**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. | **17** |
| **2.2** | Write a JavaScript program that accepts two integers and displays the larger | **18** |
| **2.3** | Write a program to implement Email Validation in JavaScript | **20** |
| **2.4** | Write a JavaScript program to sort the items of an array | **21** |
| **2.5** | Demonstrate Function call in JavaScript. | **22** |
| **2.6** | Write a JavaScript program to test the first character of a string is uppercase or not. | **23** |
| **2.7** | Write a JavaScript Program to Convert Decimal to Binary. | **25** |
| **2.8** | Create a webpage that displays the current date and time using JavaScript | **27** |
| **2.9** | Implement a JavaScript program to change the background color randomly. | **28** |
| **2.10** | Write a JavaScript program to construct the pyramid pattern, using a nested for loop. | **29** |
| **2.11** | Write a JavaScript Program to Solve Quadratic Equations. | **30** |
| **2.12** | Write a JavaScript function that accepts a string as a parameter and counts the number of vowels within the string. | **32** |
| **2.13** | Write a program to Toggle Password Visibility in JavaScript. | **34** |

|  |  |  |
| --- | --- | --- |
| **JSON** | | |
| **3.1** | Write a program to demonstrate the Fibonacci sequence in JSON. | **35** |
| **3.2** | Write a program to check leap years. | **36** |
| **PHP** | | |
| **4.1** | Write a PHP program to check whether a given number is Armstrong or not. | **37** |
| **4.2** | Write a PHP program to illustrate Cookies. | **39** |
| **4.3** | Check whether a year is a leap year or not by inserting a year in PHP. | **41** |
| **4.4** | Write a program to Implement static properties in PHP | **43** |
| **4.5** | Write a PHP program to use the global keyword to access a global variable from within a function. | **45** |
| **4.6** | Write a PHP program to convert a decimal number into a binary number | **47** |
| **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 | **48** |
| **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. | **49** |
| **4.9** | Write the PHP script to send an email. | **52** |
| **4.10** | Write a PHP program to implement MySQL databases with CRUD operations. | **54** |
| **LARAVEL FRAMEWORK** | | |
| **5.1** | Program to illustrate the workings of a simple Laravel web application. | **59** |
| **5.2** | Program to demonstrate the implementation of the Bootstrap framework in Laravel. | **60** |
| **5.3** | Program to illustrate the creation of a Laravel web application with all form validations. | **62** |
| **5.4** | Create a Login page using the Laravel framework. | **65** |
| **5.5** | Program to implement session handling using the Laravel framework. | **72** |

|  |  |  |
| --- | --- | --- |
| **5.6** | Write a program to execute database CRUD operations in the Laravel framework. | **77** |
| **5.7** | Create a simple Laravel Web Application, including:  1. A registration form with all form validations.  2. Perform CRUD operations using migrations. | **85** |
| **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. | **99** |
| **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 | **110** |
| **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. | **122** |
| **6** | **PORTFOLIO** |  |
| **7** | **PROJECT** |  |

**.**..

|  |  |
| --- | --- |
| **Program 1.1** | **Date: 01/01/2024** |
| **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">  <img src="p1\_1.jpg" alt="img">  <img src="p1\_1.jpg" alt="img">  <img src="p1\_1.jpg" alt="img">  <img src="p1\_1.jpg" alt="img">  <img src="p1\_1.jpg" alt="img">  </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;  }  </style>  </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>  <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 :**

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **Program 1.3** | **Date: 02/01/2024** |
| **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">  <img src="p1\_3.jpg" alt="Image" class="zoom-image">  <img src="p1\_3.jpg" alt="Image" class="zoom-image">  <img src="p1\_3.jpg" alt="Image" class="zoom-image">  </div>  <!-- To disable the default right-click behavior on the webpage. -->  <script>  document.addEventListener('contextmenu', function(e) {  e.preventDefault();  });  </script>  </body>  </html> |

**OUTPUT :**

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **Program 1.4** | **Date: 04/01/2024** |
| **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 :**

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **Program 1.5** | **Date: 05/01/2024** |
| **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 :**

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **Program 1.6** | **Date: 06/01/2024** |
| **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 -->  <td> <input type="button" value="=" onclick="calculate()" id="btn" /> </td>  <td> <input type="button" value="\*" onclick="display('\*')" /> </td>  </tr>  </table>    <script>  // 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;  }  </script>  </body>    </html> |

**OUTPUT :**

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **Program 1.7** | **Date: 07/01/2024** |
| **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 :**

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **Program 1.9** | **Date: 09/01/2024** |
| **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 :**

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **Program 2.1** | **Date: 10/01/2024** |
| **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 :**

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **Program 2.2** | **Date: 11/01/2024** |
| **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 :**

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **Program 2.3** | **Date: 22/01/2024** |
| **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 :**

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **Program 2.4** | **Date: 24/01/2024** |
| **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 :**

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **Program 2.5** | **Date: 28/01/2024** |
| **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 :**

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **Program 2.6** | **Date: 28/01/2024** |
| **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 :**

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **Program 2.7** | **Date: 1/02/2024** |
| **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 :**

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **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 :**

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **Program 2.9** | **Date: 2/02/2024** |
| **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("&nbsp;"); // 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 :**

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **Program 2.11** | **Date: 8/02/2024** |
| **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 :**

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **Program 2.12** | **Date: 10/02/2024** |
| **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">  <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])) {  // 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 :**

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **Program 2.13** | **Date: 14/02/2024** |
| **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 :**

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **Program 3.1** | **Date: 20/02/2024** |
| **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]);  }  var fibonacciJSON = JSON.stringify(fibonacciSequence);  document.getElementById("fibonacciOutput").textContent = "Fibonacci Sequence (JSON): " + fibonacciJSON;  }  </script>  </body>  </html> |

**OUTPUT :**

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **Program 3.2** | **Date: 21/02/2024** |
| **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 :**

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **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 :**

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **Program 4.2** | **Date: 23/02/2024** |
| **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 :**

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **Program 4.3** | **Date: 24/02/2024** |
| **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 :**

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **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 :**

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **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 :**

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **Program 4.6** | **Date: 28/02/2024** |
| **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 :**

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **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'  $currentDateTime = date("Y-m-d H:i:s");  setcookie('last\_visited', $currentDateTime, time() + (86400 \* 30), "/"); // Cookie expires in 30 days  ?>  </body>  </html> |

**OUTPUT :**

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **Program 4.8** | **Date: 08/03/2024** |
| **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 :**

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **Program 4.9** | **Date: 09/03/2024** |
| **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 :**

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **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 :**

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **Program 5.1** | **Date: 12/03/2024** |
| **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 5.2** | **Date: 12/03/2024** |
| **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" 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-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>  <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 :**

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **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>  </div>  @endsection('content') |

**login.blade.php**

|  |
| --- |
| @extends('main')  @section('content')  @if($message = Session::get('success'))  <div class="alert alert-info">  {{ $message }}  </div>  @endif  <div class="row justify-content-center">  <div class="col-md-4">  <div class="card">  <div class="card-header">Login</div>  <div class="card-body">  <form action="{{ route('sample.validate\_login') }}" method="post">  @csrf  <div class="form-group mb-3">  <input type="text" name="email" class="form-control" placeholder="Email"  />  @if($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="subit" class="btn btn-dark btn-block">Login</button>  </div>  </form>  </div>  </div>  </div>  </div>  @endsection('content') |

**dashboard.blade.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') |

**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 :**

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **Program 5.5** | **Date: 14/03/2024** |
| **Program to implement session handling using the Laravel framework.** | |

**registration.blade.php :**

|  |
| --- |
| <!-- registration.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('user-store') }} ">  <div class="card shadow mb-4">  <div class="car-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') }}  @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>  </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="car-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') }}  @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-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 :**

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **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')  <div class="row">  <div class="col-lg-12 margin-tb">  <div class="pull-left">  <h2>Add New 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.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 }}</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 |

**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>  <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 :**

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **Program 5.7** | **Date: 16/03/2024** |
| **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>  @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>  @error('password')  <p class="text-red-500 text-xs italic mt-4">  {{ $message }}  </p>  @enderror  </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') }}</span>  </label>  @if (Route::has('password.request'))  <a class="text-sm text-blue-500 hover:text-blue-700 whitespace-no-wrap no  underline hover:underline ml-auto"  href="{{ route('password.request') }}">  {{ \_\_('Forgot Your Password?') }}  </a>  @endif  </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') }}  </a>  </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')  <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-yellow-500 uppercase bg-transparent text-gray-100 text-xs font-extrabold  py-3 px-5 rounded-3xl">  Add 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>  <img src="{{ asset('images/' . $post->image\_path) }}" alt="">  </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>  @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')  131@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:**

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **Program 5.8** | **Date: 16/03/2024** |
| **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')  <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')  <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>  <img src="{{ asset('images/' . $post->image\_path) }}" alt="">  </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 :**

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **Program 5.9** | **Date: 17/03/2024** |
| **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->marsk\_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>  @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">×</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">×</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">×</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 :**

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **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('Oopps! 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.css"  integrity="sha384-  ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"  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.js"></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')  <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('user-store') }} ">  <div class="card shadow mb-4">  <div class="car-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>  </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="car-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 :**

|  |
| --- |
|  |