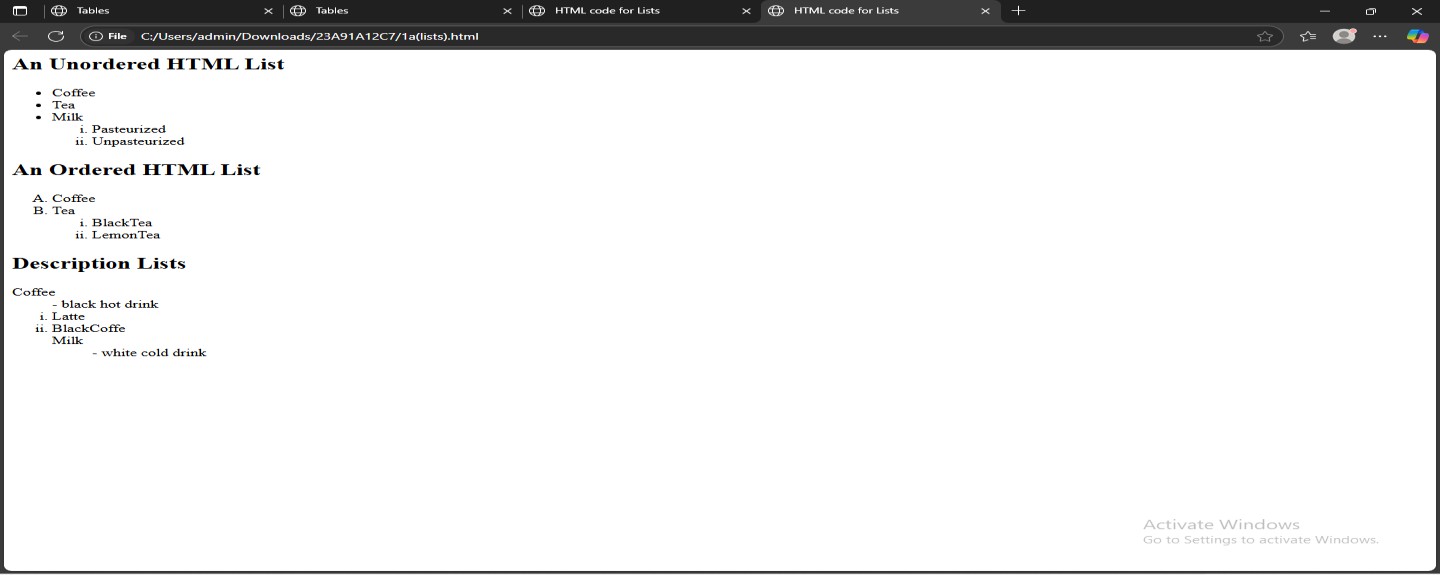
# Week-1



## 1)Lists, Links and Images

**a) Write a HTML program, to explain the working of lists. Note: It should have an ordered list, unordered list, nested lists and ordered list in an unordered list and definition lists.**

<html><head>

<title>HTML code for Lists</title></head>

<body>

<h2>An Unordered HTML List</h2>

<ul><li>Coffee</li><li>Tea</li><li>Milk</li>

<ol type="i">

<li>Pasteurized</li><li>Unpasteurized</li>

</ul>

<h2>An Ordered HTML List</h2>

<ol type="A">

<li>Coffee</li><li>Tea</li>

<ol type="i">

<li>BlackTea</li><li>LemonTea</li>

</ol></ol>

<h2>Description Lists</h2>

<dl>

<dt>Coffee</dt>

<dd>- black hot drink</dd>

<ol type="i">

<li>Latte</li><li>BlackCoffe</li>

<dt>Milk</dt>

<dd>- white cold drink</dd>

</dl>

</body>

</html>

# OUTPUT:

## Write a HTML program, to explain the working of hyperlinks using tag and href, target Attributes.



<html><head>

<title>HyperLinks in HTML</title>

</head><body>

<h1>Understanding Hyperlinks in HTML</h1>

<h2>Basic Hyperlink</h2>

<p>Click the link below to visit AEC:</p>

<a href="https://[www.aec.edu.in"](http://www.aec.edu.in/)>Visit aec</a>

<h2>Hyperlink with Target Attributes</h2>

<p>open the link in a new tab:</p>

<a href ="https:[www.aec.edu.in"](http://www.aec.edu.in/)target="-blank">Visit aec in a New Tab</a>

</body></html>

# OUTPUT:

## Create a HTML document that has your image and your friend’s image with a specific height and width. Also when clicked on the images it should navigate to their respective profiles.

<html><head>

<title>Image Link</title></head>

<body>

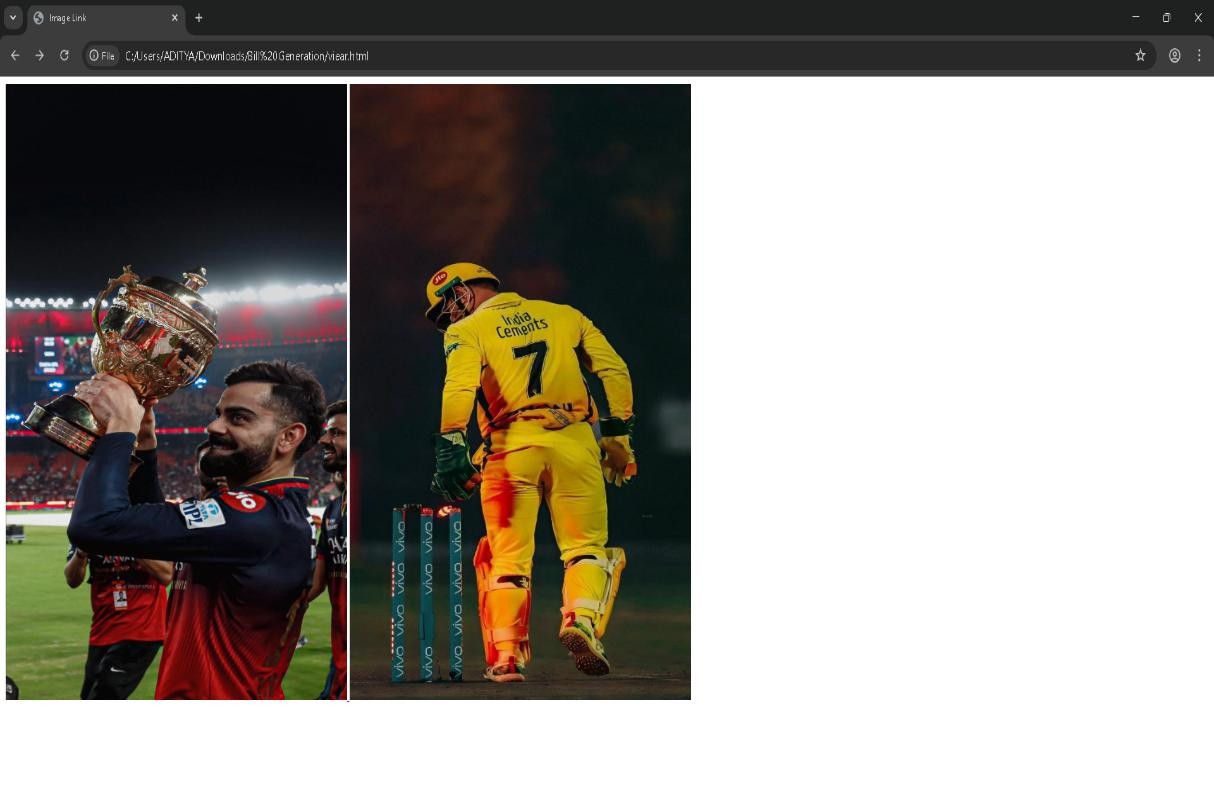
<a href ="https://[www.instagram.com/virat.kohli/?hl=en">](http://www.instagram.com/virat.kohli/?hl=en)

<img src= "https://preview.redd.it/can-someone-give-me-hd-or-higher-quality-photo- of-virat- wn08gbwghx5f1.jpeg?width=1252&auto=webp&s=a8e816d643481f32074a5fb96756d b9a2e5e4a0b" width="500", height="700">

<a href = "https://[www.instagram.com/mahi7781/?hl=en">](http://www.instagram.com/mahi7781/?hl=en)

<img src="https://i.pinimg.com/736x/c2/de/6b/c2de6b448b4177c659b76f718bc6306b.jpg" width="500" , height="700"></a>

</body></html>



# OUTPUT:

## Write a HTML program, in such a way that, rather than placing large images on a page, the preferred technique is to use thumbnails by setting the height and width parameters to something like to 100\*100 pixels. Each thumbnail image is also a link to a full sized version of the image. Create an image gallery using this technique

<!DOCTYPE html>

<html lang="en"><head>

<meta charset="UTF-8" />

<title>Thumbnail Image Gallery</title></head><body>

<h1>Image Gallery Using Thumbnails</h1>

<a href="https://images.unsplash.com/photo-1506744038136- 46273834b3fb?auto=format&fit=crop&w=800&q=80" target="\_blank">

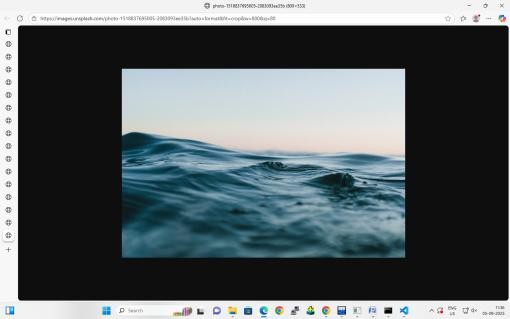
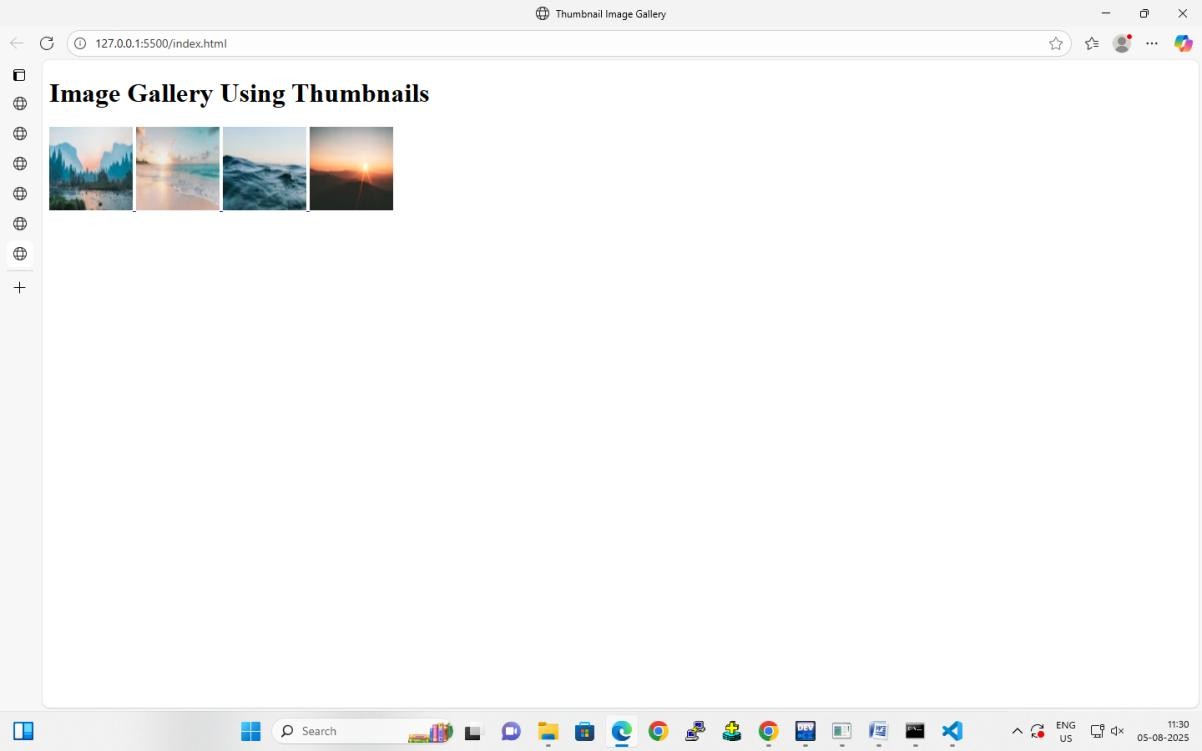
<img src="https://images.unsplash.com/photo-1506744038136- 46273834b3fb?auto=format&fit=crop&w=100&q=80" alt="Mountain" width="100" height="100" /></a>

<a href="https://images.unsplash.com/photo-1507525428034- b723cf961d3e?auto=format&fit=crop&w=800&q=80" target="\_blank">

<img src="https://images.unsplash.com/photo-1507525428034- b723cf961d3e?auto=format&fit=crop&w=100&q=80" alt="Beach" width="100" height="100" /></a>

<a href="https://images.unsplash.com/photo-1518837695005- 2083093ee35b?auto=format&fit=crop&w=800&q=80" target="\_blank">

<img src="https://images.unsplash.com/photo-1518837695005- 2083093ee35b?auto=format&fit=crop&w=100&q=80" alt="Forest" width="100" height="100" /></a>



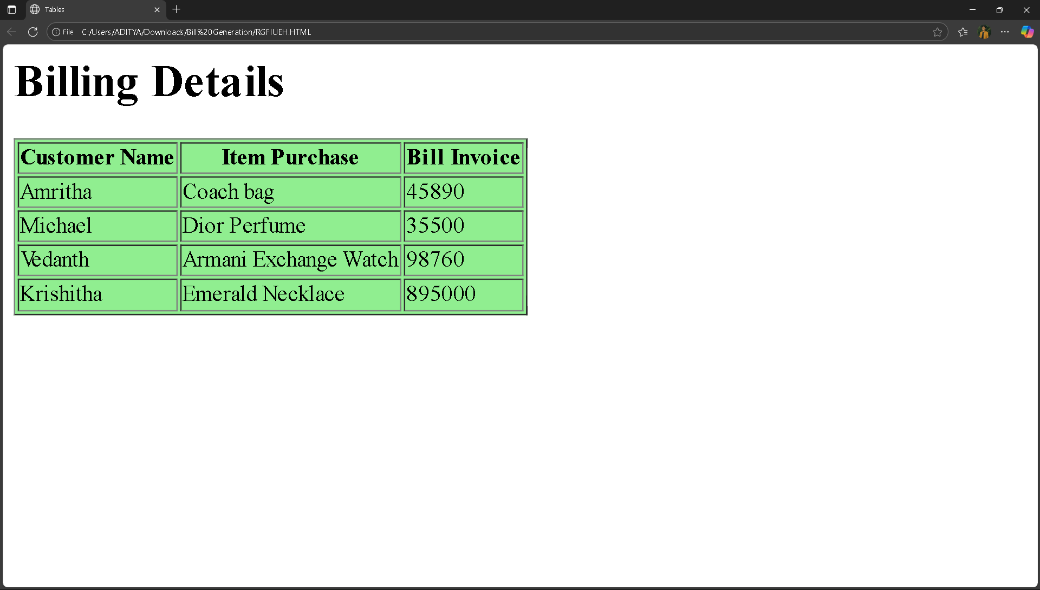
<a href="https://images.unsplash.com/photo-1500534623283- 312aade485b7?auto=format&fit=crop&w=800&q=80" target="\_blank">

<img src="https://images.unsplash.com/photo-1500534623283- 312aade485b7?auto=format&fit=crop&w=100&q=80" alt="Desert" width="100" height="100" /></a>

</body>

</html>

# OUTPUT:



**WEEK-2**

## Write a HTML program, to explain the working of tables.

<html><head>

<title>Tables</title>

</head><body>

<h1>Billing Details</h1>

<table bgcolor= "lightgreen" Border="1">

<tr>

<th>Customer Name</th>

<th>Item Purchase</th>

<th>Bill Invoice</th>

</tr>

<tr>

<td>Amritha</td>

<td>Coach bag</td>

<td>45890</td>

</tr>

<tr>

<td>Michael</td>

<td>Dior Perfume</td>

<td>35500</td>

</tr>

<tr>

<td>Vedanth</td>

<td>Armani Exchange Watch</td>

<td>98760</td>

</tr>

<tr><td>Krishitha</td> <td>Emerald Necklace</td> <td>895000</td>

</tr></table></body></html>

# OUTPUT:

## Write a HTML program, to explain the working of tables by preparing a timetable. (Note: Use tag to set the caption to the table & also use cell spacing, cell padding, border, rowspan, colspan etc.)

<title>TIME TABLE</title>

<table border="1" align="center" cellpadding="14" cellspacing="0">

<thead>

<tr><th>Room No.</th>

<th>Section</th>

<th>DAY</th>

<th>1<br>9:30-10:20</th>

<th>2<br>10:20-11:10</th>

<th>3<br>11:10-12:00</th>

<th> <br>12:00-1:00</th>

<th>4<br>1:00-1:50</th>

<th>5<br>1:50-2:40</th>

<th>6<br>2:40-3:30</th>

<th>7<br>3:30-4:20</th></tr>

</thead>

<tbody>

<tr><td rowspan="6">221</td>

<td rowspan="6">III IT-B</td>

<th>MON</th>

<td>AJ</td>

<td colspan="2" align="center">PSA</td>

<td rowspan="6" align="center">LUNCH</td>

<td>AT&CD</td>

<td>ET</td>

<td>DWDM</td>

<td>FSD LAB</td></tr>

<tr><th>TUE</th>

<td colspan="3" align="center">CN LAB</td>

<td>AJ</td>

<td>AT&CD</td>

<td>CN</td>

<td>FLUTTER LAB</td></tr>

<tr><th>WED</th>

<td colspan="3" align="center">AJ LAB</td>

<td>DWDM</td>

<td>CN</td>

<td>AJ</td>

<td>ET</td></tr>

<tr><th>THU</th>

<td colspan="3" align="center">FLUTTER LAB</td>

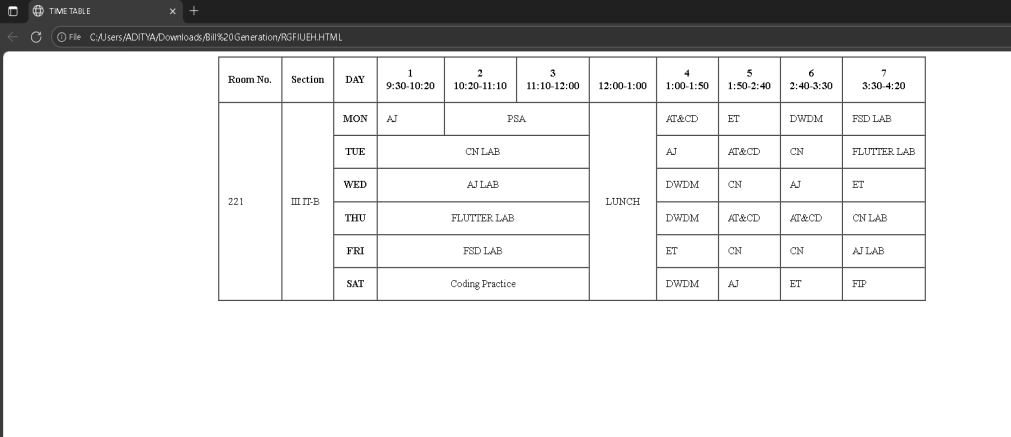
<td>DWDM</td>

<td>AT&CD</td>

<td>AT&CD</td>

<td>CN LAB</td></tr>

<tr>



<tr><th>FRI</th>

<td colspan="3" align="center">FSD LAB</td>

<td>ET</td>

<td>CN</td>

<td>CN</td>

<td>AJ LAB</td></tr>

<tr><th>SAT</th>

<td colspan="3" align="center">Coding Practice</td>

<td>DWDM</td>

<td>AJ</td>

<td>ET</td>

<td>FIP</td> </tr>

</tbody></table>

# OUTPUT:

## Write a HTML program, to explain the working of forms by designing Registration form. (Note: Include text field, password field, number field, date of birth field, checkboxes, radio buttons, list boxes using and two buttons ie: submit and reset. Use tables to provide a better view).

<title>FORM</title>

<form>

<label>First name:</label><br>

<input type="text"><br>

<label>Last name:</label><br>

<input type="text"><br>

<label for="'program">Select Program:</label><br>

<select id="program" name="'program" required >

<option>B.Tech</option>

<option>M.Tech</option>

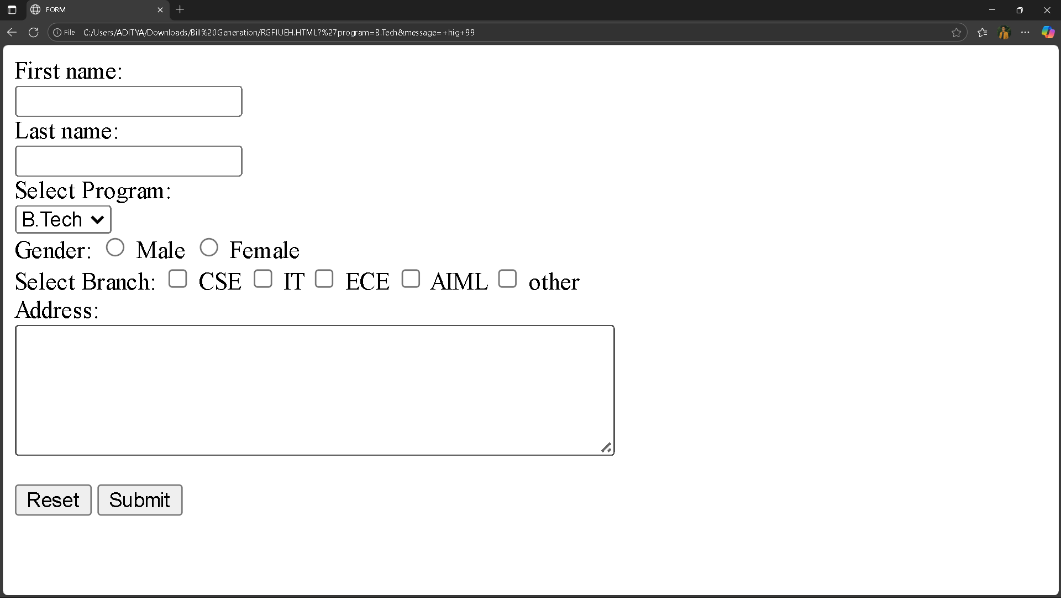
<option>MBA</option>

<option>MCA</option>

<option>BBA</option>

<option>other</option>

</select><br>



<label>Gender:</label>

<input type="radio">

<label for="html">Male</label>

<input type="radio">

<label for="css">Female</label><br>

<label>Select Branch:</label>

<input type="checkbox">

<label>CSE</label>

<input type="checkbox">

<label>IT</label>

<input type="checkbox">

<label>ECE</label>

<input type="checkbox">

<label>AIML</label>

<input type="checkbox">

<label>other</label><br>

<label>Address:</label><br>

<textarea name="message" rows="5" cols="50"> </textarea><br><br>

<input type="submit" value="Reset">

<input type="submit" value="Submit"></form>

# OUTPUT:

## Write a HTML program, to explain the working of frames, such that page is to be divided into 3 parts on either direction. (Note: first frame image, second frame paragraph, third frame ◻ hyperlink. And also make sure of using “no frame” attribute such that frames to be fixed).

<title>Frame</title>

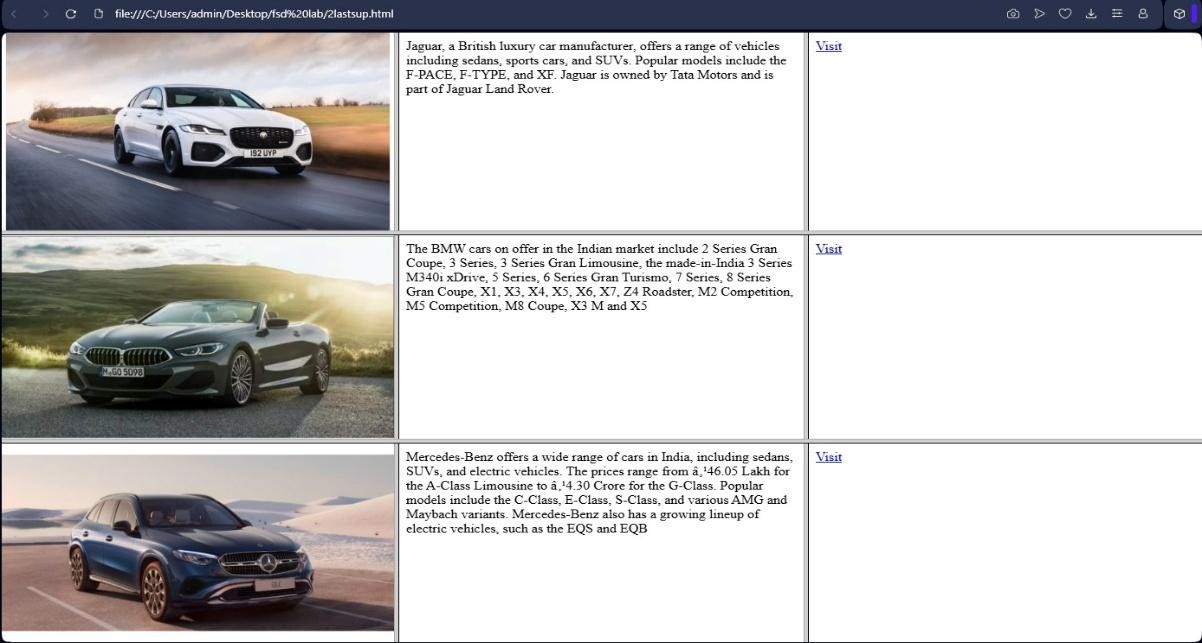
<frameset rows="33%,34%,33%"><frameset cols="33%,34%,33%">

<frame src="data:text/html,

<body style='margin:0;padding:0;overflow:hidden;'>

<img src='https://images.overdrive.in/wp-content/uploads/2021/10/2021-jaguar-xf- facelift-india-01.jpg' style='width:100%;height:100%;object-fit:contain;'>

</body>" name="frame1">



<frame src="data:text/html,<p>Jaguar, a British luxury car manufacturer, offers a range of vehicles including sedans, sports cars, and SUVs. Popular models include the F-PACE, F- TYPE, and XF. Jaguar is owned by Tata Motors and is part of Jaguar Land Rover. </p>" name="frame2">

<frame src="data:text/html,<a href='https:/[/www.carwale.com/jaguar-](http://www.carwale.com/jaguar-) cars/'target='\_blank'>Visit</a>" name="frame3"> </frameset>

<frameset cols="33%,34%,33%">

<frame src="data:text/html,<body style='margin:0;padding:0;overflow:hidden;'>

<img src='https://cdn.zeebiz.com/sites/default/files/2018/11/05/58652-bmw-8-series- convertible-twitter.PNG' style='width:100%;height:100%;object-fit:contain;'>

</body>" name="frame4">

<frame src="data:text/html,<p>The BMW cars on offer in the Indian market include 2 Series Gran Coupe, 3 Series, 3 Series Gran Limousine, the made-in-India 3 Series M340i xDrive, 5 Series, 6 Series Gran Turismo, 7 Series, 8 Series Gran Coupe, X1, X3, X4, X5, X6, X7, Z4 Roadster, M2 Competition, M5 Competition, M8 Coupe, X3 M and X5</p>" name="frame5">

<frame src="data:text/html,<a href='https:/[/www.carwale.com/bmw-](http://www.carwale.com/bmw-) cars/'target='\_blank'>Visit</a>" name="frame6">

</frameset><frameset cols="33%,34%,33%">

<frame src="data:text/html,<body style='margin:0;padding:0;overflow:hidden;'>

<img src='https:/[/www.godigit.com/content/dam/godigit/directportal/en/mercedes-benz-](http://www.godigit.com/content/dam/godigit/directportal/en/mercedes-benz-) glc-brand.jpg' style='width:100%;height:100%;object-fit:contain;'>

</body>" name="frame7">

<frame src="data:text/html,<p>Mercedes-Benz offers a wide range of cars in India, including sedans, SUVs, and electric vehicles. The prices range from ₹46.05 Lakh for the A- Class Limousine to ₹4.30 Crore for the G-Class. Popular models include the C-Class, E- Class, S-Class, and various AMG and Maybach variants. Mercedes-Benz also has a growing lineup of electric vehicles, such as the EQS and EQB</p>" noresize name="frame8">

<frame src="data:text/html,<a href='https:/[/www.carwale.com/mercedes-benz-](http://www.carwale.com/mercedes-benz-) cars/'target='\_blank'>Visit</a>" name="frame9"> </frameset>

<noframes><p>Your browser does not support frames.</p></noframes></frameset>

# OUTPUT:

**WEEK-3**

## Write a HTML program, that makes use of<article>,<aside>,<figure>,<figcaption>,

**<footer>,<header>,<main>,<nav>,<section>,<div>,<span>tags.**

<!DOCTYPE html>

<html lang="en"><head>

<meta charset="UTF-8">

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

<title>HTML Semantic Tags Example</title><style> body {

font-family: Arial, sans-serif; margin: 0;

padding: 0 } header, footer {

background-color: #333; color: white;

text-align: center; padding: 15px;} nav {

background-color: #444; padding: 10px;}

nav a {

color: white; margin: 0 10px;

text-decoration: none;} main {

display: flex; padding: 20px;} article {

flex: 3; padding: 10px;} aside {

flex: 1; padding: 10px;

background-color: #f0f0f0} figure {

margin: 10px 0;} figcaption {

font-size: 0.9em; color: gray;}

section {

margin-bottom: 20px;} div.example {

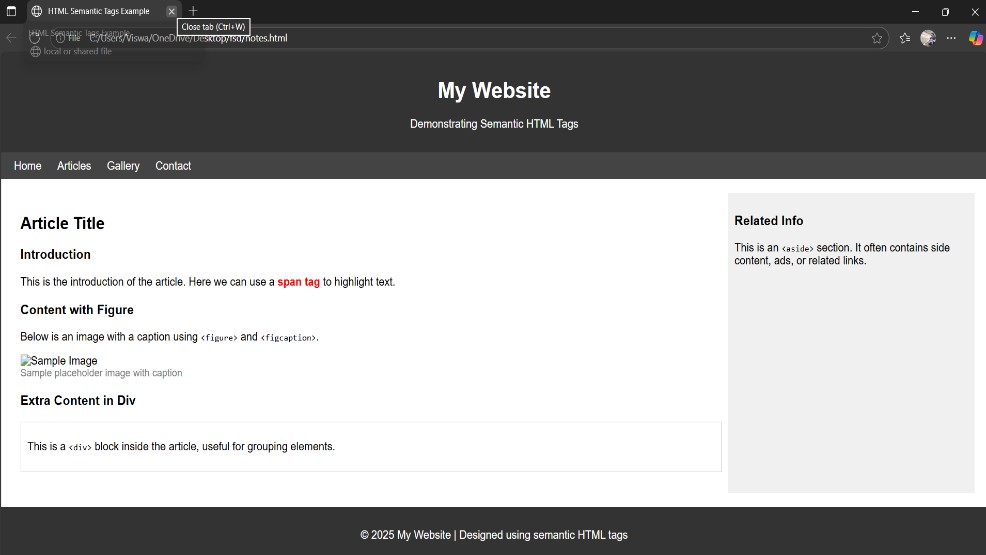
border: 1px solid #ccc; padding: 10px;

margin-top: 10px; span.highlight {

color: red;

font-weight: bold;}

</style></head><body>



<header><h1>My Website</h1>

<p>Demonstrating Semantic HTML Tags</p>

</header><nav><a href="#">Home</a>

<a href="#">Articles</a>

<a href="#">Gallery</a>

<a href="#">Contact</a></nav>

<main><article>

<h2>Article Title</h2>

<section><h3>Introduction</h3>

<p>This is the introduction of the article. Here we can use a <span class="highlight">span tag</span> to highlight text.</p></section>

<section>

<h3>Content with Figure</h3>

<p>Below is an image with a caption using <code>&lt;figure&gt;</code> and

<code>&lt;figcaption&gt;</code>.</p>

<figure>

<img src="https://via.placeholder.com/300x150" alt="Sample Image">

<figcaption>Sample placeholder image with caption</figcaption>

</figure></section>

<section>

<h3>Extra Content in Div</h3>

<div class="example">

<p>This is a <code>&lt;div&gt;</code> block inside the article, useful for grouping elements.</p>

</div></section></article><aside>

<h3>Related Info</h3>

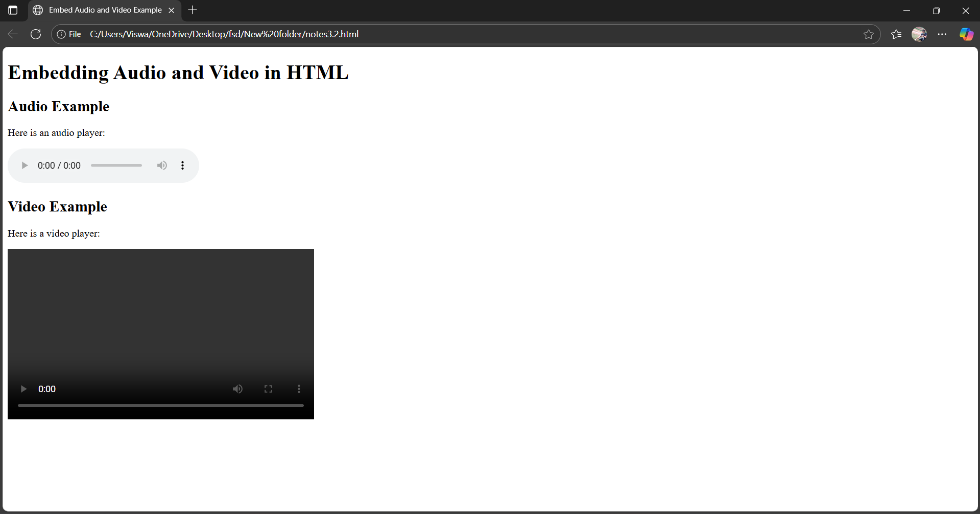
<p>This is an <code>&lt;aside&gt;</code> section. It often contains side content, ads, or related links.</p>

</aside></main><footer>

<p>&copy; 2025 My Website | Designed using semantic HTML tags</p>

</footer></body></html>

# OUTPUT:



## Write a HTML program, to embed audio and video into HTML web page.

<!DOCTYPE html><html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Embed Audio and Video Example</title>

</head><body>

<h1>Embedding Audio and Video in HTML</h1>

<h2>Audio Example</h2>

<p>Here is an audio player:</p>

<audio controls>

<source src="sample-audio.mp3" type="audio/mpeg">

<source src="sample-audio.ogg" type="audio/ogg">

Your browser does not support the audio element.</audio>

<h2>Video Example</h2>

<p>Here is a video player:</p>

<video width="480" height="270" controls>

<source src="sample-video.mp4" type="video/mp4">

<source src="sample-video.ogg" type="video/ogg"> Your browser does not support the video element.

</video></body></html>

# OUTPUT:

## C. Write a program to apply different types (of levels of styles or style specification formats) - inline, internal, external styles to HTML elements. (identify selector, property and value).

**index.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

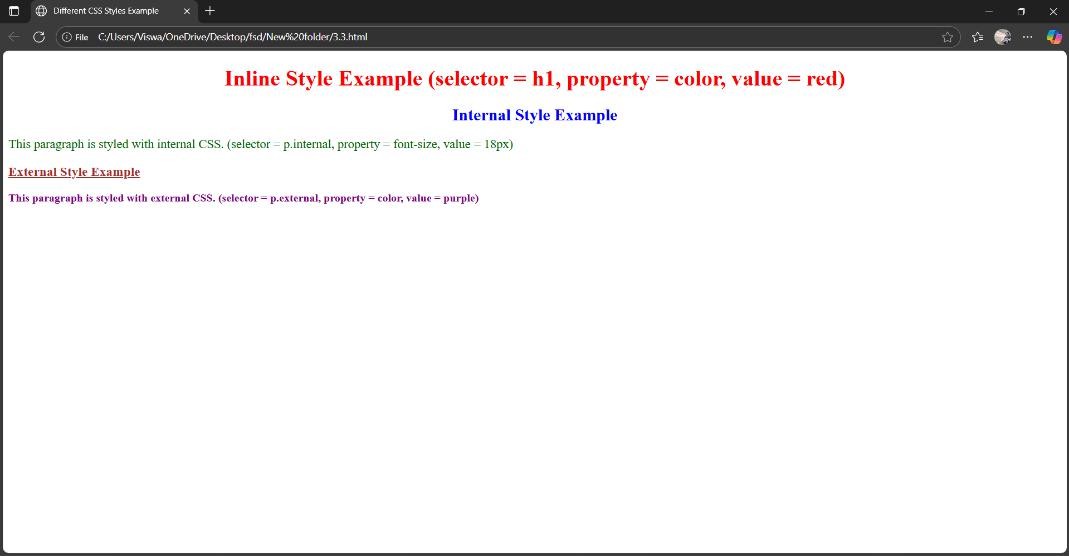
<title>Different CSS Styles Example</title>

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

<style>

/\* Internal style: selector = h2, property = color, value = blue \*/

h2 {



color: blue;

text-align: center; }

/\* Internal style for paragraph \*/ p.internal {

font-size: 18px; color: darkgreen;}

</style>

</head><body>

<h1 style="color: red; text-align:center;">

Inline Style Example (selector = h1, property = color, value = red)</h1>

<h2>Internal Style Example</h2>

<p class="internal">

This paragraph is styled with internal CSS. (selector = p.internal, property = font-size, value = 18px)</p>

<h3>External Style Example</h3>

<p class="external">

This paragraph is styled with external CSS. (selector = p.external, property = color, value

= purple)</p>

</body>

</html>

# Styles.css

/\* External style: selector = p.external, property = color, value = purple \*/ p.external {

color: purple; font-weight: bold; font-size: 16px;

}

/\* External style for h3 \*/ h3 {

color: brown;

text-decoration: underline;}

# OUTPUT:

}

# WEEK-4

## a)Write a program to apply different types of selector forms i. Simple selector (element, id, class, group, universal) ii. Combinator selector (descendant, child, adjacent sibling, general sibling) iii.Pseudo-class selector iv. Pseudo-element selector v. Attribute selector

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>CSS Selector Examples</title>

<style>

/\* Simple Selectors \*/

/\* Element Selector \*/ p {

color: darkblue;

}

/\* ID Selector \*/ #main-heading { color: green;

text-align: center;

}

/\* Class Selector \*/

.highlight {

background-color: yellow; font-weight: bold;

}

/\* Group Selector \*/ h2, h3 {

color: purple;

}

/\* Universal Selector \*/

\* {

font-family: Arial, sans-serif;

}

/\* ---------------- Combinator Selectors \*/

/\* Descendant Selector \*/ div p {

color: teal;

}

/\* Child Selector \*/ div > p {

border: 1px dashed red; padding: 5px;

}

/\* Adjacent Sibling Selector \*/ h2 + p {

color: brown;

}

/\* General Sibling Selector \*/ h2 ~ p {

font-style: italic;

}

/\* ---------------- Pseudo-class Selectors \*/

a:hover { color: red;

text-decoration: underline;

}

li:first-child { color: green;

}

li:last-child { color: blue;

}

input:focus {

border: 2px solid orange;

}

/\* ---------------- Pseudo-element Selectors \*/

p::first-line {

font-weight: bold p::first-letter { font-size: 24px; color: darkred; h1::after { content: " - ` '";

/\* Attribute Selectors \*/ a[target] {

color: darkgreen;

}

a[target="\_blank"] { font-weight: bold;

}

input[type="text"] { background-color: #f0f8ff;

}

[title~="important"] { border: 2px solid gold;

}

</style>

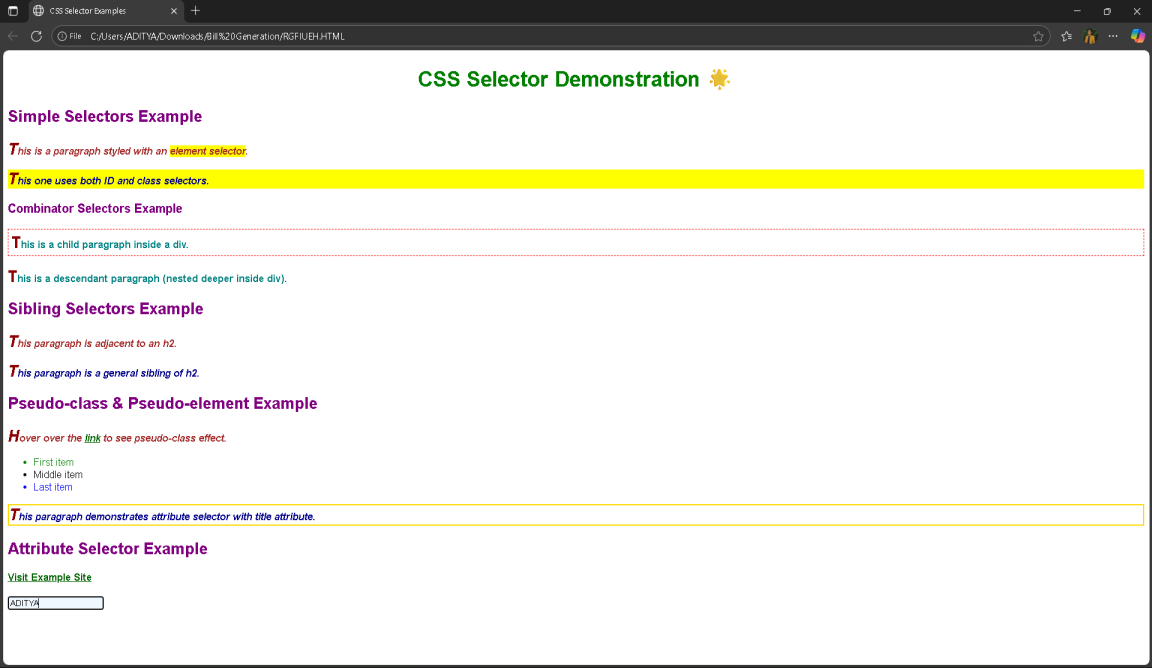
</head>

<body>

<h1 id="main-heading">CSS Selector Demonstration</h1>

<h2>Simple Selectors Example</h2>

<p>This is a paragraph styled with an <span class="highlight">element selector</span>.</p>



p id="unique-para" class="highlight">This one uses both ID and class selectors.</p>

<h3>Combinator Selectors Example</h3>

<div>

<p>This is a child paragraph inside a div.</p>

<section>

<p>This is a descendant paragraph (nested deeper inside div).</p>

</section>

</div>

<h2>Sibling Selectors Example</h2>

<p>This paragraph is adjacent to an h2.</p>

<p>This paragraph is a general sibling of h2.</p>

<h2>Pseudo-class & Pseudo-element Example</h2>

<p>Hover over the <a href="#" target="\_blank">link</a> to see pseudo-class effect.</p>

<ul>

<li>First item</li>

<li>Middle item</li>

<li>Last item</li>

</ul>

<p title="important note">This paragraph demonstrates attribute selector with title attribute.</p>

<h2>Attribute Selector Example</h2>

<a href="https://example.com" target="\_blank">Visit Example Site</a><br><br>

<input type="text" placeholder="Type something here...">

</body>

</html>

# OUTPUT:

**WEEK-5**

## a)Write a program to demonstrate the various ways you can reference a color in CSS.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Color Reference in CSS</title>

<style> body {

font-family: Arial, sans-serif; text-align: center; background-color: #f5f5f5;}

h1 {

color: #333;}

/\* 1. Named Color \*/

.named-color { background-color: blue; color: white;}

/\* 2. Hexadecimal Color \*/

.hex-color {

background-color: #ff5733; color: #ffffff;}

/\* 3. RGB Color \*/

.rgb-color {

background-color: rgb(60, 179, 113);

color: rgb(255, 255, 255);}

/\* 4. RGBA Color (with transparency) \*/

.rgba-color {

background-color: rgba(255, 0, 0, 0.5); color: black;}

/\* 5. HSL Color \*/

.hsl-color {

background-color: hsl(200, 100%, 50%); color: white;}

/\* 6. HSLA Color (with transparency) \*/

.hsla-color {

background-color: hsla(300, 70%, 60%, 0.6); color: black;}

/\* 7. currentColor keyword \*/

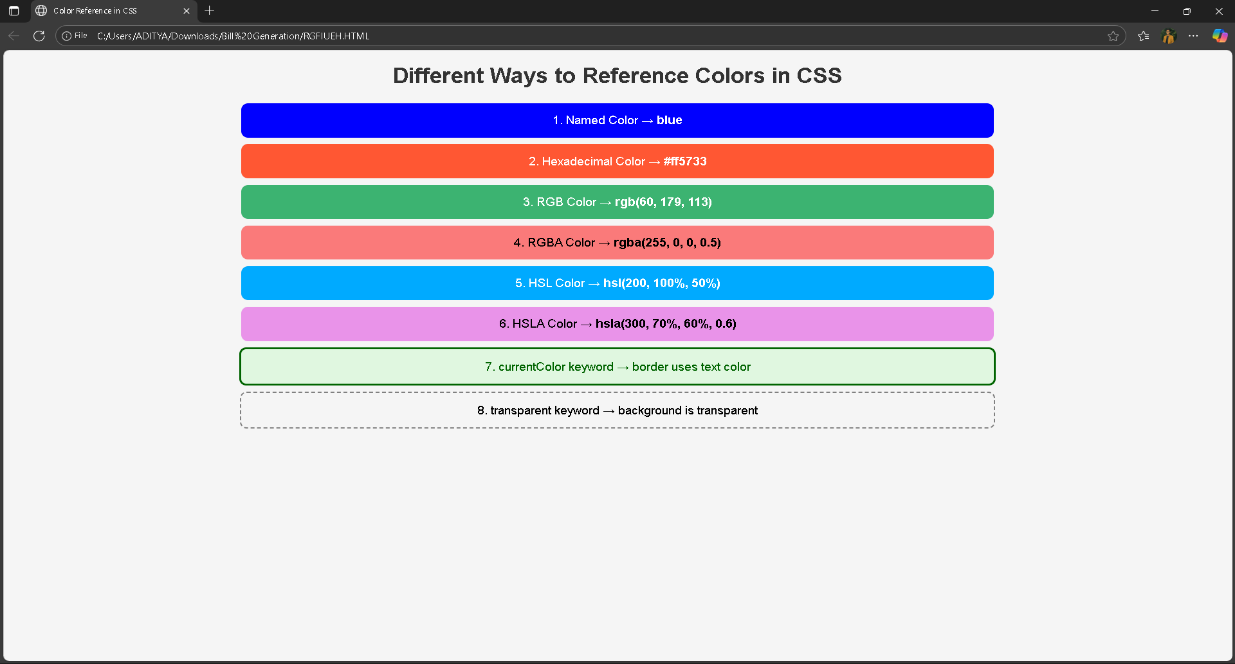
.current-color { color: darkgreen;

border: 3px solid currentColor; background-color: #e0f7e0;}

/\* 8. transparent keyword \*/

.transparent-color {

background-color: transparent;



border: 2px dashed gray; color: black;

}

/\* General box styling \*/

.color-box { margin: 10px auto; padding: 15px; width: 60%;

border-radius: 10px; font-size: 18px;

}

</style></head><body>

<h1>Different Ways to Reference Colors in CSS</h1>

<div class="color-box named-color">1. Named Color → <b>blue</b></div>

<div class="color-box hex-color">2. Hexadecimal Color → <b>#ff5733</b></div>

<div class="color-box rgb-color">3. RGB Color → <b>rgb(60, 179, 113)</b></div>

<div class="color-box rgba-color">4. RGBA Color → <b>rgba(255, 0, 0, 0.5)</b></div>

<div class="color-box hsl-color">5. HSL Color → <b>hsl(200, 100%, 50%)</b></div>

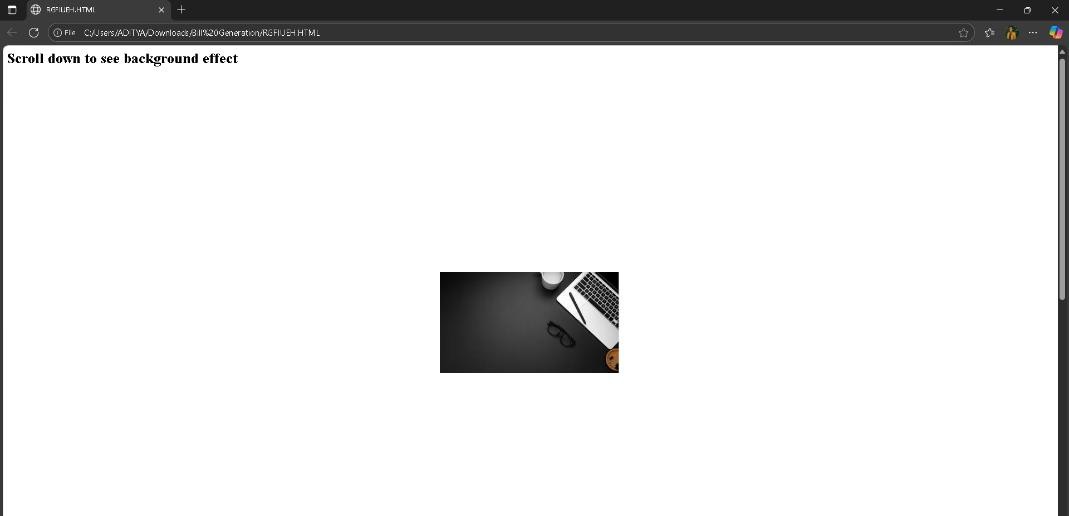
<div class="color-box hsla-color">6. HSLA Color → <b>hsla(300, 70%, 60%, 0.6)</b></div>

<div class="color-box current-color">7. currentColor keyword → border uses text color</div>

<div class="color-box transparent-color">8. transparent keyword → background is transparent</div>

</body></html>

# OUTPUT:



## b. Write a CSS rule that places a background image halfway down the page, tilting it horizontally. The image should remain in place when the user scrolls up or down. CSS File:

<html>

<head>

<style> body {

height: 2000px; /\* to allow scrolling \*/

background-image: url(‘https://img.freepik.com/free-photo/flat-lay-workstation-with-copy- space-laptop\_23-2148430879.jpg?semt=ais\_hybrid&w=740&q=80’);

background-repeat: no-repeat;

background-position: center 50%; /\* halfway down \*/ background-attachment: fixed; /\* stays fixed when scrolling \*/

}

</style>

</head>

<body>

<h2>Scroll down to see background effect</h2>

</body>

</html>

# OUTPUT:

**C.Write a program using the following terms related to CSS font and text:**

1. **font-size**
2. **font-weight**
3. **font-style**

<!DOCTYPE html>

<html lang="en">

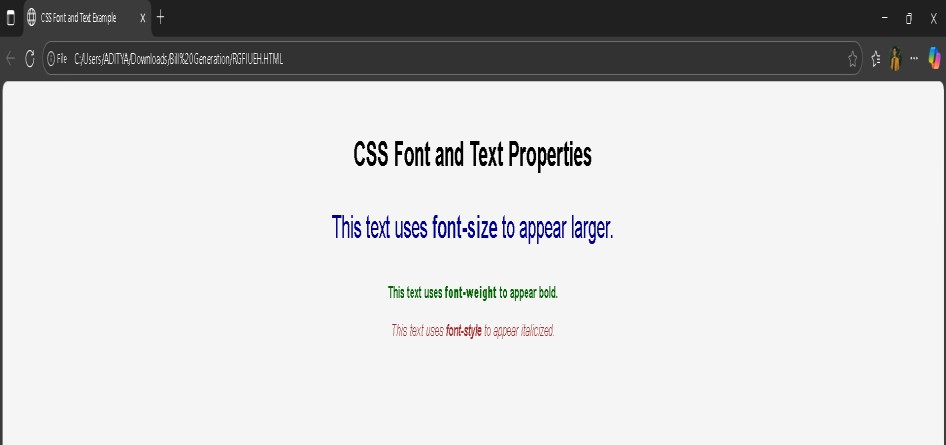
<head>

<meta charset="UTF-8">

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

<title>CSS Font and Text Example</title>

<style> body {



font-family: Arial, sans-serif; background-color: #f5f5f5; text-align: center;

margin-top: 50px;

/\* i. font-size \*/

.large-text {

font-size: 30px; /\* sets the size of the font \*/ color: darkblue;

}

/\* ii. font-weight \*/

.bold-text {

font-weight: bold; /\* makes the text thicker \*/ color: darkgreen;

}

/\* iii. font-style \*/

.italic-text {

font-style: italic; /\* makes the text slanted \*/ color: brown;

}

</style>

</head>

<body>

<h1>CSS Font and Text Properties</h1>

<p class="large-text">This text uses <b>font-size</b> to appear larger.</p>

<p class="bold-text">This text uses <b>font-weight</b> to appear bold.</p>

<p class="italic-text">This text uses <b>font-style</b> to appear italicized.</p>

</body>

</html>

# OUTPUT:

## d. Write a program, to explain the importance of CSS Box model using i. Content

**ii. Border iii. Margin iv. Padding**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>CSS Box Model Example</title>

<style> body {

font-family: Arial, sans-serif; background-color: #f9f9f9; text-align: center;

margin: 50px;

}

h1 {

color: darkblue;

}

/\* Box Model Example \*/

.box {

width: 300px; /\* Content width \*/ background-color: #e0f7fa; /\* Content area color \*/

padding: 20px; /\* Space between content and border \*/ border: 5px solid #00796b; /\* Border around the padding \*/

margin: 30px auto; /\* Space outside the box (between boxes or edges) \*/ text-align: left;

}

.box {

width: 300px; /\* Content width \*/ background-color: #e0f7fa; /\* Content area color \*/

padding: 20px; /\* Space between content and border \*/ border: 5px solid #00796b; /\* Border around the padding \*/

margin: 30px auto; /\* Space outside the box (between boxes or edges) \*/ text-align: left;

}

.box {

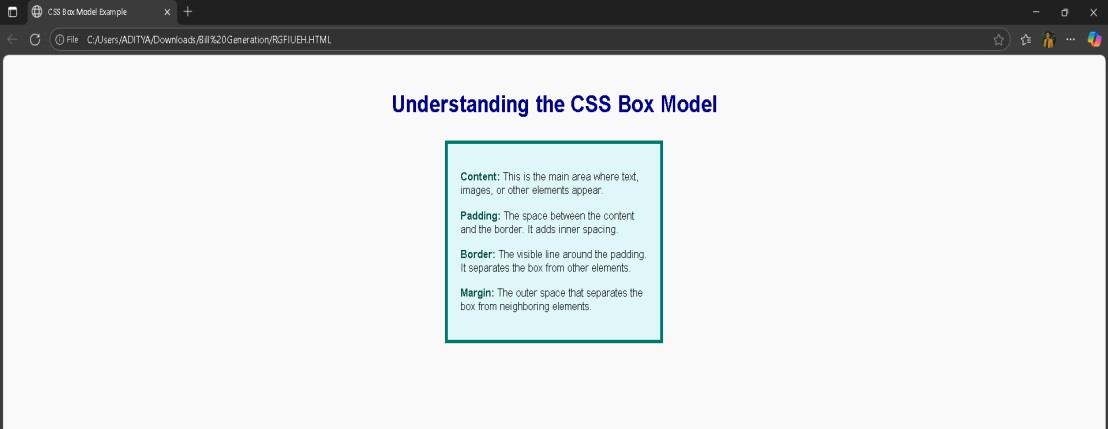
width: 300px; /\* Content width \*/ background-color: #e0f7fa; /\* Content area color \*/

padding: 20px; /\* Space between content and border \*/ border: 5px solid #00796b; /\* Border around the padding \*/

margin: 30px auto; /\* Space outside the box (between boxes or edges) \*/ text-align: left;

}

.label {



font-weight: bold; color: #004d40;

}

</style>

</head>

<body>

<h1>Understanding the CSS Box Model</h1>

<div class="box">

<p><span class="label">Content:</span> This is the main area where text, images, or other elements appear.</p>

<p><span class="label">Padding:</span> The space between the content and the border.

It adds inner spacing.</p>

<p><span class="label">Border:</span> The visible line around the padding. It separates the box from other elements.</p>

<p><span class="label">Margin:</span> The outer space that separates the box from neighboring elements.</p>

</div>

</body>

</html>

OUTPUT:

# WEEK-6

## Write a program to embed internal and external JavaScript in a web page

<!-- Internal JavaScript -->

## Save as file.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Internal and External JavaScript Example</title>

<script src="script.js" defer></script>

<style> body {

font-family: Arial, sans-serif; text-align: center;

margin-top: 50px;

}

button {

padding: 10px 20px; font-size: 16px; cursor: pointer;

}

</style>

</head>

<body>

<h1>Internal and External JavaScript Example</h1>

<button onclick="internalMessage()">Click for Internal JS</button>

<button onclick="externalMessage()">Click for External JS</button>

<!-- Internal JavaScript -->

<script>

function internalMessage() {

alert("Hello from Internal JavaScript!");

}

</script>

</body>

</html>

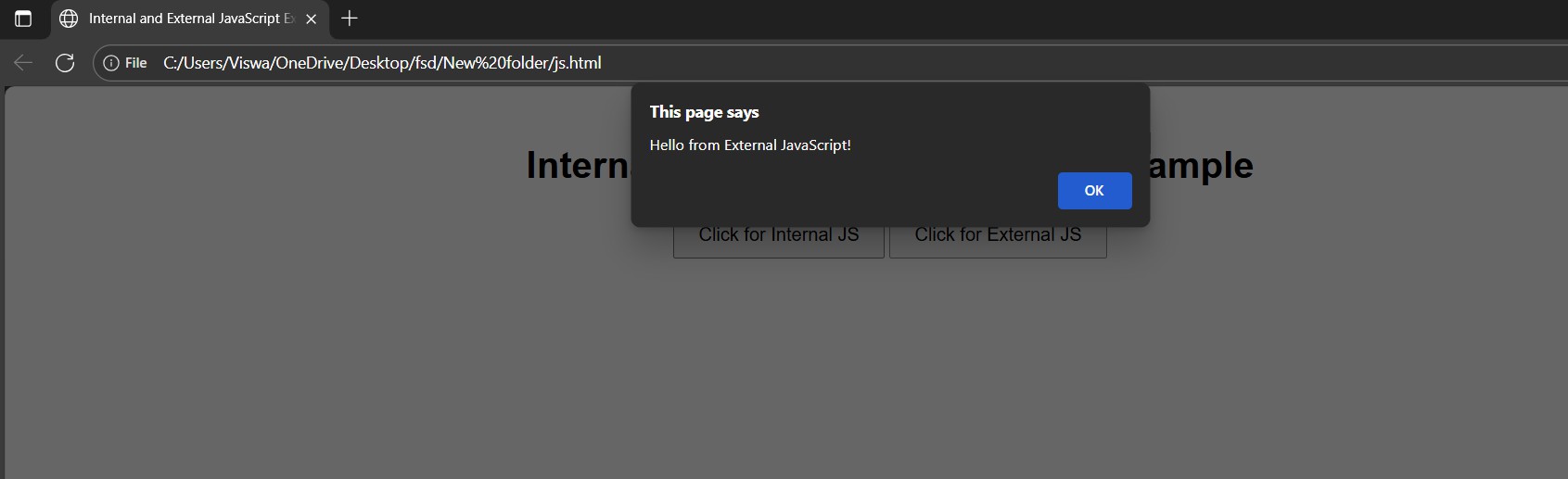
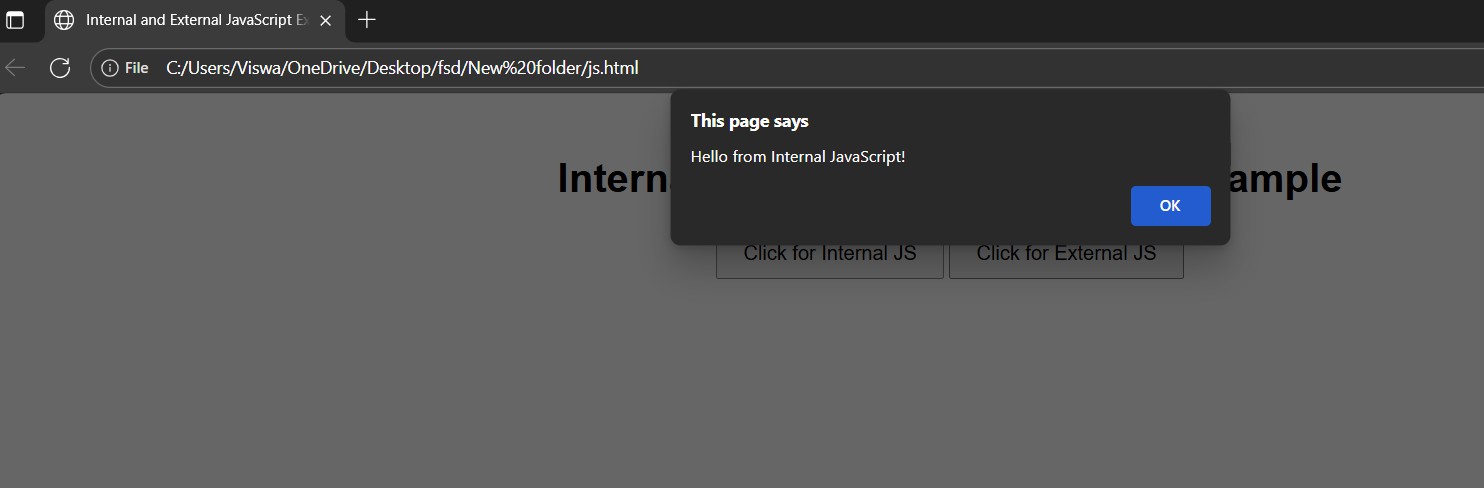
<!-- Link to External JavaScript -->

## Save as script.js

function externalMessage() {

alert("Hello from External JavaScript!");

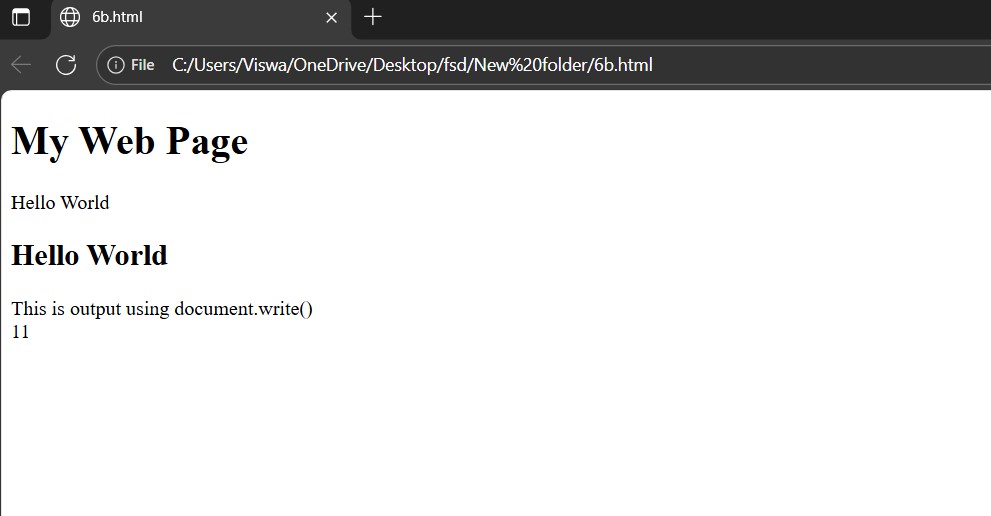
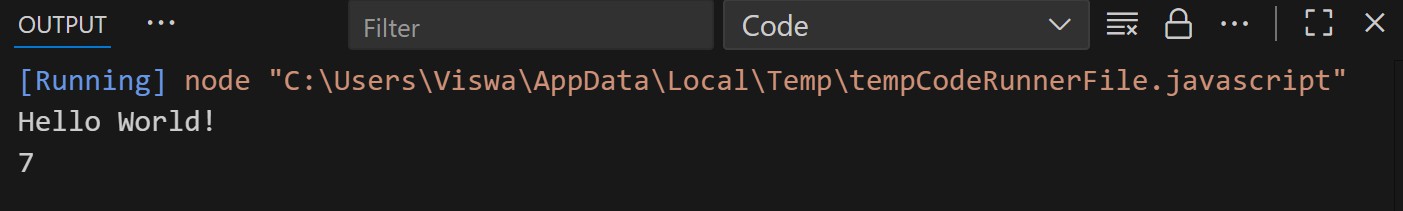
}



# OUTPUT SCREEN1:

**OUTPUT SCREEN2:** //for internal JavaScript

**OUTPUT SCREEN2:** //for external JavaScript



## Write a program to explain the different ways for displaying output.

**Using console.log()** console.log(“Hello World!”); console.log(5+2);

## Using innerText, innerHTML & document.write

<!DOCTYPE html>

<html>

<body>

<h1>My Web Page</h1>

<p id="demo1"></p>

<p id="demo2"></p>

<script>

document.getElementById("demo1").innerText = "Hello World";

</script>

<script>

document.getElementById("demo2").innerHTML = "<h2>Hello World</h2>";

</script>

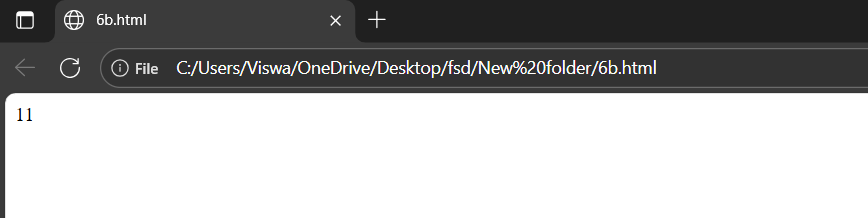
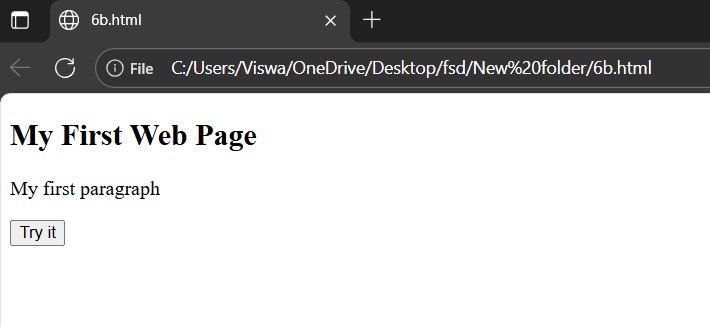
<script>

document.write("This is output using document.write()<br>"); document.write(5 + 6);

</script>

</body>

</html>



## //additional example for document.write

<!DOCTYPE html>

<html>

<body>

<h2>My First Web Page</h2>

<p>My first paragraph</p>

<button type="button" onclick="document.write(5 + 6)">Try it</button>

</body>

</html>

# OUTPUT SCREEN1:

**SCREEN2:**

## Using window.alert()

<!DOCTYPE html>

<html>

<body>

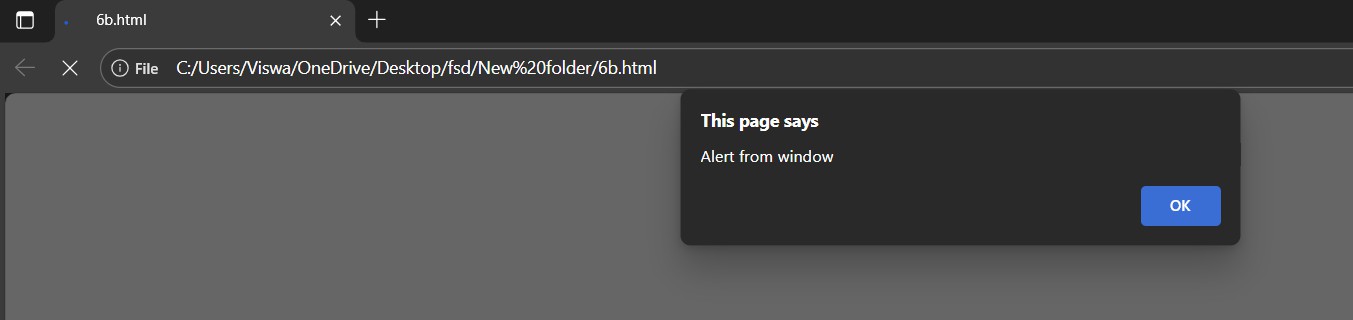
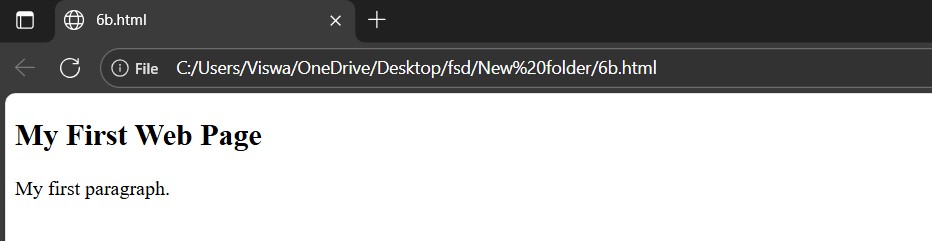
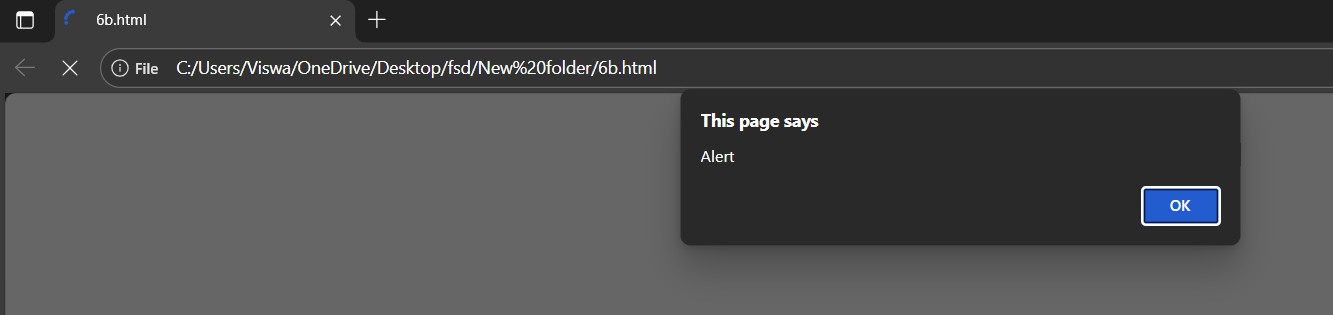
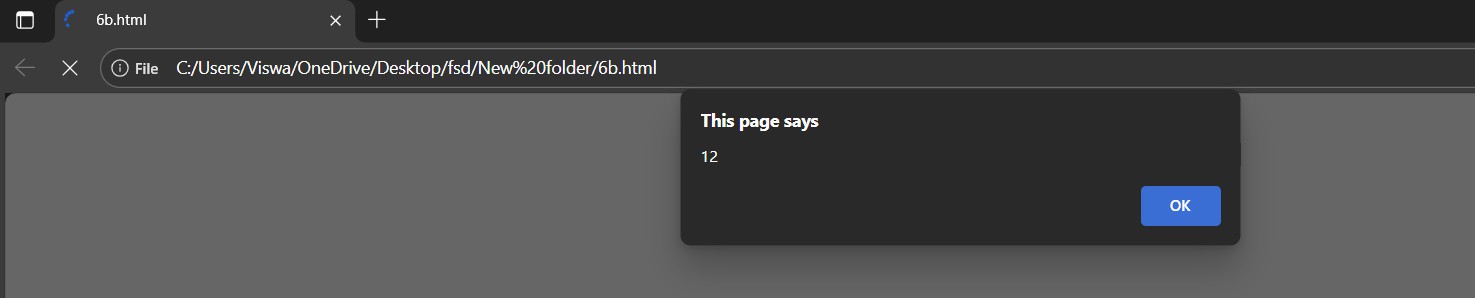
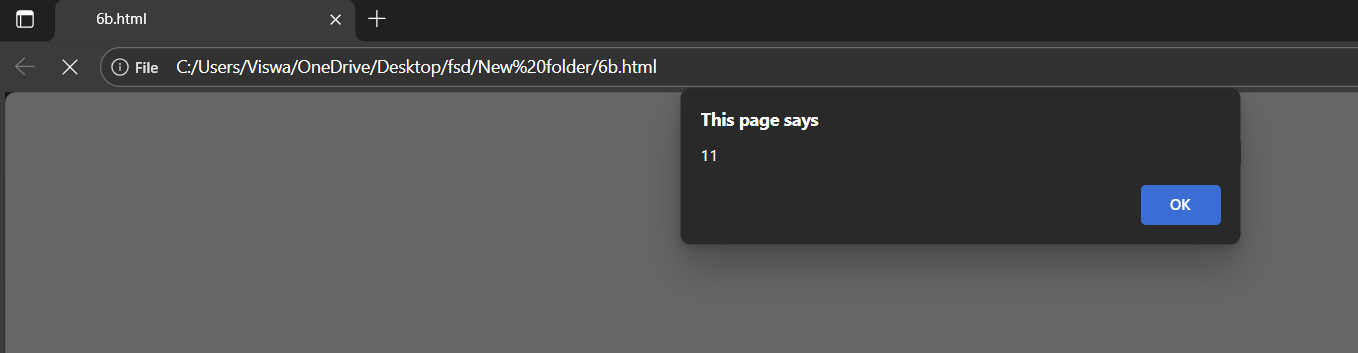
<h2>My First Web Page</h2>

<p>My first paragraph.</p>

<script> alert("Alert"); alert(5 + 6);

window.alert(5 + 6); window.alert("Alert from window");

</script></body></html>



**OUTPUT SCREEN1:**

**OUTPUT SCREEN2:**

**OUTPUT SCREEN3:**

**OUTPUT SCREEN4:**

**OUTPUT SCREEN5:**

## C.Create a web page which uses prompt dialogue box to ask a voter for his name and age. Display the information in table format along with either the voter can vote or not.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Voter Eligibility</title>

<style> body {

font-family: Arial, sans-serif; margin: 40px;

background-color: #f4f4f9;

}

table {

border-collapse: collapse; width: 50%;

margin-top: 20px;

}

th, td {

border: 1px solid #333; padding: 10px;

text-align: center;

}

th {

background-color: #0077b6; color: white;

}

</style>

</head>

<body>

<h2>Voter Eligibility Check</h2>

<div id="result"></div>

<script>

// Prompt for voter details

let name = prompt("Enter your name:"); let age = prompt("Enter your age:"); age = parseInt(age);

// Check eligibility

let eligibility = (age >= 18) ? "Eligible to Vote" : "Not Eligible to Vote";

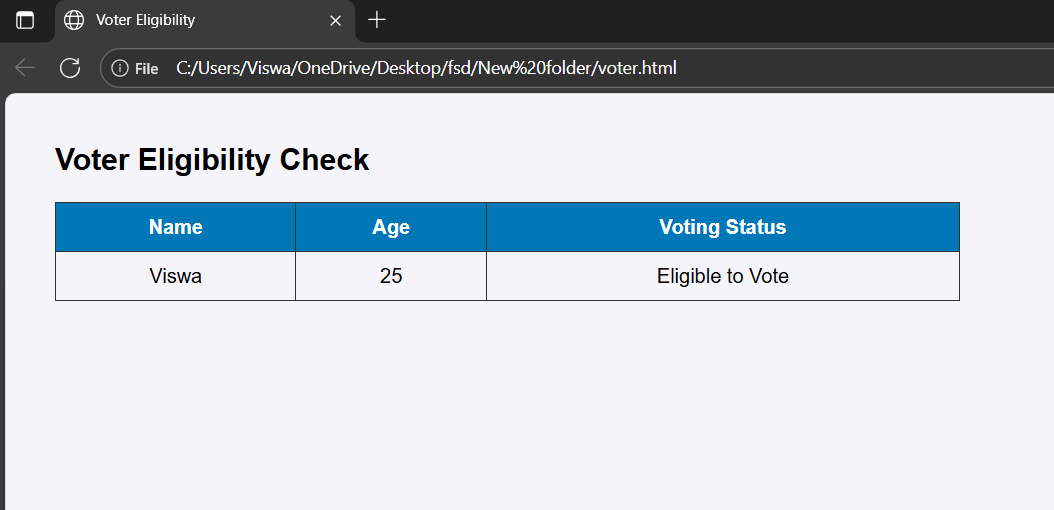
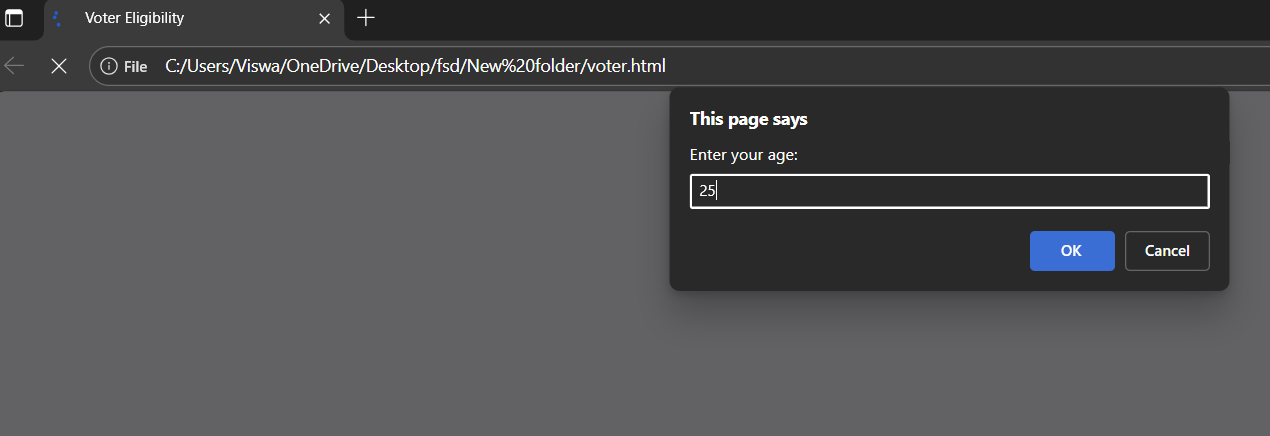
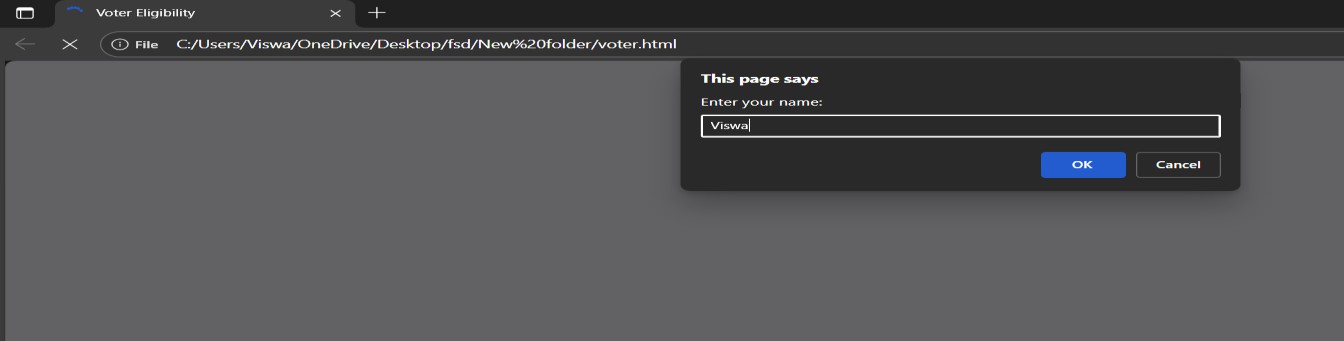
// Create table dynamically let tableHTML = `

<table>

<tr>

<th>Name</th>

<th>Age</th>



<th>Voting Status</th>

</tr>

<tr>

<td>${name}</td>

<td>${age}</td>

<td>${eligibility}</td>

</tr>

</table>

// Display table document.getElementById("result").innerHTML = tableHTML;

</script>

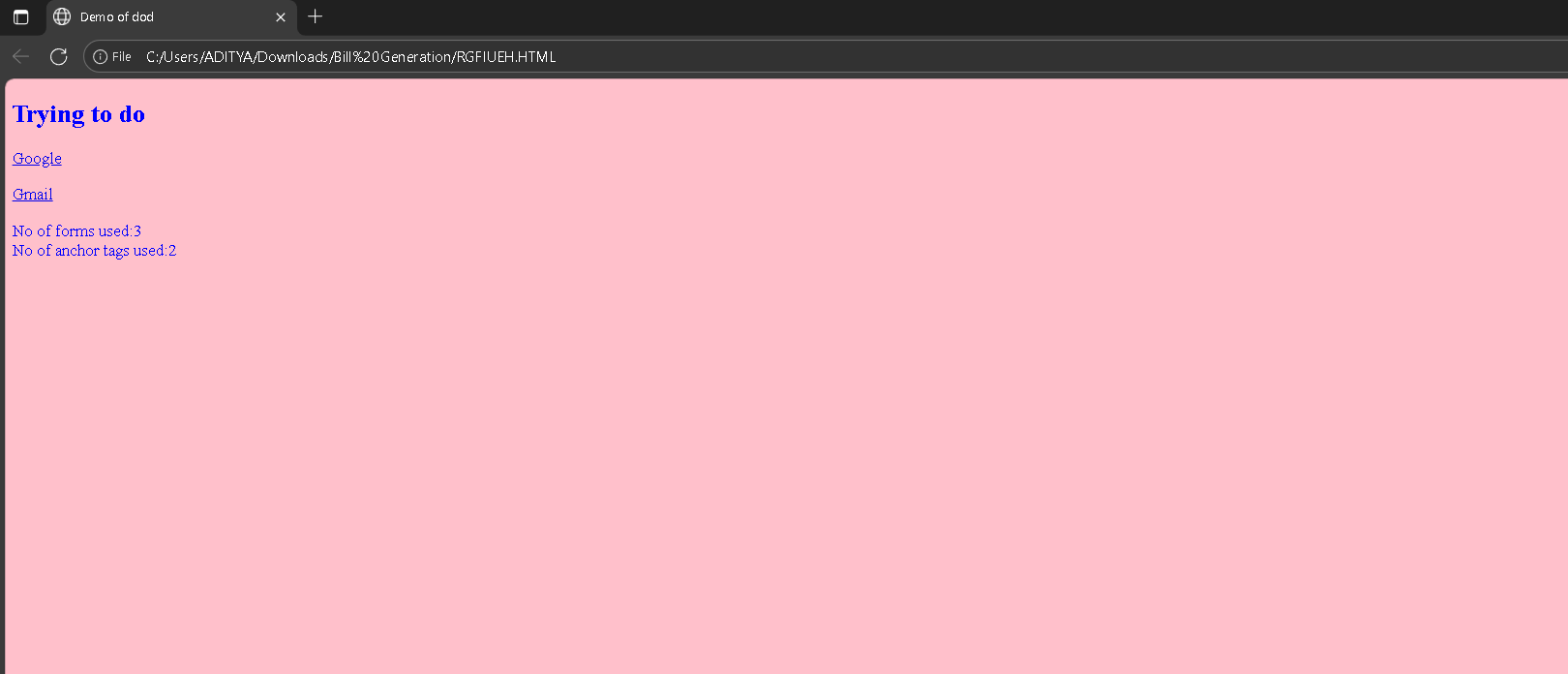
</body>

</html>

**OUTPUT SCREEN1:**

**OUTPUT SCREEN2:**

**OUTPUT SCREEN3:**



# WEEK-7

## Write a program using document object properties and methods

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Demo of dod</title>

</head>

<body>

<h2>Trying to do</h2>

<p><a href="https://[www.Google.com"](http://www.Google.com/)>Google</a></p>

<p><a href=["http://www.gmail.com"](http://www.gmail.com/)>Gmail</a></p>

<form> </form>

<form> </form>

<form> </form>

<script> document.bgColor="pink"; document.fgColor="blue";

document.write("No of forms used:" + document.forms.length); document.write("<br> No of anchor tags used:" + document.links.length);

</script>

</body>

</html>

# OUTPUT:

## Write a program using window object properties and methods.

<!DOCTYPE html>

<html>

<head>

<title>Window Object Example</title>

<style> body {

font-family: Arial, sans-serif; background-color: #f5f5f5; text-align: center;

margin-top: 50px;

}

button { margin: 10px;

padding: 10px 20px; font-size: 16px; border: none;

border-radius: 8px; background-color: #007bff; color: white;

cursor: pointer;

}

button:hover {

background-color: #0056b3;

}

</style>

</head>

<body>

<h1>JavaScript Window Object Example</h1>

<button onclick="showAlert()">Show Alert</button>

<button onclick="getUserName()">Prompt for Name</button>

<button onclick="openNewWindow()">Open New Window</button>

<button onclick="closeNewWindow()">Close New Window</button>

<button onclick="showWindowSize()">Show Window Size</button>

<p id="output"></p>

<script>

let newWin; // variable to hold reference to new window

// 1. Alert box using window.alert() function showAlert() {

window.alert("This is an alert from the window object!");

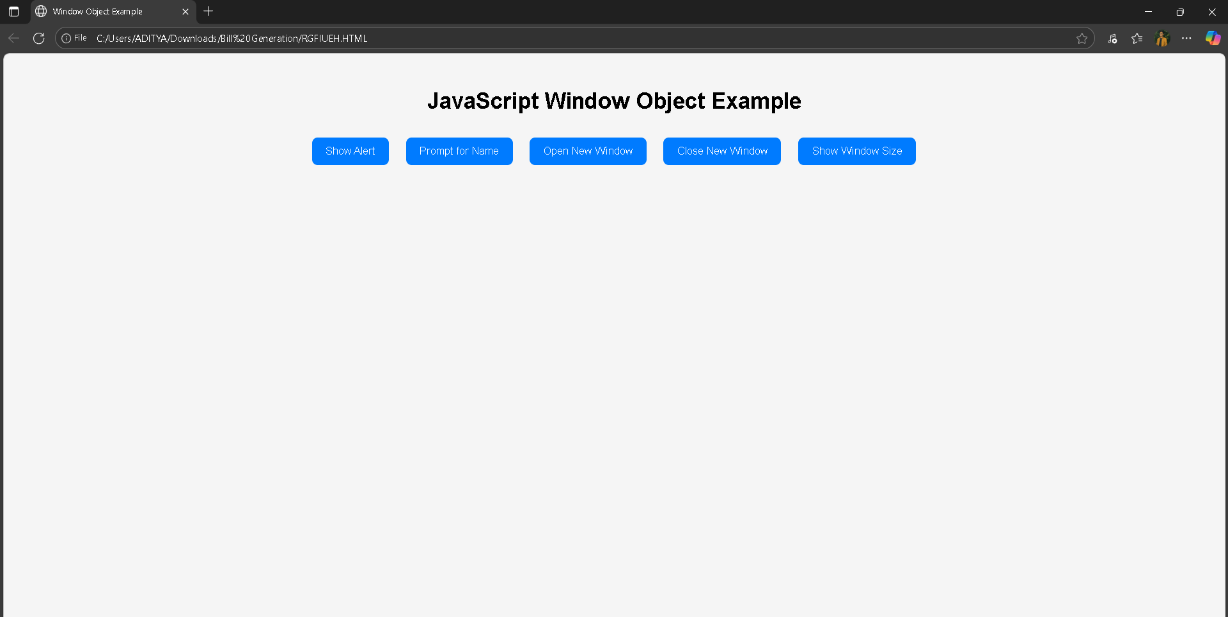
}

// 2. Prompt box using window.prompt() function getUserName() {

let name = window.prompt("Enter your name:"); if (name) {

document.getElementById("output").innerHTML = "Hello, " + name + "!";

} else {



document.getElementById("output").innerHTML = "You didn’t enter a name.";

}

}

// 3. Open a new window using window.open() function openNewWindow() {

newWin = window.open( "https://[www.example.com"](http://www.example.com/), "\_blank", "width=400,height=300"

);

}

// 4. Close the new window using window.close() function closeNewWindow() {

if (newWin && !newWin.closed) { newWin.close();

} else {

alert("No window is open to close!");

}

}

// 5. Display window size using window.innerWidth and window.innerHeight function showWindowSize() {

let width = window.innerWidth; let height = window.innerHeight;

document.getElementById("output").innerHTML = "Window Size: " + width + " x " + height + " pixels.";

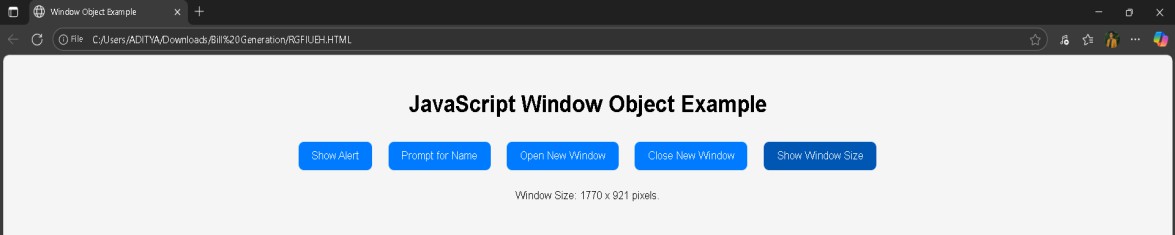
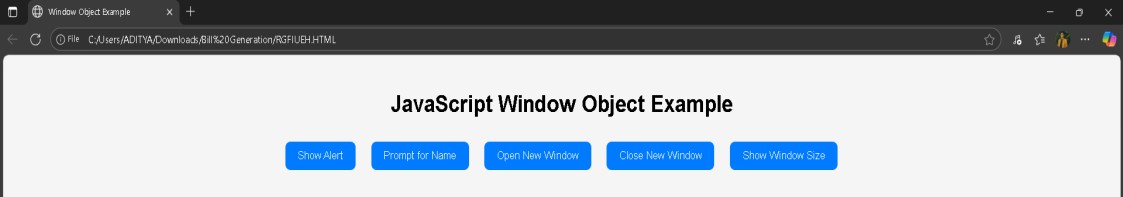
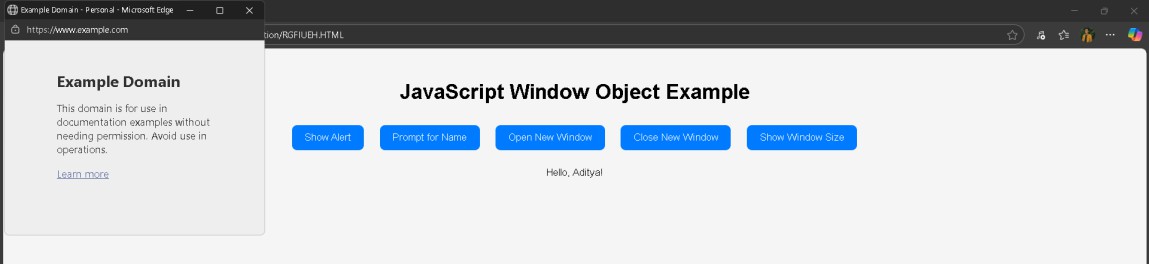
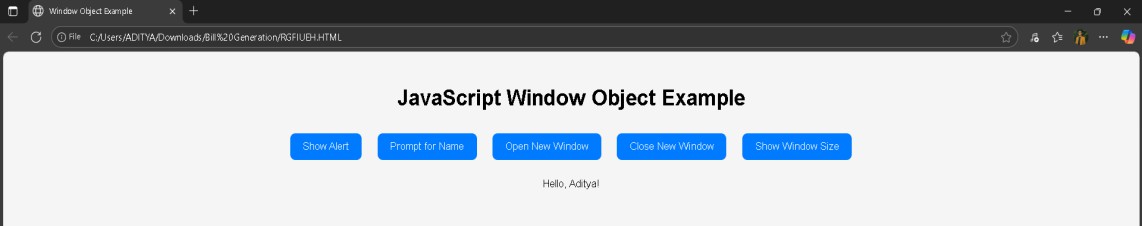
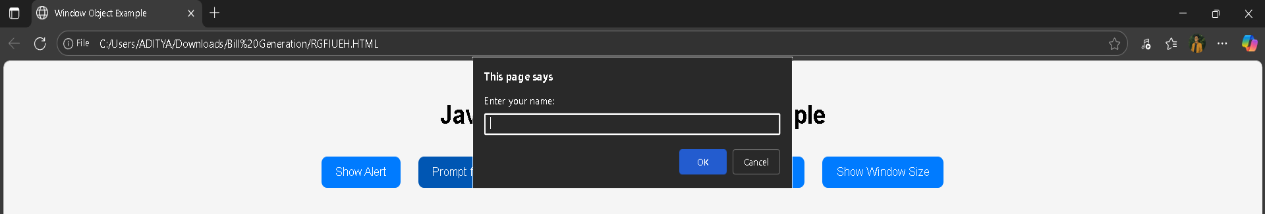
}

</script>

</body>

</html>

# OUTPUT:



**OUTPUT SCREEN1:**

**OUTPUT SCREEN2:**

**OUTPUT SCREEN2:**

**OUTPUT SCREEN3:**

**OUTPUT SCREEN4:**

**OUTPUT SCREEN5:**

## C) Write a program using array object properties and methods.

<!DOCTYPE html>

<html>

<head>

<title>Array Object Example</title>

<style> body {

font-family: Arial, sans-serif; text-align: center; background-color: #f0f8ff; margin-top: 50px;

}

button { margin: 10px;

padding: 10px 20px; font-size: 16px; border-radius: 8px; border: none;

background-color: #28a745; color: white;

cursor: pointer;

}

button:hover {

background-color: #1e7e34;

}

#output {

margin-top: 20px; background-color: #e0f7fa; padding: 20px;

border-radius: 10px; display: inline-block;

text-align: left; min-width: 400px;

}

</style></head>

<body>

<h1>JavaScript Array Object Example</h1>

<button onclick="showArray()">Show Array</button>

<button onclick="addElement()">Push Element</button>

<button onclick="removeElement()">Pop Element</button>

<button onclick="sortArray()">Sort</button>

<button onclick="reverseArray()">Reverse</button>

<button onclick="findIndex()">Find Index</button>

<div id="output"></div>

<script>

// Initialize an array

let fruits = ["Apple", "Banana", "Cherry", "Mango"];

// Display the array function showArray() {

document.getElementById("output").innerHTML = "<b>Fruits Array:</b> " + fruits.join(", ") + "<br><b>Length:</b> " + fruits.length;

}

// Add a new element at the end (push) function addElement() {

let fruit = prompt("Enter a fruit to add:"); if (fruit) {

fruits.push(fruit); showArray();

}

}

// Remove the last element (pop) function removeElement() {

let removed = fruits.pop(); alert("Removed: " + removed); showArray();

}

// Sort the array alphabetically function sortArray() { fruits.sort();

showArray();

}

// Reverse the array function reverseArray() { fruits.reverse(); showArray();

}

// Find index of a specific element function findIndex() {

let item = prompt("Enter a fruit to find its index:"); let index = fruits.indexOf(item);

if (index !== -1) {

alert(item + " found at position " + index);

} else {

alert(item + " not found in the array.");

}

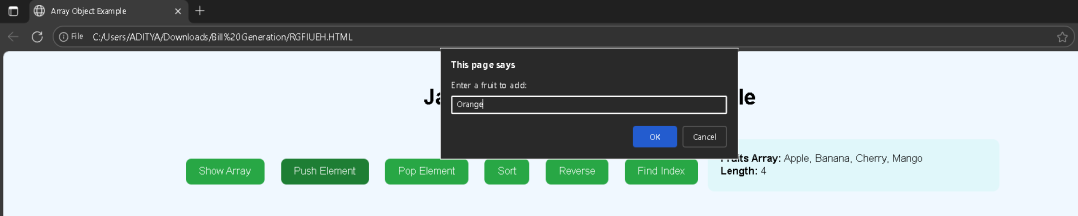
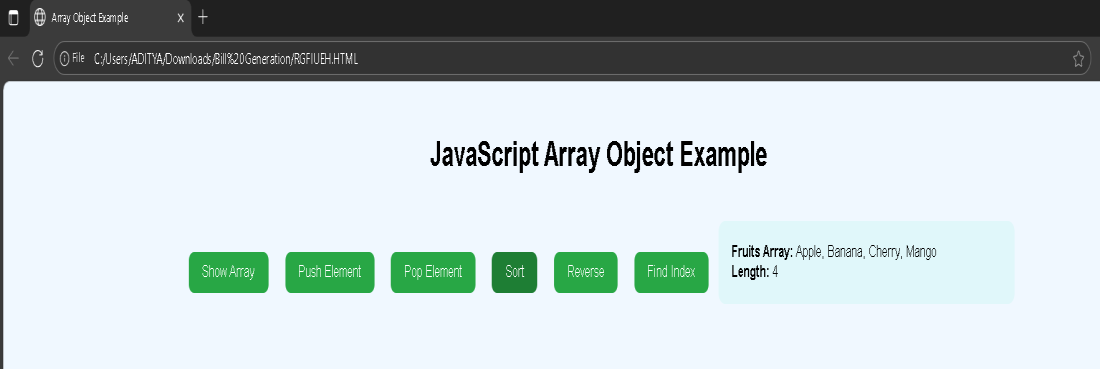
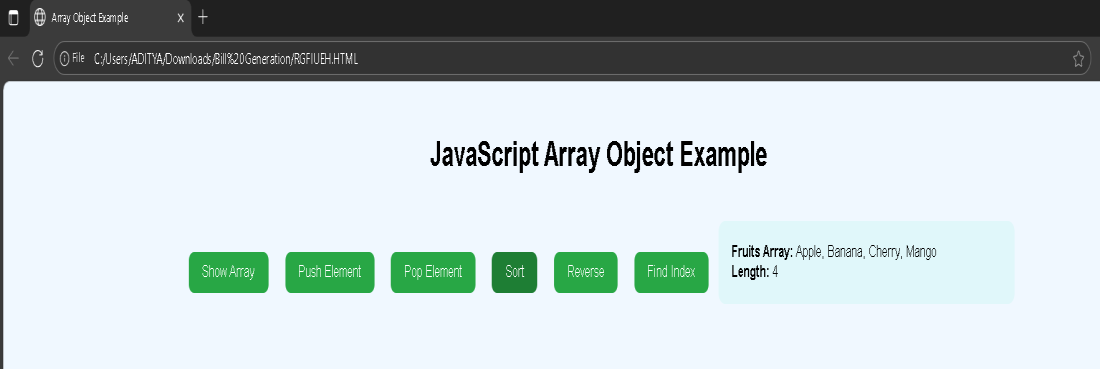
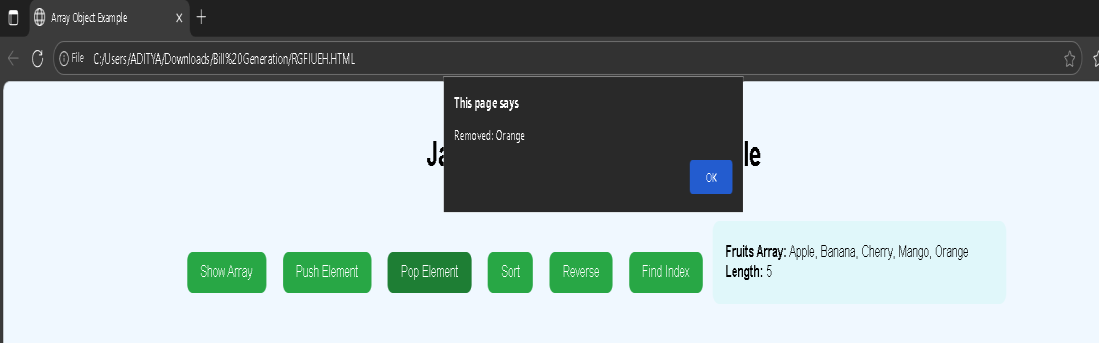
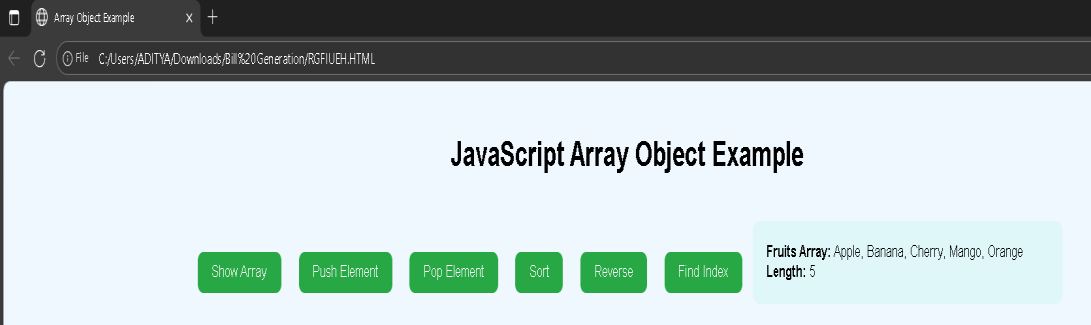
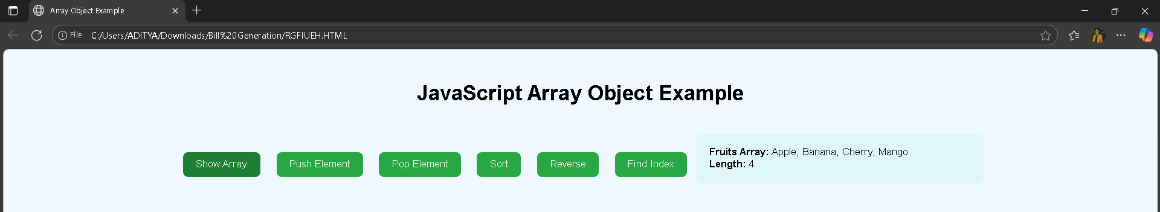
}

// Display initial array on page load showArray();

</script>

</body></html>

**OUTPUT SCREEN 1:**

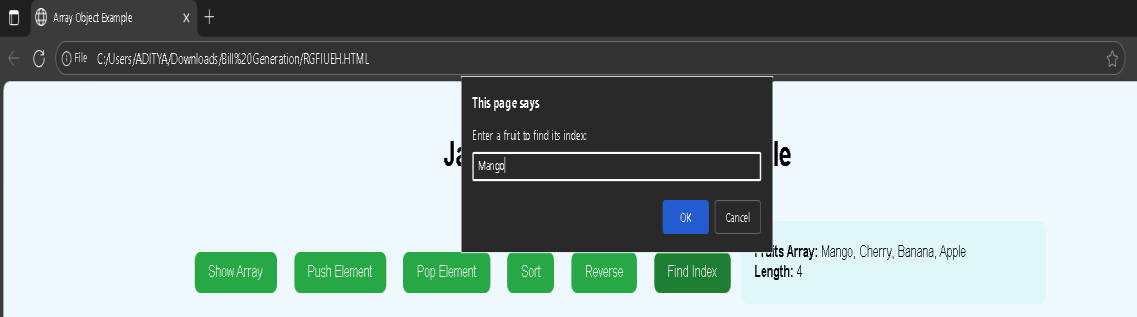


**OUTPUT SCREEN 2:**

**OUTPUT SCREEN 3:**

**OUTPUT SCREEN 4:**

**OUTPUT SCREEN 5:**



**OUTPUT SCREEN 6:**

## Write a program using math object properties and methods.

<!DOCTYPE html>

<html>

<head>

<title>Math Object Example</title>

<style> body {

font-family: Arial, sans-serif; text-align: center; background-color: #f9f9f9; margin-top: 40px;

}

button {

margin: 10px; padding: 10px 20px; font-size: 16px; border-radius: 8px; border: none;

background-color: #007bff; color: white;

cursor: pointer;

}

button:hover {

background-color: #0056b3;

}

#output {

margin-top: 25px; background-color: #e3f2fd; padding: 20px;

border-radius: 10px; display: inline-block; text-align: left;

min-width: 400px;

}

</style>

</head>

<body>

<h1>JavaScript Math Object Example</h1>

<button onclick="showConstants()">Show Constants</button>

<button onclick="calculateSquareRoot()">Square Root</button>

<button onclick="calculatePower()">Power</button>

<button onclick="roundNumbers()">Rounding Methods</button>

<button onclick="findMinMax()">Find Min/Max</button>

<button onclick="generateRandom()">Random Number</button>

<div id="output"></div>

<script>

// Display Math constants function showConstants() {

document.getElementById("output").innerHTML = `

<b>Math Constants:</b><br> Math.PI = ${Math.PI}<br> Math.E = ${Math.E}<br>

Math.SQRT2 = ${Math.SQRT2}<br> `;

}

// Calculate square root

function calculateSquareRoot() {

let num = prompt("Enter a number:"); if (num !== null) {

document.getElementById("output").innerHTML =

`Square root of ${num} = ${Math.sqrt(num)}`;

}}

// Calculate power

function calculatePower() {

let base = prompt("Enter base number:"); let exp = prompt("Enter exponent:");

if (base !== null && exp !== null) { document.getElementById("output").innerHTML =

`${base} raised to power ${exp} = ${Math.pow(base, exp)}`;

}}

// Demonstrate rounding methods function roundNumbers() {

let num = 12.67; document.getElementById("output").innerHTML = `

<b>Number:</b> ${num}<br> Math.round(${num}) = ${Math.round(num)}<br> Math.floor(${num}) = ${Math.floor(num)}<br> Math.ceil(${num}) = ${Math.ceil(num)}<br>`;

}

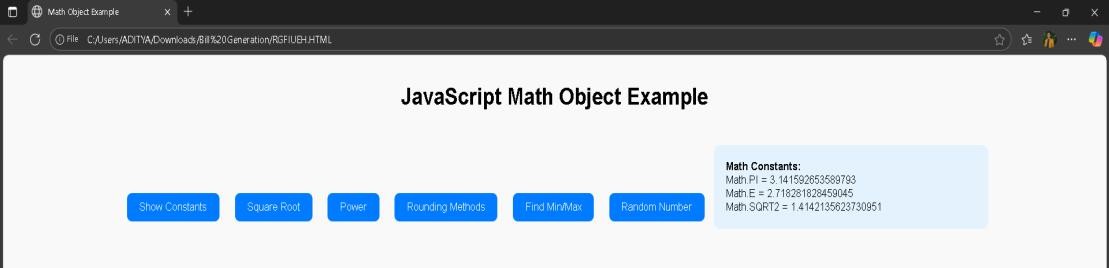
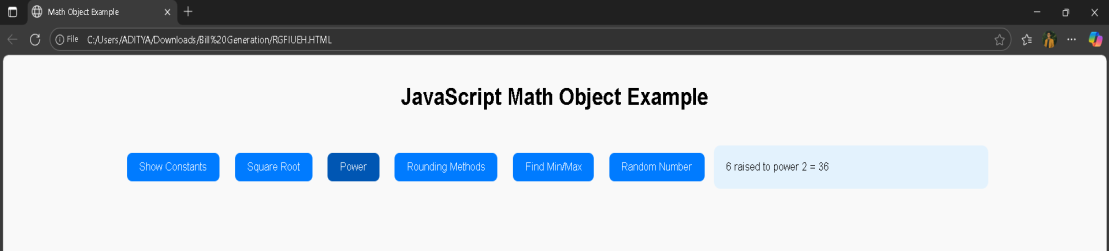
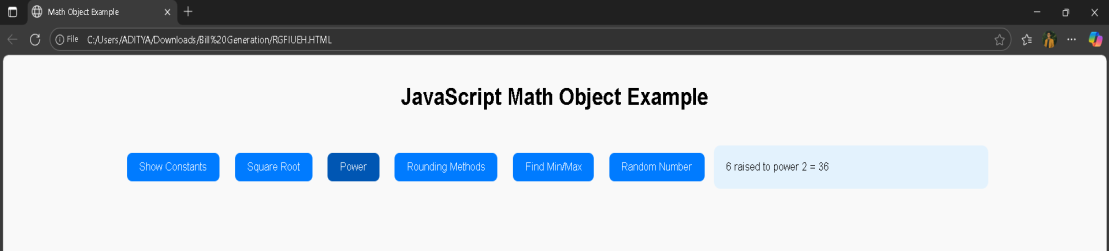
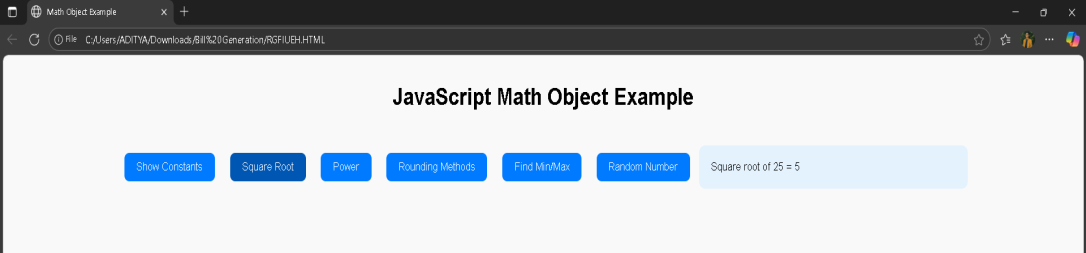
function findMinMax() {

let values = [5, 10, -3, 42, 8]; document.getElementById("output").innerHTML = `

<b>Array:</b> [${values.join(", ")}]<br> Math.max(...) = ${Math.max(...values)}<br>

Math.min(...) = ${Math.min(...values)}<br>`;

}



function generateRandom() {

let randomNum = Math.random(); // between 0 and 1

let randomInt = Math.floor(Math.random() \* 100) + 1; // between 1 and 100 document.getElementById("output").innerHTML = `

Math.random() = ${randomNum}<br> Random integer (1–100) = ${randomInt}<br>`;

}

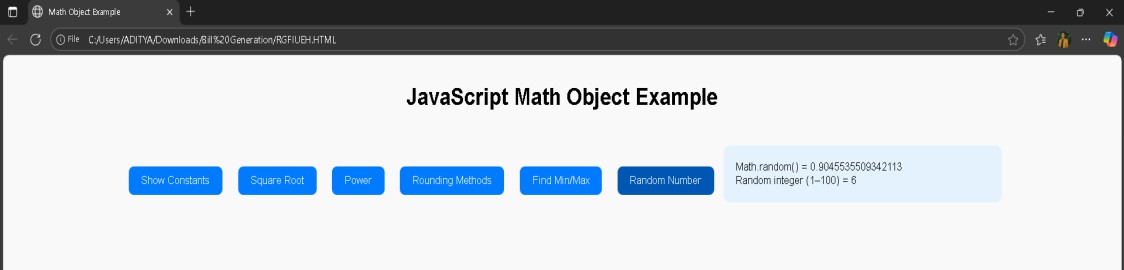
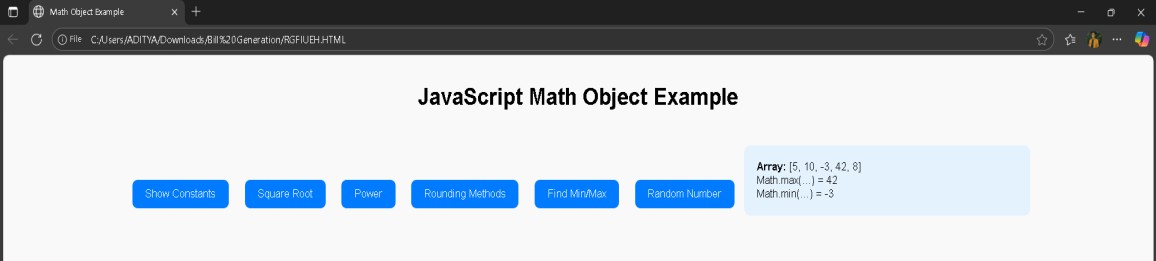
</script></body></html>

**OUTPUT SCREEN1:**

**OUTPUT SCREEN2:**

**OUTPUT SCREEN3:**

**OUTPUT SCREEN4:**



**OUTPUT SCREEN5:**

**OUTPUT SCREEN6:**

## Write a program using string object properties and methods.

<!DOCTYPE html>

<html>

<head>

<title>String Object Example</title>

<style> body {

font-family: Arial, sans-serif; text-align: center; background-color: #f9f9f9; margin-top: 40px;

}

button { margin: 8px;

padding: 10px 20px; font-size: 16px; border-radius: 8px; border: none;

background-color: #007bff; color: white;

cursor: pointer;

}

button:hover {

background-color: #0056b3;

}

#output {

margin-top: 25px;

background-color: #e3f2fd; padding: 20px;

border-radius: 10px; display: inline-block; text-align: left;

min-width: 400px;

}

</style>

</head>

<body>

<h1>JavaScript String Object Example</h1>

<button onclick="showLength()">Length</button>

<button onclick="toUpper()">Uppercase</button>

<button onclick="toLower()">Lowercase</button>

<button onclick="findChar()">Character at Position</button>

<button onclick="findIndex()">Index of Word</button>

<button onclick="extractSubstring()">Substring</button>

<button onclick="replaceWord()">Replace Word</button>

<button onclick="splitString()">Split String</button>

<button onclick="trimSpaces()">Trim Spaces</button>

<div id="output"></div>

<script>

// Sample string

let text = " JavaScript is Fun to Learn! ";

// Display string length function showLength() {

document.getElementById("output").innerHTML =

`<b>String:</b> "${text}"<br> Length: ${text.length}`;

}

// Convert to uppercase function toUpper() {

document.getElementById("output").innerHTML =

`Uppercase: ${text.toUpperCase()}`;

}

// Convert to lowercase function toLower() {

document.getElementById("output").innerHTML =

`Lowercase: ${text.toLowerCase()}`;

}

// Find character at specific position function findChar() {

let pos = prompt("Enter position number (0-based index):"); if (pos !== null && pos < text.length) { document.getElementById("output").innerHTML =

`Character at position ${pos}: "${text.charAt(pos)}"`;

} else {

alert("Invalid position!");

}

}

// Find index of a word function findIndex() {

let word = prompt("Enter a word to find:"); let index = text.indexOf(word);

if (index !== -1) { document.getElementById("output").innerHTML =

`"${word}" found at position ${index}`;

} else { document.getElementById("output").innerHTML =

`"${word}" not found in the string.`;

}

}

// Extract substring

function extractSubstring() { document.getElementById("output").innerHTML =

`Substring (0–10): "${text.substring(0, 10)}"`;

}

// Replace a word function replaceWord() {

let newText = text.replace("Fun", "Awesome"); document.getElementById("output").innerHTML =

`After replace: ${newText}`;

}

// Split the string into words function splitString() {

let words = text.trim().split(" "); document.getElementById("output").innerHTML =

`<b>Words:</b> ${words.join(", ")}`;

}

// Trim spaces

function trimSpaces() { document.getElementById("output").innerHTML =

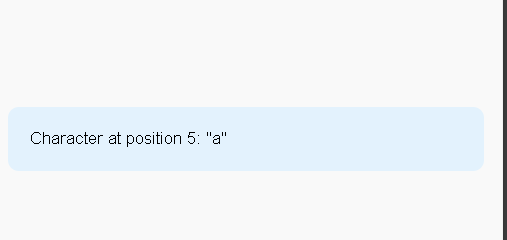
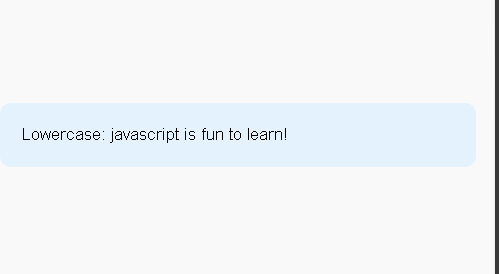
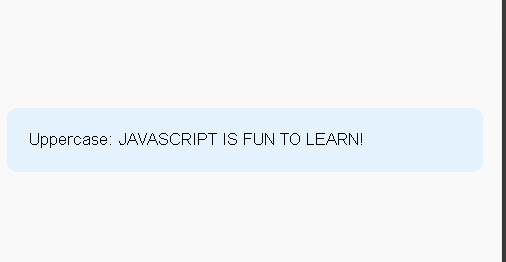
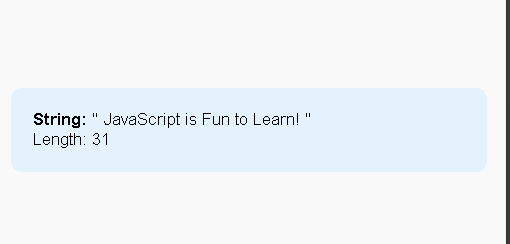
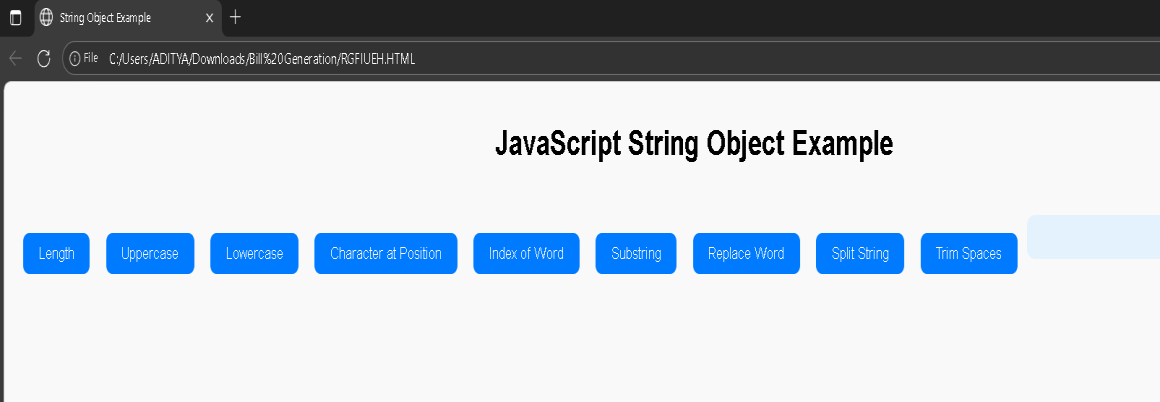
`Original: "${text}"<br>Trimmed: "${text.trim()}"`;

}

</script>

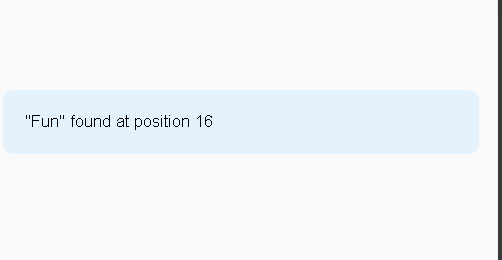
</body>

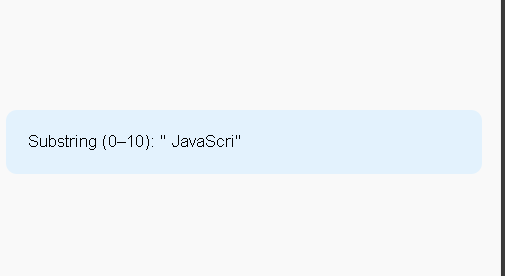
</html>

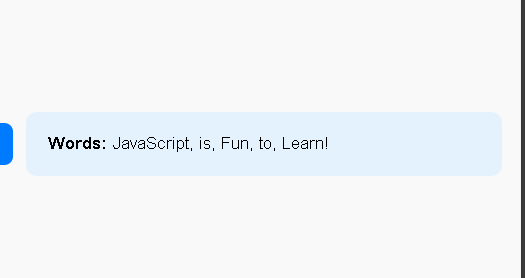


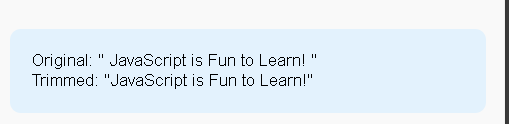
# OUTPUT:

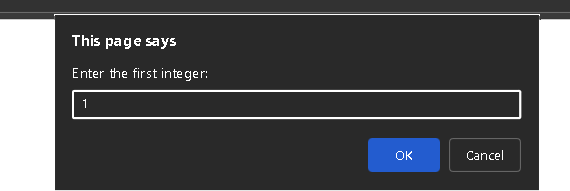




****

****

****



**WEEK-8**

## Write a program which asks the user to enter three integers, obtains the numbers from the user and outputs HTML text that displays the larger number followed by the words “LARGER NUMBER” in an information message dialog. If the numbers are equal, output HTML text as “EQUAL NUMBERS”

<!DOCTYPE html>

<html>

<head>

<title>Largest Number Finder</title>

<script>

function findLargest() {

// Ask the user to enter three integers

let num1 = parseInt(prompt("Enter the first integer:")); let num2 = parseInt(prompt("Enter the second integer:")); let num3 = parseInt(prompt("Enter the third integer:")); let message = "";

// Compare the numbers

if (num1 === num2 && num2 === num3) { message = "EQUAL NUMBERS";

} else {

let largest = Math.max(num1, num2, num3); message = largest + " LARGER NUMBER";

}

// Display result in an information message dialog alert(message);

// Also display the result on the webpage document.getElementById("result").innerHTML = "<h2>" + message + "</h2>";

}

</script>

</head>

<body onload="findLargest()">

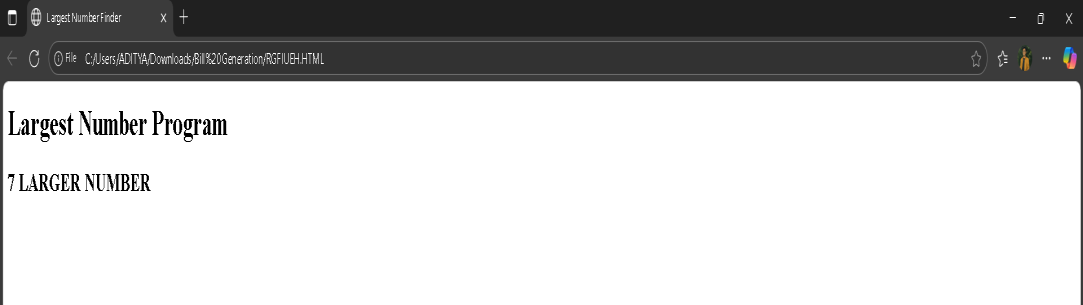
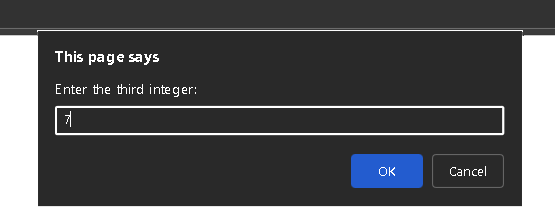
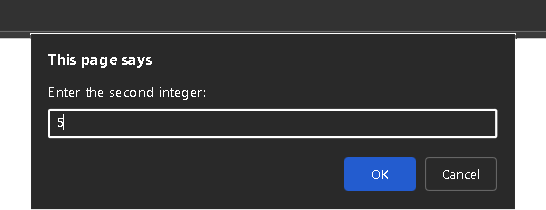
<h1>Largest Number Program</h1>

<p id="result"></p>

</body>

</html>

# OUTPUT:



## Write a program to display weekdays using switch case.

<!DOCTYPE html>

<html>

<head>

<title>Display Weekdays Using Switch Case</title>

<script>

function showWeekday() {

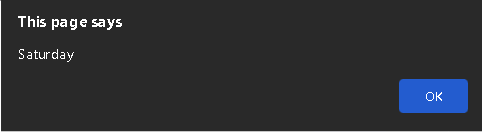
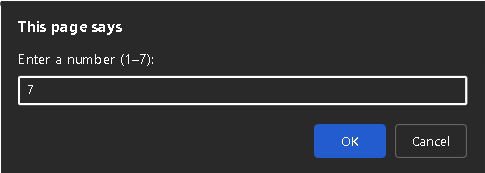
// Ask the user to enter a number between 1 and 7

let dayNumber = parseInt(prompt("Enter a number (1–7):")); let dayName;

// Use switch case to determine the weekday switch (dayNumber) {

case 1:

dayName = "Sunday"; break;



case 2:

dayName = "Monday"; break;

case 3:

dayName = "Tuesday"; break;

case 4:

dayName = "Wednesday"; break;

case 5:

dayName = "Thursday"; break;

case 6:

dayName = "Friday"; break;

case 7:

dayName = "Saturday"; break;

default:

dayName = "Invalid input! Please enter a number between 1 and 7.";

}

// Display result in an alert box and on the web page alert(dayName);

document.getElementById("result").innerHTML = "<h2>" + dayName + "</h2>";

}

</script>

</head>

<body onload="showWeekday()">

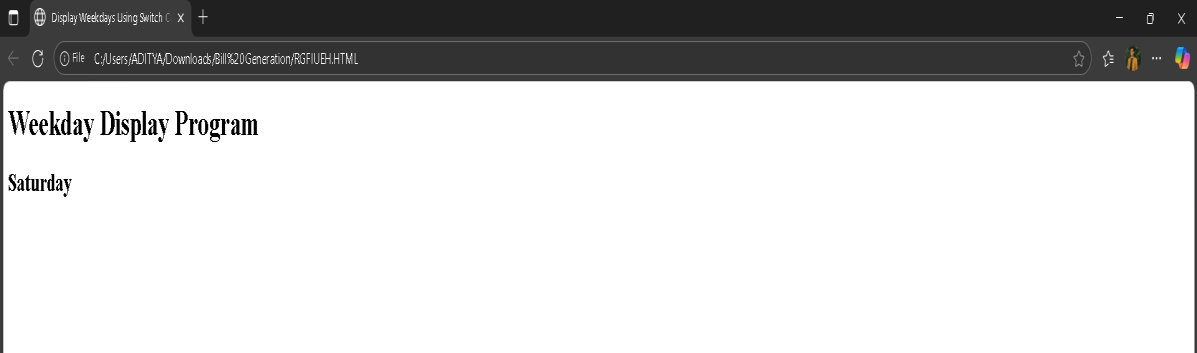
<h1>Weekday Display Program</h1>

<p id="result"></p>

</body>

</html>

# OUTPUT:



## Write a program to print1to10 numbers using for, while and do-while loops.

<!DOCTYPE html>

<html>

<head>

<title>Print 1 to 10 using Loops</title>

<script>

function printNumbers() {

let result = "<h2>Using for loop:</h2>";

// Using for loop

for (let i = 1; i <= 10; i++) { result += i + " ";

}

result = "<h2>Using while loop:</h2>"; let j = 1;

// Using while loop while (j <= 10) { result += j + " "; j++;

}

result += "<h2>Using do-while loop:</h2>"; let k = 1;

// Using do-while loop do {

result += k + " "; k++;

} while (k <= 10);

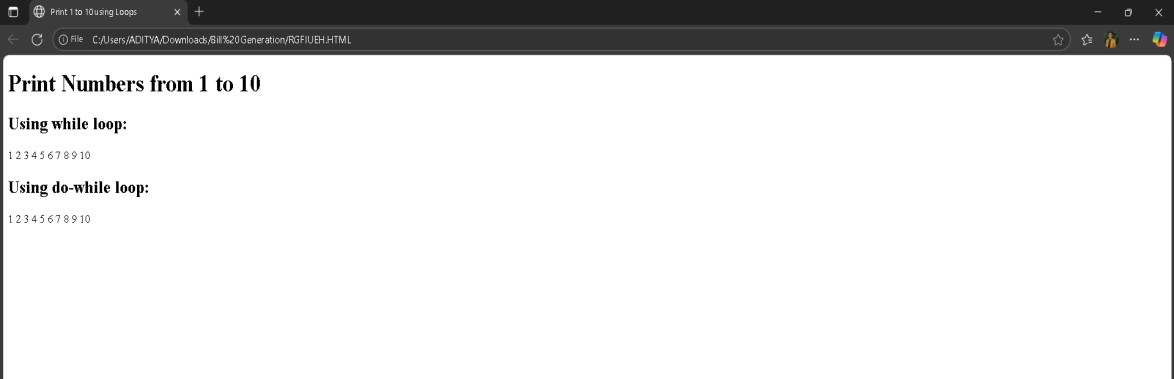
// Display the result on the web page document.getElementById("output").innerHTML = result;

}

</script>

</head>

<body onload="printNumbers()">



<h1>Print Numbers from 1 to 10</h1>

<div id="output"></div>

</body>

</html>

# OUTPUT:

## Develop a program to determine whether a given number is an ‘ARMSTRONGNUMBER’ or not. [Ex: 153 is an Armstrong number, since sum of the cube of the digits is equal to the number i.e.,13 + 53+33 = 153]

<!DOCTYPE html>

<html>

<head>

<title>Armstrong Number Check</title>

</head>

<body>

<h2>Check Armstrong Number</h2>

<script>

// Get input from user

let num = parseInt(prompt("Enter a number:"));

// Store the original number let temp = num;

let sum = 0;

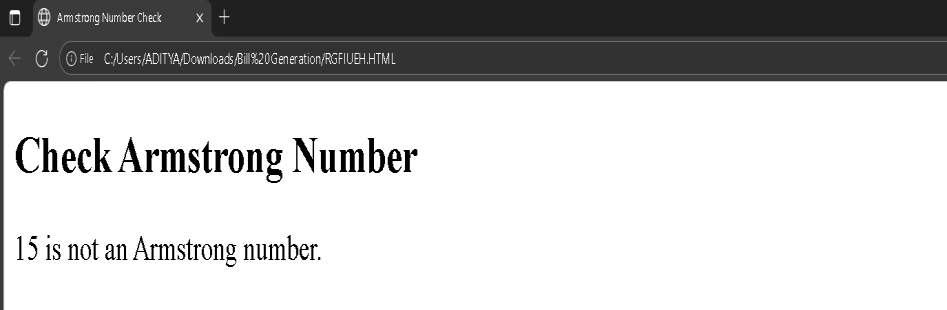
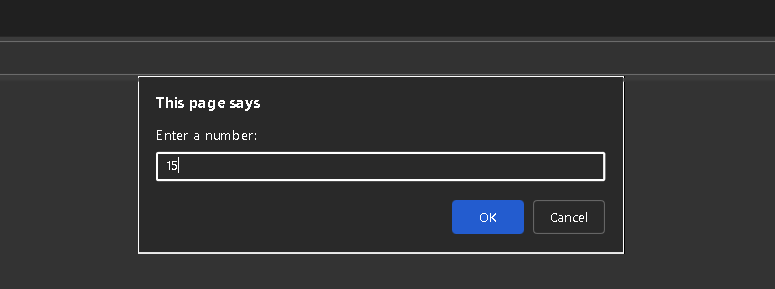
// Convert number to string to count digits let n = num.toString().length;

// Calculate the sum of each digit raised to the power of n while (temp > 0) {

let digit = temp % 10;

sum += Math.pow(digit, n); temp = Math.floor(temp / 10);

}



// Check if Armstrong or not if (sum === num)

document.write(num + " is an Armstrong number."); else

document.write(num + " is not an Armstrong number.");

</script>

</body>

</html>

# OUTPUT:

## Write a program to display the denomination of the amount deposited in the bank in terms of 100’s, 50’s,20’s,10’s,5’s,2’s&1’s.(Ex: If deposited amount is Rs.163, the output shouldbe1 100’s,1- 50’s, 1- 10’s, 1-2’s & 1-1’s)

<!DOCTYPE html>

<html>

<head>

<title>Bank Denomination Program</title>

</head>

<body>

<h2>Bank Denomination Calculator</h2>

<script>

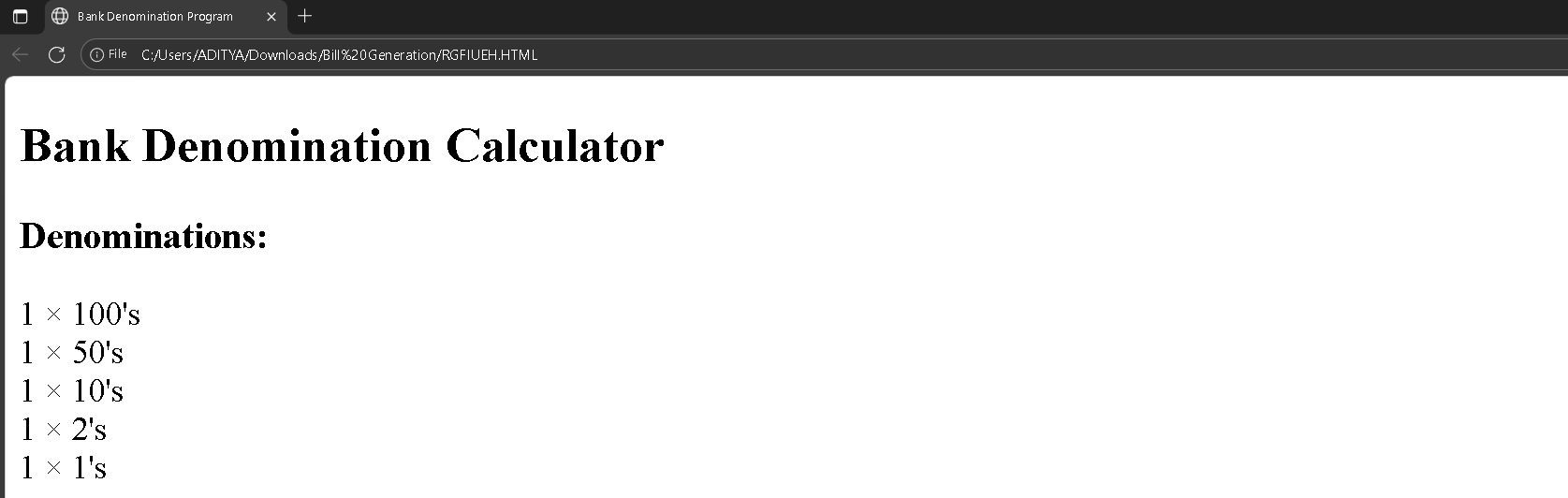
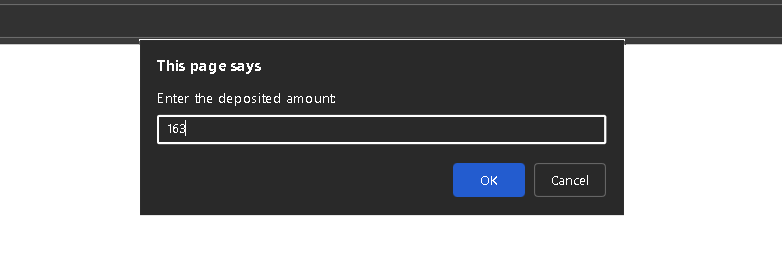
// Get input from user

let amount = parseInt(prompt("Enter the deposited amount:")); let hundreds = Math.floor(amount / 100);

amount = amount % 100;

let fifties = Math.floor(amount / 50); amount = amount % 50;

let twenties = Math.floor(amount / 20); amount = amount % 20;



let tens = Math.floor(amount / 10); amount = amount % 10;

let fives = Math.floor(amount / 5); amount = amount % 5;

let twos = Math.floor(amount / 2); amount = amount % 2;

let ones = amount; document.write("<h3>Denominations:</h3>");

if (hundreds > 0) document.write(hundreds + " × 100's<br>"); if (fifties > 0) document.write(fifties + " × 50's<br>");

if (twenties > 0) document.write(twenties + " × 20's<br>"); if (tens > 0) document.write(tens + " × 10's<br>");

if (fives > 0) document.write(fives + " × 5's<br>"); if (twos > 0) document.write(twos + " × 2's<br>"); if (ones > 0) document.write(ones + " × 1's<br>");

</script>

</body>

</html>

# OUTPUT:

**WEEK-9**

1. **Design a appropriate function should be called to display**

* **Factorial of that number**
* **Fibonacci series up to that number**
* **Prime numbers up to that number**
* **Is it palindrome or not**

<!DOCTYPE html>

<html>

<head>

<title>Number Functions Program</title>

</head>

<body>

<h2>Number Function Demonstration</h2>

<script>

// Function to calculate factorial function factorial(num) {

let fact = 1;

for (let i = 1; i <= num; i++) { fact \*= i;

}

return fact;

}

// Function to generate Fibonacci series up to n terms function fibonacci(num) {

let n1 = 0, n2 = 1, nextTerm; let series = [];

for (let i = 1; i <= num; i++) { series.push(n1);

nextTerm = n1 + n2; n1 = n2;

n2 = nextTerm;

}

return series;

}

// Function to find prime numbers up to num function primeNumbers(num) {

let primes = [];

for (let i = 2; i <= num; i++) { let isPrime = true;

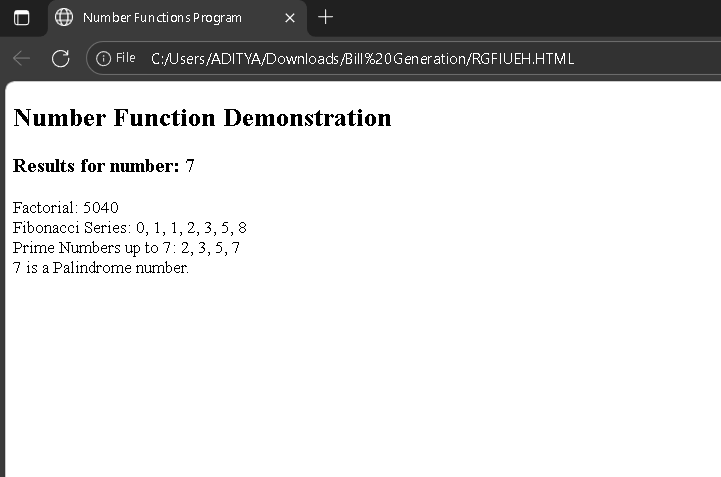
for (let j = 2; j <= Math.sqrt(i); j++) { if (i % j === 0) {

isPrime = false; break;

}

}

if (isPrime) primes.push(i);



}

return primes;

}

// Function to check if number is palindrome function isPalindrome(num) {

let str = num.toString();

let reversed = str.split('').reverse().join(''); return str === reversed;

}

// --- Main Program ---

let number = parseInt(prompt("Enter a number:"));

document.write("<h3>Results for number: " + number + "</h3>"); document.write("Factorial: " + factorial(number) + "<br>"); document.write("Fibonacci Series: " + fibonacci(number).join(", ") + "<br>"); document.write("Prime Numbers up to " + number + ": " +

primeNumbers(number).join(", ") + "<br>");

if (isPalindrome(number))

document.write(number + " is a Palindrome number.<br>"); else

document.write(number + " is not a Palindrome number.<br>");

</script>

</body>

</html>

# OUTPUT:

1. **Design a HTML having a textbox and four buttons named Factorial, Fibonacci, Prime, and Palindrome. When a button is pressed an appropriate function should be called to display**

* **Factorial of that number**
* **Fibonacci series up to that number**
* **Prime numbers up to that number**
* **Is it palindrome or not**

<!DOCTYPE html>

<html>

<head>

<title>Number Function Operations</title>

<style> body {

font-family: Arial, sans-serif; background-color: #f4f7fa; text-align: center;

margin-top: 50px;

}

input { padding: 10px;

font-size: 16px; width: 200px; border-radius: 5px;

border: 1px solid #ccc;

}

button {

padding: 10px 15px; margin: 10px;

font-size: 16px; cursor: pointer; border: none; border-radius: 5px;

background-color: #0078D7; color: white;

}

button:hover {

background-color: #005fa3;} #result {

margin-top: 30px; font-size: 18px;

background-color: #fff; display: inline-block; padding: 15px 25px; border-radius: 10px;

box-shadow: 0 0 10px rgba(0,0,0,0.1); min-width: 300px;

}

</style>

</head>

<body>

<h2>Number Function Operations</h2>

<input type="number" id="num" placeholder="Enter a number">

<br><br>

<button onclick="showFactorial()">Factorial</button>

<button onclick="showFibonacci()">Fibonacci</button>

<button onclick="showPrimes()">Prime</button>

<button onclick="showPalindrome()">Palindrome</button>

<div id="result"></div>

<script>

// Function for Factorial function showFactorial() {

let num = parseInt(document.getElementById("num").value); let fact = 1;

for (let i = 1; i <= num; i++) { fact \*= i;

}

document.getElementById("result").innerHTML = "Factorial of " + num + " is: <b>" + fact + "</b>";

}

// Function for Fibonacci function showFibonacci() {

let num = parseInt(document.getElementById("num").value); let n1 = 0, n2 = 1, nextTerm;

let series = [];

for (let i = 1; i <= num; i++) { series.push(n1);

nextTerm = n1 + n2; n1 = n2;

n2 = nextTerm;

}

document.getElementById("result").innerHTML =

"Fibonacci series up to " + num + ": <b>" + series.join(", ") + "</b>";

}

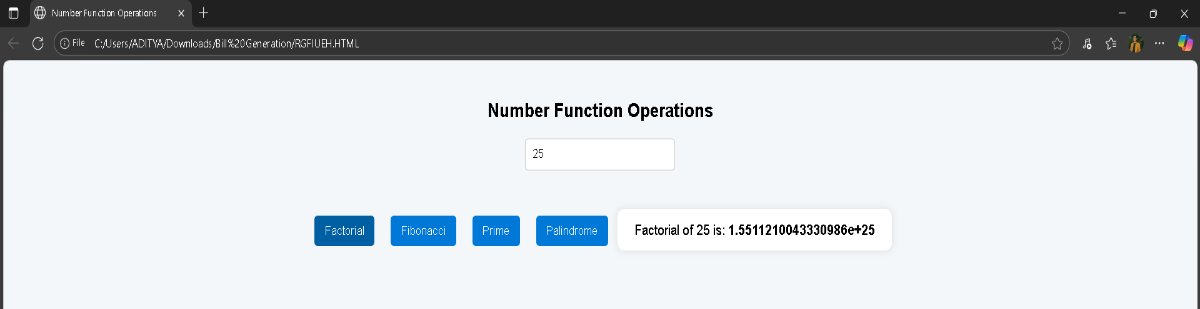
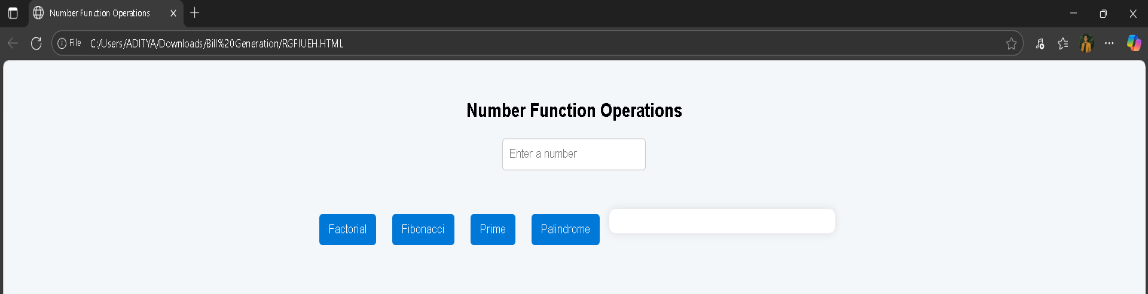
// Function for Prime numbers function showPrimes() {

let num = parseInt(document.getElementById("num").value); let primes = [];

for (let i = 2; i <= num; i++) { let isPrime = true;

for (let j = 2; j <= Math.sqrt(i); j++) {

if (i % j === 0) {



isPrime = false; break;

}

}

if (isPrime) primes.push(i);

}

document.getElementById("result").innerHTML =

"Prime numbers up to " + num + ": <b>" + primes.join(", ") + "</b>";

}

// Function for Palindrome check function showPalindrome() {

let num = document.getElementById("num").value; let reversed = num.split('').reverse().join('');

if (num === reversed) document.getElementById("result").innerHTML = num + " is a <b>Palindrome</b> number.";

else

document.getElementById("result").innerHTML = num + " is <b>not</b> a Palindrome number.";

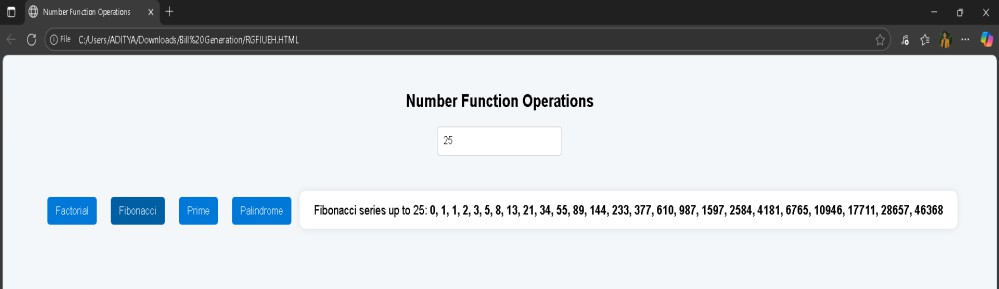
}

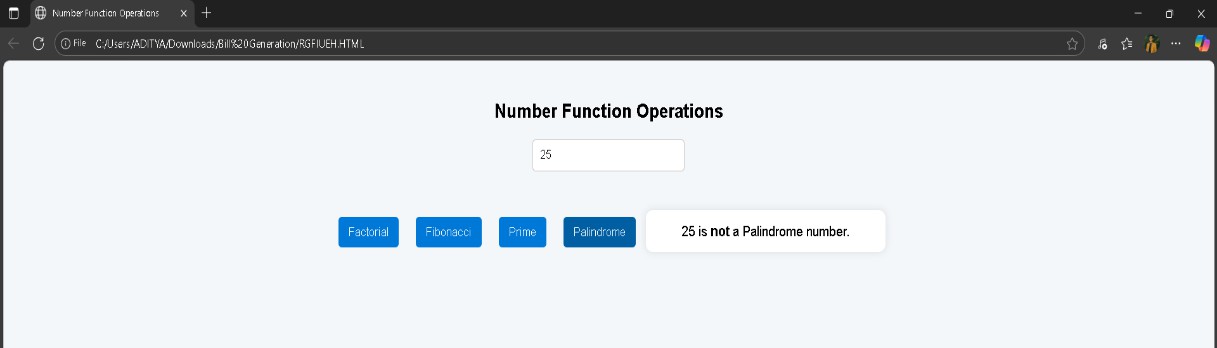
</script>

</body>

</html>

# OUTPUT:



****

## Write a program to validate the following fields in a registration page ii. Name (start with alphabet and followed by alphanumeric and the length should not be less than 6 characters) ii. Mobile (onlynumbersandlength10 digits) iii. E-mail (should contain format like [xxxxxxx@xxxxxx.xxx](mailto:xxxxxxx@xxxxxx.xxx))

<!DOCTYPE html>

<html>

<head>

<title>Registration Form Validation</title>

<style> body {

font-family: Arial, sans-serif; background-color: #f3f6fa; margin-top: 40px;

text-align: center;

}

form {

display: inline-block; background: #fff; padding: 20px 40px; border-radius: 10px;

box-shadow: 0 0 10px rgba(0,0,0,0.1);

}

input {

display: block; margin: 10px auto; padding: 8px; width: 250px;

border: 1px solid #ccc; border-radius: 5px; font-size: 16px;

}

button {

background-color: #0078D7; color: white;

border: none; padding: 10px 20px; font-size: 16px; border-radius: 5px; cursor: pointer;

}

button:hover {

background-color: #005fa3;

}

#result {

margin-top: 20px; font-weight: bold; color: darkred;

}

</style>

</head>

<body>

<h2>Registration Form Validation</h2>

<form onsubmit="return validateForm()">

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

<input type="text" id="mobile" placeholder="Enter Mobile Number">

<input type="text" id="email" placeholder="Enter Email Address">

<button type="submit">Register</button>

</form>

<div id="result"></div>

<script>

function validateForm() {

let name = document.getElementById("name").value.trim(); let mobile = document.getElementById("mobile").value.trim(); let email = document.getElementById("email").value.trim(); let result = document.getElementById("result");

// Name validation: starts with a letter, followed by alphanumeric, min length 6

let namePattern = /^[A-Za-z][A-Za-z0-9]{5,}$/;

// Mobile validation: only numbers, exactly 10 digits let mobilePattern = /^[0-9]{10}$/;

// Email validation: must follow [xxxxx@xxxxx.xxx](mailto:xxxxx@xxxxx.xxx) pattern

let emailPattern = /^[a-zA-Z0-9.\_%+-]+@[a-zA-Z0-9.-]+\.[a-z]{2,}$/;

if (!namePattern.test(name)) { result.style.color = "red";

result.innerHTML = "+ Invalid Name: Must start with a letter, contain only letters/numbers, and be at least 6 characters long.";

return false;

}

else if (!mobilePattern.test(mobile)) { result.style.color = "red";

result.innerHTML = "+ Invalid Mobile Number: Must contain exactly 10 digits."; return false;

}

else if (!emailPattern.test(email)) { result.style.color = "red";

result.innerHTML = "+ Invalid Email: Must be in format [example@domain.com"](mailto:example@domain.com); return false;

}

else {

result.style.color = "green";

result.innerHTML = "⬛ Registration Successful!"; return false; // prevent actual submission for demo

}

}

</script>

</body>

</html>

**OUTPUT:**

