# Building a Comprehensive Web Application

## Overview

For the rest of the semester, you will design and develop a comprehensive ASP.NET Core MVC Web Application. This project will allow you to apply and integrate various web development concepts and technologies covered in the course.

## Learning Objectives

By completing this project, you will:

1. Design and implement a complex data model.
2. Develop a full-stack web application using ASP.NET Core MVC.
3. Create and consume a RESTful Web API.
4. Implement client-side interactivity using JavaScript and AJAX.
5. Apply modern front-end frameworks to create responsive and accessible user interfaces.
6. Practice software development best practices, including version control and documentation.

## Milestones and Deliverables

### Project Proposal (10 points)

* One-page document describing your application idea.
* Preliminary class diagram.
* List of main functionalities as user stories.
* List of technologies you plan to use.
* AI Disclosure.
* **Due: Mar 6**

### User Stories (20 points)

* Detailed user stories and story flows covering all major functionalities.
* AI Disclosure.
* **Due: Mar 27**

### The Completed Application (70 points)

* Submit your GitHub clone link.
* AI Disclosure.
* **Due: May 1**

#### Data Model (5%)

* Design a database schema with at least 3 interconnected entities.
* Implement at least one many-to-many relationship.
* Use Entity Framework Core for database operations.
* Implement data validation using data annotations and server-side validations.

#### Application Functionalities (50%)

* Implement CRRUD (Create, Read-All, Read, Update, Delete) operations for all entities.
* Develop at least one complex functionality involving multiple entities.
* Implement user authentication and authorization.
* Create a dashboard (a home page) with summary statistics of your data.

#### REST Web API (15%)

* Design and implement a RESTful API.
* Include proper HTTP method usage (GET, POST, PUT, DELETE).

#### JavaScript and AJAX (15%)

* Implement at least two features using AJAX and JavaScript for asynchronous operations.

#### User Interface and UX (10%)

* Utilize a CSS framework (e.g., Bootstrap, Tailwind CSS) for a responsive design.
* Implement a custom theme or significant visual customizations.
* Ensure your application is accessible (follow WCAG 2.1 AA standards).
* Design and implement an intuitive navigation system.

#### Documentation and Best Practices (5%)

* Write comprehensive comments for controllers, service classes, and JavaScript code.
  + State the purpose of each class, each method, and functions.
  + Describe each parameter.
  + If your block of code is not self-explanatory, then write short inline comments to explain it.
* Use Git for version control, with regular commits and meaningful commit messages.

## Additional Notes

* This is an individual project. While you may discuss concepts with classmates, all code and design must be your own work.
* Cite any external sources or libraries used in your code comments.
* Regular check-ins with the instructor are encouraged to ensure you're on the right track.
* Creativity and going beyond the basic requirements are encouraged and may result in maximum points. If you are aiming for an ‘A’ then you must go beyond the basic requirements.

## AI Use

You are permitted to use AI tools for learning and understanding, but not for generating direct solutions to this assignment. The work you submit must be your own original creation, demonstrating your personal understanding and application of course materials.

### Example Acceptable Uses of AI

1. **Concept Clarification**
   1. **Acceptable**: Asking an AI to explain a programming concept like recursion in different ways until you understand it.
   2. **Not Acceptable**: Asking an AI to write a recursive function for your assignment.
2. **Debugging Assistance**
   1. **Acceptable**: Asking an AI to help identify why your code isn't working or explain a specific error message.
   2. **Not Acceptable**: Asking an AI to fix your code or write a working solution.
3. **Learning Best Practices**
   1. **Acceptable**: Asking an AI about coding conventions, documentation standards, or design patterns.
   2. **Not Acceptable**: Having an AI generate documented code for your assignment.
4. **Practice and Preparation**
   1. **Acceptable**: Using AI to generate practice problems like assignment questions.
   2. **Not Acceptable**: Using AI-generated solutions for actual assignment problems.

### Required Disclosure

If you use AI tools while working on this assignment, you **must** include a brief statement in your submission describing:

* Which AI tools do you use?
* How did you use them?
* What specific insights or assistance have you gained?

## Resources

* ASP.NET Core Documentation: <https://docs.microsoft.com/en-us/aspnet/core/>
* Entity Framework Core: <https://docs.microsoft.com/en-us/ef/core/>
* JavaScript MDN Web Docs: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
* WCAG 2.1 Guidelines: <https://www.w3.org/WAI/WCAG21/quickref/>

Good luck with your project! Remember, this is an opportunity to create something impressive for your portfolio.