IT241 Homework 3
v1.0
Score: _____/30

The purpose of this assignment is to deepen your understanding of ASP.NET Core MVC, APIs, and shared libraries by expanding your existing wedding registration application into a multi-project solution.

Specifications

You will extend your wedding registration application by creating a new solution with three additional projects:

- ControlPanel Project (`<project_name>.ControlPanel`): An ASP.NET Core MVC application that serves as an admin panel for managing wedding registrations.
- API Project (`<project_name>.API`): An ASP.NET Core API project that handles all data operations for the wedding registrations.
- Shared Project (`<project_name>.Shared`): A class library that contains models and any shared code between the other projects.

All three new projects should be created in the same solution as your original project (`<project_name>`). The original project will remain responsible for attendee registration. The ControlPanel project will provide functionality to view and manage the registrations, while the API project will handle all data operations.

The **ControlPanel** project should include the following features:

- A page displaying a master list of all wedding registrations. The registrations should be presented in a table format with options to:
  - Create a new registration.
  - Edit an existing registration.
  - Delete an existing registration.
- Each action (create, edit, delete) must communicate with the API project to perform the corresponding operation.
- The page should be styled using CSS, and Bootstrap may be used for layout and design enhancements.

The **API** project must expose at least one of each of the following types of requests: GET, POST, PUT, and DELETE.

The API should be able to handle validation and error responses appropriately. For example, when creating or updating a registration, ensure that all required fields are present and valid.

The **Shared** project must contain the models used by both the main project, the API project, and the ControlPanel project. These models should be placed in a directory named Models.

Integration Requirements

- Refactor your original project (`<project_name>`) to use the API project for handling all data operations instead of maintaining a static in-memory list.
- The original project should use HTTP client methods to communicate with the API project for creating new registrations.

Constraints

- All constraints from the previous homework.
- All projects should reference the Shared project to ensure consistent use of models and any shared logic.
- CSS must be used to style your application; avoid in-line or in-file styles where possible.
  - Minor in-line styles are acceptable, but the majority should be in a separate CSS file.
- JavaScript can be used, but focus on your server-side functionality, not your client-side.
  - Writing good client-side scripts will not directly improve your grade in this assignment.
  - **Do not use jQuery**; use plain JavaScript instead.

Extra Credit

- (2 pts) Enhance the appearance the ControlPanel project using Bootstrap.
  - Bootstrap is included when you create an MVC application using Rider or Visual Studio.
  - If you want to use a newer version of Bootstrap for additional features, you can update the version or use the latest CDN link.
- (2 pts) Add client-side form validation using JavaScript to enhance user experience in the ControlPanel application.
- (2 pts) Implement comprehensive validation and error handling for the API endpoints, ensuring meaningful error messages are returned and displayed to the user.

Submission

The submission for this assignment is the GitHub commit ID for the commit you want graded.

Tips

- You are encouraged to start with 3-5 hard-coded wedding registrations in your API project to test your setup and endpoints.