu bilgi koşul parametresi olarak kullanılmıştır. Eğitmen kontrol paneline ilişkin kodlar Resim 3.12’de gösterilmiştir.

Ayrıca kullanıcının uygulamadan çıkış yapması ve session üzerinden tutulan oturum bilgilerini sonlandırmak için kodlama yapılmış olup bu işleme ait kodlar Resim 3.13’te gösterilmiştir.

Resim 3.12 Eđitmen Kontrol Paneli Kodları

```
12 namespace CourseProject
13 {
14     public partial class TeacherControl : System.Web.UI.Page
15     {
16         SqlConnection baglan = new SqlConnection(WebConfigurationManager.ConnectionStrings["Veritabani"].ConnectionString);
17
18         protected void Page_Load(object sender, EventArgs e)
19         {
20             baglan.Open();
21
22             string ogretmenId = Session["teacherId"].ToString();
23             int ogrId = int.Parse(ogretmenId);
24
25             SqlCommand sorgula2 = new SqlCommand("SELECT * FROM teacher WHERE teacherId=@ogretmen", baglan);
26             sorgula2.Connection = baglan;
27
28             sorgula2.Parameters.AddWithValue("@ogretmen", ogrId);
29
30             SqlDataReader oku = sorgula2.ExecuteReader();
31
32             oku.Read();
33
34             icon.Text = oku["name"].ToString() + " " + oku["surname"].ToString();
35
36             oku.Close();
37             baglan.Close();
38
39             SqlDataSource1.SelectParameters["teacherId"].DefaultValue = ogretmenId;
40
41         }
42     }
43 }
```

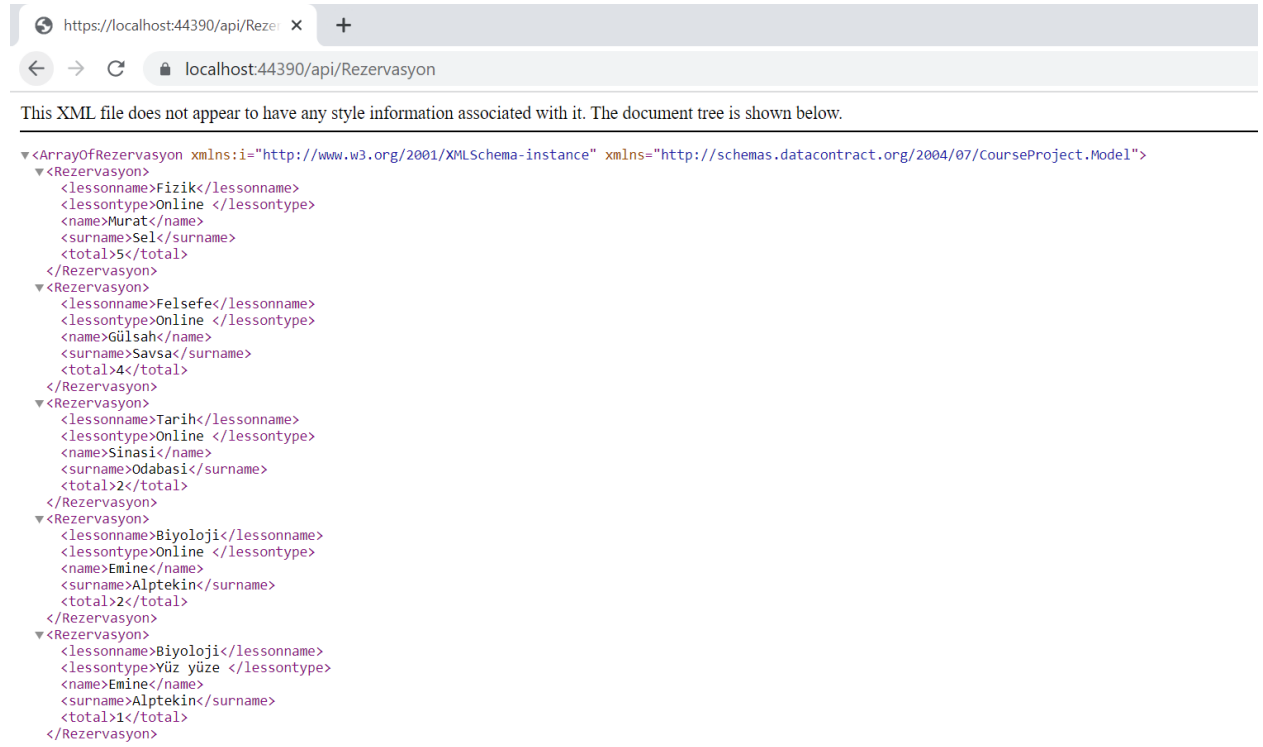
Resim 3.13 Eđitmen Kontrol Paneli Kodları

```
43
44     protected void Logout_Click(object sender, EventArgs e)
45     {
46         Session.Abandon();
47         FormsAuthentication.SignOut();
48         Response.Redirect("home3.aspx");
49     }
50 }
```

4. Api

Uygulamada vasıtasıyla oluşturulan rezervasyon bilgilerin api vasıtasıyla gönderilmesi ve gösterilmesine yönelik kodlamalar yapılmıştır. Bu doğrultuda; en çok tercih edilen eğitimcilerin ad, soyad, branş, eğitimciden kaç kere rezervasyon yapıldığı gibi bilgilerini içeren bir api oluşturmak için “controller” kullanılarak “RezervasyonController” oluşturulmuştur. Api vasıtasıyla oluşturulan verilerin ekran görüntüsü Resim 4.1’de gösterilmiştir.

Resim 4.1 Api



Api vasıtasıyla uygulamadan alınan veriler yine aynı uygulamanın ana ekranında tablo içerisinde gösterilmesi için kodlama yapılmıştır. Api ile verileri çekmek için “rezervasyon” isminde bir class oluşturulmuş ve api aracılığıyla görüntülenmesi istenen bilgiler, bu class içerisinde property olarak belirtilmiştir. Daha sonra Get methodunun yapısı “List<Rezervasyon>” tipinde oluşturularak çekilen api verilerin oluşturulacak nesnenin ilgili özelliklerine eklenmesi sağlanmıştır.

Resim 4.2 Rezervasyon Class Kodları

```
7
8 public class Rezervasyon
9 {
10
11     public string lessonname
12     { get; set; }
13     public string name
14     { get; set; }
15     public string surname
16     { get; set; }
17     public string lessonstype
18     { get; set; }
19     public int total
20     { get; set; }
21 }
```

Resim 4.3 Api ile Verilerin Alınmasının İlişkin Kodlar

```
14 [RoutePrefix("api/Rezervasyon")]
15 public class RezervasyonController : ApiController
16 {
17     SqlConnection baglan = new SqlConnection(WebConfigurationManager.ConnectionStrings["Veritabani"].ConnectionString);
18     Rezervasyon t1 = new Rezervasyon();
19
20     // GET: api/Teacher
21     [Route("")]
22     public List<Rezervasyon> Get()
23     {
24         SqlCommand cmd = new SqlCommand("SELECT reservation.[lessonname] as DersAdi, count(reservationId) as Toplam, " +
25             "teacher.name as Ad, teacher.surname as Soyad, teacher.[teacherId], reservation.[lesstontype] as DersTipi " +
26             "FROM[reservation] JOIN[teacher] ON reservation.teacherId = teacher.teacherId group by reservation.[lessonname], " +
27             "teacher.name, teacher.surname, teacher.[teacherId], reservation.[lessonname], reservation.[lesstontype] ORDER BY Toplam desc", baglan);
28
29         SqlDataAdapter da = new SqlDataAdapter();
30         da.SelectCommand = cmd;
31         DataTable dt = new DataTable();
32         da.Fill(dt);
33         List<Rezervasyon> lst = new List<Rezervasyon>();
34         if (dt.Rows.Count > 0)
35         {
36             for (int i = 0; i < dt.Rows.Count; i++)
37             {
38                 Rezervasyon r1 = new Rezervasyon();
39                 r1.name = dt.Rows[i]["Ad"].ToString();
40                 r1.surname = dt.Rows[i]["Soyad"].ToString();
41                 r1.lessonname = dt.Rows[i]["DersAdi"].ToString();
42                 r1.lesstontype = dt.Rows[i]["DersTipi"].ToString();
43                 //r1.reservationId = Convert.ToInt32(dt.Rows[i]["reservationId"]);
44                 //r1.teacherId = Convert.ToInt32(dt.Rows[i]["teacherId"]);
45                 r1.total = Convert.ToInt32(dt.Rows[i]["Toplam"]);
46                 lst.Add(r1);
47             }
48         }
49         if (lst.Count > 0)
50         {
51             return lst;
52         }
53     }
54 }
```