ANKARA
HACI BAYRAM VELİ ÜNİVERSİTESİ
UZAKTAN EĞİTİM UYGULAMA VE ARAŞTIRMA MERKEZİ

İNTERNET PROGRAMCILIĞI

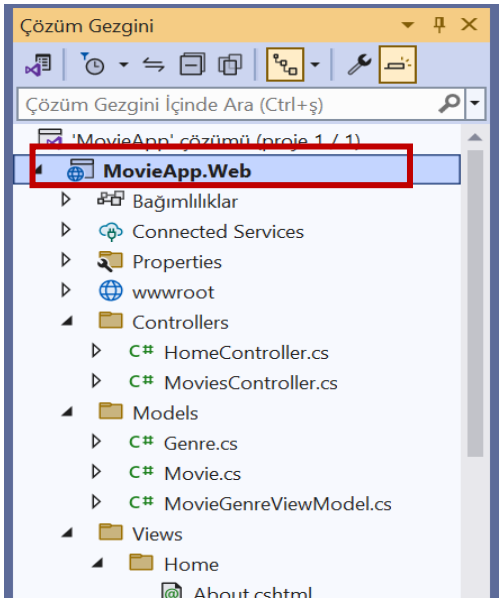
Bu bölümde proje içinde kullanılacak olan View Components, Repository uygulamalarına yer verilmiştir.

1. VIEW COMPONENTS

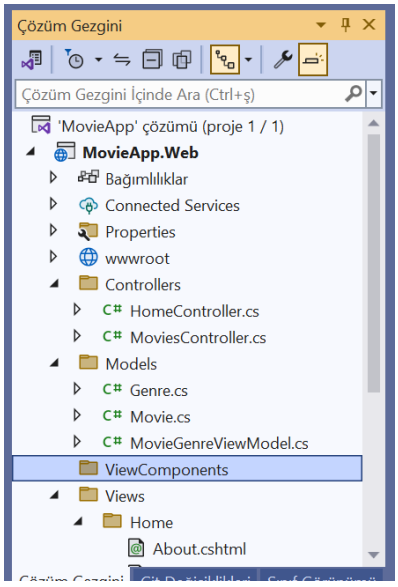
Shared klasörü içinde oluşturduğumuz .cshtml dosyaları çalışmak için bir modele ihtiyaç duyuyor.

.cshtml lerin ihtiyaç duydukları modeli kendileri elde edebilmeleri için, bu dosyalara dışardan model göndermek zorunda kalmamak için View Components ler kullanılır.

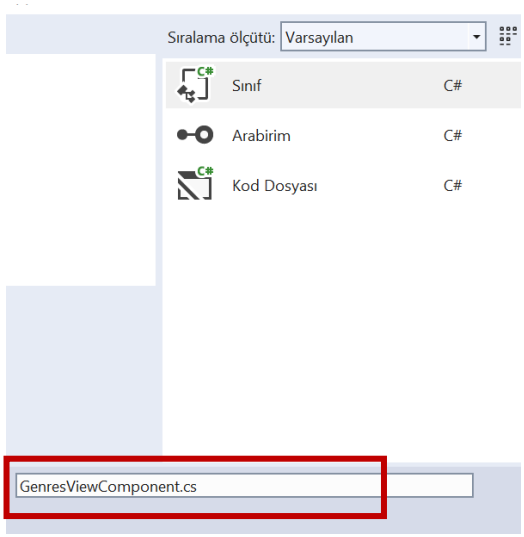
ADIM 1: Projeye sağ tıklanarak yeni klasör (new folder) eklenir.



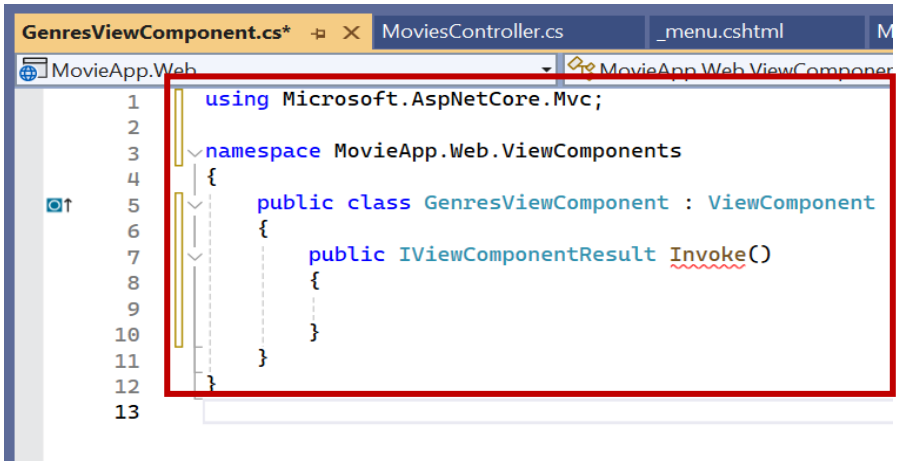
ADIM 2: Klasör **ViewComponents** olarak isimlendirilir.



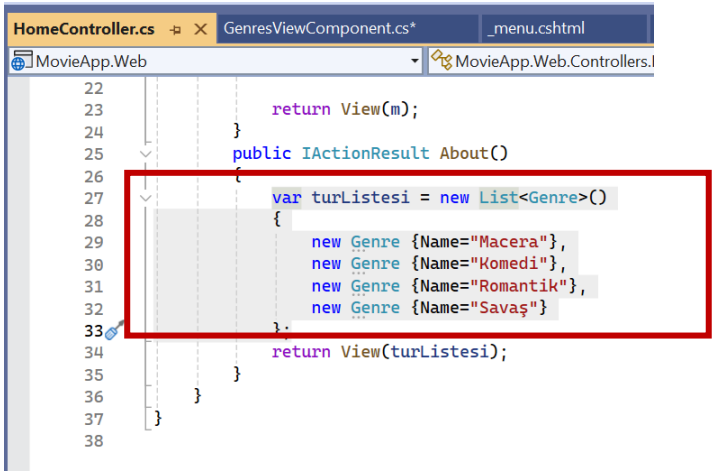
ADIM 3: ViewComponents klasörü üzerine sağ tıklanarak yeni **sınıf (class)** eklenir. **GenresViewComponent** olarak isimlendirilir.



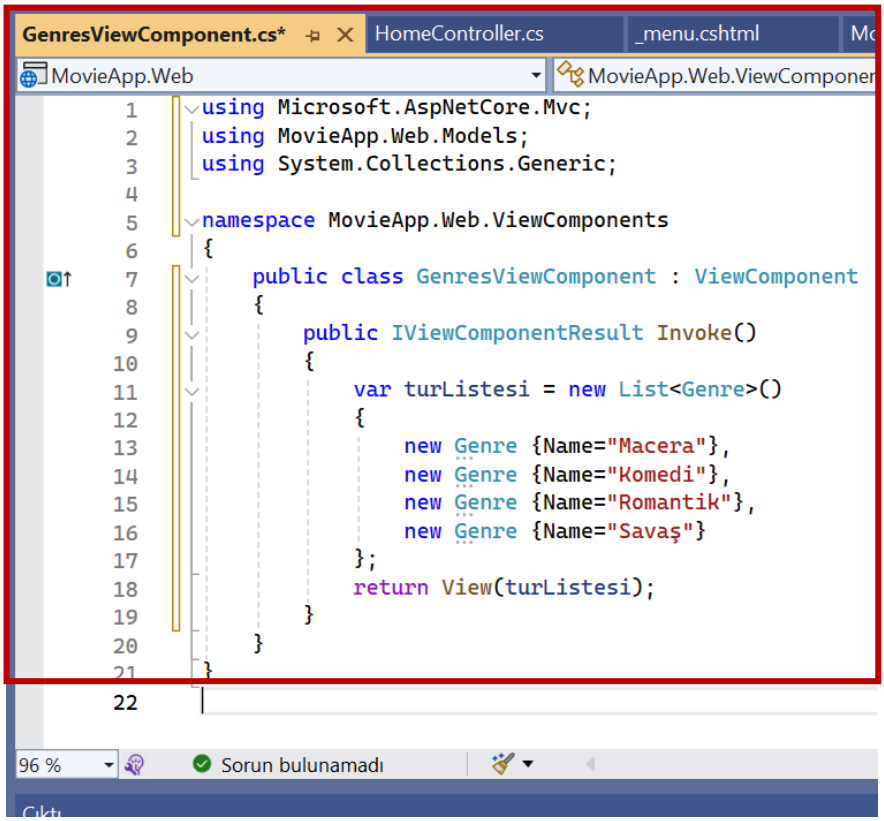
ADIM 4: GenresViewComponent.cs dosyası içine yazılması gereken kod bloğu. **Bu metod içinde bir liste talep ediyor.!!!!**



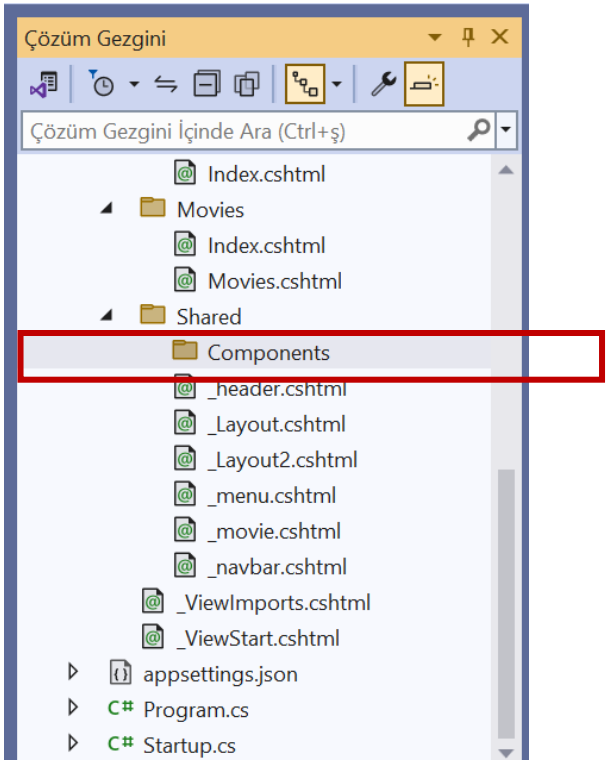
ADIM 5: HomeController.cs içinden bir liste kod bloğu **kopyalanır**.



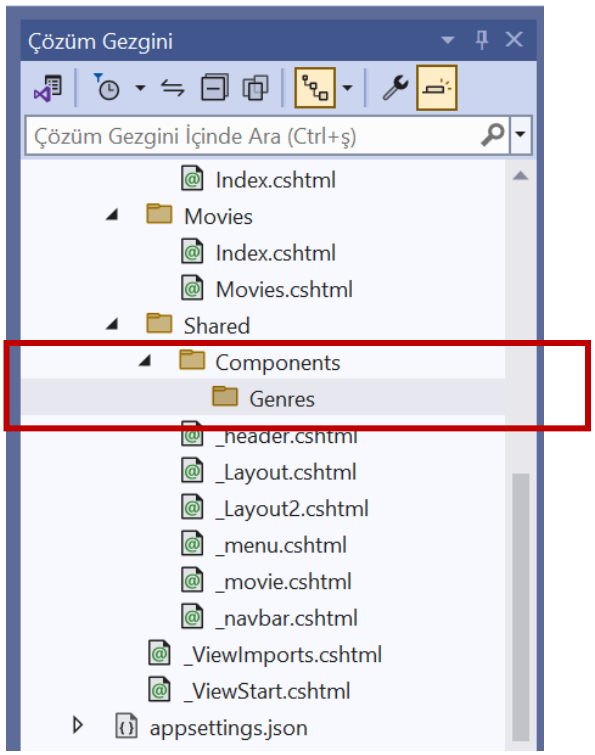
ADIM 6: GenresViewComponent.cs içinde ilgili alana yapıştırılır.



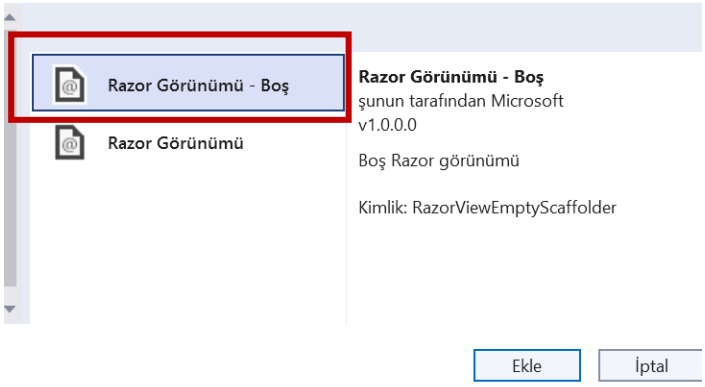
ADIM 7: Oluşturduğumuz **Model** için bir de **View** oluşturmamız gerekli. Bunun için öncelikle **Shared klasörüne sağ** tıklayarak tüm componentleri içine alabileceğimiz **Components** isminde bir **klasör** oluşturulur.



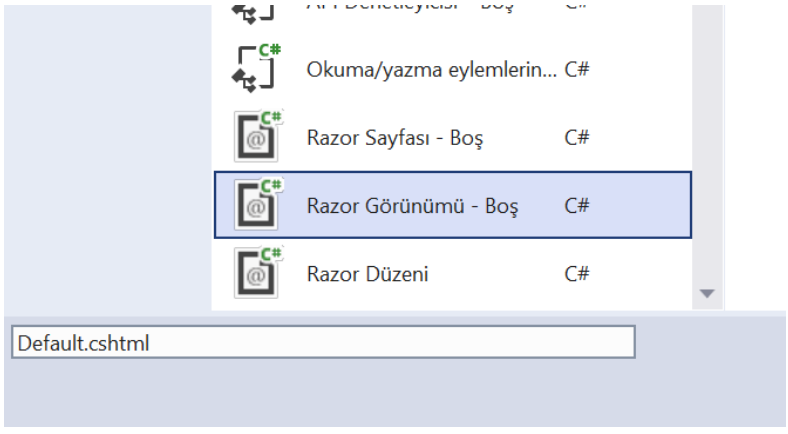
ADIM 8: Components klasörü altına da Genres adında bir klasör eklensin.



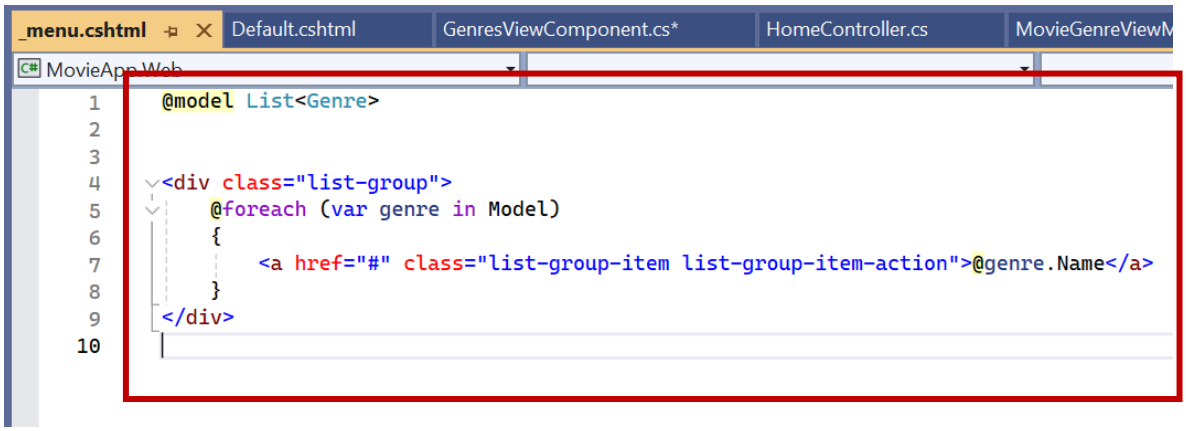
ADIM 9: **Genres** klasörüne sağ tıklanarak bir **Razor View** eklenir.



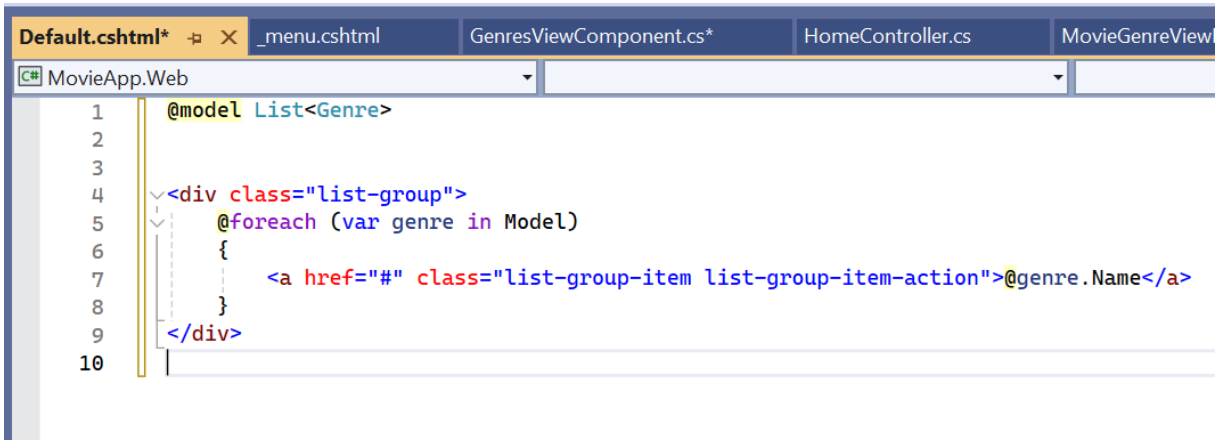
ADIM 10: **Default.cshtml** olarak isimlendirilir.



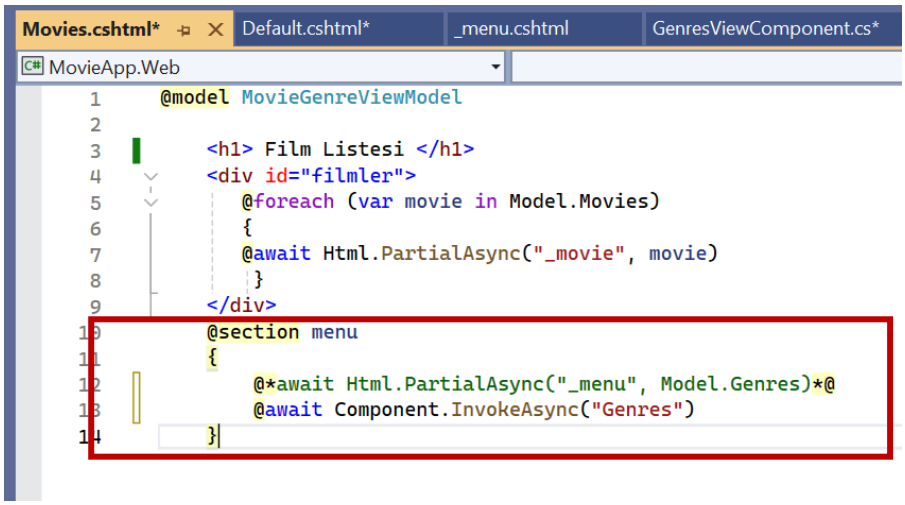
ADIM 11: _menu.cshtml deki kodlar kopyalanır.



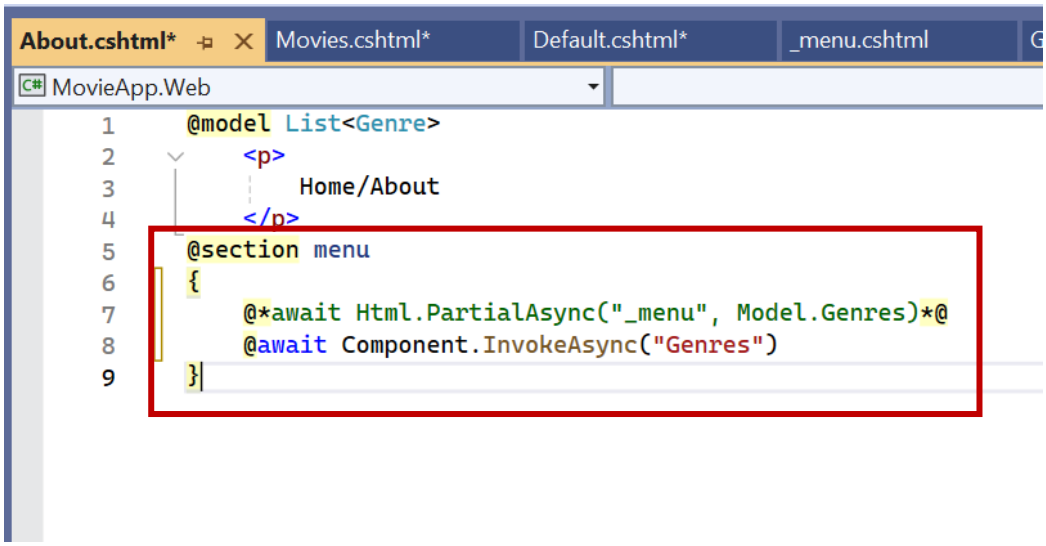
ADIM 12: Default.cshtml içine yapıştırılır.



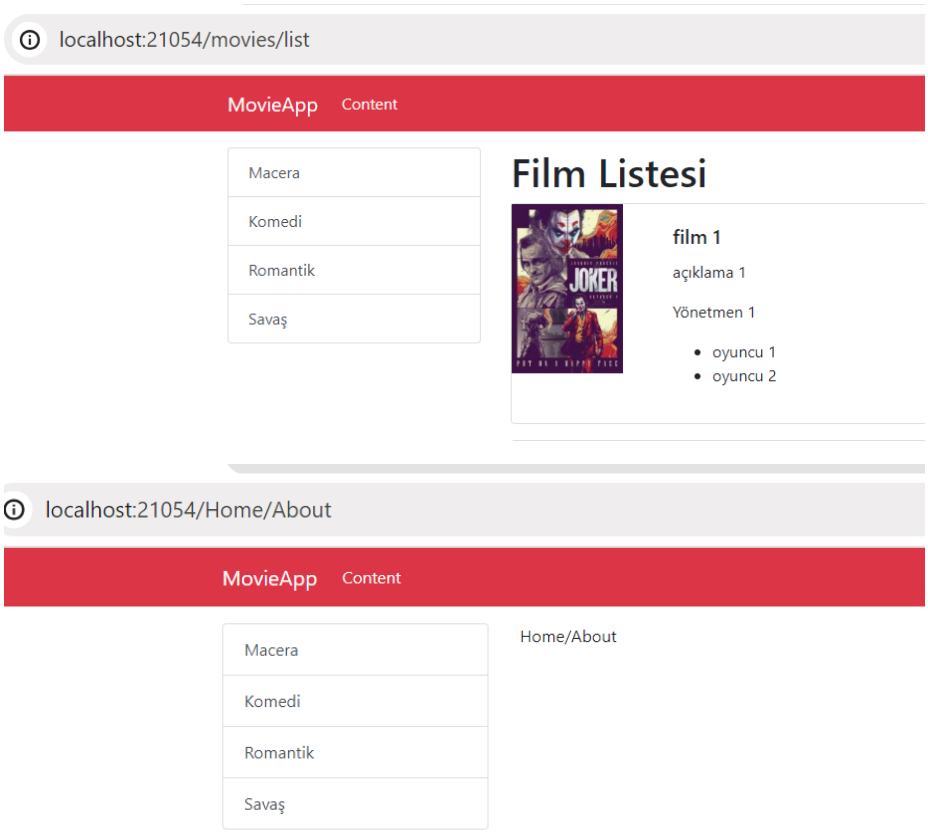
ADIM 13: Movies.cshtml içindeki kod bloğunun ilk satırı yorum satırı olarak kapatılır, altına satır eklenir.



ADIM 14: Aynı işlem **About.cshtml** için de yapılır.

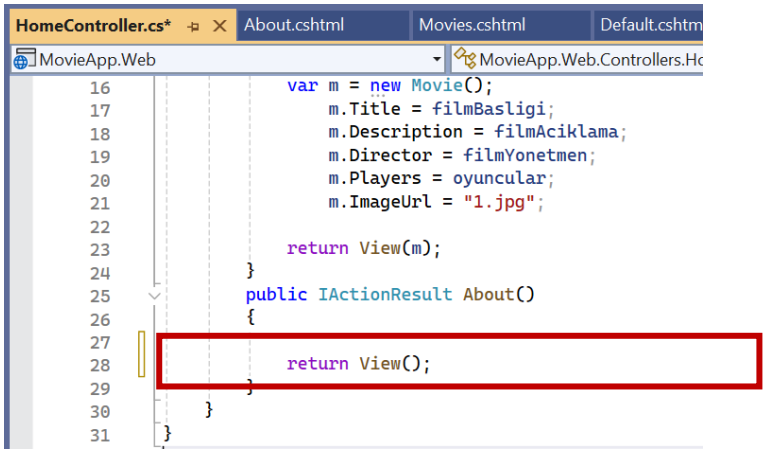


ADIM 15: Uygulama çalıştırılır. home/about ve movie/list kontrol edilir.

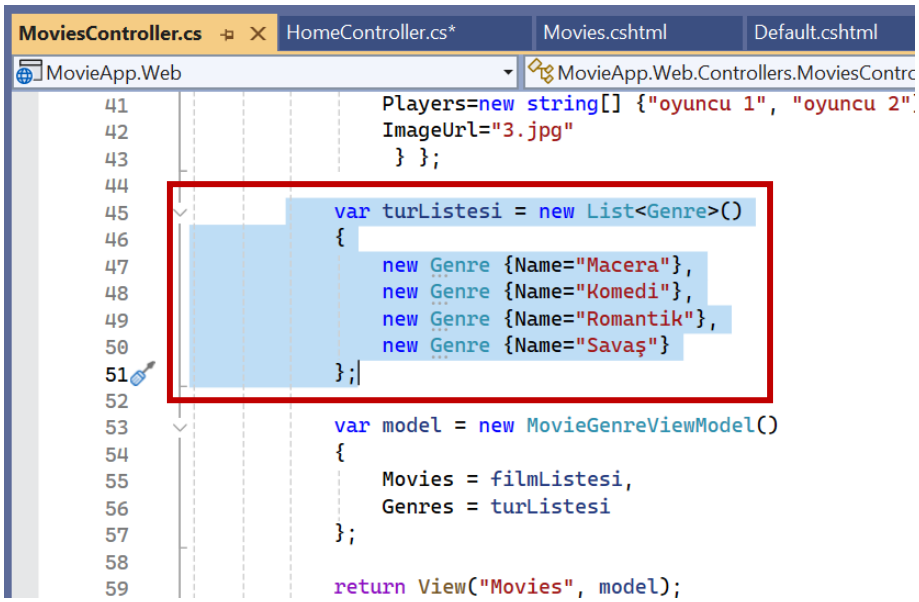


ADIM 16: HomeController.cs de ilgili tür listesine ihtiyaç kalmadı. İşaretli kod bloğu silinir.

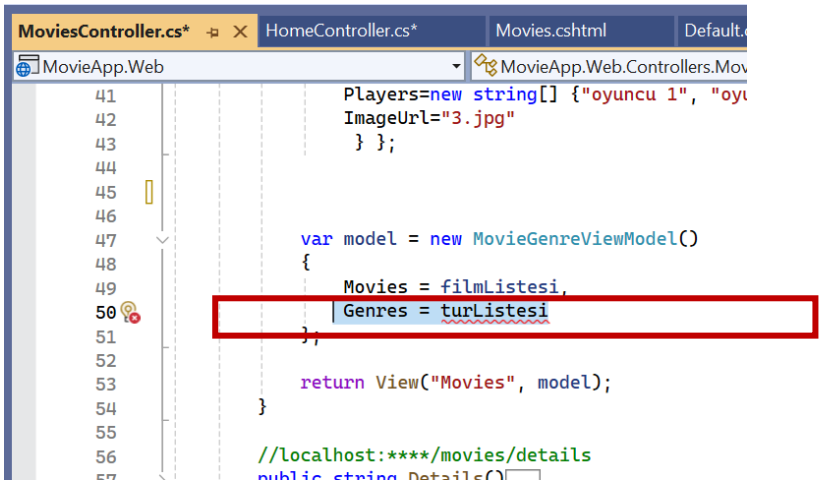
ADIM 17: return içi düzeltilir.



ADIM 18: MoviesController.cs de de artık bir tür listesine ihtiyaç kalmadı. İlgili kod bloğu buradan da silinir.



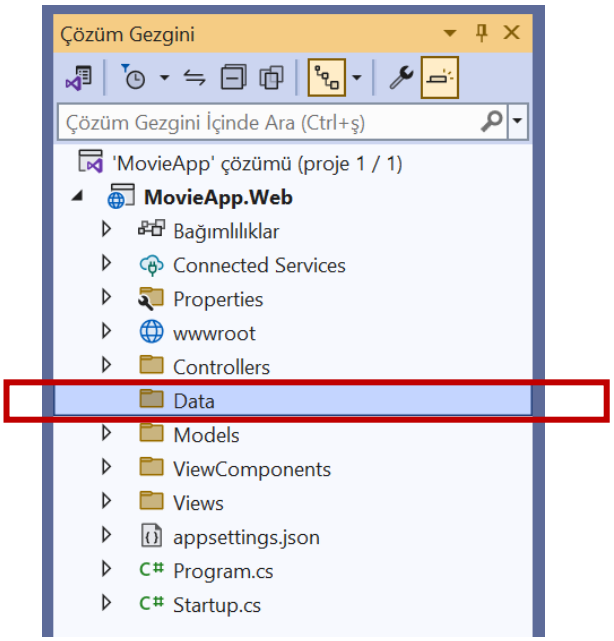
ADIM 19: Alt kod bloğundaki **Genres=turListesi** satırı da silinir.



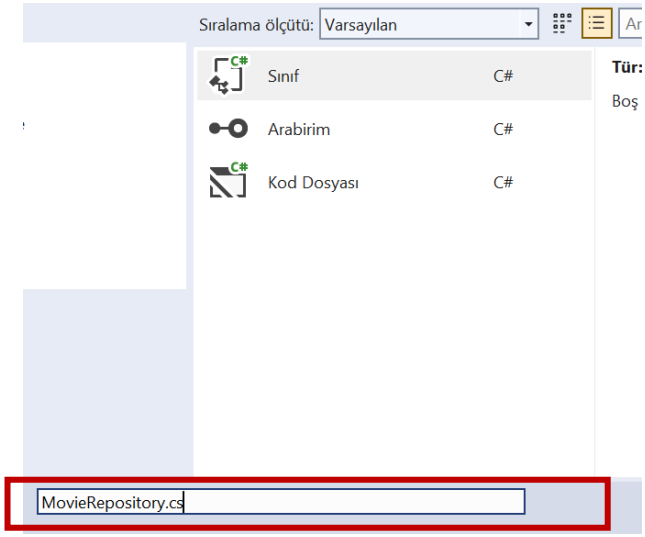
Bu derste dışarıdan gönderilen bir PartialAsync oluşturmak yerine, kendi başına çalışan bir yapı oluşturduk ve artık ilgili tür listesinin HomeController.cs de bulunmasına gerek kalmadı. Çünkü İlgili component için gelen model bilgileri **GenresViewComponent.cs** kendi içerisinde hazırlayıp kullanabiliyor.

2. REPOSITORY

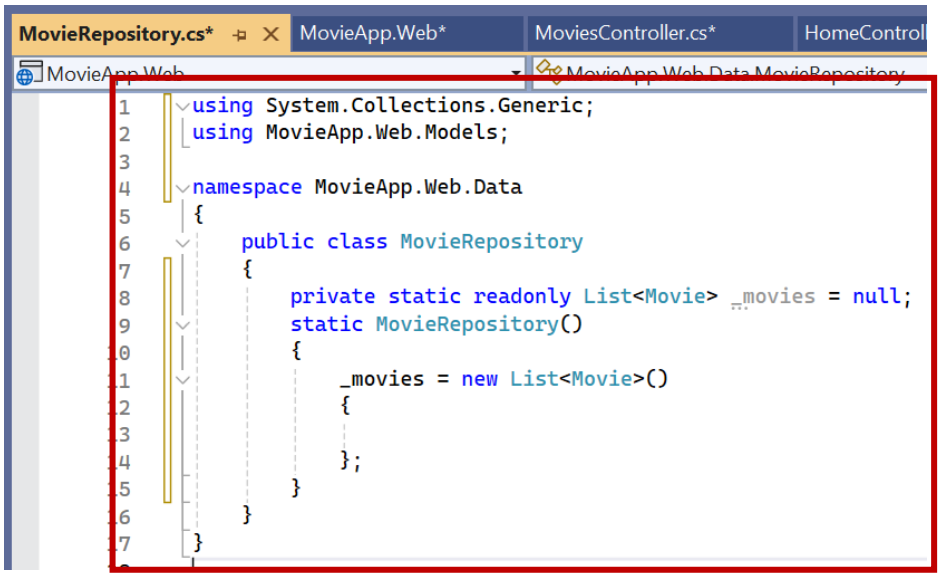
ADIM 1: Proje üzerine sağ tıklayarak yeni bir klasör oluşturulur. **Data** olarak isimlendirilir.



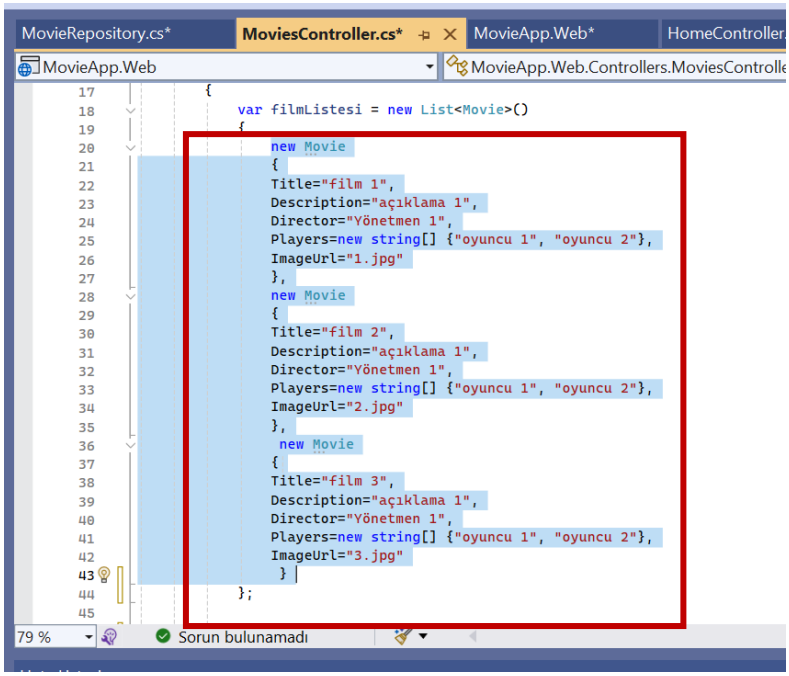
ADIM 2: Data klasörü içine yeni bir **Class** eklenir (Bu class daha sonra Genre ve Movie Modellerine veri sağlayacak şekilde kullanılacaktır). **MovieRepository.cs** olarak isimlendirilir.



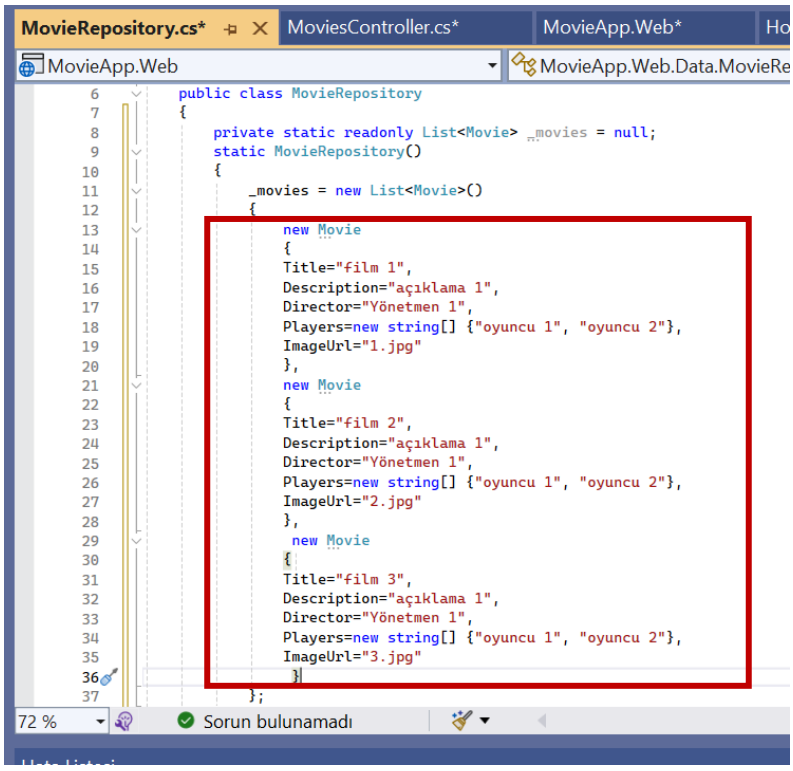
ADIM 3: Class içine sadece okunabilir dışarıdan ulaşılamayan bir liste ve ilgili kütüphaneler tanımlanır.



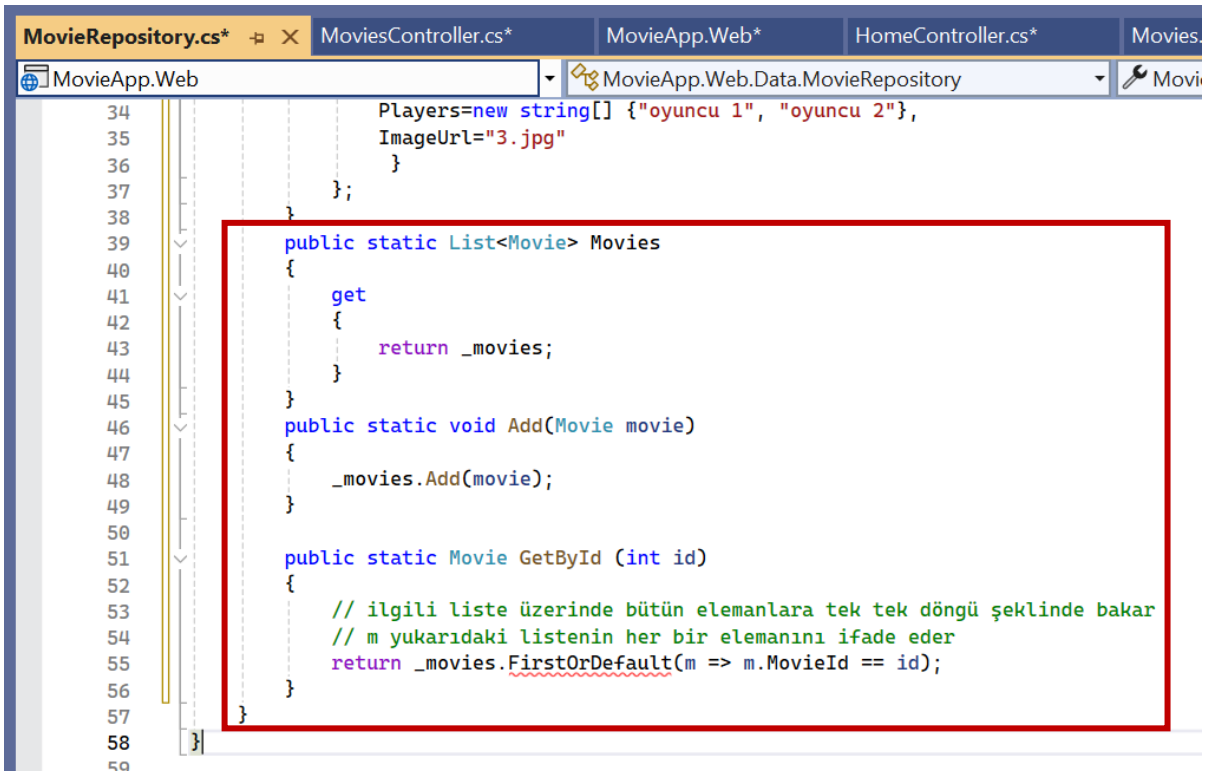
ADIM 4: MoviesController.cs içerisinden ilgili liste bilgileri kopyalanır.



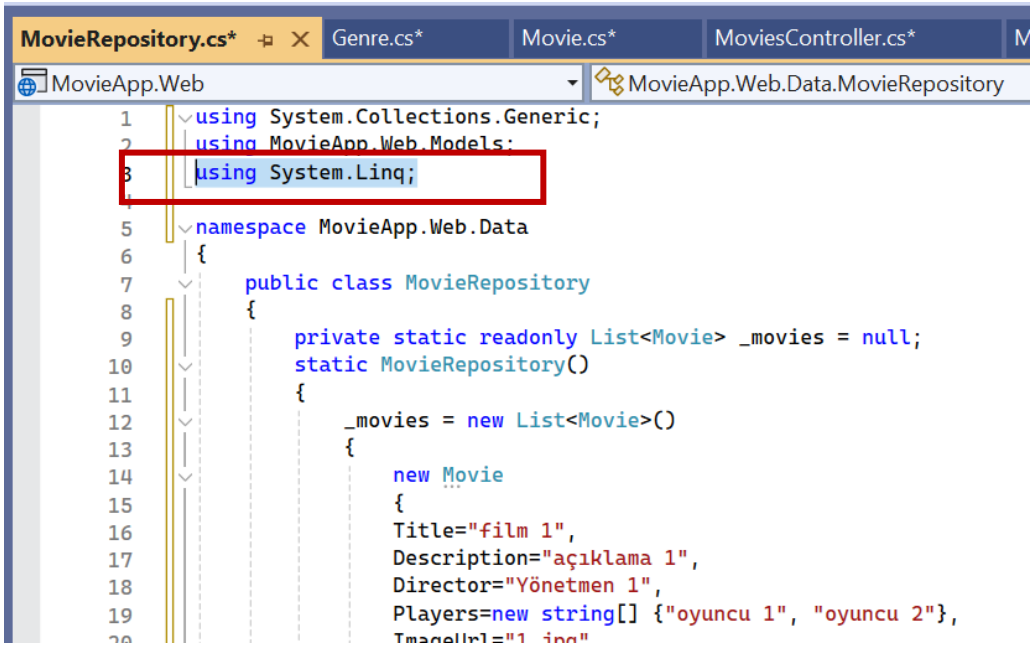
ADIM 5: MovieRepository.cs içinde ilgili alana eklenir.



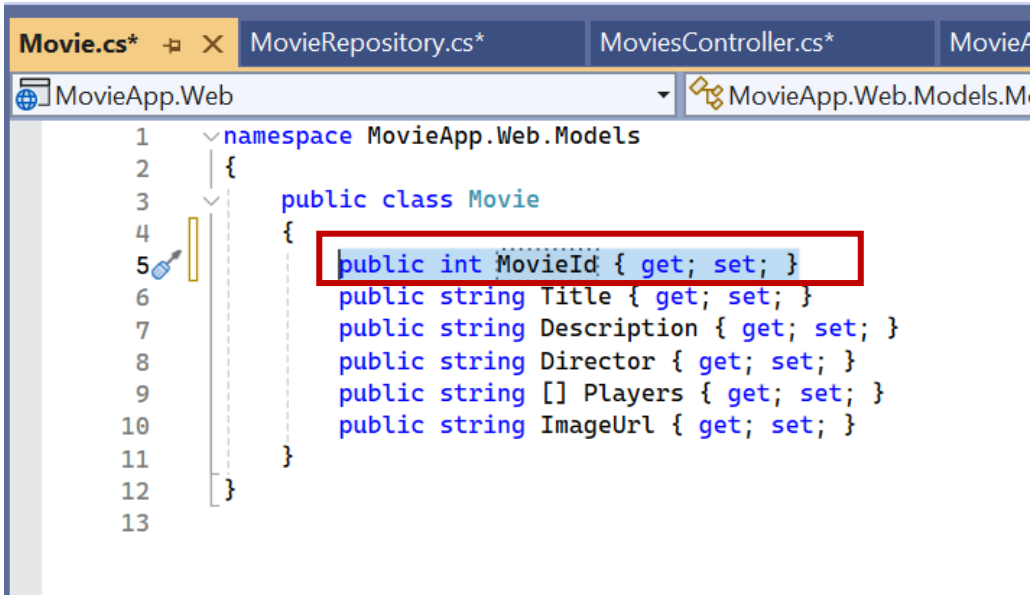
ADIM 6: Yine **MovieRepository.cs** içerisinde daha sonra kullanılacak ve açıklanacak olan metodlar için üç ayrı kod bloğunu ekliyoruz.



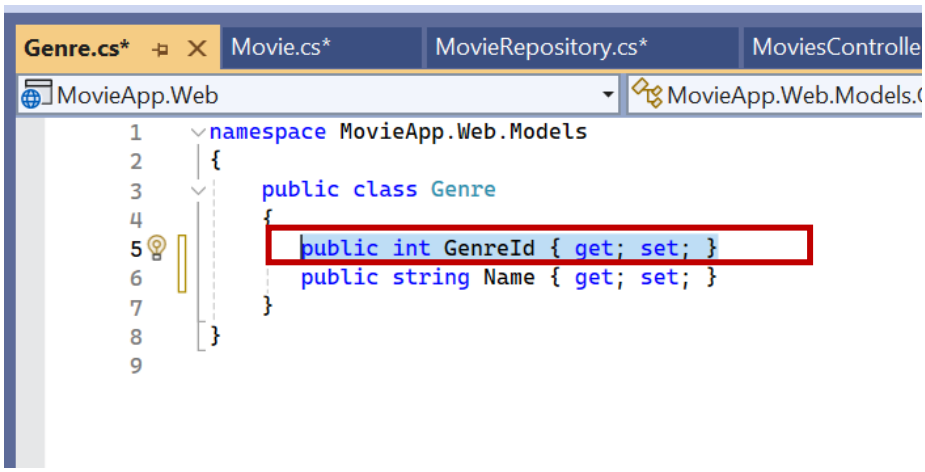
ADIM 7: FirstOrDefault kod satırının çalışması için gerekli olan **using System.Linq;** kütüphanesi **MovieRepository.cs** sayfasına eklenir.



ADIM 8: Yukarıda tanımlanan **GetById** metodu **MovieId** ye göre her bir elemanı döngü şeklinde alır ve her bir elemanı m olarak kabul eder. Fakat bizim daha önce Models Klasörü içinde tanımladığımız **Movie.cs** klasındaki kod bloğunda **MovieId** yoktu. **MovieId** ekliyoruz.

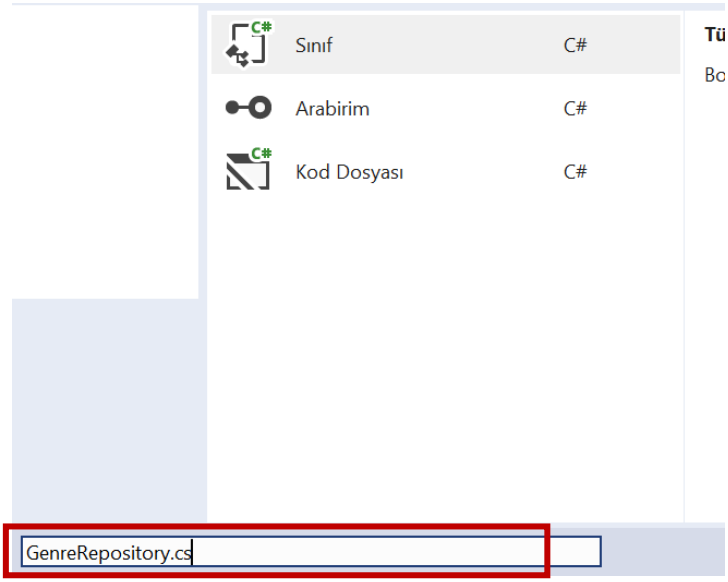


ADIM 9: Aynı şekilde **Genre.cs** sayfasına da Filmlerin tür bilgilerine ilişkin id satırını GenreId olarak ekliyoruz.

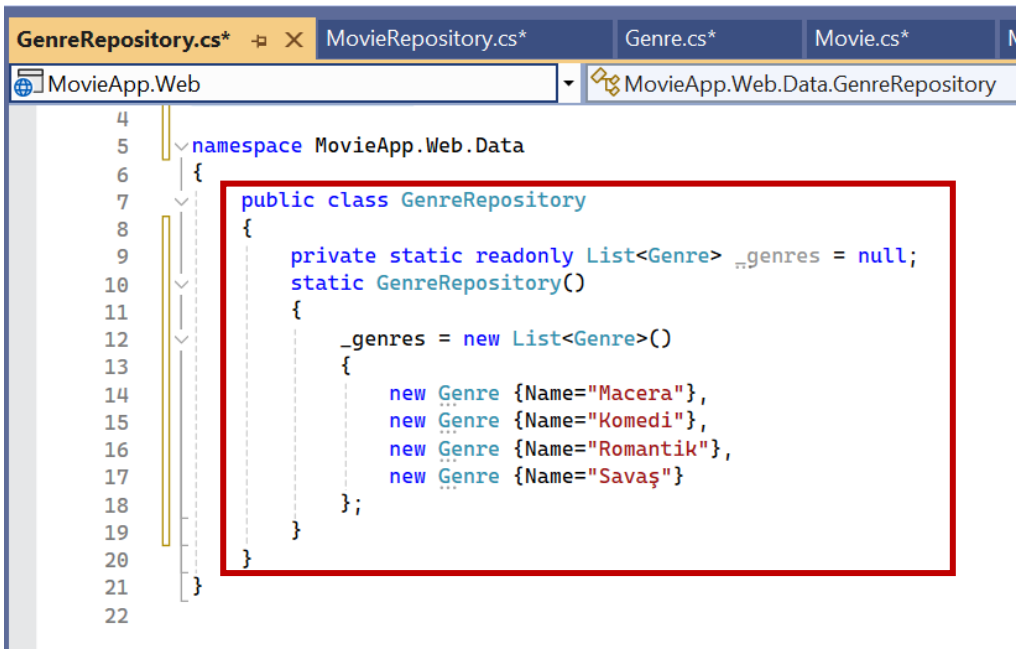


Data Klasörü içinde Movie Models için MovieRepository.cs class ı oluşturduk. Şimdi Genre Models'ı için aynı işlemleri tekrarlayalım.

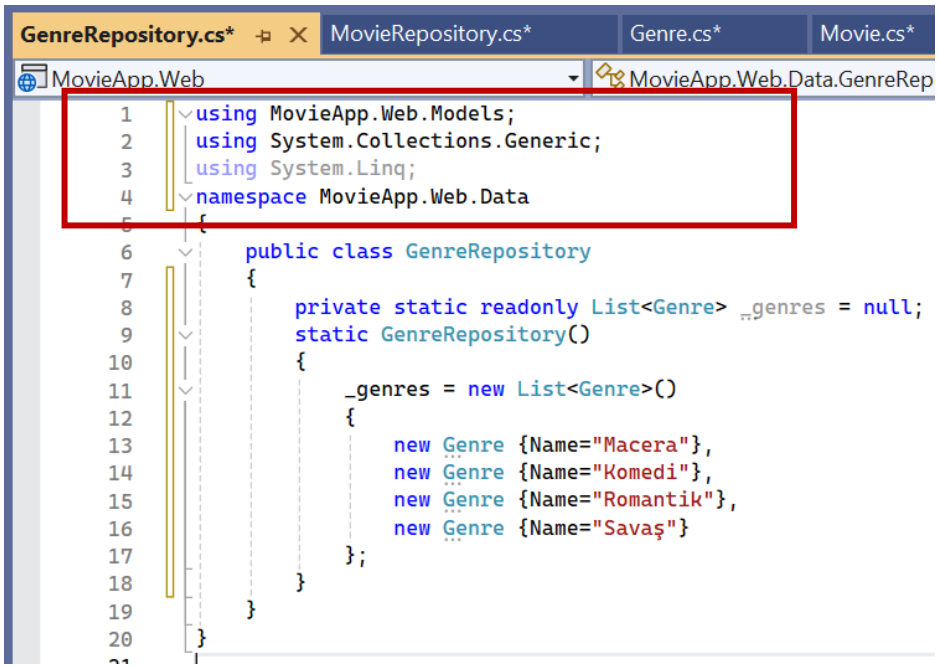
ADIM 10: Data klasörüne sağ tıklanarak yeni bir **class** oluşturulur**. GenreRepository.cs** olarak isimlendirilir.



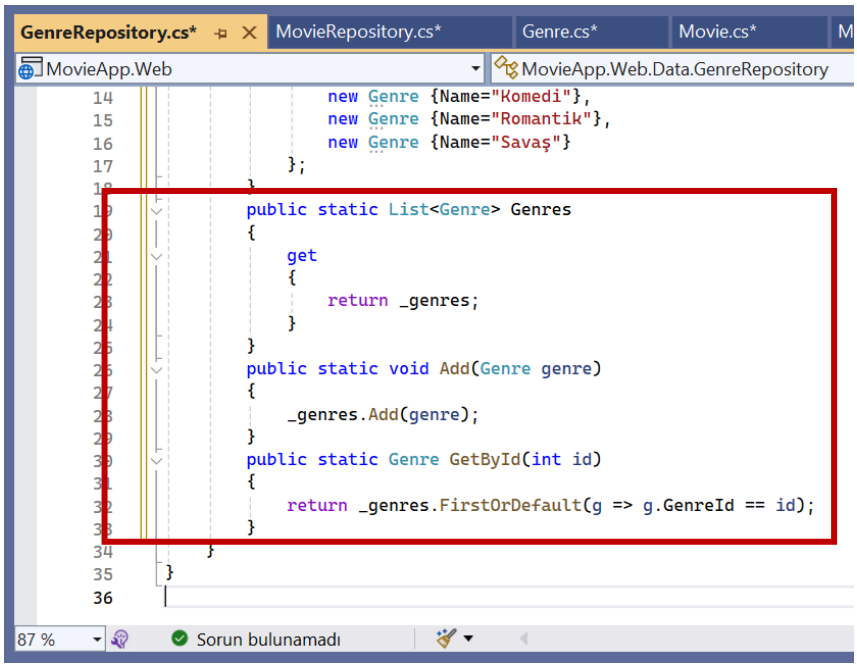
ADIM 11: GenreRepository.cs içine de aynı şekilde Genre bilgisini alacak sadece okunabilir bir liste aşağıdaki kod bloğu ile tanımlanır.



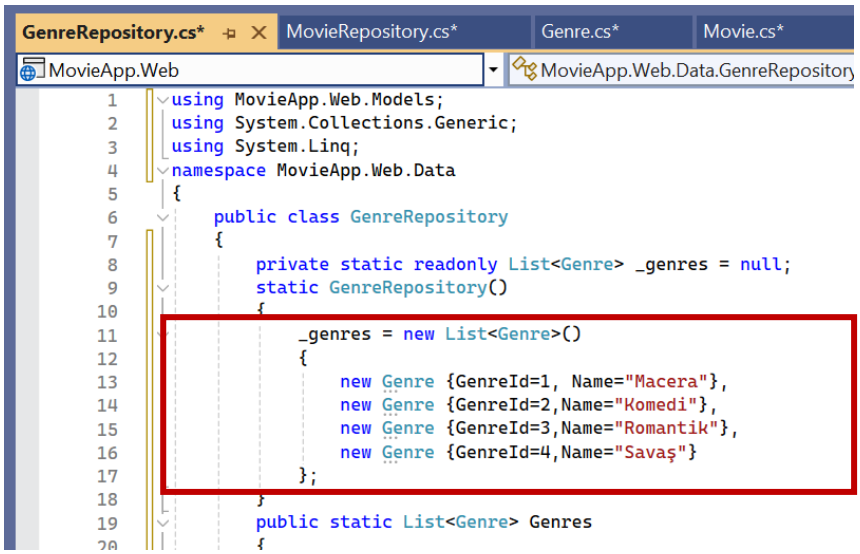
ADIM 12: Gerekli Kütüphaneler eklenir.



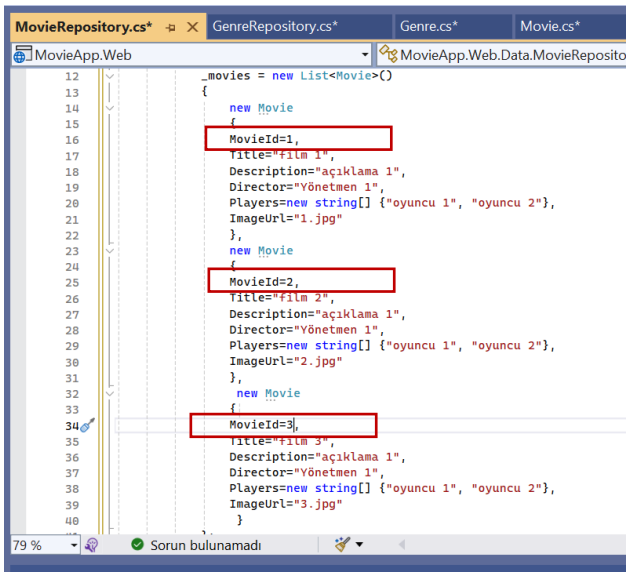
ADIM 13: İlgili kod bloğu eklenir.



ADIM 14: GenreReposity.cs de GenreId bilgileri eklenir.



ADIM 15: MovieReposity.cs dosyasında ilgili kod bloğuna MovieId ler eklenir.



ÇALIŞMA SORUSU

Uygulamada GenresViewComponent metodu ne amaçla kullanılmaktadır açıklayınız.