

**DEPARTMENT OF COMPUTER SCIENCE
RAJAGIRI COLLEGE OF SOCIAL SCIENCES
(Autonomous)
KALAMASSERY - KOCHI - 683104**



MASTER OF COMPUTER APPLICATIONS

WEB TECHNOLOGY LAB RECORD

NAME : SARATH CHANDRAN M

SEMESTER : SECOND SEMESTER

REGISTER NO. : 23204051



**DEPARTMENT OF COMPUTER SCIENCE
RAJAGIRI COLLEGE OF SOCIAL SCIENCES
(Autonomous)
KALAMASSERY - KOCHI - 683104**

MASTER OF COMPUTER APPLICATIONS

CERTIFICATE

NAME : SARATH CHANDRAN M

SEMESTER : SECOND SEMESTER

REGISTER NO. : 23204051

Certified that this is a bonafide record of work done by Sarath Chandran M in the Software Laboratory of Department of Computer Science, Rajagiri College of Social Sciences, Kalamassery.

Mr. Anas A P
Faculty in Charge

Dr. Bindya M Varghese
Dean, Computer Science

Internal Examiner

External Examiner

Place : Kalamassery
Date :

TABLE OF CONTENTS

Sl. No.	Programs	Page No.
HTML AND CSS		
1.1	Write a program using HTML and CSS, creating a horizontally scrollable container that will snap on elements when scrolling	4
1.2	Write a program to implement CSS Forms	6
1.3	How to Implement Image Zoom On Hover and disable right click on Webpage	11
1.4	How to attach audio to a webpage	14
1.5	How to embed pdf files in a webpage	16
1.6	How to implement a simple calculator using HTML	18
1.7	Write a program to create a comment box using HTML and CSS	22
1.8	Write an HTML code to implement the CSS Loader	25
1.9	Write a program to Redirect a particular section using HTML on the same page	27
JAVA SCRIPT		
2.1	Write a program to implement pop-up boxes available in JavaScript.	29
2.2	Write a JavaScript program that accepts two integers and displays the larger	31
2.3	Write a program to implement Email Validation in JavaScript	33
2.4	Write a JavaScript program to sort the items of an array	35
2.5	Demonstrate Function call in JavaScript.	37
2.6	Write a JavaScript program to test the first character of a string is uppercase or not.	39
2.7	Write a JavaScript Program to Convert Decimal to Binary.	41
2.8	Create a webpage that displays the current date and time using JavaScript	43
2.9	Implement a JavaScript program to change the background color randomly.	45
2.10	Write a JavaScript program to construct the pyramid pattern, using a nested for loop.	48
2.11	Write a JavaScript Program to Solve Quadratic Equations.	50
2.12	Write a JavaScript function that accepts a string as a parameter and counts the number of vowels within the string.	53
2.13	Write a program to Toggle Password Visibility in JavaScript.	56

JSON

3.1	Write a program to demonstrate the Fibonacci sequence in JSON.	58
3.2	Write a program to check leap years.	61

PHP

4.1	Write a PHP program to check whether a given number is Armstrong or not.	63
4.2	Write a PHP program to illustrate Cookies.	66
4.3	Check whether a year is a leap year or not by inserting a year in PHP.	68
4.4	Write a program to Implement static properties in PHP	70
4.5	Write a PHP program to use the global keyword to access a global variable from within a function.	73
4.6	Write a PHP program to convert a decimal number into a binary number	75
4.7	Write a PHP program to store the current date/time in a cookie and display the 'last visited on' date and time on the web page upon reopening the same page	78
4.8	Write a PHP program to create an array and store 10 names in the array. Do the following operations. 1. Display the contents using each statement. 2. Display the array in sorted order. 3. Display the array without the duplicate elements. 4. Remove the last element and display. 5. Display the array in reverse order. 6. Search for an element in the given array.	80
4.9	Write the PHP script to send an email.	83
4.10	Write a PHP program to implement MySQL databases with CRUD operations.	86

LARAVEL FRAMEWORK

5.1	Program to illustrate the workings of a simple Laravel web application.	91
5.2	Program to demonstrate the implementation of the Bootstrap framework in Laravel.	93
5.3	Program to illustrate the creation of a Laravel web application with all form validations.	96
5.4	Create a Login page using the Laravel framework.	100
5.5	Program to implement session handling using the Laravel framework.	108

5.6	Write a program to execute database CRUD operations in the Laravel framework.	114
5.7	Create a simple Laravel Web Application, including: 1. A registration form with all form validations. 2. Perform CRUD operations using migrations.	123
5.8	Create a Web Application for Blog Creation with a Login and Registration form and perform the following operations on blogs using the Laravel framework. 1. CREATE Blogs. 2. VIEW Blogs in a Table. 3. EDIT and UPDATE Blogs. 4. DELETE Blogs.	139
5.9	Create a simple LARAVEL web application displaying the CAE2 Marks of MSc CS students, including: 1. Student Login and Teacher login 2. Teachers can Add, View, Update, and Delete Students and Marks. 3. Students can VIEW their marks	151
5.10	Create a simple Laravel web application for students, including: 1. Student Registration Form 2. Form Validations. 3. Login for students with session handling. 4. Minimum of two web pages with a standard layout.	164
6	PORTFOLIO	
7	PROJECT	

...

Write a program using HTML and CSS, creating a horizontally scrollable container that will snap on elements when scrolling.

CODE :

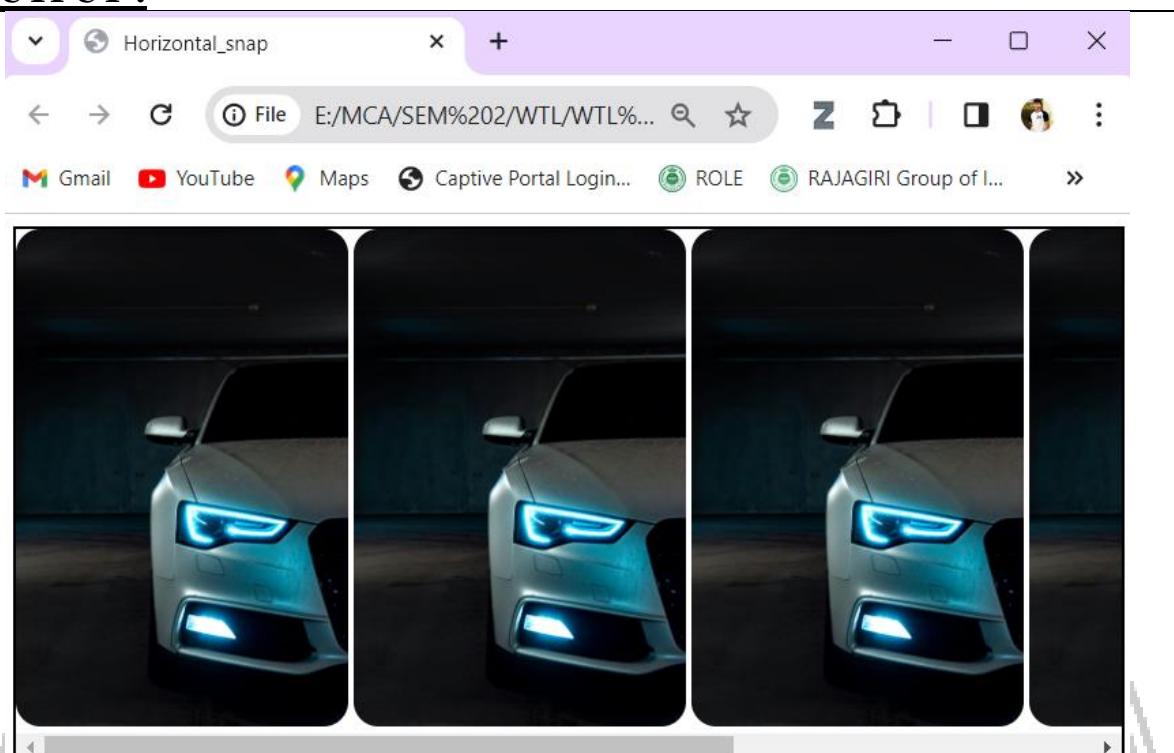
```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<style>
.horizontal-snap {
    margin: auto;
    border: solid black 2px;
    overflow-x: scroll;
    overflow-y: hidden;
    white-space: nowrap;
    scroll-snap-type: x mandatory;
}
.horizontal-snap>a {
    scroll-snap-align: center;
}
.horizontal-snap img {
    width: 30%;
    object-fit: contain;
    border-radius: 1rem;
}
</style>
<title>Horizontal_snap</title>
</head>
<body>
<div class="horizontal-snap">





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

OUTPUT :



Program 1.2

Date: 01/01/2024

Write a program to implement CSS Forms.

CODE :

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>CSS Form</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f4f4f4;
            margin: 0;
            padding: 0;
        }
        .container {
            max-width: 500px;
            margin: 50px auto;
            padding: 20px;
            background-color: #fff;
            border-radius: 5px;
            box-shadow: 0 0 10px rgba(0,0,0,0.1);
        }
        .form-group {
            margin-bottom: 20px;
        }
        label {
            /* display: block; */
            font-weight: bold;
            margin-bottom: 5px;
        }
        input[type="text"],
        input[type="email"],
        input[type="password"],
        textarea,
        select {
```

```
width: 100%;  
padding: 10px;  
border: 1px solid #ccc;  
border-radius: 5px;  
box-sizing: border-box;  
margin-top: 5px;  
margin-bottom: 10px;  
}  
input[type="radio"],  
input[type="checkbox"] {  
    margin-right: 5px;  
    display: inline-block;  
}  
input[type="submit"] {  
    background-color: #4caf50;  
    color: #fff;  
    border: none;  
    padding: 15px 30px;  
    border-radius: 5px;  
    cursor: pointer;  
    display: block;  
    margin-top: 20px;  
    font-weight: bold;  
}  
input[type="submit"]:hover {  
    background-color: #45a049;  
}  
/* Style the file input */  
input[type="file"] {  
    display: none;  
}  
.file-upload {  
    display: inline-block;  
    background-color: #ff0000;  
    color: #fff;  
    padding: 8px 16px;  
    border-radius: 5px;  
    cursor: pointer;  
}  
.file-upload:hover {  
    background-color: #cc0000;  
}  
.file-upload input[type="file"] {
```

```

        display: none;
    }

```

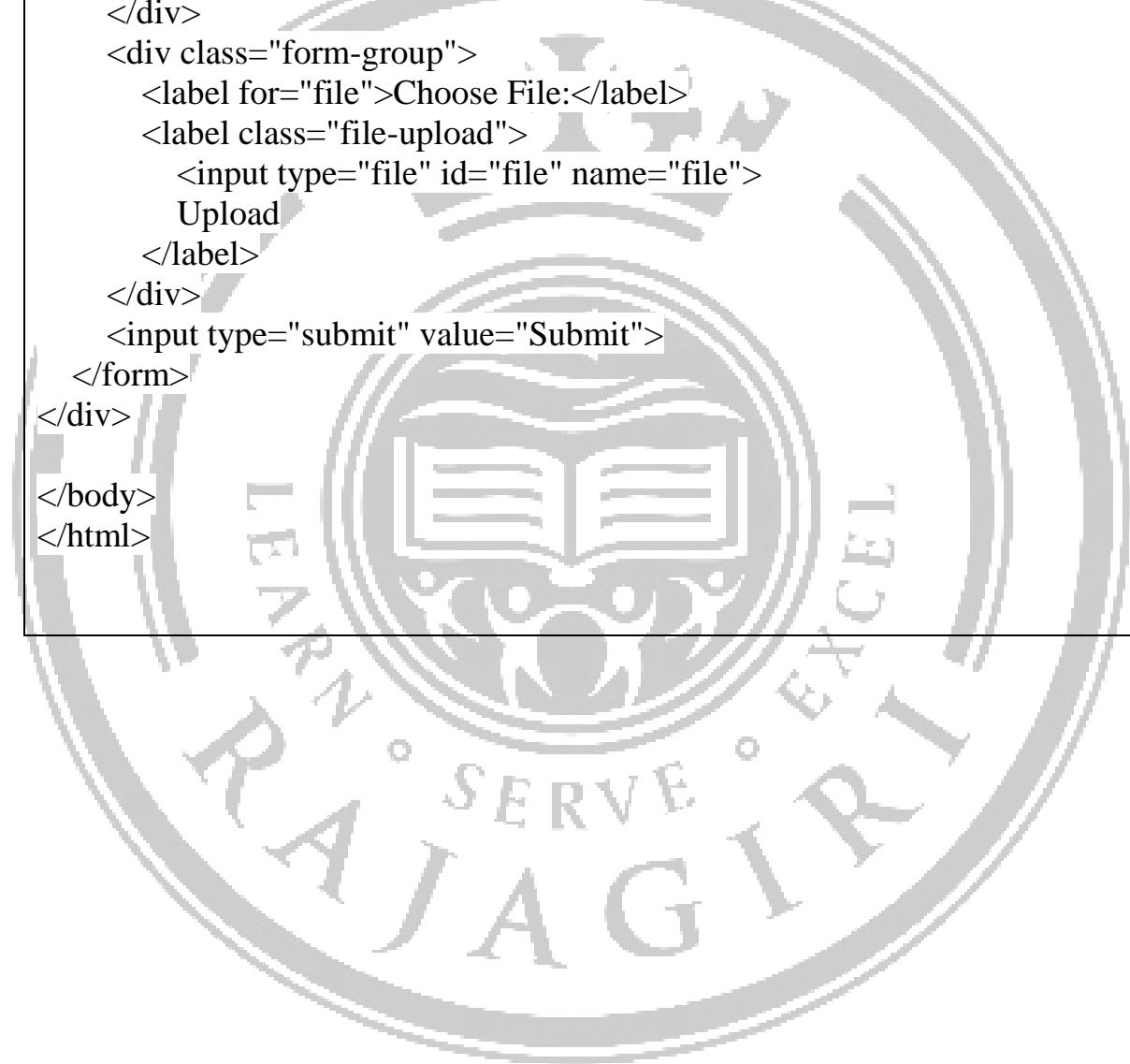
```

</head>
<body>
```

```

<div class="container">
    <h2 style="text-align: center;">Sign Up Form</h2>
    <form>
        <div class="form-group">
            <label for="username">Username:</label>
            <input type="text" id="username" name="username" required>
        </div>
        <div class="form-group">
            <label for="email">Email:</label>
            <input type="email" id="email" name="email" required>
        </div>
        <div class="form-group">
            <label for="password">Password:</label>
            <input type="password" id="password" name="password" required>
        </div>
        <div class="form-group">
            <label for="message">Message:</label>
            <textarea id="message" name="message" rows="4"></textarea>
        </div>
        <div class="form-group">
            <label for="gender">Gender:</label>
            <input type="radio" id="male" name="gender" value="male">
            <label for="male">Male</label>
            <input type="radio" id="female" name="gender" value="female">
            <label for="female">Female</label>
        </div>
        <div class="form-group">
            <label for="interests">Interests:</label>
            <input type="checkbox" id="coding" name="interests" value="coding">
            <label for="coding">Coding</label>
            <input type="checkbox" id="reading" name="interests" value="reading">
            <label for="reading">Reading</label>
            <input type="checkbox" id="sports" name="interests" value="sports">
            <label for="sports">Sports</label>
        </div>
    </form>
</div>
```

```
</div>
<div class="form-group">
    <label for="country">Country:</label>
    <select id="country" name="country">
        <option value="india">INDIA</option>
        <option value="uk">UK</option>
        <option value="canada">Canada</option>
        <option value="australia">Australia</option>
    </select>
</div>
<div class="form-group">
    <label for="file">Choose File:</label>
    <label class="file-upload">
        <input type="file" id="file" name="file">
        Upload
    </label>
</div>
<input type="submit" value="Submit">
</form>
</div>
</body>
</html>
```



OUTPUT :

The screenshot displays a web browser window with a purple header bar. The title bar reads "CSS Form". The address bar shows the URL "E:/MCA/SEM%202/WTL/WTL%20LAB%20PRO...". Below the address bar, there are links for "Gmail", "YouTube", "Maps", "Captive Portal Login...", "ROLE", and "RAJAGIRI Group of I...". The main content area features a "Sign Up Form" modal. The modal has a title "Sign Up Form". It contains the following fields:

- Username: An input field.
- Email: An input field.
- Password: An input field.
- Message: A text area.
- Gender: Radio buttons for "Male" and "Female".
- Interests: Checkboxes for "Coding", "Reading", and "Sports".
- Country: A dropdown menu set to "INDIA".
- Choose File: A button with "Upload" text.
- Submit: A green button.

How to Implement Image Zoom On Hover and disable right click on Webpage?**CODE :**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Image Zoom and Right Click Disable</title>
    <style>
        .container {
            text-align: center;
        }

        .zoom-image {
            width: 300px; /* Adjust image size as needed */
            transition: transform 0.5s;
        }

        .zoom-image:hover {
            transform: scale(1.5); /* Adjust the scale factor for zoom effect */
            cursor: zoom-in;
        }
    </style>
</head>
<body oncontextmenu="return false;">
    <div class="container">
        
        
        
    </div>

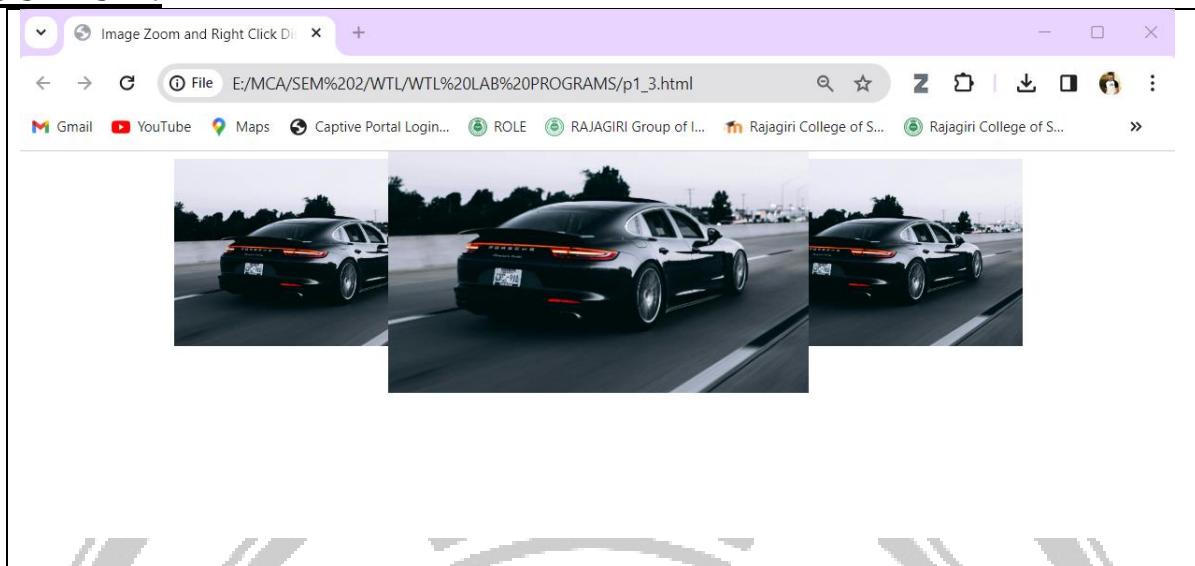
    <!-- To disable the default right-click behavior on the webpage. -->
    <script>
        document.addEventListener('contextmenu', function(e) {
            e.preventDefault();
       });
    </script>

```

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



OUTPUT :

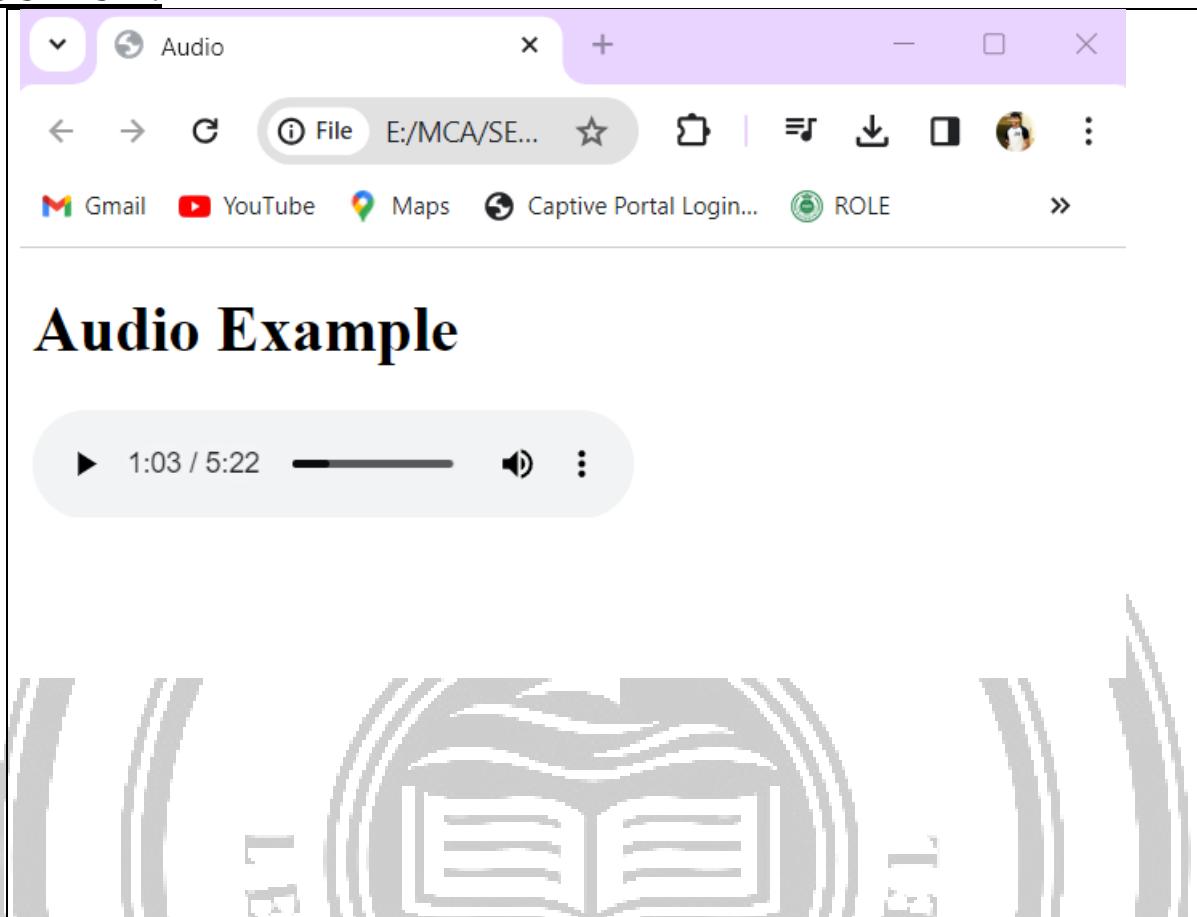


How to attach audio to a webpage?

CODE :

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Audio</title>
</head>
<body>
    <h1>Audio Example</h1>
    <audio controls>
        <source src="p1_4.mp3" type="audio/mpeg">
        Your browser does not support the audio element.
    </audio>
</body>
</html>
```

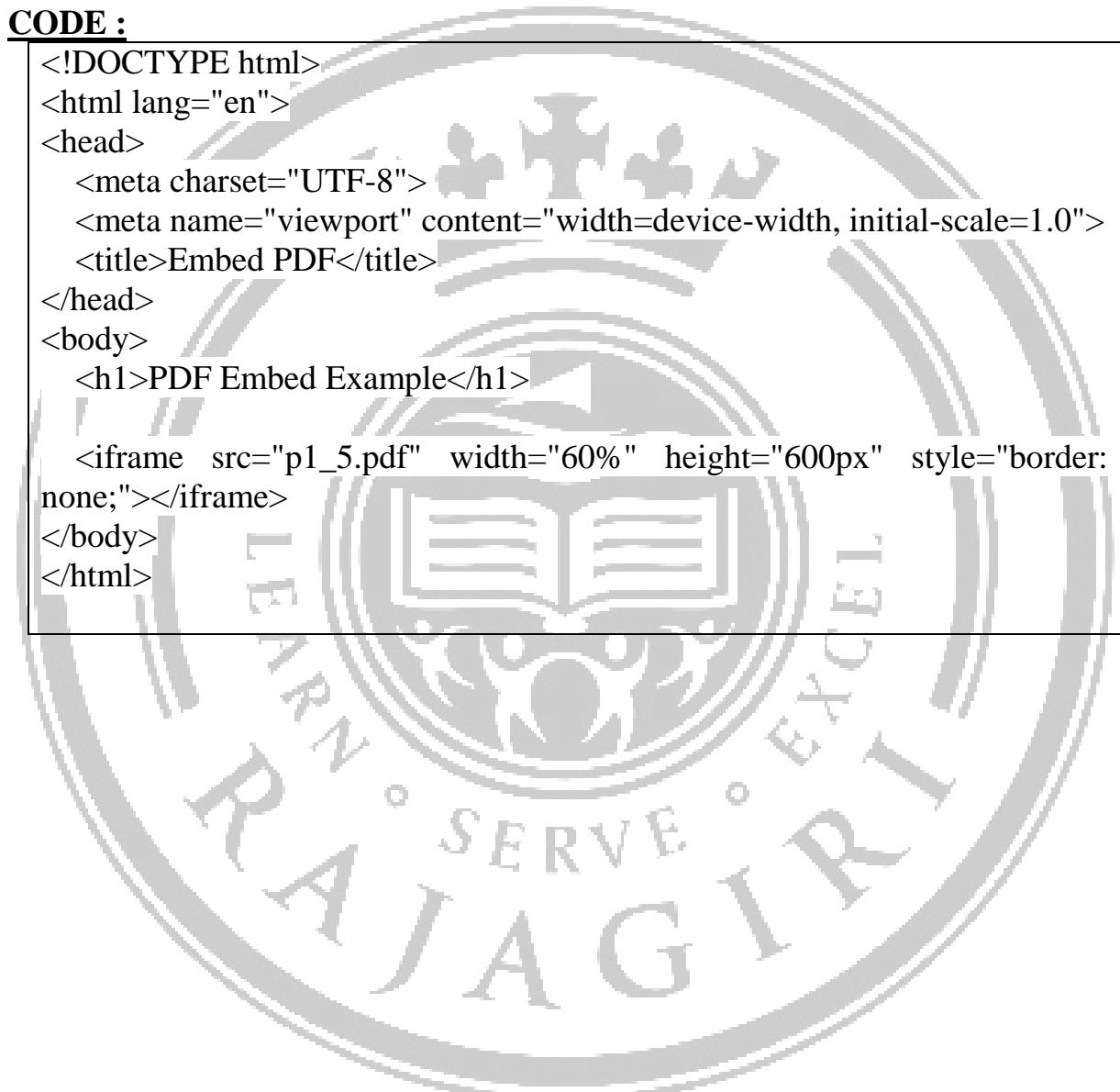
OUTPUT :



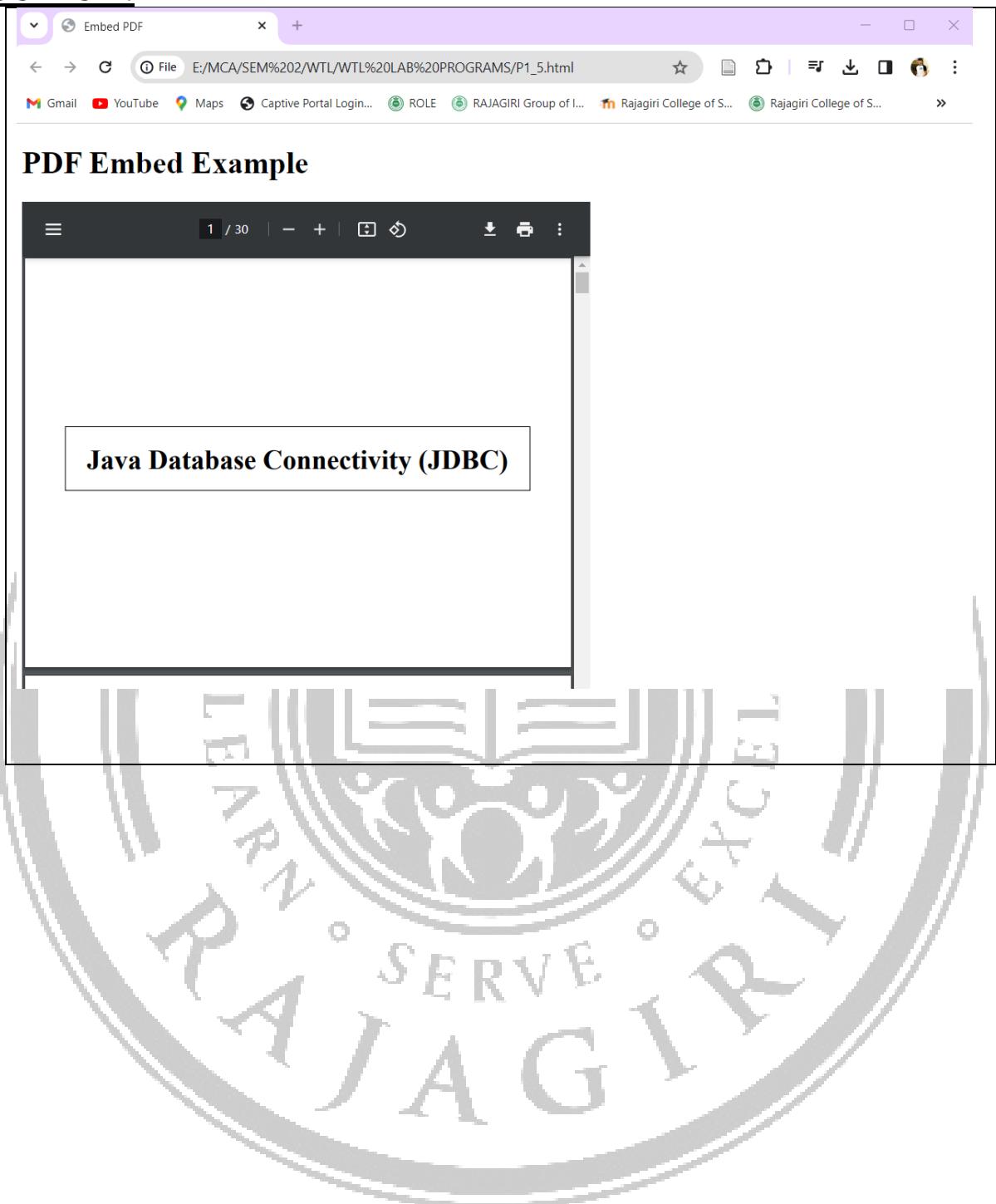
How to embed pdf files in a webpage?

CODE :

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Embed PDF</title>
</head>
<body>
  <h1>PDF Embed Example</h1>
  <iframe src="p1_5.pdf" width="60%" height="600px" style="border: none;"></iframe>
</body>
</html>
```



OUTPUT :



How to implement a simple calculator using HTML?**CODE :**

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

<head>
  <meta charset="utf-8">
  <title>Simple Calculator</title>
<style>
  @import
  url('https://fonts.googleapis.com/css2?family=Orbitron&display=swap');

  .calculator {
    padding: 10px;
    border-radius: 1em;
    height: 380px;
    width: 400px;
    margin: auto;
    background-color: #191b28;
    box-shadow: rgba(0, 0, 0, 0.19) 0px 10px 20px, rgba(0, 0, 0, 0.23) 0px
6px 6px;
  }

  .display-box {
    font-family: 'Orbitron', sans-serif;
    background-color: #dcdbe1;
    border: solid black 0.5px;
    color: black;
    border-radius: 5px;
    width: 100%;
    height: 65%;
  }

  #btn {
```

```

background-color: #fb0066;
}

input[type=button] {
    font-family: 'Orbitron', sans-serif;
    background-color: #64278f;
    color: white;
    border: solid black 0.5px;
    width: 100%;
    border-radius: 5px;
    height: 70%;
    outline: none;
}

input:active[type=button] {
    background: #e5e5e5;
    -webkit-box-shadow: inset 0px 0px 5px #c1c1c1;
    -moz-box-shadow: inset 0px 0px 5px #c1c1c1;
    box-shadow: inset 0px 0px 5px #c1c1c1;
}

</style>
</head>
<body>

<table class="calculator" >
<tr>
    <td colspan="3"> <input class="display-box" type="text" id="result" disabled /></td>
    <!-- clearScreen() function clears all the values -->
    <td> <input type="button" value="C" onclick="clearScreen()" id="btn" />
</td>
</tr>
<tr>
    <!-- display() function displays the value of clicked button -->
    <td> <input type="button" value="1" onclick="display('1')"/> </td>
    <td> <input type="button" value="2" onclick="display('2')"/> </td>
    <td> <input type="button" value="3" onclick="display('3')"/> </td>
    <td> <input type="button" value="/" onclick="display('/')"/> </td>
</tr>
<tr>
    <td> <input type="button" value="4" onclick="display('4')"/> </td>

```

```

<td> <input type="button" value="5" onclick="display('5')" /> </td>
<td> <input type="button" value="6" onclick="display('6')" /> </td>
<td> <input type="button" value="-" onclick="display('-')" /> </td>
</tr>
<tr>
<td> <input type="button" value="7" onclick="display('7')" /> </td>
<td> <input type="button" value="8" onclick="display('8')" /> </td>
<td> <input type="button" value="9" onclick="display('9')" /> </td>
<td> <input type="button" value)+" onclick="display('+')" /> </td>
</tr>
<tr>
<td> <input type="button" value="." onclick="display('.')" /> </td>
<td> <input type="button" value="0" onclick="display('0')" /> </td>

<!-- calculate() function evaluates the mathematical expression --&gt;
&lt;td&gt; &lt;input type="button" value="=" onclick="calculate()" id="btn" /&gt;
&lt;/td&gt;
&lt;td&gt; &lt;input type="button" value="*" onclick="display('*')" /&gt; &lt;/td&gt;
&lt;/tr&gt;
&lt;/table&gt;

&lt;script&gt;
    // This function clears all the values
    function clearScreen() {
        document.getElementById("result").value = "";
    }

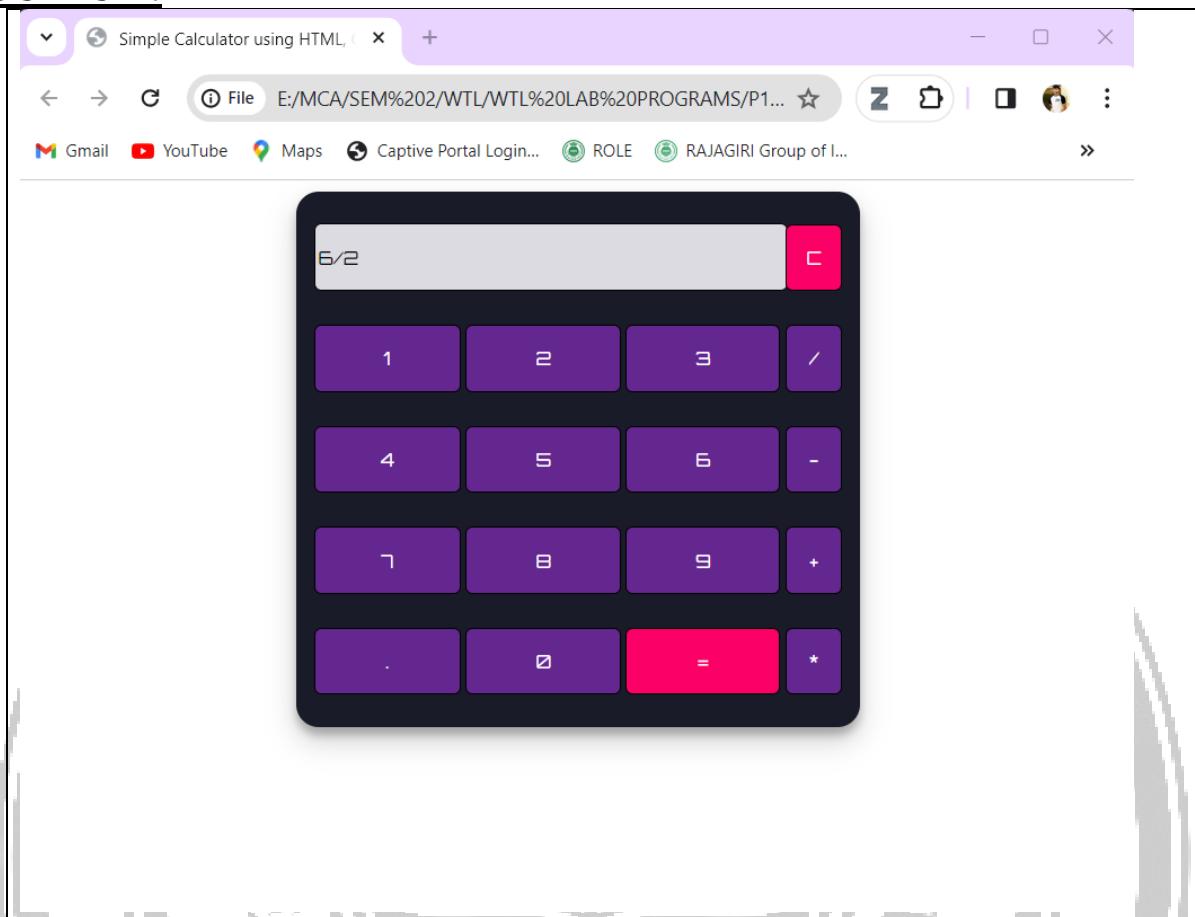
    // This function displays the values
    function display(value) {
        document.getElementById("result").value += value;
    }

    // This function evaluates the expression and returns the result
    function calculate() {
        var p = document.getElementById("result").value;
        var q = eval(p);
        document.getElementById("result").value = q;
    }
&lt;/script&gt;
&lt;/body&gt;

&lt;/html&gt;
</pre>

```

OUTPUT :



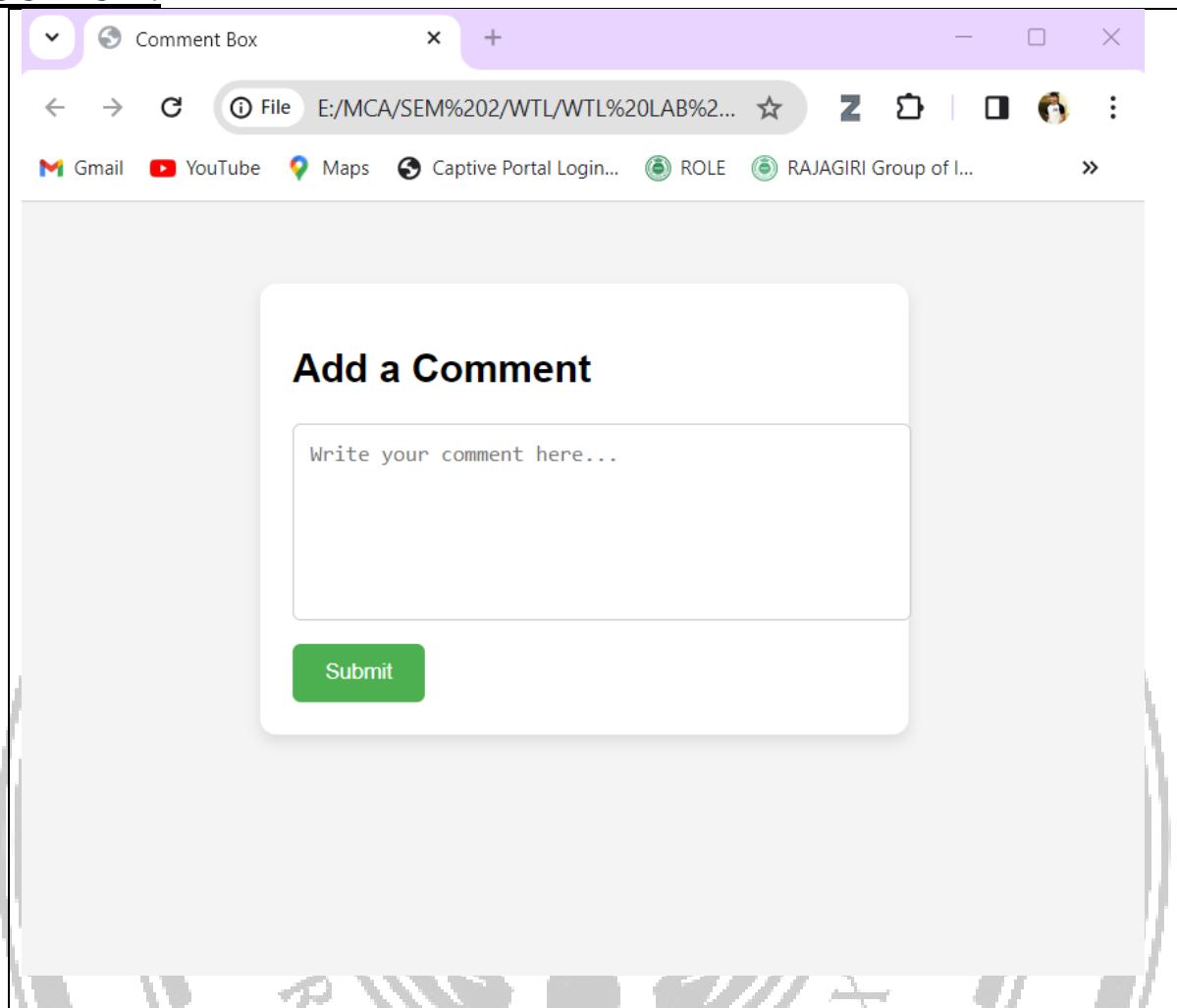
Write a program to create a comment box using HTML and CSS.

CODE :

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Comment Box</title>
<style>
body {
    font-family: Arial, sans-serif;
    background-color: #f4f4f4;
    margin: 0;
    padding: 0;
}
.comment-box {
    width: 400px;
    margin: 50px auto;
    background-color: #fff;
    border-radius: 10px;
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
    padding: 20px;
    box-sizing: border-box;
}
.comment-box textarea {
    width: 100%;
    height: 100px;
    padding: 10px;
    margin-bottom: 10px;
    border: 1px solid #ccc;
    border-radius: 5px;
    resize: none;
}
.comment-box input[type="submit"] {
```

```
background-color: #4caf50;
color: #fff;
border: none;
border-radius: 5px;
padding: 10px 20px;
cursor: pointer;
transition: background-color 0.3s;
}
.comment-box input[type="submit"]:hover {
background-color: #45a049;
}
</style>
</head>
<body>
<div class="comment-box">
<h2>Add a Comment</h2>
<form action="">
<textarea name="comment" id="comment" placeholder="Write your
comment here..." required></textarea>
<br>
<input type="submit" value="Submit">
</form>
</div>
</body>
</html>
```

OUTPUT :



Program 1.8

Date: 08/01/2024

Write an HTML code to implement the CSS Loader.

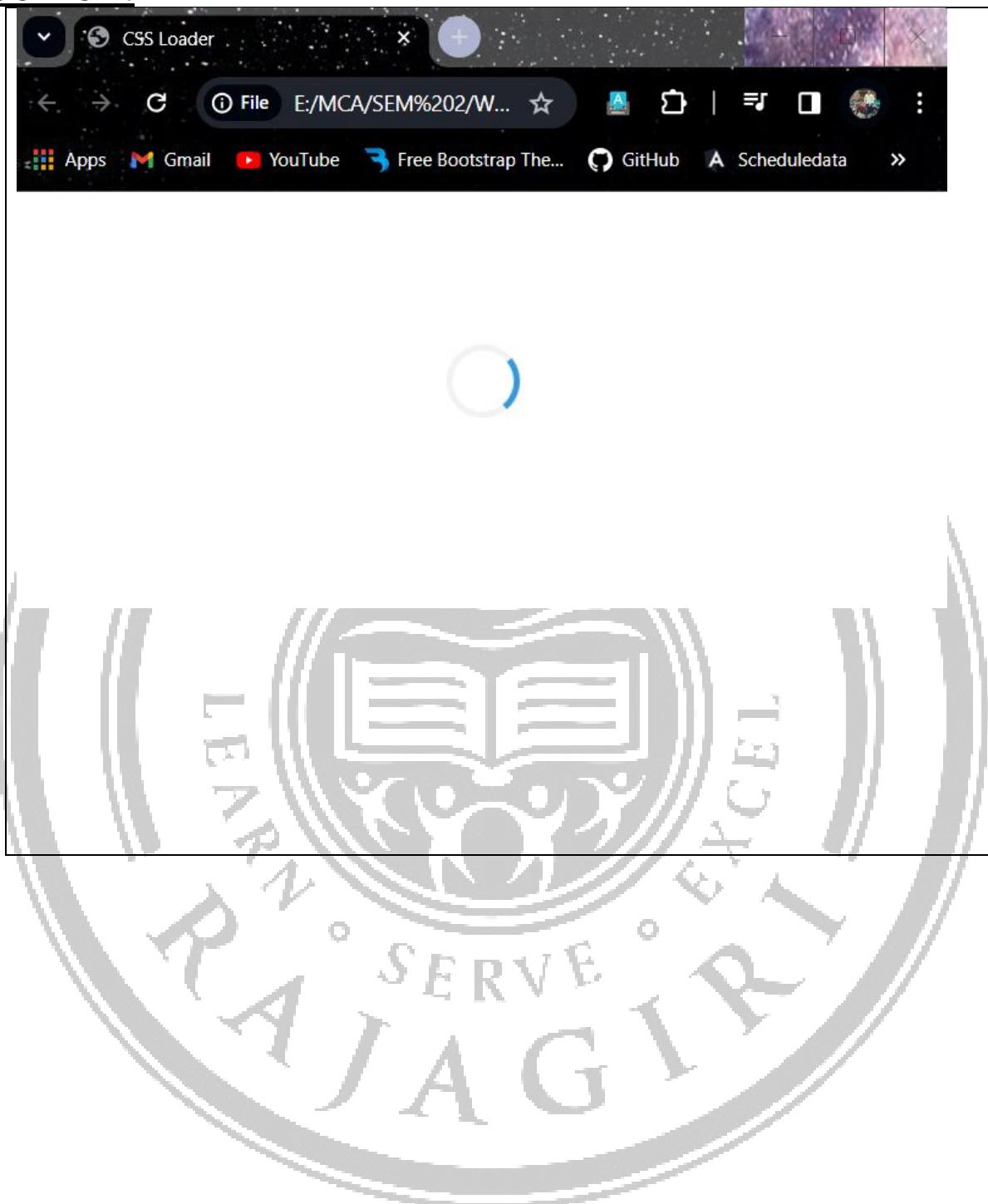
CODE :

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>CSS Loader</title>
<style>
.loader {
    border: 4px solid #f3f3f3;
    border-top: 4px solid #3498db;
    border-radius: 50%;
    width: 40px;
    height: 40px;
    animation: spin 1s linear infinite;
    margin: 100px auto;
}
@keyframes spin {
    0% { transform: rotate(0deg); }
    100% { transform: rotate(360deg); }
}
</style>
</head>
<body>

<div class="loader"></div>

</body>
</html>
```

OUTPUT :



Write a program to Redirect a particular section using HTML on the same page.

CODE :

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Redirect to Section</title>
</head>
<body>

    <h1>Welcome to My Website</h1>

    <p>This is the homepage.</p>

    <h2 id="about">About Us</h2>

    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer nec odio. Praesent libero. Sed cursus ante dapibus diam. Sed nisi.</p>

    <h2 id="contact">Contact Us</h2>

    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer nec odio. Praesent libero. Sed cursus ante dapibus diam. Sed nisi.</p>

    <p>You can <a href="#about">click here</a> to go to the About Us section.</p>
    <p>You can <a href="#contact">click here</a> to go to the Contact Us section.</p>

</body>
</html>
```

OUTPUT :

Redirect to Section

E:/MCA/SEM%202/WTL/WTL%20LA...

Gmail YouTube Maps Captive Portal Login... ROLE RAJAGIRI Group of I...

Welcome to My Website

This is the homepage.

About Us

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer nec odio. Praesent libero. Sed cursus ante dapibus diam. Sed nisi.

Contact Us

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer nec odio. Praesent libero. Sed cursus ante dapibus diam. Sed nisi.

You can [click here](#) to go to the About Us section.

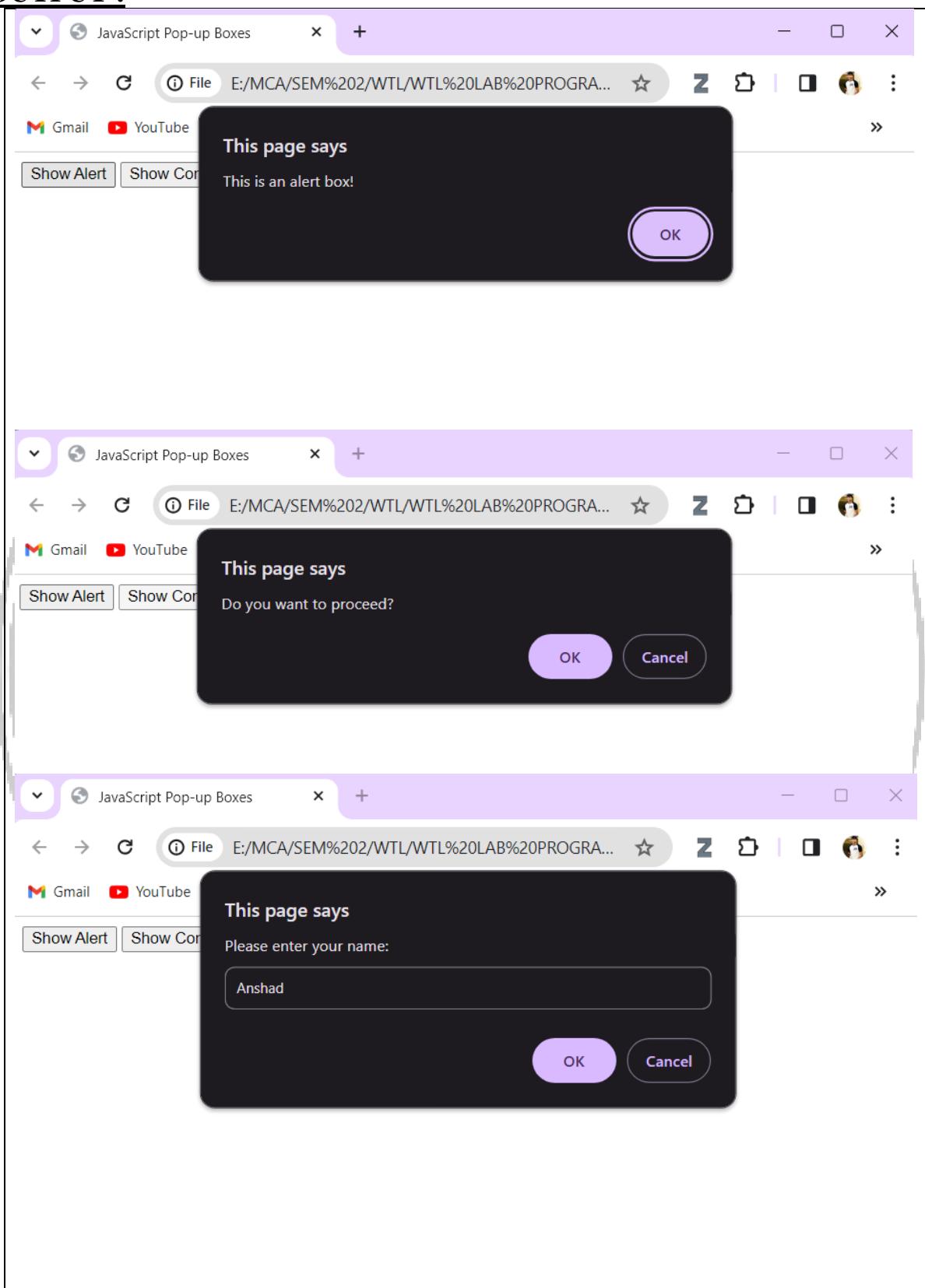
You can [click here](#) to go to the Contact Us section.

Write a program to implement pop-up boxes available in JavaScript.

CODE :

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>JavaScript Pop-up Boxes</title>
    <script>
        function showAlert() {
            alert("This is an alert box!");
        }
        function showConfirm() {
            var result = confirm("Do you want to proceed?");
            if (result) {
                alert("You clicked OK!");
            } else {
                alert("You clicked Cancel!");
            }
        }
        function showPrompt() {
            var name = prompt("Please enter your name:");
            if (name !== null && name !== "") {
                alert("Hello, " + name + "!");
            } else {
                alert("You did not enter your name.");
            }
        }
    </script>
</head>
<body>
    <button onclick="showAlert()">Show Alert</button>
    <button onclick="showConfirm()">Show Confirm</button>
    <button onclick="showPrompt()">Show Prompt</button>
</body>
</html>
```

OUTPUT :





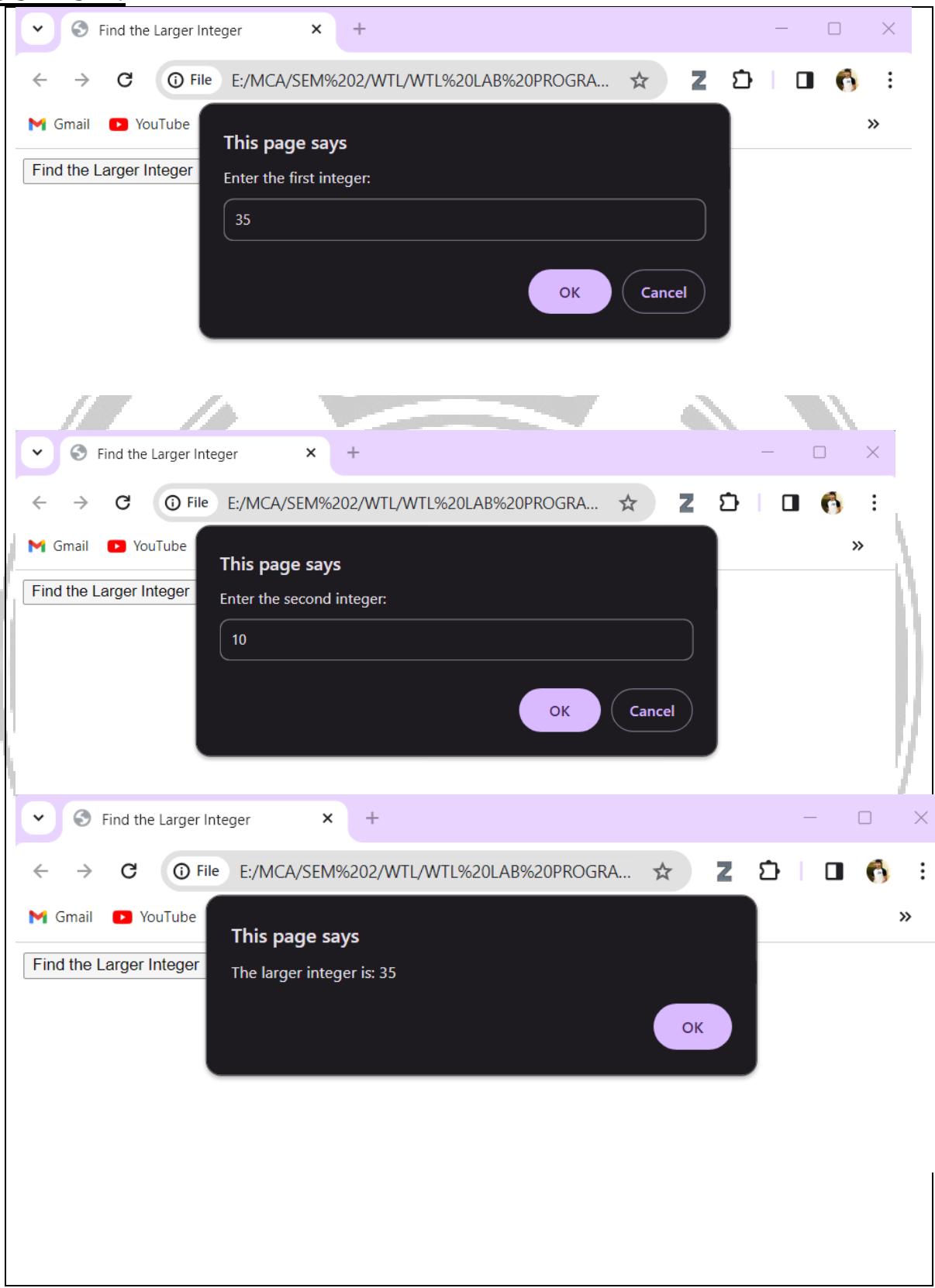
Write a program to implement pop-up boxes available in JavaScript.

CODE :

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Find the Larger Integer</title>
    <script>
        function findLarger() {
            // Get the values entered by the user
            var num1 = parseInt(prompt("Enter the first integer:"));
            var num2 = parseInt(prompt("Enter the second integer:"));

            // Check if both inputs are valid integers
            if (!isNaN(num1) && !isNaN(num2)) {
                // Compare the two integers and display the larger one
                if (num1 > num2) {
                    alert("The larger integer is: " + num1);
                } else if (num2 > num1) {
                    alert("The larger integer is: " + num2);
                } else {
                    alert("Both integers are equal.");
                }
            } else {
                alert("Please enter valid integers.");
            }
        }
    </script>
</head>
<body>
    <button onclick="findLarger()">Find the Larger Integer</button>
</body>
</html>
```

OUTPUT :





Write a program to implement Email Validation in JavaScript.

CODE :

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Email Validation</title>
    <script>
        function validateEmail() {
            // Get the email input value
            var email = document.getElementById("email").value;

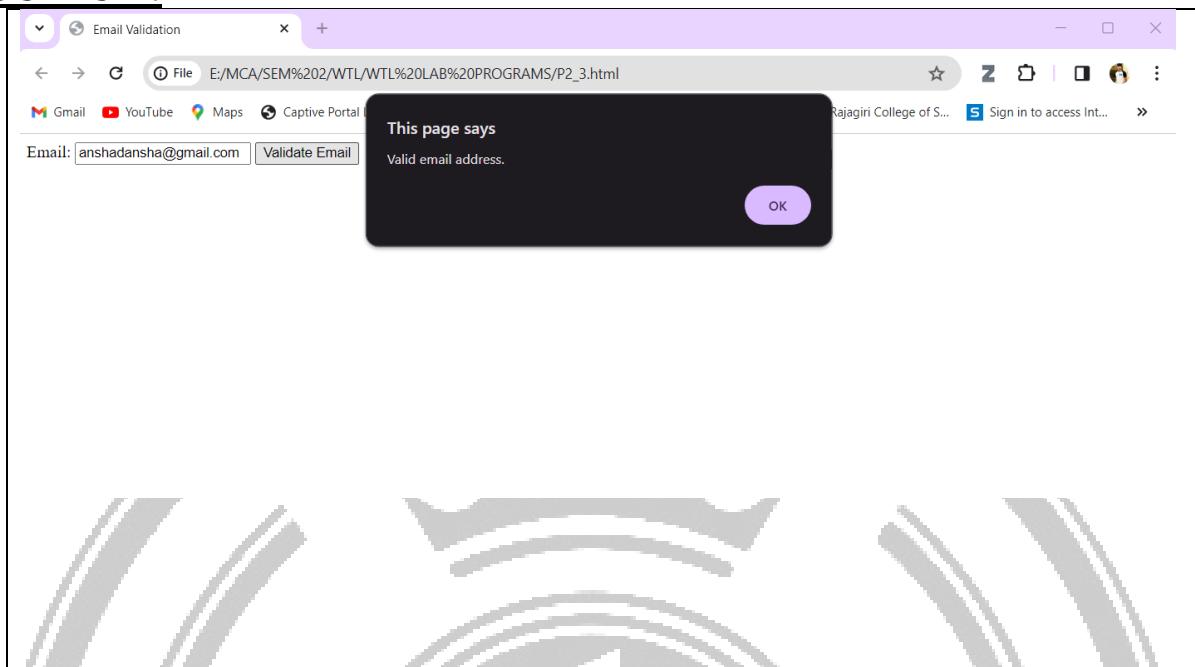
            // Regular expression for email validation
            var regex = /^[^@\s]+@[^\s@]+\.[^\s@]+$/;

            // Test if the email matches the regular expression
            if (regex.test(email)) {
                alert("Valid email address.");
            } else {
                alert("Invalid email address. Please enter a valid email.");
            }
        }
    </script>
</head>
<body>

    <label for="email">Email:</label>
    <input type="text" id="email" name="email">
    <button onclick="validateEmail()">Validate Email</button>

</body>
</html>
```

OUTPUT :



Write a JavaScript program to sort the items of an array.

CODE :

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Sort Array</title>
</head>
<body>

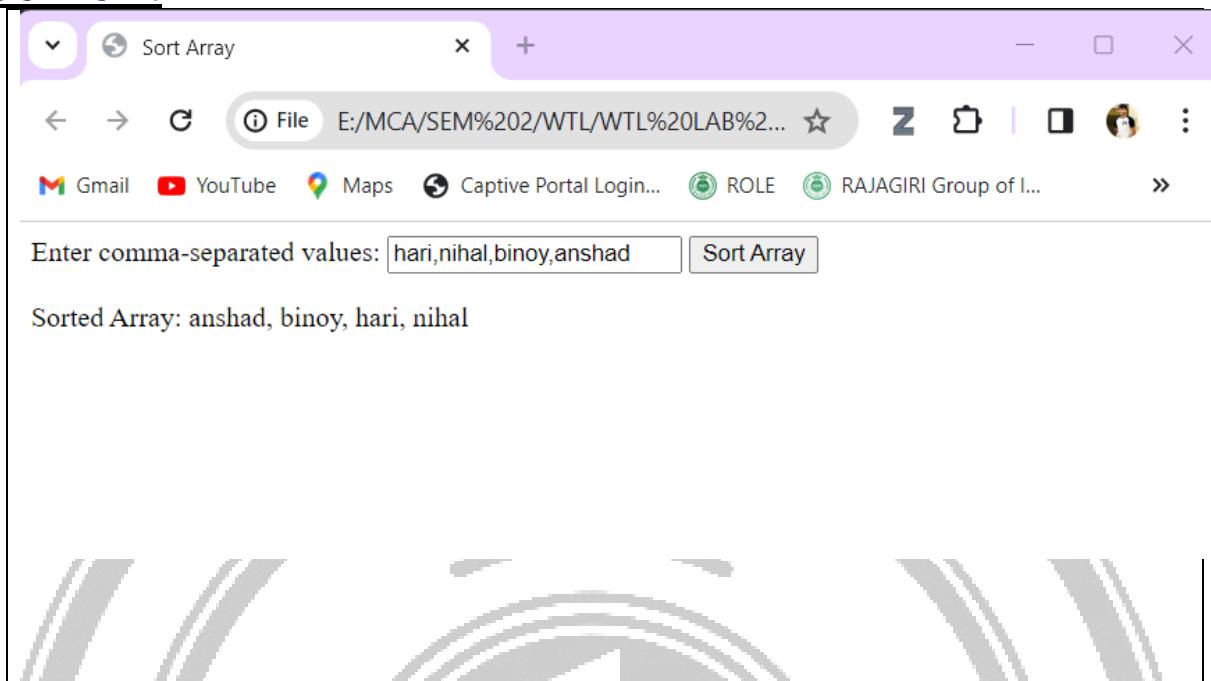
    <label for="inputValues">Enter comma-separated values:</label>
    <input type="text" id="inputValues">
    <button onclick="sortArray()">Sort Array</button>

    <p id="arrayValues"></p>

<script>
    function sortArray() {
        // Get the input value from the textbox
        var input = document.getElementById("inputValues").value;
        // Split the input string into an array
        var items = input.split(",");
        // Sort the array
        items.sort();
        // Display the sorted array
        document.getElementById("arrayValues").innerHTML = "Sorted Array:
" + items.join(", ");
    }
</script>

</body>
</html>
```

OUTPUT :



Sort Array

E:/MCA/SEM%202/WTL/WTL%20LAB%2...

Gmail YouTube Maps Captive Portal Login... ROLE RAJAGIRI Group of I...

Enter comma-separated values: hari,nihal,binoy,anshad Sort Array

Sorted Array: anshad, binoy, hari, nihal



Demonstrate Function call in JavaScript.**CODE :**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Function Call Demo</title>
</head>
<body>

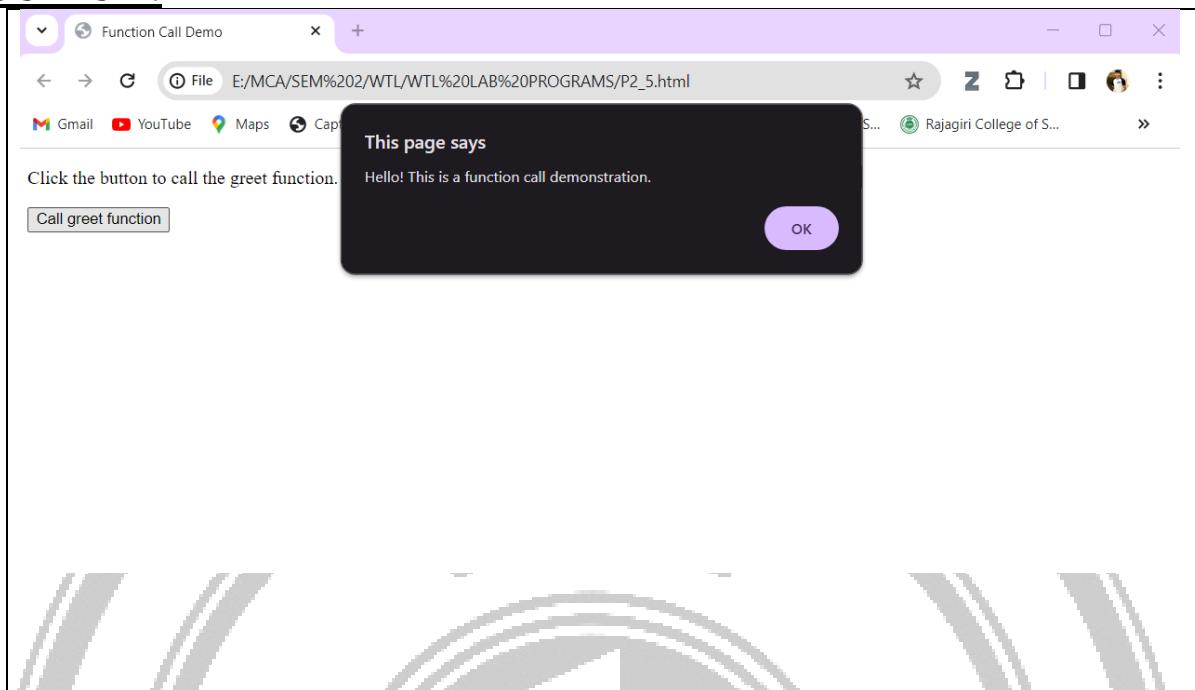
    <p>Click the button to call the greet function.</p>

    <button onclick="greet()">Call greet function</button>

    <script>
        // Define a function called greet
        function greet() {
            alert("Hello! This is a function call demonstration.");
        }
    </script>

</body>
</html>
```

OUTPUT :



Write a JavaScript program to test the first character of a string is uppercase or not.

CODE :

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Test Uppercase</title>
</head>
<body>
    <label for="inputString">Enter a string:</label>
    <input type="text" id="inputString">
    <button onclick="testUppercase()">Test Uppercase</button>
    <p id="result"></p>
    <script>
        function testUppercase() {
            // Get the input value from the text box
            var inputString = document.getElementById("inputString").value;

            // Check if the first character is uppercase
            if (inputString.charAt(0) === inputString.charAt(0).toUpperCase()) {
                document.getElementById("result").textContent = "The first character is uppercase.";
            } else {
                document.getElementById("result").textContent = "The first character is not uppercase.";
            }
        }
    </script>
</body>
</html>
```

OUTPUT :

The first character is uppercase.



Write a JavaScript Program to Convert Decimal to Binary.**CODE :**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Decimal to Binary Converter</title>
</head>
<body>
    <label for="decimalInput">Enter a decimal number:</label>
    <input type="number" id="decimalInput">
    <button onclick="convertToBinary()">Convert to Binary</button>
    <p id="binaryOutput"></p>
    <script>
        function convertToBinary() { // Get the decimal input value
            var decimalNumber =
                parseInt(document.getElementById("decimalInput").value);
            // Check if the input is a valid number
            if (!isNaN(decimalNumber)) {
                // Convert decimal to binary
                var binaryNumber = decimalNumber.toString(2);

                // Display the binary output
                document.getElementById("binaryOutput").textContent =
                    "Binary representation: " + binaryNumber;
            } else {
                // Display error message for invalid input
                document.getElementById("binaryOutput").textContent =
                    "Invalid input.  
Please enter a valid decimal number.";
            }
        }
    </script>
</body>
</html>
```

OUTPUT :

Decimal to Binary Converter

E:/MCA/SEM%202/WTL/WTL%20LAB%20P...

Gmail YouTube Maps Captive Portal Login... ROLE RAJAGIRI Group of I...

Enter a decimal number: Convert to Binary

Binary representation: 11001

LEARN • SERVE • EXCEL
RAJAGIRI

Program 2.8

Date: 2/02/2024

Create a webpage that displays the current date and time using Javascript.

CODE :

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Current Date and Time</title>
</head>
<body>

    <p id="dateTime"></p>

    <script>
        function updateDateTime() {
            // Get the current date and time
            var currentDate = new Date();

            // Format the date and time as a string
            var dateTimeString = currentDate.toLocaleString();

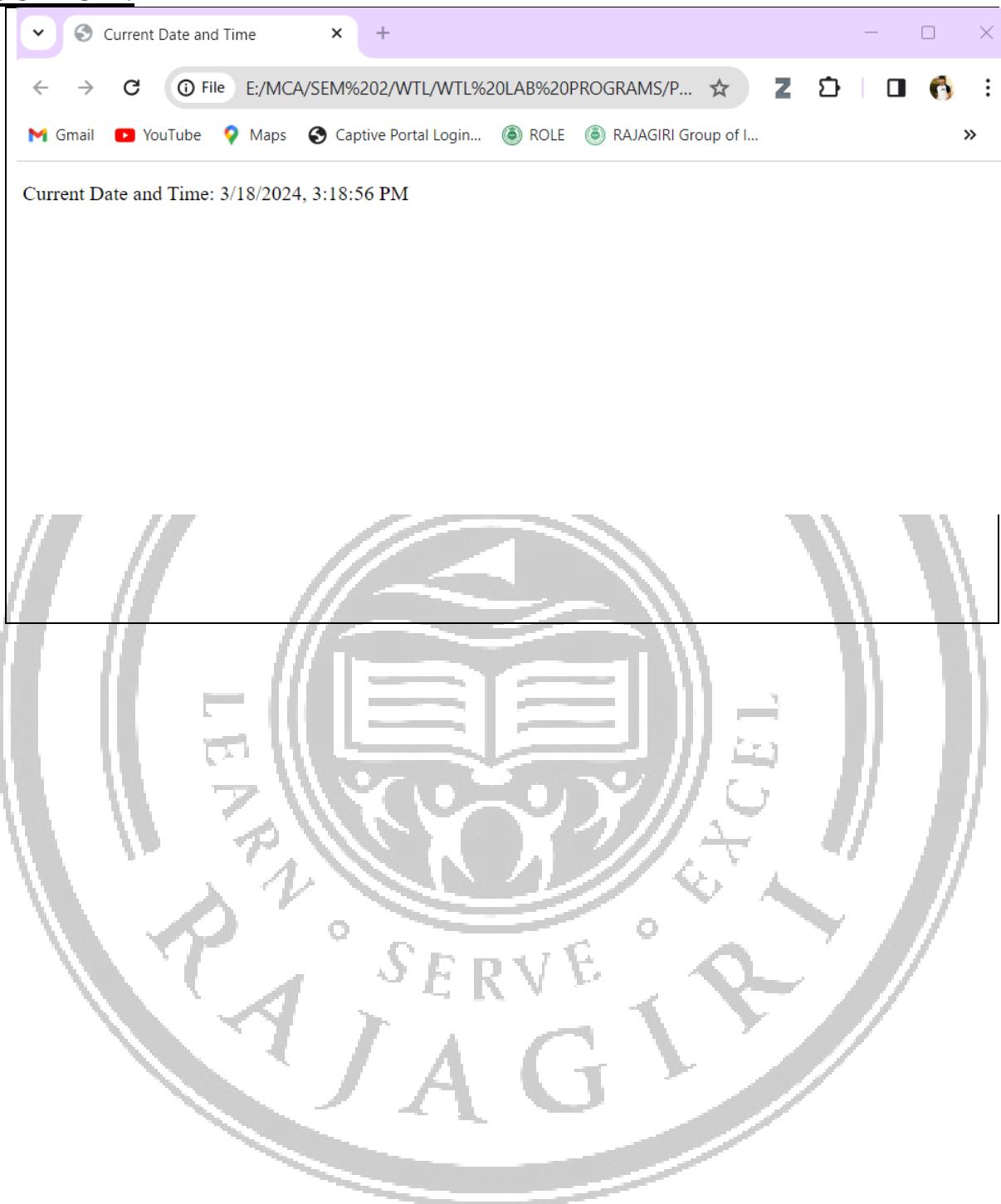
            // Update the content of the HTML element
            document.getElementById("dateTime").textContent = "Current Date and
Time: " + dateTimeString;
        }

        // Call the function initially to display the current date and time
        updateDateTime();

        // Update the date and time every second
        setInterval(updateDateTime, 1000);
    </script>

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

OUTPUT :



Implement a JavaScript program to change the background color randomly.

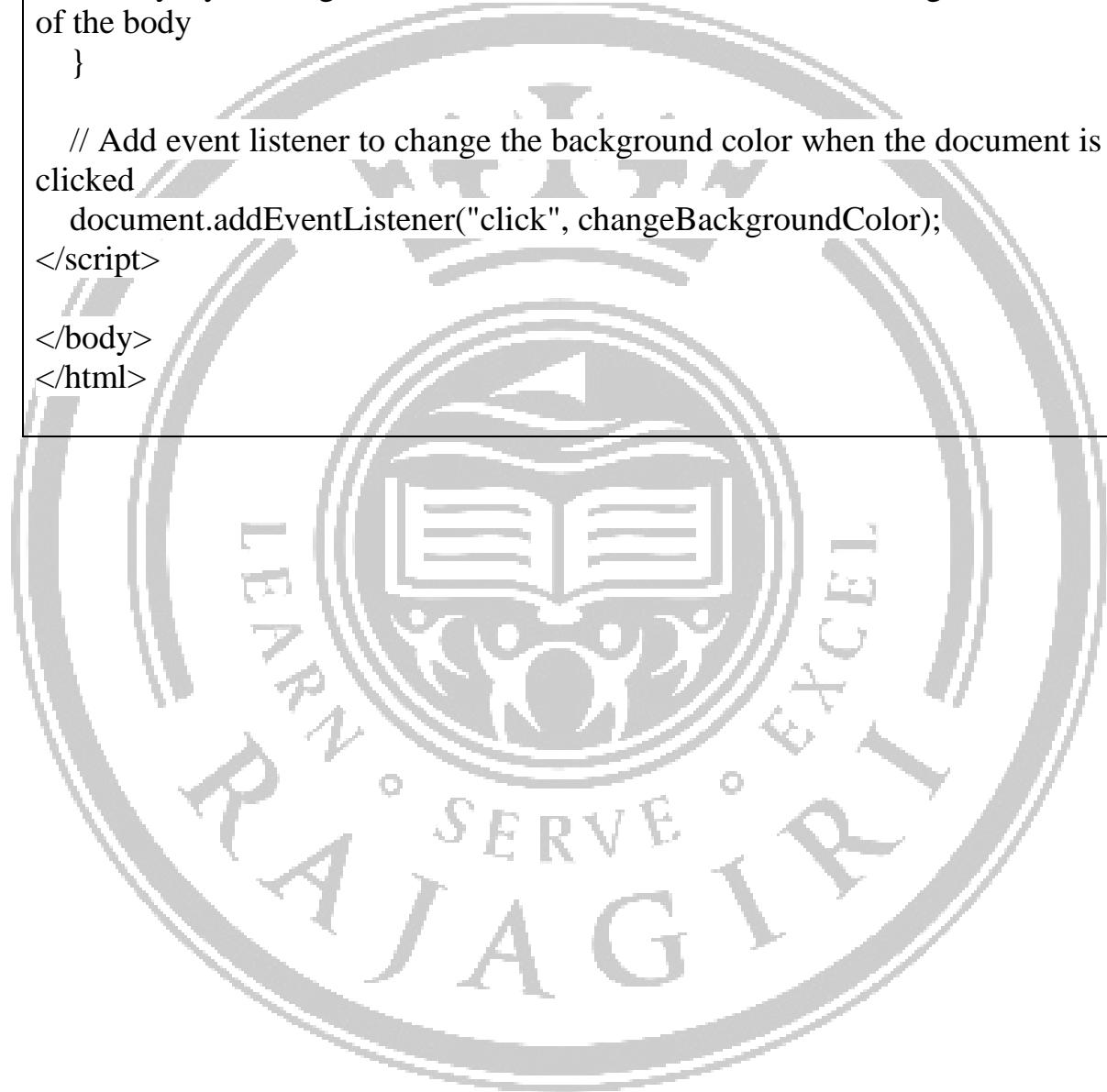
CODE :

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Random Background Color</title>
<style>
body {
    margin: 0;
    padding: 0;
    height: 100vh;
    display: flex;
    justify-content: center;
    align-items: center;
    font-size: 24px;
    color: white;
}
</style>
</head>
<body id="body">
<p>Click anywhere to change the background color randomly.</p>
<script>
// Function to generate a random RGB color
function getRandomColor() {
    var r = Math.floor(Math.random() * 256); // Random red value between
0 and 255
    var g = Math.floor(Math.random() * 256); // Random green value
between 0 and 255
    var b = Math.floor(Math.random() * 256); // Random blue value between
0 and 255
}
```

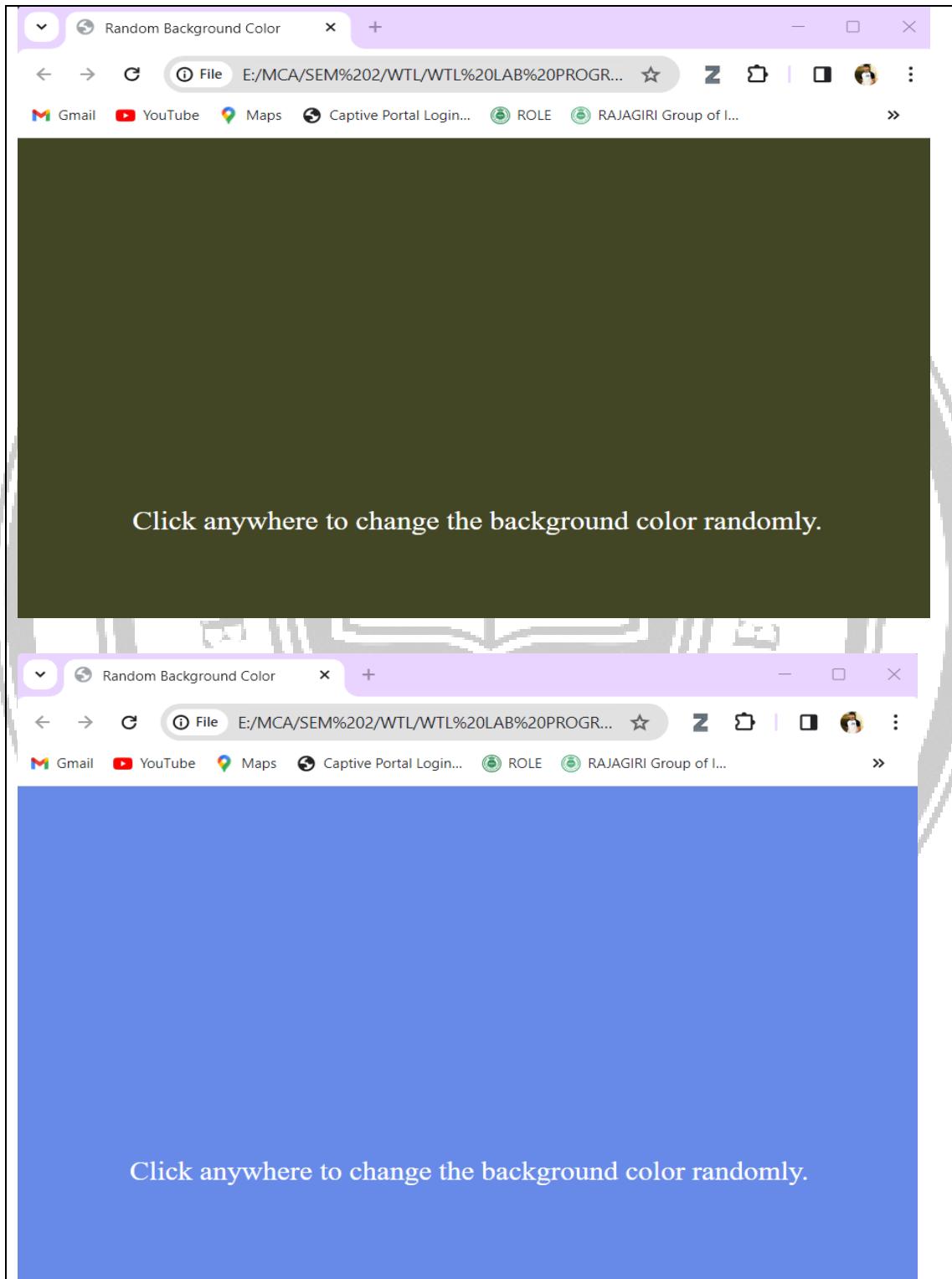
```
        return `rgb(${r}, ${g}, ${b})`; // Return the RGB color string
    }

// Function to change the background color randomly
function changeBackgroundColor() {
    var body = document.getElementById("body"); // Get the body element
    var randomColor = getRandomColor(); // Get a random color
    body.style.backgroundColor = randomColor; // Set the background color
of the body
}

// Add event listener to change the background color when the document is
clicked
document.addEventListener("click", changeBackgroundColor);
</script>
</body>
</html>
```



OUTPUT :



Program 2.10

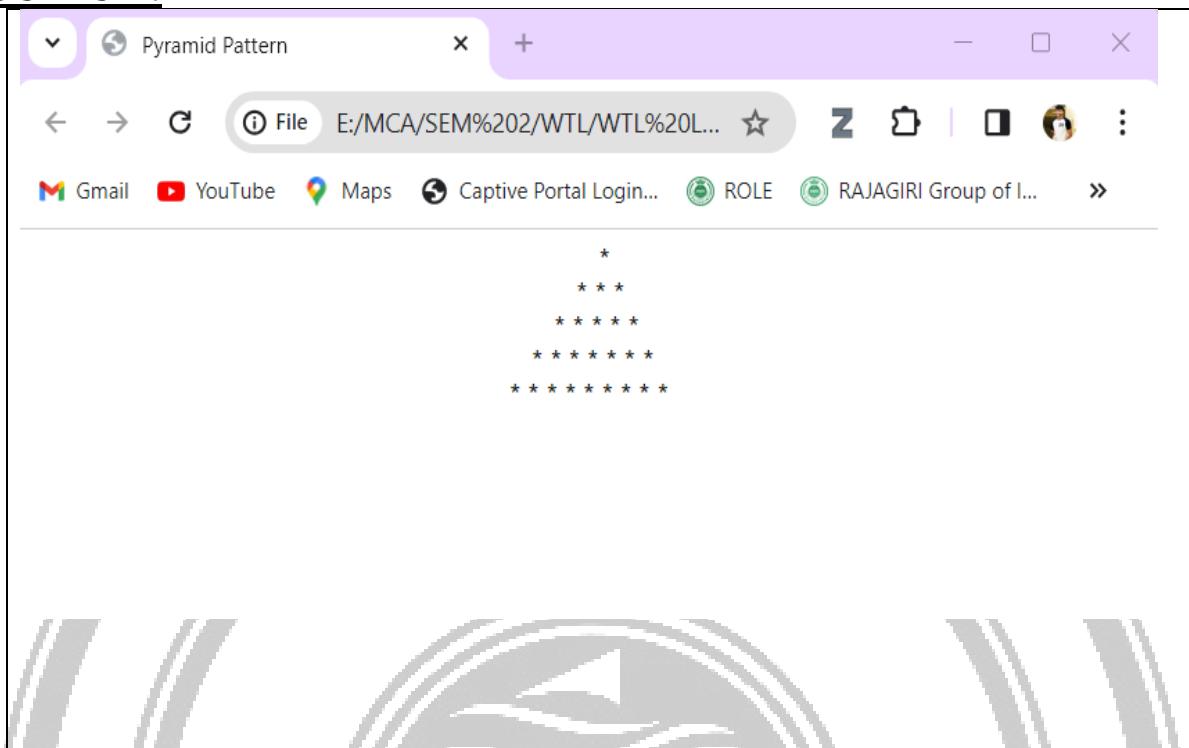
Date: 4/02/2024

Write a JavaScript program to construct the pyramid pattern, using a nested for loop.

CODE :

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Pyramid Pattern</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            text-align: center;
        }
    </style>
</head>
<body>
<script>
    // Define the height of the pyramid
    var height = 5;
    // Outer loop for rows
    for (var i = 1; i <= height; i++) {
        // Inner loop for spaces
        for (var j = height - i; j > 0; j--) {
            document.write(" "); // Output space
        }
        // Inner loop for stars
        for (var k = 1; k <= 2 * i - 1; k++) {
            document.write("* "); // Output star
        }
        document.write("<br>"); // Output newline
    }
</script>
</body>
</html>
```

OUTPUT :



Write a JavaScript Program to Solve Quadratic Equations.**CODE :**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Quadratic Equation Solver</title>
</head>
<body>

<label for="inputA">Enter the value of 'a':</label>
<input type="number" id="inputA"><br>

<label for="inputB">Enter the value of 'b':</label>
<input type="number" id="inputB"><br>

<label for="inputC">Enter the value of 'c':</label>
<input type="number" id="inputC"><br>

<button onclick="solveQuadratic()">Solve Quadratic Equation</button>

<p id="solution"></p>

<script>
function solveQuadratic() {
  // Get the values of coefficients 'a', 'b', and 'c'
  var a = parseFloat(document.getElementById("inputA").value);
  var b = parseFloat(document.getElementById("inputB").value);
  var c = parseFloat(document.getElementById("inputC").value);

  // Check if 'a' is zero
  if (a === 0) {
    document.getElementById("solution").textContent = "This is not a quadratic
equation. 'a' cannot be zero.";
    return;
  }
}
```

```

// Calculate the discriminant
var discriminant = b * b - 4 * a * c;

// Initialize variables to store solutions
var x1, x2;

// Check if the discriminant is positive, zero, or negative
if (discriminant > 0) {
    // Two real and distinct roots
    x1 = (-b + Math.sqrt(discriminant)) / (2 * a);
    x2 = (-b - Math.sqrt(discriminant)) / (2 * a);
    document.getElementById("solution").textContent = "The roots are: " + x1 +
        " and " + x2;
} else if (discriminant === 0) {
    // One real root (double root)
    x1 = -b / (2 * a);
    document.getElementById("solution").textContent = "The root is: " + x1;
} else {
    // Complex roots
    var realPart = -b / (2 * a);
    var imaginaryPart = Math.sqrt(-discriminant) / (2 * a);
    document.getElementById("solution").textContent = "The roots are complex:
        " + realPart + " + " + imaginaryPart + "i and " + realPart + " - " +
        imaginaryPart + "i";
}
}

</script>

</body>
</html>

```

OUTPUT :

Quadratic Equation Solver

File E:/MCA/SEM%202/WTL/WTL%20LAB%20PROGRAMS/P2_1...

Gmail YouTube Maps Captive Portal Login... ROLE RAJAGIRI Group of I... Rajagiri College of S...

Enter the value of 'a':
Enter the value of 'b':
Enter the value of 'c':

The roots are: -0.0752833793603962 and -19.924716620639604



Write a JavaScript function that accepts a string as a parameter and counts the number of vowels within the string.

CODE :

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Count Vowels</title>
</head>
<body>

    <label for="inputString">Enter a string:</label>
    <input type="text" id="inputString" value="Hello World!">
    <button onclick="countVowels()">Count Vowels</button>

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

<script>
    function countVowels() {
        // Get the input string
        var inputString = document.getElementById("inputString").value.toLowerCase(); // Convert to lowercase for case-insensitive counting

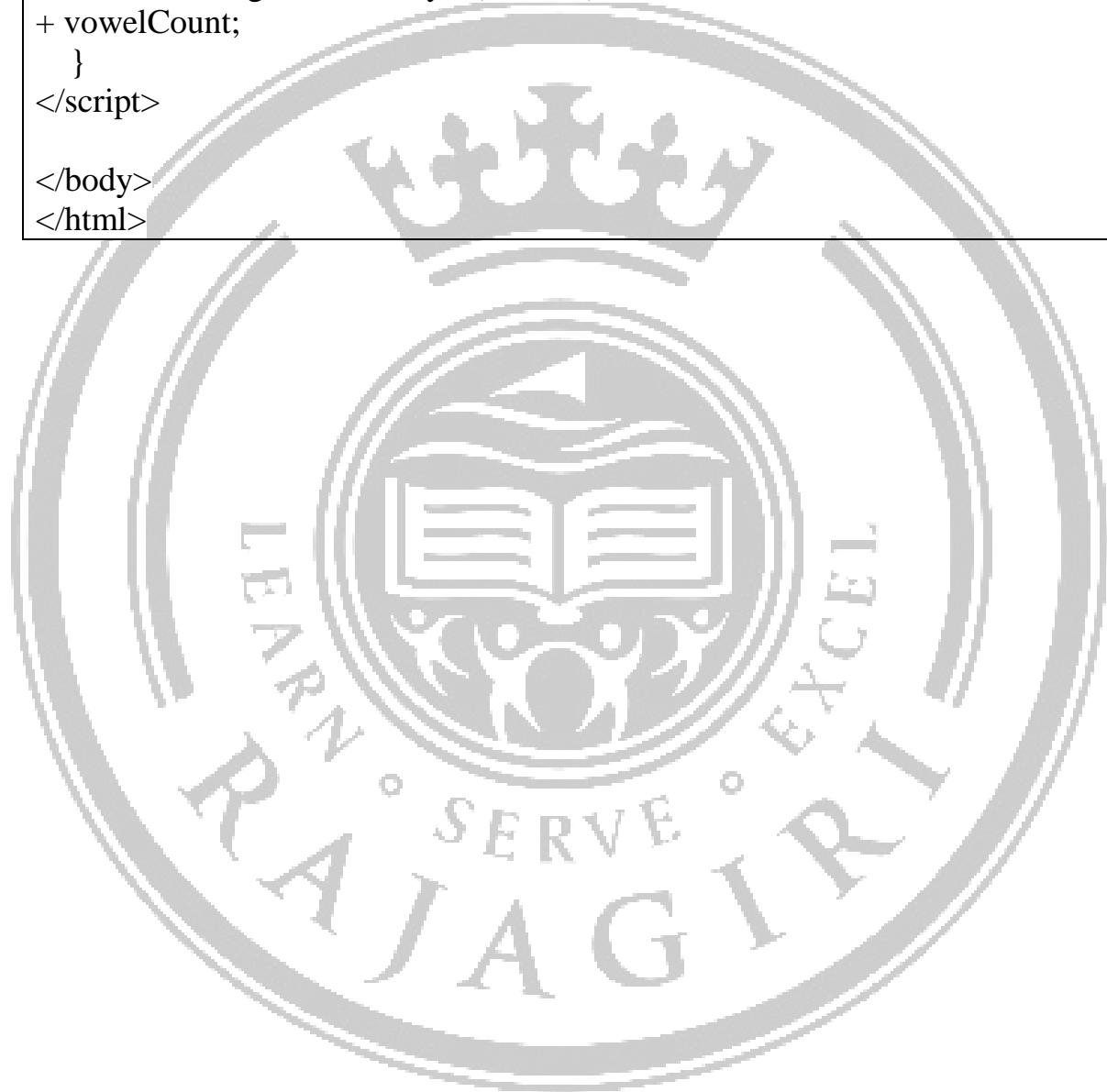
        // Define an array containing the vowels
        var vowels = ['a', 'e', 'i', 'o', 'u'];

        // Initialize a variable to count the number of vowels
        var vowelCount = 0;

        // Loop through each character in the input string
        for (var i = 0; i < inputString.length; i++) {
            // Check if the current character is a vowel
            if (vowels.includes(inputString[i])) {
                vowelCount++;
            }
        }

        // Display the result
        document.getElementById("result").innerHTML = "Number of vowels: " + vowelCount;
    }
</script>
```

```
if (vowels.includes(inputString[i])) {  
    // If it's a vowel, increment the vowel count  
    vowelCount++;  
}  
}  
  
// Display the result  
document.getElementById("result").textContent = "Number of vowels: "  
+ vowelCount;  
}  
</script>  
  
</body>  
</html>
```



OUTPUT :

Count Vowels

File E:/MCA/SEM%202/WTL/WTL%20LAB%20PRO... ★

Gmail YouTube Maps Captive Portal Login... ROLE RAJAGIRI Group of I...

Enter a string: Count Vowels

Number of vowels: 2



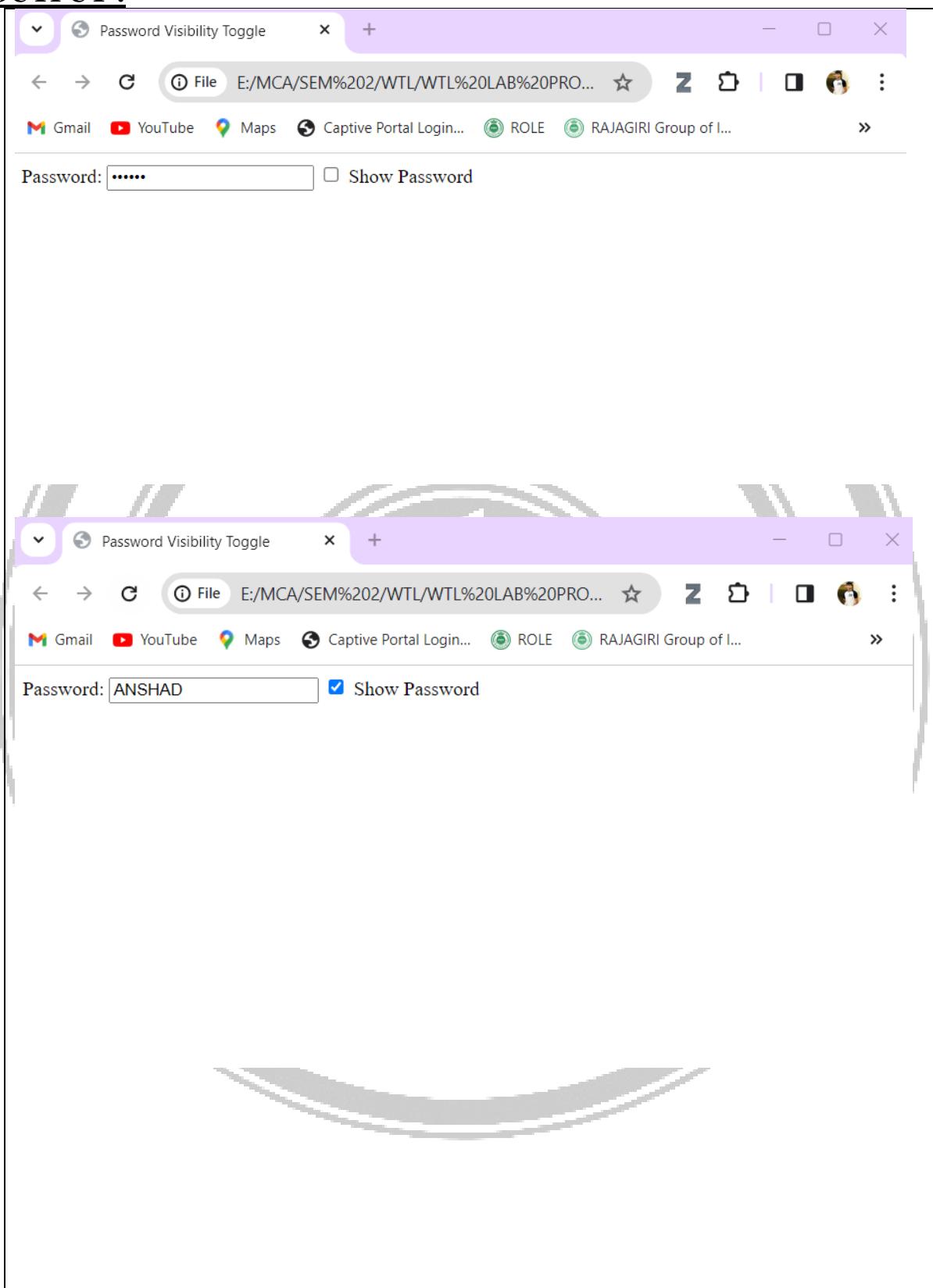
Write a program to Toggle Password Visibility in JavaScript.

CODE :

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Password Visibility Toggle</title>
</head>
<body>
<label for="password">Password:</label>
<input type="password" id="password">
<input type="checkbox" id="toggleCheckbox"
  onclick="togglePasswordVisibility()">
<label for="toggleCheckbox">Show Password</label>
<script>
function togglePasswordVisibility() { // Get the password input element and
  // the toggle checkbox
  var passwordInput = document.getElementById("password");
  var toggleCheckbox = document.getElementById("toggleCheckbox");

  // Check if the checkbox is checked
  if (toggleCheckbox.checked) {
    // If checked, set the type of the password input to "text" to show the
    // password
    passwordInput.type = "text";
  } else {
    // If not checked, set the type of the password input back to "password" to
    // hide the password
    passwordInput.type = "password";
  }
}
</script>
</body>
</html>
```

OUTPUT :





Write a program to demonstrate the Fibonacci sequence in JSON.

CODE :

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Fibonacci Sequence in JSON</title>
</head>
<body>

<label for="fibonacciCount">Enter the number of Fibonacci numbers:</label>
<input type="number" id="fibonacciCount"><br>
<button onclick="generateFibonacciJSON()">Generate Fibonacci Sequence</button>

<p id="fibonacciOutput"></p>

<script>
// Function to generate the Fibonacci sequence and return it as a JSON string
function generateFibonacciJSON() {
  var fibonacciCount = parseInt(document.getElementById("fibonacciCount").value);

  if (isNaN(fibonacciCount) || fibonacciCount <= 0) {
    document.getElementById("fibonacciOutput").textContent = "Please enter a valid positive integer.";
    return;
  }

  var fibonacciSequence = [0, 1];

  for (var i = 2; i < fibonacciCount; i++) {
    fibonacciSequence.push(fibonacciSequence[i - 1] + fibonacciSequence[i - 2]);
  }

  document.getElementById("fibonacciOutput").textContent = JSON.stringify(fibonacciSequence);
}

</script>
```

```
}
```

```
var fibonacciJSON = JSON.stringify(fibonacciSequence);
```

```
document.getElementById("fibonacciOutput").textContent = "Fibonacci
```

```
Sequence (JSON): " + fibonacciJSON;
```

```
}
```

```
</script>
```

```
</body>
```

```
</html>
```



OUTPUT :

Fibonacci Sequence in JSON

File E:/MCA/SEM%202/WTL/WTL%20LA...

Gmail YouTube Maps Captive Portal Login... ROLE RAJAGIRI Group of I...

Enter the number of Fibonacci numbers:

Generate Fibonacci Sequence

Fibonacci Sequence (JSON): [0,1,1,2,3,5,8,13,21,34,55,89,144,233,377]



Write a program to check leap years.

CODE :

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Leap Year Checker</title>
</head>
<body>
    <label for="yearInput">Enter a year:</label>
    <input type="number" id="yearInput">
    <button onclick="checkLeapYear()">Check Leap Year</button>
    <p id="result"></p>
    <script>
        function checkLeapYear() {
            // Get the input year from the user
            var year = parseInt(document.getElementById("yearInput").value);
            // Check if the year is a leap year
            var isLeapYear = ((year % 4 === 0 && year % 100 !== 0) || (year % 400
            === 0));
            // Prepare the result object in JSON format
            var result = {
                year: year,
                isLeapYear: isLeapYear
            };
            // Display the result in JSON format
            document.getElementById("result").textContent = JSON.stringify(result,
            null, 2);
        }
    </script>
</body>
</html>
```

OUTPUT :

Leap Year Checker

File E:/MCA/SEM%202/WTL/WT...

Gmail YouTube Maps Captive Portal Login... ROLE

Enter a year: 2004 Check Leap Year

{ "year": 2004, "isLeapYear": true }



Program 4.1

Date: 22/02/2024

Write a PHP program to check whether a given number is Armstrong or not.

CODE :

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Armstrong Number Checker</title>
</head>
<body>

<form method="post">
    <label for="number">Enter a number:</label>
    <input type="number" id="number" name="number">
    <button type="submit">Check</button>
</form>

<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Get the input number from the form
    $number = $_POST["number"];

    // Function to check if a number is Armstrong
    function isArmstrong($num) {
        $sum = 0;
        $temp = $num;
        $digits = strlen($num);

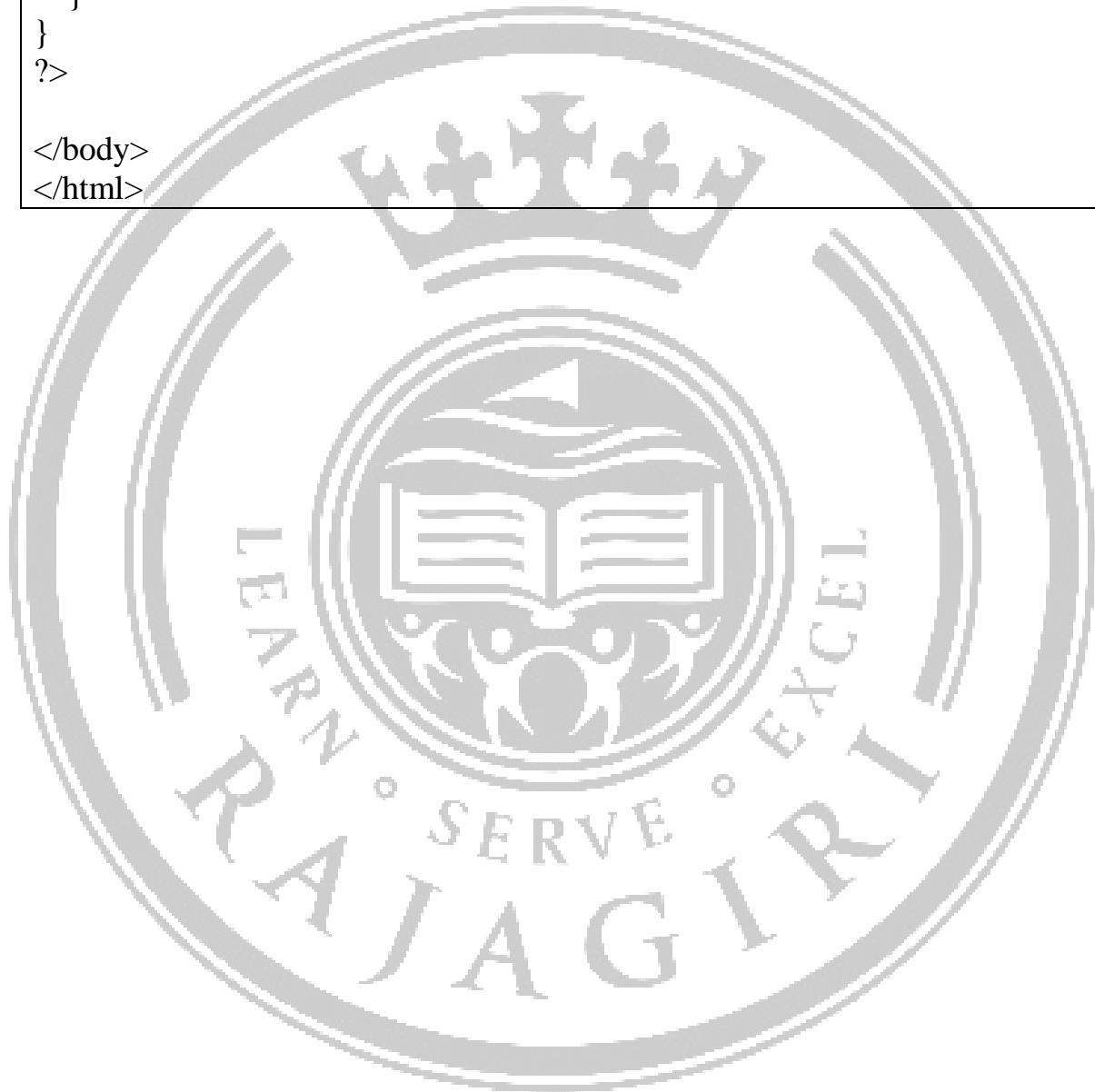
        while ($temp != 0) {
            $remainder = $temp % 10;
            $sum += pow($remainder, $digits);
            $temp = intval($temp / 10);
        }

        return $num == $sum;
    }
}
```

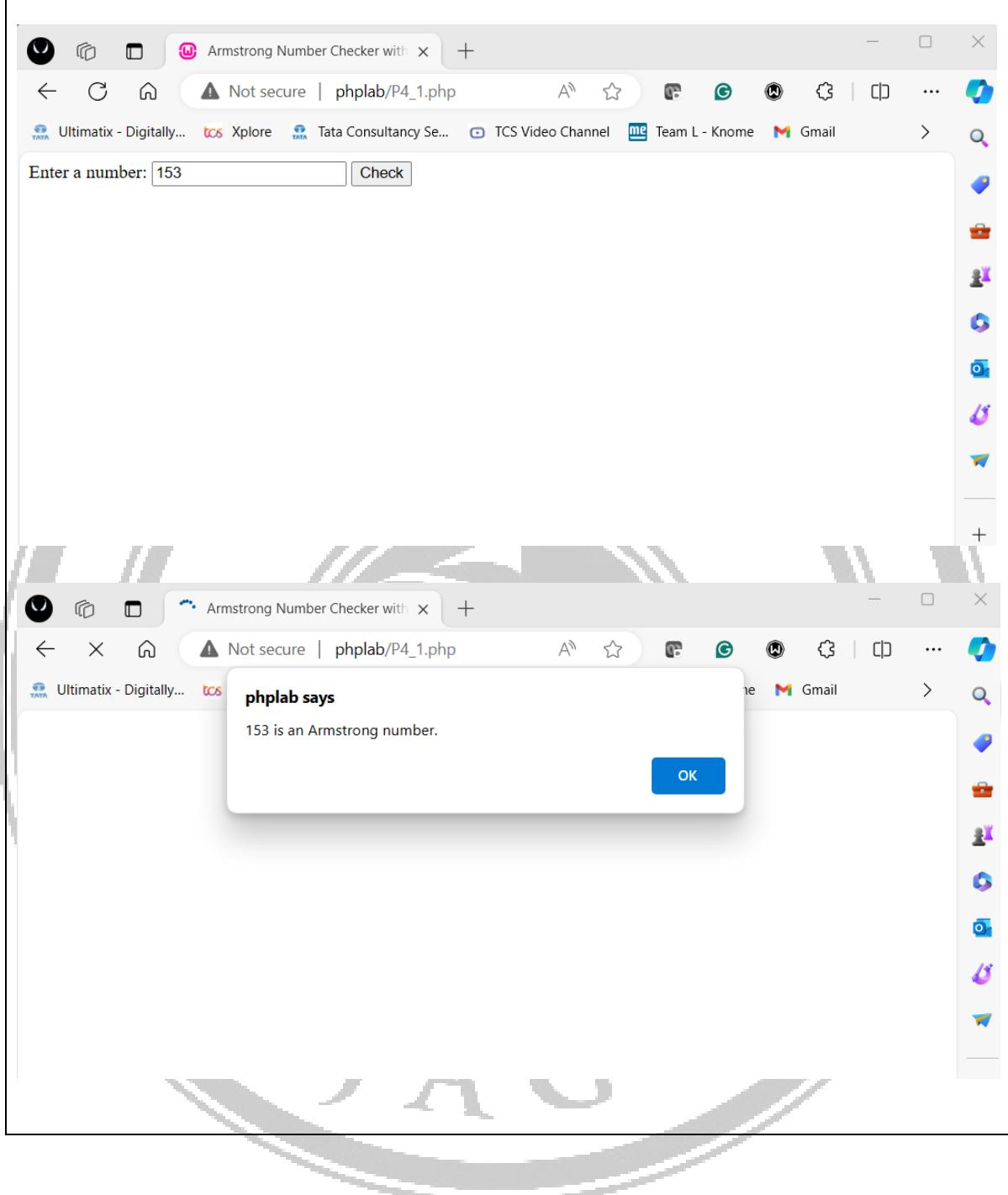
```
}

// Check if the input number is Armstrong
if (isArmstrong($number)) {
    echo "<p>{$number} is an Armstrong number.</p>";
} else {
    echo "<p>{$number} is not an Armstrong number.</p>";
}
?>

</body>
</html>
```



OUTPUT :



Write a PHP program to illustrate Cookies.

CODE :

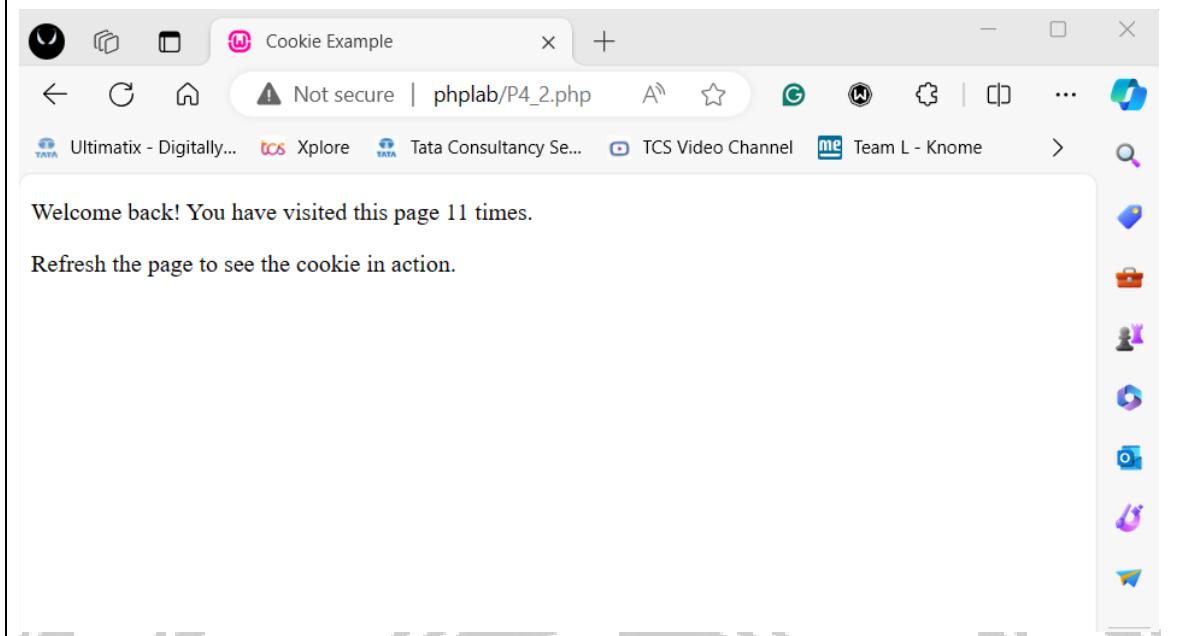
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Cookie Example</title>
</head>
<body>

<?php
// Check if the cookie is set
if (!isset($_COOKIE['visited'])) {
    // If not set, set a cookie to track the number of visits
    $count = 1;
    setcookie('visited', $count, time() + (86400 * 30), "/");
    // Cookie expires in 30 days
    echo "<p>Welcome! This is your first visit.</p>";
} else {
    // If set, retrieve the cookie value and increment the count
    $count = $_COOKIE['visited'] + 1;
    setcookie('visited', $count, time() + (86400 * 30), "/");
    // Update cookie with incremented count
    echo "<p>Welcome back! You have visited this page $count times.</p>";
}
?>

<p>Refresh the page to see the cookie in action.</p>

</body>
</html>
```

OUTPUT :



Check whether a year is a leap year or not by inserting a year in PHP.**CODE :**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Leap Year Checker</title>
</head>
<body>

<form method="post">
    <label for="year">Enter a year:</label>
    <input type="number" id="year" name="year" min="1" required>
    <button type="submit">Check</button>
</form>
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Get the input year from the form
    $year = $_POST["year"];
    // Function to check if a year is a leap year
    function isLeapYear($year) {
        return (($year % 4 == 0) && ($year % 100 != 0)) || ($year % 400 == 0);
    }
    // Check if the input year is a leap year
    if (isLeapYear($year)) {
        echo "<p>{$year} is a leap year.</p>";
    } else {
        echo "<p>{$year} is not a leap year.</p>";
    }
}
?>
</body>
</html>
```

OUTPUT :

Leap Year Checker

Not secure | phplab/P4_3.php

Enter a year: Check

2004 is a leap year.



Program 4.4

Date: 25/02/2024

Write a program to Implement static properties in PHP.

CODE :

```
<?php

class MyClass {
    public static $staticProperty = "Hello, World!";

    public function __construct() {
        // Constructor
    }

    public static function getStaticProperty() {
        return self::$staticProperty;
    }

    public static function setStaticProperty($value) {
        self::$staticProperty = $value;
    }
}

// Accessing static property without creating an instance of the class
echo MyClass::$staticProperty; // Output: Hello, World!
echo "<br>";

// Changing the value of the static property
MyClass::$staticProperty = "Greetings!";
echo MyClass::getStaticProperty(); // Output: Greetings!
echo "<br>";

// Creating an instance of the class
$obj1 = new MyClass();
echo $obj1::$staticProperty; // Output: Greetings!
echo "<br>";

// Creating another instance of the class
$obj2 = new MyClass();
```

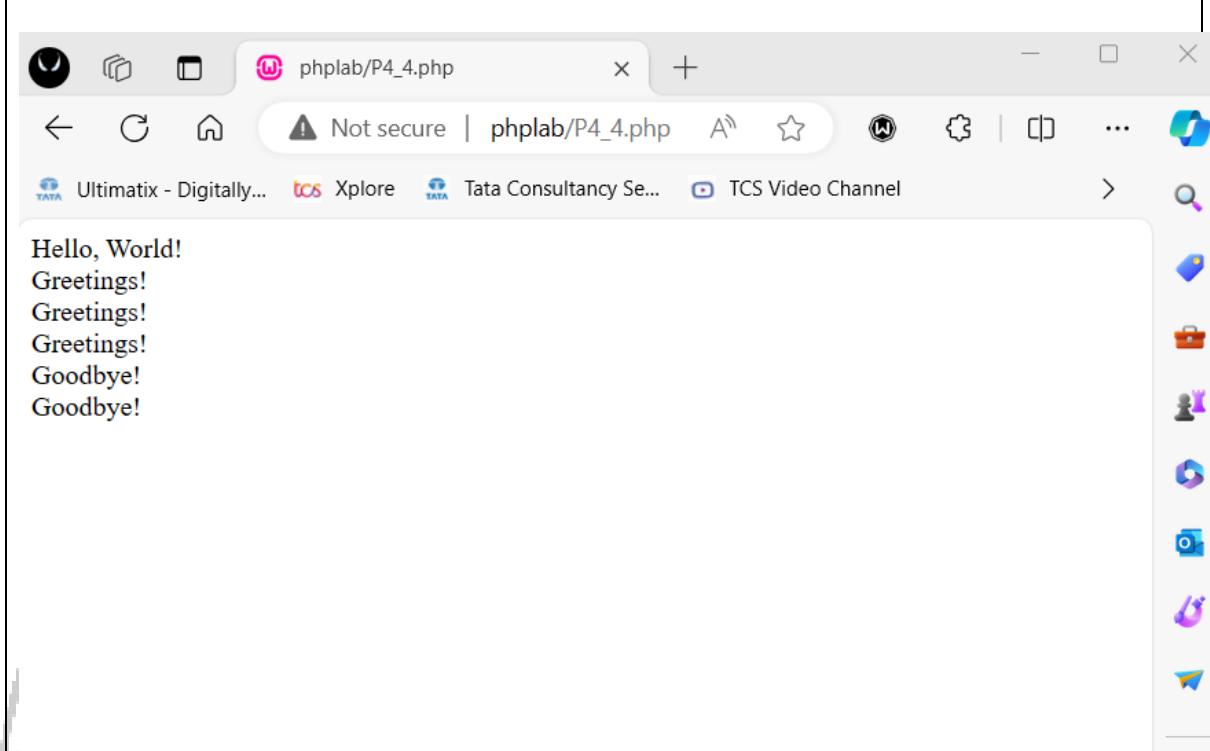
```
echo $obj2::$staticProperty; // Output: Greetings!
echo "<br>";

// Changing the value of the static property using a method
$obj1::setStaticProperty("Goodbye!");
echo $obj1::$staticProperty; // Output: Goodbye!
echo "<br>";
echo $obj2::$staticProperty; // Output: Goodbye!
echo "<br>";

?>
```



OUTPUT :



A screenshot of a web browser window titled "phplab/P4_4.php". The address bar shows a warning icon indicating "Not secure" and the URL "phplab/P4_4.php". Below the address bar, there are several tabs: "Ultimatix - Digitally...", "tcs Xplore", "Tata Consultancy Se...", and "TCS Video Channel". The main content area of the browser displays the following text:
Hello, World!
Greetings!
Greetings!
Greetings!
Goodbye!
Goodbye!



Program 4.5

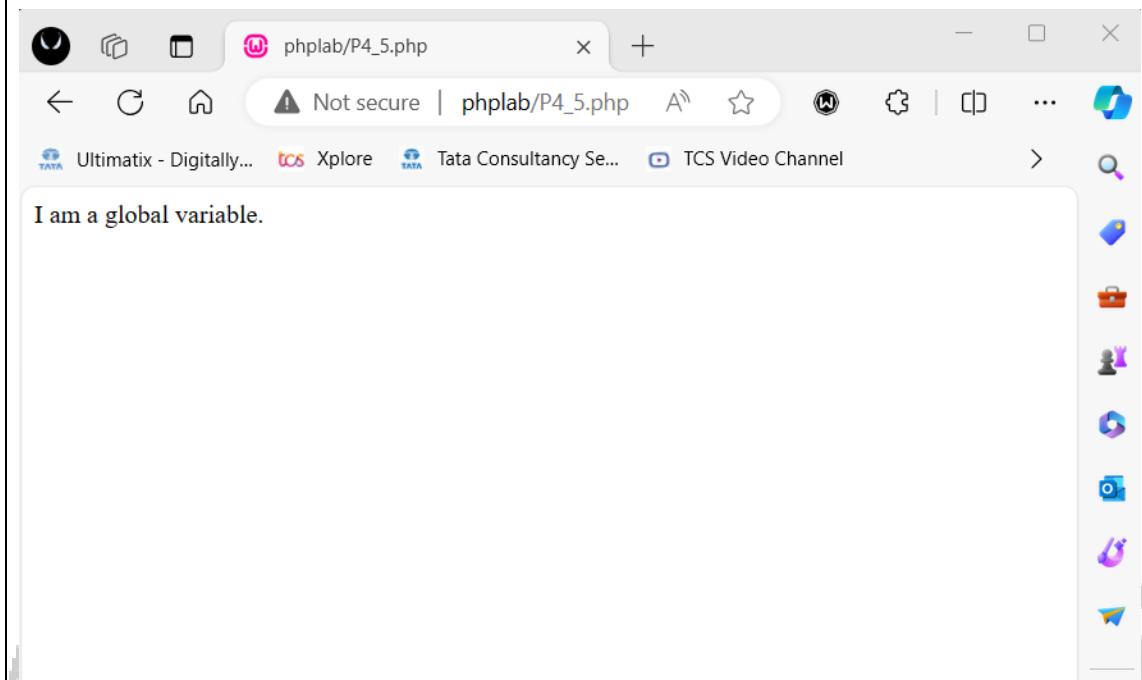
Date: 27/02/2024

Write a PHP program to use the global keyword to access a global variable from within a function.

CODE :

```
<?php  
  
// Define a global variable  
$globalVariable = "I am a global variable."  
  
// Function to access the global variable using the global keyword  
function accessGlobalVariable() {  
    // Access the global variable using the global keyword  
    global $globalVariable;  
    echo $globalVariable;  
}  
  
// Call the function to access the global variable  
accessGlobalVariable();  
  
?>
```

OUTPUT :



Write a PHP program to convert a decimal number into a binary number.

CODE :

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Decimal to Binary Converter</title>
</head>
<body>

<form method="post">
    <label for="decimal">Enter a decimal number:</label>
    <input type="number" id="decimal" name="decimal" min="0" required>
    <button type="submit">Convert</button>
</form>

<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Get the decimal number from the form
    $decimal = $_POST["decimal"];

    // Function to convert decimal to binary
    function decimalToBinary($decimal) {
        $binary = "";
        while ($decimal > 0) {
            // Append the remainder of division by 2 to the binary string
            $binary = ($decimal % 2) . $binary;
            // Divide the decimal number by 2
            $decimal = intval($decimal / 2);
        }
        return $binary;
    }

    // Convert the decimal number to binary
}
```

```
$binary = decimalToBinary($decimal);
echo "<p>Binary representation of {$decimal} is {$binary}.</p>";
}
?>

</body>
</html>
```



OUTPUT :

Decimal to Binary Converter

Not secure | phplab/P4_6.php

Enter a decimal number: Convert

Binary representation of 20 is 10100.



Program 4.7

Date: 29/02/2024

Write a PHP program to store the current date/time in a cookie and display the 'last visited on' date and time on the web page upon reopening the same page.

CODE :

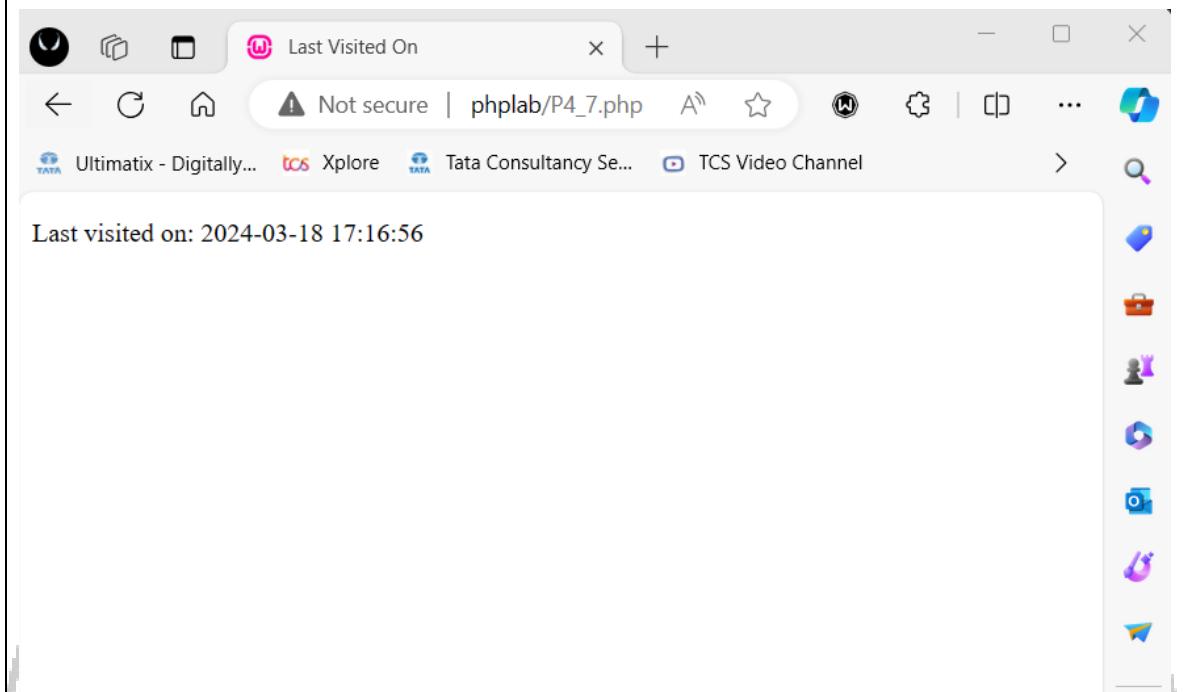
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Last Visited On</title>
</head>
<body>

<?php
// Check if the 'last_visited' cookie is set
if (isset($_COOKIE['last_visited'])) {
    $lastVisited = $_COOKIE['last_visited'];
    echo "<p>Last visited on: $lastVisited</p>";
}

// Store the current date/time in a cookie named 'last_visited'
$currentTime = date("Y-m-d H:i:s");
setcookie('last_visited', $currentTime, time() + (86400 * 30), "/"); // Cookie expires in 30 days
?>

</body>
</html>
```

OUTPUT :



Write a PHP program to create an array and store 10 names in the array. Do the following operations.

- 4.8.1 : Display the contents using each statement.
- 4.8.2 : Display the array in sorted order.
- 4.8.3 : Display the array without the duplicate elements.
- 4.8.4 : Remove the last element and display.
- 4.8.5 : Display the array in reverse order.
- 4.8.6 : Search for an element in the given array.

CODE :

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Array Operations</title>
</head>
<body>
<?php
// Fixed array values
$names = array("Majo", "Anshad", "Nihal", "Hari", "Majo");
echo "<br><b>Contents of the array using each statement:</b><br><hr>";
foreach ($names as $name) {
    echo $name . "<br>";
}
echo "<br>";
?>
<form method="post">
    <label for="search">Enter name to search:</label><br>
    <input type="text" id="search" name="search" required><br>
    <button type="submit">Submit</button>
</form>
<?php

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Get the search element from the form
}
```

```

$searchName = $_POST["search"];
// 4.8.6 : Search for an element in the given array
if (in_array($searchName, $names)) {
    echo "$searchName is found in the array.";
} else {
    echo "$searchName is not found in the array.";
}
// 4.8.2 : Display the array in sorted order
echo "<br><hr><b>Array in sorted order:</b><br><hr>";
sort($names);
foreach ($names as $name) {
    echo $name . "<br>";
}
echo "<br>";
// 4.8.3 : Display the array without duplicate elements
echo "<hr><b>Array without duplicate elements:</b><br><hr>";
$uniqueNames = array_unique($names);
foreach ($uniqueNames as $name) {
    echo $name . "<br>";
}
echo "<br>";
// 4.8.4 : Remove the last element and display
echo "<hr><b>Array after removing the last element:</b><br><hr>";
array_pop($names);
foreach ($names as $name) {
    echo $name . "<br>";
}
echo "<br>";
// 4.8.5 : Display the array in reverse order
echo "<hr><b>Array in reverse order:</b><br><hr>";
$reversedNames = array_reverse($names);
foreach ($reversedNames as $name) {
    echo $name . "<br>";
}
echo "<br>";
}
?>

</body>
</html>

```

OUTPUT :

The screenshot shows a web browser window titled "Array Operations". The address bar indicates the page is "Not secure" and the URL is "phplab/P4_8.php". The browser interface includes standard controls like back, forward, search, and settings.

Contents of the array using each statement:

```
Majo  
Anshad  
Nihal  
Hari  
Majo
```

Enter name to search:

Anshad is found in the array.

Array in sorted order:

```
Anshad  
Hari  
Majo  
Majo  
Nihal
```

Array without duplicate elements:

```
Anshad  
Hari  
Majo  
Nihal
```

Array after removing the last element:

```
Anshad  
Hari  
Majo  
Majo
```

Array in reverse order:

```
Majo  
Majo  
Hari  
Anshad
```

Write the PHP script to send an email.

CODE :

```
<!--Write the PHP script to send an email.
```

In order to send follow below steps:

1) Go to "Manage Your google account ->Security->Less secure app access" and Enable it.

2)Install phpMailer :

```
C:\wamp64\www\phplab> composer require phpmailer/phpmailer -->
```

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Email Sender</title>
</head>
<body>

<h2>Email Sender</h2>

<form method="post">
  <label for="email">Your Email Address:</label><br>
  <input type="email" id="email" name="email" required><br>
  <label for="password">Your Email Password:</label><br>
  <input type="password" id="password" name="password" required><br>
  <label for="to">Recipient's Email Address:</label><br>
  <input type="email" id="to" name="to" required><br>
  <label for="subject">Subject:</label><br>
  <input type="text" id="subject" name="subject" required><br>
  <label for="body">Message Body:</label><br>
  <textarea id="body" name="body" rows="4" required></textarea><br>
  <button type="submit">Send Email</button>
</form>

<?php
```

```
use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\Exception;
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Get input values from the form
    $email = $_POST["email"];
    $password = $_POST["password"];
    $to = $_POST["to"];
    $subject = $_POST["subject"];
    $body = $_POST["body"];

    // Include PHPMailer autoloader
    require 'vendor/autoload.php';

    // Create a new PHPMailer instance
    $mail = new PHPMailer();
    // SMTP configuration for Gmail
    $mail->isSMTP();
    $mail->Host = 'smtp.gmail.com';
    $mail->SMTPAuth = true;
    $mail->Username = $email; // Sender's email address
    $mail->Password = $password; // Sender's email password
    $mail->SMTPSecure = 'tls'; // Enable TLS encryption
    $mail->Port = 587; // TCP port to connect to
    // Sender and recipient
    $mail->setFrom($email);
    $mail->addAddress($to);

    // Email subject and body
    $mail->Subject = $subject;
    $mail->Body = $body;

    // Send email
    if($mail->send()) {
        echo 'Email sent successfully!';
    } else {
        echo 'Failed to send email. Error: ' . $mail->ErrorInfo;
    }
}
?>
</body>
</html>
```

OUTPUT :

The screenshot displays two windows side-by-side. The left window is a web browser titled 'Email Sender' showing a form to send an email. The right window is a Gmail inbox showing the sent email.

Email Sender Form:

- Your Email Address: mca2336@rajagiri.edu
- Your Email Password: (redacted)
- Recipient's Email Address: anshaddevagiri817@gmail.com
- Subject: Invitation
- Message Body:

Hello anshad,
This is an invitaion
from ANS.LTD
- Send Email

Email sent successfully!

Gmail Inbox:

- Compose button
- Inbox (2,674 messages)
- Starred, Snoozed, Sent, Drafts (7), More
- Labels
- Search bar: Search mail
- Message preview: Invitation (Inbox)
 - From: mca2336@rajagiri.edu (1 minute ago)
 - To: anshaddevagiri817@gmail.com
 - Subject: Invitation
 - Body: Hello anshad,
This is an invitaion from ANS.LTD
 - Reply, Forward, Delete buttons



Program 4.10

Date: 10/03/2024

Write a PHP program to implement MySQL databases with CRUD operations.

CODE :

```
<?php
// Database configuration
$servername = "localhost";
$username = "root";
$password = "";
$database = "mydb";

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

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Create table
$sql_create_table = "CREATE TABLE IF NOT EXISTS users (
    id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    firstname VARCHAR(30) NOT NULL,
    lastname VARCHAR(30) NOT NULL,
    email VARCHAR(50),
    reg_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON
    UPDATE CURRENT_TIMESTAMP
)";

if ($conn->query($sql_create_table) === TRUE) {
    echo "Table 'users' created successfully.<br>";
} else {
    echo "Error creating table: " . $conn->error . "<br>";
}

// Insert data
if (isset($_POST['submit'])) {
```

```

$firstname = $_POST['firstname'];
$lastname = $_POST['lastname'];
$email = $_POST['email'];

$sql_insert = "INSERT INTO users (firstname, lastname, email) VALUES
('{$firstname}', '{$lastname}', '{$email}')";

if ($conn->query($sql_insert) === TRUE) {
    echo "New record created successfully.<br>";
} else {
    echo "Error: " . $sql_insert . "<br>" . $conn->error;
}

// Read data
$sql_read = "SELECT id, firstname, lastname, email FROM users";
$result = $conn->query($sql_read);

if ($result->num_rows > 0) {
    echo "<h2>Users:</h2>";
    echo "<table border='1'>";
    echo "    <tr><th>ID</th><th>First Name</th><th>Last Name</th><th>Email</th></tr>";
    while ($row = $result->fetch_assoc()) {
        echo "    <tr><td>" . $row["id"] . "</td><td>" . $row["firstname"] . "</td><td>" . $row["lastname"] . "</td><td>" . $row["email"] . "</td></tr>";
    }
    echo "</table>";
} else {
    echo "0 results";
}

// Update data
if (isset($_POST['update'])) {
    $id = $_POST['id'];
    $firstname = $_POST['firstname'];
    $lastname = $_POST['lastname'];
    $email = $_POST['email'];

    $sql_update = "UPDATE users SET firstname='{$firstname}', lastname='{$lastname}', email='{$email}' WHERE id={$id}";

    if ($conn->query($sql_update) === TRUE) {
}

```

```

        echo "Record updated successfully.<br>";
    } else {
        echo "Error updating record: " . $conn->error;
    }
}

// Delete data
if (isset($_POST['delete'])) {
    $id = $_POST['id'];

    $sql_delete = "DELETE FROM users WHERE id=$id";

    if ($conn->query($sql_delete) === TRUE) {
        echo "Record deleted successfully.<br>";
    } else {
        echo "Error deleting record: " . $conn->error;
    }
}

$conn->close();
?>

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

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

<body>
    <h2>Add New User</h2>
    <form method="post" action="">
        First Name: <input type="text" name="firstname"><br>
        Last Name: <input type="text" name="lastname"><br>
        Email: <input type="email" name="email"><br>
        <input type="submit" name="submit" value="Add User">
    </form>

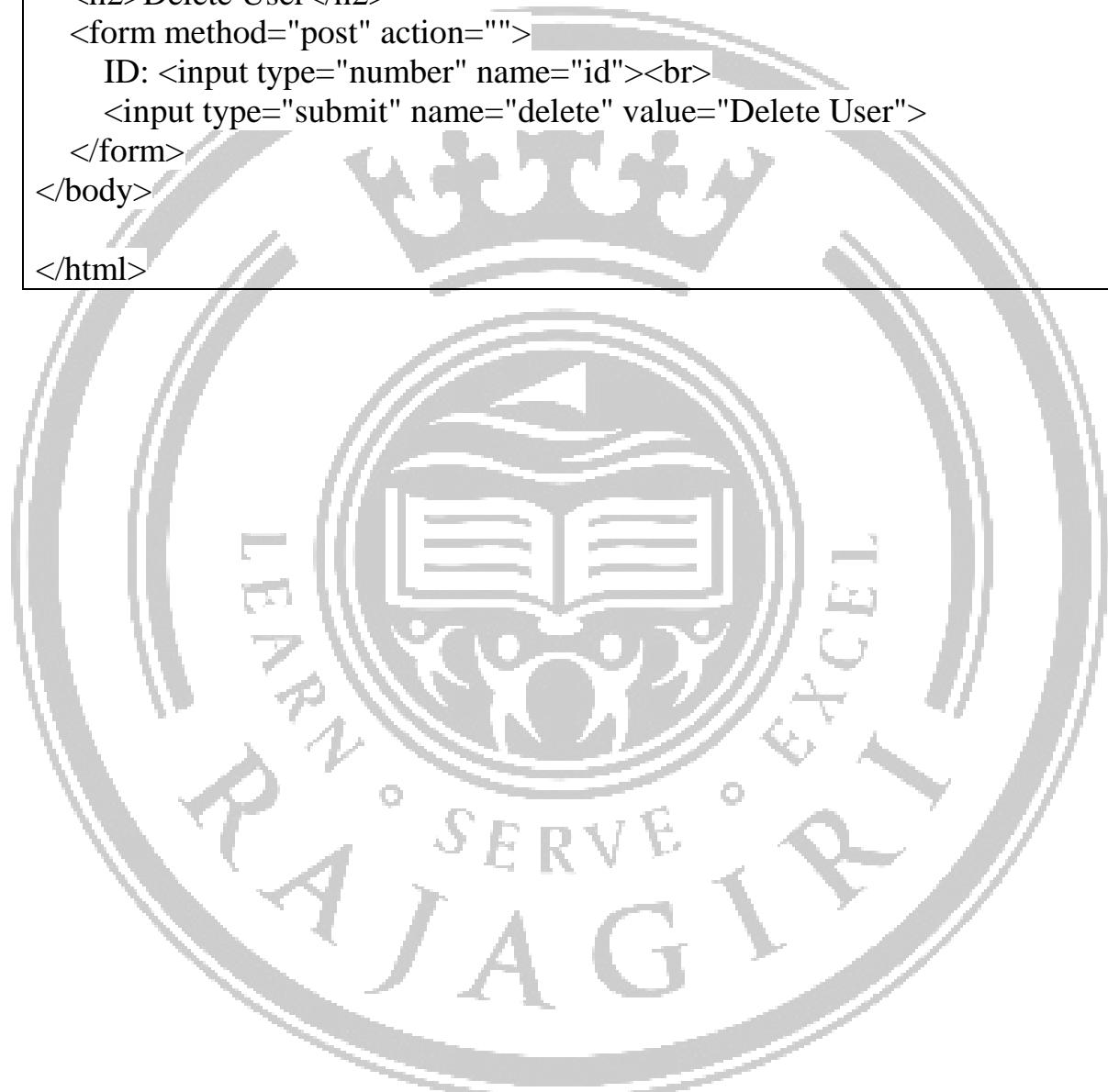
    <h2>Update User</h2>
    <form method="post" action="">

```

```
ID: <input type="number" name="id"><br>
First Name: <input type="text" name="firstname"><br>
Last Name: <input type="text" name="lastname"><br>
Email: <input type="email" name="email"><br>
<input type="submit" name="update" value="Update User">
</form>

<h2>Delete User</h2>
<form method="post" action="">
    ID: <input type="number" name="id"><br>
    <input type="submit" name="delete" value="Delete User">
</form>
</body>

</html>
```



OUTPUT :

The screenshot shows a web browser window with the title "CRUD Operations". The address bar indicates the URL is "localhost / MySQL /" and the page is "phplab/P4_10.php". The browser interface includes standard navigation buttons (back, forward, search, etc.) and a sidebar with various icons.

The main content area displays the following messages:

- Table 'users' created successfully.
- New record created successfully.

Users:

ID	First Name	Last Name	Email
1	Muhammad	Anshad P A	marshu817@gmail.com
2	Nihal	Muhammad	niga123@gmail.com

Add New User

Form fields for adding a new user:

- First Name:
- Last Name:
- Email:

Update User

Form fields for updating an existing user (ID 1):

- ID:
- First Name:
- Last Name:
- Email:

Delete User

Form field for deleting a user (ID 2):

- ID:

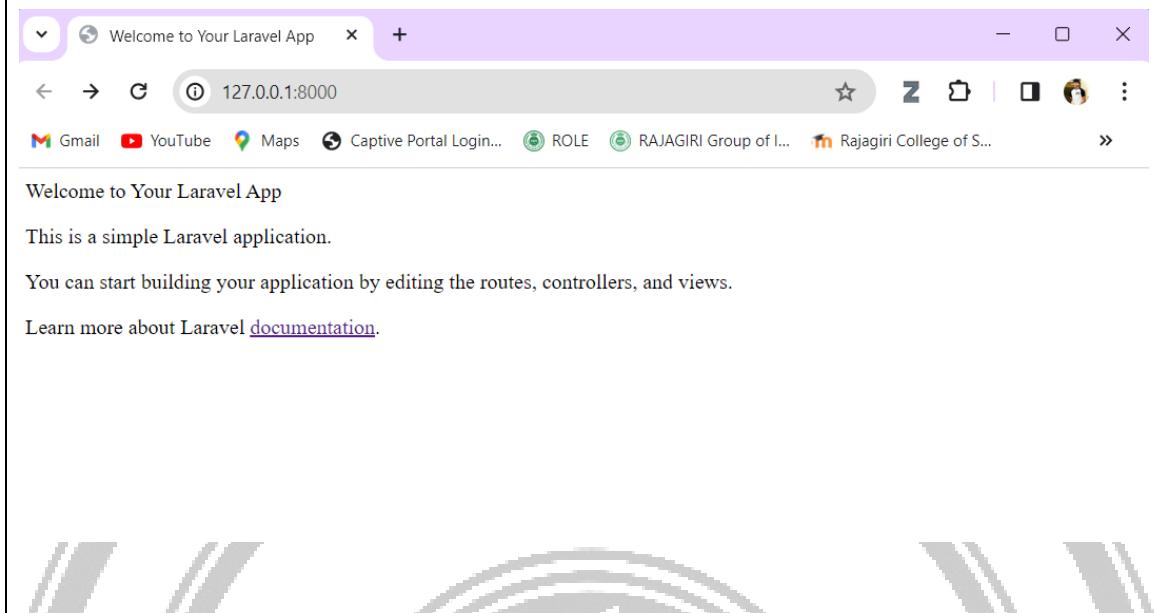
A decorative footer graphic of a smiling face is visible at the bottom of the page.



Program to illustrate the workings of a simple Laravel web application.**welcome.blade.php :**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Welcome to Your Laravel App</title>
    <link href="{{ asset('css/app.css') }}" rel="stylesheet">
</head>
<body>
    <div class="container">
        <div class="row justify-content-center">
            <div class="col-md-8 mt-5">
                <div class="card">
                    <div class="card-header">Welcome to Your Laravel App</div>
                    <div class="card-body">
                        <p>This is a simple Laravel application.</p>
                        <p>You can start building your application by editing the routes, controllers, and views.</p>
                        <p>Learn more about Laravel <a href="https://laravel.com/docs">documentation</a>.</p>
                    </div>
                </div>
            </div>
        </div>
    </div>
</body>
</html>
```

OUTPUT :



Program to demonstrate the implementation of the Bootstrap framework in Laravel.**welcome.blade.php :**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH" crossorigin="anonymous">
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jIeHz" crossorigin="anonymous"></script>
</head>
<body>
<form>
    <div class="mb-3">
        <label for="exampleInputEmail1" class="form-label">Email address</label>
        <input type="email" class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp">
        <div id="emailHelp" class="form-text">We'll never share your email with anyone else.</div>
    </div>
    <div class="mb-3">
        <label for="exampleInputPassword1" class="form-label">Password</label>
        <input type="password" class="form-control">
    </div>
</form>

```

```
id="exampleInputPassword1">
</div>
<div class="mb-3 form-check">
  <input type="checkbox" class="form-check-input" id="exampleCheck1">
    <label class="form-check-label" for="exampleCheck1">Check me out</label>
  </div>
  <button type="submit" class="btn btn-primary">Submit</button>
</form>
</body>
</html>
```



OUTPUT :

Program 5.3

Date: 12/03/2024

Program to illustrate the creation of a Laravel web application with all form validations.

web.php :

```
<?php  
  
use Illuminate\Support\Facades\Route;  
use App\Http\Controllers\RegistrationController;  
  
Route::get('/', function () {  
    return view('registration');  
});  
  
Route::get('/register', [RegistrationController::class,  
    'showRegistrationForm']);  
Route::post('/register', [RegistrationController::class, 'processRegistration']);
```

RegistrationController.php :

```
<?php  
  
namespace App\Http\Controllers;  
  
use Illuminate\Http\Request;  
use App\Models\User; // Assuming the User model exists  
  
class RegistrationController extends Controller  
{  
    public function showRegistrationForm()  
    {  
        return view('registration');  
    }  
}
```

```

public function processRegistration(Request $request)
{
    $validatedData = $request->validate([
        'name' => 'required|string|max:255',
        'email' => 'required|email|unique:users,email',
        'password' => 'required|string|min:8|confirmed',
    ]);

    // Create a new user record
    $user = new User();
    $user->name = $request->name;
    $user->email = $request->email;
    $user->password = bcrypt($request->password); // Hash the password
    $user->save();

    // Redirect the user to a success page or perform any other action
    return redirect()->route('registration.success');
}
}

```

registration.blade.php

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Registration Form</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
          rel="stylesheet" integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6h
          W+ALEwIH" crossorigin="anonymous">
        <script
            src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.mi
            n.js" integrity="sha384-YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7
            N6jIeHz" crossorigin="anonymous"></script>

    </head>
    <body>

```

```
<h1>Registration Form</h1>
<form method="POST" action="/register">
    @csrf
    <div class="mb-3">
        <label for="name" class="form-label">Name:</label>
        <input type="text" id="name" name="name"
value="{{ old('name') }}"
        @error('name')
            <div>{{ $message }}</div>
        @enderror
    </div>
    <div class="mb-3">
        <label for="email">Email:</label>
        <input type="email" class="form-label" id="email" name="email"
value="{{ old('email') }}"
        @error('email')
            <div>{{ $message }}</div>
        @enderror
    </div>
    <div class="mb-3">
        <label for="password">Password:</label>
        <input type="password" class="form-label" id="password"
name="password"
        @error('password')
            <div>{{ $message }}</div>
        @enderror
    </div>
    <div class="mb-3">
        <label for="password_confirmation">Confirm Password:</label>
        <input type="password" class="form-label"
id="password_confirmation" name="password_confirmation">
    </div>
    <button type="submit" class="btn btn-primary">Register</button>
</form>
</body>
</html>
```

OUTPUT :

Registration Form

Name:

The name field is required.

Email:

Password:

The password field must be at least 8 characters.

Confirm Password:

Register

Program 5.4

Date: 12/03/2024

Create a Login page using the Laravel framework.

create_users_table.php :

```
<?php
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;
return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('users', function (Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->string('email')->unique();
            $table->timestamp('email_verified_at')->nullable();
            $table->string('password');
            $table->rememberToken();
            $table->timestamps();
        });
    }
    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists('users');
    }
};
```



SampleController.php :

```
<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;
use Hash;
use Session;
use App\Models\User;
use Illuminate\Support\Facades\Auth;
class SampleController extends Controller
{
function index()
{
return view('login');
}
function registration()
{
return view('registration');
}
function validate_registration(Request $request)
{
$request->validate([
'name' => 'required',
'email' => 'required|email|unique:users',
'password' => 'required|min:6'
]);
$data = $request->all();
User::create([
'name' => $data['name'],
'email' => $data['email'],
'password' => Hash::make($data['password'])
]);
return redirect('login')->with('success', 'Registration Completed, now you can
login');
}
function validate_login(Request $request)
{
$request->validate([
'email' => 'required',
'password' => 'required'
]);
$credentials = $request->only('email', 'password');
```

```

if(Auth::attempt($credentials))
{
return redirect('dashboard');
}
return redirect('login')->with('success', 'Login details are not valid');
}
function dashboard()
{
if(Auth::check())
{
return view('dashboard');
}
return redirect('login')->with('success', 'you are not allowed to access');
}
function logout()
{
Session::flush();
Auth::logout();
return Redirect('login');
}
}

```

main.blade.php

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Custom Login Registration</title>
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.1/dist/css/bootstrap.min.css"
"
rel="stylesheet">
</head>
<body>
<nav class="navbar navbar-light navbar-expand-lg mb-5"
style="background-color:
#e3f2fd;">
<div class="container">
<a class="navbar-brand mr-auto" href="#">Custom Login Registration</a>

```

```

<button class="navbar-toggler" type="button" data-bs-toggle="collapse"
data-bs-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-
label="Toggle
navigation">
<span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse" id="navbarNav">
<ul class="navbar-nav">
@guest
<li class="nav-item">
<a class="nav-link" href="{{ route('login') }}>Login</a>
</li>
<li class="nav-item">
<a class="nav-link" href="{{ route('registration') }}>Register</a>
</li>
@else
<li class="nav-item">
<a class="nav-link" href="{{ route('logout') }}>Logout</a>
</li>
@endguest
</ul>
</div>
</div>
</nav>
<div class="container mt-5">
@yield('content')
</div>
</body>
</html>

```

registration.blade.php :

```

@extends('main')
@section('content')
<div class="row justify-content-center">
<div class="col-md-4">
<div class="card">
<div class="card-header">Registration</div>
<div class="card-body">
<form action="{{ route('sample.validate_registration') }}" method="POST">
@csrf

```

```

<div class="form-group mb-3">
<input type="text" name="name" class="form-control" placeholder="Name"
/>
@if($errors->has('name'))
<span class="text-danger">{ $errors->first('name') }</span>
@endif
</div>
<div class="form-group mb-3">
<input type="text" name="email" class="form-control" placeholder="Email
Address" />
@if($errors->has('email'))
<span class="text-danger">{ $errors->first('email') }</span>
@endif
</div>
<div class="form-group mb-3">
<input type="password" name="password" class="form-control"
placeholder="Password" />
@if($errors->has('password'))
<span class="text-danger">{ $errors->first('password') }</span>
@endif
</div>
<div class="d-grid mx-auto">
<button type="submit" class="btn btn-dark btn-block">Register</button>
</div>
</form>
</div>
</div>
@endsection('content')

```

login.blade.php

```

@extends('main')
@section('content')
@if($message = Session::get('success'))


{{ $message }}


@endif

```

```

<div class="card-header">Login</div>
<div class="card-body">
<form action="{{ route('sample.validate_login') }}" method="post">
@csrf
<div class="form-group mb-3">
<input type="text" name="email" class="form-control" placeholder="Email" />
@if($errors->has('email'))
<span class="text-danger">{{ $errors->first('email') }}</span>
@endif
</div>
<div class="form-group mb-3">
<input type="password" name="password" class="form-control" placeholder="Password" />
@if($errors->has('password'))
<span class="text-danger">{{ $errors->first('password') }}</span>
@endif
</div>
<div class="d-grid mx-auto">
<button type="submit" class="btn btn-dark btn-block">Login</button>
</div>
</form>
</div>
</div>
</div>
</div>
@endsection('content')

```

dashboard.blade.php :

```

@extends('main')
@section('content')


Dashboard



You are logged in to Laravel 10 Custom Login Registration Application.


@endsection

```

web.php

```
@extends('main')
@section('content')
<div class="card">
<div class="card-header">Dashboard</div>
<div class="card-body">
You are logged in to Laravel 10 Custom Login Registration Application.
</div>
</div>
@endsection('content')
```



OUTPUT :

Custom Login Registration Login Register

Registration

Sweaba

mca2254@rajagiri.edu

.....

Register

Custom Login Registration Login Register

Registration Completed, now you can login

Login

mca2254@rajagiri.edu

.....

Login

Custom Login Registration Logout

Dashboard

You are logged in to Laravel 10 Custom Login Registration Application.

Program to implement session handling using the Laravel framework.**registration.blade.php :**

```
<!-- registration.blade.php -->
@extends('master')
@section('content')


<form method="post" action=" {{ url('user-store') }} ">
<div class="card shadow mb-4">
<div class="card-header bg-success pt-2">
<div class="card-title font-weight-bold text-white text-center">User
Registration </div>
</div>
<div class="card-body">
@if(Session::has('success'))
<div class="alert alert-success">
{{ Session::get('success') }}
@endif
Session::forget('success');
<endphp
</div>
<endif>
<div class="form-group">
<label for="first_name"> First Name </label>
<input type="text" name="first_name" id="first_name" class="form
control" placeholder="Enter First Name" value="{{ old('first_name') }}"/>
{!! $errors->first('first_name', '<small class="text
danger">:message</small>') !!}
</div>
<div class="form-group">
<label for="last_name"> Last Name </label>
<input type="text" name="last_name" id="last_name" class="form-control"
placeholder="Enter Last Name" value="{{ old('last_name') }}"/>
{!! $errors->first('last_name', '<small class="text-danger">:message


```

```

        </small>') !!}
    </div>
    <div class="form-group">
        <label for="email"> E-mail </label>
        <input type="text" name="email" id="email" class="form-control"
        placeholder="Enter E-mail" value="{{ old('email') }}"/>
        {{ !! $errors->first('email', '<small class="text-danger">:message</small>')}
        !!}
    </div>
    <div class="form-group">
        <label for="password"> Password </label>
        <input type="password" name="password" id="password" class="form-control"
        placeholder="Enter Password" value="{{ old('password') }}"/>
        {{ !! $errors->first('password', '<small class="text-danger">:message</small>')}
        !!}
    </div>
    <div class="form-group">
        <label for="confirm_password"> Confirm Password </label>
        <input type="password" name="confirm_password"
        id="confirm_password" class="form-control" placeholder="Confirm
        Password" value="{{ old('confirm_password') }}"/>
        {{ !! $errors->first('confirm_password', '<small class="text-danger">:message</small>')}
        !!}
    </div>
    <div class="form-group">
        <label for="phone"> Phone </label>
        <input type="phone" name="phone" id="phone" class="form-control"
        placeholder="Enter Phone" value="{{ old('phone') }}"/>
        {{ !! $errors->first('phone', '<small class="text-danger">:message</small>')}
        !!}
    </div>
</div>
<div class="card-footer d-inline-block">
    <button type="submit" class="btn btn-success"> Register </button>
    <p class="float-right mt-2"> Already have an account? <a href="{{ url('user
    login') }}" class="text-success"> Login </a> </p>
</div>
@csrf
</div>
</form>
</div>
</div>

```

```
</div>
@endsection
```

login.blade.php :

```
<!-- login.blade.php -->
@extends('master')
@section('content')
<div class="container mt-3">
<div class="row">
<div class="col-xl-6 col-lg-6 col-md-6 col-sm-12 col-12 m-auto">
<form method="post" action="{{ url('login') }}>
<div class="card shadow">
<div class="card-header bg-success pt-2">
<div class="card-title font-weight-bold text-white text-center"> User Login
</div>
</div>
<div class="card-body">
@if(Session::has('success'))
<div class="alert alert-success">
{{ Session::get('success') }}
@endif
Session::forget('success');
@endphp
</div>
</div>
<div class="form-group">
<label for="email"> E-mail </label>
<input type="text" name="email" id="email" class="form-control"
placeholder="Enter E-mail" value="{{ old('email') }}"/>
{!! $errors->first('email', '<small class="text-danger">:message</small>')
!!}
</div>
<div class="form-group">
<label for="password"> Password </label>
<input type="password" name="password" id="password" class="form-control"
placeholder="Enter Password" value="{{ old('password') }}"/>
{!! $errors->first('password', '<small class="text-danger">:message</small>')
!!}
</div>
</div>
<div class="card-footer d-inline-block">
<button type="submit" class="btn btn-success"> Login </button>
```

```

<p class="float-right mt-2"> Don't have an account? <a href="{{ url('user
registration') }}" class="text-success"> Register </a> </p>
</div>
@csrf
</div>
</form>
</div>
</div>
</div>
@endsection

```

dashboard.blade.php :

```

@extends('master')
@section('content')
<nav class="navbar navbar-expand-lg navbar-dark bg-primary">
<a class="navbar-brand" href="#">User Registration and Login</a>
<button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarColor01" aria-controls="navbarColor01" aria-
expanded="false" aria
label="Toggle navigation">
<span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse" id="navbarColor01">
<ul class="navbar-nav ml-auto">
<li class="nav-item">
<a class="nav-link text-white"> Welcome:
{{ ucfirst(Auth()->user()->first_name) }}</a>
</li>
<li class="nav-item">
<a class="nav-link" href="{{ url('logout') }}> Logout </a>
</li>
</ul>
</div>
</nav>
@endsection

```

web.php :

```

<?php
use Illuminate\Support\Facades\Route;
Route::get('/', function () {

```

```
return view('welcome');
});

Route::get('user-registration', 'App\Http\Controllers\UserController@index');
Route::post('user-store',
'App\Http\Controllers\UserController@userPostRegistration');
Route::get('user-login',
'App\Http\Controllers\UserController@userLoginIndex');
Route::post('login', 'App\Http\Controllers\UserController@userPostLogin');
Route::get('dashboard', 'App\Http\Controllers\UserController@dashboard');
Route::get('logout', 'App\Http\Controllers\UserController@logout');
});
```



OUTPUT :

The screenshot displays a user registration and login interface. At the top, there is a blue header bar with the text "User Registration and Login" on the left and "Welcome: Sweaba Logout" on the right. Below the header, there is a "User Login" form.

Successful Login (Top):

- E-mail: mca2254@rajagiri.edu
- Password: (redacted)
- Buttons: "Login" (green) and "Don't have an account? Register"

Unsuccessful Login (Bottom):

- E-mail: Enter E-mail
- Password: Enter Password
- Message: Opps! You do not have access (red background)
- Buttons: "Login" (green) and "Don't have an account? Register"



Program 5.6

Date: 16/03/2024

Write a program to execute database CRUD operations in the Laravel framework.

create_products_table.php:

```
<?php
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;
return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('products', function (Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->text('detail');
            $table->timestamps();
        });
    }
    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists('products');
    }
};
```

Product.php :

```
<?php
namespace App\Models;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
class Product extends Model
{
```

```
use HasFactory;
protected $fillable=['name','detail'];
}
```

ProductController.php :

```
<?php
namespace App\Http\Controllers;
use App\Models\Product;
use Illuminate\Http\RedirectResponse;
use Illuminate\Http\Request;
use Illuminate\Http\Response;
use Illuminate\View\View;
class ProductController extends Controller
{
/**
 * Display a listing of the resource.
 */
public function index(): View
{
$products = Product::latest()->paginate(5);
return view('products.index',compact('products'))
->with('i', (request()->input('page', 1) - 1) * 5);
}
/**
 * Show the form for creating a new resource.
 */
public function create(): View
{
return view('products.create');
}
/**
 * Store a newly created resource in storage.
 */
public function store(Request $request): RedirectResponse
{
$request->validate([
'name' => 'required',
'detail' => 'required',
]);
Product::create($request->all());
return redirect()->route('products.index')
```

```

->with('success','Product created successfully.');
}
/***
 * Display the specified resource.
*/
public function show(Product $product): View
{
    return view('products.show',compact('product'));
}
/***
 * Show the form for editing the specified resource.
*/
public function edit(Product $product): View
{
    return view('products.edit',compact('product'));
}
/***
 * Update the specified resource in storage.
*/
public function update(Request $request, Product $product): RedirectResponse
{
    $request->validate([
        'name' => 'required',
        'detail' => 'required',
    ]);
    $product->update($request->all());
    return redirect()->route('products.index')
        ->with('success','Product updated successfully');
}
/***
 * Remove the specified resource from storage.
*/
public function destroy(Product $product): RedirectResponse
{
    $product->delete();
    return redirect()->route('products.index')
        ->with('success','Product deleted successfully');
}
}

```

layout.blade.php :

```
<!DOCTYPE html>
<html>
<head>
<title>Laravel 10 CRUD Application</title>
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
"
rel="stylesheet">
</head>
<body>
<div class="container">
@yield('content')
</div>
</body>
</html>
```

index.blade.php :

```
@extends('products.layout')
@section('content')
<div class="row">
<div class="col-lg-12 margin-tb">
<div class="pull-left">
<h2>Laravel 10 CRUD </h2>
</div>
<div class="pull-right">
<a class="btn btn-success" href="{{ route('products.create') }}> Create New
Product</a>
</div>
</div>
</div>
@if ($message = Session::get('success'))
<div class="alert alert-success">
<p>{{ $message }}</p>
</div>
@endif
<table class="table table-bordered">
<tr>
<th>No</th>
<th>Name</th>
<th>Details</th>
```

```

<th width="280px">Action</th>
</tr>
@foreach ($products as $product)
<tr>
<td>{{ ++$i }}</td>
<td>{{ $product->name }}</td>
<td>{{ $product->detail }}</td>
<td>
<form action="{{ route('products.destroy', $product->id) }}" method="POST">
<a class="btn btn-info" href="{{ route('products.show', $product->id) }}>Show</a>
<a class="btn btn-primary" href="{{ route('products.edit', $product->id) }}>Edit</a>
@csrf
@method('DELETE')
<button type="submit" class="btn btn-danger">Delete</button>
</form>
</td>
</tr>
@endforeach
</table>
{!! $products->links() !!}
@endsection

```

create.blade.php :

```

@extends('products.layout')
@section('content')


## Add New Product



Back


@if ($errors->any())


```

```

<strong>Whoops!</strong> There were some problems with your
input.<br><br>
<ul>
@foreach ($errors->all() as $error)
<li>{{ $error }}</li>
@endforeach
</ul>
</div>
@endif
<form action="{{ route('products.store') }}" method="POST">
@csrf
<div class="row">
<div class="col-xs-12 col-sm-12 col-md-12">
<div class="form-group">
<strong>Name:</strong>
<input type="text" name="name" class="form-control"
placeholder="Name">
</div>
</div>
<div class="col-xs-12 col-sm-12 col-md-12">
<div class="form-group">
<strong>Detail:</strong>
<textarea class="form-control" style="height:150px" name="detail"
placeholder="Detail"></textarea>
</div>
</div>
<div class="col-xs-12 col-sm-12 col-md-12 text-center">
<button type="submit" class="btn btn-primary">Submit</button>
</div>
</div>
</form>
@endsection

```

edit.blade.php :

```

@extends('products.layout')
@section('content')
<div class="row">
<div class="col-lg-12 margin-tb">
<div class="pull-left">
<h2>Edit Product</h2>
</div>

```

```

<div class="pull-right">
<a class="btn btn-primary" href="{{ route('products.index') }}> Back</a>
</div>
</div>
</div>
@if ($errors->any())
<div class="alert alert-danger">
<strong>Whoops!</strong> There were some problems with your
input.<br><br>
<ul>
@foreach ($errors->all() as $error)
<li>{{ $error }}</li>
@endforeach
</ul>
</div>
@endif
<form action="{{ route('products.update',$product->id) }}"
method="POST">
@csrf
@method('PUT')
<div class="row">
<div class="col-xs-12 col-sm-12 col-md-12">
<div class="form-group">
<strong>Name:</strong>
<input type="text" name="name" value="{{ $product->name }}"
class="form-control" placeholder="Name">
</div>
</div>
<div class="col-xs-12 col-sm-12 col-md-12">
<div class="form-group">
<strong>Detail:</strong>
<textarea class="form-control" style="height:150px" name="detail"
placeholder="Detail">{{ $product->detail }}

```

show.blade.php :

```
@extends('products.layout')  
@section('content')  
<div class="row">  
  <div class="col-lg-12 margin-tb">  
    <div class="pull-left">  
      <h2> Show Product</h2>  
    </div>  
    <div class="pull-right">  
      <a class="btn btn-primary" href="{{ route('products.index') }}> Back</a>  
    </div>  
  </div>  
  <div class="row">  
    <div class="col-xs-12 col-sm-12 col-md-12">  
      <div class="form-group">  
        <strong>Name:</strong>  
        {{ $product->name }}  
      </div>  
    </div>  
    <div class="col-xs-12 col-sm-12 col-md-12">  
      <div class="form-group">  
        <strong>Details:</strong>  
        {{ $product->detail }}  
      </div>  
    </div>  
  </div>  
@endsection
```

web.php :

```
<?php  
use Illuminate\Support\Facades\Route;  
use App\Http\Controllers\ProductController;  
Route::get('/', function () {  
    return view('welcome');  
});  
Route::resource('products', ProductController::class);
```

OUTPUT :

Add New Product

[Back](#)

Name:
Jeans

Detail:
Clothing

[Submit](#)

Laravel 10 CRUD

[Create New Product](#)

Product created successfully.

No	Name	Details	Action
1	Jeans	Clothing	Show Edit Delete
2	Watch	Accessories	Show Edit Delete
3	Smart Phone	Electronics	Show Edit Delete

Show Product

[Back](#)

Name: Jeans
Details: Clothing

Edit Product

[Back](#)

Name:
T-Shirts & Jeans

Detail:
Clothing

[Submit](#)

Create a simple Laravel Web Application, including:**5.7.1.** A registration form with all form validations.**5.7.2.** Perform CRUD operations using migrations.**register.blade.php :**

```
@extends('layouts.app')
@section('content')
<main class="sm:container sm:mx-auto sm:max-w-lg sm:mt-10">
<div class="flex">
<div class="w-full">
<section class="flex flex-col break-words bg-white sm:border-1 sm:rounded-md
sm:shadow-sm sm:shadow-lg">
<header class="font-semibold bg-gray-200 text-gray-700 py-5 px-6 sm:py-6
sm:px-8 sm:rounded-t-md">
{{ __('Register') }}
</header>
<form class="w-full px-6 space-y-6 sm:px-10 sm:space-y-8"
method="POST"
action="{{ route('register') }}"
@csrf
<div class="flex flex-wrap">
<label for="name" class="block text-gray-700 text-sm font-bold mb-2
sm:mb-4">
{{ __('Name') }}:
</label>
<input id="name" type="text" class="form-input w-full
@error('name') border-red-500 @enderror"
name="name" value="{{ old('name') }}" required autocomplete="name"
autofocus>
@error('name')
<p class="text-red-500 text-xs italic mt-4">
{{ $message }}
</p>
@enderror
</div>
```

```
<div class="flex flex-wrap">
<label for="email" class="block text-gray-700 text-sm font-bold mb-2
sm:mb-
4">
{{__('E-Mail Address')}:
</label>
<input id="email" type="email"
class="form-input w-full @error('email') border-red-500 @enderror"
name="email"
value="{{ old('email') }}" required autocomplete="email">
@error('email')
<p class="text-red-500 text-xs italic mt-4">
{{ $message }}
</p>
@enderror
</div>
<div class="flex flex-wrap">
<label for="password" class="block text-gray-700 text-sm font-bold mb-2
sm:mb-4">
{{__('Password')}:
</label>
<input id="password" type="password"
class="form-input w-full @error('password') border-red-500 @enderror"
name="password"
required autocomplete="new-password">
@error('password')
<p class="text-red-500 text-xs italic mt-4">
{{ $message }}
</p>
@enderror
</div>
<div class="flex flex-wrap">
<label for="password-confirm" class="block text-gray-700 text-sm font-bold
mb-2 sm:mb-4">
{{__('Confirm Password')}:
</label>
<input id="password-confirm" type="password" class="form-input w-full"
name="password_confirmation" required autocomplete="new-password">
</div>
<div class="flex flex-wrap">
<button type="submit"
class="w-full select-none font-bold whitespace-no-wrap p-3 rounded-lg">
```

```

text-base leading-normal no-underline text-gray-100 bg-blue-500 hover:bg-
blue-700 sm:py-
4">
{{__('Register') }}>
</button>
<p class="w-full text-xs text-center text-gray-700 my-6 sm:text-sm sm:my-
8">
{{__('Already have an account?')} }>
<a class="text-blue-500 hover:text-blue-700 no-underline hover:underline" href="{{ route('login') }}">
{{__('Login') }}>
</a>
</p>
</div>
</form>
</section>
</div>
</div>
</main>
@endsection

```

login.blade.php :

```

@extends('layouts.app')
@section('content')
<main class="sm:container sm:mx-auto sm:max-w-lg sm:mt-10">
<div class="flex">
<div class="w-full">
<section class="flex flex-col break-words bg-white sm:border-1 sm:rounded-
md
sm:shadow-sm sm:shadow-lg">
<header class="font-semibold bg-gray-200 text-gray-700 py-5 px-6 sm:py-6
sm:px-8 sm:rounded-t-md">
{{__('Login') }}>
</header>
<form class="w-full px-6 space-y-6 sm:px-10 sm:space-y-8"
method="POST"
action="{{ route('login') }}">
@csrf
<div class="flex flex-wrap">
<label for="email" class="block text-gray-700 text-sm font-bold mb-2
sm:mb-

```

```

4">
{{__('E-Mail Address')}}:
</label>
<input id="email" type="email"
class="form-input w-full @error('email') border-red-500 @enderror"
name="email"
value="{{ old('email') }}" required autocomplete="email" autofocus>
@error('email')
<p class="text-red-500 text-xs italic mt-4">
{{ $message }}
</p>
@endif
</div>
<div class="flex flex-wrap">
<label for="password" class="block text-gray-700 text-sm font-bold mb-2
sm:mb-4">
{{__('Password')}}:
</label>
<input id="password" type="password"
class="form-input w-full @error('password') border-red-500 @enderror"
name="password"
required>
@error('password')
<p class="text-red-500 text-xs italic mt-4">
{{ $message }}
</p>
@endif
</div>
<div class="flex items-center">
<label class="inline-flex items-center text-sm text-gray-700"
for="remember">
<input type="checkbox" name="remember" id="remember" class="form-
checkbox"
{{ old('remember') ? 'checked' : '' }}>
<span class="ml-2">{{__('Remember Me')}}

```

```

</div>
<div class="flex flex-wrap">
<button type="submit"
class="w-full select-none font-bold whitespace-no-wrap p-3 rounded-lg text
base leading-normal no-underline text-gray-100 bg-blue-500 hover:bg-blue-
700 sm:py-4">
{{__('Login')}}
</button>
@if(Route::has('register'))
<p class="w-full text-xs text-center text-gray-700 my-6 sm:text-sm sm:my-
8">
{{__("Don't have an account?")}}
<a class="text-blue-500 hover:text-blue-700 no-underline hover:underline"
href="{{ route('register') }}"
{{__('Register')}}
</p>
@endif
</div>
</form>
</section>
</div>
</div>
</main>
@endsection

```

create_posts_table.php :

```

<?php
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;
class Posts extends Migration
{
/**
 * Run the migrations.
 *
 * @return void
 */
public function up()
{
Schema::create('posts', function (Blueprint $table){

```

```
$table->increments('id');
$table->string('slug');
$table->string('title');
$table->longText('description');
$table->string('image_path');
$table->timestamps();
$table->unsignedBigInteger('user_id');
$table->foreign('user_id')->references('id')->on('users');
});
}
public function down()
{
}
```

Post.php :

```
<?php
namespace App\Models;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use Cviebrock\EloquentSluggable\Sluggable;
class Post extends Model
{
use HasFactory;
use Sluggable;
protected $fillable = ['title', 'slug', 'description', 'image_path', 'user_id'];
public function user()
{
return $this->belongsTo(User::class);
}
public function sluggable(): array
{
return [
'slug' => [
'source' => 'title'
]
];
}
```

PostsController.php :

```
<?php
namespace App\Http\Controllers;
Department of Computer Science, Rajagiri College of Social Sciences
(Autonomous) Page | 124use Illuminate\Http\Request;
use App\Models\Post;
use Cviebrock\EloquentSluggable\Services\SlugService;
class PostsController extends Controller
{
public function __construct()
{
$this->middleware('auth', ['except' => ['index', 'show']]);
}
/**
 * Display a listing of the resource.
 *
 * @return \Illuminate\Http\Response
 */
public function index()
{
return view('blog.index')
->with('posts', Post::orderBy('updated_at', 'DESC')->get());
}
/**
 * Show the form for creating a new resource.
 *
 * @return \Illuminate\Http\Response
 */
public function create()
{
return view('blog.create');
}
/**
 * Store a newly created resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @return \Illuminate\Http\Response
 */
public function store(Request $request)
{
$request->validate([
'title' => 'required',
```

```

'description' => 'required',
'image' => 'required|mimes:jpg,png,jpeg|max:5048'
]);
$newImageName = uniqid() . '-' . $request->title . '-' .
$request->image->extension();
$request->image->move(public_path('images'), $newImageName);
Post::create([
'title' => $request->input('title'),
'description' => $request->input('description'),
'slug' => SlugService::createSlug(Post::class, 'slug', $request->title),
'image_path' => $newImageName,
'user_id' => auth()->user()->id
]);
return redirect('/blog')
->with('message', 'Your post has been added!');
}
/**
 * Display the specified resource.
 *
 * @param string $slug
 * @return \Illuminate\Http\Response
 */
public function show($slug)
{
return view('blog.show')
->with('post', Post::where('slug', $slug)->first());
}
/**
 * Show the form for editing the specified resource.
 *
 * @param string $slug
 * @return \Illuminate\Http\Response
 */
public function edit($slug)
{
return view('blog.edit')
->with('post', Post::where('slug', $slug)->first());
}
/**
 * Update the specified resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @param string $slug

```

```

* @return \Illuminate\Http\Response
*/
public function update(Request $request, $slug)
{
    $request->validate([
        'title' => 'required',
        'description' => 'required',
    ]);
    Post::where('slug', $slug)
        ->update([
            'title' => $request->input('title'),
            'description' => $request->input('description'),
            'slug' => SlugService::createSlug(Post::class, 'slug', $request->title),
            'user_id' => auth()->user()->id
        ]);
    return redirect('/blog')
        ->with('message', 'Your post has been updated!');
}
/** 
 * Remove the specified resource from storage.
 */
* @param int $id
* @return \Illuminate\Http\Response
*/
public function destroy($slug)
{
    $post = Post::where('slug', $slug);
    $post->delete();
    return redirect('/blog')
        ->with('message', 'Your post has been deleted!');
}
}

```

create.blade.php:

```

@extends('layouts.app')
@section('content')
<div class="w-4/5 m-auto text-left">
<div class="py-15">
<h1 class="text-6xl">
Create Post
</h1>

```

```
</div>
</div>
@if ($errors->any())
<div class="w-4/5 m-auto">
<ul>
@foreach ($errors->all() as $error)
<li class="w-1/5 mb-4 text-gray-50 bg-red-700 rounded-2xl py-4">
{ { $error } }
</li>
@endforeach
</ul>
</div>
@endif
<div class="w-4/5 m-auto pt-20">
<form
action="/blog"
method="POST"
enctype="multipart/form-data">
@csrf
<input
type="text"
name="title"
placeholder="Title..."
class="bg-transparent block border-b-2 w-full h-20 text-6xl outline-none">
<textarea
name="description"
placeholder="Description..."
class="py-20 bg-transparent block border-b-2 w-full h-60 text-xl outline
none"></textarea>
<div class="bg-grey-lighter pt-15">
<label class="w-44 flex flex-col items-center px-2 py-3 bg-white-rounded-lg
shadow
lg tracking-wide uppercase border border-blue cursor-pointer">
<span class="mt-2 text-base leading-normal">
Select a file
</span>
<input
type="file"
name="image"
class="hidden">
</label>
</div>
<button
```

```

type="submit"
class="uppercase mt-15 bg-blue-500 text-gray-100 text-lg font-extrabold py-4 px-8 rounded-3xl">
Submit Post
</button>
</form>
</div>
@endsection

```

index.blade.php :

```

@extends('layouts.app')
@section('content')


# Blog Posts


@if (session()->has('message'))


{{ session()->get('message') }}


@endif
@if (Auth::check())


Add post


@endif
@foreach ($posts as $post)


```

```
<div>

</div>
<div>
<h2 class="text-gray-700 font-bold text-5xl pb-4">
{{ $post->title }}
</h2>
<span class="text-gray-500">
By <span class="font-bold italic text-gray-800">{{ $post->user->name }}</span>,
Created on {{ date('jS M Y', strtotime($post->updated_at)) }}
</span>
<p class="text-xl text-gray-700 pt-8 pb-10 leading-8 font-light">
{{ $post->description }}
</p>
<a href="/blog/{{ $post->slug }}" class="uppercase bg-blue-500 text-gray-100 text-lg
font-extrabold py-4 px-8 rounded-3xl">
View Full Post
</a>
@if (isset(Auth::user()->id) && Auth::user()->id == $post->user_id)
<span class="float-right">
<br>
<a
href="/blog/{{ $post->slug }}/edit"
class=" uppercase bg-green-500 text-gray-100 text-lg font-extrabold py-3 px-8
rounded-3xl" style="margin-left: 10px;">
Edit
</a>
</span>
<span class="float-right">
<form
action="/blog/{{ $post->slug }}"
method="POST">
@csrf
@method('delete')
<button
class="uppercase bg-red-500 text-gray-100 font-extrabold py-4 px-8
rounded-3xl"
type="submit">
Delete
</button>
```

```
</form>
</span>
@endif
</div>
</div>
@foreach
@endsection
```

edit.blade.php :

```
@extends('layouts.app')
@section('content')


# Update Post


@if ($errors->any())


@foreach ($errors->all() as $error)
- {{ $error }}

@endforeach


@endif


@csrf
@method('PUT')


```

```
<textarea  
name="description"  
placeholder="Description..."  
class="py-20 bg-transparent block border-b-2 w-full h-60 text-xl outline-  
none">{{ $post->description }}</textarea>  
<button  
type="submit"  
class="uppercase mt-15 bg-blue-500 text-gray-100 text-lg font-extrabold py-  
4 px-8  
rounded-3xl">  
Submit Post  
</button>  
</form>  
</div>  
@endsection
```

show.blade.php :

```
@extends('layouts.app')  
@section('content')  
<div class="w-4/5 m-auto text-left">  
<div class="py-15">  
<h1 class="text-6xl">  
{{ $post->title }}</h1>  
</div>  
</div>  
<div class="w-4/5 m-auto pt-20">  
<span class="text-gray-500">  
By <span class="font-bold italic text-gray-  
800">{{ $post->user->name }}</span>,  
Created on {{ date('jS M Y', strtotime($post->updated_at)) }}  
</span>  
<p class="text-xl text-gray-700 pt-8 pb-10 leading-8 font-light">  
{{ $post->description }}</p>  
</div>  
@endsection
```

OUTPUT:

Create Post

Demo

demo post

SELECT A FILE

SUBMIT POST

Blog Posts

Your post has been added!

ADD POST



Demo

By **Sweaba**, Created on 8th Apr 2023

demo post

VIEW FULL POST

DELETE

EDIT

Laravel

Home Blog Sweaba Logout

Update Post

Demo Post

demo blog post

SUBMIT POST

Blog Posts

Your post has been updated!

[ADD POST](#)



Demo Post
By [Sweaba](#), Created on 8th Apr 2023
demo blog post

[VIEW FULL POST](#) [DELETE](#) [EDIT](#)

Laravel

Home Blog Sweaba Logout

Blog Posts

Your post has been deleted!

[ADD POST](#)



Create a Web Application for Blog Creation with a Login and Registration

form and perform the following operations on blogs using the Laravel framework.

5.8.1. CREATE Blogs.

5.8.2. VIEW Blogs in a Table.

5.8.3. EDIT and UPDATE Blogs.

5.8.4. DELETE Blogs.

create_posts_table.php :

```
<?php
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;
class Posts extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('posts', function (Blueprint $table){
            $table->increments('id');
            $table->string('slug');
            $table->string('title');
            $table->longText('description');
            $table->string('image_path');
            $table->timestamps();
            $table->unsignedBigInteger('user_id');
            $table->foreign('user_id')->references('id')->on('users');
        });
    }
    /**
     * Reverse the migrations.
    }
```

```
/*
public function down()
{
}
}
```

Posts.php :

```
<?php
namespace App\Models;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use Cviebrock\EloquentSluggable\Sluggable;
class Post extends Model
{
    use HasFactory;
    use Sluggable;
    protected $fillable = ['title', 'slug', 'description', 'image_path', 'user_id'];
    public function user()
    {
        return $this->belongsTo(User::class);
    }
    public function sluggable(): array
    {
        return [
            'slug' => [
                'source' => 'title'
            ]
        ];
    }
}
```

PostsController.php :

```
<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;
use App\Models\Post;
use Cviebrock\EloquentSluggable\Services\SlugService;
class PostsController extends Controller
{
```

```

public function __construct()
{
    $this->middleware('auth', ['except' => ['index', 'show']]);
}
/***
 * Display a listing of the resource.
 *
 * @return \Illuminate\Http\Response
 */
public function index()
{
    return view('blog.index')
        ->with('posts', Post::orderBy('updated_at', 'DESC')->get());
}
/***
 * Show the form for creating a new resource.
 *
 * @return \Illuminate\Http\Response
 */
public function create()
{
    return view('blog.create');
}
/***
 * Store a newly created resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @return \Illuminate\Http\Response
 */
public function store(Request $request)
{
    $request->validate([
        'title' => 'required',
        'description' => 'required',
        'image' => 'required|mimes:jpg,png,jpeg|max:5048'
    ]);
    $newImageName = uniqid() . '-' . $request->title . '-' .
    $request->image->extension();
    $request->image->move(public_path('images'), $newImageName);
    Post::create([
        'title' => $request->input('title'),
        'description' => $request->input('description'),
        'slug' => SlugService::createSlug(Post::class, 'slug', $request->title),
    ]);
}

```

```
'image_path' => $newImageName,
'user_id' => auth()->user()->id
]);
return redirect('/blog')
->with('message', 'Your post has been added!');
}
/***
 * Display the specified resource.
 *
 * @param string $slug
 * @return \Illuminate\Http\Response
 */
public function show($slug)
{
return view('blog.show')
->with('post', Post::where('slug', $slug)->first());
}
/***
 * Show the form for editing the specified resource.
 *
 * @param string $slug
 * @return \Illuminate\Http\Response
 */
public function edit($slug)
{
return view('blog.edit')
->with('post', Post::where('slug', $slug)->first());
}
/***
 * Update the specified resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @param string $slug
 * @return \Illuminate\Http\Response
 */
public function update(Request $request, $slug)
{
$request->validate([
'title' => 'required',
'description' => 'required',
]);
Post::where('slug', $slug)
->update([
```

```

'title' => $request->input('title'),
'description' => $request->input('description'),
'slug' => SlugService::createSlug(Post::class, 'slug', $request->title),
'user_id' => auth()->user()->id
]);
return redirect('/blog')
->with('message', 'Your post has been updated!');
}
/***
 * Remove the specified resource from storage.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function destroy($slug)
{
$post = Post::where('slug', $slug);
$post->delete();
return redirect('/blog')
->with('message', 'Your post has been deleted!');
}
}

```

create.blade.php :

```

@extends('layouts.app')
@section('content')


# Create Post


@if ($errors->any())


@foreach ($errors->all() as $error)
- {{ $error }}

@endforeach


```

```
</ul>
</div>
@endif
<div class="w-4/5 m-auto pt-20">
<form
action="/blog"
method="POST"
enctype="multipart/form-data">
@csrf
<input
type="text"
name="title"
placeholder="Title..."
class="bg-transparent block border-b-2 w-full h-20 text-6xl outline-none">
<textarea
name="description"
placeholder="Description..."
class="py-20 bg-transparent block border-b-2 w-full h-60 text-xl outline
none"></textarea>
<div class="bg-grey-lighter pt-15">
<label class="w-44 flex flex-col items-center px-2 py-3 bg-white-rounded-lg
shadow
lg tracking-wide uppercase border border-blue cursor-pointer">
<span class="mt-2 text-base leading-normal">
Select a file
</span>
<input
type="file"
name="image"
class="hidden">
</label>
</div>
<button
type="submit"
class="uppercase mt-15 bg-blue-500 text-gray-100 text-lg font-extrabold py-
4 px-8
rounded-3xl">
Submit Post
</button>
</form>
</div>
@endsection
```

index.blade.php :

```
@extends('layouts.app')
@section('content')
<div class="w-4/5 m-auto text-center">
<div class="py-15 border-b border-gray-200">
<h1 class="text-6xl">
Blog Posts
</h1>
</div>
</div>
@if (session()->has('message'))
<div class="w-4/5 m-auto mt-10 pl-2">
<p class="w-2/6 mb-4 text-gray-50 bg-green-500 rounded-2xl py-4">
{{ session()->get('message') }}
</p>
</div>
@endif
@if (Auth::check())
<div class="pt-15 w-4/5 m-auto">
<a
href="/blog/create"
class="bg-blue-500 uppercase bg-transparent text-gray-100 text-xs font-extrabold py-3 px-5 rounded-3xl">
Create post
</a>
</div>
@endif
@foreach ($posts as $post)
<div class="sm:grid grid-cols-2 gap-20 w-4/5 mx-auto py-15 border-b border-gray-200">
<div>

</div>
<div>
<h2 class="text-gray-700 font-bold text-5xl pb-4">
{{ $post->title }}
</h2>
<span class="text-gray-500">
By <span class="font-bold italic text-gray-800">{{ $post->user->name }}</span>,
```

```

Created on {{ date('jS M Y', strtotime($post->updated_at)) }}
</span>
<p class="text-xl text-gray-700 pt-8 pb-10 leading-8 font-light">
{{ $post->description }}
</p>
<a href="/blog/{{ $post->slug }}" class="uppercase bg-blue-500 text-gray-100 text-lg
font-extrabold py-4 px-8 rounded-3xl">
Keep Reading
</a>
@if (isset(Auth::user()->id) && Auth::user()->id == $post->user_id)
<span class="float-right">
<a
href="/blog/{{ $post->slug }}/edit"
class="text-gray-700 italic hover:text-gray-900 pb-1 border-b-2">
Edit
</a>
</span>
<span class="float-right">
<form
action="/blog/{{ $post->slug }}"
method="POST">
@csrf
@method('delete')
<button
class="text-red-500 pr-3"
type="submit">
Delete
</button>
</form>
</span>
@endif
</div>
</div>
@endforeach
@endsection

```

edit.blade.php :

```

@extends('layouts.app')
@section('content')
<div class="w-4/5 m-auto text-left">
```

```
<div class="py-15">
<h1 class="text-6xl">
Update Post
</h1>
</div>
</div>
@if ($errors->any())
<div class="w-4/5 m-auto">
<ul>
@foreach ($errors->all() as $error)
<li class="w-1/5 mb-4 text-gray-50 bg-red-700 rounded-2xl py-4">
{{ $error }}
</li>
@endforeach
</ul>
</div>
@endif
<div class="w-4/5 m-auto pt-20">
<form
action="/blog/{{ $post->slug }}"
method="POST"
enctype="multipart/form-data">
@csrf
@method('PUT')
<input
type="text"
name="title"
value="{{ $post->title }}"
class="bg-transparent block border-b-2 w-full h-20 text-6xl outline-none">
<textarea
name="description"
placeholder="Description..."
class="py-20 bg-transparent block border-b-2 w-full h-60 text-xl outline-
none">{{ $post->description }}</textarea>
<button
type="submit"
class="uppercase mt-15 bg-blue-500 text-gray-100 text-lg font-extrabold py-
4 px-8
rounded-3xl">
Submit Post
</button>
</form>
```

```
</div>
@endsection
```

show.blade.php :

```
@extends('layouts.app')
@section('content')
<div class="w-4/5 m-auto text-left">
<div class="py-15">
<h1 class="text-6xl">
{{ $post->title }}
</h1>
</div>
</div>
<div class="w-4/5 m-auto pt-20">
<span class="text-gray-500">
By <span class="font-bold italic text-gray-800">{{ $post->user->name }}</span>,
Created on {{ date('jS M Y', strtotime($post->updated_at)) }}
</span>
<p class="text-xl text-gray-700 pt-8 pb-10 leading-8 font-light">
{{ $post->description }}
</p>
</div>
@endsection
```

OUTPUT :

The image displays two identical screenshots of a Laravel application's registration page. Both screenshots are framed by a thick black border.

The top navigation bar is dark blue with the word "Laravel" on the left and "Home" "Blog" "Login" "Register" on the right.

The main content area has a light gray background. It features a "Register" title at the top. Below it is a form with four input fields:

- Name:
- E-Mail Address:
- Password:
- Confirm Password:

Below the form is a large blue "Register" button. At the bottom of the page, there is a link "Already have an account? [Login](#)".

The two screenshots differ only in the color of the "Register" button: the top one is white, and the bottom one is blue.

Laravel

Home Blog Sweaba Logout

Create Post

First

[MyFirstBlogPost](#)

Blog Posts

Your post has been added!

[CREATE POST](#)



First

By [Sweaba](#), Created on 8th Apr 2023

MyFirstBlogPost

[KEEP READING](#)

[Delete](#) [Edit](#)

Laravel

Home Blog Sweaba Logout

Update Post

First

My First Blog Post

Create a simple LARAVEL web application displaying the CAE2 Marks of MSc

CS students, including:

5.9.1. Student Login and Teacher login

5.9.2. Teachers can Add, View, Update, and Delete Students and Marks.

5.9.3. Students can VIEW their marks.

Web.php :

```
Route::get('/', function () {
    return view('welcome');
});

Route::get('/dashboard', function () {
    return view('dashboard');
})->middleware(['auth', 'verified'])->name('dashboard');

Route::middleware('auth')->group(function () {
    Route::get('/profile', [ProfileController::class, 'edit'])->name('profile.edit');
    Route::patch('/profile', [ProfileController::class, 'update'])-
        >name('profile.update');
    Route::delete('/profile', [ProfileController::class, 'destroy'])-
        >name('profile.destroy');
});
Route::get('admin-home',[StudentController::class,'index'])->name('admin-
home');
Route::get('create-student',function(){
    return view('admin.create-student');
})->name('create-student');
Route::post('add-students',[StudentController::class,'store'])->name('add-
students');
Route::get('view-student/{id}',[StudentController::class,'show']);
Route::get('view-edit-student/{id}',[StudentController::class,'edit']);
Route::post('update',[StudentController::class,'update'])->name('update');
Route::get('delete-student/{id}',[StudentController::class,'destroy']);
Route::get('marks',[MarksController::class,'index'])->name('marks');
Route::get('view-marks/{id}',[MarksController::class,'show']);
Route::post('add-marks',[MarksController::class,'store'])->name('add-marks');
Route::get('delete-marks/{id}',[MarksController::class,'destroy']);
Route::get('view-edit-marks/{id}',[MarksController::class,'edit']);
```

```
Route::post('update-marks',[MarksController::class,'update']);
```

MarksController.php :

```
<?php
namespace App\Http\Controllers;
use App\Models\student;
use App\Models\marks;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;
class MarksController extends Controller
{
/**
 * Display a listing of the resource.
 */
public function index()
{
$studentdata=student::get();
$data=DB::table('students')
->join('marks','students.student_id','=','marks.student_id')
->select('students.*','marks.*')
->get();
// dd($data);
return view('admin.admin-marks',compact('studentdata','data'));
}
public function store(Request $request)
{
// dd($request->all());
$request->validate([
'student_id' => 'required',
'dv' => 'required',
'bda' => 'required',
'dip' => 'required',
'bmaar' => 'required',
'wtl' => 'required',
'python' => 'required',
]);
$mark = new marks();
$mark->student_id = $request->student_id;
$mark->dv = $request->dv;
$mark->bda = $request->bda;
$mark->dip = $request->dip;
```

```

$mark->bmaar = $request->bmaar;
$mark->wtl = $request->wtl;
$mark->python = $request->python;
if ($mark->save()) {
    return redirect()
        ->back()
        ->with('success', 'Data has been added successfully!');
} else {
    return redirect()
        ->back()
        ->withInput()
        ->with('error', 'There was a failure while adding ');
}
}
*/
/* Display the specified resource.
*/
public function show(string $id)
{
    $studentdata=student::get();
    $data=DB::table('students')
        ->join('marks','students.student_id','=','marks.student_id')
        ->select('students.*','marks.*')
        ->where('marks.marks_id','=',$id)
        ->first();
    // dd($data);
    return view('admin.admin-view-marks',compact('data'));
}
/*
* Show the form for editing the specified resource.
*/
public function edit(string $id)
{
    $data=marks::where('marks_id','=',$id)->first();
    return view('admin.admin-edit-marks',compact('data'));
}
/*
* Update the specified resource in storage.
*/
public function update(Request $request)
{
    dd($request->all());
    $request->validate([

```

```

'student_id' => 'required',
'dv' => 'required',
'bda' => 'required',
'dip' => 'required',
'bmaar' => 'required',
'wtl' => 'required',
'python' => 'required',
]);
if (marks::where('marks_id', '=', $request->marks_id)->Update([
'dv' => $request->dv,
'bda' => $request->bda,
'dip' => $request->dip,
'bmaar' => $request->bmaar,
'wtl' => $request->wtl,
'python' => $request->python,
])) {
    return redirect()
        ->back()
        ->with('success', 'Data has been added successfully!');
} else {
    return redirect()
        ->back()
        ->withInput()
        ->with('error', 'There was a failure while adding ');
}
}
/***
 * Remove the specified resource from storage.
 */
public function destroy(string $id)
{
    DB::delete('delete from marks where marks_id=?',[ $id]);
    return back();
}
}

```

StudentController.php :

```

<?php
namespace App\Http\Controllers;
use App\Models\student;
use App\Models\marks;

```

```

use Illuminate\Support\Facades\DB;
use Illuminate\Http\Request;
class StudentController extends Controller
{
/**
 * Display a listing of the resource.
 */
public function index()
{
$data = student::get();
return view('admin.admin-home', compact('data'));
}
/**
 * Show the form for creating a new resource.
 */
public function create()
{
//
}
/**
 * Store a newly created resource in storage.
 */
public function store(Request $request)
{
// dd($request->all());
$request->validate([
'name' => 'required',
'email' => 'required',
'regno' => 'required',
]);
$stu = new student();
$stu->name = $request->name;
$stu->email = $request->email;
$stu->regno = $request->regno;
$data=DB::table('students')
->join('marks','students.student_id','=','marks.student_id')
->select('students.*','marks.*')
->get();
if ($stu->save()) {
return redirect()
->back()
->with('success', 'Data has been added successfully!');
} else {
}
}

```

```

return redirect()
->back()
->withInput()
->with('error', 'There was a failure while adding ');
}
}
/***
* Display the specified resource.
*/
public function show(string $id)
{
// $data=DB::table('students')
// ->join('marks','students.student_id','=','marks.student_id')
// ->select('students.*','marks.*')
// ->where('students.student_id','=', $id)
// ->first();
$data = student::where('student_id', '=', $id)->first();
// dd($data);
return view('admin.admin-view-student', compact('data'));
}
/***
* Show the form for editing the specified resource.
*/
public function edit(string $id)
{
$data = student::where('student_id', '=', $id)->first();
// dd($data);
return view('admin.admin-edit-student', compact('data'));
}
/***
* Update the specified resource in storage.
*/
public function update(Request $request)
{
//
$request->validate([
'name' => 'required',
'email' => 'required',
'regno' => 'required',
]);
if (student::where('student_id', '=', $request->student_id)->Update([
'name' => $request->name,
'email' => $request->email,
])
);
}
}

```

```

'regno' => $request->regno,
]) {
return redirect()
->back()
->with('success', 'Data has been added successfully!');
} else {
return redirect()
->back()
->withInput()
->with('error', 'There was a failure while adding ');
}
}
/***
* Remove the specified resource from storage.
*/
public function destroy(string $id)
{
DB::delete('delete from students where student_id=?',[ $id]);
return back();
}
}

```

Mark.php(model) :

```

<?php
namespace App\Models;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
class marks extends Model
{
use HasFactory;
protected $fillable = [
'student_id',
'dv',
'bda',
'dip',
'bmaar',
'wtl',
'python',
];
}

```

Student.php(model) :

```
<?php
namespace App\Models;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
class student extends Model
{
    use HasFactory;
    protected $fillable = [
        'name',
        'email',
        'password',
        'regno',
    ];
}
```

Admin-home.blade.php:

```
@include('admin.admin-header')
<div class="main-panel">
<div class="content-wrapper">
<div class="row">
<div class="col-xl-3 col-sm-6 grid-margin stretch-card">
<div class="card">
<div class="card-body">
<div class="row">
<div class="col-9">
<div class="d-flex align-items-center align-self-start">
<h3 class="mb-0">>{{ $data->count() }}</h3>
</div>
</div>
<div class="col-3">
<div class="icon ">
</div>
</div>
</div>
<h6 class="text-muted font-weight-normal">Students</h6>
</div>
</div>
</div>
```

```

</div>
<div class="row ">
<div class="col-12 grid-margin">
<div class="card">
<div class="card-body">
<h4 class="card-title">Students</h4>
<a href="{{ route('create-student') }}" style="padding-left: 90%">Add</a>
<div class="table-responsive">
<table class="table">
<thead>
<tr>
<th># </th>
<th>Name</th>
<th>Register No</th>
{ {{ -- <th></th> -- }} }
<th>View</th>
<th>Edit</th>
<th>Delete</th>
</tr>
</thead>
<tbody>
@if ($data->count()>0)
<?php
$i=1;
?>
@foreach ($data as $value)
<tr>
<td>{{ $i }}</td>
<td> {{ $value->name }} </td>
<td> {{ $value->regno }} </td>
<td><div class="badge badge-outline-success"><a href="{{ url('view student/'.$value->student_id) }}>View</a></div></td>
<td><div class="badge badge-outline-success"><a href="{{ url('view edit-student/'.$value->student_id) }}>Edit</a></div></td>
<td>
<div class="badge badge-outline-warning"><a href="{{ url('delete student/'.$value->student_id) }}>Delete</a></div>
</td>
</tr>
<?php
$i++;
?>
@endforeach

```

```

@else
<tr>
<td colspan="6"><center>No Data Found</center></td>
</tr>
@endif
</tbody>
</table>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
@include('admin.admin-footer')

```

Create-student.blade.php :

```

@include('admin.admin-header')
<div class="main-panel">
<div class="content-wrapper">
@if ($errors->any())
<div class="col-sm-12">
<div class="alert alert-warning alert-dismissible fade show" role="alert">
@foreach ($errors->all() as $error)
<span>
<p>{{ $error }}</p>
</span>
@endforeach
<button type="button" class="close" data-dismiss="alert" aria
label="Close">
<span aria-hidden="true">&amptimes</span>
</button>
</div>
</div>
@endif
@if (session('success'))
<div class="col-sm-12">
<div class="alert alert-success alert-dismissible fade show" role="alert">
{{ session('success') }}
<button type="button" class="close" data-dismiss="alert" aria
label="Close">
<span aria-hidden="true">&amptimes</span>

```

```

</button>
</div>
</div>
@endif
@if (session('error'))
<div class="col-sm-12">
<div class="alert alert-danger alert-dismissible fade show" role="alert">
{ { session('error') } }
<button type="button" class="close" data-dismiss="alert" aria-
label="Close">
<span aria-hidden="true">>x</span>
</button>
</div>
</div>
@endif
<div class="col-12 grid-margin stretch-card">
<div class="card">
<div class="card-body">
<h4 class="card-title">Add Studnets</h4>
<form class="forms-sample" method="POST" action="{{ route('add
students') }}">
@csrf
<div class="form-group">
<label for="exampleInputName1">Name</label>
<input type="text" class="form-control" id="exampleInputName1"
name="name"
placeholder="Name">
</div>
<div class="form-group">
<label for="exampleInputName1">Email</label>
<input type="email" class="form-control"
id="exampleInputName1" name="email"
placeholder="Email">
</div>
<div class="form-group">
<label for="exampleInputName1">Register No</label>
<input type="number" class="form-control"
id="exampleInputName1" name="regno"
placeholder="Register No">
</div>
<button type="submit" class="btn btn-primary mr-
2">Submit</button>
<button class="btn btn-dark">Cancel</button>

```

```
</form>
</div>
</div>
</div>
@include('admin.admin-footer')
```



OUTPUT :

The screenshot displays a user interface for managing student data, specifically focusing on marks. The interface is organized into four distinct sections, each with a navigation bar on the left.

- Students List:** Shows a table with one student entry: # 1, Name AJAYKRISHNAN G, Register No 2217002. Actions: View, Edit, Delete.
- Add Student:** Form fields for Name and Email, and a Register No field. Buttons: Submit, Cancel.
- Marks Selection:** A series of dropdown menus for selecting marks across different subjects: Student, Data Visualization, Big Data Analytics, Digital Image Processing, and Business Modelling & Applied Analytics Using R.
- Marks List:** Shows a table with one mark entry: # 1, Name AJAYKRISHNAN G. Actions: View, Edit, Delete.

Program 5.10

Date: 19/03/2024

Create a simple Laravel web application for students, including:

- 5.10.1. Student Registration Form
- 5.10.2. Form Validations.
- 5.10.3. Login for students with session handling.
- 5.10.4. Minimum of two web pages with a standard layout

create users table.php :

```
<?php
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;
return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('users', function (Blueprint $table) {
            $table->bigIncrements('id');
            $table->string('first_name');
            $table->string('last_name');
            $table->string('name');
            $table->string('email')->unique();
            $table->timestamp('email_verified_at')->nullable();
            $table->string('password');
            $table->bigInteger('phone');
            $table->rememberToken();
            $table->timestamps();
        });
    }
    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists('users');
```

```
}
```

```
};
```

User.php :

```
<?php
namespace App\Models;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Foundation\Auth\User as Authenticatable;
use Illuminate\Notifications\Notifiable;
use Laravel\Sanctum\HasApiTokens;
class User extends Authenticatable
{
    use HasApiTokens, HasFactory, Notifiable;
    /**
     * The attributes that are mass assignable.
     *
     * @var array<int, string>
     */
    protected $fillable = [
        'first_name', 'last_name', 'email', 'password', 'phone'
    ];
    /**
     * The attributes that should be hidden for serialization.
     *
     * @var array<int, string>
     */
    protected $hidden = [
        'password',
        'remember_token',
    ];
    /**
     * The attributes that should be cast.
     *
     * @var array<string, string>
     */
    protected $casts = [
        'email_verified_at' => 'datetime',
    ];
}
```

UserController.php :

```
<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Validator;
use Illuminate\Support\Facades\Redirect;
use App\Models\User;
use Illuminate\Support\Facades\Hash;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\Session;
class UserController extends Controller {
// ----- [ user registration view ]
public function index() {
return view('registration');
}
// ----- [ Register user ]
public function userPostRegistration(Request $request) {
// validate form fields
$request->validate([
'first_name' => 'required',
'last_name' => 'required',
'email' => 'required|email',
'password' => 'required|min:6',
'confirm_password' => 'required|same:password',
'phone' => 'required|max:10'
]);
$input = $request->all();
// if validation success then create an input array
$inputArray = array(
'first_name' => $request->first_name,
'last_name' => $request->last_name,
'full_name' => $request->first_name . " " . $request->last_name,
'email' => $request->email,
'password' => Hash::make($request->password),
'phone' => $request->phone
);
// register user
$user = User::create($inputArray);
// if registration success then return with success message
if(!is_null($user)) {
return back()->with('success', 'You have registered successfully.');
}
```

```

// else return with error message
else {
    return back()->with('error', 'Whoops! some error encountered. Please try
again.');
}
}
// ----- [ User login view ] -----
public function userLoginIndex() {
    return view('login');
}
// ----- [ User login ] -----
public function userPostLogin(Request $request) {
    $request->validate([
        "email" => "required|email",
        "password" => "required|min:6"
    ]);
    $userCredentials = $request->only('email', 'password');
    // check user using auth function
    if (Auth::attempt($userCredentials)) {
        return redirect()->intended('dashboard');
    }
    else {
        return back()->with('error', 'Whoops! invalid username or password.');
    }
}
// ----- [ User Dashboard Section ] -----
public function dashboard() {
    // check if user logged in
    if(Auth::check()) {
        return view('dashboard');
    }
    return redirect::to("user-login")->withSuccess('Opps! You do not have
access');
}
// ----- [ User logout function ] -----
public function logout(Request $request) {
    $request->session()->flush();
    Auth::logout();
    return Redirect('user-login');
}
}

```

master.blade.php :

```
<!-- master.blade.php -->
<!doctype html>
<html lang="en">
<head>
<title> Student Registration and Login with Session </title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">
<!-- Bootstrap CSS -->
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.cs
s"
integrity="sha384-
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRx
T2MZw1T"
crossorigin="anonymous">
</head>
<body>
@yield('content')
<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.j
s"></script>
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js">
</script>
</body>
</html>
```

registration.blade.php :

```
<!-- registration.blade.php -->
@extends('master')
@section('content')


<form method="post" action=" {{ url('user-store') }} ">
<div class="card shadow mb-4">
<div class="card-header bg-success pt-2">
<div class="card-title font-weight-bold text-white text-center"> Student


```

```

Registration
</div>
</div>
<div class="card-body">
@if(Session::has('success'))
<div class="alert alert-success">
{{ Session::get('success') }}
@php
Session::forget('success');
@endphp
</div>
@endif
<div class="form-group">
<label for="first_name"> First Name </label>
<input type="text" name="first_name" id="first_name" class="form-control" placeholder="Enter First Name" value="{{ old('first_name') }}"/>
{!! $errors->first('first_name', '<small class="text-danger">:message</small>') !!}
</div>
<div class="form-group">
<label for="last_name"> Last Name </label>
<input type="text" name="last_name" id="last_name" class="form-control" placeholder="Enter Last Name" value="{{ old('last_name') }}"/>
{!! $errors->first('last_name', '<small class="text-danger">:message</small>') !!}
</div>
<div class="form-group">
<label for="email"> E-mail </label>
<input type="text" name="email" id="email" class="form-control" placeholder="Enter E-mail" value="{{ old('email') }}"/>
{!! $errors->first('email', '<small class="text-danger">:message</small>') !!}
</div>
<div class="form-group">
<label for="password"> Password </label>
<input type="password" name="password" id="password" class="form-control" placeholder="Enter Password" value="{{ old('password') }}"/>
{!! $errors->first('password', '<small class="text-danger">:message</small>') !!}
</div>
<div class="form-group">
<label for="confirm_password"> Confirm Password </label>
<input type="password" name="confirm_password"

```

```

        id="confirm_password" class="form-control" placeholder="Confirm
        Password" value="{{ old('confirm_password') }}"
    {!! $errors->first('confirm_password', '<small class="text
        danger">:message</small>') !!}

```

```

    </div>
    <div class="form-group">
        <label for="phone"> Phone </label>
        <input type="phone" name="phone" id="phone" class="form-control"
            placeholder="Enter Phone" value="{{ old('phone') }}"
        {!! $errors->first('phone', '<small class="text-danger">:message</small>')
        !!}
    </div>
    </div>
    <div class="card-footer d-inline-block">
        <button type="submit" class="btn btn-success"> Register </button>
        <p class="float-right mt-2"> Already have an account? <a href="{{ url('user
            login') }}" class="text-success"> Login </a> </p>
    </div>
    @csrf
</div>
</form>
</div>
</div>
@endsection

```

Login.blade.php :

```

<!-- login.blade.php -->
@extends('master')
@section('content')


<form method="post" action="{{ url('login') }}">
<div class="card shadow">
<div class="card-header bg-success pt-2">
<div class="card-title font-weight-bold text-white text-center"> Student
        Login
    </div>
    </div>
    <div class="card-body">


```

```

@if(Session::has('success'))
<div class="alert alert-success">
{ Session::get('success') }
@php
Session::forget('success');
@endphp
</div>
@endif
<div class="form-group">
<label for="email"> E-mail </label>
<input type="text" name="email" id="email" class="form-control"
placeholder="Enter E-mail" value="{{ old('email') }}"/>
{!! $errors->first('email', '<small class="text-danger">:message</small>')
!!}
</div>
<div class="form-group">
<label for="password"> Password </label>
<input type="password" name="password" id="password" class="form
control" placeholder="Enter Password" value="{{ old('password') }}"/>
{!! $errors->first('password', '<small class="text
danger">:message</small>') !!}
</div>
</div>
<div class="card-footer d-inline-block">
<button type="submit" class="btn btn-success"> Login </button>
<p class="float-right mt-2"> Don't have an account? <a href="{{ url('user
registration') }}" class="text-success"> Register </a> </p>
</div>
@csrf
</div>
</form>
</div>
</div>
</div>
@endsection

```

dashboard.blade.php :

```

@extends('master')
@section('content')
<nav class="navbar navbar-expand-lg navbar-dark bg-success">
<a class="navbar-brand" href="#">Student Dashboard</a>
<button class="navbar-toggler" type="button" data-toggle="collapse" data

```

```

target="#navbarColor01" aria-controls="navbarColor01" aria-
expanded="false" aria
label="Toggle navigation">
<span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse" id="navbarColor01">
<ul class="navbar-nav ml-auto">
<li class="nav-item">
<a class="nav-link text-white"> Welcome:
{ { ucfirst(Auth()->user()->first_name) } }
</a>
</li>
<li class="nav-item">
<a class="nav-link" href="{{ url('logout') }}> Logout </a>
</li>
</ul>
</div>
</nav>
@endsection

```

web.php :

```

<?php
use Illuminate\Support\Facades\Route;
Route::get('/', function () {
return view('welcome');
});
Route::get('user-registration', 'App\Http\Controllers\UserController@index');
Route::post('user-store',
'App\Http\Controllers\UserController@userPostRegistration');
Route::get('user-login',
'App\Http\Controllers\UserController@userLoginIndex');
Route::post('login', 'App\Http\Controllers\UserController@userPostLogin');
Route::get('dashboard', 'App\Http\Controllers\UserController@dashboard');
Route::get('logout', 'App\Http\Controllers\UserController@logout');
})

```

OUTPUT :

The image shows a web-based application interface with two main sections: "Student Registration" and "Student Login".

Student Registration:

- First Name: Enter First Name (Required)
- Last Name: Enter Last Name (Required)
- E-mail: Enter E-mail (Required)
- Password: Enter Password (Required)
- Confirm Password: Confirm Password (Required)
- Phone: Enter Phone (Required)

Student Login:

- E-mail: mca2254@rajagiri.edu
- Password: (Redacted)
- Login
- Don't have an account? [Register](#)



PORTFOLIO**CODE :**Portfolio.html

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

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Sarah Chandran M - Portfolio</title>
    <link rel="stylesheet"
    href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
    <link rel="stylesheet" href="style.css">

    <style>
        .progress {
            height: 30px;
            margin-bottom: 15px;
        }
        .profile-photo {
            text-align: center;
            margin-bottom: 60px;
        }
        .profile-photo img {
            width: 200px;
            height: 200px;
            border-radius: 50%;
            border: 5px solid #fff;
            box-shadow: 0 0 10px rgba(0, 0, 0, 0.3);
        }
    </style>
</head>

<body>
```

```

<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <a class="navbar-brand" href="#">Sarah's Portfolio</a>
  <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
  </button>

  <div class="collapse navbar-collapse" id="navbarSupportedContent">
    <ul class="navbar-nav ml-auto">
      <li class="nav-item">
        <a class="nav-link" href="#education">Education</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#skills">Skills</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#projects">Projects</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#contact">Contact</a>
      </li>
    </ul>
  </div>
</nav>

<div class="container mt-5">
  <div class="profile-photo">
    
    <h1>Sarah Chandran M</h1>
    <p>Born on January 24, 2001</p>
  </div>
  <section id="education">
    <h2>Education</h2>
    <p><strong>Bachelor's Degree:</strong> Physics - Sree Kerala Varma College Thrissur (75%)</p>
    <p><strong>Current:</strong> Master of Computer Applications - Rajagiri College of Social Sciences Kalamassery</p>
  </section>
  <section id="skills">
    <h2>Skills</h2>
    <div class="row">

```

```

<div class="col-md-6">
    <p>Java</p>
    <div class="progress">
        <div class="progress-bar" style="width: 80%;">80%</div>
    </div>
    <p>C</p>
    <div class="progress">
        <div class="progress-bar" style="width: 70%;">70%</div>
    </div>
    <p>Git</p>
    <div class="progress">
        <div class="progress-bar" style="width: 75%;">75%</div>
    </div>
</div>
<div class="col-md-6">
    <p>HTML</p>
    <div class="progress">
        <div class="progress-bar" style="width: 85%;">85%</div>
    </div>
    <p>Bootstrap</p>
    <div class="progress">
        <div class="progress-bar" style="width: 80%;">80%</div>
    </div>
    <p>CSS</p>
    <div class="progress">
        <div class="progress-bar" style="width: 75%;">75%</div>
    </div>
    <p>JavaScript</p>
    <div class="progress">
        <div class="progress-bar" style="width: 70%;">70%</div>
    </div>
</div>
</div>
</section>
<section id="projects">
    <h2>Projects</h2>
    <div id="projects-list"></div>
    <script>
        // Dynamically populate projects
        var projectsList = document.getElementById('projects-list');
        var projects = ["Blood Donor Management System (Java)", "Blood
Donor Management System (PHP)"];
        projectsList.innerHTML = '<ul>';

```

```

        projects.forEach(function(project) {
            projectsList.innerHTML += '<li>' + project + '</li>';
        });
        projectsList.innerHTML += '</ul>';
    </script>
</section>
<section id="contact">
    <h2>Contact</h2>
    <form action="submit_contact_form.php" method="post">
        <!-- Your form fields -->
        <div class="form-group">
            <label for="name">Name:</label>
            <input type="text" class="form-control" id="name" name="name"
required>
        </div>
        <div class="form-group">
            <label for="email">Email:</label>
            <input type="email" class="form-control" id="email" name="email"
required>
        </div>
        <div class="form-group">
            <label for="message">Message:</label>
            <textarea class="form-control" id="message" rows="5"
name="message" required></textarea>
        </div>
        <button type="submit" class="btn btn-primary">Submit</button>
    </form>
</section>
</div>

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
<script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.5.4/dist/umd/popper.min.js
"></script>
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></s
cript>
</body>

</html>

```

Style.css

```
/* Custom CSS for the Portfolio */

/* Body Styles */
body {
    font-family: Arial, sans-serif;
    background-color: #d9eafa;
    color: #333;
}

/* Navbar Styles */
.navbar {
    background-color: #9fb9d3 !important; /* Dark grey color */
}

.navbar-brand {
    color: #ee1e1e; /* White color */
    font-weight: bold;
    font-size: 1.5rem;
}

.navbar-nav .nav-link {
    color: #fff; /* White color */
}

.navbar-nav .nav-link:hover {
    background-color: #74da6b;
}

/* Header Styles */
header {
    text-align: center;
    margin-bottom: 50px;
}

header h1 {
    font-size: 2.5rem;
    color: #007bff; /* Blue color */
}

header p {
    font-size: 1.2rem;
}
```

```
color: #666;  
}  
  
/* Section Styles */  
section {  
    margin-bottom: 50px;  
}  
  
section h2 {  
    font-size: 2rem;  
    color: #007bff; /* Blue color */  
    margin-bottom: 20px;  
}  
  
section p {  
    font-size: 1.1rem;  
    color: #555;  
}  
  
/* Profile Photo Styles */  
.profile-photo img {  
    width: 200px;  
    height: 200px;  
    border-radius: 50%;  
    object-fit: cover;  
    border: 5px solid #fff;  
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.3);  
    transition: transform 0.3s ease-in-out;  
}  
  
.profile-photo img:hover {  
    transform: scale(1.1);  
}  
  
/* Skills Styles */  
.skill {  
    margin-bottom: 20px;  
}  
  
.skill p {  
    margin-bottom: 5px;  
    color: #333;  
}
```

```
.progress {
    background-color: #f0f0f0;
}

.progress-bar {
    background-color: #28a745;
}

/* Projects Styles */
#projects-list ul {
    list-style-type: none;
    padding: 0;
}

#projects-list li {
    margin-bottom: 10px;
    font-size: 1.1rem;
    color: #333;
}

/* Contact Form Styles */
#contact form {
    max-width: 500px;
    margin: 0 auto;
}

#contact label {
    font-size: 1.1rem;
    color: #333;
}

#contact input,
#contact textarea {
    margin-bottom: 20px;
    font-size: 1rem;
}

#contact button {
    font-size: 1.2rem;
    background-color: #007bff;
    border: none;
    transition: background-color 0.3s ease-in-out;
}
```

```
}
```

```
#contact button:hover {
```

```
    background-color: #0056b3;
```

```
}
```

db_connect.php

```
<?php
```

```
// Database connection variables
```

```
$servername = "localhost"; // Change this if your database server is different
```

```
$username = "your_username";
```

```
$password = "sarath";
```

```
$dbname = "sarath";
```

```
// Create connection
```

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

```
// Check connection
```

```
if ($conn->connect_error) {
```

```
    die("Connection failed: " . $conn->connect_error);
```

```
}
```

```
?>
```

Submit_contact_form.php

```
<?php
```

```
// Include the database connection file
```

```
include 'db_connect.php';
```

```
// Check if form is submitted
```

```
if ($_SERVER["REQUEST_METHOD"] == "POST") {
```

```
    // Escape user inputs for security
```

```
    $name = $conn->real_escape_string($_POST['name']);
```

```
    $email = $conn->real_escape_string($_POST['email']);
```

```
    $message = $conn->real_escape_string($_POST['message']);
```

```
    // Insert the form data into the database
```

```
    $sql = "INSERT INTO contact_form (name, email, message) VALUES
```

```
        ('$name', '$email', '$message');
```

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

```
        echo "Form submitted successfully!";
```

```

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

// Close the database connection
$conn->close();
?>

```

The screenshot shows a professional portfolio website for Sarath Chandran M. The header features a circular profile picture of the author, followed by his name and birth date. Below the header, there are sections for Education, Skills, and Projects, each containing a list of items with progress bars indicating proficiency levels.

Sarath's Portfolio

Sarath Chandran M
Born on January 24, 2001

Education

Bachelor's Degree: Physics - Sree Kerala Varma College Thrissur (75%)
Current: Master of Computer Applications - Rajagiri College of Social Sciences Kalamassery

Skills

Skill	Level (%)
Java	80%
C	70%
Git	75%
HTML	85%
Bootstrap	80%
CSS	75%
JavaScript	70%

Projects

- Blood Donor Management System (Java)
- Blood Donor Management System (PHP)

Contact

Name:

Email:

Message:

Submit



PROJECT**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>Blood Donor Platform</title>
    <link rel="stylesheet"
        href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
        <link rel="stylesheet"
        href="https://cdnjs.cloudflare.com/ajax/libs/OwlCarousel2/2.3.4/assets/owl.carousel.min.css">
            <link rel="stylesheet"
            href="https://cdnjs.cloudflare.com/ajax/libs/OwlCarousel2/2.3.4/assets/owl.theme.default.min.css">
                <style>
                    body {
                        background-image: url('https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTj8IqZ6CZUxRhLSAxAxN0JVRUR2MkgaNLDaw&usqp=CAU');
                        background-size: cover;
                        background-position: center;
                        color: #fbf6f6;
                        font-family: 'Arial', sans-serif;
                    }

                    .container {
                        margin-top: 100px;
                    }

                    .btn-primary,
                    .btn-danger,
```

```
.btn-warning {  
    width: 300px;  
    margin-right: 20px;  
    transition: background-color 0.3s ease, transform 0.3s ease;  
}  
  
.btn-primary:hover,  
.btn-danger:hover,  
.btn-warning:hover: hover {  
    background-color: #50f756;  
    color: #333;  
    transform: scale(1.1);  
}  
  
.btn-description {  
    font-size: 1em;  
    margin-top: 5px;  
    opacity: 0;  
    transition: opacity 0.3s ease;  
}  
  
.btn-primary:hover .btn-description,  
.btn-danger:hover .btn-description,  
.btn-warning:hover .btn-description {  
    opacity: 1;  
}  
  
.owl-carousel .item img {  
    max-width: 50%;  
    max-height: 500px;  
    margin: 0 auto;  
    border-radius: 10px;  
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.5);  
    transition: transform 0.5s ease;  
}  
  
.owl-carousel .item img:hover {  
    transform: scale(1.05);  
}  
    
```

</style>

</head>

<body>

```

<div class="container text-center">
    <h1 style="font-size: 3.25em; font-weight: bold;">Welcome To Online Blood
    Donor Platform</h1>
    <p style="font-size: 1.2em;">Choose an option below:</p>
    <button class="btn btn-primary" onclick="requestBlood()">
        Request Blood
        <div class="btn-description">Click to Request</div>
    </button>
    <button class="btn btn-danger" onclick="donateBlood()">
        Donate Blood
        <div class="btn-description">Click to Donate Your valuable Blood</div>
    </button>
    <button class="btn btn-warning" onclick="Topage()">
        Eligibility Criteria
        <div class="btn-description">Check you are Eligible for Donation</div>
    </button>
</div>
<br><br><br>

<div class="owl-carousel owl-theme" data-autoplay="true" data-autoplay-
timeout="3000">
    <div class="item"></div>
    <div class="item"></div>
    <div class="item"></div>
    <div class="item"></div>
</div>

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
<script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.5.1/dist/umd/popper.min.js
"></script>
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></s
cript>

```

```

<script
src="https://cdnjs.cloudflare.com/ajax/libs/OwlCarousel2/2.3.4/owl.carousel.min.js"></script>

<script>
$(document).ready(function () {
    $('.owl-carousel').owlCarousel({
        loop: true,
        margin: 10,
        nav: false,
        dots: true,
        items: 1,
        autoplay: $('.owl-carousel').data('autoplay')),
        autoplayTimeout: $('.owl-carousel').data('autoplay-timeout'))
    });
});

function requestBlood() {
    window.location.href = 'blood_req.html'; // Replace 'request_blood.html' with your desired URL
}

function donateBlood() {
    window.location.href = 'register.html'; // Replace 'donate_blood.html' with your desired URL
}

function Topage() {
    window.location.href = 'check_fitness.html'; // Replace 'donate_blood.html' with your desired URL
}
</script>

</body>

</html>

```

Register.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="styles.css">
<title>Donor Registration</title>
<style>
    /* Your styles go here */
</style>
</head>
<body>
    <div class="container">
        <h2>Donor Registration</h2>
        <form action="register.php" method="post" onsubmit="return
checkAge()">
            <label for="name">Name:</label>
            <input type="text" id="name" name="name" required>

            <label for="phone">Phone Number:</label>
            <input type="tel" id="phone" name="phone" required>

            <label for="gender">Gender:</label>
            <select id="bloodGroup" name="gender" required>
                <option value="male">Male</option>
                <option value="female">Female</option>
            </select>

            <label for="bloodGroup">Blood Group:</label>
            <select id="bloodGroup" name="bloodGroup" required>
                <option value="A+">A+</option>
                <option value="A->">A-</option>
                <option value="B+">B+</option>
                <option value="B->">B-</option>
                <option value="O+">O+</option>
                <option value="O->">O-</option>
                <option value="AB+">AB+</option>
                <option value="AB->">AB-</option>
            </select>

            <label for="weight">Weight (kg):</label>
            <input type="number" id="weight" name="weight" required min="51">

            <label for="address">Address:</label>
            <input type="text" id="address" name="address" required>

            <label for="dob">Date of Birth:</label>
```

```

<input type="date" id="dob" name="dob" required>

<label for="noDiseases">
    Make Sure You Are Fit For Donation
    <a href="check_fitness.html" target="_blank">Check fitness</a>
</label>

<button type="submit">Register</button>
<br>
<br>

</form>
</div>

<script>
    function checkAge() {
        var dobInput = document.getElementById("dob");
        var dob = new Date(dobInput.value);
        var currentDate = new Date();
        var age = Math.floor((currentDate - dob) / (365.25 * 24 * 60 * 60 * 1000));

        if (age < 18 && age > 65) {
            alert("You must be 18-65 years old to register as a donor.");
            return false; // Cancel the registration
        }

        // If the donor is above 18, proceed with the registration
        return true;
    }
</script>
</body>
</html>

```

Blood_req.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="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
<link rel="stylesheet" href="style_req.css">
<title>Blood Bank</title>
</head>
<body>
<header class="bg-primary text-white text-center py-4">
  <h1>Blood Bank</h1>
</header>

<main class="container mt-4">

  <section id="request" class="mt-4 text-center">
    <h2>Request Blood</h2>
    <form id="requestForm" action="getdonor.php" method="post">
      <div class="form-group">
        <label for="bloodType">Blood Type:</label>
        <select id="bloodGroup" name="bloodType" class="form-control" required>
          <option value="A+">A+</option>
          <option value="A->">A-</option>
          <option value="B+">B+</option>
          <option value="B->">B-</option>
          <option value="O+">O+</option>
          <option value="O->">O-</option>
          <option value="AB+">AB+</option>
          <option value="AB->">AB-</option>
        </select>
      </div>
      <button type="submit" class="btn btn-primary">Request</button>
    </form>
    
    <br>
    <a href="register.html" class="btn btn-link">Click here to donate your valuable Blood. </a>
  </section>
</main>

<script src="script.js"></script>
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>

```

```
<script  
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.0.7/dist/umd/popper.min.js  
"></script>  
<script  
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></s  
cript>  
</body>  
</html>
```

Check_fitness.html

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
<meta charset="UTF-8">  
<meta name="viewport" content="width=device-width, initial-scale=1.0">  
<title>Blood Donation Requirements</title>  
<!-- Bootstrap CSS -->  
<link  
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"  
rel="stylesheet">  
<style>  
body {  
background-image: linear-gradient(to right, rgb(215, 239, 174) , rgb(190,  
190, 250));  
}  
  
.container {  
margin-top: 50px;  
box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);  
background-color: #fff; /* Add a white background to the container */  
padding: 20px;  
border-radius: 8px; /* Add rounded corners to the container */  
}  
  
h2 {  
color: #dc3545;  
}  
  
ul {  
list-style-type: none;  
padding: 0;  
}
```

```
li {
    margin-bottom: 10px;
    transition: background-color 0.3s ease-in-out;
    box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
    padding: 10px;
    border-radius: 6px; /* Add rounded corners to the list items */
}

li:hover {
    background-color: #cce5ff; /* Highlight color on hover */
}

.alert {
    margin-top: 20px;
}

.alert-info {
    box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
    padding: 15px;
    border-radius: 8px; /* Add rounded corners to the alert */
}

.alert-success {
    box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
    padding: 15px;
    border-radius: 8px; /* Add rounded corners to the alert */
}

.a.btn-link {
    color: #d9534f; /* Red link color */
    text-decoration: none;
    margin-top: 10px;
    display: inline-block;
    transition: color 0.3s ease-in-out;
}

.a.btn-link:hover {
    color: #c12e2a; /* Darker red on hover */
}

</style>
</head>
<body>
```

```
<div class="container">
<h2>Blood Donation Requirements</h2>

<div class="alert alert-info" role="alert">
<strong>Note:</strong> Make sure to consult with a healthcare professional before donating blood.
</div>

<ul>
<li><strong>Age:</strong> You must be between 18 and 65 years old.</li>
<li><strong>Weight:</strong> You must weigh at least 110 pounds (50 kg).</li>
<li><strong>Health:</strong> You should be in good health on the day of donation.</li>
<li><strong>Travel:</strong> If you've traveled recently, check if you meet the eligibility criteria.</li>
<li><strong>Medication:</strong> Some medications may disqualify you from donating; consult with staff.</li>
<li><strong>Recent Donations:</strong> Respect the waiting period between donations (e.g., 56 days for whole blood).</li>
<li><strong>Lifestyle:</strong> Certain lifestyle choices may affect eligibility (e.g., recent tattoos or piercings).</li>
</ul>

<div class="alert alert-success" role="alert">
<strong>You're ready to save lives!</strong> Thank you for considering blood donation <br>
<a href="register.html" class="btn btn-link">Click here to donate your valuable Blood. </a>
</div>
</div>

<!-- Bootstrap JS and Popper.js -->
<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js">
</script>
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"></script>

</body>
</html>
```

Db.php

```
<?php
$servername = "localhost";
$username = "root";
$password = "Sarah@9747";
$database = "sarath_db";

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

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Function to retrieve donors from the database
function getDonors() {
    global $conn;

    $sql = "SELECT * FROM donors";
    $result = $conn->query($sql);

    $donors = array();

    if ($result->num_rows > 0) {
        while ($row = $result->fetch_assoc()) {
            $donors[] = $row;
        }
    }

    return $donors;
}

// Close the database connection
?>
```

Get_donor.php

```
<?php
require_once("db.php");

$selectedBloodGroup = isset($_POST['bloodType']) ? $_POST['bloodType'] : "";
$donors = [];

if (!empty($selectedBloodGroup)) {
    $sql = "SELECT * FROM php_donor WHERE bloodGroup =
'$selectedBloodGroup'";
    $result = $conn->query($sql);

    if ($result->num_rows > 0) {
        while ($row = $result->fetch_assoc()) {
            $donors[] = $row;
        }
    }
}

$conn->close();
?>

<!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_result.css">
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
    <title>Donor Details</title>
</head>
<body>
    <div class="container mt-4">
        <h1>Donor Details for Blood Group <?php echo
$selectedBloodGroup; ?></h1>
        <br>
```

```

<br>
<div class="row">
    <?php foreach ($donors as $donor) : ?>
        <div class="col-md-4 mb-4">
            <div class="card">
                <div class="card-body">
                    <h5 class="card-title"><?php echo $donor['name']; ?></h5>
                    <p class="card-text">Phone: <?php echo
$donor['phone']; ?></p>
                    <p class="card-text">Gender: <?php echo
$donor['gender']; ?></p>
                    <p class="card-text">Blood Group: <?php echo
$donor['bloodGroup']; ?></p>
                    <p class="card-text">Weight: <?php echo
$donor['weight']; ?></p>
                    <p class="card-text">Address: <?php echo
$donor['address']; ?></p>
                </div>
            </div>
        </div>
    <?php endforeach; ?>
</div>
</div>

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
<script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.0.7/dist/umd/popper.min.js
"></script>
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></s
cript>
</body>
</html>

```

Register.php

```

<?php
require_once("db.php");

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $name = isset($_POST["name"]) ? strtoupper($_POST["name"]) : "";
    $phone = isset($_POST["phone"]) ? ($_POST["phone"]) : "";

```

```

$gender = isset($_POST["gender"]) ? strtoupper($_POST["gender"]) : "";
$bloodGroup = isset($_POST["bloodGroup"]) ?
strtoupper($_POST["bloodGroup"]) : "";
$weight = isset($_POST["weight"]) ? ($_POST["weight"]) : "";
$address = isset($_POST["address"]) ? strtoupper($_POST["address"]) : "";
$noDiseases = isset($_POST["noDiseases"]) ? "Yes" : "No";

$sql = "INSERT INTO php_donor (name, phone, gender, bloodGroup, weight,
address)
VALUES ('$name', '$phone', '$gender', '$bloodGroup', '$weight',
'$address')";
```

```

if ($conn->query($sql) === TRUE) {
    echo '<script>alert("Registration successful!");</script>';
    echo '<script>window.location.href = "index.html";</script>';
} else {

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

$conn->close();
?>
```

Style.css

```

body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
    background-color: #e3f5e0;
    background-repeat: no-repeat;
}

.container {
    max-width: 400px;
    margin: 50px auto;
    background-color: #adfa69;
    padding: 20px;
    border: 1px solid #860c0c;
    border-radius: 8px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
```

```
}

h2 {
    color: #333;
    text-align: center;
}

form {
    display: flex;
    flex-direction: column;
}

label {
    margin-bottom: 8px;
    color: #333;
}

input,
select,
button {
    padding: 10px;
    margin-bottom: 16px;
    border: 1px solid #ccc;
    border-radius: 4px;
    transition: border-color 0.3s ease-in-out;
    width: 100%; /* Make inputs and selects full-width */
    box-sizing: border-box; /* Include padding and border in width */
}

input:hover,
select:hover,
button:hover {
    border-color: #4CAF50;
}

button {
    background-color: #4CAF50;
    color: white;
    border: none;
    border-radius: 4px;
    cursor: pointer;
}
```

```
button:hover {  
    background-color: #45a049;  
}  
  
#bloodGroup {  
    padding: 10px;  
}  
  
#gender {  
    padding: 10px;  
}  
  
img {  
    width: 100%;  
    height: auto;  
    border-radius: 8px;  
    margin-top: 10px;  
}  
  
input[type="radio"] {  
    margin-right: 5px;  
}  
  
/* Target radio button labels for proper alignment */  
input[type="radio"] + label {  
    display: inline-flex;  
    align-items: center;  
    margin-bottom: 8px;  
}  
  
#addressFields {  
    display: none;  
}  
  
#addressFields label,  
#addressFields input {  
    margin-bottom: 10px;  
}  
  
#addressFields input:hover {  
    border-color: #4CAF50;  
}
```

```
/* Additional Styles for Address Fields */
#addressFields input[type="text"] {
    padding: 10px;
}

#addressFields input[type="text"]:hover {
    border-color: #4CAF50;
}
```



Blood Bank

Request Blood

Blood Type:

A+

Request



[Click here to donate your valuable Blood.](#)



Donor Registration

Name:

Phone Number:

Gender:

Male

Blood Group:

A+

Weight (kg):

Address:

Date of Birth:

dd - mm - yyyy

Make Sure You Are Fit For Donation [Check fitness](#)

Register



Blood Donation Requirements

Note: Make sure to consult with a healthcare professional before donating blood.

Age: You must be between 18 and 65 years old.

Weight: You must weigh at least 110 pounds (50 kg).

Health: You should be in good health on the day of donation.

Travel: If you've traveled recently, check if you meet the eligibility criteria.

Medication: Some medications may disqualify you from donating; consult with staff.

Recent Donations: Respect the waiting period between donations (e.g., 56 days for whole blood).

Lifestyle: Certain lifestyle choices may affect eligibility (e.g., recent tattoos or piercings).

You're ready to save lives! Thank you for considering blood donation

[Click here to donate your valuable Blood.](#)

