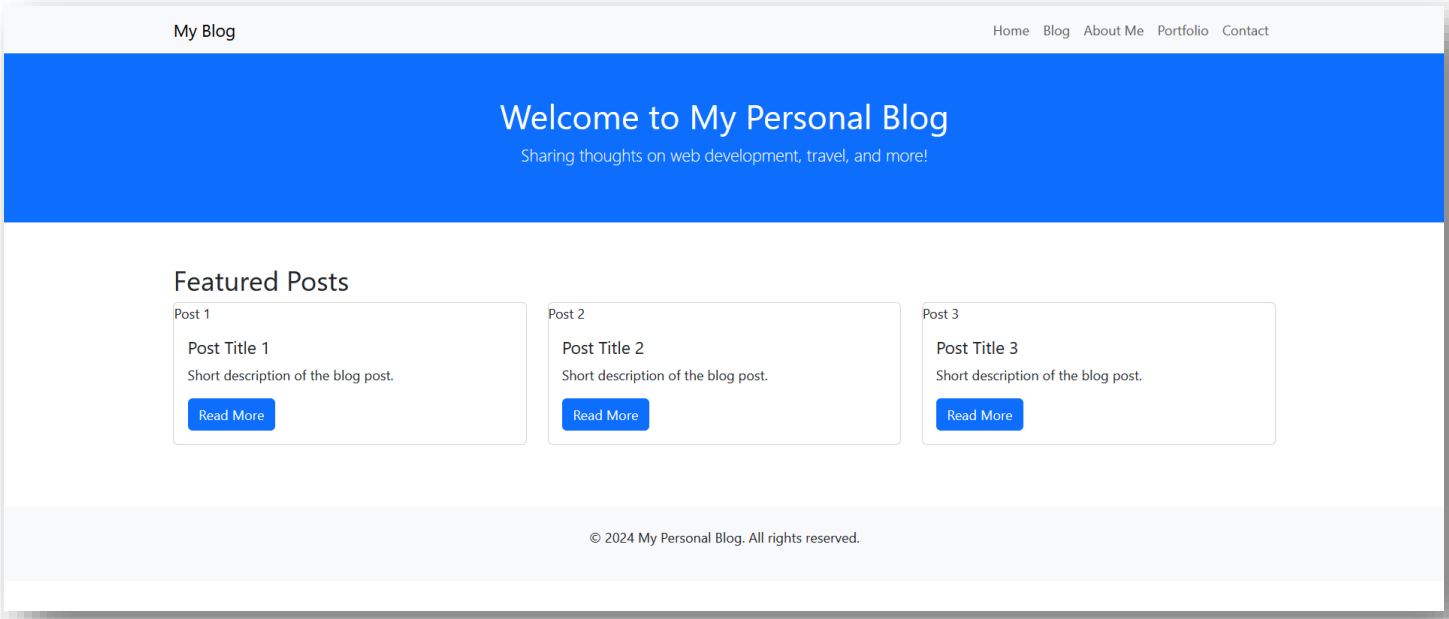
Problem 2

Add CSS to each web page using the Bootstrap framework to get the following final look:



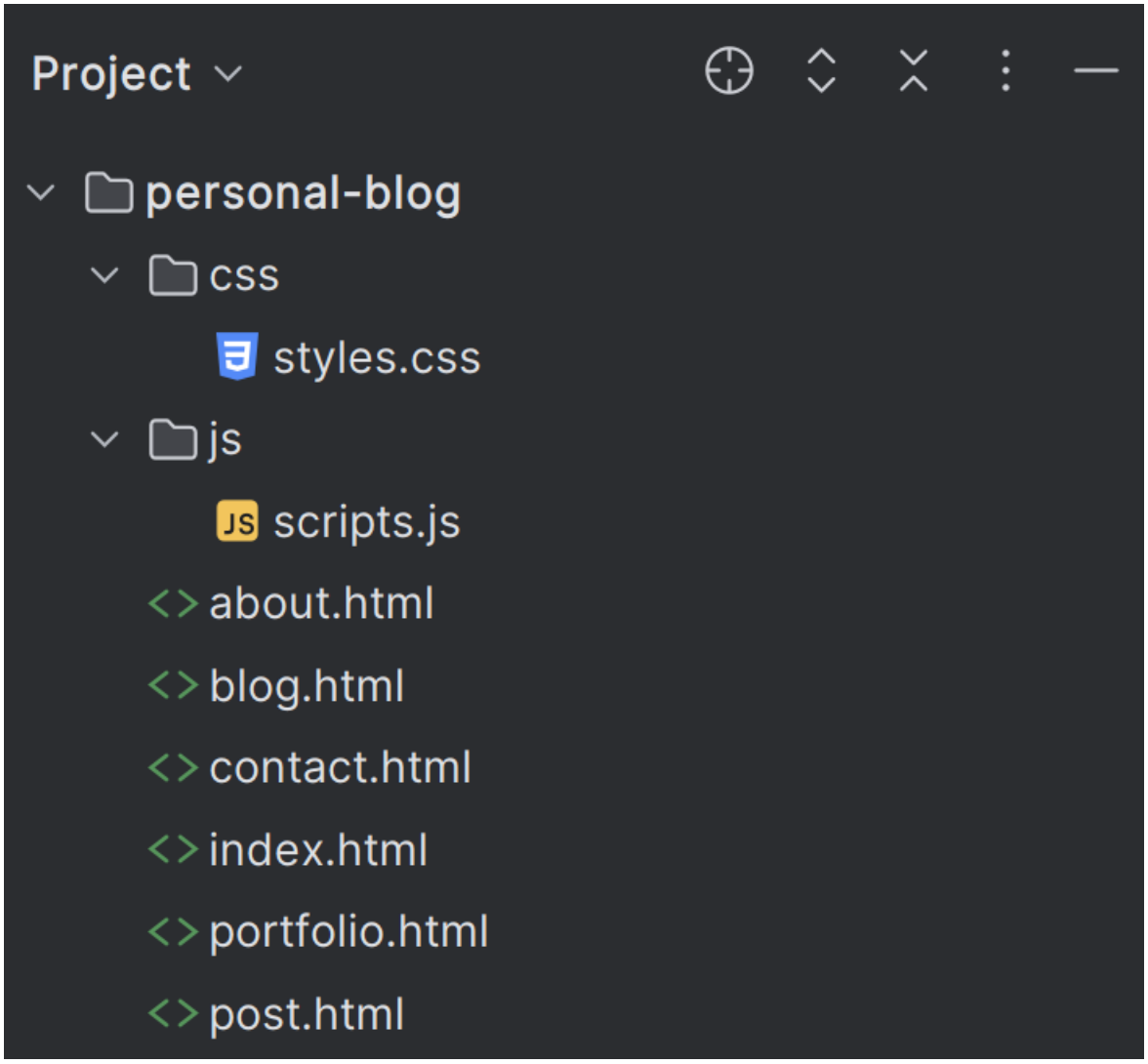
1. Create the Project Directory:

Start by creating a project directory to keep your HTML files organized. Name the project directory with your actual name "YourNameWebsite".

Here are some guidelines for structuring your website folder:

1. **Organize your files by type**: Create folders for different types of files such as HTML, CSS, JavaScript, images, and fonts. This will help you keep your files organized and easy to find.
2. **Create a root folder**: Create a root folder for your website and place all your files inside it. This will help you keep your files organized and make it easier to move your website to a different location if needed.
3. **Use relative paths**: Use relative paths when linking to other pages or files on your website. This will help you avoid broken links if you move your website to a different location.
4. **Use descriptive file names**: Use descriptive file names that reflect the content of the file. This will help you find files quickly and make it easier to maintain your website.
5. Inside the " YourNameWebsite " directory, create five HTML files for your web pages: index.html, about.html, blogs.html, portfolio.html, and contact.html.

Website structure:



2. Set Up the Basic HTML Structure:

In each HTML file, set up the basic HTML structure.

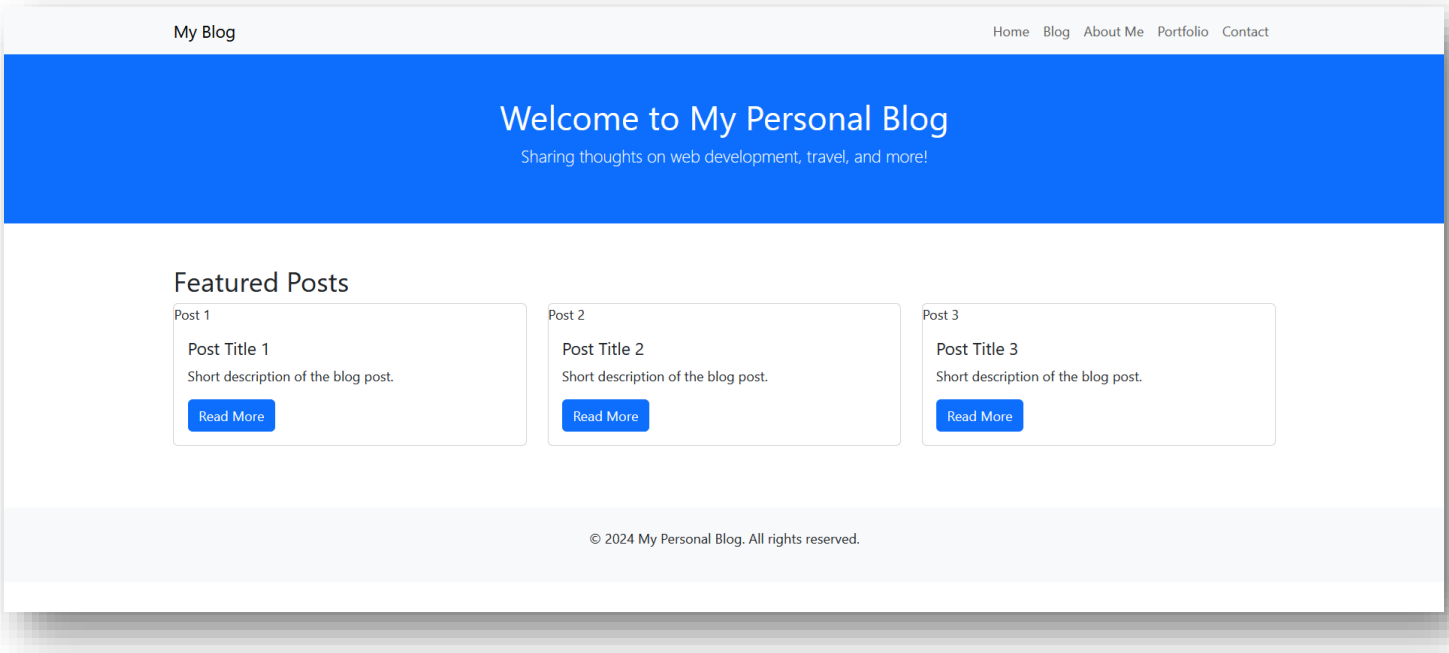
Here are some guidelines for structuring your HTML page:

- Start with a doctype declaration: All HTML documents must start with a doctype declaration. This tells the browser which version of HTML the document is using. For HTML5, the doctype declaration is <!DOCTYPE html>.
- Add the HTML tag: The <html> tag is the root element of an HTML page. It contains all other elements on the page.
- Add the head section: The head section contains metadata about the document, such as the title of the page, links to stylesheets, and scripts.
- Add the body section: The body section contains the content of the page, such as text, images, and other media.
- Use semantic tags: Semantic tags are HTML tags that describe the content they contain. They help search engines and other tools understand the structure of your page. Examples of semantic tags include <header>, <nav>, <main>, <article>, <section>, and <footer>.

3. Populate Each Page:

Inside each HTML file add content:

- Prepare the Home Page (index.html) without CSS. Prepare the HTML files for the following final result. Designing the HTML files with CSS is the next problem. Create a template HTML file first and then populate other web pages.



- **Portfolio of Projects (portfolio.html)**

My Portfolio

Welcome to my personal portfolio page.

About Me

Hello! I'm a web developer with a passion for creating engaging and functional websites. With experience in HTML, CSS, JavaScript, and various frameworks, I bring ideas to life on the web.

Skills

- HTML5
- CSS3
- JavaScript (ES6+)
- React.js
- Node.js
- Python

Projects

Project 1: Personal Blog

Description: A simple, responsive personal blog created using HTML, CSS, and Bootstrap. The blog includes features such as recent posts, categories, and a contact form.

View Project

Project 2: Portfolio Website

Description: This very portfolio website that showcases my projects, skills, and contact details.

View Project

Project 3: To-Do List App

Description: A simple to-do list application built using JavaScript, allowing users to add, edit, and delete tasks. Data is stored locally using the browser's localStorage.

View Project

-

Contact Page (contact.html):

About Me Page (about.html):

About Me

A brief overview of who I am and what I do.

Profile Photo

Hello, I'm [Your Name]

I'm a passionate web developer and technology enthusiast with a deep interest in creating functional, aesthetically pleasing websites. Over the years, I have worked with numerous web technologies to bring ideas to life on the web.

With experience in HTML, CSS, JavaScript, and various frameworks like React and Node.js, I love building engaging web applications that focus on user experience.

Skills

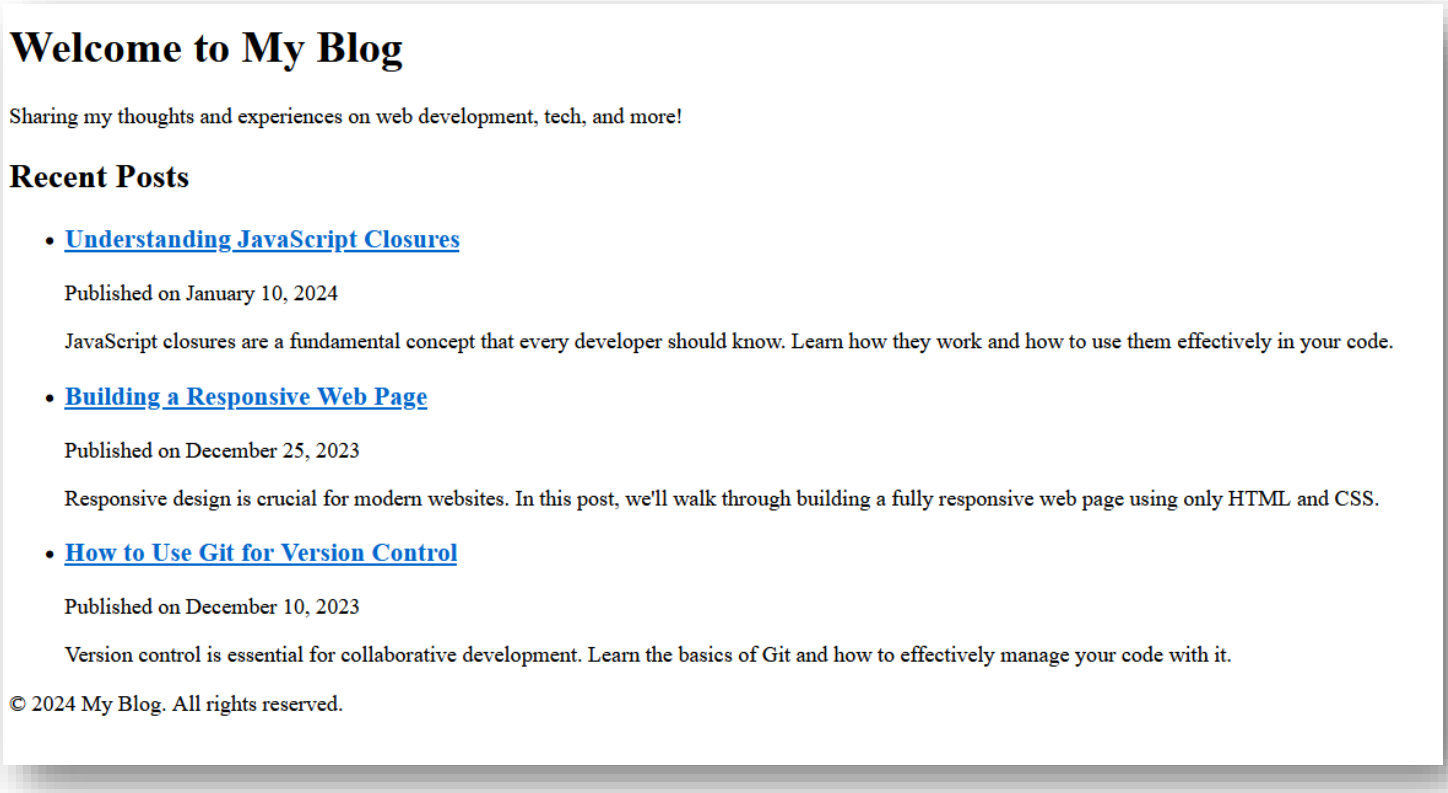
- HTML5 & CSS3
- JavaScript & ES6+
- React.js
- Node.js & Express
- Python & Django
- Version Control (Git & GitHub)

My Hobbies

When I'm not coding, I enjoy exploring other creative outlets, such as:

- Traveling and exploring new cultures
- Photography
- Reading tech blogs and staying up-to-date with the latest trends in technology
- Playing guitar

- **My Blog Page (blogs.html)**



Welcome to My Blog

Sharing my thoughts and experiences on web development, tech, and more!

Recent Posts

- **Understanding JavaScript Closures**

  Published on January 10, 2024

  JavaScript closures are a fundamental concept that every developer should know. Learn how they work and how to use them effectively in your code.

- **Building a Responsive Web Page**

  Published on December 25, 2023

  Responsive design is crucial for modern websites. In this post, we'll walk through building a fully responsive web page using only HTML and CSS.

- **How to Use Git for Version Control**

  Published on December 10, 2023

  Version control is essential for collaborative development. Learn the basics of Git and how to effectively manage your code with it.

4. Test Your Website:

Open each HTML file in a web browser to ensure that your pages are displaying correctly, and the navigation works as expected.

5. Host Your Website (Optional):

If you want to get extra bonus points to make your website accessible on the internet, you can host it on a web server by finding free hosting for your website.

Submit the link to your hosted website.

6. Submit your website

a. Add a README.txt file with any valuable comments that might help me with an assessment of your submission.
b. Archive your website using zip format.
c. Name is your own Name and Surname.