**Batch: C2 Roll No.:16010122257**

**Experiment No. 02**

|  |
| --- |
| **TITLE:** **Develop and demonstrate the usage of inline, internal and external style sheet using CSS** |

**AIM:** To demonstrate usage of CSS

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Expected Outcome of Experiment:** Use CSS to prepare the layout of web pages.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Books/ Journals/ Websites referred:**

1. https://www.w3schools.com/w3css/defaulT.asp.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Importance of CSS in designing of a website is to be explained. Explain various ways to use CSS. Also explain how to change background colour of page, adding and editing border types, adding navigation bars, usage of various types of 2D and 3D transformation.

**Description of the CSS style code with its effect at output**:

< Add code and screen shots of the pages here>

**CODE:**

**TIMETABLE:**

table, th, td {

        top: 100px;

        bottom:100px;

        border: 1px solid black;

        text-align: center;

        box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);

        background-color: #f2f2f2;

        font-size: 0.8em;

    }

    th, td {

        padding: 10px;

        text-align:center;

    }

    .center-table {

        display: flex;

        justify-content: center;

        align-items: center;

        height: 100vh;

        width: 50%;

        overflow: auto;

    }

# SIGNUP:

form {

            width: 300px;

            margin: 0 auto;

        }

        label {

            display: block;

            margin-top: 20px;

        }

        input[type="text"], input[type="email"], input[type="tel"], input[type="password"] {

            width: 100%;

            padding: 10px;

            box-sizing: border-box;

        }

        input[type="submit"] {

            margin-top: 20px;

        }

**LOGIN:**

b {

    text-align:center;

    padding:700px;

}

table {

    border: 1px solid black;

    text-align: center;

    align-items: center;

    justify-content: center;

    padding:100px;

    width:100px;

}

button {

 background-color: #4CAF50;

 color:white;

}

**HOMEPAGE:**

nav {

                background-color: #a41c1c;

                padding: 10px;

                text-align: center;

            }

            b {

                background-color:#a41c1c;

                padding:10px;

            }

            nav a {

                text-decoration: none;

                color: #0b0101;

                margin: 0 10px;

                padding: 5px 10px;

                border-radius: 5px;

                cursor: pointer;

            }

            nav a:hover {

                background-color: #540c0c;

                color: white;

            }

.center {

                display: block;

                margin-left: auto;

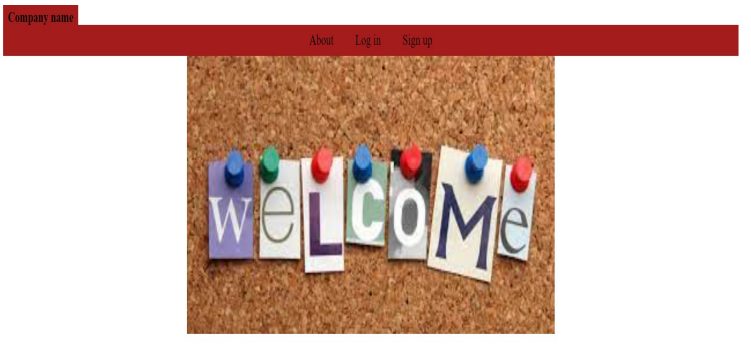
                margin-right: auto;

                width: 50%;

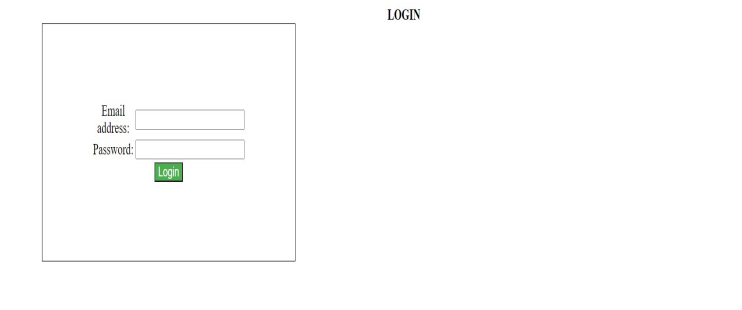
                }

**OUTPUT SCREENSHOTS:**

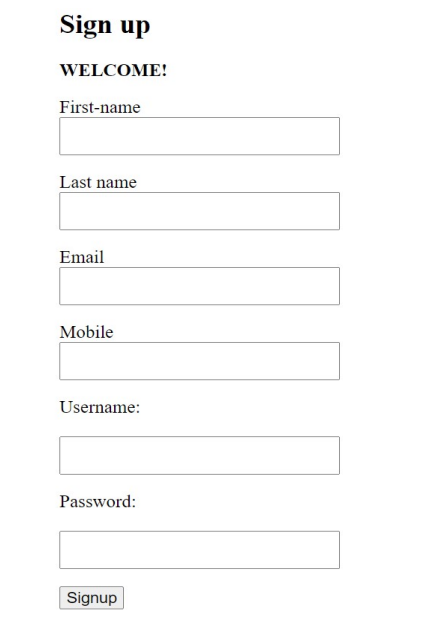
**HOMEPAGE:**



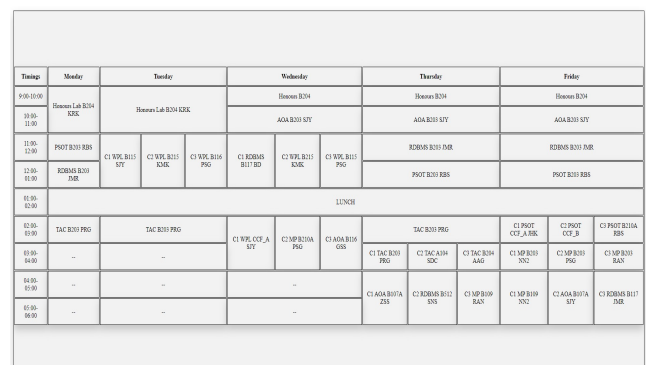
**LOGIN PAGE:**



**SIGN UP/REGISTRATION PAGE:**



**TIMETABLE:**



**Post Lab Objective with Ans (Min 5):**

* **What is the Box model in CSS? …**

In CSS, the term "box model" is used when talking about design and layout.

The CSS box model is essentially a box that wraps around every HTML element. It consists of: content, padding, borders and margins.

CONTENT:

The content area, enclosed by the content edge, encompasses the actual substance of the element, including text, images, or video players. Its measurements are determined by the content width (or content-box width) and content height (or content-box height). Frequently, it features a background color or background image.

When the box-sizing property is configured as content-box (which is the default), and the element is a block element, the size of the content area can be precisely specified using properties like width, min-width, max-width, height, min-height, and max-height.

PADDING:

The padding area, confined by the padding edge, expands the content area to encompass the padding of the element. Its size is defined by the padding-box width and the padding-box height.

The extent of the padding is influenced by properties such as padding-top, padding-right, padding-bottom, padding-left, as well as shorthand padding properties.

BORDER-AREA:

The border area, enclosed by the border edge, enlarges the padding area to encompass the element's borders. Its dimensions are denoted by the border-box width and the border-box height.

The thickness of the borders is established by the border-width property and shorthand border properties. In case the box-sizing property is configured as border-box, the size of the border area can be precisely specified using properties like width, min-width, max-width, height, min-height, and max-height.

When a background (background-color or background-image) is set on a box, it typically extends to the outer border edge, essentially going beneath the border in the z-order. This default behavior can be modified with the background-clip CSS property.

MARGINS:

The margin area, confined by the margin edge, expands the border area to include an empty space utilized to create a separation between the element and its neighboring elements. Its dimensions are specified by the margin box width and the margin box height.

The extent of the margin area is influenced by properties like margin-top, margin-right, margin-bottom, margin-left, and shorthand margin properties. In situations where margin collapsing occurs, defining the exact margin area becomes challenging as margins are shared between adjacent boxes.

It's important to note that for non-replaced inline elements, the space occupied (contribution to the height of the line) is determined by the line-height property, even though borders and padding are still visible around the content.

* **What are the advantages of using CSS? …**

**Consistency and Global Styling:**

CSS ensures uniform visual presentation across a website or multiple pages by centralizing style rules in a single external file. This approach minimizes redundancy and facilitates global styling updates.

**Easier Maintenance and Efficient Updates:**

CSS's separation of content and style simplifies website maintenance by allowing quick updates to styling through modifications in a central CSS file. This streamlines the process and enhances efficiency without the need to edit HTML elements individually.

**Time-Saving and Faster Loading:**

External CSS files enable web browsers to cache styling information, resulting in faster page loading times. This optimization enhances user experience, particularly for returning visitors, and saves time by reducing loading delays.

**Enhanced Design and Responsive Capabilities:**

CSS provides flexibility in design and supports responsive web design, ensuring adaptability across various screen sizes and devices. This versatility is essential for maintaining an appealing and functional website across different platforms.

**Accessibility and Print-Friendliness:**

CSS aids in creating accessible web content and facilitating the generation of printer-friendly versions of web pages. This inclusivity enhances user experience, particularly for those with accessibility needs, and allows for easy printing of web content.

**Platform Independence and Compatibility:**

CSS ensures consistent design across different platforms and devices, enhancing accessibility and user experience for a broader audience.

**Cascade, Inheritance, and Class/ID Usage:**

CSS offers flexibility in styling through cascading styles, inheritance, and precise class/ID usage, providing efficient control over layout and presentation.

**Advanced CSS Techniques:**

Advanced CSS techniques, including sprites, animations, and effects, enable the creation of dynamic and visually engaging web content without relying on complex scripting languages. This enhances user engagement and interaction on websites.

* **What are the limitations of CSS?**

**Cross-Browser Challenges and Browser Compatibility:**

Inconsistencies in the appearance of CSS across various web browsers pose a significant drawback, necessitating developers to create specialized CSS code for each browser or employ vendor prefixes to ensure consistent rendering.

**Learning Complexity:**

A notable challenge lies in the learning curve associated with CSS, particularly for beginners. Mastering CSS involves comprehending concepts such as selectors, properties, values, and the box model, which can be overwhelming for individuals new to web development.

**Security Considerations:**

Unlike certain web technologies, CSS lacks inherent security features and can become a potential security vulnerability if mishandled. Developers must exercise caution to prevent threats like Cross-Site Scripting (XSS) by thoroughly cleaning and validating user-generated CSS input.

**Challenges with Complex Layouts and Limited Control:**

CSS exhibits limitations in managing intricate web layouts, making tasks like achieving equal-height columns challenging without resorting to workarounds or exploring alternative technologies like Flexbox or Grid.

**File Size and Performance Impact:**

Cumbersome or poorly optimized CSS files can result in sluggish loading times for web pages. To maintain optimal website performance, it is crucial to keep CSS files minimal and optimized.

**Over-Specificity and Style Overrides:**

The functioning of CSS can sometimes lead to unexpected conflicts in styling. This occurs because CSS is "cascading," allowing styles to accumulate and occasionally override each other in unforeseen ways. Managing these conflicts poses a challenge.

**Maintenance Difficulties:**

As websites expand and undergo changes, the task of maintaining and reorganizing CSS can become intricate and time-intensive. Adhering to best practices and employing naming conventions becomes essential to ensure the manageability of styles.

* **What are the different types of Selectors in CSS?**

CSS selectors are used to "find" (or select) the HTML elements you want to style.

We can divide CSS selectors into five categories:

* Simple selectors (select elements based on name, id, class)
* Combinator selectors (select elements based on a specific relationship between them)
* Pseudo-class selectors (select elements based on a certain state)
* Pseudo-elements selectors (select and style a part of an element)
* Attribute selectors (select elements based on an attribute or attribute value)

|  |  |  |
| --- | --- | --- |
| **Selector** | **Example** | **Example description** |
| #id | #firstname | This selects the element with id as “firstname” |
| .class | .intro | Selects all elements with class as “intro” |
| element.class | p.intro | Selects only paragraph(<p>) elements with class as “intro” |
| \* | \* | Selects all elements |
| element | p | Selects all paragraph elements (<p>) |
| element,element,.. | div, p | Selects all <div> elements and all <p> elements |