Assignment: MVC Concepts for .NET Developer

Duration: 1-2 hours

Instructions:

- This assignment is designed to test your knowledge and understanding of MVC (Model-View-Controller) concepts in the .NET framework.
- You are required to complete the assignment within the given time frame.
- You should not use any ready-made code from external sources such as GitHub or any other code repositories.
- The code you submit will be evaluated using plagiarism detection tools, so make sure your work is original and does not infringe on any existing code.
- You can use any appropriate resources, such as documentation or online references, to assist you in completing the tasks.
- You should provide a written explanation or code snippets for each task, demonstrating your understanding of the concepts.
- Ensure that your code is well-organized, follows best practices, and is thoroughly commented.

Please adhere to these instructions to ensure a fair evaluation of your skills and understanding of MVC concepts. Good luck!

Tasks:

1. Model Creation and Validation
    a. Create a simple model class called "Employee" with properties such as "Id" (integer), "Name" (string), "Email" (string), and "Age" (integer).
    b. Implement validation for the "Email" property to ensure it follows a valid email format.
    c. Explain the purpose and benefits of using data annotations for model validation in MVC.
2. Controller Actions and Routing
    a. Create a controller called "EmployeeController" with actions to handle the following operations:
        i. Display a list of all employees.
        ii. Add a new employee to the list.
        iii. Edit an existing employee.
        iv. Delete an employee.
    b. Define appropriate routes and action methods for each operation.
3. View Rendering and Data Binding
    a. Create views for each of the actions defined in the "EmployeeController."
    b. Demonstrate how to bind data from the model to the views and vice versa using Razor syntax.
4. Data Access and Persistence

   a. Implement a simple repository pattern to handle CRUD (Create, Read, Update, Delete) operations for the "Employee" model.
   b. Use Entity Framework or any other ORM of your choice to perform database operations.
5. Unit Testing
   a. Write unit tests for the controller actions in the "EmployeeController" to ensure their correctness.
   b. Use a testing framework like NUnit or xUnit for writing the tests.

Submission:

- Provide the source code for the completed tasks, including all the models, controllers, views, and tests.
- Include a README file explaining any additional details or considerations about your implementation.
- Zip the entire project folder and send it to the designated recipient or provide a link to access the code repository.