06 Ocak 2023

Arapgir MYO

BİLGİSAYAR PROGRAMCILIĞI

16211625011 Ümit KOÇ

BPR207 Proje I Dersi

Restoran Rezervasyon Projesi

**Geliştirilen proje hangi amaçları gerçekleştirmek ve hangi problemin çözümü için tasarlanmıştır?**

Projeyi geliştirmedeki amacımız restoran sahiplerini farklı mecralardan da müşteri bulmalarını sağlamaktır.

Restoran sahipleri için kolay reklam ve kampanya düzenlerken birçok kişiye ulaşmaları bu sayede kolaylaşacaktır.

İnsanların buradan restoran bilgilerini görüp rahatlıkla rezervasyon oluşturabilmeleri sağlanacaktır.

**Projemde, amaçlarını gerçekleştirmek için aşağıdaki teknolojiler (veri tabanı, ara yüz) kullanılmıştır.**

(SQL server, HTML, CSS, Java Script, Visual Studio, Asp.Net vs.)

**Projemin gerçekleştirdiği temel kodlanabilir özellikler**

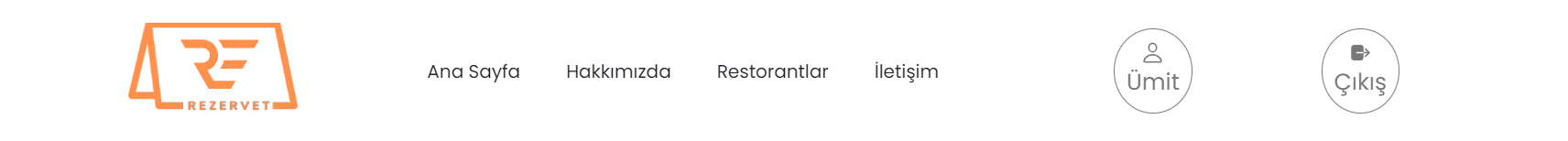
**Yönetici:**

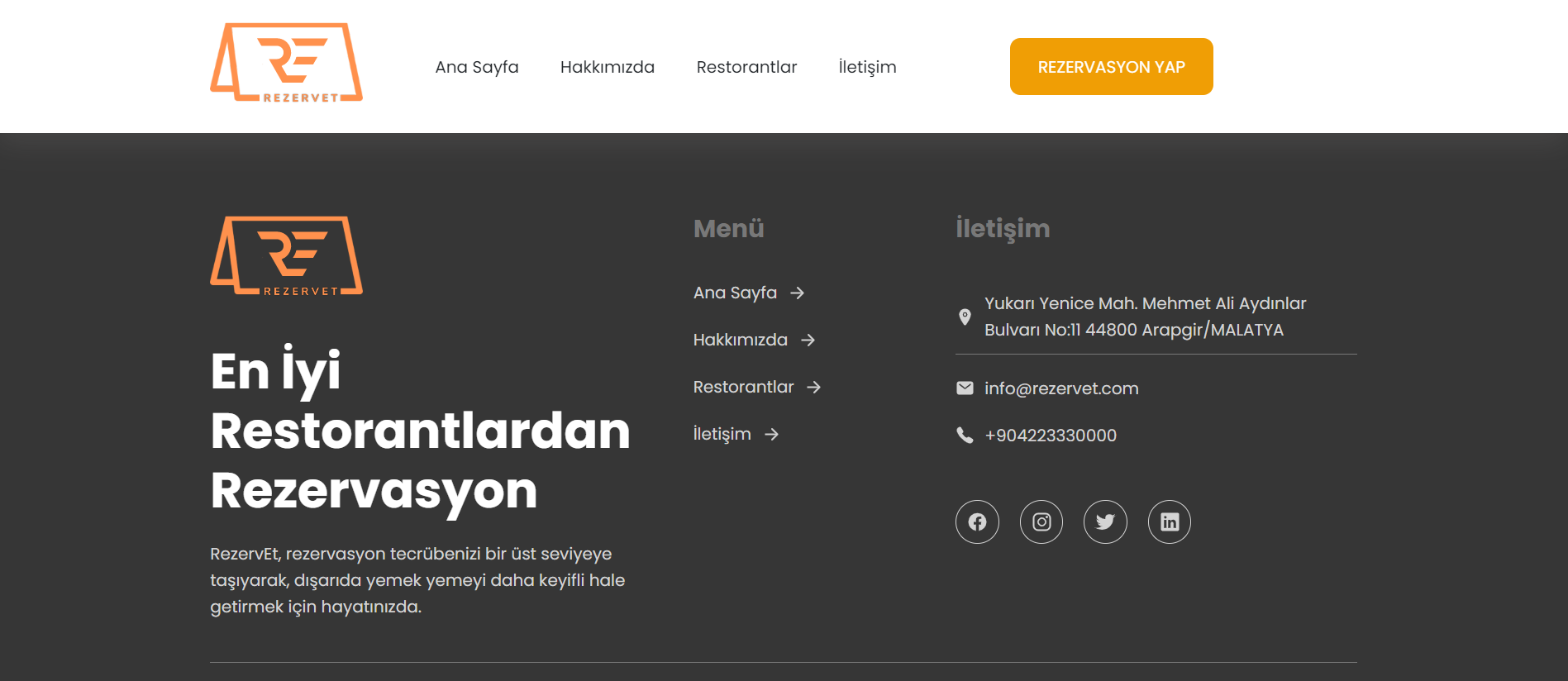
* Oluşturulan admin panelinden restoranları ekleyip düzenleme yapılabilmektedir.
* Admin kayıt yapan kullanıcı bilgilerini düzenleyebilmektedir ve kayıt olan kullanıcıyı admin olarak ekleyebilmektedir.
* Site genelinde sayfa bilgilerini düzenleyebilmektedir.
* Kullanıcıların iletişim sayfasından gönderdiği form mesajları görüntülenmektedir.

**Müşteri:**

* Rezervasyon yapabilmek için üyelik oluşturup giriş yapmalıdır.
* Yapılan tüm rezervasyonların görüntülendiği sayfa oluşturulmuştur.
* Kullanıcı bilgilerini düzenleyebileceği sayfa oluşturulmuştur.
* İletişim için form sayfası eklenmiştir.

**REZERVET- Master**

****



Bu sayfada web sitesinin master sayfası oluşturulmuştur. Master sayfasında menü ve footer bölümü bulunmaktadır.

<html>

<head runat="server">

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<link rel="shortcut icon" href="images/favicon.svg" type="image/svg+xml">

<asp:ContentPlaceHolder ID="head" runat="server">

</asp:ContentPlaceHolder>

<link rel="shortcut icon" href="image/favicon.svg" type="image/svg+xml">

<link href="css/style.css" rel="stylesheet" />

<link href="css/search.css" rel="stylesheet" />

<link rel="preconnect" href="https://fonts.googleapis.com">

<link rel="preconnect" href="https://fonts.gstatic.com">

<link href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;500;600;700&display=swap" rel="stylesheet">

</head>

<body>

Aşağıdaki kodlar ile web sitesinin menü bölümü oluşturulmuştur. Bu menü isimleri ve menü linkleri “ VeriGetir() ” metodu ile veri tabanından çekilmektedir.

<form id="form1" runat="server">

<%

Response.Write("<header class=\"header\" data-header>\r\n <div class=\"container\">\r\n\r\n <a href=" + VeriGetir(16) + " class=\"logo\">\r\n\r\n <img src=" + VeriGetir(1) + " width=\"148\" height=\"38\" alt=\"RezervEt\">\r\n </a>\r\n\r\n <nav class=\"navbar\" data-navbar>\r\n <button class=\"nav-close-btn\" aria-label=\"close menu\" data-nav-toggler>\r\n <ion-icon name=\"close-outline\" aria-hidden=\"true\"></ion-icon>\r\n </button>\r\n\r\n <a href=" + VeriGetir(16) + " class=\"logo\">\r\n <img src=" + VeriGetir(1) + " width=\"148\" height=\"38\" alt=\"RezervET\">\r\n </a>\r\n\r\n <ul class=\"navbar-list\">\r\n\r\n <li class=\"navbar-item\">\r\n <a href=" + VeriGetir(16) + " class=\"navbar-link\" data-nav-link>" + VeriGetir(12) + "</a>\r\n </li>\r\n\r\n <li class=\"navbar-item\">\r\n <a href=" + VeriGetir(17) + " class=\"navbar-link\" data-nav-link>" + VeriGetir(13) + "</a>\r\n </li>\r\n\r\n <li class=\"navbar-item\">\r\n <a href=" + VeriGetir(18) + " class=\"navbar-link\" data-nav-link>" + VeriGetir(14) + "</a>\r\n </li>\r\n\r\n <li class=\"navbar-item\">\r\n <a href=" + VeriGetir(19) + " class=\"navbar-link\" data-nav-link>" + VeriGetir(15) + "</a>\r\n </li>\r\n <li>\r\n <div class=\"search-icon\">\r\n <i class=\"fas fa-search\"></i>\r\n </div>\r\n </li>\r\n\r\n\r\n </ul>\r\n </nav>");

%>

<a runat="server" id="rezerv" href="Rezervasyon.aspx" class="btn btn-primary has-after">Rezervasyon Yap</a>

<center>

<asp:LinkButton ID="hesap" OnClick="hesap\_Click" CssClass="social-link" runat="server">LinkButton</asp:LinkButton>

</center>

<center>

<asp:LinkButton ID="cks" OnClick="cks\_Click" CssClass="social-link" runat="server"><ion-icon name="log-out"></ion-icon>Çıkış</asp:LinkButton>

</center>

<%

Response.Write("</div>\r\n\r\n <button class=\"nav-open-btn\" aria-label=\"open menu\" data-nav-toggler>\r\n <ion-icon name=\"menu-outline\" aria-hidden=\"true\"></ion-icon>\r\n </button>\r\n\r\n\r\n\r\n <div class=\"overlay\" data-overlay data-nav-toggler></div>\r\n\r\n </header>");

%>

<div>

<asp:ContentPlaceHolder ID="ContentPlaceHolder1" runat="server"></asp:ContentPlaceHolder>

</div>

Aşağıdaki kodlar ile web sitesinin Footer bölümü oluşturulmuştur. Footer bölümünde bulunan logo, kısa başlık, açıklama, menü isimleri, menü linkleri ve iletişim bölümü “ VeriGetir() ” metodu ile veri tabanından çekilmektedir.

<!-- Footer -->

<%

Response.Write("<footer class=\"footer\">\r\n <div class=\"container\">\r\n\r\n <div class=\"section footer-top grid-list\">\r\n\r\n <div class=\"footer-brand\">\r\n\r\n <a href=\"#\" class=\"logo\">\r\n <img src=" + VeriGetir(1) + " width=\"148\" height=\"38\" alt=\"fasteat home\">\r\n </a>\r\n\r\n <h2 class=\"h2 section-title\">" + VeriGetir(2) + "</h2>\r\n\r\n <p class=\"section-text\">" + VeriGetir(3) + "</p>\r\n\r\n </div>\r\n\r");

//Menu

Response.Write("<ul class=\"footer-list\">\r\n\r\n <li>\r\n <p class=\"footer-list-title h5\">Menü</p>\r\n </li>\r\n\r\n <li>\r\n <a href=" + VeriGetir(16) + " class=\"footer-link\">\r\n <span class=\"span\">" + VeriGetir(12) + "</span>\r\n\r\n <ion-icon name=\"arrow-forward\" aria-hidden=\"true\"></ion-icon>\r\n </a>\r\n </li>\r\n\r\n <li>\r\n <a href=" + VeriGetir(17) + " class=\"footer-link\">\r\n <span class=\"span\">" + VeriGetir(13) + "</span>\r\n\r\n <ion-icon name=\"arrow-forward\" aria-hidden=\"true\"></ion-icon>\r\n </a>\r\n </li>\r\n\r\n <li>\r\n <a href=" + VeriGetir(18) + " class=\"footer-link\">\r\n <span class=\"span\">" + VeriGetir(14) + "</span>\r\n\r\n <ion-icon name=\"arrow-forward\" aria-hidden=\"true\"></ion-icon>\r\n </a>\r\n </li>\r\n\r\n <li>\r\n <a href=" + VeriGetir(19) + " class=\"footer-link\">\r\n <span class=\"span\">" + VeriGetir(15) + "</span>\r\n\r\n <ion-icon name=\"arrow-forward\" aria-hidden=\"true\"></ion-icon>\r\n </a>\r\n </li>\r\n\r\n </ul>\r\n\r\n <ul class=\"footer-list\">\r\n\r\n <li>\r\n <p class=\"footer-list-title h5\">İletişim</p>\r\n </li>\r\n\r");

//Adres Satırı

Response.Write("<li><a href=" + VeriGetir(20) + " class=\"footer-link\"><address class=\"address\"><ion-icon name=\"location\" aria-hidden=\"true\"></ion-icon><span class=\"span\">" + VeriGetir(4) + "</span></address></a></li>");

//E Posta Satırı

Response.Write("<li><a href=mailto:" + VeriGetir(6) + " class=\"footer-link\"><ion-icon name=\"mail\" aria-hidden=\"true\"></ion-icon><span class=\"span\">" + VeriGetir(5) + "</span></a></li>");

//Telefon Satırı

Response.Write("<li><a href=tel:" + VeriGetir(6) + " class=\"footer-link\"><ion-icon name=\"call\" aria-hidden=\"true\"></ion-icon><span class=\"span\">" + VeriGetir(6) + "</span></a></li>");

//Sosyal Medya Satırı

Response.Write("<li><ul class=\"social-list\"><li><a href=" + VeriGetir(7) + " class=\"social-link\"><ion-icon name=\"logo-facebook\"></ion-icon></a></li><li><a href=" + VeriGetir(8) + " class=\"social-link\"><ion-icon name=\"logo-instagram\"></ion-icon></a></li><li><a href=" + VeriGetir(9) + " class=\"social-link\"><ion-icon name=\"logo-twitter\"></ion-icon></a></li><li><a href=" + VeriGetir(10) + " class=\"social-link\"><ion-icon name=\"logo-linkedin\"></ion-icon></a></li></ul></li></ul></div><div class=\"footer-bottom\">\r\n\r\n <p class=\"copyright\">\r\n Copyright &copy; 2022 umitkoc. Tüm Hakları Saklıdır.\r\n </p>\r\n\r\n </div></div></div>\r\n </footer>");

%>

<!--

- custom js link

-->

<script src="java/script.js"></script>

<!--

- ionicon link

-->

<script type="module" src="https://unpkg.com/ionicons@5.5.2/dist/ionicons/ionicons.esm.js"></script>

<script nomodule src="https://unpkg.com/ionicons@5.5.2/dist/ionicons/ionicons.js"></script>

</form>

</body>

</html>

**Master (cs)**

public partial class AnaSayfa : System.Web.UI.MasterPage

{

Bilgi bilgi = new Bilgi();

Sayfa yüklendiğine kullanıcı giriş yapmamışsa “Rezervasyon Yap” butonu görünmektedir. Giriş yapılmışsa kullanıcı bilgilerinin düzenlenebildiği ve üzerinde kullanıcının ismi yazan, buton aktif olmaktadır. Bununla birlikte kullanıcının çıkış yapabildiği buton aktif olmaktadır.

protected void Page\_Load(object sender, EventArgs e)

{

if (Session["Kullanici"] == null)

{

rezerv.Visible = true;

hesap.Visible = false;

cks.Visible = false;

}

else

{

hesap.Visible = true;

hesap.Text = "<ion-icon name=\"person-outline\"></ion-icon><span>" + Session["Kullanici"].ToString() + "</span>";

cks.Visible = true;

rezerv.Visible = false;

}

}

Çıkış Butonuna tıklandığında sessionları temizleyip ana sayfaya yönlendirmektedir.

protected void cks\_Click(object sender, EventArgs e)

{

Session.Abandon();

Response.Redirect("default.aspx");

}

Hesap butonuna tıklandığında kullanıcıyı, yaptığı rezervasyonları ve bilgilerini görüntüleyebildiği sayfaya yönlendirmektedir.

protected void hesap\_Click(object sender, EventArgs e)

{

Response.Redirect("Hesap.aspx");

}

Bu metot sayesinde gelen parametre ile sütun bilgisi alınır. Bilgi sınıfındaki “SiteGenel()” metoduyla sayfa bilgileri getirilir.

public string VeriGetir(int a)

{

DataTable dt = new DataTable();

dt = bilgi.SiteGenel();

return dt.Rows[0][a].ToString();

}

}

|  |  |
| --- | --- |
| public DataTable SiteGenel()  {  DataTable dt = new DataTable();  try  {  using (SqlConnection baglanti = b.BaglantiOlustur())  {  using (SqlCommand komut = new SqlCommand("site\_genel\_usp", baglanti))  {  komut.CommandType = CommandType.StoredProcedure;  baglanti.Open();  SqlDataAdapter da;  DataSet ds = new DataSet();  da = new SqlDataAdapter(komut);  da.Fill(ds, "Site");  dt = ds.Tables["Site"];  baglanti.Close();  }  }  }  catch (Exception hata)  {  }  return dt;  } | Bu “SiteGenel()” metodu sayesinde veri tabanından “site\_genel\_usp” prosedürü çalıştırılıp bilgiler DataTable ile geri gönderilir. |
| CREATE proc [dbo].[site\_genel\_usp]  as  begin  select \* from SiteGenel  end | “site\_genel\_usp” prosedürü |

**REZERVET- Ana Sayfa**

|  |
| --- |
|  |
|  |
|  |

Bu bölümde web sitenin Ana Sayfası oluşturulmuştur. Ana sayfada Rezervasyon yapabilmek için restoranlar sayfasına yönlendiren bir adet buton web sitesi sloganı bulunmaktadır. 3 adet restoran ve tümünün bulunduğu sayfaya yönlendiren buton bulunmaktadır. Kampanyaların eposta adresine gönderildiği bülten bölümü bulunmaktadır.

<link href="css/style.css" rel="stylesheet" />

<link rel="preconnect" href="https://fonts.googleapis.com">

<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

<link href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;500;600;700&display=swap" rel="stylesheet">

<!-- Ön Yükleyici -->

<div class="loading-container" data-loading-container>

<div class="loading-circle"></div>

</div>

Aşağıdaki kodlar ile web sitesinin Ana Sayfa bölümü oluşturulmuştur. Ana Sayfa bölümünde bulunan başlıklar, açıklamalar, resimler “VeriGetirGenel()” metodu ile veri tabanından çekilmektedir.

<%

//Başlangıç

Response.Write("<main>\r\n <article><section class=\"section hero has-bg-image\" aria-label=\"home\"\r\n style=\"background-image: url('"+VeriGetirGenel(10)+"')\">\r\n <div class=\"container\">\r\n\r\n <div class=\"hero-content\" data-reveal=\"left\">\r\n <h1 class=\"h1 hero-title\">"+VeriGetirGenel(1)+"</h1>\r\n\r\n <p class=\"hero-text\">\r\n "+VeriGetirGenel(4)+"\r\n </p>\r\n\r\n <a href=\"Rezervasyon.aspx\" class=\"btn btn-secondary has-after\">Rezervasyon Yap</a>\r\n </div>\r\n\r\n <figure class=\"hero-banner\" data-reveal>\r\n\r\n <img src="+VeriGetirGenel(7)+" width=\"680\" height=\"720\" alt=\"hero banner\" class=\"w-100\">\r\n </figure>\r\n\r\n </div>\r\n </section>");

//Restaurantlar

Response.Write("<section class=\"section top-restaurant\" aria-labelledby=\"top-restaurent-label\">\r\n <div class=\"container\">\r\n\r\n <ul class=\"grid-list\">\r\n\r\n <li data-reveal=\"left\">\r\n <h2 class=\"h2 section-title\" id=\"top-restaurent-label\">"+VeriGetirGenel(2)+"\r\n </h2>\r\n\r\n <p class=\"section-text\">\r\n "+VeriGetirGenel(5)+"\r\n </p>\r\n </li>");

Aşağıdaki kodlar ile Ana Sayfadaki restoranlar bölümü oluşturulmuştur. Restoranlar bölümünde bulunan restoran bilgileri “VeriGetirRestoran()” metodu ile veri tabanından çekilmektedir. Üç adet restoran göstermek için for döngüsü kullanılmıştır.

for (int i = 0; i < 3; i++)

{

//Restoran Logosu

Response.Write("<li data-reveal=\"left\"><div class=\"restaurant-card\"><div class=\"card-icon\"><img src=" + VeriGetirRestoran(i,5) + " width=\"100\" height=\"100\" loading=\"lazy\"alt=" + VeriGetirRestoran(i, 1) + " class=\"w-100\"></div>");

//Restoran İsmi

Response.Write("<h3 class=\"h5 card-title\">" + VeriGetirRestoran(i, 1) + "</h3><div class=\"card-meta-wrapper\"><a href=\"#\" class=\"card-meta\">Rezervasyon</a></div>");

//Restoran Bilgileri

Response.Write("<p class=\"card-text\">" + VeriGetirRestoran(i, 4) + "</p></div></li>");

}

Response.Write("</ul>\r\n\r\n <a href=\"Restaurantlar.aspx\" class=\"btn btn-secondary has-after\">\r\n <span class=\"span\">Hepsini Gör</span>\r\n\r\n <ion-icon name=\"arrow-forward\" aria-hidden=\"true\"></ion-icon>\r\n </a>\r\n\r\n </div>\r\n </section>");

//Reklam

Response.Write("<section class=\"section cta has-bg-image\" aria-labelledby=\"cta-label\"\r\n style=\"background-image: url('"+VeriGetirGenel(10)+"')\">\r\n <div class=\"container\">\r\n\r\n <figure class=\"cta-banner\" data-reveal=\"left\">\r\n <img src="+VeriGetirGenel(8)+" width=\"680\" height=\"704\" loading=\"lazy\" alt=\"cta banner\"\r\n class=\"w-100\">\r\n </figure>\r\n\r\n <div class=\"cta-content\" data-reveal=\"right\">\r\n\r\n <h2 class=\"h3 section-title\" id=\"cta-label\">"+VeriGetirGenel(1)+"\r\n </h2>\r\n\r\n <p class=\"section-text\">\r\n "+VeriGetirGenel(6)+"\r\n </p>\r\n\r\n <a href=\"giris.aspx\" class=\"btn btn-primary has-after\">RezervET</a>\r\n\r\n </div>\r\n\r\n </div>\r\n </section>");

//Bülten

Response.Write("<section class=\"section newsletter\" aria-label=\"newsletter\">\r\n <div class=\"container\">\r\n\r\n <figure class=\"newsletter-banner\" data-reveal=\"left\">\r\n <img src="+VeriGetirGenel(9)+" width=\"680\" height=\"405\" loading=\"lazy\" alt=\"Illustration\"\r\n class=\"w-100\">\r\n </figure>\r\n\r\n <div class=\"newsletter-content\" data-reveal=\"right\">\r\n <h2 class=\"h4 section-title\">"+VeriGetirGenel(3)+"\r\n </h2>\r\n\r\n <div class=\"input-wrapper\">");

%>

<asp:TextBox ID="TextBox1" runat="server" placeholder="E-mail adresiniz" CssClass="input-field"></asp:TextBox>

<asp:Button ID="Button1" runat="server" Text="Abone Ol" CssClass="btn " />

<%

Response.Write("</div>\r\n\r\n </div>\r\n\r\n </div>\r\n </section>\r\n\r\n </article>\r\n </main>");

%>

<script src="java/script.js"></script>

**Ana Sayfa (cs)**

public partial class default : System.Web.UI.Page

{

Bilgi bilgi = new Bilgi();

protected void Page\_Load(object sender, EventArgs e)

{

}

Bu metot sayesinde gelen parametre ile sütun bilgisi alınır. Bilgi sınıfındaki “AnaSayfa()” metoduyla sayfa bilgileri getirilir.

public string VeriGetirGenel(int a)

{

DataTable dt = new DataTable();

dt = bilgi.AnaSayfa();

return dt.Rows[0][a].ToString();

}

Bu metot sayesinde gelen parametre ile satır ve sütun bilgisi alınır. Bilgi sınıfındaki “RestoranGetir()” metoduyla sayfa bilgileri getirilir.

public string VeriGetirRestoran(int a, int b)

{

DataTable dt = new DataTable();

dt = bilgi.RestoranGetir();

return dt.Rows[a][b].ToString();

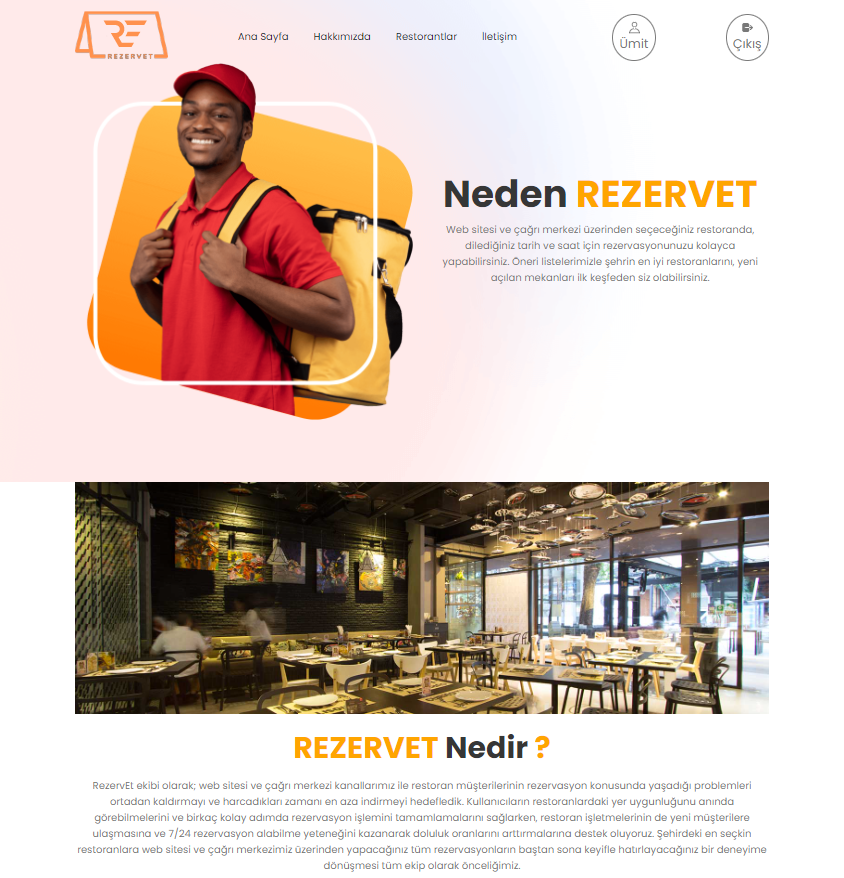
}

}

|  |  |
| --- | --- |
| public DataTable AnaSayfa()  {  DataTable dt = new DataTable();  try  {  using (SqlConnection baglanti = b.BaglantiOlustur())  {  using (SqlCommand komut = new SqlCommand("anasayfa\_usp", baglanti))  {  komut.CommandType = CommandType.StoredProcedure;  baglanti.Open();  SqlDataAdapter da;  DataSet ds = new DataSet();  da = new SqlDataAdapter(komut);  da.Fill(ds, "AnaSayfa");  dt = ds.Tables["AnaSayfa"];  baglanti.Close();  }  }  }  catch (Exception hata)  {  }  return dt;  } | Bu “AnaSayfa()” metodu sayesinde veri tabanından “anasayfa\_usp” prosedürü çalıştırılıp bilgiler DataTable ile geri gönderilir. |
| CREATE proc [dbo].[anasayfa\_usp]  as  begin  select \* from AnaSayfa  end | “anasayfa\_usp” prosedürü |

|  |  |
| --- | --- |
| public DataTable RestoranGetir()  {  DataTable dt = new DataTable();  try  {  using (SqlConnection baglanti = b.BaglantiOlustur())  {  using (SqlCommand komut = new SqlCommand("restoran\_getir\_usp", baglanti))  {  komut.CommandType = CommandType.StoredProcedure;  baglanti.Open();  SqlDataAdapter da;  DataSet ds = new DataSet();  da = new SqlDataAdapter(komut);  da.Fill(ds, "Restoranlar");    dt = ds.Tables["Restoranlar"];  baglanti.Close();  }  }  }  catch (Exception hata)  {  }  return dt;  } | Bu “RestoranGetir()” metodu sayesinde veri tabanından “restoran\_getir\_usp” prosedürü çalıştırılıp bilgiler DataTable ile geri gönderilir. |
| CREATE proc [dbo].[restoran\_getir\_usp]  as  begin  select \* from Restoranlar  end | “restoran\_getir\_usp” prosedürü |

**REZERVET- Hakkımızda**

****

Bu bölümde web sitesinin Hakkımızda Sayfası oluşturulmuştur. Hakkımızda sayfasında REZERVET ile ilgili bilgiler bulunmaktadır.

<div class="loading-container" data-loading-container>

<div class="loading-circle"></div>

</div>

Hakkımızda bölümünde bulunan başlıklar, açıklamalar, resimler “VeriGetir ()” metodu ile veri tabanından çekilmektedir.

<%

Response.Write("<section class=\"section hero has-bg-image\" aria-label=\"home\"\r\n style=\"background-image: url('" + VeriGetir(9) + "')\">\r\n <div class=\"container\">\r\n <figure class=\"hero-banner\" data-reveal>\r\n\r\n <img src=" + VeriGetir(7) + " width=\"680\" height=\"720\" alt=\"hero banner\" class=\"w-100\">\r\n </figure>\r\n <div class=\"hero-content\" data-reveal=\"right\" style=\"text-align: center\">\r\n <h1 class=\"h1 hero-title\">" + VeriGetir(1) + " <font color=\"orange\"> " + VeriGetir(2) + "</font> </h1>\r\n\r\n <p class=\"hero-text\" style=\"text-align: center\">\r\n " + VeriGetir(5) + "\r\n </p>\r\n </div>\r\n\r\n \r\n\r\n </div>\r\n </section>");

Response.Write("<section class=\"section instruction\" aria-labelledby=\"\" style=\"text-align: center\">\r\n <div class=\"container\">\r\n <figure class=\"card-banner\">\r\n <img src=" + VeriGetir(8) + " width=\"300\" height=\"154\" loading=\"lazy\"\r\n alt=\"Select Restaurant\" class=\"w-100\">\r\n </figure><br />\r\n <h2 class=\"h2 section-title\" id=\"instruction-label\" data-reveal><font color=\"orange\"> " + VeriGetir(2) + "</font> " + VeriGetir(3) + "<font color=\"orange\"> " + VeriGetir(4) + "</font></h2>\r\n\r\n <p class=\"section-text\" data-reveal>\r\n " + VeriGetir(6) + "\r\n </p>\r\n\r\n \r\n\r\n </div>\r\n </section>");

%>

<script src="java/script.js"></script>

**Hakkımızda (cs)**

public partial class Hakkimizda : System.Web.UI.Page

{

Bilgi bilgi = new Bilgi();

protected void Page\_Load(object sender, EventArgs e)

{

}

Bu metot sayesinde gelen parametre ile sütun bilgisi alınır. Bilgi sınıfındaki “Hakkimda()” metoduyla sayfa bilgileri getirilir.

public string VeriGetir(int a)

{

DataTable dt = new DataTable();

dt = bilgi.Hakkimda();

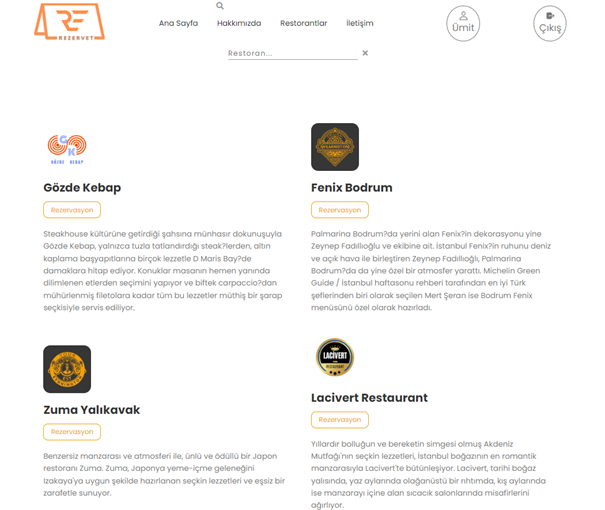
return dt.Rows[0][a].ToString();

}

}

|  |  |
| --- | --- |
| public DataTable Hakkimda()  {  DataTable dt = new DataTable();  try  {  using (SqlConnection baglanti = b.BaglantiOlustur())  {  using (SqlCommand komut = new SqlCommand("hakkimda\_usp", baglanti))  {  komut.CommandType = CommandType.StoredProcedure;  baglanti.Open();  SqlDataAdapter da;  DataSet ds = new DataSet();  da = new SqlDataAdapter(komut);  da.Fill(ds, "AnaSayfa");  dt = ds.Tables["AnaSayfa"];  baglanti.Close();  }  }  }  catch (Exception hata)  {  }  return dt;  } | Bu “AnaSayfa()” metodu sayesinde veri tabanından “hakkimda\_usp” prosedürü çalıştırılıp bilgiler DataTable ile geri gönderilir. |
| CREATE proc [dbo].[hakkimda\_usp]  as  begin  select \* from Hakkimda  end | “hakkimda\_usp” prosedürü |

**REZERVET- Restoranlar**



Bu bölümde web sitesinin Restoranlar Sayfası oluşturulmuştur. Restoranlar sayfasında tüm restoranlar bulunmaktadır.

Aşağıdaki kodlar ile restoranlar arasında arama yapılabilmektedir.

<div class="search-input-wrapper">

<div class="search-input">

<asp:TextBox OnTextChanged="text\_TextChanged" ID="ara" runat="server" placeholder="Restoran..."></asp:TextBox>

<i class="fas fa-times"></i>

</div>

</div>

<div class="loading-container" data-loading-container>

<div class="loading-circle">

</div>

</div>

Restoranların bilgileri “DataList” kullanılarak veri tabanından getirilmiştir.

<section class="section top-restaurant" aria-labelledby="top-restaurent-label">

<div class="container">

<ul class="grid-list">

<asp:DataList ID="DataList1" runat="server" OnItemCommand="DataList1Command" HorizontalAlign="Justify" Font-Bold="False" Font-Italic="False" Font-Overline="False" Font-Strikeout="False" Font-Underline="False" RepeatColumns="2" RepeatDirection="Horizontal" Width="1100px">

<ItemTemplate>

<li data-reveal="left">

<div class="restaurant-card">

<div class="card-icon">

<img src=" <%# Eval("resim") %> " width="100" height="100" loading="lazy" alt=" <%# Eval("isim") %> " class="w-100">

</div>

<h3 class="h5 card-title"><%# Eval("isim") %> </h3>

<div class="card-meta-wrapper">

Restoranların üzerindeki rezervasyon butonuna tıklandığında restoran bilgilerini alıp rezervasyon sayfasına yölendirir.

<asp:LinkButton ID="LinkButton1" CssClass="card-meta" runat="server" CommandArgument='<%# Eval("ID") %>' CommandName="git">Rezervasyon</asp:LinkButton>

</div>

<p class="card-text"><%# Eval("bilgi") %> </p>

</div>

</li>

</ItemTemplate>

</asp:DataList>

</ul>

</div>

</section>

<script src="java/script.js"></script>

<script src="java/search.js"></script>

**Restoranlar (cs)**

public partial class Restoranlar : System.Web.UI.Page

{

Bilgi bilgi = new Bilgi();

Sayfa yüklendiğinde bilgi sınıfındaki “RestoranGetir()” metoduyla restoran bilgileri getirilir. Bu bilgiler “DataList” ile sayfada gösterilir.

protected void Page\_Load(object sender, EventArgs e)

{

DataTable dt = new DataTable();

dt = bilgi.RestoranGetir();

DataList1.DataSource = dt;

DataList1.DataBind();

}

Bu metot sayesinde ara TextBox’ındaki veriyi alır. Bilgi sınıfındaki “Ara()” metoduyla aranan restoran bilgileri getirilir.

protected void text\_TextChanged(object sender, EventArgs e)

{

DataTable dt = new DataTable();

dt = bilgi.Ara(ara.Text);

DataList1.DataSource = dt;

DataList1.DataBind();

}

Bu metot sayesinde CommanArgument ile restoran ID bilgisi alınır. Restoran Session’una atılır ve rezervasyon sayfasına yölendirilir.

protected void DataList1Command(object sender, DataListCommandEventArgs e)

{

if (e.CommandName.ToString() == "git")

{

Session["Restorant"]=(e.CommandArgument.ToString());

Response.Redirect("Rezervasyon.aspx");

}

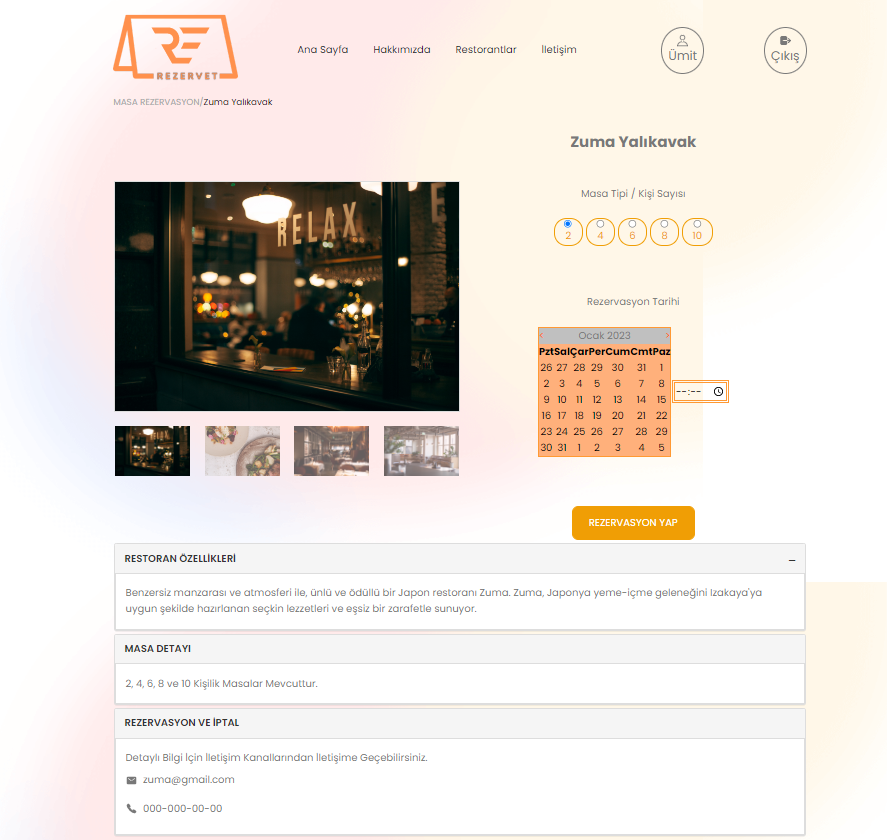
}

}

|  |  |
| --- | --- |
| public DataTable RestoranGetir()  {  DataTable dt = new DataTable();  try  {  using (SqlConnection baglanti = b.BaglantiOlustur())  {  using (SqlCommand komut = new SqlCommand("restoran\_getir\_usp", baglanti))  {  komut.CommandType = CommandType.StoredProcedure;  baglanti.Open();  SqlDataAdapter da;  DataSet ds = new DataSet();  da = new SqlDataAdapter(komut);  da.Fill(ds, "AnaSayfa");  dt = ds.Tables["AnaSayfa"];  baglanti.Close();  }  }  }  catch (Exception hata)  {  }  return dt;  } | Bu “RestoranGetir()” metodu sayesinde veri tabanından “restoran\_getir\_usp” prosedürü çalıştırılıp bilgiler DataTable ile geri gönderilir. |
| CREATE proc [dbo].[restoran\_getir\_usp]  as  begin  select \* from Restoran  end | “restoran\_getir\_usp” prosedürü |

|  |  |
| --- | --- |
| public DataTable Ara(string parametre)  {  DataTable dt = new DataTable();  try  {  using (SqlConnection baglanti = b.BaglantiOlustur())  {  using (SqlCommand komut = new SqlCommand("restoran\_ara\_usp", baglanti))  {    komut.CommandType = CommandType.StoredProcedure;  komut.Parameters.AddWithValue("@ad", parametre);  baglanti.Open();  SqlDataAdapter da;  DataSet ds = new DataSet();  da = new SqlDataAdapter(komut);  da.Fill(ds, "Restoranlar");  dt = ds.Tables["Restoranlar"];  baglanti.Close();  }  }  }  catch (Exception hata)  {    }  return dt;  } | Bu “Ara()” metodu sayesinde veri tabanından “restoran\_ara\_usp” prosedürü çalıştırılıp içerisine gönderilen parametreler ile gelen bilgiler DataTable ile geri gönderilir. |
| CREATE proc [dbo].[restoran\_ara\_usp]  (  @ad varchar(70)  )  as  begin  select \* from Restoranlar where isim like '%'+@ad+'%'  end | “restoran\_ara\_usp” prosedürü |

**REZERVET- Rezervasyon**



Bu bölümde web sitesinin Rezervasyon Sayfası oluşturulmuştur. Rezervasyon sayfasında seçilen restoran ile ilgili bilgiler bulunmaktadır. Bu sayfada kullanıcı rezervasyon yapabilmektedir.

<div class="loading-container" data-loading-container>

<div class="loading-circle"></div>

</div>

<section style="background-image: url('images/hero-bg.png')">

<br />

<br />

<br />

<br />

<br />

Restoranın adı ve resimleri “VeriGetir()” metodu kullanılarak veri tabanından getirilmiştir.

<%

//Yatay menü Başlangıç

Response.Write("<div class=\"container\">\r\n \r\n <div class=\"breadcrumb\">\r\n <ul class=\"list-unstyled\">\r\n <li><a href=\"#\" target=\"\_self\">MASA REZERVASYON/</a></li>\r\n <li><a href=\"Rezervasyon.aspx\" target=\"\_self\" class=\"active\">" + VeriGetir(1) + "</a></li>\r\n </ul>\r\n </div>");

//Galeri Başlangıç

Response.Write("<table>\r\n <tr>\r\n <td width=\"50%\">\r\n <div class=\"product-detail\">\r\n <div class=\"row\">\r\n \r\n <div class=\"col-12 col-sm-6\">\r\n <div class=\"buyuk-galeri\">\r\n <img src=" + VeriGetir(5) + " alt=\"\" id=\"imgBox\">\r\n </div>\r\n <div class=\"kucuk-resimler\">\r\n <div class=\"resimler active\">\r\n <img src=" + VeriGetir(5) + " alt=\"\" onclick=\"myFunction(this)\">\r\n </div>\r\n <div class=\"resimler\">\r\n <img src=" + VeriGetir(6) + " alt=\"\" onclick=\"myFunction(this)\">\r\n </div>\r\n <div class=\"resimler\">\r\n <img src=" + VeriGetir(7) + " alt=\"\" onclick=\"myFunction(this)\">\r\n </div>\r\n <div class=\"resimler\">\r\n <img src=" + VeriGetir(8) + " alt=\"\" onclick=\"myFunction(this)\">\r\n </div>\r\n </div>\r\n </div>\r\n </div>\r\n </div>\r\n </td>");

//Ürün İsim Başlangıç

Response.Write("<td width=\"50%\" align=\"center\">\r\n <div class=\"product-detail\" >\r\n <div class=\"row\">\r\n <div class=\"col-12 col-sm-6\">\r\n \r\n <div class=\"card-head\">\r\n <h1>" + VeriGetir(1) + "</h1>\r\n <br /><br /><span>Masa Tipi / Kişi Sayısı</span>\r\n \r\n <br />\r\n </div>\r\n ");

%>

Aşağıdaki kodlarda masa seçimi, tarih ve saat seçimi yaptıktan sonra rezervasyon işlemi yapılmaktadır.

<div>

<div class="buton">

<asp:RadioButton CssClass="buton1" ID="RadioButton1" Checked="true" runat="server" Text="2" GroupName="gender" />

<asp:RadioButton CssClass="buton1" ID="RadioButton2" runat="server" Text="4" GroupName="gender" />

<asp:RadioButton CssClass="buton1" ID="RadioButton3" runat="server" Text="6" GroupName="gender" />

<asp:RadioButton CssClass="buton1" ID="RadioButton4" runat="server" Text="8" GroupName="gender" />

<asp:RadioButton CssClass="buton1" ID="RadioButton5" runat="server" Text="10" GroupName="gender" />

</div>

<br />

<br />

<br />

<br />

</div>

<div class="card-head">

<span>Rezervasyon Tarihi</span>

<br />

<table>

<tr>

<td>

<asp:Calendar ID="tarih" BorderColor="#ff9933" BackColor="#ffb07b" DayHeaderStyle-ForeColor="#000000" NextPrevStyle-ForeColor="#f96400" runat="server" ></asp:Calendar>

</td>

<td>

<asp:TextBox ID="saat" CssClass="input-field" BorderStyle="Double" BorderColor="#ff9933" TextMode="Time" runat="server"></asp:TextBox>

</td>

</tr>

</table>

<br />

<br />

<br />

</div>

<div class="etkilesim">

<div class="buton">

<asp:Label ID="hatalbl" runat="server" Text=""></asp:Label>

<asp:Button ID="Button1" CssClass="btn" runat="server" OnClick="Rezervasyon\_Click" Text="REZERVASYON YAP" />

</div>

</div>

Restoranın genel bilgileri “VeriGetir()” metodu kullanılarak veri tabanından getirilmiştir.

<%

Response.Write(" </div>\r\n </div>\r\n </div>\r\n </td>\r\n </tr>\r\n <tr>\r\n <td colspan=\"2\">\r\n\r\n <div class=\"accordion\">\r\n <div class=\"panel-group\" id=\"accordion\" role=\"tablist\" aria-multiselectable=\"true\">");

//RESTORAN ÖZELLİKLERİ

Response.Write("<div class=\"panel panel-default\">\r\n <div class=\"panel-heading\" role=\"tab\" id=\"headingOne\">\r\n <h4 class=\"panel-title\">\r\n <a role=\"button\" data-toggle=\"collapse\" data-parent=\"#accordion\" href=\"#collapseOne\" aria-expanded=\"true\" aria-controls=\"collapseOne\">RESTORAN ÖZELLİKLERİ <span class=\"glyphicon glyphicon-plus\" aria-hidden=\"true\"></span>\r\n </a>\r\n </h4>\r\n </div>\r\n <div id=\"collapseOne\" class=\"panel-collapse collapse in\" role=\"tabpanel\" aria-labelledby=\"headingOne\">\r\n <div class=\"panel-body\">\r\n <p>" + VeriGetir(4) + "</p>\r\n </div>\r\n </div>\r\n </div>");

//MASA DETAYI

Response.Write("<div class=\"panel panel-default\">\r\n <div class=\"panel-heading\" role=\"tab\" id=\"headingTwo\">\r\n <h4 class=\"panel-title\">\r\n <a class=\"collapsed\" role=\"button\" data-toggle=\"collapse\" data-parent=\"#accordion\" href=\"#collapseTwo\" aria-expanded=\"false\" aria-controls=\"collapseTwo\">MASA DETAYI <span class=\"glyphicon glyphicon-plus\" aria-hidden=\"true\"></span>\r\n </a>\r\n </h4>\r\n </div>\r\n <div id=\"collapseTwo\" class=\"panel-collapse collapse\" role=\"tabpanel\" aria-labelledby=\"headingTwo\">\r\n <div class=\"panel-body\">\r\n <p>2, 4, 6, 8 ve 10 Kişilik Masalar Mevcuttur.</p>\r\n </div>\r\n </div>\r\n </div>");

//REZERVASYON VE İPTAL

Response.Write("<div class=\"panel panel-default\">\r\n <div class=\"panel-heading\" role=\"tab\" id=\"headingFour\">\r\n <h4 class=\"panel-title\">\r\n <a class=\"collapsed\" role=\"button\" data-toggle=\"collapse\" data-parent=\"#accordion\" href=\"#collapseFour\" aria-expanded=\"false\" aria-controls=\"collapseFour\">REZERVASYON VE İPTAL <span class=\"glyphicon glyphicon-plus\" aria-hidden=\"true\"></span>\r\n </a>\r\n </h4>\r\n </div>\r\n <div id=\"collapseFour\" class=\"panel-collapse collapse\" role=\"tabpanel\" aria-labelledby=\"headingFour\">\r\n <div class=\"panel-body\">\r\n <p>Detaylı Bilgi İçin İletişim Kanallarından İletişime Geçebilirsiniz.</p>\r\n <ul class=\"footer-list\">\r\n\r\n <li>\r\n <a href=\"mailto:" + VeriGetir(2) + "\" class=\"footer-link\">\r\n <ion-icon name=\"mail\" aria-hidden=\"true\"></ion-icon>\r\n\r\n <span class=\"span\">" + VeriGetir(2) + "</span>\r\n </a>\r\n </li>\r\n\r\n <li>\r\n <a href=\"tel:" + VeriGetir(3) + " \" class=\"footer-link\">\r\n <ion-icon name=\"call\" aria-hidden=\"true\"></ion-icon>\r\n\r\n <span class=\"span\">" + VeriGetir(3) + "</span>\r\n </a>\r\n </li>\r\n </ul>\r\n </div>\r\n </div>");

Response.Write("</div>\r\n </div>\r\n </div>\r\n\r\n </td>\r\n </tr>\r\n </table>\r\n\r\n </div>\r\n </section>");

%>

<!--İçerik Bitiş-->

<!--Script Başlangıç-->

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

<script src="java/rezervasyon.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js" integrity="sha384-Tc5IQib027qvyjSMfHjOMaLkfuWVxZxUPnCJA7l2mCWNIpG9mGCD8wGNIcPD7Txa" crossorigin="anonymous"></script>

**Rezervasyon (cs)**

public partial class Rezervasyon : System.Web.UI.Page

{

Bilgi bilgi = new Bilgi();

RezervasyonGun rezervasyon = new RezervasyonGun();

Sayfa yüklendiğinde Kullanıcı Session’u boş ise giriş sayfasına yönlendirir. Restoran Session’u boş ise Restoranlar sayfasına yönlendirir.

protected void Page\_Load(object sender, EventArgs e)

{

if (Session["Kullanici"] == null)

{

Response.Redirect("giris.aspx");

}

else if (Session["Restoran"] == null)

{

Response.Redirect("Restaurantlar.aspx");

}

}

Bu metot sayesinde Restoran Session’u içindeki restoran ID bilgisi bilgi sınıfındaki “RezervasyonGetir()” metoduna gönderilir. Bu metot sayesinde seçilen restoranın bilgileri DataTable’a atılır. Gelen parametre bilgisine göre veri geri gönderilir.

public string VeriGetir(int a)

{

DataTable dt = new DataTable();

dt = bilgi.RezervasyonGetir(Convert.ToInt32(Session["Restoran"]));

return dt.Rows[0][a].ToString();

}

Bu metot sayesinde seçilen RadioButton ile masa bilgisi geri gönderilir.

public int Masa()

{

int a;

if (RadioButton1.Checked)

a = 2;

else if (RadioButton2.Checked)

a = 4;

else if (RadioButton3.Checked)

a = 6;

else if (RadioButton4.Checked)

a = 8;

else if (RadioButton5.Checked)

a = 10;

else

a = 0;

return a;

}

Bu metot sayesinde Rezervasyon Yap butonuna tıklandığında tarih ve saat bilgileri kontrol edilir. Boş ise hata mesajı verir. Değil ise veriler RezervasyonGun sınıfındaki “RezervasyonYap()” metoduna gönderilir. Geriye hata dönmezse kayıt yapıldığına dair mesaj verir.

protected void Rezervasyon\_Click(object sender, EventArgs e)

{

if (tarih.SelectedDate<DateTime.Now)

{

hatalbl.Text = "Lütfen İleri Bir Tarih Seçimi Yapınız.";

}

else

{

if (saat.Text=="")

{

hatalbl.Text = "Lütfen Saat Seçimi Yapınız.";

}

else

{

string tarihYeni = (tarih.SelectedDate.ToString("yyyy-MM-dd") + " " + saat.Text);

bool hata = rezervasyon.RezervasyonYap(Convert.ToInt32(Session["ID"]), Convert.ToInt32(Session["Restorant"]), Masa(), tarihYeni);

if (hata == true)

{

hatalbl.Text = "Rezervasyon Oluşturulamadı.";

}

else

{

hatalbl.Text = "Rezsrvasyon Başarıyla Oluşturuldu.";

}

}

}

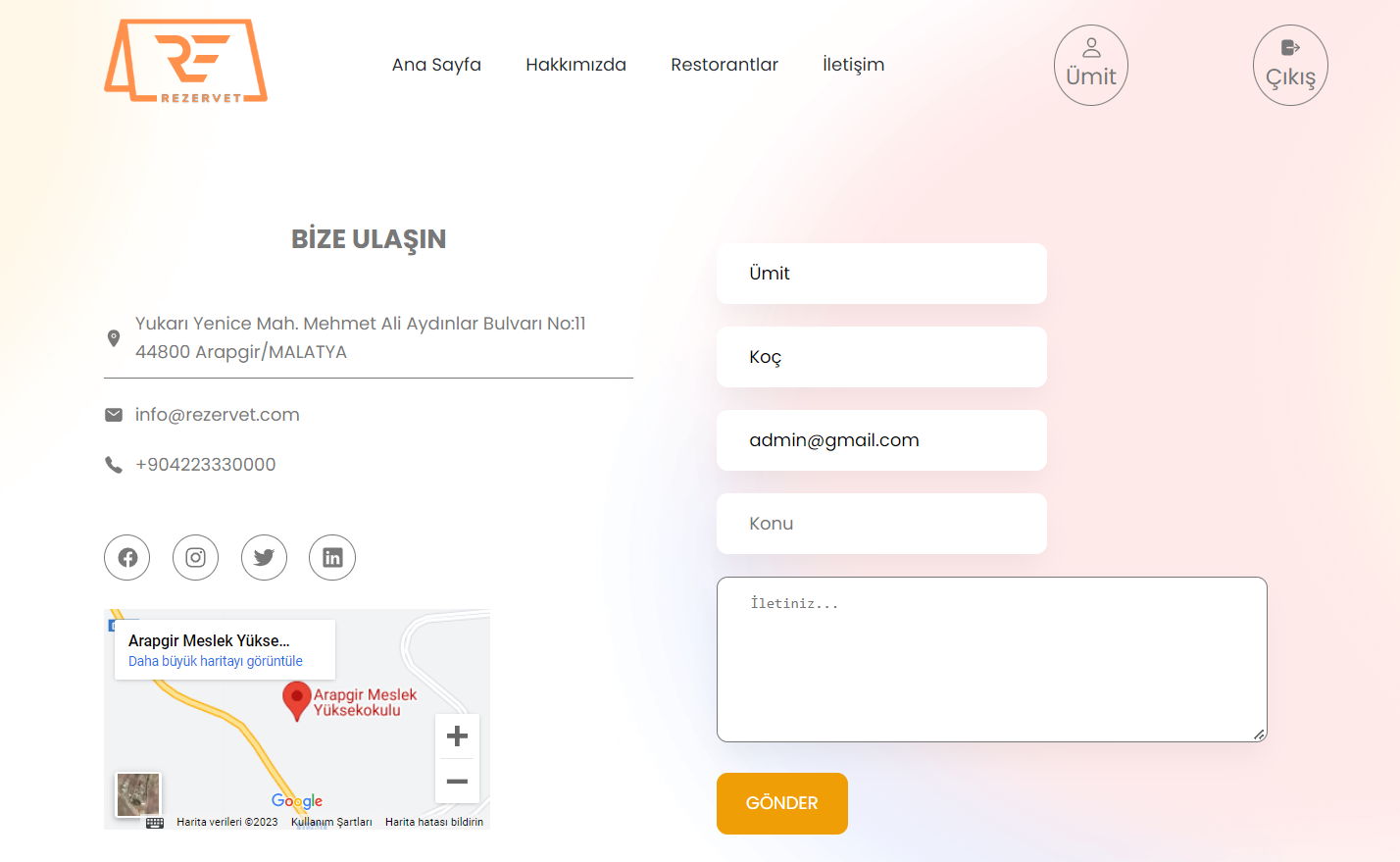
}

}

|  |  |
| --- | --- |
| public DataTable RezervasyonGetir()  {  DataTable dt = new DataTable();  try  {  using (SqlConnection baglanti = b.BaglantiOlustur())  {  using (SqlCommand komut = new SqlCommand("rezervasyon\_sayfa\_getir\_usp", baglanti))  {  komut.CommandType = CommandType.StoredProcedure;  baglanti.Open();  SqlDataAdapter da;  DataSet ds = new DataSet();  da = new SqlDataAdapter(komut);  da.Fill(ds, "AnaSayfa");  dt = ds.Tables["AnaSayfa"];  baglanti.Close();  }  }  }  catch (Exception hata)  {  }  return dt;  } | Bu “RezervasyonGetir()” metodu sayesinde veri tabanından “rezervasyon\_sayfa  \_getir\_usp” prosedürü çalıştırılıp bilgiler DataTable ile geri gönderilir. |
| CREATE proc [dbo].[rezervasyon\_sayfa\_getir\_usp]  (  @ID int    )  as  begin  select r.ID,r.isim,r.eposta,r.telefon,r.bilgi,res.resim1,res.resim2,res.resim3,res.resim4 from Restoranlar r,RestorantResim res where r.resimID=res.ID and r.ID=@ID  end | “rezervasyon\_sayfa \_getir\_usp” prosedürü |

|  |  |
| --- | --- |
| public bool RezervasyonYap(int kullaniciID, int restoranID, int kisi, string tarih)  {  bool hataOlustuMu = false;  try  {  using (SqlConnection baglanti = b.BaglantiOlustur())  {  using (SqlCommand komut = new SqlCommand("rezervasyon\_ekle\_usp", baglanti))  {  komut.CommandType = CommandType.StoredProcedure;  komut.Parameters.AddWithValue("@kullaniciID", kullaniciID);  komut.Parameters.AddWithValue("@restoranID", restoranID);  komut.Parameters.AddWithValue("@kisi", kisi);  komut.Parameters.AddWithValue("@tarih", tarih);  baglanti.Open();  komut.ExecuteNonQuery();  baglanti.Close();  }  }  }  catch (Exception hata)  {  hataOlustuMu = true;  }  return hataOlustuMu;  } | Bu “RezervasyonYap()” metodu sayesinde veri tabanından “rezervasyon\_ekle\_usp” prosedürü çalıştırılıp içerisine gönderilen parametreler ile rezervasyon tablosuna yeni ekleme yapılır. Eğer hata olursa geriye hata gönderir. |
| CREATE proc [dbo].[rezervasyon\_ekle\_usp]  @kullaniciID int,  @restoranID int,  @kisi bigint,  @tarih datetime  as  insert into Rezervasyon  (KullaniciID,  RestorantID,  KisiSayisi,  RezervasyonTarih,Onay)  values  (@kullaniciID,@restoranID,@kisi,@tarih,0) | “rezervasyon\_ekle\_usp” prosedürü |

**REZERVET- İletişim**



Bu bölümde web sitesinin İletişim Sayfası oluşturulmuştur. Bu sayfada REZERVET hakkında iletişim kanalları bulunmaktadır. Bu sayfada kullanıcının iletişim sağlayabileceği form bölümü de bulunmaktadır.

<div class="loading-container" data-loading-container>

<div class="loading-circle"></div>

</div>

İletişim kanalları “VeriGetir()” metodu kullanılarak veri tabanından getirilmiştir.

<%

Response.Write("<section class=\"section contact\" aria-labelledby=\"top-contact\" style=\"background-image: url('images/hero-bg.png')\"><div class=\"container\"><br /><ul class=\"grid-list\"><li data-reveal=\"left\"><br /><br /><br /><h1 align=\"center\">BİZE ULAŞIN</h1><br /></asp:TextBox><div><ul class=\"footer-list\">");

//Adres Satırı

Response.Write("<li><a href=" + VeriGetir(20) + " class=\"footer-link\"><address class=\"address\"><ion-icon name=\"location\" aria-hidden=\"true\"></ion-icon><span class=\"span\">" + VeriGetir(4) + "</span></address></a></li>");

//E Posta Satırı

Response.Write("<li><a href=mailto:" + VeriGetir(5) + " class=\"footer-link\"><ion-icon name=\"mail\" aria-hidden=\"true\"></ion-icon><span class=\"span\">" + VeriGetir(5) + "</span></a></li>");

//Telefon Satırı

Response.Write("<li><a href=tel:" + VeriGetir(6) + " class=\"footer-link\"><ion-icon name=\"call\" aria-hidden=\"true\"></ion-icon><span class=\"span\">" + VeriGetir(6) + "</span></a></li>");

//Sosyal Medya Satırı

Response.Write("<li><ul class=\"social-list\"><li><a href=" + VeriGetir(7) + " class=\"social-link\"><ion-icon name=\"logo-facebook\"></ion-icon></a></li><li><a href=" + VeriGetir(8) + " class=\"social-link\"><ion-icon name=\"logo-instagram\"></ion-icon></a></li><li><a href=" + VeriGetir(9) + " class=\"social-link\"><ion-icon name=\"logo-twitter\"></ion-icon></a></li><li><a href=" + VeriGetir(10) + " class=\"social-link\"><ion-icon name=\"logo-linkedin\"></ion-icon></a></li></ul></li></ul></div><br />");

//Google Map Satırı

Response.Write("<div>" + VeriGetir(11) + "</div></li>");

%>

İletişim formu bu kısımda bulunmaktadır.

<li data-reveal="right">

<section class="section newsletter" aria-label="newsletter">

<div class="container">

<div class="newsletter-content">

<div class="input-wrapper">

<asp:TextBox ID="adtxt" runat="server" Width="300px" CssClass="input-field" placeholder="Adınız"></asp:TextBox>

<asp:TextBox ID="soyadtxt" CssClass="input-field" runat="server" placeholder="Soyadınız" Width="300px"></asp:TextBox>

<asp:TextBox ID="epostatxt" CssClass="input-field" TextMode="Email" runat="server" placeholder="E-mail adresiniz" Width="300px"></asp:TextBox>

<asp:TextBox ID="konutxt" CssClass="input-field" MaxLength="100" runat="server" placeholder="Konu" Width="300px"></asp:TextBox>

<asp:TextBox ID="iletitxt" CssClass="input-field" MaxLength="500" runat="server" TextMode="MultiLine" placeholder="İletiniz..." Width="500px" Height="150px"></asp:TextBox>

<asp:Button ID="Button1" runat="server" Text="Gönder" CssClass="btn " OnClick="Button1\_Click" /><br />

<asp:Label ID="hatalbl" runat="server" Text=""></asp:Label>

</div>

</div>

</div>

</section>

</li>

</ul>

</div>

</section>

<script src="java/script.js"></script>

**İletişim (cs)**

public partial class Iletisim : System.Web.UI.Page

{

Bilgi bilgi = new Bilgi();

IletisimGun iletisim = new IletisimGun();

KullaniciGun kullanici = new KullaniciGun();

Sayfa yüklendiğinde kullanıcı giriş yapmışsa KullaniciGun sınıfındaki “TekKullaniciGoster()” metoduna kullanıcının ID verisi gönderilir ve iletişim formunun kullanıcıya ait gerekli bölümleri doldurulur.

protected void Page\_Load(object sender, EventArgs e)

{

if (!IsPostBack)

{

if (Session["ID"] != null)

{

DataTable dt = new DataTable();

dt = kullanici.TekKullaniciGoster(Convert.ToInt32(Session["ID"]));

adtxt.Text = dt.Rows[0][1].ToString();

soyadtxt.Text = dt.Rows[0][2].ToString();

epostatxt.Text = dt.Rows[0][5].ToString();

}

}

}

Bu metot sayesinde gelen parametre değeri bilgi sınıfındaki “SiteGenel()” metoduna gönderilir. Bu metot sayesinde veriler DataTable’a atılır. Gelen parametre bilgisine göre veri geri gönderilir.

public string VeriGetir(int a)

{

DataTable dt = new DataTable();

dt = bilgi.SiteGenel();

return dt.Rows[0][a].ToString();

}

Bu metot sayesinde iletişim formu doldurulurken tüm alanların doldurulduğu kontrol edilir. Tüm alanlar dolu ise IletisimGun sınıfındaki “BizeUlasin()” metoduna veriler gönderilir. Geriye hata dönmezse gönderi yapıldığına dair mesaj verir ve sonrasında kullanıcı bilgisine göre ilgili textleri temizler.

protected void Button1\_Click(object sender, EventArgs e)

{

if (adtxt.Text=="" || soyadtxt.Text == "" || epostatxt.Text == "" || konutxt.Text == "" || iletitxt.Text == "")

{

hatalbl.Text = "Lütfen Tüm Alanları Doldurun.";

}

else

{

bool hata=iletisim.BizeUlasin(adtxt.Text, soyadtxt.Text, epostatxt.Text, konutxt.Text, iletitxt.Text);

if (hata==true)

{

hatalbl.Text = "İletiniz Gönderilemedi !";

}

else

{

hatalbl.Text = "İletiniz Başarıyla Gönderildi.";

if (Session["ID"] != null)

{

Temizle();

}

else

{

adtxt.Text = "";

soyadtxt.Text = "";

epostatxt.Text = "";

Temizle();

}

}

}

}

Bu metot sayesinde konu ve ileti textleri temizlenir.

void Temizle()

{

konutxt.Text = "";

iletitxt.Text = "";

}

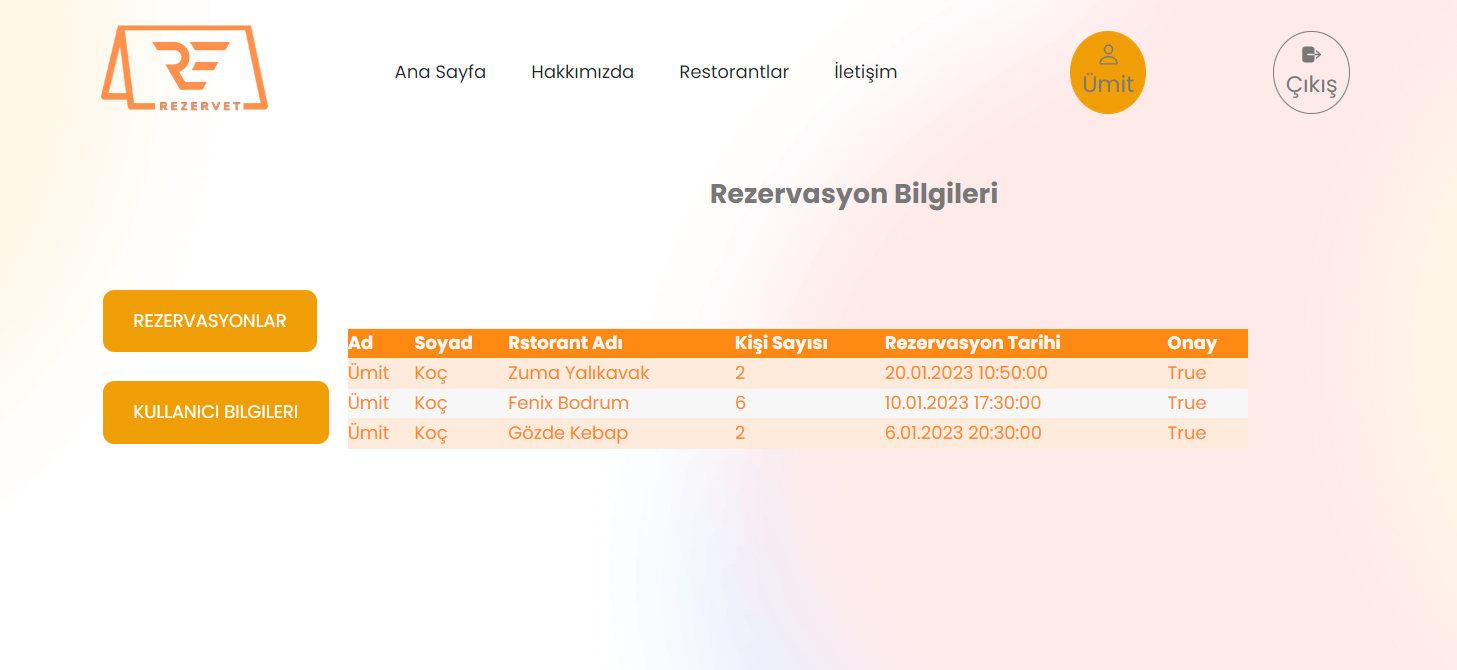
}

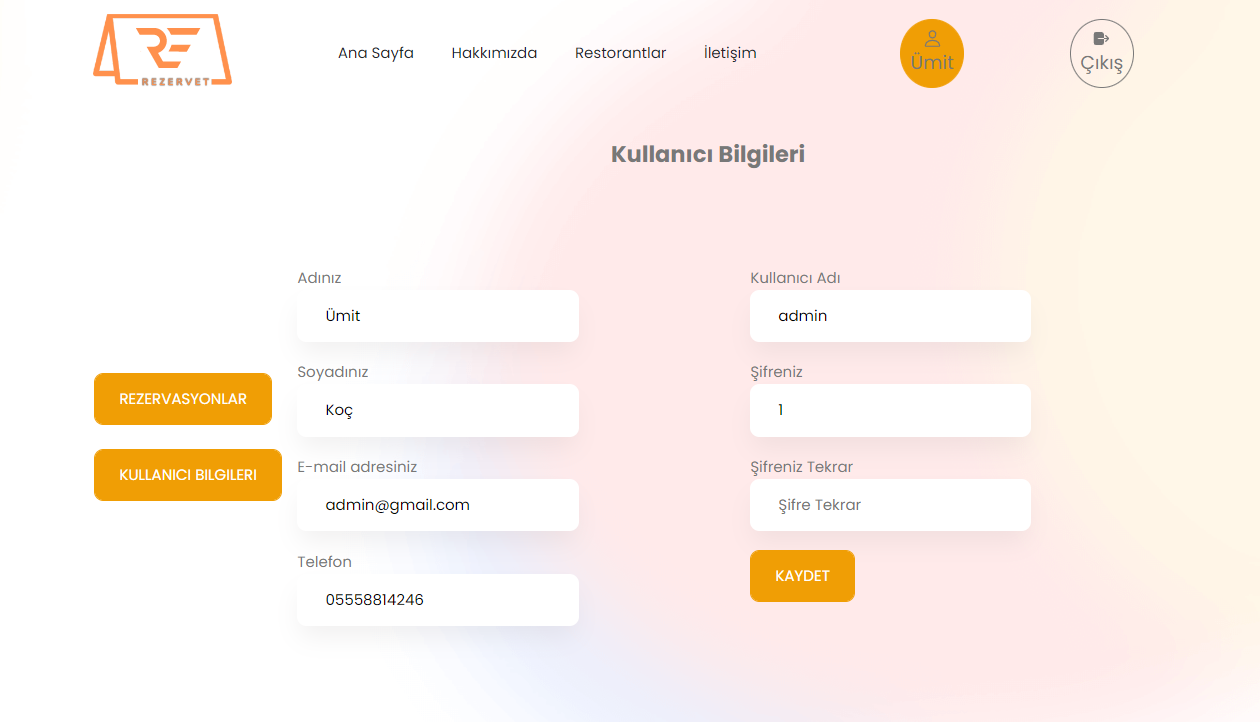
|  |  |
| --- | --- |
| public DataTable TekKullaniciGoster(int ID)  {  DataTable dt = new DataTable();  try  {  using (SqlConnection baglanti = b.BaglantiOlustur())  {  using (SqlCommand komut = new SqlCommand("tek\_kullanici\_getir\_usp", baglanti))  {  komut.CommandType = CommandType.StoredProcedure;  komut.Parameters.AddWithValue("@ID", ID);  baglanti.Open();  SqlDataAdapter da;  DataSet ds = new DataSet();  da = new SqlDataAdapter(komut);  da.Fill(ds, "Kullanicilar");  dt = ds.Tables["Kullanicilar"];  baglanti.Close();  }  }  }  catch (Exception hata)  {  }  return dt;  } | Bu “TekKullaniciGoster()” metodu sayesinde veri tabanından “tek\_kullanici\_getir\_usp” prosedürü çalıştırılıp içerisine gönderilen parametreler ile kullanıcının bilgileri getirilir. Bu bilgiler DataTable ile geri gönderilir. |
| CREATE proc [dbo].[tek\_kullanici\_getir\_usp]  @ID int  as  begin  select \* from Kullanicilar where ID=@ID  end | “tek\_kullanici\_getir\_usp” prosedürü |

|  |  |
| --- | --- |
| public DataTable SiteGenel()  {  DataTable dt = new DataTable();  try  {  using (SqlConnection baglanti = b.BaglantiOlustur())  {  using (SqlCommand komut = new SqlCommand("site\_genel\_usp", baglanti))  {  komut.CommandType = CommandType.StoredProcedure;  baglanti.Open();  SqlDataAdapter da;  DataSet ds = new DataSet();  da = new SqlDataAdapter(komut);  da.Fill(ds, "Site");  dt = ds.Tables["Site"];  baglanti.Close();  }  }  }  catch (Exception hata)  {  }  return dt;  } | Bu “SiteGenel()” metodu sayesinde veri tabanından “site\_genel\_usp” prosedürü çalıştırılıp bilgiler DataTable ile geri gönderilir. |
| CREATE proc [dbo].[site\_genel\_usp]  as  begin  select \* from SiteGenel  end | “site\_genel\_usp” prosedürü |

|  |  |
| --- | --- |
| public bool BizeUlasin(string ad, string soyad, string eposta, string konu, string ileti)  {  bool hataOlustuMu = false;  try  {  using (SqlConnection baglanti = b.BaglantiOlustur())  {  using (SqlCommand komut = new SqlCommand("ileti\_ekle\_usp", baglanti))  {  komut.CommandType = CommandType.StoredProcedure;  komut.Parameters.AddWithValue("@ad", ad);  komut.Parameters.AddWithValue("@soyad", soyad);  komut.Parameters.AddWithValue("@eposta", eposta);  komut.Parameters.AddWithValue("@konu", konu);  komut.Parameters.AddWithValue("@ileti", ileti);  baglanti.Open();  komut.ExecuteNonQuery();  baglanti.Close();  }  }  }  catch (Exception hata)  {  hataOlustuMu = true;  }  return hataOlustuMu;  } | Bu “BizeUlasin()” metodu sayesinde veri tabanından “ileti\_ekle\_usp” prosedürü çalıştırılıp içerisine gönderilen parametreler ile iletisim tablosuna yeni veri eklemesi yapılır. Eğer hata olursa geriye hata gönderir. |
| CREATE proc [dbo].[ileti\_ekle\_usp]  @ad varchar(50),  @soyad varchar(50),  @eposta varchar(70),  @konu varchar(100),  @ileti varchar(500)  as  insert into Iletisim  (isim,  soyisim,  eposta,  konu,  ileti,  okundu)  values  (@ad,@soyad,@eposta,@konu,@ileti,0) | “ileti\_ekle\_usp” prosedürü |

**REZERVET- Hesap**





Bu bölümde web sitesinin kullanıcılara ait Hesap Sayfası oluşturulmuştur. Bu sayfada kullanıcılar yaptığı rezervasyonları görüntüleyebilmektedir. Kendine ait kullanıcı bilgilerini düzenleyebilmektedir.

<div class="loading-container" data-loading-container>

<div class="loading-circle"></div>

</div>

<section class="section contact" aria-labelledby="top-contact" style="background-image: url('images/hero-bg.png')">

<div class="container">

<br />

<br />

<table>

<tr>

<td>

<asp:Button ID="p1btn" runat="server" CssClass="btn has-after" OnClick="p1btn\_Click" Text="Rezervasyonlar" /><br />

<asp:Button ID="p2btn" runat="server" CssClass="btn has-after" OnClick="p2btn\_Click" Text="Kullanıcı Bilgileri" />

</td>

<td>

Rezervasyon bilgilerinin görüntülendiği bölüm .

<asp:Panel ID="Panel1" runat="server">

<center>

<h2>Rezervasyon Bilgileri</h2>

</center>

<ul class="grid-list">

<li data-reveal="left">

<section class="section newsletter" aria-label="newsletter">

<div class="container">

<div class="newsletter-content">

<div class="input-wrapper">

<asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False" BackColor="White" BorderColor="#feebdc" BorderStyle="None" BorderWidth="1px" CellPadding="3" GridLines="Horizontal" AllowPaging="true" PageSize="5" Width="800" >

<PagerSettings Mode="NumericFirstLast" FirstPageText="İlk" LastPageText="Son" />

<AlternatingRowStyle BackColor="#F7F7F7" />

<Columns>

<asp:BoundField DataField="isim" HeaderText="Ad" InsertVisible="False" ReadOnly="True" SortExpression="isim" />

<asp:BoundField DataField="soyisim" HeaderText="Soyad" InsertVisible="False" ReadOnly="True" SortExpression="soyisim" />

<asp:BoundField DataField="resIsim" HeaderText="Rstorant Adı" InsertVisible="False" ReadOnly="True" SortExpression="resIsim" />

<asp:BoundField DataField="KisiSayisi" HeaderText="Kişi Sayısı" InsertVisible="False" ReadOnly="True" SortExpression="KisiSayisi" />

<asp:BoundField DataField="RezervasyonTarih" HeaderText="Rezervasyon Tarihi" InsertVisible="False" ReadOnly="True" SortExpression="RezervasyonTarih" />

<asp:BoundField DataField="Onay" HeaderText="Onay" InsertVisible="False" ReadOnly="True" SortExpression="Onay" />

</Columns>

<FooterStyle BackColor="#B5C7DE" ForeColor="#ff8913" />

<HeaderStyle BackColor="#ff8913" Font-Bold="True" ForeColor="#F7F7F7" />

<PagerStyle BackColor="#feebdc" ForeColor="#ff8913" HorizontalAlign="Right" />

<RowStyle BackColor="#feebdc" ForeColor="#ff8913" />

<SelectedRowStyle BackColor="#738A9C" Font-Bold="True" ForeColor="#F7F7F7" />

<SortedAscendingCellStyle BackColor="#F4F4FD" />

<SortedAscendingHeaderStyle BackColor="#5A4C9D" />

<SortedDescendingCellStyle BackColor="#D8D8F0" />

<SortedDescendingHeaderStyle BackColor="#3E3277" />

</asp:GridView>

</div>

</div>

</div>

</section>

</li>

</ul>

</asp:Panel>

Kullanıcı bilgilerinin düzenlendiği bölüm.

<asp:Panel ID="Panel2" Visible="false" runat="server">

<center>

<h2>Kullanıcı Bilgileri</h2>

</center>

<ul class="grid-list">

<li data-reveal="left">

<section class="section newsletter" aria-label="newsletter">

<div class="container">

<div class="newsletter-content">

<div class="input-wrapper">

<asp:Label ID="Label1" runat="server" Text="Adınız"></asp:Label>

<asp:TextBox ID="adtxt" runat="server" Width="300px" CssClass="input-field" placeholder="Adınız"></asp:TextBox>

<asp:Label ID="Label2" runat="server" Text="Soyadınız"></asp:Label>

<asp:TextBox ID="soyadtxt" CssClass="input-field" runat="server" placeholder="Soyadınız" Width="300px"></asp:TextBox>

<asp:Label ID="Label3" runat="server" Text="E-mail adresiniz"></asp:Label>

<asp:TextBox ID="epostatxt" CssClass="input-field" runat="server" placeholder="E-mail adresiniz" Width="300px"></asp:TextBox>

<asp:Label ID="Label4" runat="server" Text="Telefon"></asp:Label>

<asp:TextBox ID="teltxt" CssClass="input-field" runat="server" placeholder="Telefon" Width="300px"></asp:TextBox>

</div>

</div>

</div>

</section>

</li>

<li data-reveal="right">

<section class="section newsletter" aria-label="newsletter">

<div class="container">

<div class="newsletter-content">

<div class="input-wrapper">

<asp:Label ID="Label5" runat="server" Text="Kullanıcı Adı"></asp:Label>

<asp:TextBox ID="kaditxt" CssClass="input-field" runat="server" placeholder="Kullanıcı Adı" Width="300px"></asp:TextBox>

<asp:Label ID="Label6" runat="server" Text="Şifreniz"></asp:Label>

<asp:TextBox ID="sifretxt" CssClass="input-field" runat="server" placeholder="Şifre" Width="300px"></asp:TextBox>

<asp:Label ID="Label7" runat="server" Text="Şifreniz Tekrar"></asp:Label>

<asp:TextBox ID="sifretxt2" CssClass="input-field" runat="server" placeholder="Şifre Tekrar" Width="300px"></asp:TextBox>

<asp:Button ID="Button1" runat="server" Text="KAYDET" CssClass="btn has-after" OnClick="Button1\_Click" />

<asp:Label ID="hatalbl" runat="server" Text=""></asp:Label>

</div>

</div>

</div>

</section>

</li>

</ul>

</asp:Panel>

</td>

</tr>

</table>

</div>

</section>

<script src="java/script.js"></script>

**Hesap (cs)**

public partial class Hesabım : System.Web.UI.Page

{

KullaniciGun kullanici = new KullaniciGun();

RezervasyonGun rezervasyon = new RezervasyonGun();

Sayfa yüklendiğinde kullanıcı giriş yapmışsa “VeriGetir()” ve “VeriGetir2()” metodu ile veriler getirilir.

protected void Page\_Load(object sender, EventArgs e)

{

if (Session["Kullanici"] == null)

{

Response.Redirect("default.aspx");

}

if (!IsPostBack)

{

VeriGetir();

VeriGetir2();

}

}

Bu metot sayesinde DataTable’la KullaniciGun sınıfındaki “TekKullaniciGoster()” metoduna kullanıcı ID verisi gönderilerek veriler eklenir. Bu gelen veriler kullanıcı bilgilerinin düzenlenebildiği bölümdeki textlerde gösterilir.

public void VeriGetir()

{

DataTable dt = new DataTable();

dt = kullanici.TekKullaniciGoster(Convert.ToInt32(Session["ID"]));

adtxt.Text = dt.Rows[0][1].ToString();

soyadtxt.Text = dt.Rows[0][2].ToString();

kaditxt.Text = dt.Rows[0][3].ToString();

sifretxt.Text = dt.Rows[0][4].ToString();

epostatxt.Text = dt.Rows[0][5].ToString();

teltxt.Text = dt.Rows[0][6].ToString();

sifretxt2.Text = "";

}

Bu metot sayesinde DataTable’la RezervasyonGun sınıfındaki “KullaniciRezervasyonGetir()” metoduna kullanıcı ID verisi gönderilerek veriler eklenir. Bu gelen veriler kullanıcı bilgilerinin düzenlenebildiği bölümdeki textlerde gösterilir.

void VeriGetir2()

{

DataTable dt = new DataTable();

dt = rezervasyon.KullaniciRezervasyonGetir(Convert.ToInt32(Session["ID"]));

GridView1.DataSource = dt;

GridView1.DataBind();

}

Bu butona tıklandığında kullanici bilgileri düzenlenirken öncelikle kullanıcının şifreyi tekrar girmesi beklenir. Daha sonra tüm alanların doldurulduğu kontrol edilir. Tüm alanlar dolu ise KullaniciGun sınıfındaki “KullaniciGuncelle2()” metoduna veriler gönderilir. Geriye hata dönmezse güncelleme yapıldığına dair mesaj verir ve sonrasında kullanıcı bilgileri tekrar “VeriGetir()” metodu ile textlerde gösterilir.

protected void Button1\_Click(object sender, EventArgs e)

{

if (sifretxt2.Text == "")

{

hatalbl.Text = "Kaydınızı Güncellemek İçin Şifrenizi Tekrar Girin.";

}

else

{

if (sifretxt.Text != sifretxt2.Text)

{

hatalbl.Text = "Kaydınızı Güncellemek İçin Şifrelerin Aynı Olması Gerekir.";

}

else

{

if (adtxt.Text == "" || soyadtxt.Text == "" || epostatxt.Text == "" || teltxt.Text == "" || kaditxt.Text == "" || sifretxt.Text == "")

{

hatalbl.Text = "Kaydınızı Güncellemek İçin Tüm Alanları Doldurun.";

}

else

{

bool hata = kullanici.KullaniciGuncelle2(Convert.ToInt32(Session["ID"]), adtxt.Text, soyadtxt.Text, kaditxt.Text, sifretxt.Text, epostatxt.Text, teltxt.Text);

if (hata == false)

{

hatalbl.Text = "Kayıt Başarıyla Güncellendi";

}

else

{

hatalbl.Text = "Kayıt Güncellenemedi.";

}

}

}

}

VeriGetir();

}

Bu butona tıklandığında Rezervasyonların bulunduğu panel aktif olur.

protected void p1btn\_Click(object sender, EventArgs e)

{

Panel1.Visible = true;

Panel2.Visible = false;

}

Bu butona tıklandığında Kullanıcı bilgilerinin düzenlendiği panel aktif olur.

protected void p2btn\_Click(object sender, EventArgs e)

{

Panel1.Visible = false;

Panel2.Visible = true;

}

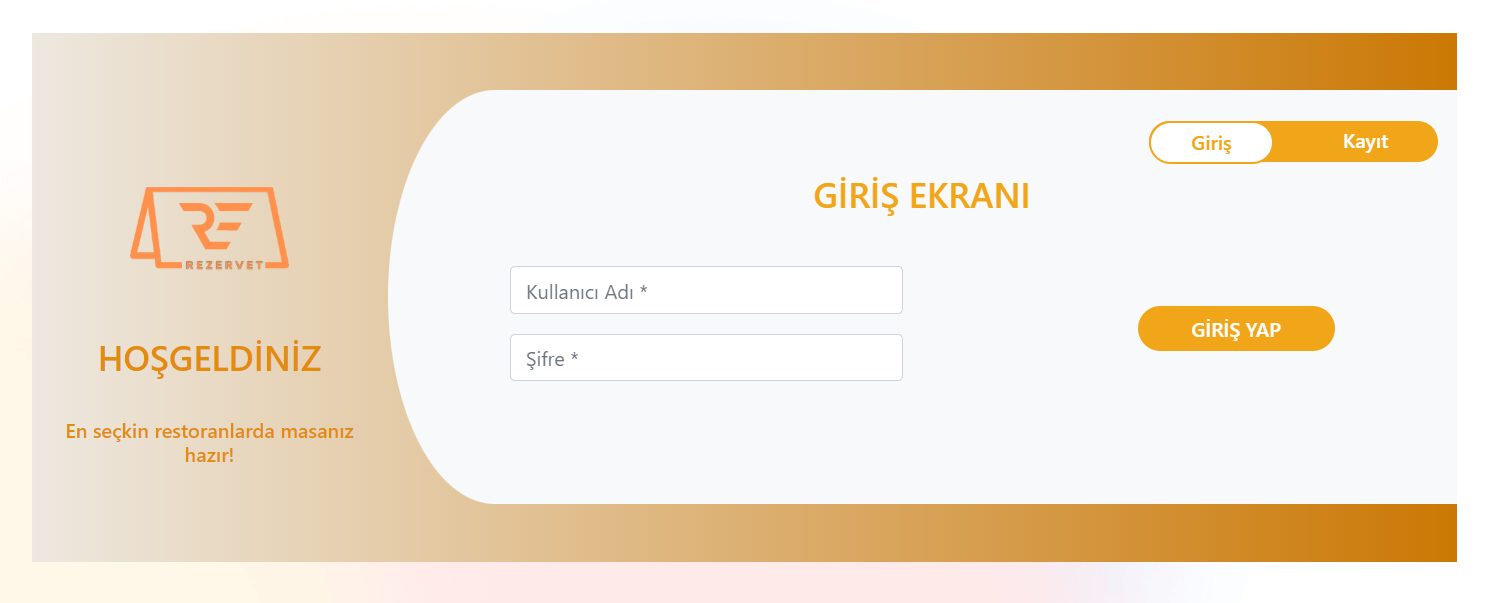
}

|  |  |
| --- | --- |
| public DataTable TekKullaniciGoster(int ID)  {  DataTable dt = new DataTable();  try  {  using (SqlConnection baglanti = b.BaglantiOlustur())  {  using (SqlCommand komut = new SqlCommand("tek\_kullanici\_getir\_usp", baglanti))  {  komut.CommandType = CommandType.StoredProcedure;  komut.Parameters.AddWithValue("@ID", ID);  baglanti.Open();  SqlDataAdapter da;  DataSet ds = new DataSet();  da = new SqlDataAdapter(komut);  da.Fill(ds, "Kullanicilar");  dt = ds.Tables["Kullanicilar"];  baglanti.Close();  }  }  }  catch (Exception hata)  {  }  return dt;  } | Bu “TekKullaniciGoster()” metodu sayesinde veri tabanından “tek\_kullanici\_getir\_usp” prosedürü çalıştırılıp içerisine gönderilen parametreler ile kullanıcının bilgileri getirilir. Bu bilgiler DataTable ile geri gönderilir. |
| CREATE proc [dbo].[tek\_kullanici\_getir\_usp]  @ID int  as  begin  select \* from Kullanicilar where ID=@ID  end | “tek\_kullanici\_getir\_usp” prosedürü |

|  |  |
| --- | --- |
| public DataTable KullaniciRezervasyonGetir(int ID)  {  DataTable dt = new DataTable();  try  {  using (SqlConnection baglanti = b.BaglantiOlustur())  {  using (SqlCommand komut = new SqlCommand("rezervasyon\_kullanici\_getir\_usp", baglanti))  {  komut.CommandType = CommandType.StoredProcedure;  komut.Parameters.AddWithValue("@ID", ID);  baglanti.Open();  SqlDataAdapter da;  DataSet ds = new DataSet();  da = new SqlDataAdapter(komut);  da.Fill(ds, "Rezervasyon");  dt = ds.Tables["Rezervasyon"];  baglanti.Close();  }  }  }  catch (Exception hata)  {  }  return dt;  } | Bu “KullaniciRezervasyonGetir()” metodu sayesinde veri tabanından “rezervasyon\_kullanici\_getir\_usp” prosedürü çalıştırılıp içerisine gönderilen parametre ile kullanıcının rezervasyon bilgileri getirilir. Bu bilgiler DataTable ile geri gönderilir. |
| CREATE proc [dbo].[rezervasyon\_kullanici\_getir\_usp]  @ID int  as  begin  select k.isim,k.soyisim,r.resIsim,rez.KisiSayisi,rez.RezervasyonTarih,rez.Onay from Kullanicilar k,Restoranlar r, Rezervasyon rez where k.ID=rez.KullaniciID and r.ID=rez.RestorantID and k.ID=@ID order by rez.RezervasyonTarih desc  end | “rezervasyon\_kullanici\_getir\_usp” prosedürü |

|  |  |
| --- | --- |
| public bool KullaniciGuncelle2(int ID, string isim, string soyisim, string kullaniciAdi, string sifre,  string eposta, string telefon)  {  bool hataOlustuMu = false;  try  {  using (SqlConnection baglanti = b.BaglantiOlustur())  {  using (SqlCommand komut = new SqlCommand("kullanici\_guncelle\_usp2", baglanti))  {  komut.CommandType = CommandType.StoredProcedure;  komut.Parameters.AddWithValue("@ID", ID);  komut.Parameters.AddWithValue("@isim", isim);  komut.Parameters.AddWithValue("@soyisim", soyisim);  komut.Parameters.AddWithValue("@kullaniciAdi", kullaniciAdi);  komut.Parameters.AddWithValue("@sifre", sifre);  komut.Parameters.AddWithValue("@eposta", eposta);  komut.Parameters.AddWithValue("@telefon", telefon);    baglanti.Open();  komut.ExecuteNonQuery();  baglanti.Close();  }  }  }  catch (Exception hata)  {  hataOlustuMu = true;  }  return hataOlustuMu;  } | Bu “KullaniciGuncelle2()” metodu sayesinde veri tabanından “kullanici\_guncelle \_usp2” prosedürü çalıştırılıp içerisine gönderilen parametreler ile Kullanıcı tablosundaki kullanıcının verilerinde düzenleme yapılır. Eğer hata olursa geriye hata gönderir. |
| CREATE proc [dbo].[kullanici\_guncelle\_usp2]  @ID int,  @isim varchar(50),  @soyisim varchar(50),  @kullaniciAdi varchar(50),  @sifre varchar(30),  @eposta varchar(70),  @telefon varchar(15)  as  update Kullanicilar set  isim=@isim,  soyisim=@soyisim,  kullaniciAdi=@kullaniciAdi,  sifre=@sifre,  eposta=@eposta,  telefon=@telefon  where ID=@ID | “kullanici\_guncelle \_usp2” prosedürü |

**REZERVET- Kayıt Giriş**





Bu bölümde kullanıcıların kayıt olabildiği ve giriş yapabildiği sayfa oluşturulmuştur.

<!DOCTYPE html>

<html lang="tr">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Kayıt & Giriş</title>

</head>

<link rel="stylesheet" href="css/giris.css">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/4.3.1/css/bootstrap.min.css">

<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/4.3.1/js/bootstrap.min.js"></script>

<body style="background-image: url('images/hero-bg2.png')">

<form runat="server" method="post">

<div class="container register">

<div class="row">

<div class="col-md-3 register-left">

<a href="default.aspx">

<img width="100px" height="100px" src="images/favicon.svg" alt="" />

</a>

<h3>HOŞGELDİNİZ</h3>

<br>

<h6>En seçkin restoranlarda masanız hazır!</h6>

</div>

<div class="col-md-9 register-right">

<ul class="nav nav-tabs nav-justified" id="myTab" role="tablist">

<li class="nav-item">

<a class="nav-link active" id="home-tab" data-toggle="tab" href="#home" role="tab" aria-controls="home" aria-selected="true">Giriş</a>

</li>

<li class="nav-item">

<a class="nav-link" id="profile-tab" data-toggle="tab" href="#profile" role="tab" aria-controls="profile" aria-selected="false">Kayıt</a>

</li>

</ul>

<div class="tab-content" id="myTabContent">

<div class="tab-pane fade show active" id="home" role="tabpanel" aria-labelledby="home-tab">

Kullanıcıların giriş yaptığı bölüm.

<h3 class="register-heading">GİRİŞ EKRANI</h3>

<div class="row register-form">

<div class="col-md-6">

<div class="form-group">

<asp:TextBox ID="kadi" CssClass="form-control" runat="server" placeholder="Kullanıcı Adı \*"></asp:TextBox>

</div>

<div class="form-group">

<asp:TextBox TextMode="Password" ID="sifre" CssClass="form-control" runat="server" placeholder="Şifre \*"></asp:TextBox>

</div>

<div class="form-group">

<font color="red">

<center>

<asp:Label ID="hata" runat="server" Text="" Font-Bold></asp:Label>

</center>

</font>

</div>

</div>

<div class="col-md-6">

<asp:Button ID="girisbtn" CssClass="btnRegister" runat="server" Text="GİRİŞ YAP" OnClick="Giris\_Click" />

</div>

</div>

</div>

<div class="tab-pane fade show" id="profile" role="tabpanel" aria-labelledby="profile-tab">

Kullanıcıların kayıt yaptığı bölüm.

<h3 class="register-heading">KAYIT EKRANI</h3>

<div class="row register-form">

<div class="col-md-6">

<div class="form-group">

<asp:TextBox ID="ad" CssClass="form-control" runat="server" placeholder="Ad \*"></asp:TextBox>

</div>

<div class="form-group">

<asp:TextBox ID="soyad" CssClass="form-control" runat="server" placeholder="Soyad \*"></asp:TextBox>

</div>

<div class="form-group">

<asp:TextBox TextMode="Email" ID="eposta" CssClass="form-control" runat="server" placeholder="E - Posta \*"></asp:TextBox>

</div>

<div class="form-group">

<asp:TextBox ID="kkadi" CssClass="form-control" runat="server" placeholder="Kullanıcı Adı \*"></asp:TextBox>

</div>

</div>

<div class="col-md-6">

<div class="form-group">

<asp:TextBox ID="telefon" CssClass="form-control" runat="server" placeholder="Telefon \*"></asp:TextBox>

</div>

<div class="form-group">

<asp:TextBox TextMode="Password" ID="ksifre" CssClass="form-control" runat="server" placeholder="Şifre \*"></asp:TextBox>

</div>

<div class="form-group">

<asp:TextBox TextMode="Password" ID="ksifre2" CssClass="form-control" runat="server" placeholder="Şifre Tekrar \*"></asp:TextBox>

<div class="form-group">

<font color="red">

<center>

<asp:Label ID="hata2" runat="server" Text="" Font-Bold></asp:Label>

</center>

</font>

</div>

</div>

<asp:Button ID="kayitbtn" CssClass="btnRegister" runat="server" Text="KAYIT OL" OnClick="Kayit\_Click" />

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</form>

</body>

</html>

**Kayıt Giriş (cs)**

public partial class giris : System.Web.UI.Page

{

KullaniciLogin kullaniciLogin = new KullaniciLogin();

protected void Page\_Load(object sender, EventArgs e)

{

}

Bu butona tıklandığında kullanici giriş yaparken öncelikle kullanıcının tüm alanları doldurması beklenir. Tüm alanlar dolu ise KullaniciLogin sınıfındaki “KullaniciGetir()” metoduna veriler gönderilir. Geriye dönen veriler DataTable’a atılır. Eğer kullanıcı var ise kullanıcının ID, isim ve yetkileri Sessionlara atanır. Eğer yetkili kullanıcı ise admin paneline, değil ise ana sayfaya yönlendirilir.

protected void Giris\_Click(object sender, EventArgs e)

{

DataTable dt = new DataTable();

dt = kullaniciLogin.KullaniciGetir(kadi.Text, sifre.Text);

if (kadi.Text == "" || sifre.Text == "")

{

hata.Text = "Lütfen Tüm Alanları Doldurun";

}

else

{

if (dt.Rows.Count > 0)

{

Session["ID"] = dt.Rows[0][0].ToString();

Session["Kullanici"] = (dt.Rows[0][1].ToString());

Session["Yetki"]=dt.Rows[0][3].ToString();

if ("0" == (dt.Rows[0][3].ToString()))

{

Response.Redirect("Yonetim/default.aspx");

}

else

{

Response.Redirect("default.aspx");

}

}

else

{

hata.Text = ("Kullanıcı Adı veya Şifre Hatalı");

}

}

}

Bu butona tıklandığında kullanici kaydı yapılırken öncelikle kullanıcının tüm alanları doldurması beklenir. Tüm alanlar dolu ise kullanıcı adı KullaniciLogin sınıfındaki “KullaniciGetir2()” metoduna gönderilir. Geriye dönen veriler DataTable’a atılır. Eğer bu kullanıcı adına sahip kişi var ise hata mesajı gösterilir. Yok ise tüm bilgiler KullaniciLogin sınıfındaki “KullaniciEkle()” metoduna gönderilir. Eğer geriye hata dönmezse kayıt yapıldığına dair mesaj gösterilir.

protected void Kayit\_Click(object sender, EventArgs e)

{

DataTable dt = new DataTable();

dt = kullaniciLogin.KullaniciGetir2(kkadi.Text);

if(ad.Text=="" || kkadi.Text == "" || ksifre.Text == "" || ksifre2.Text == "" || eposta.Text == "" || telefon.Text == "" )

{

hata2.Text = "Lütfen Tüm Alanları Doldurun";

}

else

{

if (dt.Rows.Count > 0)

{

hata2.Text = "Kayıtlı Kullanıcı";

}

else

{

if (ksifre.Text != ksifre2.Text)

{

hata2.Text = "Şifreler Aynı Değil";

}

else

{

bool hata = kullaniciLogin.KullaniciEkle(ad.Text, soyad.Text, kkadi.Text, ksifre.Text, eposta.Text, telefon.Text);

if (hata == false)

{

hata2.Text = "Kayıt Başarıyla Eklendi";

}

else

{

hata2.Text = "Kaydınız yapılamadı.";

}

}

}

}

}

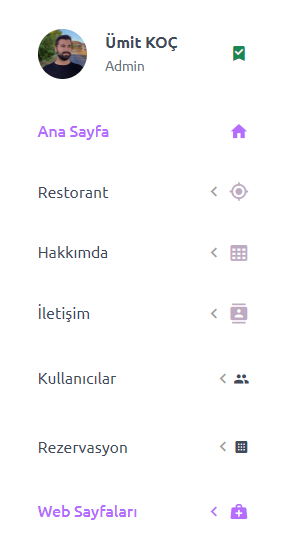
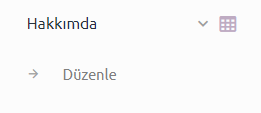
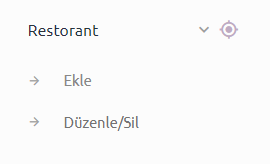
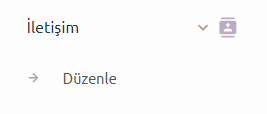
}

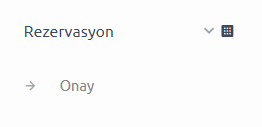
|  |  |
| --- | --- |
| public DataTable KullaniciGetir(string kullaniciAdi, string sifre)  {  DataTable dt = new DataTable();  try  {  using (SqlConnection baglanti = b.BaglantiOlustur())  {  using (SqlCommand komut = new SqlCommand("kullanici\_getir\_usp", baglanti))  {  komut.CommandType = CommandType.StoredProcedure;  komut.Parameters.AddWithValue("@kullaniciAdi", kullaniciAdi);  komut.Parameters.AddWithValue("@sifre", sifre);  baglanti.Open();  SqlDataAdapter da;  DataSet ds = new DataSet();  da = new SqlDataAdapter(komut);  da.Fill(ds, "Kullanicilar");  dt = ds.Tables["Kullanicilar"];  baglanti.Close();  }  }  }  catch (Exception hata)  {    }  return dt;  } | Bu “KullaniciGetir()” metodu sayesinde veri tabanından “kullanici\_getir \_usp” prosedürü çalıştırılıp içerisine gönderilen parametreler ile Kullanıcı tablosundaki kullanıcının verilerini geri gönderir. |
| CREATE proc [dbo].[kullanici\_getir\_usp]  @kullaniciAdi varchar(50),  @sifre varchar(16)  as  begin  select ID,isim,kullaniciAdi,yetki from Kullanicilar where kullaniciAdi=@kullaniciAdi and sifre=@sifre  end | “kullanici\_getir \_usp” prosedürü |

|  |  |
| --- | --- |
| public DataTable KullaniciGetir2(string kullaniciAdi)  {  DataTable dt = new DataTable();  try  {  using (SqlConnection baglanti = b.BaglantiOlustur())  {  using (SqlCommand komut = new SqlCommand("kullanici\_getir\_usp2", baglanti))  {  komut.CommandType = CommandType.StoredProcedure;  komut.Parameters.AddWithValue("@kullaniciAdi", kullaniciAdi);  baglanti.Open();  SqlDataAdapter da;  DataSet ds = new DataSet();  da = new SqlDataAdapter(komut);  da.Fill(ds, "Kullanicilar");  dt = ds.Tables["Kullanicilar"];  baglanti.Close();  }  }  }  catch (Exception hata)  {    }  return dt;  } | Bu “KullaniciGetir2()” metodu sayesinde veri tabanından “kullanici\_getir \_usp2” prosedürü çalıştırılıp içerisine gönderilen parametreler ile Kullanıcı tablosundaki kullanıcının verilerini geri gönderir. |
| CREATE proc [dbo].[restoran\_getir\_usp2]  @isim varchar(70)  as  begin  select ID from Kullanicilar where isim=@isim  end | “kullanici\_getir \_usp2” prosedürü |
| public bool KullaniciEkle(string isim, string soyisim, string kullaniciAdi, string sifre,  string eposta, string telefon)  {  bool hataOlustuMu = false;  try  {  using (SqlConnection baglanti = b.BaglantiOlustur())  {  using (SqlCommand komut = new SqlCommand("kullanici\_ekle\_usp", baglanti))  {  komut.CommandType = CommandType.StoredProcedure;  komut.Parameters.AddWithValue("@isim", isim);  komut.Parameters.AddWithValue("@soyisim", soyisim);  komut.Parameters.AddWithValue("@kullaniciAdi", kullaniciAdi);  komut.Parameters.AddWithValue("@sifre", sifre);  komut.Parameters.AddWithValue("@eposta", eposta);  komut.Parameters.AddWithValue("@telefon", telefon);  baglanti.Open();  komut.ExecuteNonQuery();  baglanti.Close();  }  }  }  catch (Exception hata)  {  hataOlustuMu = true;  }  return hataOlustuMu;  } | Bu “KullaniciEkle()” metodu sayesinde veri tabanından “kullanici\_ekle \_usp” prosedürü çalıştırılıp içerisine gönderilen parametreler ile Kullanıcı tablosuna yeni kayıt eklenir. Veri eklenemezse geriye hata gönderilir. |
| CREATE proc [dbo].[kullanici\_ekle\_usp]  @isim varchar(50),  @soyisim varchar(50),  @kullaniciAdi varchar(50),  @sifre varchar(70),  @eposta varchar(70),  @telefon varchar(15)  as  insert into Kullanicilar  (isim,  soyisim,  kullaniciAdi,  sifre,  eposta,  telefon,  yetki)  values  (@isim,@soyisim,@kullaniciAdi,@sifre,@eposta,@telefon,1) | “kullanici\_ekle \_usp” prosedürü |

**REZERVET- Admin- Master**



Bu bölümde admin panelinin master sayfası oluşturulmuştur. Master sayfasında admin bilgileri ve menü bölümü bulunmaktadır.

<!DOCTYPE html>

<html lang="tr">

<head>

<!-- Required meta tags -->

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<title>REZERVET Admin</title>

<!-- plugins:css -->

<link rel="stylesheet" href="assets/vendors/mdi/css/materialdesignicons.min.css">

<link rel="stylesheet" href="assets/vendors/css/vendor.bundle.base.css">

<!-- endinject -->

<!-- Plugin css for this page -->

<!-- End plugin css for this page -->

<!-- inject:css -->

<!-- endinject -->

<!-- Layout styles -->

<link rel="stylesheet" href="assets/css/style.css">

<!-- End layout styles -->

<link rel="shortcut icon" href="assets/images/favicon.svg" />

<asp:ContentPlaceHolder ID="head" runat="server">

</asp:ContentPlaceHolder>

</head>

<body>

Bu bölümde admin bilgileri bulunmaktadır.

<div class="container-scroller">

<!-- partial:partials/\_navbar.html -->

<nav class="navbar default-layout-navbar col-lg-12 col-12 p-0 fixed-top d-flex flex-row">

<div class="text-center navbar-brand-wrapper d-flex align-items-center justify-content-center">

<a class="navbar-brand brand-logo" href="default.aspx">

<img src="assets/images/logo.svg" alt="logo" /></a>

<a class="navbar-brand brand-logo-mini" href="default.aspx">

<img src="assets/images/logo-mini.svg" alt="logo" /></a>

</div>

<div class="navbar-menu-wrapper d-flex align-items-stretch">

<button class="navbar-toggler navbar-toggler align-self-center" type="button" data-toggle="minimize">

<span class="mdi mdi-menu"></span>

</button>

<ul class="navbar-nav navbar-nav-right">

<!-- çıkış1 -->

<li class="nav-item nav-profile dropdown">

<a class="nav-link dropdown-toggle" id="profileDropdown" href="#" data-bs-toggle="dropdown" aria-expanded="false">

<div class="nav-profile-img">

<img src="assets/images/faces/face.png" alt="image">

<span class="availability-status online"></span>

</div>

<div class="nav-profile-text">

<p class="mb-1 text-black">Ümit KOÇ</p>

</div>

</a>

<div class="dropdown-menu navbar-dropdown" aria-labelledby="profileDropdown">

<div class="dropdown-divider"></div>

<a class="dropdown-item" href="../default.aspx">

<i class="mdi mdi-logout me-2 text-primary"></i>Çıkış </a>

</div>

</li>

<!-- tamekran -->

<li class="nav-item d-none d-lg-block full-screen-link">

<a class="nav-link">

<i class="mdi mdi-fullscreen" id="fullscreen-button"></i>

</a>

</li>

<!-- çıkış2 -->

<li class="nav-item nav-logout d-none d-lg-block">

<a class="nav-link" href="../default.aspx">

<i class="mdi mdi-power"></i>

</a>

</li>

</ul>

<button class="navbar-toggler navbar-toggler-right d-lg-none align-self-center" type="button" data-toggle="offcanvas">

<span class="mdi mdi-menu"></span>

</button>

</div>

</nav>

Bu bölümde menüler bulunmaktadır.

<!-- partial -->

<div class="container-fluid page-body-wrapper">

<!-- partial:partials/\_sidebar.html -->

<nav class="sidebar sidebar-offcanvas" id="sidebar">

<ul class="nav">

<li class="nav-item nav-profile">

<a href="#" class="nav-link">

<div class="nav-profile-image">

<img src="assets/images/faces/face.png" alt="profile">

<span class="login-status online"></span>

<!--gerektiğinde çevrimdışı veya meşgul olarak değiştirin-->

</div>

<div class="nav-profile-text d-flex flex-column">

<span class="font-weight-bold mb-2">Ümit KOÇ</span>

<span class="text-secondary text-small">Admin</span>

</div>

<i class="mdi mdi-bookmark-check text-success nav-profile-badge"></i>

</a>

</li>

<li class="nav-item">

<a class="nav-link" href="default.aspx">

<span class="menu-title">Ana Sayfa</span>

<i class="mdi mdi-home menu-icon"></i>

</a>

</li>

<li class="nav-item">

<a class="nav-link" data-bs-toggle="collapse" href="#ui-basic" aria-expanded="false" aria-controls="ui-basic">

<span class="menu-title">Restorant</span>

<i class="menu-arrow"></i>

<i class="mdi mdi-crosshairs-gps menu-icon"></i>

</a>

<div class="collapse" id="ui-basic">

<ul class="nav flex-column sub-menu">

<li class="nav-item"><a class="nav-link" href="RestorantEkle.aspx">Ekle</a></li>

<li class="nav-item"><a class="nav-link" href="RestorantDuzenleSil.aspx">Düzenle/Sil</a></li>

</ul>

</div>

</li>

<li class="nav-item">

<a class="nav-link" data-bs-toggle="collapse" href="#ui-basic1" aria-expanded="false" aria-controls="ui-basic1">

<span class="menu-title">Hakkımda</span>

<i class="menu-arrow"></i>

<i class="mdi mdi-table-large menu-icon"></i>

</a>

<div class="collapse" id="ui-basic1">

<ul class="nav flex-column sub-menu">

<li class="nav-item"><a class="nav-link" href="HakkimdaDuzenle.aspx">Düzenle</a></li>

</ul>

</div>

</li>

<li class="nav-item">

<a class="nav-link" data-bs-toggle="collapse" href="#ui-basic2" aria-expanded="false" aria-controls="ui-basic2">

<span class="menu-title">İletişim</span>

<i class="menu-arrow"></i>

<i class="mdi mdi-contacts menu-icon"></i>

</a>

<div class="collapse" id="ui-basic2">

<ul class="nav flex-column sub-menu">

<li class="nav-item"><a class="nav-link" href="IletisimDuzenle.aspx">Düzenle</a></li>

</ul>

</div>

</li>

<li class="nav-item">

<a class="nav-link" data-bs-toggle="collapse" href="#ui-basic3" aria-expanded="false" aria-controls="ui-basic3">

<span class="menu-title">Kullanıcılar</span>

<i class="menu-arrow"></i>

<i class="mdi mdi-account-multiple"></i>

</a>

<div class="collapse" id="ui-basic3">

<ul class="nav flex-column sub-menu">

<li class="nav-item"><a class="nav-link" href="Kullanici.aspx">Kullanıcı Düzenle</a></li>

<li class="nav-item"><a class="nav-link" href="Mesajlar.aspx">Kullanıcı Mesajları</a></li>

</ul>

</div>

</li>

<li class="nav-item">

<a class="nav-link" data-bs-toggle="collapse" href="#ui-basic4" aria-expanded="false" aria-controls="ui-basic4">

<span class="menu-title">Rezervasyon</span>

<i class="menu-arrow"></i>

<i class="mdi mdi-apps-box"></i>

</a>

<div class="collapse" id="ui-basic4">

<ul class="nav flex-column sub-menu">

<li class="nav-item"><a class="nav-link" href="Rezervasyon.aspx">Onay</a></li>

</ul>

</div>

</li>

<li class="nav-item">

<a class="nav-link" data-bs-toggle="collapse" href="#general-pages1" aria-expanded="false" aria-controls="general-pages1">

<span class="menu-title">Web Sayfaları</span>

<i class="menu-arrow"></i>

<i class="mdi mdi-medical-bag menu-icon"></i>

</a>

<div class="collapse" id="general-pages1">

<ul class="nav flex-column sub-menu">

<li class="nav-item"><a class="nav-link" target="\_blank" href="../default.aspx">AnaSayfa </a></li>

<li class="nav-item"><a class="nav-link" target="\_blank" href="../Hakkimizda.aspx">Hakkımızda </a></li>

<li class="nav-item"><a class="nav-link" target="\_blank" href="../Restaurantlar.aspx">Restorantlar </a></li>

<li class="nav-item"><a class="nav-link" target="\_blank" href="../Iletisim.aspx">İletişim </a></li>

<li class="nav-item"><a class="nav-link" target="\_blank" href="../kayit-giris.aspx">Kayıt-Giriş </a></li>

</ul>

</div>

</li>

</li>

</ul>

</nav>

<!-- partial -->

<div class="main-panel">

<div class="content-wrapper">

<div class="page-header">

<asp:ContentPlaceHolder ID="ContentPlaceHolder1" runat="server">

<h3 class="page-title">Form elements </h3>

</asp:ContentPlaceHolder>

</div>

<div class="row">

<asp:ContentPlaceHolder ID="ContentPlaceHolder2" runat="server">

<div class="col-12 grid-margin stretch-card">

<div class="card">

<div class="card-body">

<h4 class="card-title">Basic form elements</h4>

<p class="card-description">Basic form elements </p>

<form class="forms-sample">

<div class="form-group">

<label for="exampleInputName1">Name</label>

<input type="text" class="form-control" id="exampleInputName1" placeholder="Name">

</div>

<div class="form-group">

<label for="exampleInputEmail3">Email address</label>

<input type="email" class="form-control" id="exampleInputEmail3" placeholder="Email">

</div>

<div class="form-group">

<label for="exampleInputPassword4">Password</label>

<input type="password" class="form-control" id="exampleInputPassword4" placeholder="Password">

</div>

<div class="form-group">

<label for="exampleSelectGender">Gender</label>

<select class="form-control" id="exampleSelectGender">

<option>Male</option>

<option>Female</option>

</select>

</div>

<div class="form-group">

<label>File upload</label>

<input type="file" name="img[]" class="file-upload-default">

<div class="input-group col-xs-12">

<input type="text" class="form-control file-upload-info" disabled placeholder="Upload Image">

<span class="input-group-append">

<button class="file-upload-browse btn btn-gradient-primary" type="button">Upload</button>

</span>

</div>

</div>

<div class="form-group">

<label for="exampleInputCity1">City</label>

<input type="text" class="form-control" id="exampleInputCity1" placeholder="Location">

</div>

<div class="form-group">

<label for="exampleTextarea1">Textarea</label>

<textarea class="form-control" id="exampleTextarea1" rows="4"></textarea>

</div>

<button type="submit" class="btn btn-gradient-primary me-2">Submit</button>

<button class="btn btn-light">Cancel</button>

</form>

</div>

</div>

</div>

</asp:ContentPlaceHolder>

</div>

</div>

<!-- content-wrapper ends -->

<!-- partial:partials/\_footer.html -->

<footer class="footer">

<div class="container-fluid d-flex justify-content-between">

<span class="text-muted d-block text-center text-sm-start d-sm-inline-block">Copyright © 2022 umitkoc. Tüm Hakları Saklıdır.</span>

</div>

</footer>

<!-- partial -->

</div>

<!-- main-panel ends -->

</div>

<!-- page-body-wrapper ends -->

</div>

<!-- container-scroller -->

<!-- plugins:js -->

<script src="assets/vendors/js/vendor.bundle.base.js"></script>

<!-- endinject -->

<!-- Plugin js for this page -->

<!-- End plugin js for this page -->

<!-- inject:js -->

<script src="assets/js/off-canvas.js"></script>

<script src="assets/js/hoverable-collapse.js"></script>

<script src="assets/js/misc.js"></script>

<!-- endinject -->

<!-- Custom js for this page -->

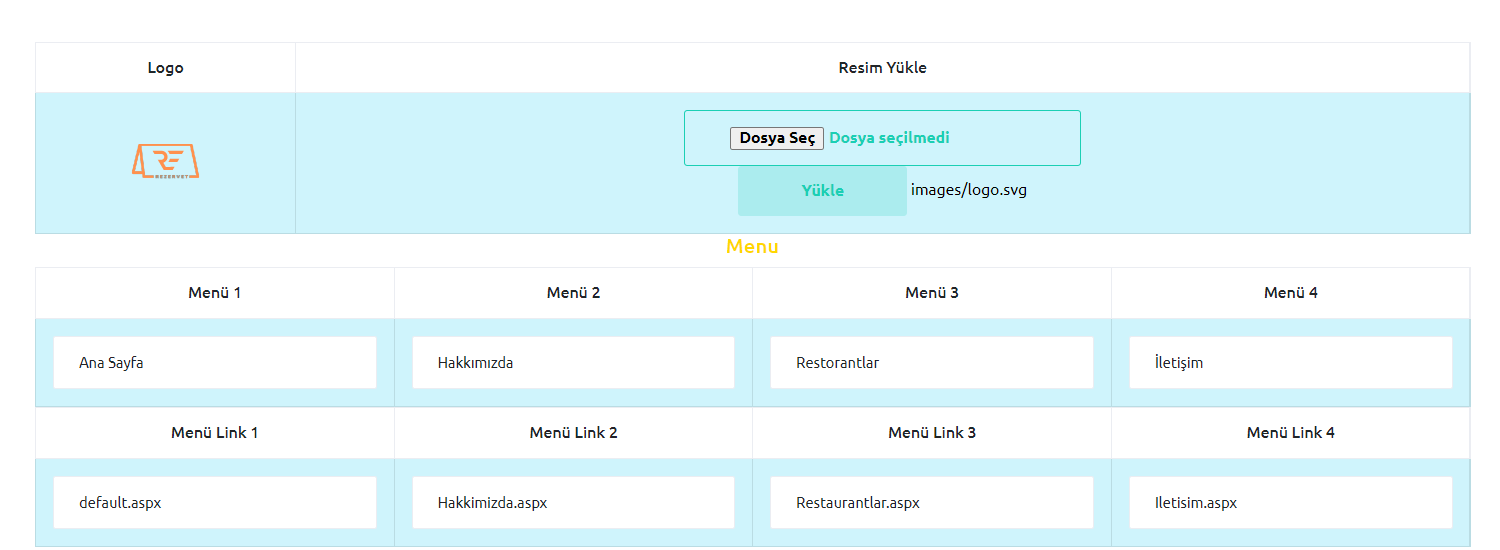
<script src="assets/js/file-upload.js"></script>

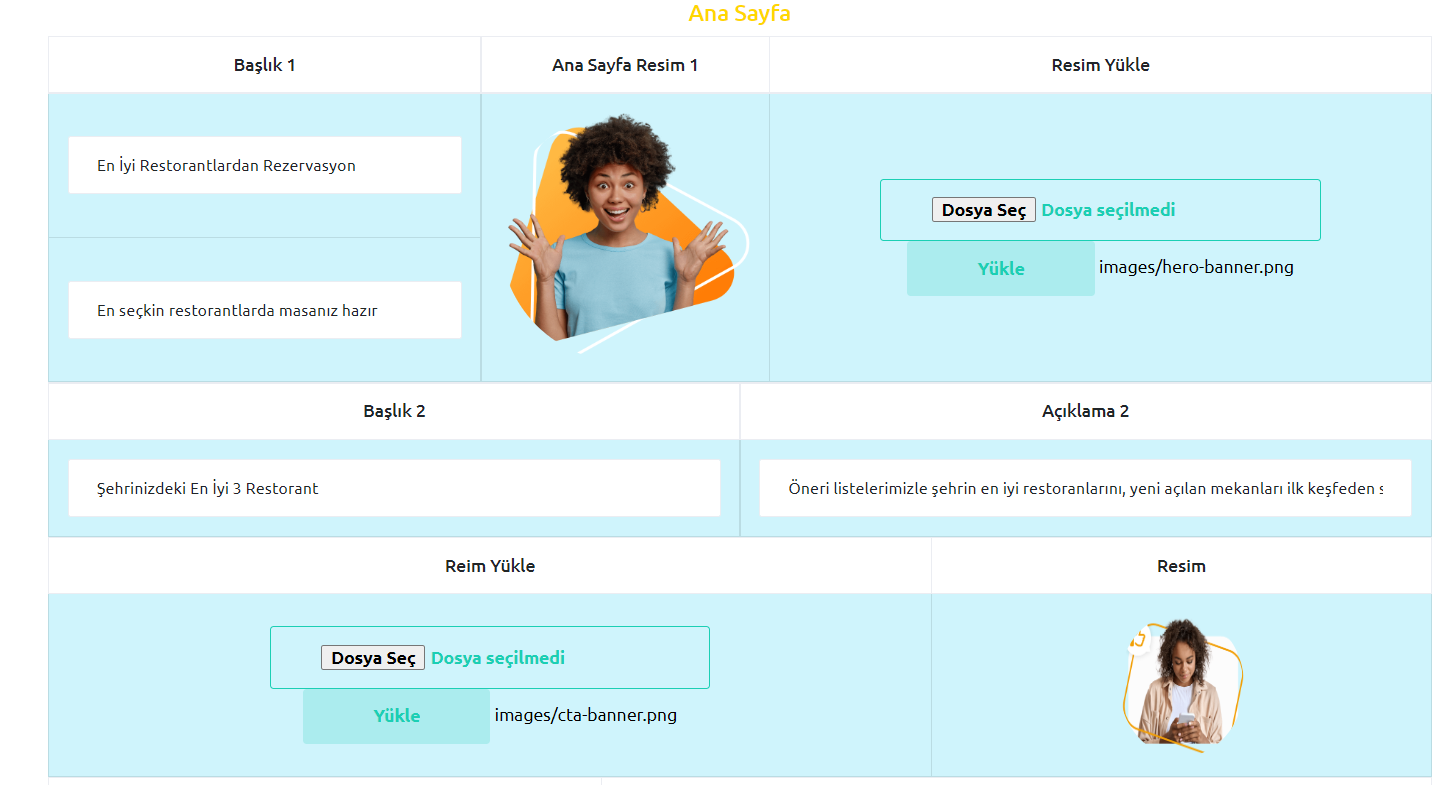
<!-- End custom js for this page -->

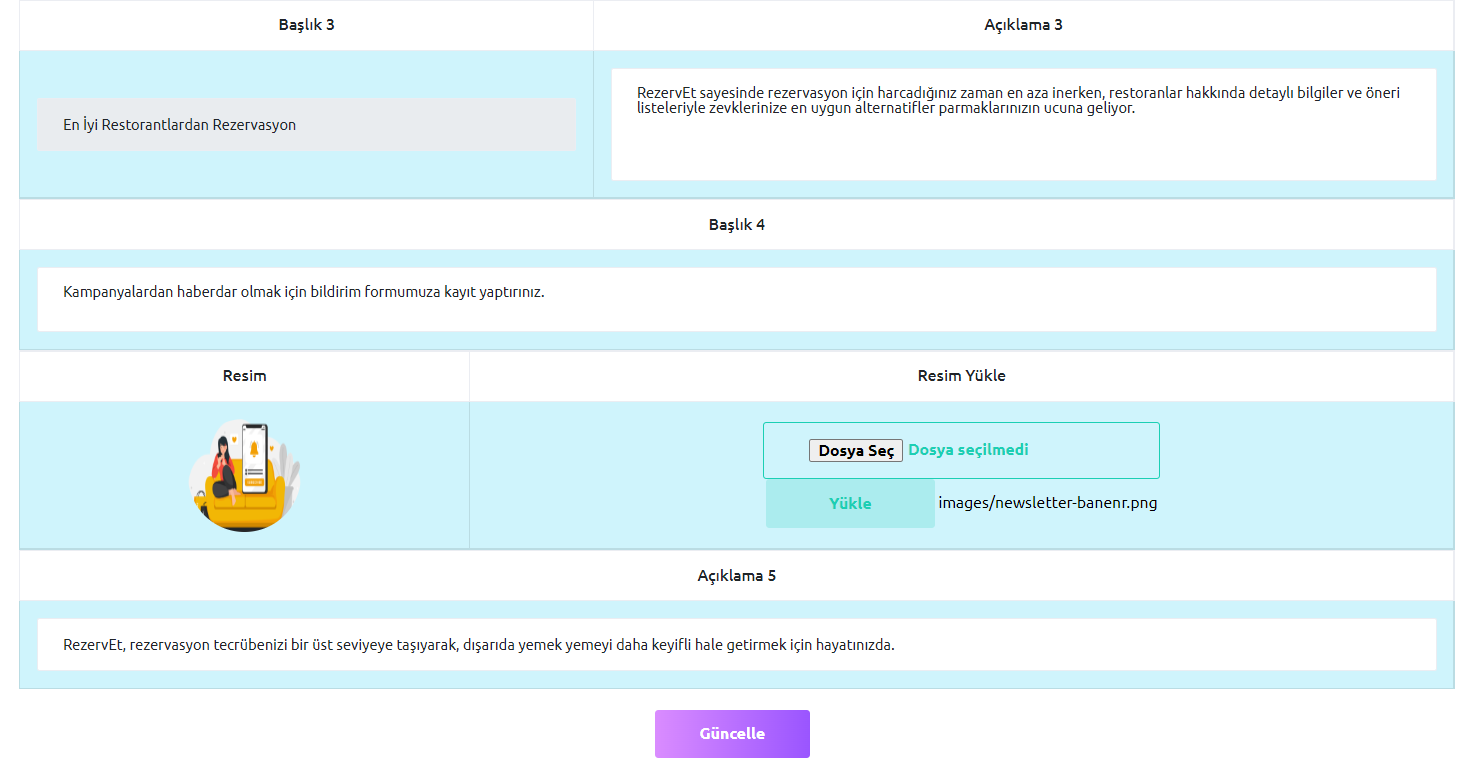
</body>

</html>

**REZERVET- Admin- Ana Sayfa**







Bu bölümde menü ve Ana Sayfadaki içerikler güncellenmektedir.

<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">

<h3 class="page-title">İşlem Yapmak İçin Lütfen Menüyü Kullanınız... </h3>

</asp:Content>

<asp:Content ID="Content3" ContentPlaceHolderID="ContentPlaceHolder2" runat="server">

<div class="col-12 grid-margin stretch-card">

<div class="card">

<div class="card-body">

<form class="forms-sample" runat="server">

<table class="table table-bordered">

<thead>

<tr align="center">

<th width="200">Logo</th>

<th>Resim Yükle </th>

</tr>

</thead>

<tbody>

<tr class="table-info" align="center">

<td width="200">

<asp:Image ID="Image1" Width="200" runat="server" />

</td>

<td>

<asp:FileUpload ID="FileUpload1" runat="server" CssClass="file-upload-browse btn btn-outline-success btn-fw" /><br />

<asp:Button ID="fotobtn1" OnClick="Resim\_Yukle" runat="server" Text="Yükle" CssClass="btn btn-inverse-success btn-fw" />

<asp:Label ID="fotolbl1" runat="server" Text=""></asp:Label>

</td>

</tr>

</tbody>

</table>

<div class="form-group" display="inline">

<h4 align="center" style="color: gold">Menu</h4>

<table class="table table-bordered">

<thead>

<tr align="center">

<th>Menü 1</th>

<th>Menü 2 </th>

<th>Menü 3</th>

<th>Menü 4 </th>

</tr>

</thead>

<tbody>

<tr class="table-info" align="center">

<td>

<asp:TextBox ID="menu1" runat="server" CssClass="form-control"></asp:TextBox>

</td>

<td>

<asp:TextBox ID="menu2" runat="server" CssClass="form-control"></asp:TextBox>

</td>

<td>

<asp:TextBox ID="menu3" runat="server" CssClass="form-control"></asp:TextBox>

</td>

<td>

<asp:TextBox ID="menu4" runat="server" CssClass="form-control"></asp:TextBox>

</td>

</tr>

</tbody>

</table>

<table class="table table-bordered">

<thead>

<tr align="center">

<th>Menü Link 1</th>

<th>Menü Link 2 </th>

<th>Menü Link 3</th>

<th>Menü Link 4 </th>

</tr>

</thead>

<tbody>

<tr class="table-info" align="center">

<td>

<asp:TextBox ID="link1" runat="server" CssClass="form-control"></asp:TextBox>

</td>

<td>

<asp:TextBox ID="link2" runat="server" CssClass="form-control"></asp:TextBox>

</td>

<td>

<asp:TextBox ID="link3" runat="server" CssClass="form-control"></asp:TextBox>

</td>

<td>

<asp:TextBox ID="link4" runat="server" CssClass="form-control"></asp:TextBox>

</td>

</tr>

</tbody>

</table>

<h4 align="center" style="color: gold">Ana Sayfa</h4>

<table class="table table-bordered">

<thead>

<tr align="center">

<th>Başlık 1</th>

<th>Ana Sayfa Resim 1</th>

<th>Resim Yükle </th>

</tr>

</thead>

<tbody>

<tr class="table-info" align="center">

<td>

<asp:TextBox ID="baslik1" runat="server" CssClass="form-control"></asp:TextBox>

</td>

<td rowspan="2" width="200">

<asp:Image ID="Image2" Width="200" Height="200" runat="server" />

</td>

<td rowspan="2">

<asp:FileUpload ID="FileUpload2" runat="server" CssClass="file-upload-browse btn btn-outline-success btn-fw" /><br />

<asp:Button ID="fotobtn2" OnClick="Resim\_Yukle2" runat="server" Text="Yükle" CssClass="btn btn-inverse-success btn-fw" />

<asp:Label ID="fotolbl2" runat="server" Text=""></asp:Label>

</td>

</tr>

<tr class="table-info" align="center">

<td>

<asp:TextBox ID="metin1" runat="server" CssClass="form-control"></asp:TextBox>

</td>

</tr>

</tbody>

</table>

</table>

<table class="table table-bordered">

<thead>

<tr align="center">

<th>Başlık 2</th>

<th>Açıklama 2 </th>

</tr>

</thead>

<tbody>

<tr class="table-info" align="center">

<td>

<asp:TextBox ID="baslik2" runat="server" CssClass="form-control"></asp:TextBox>

</td>

<td>

<asp:TextBox ID="metin2" runat="server" CssClass="form-control"></asp:TextBox>

</td>

</tr>

</tbody>

</table>

<table class="table table-bordered">

<thead>

<tr align="center">

<th>Reim Yükle </th>

<th>Resim </th>

</tr>

</thead>

<tbody>

<tr class="table-info" align="center">

<td>

<asp:FileUpload ID="FileUpload3" runat="server" CssClass="file-upload-browse btn btn-outline-success btn-fw" /><br />

<asp:Button ID="fotobtn3" OnClick="Resim\_Yukle3" runat="server" Text="Yükle" CssClass="btn btn-inverse-success btn-fw" />

<asp:Label ID="fotolbl3" runat="server" Text=""></asp:Label>

</td>

<td rowspan="2" width="400">

<asp:Image ID="Image3" Width="100" Height="115" runat="server" />

</td>

</tr>

</tbody>

</table>

<table class="table table-bordered">

<thead>

<tr align="center">

<th>Başlık 3 </th>

<th>Açıklama 3 </th>

</tr>

</thead>

<tbody>

<tr class="table-info" align="center">

<td width="40%">

<asp:TextBox ID="baslik3" runat="server" ReadOnly="true" CssClass="form-control"></asp:TextBox>

</td>

<td>

<asp:TextBox ID="metin3" TextMode="MultiLine" Height="100" runat="server" CssClass="form-control"></asp:TextBox>

</td>

</tr>

</tbody>

</table>

<table class="table table-bordered">

<thead>

<tr align="center">

<th>Başlık 4 </th>

</tr>

</thead>

<tbody>

<tr class="table-info" align="center">

<td>

<asp:TextBox ID="baslik4" TextMode="MultiLine" runat="server" CssClass="form-control"></asp:TextBox>

</td>

</tr>

</tbody>

</table>

<table class="table table-bordered">

<thead>

<tr align="center">

<th>Resim </th>

<th>Resim Yükle </th>

</tr>

</thead>

<tbody>

<tr class="table-info" align="center">

<td width="400">

<asp:Image ID="Image4" Width="100" Height="100" runat="server" />

</td>

<td>

<asp:FileUpload ID="FileUpload4" runat="server" CssClass="file-upload-browse btn btn-outline-success btn-fw" /><br />

<asp:Button ID="fotobtn4" OnClick="Resim\_Yukle4" runat="server" Text="Yükle" CssClass="btn btn-inverse-success btn-fw" />

<asp:Label ID="fotolbl4" runat="server" Text=""></asp:Label>

</td>

</tr>

</tbody>

</table>

<table class="table table-bordered">

<thead>

<tr align="center">

<th>Açıklama 5 </th>

</tr>

</thead>

<tbody>

<tr class="table-info" align="center">

<td width="40%">

<asp:TextBox ID="metin4" runat="server" CssClass="form-control"></asp:TextBox>

</td>

</tr>

</tbody>

</table>

<table><br />

<center> <asp:Button ID="guncelle" runat="server" Text="Güncelle" OnClick="Guncelle" CssClass="btn btn-gradient-primary me-2" />

<asp:Label ID="lblHata" runat="server" Text="" Font-Bold></asp:Label></center>

</table>

</form>

</div>

</div>

</div>

</asp:Content>

**Admin- Ana Sayfa (cs)**

public partial class \_default : System.Web.UI.Page

{

Bilgi bilgi = new Bilgi();

AnaSayfaGun anasayfa = new AnaSayfaGun();

SiteGenelGun sitegenel = new SiteGenelGun();

Sayfa yüklendiğinde kullanıcı yetkisi yok ise ana sayfaya yönlendirme yapılmaktadır. Yetkisi var ise “VeriGetir()” metoduyla ana sayfadaki veriler getirilmektedir.

protected void Page\_Load(object sender, EventArgs e)

{

if (Session["Yetki"] == null)

{

Response.Redirect("../default.aspx");

}

if (!IsPostBack)

{

VeriGetir();

}

}

Bu butona tıklandığında SiteGenelGun sınıfındaki “SiteGenelGuncelle()” metoduna ve AnaSayfaGun sınıfındaki “AnaSayfaGuncelle()” metoduna veriler gönderilir. Güncelleme işlemi yapılır. Daha sonra “VeriGetir()” metodu çalıştırılarak veriler getirilir.

protected void Guncelle(object sender, EventArgs e)

{

sitegenel.SiteGenelGuncelle(fotolbl1.Text, baslik1.Text, metin4.Text, menu1.Text, menu2.Text, menu3.Text, menu4.Text, link1.Text, link2.Text, link3.Text, link4.Text);

anasayfa.AnaSayfaGuncelle(baslik1.Text, baslik2.Text, baslik4.Text, metin1.Text, metin2.Text, metin3.Text, fotolbl2.Text, fotolbl3.Text, fotolbl4.Text);

VeriGetir();

}

Bu metot sayesinde DataTable’la bilgi sınıfındaki “SiteGenel()” metodundan veriler DataTable’a atılır. Bu gelen veriler menülerin düzenlenebildiği bölümdeki textlerde gösterilir. Daha sonra bilgi sınıfındaki “AnaSayfa()” metodundan veriler DataTable’a atılır. Bu gelen veriler ana sayfanın düzenlenebildiği bölümdeki textlerde gösterilir.

public void VeriGetir()

{

DataTable dt = new DataTable();

dt = bilgi.SiteGenel();

Image1.ImageUrl = "../" + dt.Rows[0][1].ToString();

fotolbl1.Text = dt.Rows[0][1].ToString();

menu1.Text = dt.Rows[0][12].ToString();

menu2.Text = dt.Rows[0][13].ToString();

menu3.Text = dt.Rows[0][14].ToString();

menu4.Text = dt.Rows[0][15].ToString();

link1.Text = dt.Rows[0][16].ToString();

link2.Text = dt.Rows[0][17].ToString();

link3.Text = dt.Rows[0][18].ToString();

link4.Text = dt.Rows[0][19].ToString();

metin4.Text = dt.Rows[0][3].ToString();

dt = bilgi.AnaSayfa();

baslik1.Text = dt.Rows[0][1].ToString();

baslik2.Text = dt.Rows[0][2].ToString();

baslik3.Text = dt.Rows[0][1].ToString();

baslik4.Text = dt.Rows[0][3].ToString();

metin1.Text = dt.Rows[0][4].ToString();

metin2.Text = dt.Rows[0][5].ToString();

metin3.Text = dt.Rows[0][6].ToString();

Image2.ImageUrl = "../" + dt.Rows[0][7].ToString();

fotolbl2.Text = dt.Rows[0][7].ToString();

Image3.ImageUrl = "../" + dt.Rows[0][8].ToString();

fotolbl3.Text = dt.Rows[0][8].ToString();

Image4.ImageUrl = "../" + dt.Rows[0][9].ToString();

fotolbl4.Text = dt.Rows[0][9].ToString();

}

Aşağıdaki “ResimYukle()” metotları sayesinde FileUpload kullanılarak resimler belirlenen yollara kaydedilir. Eğer belirlenen uzantıya sahip resimler eklenmezse hata mesajı verir.

protected void Resim\_Yukle(object sender, EventArgs e)

{

if (FileUpload1.HasFile)

{

if (FileUpload1.PostedFile.ContentType == "image/jpeg" || FileUpload1.PostedFile.ContentType == "image/jpg" || FileUpload1.PostedFile.ContentType == "image/png")

{

string resimAd = FileUpload1.FileName.ToString();

FileUpload1.SaveAs(Server.MapPath("~/images/" + resimAd));

fotolbl1.Text = "images/" + resimAd.ToString();

}

else

{

fotolbl1.Text = "Lütfen 'jpeg' , 'jpg' veya 'png' uzantılı resim seçin.";

}

}

else

{

fotolbl1.Text = "Lütfen bir resim seçin...";

}

}

protected void Resim\_Yukle2(object sender, EventArgs e)

{

if (FileUpload2.HasFile)

{

if (FileUpload2.PostedFile.ContentType == "image/jpeg" || FileUpload2.PostedFile.ContentType == "image/jpg" || FileUpload2.PostedFile.ContentType == "image/png")

{

string resimAd = FileUpload2.FileName.ToString();

FileUpload2.SaveAs(Server.MapPath("~/images/" + resimAd));

fotolbl2.Text = "images/" + resimAd.ToString();

}

else

{

fotolbl2.Text = "Lütfen 'jpeg' , 'jpg' veya 'png' uzantılı resim seçin.";

}

}

else

{

fotolbl2.Text = "Lütfen bir resim seçin...";

}

}

protected void Resim\_Yukle3(object sender, EventArgs e)

{

if (FileUpload3.HasFile)

{

if (FileUpload3.PostedFile.ContentType == "image/jpeg" || FileUpload3.PostedFile.ContentType == "image/jpg" || FileUpload3.PostedFile.ContentType == "image/png")

{

string resimAd = FileUpload3.FileName.ToString();

FileUpload1.SaveAs(Server.MapPath("~/images/" + resimAd));

fotolbl3.Text = "images/" + resimAd.ToString();

}

else

{

fotolbl3.Text = "Lütfen 'jpeg' , 'jpg' veya 'png' uzantılı resim seçin.";

}

}

else

{

fotolbl3.Text = "Lütfen bir resim seçin...";

}

}

protected void Resim\_Yukle4(object sender, EventArgs e)

{

if (FileUpload4.HasFile)

{

if (FileUpload4.PostedFile.ContentType == "image/jpeg" || FileUpload4.PostedFile.ContentType == "image/jpg" || FileUpload4.PostedFile.ContentType == "image/png")

{

string resimAd = FileUpload4.FileName.ToString();

FileUpload2.SaveAs(Server.MapPath("~/images/" + resimAd));

fotolbl4.Text = "images/" + resimAd.ToString();

}

else

{

fotolbl4.Text = "Lütfen 'jpeg' , 'jpg' veya 'png' uzantılı resim seçin.";

}

}

else

{

fotolbl4.Text = "Lütfen bir resim seçin...";

}

}

}

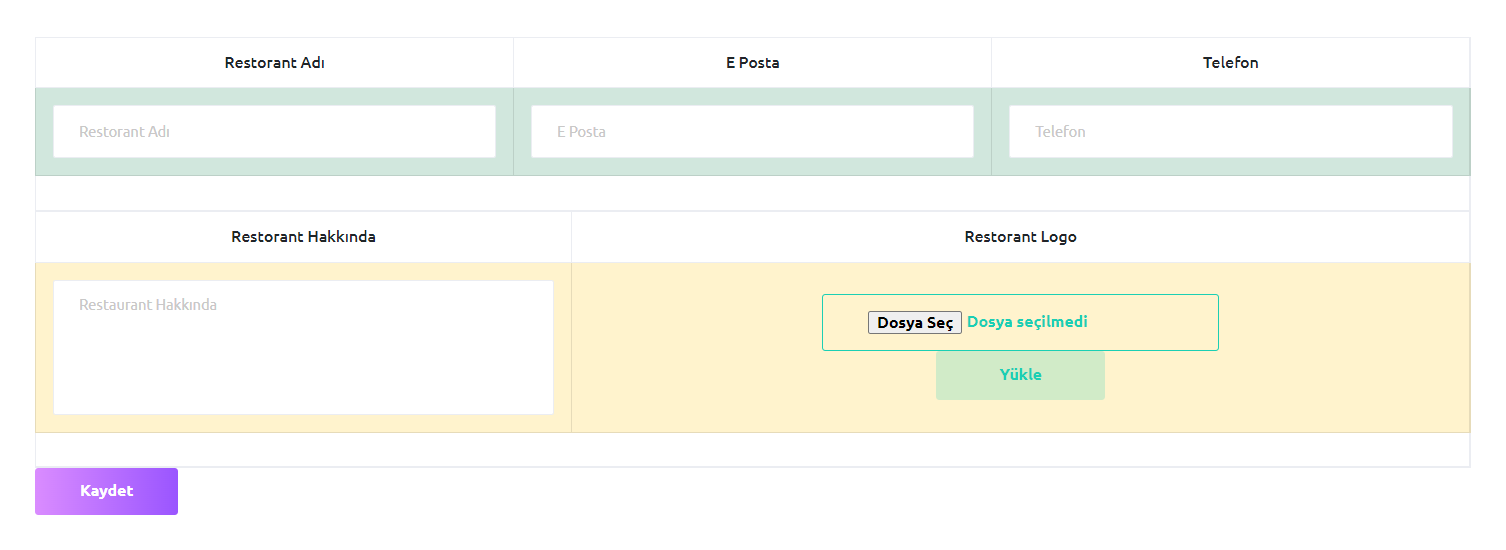
|  |  |
| --- | --- |
| public bool SiteGenelGuncelle(string logo, string baslik, string kisaAciklama, string menu1, string menu2, string menu3, string menu4, string menuLink1, string menuLink2, string menuLink3, string menuLink4)  {  bool hataOlustuMu = false;  try  {  using (SqlConnection baglanti = b.BaglantiOlustur())  {  using (SqlCommand komut = new SqlCommand("site\_genel\_guncelle\_usp", baglanti))  {  komut.CommandType = CommandType.StoredProcedure;  komut.Parameters.AddWithValue("@logo", logo);  komut.Parameters.AddWithValue("@baslik", baslik);  komut.Parameters.AddWithValue("@kisaAciklama", kisaAciklama);  komut.Parameters.AddWithValue("@menu1", menu1);  komut.Parameters.AddWithValue("@menu2", menu2);  komut.Parameters.AddWithValue("@menu3", menu3);  komut.Parameters.AddWithValue("@menu4", menu4);  komut.Parameters.AddWithValue("@menuLink1", menuLink1);  komut.Parameters.AddWithValue("@menuLink2", menuLink2);  komut.Parameters.AddWithValue("@menuLink3", menuLink3);  komut.Parameters.AddWithValue("@menuLink4", menuLink4);  baglanti.Open();  komut.ExecuteNonQuery();  baglanti.Close();  }  }  }  catch (Exception hata)  {  hataOlustuMu = true;  }  return hataOlustuMu;  } | Bu “SiteGenelGuncelle()” metodu sayesinde veri tabanından “site\_genel\_guncelle \_usp” prosedürü çalıştırılıp içerisine gönderilen parametreler ile Sitegenel tablosundaki veriler güncellenir. Veri güncellenmezse geriye hata gönderilir. |
| CREATE proc [dbo].[site\_genel\_guncelle\_usp]  @logo nvarchar(50),  @baslik nvarchar(300),  @kisaAciklama nvarchar(500),  @menu1 nvarchar(50),  @menu2 nvarchar(50),  @menu3 nvarchar(50),  @menu4 nvarchar(50),  @menuLink1 nvarchar(50),  @menuLink2 nvarchar(50),  @menuLink3 nvarchar(50),  @menuLink4 nvarchar(50)  as  update SiteGenel set  logo =@logo ,  baslik= @baslik ,  kisaAciklama= @kisaAciklama ,  menu1= @menu1 ,  menu2= @menu2 ,  menu3= @menu3 ,  menu4= @menu4,  menuLink1= @menuLink1,  menuLink2= @menuLink2 ,  menuLink3= @menuLink3,  menuLink4= @menuLink4 | “site\_genel\_guncelle \_usp” prosedürü |

|  |  |
| --- | --- |
| public bool AnaSayfaGuncelle(string baslik1, string baslik2, string baslik3, string aciklama1, string aciklama2, string aciklama3, string resim1, string resim2,string resim3)  {  bool hataOlustuMu = false;    try  {  using (SqlConnection baglanti = b.BaglantiOlustur())  {  using (SqlCommand komut = new SqlCommand("anasayfa\_guncelle\_usp", baglanti))  {  komut.CommandType = CommandType.StoredProcedure;  komut.Parameters.AddWithValue("@baslik1", baslik1);  komut.Parameters.AddWithValue("@baslik2", baslik2);  komut.Parameters.AddWithValue("@baslik3", baslik3);  komut.Parameters.AddWithValue("@aciklama1", aciklama1);  komut.Parameters.AddWithValue("@aciklama2", aciklama2);  komut.Parameters.AddWithValue("@aciklama3", aciklama3);  komut.Parameters.AddWithValue("@resim1", resim1);  komut.Parameters.AddWithValue("@resim2", resim2);  komut.Parameters.AddWithValue("@resim3", resim3);  baglanti.Open();  komut.ExecuteNonQuery();  baglanti.Close();  }  }  }  catch (Exception hata)  {  hataOlustuMu = true;  }  return hataOlustuMu;  }  } | Bu “AnaSayfaGuncelle()” metodu sayesinde veri tabanından “anasayfa\_guncelle \_usp” prosedürü çalıştırılıp içerisine gönderilen parametreler ile AnaSayfa tablosundaki veriler güncellenir. Veri güncellenmezse geriye hata gönderilir. |
| CREATE proc [dbo].[anasayfa\_guncelle\_usp]  @baslik1 nvarchar(300),  @baslik2 nvarchar(300),  @baslik3 nvarchar(300),  @aciklama1 nvarchar(500),  @aciklama2 nvarchar(500),  @aciklama3 nvarchar(500),  @resim1 nvarchar(50),  @resim2 nvarchar(50),  @resim3 nvarchar(50)  as  update AnaSayfa set  baslik1 =@baslik1 ,  baslik2= @baslik2 ,  baslik3= @baslik3 ,  aciklama1= @aciklama1 ,  aciklama2= @aciklama2 ,  aciklama3= @aciklama3 ,  resim1= @resim1,  resim2= @resim2,  resim3= @resim3 | “anasayfa\_guncelle \_usp” prosedürü |

|  |  |
| --- | --- |
| public DataTable SiteGenel()  {  DataTable dt = new DataTable();  try  {  using (SqlConnection baglanti = b.BaglantiOlustur())  {  using (SqlCommand komut = new SqlCommand("site\_genel\_usp", baglanti))  {  komut.CommandType = CommandType.StoredProcedure;  baglanti.Open();  SqlDataAdapter da;  DataSet ds = new DataSet();  da = new SqlDataAdapter(komut);  da.Fill(ds, "Site");  dt = ds.Tables["Site"];  baglanti.Close();  }  }  }  catch (Exception hata)  {  }  return dt;  } | Bu “SiteGenel()” metodu sayesinde veri tabanından “site\_genel\_usp” prosedürü çalıştırılıp bilgiler DataTable ile geri gönderilir. |
| CREATE proc [dbo].[site\_genel\_usp]  as  begin  select \* from SiteGenel  end | “site\_genel\_usp” prosedürü |

|  |  |
| --- | --- |
| public DataTable AnaSayfa()  {  DataTable dt = new DataTable();  try  {  using (SqlConnection baglanti = b.BaglantiOlustur())  {  using (SqlCommand komut = new SqlCommand("anasayfa\_usp", baglanti))  {  komut.CommandType = CommandType.StoredProcedure;  baglanti.Open();  SqlDataAdapter da;  DataSet ds = new DataSet();  da = new SqlDataAdapter(komut);  da.Fill(ds, "AnaSayfa");  dt = ds.Tables["AnaSayfa"];  baglanti.Close();  }  }  }  catch (Exception hata)  {  }  return dt;  } | Bu “AnaSayfa()” metodu sayesinde veri tabanından “anasayfa\_usp” prosedürü çalıştırılıp bilgiler DataTable ile geri gönderilir. |
| CREATE proc [dbo].[anasayfa\_usp]  as  begin  select \* from AnaSayfa  end | “anasayfa\_usp” prosedürü |

**REZERVET- Admin- Restoran Ekle**



Bu bölümde yeni Restoran ekleme işlemi yapılmaktadır.

<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">

<h3 class="page-title">Yeni Restorant Ekleme Sayfası </h3>

</asp:Content>

<asp:Content ID="Content3" ContentPlaceHolderID="ContentPlaceHolder2" runat="server">

<form class="forms-sample" runat="server">

<div class="col-12 grid-margin stretch-card">

<div class="card">

<div class="card-body">

<table class="table table-bordered">

<thead>

<tr align="center">

<th>Restorant Adı </th>

<th>E Posta</th>

<th>Telefon </th>

</tr>

</thead>

<tbody>

<tr class="table-success" align="center">

<td>

<asp:TextBox ID="txtAd" runat="server" CssClass="form-control" placeholder="Restorant Adı"></asp:TextBox>

</td>

<td>

<asp:TextBox ID="txtMail" runat="server" TextMode="Email" CssClass="form-control" placeholder="E Posta"></asp:TextBox>

</td>

<td>

<asp:TextBox ID="txtTel" runat="server" TextMode="Phone" CssClass="form-control" placeholder="Telefon"></asp:TextBox>

</td>

</tr>

<tr>

<td colspan="4">

<center>

<asp:Label ID="hatalbl" CssClass="text-danger" runat="server"></asp:Label>

</center>

</td>

</tr>

</tbody>

</table>

<table class="table table-bordered">

<thead>

<tr align="center">

<th>Restorant Hakkında </th>

<th>Restorant Logo</th>

</tr>

</thead>

<tbody>

<tr class="table-warning" align="center">

<td>

<asp:TextBox ID="txtDetay" runat="server" TextMode="MultiLine" Height="120px" CssClass="form-control" placeholder="Restaurant Hakkında"></asp:TextBox>

</td>

<td>

<asp:FileUpload ID="FileUpload1" runat="server" CssClass="file-upload-browse btn btn-outline-success btn-fw" /><br />

<asp:Label ID="lblLogo" runat="server" Text=""></asp:Label>

<asp:Button ID="Button2" runat="server" Text="Yükle" OnClick="Resim\_Yukle" CssClass="btn btn-inverse-success btn-fw" />

</td>

</tr>

<tr>

<td colspan="4">

<center>

<asp:Label ID="lblHata" runat="server" Text="" Font-Bold></asp:Label>

</center>

</td>

</tr>

</tbody>

</table>

<asp:Button ID="Button1" runat="server" Text="Kaydet" OnClick="Ekle\_Click" CssClass="btn btn-gradient-primary me-2" />

</form>

</div>

</div>

</div>

</asp:Content>

**Admin- Restoran Ekle (cs)**

public partial class RestorantEkle : System.Web.UI.Page

{

Sayfa yüklendiğinde kullanıcı yetkisi yok ise ana sayfaya yönlendirme yapılmaktadır.

protected void Page\_Load(object sender, EventArgs e)

{

if (Session["Yetki"] == null)

{

Response.Redirect("../default.aspx");

}

}

RestoranlarGun restoranlar = new RestoranlarGun();

static int a = -1;

Aşağıdaki “ResimYukle()” metodu sayesinde FileUpload kullanılarak resim belirlenen yola kaydedilir. Eğer belirlenen uzantıya sahip resim eklenmezse hata mesajı verir.

protected void Resim\_Yukle(object sender, EventArgs e)

{

if (FileUpload1.HasFile)

{

if (FileUpload1.PostedFile.ContentType == "image/jpeg" || FileUpload1.PostedFile.ContentType == "image/jpg" || FileUpload1.PostedFile.ContentType == "image/png")

{

string resimAd = FileUpload1.FileName.ToString();

FileUpload1.SaveAs(Server.MapPath("~/images/" + resimAd));

lblLogo.Text = resimAd.ToString();

a = 1;

}

else

{

lblLogo.Text = "Lütfen 'jpeg' , 'jpg' veya 'png' uzantılı resim seçin.";

}

}

else

{

lblLogo.Text = "Lütfen bir resim seçin...";

}

}

Bu butona tıklandığında restoran ekleme yapılırken öncelikle tüm alanların doldurulması beklenir. Tüm alanlar dolu ise restoran adı RestoranlarGun sınıfındaki “RestoranGetir2()” metoduna gönderilir. Geriye dönen veriler DataTable’a atılır. Eğer bu restoran adına sahip restoran var ise hata mesajı gösterilir. Yok ise tüm bilgiler RestoranlarGun sınıfındaki “RestoranEkle()” metoduna gönderilir. Eğer geriye hata dönmezse kayıt yapıldığına dair mesaj gösterilir. İşlem bittikten sonra “Temizle()” metodu çalıştırılır.

protected void Ekle\_Click(object sender, EventArgs e)

{

DataTable dt = new DataTable();

dt = restoranlar.RestoranGetir2(txtAd.Text);

if (txtAd.Text == "" || txtMail.Text == "" || txtTel.Text == "" || txtDetay.Text == "" || a != 1)

{

lblHata.Text = "Lütfen Tüm Alanları Doldurun";

}

else

{

if (dt.Rows.Count > 0)

{

lblHata.Text = "Zaten Bu İsme Sahip Bir Restoran Var...";

}

else

{

bool hata = restoranlar.RestoranEkle(txtAd.Text, txtMail.Text, txtTel.Text, txtDetay.Text,"images/"+ lblLogo.Text);

if (hata == false)

{

lblHata.Text = "Kayıt Başarıyla Eklendi";

}

else

{

lblHata.Text = "Kaydınız yapılamadı.";

}

}

}

Temizle();

}

Bu metot sayesinde bütün textboxların içindeki veriler temizlenir.

void Temizle()

{

lblHata.Text = "";

txtAd.Text = "";

txtDetay.Text = "";

txtMail.Text = "";

txtTel.Text = "";

lblLogo.Text = "";

}

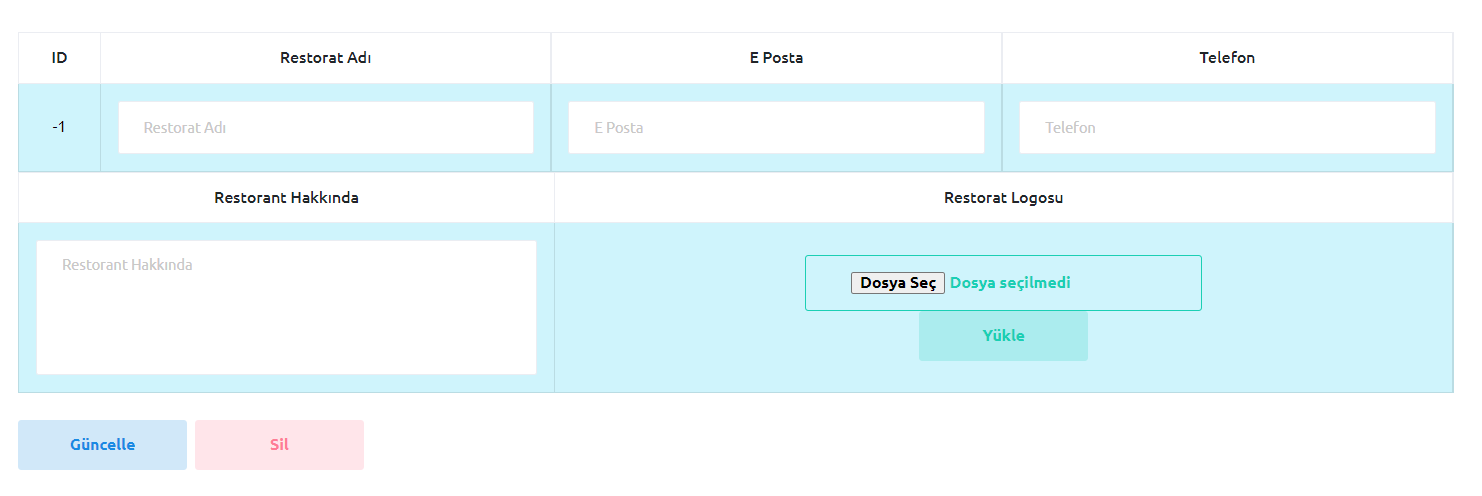
}

|  |  |
| --- | --- |
| public DataTable RestoranGetir2(string kullaniciAdi)  {  DataTable dt = new DataTable();  try  {  using (SqlConnection baglanti = b.BaglantiOlustur())  {  using (SqlCommand komut = new SqlCommand("restoran\_getir\_usp2", baglanti))  {  komut.CommandType = CommandType.StoredProcedure;  komut.Parameters.AddWithValue("@isim", kullaniciAdi);  baglanti.Open();  SqlDataAdapter da;  DataSet ds = new DataSet();  da = new SqlDataAdapter(komut);  da.Fill(ds, "Restoranlar");  dt = ds.Tables["Restoranlar"];  baglanti.Close();  }  }  }  catch (Exception hata)  {  }  return dt;  } | Bu “RestoranGetir2()” metodu sayesinde veri tabanından “restoran\_getir \_usp2” prosedürü çalıştırılıp içerisine gönderilen parametreler ile Restoranlar tablosundaki restoranın verilerini geri gönderir. |
| CREATE proc [dbo].[restoran\_getir\_usp2]  @isim varchar(70)  as  begin  select ID from Restoranlar where resIsim=@isim  end | “restoran\_getir \_usp2” prosedürü |

|  |  |
| --- | --- |
| public bool RestoranEkle(string isim, string eposta, string telefon, string detay,  string resim)  {  bool hataOlustuMu = false;  try  {  using (SqlConnection baglanti = b.BaglantiOlustur())  {  using (SqlCommand komut = new SqlCommand("restoran\_ekle\_usp", baglanti))  {  komut.CommandType = CommandType.StoredProcedure;  komut.Parameters.AddWithValue("@isim", isim);  komut.Parameters.AddWithValue("@eposta", eposta);  komut.Parameters.AddWithValue("@telefon", telefon);  komut.Parameters.AddWithValue("@detay", detay);  komut.Parameters.AddWithValue("@resim", resim);  baglanti.Open();  komut.ExecuteNonQuery();  baglanti.Close();  }  }  }  catch (Exception hata)  {  hataOlustuMu = true;  }  return hataOlustuMu;  } | Bu “RestoranEkle()” metodu sayesinde veri tabanından “restoran\_ekle \_usp” prosedürü çalıştırılıp içerisine gönderilen parametreler ile Restoranlar tablosuna yeni kayıt eklenir. Veri eklenemezse geriye hata gönderilir. |
| CREATE proc [dbo].[rezervasyon\_ekle\_usp]  @kullaniciID int,  @restoranID int,  @kisi bigint,  @tarih datetime  as  insert into Rezervasyon  (KullaniciID,  RestorantID,  KisiSayisi,  RezervasyonTarih,Onay)  values  (@kullaniciID,@restoranID,@kisi,@tarih,0) | “restoran\_ekle \_usp” prosedürü |

**REZERVET- Admin- Restoran Düzenle/Sil**





Bu bölümde Restoran güncelleme ve silme işlemi yapılmaktadır.

<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">

<h3 class="page-title">Restorant Düzenle/Sil Sayfası </h3>

</asp:Content>

<asp:Content ID="Content3" ContentPlaceHolderID="ContentPlaceHolder2" runat="server">

<div class="col-12 grid-margin stretch-card">

<div class="card">

<div class="card-body">

<form class="forms-sample" runat="server">

Restoran verilerinin gösterildiği bölüm.

<asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False" BackColor="White" BorderColor="#feebdc" BorderStyle="None" BorderWidth="1px" CellPadding="3" GridLines="Horizontal" AllowPaging="true" PageSize="5" Width="100%" OnSelectedIndexChanged="GridView1\_SelectedIndexChanged" OnPageIndexChanging="GridView1\_PageIndexChanged">

<PagerSettings Mode="NumericFirstLast" FirstPageText="İlk" LastPageText="Son" />

<AlternatingRowStyle BackColor="#F7F7F7" />

<Columns>

<asp:CommandField ShowSelectButton="True" ButtonType="Button" ControlStyle-BorderColor="#ff8913" SelectText="Seç">

<ControlStyle Height="40px" Width="50px" />

</asp:CommandField>

<asp:BoundField DataField="ID" HeaderText="ID" InsertVisible="False" ReadOnly="True" SortExpression="ID" />

<asp:BoundField DataField="resIsim" HeaderText="Restorand Adı" InsertVisible="False" ReadOnly="True" SortExpression="resIsim" />

<asp:BoundField DataField="eposta" HeaderText="E-Posta" InsertVisible="False" ReadOnly="True" SortExpression="eposta" />

<asp:BoundField DataField="telefon" HeaderText="Telefon" InsertVisible="False" ReadOnly="True" SortExpression="telefon" />

<asp:BoundField DataField="bilgi" HeaderText="Bilgi" InsertVisible="False" ReadOnly="True" SortExpression="bilgi" />

<asp:BoundField DataField="resim" HeaderText="Resim" InsertVisible="False" ReadOnly="True" SortExpression="resim" />

</Columns>

<FooterStyle BackColor="#B5C7DE" ForeColor="#ff8913" />

<HeaderStyle BackColor="#ff8913" Font-Bold="True" ForeColor="#F7F7F7" />

<PagerStyle BackColor="#feebdc" ForeColor="#ff8913" HorizontalAlign="Right" />

<RowStyle BackColor="#feebdc" ForeColor="#ff8913" />

<SelectedRowStyle BackColor="#738A9C" Font-Bold="True" ForeColor="#F7F7F7" />

<SortedAscendingCellStyle BackColor="#F4F4FD" />

<SortedAscendingHeaderStyle BackColor="#5A4C9D" />

<SortedDescendingCellStyle BackColor="#D8D8F0" />

<SortedDescendingHeaderStyle BackColor="#3E3277" />

</asp:GridView>

<br />

<br />

<br />

Restoran verilerinin güncellendiği ve silindiği bölüm.

<table class="table table-bordered">

<thead>

<tr align="center">

<th>ID </th>

<th>Restorat Adı </th>

<th>E Posta </th>

<th>Telefon </th>

</tr>

</thead>

<tbody>

<tr class="table-info" align="center">

<td>

<asp:Label ID="lblId" runat="server" Text="-1"></asp:Label>

</td>

<td>

<asp:TextBox ID="txtAd" runat="server" CssClass="form-control" placeholder="Restorat Adı"></asp:TextBox>

</td>

<td>

<asp:TextBox ID="txtMail" runat="server" TextMode="Email" CssClass="form-control" placeholder="E Posta"></asp:TextBox>

</td>

<td>

<asp:TextBox ID="txtTel" runat="server" TextMode="Phone" CssClass="form-control" placeholder="Telefon"></asp:TextBox>

</td>

</tr>

</tbody>

</table>

<table class="table table-bordered">

<thead>

<tr align="center">

<th>Restorant Hakkında </th>

<th>Restorat Logosu </th>

</tr>

</thead>

<tbody>

<tr class="table-info" align="center">

<td>

<asp:TextBox ID="txtDetay" runat="server" TextMode="MultiLine" Height="120px" CssClass="form-control" placeholder="Restorant Hakkında"></asp:TextBox>

</td>

<td>

<asp:FileUpload ID="FileUpload1" runat="server" CssClass="file-upload-browse btn btn-outline-success btn-fw" /><br />

<asp:Button ID="Button2" runat="server" Text="Yükle" CssClass="btn btn-inverse-success btn-fw" OnClick="Resim\_Yukle" />

<asp:Label ID="lblLogo" runat="server" Text=""></asp:Label>

</td>

</tr>

</tbody>

</table>

<br />

<asp:Button ID="Button1" runat="server" Text="Güncelle" CssClass="btn btn-inverse-info btn-fw" OnClick="Guncelle" />&nbsp;

<asp:Button ID="Button3" runat="server" Text="Sil" CssClass="btn btn-inverse-danger btn-fw" OnClick="Sil" />

<asp:Label ID="lblHata" runat="server" Text="" Font-Bold></asp:Label>

</form>

</div>

</div>

</div>

</asp:Content>

**Admin- Restoran Düzenle/Sil (cs)**

public partial class RestorantDuzenleSil : System.Web.UI.Page

{

Baglan b = new Baglan();

RestoranlarGun restoranlar = new RestoranlarGun();

Sayfa yüklendiğinde kullanıcı yetkisi yok ise ana sayfaya yönlendirme yapılmaktadır. Yetki var ise “VeriGetir()” metodu çalıştırılarak veriler getirilir.

protected void Page\_Load(object sender, EventArgs e)

{

if (Session["Yetki"] == null)

{

Response.Redirect("../default.aspx");

}

VeriGetir();

}

Bu butona tıklandığında RestoranlarGun sınıfındaki “RestoranGuncelle()” metoduna veriler gönderilir. Güncelleme işlemi yapılır. Daha sonra “VeriGetir()” metodu çalıştırılarak veriler getirilir.

protected void Guncelle(object sender, EventArgs e)

{

restoranlar.RestoranGuncelle(lblId.Text, txtAd.Text, txtMail.Text, txtTel.Text, txtDetay.Text, "images/"+lblLogo.Text);

VeriGetir();

}

Bu butona tıklandığında RestoranlarGun sınıfındaki “RestoranSil()” metoduna restoran ID verisi gönderilir. Silme işlemi yapılır. Daha sonra “VeriGetir()” metodu çalıştırılarak veriler getirilir.

protected void Sil(object sender, EventArgs e)

{

restoranlar.RestoranSil(Convert.ToInt32(lblId.Text));

VeriGetir();

}

GridView’de bir sütun seçildiğinde seçilen sütundaki veriler güncelleme işlemi yapılacak olan textlerde gösterilir.

protected void GridView1\_SelectedIndexChanged(object sender, EventArgs e)

{

lblId.Text = GridView1.SelectedRow.Cells[1].Text;

txtAd.Text = HttpUtility.HtmlDecode(GridView1.SelectedRow.Cells[2].Text.ToString());

txtMail.Text = HttpUtility.HtmlDecode(GridView1.SelectedRow.Cells[3].Text.ToString());

txtTel.Text = HttpUtility.HtmlDecode(GridView1.SelectedRow.Cells[4].Text.ToString());

txtDetay.Text = HttpUtility.HtmlDecode(GridView1.SelectedRow.Cells[5].Text.ToString());

lblLogo.Text = HttpUtility.HtmlDecode(GridView1.SelectedRow.Cells[6].Text.ToString());

}

GridView’de sayfalama işlemi yapmaktadır.

protected void GridView1\_PageIndexChanged(object sender, GridViewPageEventArgs e)

{

GridView1.PageIndex = e.NewPageIndex;

VeriGetir();

}

Aşağıdaki “ResimYukle()” metodu sayesinde FileUpload kullanılarak resim belirlenen yola kaydedilir. Eğer belirlenen uzantıya sahip resimler eklenmezse hata mesajı verir.

protected void Resim\_Yukle(object sender, EventArgs e)

{

if (FileUpload1.HasFile)

{

if (FileUpload1.PostedFile.ContentType == "image/jpeg" || FileUpload1.PostedFile.ContentType == "image/jpg" || FileUpload1.PostedFile.ContentType == "image/png")

{

string resimAd = FileUpload1.FileName.ToString();

FileUpload1.SaveAs(Server.MapPath("~/images/" + resimAd));

lblLogo.Text = resimAd.ToString();

}

else

{

lblLogo.Text = "Lütfen 'jpeg' , 'jpg' veya 'png' uzantılı resim seçin.";

}

}

else

{

lblLogo.Text = "Lütfen bir resim seçin...";

}

}

Bu metot sayesinde GridView’e RestoranlarGun sınıfındaki “RestoranGoster()” metodundan gelen veriler eklenir.

protected void VeriGetir()

{

GridView1.DataSource = restoranlar.RestoranGoster();

GridView1.DataBind();

}

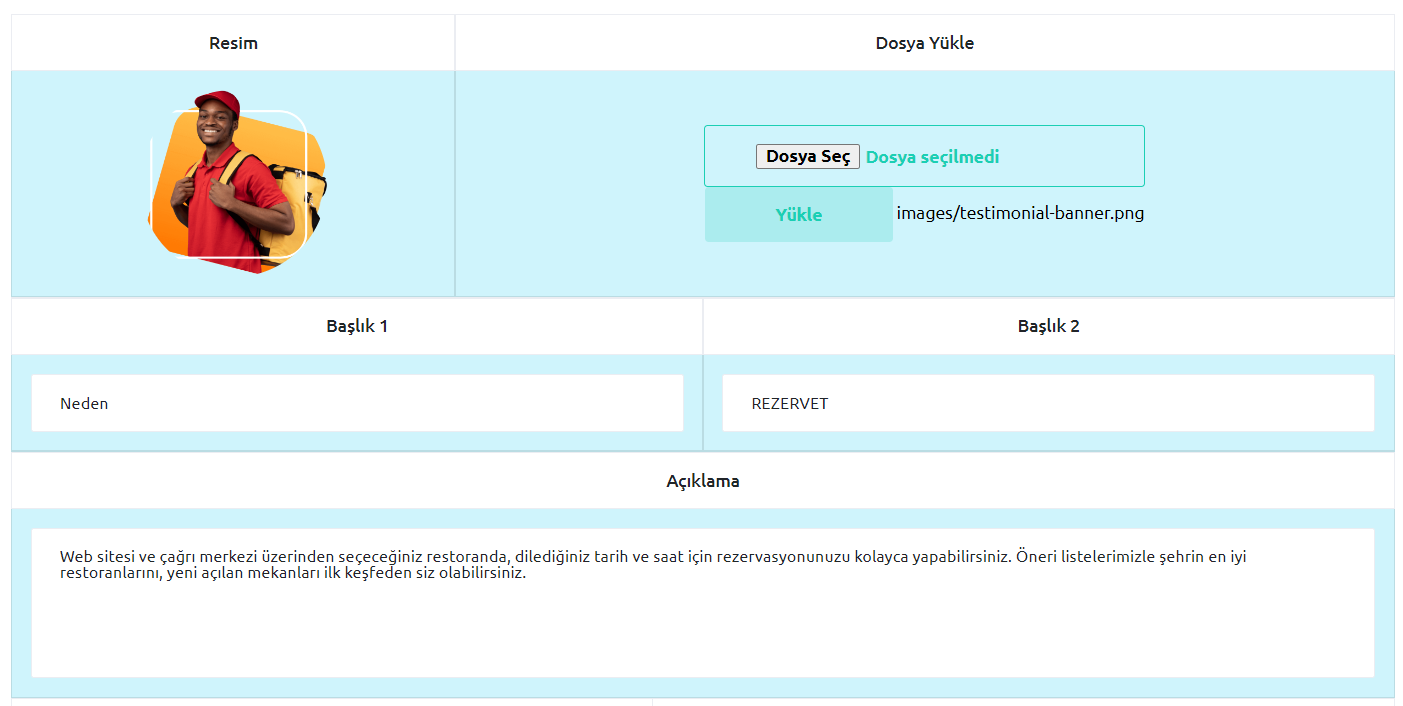
}

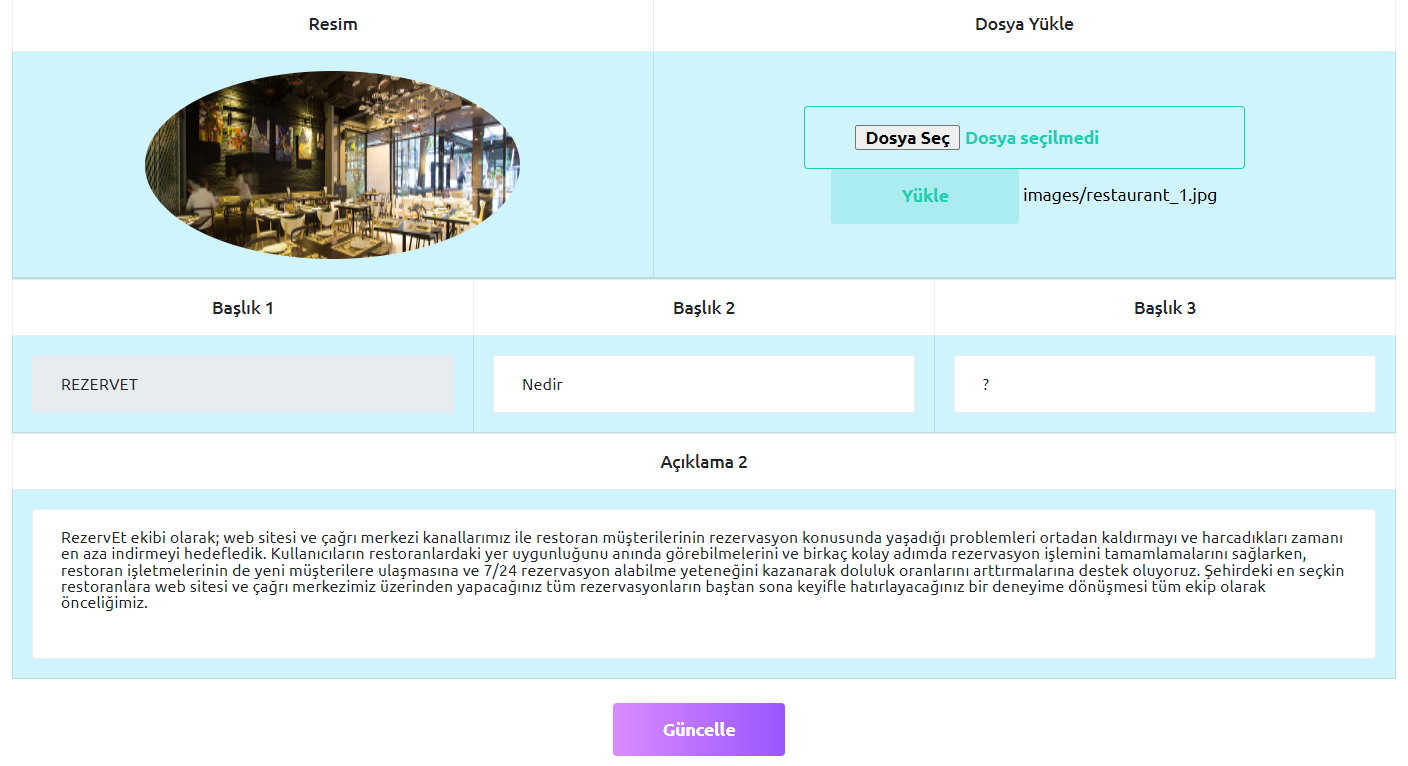
|  |  |
| --- | --- |
| public bool RestoranGuncelle(string ID, string isim, string eposta, string telefon, string bilgi, string resim)  {  bool hataOlustuMu = false;  try  {  using (SqlConnection baglanti = b.BaglantiOlustur())  {  using (SqlCommand komut = new SqlCommand("restoran\_guncelle\_usp", baglanti))  {  komut.CommandType = CommandType.StoredProcedure;  komut.Parameters.AddWithValue("@ID", ID);  komut.Parameters.AddWithValue("@isim", isim);  komut.Parameters.AddWithValue("@eposta", eposta);  komut.Parameters.AddWithValue("@telefon", telefon);  komut.Parameters.AddWithValue("@bilgi", bilgi);  komut.Parameters.AddWithValue("@resim", resim);    baglanti.Open();  komut.ExecuteNonQuery();  baglanti.Close();  }  }  }  catch (Exception hata)  {  hataOlustuMu = true;  }  return hataOlustuMu;  } | Bu “SiteGenelGuncelle()” metodu sayesinde veri tabanından “restoran\_guncelle \_usp” prosedürü çalıştırılıp içerisine gönderilen parametreler ile Restoranlar tablosundaki restoran verisi güncellenir. Veri güncellenmezse geriye hata gönderilir. |
| CREATE proc [dbo].[restoran\_guncelle\_usp]  @ID int,  @isim varchar(70),@eposta varchar(70),@telefon varchar(15),  @bilgi nvarchar(500),@resim nvarchar(50)  as  update Restoranlar set resIsim=@isim,  eposta=@eposta,  telefon=@telefon,  bilgi=@bilgi,  resim=@resim  where ID=@ID | “restoran\_guncelle \_usp” prosedürü |

|  |  |
| --- | --- |
| public bool RestoranSil(int ID)  {  bool hataOlustuMu = false;  try  {  using (SqlConnection baglanti = b.BaglantiOlustur())  {  using (SqlCommand komut = new SqlCommand("restoran\_sil\_usp", baglanti))  {  komut.CommandType = CommandType.StoredProcedure;  komut.Parameters.AddWithValue("@ID", ID);  baglanti.Open();  komut.ExecuteNonQuery();  baglanti.Close();  }  }  }  catch (Exception hata)  {    hataOlustuMu = true;  }  return hataOlustuMu;  } | Bu “RestoranSil()” metodu sayesinde veri tabanından “restoran\_sil \_usp” prosedürü çalıştırılıp içerisine gönderilen parametre ile Restoranlar tablosundaki restoran verisi silinir. Veri silinmezse geriye hata gönderilir. |
| CREATE proc [dbo].[restoran\_sil\_usp]  @ID int  as  begin  DELETE FROM Restoranlar WHERE ID=@ID  End | “restoran\_sil \_usp” prosedürü |

|  |  |
| --- | --- |
| public DataTable RestoranGoster()  {  DataTable dt = new DataTable();  try  {  using (SqlConnection baglanti = b.BaglantiOlustur())  {  using (SqlCommand komut = new SqlCommand("restoran\_usp", baglanti))  {  komut.CommandType = CommandType.StoredProcedure;  baglanti.Open();  SqlDataAdapter da;  DataSet ds = new DataSet();  da = new SqlDataAdapter(komut);  da.Fill(ds, "Kullanici");  dt = ds.Tables["Kullanici"];  baglanti.Close();  }  }  }  catch (Exception hata)  {  }  return dt;  } | Bu “RestoranGoster()” metodu sayesinde veri tabanından “restoran\_ usp” prosedürü çalıştırılıp Restoranlar tablosundaki tüm restoranların verilerini geri gönderir. |
| CREATE proc [dbo].[restoran\_usp]  as  begin  select \* from Restoranlar  end | “restoran\_ usp” prosedürü |

**REZERVET- Admin- Hakkımda Düzenle**





Bu bölümde Hakkımda sayfasının güncelleme işlemi yapılmaktadır.

<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">

<h3 class="page-title">Hakkımda Bilgi Düzenleme Sayfası </h3>

</asp:Content>

<asp:Content ID="Content3" ContentPlaceHolderID="ContentPlaceHolder2" runat="server">

<div class="col-12 grid-margin stretch-card">

<div class="card">

<div class="card-body">

<form class="forms-sample" runat="server">

<table class="table table-bordered">

<thead>

<tr align="center">

<th>Resim </th>

<th>Dosya Yükle </th>

</tr>

</thead>

<tbody>

<tr class="table-info" align="center">

<td>

<asp:Image ID="Image1" Width="150" Height="150" runat="server" />

</td>

<td>

<asp:FileUpload ID="FileUpload1" runat="server" CssClass="file-upload-browse btn btn-outline-success btn-fw" /><br />

<asp:Button ID="Button4" runat="server" Text="Yükle" CssClass="btn btn-inverse-success btn-fw" OnClick="Resim\_Yukle" />

<asp:Label ID="lblLogo" runat="server" Text=""></asp:Label>

</td>

</tr>

</tbody>

</table>

<table class="table table-bordered">

<thead>

<tr align="center">

<th>Başlık 1 </th>

<th>Başlık 2 </th>

</tr>

</thead>

<tbody>

<tr class="table-info" align="center">

<td>

<asp:TextBox ID="baslik1a" runat="server" CssClass="form-control" ></asp:TextBox>

</td>

<td>

<asp:TextBox ID="baslik1b" runat="server" CssClass="form-control" ></asp:TextBox>

</td>

</tr>

</tbody>

</table>

<table class="table table-bordered">

<thead>

<tr align="center">

<th>Açıklama</th>

</tr>

</thead>

<tbody>

<tr class="table-info" align="center">

<td>

<asp:TextBox ID="aciklama1" runat="server" TextMode="MultiLine" Height="120px" CssClass="form-control" ></asp:TextBox>

</td>

</tr>

</tbody>

</table>

<table class="table table-bordered">

<thead>

<tr align="center">

<th>Resim </th>

<th>Dosya Yükle </th>

</tr>

</thead>

<tbody>

<tr class="table-info" align="center">

<td>

<asp:Image ID="Image2" Width="300" Height="150" runat="server" />

</td>

<td>

<asp:FileUpload ID="FileUpload2" runat="server" CssClass="file-upload-browse btn btn-outline-success btn-fw" /><br />

<asp:Button ID="Button5" runat="server" Text="Yükle" CssClass="btn btn-inverse-success btn-fw" OnClick="Resim\_Yukle2" />

<asp:Label ID="lblLogo2" runat="server" Text=""></asp:Label>

</td>

</tr>

</tbody>

</table>

<table class="table table-bordered">

<thead>

<tr align="center">

<th>Başlık 1 </th>

<th>Başlık 2 </th>

<th>Başlık 3 </th>

</tr>

</thead>

<tbody>

<tr class="table-info" align="center">

<td>

<asp:TextBox ID="baslik2c" runat="server" ReadOnly="true" CssClass="form-control" ></asp:TextBox>

</td>

<td>

<asp:TextBox ID="baslik2a" runat="server" CssClass="form-control" ></asp:TextBox>

</td>

<td>

<asp:TextBox ID="baslik2b" runat="server" CssClass="form-control" ></asp:TextBox>

</td>

</tr>

</tbody>

</table>

<table class="table table-bordered">

<thead>

<tr align="center">

<th>Açıklama 2 </th>

</tr>

</thead>

<tbody>

<tr class="table-info" align="center">

<td>

<asp:TextBox ID="aciklama2" runat="server" TextMode="MultiLine" Height="120px" CssClass="form-control" ></asp:TextBox>

</td>

</tr>

</tbody>

</table>

<tr><br />

<center><asp:Button ID="Button1" runat="server" Text="Güncelle" OnClick="Guncelle" CssClass="btn btn-gradient-primary me-2" />

<asp:Label ID="lblHata" runat="server" Text="" Font-Bold></asp:Label></center>

</tr>

</form>

</div>

</div>

</div>

</asp:Content>

**Admin- Hakkımda Düzenle (cs)**

public partial class HakkimdaDuzenleSil : System.Web.UI.Page

{

Bilgi bilgi = new Bilgi();

HakkimdaGun hakkimda = new HakkimdaGun();

Sayfa yüklendiğinde kullanıcı yetkisi yok ise ana sayfaya yönlendirme yapılmaktadır. Yetki var ise “VeriGetir()” metodu çalıştırılarak veriler getirilir.

protected void Page\_Load(object sender, EventArgs e)

{

if (Session["Yetki"] == null)

{

Response.Redirect("../default.aspx");

}

if (!IsPostBack)

{

VeriGetir();

}

}

Bu butona tıklandığında HakkimdaGun sınıfındaki “HakkimdaGuncelle()” metoduna veriler gönderilir. Güncelleme işlemi yapılır. Daha sonra “VeriGetir()” metodu çalıştırılarak veriler getirilir.

protected void Guncelle(object sender, EventArgs e)

{

hakkimda.HakkimdaGuncelle(baslik1a.Text, baslik1b.Text, baslik2a.Text, baslik2b.Text, aciklama1.Text, aciklama2.Text, lblLogo.Text, lblLogo2.Text);

VeriGetir();

}

Bu metot sayesinde DataTable’a bilgi sınıfındaki “Hakkimda()” metodundan gelen veriler eklenir. Bu veriler güncelleme yapılacak olan textlerde gösterilir.

public void VeriGetir()

{

DataTable dt = new DataTable();

dt = bilgi.Hakkimda();

baslik1a.Text = dt.Rows[0][1].ToString();

baslik1b.Text = dt.Rows[0][2].ToString();

baslik2c.Text = dt.Rows[0][2].ToString();

baslik2a.Text = dt.Rows[0][3].ToString();

baslik2b.Text = dt.Rows[0][4].ToString();

aciklama1.Text = dt.Rows[0][5].ToString();

aciklama2.Text = dt.Rows[0][6].ToString();

Image1.ImageUrl = "../" + dt.Rows[0][7].ToString();

lblLogo.Text = dt.Rows[0][7].ToString();

Image2.ImageUrl = "../" + dt.Rows[0][8].ToString();

lblLogo2.Text = dt.Rows[0][8].ToString();

}

Aşağıdaki “ResimYukle()” metodları sayesinde FileUpload kullanılarak resimler belirlenen yollara kaydedilir. Eğer belirlenen uzantıya sahip resimler eklenmezse hata mesajı verir.

protected void Resim\_Yukle(object sender, EventArgs e)

{

if (FileUpload1.HasFile)

{

if (FileUpload1.PostedFile.ContentType == "image/jpeg" || FileUpload1.PostedFile.ContentType == "image/jpg" || FileUpload1.PostedFile.ContentType == "image/png")

{

string resimAd = FileUpload1.FileName.ToString();

FileUpload1.SaveAs(Server.MapPath("~/images/" + resimAd));

lblLogo.Text = "images/" + resimAd.ToString();

}

else

{

lblLogo.Text = "Lütfen 'jpeg' , 'jpg' veya 'png' uzantılı resim seçin.";

}

}

else

{

lblLogo.Text = "Lütfen bir resim seçin...";

}

}

protected void Resim\_Yukle2(object sender, EventArgs e)

{

if (FileUpload1.HasFile)

{

if (FileUpload2.PostedFile.ContentType == "image/jpeg" || FileUpload2.PostedFile.ContentType == "image/jpg" || FileUpload2.PostedFile.ContentType == "image/png")

{

string resimAd = FileUpload2.FileName.ToString();

FileUpload2.SaveAs(Server.MapPath("~/images/" + resimAd));

lblLogo2.Text = "images/" + resimAd.ToString();

}

else

{

lblLogo2.Text = "Lütfen 'jpeg' , 'jpg' veya 'png' uzantılı resim seçin.";

}

}

else

{

lblLogo2.Text = "Lütfen bir resim seçin...";

}

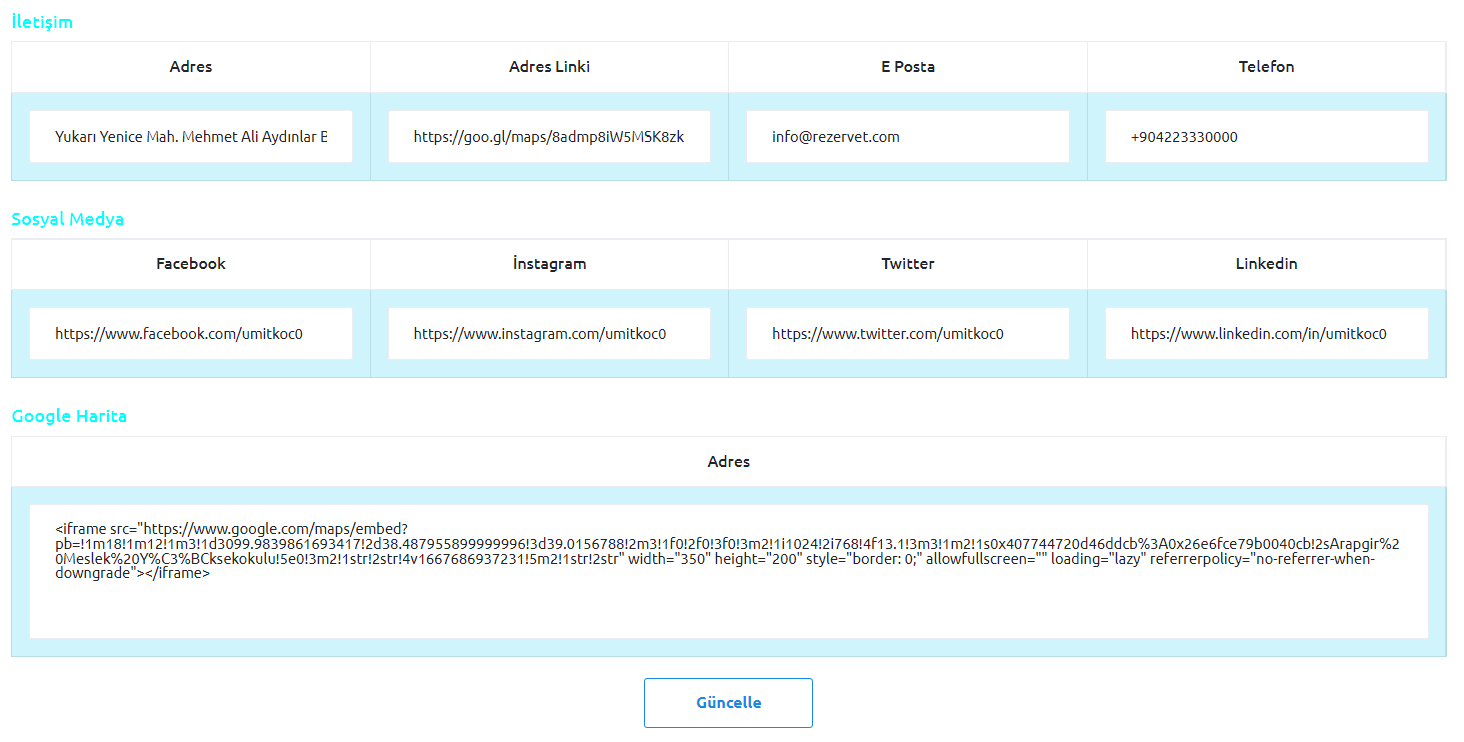
}

}

|  |  |
| --- | --- |
| public bool HakkimdaGuncelle(string bas1a, string bas1b, string bas2a, string bas2b, string acklm1, string acklm2, string resim1, string resim2)  {  bool hataOlustuMu = false;  try  {  using (SqlConnection baglanti = b.BaglantiOlustur())  {  using (SqlCommand komut = new SqlCommand("hakkimda\_guncelle\_usp", baglanti))  {  komut.CommandType = CommandType.StoredProcedure;  komut.Parameters.AddWithValue("@bas1a", bas1a);  komut.Parameters.AddWithValue("@bas1b", bas1b);  komut.Parameters.AddWithValue("@bas2a", bas2a);  komut.Parameters.AddWithValue("@bas2b", bas2b);  komut.Parameters.AddWithValue("@acklm1", acklm1);  komut.Parameters.AddWithValue("@acklm2", acklm2);  komut.Parameters.AddWithValue("@resim1", resim1);  komut.Parameters.AddWithValue("@resim2", resim2);  baglanti.Open();  komut.ExecuteNonQuery();  baglanti.Close();  }  }  }  catch (Exception hata)  {  hataOlustuMu = true;  }  return hataOlustuMu;  } | Bu “HakkimdaGuncelle()” metodu sayesinde veri tabanından “hakkimda\_guncelle \_usp” prosedürü çalıştırılıp içerisine gönderilen parametreler ile Hakkimda tablosundaki veriler güncellenir. Veri güncellenmezse geriye hata gönderilir. |
| CREATE proc [dbo].[hakkimda\_guncelle\_usp]  @bas1a nvarchar(50),  @bas1b nvarchar(50),  @bas2a nvarchar(50),  @bas2b nvarchar(50),  @acklm1 nvarchar(500),  @acklm2 nvarchar(MAX),  @resim1 nvarchar(50),  @resim2 nvarchar(50)  as  update hakkimda set  baslik1a= @bas1a ,  baslik1b= @bas1b ,  baslik2a= @bas2a ,  baslik2b= @bas2b ,  aciklama1= @acklm1 ,  aciklama2= @acklm2 ,  resim1= @resim1,  resim2= @resim2 | “hakkimda\_guncelle \_usp” prosedürü |

|  |  |
| --- | --- |
| public DataTable Hakkimda()  {  DataTable dt = new DataTable();  try  {  using (SqlConnection baglanti = b.BaglantiOlustur())  {  using (SqlCommand komut = new SqlCommand("hakkimda\_usp", baglanti))  {  komut.CommandType = CommandType.StoredProcedure;  baglanti.Open();  SqlDataAdapter da;  DataSet ds = new DataSet();  da = new SqlDataAdapter(komut);  da.Fill(ds, "Hakkimda");  dt = ds.Tables["Hakkimda"];  baglanti.Close();  }  }  }  catch (Exception hata)  {  }  return dt;  } | Bu “Hakkimda()” metodu sayesinde veri tabanından “hakkimda\_ usp” prosedürü çalıştırılıp Hakkimda tablosundaki tüm verileri geri gönderir. |
| CREATE proc [dbo].[hakkimda\_usp]  as  begin  select \* from Hakkimda  end | “hakkimda\_ usp” prosedürü |

**REZERVET- Admin- İletişim Düzenle**



Bu bölümde İletişim sayfasının güncelleme işlemi yapılmaktadır.

<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">

<h3 class="page-title">İletişim Bilgisi Düzenleme Sayfası </h3>

</asp:Content>

<asp:Content ID="Content3" ContentPlaceHolderID="ContentPlaceHolder2" runat="server">

<div class="col-12 grid-margin stretch-card">

<div class="card">

<div class="card-body">

<form class="forms-sample" runat="server" >

<div class="form-group">

<h5 style="color: aqua">İletişim</h5>

<table class="table table-bordered">

<thead>

<tr align="center">

<th>Adres </th>

<th>Adres Linki </th>

<th>E Posta </th>

<th>Telefon </th>

</tr>

</thead>

<tbody>

<tr class="table-info" align="center">

<td>

<asp:TextBox ID="adres" runat="server" CssClass="form-control" ></asp:TextBox>

</td>

<td>

<asp:TextBox ID="adresLink" runat="server" CssClass="form-control"></asp:TextBox>

</td>

<td>

<asp:TextBox ID="eposta" runat="server" CssClass="form-control"></asp:TextBox>

</td>

<td>

<asp:TextBox ID="telefon" runat="server" CssClass="form-control"></asp:TextBox>

</td>

</tr>

</tbody>

</table>

</div>

<div class="form-group">

<h5 style="color: aqua">Sosyal Medya</h5>

<table class="table table-bordered">

<thead>

<tr align="center">

<th>Facebook </th>

<th>İnstagram </th>

<th>Twitter </th>

<th>Linkedin </th>

</tr>

</thead>

<tbody>

<tr class="table-info" align="center">

<td>

<asp:TextBox ID="facebook" runat="server" CssClass="form-control"></asp:TextBox>

</td>

<td>

<asp:TextBox ID="instagram" runat="server" CssClass="form-control"></asp:TextBox>

</td>

<td>

<asp:TextBox ID="twitter" runat="server" CssClass="form-control"></asp:TextBox>

</td>

<td>

<asp:TextBox ID="linkedin" runat="server" CssClass="form-control"></asp:TextBox>

</td>

</tr>

</tbody>

</table>

</div>

<div class="form-group">

<h5 style="color: aqua">Google Harita</h5>

<table class="table table-bordered">

<thead>

<tr align="center">

<th>Adres </th>

</tr>

</thead>

<tbody>

<tr class="table-info" align="center">

<td colspan="4">

<asp:TextBox ID="googleMap" runat="server" TextMode="MultiLine" Height="120px" CssClass="form-control"></asp:TextBox>

</td>

</tr>

</tbody>

</table><br />

<center>

<asp:Button ID="guncelle" runat="server" Text="Güncelle" OnClick="Guncelle" CssClass="btn btn-outline-info btn-fw" />

<asp:Label ID="lblHata" runat="server" Text="" Font-Bold></asp:Label>

</center>

</div>

</form>

</div>

</div>

</div>

</asp:Content>

**Admin- İletişim Düzenle (cs)**

public partial class IletisimDuzenleSil : System.Web.UI.Page

{

Bilgi bilgi = new Bilgi();

IletisimGun iletisim = new IletisimGun();

Sayfa yüklendiğinde kullanıcı yetkisi yok ise ana sayfaya yönlendirme yapılmaktadır. Yetki var ise “VeriGetir()” metodu çalıştırılarak veriler getirilir.

protected void Page\_Load(object sender, EventArgs e)

{

if (Session["Yetki"] == null)

{

Response.Redirect("../default.aspx");

}

if (!IsPostBack)

{

VeriGetir();

}

}

Bu butona tıklandığında IletisimGun sınıfındaki “IletisimGuncelle()” metoduna veriler gönderilir. Güncelleme işlemi yapılır. Daha sonra “VeriGetir()” metodu çalıştırılarak veriler getirilir.

protected void Guncelle(object sender, EventArgs e)

{

iletisim.IletisimGuncelle(adres.Text, adresLink.Text, eposta.Text, telefon.Text, facebook.Text, instagram.Text, twitter.Text, linkedin.Text, googleMap.Text);

VeriGetir();

}

Bu metot sayesinde DataTable’a bilgi sınıfındaki “SiteGenel()” metodundan gelen veriler eklenir. Bu veriler güncelleme yapılacak olan textlerde gösterilir.

public void VeriGetir()

{

DataTable dt = new DataTable();

dt = bilgi.SiteGenel();

adres.Text = dt.Rows[0][4].ToString();

adresLink.Text = dt.Rows[0][20].ToString();

eposta.Text = dt.Rows[0][5].ToString();

telefon.Text = dt.Rows[0][6].ToString();

facebook.Text = dt.Rows[0][7].ToString();

instagram.Text = dt.Rows[0][8].ToString();

twitter.Text = dt.Rows[0][9].ToString();

linkedin.Text = dt.Rows[0][10].ToString();

googleMap.Text = dt.Rows[0][11].ToString();

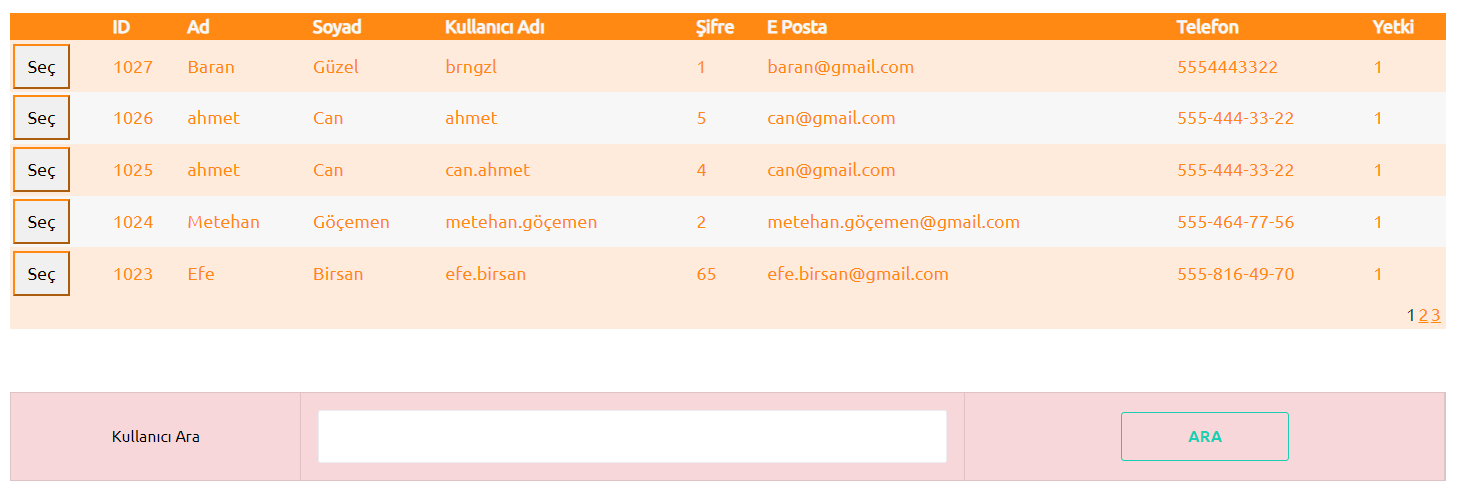
}

}

|  |  |
| --- | --- |
| public bool IletisimGuncelle(string adres, string adresLink, string eposta, string telefon, string facebook, string instagram, string twitter, string linkedin, string googleMap)  {  bool hataOlustuMu = false;  try  {  using (SqlConnection baglanti = b.BaglantiOlustur())  {  using (SqlCommand komut = new SqlCommand("iletisim\_guncelle\_usp", baglanti))  {  komut.CommandType = CommandType.StoredProcedure;  komut.Parameters.AddWithValue("@adres", adres);  komut.Parameters.AddWithValue("@adresLink", adresLink);  komut.Parameters.AddWithValue("@eposta", eposta);  komut.Parameters.AddWithValue("@telefon", telefon);  komut.Parameters.AddWithValue("@facebook", facebook);  komut.Parameters.AddWithValue("@instagram", instagram);  komut.Parameters.AddWithValue("@twitter", twitter);  komut.Parameters.AddWithValue("@linkedin", linkedin);  komut.Parameters.AddWithValue("@googleMap", googleMap);  baglanti.Open();  komut.ExecuteNonQuery();  baglanti.Close();  }  }  }  catch (Exception hata)  {  hataOlustuMu = true;  }  return hataOlustuMu;  } | Bu “IletisimGuncelle()” metodu sayesinde veri tabanından “iletisim\_guncelle \_usp” prosedürü çalıştırılıp içerisine gönderilen parametreler ile SiteGenel tablosundaki veriler güncellenir. Veri güncellenmezse geriye hata gönderilir. |
| CREATE proc [dbo].[iletisim\_guncelle\_usp]  @adres nvarchar(300),  @adresLink nvarchar(300),  @eposta nvarchar(100),  @telefon nvarchar(50),  @facebook nvarchar(300),  @instagram nvarchar(300),  @twitter nvarchar(300),  @linkedin nvarchar(300),  @googleMap nvarchar(500)  as  update SiteGenel set  adres= @adres ,  adresLink= @adresLink ,  eposta= @eposta ,  telefon= @telefon ,  facebook= @facebook ,  instagram= @instagram ,  twitter= @twitter ,  linkedin= @linkedin ,  googleMap= @googleMap  where ID=1 | “iletisim\_guncelle \_usp” prosedürü |

|  |  |
| --- | --- |
| public DataTable SiteGenel()  {  DataTable dt = new DataTable();  try  {  using (SqlConnection baglanti = b.BaglantiOlustur())  {  using (SqlCommand komut = new SqlCommand("site\_genel\_usp", baglanti))  {  komut.CommandType = CommandType.StoredProcedure;  baglanti.Open();  SqlDataAdapter da;  DataSet ds = new DataSet();  da = new SqlDataAdapter(komut);  da.Fill(ds, "Site");  dt = ds.Tables["Site"];  baglanti.Close();  }  }  }  catch (Exception hata)  {  }  return dt;  } | Bu “SiteGenel()” metodu sayesinde veri tabanından “site\_genel\_ usp” prosedürü çalıştırılıp SiteGenel tablosundaki tüm verileri geri gönderir. |
| CREATE proc [dbo].[site\_genel\_usp]  as  begin  select \* from SiteGenel  end | “site\_genel\_ usp” prosedürü |

**REZERVET- Admin- Kullanıcı Düzenle**





Bu bölümde Kullanıcı bilgilerinin güncelleme, silme ve ekleme işlemi yapılmaktadır.

<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">

<meta charset="utf-8" />

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">

<h3 class="page-title">Kullanıcı Bilgileri Düzenleme Sayfası </h3>

</asp:Content>

<asp:Content ID="Content3" ContentPlaceHolderID="ContentPlaceHolder2" runat="server">

<form runat="server">

<div class="col-lg-12 stretch-card">

<div class="card">

<div class="card-body">

Kullanıcı verilerinin gösterildiği bölüm.

<asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False" BackColor="White" BorderColor="#feebdc" BorderStyle="None" BorderWidth="1px" CellPadding="3" GridLines="Horizontal" AllowPaging="true" PageSize="5" Width="100%" OnSelectedIndexChanged="GridView1\_SelectedIndexChanged" OnPageIndexChanging="GridView1\_PageIndexChanged">

<PagerSettings Mode="NumericFirstLast" FirstPageText="İlk" LastPageText="Son" />

<AlternatingRowStyle BackColor="#F7F7F7" />

<Columns>

<asp:CommandField ShowSelectButton="True" ButtonType="Button" ControlStyle-BorderColor="#ff8913" SelectText="Seç">

<ControlStyle Height="40px" Width="50px" />

</asp:CommandField>

<asp:BoundField DataField="ID" HeaderText="ID" InsertVisible="False" ReadOnly="True" SortExpression="ID" />

<asp:BoundField DataField="isim" HeaderText="Ad" InsertVisible="False" ReadOnly="True" SortExpression="isim" />

<asp:BoundField DataField="soyisim" HeaderText="Soyad" InsertVisible="False" ReadOnly="True" SortExpression="soyisim" />

<asp:BoundField DataField="kullaniciAdi" HeaderText="Kullanıcı Adı" InsertVisible="False" ReadOnly="True" SortExpression="kullaniciAdi" />

<asp:BoundField DataField="sifre" HeaderText="Şifre" InsertVisible="False" ReadOnly="True" SortExpression="sifre" />

<asp:BoundField DataField="eposta" HeaderText="E Posta" InsertVisible="False" ReadOnly="True" SortExpression="eposta" />

<asp:BoundField DataField="telefon" HeaderText="Telefon" InsertVisible="False" ReadOnly="True" SortExpression="telefon" />

<asp:BoundField DataField="yetki" HeaderText="Yetki" InsertVisible="False" ReadOnly="True" SortExpression="yetki" />

</Columns>

<FooterStyle BackColor="#B5C7DE" ForeColor="#ff8913" />

<HeaderStyle BackColor="#ff8913" Font-Bold="True" ForeColor="#F7F7F7" />

<PagerStyle BackColor="#feebdc" ForeColor="#ff8913" HorizontalAlign="Right" />

<RowStyle BackColor="#feebdc" ForeColor="#ff8913" />

<SelectedRowStyle BackColor="#738A9C" Font-Bold="True" ForeColor="#F7F7F7" />

<SortedAscendingCellStyle BackColor="#F4F4FD" />

<SortedAscendingHeaderStyle BackColor="#5A4C9D" />

<SortedDescendingCellStyle BackColor="#D8D8F0" />

<SortedDescendingHeaderStyle BackColor="#3E3277" />

</asp:GridView>

<br />

<br />

<br />

<center>

<table class="table table-bordered">

<tbody>

<tr class="table-danger" align="center">

<td>Kullanıcı Ara

</td>

<td>

<asp:TextBox ID="aratxt" runat="server" CssClass="form-control"></asp:TextBox>

</td>

<td>

<asp:Button ID="ara" runat="server" Text="ARA" CssClass="btn btn-outline-success btn-fw" OnClick="ara\_Click" />

</td>

</tr>

</tbody>

</table>

</center>

<br />

<br />

Restoran bilgilerinin düzenleme, silme ve ekleme yapıldığı bölüm.

<table class="table table-bordered">

<thead>

<tr align="center">

<th width="77">ID </th>

<th>Ad </th>

<th>Soyad </th>

<th>E Posta </th>

</tr>

</thead>

<tbody>

<tr class="table-info" align="center">

<td width="77">

<asp:Label ID="idlbl" runat="server" Text="-1"></asp:Label>

</td>

<td>

<asp:TextBox ID="adtxt" runat="server" CssClass="form-control"></asp:TextBox>

</td>

<td>

<asp:TextBox ID="soyadtxt" runat="server" CssClass="form-control"></asp:TextBox>

</td>

<td>

<asp:TextBox ID="epostatxt" runat="server" CssClass="form-control"></asp:TextBox>

</td>

</tr>

</tbody>

</table>

<table class="table table-bordered">

<thead>

<tr align="center">

<th>Yetki </th>

<th>Kullanıcı Adı</th>

<th>Şifre </th>

<th>Telefon</th>

</tr>

</thead>

<tbody>

<tr class="table-success" align="center">

<td width="80">

<asp:TextBox Width="80" TextMode="Number" ID="yetkitxt" runat="server" CssClass="form-control"></asp:TextBox>

</td>

<td>

<asp:TextBox ID="kaditxt" runat="server" CssClass="form-control"></asp:TextBox>

</td>

<td>

<asp:TextBox ID="sifretxt" runat="server" CssClass="form-control"></asp:TextBox>

</td>

<td>

<asp:TextBox TextMode="Phone" ID="teltxt" runat="server" CssClass="form-control"></asp:TextBox>

</td>

</tr>

<tr>

<td colspan="4">

<center>

<asp:Label ID="hatalbl" CssClass="text-danger" runat="server"></asp:Label>

</center>

</td>

</tr>

</tbody>

</table>

<br />

<br />

<br />

<table>

<tr>

<td>

<asp:Button ID="temizle" runat="server" Text="TEMİZLE" CssClass="btn btn-primary btn-fw" OnClick="temizle\_Click" />

</td>

<td>

<asp:Button ID="ekle" runat="server" Text="EKLE" CssClass="btn btn-success btn-fw" OnClick="ekle\_Click" />

</td>

<td>

<asp:Button ID="sil" runat="server" Text="SİL" CssClass="btn btn-danger btn-fw" OnClick="sil\_Click" />

</td>

<td>

<asp:Button ID="guncelle" runat="server" Text="GÜNCELLE" CssClass="btn btn-warning btn-fw" OnClick="guncelle\_Click" />

</td>

</tr>

</table>

</div>

</div>

</div>

</form>

</asp:Content>

**Admin- Kullanıcı Düzenle (cs)**

public partial class Kullanici : System.Web.UI.Page

{

KullaniciGun kullanici = new KullaniciGun();

KullaniciLogin kullaniciLogin = new KullaniciLogin();

Sayfa yüklendiğinde kullanıcı yetkisi yok ise ana sayfaya yönlendirme yapılmaktadır. Yetki var ise “VeriGetir()” metodu çalıştırılarak veriler getirilir.

protected void Page\_Load(object sender, EventArgs e)

{

if (Session["Yetki"] == null)

{

Response.Redirect("../default.aspx");

}

if (!IsPostBack)

{

VeriGetir();

}

}

Bu metot sayesinde GridView’e KullaniciGun sınıfındaki “KullaniciGoster()” metodundan gelen veriler eklenir.

void VeriGetir()

{

GridView1.DataSource = kullanici.KullaniciGoster();

GridView1.DataBind();

}

Bu metot sayesinde bütün textler temizlenir ve seçili olan GridView sütununun seçimi kaldırılır.

void Temizle()

{

GridView1.SelectedIndex = -1;

idlbl.Text = "-1";

adtxt.Text = "";

soyadtxt.Text = "";

kaditxt.Text = "";

sifretxt.Text = "";

epostatxt.Text = "";

teltxt.Text = "";

yetkitxt.Text = "";

hatalbl.Text = "";

aratxt.Text = "";

}

GridView’de bir sütun seçildiğinde seçilen sütundaki veriler güncelleme işlemi yapılacak olan textlerde gösterilir.

protected void GridView1\_SelectedIndexChanged(object sender, EventArgs e)

{

hatalbl.Text = "";

idlbl.Text = GridView1.SelectedRow.Cells[1].Text;

adtxt.Text = HttpUtility.HtmlDecode(GridView1.SelectedRow.Cells[2].Text.ToString());

soyadtxt.Text = HttpUtility.HtmlDecode(GridView1.SelectedRow.Cells[3].Text.ToString());

kaditxt.Text = HttpUtility.HtmlDecode(GridView1.SelectedRow.Cells[4].Text.ToString());

sifretxt.Text = HttpUtility.HtmlDecode(GridView1.SelectedRow.Cells[5].Text.ToString());

epostatxt.Text = HttpUtility.HtmlDecode(GridView1.SelectedRow.Cells[6].Text.ToString());

teltxt.Text = HttpUtility.HtmlDecode(GridView1.SelectedRow.Cells[7].Text.ToString());

yetkitxt.Text = HttpUtility.HtmlDecode(GridView1.SelectedRow.Cells[8].Text.ToString());

}

GridView’de sayfalama işlemi yapmaktadır.

protected void GridView1\_PageIndexChanged(object sender, GridViewPageEventArgs e)

{

GridView1.PageIndex = e.NewPageIndex;

VeriGetir();

}

Bu butona tıklandığında “VeriGetir()” ve “Temizle()” metodu çalıştırılır.

protected void temizle\_Click(object sender, EventArgs e)

{

VeriGetir();

Temizle();

}

Bu butona tıklandığında restoran ekleme yapılırken öncelikle tüm alanların doldurulması beklenir. Tüm alanlar dolu ise kullanıcı adı KullaniciLogin sınıfındaki “KullaniciGetir2()” metoduna gönderilir. Geriye dönen veriler DataTable’a atılır. Eğer bu kullanıcı adına sahip kullanıcı var ise hata mesajı gösterilir. Yok ise tüm bilgiler KullaniciLogin sınıfındaki “KullaniciEkle()” metoduna gönderilir. Eğer geriye hata dönmezse kayıt yapıldığına dair mesaj gösterilir. İşlem bittikten sonra “VeriGetir()” metodu çalıştırılır.

protected void ekle\_Click(object sender, EventArgs e)

{

DataTable dt = new DataTable();

dt = kullaniciLogin.KullaniciGetir2(kaditxt.Text);

if (adtxt.Text == "" || soyadtxt.Text == "" || kaditxt.Text == "" || sifretxt.Text == "" || epostatxt.Text == "" || teltxt.Text == "")

{

hatalbl.CssClass = "text-danger";

hatalbl.Text = "Lütfen Tüm Alanları Doldurun";

}

else

{

if (dt.Rows.Count > 0)

{

hatalbl.CssClass = "text-info";

hatalbl.Text = "Kayıtlı Kullanıcı";

}

else

{

bool hata = kullaniciLogin.KullaniciEkle(adtxt.Text, soyadtxt.Text, kaditxt.Text, sifretxt.Text, epostatxt.Text, teltxt.Text);

if (hata == false)

{

hatalbl.CssClass = "text-success";

hatalbl.Text = "Kayıt Başarıyla Eklendi";

}

else

{

hatalbl.CssClass = "text-danger";

hatalbl.Text = "Kaydınız yapılamadı.";

}

}

}

VeriGetir();

}

Bu butona tıklandığında kullanıcı silme işlemi yapılırken öncelikle kullanıcı seçilmesi beklenir. Kullanıcı seçilmiş ise kullanıcı IDsi KullaniciGun sınıfındaki “KullaniciSil()” metoduna gönderilir. Geriye hata dönmez ise seçilen kayıt silinir. İşlem bittikten sonra “VeriGetir()” metodu çalıştırılır.

protected void sil\_Click(object sender, EventArgs e)

{

if (idlbl.Text == "-1")

{

hatalbl.CssClass = "text-danger";

hatalbl.Text = "Lütfen Silinecek Kaydı Seçin.";

}

else

{

bool hata = kullanici.KullaniciSil(Convert.ToInt32(idlbl.Text));

if (hata == false)

{

hatalbl.CssClass = "text-success";

hatalbl.Text = "Kayıt Başarıyla Silindi";

}

else

{

hatalbl.CssClass = "text-danger";

hatalbl.Text = "Kayıt Silinemedi.";

}

}

VeriGetir();

}

Bu butona tıklandığında kullanıcı güncelleme işlemi yapılırken öncelikle kullanıcı seçilmesi beklenir. Kullanıcı seçilmiş ise kullanıcı IDsi KullaniciGun sınıfındaki “KullaniciGuncelle()” metoduna veriler gönderilir. Geriye hata dönmez ise seçilen kayıt güncellenir. Daha sonra “VeriGetir()” metodu çalıştırılarak veriler getirilir.

protected void guncelle\_Click(object sender, EventArgs e)

{

if (idlbl.Text == "-1")

{

hatalbl.CssClass = "text-danger";

hatalbl.Text = "Lütfen Güncellenecek Kaydı Seçin.";

}

else

{

bool hata = kullanici.KullaniciGuncelle(Convert.ToInt32(idlbl.Text), adtxt.Text, soyadtxt.Text, kaditxt.Text, sifretxt.Text, epostatxt.Text, teltxt.Text,yetkitxt.Text);

if (hata == false)

{

hatalbl.CssClass = "text-success";

hatalbl.Text = "Kayıt Başarıyla Güncellendi";

}

else

{

hatalbl.CssClass = "text-danger";

hatalbl.Text = "Kayıt Güncellenemedi.";

}

}

VeriGetir();

}

Butona tıklandığında GridView’e KullaniciGun sınıfındaki “KullaniciAra()” metoduna gönderilen parametre ile geriye dönen veriler eklenir.

protected void ara\_Click(object sender, EventArgs e)

{

GridView1.DataSource = kullanici.KullaniciAra(aratxt.Text);

GridView1.DataBind();

}

}

|  |  |
| --- | --- |
| public DataTable KullaniciGetir2(string kullaniciAdi)  {  DataTable dt = new DataTable();  try  {  using (SqlConnection baglanti = b.BaglantiOlustur())  {  using (SqlCommand komut = new SqlCommand("kullanici\_getir\_usp2", baglanti))  {  komut.CommandType = CommandType.StoredProcedure;  komut.Parameters.AddWithValue("@kullaniciAdi", kullaniciAdi);  baglanti.Open();  SqlDataAdapter da;  DataSet ds = new DataSet();  da = new SqlDataAdapter(komut);  da.Fill(ds, "Kullanicilar");  dt = ds.Tables["Kullanicilar"];  baglanti.Close();  }  }  }  catch (Exception hata)  {    }  return dt;  } | Bu “KullaniciGetir2()” metodu sayesinde veri tabanından “kullanici\_getir \_usp2” prosedürü çalıştırılıp içerisine gönderilen parametreler ile Kullanıcı tablosundaki kullanıcının verilerini geri gönderir. |
| CREATE proc [dbo].[restoran\_getir\_usp2]  @isim varchar(70)  as  begin  select ID from Kullanicilar where isim=@isim  end | “kullanici\_getir \_usp2” prosedürü |

|  |  |
| --- | --- |
| public bool KullaniciEkle(string isim, string soyisim, string kullaniciAdi, string sifre,  string eposta, string telefon)  {  bool hataOlustuMu = false;  try  {  using (SqlConnection baglanti = b.BaglantiOlustur())  {  using (SqlCommand komut = new SqlCommand("kullanici\_ekle\_usp", baglanti))  {  komut.CommandType = CommandType.StoredProcedure;  komut.Parameters.AddWithValue("@isim", isim);  komut.Parameters.AddWithValue("@soyisim", soyisim);  komut.Parameters.AddWithValue("@kullaniciAdi", kullaniciAdi);  komut.Parameters.AddWithValue("@sifre", sifre);  komut.Parameters.AddWithValue("@eposta", eposta);  komut.Parameters.AddWithValue("@telefon", telefon);  baglanti.Open();  komut.ExecuteNonQuery();  baglanti.Close();  }  }  }  catch (Exception hata)  {  hataOlustuMu = true;  }  return hataOlustuMu;  } | Bu “KullaniciEkle()” metodu sayesinde veri tabanından “kullanici\_ekle \_usp” prosedürü çalıştırılıp içerisine gönderilen parametreler ile Kullanıcı tablosuna yeni kayıt eklenir. Veri eklenemezse geriye hata gönderilir. |
| CREATE proc [dbo].[kullanici\_ekle\_usp]  @isim varchar(50),  @soyisim varchar(50),  @kullaniciAdi varchar(50),  @sifre varchar(70),  @eposta varchar(70),  @telefon varchar(15)  as  insert into Kullanicilar  (isim,  soyisim,  kullaniciAdi,  sifre,  eposta,  telefon,  yetki)  values  (@isim,@soyisim,@kullaniciAdi,@sifre,@eposta,@telefon,1) | “kullanici\_ekle \_usp” prosedürü |

|  |  |
| --- | --- |
| public bool KullaniciSil(int ID)  {  bool hataOlustuMu = false;  try  {  using (SqlConnection baglanti = b.BaglantiOlustur())  {  using (SqlCommand komut = new SqlCommand("kullanici\_sil\_usp", baglanti))  {  komut.CommandType = CommandType.StoredProcedure;  komut.Parameters.AddWithValue("@ID", ID);  baglanti.Open();  komut.ExecuteNonQuery();  baglanti.Close();  }  }  }  catch (Exception hata)  {  hataOlustuMu = true;  }  return hataOlustuMu;  } | Bu “KullaniciSil()” metodu sayesinde veri tabanından “kullanici\_sil \_usp” prosedürü çalıştırılıp içerisine gönderilen parametre ile Kullanıcı tablosundaki kayıt silinir. Veri silinmezse geriye hata gönderilir. |
| CREATE proc [dbo].[kullanici\_sil\_usp]  @ID int  as  begin  DELETE FROM Kullanicilar WHERE ID=@ID  end | “kullanici\_sil \_usp” prosedürü |

|  |  |
| --- | --- |
| public bool KullaniciGuncelle(int ID, string isim, string soyisim, string kullaniciAdi, string sifre,  string eposta, string telefon, string yetki)  {  bool hataOlustuMu = false;  try  {  using (SqlConnection baglanti = b.BaglantiOlustur())  {  using (SqlCommand komut = new SqlCommand("kullanici\_guncelle\_usp", baglanti))  {  komut.CommandType = CommandType.StoredProcedure;  komut.Parameters.AddWithValue("@ID", ID);  komut.Parameters.AddWithValue("@isim", isim);  komut.Parameters.AddWithValue("@soyisim", soyisim);  komut.Parameters.AddWithValue("@kullaniciAdi", kullaniciAdi);  komut.Parameters.AddWithValue("@sifre", sifre);  komut.Parameters.AddWithValue("@eposta", eposta);  komut.Parameters.AddWithValue("@telefon", telefon);  komut.Parameters.AddWithValue("@yetki", yetki);  baglanti.Open();  komut.ExecuteNonQuery();  baglanti.Close();  }  }  }  catch (Exception hata)  {  hataOlustuMu = true;  }  return hataOlustuMu;  } | Bu “KullaniciGuncelle()” metodu sayesinde veri tabanından “kullanici\_guncelle \_usp” prosedürü çalıştırılıp içerisine gönderilen parametreler ile Kullanıcı tablosundaki kullanıcının verilerinde düzenleme yapılır. Eğer hata olursa geriye hata gönderir. |
| CREATE proc [dbo].[kullanici\_guncelle\_usp]  @ID int,  @isim varchar(50),  @soyisim varchar(50),  @kullaniciAdi varchar(50),  @sifre varchar(30),  @eposta varchar(70),  @telefon varchar(15),  @yetki tinyint  as  update Kullanicilar set  isim=@isim,  soyisim=@soyisim,  kullaniciAdi=@kullaniciAdi,  sifre=@sifre,  eposta=@eposta,  telefon=@telefon,  yetki=@yetki  where ID=@ID | “kullanici\_guncelle \_usp” prosedürü |

|  |  |
| --- | --- |
| public DataTable KullaniciAra(string parametre)  {  DataTable dt = new DataTable();  try  {  using (SqlConnection baglanti = b.BaglantiOlustur())  {  using (SqlCommand komut = new SqlCommand("kullanici\_ara\_usp", baglanti))  {  komut.CommandType = CommandType.StoredProcedure;  komut.Parameters.AddWithValue("@parametre", parametre);  baglanti.Open();  SqlDataAdapter da;  DataSet ds = new DataSet();  da = new SqlDataAdapter(komut);  da.Fill(ds, "Kullanicilar");  dt = ds.Tables["Kullanicilar"];  baglanti.Close();  }  }  }  catch (Exception hata)  {  }  return dt;  } | Bu “KullaniciAra()” metodu sayesinde veri tabanından “kullanici\_ara \_usp” prosedürü çalıştırılıp içerisine gönderilen parametreler ile Kullanıcı tablosundaki kullanıcının verilerini geri gönderir. |
| CREATE proc [dbo].[kullanici\_ara\_usp]  (  @parametre nvarchar(100)    )  as  begin  select \* from Kullanicilar where    isim like '%'+@parametre+'%' or  soyisim like '%'+@parametre+'%' or  kullaniciAdi like '%'+@parametre+'%' or  sifre like '%'+@parametre+'%' or  eposta like '%'+@parametre+'%' or  telefon like '%'+@parametre+'%'    end | “kullanici\_ara \_usp” prosedürü |

**REZERVET- Admin- Kullanıcı Mesajlar**



Bu bölümde Kullanıcı mesajları görüntülenmektedir.

<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">

<h3 class="page-title">Kullanıcı Mesajları </h3>

</asp:Content>

<asp:Content ID="Content3" ContentPlaceHolderID="ContentPlaceHolder2" runat="server">

<form runat="server">

<div class="col-lg-12 stretch-card">

<div class="card">

<div class="card-body">

<asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False" BackColor="White" BorderColor="#feebdc" BorderStyle="None" BorderWidth="1px" CellPadding="3" GridLines="Horizontal" AllowPaging="true" PageSize="5" Width="100%" OnSelectedIndexChanged="GridView1\_SelectedIndexChanged" OnPageIndexChanging="GridView1\_PageIndexChanged">

<PagerSettings Mode="NumericFirstLast" FirstPageText="İlk" LastPageText="Son" />

<AlternatingRowStyle BackColor="#F7F7F7" />

<Columns>

<asp:CommandField ShowSelectButton="True" HeaderText="Okundu Olarak İşaretle" ButtonType="Button" ControlStyle-BorderColor="#ff8913" SelectText=".">

<ControlStyle Height="40px" Width="50px" />

</asp:CommandField>

<asp:BoundField DataField="ID" HeaderText="ID" InsertVisible="False" ReadOnly="True" SortExpression="ID" />

<asp:BoundField DataField="isim" HeaderText="Ad" InsertVisible="False" ReadOnly="True" SortExpression="isim" />

<asp:BoundField DataField="soyisim" HeaderText="Soyad" InsertVisible="False" ReadOnly="True" SortExpression="soyisim" />

<asp:BoundField DataField="eposta" HeaderText="E Posta" InsertVisible="False" ReadOnly="True" SortExpression="eposta" />

<asp:BoundField DataField="konu" HeaderText="Konu" InsertVisible="False" ReadOnly="True" SortExpression="konu" />

<asp:BoundField DataField="ileti" HeaderText="İleti" InsertVisible="False" ReadOnly="True" SortExpression="ileti" />

</Columns>

<FooterStyle BackColor="#B5C7DE" ForeColor="#ff8913" />

<HeaderStyle BackColor="#ff8913" Font-Bold="True" ForeColor="#F7F7F7" />

<PagerStyle BackColor="#feebdc" ForeColor="#ff8913" HorizontalAlign="Right" />

<RowStyle BackColor="#feebdc" ForeColor="#ff8913" />

<SelectedRowStyle BackColor="#738A9C" Font-Bold="True" ForeColor="#F7F7F7" />

<SortedAscendingCellStyle BackColor="#F4F4FD" />

<SortedAscendingHeaderStyle BackColor="#5A4C9D" />

<SortedDescendingCellStyle BackColor="#D8D8F0" />

<SortedDescendingHeaderStyle BackColor="#3E3277" />

</asp:GridView>

</div>

</div>

</div>

</form>

</asp:Content>

**Admin- Kullanıcı Mesajlar (cs)**

public partial class Mesajlar : System.Web.UI.Page

{

MesajGun mesaj =new MesajGun();

Sayfa yüklendiğinde kullanıcı yetkisi yok ise ana sayfaya yönlendirme yapılmaktadır. Yetki var ise “VeriGetir()” metodu çalıştırılarak veriler getirilir.

protected void Page\_Load(object sender, EventArgs e)

{

if (Session["Yetki"] == null)

{

Response.Redirect("../default.aspx");

}

if (!IsPostBack)

{

VeriGetir();

}

}

GridView’de bir sütun seçildiğinde seçilen sütundaki ID verisi MesajGun sınıfındaki “MesajGuncelle()” metoduna veriler gönderilir. Geriye hata dönmez ise seçilen mesajın okundu bilgisi güncellenir. Daha sonra “VeriGetir()” metodu çalıştırılarak veriler getirilir.

protected void GridView1\_SelectedIndexChanged(object sender, EventArgs e)

{

mesaj.MesajGuncelle(Convert.ToInt32(GridView1.SelectedRow.Cells[1].Text));

VeriGetir();

GridView1.SelectedIndex = -1;

}

GridView’de sayfalama işlemi yapmaktadır.

protected void GridView1\_PageIndexChanged(object sender, GridViewPageEventArgs e)

{

GridView1.PageIndex = e.NewPageIndex;

VeriGetir();

}

Bu metot sayesinde GridView’e MesajGun sınıfındaki “MesajGoster()” metodundan gelen veriler eklenir.

void VeriGetir()

{

GridView1.DataSource = mesaj.MesajGoster();

GridView1.DataBind();

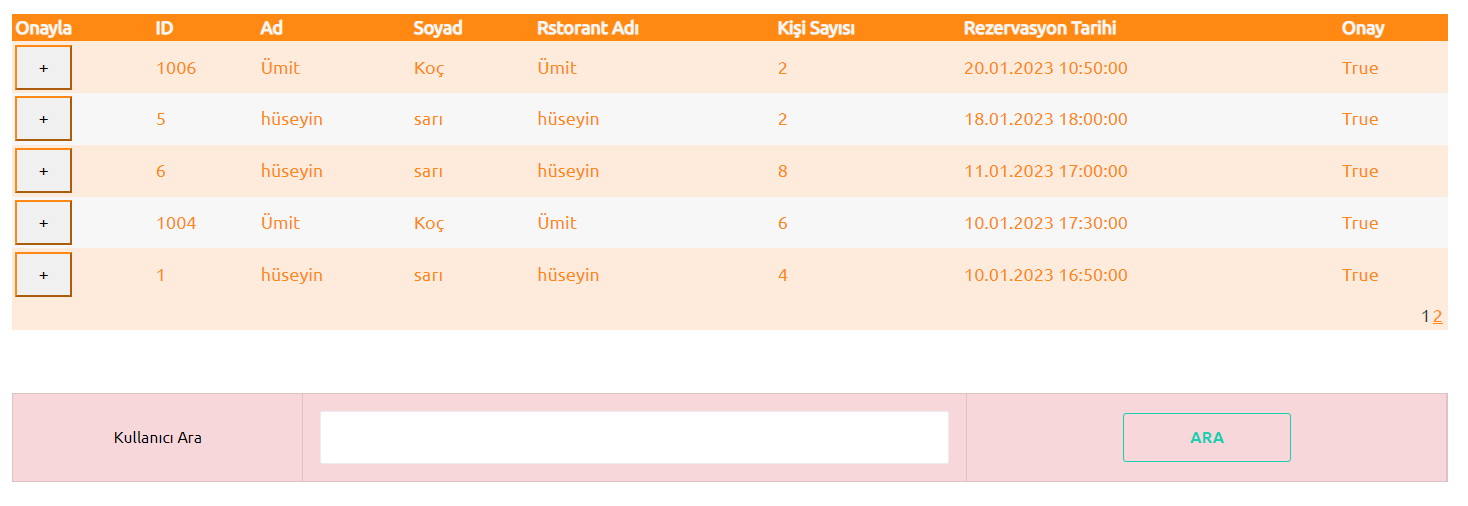
}

}

|  |  |
| --- | --- |
| public bool MesajGuncelle(int ID)  {  bool hataOlustuMu = false;  try  {  using (SqlConnection baglanti = b.BaglantiOlustur())  {  using (SqlCommand komut = new SqlCommand("mesaj\_guncelle\_usp", baglanti))  {  komut.CommandType = CommandType.StoredProcedure;  komut.Parameters.AddWithValue("@ID", ID);  baglanti.Open();  komut.ExecuteNonQuery();  baglanti.Close();  }  }  }  catch (Exception hata)  {  hataOlustuMu = true;  }  return hataOlustuMu;  } | Bu “MesajGuncelle()” metodu sayesinde veri tabanından “mesaj\_guncelle \_usp” prosedürü çalıştırılıp içerisine gönderilen mesaj IDsi ile Iletisim tablosundaki veriler güncellenir. Veri güncellenmezse geriye hata gönderilir. |
| CREATE proc [dbo].[mesaj\_guncelle\_usp]  @ID int  as  update Iletisim set  okundu=1  where ID=@ID | “mesaj\_guncelle \_usp” prosedürü |

|  |  |
| --- | --- |
| public DataTable MesajGoster()  {  DataTable dt = new DataTable();  try  {  using (SqlConnection baglanti = b.BaglantiOlustur())  {  using (SqlCommand komut = new SqlCommand("mesaj\_goster\_usp", baglanti))  {  komut.CommandType = CommandType.StoredProcedure;  baglanti.Open();  SqlDataAdapter da;  DataSet ds = new DataSet();  da = new SqlDataAdapter(komut);  da.Fill(ds, "Kullanici");  dt = ds.Tables["Kullanici"];  baglanti.Close();  }  }  }  catch (Exception hata)  {  }  return dt;  } | Bu “MesajGoster()” metodu sayesinde veri tabanından “mesaj\_goster\_ usp” prosedürü çalıştırılıp Iletisim tablosundaki okunmamış tüm mesajları geri gönderir. |
| CREATE proc [dbo].[mesaj\_goster\_usp]  as  begin  select ID,isim,soyisim,eposta,konu,ileti from Iletisim where okundu=0 order by ID desc  end | “mesaj\_goster\_ usp” prosedürü |

**REZERVET- Admin- Rezervasyon**



Bu bölümde Rezervasyonların onaylanma işlemi yapılmaktadır.

<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">

<h3 class="page-title">Rezervasyon Onay Sayfası </h3>

</asp:Content>

<asp:Content ID="Content3" ContentPlaceHolderID="ContentPlaceHolder2" runat="server">

<form runat="server">

<div class="col-lg-12 stretch-card">

<div class="card">

<div class="card-body">

<asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False" BackColor="White" BorderColor="#feebdc" BorderStyle="None" BorderWidth="1px" CellPadding="3" GridLines="Horizontal" AllowPaging="true" PageSize="5" Width="100%" OnSelectedIndexChanged="GridView1\_SelectedIndexChanged" OnPageIndexChanging="GridView1\_PageIndexChanged">

<PagerSettings Mode="NumericFirstLast" FirstPageText="İlk" LastPageText="Son" />

<AlternatingRowStyle BackColor="#F7F7F7" />

<Columns>

<asp:CommandField ShowSelectButton="True" HeaderText="Onayla" ButtonType="Button" ControlStyle-BorderColor="#ff8913" SelectText="+">

<ControlStyle Height="40px" Width="50px" />

</asp:CommandField>

<asp:BoundField DataField="ID" HeaderText="ID" InsertVisible="False" ReadOnly="True" SortExpression="ID" />

<asp:BoundField DataField="isim" HeaderText="Ad" InsertVisible="False" ReadOnly="True" SortExpression="isim" />

<asp:BoundField DataField="soyisim" HeaderText="Soyad" InsertVisible="False" ReadOnly="True" SortExpression="soyisim" />

<asp:BoundField DataField="isim" HeaderText="Rstorant Adı" InsertVisible="False" ReadOnly="True" SortExpression="isim" />

<asp:BoundField DataField="KisiSayisi" HeaderText="Kişi Sayısı" InsertVisible="False" ReadOnly="True" SortExpression="KisiSayisi" />

<asp:BoundField DataField="RezervasyonTarih" HeaderText="Rezervasyon Tarihi" InsertVisible="False" ReadOnly="True" SortExpression="RezervasyonTarih" />

<asp:BoundField DataField="Onay" HeaderText="Onay" InsertVisible="False" ReadOnly="True" SortExpression="Onay" />

</Columns>

<FooterStyle BackColor="#B5C7DE" ForeColor="#ff8913" />

<HeaderStyle BackColor="#ff8913" Font-Bold="True" ForeColor="#F7F7F7" />

<PagerStyle BackColor="#feebdc" ForeColor="#ff8913" HorizontalAlign="Right" />

<RowStyle BackColor="#feebdc" ForeColor="#ff8913" />

<SelectedRowStyle BackColor="#738A9C" Font-Bold="True" ForeColor="#F7F7F7" />

<SortedAscendingCellStyle BackColor="#F4F4FD" />

<SortedAscendingHeaderStyle BackColor="#5A4C9D" />

<SortedDescendingCellStyle BackColor="#D8D8F0" />

<SortedDescendingHeaderStyle BackColor="#3E3277" />

</asp:GridView>

<br />

<br />

<br />

<center>

<table class="table table-bordered">

<tbody>

<tr class="table-danger" align="center">

<td>Kullanıcı Ara

</td>

<td>

<asp:TextBox ID="aratxt" runat="server" CssClass="form-control"></asp:TextBox>

</td>

<td>

<asp:Button ID="ara" runat="server" Text="ARA" CssClass="btn btn-outline-success btn-fw" OnClick="ara\_Click" />

</td>

</tr>

</tbody>

</table>

</center>

<br />

<br />

</div>

</div>

</div>

</form>

</asp:Content>

**Admin- Rezervasyon (cs)**

public partial class Rezervasyon : System.Web.UI.Page

{

RezervasyonGun rezervasyon =new RezervasyonGun();

Sayfa yüklendiğinde kullanıcı yetkisi yok ise ana sayfaya yönlendirme yapılmaktadır. Yetki var ise “VeriGetir()” metodu çalıştırılarak veriler getirilir.

protected void Page\_Load(object sender, EventArgs e)

{

if (Session["Yetki"] == null)

{

Response.Redirect("../default.aspx");

}

if (!IsPostBack)

{

VeriGetir();

}

}

GridView’de bir sütun seçildiğinde seçilen sütundaki ID verisi RezervasyonGun sınıfındaki “RezervasyonGuncelle()” metoduna veriler gönderilir. Geriye hata dönmez ise seçilen rezervasyona onay verilir. Daha sonra “VeriGetir()” metodu çalıştırılarak veriler getirilir.

protected void GridView1\_SelectedIndexChanged(object sender, EventArgs e)

{

rezervasyon.RezervasyonGuncelle(Convert.ToInt32(GridView1.SelectedRow.Cells[1].Text));

VeriGetir();

Temizle();

}

GridView’de sayfalama işlemi yapmaktadır.

protected void GridView1\_PageIndexChanged(object sender, GridViewPageEventArgs e)

{

GridView1.PageIndex = e.NewPageIndex;

VeriGetir();

}

Bu metot sayesinde RezervasyonGun sınıfındaki “AdminRezervasyonGetir()” metodundan gelen veriler GridView’e eklenir.

void VeriGetir()

{

GridView1.DataSource = rezervasyon.AdminRezervasyonGetir();

GridView1.DataBind();

}

Bu metot sayesinde bütün textler temizlenir ve seçili olan GridView sütununun seçimi kaldırılır.

void Temizle()

{

GridView1.SelectedIndex = -1;

aratxt.Text = "";

}

Butona tıklandığında RezervasyonGun sınıfındaki “RezervasyonAra()” metoduna gönderilen parametre ile geriye dönen veriler GridView’e eklenir.

protected void ara\_Click(object sender, EventArgs e)

{

if (aratxt.Text!="")

{

GridView1.DataSource = rezervasyon.RezervasyonAra(aratxt.Text);

GridView1.DataBind();

}

else

{

VeriGetir();

}

}

}

|  |  |
| --- | --- |
| public DataTable AdminRezervasyonGetir()  {  DataTable dt = new DataTable();  try  {  using (SqlConnection baglanti = b.BaglantiOlustur())  {  using (SqlCommand komut = new SqlCommand("rezervasyon\_admin\_getir\_usp", baglanti))  {  komut.CommandType = CommandType.StoredProcedure;  baglanti.Open();  SqlDataAdapter da;  DataSet ds = new DataSet();  da = new SqlDataAdapter(komut);  da.Fill(ds, "Rezervasyon");  dt = ds.Tables["Rezervasyon"];  baglanti.Close();  }  }  }  catch (Exception hata)  {  }  return dt;  } | Bu “AdminRezervasyonGetir()” metodu sayesinde veri tabanından “rezervasyon\_admin\_getir \_usp” prosedürü çalıştırılıp içerisine gönderilen parametreler ile Kullanıcı tablosundaki kullanıcının verilerini geri gönderir. |
| CREATE proc [dbo].[rezervasyon\_admin\_getir\_usp]  as  begin  select rez.ID,k.isim,k.soyisim,r.resIsim,rez.KisiSayisi,rez.RezervasyonTarih,rez.Onay from Kullanicilar k,Restoranlar r, Rezervasyon rez where k.ID=rez.KullaniciID and r.ID=rez.RestorantID order by rez.RezervasyonTarih desc  end | “rezervasyon\_admin\_getir \_usp” prosedürü |

|  |  |
| --- | --- |
| public bool RezervasyonGuncelle(int ID)  {  bool hataOlustuMu = false;  try  {  using (SqlConnection baglanti = b.BaglantiOlustur())  {  using (SqlCommand komut = new SqlCommand("rezervasyon\_guncelle\_usp", baglanti))  {  komut.CommandType = CommandType.StoredProcedure;  komut.Parameters.AddWithValue("@ID", ID);  baglanti.Open();  komut.ExecuteNonQuery();  baglanti.Close();  }  }  }  catch (Exception hata)  {  hataOlustuMu = true;  }  return hataOlustuMu;  } | Bu “RezervasyonGuncelle()” metodu sayesinde veri tabanından “rezervasyon\_guncelle \_usp” prosedürü çalıştırılıp içerisine gönderilen ID ile Rezervasyon tablosundaki rezervasyon onaylanır. Eğer hata olursa geriye hata gönderir. |
| CREATE proc [dbo].[rezervasyon\_guncelle\_usp]  @ID int  as  update Rezervasyon set  Onay=1  where ID=@ID | “rezervasyon\_guncelle \_usp” prosedürü |

|  |  |
| --- | --- |
| public DataTable RezervasyonAra(string parametre)  {  DataTable dt = new DataTable();  try  {  using (SqlConnection baglanti = b.BaglantiOlustur())  {  using (SqlCommand komut = new SqlCommand("rezervasyon\_ara\_usp", baglanti))  {  komut.CommandType = CommandType.StoredProcedure;  komut.Parameters.AddWithValue("@parametre", parametre);  baglanti.Open();  SqlDataAdapter da;  DataSet ds = new DataSet();  da = new SqlDataAdapter(komut);  da.Fill(ds, "Kullanicilar");  dt = ds.Tables["Kullanicilar"];  baglanti.Close();  }  }  }  catch (Exception hata)  {  }  return dt;  } | Bu “RezervasyonAra()” metodu sayesinde veri tabanından “rezervasyon\_ara \_usp” prosedürü çalıştırılıp içerisine gönderilen parametreler ile Rezervasyon tablosundaki rezervasyonların verilerini geri gönderir. |
| CREATE proc [dbo].[rezervasyon\_ara\_usp]  (  @parametre nvarchar(100)    )  as  begin  select rez.ID,k.isim,k.soyisim,r.resIsim,rez.KisiSayisi,rez.RezervasyonTarih,rez.Onay from Kullanicilar k,Restoranlar r, Rezervasyon rez where k.ID=rez.KullaniciID and r.ID=rez.RestorantID and    k.isim like '%'+@parametre+'%' or  k.soyisim like '%'+@parametre+'%'    end | “rezervasyon\_ara \_usp” prosedürü |