Web Dev I Lab Assignment 1

Personal Portfolio Website

Problem Statement

Build a one-page portfolio website using semantic HTML tags such as <header>, <nav>, <section>, and <footer>, featuring distinct sections for About, Projects, and Contact. The site should include a hero section with a title and short introduction, along with a profile image in the About section that has appropriate alt text. List 2–3 projects using <ul> or <article> elements, and present your technical skills or tools in a well-structured <table>. Add a functional contact form with fields for name, email, and message, ensuring the use of label, required, and placeholder attributes for usability. Include internal navigation links for smooth movement between sections and a "Skip to main content" link to improve accessibility. Make sure all links are functional and clearly labeled, and structure your HTML cleanly to allow for easy CSS styling in future enhancements.

Objective:

Build a one-page portfolio website using semantic HTML tags, featuring sections for About, Projects, and Contact. Include a hero section, profile image, project list, skills table, and a functional contact form with proper labels, placeholders, and accessibility features.

Learning Outcomes

Upon completion of this assignment, the student will be able to:

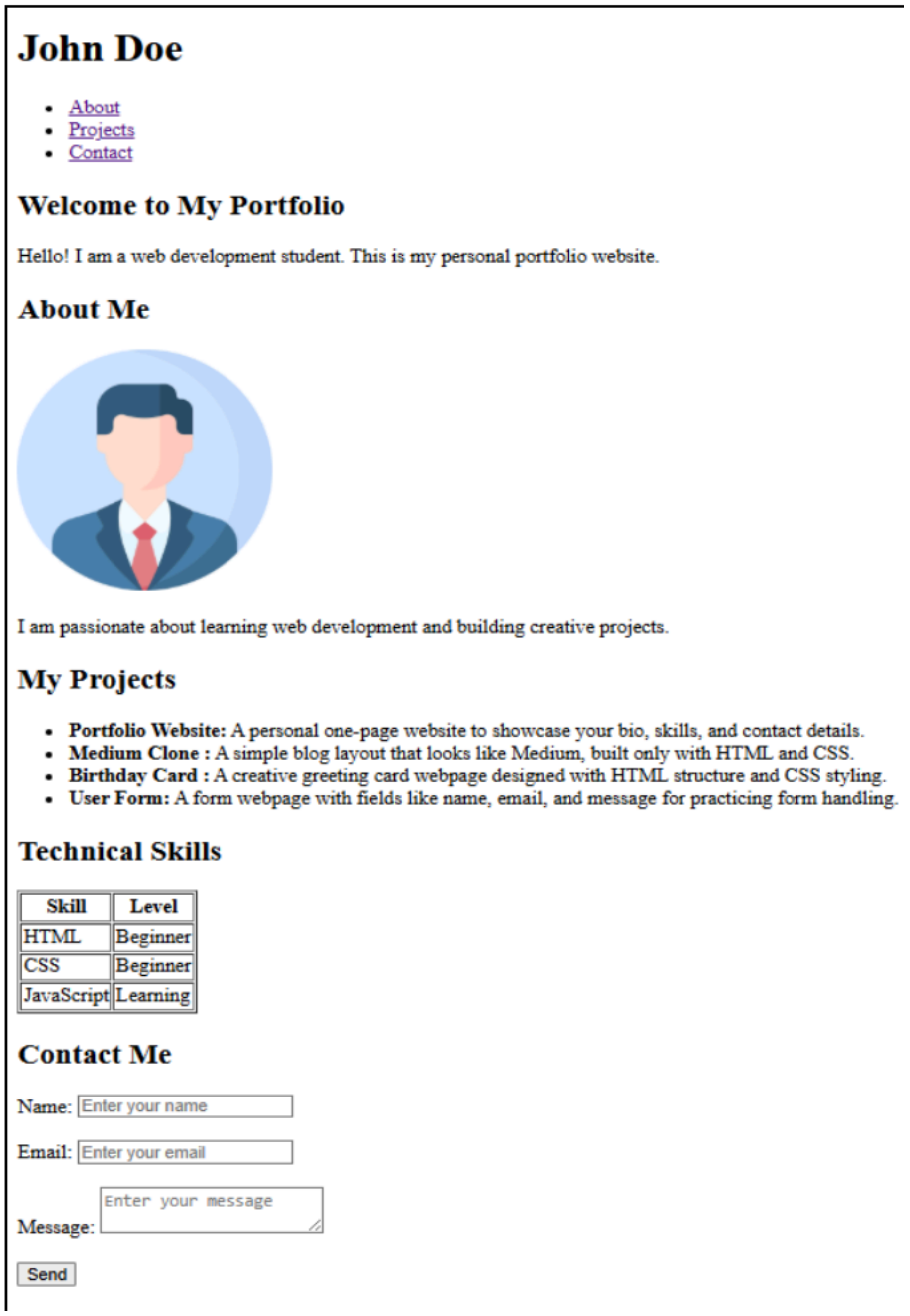
1. Learn how to use HTML tags to make a well-structured webpage.
2. Create links inside the page to move smoothly between sections.
3. Add images with proper descriptions (alt text) for better accessibility.
4. Show information using lists and tables in a clear way.
5. Make a contact form with labels, required fields, and placeholder text.

6. Write clean and organized HTML code that is easy to style later with CSS.

Detailed Instructions

1. **Create a Project Folder:**
   Make a new folder (e.g., portfolio-project) and save your HTML file inside it as index.html.
2. **Build the Structure:**
   Use semantic HTML tags – <header>, <nav>, <main>, <section>, and <footer> to organize the page properly.
3. **Navigation Bar:**
   Add a menu with links to different sections like **About**, **Projects**, and **Contact** (all links must scroll to correct sections).
4. **Hero & About Section:**
   Write a welcome message, add your profile picture (store images in an images folder), and a short bio paragraph.
5. **Projects Section:**
   List at least 2–3 sample projects with short one-line descriptions.
6. **Skills Section:**
   Create a table that lists technical skills (e.g., HTML, CSS, JavaScript) along with learning levels.
7. **Contact Section:**
   Add a form with **Name, Email, and Message** fields, including labels, placeholders, and required attributes.
8. **Validation:**
   Ensure all links and form inputs are working correctly and clearly labeled.
9. **Clean Code:**
   Write neat and well-structured HTML so it will be easy to style later with CSS.

Expected Output



John Doe

- About
- Projects
- Contact

Welcome to My Portfolio

Hello! I am a web development student. This is my personal portfolio website.

About Me

I am passionate about learning web development and building creative projects.

My Projects

- **Portfolio Website:** A personal one-page website to showcase your bio, skills, and contact details.
- **Medium Clone :** A simple blog layout that looks like Medium, built only with HTML and CSS.
- **Birthday Card :** A creative greeting card webpage designed with HTML structure and CSS styling.
- **User Form:** A form webpage with fields like name, email, and message for practicing form handling.

Technical Skills

Skill	Level
HTML	Beginner
CSS	Beginner
JavaScript	Learning

Contact Me

Name: Enter your name

Email: Enter your email

Message: Enter your message

Guidelines to Students

1. **Organize Your Work** – Always create a separate folder for each project and keep files (HTML, images, assets) properly named
2. **Follow Standards** – Write clean, semantic, and well-indented HTML so it is easy to read and update later, Ensure form fields are properly labeled, accessible, and validated.
3. **Plan Before Coding** – Decide the sections (header, about, projects, contact, etc.) on paper before starting.
4. **Focus on Basics First** – Complete the HTML structure fully before adding CSS in future labs.
5. **Self-Check** – After finishing, open your file in a browser and test if all parts are visible and links/forms work.

Improvements/Adjustments

1. Write a short comment at the top of your HTML file with section name, project name, page name.
2. Make a rough sketch in your notebook before coding to plan where the header, about, projects, and contact sections will be.
3. After finishing, open the project in a browser and test everything: check links, form fields, and alt text on images.
4. Keep file names lowercase and without spaces (e.g., index.html, profile.jpg).
5. Use clean, semantic, and properly indented HTML code for better readability.

Submission Guidelines

1. Push your index.html and image assets to a GitHub repository.
2. Ensure all links, navigation, and form fields work correctly.

3. Include a README.md describing the project structure and purpose.
4. Share the GitHub repository link with the concerned faculty.

Performance Metrics (Out of 5 Marks)

Criteria	**Marks**
Correct use of semantic tags and proper layout.	1
Internal navigation and accessible design	1
About, Projects, Contact are included with meaningful content.	1
Form has proper fields, uses validation attributes and is well-structured.	1
Code is clean, indented, with lowercase file names and no spaces.	1