INTERNSHIP REPORT

WEEK 3 DAY 4

Submitted to:	**Ali Hyder**	Submission Date:	**10TH July, 2025**
Internship Domain:	**Front Development**	Internship Name:	**ProSensia**
Student Name:	**Yasal Qamar**	Roll No.	**S25031**

CSS Pseudo-classes & Pseudo-elements

Objective

To explore and implement **CSS pseudo-classes and pseudo-elements** for enhancing the user experience and styling of web elements without modifying the HTML structure. The goal was to learn how to add dynamic and decorative behavior using only CSS.

Introduction:

CSS pseudo-classes (:hover, :nth-child(), :first-child, etc.) apply styles based on an element's **state or position**. Pseudo-elements (::before, ::after, ::first-letter, etc.) target **specific parts** of elements or add **virtual content** to enhance design.

Using these techniques, developers can create interactive, accessible, and visually appealing web interfaces improving both **functionality** and **design**.

Key Learnings:

Pseudo-Classes:

- **:link, :visited**: Used for styling normal and visited links.
- **:hover, :active**: Used to provide interactivity on user actions like hovering and clicking.
- **:first-child, :last-child, :nth-child(n**): Used to style specific list items based on position.
- Applied dynamic styles to list items and buttons based on user interaction.

Pseudo-Elements:

- **::first-letter, ::first-line:** Styled specific parts of text content for visual emphasis.
- **::before, ::after:** Inserted symbols and emojis before/after elements for decorative design.

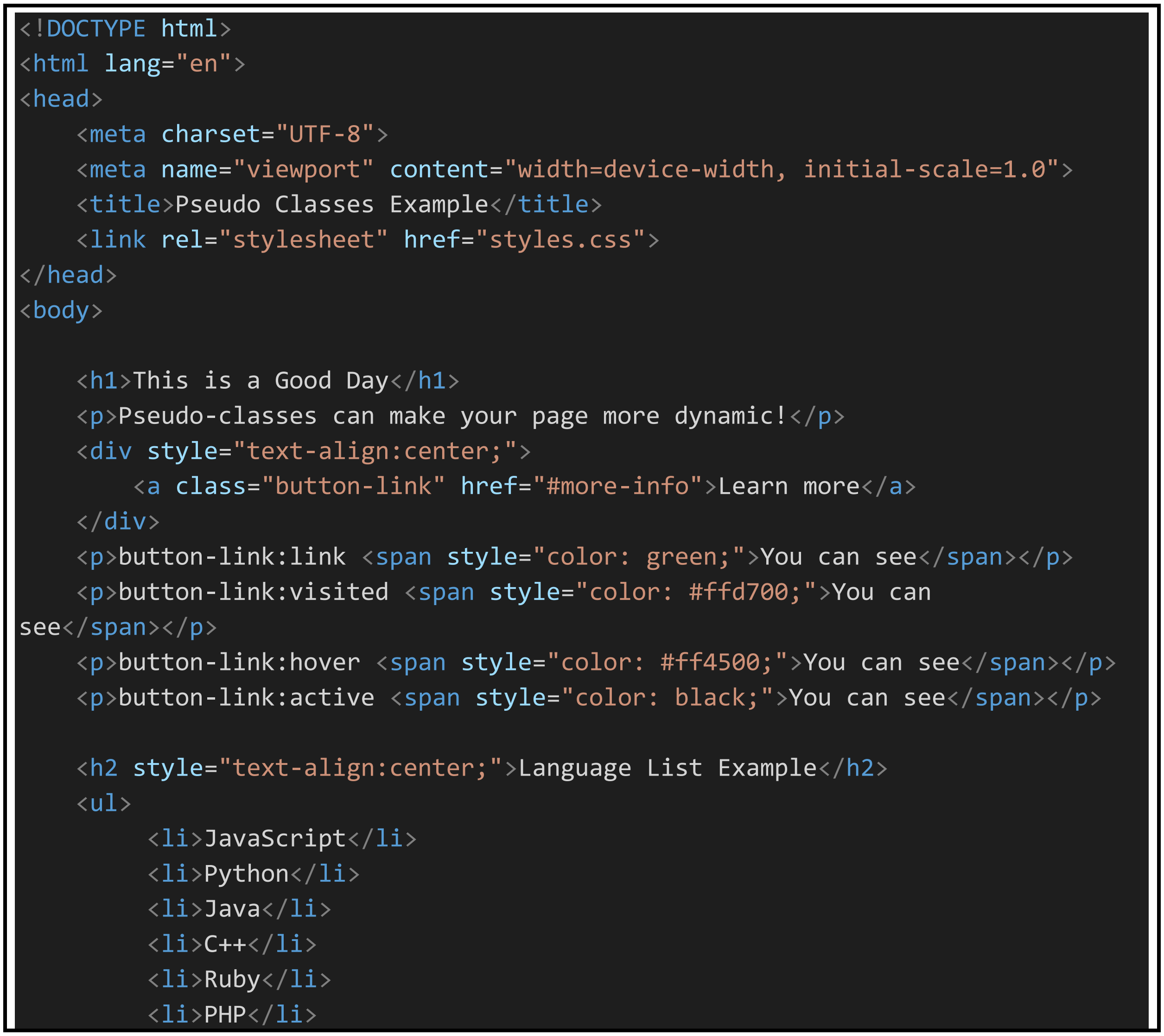
- Demonstrated how content can be styled and inserted without altering HTML.

Practical Implementation:

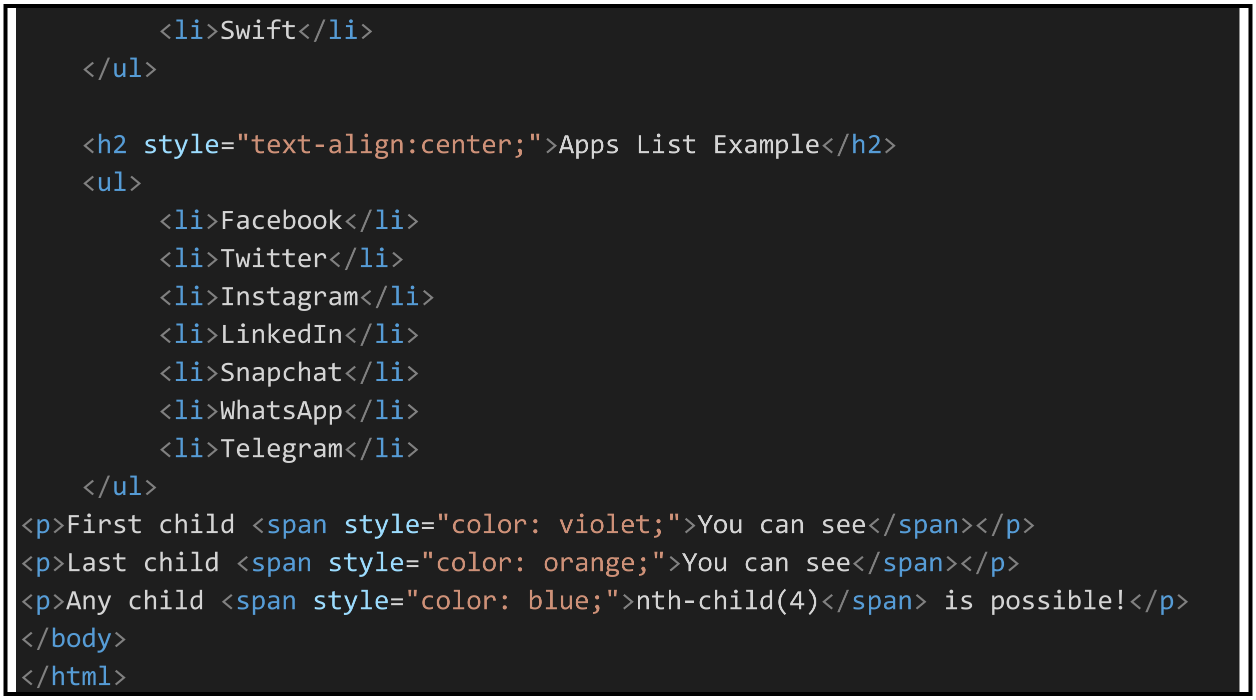
- I developed a combined HTML + CSS webpage.
- Implemented both pseudo-classes and pseudo-elements in lists, buttons, headings, and paragraphs.
- Observed how design can improve usability and aesthetics using minimal code.

1. CSS PSEUDO-CLASSES:

.HTML

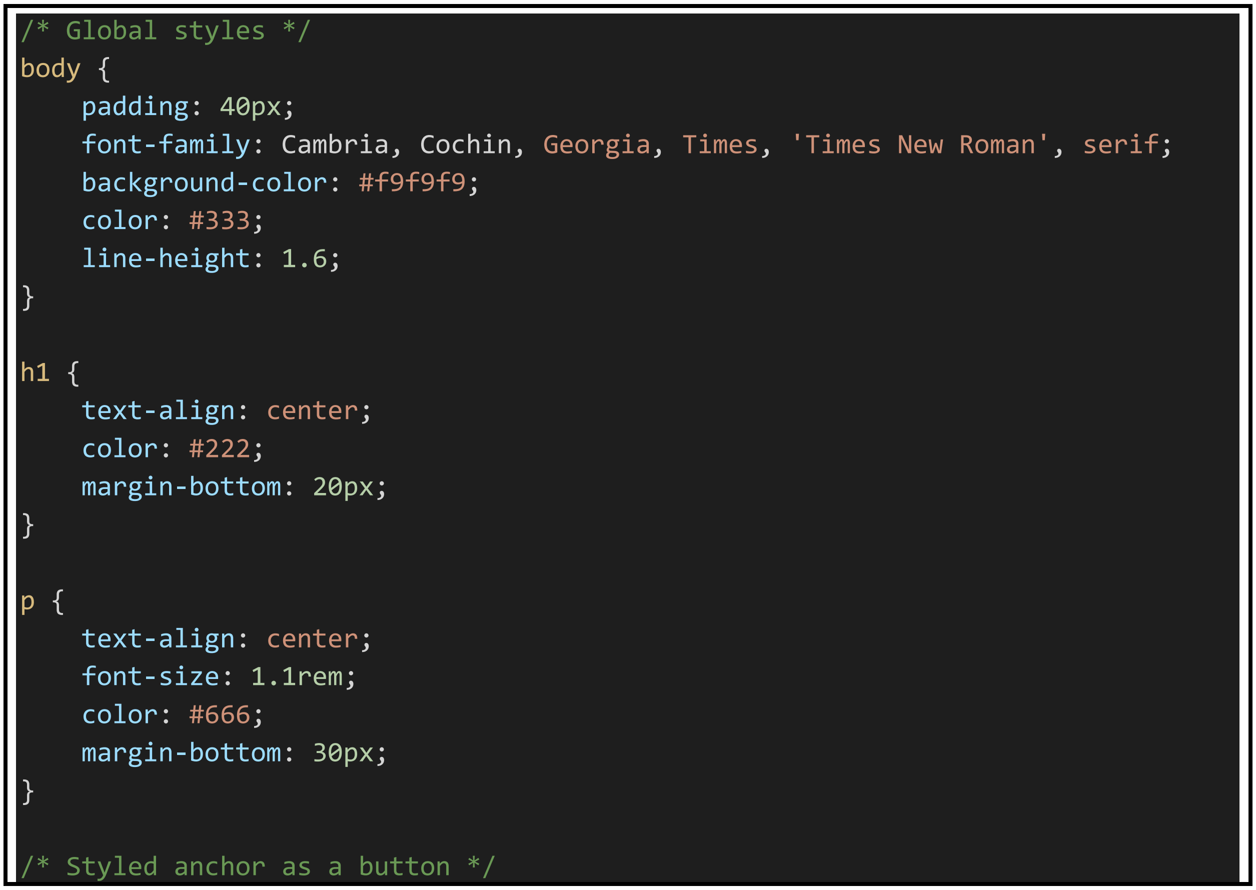


```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Pseudo Classes Example</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>

    <h1>This is a Good Day</h1>
    <p>Pseudo-classes can make your page more dynamic!</p>
    <div style="text-align:center;">
        <a class="button-link" href="#more-info">Learn more</a>
    </div>
    <p>button-link:link <span style="color: green;">You can see</span></p>
    <p>button-link:visited <span style="color: #ffd700;">You can
see</span></p>
    <p>button-link:hover <span style="color: #ff4500;">You can see</span></p>
    <p>button-link:active <span style="color: black;">You can see</span></p>

    <h2 style="text-align:center;">Language List Example</h2>
    <ul>
         <li>JavaScript</li>
         <li>Python</li>
         <li>Java</li>
         <li>C++</li>
         <li>Ruby</li>
         <li>PHP</li>
```

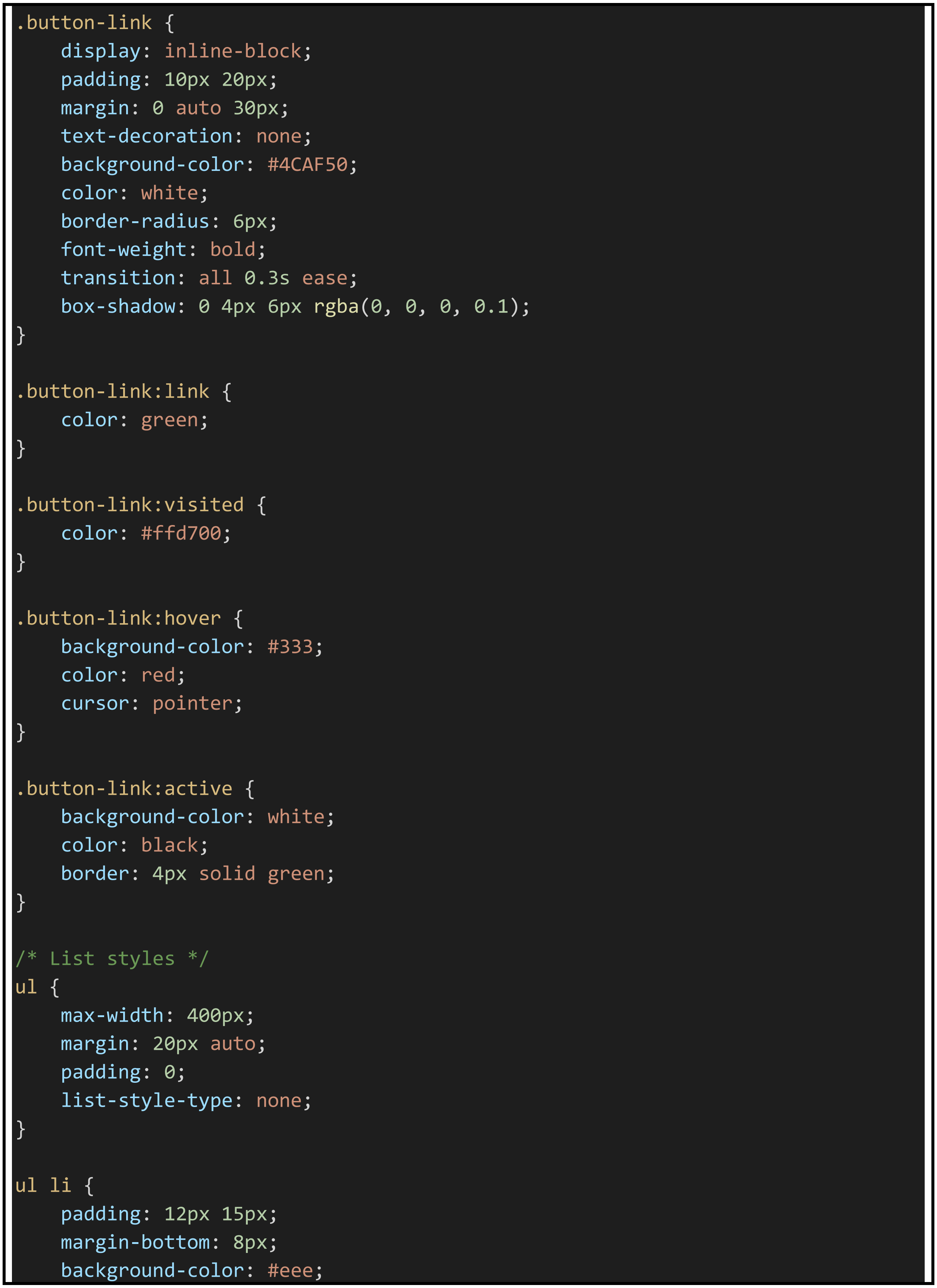


```html
         <li>Swift</li>
    </ul>

    <h2 style="text-align:center;">Apps List Example</h2>
    <ul>
         <li>Facebook</li>
         <li>Twitter</li>
         <li>Instagram</li>
         <li>LinkedIn</li>
         <li>Snapchat</li>
         <li>WhatsApp</li>
         <li>Telegram</li>
    </ul>
<p>First child <span style="color: violet;">You can see</span></p>
<p>Last child <span style="color: orange;">You can see</span></p>
<p>Any child <span style="color: blue;">nth-child(4)</span> is possible!</p>
</body>
</html>
```

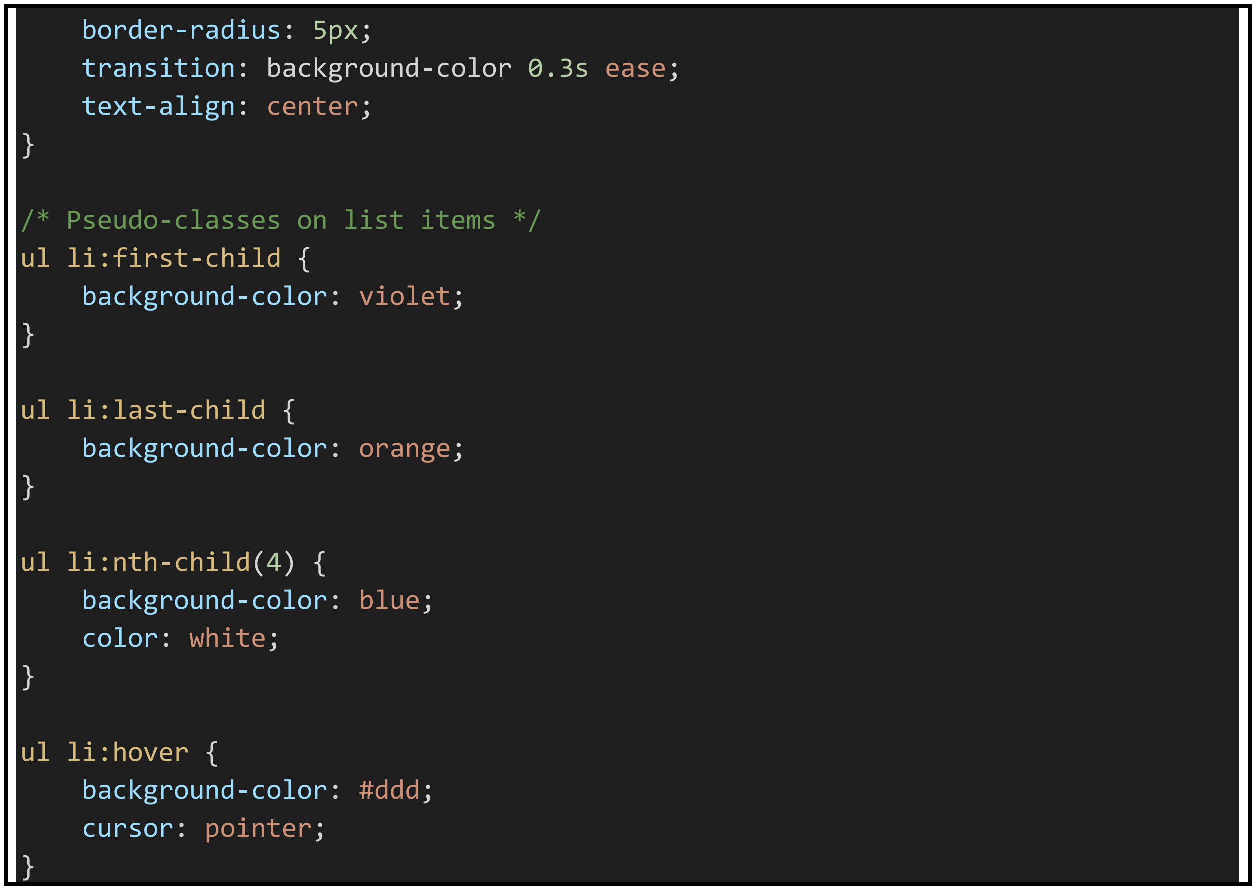
.CSS



```css
/* Global styles */
body {
    padding: 40px;
    font-family: Cambria, Cochin, Georgia, Times, 'Times New Roman', serif;
    background-color: #f9f9f9;
    color: #333;
    line-height: 1.6;
}

h1 {
    text-align: center;
    color: #222;
    margin-bottom: 20px;
}

p {
    text-align: center;
    font-size: 1.1rem;
    color: #666;
    margin-bottom: 30px;
}

/* Styled anchor as a button */
```

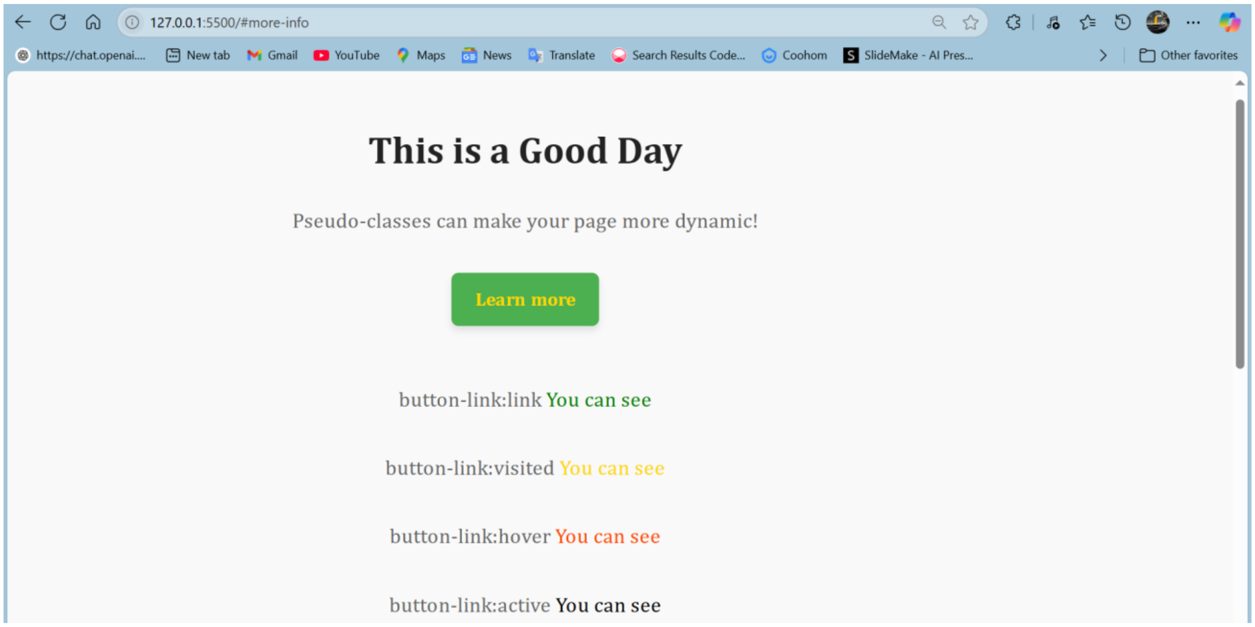


```css
.button-link {
    display: inline-block;
    padding: 10px 20px;
    margin: 0 auto 30px;
    text-decoration: none;
    background-color: #4CAF50;
    color: white;
    border-radius: 6px;
    font-weight: bold;
    transition: all 0.3s ease;
    box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
}

.button-link:link {
    color: green;
}

.button-link:visited {
    color: #ffd700;
}

.button-link:hover {
    background-color: #333;
    color: red;
    cursor: pointer;
}

.button-link:active {
    background-color: white;
    color: black;
    border: 4px solid green;
}

/* List styles */
ul {
    max-width: 400px;
    margin: 20px auto;
    padding: 0;
    list-style-type: none;
}

ul li {
    padding: 12px 15px;
    margin-bottom: 8px;
    background-color: #eee;
```



```
    border-radius: 5px;
    transition: background-color 0.3s ease;
    text-align: center;
}

/* Pseudo-classes on list items */
ul li:first-child {
    background-color: violet;
}

ul li:last-child {
    background-color: orange;
}

ul li:nth-child(4) {
    background-color: blue;
    color: white;
}

ul li:hover {
    background-color: #ddd;
    cursor: pointer;
}
```

Output:



This is a Good Day

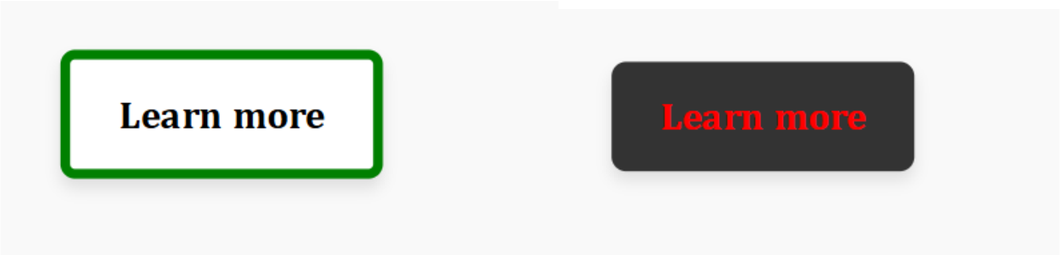
Pseudo-classes can make your page more dynamic!

button-link:link You can see

button-link:visited You can see

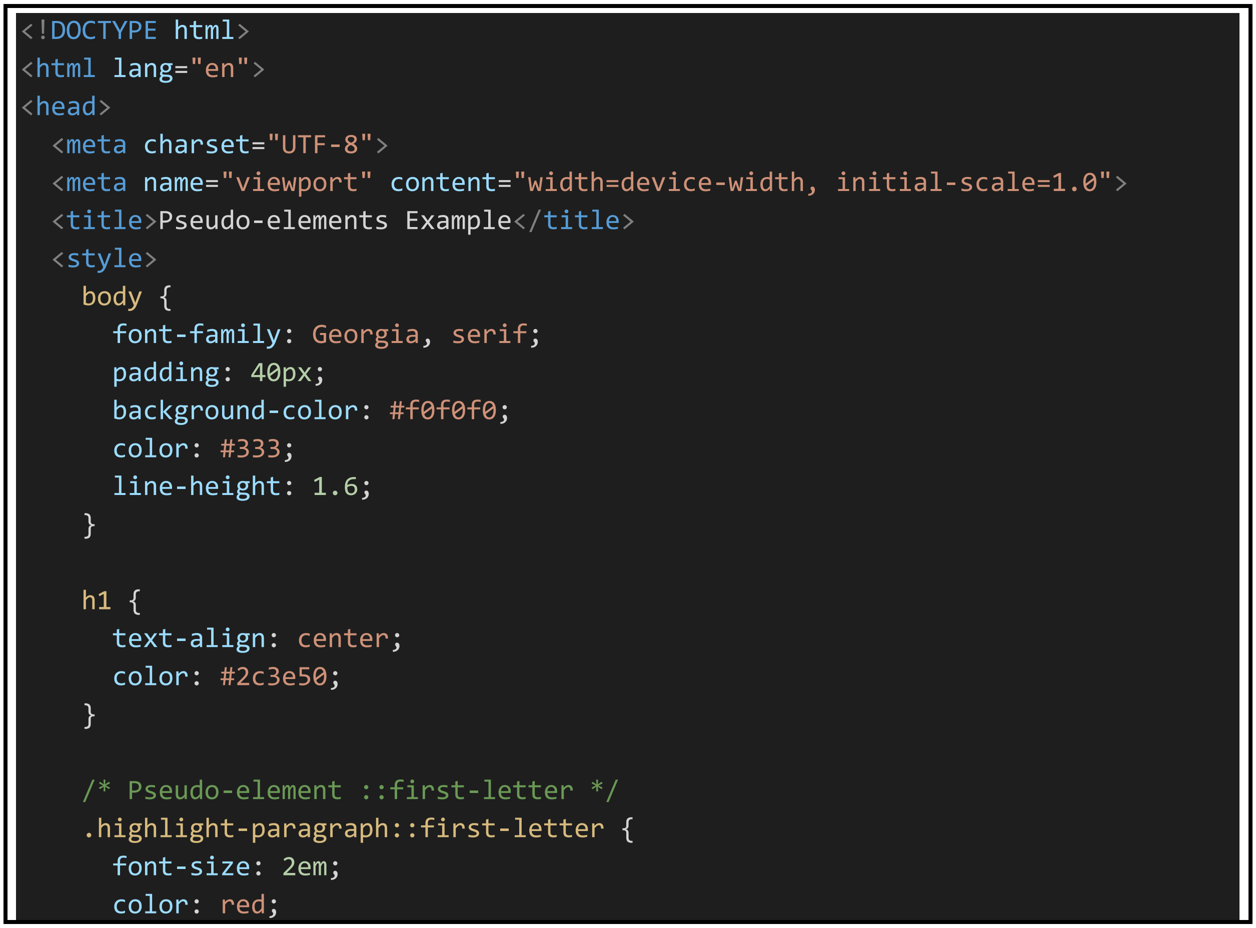
button-link:hover You can see

button-link:active You can see

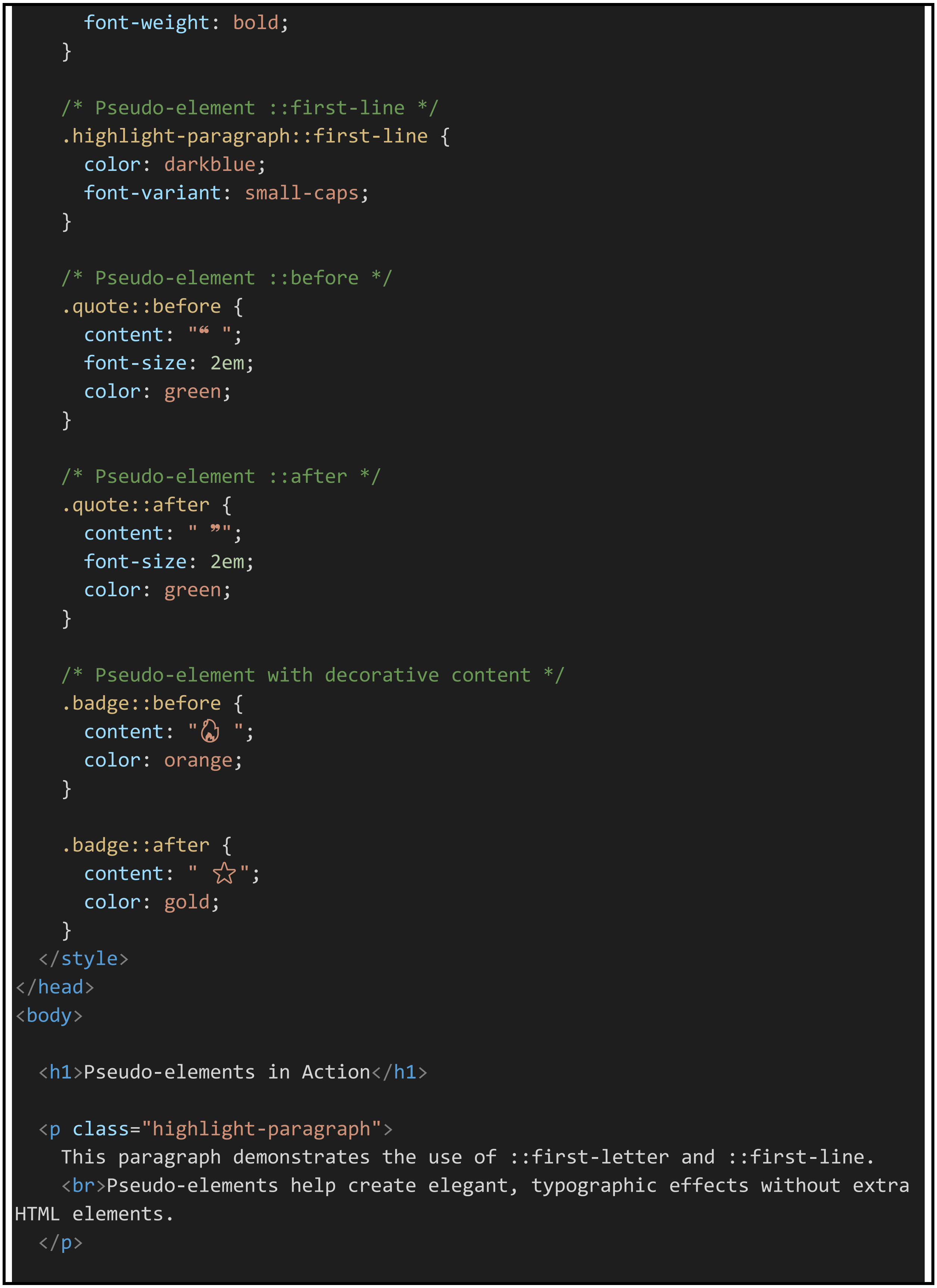


Pseudo-class	Function	Example in your code
:link	Styles unvisited links	.button-link:link
:visited	Styles visited links	.button-link:visited
:hover	When the mouse is over the element	.button-link:hover, li:hover
:active	When the element is being clicked	.button-link:active

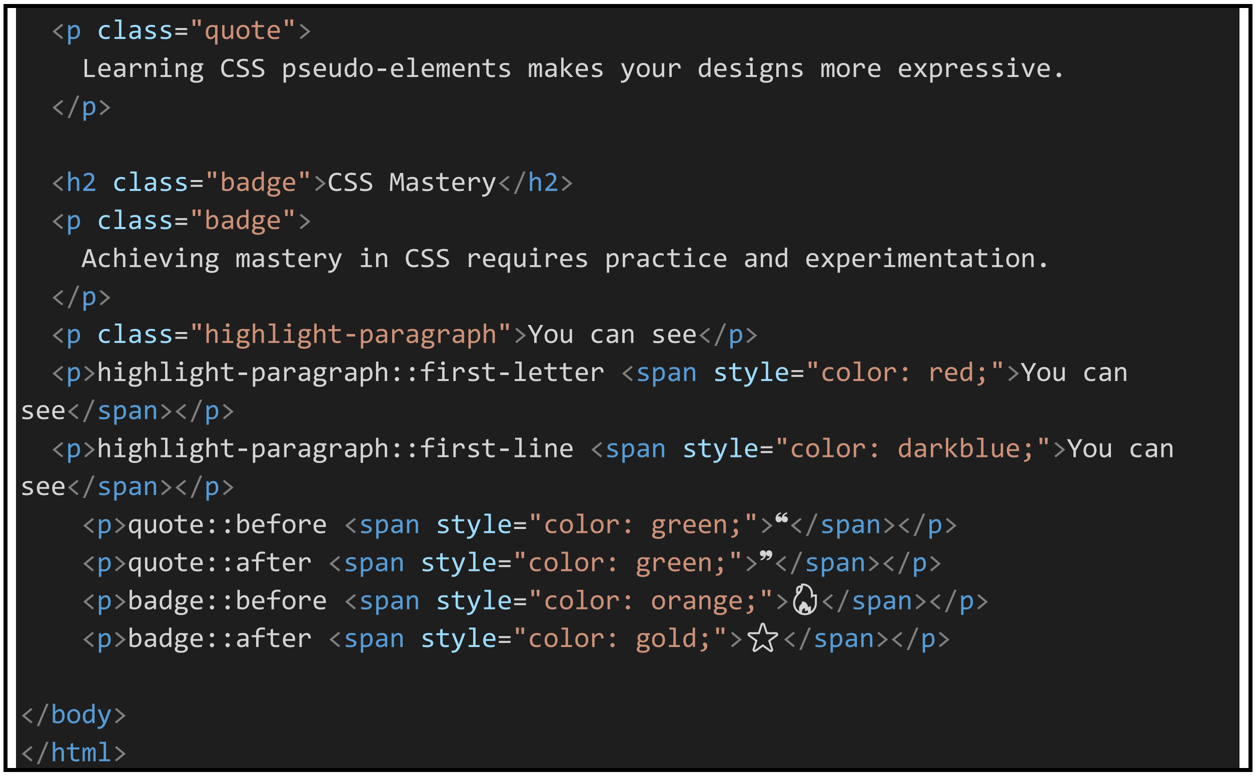
2. CSS PSEUDO-ELEMENTS:



```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Pseudo-elements Example</title>
  <style>
    body {
      font-family: Georgia, serif;
      padding: 40px;
      background-color: #f0f0f0;
      color: #333;
      line-height: 1.6;
    }

    h1 {
      text-align: center;
      color: #2c3e50;
    }

    /* Pseudo-element ::first-letter */
    .highlight-paragraph::first-letter {
      font-size: 2em;
      color: red;
```

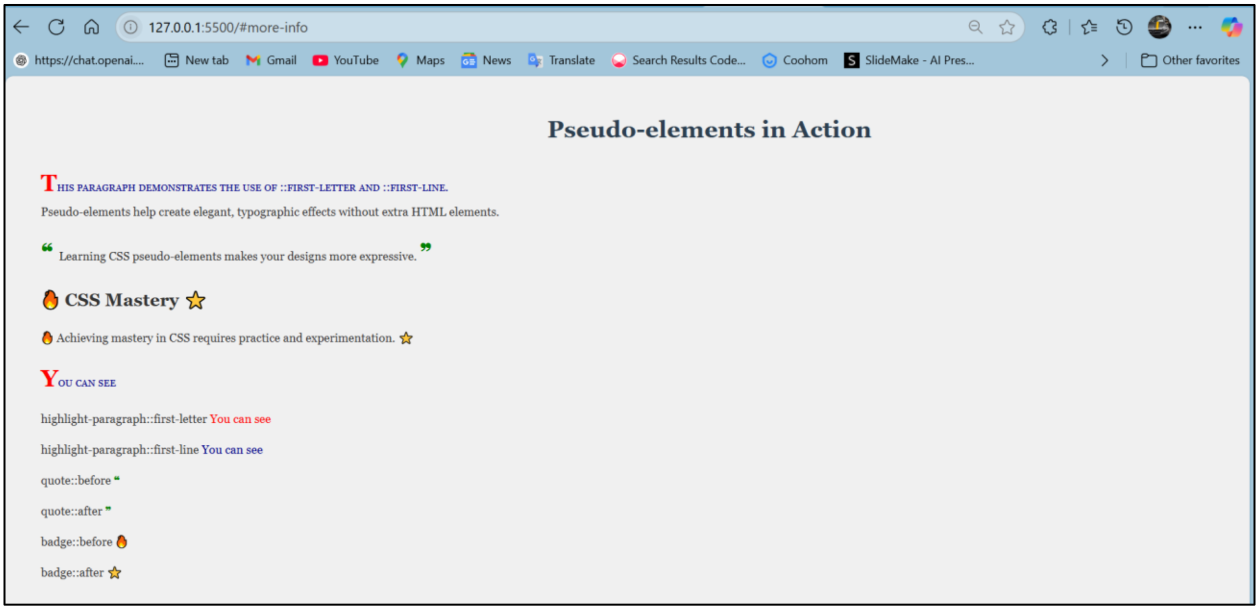


```
      font-weight: bold;
    }

    /* Pseudo-element ::first-line */
    .highlight-paragraph::first-line {
      color: darkblue;
      font-variant: small-caps;
    }

    /* Pseudo-element ::before */
    .quote::before {
      content: "❝ ";
      font-size: 2em;
      color: green;
    }

    /* Pseudo-element ::after */
    .quote::after {
      content: " ❞";
      font-size: 2em;
      color: green;
    }

    /* Pseudo-element with decorative content */
    .badge::before {
      content: "🔥 ";
      color: orange;
    }

    .badge::after {
      content: " ⭐";
      color: gold;
    }
  </style>
</head>
<body>

  <h1>Pseudo-elements in Action</h1>

  <p class="highlight-paragraph">
    This paragraph demonstrates the use of ::first-letter and ::first-line.
    <br>Pseudo-elements help create elegant, typographic effects without extra
HTML elements.
  </p>
```



```
  <p class="quote">
    Learning CSS pseudo-elements makes your designs more expressive.
  </p>

  <h2 class="badge">CSS Mastery</h2>
  <p class="badge">
    Achieving mastery in CSS requires practice and experimentation.
  </p>
  <p class="highlight-paragraph">You can see</p>
  <p>highlight-paragraph::first-letter <span style="color: red;">You can
see</span></p>
  <p>highlight-paragraph::first-line <span style="color: darkblue;">You can
see</span></p>
    <p>quote::before <span style="color: green;">“</span></p>
    <p>quote::after <span style="color: green;">”</span></p>
    <p>badge::before <span style="color: orange;">🔥</span></p>
    <p>badge::after <span style="color: gold;">☆</span></p>

</body>
</html>
```

Output:



Pseudo-elements in Action

THIS PARAGRAPH DEMONSTRATES THE USE OF ::FIRST-LETTER AND ::FIRST-LINE.

Pseudo-elements help create elegant, typographic effects without extra HTML elements.

“ Learning CSS pseudo-elements makes your designs more expressive. ”

🔥 CSS Mastery ⭐

🔥 Achieving mastery in CSS requires practice and experimentation. ⭐

YOU CAN SEE

highlight-paragraph::first-letter You can see

highlight-paragraph::first-line You can see

quote::before “

quote::after ”

badge::before 🔥

badge::after ⭐

Pseudo-element	**What It Does**
::first-letter	Styles the **first letter** of a paragraph
::first-line	Styles the **first line** of a paragraph
::before	Inserts content **before** the element
::after	Inserts content **after** the element

Conclusion:

CSS pseudo-classes and pseudo-elements are essential tools for every front-end developer. They allow us to create rich, responsive designs and interactive experiences HTML elements. Through this hands-on session, I deepened my understanding of **dynamic styling**, **state-based design**, and **typographic enhancements**.