***Music Store – part 1***

In this exercise we will program a web shop for music albums called *Music Store*. In the web shop you can search for albums, put them in your shopping cart and then checkout and pay. We will program both the user part and the administrator part.

During this first part of the exercise we'll create the website, create the tables and set up the security for the site.

Create a new *ASP.NET Core Web App (Model-View-Controller)* in the *www* directory with the name *MusicStore*. Use *.NET 7.0 and Individual Accounts*.

You will need a number of files to develop the website. Copy the contents of the *css* and *images* folders under *wwwroot* to the new application. Existing files may be overwritten. Also copy the complete folder *content* to *wwwroot* of the application. In this folder you will find most of the album covers that are sold in the shop.

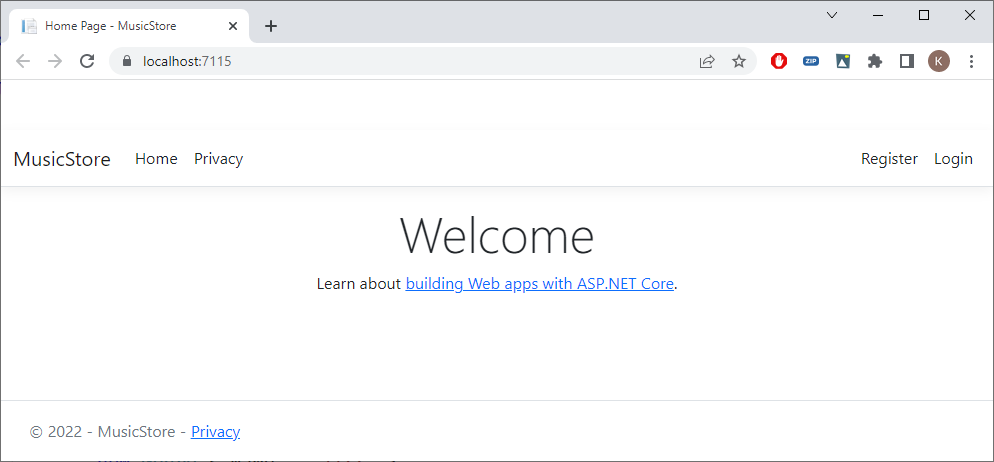
***1. Models***

In the *Models* folder, copy the three classes: *Album*, *Artist* and *Genre*. Open the classes and view their content.

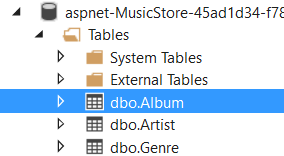
Copy the classes *StoreContext.cs* and *DBInitializer.cs* into the *Data* folder. Have a look at these classes carefully!

**Do not forget to change the Nullable attribute in your project file to annotations.**

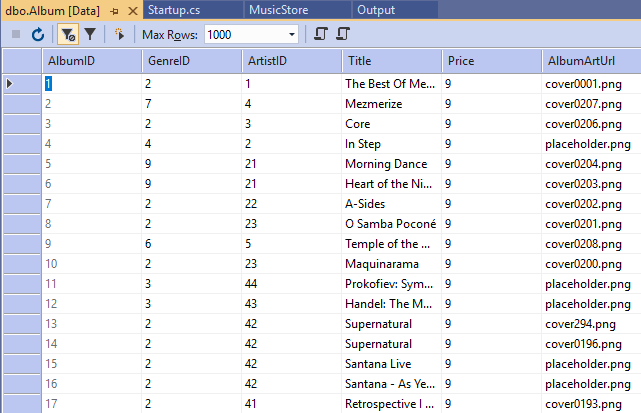
Now make your own final adjustments to your project so that on startup, the database and tables are created and the data is added. When starting the project, you get (for now):



Open the *SQL Server Object Explorer* to check whether the tables are there:



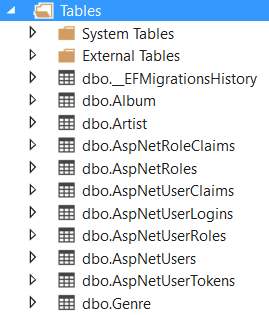
For example, the *Album* table contains:



***2. Security***

A part of our web shop should only be accessible to an *administrator*.

First apply the migrations so that the user/role tables are created:



Then create an administrator *admin@musicstore.com* with password *Mus!cst0re*. Create an *Administrator* Role and add the *admin* user to it.

***3. Admin Area***

Create a separate *Area* for all *Admin* functions. Put a controller *AlbumsController.cs* with one method *Index* in this *Area*. This method will call a view. Create this view as follows:

@{

ViewData["Title"] = "Albums Index";

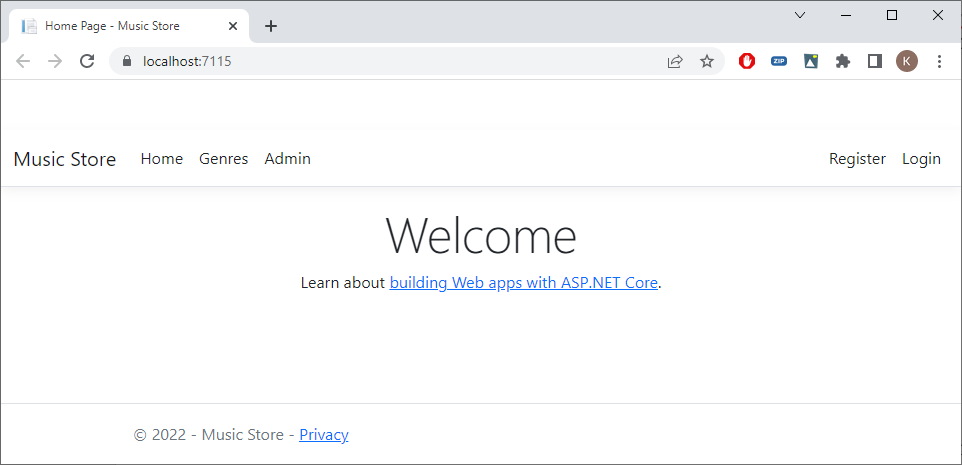
}

<h2>@ViewData["Title"]</h2>

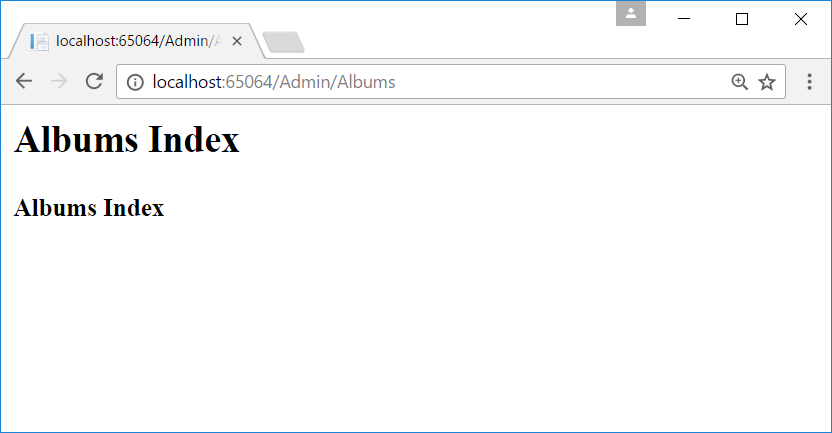
<h4>Albums Index</h4>

Make appropriate modifications for the use of *Areas*.

Adjust the layout template of the site and write *Music Store* in two words in three places. Get rid of *Privacy* in the menu. Create a menu item *Genres* which jumps to the *ListGenres* method of the *Store* Controller. Also provide a menu item *Admin* which jumps to the *Index* method of the *Albums* controller (in the *Admin* Area) from earlier:



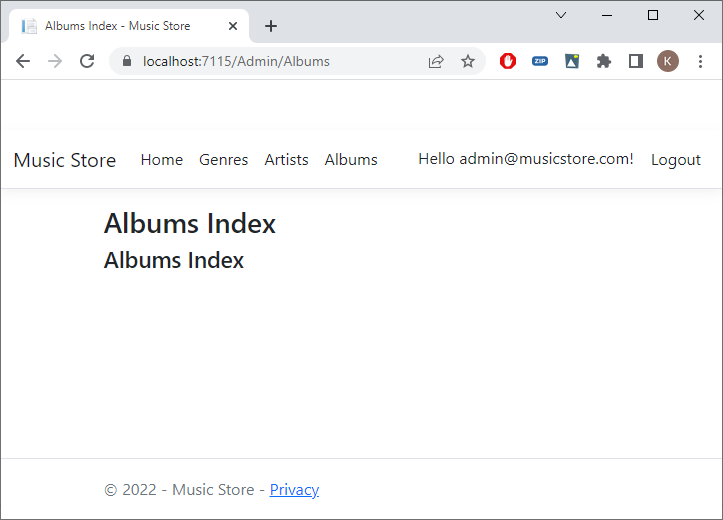
Check whether you can access the *Albums Index* page from the *Admin* menu item:



***4. Admin layout***

Make sure that another layout page *\_LayoutAdmin.cshtml* is used for the *Admin* pages. This layout pages has a menu with the menu items *Genres*, Artists and *Albums* at the top. Call the *Index* method of the (*Genres*, *Artists* and *Albums*) controllers. These controllers do not yet exist. All these controllers will later appear in the *Admin* Area. Don't forget to do the proper *Default Imports* for the *tag* helpers, ...

Try out if the *Albums Index* page has the right layout and menu:



***5. Authorization***

Make sure that only the users of the *Administrator Role* can access the *Albums* page. Create a second user and check whether or not this user can access the page.

***6. Home page***

To complete this part, we provide a customized homepage. You can just copy the code from the *resources*. The result:

