Task3-CSS Selectors (ChatGPT)  
CSS selectors are patterns used to select and style specific HTML elements. They allow developers to apply styles to elements based on their tag name, class, ID, attributes, and more. Selectors define the target elements in the HTML document that the associated CSS rules will apply to.

**Common Types of CSS Selectors:**

1. **Element Selector**: Targets HTML elements by their tag name (e.g., div, p, h1).
2. **Class Selector**: Targets elements with a specific class attribute. Classes can be reused across multiple elements (e.g., .class-name).
3. **ID Selector**: Targets a unique element with a specific ID (e.g., #id-name). IDs must be unique within a document.
4. **Attribute Selector**: Targets elements with a specific attribute or attribute value (e.g., [type="text"]).
5. **Pseudo-class Selector**: Targets elements based on their state or position (e.g., :hover, :nth-child(2)).

**Example HTML Document with Various Selectors**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>CSS Selectors Example</title>

    <style>

        /\* Element Selector - targets all <p> elements \*/

        p {

            color: blue;

            font-size: 18px;

        }

        /\* Class Selector - targets elements with class 'highlight' \*/

        .highlight {

            background-color: yellow;

            font-weight: bold;

        }

        /\* ID Selector - targets the element with ID 'unique-header' \*/

        #unique-header {

            color: green;

            font-size: 24px;

            text-align: center;

        }

        /\* Attribute Selector - targets input elements of type 'text' \*/

        input[type="text"] {

            border: 2px solid red;

            padding: 5px;

        }

        /\* Pseudo-class Selector - targets links when hovered \*/

        a:hover {

            color: red;

            text-decoration: underline;

        }

        /\* Pseudo-class Selector - targets the second <li> element \*/

        li:nth-child(2) {

            color: purple;

        }

    </style>

</head>

<body>

    <!-- Element Selector -->

    <p>This paragraph is styled using an element selector.</p>

    <!-- Class Selector -->

    <p class="highlight">This paragraph uses the 'highlight' class.</p>

    <!-- ID Selector -->

    <h1 id="unique-header">This heading is styled using an ID selector.</h1>

    <!-- Attribute Selector -->

    <form>

        <label for="name">Name:</label>

        <input type="text" id="name" name="name" placeholder="Enter your name">

    </form>

    <!-- Pseudo-class Selector (Hover) -->

    <a href="#">Hover over this link to see the effect.</a>

    <!-- Pseudo-class Selector (nth-child) -->

    <ul>

        <li>First item</li>

        <li>Second item - This is the second list item</li>

        <li>Third item</li>

    </ul>

</body>

</html>

**Explanation of Selectors and Targeting**

1. **Element Selector (p)**:  
   Targets all <p> elements in the document and styles them with blue text and a font size of 18px.
2. **Class Selector (.highlight)**:  
   Targets any element with the class="highlight" and applies a yellow background and bold font. It can be reused on multiple elements.
3. **ID Selector (#unique-header)**:  
   Targets the element with id="unique-header" (in this case, the <h1>). IDs should be unique within a page, so this selector is used to apply styles only to this particular element.
4. **Attribute Selector (input[type="text"])**:  
   Targets <input> elements with the attribute type="text", styling text input fields with a red border and padding.
5. **Pseudo-class Selector (a:hover)**:  
   Targets <a> elements when the user hovers over them, changing the link color to red and underlining the text when the cursor is over the link.
6. **Pseudo-class Selector (li:nth-child(2))**:  
   Targets the second <li> element inside a list and changes its text color to purple, applying styles based on the element's position within its parent.