

CSI-250

Midterm Exam—Spring 2023 – 250 Points

Due 5/12/2023

You have been provided a starting project. It is a .Net 6 MVC Application. It is located in the Spring23\_Midterm folder of the repository. The Album and Publisher Models have been created as well as the IAlbumList, AlbumList, IPublisherList and Publisher list. These have been registered with Services container and are ready to be injected.

1. **Create an AlbumController** with the following Actions and Views **(100 pts)**
   1. Use Dependency Injection to add the AlbumList and the Publisher list to the Controller – **10 points**
   2. **Index**() endpoint – Shows each of the albums info in a View – **20 points**
      1. Provide links to the following actions for each album.
      2. **Edit, Details, Delete**
      3. Add a select to the top of the view with each distinct price that will filter by price when the user chooses a price.
      4. Include the Name of the publisher for each album on the Index View
      5. Add a select to the top of the view with each distinct publisher that will filter by publisher when the user chooses a publisher.
   3. **Details**(int Id) – Show the information of one album – **20 points**
      1. On the details view provide access to **Edit**, **Delete** and **back to List (index)**
      2. Include the Publisher Name and City on the Album/Details view
   4. **Edit**(int id) – Allows the information about the album to be changed **GET** and **POST** **– 20 points**
      1. **Do not allow the user to change the publisher.**
      2. Use Client and Server-side validation to check any data that posts back from the view.
   5. **Create**() – Allows the user to create a new album **GET** and **POST** **– 20 points**
      1. In the create view pass a list of all the publishers and allow the user to choose the publisher from a select when creating a new album.
      2. Use Client and Server-side validation to check any data that posts back from the view.
   6. **AlbumsByPublisher**(int publisherId) **– 10 points**
      1. Create an endpoint that takes a publisherId and displays all albums with a matching publisherId.
2. **Create a PublisherController (50 pts)**
   1. Use Dependency Injection to add the AlbumList and the Publisher list to the Controller – **10 points**
   2. **Index**() endpoint – Show each of the publishers information in a view **– 20 points**
      1. Add a button that will Order the publishers by Name to the top of the Index View.
      2. When a publisher is clicked on send the user to a view that shows all the Albums that publisher has published.
   3. **Details**(int id) endpoint – **20 points**
      1. On the details view for publishers show the information about the publisher
      2. Provide a link to the Album Controller AlbumsByPublisher endpoint to display all albums by a given publisher.
3. **Calculate Average Price by Publisher. (30 pts)** Your record store feels that some publishers are setting prices too high. You have been asked by your boss to compile some data to investigate this issue. You may do this in either Controller or even create a new Controller to make this calculation.
   1. Display each of the publishers’ names as well as the total number of albums they have published, total cost of the albums and average cost in an HTML View.
   2. Keep in mind that in our sample project the publishers are assigned at random so you will get different results each time you run the application.
4. **Add Links to the Album Index and Publish Index to the Navbar (20pts)**
5. **Short Answer:** In the comments of Program.CS answer the following questions – **(50 pts)**
   1. Explain the MVC Design Pattern.
   2. What is the difference between a GET request and a POST request?
   3. What is dependency injection and why is it useful?
   4. In an ASP.NET MVC project what file would you need to edit to change the default NAVBAR that comes with the project?
   5. What is the difference between client and server-side validation? Why are these important?
6. Create a commit with the message submission for midterm. Push the changes to your repository.