Final Project Guidelines

1. Project Objective

- The project aims for students to apply their skills in HTML5, CSS3, and JavaScript to create a polished, functional personal website.

2. Requirements

A. Structure & Content

- **Homepage:** Include a professional-looking homepage with the student's name, a brief introduction, and a professional photo.
- **About Me:** A section/page with details about themselves, including their background, interests, skills, or career goals.
- **Portfolio:** Showcase past projects or any work relevant to web development, with project descriptions and links to live demos or code repositories (if available).
- **Contact:** Include a contact form or links.
- **Resume (Optional):** Provide a downloadable PDF resume or a resume section.

B. Layout & Navigation

- Use semantic HTML5 tags for clear structure (e.g., `<header>`, `<section>`, `<footer>`).
- Responsive navigation with clear, intuitive links to each section.
- The site should be fully responsive, adapting smoothly to various screen sizes (mobile, tablet, and desktop).

C. Styling & Design

- Apply CSS3 styles to enhance visual appeal—use colors, typography, and whitespace effectively.
- Include CSS animations or transitions (hover effects, fade-ins, etc.).
- Use background images, gradients, or textures appropriately for visual enhancement.
- CSS Grid or Flexbox should be used for layout to demonstrate modern CSS layout techniques.

D. Interactivity with JavaScript

- **Form Validation:** Add form validation for the contact form using JavaScript.
- **Dynamic Content:** Use JavaScript to dynamically update or display content (e.g., a "Skills" progress bar or a portfolio gallery).
- **Interactive Elements:** Include at least two JavaScript-driven interactive elements, like a lightbox for images, a modal popup, or a simple JavaScript-based game or quiz.

3. Performance & Accessibility

- Ensure images are optimized for web performance (use responsive images and compression).
- Implement alt text for images and ARIA roles for improved accessibility.
- Test for fast loading times and minimize external dependencies.

4. Documentation

- Provide a brief README file with an overview of the project, setup instructions, and any special features implemented.
- Include comments in the code to explain functions and structures where applicable.

5. Grading Criteria

- **Completeness:** All required sections and elements are present and functional.
- **Design & Aesthetics:** Quality of design, style, and use of animations/transitions.
- **Functionality:** Interactivity, JavaScript functionality, and form validation.
- **Responsiveness:** Consistent and functional layout across devices.
- **Code Quality:** Clean, organized code with comments and meaningful naming.