1. Write a HTML program, to explain the working of lists.

Program:

```
<!DOCTYPE html>
<html>
<head>
   <title>HTML List Types</title>
</head>
<body>

   <h2>1. Ordered List</h2>
   <ol>
      <li>Start</li>
      <li>Process</li>
      <li>End</li>
   </ol>

   <h2>2. Unordered List</h2>
   <ul>
      <li>Red</li>
      <li>Green</li>
      <li>Blue</li>
   </ul>

   <h2>3. Nested List</h2>
   <ul>
      <li>Fruits
         <ul>
            <li>Apple</li>
            <li>Mango</li>
         </ul>
      </li>
      <li>Vegetables
         <ul>
            <li>Carrot</li>
            <li>Beans</li>
```

```html
        </ul>
      </li>
  </ul>

  <h2>4. Ordered List in Unordered List</h2>
  <ul>
    <li>Registration Steps
      <ol>
        <li>Open site</li>
        <li>Fill form</li>
        <li>Submit</li>
      </ol>
    </li>
  </ul>

  <h2>5. Definition List</h2>
  <dl>
    <dt>HTML</dt>
    <dd>Markup language for web pages</dd>
    <dt>CSS</dt>
    <dd>Style sheet language</dd>
  </dl>

</body>
</html>
```

Output:

1. Ordered List

1. Start
2. Process
3. End

2. Unordered List

- Red
- Green
- Blue

3. Nested List

- Fruits
  - Apple
  - Mango
- Vegetables
  - Carrot
  - Beans

4. Ordered List in Unordered List

- Registration Steps
  1. Open site
  2. Fill form
  3. Submit

5. Definition List

HTML
  Markup language for web pages
CSS
  Style sheet language

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

Program:

```
<!DOCTYPE html>
<html>
<head>
   <title>Hyperlink Demo</title>
</head>
<body>

  <!-- Text Hyperlink -->
  <h2>1. Text Link</h2>
  <p>
    Visit our official website:
    <a href="https://www.aec.edu.in/" target="_blank">AEC Official Website</a>
  </p>

  <!-- Image Hyperlink -->
  <h2>2. Image Link</h2>
  <p>
    Click the image to open the AEC Gallery:
    <br>
    <a href="https://www.aec.edu.in/?p=Gallery" target="_blank">
      <img src="University_Logo.jpg" alt="AEC Logo" width="200">
    </a>
  </p>

</body>
</html>
```
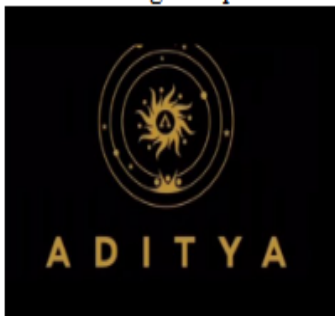
Output:

---

1. Text Link

Visit our official website: [AEC Official Website](#)

2. Image Link

Click the image to open the AEC Gallery:

3. 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

Program:

```html
<!DOCTYPE html>
<html>
<head>
   <title>Profile Images</title>
   <style>
     body {
        text-align: center;
        font-family: Arial;
        margin-top: 50px;
     }
   </style>
</head>
<body>

   <h2>My Profile</h2>
   <a href="http://www.linkedin.com/in/chandra-sekhar-andraju-12164b32a" target="_blank">
     <img src="my_image.jpg" alt="My Image">
   </a>

   <h2>My Friend's Profile</h2>
   <a href="www.linkedin.com/in/vamsi-pilli-795917331" target="_blank">
     <img src="friend_image.jpg" alt="Friend's Image">
   </a>
</body>
</html>
```
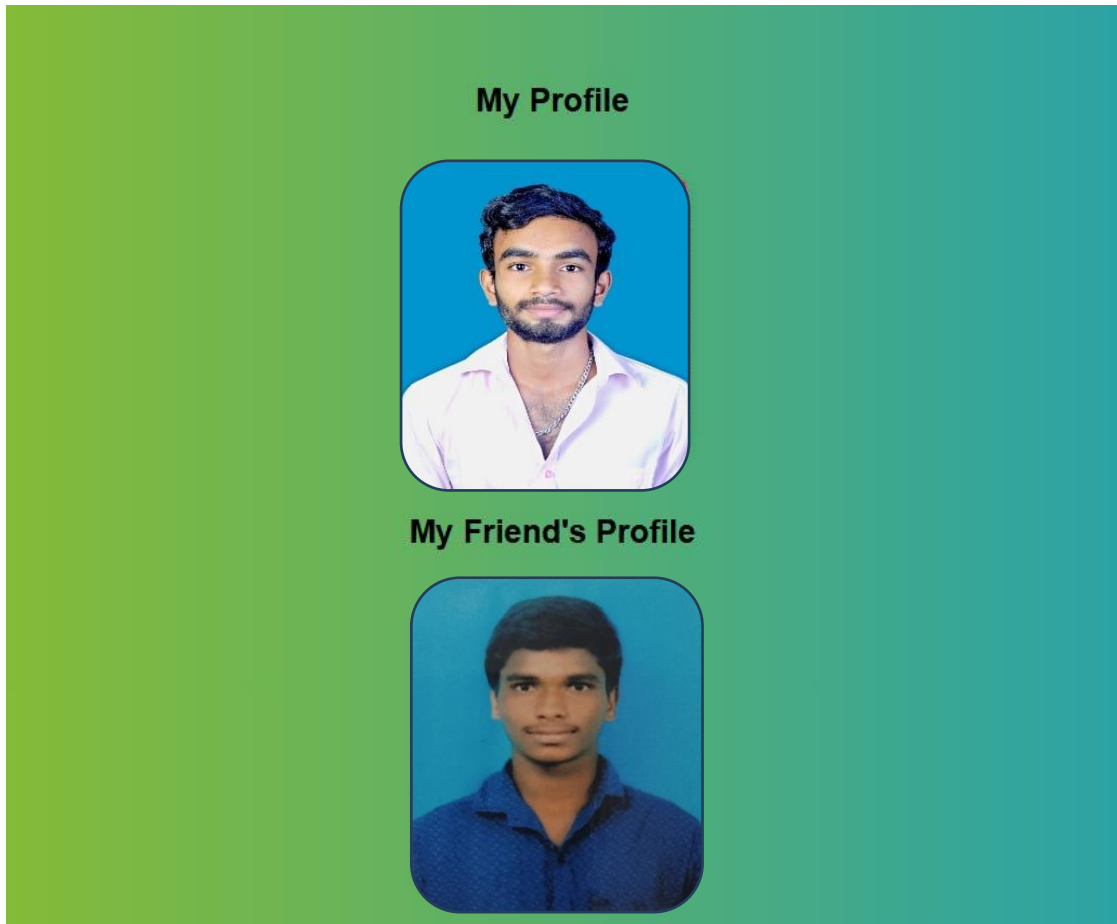
Output:

4.To create an image gallery using thumbnail images (100×100 pixels), where each image links to its full-sized version.

Program:

```
<!DOCTYPE html>
<html>
<head>
    <title>Thumbnail Gallery</title>
    <style>
        body {
            background: linear-gradient(to right, #e0f7fa, #fff);
            font-family: sans-serif;
            text-align: center;
            padding: 40px;
        }
    </style>
</head>
<body>

    <h2>Thumbnail Image Gallery</h2>

    <a href="image1.jpg" target="_blank">
        <img src="University_Logo.jpg" alt="Image 1">
    </a>
    <a href="image2.jpg" target="_blank">
        <img src="my_image.jpg" alt="Image 2">
    </a>
    <a href="image3.jpg" target="_blank">
        <img src="friend_image.jpg" alt="Image 3">
    </a>

</body>
</html>
```

Output:

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

Program:

```html
<!DOCTYPE html>
<html>
<head>
   <title>HTML Table Demo</title>
   <style>
     body {
        background: linear-gradient(to right, #f0f8ff, #e0ffff);
        font-family: Arial, sans-serif;
        text-align: center;
        padding-top: 50px;
     }
     table {
        margin: auto;
        border-collapse: collapse;
        background-color: #fff;
        box-shadow: 0 0 10px rgba(0,0,0,0.1);
     }
     th, td {
        border: 1px solid #999;
        padding: 10px 20px;
     }
     th {
        background-color: #87cefa;
     }
   </style>
</head>
<body>

   <h2>Student Marks Table</h2>
```

```html
<table>
  <tr>
    <th rowspan="2">Name</th>
    <th colspan="3">Marks</th>
  </tr>
  <tr>
    <th>Math</th>
    <th>Science</th>
    <th>English</th>
  </tr>
  <tr>
    <td>Ravi</td>
    <td>85</td>
    <td>90</td>
    <td>88</td>
  </tr>
  <tr>
    <td>Priya</td>
    <td>78</td>
    <td>84</td>
    <td>92</td>
  </tr>
</table>

</body>
</html>
```

Output:

Student Marks Table

Name	Marks		
	Math	Science	English
Ravi	85	90	88
Priya	78	84	92

6. 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.)

Program:

```html
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <title>Time Table</title>
   <style>
     body { background: #f0f8ff; font-family: sans-serif; text-align: center; }
     table { margin: auto; border-collapse: collapse; width: 95%; }
     th, td { border: 2px solid #000; padding: 8px; }
     caption { font-size: 20px; font-weight: bold; margin-bottom: 10px; }
     th { background: #ffe066; }
     .marquee { background: aqua; padding: 10px; font-size: 24px; }
     .footer { margin-top: 40px; width: 90%; display: flex; justify-content: space-between; padding: 0 40px; }
   </style>
</head>
<body>
   <marquee class="marquee" behavior="alternate" scrollamount="15">ADITYA UNIVERSITY</marquee>
   <hr>
   <table cellpadding="5" cellspacing="2">
     <caption>Department Of Computer Application</caption>
     <tr>
       <th rowspan="2">DAY</th>
       <th>1</th><th>2</th><th>3</th><th>4</th>
       <th>5</th><th>6</th><th>7</th><th>8</th>
     </tr>
     <tr>
```

```html
      <th>9:30</th><th>10:20</th><th>11:10</th><th>12:00</th>
      <th>1:00</th><th>1:50</th><th>2:40</th><th>3:30</th>
    </tr>
    <tr>
      <th>MON</th><td>CN</td><td colspan="2">MST LAB</td>
      <td
rowspan="5">LUNCH</td><td>NSD</td><td>CN</td><td
colspan="2">AEP</td>
    </tr>
    <tr>
      <th>TUE</th><td
colspan="2">ADSA</td><td>MST</td><td>CN</td>
      <td colspan="2">MLP</td><td>NSD</td>
    </tr>
    <tr>
      <th>WED</th><td colspan="3">CC</td><td
colspan="4">MLP LAB</td>
    </tr>
    <tr>
      <th>THU</th><td>SPTM</td><td colspan="2">MST
LAB</td><td>NSD</td>
      <td>SPTM</td><td colspan="2">PSA</td>
    </tr>
    <tr>
      <th>FRI</th><td>SPTM</td><td
colspan="2">ADSA</td><td>NSD</td>
      <td>SPTM</td><td colspan="2">COUN</td>
    </tr>
  </table>
  <div class="footer">
    <h3>Dept. Time Table Coordinator</h3>
    <h3>Head Of The Department</h3>
  </div>

</body>
</html>
```

Output:

ADITYA UNIVERSITY								
Department Of Computer Application								

DAY	1	2	3	4	5	6	7	8
	9:30	10:20	11:10	12:00	1:00	1:50	2:40	3:30
MON	CN	MST LAB			NSD	CN	AEP	
TUE	ADSA		MST		CN	MLP		NSD
WED	CC			LUNCH	MLP LAB			
THU	SPTM	MST LAB			NSD	SPTM	PSA	
FRI	SPTM	ADSA			NSD	SPTM	COUN	

Dept. Time Table Coordinator   **Head Of The Department**

7. Write a HTML program, to explain the working of forms by designing Registration form.

Program:

```html
<!DOCTYPE html>
<html>
<head>
   <title>Registration Form</title>
   <style>
     body {
        background: #f0f8ff;
        font-family: sans-serif;
        text-align: center;
     }
     table {
        margin: auto;
        padding: 20px;
        background: #fff;
        border-radius: 10px;
        box-shadow: 0 0 10px #888;
     }
     td { padding: 8px; }
     input, select, textarea {
        width: 100%;
        padding: 5px;
     }
   </style>
</head>
<body>

   <h2>Student Registration</h2>
   <form>
     <table>
        <tr><td>Name:</td><td><input type="text"
required></td></tr>
```

```html
        <tr><td>Password:</td><td><input type="password" required></td></tr>
        <tr><td>Mobile:</td><td><input type="number" required></td></tr>
        <tr><td>DOB:</td><td><input type="date" required></td></tr>
        <tr>
          <td>Gender:</td>
          <td><input type="radio" name="g"> Male <input type="radio" name="g"> Female</td>
        </tr>
        <tr>
          <td>Courses:</td>
          <td><input type="checkbox"> HTML <input type="checkbox"> CSS</td>
        </tr>
        <tr>
          <td>Department:</td>
          <td>
            <select>
              <option>MCA</option>
              <option>BCA</option>
              <option>B.Tech</option>
            </select>
          </td>
        </tr>
        <tr><td>Address:</td><td><textarea rows="2"></textarea></td></tr>
        <tr>
          <td colspan="2">
            <input type="submit"> <input type="reset">
          </td>
        </tr>
      </table>
    </form>
</body>
</html>
```

Output:

Student Registration

Name:	Jagadesh Kumar Sangani
Password:	••••••
Mobile:	07997930250
DOB:	07/06/2002 📅
Gender:	◉ Male ○ Female
Courses:	☑ HTML ☑ CSS
Department:	MCA
Address:	

Submit

Reset

8.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).

Program:

index.html – Main frameset page

```
<!DOCTYPE html>
<html>
<head>
    <title>Frames Example</title>
</head>
<frameset cols="33%,34%,33%" frameborder="1" noresize>
    <frame src="image.html" noresize>
    <frame src="paragraph.html" noresize>
    <frame src="link.html" noresize>
    <noframes>
        <body>Your browser does not support frames.</body>
    </noframes>
</frameset>
</html>
```

image.html

```
<!DOCTYPE html>
<html>
<body style="margin:0; text-align:center; background-color:
lightblue;">
    <img src="my_image.jpg" alt="Image 1" style="width:90%;
margin:10px;">
    <img src="University_2nd.jpg" alt="Image 2" style="width:90%;
margin:10px;">
</body>
</html>
```

paragraph.html

```html
<!DOCTYPE html>
<html>
<body style="font-family:sans-serif; padding:15px; background-color:
chartreuse;">
    <p>This is a paragraph inside the second frame. Frames divide a
page into multiple sections.</p>
</body>
</html>
```
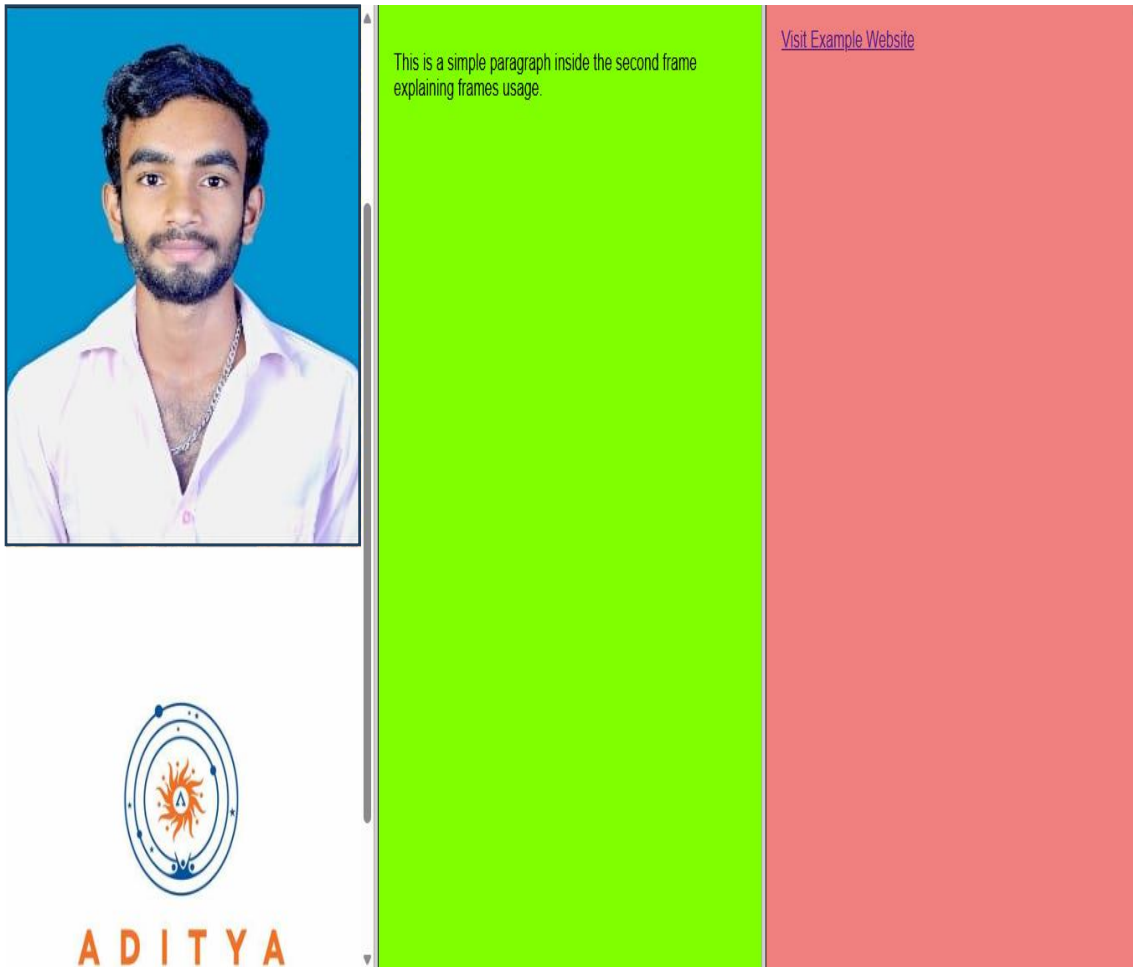
link.html

```html
<!DOCTYPE html>
<html>
<body style="font-family:sans-serif; padding:15px; background-color:
lightcoral;">
    <a href="https://www.example.com" target="_blank">Visit Example
Website</a>
</body>
</html>
```

Output:

This is a simple paragraph inside the second frame explaining frames usage.

Visit Example Website

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

Program:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>HTML5 Semantic Elements Example</title>
    <style>
        * {margin: 0; padding: 0; box-sizing: border-box;}
        body {font-family: Arial, sans-serif; background: #e6ecf0;}
        header, footer, nav, main, section, article, aside {margin: 10px;
border-radius: 8px;}
        header {background: #4CAF50; color: #fff; text-align: center;
padding: 20px;}
        nav {background: #333; text-align: center; padding: 10px;}
        nav a {color: #fff; margin: 0 15px; text-decoration: none; font-
weight: bold;}
        nav a:hover {color: #ffd700;}
        main {display: flex; flex-wrap: wrap; padding: 10px;}
        section, aside {padding: 15px; box-shadow: 0 2px 5px
rgba(0,0,0,0.1);}
        section {flex: 2; background: #fff;}
        aside {flex: 1; background: #f9d976;}
        article {background: #e0f7fa; padding: 15px; border-radius: 8px;}
        figure {text-align: center; margin-top: 15px;}
        figcaption {font-size: 0.9em; color: #555;}
        .highlight {color: #ffeb3b; font-weight: bold;}
        footer {background: #2c3e50; color: #fff; text-align: center;
padding: 15px;}
    </style>
</head>
<body>
```

```html
    <header>
        <h1>My Personal Blog</h1>
        <p>Welcome to my website about <span class="highlight">web
development</span> and design.</p>
    </header>
    <nav>
        <a href="#">Home</a>
        <a href="#">Articles</a>
        <a href="#">About</a>
    </nav>
    <main>
        <section>
            <article>
                <h2>Learning HTML5</h2>
                <p>HTML5 introduces many semantic elements like
<code>&lt;article&gt;</code>, <code>&lt;section&gt;</code>, and
<code>&lt;nav&gt;</code>.</p>
                <figure>
                    <img src="html_logo.jpg" alt="HTML5 Logo"
width="150">
                    <figcaption>The official HTML5 logo</figcaption>
                </figure>
            </article>
        </section>
        <aside>
            <h3>Quick Tip</h3>
            <p>Use semantic tags to improve your website's accessibility
and SEO.</p>
        </aside>
    </main>
    <footer>
        <p>&copy; 2025 My Personal Blog. All rights reserved.</p>
    </footer>

</body>
</html>
```

Output:
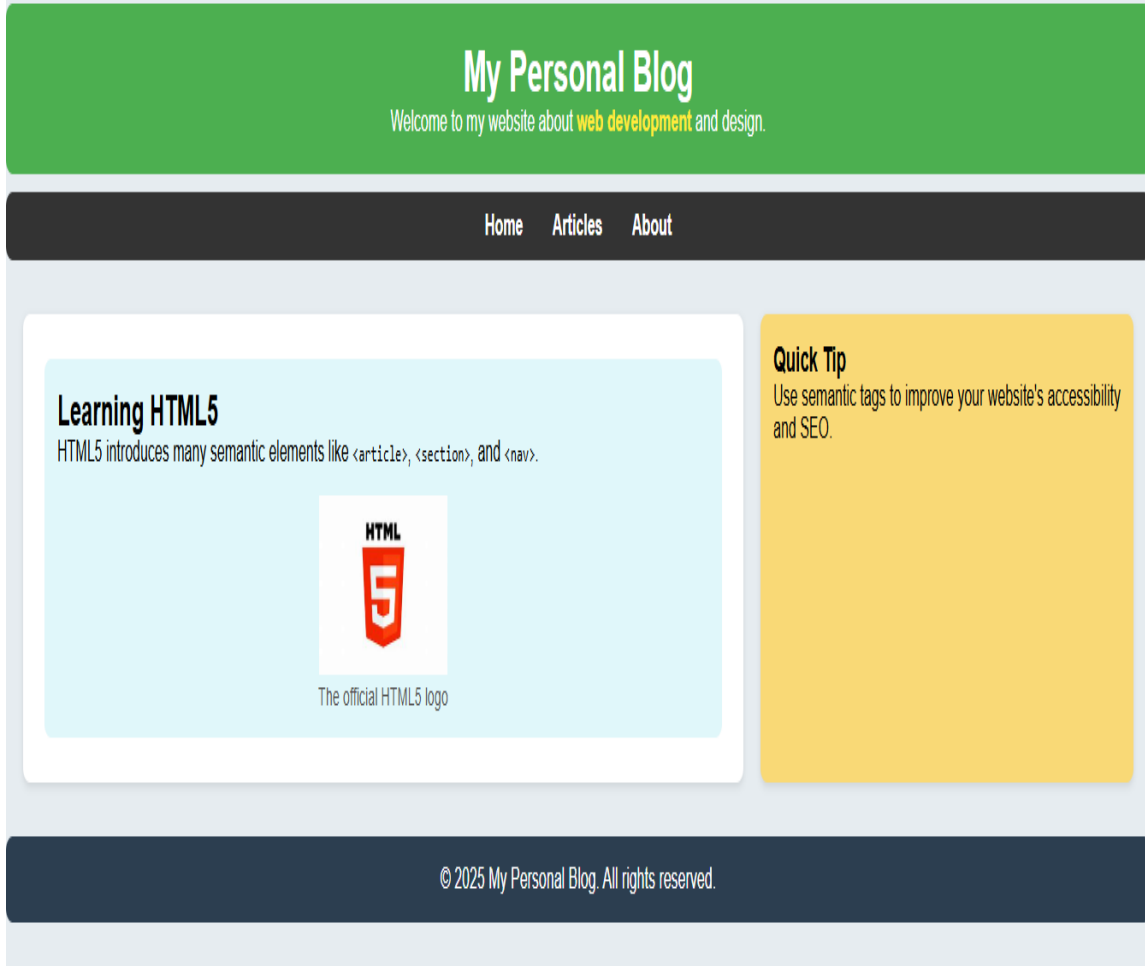
10.   Write a HTML program, to embed audio and video into HTML web page

Program:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Embed Audio and Video</title>
    <style>
        body { font-family: Arial, sans-serif; background-color: #f0f0f0;
text-align: center; padding: 40px; }
        h1 { color: #333; }
        audio, video { margin-top: 20px; width: 80%; max-width: 600px;
border: 2px solid #ccc; border-radius: 10px; }
    </style>
</head>
<body>

    <h1>Audio and Video Embed Example</h1>

    <h2>Audio</h2>
    <audio controls>
        <source src="sample-audio.mp3" type="audio/mpeg">
        Your browser does not support the audio element.
    </audio>

    <h2>Video</h2>
    <video controls>
        <source src="sample-video.mp4" type="video/mp4">
        Your browser does not support the video tag.
    </video>

</body>
</html>
```
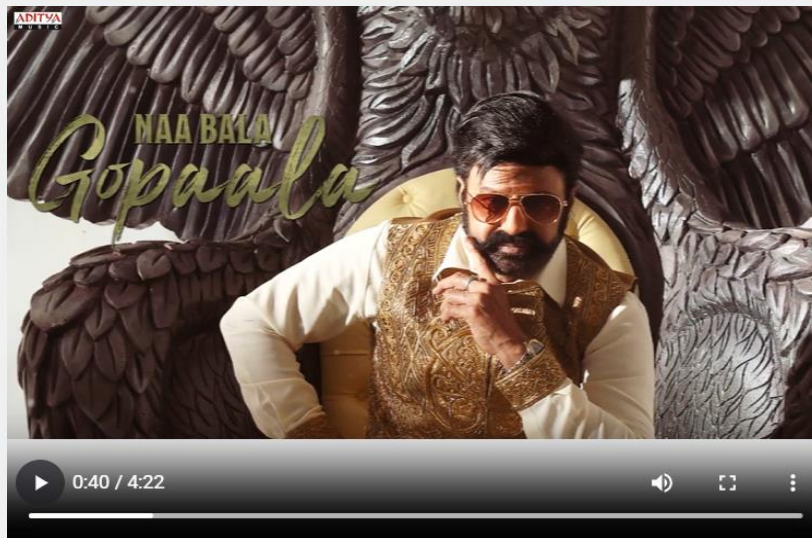
Output:

11.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).

Program:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <title>CSS Styling Example</title>

   <!-- Internal CSS -->
   <style>
     body {
       background-color: #eef; /* selector: body, property:
background-color, value: #eef */
       font-family: Arial, sans-serif;
     }
     .internal-text {
       color: darkblue; /* selector: .internal-text, property: color,
value: darkblue */
       font-size: 20px;
     }
   </style>

   <!-- Link to External CSS -->
   <link rel="stylesheet" href="style.css">
</head>
<body>

   <!-- Inline CSS -->
   <h1 style="color: red; background-color: yellow;">This is an Inline
Styled Heading</h1>

   <!-- Internal CSS -->
```

```html
    <p class="internal-text">This paragraph is styled using Internal
CSS.</p>

    <!-- External CSS -->
    <p class="external-text">This paragraph is styled using External
CSS.</p>

</body>
</html>
```

style.css

```css
/* selector: .external-text, property: color, value: green */
.external-text {
    color: green;
    font-style: italic;
    font-size: 18px;
}
```

Output:

This is an Inline Styled Heading

This paragraph is styled using Internal CSS.

This paragraph is styled using External CSS.

12. 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

Program:

```html
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <title>CSS Selector Examples</title>
   <style>
     /* i. SIMPLE SELECTORS */

     /* Element selector */
     h1 {
        color: teal;
     }

     /* ID selector */
     #main-title {
        text-decoration: underline;
     }

     /* Class selector */
     .highlight {
        background-color: yellow;
     }
```

```css
/* Group selector */
h2, p {
    font-family: Arial, sans-serif;
}

/* Universal selector */
* {
    margin: 5px;
}

/* ii. COMBINATOR SELECTORS */

/* Descendant selector */
div p {
    color: green;
}

/* Child selector */
ul > li {
    font-weight: bold;
}

/* Adjacent sibling selector */
h3 + p {
    color: darkorange;
}

/* General sibling selector */
h4 ~ p {
    font-style: italic;
}
```

```css
    /* iii. PSEUDO-CLASS SELECTOR */
    a:hover {
      color: red;
    }

    li:first-child {
      color: blue;
    }

    /* iv. PSEUDO-ELEMENT SELECTOR */
    p::first-letter {
      font-size: 24px;
      color: purple;
    }

    p::after {
      content: " □";
    }

    /* v. ATTRIBUTE SELECTOR */
    input[type="text"] {
      border: 2px solid blue;
    }

    a[target="_blank"] {
      background-color: #ddd;
    }
  </style>
</head>
```

```html
<body>

  <h1 id="main-title">CSS Selector Demo</h1>

  <h2 class="highlight">Simple Selectors</h2>
  <p>This is a paragraph showing element and class selector.</p>

  <div>
    <p>This is inside a div — descendant selector.</p>
  </div>

  <ul>
    <li>First item (child selector)</li>
    <li>Second item</li>
  </ul>

  <h3>Sibling Selector Demo</h3>
  <p>This paragraph follows an h3 — adjacent sibling selector.</p>

  <h4>General Sibling Example</h4>
  <p>This follows h4 — general sibling selector.</p>
  <p>Another paragraph showing general sibling effect.</p>

  <a href="https://example.com" target="_blank">Hover me (pseudo-class + attribute selector)</a><br><br>

  <input type="text" placeholder="Type here (attribute selector)">

</body>
</html>
```

Output:

CSS Selector Demo

Simple Selectors

This is a paragraph showing element and class selector. ☀

This is inside a div — descendant selector. ☀

First item (child selector)

Second item

Sibling Selector Demo

This paragraph follows an h3 — adjacent sibling selector. ☀

General Sibling Example

This follows h4 — general sibling selector. ☀

Another paragraph showing general sibling effect. ☀

Hover me (pseudo-class + attribute selector)

Type here (attribute selector)

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

Program:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <title>CSS Color Reference Methods</title>
   <style>
     /* 1. Named Color */
     .named-color {
        color: blue;
     }

     /* 2. Hexadecimal Color */
     .hex-color {
        color: #FF5733;
     }

     /* 3. RGB Color */
     .rgb-color {
        color: rgb(255, 99, 71);
     }

     /* 4. RGBA Color (with transparency) */
     .rgba-color {
        color: rgba(255, 0, 0, 0.5);
     }
```

```css
/* 5. HSL Color */
    .hsl-color {
        color: hsl(120, 100%, 25%);
    }

    /* 6. HSLA Color (with transparency) */
    .hsla-color {
        color: hsla(240, 100%, 50%, 0.5);
    }
</style>
</head>
<body style="font-family: sans-serif; line-height: 1.8;">

    <h2 class="named-color">1. Named Color: blue</h2>
    <h2 class="hex-color">2. Hexadecimal Color: #FF5733</h2>
    <h2 class="rgb-color">3. RGB Color: rgb(255, 99, 71)</h2>
    <h2 class="rgba-color">4. RGBA Color: rgba(255, 0, 0, 0.5)</h2>
    <h2 class="hsl-color">5. HSL Color: hsl(120, 100%, 25%)</h2>
    <h2 class="hsla-color">6. HSLA Color: hsla(240, 100%, 50%,
0.5)</h2>

</body>
</html>
```

Output:

1. Named Color: blue

2. Hexadecimal Color: #FF5733

3. RGB Color: rgb(255, 99, 71)

4. RGBA Color: rgba(255, 0, 0, 0.5)

5. HSL Color: hsl(120, 100%, 25%)

6. HSLA Color: hsla(240, 100%, 50%, 0.5)

14.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.

Program:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Background Image Halfway Down</title>
  <style>
    body {
      margin: 0;
      font-family: Arial, sans-serif;
      height: 2000px; /* Add height to enable scrolling */
    }

    .bg-half {
      position: fixed;
      top: 50%;
      left: 0;
      width: 100%;
      height: 300px;
      background-image: url("your-image.jpg"); /* Replace with your image */
      background-repeat: no-repeat;
      background-position: center;
      background-size: cover;
      transform: scaleX(-1); /* Flip horizontally */
      z-index: -1;
      pointer-events: none; /* Allow interaction with page content */
    }
```

```html
    .content {
        padding: 20px;
    }
    </style>
</head>
<body>

    <div class="bg-half"></div>

    <div class="content">
       <h1>Scroll Down</h1>
       <p>This is a demonstration of a background image placed halfway
down the page, flipped horizontally, and fixed in place while
scrolling.</p>
       <p>Keep scrolling to see the effect...</p>
    </div>

</body>
</html>
```

Output:

Scroll Down

This is a demonstration of a background image placed halfway down the page, flipped horizontally, and fixed in place while scrolling.

Keep scrolling to see the effect...

15. 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

Program:

```html
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <title>CSS Font & Text Styling Showcase</title>
   <style>
     body {
        font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
        background: linear-gradient(to right, #e0f7fa, #fff3e0);
        margin: 0;
        padding: 40px;
        color: #333;
     }

     header {
        background-color: #00acc1;
        color: white;
        padding: 20px;
        border-radius: 10px;
        text-align: center;
     }

     header h1 {
        font-size: 36px;                  /* i. font-size */
        font-weight: 700;                 /* ii. font-weight */
        font-style: italic;               /* iii. font-style */
        text-transform: uppercase;    /* v. text-transform */
        text-decoration: underline; /* iv. text-decoration */
        margin: 0;
     }
```

```css
    .content {
        background-color: #ffffffaa;
        padding: 25px;
        margin-top: 30px;
        border-radius: 12px;
        box-shadow: 0 4px 10px rgba(0,0,0,0.1);
    }

    p {
        font-size: 18px;
        line-height: 1.6;
        text-align: justify;           /* vi. text-align */
        text-transform: capitalize; /* v. text-transform */
    }

    .highlight {
        font-weight: 600;
        font-style: oblique;
        color: #d84315;
        text-decoration: line-through;
    }

    footer {
        margin-top: 40px;
        text-align: center;
        font-size: 14px;
        color: #555;
        text-decoration: overline;
    }
    </style>
</head>
<body>

    <header>
        <h1>css text & font styling magic</h1>
    </header>
```

```html
<div class="content">
    <p>
        Welcome to the world of <span class="highlight">beautiful
typography</span> using CSS! With just a few lines of code,
        you can transform ordinary text into a stunning design element.
Explore how text size, weight, style,
        alignment, decoration, and transformation can enhance your
content visually and semantically.
    </p>
</div>

<footer>
    © 2025 Text Styling by CSS – Practice Makes Perfect
</footer>

</body>
</html>
```

Output:

CSS TEXT & FONT STYLING MAGIC

Welcome To The World Of *Beautiful Typography* Using CSS! With Just A Few Lines Of Code, You Can Transform Ordinary Text Into A Stunning Design Element. Explore How Text Size, Weight, Style, Alignment, Decoration, And Transformation Can Enhance Your Content Visually And Semantically.

16.Write a program, to explain the importance of CSS Box model using
i. Content ii. Border iii. Margin iv. padding

Program:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>CSS Box Model Demo</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      background-color: #f5f5f5;
      padding: 40px;
      text-align: center;
    }
    h1 {
      color: #333;
    }

    .box-model-container {
      margin: 50px auto;
      width: 300px;
      background-color: #fff;
    }
    .box {
      margin: 30px;                        /* iv. Margin (outside spacing) */
      border: 5px solid #4caf50;     /* ii. Border */
      padding: 20px;                      /* iii. Padding (inside spacing) */
      background-color: #c8e6c9;     /* Content background */
      color: #000;
    }
```

```
    .box span {
        font-weight: bold;
        color: #d84315;
    }

    .legend {
        text-align: left;
        margin-top: 40px;
        display: inline-block;
        background: #fff3e0;
        padding: 15px;
        border-radius: 10px;
        box-shadow: 0 0 10px rgba(0,0,0,0.1);
    }

    .legend li {
        margin: 10px 0;
    }

    .legend li span {
        font-weight: bold;
        color: #388e3c;
    }
  </style>
</head>
<body>

  <h1>CSS Box Model Example</h1>

  <div class="box-model-container">
    <div class="box">
      <p><span>Content:</span> This is where your text or images
live!</p>
    </div>
  </div>
```

```html
    <ul class="legend">
      <li><span>Content:</span> Actual text or image inside the
box.</li>
      <li><span>Padding:</span> Space between content and border.
(Green area)</li>
      <li><span>Border:</span> Line that surrounds the padding and
content. (Thick green line)</li>
      <li><span>Margin:</span> Space outside the border, separating
this box from others.</li>
    </ul>

</body>
</html>
```
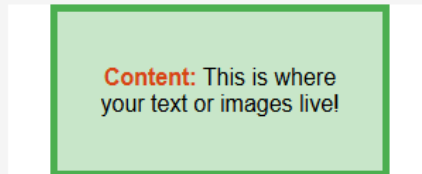
Output:

CSS Box Model Example

Content: This is where your text or images live!

- **Content:** Actual text or image inside the box.
- **Padding:** Space between content and border. (Green area)
- **Border:** Line that surrounds the padding and content. (Thick green line)
- **Margin:** Space outside the border, separating this box from others.

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

Program:

```html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Internal and External JavaScript</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      text-align: center;
      margin-top: 50px;
    }

    button {
      margin: 10px;
      padding: 10px 20px;
      font-size: 16px;
      cursor: pointer;
    }
  </style>
</head>
<body>

  <h1>JavaScript Integration Example</h1>

  <!-- Button to trigger internal JavaScript -->
  <button onclick="internalFunction()">Run Internal JS</button>

  <!-- Button to trigger external JavaScript -->
  <button onclick="externalFunction()">Run External JS</button>

  <!-- Internal JavaScript -->
  <script>
```

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

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

</body>
</html>
```

External JavaScript File (script.js)

```
function externalFunction() {
    alert("Hello from External JavaScript!");
}
```

Output:

JavaScript Integration Example

[Run Internal JS] [Run External JS]

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

Program:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>JavaScript Output Methods</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 50px;
      background-color: #f0f8ff;
    }

    button {
      padding: 10px 20px;
      margin: 10px;
      font-size: 16px;
      cursor: pointer;
    }

    #output {
      margin-top: 20px;
      padding: 10px;
      border: 1px solid #ccc;
      background-color: #e6f7ff;
    }
  </style>
</head>
```

```html
<body>

    <h1>Different Output Methods in JavaScript</h1>

    <button onclick="showAlert()">Alert Output</button>
    <button onclick="writeDocument()">Document Write</button>
    <button onclick="writeConsole()">Console Log</button>
    <button onclick="writeInnerHTML()">innerHTML Output</button>

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

    <script>
      function showAlert() {
         alert("This is an alert box!");
      }

      function writeDocument() {
         document.write("This text is written using document.write().");
      }

      function writeConsole() {
         console.log("This message is logged in the console.");
      }

      function writeInnerHTML() {
         document.getElementById("output").innerHTML = "This
content is added using innerHTML.";
      }
    </script>

</body>
</html>
```

Output:

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

Program:

```
<!DOCTYPE html>
<html>
<head>
   <title>Input Methods</title>
   <style>
      body { font-family: sans-serif; background: #f2f2f2; padding:
20px; }
      input, textarea, select, button { margin: 5px 0; width: 100%;
padding: 8px; }
      div { background: #fff; padding: 15px; margin-bottom: 10px;
border-radius: 6px; }
   </style>
</head>
<body>

   <h2>Input Methods in JavaScript</h2>

   <div>
      <button onclick="let name=prompt('Your name?'); alert('Hi, ' +
name);">Prompt Input</button>
   </div>

   <div>
      <input type="text" id="txt" placeholder="Enter name">
      <button onclick="alert('Hello ' + txt.value)">Show</button>
   </div>
```

```html
  <div>
    <textarea id="ta" placeholder="Your feedback..."></textarea>
    <button onclick="alert('Feedback: ' + ta.value)">Submit</button>
  </div>

  <div>
    <label><input type="checkbox" id="c1"> HTML</label>
    <label><input type="checkbox" id="c2"> CSS</label>
    <button onclick="alert('Skills: ' + (c1.checked?'HTML ':'') +
(c2.checked?'CSS':''))">Show Skills</button>
  </div>

  <div>
    <label><input type="radio" name="g" value="Male">
Male</label>
    <label><input type="radio" name="g" value="Female">
Female</label>
    <button onclick="alert('Gender: ' +
document.querySelector('input[name=g]:checked')?.value)">Show
Gender</button>
  </div>

  <div>
    <select id="country">

<option>India</option><option>USA</option><option>UK</option>
    </select>
    <button onclick="alert('Country: ' +
country.value)">Select</button>
  </div>

</body>
</html>
```

20.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.

Program:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <title>Voter Eligibility</title>
   <style>
     body {
        font-family: Arial, sans-serif;
        background: #f2f2f2;
        padding: 40px;
        text-align: center;
     }
     table {
        margin: 20px auto;
        border-collapse: collapse;
        width: 60%;
        background: #fff;
     }
     th, td {
        border: 1px solid #ccc;
        padding: 12px;
     }
     th {
        background-color: #4CAF50;
        color: white;
     }
     td {
        font-weight: bold;
     }
   </style>
</head>
```
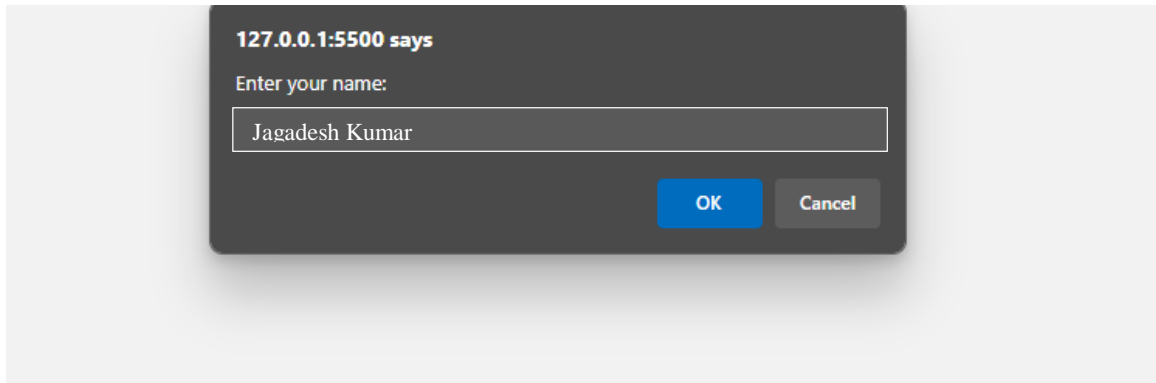
```html
<body>

<h2>Voter Eligibility Check</h2>

<table id="resultTable" style="display:none;">
  <tr><th>Name</th><th>Age</th><th>Status</th></tr>
  <tr>
    <td id="nameCell"></td>
    <td id="ageCell"></td>
    <td id="statusCell"></td>
  </tr>
</table>

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

  if (name && !isNaN(age)) {
    document.getElementById("nameCell").innerText = name;
    document.getElementById("ageCell").innerText = age;
    document.getElementById("statusCell").innerText = (age >= 18) ?
"Eligible to Vote" : "Not Eligible";
    document.getElementById("resultTable").style.display = "table";
  } else {
    alert("Invalid input! Please reload and enter correct details.");
  }
</script>

</body>
</html>
```

Output:

Voter Eligibility Check

Name	Age	Status
Jagadesh Kumar	23	Eligible to Vote

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

Program:
```html
<!DOCTYPE html>
<html>
<head>
   <title>Document Object Demo</title>
</head>
<body style="font-family: Arial; padding: 20px;">
   <h2>Document Object Demo</h2>
   <button onclick="showDetails()">Show Document Details</button>
   <div id="output" style="margin-top: 20px; border: 1px solid #ccc; padding: 10px;"></div>

   <script>
     function showDetails() {
        let info = `
           <strong>Title:</strong> ${document.title}<br>
           <strong>URL:</strong> ${document.URL}<br>
           <strong>Last Modified:</strong> ${document.lastModified}<br>
           <strong>Domain:</strong> ${document.domain}<br>
           <strong>Body Inner HTML:</strong><br>
${document.body.innerHTML.slice(0, 100)}...
        `;
        document.getElementById("output").innerHTML = info;
        document.body.style.backgroundColor = "#f0f8ff"; // using document.body
     }
   </script>
</body>
</html>
```
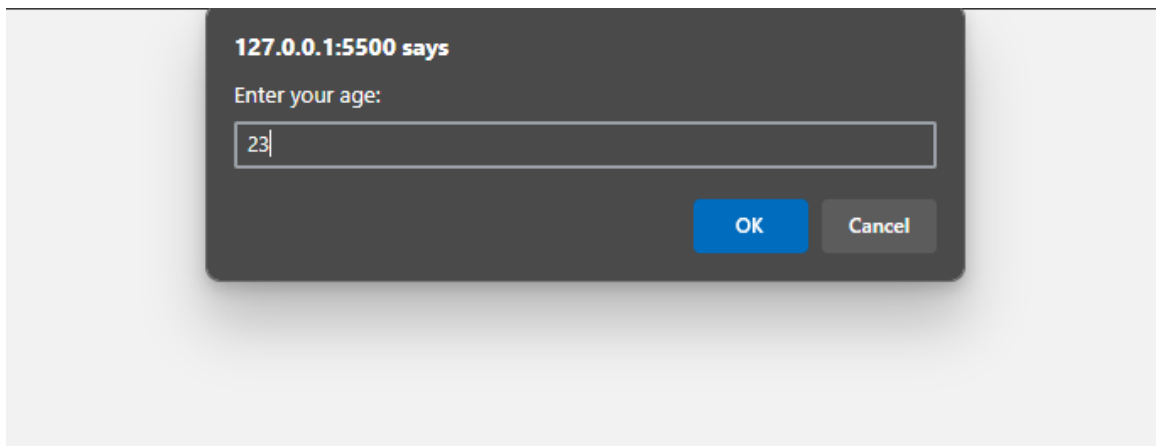
Output:

Document Object Demo

Show Document Details

Title: Document Object Demo
URL: http://127.0.0.1:5500/MERN_MANUAL/UNIT-3/5th.html
Last Modified: 05/18/2025 12:55:32
Domain: 127.0.0.1
Body Inner HTML:

Document Object Demo

Show Document Details ...

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

Program:

```html
<!DOCTYPE html>
<html>
<head>
   <title>Window Object Demo</title>
</head>
<body style="font-family: Arial; padding: 20px;">
   <h2>Window Object Demo</h2>
   <button onclick="showInfo()">Show Window Info</button>
   <button onclick="openNewWindow()">Open New
Window</button>
   <div id="output" style="margin-top: 20px;"></div>
   <script>
     function showInfo() {
        let info = `
          <strong>Window Width:</strong>
${window.innerWidth}px<br>
          <strong>Window Height:</strong>
${window.innerHeight}px<br>
          <strong>Location:</strong> ${window.location.href}<br>
          <strong>Browser Name:</strong>
${window.navigator.appName}
        `;
        document.getElementById("output").innerHTML = info;
     }

     function openNewWindow() {
        window.open("https://www.example.com", "_blank",
"width=400,height=300");
     }
   </script>
</body>
</html>
```
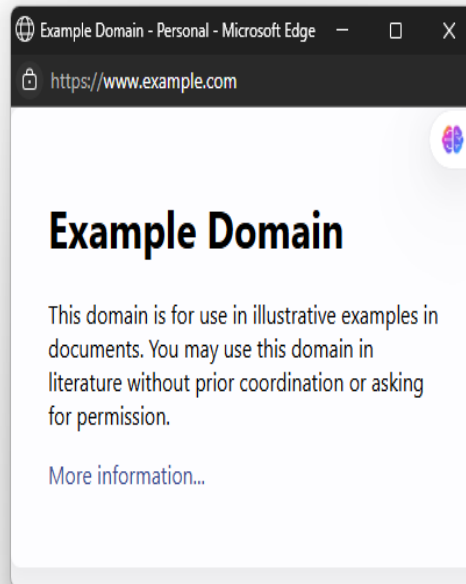
Output:

Window Object Demo

[Show Window Info] [Open New Window]

Window Width: 1810px
Window Height: 917px
Location: http://127.0.0.1:5500/MERN_MANUAL/UNIT-3/6th.html
Browser Name: Netscape

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

Program:

```html
<!DOCTYPE html>
<html>
<head>
    <title>Array Methods Demo</title>
</head>
<body style="font-family: Arial; padding: 20px;">
    <h2>Array Object Demo</h2>
    <button onclick="showArrayInfo()">Show Array Info</button>
    <div id="output" style="margin-top: 20px;"></div>
    <script>
        function showArrayInfo() {
            let fruits = ["Apple", "Banana", "Cherry"];
            // Use of Array methods and properties
            fruits.push("Mango");              // Add element
            let removed = fruits.pop();        // Remove last element
            fruits.unshift("Grapes");          // Add to beginning
            let joined = fruits.join(", "); // Join elements

            let info = `
                <strong>Original Array:</strong> Apple, Banana, Cherry<br>
                <strong>After Push & Pop:</strong> ${fruits}<br>
                <strong>Removed Element:</strong> ${removed}<br>
                <strong>Joined:</strong> ${joined}<br>
                <strong>Length:</strong> ${fruits.length}<br>
                <strong>Sorted:</strong> ${fruits.sort().join(", ")}
            `;
            document.getElementById("output").innerHTML = info;
        }
    </script>
</body>
</html>
```

Output:

Array Object Demo

Show Array Info

Original Array: Apple, Banana, Cherry
After Push & Pop: Grapes,Apple,Banana,Cherry
Removed Element: Mango
Joined: Grapes, Apple, Banana, Cherry
Length: 4
Sorted: Apple, Banana, Cherry, Grapes

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

Program:

```html
<!DOCTYPE html>
<html>
<head>
   <title>Math Object Demo</title>
</head>
<body style="font-family: Arial; padding: 20px;">
   <h2>Math Object Demo</h2>
   <button onclick="showMathOperations()">Show Math Info</button>
   <div id="output" style="margin-top: 20px;"></div>
   <script>
     function showMathOperations() {
        let number = 7.65;
        let info = `
           <strong>Math.PI:</strong> ${Math.PI}<br>
           <strong>Math.E:</strong> ${Math.E}<br>
           <strong>Round(${number}):</strong>
${Math.round(number)}<br>
           <strong>Floor(${number}):</strong>
${Math.floor(number)}<br>
           <strong>Ceil(${number}):</strong>
${Math.ceil(number)}<br>
           <strong>Square Root of 25:</strong> ${Math.sqrt(25)}<br>
           <strong>2 Power 3:</strong> ${Math.pow(2, 3)}<br>
           <strong>Random (0 to 1):</strong>
${Math.random().toFixed(3)}<br>
           <strong>Max of 10, 25, 5:</strong> ${Math.max(10, 25,
5)}<br>
           <strong>Min of 10, 25, 5:</strong> ${Math.min(10, 25, 5)}
`;
        document.getElementById("output").innerHTML = info;}
   </script>
```

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

Output:

Math Object Demo

Show Math Info

Math.PI: 3.141592653589793
Math.E: 2.718281828459045
Round(7.65): 8
Floor(7.65): 7
Ceil(7.65): 8
Square Root of 25: 5
2 Power 3: 8
Random (0 to 1): 0.929
Max of 10, 25, 5: 25
Min of 10, 25, 5: 5

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

Program:

```html
<!DOCTYPE html>
<html>
<head>
   <title>String Object Demo</title>
</head>
<body style="font-family: Arial; padding: 20px;">
   <h2>String Object Demo</h2>
   <button onclick="showStringMethods()">Show String Info</button>
   <div id="output" style="margin-top: 20px;"></div>
   <script>
     function showStringMethods() {
        let str = " Hello JavaScript! ";
        let info = `
           <strong>Original String:</strong> "${str}"<br>
           <strong>Length:</strong> ${str.length}<br>
           <strong>Trimmed:</strong> "${str.trim()}"<br>
           <strong>Uppercase:</strong> ${str.toUpperCase()}<br>
           <strong>Lowercase:</strong> ${str.toLowerCase()}<br>
           <strong>Substring (7, 17):</strong> ${str.substring(7, 17)}<br>
           <strong>Index of 'Java':</strong> ${str.indexOf("Java")}<br>
           <strong>Replace 'JavaScript' with 'HTML':</strong> ${str.replace("JavaScript", "HTML")}<br>
           <strong>Split by space:</strong> ${str.trim().split(" ")}`;
        document.getElementById("output").innerHTML = info;
     }
   </script>
</body>
```

</html>

Output:

String Object Demo

Show String Info

Original String: " Hello JavaScript! "
Length: 19
Trimmed: "Hello JavaScript!"
Uppercase: HELLO JAVASCRIPT!
Lowercase: hello javascript!
Substring (7, 17): JavaScript
Index of 'Java': 7
Replace 'JavaScript' with 'HTML': Hello HTML!
Split by space: Hello,JavaScript!

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

Program:

```html
<!DOCTYPE html>
<html>
<head>
   <title>RegExp Demo</title>
</head>
<body style="font-family: Arial; padding: 20px;">
   <h2>RegExp Object Demo</h2>
   <button onclick="testRegex()">Run Regex Test</button>
   <div id="output" style="margin-top: 20px;"></div>

   <script>
     function testRegex() {
        const text = "The rain in Spain stays mainly in the plain.";
        const pattern = /ain/g;

        const result = `
           <strong>Original Text:</strong> ${text}<br>
           <strong>Pattern:</strong> /ain/g<br>
           <strong>Test 'Spain':</strong> ${/Spain/.test(text)}<br>
           <strong>Match all 'ain':</strong> ${text.match(pattern)}<br>
           <strong>First index of 'ain':</strong>
${text.search(/ain/)}<br>
           <strong>Replace 'ain' with '**':</strong>
${text.replace(pattern, "**")}<br>
        `;

        document.getElementById("output").innerHTML = result;
     }
   </script>
</body>
</html>
```

Output:

RegExp Object Demo

Run Regex Test

Original Text: The rain in Spain stays mainly in the plain.
Pattern: /ain/g
Test 'Spain': true
Match all 'ain': ain,ain,ain,ain
First index of 'ain': 5
Replace 'ain' with '':** The r** in Sp** stays m**ly in the pl**.

27. Write a program using date object properties and methods

Program:

```
<!DOCTYPE html>
<html>
<head>
    <title>Date Object Demo</title>
</head>
<body style="font-family: Arial; padding: 20px;">
    <h2>Date Object Demo</h2>
    <button onclick="showDateInfo()">Show Date Info</button>
    <div id="output" style="margin-top: 20px;"></div>

    <script>
        function showDateInfo() {
            const now = new Date();
            const output = `
                <strong>Full Date & Time:</strong> ${now} <br>
                <strong>Year:</strong> ${now.getFullYear()} <br>
                <strong>Month (0-11):</strong> ${now.getMonth()} <br>
                <strong>Date:</strong> ${now.getDate()} <br>
                <strong>Day (0-6):</strong> ${now.getDay()} <br>
                <strong>Hours:</strong> ${now.getHours()} <br>
                <strong>Minutes:</strong> ${now.getMinutes()} <br>
                <strong>Seconds:</strong> ${now.getSeconds()} <br>
                <strong>Milliseconds:</strong> ${now.getMilliseconds()}
<br>
                <strong>Time in ms since Jan 1, 1970:</strong>
${now.getTime()}
            `;

            document.getElementById("output").innerHTML = output;
        }
    </script>
```

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

Output:

Date Object Demo

Show Date Info

Full Date & Time: Sun May 18 2025 13:26:35 GMT+0530 (India Standard Time)
Year: 2025
Month (0-11): 4
Date: 18
Day (0-6): 0
Hours: 13
Minutes: 26
Seconds: 35
Milliseconds: 598
Time in ms since Jan 1, 1970: 1747554995598

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

Program:

```html
<!DOCTYPE html>
<html>
<head>
  <title>User-Defined Object Example</title>
</head>
<body style="font-family: Arial; padding: 20px;">
  <h2>
      User-Defined Object using Constructor, Methods & Accessors
  </h2>
  <button onclick="showPerson()">Show Person Info</button>
  <div id="info" style="margin-top: 20px;"></div>

  <script>
    // Constructor Function
    function Person(name, age) {
      this.name = name;
      this.age = age;

      // Method
      this.greet = function() {
          return `Hello, I'm ${this.name} and I'm ${this.age} years
old.`;
      };

      // Accessor - getter
      this.getAgeInMonths = function() {
          return this.age * 12;
      };
```

```
        // Accessor - setter
        this.setName = function(newName) {
            this.name = newName;
        };
    }

    // Create object using constructor
    const person1 = new Person("John", 23);

    function showPerson() {
        // Modify name using setter
        person1.setName("PRADEEP");

        // Display data
        document.getElementById("info").innerHTML = `
            <strong>Name:</strong> ${person1.name} <br>
            <strong>Age:</strong> ${person1.age} <br>
            <strong>Greeting:</strong> ${person1.greet()} <br>
            <strong>Age in Months:</strong>
${person1.getAgeInMonths()}
        `;
    }
  </script>
</body>
</html>
```

OutPut:

User-Defined Object using Constructo

Show Person Info

Name: Jagadesh kumar
Age: 23
Greeting: Hello, I'm Jagadesh kumar and I'm 23 years old.
Age in Months: 276

29. 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".

Program:

```
<!DOCTYPE html>
<html>
<head>
   <title>Compare Two Numbers</title>
</head>
<body style="font-family: Arial; padding: 20px;">
   <h2>Compare Two Numbers</h2>
   <button onclick="compareNumbers()">Compare</button>
   <div id="result" style="margin-top: 20px; font-weight:
bold;"></div>
   <script>
     function compareNumbers() {
        let num1 = parseFloat(prompt("Enter first number:"));
        let num2 = parseFloat(prompt("Enter second number:"));
        let message = "";
        if (num1 > num2) {
           message = `${num1} - LARGER NUMBER`;
        } else if (num2 > num1) {
           message = `${num2} - LARGER NUMBER`;
        } else {
           message = "EQUAL NUMBERS";
        }
        alert(message); // Info dialog
        document.getElementById("result").innerHTML = message;
     }
   </script>
```

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

Output:

Compare Two Numbers

Compare

127.0.0.1:5500 says

Enter first number:

12

OK   Cancel

127.0.0.1:5500 says

Enter second number:

23

OK   Cancel

Compare Two Numbers

Compare

23 - LARGER NUMBER

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

Program:

```
<!DOCTYPE html>
<html>
<head>
   <title>Weekday Switch</title>
</head>
<body style="font-family: Arial; padding: 20px;">
   <h2>Display Weekday</h2>
   <button onclick="showWeekday()">Enter Day Number</button>
   <div id="output" style="margin-top: 20px; font-size: 18px; color:
darkblue;"></div>

   <script>
     function showWeekday() {
        let dayNum = parseInt(prompt("Enter a number (1 to 7):"));
        let dayName;
        switch(dayNum) {
           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
from 1 to 7.";
      }
      alert(dayName);
      document.getElementById("output").innerText = dayName;
    }
  </script>
</body>
</html>
```

Output:

Display Weekday

Enter Day Number

127.0.0.1:5500 says

Enter a number (1 to 7):

7

OK   Cancel

Display Weekday

Enter Day Number

Saturday

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

Program:
```html
<!DOCTYPE html>
<html>
<head>
   <title>Loops Example</title>
   <style>
     body { font-family: Arial; padding: 20px; }
     h2 { color: #2c3e50; }
     pre { background: #f4f4f4; padding: 10px; border-radius: 6px; }
   </style>
</head>
<body>
   <h2>Print 1 to 10 Using Different Loops</h2>
   <pre id="output"></pre>
   <script>
     let output = "";
     output += "Using for loop:\n";
     for (let i = 1; i <= 10; i++) {
        output += i + " ";}
     output += "\n\nUsing while loop:\n";
```

```
        let j = 1;
        while (j <= 10) {
            output += j + " ";
            j++;}
        output += "\n\nUsing do-while loop:\n";
        let k = 1;
        do {
            output += k + " ";
            k++;} while (k <= 10);
        document.getElementById("output").innerText = output;
    </script>
</body>
</html>
```

Output:

Print 1 to 10 Using Different Loops

```
Using for loop:
1 2 3 4 5 6 7 8 9 10

Using while loop:
1 2 3 4 5 6 7 8 9 10

Using do-while loop:
1 2 3 4 5 6 7 8 9 10
```

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

Program:

```
<!DOCTYPE html>
<html>
<body>
  <h3>Looping Through Object Data</h3>
  <pre id="output"></pre>

  <script>
    const person = { name: "Chandu", age: 23, city: "Mattagunta" };
    let result = "For-in:\n";
    // for-in loop (used with objects)
    for (let key in person) {
       result += `${key}: ${person[key]}\n`;
    }

    // Convert object to entries for forEach
    result += "\nforEach:\n";
```

```
      Object.entries(person).forEach(([key, value]) => {
        result += `${key}: ${value}\n`;
      });

      // for-of loop with Object.entries
      result += "\nFor-of:\n";
      for (let [key, value] of Object.entries(person)) {
        result += `${key}: ${value}\n`;
      }

      document.getElementById("output").innerText = result;
    </script>
  </body>
</html>
```

Output:

Looping Through Object Data

```
For-in:
name: Jagadesh Kumar
age: 23
city: Sompalli

forEach:
name: Jagadesh Kumar
age: 23
city: Sompalli

For-of:
name: Jagadesh Kumar
age: 23
city: Sompalli
```

33.Design a appropriate function should be called to display
i. Factorial of that number
ii. Fibonacci series up to that number

Program:

```html
<!DOCTYPE html>
<html>
<body>
    <h3>Factorial and Fibonacci Calculator</h3>
    <input type="number" id="num" placeholder="Enter number">
    <button onclick="calculate()">Show Result</button>
    <pre id="result"></pre>
    <script>
      function factorial(n) {
          if (n < 0) return "Not defined";
          let f = 1;
          for (let i = 1; i <= n; i++) f *= i;
          return f;
      }
      function fibonacci(n) {
          let a = 0, b = 1, res = "0";
          while (b <= n) [res, [a, b]] = [res + ", " + b, [b, a + b]];
          return res;
      }
      function calculate() {
          const n = +document.getElementById("num").value;
          document.getElementById("result").innerText =
             `Factorial of ${n}: ${factorial(n)}\nFibonacci up to ${n}:
${fibonacci(n)}`;
      }
   </script>
</body>
</html>
```

Output:

Factorial and Fibonacci Calculator

Enter number | Show Result

Factorial and Fibonacci Calculator

6 | Show Result

```
Factorial of 6: 720
Fibonacci up to 6: 0, 1, 1, 2, 3, 5
```

34. Design a HTML having a text box and four buttons named Factorial, Fibonacci, 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

Program:

```html
<!DOCTYPE html>
<html>
<head>
  <style>
    body {
      margin: 0;
      font-family: Arial, sans-serif;
      background: #f0f0f0;
      display: flex;
      height: 100vh;
      justify-content: center;
      align-items: center;
    }
    .container {
      background: white;
      padding: 30px;
      border-radius: 10px;
      text-align: center;
      box-shadow: 0 0 10px rgba(0,0,0,0.1);
    }
    input {
      padding: 8px;
      width: 200px;
      margin-bottom: 15px;
      font-size: 16px;
    }
    button {
      margin: 5px;
      padding: 10px 20px;
      font-size: 15px;
      cursor: pointer;
    }
    pre {
      margin-top: 15px;
```

```
        font-size: 16px;
        color: #333;
      }
    </style>
  </head>
  <body>

  <div class="container">
    <h3>Number Operations</h3>
    <input type="number" id="num" placeholder="Enter number"><br>
    <button onclick="showFactorial()">Factorial</button>
    <button onclick="showFibonacci()">Fibonacci</button>
    <pre id="result"></pre>
  </div>
  <script>
    function showFactorial() {
      let n = +document.getElementById("num").value, f = 1;
      for (let i = 1; i <= n; i++) f *= i;
      document.getElementById("result").innerText = `Factorial of
${n}: ${f}`;
    }
    function showFibonacci() {
      let n = +document.getElementById("num").value, a = 0, b = 1, s =
"0";
      while (b <= n) [s, [a, b]] = [s + ", " + b, [b, a + b]];
      document.getElementById("result").innerText = `Fibonacci up to
${n}: ${s}`;
    }
  </script>
  </body>
  </html>
```

Output:

Number Operations

Enter number

| Factorial | Fibonacci |

Number Operations

6

| Factorial | Fibonacci |

Factorial of 6: 720

Number Operations

6 ⬍

| Factorial | Fibonacci |

Fibonacci up to 6: 0, 1, 1, 2, 3, 5

---

35. 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)

Program:

```html
<!DOCTYPE html>
<html>
<head>
  <title>Registration Form</title>
  <style>
    body {
      font-family: Arial;
      display: flex;
      justify-content: center;
      align-items: center;
      height: 100vh;
      background: #f5f5f5;
    }
    .form-container {
      background: white;
      padding: 25px;
      border-radius: 8px;
      box-shadow: 0 0 10px #ccc;
    }
    input {
      display: block;
      margin-bottom: 15px;
      padding: 8px;
      width: 250px;
    }
    .error {
      color: red;
      font-size: 14px;
    }
  </style>
</head>
```

```html
<body>
    <div class="form-container">
        <h3>Registration Form</h3>
        <input type="text" id="name" placeholder="Enter Name" />
        <input type="text" id="mobile" placeholder="Enter Mobile" />
        <input type="text" id="email" placeholder="Enter Email" />
        <button onclick="validate()">Submit</button>
        <div id="msg" class="error"></div>
    </div>
    <script>
        function validate() {
            const name = document.getElementById("name").value.trim();
            const mobile = document.getElementById("mobile").value.trim();
            const email = document.getElementById("email").value.trim();
            const msg = document.getElementById("msg");
            const nameRegex = /^[A-Za-z][A-Za-z0-9]{5,}$/;
            const mobileRegex = /^\d{10}$/;
            const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
            if (!nameRegex.test(name)) {
                msg.innerText = "Invalid Name: Start with letter, at least 6 characters.";
            } else if (!mobileRegex.test(mobile)) {
                msg.innerText = "Invalid Mobile: Must be exactly 10 digits.";
            } else if (!emailRegex.test(email)) {
                msg.innerText = "Invalid Email: Use format name@example.com";
            } else {
                msg.style.color = "green";
                msg.innerText = "All fields are valid!";
            }
        }
    </script>
</body>
</html>
```
Output:

Registration Form

Enter Name

Enter Mobile

Enter Email

Submit

Registration Form

Jagadesh

2345761298

24M11mc155@adityauniversity.in

Submit
All fields are valid!

Registration Form

Jagadesh kumar

2345761298

24M11mc155@adityauniversity.in

Submit
Invalid Name: Start with letter, at least 6 characters.

36. Write a program to show the workflow of JavaScript code executable by creating web server in Node.js.

Program:

Step 1: server.js – Node.js Web Server

```
const http = require('http');
const fs = require('fs');
const path = require('path');

http.createServer((req, res) => {
   if (req.url === '/') {
      fs.readFile(path.join(__dirname, 'index.html'), (err, data) => {
         if (err) {
            res.writeHead(500);
            res.end('Error loading HTML file');
         } else {
            res.writeHead(200, { 'Content-Type': 'text/html' });
            res.end(data);
         }
      });
   }
}).listen(3000, () => console.log('Server running at
http://localhost:3000'));
```

Step 2: index.html – HTML + JavaScript Workflow

```
<!DOCTYPE html>
<html>
<head>
   <title>Node.js JavaScript Workflow</title>
</head>
<body style="text-align:center; font-family:sans-serif; margin-
top:50px;">
   <h2>JavaScript Executed via Node.js Server</h2>
```

```html
<button onclick="showMessage()">Click Me</button>
<p id="output"></p>

<script>
  function showMessage() {
    document.getElementById("output").textContent = "JavaScript is working!";
  }
</script>
</body>
</html>
```

How to Run:

- ➤ Make sure Node.js is installed.
- ➤ Save the above files (server.js, index.html) in the same folder.
- ➤ Open terminal and run:

> node server.js

Visit http://localhost:3000 in your browser.

Output:

JavaScript Executed via Node.js Server

Click Me

JavaScript is working!

37. Write a program to transfer data over http protocol using http module.

Program:

```
// server.js

const http = require('http');


// Create an HTTP server

const server = http.createServer((req, res) => {

    res.writeHead(200, { 'Content-Type': 'text/plain' });

    res.write('Hello! This data is sent over HTTP using the http module.');

    res.end();

});


// Server listens on port 3000

server.listen(3000, () => {

    console.log('Server running at http://localhost:3000');

});
```
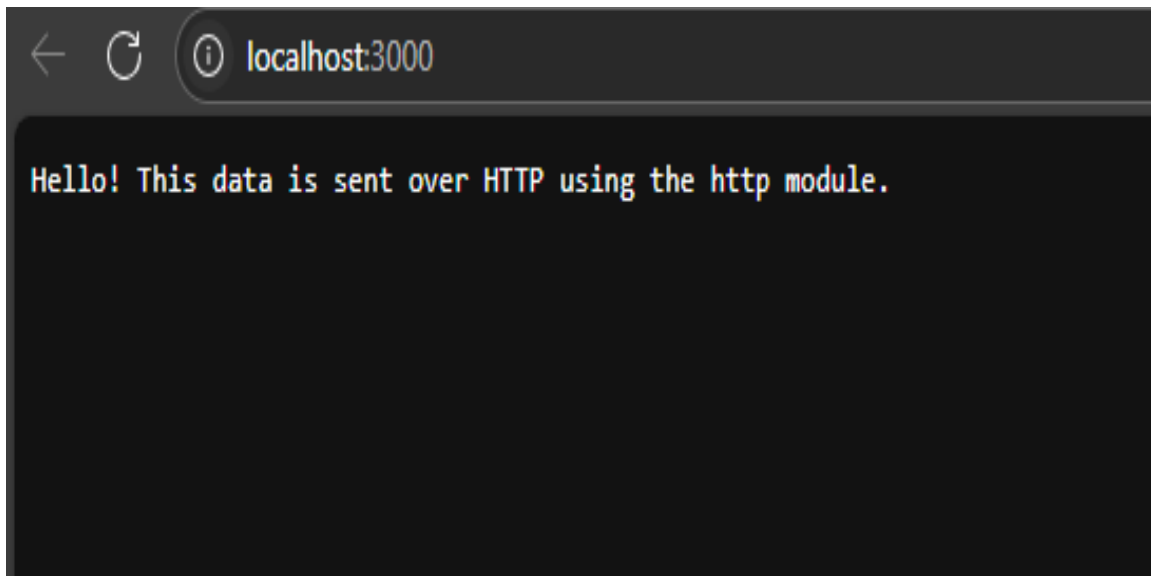
Output:

38.Create a text file src.txt and add the following content to it. (HTML, CSS, Javascript, Typescript, MongoDB, Express.js, React.js, Node.js)

Program:

➢ **using Node.js to do it programmatically:**

```
const fs = require('fs');

const content = 'HTML, CSS, Javascript, Typescript, MongoDB, Express.js, React.js, Node.js';

fs.writeFile('src.txt', content, (err) => {
    if (err) throw err;
    console.log('src.txt created and content written.');
});
```
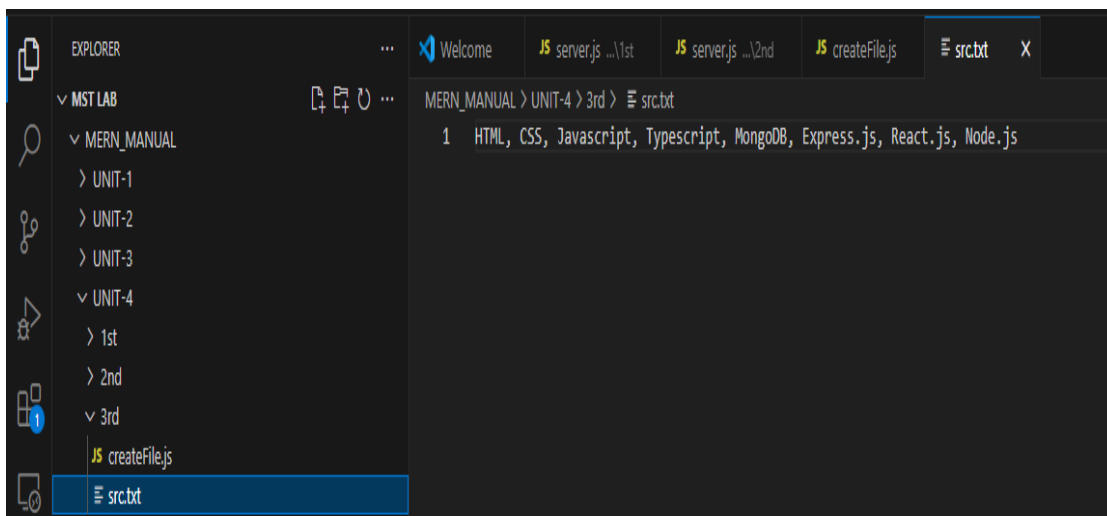
➢ **Save the code in a file like createFile.js and run it with:**

```
node createFile.js
```

Output:

39.Write a program to parse an URL using URL module.

Program:

```javascript
// Import the URL module

const { URL } = require('url');

// Given URL

const myURL = 'https://erp.adityauniversity.in/stu_studentProfile.htm';


// Parse the URL

const parsedUrl = new URL(myURL);


// Display components

console.log('Full URL:', parsedUrl.href);

console.log('Protocol:', parsedUrl.protocol);

console.log('Host:', parsedUrl.host);

console.log('Hostname:', parsedUrl.hostname);

console.log('Port:', parsedUrl.port || 'default');

console.log('Pathname:', parsedUrl.pathname);

console.log('Search (Query):', parsedUrl.search || 'none');

console.log('Search Params:', parsedUrl.searchParams.get('') || 'null');

console.log('Hash (Fragment):', parsedUrl.hash || 'none');
```
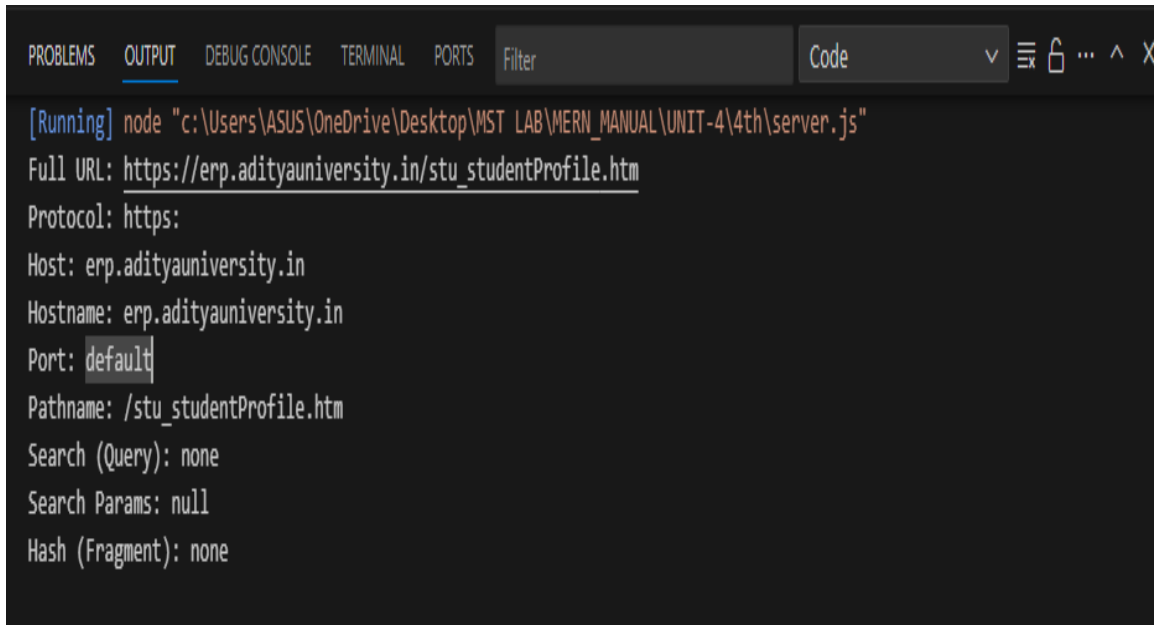
Output:

40. Write a program to create an user-defined module and show the workflow of Modularization of application using Node.js

Program:

mathUtils.js – User-Defined Module

```
// Function to add two numbers

function add(a, b) {

    return a + b;

}

// Function to multiply two numbers

function multiply(a, b) {

    return a * b;

}

// Exporting functions

module.exports = { add, multiply };
```

app.js – Main Application Using the Module

```
// Importing the user-defined module
const math = require('./mathUtils');

// Using the functions
console.log("Add(5, 3) =", math.add(5, 3));
console.log("Multiply(5, 3) =", math.multiply(5, 3));
```

Output:

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

PS C:\Users\ASUS\OneDrive\Desktop\MST LAB> cd MERN_MANUAL
PS C:\Users\ASUS\OneDrive\Desktop\MST LAB\MERN_MANUAL> cd UNIT-4
PS C:\Users\ASUS\OneDrive\Desktop\MST LAB\MERN_MANUAL\UNIT-4> cd 5th
PS C:\Users\ASUS\OneDrive\Desktop\MST LAB\MERN_MANUAL\UNIT-4\5th> node app.js
Add(5, 3) = 8
Multiply(5, 3) = 15
PS C:\Users\ASUS\OneDrive\Desktop\MST LAB\MERN_MANUAL\UNIT-4\5th>
```

41. Write a program to understand simple and special types

Program:

```html
<!DOCTYPE html>
<html>
<head>
   <title>Simple & Special Types in JavaScript</title>
</head>
<body>
   <h2 style="text-align:center;color:darkblue;">Check the Console for Output</h2>

   <script>

     // Simple Types

     let name = "Chandu";
     let age = 23;
     let isStudent = true;
     let city;
     let empty = null;
     let id = Symbol("id");
     let bigNum = 12345678901234567890012345n;

     console.group('%c□ SIMPLE TYPES', 'color:green;font-weight:bold;');

     console.log("%cname (String):", "color:purple;", name);
     console.log("%cage (Number):", "color:purple;", age);
     console.log("%cisStudent (Boolean):", "color:purple;", isStudent);
     console.log("%ccity (Undefined):", "color:purple;", city);
     console.log("%cempty (Null):", "color:purple;", empty);
     console.log("%cid (Symbol):", "color:purple;", id.toString());
     console.log("%cbigNum (BigInt):", "color:purple;", bigNum);
     console.groupEnd();
```

```
// Special Types


    let person = { name: "Chandu", age: 23 };
    let colors = ["red", "green", "blue"];
    let greet = function () { return "Hello!"; };
    let today = new Date();
    let pattern = /[a-z]+/;

    console.group('%c□ SPECIAL TYPES', 'color:blue;font-weight:bold;');

    console.log("%cperson (Object):", "color:teal;", person);
    console.log("%ccolors (Array):", "color:teal;", colors);
    console.log("%cgreet() (Function):", "color:teal;", greet());
    console.log("%ctoday (Date):", "color:teal;", today);
    console.log("%cpattern (RegExp):", "color:teal;", pattern);
    console.groupEnd();
  </script>
</body>
</html>
```
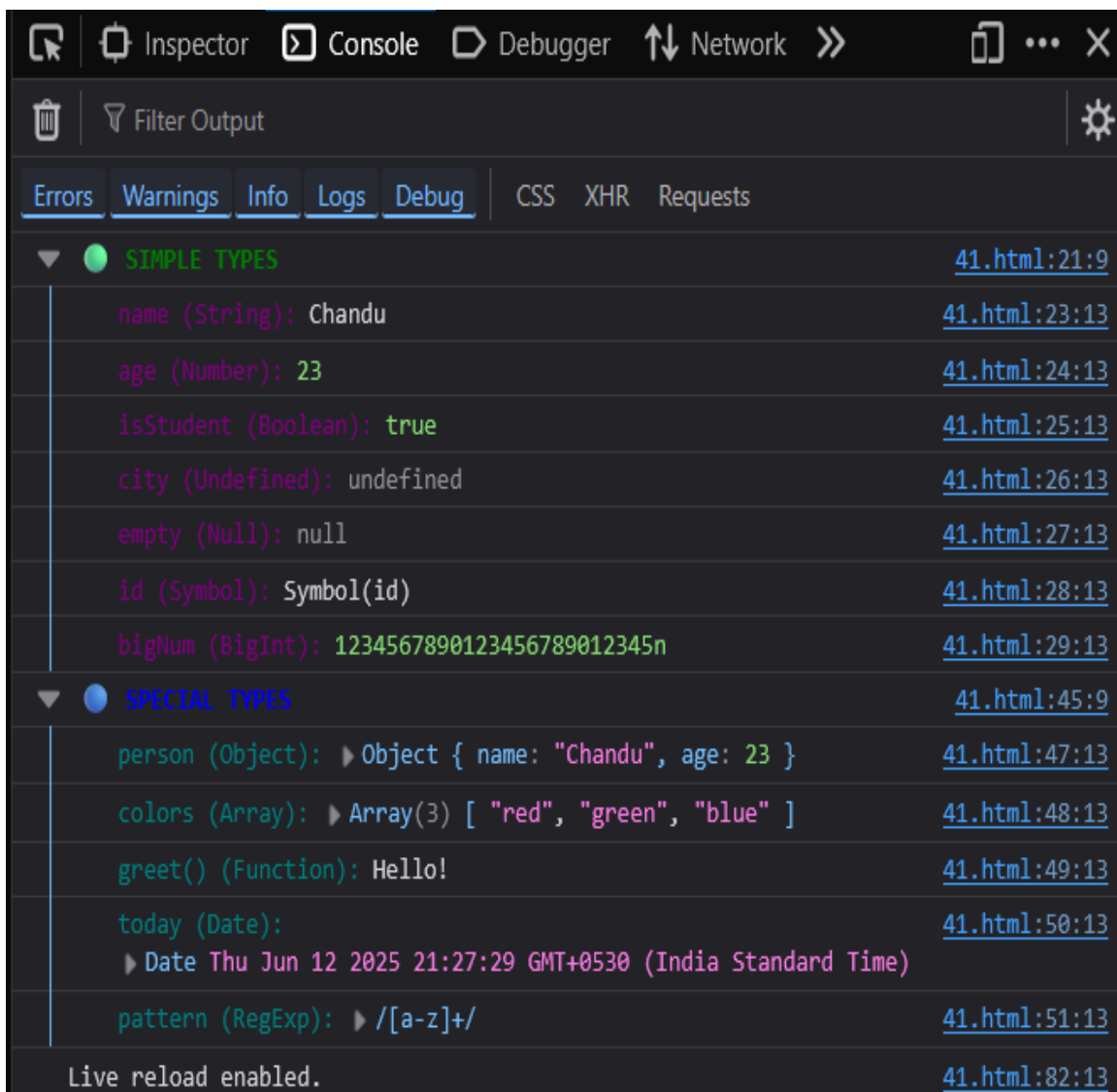
Output:

Check the Console for Output

42. Write a program to understand function parameter and return types

Program:

```
<!DOCTYPE html>
<html>
<head>
   <title>Function Parameters & Return Types</title>
   <style>
     body {
        font-family: Arial, sans-serif;
        padding: 20px;
        text-align: center;
     }
     input, button {
        padding: 8px;
        margin: 5px;
     }
     pre {
        background: #f4f4f4;
        padding: 15px;
        border-radius: 5px;
        width: 80%;
        margin: auto;
        text-align: left;
     }
   </style>
</head>
<body>
   <h2>Function Parameters & Return Types</h2>
   <input type="number" id="num1" placeholder="Enter first number">
   <input type="number" id="num2" placeholder="Enter second number">
   <br>
   <button onclick="showResults()">Calculate</button>
```

```html
    <pre id="output"></pre>
    <script>
      // Function with parameters and return type - Number
      function add(a, b) {
        return a + b;
      }
      // Function with parameters and return type - String
      function describeSum(a, b) {
        return `The sum of ${a} and ${b} is ${a + b}`;
      }
      // Function with return type - Array
      function getNumbers(a, b) {
        return [a, b];
      }
      // Function with return type - Object
      function getSummary(a, b) {
        return {
          num1: a,
          num2: b,
          sum: a + b
        };
      }
      // Main function
      function showResults() {
        const a = parseInt(document.getElementById('num1').value);
        const b = parseInt(document.getElementById('num2').value);
        const output = `
➤ add(a, b) → ${add(a, b)}
➤ describeSum(a, b) → "${describeSum(a, b)}"
➤ getNumbers(a, b) → [${getNumbers(a, b)}]
➤ getSummary(a, b) → ${JSON.stringify(getSummary(a, b), null, 2)}
        `;
        document.getElementById("output").innerText = output;
      }
    </script>
</body>
</html>
```

Output:

Function Parameters & Return Types

Enter first number	Enter second number

Calculate

Function Parameters & Return Types

5	8

Calculate

```
➤ add(a, b) → 13
➤ describeSum(a, b) → "The sum of 5 and 8 is 13"
➤ getNumbers(a, b) → [5,8]
➤ getSummary(a, b) → {
  "num1": 5,
  "num2": 8,
  "sum": 13
}
```

43.Write a program to show the importance with Arrow function. Use optional, default and REST parameters.

Program:

```
<!DOCTYPE html>
<html>
<head>
  <title>Arrow Function Demo</title>
  <style>
    body { font-family: Arial; text-align: center; padding: 20px; }
    pre { background: #f4f4f4; padding: 10px; border-radius: 5px;
display: inline-block; text-align: left; }
  </style>
</head>
<body>
  <h2>Arrow Functions with Parameters</h2>
  <pre id="output"></pre>
  <script>
    // Arrow function with default & optional parameter
    const greet = (name = "Guest", title) => `Hello ${title ? title + " " :
""}${name}!`;
    // Arrow function with REST parameter
    const sum = (...nums) => nums.reduce((a, b) => a + b, 0);
    // Display results
    const output = `
→ greet("Chandu", "Mr.") → ${greet("Chadu", "Mr.")}
→ greet("Sekhar") → ${greet("Sekhar")}
→ greet() → ${greet()}
→ sum(10, 20, 30) → ${sum(10, 20, 30)}
→ sum(5, 5, 5, 5, 5) → ${sum(5, 5, 5, 5, 5)}
→ sum() → ${sum()}
    `;
    document.getElementById("output").innerText = output;
  </script>
</body>
</html>
```

Output:

Arrow Functions with Parameters

```
→ greet("jagadesh", "Mr.") → Hello Mr. jagadesh!
→ greet("Kumar") → Hello Kumar!
→ greet() → Hello Guest!
→ sum(10, 20, 30) → 60
→ sum(5, 5, 5, 5, 5) → 25
→ sum() → 0
```

44.Write a program to understand the working of typescript with class, constructor, properties, methods and access specifiers.

To Run this TypeScript Code:

1.Install TypeScript (if not yet):

npm install -g typescript

2.Save the code in a file named main.ts

3.Compile to JavaScript:

tsc main.ts

4.Run it using Node.js:

node main.js

Program:

```typescript
class Student {
    // Properties with access specifiers
    public name: string;
    protected age: number;
    private rollNo: string;

    // Constructor
    constructor(name: string, age: number, rollNo: string) {
        this.name = name;
        this.age = age;
        this.rollNo = rollNo;
    }
```

```typescript
  // Public Method
  public displayInfo(): void {
    console.log(`Name: ${this.name}`);
    console.log(`Age: ${this.age}`);
    console.log(`Roll No: ${this.rollNo}`);
  }

  // Private Method
  private secretMessage(): string {
    return "This is private!";
  }
}

// Inherited class
class GraduateStudent extends Student {
  constructor(name: string, age: number, rollNo: string) {
    super(name, age, rollNo);
  }

  showAge(): void {
    console.log(`(Protected) Age: ${this.age}`);
  }
}

// Creating object
const student1 = new GraduateStudent("Chandu", 23,
"24M11MC003");
student1.displayInfo();
student1.showAge();
```

Output:

```
PS D:\TypeScript> tsc Student.ts
PS D:\TypeScript> node Student.js
Name: jagadesh
Age: 23
Roll No: 24M11MC155
(Protected) Age: 23
PS D:\TypeScript>
```

45.Write a CSS program,to apply 2D and 3D transformations in a web page

Program:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>2D and 3D Transformations</title>
  <style>
    body {
      display: flex;
      flex-direction: column;
      align-items: center;
      justify-content: center;
      min-height: 100vh;
      background: #f2f2f2;
      font-family: sans-serif;
    }

    h2 {
      margin-bottom: 20px;
      color: #333;
    }

    .container {
      display: flex;
      gap: 50px;
    }

    .box {
      width: 150px;
```

```css
      height: 150px;
      background: #4CAF50;
      color: white;
      display: flex;
      align-items: center;
      justify-content: center;
      font-size: 1.2rem;
      transition: all 0.5s ease;
      cursor: pointer;
    }

    .box:hover {
      box-shadow: 0 0 20px rgba(0,0,0,0.3);
    }

    /* 2D Transformation */
    .transform2d:hover {
      transform: rotate(45deg) scale(1.2);
    }

    /* 3D Transformation */
    .transform3d {
      perspective: 600px;
    }

    .transform3d:hover .inner {
      transform: rotateY(180deg);
    }

    .inner {
      width: 100%;
      height: 100%;
      background: #2196F3;
      display: flex;
      align-items: center;
      justify-content: center;
      transform-style: preserve-3d;
```

```
        transition: transform 0.8s;
      }
    </style>
  </head>
  <body>

    <h2>2D & 3D Transformations in CSS</h2>
    <div class="container">
      <div class="box transform2d">2D Box</div>

      <div class="box transform3d">
        <div class="inner">3D Box</div>
      </div>
    </div>

  </body>
</html>
```
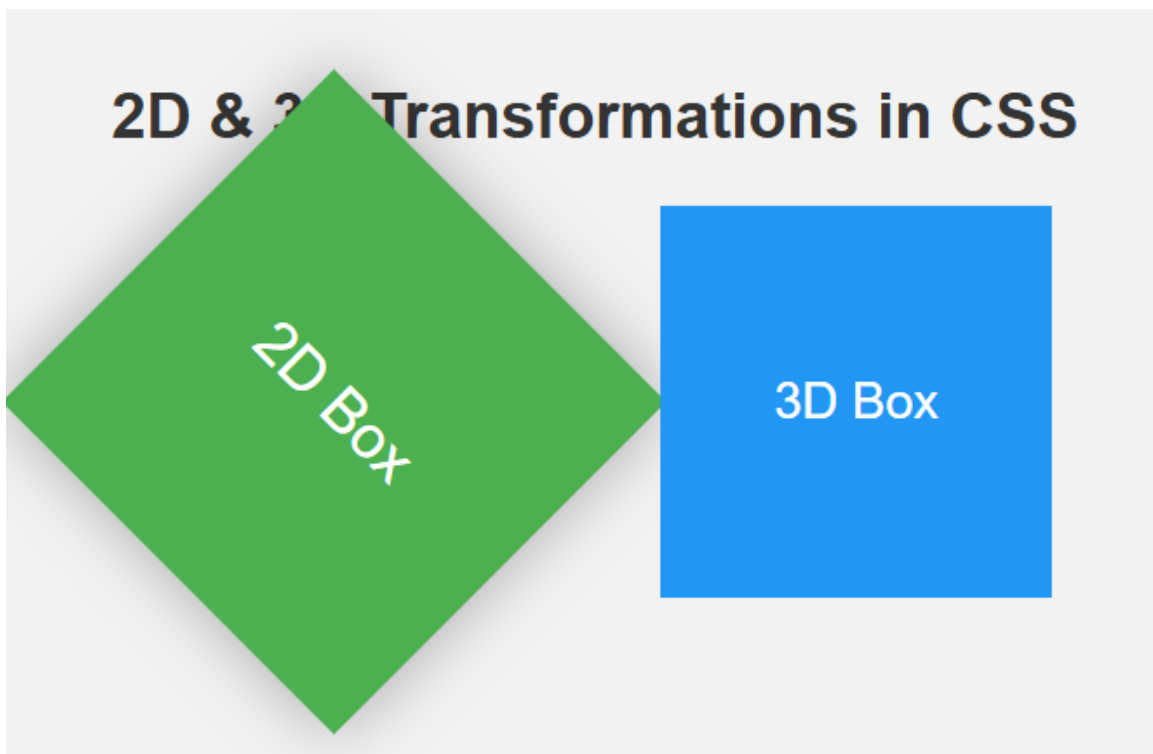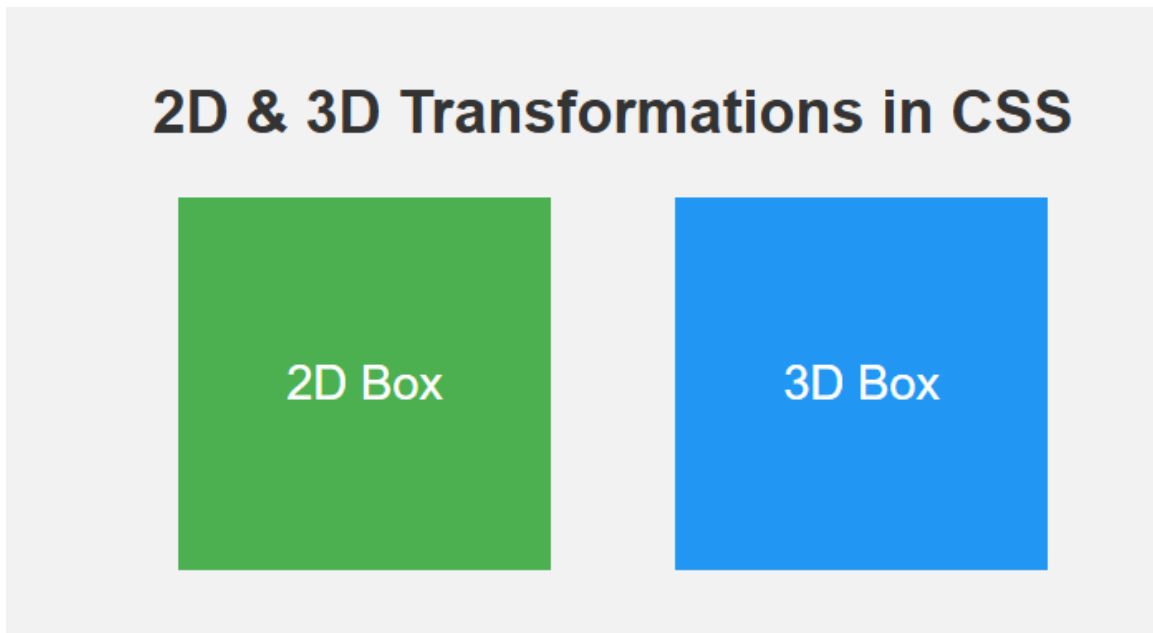
Output:

46.Design a web page with page with new features of HTML file and CSS3.

Program:

```html
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <title>HTML5 & CSS3 Modern Page</title>
   <style>
      body {
         margin: 0;
         font-family: 'Segoe UI', sans-serif;
         background: linear-gradient(to right, #e3f2fd, #fff);
         color: #333;
      }

      header {
         background: #2196F3;
         color: white;
         padding: 20px;
         text-align: center;
         box-shadow: 0 4px 10px rgba(0, 0, 0, 0.2);
      }

      nav {
         background: #1976D2;
         display: flex;
         justify-content: center;
         gap: 20px;
         padding: 10px;
      }

      nav a {
         color: white;
         text-decoration: none;
```

```css
    font-weight: bold;
    transition: color 0.3s;
}

nav a:hover {
    color: #FFD700;
}

main {
    display: flex;
    flex-wrap: wrap;
    justify-content: space-around;
    padding: 30px;
}

section, article {
    background: #fff;
    border-radius: 10px;
    padding: 20px;
    margin: 20px;
    box-shadow: 0 2px 10px rgba(0,0,0,0.1);
    width: 300px;
    transition: transform 0.3s;
}

section:hover, article:hover {
    transform: scale(1.03);
}

footer {
    text-align: center;
    background: #0D47A1;
    color: white;
    padding: 15px;
    margin-top: 30px;
}
```

```css
      video {
        width: 100%;
        border-radius: 10px;
      }

      form input, form button {
        display: block;
        width: 100%;
        margin: 10px 0;
        padding: 10px;
        border-radius: 6px;
        border: 1px solid #ccc;
      }

      form button {
        background: #2196F3;
        color: white;
        border: none;
        cursor: pointer;
      }

      form button:hover {
        background: #1976D2;
      }
    </style>
  </head>
  <body>

    <header>
      <h1>Welcome to HTML5 & CSS3 Demo Page</h1>
      <p>Modern features using semantic tags and stylish CSS</p>
    </header>

    <nav>
      <a href="#">Home</a>
      <a href="#">Features</a>
      <a href="#">Gallery</a>
```

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

   <main>
     <section>
        <h2>Embedded Video</h2>
        <video controls>
           <source
src="https://www.w3schools.com/html/mov_bbb.mp4"
type="video/mp4">
           Your browser does not support HTML5 video.
        </video>
     </section>

     <article>
        <h2>HTML5 Form</h2>
        <form>
           <input type="text" placeholder="Your Name" required>
           <input type="email" placeholder="Email Address" required>
           <input type="number" placeholder="Phone Number">
           <button type="submit">Submit</button>
        </form>
     </article>
   </main>

   <footer>
     &copy; 2025 HTML5 CSS3 Web Page. All rights reserved.
   </footer>

</body>
</html>
```
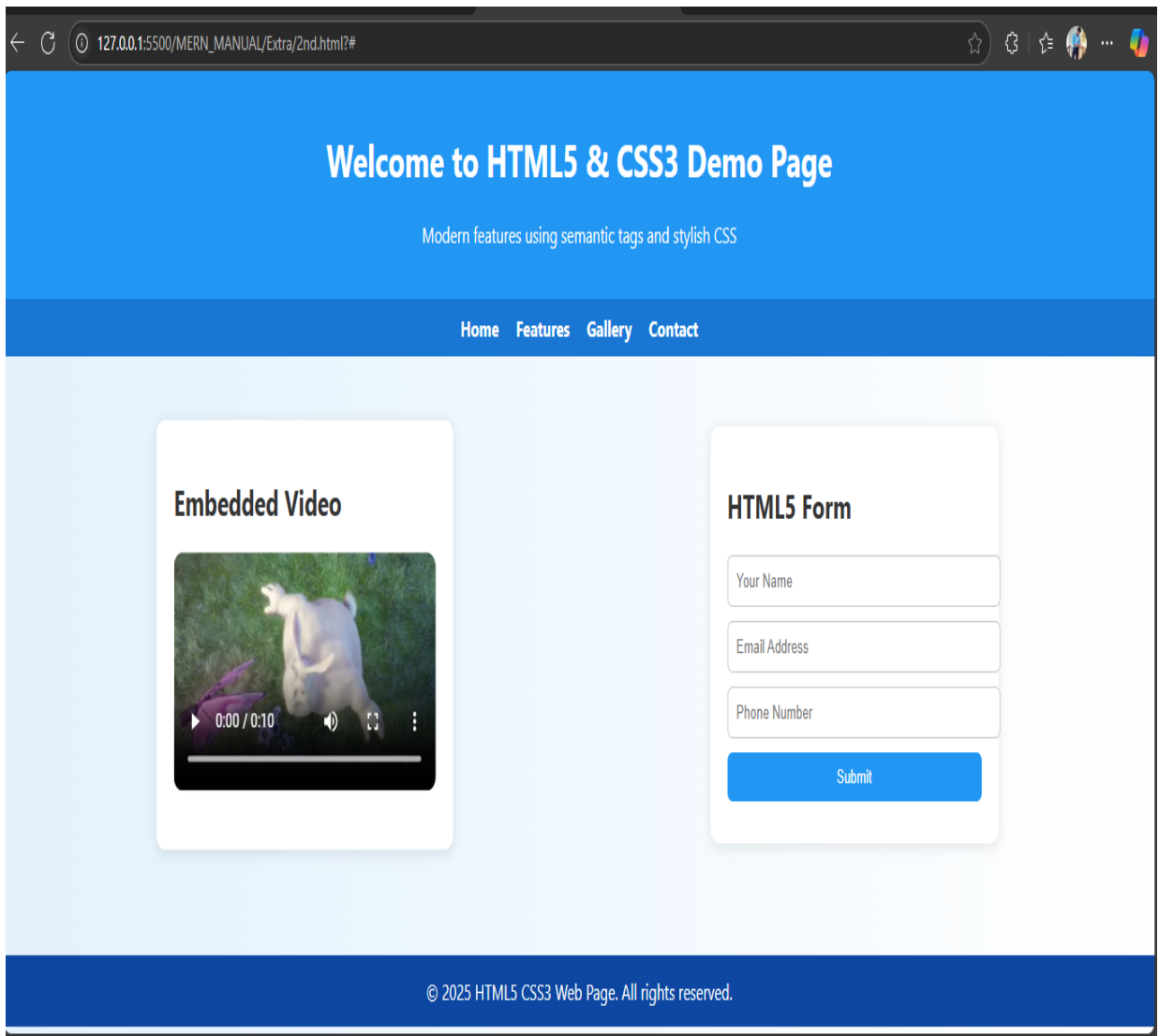
Output:

47.Design a to-do list application using javascript

Program:

```html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <title>Simple To-Do List</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      background: #f4f7f8;
      display: flex;
      justify-content: center;
      padding: 50px;
    }
    .todo-container {
      background: white;
      padding: 30px;
      border-radius: 8px;
      box-shadow: 0 8px 20px rgba(0,0,0,0.1);
      width: 350px;
    }
    h2 {
      margin-bottom: 20px;
      text-align: center;
      color: #333;
    }
    input[type="text"] {
      width: 100%;
      padding: 10px;
      border: 2px solid #ddd;
      border-radius: 5px;
      font-size: 16px;
      box-sizing: border-box;
```

```css
    }
    button {
        margin-top: 10px;
        width: 100%;
        background: #28a745;
        color: white;
        font-size: 16px;
        padding: 10px;
        border: none;
        border-radius: 5px;
        cursor: pointer;
        transition: background 0.3s ease;
    }
    button:hover {
        background: #218838;
    }
    ul {
        list-style: none;
        padding-left: 0;
        margin-top: 20px;
    }
    li {
        background: #fafafa;
        margin-bottom: 10px;
        padding: 12px 15px;
        border-radius: 5px;
        display: flex;
        justify-content: space-between;
        align-items: center;
        border: 1px solid #ddd;
        transition: background 0.3s ease;
    }
    li.completed {
        text-decoration: line-through;
        color: #888;
        background: #d4edda;
        border-color: #c3e6cb;
```

```
        }
        li button {
            background: #dc3545;
            border: none;
            color: white;
            padding: 5px 10px;
            border-radius: 4px;
            cursor: pointer;
            font-size: 14px;
            transition: background 0.3s ease;
        }
        li button:hover {
            background: #c82333;
        }
    </style>
</head>
<body>

    <div class="todo-container">
        <h2>My To-Do List</h2>
        <input type="text" id="taskInput" placeholder="Add a new
task..." />
        <button onclick="addTask()">Add Task</button>
        <ul id="taskList"></ul>
    </div>

    <script>
        const taskInput = document.getElementById('taskInput');
        const taskList = document.getElementById('taskList');

        function addTask() {
            const taskText = taskInput.value.trim();
            if (taskText === '') {
                alert('Please enter a task!');
                return;
            }
```

```
      // Create list item
      const li = document.createElement('li');
      li.textContent = taskText;

      // Toggle completion on click
      li.addEventListener('click', () => {
        li.classList.toggle('completed');
      });

      // Create delete button
      const delBtn = document.createElement('button');
      delBtn.textContent = 'Delete';
      delBtn.onclick = (e) => {
        e.stopPropagation(); // prevent toggling complete
        taskList.removeChild(li);
      };

      li.appendChild(delBtn);
      taskList.appendChild(li);

      taskInput.value = '';
      taskInput.focus();
    }

    // Optional: allow adding task by pressing Enter key
    taskInput.addEventListener('keypress', (e) => {
      if (e.key === 'Enter') addTask();
    });
  </script>
</body>
</html>
```

Output:

My To-Do List

Add a new task...

Add Task

hello   Delete

~~chandu~~   ~~Delete~~

linux   Delete

~~study~~   ~~Delete~~