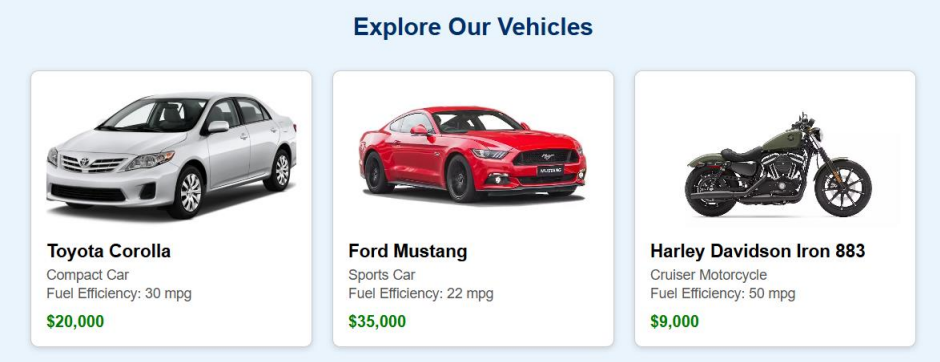
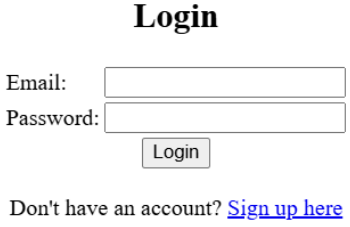
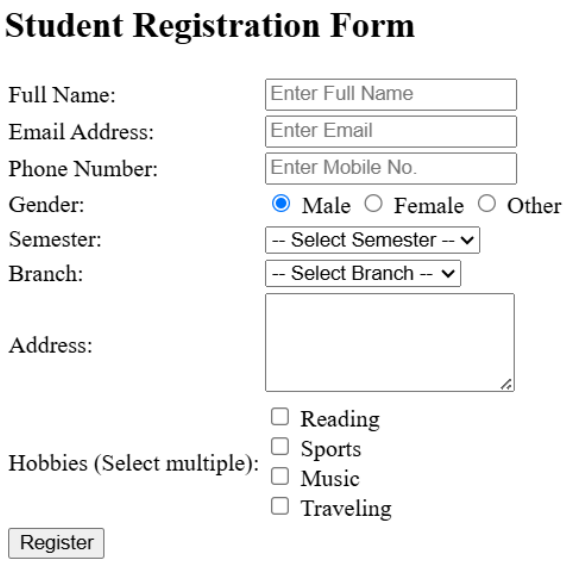
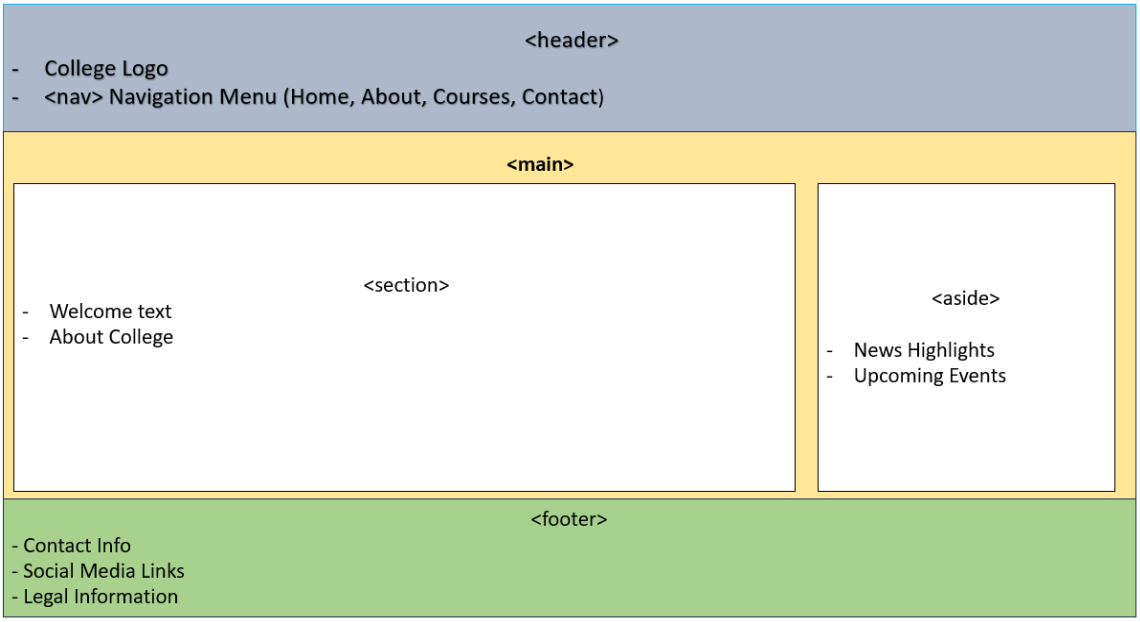
<table>
<tr><th>Lab</th><th>Type</th><th>No</th><th>Practical</th></tr>
<tr><td colspan="4">Practical -1: Structuring content with table, lists, and div tags</td></tr>
<tr><td rowspan="3">LAB-1</td><td>A</td><td>1.</td><td>Create a weekly class timetable using HTML. Use the <table> tag to show the days and time slots clearly. Add some basic CSS to make the timetable look neat and easy to read.</td></tr>
<tr><td>A</td><td>2.</td><td>Create the HTML table given below that demonstrates the use of bold, italic, and underline text, along with rowspan and colspan. Apply basic CSS styling to enhance the table's appearance.
<table>
<tr><th colspan="6">Data about Various Vehicles (Source: Vehicle Data Source).</th></tr>
<tr><th></th><th>Name</th><th>Type</th><th>Price ($)</th><th>Year</th><th>Notes</th></tr>
<tr><td rowspan="4">Cars</td><td>Toyota Corolla</td><td>Compact Car</td><td>20,000</td><td>2024</td><td>Reliable and fuel-efficient</td></tr>
<tr><td>Honda Accord</td><td>Midsize Car</td><td>25,000</td><td>2024</td><td>Spacious and comfortable</td></tr>
<tr><td>Ford Mustang</td><td>Sports Car</td><td>35,000</td><td>2024</td><td>Performance-oriented</td></tr>
<tr><td>Tesla Model S</td><td>Electric Car</td><td>95,000</td><td>2024</td><td>High range and performance</td></tr>
<tr><td rowspan="3">Motorcycles</td><td>Harley Davidson Iron 883</td><td>Cruiser</td><td>9,000</td><td>2024</td><td>Classic design</td></tr>
<tr><td>Yamaha YZF-R6</td><td>Sportbike</td><td>12,000</td><td>2024</td><td>Track-focused</td></tr>
<tr><td>Kawasaki KLR 650</td><td>Adventure Bike</td><td>7,500</td><td>2024</td><td>Versatile</td></tr>
<tr><td rowspan="2">Trucks</td><td>Ford F-150</td><td>Pickup Truck</td><td>30,000</td><td>2024</td><td>Heavy-duty</td></tr>
<tr><td>Chevrolet Silverado 1500</td><td>Pickup Truck</td><td>32,000</td><td>2024</td><td>Popular choice</td></tr>
<tr><td rowspan="3">Summary by Category</td><td colspan="3">Cars</td><td colspan="2">Total Vehicles: 4</td></tr>
<tr><td colspan="3">Motorcycles</td><td colspan="2">Total Vehicles: 3</td></tr>
<tr><td colspan="3">Trucks</td><td colspan="2">Total Vehicles: 2</td></tr>
<tr><td>Overall Summary</td><td colspan="4">Total Vehicles</td><td>9</td></tr>
</table></td></tr>
<tr><td>A</td><td>3.</td><td>Create an HTML document with an ordered list (<ol>), unordered list (<ul>), and description list (<dl>).

<h2>Vehicle Overview</h2>
<ol>
<li>Car<ul><li>Toyota Corolla</li><li>Honda Accord</li><li>Ford Mustang</li></ul></li>
<li>Motorcycle<ul><li>Harley Davidson Iron 883</li><li>Yamaha YZF-R6</li></ul></li>
<li>Truck<ul><li>Ford F-150</li></ul></li>
</ol>
<h2>Vehicle Type Descriptions</h2>
<dl>
<dt>Compact Car</dt><dd>Small and fuel-efficient, suitable for city driving.</dd>
<dt>Sportbike</dt><dd>High-performance motorcycle built for speed and agility.</dd>
<dt>Pickup Truck</dt><dd>Strong utility vehicle used for hauling and towing.</dd>
</dl></td></tr>
</table>

	A	4.	Create an HTML document using the tag to organize content into sections. Apply basic CSS to style the sections. 
Practical -2: Design and implementation of a static website Interface using HTML, CSS and Bootstrap			
LAB-2	A	1.	Design an HTML-based login interface that includes input fields for both username and password. 
	A	2.	Create an HTML student registration page that includes following input fields such as name, email, phone number, gender, semester, branch, address and hobbies. 

	A	3.	Create a static webpage for college using HTML5 tags, including a <header>, <footer>, <nav>, <aside>, <main> and, <section>. Structure the content within these elements and apply basic CSS for layout and styling, without using Bootstrap. 
	A	4.	Create a static Blog website using Bootstrap.
Practical -3: Demonstration of basics of PHP programming			
LAB-3	A	1.	WAP to display “Hello World”.
	A	2.	WAP to display a message using variable.
	A	3.	Create a PHP script - Declare a variable $value and assign it an integer value (e.g., $value = 10;). - Use the gettype() function to display the type of $value. - Use settype($value, 'string') to change the type of $value to a string. - Use gettype() again to display the new type of $value. - Use var_dump($value) to display the value and type of $value after the conversion.
	B	4.	WAP to print name of month based on given number of month (i.e. 1 -> January, 2 -> February…).
	B	5.	WAP to swap values of two variables with the help of 3rd variable.
	C	6.	WAP to swap values of two variables without using 3rd variable.
Practical -4: Implementation of decision-making statements (Part – I)			
	A	1.	WAP to check whether the given number is odd or even.

LAB-4	A	2.	WAP to check whether the given number is positive, negative or zero.
	A	3.	WAP to find greatest number from 2 numbers.
	B	4.	WAP to find greatest number from 3 numbers.
	B	5.	WAP to convert temperature from Fahrenheit to Celsius.
	C	6.	WAP to print class of result based on percentage (i.e. less than 40% -> Fail, 40% to 50% -> Pass Class, 50% to 60% -> Second Class, 60% to 70% -> First Class, above 70% -> Distinction).
	C	7.	WAP that reads a number in meters, converts it to feet, and displays the result.
Practical -5: Implementation of decision-making statements (Part – II)			
LAB-5	A	1.	WAP to take a value from 1-7 and display current day using switch case. (If 1-Monday, 2-Tuesday, etc….)
	A	2.	WAP to find a diameter from given area of circle.
	B	3.	WAP to make a Simple Calculator using switch case.
	B	4.	WAP to print class of result based on percentage using switch case (i.e. less than 40% -> Fail, 40% to 50% -> Pass Class, 50% to 60% -> Second Class, 60% to 70% -> First Class, above 70% -> Distinction).
	C	5.	WAP that check whether a letter is a vowel or consonants.
	C	6.	Three sides of a triangle are entered through the keyboard. WAP to check whether the triangle is isosceles, equilateral, scalene or right-angled triangle.
Practical -6: Implementation of various loops in PHP			
LAB-6	A	1.	WAP to print first n numbers using for, while and do while loop.
	A	2.	WAP to print first n odd numbers using for, while and do while loop.
	A	3.	WAP to demonstrate foreach loop.
	B	4.	WAP to generate Fibonacci series of N number.
	B	5.	WAP to calculate and display sum and product of first N number.
	B	6.	WAP to check whether the given number is prime or not.
	C	7.	WAP to check given number is palindrome or not.
	C	8.	WAP to check given number is Armstrong or not.

<table>
<tr><th colspan="4">Practical -7: Implementation of nested loops in PHP</th></tr>
<tr><td rowspan="3">LAB-7</td><td>A</td><td>1.</td><td>WAP to print following patterns:

<pre>
*   1   5   1
* *   1 2   5 4   2 2
* * *   1 2 3   5 4 3   3 3 3
* * * *   1 2 3 4   5 4 3 2   4 4 4 4
* * * * * 1 2 3 4 5 5 4 3 2 1 5 5 5 5 5
  (a)   (b)   (c)   (d)
</pre></td></tr>
<tr><td>B</td><td>2.</td><td>WAP to print following patterns:

<pre>
    *   * * * * *   *****
   **   * * * *   ****
  ***   * * *   ***
 ****   * *   **
*****   *   *
  (a)   (b)   (c)
</pre></td></tr>
<tr><td>C</td><td>3.</td><td>WAP to print following patterns:

<pre>
1   *   1 2 3 4 5
2 2   * *   2 3 4 5
3 3 3   * * *   3 4 5
4 4 4 4   * * * *   4 5
5 5 5 5 5 * * * * *   5
(a)   (b)   (c)
</pre></td></tr>
<tr><th colspan="4">Practical -8: Implementation of array (Part – I)</th></tr>
<tr><td rowspan="4">LAB-8</td><td>A</td><td>1.</td><td>WAP to create numeric array and print it.</td></tr>
<tr><td>A</td><td>2.</td><td>WAP to create associative array and print it.</td></tr>
<tr><td>B</td><td>3.</td><td>WAP to create multidimensional array and print it.</td></tr>
<tr><td>B</td><td>4.</td><td>Create a PHP script that performs the following tasks using a numeric array of student scores (derived from the previous associative array):

$scores = [85, 78, 92, 67, 90];

- Shuffle the scores randomly and display the result.
- Add a new score to the end of the list.
- Remove the last score.
- Remove the first score.
- Add a new score to the beginning of the list.</td></tr>
</table>

	C	5.	Create a PHP script that performs the following tasks using the associative array of student names and their scores: $students = ["Alice" => 85, "Bob" => 78, "Charlie" => 92, "David" => 67, "Eve" => 90]; - Display the total number of students. - Print all student names and all scores separately. - Calculate and display the total and product of all scores. - Find the lowest and highest score in the list. - Check if any student has a score of 90. - Find the student who scored 92. - Sort the scores in ascending and descending order. - Sort the student names in alphabetical and reverse alphabetical order.
Practical -9: Implementation of array (Part – II)			
LAB-9	A	1.	WAP to count number of even or odd number from an array of n number.
	A	2.	WAP to find the average of n number using array.
	B	3.	WAP to accept n numbers in an array. Display the sum of all the numbers which are divisible by either 3 or 5.
	C	4.	WAP to accept n numbers in an array. Now, enter a number and search whether the number is present or not in the list of array elements by using linear search.
Practical -10: Demonstration of function (Part – I)			
LAB-10	A	1.	WAP to create user de ne function for print your name and call it using all user defined functions.
	A	2.	WAP to create user define function for adding two numbers and display the result.
	A	3.	WAP to create calculator using all four types of UDF.
	B	4.	WAP to calculate simple interest using method.
	C	5.	WAP to generate Fibonacci series of N given number using method.
Practical -11: Demonstration of function (Part – II)			
LAB-11	A	1.	WAP that calculates area of circle, triangle and square using user defined function.
	A	2.	WAP to find maximum number from given three numbers using user defined function.

	B	3.	WAP to accept a number and check whether the number is prime or not. Use user defined function name as check (int n). The method returns 1, if the number is prime otherwise, it returns 0.
	C	4.	WAP to take two values as an input from the user and display all the prime numbers between the two given numbers using function.
Practical -12: Implementation of recursion and Web page partition			
LAB-12	A	1.	WAP to calculate sum of first n numbers using recursion.
	A	2.	WAP to demonstrate the use of include, require, include_once and require_once.
	B	3.	WAP to calculate factorial of a number using recursion.
	C	4.	Demonstrate the use of PHP's include, require, include_once, and require_once functions to partition a webpage into different sections in Lab 2(3).
	C	5.	WAP to check whether the number is prime or not using recursion.
Practical -13: Implementation of form processing and String function			
LAB-13	A	1.	Design a student registration form and retrieve data in controller page using following Method: GET, POST, and REQUEST.
	A	2.	Create a PHP script that performs the following tasks using given a string with some text, and you need to use the string functions listed below to perform various operations on it. $text = " PHP is a powerful scripting language for web development. "; - Find the length of the string. - Convert the string to uppercase and display the result. - Convert the string to lowercase and display the result. - Find the position of the word "powerful" in the string. - Replace the word "scripting" with the word "programming". - Extract the word "PHP" from the string. - Reverse the string and display the result. - Trim the leading and trailing spaces from the string. - Repeat the string "PHP " 5 times and display the result. - Implode the array of words ["PHP", "is", "a", "powerful", "language"] into a string, with a space separating each word. - Explode the string into an array using spaces as the delimiter, and display the array. - Shuffle the characters of the string randomly and display the result. - Convert newlines in a string to tags. Use the string "Hello\nWorld" for this task.
	B	3.	Design an employee registration form and retrieve data in controller page using following Method: GET, POST, and REQUEST.
	C	4.	Create form with all input types and retrieve data in controller page using following Method: GET, POST, and REQUEST.

Practical -14: Demonstration of Regular Expression and Server-side validation			
LAB -14	A	1.	Implement server-side validation on student registration form using PHP.
	B	2.	Implement server-side validation on employee registration form using PHP.
	C	3.	Implement server-side validation on form with all input types using PHP.
Practical -15: Demonstration of file upload			
LAB-15	A	1.	Create a webpage which accepts a file and upload it in specified folder on server.
	B	2.	Design a profile page which allows changing profile picture dynamically.
	C	3.	Create a webpage which accepts an image file in .jpg or .jpeg or .png format only and that to maximum of 1MB.
Practical -16: Implementation of files			
LAB-16	A	1.	WAP to check the given file name is exits or not if exits than print the size of the give file.
	A	2.	Write a program to create file named “students.txt” to store names of the students.
	A	3.	Write a program to read names of the students from “students.txt” file.
	A	4.	Write a program to create file named “employee.txt” to store empno, name, gender and mobileno. Open the same file again to display the content of the file.
	B	5.	Write a file named “teachers.txt” and copy the content of file named “employee.txt” in it.
	B	6.	Write a program to delete a file named “employee.txt”.
	C	7.	Write a program to append the file named “students.txt” to add details of 3 new students in a file.
	C	8.	Write a program to read a csv file named “result.csv”.
Practical -17: Demonstration of session management using COOKIES and SESSION			
LAB-17	A	1.	WAP to demonstrate the use of COOKIE.
	A	2.	WAP to demonstrate the use of SESSION.
	B	3.	Create static login application using COOKIE in PHP.
	B	4.	Create static login application using SESSION in PHP.
	C	5.	Create a webpage showing multiple items with add to cart and purchase functionality and on click of purchase it will redirect user to bill page that will show the detailed bill using COOKIES.

	C	6.	Create a webpage showing multiple items with add to cart and purchase functionality and on click of purchase it will redirect user to bill page that will show the detailed bill using SESSION.
Practical -18: Demonstration of basic SQL commands and PHP functions for database connectivity			
LAB-18	A	1.	Understanding and working with Basic SQL Commands.
	A	2.	WAP to work with all the basic SQL commands on a sample database.
	B	3.	Understanding and working with PHP functions for database connectivity.
	C	4.	WAP to check database connectivity on a sample database.
Practical -19: Implementation of CRUD operation using PHP (Part – I)			
LAB-19	A	1.	Implement Select and Insert database operation on students table from webpage using PHP.
	A	2.	Implement Update and Delete database operation on students table from webpage using PHP.
	B	3.	Create a Student Registration form to implement insert and select database operation.
Practical -20: Implementation of CRUD operation using PHP (Part – II)			
LAB-20	A	1.	Implement select and insert database operation on employees table from webpage using PHP.
	A	2.	Implement update and Delete database operation on employees table from webpage using PHP.
	B	3.	Modify the program Lab 19(3) of registration form to implement update and delete database operation.
Practical -21: Demonstration of prepared statement and stored procedure			
LAB-21	A	1.	Implement CRUD operations on students table using prepared statement in PHP.
	A	2.	Implement CRUD operations on students table using stored procedure in PHP.
	B	3.	Modify the program Lab 19(3) and Lab 20(3) to implement prepared statement in it.
	B	4.	Modify the program Lab 19(3) and Lab 20(3) to implement stored procedure in it.
Practical -22: Implementation of CRUD, SESSION/COOKIES, Prepared Statement, Stored Procedure in a website			
LAB-22	A	1.	Implement server-side validation on login form using PHP.
	B	2.	Apply the prepared statement/stored procedure in login page created in Lab 22(1).
	C	3.	Implement cookie/session to provide remember me functionality in login page created in Lab 22(1).

Practical -23: Demonstration of WordPress installation, post creation and post publishing and WordPress pages			
LAB-23	A	1.	Download and install WordPress on the XAMPP.
	A	2.	Create minimum 3 posts in WordPress and publish it with text and images. Use different post formats depending on what you want to publish.
	A	3.	Create minimum 2 pages in WordPress and publish. Use different page formats depending on the content that is to be published.
	B	4.	Create 2 posts and pages for travelling website.
	C	5.	Create 2 posts and pages for ecommerce website.
Practical -24: Demonstration of WordPress menus			
LAB-24	A	1.	Create, edit, and change categories and Tags for Multiple Posts and pages.
	B	2.	Create, edit, and change categories and Tags for travelling website pages created in Lab 23(4).
	C	3.	Create, edit, and change categories and Tags for ecommerce website pages created in Lab 23 (5).
Practical -25: Demonstration of WordPress widget and WordPress theme			
LAB-25	A	1.	Use different Widgets depending on the posts and pages.
	A	2.	Set the core settings of WordPress for the given content and add a new user.
	A	3.	Install and activate the Theme in WordPress and Create 2-3 different pages using different themes.
	B	4.	Use different Widgets and apply theme for travelling website pages created in Lab 23 (4).
	C	5.	Use different Widgets and apply theme for ecommerce website pages created in Lab 23 (5).
Practical -26: Demonstration of various plugins of WordPress and Implementation of WordPress in a website			
LAB-26	A	1.	Install and activate various Plugins in WordPress. Create content writing for E-commerce sites and product description that sells online. Install and activate various Plugins in WordPress. Design coupon code for selling product and upload stock of item on site.
	A	2.	Create content writing for its website using various plugins.
	B	3.	Update and continue working in travelling website using any plugins.
	C	4.	Update and continue working in ecommerce website using any plugins.