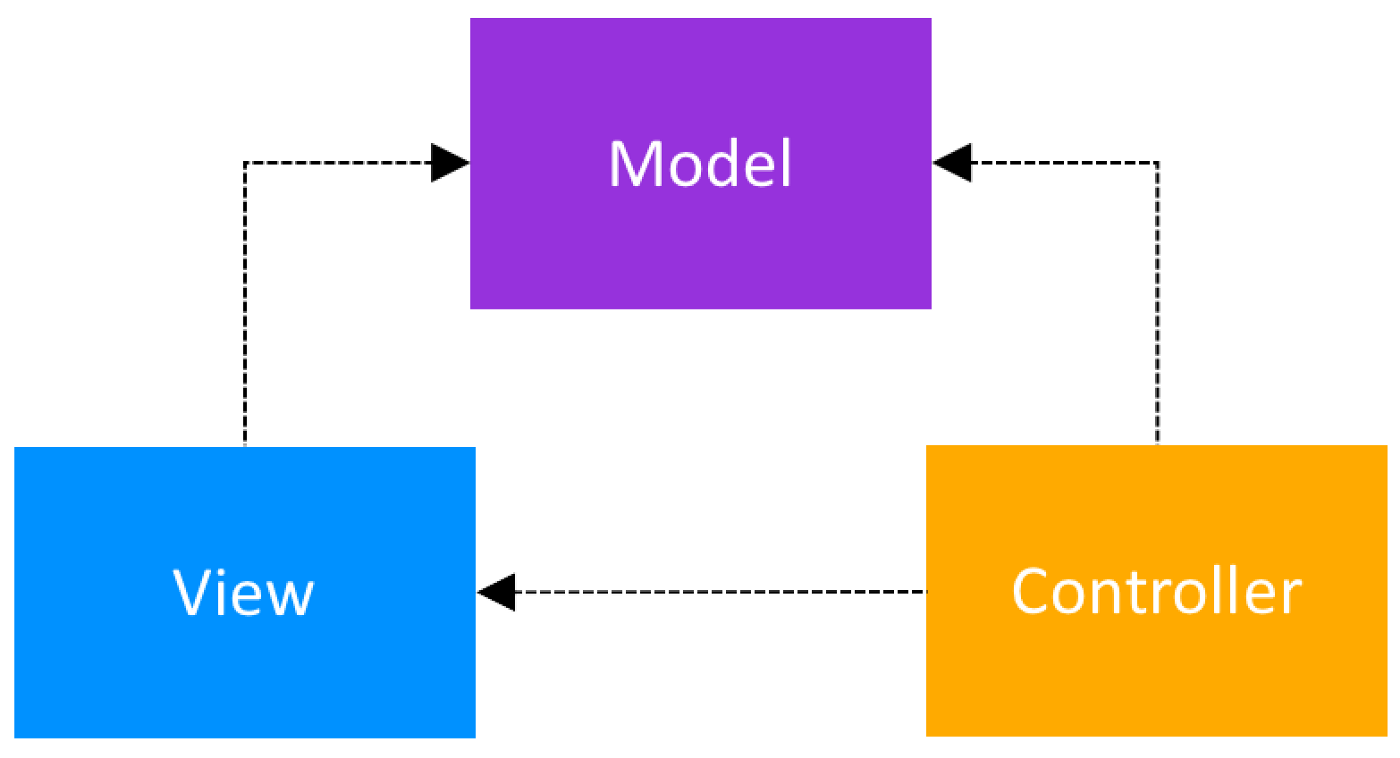
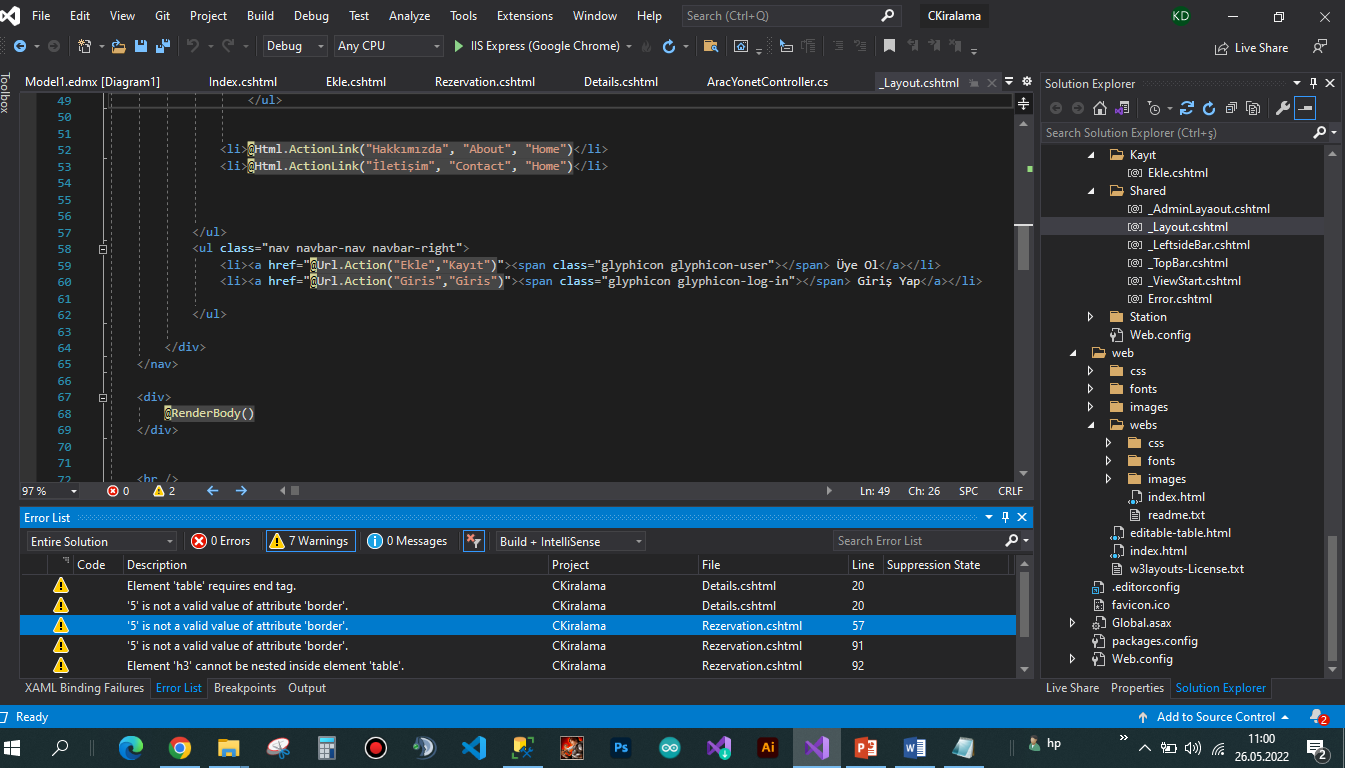
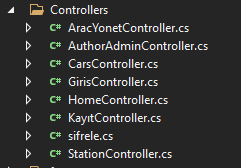
* **DERS ADI: İnternet Programcılığı**
* **DERS HOCASI: SERDAR ARPACI**
* **HAZIRLAYAN: CENKAY DURMUŞ**
* **NUMARASI:201501051**
* **Konu: Bir araba kiralama firmasının araç, müşteri ve kiralama kayıtlarının takibini yapmak amaçlı kullanılacak bir web**
* **uygulaması**

**Proje Hazırlanırken kullanılan ekran kaydı ve kod parçacıkları.**

**1. ASP NET MVC MİMARİSİ İÇİNDE HAZIRLANAN SİTE**



****

**2. SİTENİN SAYFLARI**

**KULLANICI ADMİN /GİRİŞ/TEK SAYFA**

**ÜYE OL EKRANI**

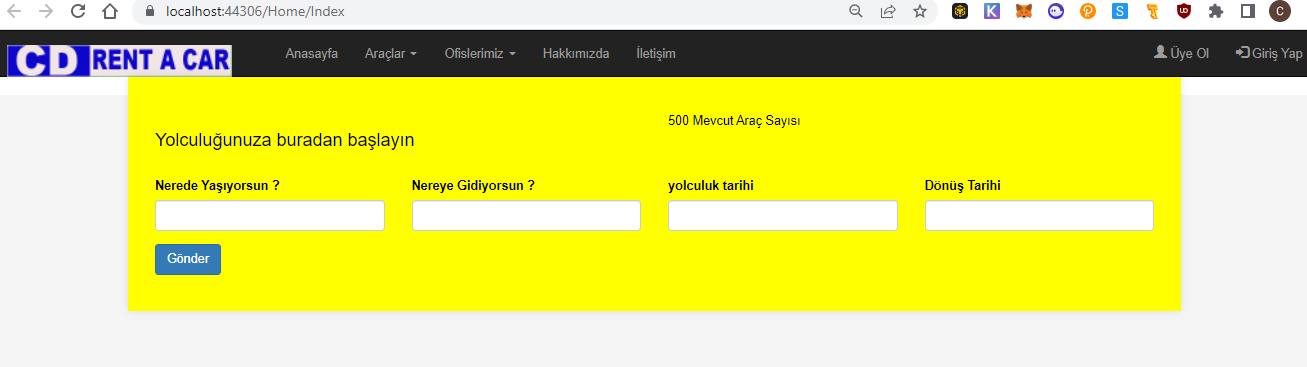
**ANASAYFA**

**ARAÇLAR**

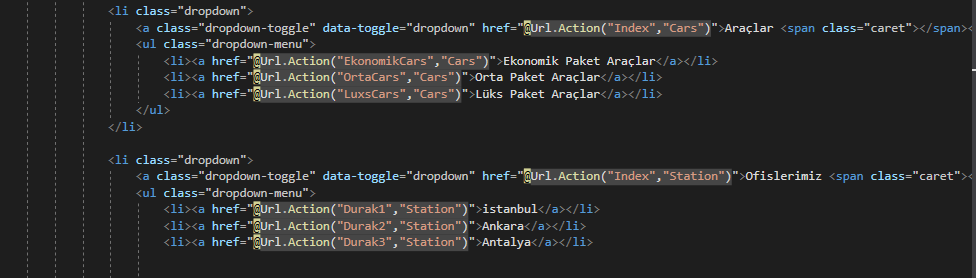
**OFİSLERİMİZ**

**HAKKIMIZDA**

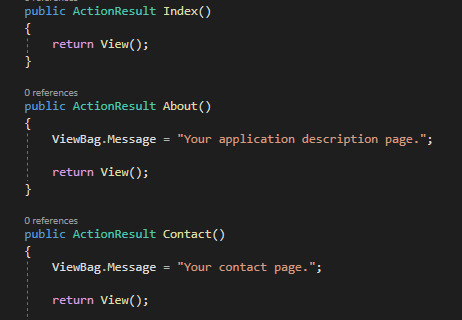
**İLETİŞİM**

****

**ÖRNEK HTML5 KODU LAYOUT**



**ÖRNEK HOME CONTROLLUR KODU**



**SAYFALARIN KOD KISMI VE ARAYÜZÜ**

**1 GİRİŞ EKRANI**



**Kod:**

public class GirisController : Controller

{

private readonly CKiralamaEntities7 db = new CKiralamaEntities7();

// GET: Giris

public ActionResult Index()

{

return View();

}

public ActionResult Giris()

{

return View();

}

[HttpPost]

public ActionResult Giris(Kullanıcı k)

{

var bilgiler = db.Kullanıcı.FirstOrDefault(x => x.Email == k.Email && x.Sifre == k.Sifre);

if (bilgiler != null)

{

FormsAuthentication.SetAuthCookie(bilgiler.Email, false);

if (bilgiler.YetkiID == 2)

return RedirectToAction("Index", "Home");

if (bilgiler.YetkiID==1)

return RedirectToAction("Create", "AuthorAdmin");

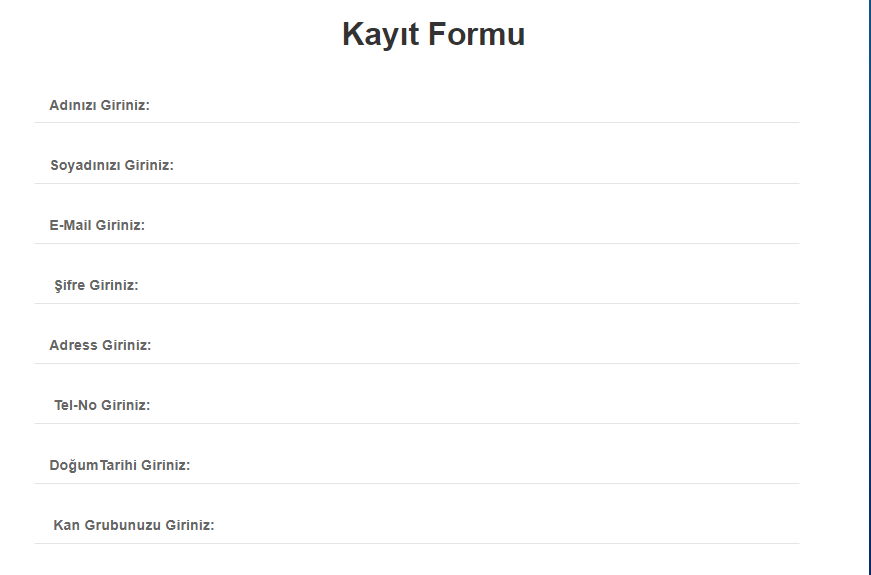
}

ViewBag.Hata("Kullanıcı Adı veya Şifre Yanlış");

return View();

}

**2.ÜYE OL**



**Kod:**

public class KayıtController : Controller

{

private readonly CKiralamaEntities7 db = new CKiralamaEntities7();

// GET: Kayıt

public ActionResult Index()

{

return View();

}

[HttpGet]

public ActionResult Ekle()

{

return View();

}

[HttpPost]

public ActionResult Ekle(Kullanıcı p1)

{

db.Kullanıcı.Add(p1);

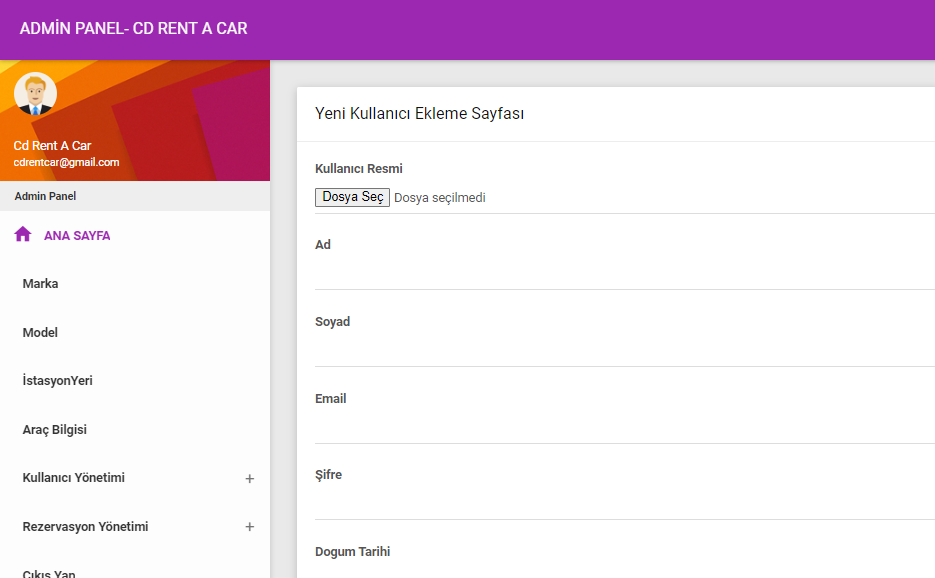
db.SaveChanges();

return RedirectToAction("Giris", "Giris");

}

}

**ADMİN GİRİŞ EKRANI**



**Kod:**

public class AuthorAdminController : Controller

{

private readonly CKiralamaEntities7 db = new CKiralamaEntities7();

// GET: AuthorAdmin

public ActionResult Index()

{

return View();

}

public ActionResult Create()

{

return View();

}

[HttpPost]

public ActionResult Create(Kullanıcı author, HttpPostedFileBase File)

{

var authorExist = db.Kullanıcı.Any(m => m.Email == author.Email);

if (authorExist == false)

{

//author.DogumTarihi = DateTime.Now;//session kullanarak yapılacak alan.

// author.Ad = "";

//author.Soyad = "";

if (File != null)

{

FileInfo fileinfo = new FileInfo(File.FileName);

WebImage img = new WebImage(File.InputStream);

string uzanti = (Guid.NewGuid().ToString() + fileinfo.Extension).ToLower();

img.Resize(225, 180, false, false);

string tamyol = "~/images/users/" + uzanti;

img.Save(Server.MapPath(tamyol));

author.Image = "/images/users/" + uzanti;

}

db.Kullanıcı.Add(author);

db.SaveChanges();

}

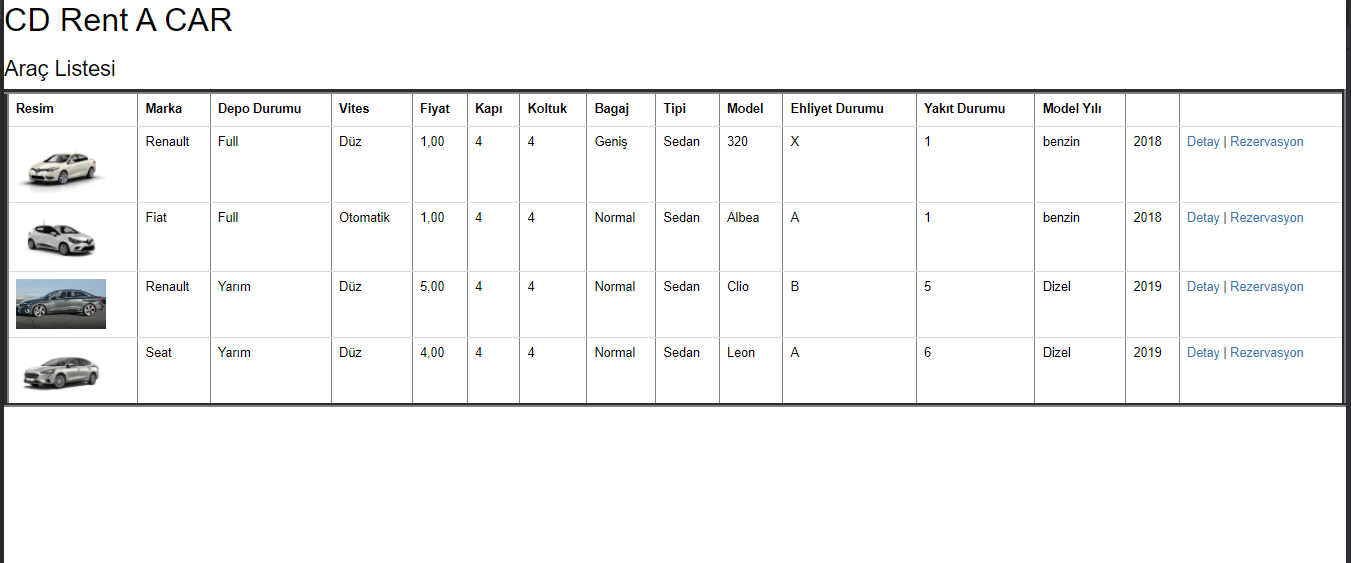
return RedirectToAction("Index", "AuthorAdmin");

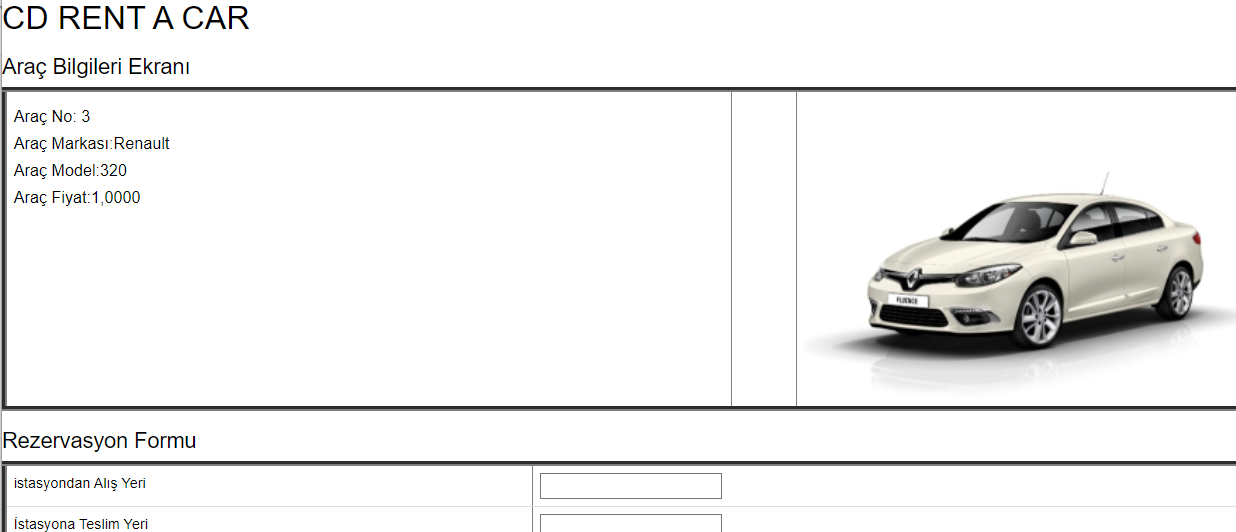
}

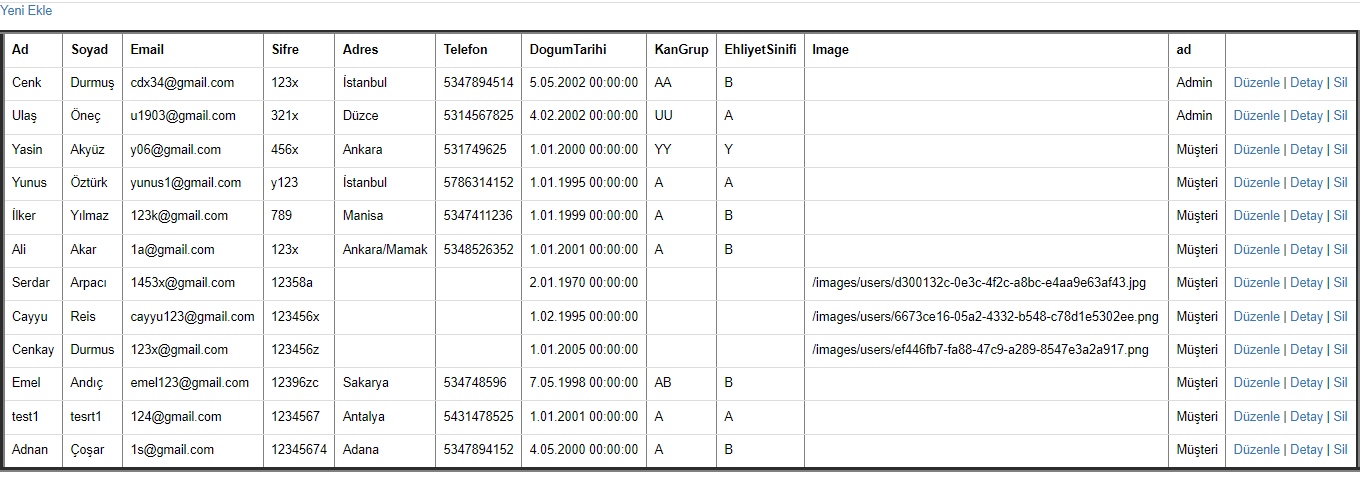
}

}

Araç bilgi ve Rezervasyon sayfası





Kullanıcı Edit

KOD

namespace CKiralama.Controllers

{

public class KullanıcıEditController : Controller

{

private CKiralamaEntities8 db = new CKiralamaEntities8();

// GET: KullanıcıEdit

public ActionResult Index()

{

var kullanıcı = db.Kullanıcı.Include(k => k.Yetki);

return View(kullanıcı.ToList());

}

// GET: KullanıcıEdit/Details/5

public ActionResult Details(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Kullanıcı kullanıcı = db.Kullanıcı.Find(id);

if (kullanıcı == null)

{

return HttpNotFound();

}

return View(kullanıcı);

}

// GET: KullanıcıEdit/Create

public ActionResult Create()

{

ViewBag.YetkiID = new SelectList(db.Yetki, "YetkiID", "ad");

return View();

}

// POST: KullanıcıEdit/Create

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

// more details see https://go.microsoft.com/fwlink/?LinkId=317598.

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Create([Bind(Include = "KullanıcıID,Ad,Soyad,Email,Sifre,Adres,Telefon,DogumTarihi,KanGrup,EhliyetSinifi,YetkiID,Image")] Kullanıcı kullanıcı)

{

if (ModelState.IsValid)

{

db.Kullanıcı.Add(kullanıcı);

db.SaveChanges();

return RedirectToAction("Index");

}

ViewBag.YetkiID = new SelectList(db.Yetki, "YetkiID", "ad", kullanıcı.YetkiID);

return View(kullanıcı);

}

// GET: KullanıcıEdit/Edit/5

public ActionResult Edit(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Kullanıcı kullanıcı = db.Kullanıcı.Find(id);

if (kullanıcı == null)

{

return HttpNotFound();

}

ViewBag.YetkiID = new SelectList(db.Yetki, "YetkiID", "ad", kullanıcı.YetkiID);

return View(kullanıcı);

}

// POST: KullanıcıEdit/Edit/5

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

// more details see https://go.microsoft.com/fwlink/?LinkId=317598.

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Edit([Bind(Include = "KullanıcıID,Ad,Soyad,Email,Sifre,Adres,Telefon,DogumTarihi,KanGrup,EhliyetSinifi,YetkiID,Image")] Kullanıcı kullanıcı)

{

if (ModelState.IsValid)

{

db.Entry(kullanıcı).State = EntityState.Modified;

db.SaveChanges();

return RedirectToAction("Index");

}

ViewBag.YetkiID = new SelectList(db.Yetki, "YetkiID", "ad", kullanıcı.YetkiID);

return View(kullanıcı);

}

// GET: KullanıcıEdit/Delete/5

public ActionResult Delete(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Kullanıcı kullanıcı = db.Kullanıcı.Find(id);

if (kullanıcı == null)

{

return HttpNotFound();

}

return View(kullanıcı);

}

// POST: KullanıcıEdit/Delete/5

[HttpPost, ActionName("Delete")]

[ValidateAntiForgeryToken]

public ActionResult DeleteConfirmed(int id)

{

Kullanıcı kullanıcı = db.Kullanıcı.Find(id);

db.Kullanıcı.Remove(kullanıcı);

db.SaveChanges();

return RedirectToAction("Index");

}

protected override void Dispose(bool disposing)

{

if (disposing)

{

db.Dispose();

}

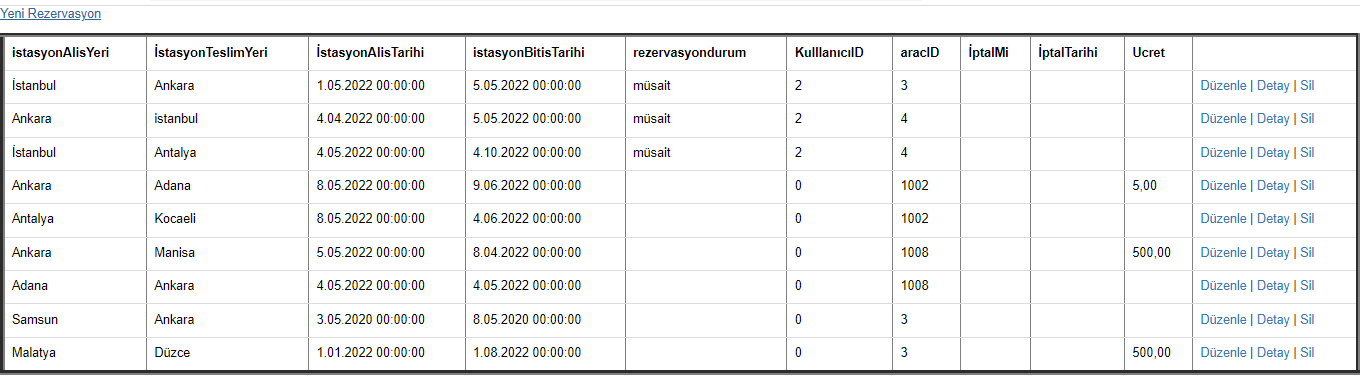
base.Dispose(disposing);

}

}

}

Rezervasyon Edit



Kod:

namespace CKiralama.Controllers

{

public class AracRezervasyonEditController : Controller

{

private CKiralamaEntities8 db = new CKiralamaEntities8();

// GET: AracRezervasyonEdit

public ActionResult Index()

{

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

}

// GET: AracRezervasyonEdit/Details/5

public ActionResult Details(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

AracRezervasyon aracRezervasyon = db.AracRezervasyon.Find(id);

if (aracRezervasyon == null)

{

return HttpNotFound();

}

return View(aracRezervasyon);

}

// GET: AracRezervasyonEdit/Create

public ActionResult Create()

{

return View();

}

// POST: AracRezervasyonEdit/Create

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

// more details see https://go.microsoft.com/fwlink/?LinkId=317598.

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Create([Bind(Include = "rezervasyonID,istasyonAlisYeri,İstasyonTeslimYeri,İstasyonAlisTarihi,istasyonBitisTarihi,rezervasyondurum,KulllanıcıID,aracID,İptalMi,İptalTarihi,Ucret")] AracRezervasyon aracRezervasyon)

{

if (ModelState.IsValid)

{

db.AracRezervasyon.Add(aracRezervasyon);

db.SaveChanges();

return RedirectToAction("Index");

}

return View(aracRezervasyon);

}

// GET: AracRezervasyonEdit/Edit/5

public ActionResult Edit(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

AracRezervasyon aracRezervasyon = db.AracRezervasyon.Find(id);

if (aracRezervasyon == null)

{

return HttpNotFound();

}

return View(aracRezervasyon);

}

// POST: AracRezervasyonEdit/Edit/5

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

// more details see https://go.microsoft.com/fwlink/?LinkId=317598.

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Edit([Bind(Include = "rezervasyonID,istasyonAlisYeri,İstasyonTeslimYeri,İstasyonAlisTarihi,istasyonBitisTarihi,rezervasyondurum,KulllanıcıID,aracID,İptalMi,İptalTarihi,Ucret")] AracRezervasyon aracRezervasyon)

{

if (ModelState.IsValid)

{

db.Entry(aracRezervasyon).State = EntityState.Modified;

db.SaveChanges();

return RedirectToAction("Index");

}

return View(aracRezervasyon);

}

// GET: AracRezervasyonEdit/Delete/5

public ActionResult Delete(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

AracRezervasyon aracRezervasyon = db.AracRezervasyon.Find(id);

if (aracRezervasyon == null)

{

return HttpNotFound();

}

return View(aracRezervasyon);

}

// POST: AracRezervasyonEdit/Delete/5

[HttpPost, ActionName("Delete")]

[ValidateAntiForgeryToken]

public ActionResult DeleteConfirmed(int id)

{

AracRezervasyon aracRezervasyon = db.AracRezervasyon.Find(id);

db.AracRezervasyon.Remove(aracRezervasyon);

db.SaveChanges();

return RedirectToAction("Index");

}

protected override void Dispose(bool disposing)

{

if (disposing)

{

db.Dispose();

}

base.Dispose(disposing);

}

}

}

Kod:

public class AracYonetController : Controller

{

CKiralamaEntities7 db = new CKiralamaEntities7();

// GET: AracYonet

public ActionResult Index()

{

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

}

public ActionResult Details(int? id)

{

AracBilgisi aracverisi = db.AracBilgisi.Find(id);

return View(aracverisi);

}

[HttpGet]

public ActionResult Rezervation(int? id)

{

AracBilgisi aracverisi = db.AracBilgisi.Find(id);

ViewData ["Marka"] = aracverisi.aracMarka;

ViewData ["Model"] = aracverisi.aracModel;

ViewData ["Fiyat"] = aracverisi.aracFiyat;

return View();

}

[HttpPost]

public ActionResult Rezervation([Bind(Include = "RezervasyonID,AracID,KulllanıcıID,İstasyonAlisTarihi,istasyonBitisTarihi,Ucret,istasyonAlisYeri,İstasyonTeslimYeri")] AracRezervasyon rezervasyon)

{

if(ModelState.IsValid)

{

AracRezervasyon aracRezervasyon = db.AracRezervasyon.Add(rezervasyon);

NewMethod();

}

ViewBag.Message = "Tebrikler Rezervasyon işleminiz Başrıyla Gerçekleşti.";

return View();

void NewMethod()

{

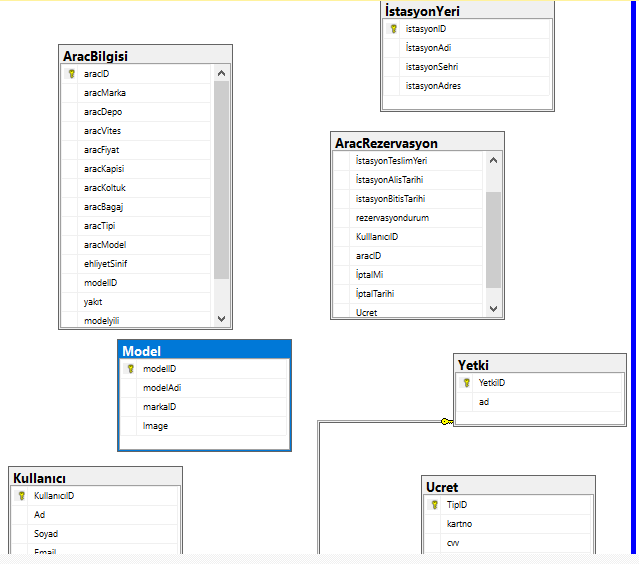
db.SaveChanges();

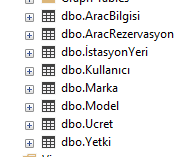
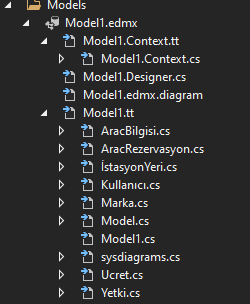
}

}

}

Veri Tabanı Model :



****

**Proje Açıklama: Proje de mvc mimarisi kullanılmıştır.**

**Contollerle güvenlik ve giriş tarafın da kontroller yapılmıştır.**

**Querystring, session, cookie**

**C#, veritabanı olarak SQL Server**

**Entity Framework, Bootstrap, Javascript**

**Vb kullanılmıştır.**

**Rezervasyon kısmın da bazı özelliklerden yoksun kalınmış ve veri tabanı hataları dolayısıyla**

**Kullanıcı kayıt oluşurken**

**Sil güncelle alanların da çeşitli sorunlar yaşanmıştır.**