# BOZOK ÜNİVERSİTESİ MÜHENDİSLİK FAKÜLTESİ

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**



**ONLİNE OTOPARK SİSTEMİ**

**TEKNİK DÖKÜMAN**

**SEMİHA ÜNEN REYYAN AYBÜKE DEMİRKAN**

**2019-2020 BAHAR DÖNEMİ**

MÜŞTERİ GİRİŞİ

LOGİN GİRİŞ KISMI KODLARI:

using System;

using System.Collections.Generic; using System.Linq;

using System.Web; using System.Web.UI;

using System.Web.UI.WebControls; using System.Data.SqlClient;

using System.Data; namespace mvcyönc

{

public partial class WebForm2 : System.Web.UI.Page

{

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

{

}

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

{

String mycon = "Data Source=DESKTOP-3ANJLN8;Initial Catalog=mvcmusteri;Integrated Security = True";

SqlConnection scon = new SqlConnection(mycon); String myquery = "select \* from logindetail"; SqlCommand cmd = new SqlCommand(); cmd.CommandText = myquery; cmd.Connection = scon;

SqlDataAdapter da = new SqlDataAdapter(); da.SelectCommand = cmd;

DataSet ds = new DataSet(); da.Fill(ds);

String uname; String pass;

uname = ds.Tables[0].Rows[0]["uname"].ToString(); pass = ds.Tables[0].Rows[0]["pass"].ToString(); scon.Close();

if (uname == TextBox1.Text && pass == TextBox2.Text)

{

Session["username"] = uname; Response.Redirect("Webform1.aspx");

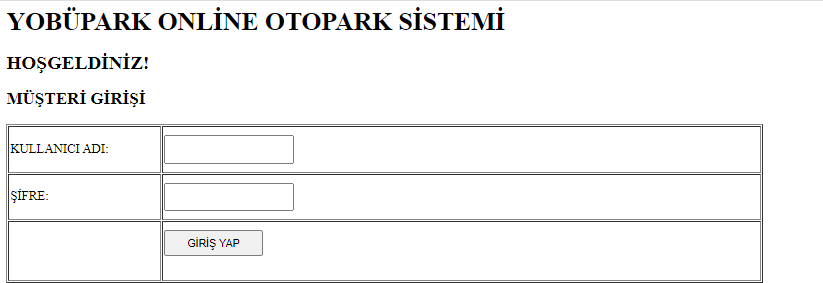
}

else

{

Label3.Text = "Geçersiz Kullanıcı Adı veya Şifre - Doğru Kullanıcı Adı Şifresi ile Yeniden Giriş Yapın";

}



}

}

}

REZERVASYON İŞLEMLERİ KODU:

using System;

using System.Collections.Generic; using System.Linq;

using System.Web; using System.Web.UI;

using System.Web.UI.WebControls; using System.Data;

using System.Data.SqlClient; namespace mvcyönc

{

public partial class WebForm1 : System.Web.UI.Page

{

static int seatbookno = -1;

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

{

if (!IsPostBack)

{

seatstatus();

}

}

protected void Button17\_Click(object sender, EventArgs e)

{

String mycon = "Data Source=DESKTOP-3ANJLN8; Initial Catalog=mvcmusteri; Integrated Security=True";

String updatedata = "Update SeatStatus set status='B' where seatno=" + seatbookno; SqlConnection con = new SqlConnection(mycon);

con.Open();

SqlCommand cmd = new SqlCommand(); cmd.CommandText = updatedata; cmd.Connection = con; cmd.ExecuteNonQuery();

Response.Redirect("Thanks.aspx?seatno=" + seatbookno);

}

private void seatstatus()

{

String mycon = "Data Source=DESKTOP-3ANJLN8; Initial Catalog=mvcmusteri; Integrated Security=True";

String myquery = "Select \* from SeatStatus"; SqlConnection con = new SqlConnection(mycon); SqlCommand cmd = new SqlCommand(); cmd.CommandText = myquery;

cmd.Connection = con;

SqlDataAdapter da = new SqlDataAdapter(); da.SelectCommand = cmd;

DataSet ds = new DataSet(); da.Fill(ds);

int rows = ds.Tables[0].Rows.Count; int i = 0;

while (i < rows)

{

String status;

status = ds.Tables[0].Rows[i]["status"].ToString(); if (status == "A")

{

if (i == 0 && seatbookno - 1 != i)

{

Button1.BackColor = System.Drawing.Color.Gray;

}

if (i == 1 && seatbookno - 1 != i)

{

Button2.BackColor = System.Drawing.Color.Gray;

}

if (i == 2 && seatbookno - 1 != i)

{

Button3.BackColor = System.Drawing.Color.Gray;

}

if (i == 3 && seatbookno - 1 != i)

{

Button4.BackColor = System.Drawing.Color.Gray;

}

if (i == 4 && seatbookno - 1 != i)

{

Button5.BackColor = System.Drawing.Color.Gray;

}

if (i == 5 && seatbookno - 1 != i)

{

Button6.BackColor = System.Drawing.Color.Gray;

}

if (i == 6 && seatbookno - 1 != i)

{

Button7.BackColor = System.Drawing.Color.Gray;

}

if (i == 7 && seatbookno - 1 != i)

{

Button8.BackColor = System.Drawing.Color.Gray;

}

if (i == 8 && seatbookno - 1 != i)

{

Button9.BackColor = System.Drawing.Color.Gray;

}

if (i == 9 && seatbookno - 1 != i)

{

Button10.BackColor = System.Drawing.Color.Gray;

}

if (i == 10 && seatbookno - 1 != i)

{

Button11.BackColor = System.Drawing.Color.Gray;

}

if (i == 11 && seatbookno - 1 != i)

{

Button12.BackColor = System.Drawing.Color.Gray;

}

if (i == 12 && seatbookno - 1 != i)

{

Button13.BackColor = System.Drawing.Color.Gray;

}

if (i == 13 && seatbookno - 1 != i)

{

Button14.BackColor = System.Drawing.Color.Gray;

}

if (i == 14 && seatbookno - 1 != i)

{

Button15.BackColor = System.Drawing.Color.Gray;

}

if (i == 15 && seatbookno - 1 != i)

{

Button16.BackColor = System.Drawing.Color.Gray;

}

}

if (status == "B")

{

if (i == 0)

{

Button1.BackColor = System.Drawing.Color.Red; Button1.Enabled = false;

}

if (i == 1)

{

Button2.BackColor = System.Drawing.Color.Red; Button2.Enabled = false;

}

if (i == 2)

{

Button3.BackColor = System.Drawing.Color.Red; Button3.Enabled = false;

}

if (i == 3)

{

Button4.BackColor = System.Drawing.Color.Red;

Button4.Enabled = false;

}

if (i == 4)

{

Button5.BackColor = System.Drawing.Color.Red; Button5.Enabled = false;

}

if (i == 5)

{

Button6.BackColor = System.Drawing.Color.Red; Button6.Enabled = false;

}

if (i == 6)

{

Button7.BackColor = System.Drawing.Color.Red; Button7.Enabled = false;

}

if (i == 7)

{

Button8.BackColor = System.Drawing.Color.Red; Button8.Enabled = false;

}

if (i == 8)

{

Button9.BackColor = System.Drawing.Color.Red; Button9.Enabled = false;

}

if (i == 9)

{

Button10.BackColor = System.Drawing.Color.Red; Button10.Enabled = false;

}

if (i == 10)

{

Button11.BackColor = System.Drawing.Color.Red; Button11.Enabled = false;

}

if (i == 11)

{

Button12.BackColor = System.Drawing.Color.Red; Button12.Enabled = false;

}

if (i == 12)

{

Button13.BackColor = System.Drawing.Color.Red; Button13.Enabled = false;

}

if (i == 13)

{

Button14.BackColor = System.Drawing.Color.Red; Button14.Enabled = false;

}

if (i == 14)

{

Button15.BackColor = System.Drawing.Color.Red; Button15.Enabled = false;

}

if (i == 15)

{

Button16.BackColor = System.Drawing.Color.Red; Button16.Enabled = false;

}

}

i = i + 1;

}

}

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

{

seatbookno = 1;

Button1.BackColor = System.Drawing.Color.Green; seatstatus();

}

protected void Button2\_Click(object sender, EventArgs e)

{

seatbookno = 2;

Button2.BackColor = System.Drawing.Color.Green; seatstatus();

}

protected void Button3\_Click(object sender, EventArgs e)

{

seatbookno = 3;

Button3.BackColor = System.Drawing.Color.Green; seatstatus();

}

protected void Button4\_Click(object sender, EventArgs e)

{

seatbookno = 4;

Button4.BackColor = System.Drawing.Color.Green; seatstatus();

}

protected void Button5\_Click(object sender, EventArgs e)

{

seatbookno = 5;

Button5.BackColor = System.Drawing.Color.Green; seatstatus();

}

protected void Button6\_Click(object sender, EventArgs e)

{

seatbookno = 6;

Button6.BackColor = System.Drawing.Color.Green; seatstatus();

}

protected void Button7\_Click(object sender, EventArgs e)

{

seatbookno = 7;

Button7.BackColor = System.Drawing.Color.Green; seatstatus();

}

protected void Button8\_Click(object sender, EventArgs e)

{

seatbookno = 8;

Button8.BackColor = System.Drawing.Color.Green; seatstatus();

}

protected void Button9\_Click(object sender, EventArgs e)

{

seatbookno = 9;

Button9.BackColor = System.Drawing.Color.Green; seatstatus();

}

protected void Button10\_Click(object sender, EventArgs e)

{

seatbookno = 10;

Button10.BackColor = System.Drawing.Color.Green; seatstatus();

}

protected void Button11\_Click(object sender, EventArgs e)

{

seatbookno = 11;

Button11.BackColor = System.Drawing.Color.Green; seatstatus();

}

protected void Button12\_Click(object sender, EventArgs e)

{

seatbookno = 12;

Button12.BackColor = System.Drawing.Color.Green; seatstatus();

}

protected void Button13\_Click(object sender, EventArgs e)

{

seatbookno = 13;

Button13.BackColor = System.Drawing.Color.Green; seatstatus();

}

protected void Button14\_Click(object sender, EventArgs e)

{

seatbookno = 14;

Button14.BackColor = System.Drawing.Color.Green; seatstatus();

}

protected void Button15\_Click(object sender, EventArgs e)

{

seatbookno = 15;

Button15.BackColor = System.Drawing.Color.Green; seatstatus();

}

protected void Button16\_Click(object sender, EventArgs e)

{

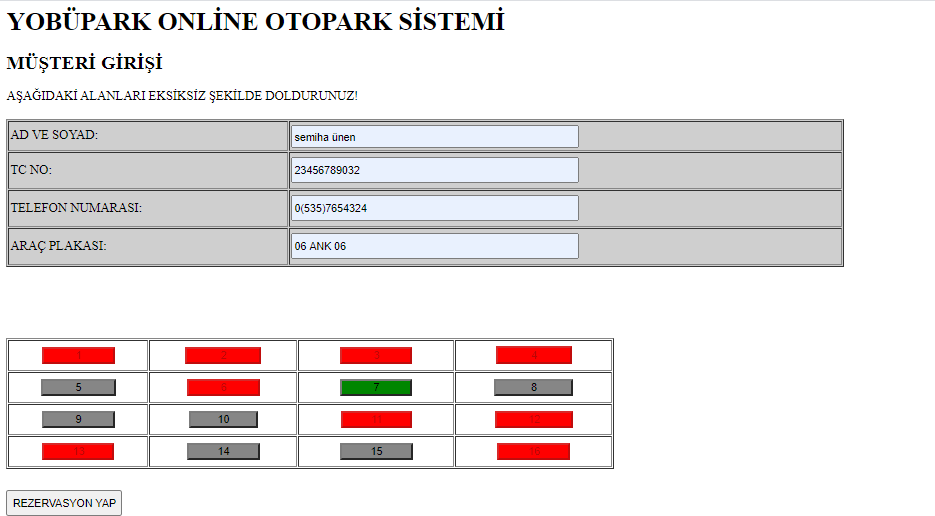
seatbookno = 16;

Button16.BackColor = System.Drawing.Color.Green; seatstatus();

}

}

}



REZERVASYON YAPTIĞI YERİN GÖSTERİMİ:

using System;

using System.Collections.Generic; using System.Linq;

using System.Web; using System.Web.UI;

using System.Web.UI.WebControls;

namespace mvcyönc

{

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

{

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

{

Label1.Text ="SEÇTİĞİNİZ PARK YERİ: " + Request.QueryString["Seatno"];

}

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

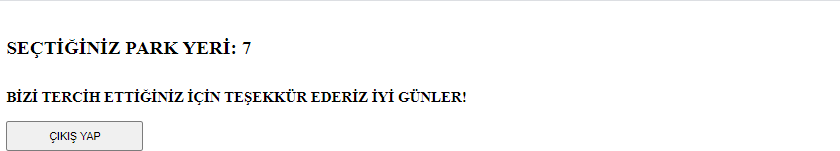
{

Session.Abandon(); Response.Redirect("WebForm2.aspx");

}

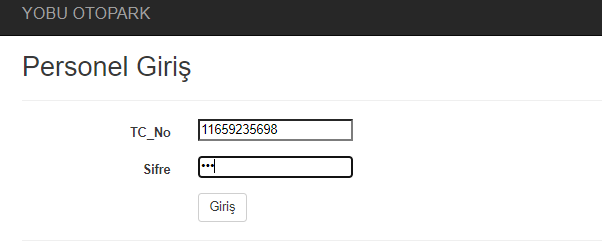
}

}



PERSONEL GİRİŞ İŞLEMLERİ KODLARI:

Personel Giriş Sayfası: Personel TC kimlik numarası ve şifresi ile personel ana sayfasına giriş yapabilmektedir. Şifreleme kısmı için if-else deyimi içerisinde veri tabanı bağlantısı ile kullanıldı hatalı giriş yapılığında uyarı vermektedir.



using System;

using Garaj\_Otomasyonu.Models;

using Garaj\_Otomasyonu.Models.Entity; using System.Collections.Generic;

using System.Linq; using System.Web; using System.Web.Mvc;

using System.Web.Security;

namespace Garaj\_Otomasyonu.Controllers

{

public class PersonelGirişController : Controller

{

Otopark\_OtomasyonuEntities db = new Otopark\_OtomasyonuEntities();

//private object FormsAuthentication;

// GET: PersonelGiriş public ActionResult Login()

{

return View();

}

[HttpPost]

public ActionResult Login(Personel\_Giriş Personel)

{

var personel = db.Personel\_Giriş.FirstOrDefault(x => x.TC\_No == Personel.TC\_No && x.Sifre == Personel.Sifre);

if (personel != null)

{

FormsAuthentication.SetAuthCookie(personel.TC\_No, false); return RedirectToAction("Index", "PersonelGiriş");

}

else

{

ViewBag.Mesaj = "Geçersiz Kullanıcı Adı veya Şifre"; return View();

}

}

public ActionResult Logout()

{

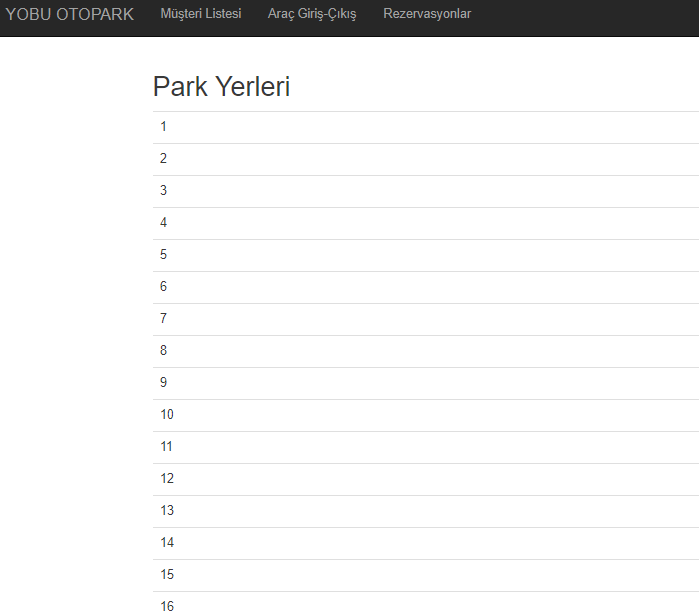
FormsAuthentication.SignOut(); return View("Login");

}

}

}

Personel Ana Sayfası: Sayfalar oluşturuldu arasındaki geçişler sağlandı.



using System;

using System.Collections.Generic; using System.Linq;

using System.Data.Entity; using System.Web;

using System.Web.Mvc;

using Garaj\_Otomasyonu.Models.Entity;

namespace Garaj\_Otomasyonu.Controllers

{

public class PersonelAnasayfaController : Controller

{

private Otopark\_OtomasyonuEntities db = new Otopark\_OtomasyonuEntities();

// GET: PersonelAnasayfa public ActionResult Index()

{

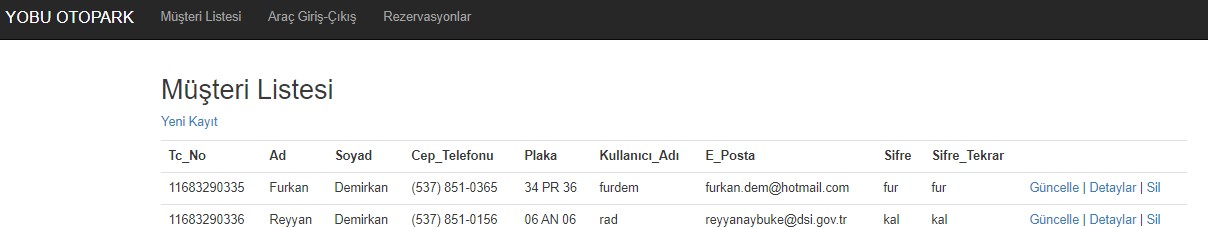
return View(db.Parkdurumu.ToList());

}

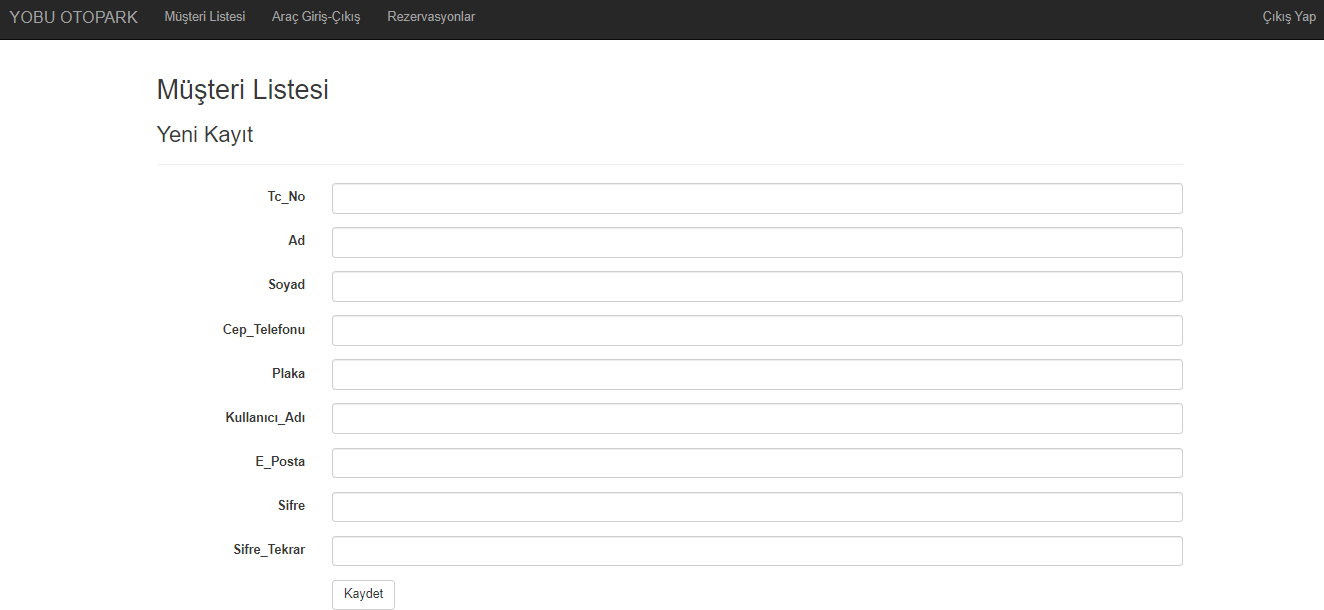
}

}

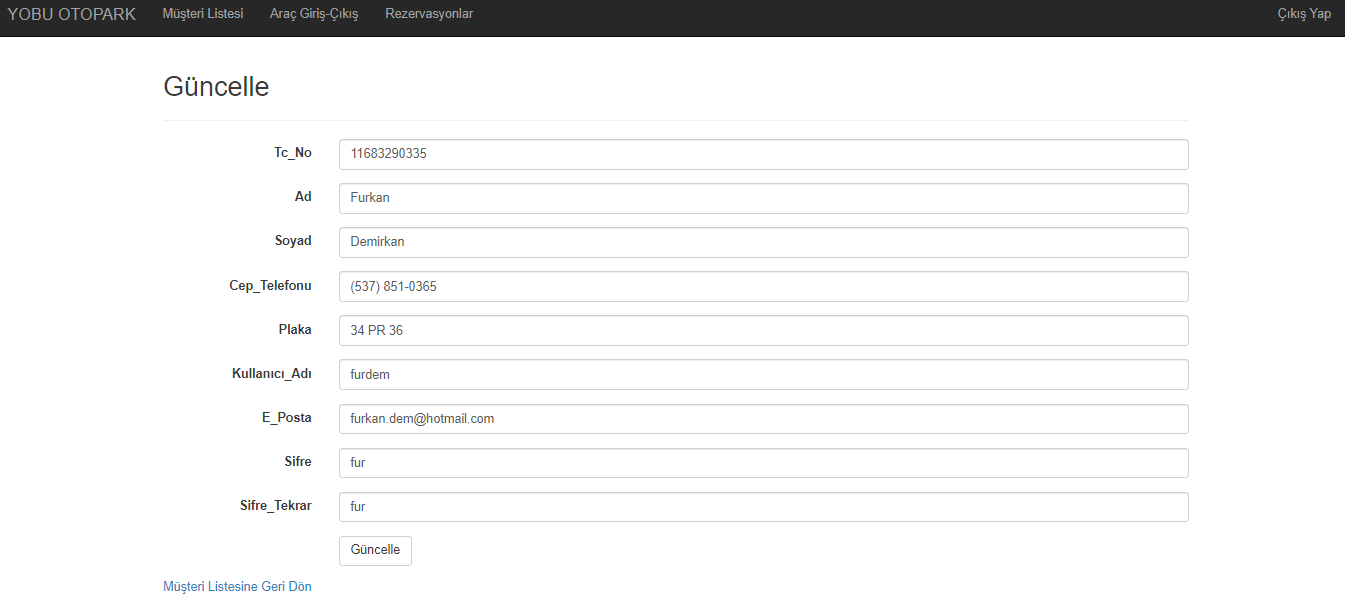
Müşteri Listesi: Otoparkta kayıtlı olan müşterileri bilgileri ile gösterir.



Müşteri Ekle (Kayıt): Otoparka gelen müşteriler kayıtlı değil ise buradan kaydedilir.



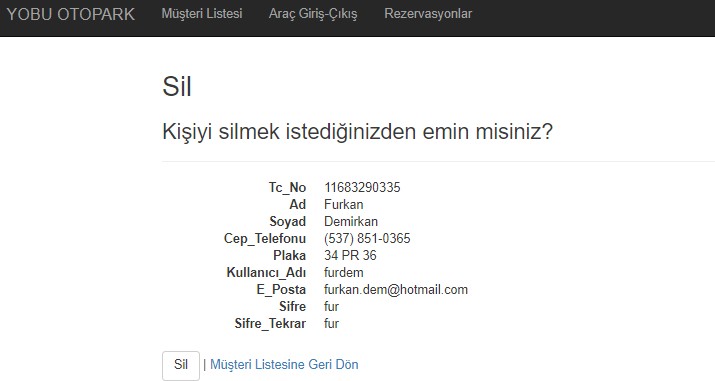
Müşteri Güncelle: Otoparktaki kayıtlı müşteriyi günceller.



Müşteri Detay: Otoparktaki kayıtlı bir müşterinin bilgilerini gösterir.



Müşteri Sil: Otoparktaki kayıtlı müşteriyi siler.



using System;

using System.Collections.Generic; using System.Data;

using System.Data.Entity; using System.Linq;

using System.Net; using System.Web;

using System.Web.Mvc;

using Garaj\_Otomasyonu.Models.Entity;

namespace Garaj\_Otomasyonu.Controllers

{

public class MusteriListesiController : Controller

{

private Otopark\_OtomasyonuEntities db = new Otopark\_OtomasyonuEntities();

// GET: MusteriListesi public ActionResult Index()

{

return View(db.Musteri\_Listesi.ToList());

}

// GET: MusteriListesi/Details/5 public ActionResult Details(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Musteri\_Listesi musteri\_Listesi = db.Musteri\_Listesi.Find(id); if (musteri\_Listesi == null)

{

return HttpNotFound();

}

return View(musteri\_Listesi);

}

// GET: MusteriListesi/Create public ActionResult Create()

{

return View();

}

// POST: MusteriListesi/Create

// To protect from overposting attacks, please enable the specific properties you want to bind to, for

// more details see [http://go.microsoft.com/fwlink/?LinkId=317598.](http://go.microsoft.com/fwlink/?LinkId=317598) [HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Create([Bind(Include = "id,Tc\_No,Ad,Soyad,Cep\_Telefonu,Plaka,Kullanıcı\_Adı,E\_Posta,Sifre,Sifre\_Tekrar")] Musteri\_Listesi musteri\_Listesi)

{

if (ModelState.IsValid)

{

db.Musteri\_Listesi.Add(musteri\_Listesi); db.SaveChanges();

return RedirectToAction("Index");

}

return View(musteri\_Listesi);

}

// GET: MusteriListesi/Edit/5 public ActionResult Edit(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Musteri\_Listesi musteri\_Listesi = db.Musteri\_Listesi.Find(id); if (musteri\_Listesi == null)

{

return HttpNotFound();

}

return View(musteri\_Listesi);

}

// POST: MusteriListesi/Edit/5

// To protect from overposting attacks, please enable the specific properties you want to bind to, for

// more details see [http://go.microsoft.com/fwlink/?LinkId=317598.](http://go.microsoft.com/fwlink/?LinkId=317598) [HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Edit([Bind(Include = "id,Tc\_No,Ad,Soyad,Cep\_Telefonu,Plaka,Kullanıcı\_Adı,E\_Posta,Sifre,Sifre\_Tekrar")] Musteri\_Listesi musteri\_Listesi)

{

if (ModelState.IsValid)

{

db.Entry(musteri\_Listesi).State = EntityState.Modified; db.SaveChanges();

return RedirectToAction("Index");

}

return View(musteri\_Listesi);

}

// GET: MusteriListesi/Delete/5 public ActionResult Delete(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Musteri\_Listesi musteri\_Listesi = db.Musteri\_Listesi.Find(id); if (musteri\_Listesi == null)

{

return HttpNotFound();

}

return View(musteri\_Listesi);

}

// POST: MusteriListesi/Delete/5 [HttpPost, ActionName("Delete")] [ValidateAntiForgeryToken]

public ActionResult DeleteConfirmed(int id)

{

Musteri\_Listesi musteri\_Listesi = db.Musteri\_Listesi.Find(id); db.Musteri\_Listesi.Remove(musteri\_Listesi); db.SaveChanges();

return RedirectToAction("Index");

}

protected override void Dispose(bool disposing)

{

if (disposing)

{

db.Dispose();

}

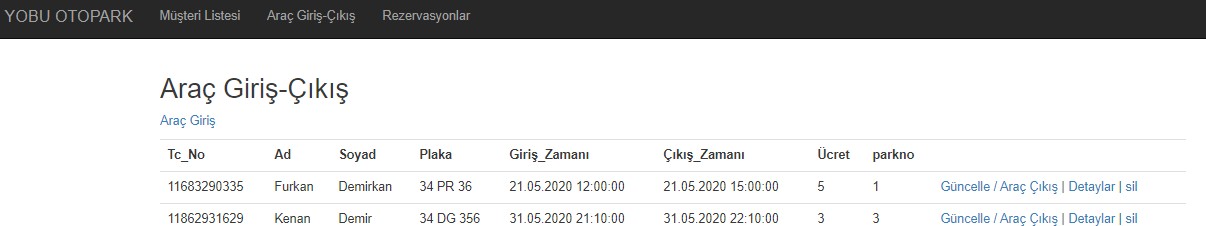
base.Dispose(disposing);

}

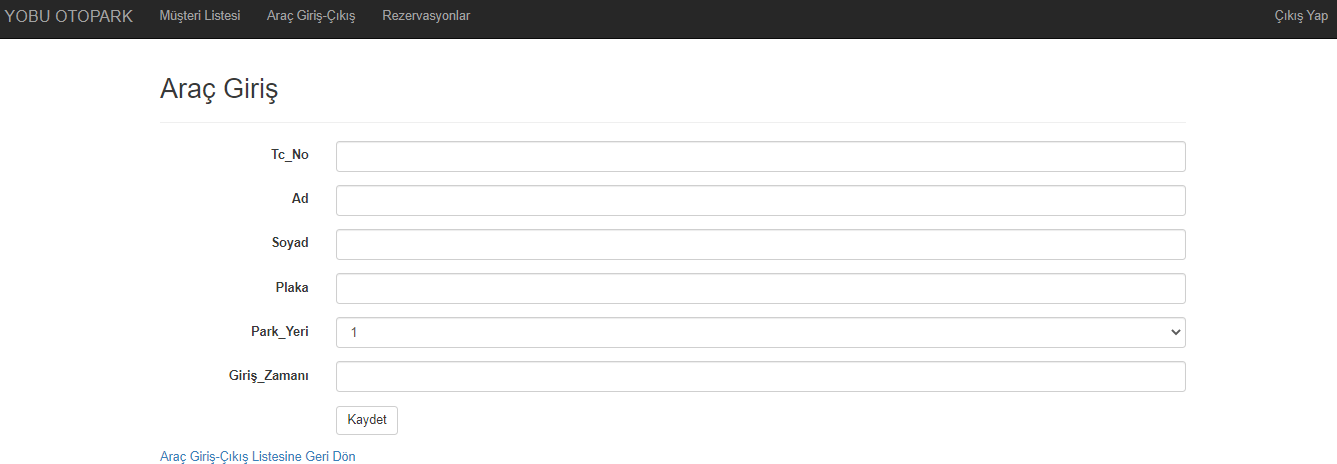
}

}

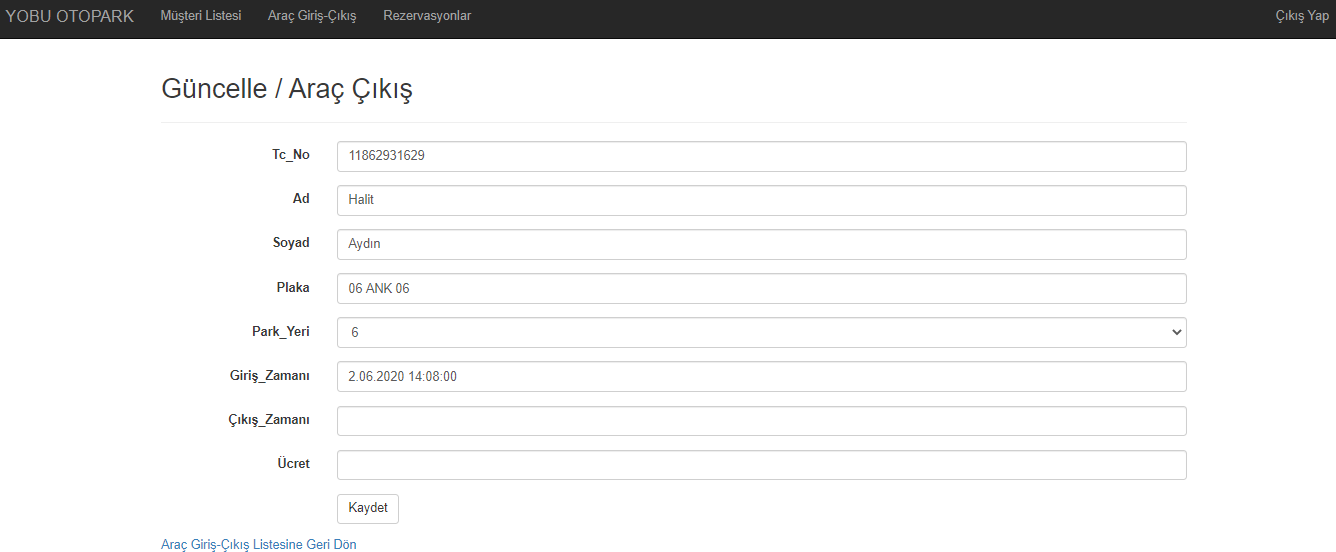
Araç giriş/çıkış: Otoparka gelen aracın girişi ve çıkışı listesi burada tutulur.



Araç giriş: Otoparka gelen aracın girişi buradan yapılır.



Araç Güncelleme/Çıkış: Otoparka gelen aracın güncellemesi ve çıkışı buradan yapılır.



Araç Detay: Otoparktaki giriş yapmış veya çıkış yapmış bir aracın bilgilerini gösterir.



Araç Sil: Otoparktaki giriş yapmış veya çıkış yapmış bir aracı siler.



using System;

using System.Collections.Generic; using System.Data;

using System.Data.Entity; using System.Linq;

using System.Net; using System.Web;

using System.Web.Mvc;

using Garaj\_Otomasyonu.Models.Entity;

namespace Garaj\_Otomasyonu.Controllers

{

public class Araç\_Giriş\_ÇıkışController : Controller

{

private Otopark\_OtomasyonuEntities db = new Otopark\_OtomasyonuEntities();

// GET: Araç\_Giriş\_Çıkış public ActionResult Index()

{

var park\_Kayıt = db.Park\_Kayıt.Include(p => p.Parkdurumu); return View(park\_Kayıt.ToList());

}

// GET: Araç\_Giriş\_Çıkış/Details/5 public ActionResult Details(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Park\_Kayıt park\_Kayıt = db.Park\_Kayıt.Find(id); if (park\_Kayıt == null)

{

return HttpNotFound();

}

return View(park\_Kayıt);

}

// GET: Araç\_Giriş\_Çıkış/Create public ActionResult Create()

{

ViewBag.Park\_Yeri = new SelectList(db.Parkdurumu, "id", "parkno"); return View();

}

// POST: Araç\_Giriş\_Çıkış/Create

// To protect from overposting attacks, please enable the specific properties you want to bind to, for

// more details see [http://go.microsoft.com/fwlink/?LinkId=317598.](http://go.microsoft.com/fwlink/?LinkId=317598) [HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Create([Bind(Include = "id,Tc\_No,Ad,Soyad,Plaka,Park\_Yeri,Giriş\_Zamanı,Çıkış\_Zamanı,Ücret")] Park\_Kayıt park\_Kayıt)

{

if (ModelState.IsValid)

{

db.Park\_Kayıt.Add(park\_Kayıt); db.SaveChanges();

return RedirectToAction("Index");

}

ViewBag.Park\_Yeri = new SelectList(db.Parkdurumu, "id", "parkno", park\_Kayıt.Park\_Yeri);

return View(park\_Kayıt);

}

// GET: Araç\_Giriş\_Çıkış/Edit/5 public ActionResult Edit(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Park\_Kayıt park\_Kayıt = db.Park\_Kayıt.Find(id); if (park\_Kayıt == null)

{

return HttpNotFound();

}

ViewBag.Park\_Yeri = new SelectList(db.Parkdurumu, "id", "parkno", park\_Kayıt.Park\_Yeri);

return View(park\_Kayıt);

}

// POST: Araç\_Giriş\_Çıkış/Edit/5

// To protect from overposting attacks, please enable the specific properties you want to bind to, for

// more details see [http://go.microsoft.com/fwlink/?LinkId=317598.](http://go.microsoft.com/fwlink/?LinkId=317598) [HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Edit([Bind(Include = "id,Tc\_No,Ad,Soyad,Plaka,Park\_Yeri,Giriş\_Zamanı,Çıkış\_Zamanı,Ücret")] Park\_Kayıt park\_Kayıt)

{

if (ModelState.IsValid)

{

db.Entry(park\_Kayıt).State = EntityState.Modified; db.SaveChanges();

return RedirectToAction("Index");

}

ViewBag.Park\_Yeri = new SelectList(db.Parkdurumu, "id", "parkno", park\_Kayıt.Park\_Yeri);

return View(park\_Kayıt);

}

// GET: Araç\_Giriş\_Çıkış/Delete/5 public ActionResult Delete(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Park\_Kayıt park\_Kayıt = db.Park\_Kayıt.Find(id); if (park\_Kayıt == null)

{

return HttpNotFound();

}

return View(park\_Kayıt);

}

// POST: Araç\_Giriş\_Çıkış/Delete/5 [HttpPost, ActionName("Delete")] [ValidateAntiForgeryToken]

public ActionResult DeleteConfirmed(int id)

{

Park\_Kayıt park\_Kayıt = db.Park\_Kayıt.Find(id);

db.Park\_Kayıt.Remove(park\_Kayıt); db.SaveChanges();

return RedirectToAction("Index");

}

protected override void Dispose(bool disposing)

{

if (disposing)

{

db.Dispose();

}

base.Dispose(disposing);

}

}

}

# YÖNETİCİ GİRİŞ İŞLEMLERİ KODLARI YÖNETİCİ GİRİŞ İŞLEMLERİ:

using System;

using System.Collections.Generic; using System.Linq;

using System.Web; using System.Web.Mvc;

using mvcbir.Models.Entity; using System.Web.Security; namespace mvcbir.Controllers

{

public class GÜVENLİKController : Controller

{

// GET: GÜVENLİK

mvcporEntities db = new mvcporEntities();

public ActionResult GİRİSYAP()

{

return View();

}

[HttpPost]

public ActionResult GİRİSYAP(YÖNETİCİ pr)

{

var bilgiler = db.YÖNETİCİ.FirstOrDefault(x => x.yöneticiad == pr.yöneticiad && x.yöneticisifre == pr.yöneticisifre);

if (bilgiler != null)

{

FormsAuthentication.SetAuthCookie(bilgiler.yöneticiad, false); return RedirectToAction("Index", "PERSONEL");

}

else

{

return View();

}

}

public ActionResult CIKISYAP()

{

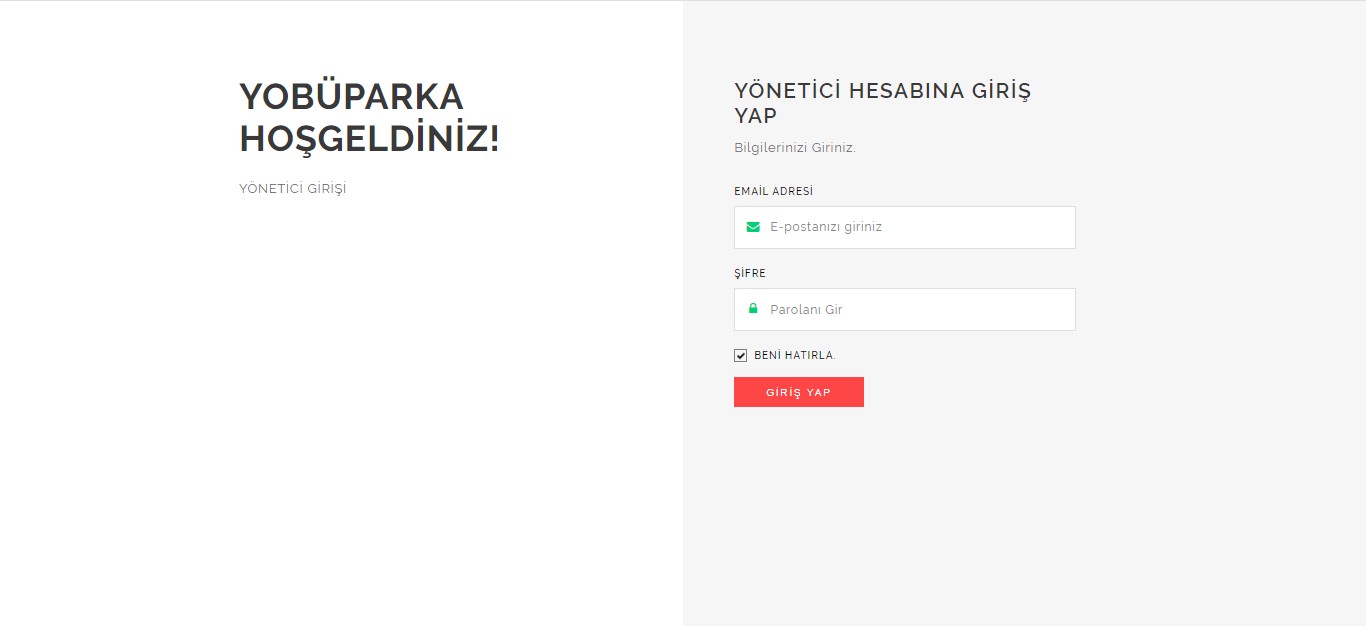
FormsAuthentication.SignOut();

return RedirectToAction("GİRİSYAP");

}

}

}



# PERSONELLERİN LİSTELENDİĞİ KISIM VE PERSONEL EKLE/SİL/GÜNCELLE KISIMLARI:

using System;

using System.Collections.Generic; using System.Linq;

using System.Web; using System.Web.Mvc;

using mvcbir.Models.Entity; namespace mvcbir.Controllers

{

public class PERSONELController : Controller

{

// GET: PERSONEL

mvcporEntities db = new mvcporEntities(); [Authorize]

public ActionResult Index()

{

var degerler = db.PERSONEL.ToList(); return View(degerler);

}

[HttpGet]

public ActionResult YENİPERSONEL()

{

return View();

}

[HttpPost]

public ActionResult YENİPERSONEL(PERSONEL p2)

{

db.PERSONEL.Add(p2);

db.SaveChanges(); return View();

}

public ActionResult SİL(int id)

{

var personel = db.PERSONEL.Find(id); db.PERSONEL.Remove(personel); db.SaveChanges();

return RedirectToAction("Index");

}

public ActionResult PERSONELGETİR(int id)

{

var perso = db.PERSONEL.Find(id); return View("PERSONELGETİR", perso);

}

public ActionResult GÜNCELLE(PERSONEL p1)

{

var prs = db.PERSONEL.Find(p1.personelid); prs.personelad = p1.personelad; prs.personelsoyad = p1.personelsoyad; db.SaveChanges();

return RedirectToAction("Index");

}

}

}



# PERSONEL EKLEME SAYFASI:



PERSONEL GÜNCELLEME SAYFASI:



# PARK BİLGİLERİNİ GÖSTERİLDİĞİ KISIM:

namespace mvcbir.Controllers

{

public class PARKBILGISIController : Controller

{

mvcporEntities db = new mvcporEntities();

// GET: PARKBILGISI

public ActionResult Index()

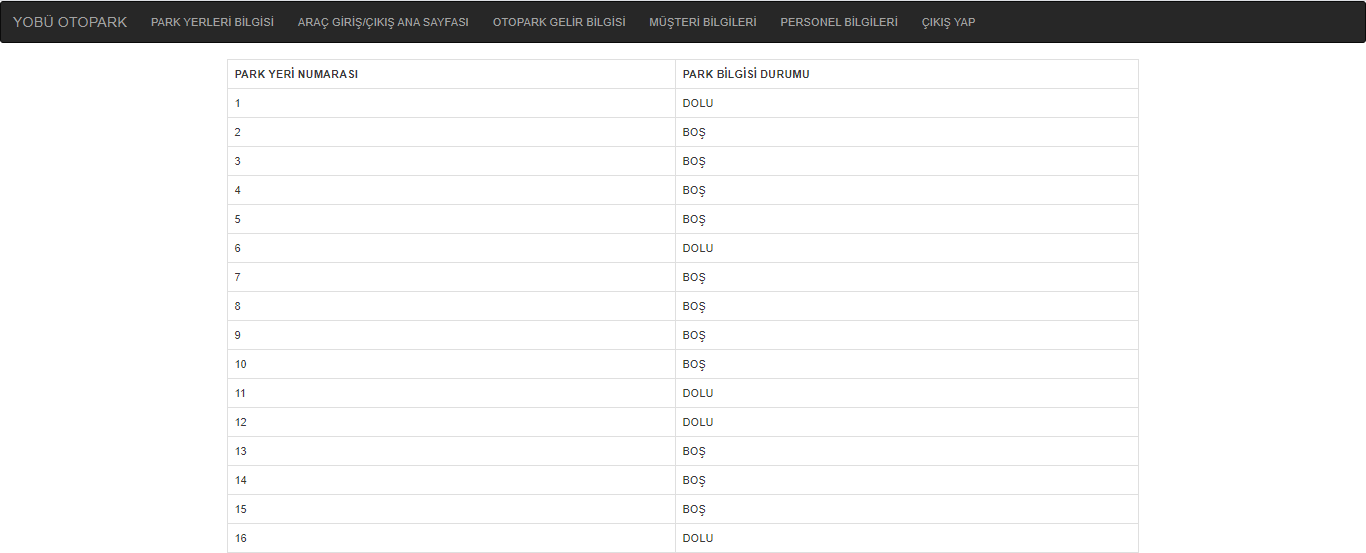
{

var degerler = db.PARKBILGISI.ToList(); return View(degerler);

}

}

}



# ARAÇ/GİRİŞ İŞLEMLERİNİN LİSTELENDİĞİ KISIM:

using System;

using System.Collections.Generic; using System.Linq;

using System.Web; using System.Web.Mvc;

using mvcbir.Models.Entity; namespace mvcbir.Controllers

{

public class ARACGİRCIKController : Controller

{

// GET: ARACGİRCIK

mvcporEntities db = new mvcporEntities();

public ActionResult Index()

{

var degerler = db.ARACGİRCIK.ToList();

return View(degerler);

}

}

}



# ÜCRET BİLGİLERİNİN GÖSTERİLDİĞİ KISIM:

using System;

using System.Collections.Generic; using System.Linq;

using System.Web; using System.Web.Mvc;

using mvcbir.Models.Entity; namespace mvcbir.Controllers

{

public class ÜCRETController : Controller

{

// GET: ÜCRET

mvcporEntities db = new mvcporEntities();

public ActionResult Index()

{

var degerler = db.ÜCRET.ToList(); return View(degerler);

}

}

}



# MÜŞTERİ BİLGİLERİNİN LİSTELENDİĞİ KISIM:

using System;

using System.Collections.Generic; using System.Linq;

using System.Web; using System.Web.Mvc;

using mvcbir.Models.Entity;

namespace mvcbir.Controllers

{

public class MÜSTERIController : Controller

{

// GET: MÜSTERI

mvcporEntities db = new mvcporEntities();

public ActionResult Index()

{

var degerler = db.MÜSTERI.ToList(); return View(degerler);

}

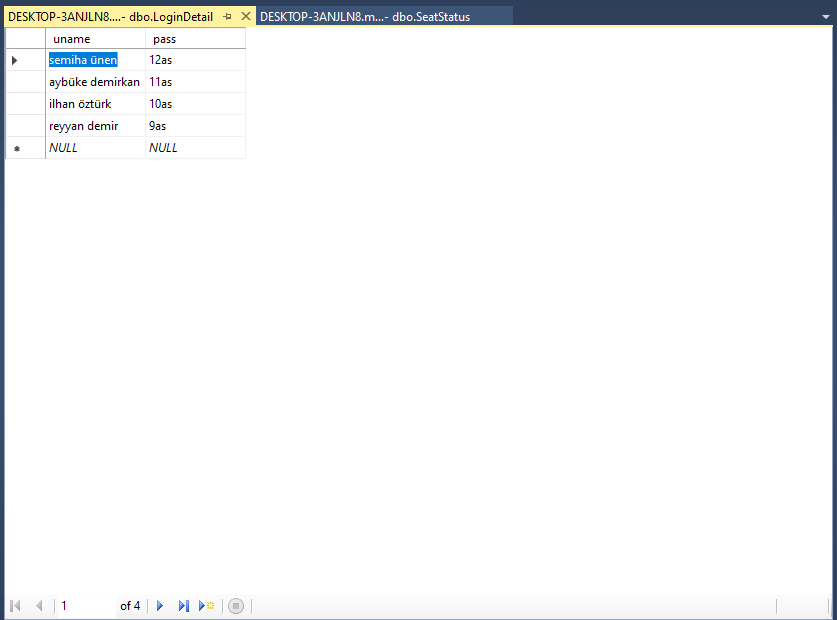
}

}

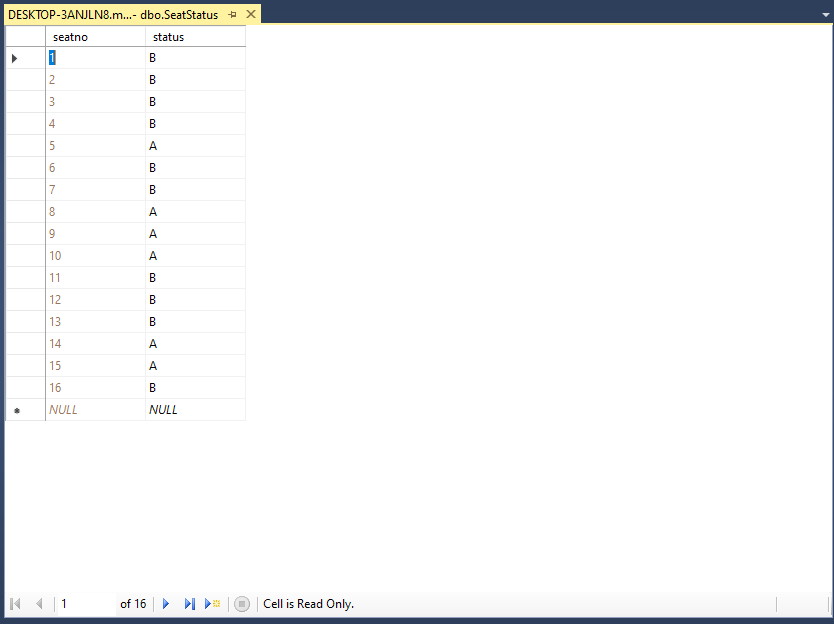


VERİ TABANI İŞLEMLERİ:

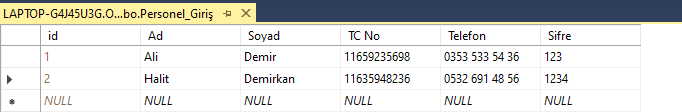
MÜŞTERİNİ VERİ TABANI TABLOLARI: MÜŞTERİ GİRİŞ KISMI:

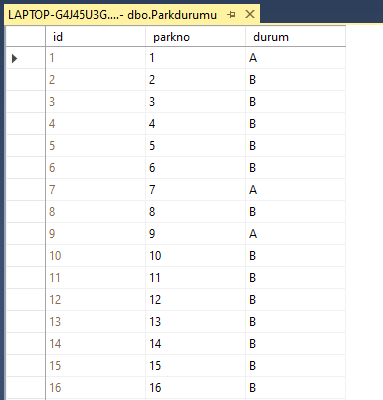


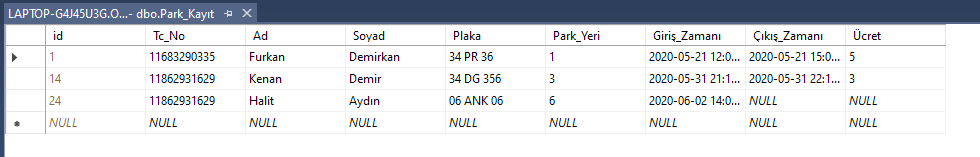
MÜŞTERİ REZERVASYON KISMINDAKİ BUTONLARIN İŞLEMİ:



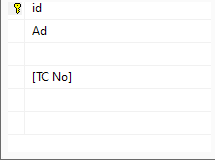
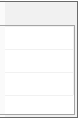
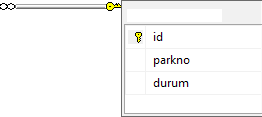
PERSONEL VERİ TABANI TABLOLARI:











Personel\_Giri§

Musteri\_Listesi

ParL Ka

ParLdurumu

Ucret

Park Yeri

g id

Tc No Ad Soyad

g id

Tc No Ad Soyad

C ep Telefon u pIaka Kullanici Adi [E- po sta] Sifre

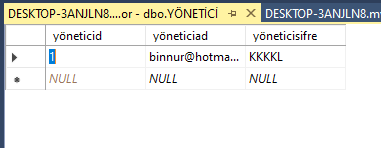
Sifre Tekrar

Telefon

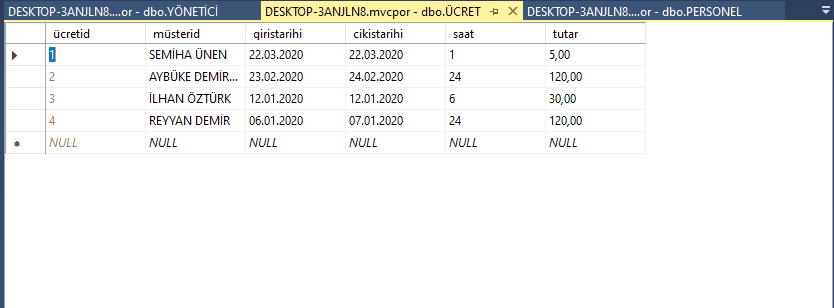
Sifre

Soyad

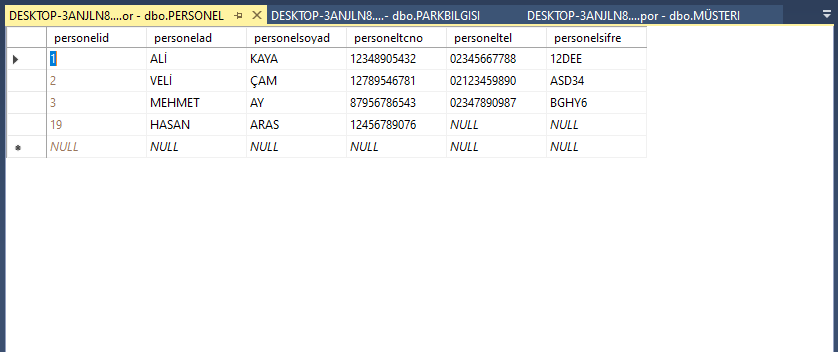
YÖNETİCİ VERİ TABANI TABLOLARI: YÖNETİCİ GİRİŞİ:



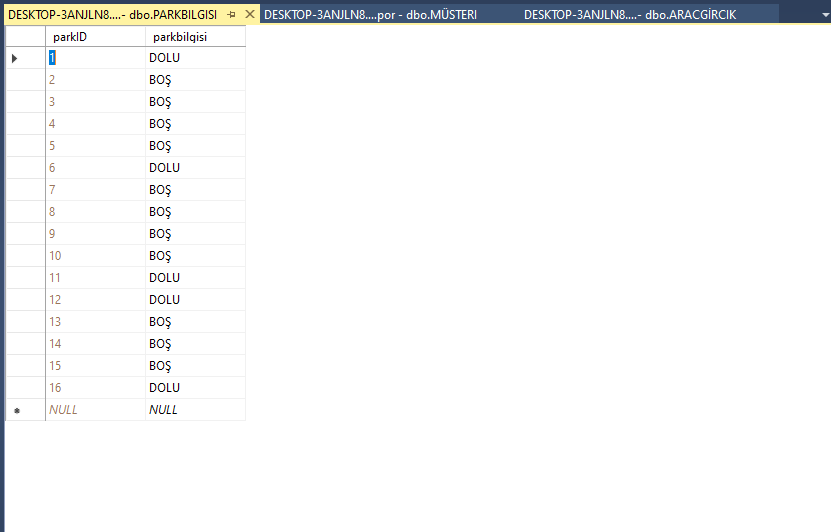
OTOPARKIN ÜCRET BİLGİSİ TABLOSU:



PERSONEL BİLGİLERİNİN BULUNDUĞU TABLO:



PARK BİLGİSİ DURUMLARININ GÖSTERİLDİĞİ TABLO:



MÜŞTERİ BİLGİLERİNİN BULUNDUĞU TABLO:



ARAÇLARIN GİRİŞ/ÇIKIŞ BİLGİLERİNİN TUTULDUĞU TABLO

