**2.1 Proje Oluşturulması**

Master Page veya Layout sayfada sabit kalan alana denir.

**2.2 Modelin Kurulması**

Modelde Entitiy klasörü tanımlandı. Entitiy klasörüne yeni data ögesi eklendi.DB first kullanıldı   
DB First: Yani dışarıda harici bir veritabanı var sen bunu projeye entegre ediceksin  
Code First: Dışarıdan veritabanı ile çalışmak istemezsek herşeyi kod ile yazıcaksak bu kullanılır

**2.3 Class ve Propertyler**

**2.4 View ve Layout**

W3schools ‘ tan sidebar kopyalandı . Shared klasörüne kendi Layout umuz oluşturduk.Kategori Controllerı oluşturuldu.KategoriController ‘da Index üzerinden View oluşturduk ve kendi Layout umuzu seçtik.Layoutta Renderbody yani viewlerin olacağı kısmı yazdık.

**2.5 Sidebar Kurulması**

Sidebar kodları layouta kopyalandı css ayarları yapıldı

**2.6 Sidebarın Hazırlanması**

Adları değiştirildi

**2.7 Kategori Controllerı oluşturma**

Veri listelemek için controller da entitiy kütüphanesini ekledik ve nesne üretip listeleme metodu kullandık

**2.8 Table Kullanımı**

<td> ile tablo oluşturuldu ve border ile çerçevesi verildi

**2.9 Kategori Listeleme**

Kategorinin index sayfasına Models.entity kütüphanesi ekledik sonra tblkategori için list oluşturuldu ve modele atandı . foreach döngüsü ile model gezildi ve veriler okundu tablo satırlarına yazdırıldı

**2.10 Kategori Ekleme**

Kategori Controller ın içine action result eklendi onun üstüne view oluşturuldu view de form oluşturuldu

**2.11 HTTPGET HTTPPOST**

Kategori controllerın da ekleme işlemi için ekleme action resultun başına HTTPPOST yazıldı  
HTTPPOST sayfaya tıklandığında veya gönderilidiğinde çalışır  
HTTPGET sayfa yüklendiğinde

**2.12 Kategori Silme**

Kategorinin ındex sayfasındaki silme butonuna kategorisil action resultuna gitme yolu yazıldı ve kategori controllerda KategoriSil actionresult u yazıldı

**2.13 Kategori Güncelleme**

KategoriGetir view ı oluşturuldu ve controller da kategorigetir action resultu yazıldı daha sonra view da butona basılıdığında kategoriguncelle action resultu oluşturuldu ve kodları yazıldı.

**3.1 Yazar Listeleme**

Yazar controllerı oluşturuldu ve entitiy kütüphanesi eklendi entitiy den database nesnesi oluşturuldu.Index te değerler değişkenine tablo listelendi ve view da döndürüldü.Daha sonra Index view oluşturuldu

**3.2 Yazar Ekleme**

Action Result u oluşturukdu ve HTTPPOST HTTPGET kullanıldı yazarekle view ı oluşturuldu view a kütüphane eklendi texboxfor kullanılarak

**3.3 Yazar Silme**

Index e yazar silme butonu oluşturuldu ve kodları yazıldı action result oluşturuldu.

**3.4 Yazar Getirme**

**4.1 Kitap Listeleme**

Herzaman ki listeleme tek fark ortak sütunlarda tablo adını kullanarak view da gösterme

**4.2 Dropdown kullanımı**

Kitap contollerında kitapekle actionresultun ın içine fakat HTTPGET olan kodlar açıklamalı yazıldı daha sonra KitapEkle View da Dropdownlistfor ile gösterildi

**4.3 Kitap Ekleme**

Controller da KitapEkle ActionResult lar oluşturuldu HTTPGET kısmını yazar ve kategorinin dropdownlist olarak gözükmesi için gerekli kodları yazıldı.HTTPOST kısmına ekleme işlemi için biraz farklı kod yazıldı.Sonra View oluşturuldu yazar ile kategori için DropDownListFor kullanıldı.Viewbag ile taşındı

**4.4 Kitap Silme**

Kitap silme işlemi herzaman ki gibi

**4.6 Kitap Güncelleme**

Kitap Controller a KitapGetir ActionResultı yazıldı ve bundan view oluşturuldu KitapGetir ActionR’ ye Dropdownlist için ekstra kodlar yazıldı. Daha sonra KitapGetir view a TextBoxFor ve DropdownList ile verilr getirildi.Sonra KitapGuncelle ActionR yazıldı

**4.7 Kitap Arama**

Controlller da Index ActionR ‘da arama için farklı kodlar kullanıldı ve IndexView da BeginForm ile arama işlemi yapıldı

**5.1 Personel Listesi**

Her zamanki listeleme işlemleri…

**5.2 Personel Ekleme ve DATA ANNOTATİONS**

Data Annotations : Veri girilen alanda gerekli şartları sağlamasını istersek bunu kullanırız

Model içine entitiy klasörünün içinde modelinin içinde tablomuza tıkladık . using ile DataAnnotations kütüphanesini ekledik Required yani boş geçilemez kodunu kullandık. Sonra Controller da PersonelEkle ActionR’a Dataannotations şartlarını sağlaması için kod yazdık.View da gerekli kodlarıda yazdık

**5.3 Personel Silme ve Güncelleme**

Her zamanki işlemler

**6.1 Üye Listeleme**

Her zamanki listeleme işlemleri

**6.2 Üye Ekleme (DATA ANNOTATİONS)**

UyeController a UyeEkle ActionR’ ları yazıldı.Sonra Model de TBLUYELERE’e girildi kütüphane eklend ive [Required] [StringLength] vb. şeyler eklendi ve View ‘a geçildi.

**6.3 Üyeleri Sayfalama (PAGGİNG)**

Controller da Index için içine “Sayfa” diye değişten oluşturuldu ve 1 değeri verildi.Projeye Nuget eklentilerinden “PAGEDLİST” yüklendi. Hem Controller a hem de Index View a kütüphanesi eklendi ve List kütüphanesi Pagedlist ile değiştirildi..Index ActionR ‘ a farklı bir listeleme kodu yazıldı.

**6.4 Silme ve Güncelleme**

Klasik silme ve güncelleme işlemleri

**7.1 Ödünç Verme Sayfa Tasarımı**Klasik sayfa tasarımı

**7.2 Ödünç Verme İşlemi**

Klasik ekleme işlemi yapıldı fakat View kısmında Tarihler için input kullanıldı ve datetime ile tarih formatı olması sağlandı kodların açıklamasını yaptım

**7.3 İade Sayfa Tasarımı**  
Klasik listeleme işlemleri yapıldı

**7.4 Tarih Formatları Düzenleme**

Index View da tarihler istenilen formatta gösterildi

**8.1 Durum Değeri Ayarı**

Eklenilen kitapların durumunun başlangıç değerinin TRUE olması sağlandı

**8.3 Trigger İle Durum Değerini Değiştirme**

SQL SERVER’da sorgu ile TBLHAREKET tablosu üzerinde trigger komutları yazıldı ve Execute edildi

**8.4 Kitap İade Etme İşlemi**

Odunciade ActionResultu ı yazıldı ve Odunciade’den View oluşturuldu . View’a gerekli kodlar yazıldı

10.1 Vitrin Paneli Projeye Dahil Etme