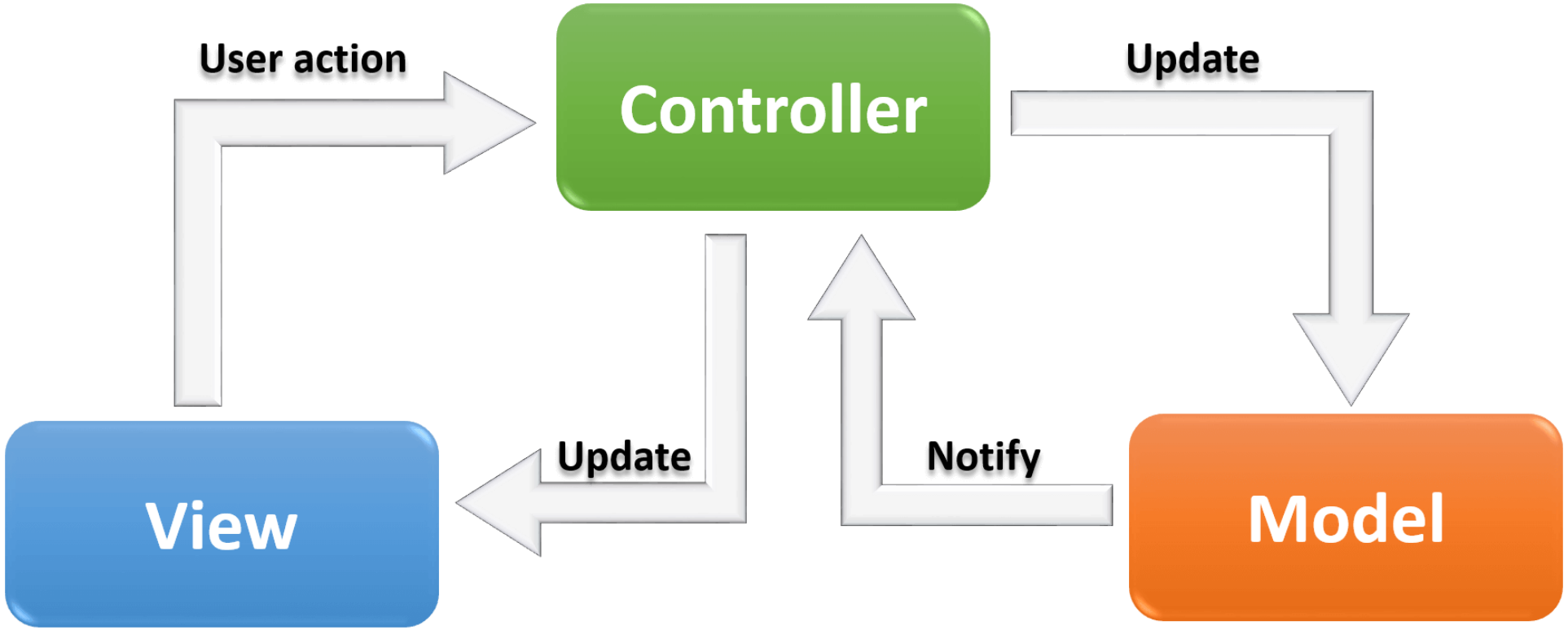
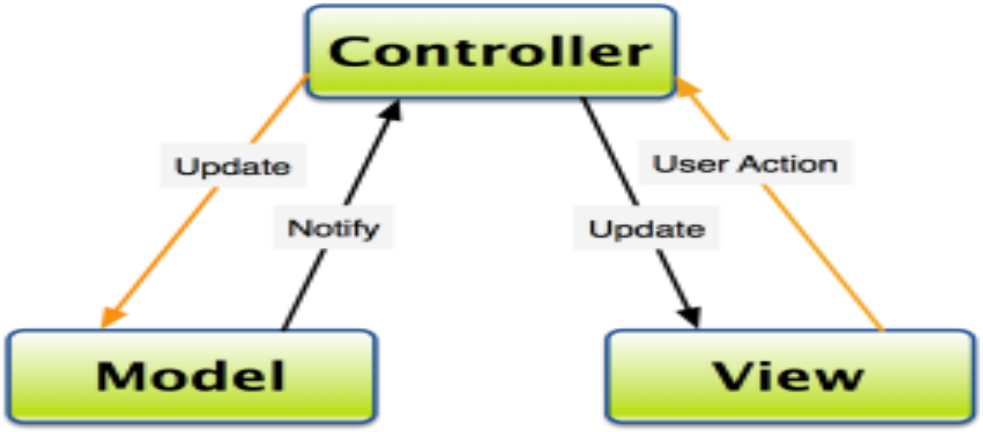
ASP.NET(C#) MVC 5





Models: Kullanacağınız class'lar bulunur. Bu katmanda database'e erişim yollarınız ve metotlarınız yer almaktadır. Bu katman data erişim katmanı(access layer) olarak Entity Framework, ADO.NET ile database üzerinde veri işlemleri yapılması için kullanılır.

View: Basitçe, uygulamanızın kullanıcılarınızın gözüyle gördüğü kısmıdır, arayüzdür. Uygulanmanın arayüz tasarımı burada tasarlanır, projenin html dosyaları burada bulunur.

Controller: View ve Model arasında ki iletişimi sağlar.
Kullanıcıların View üzerinden gerçekleştirdiği gelen istekleri (request) Model'e taşır, Model'den aldığı veriyi View üzerinden kullanıcıya gösterir.

MVC’nin Çalışma Mantığı

- Şimdi bu durumu örneklendirecek olursak eğer siz Instagram da bir kullanıcının profilini görmek için istek(request) göndermeniz durumunda, router dosyasında bu istek kontrol edilir.

- Kontrol edildikten sonra controller dosyasında profile ait fonksiyon çalışır. Fonksiyonda gerekli veritabanı işleri yapılır .

- Ardından sizi o fonksiyona ait view’i yani kullanıcının profilini cevap(response) olarak geri döndürür.

View Engine Nedir? Razor Nedir?

View Engine

View'lerinizi HTML çıktısı olarak render etmek için kullanılan bir mekanizma/teknoloji'dir.

Razor View Engine

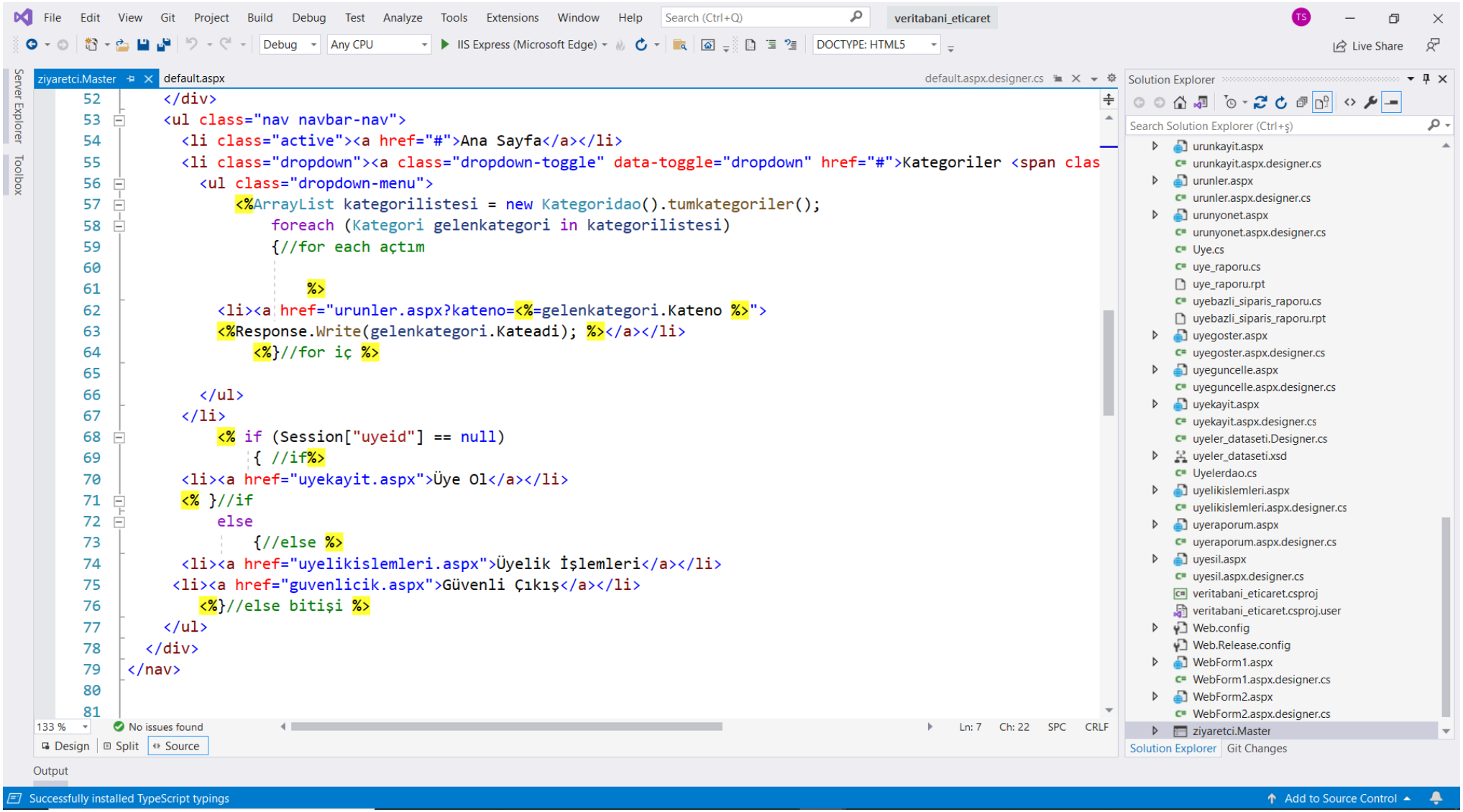
Razor söz dizimi, programcının bir HTML yapı akışını kullanabilmesine olanak tanıyan ve C# tabanlı bir şablon etiketleme söz dizimidir.

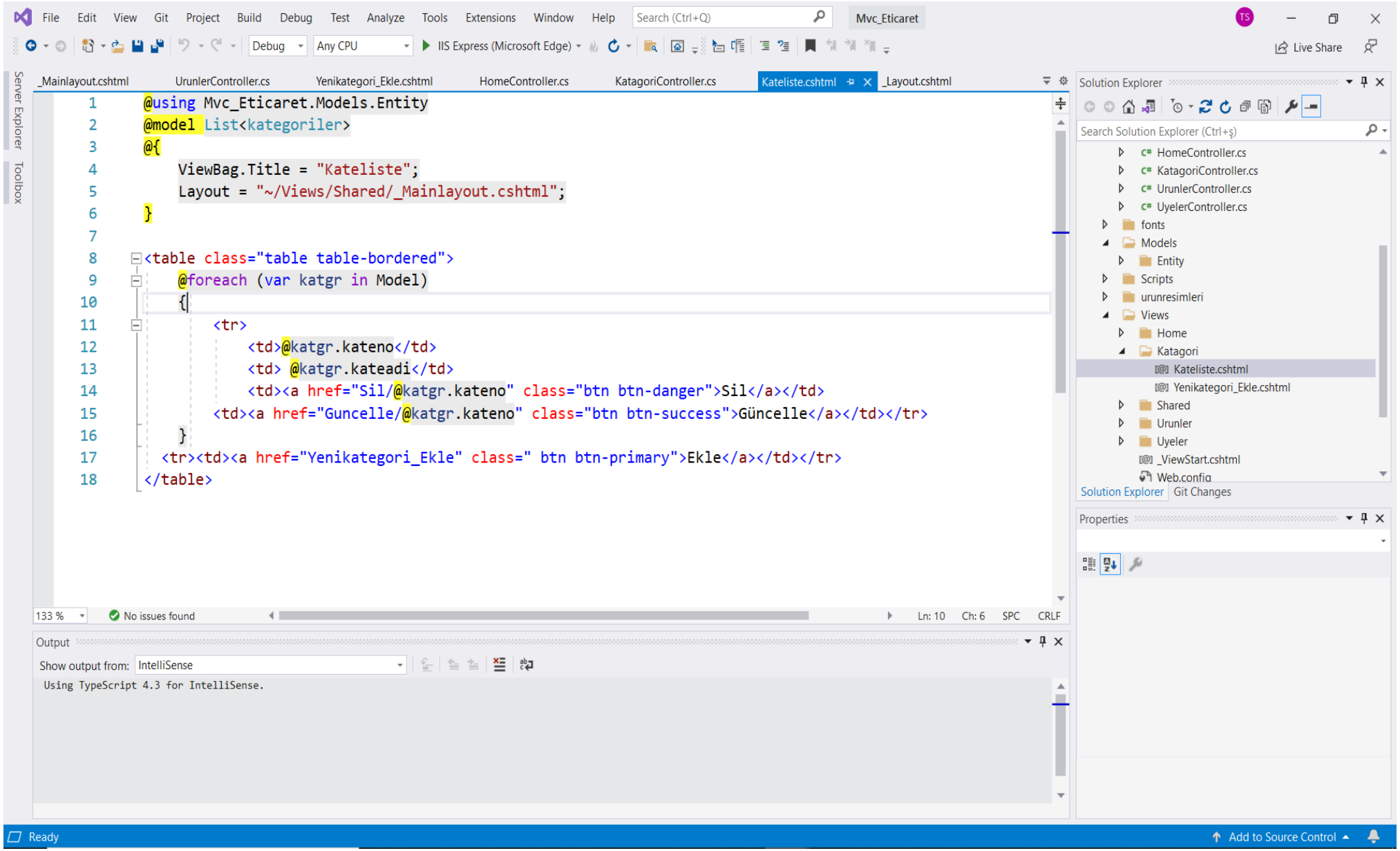
MVC3 ile birlikte Razor çıkana kadar MVC 1 ve 2'de defaulf olarak kullanılan Web Forms View Engine'di. Çalışmasında herhangi bir sorun yoktu fakat yazımı, kullanımı ve bakımı uğraştırıcı ve zahmetliydi. Razor'la artık çok güçlü bir View Engine kavuştuk.

Razor View Engine'in yazım kurallarının temelinde @ sembolü bulunmaktadır.
Asp.Net Web Form ile kodlama yapılırken HTML kodlarının arasında yazılan sunucu taraflı kodlamada <% … %> yazımı kullanılmaktadır.

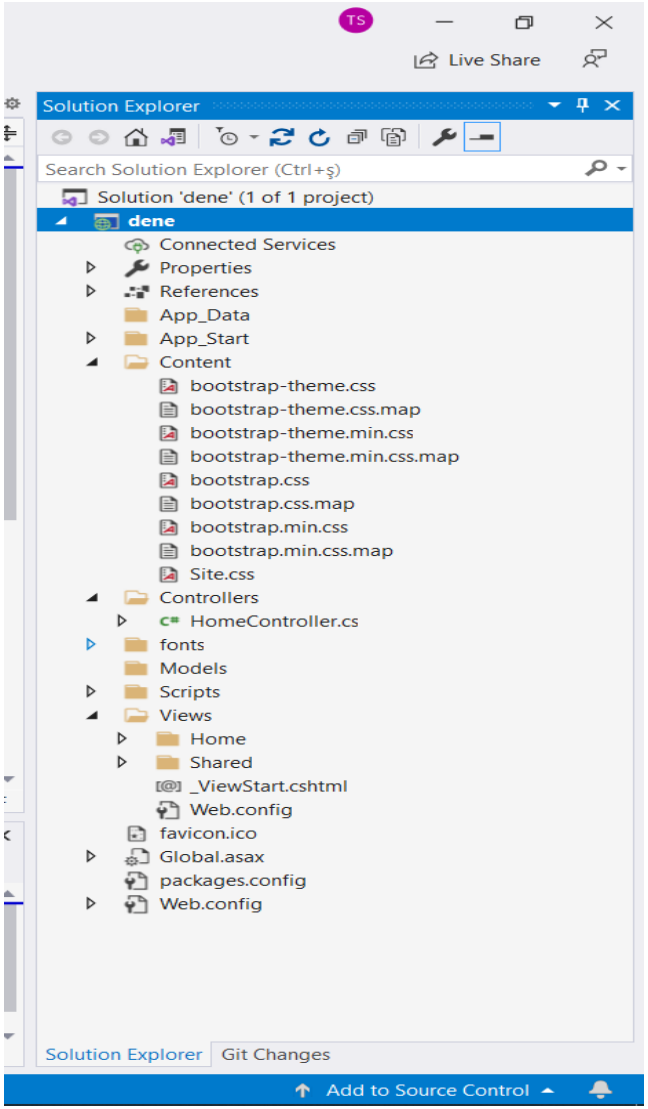
Razor için bu yapının yerine @ sembolü kullanılmaktadır. Yazım olarak Web Form'a göre oldukça kolaydır ve Asp.Net MVC'de kodlama daha hızlı olmaktadır.

Web Forms kullanımı ve Razor kullanımına örnek:




File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
Mvc_Eticaret
Debug
Any CPU
IIS Express (Microsoft Edge)
Live Share
_Mainlayout.cshtml
UrunlerController.cs
Yenikategori_Ekle.cshtml
HomeController.cs
KatagoriController.cs
Kateliste.cshtml
_Layout.cshtml
@using Mvc_Eticaret.Models.Entity
@model List<kategoriler>
@{
ViewBag.Title = "Kateliste";
Layout = "~/Views/Shared/_Mainlayout.cshtml";
}
<table class="table table-bordered">
@foreach (var katgr in Model)
{
<tr>
<td>@katgr.kateno</td>
<td> @katgr.kateadi</td>
<td>Sil</td>
<td>Güncelle</td></tr>
}
<tr><td>Ekle</td></tr>
</table>
133 %
No issues found
Ln: 10 Ch: 6 SPC CRLF
Output
Show output from: IntelliSense
Using TypeScript 4.3 for IntelliSense.
Solution Explorer
Search Solution Explorer (Ctrl+ş)
HomeController.cs
KatagoriController.cs
UrunlerController.cs
UyelerController.cs
fonts
Models
Entity
Scripts
urunresimleri
Views
Home
Katagori
Kateliste.cshtml
Yenikategori_Ekle.cshtml
Shared
Urunler
Uyeler
_ViewStart.cshtml
Web.config
Solution Explorer
Git Changes
Properties
Ready
Add to Source Control

Proje İçerisinde Öğelerin Anlamları Neler?



Controllers: URL isteklerini handle eden Controller class'larınızın bulunduğu klasördür.

Models: Data ve Business(iş) objelerinizi manipule(işleyen) eden ve temsil eden class'larınızın bulunduğu klasördür

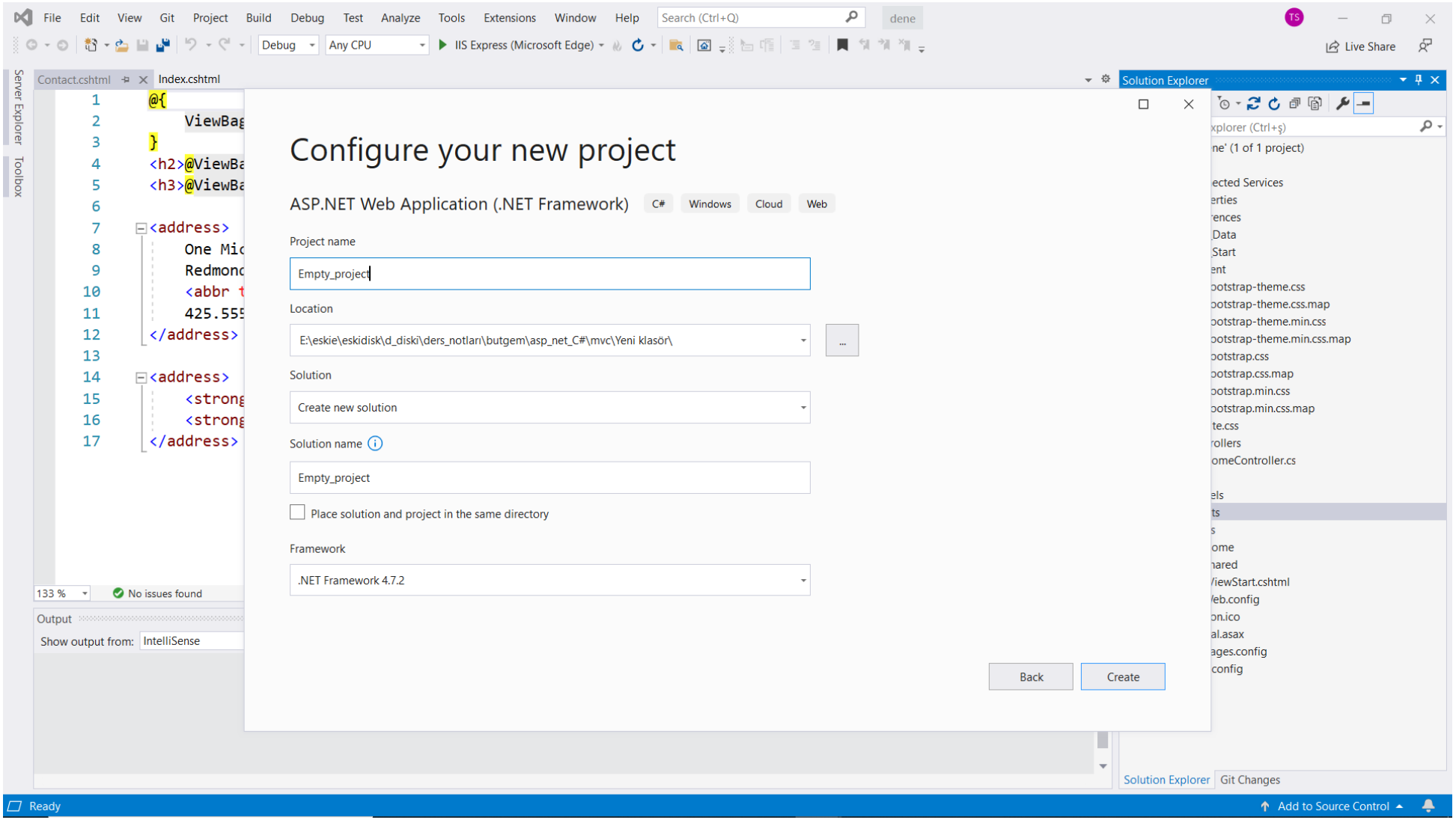
Views: HTML çıktısı olarak karşımıza çıkan View ve Partial View'leri tutar. İçerisinde Controller'in isminde bir klasör olur ve içerisinde View'ler vardır. Ayrıca Shared klasörü adı altında layout'lar tutulur, ASP.NET teki master page sayfaların işlevi yaparlar. Views içerisindeki Web.Config default Web.Config ten farklı olarak bu viewlere doğrudan erişimi engeller yani Controller'sız çalıştırmasını engeller.

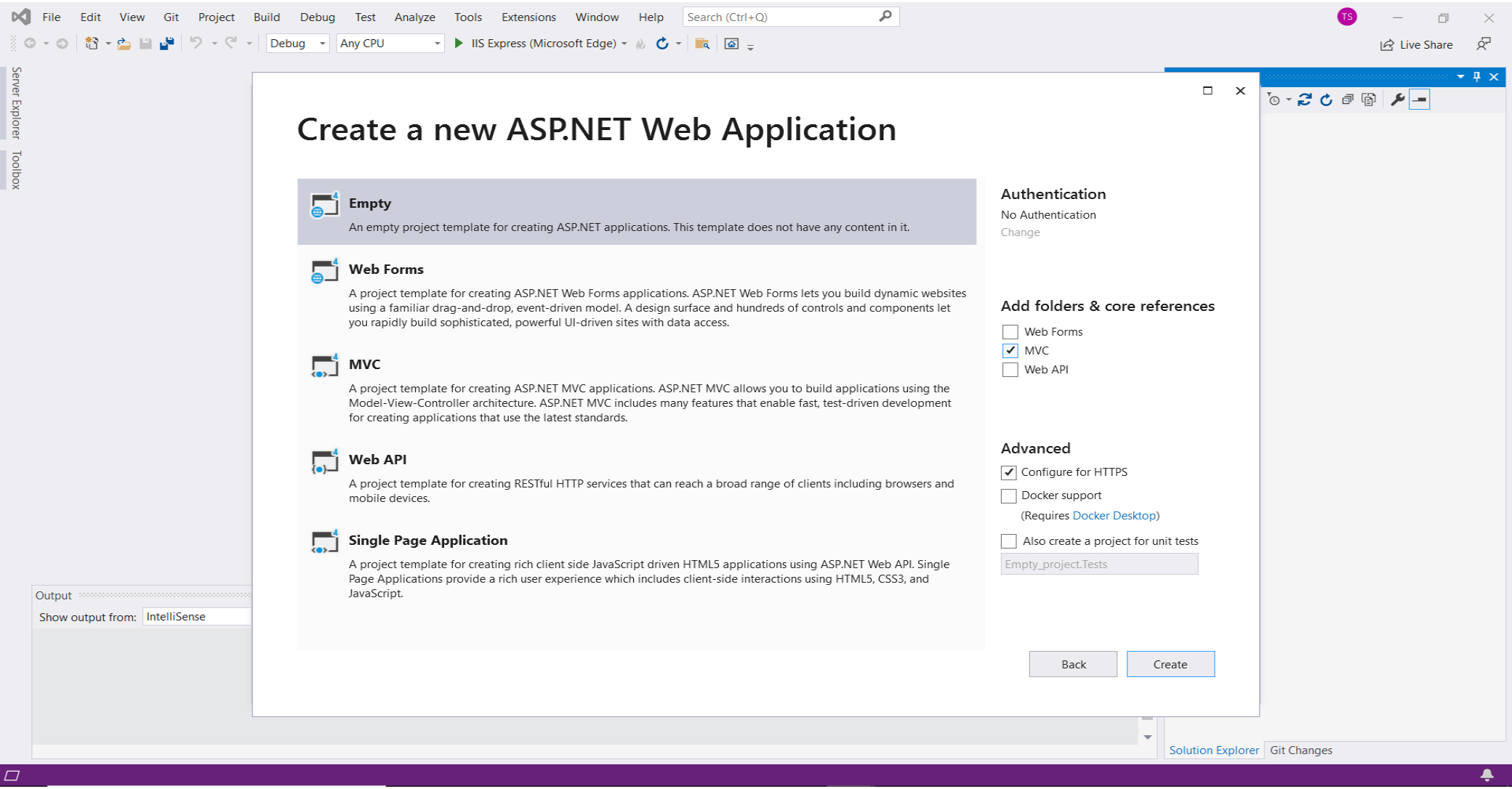
App_Data: Okuma/yazma işlemi yapacağız dosyaların bulunduğu klasördür.

Content: Dinamik olarak değişmeyecek olan ve JavaScript olmayan, css/image dosyalarınız gibi statik dosyaları tutabileceğiniz klasördür

Scripts: Bu klasör içerisinde, Script'ler ve JavaScript kütüphaneleri tutulur.

İlk Uygulama (Empty Template)



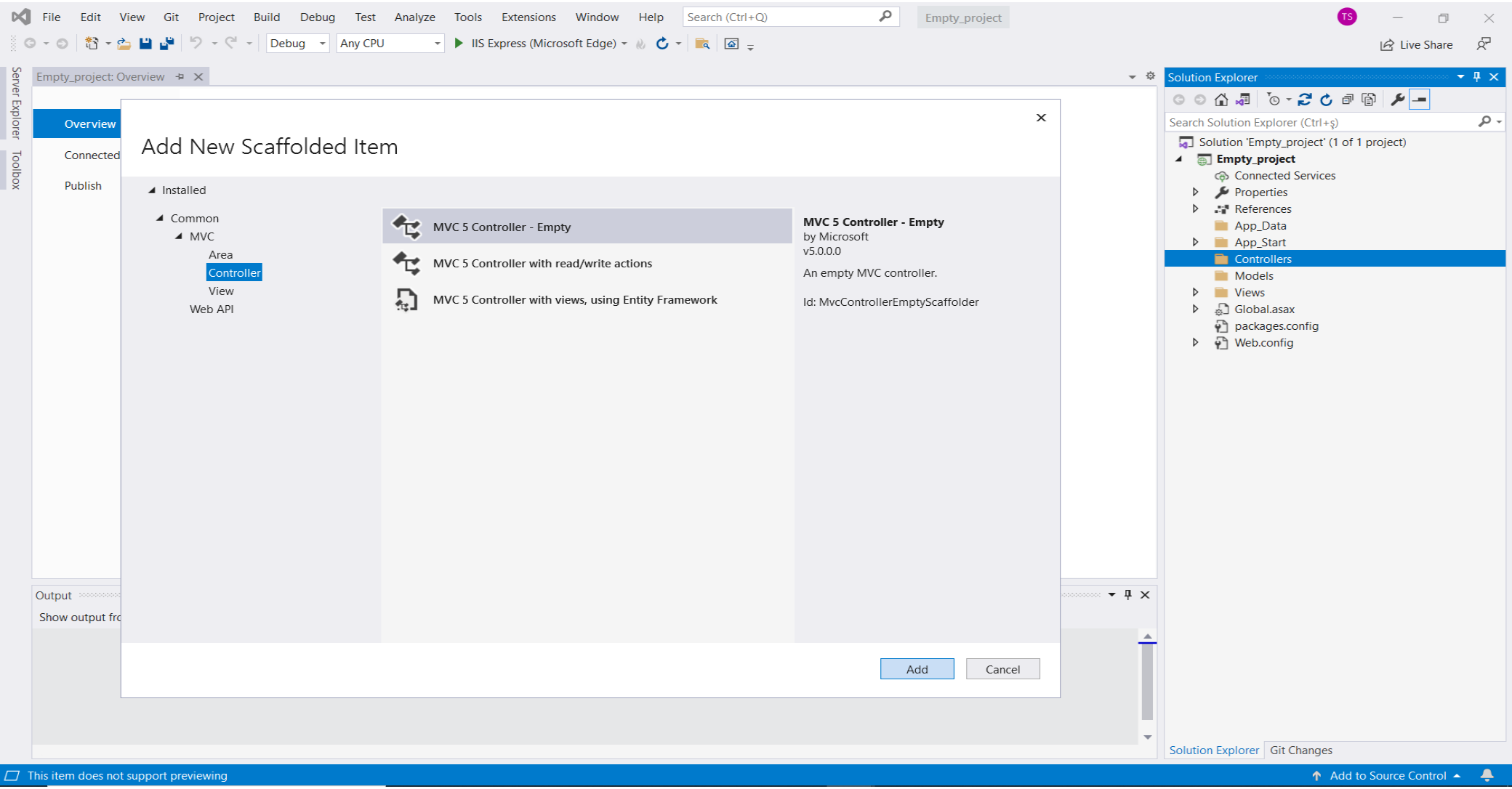

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
Debug
Any CPU
IIS Express (Microsoft Edge)
Live Share
Server Explorer
Toolbox
Create a new ASP.NET Web Application
Empty
An empty project template for creating ASP.NET applications. This template does not have any content in it.
Web Forms
A project template for creating ASP.NET Web Forms applications. ASP.NET Web Forms lets you build dynamic websites using a familiar drag-and-drop, event-driven model. A design surface and hundreds of controls and components let you rapidly build sophisticated, powerful UI-driven sites with data access.
MVC
A project template for creating ASP.NET MVC applications. ASP.NET MVC allows you to build applications using the Model-View-Controller architecture. ASP.NET MVC includes many features that enable fast, test-driven development for creating applications that use the latest standards.
Web API
A project template for creating RESTful HTTP services that can reach a broad range of clients including browsers and mobile devices.
Single Page Application
A project template for creating rich client side JavaScript driven HTML5 applications using ASP.NET Web API. Single Page Applications provide a rich user experience which includes client-side interactions using HTML5, CSS3, and JavaScript.
Authentication
No Authentication
Change
Add folders & core references
Web Forms
MVC
Web API
Advanced
Configure for HTTPS
Docker support
(Requires Docker Desktop)
Also create a project for unit tests
Empty_project.Tests
Back
Create
Output
Show output from: IntelliSense
Solution Explorer
Git Changes

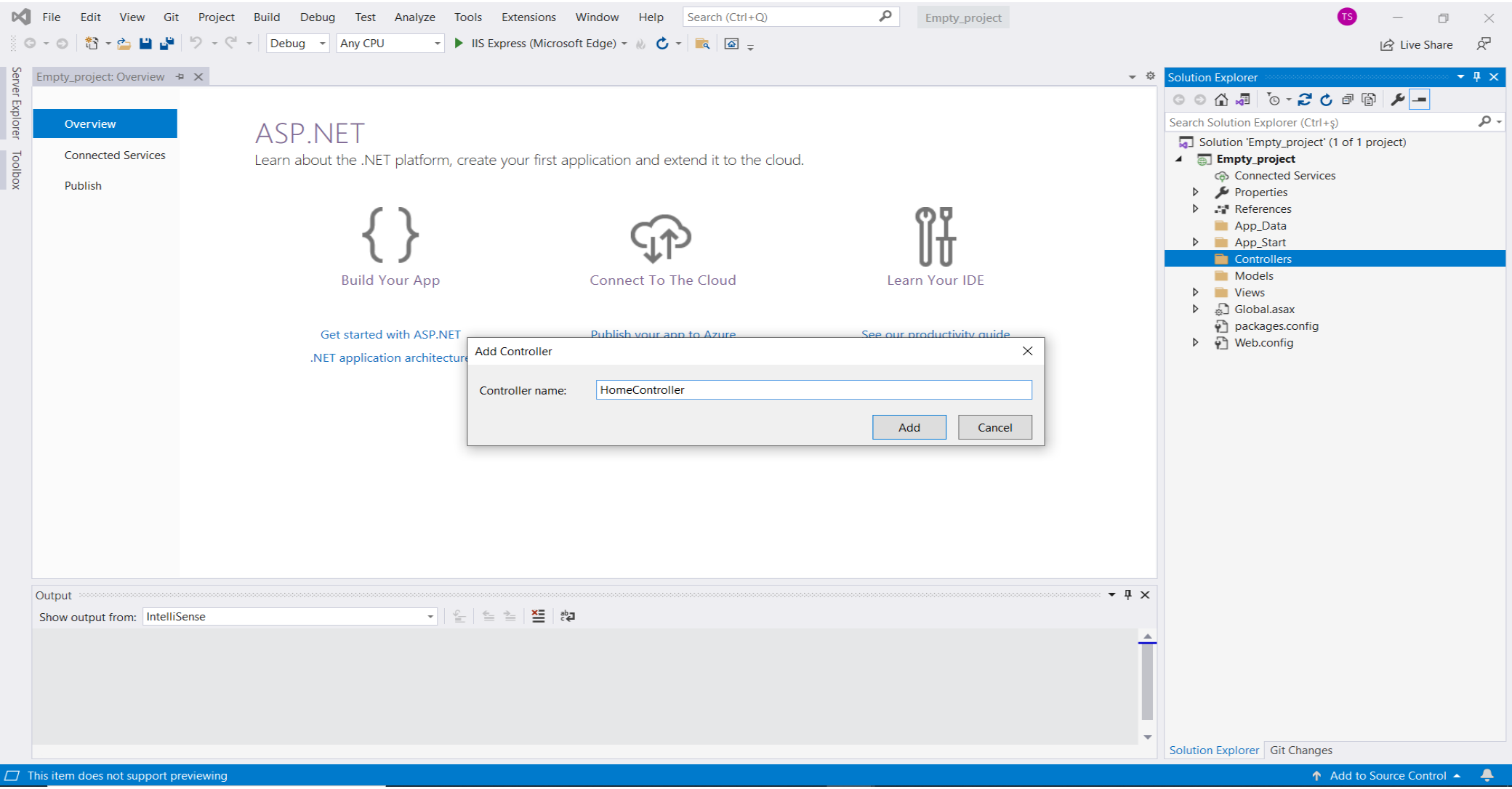
- Biz uygulamamı çalıştırıyoruz fakat uygulamamızın hat veriyor!!!!? Kullanıcının gördüğü dosyalar “View“‘lerdir.

- View oluşturmadan önce “Controller” oluşturmak gerekiyor. Çünkü View’ler Controller den oluşturulur ve çağırılır.

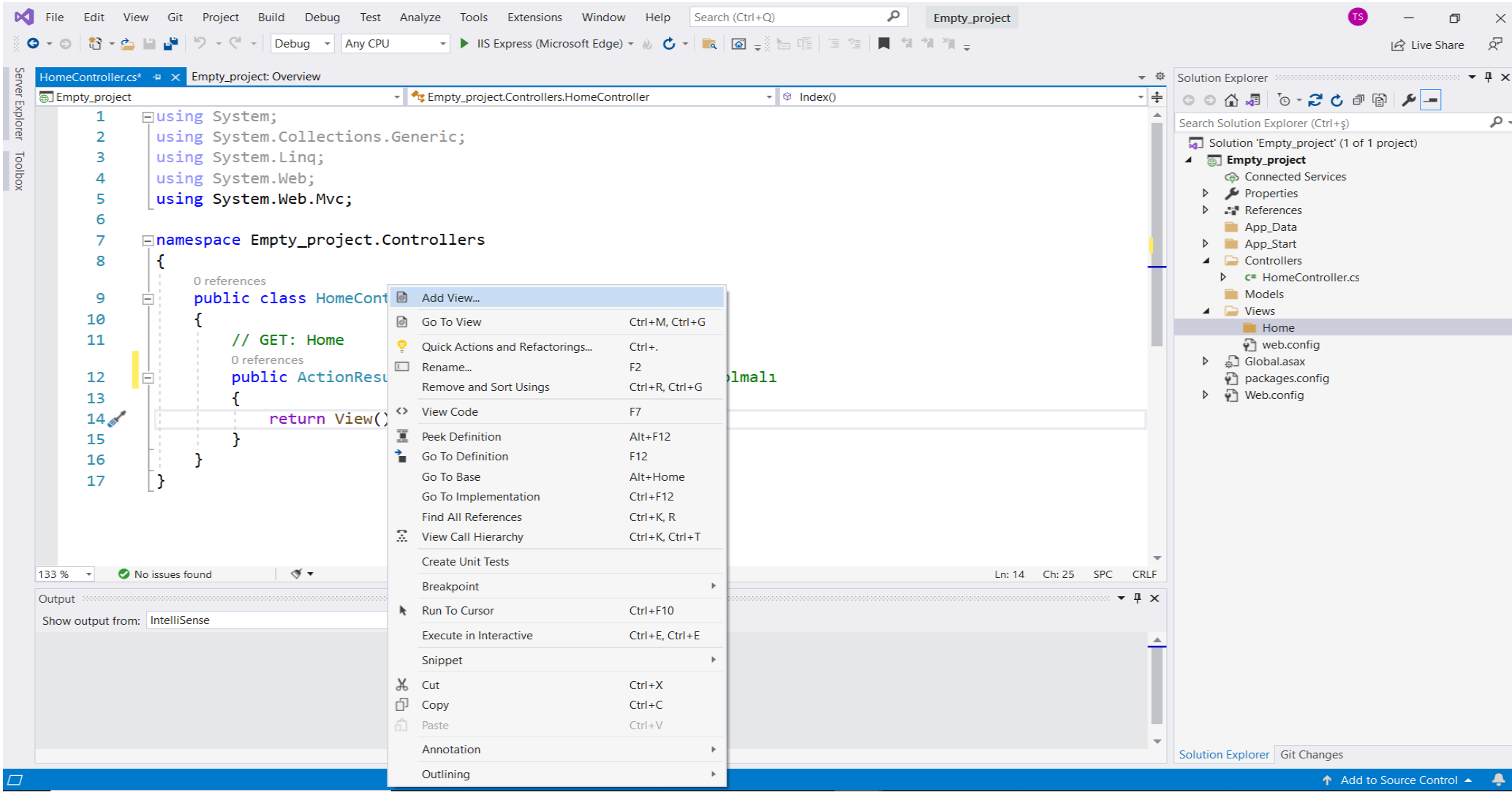
- Bir Controller olmadan View’inde kullanılması düşünülemez.

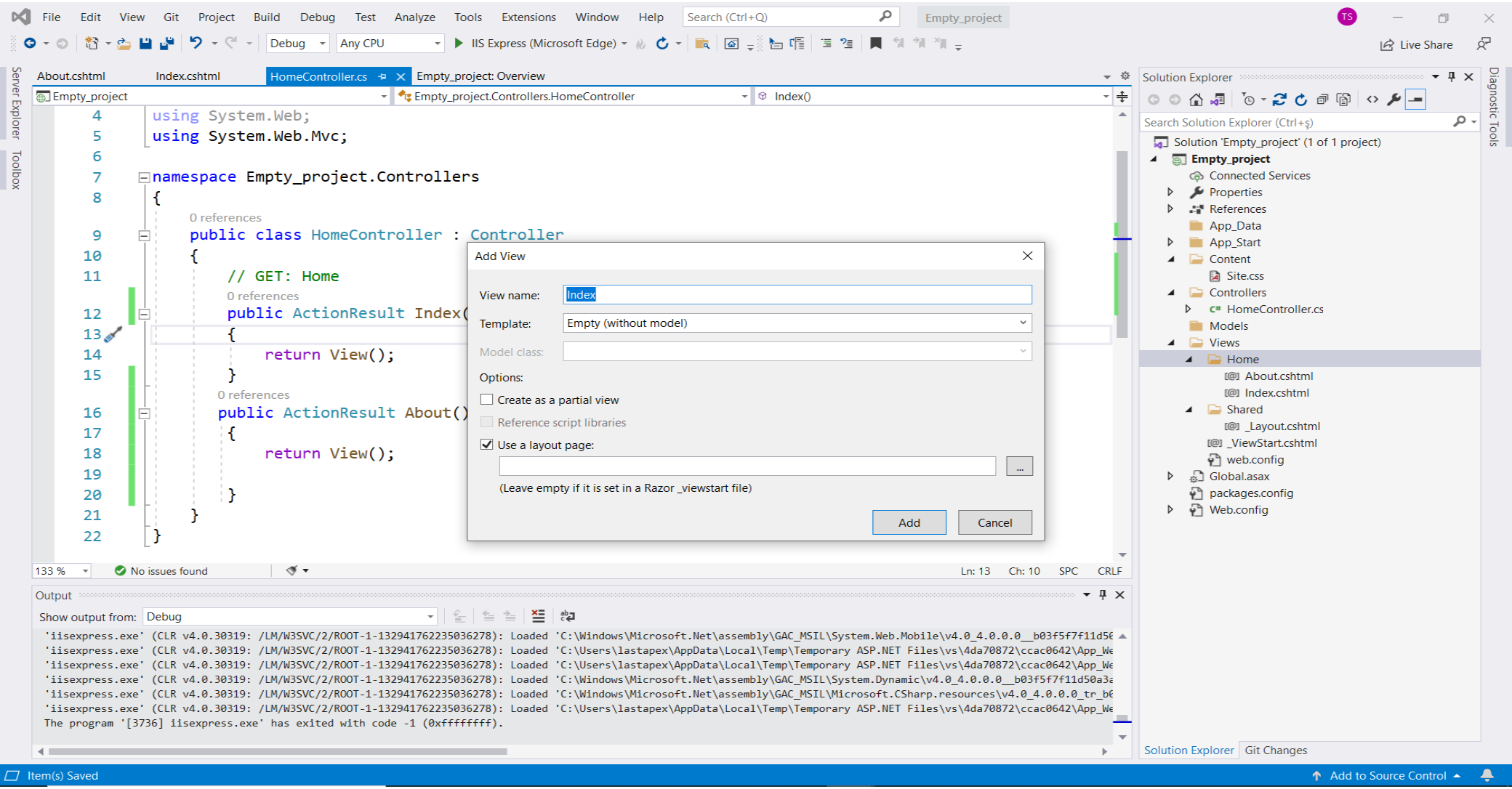
- Uygulamamızda “Controllers” klasörüne sağ tıklayıp Add daha sonra da Controller diyoruz.


File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
Empty_project
Debug
Any CPU
IIS Express (Microsoft Edge)
Live Share
Empty_project: Overview
Overview
Connected
Publish
Add New Scaffolded Item
Installed
Common
MVC
Area
Controller
View
Web API
MVC 5 Controller - Empty
MVC 5 Controller with read/write actions
MVC 5 Controller with views, using Entity Framework
MVC 5 Controller - Empty
by Microsoft
v5.0.0.0
An empty MVC controller.
Id: MvcControllerEmptyScaffolder
Add
Cancel
Output
Show output fro
Solution Explorer
Search Solution Explorer (Ctrl+ş)
Solution 'Empty_project' (1 of 1 project)
Empty_project
Connected Services
Properties
References
App_Data
App_Start
Controllers
Models
Views
Global.asax
packages.config
Web.config
Solution Explorer
Git Changes
This item does not support previewing
Add to Source Control

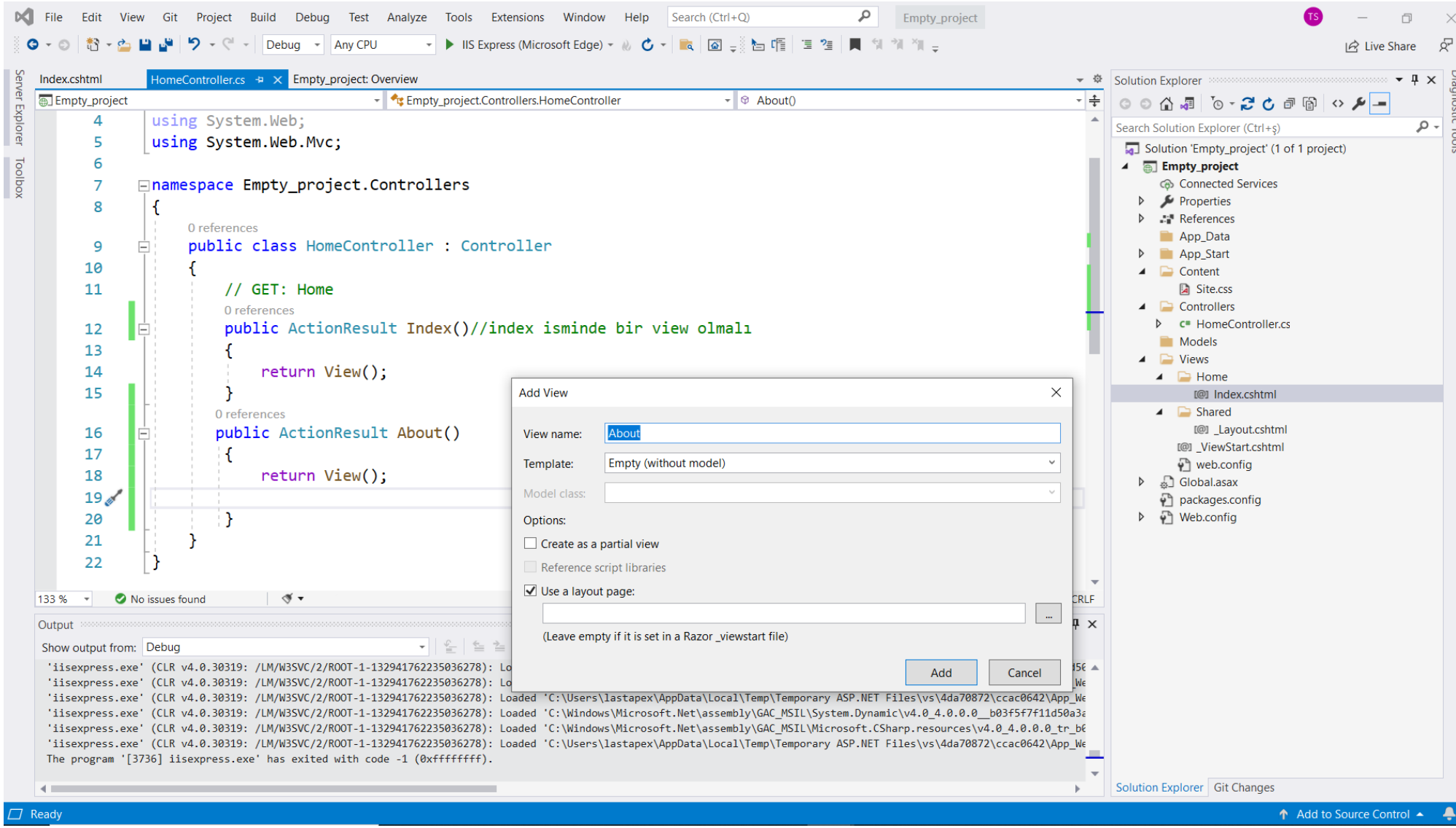

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
Empty_project
Debug
Any CPU
IIS Express (Microsoft Edge)
Live Share
Empty_project: Overview
Overview
Connected Services
Publish
ASP.NET
Learn about the .NET platform, create your first application and extend it to the cloud.
Build Your App
Connect To The Cloud
Learn Your IDE
Get started with ASP.NET
.NET application architecture
Publish your app to Azure
See our productivity guide
Add Controller
Controller name:
HomeController
Add
Cancel
Solution Explorer
Search Solution Explorer (Ctrl+ş)
Solution 'Empty_project' (1 of 1 project)
Empty_project
Connected Services
Properties
References
App_Data
App_Start
Controllers
Models
Views
Global.asax
packages.config
Web.config
Output
Show output from: IntelliSense
Solution Explorer
Git Changes
This item does not support previewing
Add to Source Control

HomeController'im içerisindeki Index adındaki ActionResult metotu içerisinde herhangi bir yer sağ tıklayıp "Add View" diyoruz.

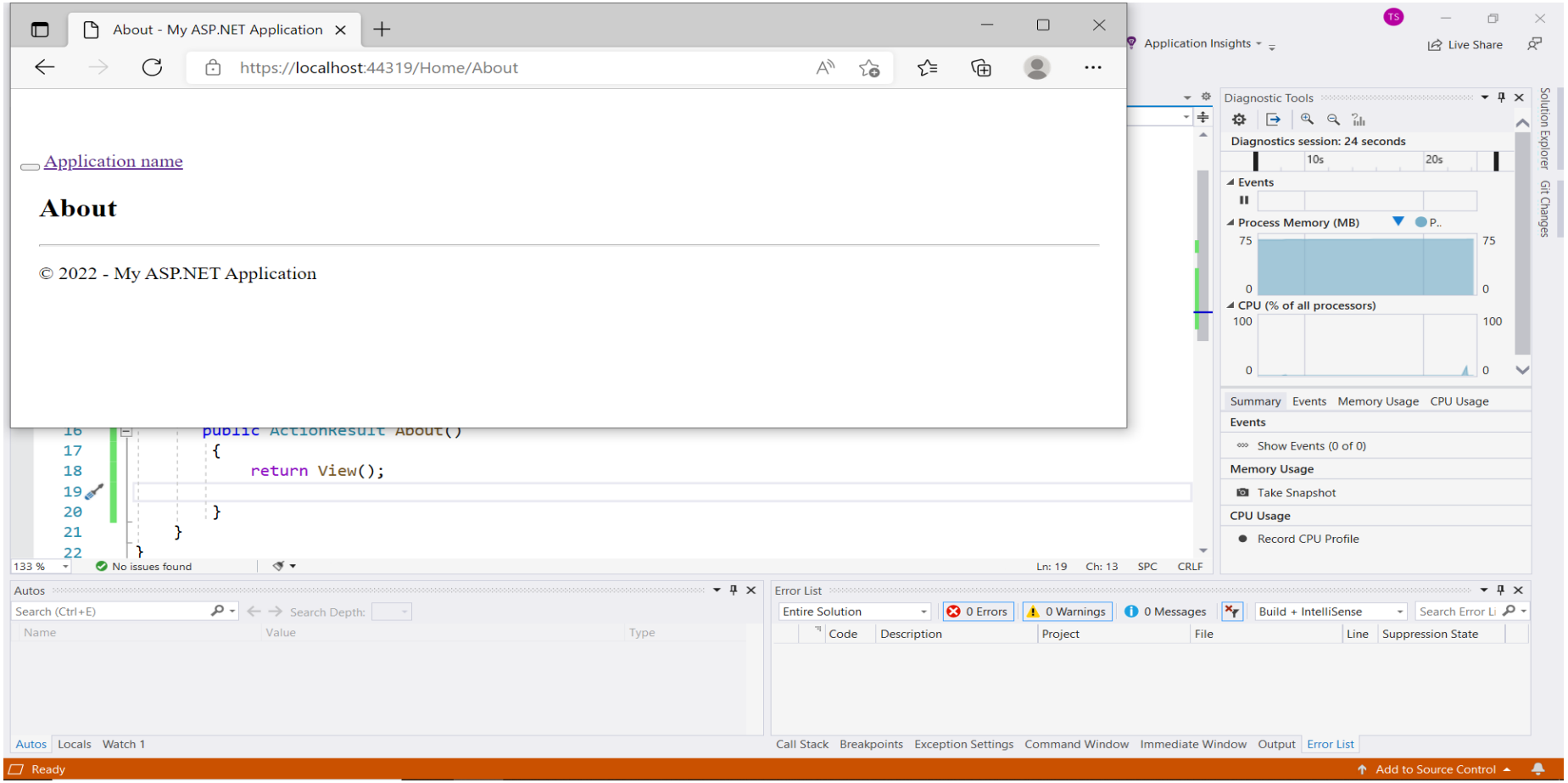



File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
Empty_project
Debug
Any CPU
IIS Express (Microsoft Edge)
Live Share
About.cshtml
Index.cshtml
HomeController.cs
Empty_project: Overview
Empty_project
Empty_project.Controllers.HomeController
Index()
using System.Web;
using System.Web.Mvc;
namespace Empty_project.Controllers
{
0 references
public class HomeController : Controller
{
// GET: Home
0 references
public ActionResult Index(
{
return View();
}
0 references
public ActionResult About()
{
return View();
}
}
}
Add View
View name:
Index
Template:
Empty (without model)
Model class:
Options:
Create as a partial view
Reference script libraries
Use a layout page:
...
(Leave empty if it is set in a Razor _viewstart file)
Add
Cancel
133 %
No issues found
Ln: 13
Ch: 10
SPC
CRLF
Output
Show output from:
Debug
'iisexpress.exe' (CLR v4.0.30319: /LM/W3SVC/2/ROOT-1-132941762235036278): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\System.Web.Mobile\v4.0_4.0.0.0__b03f5f7f11d50
'iisexpress.exe' (CLR v4.0.30319: /LM/W3SVC/2/ROOT-1-132941762235036278): Loaded 'C:\Users\lastapex\AppData\Local\Temp\Temporary ASP.NET Files\vs\4da70872\ccac0642\App_We
'iisexpress.exe' (CLR v4.0.30319: /LM/W3SVC/2/ROOT-1-132941762235036278): Loaded 'C:\Users\lastapex\AppData\Local\Temp\Temporary ASP.NET Files\vs\4da70872\ccac0642\App_We
'iisexpress.exe' (CLR v4.0.30319: /LM/W3SVC/2/ROOT-1-132941762235036278): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\System.Dynamic\v4.0_4.0.0.0__b03f5f7f11d50a3a
'iisexpress.exe' (CLR v4.0.30319: /LM/W3SVC/2/ROOT-1-132941762235036278): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\Microsoft.CSharp.resources\v4.0_4.0.0.0_tr_b0
'iisexpress.exe' (CLR v4.0.30319: /LM/W3SVC/2/ROOT-1-132941762235036278): Loaded 'C:\Users\lastapex\AppData\Local\Temp\Temporary ASP.NET Files\vs\4da70872\ccac0642\App_We
The program '[3736] iisexpress.exe' has exited with code -1 (0xffffffff).
Solution Explorer
Search Solution Explorer (Ctrl+ş)
Solution 'Empty_project' (1 of 1 project)
Empty_project
Connected Services
Properties
References
App_Data
App_Start
Content
Site.css
Controllers
HomeController.cs
Models
Views
Home
About.cshtml
Index.cshtml
Shared
_Layout.cshtml
_ViewStart.cshtml
web.config
Global.asax
packages.config
Web.config
Solution Explorer
Git Changes
Item(s) Saved
Add to Source Control

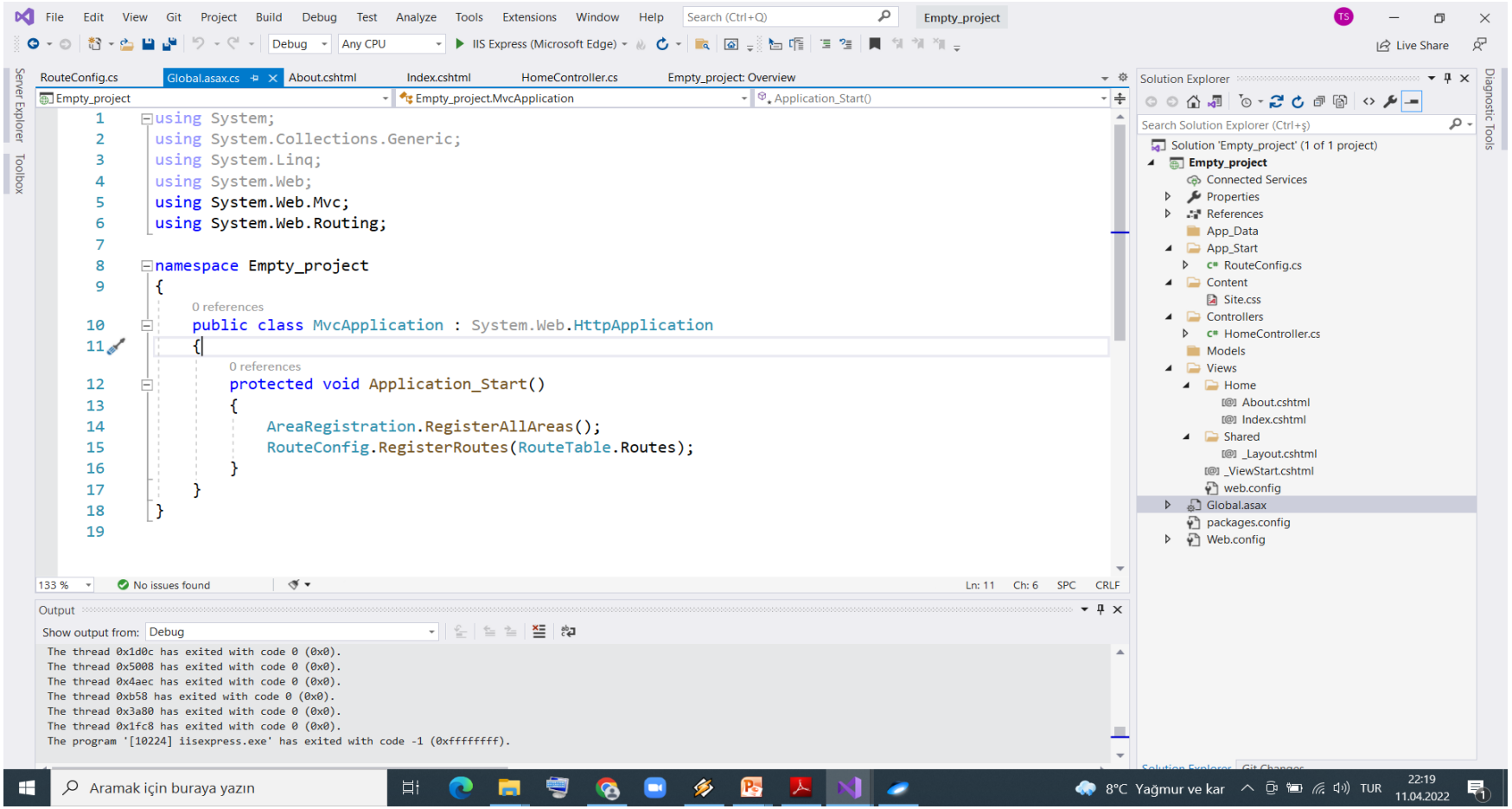
HomeController'ime geliyoruz ve birde About diye ActionResult tipinde bir metot ve About view oluşturuyoruz.



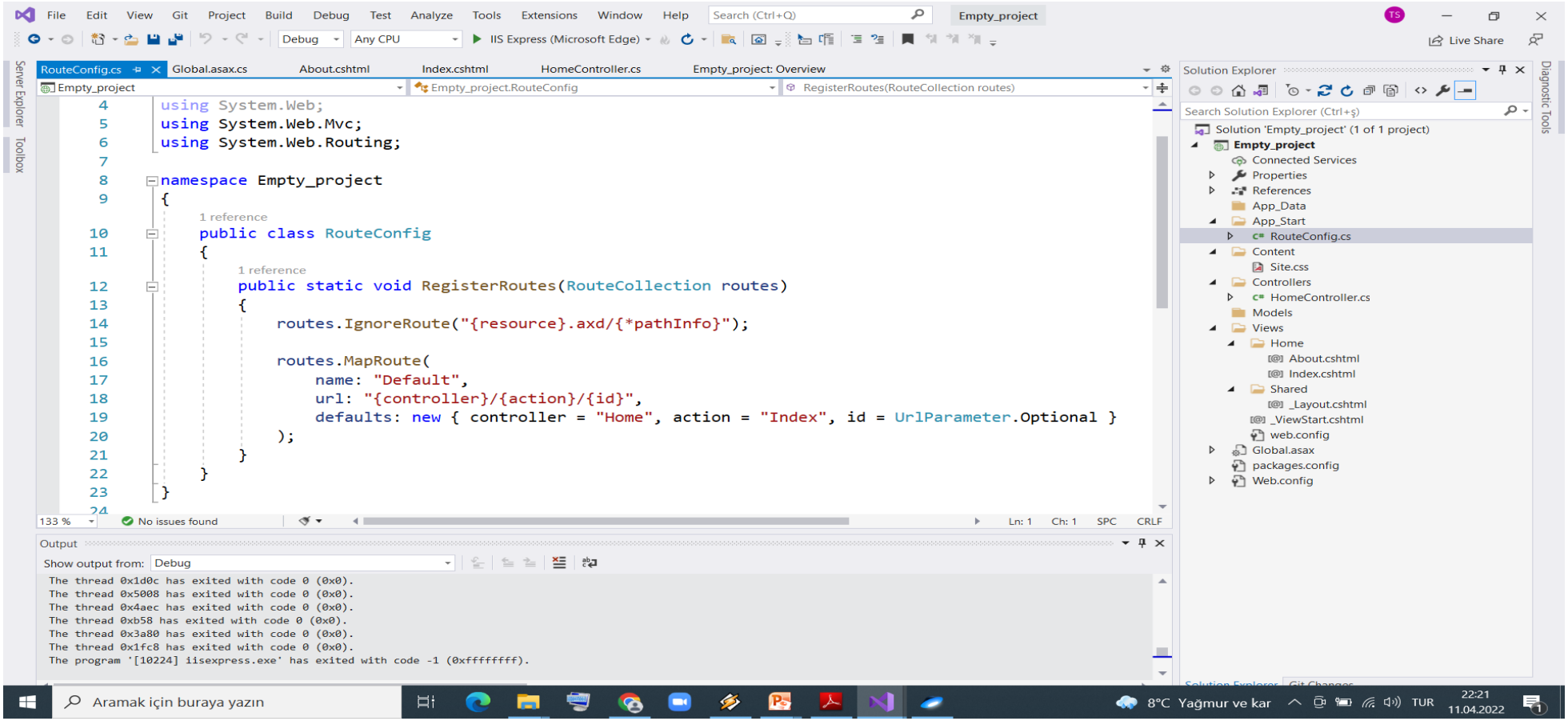
About sayfasını çağırmak için, Home/About yazdık.
Home: Oluşturduğum Controller'im adı About: HomeController içerisindeki geriye View döndüren metotumun adı.
Sonuç klasik ASP.NET kullanırken sayfaları çağırır ve öyle çalıştırırız, ASP.NET MVC de ise metotları çalıştırırız ve onlarla ,istisna dışında, aynı isimde bağlı olan View'ler görüntülenir.



Global.asax ve Route Mantığını Kavrama



Static olarak tanımlanmış RouteConfig sınıfı içindeki "RegisterRoutes" adındaki metotumuzu inceleyelim.



Bu metot içerisinde MapRoute'larım mevcut. Burada Default isminde Route tanımlanmıştır. Daha sonra çalışma şekli olarak; domain/controller adi/action adi

Örnek: deneme.com/home/index Bu tanımlanan Route içerisinde default olarak Controller=Home, Action = Index olarak verilmiştir. İlk çalıştığı an siz aksini yazmazsanız, Home/Index i arayacaktır. bulamadığı içinde hata fırlatacaktır.

Kendimiz var olan Route üzerinde değişiklik yapabileceğimiz için kendimizde yeni MapRoute'lar ekleyebiliriz.

Razor ile Web Form Kullanım Farklılıkları Neler?
Implicit Code Expression, Explicit Code Expression

Implicit Code Expression

@{string mesaj = "Implicit Code Expression";}

Razor @mesaj

Explicit Code Expression

Web Forms <%: mesaj %>

Bazı Özel Durumlar

@{string kisaisim = "tuncaysali";}

<h3>@kisaisim.com </h3>→string olduğu için .com metodu arar hata olur

<h3>@(kisaisim).com </h3>→(…….) içine alınarak metin olarak değerlendirilir.

Email adresi yazarken:tsali@gmail.com

Normalde yukarıkida kullanımdan hata alacağımızı düşünürüz fakat Razor bu ifadesinin email paternine sahip olduğunu anlamaktadır ve bu yüzden hata almayız.

Ancak @tunsali, @BillGates hata verir.

Bu durumda @@tunsali, @@BillGates yazarız.

HTML ve C# İç İçe

Razor

```
@for (int i = 1; i < 11; i++)
{
<p>Sayı @i</p>
}
```

Web Forms

```
<% for (int i=1; i < 11; i++) { %>
<p> Sayı <%: i %></span>
<% } %>
```

Razor Yorum satırları

```
@*
   yorum satırları buraya yazılır
…….
*@
// tek satır yorum eklemek için yine kullabiliriz.
```

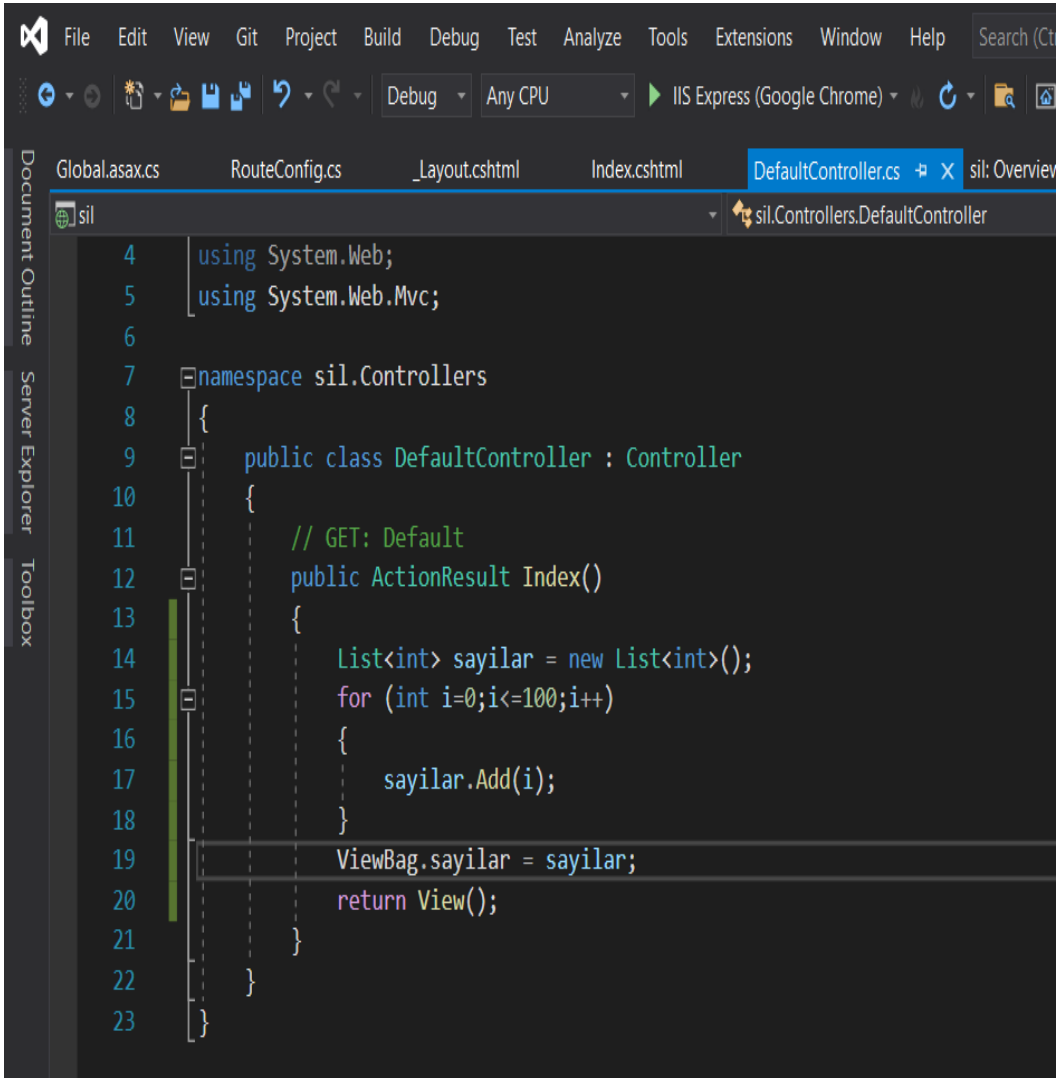
ViewBag

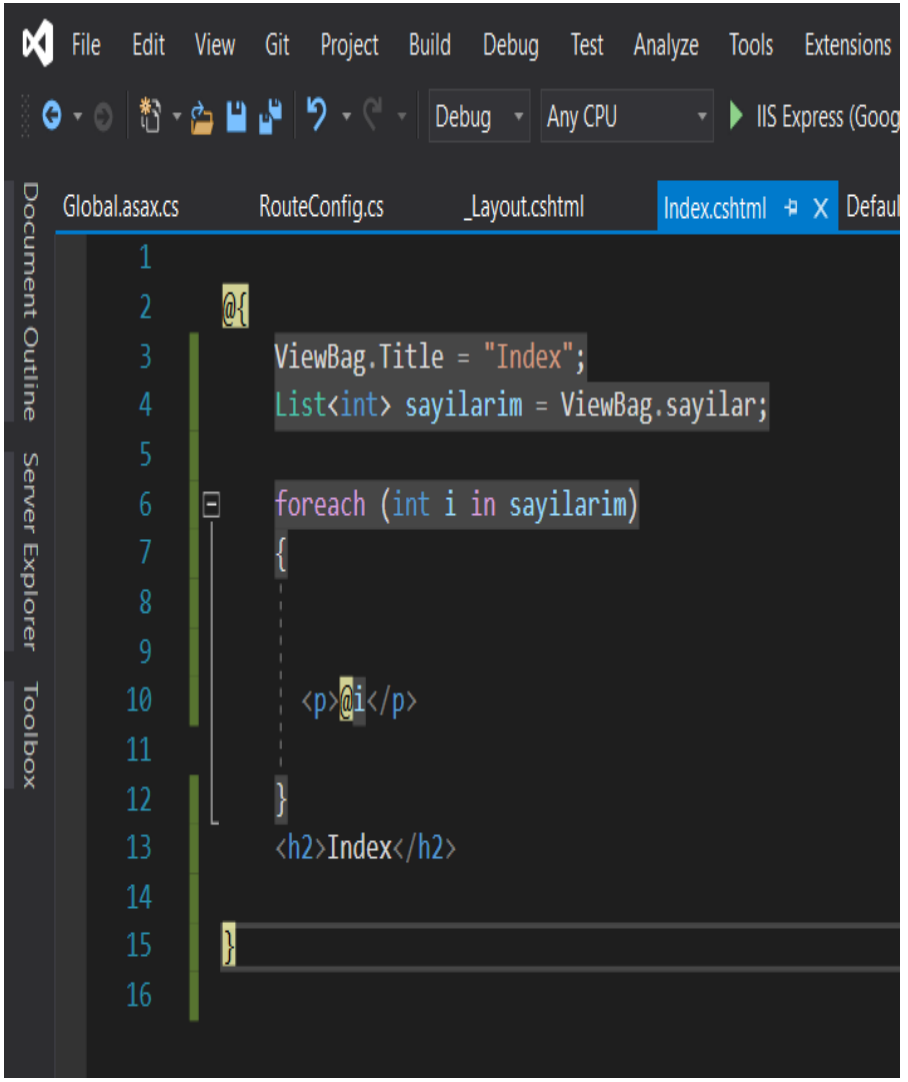
ASP.NET MVC'de Controller ile HTML arasında veri geçişlerini yapmaktan sorumludur.

Controller'da içerisinde data koyarız ve View içerisinden erişebiliriz,object veri almaktadır.

Ayrıca içerisine atayacağız herhangi bir list, tabloya View içerisinde erişip verilerinizi kullanabilirsiniz. Bu sayede Controller içerisinde atadığınız verileri herhangi bir işlem yapmadan SelectList içerisinde görebilirsiniz

ViewBag


File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Debug
Any CPU
IIS Express (Google Chrome)
Global.asax.cs
RouteConfig.cs
_Layout.cshtml
Index.cshtml
DefaultController.cs
sil
sil.Controllers.DefaultController
Document Outline
Server Explorer
Toolbox
using System.Web;
using System.Web.Mvc;
namespace sil.Controllers
{
public class DefaultController : Controller
{
// GET: Default
public ActionResult Index()
{
List<int> sayilar = new List<int>();
for (int i=0;i<=100;i++)
{
sayilar.Add(i);
}
ViewBag.sayilar = sayilar;
return View();
}
}
}


File Edit View Git Project Build Debug Test Analyze Tools Extensions
Debug
Any CPU
IIS Express (Goog
Global.asax.cs
RouteConfig.cs
_Layout.cshtml
Index.cshtml
Document Outline
Server Explorer
Toolbox
@{
ViewBag.Title = "Index";
List<int> sayilarim = ViewBag.sayilar;
foreach (int i in sayilarim)
{
@i
}
<h2>Index</h2>
}

```
        }
        0 references
        public ActionResult İletisim()
        {
            return View();
        }
        0 references
        public ActionResult viewbag_ornek()
        {
            var sayilar = new List<int>();
            for (int i=0;i<=10;i++)
            {
                sayilar.Add(i);
            }
            ViewBag.sayilarim = new SelectList(sayilar);
            return View();
        }
    }
```

```
@{
    ViewBag.Title = "viewbag_ornek";
    Layout = "~/Views/Shared/_benim_sablonum.
}

<h2>viewbag_ornek</h2>
@Html.DropDownList("sayilarim");
```

benim şablonum

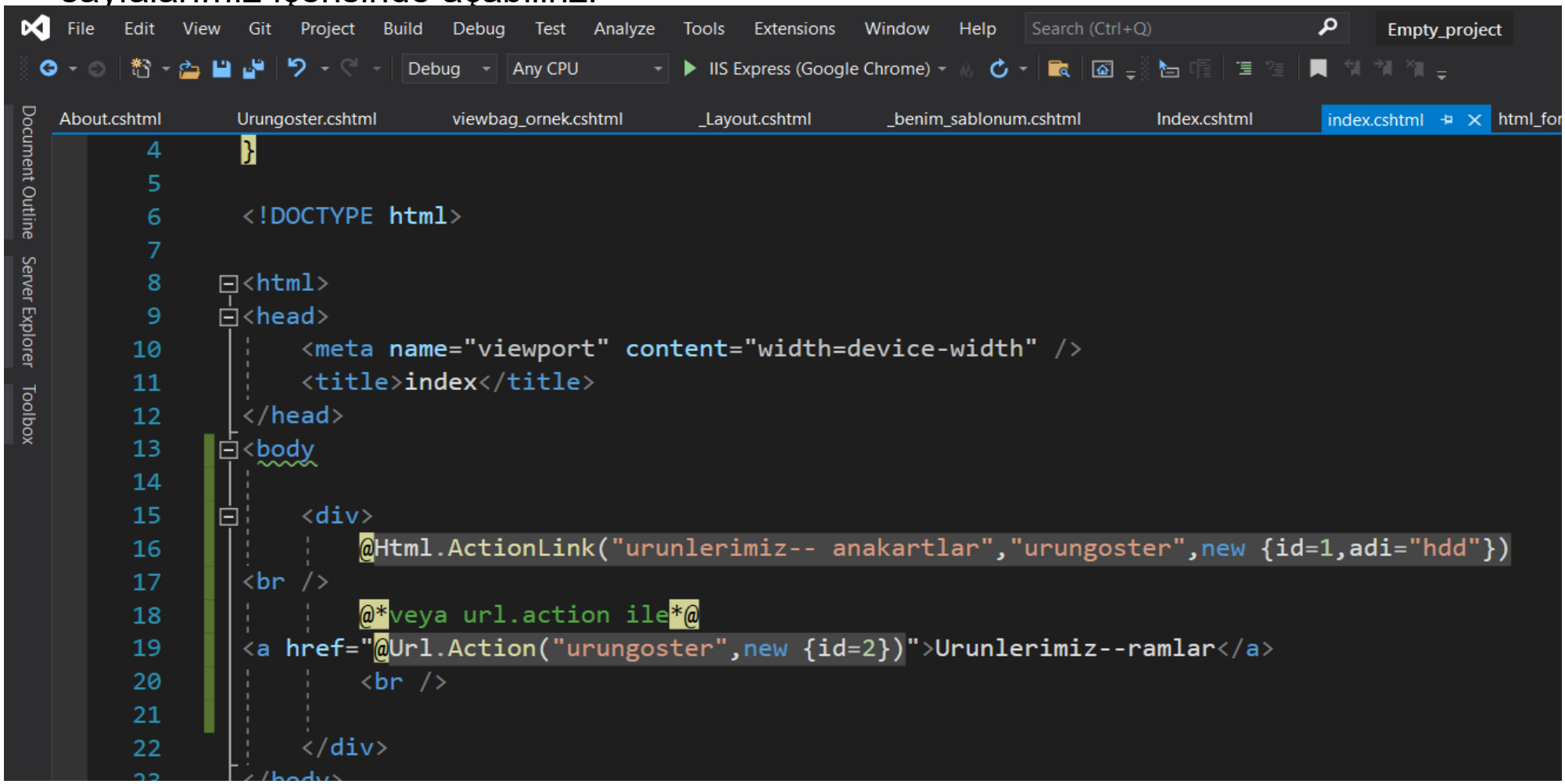
viewbag_ornek

0 ;

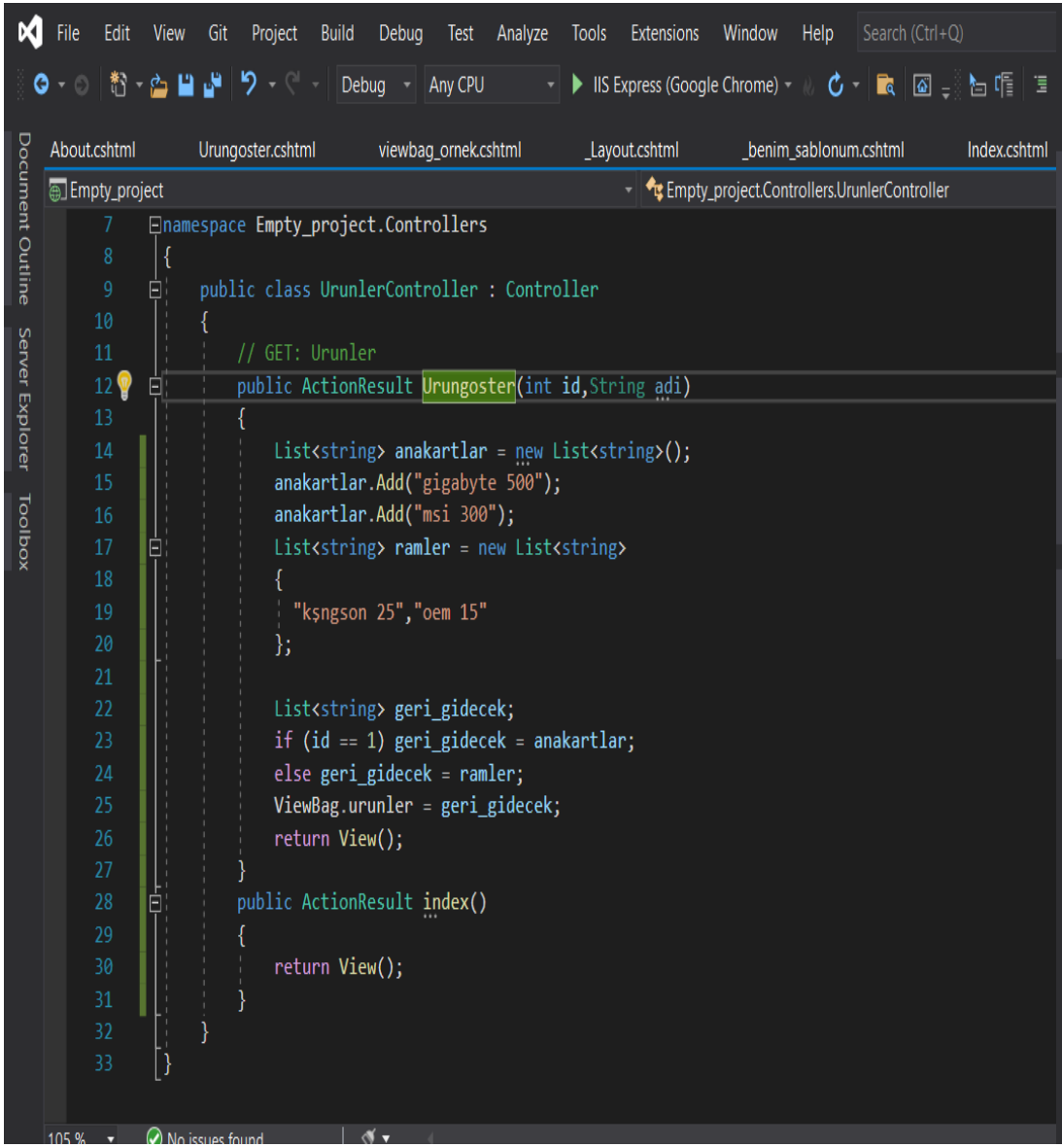
0
1
2
3
4
5
6
7
8
9
10

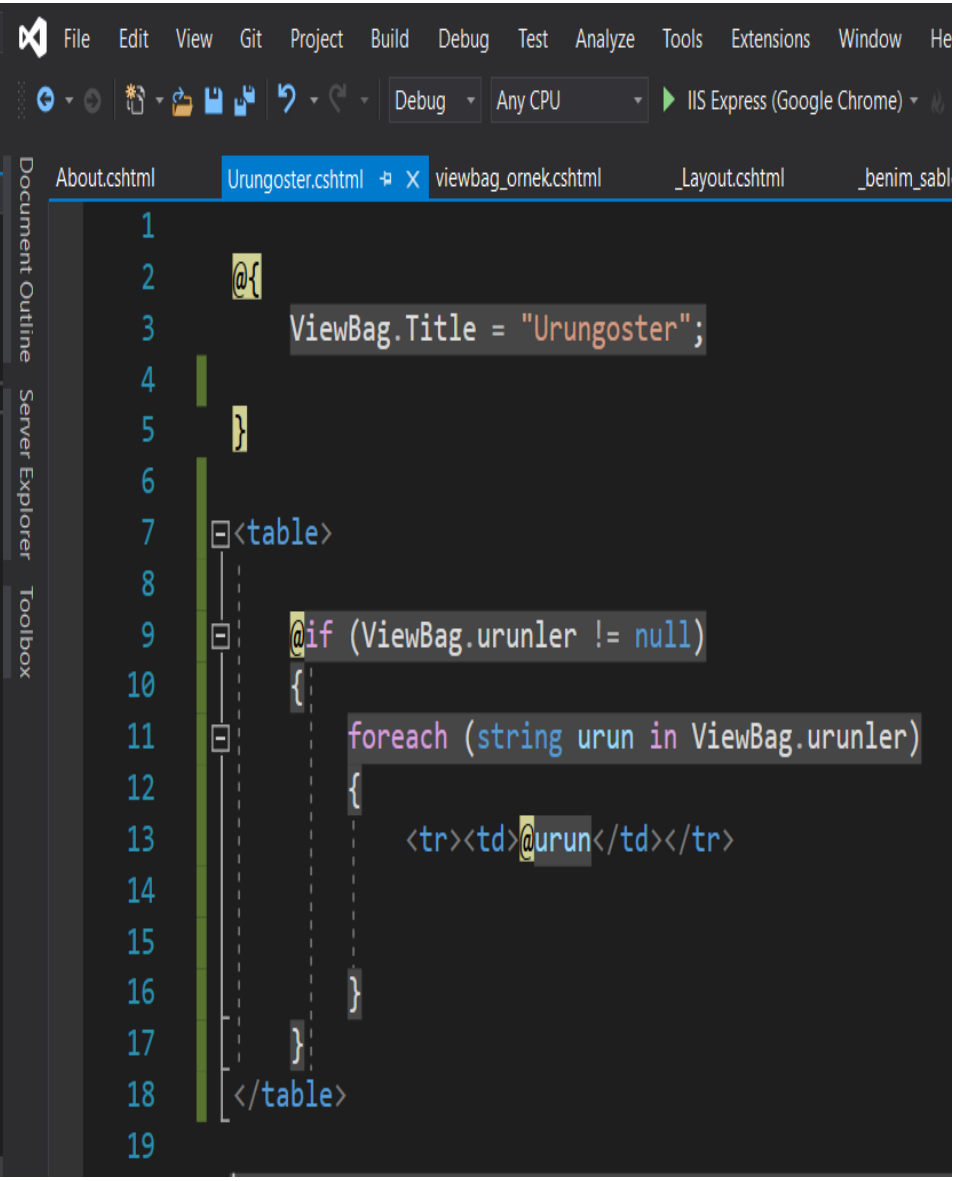
ActionLink,Url.Action

Herhangi bir uygulama içerisinden diğer kaynaklara linkler verebilir, View'leri parçalara ayırarak bölümler halinde tekrar tekrar kullanabilir ya da diğer sayfaları kendi sayfalarımız içerisinde açabiliriz.

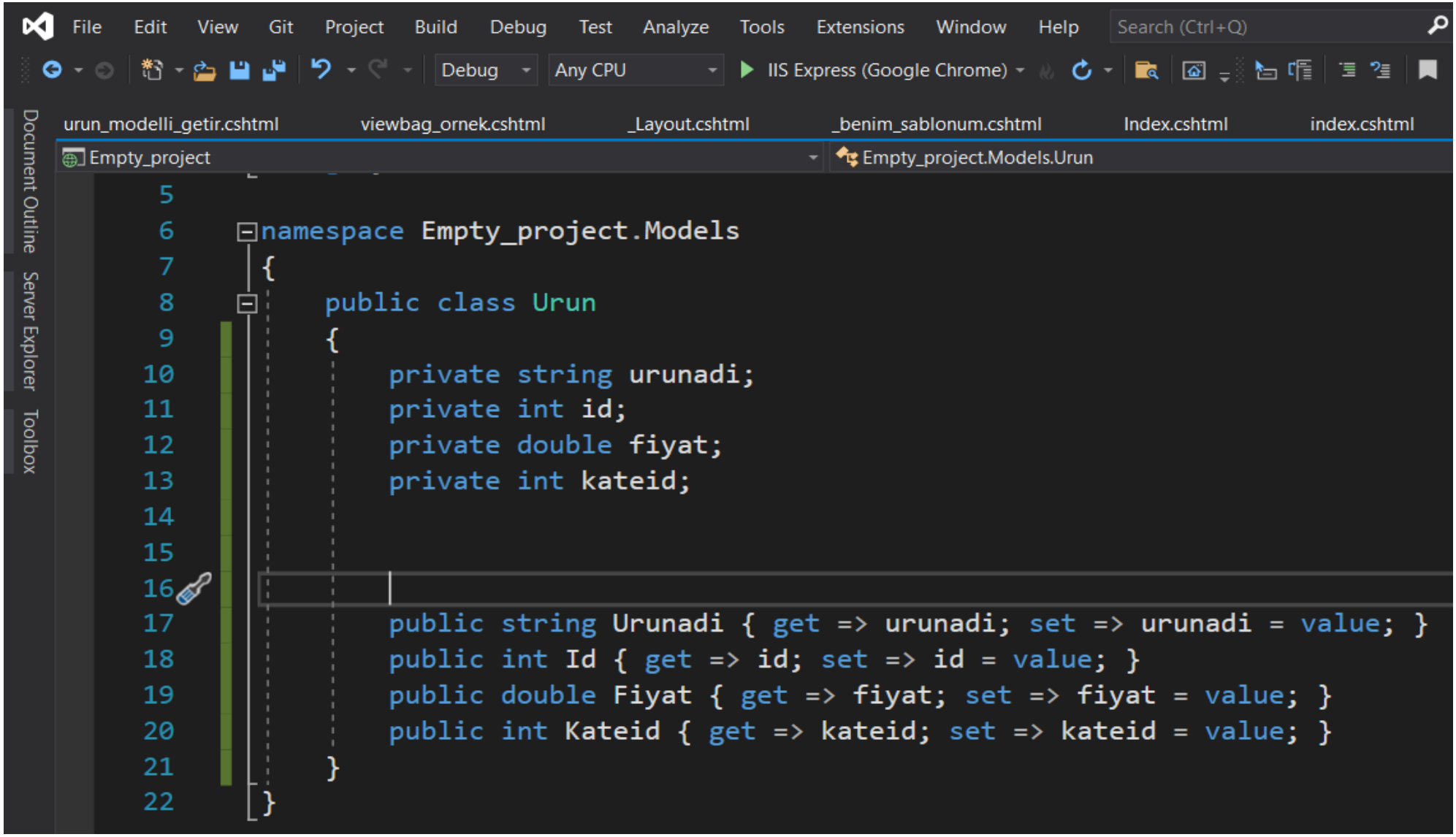


ActionLink,Url.Action

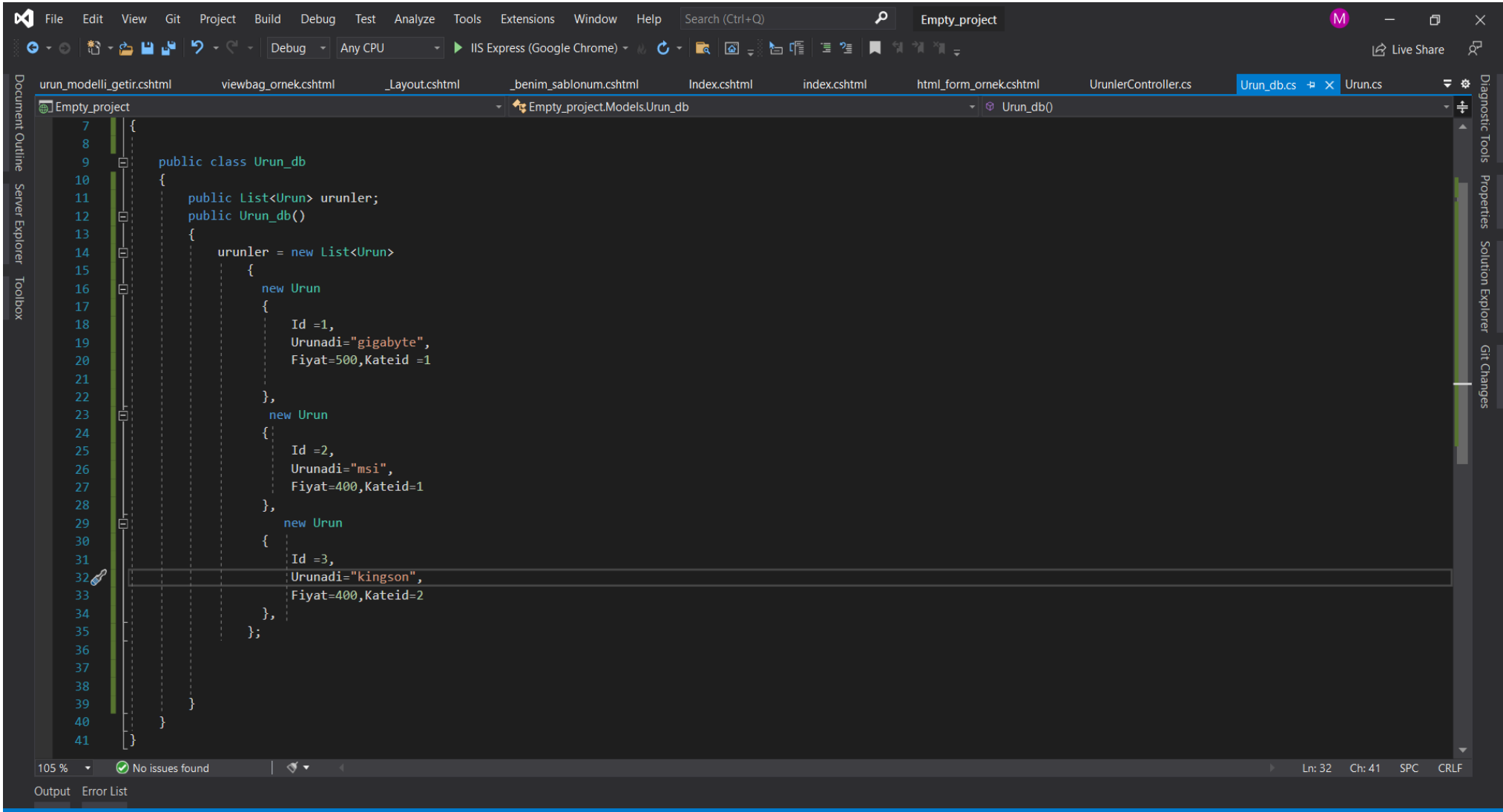




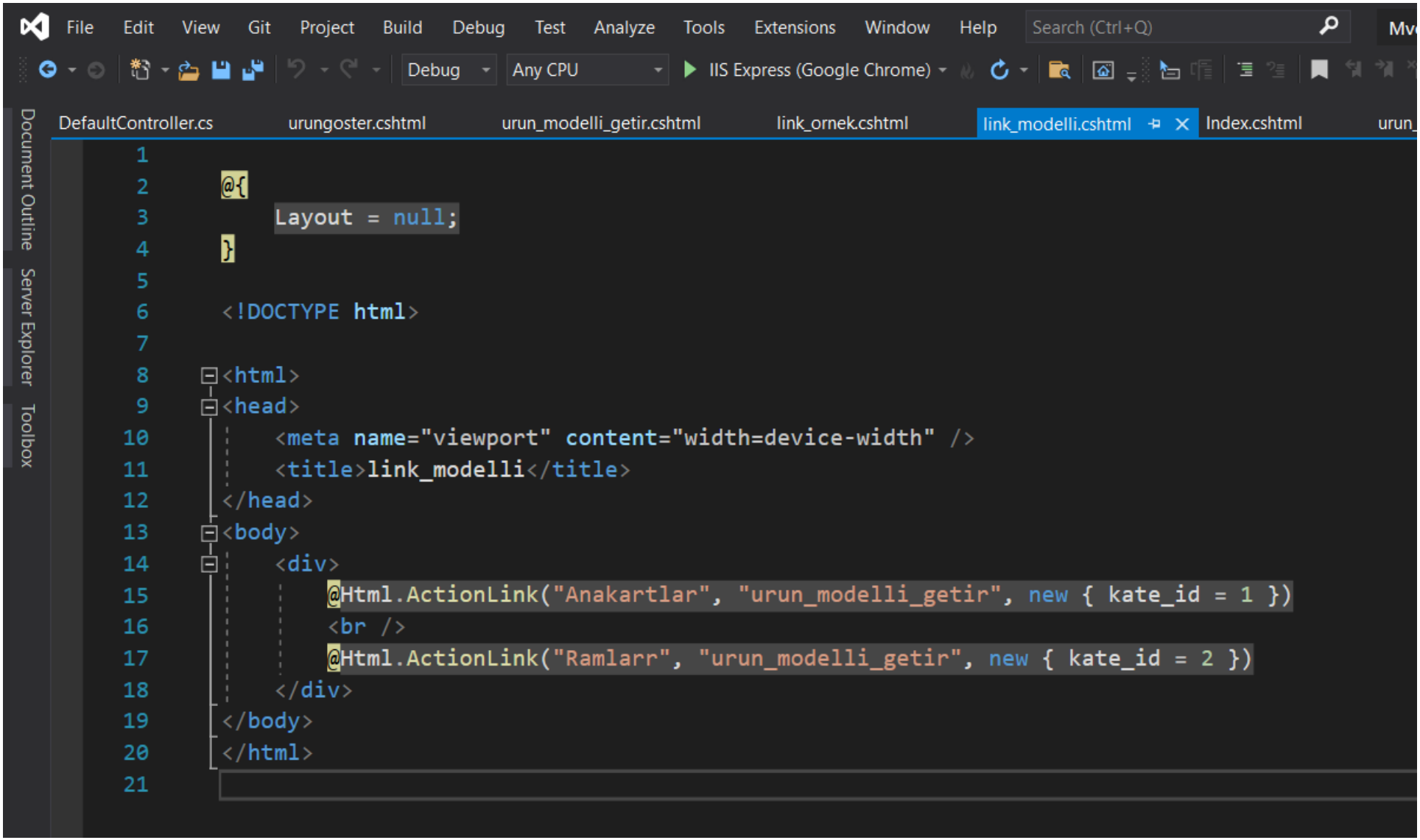
ActionLink,Url.Action-MODEL OLUŞTURALIM



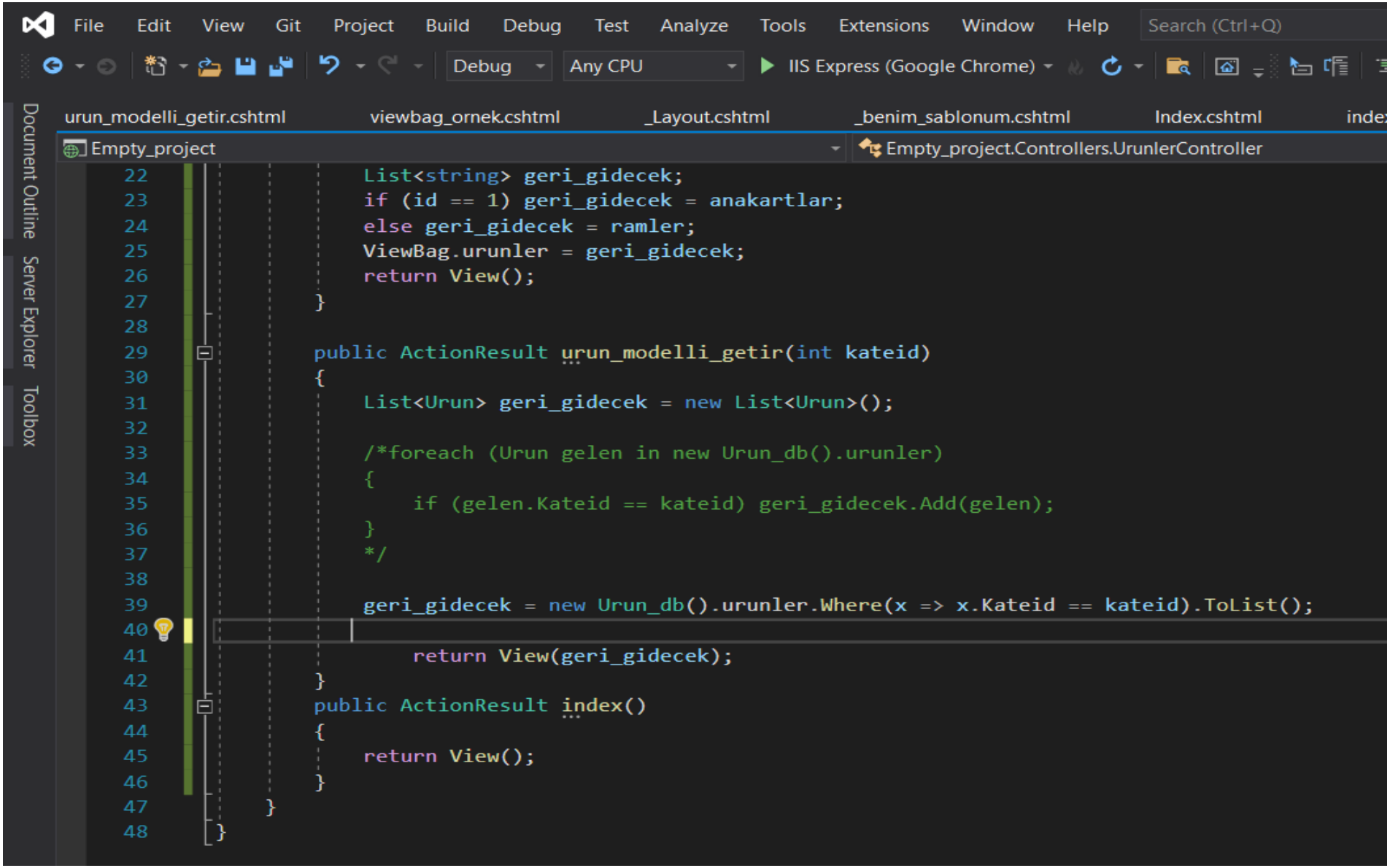
ActionLink,Url.Action-MODEL OLUŞTURALIM



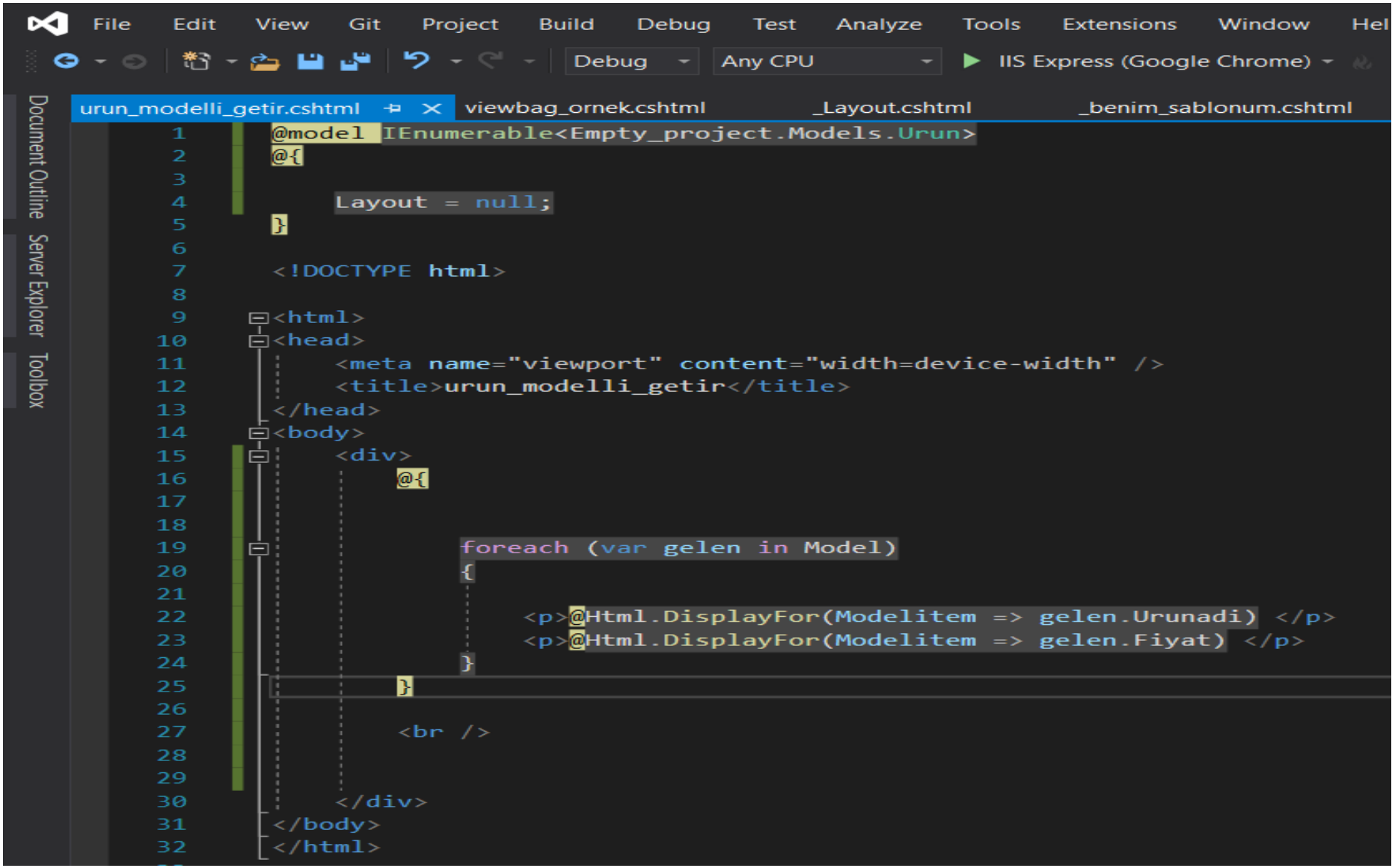
ActionLink,Url.Action-MODEL OLUŞTURALIM



ActionLink,Url.Action-MODEL OLUŞTURALIM



ActionLink,Url.Action-MODEL OLUŞTURALIM



Şablon Oluşturuken Kendi Stil Dosyamız nasıl Ekleyeceğiz?

- <link href="~/Content/sablonstil.css" rel="stylesheet" /> veya

- @Styles.Render("~/Content/css") çağırabiliriz ancak bunun için Appstart altındaki BundleConfig dosyasında

```
bundles.Add(new StyleBundle("~/Content/css").Include(
"~/Content/bootstrap.css",
"~/Content/site.css",
"~/Content/sablonstil.css"));
```

Eklemeliyiz.

Yeni BundleConfig Ekleme

- **Tools > NugetPackageManager** > Package Manager Console

Install-Package Microsoft.AspNet.Web.Optimization ile eklentiyi yüklüyoruz.

- **MVC şablonlu yeni proje açıyoz .**Ordan App_start altındaki BundleConfig.cs alıp kendi projemize kopyalıyoruz.

namespace kendi projemizle değiştiriyoruz..

- **Global.aspx** içerisine

using System.Web.Optimization;

BundleConfig.RegisterBundles(BundleTable.Bundles);

- **Views/web.Config içindeki namespaces**

< add namespace="System.Web.Optimization" />

- **.cshtml sayfanızda css** dosyalarının olması gereken yere

@Styles.Render("~/Content/css") yazıyoruz

@Html.ActionLink("Anakartlar", "urungoster", new { id = 1 }, new { @class = "btn btn-danger" })

HTML.Form ELEMANLARI

- @Html.TextBox(string name, object value)

Kullanımı: @Html.TextBox("txtEmail", "Email Giriniz...")

Html Çıktısı:<input id="txtEmail" name="txtEmail" type="text" value="Email Giriniz" />

- @Html.TextArea("txtMesaj", "Lütfen mesajınızı yazınız.")

Html Çıktısı:<textarea cols="20" id="txtMesaj" name="txtMesaj" rows="2"> Lütfen mesaj; mesajınızı yazınız.</textarea>

- Html.CheckBox

CheckBox Helper'i, 2 input element render ettiği için benzersizdir.

@Html.CheckBox(string name, bool isChecked)

Kullanımı:@Html.CheckBox("cbEhliyetVarmi",false)

Html Çıktısı:

<input id="cbEhliyetVarmi" name="cbEhliyetVarmi" type="checkbox" value="true" /><input name="cbEhliyetVarmi" type="hidden" value="false" />

Tarayıcının checkbox'ın sadece selected olma durumunda değeri submit etmesini sağlamak için iki input element render eder

HTML.Form ELEMANLARI

- @Html.RadioButton

Html.RadioButton‘lar, Genellikle tek bir sonuç için muhtemel seçenekleri sağlayan gruplanmış kontrollerdir. Radiobutton’ları gruplamak için, her birisine aynıismi(name) vermelisiniz.

Kullanımı: @Html.RadioButton("pet", "Köpek",true)

@Html.RadioButton("pet", "Kedi")

Html Çıktısı:

```
<input checked="checked" id="pet" name="pet" type="radio" value="Köpek" />
<input id="pet" name="pet" type="radio" value="Kedi" />
```

HTML.Form ELEMANLARI

- @Html.DropDownList(**string** name, **IEnumrable** selectList, **string** optionLabel, **IDictionary<string, object>** htmlAttributes)
- @Html.ListBox(**string** name, **IEnumrable** selectList, **string** optionLabel, **IDictionary<string, object>** htmlAttributes)

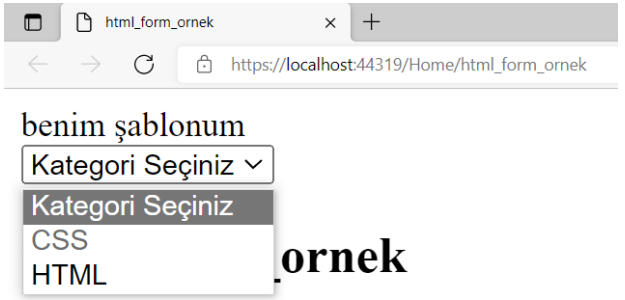
string name = "Kontrolün ismini"

IEnumrable = "Listenin tanımlandığı parametredir"

string optionLabel = "Default değeridir -Bir seçim yapınız-"

IDictionary<string, object> = "DropDownListimizde HTML kodlarını kullanacağımız

```
tml_form_ornek.cshtml*  viewbag_ornek.cshtml  İletisim.cshtml  _benim_sablonum.cshtml  Index.cshtml  _Layout.cshtml

1
2       @{
3           ViewBag.Title = "html_form_ornek";
4           Layout = "~/Views/Shared/_benim_sablonum.cshtml";
5       }
6       @Html.DropDownList("kategoriler",
7           new List<SelectListItem>
8           {
9           new SelectListItem
10          { Text="CSS", Value="1",
11          //Selected = true, //seçili olarak gelsin
12          Disabled = true //disable olarak gelsin istersek kullanıyoruz
13          },
14          new SelectListItem
15          { Text="HTML", Value="2" },
16          },"Kategori Seçiniz",new { @class = "form-control"})
17
18          <h2>html_form_ornek</h2>
19
20
```



HTML.Form ELEMANLARI

- @Html.Label(string expression,string text)

Ik parametre olarak For Attribute'une atacağız değerdir. Bir radiobutton'a,checkbox 'a For diyerek expression verebilir ve kullanabilirsiniz.

Kullanımı:

@Html.Label("cbEhliyetVarmi", "Ehliyetiniz Varmı?") @Html.CheckBox("cbEhliyetVarmi")

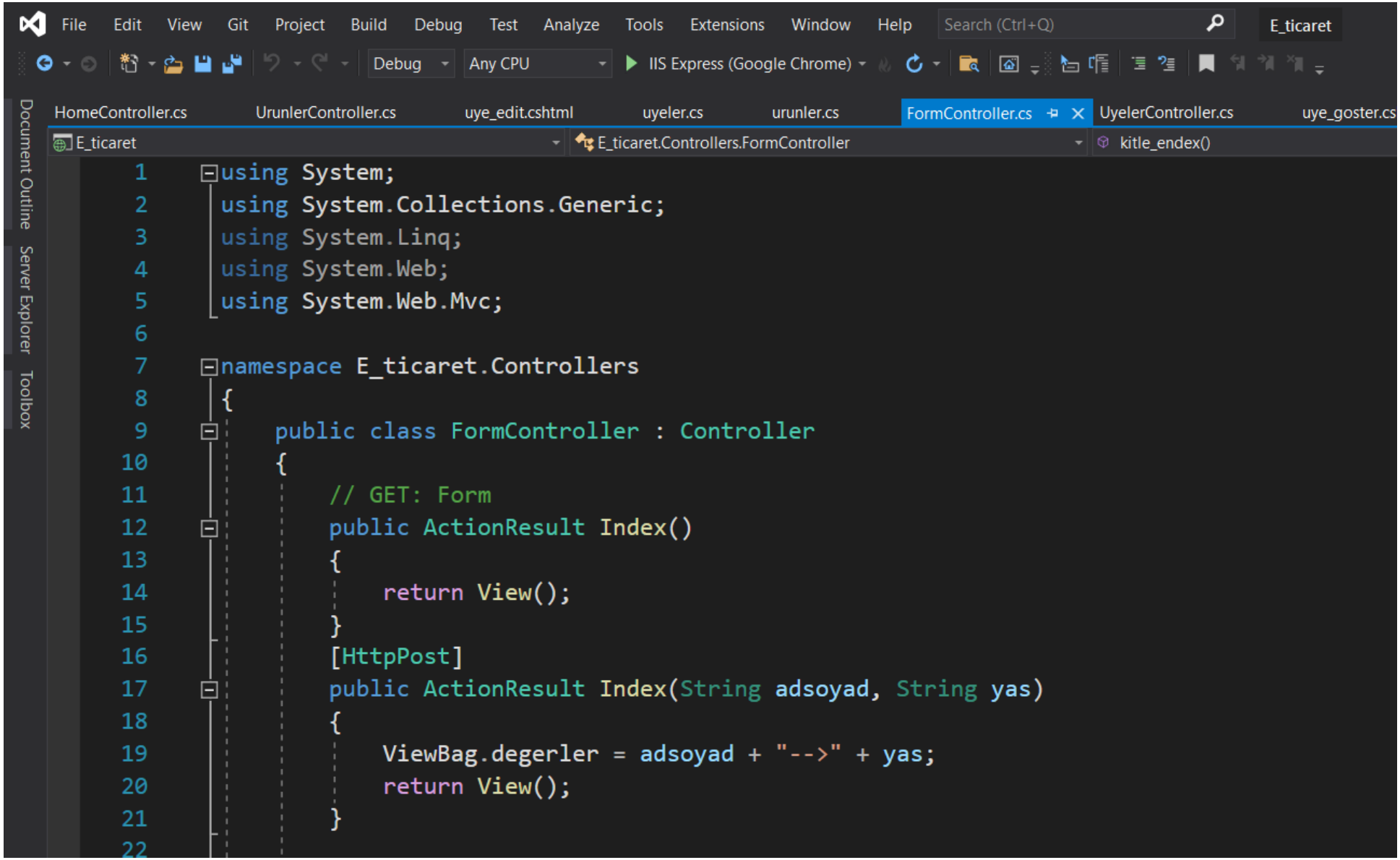
- @Html.Password

Password helper'i, password alanı render eder. TextBox helper'ina benzer fakat post edilen değeri içerisinde tutmaz ve isminde anlaşıldığı gibi password maskeli olarak giriş yapılır.

Kullanımı:

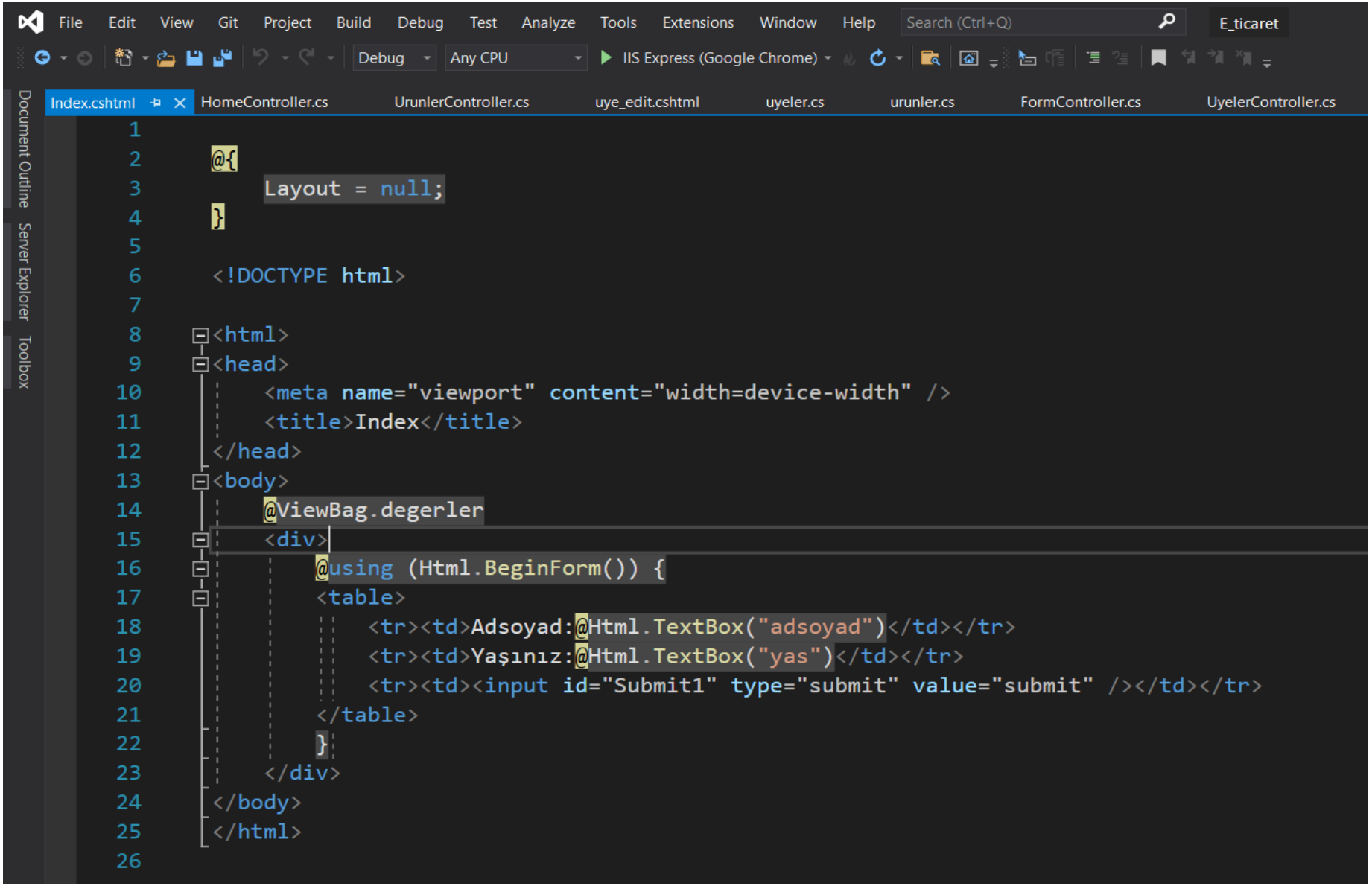
@Html.Password("txtSifre")

HTML.Form ELEMANLARI

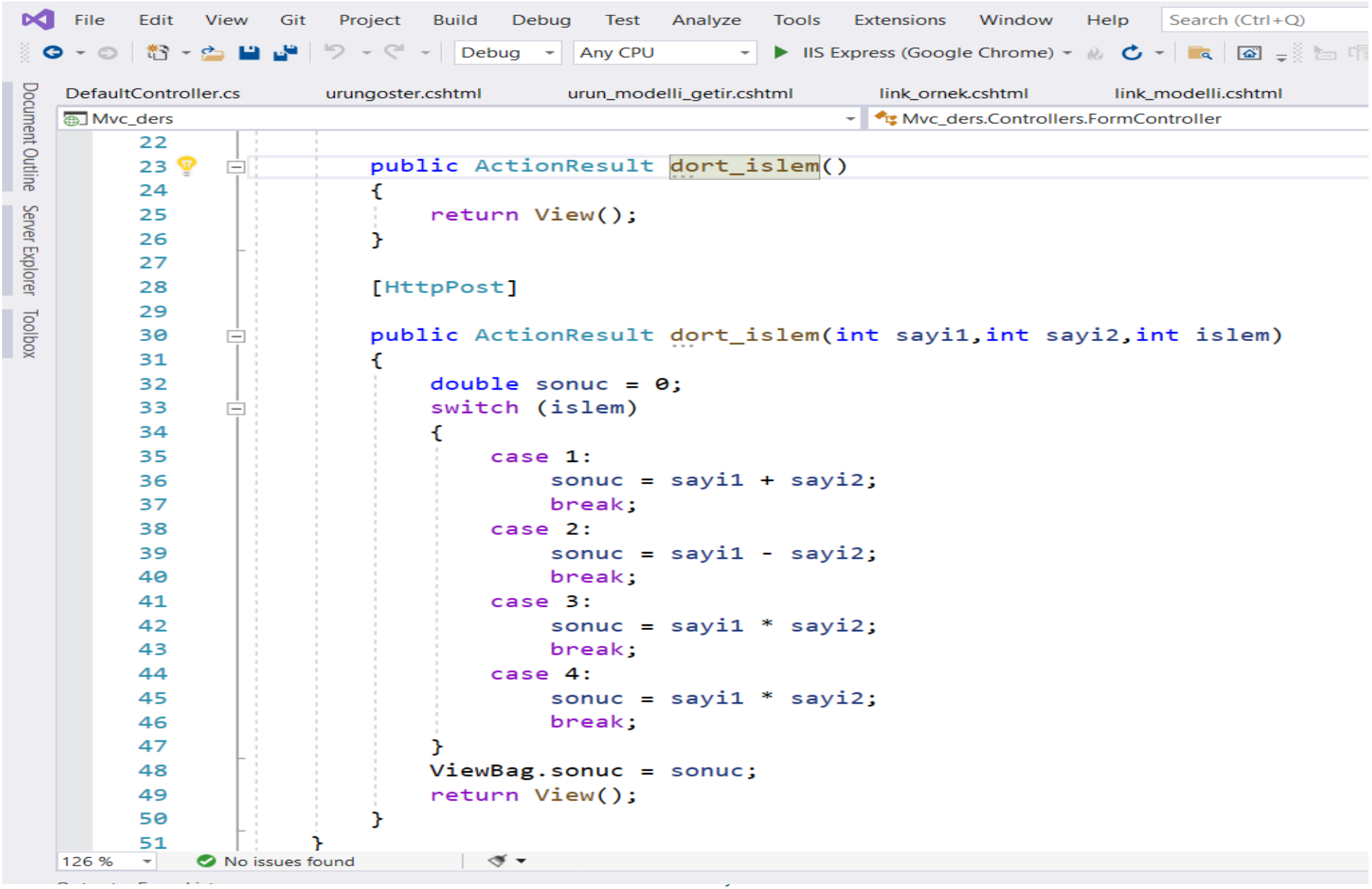



```
1   using System;
2   using System.Collections.Generic;
3   using System.Linq;
4   using System.Web;
5   using System.Web.Mvc;
6
7   namespace E_ticaret.Controllers
8   {
9       public class FormController : Controller
10      {
11          // GET: Form
12          public ActionResult Index()
13          {
14              return View();
15          }
16          [HttpPost]
17          public ActionResult Index(String adsoyad, String yas)
18          {
19              ViewBag.degerler = adsoyad + "-->" + yas;
20              return View();
21          }
22
```

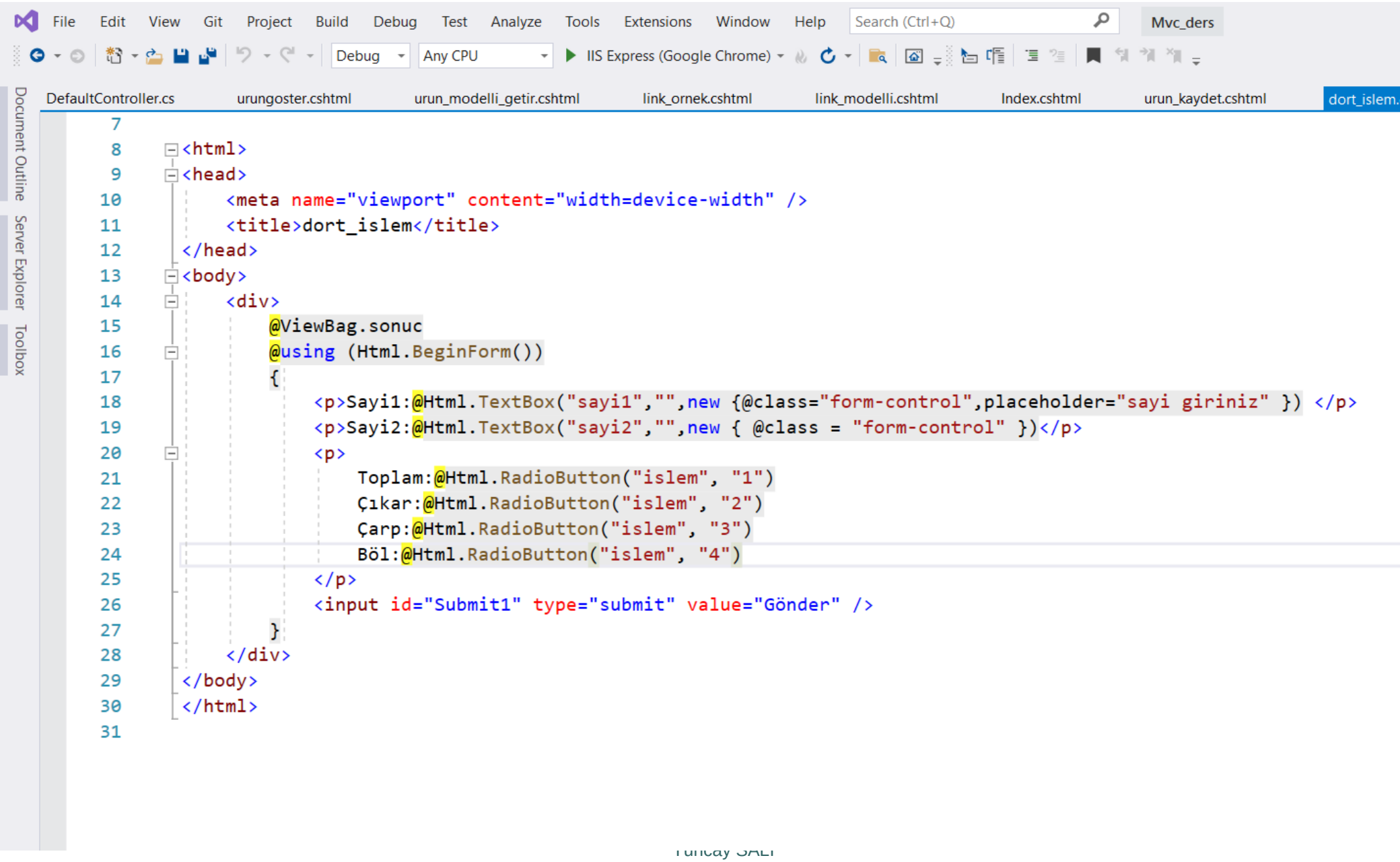
HTML.Form ELEMANLARI



HTML.Form ELEMANLARI

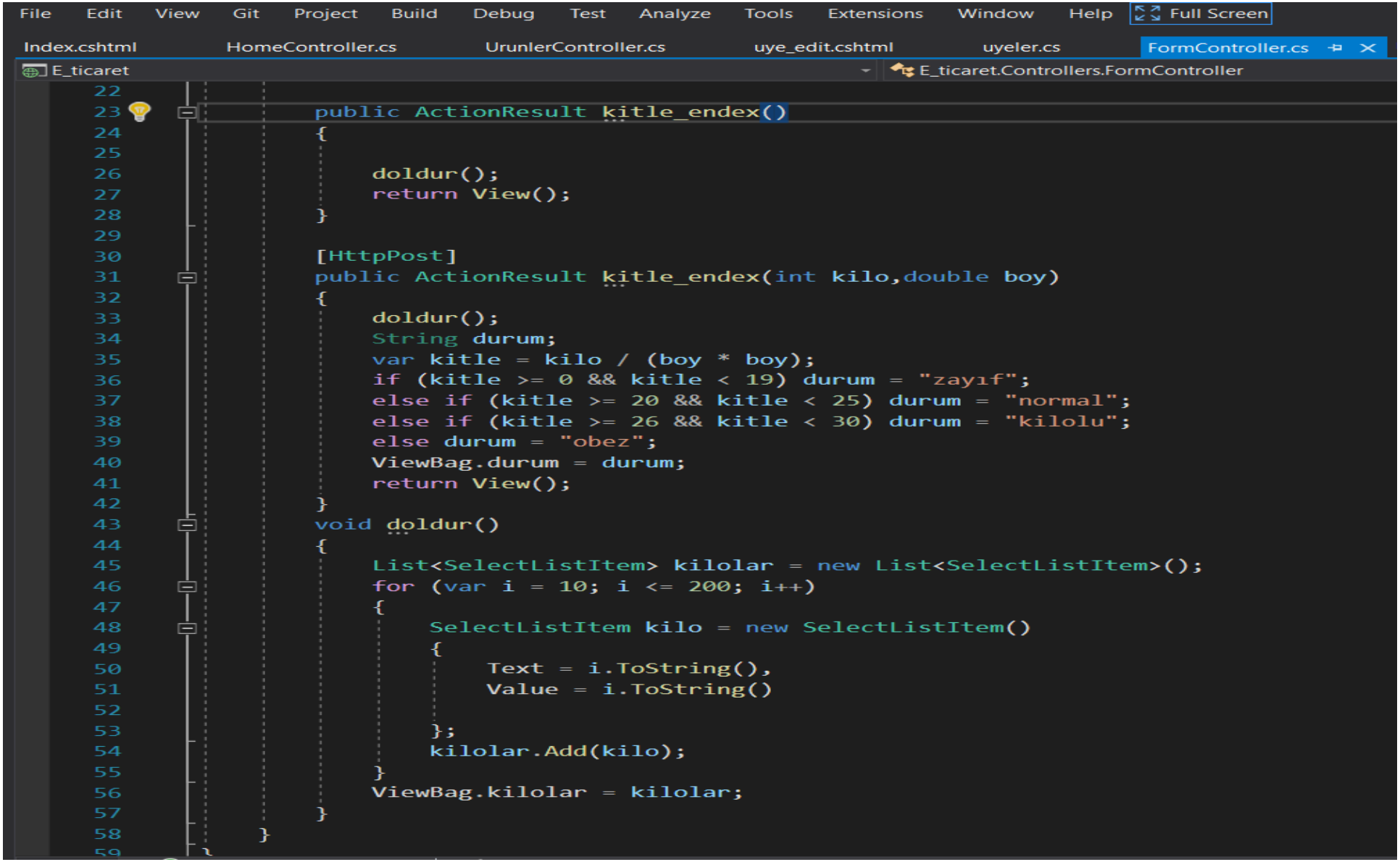


HTML.Form ELEMANLARI




```
7
8   <html>
9   <head>
10      <meta name="viewport" content="width=device-width" />
11      <title>dort_islem</title>
12  </head>
13  <body>
14      <div>
15          @ViewBag.sonuc
16          @using (Html.BeginForm())
17          {
18              <p>Sayi1:@Html.TextBox("sayi1","",new {@class="form-control",placeholder="sayi giriniz" }) </p>
19              <p>Sayi2:@Html.TextBox("sayi2","",new { @class = "form-control" })</p>
20              <p>
21                  Toplam:@Html.RadioButton("islem", "1")
22                  Çıkar:@Html.RadioButton("islem", "2")
23                  Çarp:@Html.RadioButton("islem", "3")
24                  Böl:@Html.RadioButton("islem", "4")
25              </p>
26              <input id="Submit1" type="submit" value="Gönder" />
27          }
28      </div>
29  </body>
30  </html>
31
```

HTML.Form ELEMANLARI



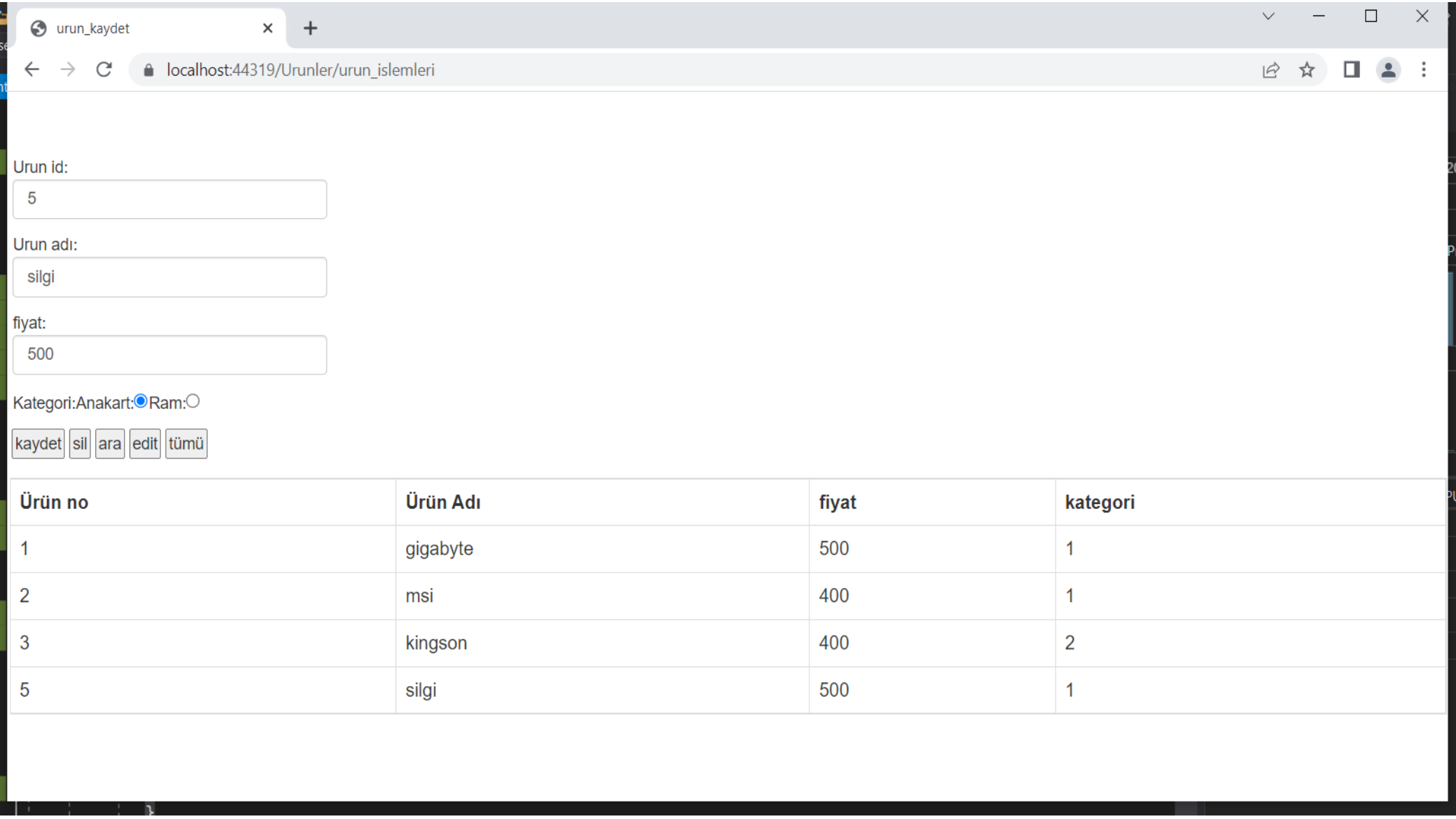
HTML.Form ELEMANLARI

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help Full Screen Search (Ctrl+Q)

Index.cshtml | HomeController.cs | UrunlerController.cs | uye_edit.cshtml | uyeler.cs | FormController.cs | UyelerController.cs | kitle_endex.cshtml | urunler.cs | uye_goster.cshtml | _Layout.cshtml

```
1
2
3       Layout = null;
4
5
6     )OCTYPE html>
7
8     :ml>
9     ead>
10      <meta name="viewport" content="width=device-width" />
11      <title>kitle_endex</title>
12    ead>
13    dy>
14      <div>
15          @using (Html.BeginForm())
16          {
17          <table>
18              <tr><td>Kilonuz:@Html.DropDownList("kilo", (List<SelectListItem>)ViewBag.kilolar, new { onchange = "this.form.submit()" })</td></tr>
19              <tr><td>Boyunu:@Html.TextBox("boy")</td></tr>
20              <tr><td>sonuç:@ViewBag.durum</td></tr>
21              <tr><td><input id="Submit1" type="submit" value="submit" /></td></tr>
22          </table>
23          }
24
25      </div>
26    ody>
27    tml>
```

HTML.Form ELEMANLARI



HTML.Form ELEMANLARI

i.cshtml* | Urun_db.cs | Urun.cs | UrunlerController.cs | BundleConfig.cs

```
@model IEnumerable<Empty_project.Models.Urun>
@{
    Layout = null;
}
@Styles.Render("~/Content/css")
<!DOCTYPE html>
<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title>urun_kaydet</title>
</head>
<body>
    <div>
        @using (Html.BeginForm())
        {
            <p>Urun id:@Html.TextBox("id", "", new { @class = "form-control" })</p>
            <p>Urun adı:@Html.TextBox("urunadi", "", new { @class = "form-control" })</p>
            <p>fiyat:@Html.TextBox("fiyat", "", "", new { @class = "form-control" })</p>
            <p>Kategori:Anakart:@Html.RadioButton("kateid", "1", true)Ram:@Html.RadioButton("kateid", "2", true)</p>

            <input name="islem" type="submit" value="kaydet" />
            <input name="islem" type="submit" value="sil" />
            <input name="islem" type="submit" value="ara" />
            <input name="islem" type="submit" value="edit" />
            <input name="islem" type="submit" value="tümü" />
        }

    </div>
    <table class="table table-bordered ">
        <tr><th>Ürün no</th> <th>Ürün Adı</th><th>fiyat</th><th>kategori</th></tr>
        @foreach (var gelen_urun in Model)
        {
        <tr>
            <td>@Html.DisplayFor(Modelitem => gelen_urun.Id) </td>
            <td>@Html.DisplayFor(Modelitem => gelen_urun.Urunadi) </td>
            <td>
```

HTML.Form ELEMANLARI

Git Project Build Debug Test Analyze Tools Extensions Window Help Full Screen

Urun_db.cs | Urun.cs | UrunlerController.cs* | BundleConfig.cs

Empty_project.Controllers.UrunlerController

```
[HttpGet]
public ActionResult urun_islemleri()
{
    List<Urun> urunlerim = new Urun_db().urunler;

     return View(urunlerim);
}
[HttpPost]
public ActionResult urun_islemleri(Urun urunum,string islem)
{
     List<Urun> urunlerim;
     List<Urun> arama_listesi=null;
     if (Session["urunlerim"] != null) urunlerim = (List<Urun>)Session["urunlerim"];

      else urunlerim = new Urun_db().urunler;

     //string islem = Request.Form["islem"];
     if (islem.Equals("kaydet"))
     {
         urunlerim.Add(urunum);
     }
     else if (islem.Equals("sil"))
     {
         Urun sil_urun = urunlerim.Where(x => x.Id == urunum.Id).First();
         urunlerim.Remove(sil_urun);
     }
     else if (islem.Equals("ara"))
     {
         arama_listesi = urunlerim.Where(x => x.Id == urunum.Id).ToList();
     }
     else if (islem.Equals("edit"))
     {
         int z = urunlerim.FindIndex(x => x.Id == urunum.Id);
         urunlerim[z] = urunum;
     }
     Session["urunlerim"] = urunlerim;
     if (islem.Equals("ara")) return View(arama_listesi);
     else return View(urunlerim);
```

sues found

Entity Framework ile Veritabanı Modelleme

- **Entity Framework nedir?**

Entity Framework 2008 yılından itibaren Microsoft tarafında geliştirilen ORM aracıdır.

- **ORM nedir?**

ORM veya **O**bject to **R**elational **M**apping temel olarak veritabanında yer alan tablo ve alanları nesne olarak kullanmamıza imkan veren bir yazılım mimarisidir.

- **Neden kullanılır?**

.NET veya herhangi bir programlama dili ile veritabanı uygulamaları yaparken ilk olarak veritabanı bağlantısı yapılır.

Daha sonra SQL komutları ile veritabanındaki veriler alınır.

Alınan veriler programlama diline uygun veri yapılarında saklanarak işlem yapılır.

Verilerin programlama diline uygun veri yapılarına dönüştürülmesi sırasında beklenmedik hatalar, sorunlar oluşabilir.

Ayrıca karmaşık veritabanı sorguları geliştirmeyi daha da zor hale getirir.

ORM araçlarının temel kullanım nedeni bu ve bunun gibi sorunları ortadan kaldırmaktır.

Entity Framework ile Veritabanı Modelleme

- ***Avantajları***
  - OOP olarak kod geliştirmemize olanak tanır.
  - Hiçbir SQL bilgisi olmayan bir kimse veritabanı işlemlerini **EF** ile gerçekleştirebilir.
  - Herhangi bir veritabanına bağımlılık yoktur. Oracle, MS SQL ile kullanılabilir.
  - Code First sayesinde projenizi veritabanınızı taşıma gereği duymadan istediğiniz yerde oluşturabilirsiniz. Buda projelerinize büyük bir esneklik sağlamaktadir.
  - Yazılım geliştirme zamanını azaltır.
  - Yazılım geliştirme maaliyetini azaltır.
- ***Dezavantajları***
  - En büyük sorunumuz performans. Ado.Net gibi hızlı bir performansı yoktur. Tabi bu yavaş olduğu anlamına gelmez.
  - Veritabanından veri alış-verişi yapılacağı zaman kontrol bizde değil Entity Frameworktedir. Yani arka planda veritabanı işlemleri için kendisi sorgular oluşturmaktadır. Basit bir veri işlemi için karmaşık bir sorgu gönderebilmektedir. Syntax'ı yeni kullanacak kişiler için karmaşık gelebilmektedir ancak zamanla alışacaksınız.
  - Schema'da herhangi bir değişiklik yapıldığı zaman EF çalışmayacaktır. Sizin bu Schema'yı solution'da güncellemeniz gerekmektedir.

Entity Framework ile Veritabanı Modelleme

- **Entity Framework kurulumu**

Entity Framework ORM aracı bir Nuget paketi olduğundan Visual studio geliştirme ortamında projeye sağ tıklayıp **Manage NuGet Packages** seçeneği ile kurulum yapılabilir.

Ayrıca **Package Manager Console** alanına aşağıdaki komut yazılarakta indirilebilir.

Install-Package EntityFramework

Model oluşturma

Entity Framework ile model oluşturmak için Database First, Model First ve Code First yaklaşımları kullanılır.

- **Database First**

Bu yapıda daha önceden hazırlanmış veritabanı tablolarına uygun nesneler Visual Studio veya komut satırı ile otomatik olarak oluşturulur.

Bu işlem için Visual Studio IDE ile projeye **ADO.NET Entity Data Model** ve **EF Designer from database** seçeneği ile veritabanı bağlantısı yapılarak işlem yapılacak tabloların eklenmesi yeterli olacaktır.

- **Model First**

Bu yapıda Database First yapısına benzer şekilde **ADO.NET Entity Data Model** ve **EF Designer from model** seçeneği ile oluşturulur.,

Oluşturulan modele görsel olarak tablolar, tablo alanları eklenerek model oluşturulur.

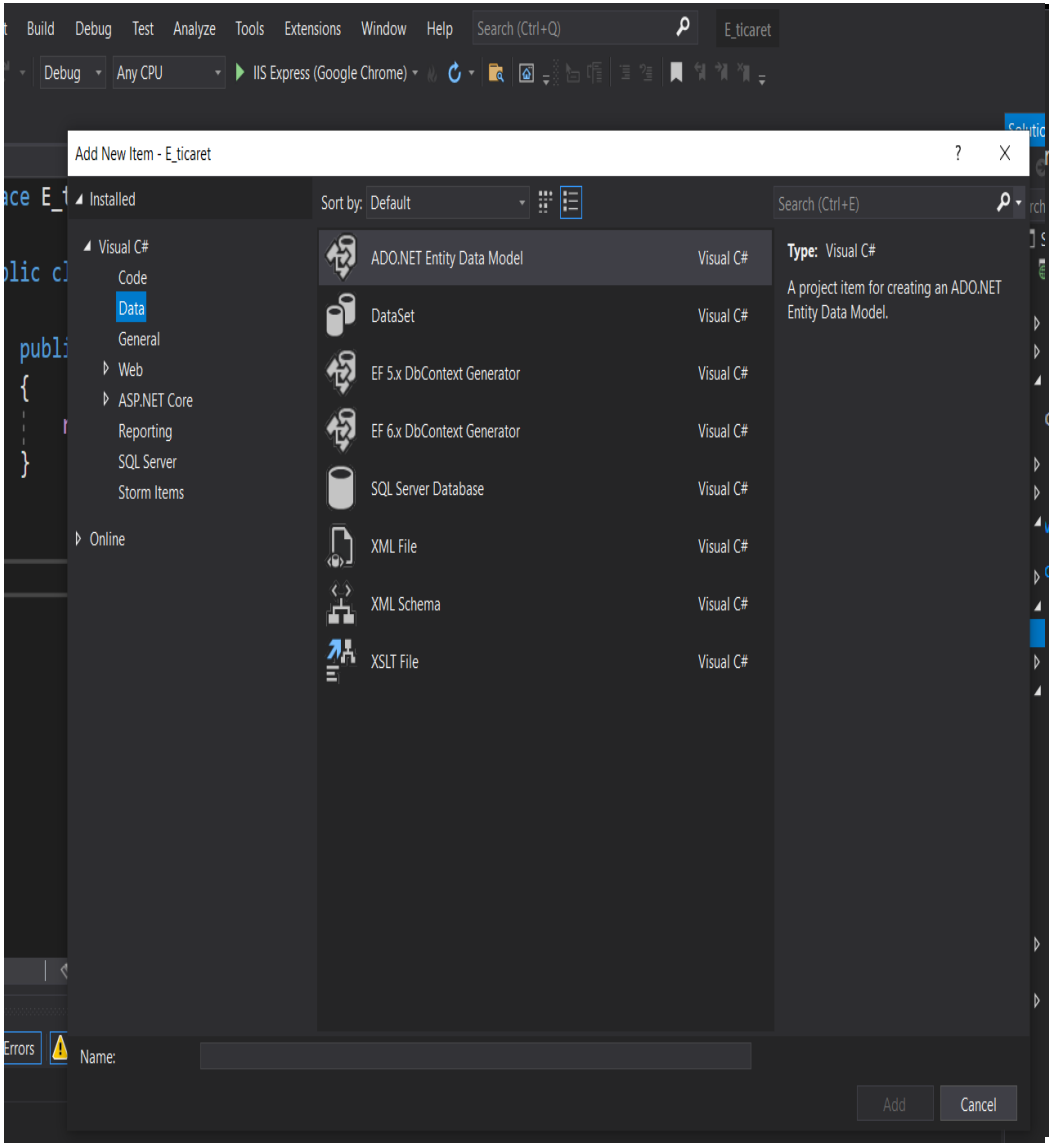
- **Code First**

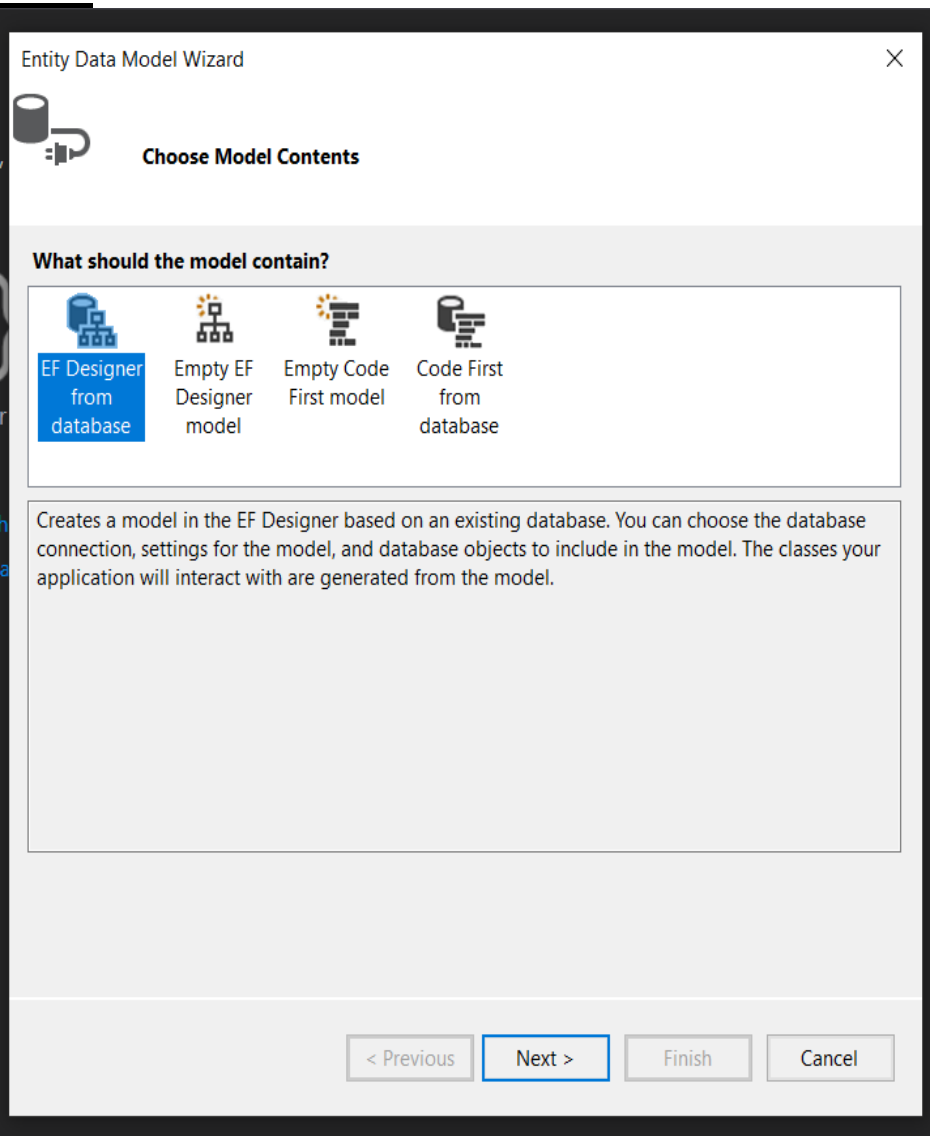
Yukarıda yer alan Database First ve Model First ile aslında yapılan görsel arayüz ile veritabanı veya model uygun kodların oluşturulmasıdır.

Code First yaklaşımında ilk olarak tabloya karşılık gelen sınıf ve özellikler yazılır.

Entity Framework ile Veritabanı Modelleme

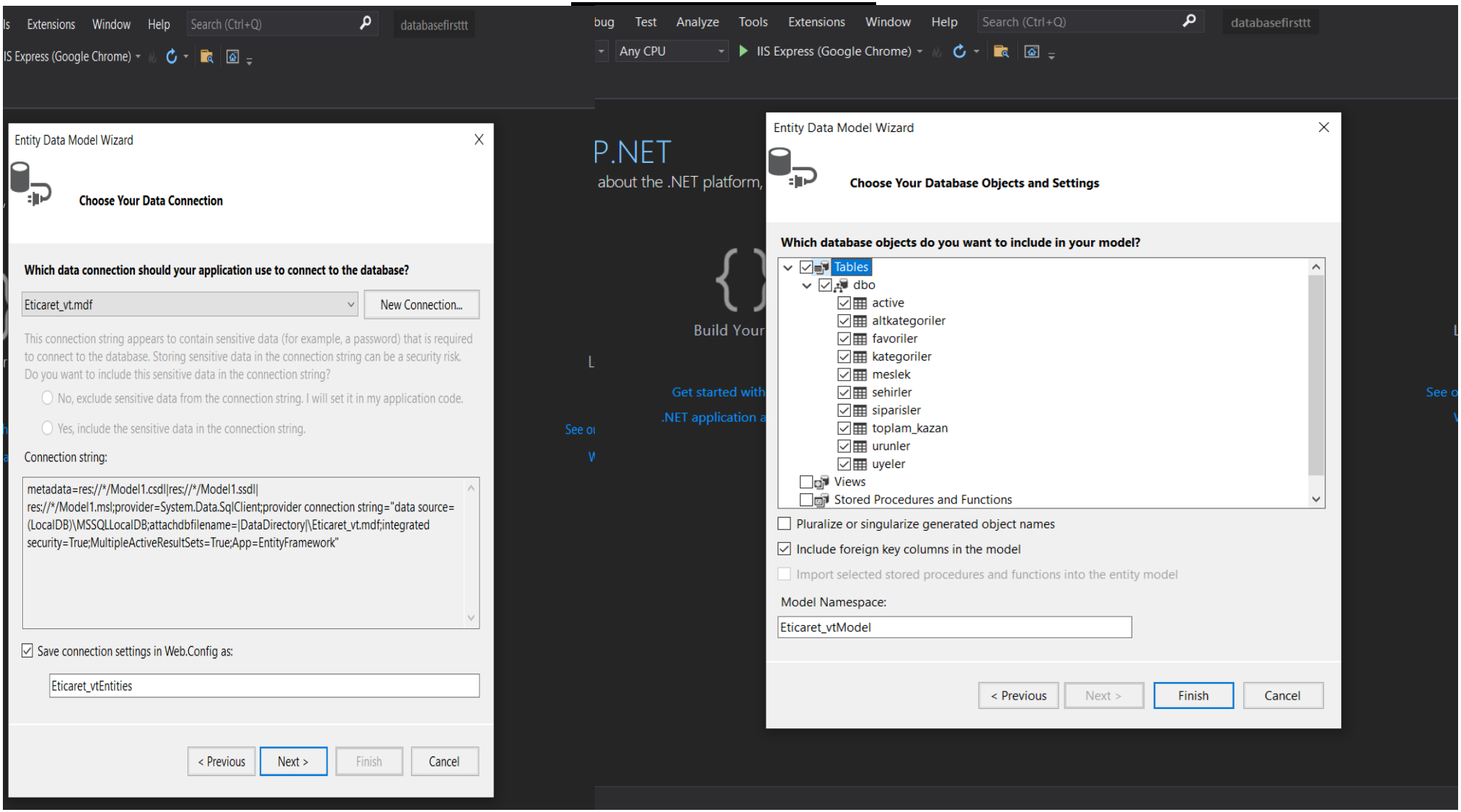
Database First





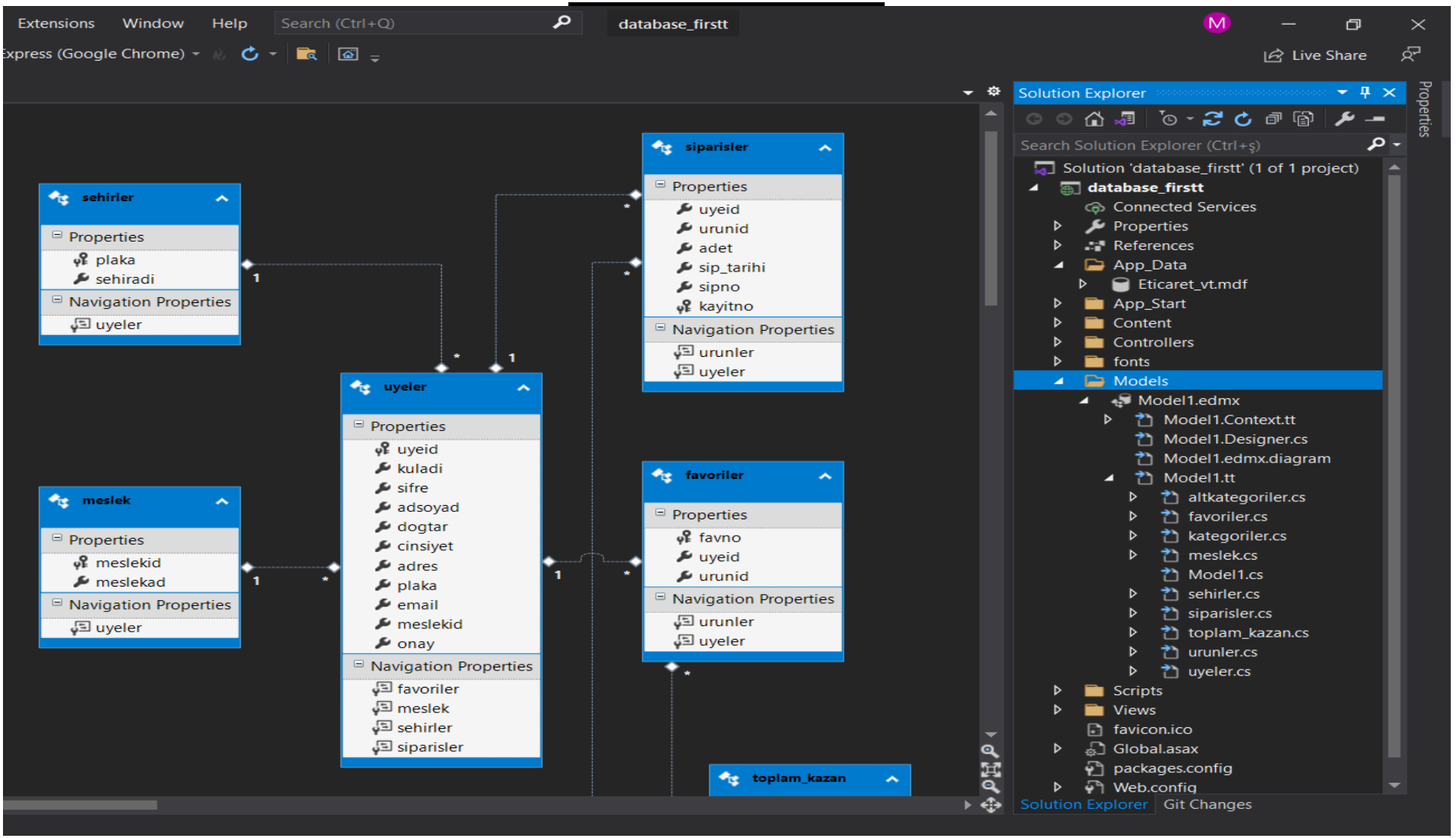
Entity Framework ile Veritabanı Modelleme

Database First



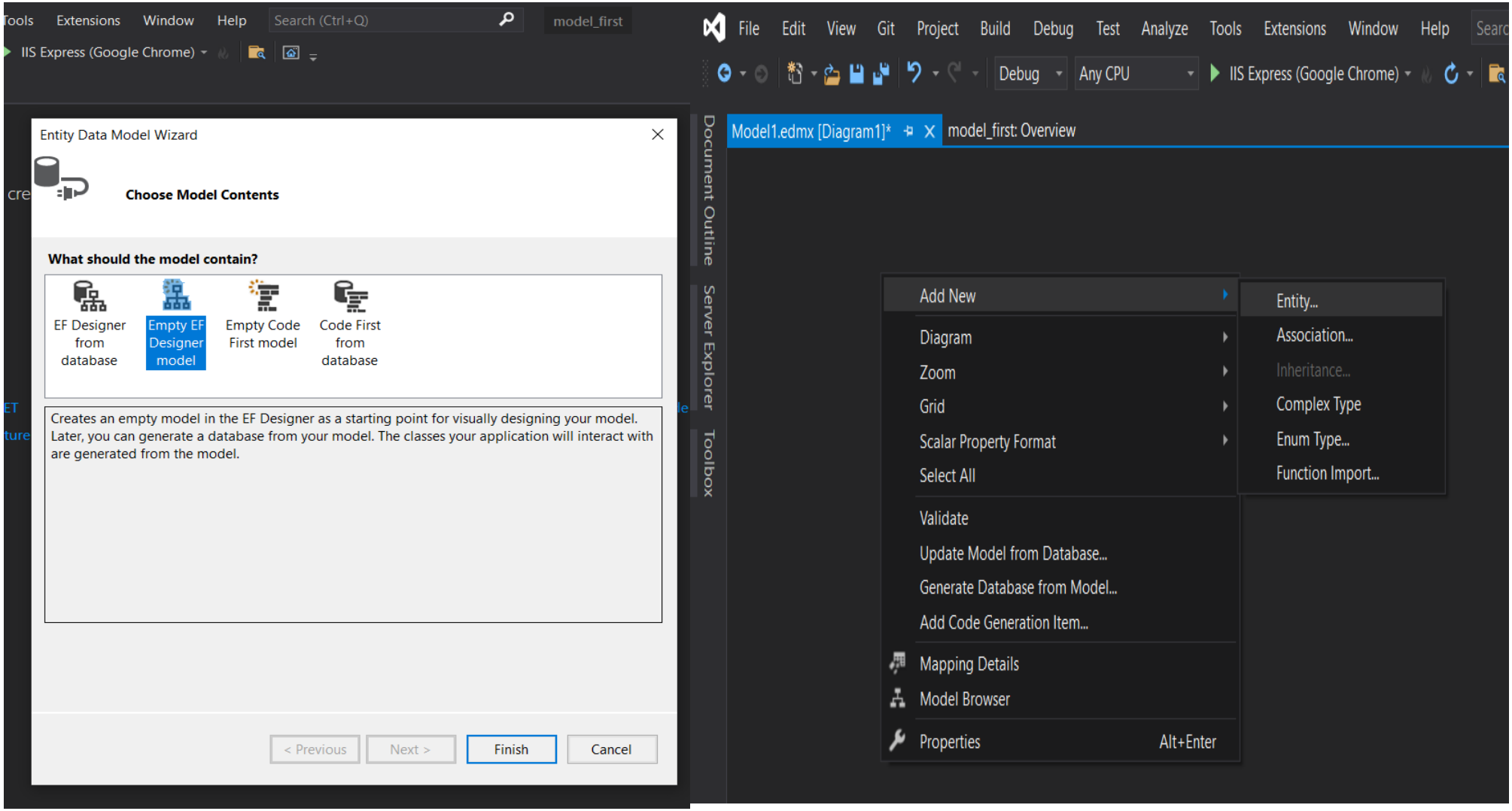
Entity Framework ile Veritabanı Modelleme

Database First

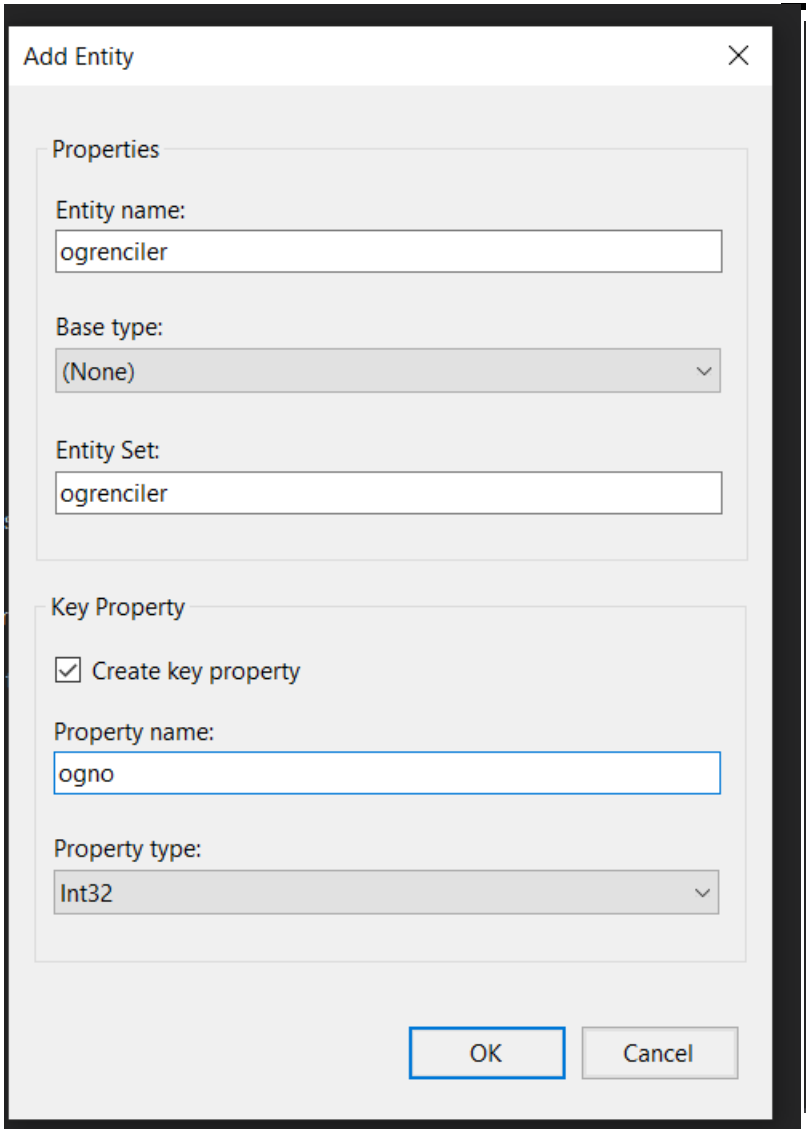


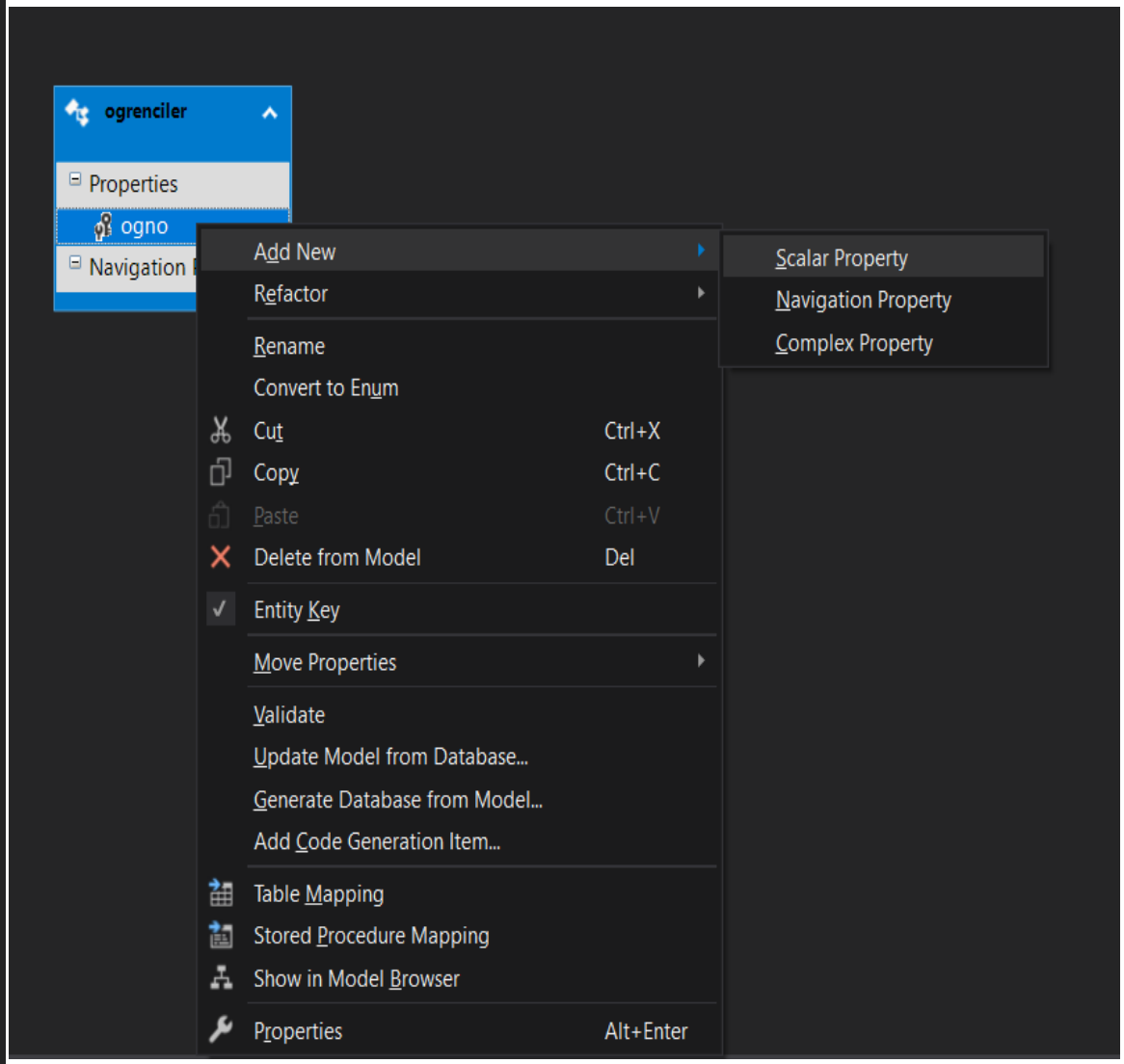
Entity Framework ile Veritabanı Modelleme

Model First



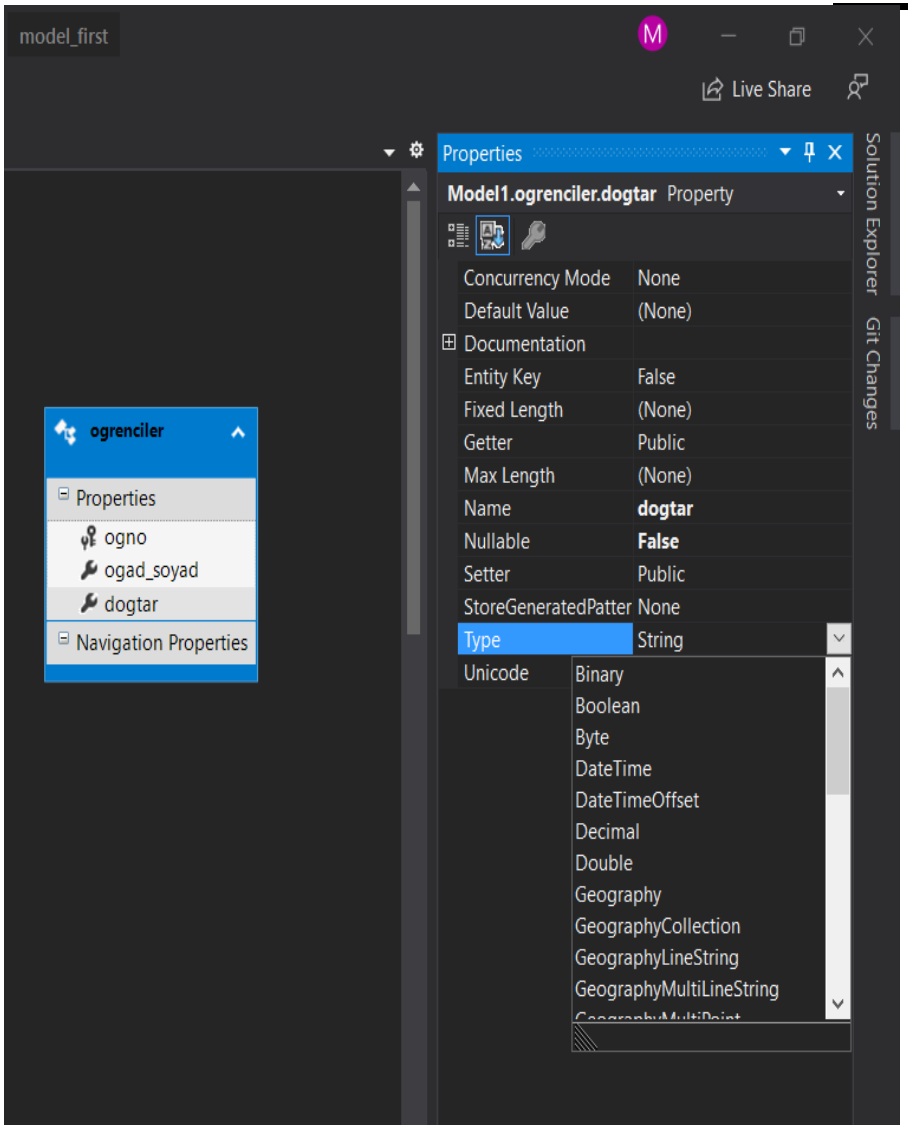
Model First

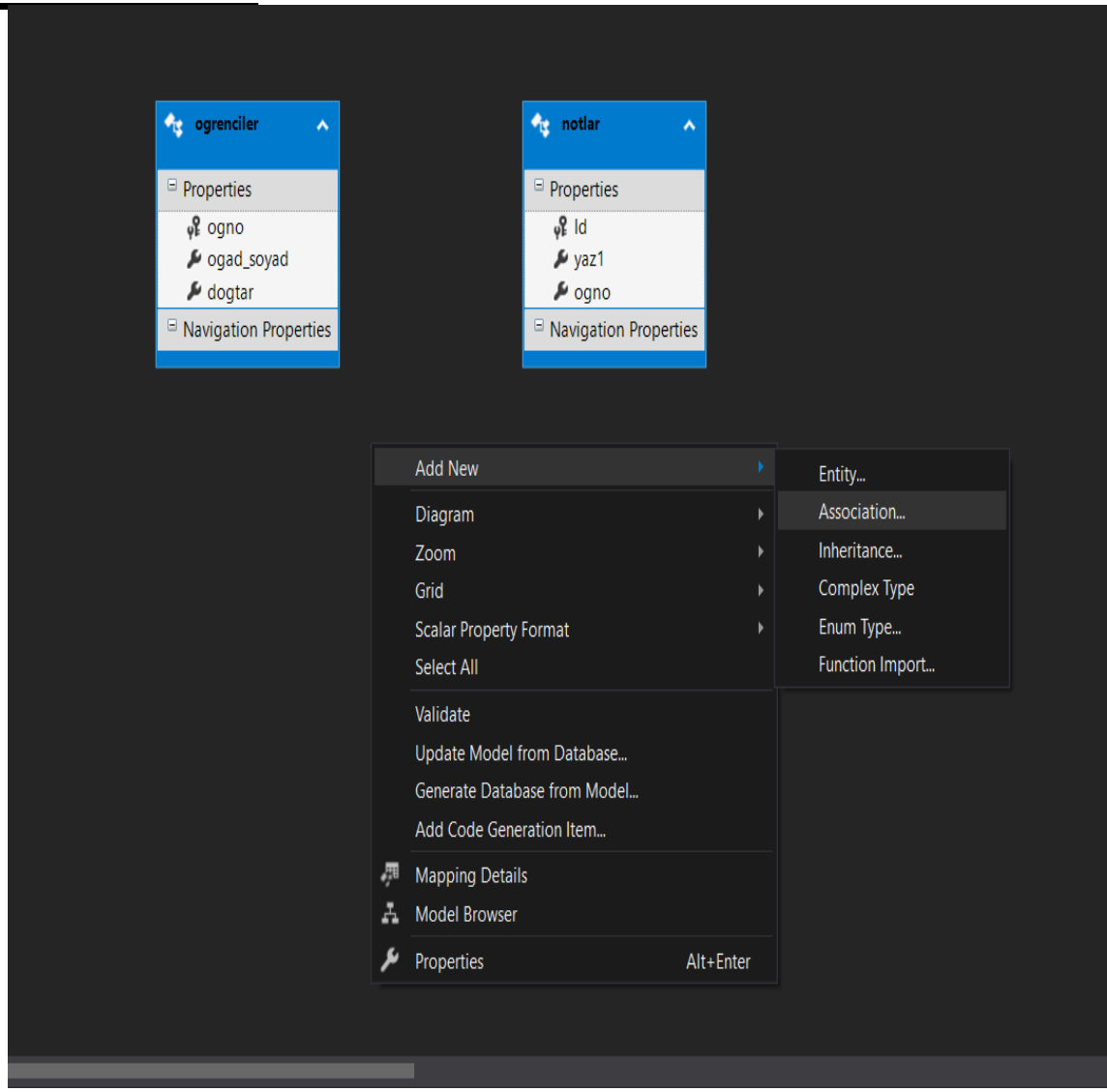

Add Entity
Properties
Entity name:
ogrenciler
Base type:
(None)
Entity Set:
ogrenciler
Key Property
Create key property
Property name:
ogno
Property type:
Int32
OK
Cancel


ogrenciler
Properties
ogno
Navigation
Add New
Refactor
Rename
Convert to Enum
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Delete from Model Del
Entity Key
Move Properties
Validate
Update Model from Database...
Generate Database from Model...
Add Code Generation Item...
Table Mapping
Stored Procedure Mapping
Show in Model Browser
Properties Alt+Enter
Scalar Property
Navigation Property
Complex Property

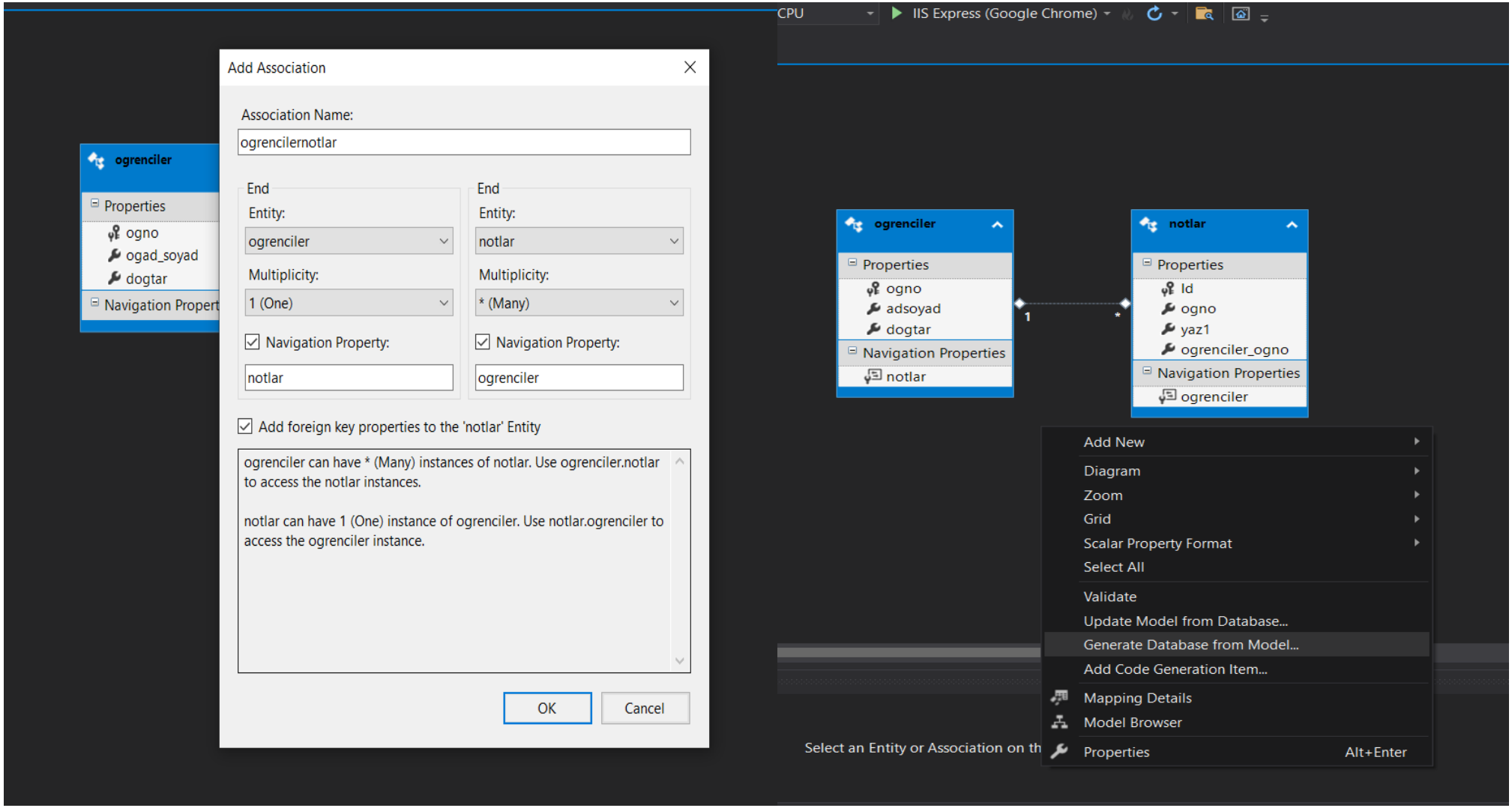
Entity Framework ile Veritabanı Modelleme

Model First

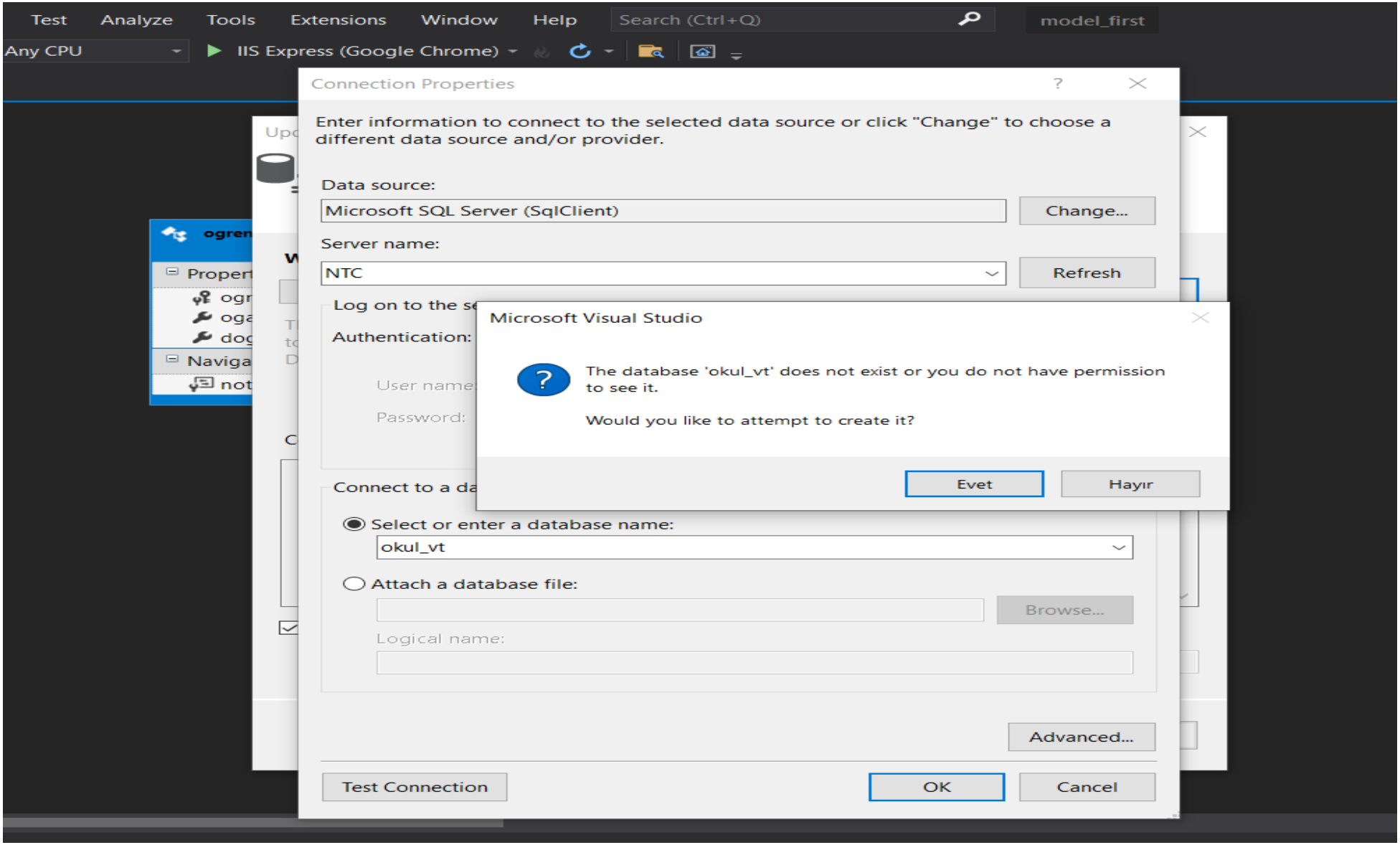




Model First

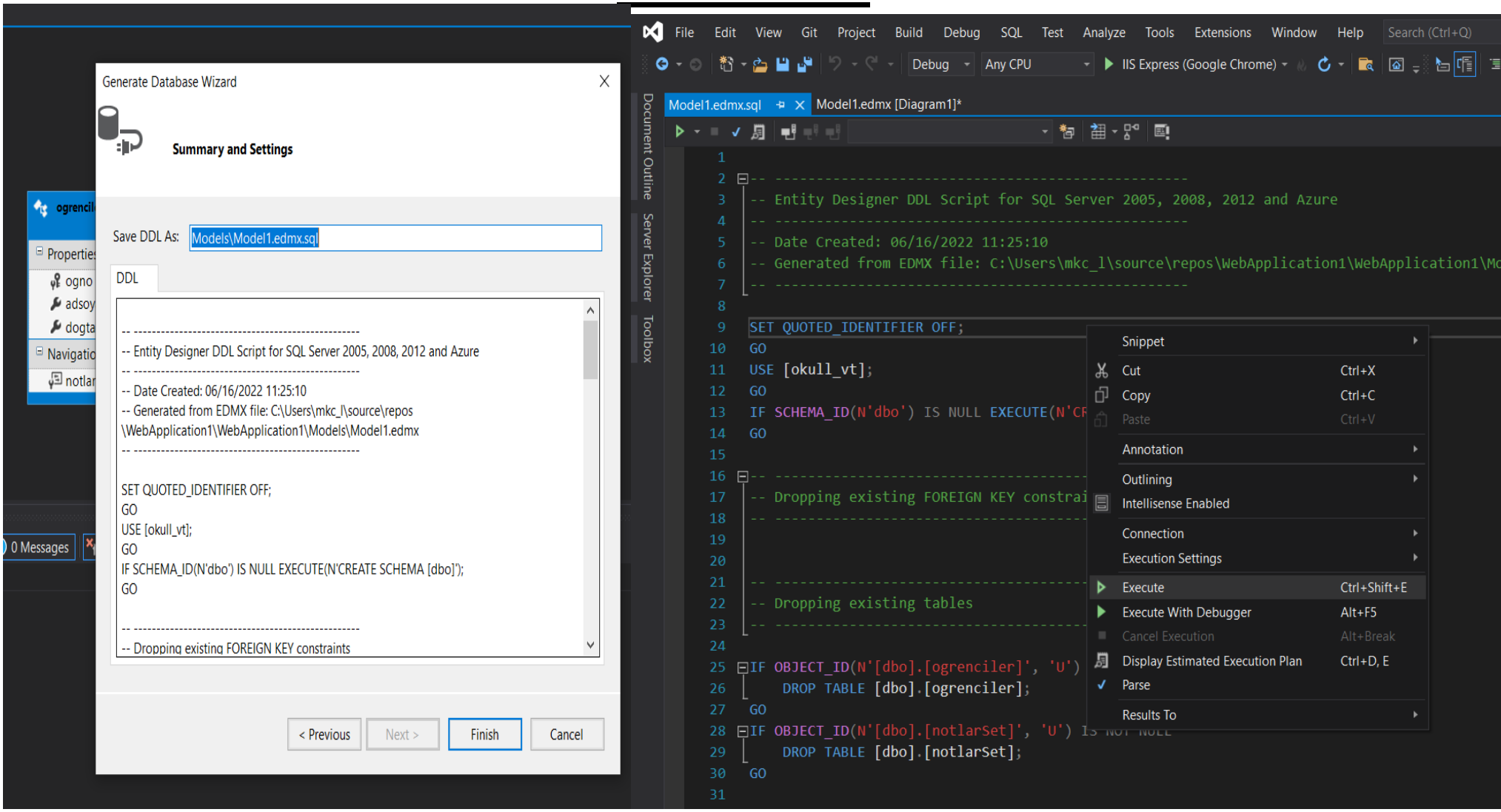

Add Association
Association Name:
ogrencilernotlar
End
Entity:
ogrenciler
Multiplicity:
1 (One)
Navigation Property:
notlar
End
Entity:
notlar
Multiplicity:
* (Many)
Navigation Property:
ogrenciler
Add foreign key properties to the 'notlar' Entity
ogrenciler can have * (Many) instances of notlar. Use ogrenciler.notlar to access the notlar instances.
notlar can have 1 (One) instance of ogrenciler. Use notlar.ogrenciler to access the ogrenciler instance.
OK
Cancel
ogrenciler
Properties
ogno
adsoyad
dogtar
Navigation Properties
notlar
notlar
Properties
Id
ogno
yaz1
ogrenciler_ogno
Navigation Properties
ogrenciler
1
*
Add New
Diagram
Zoom
Grid
Scalar Property Format
Select All
Validate
Update Model from Database...
Generate Database from Model...
Add Code Generation Item...
Mapping Details
Model Browser
Properties
Alt+Enter
Select an Entity or Association on th
CPU
IIS Express (Google Chrome)

Entity Framework ile Veritabanı Modelleme



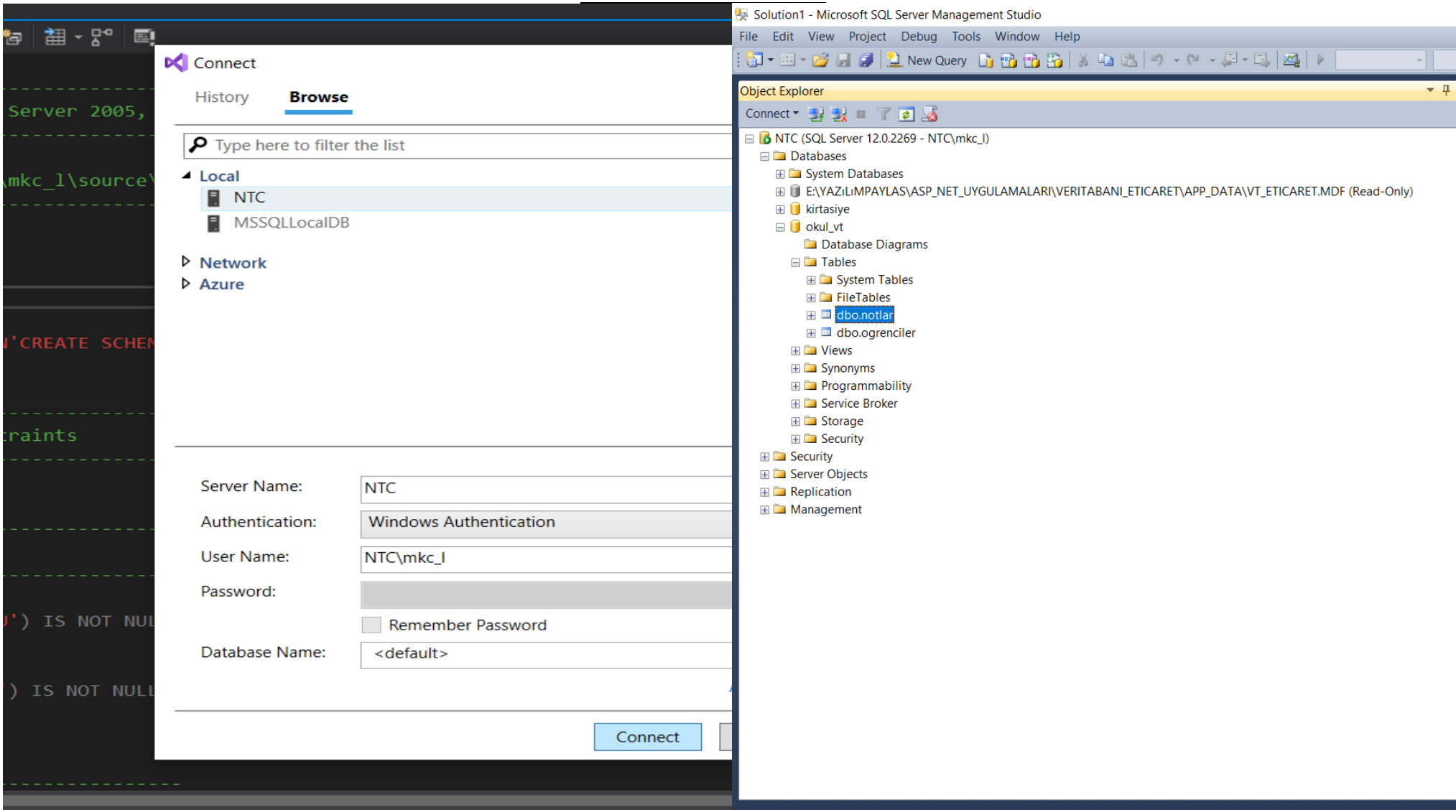
Entity Framework ile Veritabanı Modelleme

Model First



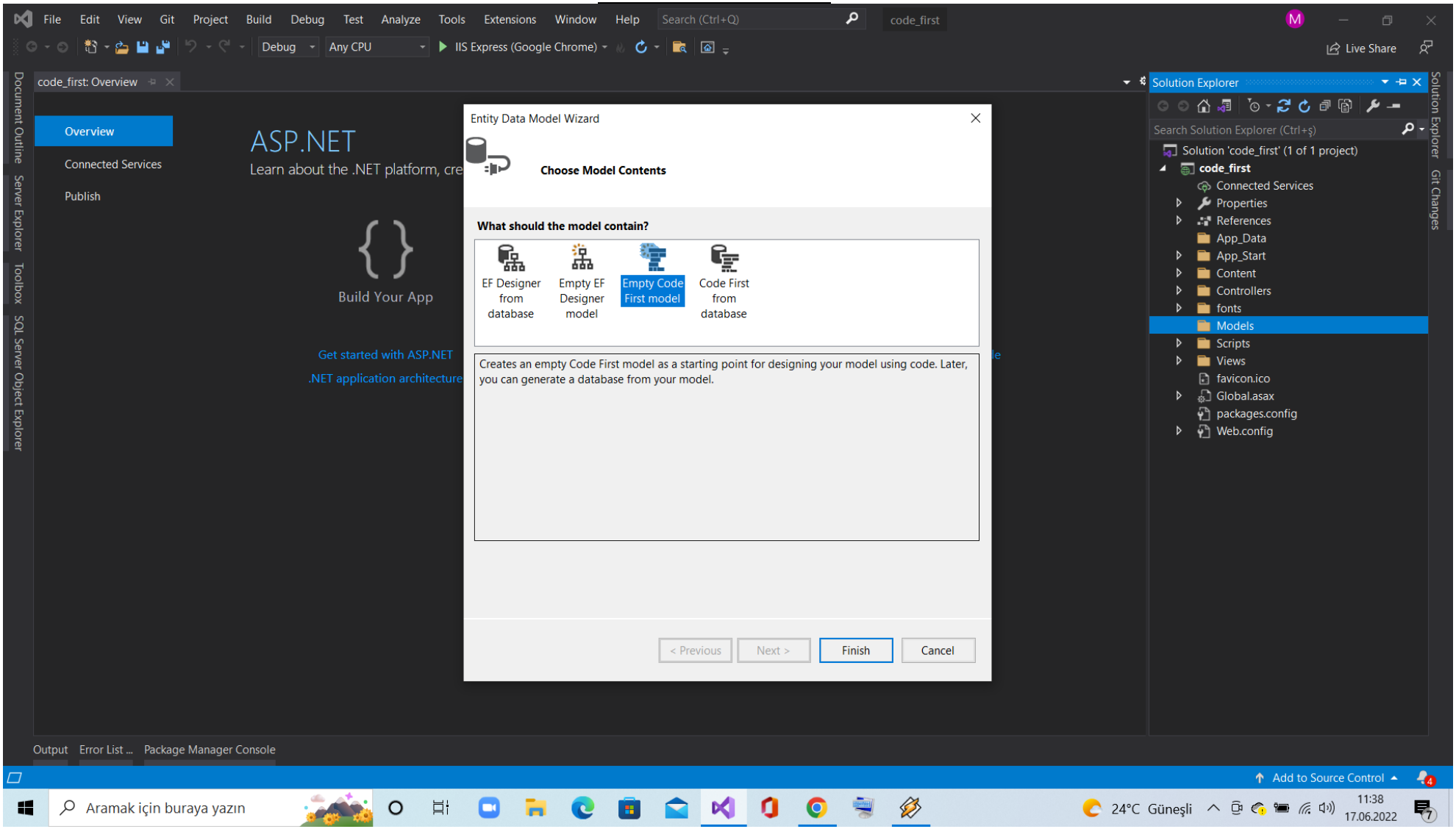
Entity Framework ile Veritabanı Modelleme

Model First

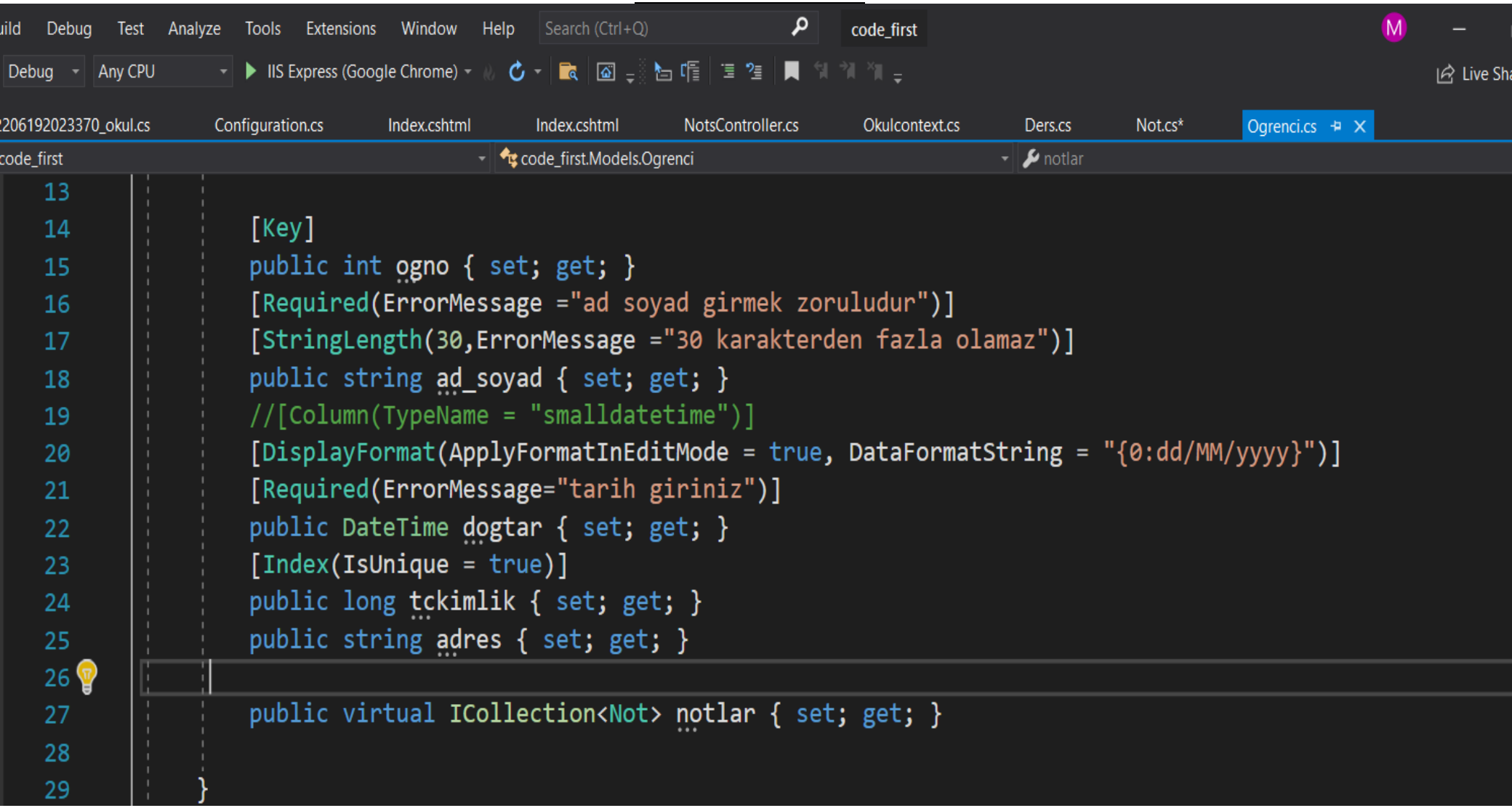


Entity Framework ile Veritabanı Modelleme

Code First



Code First


uild Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
code_first
Debug
Any CPU
IIS Express (Google Chrome)
Live Sha
2206192023370_okul.cs
Configuration.cs
Index.cshtml
Index.cshtml
NotsController.cs
Okulcontext.cs
Ders.cs
Not.cs*
Ogrenci.cs
code_first
code_first.Models.Ogrenci
notlar
13
14 [Key]
15 public int ogno { set; get; }
16 [Required(ErrorMessage ="ad soyad girmek zoruludur")]
17 [StringLength(30,ErrorMessage ="30 karakterden fazla olamaz")]
18 public string ad_soyad { set; get; }
19 //[Column(TypeName = "smalldatetime")]
20 [DisplayFormat(ApplyFormatInEditMode = true, DataFormatString = "{0:dd/MM/yyyy}")]
21 [Required(ErrorMessage="tarih giriniz")]
22 public DateTime dogtar { set; get; }
23 [Index(IsUnique = true)]
24 public long tckimlik { set; get; }
25 public string adres { set; get; }
26
27 public virtual ICollection<Not> notlar { set; get; }
28
29 }

Doğrulama Kontrolleri

- using System.ComponentModel.DataAnnotations; kütüphanesi eklenmelidir

Required	Özelliğin gerekli bir alan olduğunu belirtir.
StringLength	Dize alanı için maksimum uzunluk tanımlar
Range	Sayısal bir alan için maksimum ve minimum değer tanımlar
RegularExpression	Alan değerinin belirtilen Normal İfade ile eşleşmesi gerektiğini belirtir.
CreditCard	Belirtilen alanın bir kredi kartı numarası olduğunu belirtir.
CustomValidation	Alanı doğrulamak için belirtilen özel doğrulama yöntemi
EmailAddress	E-posta adresi formatıyla doğrulanır
FileExtension	Dosya uzantısıyla doğrulanır
MaxLength	Bir dize alanı için maksimum uzunluğu belirtir.
MinLength	Dize alanı için minimum uzunluk belirtir
Phone	Alanın telefon numaraları için normal ifadeyi kullanan bir telefon numarası olduğunu belirtir.

Doğrulama Kontrolleri

- Eğer kendimiz Custom validator yazmak istersek.Model içinde bir class oluşturur ve validationAttribute den türetiriz.

```
{
    public class Not_dogrulama:ValidationAttribute
    {

        public override bool IsValid(object value)
        {
            bool sonuc = false;
            if (Convert.ToInt32(value) >= 0 && Convert.ToInt32(value) <= 100) sonuc = true;
            return sonuc;
        }
    }
}
```

```
public class Not
{
    public int id { set; get; }
    public int ogno { set; get; }
    public byte Ders_no { set; get; }
    [Range(0,100,ErrorMessage ="0 ile 100 arasında değer girin")]
    public byte yaz1 { set; get; }
    [Not_dogrulama(ErrorMessage = "0 ile 100 arasında değer girin")]
    public byte? yaz2 { set; get; }
    public Ogrenci ogrenci { set; get; }
    public Ders Ders { get; set; }
}
```

Doğrulama Kontrolleri(Örnek mesela uye tablomuz)

- Daha sonra model içerisinde ilgili sınıfımıza gideriz.ve alanlardan önce doğrulama tanımlamaları yapılır.

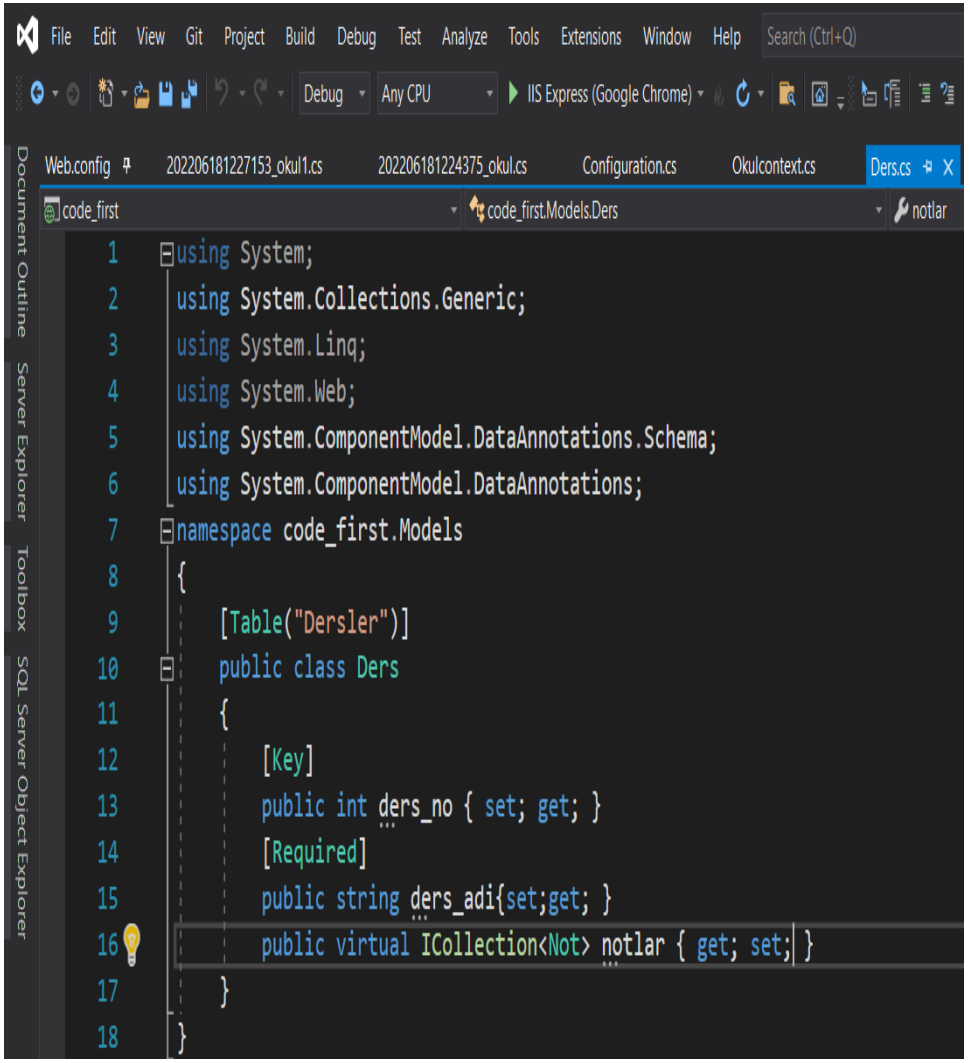
sehirler.cs | uye_kaydet.cshtml | uyeler.cs* | Index.cshtml | HomeController.cs | login_giris.cshtml | _Layout.cshtml

E_ticaret.Models.Entity.uyeler

```
        this.siparisler = new HashSet<siparisler>();
    }


    public int uyeid { get; set; }
    [Required (ErrorMessage="Kullanıcı Adı Zorunludur")]
    public string kuladi { get; set; }
    [Required (ErrorMessage ="Şifre zorunludur")]
    [MinLength(8,ErrorMessage="şifre en az 8 karakter olamlı")]
    public string sifre { get; set; }
    [Required (ErrorMessage ="Ad soyad Zorunludur")]
    public string adsoyad { get; set; }
    [Required(ErrorMessage ="tarih girilmeli")]
    [DataType(DataType.Date,ErrorMessage ="tarih bilgisi giriniz")]
    public System.DateTime dogtar { get; set; }
    [Required(ErrorMessage ="cinsiyet seçilmeli")]
    public bool? cinsiyet { get; set; }
    [Required(ErrorMessage ="Adres Seçiniz")]
    public string adres { get; set; }
    //plaka doğrulaması sehirler classında çünkü compositon var
    public byte plaka { get; set; }
    [EmailAddress(ErrorMessage ="Doğru email giriniz")]
    public string email { get; set; }
    //meslekid doğrulaması meslek classında çünkü compositon var
    public short meslekid { get; set; }
    public Nullable<bool> onay { get; set; }
```

Code First

```
public class Not
{
    public int id { set; get; }
    public int ogno { set; get; }
    public byte Ders_no { set; get; }
    [Range(0,100,ErrorMessage ="0 ile 100 arasında değer girin")]
    public byte yaz1 { set; get; }
    [Not_dogrulama(ErrorMessage = "0 ile 100 arasında değer girin")]
    public byte? yaz2 { set; get; }
    public Ogrenci ogrenci { set; get; }
    public Ders Ders { get; set; }
}
```



Entity Framework ile Veritabanı Modelleme

Code First

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help Full Screen

Web.config | 202206181227153_okul1.cs | 202206181224375_okul.cs | Configuration.cs | Okulcontext.cs* | Ders.cs | Not.cs | Ogrenci.cs

code_first | code_first.Models.Okulcontext | OnModelCreating(DbModelBuil

```
4
5   namespace code_first.Models
6   {
7       public class Okulcontext : DbContext
8       {
9
10          public Okulcontext()
11              : base("name=okul_vt2")//connection string ismi verilebilir.
12                                     //Eğer verilmesse okul_context isminde bir connection string aranır
13          {
14          }
15
16
17          public virtual DbSet<Ogrenci> ogrenciler { set; get; }
18          public virtual DbSet<Not> notlar { set; get; }
19
20          protected override void OnModelCreating(DbModelBuilder modelBuilder)
21          {
22              modelBuilder.Entity<Ogrenci>()
23
24                      .HasMany(x => x.notlar)//ogrenci tablosunu notlar tablosu ile 1--çok ilişkilendir
25                      .WithRequired(x => x.Ogrenci)//öğrenci yoksa notlara veri girilemez bilgi tutarlığı
26                      .WillCascadeOnDelete(true);//ard arda sil
27              modelBuilder.Entity<Ogrenci>().Property(x => x.dogtar).HasColumnType("smalldatetime");//alan da değişiklik
28
29
30              modelBuilder.Entity<Ders>()
31                  .HasMany (x=>x.notlar)
32                  .WithRequired (x=>x.Ders)
33                  .WillCascadeOnDelete(false);
34
```

Code First

Tasarımızı tanımladıktan sonra

- Web config içerisine <connectionStrings> içerisene

<add name="okul_vt" connectionString="data source=NTC;
AttachDbFilename=|DataDirectory|\okul_vt.mdf;integrated security=True;
MultipleActiveResultSets=True;App=EntityFramework" providerName="System.Data.SqlClient" />

Veya

<add name="okul_vt2" connectionString="data source=NTC;initial catalog=okul_vt;integrated security=True;MultipleActiveResultSets=True;App=EntityFramework"
providerName="System.Data.SqlClient" />

ekliyoruz

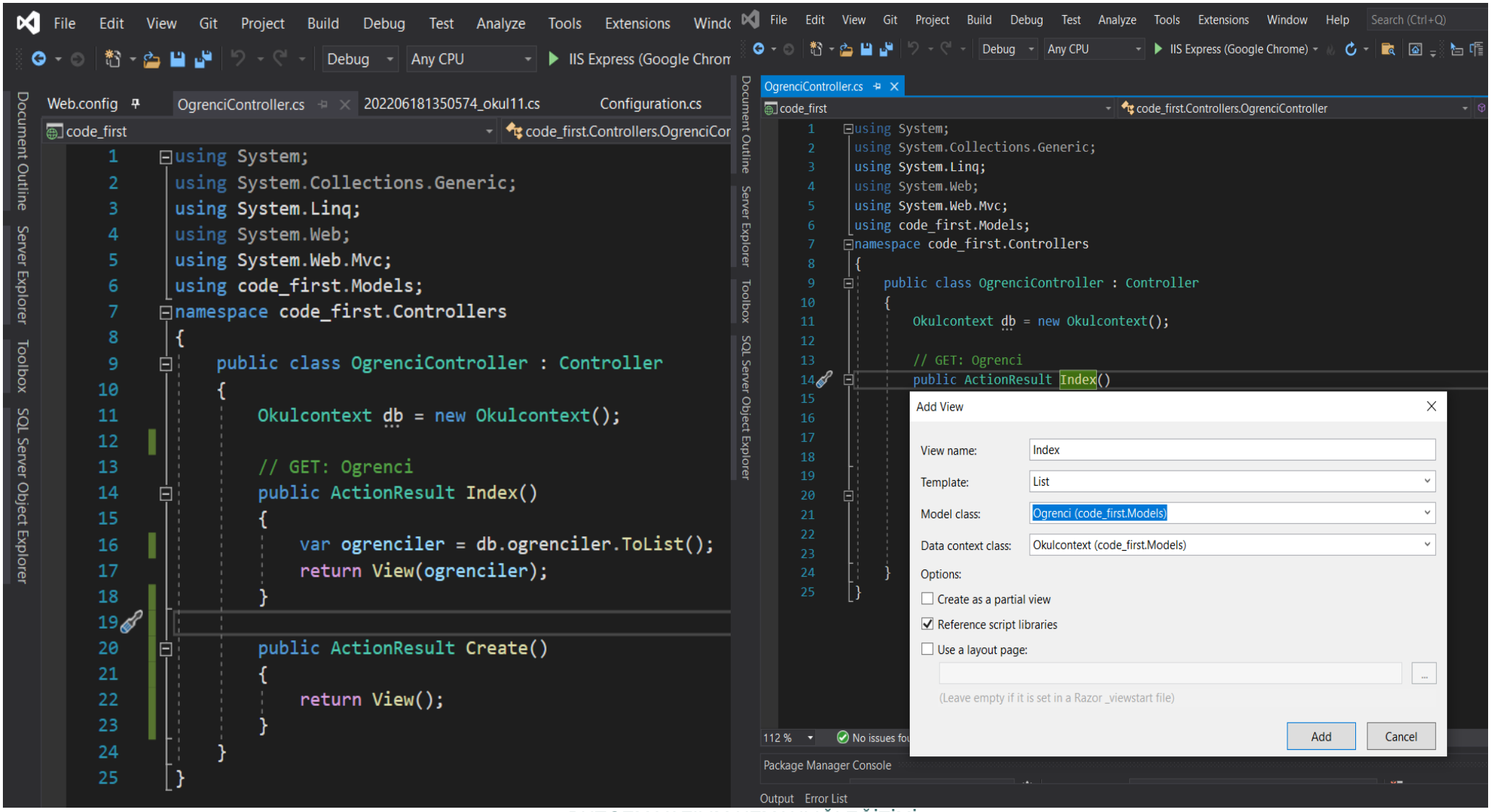
- Package Manager Console açıyoruz.
- Enable-Migrations→.migrations klasörü config oluşur
- Add-Migration okul→okul.cs(Create cümleleri oluşur)
- Update-database ile veritabanımız oluşur.

Alan eklemek veya güncellemek istersek

- Add-Migration okul2 ekliyoruz
- Update-database

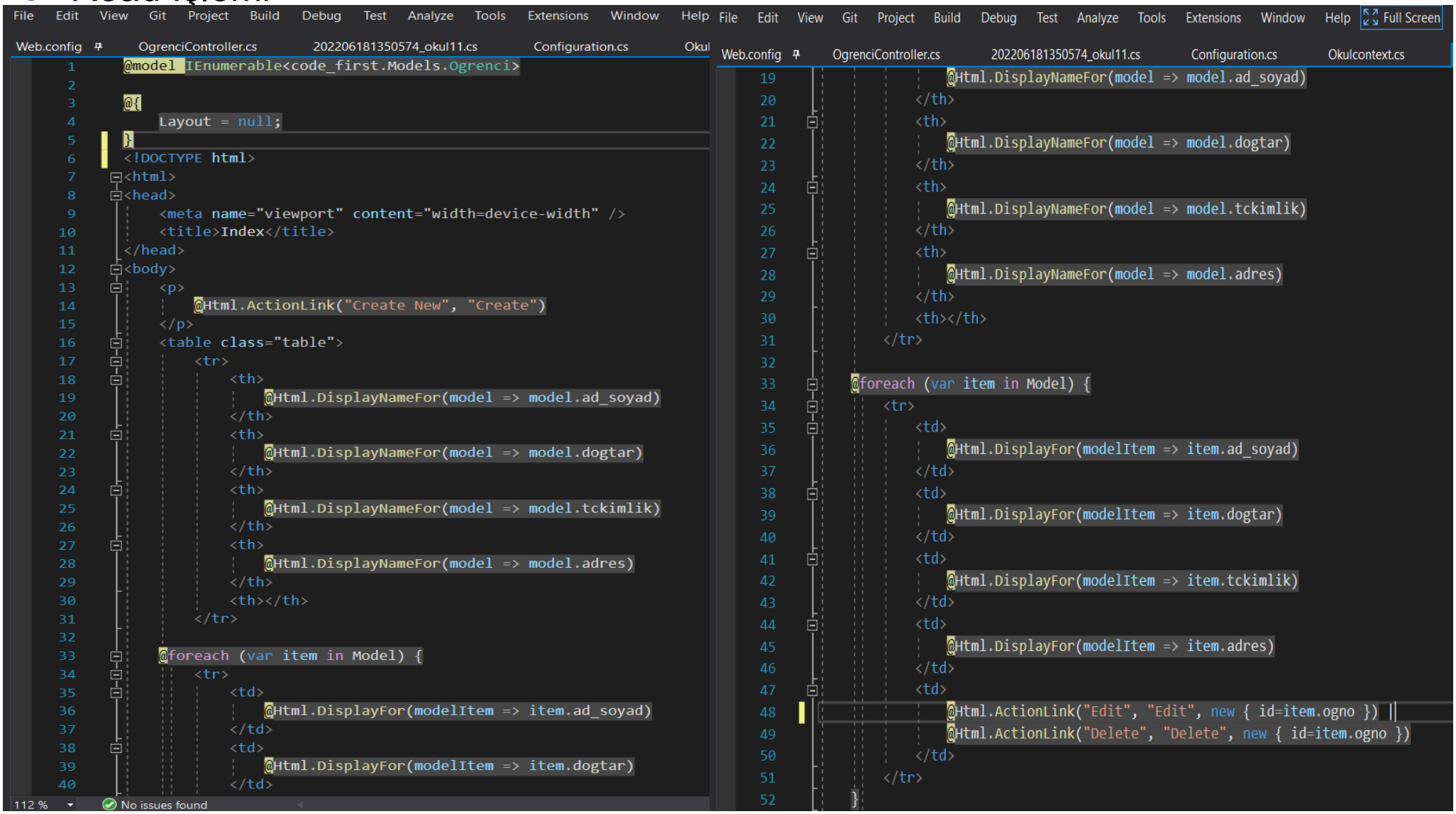
Veritabanı Üzerinde “CRUD” İşlemleri

- Read İşlemi



Veritabanı Üzerinde “CRUD” İşlemleri

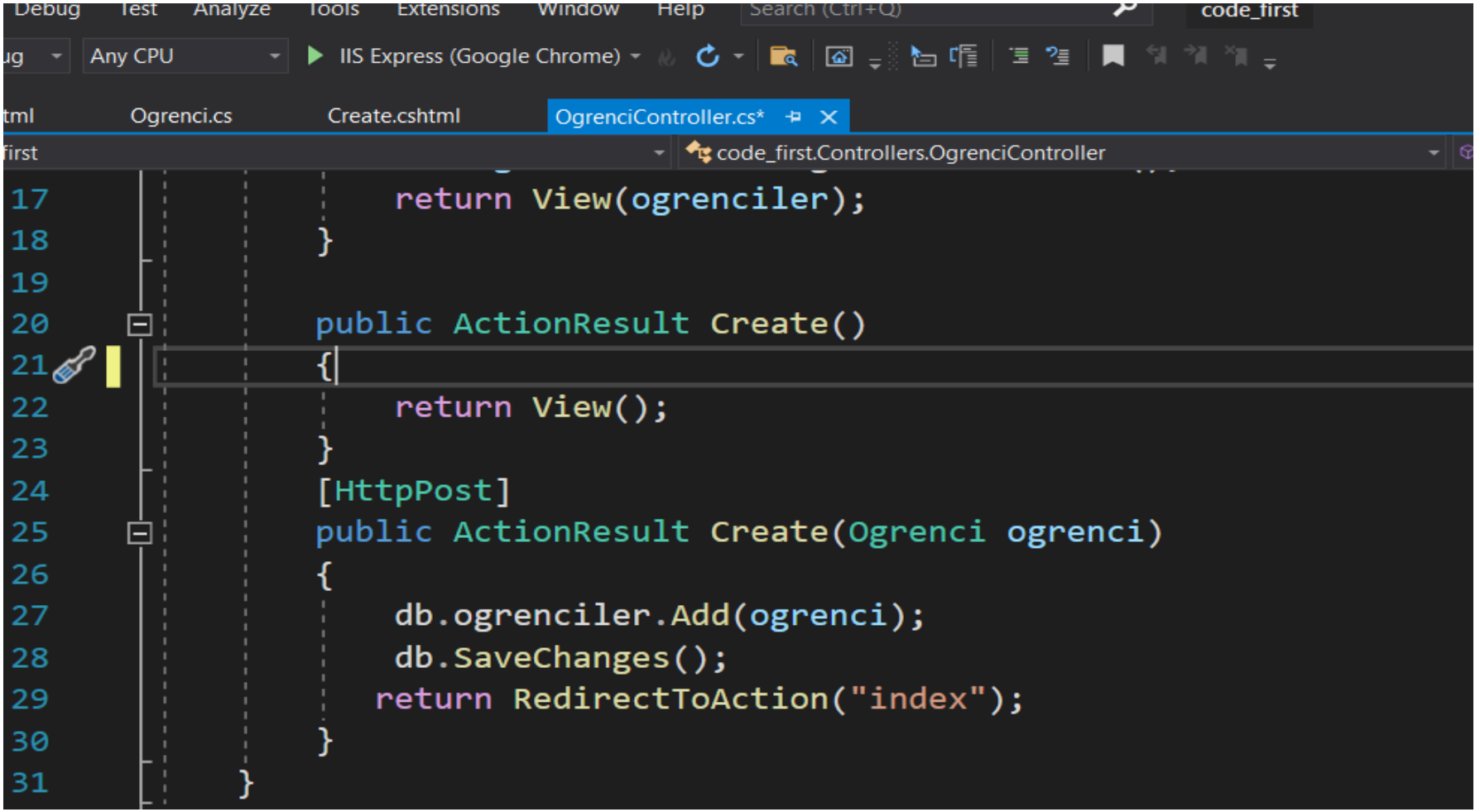
- Read İşlemi



```
@model IEnumerable<code_first.Models.Ogrenci>

@{
    Layout = null;
}
<!DOCTYPE html>
<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title>Index</title>
</head>
<body>
    <p>
        @Html.ActionLink("Create New", "Create")
    </p>
    <table class="table">
        <tr>
            <th>
                @Html.DisplayNameFor(model => model.ad_soyad)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.dogtar)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.tckimlik)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.adres)
            </th>
            <th></th>
        </tr>

    @foreach (var item in Model) {
        <tr>
            <td>
                @Html.DisplayFor(modelItem => item.ad_soyad)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.dogtar)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.tckimlik)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.adres)
            </td>
            <td>
                @Html.ActionLink("Edit", "Edit", new { id=item.ogno }) |
                @Html.ActionLink("Delete", "Delete", new { id=item.ogno })
            </td>
        </tr>
    }
```

Veritabanı Üzerinde “CRUD” İşlemleri

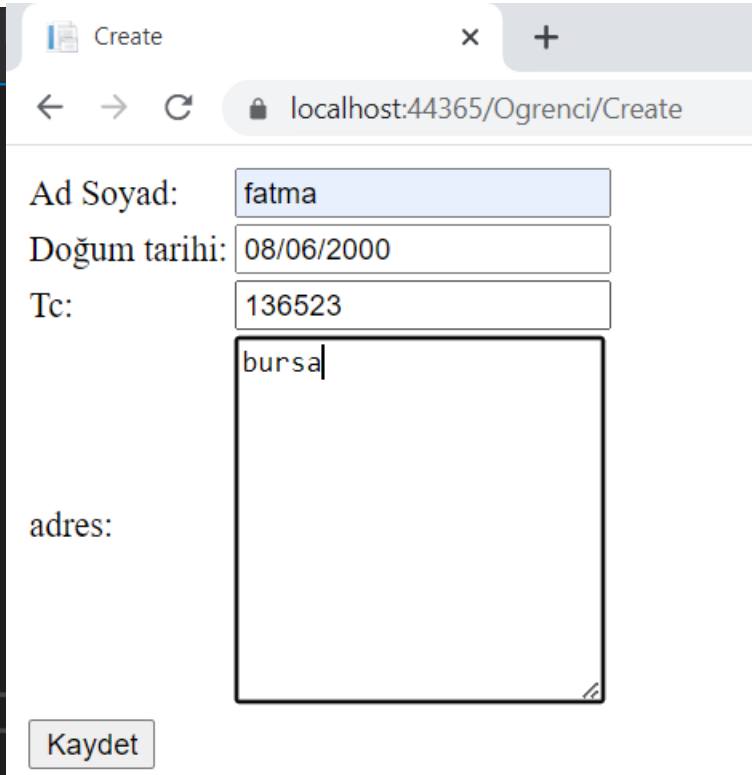
- Create İşlemi



Veritabanı Üzerinde “CRUD” İşlemleri

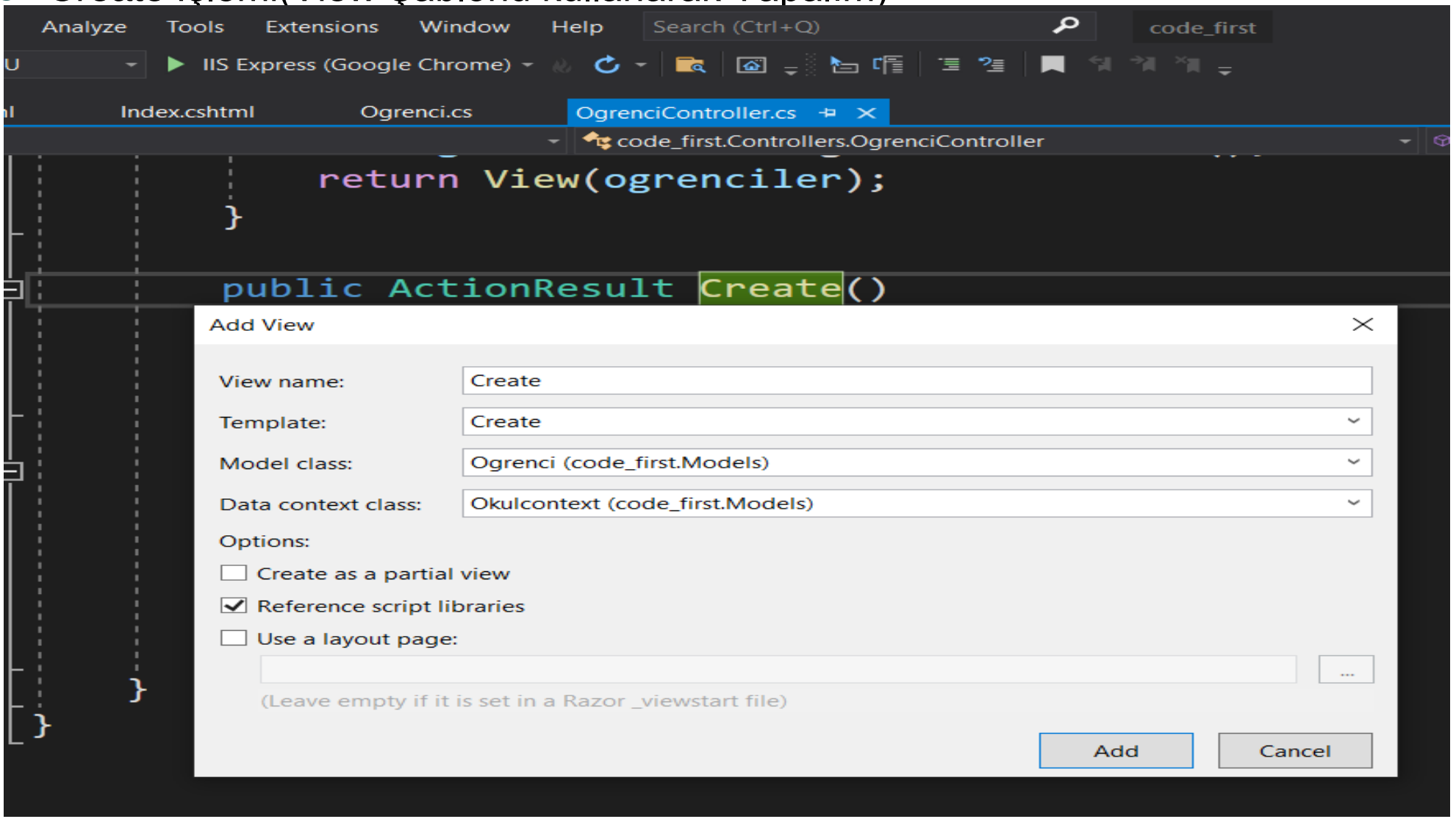
- Create İşlemi(Kendimiz Form Hazırlayabiliriz)

```
@{
    Layout = null;
}

<!DOCTYPE html>

<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title>Create</title>
</head>
<body>
    <div>
        <table>

            @using (Html.BeginForm())
            {
                <tr><td>Ad Soyad:</td><td>@Html.TextBox("ad_soyad") </td></tr>
                <tr><td>Doğum tarihi:</td><td>@Html.TextBox("dogtar") </td></tr>
                <tr><td>Tc:</td><td>@Html.TextBox("tckimlik") </td></tr>
                <tr><td>adres:</td><td>@Html.TextArea("adres", "", 10, 20, new { }) </td></tr>
                <tr><td><input id="Submit1" type="submit" value="Kaydet" /></td></tr>


            }
        </table>
```



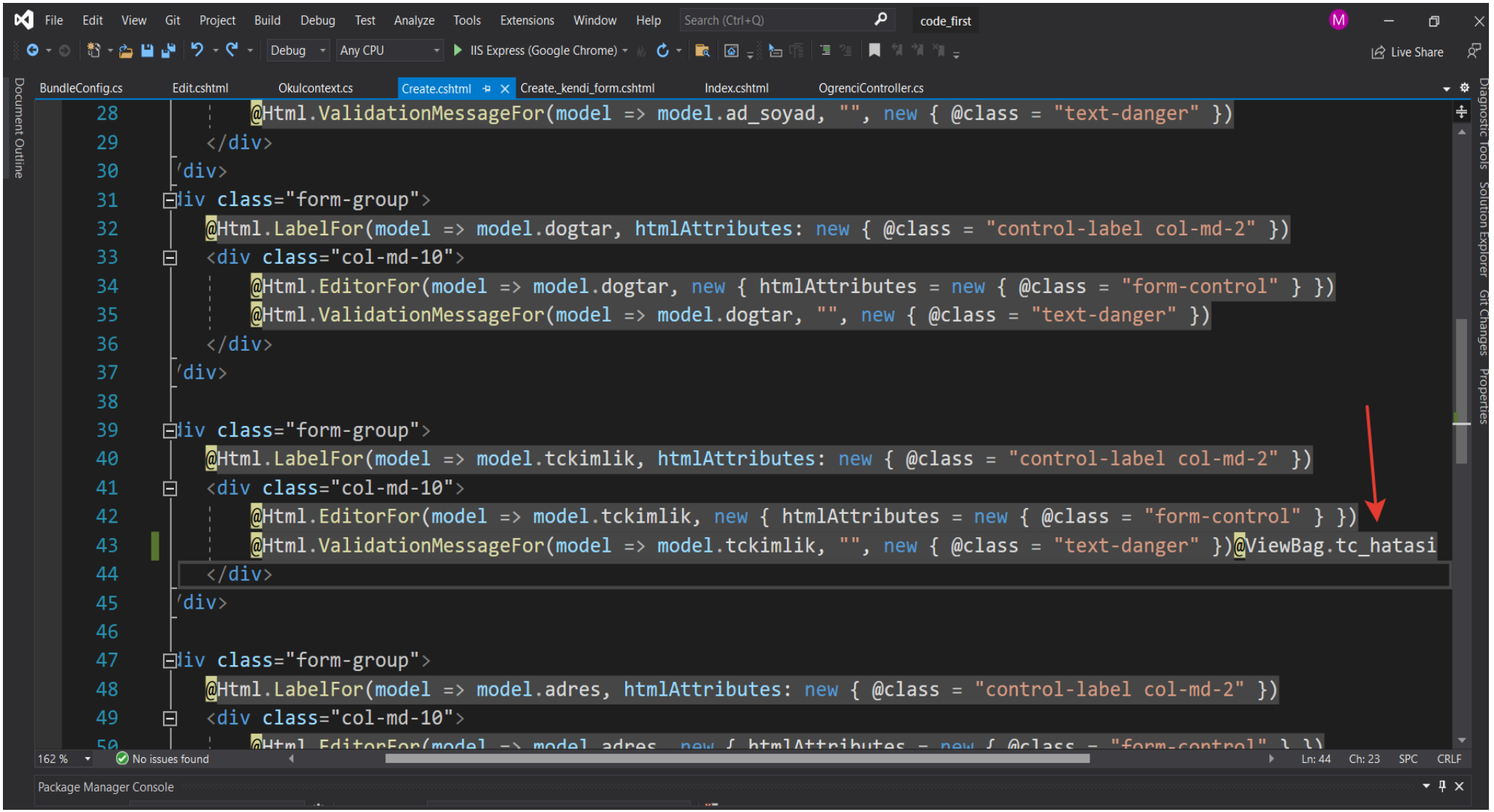
Veritabanı Üzerinde “CRUD” İşlemleri

- Create İşlemi(View Şablonu kullanarak Yapalım)



Veritabanı Üzerinde “CRUD” İşlemleri

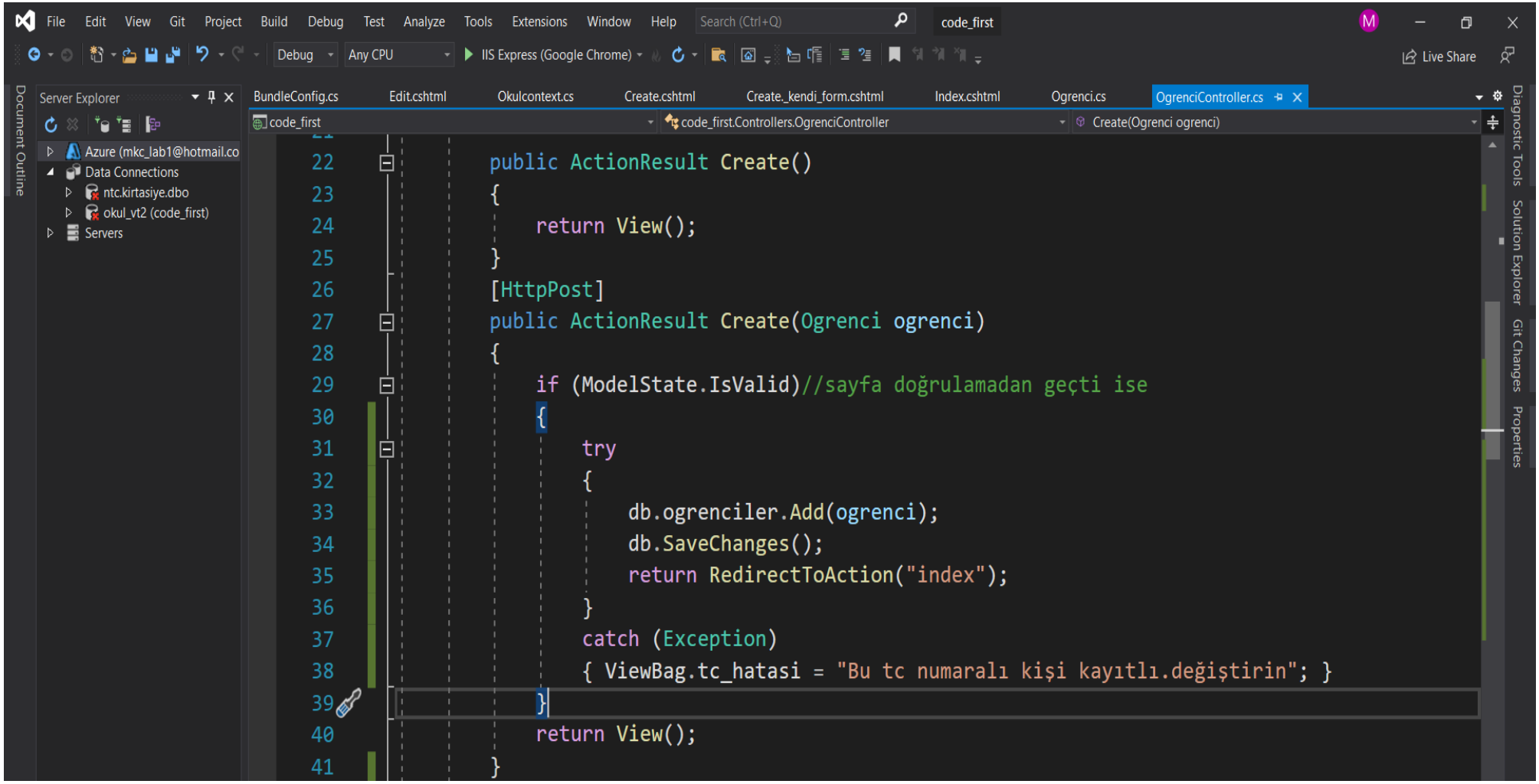
- Create İşlemi(View Şablonu kullanarak Yapalım)



Veritabanı Üzerinde “CRUD” İşlemleri

- Create İşlemi(View Şablonu kullanarak Yapalım)

Kayıt Action nımızda doğrulama kontrolü kodlarımızı ekleyeceğiz..



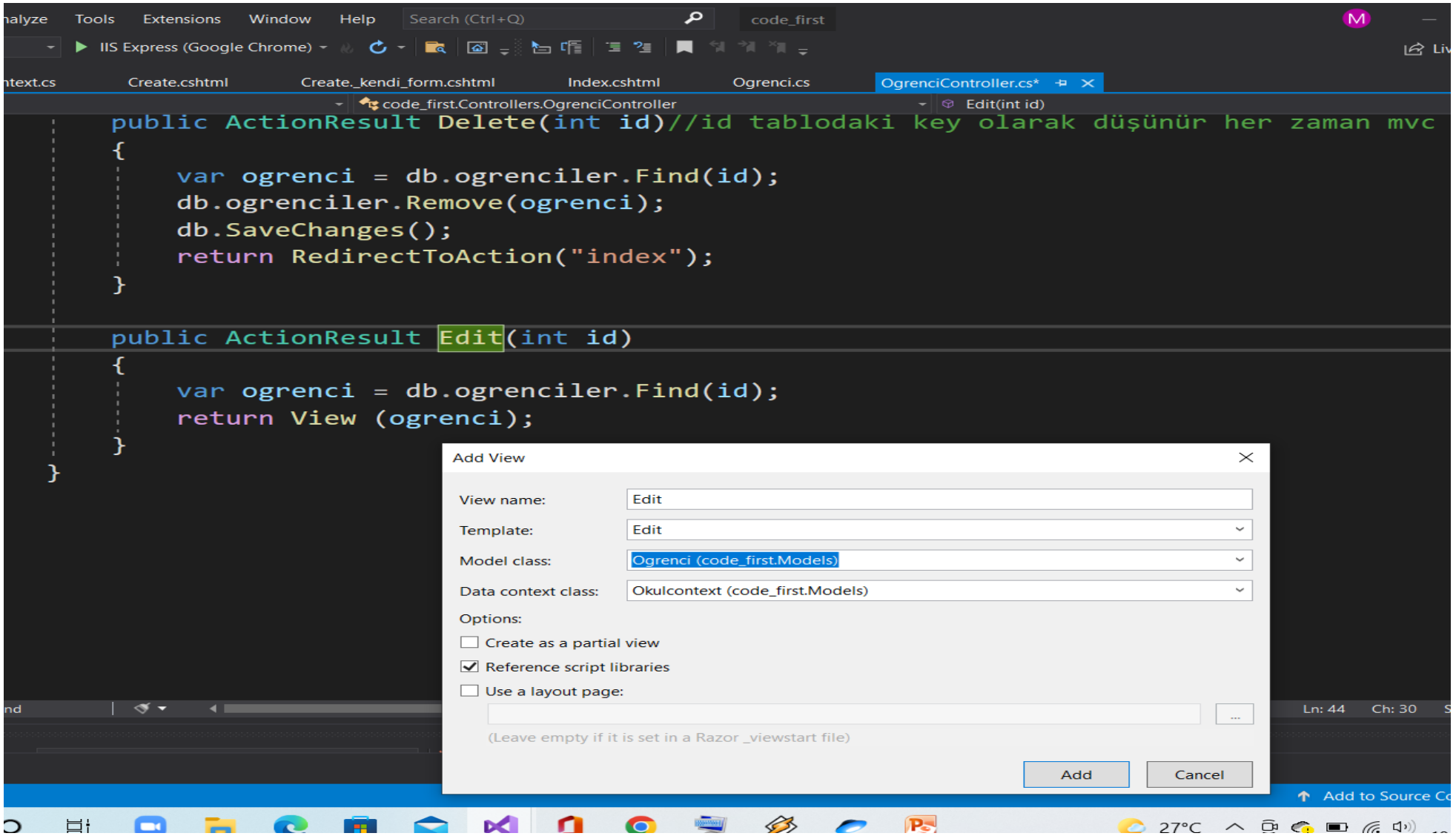
Veritabanı Üzerinde “CRUD” İşlemleri

- Delete İşlemi

```
    }

    public ActionResult Delete(int id)//id tablodaki key olarak düşünür her zaman mvc
    {
        var ogrenci = db.ogrenciler.Find(id);
        db.ogrenciler.Remove(ogrenci);
        db.SaveChanges();
        return RedirectToAction("index");
```

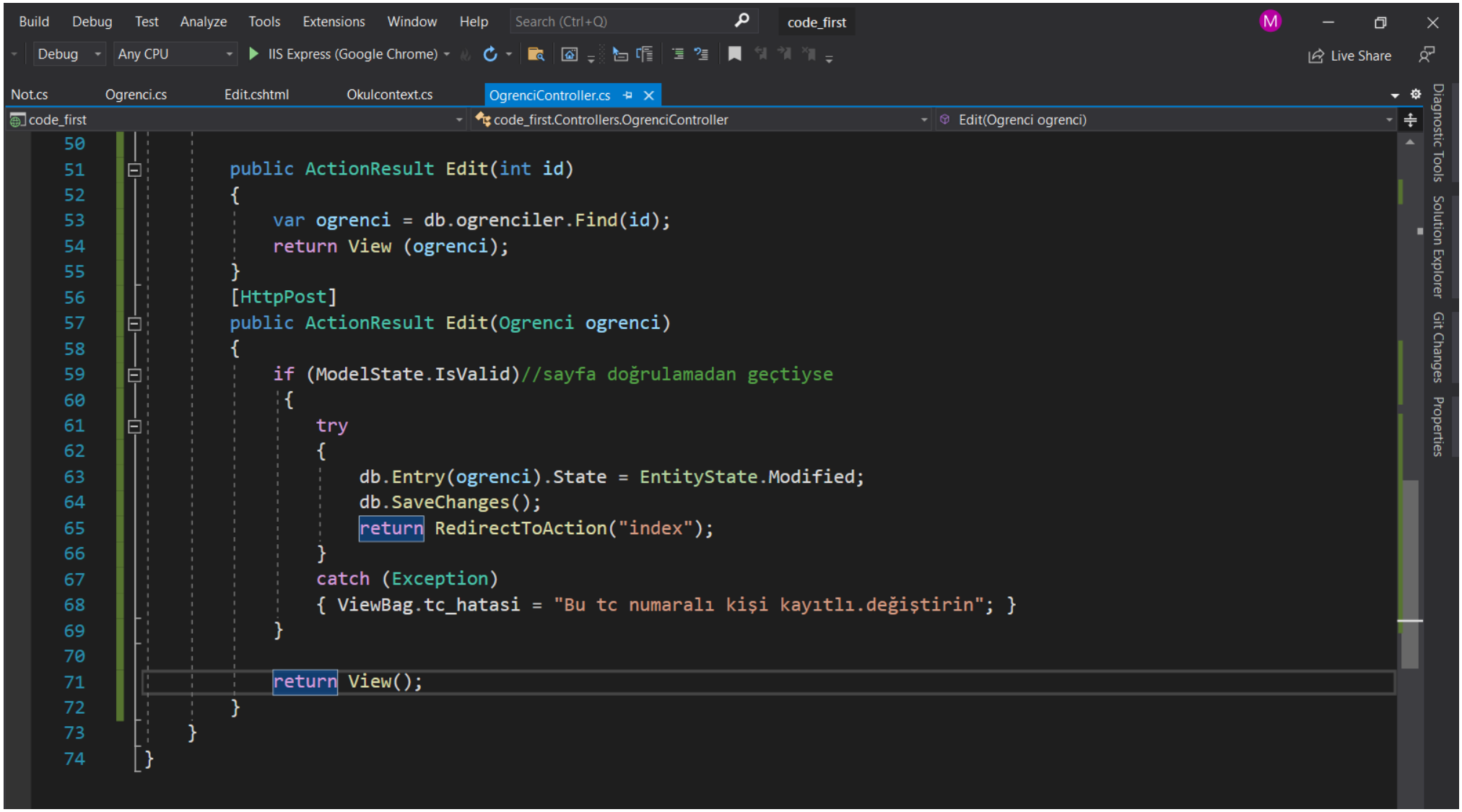
Veritabanı Üzerinde “CRUD” İşlemleri

- Update İşlemi



Veritabanı Üzerinde “CRUD” İşlemleri

- Update İşlemi



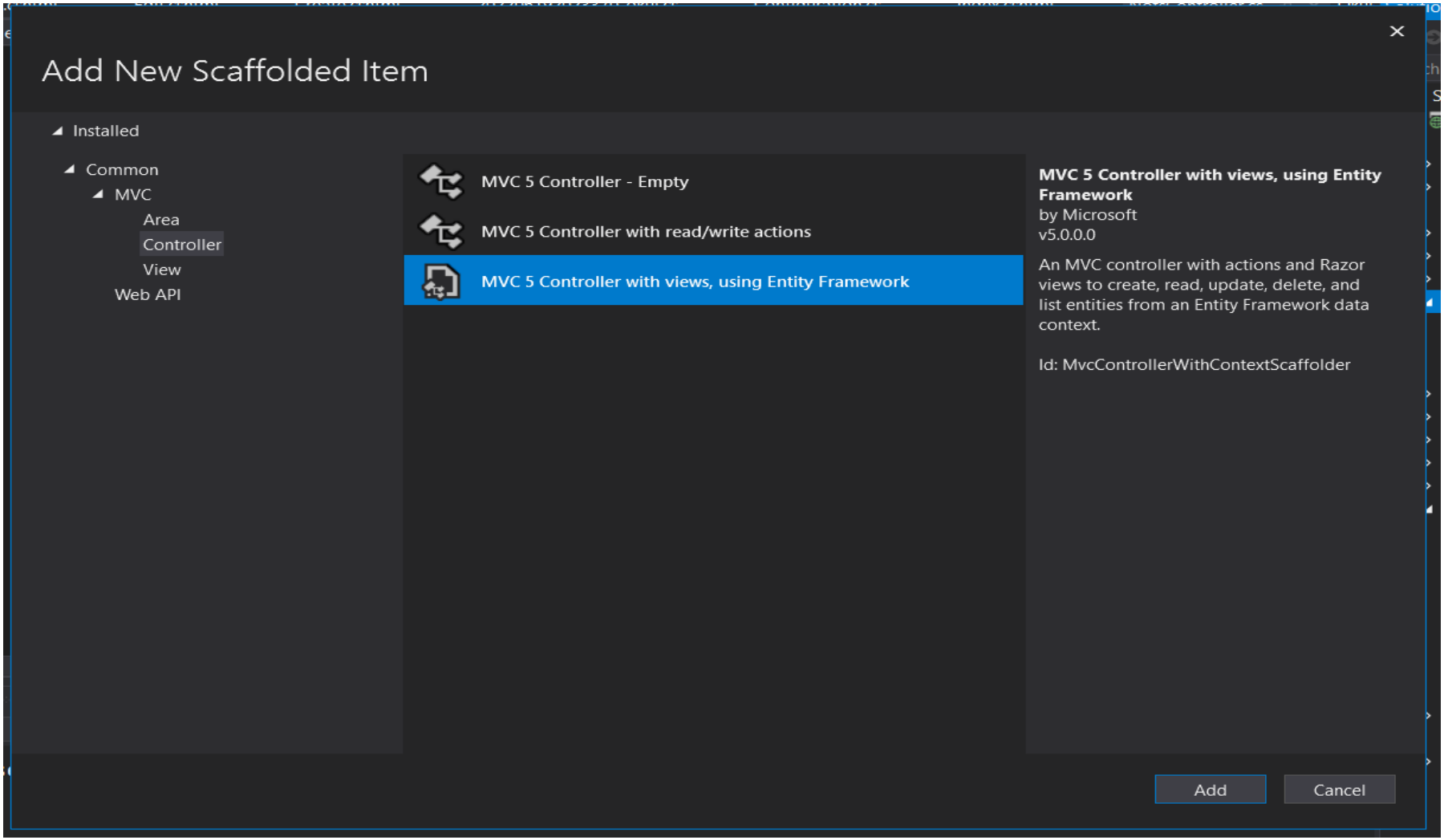

```
50
51          public ActionResult Edit(int id)
52          {
53              var ogrenci = db.ogrenciler.Find(id);
54              return View (ogrenci);
55          }
56          [HttpPost]
57          public ActionResult Edit(Ogrenci ogrenci)
58          {
59              if (ModelState.IsValid)//sayfa doğrulamadan geçtiyse
60              {
61                  try
62                  {
63                      db.Entry(ogrenci).State = EntityState.Modified;
64                      db.SaveChanges();
65                      return RedirectToAction("index");
66                  }
67                  catch (Exception)
68                  { ViewBag.tc_hatasi = "Bu tc numaralı kişi kayıtlı.değiştirin"; }
69              }
70
71              return View();
72          }
73      }
74  }
```

Veritabanı Üzerinde “CRUD” İşlemleri

- Dersler Tablosu için Gerekli Action ve View leri siz hazırlayınız.

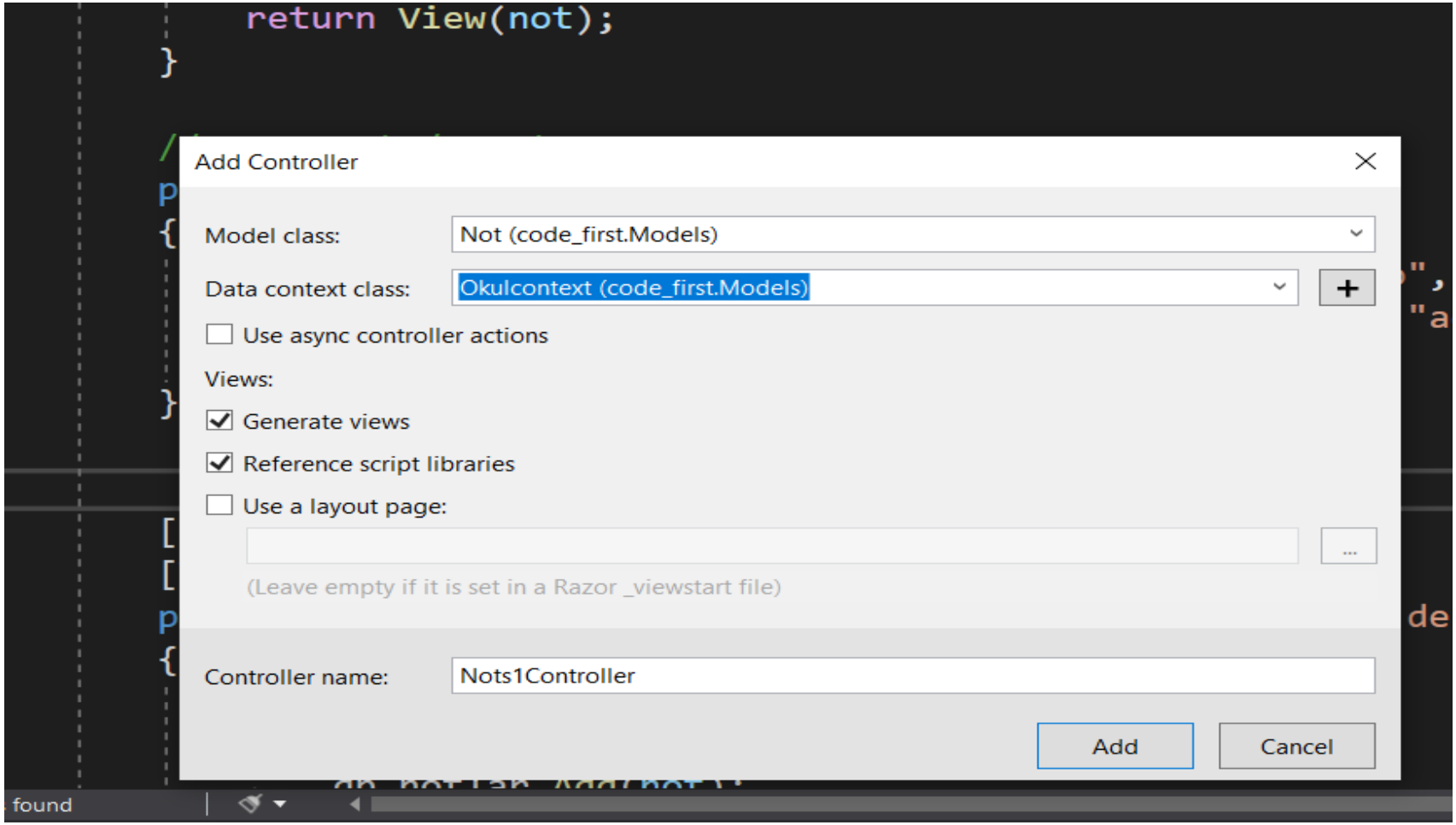
Veritabanı Üzerinde “CRUD” İşlemleri

Birden Fazla Tablodan Veri Çekme ve sihirbazla Crud Action Ve Viewlar Oluşturma (Notlar Tablosu)



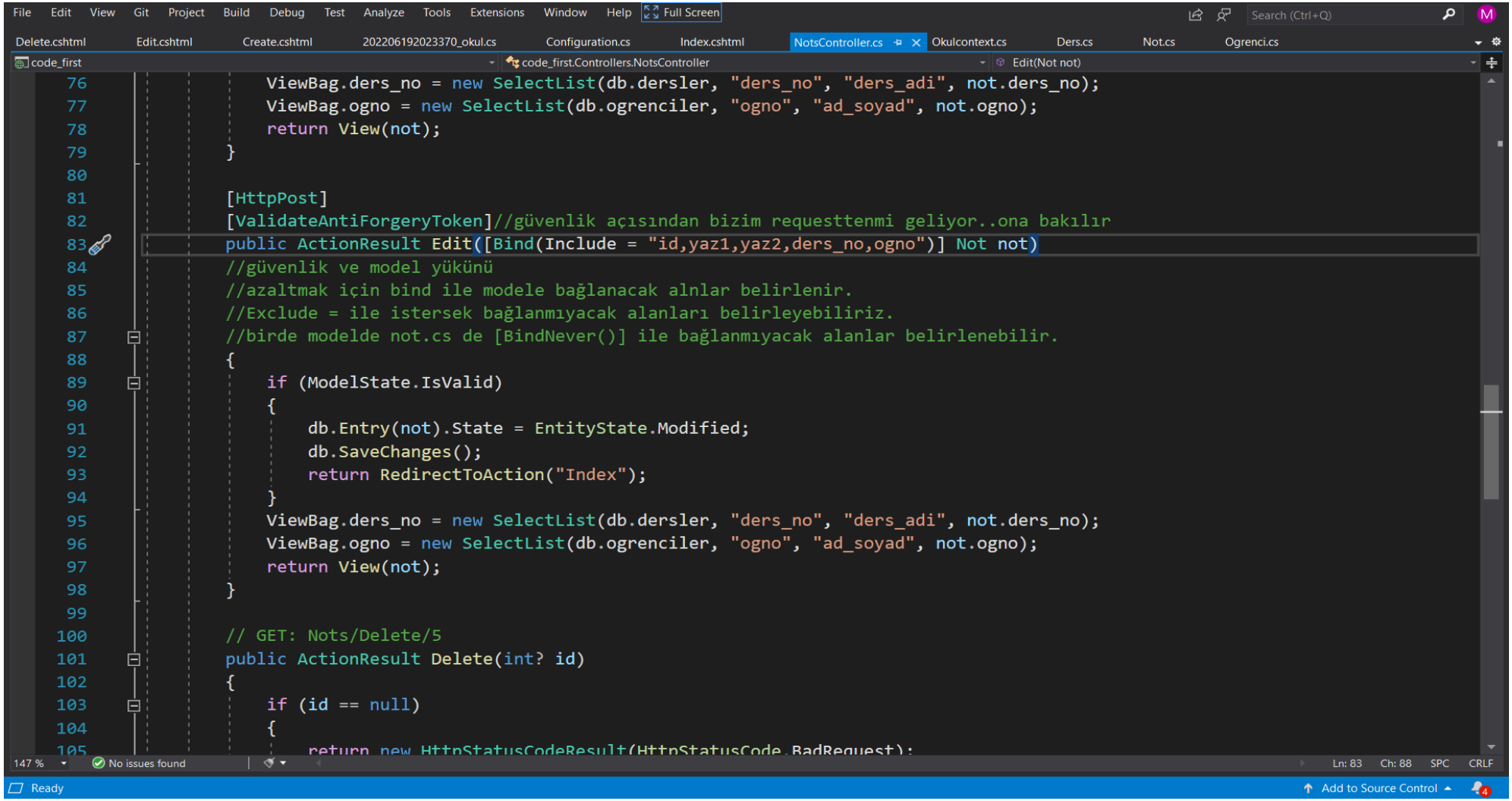
Veritabanı Üzerinde “CRUD” İşlemleri

Birden Fazla Tablodan Veri Çekme ve sihirbazla Crud Action Ve Viewlar Oluşturma (Notlar Tablosu)



Veritabanı Üzerinde “CRUD” İşlemleri

Birden Fazla Tablodan Veri Çekme ve sihirbazla Crud Action Ve Viewlar Oluşturma (Notlar Tablosu)



İstediğimiz Öğrenciye Ait Notları Raporlayalım

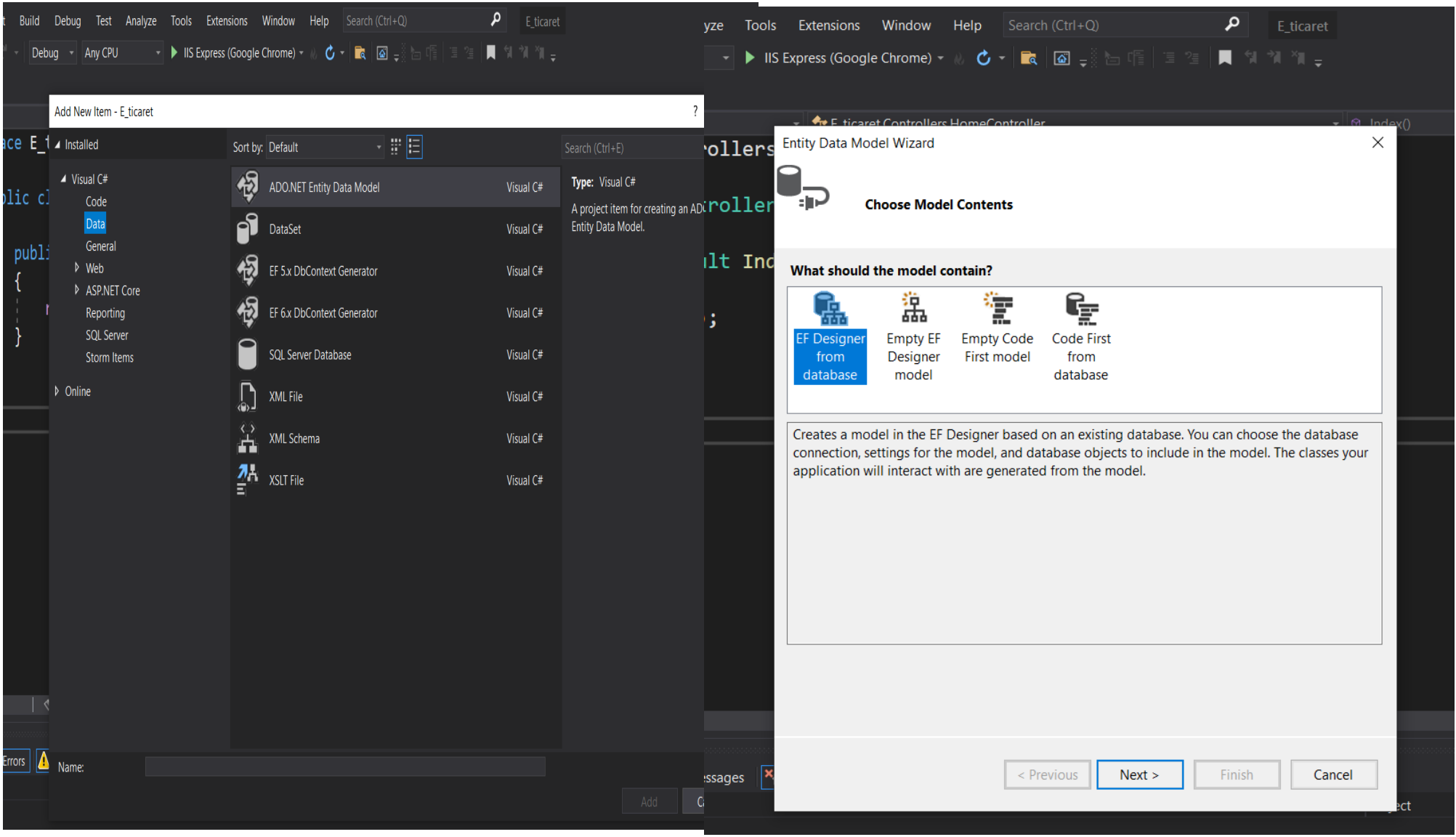
```
137    public ActionResult ogrenciye_gore_not_getir()
138    {
139        var notlar = db.notlar.Include(n => n.Ders).Include(n => n.ogrenci);
140        ViewBag.ogno = new SelectList(db.ogrenciler, "ogno", "ad_soyad");
141        return View(notlar);
142    }
143
144    [HttpPost]
145    public ActionResult ogrenciye_gore_not_getir(int? ogno)
146    {
147        IEnumerable<Not> notlar;
148        if (ogno == null)
149            notlar = db.notlar.Include(n => n.Ders).Include(n => n.ogrenci);
150        else
151            notlar =db.notlar.Include(n => n.Ders).Include(n => n.ogrenci).Where(x => x.ogno == ogno);
152        ViewBag.ogno = new SelectList(db.ogrenciler, "ogno", "ad_soyad");
153        return View(notlar);
154    }
155    }
```

İstediğimiz Öğrenciye Ait Notları Raporlayalım

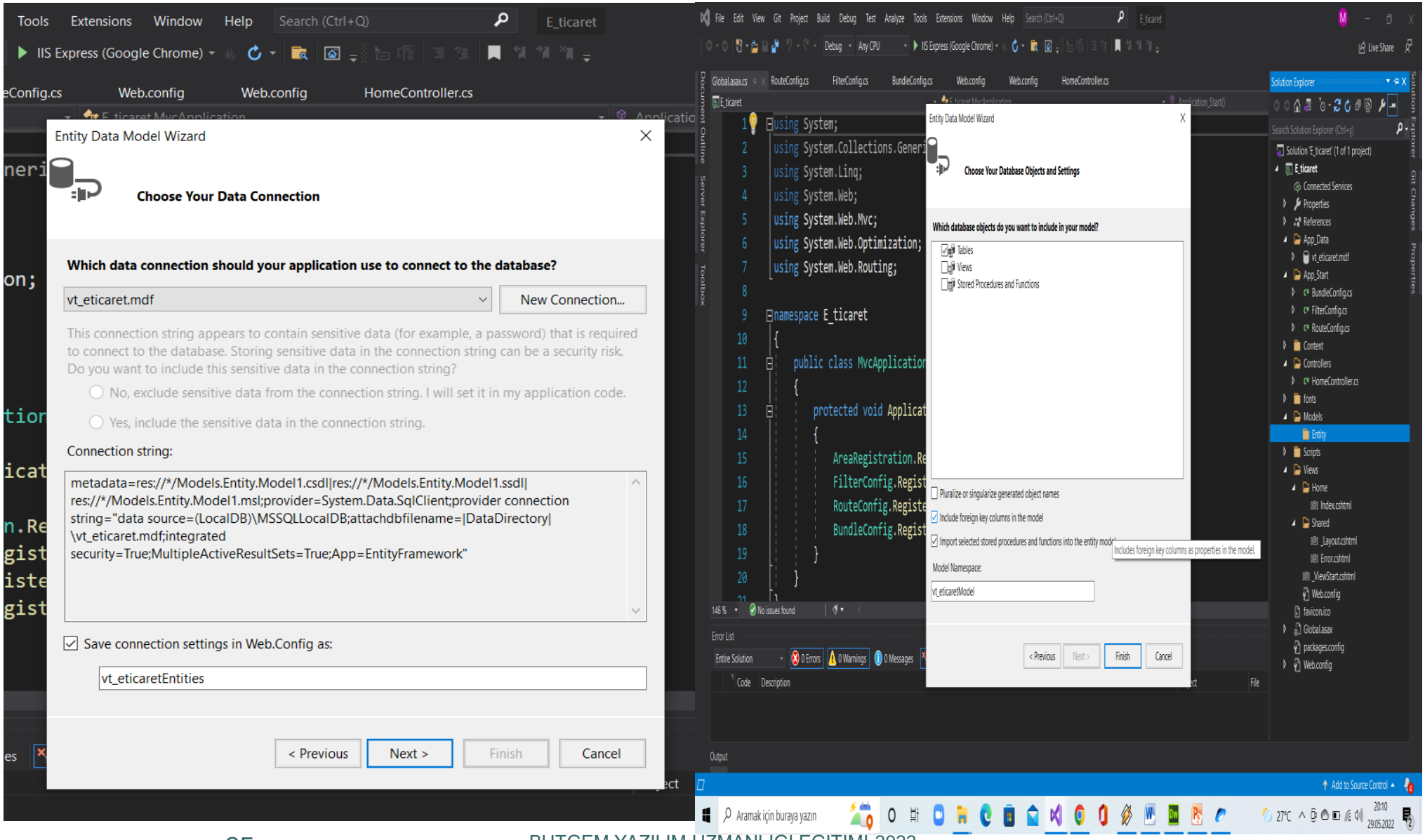
roller.cs | Create.cshtml | Okulcontext.cs | Ogrenci.cs | Ders.cs | Create._kendi_form.cshtml | OgrenciController.cs | Index.cshtml | Ogrenci_notu_getir.cshtml* | Tc_dogrula.cs | Index.cshtml

```
13      </head>
14      <body>
15          @using (Html.BeginForm())
16              {
17                  <p>@Html.DropDownList("ogno",null,"Seçiniz",new {onchange="this.form.submit()" })  </p>
18              }
19          <table class="table">
20              <tr>
21                  <th>
22                      @Html.DisplayNameFor(model => model.Ders.ders_adi)
23                  </th>
24                  <th>…</th>
27                  <th>…</th>
30                  <th>…</th>
33                  <th></th>
34              </tr>
35          @foreach (var item in Model) {
36              <tr>
37                  <td>
38                      @Html.DisplayFor(modelItem => item.Ders.ders_adi)
39                  </td>
40                  <td>
41                      @Html.DisplayFor(modelItem => item.Ogrenci.ad_soyad)
42                  </td>
43                  <td>
44                      @Html.DisplayFor(modelItem => item.yaz1)
45                  </td>
```

Veritabanımız Entity Framework ile Model olarak Ekleyelim



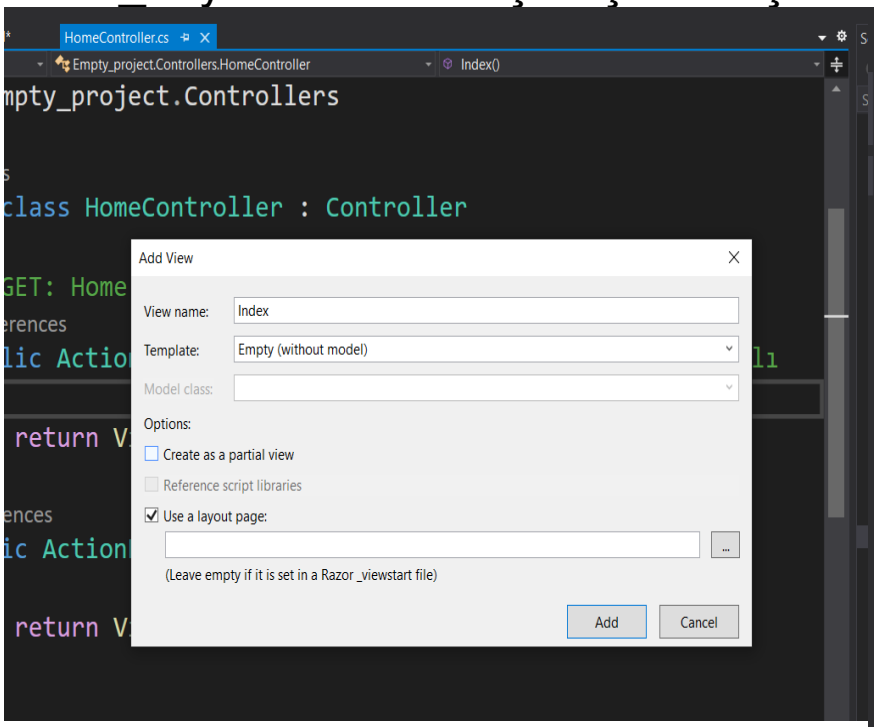
Veritabanımız Entity Framework ile Model olarak Ekleyelim

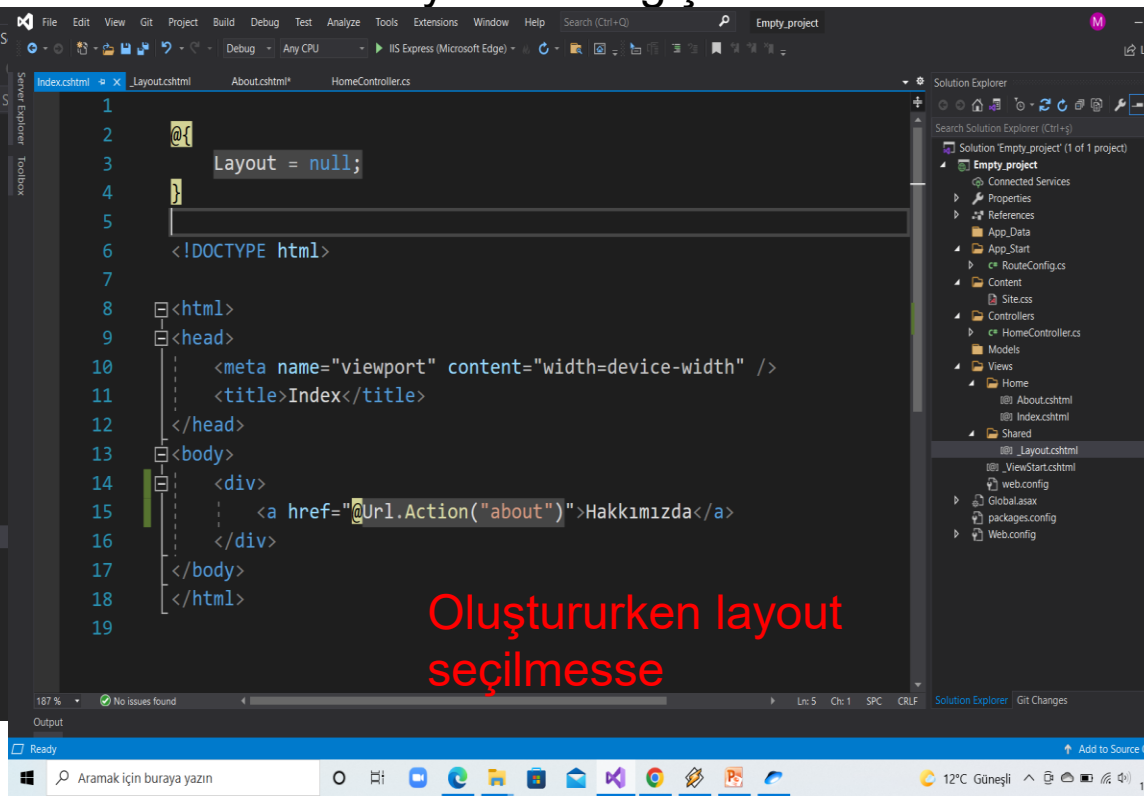


Layout(Master Page/Template)

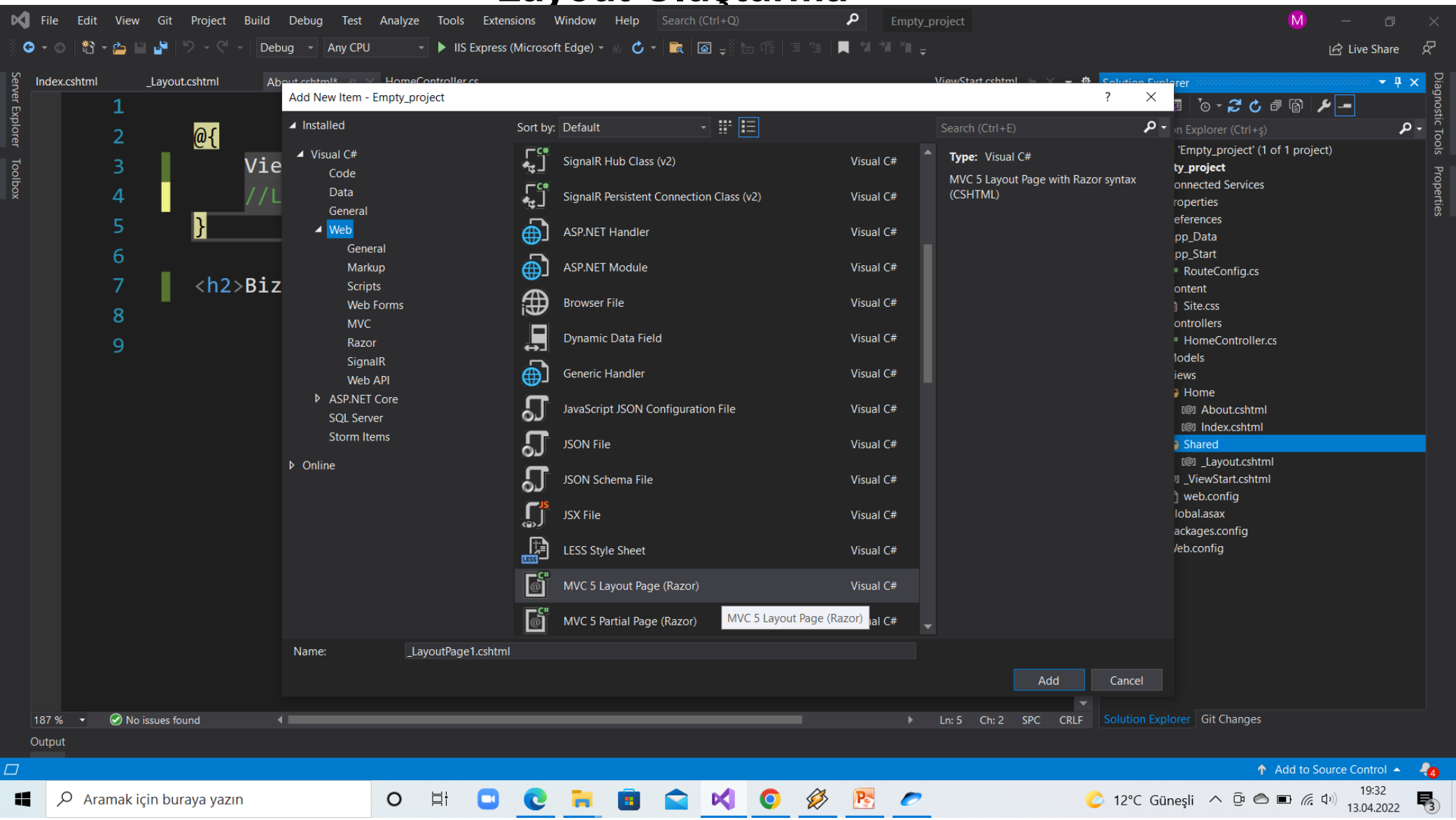
Razor'da Layout'lar, tutarlı bir görünüm(Şablon) sağlamaya yardımcı olmanın yanında bir çok View arasında görüntülenir, kullanılırlar. Web Form'lar ASP.NET sayfaları ile çalıştıysanız,Layout'lar, Master Page'ler ile aynı amaca hizmet ederler.

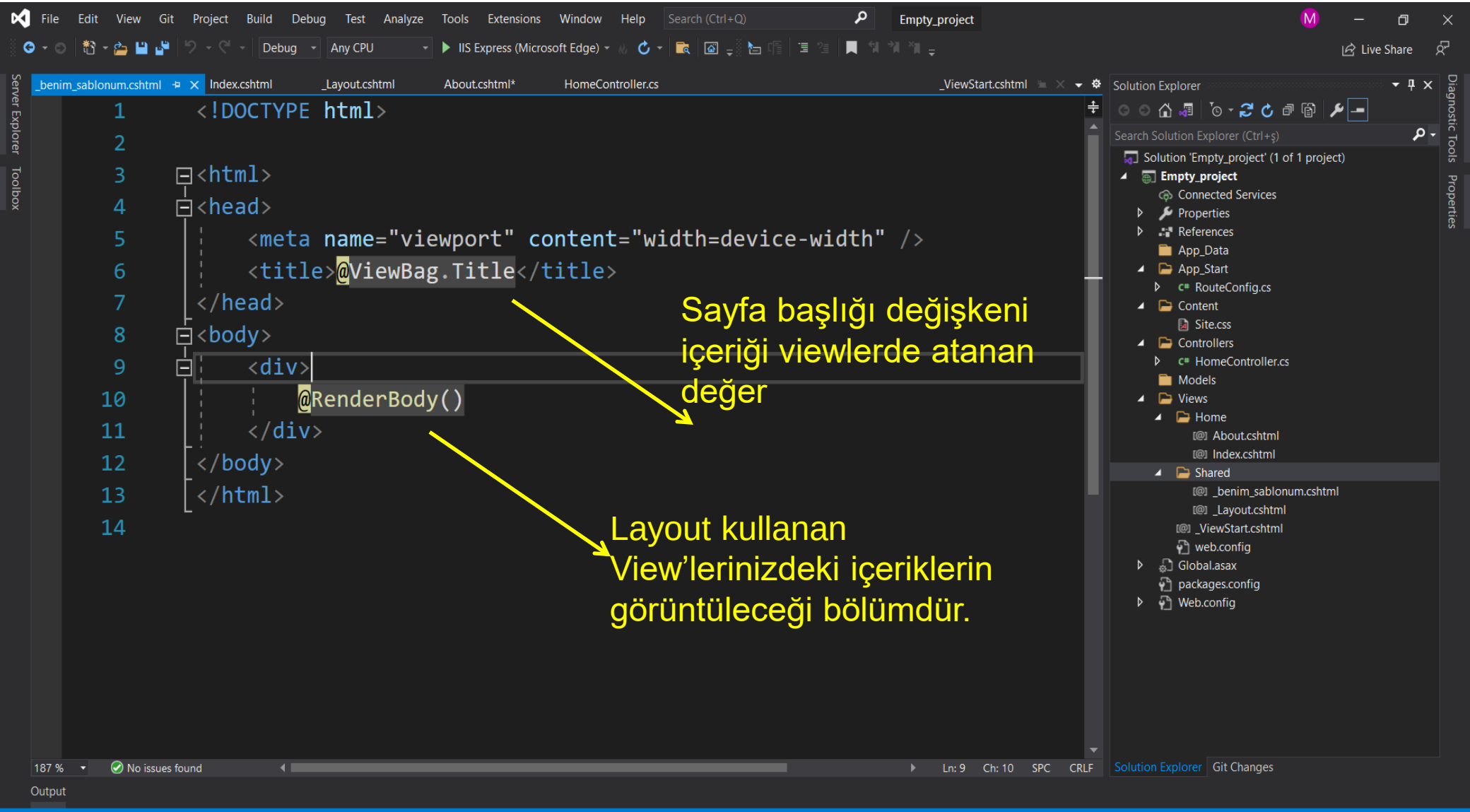
Yeni bir View oluşturduysanız ve layout(chekbox) seçimi yapıldıysa default _Layout.cshtml seçilmiş olur.Açılan kutudan farklı bir layout ile değiştirebilir.





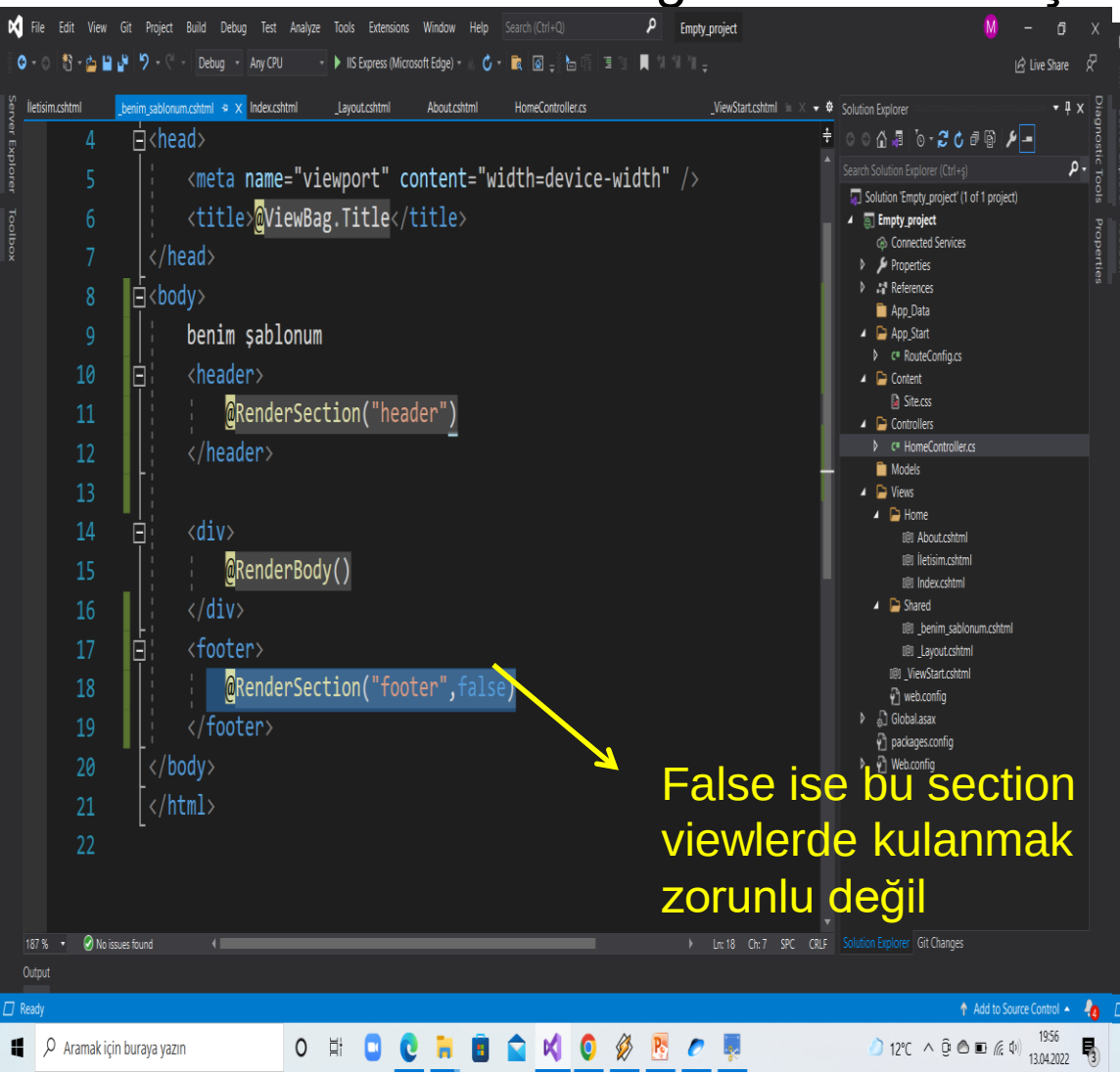
Layout Oluşturma

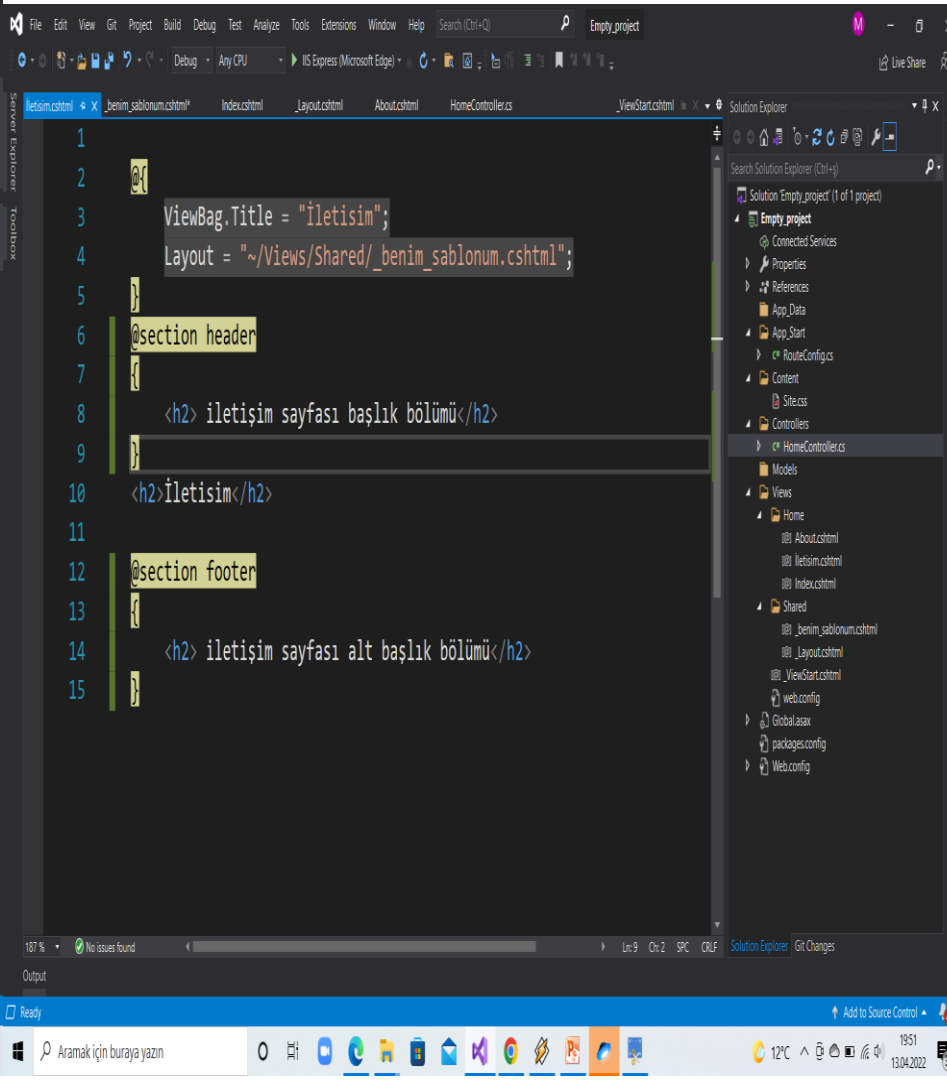



File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
Empty_project
Debug
Any CPU
IIS Express (Microsoft Edge)
Live Share
_benim_sablonum.cshtml
Index.cshtml
_Layout.cshtml
About.cshtml*
HomeController.cs
_ViewStart.cshtml
1 <!DOCTYPE html>
2
3 <html>
4 <head>
5 <meta name="viewport" content="width=device-width" />
6 <title>@ViewBag.Title</title>
7 </head>
8 <body>
9
10 @RenderBody()
11
12 </body>
13 </html>
14
Sayfa başlığı değişkeni içeriği viewlerde atanan değer
Layout kullanan View'lerinizdeki içeriklerin görüntüleceği bölümdür.
Solution Explorer
Search Solution Explorer (Ctrl+ş)
Solution 'Empty_project' (1 of 1 project)
Empty_project
Connected Services
Properties
References
App_Data
App_Start
RouteConfig.cs
Content
Site.css
Controllers
HomeController.cs
Models
Views
Home
About.cshtml
Index.cshtml
Shared
_benim_sablonum.cshtml
_Layout.cshtml
_ViewStart.cshtml
web.config
Global.asax
packages.config
Web.config
Solution Explorer
Git Changes
No issues found
Ln: 9 Ch: 10 SPC CRLF
Output
Ready
Add to Source Control
Aramak için buraya yazın
12°C Güneşli
19:33
13.04.2022

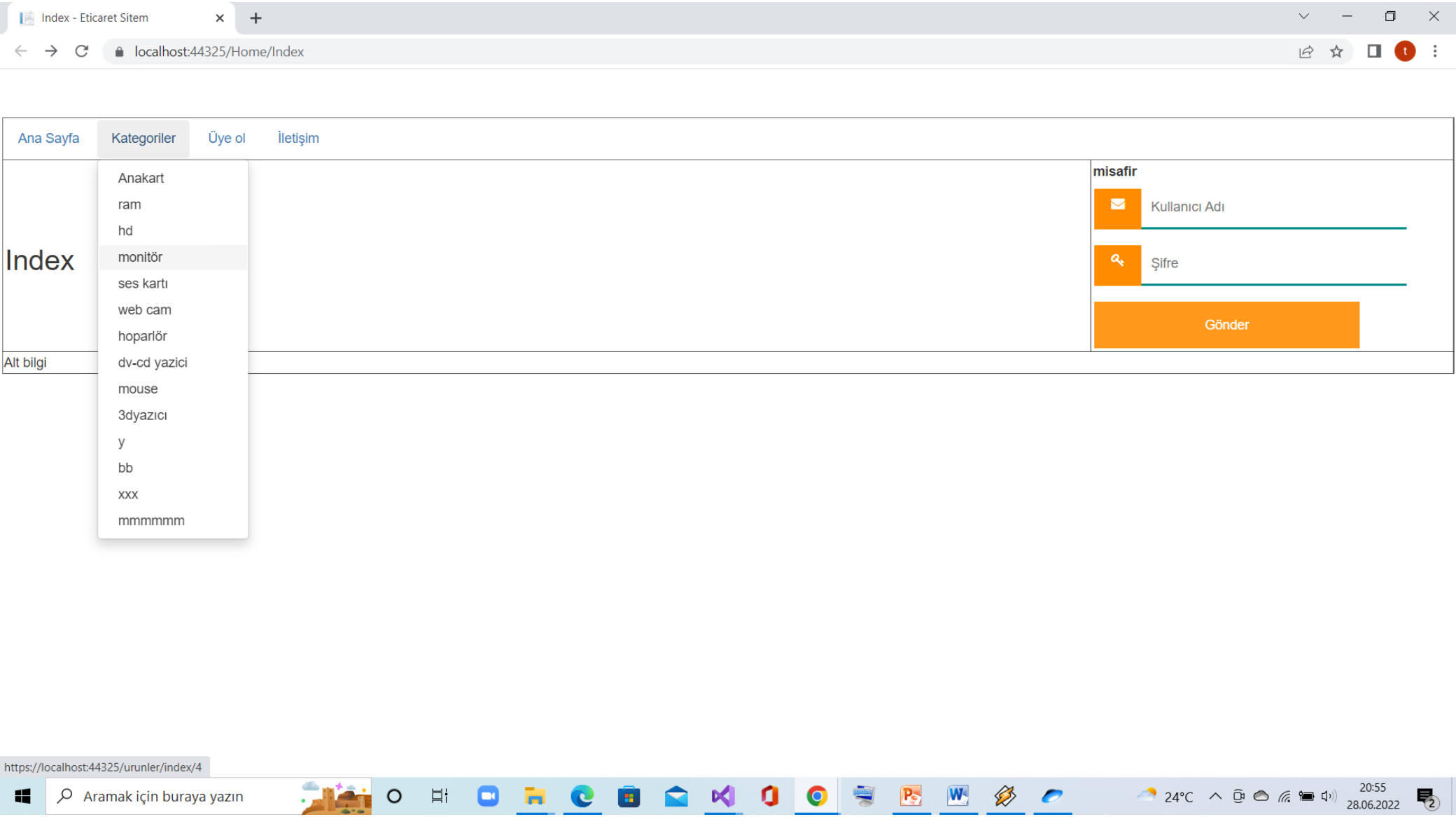
Layout İçerisinde Bir Çok Section(bölüm) Nasıl Kullanılır

Farklı viewlerde farklı görünümler oluşturmamızı sağlar





Layout Hazırlayalım



Layout Hazırlayalım

- Üst menümüzü hazırlayalım Bootstrap nav -bar kullanacağız.

HomeController.cs | Kategori_doldur.cshtml | Index_modelli.cshtml | BundleConfig.cs | urun_kaydet.cshtml | Index.cshtml | UrunlerController.cs | _Layout.cshtml | Sablonstil.css | Index.cshtml

```
16
17      </head>
18      <body>
19          <table class="tablo">
20              <tr>
21                  <td colspan="3" class="ustmenu">
22                      <ul class="nav nav-pills">
23                          <li class="nav-item">
24                              <a class="nav-link active" aria-current="page" href="@Url.Action("index","home")">Ana Sayfa</a>
25                          </li>
26                          <li class="nav-item dropdown">
27                              <a class="nav-link dropdown-toggle" data-bs-toggle="dropdown" href="#" role="button" aria-expanded="false">Kategorile
28                              <ul class="dropdown-menu">
29                                  @{Html.RenderAction("Kategori_doldur", "home");}
30                              </ul>
31                          </li>
32                          <li class="nav-item">
33                              @if (Session["uyem"] == null)//üye giriş yapılmışsa
34                                  {
35                                      <a class="nav-link" href="@Url.Action("uye_kaydet", "uyeler")">Üye ol</a>
36                                  }else
37                                  {
38                                      <a class="nav-link" href="@Url.Action("uye_edit","uyeler",new { uyeid = ((uyeler)Session["uyem"]).uyeid })  ">
39                                  }
40
41                              </li>
42
43                          <li class="nav-item">
44                              <a class="nav-link disabled">İletişim</a>
```

Partialview kullanacağız

132 % | No issues found | Ln: 39 | Ch: 31 | SPC | CR

Layout İçerisinde Partialview kullanma

HomeController.cs* | Kategori_doldur.cshtml | Index.cshtml | BundleConfig.cs | urun_kaydet.cshtml | Index._sablon.cshtml | UrunlerController.cs | _Layout.cshtml* | Sablonstil.css | Index.cshtml

E_ticaret | E_ticaret.Controllers.HomeController | login_giris(uyeler uyem)

```
4   using System.Web;
5   using System.Web.Mvc;
6   using E_ticaret.Models.Entity;
7   namespace E_ticaret.Controllers
8   {
9       public class HomeController : Controller
10      {
11          vt_eticaretEntities db = new vt_eticaretEntities();//yaratırken web confige eklediği isim
12          public ActionResult Index()
13          {
14
15              return View();
16          }
17
18          public ActionResult Kategori_doldur()
19          {
20              return PartialView(db.kategoriler.ToList());
21          }
22
23
```

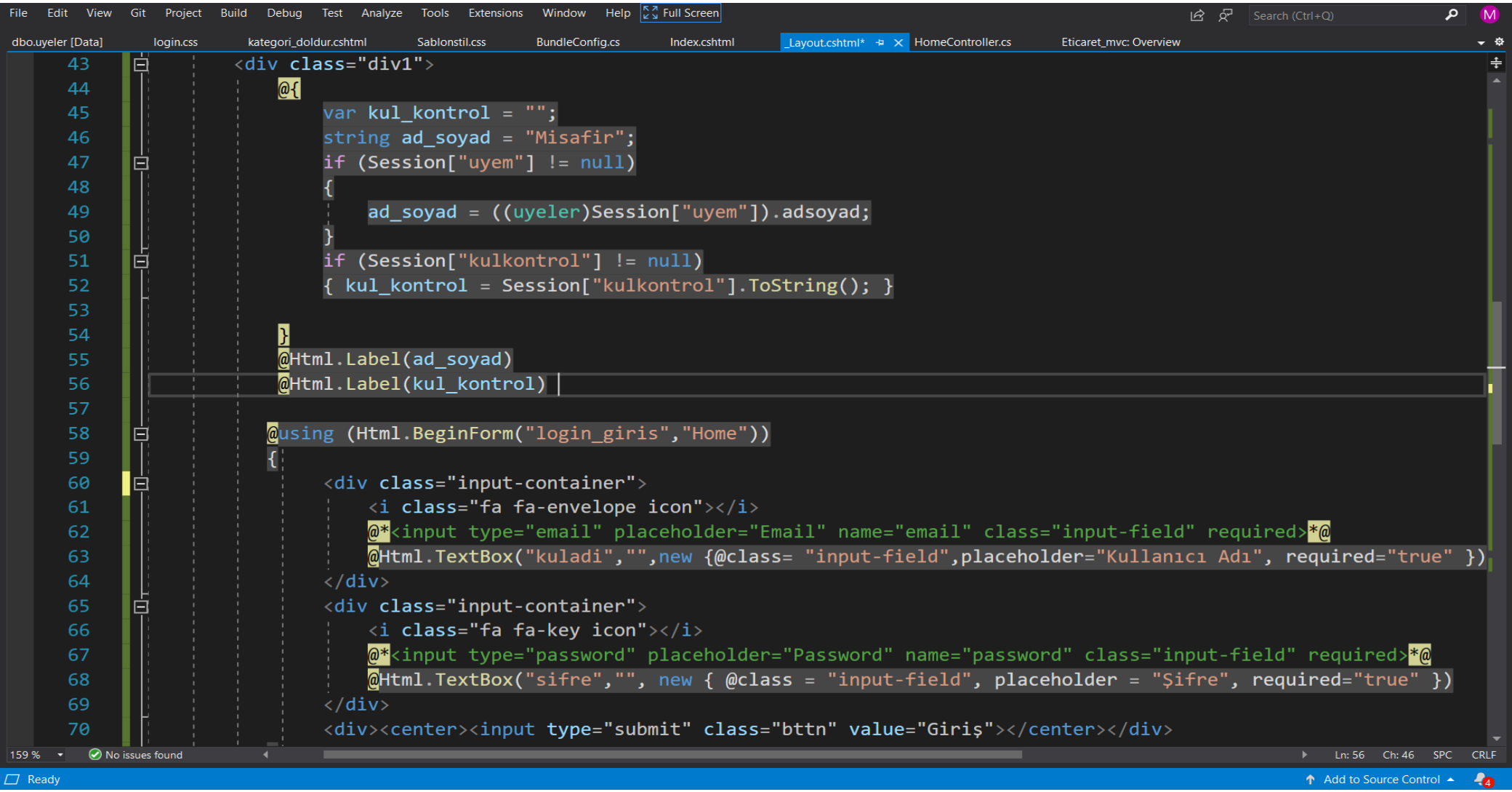
HomeController.cs | Kategori_doldur.cshtml | Index_modelli.cshtml | BundleConfig.cs | urun_kaydet.cshtml | Index.cshtml | UrunlerController.cs | _Layout.cshtml | Sablonstil.css | Index.cshtml

Document Outline | Server Explorer | Toolbox | SQL

```
1   @model IEnumerable<E_ticaret.Models.Entity.kategoriler>
2
3   @foreach (var kategorim in Model)
4   {
5       <li>
6           @Html.ActionLink(kategorim.kateadi, "index", "urunler", new { id = kategorim.kateno },
7           new { @class = "dropdown-item" })
8       </li>
9   }
10
```

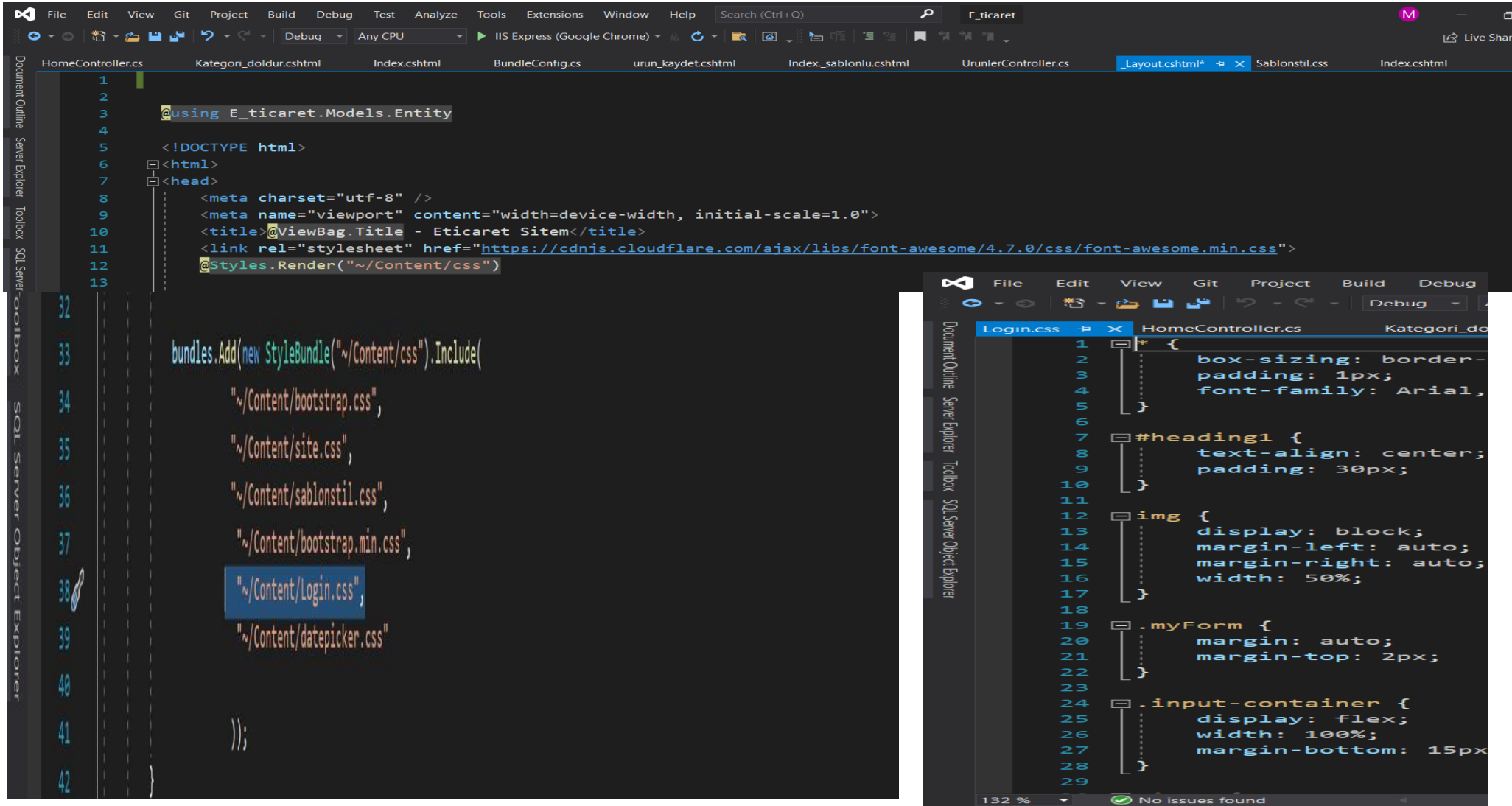
Layout Hazırlayalım

- Sağdaki login template https://codepen.io/hammadhz/pen/PoOjovM indirelim



Layout Hazırlayalım

- Sağdaki login template https://codepen.io/hammadhz/pen/PoOjovM indirelim



Layout Hazırlayalım

- Sağdaki login template https://codepen.io/hammadhz/pen/PoOjovM indirelim

```
19          public ActionResult kategori_doldur()
20          {
21              return PartialView(db.kategoriler.ToList());
22          }
23
24          public ActionResult login_giris(uyeler uyem)
25          {
26              Session["kulkontrol"] = null;
27              var uye = db.uyeler.FirstOrDefault(x => x.kuladi == uyem.kuladi && x.sifre == uyem.sifre);
28
29              if(uye==null)
30              Session["kulkontrol"] = "Yanlış kullanıcı adı veya Şifre";
31              else
32              Session["uyem"] = uye;
33              return RedirectToAction("index");
34          }
35      }
```