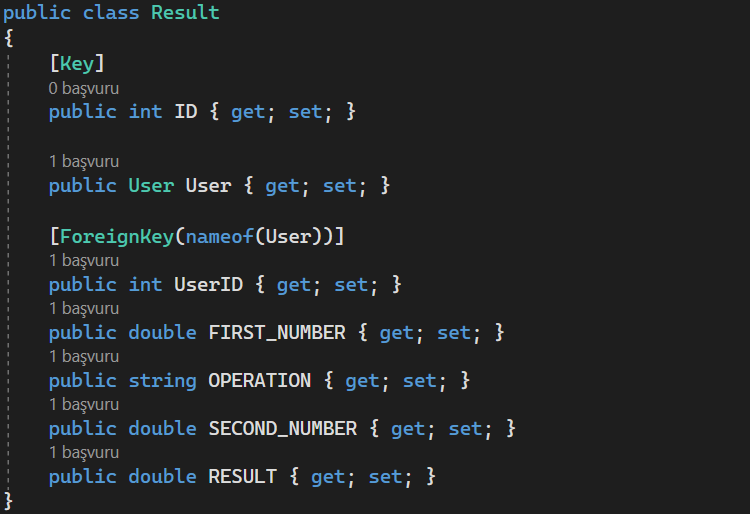
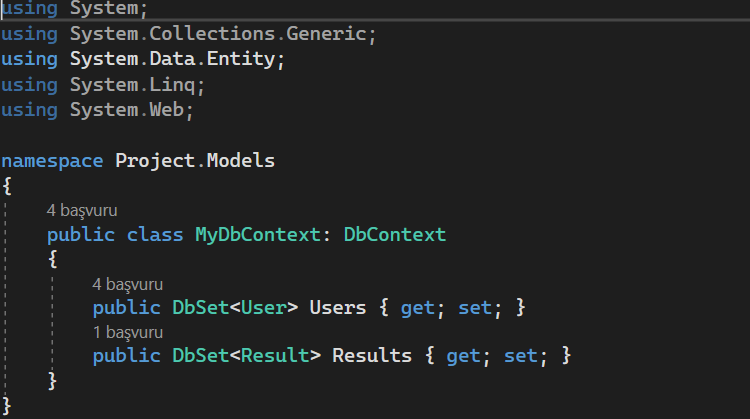


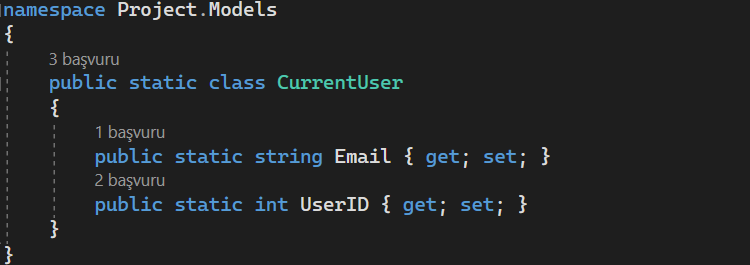
Models klasöründe bulunan "User" sınıfı, kullanıcı bilgilerini temsil eder. Bu sınıf, kullanıcının kimlik bilgilerini (EMAIL ve PASSWORD) ve kullanıcının hesaplama sonuçlarını içeren bir "Results" listesini içerir.



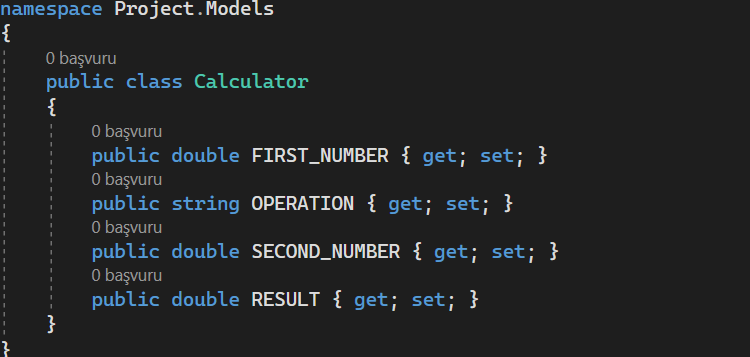
"Result" sınıfı, hesaplama sonuçlarını temsil eder. Bu sınıf, bir sonuç kaydının ID'sini, kullanıcıyı temsil eden bir User nesnesini, ilgili kullanıcının kimlik numarasını (UserID), işlemi temsil eden bir OPERATION dizesini, ilk sayıyı (FIRST\_NUMBER), ikinci sayıyı (SECOND\_NUMBER) ve sonucu (RESULT) içerir.



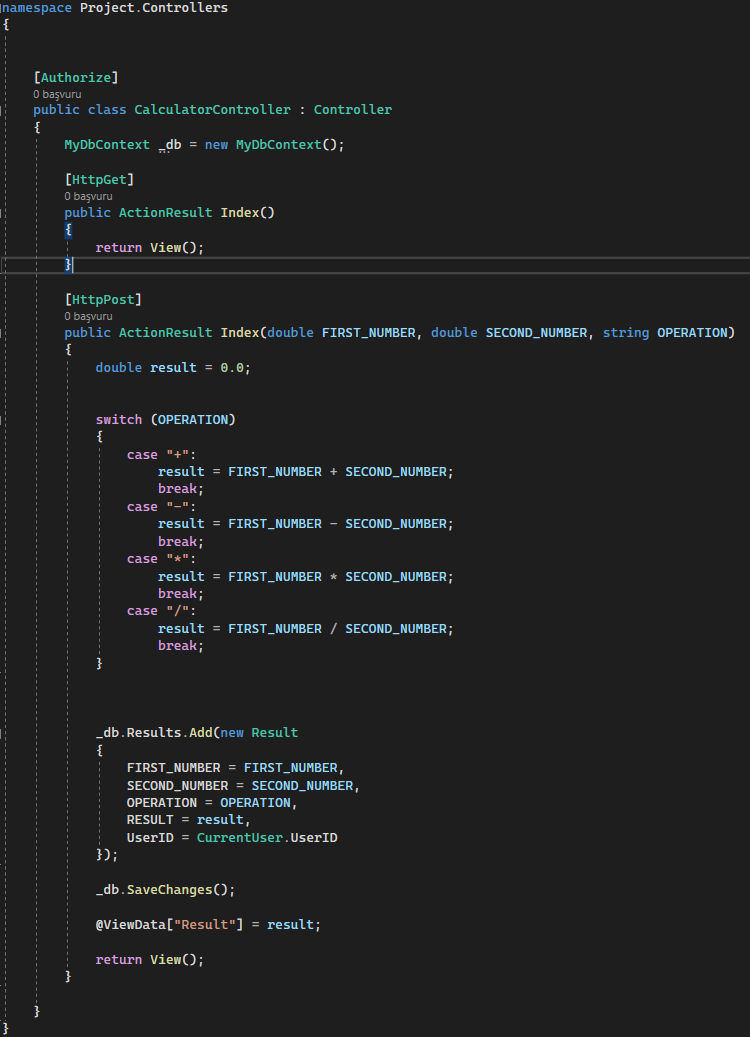
"MyDbContext" sınıfı, veritabanı bağlantısını ve ilgili tabloları temsil eder. Bu sınıf, Users tablosunu ve Results tablosunu DbSet olarak içerir.



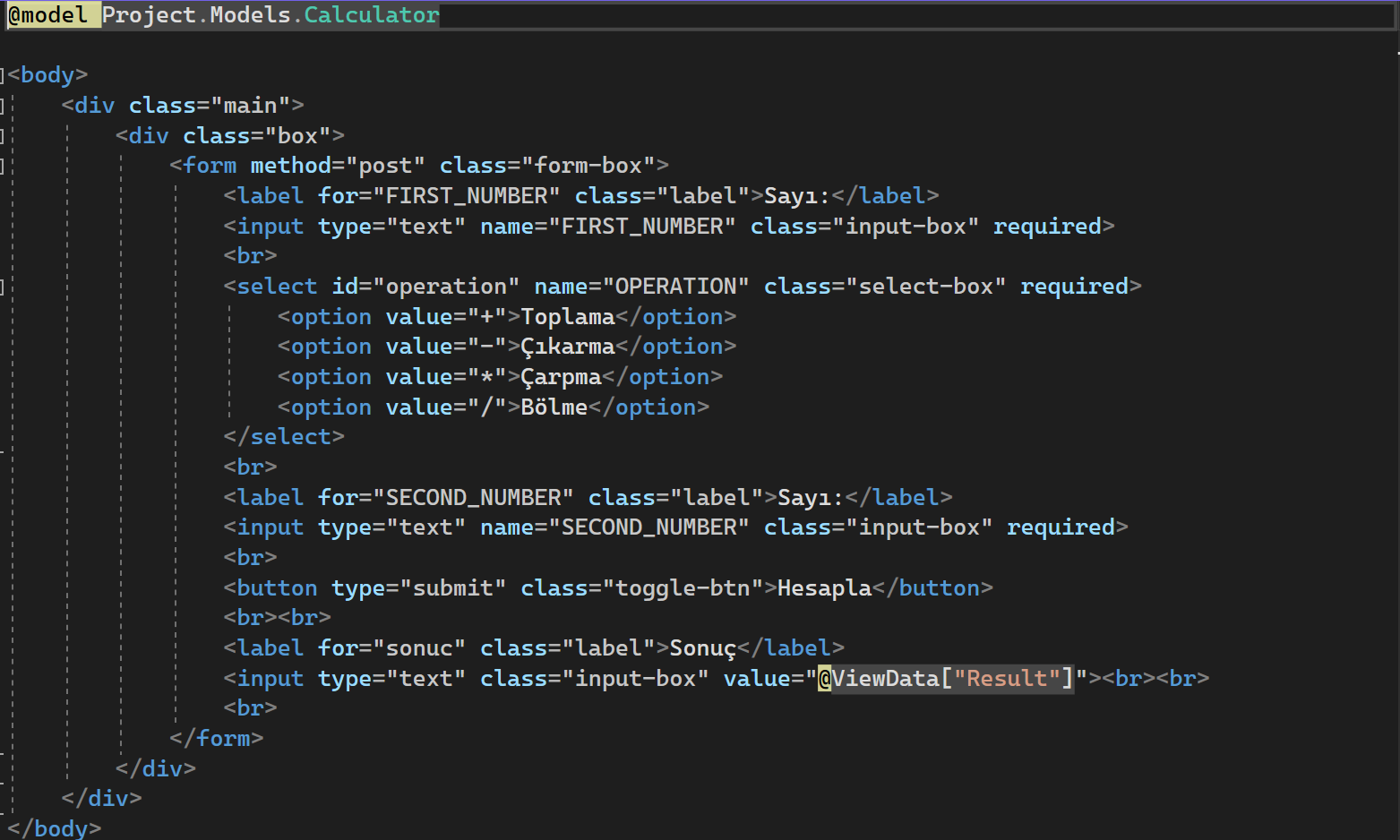
"CurrentUser" sınıfı, mevcut kullanıcının bilgilerini depolamak için kullanılır. Bu sınıf, kullanıcının e-posta adresini (Email) ve kimlik numarasını (UserID) içerir.



Model klasöründe bulunan "Calculator" sınıfı, hesap makinesinde kullanılacak verileri temsil eden bir modeldir. Bu sınıf, "FIRST\_NUMBER", "OPERATION", "SECOND\_NUMBER" ve "RESULT" özelliklerini içerir.

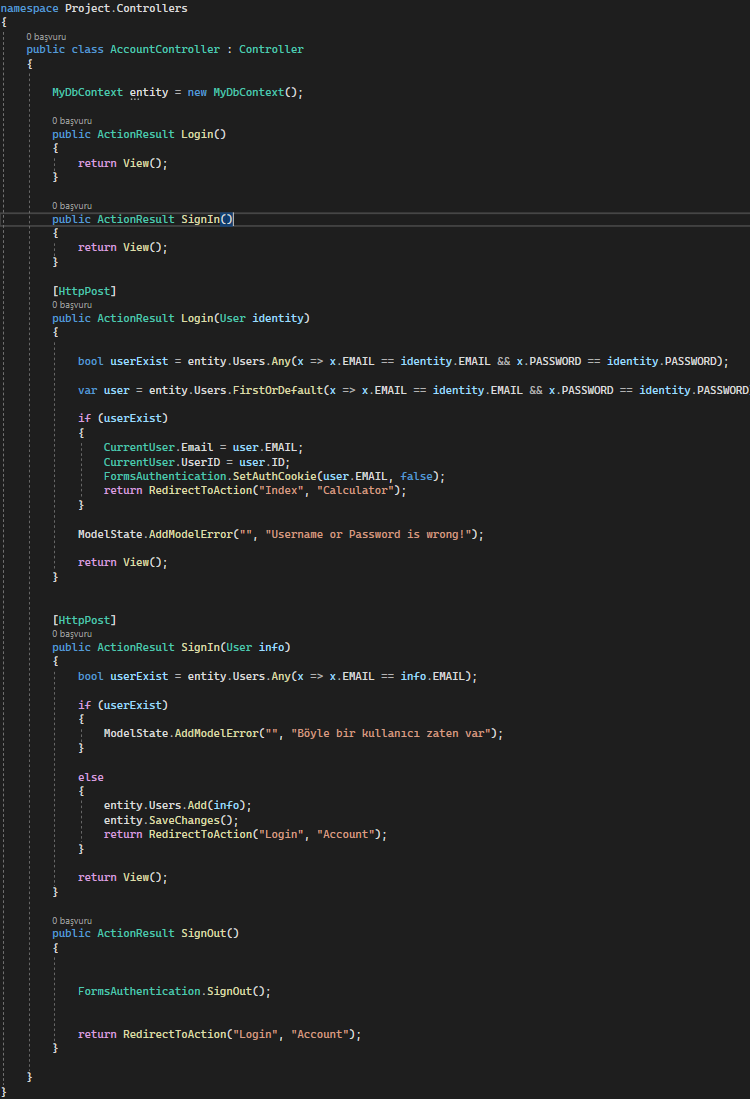


Controller klasöründe bulunan "CalculatorController" sınıfı, hesap makinesi işlemlerini gerçekleştiren kontrolcüdür. Bu sınıf, kullanıcının hesaplama işlemini gerçekleştirebileceği bir form gösterir. HTTP GET isteğiyle çağrıldığında, "Index" adlı bir görünümü döndürür. HTTP POST isteğiyle çağrıldığında, formdan gelen verileri alır, belirtilen işlemi gerçekleştirir ve sonucu görünüme gönderir. Ayrıca, sonucu veritabanına kaydeder.



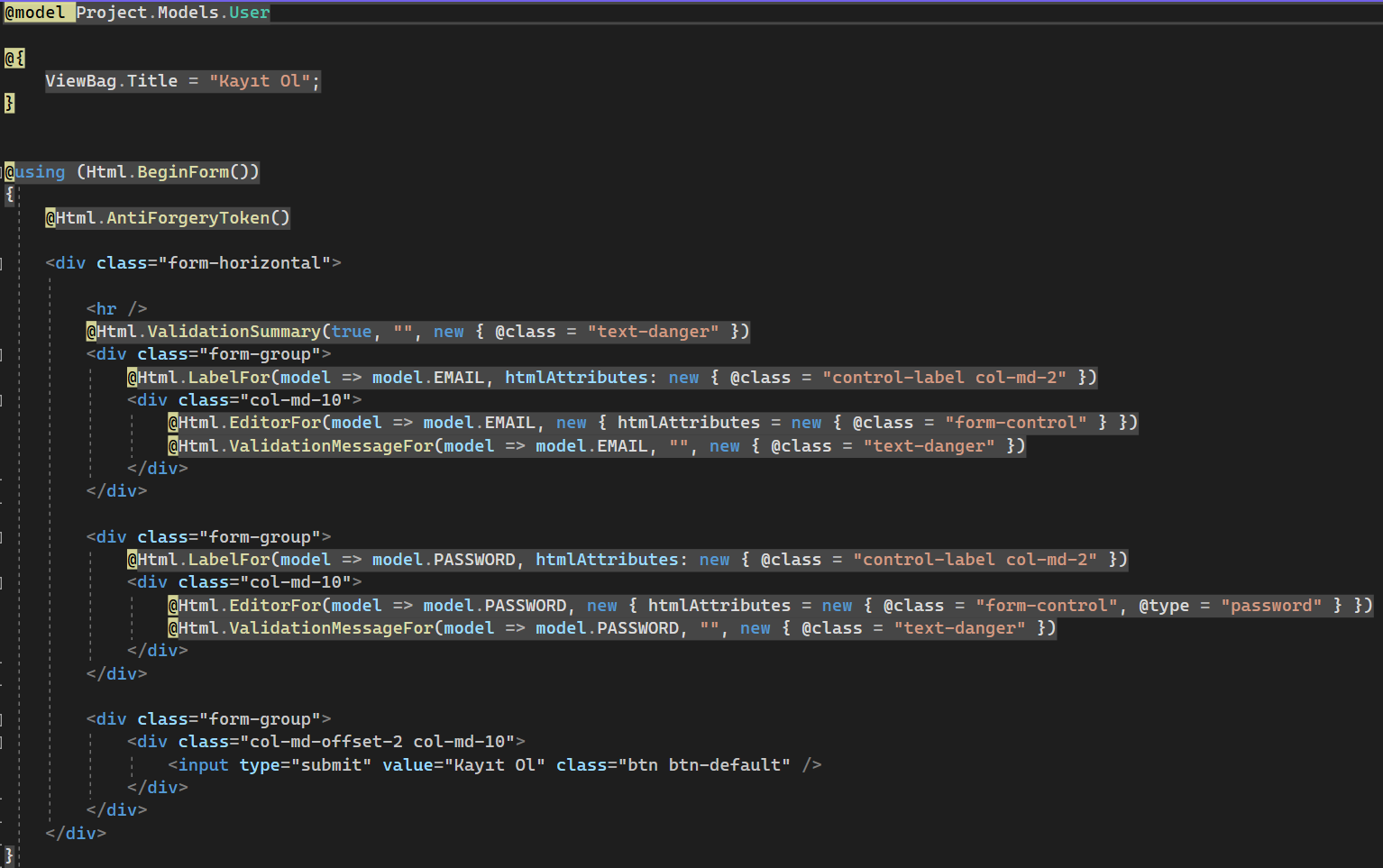
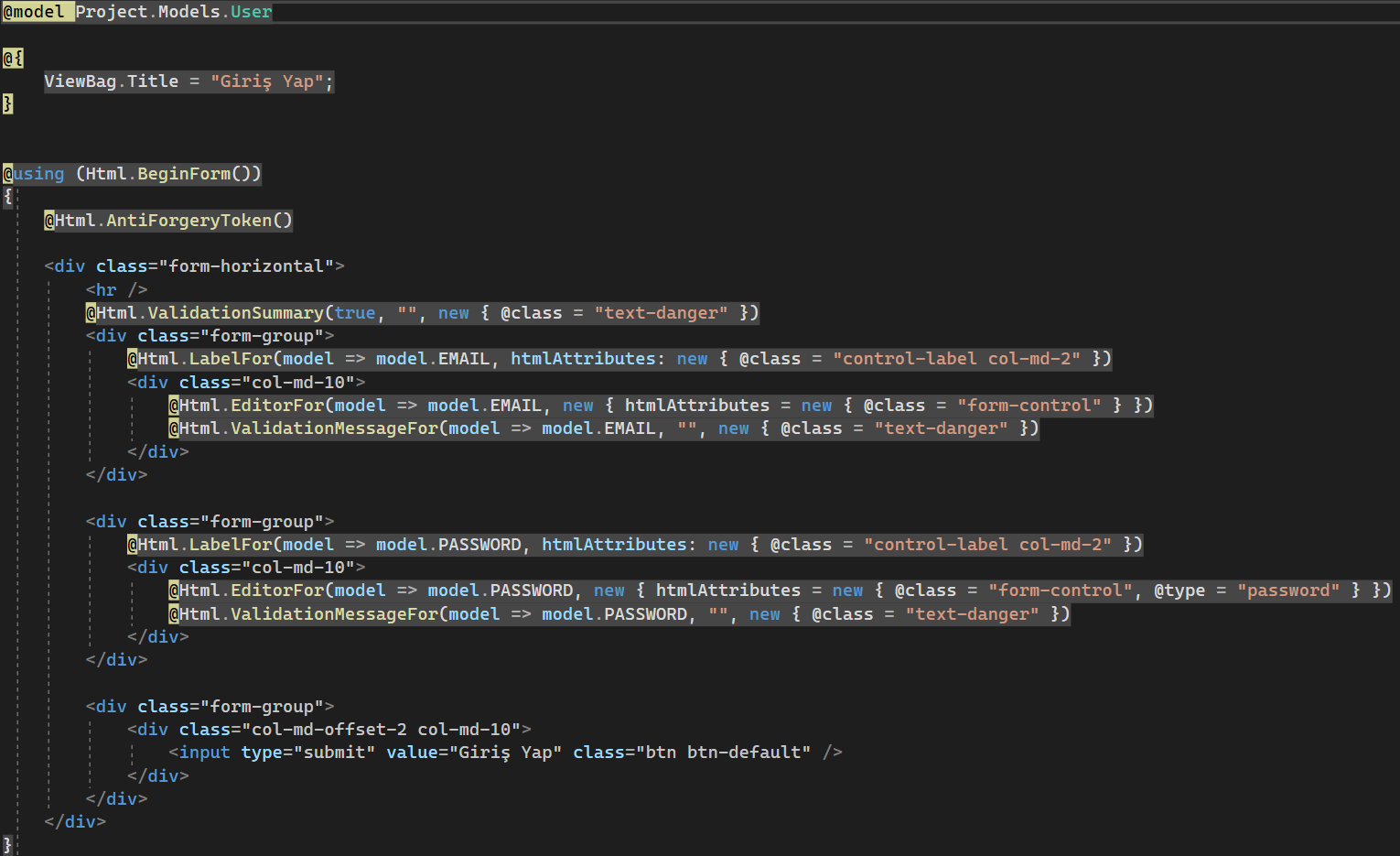
View klasöründe bulunan "CalculatorController" sınıfının oluşturduğu kodlar, kullanıcıya hesap makinesi formunu gösteren bir HTML formunu içerir. Bu formda, kullanıcı ilk sayıyı, işlemi ve ikinci sayıyı seçebilir. Hesaplama butonuna tıklandığında, form verileri POST isteğiyle sunucuya gönderilir ve sonuç görünümüne yönlendirilir. Sonuç, kullanıcıya görüntülenir.

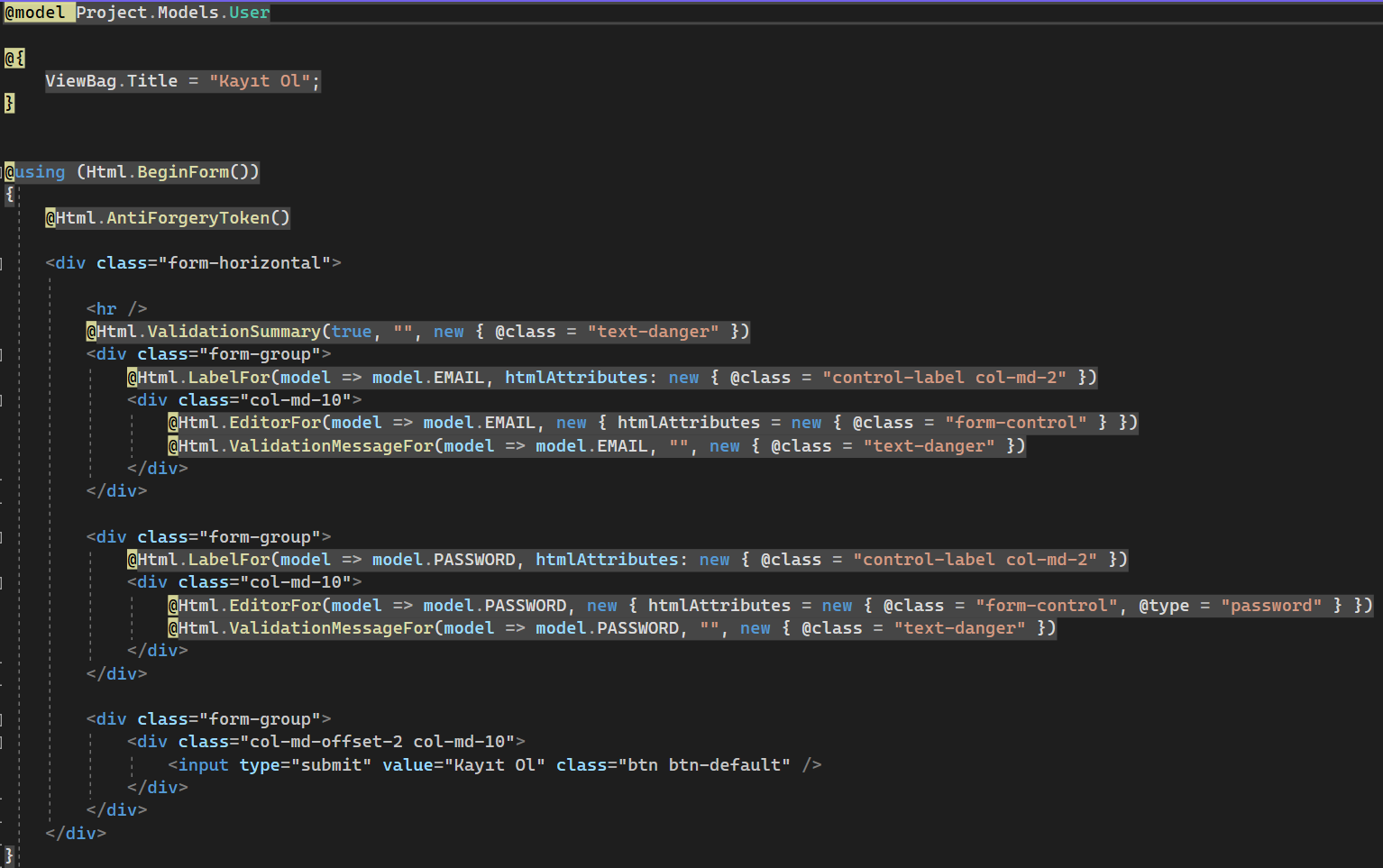
Bu kodlar, ASP.NET MVC Framework kullanarak basit bir hesap makinesi oluşturmayı sağlar. Kullanıcı, form aracılığıyla sayıları ve işlemi seçerek hesaplama işlemini gerçekleştirebilir ve sonucu görüntüleyebilir. Ayrıca, hesaplama sonuçları veritabanına kaydedilir.

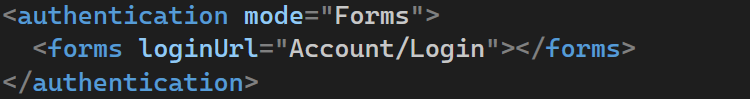


"AccountController" sınıfı, kullanıcı girişi, kaydı ve çıkışı işlemlerini gerçekleştiren bir kontrolcüdür. Login ve SignIn adında iki görünüm döndüren metotlar içerir. HTTP POST isteğiyle çağrıldığında, kullanıcının giriş yapmasını sağlar ve kimlik bilgilerini doğrular. Eğer kullanıcı bilgileri doğru ise, oturum kimliğini ayarlar, mevcut kullanıcıyı depolar ve "Calculator" kontrolcüsünün "Index" metotuna yönlendirir. Kayıt işlemi için ise kullanıcının veritabanında olup olmadığını kontrol eder ve kullanıcıyı kaydeder. Ayrıca, çıkış işlemini gerçekleştirir ve oturumu sonlandırır.

"Login" ve "SignIn" görünümleri, kullanıcı girişi ve kaydı için HTML formunu içerir. Bu formlar, kullanıcının e-posta adresini ve şifresini girmesine olanak tanır. Giriş yapma ve kayıt olma düğmeleri bulunur.







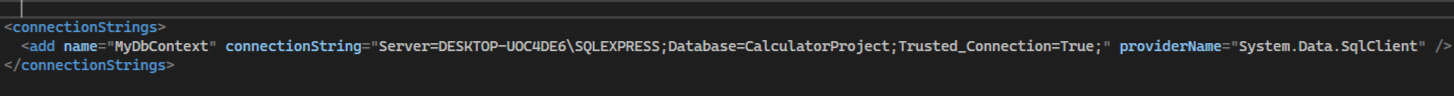
Bu kod, ASP.NET MVC Framework projesinin Web.config dosyasında bulunan bir yapılandırma ayarıdır. Bu kod, web uygulamasında form tabanlı kimlik doğrulama modunu etkinleştirir ve kimlik doğrulama için kullanılacak giriş sayfasının URL'sini belirtir.

Aşağıda kodun ne işe yaradığını açıklıyorum:

<authentication mode="Forms">: Bu ayar, Forms kimlik doğrulama modunu etkinleştirir. Forms kimlik doğrulama modu, kullanıcıların giriş yapmasını gerektiren sayfalara erişimi kontrol etmek için kullanılır.

<forms loginUrl="Account/Login"></forms>: Bu ayar, kimlik doğrulama başarısız olduğunda yönlendirilecek giriş sayfasının URL'sini belirtir. Bu örnekte, "Account" kontrolcüsündeki "Login" metodu kullanılarak giriş yapma işlemi gerçekleştirileceği için loginUrl değeri "Account/Login" olarak belirtilmiştir. Yani, kullanıcı kimlik doğrulama gerektiren bir sayfaya erişmeye çalıştığında, otomatik olarak "Account/Login" sayfasına yönlendirilir.

Bu yapılandırma ayarı, projenin kimlik doğrulama ayarlarını belirler ve kullanıcılara giriş yapma gerekliliği sağlar. Eğer kullanıcı kimlik doğrulama gerektiren bir sayfaya erişmeye çalışırsa, sistem otomatik olarak belirtilen giriş sayfasına yönlendirilir. Bu sayede, kullanıcıların oturum açmadan erişemeyecekleri sayfaları korumak ve kimlik doğrulamasını sağlamak için kullanılır.



Bu kod, ASP.NET MVC Framework projesinin Web.config dosyasında bulunan bir yapılandırma ayarıdır. Bu kod, veritabanı bağlantı dizesini belirtmek için kullanılır.

<connectionStrings>: Bu etiket, Web.config dosyasında veritabanı bağlantı dizesi yapılandırmasını tanımlamak için kullanılır.

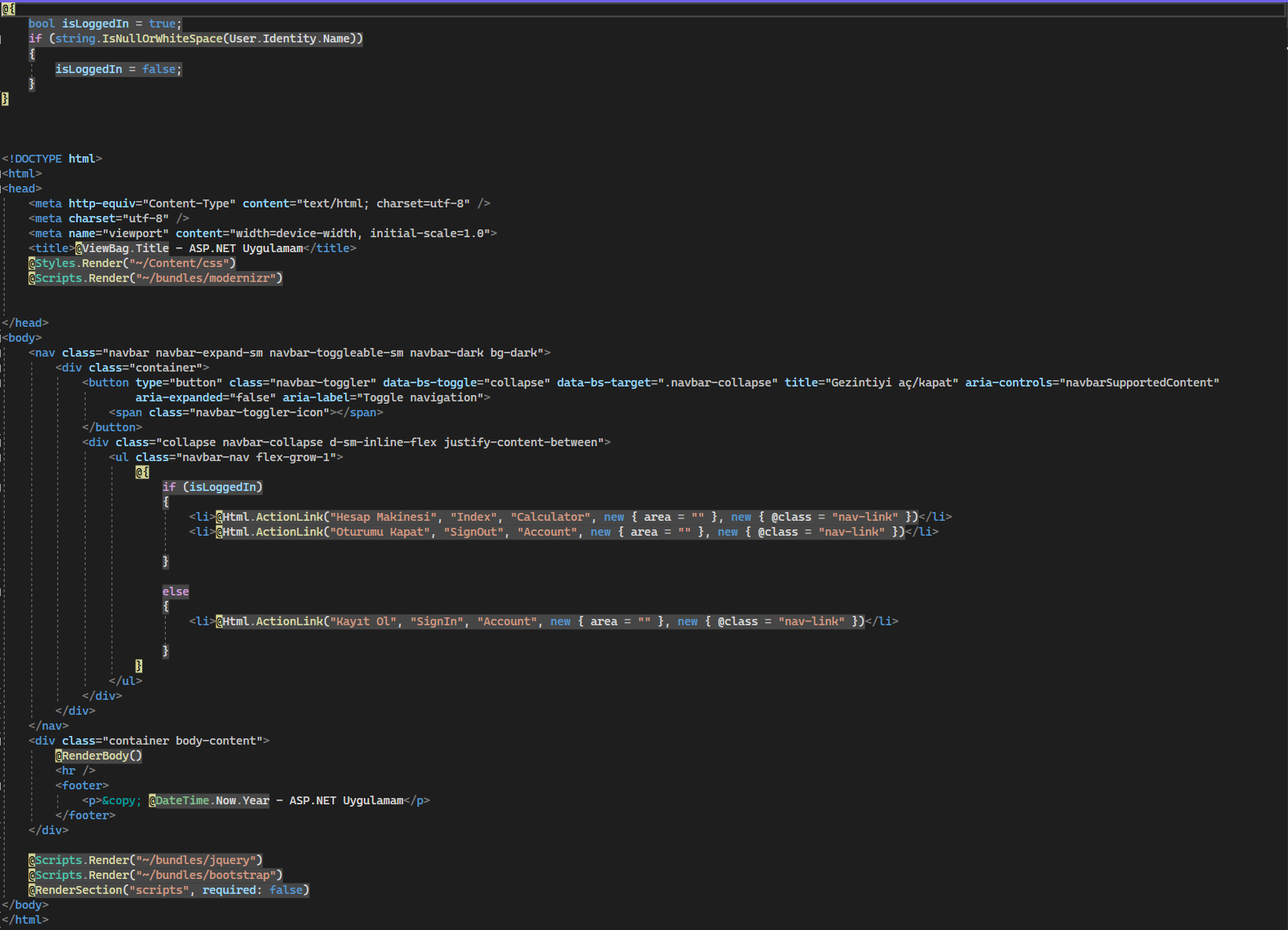
<add name="MyDbContext" connectionString="Server=;Database=;Trusted\_Connection=;" providerName="System.Data.SqlClient" />: Bu etiket, bir veritabanı bağlantı dizesi tanımıdır. Bu tanım, "MyDbContext" adında bir bağlantı adı kullanılarak bir SQL Server veritabanına bağlanmayı belirtir.

name="MyDbContext": Bu özellik, bağlantı dizesi tanımının adını belirtir. Bu ad, projede veritabanı bağlantısı yapılırken bu bağlantı dizesinin kullanılacağını belirtmek için kullanılabilir.

connectionString="": Bu özellik, gerçek veritabanına bağlanmak için kullanılacak bağlantı dizesini belirtir. Bu örnekte, SQL Server'a "DESKTOP-UOC4DE6\SQLEXPRESS" sunucusunda bulunan "CalculatorProject" adlı veritabanına güvenilen bağlantı ile bağlanılacağı belirtilmiştir. Bağlantı dizesi, sunucu adı, veritabanı adı ve bağlantı güvenliği ayarlarını içerir.

providerName="System.Data.SqlClient": Bu özellik, veritabanı sağlayıcısını belirtir. Bu örnekte, SQL Server veritabanı sağlayıcısı olan "System.Data.SqlClient" kullanılmaktadır.

Bu yapılandırma ayarı, projenin veritabanı bağlantısını tanımlar. Bağlantı dizesi, projenin hangi veritabanına bağlanacağını ve nasıl bağlanacağını belirtir. Bu sayede, projede veritabanı işlemleri gerçekleştirilirken ilgili veritabanına doğru şekilde bağlanılır ve kullanıcılar veritabanı ile etkileşimde bulunabilir.



Bu kod, ASP.NET MVC Framework projesinin Shared klasöründe bulunan \_Layout.cshtml dosyasının içeriğidir. Bu dosya, projenin genel görünümünü ve düzenini belirlemek için kullanılır.

bool isLoggedIn = true;: isLoggedIn adında bir boolean değişken tanımlanır ve başlangıçta true olarak atanır.

if (string.IsNullOrWhiteSpace(User.Identity.Name)): Kullanıcının kimliği adı boş veya boşluklardan oluşuyorsa (yani kullanıcı giriş yapmamışsa), isLoggedIn değeri false olarak değiştirilir.

@ViewBag.Title - ASP.NET Uygulamam: HTML başlığı olarak ViewBag üzerinden gelen Title değeri kullanılır. Başlık, "ASP.NET Uygulamam" metniyle birleştirilir.

@Styles.Render("~/Content/css"): Belirtilen CSS dosyalarını projede kullanmak için dahil eder. "~/Content/css" yolunda yer alan CSS dosyalarının projede kullanılmasını sağlar.

@Scripts.Render("~/bundles/modernizr"): Belirtilen modernizr dosyalarını projede kullanmak için dahil eder. "~/bundles/modernizr" yolunda yer alan modernizr dosyalarının projede kullanılmasını sağlar.

<nav>: Projenin üstündeki gezinme çubuğunu tanımlar. Bu çubuk, projedeki farklı sayfalara veya işlevlere yönlendiren bağlantıları içerir.

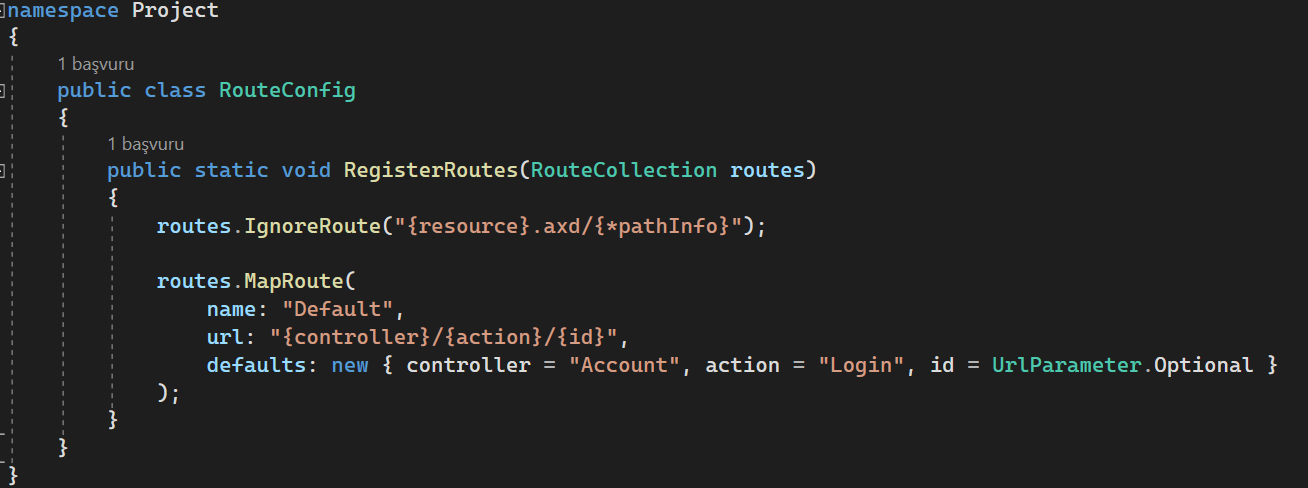
@{ ... }: Razor syntax kullanılarak C# kodu yazılır. Bu kod parçası, kullanıcının giriş yapmış olup olmadığını kontrol eder ve buna göre farklı bağlantıları oluşturur. Kullanıcı giriş yapmışsa, "Hesap Makinesi" ve "Oturumu Kapat" bağlantılarını gösterir. Kullanıcı giriş yapmamışsa, "Kayıt Ol" bağlantısını gösterir.

@RenderBody(): Sayfanın içeriğini temsil eden RenderBody metodu çağrılır. Bu kısım, projenin diğer sayfalarından gelen içeriğin yerleştirileceği alandır.

<footer>: Sayfanın altında yer alan bir footer alanıdır. Genellikle telif hakkı bilgileri veya diğer metinler için kullanılır.

@Scripts.Render("~/bundles/jquery") ve @Scripts.Render("~/bundles/bootstrap"): Belirtilen jQuery ve Bootstrap dosyalarını projede kullanmak için dahil eder. "/bundles/jquery" ve "/bundles/bootstrap" yolunda yer alan dosyaların projede kullanılmasını sağlar.

@RenderSection("scripts", required: false): İsteğe bağlı olarak "scripts" bölümünü Render eder. Bu bölüm, projenin diğer sayfalarından gelen ek script dosyalarının yerleştirileceği alandır. Gerekli olmadığı belirtilmiştir.



Bu kod, bir ASP.NET MVC Framework projesindeki App\_Start klasörü altındaki RouteConfig.cs dosyasını temsil eder. Bu dosya, uygulamanın routing ayarlarını yapılandırır.

Bu satır, bir varsayılan yönlendirme kuralı tanımlar.

MapRoute yöntemi, URL'leri belirli bir denetleyici (controller) ve eylem (action) ile eşleştirmek için kullanılır.

name parametresi, bu yönlendirme kuralının bir adını belirtir (genellikle kullanılmaz).

url parametresi, URL'nin nasıl yapısının eşleşeceğini belirler. {controller}, {action} ve {id} gibi yer tutucu değerler kullanılabilir.

defaults parametresi, belirli bir denetleyici ve eylemin varsayılan olarak kullanılacağını belirtir. Ayrıca, {id} parametresi opsiyonel olarak tanımlanır ve URL'de belirtilmediğinde kullanılacak değeri belirler.

Bu kod, proje için varsayılan yönlendirme kuralını belirler. Yani, gelen istekler hangi denetleyici ve eyleme yönlendirileceğini belirlemek için URL yapısını kullanır. Varsayılan olarak, Account denetleyici ve Login eylemi belirtilmiştir. Örneğin, http://www.example.com/Account/Login URL'si varsayılan olarak kullanılacaktır.