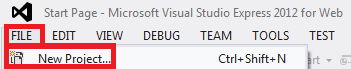
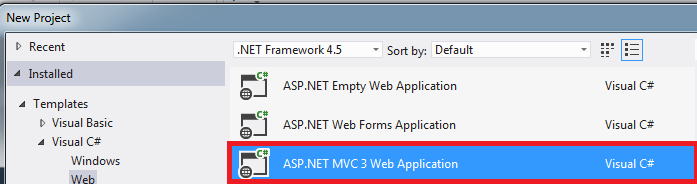
UBI 543 Güvenli Web Dizaynı Final Proje Raporu

1. Projenin Oluşturulması

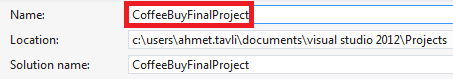
Öncelikle Dosya 🡪 Yeni Proje Oluştur veya Ctrl + Shift + N kombinasyonunu Kullanarak



ASP.NET MVC 3 Web Application projesini seçiyorum.

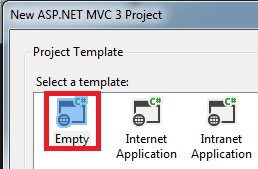


Projemin Adına “CoffeeBuyFinalProject” yazdıktan sonra



Ardından Ok tuşuna basıp Bir sonraki adıma geçiyorum.

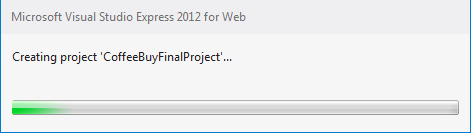
Bir sonraki adımda 3 seçenekten Empty yi seçiyorum.



View Engine Razor ı seçip, HTML 5 kutucuğunu seçiyorum.



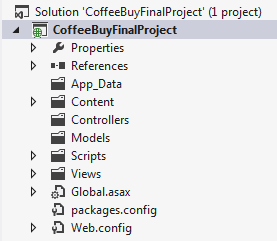
Ok e Bastığımda Proje Oluşuyor. Oluşması biraz uzun sürebilir.



1. Oluşturulan Projeye Genel Bir Bakış

Empty Seçeneği bize tamamen boş bir proje vermez. Sadece Grafiksel arayüzü sade bir proje verir.

Boş proje oluşturulduğunda karşımıza çıkan Durum :



Controllers : Arayüzde girilen verileri algılar, onu nasıl işleyeceğini karar verir, daha sonra kullanıcıya sonuçla ilgili yanıt döner.

Views : Sistemin Grafik Arayüzü şablonlarını tutan sınıfların klasörüdür.

Models : Verilerin tutulduğu ve kullanıldığı sınıfları içeren klasördür.

Content : Resim, CSS gibi statik değerlerin tutulduğu klasördür.

Scripts : JavaScript dosyalarının tutulduğu klasördür.

Buna ek olarak sistemin işleyişi aşağıdaki gibi olacaktır.



Yukarıda kısaca anlatmak istediği:

Controller Kullanıcı’nın sistemde yaptığı işlemi algılayıp, bu istek veya işleme ait modele bilgi gönderiyor.

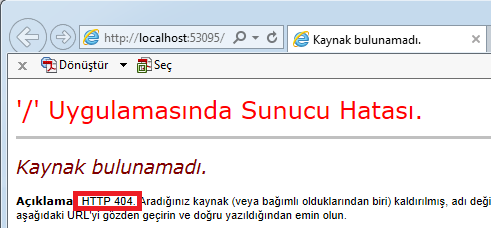
Model ilgili bilgiye göre işlemini yapıp, grafik ara yüzünü güncelliyor.

View ise model den aldığı güncellemeyi gösteriyor.

1. Projenin ilk Kez Çalıştırılması

Projeyi ilk kez çalıştırdığımda hata vermesi gerekir.

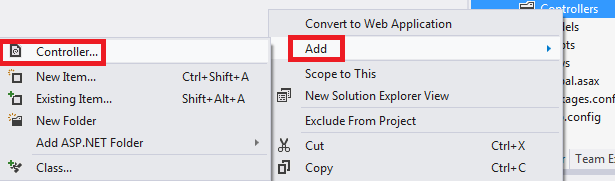
 ikonuna bastığımda



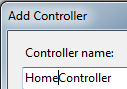
Yukarıdaki hatayı alıyorum.

Öncelikle Bu Hatayı almamın sebebi: Projede kullanıcının isteğini algılayabilecek herhangi bir Controller yok.

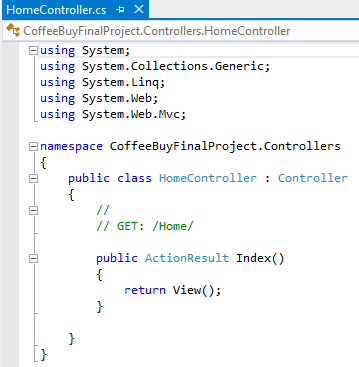
1. HomeController Eklenmesi

Controller Klasörüne Sağ Tıklayıp Yeni bir Controller ekliyorum.

Adına HomeController diyip Ok tuşuna basıyorum.



Karşıma çıkan Sayfa



Şimdi sayfamı çalıştırdığımda bu sınıfta olduğunu anlamak için yukarıdaki kodda Index metodumun tipini String yapıyorum.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace CoffeeBuyFinalProject.Controllers

{

public class HomeController : Controller

{

//

// GET: /Home/

public String Index()

{

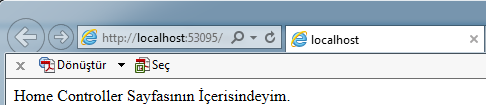
return " Home Controller Sayfasının İçerisindeyim.";

}

}

}

Sayfayı çalıştırdığımda, aşağıdaki mesajı alıyorum.



1. UrunlerController eklenmesi

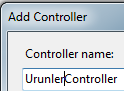
Öncelikle bunu neden UrunlerController’ı ekliyoruz.

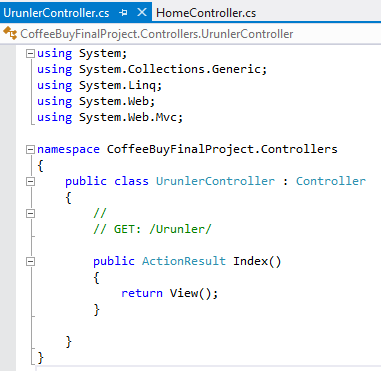
1. Bütün Kahve ürünlerimizi listelemek için
2. Bütün Ürünleri kategorize edebilmek için
3. Ürünler Hakkında detaylı bilgi verebilmek için

Öncelikle Yeni bir Kontroller ekliyorum.



Çıkan sayfada UrunlerController yazıyorum.



Daha sonra Karşıma çıkan Sayfa;

Aşağıdaki değişiklikleri yapıyorum.

1. Index metodunun tipini string yaptım.
2. Arastir ve Detay metodlarını ekliyorum.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace CoffeeBuyFinalProject.Controllers

{

public class UrunlerController : Controller

{

//

// GET: /Urunler/

public string Index()

{

return "Urunler Ana Sayfa";

}

//

// GET: /Urunler/Arastir

public string Arastir()

{

return "Urunlerin Ana Sayfası Araştır Kısmı";

}

//

//GET: /Urunler/Detay

public string Detay()

{

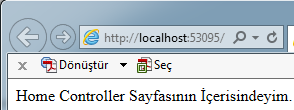
return "Urunlerin Detay Sayfası";

}

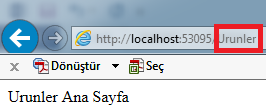
}

}

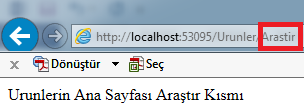
Daha sonra çalıştırıyorum.



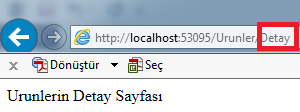
Url kısmının sonuna Urunler yazıyorum.



Urunlerin sağına /Arastir yazıyorum.



/Arastir yerine Detay yazıyorum.



Böylelikle ürünlerle alakalı yazdığım metodların hepsine ulaşabiliyorum.

Burda anlatmak istediğim diğer bir şey ise, Araştır metoduna url den parametre göndermek.

Bunun için Arastir metodunun altına

//

//GET: /Urunler/Arastir?deger=Espresso

public string DegerArastir(string deger)

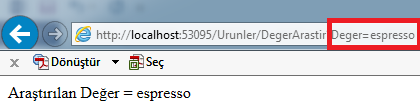
{

string message = HttpUtility.HtmlEncode("Araştırılan Değer = " + deger);

return message;

}

Sitemi çalıştırdığımda



Metodumda kullandığım HttpUtility.HtmlEncode url kısmında girilebilen zararlı içerikleri önlemek amacıyla kullandım.

Şimdi Aynen Araştır metoduma yaptığım bir benzer değişikliği Detay metodumada uyguluyorum.

//

//GET: /Urunler/DegerDetay/3

public string DegerDetay(int deger)

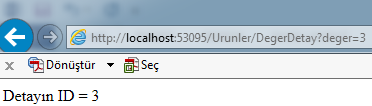
{

string message = "Detayın ID = " + deger;

return message;

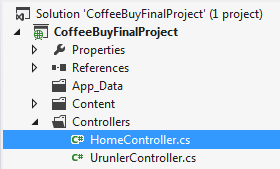
}

Sitemi çalıştırdığımda



1. HomeController Grafiksel Ara Yüzünü oluşturmak.

Bunu yapmak için öncelikle HomeController’ı seçiyorum.



Daha önce oluşturulmuş olan Index metodumun tipini string tipinden ActionResult tipi yapıyorum.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace CoffeeBuyFinalProject.Controllers

{

public class HomeController : Controller

{

//

// GET: /Home/

public ActionResult Index()

{

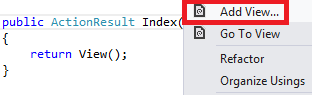
return View();

}

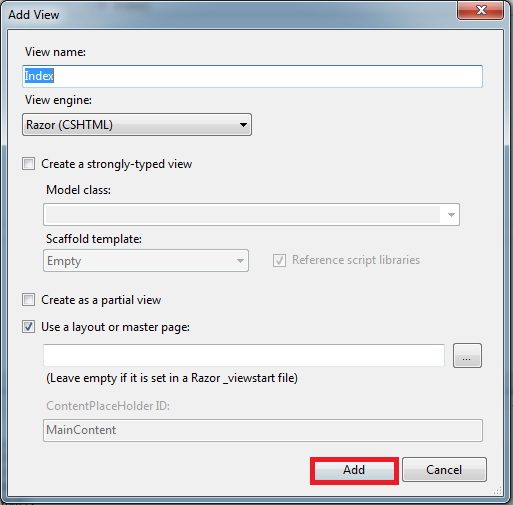
}

}

Daha sonra Index() yazısının üstüne gelip sağ tıklıyorum.



Add View a tıklıyorum.

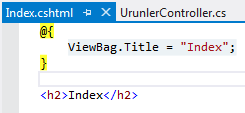


Add tuşuna tıklıyorum.

Ardından projemde Home Klasörümün altında Index.cshtml diye bir dosya beliriyor.



Index.cshtml aşağıda gözüktüğü gibi bir sınıf oluşturuyor.



Yukardaki Kodu aşağıdaki ile değiştiriyorum.

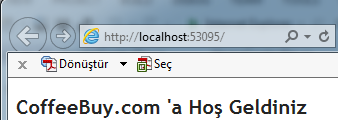
@{

ViewBag.Title = "Index";

}

<h2>CoffeeBuy.com 'a Hoş Geldiniz</h2>

Uygulamayı çalıştırdığımda

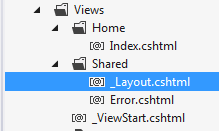


Mesajını görüyorum.

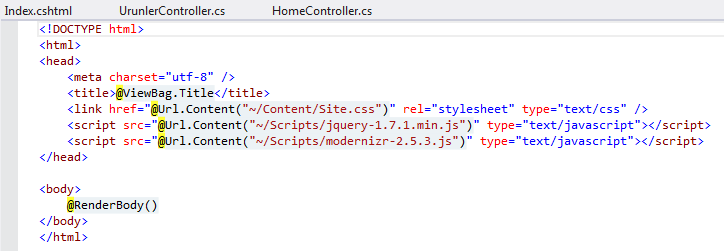
1. \_Layout.cshtml Değiştirmek.

\_Layout.cshtml Razor motoru tarafından oluşturulmuş, Sayfada genel bir düzenleme yapılmak istendiğinde değişiklik yapılan sınıftır.

View klasörümün altında



\_Layout.cshtml’e tıkladığımda



Yukarıdaki kodları görüyorum.

Sitede yapmak istediğim tüm değişiklikleri:



Öncelikle bir <div> tagi yerleştiriyorum.

<body>

<div>

</div>

@RenderBody()

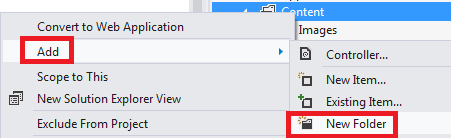
</body>

Daha sonra div in id si olarak kullancağım baslik nesnesini CSS dosyamda oluşturucam.

Önce Projemden Site.css i seçiyorum.



Daha sonra logo mu eklemek üzere Content klasörüme sağ tıklayıp New Folder seçiyorum.



Daha sonra klasörümün adını Images olarak değiştiriyorum. Bu klasör projemdeki bütün resim dosyalarını saklıcak olan klasör.



Daha sonra baslik id sini ekliyorum.

#baslik {

float: left;

width: 100%;

border-bottom: 1px dotted #5D5A53;

background: url(/content/Images/logo.jpg) no-repeat;

padding: 240px 500px 0px 10px;

margin-bottom: 10px;

}

Daha sonra bunu \_Layout.cshtml teki div içine ekliyrum.

<body>

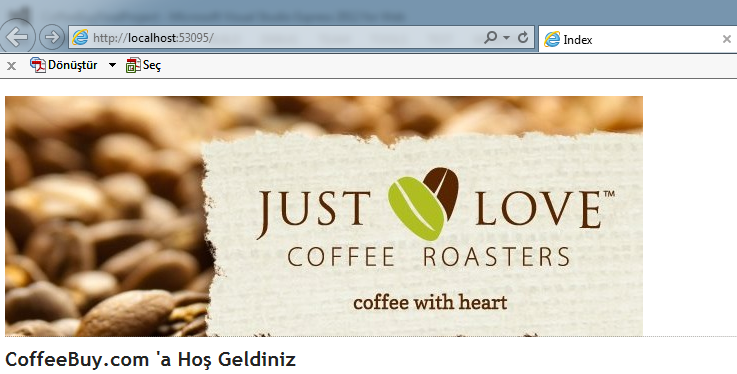
<div id="baslik">

</div>

@RenderBody()

</body>

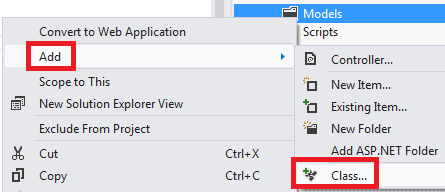
Sitemi çalıştırdığımda



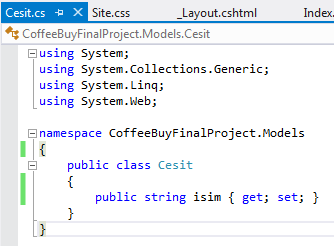
1. Modelleri Oluşturmak

Modeller 2. Kısımda da bahsettiğim gibi verilerin saklanması ve kullanmasında işe yarıyordu.

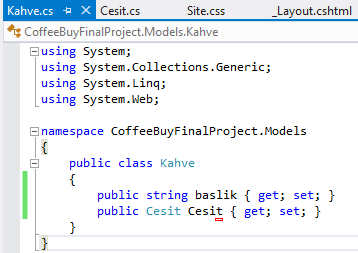
İlk olarak Model klasörüme sağ tıklayıp yeni bir sınıf oluşturuyorum.



Yeni oluşturduğum sınıfımın adına Cesit.cs veriyorum. Nedeni ise Kahvelerimin Çesidi olcağından dolayı, örnek : Espresso, americano, filtre vs…



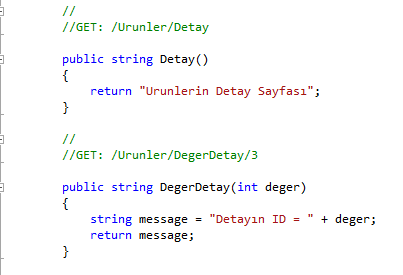
Şimdi Yeni bir sınıf daha oluşturuyorum. İsmine ise Kahve.cs diyorum.



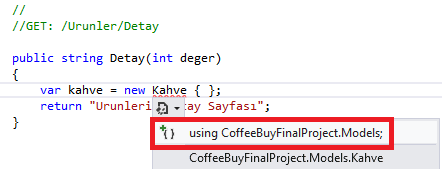
1. UrunlerController ı Düzeltmek

UrunlerController da Detay ve DetayDeger adlı string tipinde 2 tane metodum vardı.

1. DetayDeger i silicem
2. Detay metodunun tipini ActionResult yapıp int parametre alcak hale getiricem. Böylelikle kahvelerle ilgili detaylar ekranda görülebilcek.



Öncelikle Modelde oluşturduğum Kahve sınıfını kullanabilmek için



Models ( kırmızıyla çizili kısmı eklemem ) gerekiyor.

//

//GET: /Urunler/Detay

public ActionResult Detay(int deger)

{

var kahve = new Kahve { baslik = "Kahve " + deger };

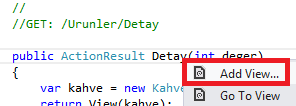
return View(kahve);

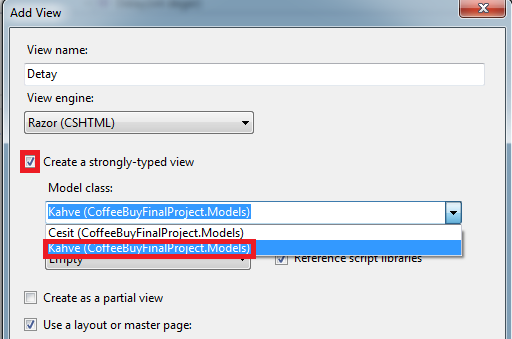
}

Burdaki yaptığım değişikliklerin etkili olması için projemi derliyorum.



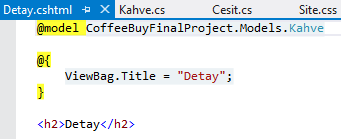
Şimdi burda oluşturduğum verinin Ekranda gözükmesi için Detay ın üzerine sağ tıklayıp, Add View diyorum.





Yukarıdaki kırmızı işaretlerle olan değişiklikleri yaptıktan sonra Add tuşuna basıyorum.

Aşağıdaki gibi bir sayfa karşıma çıkıyor.



Burda <h2> tagleri ile işaretlenmiş alanı düzeltiyorum.

@model CoffeeBuyFinalProject.Models.Kahve

@{

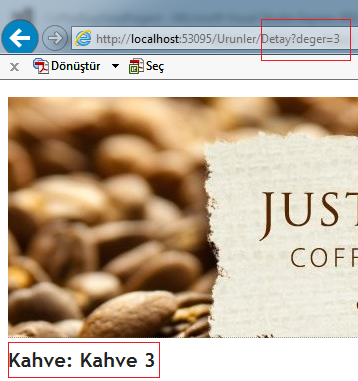
ViewBag.Title = "Detay";

}

<h2>Kahve: @Model.baslik</h2>

Öncelikle ekranda url den girdiğim veriyi ekranda görüp göremediğime bakmalıyım.

Sitemi çalıştırıyorum.



Url den 3 değeri girdiğimde, ekranda Kahve 3 değerini görüyorum.

Bu değeri nasıl gördüm? Veya neden Kahve : Kahve 3 çıktı.

Öncelikle Detay View Kısmında Kahve : @Model.baslik demiştim.

Kahve : kısmı ordan geliyor.

Daha sonra Kahve : @Model.baslik

@Model.baslik kısmını Urunler Controller sınıfında Kahve olarak tanımlamıştım.

var kahve = new Kahve { baslik = "Kahve " + deger };

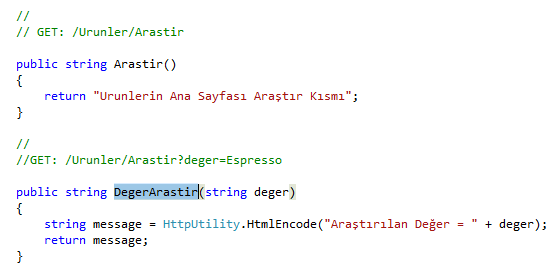
Daha sonra değer verisini url den 3 olarak verdim.

Sonuç :

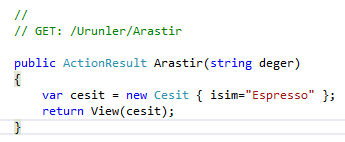
Kahve : Kahve 3

UrunlerController da Arastir ve DegerArastir adlı string tipinde 2 tane metodum vardı.

1. DegerArastir i silicem
2. Detay metodunun tipini ActionResult yapıp int parametre alcak hale getiricem. Böylelikle kahvelerle ilgili detaylar ekranda görülebilcek.

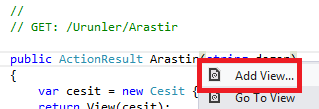


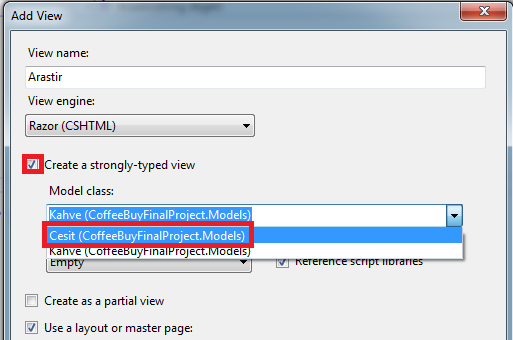
Öncelikle Modelde oluşturduğum Kahve sınıfını kullanabilmek için



Burdaki yaptığım değişikliklerin etkili olması için projemi derliyorum. 

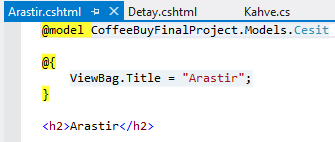
Şimdi burda oluşturduğum verinin Ekranda gözükmesi için Arastir ın üzerine sağ tıklayıp, Add View diyorum.





Yukarıdaki kırmızı işaretlerle olan değişiklikleri yaptıktan sonra Add tuşuna basıyorum.

Aşağıdaki gibi bir sayfa karşıma çıkıyor.



Burda <h2> tagleri ile işaretlenmiş alanı düzeltiyorum.

@model CoffeeBuyFinalProject.Models.Cesit

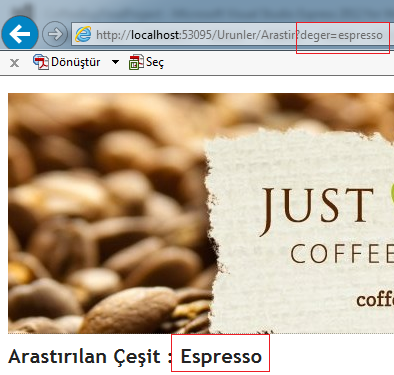
@{

ViewBag.Title = "Arastir";

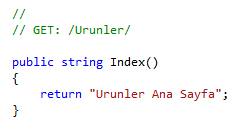
}

<h2>Arastırılan Çeşit : @Model.isim</h2>

Öncelikle ekranda url den girdiğim veriyi ekranda görüp göremediğime bakmalıyım. Sitemi çalıştırıyorum.



Son olarak Index i düzenliyorum.



Burda indexin string tipini ActionResult yapıcam.

//

// GET: /Urunler/

public ActionResult Index()

{

var cesitler = new List<Cesit>

{

new Cesit { isim = "Espresso"},

new Cesit { isim = "Americano"},

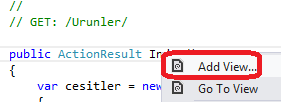
new Cesit { isim = "Filtre"}

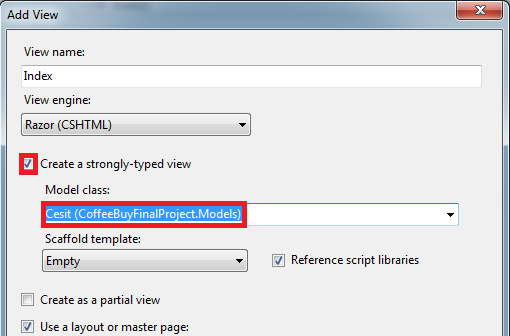
};

return View(cesitler);

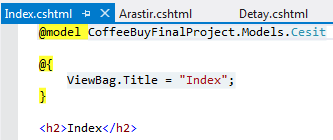
}

Index() sağ tıklayıp Add View diyorum.





Yukarıdaki kırmızı işaretlerle olan değişiklikleri yaptıktan sonra Add tuşuna basıyorum.



Şimdi elimde birden fazla kahve çeşidi var.

Bunları gösterebilmek için.

@model IEnumerable<CoffeeBuyFinalProject.Models.Cesit>

@{

ViewBag.Title = "Ürünler";

}

<h2>Ürünlerde Araştır</h2>

<p>

@Model.Count() tane Çeşidimizden Seçebilirsiniz;

Çeşitler :

</p>

<ul>

@foreach (var cesit in Model)

{

<a href="/Store/Browse?genre=@cesit.isim">@cesit.isim</a>

}

</ul>

Sitemi çalıştırdığımda beklediğim, Her bir Çeşidi alt alta sıralaması ve herbirine link vermesi.



Şu ana kadar yaptığım foreach döngüsünde her bir çeşit için link vermek oldu.

Fakat bu kodu geliştirebilirim.

@model IEnumerable<CoffeeBuyFinalProject.Models.Cesit>

@{

ViewBag.Title = "Ürünler";

}

<h2>Ürünlerde Araştır</h2>

<p>

@Model.Count() tane Çeşidimizden Seçebilirsiniz;

Çeşitler :

</p>

<ul>

@foreach (var cesit in Model)

{

<li>

@Html.ActionLink(cesit.isim, "Araştır", new { cesit = cesit.isim })

</li>

}

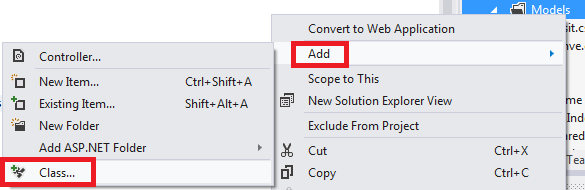
</ul>

Burda yaptığım, Controller Uygulama Yapabileceği bir Link oluşturmaktı, o yüzden Html.ActionLink kullandım.

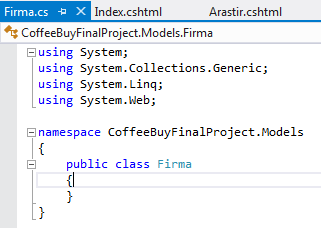
Html.ActionLink 3 parametre almış

1. Linkin İsmi : Ben Her bir Linkin İsmi ni Çeşidin İsmine eşitledim.
2. Uygulamanın İsmi : Burda Araştır metodunun içinde olduğundan dolayı, uygulamanın ismine Araştır diye statik bir değer verdim.
3. Obje Değeri : Cesit İsimlerinin herbirisini yeni bir obje oluşturark bu parametreye de değerini vermiş oldum.
4. Firma.cs Sınıfını Modele Eklemek

Models klasörüme sağ tıklayıp Add Class diyerek Firma.cs sınıfımı ekliyorum.

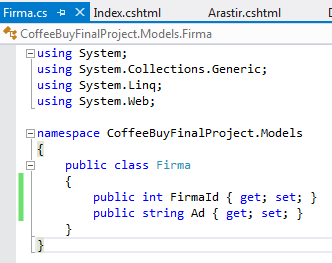


Sınfımın ilk Açılışnda

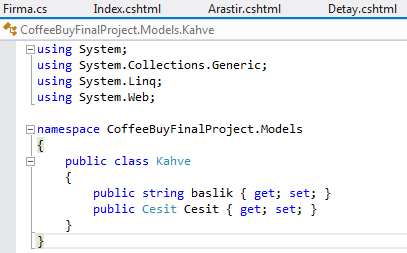


Firma.cs sınıfıma FirmaId ve Ad özelliklerini ekliyorum.

FirmaId primary key , Ad ise Firma Adını belirleyen özelliklerim.

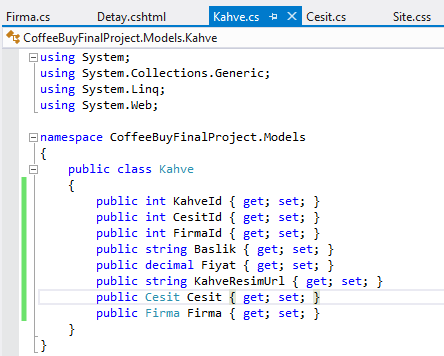


Şimdi Kahve.cs Sınıfımı geliştirmek istiyorum.



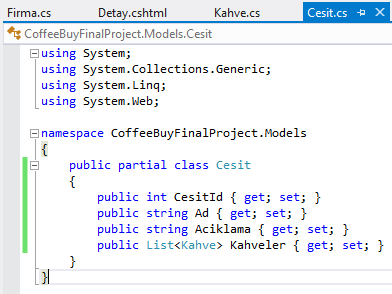
Buna Öncelikle

* KahveId Primary Key int
* CesitId Foreign Key int
* FirmaId Foreign Key int
* Fiyat decimal tipinde
* KahveResimUrl string tipinde
* Firma Firma tipinde



Şimdi sırada Cesit.cs dosyamı güncelliyorum.

* Class 🡪 partial class olarak değiştiriyorum.
* CesitId Primary Key int
* Aciklama string tipinde
* List<Kahve> Kahveler



Sınıfları istediğim gibi düzenledikten sonra bağlantıyı ayarlamam gerekiyor.

1. Web.config düzenlenmesi

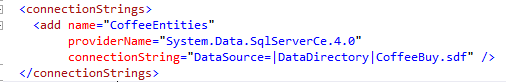


Yukarıda kırmızı işaretli olan yerleri düzelticem.

Name =CoffeeEntities

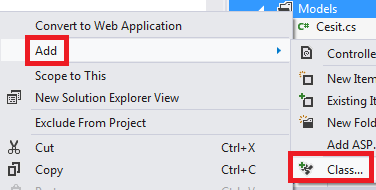
connectionString = DataSource=|DataDirectory|CoffeeBuy.sdf

providerName = System.Data.SqlServerCe.4.0

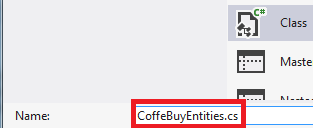


1. Bağlam sınıfının Eklenmesi

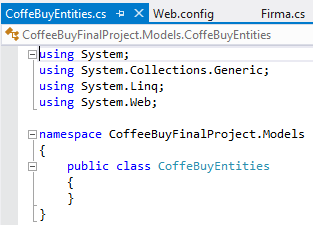
Öncelikle Models klasörüme sağ tıklayıp yeni bir sınıf ekliyorum.



Çıkan Ekranda Sınfımın ismine CoffeeBuyEntities.cs ismini veriyorum.

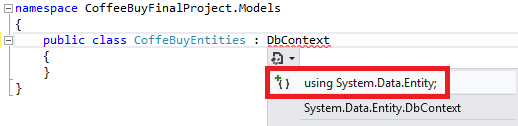


Karşıma çıkan ilk başta

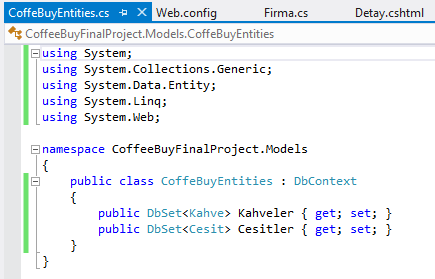


DbContext i ekliyorum.

DbContext i ekleme için System.Data.Entity yi kullanmam gerekiyor.



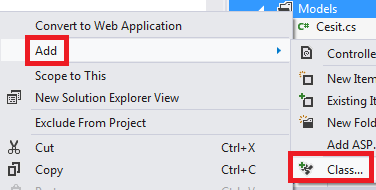
DbContext bana DbSet<> metodlarını kullanmamı sağlayacak.

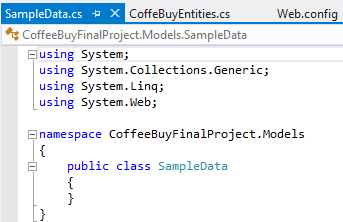


1. Örnek Veri Eklemek

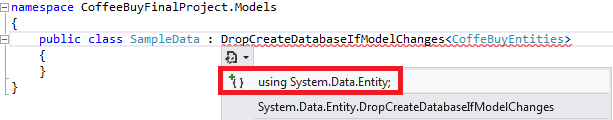
Burdaki amacım daha önceden hazır belirlenmiş veri yerine veritabanından veri alan bir sistem geliştirmek.

Öncelikle Models klasörüme sağ tıklayıp yeni bir sınıf ekliyorum.





DropCreateDatabaseIfModelChanges : Eğer Model değişirse veritabanını yeniden kur.



Oluşturduğum SampleData.cs sınıfına

Çeşitler, firmalar ve kahveler var tipinde değerler oluşturuyorum.

Daha Sonra sırasıyla firmalara veri ekliyorum.

Bu veri ekleme işini List yapısını kullanarak oluşturuyorum.

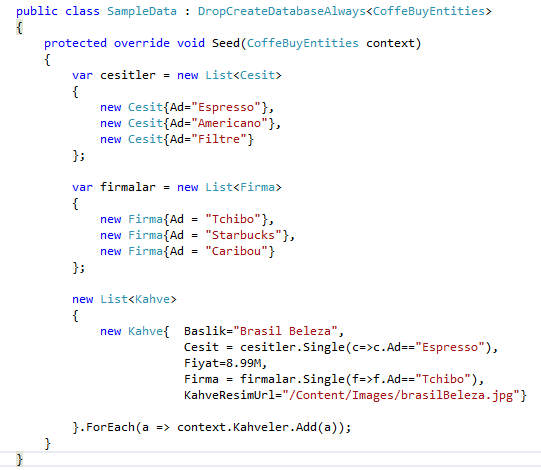
Örnek :

İlk önce verimi Tanımlıyorum

var cesitler= new List<Cesit> { };

Daha sonra veriyle doldurmak için

var cesitler= new List<Cesit> {new Cesit { Ad = “Espresso” } };

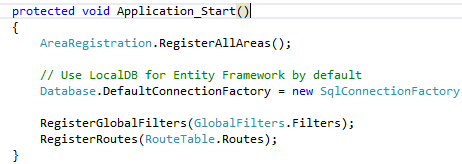


Yukarda sadece Espresso çeşidini eklediğimden dolayı, ekranda sadece Espresso yazısının gözükmesini planlıyorum.

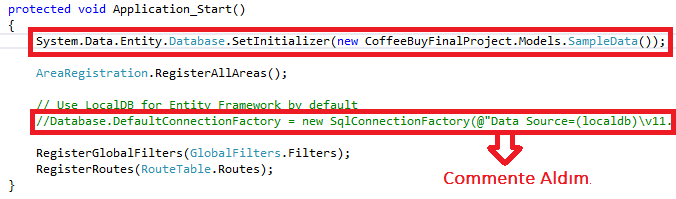
Daha sonra bu yaptığım değişikliklerin etkili olabilmesi için Global.asax dosyasını açıyorum.



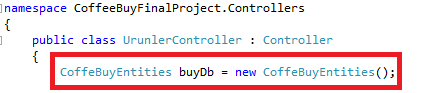
Global.asax sınıfnda Application\_Start metodunun ilk satırını



DbInitializer ile SampleData ilk olarak çalışsın komutunu veriyorum.

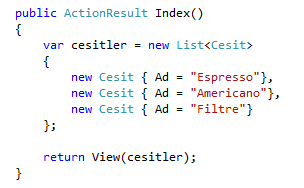


Son Olarak UrunlerController.cs te değişiklik yapıcam. Bu yaptığım değşikliklerin ekranda gözükmesi için.

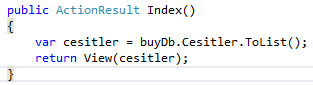


Yukarıda Kırmızı ile gösterilmiş yeri ekledikten sonra;

Eski Index Metodum, daha önceden elle veri girilmiş metodumu,



Yeniliyorum, veritabanından veri alçak şekilde.

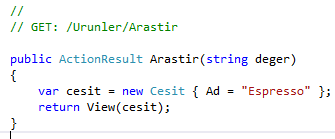


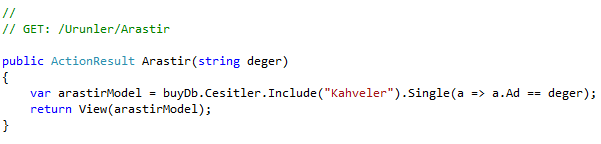
Sitemi çalıştırdığımda :



Sırada UrunlerController.cs sınıfında yer alan Arastir metodunu düzenlemek

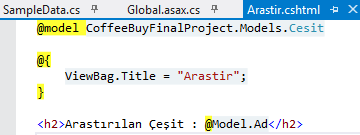
Eski Arastir metodum, url den aldığı değeri ekrana basan metodum.



Yeni Arastir metodum, burda yapmak istediğim araştırılcak olan değeri Kahveler ve Cesitler DbSet Entitilerinde aratmak.

Sırada Arastir metodunun View kısmı olan Arastir.cshtml kısmını düzenlemek var.

Eski Arastir.cshtml metodum



Yeni Arastir metodum. Kırmızı kısım yeni eklediğim

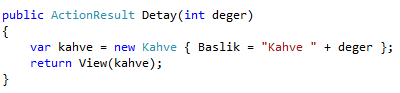


Sitemi çalıştırdığımda

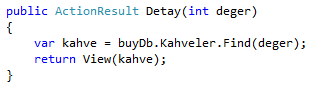


Son Olarak Detay metodumu düzenliyorum.

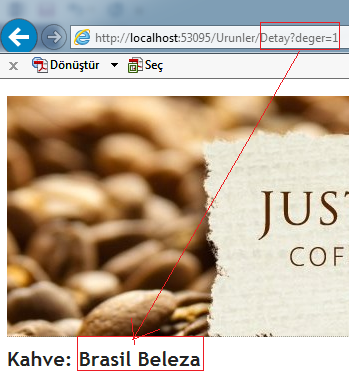
Eski Detay Metodum



Yeni Detay Metodum



Sitemi çalıştırdığımda

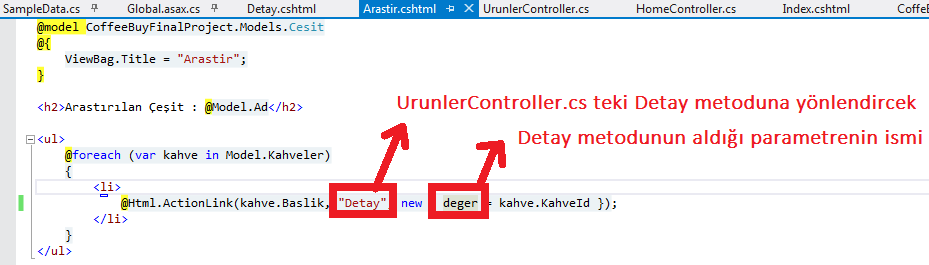


Şimdi burdaki bu Detay bilgisine url den girince geliyordu.

Bunu düzenlemek için Arastir.cshtml kısmında

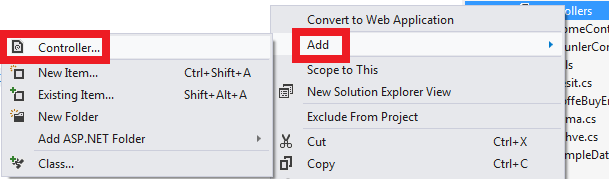
Foreach döngüsünde @kahve.Baslik yerine Html.Link olarak değiştircem.

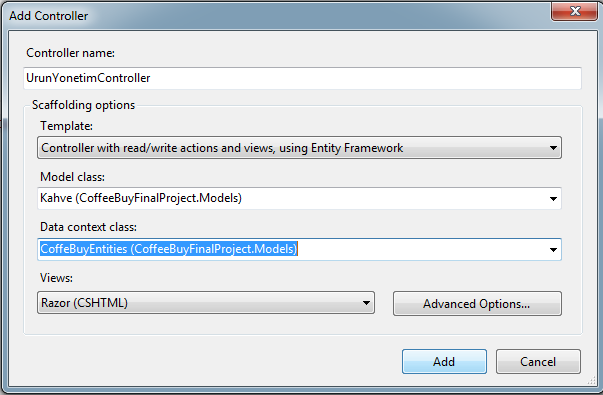
Yeni Arastir.cshtml sınıfım



1. UrunYonetimController ı Oluşturmak

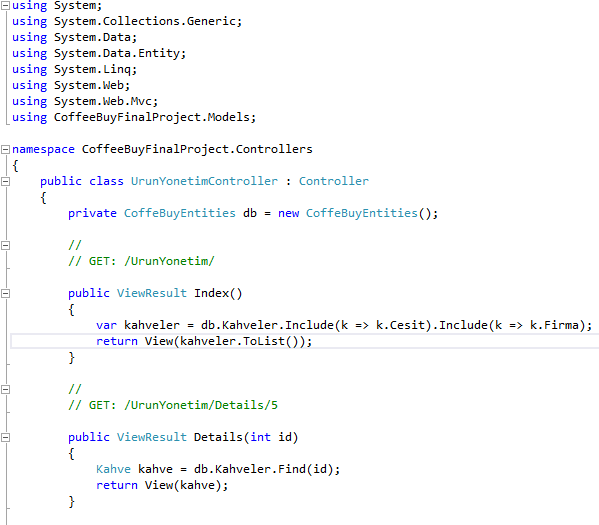
İlk önce Controller ‘a sağ tıklayıp yeni bir Controller oluşturuyoruz.



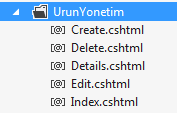


Yukardaki Seçenekleri uygulayıp Add tuşuna basıyorum.

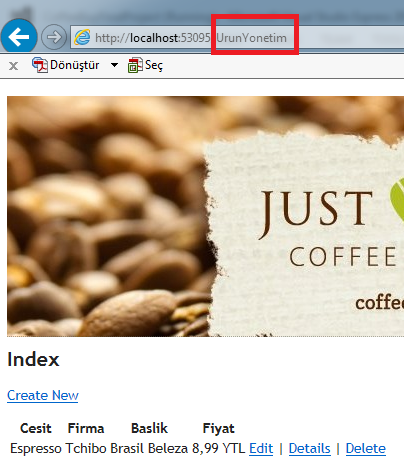
Altta ise Oluşan Sınıfın bir görüntüsü



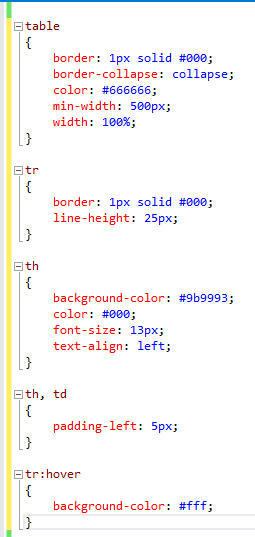
View kısmında oluşturulan Sınıflar



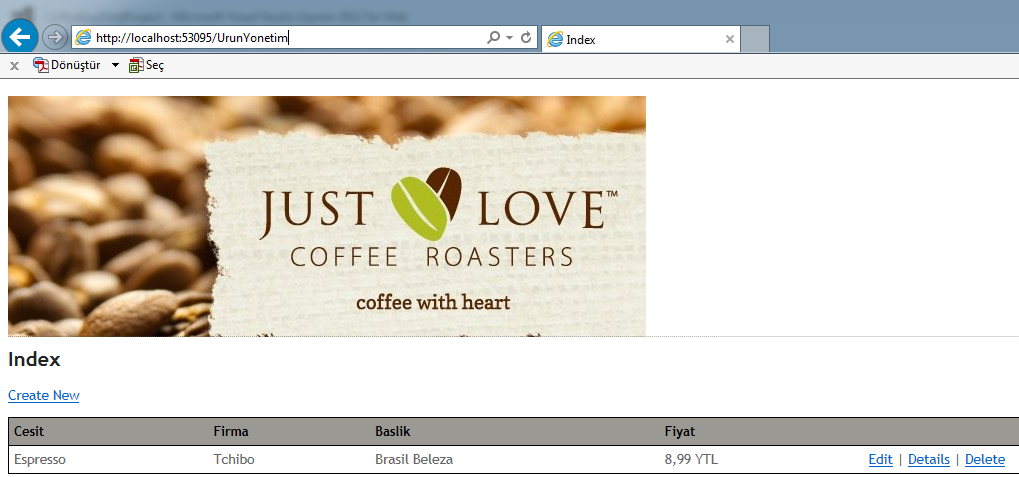
İlk olarak sitemi çalıştırıyorum



Daha sonra Site.css kısmına aşağıdaki verileri ekliyorum

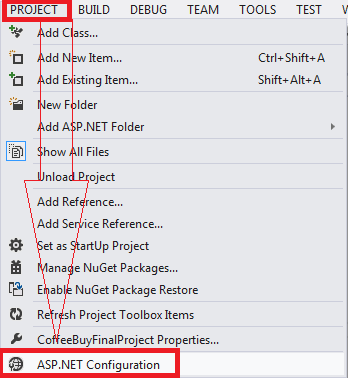


Sitemi tekrar çalıştırdığımda

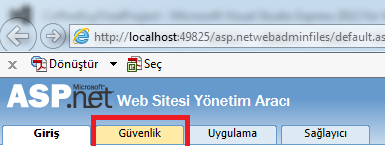


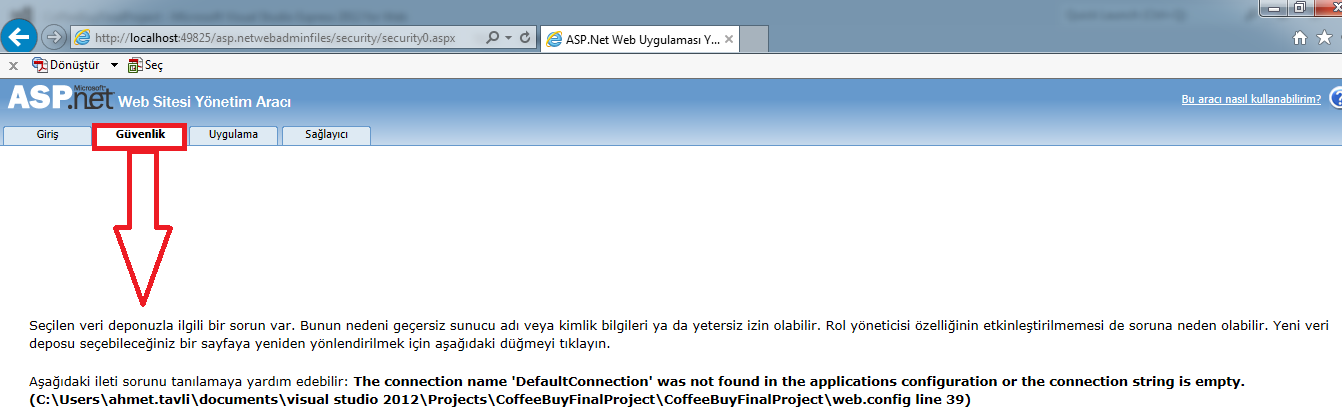
1. Admin Kullanıcısı Eklemek

Kahve Eklemek , Çıkarmak veya Silmek Admin Kullanıcısının YAapabileceği işler olduğundan dolayı sistemem Admin Kullanıcısı Eklemem Gerekiyor.



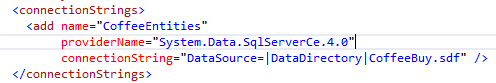
ASP.NET Configuration a tıkladığımda karşıma çıkan Sayfada Güvenliğe tıklıcam



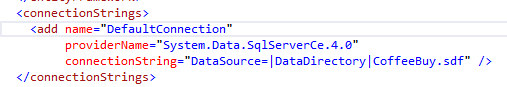
Güvenlik Tabına tıkladğımda aşağıdaki sorunla karşılaşıyorum.

Sorunun web.config deki connectionString de olduunu söylüyor.

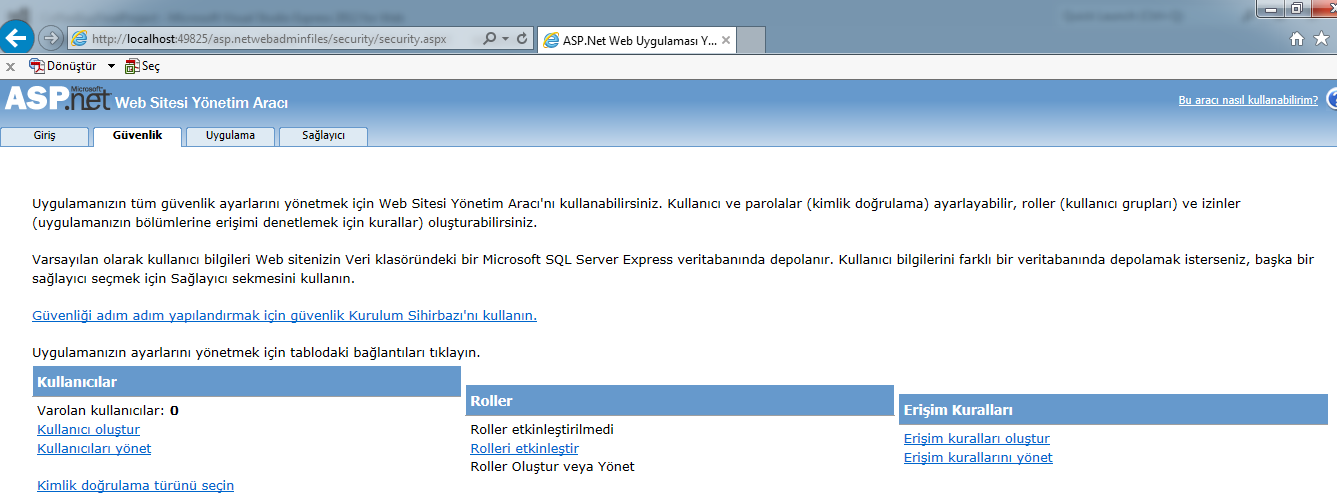
WebConfig deki ConnectionString verilerim



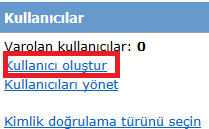
Burda add name = “DefaultConnection” yapıyorum.



Projemi Derleyip tekrar ASP.NET Configuation ı çalıştırdığımda



Burdan Kullanıcılar kısmından Kullanıcı oluştur ‘a tıklıyorum



Daha sonra Admin kullanıcısını ekliyorum.