

**Prof. Ramkrishna More**

**Arts, Commerce and Science College(Autonomous)**

**Akurdi Pradhikaran, Pune-44**

**M. SC - COMPUTER APPLICATIONS**



Semester-I

Lab Course: Web Technology Laboratory

(CAMAEP-519-A)

WorkBook

**Name of Student: Jagruti Krushna Shirsath**

**Roll No:34**

**Academic Year: 2024-25**

**TABLE OF CONTENTS**

|  |  |
| --- | --- |
| 1. | Create a simple HTML webpage with headings, paragraphs, and links. |
| 2. | Build a website that displays images and uses image maps for client-server interaction. |
| 3. | Develop a webpage with a form that captures user input and displays the submitted data. |
| 4. | Design a responsive website using CSS media queries to adapt to different screen sizes. |
| 5. | Implement a CSS dropdown menu for website navigation. |
| 6. | Create a webpage with a table that dynamically populates data using JavaScript. |
| 7. | Build a JavaScript calculator that performs basic arithmetic operations. |
| 8. | Develop a JavaScript slideshow that displays images with automatic transitions. |
| 9. | Implement form validation using JavaScript to ensure proper user input. |
| 10. | Design a webpage layout using CSS grid or flexbox for responsive design. |
| 11. | Build a website that incorporates CSS animations for visual effects. |
| 12. | Develop a webpage with a contact form that sends user input to a server using AJAX. |
| 13. | Create a JavaScript quiz that presents multiple-choice questions and tracks the user's score. |
| 14. | Implement client-side form validation using HTML5 form input types and attributes. |
| 15. | Build a responsive image gallery using HTML, CSS, and JavaScript. |
| 16. | Design a webpage layout using Bootstrap framework. |
| 17. | Create a JavaScript program that generates a random password based on user-specified criteria. |
| 18. | Develop a website that utilizes cookies for user session management. |
| 19. | Implement CSS transitions to create smooth hover effects on webpage elements. |
| 20. | Build a JavaScript program that performs basic string manipulation tasks. |
| 21. | a. Create an XML document that represents a simple data structure and validate it against a DTD. |
| 22. | Design a webpage layout using CSS float and positioning techniques. |
| 23. | Build a website that retrieves data from a web API using JavaScript and displays it dynamically. |
| 24. | Develop a webpage with a responsive navigation bar that collapses into a menu on smaller screens. |
| 25. | Implement CSS transforms to create visually appealing 3D effects on webpage elements. |
| 26. | Create a JavaScript program that interacts with the browser's local storage to store and retrieve data. |
| 27. | Build a webpage that incorporates parallax scrolling using CSS and JavaScript. |
| 28. | Design a webpage layout using CSS framework such as Foundation or Bulma. |
| 29. | Develop a JavaScript program that manipulates the Document Object Model (DOM) to dynamically update webpage content. |
| 30. | Implement XML transformation using XSLT to convert an XML document into a different format. |

**Certificate**

This is to certify that Mr./Ms. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

has successfully completed the programming assignments in the Web

Technology Laboratory.

(Prof.Ankush Dhamal) (Dr. Santosh Jagtap)

**Lab In-charge**  **H.O.D**

**Internal Examiner External Examiner**

**Program 1:Create a simple HTML webpage with headings, paragraphs, and links.**

**Source code:**

<!DOCTYPE html>

<html>

<head>

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

<title>Groceries Website</title>

</head>

<body>

<header>

<h1>Welcome to Groceries Emporium</h1>

<p>Your Source for the Best Quality Groceries</p>

</header>

</header>

<nav>

<ul>

<li><a href="index.html">Home</a></li>

<li><a href="groceries.html">Groceries</a></li>

</ul>

</nav>

<main>

<p>At Groceries Emporium, we take pride in offering a wide selection of premium groceries that are not only delicious

but also packed with essential nutrients. Our hand-picked collection includes a variety of products, from fresh

vegetables to organic snacks and beverages.</p>

</main>

<footer>

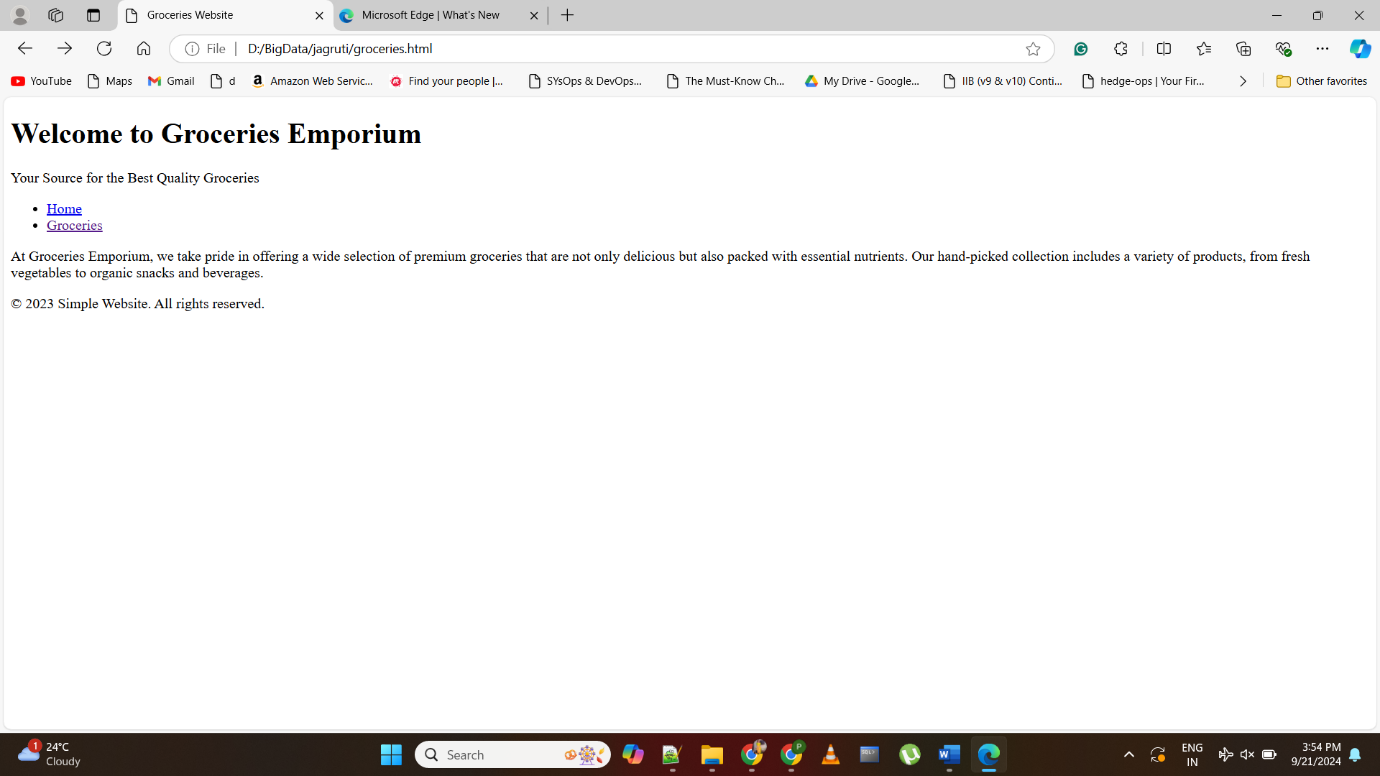
<p>&copy; 2023 Simple Website. All rights reserved.</p>

</footer>

</body>

</html>

**Output:-**



**Program 2**

**Build a website that displays images and uses image maps for client-server interaction**.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Image Map Example</title>

<style>

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 0;

display: flex;

flex-direction: column;

align-items: center;

justify-content: center;

height: 100vh;

background-color: #f0f0f0;

}

h1 {

margin-bottom: 20px;

}

img {

max-width: 100%;

height: auto;

}

</style>

</head>

<body>

<h1>Interactive Image Map</h1>

<img src="map.jpg" alt="Interactive Map" usemap="#imagemap">

<map name="imagemap">

<!-- Define clickable areas on the image -->

<area shape="circle" coords="100,100,50" alt="Area 1" href="#" onclick="handleClick(1)">

<area shape="rect" coords="200,200,300,300" alt="Area 2" href="#" onclick="handleClick(2)">

</map>

<script>

function handleClick(area) {

switch (area) {

case 1:

alert("Clicked Area 1");

break;

case 2:

alert("Clicked Area 2");

break;

// Add more cases for additional areas if needed

}

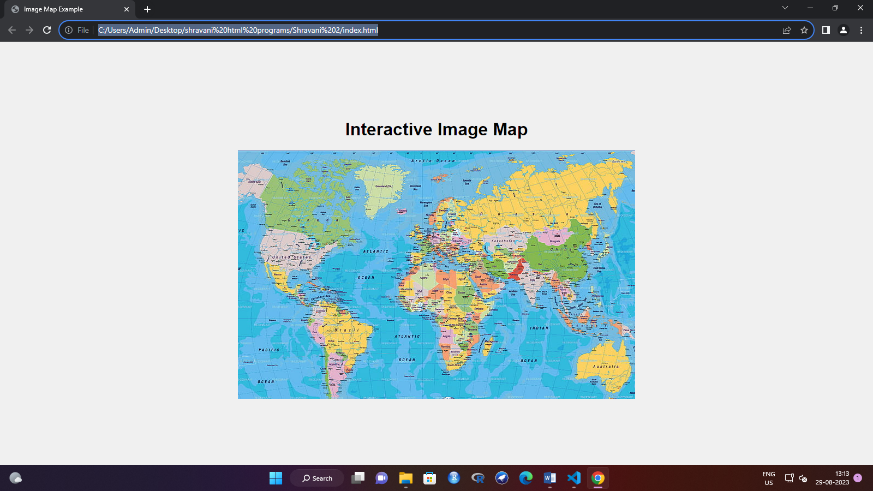
}

</script>

</body>

</html>

**Output:-**



**Program 3**

**Develop a webpage with a form that captures user input and displays the submitted data.**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>User Input Form</title>

<style>

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 0;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

background-color: #f0f0f0;

}

.container {

background-color: #fff;

border-radius: 10px;

box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);

padding: 20px;

width: 400px;

}

form label {

display: block;

margin-bottom: 5px;

}

form input[type="text"],

form input[type="email"],

form textarea {

width: 100%;

padding: 10px;

margin-bottom: 15px;

border: 1px solid #ccc;

border-radius: 4px;

}

form button {

background-color: #007bff;

color: #fff;

padding: 10px 20px;

border: none;

border-radius: 4px;

cursor: pointer;

}

#displayData {

margin-top: 20px;

font-weight: bold;

}

</style>

</head>

<body>

<div class="container">

<h1>User Input Form</h1>

<form id="userForm">

<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" rows="4" required></textarea>

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

</form>

<div id="displayData">

<!-- Submitted data will be displayed here -->

</div>

</div>

<script>

const userForm = document.getElementById("userForm");

const displayData = document.getElementById("displayData");

userForm.addEventListener("submit", function(event) {

event.preventDefault();

const name = document.getElementById("name").value;

const email = document.getElementById("email").value;

const message = document.getElementById("message").value;

const submittedData = `

<p>Name: ${name}</p>

<p>Email: ${email}</p>

<p>Message: ${message}</p>

`; displayData.innerHTML = submittedData;

userForm.reset();

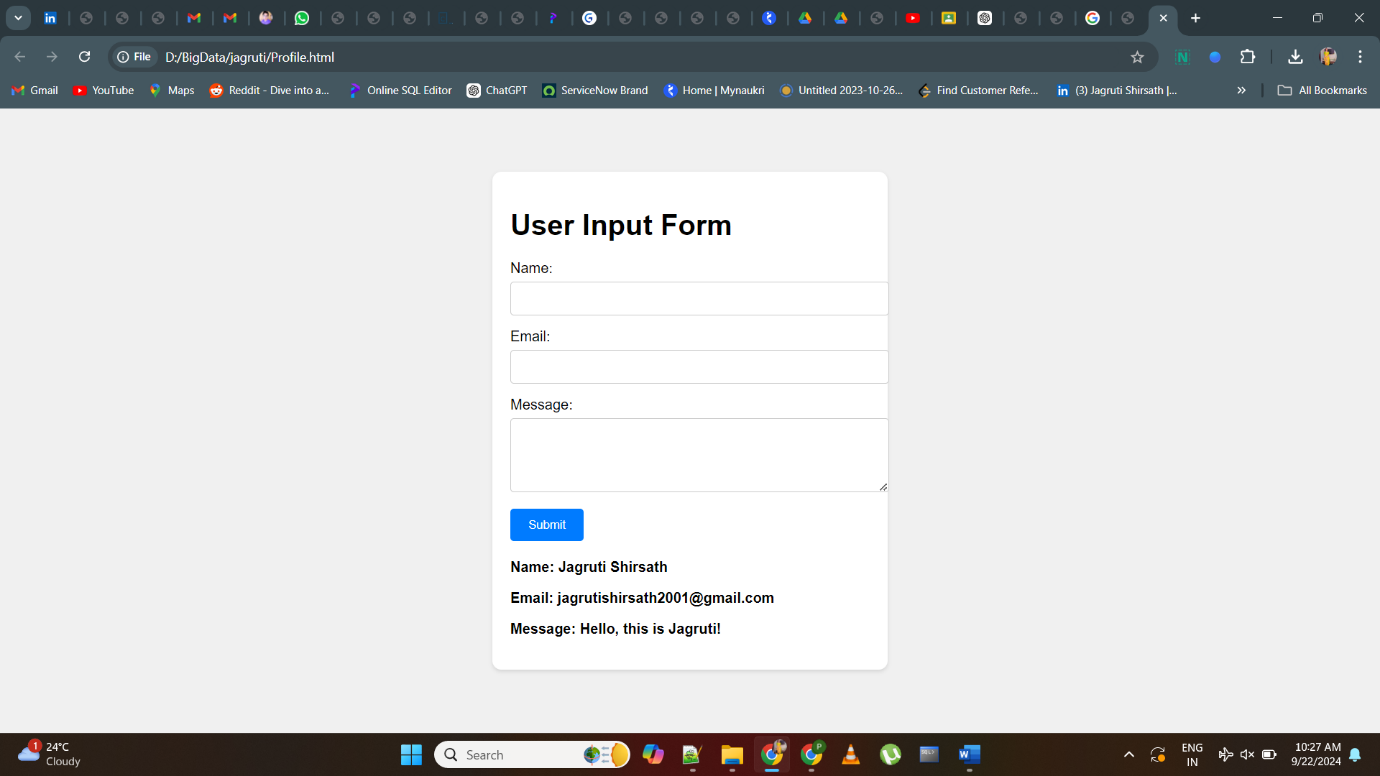
});

</script>

</body>

</html>

**Output:-**



**Program 4**

**Design a responsive website using CSS media queries to adapt to different screen sizes.**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Responsive Website</title>

<style>

/\* Reset some default styles \*/

body, h1, h2, p, ul, li {

margin: 0;

padding: 0;

}

body {

font-family: Arial, sans-serif;

background-color: #f0f0f0;

}

/\* Header Styles \*/

header {

background-color: #007bff;

color: #fff;

padding: 20px;

text-align: center;

}

nav ul {

list-style: none;

}

nav ul li {

display: inline;

margin-right: 20px;

}

nav ul li a {

text-decoration: none;

color: #fff;

}

/\* Main Content Styles \*/

main {

padding: 20px;

}

.content {

background-color: #fff;

border-radius: 5px;

padding: 20px;

box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);

}

/\* Footer Styles \*/

footer {

background-color: #333;

color: #fff;

text-align: center;

padding: 10px;

}

/\* Media Queries for Responsive Design \*/

@media (max-width: 768px) {

header {

padding: 10px;

}

nav ul li {

display: block;

margin-bottom: 10px;

}

main {

padding: 10px;

}

}

</style>

</head>

<body>

<header>

<h1>Responsive Website</h1>

<nav>

<ul>

<li><a href="#">Home</a></li>

<li><a href="#">About</a></li>

<li><a href="#">Services</a></li>

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

</ul>

</nav>

</header>

<main>

<section class="content">

<h2>Welcome to Our Website</h2>

<p>we're dedicated to delivering exceptional content that informs, inspires, and engages our visitors. Our platform is a

hub of knowledge, offering a wide range of articles, guides, and resources that cater to your interests.</p>

</section>

</main>

<footer>

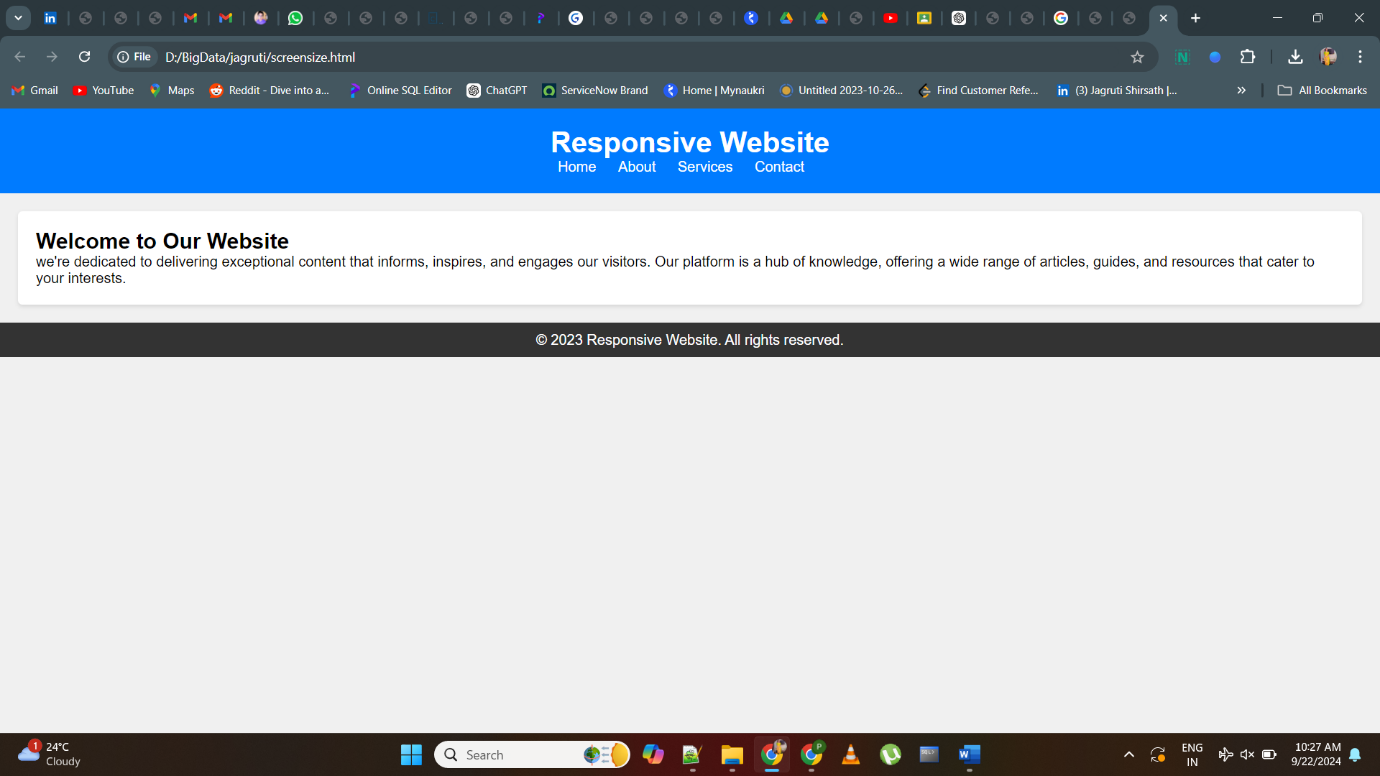
<p>&copy; 2023 Responsive Website. All rights reserved.</p>

</footer>

</body>

</html>

**Output:-**



**Program 5**

**Implement a CSS dropdown menu for website navigation.**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Dropdown Menu</title>

<style>

body {

font-family: Arial, sans-serif;

}

.menu {

background-color: #A9A9A9;

color: #000000;

}

.menu ul {

list-style: none;

padding: 0;

margin: 0;

display: flex;

}

.menu ul li {

margin-right: 20px;

position: relative;

}

.menu ul li a {

text-decoration: none;

color: black;

}

/\* Dropdown Styles \*/

.dropdown-content {

display: none;

position: absolute;

background-color: grey;

min-width: 160px;

box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);

z-index: 1;

}

.dropdown:hover .dropdown-content {

display: flex;

flex-direction: column;

}

</style>

</head>

<body>

<nav class="menu">

<ul>

<li><a href="#">Home</a></li>

<li class="dropdown">

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

<div class="dropdown-content">

<a href="#">Service 1</a>

<a href="#">Service 2</a>

<a href="#">Service 3</a>

</div>

</li>

<li><a href="#">About</a></li>

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

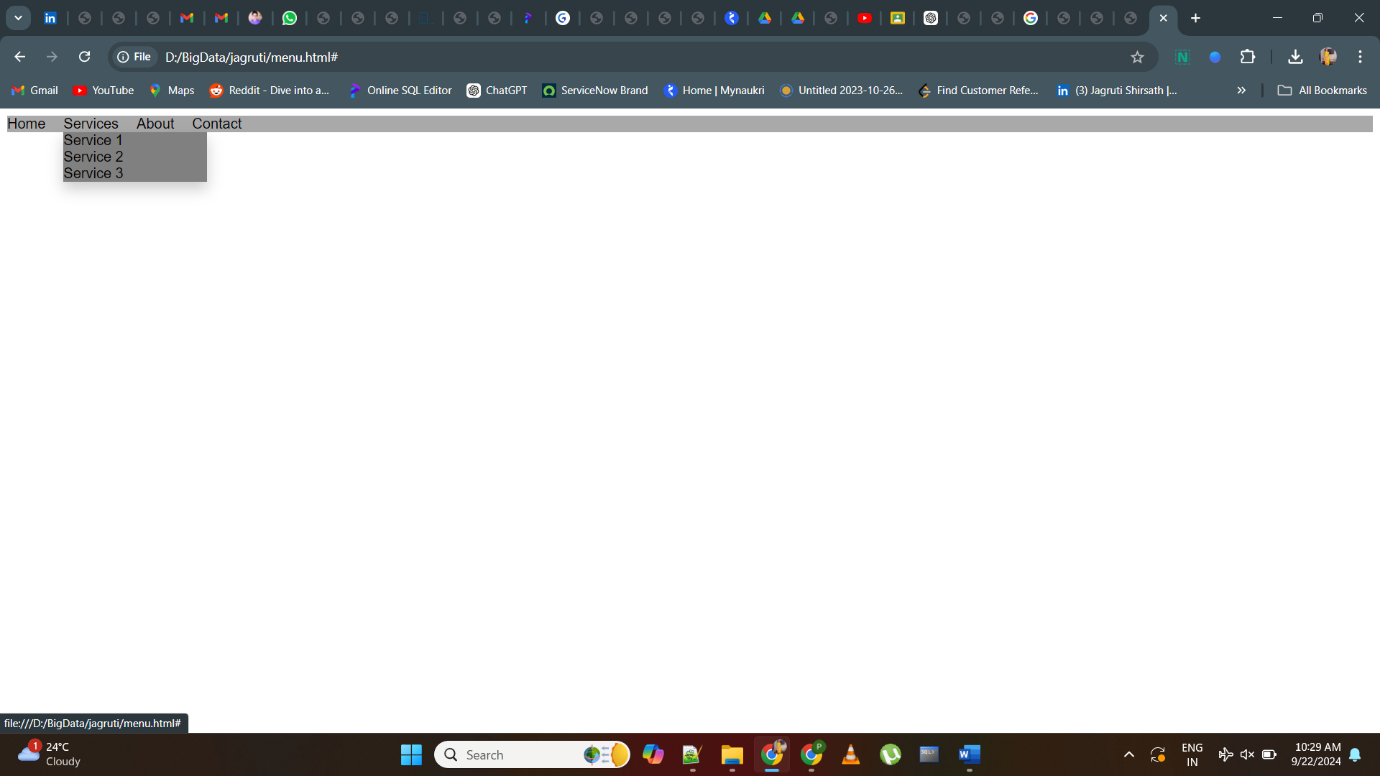
</ul>

</nav>

</body>

</html>

**Output:-**



**Program 6**

**Create a webpage with a table that dynamically populates data using JavaScript.**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Dynamic Table</title>

<style>

table {

border-collapse: collapse;

width: 100%;

margin-top: 20px;

}

th, td {

border: 1px solid #ccc;

padding: 8px;

text-align: left;

}

th {

background-color: #f5f5f5;

}

</style>

</head>

<body>

<h1>Dynamic Table Example</h1>

<table id="dataTable">

<thead>

<tr>

<th>Name</th>

<th>Age</th>

<th>Location</th>

</tr>

</thead>

<tbody></tbody>

</table>

<script>

const data = [

{ name: 'John', age: 25, location: 'New York' },

{ name: 'Jane', age: 30, location: 'Los Angeles' },

{ name: 'Mike', age: 22, location: 'Chicago' }

];

const tableBody = document.querySelector('#dataTable tbody');

data.forEach(item => {

const row = tableBody.insertRow();

const nameCell = row.insertCell(0);

const ageCell = row.insertCell(1);

const locationCell = row.insertCell(2);

nameCell.textContent = item.name;

ageCell.textContent = item.age;

locationCell.textContent = item.location;

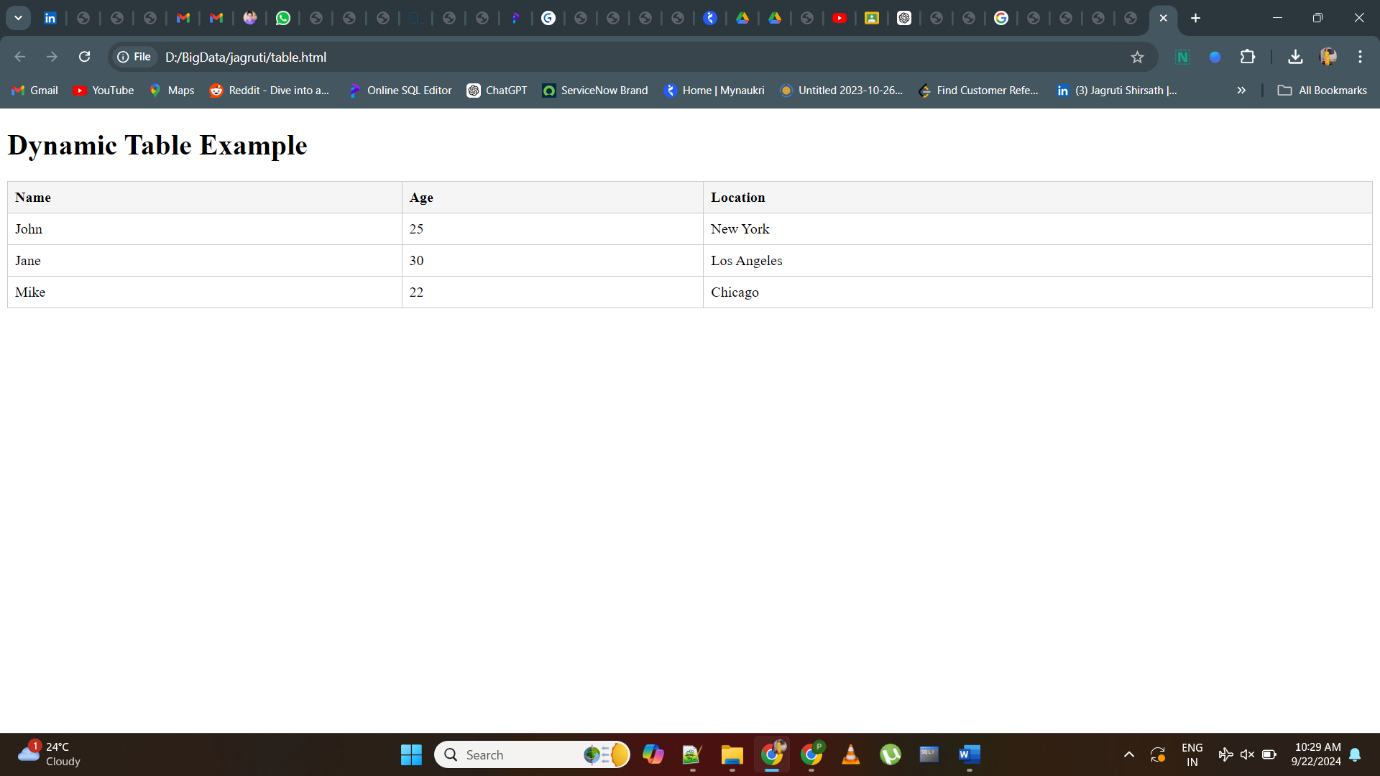
});

</script>

</body>

</html>

**Output:-**



**Program 7**

**Build a JavaScript calculator that performs basic arithmetic operations.**

<!DOCTYPE html>

<html>

<head>

<title>Basic Calculator</title>

<style>

#calculator {

width: 250px;

margin: 0 auto;

padding: 10px;

border: 1px solid #ccc;

border-radius: 5px;

}

input[type="number"] {

width: 100%;

padding: 5px;

margin: 5px 0;

}

button {

width: 48%;

padding: 5px;

margin: 5px;

}

</style>

</head>

<body>

<div id="calculator">

<input type="number" id="num1" placeholder="Enter a number">

<input type="number" id="num2" placeholder="Enter another number">

<button onclick="add()">Add</button>

<button onclick="subtract()">Subtract</button>

<button onclick="multiply()">Multiply</button>

<button onclick="divide()">Divide</button>

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

</div>

<script>

function add() {

var num1 = parseFloat(document.getElementById("num1").value);

var num2 = parseFloat(document.getElementById("num2").value);

var result = num1 + num2;

document.getElementById("result").textContent = "Result: " + result;

}

function subtract() {

var num1 = parseFloat(document.getElementById("num1").value);

var num2 = parseFloat(document.getElementById("num2").value);

var result = num1 - num2;

document.getElementById("result").textContent = "Result: " + result;

}

function multiply() {

var num1 = parseFloat(document.getElementById("num1").value);

var num2 = parseFloat(document.getElementById("num2").value);

var result = num1 \* num2;

document.getElementById("result").textContent = "Result: " + result;

}

function divide() {

var num1 = parseFloat(document.getElementById("num1").value);

var num2 = parseFloat(document.getElementById("num2").value);

if (num2 !== 0) {

var result = num1 / num2;

document.getElementById("result").textContent = "Result: " + result;

} else {

document.getElementById("result").textContent = "Result: Cannot divide by zero";

}

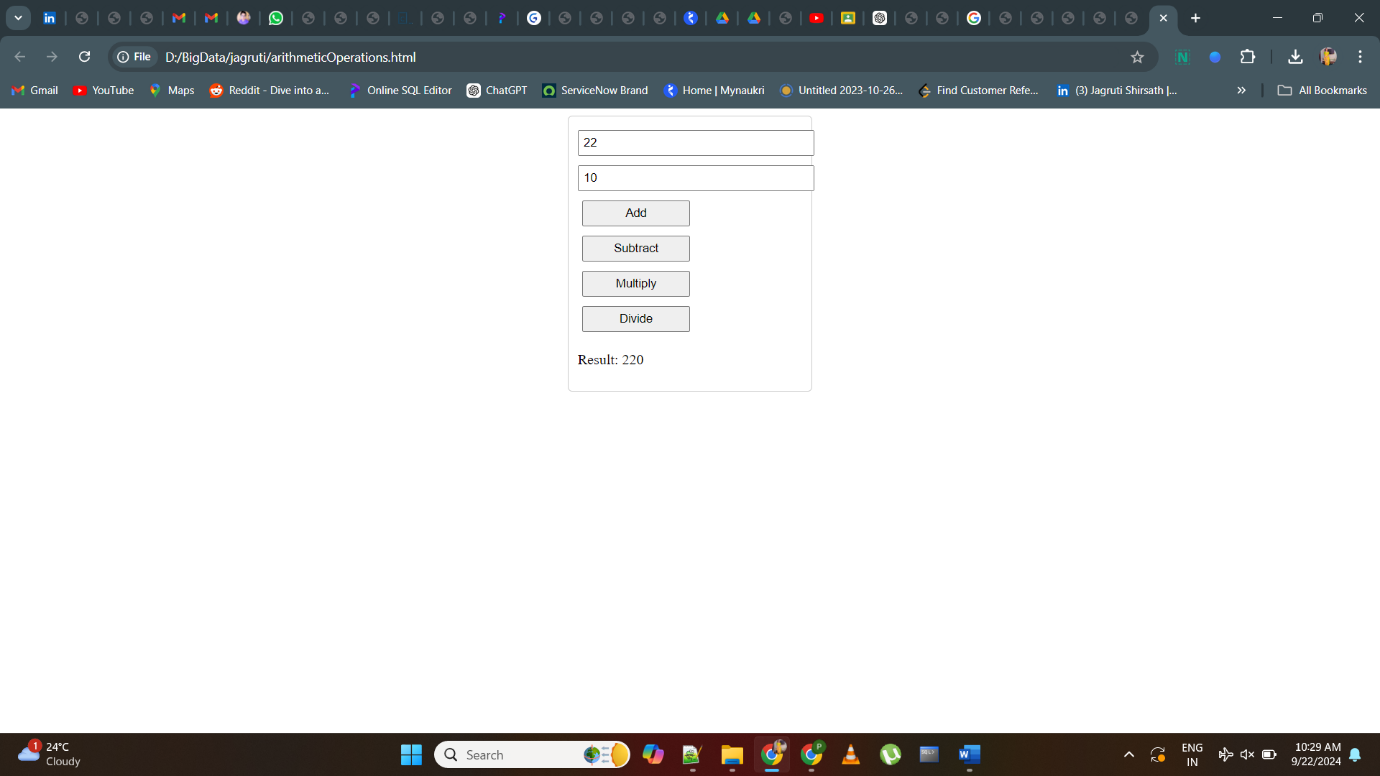
}

</script>

</body>

</html>

**Output:-**



**Program 8**

**Develop a JavaScript slideshow that displays images with automatic transitions.**

<!DOCTYPE html>

<html>

<head>

<title>Image Slideshow</title>

<style>

#slideshow {

max-width: 600px;

margin: 0 auto;

overflow: hidden;

position: relative;

}

#slideshow img {

width: 100%;

height: auto;

}

</style>

</head>

<body>

<div id="slideshow">

<img src="D:\BigData\jagruti\Almonds.jpg" alt="Image 1">

<img src="D:\BigData\jagruti\Cashewnuts.jpg" alt="Image 2">

<img src="D:\BigData\jagruti\raisins.jpg" alt="Image 3">

</div>

<script>

const images = document.querySelectorAll("#slideshow img");

let currentImageIndex = 0;

function showImage(index) {

images.forEach((image, i) => {

if (i === index) {

image.style.display = "block";

} else {

image.style.display = "none";

}

});

}

function nextImage() {

currentImageIndex = (currentImageIndex + 1) % images.length;

showImage(currentImageIndex);

}

setInterval(nextImage, 3000); // Automatic transition every 3 seconds

// Initial setup

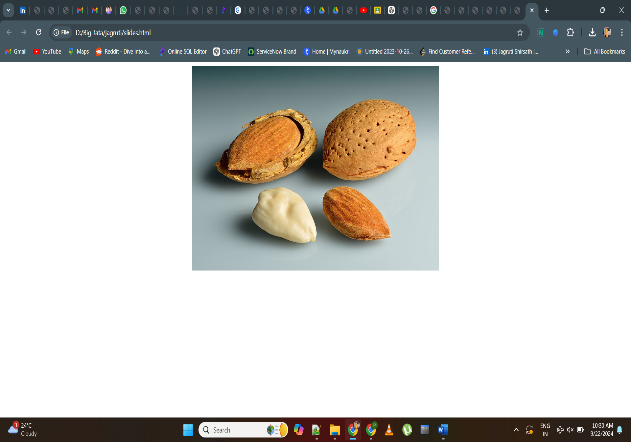
showImage(currentImageIndex);

</script>

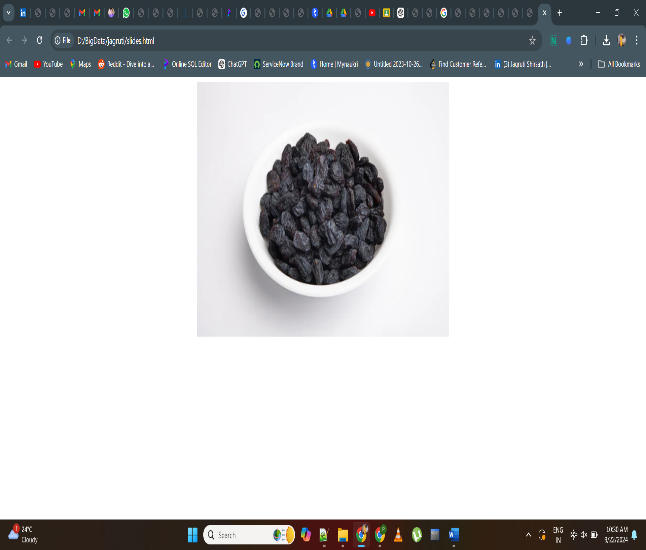
</body>

</html>

**Output:-**







**Program 9**

**Implement from validation using JavaScript to ensure proper user input.**

<!DOCTYPE html>

<html>

<head>

<title>Form Validation Example</title>

<style>

.error {

color: grey;

font-size: 0.8rem;

display: block;

margin-top: 5px;

}

.input-container {

margin-bottom: 10px;

}

</style>

</head>

<body>

<h1>Registration Form</h1>

<form id="registrationForm" onsubmit="return validateForm()">

<div class="input-container">

<label for="username">Username:</label>

<input type="text" id="username" name="username">

<span id="usernameError" class="error"></span>

</div>

<div class="input-container">

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

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

<span id="emailError" class="error"></span>

</div>

<div class="input-container">

<label for="password">Password:</label>

<input type="password" id="password" name="password">

<span id="passwordError" class="error"></span>

</div>

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

</form>

<script>

function validateForm() {

const username = document.getElementById("username").value;

const email = document.getElementById("email").value;

const password = document.getElementById("password").value;

// Reset error messages

document.getElementById("usernameError").textContent = "";

document.getElementById("emailError").textContent = "";

document.getElementById("passwordError").textContent = "";

let isValid = true;

// Validate username (required)

if (username === "") {

document.getElementById("usernameError").textContent = "Username is required";

isValid = false;

}

// Validate email (required and format)

if (email === "") {

document.getElementById("emailError").textContent = "Email is required";

isValid = false;

} else if (!validateEmail(email)) {

document.getElementById("emailError").textContent = "Invalid email format";

isValid = false;

}

// Validate password (required and minimum length)

if (password === "") {

document.getElementById("passwordError").textContent = "Password is required";

isValid = false;

} else if (password.length < 8) {

document.getElementById("passwordError").textContent = "Password must be at least 8 characters";

isValid = false;

}

return isValid;

}

function validateEmail(email) {

const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;

return emailRegex.test(email);

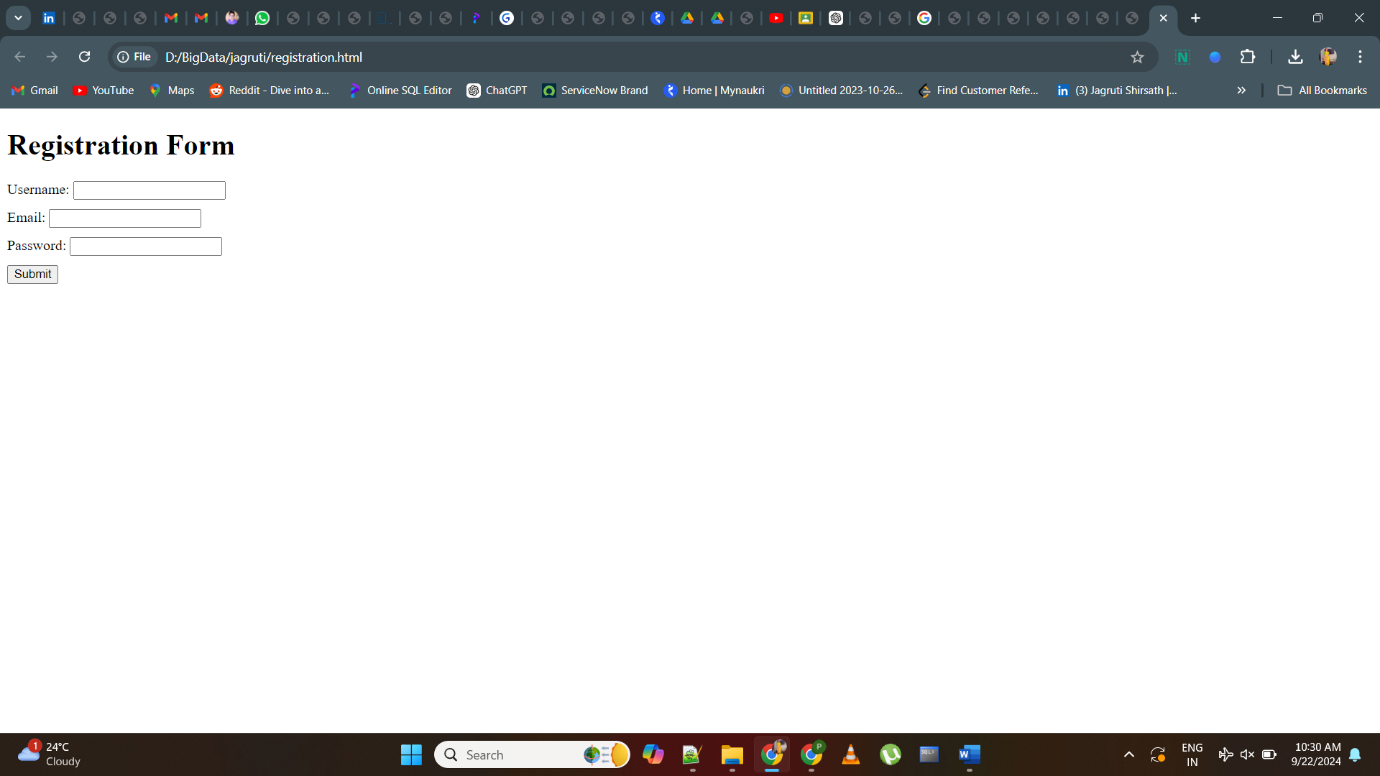
}

</script>

</body>

</html>

**Output:-**



**Program 10**

**Design a webpage layout using CSS grid or flexbox for responsive design.**

<!DOCTYPE html>

<html>

<head>

<title>Responsive Layout Example</title>

<style>

body {

margin: 0;

font-family: Arial, sans-serif;

background-color: #f0f0f0;

}

.container {

display: grid;

grid-template-columns: 1fr 2fr;

grid-gap: 20px;

padding: 20px;

max-width: 1200px;

margin: 0 auto;

background-color: white;

}

.sidebar {

background-color: #e0e0e0;

padding: 20px;

}

.main-content {

display: flex;

flex-direction: column;

justify-content: space-between;

padding: 20px;

background-color: #ffffff;

}

.header {

text-align: center;

background-color: #333;

color: pink;

padding: 10px;

}

.footer {

text-align: center;

background-color: #333;

color: pink;

padding: 10px;

}

@media screen and (max-width: 768px) {

.container {

grid-template-columns: 1fr;

}

.main-content {

flex-direction: column;

}

}

</style>

</head>

<body>

<div class="header">

<h1>Responsive Layout</h1>

</div>

<div class="container">

<div class="sidebar">

<h2>Sidebar</h2>

<p>This is the sidebar content.</p>

</div>

<div class="main-content">

<h2>Main Content</h2>

<p>This is the main content section.</p>

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec sit amet venenatis dui.</p>

</div>

</div>

<div class="footer">

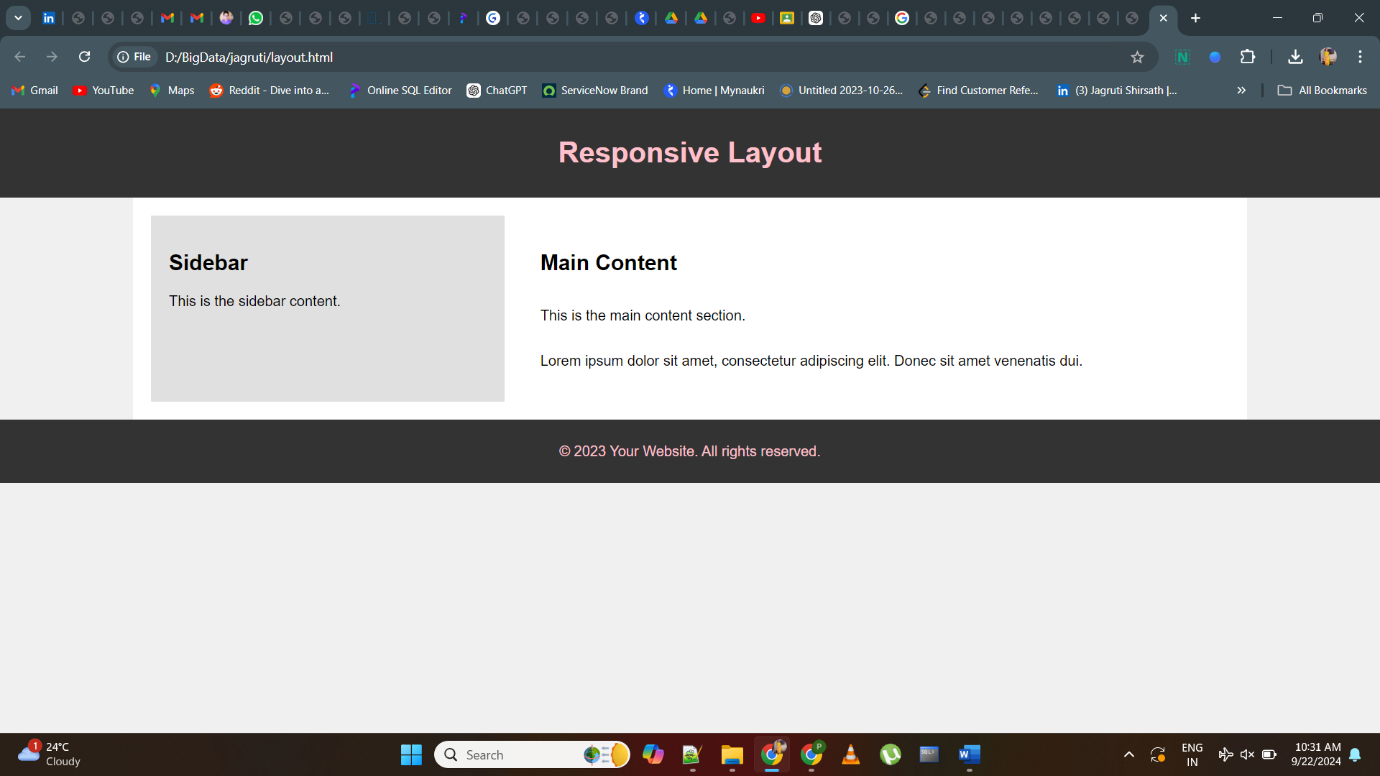
<p>&copy; 2023 Your Website. All rights reserved.</p>

</div>

</body>

</html>

**Output:-**



**Program 11**

**Build a website that incorporates CSS animations for visual effects.**

<!DOCTYPE html>

<html>

<head>

<title>CSS Animation Example</title>

<style>

body {

margin: 0;

font-family: Arial, sans-serif;

background-color: #f0f0f0;

}

.header {

position: relative;

height: 500px;

background: linear-gradient(rgba(0, 0, 0, 0.7), rgba(0, 0, 0, 0.7)), url('header-background.jpg');

background-size: cover;

background-position: center;

color: pink;

text-align: center;

display: flex;

flex-direction: column;

justify-content: center;

}

.header h1 {

margin: 0;

font-size: 3rem;

}

.button-container {

display: flex;

justify-content: center;

margin-top: 20px;

}

.button {

background-color: #3498db;

color: pink;

padding: 10px 20px;

border: none;

border-radius: 5px;

font-size: 1rem;

cursor: pointer;

margin: 0 10px;

transition: background-color 0.3s ease;

}

.button:hover {

background-color: #2980b9;

}

.animation-box {

width: 100px;

height: 100px;

background-color: #e74c3c;

margin: 20px auto;

animation: moveBox 2s infinite alternate ease-in-out;

}

@keyframes moveBox {

0% {

transform: translateX(0);

}

100% {

transform: translateX(200px);

}

}

</style>

</head>

<body>

<div class="header">

<h1>Welcome to Our Website</h1>

<div class="button-container">

<button class="button">Learn More</button>

<button class="button">Get Started</button>

</div>

</div>

<div class="animation-box"></div>

<div class="footer">

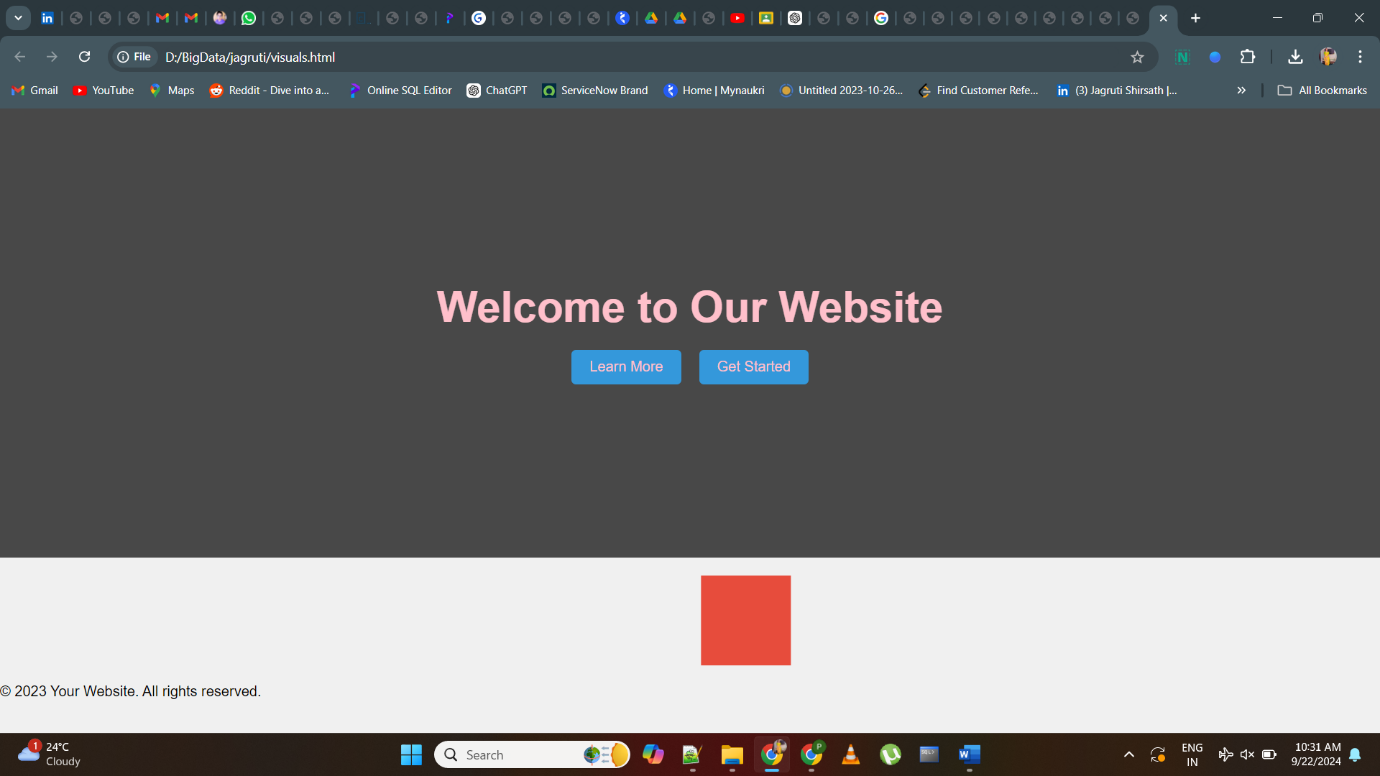
<p>&copy; 2023 Your Website. All rights reserved.</p>

</div>

</body>

</html>

**Output:-**



**Program 12**

**Develop a webpage with a contact form that sends user input to a server using AJAX.**

<!DOCTYPE html>

<html>

<head>

<title>Contact Form with AJAX</title>

<script>

function submitForm() {

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

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

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

var xhr = new XMLHttpRequest();

xhr.open("POST", "process\_form.php", true);

xhr.setRequestHeader("Content-type", "application/x-www-form-urlencoded");

xhr.onreadystatechange = function () {

if (xhr.readyState === 4 && xhr.status === 200) {

document.getElementById("result").innerHTML = xhr.responseText;

}

};

xhr.send("name=" + name + "&email=" + email + "&message=" + message);

}

</script>

</head>

<body>

<h1>Contact Us</h1>

<form>

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

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

<br><br>

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

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

<br><br>

<label for="message">Message:</label>

<textarea id="message" name="message" rows="4" required></textarea>

<br><br>

<input type="button" value="Submit" onclick="submitForm()">

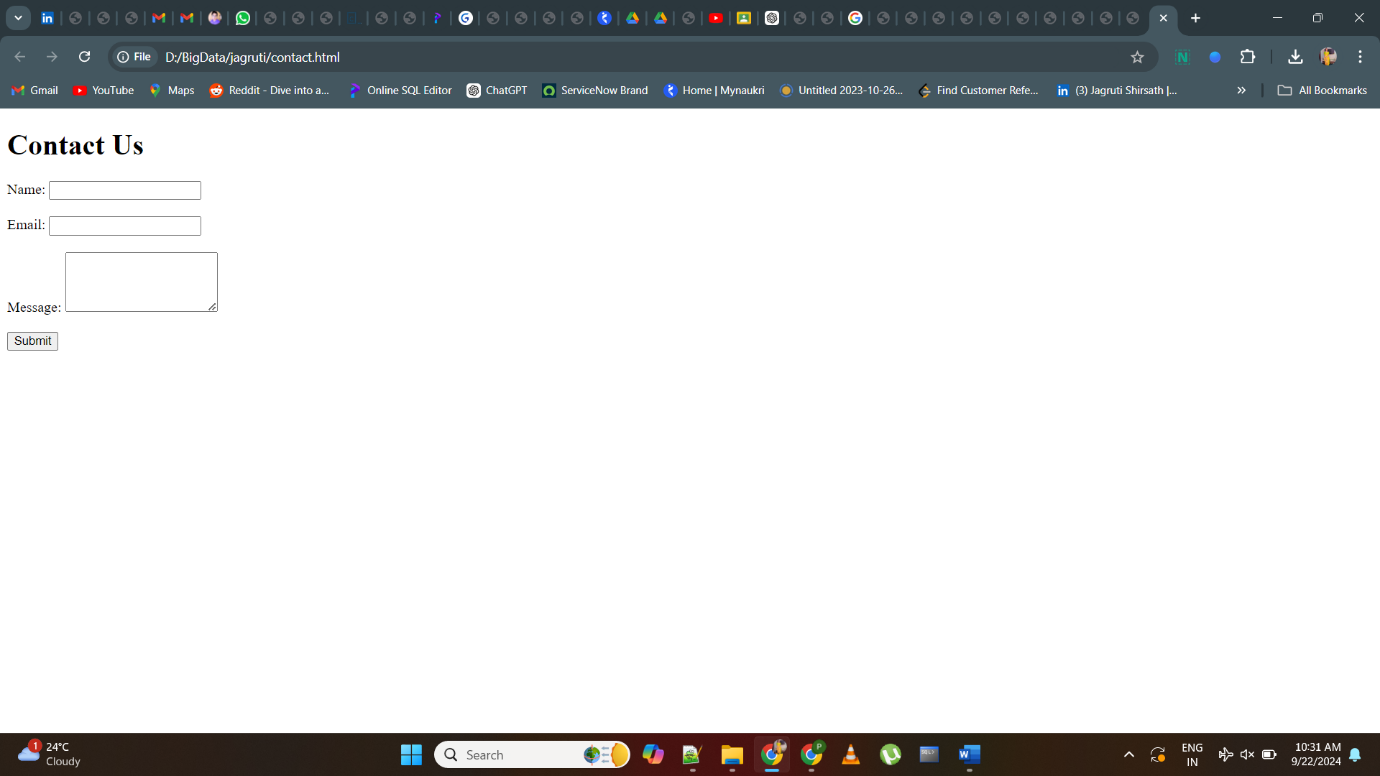
</form>

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

</body>

</html>

**Output:-**



**Program 13**

**Create a JavaScript quiz that presents multiple-choice questions and tracks the user’s score.**

<!DOCTYPE html>

<html>

<head>

<title>JavaScript Quiz</title>

<style>

body {

font-family: Times New Roman, Serif;

}

.quiz-container {

max-width: 600px;

margin: 0 auto;

padding: 20px;

background-color: #f5f5f5;

border-radius: 5px;

}

.question {

margin-bottom: 10px;

}

.options {

margin-top: 5px;

}

.result {

font-weight: bold;

margin-top: 20px;

}

</style>

</head>

<body>

<div class="quiz-container">

<h1>JavaScript Quiz</h1>

<div class="question" id="question"></div>

<div class="options" id="options"></div>

<button id="nextButton">Next</button>

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

</div>

<script>

const questions = [

{

question: "What is the capital of France?",

options: ["Paris", "Madrid", "Rome", "Berlin"],

correctIndex: 0

},

{

question: "Which planet is known as the Red Planet?",

options: ["Earth", "Mars", "Venus", "Jupiter"],

correctIndex: 1

},

{

question: "What is 2 + 2?",

options: ["3", "4", "5", "6"],

correctIndex: 1

}

];

let currentQuestionIndex = 0;

let score = 0;

const questionElement = document.getElementById("question");

const optionsElement = document.getElementById("options");

const resultElement = document.getElementById("result");

const nextButton = document.getElementById("nextButton");

function loadQuestion(index) {

const question = questions[index];

questionElement.textContent = question.question;

optionsElement.innerHTML = "";

question.options.forEach((option, i) => {

const optionButton = document.createElement("button");

optionButton.textContent = option;

optionButton.addEventListener("click", () => checkAnswer(i));

optionsElement.appendChild(optionButton);

});

}

function checkAnswer(selectedIndex) {

const currentQuestion = questions[currentQuestionIndex];

if (selectedIndex === currentQuestion.correctIndex) {

score++;

resultElement.textContent = "Correct!";

} else {

resultElement.textContent = "Incorrect!";

}

nextButton.disabled = false;

optionsElement.querySelectorAll("button").forEach(button => {

button.disabled = true;

});

}

function nextQuestion() {

currentQuestionIndex++;

if (currentQuestionIndex < questions.length) {

loadQuestion(currentQuestionIndex);

resultElement.textContent = "";

nextButton.disabled = true;

optionsElement.querySelectorAll("button").forEach(button => {

button.disabled = false;

});

} else {

questionElement.textContent = "Quiz completed!";

optionsElement.innerHTML = "";

resultElement.textContent = `Your score: ${score}/${questions.length}`;

nextButton.disabled = true;

}

}

loadQuestion(currentQuestionIndex);

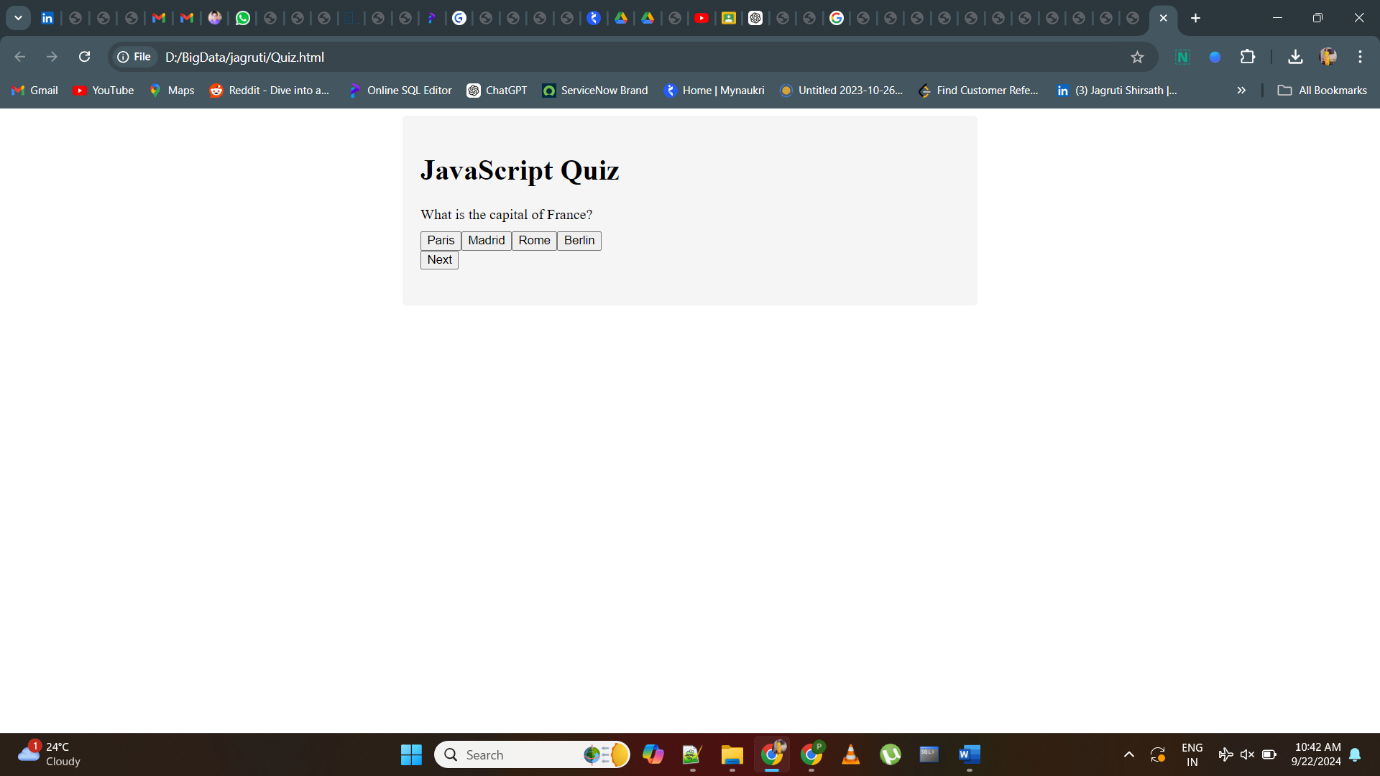
nextButton.addEventListener("click", nextQuestion);

</script>

</body>

</html>

**Output:-**



**Program 14**

**Implement client-side form validation using HTML5 form input types and attributes.**

<!DOCTYPE html>

<html>

<head>

<title>Client-Side Form Validation</title>

<style>

body {

font-family: Times New Roman, Serif;

}

.error {

color: pink;

}

</style>

</head>

<body>

<h1>Contact Us</h1>

<form id="contactForm" onsubmit="return validateForm()">

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

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

<span class="error" id="nameError"></span>

<br><br>

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

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

<span class="error" id="emailError"></span>

<br><br>

<label for="message">Message:</label>

<textarea id="message" name="message" rows="4" required></textarea>

<span class="error" id="messageError"></span>

<br><br>

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

</form>

<script>

function validateForm() {

const nameInput = document.getElementById("name");

const emailInput = document.getElementById("email");

const messageInput = document.getElementById("message");

const nameError = document.getElementById("nameError");

const emailError = document.getElementById("emailError");

const messageError = document.getElementById("messageError");

nameError.textContent = "";

emailError.textContent = "";

messageError.textContent = "";

let isValid = true;

if (nameInput.validity.valueMissing) {

nameError.textContent = "Name is required";

isValid = false;

}

if (emailInput.validity.valueMissing) {

emailError.textContent = "Email is required";

isValid = false;

} else if (emailInput.validity.typeMismatch) {

emailError.textContent = "Invalid email format";

isValid = false;

}

if (messageInput.validity.valueMissing) {

messageError.textContent = "Message is required";

isValid = false;

}

return isValid;

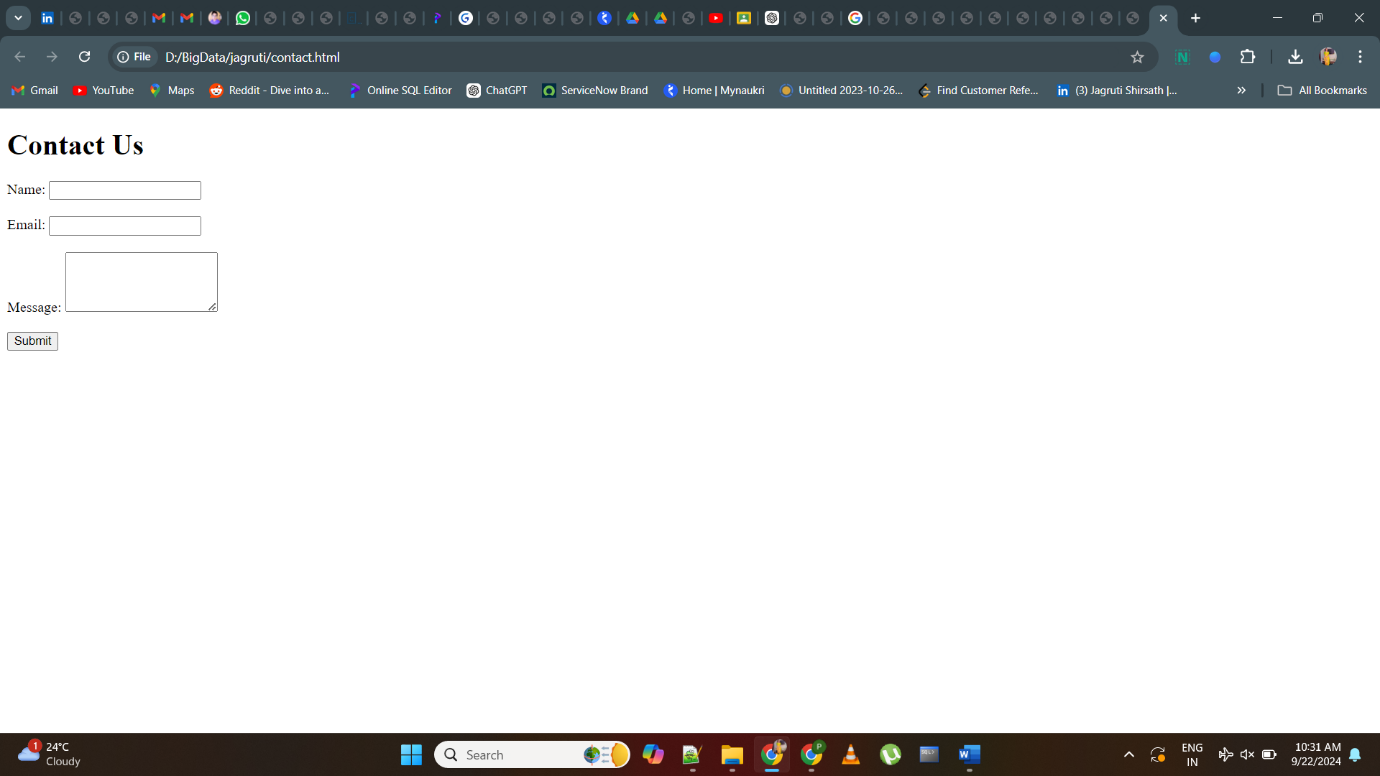
}

</script>

</body>

</html>

**Output:-**



**Program 15**

**Build a responsive image gallery using HTML, CSS and JavaScript**.

<!DOCTYPE html>

<html>

<head>

<title>Responsive Image Gallery</title>

<style>

body {

margin: 0;

font-family: Times New Roman, Serif;

background-color: #808080;

}

.gallery {

display: grid;

grid-template-columns: repeat(auto-fill, minmax(200px, 1fr));

gap: 10px;

padding: 20px;

}

.gallery img {

max-width: 100%;

height: auto;

cursor: pointer;

transition: transform 0.2s;

}

.modal {

display: none;

position: fixed;

top: 0;

left: 0;

width: 100%;

height: 100%;

background-color: rgba(0, 0, 0, 0.8);

align-items: center;

justify-content: center;

}

.modal img {

max-width: 90%;

max-height: 90%;

}

.close-btn {

position: absolute;

top: 10px;

right: 10px;

color: white;

font-size: 20px;

cursor: pointer;

}

</style>

</head>

<body>

<div class="gallery">

<img src="D:\BigData\jagruti\Almonds.jpg" alt="Image 1" onclick="openModal('image1.jpg')">

<img src="D:\BigData\jagruti\Cashewnuts.jpg"" alt="Image 2" onclick="openModal('image2.jpg')">

<img src="D:\BigData\jagruti\raisins.jpg" alt="Image 3" onclick="openModal('image3.jpg')">

<!-- Add more images here -->

</div>

<div class="modal" id="modal">

<span class="close-btn" onclick="closeModal()">&times;</span>

<img id="modalImage">

</div>

<script>

function openModal(imageSrc) {

const modal = document.getElementById("modal");

const modalImage = document.getElementById("modalImage");

modal.style.display = "flex";

modalImage.src = imageSrc;

}

function closeModal() {

const modal = document.getElementById("modal");

modal.style.display = "none";

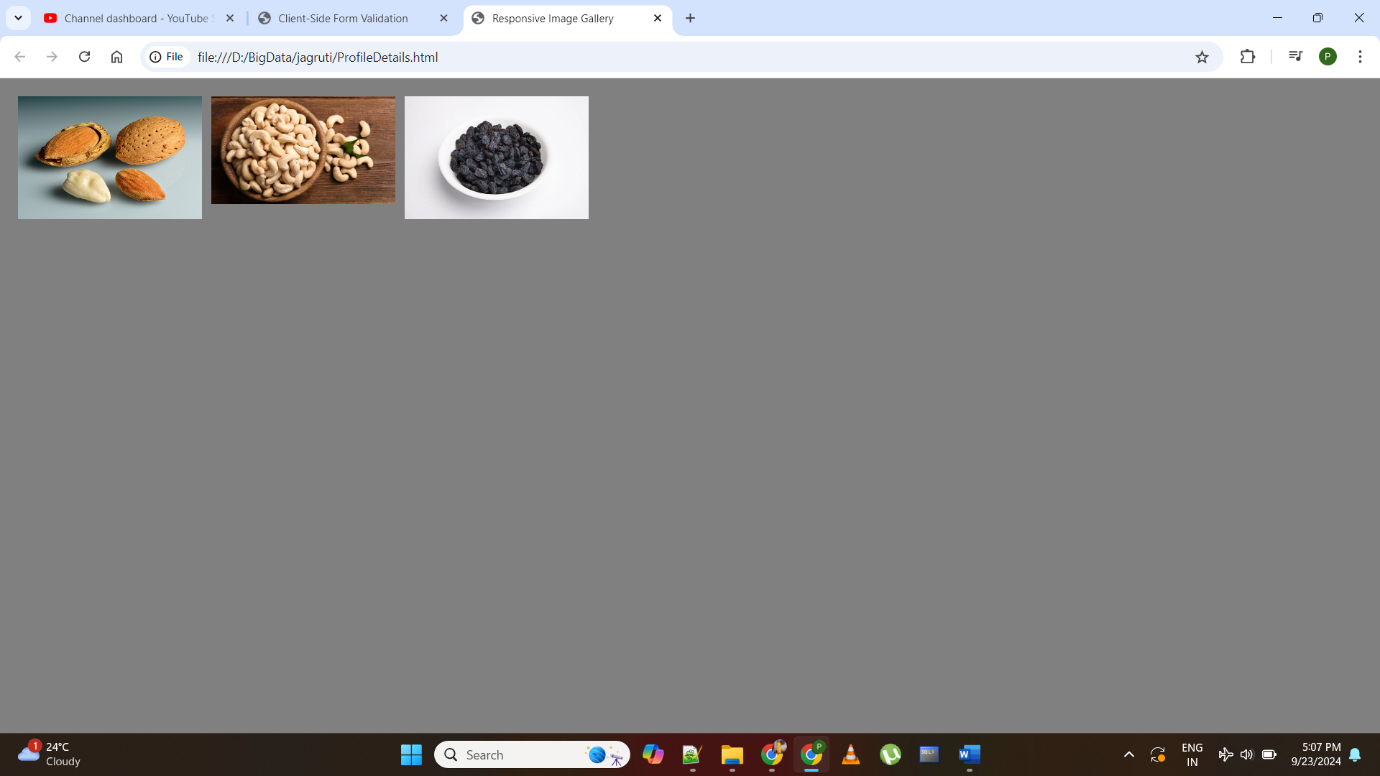
}

</script>

</body>

</html>

**Output:-**



**Program 16**

**Design a webpage layout using Bootstrap framework.**

<!DOCTYPE html>

<html>

<head>

<title>Bootstrap Webpage Layout</title>

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

<style>

body {

font-family: Arial, sans-serif;

}

.header {

background-color: #333;

color: pink;

padding: 10px;

text-align: center;

}

.content {

padding: 20px;

}

.footer {

background-color: #333;

color: pink;

padding: 10px;

text-align: center;

}

</style>

</head>

<body>

<div class="header">

<h1>Welcome to Our Website</h1>

</div>

<div class="container content">

<div class="row">

<div class="col-md-6">

<h2>About Us</h2>

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla facilisi.</p>

</div>

<div class="col-md-6">

<h2>Contact Us</h2>

<p>Email: contact@example.com</p>

<p>Phone: +1 123-456-7890</p>

</div>

</div>

</div>

<div class="footer">

<p>&copy; 2023 Your Website. All rights reserved.</p>

</div>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

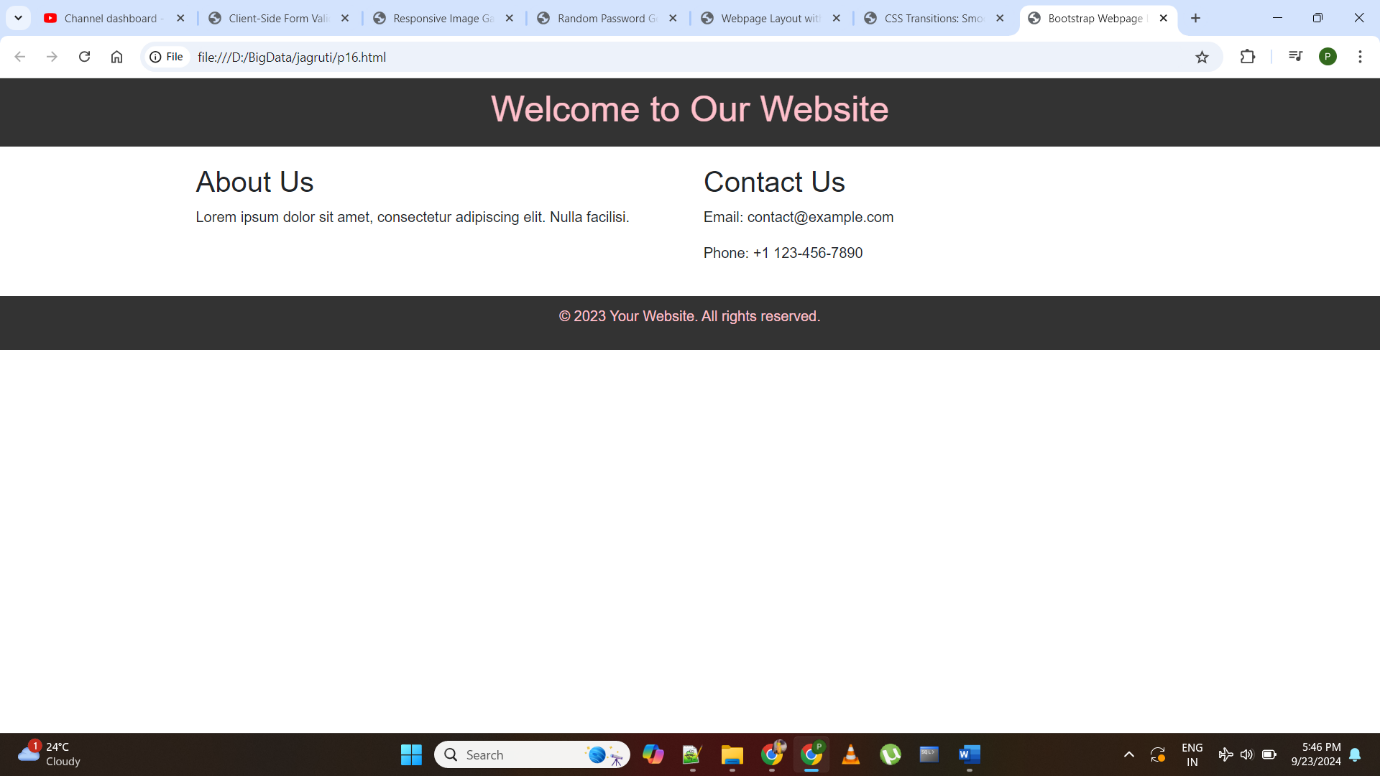
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

</body>

</html>

**Output:-**



**Program 17**

**Create a JavaScript program that generates a random password based on user specified criteria.**

<!DOCTYPE html>

<html>

<head>

<title>Random Password Generator</title>

<style>

body {

font-family: Times New Roman, Serif;

}

.generator {

max-width: 300px;

margin: 0 auto;

padding: 20px;

border: 1px solid #ccc;

border-radius: 5px;

background-color: #f0f0f0;

text-align: center;

}

.output {

margin-top: 20px;

font-weight: bold;

}

</style>

</head>

<body>

<div class="generator">

<h1>Random Password Generator</h1>

<form id="passwordForm">

<label for="length">Password Length:</label>

<input type="number" id="length" name="length" min="4" max="50" required>

<br><br>

<label for="uppercase">Include Uppercase Letters:</label>

<input type="checkbox" id="uppercase" name="uppercase">

<br><br>

<label for="lowercase">Include Lowercase Letters:</label>

<input type="checkbox" id="lowercase" name="lowercase">

<br><br>

<label for="digits">Include Digits:</label>

<input type="checkbox" id="digits" name="digits">

<br><br>

<label for="special">Include Special Characters:</label>

<input type="checkbox" id="special" name="special">

<br><br>

<input type="submit" value="Generate Password">

</form>

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

</div>

<script>

const form = document.getElementById("passwordForm");

const output = document.getElementById("output");

form.addEventListener("submit", function (event) {

event.preventDefault();

const length = parseInt(document.getElementById("length").value);

const uppercase = document.getElementById("uppercase").checked;

const lowercase = document.getElementById("lowercase").checked;

const digits = document.getElementById("digits").checked;

const special = document.getElementById("special").checked;

const charSets = [];

if (uppercase) charSets.push("ABCDEFGHIJKLMNOPQRSTUVWXYZ");

if (lowercase) charSets.push("abcdefghijklmnopqrstuvwxyz");

if (digits) charSets.push("0123456789");

if (special) charSets.push("!@#$%^&\*()\_+-=[]{}|;:,.<>?");

const allChars = charSets.join("");

let password = "";

if (charSets.length > 0 && length > 0) {

for (let i = 0; i < length; i++) {

const randomIndex = Math.floor(Math.random() \* allChars.length);

password += allChars.charAt(randomIndex);

}

output.textContent = `Generated Password: ${password}`;

} else {

output.textContent = "Please select at least one character set and specify a password length.";

}

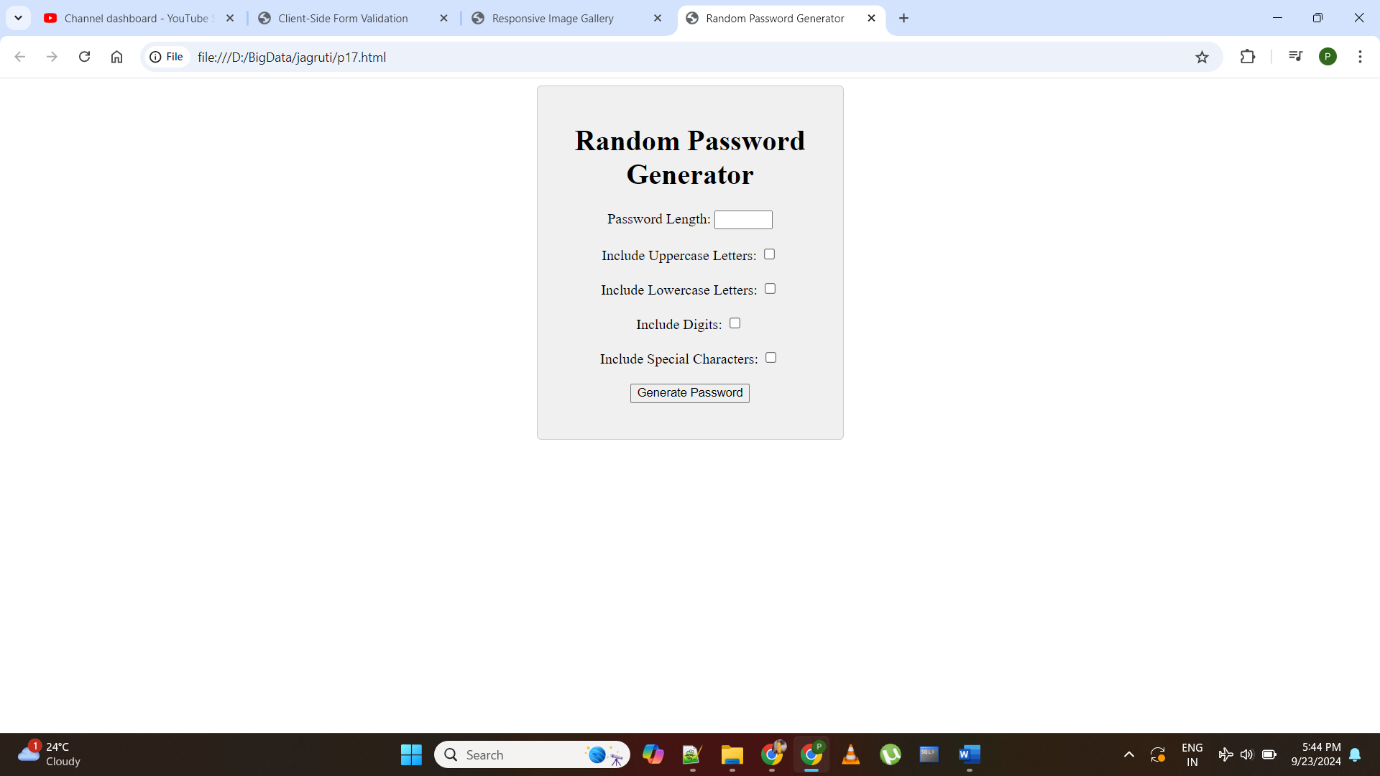
});

</script>

</body>

</html>

**Output:-**



**Program 18**

**Develop a website that utilizes cookies for user session management.**

<!DOCTYPE html>

<html>

<head>

<title>Session Management with Cookies</title>

<style>

body {

font-family: Times New Roman, Serif;

background-color: #f0f0f0;

margin: 0;

padding: 0;

display: flex;

justify-content: center;

align-items: center;

height: 90vh;

}

.container {

text-align: center;

padding: 20px;

border-radius: 5px;

background-color: #ffffff;

box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);

}

input[type="text"] {

padding: 5px;

border: 1px solid #ccc;

border-radius: 3px;

margin-right: 10px;

}

button {

padding: 10px 20px;

background-color: #007bff;

border: none;

border-radius: 3px;

color: pink;

cursor: pointer;

}

#welcomeMessage {

margin-top: 20px;

}

</style>

</head>

<body>

<div class="container">

<h1>Welcome to Our Website</h1>

<p>Enter your name to continue:</p>

<input type="text" id="usernameInput" placeholder="Your Name">

<button id="submitButton">Submit</button>

<p id="welcomeMessage"></p>

</div>

<script>

const usernameInput = document.getElementById("usernameInput");

const submitButton = document.getElementById("submitButton");

const welcomeMessage = document.getElementById("welcomeMessage");

// Function to set a cookie with a specified name, value, and expiration time

function setCookie(name, value, days) {

const date = new Date();

date.setTime(date.getTime() + (days \* 24 \* 60 \* 60 \* 1000));

const expires = "expires=" + date.toUTCString();

document.cookie = name + "=" + value + ";" + expires + ";path=/";

}

// Function to get the value of a cookie by its name

function getCookie(name) {

const decodedCookies = decodeURIComponent(document.cookie);

const cookiesArray = decodedCookies.split(";");

for (let i = 0; i < cookiesArray.length; i++) {

const cookie = cookiesArray[i].trim();

if (cookie.indexOf(name + "=") === 0) {

return cookie.substring(name.length + 1, cookie.length);

}

}

return "";

}

submitButton.addEventListener("click", function () {

const username = usernameInput.value;

if (username.trim() !== "") {

setCookie("username", username, 30); // Set a cookie that expires in 30 days

welcomeMessage.textContent = "Welcome, " + username + "!";

}

});

// Check if the user has a session cookie

const storedUsername = getCookie("username");

if (storedUsername !== "") {

welcomeMessage.textContent = "Welcome back, " + storedUsername + "!";

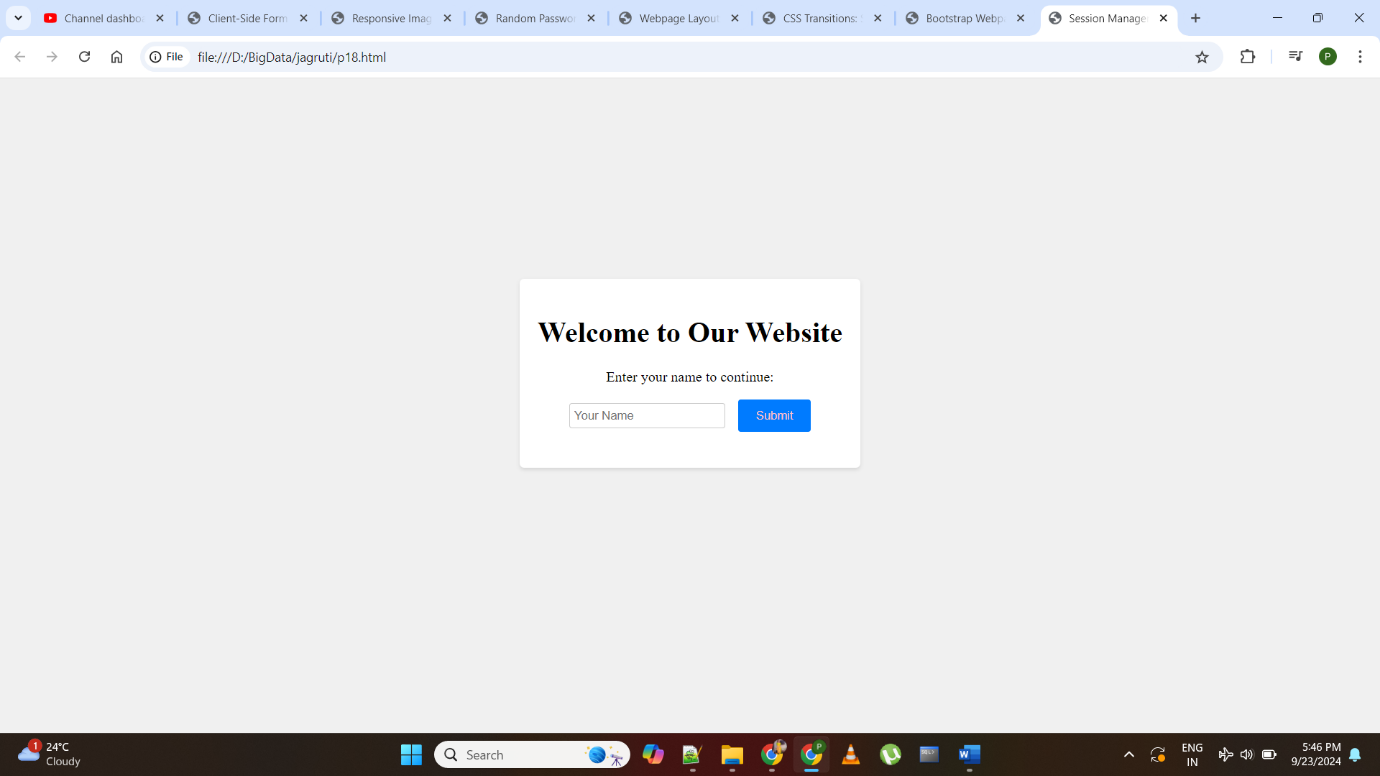
}

</script>

</body>

</html>

**Output:-**



**Program 19**

**Implement CSS transitions to create smooth hover effects on webpage elements.**

<!DOCTYPE html>

<html>

<head>

<title>CSS Transitions: Smooth Hover Effects</title>

<style>

body {

font-family: Times New Roman, Serif;

background-color: #f0f0f0;

margin: 0;

padding: 0;

}

.box {

width: 150px;

height: 150px;

background-color: #007bff;

color: grey;

text-align: center;

line-height: 150px;

margin: 20px;

border-radius: 5px;

cursor: pointer;

transition: background-color 0.3s, transform 0.3s;

}

.box:hover {

background-color: #0056b3;

transform: scale(1.1);

}

</style>

</head>

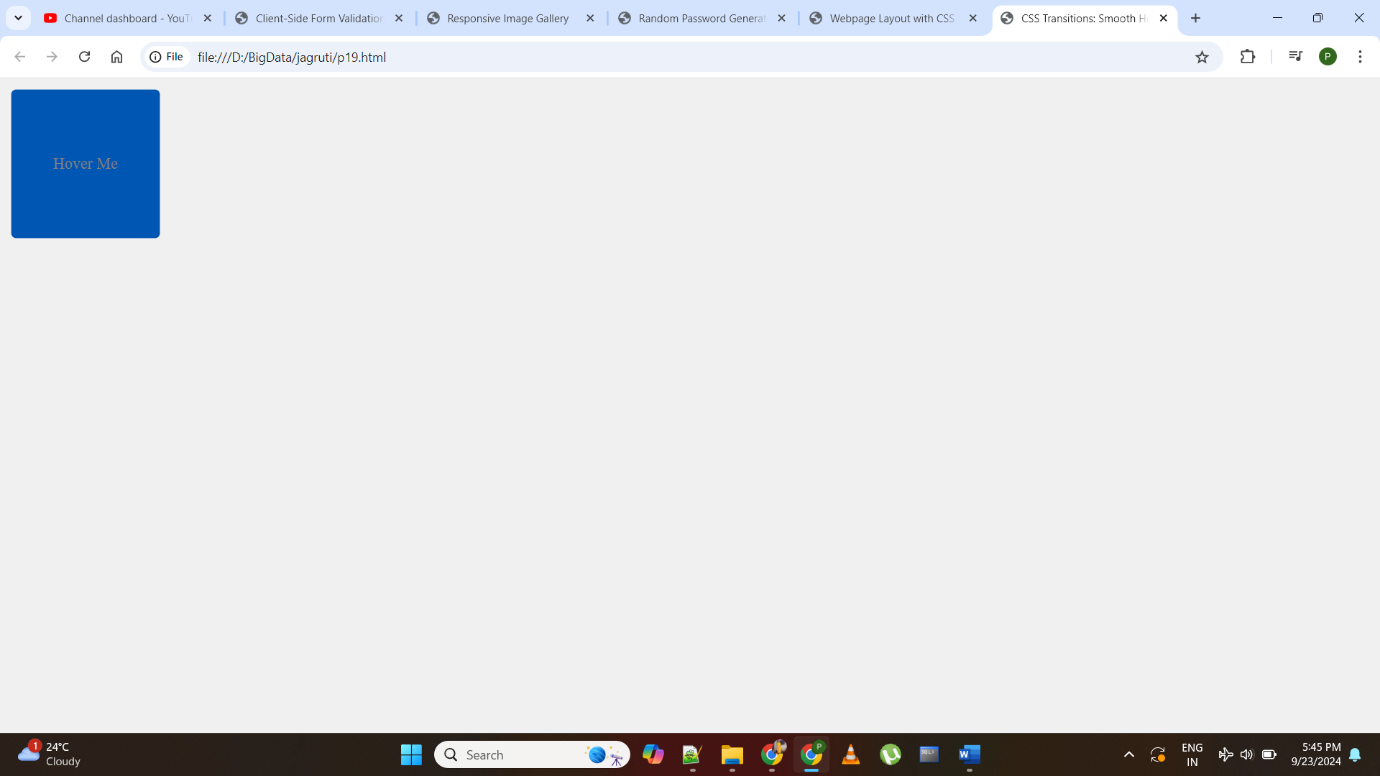
<body>

<div class="box">Hover Me</div>

</body>

</html>

**Output:-**



**Program 20**

**Build a JavaScript program that performs basic string manipulation tasks.**

<html>

<head>

<title>String Manipulation with CSS Styling</title>

<style>

body {

font-family: Times New Roman, Serif;

background-color: #f0f0f0;

margin: 0;

padding: 0;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

}

.container {

text-align: center;

padding: 20px;

border-radius: 5px;

background-color: #ffffff;

box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);

}

label {

font-weight: bold;

margin-right: 10px;

}

input[type="text"] {

padding: 5px;

border: 1px solid #ccc;

border-radius: 3px;

margin-right: 10px;

}

button {

padding: 10px 20px;

background-color: #007bff;

border: none;

border-radius: 3px;

color: white;

cursor: pointer;

}

p {

margin-top: 20px;

font-weight: bold;

}

</style>

</head>

<body>

<div class="container">

<h1>String Manipulation with CSS Styling</h1>

<form id="stringManipulationForm">

<label for="inputText">Enter Text:</label>

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

<button id="submitButton">Submit</button>

</form>

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

</div>

<script>

const form = document.getElementById("stringManipulationForm");

const inputText = document.getElementById("inputText");

const submitButton = document.getElementById("submitButton");

const output = document.getElementById("output");

form.addEventListener("submit", function (event) {

event.preventDefault();

const inputValue = inputText.value;

// Display the input string in uppercase and reversed

const upperCaseText = inputValue.toUpperCase();

const reversedText = inputValue.split("").reverse().join("");

output.textContent = `Uppercase: ${upperCaseText} | Reversed: ${reversedText}`;

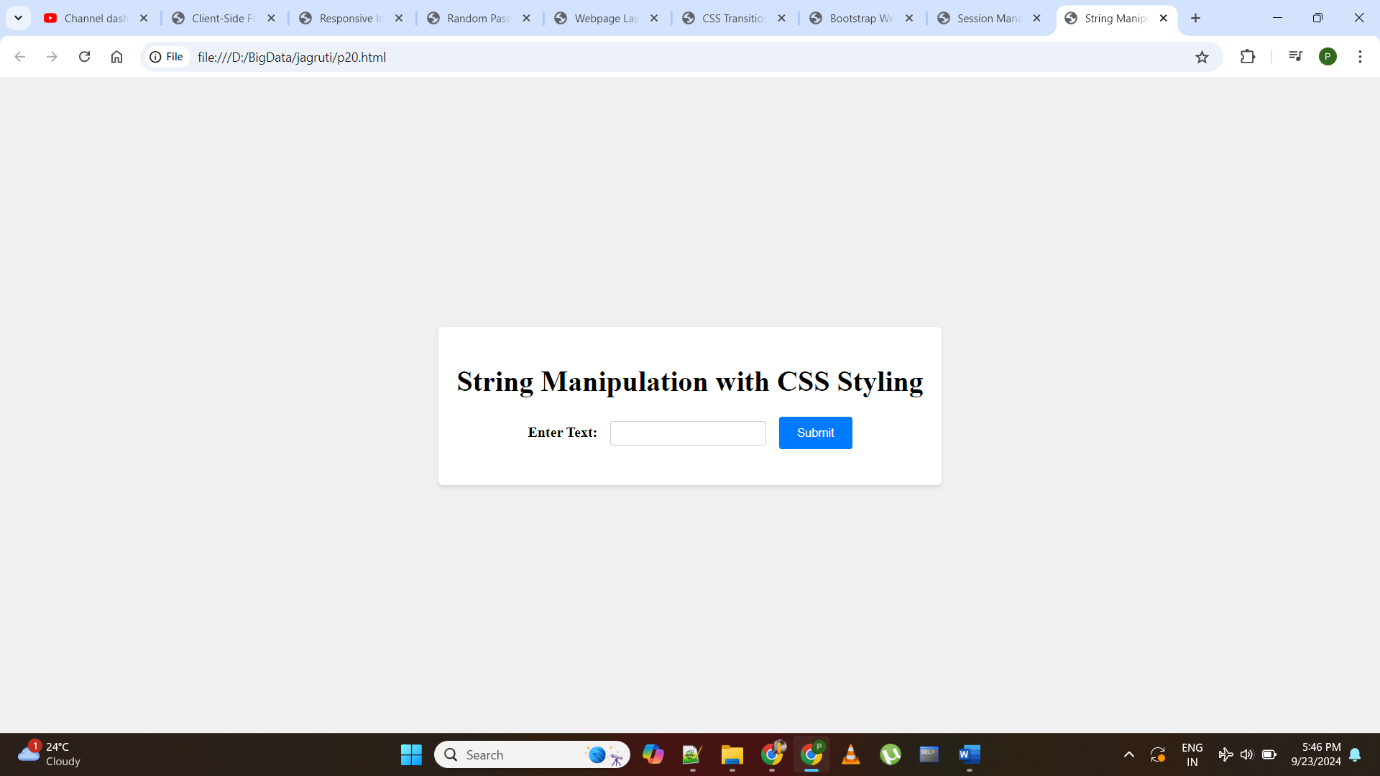
});

</script>

</body>

</html>

**Output:-**



**Program 22**

**Design a webpage layout using CSS float and positioning techniques.**

<!DOCTYPE html>

<html>

<head>

<title>Webpage Layout with CSS Float and Positioning</title>

<style>

body {

font-family: Times New Roman, Serif;

margin: 0;

padding: 0;

}

.header {

background-color: #333;

color: grey;

text-align: center;

padding: 10px;

}

.navbar {

background-color: #444;

color: grey;

overflow: hidden;

}

.navbar a {

float: left;

display: block;

color: grey;

text-align: center;

padding: 14px 16px;

text-decoration: none;

}

.navbar a:hover {

background-color: #555;

}

.content {

padding: 20px;

}

.footer {

background-color: #333;

color: grey;

text-align: center;

padding: 10px;

}

</style>

</head>

<body>

<div class="header">

<h1>Webpage Layout</h1>

</div>

<div class="navbar">

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

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

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

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

</div>

<div class="content">

<h2>Welcome to Our Website</h2>

<p>This is the main content area of the webpage.</p>

</div>

<div class="footer">

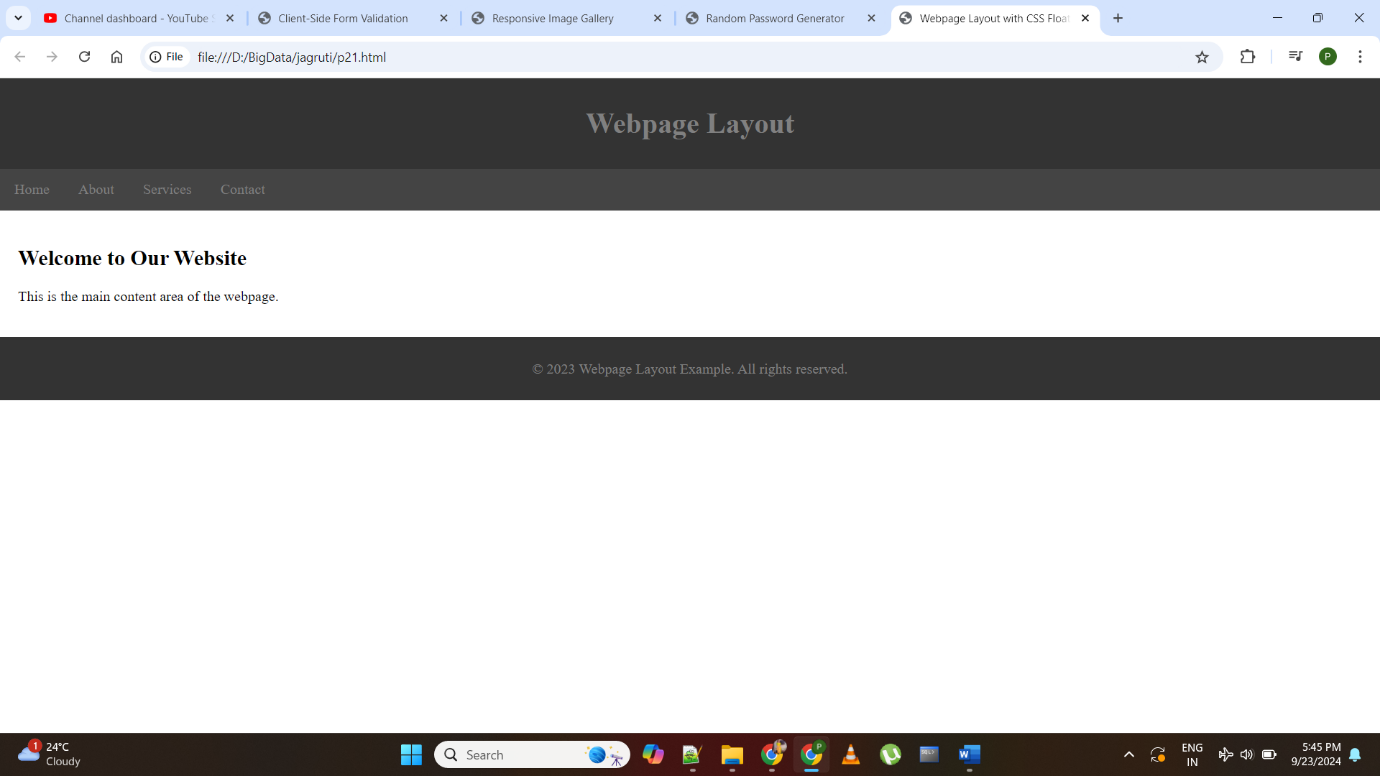
<p>&copy; 2023 Webpage Layout Example. All rights reserved.</p>

</div>

</body>

</html>

**Output:-**



**Program 23**

**Build a website that retrieves data from a web API using JavaScript and displays it dynamically**.

<!DOCTYPE html>

<html>

<head>

<title>API Data Retrieval</title>

<style>

body {

font-family: Times New Roman, Serif;

margin: 0;

padding: 0;

}

.header {

background-color: #333;

color: pink;

text-align: center;

padding: 10px;

}

.content {

padding: 20px;

}

.post {

border: 1px solid #ccc;

padding: 10px;

margin-bottom: 10px;

}

.post-title {

font-weight: bold;

}

.footer {

background-color: #333;

color: white;

text-align: center;

padding: 10px;

}

</style>

</head>

<body>

<div class="header">

<h1>API Data Retrieval</h1>

</div>

<div class="content" id="postList">

<!-- Posts will be dynamically added here -->

</div>

<div class="footer">

<p>&copy; 2023 API Data Retrieval Example. All rights reserved.</p>

</div>

<script>

const postList = document.getElementById("postList");

// Fetch data from the API

fetch("https://jsonplaceholder.typicode.com/posts")

.then(response => response.json())

.then(posts => {

posts.forEach(post => {

const postElement = document.createElement("div");

postElement.className = "post";

postElement.innerHTML = `

<h2 class="post-title">${post.title}</h2>

<p>${post.body}</p>

`;

postList.appendChild(postElement);

});

})

.catch(error => {

console.error("Error fetching data:", error);

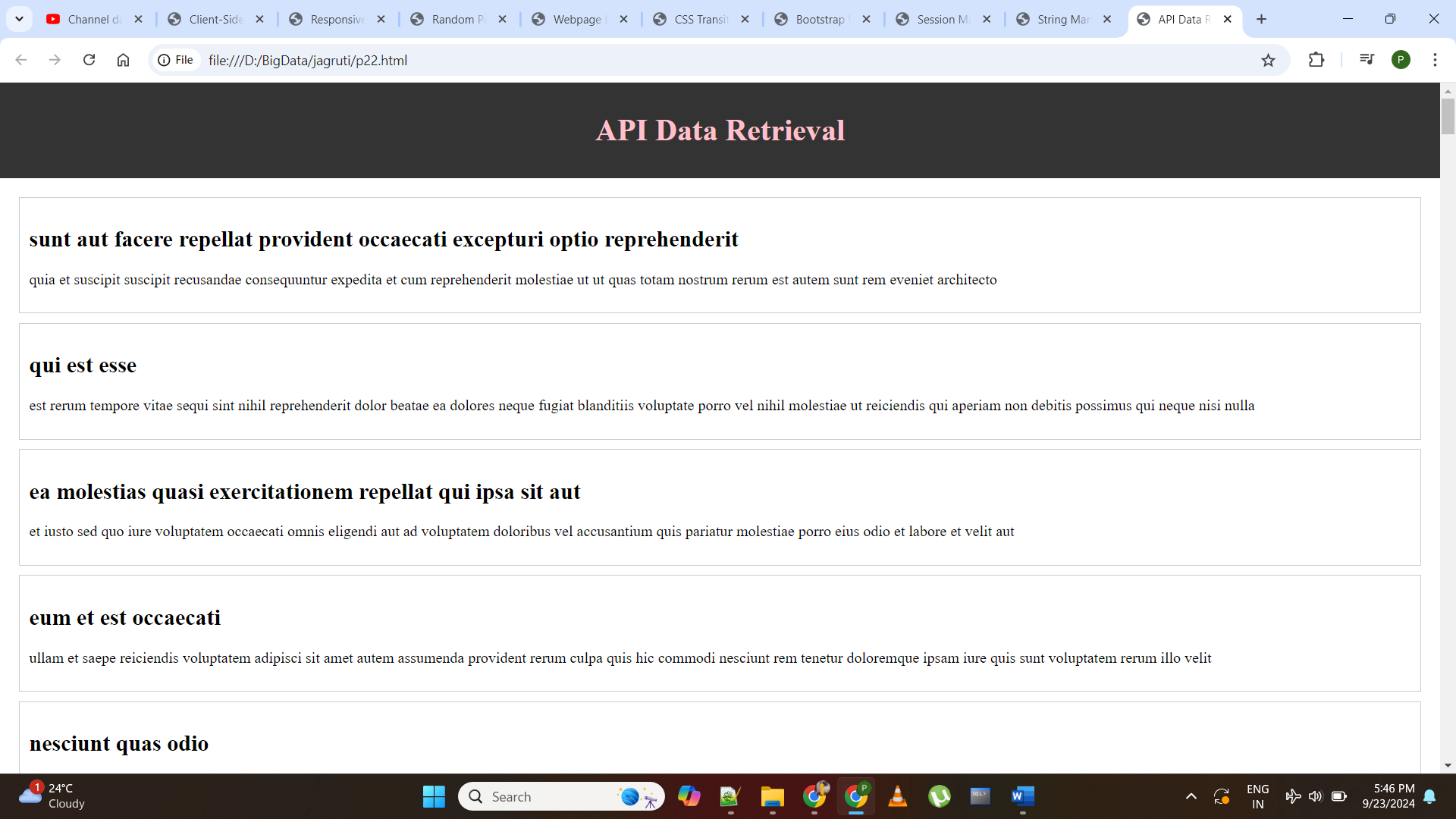
});

</script>

</body>

</html>

**Output:-**



**Program 24**

**Develop a webpage with a responsive navigation bar that collapses into a menu on a smaller screens.**

<!DOCTYPE html>

<html>

<head>

<title>Responsive Navigation Bar</title>

<style>

body {

font-family: Times New Roman, Serif;

margin: 0;

padding: 0;

}

.header {

background-color: #333;

color: pink;

text-align: center;

padding: 10px;

}

.navbar {

background-color: #444;

color: pink;

overflow: hidden;

}

.navbar a {

float: left;

display: block;

color: white;

text-align: center;

padding: 14px 16px;

text-decoration: none;

}

.navbar a:hover {

background-color: #555;

}

/\* Media query for small screens \*/

@media screen and (max-width: 600px) {

.navbar a {

float: none;

display: block;

text-align: left;

}

}

</style>

</head>

<body>

<div class="header">

<h1>Responsive Navigation Bar</h1>

</div>

<div class="navbar">

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

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

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

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

</div>

<div class="content">

<h2>Welcome to Our Website</h2>

<p>This is the main content area of the webpage.</p>

</div>

<div class="footer">

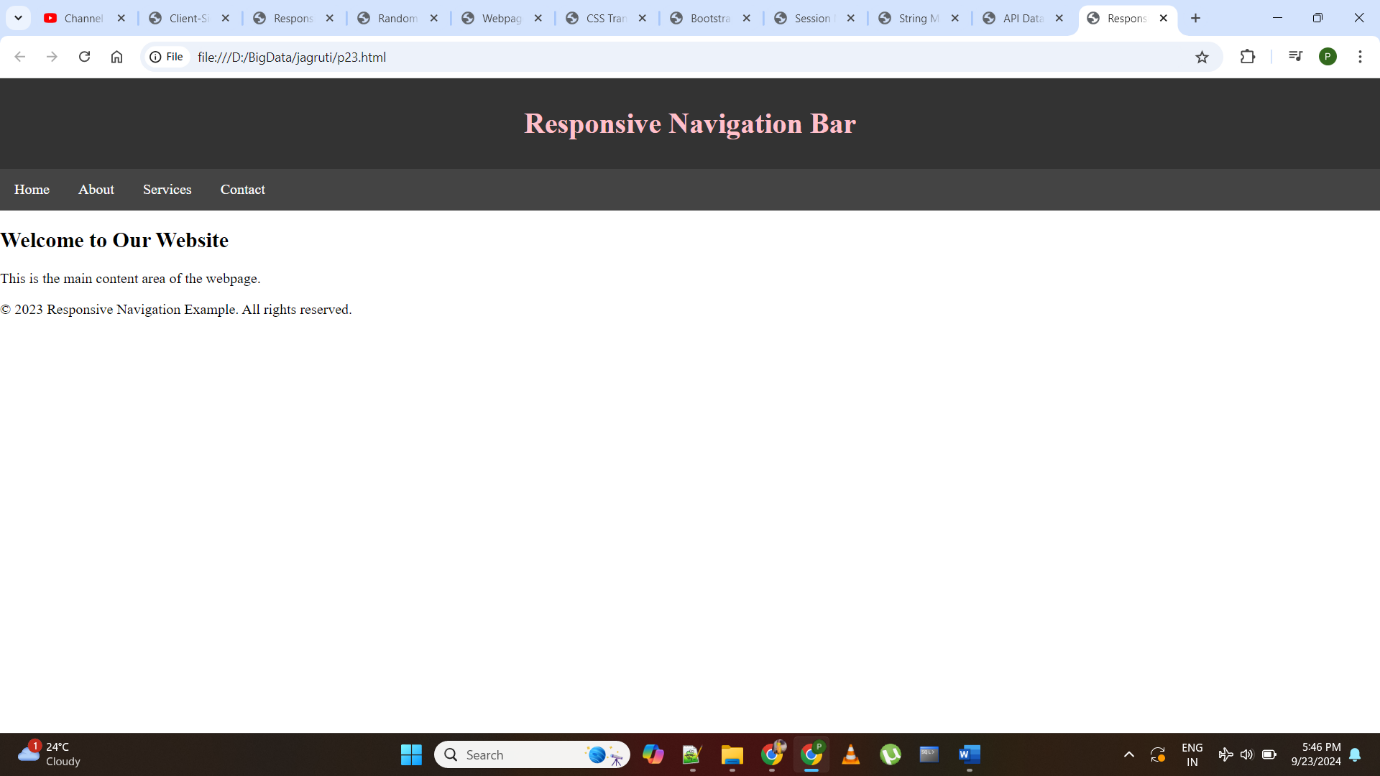
<p>&copy; 2023 Responsive Navigation Example. All rights reserved.</p>

</div>

</body>

</html>

**Output:-**



**Program 25**

**Implement CSS transforms to create visually appealing 3D effects on webpage elements.**

<!DOCTYPE html>

<html>

<head>

<title>3D Card Flip Effect</title>

<style>

body {

font-family: Times New Roman, Serif;

margin: 0;

padding: 0;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

background-color: #f0f0f0;

}

.card {

width: 200px;

height: 300px;

perspective: 1000px;

}

.card-inner {

width: 100%;

height: 100%;

transition: transform 0.5s;

transform-style: preserve-3d;

}

.card:hover .card-inner {

transform: rotateY(180deg);

}

.card-front,

.card-back {

width: 100%;

height: 100%;

position: absolute;

backface-visibility: hidden;

}

.card-front {

background-color: #007bff;

color: grey;

display: flex;

justify-content: center;

align-items: center;

}

.card-back {

background-color: #28a745;

color: grey;

display: flex;

justify-content: center;

align-items: center;

transform: rotateY(180deg);

}

</style>

</head>

<body>

<div class="card">

<div class="card-inner">

<div class="card-front">

<h2>Front</h2>

</div>

<div class="card-back">

<h2>Back</h2>

</div>

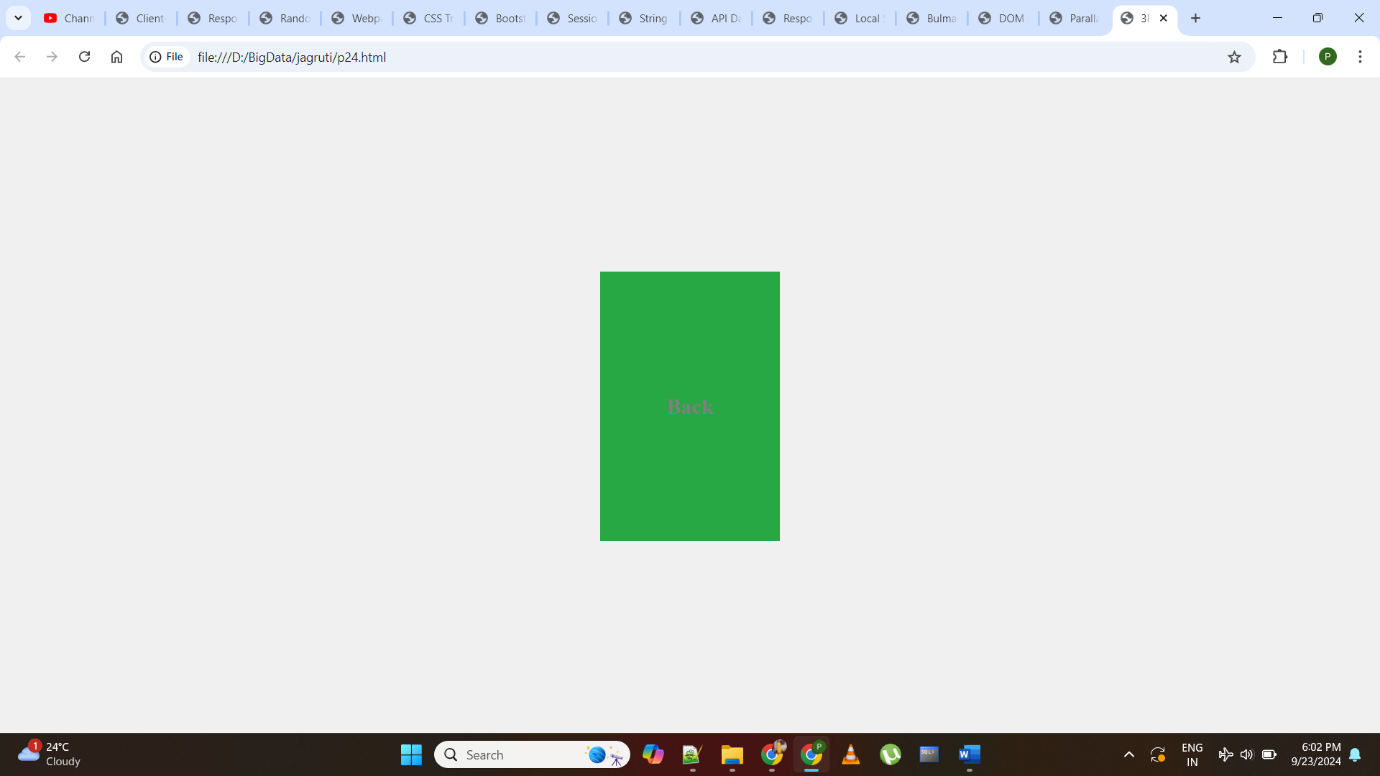
</div>

</div>

</body>

</html>

**Output:-**



**Program 26**

**Create a JavaScript program that interacts with the browsers local storage to store and retrieve data.**

<!DOCTYPE html>

<html>

<head>

<title>Local Storage Interaction</title>

<style>

body {

font-family: Times New Roman, Serif;

margin: 0;

padding: 0;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

background-color: #f0f0f0;

}

.container {

text-align: center;

padding: 20px;

border-radius: 5px;

background-color: #ffffff;

box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);

}

input[type="text"] {

padding: 5px;

border: 1px solid #ccc;

border-radius: 3px;

margin-right: 10px;

}

button {

padding: 10px 20px;

background-color: #007bff;

border: none;

border-radius: 3px;

color: white;

cursor: pointer;

}

#dataDisplay {

margin-top: 20px;

}

</style>

</head>

<body>

<div class="container">

<h1>Local Storage Interaction</h1>

<label for="dataInput">Enter Data:</label>

<input type="text" id="dataInput" placeholder="Your Data">

<button id="storeButton">Store Data</button>

<div id="dataDisplay"></div>

</div>

<script>

const dataInput = document.getElementById("dataInput");

const storeButton = document.getElementById("storeButton");

const dataDisplay = document.getElementById("dataDisplay");

// Store data in local storage

storeButton.addEventListener("click", function () {

const inputData = dataInput.value;

if (inputData.trim() !== "") {

localStorage.setItem("userInput", inputData);

dataInput.value = ""; // Clear the input field

dataDisplay.textContent = "Data stored in local storage.";

}

});

// Retrieve data from local storage

const storedData = localStorage.getItem("userInput");

if (storedData) {

dataDisplay.textContent = "Stored Data: " + storedData;

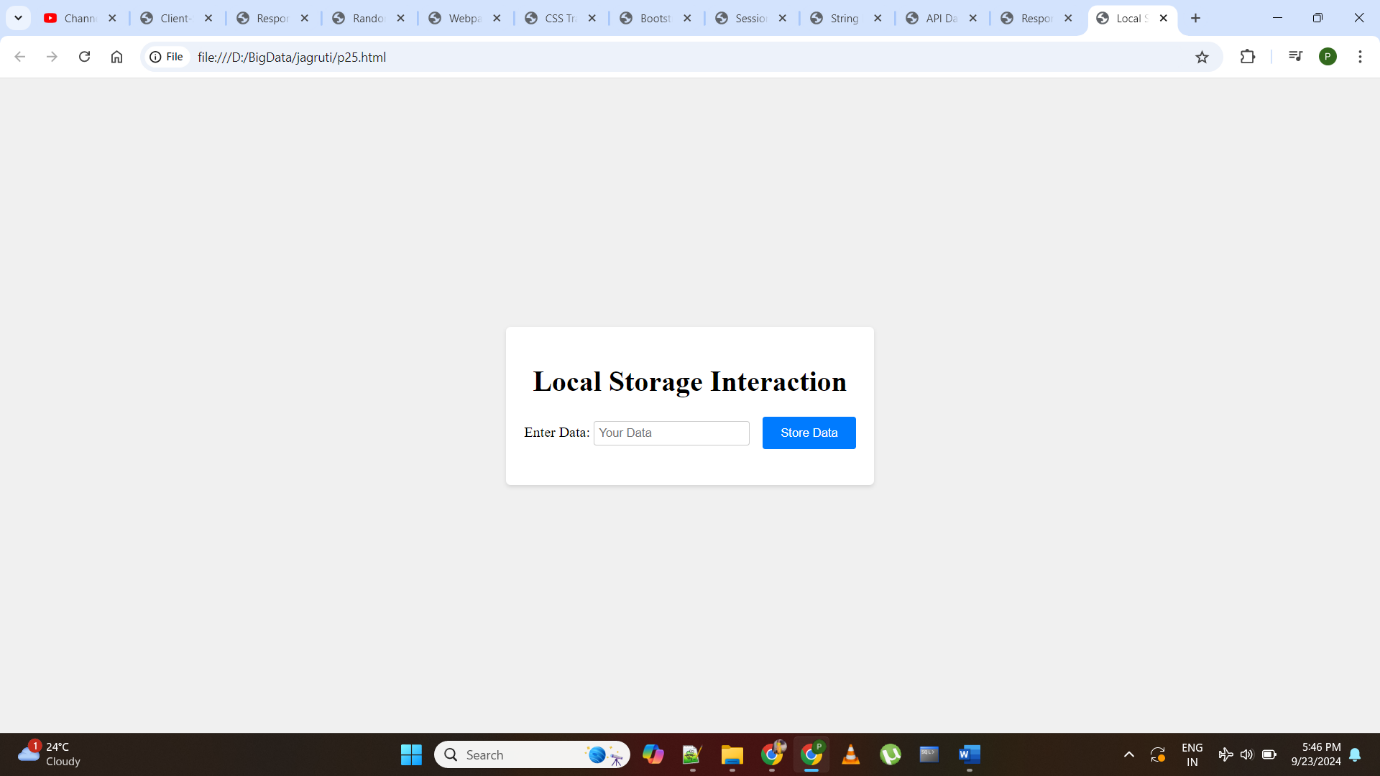
}

</script>

</body>

</html>

**Output:-**



**Program 27**

**Build a webpage that incorporates parallax scrolling using CSS and JavaScript.**

<!DOCTYPE html>

<html>

<head>

<title>Parallax Scrolling Example</title>

<style>

body {

font-family: Times New Roman, Serif;

margin: 0;

padding: 0;

background-color: #f5f5f5;

}

.parallax-container {

height: 100vh;

overflow: hidden;

position: relative;

}

.parallax-background {

position: absolute;

top: 0;

left: 0;

width: 100%;

height: 100%;

background-image: url('background.jpg');

background-size: cover;

background-position: center;

transform: translateZ(-1px) scale(2);

}

.parallax-content {

position: relative;

z-index: 1;

text-align: center;

padding: 100px 0;

color: pink;

}

</style>

</head>

<body>

<div class="parallax-container">

<div class="parallax-background"></div>

<div class="parallax-content">

<h1>Welcome to Parallax Scrolling</h1>

<p>Scroll down to see the effect</p>

</div>

</div>

<div class="content">

<h2>More Content</h2>

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>

</div>

<script>

window.addEventListener("scroll", function () {

const parallaxBackground = document.querySelector(".parallax-background");

const scrollValue = window.scrollY;

parallaxBackground.style.transform = `translateY(${scrollValue \* -0.5}px)`;

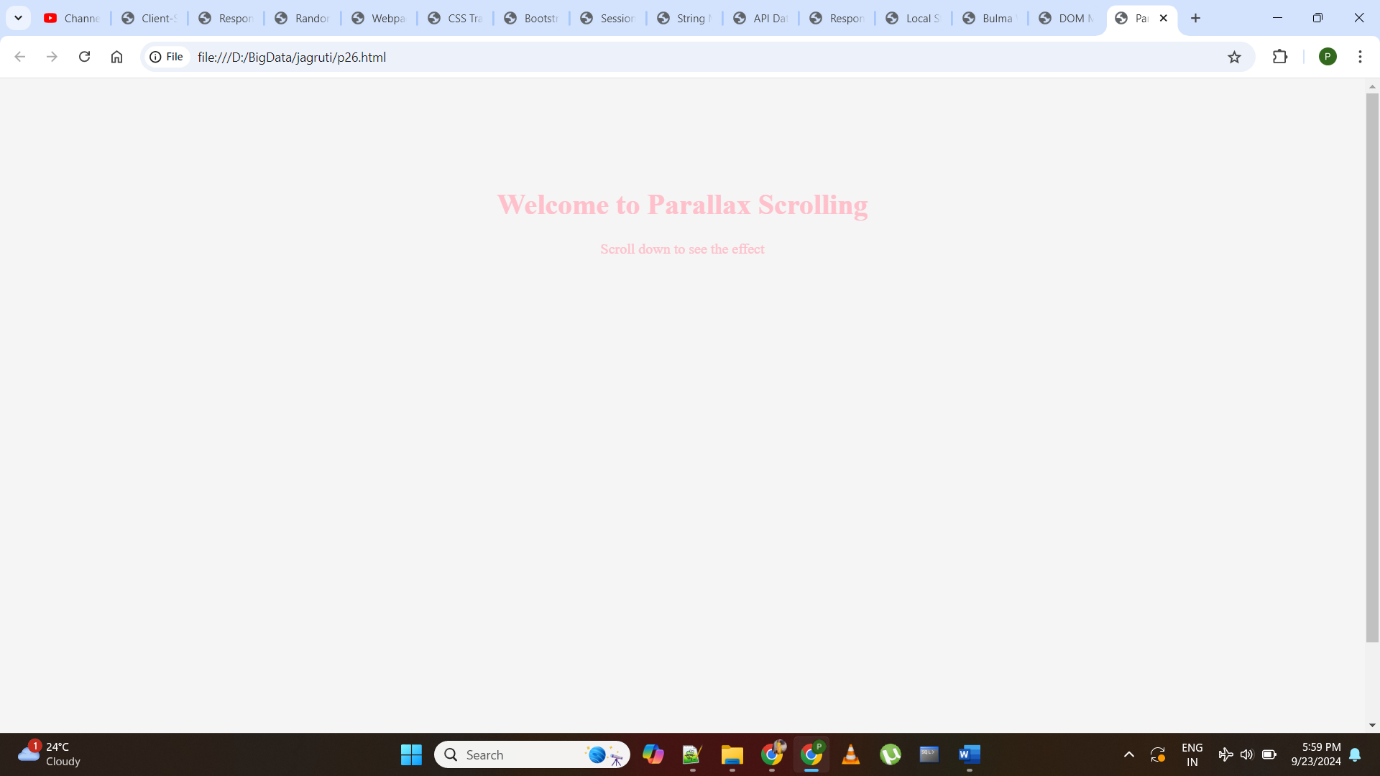
});

</script>

</body>

</html>

**Output:-**



**Program 28**

**Design a webpage layout using CSS framework such as foundation or Bulma.**

<!DOCTYPE html>

<html>

<head>

<title>Bulma Webpage Layout</title>

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/bulma/0.9.3/css/bulma.min.css">

<style>

body {

font-family: Times New Roman, Serif;

margin: 0;

padding: 0;

}

.header {

background-color: #333;

color: pink;

text-align: center;

padding: 10px;

}

.content {

padding: 20px;

}

.footer {

background-color: #333;

color: pink;

text-align: center;

padding: 10px;

}

</style>

</head>

<body>

<div class="header">

<h1 class="title">Bulma Webpage Layout</h1>

</div>

<div class="container content">

<div class="columns">

<div class="column is-one-third">

<h2 class="subtitle">Sidebar</h2>

<p>This is the sidebar content.</p>

</div>

<div class="column">

<h2 class="subtitle">Main Content</h2>

<p>This is the main content area.</p>

</div>

</div>

</div>

<div class="footer">

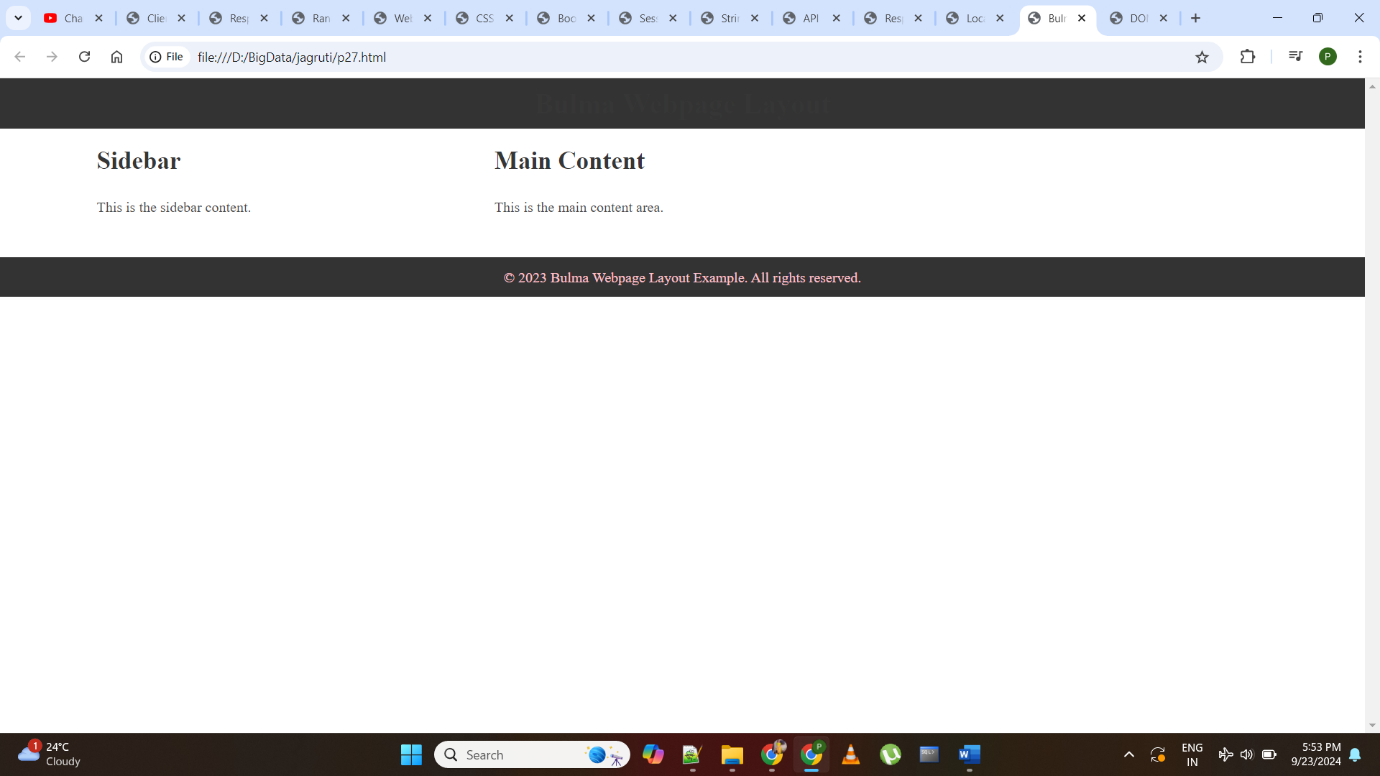
<p>&copy; 2023 Bulma Webpage Layout Example. All rights reserved.</p>

</div>

</body>

</html>

**Output:-**



**Program 29**

**Develop a JavaScript program that manipulates the Document Object Model (DOM).**

<!DOCTYPE html>

<html>

<head>

<title>DOM Manipulation Example</title>

</head>

<body>

<h1 id="pageTitle">Welcome to Our Website</h1>

<p id="introText">This is the introduction text.</p>

<button id="changeContentButton">Change Content</button>

<script>

const pageTitle = document.getElementById("pageTitle");

const introText = document.getElementById("introText");

const changeContentButton = document.getElementById("changeContentButton");

changeContentButton.addEventListener("click", function () {

// Update the content dynamically

pageTitle.textContent = "Updated Title";

introText.textContent = "Content has been updated!";

});

</script>

</body>

</html>

**Output:-**

