***Music Store – part 3***

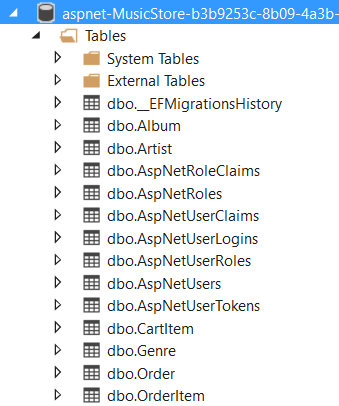
In part 3 of the exercise, we will code a shopping cart in our *Music Store*.

***1. Database Migration***

First create an initial migration script from the existing models. Add this migration to the table *EFMigrationsHistory*, so that the database is up to date.

Then, from the resources, add the classes *CartItem.cs*, *OrderItem.cs* and *Order.cs* to the *Models* folder. Copy the *StoreContext.cs* class to the *Data* folder (overwrite the previous one).

Rebuild your application and create a new migration script for the additional tables. Run that migration script so that the new tables are added to your database. Check in the database if the new tables are indeed there:



Check the content of *EFMigrationsHistory*.

***2. Sessions***

Add the class for the shopping cart *ShoppingCart.cs* to the *Model* folder.

The business logic class of the shopping cart uses session variables. Configure your *Music Store* project to use sessions.

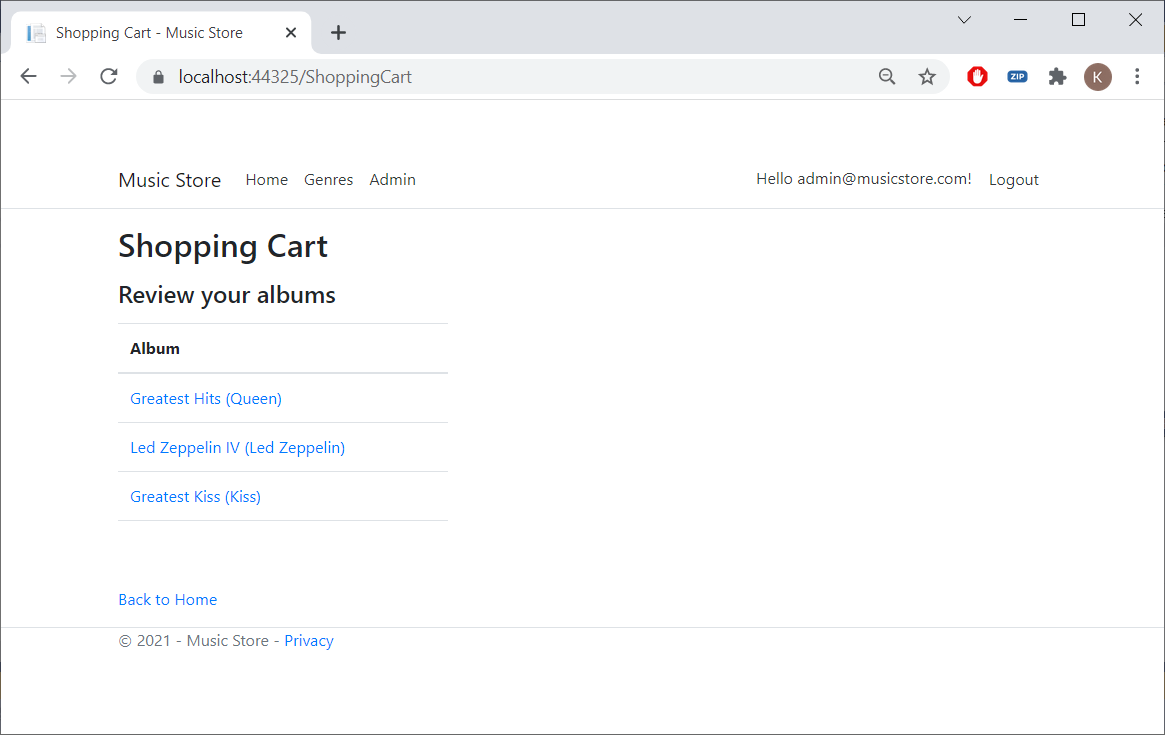
***3. ShoppingCartController***

Everything is now ready to program the shopping cart in our *Music Store* application. First create a new controller *ShoppingCartController.cs*. Create the usual constructor for a controller.

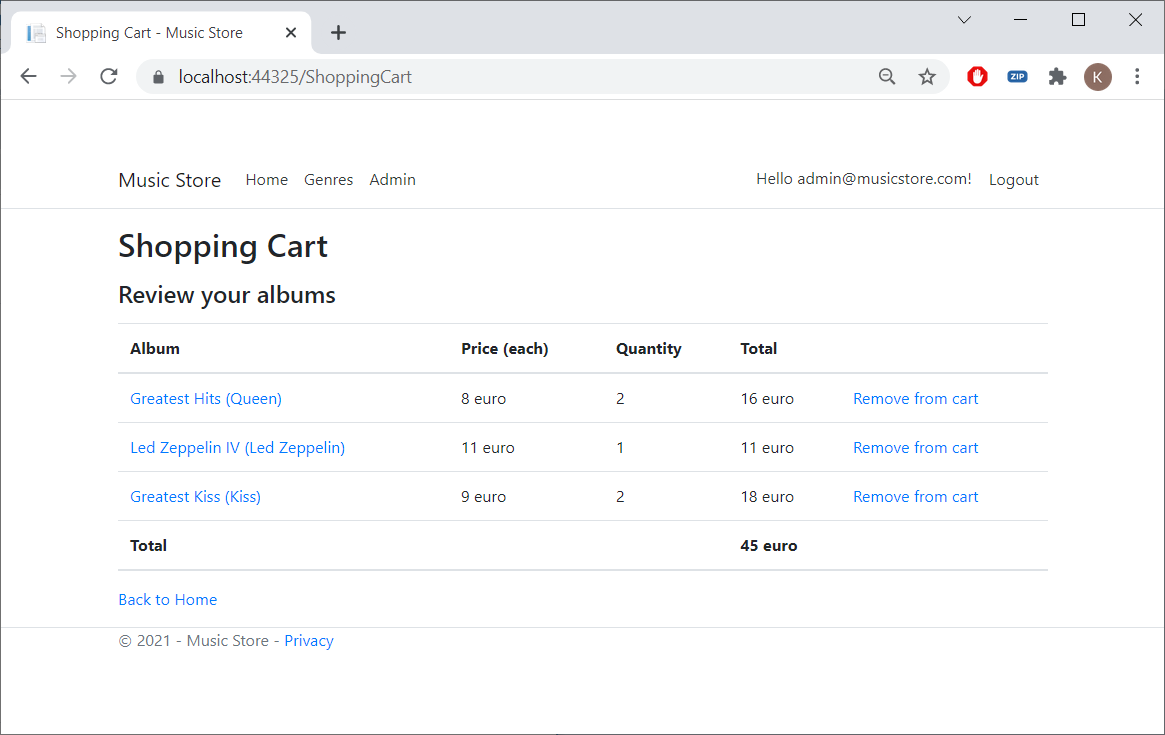
Then program the *Index* method of that controller. Get all *cartItems* with the business logic class and pass it on to the view. In the view, print all titles of albums in the cart. You cannot test this method yet as there are no albums in your cart.

***4. AddToCart & RemoveFromCart***

Program the *AddToCart* method in the *ShoppingCartController*. The method has one parameter, namely the *id* of the album to be added to the shopping cart. Retrieve the album in the method and add it to the cart (using the business logic class). Then call the *Index* method to display the shopping cart. After adding three albums to the cart, for example:



Now modify the view so that the shopping cart looks more attractive:

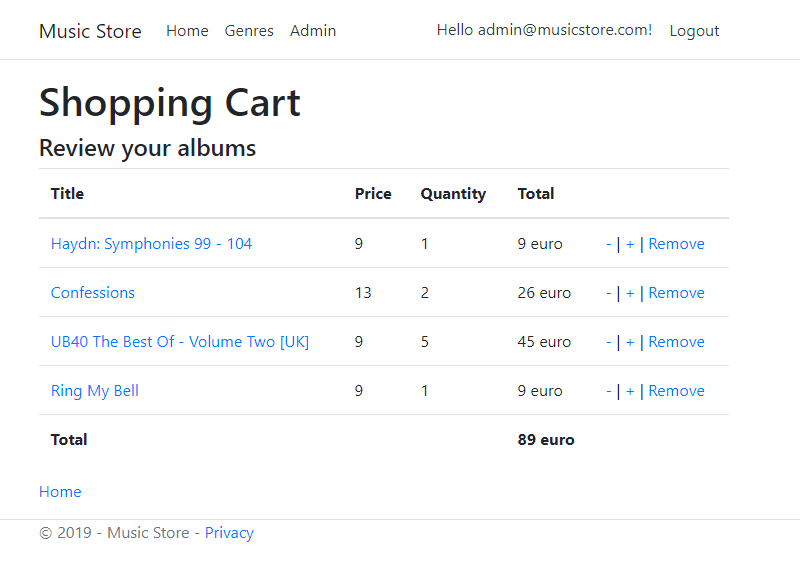


The albums are represented by a hyperlink. If you click on it, you will jump to the details of the album. There is also a hyperlink (*Remove from cart*) to remove an album from the cart. Clicking it will call the *RemoveFromCart* method in the *ShoppingCart* controller (pass the *CartItemId* as parameter). You have to program this method yourself (in the business logic class). After removing, show the modified cart. Try it out!

Note that programming this controller was very easy once the business logic class was there.

***5. Extra methods***

Adjust your shopping cart further so you get the following result:



Using the minus sign, you subtract 1 from the number, with plus you add 1. *Remove* removes the complete line.

Use the same structure as in the rest of the exercise. Create additional methods in the *ShoppingCart* class and call them from within your controller. In this way your controller will remain neat and tidy.