Graphic Era
HILL UNIVERSITY
University under section 2(f) of UGC Act, 1956
Bhimtal Campus

Term work

of

Full Stack Web Development Lab (PCS – 693)

Submitted in partial fulfillment of the requirement for the VI semester

Bachelor of Technology

By

Shailesh Jukaria

2261519

Under the Guidance of

Ms. Vaishali Dev

Faculty In-Charge

Department of CSE

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

GRAPHIC ERA HILL UNIVERSITY, BHIMTAL CAMPUS

SATTAL ROAD, P.O. BHOWALI DISTRICT-

NAINITAL-263132

2024-2025

Graphic Era
HILL UNIVERSITY
University under section 2(f) of UGC Act, 1956
Bhimtal Campus

CERTIFICATE

The term work of Full Stack Web Development Lab, being submitted by Shailesh Jukaria S/O Mr. Navin Chandra Jukaria , University Roll Number 2261519 to Graphic Era Hill University, Bhimtal Campus for the award of bona fide work carried out by him. He has worked under my guidance and supervision and fulfilled the requirement for the submission of this work report.

(Ms. Vaishali Dev) **(Dr. Ankur Singh Bisht)**

Faculty In-Charge **HOD, CSE Dept.**

Graphic Era
HILL UNIVERSITY
University under section 2(f) of UGC Act, 1956
Bhimtal Campus

ACKNOWLEDGEMENT

I take immense pleasure in thanking Honorable **Ms. Vaishali Dev** (Faculty In- Charge, Dept. of CSE, GEHU Bhimtal Campus) for allowing me to carry out this practical work under her excellent and optimistic supervision. This has all been possible due to her novel inspiration, able guidance and useful suggestions that have helped me in developing my subject concepts as a student.

I want to extend thanks to our President **Prof. (Dr.) Kamal Ghanshala** for providing us all infrastructure and facilities to work in need without which this work would not be possible.

(Shailesh Jukaria)

University Roll Number: 2261519

INDEX

S.NO.	OBJECTIVE	DATE	SIGN.
1.	Implement <iframe> tag by embedding elements from another webpages.		
2.	Implement the CSS3 and add different functionalities on the webpage.		
3.	Implement JavaScript for creating Student Registration Form, To-Do List, Digital Clock, Color Changer, Image Slider		
4.	Create an XML file to store information about students, books, or employees.		
5.	Create a simple CRUD (Create, Read, Update, Delete) web application using PHP and MySQL.		
6.	Build a simple calculator using React that can perform basic arithmetic operations: addition, subtraction, multiplication, and division.		
7.	Build a multi-page React application using React Router to demonstrate routing between different components.		
8.	Create a React component that fetches a list of users from the API https://jsonplaceholder.typicode.com/users using axios and displays their names and email addresses in a list.		

Q1. Use <iframe> tag:-

- Implement <iframe> tag by embedding the following elements from another webpage: <header>, <nav>, <section>, <article>, <footer>. Add a navigation bar inside <nav> with links to different sections of the page using the <a> tag.
- Embed a table from another webpage using <iframe>. The structure of the table should be as:-

Day	Seminar		
	Schedule		Topic
	Begin	End	
Monday	8:00 a.m.	5:00 p.m.	Introduction to XML
			Validity: DTD and Relax NG
Tuesday	8:00 a.m.	11:00 a.m.	XPath
	11:00 a.m.	2:00 p.m.	
	2:00 p.m.	5:00 p.m.	XSL Transformations
Wednesday	8:00 a.m.	12:00 p.m.	XSL Formatting Objects

- Embed <map> and <area> from another web page using <iframe>. Create a clickable map of a school or college campus.
  - Use an <img> tag to display a campus map.
  - Use a <map> with multiple <area> tags to create clickable zones for different buildings (e.g., Library, Cafeteria, Sports Complex).
  - Each <area> should link to a new page with information about that building. Use alt attributes to describe each area.

CODE:

// Main.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Iframe1 Example</title>
</head>
<body>
<h2>Embedding Webpage Sections with &lt;iframe&gt;</h2>

<iframe src="example.html" width="500px" height="700px"></iframe>
<iframe src="tablepage.html" width="300px" height="700px"></iframe>
<iframe src="clickablemap.html" width="600px" height="700px" style="border: 2px solid
black;"></iframe>

</body>
</html>
```

// Example.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Document</title>
<style>
body {
font-family: Arial, sans-serif;
margin: 0;
padding: 0;
}
header {
background-color: #333; color:
white;
text-align: center;
padding: 15px;
font-size: 24px;
}
```

```
nav {
background-color: #444;
padding: 10px;
text-align: center;
}
nav a { color:
white;
text-decoration: none;
margin: 0 15px;
font-size: 18px;
}
nav a:hover {
text-decoration: underline;
}
section {
padding: 20px;
}
article {
background-color: #f4f4f4;
padding: 15px;
margin: 10px 0;
border-left: 5px solid #333;
}
footer {
background-color: #222; color:
white;
text-align: center;
padding: 10px; position:
relative; bottom: 0;
width: 100%;
}
</style>
</head>
<body>
<header>
Welcome to My Embedded Page
</header>
```

```
<nav>
<a href="#home">Home</a>
<a href="#about">About</a>
<a href="#services">Services</a>
<a href="#contact">Contact</a>
</nav>

<section>
<h2>About This Page</h2>
<p>This page is embedded using an &lt;iframe&gt;. It contains multiple sections like a header, navigation bar, content area, and a footer.</p>
<article>
<h3>Article 1</h3>
<p>This is the first article on the page.</p>
</article>
</section>
<section id="home">
<h2>Home</h2>
<p>Welcome to the homepage. Explore our website and learn more.</p>
</section>
<section id="about">
<h2>About</h2>
<p>We are a company dedicated to providing excellent services.</p>
</section>
<section id="services">
<h2>Services</h2>
<p>We offer various services, including web development and design.</p>
</section>
<section id="contact">
<h2>Contact</h2>
<p>Get in touch with us via email at contact@example.com.</p>
</section>
<footer>
&copy; 2025 Example Page | All Rights Reserved
</footer>
</body>
</html>
```

//tablepage.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Document</title>
</head>
<body>
<table border="1">
<tr>
<th rowspan="3">Day</th>
<th colspan="4">Seminar</th>
</tr>
<tr>
<th colspan="2">Schedule</th>
<th colspan="2" rowspan="2">Topic</th>
</tr>
<tr>
<th>Begin</th>
<th>End</th>
</tr>
<tr>
<td rowspan="2">Monday</td>
<td rowspan="2">8:00 a.m.</td>
<td rowspan="2">5.00 p.m.</td>
<td colspan="2">Introduction to XML</td>
</tr>
<tr>
<td colspan="2">Validity:DTD and Relax NG</td>
</tr>
<td rowspan="3">Tuesday</td>
<td>8.00 a.m.</td>
<td>11.00 a.m.</td>
<td colspan="2">XPath</td>
</tr>
<tr>
<td>11.00 a.m.</td>
<td>2.00 p.m.</td>
```

```
<td rowspan="2" colspan="2">XSL Transformations</td>
</tr>
<tr>
<td>2.00 p.m.</td>
<td>5.00 p.m.</td>
</tr>
<tr>
<td>Wednesday</td>
<td>8.00 a.m.</td>
<td>12.00 p.m.</td>
<td colspan="2">XSL Formatting Objects</td>
</tr>
</table>
</body>
</html>
```

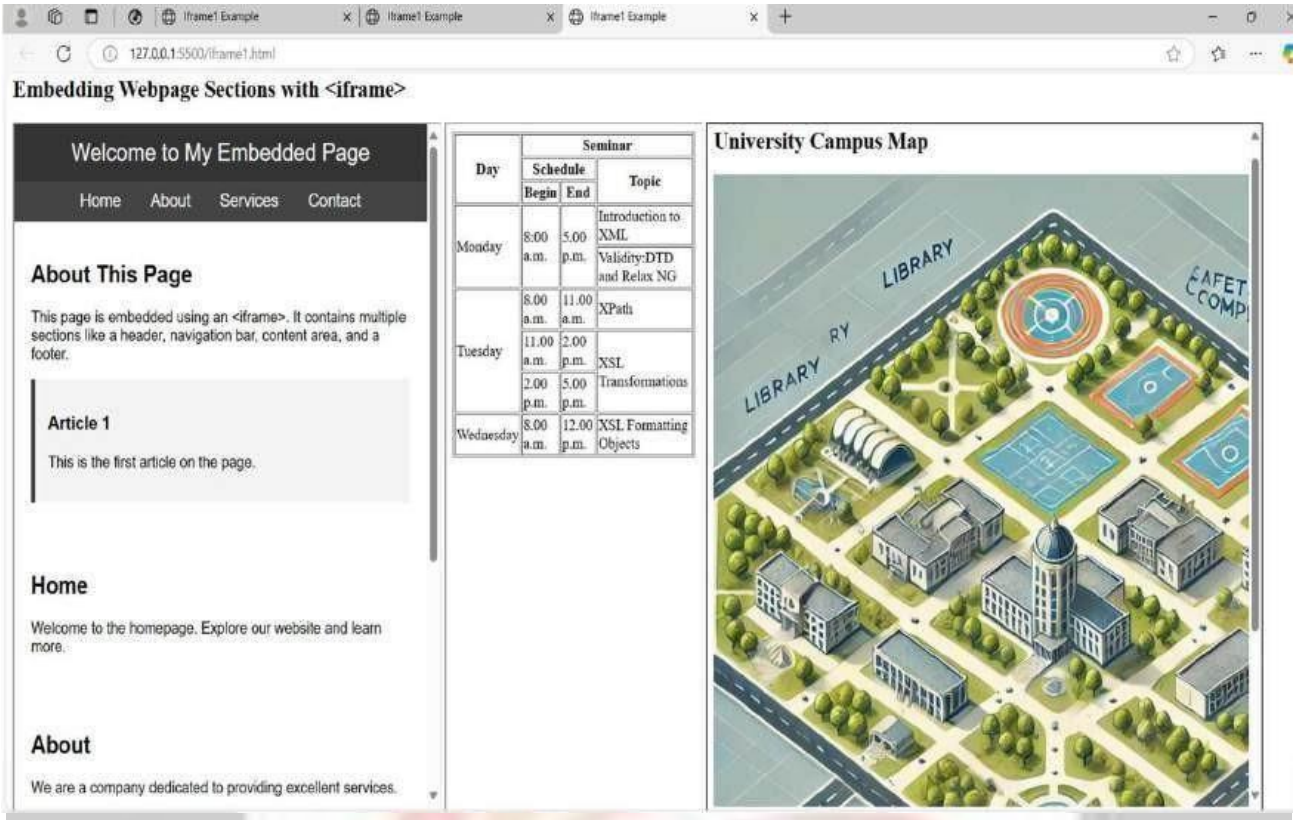
<! -- ClickableMap.html -->

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Clickable Campus Map</title>
</head>
<body>
<h2>University Campus Map</h2>
<img src="campus.jpg" usemap="#campusmap" width="800px" height="800px" alt="University
Campus Map">
<map name="campusmap">
<area shape="rect" coords="50,100,200,200" href="library.html" alt="Library">
<area shape="circle" coords="400,300,60" href="cafeteria.html" alt="Cafeteria">
<area shape="rect" coords="600,150,750,300" href="sports.html" alt="Sports Complex">
<area shape="poly" coords="300,500,350,550,400,500,350,450" href="admin.html" alt="Administration">
<area shape="rect" coords="500,600,650,750" href="hostel.html" alt="Hostel">
</map>
<p>Click on a building to view more details.</p>
</body>
</html>
```

```
<! -- library.html -->
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Library</title>
</head>
<body>
<h2>Library</h2>
<p>The library contains a vast collection of books, research papers, and digital resources for students.</p>
<a href="mapembed.html">Go Back to Campus Map</a>
</body>
</html>
<! -- cafeteria.html -->
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Cafeteria</title>
</head>
<body>
<h2>Cafeteria</h2>
<p>The cafeteria serves a variety of meals, snacks, and beverages throughout the day.</p>
<a href="mapembed.html">Go Back to Campus Map</a>
</body>
</html>
<! -- sports.html -->
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Sports Complex</title>
</head>
<body>
<h2>Sports Complex</h2>
```

```
<p>The sports complex includes a gym, basketball courts, and a football field.</p>
<a href="mapembed.html">Go Back to Campus Map</a>
</body>
</html>
<! -- hostel.html -->
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Hostel</title>
</head>
<body>
<h2>Hostel</h2>
<p>The hostel provides comfortable accommodation for students, with modern facilities and security.</p>
<a href="mapembed.html">Go Back to Campus Map</a>
</body>
</html>
<! -- admin.html -->
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Administration</title>
</head>
<body>
<h2>Administration Office</h2>
<p>The administration office handles admissions, student records, and other official tasks.</p>
<a href="mapembed.html">Go Back to Campus Map</a>
</body>
</html>
```

OUTPUT:



Q2. Implement the following using CSS3:

- Style a webpage by changing the font, color, and size of headings and paragraphs.
- Create a button with background color, padding, and rounded corners. Add a hover effect that changes the background color.
- Display an image and add a border-radius for rounded corners and add a shadow effect to the image.
- Create an animated gradient border around a div using @keyframes and border-image-source.
- Apply an animated gradient border that changes colors. Use border-image-source instead of solid borders.
- Create a parallax scrolling effect where the background image moves at a different speed than the content.
- Use background-attachment: fixed; for the parallax effect. Ensure the content scrolls over the background smoothly. Use Flexbox to center a box in the middle of the page:- Create a div box and center it both vertically and horizontally. Add padding, background color, and a shadow to the box.

CODE:

//Main.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Complete Styled Page</title>
<link rel="stylesheet" href="styles2.css">
</head>
<body>


<h1>Welcome to My Styled Webpage</h1>
<h2>About Us</h2>
<p>This is a sample paragraph to demonstrate font, color, and size styling using CSS.</p>

<button class="custom-button">Click Me</button>

<img src="image1.jpg" alt="Beautiful Scenery" class="styled-image">

<div class="animated-border">
<h2>Gradient Border Animation</h2>
<p>This div has an animated gradient border!</p>
</div>

<div class="parallax"></div>
<div class="content">
<h2>Parallax Scrolling Effect</h2>
<p>This is a simple example of a parallax scrolling effect. The background moves at a different
speed than the text.</p>
<p>You can add more content here and observe how it smoothly scrolls over the fixed background.</p>
</div>
<div class="parallax"></div>

<div class="center-container">
<div class="center-box">
<h2>Centered Box</h2>
<p>This box is perfectly centered using Flexbox.</p>
</div>
</div>

</body>
</html>
```

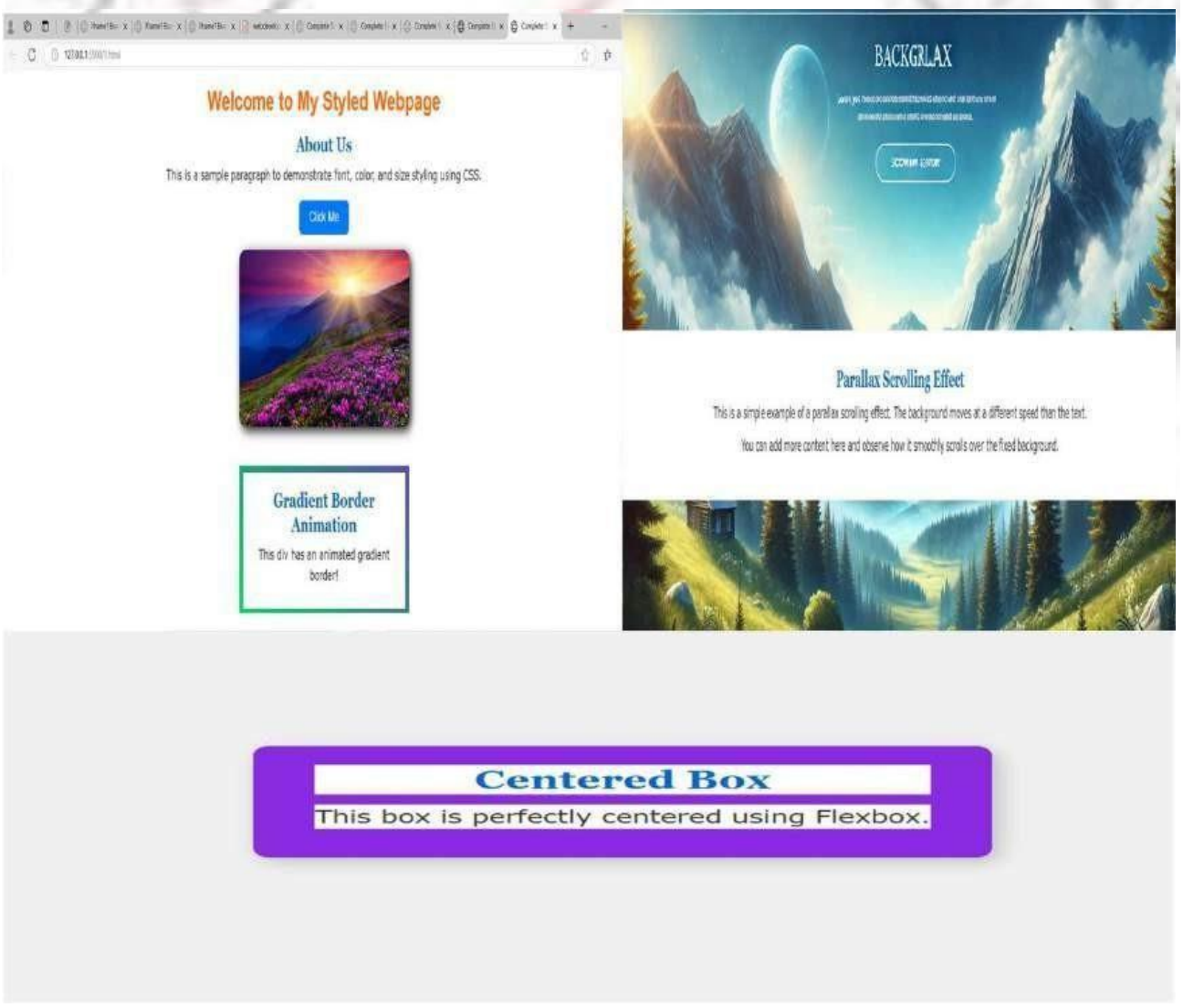
//style.css

```
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
    background-color: white;
}
h1 {
    font-family: Arial, sans-serif;
    font-size: 36px;
    color: #ff6600;
    text-align: center;
    margin: 20px 0;
}
h2 {
    font-family: 'Georgia', serif;
    font-size: 28px;
    color: #0066cc;
    text-align: center;
}
p {
    font-family: 'Verdana', sans-serif;
    font-size: 18px;
    color: #333333;
    line-height: 1.5;
    text-align: center;
    margin: 10px 0;
}
.custom-button {
    display: block;
    margin: 20px auto;
    background-color: #007BFF;
    color: white;
    font-size: 18px;
    padding: 12px 24px;
    border: none; border-
    radius: 8px; cursor:
    pointer;
    transition: background-color 0.3s ease-in-out;
}
.custom-button:hover { background-
    color: yellowgreen;
}
```

```
.styled-image {
    display: block;
    width: 400px;
    height: auto;
    border-radius: 15px;
    box-shadow: 5px 5px 15px rgba(0, 0, 0, 0.9);
    margin: 20px auto;
}

@keyframes gradientAnimation {
    0% { border-image-source: linear-gradient(45deg, #ff0000, #ff7300); } 25% {
    border-image-source: linear-gradient(45deg, #ff7300, #ffeb00); } 50% { border-
    image-source: linear-gradient(45deg, #ffeb00, #00ff00); } 75% { border-image-
    source: linear-gradient(45deg, #00ff00, #007bff); } 100% { border-image-source:
    linear-gradient(45deg, #007bff, #ff0000); }
}
.animated-border {
    width: 400px;
    padding: 20px;
    text-align: center;
    font-family: Arial, sans-serif;
    font-size: 18px;
    background-color: white;
    border: 10px solid; border-
    image-slice: 1;
    animation: gradientAnimation 5s linear infinite;
    margin: 50px auto;
    border-radius: 15px;
}

.parallax {
    background-image: url('background.jpg'); height:
    400px;
    background-size: cover;
    background-position: center;
    background-attachment: fixed;
}
.content {
    padding: 40px;
    font-family: Arial, sans-serif;
    font-size: 18px;
    color: #333;
    text-align: center;
    background-color: white;
}
```

```
.center-container {
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
    background-color: #f0f0f0;
}
.center-box {
    background-color: blueviolet;
    padding: 20px 40px;
    border-radius: 10px;
    box-shadow: 5px 5px 15px rgba(0, 0, 0, 0.2);
    text-align: center;
    font-family: Arial, sans-serif;
}
```

OUTPUT:



Q3. Implement JavaScript for following:-

- Create a **Student Registration Form** using **HTML** that includes the following fields:

Form Requirements:

- **Student Name** – (Text input, Required)
- **Email** – (Email input, Required, Must be in a valid email format)
- **Phone Number** – (Number input, Required, Must be exactly 10 digits)
- **Date of Birth** – (Date input, Required)
- **Gender** – (Radio buttons: Male, Female, Other, Required)
- **Course Selection** – (Dropdown with at least 3 options, Required)
- **Address** – (Textarea, Required)
- **Password** – (Password input, Required, Minimum 6 characters)
- **Confirm Password** – (Password input, Must match Password)
- **Submit Button** – (Button to submit the form)

Validate a user registration form (e.g., check email format, password strength).

- **To-Do List** – Create a simple to-do list where users can add and remove tasks.
- **Digital Clock** – Display the current time that updates every second.
- **Color Changer** – Allow users to change the background color by clicking buttons.
- **Image Slider** – Create a basic image carousel that auto-plays or changes images on button click

Create a Student Registration Form.

CODE:

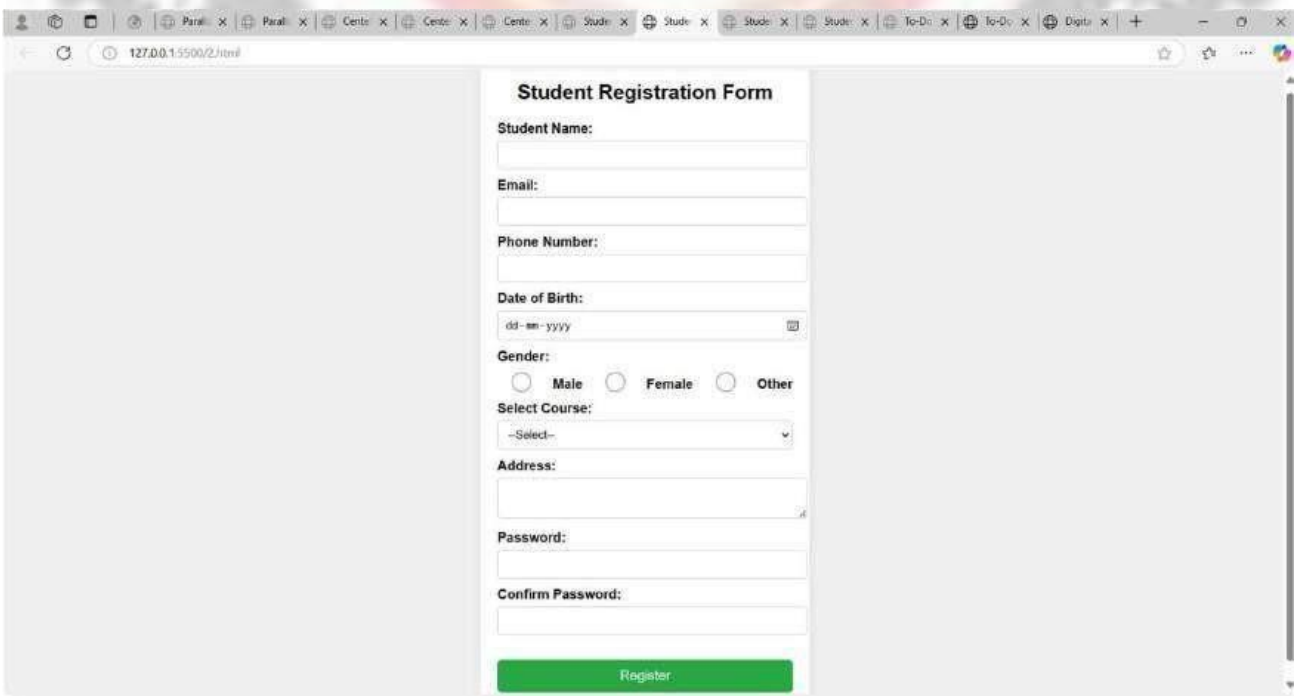
```
< !- -Main.html - -><!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Student Registration Form</title>
<link rel="stylesheet" href="styles2.css">
<script defer src="script.js"></script>
</head>
<body>
<div class="form-container">
<h2>Student Registration Form</h2>
<form id="registrationForm">

<!-- Student Name -->
<label for="name">Student Name:</label>
<input type="text" id="name" name="name" required>

<!-- Email -->
<label for="email">Email:</label>
<input type="email" id="email" name="email" required>

<!-- Phone Number -->
<label for="phone">Phone Number:</label>
<input type="text" id="phone" name="phone" pattern="[0-9]{10}" title="Enter a valid 10- digit
phone number" required>

<!-- Date of Birth -->
<label for="dob">Date of Birth:</label>
<input type="date" id="dob" name="dob" required>

<!-- Gender -->
<label>Gender:</label>
<div class="gender-group">
<input type="radio" id="male" name="gender" value="Male" required>
<label for="male">Male</label>

<input type="radio" id="female" name="gender" value="Female" required>
<label for="female">Female</label>

<input type="radio" id="other" name="gender" value="Other" required>
<label for="other">Other</label>
</div>
```

```
<!-- Course Selection -->
<label for="course">Select Course:</label>
<select id="course" name="course" required>
<option value="">--Select--</option>
<option value="B.Tech">B.Tech</option>
<option value="B.Sc">B.Sc</option>
<option value="BBA">BBA</option>
</select>

<!-- Address -->
<label for="address">Address:</label>
<textarea id="address" name="address" required></textarea>

<!-- Password -->
<label for="password">Password:</label>
<input type="password" id="password" name="password" minlength="6" required>

<!-- Confirm Password -->
<label for="confirmPassword">Confirm Password:</label>
<input type="password" id="confirmPassword" name="confirmPassword" required>

<p id="error-message" class="error-message"></p>

<!-- Submit Button -->
<button type="submit">Register</button>
</form>
</div>
</body>
</html>

<!--style2.css -->

body {
font-family: Arial, sans-serif;
background-color: #f0f0f0;
display: flex;
justify-content: center;
align-items: center;
height: 100vh; margin:
0;
}
.form-container {
background: white;
padding: 20px;
border-radius: 8px;
```

```
box-shadow: 0px 4px 10px rgba(0, 0, 0, 0.1); width:
350px;
}

h2 {
text-align: center;
}

label {
font-weight: bold;
display: block;
margin-top: 10px;
}

input, select, textarea {
width: 100%;  padding:
8px;
margin-top: 5px;
border: 1px solid #ccc;
border-radius: 5px;
}

textarea { resize:
vertical;
}
.gender-group {
display: flex;
gap: 10px;
margin-top: 5px;
}

button {
width: 100%;
background-color: #28a745;
color: white;
border: none;
padding: 10px;
margin-top: 15px;
border-radius: 5px;
cursor: pointer;
font-size: 16px;
}

button:hover {
background-color: #218838;
}
```

```
.error-message {
color: red;
font-size: 14px;
text-align: center;
margin-top: 10px;
}

<!- -script.js - -> document.getElementById("registrationForm").addEventListener("submit",
function(event) {
event.preventDefault(); // Prevent form from submitting

let password = document.getElementById("password").value;
let confirmPassword = document.getElementById("confirmPassword").value; let
errorMessage = document.getElementById("error-message");

if (password !== confirmPassword) { errorMessage.textContent =
"Passwords do not match!";
} else { errorMessage.textContent
= ""; alert("Registration
successful!");
this.submit(); // Submit the form if everything is valid
}
});
```

OUTPUT:



To-Do List – Create a simple to-do list where users can add and remove tasks.

CODE:

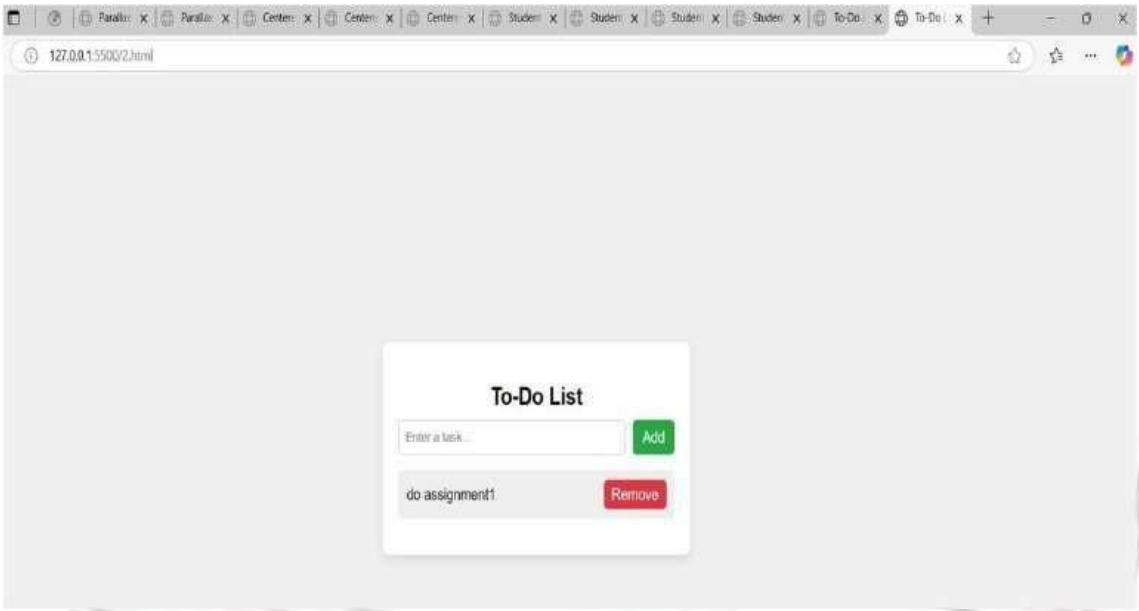
<! –Main.html - - >

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>To-Do List</title>
<link rel="stylesheet" href="styles2.css">
<script defer src="script.js"></script>
</head>
<body>
<div class="todo-container">
<h2>To-Do List</h2>
<div class="input-section">
<input type="text" id="taskInput" placeholder="Enter a task...">
<button onclick="addTask()">Add</button>
</div>
<ul id="taskList"></ul>
</div>

</body>
</html>
<! - -styles2.css - - >
body {
font-family: Arial, sans-serif;
background-color: #f0f0f0;
display: flex;
justify-content: center;
align-items: center;
height: 100vh; margin:
0;
}

.todo-container {
background: white;
padding: 20px;
border-radius: 8px;
```

```
box-shadow: 0px 4px 10px rgba(0, 0, 0, 0.1); width:
350px;
text-align: center;
}
h2 {
margin-bottom: 10px;
}
.input-section {
display: flex; gap:
10px;
}
input {
flex: 1;
padding: 8px;
border: 1px solid #ccc;
border-radius: 5px;
}
button {
background-color: #28a745;
color: white;
border: none;


padding: 8px 12px;
border-radius: 5px;
cursor: pointer;
font-size: 16px;
}

button:hover {
background-color: #218838;
}
ul {
list-style: none;
padding: 0;
margin-top: 15px;
}
```

```
li {
background: #eee;
padding: 10px;
margin: 5px 0;
border-radius: 5px;
display: flex;
justify-content: space-between;
align-items: center;
}

li button {
background-color: #dc3545;
padding: 5px 10px;
}

li button:hover { background-
color: #c82333;
}
<! - -script.js - ->
function addTask() {
let taskInput = document.getElementById("taskInput"); let
taskList = document.getElementById("taskList");
if (taskInput.value.trim() === "") {
alert("Please enter a task."); return;
}

let li = document.createElement("li"); li.innerHTML
= `${taskInput.value} <button
onclick="removeTask(this)">Remove</button>`;
taskList.appendChild(li);

taskInput.value = ""; // Clear input field after adding task
}

function removeTask(button) {
let taskList = document.getElementById("taskList"); taskList.removeChild(button.parentElement);
}
```

OUTPUT:



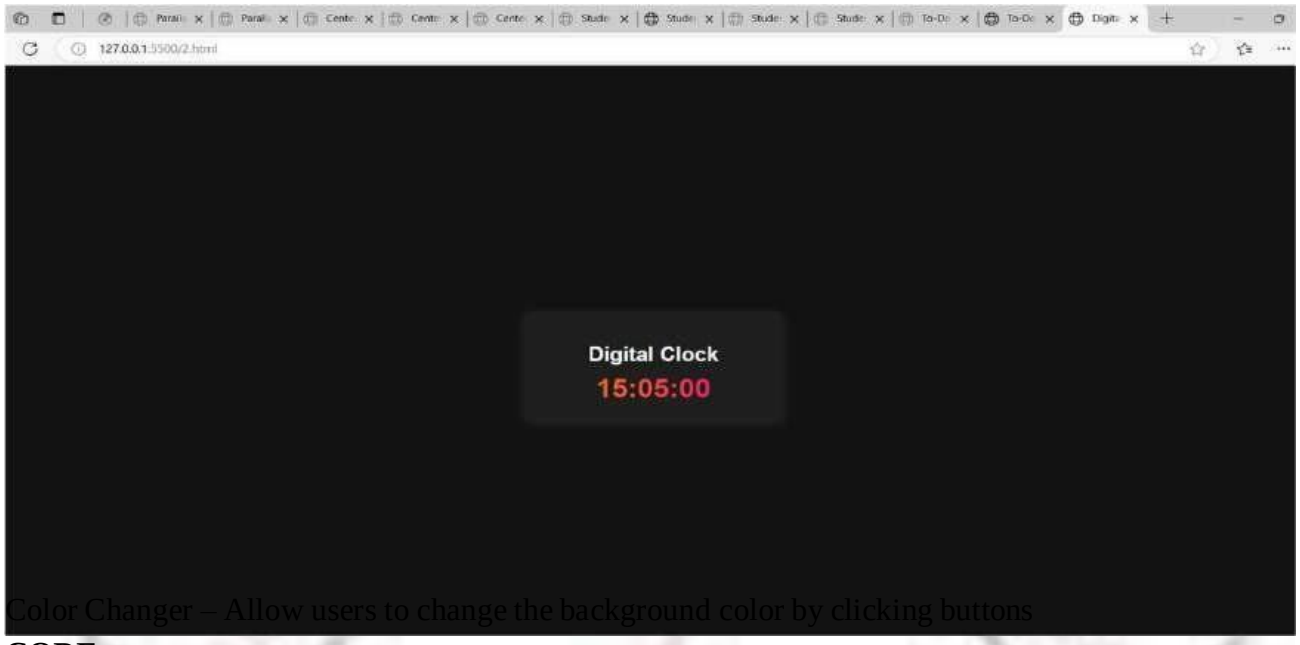
Digital Clock – Display the current time that updates every second.

CODE:

```
<! –Main.html - - >
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Digital Clock</title>
<link rel="stylesheet" href="styles2.css">
<script defer src="script.js"></script>
</head>

<body>
<div class="clock-container">
<h2>Digital Clock</h2>
<div id="clock">00:00:00</div>
</div>
</body>
</html>
<! - -styles2.css - - >
body {
font-family: Arial, sans-serif;
background-color: #121212;
```

```
color: white;
display: flex;
justify-content: center;
align-items: center;
height: 100vh; margin:
0;
}

.clock-container {
background: #1e1e1e;
padding: 20px; border-
radius: 8px;
box-shadow: 0px 4px 10px rgba(255, 255, 255, 0.1);
text-align: center;
width: 250px;
}
h2 {
margin-bottom: 10px;
}
#clock {
font-size: 32px;
font-weight: bold;
background: linear-gradient(90deg, #ff8c00, #ff0080);
background-clip: text;
-webkit-background-clip: text;
-webkit-text-fill-color: transparent;
}
<! - -script.js - -> function
updateClock() { let now =
new Date();
let hours = now.getHours().toString().padStart(2, '0');
let minutes = now.getMinutes().toString().padStart(2, '0'); let
seconds = now.getSeconds().toString().padStart(2, '0');

let timeString = `${hours}:${minutes}:${seconds}`; document.getElementById("clock").textContent =
timeString;
}

setInterval(updateClock, 1000);
updateClock();
```

OUTPUT:



Color Changer – Allow users to change the background color by clicking buttons

CODE:

```
<! –Main.html - - >
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Color Changer</title>
<link rel="stylesheet" href="styles2.css">

<script defer src="script.js"></script>
</head>
<body>
<div class="container">
<h2>Background Color Changer</h2>
<div class="buttons">
<button class="color-btn" data-color="red">Red</button>
<button class="color-btn" data-color="blue">Blue</button>
<button class="color-btn" data-color="green">Green</button>
<button class="color-btn" data-color="yellow">Yellow</button>
<button class="color-btn" data-color="purple">Purple</button>
<button class="color-btn" data-color="random">Random</button>
</div>
</div>
```

```
</body>
</html>
<! - -styles2.css - - >
body {
font-family: Arial, sans-serif;
display: flex;
justify-content: center;
align-items: center;
height: 100vh; margin:
0;
background-color: white;
transition: background-color 0.5s ease-in-out;
}

.container {
text-align: center;
background: #fff;
padding: 20px;
border-radius: 10px;
box-shadow: 0px 4px 10px rgba(0, 0, 0, 0.2);
}
h2 {
margin-bottom: 15px;
}


.buttons {
display: flex;
gap: 10px;
flex-wrap: wrap;
justify-content: center;
}

.color-btn {
border: none;
padding: 10px 15px;
font-size: 16px;
cursor: pointer;
border-radius: 5px;
```

```
transition: transform 0.2s;
}

.color-btn:hover {
transform: scale(1.1);
}
.color-btn:nth-child(1) { background-color: red; color: white; }
.color-btn:nth-child(2) { background-color: blue; color: white; }
.color-btn:nth-child(3) { background-color: green; color: white; }
.color-btn:nth-child(4) { background-color: yellow; color: black; }
.color-btn:nth-child(5) { background-color: purple; color: white; }
.color-btn:nth-child(6) { background-color: gray; color: white; }
<! - -script.js - ->
document.addEventListener("DOMContentLoaded", () => { const
buttons = document.querySelectorAll(".color-btn");

buttons.forEach(button => {
button.addEventListener("click", () => {
const color = button.getAttribute("data-color"); if
(color === "random") {
document.body.style.backgroundColor = getRandomColor();
}
 else {
document.body.style.backgroundColor = color;
}
});


});

function getRandomColor() {
const letters = "0123456789ABCDEF"; let
color = "#";
for (let i = 0; i < 6; i++) {
color += letters[Math.floor(Math.random() * 16)];
}
return color;
}
});
```

OUTPUT:

Image Slider – Create a basic image carousel that auto-plays or changes images on button clicks.

CODE:

<! –Main.html - - >

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Image Slider</title>
<link rel="stylesheet" href="styles2.css">
<script defer src="script.js"></script>
</head>
<body>

<div class="slider-container">
<div class="slider">
<img src="image1.jpg" class="slide" alt="Image 1">
<img src="image2.jpg" class="slide" alt="Image 2">
<img src="image3.jpg" class="slide" alt="Image 3">
<img src="image4.jpg" class="slide" alt="Image 4">
</div>
<button class="prev" onclick="prevSlide()">&#10094;</button>
<button class="next" onclick="nextSlide()">&#10095;</button>
</div>
</body>
</html>
```

```
<! - -styles2.css - - >
body {
display: flex;
justify-content: center;
align-items: center;
height: 100vh; margin:
0;
background-color: #f5f5f5;
}

.slider-container {
position: relative;
width: 500px; height:
300px; overflow:
hidden; border-
radius: 10px;
box-shadow: 0 4px 10px rgba(0, 0, 0, 0.2);
}

.slider {
display: flex;
width: 100%;
height: 100%;
}
.slide { width:
100%;
height: 100%;
object-fit: cover;
display: none;
}


.slide.active {
display: block;
}

button {
position: absolute;
top: 50%;
transform: translateY(-50%);
background-color: rgba(0, 0, 0, 0.5);
color: white;
border: none;
```

```
padding:    10px;
cursor:    pointer;
font-size: 18px;
}

.prev { left:
10px;
}
.next { right:
10px;
}

button:hover {
background-color: rgba(0, 0, 0, 0.8);
}
<! - -script.js - -> let
slideIndex = 0;
const slides = document.querySelectorAll(".slide");

function showSlide(index) {
slides.forEach((slide, i) => {
slide.style.display = i === index ? "block" : "none";
});
}

function nextSlide() {
slideIndex = (slideIndex + 1) % slides.length; showSlide(slideIndex);
}

function prevSlide() {
slideIndex = (slideIndex - 1 + slides.length) % slides.length;
showSlide(slideIndex);
}


setInterval(nextSlide, 3000);
showSlide(slideIndex);
```

OUTPUT:

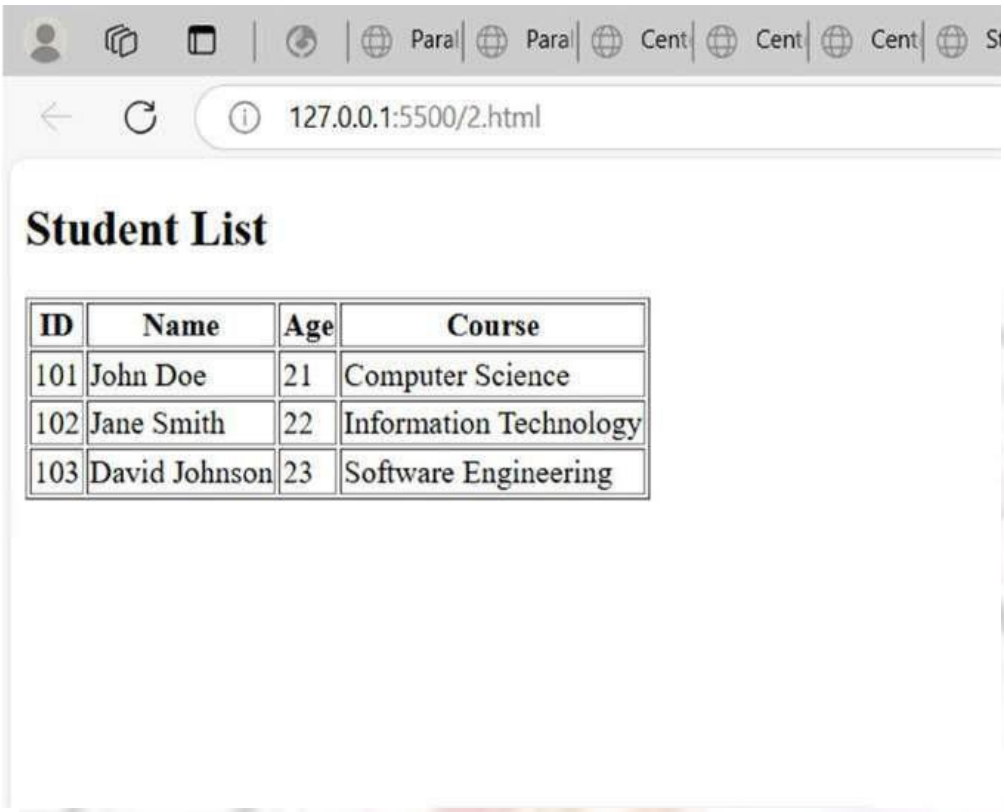
Q4. Create an XML file to store information about students, books, or employees.

CODE:

```
<! - -index.html - ->
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Student List</title>
</head>
<body>
<h2>Student List</h2>
<table border="1">
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Age</th>
<th>Course</th>
</tr>
</thead>
<tbody id="studentTable"></tbody>
</table>

<script>
fetch("students.xml")
.then(response => response.text())
.then(data => {
let parser = new DOMParser();
let xml = parser.parseFromString(data, "application/xml"); let
students = xml.getElementsByTagName("student");

let output = "";
for (let i = 0; i < students.length; i++) {
let id = students[i].getElementsByTagName("id")[0].textContent;
let name = students[i].getElementsByTagName("name")[0].textContent; let age
= students[i].getElementsByTagName("age")[0].textContent;
let course = students[i].getElementsByTagName("course")[0].textContent;

output += `<tr>
```

```
<td>${id}</td>
<td>${name}</td>
<td>${age}</td>
<td>${course}</td>
</tr>`;
}
document.getElementById("studentTable").innerHTML = output;
})
.catch(error => console.log("Error loading XML:", error));
</script>
</body>
</html>
<! - -students.xml - - >
<?xml version="1.0" encoding="UTF-8"?>
<students>
<student>
<id>101</id>
<name>John Doe</name>
<age>21</age>
<course>Computer Science</course>
</student>

<student>
<id>102</id>
<name>Jane Smith</name>
<age>22</age>
<course>Information Technology</course>
</student>
<student>
<id>103</id>
<name>David Johnson</name>
<age>23</age>
<course>Software Engineering</course>
</student>
</students>
```

OUTPUT:


Para
Para
Cent
Cent
Cent
127.0.0.1:5500/2.html
Student List
ID
Name
Age
Course
101
John Doe
21
Computer Science
102
Jane Smith
22
Information Technology
103
David Johnson
23
Software Engineering

Q5. Create a simple CRUD (Create, Read, Update, Delete) web application using PHP and MySQL.

CODE.

```
<?php
$host = "localhost";
$user = "root";
$pass = "";
$db = "student_db";

$conn = new mysqli($host, $user, $pass, $db);

if (!$conn) {
    echo "Connection Failed";
}
 else {
  echo "<div class='connection-message'>Connection established</div>";
}

// Record add karne ke liye
if (isset($_POST['insert'])) {
    $roll = $_POST['roll'];
    $name = $_POST['name'];
    $email = $_POST['email'];

    $sql = "INSERT INTO students (roll, name, email) VALUES ('$roll', '$name', '$email')";
    $conn->query($sql);
}

// record ko update karne ke liye
if (isset($_POST['update'])) {
  $roll = $_POST['roll'];
  $name = $_POST['name'];
  $email = $_POST['email'];

  $sql = "UPDATE students SET name='$name', email='$email', roll='$roll'
        WHERE roll='$roll' OR name='$name' OR email='$email'";
  $conn->query($sql);
}

// record delete karne ke liye
if (isset($_POST['delete'])) {
    $roll = $_POST['roll'];
```

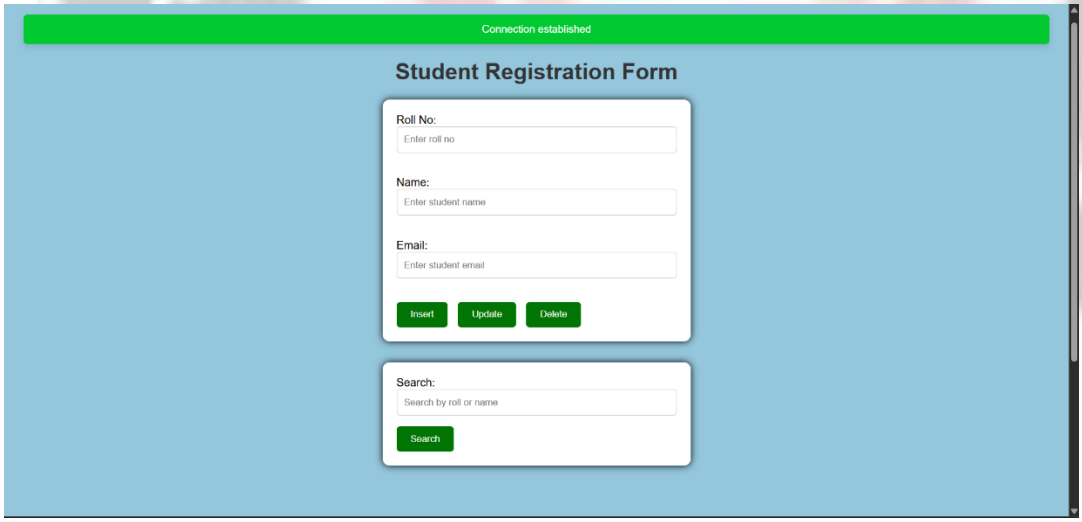
```
    $sql = "DELETE FROM students WHERE roll='$roll'";
    $conn->query($sql);
}

// record ko show karne ke liye
$students = $conn->query("SELECT * FROM students");

// search ke liye
$searchResults = null;
$searchKeyword = '';
if (isset($_POST['searchSubmit']) && !empty($_POST['search'])) {
    $searchKeyword = $_POST['search'];
    $searchResults = $conn->query("SELECT * FROM students WHERE roll LIKE '%$searchKeyword%'
OR name LIKE '%$searchKeyword%' ");
}
?>

<!-- to show landing page -->

<!DOCTYPE html>
<html>
<head>
    <title>Student Registration Form</title>
    <style>
        .connection-message {
            background-color:rgb(0, 201, 47);
            color: white;
            padding: 12px;
            text-align: center;
            font-size: 15px;
            border-radius: 5px;
            box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
            transition: opacity 0.3s ease-in-out;
            opacity: 1;
        }

        .connection-message:hover {
            opacity: 0.8;
        }

        body {
            font-family: Arial, sans-serif;
            background:rgba(6, 122, 175, 0.43);
            padding: 20px;
```

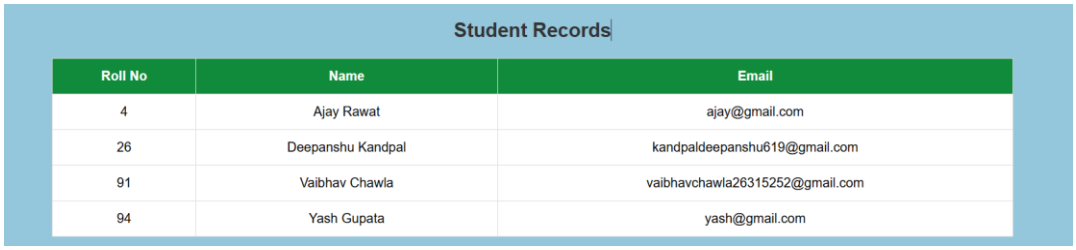
```
}

h1, h2 {
    color: #333;
    text-align: center;
}

form {
    background: #fff;
    padding: 20px;
    max-width: 400px;
    margin: 0 auto 30px auto;
    border-radius: 10px;
    box-shadow: 0 0 10px rgb(0, 0, 0);
}

input[type="text"],
input[type="number"],
input[type="email"] {
    width: 100%;
    padding: 10px;
    margin-bottom: 15px;
    border: 1px solid #ccc;
    border-radius: 5px;
}

button {
    padding: 10px 20px;
    margin-right: 10px;
    background: rgb(0, 117, 4);
    color: #fff;
    border: none;
    border-radius: 5px;
    cursor: pointer;
    transition: background 0.3s ease, transform 0.3s ease;
}

button:hover {
    background: rgb(179, 0, 0);
    transform: scale(1.1);
}

table {
    width: 80%;
```

```
        margin: 0 auto;
        border-collapse: collapse;
        background: #fff;
      }

      th, td {
        padding: 12px;
        text-align: center;
        border: 1px solid #ddd;
      }

      th {
        background-color:rgb(15, 139, 59);
        color: white;
      }
    </style>
</head>
<body>
    <h1>Student Registration Form</h1>
    <form method="post">
      Roll No: <input type="number" name="roll" required placeholder="Enter roll no"><br><br>
      Name: <input type="text" name="name" required placeholder="Enter student name"><br><br>
      Email: <input type="email" name="email" required placeholder="Enter student email"><br><br>

      <button type="submit" name="insert">Insert</button>
      <button type="submit" name="update">Update</button>
      <button type="submit" name="delete">Delete</button>
    </form>
    <form method="post">
      Search: <input type="text" name="search" placeholder="Search by roll or name">
      <button type="submit" name="searchSubmit">Search</button>
    </form>
    <br><br>

    <?php if (isset($searchResults)) { ?>
    <h2>Search Results</h2>
    <table>
      <tr><th>Roll No</th><th>Name</th><th>Email</th></tr>
      <?php if ($searchResults && $searchResults->num_rows > 0) {
        while ($row = $searchResults->fetch_assoc()) { ?>
          <tr>
            <td><?= $row['roll'] ?></td>
            <td><?= $row['name'] ?></td>
            <td><?= $row['email'] ?></td>
```

```
            </tr>
          <?php }
        } else { ?>
          <tr><td colspan="3">No matching records found.</td></tr>
        <?php } ?>
    </table>
<?php } ?>

    <h2>Student Records</h2>
    <table border="1" cellpadding="5">
        <tr><th>Roll No</th><th>Name</th><th>Email</th></tr>
        <?php while ($row = $students->fetch_assoc()) { ?>
            <tr>
                <td><?= $row['roll'] ?></td>
                <td><?= $row['name'] ?></td>
                <td><?= $row['email'] ?></td>
            </tr>
        <?php } ?>
    </table>

</body>
</html>
```

OUTPUT:





Q6. Build a simple calculator using React that can perform basic arithmetic operations: addition, subtraction, multiplication, and division.

CODE.

App.js

```
import React, { useState } from 'react';
import './App.css';
function App() {
const [input, setInput] = useState("");
const handleClick = (value) => {
setInput(input + value);
};
const clear = () => {
setInput("");
};
const backspace = () => {
setInput(input.slice(0, -1));
};
const calculate = () => {
try {
setInput(eval(input).toString());
} catch {
setInput("Error");
}
};
return (
<div className="app-container">
<h1 className="title"> Calculator</h1>
<div className="calculator">
<input type="text" value={input} readOnly className="display" />
<div className="buttons">
<button onClick={clear} className="btn red">C</button>
<button onClick={backspace} className="btn">⌫</button>
<button onClick={() => handleClick("/")} className="btn">/</button>
<button onClick={() => handleClick("*")} className="btn">×</button>
<button onClick={() => handleClick("7")} className="btn">7</button>
<button onClick={() => handleClick("8")} className="btn">8</button>
<button onClick={() => handleClick("9")} className="btn">9</button>
<button onClick={() => handleClick("-")} className="btn">−</button>
<button onClick={() => handleClick("4")} className="btn">4</button>
<button onClick={() => handleClick("5")} className="btn">5</button>
<button onClick={() => handleClick("6")} className="btn">6</button>
<button onClick={() => handleClick("+")} className="btn">+</button>
```

```
<button onClick={() => handleClick("1")} className="btn">1</button>
<button onClick={() => handleClick("2")} className="btn">2</button>
<button onClick={() => handleClick("3")} className="btn">3</button>
<button onClick={calculate} className="btn equals">=</button>
<button onClick={() => handleClick("0")} className="btn wide">0</button>
<button onClick={() => handleClick(".")} className="btn">.</button>
</div>
</div>
</div>
);
}
export default App;
```

Index.js

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import App from './App';
import './App.css';
const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(<App />);
```

App.css

```
body {
margin: 0;
padding: 0;
background: linear-gradient(135deg, #ffffff, #ffffff);
font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
height: 100vh;
display: flex;
align-items: center;
justify-content: center;
}
.app-container {
text-align: center;
}
.title {
margin-bottom: 20px;
color: #000000;
text-shadow: 1px 1px 3px rgba(0, 0, 0, 0.3);
}
.calculator {
background: rgb(0, 4, 27);
backdrop-filter: blur(10px);
border-radius: 20px;
box-shadow: 0 8px 32px 0 rgba(31, 38, 135, 0.3);
padding: 30px 20px;
```

```
width: 320px;
}
.display {
width: 90%;
height: 60px;
font-size: 28px;
margin-bottom: 20px;
border: none;
border-radius: 10px;
text-align: right;
padding: 10px;
background: rgba(255, 255, 255, 0.9);
box-shadow: inset 2px 2px 5px rgba(0,0,0,0.1);
}
.buttons {
display: flex;
f
lex-wrap: wrap;
gap: 10px;
justify-content: center;
}
.btn {
width: 60px;
height: 60px;
font-size: 22px;
border: none;
border-radius: 15px;
background: #ffffffc9;
color: #333;
box-shadow: 2px 2px 10px rgba(0,0,0,0.1);
cursor: pointer;
transition: 0.2s;
}
.btn:hover {
background-color: #e0f7fa;
transform: scale(1.05);
}
.equals {
background-color: #4caf50;
color: white;
}
.red {
background-color:
#f44336;

color: white;
}
.wide {
width: 130px;
}
```

Build a multi-page React application using React Router to

OUTPUT:

Q7. Build a multi-page React application using React Router to demonstrate routing between different components.

CODE.

App.js

```
import React from 'react';
import { BrowserRouter as Router, Routes, Route, Link, useLocation } from 'react-router-dom';
import Home from './pages/Home';
import About from './pages/About';
import Contact from './pages/Contact';
import './App.css';
const NavLink = ({ to, label }) => {
const location = useLocation();
const isActive = location.pathname === to;
return (
<Link to={to} style={{ textDecoration: isActive ? 'underline' : 'none' }}>
{label}
</Link>
);
};
function App() {
return (
<Router>
<nav>
<NavLink to="/" label="Home" />
<NavLink to="/about" label="About" />
<NavLink to="/contact" label="Contact" />
</nav>
<div className="container">
<Routes>
<Route path="/" element={<Home />} />
<Route path="/about" element={<About />} />
<Route path="/contact" element={<Contact />}
/>
</Routes>
</div>
</Router>
);
}
export default App;
```

App.css

```
.App {
text-align: center;
}

.App-logo {
height: 40vmin;
pointer-events: none;
}
@media (prefers-reduced-motion: no-preference)
{
.App-logo {
animation: App-logo-spin infinite 20s linear;
}
}
.App-header {
background-color: #282c34;
min-height: 100vh;
display: flex;
f
lex-direction: column;
align-items: center;
justify-content: center;
font-size: calc(10px + 2vmin);
color: white;
}
```

```
.App-link {
color: #61dafb;
}
@keyframes App-logo-spin {
from {
transform: rotate(0deg);
}
to {
transform: rotate(360deg);
}
}
body {
font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
margin: 0;
padding: 0;
background-color: #f7f9fc;
}
nav {
background-color: #282c34;
padding: 15px 20px;
display: flex;
gap: 20px;
}
nav a {
color: white;
text-decoration: none;
font-weight: bold;
}
```

About.js

```
import React from 'react';
function About() {
return (
<div>
<h2>About Us</h2>
<p>This page contains information about our website.</p>
</div>
);
}
export default About;
```

Home.js

```
import React from 'react';
function Home() {
return (
<div>
<h2>Welcome to the Home Page</h2>
<p>This is the homepage of your multi-page React app.</p>
</div>
);
}
export default Home;
nav a:hover {
text-decoration: underline;
}
.container {
max-width: 800px;
margin: 30px auto;
padding: 20px;
background: white;
border-radius: 10px;
box-shadow: 0 2px 8px rgba(0, 0, 0, 0.1);
}
h2 {
color: #333;
}
```

Contact.js

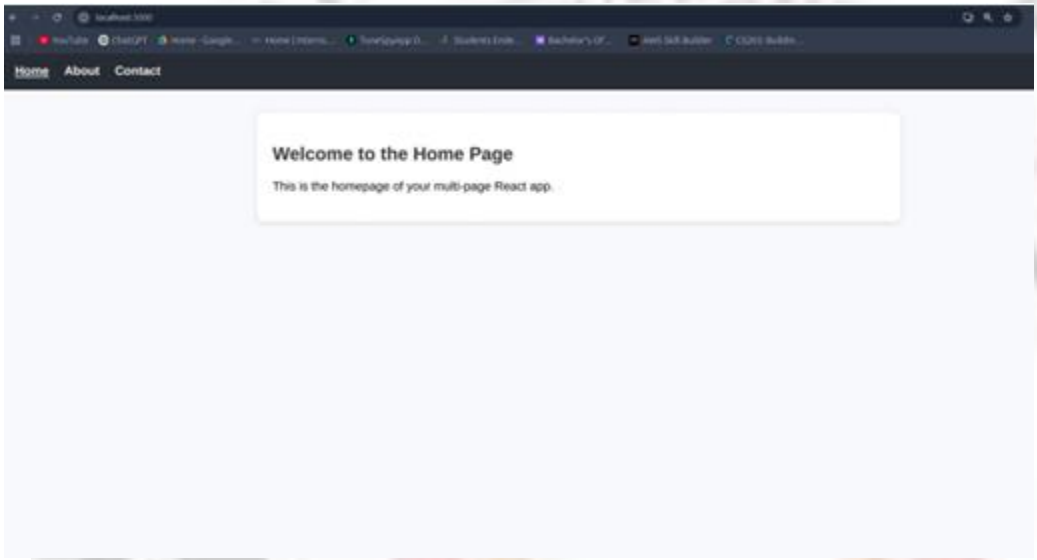
```
import React from 'react';
function Contact() {
return (
<div>
<h2>Contact Page</h2>
<p>You can contact us at contact@example.com</p>
</div>
);
}
export default Contact;
```

Index.js

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import App from './App';
const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
<React.StrictMode>
```

```
<App />
 </React.StrictMode>
 );
```

OUTPUT:



Q8. Create a React component that fetches a list of users from the API https://jsonplaceholder.typicode.com/users using axios and displays their names and email addresses in a list.

CODE.

App.js

```
import React from 'react';
import UserList from './UserList';
import './App.css';
function App() {
return (
<div className="container">
<h1>User List</h1>
<UserList />
</div>
);
}
export default App;
```

App.css

```
.App {
text-align: center;
}
.App-logo {
height: 40vmin;
pointer-events: none;
}
@media (prefers-reduced-motion: no-preference)
{
.App-logo {
animation: App-logo-spin infinite 20s linear;
}
}
.App-header {
background-color: #282c34;
min-height: 100vh;
display: flex;
f
lex-direction: column;
align-items: center;
justify-content: center;
font-size: calc(10px + 2vmin);
color: white;
}
.App-link {
color: #61dafb;
}
@keyframes App-logo-spin {
```

Index.js

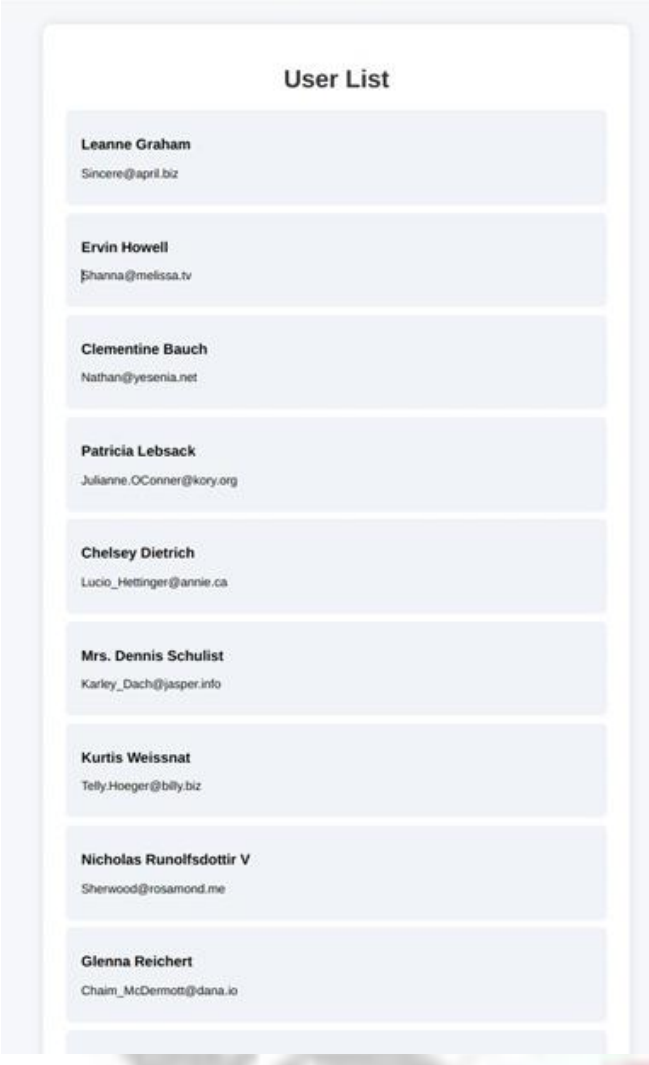
```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';
from {
transform: rotate(0deg);
}
to {
transform: rotate(360deg);
}
}
body {
margin: 0;
padding: 0;
font-family: Arial, sans-serif;
background-color: #f5f7fa;
}
.container {
max-width: 700px;
margin: 40px auto;
background-color: white;
padding: 30px;
border-radius: 12px;
box-shadow: 0 0 12px rgba(0, 0, 0, 0.1);
}
h1 {
text-align: center;
color: #333;
```

```
}

list-style-type: none;
 padding: 0;
 }
 .user-card {
 background: #f0f4f8;
 margin: 10px 0;
 padding: 15px 20px;
 border-radius: 8px;
 transition: 0.2s;
 }
 .user-card:hover {
 background-color: #dfeeff;
 }
const root =
ReactDOM.createRoot(document.getElementById('root'));
 root.render(
 <React.StrictMode>
 <App />
 </React.StrictMode>
 );
 // If you want to start measuring performance in your app, pass a function
 // to log results (for example: reportWebVitals(console.log))
 // or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
 reportWebVitals();
```

UserList.js

```
 import React, { useEffect, useState } from 'react';
 import axios from 'axios';

.user-list {

function UserList() {
const [users, setUsers] = useState([]);
const [loading, setLoading] = useState(true);
useEffect(() => {

axios.get('https://jsonplaceholder.typicode.com/users')
 .then((res) => {
 setUsers(res.data);
 setLoading(false);
 })
 .catch((err) => {
 console.error('Error fetching users:', err);
 setLoading(false);
 });
 }, []);
 if (loading) return <p>Loading users...</p>;
 return (
 <ul className="user-list">
 {users.map((user) => (
 <li key={user.id} className="user-card">
 <h3>{user.name}</h3>
 <p>{user.email}</p>
 </li>
 ))}
 </ul>
 );
 }
 export default UserList;
```

OUTPUT:


User List
Leanne Graham
Sincere@april.biz
Ervin Howell
Shanna@melissa.tv
Clementine Bauch
Nathan@yesenia.net
Patricia Lebsack
Julianne.OConner@kory.org
Chelsey Dietrich
Lucio_Hettinger@annie.ca
Mrs. Dennis Schulist
Karley_Dach@jasper.info
Kurtis Weissnat
Telly.Hoeger@billy.biz
Nicholas Runolfsdottir V
Sherwood@rosamond.me
Glenna Reichert
Chaim_McDermott@dana.io