**Module (CSS and CSS 3) -2**

**Name:- Vijay Ratnotar Mo.No.9429099686**

1. **What are the benefits of using CSS?**

**Answer:**

Cascading Style Sheets, or CSS, provides many advantages. It offers for greater flexibility and control over the design and layout of a website, separates the presentation of a website from its content, makes it simpler to manage and update the styling and layout of a website, and can speed up website loading times. For instance, by utilising CSS to style a website, you may modify only one CSS file to alter the font size, colour, and other visual components on all of the website's pages. Similar to responsive design rules, CSS makes it simple to establish hover effects and animations on elements for more interactivity and aesthetic appeal. It also makes it simple to construct responsive design rules, where items on a page alter their layout based on screen size.

1. **What are the disadvantages of CSS?**

**Answer:**

As I have stated, major drawbacks of CSS include its steep learning curve, the expense and effort required to adopt it for larger websites, and the uneven rendering across different browsers. A developer might invest a lot of time and money into building a stunning, intricate website with sophisticated CSS features, only to discover that certain people cannot view it properly because their browser does not support those features or renders them differently.

1. **What is the difference between CSS2 and CSS3?**

**Answer:**

The most recent version of CSS, CSS3, has mostly replaced the older version CSS2, which was introduced in 1998. Numerous improvements and new capabilities are included in CSS3, including new layout modules, enhanced selectors, animations, and transitions. For instance, CSS3 made it considerably simpler to build aesthetically beautiful designs by introducing the "border-radius" property, which enables developers to create rounded corners on elements without the usage of images or tricky CSS hacks. The "flexbox" and "grid" modules provided by CSS3 also make it simpler to develop flexible, responsive layouts without the need for external libraries or frameworks.

1. **Name a few CSS style components**

**Answer:**

Sure! The following are some typical CSS style elements:

1. Selectors: These are used to apply styles to particular HTML elements. For instance, a navigation bar might have styles applied to it using the selector ".navbar".

2. Properties: These are the actual styling options you have for the elements you've chosen. The "colour" property, for instance, can be used to alter an element's text colour.

3. Values: The particular values that can be applied to a property are listed here. To make the text red, for instance, the "colour" property and the value "red" may be used.

An example of CSS code using these components would be something like this:

**Example:**

.navbar {

background-color: blue;

color: white;

font-weight: bold;

}

This code would select all elements with the class "navbar" and apply a blue background color, white text color, and bold font weight to them.

**Output:**

1. **What do you understand by CSS opacity?**

**Answer:**

CSS opacity is a property that controls the transparency of an element. It allows you to make an element translucent or transparent. The opacity value ranges from 0.0 (completely transparent) to 1.0 (completely opaque). Here's an example of CSS code that sets the opacity of an element to 0.5:

**Example:**

div {

opacity: 0.5;

}

This would make the entire div element semi-transparent, allowing any content behind it to show through.

**Output:**

1. **How can the background color of an element be changed?**

**Answer:**

To change the background color of an HTML element, you can use CSS (Cascading Style Sheets). CSS allows you to control the appearance of HTML elements, including their background color. Here's an example of how you can change the background color of an element:

**Example:**

HTML:

<div id="myElement">Hello, world!</div>

CSS:

#myElement {

background-color: blue;

}

In this example, we have a `<div>` element with the ID "myElement" that contains the text "Hello, world!". To change its background color, we use CSS to target the element by its ID (`#myElement`) and set the `background-color` property to the desired color. In this case, the background color is set to blue.

You can use different ways to specify colors in CSS. You can use color names (e.g., "blue", "red"), hexadecimal values (e.g., "#FF0000" for red), RGB values (e.g., "rgb(255, 0, 0)" for red), or HSL values (e.g., "hsl(0, 100%, 50%)" for red). CSS provides various color options to choose from.

By adjusting the `background-color` property and specifying the desired color value, you can change the background color of any HTML element on your webpage.

1. **How can image repetition of the backup be controlled?**

**Answer:**

Deduplication techniques can be used to control image repetition in backups. Deduplication is the act of finding and removing duplicate data, in this case, duplicate photos, in order to save storage space and improve backup efficiency. There are several ways for achieving deduplication, including file-level and block-level deduplication.

File-level deduplication identifies duplicates by comparing complete picture files. If a file already exists in the backup, a reference to the existing file is established rather than saving a duplicate. Assume you have a backup of your photo collection and mistakenly add the identical image again. The backup system recognises that the image already exists and keeps simply a reference to the original image, saving storage space.

In contrast, block-level deduplication divides picture files into smaller chunks and examines those blocks for duplicates. This more detailed approach can detect duplicate blocks in several photos or files. Consider two almost identical images, with just a little area of the image differing. Only the unique blocks are kept using block-level deduplication, while the standard blocks are accessed from the existing backup.

By finding and removing duplicate data, both file-level and block-level deduplication algorithms help limit image recurrence in backups. These technologies dramatically reduce storage needs and optimise backup procedures, resulting in more efficient and cost-effective backup processes. It's vital to note that many types of deduplication algorithms exist, such as file-level deduplication, block-level deduplication, and variable-length deduplication, each with its own approach and benefits. Depending on the backup programme or system utilised, the particular implementation may differ.

1. **What is the use of the background-position property?**

**Answer:**

In CSS, the 'background-position' property specifies the initial position of a background image within its container. It specifies where the image will be positioned about the container's bounds.

The 'background-position' property can be expressed in two ways: longhand and shorthand. The shorthand notation specifies the horizontal and vertical locations of the background image in a more simple manner.

Here's an example of how to use the `background-position` property in shorthand notation:

css

div {

background-image: url('image.jpg');

background-position: right bottom;

}

In this example, the `background-image` property specifies the image file to be used as the background. The `background-position` property positions the background image at the bottom right corner of the container. The horizontal position is specified by the keyword "right," and the vertical position is specified by the keyword "bottom."

You can also use other keywords or length values to define the position. Here are some examples: css

div {

background-image: url('image.jpg');

background-position: center top;

}

In this case, the background image will be positioned at the center horizontally and at the top vertically within the container.

css

div {

background-image: url('image.jpg');

background-position: 50% 50%;

}

Here, the background image will be centered both horizontally and vertically within the container using percentage values.

css

div {

background-image: url('image.jpg');

background-position: 10px 20px;

}

In this example, the background image will be positioned 10 pixels from the left edge and 20 pixels from the top edge of the container.

By using the `background-position` property, you can precisely control the placement of the background image within its container, allowing you to achieve the desired visual effect for your webpage or application.

1. **Which property controls the image scroll in the background?**

**Answer:**

The property that controls image scrolling in the background is called "background-attachment." It determines whether the background image scrolls with the content or remains fixed in place.

There are three possible values for the background-attachment property:

1. "scroll" (default): This value allows the background image to scroll along with the content. As you scroll through the webpage, the image moves accordingly. Here's an example of CSS code using this value:

css

body {

background-image: url('background-image.jpg');

background-attachment: scroll;

}

2. "fixed": This value keeps the background image fixed in place while the content scrolls. The image remains stationary, creating an effect where it appears as if the content is scrolling over the image. Here's an example:

css

body {

background-image: url('background-image.jpg');

background-attachment: fixed;

}

3. "local": This value allows the background image to scroll with its container element. If a particular element, such as a div, has a background image, the image will scroll within that element while the content scrolls independently. Here's an example:

css

.container {

background-image: url('background-image.jpg');

background-attachment: local;

}

By using the appropriate value for the `background-attachment` property, you can control how the background image behaves when scrolling through a webpage.

1. **Why should background and color be used as separate properties?**

**Answer:**

Background and colour are two distinct CSS attributes that serve various functions when it comes to decorating web page components.

The "background" attribute is used to specify an element's background style. Background-color, background-image, background-repeat, and other parameters are included. It enables you to change an element's visual backdrop, such as a solid colour or an image.

Here's an example:

css

div {

background-color: #F5F5F5;

background-image: url('image.jpg');

background-repeat: repeat-x;

}

In this example, the background-color property sets the background color of the div element to a light gray (#F5F5F5), while the background-image property specifies an image ('image.jpg') to be repeated horizontally across the element. The background-repeat property determines how the image is repeated.

On the other hand, the "color" property is used to define the text color of an element. It allows you to specify the color of the text within an element.

css

p {

color: blue;

}

In this example, the color property sets the text color of all paragraphs to blue.

Separating background and color properties allows you to apply different styles to the background and text independently. For example, you may want to have a white background with black text, or a colored background with contrasting text color for better readability.

By keeping these properties separate, you have greater flexibility in designing the visual appearance of your web page.

1. **How to center block elements using CSS1?**

**Answer:**

In CSS1, there is no direct method to center block elements horizontally. However, you can achieve center alignment by using a combination of CSS properties. Here's an example of how you can center a block element horizontally using CSS1:

HTML:

<div class="centered">

<!-- Your content here -->

</div>

CSS:

.centered {

margin-left: auto;

margin-right: auto;

width: 50%; /\* Adjust this value as needed \*/

}

1. The material you wish to centre is contained within the "div>" element with the class "centred."

2. The left and right margins of the element are set to "auto" via the'margin-left: auto' and'margin-right: auto' attributes. The element is now horizontally centred within its parent container as a result.

3. A percentage value, such as 50%, is assigned to the 'width' attribute. According to your needs, change this number to regulate the width of the centred element.

The block element will be centred horizontally within its parent container if the left and right margins are set to "auto" and a percentage width is specified.

Please note that CSS1 is a very old version of CSS, and modern web development practices use CSS3 and later versions, which provide more efficient and flexible methods for centering elements.

1. **How to maintain the CSS specifications?**

**Answer:**

When it comes to maintaining CSS specifications, it's essential to keep them concise and straightforward. Here are a few tips for achieving that:

1. Use a naming convention that is consistent: Choose a name convention that is simple to comprehend and implement. This will make it simpler to manage the codebase and help developers rapidly understand the intent behind each style rule. The BEM (Block Element Modifier) technique, for instance, uses classes with a structure of "block\_\_element--modifier."

2. Combine similar styles: Use related styles to organise your CSS rules. For instance, instead of dispersing the styles for a navigation menu throughout the code, group them under a single selector. This makes it easy to discover and alter styles later on and enhances readability.

3. Use shorthand properties: For frequently used style declarations, CSS offers shorthand properties. You may accomplish the same result by using shorthand notation rather than defining each property individually. You may accomplish the same effect by using'margin: 10px 20px;' as opposed to writing'margin-top: 10px; margin-right: 20px; margin-bottom: 10px; and margin-left: 20px;'.

4. Reduce redundant efforts: By efficiently utilising inheritance and cascade, unnecessary styles may be avoided. Apply the same styles to a common parent element if many elements have the same styles, then let the children inherit those styles. Use CSS's cascading feature to prevent repetition of styles whenever feasible.

Here's a simple example to illustrate these principles:

html

<!DOCTYPE html>

<html>

<head>

<style>

.menu {

background-color: #f1f1f1;

padding: 10px;

margin-bottom: 20px;

}

css

.menu\_\_item {

display: inline-block;

margin-right: 10px;

color: #333;

}

.menu\_\_item--active {

font-weight: bold;

}

</style>

</head>

<body>

<div class="menu">

<span class="menu\_\_item menu\_\_item--active">Home</span>

<span class="menu\_\_item">About</span>

<span class="menu\_\_item">Contact</span>

</div>

<div class="menu">

<span class="menu\_\_item">Products</span>

<span class="menu\_\_item">Services</span>

<span class="menu\_\_item">Support</span>

</div>

</body>

</html>

In this example, we have a simple menu with multiple items. The styles are organized using classes, making it easy to understand their purpose. Related styles are grouped together under the `.menu` selector, and the active menu item is differentiated using the `.menu\_\_item--active` modifier. Shorthand properties like `margin` are used to keep the code concise, and inheritance is leveraged by applying styles to the parent `.menu` element.

1. **What are the ways to integrate CSS as a web page?**

**Answer:**

To integrate CSS (Cascading Style Sheets) into a web page, you can use three different methods: inline styles, internal stylesheets, and external stylesheets. Here's a brief explanation of each method with an example:

1. Inline Styles:

Inline styles involve adding the CSS directly to individual HTML elements using the "style" attribute. This method is useful for applying unique styles to specific elements. Here's an example:

Html

<!DOCTYPE html>

<html>

<head>

<title>Inline Styles Example</title>

</head>

<body>

<h1 style="color: blue; font-size: 24px;">Hello, World!</h1>

<p style="background-color: yellow;">This is a paragraph with inline styles.</p>

</body>

</html>

In the above example, the `style` attribute is used to apply CSS properties directly to the `<h1>` and `<p>` elements.

2. Internal Stylesheets:

Internal stylesheets involve placing CSS code within the `<style>` tags in the `<head>` section of an HTML document. This method allows you to define styles for multiple elements within the same HTML file. Here's an example:

html

<!DOCTYPE html>

<html>

<head>

<title>Internal Stylesheet Example</title>

<style>

h1 {

color: blue;

font-size: 24px;

}

p {

background-color: yellow;

}

</style>

</head>

<body>

<h1>Hello, World!</h1>

<p>This is a paragraph with internal styles.</p>

</body>

</html>

In this example, the CSS styles for the `<h1>` and `<p>` elements are defined within the `<style>` tags in the `<head>` section.

3. External Stylesheets:

External stylesheets involve linking an external CSS file to the HTML document using the `<link>` tag. This method allows you to separate the CSS code into a separate file, which can be reused across multiple web pages. Here's an example:

HTML file (`index.html`):

<!DOCTYPE html>

<html>

<head>

<title>External Stylesheet Example</title>

<link rel="stylesheet" type="text/css" href="styles.css">

</head>

<body>

<h1>Hello, World!</h1>

<p>This is a paragraph with external styles.</p>

</body>

</html>

CSS file (`styles.css`):

h1 {

color: blue;

font-size: 24px;

}

p {

background-color: yellow;

}

In this example, the CSS styles are defined in a separate file named `styles.css`. The `<link>` tag in the HTML file specifies the path to the CSS file using the `href` attribute.

These are the basic ways to integrate CSS into a web page. Each method has its own use case, depending on the complexity and requirements of your project.

1. **What is embedded style sheets?**

**Answer:**

CSS rules that are directly included into an HTML document are referred to as embedded style sheets, sometimes known as embedded CSS (Cascading Style Sheets). A style sheet language called CSS is used to specify the visual presentation and formatting of an HTML or XML document.

In an HTML page with embedded style sheets, CSS rules are written inside the "style" tags found in the "head" section. The styles specified in the embedded style sheet only apply to the HTML document that contains them.

Here's an example of how embedded style sheets are used:

html

<!DOCTYPE html>

<html>

<head>

<title>Embedded Style Sheets Example</title>

<style>

h1 {

color: blue;

font-size: 24px;

}

p {

color: red;

font-size: 16px;

}

</style>

</head>

<body>

<h1>This is a heading</h1>

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

</body>

</html>

The embedded style sheet in the aforementioned example specifies two CSS rules: one for "h1" elements and one for "p" elements. The "p>" elements will have red text and a 16-pixel font size, whereas the "h1>" elements will have blue text and a 24-pixel font size.

When you wish to apply particular styles to a single HTML document without impacting other papers on your website, embedded style sheets are helpful. However, using an external style sheet, where CSS rules are set in a separate.css file and linked to various HTML documents, can be more effective if you have many HTML documents that call for the same styles.

1. **What are the external style sheets?**

**Answer:**

To specify the visual look and layout of web pages, external style sheets are files containing CSS (Cascading Style Sheets) code that are connected to HTML documents. They offer a means of separating a website's presentation from its organisation and content.

The CSS code is written in a separate file with a.css extension when utilising external style sheets. The link> element in the head> section of the HTML document is then used to refer to this file. The rel tag with the value "stylesheet" and the href attribute pointing to the CSS file's location are commonly present in links to external style sheets.

Here's an example of how an external style sheet is linked to an HTML document:

**Example:**

html

<!DOCTYPE html>

<html>

<head>

<link rel="stylesheet" href="styles.css">

</head>

<body>

<!-- HTML content goes here -->

</body>

</html>

In this example, the CSS code is contained in a file named "styles.css" located in the same directory as the HTML file. The browser reads the CSS file and applies the specified styles to the corresponding HTML elements.

Using outside style sheets has the following benefits:

**1. Separation of concerns:** It makes it simpler to maintain and update the aesthetic appearance of a website by allowing the separation of design (CSS) from the structure and content (HTML).

**2. Consistency:** To ensure uniform styling throughout the whole website, several HTML pages can refer to the same external style sheet.

**3. Caching:** Since the CSS file is already locally saved, subsequent page views may be quicker when external style sheets are cached by the browser.

By utilizing external style sheets, web developers can create and manage the visual aspects of a website efficiently and make global changes easily by modifying a single CSS file.

1. **What are the advantages and disadvantages of using external style sheets?**

**Answer:**

External Style Sheets are a method of organizing and managing the styles of a website or web application in a separate file. Here are the advantages and disadvantages of using external style sheets:

Advantages:

1. Centralized Styling: With external style sheets, you can define all the styles in a single file and link it to multiple HTML documents. This allows you to maintain consistent styling across your entire website. If you need to make changes to the styles, you can update the external style sheet, and the changes will be applied to all linked documents automatically.

Example:

Let's say you have a website with multiple web pages. Instead of writing the styles for each page individually, you can create a separate style sheet file, for example, "styles.css". In this file, you define all the styles for your website, such as the font, color, and layout. Then, you link this style sheet to each HTML document using the following code:

html

<link rel="stylesheet" type="text/css" href="styles.css">

Now, whenever you update "styles.css", all the linked HTML documents will reflect the changes.

2. Easy Maintenance: External style sheets simplify the process of maintaining and updating your website's styles. Since the styles are in a separate file, you can focus on editing the content of your HTML documents without worrying about the styling. This separation of concerns makes it easier to collaborate with other developers or designers who can work independently on the styles.

Disadvantages:

1. Additional HTTP Request: When a web page references an external style sheet, the browser needs to make an additional HTTP request to fetch the style sheet file. This can slightly increase the page loading time, especially if there are multiple style sheets linked. However, modern browsers mitigate this issue through caching mechanisms.

Example:

Consider a web page that includes an external style sheet. The HTML code would look like this:

html

<link rel="stylesheet" type="text/css" href="styles.css">

Here, the browser needs to retrieve the "styles.css" file from the server, which adds an extra HTTP request.

2. Dependency on File: In order for external style sheets to be used, the style sheet file must be accessible and correctly connected. The webpage could not have the desired style if the file is missing or wrongly linked, which could give it a broken or unformatted appearance.

For instance, the browser won't be able to locate and apply the styles if the connection to the external style sheet is broken or the file is mistakenly removed, resulting in an unstyled webpage.

Finally, external style sheets give your website centralised, simple-to-maintain styling that encourages cooperation and uniformity. However, they could incur a small amount of cost in the form of an extra HTTP request and are vulnerable to problems if the file or the linking.

1. **What is the meaning of the CSS selector?**

**Answer:**

In CSS (Cascading Style Sheets), a selector is used to target specific HTML elements on a web page and apply styles to them. There are various types of selectors available in CSS, including the shorthand selector. However, the term "shorthand selector" is not commonly used in CSS terminology. It is possible that you might be referring to a different concept or term.

To clarify, let's go over some commonly used CSS selectors and provide examples:

1. Element Selector:

This selector targets HTML elements based on their tag name. For example, to select all the paragraphs on a page and give them a specific style, you can use the element selector "p". Here's an example:

css

p {

color: blue;

font-size: 16px;

}

2. Class Selector:

This selector targets HTML elements based on their class attribute. It is denoted by a dot (.) followed by the class name. Multiple elements can share the same class. Here's an example:

html

<p class="highlight">This is a highlighted paragraph.</p>

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

css

.highlight {

background-color: yellow;

}

3. ID Selector:

This selector targets a specific HTML element based on its unique ID attribute. It is denoted by a hash (#) followed by the ID name. An ID should be unique within the HTML document. Here's an example:

html

<h1 id="title">Welcome to my website!</h1>

css

#title {

color: red;

}

4. Attribute Selector:

This selector targets HTML elements based on the presence or value of their attributes. For example, to select all input elements with a type of "text", you can use an attribute selector. Here's an example:

html

<input type="text" name="username">

<input type="password" name="password">

css

input[type="text"] {

border: 1px solid black;

}

These are just a few examples of commonly used CSS selectors. CSS provides a wide range of selectors to target elements based on various criteria, allowing you to apply styles and modify the appearance of your web page as desired.

1. **What are the media types allowed by CSS?**

**Answer:**

In CSS (Cascading Style Sheets), there are several media types that allow you to apply different styles to different devices or media. Media types define the output devices or environments in which a document is displayed. Here are some commonly used media types:

1. `all`: This is the default media type that applies to all devices.

Example:

css

@media all {

/\* CSS rules for all devices \*/

body {

font-size: 16px;

}

}

2. `screen`: This media type is used for computer screens, tablets, or smartphones.

Example:

css

@media screen {

/\* CSS rules for screens \*/

body {

background-color: #f2f2f2;

}

}

3. `print`: This media type is used when the document is intended for printing.

Example:

css

@media print {

/\* CSS rules for printing \*/

body {

font-family: Arial, sans-serif;

}

}

4. `speech`: This media type is used for screen readers or speech synthesizers.

Example:

css

@media speech {

/\* CSS rules for speech synthesizers \*/

h1 {

font-size: 24px;

}

}

These are just a few examples of media types supported by CSS. You can also use other media types like `projection` for projected presentations, `tv` for television-type devices, and `aural` for speech synthesizers with earcons. Media types allow you to tailor your styles for different output devices or environments, ensuring a better user experience across various platforms.

1. **What is the rule set?**

**Answer:**

A rule set refers to a collection of rules that are used to guide or govern a particular process or system. The rules in a rule set are typically organized in a hierarchical manner, with more general rules at the top and more specific rules at the bottom.

Here's a simple example of a rule set for a traffic signal:

- If the traffic light is red, then all vehicles must stop.

- If the traffic light is yellow, then vehicles should slow down and prepare to stop.

- If the traffic light is green, then vehicles may proceed through the intersection.

In this rule set, the rules are hierarchical, with the more general rule at the top ("If the traffic light is red") and the more specific rules at the bottom ("If the traffic light is green"). The rules are also conditional, meaning that they are triggered by a particular event or condition (i.e., the color of the traffic light).

Rule sets are used in many different applications, including business process management, decision-making systems, and artificial intelligence. They provide a way to formalize and automate decision-making processes, making them more efficient and consistent.

1. Create Layouts

**Answer:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

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

    <title>Create WebPage HTML& CSS Layouts</title>

    <style>

        \* {

            margin: 0;

            padding: 0;

            box-sizing: border-box;

        }

        #header {

            height: 100px;

            background-color: aqua;

            display: flex;

        }

        #logo {

            height: inherit;

            width: 20%;

            background-color: blue;

        }

        #link {

            height: inherit;

            width: 80%;

            background-color: crimson;

            display: flex;

        }

        #l1 {

            height: 50%;

            background-color: chartreuse;

            display: flex;

        }

        #l2 {

            height: 50%;

            background-color: rgba(88, 167, 10, 0.781);

            flex: 1;

        }

        #main {

            background-color: darkgreen;

            height: 500px;

        }

        #footer {

            height: 100px;

            background-color: black;

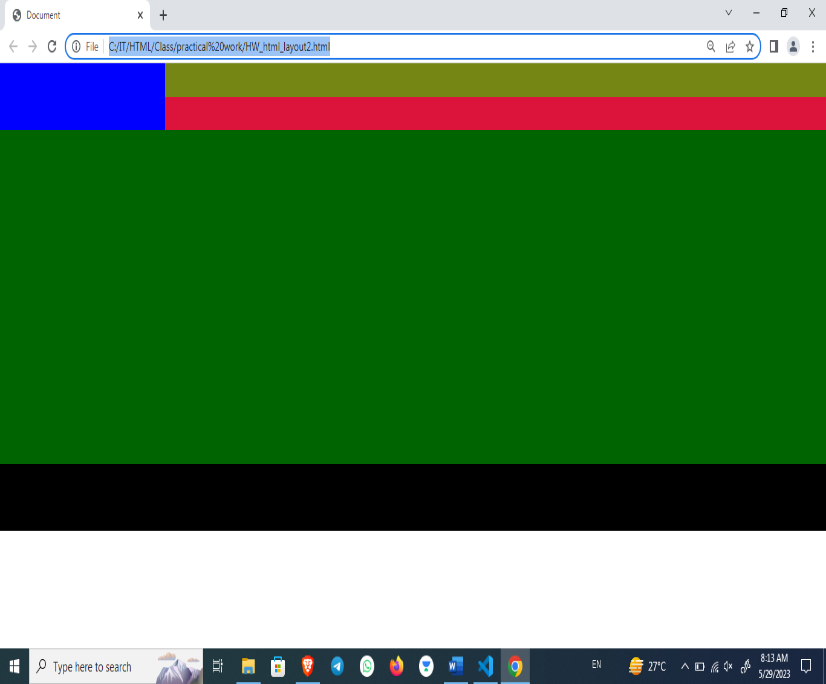
        }

    </style>

</head>

<body>

    <div id="header">

        <div id="logo"></div>

        <div id="link">

            <div id="l1"></div>

            <div id="l2"></div>

        </div>

    </div>

    <div id="main"></div>

    <div id="footer"></div>

</body>

</html>

Vijay Ratnottar