**Batch: T7**

**Practical No. 1**

**Title of Assignment: Study of Web and Basics**

**Student Name: Shreyash Patil**

**Student PRN: 22510039**

**Problem Statement 1: Basic HTML**

### **1. Evolution of Web including Web 3.0**

* **Web 1.0 (Static Web):** The early stage of the web, characterized by static websites that displayed content without user interaction. Pages were primarily text-based with limited visuals.
* **Web 2.0 (Dynamic Web):** This phase introduced interactive web experiences, user-generated content, social media platforms, and web applications. Technologies like AJAX, JavaScript, and APIs played a significant role in creating dynamic, responsive websites.
* **Web 3.0 (Semantic Web):** The next phase of the web aims to create a more intelligent and autonomous web. It focuses on decentralization, blockchain technology, and artificial intelligence to enable machines to understand and process information more effectively. Web 3.0 also emphasizes user privacy and data ownership.

### 2. Ports and Protocols Used by Web

* **HTTP/HTTPS (Protocols):**
  + **HTTP (HyperText Transfer Protocol):** The protocol used for transferring web pages on the internet. It operates over TCP/IP and is stateless and connectionless.
  + **HTTPS (HTTP Secure):** An extension of HTTP that uses SSL/TLS to encrypt data, ensuring secure communication between the client and server.
* **Ports:**
  + **Port 80:** Used by HTTP to serve web pages.
  + **Port 443:** Used by HTTPS for secure web communication.

### 3. Difference Between HTTP & HTTPS

* **HTTP:**
  + No encryption; data is transmitted in plaintext.
  + Vulnerable to eavesdropping and man-in-the-middle attacks.
  + Operates on port 80.
* **HTTPS:**
  + Encrypts data using SSL/TLS, providing confidentiality and integrity.
  + Protects against various attacks, making it essential for secure transactions.
  + Operates on port 443.

### 4. Web Developer Tools

* **Web Developer Tools:** A set of in-browser utilities that allow developers to inspect, debug, and optimize web pages. Tools like Chrome DevTools provide features like:
  + **Element Inspector:** Examine and modify HTML and CSS on the fly.
  + **JavaScript Debugger:** Step through code, set breakpoints, and identify issues.
  + **Network Analyzer:** Monitor network requests, inspect headers, and analyze load times.
* **Need:** Essential for ensuring that web applications are optimized, bug-free, and provide a consistent user experience across different browsers.

### 5. Client-Server Architecture vs. MVC Architecture

* **Client-Server Architecture:** A model where the client requests resources from a server, which processes the request and sends back the response. It's used in web applications, email services, etc.
  + **Diagram:**
  + **Use Case:** Best for applications that require centralized data processing, such as websites, online games, etc.
* **MVC Architecture (Model-View-Controller):** A design pattern that separates an application into three interconnected components:
  + **Model:** Manages data and business logic.
  + **View:** Displays data and handles user interaction.
  + **Controller:** Mediates between the model and view, handling input and updating the model or view accordingly.
  + **Diagram:**
  + **Use Case:** Ideal for applications that require a clear separation of concerns, making the codebase more modular and maintainable.

### 6. HTML and HTML5

* **HTML (HyperText Markup Language):** The standard language for creating web pages. It defines the structure of a webpage using elements like headings, paragraphs, links, etc.
* **HTML5:** The latest version of HTML, introduced new elements (e.g., <article>, <section>), APIs (e.g., Geolocation API), and support for multimedia (e.g., <audio>, <video>). It also improved semantic structure and cross-platform compatibility.

### 7. Current Versions of HTML and CSS

* **HTML:** HTML5 is the latest standard.
* **CSS:** CSS3 is the current standard, offering advanced features like animations, grid layouts, and transitions.

### 8. Tools for Front-End and Back-End Development

* **Front-End Development:**
  + **Frameworks/Libraries:** React.js, Angular.js, Vue.js
  + **Tools:** Webpack, Babel, npm/yarn, Bootstrap
* **Back-End Development:**
  + **Languages/Frameworks:** Node.js, Django, Flask, Ruby on Rails
  + **Databases:** MySQL, PostgreSQL, MongoDB
  + **Tools:** Docker, Kubernetes, Jenkins

### 9. MERN Stack

* **MERN Stack:** A collection of JavaScript-based technologies used for building web applications.
  + **Includes:**
    - **MongoDB:** NoSQL database.
    - **Express.js:** Web framework for Node.js.
    - **React.js:** Front-end library.
    - **Node.js:** Server-side runtime environment.
  + **Why Preferred:** Enables full-stack development using a single language (JavaScript). It’s efficient, flexible, and scalable.
  + **When to Use:** Ideal for dynamic single-page applications (SPAs) and applications that require high performance and scalability.

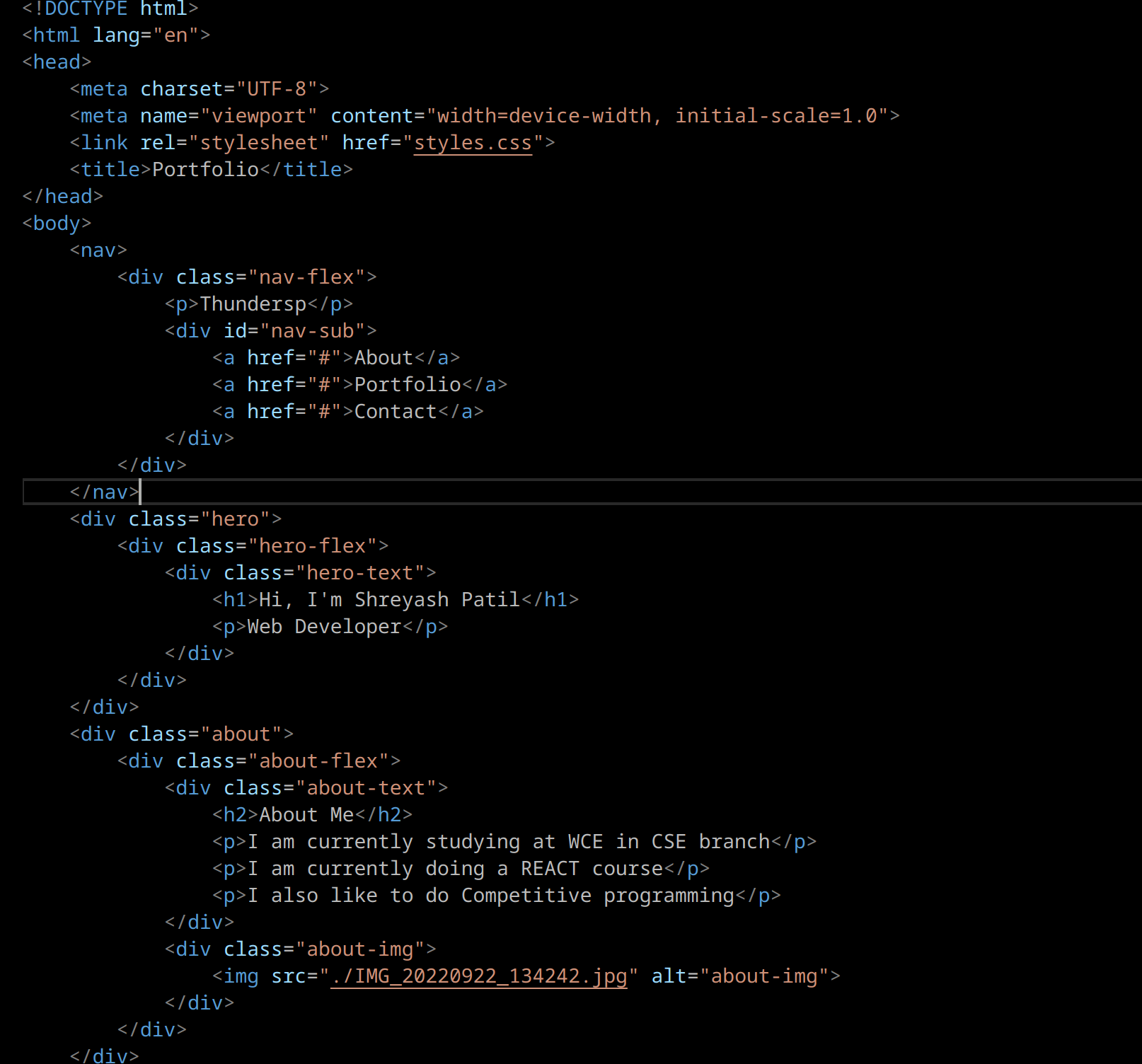
### 10. HTML5 Input Types, APIs, and Form Elements

* **New Input Types:** email, date, number, range, color, etc.
* **APIs:** Geolocation API, Web Storage API, Canvas API, WebSockets.
* **Form Elements:** <datalist>, <output>, <progress>, <meter>.
* **Media Elements:** <audio>, <video>, <track>.

### 11. HTML5 Web Storage

* **HTML5 Web Storage:** A mechanism that allows web applications to store data locally within the user's browser.
  + **Types:**
    - **Local Storage:** Stores data with no expiration date.
    - **Session Storage:** Stores data for the duration of the page session.
  + **Advantages:** Faster access, reduces server load, more storage capacity compared to cookies.
  + **Use Case:** Saving user preferences, maintaining state between page reloads, etc

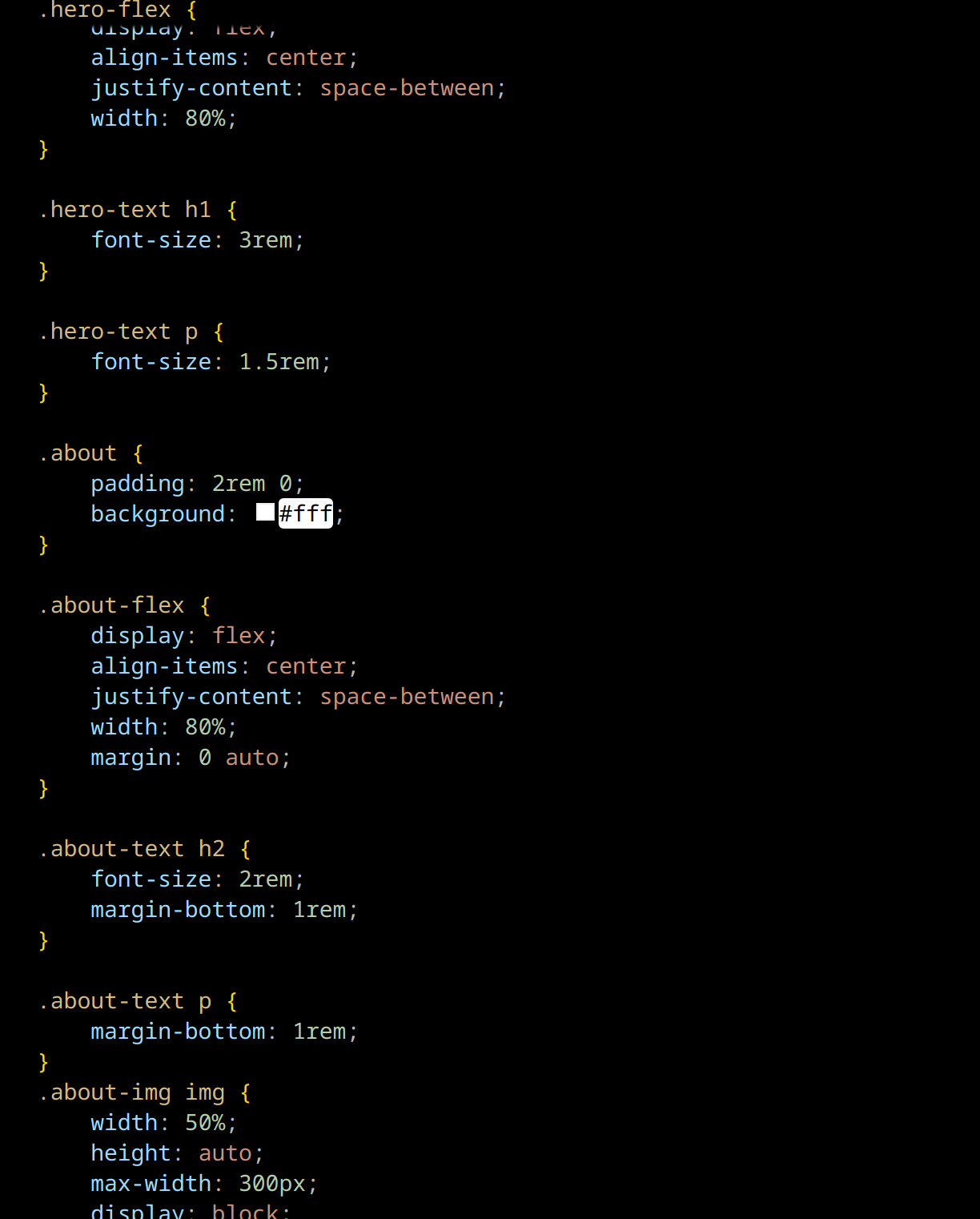
**Index.html:**



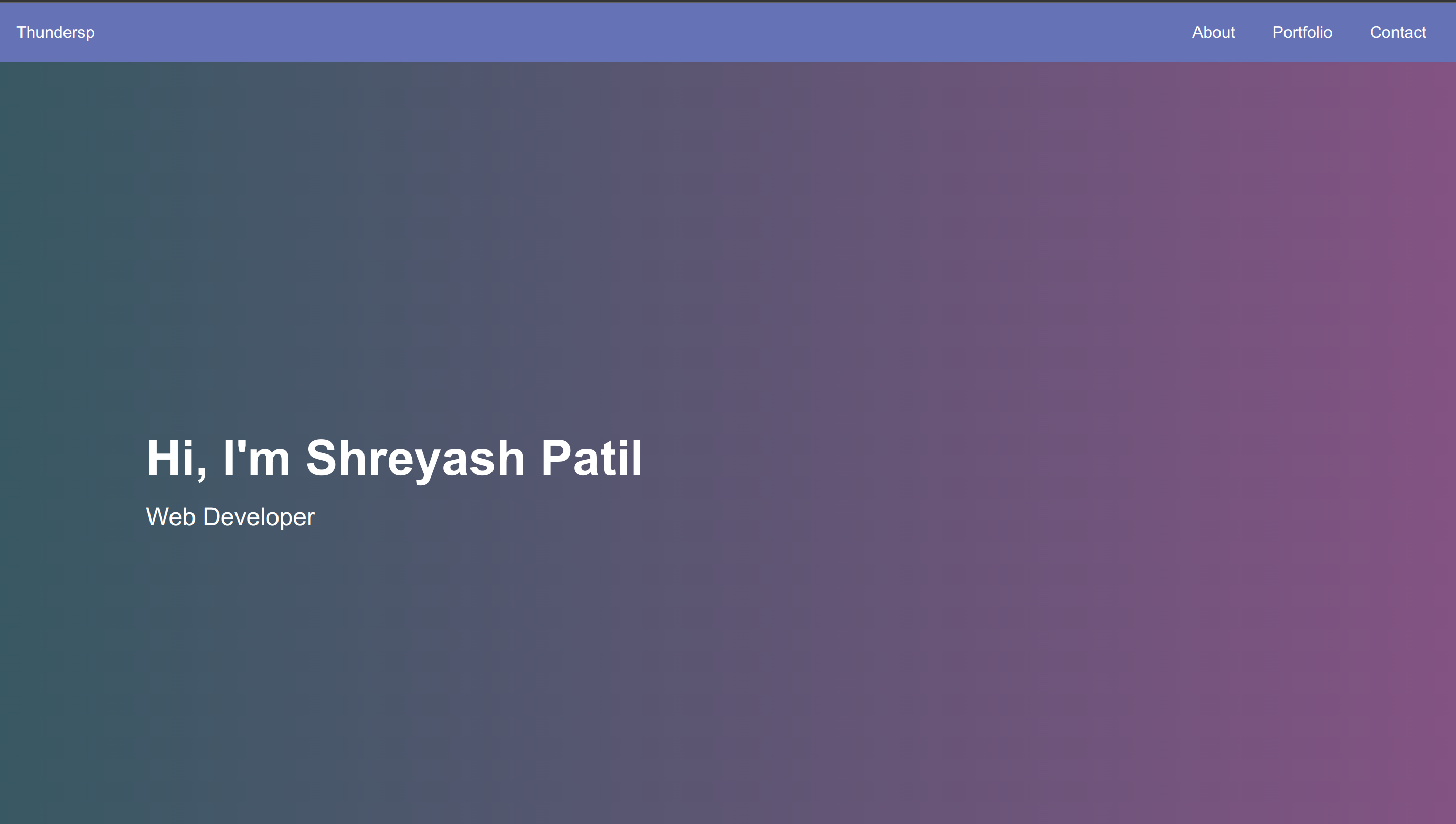


**Styles.css**





**Output:**

****