

K. J. Somaiya College of Engineering, Mumbai-77
(A Constituent College of Somaiya Vidyavihar University)

Batch: A3 Roll No.: 16010121045

Experiment No. 02

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

AIM: To demonstrate usage of CSS

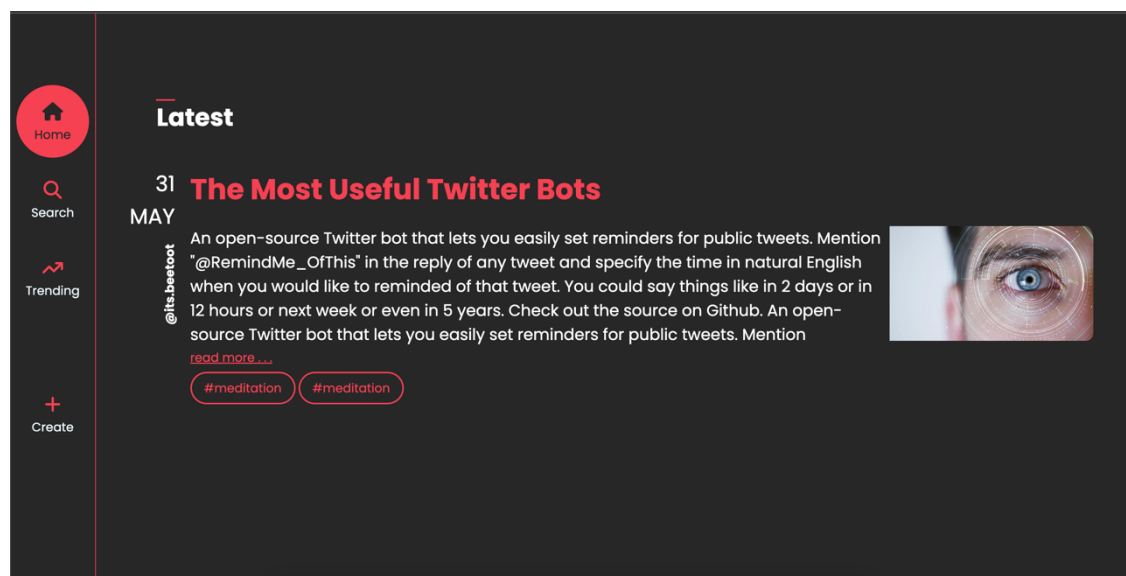
Expected Outcome of Experiment: Use CSS to prepare the layout of web pages.

Books/ Journals/ Websites referred:

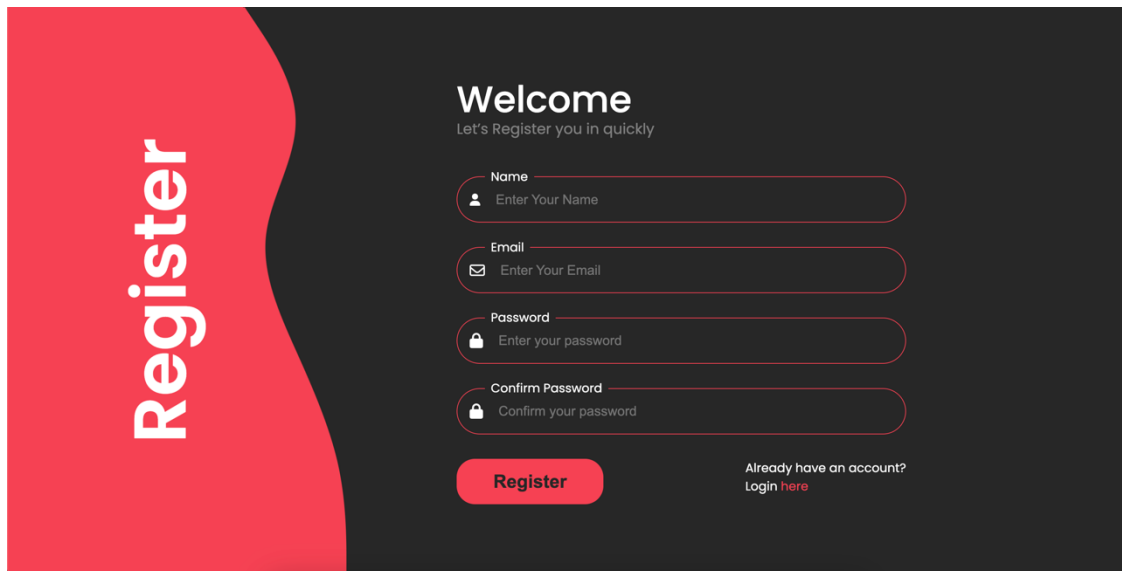
1. .

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:



K. J. Somaiya College of Engineering, Mumbai-77
(A Constituent College of Somaiya Vidyavihar University)



The image shows a registration form with a dark background and a red accent. On the left, a large red shape contains the word "Register" in white. On the right, the text "Welcome" is followed by "Let's Register you in quickly". Below this are four input fields: "Name" (with a person icon), "Email" (with an envelope icon), "Password" (with a lock icon), and "Confirm Password" (with a lock icon). Each field has a placeholder text: "Enter Your Name", "Enter Your Email", "Enter your password", and "Confirm your password". At the bottom, there is a red "Register" button and a link "Already have an account? Login here".

```
:root {
  --bg: #272727;
  --accent: #F64253
}

/* Navbar */

.navbar {
  display: flex;
  position: fixed;
  height: 100vh;
  z-index: 100;
}

nav {
  display: flex;
  position: relative;
  align-items: center;
  text-align: center;
  overflow: none;
  order: 1;
  border-right: 1px solid var(--accent);
  width: 7rem;
}
```

K. J. Somaiya College of Engineering, Mumbai-77
(A Constituent College of Somaiya Vidyavihar University)

```
nav ul {
  display: flex;
  flex-direction: column;
  list-style-type: none;
  width: 100%;
  padding: 0 0.5rem;
}

.icons {
  display: flex;
  flex-direction: column;
  align-items: center;
  width: 100%;
}

.top-icons {
  justify-content: space-between;
  height: 40vh;
}

.bottom-icons {
  height: 35vh;
  justify-content: center;
}

.nav-buttons {
  aspect-ratio: 1/1;
  width: 100%;
}

.nav-buttons i {
  padding: 1rem;
}

.nav-buttons a {
  display: flex;
  flex-direction: column;
  justify-content: center;
  width: 100%;
  height: 100%;
}
```

K. J. Somaiya College of Engineering, Mumbai-77
(A Constituent College of Somaiya Vidyavihar University)

```
text-decoration: none;
color: var(--accent);
transition: all 0.4s linear;
}

.nav-buttons a:hover {
  background-color: var(--accent);
  border-radius: 100%;
  color: var(--bg);
}

.nav-buttons p {
  display: inline-block;
  margin-top: 0.1rem;
  font-size: 1rem;
  color: aliceblue;
  transition: all 0.4s linear;
}

.nav-buttons a:hover p {
  color: var(--bg);
}

.selected {
  background-color: var(--accent);
  border-radius: 100%;
}

.selected i {
  color: var(--bg);
}

.selected p {
  color: var(--bg);
}

@media only screen and (max-width : 720px) {

  nav {
    position: fixed;
```

K. J. Somaiya College of Engineering, Mumbai-77
(A Constituent College of Somaiya Vidyavihar University)

```
z-index: 10;
bottom: 0;
left: 50%;
transform: translate(-50%, -20%);
background-color: var(--bg);
width: 95%;
order: 3;
border: 1px solid var(--accent);
border-radius: 2rem;
}

nav ul {
  flex-direction: row;
  justify-content: center;
  width: 100%;
  padding: 0;
}

.nav-buttons {
  padding: 0.5rem;
  margin: auto 0.5rem;
}

.nav-buttons p {
  display: none;
}

.icons {
  height: 5rem;
  width: auto;
}

.top-icons,
.bottom-icons {
  flex-direction: row;
}
}
```

K. J. Somaiya College of Engineering, Mumbai-77
(A Constituent College of Somaiya Vidyavihar University)

Post Lab Objective with Ans (Min 5):

● **What is the Box model in CSS? ...**

The Box Model in CSS is a concept that defines how elements are rendered on a web page by breaking them down into several nested boxes, each with its own set of properties. Each box represents a rectangular area on the web page that can contain content, padding, borders, and margins.

The Box Model consists of four components:

- **Content:** This is the actual content of the element, such as text, images, or videos.
- **Padding:** This is the space between the content and the border of the element.
- **Border:** This is the line that surrounds the padding and content of the element.
- **Margin:** This is the space between the border of the element and other elements on the page.

By default, the width and height of an element are calculated based on its content, but you can adjust the size of each of the components using CSS properties. For example, you can add padding to an element by using the padding property or add a border by using the border property. You can also adjust the margin around an element by using the margin property. Understanding the Box Model is essential for designing and laying out web pages using CSS.

● **What are the advantages of using CSS? ...**

There are many advantages to using CSS (Cascading Style Sheets) for web design and development. Here are some of the key benefits:

- **Separation of presentation and content:** CSS allows for a clear separation of presentation (layout and styling) and content (text, images, and other media), which makes it easier to maintain and update the design of a website without affecting the content.
- **Consistent styling:** CSS allows for the creation of reusable style rules that can be applied consistently across multiple pages, which helps to ensure a consistent and professional appearance for a website.
- **Greater flexibility and control:** CSS provides a wide range of styling options and allows for greater control over the layout and appearance of a website, which makes it easier to create complex designs and layouts.
- **Accessibility:** CSS allows for the creation of accessible designs that can be easily read and understood by people with disabilities, such as those using screen readers or other assistive technologies.

K. J. Somaiya College of Engineering, Mumbai-77
(A Constituent College of Somaiya Vidyavihar University)

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

Here are some of the key limitations of CSS:

- Limited layout control: CSS has limitations in terms of controlling layout, especially for complex layouts or layouts that require precise positioning of elements. This can make it challenging to create designs that require highly customized layouts.
- Browser compatibility issues: CSS may not be supported uniformly across all web browsers, which can lead to inconsistencies in how a website is displayed in different browsers. This can require additional effort to ensure cross-browser compatibility.
- Lack of interactivity: CSS is primarily a styling language and does not provide much support for interactivity or dynamic behavior, such as animations or user interactions.
- Large file sizes: CSS files can become large and complex, especially for larger websites, which can impact page loading times and overall website performance.
- Limited support for older browsers: While modern browsers support the latest versions of CSS, older browsers may not support some of the newer features, which can limit the design options for websites that need to support older browsers.

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

There are several types of selectors in CSS, including:

- Type selectors: These selectors target specific types of HTML elements, such as h1, p, or div.
- Class selectors: These selectors target HTML elements with a specific class attribute, using the syntax .class name.
- ID selectors: These selectors target HTML elements with a specific ID attribute, using the syntax #elementid.
- Attribute selectors: These selectors target HTML elements with a specific attribute or attribute value, using the syntax [attribute=value].
- Descendant selectors: These selectors target HTML elements that are descendants of other elements, using the syntax parent element child element.
- Child selectors: These selectors target HTML elements that are direct children of other elements, using the syntax parent element > child element.
- Adjacent sibling selectors: These selectors target HTML elements that come immediately after another element, using the syntax element + sibling.
- General sibling selectors: These selectors target HTML elements that come after another element, using the syntax element ~ sibling.
- Pseudo-classes: These selectors target HTML elements based on their state or position, such as :hover, :active, or :nth-child().