### Exercise 6: Authentication & Identity

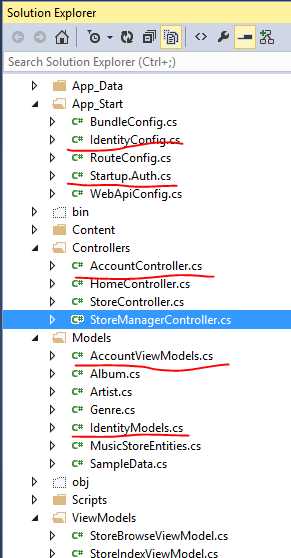
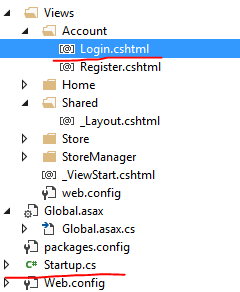
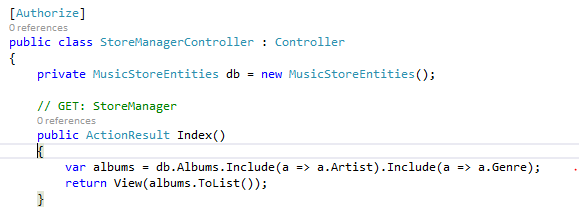
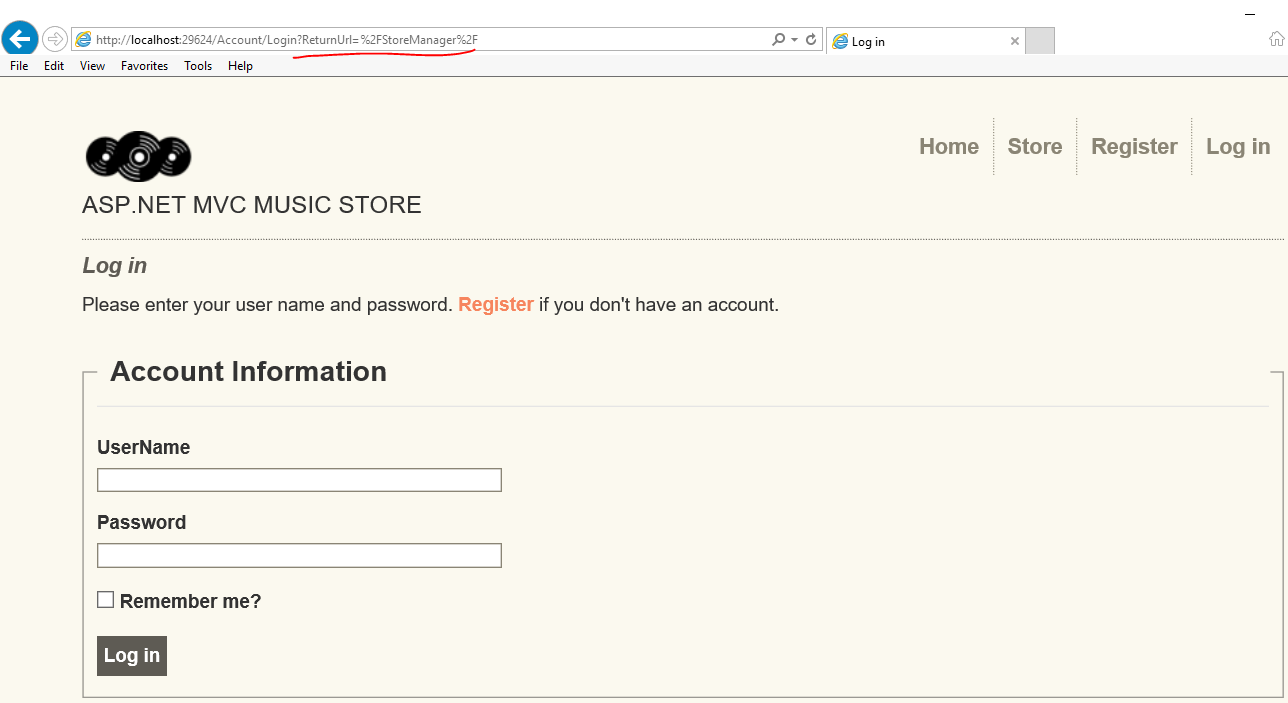
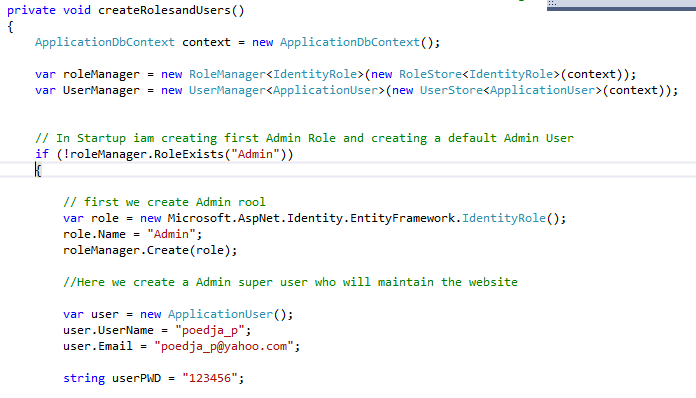
In this exercise, you will learn how to use the Code First approach to create a database with the tables of the MusicStore application, and how to access its data.

Once the model is generated, you will modify the StoreController to provide the View template with the data taken from the database, instead of using hardcoded values.

**Note:** Add these nuget package :

01) **EntityFramework** (Entity Framework)  
02) **EntityFramework.SqlServerCompact** (Entity Framework SQL Server Compact)  
03) **Microsoft.AspNet.Identity.Core** (ASP.NET Identity Core)  
04) **Microsoft.AspNet.Identity.EntityFramework** (ASP.NET Identity EntityFramework)  
05) **Microsoft.AspNet.Identity.Owin** (ASP.NET Identity Owin)  
06) **Microsoft.Owin.Host.SystemWeb** (Owin.Host.SystemWeb)

#### **Task 1 – Create Account Infrastructure**

1. Copy AccountController.cs in the Controllers directory
2. Copy AccountViewModels & IdentityModels in the Models directory
3. Copy IdentityConfig & Startup.Auth.cs in the App\_Start  
     
   
4. Copy Startup.cs in the root of the project
5. Create an Account directory inside the Views directory and copy Login.cshtml  
     
   
6. Modify your \_Layout.csthml with code snippet \_Layout.cshtml from Identity folder
7. Add “Authorize” decorator in StoreManagerController  
   
8. Run the application. Now access /StoreManager and find that you will be redirect to the Login page.  
     
   
9. Open Startup.cs and find method createRolesAndUsers below. Change username/password and any data according your preferences  
   
10. Try to login to the system with the default credential you built. Now you will be redirect to the Store Manager   
    