Web Technologies LAB

(CIE - 356T)

Faculty Name : Mr. Alok Kumar

Name : Amit Singhal

Enrollment No. : 11614802722

Semester : 6

Group : AIML-II-B

Maharaja Agrasen Institute of Technology, PSP Area,

Sector – 22, Rohini, New Delhi – 110085

MAHARAJA AGRASEN INSTITUTE OF TECHNOLOGY

VISION

"To attain global excellence through **education, innovation, research,** and **work ethics** with the commitment to **serve humanity."**

MISSION

M1. To promote diversification by adopting advancement in science, technology, management, and allied discipline through continuous learning

M2. To foster **moral values** in students and equip them for developing sustainable solutions to serve both national and global needs in society and industry.

M3. To **digitize educational resources and process** for enhanced teaching and effective learning.

M4. To cultivate an **environment** supporting **incubation, product development, technology transfer, capacity building and entrepreneurship**.

M5. To encourage **faculty-student networking with alumni, industry, institutions,** and other **stakeholders** for collective engagement.

Department of Computer Science and Engineering

MAHARAJA AGRASEN INSTITUTE OF TECHNOLOGY

VISION

"To attain global excellence through education, innovation, research, and work ethics in the field of Computer Science and engineering with the commitment to serve humanity."

MISSION

M1. To lead in the advancement of computer science and engineering through internationally recognized research and education.

M2. To prepare students for full and ethical participation in a diverse society and encourage lifelong learning.

M3. To foster development of problem solving and communication skills as an integral component of the profession.

M4. To impart knowledge, skills and cultivate an environment supporting incubation, product development, technology transfer, capacity building and entrepreneurship in the field of computer science and engineering.

M5. To encourage faculty, student's networking with alumni, industry, institutions, and other stakeholders for collective engagement.

LAB INDEX

S.No	Experiment	R1	R2	R3	R4	R5	Total Marks (10)	Remarks	Faculty Signature
		Is able to identify and define the objective of the given problem?	Is proposed design /procedure /algorithm solves the problem?	Has the understanding of the tool/programming language to implement the proposed solution?	Are the result(s) verified using sufficient test data to support the conclusions?	Individuality of submission?			
		2 Marks	2 Marks	2 Marks	2 Marks	2 Marks			

Experiment - 1

AIM :: Design web pages for your college containing a description of the courses, departments, faculties, library etc, use href, list tags.

Theory ::

Introduction to HTML: Tags, Attributes, and Structure

What is HTML?

HTML (HyperText Markup Language) is the standard language for creating web pages. It defines the structure of a webpage using elements enclosed in tags.

HTML Tags

HTML tags define the elements of a webpage. Tags are enclosed in angle brackets (< >) and usually come in pairs:

- **Opening tag** (`<tag>`)
- **Closing tag** (`</tag>`)

For example:

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

Common HTML Tags

- `<html>`: Defines the root of an HTML document
- `<head>`: Contains meta-information and links
- `<title>`: Sets the page title
- `<body>`: Contains the visible content
- `<h1>` to `<h6>`: Headings
- `<p>`: Paragraph
- `<a>`: Hyperlinks
- `<img>`: Images
- `<ul>`, `<ol>`, `<li>`: Lists

HTML Attributes

Attributes provide additional information about an element. They are written inside the opening tag.

Example:

```
<a href="https://example.com">Click here</a>
```

- `href`: Specifies the link destination
- `src`: Defines the image source (`<img>`)
- `alt`: Alternative text for images

Conclusion

HTML is the foundation of web development. Understanding tags, attributes, and structure is essential for building web pages.

Code :: **index.html**

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

 <head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="style.css">
  <title>MAIT College</title>
 </head>

 <body>
  <header>
   <div class="logo">
    <img src="MAIT.png" alt="College Logo">
   </div>
   <div class="title">
    <h1>Maharaja Agrasen Institute of Technology</h1>
    <p>Excellence in Education & Research</p>
   </div>
  </header>

  <nav>
   <ul>
    <li><a href="#home">Home</a></li>
    <li><a href="#about">About</a></li>
    <li><a href="#courses">Courses</a></li>
    <li><a href="#departments">Departments</a></li>
    <li><a href="#faculty">Faculty</a></li>
    <li><a href="#library">Library</a></li>
    <li><a href="#contact">Contact</a></li>
   </ul>
  </nav>

  <section id="about">
   <h2>About Us</h2>
   <p>
```

```
        Maharaja Agrasen Institute of Technology (MAIT) is a premier engineering institution offering world-class
        education in various fields of technology and management. Established with a mission to foster innovation
        and excellence, MAIT provides industry-relevant training and hands-on experience to its students.
      </p>
    </section>

    <section id="courses">
      <h2>Courses Offered</h2>
      <div class="course-list">
        <div class="course">
          <h3>B.Tech in Computer Science</h3>
          <p>Duration: 4 years</p>
        </div>
        <div class="course">
          <h3>B.Tech in Electronics</h3>
          <p>Duration: 4 years</p>
        </div>
        <div class="course">
          <h3>MBA</h3>
          <p>Duration: 2 years</p>
        </div>
        <div class="course">
          <h3>B.Sc in Physics</h3>
          <p>Duration: 3 years</p>
        </div>
      </div>
    </section>

    <section id="departments">
      <h2>Departments</h2>
      <ul>
        <li><a href="#cse">Computer Science & Engineering</a></li>
        <li><a href="#ece">Electronics & Communication Engineering</a></li>
        <li><a href="#mech">Mechanical Engineering</a></li>
        <li><a href="#civil">Civil Engineering</a></li>
        <li><a href="#mba">Management Studies</a></li>
      </ul>
    </section>

    <section id="faculty">
```

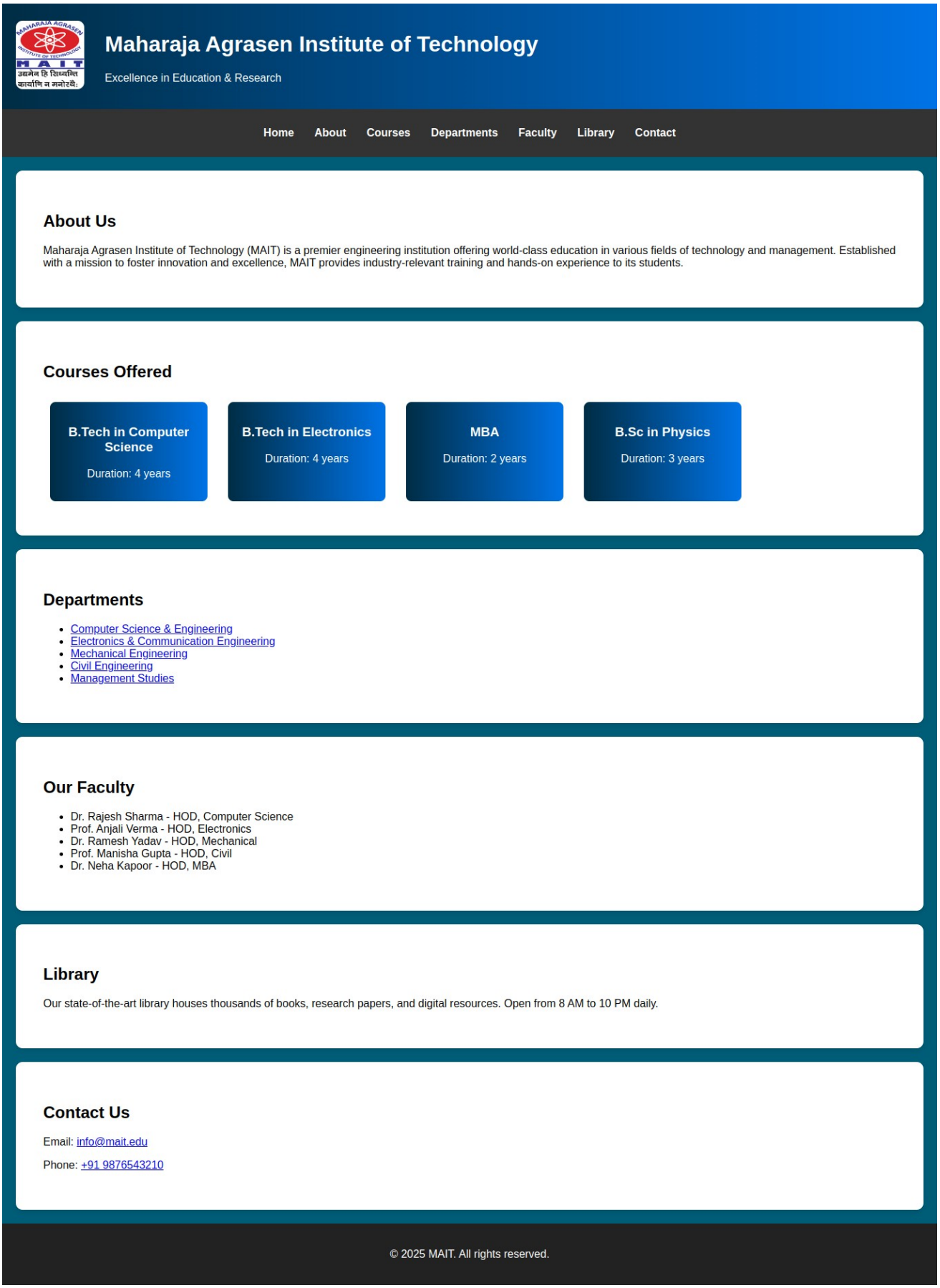
```
    <h2>Our Faculty</h2>
    <ul>
      <li>Dr. Rajesh Sharma - HOD, Computer Science</li>
      <li>Prof. Anjali Verma - HOD, Electronics</li>
      <li>Dr. Ramesh Yadav - HOD, Mechanical</li>
      <li>Prof. Manisha Gupta - HOD, Civil</li>
      <li>Dr. Neha Kapoor - HOD, MBA</li>
    </ul>
  </section>

  <section id="library">
    <h2>Library</h2>
    <p>
      Our state-of-the-art library houses thousands of books, research papers, and digital
resources.
      Open from 8 AM to 10 PM daily.
    </p>
  </section>

  <section id="contact">
    <h2>Contact Us</h2>
    <p>Email: <a href="mailto:info@mait.edu">info@mait.edu</a></p>
    <p>Phone: <a href="tel:+919876543210">+91 9876543210</a></p>
  </section>

  <footer>
    <p>&copy; 2025 MAIT. All rights reserved.</p>
  </footer>
 </body>

</html>
```

style.css

```
body {
  font-family: 'Arial', sans-serif;
  margin: 0;
  padding: 0;
  background-color: #005d78;
}

header {
  display: flex;
  align-items: center;
  background: linear-gradient(to right, #002d45, #0073e6);
  color: white;
  padding: 20px;
}

.logo img {
  width: 100px;
  height: 100px;
  border-radius: 10%;
}

.title {
  margin-left: 30px;
}

nav {
  background: #333;
  padding: 10px 0;
}

nav ul {
  list-style: none;
  display: flex;
  justify-content: center;
  padding: 0;
}

nav ul li {
  margin: 0 15px;
}
```

```
nav ul li a {
  color: white;
  text-decoration: none;
  font-weight: bold;
}

nav ul li a:hover {
  color: #f4a261;
}

section {
  padding: 40px;
  background: white;
  margin: 20px;
  border-radius: 10px;
  box-shadow: 0px 4px 8px rgba(0, 0, 0, 0.1);
}

.course-list {
  display: flex;
  justify-content: left;
  gap: 10px;
  flex-wrap: wrap;
}

.course {
  background: linear-gradient(to right, #002d45, #0073e6);
  color: white;
  padding: 15px;
  border-radius: 8px;
  width: 200px;
  text-align: center;
  justify-content: center;
  margin: 10px;
}

footer {
  background: #222;
  color: white;
  text-align: center;
  padding: 20px;
  margin-top: 20px;
}
```

Output ::


Maharaja Agrasen Institute of Technology
Excellence in Education & Research
Home
About
Courses
Departments
Faculty
Library
Contact
About Us
Maharaja Agrasen Institute of Technology (MAIT) is a premier engineering institution offering world-class education in various fields of technology and management. Established with a mission to foster innovation and excellence, MAIT provides industry-relevant training and hands-on experience to its students.
Courses Offered
B.Tech in Computer Science
Duration: 4 years
B.Tech in Electronics
Duration: 4 years
MBA
Duration: 2 years
B.Sc in Physics
Duration: 3 years
Departments
Computer Science & Engineering
Electronics & Communication Engineering
Mechanical Engineering
Civil Engineering
Management Studies
Our Faculty
Dr. Rajesh Sharma - HOD, Computer Science
Prof. Anjali Verma - HOD, Electronics
Dr. Ramesh Yadav - HOD, Mechanical
Prof. Manisha Gupta - HOD, Civil
Dr. Neha Kapoor - HOD, MBA
Library
Our state-of-the-art library houses thousands of books, research papers, and digital resources. Open from 8 AM to 10 PM daily.
Contact Us
Email: info@mait.edu
Phone: +91 9876543210
© 2025 MAIT. All rights reserved.

Experiment - 2

AIM :: Write html code to develop a webpage having two frames that divide the webpage into two equal rows and then divide the row into equal columns fill each frame with a different background color.

Theory ::

Developing a Webpage with Frames

Introduction to Frames in Web Design

In modern web design, creating layouts with multiple sections can be efficiently handled using CSS. While the older `<frame>` and `<frameset>` tags are deprecated in HTML5 and are no longer recommended, developers can use `<div>` elements with CSS to achieve a similar effect. This approach allows for better flexibility, responsiveness, and control over the layout of the webpage.

Understanding the Structure

To simulate the functionality of frames using modern HTML and CSS, we use `<div>` elements inside a container. The layout is organized into two main rows, and each row is further divided into two equal columns, similar to how `<frameset>` and `<frame>` would have been used in the past. Here's a breakdown:

1. **`<div class="frames">`**: This is the container for all frames (or sections of the page). It's used to hold the two rows of content.
2. **`<div class="row">`**: Each row within the frames container holds two columns. This is equivalent to dividing the page into horizontal sections.
3. **`<div class="frame">`**: Each frame is represented by a `<div>` element inside a row. These frames are styled to have a fixed background color and a flexible size, which adapts according to the screen size.

CSS Flexbox for Layout

Flexbox is used in this method to create a flexible layout. It divides the available space into equal portions for the columns and rows. Here are the key properties used:

- **`display: flex;`**: This property is used on the container (`.frames` and `.row`) to create a flexible box layout.
- **`flex: 1;`**: This ensures each frame takes up an equal share of space in the row or column.
- **`background-color`**: Each frame is assigned a unique background color for distinction.

Conclusion

By using modern HTML5 elements like `<div>` combined with CSS, web designers can achieve complex layouts without relying on deprecated tags like `<frame>` and `<frameset>`. This method is responsive, easy to customize, and compatible with modern web standards.

Code :: **index.html**

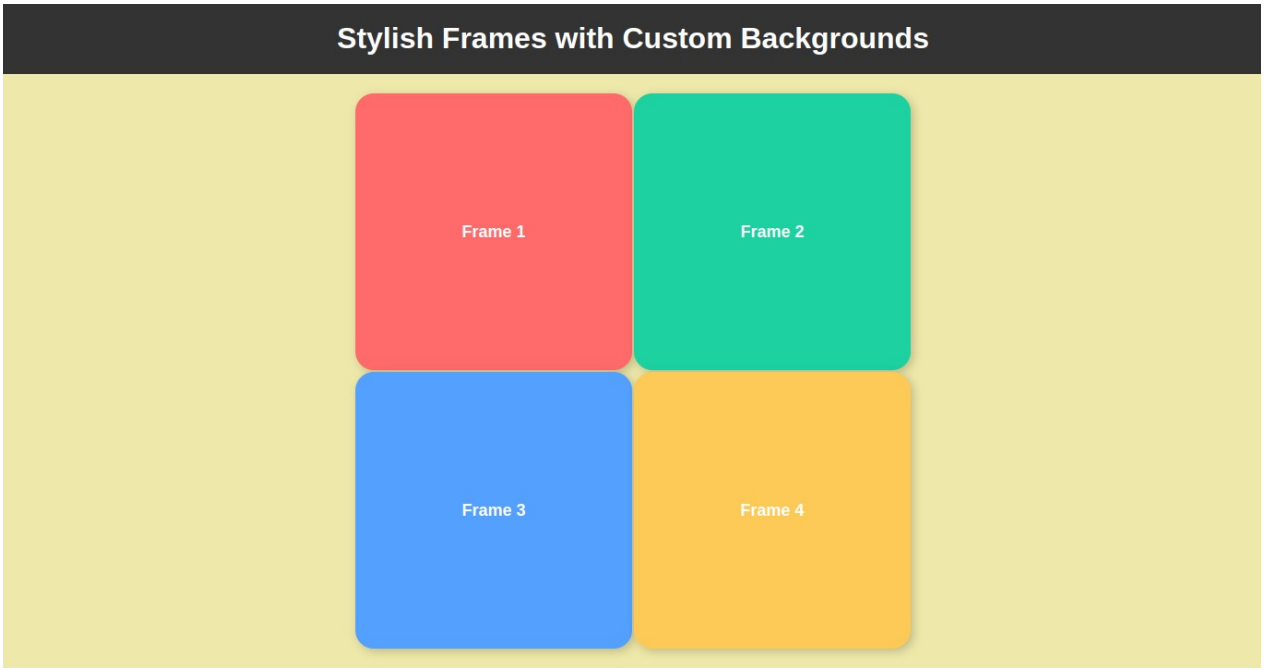
```
<!DOCTYPE html>
<html lang="en">

 <head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <link rel="stylesheet" href="style.css" />
  <title>Stylish Frames Webpage</title>
 </head>

 <body>
  <h1>Stylish Frames with Custom Backgrounds</h1>
  <div class="frames">
   <div class="row">
    <div class="frame frame1">Frame 1</div>
    <div class="frame frame2">Frame 2</div>
   </div>
   <div class="row">
    <div class="frame frame3">Frame 3</div>
    <div class="frame frame4">Frame 4</div>
   </div>
  </div>
 </body>

</html>
```

style.css

```
body {
  font-family: 'Arial', sans-serif;
  background-color: palegoldenrod;
  text-align: center;
  margin: 0;
  padding: 0;
}

h1 {
  background: #333;
  color: white;
  padding: 20px;
```

```
  margin: 0;
}

.frames {
  display: flex;
  flex-direction: column;
  align-items: center;
  margin: 20px;
}

.row {
  display: flex;
}

.frame {
  width: 300px;
  height: 300px;
  margin: 1px;
  display: flex;
  align-items: center;
  justify-content: center;
  font-size: 18px;
  font-weight: bold;
  color: white;
  border-radius: 20px;
  box-shadow: 3px 3px 10px rgba(0, 0, 0, 0.2);
}

.frame1 {
  background-color: #ff6b6b;
}

.frame2 {
  background-color: #1dd1a1;
}

.frame3 {
  background-color: #54a0ff;
}

.frame4 {
  background-color: #feca57;
}
```

Output ::


Stylish Frames with Custom Backgrounds
Frame 1
Frame 2
Frame 3
Frame 4

Experiment - 3

AIM :: Design a web page of your hometown with an attractive background color, text color, an Image, font etc. (use internal CSS).

Theory ::

Designing a Web Page of Your Hometown with Internal CSS

Creating a webpage to represent a hometown involves showcasing key elements like the town's name, image, and relevant information, all while maintaining an appealing design. One effective way to style a webpage is by using **internal CSS**, which allows you to define the styles directly within the HTML document. This approach is particularly useful when you want to quickly style a single page or for small projects without creating separate external CSS files.

What is Internal CSS?

Internal CSS is placed within the `<style>` tag in the `<head>` section of an HTML document. The styles defined here apply only to that specific page. It is useful when you need specific styles for a page without the overhead of linking external stylesheets.

Key Design Elements for a Hometown Webpage

1. **Background Color**:
   The background color is the foundation of your page's visual appeal. You can choose colors that reflect the essence of your hometown or simply use vibrant, attractive colors to engage visitors. Using CSS, you can easily set the background of the entire page or specific sections (such as the header, main content area, or footer).

2. **Text Color**:
   To ensure readability and good contrast, it's important to choose a text color that stands out against the background. For example, light text on dark backgrounds or dark text on light backgrounds usually work well. CSS allows you to control the text color globally (for the body) or for individual elements (like headings, paragraphs, or links).

3. **Images**:
   A page about your hometown can greatly benefit from an image that showcases its beauty or landmarks. With CSS, you can style images, add borders, and control their size and positioning. You can also set a background image to enhance the overall design, making it more visually engaging.

4. **Fonts**:
   The choice of font plays a major role in the aesthetics of a page. You can use custom fonts through CSS by either importing from web font services like Google Fonts or using system fonts. Fonts can be styled to create emphasis, for example by setting headings in bold or large, and keeping body text simple and legible.

5. **Layout and Spacing**:
   CSS helps you structure the layout of your page. By using properties like padding,

5. **Layout and Spacing**:
   CSS helps you structure the layout of your page. By using properties like padding, margins, and border, you can create sufficient spacing between elements to ensure the content is visually appealing and not cluttered. For a hometown webpage, you might want to have a neat structure, where the introduction is at the top, followed by an image gallery or a paragraph of descriptive text about the town.

Advantages of Using Internal CSS for Webpage Design

- **Quick and Easy Setup**: Since the CSS is within the HTML file itself, there's no need to link separate stylesheets.
- **Simple for Small Projects**: Internal CSS is ideal for simple projects, such as a single-page town website.
- **Easy to Modify**: For a small website, having all styles in one document makes it easier to edit and tweak the design without switching between multiple files.

Conclusion

Using internal CSS allows you to quickly design and style a webpage for your hometown with minimal effort. It gives you full control over the layout, colors, fonts, and images, and helps create a visually pleasing and engaging web page. By considering the background, text, images, and fonts, you can build a page that captures the spirit of your hometown while offering a pleasant browsing experience to visitors.

Code ::

index.html

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

 <head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Welcome to Roorkee, Uttarakhand</title>
  <style>
   body {
    margin: 0;
    padding: 0;
    font-family: "Arial", sans-serif;
    background-color: #e6f7ff;
    color: #333;
   }

   header {
    background-color: #004d99;
    color: white;
```

```
  padding: 1rem 0;
  text-align: center;
  box-shadow: 0 4px 6px rgba(0, 0, 0, 0.2);
}

header h1 {
  margin: 0;
  font-size: 2.5rem;
}

.content {
  padding: 2rem;
  max-width: 80%;
  margin: auto;
}

.content h2 {
  font-size: 2rem;
  color: #004d99;
}

.content p {
  font-size: 1.2rem;
  line-height: 1.6;
  margin: 1rem 0;
  text-align: justify;
}

.image-right {
  float: right;
  width: 300px;
  height: 300px;
  object-fit: cover;
  margin: 0 0 1rem 1rem;
  border-radius: 10px;
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.3);
  transition: transform 0.8s ease;
}

.image-right:hover {
  transform: scale(1.05);
}
```

```
      .image-left {
        float: left;
        width: 300px;
        height: 300px;
        object-fit: cover;
        margin: 0 1rem 1rem 0;
        border-radius: 10px;
        box-shadow: 0 4px 8px rgba(0, 0, 0, 0.3);
        transition: transform 0.8s ease;
      }

      .image-left:hover {
        transform: scale(1.05);
      }

      footer {
        background-color: #004d99;
        color: white;
        padding: 1rem 0;
        text-align: center;
        margin-top: 2rem;
        font-size: 0.9rem;
        clear: both;
      }
    </style>
  </head>

  <body>
    <header>
      <h1>Welcome to Roorkee, Uttarakhand</h1>
    </header>

    <div class="content">
      <h2>About Roorkee</h2>
      <img
        src="https://ugcounselor-content.s3.ap-south-1.amazonaws.com/wp-content/uploads/
2024/02/03180310/IIT-Roorkee.jpg"
        alt="IIT Roorkee" class="image-right" />
      <p>Roorkee is a historically significant city located in the northern Indian state of
Uttarakhand, nestled in the
        fertile plains near the Himalayan foothills. Established during the British colonial period, the
city has
```

```
        transformed from a small settlement to an important educational and engineering hub of
national importance.
      </p>
      <br />

      <p>The city's most prominent institution, the Indian Institute of Technology Roorkee (IIT
Roorkee), stands as a
        testament to India's technological prowess. Founded in 1847 as the Roorkee College of
Engineering, it is one of
        the oldest technical institutions in Asia, predating India's independence and serving as a
crucial center for
        engineering education and research.
      </p>

      <img
        src="https://images.hindustantimes.com/rf/image_size_630x354/HT/p2/2020/07/05/
Pictures/_7f407ab4-beb1-11ea-8289-eb9e57c07d2f.JPG"
        alt="Ganga Canal, Roorkee" class="image-left" />
      <p>Roorkee's engineering legacy extends beyond academic institutions. The Ganga Canal, an
extraordinary hydraulic
        engineering project completed during British rule, dramatically transformed agricultural
practices in North
        India. This remarkable infrastructure project showcases the city's historical significance in
water management
        and irrigation technology.</p>

      <p>Geographically, Roorkee serves as a strategic gateway to some of Uttarakhand's most
revered destinations. Its
        proximity to Haridwar and Rishikesh makes it an ideal base for spiritual seekers, pilgrims,
and tourists
        interested in exploring the rich cultural and natural landscapes of the Himalayan region.</p>
    </div>

    <footer>&amp;copy; 2025 Roorkee, Uttarakhand | All Rights Reserved</footer>
  </body>

</html>
```

Output ::

Welcome to Roorkee, Uttarakhand

About Roorkee

Roorkee is a historically significant city located in the northern Indian state of Uttarakhand, nestled in the fertile plains near the Himalayan foothills. Established during the British colonial period, the city has transformed from a small settlement to an important educational and engineering hub of national importance.

The city's most prominent institution, the Indian Institute of Technology Roorkee (IIT Roorkee), stands as a testament to India's technological prowess. Founded in 1847 as the Roorkee College of Engineering, it is one of the oldest technical institutions in Asia, predating India's independence and serving as a crucial center for engineering education and research.

Roorkee's engineering legacy extends beyond academic institutions. The Ganga Canal, an extraordinary hydraulic engineering project completed during British rule, dramatically transformed agricultural practices in North India. This remarkable infrastructure project showcases the city's historical significance in water management and irrigation technology.

Geographically, Roorkee serves as a strategic gateway to some of Uttarakhand's most revered destinations. Its proximity to Haridwar and Rishikesh makes it an ideal base for spiritual seekers, pilgrims, and tourists interested in exploring the rich cultural and natural landscapes of the Himalayan region.

Experiment - 4

AIM :: Design Page for your college containing a description of the courses, dept. , faculties, library etc using href, list tags Using External, Internal, and Inline CSS ?

Theory ::

Designing a Web Page of Your Hometown with External CSS

HTML (HyperText Markup Language) is the backbone of web pages, defining their structure with elements like headings, paragraphs, and links. CSS (Cascading Style Sheets) enhances HTML by controlling the visual presentation, including colors, fonts, layouts, and responsiveness.

Types of CSS

1. **Inline CSS** – Applied directly within an HTML element using the `style` attribute.
2. **Internal CSS** – Defined within a `<style>` tag inside the HTML `<head>`.
3. **External CSS** – Written in a separate `.css` file and linked using a `<link>` tag.

External CSS is preferred for better maintainability and consistency across multiple web pages.

Code ::

index.html

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

  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="style.css">
    <title>MAIT College</title>
  </head>

  <body>
    <header>
      <div class="logo">
        <img src="MAIT.png" alt="College Logo">
      </div>
      <div class="title">
```

```
        <h1>Maharaja Agrasen Institute of Technology</h1>
        <p>Excellence in Education & Research</p>
      </div>
    </header>

    <nav>
      <ul>
        <li><a href="#home">Home</a></li>
        <li><a href="#about">About</a></li>
        <li><a href="#courses">Courses</a></li>
        <li><a href="#departments">Departments</a></li>
        <li><a href="#faculty">Faculty</a></li>
        <li><a href="#library">Library</a></li>
        <li><a href="#contact">Contact</a></li>
      </ul>
    </nav>

    <section id="about">
      <h2>About Us</h2>
      <p>
        Maharaja Agrasen Institute of Technology (MAIT) is a premier
engineering institution offering world-class
        education in various fields of technology and management. Established
with a mission to foster innovation
        and excellence, MAIT provides industry-relevant training and hands-on
experience to its students.
      </p>
    </section>

    <section id="courses">
      <h2>Courses Offered</h2>
      <div class="course-list">
        <div class="course">
          <h3>B.Tech in Computer Science</h3>
          <p>Duration: 4 years</p>
        </div>
        <div class="course">
          <h3>B.Tech in Electronics</h3>
          <p>Duration: 4 years</p>
        </div>
        <div class="course">
          <h3>MBA</h3>
          <p>Duration: 2 years</p>
```

```
        </div>
        <div class="course">
          <h3>B.Sc in Physics</h3>
          <p>Duration: 3 years</p>
        </div>
      </div>
    </section>

    <section id="departments">
      <h2>Departments</h2>
      <ul>
        <li><a href="#cse">Computer Science & Engineering</a></li>
        <li><a href="#ece">Electronics & Communication Engineering</a></li>
        <li><a href="#mech">Mechanical Engineering</a></li>
        <li><a href="#civil">Civil Engineering</a></li>
        <li><a href="#mba">Management Studies</a></li>
      </ul>
    </section>

    <section id="faculty">
      <h2>Our Faculty</h2>
      <ul>
        <li>Dr. Rajesh Sharma – HOD, Computer Science</li>
        <li>Prof. Anjali Verma – HOD, Electronics</li>
        <li>Dr. Ramesh Yadav – HOD, Mechanical</li>
        <li>Prof. Manisha Gupta – HOD, Civil</li>
        <li>Dr. Neha Kapoor – HOD, MBA</li>
      </ul>
    </section>

    <section id="library">
      <h2>Library</h2>
      <p>
        Our state-of-the-art library houses thousands of books, research
papers, and digital resources.
        Open from 8 AM to 10 PM daily.
      </p>
    </section>

    <section id="contact">
      <h2>Contact Us</h2>
      <p>Email: <a href="mailto:info@mait.edu">info@mait.edu</a></p>
      <p>Phone: <a href="tel:+919876543210">+91 9876543210</a></p>
```

```
    </section>

    <footer>
      <p>&copy; 2025 MAIT. All rights reserved.</p>
    </footer>
  </body>

</html>
```

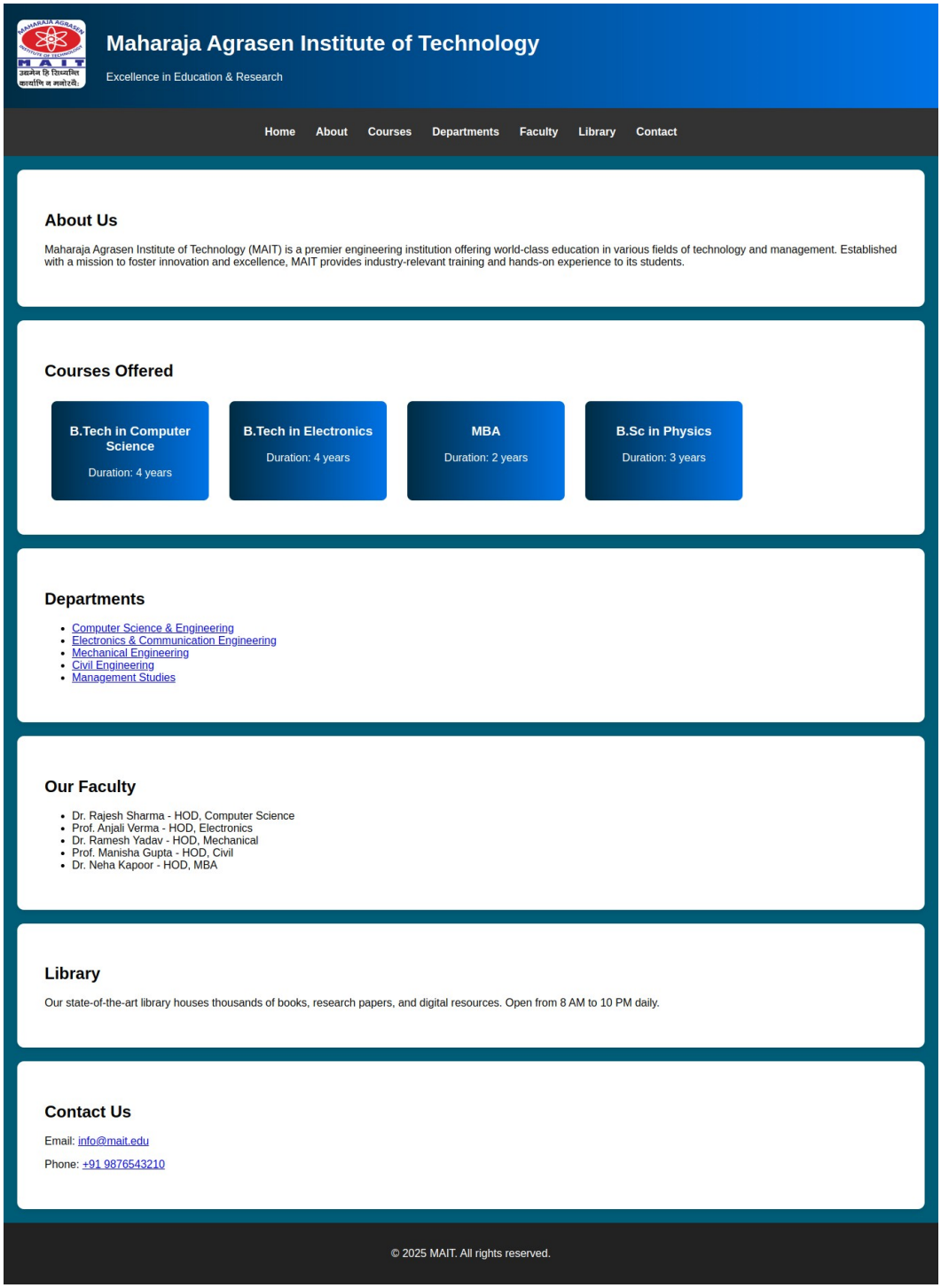
style.css

```
body {
  font-family: 'Arial', sans-serif;
  margin: 0;
  padding: 0;
  background-color: #005d78;
}

header {
  display: flex;
  align-items: center;
  background: linear-gradient(to right, #002d45, #0073e6);
  color: white;
  padding: 20px;
}

.logo img {
  width: 100px;
  height: 100px;
  border-radius: 10%;
}

.title {
  margin-left: 30px;
}

nav {
  background: #333;
  padding: 10px 0;
}

nav ul {
  list-style: none;
```

```
  display: flex;
  justify-content: center;
  padding: 0;
}

nav ul li {
  margin: 0 15px;
}

nav ul li a {
  color: white;
  text-decoration: none;
  font-weight: bold;
}

nav ul li a:hover {
  color: #f4a261;
}

section {
  padding: 40px;
  background: white;
  margin: 20px;
  border-radius: 10px;
  box-shadow: 0px 4px 8px rgba(0, 0, 0, 0.1);
}

.course-list {
  display: flex;
  justify-content: left;
  gap: 10px;
  flex-wrap: wrap;
}

.course {
  background: linear-gradient(to right, #002d45, #0073e6);
  color: white;
  padding: 15px;
  border-radius: 8px;
  width: 200px;
  text-align: center;
  justify-content: center;
  margin: 10px;
```

```
}

footer {
  background: #222;
  color: white;
  text-align: center;
  padding: 20px;
  margin-top: 20px;
}
```

Output ::



Experiment - 5

AIM :: Create HTML Page with JavaScript which takes Integer number as input and tells whether the number is ODD or EVEN?

Theory ::

A webpage is a digital document displayed in a web browser, typically built using three core technologies: HTML, CSS, and JavaScript. These three files work together to create a structured, styled, and interactive web experience.

1. HTML File (`index.html`)

HTML (HyperText Markup Language) provides the structure of a webpage using elements like headings, paragraphs, links, images, and forms. It defines the content but does not control styling or behavior.

2. CSS File (`style.css`)

CSS (Cascading Style Sheets) controls the visual appearance of the webpage, including colors, fonts, layouts, and animations. It ensures a responsive and visually appealing design.

3. JavaScript File (`script.js`)

JavaScript adds interactivity to the webpage. It enables dynamic content updates, user input handling, animations, and API interactions, making the webpage more functional and engaging.

Code ::

index.html

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

  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <link rel="stylesheet" href="style.css" />
    <title>Odd or Even Check</title>
  </head>

  <body>
```

```
    <header>
      <h1>Welcome to Odd/Even Checker</h1>
      <p class="subtitle">Find out whether your number is odd or even!</p>
    </header>
    <main>
      <section class="container">
        <h2>Check Your Number</h2>
        <p>Enter an integer number below:</p>
        <input type="text" id="numberInput" placeholder="Enter a number" />
        <button id="checkBtn">Check</button>
        <p id="result"></p>
      </section>
    </main>
    <footer>
      <p>&copy; 2025 singhal-amit | All Rights Reserved</p>
    </footer>
    <script src="script.js" defer></script>
  </body>
</html>
```

style.css

```
body {
  background: linear-gradient(to right, #1e3c72, #2a5298);
  font-family: "Arial", sans-serif;
  display: flex;
  justify-content: center;
  align-items: center;
  min-height: 100vh;
  flex-direction: column;
  text-align: center;
  color: white;
  margin: 0;
}

header {
  padding: 20px;
  border-radius: 10px;
  text-align: center;
}

.subtitle {
  font-size: 1.2em;
```

```
  color: #f39c12;
}

main {
  width: 90%;
  max-width: 500px;
  background-color: white;
  padding: 30px;
  border-radius: 12px;
  box-shadow: 0 8px 16px rgba(0, 0, 0, 0.2);
  color: #333;
}

input {
  width: 90%;
  padding: 12px;
  margin-bottom: 15px;
  border: 2px solid #34495e;
  border-radius: 8px;
  font-size: 1.1em;
}

button {
  width: 100%;
  padding: 12px;
  background: #3498db;
  color: white;
  font-size: 1.2em;
  border: none;
  border-radius: 8px;
  cursor: pointer;
  transition: 0.3s ease-in-out;
}

button:hover {
  background: #2980b9;
}

#result {
  font-size: 1.3em;
  margin-top: 20px;
  font-weight: bold;
}
```

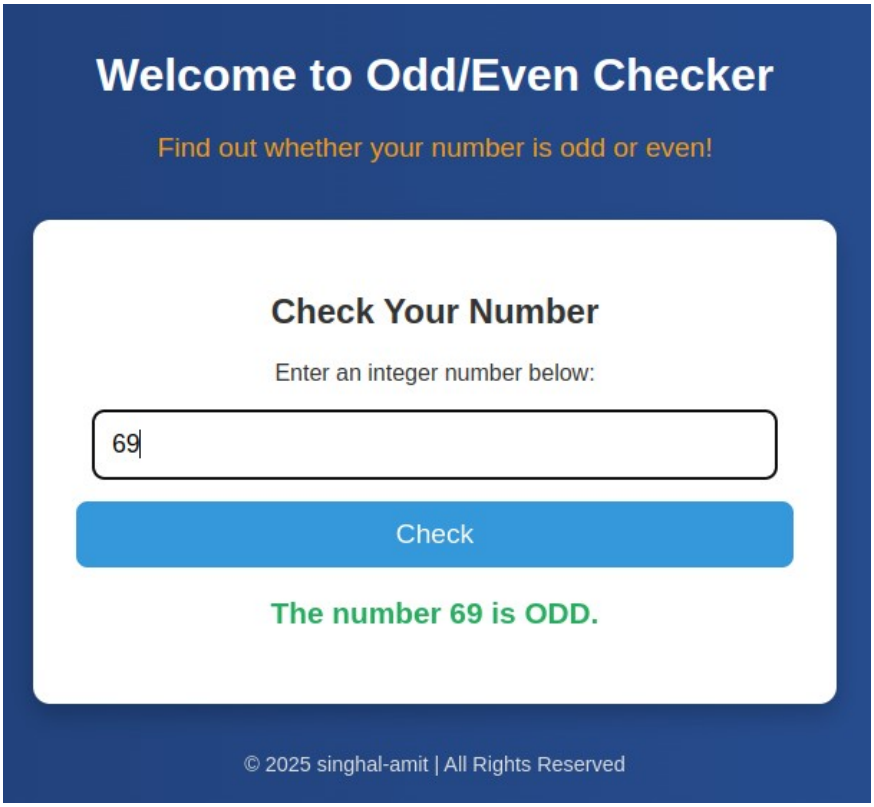
```
/* Result success and error styles */
#result.success {
  color: #27ae60;
}

#result.error {
  color: #c0392b;
}

footer {
  margin-top: 20px;
  font-size: 0.9em;
  color: #ddd;
}
```

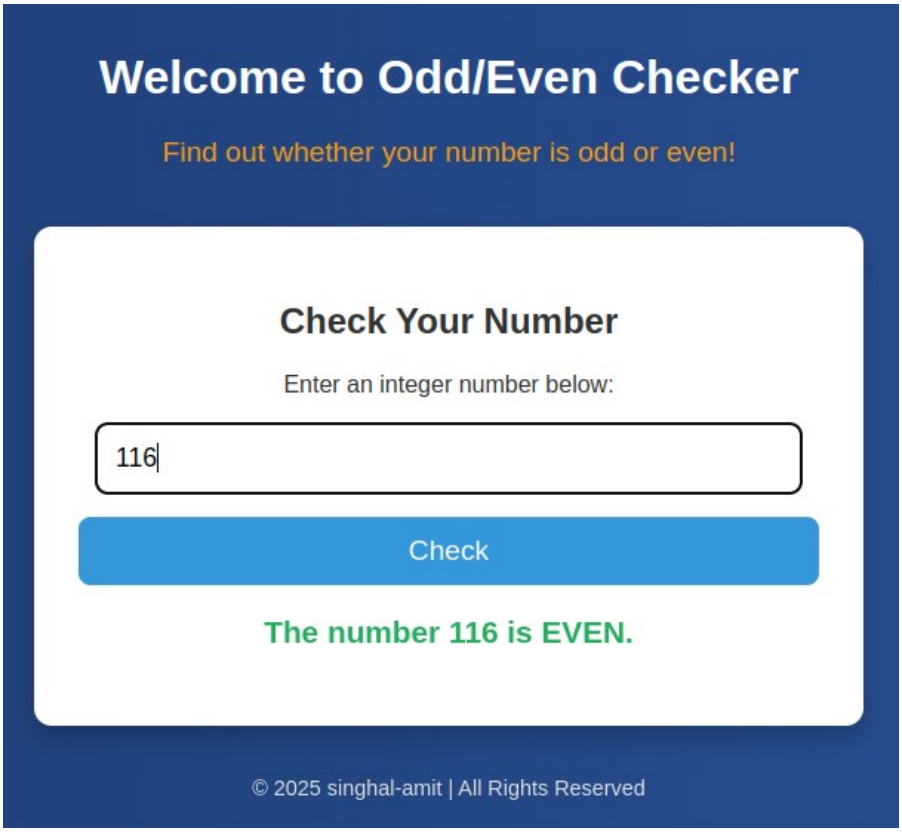
script.js

```
document.addEventListener( "DOMContentLoaded", function ()
{
  const checkBtn = document.getElementById( "checkBtn" );
  checkBtn.addEventListener( "click", checkOddOrEven );
} );

function checkOddOrEven ()
{
  let numberInput = document.getElementById( "numberInput" );
  let resultText = document.getElementById( "result" );
  let number = parseInt( numberInput.value );

  if ( isNaN( number ) )
  {
    resultText.textContent = "Please enter a valid number.";
    resultText.className = "error";
  } else
  {
    if ( number % 2 === 0 )
    {
      resultText.textContent = `The number ${ number } is EVEN.`;
      resultText.className = "success";
    } else
    {
      resultText.textContent = `The number ${ number } is ODD.`;
```

```
        resultText.className = "success";
      }
    }

    // Clear input after checking
    numberInput.value = "";
}
```

Output ::





Experiment - 6

AIM :: Create HTML Page that contains form with fields Name, Email, Mobile No, Gender , Fav. Colour and a button To Show JS code to display the information in textbox.

Theory ::

Forms in HTML allow users to input and submit data to a server. They are essential for collecting user information, such as login credentials, feedback, or search queries. Forms are created using the `<form>` element, which contains input fields, labels, and buttons.

Basic Fields in an HTML Form

1. **Text Input (**`<input type="text">`**)** – Allows users to enter a single line of text.
2. **Password Field (**`<input type="password">`**)** – Masks input for secure password entry.
3. **Email Field (**`<input type="email">`**)** – Validates email format.
4. **Number Field (**`<input type="number">`**)** – Accepts only numerical values.
5. **Radio Buttons (**`<input type="radio">`**)** – Allows users to select one option from multiple choices.
6. **Checkbox (**`<input type="checkbox">`**)** – Enables multiple selections.
7. **Dropdown (**`<select>` **and** `<option>`**)** – Provides a list of options in a dropdown menu.
8. **Textarea (**`<textarea>`**)** – Allows multi-line text input.
9. **Submit Button (**`<input type="submit">`**)** – Sends form data to the server.

Basic Structure of an HTML Form

```
<form action="submit.php" method="post">
    <label for="name">Name:</label>
    <input type="text" id="name" name="name" required>

    <label for="email">Email:</label>
    <input type="email" id="email" name="email" required>

    <label for="message">Message:</label>
    <textarea id="message" name="message"></textarea>

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

The `action` attribute specifies where to send the form data, and the `method` determines how it is sent (commonly `GET` or `POST`).

Code ::

index.html

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

  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Personal Information Form</title>
    <link rel="stylesheet" href="styles.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/bootstrap/5.3.0/css/bootstrap.mi
n.css">
  </head>

  <body class="bg-light">
    <div class="container mt-5">
      <h1 class="text-center mb-4">Personal Information Form</h1>
      <form id="personalForm" novalidate>
        <div class="row mb-3">
          <div class="col-md-6">
            <label for="name" class="form-label">Name:</label>
            <input type="text" id="name" name="name" class="form-control"
placeholder="Enter your name" required>
            <div class="invalid-feedback">Please enter your name.</div>
          </div>
          <div class="col-md-6">
            <label for="email" class="form-label">Email:</label>
            <input type="email" id="email" name="email" class="form-control"
placeholder="Enter your email" required>
            <div class="invalid-feedback">Please enter a valid email.</div>
          </div>
        </div>

        <div class="row mb-3">
          <div class="col-md-6">
            <label for="mobile" class="form-label">Mobile No:</label>
            <input type="tel" id="mobile" name="mobile" class="form-control"
placeholder="Enter your mobile number"
              required>
```

```
            <div class="invalid-feedback">Please enter a valid mobile
number.</div>
          </div>
          <div class="col-md-6">
            <label class="form-label">Gender:</label><br>
            <div class="form-check form-check-inline">
              <input type="radio" id="male" name="gender" value="Male"
class="form-check-input" required>
              <label for="male" class="form-check-label">Male</label>
            </div>
            <div class="form-check form-check-inline">
              <input type="radio" id="female" name="gender" value="Female"
class="form-check-input" required>
              <label for="female" class="form-check-label">Female</label>
            </div>
            <div class="invalid-feedback d-block">Please select your
gender.</div>
          </div>
        </div>

        <div class="mb-3">
          <label for="color" class="form-label">Favourite Colour:</label>
          <input type="text" id="color" name="color" class="form-control"
placeholder="Enter your favourite colour"
            required>
          <div class="invalid-feedback">Please enter your favourite
colour.</div>
        </div>

        <button type="submit" class="btn btn-outline-danger mt-3
w-100">Submit</button>
      </form>

      <div class="mt-4 mb-4">
        <h3>Form Information:</h3>
        <textarea id="result" rows="5" class="form-control"
readonly></textarea>
      </div>
    </div>

    <script src="script.js"></script>
  </body>
```

```
</html>
```

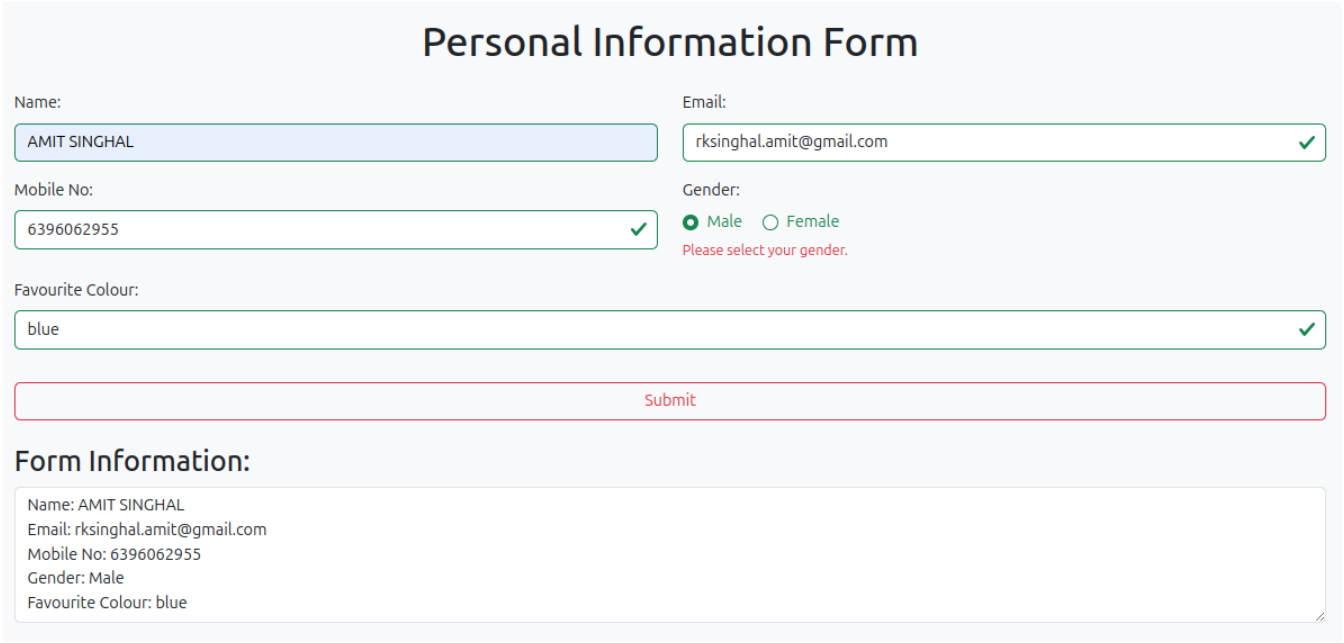
style.css

```
body {
  background-color: #f8f9fa;
  color: #333;
  font-family: Arial, sans-serif;
}

h1 {
  font-size: 2rem;
  font-weight: bold;
  color: #1e3c72;
}

.container {
  background: white;
  padding: 30px;
  border-radius: 10px;
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);
}

button {
  transition: all 0.3s ease-in-out;
}

button:hover {
  background: #dc3545;
  color: white;
}

textarea {
  font-family: Arial, sans-serif;
  resize: none;
}

.invalid-feedback {
  font-size: 0.9rem;
}
```

script.js

```
document.getElementById( "personalForm" ).addEventListener( "submit",
function ( event )
{
  event.preventDefault(); // Prevent default form submission

  let isValid = true;
  const form = event.target;

  // Loop through inputs and validate them
  form.querySelectorAll( "input" ).forEach( input ⇒
  {
    if ( !input.checkValidity() )
    {
      input.classList.add( "is-invalid" );
      isValid = false;
    } else
    {
      input.classList.remove( "is-invalid" );
      input.classList.add( "is-valid" );
    }
  } );

  if ( !isValid ) return;

  // Get values from form fields
  const name = document.getElementById( "name" ).value;
  const email = document.getElementById( "email" ).value;
  const mobile = document.getElementById( "mobile" ).value;
  const gender =
document.querySelector( 'input[name="gender"]:checked' )?.value;
  const color = document.getElementById( "color" ).value;

// Display the result in the textarea
  document.getElementById( "result" ).value = `
Name: ${ name }
Email: ${ email }
Mobile No: ${ mobile }
Gender: ${ gender }
Favourite Colour: ${ color }
    `.trim();
} );
```

Output ::


Personal Information Form
Name:
AMIT SINGHAL
Email:
rksinghal.amit@gmail.com
Mobile No:
6396062955
Gender:
Male
Female
Please select your gender.
Favourite Colour:
blue
Submit
Form Information:
Name: AMIT SINGHAL
Email: rksinghal.amit@gmail.com
Mobile No: 6396062955
Gender: Male
Favourite Colour: blue

Experiment - 7

AIM :: Create XML file to store student info. like Enrolment No., Name Mobile No. , Email Id.

Theory ::

XML (Extensible Markup Language) is a flexible, text-based language used to store, transport, and organize data. It is designed to be both human-readable and machine-readable, making it widely used for data exchange between systems, configuration files, and web services.

Key Features of XML

- **Self-descriptive** – Uses custom tags to define data structure.
- **Hierarchical Structure** – Data is organized in a tree-like format with parent and child elements.
- **Platform-Independent** – Works across different software and hardware environments.
- **Extensible** – No predefined tags; users can define their own.

Basic XML Structure

```
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>
    <book>
        <title>XML Basics</title>
        <author>John Doe</author>
        <price>19.99</price>
    </book>
</bookstore>
```

XML uses tags (`<title>`, `<author>`) to define data, making it structured and easy to parse. Unlike HTML, XML does not have predefined tags, giving developers flexibility to create custom data formats.

Code ::

index.html

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

  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Student List</title>
```

```
    <link rel="stylesheet" href="style.css">
  </head>

  <body>
    <div class="container">
      <h1>Student Information</h1>
      <table>
        <thead>
          <tr>
            <th>Enrolment Number</th>
            <th>Name</th>
            <th>Mobile Number</th>
            <th>Email ID</th>
          </tr>
        </thead>
        <tbody id="studentTable"></tbody>
      </table>
    </div>
    <script src="script.js"></script>
  </body>

</html>
```

style.css

```
body {
  font-family: Arial, sans-serif;
  background-color: #f5f5f5;
  margin: 0;
  padding: 0;
  text-align: center;
}

.container {
  max-width: 80%;
  margin: 50px auto;
  background: white;
  padding: 20px;
  border-radius: 10px;
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);
}

h1 {
```

```
  color: #2c3e50;
}

table {
  width: 100%;
  border-collapse: collapse;
  margin-top: 20px;
}

th, td {
  border: 1px solid #ddd;
  padding: 12px;
  text-align: center;
}

th {
  background-color: #2c3e50;
  color: white;
}

tr:nth-child(even) {
  background-color: #f2f2f2;
}

tr:hover {
  background-color: #dcdde1;
  transition: 0.3s ease;
}
```

script.js

```
document.addEventListener( "DOMContentLoaded", function ()
{
  loadXML();
} );

async function loadXML ()
{
  try
  {
    let response = await fetch( "students.xml" );
    if ( !response.ok ) throw new Error( "Failed to fetch student data." );
```

```
    let xmlText = await response.text();
    let parser = new DOMParser();
    let xmlDoc = parser.parseFromString( xmlText, "text/xml" );
    displayData( xmlDoc );
  } catch ( error )
  {
    console.error( "Error loading XML:", error );
  }
}

function displayData ( xml )
{
  let table = document.getElementById( "studentTable" );
  let students = xml.getElementsByTagName( "Student" );

  Array.from( students ).forEach( student ⇒
  {
    let row = document.createElement( "tr" );

    [ "EnrolmentNumber", "Name", "MobileNumber", "EmailId" ].forEach( tag ⇒
    {
      let cell = document.createElement( "td" );
      cell.textContent = student.getElementsByTagName( tag )
[ 0 ].textContent;
      row.appendChild( cell );
    } );

    table.appendChild( row );
  } );
}
```

students.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<Students>
  <Student>
    <EnrolmentNumber>001</EnrolmentNumber>
    <Name>Amit</Name>
    <MobileNumber>123421</MobileNumber>
    <EmailId>abc@example.com</EmailId>
  </Student>

  <Student>
```

```
    <EnrolmentNumber>002</EnrolmentNumber>
    <Name>Yash</Name>
    <MobileNumber>567822</MobileNumber>
    <EmailId>def@example.com</EmailId>
  </Student>

  <Student>
    <EnrolmentNumber>003</EnrolmentNumber>
    <Name>Shaswat</Name>
    <MobileNumber>91011</MobileNumber>
    <EmailId>ghi@example.com</EmailId>
  </Student>

  <Student>
    <EnrolmentNumber>004</EnrolmentNumber>
    <Name>Lakshya</Name>
    <MobileNumber>121314</MobileNumber>
    <EmailId>jkl@example.com</EmailId>
  </Student>

  <Student>
    <EnrolmentNumber>005</EnrolmentNumber>
    <Name>Priyanshu</Name>
    <MobileNumber>151617</MobileNumber>
    <EmailId>mno@example.com</EmailId>
  </Student>

  <Student>
    <EnrolmentNumber>006</EnrolmentNumber>
    <Name>Gautam</Name>
    <MobileNumber>181920</MobileNumber>
    <EmailId>pqr@example.com</EmailId>
  </Student>

</Students>
  loadXML();
} );
```

Output ::

Student Information

Enrolment Number	Name	Mobile Number	Email ID
001	Amit	123421	abc@example.com
002	Yash	567822	def@example.com
003	Shaswat	91011	ghi@example.com
004	Lakshya	121314	jkl@example.com
005	Priyanshu	151617	mno@example.com
006	Gautam	181920	pqr@example.com

Experiment - 8

AIM :: Write a php script to read data from txt file and display it in html table (the file contains info in format - Name: Password: Email)

Theory ::

PHP File Reading and HTML Table Display

PHP is a server-side scripting language ideal for web development. It can read files, process content, and generate dynamic HTML. By using functions like `file()` and `explode()`, PHP can extract structured data from a text file and display it in an HTML table format.

Key Features of PHP File Handling

- File Access – Use `fopen()`, `file()`, or `file_get_contents()` to read content.
- Line Parsing – Split strings using `explode()` to separate values.
- Dynamic Output – Generate HTML content programmatically.
- Server-side Processing – Ensures secure and efficient data handling.

Basic PHP File Processing Example

```
$lines = file("data.txt");
foreach ($lines as $line) {
    list($name, $pass, $email) = explode(":", $line);
    echo "<tr><td>$name</td><td>$pass</td><td>$email</td></tr>";
}
```

PHP reads each line of the file, splits it into components using the colon : as a delimiter, and dynamically generates table rows in HTML. This approach is efficient for displaying structured text data on a web page.

Code ::

index.php

```
<?php
// Read the contents of the file into an array, each line represents a record
$filename = 'data.txt';
$lines = file($filename, FILE_IGNORE_NEW_LINES | FILE_SKIP_EMPTY_LINES);
// Function to parse each line into an associative array
function parseLine($line)
{
  $data = [];
```

```
  preg_match('/Name:\s*(.*?)\s*Password:\s*(.*?)\s*Email:\s*(.*)/', $line,
$matches);
  if (count($matches) === 4) {
    $data['name'] = $matches[1];
    $data['password'] = $matches[2];
    $data['email'] = $matches[3];
  }
  return $data;
}
?>
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>User Information</title>
  <style>
    table {
      width: 100%;
      border-collapse: collapse;
    }

    table,
    th,
    td {
      border: 1px solid black;
    }

    th,
    td {
      padding: 8px;
      text-align: left;
    }
  </style>
</head>

<body>
  <h1>User Information</h1>
  <table>
    <thead>
      <tr>
        <th>Name</th>
```

```
        <th>Password</th>
        <th>Email</th>
      </tr>
    </thead>
    <tbody>
      <?php
      // Loop through the lines from the file and parse them into a table
      foreach ($lines as $line) {
        $data = parseLine($line);
        if ($data) {
          echo "<tr>";
          echo "<td>" . htmlspecialchars($data['name']) . "</td>";
          echo "<td>" . htmlspecialchars($data['password']) . "</td>";
          echo "<td>" . htmlspecialchars($data['email']) . "</td>";
          echo "</tr>";
        }
      }
      ?>
    </tbody>
  </table>
</body>

</html>
```

data.txt

```
Name: John Doe Password: secret123 Email: john@example.com
Name: Jane Smith Password: 1234password Email: jane@example.com
```

Output ::

User Information

Name	Password	Email
John Doe	secret123	john@example.com
Jane Smith	1234password	jane@example.com

Experiment - 9

AIM :: Write a PHP Script for login authentication. Design an html form which takes username and password from user and validate against stored username and password in file.

Theory ::

Login Authentication Using PHP and Text File

Login authentication is the process of verifying a user's credentials to allow access to a restricted area or system. PHP, a powerful server-side scripting language, can be used to handle form submissions and compare user inputs against stored data.

Key Concepts of PHP Login Authentication

• **Form Handling** – PHP captures input from HTML forms using `$_POST` or `$_GET`.
• **File Access** – Credentials are stored in a `.txt` file, with each line containing a username and password pair.
• **String Parsing** – PHP uses functions like `explode()` to split and compare username-password pairs.
• **Validation** – The script checks if the input matches any record in the file and grants or denies access accordingly.

Security Considerations

• Plain text passwords are insecure; storing hashed passwords is safer.
• Input validation is important to prevent injection attacks.
• Error messages should be generic to avoid revealing valid usernames.

This method is suitable for small-scale applications or learning purposes. For real-world systems, databases and secure hashing mechanisms are preferred.

Code ::

index.php

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

  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login Form</title>
  </head>

  <body>
```

```
    <h2>Login</h2>
    <form action="login.php" method="POST">
      <label for="username">Username:</label>
      <input type="text" id="username" name="username" required>
      <br><br>
      <label for="password">Password:</label>
      <input type="password" id="password" name="password" required>
      <br><br>
      <input type="submit" value="Login">
    </form>

    <?php
    // Function to validate the user
    function validate_user($username, $password)
    {
      $file = fopen("users.txt", "r");
      if (!$file) {
        die("Unable to open file!");
      }
      while ($line = fgets($file)) {
        $line = trim($line);
        list($stored_username, $stored_password) = explode(":", $line);
        if ($username === $stored_username && $password === $stored_password)
{
          fclose($file);
          return true;
        }
      }
      fclose($file);
      return false;
    }

    // Handle form submission
    if ($_SERVER["REQUEST_METHOD"] === "POST") {
      $username = $_POST["username"];
      $password = $_POST["password"];

      if (validate_user($username, $password)) {
        echo "<p>Login successful! Welcome, " . htmlspecialchars($username) .
".</p>";
      } else {
        echo "<p style='color:red;'>Invalid username or password.</p>";
      }
```

```
    }
    ?>
  </body>

</html>
```

users.txt

```
Amit:test1234
```

Output ::

Experiment - 10

AIM :: Write PHP Script for storing and retrieving user information from MySql table

1) Design A HTML page which takes Name, Address, Email and Mobile No. From user (register.php)

2) Store this data in Mysql database / text file.

3) Next page display all user in html table using PHP (display.php)

Theory ::

Storing and Retrieving User Information Using PHP and MySQL

PHP and MySQL are commonly used together to create dynamic, data-driven web applications. PHP handles form submissions and database interactions, while MySQL is used to store and manage user data efficiently.

Key Concepts

1. HTML Form Handling

- HTML forms collect user inputs such as Name, Address, Email, and Mobile No.
- The form uses the `POST` method to send data securely to a PHP script.

2. Data Storage (register.php)

- PHP receives form data using `$_POST`.
- A connection to the MySQL database is established using `mysqli_connect()` or PDO.
- The form data is inserted into a table using an `INSERT` SQL query.
- Input validation and SQL injection protection (like prepared statements) are recommended for security.

3. Data Retrieval (display.php)

- PHP executes a `SELECT` SQL query to fetch user records from the MySQL table.
- Retrieved data is displayed in an HTML table using loops.

Benefits

- Centralized data storage in MySQL enables easy data management.
- PHP automates data collection and display.
- The combination of PHP and MySQL supports real-time, interactive web applications.

This process is foundational in creating user registration systems, content management systems, and other web applications that require persistent data storage and display.

Code ::

mysql query

```
CREATE DATABASE user_registration;

USE user_registration;

CREATE TABLE users (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(255) NOT NULL,
    address TEXT NOT NULL,
    email VARCHAR(255) NOT NULL,
    mobile VARCHAR(20) NOT NULL
);

EXIT;
```

register.php

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>User Registration</title>
</head>
<body>
<h2>User Registration</h2>
<form action="register.php" method="POST">
  <label for="name">Name:</label>
  <input type="text" id="name" name="name" required><br><br>

  <label for="address">Address:</label>
  <textarea id="address" name="address" required></textarea><br><br>

  <label for="email">Email:</label>
  <input type="email" id="email" name="email" required><br><br>

  <label for="mobile">Mobile No:</label>
  <input type="text" id="mobile" name="mobile" required><br><br>
```

```
  <input type="submit" value="Register">
</form>

<?php
// Database config
$host = "localhost";
$username = "root";
$password = "";
$dbname = "user_registration";

// Connect to DB
$conn = new mysqli($host, $username, $password, $dbname);
if ($conn->connect_error) {
  die("Connection failed: " . $conn->connect_error);
}

// Handle form submission
if ($_SERVER["REQUEST_METHOD"] == "POST") {
  $name = $_POST['name'];
  $address = $_POST['address'];
  $email = $_POST['email'];
  $mobile = $_POST['mobile'];

  $sql = "INSERT INTO users (name, address, email, mobile) VALUES
(?, ?, ?, ?)";
  $stmt = $conn->prepare($sql);
  $stmt->bind_param("ssss", $name, $address, $email, $mobile);

  if ($stmt->execute()) {
    echo "<p>Registration successful!</p>";
  } else {
    echo "<p>Error: " . $stmt->error . "</p>";
  }

  $stmt->close();
}

// Display all users
$sql = "SELECT id, name, address, email, mobile FROM users";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
  echo "<h2>User Information</h2>";
```

```
  echo "<table
border='1'><tr><th>ID</th><th>Name</th><th>Address</th><th>Email</
th><th>Mobile</th></tr>";

  while ($row = $result->fetch_assoc()) {
    echo "<tr>
      <td>{$row['id']}</td>
      <td>{$row['name']}</td>
      <td>{$row['address']}</td>
      <td>{$row['email']}</td>
      <td>{$row['mobile']}</td>
    </tr>";
  }

  echo "</table>";
} else {
  echo "<p>No users found.</p>";
}

$conn->close();
?>
</body>
</html>
```

Output ::

User Information

ID	Name	Address	Email	Mobile
1	Amit Singhal	Rohini, Delhi	abc@example.com	123-456-789

Experiment - 11

AIM :: Create a Basic Calculator App using JS.

Code :

index.html

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

  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet" />
    <link rel="stylesheet" href="style.css" />
    <title>Basic Calc</title>
  </head>

  <body>
    <div class="container mt-5">
      <h1 class="text-center mb-4">Simple Calculator</h1>
      <div class="calculator bg-light p-3">
        <div class="display" id="Text-Field">
          <span class="result" id="Result-Part"></span>
          <span class="expression" id="Expr-Part"></span>
        </div>
        <div class="row g-2">
          <div class="col-3">
            <button class="btn btn-light btn-custom" onclick="ClearText()">
              AC
            </button>
          </div>
          <div class="col-3">
            <button class="btn btn-light btn-custom" onclick="ClearLastCharacter()">
              ⇐
            </button>
          </div>
          <div class="col-3">
            <button class="btn btn-light btn-custom" onclick="SquareValue()">
              x<sup>2</sup>
            </button>
```

```
</div>
<div class="col-3">
  <button class="btn btn-light btn-custom" onclick="AddNumber('/')">
    /
  </button>
</div>

<div class="col-3">
  <button class="btn btn-light btn-custom" onclick="AddNumber(7)">
    7
  </button>
</div>
<div class="col-3">
  <button class="btn btn-light btn-custom" onclick="AddNumber(8)">
    8
  </button>
</div>
<div class="col-3">
  <button class="btn btn-light btn-custom" onclick="AddNumber(9)">
    9
  </button>
</div>
<div class="col-3">
  <button class="btn btn-light btn-custom" onclick="AddNumber('*')">
    *
  </button>
</div>

<div class="col-3">
  <button class="btn btn-light btn-custom" onclick="AddNumber(4)">
    4
  </button>
</div>
<div class="col-3">
  <button class="btn btn-light btn-custom" onclick="AddNumber(5)">
    5
  </button>
</div>
<div class="col-3">
  <button class="btn btn-light btn-custom" onclick="AddNumber(6)">
    6
  </button>
</div>
```

```
<div class="col-3">
  <button class="btn btn-light btn-custom" onclick="AddNumber('-')">
    -
  </button>
</div>

<div class="col-3">
  <button class="btn btn-light btn-custom" onclick="AddNumber(1)">
    1
  </button>
</div>
<div class="col-3">
  <button class="btn btn-light btn-custom" onclick="AddNumber(2)">
    2
  </button>
</div>
<div class="col-3">
  <button class="btn btn-light btn-custom" onclick="AddNumber(3)">
    3
  </button>
</div>
<div class="col-3">
  <button class="btn btn-light btn-custom" onclick="AddNumber('+')">
    +
  </button>
</div>

<div class="col-3">
  <button class="btn btn-light btn-custom" onclick="AddNumber(0)">
    0
  </button>
</div>
<div class="col-3">
  <button class="btn btn-light btn-custom" onclick="AddNumber('00')">
    00
  </button>
</div>
<div class="col-3">
  <button class="btn btn-light btn-custom" onclick="AddNumber('.')">
    .
  </button>
</div>
<div class="col-3">
```

```
            <button class="btn btn-primary btn-custom" onclick="Evaluate()">
              =
            </button>
          </div>
        </div>
      </div>
    </div>
    <script src="script.js"></script>
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></
script>
  </body>

</html>
```

style.css

```
body {
  background-color: #f2f4f8;
  font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
}

.calculator {
  max-width: 350px;
  margin: auto;
  background-color: #ffffff;
  padding: 20px;
  border-radius: 20px;
  box-shadow: 0 10px 25px rgba(0, 0, 0, 0.1);
}

.display {
  height: 60px;
  background-color: #f1f3f4;
  color: #000;
  font-size: 22px;
  font-weight: bold;
  padding: 10px 15px;
  border: none;
  border-radius: 10px;
  text-align: left;
  display: flex;
  justify-content: space-between;
```

```
  align-items: center;
}

.display .result {
  color: #000;
  font-weight: bold;
}

.display .expression {
  color: #888;
  font-size: 18px;
}

.btn-custom {
  width: 100%;
  height: 60px;
  font-size: 22px;
  font-weight: bold;
  border-radius: 12px;
  transition: background-color 0.2s ease;
}

.btn-custom:hover {
  background-color: #dcdcdc;
}

.btn-primary.btn-custom {
  background-color: #0d6efd;
  color: #fff;
  border: none;
}

.btn-primary.btn-custom:hover {
  background-color: #0b5ed7;
}
```

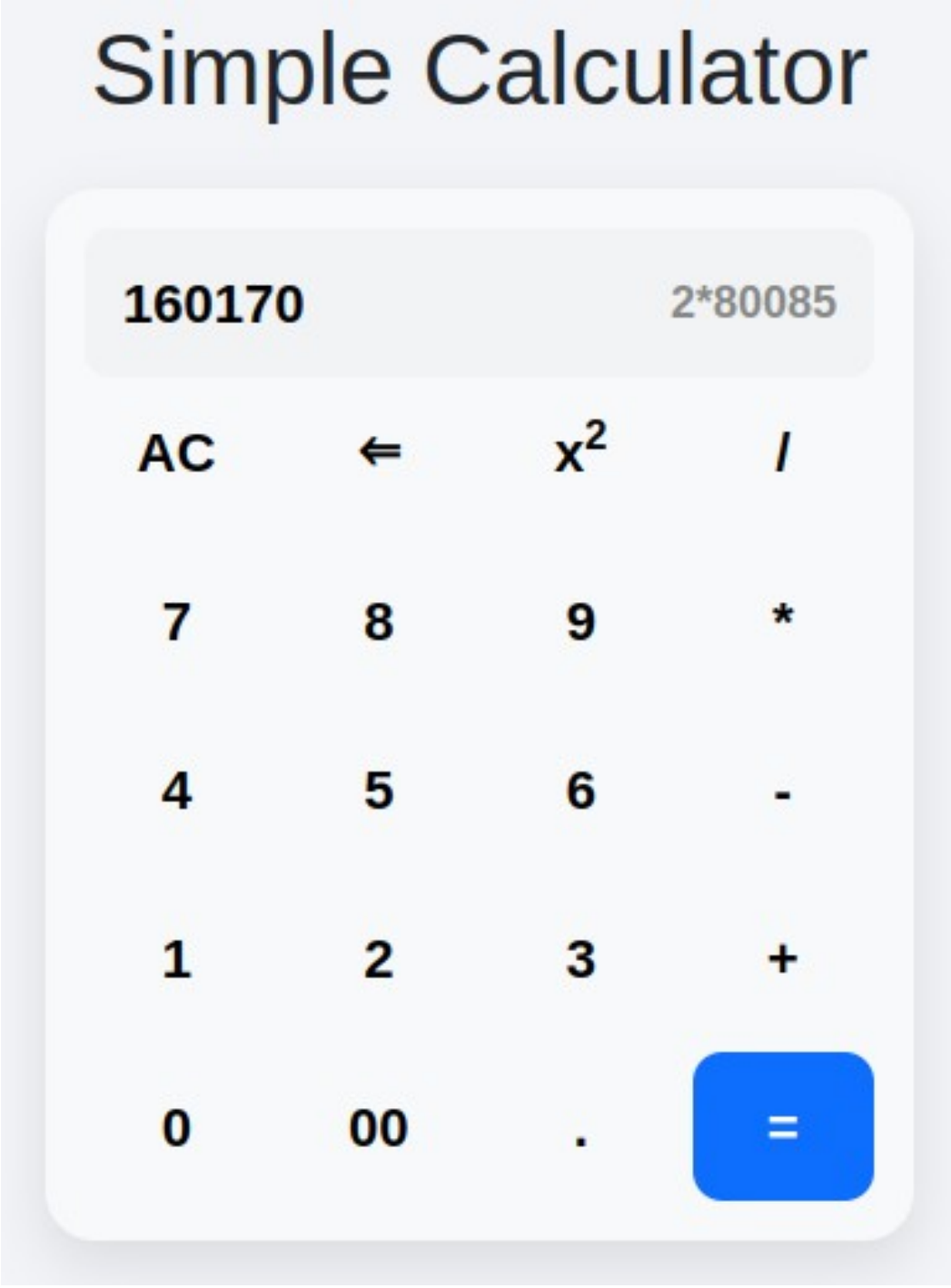
script.js

```
let expression = "";

function AddNumber ( value )
{
  expression += value;
```

```
  UpdateDisplay();
}

function ClearText ()
{
  expression = "";
  UpdateDisplay();
}

function ClearLastCharacter ()
{
  expression = expression.slice( 0, -1 );
  UpdateDisplay();
}

function SquareValue ()
{
  try
  {
    const squared = Math.pow( eval( expression ), 2 );
    expression = squared.toString();
    UpdateDisplay();
  } catch
  {
    expression = "";
    UpdateDisplay( "Error" );
  }
}

function Evaluate ()
{
  try
  {
    const result = eval( expression );
    UpdateDisplay( result );
    expression = result.toString(); // to continue calculations
  } catch
  {
    UpdateDisplay( "Error" );
    expression = "";
  }
}
```

```
function UpdateDisplay ( result = "" )
{
  const resultSpan = document.getElementById( "Result-Part" );
  const exprSpan = document.getElementById( "Expr-Part" );

  resultSpan.textContent = result =/= "" ? result : "";
  exprSpan.textContent = expression;
}
```

Output ::



Experiment - 12

AIM :: Create a Basic Quiz App using JS.

Code :

index.html

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

  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <title>Web Dev Quiz</title>
    <!— Bootstrap CSS —>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
rel="stylesheet" />
    <link rel="stylesheet" href="style.css">
  </head>

  <body>
    <div class="container">
      <!— Heading on top —>
      <h1>Web Dev Quiz</h1>
      <!— Quiz Form —>
      <form id="quizForm">
        <div id="quizContainer">
          <!— Quiz questions will be built here by JS —>
        </div>
        <button type="submit" class="btn btn-primary btn-submit">Submit Quiz</button>
      </form>
    </div>
    <script src="script.js"></script>
  </body>

</html>
```

result.html

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

```
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <title>Quiz Results</title>
    <!— Bootstrap CSS —>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
rel="stylesheet" />
    <link rel="stylesheet" href="style.css" />
  </head>

  <body>
    <div class="container">
      <div id="resultSummary" class="result-summary"></div>
      <div id="detailedResults"></div>
      <button class="btn btn-secondary btn-retake"
onclick="window.location.href='index.html'">Retake Quiz</button>
    </div>
    <script>

      // The same questions data is used on the results page to build the detailed review.
      const questions = [
        {
          q: "Which tag is used to create a hyperlink in HTML?",
          options: { "1": "&lt;a&gt;", "2": "&lt;link&gt;", "3": "&lt;href&gt;", "4":
"&lt;hyperlink&gt;" },
          correct: "1"
        },
        {
          q: "Which CSS property is used to change the background color of an element?",
          options: { "1": "background-color", "2": "color", "3": "bgcolor", "4":
"background" },
          correct: "1"
        },
        {
          q: "What does the 'id' attribute in HTML do?",
          options: { "1": "Defines the style of an element", "2": "Identifies a unique
element", "3": "Links the element to a CSS class", "4": "Creates a new HTML element" },
          correct: "2"
        },
        {
          q: "How do you add a comment in JavaScript?",
          options: { "1": "/* comment */", "2": "// comment", "3": "&lt;!-- comment --
&gt;", "4": "# comment" },
```

```
          correct: "2"
        },
        {
          q: "Which HTML element is used to define a paragraph?",
          options: { "1": "&lt;p&gt;", "2": "&lt;h1&gt;", "3": "&lt;div&gt;", "4":
"&lt;span&gt;" },
          correct: "1"
        }
      ];

      const storedResults = localStorage.getItem( "quizResults" );
      if ( !storedResults )
      {
        document.getElementById( "resultSummary" ).innerHTML = "<p>No quiz results found.
Please take the quiz first.</p>";
      } else
      {
        const results = JSON.parse( storedResults );
        const totalQuestions = questions.length;
        const maxScore = totalQuestions * 10;
        const userScore = results.correctCount * 10; // 10 points for each correct answer
        const scorePercentage = ( userScore / maxScore ) * 100;

        const summaryHTML = `
        <h2>Quiz Results</h2>
        <div class="score">Score: ${ userScore }/${ maxScore } ($
{ scorePercentage.toFixed( 1 ) }%)</div>
        <p>Total Questions: ${ totalQuestions }</p>
        <p>Correct Answers: ${ results.correctCount } (10 marks each)</p>
        <p>Incorrect Answers: ${ results.incorrectCount } (0 marks)</p>
        <p>Not Attempted: ${ results.nonSubmittedCount } (0 marks)</p>
      `;
        document.getElementById( "resultSummary" ).innerHTML = summaryHTML;

        const detailedDiv = document.getElementById( "detailedResults" );
        results.detailedResults.forEach( ( result, index ) =>
        {
          const questionObj = questions[ index ];
          let optionsHTML = "";

          // For each option, assign a class based on whether it is correct,
          // whether it was the user's selected answer, or if not submitted.
          for ( const [ key, text ] of Object.entries( questionObj.options ) )
```

```
        {
          let optionClass = "option";
        // Determine highlight: if the option is the correct answer, always mark green.
        // If the user selected an answer and it is wrong, mark that option red.
          if ( result.userAnswer === key )
          {
            if ( key === questionObj.correct )
            {
              optionClass += " correct";
            } else
            {
              optionClass += " incorrect";
            }
          } else if ( key === questionObj.correct )
          {
            optionClass += " correct";
          }
          optionsHTML += `<div class="${ optionClass }">${ text }</div>`;
        }

        // Calculate question score - 10 for correct, 0 for incorrect or not attempted
        let questionScore = 0;
        if ( result.userAnswer === questionObj.correct )
        {
          questionScore = 10;
        }

        const questionHTML = `
        <div class="question">
          <h5>Q${ index + 1 }. ${ questionObj.q }</h5>
          ${ optionsHTML }
          <div class="status-text">${ result.status }</div>
          <div class="question-score">Marks: ${ questionScore }/10</div>
        </div>
      `;
          detailedDiv.innerHTML += questionHTML;
        } );
      }
    </script>
  </body>

</html>
```

style.css

```
body {
  background: #f8f9fa;
}

.container {
  max-width: 800px;
  margin-top: 30px;
}

h1 {
  text-align: center;
  margin-bottom: 20px;
}

.question {
  padding: 15px;
  margin-bottom: 20px;
  border: 1px solid #dee2e6;
  border-radius: 5px;
  background-color: #fff;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.05);
}

.question h5 {
  font-weight: 600;
}

.option {
  display: block;
  margin: 5px 0;
  padding: 8px;
  border: 1px solid #ccc;
  border-radius: 4px;
  cursor: pointer;
  transition: background-color 0.3s ease;
}

.option:hover {
  background-color: #e9ecef;
}
```

```
.btn-submit {
  display: block;
  margin: 30px auto;
  width: 50%;
  font-size: 1.1rem;
  padding: 10px;
}

.result-summary {
  margin-bottom: 30px;
  text-align: center;
  padding: 20px;
  background: #fff;
  border: 1px solid #dee2e6;
  border-radius: 5px;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.05);
}

.score {
  font-size: 24px;
  font-weight: bold;
  color: #0d6efd;
  margin: 15px 0;
}

.correct {
  background-color: #d4edda;
  border-color: #c3e6cb;
}

.incorrect {
  background-color: #f8d7da;
  border-color: #f5c6cb;
}

.not-attempted {
  background-color: #ffeeba;
  border-color: #ffeeba;
}

.status-text {
  font-weight: bold;
  margin-top: 10px;
```

```
}

.question-score {
  text-align: right;
  font-weight: bold;
  margin-top: 10px;
}

.btn-retake {
  display: block;
  margin: 30px auto;
  width: 40%;
  font-size: 1.1rem;
  padding: 10px;
}
```

script.js

```
// Quiz questions data structure
const questions = [
  {
    q: "Which tag is used to create a hyperlink in HTML?",
    options: { "1": "&lt;a&gt;", "2": "&lt;link&gt;", "3": "&lt;href&gt;", "4":
"&lt;hyperlink&gt;" },
    correct: "1"
  },
  {
    q: "Which CSS property is used to change the background color of an element?",
    options: { "1": "background-color", "2": "color", "3": "bgcolor", "4": "background" },
    correct: "1"
  },
  {
    q: "What does the 'id' attribute in HTML do?",
    options: { "1": "Defines the style of an element", "2": "Identifies a unique element",
"3": "Links the element to a CSS class", "4": "Creates a new HTML element" },
    correct: "2"
  },
  {
    q: "How do you add a comment in JavaScript?",
    options: { "1": "/* comment */", "2": "// comment", "3": "&lt;!-- comment --&gt;",
"4": "# comment" },
    correct: "2"
  },
```

```
  {
    q: "Which HTML element is used to define a paragraph?",
    options: { "1": "&lt;p&gt;", "2": "&lt;h1&gt;", "3": "&lt;div&gt;", "4":
"&lt;span&gt;" },
    correct: "1"
  }
];

// Build the quiz HTML dynamically
const quizContainer = document.getElementById( "quizContainer" );
questions.forEach( ( question, index ) =>
{
  const questionDiv = document.createElement( "div" );
  questionDiv.className = "question";
  questionDiv.innerHTML = `
  <h5>Q${ index + 1 }. ${ question.q }</h5>
  <div>
    ${ Object.entries( question.options ).map( ( [ key, value ] ) => `
            <label class="option">
              <input type="radio" name="q${ index + 1 }" value="${ key }"> ${ value }
            </label>
          `).join( '' ) }
  </div>
  `;
  quizContainer.appendChild( questionDiv );
} );

// Listen for quiz submit event
document.getElementById( "quizForm" ).addEventListener( "submit", function ( e )
{
  e.preventDefault();
  let correctCount = 0, incorrectCount = 0, nonSubmittedCount = 0;
  const detailedResults = [];

  // Evaluate each question
  questions.forEach( ( question, index ) =>
  {
    const qNumber = index + 1;
    const selected = document.querySelector( `input[name="q${ qNumber }"]:checked` );
    let status = "";
    if ( !selected )
    {
      nonSubmittedCount++;
```

```
      status = "No submitted answer";
    } else if ( selected.value === question.correct )
    {
      correctCount++;
      status = "Your answer is correct";
    } else
    {
      incorrectCount++;
      status = "Your answer is incorrect";
    }
    detailedResults.push( {
      q: question.q,
      options: question.options,
      correct: question.correct,
      userAnswer: selected ? selected.value : null,
      status: status
    } );
  } );

  const results = {
    correctCount,
    incorrectCount,
    nonSubmittedCount,
    detailedResults
  };

  // Store the results in localStorage to be picked up on results page
  localStorage.setItem( "quizResults", JSON.stringify( results ) );
  // Redirect to results page
  window.location.href = "result.html";
} );
```

Output ::

Web Dev Quiz

Q1. Which tag is used to create a hyperlink in HTML?

- ○ <a>
- ○ <link>
- ○ <href>
- ○ <hyperlink>

Q2. Which CSS property is used to change the background color of an element?

- ○ background-color
- ○ color
- ○ bgcolor
- ○ background

Q3. What does the 'id' attribute in HTML do?

- ○ Defines the style of an element
- ○ Identifies a unique element
- ○ Links the element to a CSS class
- ○ Creates a new HTML element

Q4. How do you add a comment in JavaScript?

- ○ /* comment */
- ○ // comment
- ○ <!-- comment -->
- ○ # comment

Q5. Which HTML element is used to define a paragraph?

- ○ <p>
- ○ <h1>
- ○
- ○

Submit Quiz

Quiz Results

Score: 10/50 (20.0%)

Total Questions: 5

Correct Answers: 1 (10 marks each)

Incorrect Answers: 3 (0 marks)

Not Attempted: 1 (0 marks)

Q1. Which tag is used to create a hyperlink in HTML?

- <a>
- <link>
- <href>
- <hyperlink>

Your answer is incorrect

Marks: 0/10

Q2. Which CSS property is used to change the background color of an element?

- background-color
- color
- bgcolor
- background

Your answer is incorrect

Marks: 0/10

Q3. What does the 'id' attribute in HTML do?

- Defines the style of an element
- Identifies a unique element
- Links the element to a CSS class
- Creates a new HTML element

Your answer is correct

Marks: 10/10

Q4. How do you add a comment in JavaScript?

- /* comment */
- // comment
- <!-- comment -->
- # comment

No submitted answer

Marks: 0/10

Q5. Which HTML element is used to define a paragraph?

- <p>
- <h1>
-
-

Your answer is incorrect

Marks: 0/10

Retake Quiz