DAYANANDA SAGAR ACADEMY OF TECHNOLOGY AND MANAGEMENT

DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS

WEB TECHNOLOGY LAB MANUAL

Academic year 2024-2025

II Semester
(even semester)

COURSE CODE: MMC205

Prepared by

Mr. BHARATH KUMARA J

ASSISTANT PROFESSOR
MCA, DSATM

HOD-MCA

Dr. MANJULA SANJAY

PRINCIPAL

Dr. RAVISHANKAR M

VISION OF THE DEPARTMENT

Nurture Continuous Learning through research and innovations in the field of Computer Science, Technology and Applications, to build competent professionals

MISSION OF THE DEPARTMENT

Create a learning environment to motivate students to build strong technology skills.

Promote value based ethical practices in all facets of learning.

Instil Entrepreneurial collaborative thinking through structured interventions and industry participation.

PROGRAM EDUCATION OBJECTIVES (PEO'S):

PEO1: Analyse real life problems, design computing systems appropriate to its solutions that are technically sound, economically feasible and socially acceptable.

PEO2: Exhibit professionalism, ethical attitude, communication skills, team work in their profession and adapt to current trends by engaging in lifelong learning.

PEO3: Demonstrate Leadership and Entrepreneurship Skills by incorporating organizational goals.

PROGRAM OUTCOMES (PO's)

MCA Graduates will be able to:

1. Foundation Knowledge: Apply knowledge of mathematics, programming logic and coding fundamentals for solution architecture and problem solving.

2. Problem Analysis: Identify, review, formulate and analyse problems for primarily focussing on customer requirements using critical thinking frameworks.

3. Development of Solutions: Design, develop and investigate problems with as an innovative approach for solutions incorporating ESG/SDG goals.

4. Modern Tool Usage: Select, adapt and apply modern computational tools such as development of algorithms with an understanding of the limitations including human biases.

5. Individual and Teamwork: Function and communicate effectively as an individual or a team leader in diverse and multidisciplinary groups. Use methodologies such as agile.

6. Project Management and Finance: Use the principles of project management such as scheduling, work breakdown structure and be conversant with the principles of Finance for profitable project management.

7. Ethics: Commit to professional ethics in managing software projects with financial aspects. Learn to use new technologies for cyber security and insulate customers from malware

8. Life-long learning: Change management skills and the ability to learn, keep up with contemporary technologies and ways of working.

PROGRAM SPECIFIC OUTCOMES (PSO's)

PSO 1: The graduates of the Program will have skills to develop, deploy and maintain applications for desktop, web, mobile, cloud and cross platforms using modern tools and technologies.

PSO 2: The graduates of the program analyse the societal needs to provide novel solutions through technological based research.

COURSE OUTCOMES (CO'S)

CO	Course Outcomes	RBT Level	Level Indicator
CO1	Understand the concept and usages of web-based programming techniques. XML, HTML, CSS, JavaScript, jQuery	Understand	L1, L2
CO2	Apply the features of web programming using Html, CSS, Java script, jQuery And Bootstrap for the given web-based problem	Apply	L3
CO3	Analyze a web page and identify its elements attributes of HTML, XML, Bootstrap, Java script and j query	Analyze	L4
CO4	Design applications of Angular JS and jQuery for the given problem and build dynamic web pages using Client-side programming	Evaluate	L5
CO5	Create user interactive dynamic web-based applications.	Implement	L6

COMPUTER LAB RULES AND REGULATIONS

DO's

- Come prepared to the Lab.
- Submit your Records to the faculty and sign in the Log Book on entering the Lab
- Observation books have to be brought for all the labs.
- Backlog exercises to be executed after completing regular exercises.
- Regularly attend all the labs
- Put the chairs back to its position before you leave.
- Treat all the devices with care and consideration.
- Behave in a responsible manner at all the times and maintain silence.
- Before leaving the lab shut down the system and rearrange the chairs
- Keep your premises clean

DON'T

- Use Mobile phones and pen drives
- Move around in the lab during the lab session.
- Tamper System Files or Try to access the Server.
- Write Records in the Lab
- Change the system assigned to you without the notice of the Lab Staff.
- Write on the table or mouse pads.
- Do not install or download any software or modify or delete any system files on any lab computers.

EVALUATION PROCEDURE

TOTAL MARKS: 20

Type of Assessments			**Maximum Marks**	**Average**	**Reduced Marks**	**Minimum Passing Marks**	**Evaluation Details**
Conduction of Experiments	Performance-Continuous Evaluation of each experiment	05	10	Average of all Experiments	10	5	Performance of the Experiment (On completion of every experiment/program in the laboratory, the students shall be evaluated and marks shall be awarded on the same day. 10 marks are for conducting the experiment and calculations/observations/output)
	Record	03					
	Observation book	02					
Practical Test	Write up	15	50	----	05	3	One Internal Practical Test after conduction of all Experiments for 50 Marks
	Execution	25					
	Viva-voce	10					
Open Ended Experiment	Write up	05	20	----	05	2	One experiment for 20 marks. 20 marks reduced to 05 marks
	Execution	10					
	Viva-voce	05					
Total CIE Practical					**20**	**10**	Scale down Marks of Experiments, Record, Observation, Practical Test and Open-Ended Experiment

LIST OF EXPERIMENTS OR PROGRAMS

Sl.no	Experiments/Programs	COs
1	Create your Resume using Html & CSS	**CO1**
2	Create a registration form using HTML and CSS with Database connectivity	**CO3**
3	Write a Bootstrap program to demonstrate Cards with Data Insertion	**CO3**
4	Write a PHP program to connect to a MySQL database which retrieves the data from the tables and displays them to the user.	**CO4**
5	Write a Java script to design a simple calculator to perform the following operations: sum, product, difference, and quotient.	**CO2**
6	Write an Angular JS application to calculate Factorial and compute Square based on given user input	**CO3**
7	Write a JS code to validate the user login page	**CO3**
8	Write an Angular JS program to create simple CRUD applications for managing users.	**CO3**
9	a. Write a PHP program to hit counter using cookies b. Write a PHP program to calculate Date and Time function.	**CO4**
10	Create a web page for software company website	**CO4**

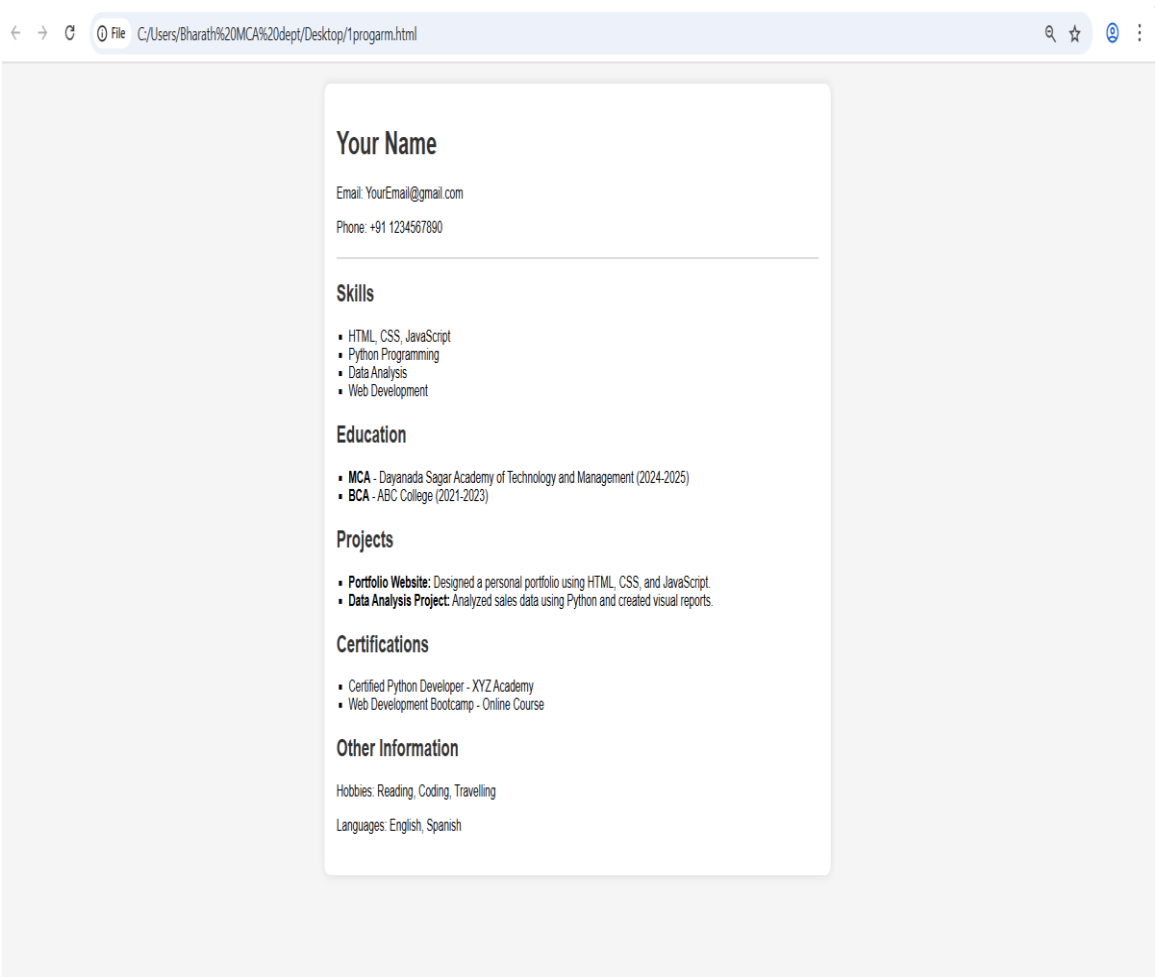
1. Create your resume using HTML and CSS

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>My Resume</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 20px;
            background-color: #f5f5f5;
        }
        .resume {
            background-color: #fff;
            padding: 20px;
            border-radius: 10px;
            max-width: 800px;
            margin: auto;
            box-shadow: 0 0 10px rgba(0,0,0,0.1);
        }
        h1, h2 {
            color: #333;
        }
        .contact-info, .skills, .education, .projects, .certifications, .other {
            margin-bottom: 20px;
        }
        ul {
            list-style-type: square;
            padding-left: 20px;
        }
```

```
        hr {
            border: 1px solid #ddd;
        }
    </style>
</head>
<body>
    <div class="resume">
        <h1>Your Name</h1>
        <div class="contact-info">
            <p>Email: YourEmail@gmail.com</p>
            <p>Phone: +91 1234567890</p>
        </div>

        <hr>

        <div class="skills">
            <h2>Skills</h2>
            <ul>
                <li>HTML, CSS, JavaScript</li>
                <li>Python Programming</li>
                <li>Data Analysis</li>
                <li>Web Development</li>
            </ul>
        </div>

        <div class="education">
            <h2>Education</h2>
            <ul>
                <li><strong>MCA</strong> - Dayanada Sagar Academy of Technology and Management (2024-2025)</li>
                <li><strong>BCA</strong> - ABC College (2021-2023)</li>
            </ul>
```

```
        </div>

        <div class="projects">
            <h2>Projects</h2>
            <ul>
                <li><strong>Portfolio Website:</strong> Designed a personal portfolio using HTML, CSS, and JavaScript.</li>
                <li><strong>Data Analysis Project:</strong> Analyzed sales data using Python and created visual reports.</li>
            </ul>
        </div>

        <div class="certifications">
            <h2>Certifications</h2>
            <ul>
                <li>Certified Python Developer - XYZ Academy</li>
                <li>Web Development Bootcamp - Online Course</li>
            </ul>
        </div>

        <div class="other">
            <h2>Other Information</h2>
            <p>Hobbies: Reading, Coding, Travelling</p>
            <p>Languages: English, Spanish</p>
        </div>
    </div>
</body>
</html>
```

OUTPUT


File
C:/Users/Bharath%20MCA%20dept/Desktop/1progarm.html
Your Name
Email: YourEmail@gmail.com
Phone: +91 1234567890
Skills
HTML, CSS, JavaScript
Python Programming
Data Analysis
Web Development
Education
MCA - Dayanada Sagar Academy of Technology and Management (2024-2025)
BCA - ABC College (2021-2023)
Projects
Portfolio Website: Designed a personal portfolio using HTML, CSS, and JavaScript.
Data Analysis Project: Analyzed sales data using Python and created visual reports.
Certifications
Certified Python Developer - XYZ Academy
Web Development Bootcamp - Online Course
Other Information
Hobbies: Reading, Coding, Travelling
Languages: English, Spanish

2. Create a registration form using HTML and CSS with Database connectivity

form.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Registration Form</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      background: #f1f1f1;
      display: flex;
      justify-content: center;
      align-items: center;
      height: 100vh;
    }
    .container {
      background: white;
      padding: 30px;
      border-radius: 8px;
      box-shadow: 0 4px 8px rgba(0,0,0,0.1);
      width: 350px;
    }
    h2 {
      text-align: center;
    }
    input[type="text"],
    input[type="email"],
    input[type="password"],
    input[type="tel"] {
      width: 100%;
      padding: 10px;
      margin: 8px 0 16px;
      border: 1px solid #ccc;
      border-radius: 4px;
    }
    .gender {
      margin-bottom: 16px;
    }
    input[type="submit"] {
      background-color: #4CAF50;
      color: white;
      border: none;
      padding: 12px;
      width: 100%;
      border-radius: 4px;
      cursor: pointer;
    }
```

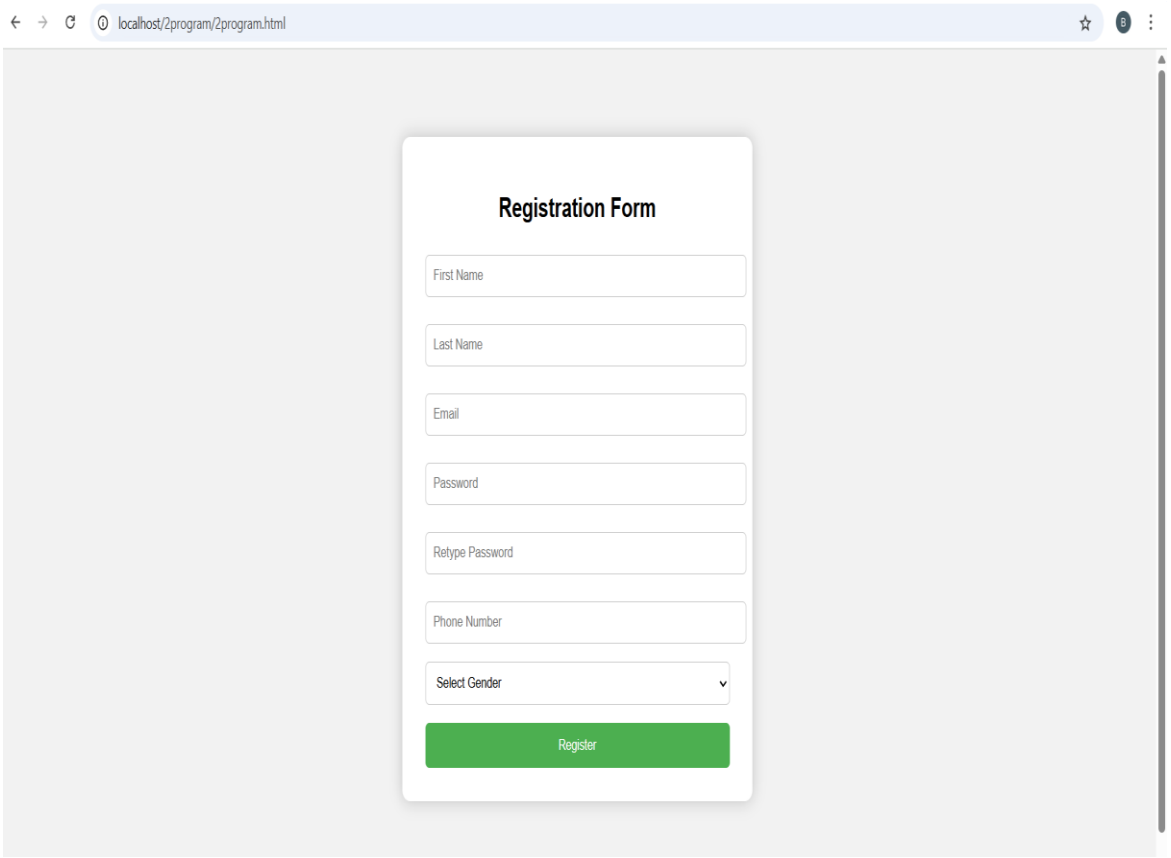
```
    input[type="submit"]:hover {
      background-color: #45a049;
    }
  </style>
</head>
<body>
  <div class="container">
    <h2>Register</h2>
    <form action="register.php" method="post">
      <input type="text" name="first_name" placeholder="First Name" required>
      <input type="text" name="last_name" placeholder="Last Name" required>
      <input type="email" name="email" placeholder="Email" required>
      <input type="password" name="password" placeholder="Password" required>
      <input type="password" name="retype_password" placeholder="Retype Password"
required>
      <input type="tel" name="phone" placeholder="Phone Number" required>

      <div class="gender">
        <label>Gender:</label><br>
        <input type="radio" name="gender" value="Male" required> Male
        <input type="radio" name="gender" value="Female" required> Female
        <input type="radio" name="gender" value="Other" required> Other
      </div>

      <input type="submit" value="Register">
    </form>
  </div>
</body>
</html>
```

Register.php

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "registration_db";

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

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

// Collect form data
$first_name = $_POST['first_name'];
$last_name = $_POST['last_name'];
$email = $_POST['email'];
```

```
$password = $_POST['password'];
$confirm_password = $_POST['confirm_password'];
$phone = $_POST['phone'];
$gender = $_POST['gender'];

// Password match check
if ($password !== $confirm_password) {
    die("Passwords do not match!");
}

// Hash the password for security
$hashed_password = password_hash($password, PASSWORD_DEFAULT);

// Insert into database
$sql = "INSERT INTO users (first_name, last_name, email, password, phone, gender)
        VALUES ('$first_name', '$last_name', '$email', '$hashed_password', '$phone',
'$gender')";

if ($conn->query($sql) === TRUE) {
  echo "Registration successful!";
} else {
  echo "Error: " . $sql . "<br>" . $conn->error;
}

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

OUTPUT

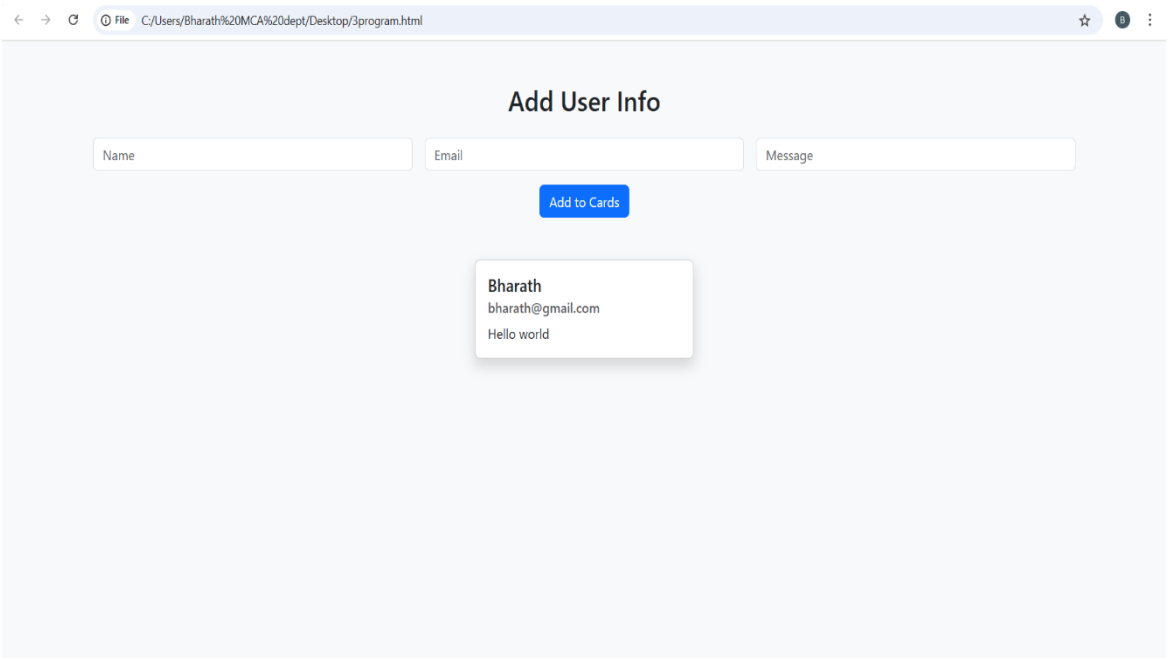


3. Write a Bootstrap program to demonstrate Cards with Data Insertion

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Bootstrap Cards with Data Insertion</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet">
  <style>
    body {
      background-color: #f8f9fa;
    }
    .card-container {
      display: flex;
      flex-wrap: wrap;
      gap: 20px;
    }
  </style>
</head>
<body>
  <div class="container py-5">
    <h2 class="text-center mb-4">Add User Info</h2>
    <form id="dataForm" class="row g-3 mb-5">
      <div class="col-md-4">
        <input type="text" id="name" class="form-control" placeholder="Name" required>
      </div>
      <div class="col-md-4">
        <input type="email" id="email" class="form-control" placeholder="Email" required>
      </div>
      <div class="col-md-4">
        <input type="text" id="message" class="form-control" placeholder="Message"
required>
      </div>
      <div class="col-12 text-center">
        <button type="submit" class="btn btn-primary">Add to Cards</button>
      </div>
    </form>

    <div id="cardContainer" class="card-container justify-content-center"></div>
  </div>

  <script>
    document.getElementById('dataForm').addEventListener('submit', function(e) {
      e.preventDefault();

      const name = document.getElementById('name').value.trim();
      const email = document.getElementById('email').value.trim();
      const message = document.getElementById('message').value.trim();
```

```
    if (name && email && message) {
      const cardHTML = `
        <div class="card shadow" style="width: 18rem;">
          <div class="card-body">
            <h5 class="card-title">${name}</h5>
            <h6 class="card-subtitle mb-2 text-muted">${email}</h6>
            <p class="card-text">${message}</p>
          </div>
        </div>
      `;

      document.getElementById('cardContainer').insertAdjacentHTML('beforeend',
cardHTML);

      // Clear form
      document.getElementById('dataForm').reset();
    }
  });
 </script>

  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>
```

OUTPUT

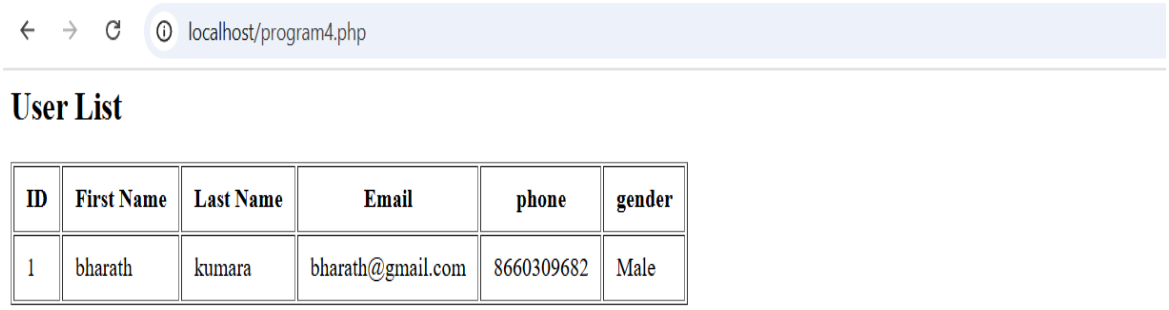


4.Write a PHP program to connect to a MySQL database which retrieves the data from the tables and displays them to the user.

```
<?php
// Database configuration
$servername = "localhost"; // or your host
$username = "root";        // default username for XAMPP/WAMP
$password = "";            // default is empty in local setups
$dbname = "registration_db";   // replace with your actual database name

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

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

// SQL query to fetch data
$sql = "SELECT id, first_name,last_name, email,phone,gender FROM users";
$result = $conn->query($sql);

echo "<h2>User List</h2>";

if ($result->num_rows > 0) {
    echo "<table border='1' cellpadding='10'>";
    echo "<tr><th>ID</th><th>First Name</th><th>Last
Name</th><th>Email</th><th>phone</th><th>gender</th></tr>";

    // Output data of each row
```

```
    while($row = $result->fetch_assoc()) {
        echo "<tr>";
        echo "<td>". htmlspecialchars($row["id"]) ."</td>";
        echo "<td>". htmlspecialchars($row["first_name"]) ."</td>";
            echo "<td>". htmlspecialchars($row["last_name"]) ."</td>";
            echo "<td>". htmlspecialchars($row["email"]) ."</td>";
            echo "<td>". htmlspecialchars($row["phone"]) ."</td>";
        echo "<td>". htmlspecialchars($row["gender"]) ."</td>";
        echo "</tr>";
    }
    echo "</table>";
} else {
    echo "No records found.";
}
// Close the connection
$conn->close();
?>
```

OUTPUT



5.Write a Java script to design a simple calculator to perform the following operations: sum, product, difference, and quotient.

```
<!DOCTYPE html>
<html>
<head>
  <title>Simple Calculator</title>
  <style>
    body {
      font-family: sans-serif;
      background: #f2f2f2;
      display: flex;
      justify-content: center;
      align-items: center;
      height: 100vh;
    }

    .calculator {
      background: white;
      padding: 20px;
      border-radius: 10px;
      box-shadow: 0 0 10px rgba(0,0,0,0.1);
      text-align: center;
    }

    #display {
      width: 220px;
      height: 40px;
      font-size: 18px;
      text-align: right;
      margin-bottom: 15px;
      padding: 5px;
    }

    .btn {
      width: 50px;
      height: 50px;
      font-size: 18px;
      margin: 5px;
      cursor: pointer;
    }

    .btn-row {
      display: flex;
      justify-content: center;
    }
  </style>
</head>
<body>
```

```
<div class="calculator">
  <input type="text" id="display" disabled>

  <div class="btn-row">
    <button class="btn" onclick="clearDisplay()">C</button>
    <button class="btn" onclick="backspace()">&lt</button>
    <button class="btn" onclick="append('/')">/</button>
    <button class="btn" onclick="append('*')">*</button>
  </div>
  <div class="btn-row">
    <button class="btn" onclick="append('7')">7</button>
    <button class="btn" onclick="append('8')">8</button>
    <button class="btn" onclick="append('9')">9</button>
    <button class="btn" onclick="append('-')">-</button>
  </div>
  <div class="btn-row">
    <button class="btn" onclick="append('4')">4</button>
    <button class="btn" onclick="append('5')">5</button>
    <button class="btn" onclick="append('6')">6</button>
    <button class="btn" onclick="append('+')">+</button>
  </div>
  <div class="btn-row">
    <button class="btn" onclick="append('1')">1</button>
    <button class="btn" onclick="append('2')">2</button>
    <button class="btn" onclick="append('3')">3</button>
    <button class="btn" onclick="calculate()">=</button>
  </div>
  <div class="btn-row">
    <button class="btn" onclick="append('0')">0</button>
    <button class="btn" onclick="append('.')">.</button>
  </div>
</div>

<script>
  const display = document.getElementById("display");

  function append(value) {
    display.value += value;
  }

  function clearDisplay() {
    display.value = "";
  }

  function backspace() {
    display. Value = display.value.slice(0, -1);
  }

  function calculate() {
    try {
```

```
      display.value = eval(display.value);
    } catch {
      display.value = "Error";
    }
  }

  // Allow keyboard typing
  document.addEventListener("keydown", function(e) {
    const key = e.key;
    if (!isNaN(key) || "+-*/.".includes(key)) {
      append(key);
    } else if (key === "Enter") {
      calculate();
    } else if (key === "Backspace") {
      backspace();
    } else if (key === "Escape") {
      clearDisplay();
    }
  });
</script>

</body>
</html>
```

OUTPUT

6.Write an Angular JS application to calculate Factorial and compute Square based on given user input

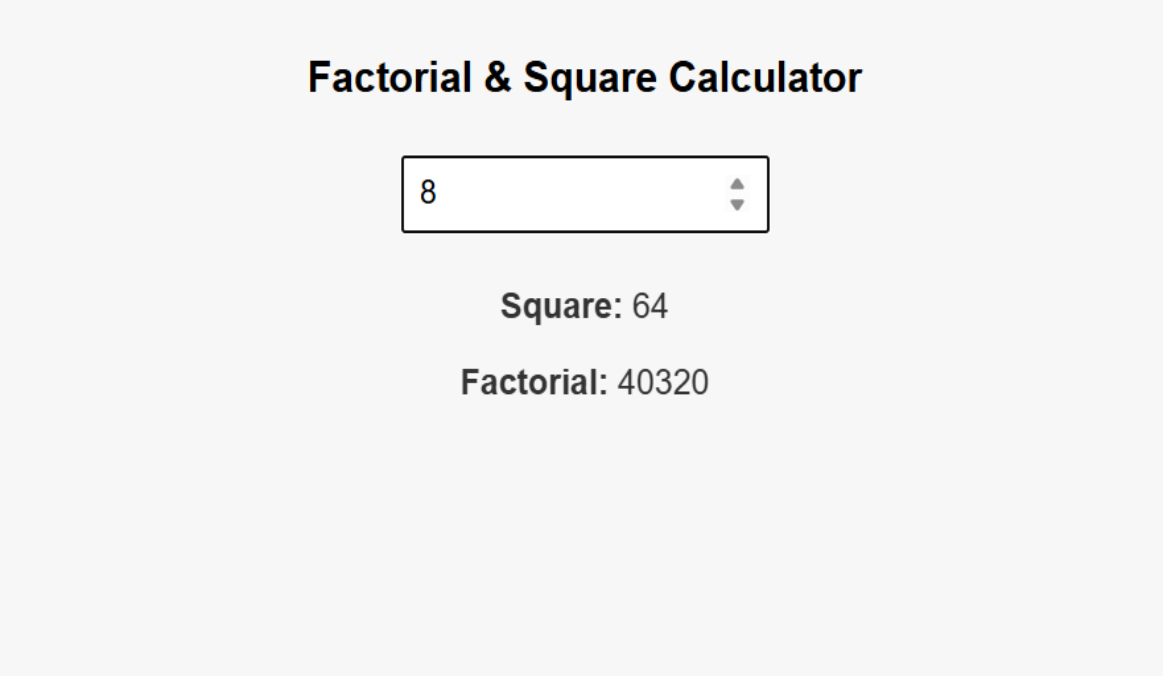
```
<!DOCTYPE html>
<html ng-app="calcApp">
<head>
  <meta charset="UTF-8">
  <title>Factorial & Square Calculator</title>
  <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
  <style>
    body {
      font-family: Arial, sans-serif;
      padding: 30px;
      background: #f7f7f7;
      text-align: center;
    }
    input {
      padding: 10px;
      font-size: 18px;
      width: 200px;
      margin: 10px;
    }
    .result {
      margin-top: 20px;
      font-size: 20px;
      color: #333;
    }
  </style>
```

```
</head>
<body ng-controller="CalcController">

  <h2>Factorial & Square Calculator</h2>

  <input type="number" ng-model="number" placeholder="Enter a number">

  <div class="result" ng-if="number !== undefined">
    <p><strong>Square:</strong> {{ number * number }}</p>
    <p><strong>Factorial:</strong> {{ factorial(number) }}</p>
  </div>

  <script>
    angular.module('calcApp', [])
    .controller('CalcController', ['$scope', function($scope) {

      // Factorial function
      $scope.factorial = function(n) {
        if (n < 0) return 'Undefined';
        if (n === 0 || n === 1) return 1;
        let result = 1;
        for (let i = 2; i <= n; i++) {
          result *= i;
        }
        return result;
      };

    }]);
  </script>
```

</body>

</html>

OUTPUT



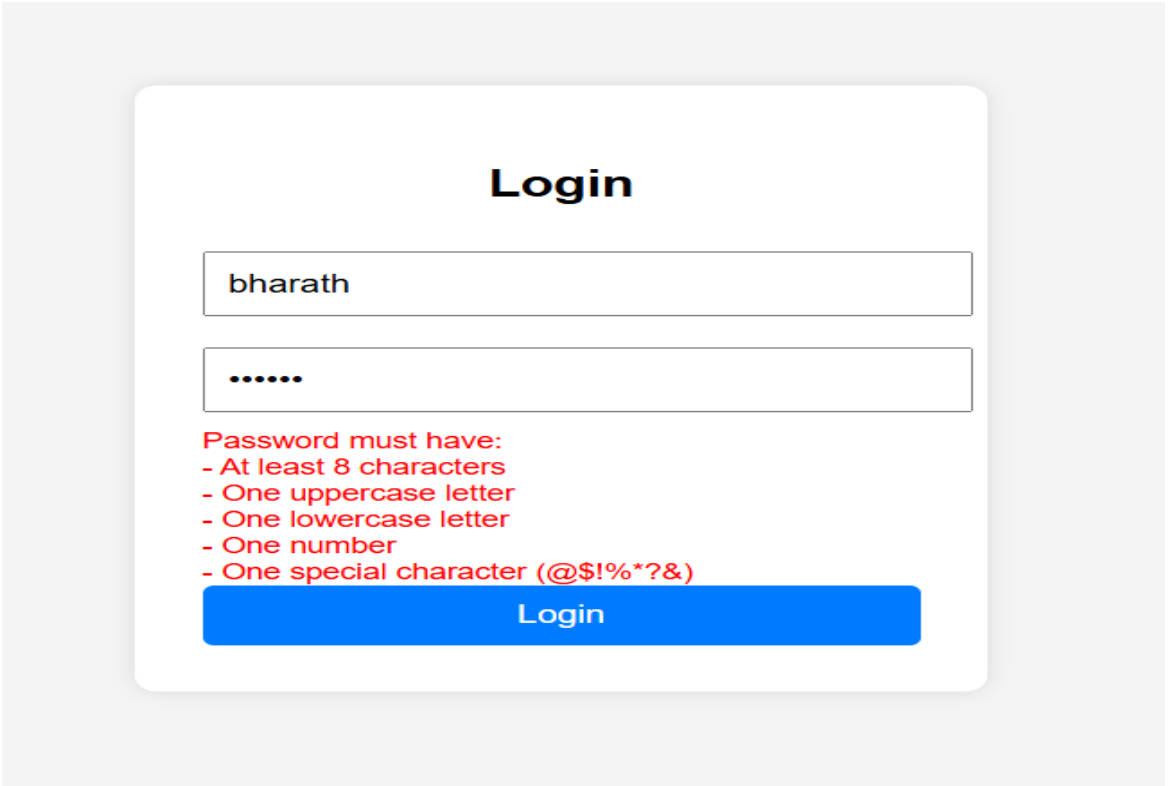
7.Write a JS code to validate the user login page

```
<!DOCTYPE html>
<html>
<head>
   <title>Login Validation</title>
   <style>
      body {
         font-family: Arial, sans-serif;
         background: #f4f4f4;
         display: flex;
         justify-content: center;
         align-items: center;
         height: 100vh;
      }

      .login-box {
         background: white;
         padding: 30px;
         border-radius: 10px;
         box-shadow: 0 0 10px rgba(0,0,0,0.1);
         width: 320px;
      }

      .login-box h2 {
         text-align: center;
      }

      input {
         width: 100%;
         padding: 10px;
         margin: 10px 0;
         font-size: 16px;
      }

      .error {
         color: red;
         font-size: 14px;
      }

      button {
```

```
            width: 100%;
            padding: 10px;
            background: #007bff;
            color: white;
            border: none;
            font-size: 16px;
            border-radius: 5px;
            cursor: pointer;
        }

        button:hover {
            background: #0056b3;
        }
    </style>
</head>
<body>

<div class="login-box">
    <h2>Login</h2>
    <form id="loginForm" onsubmit="return validateLogin()">
        <input type="text" id="username" placeholder="Username">
        <input type="password" id="password" placeholder="Password">
        <div class="error" id="errorMsg"></div>
        <button type="submit">Login</button>
    </form>
</div>

<script>
    function validateLogin() {
        const username = document.getElementById("username").value.trim();
        const password = document.getElementById("password").value.trim();
        const errorMsg = document.getElementById("errorMsg");

        // Check if fields are empty
        if (username === "" || password === "") {
            errorMsg.textContent = "Both username and password are required.";
            return false;
        }

        // Password strength check
        const passwordRegex = /^(?=.*[a-z])(?=.*[A-
Z])(?=.*\d)(?=.*[@$!%*?&])[A-Za-z\d@$!%*?&]{8,}$/;
```

```
        if (!passwordRegex.test(password)) {
            errorMsg.innerHTML = `
                Password must have:<br>
                - At least 8 characters<br>
                - One uppercase letter<br>
                - One lowercase letter<br>
                - One number<br>
                - One special character (@$!%*?&)
            `;
            return false;
        }

        // If all validations pass
        errorMsg.textContent = "";
        alert("Login successful! ");
        return false; // prevent actual form submission in this demo
    }
</script>

</body>
</html>
```

OUTPUT



8. Write an Angular JS program to create simple CRUD applications for managing users.

```
<!DOCTYPE html>
<html ng-app="userApp">
<head>
  <meta charset="UTF-8">
  <title>AngularJS User CRUD</title>
  <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></scr
ipt>
  <style>
    body {
      font-family: Arial;
      margin: 30px;
      background: #f4f4f4;
    }
    .form-section, table {
      background: white;
      padding: 20px;
      border-radius: 10px;
      margin-bottom: 30px;
      box-shadow: 0 2px 8px rgba(0,0,0,0.1);
    }
    input {
      padding: 10px;
      margin: 5px;
      width: 200px;
    }
    button {
      padding: 8px 12px;
      margin: 5px;
      border: none;
      border-radius: 5px;
      cursor: pointer;
    }
    .btn-add { background: #28a745; color: white; }
    .btn-update { background: #ffc107; color: black; }
    .btn-delete { background: #dc3545; color: white; }
  </style>
</head>
<body ng-controller="UserController">
```

```
  <h2>User Management CRUD App (AngularJS)</h2>

  <div class="form-section">
    <h3>{{ isEdit ? "Update User" : "Add New User" }}</h3>
    <input type="text" ng-model="user.name" placeholder="Name">
    <input type="email" ng-model="user.email" placeholder="Email">
    <button class="btn-add" ng-click="saveUser()" ng-if="!isEdit">Add
User</button>
    <button class="btn-update" ng-click="updateUser()" ng-if="isEdit">Update
User</button>
  </div>

  <div class="table-section">
    <table border="1" width="100%" cellpadding="10">
      <thead>
        <tr>
          <th>#</th>
          <th>Name</th>
          <th>Email</th>
          <th>Actions</th>
        </tr>
      </thead>
      <tbody>
        <tr ng-repeat="u in users">
          <td>{{$index + 1}}</td>
          <td>{{u.name}}</td>
          <td>{{u.email}}</td>
          <td>
            <button class="btn-update" ng-click="editUser($index)">Edit</button>
            <button class="btn-delete" ng-
click="deleteUser($index)">Delete</button>
          </td>
        </tr>
        <tr ng-if="users.length === 0">
          <td colspan="4" align="center">No users available</td>
        </tr>
      </tbody>
    </table>
  </div>

  <script>
    angular.module('userApp', [])
```

```
    .controller('UserController', function($scope) {
      $scope.users = []; // initial empty list
      $scope.user = {};
      $scope.isEdit = false;
      let editingIndex = -1;

      $scope.saveUser = function() {
        if ($scope.user.name && $scope.user.email) {
          $scope.users.push(angular.copy($scope.user));
          $scope.user = {};
        }
      };

      $scope.editUser = function(index) {
        $scope.isEdit = true;
        editingIndex = index;
        $scope.user = angular.copy($scope.users[index]);
      };

      $scope.updateUser = function() {
        if ($scope.user.name && $scope.user.email) {
          $scope.users[editingIndex] = angular.copy($scope.user);
          $scope.user = {};
          $scope.isEdit = false;
        }
      };

      $scope.deleteUser = function(index) {
        if (confirm("Are you sure you want to delete this user?")) {
          $scope.users.splice(index, 1);
        }
      };
    });
  </script>

</body>
</html>
```

OUTPUT

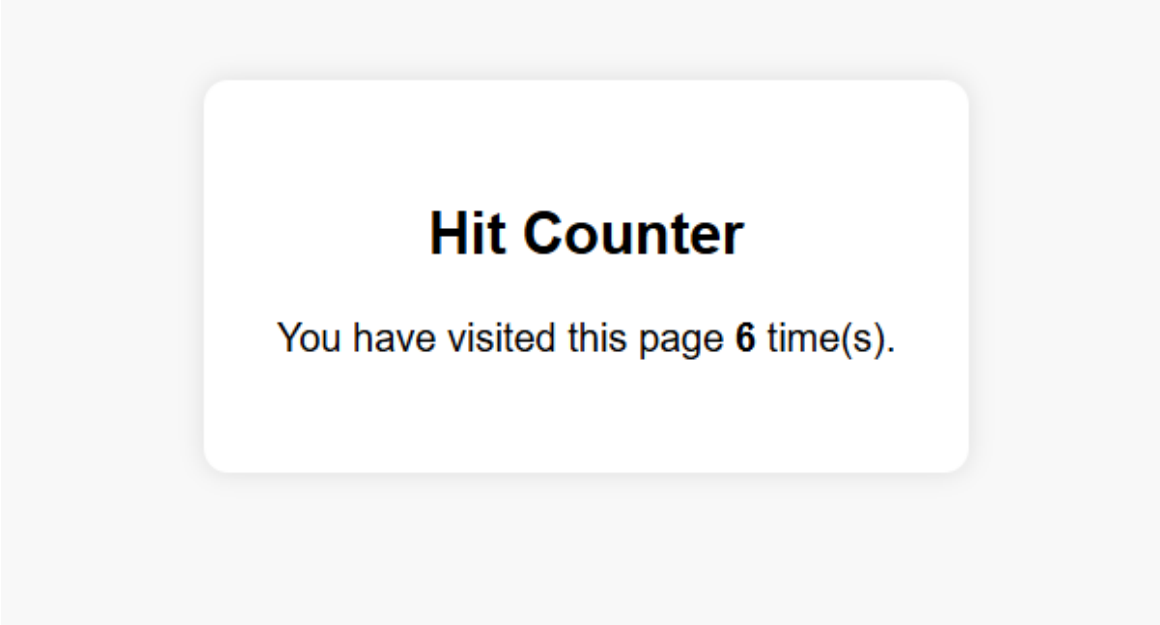

User Management CRUD App (AngularJS)
Add New User
Name
Email
Add User
#
Name
Email
Actions
1
bharath
bharath@gmail.com
Edit
Delete

9a. Write a PHP program to hit counter using cookies

```
<?php
// Set cookie name
$cookie_name = "visit_count";

// Check if cookie exists
if(isset($_COOKIE[$cookie_name])) {
    $visit_count = $_COOKIE[$cookie_name] + 1;
} else {
    $visit_count = 1; // First visit
}

// Set/update the cookie (expires in 1 year)
setcookie($cookie_name, $visit_count, time() + (365 * 24 * 60 * 60)); // 1 year

?>

<!DOCTYPE html>
<html>
<head>
    <title>PHP Hit Counter Using Cookies</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background: #f8f8f8;
```

```
            padding: 30px;
            text-align: center;
        }
        .counter-box {
            background: white;
            display: inline-block;
            padding: 30px;
            border-radius: 10px;
            box-shadow: 0 0 10px rgba(0,0,0,0.1);
        }
    </style>
</head>
<body>

<div class="counter-box">
    <h2>Hit Counter</h2>
    <p>
        <?php
        echo "You have visited this page <strong>$visit_count</strong> time(s).";
        ?>
    </p>
</div>

</body>
</html>
```

OUTPUT

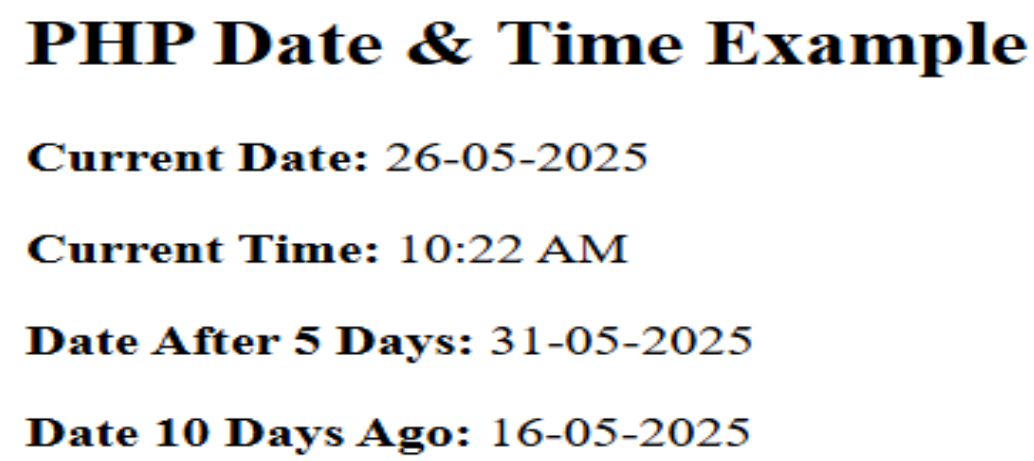


9b. Write a PHP program to calculate Date and Time function.

```
<?php
// Set your timezone
date_default_timezone_set("Asia/Kolkata");

// Current date and time
$currentDate = date("d-m-Y");
$currentTime = date("h:i A");

// Date after 5 days
$dateAfter5Days = date("d-m-Y", strtotime("+5 days"));

// Date before 10 days
$dateBefore10Days = date("d-m-Y", strtotime("-10 days"));
?>

<!DOCTYPE html>
<html>
<head>
    <title>Simple Date & Time in PHP</title>
</head>
<body>
```

```
<h2>PHP Date & Time Example</h2>

<p><strong>Current Date:</strong> <?php echo $currentDate; ?></p>
<p><strong>Current Time:</strong> <?php echo $currentTime; ?></p>
<p><strong>Date After 5 Days:</strong> <?php echo $dateAfter5Days; ?></p>
<p><strong>Date 10 Days Ago:</strong> <?php echo $dateBefore10Days;
?></p>

</body>
</html>
```

OUTPUT



10.Create a web page for software company website

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>TechNova Software Solutions</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <!-- Bootstrap CSS CDN -->
  <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet">
</head>
<body>

<!-- Navbar -->
<nav class="navbar navbar-expand-lg navbar-dark bg-primary">
  <div class="container">
    <a class="navbar-brand" href="#">TechNova</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
data-bs-target="#navbarNav">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNav">
      <ul class="navbar-nav ms-auto">
        <li class="nav-item"><a class="nav-link"
href="#services">Services</a></li>
        <li class="nav-item"><a class="nav-link" href="#about">About</a></li>
        <li class="nav-item"><a class="nav-link" href="#team">Team</a></li>
        <li class="nav-item"><a class="nav-link"
href="#contact">Contact</a></li>
      </ul>
    </div>
  </div>
</nav>

<!-- Hero Section -->
<section class="bg-light text-center py-5">
  <div class="container">
    <h1 class="display-4 fw-bold">Innovative Software for a Smarter
Future</h1>
    <p class="lead">We build reliable and scalable digital solutions to empower
businesses.</p>
```

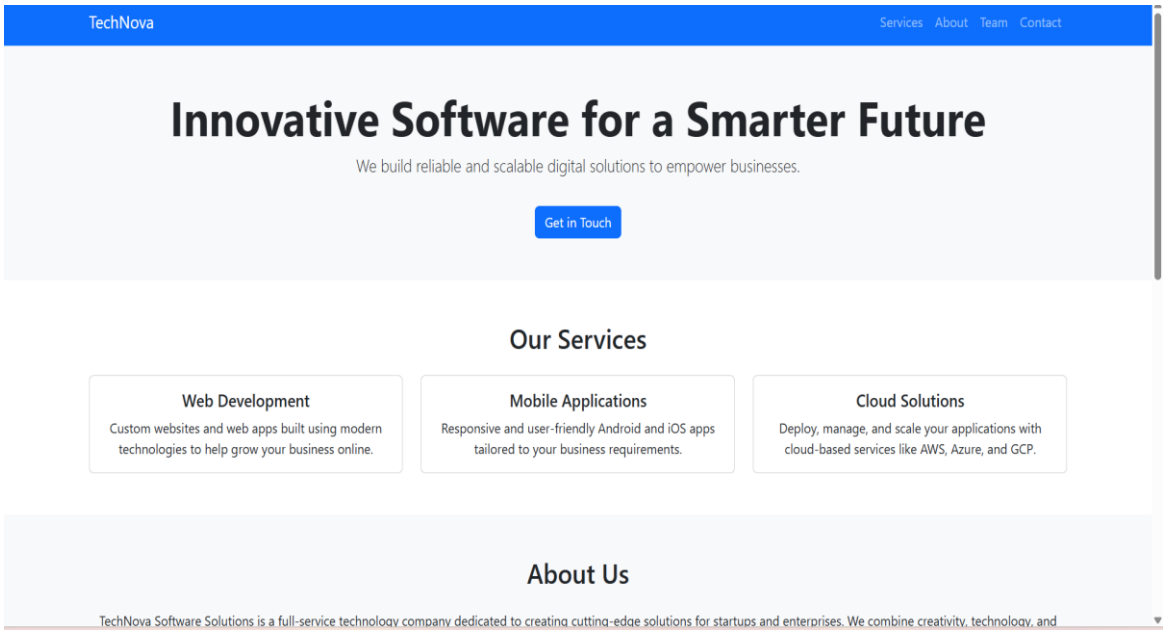
```
    <a href="#contact" class="btn btn-primary mt-3">Get in Touch</a>
  </div>
</section>

<!-- Services Section -->
<section id="services" class="py-5">
  <div class="container text-center">
    <h2 class="mb-4">Our Services</h2>
    <div class="row g-4">
      <div class="col-md-4">
        <div class="card h-100">
          <div class="card-body">
            <h5 class="card-title">Web Development</h5>
            <p class="card-text">Custom websites and web apps built using modern
technologies to help grow your business online.</p>
          </div>
        </div>
      </div>
      <div class="col-md-4">
        <div class="card h-100">
          <div class="card-body">
            <h5 class="card-title">Mobile Applications</h5>
            <p class="card-text">Responsive and user-friendly Android and iOS
apps tailored to your business requirements.</p>
          </div>
        </div>
      </div>
      <div class="col-md-4">
        <div class="card h-100">
          <div class="card-body">
            <h5 class="card-title">Cloud Solutions</h5>
            <p class="card-text">Deploy, manage, and scale your applications with
cloud-based services like AWS, Azure, and GCP.</p>
          </div>
        </div>
      </div>
    </div>
  </div>
</section>

<!-- About Section -->
<section id="about" class="bg-light py-5">
  <div class="container">
```

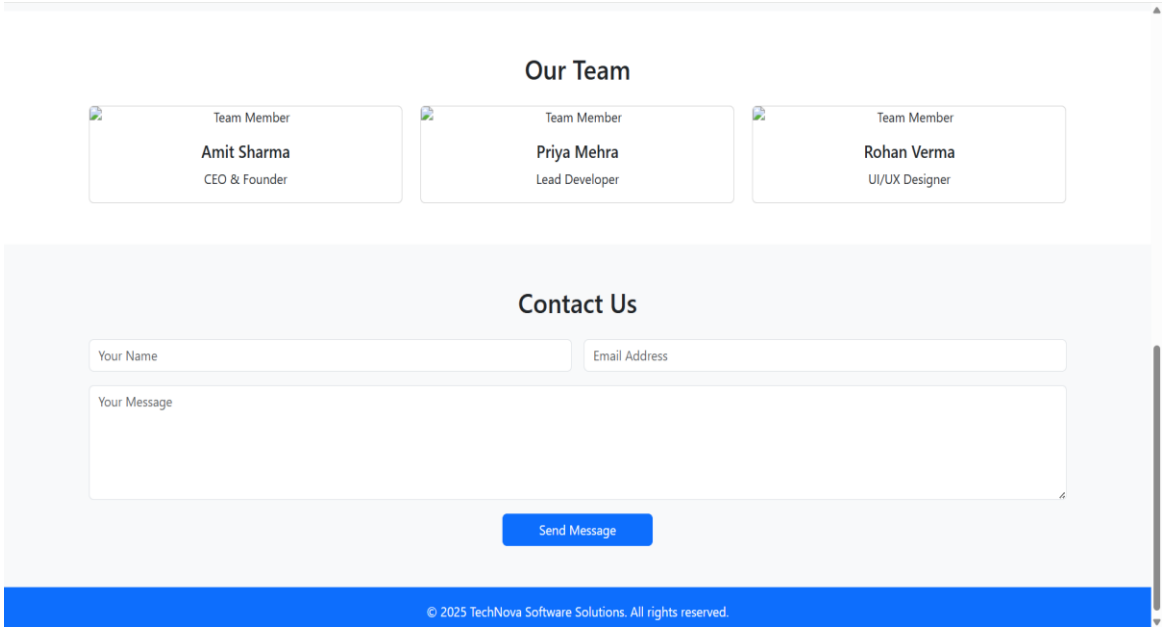
```
    <h2 class="text-center mb-4">About Us</h2>
    <p class="text-center">TechNova Software Solutions is a full-service
technology company dedicated to creating cutting-edge solutions for startups
and enterprises. We combine creativity, technology, and business expertise to
deliver powerful applications.</p>
    <ul>
      <li>Founded in 2015 with a mission to innovate</li>
      <li>Clients in over 10 countries</li>
      <li>Dedicated team of 50+ developers and designers</li>
    </ul>
  </div>
</section>

<!-- Team Section -->
<section id="team" class="py-5">
  <div class="container text-center">
    <h2 class="mb-4">Our Team</h2>
    <div class="row g-4">
      <div class="col-md-4">
        <div class="card">
          <img src="https://via.placeholder.com/300x200" class="card-img-top"
alt="Team Member">
          <div class="card-body">
            <h5 class="card-title">Amit Sharma</h5>
            <p class="card-text">CEO & Founder</p>
          </div>
        </div>
      </div>
      <div class="col-md-4">
        <div class="card">
          <img src="https://via.placeholder.com/300x200" class="card-img-top"
alt="Team Member">
          <div class="card-body">
            <h5 class="card-title">Priya Mehra</h5>
            <p class="card-text">Lead Developer</p>
          </div>
        </div>
      </div>
      <div class="col-md-4">
        <div class="card">
          <img src="https://via.placeholder.com/300x200" class="card-img-top"
alt="Team Member">
          <div class="card-body">
```

```
          <h5 class="card-title">Rohan Verma</h5>
          <p class="card-text">UI/UX Designer</p>
        </div>
      </div>
    </div>
  </div>
</section>

<!-- Contact Section -->
<section id="contact" class="bg-light py-5">
  <div class="container">
    <h2 class="text-center mb-4">Contact Us</h2>
    <form class="row g-3">
      <div class="col-md-6">
        <input type="text" class="form-control" placeholder="Your Name"
required>
      </div>
      <div class="col-md-6">
        <input type="email" class="form-control" placeholder="Email Address"
required>
      </div>
      <div class="col-12">
        <textarea class="form-control" rows="5" placeholder="Your
Message"></textarea>
      </div>
      <div class="col-12 text-center">
        <button type="submit" class="btn btn-primary px-5">Send
Message</button>
      </div>
    </form>
  </div>
</section>

<!-- Footer -->
<footer class="bg-primary text-white text-center py-3">
  <p class="mb-0">&amp;copy; 2025 TechNova Software Solutions. All rights
reserved.</p>
</footer>

<!-- Bootstrap JS CDN -->
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.j
```

s"></script>
</body>
</html>

OUTPUT





VIVA QUESTIONS

1. **What is the purpose of the <!DOCTYPE html> declaration?**
   The <!DOCTYPE html> declaration defines the document type and version of HTML to the browser. In HTML5, it ensures the browser renders the page in standards mode (not quirks mode), making it interpret the page according to the HTML5 specification, which improves compatibility and consistency across browsers.
2. **What are semantic HTML elements? Give examples.**
   Semantic HTML elements clearly describe their meaning both to the browser and developers. Examples include <header>, <footer>, <nav>, <article>, <section>, and <aside>. These elements improve accessibility, SEO, and code readability by defining the structure and content roles clearly.
3. **What is the difference between and ?**
    is a block-level element that takes up the full width available and starts on a new line. It's used for grouping larger blocks of content. is an inline element that only takes as much width as needed and does not cause line breaks, used for styling parts of text inside other elements.
4. **How do you create a hyperlink in HTML?**
   A hyperlink is created using the <a> (anchor) tag with the href attribute specifying the URL. Example: Visit Example. This makes the text clickable, linking users to the specified resource.
5. **What are the different types of lists in HTML?**
   There are three main types:
   - Ordered list (<ol>): numbered list.
   - Unordered list (<ul>): bulleted list.
   - Description list (<dl>): list of terms and descriptions, with <dt> for terms and <dd> for definitions.
6. **Explain the purpose of the <form> element and its attributes.**
   The <form> element collects user inputs to send to a server. Attributes include:
   - action: URL where form data is sent.
   - method: HTTP method to send data (GET or POST).
   - enctype: encoding type for the data (e.g., multipart/form-data for file uploads). Forms are used for login, search, data submission,

etc.

7. **What is the difference between id and class attributes in HTML?**
   id uniquely identifies a single element in the DOM and should not be repeated. It is often used for JavaScript or CSS targeting unique elements. class can be shared by multiple elements to group them for styling or scripting purposes.
8. **How do you include multimedia (audio/video) in a webpage?**
   Use the <audio> and <video> elements with the src attribute pointing to the media file. You can include controls by adding the controls attribute so users can play/pause, adjust volume, etc. Example:

html
CopyEdit
<video controls>
  <source src="movie.mp4" type="video/mp4">
  Your browser does not support the video tag.
</video>

CSS

9. **What are the different ways to apply CSS to a webpage?**
   - **Inline CSS:** Using the style attribute inside an HTML element, e.g., <p style="color:red;">.
   - **Internal CSS:** Inside a <style> tag in the <head> section of HTML.
   - **External CSS:** Linking an external .css file via <link> tag. External CSS is preferred for maintainability and reusability.
10. **What is the Box Model in CSS?**
    The CSS Box Model represents the rectangular boxes generated for elements. It consists of:
    - **Content:** The actual content like text/images.
    - **Padding:** Space between content and border.
    - **Border:** Surrounds the padding and content.
    - **Margin:** Space outside the border separating elements. Understanding it is crucial for spacing and layout control.
11. **Explain the difference between relative, absolute, and fixed positioning.**
    - **Relative:** The element is positioned relative to its normal position, allowing it to move without affecting other elements.

- **Absolute:** Positioned relative to the nearest positioned ancestor (non-static). If none, relative to the document body; removed from normal flow.
- **Fixed:** Positioned relative to the viewport and stays fixed on screen even when scrolling. Useful for sticky headers or menus.

12. **How does the CSS specificity hierarchy work?**
    Specificity determines which CSS rule applies if multiple rules target the same element. Hierarchy:
    Inline styles (style="") have highest specificity > IDs (#id) > Classes, attributes, pseudo-classes (.class, [attr], :hover) > Elements and pseudo-elements (div, p, ::before). If specificity ties, the last declared rule applies.
13. **What are pseudo-classes and pseudo-elements? Give examples.**
    - **Pseudo-classes** select elements in a specific state, e.g., :hover (mouse over), :focus (input focus).
    - **Pseudo-elements** select parts of elements, e.g., ::before adds content before an element, ::after after the element.
14. **How can you center a div horizontally and vertically using CSS?**
    Using Flexbox:

```
css
CopyEdit
container {
  display: flex;
  justify-content: center; /* horizontal center */
  align-items: center;     /* vertical center */
  height: 100vh;           /* full viewport height */
}
```

15. **What is the difference between em, rem, and px units?**
    - px: Absolute pixels, fixed size.
    - em: Relative to the font size of the parent element. Multiples multiply.
    - rem: Relative to the root (<html>) font size. More predictable than em when nested.
16. **Explain the use of Flexbox and how it differs from CSS Grid.**
    Flexbox is a one-dimensional layout system for aligning items in a row or column. It excels at distributing space and aligning items in a linear direction. CSS Grid is a two-dimensional system allowing control over

rows *and* columns, ideal for complex layouts.

JavaScript

17. **What are the different data types in JavaScript?**
    - **Primitive types:** string, number, boolean, null, undefined, symbol, bigint.
    - **Objects:** Collections of key-value pairs including arrays, functions, dates, etc.
18. **Explain the difference between var, let, and const.**
    - var is function-scoped and can be redeclared and updated.
    - let is block-scoped and can be updated but not redeclared within the same scope.
    - const is block-scoped and cannot be updated or redeclared; used for constants.
19. **What is a closure in JavaScript?**
    A closure is a function that has access to variables from its outer (enclosing) function even after the outer function has returned. It "remembers" the environment in which it was created.
20. **How does event bubbling and capturing work in JavaScript?**
    When an event occurs on an element, it first travels down from the root to the target (capturing), then bubbles up from the target back to the root (bubbling). By default, event handlers listen during the bubbling phase.
21. **What is the difference between == and ===?**
    - == compares values after type coercion (e.g., '5' == 5 is true).
    - === compares both value and type strictly (e.g., '5' === 5 is false).
22. **What are callback functions and promises?**
    - A **callback** is a function passed to another function to be executed later, often after an asynchronous operation.
    - A **promise** represents a value that may be available now or in the future, providing a more structured way to handle async operations and avoid "callback hell."
23. **How can you create an object in JavaScript?**
    - Using object literal: const obj = { key: 'value' };
    - Using constructor functions or classes with new.
    - Using Object.create() for prototype-based creation.
24. **What is the purpose of the this keyword in JavaScript?**
    this refers to the object context from which a function is called. Its value

depends on how the function is invoked: in global scope it refers to the global object; inside a method, the object owning the method; inside event handlers, the element triggering the event.

Bootstrap

25. **What is Bootstrap and why is it used?**
    Bootstrap is a popular open-source CSS framework that provides pre-built responsive styles, components, and JavaScript plugins to quickly develop modern websites that work on all devices and screen sizes.
26. **What is the grid system in Bootstrap?**
    A flexible 12-column system allowing developers to create responsive page layouts. You define rows and columns which adapt based on screen size using classes like .col-md-6, .col-lg-4.
27. **How do you create a responsive navbar in Bootstrap?**
    By using classes such as .navbar, .navbar-expand-lg to control collapse behavior, and .navbar-toggler button to toggle visibility on smaller screens.
28. **What are Bootstrap utilities and how do they help in styling?**
    Utilities are small helper classes like .m-3 (margin), .text-center (text alignment), .bg-primary (background color), which help quickly style elements without writing custom CSS.
29. **How do you include Bootstrap in your project?**
    By linking Bootstrap's CSS and JavaScript files via CDN links in <head> and before closing </body> tag, or by downloading and hosting the files locally.
30. **What are Bootstrap components? Name some.**
    Reusable UI elements like Buttons, Cards, Modals, Alerts, Forms, Navbars, Carousels, and Dropdowns that come pre-styled and functional.

PHP

31. **What are the main data types in PHP?**
    PHP supports scalar types: string, integer, float, boolean; compound types: array, object; special types: NULL, and resource types (external connections).
32. **How does PHP handle form data?**
    Using superglobals $_GET (data sent in URL query string) and $_POST (data sent via HTTP POST). Data from form fields is accessible as key-

value pairs, e.g., $_POST['username'].

33. **What is the difference between include and require in PHP?**
    Both include code from external files. include issues a warning if the file is missing and continues execution. require produces a fatal error and stops script execution if the file is not found.
34. **How do sessions work in PHP?**
    Sessions allow storing user-specific data on the server with a unique session ID stored in a cookie on the client. Data persists across page requests, enabling login states, carts, etc.
35. **Explain the difference between GET and POST methods.**
    - GET appends data in the URL, visible to users, limited length, used for retrieving data.
    - POST sends data in the request body, not visible in URL, used for sending sensitive or large data.
36. **What are associative arrays in PHP?**
    Arrays with string keys instead of numeric indexes, allowing key-value pairs where keys are custom identifiers.
37. **How do you connect to a MySQL database using PHP?**
    Using mysqli_connect(host, username, password, database) or PDO class. You must handle connection success/failure and query execution.
38. **What is the purpose of the $_SERVER superglobal?**
    It holds information about headers, paths, and script locations. For example, $_SERVER['REQUEST_METHOD'] tells if the request is GET or POST.

---

jQuery

39. **What is jQuery and why is it used?**
    A lightweight JavaScript library that simplifies HTML document traversal, event handling, animations, and Ajax interactions, allowing for concise and cross-browser compatible code.
40. **How do you select elements using jQuery?**
    Using the $() function with CSS selectors, e.g., $("#id") selects element with that ID, $(".class") selects all with that class, or $("div") for all divs.
41. **Explain event handling in jQuery.**
    jQuery provides methods like .on(), .click(), .hover() to attach event handlers to elements easily, supporting event delegation and cross-browser compatibility.

42. **What are some common jQuery effects/animations?**
    Methods such as .hide(), .show(), .fadeIn(), .fadeOut(), .slideUp(), .slideDown() allow simple visual effects and transitions.
43. **How do you make an AJAX request with jQuery?**
    Using $.ajax(), $.get(), or $.post() functions, allowing asynchronous HTTP requests without reloading the page.

AngularJS

44. **What is AngularJS?**
    A JavaScript-based open-source framework developed by Google, used to build dynamic single-page applications with features like two-way data binding, dependency injection, and modular components.
45. **Explain the concept of two-way data binding in AngularJS.**
    Changes to the model (JavaScript variables) automatically update the view (HTML), and user inputs on the view update the model, keeping both synchronized without manual DOM manipulation.
46. **What are directives in AngularJS? Name a few.**
    Special markers in HTML (ng- attributes) that tell Angular to attach behavior or transform DOM. Examples:
    - ng-model: binds input/select to model
    - ng-repeat: loops over collections
    - ng-if: conditionally renders elements
47. **How do you create and use controllers in AngularJS?**
    Controllers are defined with .controller() function and manage $scope data and logic, which the view can bind to.
48. **What is the purpose of $scope in AngularJS?**
    An object that connects the controller and the view. Variables and functions attached to $scope are accessible in the HTML for data binding.
49. **How do you make HTTP requests in AngularJS?**
    Using the $http service, which returns promises for asynchronous requests to REST APIs.
50. **What are filters in AngularJS?**
    Filters format the data displayed to the user, such as formatting dates (date), converting text to uppercase (uppercase), currency formatting (currency), and more.