GOLD ROSE BEBEK BAKICISI SİTESİ

MVC bebek bakıcısı uygulamamiza, sitemizin Ana sayfasina URL'leri işleyecek bir Controller sınıfı ekleyerek baslayacağız.

ASP.NET MVC'nin varsayilan adlandirma kurallarina uyup HomeController olarak adlandiracagiz.

Çözüm Gezgini'nde “Denetleyiciler” klasöründen ekle kismindan yapiyoruz.

Sitemizin Ana Sayfasını uygulayan basit bir HomeController ekledik. Simdi bebek bakıcısı sitemizin tarama islevini uygulamak için kullanacagimiz baska bir denetleyici ekliyoruz.

Magaza denetleyicimiz üç maddeyi destekleyecek sekilde olacak:

1)Bebek bakıcısı sitemizdeki bakıcıları listeleme

2)Şehirleri listeleme

3)Bakıcı tutulacak bebeğin yaş grubunu seçme

Bir görünüm sablonu kullanmak için, bir ActionResult döndürmek için HomeController kullandık.

Dizini yöntemini degistirecegiz View () döndürmesini saglayacagiz:

İçerisindeki index,browse ve details görünümlerini kullanacağımız StoreController ekliyoruz.

Şimdi projemize uygun bir Görünüm Şablonu ekleyecegiz. Bunu yapmak için metin imlecini

dizin işlemi yönteminin içine yerleştireceğiz, ardından sağ tıklayıp “Görünüm Ekle” yi seçeceğiz.

Bu, Görünüm Ekle iletişim kutusunu görüntüler.Çoğu web sitesinde birçok sayfa arasında paylaşılan içerik bulunur: gezinme, altbilgiler, logo resimleri, stil sayfası referansları vb.Razor görünüm motoru,Views,Shared klasöründe bizim için otomatik olarak oluşturulan \_Layout.cshtml adli bir sayfa kullanarak bunu yönetmeyi kolaylastirir.

Öncelikle magazamızdaki bakıcıları ve categorileri temsil etmek için bazi Model siniflari olusturacagiz. Bir tür sınıfı oluşturarak başlayalım.

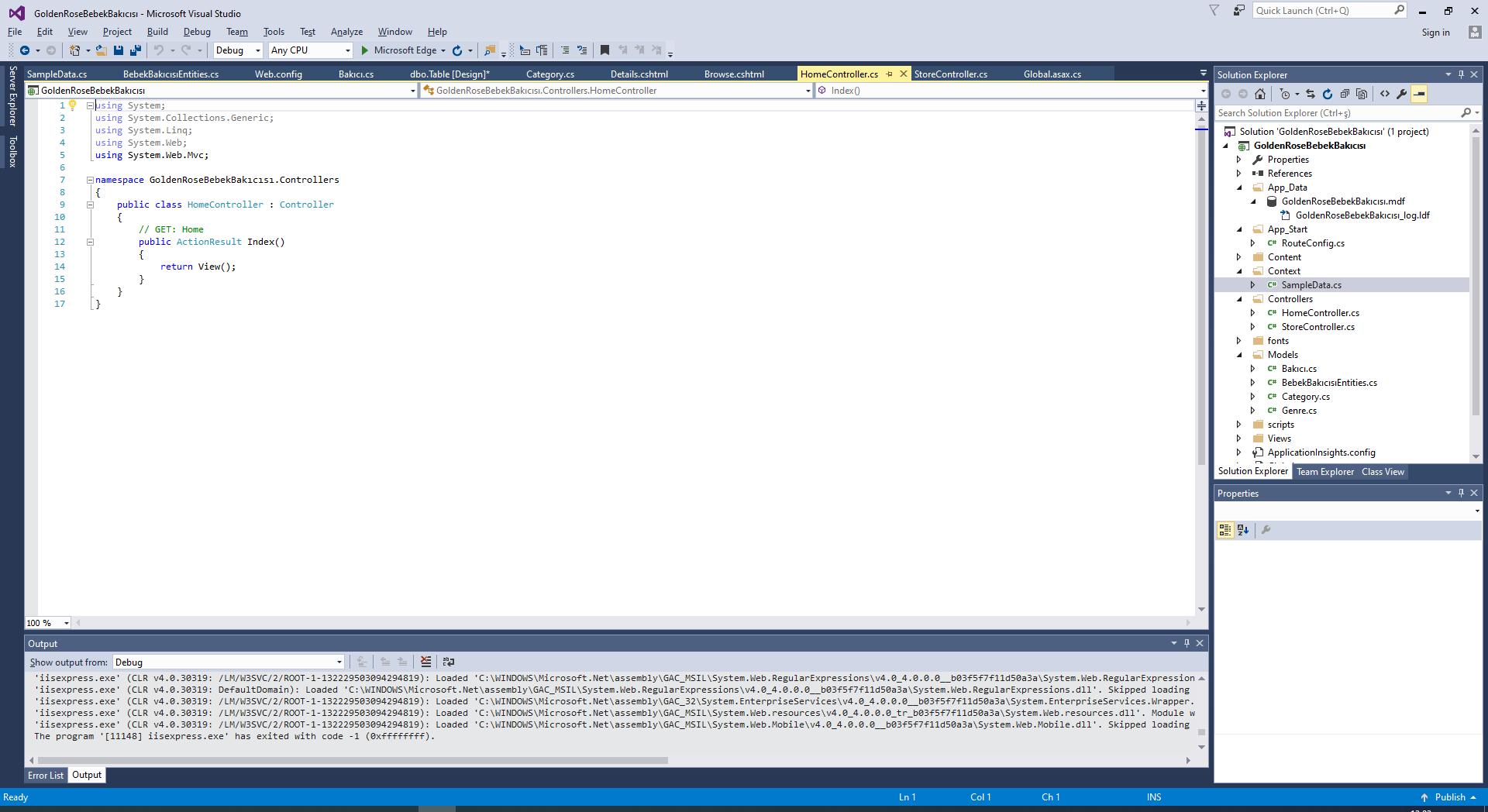
Projenizdeki “Modeller” klasörünü sag tıklayın, “Sınıf Ekle” seçeneğini seçin ve “Bakıcı.cs” dosyasını adlandırın.

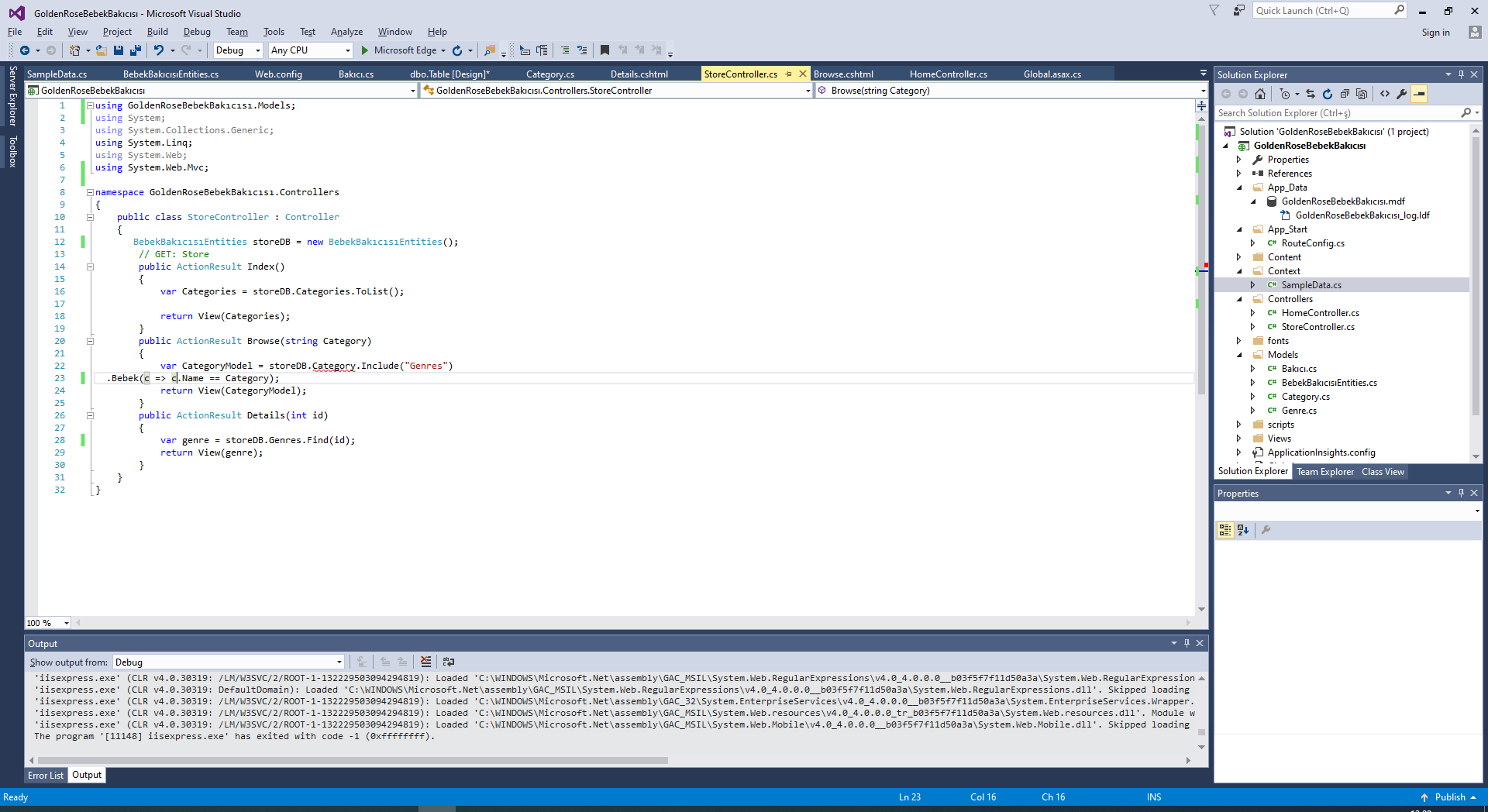
Veritabanını sorgulamak ve güncellemek için ASP.NET MVC 3 projelerinde bulunan Entity Framework (EF) destegini kullanacagiz.

EF, gelistiricilerin bir veritabaninda depolanan verileri nesne yönelimli bir sekilde sorgulamasini ve güncellemesini saglayan esnek bir nesne ilişkisel esleme (ORM) veri API'sidir.

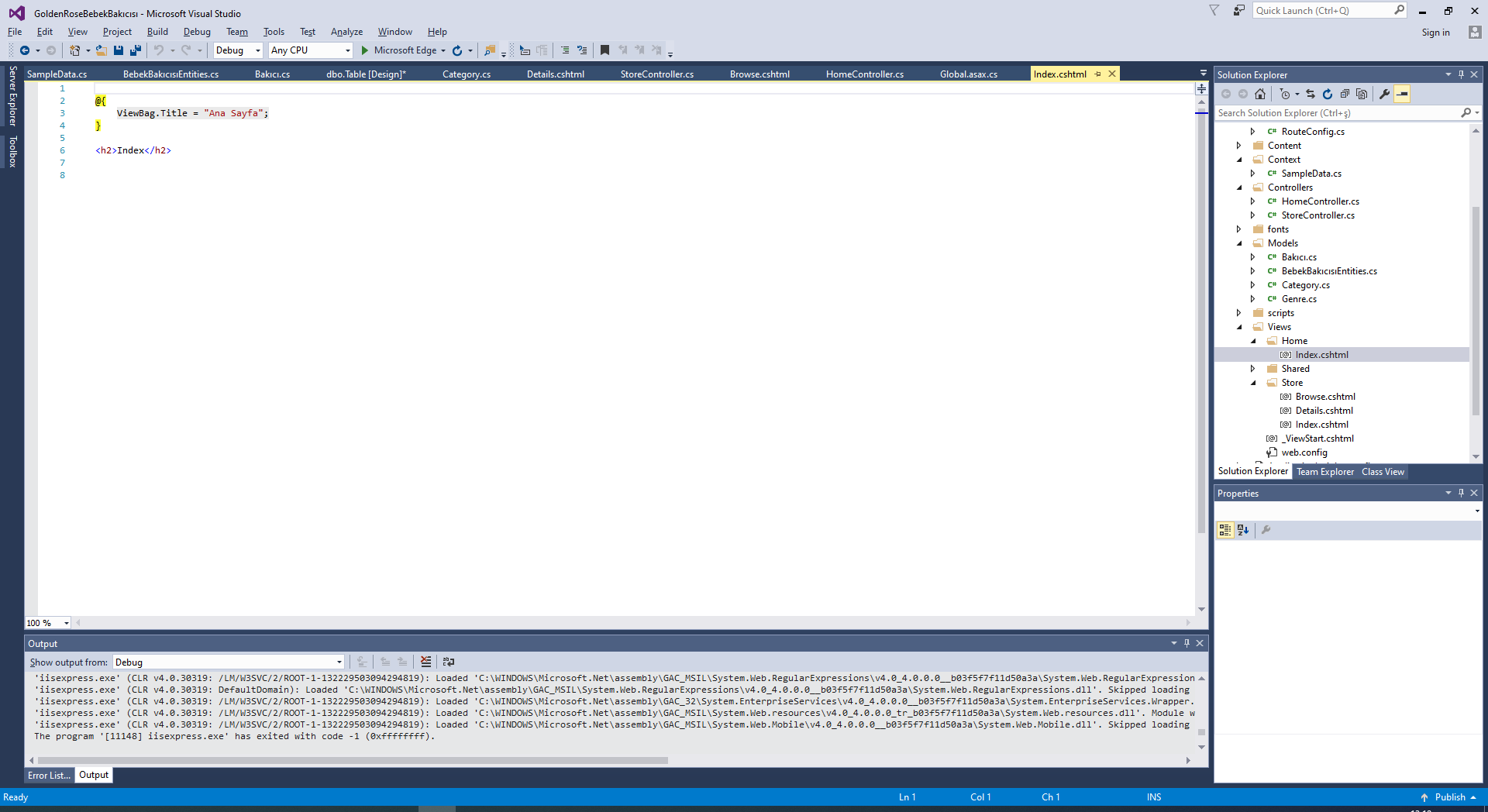
SQL Server Express veritabani dosyalarımızı tutmak için projemize bir App\_Data dizini ekleyecegiz.Entity Framework'ün veritabanimiza nasil baglanacagini bilmesi için web sitesinin yapilandirma dosyasina birkaç satir ekleyecegiz.

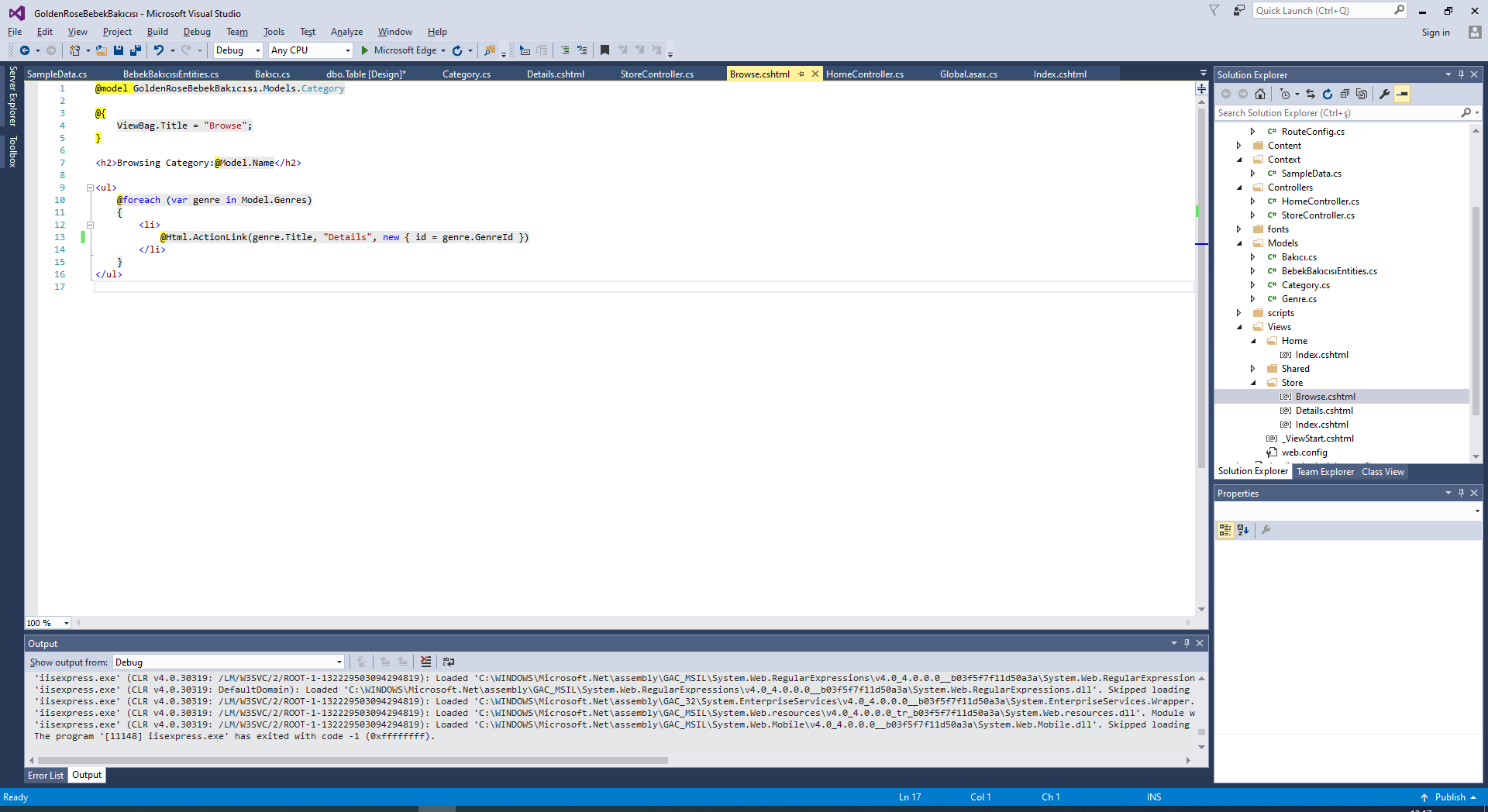
Şimdi StoreController'imizi güncelleyelim, böylece "kukla veriler" kullanmak yerine tüm bilgilerini sorgulamak için veritabanimizi çagirir.

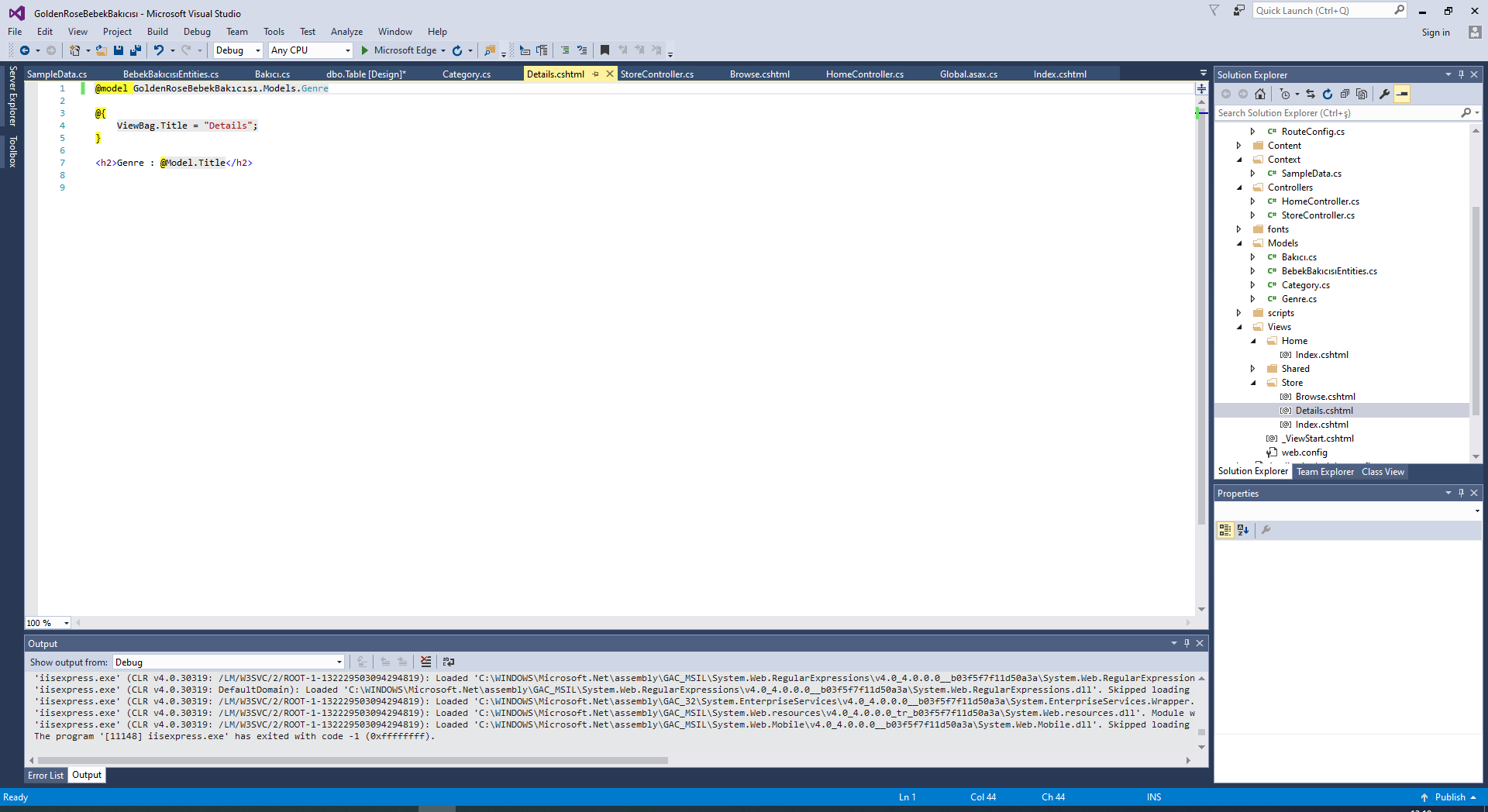


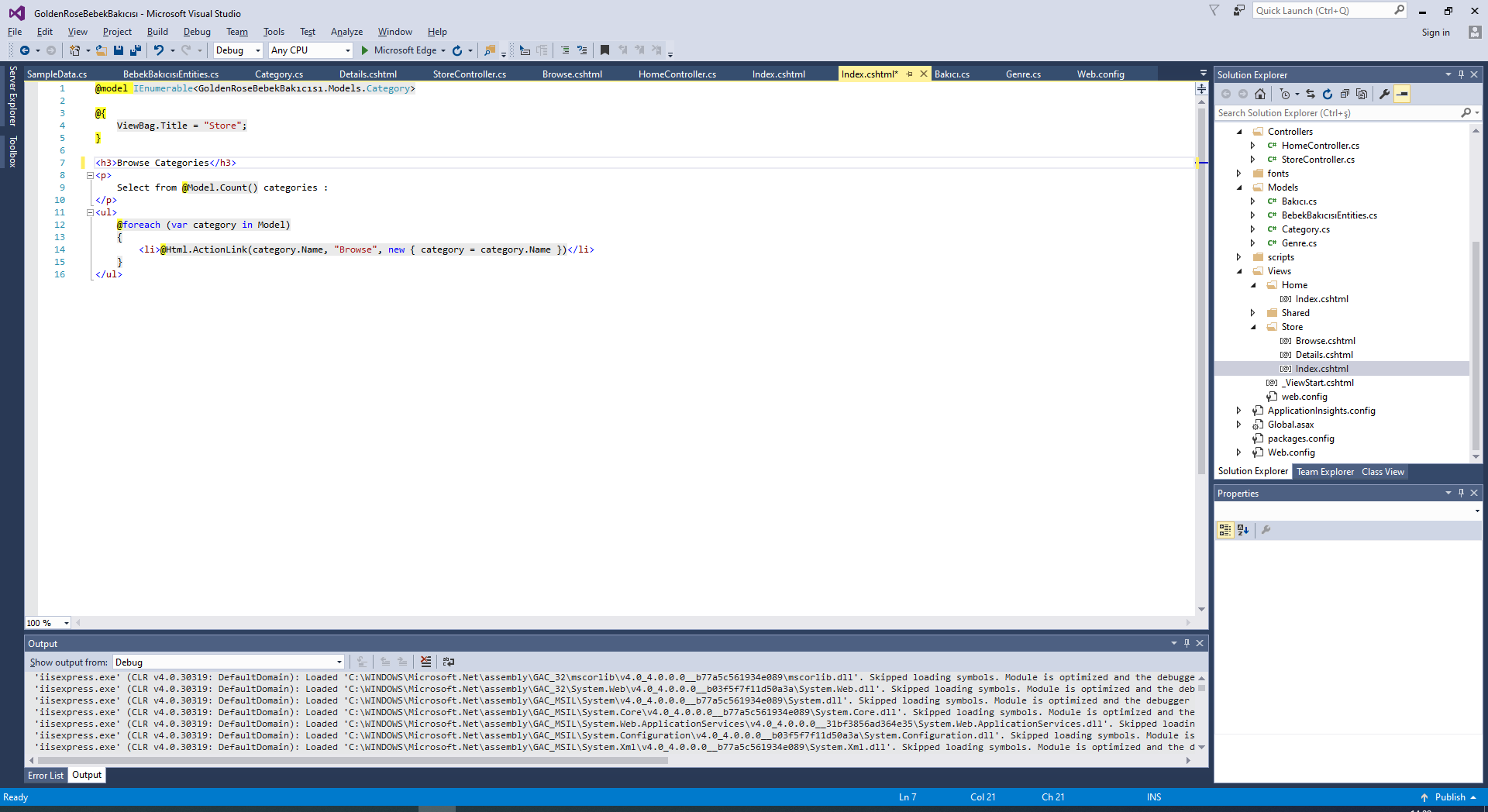


\*\*\*using GoldenRoseBebekBakıcısı.Models’i eklemediğim zaman hata aldım.Daha sonra eklediğim de hatam düzeldi.

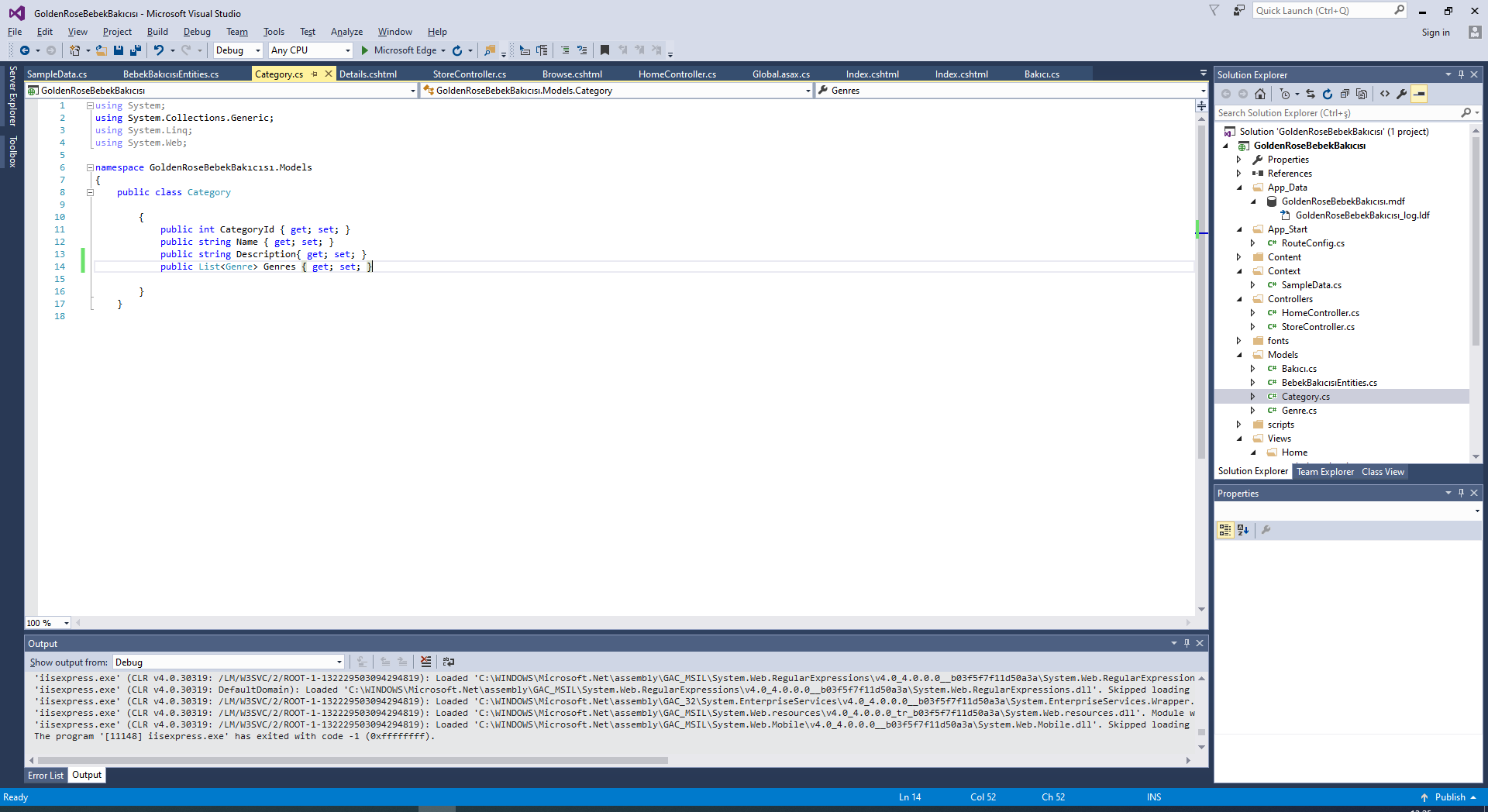




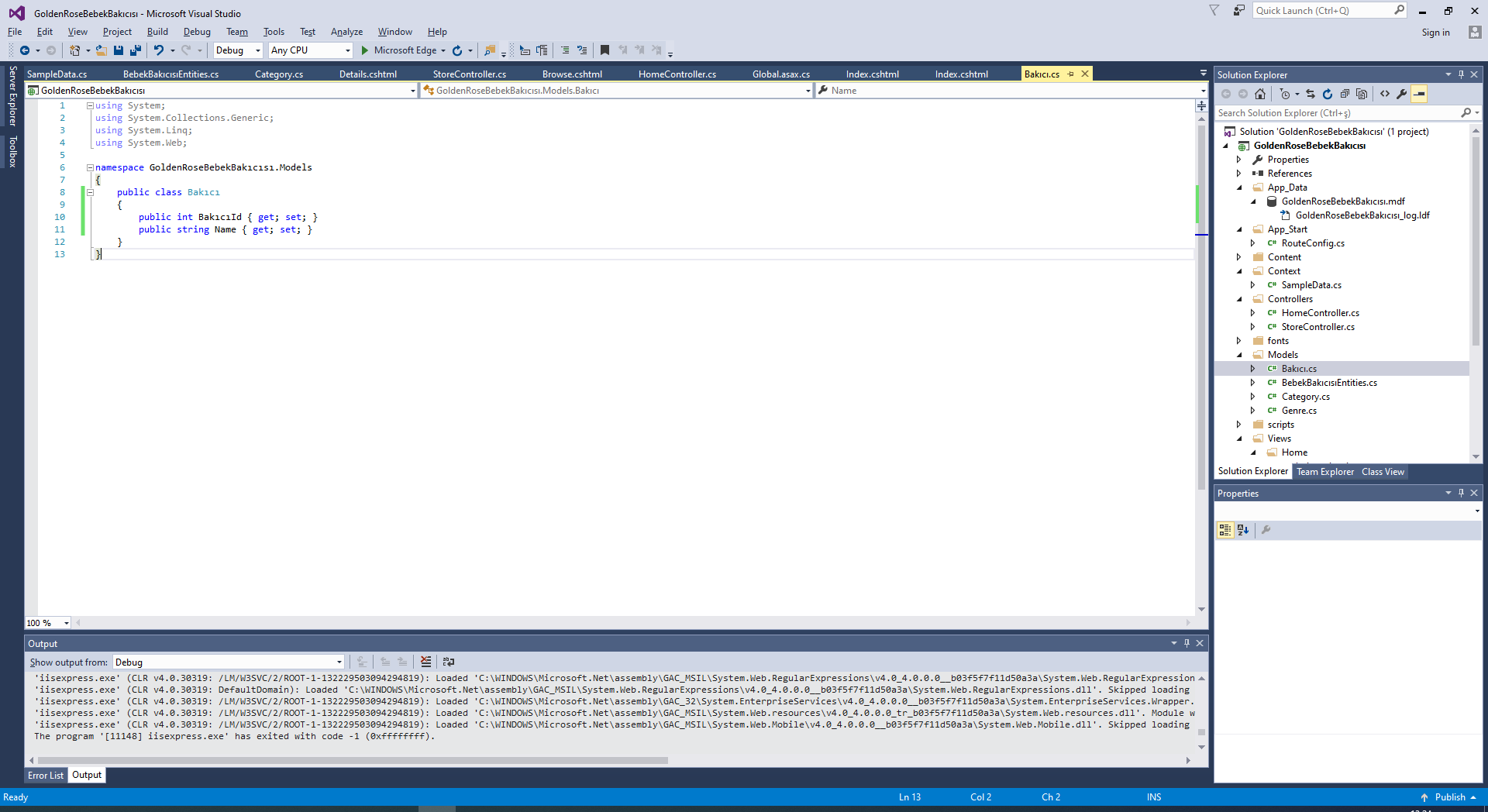


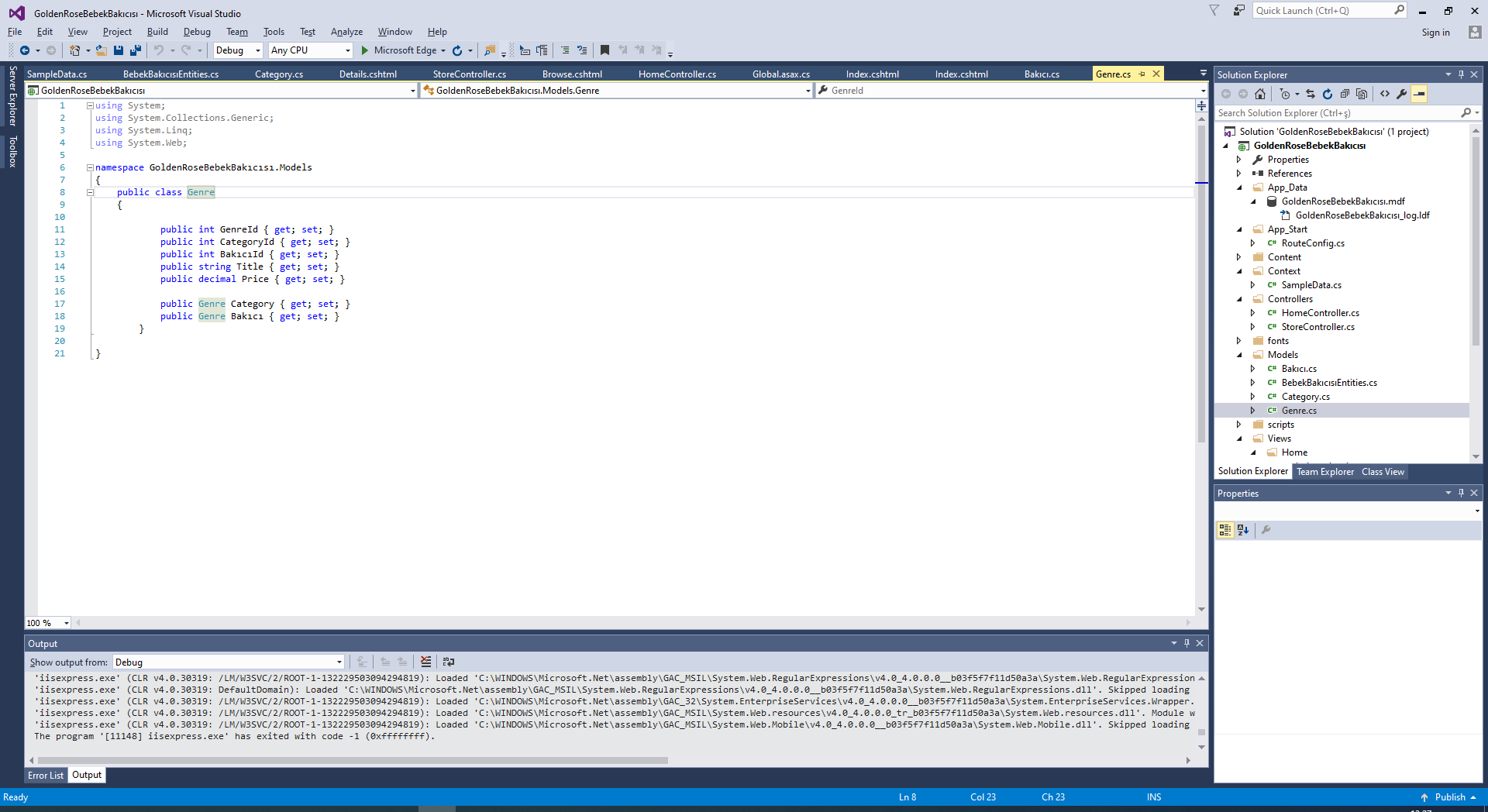


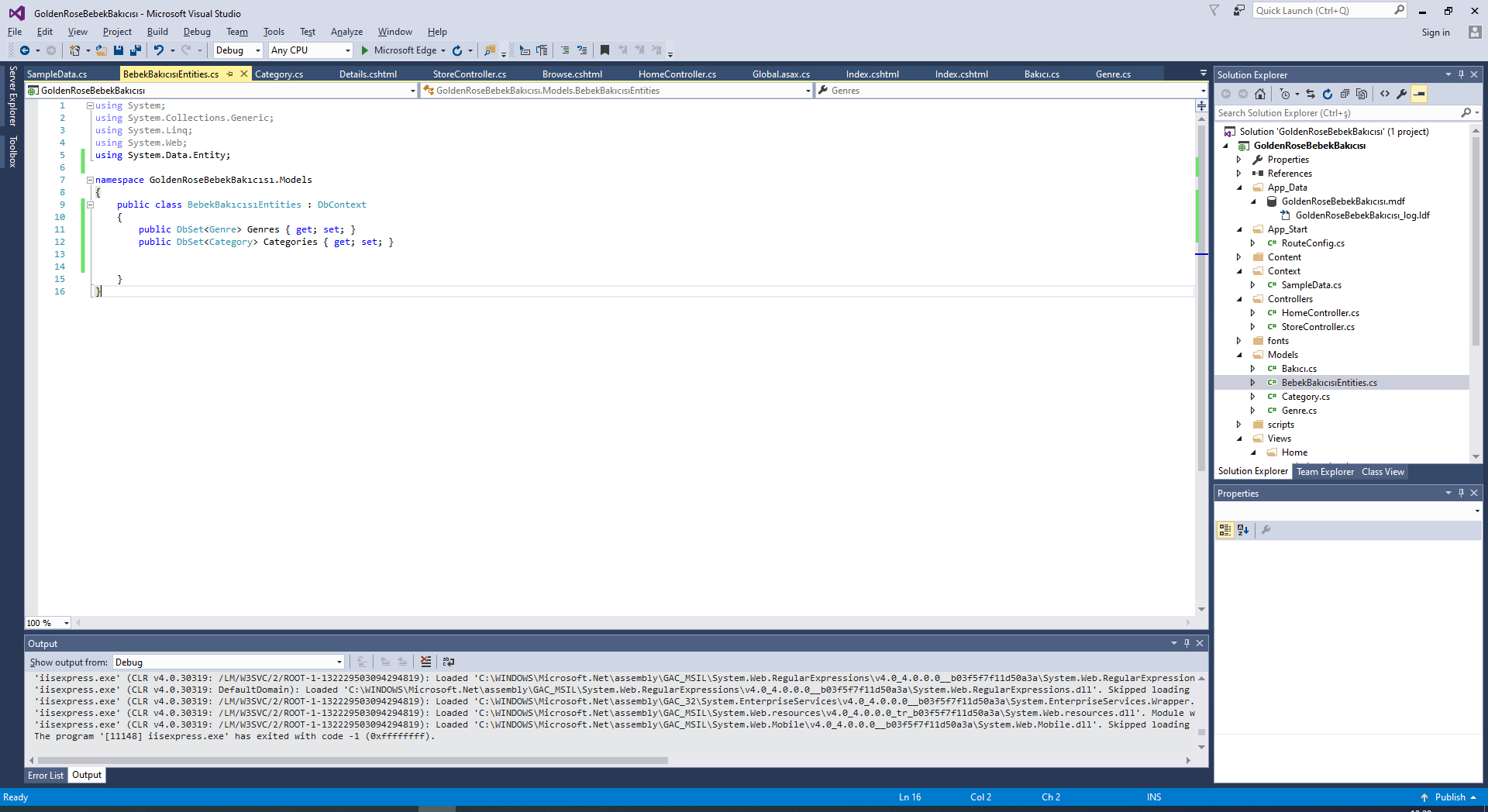
\*\*\*Store ‘ un index’ inde category yerine yanlış olan başka bir model sınıfı yazınca hata aldım.

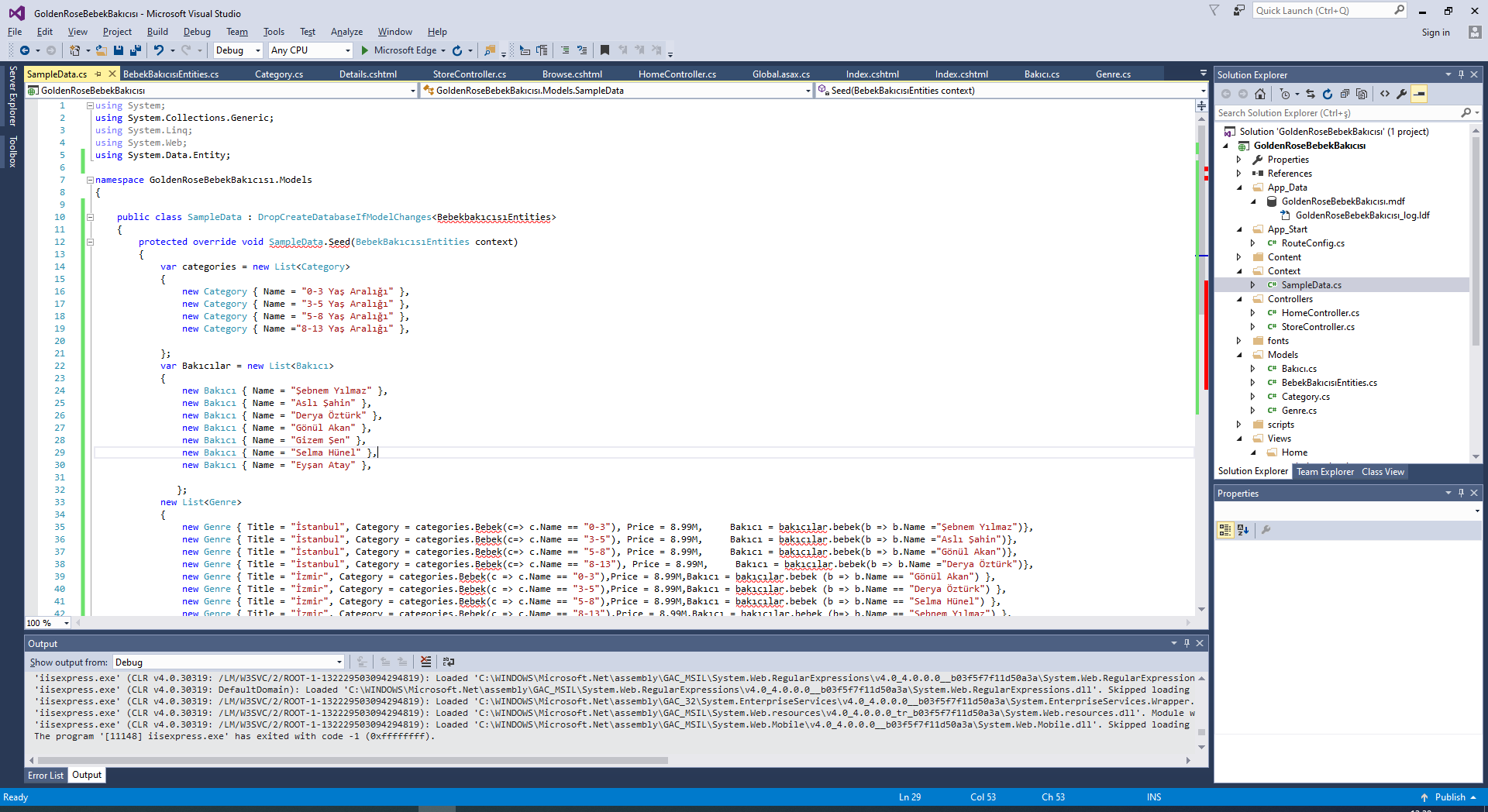


\*\*\*String’ leri ve List<> kısmını yanlış yazdığımda hata aldım.

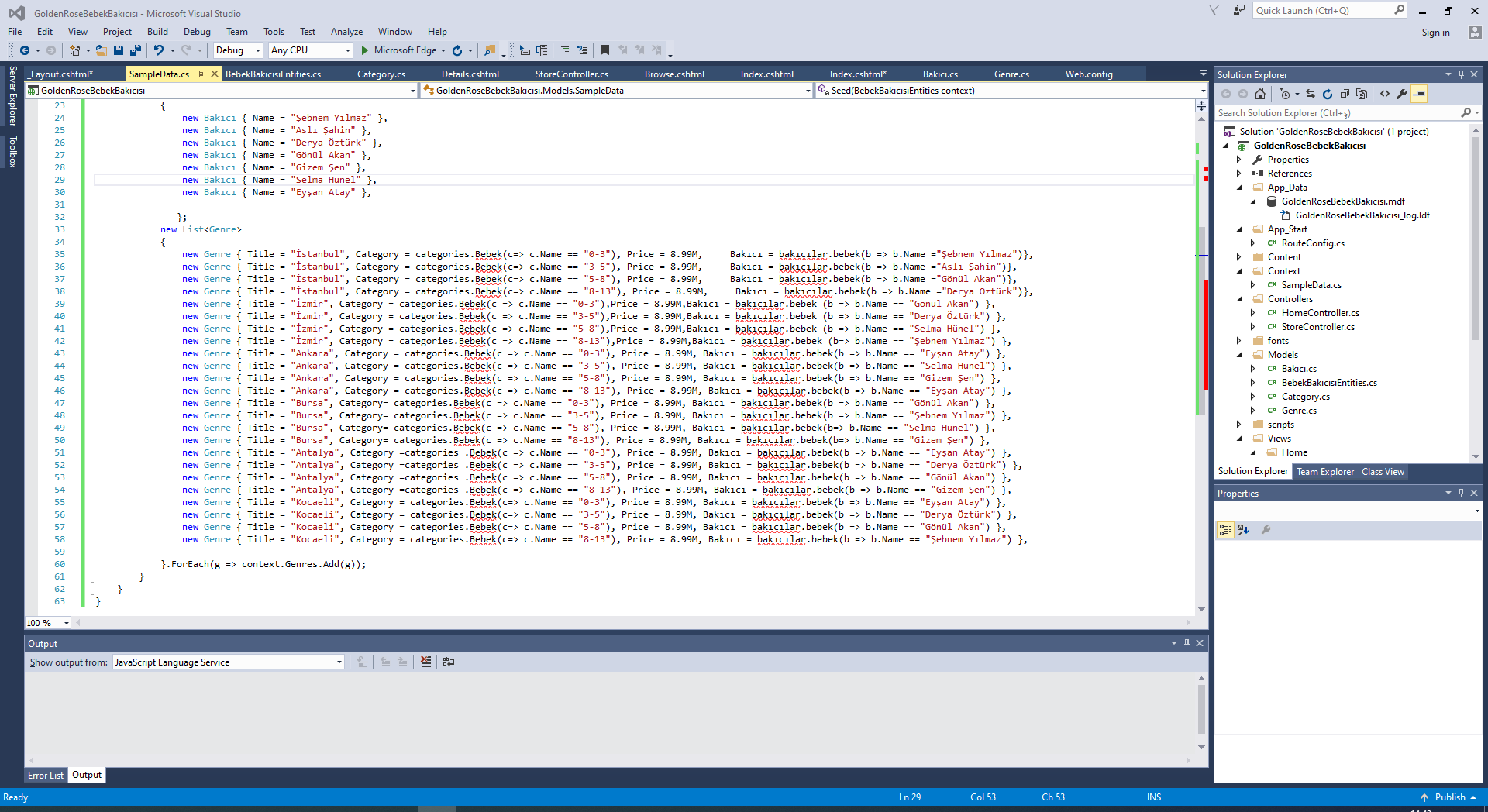




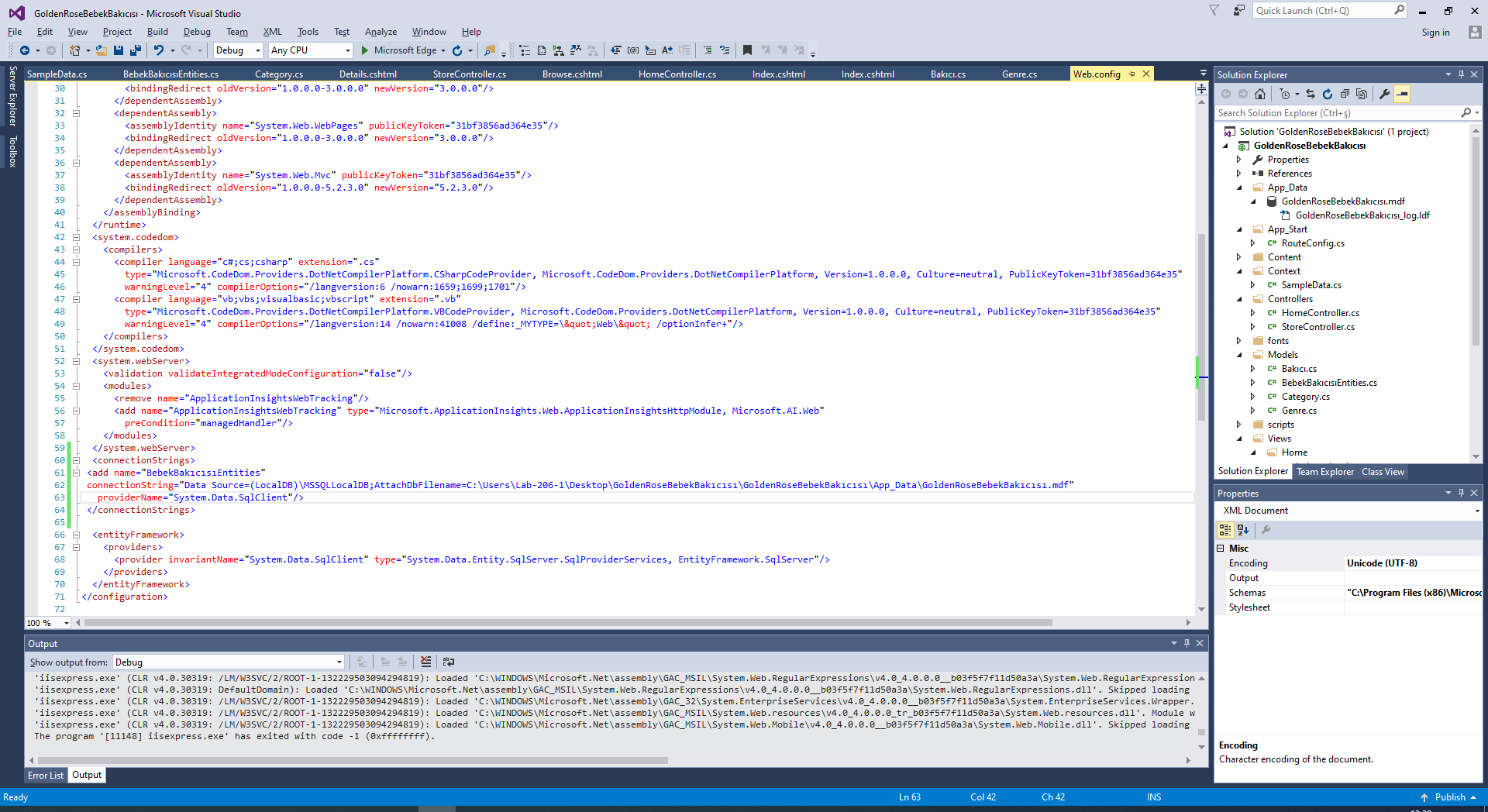




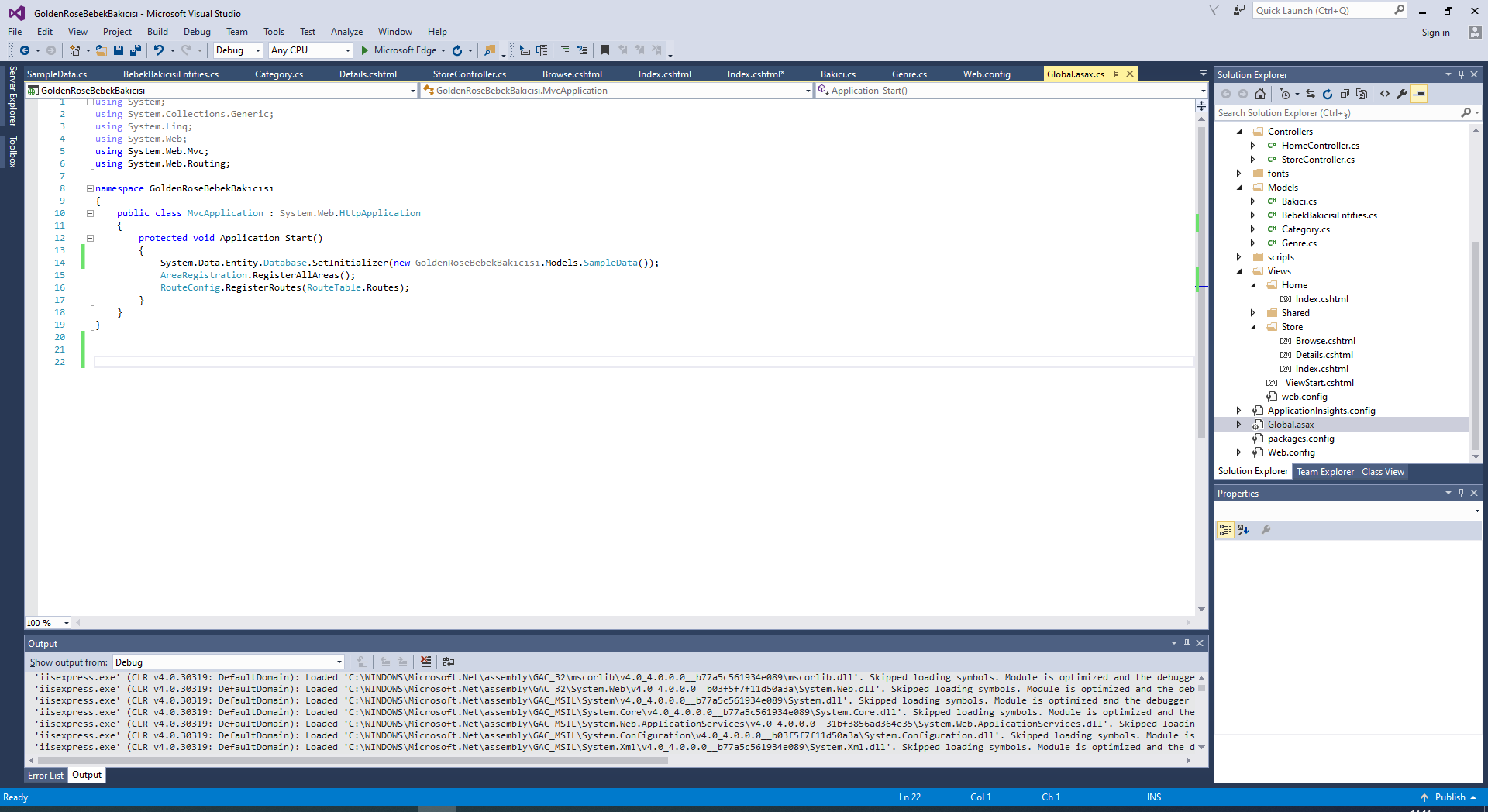
\*\*\*SampleData ‘ da kütüphaneyi eklemediğim için hata aldım. Kütüphaneyi ekledikten sonra hatam düzeldi.



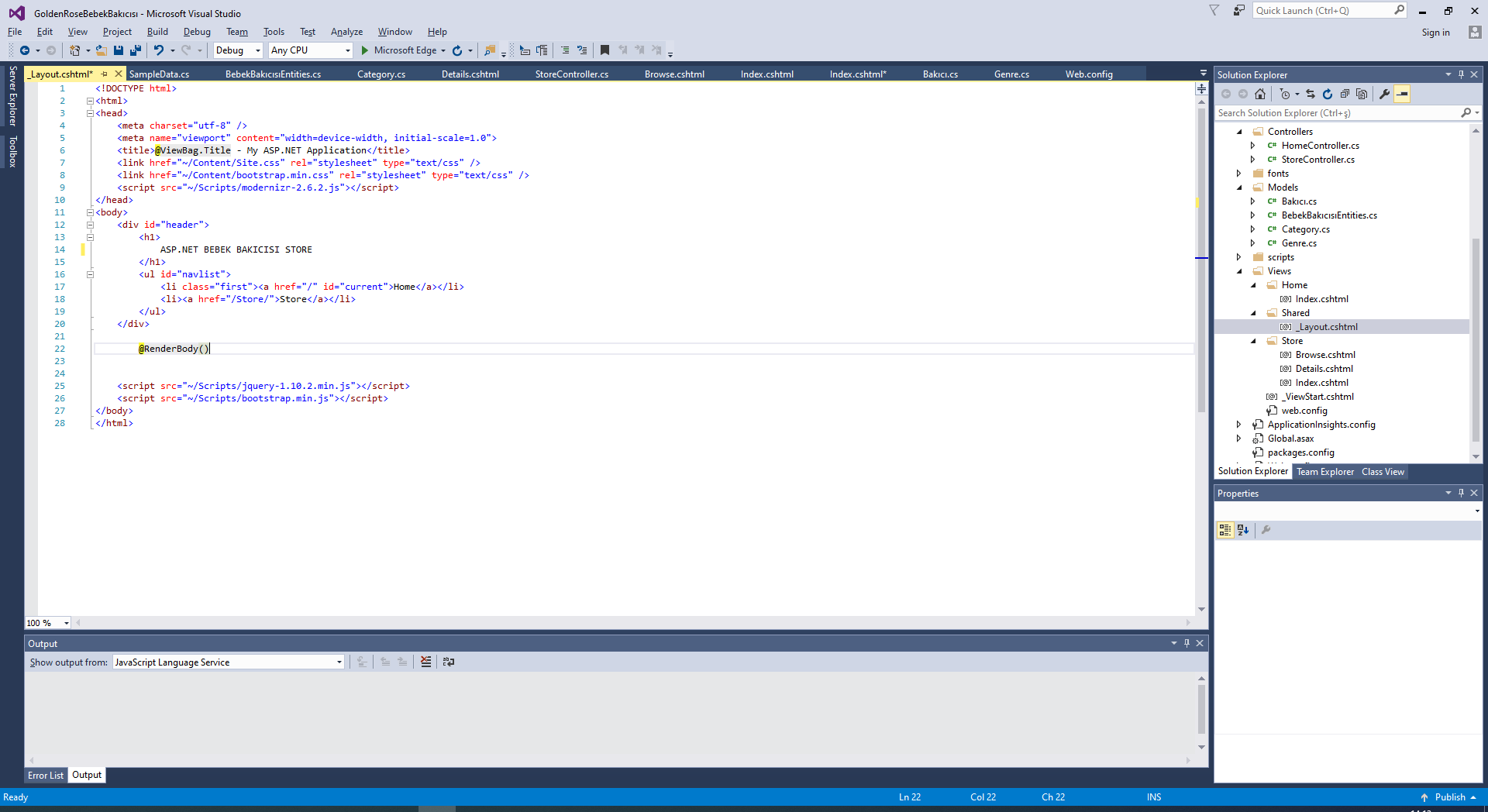
\*\*\*Category ve Bakıcı’ nın baş harfini eklemediğim için hata aldım.Daha sonra onları da ekleyip,ona göre düzenledikten sonra hatam düzeldi.



\*\*\*Web config sayfasında data source değiştirmediğim için hata aldım.Daha sonra data ekleyip,connectionString’i kopyalayıp yapıştırdıktan sonra düzeldi.



\*\*\*Global.asax ’ da System.Data.Entity ile başlayan kısmı eklemediğim için hata almıştım. Ekleyip, düzenledikten sonra hata düzeldi.



\*\*\*Bulunmaması gereken kısımlar olduğunda birçok hata aldım, daha sonra onları silip, programıma uygun bir hale gelecek şekilde yazdığımda hatam düzeldi.