# Yönetici sınıfları Yönetici ana sayfa

public class YoneticiAnasayfaController : Controller

{

Otopark\_OtomasyonuEntities1 db = new Otopark\_OtomasyonuEntities1();

// GET: Yonetici

public ActionResult Index()

{

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

}

}

# Yönetici login girişi:

public class YoneticiGirişController : Controller

{

Otopark\_OtomasyonuEntities1 db = new Otopark\_OtomasyonuEntities1();

// GET: YoneticiGiriş public ActionResult Login()

{

return View();

}

[HttpPost]

public ActionResult Login(Yonetici\_Giriş Yonetici)

{

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

if (yonetici != null)

{

FormsAuthentication.SetAuthCookie(yonetici.TC\_No, false); return RedirectToAction("Index", "YoneticiAnasayfa");

}

else

{

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

}

}

public ActionResult Logout()

{

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

}

}

# Yönetici park bilgisi görme

public class ARACGİRCIKController : Controller

{

Otopark\_OtomasyonuEntities1 db = new Otopark\_OtomasyonuEntities1();

// GET: ARACGİRCIK

public ActionResult Index()

{

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

}

}

# Müşteri listeleri:

public class MUSTERIController : Controller

{

Otopark\_OtomasyonuEntities1 db = new Otopark\_OtomasyonuEntities1();

// GET: MUSTERI

public ActionResult Index()

{

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

}

}

# Personel listesi:

public class PersonelListesiController : Controller

{

private Otopark\_OtomasyonuEntities1 db = new Otopark\_OtomasyonuEntities1();

// GET: PersonelListesi public ActionResult Index()

{

return View(db.Personel\_Giriş.ToList());

}

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

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Personel\_Giriş personel\_Giriş = db.Personel\_Giriş.Find(id); if (personel\_Giriş == null)

{

return HttpNotFound();

}

return View(personel\_Giriş);

}

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

{

return View();

}

// POST: PersonelListesi/Create [HttpPost] [ValidateAntiForgeryToken]

public ActionResult Create([Bind(Include = "id,Ad,Soyad,TC\_No,Telefon,Sifre")] Personel\_Giriş personel\_Giriş)

{

if (ModelState.IsValid)

{

db.Personel\_Giriş.Add(personel\_Giriş); db.SaveChanges();

return RedirectToAction("Index");

}

return View(personel\_Giriş);

}

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

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Personel\_Giriş personel\_Giriş = db.Personel\_Giriş.Find(id); if (personel\_Giriş == null)

{

return HttpNotFound();

}

return View(personel\_Giriş);

}

// POST: PersonelListesi/Edit/5 [HttpPost] [ValidateAntiForgeryToken]

public ActionResult Edit([Bind(Include = "id,Ad,Soyad,TC\_No,Telefon,Sifre")] Personel\_Giriş personel\_Giriş)

{

if (ModelState.IsValid)

{

db.Entry(personel\_Giriş).State = EntityState.Modified; db.SaveChanges();

return RedirectToAction("Index");

}

return View(personel\_Giriş);

}

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

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Personel\_Giriş personel\_Giriş = db.Personel\_Giriş.Find(id); if (personel\_Giriş == null)

{

return HttpNotFound();

}

return View(personel\_Giriş);

}

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

public ActionResult DeleteConfirmed(int id)

{

Personel\_Giriş personel\_Giriş = db.Personel\_Giriş.Find(id); db.Personel\_Giriş.Remove(personel\_Giriş); db.SaveChanges();

return RedirectToAction("Index");

}

protected override void Dispose(bool disposing)

{

if (disposing)

{

db.Dispose();

}

base.Dispose(disposing);

}

}