Capstone Project: Building an Interactive Web Application

Objective

This capstone project is designed to allow students to demonstrate their understanding of the concepts covered in the previous lessons on JavaScript. The project will involve creating a fully functional, interactive web application that incorporates HTML, CSS, and JavaScript.

Project Description

Title: Personal Portfolio Website with Interactive Features

Overview: Students will build a personal portfolio website showcasing their skills, projects, and contact information. The website must include interactive elements such as a dynamic navigation bar, a form with validation, and DOM manipulation to enhance the user experience.

Key Features:

1. **Home Page**: An introduction with a brief bio and navigation links to different sections of the site.
2. **Portfolio Section**: A showcase of projects or works with images and descriptions. Clicking on a project should display more details dynamically.
3. **Contact Form**: A form that allows users to send messages. The form should include validation to ensure all required fields are filled out correctly.
4. **Responsive Design**: The website should be responsive and accessible on both desktop and mobile devices.
5. **Interactive Navigation Bar**: A dynamic navigation bar that highlights the current section of the page.
6. **JavaScript Interactions**: Use JavaScript to create interactivity, such as toggling content visibility, form validation, and dynamically updating content.

Project Requirements

1. **HTML Structure**:

   - Proper use of HTML5 semantic elements (`<header>` , `<footer>` , `<section>` , `<article>` , etc.).
   - Clean, well-structured code that is easy to read and maintain.

2. **CSS Styling**:

   - Consistent and aesthetically pleasing design.
   - Responsive layout using media queries.
   - Use of CSS for layout and styling, avoiding inline styles.

3. **JavaScript Functionality**:

   - Effective use of DOM manipulation to enhance the user experience.
   - Form validation to ensure data integrity before submission.
   - Implementation of interactivity such as click events, toggling content, and dynamic updates.

4. **Form Validation**:

   - The contact form must validate inputs for required fields, email format, and message length.
   - Provide appropriate feedback to users for incorrect inputs.

5. **Deployment**:
    - The final project should be deployed to a web server (e.g., GitHub Pages, Netlify).
    - Provide a link to the live site.

Submission Requirements

- **Source Code**: Submit all HTML, CSS, and JavaScript files.
- **Deployed Website**: Provide the URL to the deployed website.
- **Documentation**: Include a brief report (1-2 pages) explaining the design choices, JavaScript functions used, and any challenges faced during development.

Evaluation Criteria (Total: 100 Marks)

1. **HTML Structure and Semantics (15 Marks)**
    - Proper use of semantic HTML elements (5 Marks)
    - Well-organized and clean code (5 Marks)
    - Accessibility considerations (5 Marks)
2. **CSS Styling and Responsiveness (20 Marks)**
    - Aesthetic and consistent design (10 Marks)
    - Responsive layout for different devices (10 Marks)
3. **JavaScript Functionality (30 Marks)**
    - Effective DOM manipulation (10 Marks)
    - Interactivity such as click events and dynamic content (10 Marks)
    - Proper use of functions and events (10 Marks)
4. **Form Validation (15 Marks)**
    - Proper validation of required fields (5 Marks)
    - Effective feedback for incorrect inputs (5 Marks)
    - Overall usability and user experience (5 Marks)
5. **Deployment and Documentation (10 Marks)**
    - Successful deployment and accessible URL (5 Marks)
    - Clear and concise project documentation (5 Marks)
6. **Creativity and Originality (10 Marks)**
    - Unique and creative design elements (5 Marks)
    - Implementation of additional features beyond the requirements (5 Marks)

Rubric

Criteria	Excellent (90-100%)	Good (70-89%)	Fair (50-69%)	Poor (0-49%)
HTML Structure & Semantics	Excellent use of semantic HTML, well-organized, accessible.	Good use of semantic HTML, minor issues in organization or accessibility.	Basic use of HTML, lacks proper structure, accessibility issues.	Poorly organized HTML, no semantic elements, not accessible.
CSS Styling & Responsiveness	Aesthetic design, fully responsive, professional look.	Good design, mostly responsive, minor issues with layout.	Basic design, partially responsive, lacks polish.	Poor design, not responsive, lacks consistency.
JavaScript Functionality	Highly effective DOM manipulation, smooth interactivity.	Good use of JavaScript, some minor issues in functionality.	Basic JavaScript implementation, limited interactivity.	Little to no JavaScript functionality, many issues.
Form Validation	Comprehensive validation, excellent user feedback.	Good validation, minor issues in user feedback.	Basic validation, some issues with feedback.	Poor or no validation, poor user experience.
Deployment & Documentation	Successfully deployed, clear and detailed documentation.	Deployed, with some documentation issues.	Deployed, lacks clear documentation.	Not deployed, no documentation.
Creativity & Originality	Highly creative, unique features beyond requirements.	Some creativity, meets basic requirements.	Minimal creativity, lacks unique elements.	No creativity, basic implementation.