Assignment Title: Full-Stack Web Application for Task Management

Objective: Create a full-stack web application that allows users to manage their daily tasks. The front-end will be built using HTML, CSS, and JavaScript, while the back-end will utilize C#, .NET, and SQL Server.

Requirements:

1. **Database**:
    - Design and create a `TaskDB` SQL Server database.
    - Include a `Tasks` table with columns for `TaskId`, `Title`, `Description`, `Priority`, `DueDate`, and `Status`.
2. **Back-End**:
    - Develop a RESTful API using C# and .NET that provides endpoints for CRUD operations on tasks.
    - Implement data access using Entity Framework Core.
    - Ensure proper error handling and validation.
3. **Front-End**:
    - Build an interactive user interface using HTML, CSS, and JavaScript.
    - Use AJAX to communicate with the back-end API and perform CRUD operations.
    - Implement a responsive design using modern CSS techniques like Flexbox or Grid.
4. **Functionality**:
    - Users should be able to create, read, update, and delete tasks.
    - Include a feature to filter tasks by priority and status.
    - Add a date picker for selecting due dates.
5. **Testing**:
    - Write unit tests for the back-end to validate API functionality.
    - Perform basic cross-browser testing for the front-end.

Deliverables:

- SQL script for creating the database and tables.
- Source code for the C# .NET back-end.
- Source code for the HTML, CSS, and JavaScript front-end.
- A `README.md` file with instructions on how to set up and run the application.

Evaluation Criteria:

- Functionality: The application meets all functional requirements without errors.
- Code Quality: The code is clean, well-organized, and follows best practices.
- User Experience: The front-end is intuitive, user-friendly, and visually appealing.
- Testing: The back-end unit tests cover a wide range of scenarios and pass successfully.
- Documentation: The `README.md` provides clear setup and usage instructions.