

VIEW

Dr. Öğretim Üyesi Fatma AKALIN

VIEW KULLANIMI

View, MVC'de projede kullanıcının gördüğü ve kullandığı arayüzdür. Bu bölümde projenin kullanıcılara sunulacak olan html dosyaları yer almaktadır. Sizler tarafından yapılan projenin kodları sunucuya aktarıldıktan sonra kullanıcının internet tarayıcısı üzerinden siteyi açtığında karşısına gelen sayfanın kodlamasıdır

Kullanıcılardan alınacak olan istekler Viewlar aracılığı ile ilgili Controller'a iletilir. Farklı bir deyişle kullanıcılar sadece viewları gördükleri için yapacakları tüm istekler yine bu viewlar aracılığıyla yapılır. Ardından bu istekler sunucu tarafından alınıp ilgili controllera verilir

Örneğin <https://www.hepsiburada.com/ara?q=android> sitesinde anasayfada yer alan bir ürünün linkine tıkladığımız zaman isteği anasayfa View'ından yapmış oluruz. Ardından istek sunucuya gider. Sunucuda bu istekle hangi controller ilgileniyor ise ona yönlendirme sağlanır.

Dr. Öğr. Üyesi Fatma AKYILMAZ

Şimdi sıfırdan yeni bir proje oluşturalım.

Yeni projenizi yapılandırın

ASP.NET Web Uygulaması (.NET Framework)

C#

Windows

Bulut

Web

Proje adı

denemeikinciders

Konum

D:\aspnet\1_ADESLER

Çözüm adı ⓘ

denemeikinciders

☐ Çözümü ve projeyi aynı dizine yerleştirin

Altyapı

.NET Framework 4.6.2

Proje "D:\aspnet\1_ADESLER\denemeikinciders\denemeikinciders\" içinde oluşturulacak

Yeni ASP.NET Web Uygulaması oluřtur



Boř

ASP.NET uygulamaları oluřturmak için boř bir proje řablonu. Bu řablonda ierik yoktur.



Web Forms

ASP.NET Web Forms uygulamaları oluřturmak için bir proje řablonu. ASP.NET Web Forms, bilinen srkle ve bırak yntemiyle olay denetimli modeli kullanarak dinamik web siteleri oluřturmanızı saęlar. Tasarım yzeyi ile yzlerce denetim ve bileřen, veri eriřimi olan geliřmiř, g, kullanıcı arabirimi denetimli web sitelerini hızlı bir řekilde oluřturmanızı saęlar.



MVC

ASP.NET MVC uygulamaları oluřturmaya yarayan bir proje řablonu. ASP.NET MVC, Model-View-Controller mimarisini kullanarak uygulamalar oluřturmanıza olanak saęlar. ASP.NET MVC en son standartları kullanan uygulamalar oluřturmak için hızlı, test gdml geliřtirmeye olanak saęlayan birok zellik ierir.



Web API

Tarayıcılar ve mobil cihazlar dahil olmak zere ok eřitli istemcilere eriřebilen RESTful HTTP hizmetleri oluřturmaya yarayan bir proje řablonudur.



Tek Sayfalı Uygulama

ASP.NET Web API kullanarak JavaScript temelli zengin istemci tarafı HTML5 uygulamaları oluřturmak zere kullanılan proje řablonudur. Tek Sayfalık Uygulamalar, HTML5, CSS3 ve JavaScript'in kullanıldıęı istemci tarafı etkileřimler ieren zengin bir kullanıcı deneyimi saęlar.

Kimlik Doęrulama

Yok

Klasrleri ve ekirdek bařvurularını ekle

- ☐ Web Forms
☒ MVC
☐ Web API'si

Geliřmiř

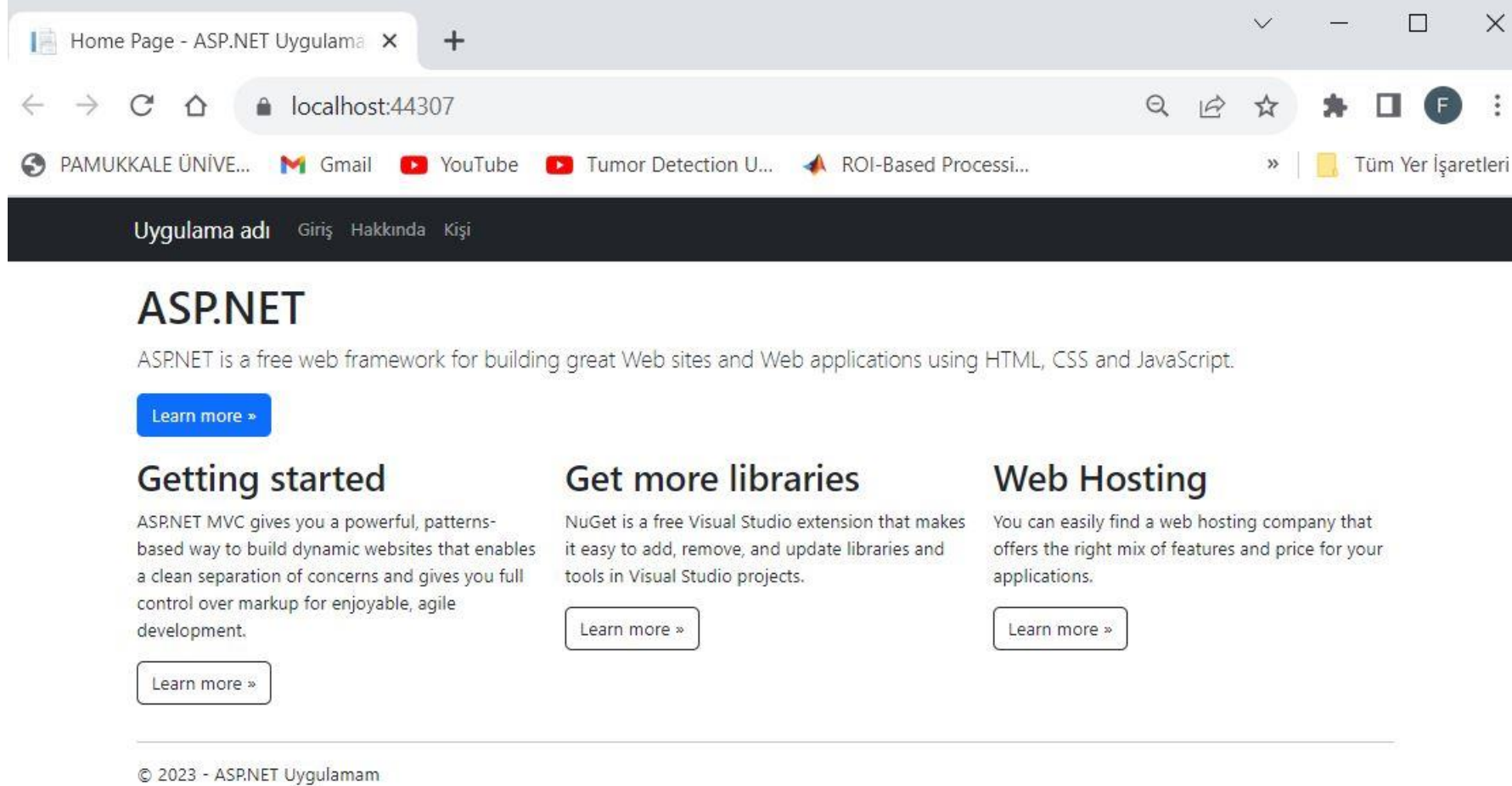
- ☒ HTTPS'yi Yapılandır
☐ Docker desteęi
([Docker Desktop](#) gerektiriyor)
☐ Ayrıca birim testleri için bir proje oluřtur

denemeikinciders.Tests

Geri

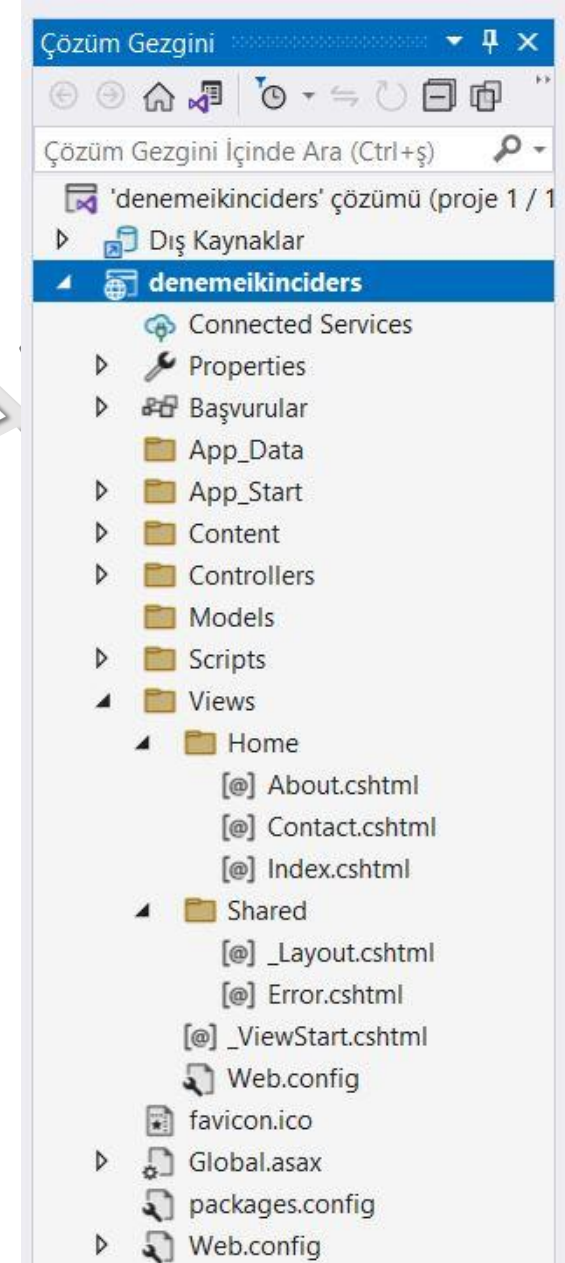
Oluřtur

Boş bir MVC yapısı seçmediğimiz için karşımıza çıkan ekran aşağıdaki gibidir.



DİZİN YAPISI

Dizin yapısına baktığımızda kullanıcının gördüğü ekran olarak nitelendirdiğimiz Viewlar, projede Views klasörünün altında yer almaktadır. Oluşturduğumuz projenin dizin yapısı sağda verilmiştir.



Yeni bir proje oluşturduğunuzda View dizininin altında Home ve shared dizinleri bulunur. Bu dizinler içerisinde cshtml uzantılı viewlar bulunur.

Tüm viewların uzantısı aynıdır fakat farklı yapıda olabilirler.

Bu doğrultuda View dizininin içerisinde yer alan alt dizinlerdeki viewların dizin yapısı ile kullanım yöntemleri arasındaki ilişkiyi inceleyelim

Dr. Öğr. Üyesi Fatma AKALIN

1- Home : HomeController içerisindeki Actionlara ait viewların yer aldığı bir dizindir. Her action için bir adet View oluşturulmuştur.

2- Shared : bu klasör tüm viewlar tarafından ortak kullanılan view dosyalarıdır.

2.1. _Layout.cshtml:Projenin temel şablonudur. Diğer viewlar bu şablon içerisinde görüntülenebilir.

2.2. _LoginPartial.cshtml: Kullanıcı giriş işleminin yapıldığı viewdır.

2.3. Error.cshtml:Projenin hata sayfasıdır. Bir hata ile karşılaşıldığında bu sayfaya yönlendirme sağlanır.

CONTROLLER – VIEW İLİŞKİSİ

Controllerlar, action result türünde bir veri döndürür ve geri döndürdüğü değer View metodudur. View metodu, HomeController'ın kalıtımını aldığı Controller sınıfı içerisinde yer almaktadır.

İlk derslerde Empty olarak nitelendirilen bir proje açmıştık. İçerisinde Controller ya da view yoktu. BU sayfayı tarayıcıda görüntülemek istemiştik fakat hata almıştır. Bu nedenle ilk olarak controller ve ardından view tasarlayarak bu sorunu çözmüştük.

LAYOUT(ŞABLON SAYFA) KULLANIMI

ASP.NET MVC ile proje geliştirirken tasarımsal olarak aynı bölümlerin birçok sayfada kullanılması gerekebilir. Web sitelerimizin sabit ve değişken kısımları vardır. Sabit olan sayfalara layout ismini veriyoruz. Layout sayesinde tüm view'lerde sabit olan bölümleri bir kere oluşturup değişken yerleri View bazında düzenleyebiliriz. Sabit yerleri layout üzerinden tek bir noktadan değiştirdiğinizde bu değişim çalışma anında tüm sayfalarda da otomatik olarak sağlanacaktır

Layout şablonunu anlatmak üzere yeni bir proje oluşturalım..

Yeni ASP.NET Web Uygulaması oluştur



Boş

ASP.NET uygulamaları oluşturmak için boş bir proje şablonu. Bu şablonda içerik yoktur.



Web Forms

ASP.NET Web Forms uygulamaları oluşturmak için bir proje şablonu. ASP.NET Web Forms, bilinen sürükle ve bırak yöntemiyle olay denetimli modeli kullanarak dinamik web siteleri oluşturmanızı sağlar. Tasarım yüzeyi ile yüzlerce denetim ve bileşen, veri erişimi olan gelişmiş, güçlü, kullanıcı arabirimi denetimli web sitelerini hızlı bir şekilde oluşturmanızı sağlar.



MVC

ASP.NET MVC uygulamaları oluşturmaya yarayan bir proje şablonu. ASP.NET MVC, Model-View-Controller mimarisini kullanarak uygulamalar oluşturmanıza olanak sağlar. ASP.NET MVC en son standartları kullanan uygulamalar oluşturmak için hızlı, test güdümlü geliştirmeye olanak sağlayan birçok özellik içerir.



Web API

Tarayıcılar ve mobil cihazlar dahil olmak üzere çok çeşitli istemcilere erişebilen RESTful HTTP hizmetleri oluşturmaya yarayan bir proje şablonudur.



Tek Sayfalı Uygulama

ASP.NET Web API kullanarak JavaScript temelli zengin istemci tarafı HTML5 uygulamaları oluşturmak üzere kullanılan proje şablonudur. Tek Sayfalık Uygulamalar, HTML5, CSS3 ve JavaScript'in kullanıldığı istemci tarafı etkileşimler içeren zengin bir kullanıcı deneyimi sağlar.

Kimlik Doğrulama

Yok

Klasörleri ve çekirdek başvurularını ekle

- ☐ Web Forms
☒ MVC
☐ Web API'si

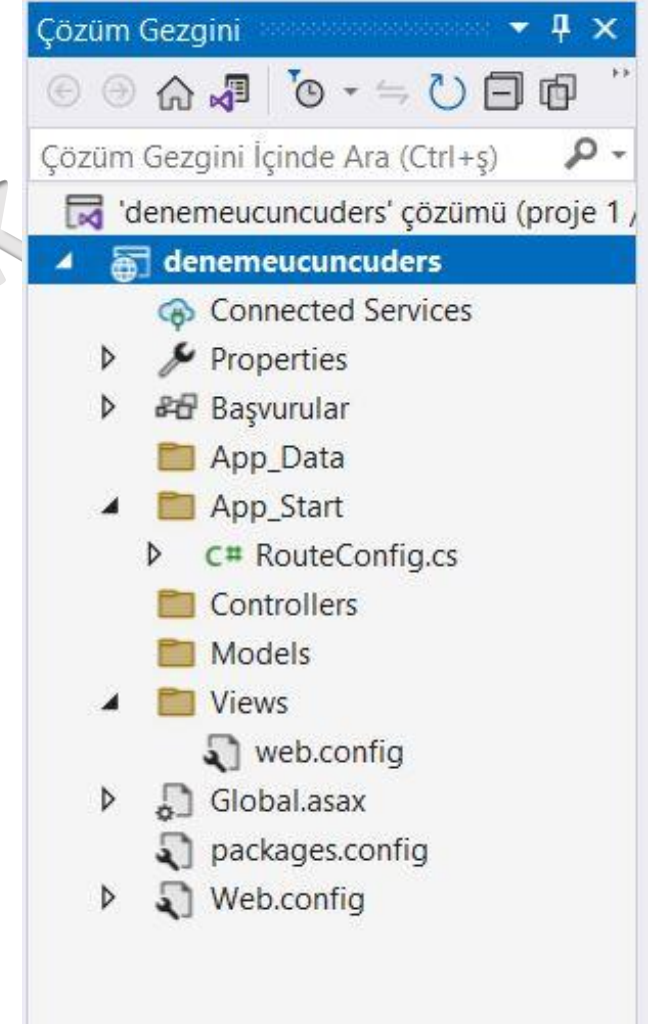
Gelişmiş

- ☒ HTTPS'yi Yapılandır
☐ Docker desteği
(Docker Desktop gerekiyor)
☐ Ayrıca birim testleri için bir proje oluştur

denemeucuncuders.Tests

Geri

Oluştur



HomeController isminde bir denetleyici oluřturalım.

Yeni İskeleli Öęe Ekle

Yükli

Ortak

MVC

Alan

Denetleyici

Göster

Web API'si

Entity Framework kullanan, görünümleri olan MVC 5 Denetleyici

MVC 5 Denetleyici - Boř

Okuma/yazma eylemleri olan MVC 5 Denetleyici

MVC 5 Denetleyici - Boř
řunun tarafından Microsoft
v5.0.0.0
Boř bir MVC denetleyici.
Kimlik: MvcControllerEmptyScaffolder

AKALIN

Denetleyici Ekle

Denetleyici adı:

HomeController

Ekle

İptal

Ekle

İptal

Controllerlarımızı
oluşturalım.

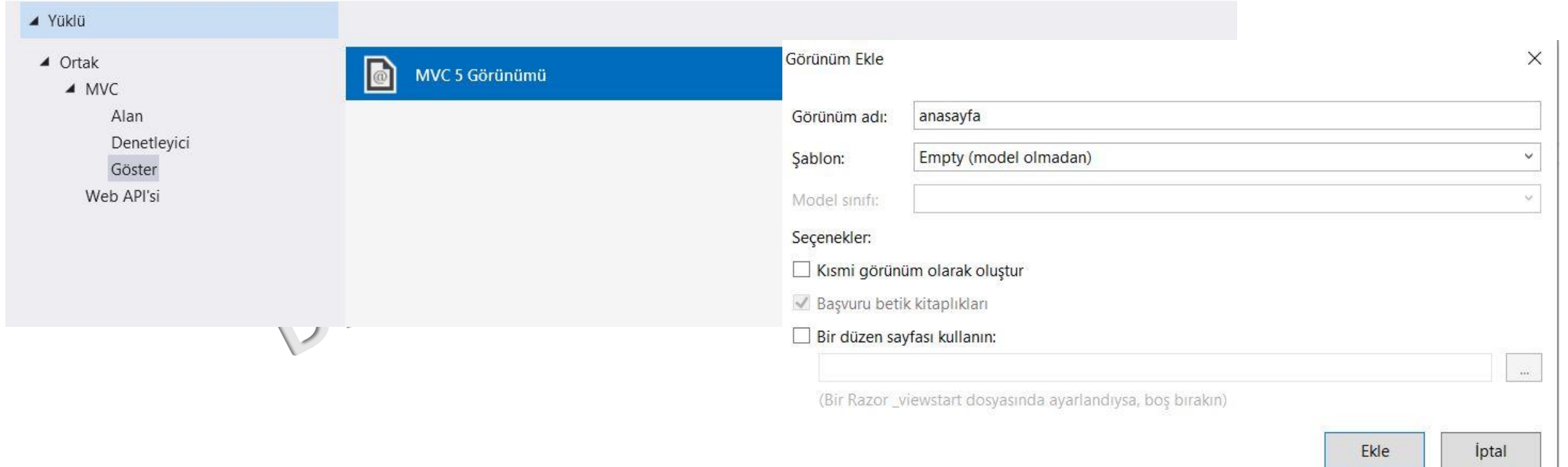
Dr. Öğr. Üyesi

```
namespace denemeucuncuders.Controllers
{
    0 başvuru
    public class HomeController : Controller
    {
        // GET: Home
        0 başvuru
        public ActionResult anasayfa()
        {
            return View();
        }
        0 başvuru
        public ActionResult hakkimizda()
        {
            return View();
        }
        0 başvuru
        public ActionResult iletisim()
        {
            return View();
        }
        0 başvuru
        public ActionResult icicelayoutkullanimi()
        {
            return View();
        }
    }
}
```

Anasayfa ve İletişim controllerlarımız için Viewlarımızı yandaki gibi oluşturalım..



Yeni İskeleli Öğe Ekle



Anasayfa ve iletişim sayfalarının içerisini dolduralım aşağıdaki gibi...

```
iletisim.cshhtml  anasayfa.cshhtml  HomeController.cs
1
2  @{
3      Layout = null;
4  }
5  <!DOCTYPE html>
6  <html>
7  <head>
8      <meta name="viewport" content="width=device-width" />
9      <title>anasayfa</title>
10 </head>
11 <body>
12     <div>
13         <h2>anasayfa</h2>
14         <p>
15             Fatma AKALIN Fatma AKALIN Fatma AKALIN Fatma AKALIN
16             Fatma AKALIN Fatma AKALIN Fatma AKALIN Fatma AKALIN
17             Fatma AKALIN Fatma AKALIN Fatma AKALIN Fatma AKALIN
18             Fatma AKALIN Fatma AKALIN Fatma AKALIN Fatma AKALIN
19             Fatma AKALIN Fatma AKALIN Fatma AKALIN Fatma AKALIN
20             Fatma AKALIN Fatma AKALIN Fatma AKALIN Fatma AKALIN
21             Fatma AKALIN Fatma AKALIN Fatma AKALIN Fatma AKALIN
22             Fatma AKALIN Fatma AKALIN Fatma AKALIN Fatma AKALIN
23             Fatma AKALIN Fatma AKALIN Fatma AKALIN Fatma AKALIN
24             Fatma AKALIN Fatma AKALIN Fatma AKALIN Fatma AKALIN
25             Fatma AKALIN Fatma AKALIN Fatma AKALIN Fatma AKALIN
26             Fatma AKALIN Fatma AKALIN Fatma AKALIN Fatma AKALIN
27         </p>
28     </div>
29 </body>
30 </html>
```

```
iletisim.cshhtml  anasayfa.cshhtml  HomeController.cs
1
2  @{
3      Layout = null;
4  }
5
6  <!DOCTYPE html>
7
8  <html>
9  <head>
10     <meta name="viewport" content="width=device-width" />
11     <title>iletisim</title>
12 </head>
13 <body>
14     <div>
15     </div>
16 </body>
17 </html>
18
```

Şimdi layout aşamasına geçelim.

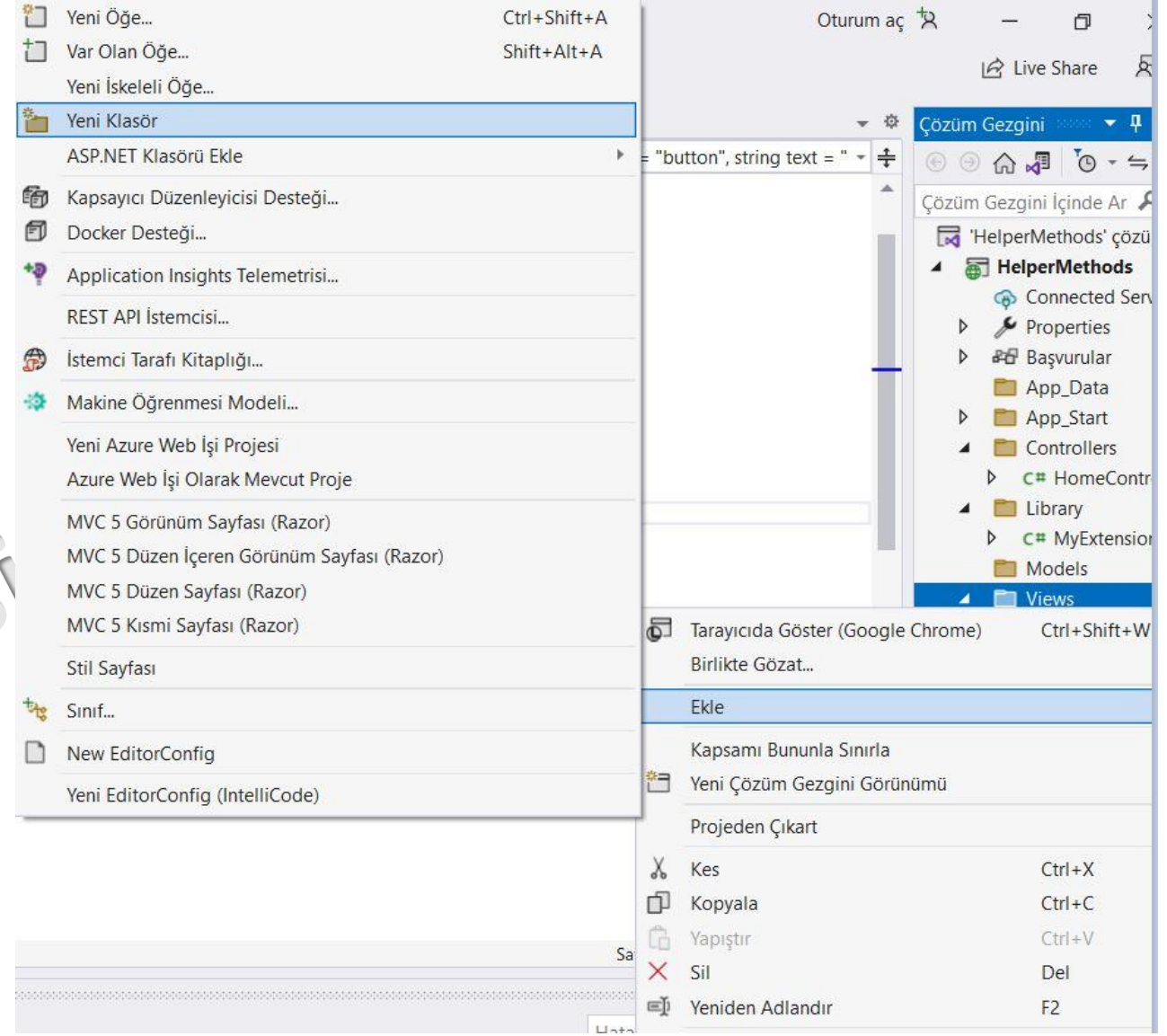
Örneğimizde, Home Controller oluşturdum ve içerisinde 3 ActionResult tipinde metot yazdım. İletişim ve anasayfa kısımlarında herhangi bir layout kullanmaksızın bu iki sayfayı oluşturdum. Fakat diğer iki metotta layout kullanmaya karar verdim. Layoutu şu şekilde açıklayabilirim.

Web sitelerimizin sabit ve değişken kısımları vardır. Layout kullanacaksak öncelikle sabit olan kısımlarımızı tutan bir HTML sayfa oluşturmamız gerekiyor. Yani Bu sabit olan sayfaları tutan kısımlara layout diyoruz. Layoutlar view klasörünün altında genelde shared isimli paylaşılmış anlamına gelen bir klasörde tutulur.

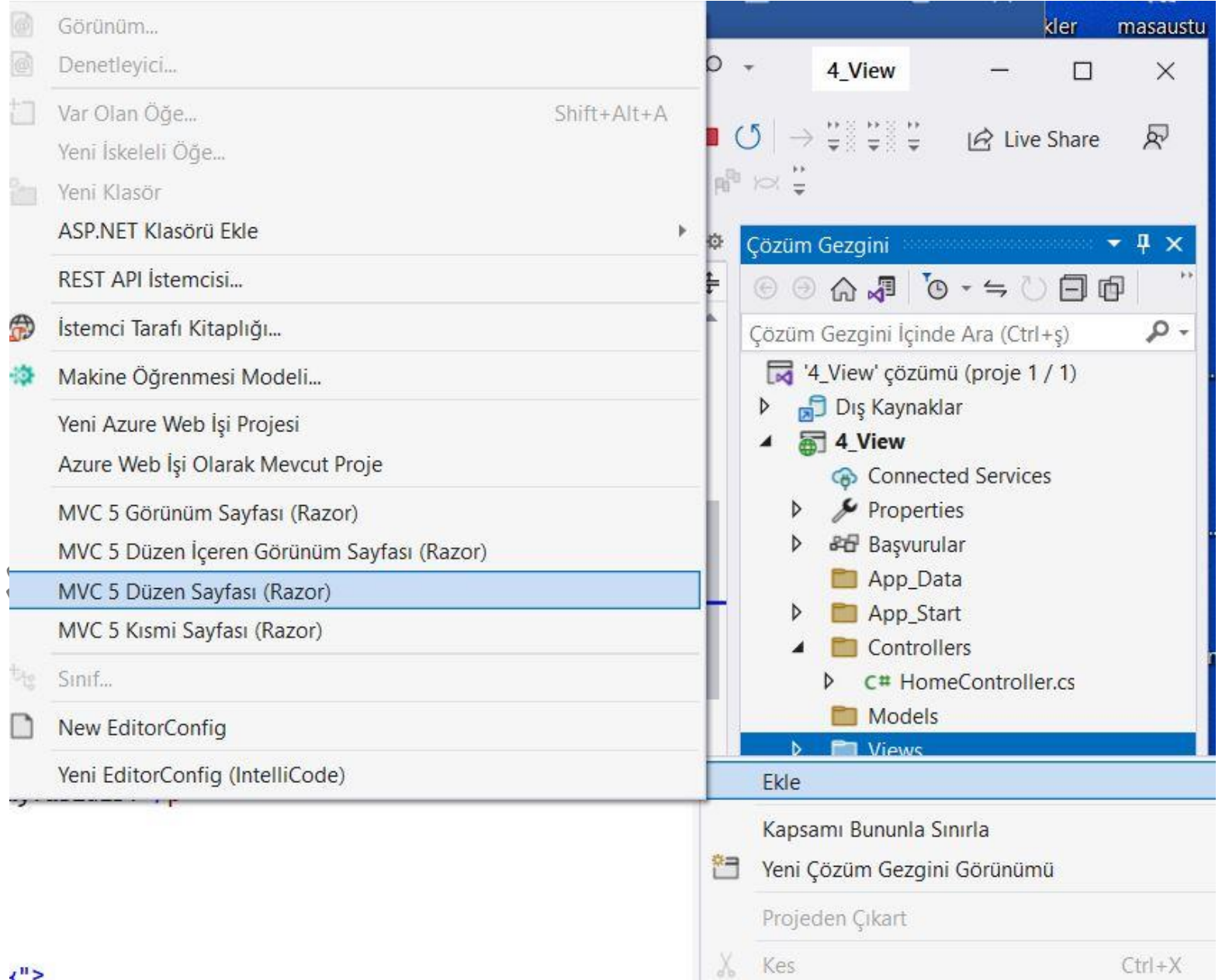
LAYOUT OLUŖTURMA

Öncelikle klasör
oluŖturuyoruz

Dr. Öğr. Üyesi

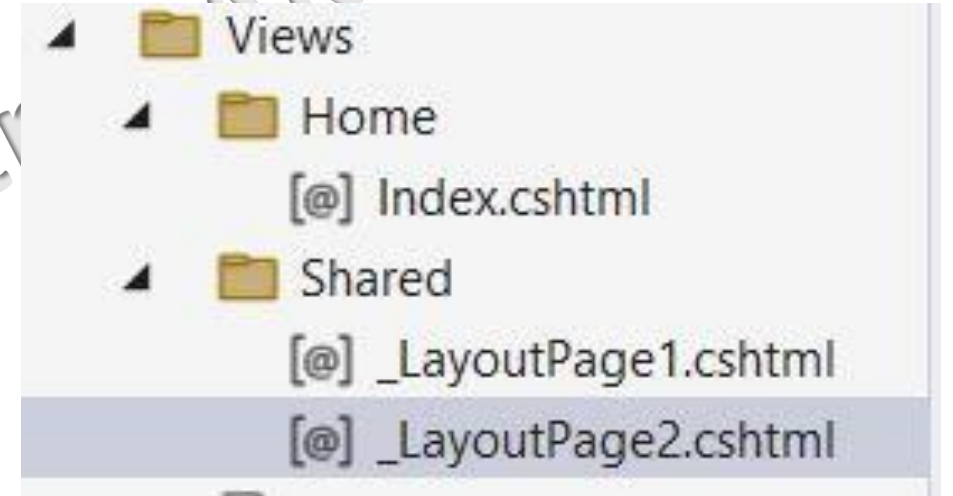


Bu klasörün
içerisine bir tane
layout page
ekleyeceğim.
Ekle ve ardından
MVC5 Layout
Page(Razor)
kısımından bir
ekleme
yapabilirsin.



Layout 1 ve
Layout 2
sayfalarının
içeriğini web
teknolojileri
dersinde
aldığımız temel
ile inşa ediyoruz.

KLASÖR YAPISI



Şimdi diğer hakkımızda View'ı oluştururken Layout2 sayfasını dikkate alacağız.

Görünüm Ekle

Görünüm adı:

hakkimizda

Şablon:

Empty (model olmadan)

Model sınıfı:

Seçenekler:

☐ Kısmi görünüm olarak oluştur

☐ Başvuru betik kitaplıkları

☒ Bir düzen sayfası kullanın:

~/Views/Shared/_LayoutPage2.cshtml

(Bir Razor _viewstart dosyasında ayarlandıysa, boş bırakın)

Ekle

İptal

Layout sayfalarını incelersek, tüm sayfaar için değişken olan bölümler için tanımlamayı C#'ın RenderBody() metodu ile yapıyoruz.

```
</head>
<body>
  <div id="menu" class="block">
    @*Header*@
    <ul>
      <li>@Html.ActionLink("Anasayfa", "anasayfa", "Home")</li>
      <li>@Html.ActionLink("Hakkımızda ", "hakkimizda", "Home")</li>
      <li>@Html.ActionLink("İletişim", "iletisim", "Home")</li>
      <li>@Html.ActionLink("Nesned Layout", "icicelayoutkullanimi", "Home")</li>
    </ul>
  </div>
  <div class="block">
    @RenderBody()
  </div>
  <div class="block">
    @*Footer*@
    <b>Tüm hakları saklıdır. &copy;</b>
  </div>
</body>
</html>
```


Bu şekilde aynı layoutu istediğiniz sayfalarınıza uygulayabilirsiniz.

View dosyanızın başındaki

`Layout = "~/Views/Shared/_LayoutPage2.cshtml";` ifadesi
istediğiniz sayfalara uygulama işlevselliğini sağlıyor.

Bu projede Controllerda oluşturduğum İcicelayoutkullanımı isimli action metodunu kullanmadım. Ek olarak _layoutPage1.cshtml 'ide kullanmadım. AKLİMİZİ KARŞITIRMASIN!!!

İÇ İÇE(NESTED) LAYOUT OLUŞTURMA

ASP.NET MVC ile yazılım geliştirme sırasında iç içe layout oluşturma ihtiyacı oluşabilir. Peki iç içe layout nedir?





Layoutların birbirinden türeme durumunu ifade eden bir işlevselliktir.

Bir örnek üzerinden açıklayalım...

Dr. Öğr. Üyesi Fatma AKALIN

Yeni bir proje oluşturun

Son proje şablonları

 ASP.NET Web Uygulaması (.NET Framework)	C#
 Windows Forms Uygulaması (.NET Framework)	C#
 ASP.NET Core Web Uygulaması (Model-Görünüm-Denetleyici)	C#
 Konsol Uygulaması	C++

Yeni projenizi yapılandırın

ASP.NET Web Uygulaması (.NET Framework) C# Windows Bulut Web

Proje adı

deneme_dorduncuders

Konum

D:\aspnet\1_ADERSLER

Çözüm adı ⓘ

deneme_dorduncuders

☐ Çözümü ve projeyi aynı dizine yerleştirin

Altyapı

.NET Framework 4.6.2

Proje "D:\aspnet\1_ADERSLER\deneme_dorduncuders\deneme_dorduncuders\" içinde oluşturulacak

Şablon ara (Alt+S)

Tüm diller

Tüm platformlar

Tüm proje türleri



Konsol Uygulaması

Windows, Linux ve macOS'de .NET üzerinde çalışabilen bir komut satırı uygulaması oluşturma projesi

C#

Linux

macOS

Windows

Konsol



ASP.NET Core Web Uygulaması

ASP.NET Core Razor Sayfalar içeriği örneğiyle ASP.NET Core uygulaması oluşturmak için proje şablonu

C#

Linux

macOS

Windows

Bulut

Hizmet

Web



Blazor Server Uygulaması



Yeni ASP.NET Web Uygulaması oluştur



Boş

ASP.NET uygulamaları oluşturmak için boş bir proje şablonu. Bu şablonda içerik yoktur.



Web Forms

ASP.NET Web Forms uygulamaları oluşturmak için bir proje şablonu. ASP.NET Web Forms, bilinen sürük ve bırak yöntemiyle olay denetimli modeli kullanarak dinamik web siteleri oluşturmaya sağlar. Tasarım yüzeyi ile yüzlerce denetim ve bileşen, veri erişimi olan gelişmiş, güçlü, kullanıcı arabirimi denetimli web sitelerini hızlı bir şekilde oluşturmaya sağlar.



MVC

ASP.NET MVC uygulamaları oluşturmaya yarayan bir proje şablonu. ASP.NET MVC, Model-View-Controller mimarisini kullanarak uygulamalar oluşturmaya olanak sağlar. ASP.NET MVC en son standartları kullanan uygulamalar oluşturmak için hızlı, test güdümlü geliştirmeye olanak sağlayan birçok özellik içerir.



Web API

Tarayıcılar ve mobil cihazlar dahil olmak üzere çok çeşitli istemcilere erişebilen RESTful HTTP hizmetleri oluşturmaya yarayan bir proje şablonudur.



Tek Sayfalı Uygulama

ASP.NET Web API kullanarak JavaScript temelli zengin istemci tarafı HTML5 uygulamaları oluşturmak üzere kullanılan proje şablonudur. Tek Sayfalı Uygulamalar, HTML5, CSS3 ve JavaScript'in kullandığı istemci tarafı etkileşimler içeren zengin bir kullanıcı deneyimi sağlar.

Kimlik Doğrulama

Yok

Klasörleri ve çekirdek başvurularını ekle

☐ Web Forms

☒ MVC

☐ Web API'si

Gelişmiş

☒ HTTPS'yi Yapılandır

☐ Docker desteği

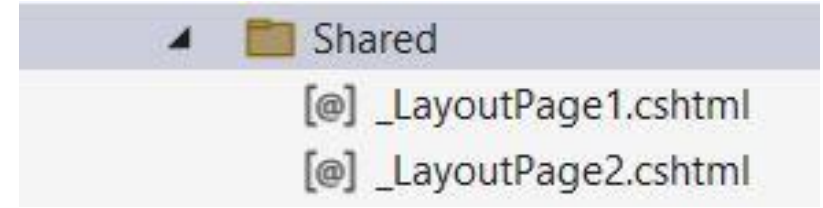
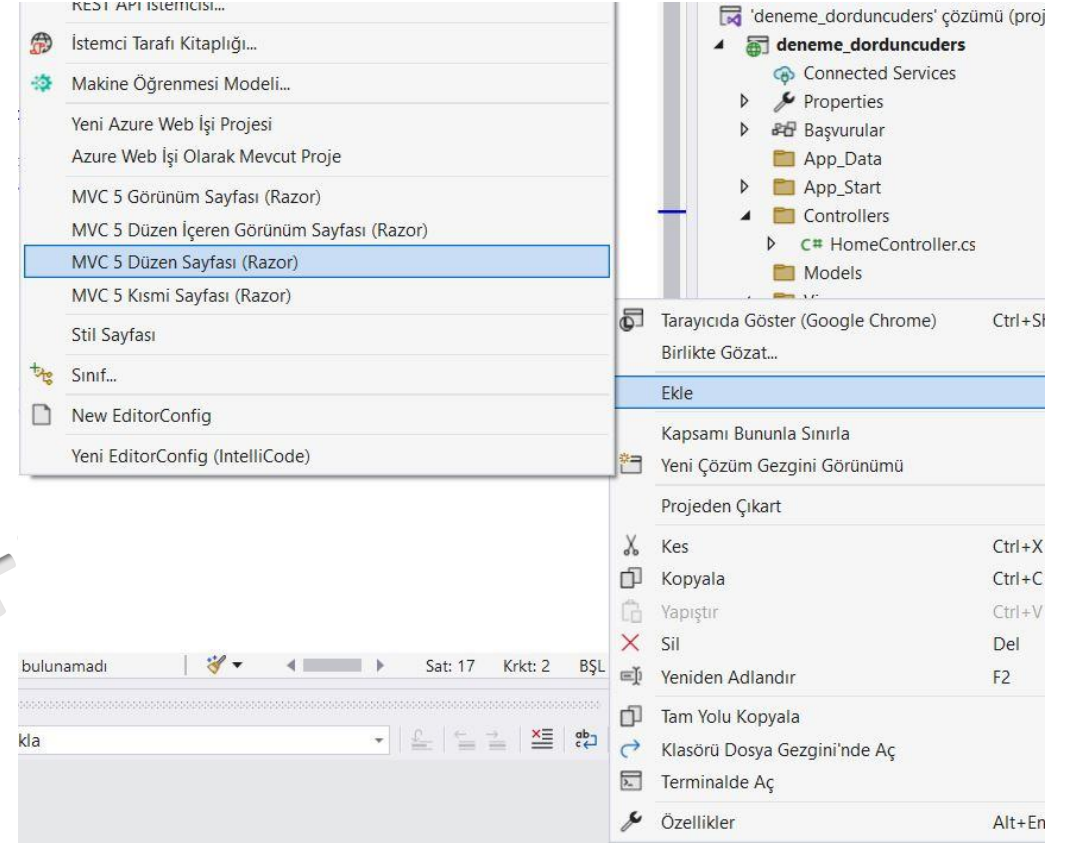
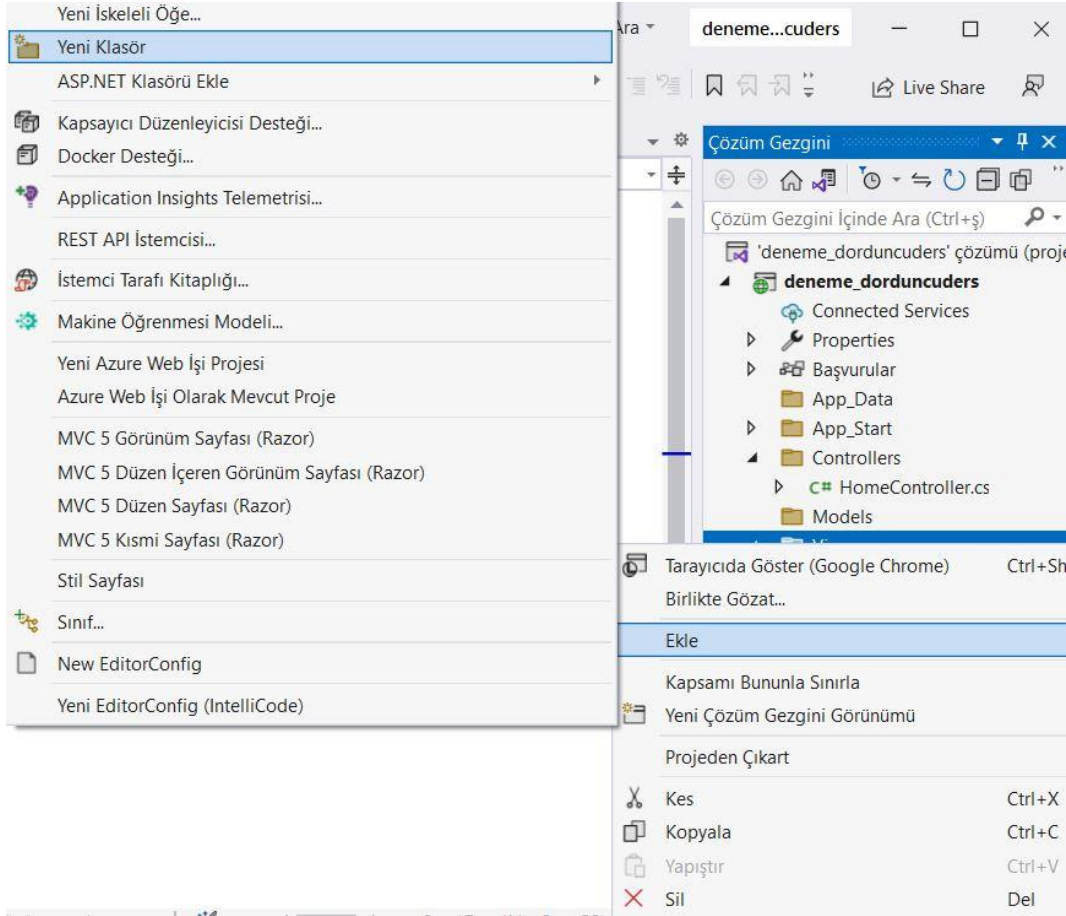
(Docker Desktop gerektiriyor)

☐ Ayrıca birim testleri için bir proje oluştur

deneme_dorduncuders.Tests

Geri

Oluştur



Controller yapımızın içinde oluşturduğumuz 2 metoda view ekleyelim



```
ml _LayoutPage1.cshtml HomeController.cs
Incuders deneme_dorduncuder
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace deneme_dorduncoders.Controllers
{
    0 başvuru
    public class HomeController : Controller
    {
        // GET: Home
        0 başvuru
        public ActionResult Index1()
        {
            return View();
        }
        0 başvuru
        public ActionResult Index2()
        {
            return View();
        }
    }
}
```

Index1 View'ımız, LayoutPage1'i
Index2 View'ımız, LayoutPage2'i kullansın istiyoruz

Görünüm Ekle

×

Görünüm adı:

Index1

Şablon:

Empty (model olmadan)

Model sınıfı:

Seçenekler:

☐ Kısmi görünüm olarak oluştur

☒ Başvuru betik kitaplıkları

☒ Bir düzen sayfası kullanın:

~/Views/Shared/_LayoutPage1.cshtml

...

(Bir Razor _viewstart dosyasında ayarlandıysa, boş bırakın)

Ekle

İptal

Görünüm Ekle

×

Görünüm adı:

Index2

Şablon:

Empty (model olmadan)

Model sınıfı:

Seçenekler:

☐ Kısmi görünüm olarak oluştur

☐ Başvuru betik kitaplıkları

☒ Bir düzen sayfası kullanın:

~/Views/Shared/_LayoutPage2.cshtml

...

(Bir Razor _viewstart dosyasında ayarlandıysa, boş bırakın)

Ekle

İptal

Eğitim amaçlı olarak tasarlanan bu örnekleri temel alarak geçmiş yıl kazanmış olduğunuz frontend bilgisi ile daha güçlü layoutlar tasarlayabilirsiniz. Ek olarak zaman kalırsa bende son haftalarda sizlere verdiğim proje ödevine katkıda bulunması amacı ile frontend bilgilerinizi güçlendirmek amacıyla frontend projeleri anlatıyor olacağım

KAYNAKÇA

<https://learn.microsoft.com/tr-tr/aspnet/mvc/>

Veysel Uğur Kızmaz, ASP.NET MVC5, Kodlab Yayınları

<https://github.com/muratbaseren/udemy-yazilimcilarin-yukselisi-aspnet-my-evernote-sample>

Dr. Öğr. Üyesi Fatma AKALIN