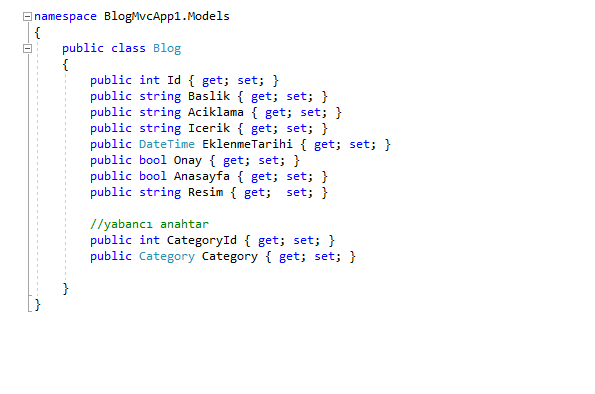
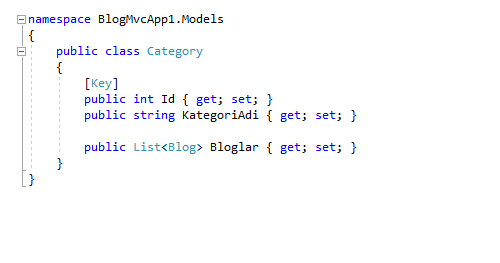
**BlogMvcApp1 ASP.NET PROJESİ**

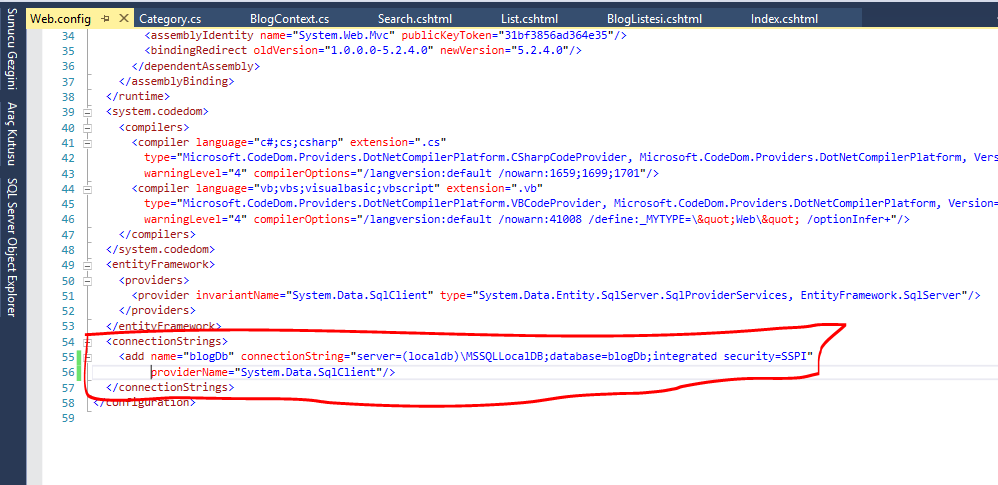
Bu ASP.NET MVC projesinde blog oluşturacağımız bir yapı kullandık. İlk olarak

Blog ve Category



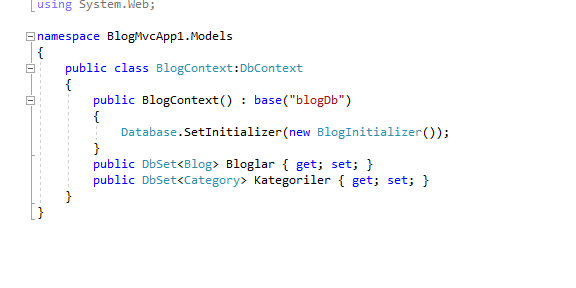
Adında iki tane sınıfımızı oluşturduk.

Webconfig dosyasında connection string ile database i istediğimiz şekilde oluşturduk.



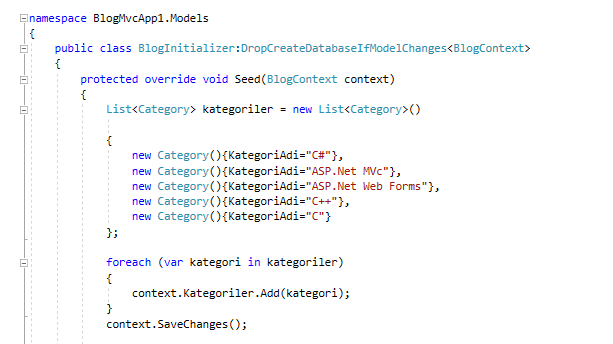
-Daha sonra Blogcontext sınıfında Base(“blogDb”) metodu ile database bağlantısını yaptık.Entityframework nuget paket yöneticisnden projemize ekliyoruz. Bu sayede daha basit db yönetimi sağlıyor.

Ve Db tablolarını oluşturduk.



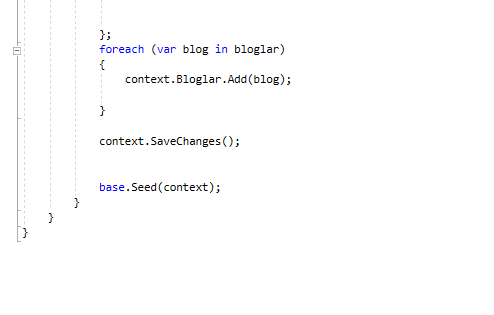
Database.SetInitializer(new BlogInitializer()); metodu ile Db’e rastgele oluşturduğumuz verilerimizi atıyoruz.

BlogInitializer sınıfında Database ‘in Model sınıflarında değşiklik olduğunda yeniden oluştur(DropCreateDatabaseIfModelChanges<BlogContext>) ve Seed metodu ile rastgele verilerimizi db ye ekliyoruz.Rastgele kategoriler verilerimizi db ye liste olarak atıyoruz. Daha sonra kategoriler verilerimizi foreach ile dolaşarak tek tek context yani db e ekliyoruz daha sonra değişiklikleri kaydediyoruz



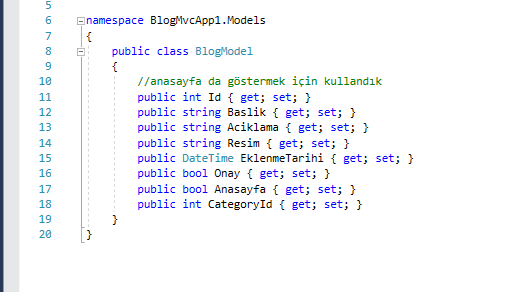
Aynı işlemi Blog verilerimiz için de tekrarlıyoruz.



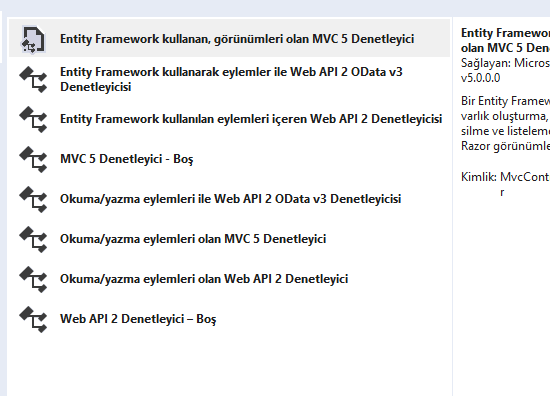


Değişiklikleri kaydettikten sonra base.Seed(context) metodu ile bütün değişiklikleri context ile db ye aktarıyoruz.

BlogModel i anasayfa da bloglarımızı göstermek için oluşturuyoruz.



Projemize Controller ekliyoruz . Controller eklerken ilk sçeneği seçiyoruz ve otomatik olarak hazır View leri kullanıyoruz.

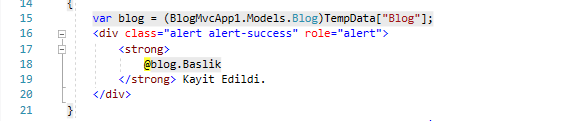


Oluşturulan Controllerla ilk olarak blog İndeximizi değiştiriyoruz.

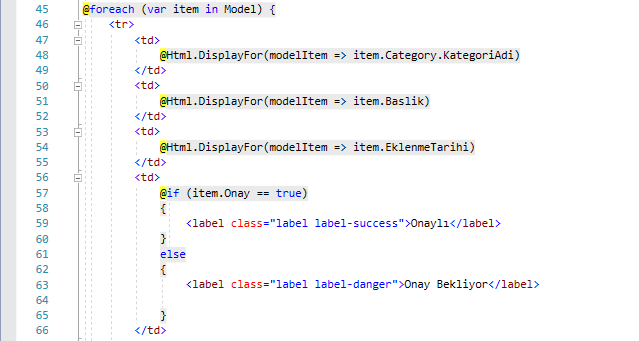


@model ile Modelimizin bizden beklediğ veri tipini belirtiyoruz.

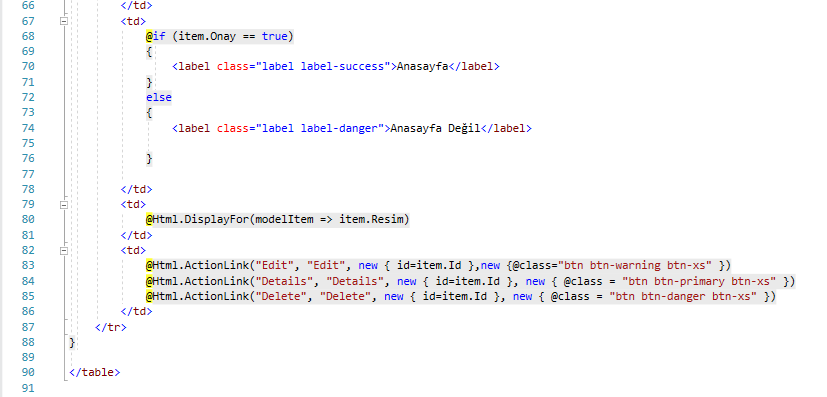
@Html.ActionLink(“Yeni Blog Ekle”,”Create”,null,new{@class=”btn btn-primary”}) metoduyla yeni bir blog oluşturma linki oluşturuyoruz ve daha sonra linkimizi buton yapıyoruz…

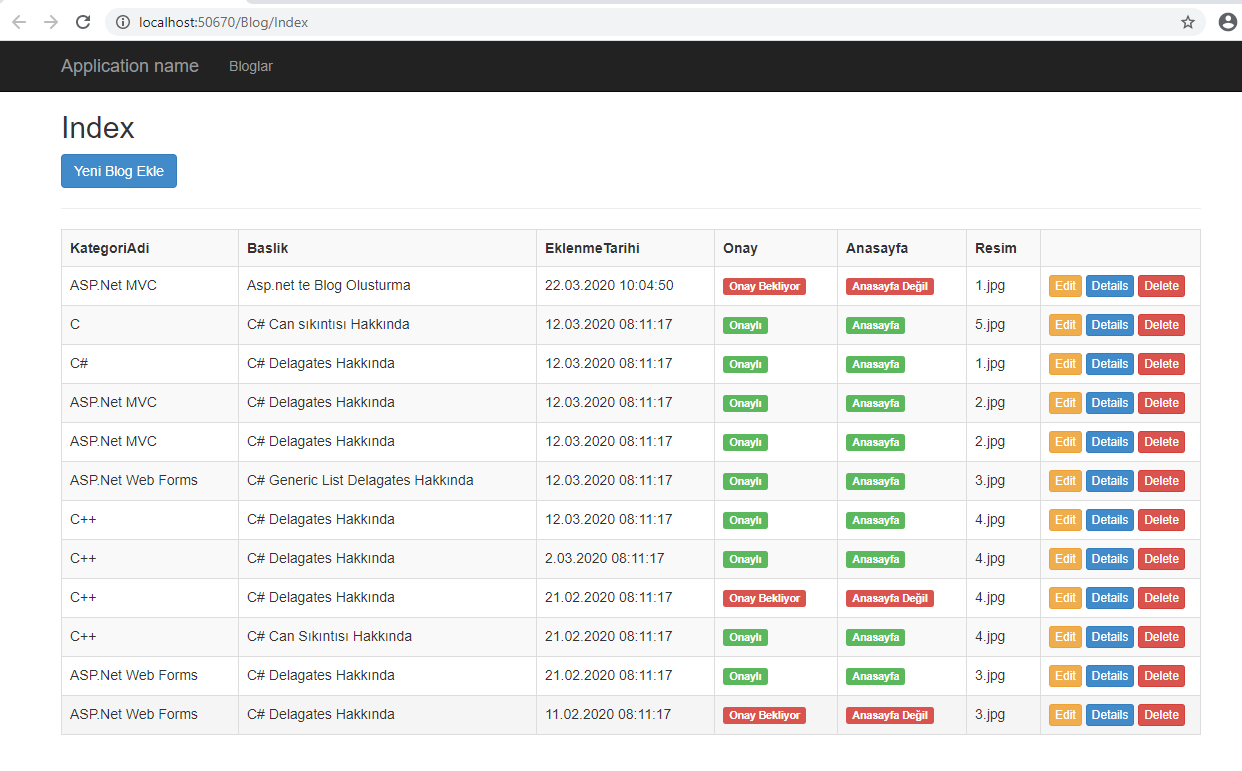


TempData yardımıyla yeni eklenen kaydı Blog/index sayfasında kaydın ismini gösteriyoruz. TempData ‘nın cinsini (BlogMvcApp1.Models.Blog) olarak cast ediyoruz.

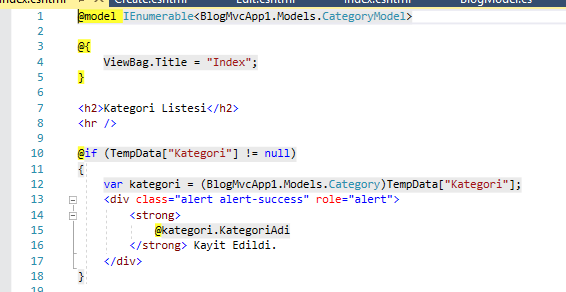


Foreach yardımıyla bloglarımızı sayfamıza yazdırıp onaylıları ve onaylı olmayanları kendi isteğimize göre düzenliyoruz.

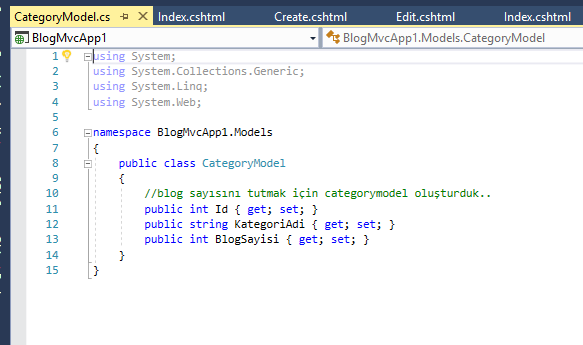


Gerekli işlemlerden sonra Blog/Index sayfamızın genel görünümü bu şekilde olmuştur.

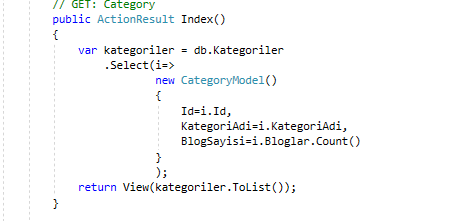
Daha Sonra Category sınıfımız için aynı gerekli işlemleri yapıyoruz fakat category/index te aynı zamanda blog sayısını da göstermek istiyoruz. Bu yüzden CategoryModel adında sınıf oluşturuyoruz bu sayede blog sayısını tutmayı amaçlıyoruz.



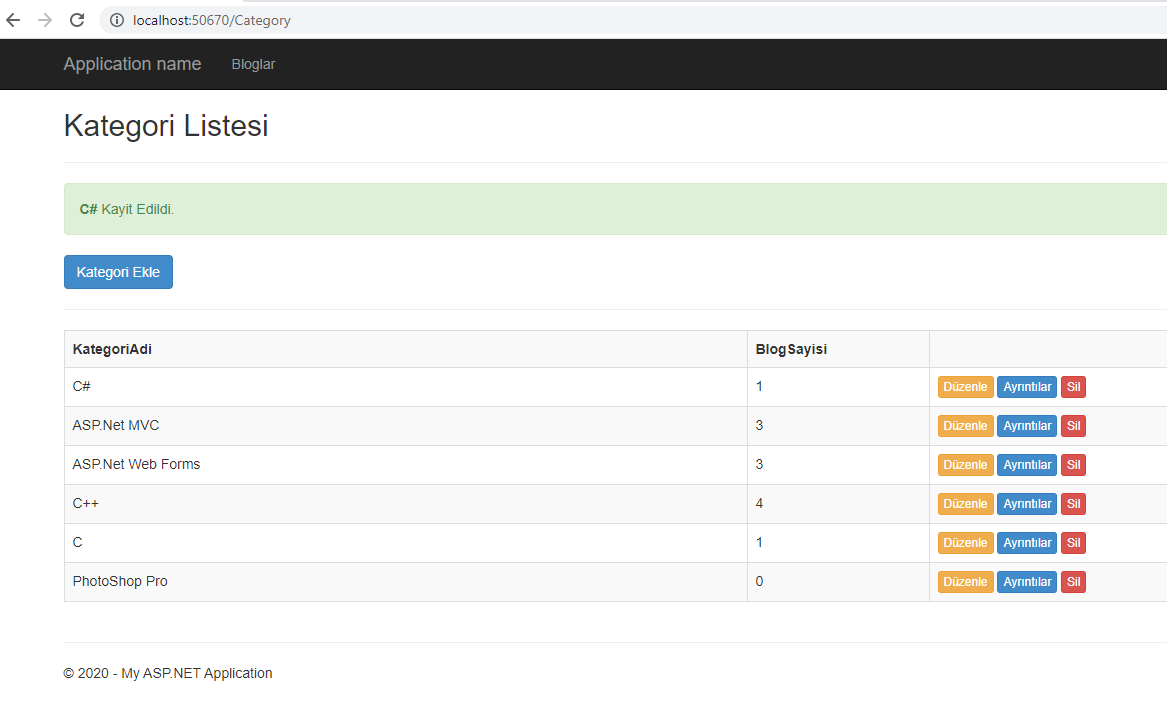
CategoryModel sınıfı



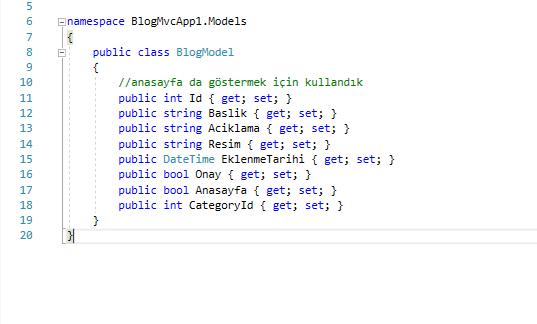
CategoryController



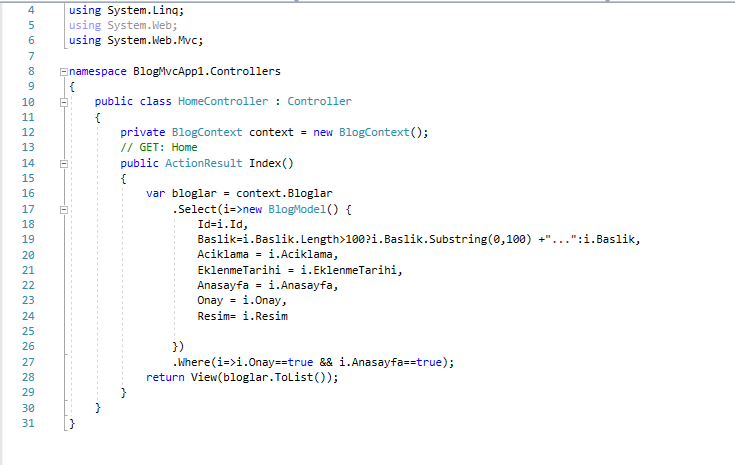
Son olarak Category/Index view imiz.



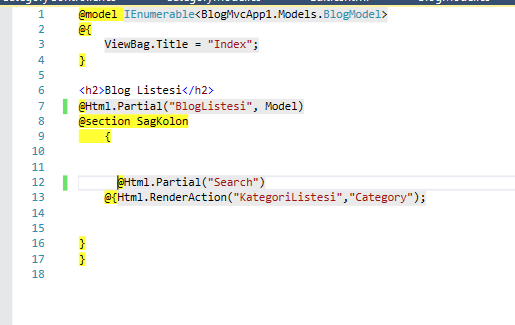
Anasayfada Blog Listesi için BlogModel adında yeni bir class oluşturuyoruz.



HomeController



Home Index



BlogListesi adlı PartialView de bloglarımızı foreach ile dönüp listeledik .Bu partial View imizi aynı zamanda KategoriListesi için de kullandık çünkü bloglarımızı hem id ye göre hemde aranankelimeye göre de listeliyebiliyoruz.



