***WEB TECHNOLOGIES LAB***

***PRACTICAL FILE***

Faculty Name: Mr. Ajay Tiwari Student’s Name: Paras Jain

Roll No.: 02014812721

Semester: 6

Group: 6 CST AIML

****

**Maharaja Agrasen Institute of Technology**

**Sector – 22, Rohini, New Delhi – 110085**

**INDEX**

**Name: Paras Jain**

**Enrolment Number: 02014812721**

**Branch: Computer Science and Technology**

**Group: 6 CST1 AIML**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SNo.** | **Experiment Name** | **Date** | **Marks** | | | | | **Signature** |
| **R1** | **R2** | **R3** | **R4** | **R5** |
| **1.** | Introduction to XAMPP |  |  |  |  |  |  |  |
| **2.** | Design web pages for your college containing a description of the courses, departments, faculties, library etc, use href, list tags. |  |  |  |  |  |  |  |
| **3.** | 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. |  |  |  |  |  |  |  |
| **4.** | Design a web page of your home town with an attractive background color, text color, an Image, font etc. (use internal CSS). |  |  |  |  |  |  |  |
| **5.** | Use External, Internal, and Inline CSS to format college web page that you created. |  |  |  |  |  |  |  |
| **6.** | Create HTML Page with JavaScript which takes Integer number as input and tells whether the number is ODD or EVEN |  |  |  |  |  |  |  |
| **7.** | Create HTML Page that contains form with fields Name, Email, Mobile No, Gender , Favourite Colour and a button now write a JavaScript code to combine and display the information in textbox when the button is clicked and implement validation. |  |  |  |  |  |  |  |
| **8.** | Create XML file to store student information like Enrolment Number, Name Mobile Number , Email Id. |  |  |  |  |  |  |  |
| **9.** | Write s PHP script to read data from txt file and display it in html table. |  |  |  |  |  |  |  |
| **10.** | 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. |  |  |  |  |  |  |  |
| **11.** | 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) |  |  |  |  |  |  |  |

# EXPERIMENT - 1

# AIM: Introduction to XAMPP

**XAMPP** is a free and open-source cross-platform web server solution stack that allows you to easily set up a development environment on your local machine. This means you can create and test websites and web applications without needing an internet connection or a remote server.

**What's in the XAMPP Package?**

XAMPP bundles several essential components for web development:

* **Apache HTTP Server:** The core of XAMPP, Apache is a powerful and widely used web server software that processes incoming requests and delivers web content.
* **MySQL Database Management System**: MySQL is a popular open-source relational database management system that allows you to store and manage data efficiently for your web applications.
* **PHP Hypertext Preprocessor**: PHP is a server-side scripting language commonly used for creating dynamic and interactive web pages. It can access databases, generate HTML content, and handle user interactions.
* **Perl**: An additional scripting language included in XAMPP, Perl can be used for web development tasks like data processing and automation.

**Benefits of Using XAMPP:**

* **Easy Setup and Use**: XAMPP provides a one-click installation process, making it ideal for beginners to get started with web development.
* **Local Development Environment**: Develop and test your projects offline, without relying on external servers or internet connectivity.

**Getting Started with XAMPP:**

1. **Download and Install XAMPP**: Download the free installer from the official Apache Friends website [(https://www.apachefriends.org/](https://www.apachefriends.org/)). Choose the installer compatible with your operating system and follow the on-screen instructions.
2. **Start the XAMPP Services**: Once installed, launch the XAMPP Control Panel. Activate and start Apache and MySQL services using the corresponding buttons.
3. **Access Your Localhost**: Open a web browser and navigate to http://localhost/. This should display the default XAMPP welcome page, confirming your successful setup.

**EXPERIMENT - 2**

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

**THEORY:**

**HTML**: The building block of web pages, HTML defines the structure and content of your webpage using tags and attributes.

**Hyperlinks (href)**: These create clickable links within the webpage, allowing users to navigate to different sections or external websites.

**Lists (list tags)**: Used to structure information in an organized and visually appealing manner.

Different types of lists like ordered (numbered) or unordered (bulleted) can be employed

**SOURCE CODE:**

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

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

    <title>College Website</title>

  </head>

  <body>

    <header>

      <h1>Welcome to Our College</h1>

      <nav>

        <ul>

          <li><a href="#courses">Courses</a></li>

          <li><a href="#departments">Departments</a></li>

          <li><a href="#faculties">Faculties</a></li>

          <li><a href="#library">Library</a></li>

        </ul>

      </nav>

    </header>

    <main>

      <section id="courses">

        <h2>Courses Offered</h2>

        <p>Here are some of the courses offered at our college:</p>

        <ul>

          <li>Computer Science</li>

          <li>Engineering</li>

          <li>Business Administration</li>

          <!-- Add more courses as needed -->

        </ul>

      </section>

      <section id="departments">

        <h2>Departments</h2>

        <p>Our college has the following departments:</p>

        <ul>

          <li>Department of Computer Science</li>

          <li>Department of Engineering</li>

          <li>Department of Business</li>

          <!-- Add more departments as needed -->

        </ul>

      </section>

      <section id="faculties">

        <h2>Faculties</h2>

        <p>Meet some of our esteemed faculty members:</p>

        <ul>

          <li>Dr. John Doe - Computer Science</li>

          <li>Prof. Jane Smith - Engineering</li>

          <li>Dr. Michael Johnson - Business</li>

          <!-- Add more faculty members as needed -->

        </ul>

      </section>

      <section id="library">

        <h2>Library</h2>

        <p>

          Our college library is equipped with a vast collection of books and

          resources to support learning:

        </p>

        <ul>

          <li>Textbooks</li>

          <li>Reference materials</li>

          <li>Online databases</li>

          <!-- Add more library resources as needed -->

        </ul>

      </section>

    </main>

    <footer>

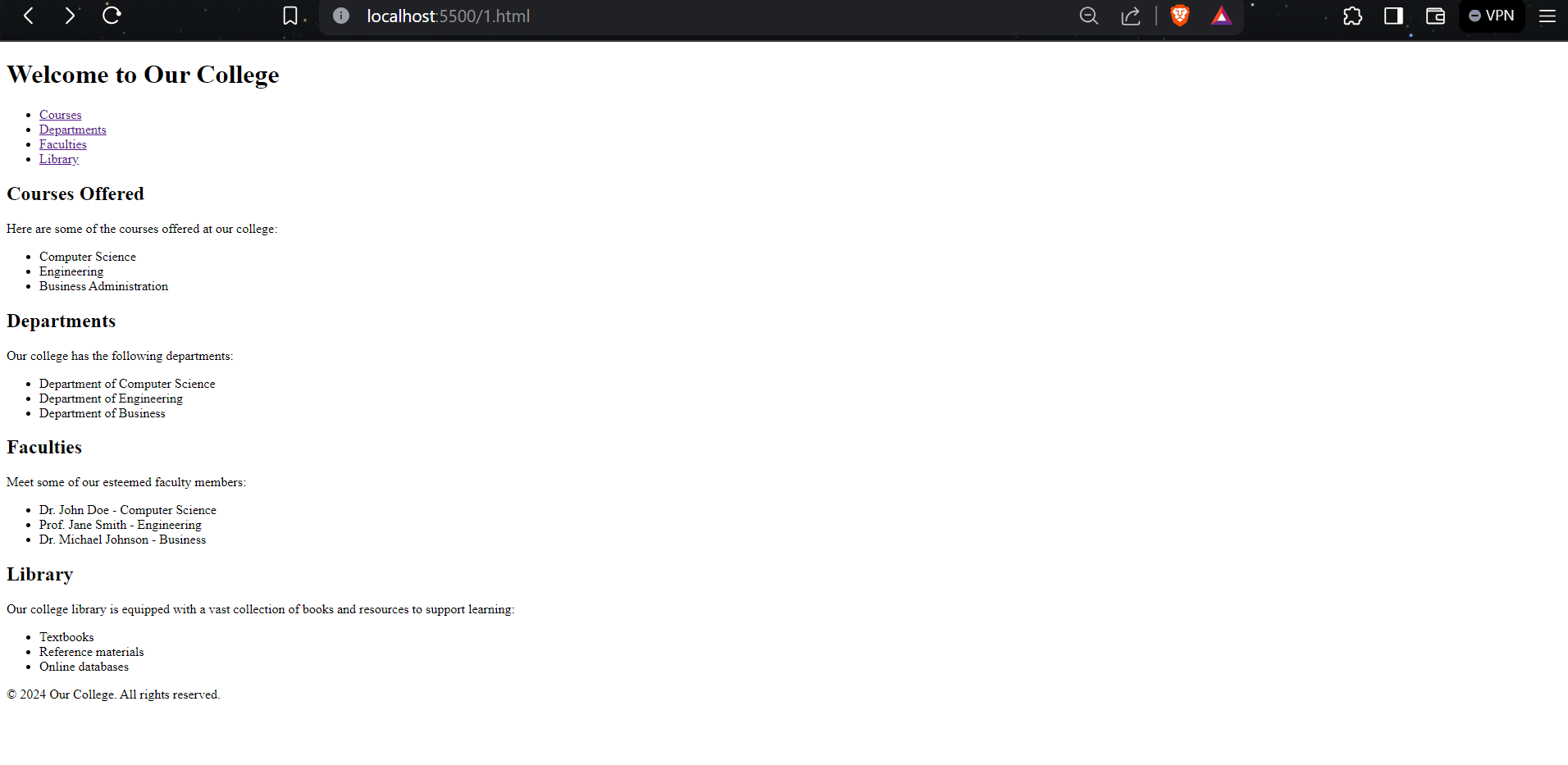
      <p>&copy; 2024 Our College. All rights reserved.</p>

    </footer>

  </body>

</html>

**OUTPUT:**



**EXPERIMENT - 3**

**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:**

**Frames (Deprecated)**: HTML frames allow dividing a web page into multiple sections, each displaying a separate HTML document. This can be useful for creating layouts with fixed regions.

**Modern Alternative:** While frames offered a way to structure webpages in the past, they are considered deprecated in modern web development. CSS (Cascading Style Sheets) provides a more flexible and recommended approach for creating layouts.

**SOURCE CODE:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Two Frames Webpage</title>

    <style>

        /\* Styling for the frames and columns \*/

        .frame {

            width: 100%;

            height: 50vh; /\* 50% of viewport height \*/

            display: flex;

        }

        .column {

            flex: 1; /\* Equal width for each column \*/

            padding: 20px;

            box-sizing: border-box;

        }

        /\* Background colors for columns \*/

        .column:nth-child(odd) {

            background-color: #f0f0f0; /\* Light gray \*/

        }

        .column:nth-child(even) {

            background-color: #e0e0e0; /\* Lighter gray \*/

        }

    </style>

</head>

<body>

    <div class="frame" id="frame1">

        <div class="column">

            <h2>Frame 1, Column 1</h2>

            <p>This is the content of the first column in the first frame.</p>

        </div>

        <div class="column">

            <h2>Frame 1, Column 2</h2>

            <p>This is the content of the second column in the first frame.</p>

        </div>

    </div>

    <div class="frame" id="frame2">

        <div class="column">

            <h2>Frame 2, Column 1</h2>

            <p>This is the content of the first column in the second frame.</p>

        </div>

        <div class="column">

            <h2>Frame 2, Column 2</h2>

            <p>This is the content of the second column in the second frame.</p>

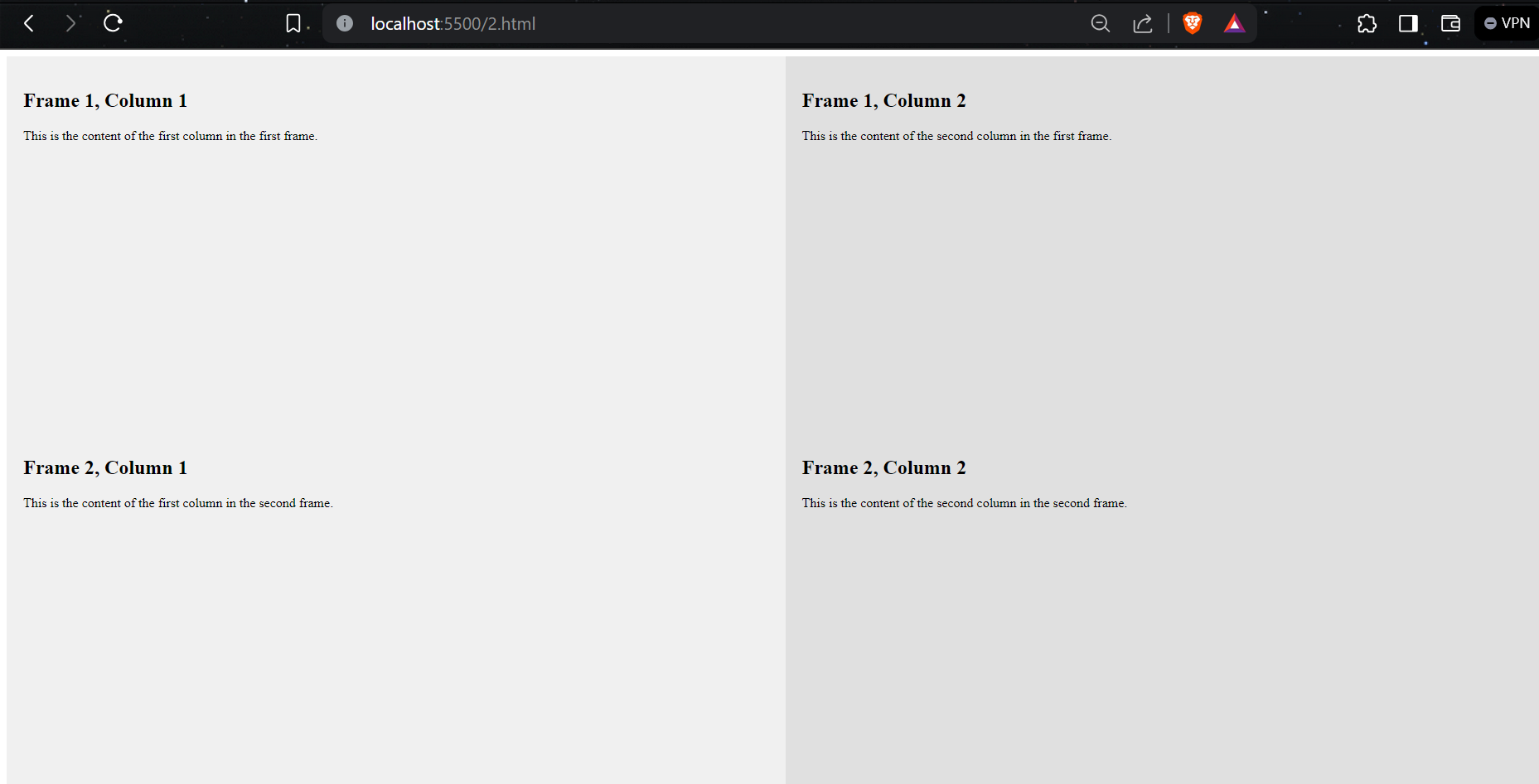
        </div>

    </div>

</body>

</html>

**OUTPUT:**



**EXPERIMENT - 4**

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

## **THEORY:**

Inline CSS provides a way to define styles for HTML elements directly within the opening tag itself. This approach offers fine-grained control over the appearance of specific elements but can lead to less maintainable code for larger projects.

**SOURCE CODE:**

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

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

    <title>My Hometown</title>

    <style>

      body {

        background-color: #f4f4f4; /\* Light gray background \*/

        color: #333; /\* Dark gray text color \*/

        font-family: Arial, sans-serif; /\* Font family \*/

        margin: 0;

        padding: 0;

      }

      header {

        background-color: #4682b4; /\* Steel blue header background \*/

        color: #fff; /\* White header text color \*/

        text-align: center;

        padding: 20px 0;

      }

      h1 {

        margin-top: 0;

        font-size: 2.5em;

      }

      main {

        max-width: 800px;

        margin: 0 auto;

        padding: 20px;

      }

      p {

        font-size: 1.1em;

        line-height: 1.6;

      }

      img {

        display: block;

        max-width: 100%;

        height: auto;

        margin: 20px auto;

        border-radius: 10px; /\* Rounded corners for the image \*/

        box-shadow: 0 0 10px rgba(0, 0, 0, 0.1); /\* Box shadow for the image \*/

      }

    </style>

  </head>

  <body>

    <header>

      <h1>Welcome to My Hometown</h1>

    </header>

    <main>

      <p>

        Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed id massa

        metus. Nullam auctor suscipit lacinia. Sed lacinia non felis id aliquet.

        Aliquam tincidunt malesuada nunc, in interdum est lobortis et. Ut

        commodo mi id volutpat tristique.

      </p>

      <img src="hometown.png" alt="Hometown Image" />

      <p>

        Integer vel sagittis magna. Quisque nec convallis nisi. Fusce lobortis

        justo id magna feugiat, a malesuada odio consequat. Duis sem metus,

        commodo id risus non, tincidunt molestie ligula. Sed et tortor in odio

        fermentum commodo.

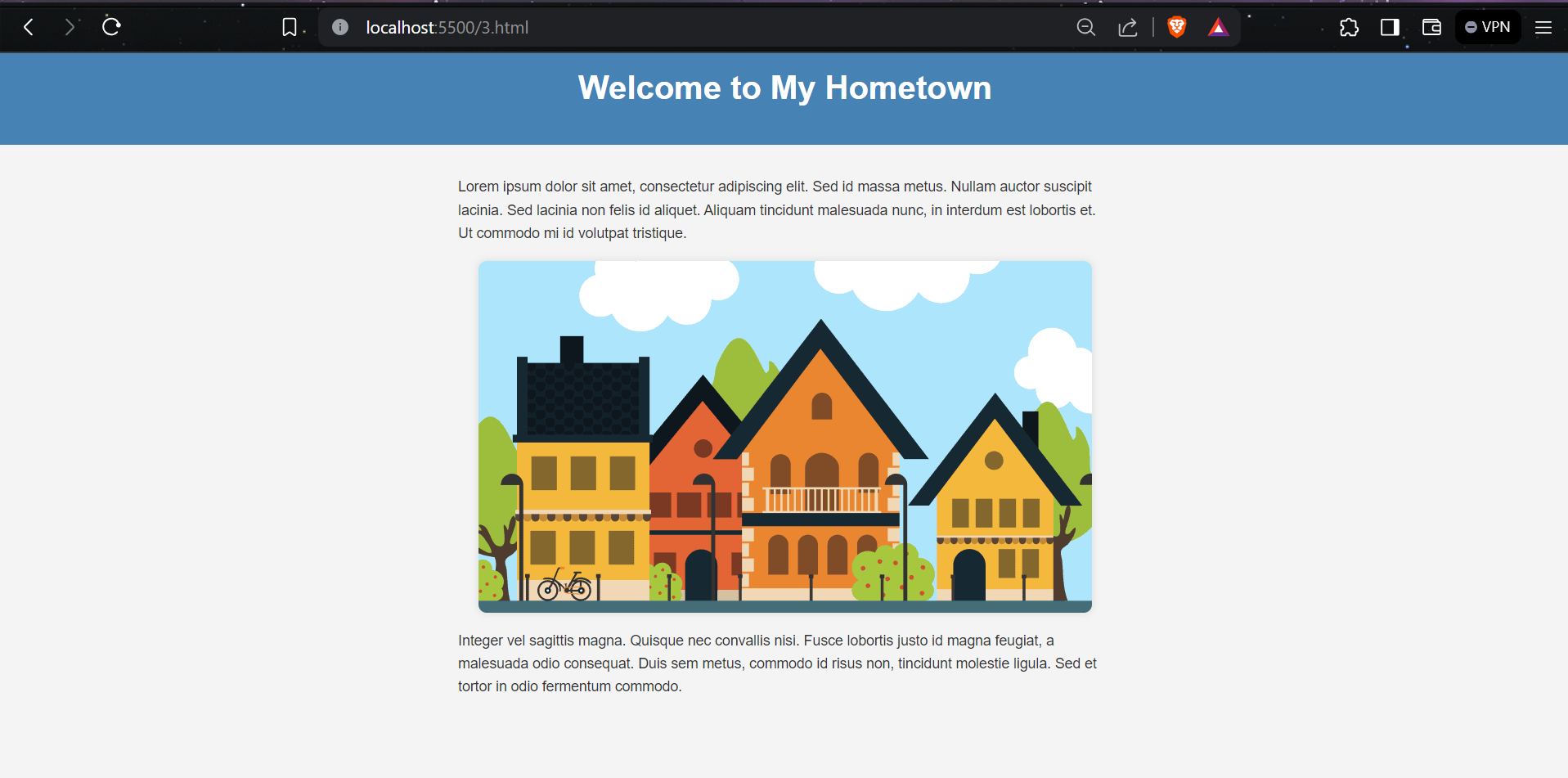
      </p>

    </main>

  </body>

</html>

**OUTPUT:**



**EXPERIMENT - 5**

**AIM: Use External, Internal, and Inline CSS to format college web page that you created.**

**THEORY:**

**External CSS (Recommended**): Styles are defined in a separate .css file linked to the HTML document using the <link> tag in the <head> section. This promotes reusability, maintainability, and separation of concerns.

**Internal CSS**: Styles are defined within a <style> tag placed in the <head> section of the HTML document. This approach offers more control than inline styles but is less reusable than external stylesheets.

**SOURCE CODE:**

**HTML FILE**

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

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

    <title>College Web Page</title>

    <link rel="stylesheet" href="styles4.css" />

    <style>

      /\* Additional internal styles \*/

      footer {

        background-color: #333;

        color: #fff;

        text-align: center;

        padding: 10px 0;

      }

    </style>

  </head>

  <body>

    <header

      style="

        background-color: #4682b4;

        color: #fff;

        text-align: center;

        padding: 20px 0;

      "

    >

      <h1 style="margin-top: 0; font-size: 2.5em">Welcome to Our College</h1>

    </header>

    <main style="max-width: 800px; margin: 0 auto; padding: 20px">

      <p style="font-size: 1.1em; line-height: 1.6">

        This is the main content of our college webpage.

      </p>

      <img

        src="college.png"

        alt="College Image"

        style="

          display: block;

          max-width: 100%;

          height: auto;

          margin: 20px auto;

          border-radius: 10px;

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

        "

      />

    </main>

    <footer

      style="

        background-color: #333;

        color: #fff;

        text-align: center;

        padding: 10px 0;

      "

    >

      <p>&copy; 2024 Our College. All rights reserved.</p>

    </footer>

  </body>

</html>

**EXTERNAL CSS FILE**

/\* styles4.css \*/

body {

  font-family: Arial, sans-serif;

  background-color: #f4f4f4;

  color: #333;

  margin: 0;

  padding: 0;

}

header {

  background-color: #4682b4;

  color: #fff;

  text-align: center;

  padding: 20px 0;

}

h1 {

  margin-top: 0;

  font-size: 2.5em;

}

main {

  max-width: 800px;

  margin: 0 auto;

  padding: 20px;

}

p {

  font-size: 1.1em;

  line-height: 1.6;

}

img {

  display: block;

  max-width: 100%;

  height: auto;

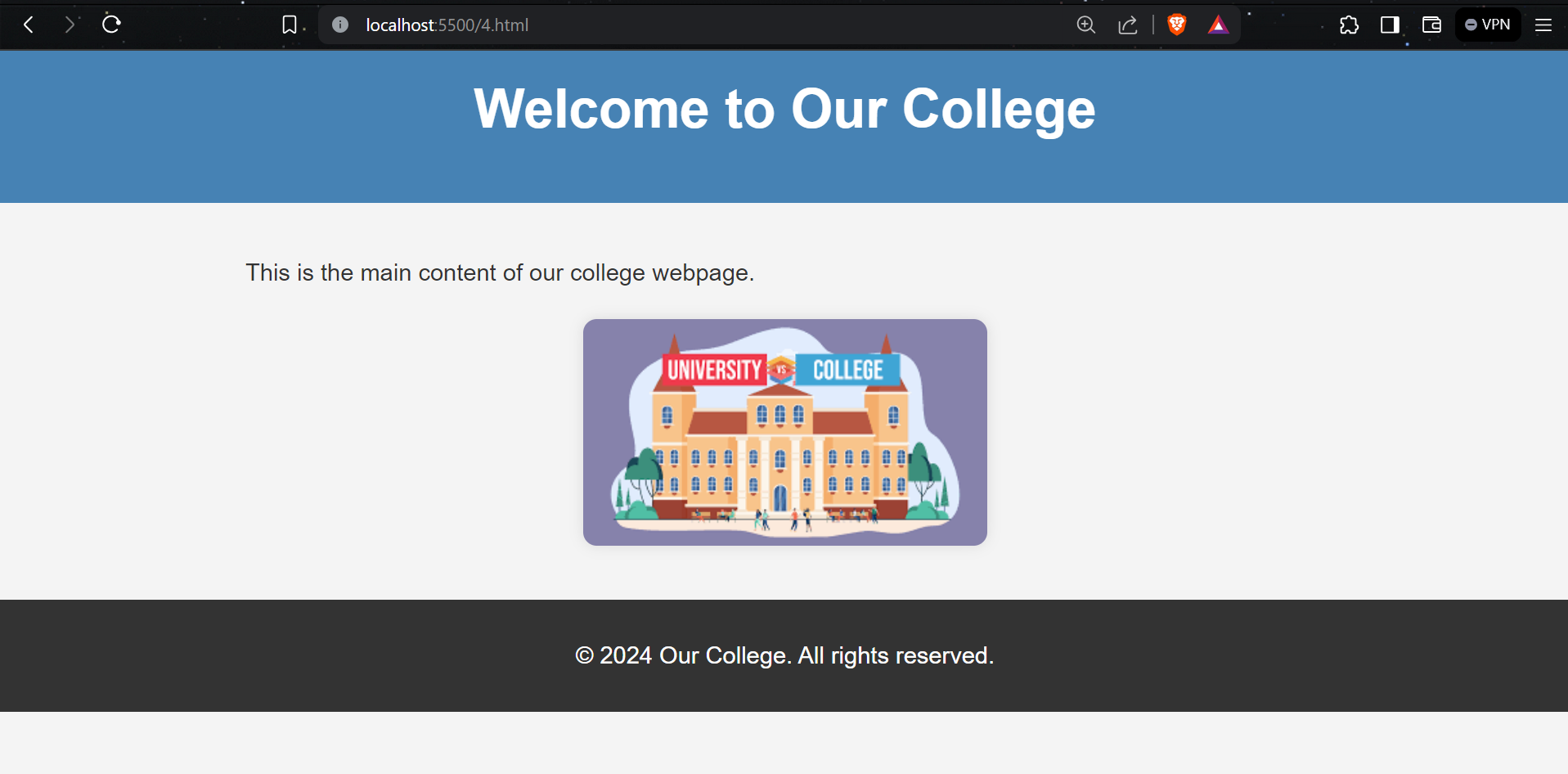
  margin: 20px auto;

  border-radius: 10px;

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

}

**OUTPUT:**



**EXPERIMENT - 6**

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

**THEORY:**

JavaScript (JS) is a versatile scripting language that brings web pages to life. It runs directly in web browsers, allowing manipulation of HTML content and user interactions. This makes web pages dynamic and responsive to user actions. JS is interpreted, meaning code is executed line by line without pre-compilation. With its object-oriented features and event-driven nature, JS empowers the creation of interactive web experiences.

**SOURCE CODE:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Odd or Even Checker</title>

</head>

<body>

    <h1>Odd or Even Checker</h1>

    <label for="numberInput">Enter a number:</label>

    <input type="number" id="numberInput">

    <button onclick="checkNumber()">Check</button>

    <p id="result"></p>

    <script>

        function checkNumber() {

            // Get the input value

            var number = document.getElementById("numberInput").value;

            // Check if the input is a valid integer

            if (number.trim() === '' || isNaN(number)) {

                alert("Please enter a valid integer number.");

                return;

            }

            // Convert the input to an integer

            number = parseInt(number);

            // Check if the number is odd or even

            if (number % 2 === 0) {

                document.getElementById("result").innerText = number + " is EVEN.";

            } else {

                document.getElementById("result").innerText = number + " is ODD.";

            }

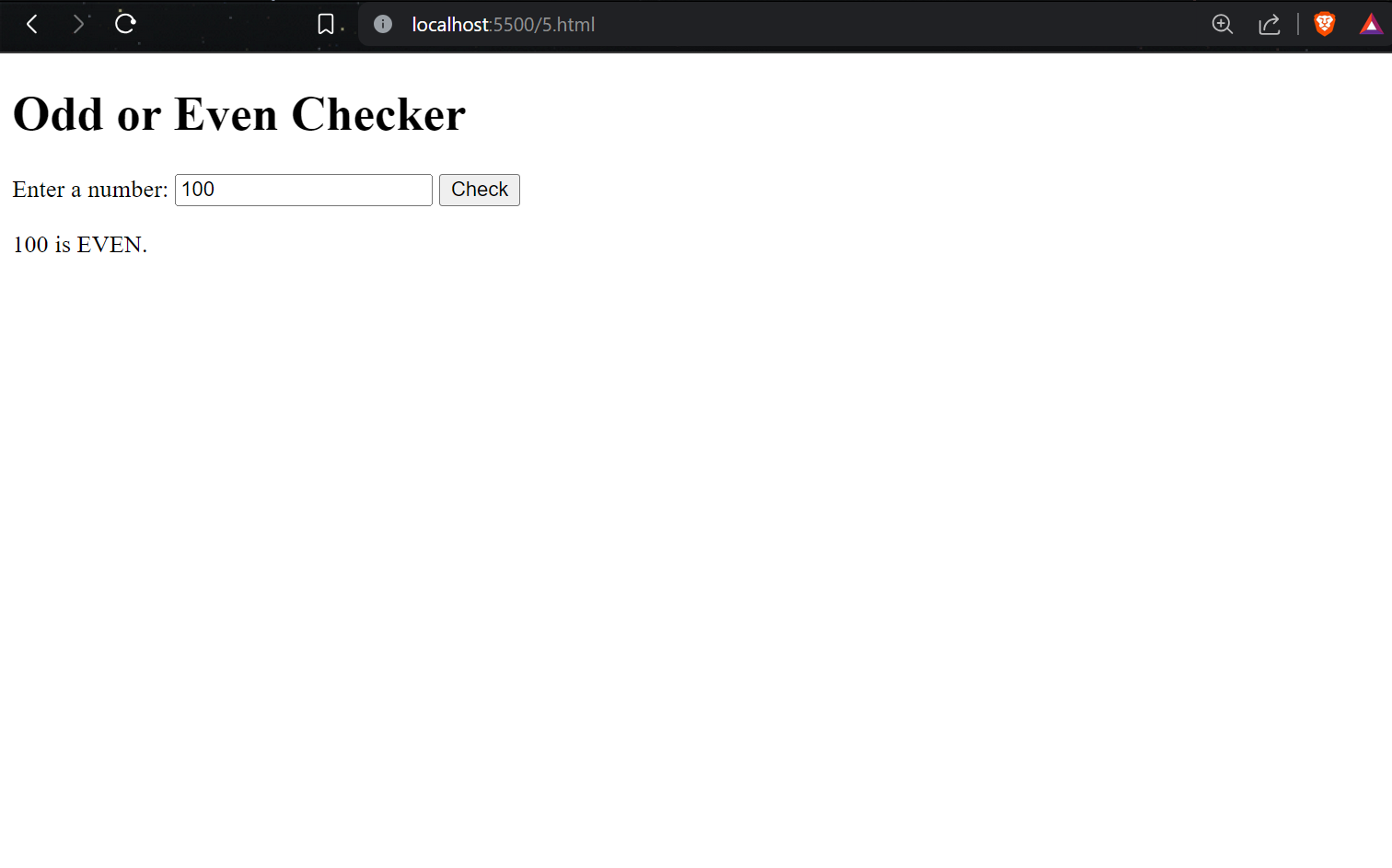
        }

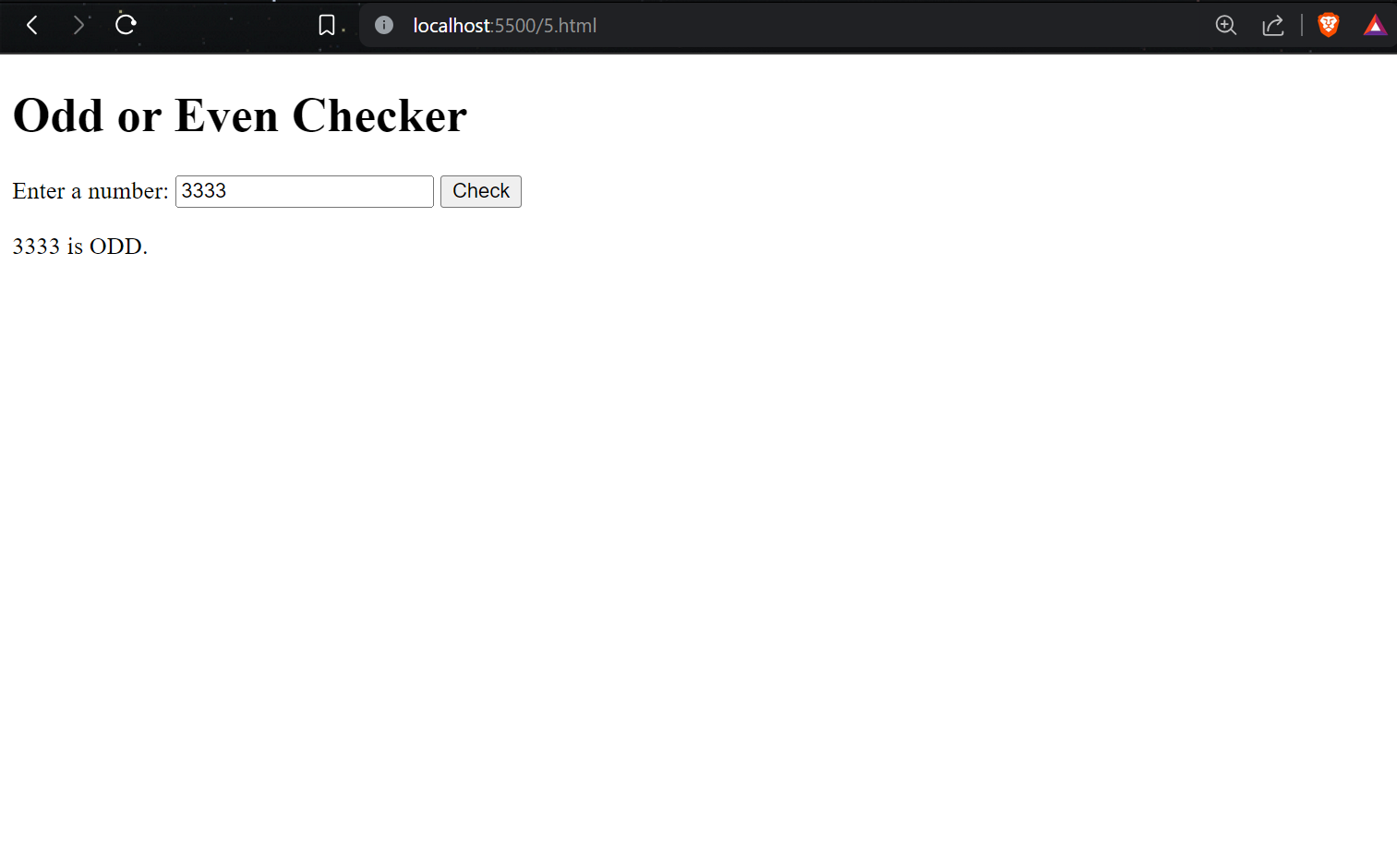
    </script>

</body>

</html>

**OUTPUT:**





**EXPERIMENT - 7**

**AIM: Create HTML Page that contains form with fields Name, Email, Mobile No, Gender , Favourite Colour and a button now write a JavaScript code to combine and display the information in textbox when the button is clicked and implement validation.**

**THEORY:**

JavaScript forms elevate data collection on web pages by making it interactive and dynamic. It leverages the Document Object Model (DOM) to access and manipulate form elements like input fields and buttons. By responding to user interactions such as form submissions or button clicks, JavaScript can validate user input in real-time, ensuring data accuracy and providing immediate feedback.

**SOURCE CODE:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>User Information Form</title>

</head>

<body>

    <h1>User Information Form</h1>

    <form id="userInfoForm">

        <label for="name">Name:</label><br>

        <input type="text" id="name" name="name" required><br><br>

        <label for="email">Email:</label><br>

        <input type="email" id="email" name="email" required><br><br>

        <label for="mobile">Mobile No:</label><br>

        <input type="tel" id="mobile" name="mobile" pattern="[0-9]{10}" required><br><br>

        <label for="gender">Gender:</label><br>

        <select id="gender" name="gender">

            <option value="male">Male</option>

            <option value="female">Female</option>

            <option value="other">Other</option>

        </select><br><br>

        <label for="color">Favorite Colour:</label><br>

        <input type="color" id="color" name="color"><br><br>

        <button type="button" onclick="displayUserInfo()">Submit</button>

    </form><br>

    <textarea id="userInfoDisplay" rows="5" cols="50" readonly></textarea>

    <script>

        function displayUserInfo() {

            // Get form data

            var name = document.getElementById("name").value.trim();

            var email = document.getElementById("email").value.trim();

            var mobile = document.getElementById("mobile").value.trim();

            var gender = document.getElementById("gender").value;

            var color = document.getElementById("color").value;

            // Validation

            if (name === '' || email === '' || mobile === '') {

                alert("Please fill in all required fields.");

                return;

            }

            // Combine and display information

            var userInfo = "Name: " + name + "\n" +

                           "Email: " + email + "\n" +

                           "Mobile No: " + mobile + "\n" +

                           "Gender: " + gender + "\n" +

                           "Favorite Colour: " + color;

            document.getElementById("userInfoDisplay").value = userInfo;

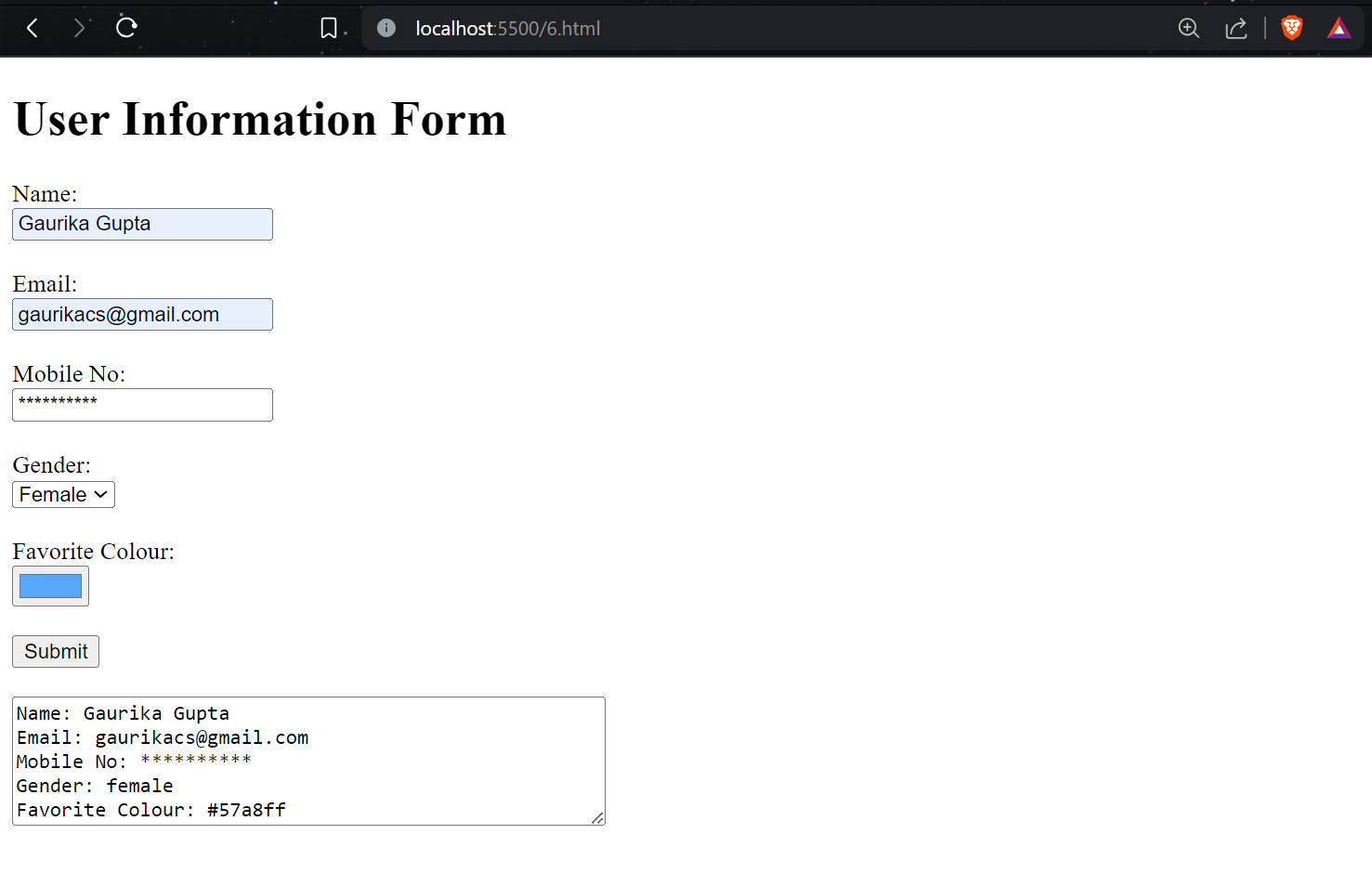
        }

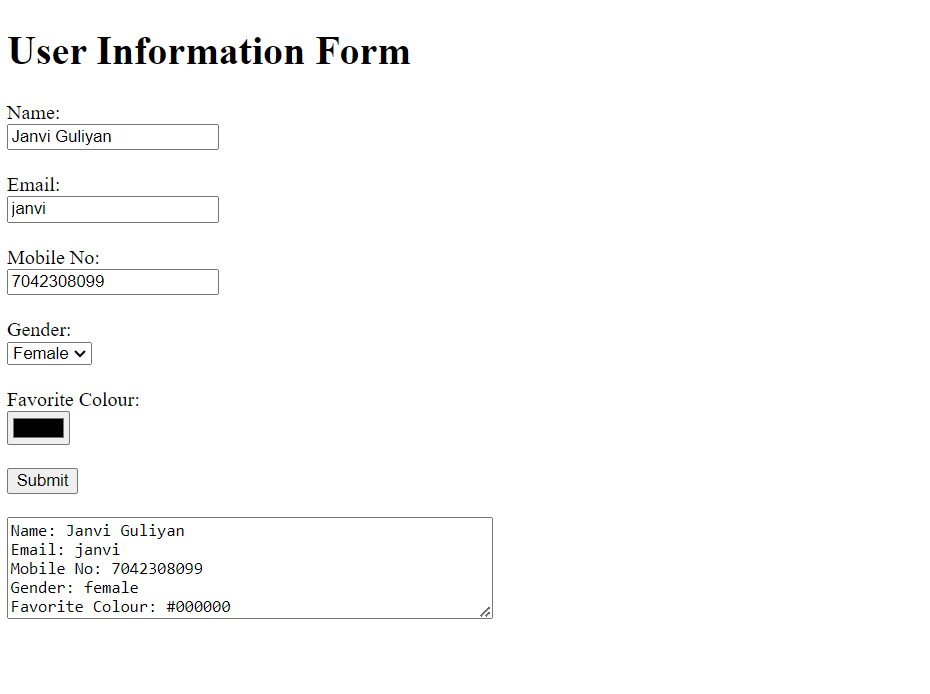
    </script>

</body>

</html>

**OUTPUT:**





**EXPERIMENT - 8**

**AIM: Create XML file to store student information like Enrolment Number, Name Mobile Number , Email Id.**

**THEORY:**

XML (Extensible Markup Language) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. It is designed to store and transport data, allowing efficient data exchange and integration between different systems and applications. XML documents have a hierarchical structure, consisting of elements, attributes, and text content, organized in a tree-like structure.

To create an XML file to store student information like enrollment number, name, mobile number, and email ID, you would typically have a root element (e.g., `<students>`) that contains multiple child elements (e.g., `<student>`). Each `<student>` element would then have child elements to represent the different pieces of information, such as `<enrollment\_number>`, `<name>`, `<mobile\_number>`, and `<email>`. The element names should follow XML naming conventions, and the data for each student would be enclosed within the respective elements. This hierarchical structure allows for easy organization and retrieval of student data in a standardized format.

**SOURCE CODE:**

<?xml version="1.0" encoding="UTF-8"?>

<students>

    <student>

        <enrollment\_number>1001</enrollment\_number>

        <name>John Doe</name>

        <mobile\_number>1234567890</mobile\_number>

        <email>john@example.com</email>

    </student>

    <student>

        <enrollment\_number>1002</enrollment\_number>

        <name>Jane Smith</name>

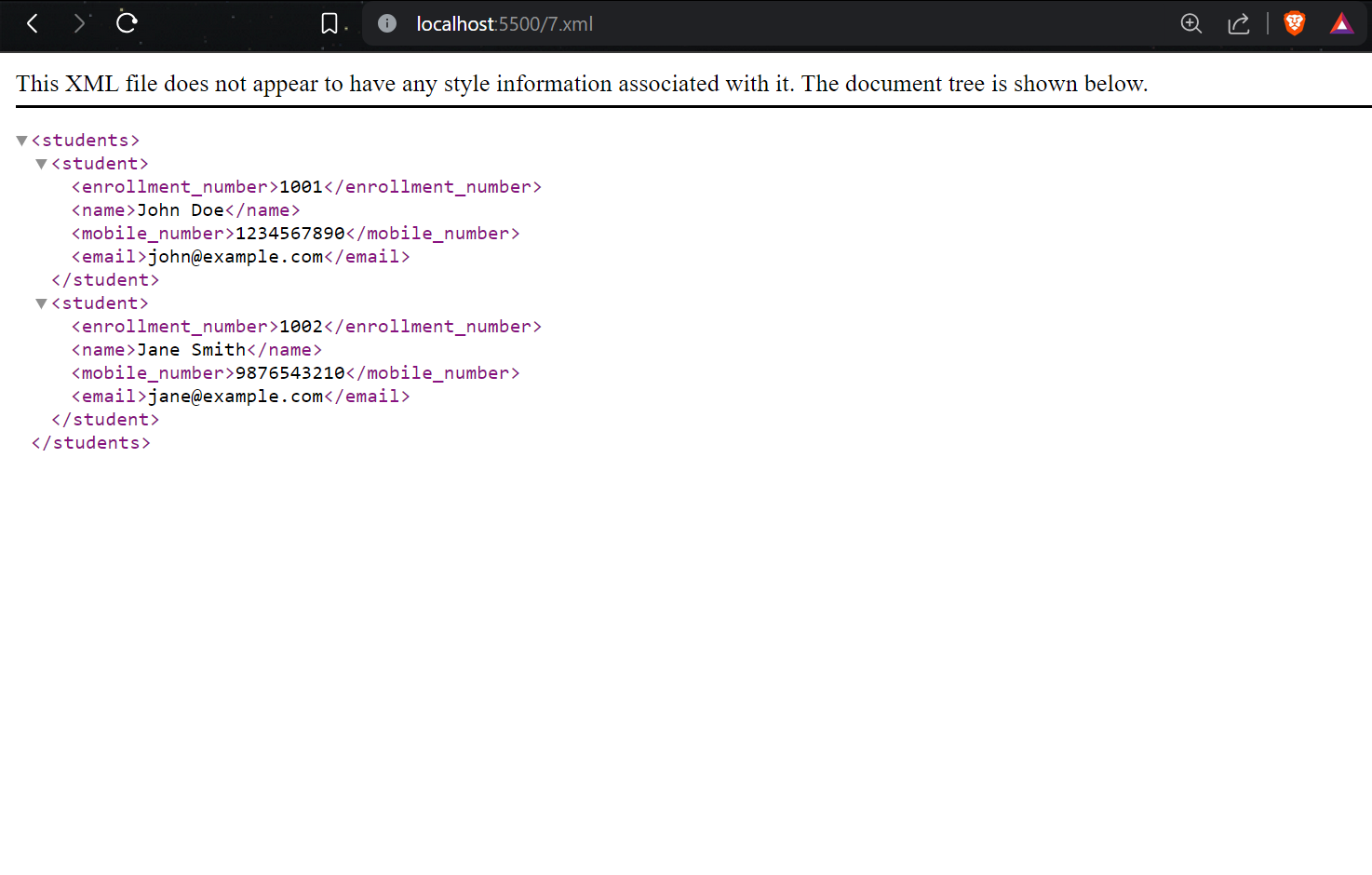
        <mobile\_number>9876543210</mobile\_number>

        <email>jane@example.com</email>

    </student>

</students>

**OUTPUT:**



**EXPERIMENT - 9**

**AIM: Write s PHP script to read data from txt file and display it in html table.**

**THEORY:**

PHP is a server-side scripting language primarily used for web development. It can be embedded into HTML code and is commonly used to interact with databases to retrieve, store, and manipulate data. In this script, we will use PHP to read data from a text file and display it in an HTML table.

File Handling: PHP provides functions to read, write, and manipulate files on the server. We will use the fopen(), fgets(), and fclose() functions to open, read, and close the text file, respectively.

String Manipulation: We will use string manipulation functions like explode() to split the data read from the file into separate values (Name, Password, Email).

**SOURCE CODE:**

**Data.txt**

John Doe: password123: john@example.com

Jane Smith: qwerty: jane@example.com

**Index.php**

<?php

// Specify the file path

$file = 'data.txt';

// Check if the file exists

if (file\_exists($file)) {

// Read the contents of the file

$contents = file\_get\_contents($file);

// Split the contents into lines

$lines = explode("\n", $contents);

// Start the HTML table with borders

echo '<table style="border: 1px solid black; border-collapse: collapse;">';

echo '<tr><th style="border: 1px solid black; padding: 5px;">Name</th><th style="border: 1px solid black; padding: 5px;">Password</th><th style="border: 1px solid black; padding: 5px;">Email</th></tr>';

// Loop through each line

foreach ($lines as $line) {

// Skip empty lines

if (trim($line) === '') {

continue;

}

// Split the line into Name, Password, and Email

$parts = explode(':', $line);

$name = isset($parts[0]) ? trim($parts[0]) : '';

$password = isset($parts[1]) ? trim($parts[1]) : '';

$email = isset($parts[2]) ? trim($parts[2]) : '';

// Display the data in a table row with borders

echo '<tr>';

echo '<td style="border: 1px solid black; padding: 5px;">' . htmlspecialchars($name) . '</td>';

echo '<td style="border: 1px solid black; padding: 5px;">' . htmlspecialchars($password) . '</td>';

echo '<td style="border: 1px solid black; padding: 5px;">' . htmlspecialchars($email) . '</td>';

echo '</tr>';

}

// End the HTML table

echo '</table>';

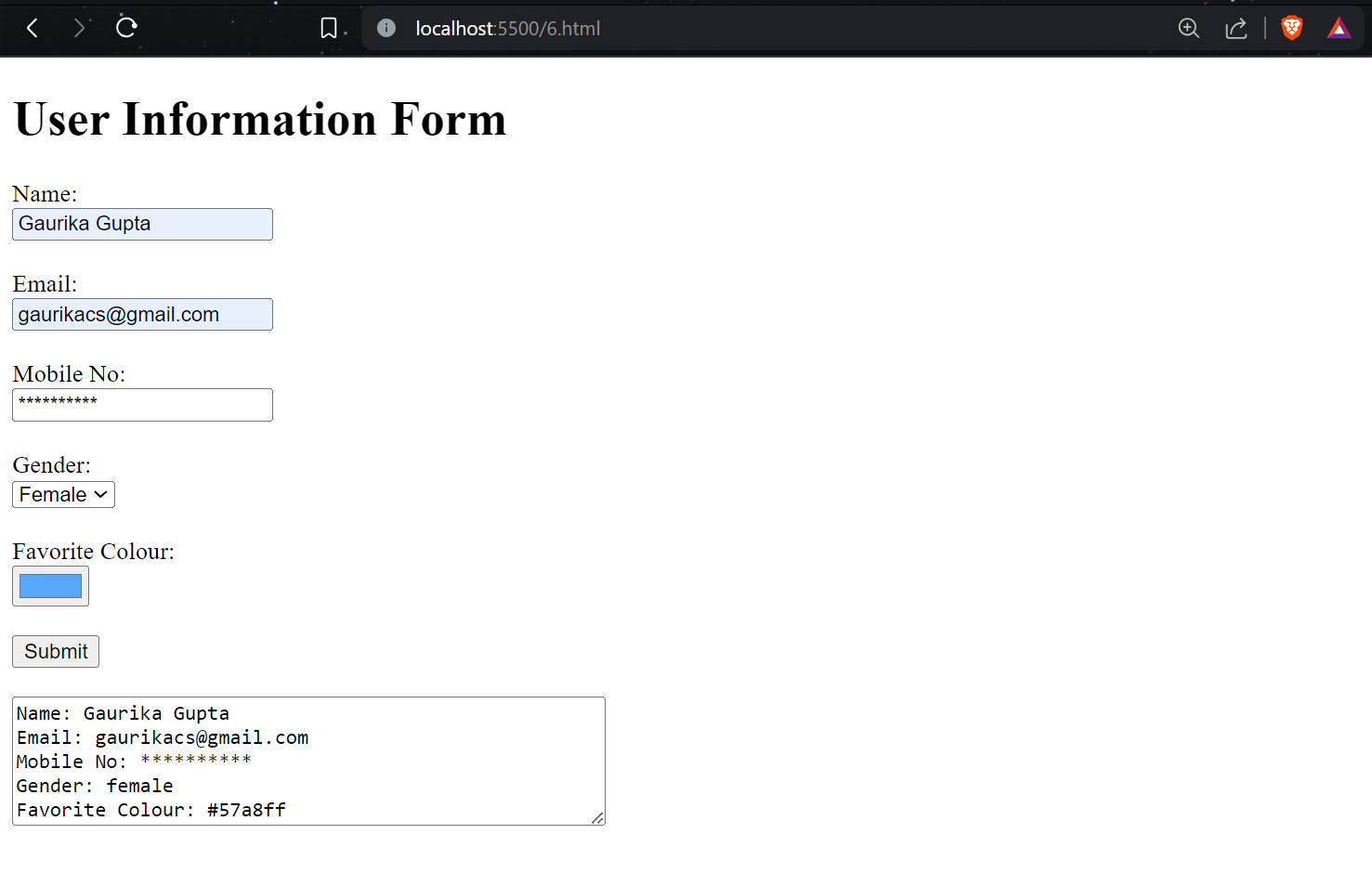
} else {

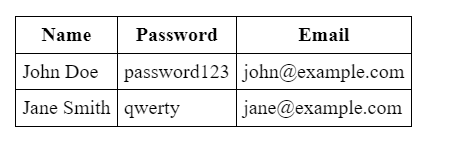
echo 'File not found.';

}

?>

**OUTPUT:**



****

**EXPERIMENT - 10**

**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:**

For login authentication in PHP, we will create a form in HTML to collect the username and password from the user. We will then use PHP to validate the entered credentials against a stored list of usernames and passwords in a text file.

File Handling: We will use PHP file handling functions to read the stored usernames and passwords from a text file.

Form Handling: PHP will be used to process the form data submitted by the user.

Security: We will hash the passwords in the text file using PHP's password\_hash() function for security.

**SOURCE CODE:**

**Users.txt**

john:password123

jane:qwerty

**index.php**

<?php

// Function to check if the username and password match the data in the file

function authenticate($username, $password) {

$file = 'users.txt';

$handle = fopen($file, 'r');

if ($handle) {

while (($line = fgets($handle)) !== false) {

$parts = explode(':', trim($line));

if (count($parts) == 2) {

$storedUsername = $parts[0];

$storedPassword = $parts[1];

if ($username === $storedUsername && $password === $storedPassword) {

fclose($handle);

return true; // Authentication successful

}

}

}

fclose($handle);

}

return false; // Authentication failed

}

// Check if the form is submitted

if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {

$username = $\_POST['username'];

$password = $\_POST['password'];

if (authenticate($username, $password)) {

echo 'Login successful!';

} else {

echo 'Invalid username or password.';

}

}

?>

<!DOCTYPE html>

<html>

<head>

<title>Login Form</title>

</head>

<body>

<h1>Login</h1>

<form method="post" action="<?php echo htmlspecialchars($\_SERVER['PHP\_SELF']); ?>">

<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>

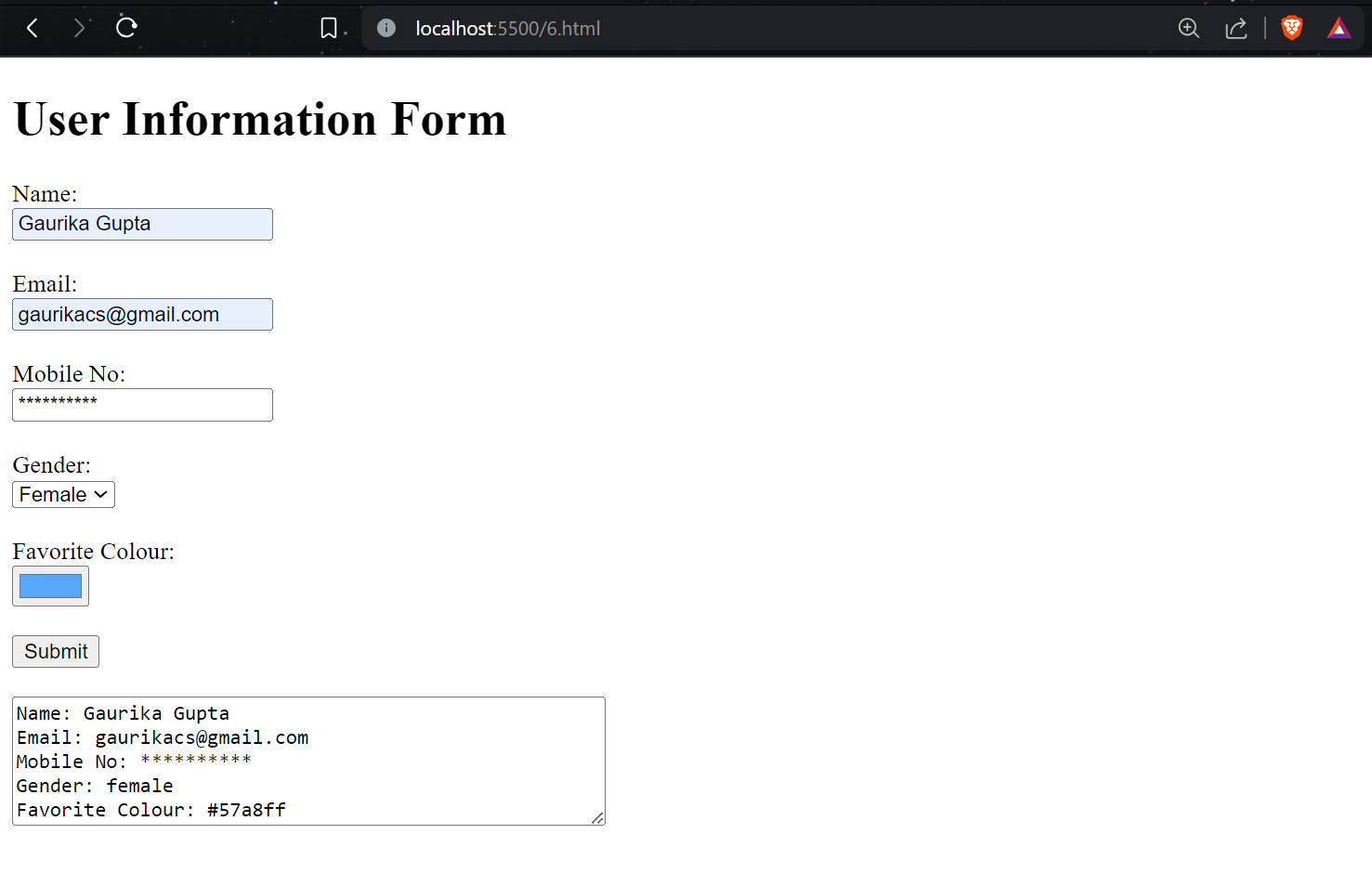
<input type="submit" value="Login">

</form>

</body>

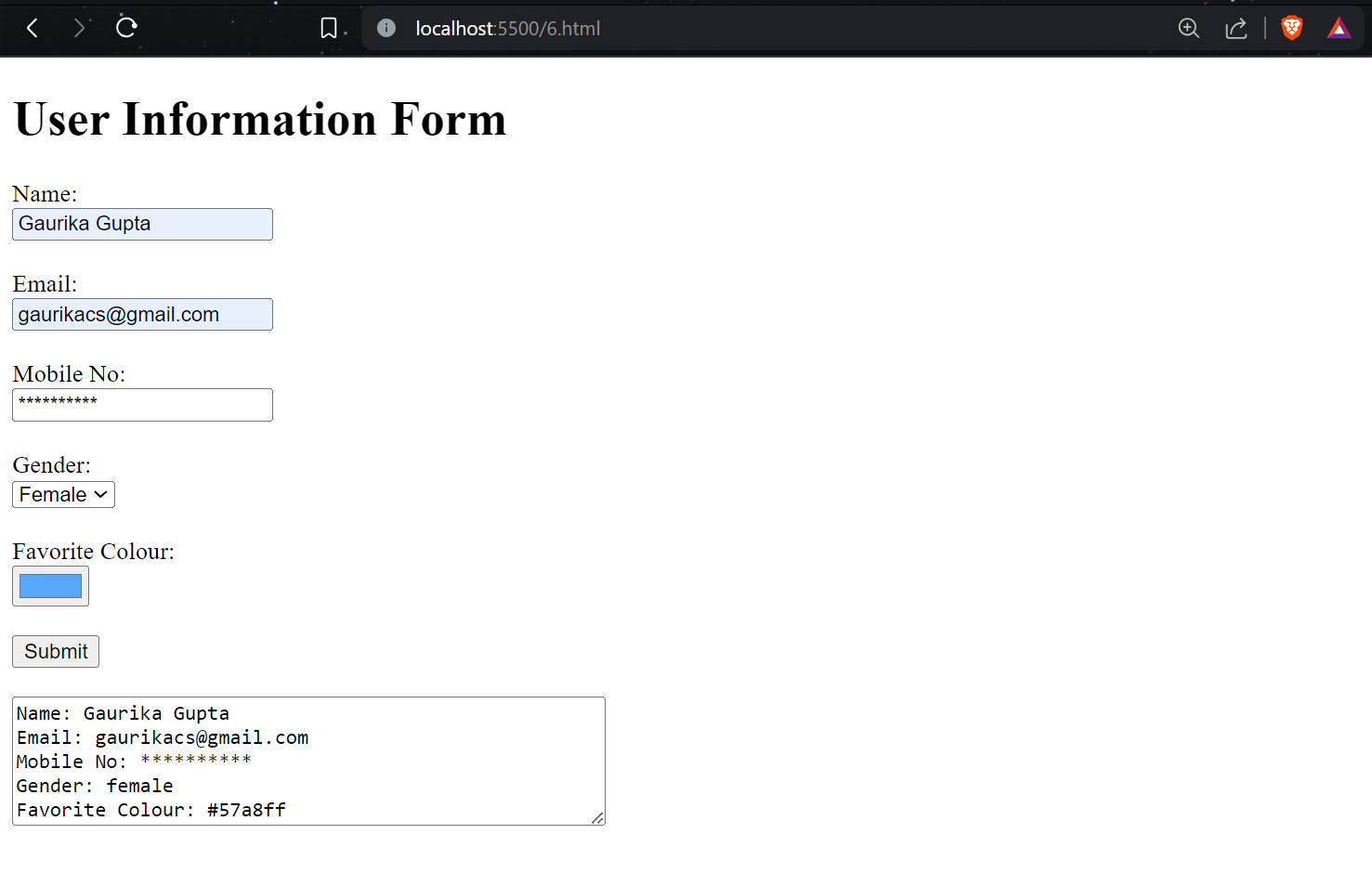
</html>

**OUTPUT:**



****

After Login:



****

**EXPERIMENT -11**

**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:**

In this example, we will create a simple registration form in HTML (register.php) to collect user information such as Name, Address, Email, and Mobile Number. We will then use PHP to store this data in a MySQL database. We will also create a PHP script (display.php) to retrieve and display all user information in an HTML table.

HTML Form: Collects user information.

PHP Form Handling: Processes form data and stores it in a database.

MySQL Database: Stores user information.

PHP MySQLi Extension: Used for database connectivity and operations.

MySQL Database Setup: Before proceeding, make sure you have a MySQL database created with a table named users. The table should have columns for id (auto-increment), name, address, email, and mobile.

**SOURCE CODE:**

**Register.php**

<?php

// Database connection details

$servername = "localhost";

$username = "your\_username";

$password = "your\_password";

$database = "your\_database";

// Create connection

$conn = new mysqli($servername, $username, $password, $database);

// Check connection

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

// Check if the form is submitted

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

// Retrieve form data

$name = $\_POST["name"];

$address = $\_POST["address"];

$email = $\_POST["email"];

$mobile = $\_POST["mobile"];

// Prepare and execute SQL query

$sql = "INSERT INTO users (name, address, email, mobile) VALUES ('$name', '$address', '$email', '$mobile')";

if ($conn->query($sql) === TRUE) {

echo "User information stored successfully.";

} else {

echo "Error: " . $sql . "<br>" . $conn->error;

}

}

// Close connection

$conn->close();

?>

<!DOCTYPE html>

<html>

<head>

<title>User Registration</title>

</head>

<body>

<h1>User Registration</h1>

<form method="post" action="<?php echo htmlspecialchars($\_SERVER["PHP\_SELF"]);?>">

<label for="name">Name:</label>

<input type="text" id="name" name="name" required><br><br>

<label for="address">Address:</label>

<input type="text" id="address" name="address" required><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="Submit">

</form>

</body>

</html>

**Display.php**

<?php

// Database connection details

$servername = "localhost";

$username = "your\_username";

$password = "your\_password";

$database = "your\_database";

// Create connection

$conn = new mysqli($servername, $username, $password, $database);

// Check connection

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

// Retrieve data from the users table

$sql = "SELECT \* FROM users";

$result = $conn->query($sql);

// Close connection

$conn->close();

?>

<!DOCTYPE html>

<html>

<head>

<title>User Information</title>

<style>

table {

border-collapse: collapse;

width: 100%;

}

th, td {

padding: 8px;

text-align: left;

border-bottom: 1px solid #ddd;

}

th {

background-color: #f2f2f2;

}

</style>

</head>

<body>

<h1>User Information</h1>

<table>

<tr>

<th>Name</th>

<th>Address</th>

<th>Email</th>

<th>Mobile No.</th>

</tr>

<?php

if ($result->num\_rows > 0) {

while ($row = $result->fetch\_assoc()) {

echo "<tr>";

echo "<td>" . $row["name"] . "</td>";

echo "<td>" . $row["address"] . "</td>";

echo "<td>" . $row["email"] . "</td>";

echo "<td>" . $row["mobile"] . "</td>";

echo "</tr>";

}

} else {

echo "<tr><td colspan='4'>No user information found.</td></tr>";

}

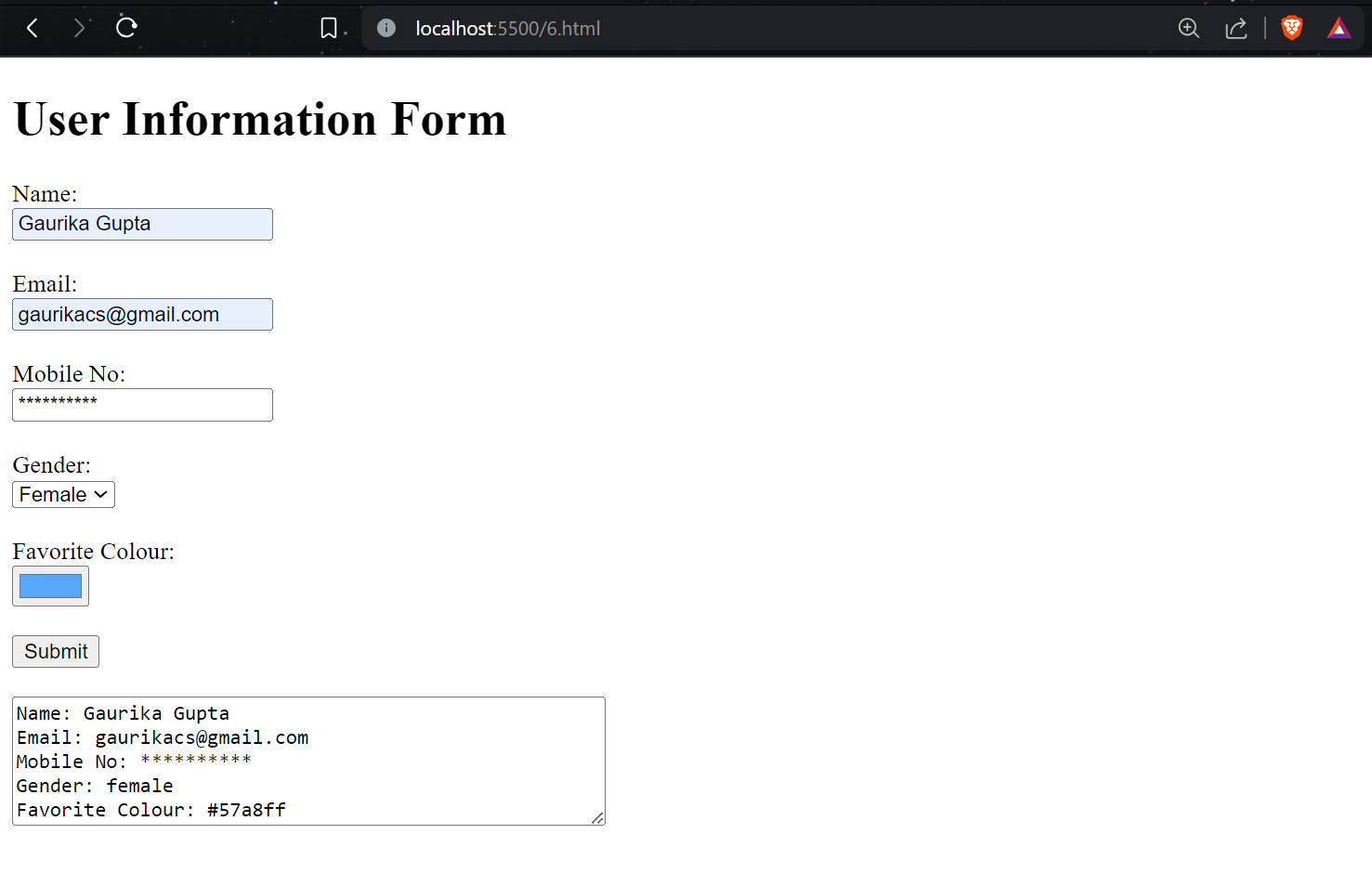
?>

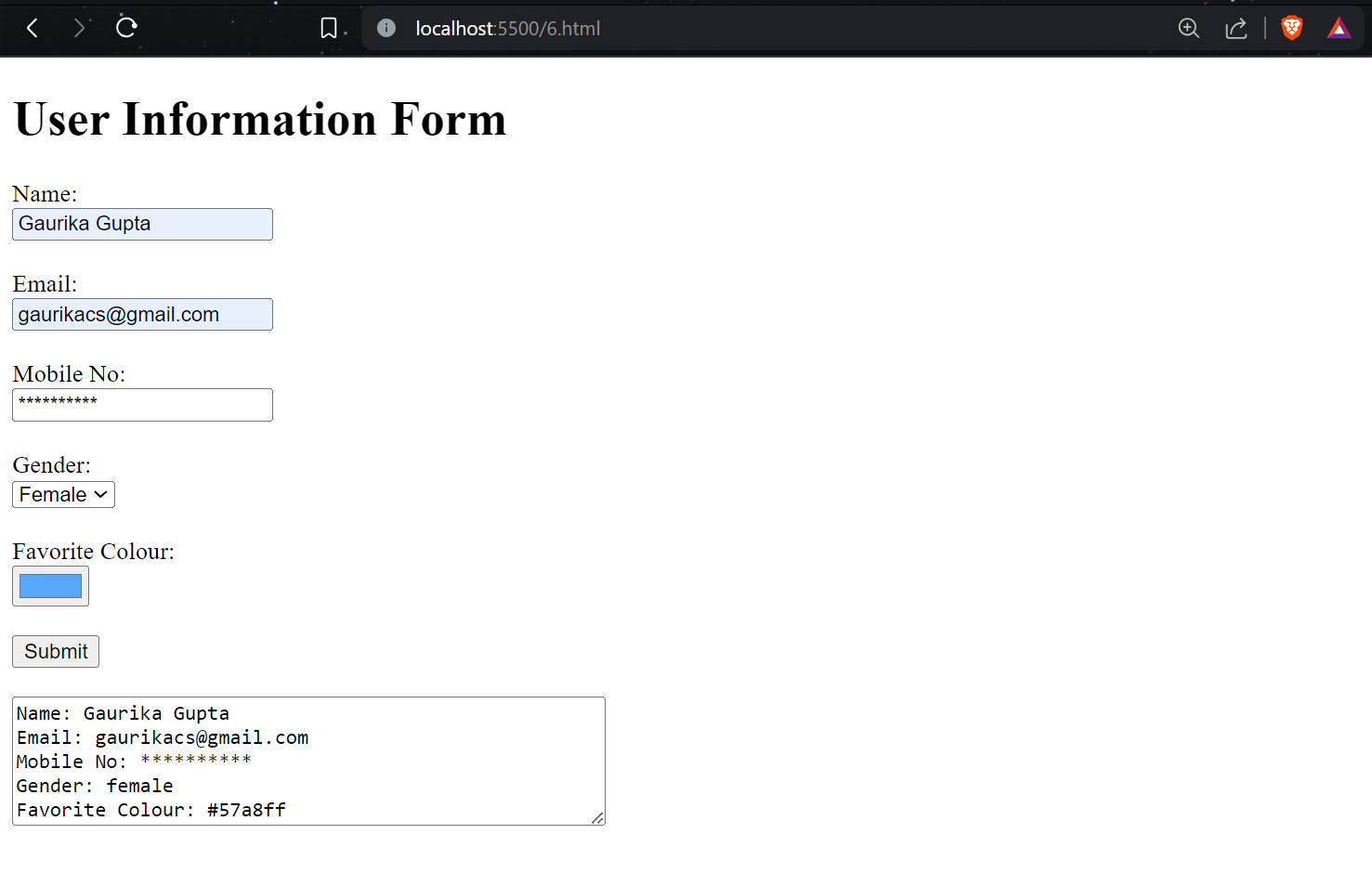
</table>

</body>

</html>

**OUTPUT:**

****



****