**Goal of the project**

The goal of the project is to develop an ASP.NET MVC (Model-View-Controller) application for managing data related to a school application. Specifically, the application should allow an admin user to perform CRUD (Create, Read, Update, Delete) operations on three main tables in the database: Student, Subject, and Class.

Here's a breakdown of the key components and features your MVC application should include:

1. **Database Design:**
   * Design the database schema with three main tables: Student, Subject, and Class.
   * Establish relationships between these tables if necessary (e.g., a student may belong to a specific class).
2. **ASP.NET MVC Architecture:**
   * Implement the MVC architectural pattern to organize your code into models, views, and controllers.
   * Models should represent the data entities (e.g., Student, Subject, Class) and interact with the database.
   * Views should handle the presentation logic and user interface.
   * Controllers should manage the flow of data between models and views, handling user input and business logic.
3. **CRUD Operations:**
   * Implement Create, Read, Update, and Delete operations for each of the three entities (Student, Subject, Class).
   * Create forms for adding new records (Create), displaying existing records (Read), updating records (Update), and deleting records (Delete).
4. **User Authentication and Authorization:**
   * Implement user authentication to ensure that only authorized administrators can access and modify the data.
   * Define roles (e.g., admin) and restrict certain actions based on the user's role.
5. **User Interface:**
   * Design a user-friendly interface for the admin to interact with the application.
   * Use appropriate views and layouts to present the data in a clear and organized manner.
6. **Validation:**
   * Implement validation to ensure that only valid data is entered into the system.
   * Validate user input on both the client and server sides.
7. **Error Handling:**
   * Implement error handling to provide meaningful error messages in case of failures or incorrect input.
8. **Testing:**
   * Perform thorough testing of the application to identify and fix any bugs or issues.
   * Test the application with different scenarios to ensure its reliability and robustness.
9. **Documentation:**
   * Provide documentation for the codebase, explaining the structure, key components, and any special considerations for future development or maintenance.

By achieving these objectives, your ASP.NET MVC application should effectively manage the school's data, providing a reliable and user-friendly interface for administrators to perform necessary CRUD operations.

GitHub Link: [Bikram7852/SchoolManagementSystemInMVC (github.com)](https://github.com/Bikram7852/SchoolManagementSystemInMVC)