Daily Diary - Web Development Training

Day 1: Introduction to HTML

Date: 5-06-24

Summary of the Day: Today marked the beginning of our web development training. The focus of the session was on understanding the basics of HTML (HyperText Markup Language) and setting up our development environment using Visual Studio Code (VSCode). The day was productive, providing a solid foundation for our journey into web development.

Detailed Notes:

1. Introduction to HTML:

- **What is HTML?**
    - HTML stands for HyperText Markup Language.
    - It is the standard language used to create and design documents on the web.
    - HTML structures the web content and tells the browser how to display the text, images, and other forms of multimedia on the webpage.
- **Importance of HTML:**
    - HTML is the backbone of any website.
    - It is essential for creating web pages and web applications.
    - It enables the embedding of images, videos, and forms, creating interactive web pages.

2. Setting Up the Editor (VSCode):

- **What is VSCode?**
    - Visual Studio Code (VSCode) is a free, open-source code editor developed by Microsoft.
    - It supports various programming languages and comes with a multitude of extensions to enhance productivity.
- **Installation and Setup:**
    - Downloaded and installed VSCode from the official website.
    - Configured basic settings and installed essential extensions for web development (e.g., HTML, CSS, Live Server).

3. Basic HTML Structure:

- **DOCTYPE Declaration:**

```
<!DOCTYPE html>
```

- This declaration defines the document type and version of HTML.

- It is mandatory and should be the very first line in any HTML document.

 HTML Document Structure:

- <html>
- <head>
-   <title>Page Title</title>
- </head>
- <body>
-   <h1>This is a Heading</h1>
-   <p>This is a paragraph.</p>
- </body>
- </html>
-
    - <html>: The root element that wraps all the content of the webpage.
    - <head>: Contains meta-information about the HTML document (e.g., title, meta tags, links to CSS).
    - <title>: Sets the title of the webpage, which is displayed in the browser's title bar or tab.
    - <body>: Contains the content of the HTML document, such as headings, paragraphs, images, links, etc.

4. Basic HTML Tags:

- **Headings:**

 <h1> to <h6>

- Headings range from <h1> (most important) to <h6> (least important).

 Paragraph:

- <p>This is a paragraph.</p>
-
    - The <p> tag defines a paragraph.
- **Lists:**
    - Ordered List: <ol>
    - Unordered List: <ul>
    - List Item: <li>

Practical Exercise:

- Created a simple HTML document using the basic structure and tags learned today.
- Tested the document in a web browser to see the rendered output.

Reflection: Today's session was a great start to understanding the fundamentals of web development. Learning about HTML and setting up VSCode has provided a good base to build upon. I look forward to delving deeper into HTML and exploring more advanced concepts in the coming days.

Day 2: HTML Elements and Attributes

Date: 6-06-24

---

1. HTML Elements and Attributes:

- **HTML Elements:**
  - An HTML element is defined by a start tag, some content, and an end tag.
  - Example:
-
- <p>This is a paragraph.</p>

 HTML Attributes:

- Attributes provide additional information about HTML elements.
- They are always included in the opening tag and usually come in name/value pairs.
- Example:
-
  - <a href="https://www.example.com">Visit Example</a>
  -

2. Basic HTML Tags:

- **Paragraph (<p>):**

 <p>This is a paragraph.</p>

- Used to define a block of text.

 Headings (<h1> to <h6>):

 <h1>This is an H1 heading</h1>

<h2>This is an H2 heading</h2>

- Used to define headings of different levels, with <h1> being the most important and <h6> the least.

 Ordered List (<ol>) and Unordered List (<ul>):

```
 <ol>
  <li>First item</li>
  <li>Second item</li>
</ol>

<ul>
  <li>First item</li>
  <li>Second item</li>
</ul>
```

- <ol> is used for ordered lists with numbers.
- <ul> is used for unordered lists with bullets.
- <li> is used to define each list item.

 Links (<a>):

```
 <a href="https://www.example.com">Visit Example</a>
```

- The <a> tag defines a hyperlink.
- The href attribute specifies the URL of the page the link goes to.

 Sections (<section>):

```
 <section>
  <h1>Section Title</h1>
  <p>Section content goes here.</p>
</section>
```

- The <section> tag defines a section in a document, typically with a heading.

 Images (<img>):

- <img src="image.jpg" alt="Description of image">
-
    - The <img> tag is used to embed an image in an HTML page.
    - The src attribute specifies the path to the image.
    - The alt attribute provides an alternative text for the image if it cannot be displayed.

Day 3: HTML Forms and Enhancements

Date: 7-06-24

---

1. HTML Form Elements:

- **Form (<form>):**
    - The <form> tag is used to create an HTML form for user input.
    - It can contain various input elements like text fields, checkboxes, radio buttons, submit buttons, etc.
    - Example:
-
- <form action="/submit-form" method="post">
-   <!-- form elements go here -->
- </form>

 Input (<input>):

- The <input> tag specifies an input field where the user can enter data.
- Types of input fields include text, password, email, number, submit, etc.
- Example:
-
- <input type="text" name="username">

 Label (<label>):

- The <label> tag defines a label for an <input> element.
- It improves the usability by making the form more accessible.
- Example:
-
- <label for="username">Username:</label>
- <input type="text" id="username" name="username">

 Button (<button>):

- The <button> tag defines a clickable button.
- Types include submit, reset, and button.
- Example:

-
    - <button type="submit">Submit</button>
    -

2. Creating a Basic Form:

- **Form Structure:**
- <form action="/submit-form" method="post">
- <label for="name">Name:</label>
- <input type="text" id="name" name="name"><br><br>
-
- <label for="email">Email:</label>
- <input type="email" id="email" name="email"><br><br>
-
- <label for="password">Password:</label>
- <input type="password" id="password" name="password"><br><br>
-
- <button type="submit">Submit</button>
- </form>
-

3. Enhancing Our HTML Page:

- **Adding Images:**

 <h2>My Favorite Place</h2>

<img src="place.jpg" alt="A beautiful scenery">

 Adding Text:

 <p>This paragraph describes my favorite place. It is a serene and beautiful spot perfect for relaxation and unwinding.</p>

 Adding Links:

- <p>Visit <a href="https://www.example.com">Example.com</a> for more information.</p>
-

4. Combined HTML Page:

<!DOCTYPE html>

<html>

```
<head>
  <title>My First Web Page</title>
</head>
<body>
  <h1>Welcome to My First Web Page</h1>
  <p>This is a paragraph introducing the content of the page.</p>

  <h2>My Favorite Foods</h2>
  <ul>
    <li>Pizza</li>
    <li>Sushi</li>
    <li>Ice Cream</li>
  </ul>

  <h2>Steps to Make a Sandwich</h2>
  <ol>
    <li>Get two slices of bread.</li>
    <li>Add your favorite fillings.</li>
    <li>Put the slices together.</li>
  </ol>

  <h2>Learn More</h2>
  <p>Visit <a href="https://www.example.com">Example.com</a> for more information.</p>

  <h2>My Favorite Place</h2>
  <img src="place.jpg" alt="A beautiful scenery">
  <p>This paragraph describes my favorite place. It is a serene and beautiful spot perfect for
relaxation and unwinding.</p>

  <h2>Contact Me</h2>
  <form action="/submit-form" method="post">
```

```
    <label for="name">Name:</label>
    <input type="text" id="name" name="name"><br><br>

    <label for="email">Email:</label>
    <input type="email" id="email" name="email"><br><br>

    <label for="password">Password:</label>
    <input type="password" id="password" name="password"><br><br>

    <button type="submit">Submit</button>
  </form>
</body>
</html>
```

Day 4: HTML Tables

Date: 10-06-24

---

1. HTML Table Elements:

- **Table (<table>):**
    - The <table> tag defines an HTML table.
    - Example:
-
- <table>
-   <!-- table content goes here -->
- </table>

 Table Row (<tr>):

- The <tr> tag defines a row in a table.
- Example:
-
- <tr>
-   <!-- table cells go here -->

- </tr>

 Table Header (<th>):

- The <th> tag defines a header cell in a table.
- Header cells are bold and centered by default.
- Example:
-
- <th>Header</th>

 Table Data (<td>):

- The <td> tag defines a standard data cell in a table.
- Example:
-
- <td>Data</td>

 Table Caption (<caption>):

- The <caption> tag defines a table caption.
- Example:
-
    - <caption>Table Caption</caption>
    -

2. HTML Table Attributes:

- **Border:**
    - Adds a border around the table and its cells.
    - Example:
-
- <table border="1">
-   <!-- table content goes here -->
- </table>

 Cell Padding:

- Specifies the space between the cell content and its borders.
- Example:
-
- <table cellpadding="10">

- <!-- table content goes here -->
- </table>

 Cell Spacing:

- Specifies the space between cells.
- Example:
-
    - <table cellspacing="5">
    - <!-- table content goes here -->
    - </table>
    -

3. Creating a Table with Data:

- **Table Structure:**

```
<table border="1" cellpadding="5" cellspacing="0">
 <caption>Student Grades</caption>
 <tr>
  <th>Name</th>
  <th>Subject</th>
  <th>Grade</th>
 </tr>
 <tr>
  <td>John Doe</td>
  <td>Math</td>
  <td>A</td>
 </tr>
 <tr>
  <td>Jane Smith</td>
  <td>Science</td>
  <td>B</td>
 </tr>
 <tr>
  <td>Emily Johnson</td>
```

```
    <td>History</td>
    <td>A</td>
  </tr>
</table>
```

Day 5: Building a Comprehensive Web Page

Date: 11-06-24

---

1. Task Overview:

- Create a simple, comprehensive webpage.
- Incorporate headings, paragraphs, lists, links, images, forms, and tables.
- Use semantic HTML to structure the webpage.

2. Web Page Structure and Content:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>My Comprehensive Web Page</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 20px;
    }
    header, nav, main, aside, footer {
      margin-bottom: 20px;
    }
    header h1 {
      margin-bottom: 0;
    }
    nav ul {
      list-style-type: none;
      padding: 0;
    }
    nav ul li {
      display: inline;
      margin-right: 10px;
    }
    table {
      width: 100%;
      border-collapse: collapse;
    }
    table, th, td {
      border: 1px solid black;
    }
    th, td {
      padding: 10px;
      text-align: left;
    }
  </style>
</head>
<body>
  <header>
    <h1>Welcome to My Comprehensive Web Page</h1>
    <nav>
      <ul>
        <li><a href="#home">Home</a></li>
        <li><a href="#about">About</a></li>
        <li><a href="#contact">Contact</a></li>
      </ul>
    </nav>
  </header>
```

```html
<main>
  <section>
    <h2>Introduction</h2>
    <p>This is a comprehensive webpage built using various HTML elements and semantic tags.</p>
  </section>

  <section>
    <h2>My Favorite Foods</h2>
    <ul>
      <li>Pizza</li>
      <li>Sushi</li>
      <li>Ice Cream</li>
    </ul>
  </section>

  <section>
    <h2>Steps to Make a Sandwich</h2>
    <ol>
      <li>Get two slices of bread.</li>
      <li>Add your favorite fillings.</li>
      <li>Put the slices together.</li>
    </ol>
  </section>

  <article>
    <h2>Learn More</h2>
    <p>Visit <a href="https://www.example.com">Example.com</a> for more information.</p>
  </article>

  <section>
```

```
    <h2>My Favorite Place</h2>
    <img src="place.jpg" alt="A beautiful scenery" style="width:100%; max-width:600px;">
    <p>This paragraph describes my favorite place. It is a serene and beautiful spot perfect for
relaxation and unwinding.</p>
  </section>

  <section>
    <h2>Contact Me</h2>
    <form action="/submit-form" method="post">
      <label for="name">Name:</label>
      <input type="text" id="name" name="name"><br><br>

      <label for="email">Email:</label>
      <input type="email" id="email" name="email"><br><br>

      <label for="password">Password:</label>
      <input type="password" id="password" name="password"><br><br>

      <button type="submit">Submit</button>
    </form>
  </section>

  <section>
    <h2>Student Grades</h2>
    <table>
      <caption>Student Grades</caption>
      <tr>
        <th>Name</th>
        <th>Subject</th>
        <th>Grade</th>
      </tr>
      <tr>
        <td>John Doe</td>
        <td>Math</td>
        <td>A</td>
      </tr>
      <tr>
        <td>Jane Smith</td>
        <td>Science</td>
        <td>B</td>
      </tr>
      <tr>
        <td>Emily Johnson</td>
        <td>History</td>
        <td>A</td>
      </tr>
    </table>
  </section>
 </main>

 <aside>
  <h2>Related Articles</h2>
  <p>Check out these related topics...</p>
 </aside>

 <footer>
  <p>&copy; 2024 My Website</p>
 </footer>
</body>
</html>
```

Day 6: Quick Review Of HTML

Date: 12-06-24

2. Quick Review of Concepts:

- **Day 1: Introduction to HTML**
  - Definition and purpose of HTML.
  - Setting up the VS Code editor.
  - Basic HTML structure and tags (e.g., <!DOCTYPE html>, <html>, <head>, <body>).
- **Day 2: HTML Elements and Attributes**
  - Basic tags like <p>, <h1>, <ol>, <ul>, <a>, src, alt.
  - Creating our first HTML page using learned elements.
- **Day 3: Forms and Media**
  - Tags like <form>, <input>, <label>, <button>.
  - Adding images and links to the webpage.
- **Day 4: HTML Tables**
  - Table elements (<table>, <tr>, <th>, <td>, <caption>).
  - Table attributes (border, cellpadding, cellspacing).
- **Day 5: Semantic HTML**
  - Importance of semantic HTML for accessibility, SEO, and maintainability.
  - Common semantic tags like <header>, <nav>, <main>, <section>, <article>, <aside>, <footer>.
- **Day 6: Comprehensive Web Page Creation**
  - Tasked with building a complete webpage using all learned elements and semantic tags.
  - Integration of a form and a table.

3. Doubt Clearing Session:

- Addressed common questions and challenges faced during the week.
- Clarified misunderstandings and reviewed complex concepts.
- Discussed practical applications and best practices.

4. Feedback on HTML Webpages:

- **General Feedback:**
  - Overall structure and organization.
  - Use of semantic tags and HTML elements.
  - Clarity and readability of code.

- **Specific Feedback:**
  - Suggestions for improving layout and design.
  - Tips for better form and table integration.
  - Advice on optimizing images and links.

5. Actionable Feedback:

- **Improving Code Structure:**
  - Ensure consistent indentation and spacing for readability.
  - Use comments to explain sections of the code.
- **Enhancing Accessibility:**
  - Use descriptive alt attributes for images.
  - Ensure forms are labeled properly for screen readers.
- **Optimizing Tables:**
  - Use appropriate table attributes for better presentation.
  - Ensure tables are responsive and readable on all devices.

Day 7: Introduction to CSS

Date:13-08-24

---

1. What is CSS?

- **Definition:**
  - CSS stands for Cascading Style Sheets.
  - It is used to control the layout and appearance of HTML elements on a web page.
- **Purpose:**
  - Separates content (HTML) from presentation (CSS).
  - Enhances the visual appeal and user experience of web pages.
  - Enables consistent styling across multiple web pages.

2. Types of CSS:

- **Inline CSS:**
  - Applied directly to an HTML element using the style attribute.
  - Example:
- ▯

- <p style="color: blue; font-size: 14px;">This is a blue paragraph.</p>
- **Pros:**
    - Quick and easy to apply for single elements.
- **Cons:**
    - Not suitable for applying styles across multiple elements or pages.
    - Can make HTML code cluttered and harder to maintain.

 Internal CSS:

- Defined within a <style> tag inside the <head> section of an HTML document.
- Example:
-
- <head>
-   <style>
-     p {
-       color: blue;
-       font-size: 14px;
-     }
-   </style>
- </head>
- <body>
-   <p>This is a blue paragraph.</p>
- </body>
- **Pros:**
    - Useful for applying styles to a single page.
- **Cons:**
    - Styles are not shared across multiple pages.
    - Can increase the size of the HTML document.

 External CSS:

- Defined in a separate .css file, which is linked to the HTML document.
- Example:
-
    - <!-- HTML file -->

    - <head>
    - <link rel="stylesheet" type="text/css" href="styles.css">
    - </head>
    - <body>
    - <p>This is a blue paragraph.</p>
    - </body>
    -
    - <!-- styles.css file -->
    - p {
    - color: blue;
    - font-size: 14px;
    - }
    -
    - **Pros:**
        - Allows for separation of content and style.
        - Styles can be applied across multiple pages.
        - Easier to maintain and update.
    - **Cons:**
        - Requires an additional HTTP request to load the CSS file.

3. Basic CSS Syntax:

- **Selectors:**
    - Target HTML elements to apply styles.
    - Examples:
        - element (e.g., p, h1)
        - class (e.g., .my-class)
        - id (e.g., #my-id)
- **Properties and Values:**
    - Define the style to be applied to the selected elements.
    - Syntax:

```
 selector {
  property: value;
```

}

 Example:

-
  - p {
  -   color: blue;
  -   font-size: 14px;
  - }
  -

4. Example of Using Different Types of CSS:

- **HTML File:**

```
 <!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>CSS Example</title>
  <style>
    /* Internal CSS */
    h1 {
      color: green;
      text-align: center;
    }
  </style>
  <link rel="stylesheet" type="text/css" href="styles.css">
</head>
<body>
  <h1>This is a heading</h1>
  <p style="color: blue; font-size: 14px;">This is an inline styled paragraph.</p>
  <p>This is an externally styled paragraph.</p>
</body>
</html>
```

□ **External CSS File (styles.css):**

```
/* External CSS */
p {
  color: red;
  font-size: 16px;
}
```

Day 9: CSS Selectors and Properties

Date: 14-06-24

---

1. CSS Selectors: Selectors are patterns used to select the elements you want to style.

- **Basic Selectors:**
  - **Element Selector:** Selects all elements of a given type.

```
□ p {
  color: blue;
}
```

□ **Class Selector:** Selects elements with a specific class attribute.

```
□ .my-class {
  font-size: 14px;
}
```

□ **ID Selector:** Selects an element with a specific ID attribute.

- □
- #my-id {
-   text-align: center;
- }

□ **Attribute Selectors:** Select elements based on an attribute or attribute value.

```
□ a[href] {
  color: green;
}
a[target="_blank"] {
  font-weight: bold;
}
```

 Combinator Selectors:

- **Descendant Selector:** Selects all elements that are descendants of a specified element.

 div p {

 color: red;

}

 Child Selector: Selects all elements that are direct children of a specified element.

 div > p {

 font-size: 18px;

}

 Adjacent Sibling Selector: Selects an element that is the next sibling of a specified element.

 h1 + p {

 margin-top: 20px;

}

 General Sibling Selector: Selects all siblings of a specified element.

-
- h1 ~ p {
-   color: gray;
- }

 Pseudo-class Selectors: Apply styles to elements based on their state.

 a:hover {

 color: red;

}

input:focus {

 border: 2px solid blue;

}

 Pseudo-element Selectors: Apply styles to a part of an element.

- p::first-line {
-   font-weight: bold;
- }
- p::before {
-   content: "Note: ";

- font-weight: bold;
- }
-

2. CSS Properties: Properties define the styles applied to the selected elements.

- **Text Properties:**
  - color: Sets the color of the text.

 p {

 color: blue;

}

 font-size: Sets the size of the font.

 p {

 font-size: 16px;

}

 text-align: Aligns the text inside an element.

-
- h1 {
- text-align: center;
- }

 Box Model Properties:

- width and height: Set the width and height of an element.

 div {

 width: 100px;

 height: 50px;

}

 padding: Adds space inside the element, around the content.

 div {

 padding: 10px;

}

 margin: Adds space outside the element, around the border.

 div {

 margin: 20px;

}

 border: Sets the border around an element.

-
- div {
-   border: 1px solid black;
- }

 Background Properties:

- background-color: Sets the background color of an element.

 body {

  background-color: #f0f0f0;

}

 background-image: Sets a background image for an element.

-
- div {
-   background-image: url('image.jpg');
- }

 Display and Positioning Properties:

- display: Specifies the display behavior of an element.

 .hidden {

  display: none;

}

 position: Specifies the positioning method used for an element (static, relative, absolute, fixed, sticky).

-
- .absolute {
-   position: absolute;
-   top: 50px;
-   left: 50px;
- }

 Flexbox Properties:

- display: flex: Defines a flex container and enables a flex context for all its direct children.

 .container {

 display: flex;

}

 justify-content: Aligns flex items along the main axis.

 .container {

 justify-content: center;

}

 align-items: Aligns flex items along the cross axis.

-
    - .container {
    - align-items: center;
    - }
    -

3. Practical Examples:

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>CSS Selectors and Properties</title>
 <style>
  /* Element Selector */
  p {
   color: blue;
   font-size: 14px;
  }

  /* Class Selector */
  .highlight {
   background-color: yellow;
  }
```

```
/* ID Selector */
#unique {
  font-weight: bold;
  text-align: center;
}

/* Attribute Selector */
a[href^="https"] {
  color: green;
}

/* Descendant Selector */
div p {
  margin-left: 20px;
}

/* Child Selector */
ul > li {
  list-style-type: square;
}

/* Adjacent Sibling Selector */
h1 + p {
  font-style: italic;
}

/* General Sibling Selector */
h1 ~ p {
  color: gray;
}
```

```
    /* Pseudo-class Selector */
    a:hover {
      text-decoration: underline;
    }

    /* Pseudo-element Selector */
    p::first-letter {
      font-size: 20px;
      color: red;
    }
  </style>
</head>
<body>
  <h1>This is a heading</h1>
  <p>This is a paragraph.</p>
  <p class="highlight">This is a highlighted paragraph.</p>
  <p id="unique">This is a unique paragraph.</p>
  <a href="https://example.com">This is a link.</a>
  <div>
    <p>This is a paragraph inside a div.</p>
  </div>
  <ul>
    <li>List item 1</li>
    <li>List item 2</li>
  </ul>
  <a href="https://example.com">Hover over this link.</a>
</body>
</html>
```

Day 10: CSS Box Model and Fluid Layouts

Date: 17-06-24

Summary of the Day: On the tenth day of our web development training, we explored two important concepts in CSS: the CSS Box Model and fluid layouts. Understanding these topics is essential for creating well-structured and responsive web pages. The session covered the components of the CSS Box Model and techniques for designing fluid, flexible layouts.

Detailed Notes:

1. CSS Box Model: The CSS Box Model is a fundamental concept that describes how elements are structured and spaced on a web page.

- **Components of the Box Model:**
  - **Content:** The actual content of the element, such as text or an image.
  - **Padding:** The space between the content and the border. It increases the size of the element without affecting its external dimensions.
  - **Border:** A line surrounding the padding (if any) and content.
  - **Margin:** The space outside the border, separating the element from other elements on the page.
- **Visual Representation:**

```
 element {
  width: 100px;
  height: 100px;
  padding: 10px;
  border: 5px solid black;
  margin: 15px;
}
```

This would result in:

- Content: 100px x 100px
- Padding: 10px on all sides (total size becomes 120px x 120px)
- Border: 5px on all sides (total size becomes 130px x 130px)
- Margin: 15px on all sides (total space occupied becomes 160px x 160px)

 Example:

- <style>
-   .box {
-     width: 100px;
-     height: 100px;

- padding: 10px;
- border: 5px solid black;
- margin: 15px;
- background-color: lightblue;
- }
- </style>
- <div class="box">Box Model Example</div>
-

2. Fluid Layouts: Fluid layouts, also known as liquid layouts, adapt to the size of the user's viewport, making web pages more responsive.

- **Percentage-Based Widths:** Using percentages allows elements to resize relative to their parent container.

```
□ .container {
  width: 80%; /* 80% of the parent container's width */
  margin: 0 auto; /* Center the container */
}
```

□ **Viewport Units:** Viewport units (vw and vh) are relative to the size of the viewport.

- 1vw is 1% of the viewport width.
- 1vh is 1% of the viewport height.

```
□ .responsive-box {
  width: 50vw; /* 50% of the viewport width */
  height: 50vh; /* 50% of the viewport height */
  background-color: lightgreen;
}
```

□ **Flexbox:** Flexbox is a powerful layout module that allows for the creation of flexible and responsive layouts.

```
.flex-container {
  display: flex;
  flex-wrap: wrap;
  justify-content: space-around;
}
```

```
.flex-item {
  flex: 1 1 auto;
  margin: 10px;
  background-color: lightcoral;
}
```

Example:

```
<style>
  .flex-container {
    display: flex;
    flex-wrap: wrap;
    justify-content: space-around;
  }

  .flex-item {
    flex: 1 1 auto;
    margin: 10px;
    background-color: lightcoral;
    padding: 20px;
    text-align: center;
  }
</style>
<div class="flex-container">
  <div class="flex-item">Item 1</div>
  <div class="flex-item">Item 2</div>
  <div class="flex-item">Item 3</div>
</div>

```

3. Practical Examples:

Example with Box Model:

```
<!DOCTYPE html>
<html lang="en">
```

```
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Box Model Example</title>
  <style>
    .box {
      width: 200px;
      padding: 20px;
      border: 5px solid black;
      margin: 15px;
      background-color: lightblue;
    }
  </style>
</head>
<body>
  <div class="box">This is an example of the box model.</div>
</body>
</html>
```

Example with Fluid Layout:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Fluid Layout Example</title>
  <style>
    .container {
      width: 80%;
      margin: 0 auto;
      background-color: lightgray;
      padding: 20px;
    }

    .responsive-box {
      width: 50vw;
      height: 50vh;
      background-color: lightgreen;
      margin: 20px 0;
    }

    .flex-container {
      display: flex;
      flex-wrap: wrap;
      justify-content: space-around;
    }

    .flex-item {
      flex: 1 1 200px;
      margin: 10px;
      background-color: lightcoral;
      padding: 20px;
      text-align: center;
    }
  </style>
</head>
<body>
  <div class="container">
    <h1>Fluid Layout Example</h1>
    <div class="responsive-box">Responsive Box</div>
    <div class="flex-container">
      <div class="flex-item">Flex Item 1</div>
      <div class="flex-item">Flex Item 2</div>
    <div class="flex-item">Flex Item 3</div>
   </div>
  </div>
</body>
</html>
```

Day 11: CSS Layouts

Date: 18-06-24

---

1. CSS Layout Basics:

- **Block Layout:**
  - Block-level elements occupy the full width of their container and start on a new line.
  - Examples: <div>, <p>, <h1>, <section>
  - Properties:
-
- div {
-   display: block;
-   width: 100%;
- }

 Inline Layout:

- Inline elements do not start on a new line and only occupy as much width as necessary.
- Examples: <span>, <a>, <strong>
- Properties:
-
- a {
-   display: inline;
- }

 Inline-Block Layout:

- Inline-block elements are similar to inline elements but can have width and height set.
- Examples: <img>, <button>
- Properties:
-

    - .inline-block {
    - display: inline-block;
    - width: 100px;
    - height: 50px;
    - }
    -

2. Modern Layout Techniques:

- **Flexbox:**
    - Flexbox is designed for one-dimensional layouts. It allows items to align and distribute space within a container.
    - Properties:

```
 .flex-container {
  display: flex;
  justify-content: space-between; /* Align items horizontally */
  align-items: center; /* Align items vertically */
}
.flex-item {
  flex: 1; /* Grow items to fill available space */
  margin: 10px;
}
```

 Example:

-
- <style>
-   .flex-container {
-     display: flex;
-     justify-content: space-between;
-     align-items: center;
-     background-color: lightgray;
-     padding: 20px;
-   }
-   .flex-item {

- flex: 1;
- margin: 10px;
- background-color: lightcoral;
- text-align: center;
- padding: 20px;
- }
- </style>
- <div class="flex-container">
- <div class="flex-item">Item 1</div>
- <div class="flex-item">Item 2</div>
- <div class="flex-item">Item 3</div>
- </div>

 CSS Grid:

- CSS Grid is a two-dimensional layout system that allows for both rows and columns.
- Properties:

 .grid-container {

```
  display: grid;
  grid-template-columns: repeat(3, 1fr); /* Three equal columns */
  grid-gap: 10px; /* Gap between items */
}
.grid-item {
  background-color: lightblue;
  text-align: center;
  padding: 20px;
}
```

 Example:

-
    - <style>
    - .grid-container {
    - display: grid;
    - grid-template-columns: repeat(3, 1fr);

  - grid-gap: 10px;
  - }
  - .grid-item {
  - background-color: lightblue;
  - text-align: center;
  - padding: 20px;
  - }
  - </style>
  - <div class="grid-container">
  - <div class="grid-item">Item 1</div>
  - <div class="grid-item">Item 2</div>
  - <div class="grid-item">Item 3</div>
  - <div class="grid-item">Item 4</div>
  - <div class="grid-item">Item 5</div>
  - <div class="grid-item">Item 6</div>
  - </div>
  -

3. Positioning Techniques:

- **Static Positioning:**
  - Default positioning of elements.
  - Example:
-
- .static {
- position: static;
- }

 Relative Positioning:

- Positioned relative to its normal position.
- Example:
-
- .relative {
- position: relative;

- top: 10px;
- left: 20px;
- }

 Absolute Positioning:

- Positioned relative to its nearest positioned ancestor.
- Example:
-
- .absolute {
- position: absolute;
- top: 50px;
- left: 50px;
- }

 Fixed Positioning:

- Positioned relative to the browser window.
- Example:
-
- .fixed {
- position: fixed;
- bottom: 0;
- width: 100%;
- background-color: lightgray;
- }

 Sticky Positioning:

- Switches between relative and fixed positioning based on the user's scroll position.
- Example:
-
    - .sticky {
    - position: -webkit-sticky; /* For Safari */
    - position: sticky;
    - top: 0;
    - background-color: yellow;

- }
-

Day 19-06-24

---

1. Introduction to Flexbox:

- Flexbox is designed for one-dimensional layouts, either in a row or a column.
- It consists of a flex container and flex items.

2. Flex Container Properties:

- **display: flex;** Defines a flex container and enables flex context for all its direct children.

 .flex-container {

 display: flex;

}

 flex-direction: Specifies the direction of the flex items.

 .flex-container {

 flex-direction: row; /* Default */

}

/* Other values: row-reverse, column, column-reverse */

 flex-wrap: Determines whether flex items should wrap or not.

 .flex-container {

 flex-wrap: nowrap; /* Default */

}

/* Other values: wrap, wrap-reverse */

 flex-flow: A shorthand for setting both flex-direction and flex-wrap.

 .flex-container {

 flex-flow: row wrap;

}

 justify-content: Aligns flex items along the main axis.

 .flex-container {

 justify-content: flex-start; /* Default */

}

/* Other values: flex-end, center, space-between, space-around, space-evenly */

 align-items: Aligns flex items along the cross axis.

 .flex-container {

  align-items: stretch; /* Default */

}

/* Other values: flex-start, flex-end, center, baseline */

 align-content: Aligns flex lines when there is extra space in the cross axis.

- .flex-container {
- align-content: stretch; /* Default */
- }
- /* Other values: flex-start, flex-end, center, space-between, space-around */
-

3. Flex Item Properties:

- **order:** Controls the order of the flex items.

 .flex-item {

  order: 1; /* Default is 0 */

}

 flex-grow: Specifies how much a flex item will grow relative to the rest.

 .flex-item {

  flex-grow: 1; /* Default is 0 */

}

 flex-shrink: Specifies how much a flex item will shrink relative to the rest.

 .flex-item {

  flex-shrink: 1; /* Default */

}

 flex-basis: Defines the initial size of a flex item.

 .flex-item {

  flex-basis: 100px; /* Default is auto */

}

 flex: A shorthand for flex-grow, flex-shrink, and flex-basis.

 .flex-item {

```
  flex: 1 1 100px;
}
```

 align-self: Allows the default alignment (or the one specified by align-items) to be overridden for individual flex items.

- .flex-item {
-   align-self: auto; /* Default */
- }
- /* Other values: flex-start, flex-end, center, baseline, stretch */
-

Example with Flex Properties:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Flexbox Properties</title>
  <style>
    .flex-container {
      display: flex;
      flex-direction: column;
      flex-wrap: wrap;
      justify-content: center;
      align-items: flex-start;
      align-content: space-between;
      height: 300px;
      background-color: lightblue;
    }

    .flex-item {
      background-color: lightgreen;
      margin: 10px;
      padding: 20px;
      text-align: center;
      order: 2;
      flex: 1 1 100px;
      align-self: center;
    }

    .flex-item:first-child {
      order: 1;
      flex: 2 1 150px;
    }
  </style>
</head>
<body>
  <div class="flex-container">
    <div class="flex-item">Item 1</div>
    <div class="flex-item">Item 2</div>
    <div class="flex-item">Item 3</div>
    <div class="flex-item">Item 4</div>
  </div>
</body>
</html>
```

Day 13: CSS Grid

Date: 20-06-24

Summary of the Day: On the thirteenth day of our web development training, we delved into CSS Grid. CSS Grid is a powerful two-dimensional layout system that allows developers to create complex and responsive grid-based layouts with ease. The session focused on understanding the basics of CSS Grid, including container properties and grid item properties.

Detailed Notes:

1. Introduction to CSS Grid: CSS Grid is designed to handle both rows and columns, providing a more robust layout system compared to Flexbox, which is primarily one-dimensional.

2. CSS Grid Container Properties: The following properties are used to define a grid container and specify its behavior.

- **display: grid;** Turns an element into a grid container.

```
 .grid-container {
  display: grid;
}
```

 grid-template-columns and grid-template-rows: Define the number and size of the columns and rows in the grid.

```
 .grid-container {
  grid-template-columns: 1fr 1fr 1fr; /* Three equal columns */
  grid-template-rows: 100px 200px; /* Two rows with specific heights */
}
```

 grid-template-areas: Names grid areas for easier reference and layout.

```
 .grid-container {
  grid-template-areas:
    'header header header'
    'sidebar content content'
    'footer footer footer';
}
```

 grid-gap (or gap): Sets the spacing between rows and columns.

```
 .grid-container {
  grid-gap: 10px; /* Space between rows and columns */
}
```

 grid-auto-flow: Controls how auto-placed items are inserted into the grid.

- .grid-container {
- grid-auto-flow: row; /* Default */
- }
- /* Other values: column, dense, row dense, column dense */
-

3. CSS Grid Item Properties: The following properties are used to define the placement and behavior of grid items within a grid container.

- **grid-column-start, grid-column-end, grid-row-start, grid-row-end:** Specify the start and end positions of a grid item.

```
 .grid-item {
  grid-column-start: 1;
  grid-column-end: 3; /* Span across the first two columns */
  grid-row-start: 1;
  grid-row-end: 2; /* Span across the first row */
}
```

 grid-column and grid-row: Shorthand for setting both the start and end positions.

```
 .grid-item {
  grid-column: 1 / 3; /* Span across the first two columns */
  grid-row: 1 / 2; /* Span across the first row */
}
```

 grid-area: Assigns an item to a named grid area.

```
 .grid-item {
  grid-area: header;
}
```

 justify-self: Aligns the grid item within its column.

```
 .grid-item {
  justify-self: center; /* Default is stretch */
}
/* Other values: start, end, stretch */
```

 align-self: Aligns the grid item within its row.

- .grid-item {
-   align-self: center; /* Default is stretch */
- }
- /* Other values: start, end, stretch */
-

4. Practical Examples:

Basic Grid Layout:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>CSS Grid Example</title>
  <style>
    .grid-container {
      display: grid;
      grid-template-columns: 1fr 1fr 1fr;
      grid-template-rows: 100px 100px;
      grid-gap: 10px;
      background-color: lightgray;
    }
    .grid-item {
      background-color: lightcoral;
      text-align: center;
      padding: 20px;
    }
  </style>
</head>
<body>
  <div class="grid-container">
    <div class="grid-item">Item 1</div>
    <div class="grid-item">Item 2</div>
    <div class="grid-item">Item 3</div>
    <div class="grid-item">Item 4</div>
    <div class="grid-item">Item 5</div>
    <div class="grid-item">Item 6</div>
  </div>
</body>
```

```
</html>
```

Grid Layout with Named Areas:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>CSS Grid Named Areas</title>
  <style>
    .grid-container {
      display: grid;
      grid-template-areas:
        'header header header'
        'sidebar content content'
        'footer footer footer';
      grid-gap: 10px;
      background-color: lightgray;
    }
    .header {
      grid-area: header;
      background-color: lightblue;
      text-align: center;
      padding: 20px;
    }
    .sidebar {
      grid-area: sidebar;
      background-color: lightgreen;
      text-align: center;
      padding: 20px;
    }
    .content {
      grid-area: content;
      background-color: lightcoral;
      text-align: center;
      padding: 20px;
    }
    .footer {
      grid-area: footer;
      background-color: lightgoldenrodyellow;
      text-align: center;
      padding: 20px;
    }
  </style>
</head>
<body>
  <div class="grid-container">
    <div class="header">Header</div>
    <div class="sidebar">Sidebar</div>
    <div class="content">Content</div>
    <div class="footer">Footer</div>
  </div>
</body>
</html>
```

Day 14: Applying External CSS

Date: 21-06-24

---

1. Setting Up External CSS:

- Create a separate CSS file (e.g., styles.css).
- Link the external CSS file to the HTML document using the <link> tag within the <head> section.
- <link rel="stylesheet" href="styles.css">
-

2. Structuring the HTML:

- Ensure the HTML document is well-structured with semantic tags.
- Example of a basic HTML structure:
- <!DOCTYPE html>
- <html lang="en">
- <head>
-   <meta charset="UTF-8">
-   <meta name="viewport" content="width=device-width, initial-scale=1.0">
-   <title>My Webpage</title>
-   <link rel="stylesheet" href="styles.css">
- </head>
- <body>
-   <header class="header">Header Content</header>
-   <nav class="navbar">Navigation Bar</nav>
-   <main class="main-content">
-     <section class="content">Main Content</section>
-     <aside class="sidebar">Sidebar Content</aside>
-   </main>
-   <footer class="footer">Footer Content</footer>
- </body>
- </html>
-

3. Writing External CSS:

Basic Styling:

```
/* styles.css */


body {
  font-family: Arial, sans-serif;
  margin: 0;
  padding: 0;
  background-color: #f0f0f0;
}

.header, .navbar, .footer {
  background-color: #333;
  color: white;
  text-align: center;
  padding: 10px 0;
}

.main-content {
  display: flex;
  justify-content: space-between;
  margin: 20px;
}

.content, .sidebar {
  background-color: #fff;
  padding: 20px;
  margin: 10px;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
  flex: 1;
}
```

Advanced Styling with Flexbox and Grid:

```
/* Flexbox for Navigation Bar */
.navbar {
  display: flex;
  justify-content: space-around;
  align-items: center;
}

/* Grid for Main Content Layout */
```

```
.main-content {
  display: grid;
  grid-template-columns: 2fr 1fr;
  gap: 20px;
}

.content {
  grid-column: 1 / 2;
}

.sidebar {
  grid-column: 2 / 3;
}
```

Adding Visual Enhancements:

```
/* Adding Hover Effects */
.header, .navbar, .footer {
  transition: background-color 0.3s;
}

.header:hover, .navbar:hover, .footer:hover {
  background-color: #555;
}

/* Styling Links */
a {
  color: #333;
  text-decoration: none;
  transition: color 0.3s;
}

a:hover {
  color: #007BFF;
}

/* Styling Buttons */
button {
  background-color: #007BFF;
  color: white;
  border: none;
  padding: 10px 20px;
  cursor: pointer;
  transition: background-color 0.3s;
}

button:hover {
  background-color: #0056b3;
}
```

Day 15: Introduction to JavaScript

Date23-06-24

Detailed Notes:

1. What is JavaScript?

- JavaScript is a scripting language used to create and control dynamic website content.
- It is an essential technology alongside HTML and CSS.
- JavaScript can be used for client-side (in the browser) and server-side (with Node.js) development.

2. Including JavaScript in HTML:

- **Inline JavaScript:**

 <button onclick="alert('Hello, World!')">Click Me</button>

 Internal JavaScript:

 <!DOCTYPE html>

<html lang="en">

```
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Internal JavaScript Example</title>
  <script>
    function showAlert() {
      alert('Hello, World!');
    }
  </script>
</head>
<body>
  <button onclick="showAlert()">Click Me</button>
</body>
</html>
```

 External JavaScript:

- Create a separate JavaScript file (e.g., script.js).
- Link the external JavaScript file in the HTML document.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>External JavaScript Example</title>
  <script src="script.js" defer></script>
</head>
<body>
  <button onclick="showAlert()">Click Me</button>
</body>
</html>
```

- script.js:
-

- function showAlert() {
-   alert('Hello, World!');
- }
-

3. Basic JavaScript Syntax:

- **Variables:**

```
 var x = 5;        // ES5: function-scoped
let y = 10;       // ES6: block-scoped
const z = 15;     // ES6: block-scoped, constant
```

 Data Types:

- Strings, Numbers, Booleans, Arrays, Objects, etc.

```
 let name = "John";
let age = 25;
let isStudent = true;
let hobbies = ["reading", "sports"];
let person = {
  firstName: "Jane",
  lastName: "Doe"
};
```

 Operators:

- Arithmetic: +, -, *, /, %
- Assignment: =, +=, -=, *=, /=
- Comparison: ==, ===, !=, !==, <, >, <=, >=
- Logical: &&, ||, !

 Functions:

```
 function greet(name) {
  return "Hello, " + name;
}

let greeting = greet("Alice");
console.log(greeting); // Output: Hello, Alice
```

 Control Structures:

- **Conditional Statements:**

```
 if (age >= 18) {
  console.log("Adult");
} else {
  console.log("Minor");
}
```

 Loops:

-
    - for (let i = 0; i < 5; i++) {
    -   console.log(i);
    - }
    -
    - let j = 0;
    - while (j < 5) {
    -   console.log(j);
    -   j++;
    - }
    -

4. Practical Examples:

Example: Changing Text Content:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Change Text Example</title>
  <script>
    function changeText() {
      document.getElementById('text').innerHTML = "Text has been changed!";
    }
  </script>
</head>
<body>
  <p id="text">This is the original text.</p>
  <button onclick="changeText()">Change Text</button>
</body>
</html>
```

Example: Simple Calculator:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Simple Calculator</title>
  <script>
    function calculate() {
      let num1 = parseFloat(document.getElementById('num1').value);
      let num2 = parseFloat(document.getElementById('num2').value);
      let result = num1 + num2;
      document.getElementById('result').innerHTML = "Result: " + result;
    }
  </script>
</head>
<body>
  <input type="number" id="num1" placeholder="Enter first number">
  <input type="number" id="num2" placeholder="Enter second number">
  <button onclick="calculate()">Calculate</button>
  <p id="result"></p>
</body>
</html>
```

Day 16: JavaScript Functions and Events

Date:25-06-24

---

1. JavaScript Functions:

- **Defining Functions:** Functions are blocks of code designed to perform specific tasks and can be reused throughout the code.

```
 // Function declaration
function greet(name) {
  return "Hello, " + name;
}


// Function expression
const greet = function(name) {
  return "Hello, " + name;
};


// Arrow function (ES6)
const greet = (name) => "Hello, " + name;
```

 Calling Functions:

```
 console.log(greet("Alice")); // Output: Hello, Alice
```

 Parameters and Arguments: Functions can accept parameters and use them within the function body.

```
 function add(a, b) {
  return a + b;
}


console.log(add(5, 3)); // Output: 8
```

 Default Parameters: Parameters can have default values if no arguments are provided.

```
 function greet(name = "Guest") {
  return "Hello, " + name;
}


console.log(greet()); // Output: Hello, Guest
```

 Returning Values: Functions can return values using the return statement.

- function multiply(a, b) {
-   return a * b;
- }
-
- let result = multiply(4, 5);
- console.log(result); // Output: 20
-

2. JavaScript Events:

- **Understanding Events:** Events are actions or occurrences that happen in the browser, such as clicking a button, hovering over an element, or loading a page. JavaScript can be used to listen for and respond to these events.
- **Common Events:**
  - onclick - Triggered when an element is clicked.
  - onmouseover - Triggered when the mouse pointer is over an element.
  - onmouseout - Triggered when the mouse pointer leaves an element.
  - onkeydown - Triggered when a key is pressed down.
  - onkeyup - Triggered when a key is released.
  - onload - Triggered when the page has finished loading.
- **Event Listeners:** Event listeners can be added to HTML elements to handle events.
- // Inline event
- <button onclick="showAlert()">Click Me</button>
-
- // Event listener
- document.getElementById("myButton").addEventListener("click", showAlert);
-

3. Practical Examples:

Example: Button Click Event:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Button Click Event</title>
  <script>
    function showAlert() {
      alert('Button was clicked!');
    }
  </script>
</head>
<body>
  <button onclick="showAlert()">Click Me</button>
</body>
</html>
```

Example: Mouseover and Mouseout Events:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Mouseover and Mouseout Events</title>
  <script>
    function changeColorOnMouseOver(element) {
      element.style.backgroundColor = 'yellow';
    }

    function changeColorOnMouseOut(element) {
      element.style.backgroundColor = '';
    }
  </script>
</head>
<body>
```

```
  <p onmouseover="changeColorOnMouseOver(this)"
onmouseout="changeColorOnMouseOut(this)">
    Hover over this text to change its background color.
  </p>
</body>
</html>
```

Example: Form Validation:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Form Validation</title>
  <script>
    function validateForm() {
      let name = document.getElementById('name').value;
      if (name == '') {
        alert('Name must be filled out');
        return false;
      }
      return true;
    }
  </script>
</head>
<body>
  <form onsubmit="return validateForm()">
    <label for="name">Name:</label>
    <input type="text" id="name" name="name">
    <button type="submit">Submit</button>
  </form>
</body>
```

```
</html>
```

Day 17:CREATE PARRALLEX WEBSITE BY USING CSS AND HTML

DATE:26-06-24

Today was an interesting day! Mam asked us to create tasks specifically related to **Flexbox** and **CSS Grid**, which are two important layout techniques in web design. I was a bit excited and nervous at the same time because these concepts are very powerful but can be tricky to get right.

Tasks Given:

1. **Flexbox Tasks**:
   - We had to create a basic layout using **Flexbox**, arranging items in rows and columns.
   - The main focus was to play with properties like justify-content, align-items, and flex-wrap. I created a simple layout where I had some boxes inside a flex container, adjusting their position both horizontally and vertically.
2. **CSS Grid Tasks**:
   - The second part of the task was focused on **CSS Grid**, where we had to divide a container into rows and columns.
   - The task was to build a responsive grid layout that adjusts itself when the screen size changes. We worked with properties like grid-template-rows, grid-template-columns, and gap. I also learned how to use grid areas for placing items precisely in a grid structure.

Challenges:

- **Flexbox**: It was a bit tricky at first to understand how to align items both horizontally and vertically inside a container. I had to do a lot of testing with align-items and justify-content to get the right positioning.
- **CSS Grid**: Grid felt a little easier once I understood how to define the rows and columns, but getting the grid items to fit properly without overlapping was a challenge. I had to use media queries to make the layout responsive, which was fun but took time.

Code for parrallex website:

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Document</title><style>

  .container{

    height:100vh;
    width:100%;
    background-image: url(https://wallpaperset.com/w/full/b/b/a/461648.jpg);
    background-repeat:no repeat;
    background-position:center;
    background-size:cover;
    background-attachment:fixed;
    background-blend-mode:darken;
    position:relative;




}
.center{

    top:50%;
    left:90px;
    color:white;
    text-align:center;
    font-size:80px;
    font-weight:bold;
    font-family:cursive;
    position:absolute;
    text-decoration-line:underline;
    text-decoration-style:solid;
}
    .box{
        text-align:justify;
        font-size:18px;
        line-height:1.5;
        color:black;
```

```
          background-color:white;
          margin:0 20px;

   }.box5{
        height:100vh;
        width:100%;
        background-image:
url(https://image.tmdb.org/t/p/original/z5VKhNSQKQyxm0co68HAkCqHnmX.jpg);
        background-repeat:no repeat;
        background-position:center;
        background-size:cover;
        background-attachment:fixed;
        background-blend-mode:darken;}
        .box2{
          text-align:justify;
          font-size:18px;
          line-height:1.5;
          color:black;
          background-color:white;
          margin:0 20px;

        }
        .box6{
        height:100vh;
        width:100%;
        background-image:
url(https://image.tmdb.org/t/p/original/p5HxqSZr6WyXt7ARtK3PPuJehKC.jpg);
        background-repeat:no repeat;
        background-position:center;
        background-size:cover;
        background-attachment:fixed;
        background-blend-mode:darken;}
```

```
.box3{
   text-align:justify;
   font-size:18px;
   line-height:1.5;
   color:black;
   background-color:white;
   margin:0 20px;


}.box7{
height:100vh;
width:100%;
background-image: url(https://cdn.wrytin.com/images/wrytup/r/1024/the-exorcist-
jxe4fmz4.jpeg);
background-repeat:no repeat;
background-position:center;
background-size:cover;
background-attachment:fixed;
background-blend-mode:darken;}
.box4{
   text-align:justify;
   font-size:18px;
   line-height:1.5;
   color:black;
   background-color:white;
   margin:0 20px;
}
   .box8{


      height:100vh;
width:100%;
```

```
        background-image: url(
https://www.americamagazine.org/sites/default/files/main_image/2021/03/10/Screen%20Shot%20
2021-03-10%20at%2011.49.34%20AM.png.png);

        background-repeat:no repeat;

        background-position:center;

        background-size:cover;

        background-attachment:fixed;

        background-blend-mode:darken;

    position:relative;}

        .center1{

            top:50%;

            width:100%;

        color:azure;

        text-align:center;

        font-size:80px;

        font-weight:bold;

        font-family:cursive;

        position:absolute;

        text-decoration-style:solid;

        text-decoration-line:underline;


        }

        h1{

            text-align:center;

        }


    </style>

</head>

<body>

    <div class="container"><div class="center">HORROR MOVIES</div></div>

    <div class="box"><h1>1.THE CONJURING</h1>Lorem ipsum dolor sit amet, consectetur adipisicing
elit. Cupiditate eveniet nihil illum nemo ex nam, aut dolorem sunt molestiae minus sint deleniti odit
```

necessitatibus? Minus ut, alias numquam iure vel, facilis labore molestias voluptatibus dolores ex officia tempora ab fugiat. Voluptas atque vero modi deleniti magni dolor mollitia autem aliquam illo commodi maiores cum soluta aut, laudantium minima possimus ex earum eligendi harum! Earum, temporibus. Sunt nisi animi dolorem corrupti nihil assumenda cum labore id. Facere ullam provident repellendus voluptatem doloribus. Laborum ullam possimus veritatis repellat impedit quis repudiandae cupiditate tempore. Nemo accusantium facilis laborum quia vero aspernatur reprehenderit quasi deleniti voluptatem quaerat eos, nihil totam itaque assumenda ad? Iure nihil, odio accusamus eius omnis porro debitis quos, enim architecto cum quis praesentium magni tenetur numquam unde vero veritatis blanditiis? Labore similique delectus illum praesentium aliquid dolor fuga quam, voluptatibus asperiores eius facere velit dignissimos voluptatem sint impedit ab molestiae animi rem sed iure. A rem iusto laudantium sunt. Delectus quo quisquam soluta, nam aperiam sapiente distinctio eaque tempora praesentium doloribus quis ut maiores reprehenderit iure minus blanditiis dolor amet ipsam temporibus quidem. Obcaecati totam quia vel beatae, sunt, reiciendis cumque aperiam aspernatur numquam laudantium non incidunt exercitationem officiis illum expedita ad eveniet! Eveniet exercitationem enim possimus, quos libero nam excepturi neque illum fugiat, earum quod nesciunt veritatis rerum tempore, a quasi deleniti. Vel earum nesciunt consectetur, error dolore minus officia magni quaerat doloribus laborum quisquam aspernatur deserunt iusto debitis numquam tempore vitae deleniti consequuntur. Nostrum a, nesciunt culpa necessitatibus sequi illum quisquam sed delectus ipsam sunt amet obcaecati optio aliquam quibusdam repellendus provident numquam minima nobis tempora quasi praesentium! Esse officia expedita similique distinctio quasi debitis, magnam saepe, architecto laboriosam voluptatem eligendi, laudantium consequuntur incidunt quo ea! Omnis aspernatur quibusdam laborum aliquid inventore, accusamus recusandae porro neque impedit a repellat quaerat saepe sit soluta corrupti sapiente placeat illo enim, provident unde? Culpa quia nesciunt repudiandae minus quo aut cum, doloremque inventore magni esse illum nostrum, voluptas placeat nam eos! Consequuntur cumque vel officiis. Numquam nobis corrupti blanditiis quaerat. Dolores accusantium cum, rem aut saepe nemo iusto exercitationem facere magnam sit voluptatum eius repudiandae doloremque fugiat temporibus corrupti molestias quos quo reiciendis sequi. Excepturi sit non itaque dolore sint consequuntur quas voluptatum magni doloremque sequi est ea, natus, possimus ut pariatur fuga nobis commodi enim harum libero. Dicta totam ipsum est delectus ea culpa, accusamus voluptates eligendi quisquam fugiat sint! </div>

<div class="box5"></div>

<div class="box2"><h1>2.INSIDIOUS</h1>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Cupiditate eveniet nihil illum nemo ex nam, aut dolorem sunt molestiae minus sint deleniti odit necessitatibus? Minus ut, alias numquam iure vel, facilis labore molestias voluptatibus dolores ex officia tempora ab fugiat. Voluptas atque vero modi deleniti magni dolor mollitia autem aliquam illo commodi maiores cum soluta aut, laudantium minima possimus ex earum eligendi harum! Earum, temporibus. Sunt nisi animi dolorem corrupti nihil assumenda cum labore id. Facere ullam provident repellendus voluptatem doloribus. Laborum ullam possimus veritatis repellat impedit quis repudiandae cupiditate tempore. Nemo accusantium facilis laborum quia vero aspernatur reprehenderit quasi deleniti voluptatem quaerat eos, nihil totam itaque assumenda ad? Iure nihil, odio accusamus eius omnis porro debitis quos, enim architecto cum quis praesentium magni tenetur numquam unde vero veritatis blanditiis? Labore similique delectus illum praesentium aliquid dolor fuga quam, voluptatibus asperiores eius facere velit dignissimos voluptatem sint impedit ab molestiae animi rem sed iure. A rem iusto laudantium sunt. Delectus quo quisquam soluta, nam aperiam sapiente distinctio eaque tempora praesentium doloribus quis ut maiores reprehenderit iure minus blanditiis dolor amet ipsam temporibus quidem. Obcaecati totam quia vel beatae, sunt,

reiciendis cumque aperiam aspernatur numquam laudantium non incidunt exercitationem officiis illum expedita ad eveniet! Eveniet exercitationem enim possimus, quos libero nam excepturi neque illum fugiat, earum quod nesciunt veritatis rerum tempore, a quasi deleniti. Vel earum nesciunt consectetur, error dolore minus officia magni quaerat doloribus laborum quisquam aspernatur deserunt iusto debitis numquam tempore vitae deleniti consequuntur. Nostrum a, nesciunt culpa necessitatibus sequi illum quisquam sed delectus ipsam sunt amet obcaecati optio aliquam quibusdam repellendus provident numquam minima nobis tempora quasi praesentium! Esse officia expedita similique distinctio quasi debitis, magnam saepe, architecto laboriosam voluptatem eligendi, laudantium consequuntur incidunt quo ea! Omnis aspernatur quibusdam laborum aliquid inventore, accusamus recusandae porro neque impedit a repellat quaerat saepe sit soluta corrupti sapiente placeat illo enim, provident unde? Culpa quia nesciunt repudiandae minus quo aut cum, doloremque inventore magni esse illum nostrum, voluptas placeat nam eos! Consequuntur cumque vel officiis. Numquam nobis corrupti blanditiis quaerat. Dolores accusantium cum, rem aut saepe nemo iusto exercitationem facere magnam sit voluptatum eius repudiandae doloremque fugiat temporibus corrupti molestias quos quo reiciendis sequi. Excepturi sit non itaque dolore sint consequuntur quas voluptatum magni doloremque sequi est ea, natus, possimus ut pariatur fuga nobis commodi enim harum libero. Dicta totam ipsum est delectus ea culpa, accusamus voluptates eligendi quisquam fugiat sint! </div>

<div class="box6"></div>

<div class="box3"><h1>3.THE EXORCIST</h1>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Cupiditate eveniet nihil illum nemo ex nam, aut dolorem sunt molestiae minus sint deleniti odit necessitatibus? Minus ut, alias numquam iure vel, facilis labore molestias voluptatibus dolores ex officia tempora ab fugiat. Voluptas atque vero modi deleniti magni dolor mollitia autem aliquam illo commodi maiores cum soluta aut, laudantium minima possimus ex earum eligendi harum! Earum, temporibus. Sunt nisi animi dolorem corrupti nihil assumenda cum labore id. Facere ullam provident repellendus voluptatem doloribus. Laborum ullam possimus veritatis repellat impedit quis repudiandae cupiditate tempore. Nemo accusantium facilis laborum quia vero aspernatur reprehenderit quasi deleniti voluptatem quaerat eos, nihil totam itaque assumenda ad? Iure nihil, odio accusamus eius omnis porro debitis quos, enim architecto cum quis praesentium magni tenetur numquam unde vero veritatis blanditiis? Labore similique delectus illum praesentium aliquid dolor fuga quam, voluptatibus asperiores eius facere velit dignissimos voluptatem sint impedit ab molestiae animi rem sed iure. A rem iusto laudantium sunt. Delectus quo quisquam soluta, nam aperiam sapiente distinctio eaque tempora praesentium doloribus quis ut maiores reprehenderit iure minus blanditiis dolor amet ipsam temporibus quidem. Obcaecati totam quia vel beatae, sunt, reiciendis cumque aperiam aspernatur numquam laudantium non incidunt exercitationem officiis illum expedita ad eveniet! Eveniet exercitationem enim possimus, quos libero nam excepturi neque illum fugiat, earum quod nesciunt veritatis rerum tempore, a quasi deleniti. Vel earum nesciunt consectetur, error dolore minus officia magni quaerat doloribus laborum quisquam aspernatur deserunt iusto debitis numquam tempore vitae deleniti consequuntur. Nostrum a, nesciunt culpa necessitatibus sequi illum quisquam sed delectus ipsam sunt amet obcaecati optio aliquam quibusdam repellendus provident numquam minima nobis tempora quasi praesentium! Esse officia expedita similique distinctio quasi debitis, magnam saepe, architecto laboriosam voluptatem eligendi, laudantium consequuntur incidunt quo ea! Omnis aspernatur quibusdam laborum aliquid inventore, accusamus recusandae porro neque impedit a repellat quaerat saepe sit soluta corrupti sapiente placeat illo enim, provident unde? Culpa quia nesciunt repudiandae minus quo aut cum, doloremque inventore magni esse illum nostrum, voluptas placeat nam eos! Consequuntur cumque vel officiis. Numquam nobis corrupti blanditiis quaerat. Dolores accusantium cum, rem aut saepe

```
nemo iusto exercitationem facere magnam sit voluptatum eius repudiandae doloremque fugiat
temporibus corrupti molestias quos quo reiciendis sequi. Excepturi sit non itaque dolore sint
consequuntur quas voluptatum magni doloremque sequi est ea, natus, possimus ut pariatur fuga
nobis commodi enim harum libero. Dicta totam ipsum est delectus ea culpa, accusamus voluptates
eligendi quisquam fugiat sint! </div>

   <div class="box7"></div>


   <div class="box4"><h1>4.THE NUN</h1>Lorem ipsum dolor sit amet, consectetur adipisicing elit.
Cupiditate eveniet nihil illum nemo ex nam, aut dolorem sunt molestiae minus sint deleniti odit
necessitatibus? Minus ut, alias numquam iure vel, facilis labore molestias voluptatibus dolores ex
officia tempora ab fugiat. Voluptas atque vero modi deleniti magni dolor mollitia autem aliquam illo
commodi maiores cum soluta aut, laudantium minima possimus ex earum eligendi harum! Earum,
temporibus. Sunt nisi animi dolorem corrupti nihil assumenda cum labore id. Facere ullam provident
repellendus voluptatem doloribus. Laborum ullam possimus veritatis repellat impedit quis
repudiandae cupiditate tempore. Nemo accusantium facilis laborum quia vero aspernatur
reprehenderit quasi deleniti voluptatem quaerat eos, nihil totam itaque assumenda ad? Iure nihil,
odio accusamus eius omnis porro debitis quos, enim architecto cum quis praesentium magni tenetur
numquam unde vero veritatis blanditiis? Labore similique delectus illum praesentium aliquid dolor
fuga quam, voluptatibus asperiores eius facere velit dignissimos voluptatem sint impedit ab
molestiae animi rem sed iure. A rem iusto laudantium sunt. Delectus quo quisquam soluta, nam
aperiam sapiente distinctio eaque tempora praesentium doloribus quis ut maiores reprehenderit
iure minus blanditiis dolor amet ipsam temporibus quidem. Obcaecati totam quia vel beatae, sunt,
reiciendis cumque aperiam aspernatur numquam laudantium non incidunt exercitationem officiis
illum expedita ad eveniet! Eveniet exercitationem enim possimus, quos libero nam excepturi neque
illum fugiat, earum quod nesciunt veritatis rerum tempore, a quasi deleniti. Vel earum nesciunt
consectetur, error dolore minus officia magni quaerat doloribus laborum quisquam aspernatur
deserunt iusto debitis numquam tempore vitae deleniti consequuntur. Nostrum a, nesciunt culpa
necessitatibus sequi illum quisquam sed delectus ipsam sunt amet obcaecati optio aliquam
quibusdam repellendus provident numquam minima nobis tempora quasi praesentium! Esse officia
expedita similique distinctio quasi debitis, magnam saepe, architecto laboriosam voluptatem
eligendi, laudantium consequuntur incidunt quo ea! Omnis aspernatur quibusdam laborum aliquid
inventore, accusamus recusandae porro neque impedit a repellat quaerat saepe sit soluta corrupti
sapiente placeat illo enim, provident unde? Culpa quia nesciunt repudiandae minus quo aut cum,
doloremque inventore magni esse illum nostrum, voluptas placeat nam eos! Consequuntur cumque
vel officiis. Numquam nobis corrupti blanditiis quaerat. Dolores accusantium cum, rem aut saepe
nemo iusto exercitationem facere magnam sit voluptatum eius repudiandae doloremque fugiat
temporibus corrupti molestias quos quo reiciendis sequi. Excepturi sit non itaque dolore sint
consequuntur quas voluptatum magni doloremque sequi est ea, natus, possimus ut pariatur fuga
nobis commodi enim harum libero. Dicta totam ipsum est delectus ea culpa, accusamus voluptates
eligendi quisquam fugiat sint! </div>

   <div class="box8"><div class="center1">THANKS FOR VISIT US</div></div>

</body>

</html>
```

Day 18: FORM DESIGN FOR ANY GROCERY STORE

Date: 27-06-24

Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Document</title>
   <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.5.2/css/all.min.css">
  <style>*{
   padding:0;
   margin:0;

   box-sizing:border-box;

  }
header{
   height:100vh;
   width:100%;
      background:url(https://c.pxhere.com/photos/2c/e6/supermarket_nice_woma-898411.jpg!d)
no-repeat center/cover fixed;
      opacity:0.8;
      position:relative;
      overflow:hidden;



       } nav .logo{

          font-size:60px;
      transition:all 2s;
```

```
margin-left:50px;
gap:30px;
 }


nav ul {
   display:flex;
   flex-direction:row;
   justify-content:center;
   font-size:25px;
   gap:70px;
   flex-wrap:wrap;
   float:right;
   margin-right:100px;



}
 a{
   text-decoration:none;
   font-weight:bold;

 } a:hover{
   color:grey;
 }


 .box{
height:450px;
width:450px;
border:2px solid yellowgreen;
border-radius:20px;
```

```
        position:absolute;
       top:150px;
       left:150px;
        box-shadow:0 0 8px black;
        text-align:center;
        backdrop-filter:blur(1px);

    }

    h3{
        font-size:30px;
        font-family:'Trebuchet MS', 'Lucida Sans Unicode', 'Lucida Grande', 'Lucida Sans', Arial, sans-
serif;
        text-align:center;
        color:rgb(76, 76, 76);
        margin:0 5px;

    }input[type=text],input[type=tel],textarea{
        text-align:center;
        font-size:20px;
        font-family:'Trebuchet MS', 'Lucida Sans Unicode', 'Lucida Grande', 'Lucida Sans', Arial, sans-
serif;
margin:20px auto;
border:2px solid transparent;
border-radius:20px;
box-shadow:0 0 8px grey;

    }

   </style>
</head>
<body><header>
```

```
  <div class="container">


    <nav>
    <label class="logo"> <i class="fa-solid fa-cart-shopping"></i></label>


        <ul type="none">
      <li><a href="#">HOME</a></li>
      <li><a href="#">ABOUT US</a></li>
      <li><a href="#">GROCERY LIST</a></li>
      <li><a href="#">CONTACT US</a></li> </ul>


  </nav></div></header>
    <div class="box">
      <form action="">
        <h3>Register For The Grocery service</h3>
        <input type="text" placeholder="Enter Your Name"><br/>
        <input type="tel" placeholder=" Enter Mobile Number"><br/>
        <input type="text" placeholder="Enter your Pincode"><br/>
         <textarea placeholder="Enter Full Address"></textarea><br/>


      </form>


  </div>
</body>
</html>
```

Day 19: discussion related to project on HTML and CSS

Date: 28-06-24

Today was a pivotal day for my project. I had a meeting with my teacher to discuss my progress, and it turned out to be really helpful. Here’s how it went:

Preparation:
Before the meeting, I took some time to review the work I've done so far. I made a list of the main points I wanted to address during our discussion:

- What aspects of the project are going well?
- What challenges have I encountered?
- What areas of the project need improvement?
- Clarifications on the project requirements (if any).

I also gathered my project materials—notes, drafts, research, and the outline—so I'd have everything I might need for the meeting.

The Discussion: The conversation was really productive. I first shared my progress and explained what I've completed up to this point. My teacher asked questions to ensure I was on the right track and to see if I'd missed any important details.

Key Points from the Meeting:

- **Strengths of the Project:** My teacher said I'm doing well with the research and the overall structure, especially the way I've organized the information.
- **Areas for Improvement:** One of the areas that needs more work is the analysis section. My teacher suggested that I should go deeper into the interpretation of my findings and explain the significance more clearly.
- **Next Steps:** Based on the feedback, I now know exactly where to focus my energy over the next few days. I need to refine the analysis, double-check my citations, and ensure that the conclusion ties everything together logically.
- **Clarifications:** We also discussed some of the project guidelines, especially the formatting requirements, which I wasn't entirely sure about. Now I have a clear understanding of what is expected.

Personal Reflections:
I feel more confident after today's discussion. I was a bit worried about whether I was heading in the right direction, but my teacher's feedback reassured me that I'm on track. The areas of improvement are manageable, and I now have a clearer roadmap for how to move forward.

I also appreciated the constructive criticism. It's easy to get stuck in your own perspective, but getting an outside point of view has been invaluable.

To-Do List:

- Revise the analysis section.
- Double-check formatting and citations.
- Finish the conclusion.
- Plan a follow-up meeting (if needed).

It feels like everything is starting to come together!

Day 20: Integrating HTML, CSS, and JavaScript basic

Date: 1-07-24

1. Project Requirements:

- Create a webpage that includes a structured HTML layout.
- Apply CSS to style the webpage and make it visually appealing.
- Add JavaScript to introduce interactivity, such as event handling and DOM manipulation.

2. HTML Structure:

- Basic structure with DOCTYPE, head, and body tags.
- Use of semantic elements like <header>, <nav>, <main>, and <footer>.
- Inclusion of elements like paragraphs, headings, lists, images, links, forms, and tables.

Example HTML:

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>My Webpage</title>
 <link rel="stylesheet" href="styles.css">
</head>
<body>
 <header>
  <h1>Welcome to My Webpage</h1>
  <nav>
   <ul>
    <li><a href="#home">Home</a></li>
    <li><a href="#about">About</a></li>
    <li><a href="#contact">Contact</a></li>
   </ul>
  </nav>
 </header>
```

```
<main>
  <section id="home">
    <h2>Home</h2>
    <p>This is the home section.</p>
  </section>
  <section id="about">
    <h2>About</h2>
    <p>This is the about section.</p>
  </section>
  <section id="contact">
    <h2>Contact</h2>
    <form id="contactForm">
      <label for="name">Name:</label>
      <input type="text" id="name" name="name">
      <label for="email">Email:</label>
      <input type="email" id="email" name="email">
      <button type="button" onclick="submitForm()">Submit</button>
    </form>
    <div id="formMessage"></div>
  </section>
  <section id="data">
    <h2>Data Table</h2>
    <table>
      <thead>
        <tr>
          <th>Name</th>
          <th>Age</th>
          <th>City</th>
        </tr>
      </thead>
      <tbody>
```

```
        <tr>
          <td>John Doe</td>
          <td>25</td>
          <td>New York</td>
        </tr>
        <tr>
          <td>Jane Smith</td>
          <td>30</td>
          <td>Los Angeles</td>
        </tr>
      </tbody>
    </table>
  </section>
 </main>
 <footer>
  <p>&copy; 2024 My Webpage</p>
 </footer>
 <script src="script.js"></script>
</body>
</html>
```

3. CSS Styling:

- Use of external CSS to style the webpage.
- Application of layout properties like flexbox and grid.
- Styling of elements such as headers, navigation menus, sections, forms, and tables.

Example CSS (styles.css):

```
body {
 font-family: Arial, sans-serif;
 margin: 0;
 padding: 0;
 box-sizing: border-box;
}
```

```
header {
  background-color: #4CAF50;
  color: white;
  padding: 1em;
  text-align: center;
}

nav ul {
  list-style-type: none;
  padding: 0;
}

nav ul li {
  display: inline;
  margin: 0 10px;
}

nav ul li a {
  color: white;
  text-decoration: none;
}

main {
  padding: 1em;
}

section {
  margin-bottom: 2em;
}
```

```
form {
  display: flex;
  flex-direction: column;
}

form label, form input, form button {
  margin-bottom: 10px;
}

table {
  width: 100%;
  border-collapse: collapse;
}

table, th, td {
  border: 1px solid black;
}

th, td {
  padding: 8px;
  text-align: left;
}

footer {
  background-color: #333;
  color: white;
  text-align: center;
  padding: 1em;
  position: fixed;
  width: 100%;
  bottom: 0;
}
```

LAST DAY: Some tasks for revision:

Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Document</title>
   <style>*{
      box-sizing:border-box;


   }
      body{
      background: #30E8BF;  /* fallback for old browsers */
background: -webkit-linear-gradient(to right, #FF8235, #30E8BF);  /* Chrome 10-25, Safari 5.1-6 */
background: linear-gradient(to right, #FF8235, #30E8BF); /* W3C, IE 10+/ Edge, Firefox 16+, Chrome
26+, Opera 12+, Safari 7+ */


      }.container{
      height:500px;
      width:450px;
      border:7px solid;
      border-radius:20px;
      position:absolute;
      top:50%;
      left:50%;
      transform:translate(-50%,-50%);
      perspective:900px;
     }
     .container .box-img{
```

```
  height:100%;
  width:100%;
  background-color:aliceblue;
  border-radius:20px;
  position:absolute;

}
.box-img img{
  height:100%;
  width:100%;
  opacity:0.9;
  border-radius:20px;

}
.container .box1{
  height:100%;
  width:100%;
  border-radius:20px;
 transition:all 1s;
 border:5px solid;
 position:absolute;
 top:-100%;
 opacity:0;
 transition:all 0.5s;
 background-color:lightcyan;
}
.container:hover .box1{
  opacity:1;
  top:0;
}
```

```
   h1{
      text-align:center;
      font-size:60px;
      font-family:cursive;
      color:black;

   }p{
      line-height:1.5;
      color:black;
      font-family:cursive;
      font-size:20px;
   }

   </style>



</head>
<body>
   <div class="container">
      <div class="box-img"><img src="http://wallpapercave.com/wp/Qz9p9vd.jpg" alt=""></div>
      <div class="box1"><h1>PINK ROSE</h1><p>Lorem ipsum dolor sit amet consectetur adipisicing
elit. Aliquam voluptatum aperiam laboriosam accusantium! Pariatur quam eum aliquam maxime
facilis quis officiis eius eos voluptate, quas veritatis qui odit rem consectetur exercitationem aut
magnam, repellendus corrupti. Fuga fugit blanditiis fugiat in?</p></div>

      </div>
   </body>
</html>
```

CODE:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Document</title>
   <style>
  .container{
   height:100vh;
   width:100%;
   box-sizing:border-box;
   position:relative;
   animation:change 10s linear 0s infinite;


  }
  @keyframes change{
   0%{
      background:url(https://tse2.mm.bing.net/th?id=OIP.vuFRpwPLqeVOKkfYARhKYgHaEo&pid=Api
&P=0&h=180) no-repeat center/cover;
     transition:all 2s;
     opacity:0.8;
   }50%{


background:url(https://wallpaperaccess.com/full/36301.jpg) no-repeat center/cover;
transition:all 2s;
opacity:0.5;
   }100%{
      background:url(https://images.pexels.com/photos/459225/pexels-photo-
459225.jpeg?cs=srgb&dl=daylight-environment-forest-459225.jpg&fm=jpg) no-repeat center/cover;
      transition:all 2s;
      opacity:0.7;
   }
  }



img{
  height:250px;
  width:250px;
   animation:bird 10s linear 0s infinite;
}@keyframes bird{
   0%{
      position:absolute;
      top:0px;
      left:100px;

   }20%{
      position:absolute;
      top:0px;
      left:300px}
      40%{
      position:absolute;
      top:0px;
      left:700px;}

   60%{
      position:absolute;
      top:0px;
      left:1100px;}

   80%{
      position:absolute;
      top:0px;
```

```
        left:1500px;
    }100%{
        position:absolute;
        top:0px;
        right:-300px;

    }
}

    </style>
</head>
<body>
    <div class="container">

        <div class="img2"> <img src="./bird.gif.crdownload" height="300px" width="300px">



    </div>
</body>
</html>
```

MY PROJECT:

HTML SECTION:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>web vrindavan</title>
    <link rel="stylesheet" href="style1.css">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.3.0/css/all.min.css">
    <link href="https://unpkg.com/aos@2.3.1/dist/aos.css" rel="stylesheet">
    <script src="https://unpkg.com/aos@2.3.1/dist/aos.js"></script>
  </head>
  <body>
    <script>
      AOS.init();
    </script>
    <header>
      <nav class="navbar">
        <div class="logo">
          <img src="vhb.webp" alt="">
      </div>
        <input type="checkbox" id="menu-toggler">
        <label for="menu-toggler" id="hamburger-btn">

        </label>
        <ul class="all-links">
          <li><a href="#home">Home</a></li>
          <li><a href="#services">Services</a></li>
          <li><a href="#about-section">About Us</a></li>
          <li><a href="#contact">Contact Us</a></li>
        </ul>
      </nav>
    </header>

    <section class="homepage" id="home">
      <div class="content">
        <div class="text">
          <h1>WELCOME TO OUR VRINDAVAN</h1>
          <p>
            It is a place of pilgrimage, devotion, and spiritual learning, which attracts people from all over
the world. <br> Vrindavan, considered to be the place where Lord Krishna spent his childhood, has
for centuries been a refuge for widowed Hindu women. Shunned by society, the widows of
Vrindavan live difficult lives, dependent on charity from the State, NGOs and the many temples and
ashrams of the holy town.</p>

        </div>

        <a href="#services">Our Services</a>

      </div>

    </section>


    <section class="services" id="services">

      <h2>Our Services</h2>

      <p>Explore our wide range of services.</p>

      <ul class="cards">


        <li class="card">

          <div data-aos="zoom-in-right">

          <img src="PREM1.jpg" alt="img">

          <h3>MATHURA</h3>

          <p>Mathura is a sacred city in the state of Uttar Pradesh, India, and is considered the
birthplace of Lord Krishna. </p>

        </div>

        </li>


        <li class="card">

          <div data-aos="zoom-in-left">

          <img src="PREM2.jpg" alt="img">

          <h3>VRINDAVAN</h3>

          <p> It is one of the most sacred places for Vaishnavism tradition.</p>

        </div>

        </li>

        <li class="card">
```

```
           <div data-aos="zoom-in">
           <img src="PREM3.jpeg" alt="img">
           <h3>BARSANA</h3>
           <p>Barsana is a village in the Braj region of Uttar Pradesh, India and is known as the birthplace
of Radha, the beloved of Lord Krishna.</p>
         </div>
         </li>
         <li class="card">
           <div data-aos="zoom-in">
           <img src="PREM4.jpeg" alt="img">
           <h3>YAMUNA</h3>
           <p>The Yamuna River is a major river in northern India that flows through Uttarakhand and
Uttar Pradesh, and is a tributary of the Ganges River.</p>
         </div>
         </li>
         <li class="card">
           <div data-aos="zoom-in">
           <img src="PREM5.jpeg" alt="img">
           <h3>BANKEBIHARI</h3>
           <p> a Hindu temple in Vrindavan, Uttar Pradesh, India, dedicated to Lord Krishna.</p>
         </div>
         </li>
         <li class="card">
           <div data-aos="zoom-in">
           <img src="PREM6.JPG" alt="img">
           <h3>PREM MANDIR</h3>
           <p>dedicated to the major events of Lord Krishna's life, and the interiors depict those scenes,
including the raising of Govardhan Hill.</p>
         </div>
         </li>
       </ul>
```

```
</section>
<div id="about-section">
  <h1>About Us </h1>
</div>

<h2 style="text-align:center">Our Team</h2>
<div class="row">
  <div class="column">
    <div class="card">
      <img src="./mooney.jpg" alt="mooney" style="width:100%">
      <div class="container">
        <h2>beth mooney</h2>
        <p class="title">CEO & Founder</p>
        <p>Some text that describes me lorem ipsum ipsum lorem.</p>
        <p>jane@example.com</p>
        <p><button class="button">Contact</button></p>
      </div>
    </div>
  </div>

  <div class="column">
    <div class="card">
      <img src="./mike.jpg_large" alt="Mike" style="width:100%">
      <div class="container">
        <h2>Mike Ross</h2>
        <p class="title">Art Director</p>
        <p>Some text that describes me lorem ipsum ipsum lorem.</p>
        <p>mike@example.com</p>
        <p><button class="button">Contact</button></p>
      </div>
    </div>
  </div>


  <div class="column">
   <div class="card">
    <img src="./mbappe.jpg" alt="John" style="width:100%">
    <div class="container">
     <h2>John Doe</h2>
     <p class="title">Designer</p>
     <p>Some text that describes me lorem ipsum ipsum lorem.</p>
     <p>john@example.com</p>
     <p><button class="button">Contact</button></p>
    </div>
   </div>
  </div>
</div>
<section class="contact" id="contact">
  <h2>Contact Us</h2>
  <p>Reach out to us for any inquiries or feedback.</p>
  <div class="row">
   <div class="col information">
    <div class="contact-details">
     <p><i class="fas fa-map-marker-alt"></i> 123 Campsite Avenue, Wilderness, CA 98765</p>
     <p><i class="fas fa-envelope"></i> info@campinggearexperts.com</p>
     <p><i class="fas fa-phone"></i> (123) 456-7890</p>
     <p><i class="fas fa-clock"></i> Monday - Friday: 9:00 AM - 5:00 PM</p>
     <p><i class="fas fa-clock"></i> Saturday: 10:00 AM - 3:00 PM</p>
     <p><i class="fas fa-clock"></i> Sunday: Closed</p>
     <p><i class="fas fa-globe"></i> www.codingenpalweb.com</p>
    </div>
   </div>
   <div class="col form">
```

```
      <form>
        <input type="text" placeholder="Name*" required>
        <input type="email" placeholder="Email*" required>
        <textarea placeholder="Message*" required></textarea>
        <button id="submit" type="submit">Send Message</button>
      </form>
    </div>
  </div>
</section>

<footer>
  <div class="footer-row">
      <div class="footer-col">
        <h4>Info</h4>
        <ul class="links">
          <li><a href="#">About Us</a></li>
          <li><a href="#">Compressions</a></li>
          <li><a href="#">Customers</a></li>
          <li><a href="#">Service</a></li>
          <li><a href="#">Collection</a></li>
        </ul>
      </div>

      <div class="footer-col">
        <h4>Explore</h4>
        <ul class="links">
          <li><a href="#">Free Designs</a></li>
          <li><a href="#">Latest Designs</a></li>
          <li><a href="#">Themes</a></li>
          <li><a href="#">Popular Designs</a></li>
          <li><a href="#">Art Skills</a></li>
      <li><a href="#">New Uploads</a></li>
    </ul>
  </div>

  <div class="footer-col">
    <h4>Legal</h4>
    <ul class="links">
      <li><a href="#">Customer Agreement</a></li>
      <li><a href="#">Privacy Policy</a></li>
      <li><a href="#">GDPR</a></li>
      <li><a href="#">Security</a></li>
      <li><a href="#">Testimonials</a></li>
      <li><a href="#">Media Kit</a></li>
    </ul>
  </div>

  <div class="footer-col">
    <h4>Newsletter</h4>
    <p>
      Subscribe to our newsletter for a weekly dose
      of news, updates, helpful tips, and
      exclusive offers.
    </p>
    <form action="#">
      <input type="text" placeholder="Your email" required>
      <button type="submit">SUBSCRIBE</button>
    </form>
    <div class="icons">
      <i class="fa-brands fa-facebook-f"></i>
      <i class="fa-brands fa-twitter"></i>
      <i class="fa-brands fa-linkedin"></i>
```

```
              <i class="fa-brands fa-github"></i>
            </div>
          </div>
        </div>
      </footer>

  </body>
</html>
```

CSS SECTION:

```
@import
url("https://fonts.googleapis.com/css2?family=Poppins:wght@200;300;400;500;600;700&display=s
wap");

* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  font-family: "Oswald", sans-serif;
}

html {
  scroll-behavior: smooth;
}
.content img{
  height:400px;
  width: 400px;
}

body {
  background: #f8f5f5;
}
```

```
header {
  position: fixed;
  top: 0;
  left: 0;
  z-index: 5;
  width: 100%;
  display: flex;
  justify-content: center;
  background:rgb(251, 249, 249);
  backdrop-filter: blur(50px);
}

.navbar {
  display: flex;
  padding: 0 10px;
  max-width: 1200px;
  width: 100%;
  align-items: center;
  justify-content: space-between;
}

.navbar input#menu-toggler {
  display: none;
}

.navbar #hamburger-btn {
  cursor: pointer;
  display: none;
}
```

```
.navbar .all-links {
  display: flex;
  align-items: center;
}

.navbar .all-links li {
  position: relative;
  list-style: none;
}

.logo{
    margin:10px 34px;
 }
 .logo img{
     height:40px;
     margin:3px 6px;
 }

header a, footer a {
  margin-left: 40px;
  text-decoration: none;
  color: #0a0a0a;
  height: 100%;
  padding: 20px 0;
  display: inline-block;
}

header a:hover, footer a:hover {
  color: #0519fc;
}
```

```
.homepage {
  height: 100vh;
  width: 100%;
  position: relative;
  background: url(prem.jpg);
  background-position: center 65%;
  background-size: cover;
}

.homepage::before {
  content: "";
  position: absolute;
  left: 0;
  top: 0;
  height: 100%;
  width: 100%;
  background: rgba(0, 0, 0, 0.2);
}

.homepage .content {
  display: flex;
  height: 85%;
  z-index: 3;
  align-items: center;
  justify-content: center;
  flex-direction: column;
}

.homepage .content h1 {
  font-size: 60px;
  font-weight: 700;
  margin-bottom: 10px;
}

.homepage .content .text {
  margin-bottom: 50px;
  color: #fff;
  font-size: 20px;
  text-align: center;
  text-shadow: 0px 0px 10px rgba(0, 0, 0, 0.3);
}

.content a {
  color: #000;
  display: block;
  text-transform: uppercase;
  font-size: 18px;
  margin: 0 10px;
  padding: 10px 30px;
  border-radius: 5px;
  background: #fff;
  border: 2px solid #fff;
  transition: 0.4s ease;
  box-shadow: 0 10px 20px rgba(0, 0, 0, 0.3);
  text-decoration: none;
}

.content a:hover {
  color: #fff;
  background: rgba(255,255,255,0.3);
}
```

```
section {
 display: flex;
 align-items: center;
 flex-direction: column;
 padding: 80px 0 0;
}

section h2 {
 font-size: 2rem;
}

section > p {
 text-align: center;
}

section .cards {
 display: flex;
 flex-wrap: wrap;
 max-width: 1200px;
 margin-top: 50px;
 padding: 0 10px;
 justify-content: space-between;
}
.column {
   float: left;
   width: 33.3%;
   margin-bottom: 16px;
   padding: 0 8px;
 }

 .card {
   box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2);
   margin: 8px;
 }

 #about-section {
   padding: 50px;
   text-align: center;
   background-color: #474e5d;
   color: white;
 }

 .container {
   padding: 0 16px;
 }

 .container::after, .row::after {
   content: "";
   clear: both;
   display: table;
 }

 .title {
   color: grey;
 }

 .button {
   border: none;
   outline: 0;
   display: inline-block;
   padding: 8px;
   color: white;
```

```
    background-color: #000;
    text-align: center;
    cursor: pointer;
    width: 100%;
  }

  .button:hover {
    background-color: #555;
  }

section .cards .card {
  background: #fff;
  padding: 40px 15px;
  list-style: none;
  border-radius: 5px;
  box-shadow: 0px 5px 10px rgba(0, 0, 0, 0.04);
  margin-bottom: 40px;
  width: calc(100% / 3 - 30px);
  text-align: center;
}



.services .card img {
  width: 120px;
  height: 120px;
  margin-bottom: 20px;
  border-radius: 100%;
  object-fit: cover;
}
```

```
.cards .card p {
  padding: 0 15px;
  margin-top: 5px;
}

.about .row {
  padding: 0 10px;
}

.contact .row {
  margin: 60px 0 90px;
  display: flex;
  max-width: 1200px;
  width: 100%;
  align-items: center;
  justify-content: space-between;
}

.contact .row .col {
  padding: 0 10px;
  width: calc(100% / 2 - 50px);
}

.contact .col p {
  margin-bottom: 10px;
}

.contact .col p i {
  color: #7a7a7a;
  margin-right: 10px;
}

.contact form input {
  height: 45px;
  margin-bottom: 20px;
  padding: 10px;
  width: 100%;
  font-size: 16px;
  outline: none;
  border: 1px solid #bfbfbf;
}

.contact form textarea {
  padding: 10px;
  width: 100%;
  font-size: 16px;
  height: 150px;
  outline: none;
  resize: vertical;
  border: 1px solid #bfbfbf;
}

.contact form button {
  margin-top: 10px;
  padding: 10px 20px;
  font-size: 17px;
  color: #fff;
  border: none;
  cursor: pointer;
  border-radius: 5px;
  background: #333;
  transition: 0.2s ease;
}

.contact form button:hover {
  background: #525252;
}

footer {
    background: #10182F;
 }

footer .footer-row {
display: flex;
flex-wrap: wrap;
justify-content: space-between;
gap: 3.5rem;
padding: 60px;
}

.footer-row .footer-col h4 {
color: #fff;
font-size: 1.2rem;
font-weight: 400;
}
.footer-col h4{
    margin-left: 50px;
}

.footer-col .links {
margin-top: 20px;
```

```
}

.footer-col .links li {
list-style: none;
margin-bottom: 10px;
}

.footer-col .links li a {
text-decoration: none;
color: #bfbfbf;
}

.footer-col .links li a:hover {
color: #fff;
}

.footer-col p {
margin: 20px 0;
color: #bfbfbf;
max-width: 300px;
}

.footer-col form {
display: flex;
gap: 5px;
}

.footer-col input {
height: 40px;
border-radius: 6px;
background: none;
```

```
width: 100%;
outline: none;
border: 1px solid #7489C6 ;
caret-color: #fff;
color: #fff;
padding-left: 10px;
}

.footer-col input::placeholder {
color: #ccc;
}

.footer-col form button {
background: #fff;
outline: none;
border: none;
padding: 10px 15px;
border-radius: 6px;
cursor: pointer;
font-weight: 500;
transition: 0.2s ease;
}

.footer-col form button:hover {
background: #cecccc;
}

.footer-col .icons {
display: flex;
margin-top: 30px;
gap: 30px;
```

```
cursor: pointer;
}

.footer-col .icons i {
color: #afb6c7;
}

.footer-col .icons i:hover  {
color: #fff;
}

@media screen and (max-width: 860px) {
 .navbar .all-links {
   position: fixed;
   left: -100%;
   width: 300px;
   display: block;
   height: 100vh;
   top: 75px;
   background: #333;
   transition: left 0.3s ease;
 }

 .navbar #menu-toggler:checked~.all-links {
   left: 0;
 }

 .navbar .all-links li {
   font-size: 18px;
 }
```

```
.navbar #hamburger-btn {
  display: block;
}

section > p {
  text-align: center;
}

section .cards .card {
  width: calc(100% / 2 - 15px);
  margin-bottom: 30px;
}

.homepage .content h1 {
  font-size: 40px;
  font-weight: 700;
  margin-bottom: 10px;
}

.homepage .content .text {
  font-size: 17px;
}

.content a {
  font-size: 17px;
  padding: 9px 20px;
}

.contact .row {
  flex-direction: column;
}
```

```
 .contact .row .col {
   width: 100%;
 }

 .contact .row .col:last-child {
   margin-top: 40px;
 }

 footer a {
   height: 0;
 }
}
@media screen and (max-width: 650px) {
   .column {
     width: 100%;
     display: block;
   }
 }
```