**[Assignment #2](https://confederation.blackboard.com/webapps/assignment/uploadAssignment?content_id=_1243217_1&course_id=_73394_1&group_id=&mode=view)**

Use MVC and .NET 5.0 to create a simple record store inventory application.

The application will contain **three data models Band, Album, and Genre**. The Band model should have a BandName and DateFormed properties.

The Album model should have AlbumTitle and ReleaseDate properties. The Genre model should have a GenreName property. Any keys, navigation properties, or join tables that are required should also be added to the data models. **Use data annotations on the models to enforce data types and validation rules.**

The data models are related as follows: **A Band can have many Albums**, **an Album belongs to one Band**. **An Album can have many Genres**, and **Genres can belong to many Albums**. **Albums to Genres is a many-to-many relationship and will require a join table and configuration using FluentAPI in the DbContext**. **For help with configuring the data model relationships, see the learning resources under "Week 6 - Razor Pages 4" and the lecture from week 13 "MVC Part 3**". Use Entity Framework and code-first Migrations to setup the database schema, as we did in class.

**Use the Scaffolder tool to generate Controllers and View templates for the Band, Album, and Genre data models.** **The scaffolder should create action methods and views for the Create, Delete, Details, Edit, and Index actions for each of the three data models.**

Update the Album Create, and Edit action methods and views to allow for multiple Genres to be added to an album. **Use a ViewModel class to accomplish this. The ViewModel will contain MultiSelectList data and an array of integers for storing selected Genres.** **Refer to the lecture from week 13 "MVC Part 3" for more information on using ViewModels.**

The **Band Details** page should display a list of Albums by the band in a table. The **Album Details** page should display the list of Genres for the album. The application **home page should** have the deafult text removed and **have a menu created using HTML headings, ul, li, and anchor elements, and styled using Bootstrap CSS classes**. We covered this in one of our RazorPages lectures.

You will be graded on how well the application works, by meeting all of the requirements listed above, and by overall presentation of the application's UI and code. Submit your project by zipping your project folder and attaching the zip file in Blackboard. Exclude the "bin" folder from the solution as it contains binaries that will make your zip file large.

Due Date: December 17th, 2021.

Note: Late submissions will receive a 10% penalty for each week past the due date they are late.