

# ENTITY FRAMEWORK

Dr. Öğr. Üyesi Fatma AKALIN

Bu derste veritabanı işlemlerine deðinilecek. Veri tabanı işlemlerini yaparken entity framework kullanılacak. Veri tabanı tarafına geçilecek ve veritabanı programı üzerinden database oluşturulacaktır.

Bu doğrultuda Visual studioda MVC projemizin içerisine bir veritabanı oluşturmak için kod yazacağız. Sonra programı ilk çalıştırduğımızda ilgili veritabanının burada olmasını sağlayacağız.

Dr. Öğ.

İlk aşamada classlarla kod yapısı tarafından bir veritabanı kuracağız. Bu veritabanının program çalıştığında oluşmasını sağlayıp insert update select ve delete işlemlerini nasıl gerçekleştireceğimizi entity framework controls yaklaşımı ile gösteriyor olacağız. (ANCAK entity framework detayına girmeyeceğiz.)

MVC projelerimizde veritabanı tarafından veri tutmak istediğimizde hızlı bir şekilde veritabanı programlamayı yani herhangi bir MSSQL serveri kullanmayı bilmiyorsak kodlarımızı yazıp nasıl hızlı bir şekilde database oluştururuz ona degeneceğiz ve ilgili veritabanı için örnek datalarda basacağız.

Böylece hem veritabanı hem de örnek data oluşturulacak ve program çalışlığında karşımıza veriler gelecek. Ardından insert updata delete ve select işlemleri yapıyor olacağız .

Entity Framework ile kodlamaya başlamadan önce ORM  
(object relational mapping) kavramının ne olduğunu  
öğrenelim.

Dr. Öğr. Üyesi Fatma AKALIN

# ORM KAVRAMI

ORM veri tabanında oluşturulan her bir tabloya karşılık uygulama tarafında bir nesne oluşturma işidir. Bu işlem bazı frameworkler de ara yazılımlar sayesinde (ORM Tools), bazı frameworkler de ise elle gerçekleştirilmektedir

Dr. Öğr. ÜYESİ FETTUHALIN

# ORM ile Neler Yapılabilir?

ORM sayesinde SQL sorguları ile yapılan birçok işlem SQL sorgusu kullanılmadan gerçekleştirilmektedir. Örneğin veritabanından veri çekmek için SELECT sorgusu oluşturmak yerine oluşturulan nesnenin ID'si olan x elemanı gelsin diyerek veritabanından o elemanı alabiliriz. Başka bir şekilde örneklersek adı y olan elemanları Liste şeklinde getirmek istiyorsak «bu nesneden adı y olanları getir» diyerek SELECT sorgusu yazıp sonra onu list'e atma eylemi ile uğraşmadan getirebiliyoruz.

Dr.

# ENTITY FRAMEWORK

Entity Framework, bir ORM aracıdır. Veritabanındaki tablolara doğrudan (SELECT, INSERT, UPDATE vb.) sorgularla erişip işlem yapmak yerine nesneler üzerinden işlem yapmayı sağlar.

Özetle “Entity Framework, .NET geliştiricilerinin .NET nesnelerini kullanarak bir veritabanıyla çalışmasını sağlayan bir nesne ilişkisel eşleştiricisidir (O/RM). Geliştiricilerin genellikle yazması gereken veri erişim kodunun çoğuna olan ihtiyacı ortadan kaldırıyor.

# Entity Framework 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 gibi ORM araçları farklı veri sağlayıcıları (SQL Server, MySQL, SQLite gibi) için aynı komutları kullanarak işlem yapmaya imkan verir.

**ENTITY FRAMEWORK  
YAPISI DERSİN AKIŞINDA  
DETAYI BİR ŞEKİLDE  
AÇIKLANACAKTIR...**

**Dr. Oğuzhan**

# Veritabanı Tablolarımıza Ait Class'ları(Entities) Oluşturma

## Entity Framework CodeFirst Entegrasyonu

Bir veritabanımız olsun. İçerisinde kişiler ve adresler isminde iki ayrı tablo barındırsın. İki tablo yeter bize çünkü amacımız veritabanı oluşturmak değil, code first ile mvc üzerinde insert update ve delete işlemlerini yapmaktır.

Tabloların, primary keyi ve özellikleri olacak. Aralarında 1'e sonsuz bir ilişki olacak. Yani iki tablo arasında 1 kişinin sonsuz kaydı olacak hiyerarşisi tasarlanaacak.

Veritabanımızda iki tablo olduğu için bu iki tabloyu temsil eden iki tane classımızın tasarlaması gerekmektedir. Şimdi bu yapıları oluşturalım.

# Boş bir mvc projesi açtık.

Yeni projenizi yapılandırın

ASP.NET Web Uygulaması (.NET Framework)

Proje adı

denemeonikinciders

Konum

D:\aspnet\1\_ADERSLER

Çözüm adı [\(i\)](#)

denemeonikinciders

Çözümü ve projeyi aynı dizine yerleştir

Altyapı

.NET Framework 4.6.2

Proje "D:\aspnet\1\_ADERSLER\denemeonikinciders"

## Yeni ASP.NET Web Uygulaması oluştur



Boş

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



Web Forms

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



MVC

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



Web API

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



Tek Sayfalı Uygulama

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

### Kimlik Doğrulama

Yok

### Klasörleri ve çekirdek başvurularını ekle

- Web Forms
- MVC
- Web API'si

### Gelişmiş

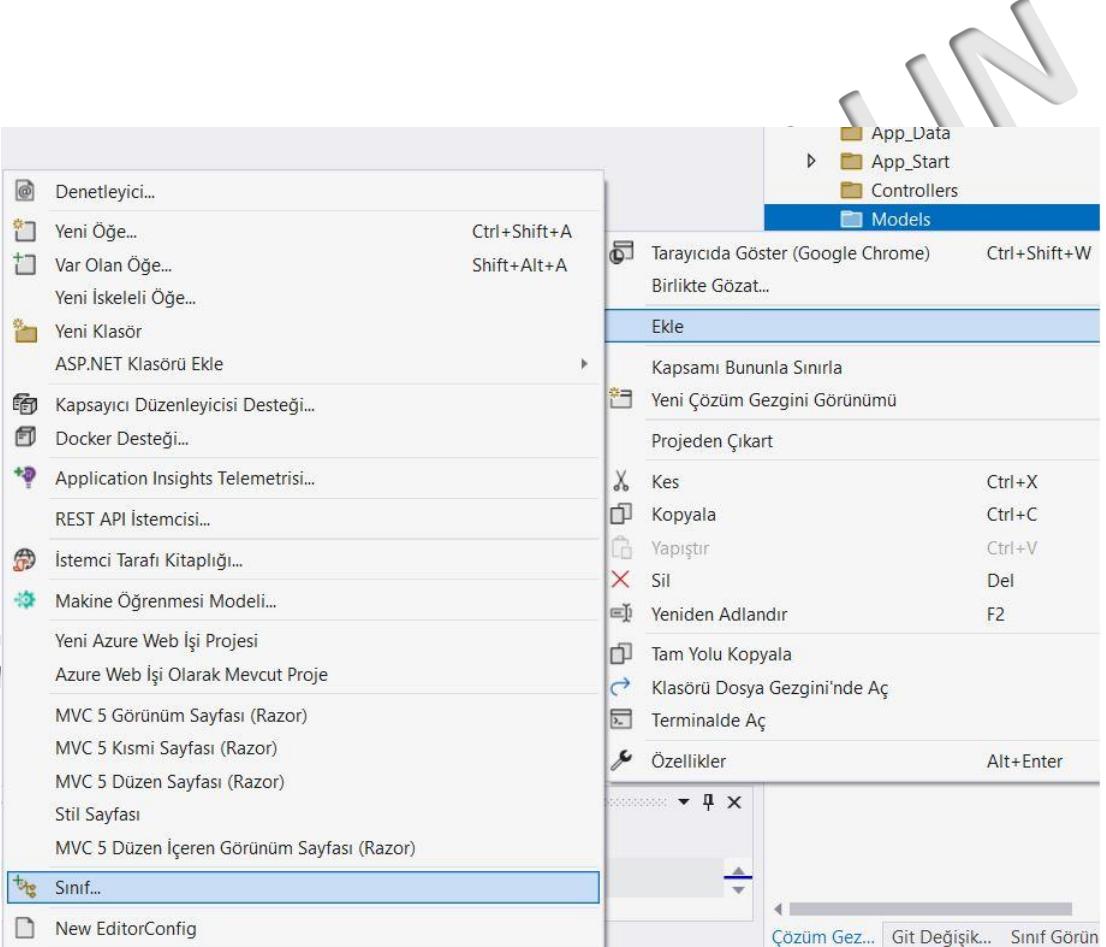
- HTTPS'yi Yapılandır
- Docker desteği  
(Docker Desktop gereklidir)
- Ayrıca birim testleri için bir proje oluştur  
denemeonikinciders.Tests

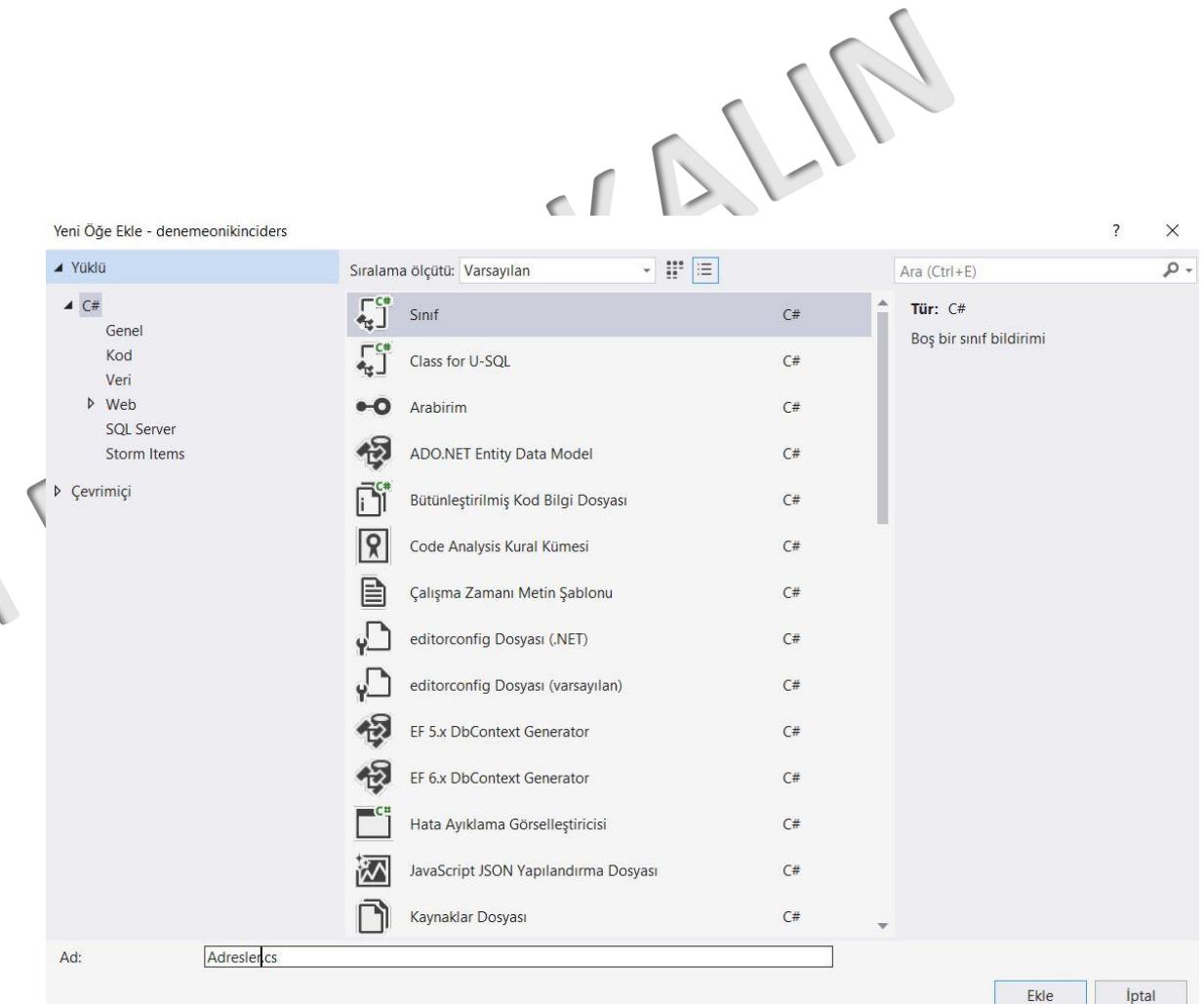
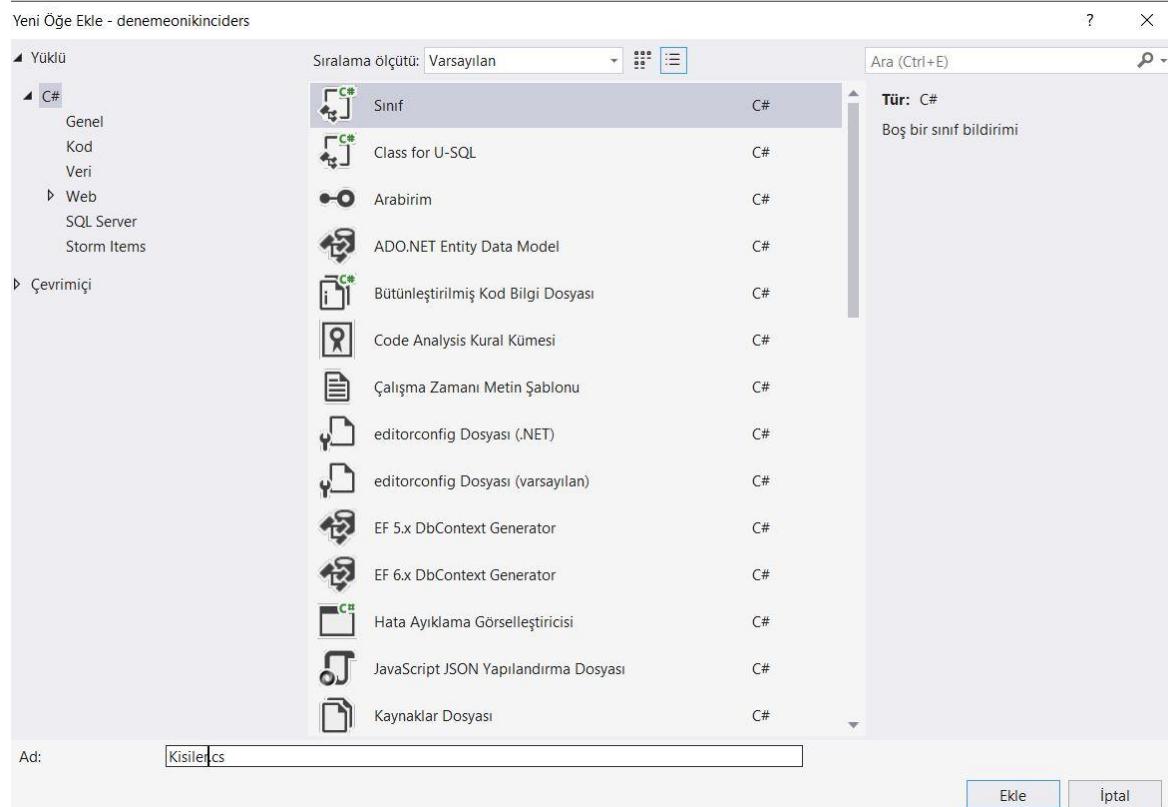
Geri

Oluştur

Ardından models klasörü  
içerisinde 2 ayrı sınıf  
oluşturacağım

Dr. Öğr. Üye





Kisiler.cs ve Adresler.cs şeklinde tanımlanan iki ayrı sınıf içerisinde propertyleri eklemeliyim.

Adresler.cs ➔ X Kisiler.cs

denemeonikinciders denemeonikinciders

```
1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel.DataAnnotations;
4  using System.ComponentModel.DataAnnotations.Schema;
5  using System.Linq;
6  using System.Web;
7
8  namespace denemeonikinciders.Models
9  {
10    [Table("Adresler")]
11    public class Adresler
12    {
13      [Key, DatabaseGenerated(DatabaseGeneratedOption.Identity)]
14      public int ID { get; set; }
15
16      [StringLength(300), Required]
17      public string AdresTanim { get; set; }
18
19      public virtual Kisiler Kisi { get; set; }
20    }
}
```

Kisiler.cs ➔ X

denemeonikinciders denemeonikinciders.Models.Kisiler

```
1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel.DataAnnotations;
4  using System.ComponentModel.DataAnnotations.Schema;
5  using System.Linq;
6  using System.Web;
7
8  namespace denemeonikinciders.Models
9  {
10    [Table("Kisiler")] //Veri tabanı tablosu adı
11    public class Kisiler
12    {
13      [Key, DatabaseGenerated(DatabaseGeneratedOption.Identity)]
14      public int ID { get; set; }
15
16      [StringLength(20), Required]
17      public string Ad { get; set; }
18
19      [StringLength(20), Required]
20      public string Soyad { get; set; }
21
22      public int Yas { get; set; }
23
24      public virtual List<Adresler> Adresler { get; set; } //Tablolar arası ilişkili olanları virtual ile tanımlıyoruz
25    }
}
```

1 kişinin birden fazla adrese sahip olma durumundan  
ötürü Kisiler.cs dosyasına ek olarak list yapısında  
Adresler de eklemeliyiz.

Entity framework classlarımızı oluşturken tablolar  
arasında ilişkiliyi tanımlayan proportileri de virtual  
olarak tanımlamalıyız.

Burada Entity frameworkte tablolarımızı temsil eden classlarımız için bazı tanımlamalar yapacağız. Öncelikle veri tabanı tablosunun adını istediğiniz gibi verebiliriz.

```
[Table("Kisiler")] //Veri tabanı tablosu adı
```

Akabinde table attributesinin usingi eklenilmelidir.

```
using System.ComponentModel.DataAnnotations.Schema;
```

ID'nin anahtar kolon olduğunu key attributesi [Key] ile belirtiyoruz. Yine burada using eklenmelidir.

(**using** System.ComponentModel.DataAnnotations;).

Bu işlem ile anahtar kolon olduğunu belirttik ancak otomatik artan özelliği eklemek istersek DatabaseGenerated isimli bir class yardımıyla bu classa DatabaseGeneratedOption enumundan ilgili kolonun identity olmasını sağlayabilirsin. Böylece anahtar *primary key* ve *Identity otomatik artan* özellikleri eklenmiş olacak.

[Key , DatabaseGenerated(DatabaseGeneratedOption.Identity)]

Ad ve soyad kolonunun başına da maximum karakter sınırı ve boş geçilemez tanımlaması yapalım. [StringLength(20), Required]

Bu tanımlamalara göre veritabanındaki Kişiler tablomuz oluşurken gerekli isimler ve özellikler dahil ediliyor olacak Bu oluşum aşamasında...

Aynı işlemleri adresler tablosunda da yapalım.

AdresTanim properties için required yazmayacağım çünkü boş geçilebilir olmasını istiyorum.

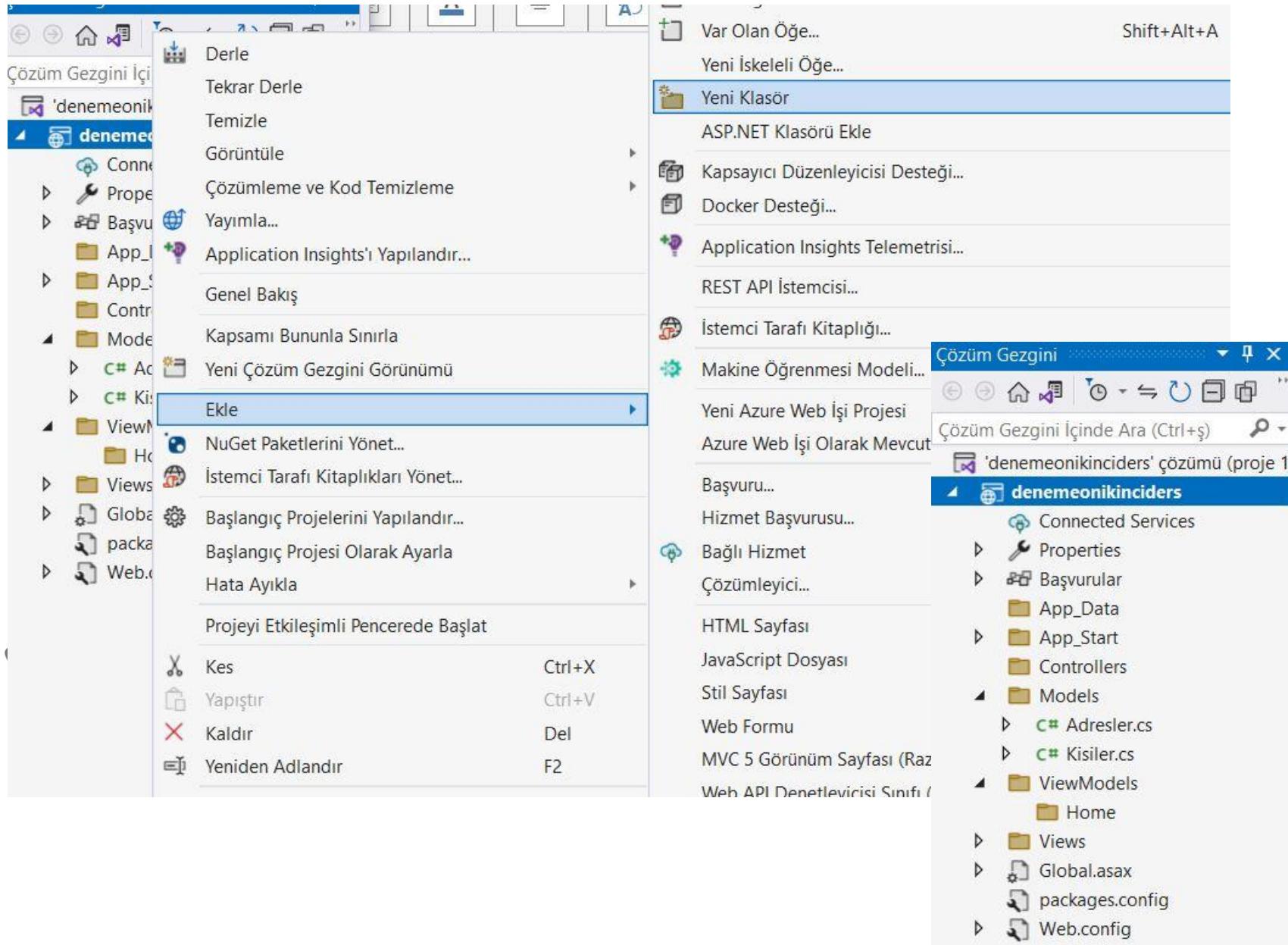
Bu tanımlamalar ile Kişiler classımızın ve Adres classımızın karşılıklarındaki tabloların olmasını sağlayacak classları yapmış olacağız.

Model'in altında anlatılan örnekte birden fazla sınıf var ise bu iki sınıfın birleştirilmesi gerektiğini konuşmuştu. Hem tek sınıf hem de iki sınıf için örnekler yapmıştık.

Şimdi Ana proje klasörünün altında ViewModels isminde yeni bir klasör oluşturacağım.

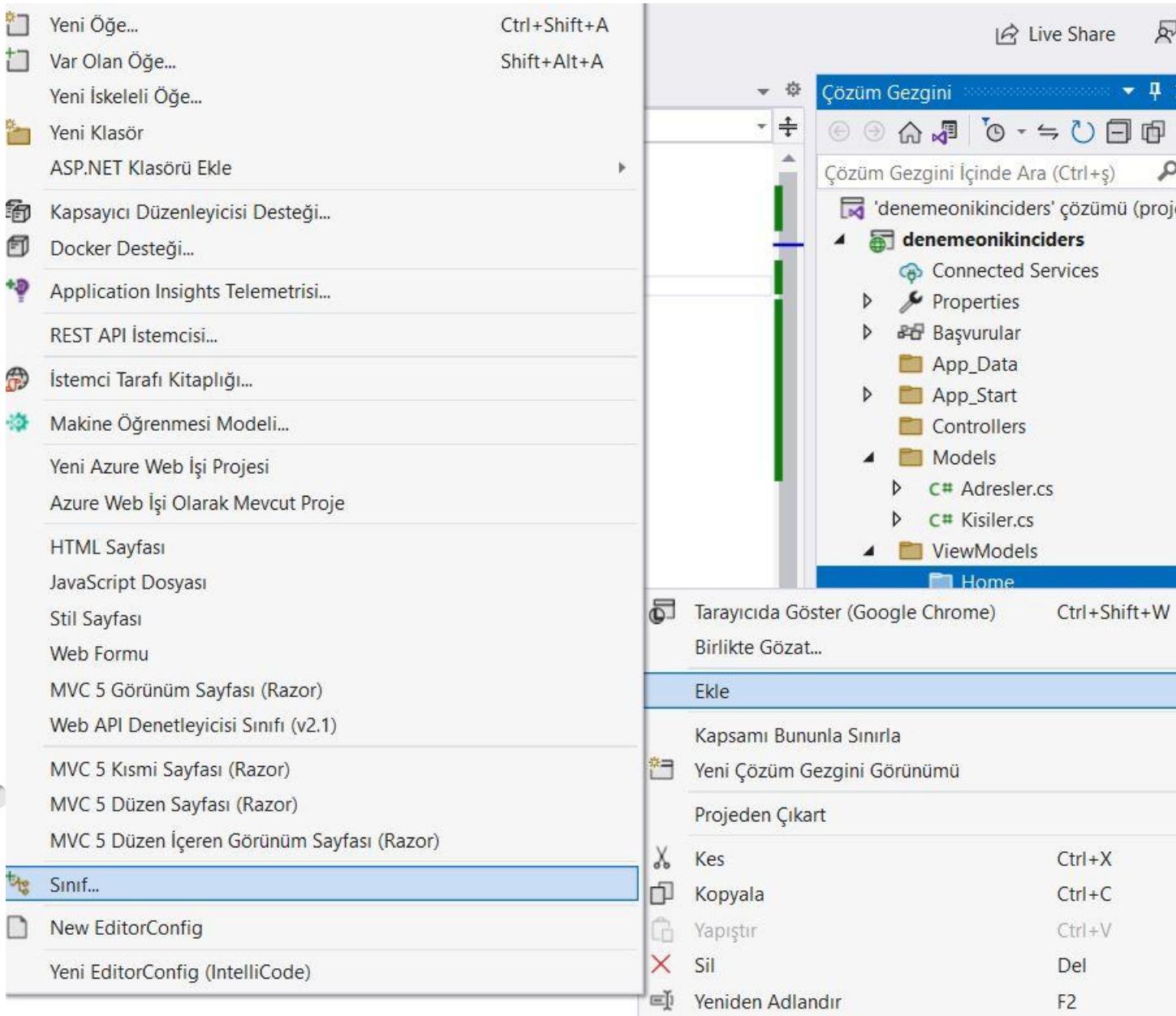
Onun altında da Home klasörü oluşturacağım. Bu son Home klasörün içerisinde HomePageViewModel.cs isimli sınıfımı ekleyeceğim. Bu sınıfta Kisiler ve Adresler sınıflarını birleştirmeyi hedefliyorum.

Dr. Öğr.



Kisiler ve  
Adresler  
sınıflarını  
birleştireceğim  
sınıfı  
oluşturacağım  
şimdi.

Dr. Öğr.



```
HomePageViewModel.cs
```

```
denemeonikinciders
```

```
denemeonikinciders.Views
```

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Web;
5
6  namespace denemeonikinciders.ViewModels.Home
7  {
8      public class HomePageViewModel
9      {
10  }
11 }
```

```
HomePageViewModel.cs
```

```
denemeonikinciders
```

```
denemeonikinciders
```

```
1  using denemeonikinciders.Models;
2  using System;
3  using System.Collections.Generic;
4  using System.Linq;
5  using System.Web;
6
7  namespace denemeonikinciders.ViewModels.Home
8  {
9      public class HomePageViewModel
10     {
11         public List<Kisiler> Kisiler { get; set; }
12         public List<Adresler> Adresler { get; set; }
13     }
14 }
```

```
Çözüm Gezgini
```

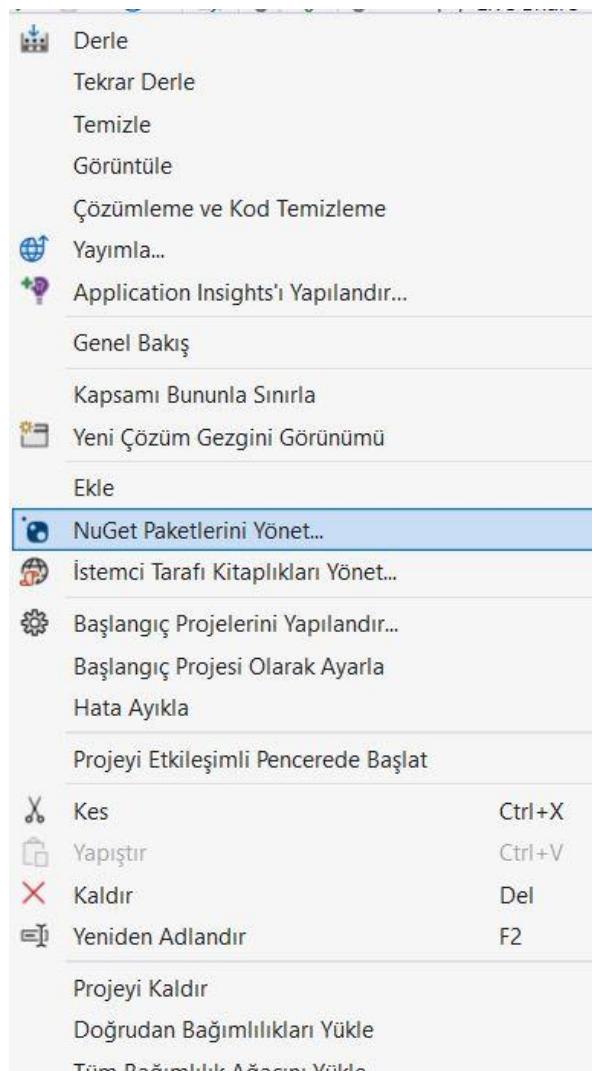
```
Çözüm Gezgini İçinde Ara (Ctrl+Ş)
```

```
'denemeonikinciders' çözümü (proje 1 / 1)
```

```
denemeonikinciders
    Connected Services
    Properties
    Başvurular
    App_Data
    App_Start
    Controllers
    Models
        C# Adresler.cs
        C# Kisiler.cs
    ViewModels
        Home
            C# HomePageViewModel.cs
        Views
        Global.asax
        packages.config
        Web.config
```

**using denemeonikinciders.Models;**  
eklemeyi unutmamalısın!!!

Önemli bir nokta da Entity Framework’ü Nuget Manager’dan indirmektir.  
(Aynı zamanda bir önceki derste sunulan yapılarda indirilmeli) Ancak  
tekrardan hatırlatalım...



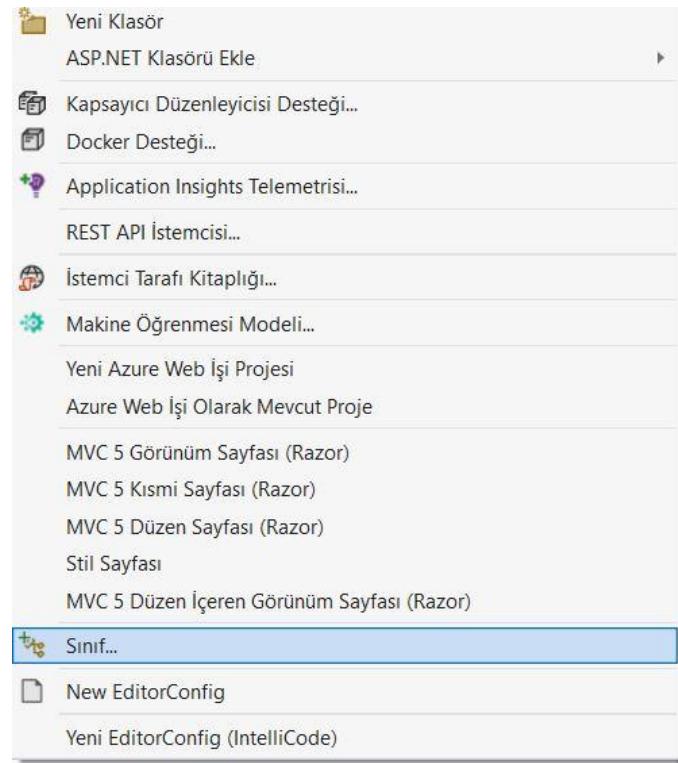
A screenshot of the Nuget.org search results for 'bootstrap'. The search bar shows 'bootstrap'. The results list 'bootstrap' by 'The Bootstrap Team' at version 5.3.1, described as 'Bootstrap framework in CSS. Includes JavaScript.' Below it are results for 'Modernizr', 'jQuery', 'jQuery.Validation', 'EntityFramework', and 'Microsoft.EntityFrameworkCore'.

Paket	Yayıncı	Sürüm	İndirme Sayısı	Açıklama
bootstrap	The Bootstrap Team	5.3.1	162M	Bootstrap framework in CSS. Includes JavaScript.
Modernizr	Modernizr	En yeni kararlı 2.8.3	3.7.0	Modernizr is a small and simple JavaScript library t...
jQuery	jQuery Foundation, Inc.	1.19.5	162M	jQuery is a fast and concise JavaScript Library that simplifies HTML document traversing, event handling...
jQuery.Validation	Jörn Zaefferer	1.19.5	82,4M	jQuery.Validation is a plugin for jQuery that adds form validation capabilities to your web application.
EntityFramework	Microsoft	6.4.4	225M	Entity Framework 6 (EF6) is a tried and tested object-relational mapper for .NET with many years of feature de...
Microsoft.EntityFrameworkCore	Microsoft	7.0.10	7.0.10	Microsoft.EntityFrameworkCore is a core component of Entity Framework Core, providing the Entity Data Model (EDM)...

## VERİTABANI İŞLEMLERİ YÖNETECEK SINIFI(CONTEXT) OLUŞTURMA

Veri tabanı tablolarımızı temsil eden classları oluşturduk. Şimdi Veritabanı üzerinde işlem yapacak(sorguları çalıştıracak, connectionu açıp kapayacak,) manager classımızı oluşturalım.

Bunun için Models klasörünün içerisinde Managers isminde bir klasör tanımlayıp bu klasöre DatabaseContext.cs isminde bir class eklemeliyim. Ardından bu classımızın bizim veritabanı işlemlerimizi yapacak class olduğunu belirtmek için DbContext ismindeki classtan miras almalı ve **using System.Data.Entity;** bilgisini eklemeyi unutmamalıyım. Miras almamız, bizim databasecontext classımızı veritabanı işlemleri yapmak için yetkili kılacaktır.



Çözüm Gezgini  
Çözüm Gezgini İçinde Ara (Ctrl+Ş)

'denemeonikinciders' çözümü (proje 1 / 1)

- denemeonikinciders
  - Connected Services
  - Properties
  - Başvurular
  - App\_Data
  - App\_Start
  - Content
  - Controllers
  - Models
  - Managers** (highlighted in blue)

Tarayıcıda Göster (Google Chrome) Ctrl+Shift+W  
Birlikte Gözat...

Ekle

Kapsamı Bununla Sınırla

Yeni Çözüm Gezgini Görünümü

Projeden Çıkart

Kes Ctrl+X

Kopyala Ctrl+C

Vapıstır Ctrl+V

Yükülu

Sıralama ölçütü: Varsayılan

Ara (Ctrl+E)

Tür: C#

Boş bir sınıf bildirimi

	Sınıf C#
	Class for U-SQL C#
	Arabirim C#
	ADO.NET Entity Data Model C#
	Bütünleştirilmiş Kod Bilgi Dosyası C#
	Code Analysis Kural Kümesi C#
	Çalışma Zamanı Metin Şablonu C#
	editorconfig Dosyası (.NET) C#
	editorconfig Dosyası (varsayılan) C#
	EF 5.x DbContext Generator C#
	EF 6.x DbContext Generator C#
	Hata Ayıklama Görselleştirici C#
	JavaScript JSON Yapılandırma Dosyası C#
	Kaynaklar Dosyası C#

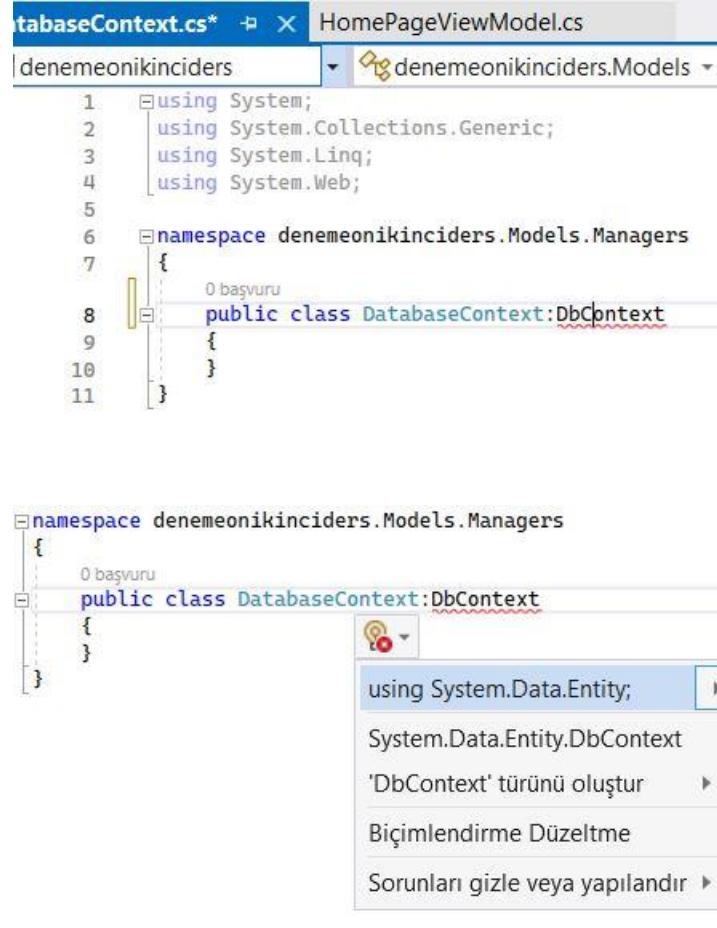
Ad:

Ekle İptal

AKALIN

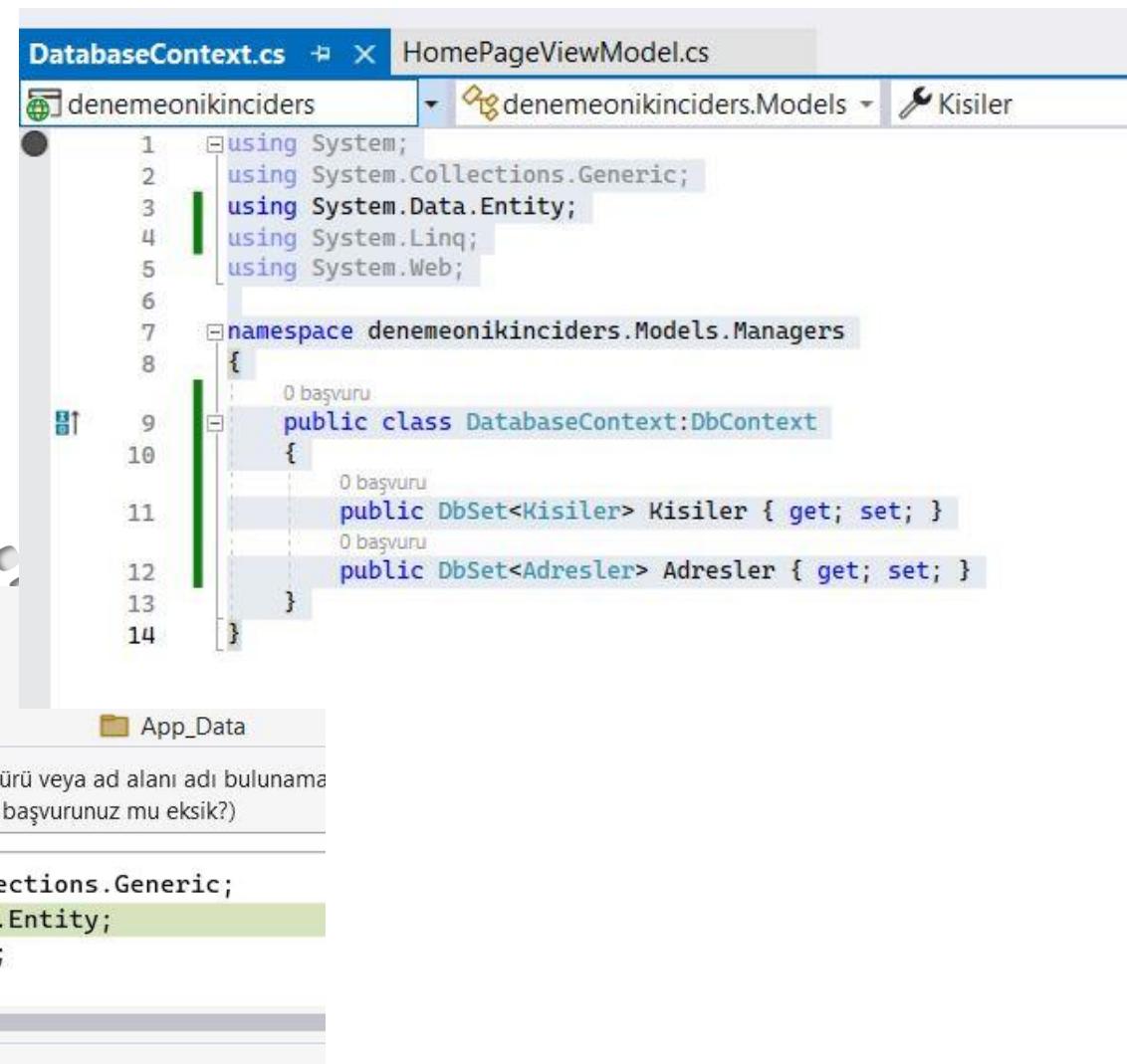
Dr. Öğr.

# Alt+Enter



The screenshot shows a code editor window with two tabs: `DatabaseContext.cs*` and `HomePageViewModel.cs`. The `DatabaseContext.cs*` tab is active. A context menu is open over the `using System.Data.Entity;` line in the `DatabaseContext.cs` code. The menu items are:

- using System.Data.Entity;
- System.Data.Entity.DbContext
- 'DbContext' türünü oluştur
- Birimlendirme Düzeltme
- Sorunları gizle veya yapılandır



The screenshot shows a code editor window with two tabs: `DatabaseContext.cs` and `HomePageViewModel.cs`. The `DatabaseContext.cs` tab is active. A context menu is open over the `using System.Data.Entity;` line in the `DatabaseContext.cs` code. The menu items are:

- using System;
- using System.Collections.Generic;
- using System.Data.Entity;
- using System.Linq;
- using System.Web;

A tooltip message is displayed below the menu:  
CS0246 'DbContext' türü veya ad alanı adı bulunamadı. Yüzeriniz veya derleme başvurunuz mu eksik?

Bizler Kişiler ve adresler tablomuzun class karşılığını oluşturmuştuk. Fakat bu class karşılıkları genelde o tabloların bir satırındaki verileri tutmak için oluşturulan classlardır. Bizim tablomuzu temsil edecek özellikler ise DbSet tipinde oluşturulan yapılardır. Tablolarımızın içinde insert, update ve delete işlemleri yaparken bu proportleri kullanıiyor olacağız.

Tabii işlemlerimiz yaparken kullanacağımız classlar o tablonun class türleri olacaktır.

Bununla birlikte yapacağımız işlemler devam ediyor...

Veritabanı yoksa ve veritabanı olacaksa bu veritabanının oluşması esnasında örnek datalarında insert olmasını sağlamak istiyorum. Bunun için initializer denilen classlar oluşturulur. VeriTabaniOlusturucusu olarak nitelendirdiğim bu classı CreateDatabaseIfNotExist 'ten miras aldirecagım. Burada database yoksa oluşturan tipinde bir classtan miras alarak bir oluşturucu class ekliyorum. Generic bir class olduğu için içine ne ile çalışacağını vermek gerek. **Databasemizi temsil eden Contextimizi bunun içine ekliyorum.** Bu şekilde buradaki işlemi yapacak olan oluşturucu classımızı eklemiş oluyoruz.

Böylece database yoksa database oluşturma eylemini tetikleyecek olan işlemi yaptık.

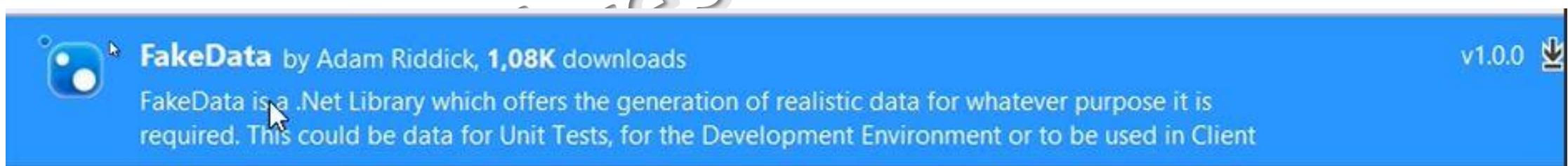
Bu classımızın içinde gelen iki tane metodu inceleyelim..

**1-InitializeDatabase** -> Database oluşturulmadan önce çalışan metodumuz.

**2-Seed** -> Database oluşturulduktan sonra çalışan metodumuz.

Örneğin database oluşturulduktan sonra kod yazmak istiyorsan. (Şu an öyle yapacağımız çünkü biz örnek data basacağız ve bunun için database'in olması gereklidir). O zaman Seed metodunu ezip uygun kodlarımızı yazacağız. Bu metod bize databaseyi temsil eden context nesnesini de getiriyor.

Bununla birlikte bir hatırlatma yapmak istiyorum. Örnek data basarken yine nuget paketinden yararlanacağım (Fakedata dll'inden). İçerisinde kişi isimleri, lokasyon bilgileri, telefon numaralı, adres isimleri gibi örnek data basabilecek metodlar geliyor. Örnek veriler türetmemize hiç gerek kalmadan projemizde kullanabiliyor olacağız. Install ettikten sonra References altına ilgili dll ekleniyor. Şimdi fakedatayı kullanacağımız yani databasemize insert edeceğiz.



DatabaseContext.cs isimli sınıfımızın kodları aşağıda sunulmaktadır. Kodlar üzerinde süreci yeniden tekrar edelim.

İlk olarak kişilere ilişkin ad soyad ve yaş bilgilerini ekleyeceğiz. Ardından bu kişilere adres bilgisi gireceğiz. Veritabanı oluşturucu classımızın override ettiğimiz seed metodunu kodu çalıştırıp veri tabanı oluşturuktan sonra 10 tane kişinin olmasını ve her bir kişileri çekip içlerinde dönüp her bir kişi için 1-5 arasında random sayıda bir adres insert etmesini sağlamış olacağız.

**Şimdi bu oluşturucu classımızın (DatabaseContext.cs) veritabanı oluştugunda çalışmasını sağlamamız** için veritabanımı temsil eden ilgili **DatabaseContext classımızın constructörüne bu oluşturucuyu eklemek** gerek. Bu doğrultuda ilgili veritabanı oluşturucu classımızı newleyip aşağıdaki gibi yeniden yazmalıyım.

```
public class DatabaseContext : DbContext
{
    public DbSet<Kisiler> Kisiler { get; set; }
    public DbSet<Adresler> Adresler { get; set; }
    public DatabaseContext()
    {
        Database.SetInitializer(new VeriTabaniOlusturucu());
    }
}
```

Böylece databaseyi initializer ediyorum. Burada veritabanı yoksa çalış dediği için veritabanı yoksa çalışan bir class set etmiş oldum veritabanı ayağa kalktığı anda.

Artık herhangi bir sayfadan veritabanımıza bir select atmamız için veritabanına erişen bir kod MVC'de yazabilirim.

DatabaseContext.cs X HomePageViewModel.cs

```
denemeonikinciders
1  using System;
2  using System.Collections.Generic;
3  using System.Data.Entity;
4  using System.Linq;
5  using System.Web;
6
7  namespace denemeonikinciders.Models.Managers
8  {
9      public class DatabaseContext : DbContext
10     {
11         public DbSet<Kisiler> Kisiler { get; set; }
12         public DbSet<Adresler> Adresler { get; set; }
13         public DatabaseContext()
14         {
15             Database.SetInitializer(new VeriTabaniOlusturucu());
16         }
17     }
}
```

```
denemeonikinciders
1 1 başvuru
2  public class VeriTabaniOlusturucu : CreateDatabaseIfNotExists<DatabaseContext>
3  {
4      //public override void InitializeDatabase(DatabaseContext context)
5      //{
6      //    base.InitializeDatabase(context);
7      //}
8  0 başvuru
9  protected override void Seed(DatabaseContext context)
10 {
11     //Kişiler insert ediliyor
12     for (int i = 0; i < 10; i++)
13     {
14         Kisiler kisi = new Kisiler();
15         kisi.Ad = FakeData.NameData.GetFirstName();
16         kisi.Soyad = FakeData.NameData.GetSurname();
17         kisi.Yas = FakeData.NumberData.GetNumber(10, 90); //Min max arası
18
19         context.Kisiler.Add(kisi); //Bu adım 10 kişiyi benim databasemindeki kişiler tablosuna insertedecek
20     }
21     context.SaveChanges();
22
23     List<Kisiler> tumKisiler = context.Kisiler.ToList();
24     foreach (Kisiler kisi in tumKisiler)
25     {
26         for (int i = 0; i < FakeData.NumberData.GetNumber(1, 5); i++)
27         {
28             Adresler adres = new Adresler();
29             adres.AdresTanim = FakeData.PlaceData.GetAddress();
30             adres.Kisi = kisi;
31
32             context.Adresler.Add(adres);
33         }
34     }
35     context.SaveChanges();
36 }
```

Insert işlemlerinde DatabaseContext i kullanıyoruz.

DatabaseContext'i context temsil ediyordu ve context ile kişiler tablosuna benim kişi nesnemi eklemeyi sağlıyorum..

context.Kisiler.Add(kisi); //Bu adım 10 kişiyi  
benim databasemindeki kişiler tablosuna  
insertedecek

Ama insert daha gerçekleşmedi. Ne zaman gerçekleşecek?

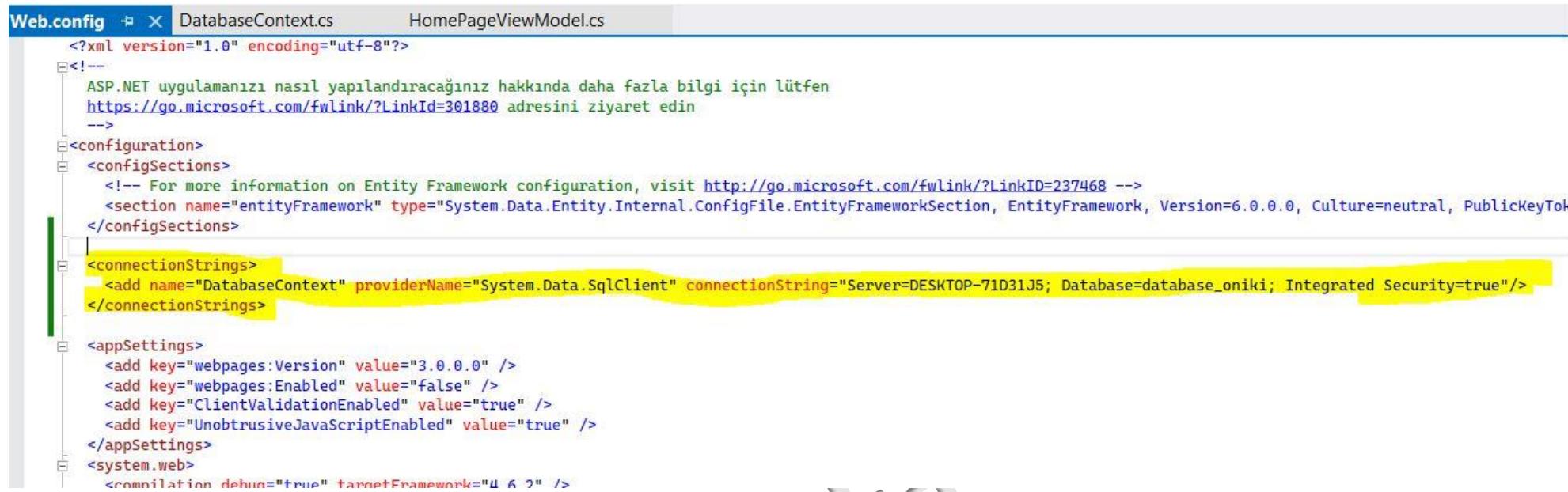
- EntityFramework Databasemi temsil eden classının  
SaveChanges metodunu çağrırdığında gerçekleşecektir.

Kişiler insert olduktan sonra bu kişiler nesnesini databaseden çekebiliriz.

List<Kisiler> tumKisiler=context.Kisiler.ToList(); kod parçası ile...

# Veritabanının Oluşturulması (İlk Çalıştırma)

Veritabanımızın oluşması için son bir şey kaldı. Veritabanımızın hangi servera hangi database ismi ile hangi yetkiler ile bağlanacağını tanımlamalıyız. Bunun için DatabaseContext gibi veritabanına erişen bir bağlantı cümlesine ihtiyacımız var. Bu bağlantı cümlesini **webconfig**'e ekleyeceğiz.



```
<?xml version="1.0" encoding="utf-8"?>
<!--
  ASP.NET uygulamanızı nasıl yapılandıracağınız hakkında daha fazla bilgi için lütfen
  https://go.microsoft.com/fwlink/?LinkId=301880 adresini ziyaret edin
-->
<configuration>
  <configSections>
    <!-- For more information on Entity Framework configuration, visit http://go.microsoft.com/fwlink/?LinkID=237468 -->
    <section name="entityFramework" type="System.Data.Entity.Internal.ConfigFile.EntityFrameworkSection, EntityFramework, Version=6.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" />
  </configSections>
  <connectionStrings>
    <add name="DatabaseContext" providerName="System.Data.SqlClient" connectionString="Server=DESKTOP-71D31J5; Database=database_oniki; Integrated Security=true" />
  </connectionStrings>
  <appSettings>
    <add key="webpages:Version" value="3.0.0.0" />
    <add key="webpages:Enabled" value="false" />
    <add key="ClientValidationEnabled" value="true" />
    <add key="UnobtrusiveJavaScriptEnabled" value="true" />
  </appSettings>
  <system.web>
    <compilation debug="true" targetFramework="4.6.2" />
  </system.web>

```

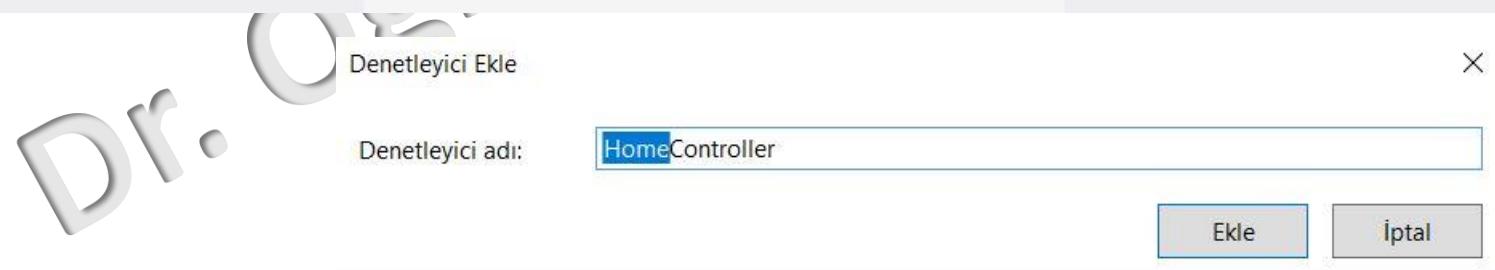
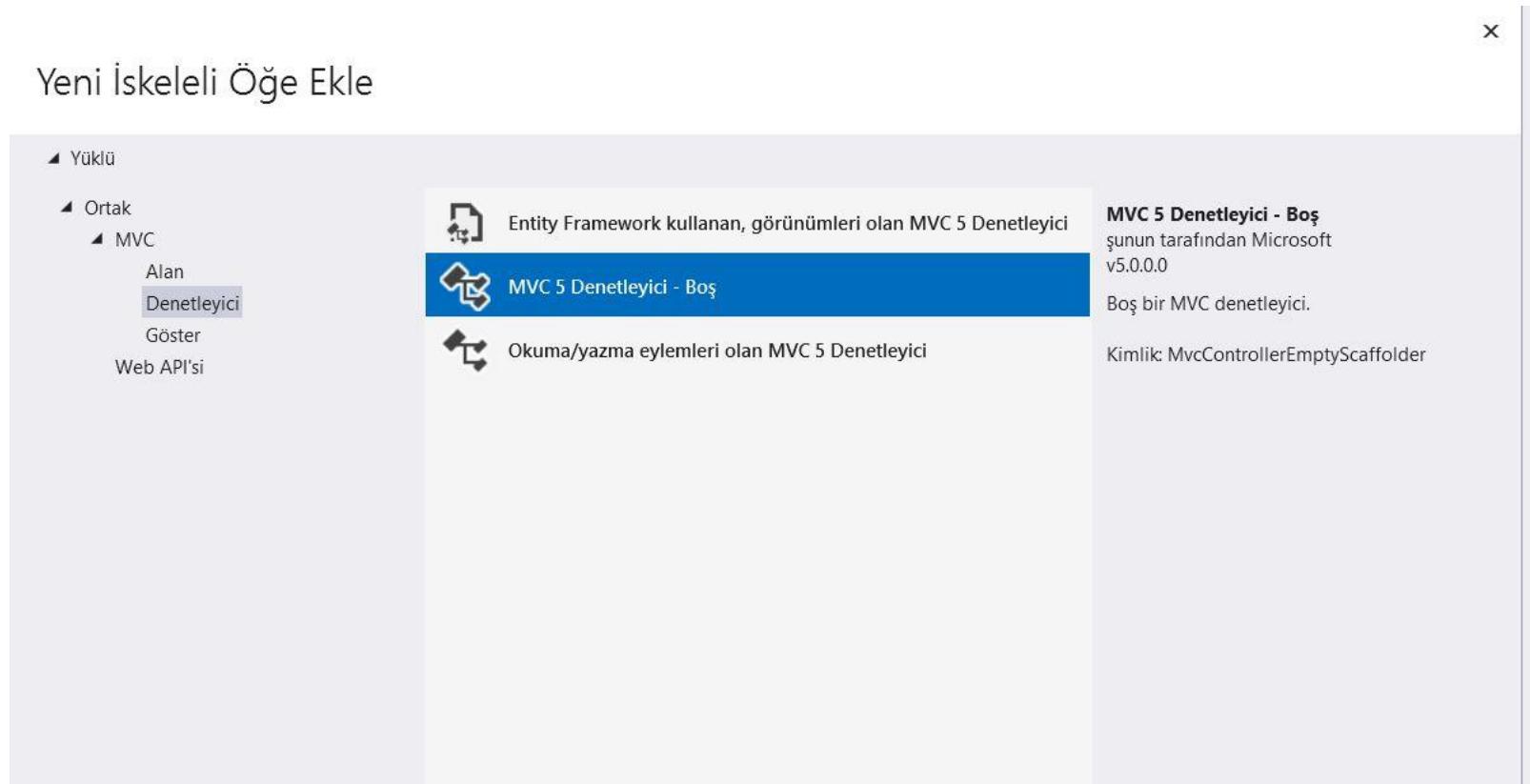
Bu bağlantı cümlesini configsectiondan sonra eklemeliyiz. ConfigSection en yukarıda olmak zorundadır.

ConnectionString etiketimi açtım.

```
<connectionStrings>
  <add name="DatabaseContext" providerName="System.Data.SqlClient" connectionString="Server=DESKTOP-71D31J5;
  Database=Database_bolum8; Integrated Security=true" />
</connectionStrings>
```

İlk olarak isim kısmı önemlidir ve ismin aynı olması gereklidir (DatabaseContext). İkinci olarak Provider name verelim. Yani databaseye bağlanırken kullanacağım dll nedir? Üçüncü olarak ConnectionStringi de biz oluşturacağız (ConnectionString ile servera bağlanacağım). Dördüncü olarak Database kelimesi ile veritabanımızın adı ne olacak onu yazacağım. Ek olarak veritabanına kullanıcı adı ve şifre ile bağlanılacak ise UserID=sasa; pwd=asas yazabilirsın ama eğer Windows yetkileri ile bağlanılacaksa Integrated Security=true yazmak yeterli. Böylece serverda ilgili database ismi oluşturulacak ve Windows yetkilerimle benim veritabanım oluşturulacak. (En dıştaki web-config)

BU işlemin ardından veritabanını oluşturabiliriz. Şimdi hızlı bir şekilde homecontroller ekleyeceğiz.



```
HomeController.cs + X
denemeonikinciders Controllers.cshtml Index()
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Web;
5  using System.Web.Mvc;
6
7  namespace denemeonikinciders.Controllers
8  {
9      public class HomeController : Controller
10     {
11         // GET: Home
12         public ActionResult Index()
13         {
14             return View();
15         }
16     }
17 }
```

```
//List<Kisiler> kisiler = db.Kisiler.ToList(); //Select * from Kisiler
```

```
HomeController.cs* + X homepage.cshtml modern
denemeonikinciders Controllers.cshtml homepage()
1  using denemeonikinciders.Models.Managers;
2  using denemeonikinciders.ViewModels.Home;
3  using denemeonikinciders.Models;
4  using System;
5  using System.Collections.Generic;
6  using System.Linq;
7  using System.Web;
8  using System.Web.Mvc;
9
10 namespace denemeonikinciders.Controllers
11 {
12     public class HomeController : Controller
13     {
14         // GET: Home
15         public ActionResult homepage()
16         {
17             DatabaseContext db = new DatabaseContext();
18
19             HomePageViewModel model = new HomePageViewModel();
20             model.Kisiler = db.Kisiler.ToList();
21             model.Adresler = db.Adresler.ToList();
22             return View(model);
23         }
24     }
25 }
26 }
```

veritabanına select atma işlevselligi gerçekleştiği gibi veritabanı oluşumunu da tetiklenecek. Veritabanı oluşumu tetiklendiğinde databaseContext classında bizim tanımladığımız initializer da devreye girecek ve veritabanı oluştuktan sonra kişiler datasını basacak. 10 kişi datası insert edecek ve ilgili kişiler üzerinden adresleri insert edecek. Ondan sonra homecontrollerda kişileri select edip değişkene atayacak. Yani ilk önce veritabanını ayağa kaldırıracak ve ardından bizim yaptığımız ayarlar doğrultusunda verileri bize sunuyor olacak. Sayfaya göndermek için return View(model);

## Yeni İskeleli Öğe Ekle

▲ Yüklü

- ◀ Ortak
- ◀ MVC
  - Alan
  - Denetleyici
  - Göster
- Web API'si

 **MVC 5 Görünümü**

**MVC 5 Görünümü**  
şunun tarafından Microsoft  
v5.0.0.0

Türü kesin belirtilmiş bir Modelle veya  
Model olmadan MVC görünümü

Görünüm Ekle X

Görünüm adı:

Şablon:

Model sınıfı:

Veri bağlamı sınıfı:

Seçenekler:

Kısıtlı görünüm olarak oluştur

Başvuru betik kitaplıklarını

Bir düzen sayfası kullanın:  
 ...

(Bir Razor \_viewstart dosyasında ayarlandıysa, boş bırakın)

Ekle İptal

Homepage View'ını ekliyoruz

Son adımda homepage view sayfasını oluşturacağız. İndirdiğimiz css ve script dosyalarını buraya ekleyeceğiz.

```
<link href="~/Content/bootstrap.min.css" rel="stylesheet" />
<link href="~/Content/Site.css" rel="stylesheet" />
<script src="~/Scripts/jquery-3.7.1.min.js"></script>
<script src="~/Scripts/bootstrap.min.js"></script>
<script src="~/Scripts/modernizr-2.8.3.js"></script>
```

Homepage sayfasının en tepesine bir model tanımı yapmayı kesinlikle unutmayalım

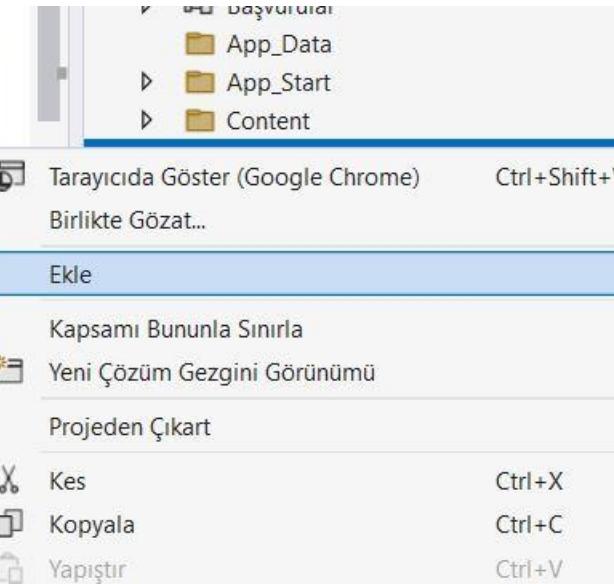
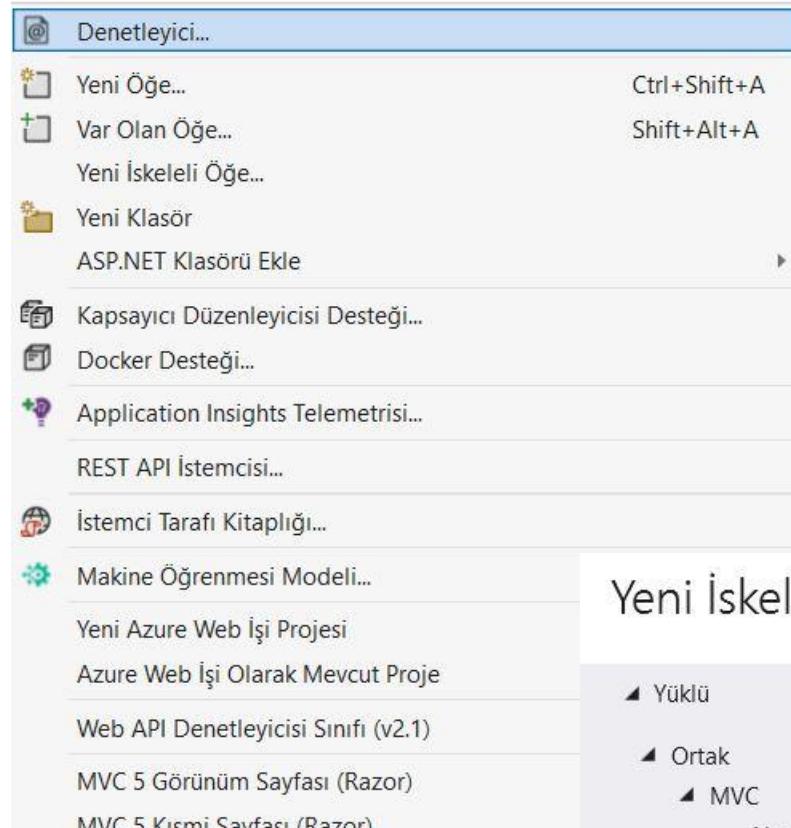
```
@using denemeonikinciders.ViewModels.Home
@model HomePageViewModel
```

NOT : Bu arada yaptığımız değişiklikler hemen algılanmayabilir. Bazen Rebuild ve clean yapmakta değişikliklerin algılanmasında yardımcı olmayı bilir. Bu hususta çözümü kapatıp tekrar açmak probleme muhtemelen çözüm üretecektir.

Fakat üzerinde çalıştığımız homepage.cshtml'de Adres ve Kişi nesneleri üzerinden veri tabanımızda ekleme ve güncelleme işlemleri yapacağımız için View ve Controller yapılarını oluşturmak zorundayız. Şimdi homepage.cshtml bu şekilde kalsın.

```
homepage.cshtml*  ✘ X
1 @using denemeonikinciders.ViewModels.Home
2 @model HomePageViewModel
3 @{
4     Layout = null;
5 }
6
7 <!DOCTYPE html>
8
9 <html>
10 <head>
11     <meta name="viewport" content="width=device-width" />
12
13     <link href("~/Content/bootstrap.min.css" rel="stylesheet" />
14     <link href("~/Content/Site.css" rel="stylesheet" />
15
16     <script src "~/Scripts/jquery-3.7.1.min.js"></script>
17     <script src "~/Scripts/bootstrap.min.js"></script>
18     <script src "~/Scripts/modernizr-2.8.3.js"></script>
19
20     <title>homepage</title>
21
22 </head>
```

HomeController'ı  
oluşturmuştu. Şimdi  
KisiController ve  
AdresController'ı  
oluşturalım.



## Yeni İskeleli Öğe Ekle

- ▲ Yüklü
- ▲ Ortak
  - ▲ MVC
    - Alan
    - Denetleyici
    - Göster
    - Web API'si



Entity Framework kullanan, görünümleri olan MVC 5 Denetleyici



MVC 5 Denetleyici - Boş



Okuma/yazma eylemleri olan MVC 5 Denetleyici

**MVC 5 Denetleyici - Boş**  
şunun tarafından Microsoft  
v5.0.0.0

Boş bir MVC denetleyici.

Kimlik: MvcControllerEmptyScaffolder

Denetleyici Ekle

Denetleyici adı:

Denetleyici Ekle

Denetleyici adı:



Dr. Öğl.

Fatih AKALIN

```
KisiController.cs + X
denemeonikinciders
denemeonikinciders.Controllers.K
1 using denemeonikinciders.Models;
2 using denemeonikinciders.Models.Managers;
3 using System;
4 using System.Collections.Generic;
5 using System.Linq;
6 using System.Web;
7 using System.Web.Mvc;
8 namespace denemeonikinciders.Controllers
9 {
10    public class KisiController : Controller
11    {
12        // GET: Kisi
13        // GET: Kisi
14        public ActionResult Yeni()
15        {
16            return View();
17        }
18        [HttpPost]
19        public ActionResult Yeni(Kisiler kisi)
20        {
21            DatabaseContext db = new DatabaseContext();
22            db.Kisiler.Add(kisi);
23
24            int sonuc = db.SaveChanges();
25            if (sonuc > 0)
26            {
27                ViewBag.Result = "Kişi kaydedilmiştir.";
28                ViewBag.Status = "success";
29            }
29            else
30            {
31                ViewBag.Result = "Kişi kaydedilememiştir.";
32                ViewBag.Status = "danger";
33            }
34            return View();
35        }
36    }
37}
```

```
KisiController.cs + X
denemeonikinciders
denemeonikinciders.Controllers.K
37
38    0 başvuru
39    public ActionResult Duzenle(int? kisiid)
40    {
41        Kisiler kisi = null;
42        if (kisiid != null)
43        {
44            DatabaseContext db = new DatabaseContext();
45            kisi = db.Kisiler.Where(x => x.ID == kisiid).FirstOrDefault();
46        }
47        return View(kisi);
48    }
49    [HttpPost]
50    0 başvuru
51    public ActionResult Duzenle(Kisiler model, int? kisiid)
52    {
53        DatabaseContext db = new DatabaseContext();
54        Kisiler kisi = db.Kisiler.Where(x => x.ID == kisiid).FirstOrDefault();
55        if (kisi != null)
56        {
57            kisi.Ad = model.Ad;
58            kisi.Soyad = model.Soyad;
59            kisi.Yas = model.Yas;
60
61            int sonuc = db.SaveChanges();
62            if (sonuc > 0)
63            {
64                ViewBag.Result = "Kişi güncellenmiştir.";
65                ViewBag.Status = "success";
66            }
67            else
68            {
69                ViewBag.Result = "Kişi güncellenmemiştir.";
70                ViewBag.Status = "danger";
71            }
72        }
73    }
74}
75}
```

## AdresController.cs

denemeonikinciders

```
1  using denemeonikinciders.Models;
2  using denemeonikinciders.Models.Managers;
3  using System;
4  using System.Collections.Generic;
5  using System.Linq;
6  using System.Web;
7  using System.Web.Mvc;
8
9  namespace denemeonikinciders.Controllers
10 {
11     public class AdresController : Controller
12     {
13         // GET: Adres
14         public ActionResult Yeni()
15         {
16             DatabaseContext db = new DatabaseContext();
17
18             List<SelectListItem> kisilerList = (from kisi in db.Kisiler.ToList()
19                                                 select new SelectListItem() //SelectListItem VE TOLIST İLE LİSTE TÜRÜNDE DÖNSÜN DİYORUZ
20                                                 {
21                                                     Text = kisi.Ad + " " + kisi.Soyad,
22                                                     Value = kisi.ID.ToString()
23                                                 }).ToList();
24
25
26             TempData["kisiler"] = kisilerList;//Kişiler listemizi viewbag ile sayfaya göndereceğiz.
27             ViewBag.kisiler = kisilerList;
28             return View();
29         }
30     }
```

```
[HttpPost]
0 başvuru
public ActionResult Yeni(Adresler adres)
{
    DatabaseContext db = new DatabaseContext();
    Kisiler kisi = db.Kisiler.Where(x => x.ID == adres.Kisi.ID).FirstOrDefault();
    if (kisi != null)
    {
        adres.Kisi = kisi;
        db.Adresler.Add(adres);
        int sonuc = db.SaveChanges();
        if (sonuc > 0)
        {
            ViewBag.Result = "Adres kaydedilmiştir.";
            ViewBag.Status = "success";
        }
        else
        {
            ViewBag.Result = "Adres kaydedilememiştir.";
            ViewBag.Status = "danger";
        }
    }
    ViewBag.kisiler = TempData["kisiler"];
    return View();
}
```

**Dr. Öğr. Üyesi: ALIN**

```
0 başvuru
public ActionResult Duzenle(int? adresid)
{
    Adresler adres = null;
    if (adresid != null)
    {
        DatabaseContext db = new DatabaseContext();

        List<SelectListItem> kisilerList = (from kisi in db.Kisiler.ToList()
                                              select new SelectListItem() //SelectListItem VE TOLIST İLE LİSTE TÜRÜNDE DÖNSÜN DİYORUZ
                                              {
                                                  Text = kisi.Ad + " " + kisi.Soyad,
                                                  Value = kisi.ID.ToString()
                                              }).ToList();

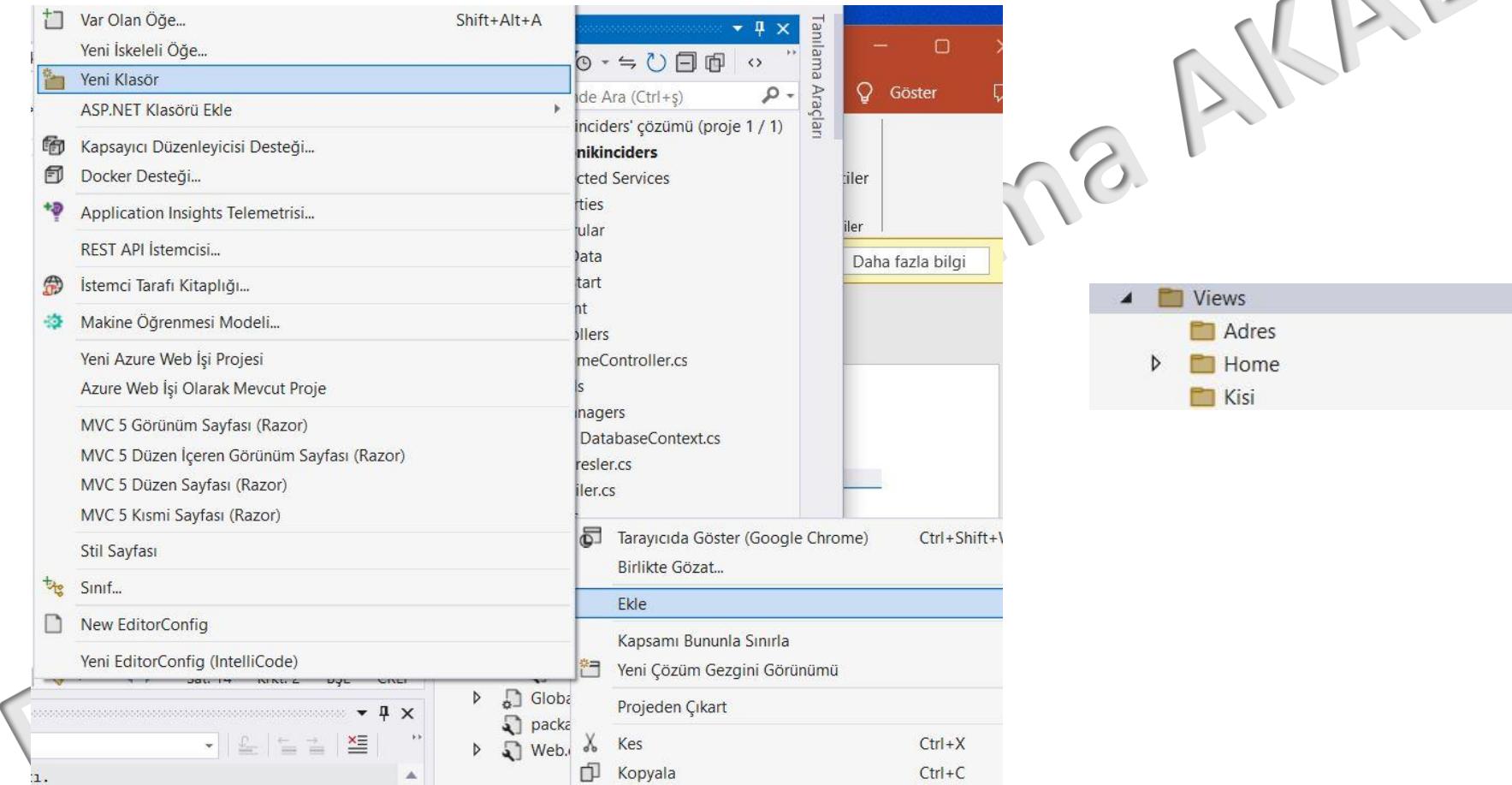
        TempData["kisiler"] = kisilerList;//Kişiler listemizi viewbag ile sayfaya göndereceğiz.
        ViewBag.kisiler = kisilerList;

        adres = db.Adresler.Where(x => x.ID == adresid).FirstOrDefault();
    }

    return View(adres);
}

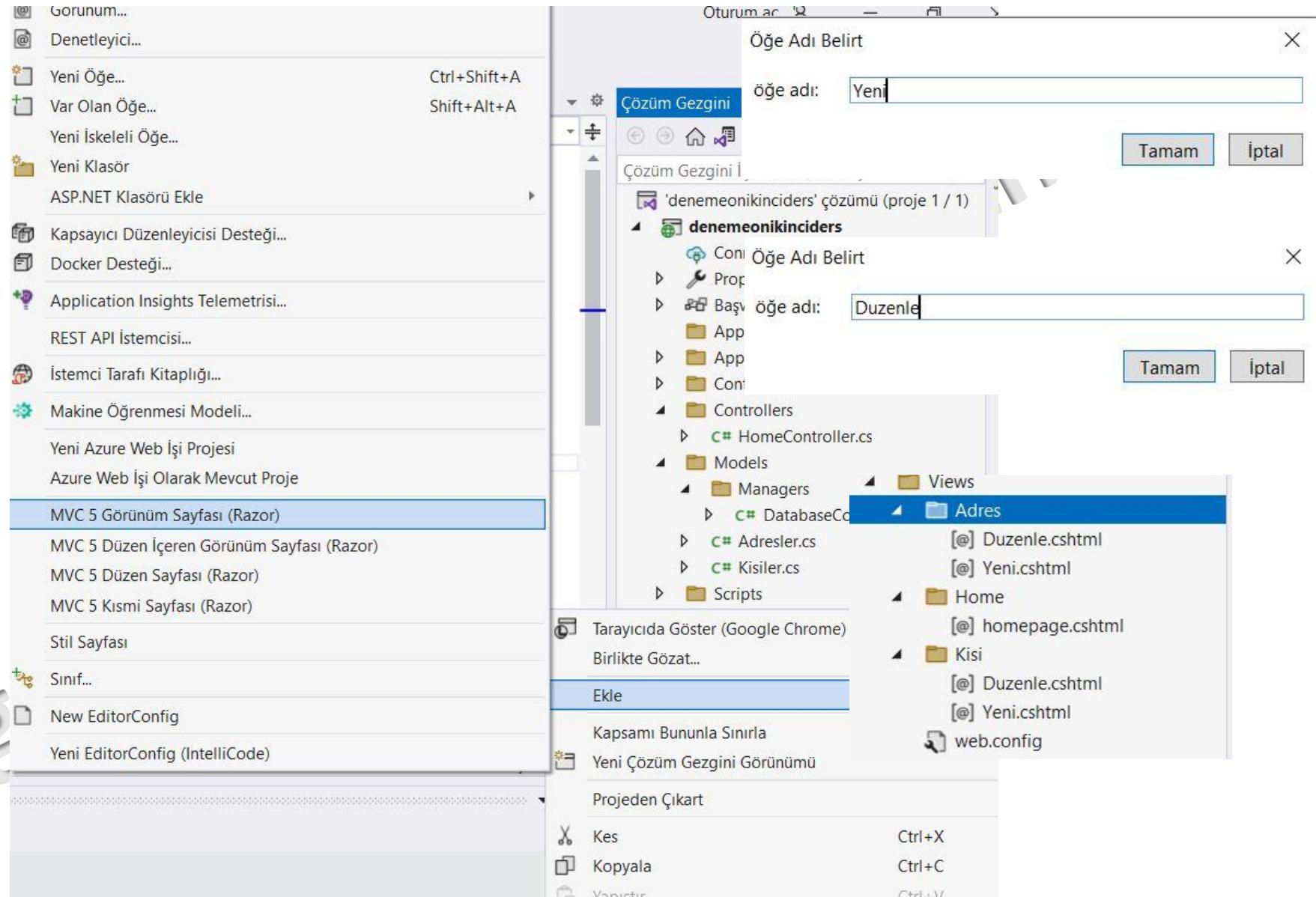
[HttpPost]
0 başvuru
public ActionResult Duzenle(Adresler model, int? adresid)
{
    DatabaseContext db = new DatabaseContext();
    Kisiler kisi = db.Kisiler.Where(x => x.ID == model.Kisi.ID).FirstOrDefault();
    Adresler adres = db.Adresler.Where(x => x.ID == adresid).FirstOrDefault();
    if (kisi != null)
    {
        adres.Kisi = kisi;
        adres.AdresTanim = model.AdresTanim;
        int sonuc = db.SaveChanges();
        if (sonuc > 0)
        {
            ViewBag.Result = "Adres güncellenmiştir.";
            ViewBag.Status = "success";
        }
        else
        {
            ViewBag.Result = "Adres güncellenmemiştir.";
            ViewBag.Status = "danger";
        }
    }
    ViewBag.kisiler = TempData["kisiler"];
    return View();
}
```

# View->Ekle->Yeni Klasör->Adres/Kisi



# Kisi klasörünün altında Duzenle.cshtml ve Yeni.cshtml isimli iki ayrı

Dr. Öğ



# Kisi-Yeni.cshtml->Modelimizi ekledik ve Ardından düzenledik.



```
Yeni.cshtml ✎ X
1  @model denemeonikinciders.Models.Kisiler
2  @{
3      Layout = null;
4  }
5
6  <!DOCTYPE html>
7
8  <html>
9  <head> |
10 <link href="~/Content/bootstrap.min.css" rel="stylesheet" />
11 <link href="~/Content/Site.css" rel="stylesheet" />
12
13 <script src="~/Scripts/jquery-3.7.1.min.js"></script>
14 <script src="~/Scripts/bootstrap.min.js"></script>
15 <script src="~/Scripts/modernizr-2.8.3.js"></script>
16
17 <h2 style="margin:50px">Yeni Kişi</h2>
18 <meta name="viewport" content="width=device-width" />
19 <title></title>
20 </head>
21 <body>
```

```
Yeni.cshtml ✎ X
20  </head>
21  <body>
22      @Html.ActionLink("Home Page", "homepage", "Home", new { area = "" }, new { @class = "navbar-brand m-5" })
23      @using (Html.BeginForm())
24  {
25          <div class="input-group m-5">
26              @*input kontrolü span kontrolüne id üzerinden bağlanmıştır aşağıda*@
27              <div class="input-group-prepend">
28                  <span class="input-group-text" id="txtad">Ad</span>
29                  @Html.TextBoxFor(x => x.Ad, new { @class = "form-control", placeholder = "Kişi Adı", aria_label = "Username", aria_describedby = "txtad" })
30              </div>
31          </div>
32          <div class="input-group m-5">
33              @*input kontrolü span kontrolüne id üzerinden bağlanmıştır aşağıda*@
34              <div class="input-group-prepend">
35                  <span class="input-group-text" id="txtsoyad">Soyad</span>
36                  @Html.TextBoxFor(x => x.Soyad, new { @class = "form-control", placeholder = "Kişi SoyAdı", aria_label = "Username", aria_describedby = "txtsoyad" })
37              </div>
38          </div>
39          <div class="input-group m-5">
40              @*input kontrolü span kontrolüne id üzerinden bağlanmıştır aşağıda*@
41              <div class="input-group-prepend">
42                  <span class="input-group-text" id="txtyas">Yaş</span>
43                  @Html.TextBoxFor(x => x.Yas, new { @class = "form-control", placeholder = "Kişi Yaşı", aria_label = "Username", aria_describedby = "txtyas" })
44              </div>
45          </div>
46          <div class="input-group m-5">
47              <input type="submit" value="Kaydet" class="btn btn-success" />
48          </div>
49          <br />
50          <div class="input-group">
51              @if (ViewBag.Result != null)
52              {
53                  <div class="alert alert-@ViewBag.Status m-5" role="alert">
54                      @ViewBag.Result
55                  </div>
56              }
57          </div>
58      }
59  </body>
60  </html>
```

# Kisi-Duzenle.cshtml->Modelimizi ekledik ve Ardından düzenledik.

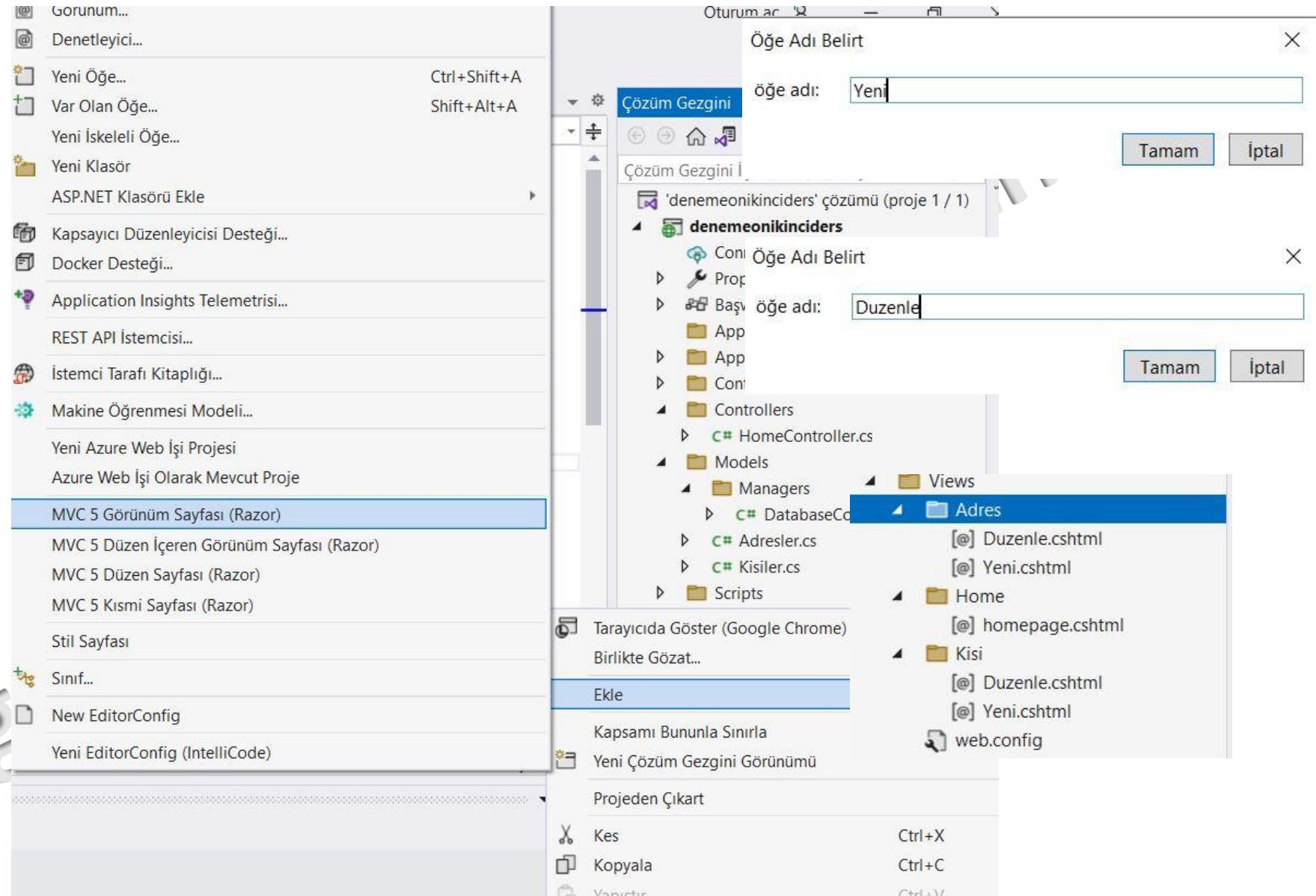


```
Duzenle.cshtml ✘ X
1  @model denemeonikinciders.Models.Kisiler
2  @{
3      Layout = null;
4  }
5
6  <!DOCTYPE html>
7
8  <html>
9  <head>
10     <meta name="viewport" content="width=device-width" />
11
12     <link href="~/Content/bootstrap.min.css" rel="stylesheet" />
13     <link href="~/Content/Site.css" rel="stylesheet" />
14
15     <script src="~/Scripts/jquery-3.7.1.min.js"></script>
16     <script src="~/Scripts/bootstrap.min.js"></script>
17     <script src="~/Scripts/modernizr-2.8.3.js"></script>
18
19     <h2>Kişi Düzenle</h2>
20     <meta name="viewport" content="width=device-width" />
21     <title></title>
22 </head>
```

```
Duzenle.cshtml X
22  </head>
23  <body>
24      @Html.ActionLink("Home Page", "homepage", "Home", new { area = "" }, new { @class = "navbar-brand m-5" })
25      @using (Html.BeginForm())
26      {
27          <div class="input-group m-5">
28              @*input kontrolü span kontrolüne id üzerinden bağlanmıştır aşağıda*@
29              <div class="input-group-prepend">
30                  <span class="input-group-text" id="txtad">Ad</span>
31                  @Html.TextBoxFor(x => x.Ad, new { @class = "form-control", placeholder = "Kişi Adı", aria_label = "Username", aria_describedby = "txtad" })
32              </div>
33          </div>
34          <div class="input-group m-5">
35              @*input kontrolü span kontrolüne id üzerinden bağlanmıştır aşağıda*@
36              <div class="input-group-prepend">
37                  <span class="input-group-text" id="txtsoyad">Soyad</span>
38                  @Html.TextBoxFor(x => x.Soyad, new { @class = "form-control", placeholder = "Kişi SoyAdı", aria_label = "Username", aria_describedby = "txtsoyad" })
39              </div>
40          </div>
41          <div class="input-group m-5">
42              @*input kontrolü span kontrolüne id üzerinden bağlanmıştır aşağıda*@
43              <div class="input-group-prepend">
44                  <span class="input-group-text" id="txtyas">Yaş</span>
45                  @Html.TextBoxFor(x => x.Yas, new { @class = "form-control", placeholder = "Kişi Yaşı", aria_label = "Username", aria_describedby = "txtyas" })
46              </div>
47          </div>
48          <div class="input-group m-5">
49              <input type="submit" value="Kaydet" class="btn btn-success" />
50          </div>
51          <br />
52          <div class="input-group">
53              @if (ViewBag.Result != null)
54              {
55                  <div class="alert alert-@ViewBag.Status m-5" role="alert">
56                      @ViewBag.Result
57                  </div>
58              }
59          </div>
60      }
61  </body>
62  </html>
```

Adres  
klasörünün  
altında  
Duzenle.cshtml  
ve Yeni.cshtml  
isimli iki ayrı

Dr. Öğ



Adres-Yeni.cshtml->Modelimizi ekledik ve  
Ardından düzenledik.

```
Yeni.cshtml X
1 @model denemeonikinciders.Models.Adresler
2 @{
3     Layout = null;
4 }
5
6 <!DOCTYPE html>
7
8 <html>
9 <head>
10    <link href="~/Content/bootstrap.min.css" rel="stylesheet" />
11    <link href="~/Content/Site.css" rel="stylesheet" />
12
13    <script src="~/Scripts/jquery-3.7.1.min.js"></script>
14    <script src="~/Scripts/bootstrap.min.js"></script>
15    <script src="~/Scripts/modernizr-2.8.3.js"></script>
16
17    <h2 style="margin:50px">Yeni Adres</h2>
18    <meta name="viewport" content="width=device-width" />
19    <title></title>
20 </head>
```

Dr. Öğ

sa AKALIN

```
Veni.cshtml ✘ X
19     <title></title>
20   </head>
21   <body>
22     @Html.ActionLink("Home Page", "homepage", "Home", new { area = "" }, new { @class = "navbar-brand m-5" })
23
24   @using (Html.BeginForm())
25   {
26     <div class="input-group m-5">
27       @*input kontrolü span kontrolüne id üzerinden bağlanmıştır aşağıda*@
28       <div class="input-group-prepend">
29         <span class="input-group-text" id="txtadres">Adres</span>
30         @Html.TextAreaFor(x => x.AdresTanim, new { @class = "form-control", placeholder = "Adres Tanımınız", aria_label = "Username", aria_describedby = "txtadres", cols = "30", rows = "10" })
31       </div>
32     </div>
33
34     <div class="input-group m-5">
35       @*input kontrolü span kontrolüne id üzerinden bağlanmıştır aşağıda*@
36       <div class="input-group-prepend">
37         <span class="input-group-text" id="ddlkisi">Soyad</span>
38         @Html.DropDownListFor(x => x.Kisi.ID, (List<SelectListItem>)ViewBag.kisiler, new { @class = "form-control", aria_describedby = "ddlkisi" })
39       @*DEDİKKİ yukarıdaki satır ile listeden seçilen kişinin ID'sini modelimin içindeki Kisnin içindeki Id'yebağla'*@
40       @*İçine dolduracağım liste ama obje olarak gördüğü için tip dönüşümü yap ViewBag ile gelen bilgileri*@
41       </div>
42     </div>
43
44
45     <div class="input-group m-5">
46       <input type="submit" value="Kaydet" class="btn btn-success" />
47     </div>
48     <br />
49     <div class="input-group">
50       @if (ViewBag.Result != null)
51       {
52         <div class="alert alert-@ViewBag.Status m-5" role="alert">
53           @ViewBag.Result
54         </div>
55       }
56
57     </div>
58   }
59   </body>
60   </html>
```

Dr.

Adres-Duzenle.cshtml->Modelimizi ekledik ve  
Ardından düzenledik.

```
Duzenle.cshtml X Yeni.cshtml
1 @model denemeonikinciders.Models.Adresler
2 @{
3     Layout = null;
4 }
5
6 <!DOCTYPE html>
7
8 <html>
9 <head>
10    <link href="~/Content/bootstrap.min.css" rel="stylesheet" />
11    <link href="~/Content/Site.css" rel="stylesheet" />
12
13    <script src="~/Scripts/jquery-3.7.1.min.js"></script>
14    <script src="~/Scripts/bootstrap.min.js"></script>
15    <script src="~/Scripts/modernizr-2.8.3.js"></script>
16
17    <h2 style="margin:50px"> Adres Duzenle</h2>
18    <meta name="viewport" content="width=device-width" />
19    <title></title>
20 </head>
```

Duzenle.cshtml X Yeni.cshtml

```
19     <title></title>
20   </head>
21   <body>
22
23     @Html.ActionLink("Home Page", "homepage", "Home", new { area = "" }, new { @class = "navbar-brand m-5" })
24
25   @using (Html.BeginForm())
26   {
27     <div class="input-group m-5">
28       /*input kontrolü span kontrolüne id üzerinden bağlanmıştır aşağıda*/
29       <div class="input-group-prepend">
30         <span class="input-group-text" id="txtadres">Adres</span>
31         @Html.TextAreaFor(x => x.AdresTanim, new { @class = "form-control", placeholder = "Adres Tanımınız", aria_label = "Username", aria_describedby = "txtadres", cols = "30", rows = "10" })
32       </div>
33     </div>
34
35     <div class="input-group m-5">
36       /*input kontrolü span kontrolüne id üzerinden bağlanmıştır aşağıda*/
37       <div class="input-group-prepend">
38         <span class="input-group-text" id="ddlkisi">Soyad</span>
39         @Html.DropDownListFor(x => x.Kisi.ID, (List<SelectListItem>)ViewBag.Kisiler, new { @class = "form-control", aria_describedby = "ddlkisi" })
40       /*DEDİKKİ yukarıdaki satır ile listeden seçilen kişinin ID'sini modelimin içindeki Kisnin içindeki Id'yebağla*/
41       /*İçine dolduracağım liste ama obje olarak gördüğü için tip dönüşümü yap ViewBag ile gelen bilgileri*/
42     </div>
43   </div>
44   <div class="input-group m-5">
45     <input type="submit" value="Kaydet" class="btn btn-success" />
46   <br />
47   <div class="input-group">
48     @if (ViewBag.Result != null)
49     {
50       <div class="alert alert-@ViewBag.Status m-5" role="alert">
51         @ViewBag.Result
52       </div>
53     }
54   </div>
55 }
56 </body>
57 </html>
```

Dr. Ü

# homepage.cshtml dosyasını yazıyorum

```
homepage.cshtml  ✎ X Duzenle.cshtml      Yeni.cshtml
1  @using denemeonikinciders.ViewModels.Home
2  @model HomePageViewModel
3  @{
4      Layout = null;
5  }
6
7  <!DOCTYPE html>
8
9  <html>
10 <head>
11     <meta name="viewport" content="width=device-width" />
12
13     <link href("~/Content/bootstrap.min.css" rel="stylesheet" />
14     <link href("~/Content/Site.css" rel="stylesheet" />
15
16     <script src "~/Scripts/jquery-3.7.1.min.js"></script>
17     <script src "~/Scripts/bootstrap.min.js"></script>
18     <script src "~/Scripts/modernizr-2.8.3.js"></script>
19
20
21     <title>homepage</title>
22 </head>
```

```
<body>

    <div class="container-fluid">
        <div class="col-md-12">
            <h2>homepage</h2>
            <hr />
        </div>
        <div class="row">
            <div class="col-md-6">

                <div>
                    @Html.ActionLink("Yeni Kişi", "Yeni", "Kisi", null, new
{
    @class = "btn btn-primary"
})
                    <br /><br />
                </div>

                <h4>Kisiler</h4>
                <hr />
                <p>Kişi ad - soyad ve yas</p>
                <ul class="list-group">
                    @foreach (var kisi in Model.Kisiler)
                    {

                        <li class="list-group-item">
                            @Html.ActionLink("Duzenle", "Duzenle", "Kisi", new { kisiid = kisi.ID }, new { @class = "btn btn-warning btn-xs" })

                            <span class="glyphicon glyphicon-user red"></span> @(kisi.Ad + " " + kisi.Soyad)
                            <b>@(kisi.Yas)</b>
                        </li>
                    }
                </ul>
            </div>
        </div>
    </div>
```

```
<div class="col-md-6">
    <div>
        @Html.ActionLink("Yeni Adres", "Yeni", "Adres", null, new
        {
            @class = "btn btn-primary"
        })
        <br /><br />
    </div>

    <h4>Adresler</h4>
    <hr />
    <p>Adres tanım ve adressteki Kişi ad ve soyad</p>
    <ul class="list-group">
        @foreach (var adres in Model.Adresler)
        {
            <li class="list-group-item">
                @Html.ActionLink("Duzenle", "Duzenle", "Adres", new { adresid = adres.ID }, new { @class = "btn btn-warning btn-xs" })

                <span class="glyphicon glyphicon-home"></span> @adres.AdresTanim
                <b>@(adres.Kisi.Ad + " " + adres.Kisi.Soyad)</b>
            </li>
        }
    </ul>
</div>
</div>

</body>
</html>
```

D1.

homepage

localhost:44385/Home/homepage

PAMUKKALE ÜNİVE... Gmail YouTube Tumor Detection U... ROI-Based Processi... Cross-validated, bin... Using MATLAB with... GEO DataSet Browser GEO Accession vie...

Kisiler

Kişi ad - soyad ve yaş

Duzenle	Nathan Silva <b>86</b>
Duzenle	Charles Rossi <b>79</b>
Duzenle	Gabriel Rossi <b>54</b>
Duzenle	Genesis Watson <b>35</b>
Duzenle	Kayden Riddick <b>48</b>
Duzenle	Madelyn Bradshaw <b>70</b>
Duzenle	Owen Summers <b>74</b>
Duzenle	Emily Seymour <b>85</b>
Duzenle	Jocelyn Little <b>31</b>
Duzenle	Aubrey Rossi <b>76</b>

Yeni Kişi

Adresler

Adres tanım ve adressteki Kişi ad ve soyad

Duzenle	818 Hilton Terrace <b>Nathan Silva</b>
Duzenle	89 McGrath Hill <b>Nathan Silva</b>
Duzenle	81 Wilson Square <b>Charles Rossi</b>
Duzenle	869 Nash Square <b>Gabriel Rossi</b>
Duzenle	622 Robinson Terrace <b>Genesis Watson</b>
Duzenle	655 Taylor Highland <b>Genesis Watson</b>
Duzenle	837 Bradshaw Grove <b>Kayden Riddick</b>
Duzenle	227 Schofield Park <b>Kayden Riddick</b>
Duzenle	284 Whitehead Close <b>Madelyn Bradshaw</b>
Duzenle	586 Mellor Crescent <b>Madelyn Bradshaw</b>
Duzenle	225 Akhtar Lane <b>Owen Summers</b>
Duzenle	675 Williams Lane <b>Owen Summers</b>
Duzenle	302 Burrows Rise <b>Emily Seymour</b>

Yeni Adres

AKALIN

Dr. U

localhost:44385/Kisi/Yeni

localhost:44385/Kisi/Yeni

PAMUKKALE ÜNİVE... Gmail YouTube

---

## Yeni Kişi

Home Page

Ad	Nathan
Kişi Adı	
Soyad	Silva
Kişi SoyAdı	
Yaş	86
Kişi Yası	

**Kaydet**

localhost:44385/Kisi/Duzenle?kisiid=1

localhost:44385/Kisi/Duzenle?kisiid=1

PAMUKKALE ÜNİVE... Gmail YouTube Tumor Detect

## Kişi Düzenle

Home Page

Ad	Nathan
Soyad	Silva
Yaş	86

**Kaydet**

VALIN

Dr. ÜÜY

The screenshot displays a web browser window with two tabs open, illustrating a comparison between a new address entry page and an edit address page.

**Top Tab (localhost:44385/Adres/Duzenle?adresid=1):**

- Address:** Adres Duzenle
- Address Input:** 818 Hilton Terrace
- Surname Input:** NathanSilva
- Save Button:** Kaydet

**Bottom Tab (localhost:44385/Adres/Yeni):**

- Address:** Adres
- Address Input:** Adres Taniminiz
- Surname:** Soyad
- Surname Input:** NathanSilva
- Save Button:** Kaydet

A large watermark reading "KALIN" is diagonally across the right side of the image.

# KAYNAKLAR

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

<https://www.entityframeworktutorial.net/what-is-entityframework.aspx>

<https://muratbaseren.wordpress.com/category/kitap-2/>

<https://www.yusufsezer.com.tr/entity-framework/>

Veysel Uğur Kızmaz, ASP.NET MVC5, Kodlab Yayınları

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