LAKIREDDY BALI REDDY COLLEGE OF ENGINEERING

(AUTONOMOUS)

23CSS2 – Full Stack Development 1 Lab Manual

Department of Computer Science and Engineering

Vision of the Department

Computer Science & Engineering aims at providing continuously stimulating educational environment to its students for attaining their professional goals and meeting the global challenges.

Mission of the Department

- **DM1:** To develop a strong theoretical and practical background across the computer science discipline with an emphasis on problem solving.
- **DM2:** To inculcate professional behaviors with strong ethical values, leadership qualities, innovative thinking and analytical abilities into the student.
- **DM3:** Expose the students to cutting edge technologies which enhance their employability and knowledge.
- **DM4:** Facilitate the faculty to keep track of the latest developments in their research areas and encourage the faculty to foster healthy interaction with industry.

Program Educational Objectives (PEOs)

- **PEO1:** Pursue higher education, entrepreneurship and research to compete at global level.
- **PEO2:** Design and develop products innovatively in computer science and engineering and in other allied fields.
- **PEO3:** Function effectively as individuals and as members of a team in the conduct of interdisciplinary projects; and even at all levels with ethics and necessary attitude.
- **PEO4:** Serve ever-changing needs of society with a pragmatic perception.

PROGRAMME OUTCOMES (POs):

PO 1	**Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
PO 2	**Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
PO 3	**Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
PO 4	**Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
PO 5	**Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations
PO 6	**The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice
PO 7	**Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
PO 8	**Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
PO 9	**Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
PO 10	**Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to
PO 11	**Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO 12	**Life-long learning:** Recognize the need for and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAMME SPECIFIC OUTCOMES (PSOs):

PSO 1	The ability to apply Software Engineering practices and strategies in software project development using open-source programming environment for the success of organization.
PSO 2	The ability to design and develop computer programs in networking, web applications and IoT as per the society needs.
PSO3	To inculcate an ability to analyze, design and implement database applications.

23CSS2-FSD 1

L	T	P	C
0	1	2	2

Course Objectives:

The main objectives of the course are to

- Make use of HTML elements and their attributes for designing static web pages
- Build a web page by applying appropriate CSS styles to HTML elements
- Experiment with JavaScript to develop dynamic web pages and validate forms.

Course Outcomes:

CO1: Design static web pages by using HTML elements. **(Apply-L3)**
CO2: Develop a web page by applying appropriate CSS styles to HTML elements. **(Apply-L3)**
CO3: Develop dynamic web pages and validate forms using JavaScript. **(Apply-L3)**
CO4: Improve individual / teamwork skills, communication and report writing skills with ethical values.

Experiments covering the Topics:

• Lists, Links and Images
• HTML Tables, Forms and Frames
• HTML 5 and Cascading Style Sheets, Types of CSS
• Selector forms
• CSS with Color, Background, Font, Text and CSS Box Model
• Applying JavaScript - internal and external, I/O, Type Conversion
• JavaScript Conditional Statements and Loops, Pre-defined and User-defined Objects
• JavaScript Functions and Events
• Node.js

List of Experiments

1.Lists, Links and Images

a. Write an 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.

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

c. 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.

d. 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.

2. HTML Tables, Forms and Frames

a. Write a HTML program, to explain the working of tables. (use tags: <table>, <tr>, <th>, <td> and attributes: border, rowspan, colspan)
b. Write a HTML program, to explain the working of tables by preparing a timetable. (Note: Use <caption> tag to set the caption to the table & also use cell spacing, cell padding, border, rowspan, colspan etc.).
c. 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 <select>& <option> tags, <text area> and two buttons ie: submit and reset. Use tables to provide a better view).
d. 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).

3. HTML 5 and Cascading Style Sheets, Types of CSS

a. Write a HTML program, that makes use of <article>, <aside>, <figure>, <figcaption>, <footer>, <header>, <main>, <nav>, <section>, , tags.
b. Write a HTML program, to embed audio and video into HTML web page.
c. Write a program to apply different types (or levels of styles or style specification formats) - inline, internal, external styles to HTML elements. (identify selector, property and value).

4. Selector forms

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

5. CSS with Color, Background, Font, Text and CSS Box Model

a. Write a program to demonstrate the various ways you can reference a color in CSS.
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.
c. Write a program using the following terms related to CSS font and text: i. font-size ii. font-weight iii. font-style iv. text-decoration v. text-transformation vi. text-alignment d. Write a program, to explain the importance of CSS Box model using i.Content ii. Border iii. Margin iv. Padding

6. Applying JavaScript - internal and external, I/O, Type Conversion

a. Write a program to embed internal and external JavaScript in a web page.
b. Write a program to explain the different ways for displaying output.
c. Write a program to explain the different ways for taking input.
d. Create a webpage 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.

7. Java Script Pre-defined and User-defined Objects

a. Write a program using document object properties and methods.
b. Write a program using window object properties and methods.
c. Write a program using array object properties and methods.
d. Write a program using math object properties and methods.
e. Write a program using string object properties and methods.
f. Write a program using regex object properties and methods.

g. Write a program using date object properties and methods. h. Write a program to explain user-defined object by using properties, methods, accessors, constructors and display.

8. Java Script Conditional Statements and Loops

a. 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".

b. Write a program to display week days using switch case.

c. Write a program to print 1 to 10 numbers using for, while and do-while loops.

d. Write a program to print data in object using for-in, for-each and for-of loops

e. Develop a program to determine whether a given number is an 'ARMSTRONG NUMBER' or not. [Eg: 153 is an Armstrong number, since sum of the cube of the digits is equal to the number i.e.,13 + 53+ 33 = 153]

f. 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. (Eg: If deposited amount is Rs.163, the output should be 1- 100's, 1-50's, 1- 10's, 1-2's & 1-1's)

9. Java Script Functions and Events

a. Design a appropriate function should be called to display

i. Factorial of that number

ii. Fibonacci series up to that number

iii. Prime numbers up to that number

iv. palindrome or not

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

i. Factorial of that number

ii. Fibonacci series up to that number

iii. Prime numbers up to that number

iv. Is it palindrome or not

c. Write a program to validate the following fields in a registration page i. Name (start with alphabet and followed by alphanumeric and the length should not be less than 6 characters) ii. Mobile (only numbers and length 10 digits) iii. E-mail (should contain format like xxxxxxx@xxxxxx.xxx)

Text Books:

1. Programming the World Wide Web, 7th Edition, Robet W Sebesta, Pearson, 2013.

2. Web Programming with HTML5, CSS and JavaScript, John Dean, Jones & Bartlett Learning, 2019 (Chapters 1-11).

3. Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node, Vasan Subramanian, 2nd edition, A Press, O'Reilly.

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 Lists Example</title>
</head>
<body>
  <h1>HTML Lists Example</h1>
  <h2>Ordered List</h2>
  <ol>
        <li>First Item</li>
         <li>Second Item</li>
        <li>Third Item</li>
  </ol>
  <h2>Unordered List</h2>
  <ul>
         <li>Item A</li>
        <li>Item B</li>
         <li>Item C</li>
  </ul>
  <h2>Nested List</h2>
  <ul>
        <li>Fruit
        <ul>
                <li>Apple</li>
                <li>Banana</li>
                <li>Cherry</li>
        </ul>
    </li>
    <li>Vegetables
        <ul>
                <li>Carrot</li>
                <li>Broccoli</li>
        </ul>
    </li>
  </ul>
  <h2>Ordered List Inside Unordered List</h2>
  <ul>
    <li>First Category
        <ol>
                <li>Sub-item 1</li>
                <li>Sub-item 2</li>
        </ol>
    </li>
    <li>Second Category
        <ol>
                 <li>Sub-item 1</li>
                <li>Sub-item 2</li>
        </ol>
    </li>
```

```
  </ul>
  <h2>Definition List</h2>
  <dl>
    <dt>HTML</dt>
    <dd>Hypertext Markup Language</dd>
    <dt>CSS</dt>
    <dd>Cascading Style Sheets</dd>
    <dt>JS</dt>
    <dd>JavaScript</dd>
  </dl>
</body>
</html>
```

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

```
<html >
<head>
    <title>HTML Hyperlinks Example</title>
</head>
<body>
  <h1>HTML Hyperlinks Example</h1>
  <h2>Visit OpenAI</h2>
  <a href="https://www.google.com/" target="_blank">Go to Website</a>
  <h2>Navigation within the page</h2>
  <a href="#section2">Go to Section 2</a>
  <h2 id="section2">Section 2</h2>
  <p>This is Section 2. You came here from the hyperlink above.</p>
  <h2>Download File</h2>
  <a href="example.pdf" download>Download PDF</a>
</body>
</html>
```

c. 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>HTML Images with Links</title>
</head>
<body>
         <h1>Profile Images</h1>
         <!-- My Profile Image -->
         <a href="https://myprofile.com" target="_blank">
                 <img src="my_image.jpg" alt="My Image" width="150" height="150">
         </a>
         <!-- Friend's Profile Image -->
        <a href="https://myfriendsprofile.com" target="_blank">
        <img src="friend_image.jpg" alt="Friend's Image" width="150" height="150">
        </a>
</body>
</html>
```

d. 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.

```
<html>
<head>
    <title>Image Gallery</title>
</head>
<body>
  <h1>Image Gallery</h1>
  <!-- Thumbnail Image 1 -->
  <a href="fullsize_image1.jpg" target="_blank">
    <img src="thumb_image1.jpg" alt="Image 1" width="100" height="100">
  </a>
  <!-- Thumbnail Image 2 -->
  <a href="fullsize_image2.jpg" target="_blank">
    <img src="thumb_image2.jpg" alt="Image 2" width="100" height="100">
  </a>
  <!-- Thumbnail Image 3 -->
  <a href="fullsize_image3.jpg" target="_blank">
    <img src="thumb_image3.jpg" alt="Image 3" width="100" height="100">
  </a>
  <!-- Thumbnail Image 4 -->
  <a href="fullsize_image4.jpg" target="_blank">
    <img src="thumb_image4.jpg" alt="Image 4" width="100" height="100">
  </a>
</body>
</html>
```

2. HTML Tables, Forms and Frames

a. Write a HTML program, to explain the working of tables. (use tags: <table>, <tr>, <th>, <td> and attributes: border, rowspan, colspan)

```
<html>
<head>
    <title>HTML Table Example</title>
</head>
<body>
<h2>HTML Table Example</h2>
<table border="1">
   <tr>
         <th>Header 1</th>
         <th>Header 2</th>
          <th>Header 3</th>
   </tr>
   <tr>
          <td>Row 1, Column 1</td>
         <td>Row 1, Column 2</td>
         <td>Row 1, Column 3</td>
   </tr>
   <tr>
         <td rowspan="2">Rowspan 2</td>
         <td>Row 2, Column 2</td>
          <td>Row 2, Column 3</td>
   </tr>
   <tr>
          <td colspan="2">Colspan 2</td>
   </tr>
</table>
</body>
</html>
```

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

```
<html>
<head>
    <title>Timetable</title>
   <style>
      table {
         border-collapse: collapse;
         width: 60%;
      }
      th, td {
         padding: 10px;
         text-align: center;
      }
   </style>
</head>
<body>
```

```
<h2>Weekly Timetable</h2>
<table border="3" cellspacing="2" cellpadding="5">
   <caption>My Weekly Timetable</caption>
   <tr>
      <th>Time</th>
      <th>9:00 AM - 10:00 AM</th>
      <th>10:00 AM - 11:00 AM</th>
      <th>11:00 AM - 12:00 PM</th>
        <th>Break</th>
      <th>1:00 PM - 2:00 PM</th>
      <th>2:00 PM - 3:00 PM</th>
       <th>3:00 PM - 4:00 PM</th>
   </tr>
   <tr>
      <td>Monday</td>
      <td>DMS</td>
      <td>OOP</td>
      <td>ADS</td>
        <td>Break</td>
      <td>DAA</td>
      <td>CD</td>
       <td>TOC</td>
   </tr>
   <tr>
          <td>Tuesday</td>
         <td>CD</td>
         <td>ADS</td>
         <td>DAA</td>
          <td>Break</td>
         <td>TOC</td>
          <td>DMS</td>
         <td>OOP</td>
   </tr>
</table>
</body>
</html>
```

c. 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 <select>& <option> tags, <text area> and two buttons ie: submit and reset. Use tables to provide a better view).

```
<html>
<head>
    <title>Registration Form</title>
</head>
<body>
<h2>Registration Form</h2>
<form action="#" method="POST">
   <table border="1" cellpadding="10">
      <tr>
                        <td>Name:</td>
                        <td><input type="text" name="name" required></td>
```

```
    </tr>
    <tr>
                <td>Password:</td>
                <td><input type="password" name="password" required></td>
    </tr>
    <tr>
                <td>Age:</td>
                        <td><input type="number" name="age" required></td>
    </tr>
    <tr>
                <td>Date of Birth:</td>
                <td><input type="date" name="dob" required></td>
    </tr>
    <tr>
                <td>Gender:</td>
                 <td>
               <input type="radio" name="gender" value="Male" required> Male
               <input type="radio" name="gender" value="Female" required> Female
                        </td>
    </tr>
    <tr>
                        <td>Hobbies:</td>
                        <td>
                  <input type="checkbox" name="hobbies" value="Reading"> Reading
                 <input type="checkbox" name="hobbies" value="Sports"> Sports
                <input type="checkbox" name="hobbies" value="Music"> Music
                        </td>
    </tr>
    <tr>
                <td>Country:</td>
                <td>
                        <select name="country">
                         <option value="USA">USA</option>
                        <option value="India">India</option>
                         <option value="UK">UK</option>
                         <option value="Canada">Canada</option>
                        </select>
                        </td>
    </tr>
    <tr>
                <td>Address:</td>
                        <td><textarea name="address" rows="4" cols="50"></textarea></td>
    </tr>
    <tr>
                <td colspan="2" align="center">
               <input type="submit" value="Submit">
                <input type="reset" value="Reset">
                        </td>
    </tr>
  </table>
</form>
</body>
</html>
```

d. 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).

```
<html>
<head>
   <title>Frames Example</title>
</head>
<body>
<frameset cols="25%, 50%, 25%">
          <frame src="image.html" name="imageFrame">
         <frame src="text.html" name="textFrame">
          <frame src="links.html" name="linkFrame">
</frameset>
<noframes>
          <h3>Your browser does not support frames.</h3>
</noframes>
</body>
</html>
```

3. HTML 5 and Cascading Style Sheets, Types of CSS

a. Write a HTML program, that makes use of <article>, <aside>, <figure>, <figcaption>, <footer>, <header>, <main>, <nav>, <section>, , tags.

```
<html>
<head>
  <title>HTML5 Tags Example</title>
  <style>
    header, footer {
      background-color: #f1f1f1;
      padding: 10px;
      text-align: center;
    }
    article {
      background-color: #e0f7fa;
      padding: 20px;
      margin: 10px;
    }
    section {
      margin: 10px 0;
      background-color: #f4e1d2;
    }
    nav {
      background-color: #4CAF50;
      padding: 10px;
      text-align: center;
    }
    aside {
      background-color: #ffeb3b;
      padding: 10px;
      margin-top: 20px;
    }
  </style>
</head>
<body>

<header>
  <h1>Welcome to My Website</h1>
  <nav>
    <a href="#">Home</a> | <a href="#">About</a> | <a href="#">Contact</a>
  </nav>
</header>

<main>
  <section>
    <h2>Introduction</h2>
    <p>This is an example of using various HTML5 semantic tags.</p>
  </section>

  <article>
    <h2>Article Heading</h2>
    <p>This is an article content. It can contain text, images, or other elements that represent a stand-alone
```

```
    piece of content.</p>
          <figure>
            <img src="images.png" alt="Example Image">
            <figcaption>Figure 1: Example image with a caption.</figcaption>
          </figure>
        </article>

        <aside>
          <h3>Aside Section</h3>
          <p>This is an aside. It can contain content related to the main article but is separate in context.</p>
        </aside>
    </main>

    <footer>
        <p>Footer content here. &copy; 2025 My Website</p>
    </footer>

    <div>
        <h2>Div Section</h2>
        <p>This section is inside a <code>&lt;div&gt;</code> tag, which is used for grouping content.</p>
    </div>
    <span style="color: blue;">This text is inside a span tag with inline styling.</span>
    </body>
    </html>
```

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

```
<html>
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Audio and Video Embedding</title>
</head>
<body>

<h2>Embedding Audio</h2>
<audio controls>
  <source src="https://www.soundhelix.com/examples/mp3/SoundHelix-Song-1.mp3" type="audio/mp3">
  Your browser does not support the audio element.
</audio>

<h2>Embedding Video</h2>
<video width="320" height="240" controls>
  <source src="https://www.w3schools.com/html/movie.mp4" type="video/mp4">
  Your browser does not support the video tag.
</video>

</body>
</html>
```

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

```
<html >
<head>
   <title>CSS Types Example</title>
   <style>
      /* Internal CSS */
      h1 {
         color: blue;
      }
      p {
         font-size: 18px;
         color: green;
      }
   </style>

   <!-- Link to external stylesheet (external.css) -->
   <link rel="stylesheet" href="external.css">

</head>
<body>
<!-- Inline CSS -->
<h1 style="text-align: center; color: red;">Welcome to CSS Example</h1>
<p>This is a paragraph with internal CSS applied.</p>
<p style="font-weight: bold; color: orange;">This is a paragraph with inline CSS.</p>
</body>
</html>
```

Here's an example of what `external.css` could look like:

```
/* external.css */
p {
   color: purple;
   font-size: 16px;
}

h2 {
   font-family: 'Arial', sans-serif;
}
```

4. Selector forms

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

```
<html>
<head>
  <title>CSS Selector Forms</title>
  <style>
    /* i. Simple selectors */

    /* Element Selector */
    p {
      color: green;
    }

    /* ID Selector */
    #special-text {
      font-weight: bold;
      color: red;
    }
    /* Class Selector */
    .highlight {
      background-color: yellow;
    }
    /* Group Selector */
    h1, h2, h3 {
      font-family: Arial, sans-serif;
    }
    /* Universal Selector */
    * {
      margin: 0;
      padding: 0;
    }
    /* ii. Combinator selectors */
    /* Descendant Selector */
    .container p {
      color: blue;
    }
    /* Child Selector */
    .container > p {
      font-style: italic;
    }
    /* Adjacent Sibling Selector */
    h2 + p {
      font-size: 20px;
    }

    /* General Sibling Selector */
```

```
    h2 ~ p {
        color: purple;
    }
    /* iii. Pseudo-class selector */
    /* Hover pseudo-class */
    a:hover {
        color: orange;
    }
    /* iv. Pseudo-element selector */
    /* Before pseudo-element */
    p::before {
        content: "Note: ";
        font-weight: bold;
        color: blue;
    }
    /* After pseudo-element */
    p::after {
        content: " (end of paragraph)";
        font-style: italic;
        color: gray;
    }
    /* v. Attribute Selector */
    /* Selecting elements with a specific attribute */
    input[type="text"] {
        border: 2px solid green;
    }
    /* Selecting elements with a specific attribute value */
    a[href^="https://"] {
        color: green;
    }
  </style>
</head>
<body>
<h1>CSS Selector Forms Demonstration</h1>
<p>This is a simple paragraph to demonstrate different types of CSS selectors.</p>
<p id="special-text">This paragraph has an ID selector applied to it.</p>
<p class="highlight">This paragraph has a class selector applied to it.</p>
<div class="container">
  <p>This is a descendant selector example. It will be colored blue.</p>
  <p>This is a child selector example. It will be italicized.</p>
</div>
<h2>Sibling Selector Example</h2>
<p>This paragraph follows an <code>h2</code> element, so it is affected by the adjacent sibling selector.</p>
<h2>General Sibling Example</h2>
<p>This paragraph will be colored purple as it is a sibling of the previous <code>h2</code> element.</p>
<a href="https://www.example.com">Visit Example Website</a>
<input type="text" placeholder="Type here">
<p>Hover over this <a href="#">link</a> to see the hover effect.</p>
</body>
</html>
```

5. CSS with Color, Background, Font, Text and CSS Box Model

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

```
<html>
<head>
  <title>Color Demonstration</title>
  <style>
    /* Different ways to specify color in CSS */

    .color-named {
      background-color: red; /* Named color */
      color: white;
      padding: 10px;
      margin: 5px;
    }

    .color-hex {
      background-color: #3498db; /* Hexadecimal color */
      color: white;
      padding: 10px;
      margin: 5px;
    }

    .color-rgb {
      background-color: rgb(34, 193, 195); /* RGB color */
      color: white;
      padding: 10px;
      margin: 5px;
    }

    .color-rgba {
      background-color: rgba(255, 99, 71, 0.7); /* RGBA color with transparency */
      color: white;
      padding: 10px;
      margin: 5px;
    }

    .color-hsl {
      background-color: hsl(120, 100%, 50%); /* HSL color */
      color: white;
      padding: 10px;
      margin: 5px;
    }

    .color-hsla {
      background-color: hsla(240, 100%, 50%, 0.5); /* HSLA color with transparency */
      color: white;
      padding: 10px;
      margin: 5px;
    }
  </style>
</head>
<body>
```

```
<h2>Demonstrating Various Ways to Define Colors in CSS</h2>

<div class="color-named">Named Color: Red</div>
<div class="color-hex">Hex Color: #3498db</div>
<div class="color-rgb">RGB Color: rgb(34, 193, 195)</div>
<div class="color-rgba">RGBA Color: rgba(255, 99, 71, 0.7)</div>
<div class="color-hsl">HSL Color: hsl(120, 100%, 50%)</div>
<div class="color-hsla">HSLA Color: hsla(240, 100%, 50%, 0.5)</div>
</body>
</html>
```

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.

```
<html>
<head>
  <title>Background Image Example</title>
  <style>
    body {
      height: 200vh; /* Make the body tall enough to scroll */
      margin: 0;
      padding: 0;
      background-image: url('https://via.placeholder.com/150'); /* Background image URL */
      background-position: center 50%; /* Position halfway down the page */
      background-attachment: fixed; /* Keep the background fixed while scrolling */
      background-size: cover; /* Ensure the background image covers the screen */
    }
  </style>
</head>
<body>
<h1>Scroll Down to See the Background Image</h1>
<p>When you scroll, the background image stays in place.</p>
</body>
</html>
```

c. Write a program using the following terms related to CSS font and text: i. font-size ii. font-weight iii. font-style iv. text-decoration v. text-transformation vi. text-alignment d. Write a program, to explain the importance of CSS Box model using i.Content ii. Border iii. Margin iv. Padding

```
<html>
<head>
  <title>Font and Text Properties</title>
  <style>
    h1 {
      font-size: 36px; /* Font size */
      font-weight: bold; /* Font weight */
      font-style: italic; /* Font style */
      text-decoration: underline; /* Text decoration */
```

```
        text-transform: uppercase; /* Text transformation */
        text-align: center; /* Text alignment */
        color: #333;
    }

    p {
        font-size: 16px;
        font-weight: normal;
        font-style: normal;
        text-decoration: line-through; /* Strikethrough text */
        text-transform: capitalize; /* Capitalize first letter of each word */
        text-align: justify; /* Justify alignment */
        color: #555;
    }
  </style>
</head>
<body>
<h1>CSS Font and Text Properties</h1>
<p>This paragraph demonstrates font and text properties like font-size, font-weight, font-style, text-decoration, text-transform, and text-align.</p>
</body>
</html>
```

6. Applying JavaScript - internal and external, I/O, Type Conversion

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

```
<html l>
<head>
    <title>Internal & External JavaScript</title>
</head>
<body>

   <h1>JavaScript Embedding Example</h1>

   <button onclick="internalFunction()">Run Internal Script</button>
   <button onclick="externalFunction()">Run External Script</button>

   <!-- Internal JavaScript -->
   <script>
      function internalFunction() {
         alert("Hello from Internal JavaScript!");
      }
   </script>

   <!-- External JavaScript -->
   <script src="script.js"></script>

</body>
</html>
```

External JavaScript: -

```
alert(“Hello”);
```

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

```
<html>
<head>
   <title>JavaScript Output Methods</title>
</head>
<body>

   <h2>JavaScript Output Methods</h2>

   <button onclick="showConsoleLog()">Console Log</button>
   <button onclick="showDocumentWrite()">Document Write</button>
   <button onclick="showAlert()">Alert Box</button>
   <button onclick="showInnerHTML()">Inner HTML</button>

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

   <script>
      // 1. Console.log() - Outputs to the browser console
      function showConsoleLog() {
         console.log("This is a console log message.");
```

```
    }

    // 2. document.write() - Writes directly to the page (not recommended for dynamic sites)
    function showDocumentWrite() {
        document.write("This message is written using document.write()!");
    }

    // 3. alert() - Displays an alert box
    function showAlert() {
        alert("This is an alert box message!");
    }

    // 4. innerHTML - Inserts content inside an HTML element
    function showInnerHTML() {
        document.getElementById("output").innerHTML = "This message is displayed using innerHTML!";
    }
  </script>

</body>
</html>
```

c. Write a program to explain the different ways for taking input.

```
<html>
<head>
  <title>JavaScript Input Methods</title>
</head>
<body>

  <h2>JavaScript Input Methods</h2>

  <button onclick="getInputPrompt()">Prompt Input</button>
  <button onclick="getConfirmation()">Confirm Input</button>

  <br><br>

  <label for="textInput">Enter Text:</label>
  <input type="text" id="textInput">
  <button onclick="getInputFromTextBox()">Get Input</button>

  <br><br>

  <label for="dynamicInput">Type Something:</label>
  <input type="text" id="dynamicInput" oninput="dynamicInputHandler()">
  <p id="dynamicOutput"></p>

  <script>
    // 1. Using prompt() - Takes input through a pop-up window
    function getInputPrompt() {
        let name = prompt("Enter your name:");
        alert("You entered: " + name);
```

```
    }

    // 2. Using an input field and getElementById()
    function getInputFromTextBox() {
        let userInput = document.getElementById("textInput").value;
        alert("Text Box Input: " + userInput);
    }

    // 3. Using window.confirm() - Yes/No confirmation
    function getConfirmation() {
        let confirmation = confirm("Do you want to proceed?");
        if (confirmation) {
            alert("You clicked OK!");
        } else {
            alert("You clicked Cancel!");
        }
    }

    // 4. Using oninput event listener for dynamic input
    function dynamicInputHandler() {
        let inputText = document.getElementById("dynamicInput").value;
        document.getElementById("dynamicOutput").innerHTML = "You typed: " + inputText;
    }
  </script>

</body>
</html>
```

d. Create a webpage 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.

```
<html>
<head>
  <title>Voter Eligibility Check</title>
</head>
<body>

  <h2>Voter Eligibility Check</h2>
  <button onclick="getVoterDetails()">Check Voter Eligibility</button>

  <br><br>

  <table id="voterTable">
    <tr>
      <th>Name</th>
      <th>Age</th>
      <th>Eligibility</th>
    </tr>
    <tr>
      <td id="voterName"></td>
      <td id="voterAge"></td>
      <td id="voterStatus"></td>
```

```
    </tr>
  </table>

  <script>
    function getVoterDetails() {
      let name = prompt("Enter your name:");
      let age = prompt("Enter your age:");

      if (name && age) {
        age = parseInt(age); // Convert age to number

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

        document.getElementById("voterName").innerText = name;
        document.getElementById("voterAge").innerText = age;
        document.getElementById("voterStatus").innerText = eligibility;

        document.getElementById("voterTable").style.display = "table";
      } else {
        alert("Please enter valid details!");
      }
    }
  </script>

</body>
</html>
```

7. Java Script Pre-defined and User-defined Objects

a. Write a program using document object properties and methods.

Demonstrating methods of document object: -

```
<html>
	<head>
		<title>Demonstrating Document Object</title>
	</head>
	<body>
		<p id="demo">Click the button to change the text in this paragraph.</p>
		<input type="button" onClick="myFunction1()" value="id"/>

		<p class="example">First p element with class="example"</p>
		<p class="example">Second p element with class="example"</p>
		<p>Click the button to change the text of the first p element with class="example" (index 0).</p>
		<input type="button" onClick="myFunction2()"  value="Try it"/>

		<p>An unordered list:</p>
		<ul>
			 <li>Coffee</li>
			<li>Tea</li>
			 <li>Milk</li>
		</ul>

		<p id="first">Click the button to display the innerHTML of the second li element (index 1).</p>
		<input type="button" onClick="myFunction3()"  value="Try it"/>

		<script>
			function myFunction1()
			{
				document.getElementById("demo").innerHTML = "Hello World";
			}
			function myFunction2()
			{
				var x = document.getElementsByClassName("example");
				x[0].innerHTML = "Hello World!";
			}
			function myFunction3() {
				 var x = document.getElementsByTagName("LI");
				   document.getElementById("first").innerHTML = x[1].innerHTML;
			}



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

Demonstrating Properties of document object: -

```
<html>
	<head>
		<title>Demonstrating Document Object</title>
	</head>
	<body>

		<p><a href="http://www.google.com">GOOGLE</a></p>
		<p><a href="http://www.gmail.com" name="first">GMAIL</a></p>
		<form>	</form>
		<form>	</form>
		<form>	</form>
		<p>
		<img src="028.jpg" height="200" width="200" border="10" usemap="#mymap"/>
		<map name="mymap">
			<area shape="rectangle" coords="0,0,20,20" href="font.html"/>
			<area shape="circle" coords="70,60,20" href="font styles.html"/>
		</map>
		</p>

		<script>
			document.bgColor="pink";
			document.fgColor="red";
			document.title="Setting title using title property";
			document.write("Number of forms used in the document are :"+document.forms.length);
			document.write("<br/>Number of links used in the document are :"+document.links.length);
			document.write("<br/>Number of anchors used in the document are
:"+document.anchors.length);

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

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

```
<html>
<body>
// Display window properties
console.log("Window Width: " + window.innerWidth);
console.log("Window Height: " + window.innerHeight);
console.log("Current URL: " + window.location.href);
console.log("Browser Name: " + window.navigator.appName);
console.log("Platform: " + window.navigator.platform);

// Alert message
window.alert("Welcome to the JavaScript Window Object Demo!");

// Confirm dialog
let userResponse = window.confirm("Do you like JavaScript?");
if (userResponse) {
   console.log("User likes JavaScript! 😃");
} else {
```

```
    console.log("User does not like JavaScript! 😢");
}

// Prompt dialog
let userName = window.prompt("Enter your name:");
if (userName) {
    console.log("Hello, " + userName + "! 👋");
}

// Open a new tab (commented to prevent accidental execution)
// window.open("https://www.google.com", "_blank");

// SetTimeout Example (Executes after 3 seconds)
window.setTimeout(() => {
    console.log("This message appears after 3 seconds!");
}, 3000);

// SetInterval Example (Repeats every 2 seconds)
let count = 0;
let intervalId = window.setInterval(() => {
    count++;
    console.log("Interval running... Count: " + count);
    if (count === 5) {
        clearInterval(intervalId); // Stop after 5 times
        console.log("Interval stopped.");
    }
}, 2000);
</body>
</html>
```

c. Write a program using array object properties and methods.

```
<html>
        <head>
                <title>Object Based Array Functions</title>
        </head>
        <body>
                <script>
                        var a=[70,20,300,60,50];
                        var b=new Array(1.1,2.2,3.3);
                        var c=["ramu","naresh"];
                        var concat=a.concat(b,c);
                        document.write("<br/>After concatenation resultant array is :" +concat.join(","));
                        a.pop();
                        document.writeln("<br/>After pop operation resultant array is: " +a.join(" "));
                        b.push(4.4,5.5);
                        document.write("<br/>After push operation resultant array is: "+b.join(" "));
                        c.reverse();
                        document.write("<br/>After reversing resultant array is :"+c.join(" "));
                        a.sort();
                        document.write("<br/>After sorting resultant array is :"+a.join(" "));
                        a.splice(2,1,"suresh","ramesh");
                        document.write("<br/>After splice operaion resultant array is :"+a.join(" "));
```

```
                a.shift();
                document.writeln("<br/>After shift operation resultant array is: " +a.join(" "));
                a.unshift(100,200);
                document.writeln("<br/>After unshift operation resultant array is: " +a.join(" "));
            </script>
        </body>
</html>
```

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

Java script to demonstrate math object methods: -

```
<html>
    <head>
        <title>Demonstrating Math Object</title>
    </head>
    <body>
        <script>
            document.writeln("<h5/>Absolute Value of -4.7 is :" +Math.abs(-4.7)+ "</h2>"); // 4.7
            document.writeln("<h5/>Ceil Value of 4.4 is :" +Math.ceil(4.4)+ "</h2>");  //5
            document.writeln("<h5/>floor Value of 4.4 is :" +Math.floor(4.4)+ "</h2>");  //4
            document.writeln("<h5/>round Value of 4.4 is :" +Math.round(4.4)+ "</h2>"); //4
            document.writeln("<h5/>round Value of 4.6 is :" +Math.ceil(4.6)+ "</h2>");  //5
            document.writeln("<h5/>sin(0) value is :" +Math.sin(0)+ "</h2>"); //0
            document.writeln("<h5/>cos(0) value is :" +Math.cos(0)+ "</h2>"); //1
            document.writeln("<h5/>tan(0) value is :" +Math.tan(0)+ "</h2>");  //0
            document.writeln("<h5/>Minimum value is : " +Math.min(0, 150, 30, 20)+ "</h2>"); //0
            document.writeln("<h5/>Maximum value is : " +Math.max(0, 150, 30, 20, -8)+ "</h2>");
//150
            docume nt.writeln("<h5/>Random value is : " +Math.random()+ "</h2>");  //0.8
            document.writeln("<h5/>Random value is : " +Math.random()+ "</h2>");  //0.5
            document.writeln("<h5/>Random value is : " +Math.random()+ "</h2>");  //0.2
            document.writeln("<h5> pow(3,4) is : "+Math.pow(3,4) +"</h2>");  //81
            document.writeln("<h5>log(4) result is: "+Math.log(4) +"</h2>"); //1.3
            document.writeln("<h5>exp(2) result is: "+Math.exp(2) +"</h2>"); //7.3
            document.writeln("<h5>sqrt(121) result is: "+Math.sqrt(121) +"</h2>");  //11
            document.writeln("<h5>Math.E result is: "+Math.E +"</h2>");//2.71
            document.writeln("<h5>Math.SQRT2 result is: "+Math.SQRT2+"</h2>"); //1.414
            document.writeln("<h5>Math.PI result is: "+Math.PI +"</h2>");  //3.14

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

Java script to demonstrate math object Properties: -

```
< html>
<body>

<h2>JavaScript Math Constants</h2>
```

```
<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML =
"<p><b>Math.E:</b> " + Math.E + "</p>" +
"<p><b>Math.PI:</b> " + Math.PI + "</p>" +
"<p><b>Math.SQRT2:</b> " + Math.SQRT2 + "</p>" +
"<p><b>Math.SQRT1_2:</b> " + Math.SQRT1_2 + "</p>" +
"<p><b>Math.LN2:</b> " + Math.LN2 + "</p>" +
"<p><b>Math.LN10:</b> " + Math.LN10 + "</p>" +
"<p><b>Math.LOG2E:</b> " + Math.LOG2E + "</p>" +
"<p><b>Math.Log10E:</b> " + Math.LOG10E + "</p>";
</script>

</body>
</html>
```

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

```
<html>
	<head>
		<title>Demonstring String object Methods and Properties</title>
	</head>
	<body>
		<script>
			var s="abcdefghijklmnopqrstuvwxyzabcdefghi";
			document.writeln("character at 3rd index is :"+s.charAt(3)+ "<br/>");
			document.writeln("ASCII value of character at 3rd index is :"+s.charCodeAt(25) +"<br/>");
			document.writeln("After Converting ASCII values to string
:"+String.fromCharCode(97,98,99) +"<br/>");
			document.writeln("After Converting String to Upprcase result is :"+s.toUpperCase()+
"<br/>");
			var a="XYZ";
			document.writeln("After Converting String to Lowercase result is :"+a.toLowerCase()+
"<br/>");
			document.writeln("After concatenation result is :" +a.concat(s)+ "<br/>");
			document.writeln("indexOf ghi string is :" +s.indexOf('ghi')+"<br/>");
			document.writeln("indexOf ghi string is :" +s.indexOf('abc',10)+"<br/>");
			document.writeln("last indexOf ghi string is :" +s.lastIndexOf('ghi')+"<br/>");
			document.writeln("last indexOf ghi string is :" +s.lastIndexOf('abc',10)+"<br/>");
			document.writeln("After extracting substring using slice method result is :"+s.slice(3,6)+
"<br/>");
			document.writeln("After extracting substring using substring method result is
:"+s.substring(3,6)+ "<br/>");
			document.writeln("After extracting substring using substr result is method:"+s.substr(3,6)+
"<br/>");
			var s="hello good day";
			a=s.split(" ");
			document.write(a.join("+"));

		</script>
	</body>
```

```
</html>
```

f. Write a program using regex object properties and methods.

```
<html>
<body>
var regex = /hello/i; // Case-insensitive match
var text = "Hello World!";
console.log(regex.test(text));
console.log(text.match(regex));
console.log(text.replace(regex, "Hi"));
console.log(text.search(regex));
console.log(text.split(/o/));
</body>
</html>
```

g. Write a program using date object properties and methods.

```
<html>
    <head>
        <title>Demonstrating Date object methods</title>
    </head>
    <body>
        <script>
            var current=new Date();
            document.writeln("Date in String representation is :"+current.toString()+"<br/>");
            document.writeln("Date in local computer representation is
:"+current.toLocaleString()+"<br/>");
            document.writeln("Date in Universal representation is :"+current.toUTCString()+"<br/>");
            document.writeln("<h1>Get methods for local time zone</h1>");
            document.writeln("getDate is :" +current.getDate()+"<br/>");
            document.writeln("getDay is :" +current.getDay()+"<br/>");
            document.writeln("getMonth is :" +current.getMonth()+"<br/>");
            document.writeln("getFullYear is :" +current.getFullYear()+"<br/>");
            document.writeln("getHours is :" +current.getHours()+"<br/>");
            document.writeln("getMinutes is :" +current.getMinutes()+"<br/>");
            document.writeln("getSeconds is :" +current.getSeconds()+"<br/>");
            document.writeln("getMilliseconds is :" +current.getMilliseconds()+"<br/>");
            document.writeln("getTimeZoneOffSet is :" +current.getTimezoneOffset()+"<br/>");
            document.writeln("<h1>Get methods for Universal time zone</h1>");
            document.writeln("getUTCDate is :" +current.getUTCDate()+"<br/>");
            document.writeln("getUTCDay is :" +current.getUTCDay()+"<br/>");
            document.writeln("getUTCMonth is :" +current.getUTCMonth()+"<br/>");
            document.writeln("getUTCFullYear is :" +current.getUTCFullYear()+"<br/>");
            document.writeln("getUTCHours is :" +current.getUTCHours()+"<br/>");
            document.writeln("getUTCMinutes is :" +current.getUTCMinutes()+"<br/>");
            document.writeln("getUTCSeconds is :" +current.getUTCSeconds()+"<br/>");
            document.writeln("getUTCMilliseconds is :" +current.getUTCMilliseconds()+"<br/>");
            document.writeln("<h1>Specifying Arguments for new Date :</h1>");
            var anotherDate=new Date(2000,7,16,04,30,40,90);
            document.write("New Date is :"+anotherDate);
            document.writeln("<h1>Set methods for local time zone</h1>");
            anotherDate.setDate(22);
```

```
                    anotherDate.setMonth(9);
                    anotherDate.setFullYear(2017);
                    anotherDate.setHours(3);
                    anotherDate.setMinutes(13);
                    anotherDate.setSeconds(5);
                    anotherDate.setMilliseconds(528);
                    document.write("Modified Date is :"+anotherDate);
                    document.write("<h1>getTime is :"+current.getTime());
                </script>
        </body>
</html>
```

h. Write a program to explain user-defined object by using properties, methods, accessors, constructors and display.

```
// User-defined object using a constructor function
class Person {
    // Constructor to initialize properties
    constructor(name, age, country) {
        this._name = name;  // Private convention (_name)
        this.age = age;
        this.country = country;
    }

    // Method to display person details
    display() {
        console.log(`Name: ${this._name}, Age: ${this.age}, Country: ${this.country}`);
    }

    // Getter method to access name
    get name() {
        return this._name;
    }

    // Setter method to modify name
    set name(newName) {
        if (newName.length > 0) {
            this._name = newName;
        } else {
            console.log("Invalid Name!");
        }
    }

    // Another method
    celebrateBirthday() {
        this.age += 1;
        console.log(`${this._name} just turned ${this.age}! 🎉`);
    }
}

// Creating an object using the constructor
let person1 = new Person("Ravi", 25, "India");
```

```
// Accessing properties and methods
person1.display(); // Name: Ravi, Age: 25, Country: India

// Using getter
console.log(`Person Name: ${person1.name}`); // Person Name: Ravi

// Using setter
person1.name = "Rajesh";  // Updating name
console.log(`Updated Name: ${person1.name}`); // Updated Name: Rajesh

// Calling another method
person1.celebrateBirthday(); // Rajesh just turned 26!
</script>
</body>
</html>
```

8. Java Script Conditional Statements and Loops

a. 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".

```
<html>
<head>
   <title>Find Larger Number</title>
</head>
<body>
   <h2>Enter Three Numbers to Find the Largest</h2>

   <script>
      // Prompt user to enter three integers
      let num1 = parseInt(prompt("Enter the first number:"));
      let num2 = parseInt(prompt("Enter the second number:"));
      let num3 = parseInt(prompt("Enter the third number:"));

      // Check if the numbers are equal
      if (num1 === num2 && num2 === num3) {
         alert("EQUAL NUMBERS");
      } else {
         // Compare numbers and display the largest one
         let largest = Math.max(num1, num2, num3);
         alert(largest + " LARGER NUMBER");
      }
   </script>
</body>
</html>
```

b. Write a program to display week days using switch case.

```
<html>
<head>
   <title>Weekdays Display</title>
</head>
<body>
   <h2>Enter a Number (0-6) to Display the Day of the Week</h2>
    <script>
      // Prompt the user to enter a number (0-6)
      let dayNumber = parseInt(prompt("Enter a number between 0 and 6:"));

      // Use a switch statement to display the corresponding day
      switch(dayNumber) {
         case 0:
            alert("Sunday");
            break;
         case 1:
```

```
            alert("Monday");
            break;
        case 2:
            alert("Tuesday");
            break;
        case 3:
            alert("Wednesday");
            break;
        case 4:
            alert("Thursday");
            break;
        case 5:
            alert("Friday");
            break;
        case 6:
            alert("Saturday");
            break;
        default:
            alert("Invalid input! Please enter a number between 0 and 6.");
            break;
    }
  </script>
</body>
</html>
```

c. Write a program to print 1 to 10 numbers using for, while and do-while loops.

Using for loop:-

```
<html>
<body>
<script>
for (i = 1; i <= 10; i++) {
  document.writeln(i);
}
</script>
<body>
</html>
```

Using while loop:-

```
<html>
<body>
<script>
i=1;
while(i <= 10) {
  document.writeln(i);
  i++;
}
</script>
<body>
</html>
```

Using do-while loop:-

```
<html>
<body>
<script>
i=1;
do {
    document.writeln(i);
    i++;
}while(i<=10);
</script>
<body>
</html>
```

d. Write a program to print data in object using for-in, for-each and for-of loops

Using for-in loop:-

```
<html>
<body>
<script>
// Example Object
let person = {
    name: "John",
    age: 30,
    city: "New York"
};

console.log("Using for-in loop:");
for (let key in person) {
    console.log(`${key}: ${person[key]}`);
</script>
<body>
</html>
```

Using for – each loop:-

```
<html>
<body>
<script>
// Example Object
let person = {
    name: "John",
    age: 30,
    city: "New York"
};

console.log("Using forEach loop:");
Object.entries(person).forEach(([key, value]) => {
    console.log(`${key}: ${value}`);
});
</script>
```

```
</body>
</html>
```

Using for – of loop:-

```
<html>
<body>
<script>
// Example Object
let person = {
    name: "John",
    age: 30,
    city: "New York"
};

console.log("Using for-of loop:");
for (let [key, value] of Object.entries(person)) {
    console.log(`${key}: ${value}`);
}
</script>
</body>
<html>
```

e. Develop a program to determine whether a given number is an 'ARMSTRONG NUMBER' or not. [Eg: 153 is an Armstrong number, since sum of the cube of the digits is equal to the number i.e.,13 + 53+ 33 = 153]

```
<html>
<body>
<script>
n=parseInt(prompt("enter a number"));
temp=n;
count=0;
armstrong=n;
sum=0;
while(temp>0)
{
count++;
temp=parseInt(temp/10);
}
while(n>0)
{
r=n%10;
sum=sum+Math.pow(r,count);
n=parseInt(n/10);
}
if(armstrong==sum)
document.write("Armstrong number");
else
document.write("Not an armstrong number");
</script>
</body>
</html>
```

f. 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. (Eg: If deposited amount is Rs.163, the output should be 1- 100's, 1-50's, 1- 10's, 1-2's & 1-1's)

```
<html>
<head>
  <title>Denomination Breakdown</title>
</head>
<body>
  <h2>Enter an Amount to Get Denomination Breakdown</h2>
  <script>
    // Function to calculate and display denomination breakdown
    function calculateDenominations(amount) {
      // Denominations in an array
      const denominations = [100, 50, 20, 10, 5, 2, 1];
      let result = "";

      // Loop through each denomination
      for (let i = 0; i < denominations.length; i++) {
        let count = Math.floor(amount / denominations[i]);
        if (count > 0) {
          result += `${count} - ${denominations[i]}'s, `;
        }
        amount %= denominations[i]; // Update the remaining amount
      }

      // Trim the last comma and space
      result = result.slice(0, -2);

      // Output the result
      alert(result);
    }

    // Prompt user for the deposited amount
    let amount = parseInt(prompt("Enter the amount deposited in Rs:"));

    // Call the function with the entered amount
</script>
<body>
</html>
```

9. Java Script Functions and Events

a. Design a appropriate function should be called to display

i. Factorial of that number

ii. Fibonacci series up to that number

iii. Prime numbers up to that number

iv. palindrome or not

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

i. Factorial of that number

ii. Fibonacci series up to that number

iii. Prime numbers up to that number

iv. Is it palindrome or not

```
<html>
  <head>
        <script>
//Find factorial of given number
         function factorial()
         {
           var n=parseInt(document.getElementById("number").value);
           fact=1;
           for(i=1;i<=n;i++)
                fact=fact*i;
           document.getElementById("result").innerHTML="factorial is:"+fact;
       }
//Find fibonacci within range
      function fib() {
        var n = parseInt(document.getElementById("number").value);
        result="fibonacci sequence is 0  1"
        f1=0,f2=1,f3=f1+f2;
        while(f3<=n)
         {
            result=result+ " " +f3;
            f1=f2;
            f2=f3;
            f3=f1+f2;
         }
        document.getElementById("result").innerHTML=result;
      }


//Find prime numbers within range
function prime()
      {
        var n = parseInt(document.getElementById("number").value);
         result = "Prime in given range ";
         for(i=1;i<=n;i++)
          {
           c=0;
           for(j=1;j<=i;j++)
           {
                if(i%j==0)
                    c++;
```

```
            }
           if(c==2)
             result=result+ " " +i;
          }
       document.getElementById("result").innerHTML=result;
      }
//Find whether number is palindrome or not
      function palindrome()
      {
       var n = parseInt(document.getElementById("number").value);
       pal=n;
       rev=0;
       while(n>0)
       {
            r=n%10;
            rev=rev*10+r;
            n=parseInt(n/10);
      }
      if(rev==pal)
          document.getElementById('result').innerHTML="Given number is Palindrome";
     else
        document.getElementById('result').innerHTML="Given number is not a Palindrome";
      }
       </script>
       <body>
        <h2>Number Operations</h2>
        <input type="text" name="number" id="number"/><br/>
        <input type="button" value="Fact" onClick="factorial()"/>
         <input type="button" value="Fib" onClick="fib()"/>
         <input type="button" value="Prime" onClick="prime()" />
         <input type="button" value="Palindrome" onClick="palindrome()" />
        <p id="result"></p>
       </body>
 </html>
```

c. Write a program to validate the following fields in a registration page i. Name (start with alphabet and followed by alphanumeric and the length should not be less than 6 characters) ii. Mobile (only numbers and length 10 digits)
iii. E-mail (should contain format like xxxxxxx@xxxxxx.xxx)

```
<html>
 <head>
<script>
       function validateForm()
       {
              var un =document.registration.uname.value;
              var letters=/^[a-zA-Z0-9_\s]+$/;
              var pass = document.registration.pwd.value;
                        var checkSpecial = /[*@!#%&]+/.test(pass);
                        var checkUpper = /[A-Z]+/.test(pass);
                        var checkLower = /[a-z]+/.test(pass);
var mobileno = document.registration.mobile.value;
                        var mno=/^[0-9]{10}$/;
```

```
 var mail = document.registration.email.value;



		if(un=="" || un=="null")
		{
			alert("Plz enter your user name");
			document.registration.uname.focus();
			return false;
		}
		if(un.length<6)
		{
			alert("Plz enter minimum 6 characters");
			document.registration.uname.focus();
			return false;
		}
		if(!un.match(letters))
		{
			alert("Plz enter valid user name");
			document.registration.uname.focus();
			return false;
		}
		if(pass=="" || pass=="null")
		{
			alert("please enter ur password");
			document.registration.pwd.focus();
			return false;
		}
		if(checkSpecial &&checkLower && checkUpper)
		{
			flag=1;
		}

		if(flag==0)
		{
			alert("Password should be the combination of atleast one lowercase letter, uppercase letter &
special symbols like ($, & #)");
			document.registration.pwd.focus();
			return false;
		}

		if(mobileno=="" || mobileno=="null")
		{
			alert("please enter ur mobile no");
			document.registration.mobile.focus();
			return false;
		}
		if(!mobileno.match(mno))
		{
			alert("Plz enter valid mobile numbe");
			document.registration.mobile.focus();
			return false;
		}
```

```
                        if(mail=="" || mail=="null")
                        {
                                alert("please enter your email");
                                document.registration.email.focus();
                                return false;
                        }
                        var atpos = mail.indexOf("@");
                        var dotpos = mail.lastIndexOf(".");
                        if (atpos < 1 || dotpos < atpos + 2 || dotpos + 2 >= mail.length)
                        {
                                alert("Not a valid e-mail address");
                                return false;
                        }

                alert("Successful");

        }
</script>
</head>
<body>
        <form name="registration" onsubmit="return validateForm();">
                Enter your name: <input type="text" name="uname"><br>
                Enter your password: <input type="text" name="pwd"><br>
Enter your mobile number: <input type="text" name="mobile"><br>
Enter your email: <input type="text" name="email"><br>
                <input type="submit"/>
        </form>
</body>
</html>
```