Problem No: 01

Problem Name: Create a simple HTML page which demonstrates the use of the various types of lists.

Objectives: The objective of this report is to demonstrate the use of various types of lists available in HTML. Lists are a crucial component in HTML, which allow the structuring of content in an organized manner. This report will provide an overview of different types of lists available in HTML and their practical implementation.

Theory: HTML provides three types of lists - ordered, unordered and description lists.

1. Ordered Lists: An ordered list is a list of items where each item is numbered. It can be used to represent a sequence of steps in a process or a set of instructions. The items in an ordered list are placed within <ol> and tags. Each item in the list is defined using <li> and tags. The following code demonstrates how to create an ordered list:

   ```
   <ol>
     <li>Step 1</li>
     <li>Step 2</li>
     <li>Step 3</li>
   </ol>
   ```
2. Unordered Lists: An unordered list is a list of items where each item is marked with a bullet point. It can be used to represent a set of related items. The items in an unordered list are placed within <ul> and tags. Each item in the list is defined using <li> and tags. The following code demonstrates how to create an unordered list:
3.
   ```
   <ul>
     <li>Item 1</li>
     <li>Item 2</li>
     <li>Item 3</li>
   </ul>
   ```
4. Description Lists: A description list is a list of terms and their descriptions. It can be used to provide definitions or explanations of concepts. The items in a description list are placed within <dl> and </dl> tags. The term is defined using <dt> and </dt> tags, while the description is defined using <dd> and </dd> tags. The following code demonstrates how to create a description list:

   ```
   <dl>
     <dt>Term 1</dt>
     <dd>Description 1</dd>
     <dt>Term 2</dt>
     <dd>Description 2</dd>
     <dt>Term 3</dt>
     <dd>Description 3</dd>
   </dl>
   ```

Required Components:

1. HTML file: A simple HTML file containing the code for the lists.
2. Text editor: A text editor is required to write the HTML code.
3. Text editor: A text editor is required to write the HTML code.

Source Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <style>

      h1{
       text-align: center;
      }
       .wrap{
           overflow: hidden;
           width: 100%;

       }
       .unorderlist{
           width: 33%;
           float: left;
       }
       .orderList{
           width: 33%;
           float: left;
```

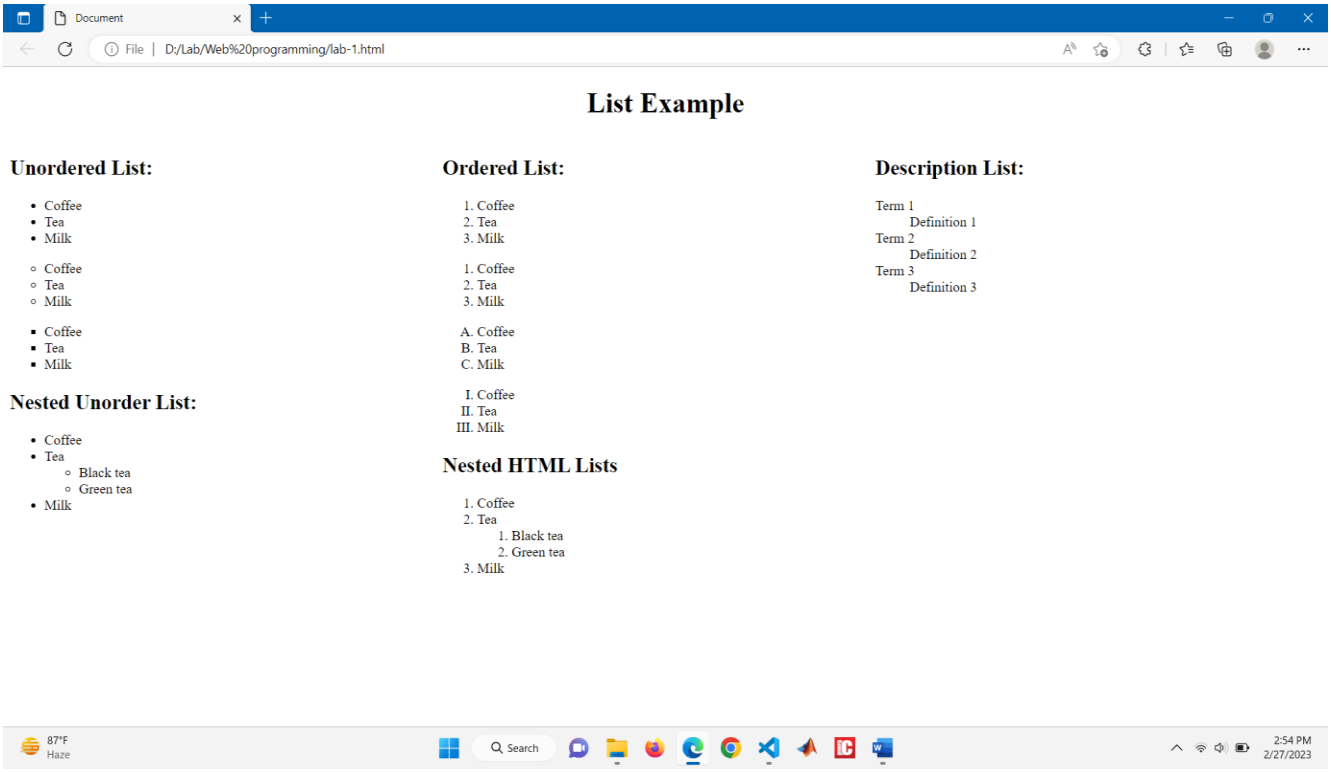
```
        }
        .DescriptionList{
            width: 33%;
            float: left;
        }
    </style>
</head>
<body>
    <h1>List Example</h1>
    <div class="wrap">
        <div class="unorderlist">
            <!-- Unorder list -->
                <h2>Unordered List:</h2>
                <ul style="list-style-type:disc;">
                <li>Coffee</li>
                <li>Tea</li>
                <li>Milk</li>
            </ul>

            <ul style="list-style-type:circle;">
                <li>Coffee</li>
                <li>Tea</li>
                <li>Milk</li>
            </ul>

            <ul style="list-style-type:square;">
                <li>Coffee</li>
                <li>Tea</li>
```

```
            <li>Milk</li>
        </ul>
<!-- Nested unorder list-->
            <h2>Nested Unorder List:</h2>
            <ul>
            <li>Coffee</li>
            <li>Tea
            <ul>
                <li>Black tea</li>
                <li>Green tea</li>
            </ul>
            </li>
            <li>Milk</li>
        </ul>

    </div>
     <!-- Order list-->
    <div class="orderList">
        <h2>Ordered List:</h2>
            <ol>
            <li>Coffee</li>
            <li>Tea</li>
            <li>Milk</li>
        </ol>

        <ol type="1">
            <li>Coffee</li>
            <li>Tea</li>
```

```
            <li>Milk</li>
        </ol>

        <ol type="A">
            <li>Coffee</li>
            <li>Tea</li>
            <li>Milk</li>
        </ol>

        <ol type="I">
            <li>Coffee</li>
            <li>Tea</li>
            <li>Milk</li>
        </ol>
<!--Nested HTML Lists-->

        <h2>Nested HTML Lists</h2>
        <ol>
            <li>Coffee</li>
            <li>Tea
                <ol>
                    <li>Black tea</li>
                    <li>Green tea</li>
                </ol>
            </li>
            <li>Milk</li>
        </ol>
    </div>
```

```
        <div class="DescriptionList">
            <h2>Description List:</h2>
                <dl>
                    <dt>Term 1</dt>
                        <dd>Definition 1</dd>
                    <dt>Term 2</dt>
                        <dd>Definition 2</dd>
                    <dt>Term 3</dt>
                        <dd>Definition 3</dd>
                </dl>
        </div>
    </div>
</body>
</html>
```

Output:



Problem No: 02

Problem Naame: Write HTML page named home.html to create a frameset with two vertical frames: the first frame is 250 pixels wide. Fill the first frame (left_vertical) with links.html. Second frame further divided into two horizontal frames (400px, 350px). Fill the Top frame (right_top) with ice.html and Bottom (right_bottom) with it.html.

Objective'(s): The objective of this report is to demonstrate the process of creating a frameset in HTML, with specific dimensions and contents**.**

Theory: Framesets allow web developers to divide the browser window into separate frames, each of which can display a different HTML document. The frameset is created using the <frameset> element, which specifies the number, size, and location of each frame. The <frame> element is used to define the content of each frame.

Required Components:

1. A text editor or Integrated Development Environment (IDE) for writing HTML code.
2. An HTML document named home.html, which will contain the frameset code.
3. Three additional HTML documents to fill the frames: links.html, ice.html, and it.html.
4. Basic knowledge of HTML syntax and the <frameset> and <frame> elements.

Source Code:

//home.html page

```
<!DOCTYPE html>
<html>
<head>
    <title>Frameset Example</title>
</head>
<frameset cols="250,*">
    <frame src="links.html" name="left_vertical" />
    <frameset rows="400,350">
        <frame src="ice.html" name="right_top" />
        <frame src="it.html" name="right_bottom" />
    </frameset>
```

```
</frameset>
</html>
```

//link.html page

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>


    <style>
        div{
         background-color:crimson;
         height: 650px;
         }
     </style>
</head>
<body>
    <div style = "text-align: center; padding: 30px; color:white;">
        <h2>HELLO WORLD!!</h2>
        <p>I am the FIRST Frame.</p>
    </div>
</body>
</html>
```

```
//ice.html page
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Document</title>
   <style>
      div{
       background-color:gray;
        height:340px;
       }
    </style>

</head>
<body>
   <div style = "text-align: center; padding: 30px; color:black;">
      <h2>HELLO WORLD!!</h2>
      <p>I am the SECOND Frame.</p>
   </div>
</body>
</html>

//it.html page
<!DOCTYPE html>
<html lang="en">
<head>
```

```
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>

    <style>
        div{
         background-color:black;
         height: 400px;
         }
     </style>
</head>
<body>
    <div style = "text-align: center; padding: 30px; color:white;">
        <h2>HELLO WORLD!!</h2>
        <p>I am the THIRD Frame.</p>
    </div>
</body>
</html>
```

Output:

Problem No:03

Problem Name: Create a HTML document giving details of your [Name, Age], [Address, Phone] and [Register Number, Class] aligned in proper order using alignment attributes of Paragraph tag.

Objective's

The objective of this report is to demonstrate how to create a HTML document that provides details of a person's name, age, address, phone number, register number, and class, aligned in proper order using the alignment attributes of the paragraph tag.

Theory:

HTML (Hypertext Markup Language) is the standard markup language used to create web pages. It allows developers to add content to web pages using a set of predefined tags. One such tag is the paragraph tag (<p>), which is used to define a paragraph of text.

The paragraph tag has several alignment attributes that can be used to align text and other content within it. These alignment attributes include align-left, align-center, align-right, and align-justify. By using these alignment attributes, we can align the content of a paragraph tag in the desired order.

Required Components

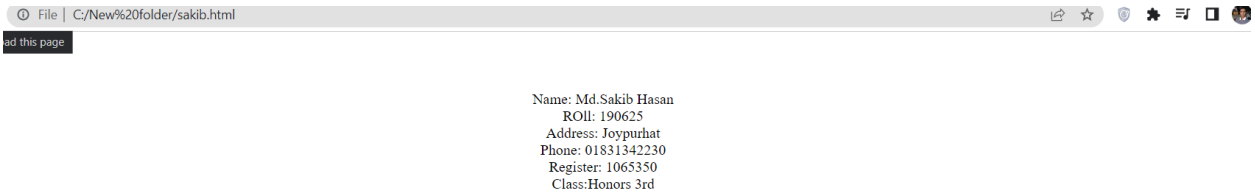
1. **Text editor:** We can use any text editor like Notepad, Sublime Text, or Visual Studio Code to write the HTML code.
2. **Web browser:** We need a web browser to view the HTML document after it's created.

Source Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
```

```
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <style>
        .text_formating{
            text-align: center;
            padding: 40px;
        }
        .list_style{
            list-style: none;
        }
    </style>
</head>
<body>
    <div class="text_formating">
        <p>
            <ul class="list_style">
                <li>Name: Md.Sakib Hasan</li>
                <li>ROll: 190625</li>
                <li>Address: Joypurhat</li>
                <li>Phone: 01831342230</li>
                <li>Register: 1065350</li>
                <li>Class:Honors 3rd </li>
            </ul>

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

Output:



Problem No: 04

Problem Name: Create a web page for internal links; when the user clicks on different links on the webpage it should go to the appropriate locations/sections in the same page.

Objective'(s):

The main objective of this project is to create a web page with internal links that will take the user to different sections of the same page when clicked**.**

Theory:

Internal links are links that navigate to different sections of the same webpage. They are also called bookmarks, jump links, or anchor links. They are useful for making a webpage more interactive and user-friendly. Internal links make it easier for users to navigate through different sections of the same page.

To create an internal link, you need to create an anchor tag and set the href attribute to the ID of the section you want to link to. The ID attribute is used to identify the section you want to link to. You can create an ID by using the id attribute and a unique name for the section.

For example, to create an internal link to a section with the ID "section1", you would create an anchor tag with the href attribute set to "#section1".

Required Components:

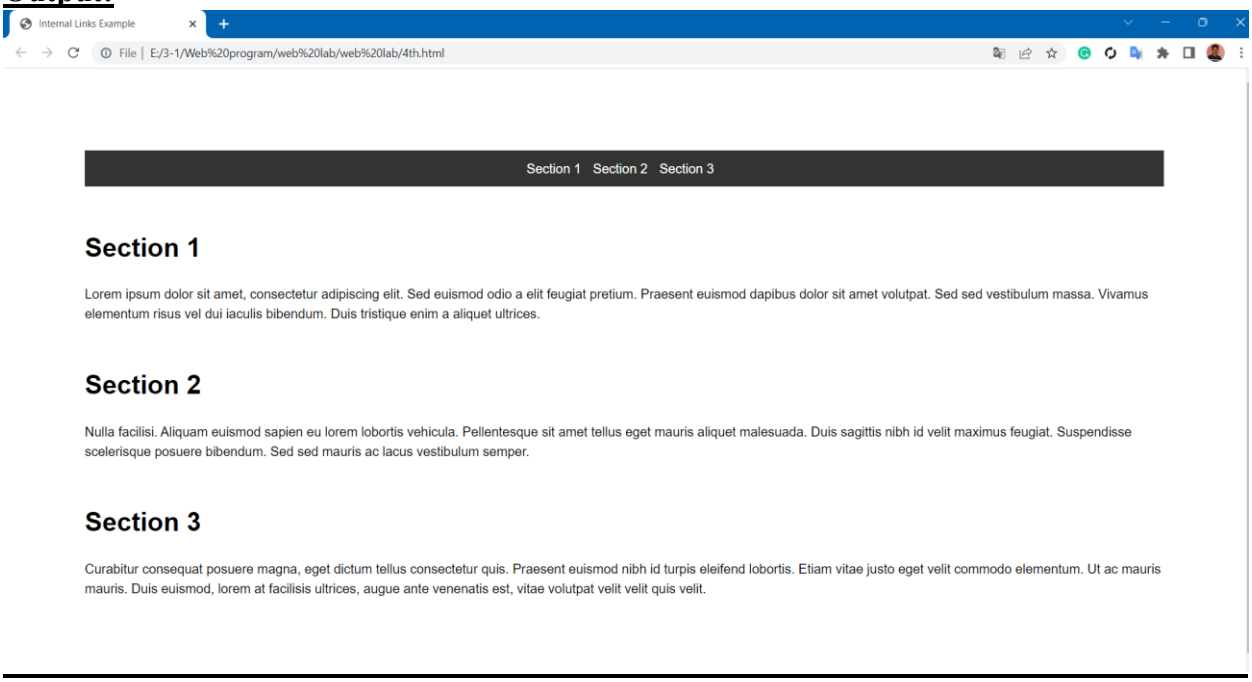
- HTML Editor (e.g., Visual Studio Code, Sublime Text, Atom, etc.)
- Web Browser (e.g., Google Chrome, Mozilla Firefox, Safari, etc.)

Source Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Internal Links</title>
</head>
<body>
    <header>
        <h1>Internal Links Example</h1>
    </header>
    <nav>
        <ul>
            <li><a href="#section1">Section 1</a></li>
            <li><a href="#section2">Section 2</a></li>
            <li><a href="#section3">Section 3</a></li>
            <li><a href="#section4">Section 4</a></li>
        </ul>
    </nav>
    <section id="section1">
        <h2>Section 1</h2>
        <p>This is the first section of the page.</p>
        <p><a href="#top">Back to top</a></p>
    </section>
```

```
        <section id="section2">
            <h2>Section 2</h2>
            <p><a href="#top">Back to top</a></p>
        </section>
        <section id="section3">
            <h2>Section 3</h2>
            <p>This is the third section of the page.</p>
            <p><a href="#top">Back to top</a></p>
        </section>
        <section id="section4">
            <h2>Section 4</h2>
            <p>This is the fourth section of the page.</p>
            <p><a href="#top">Back to top</a></p>
        </section>
</body>
</html>
```

Output:


Internal Links Example
File | E:/3-1/Web%20program/web%20lab/web%20lab/4th.html
Section 1 Section 2 Section 3
Section 1
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed euismod odio a elit feugiat pretium. Praesent euismod dapibus dolor sit amet volutpat. Sed sed vestibulum massa. Vivamus elementum risus vel dui iaculis bibendum. Duis tristique enim a aliquet ultrices.
Section 2
Nulla facilisi. Aliquam euismod sapien eu lorem lobortis vehicula. Pellentesque sit amet tellus eget mauris aliquet malesuada. Duis sagittis nibh id velit maximus feugiat. Suspendisse scelerisque posuere bibendum. Sed sed mauris ac lacus vestibulum semper.
Section 3
Curabitur consequat posuere magna, eget dictum tellus consectetur quis. Praesent euismod nibh id turpis eleifend lobortis. Etiam vitae justo eget velit commodo elementum. Ut ac mauris mauris. Duis euismod, lorem at facilisis ultrices, augue ante venenatis est, vitae volutpat velit velit quis velit.

Problem No:05

Problem Name : Write HTML & CSS code for following table.

A test table with merged cells

	Average		Red eyes
	height	weight	
Males	1.9	0.003	40%
Females	1.7	0.002	43%

Objective('s):

The objective of this report is to explain how to create an HTML & CSS table. We will discuss the theory behind creating tables, the required components, and provide a sample source code.

Theory:

An HTML table is a structured way to present data in rows and columns. Tables are often used to display tabular data, such as financial reports, product catalogs, or sports scores. HTML tables consist of three main components: the table, the rows, and the cells.

The table is created using the <table> element. The rows of the table are created using the <tr> element, and the cells of the table are created using the <td> element. The <th> element is used to create a table header cell.

CSS can be used to style the table, including changing the border, padding, and font properties. We can also style specific table elements, such as table headers or table cells.

Required Components:

1. Text editor: We will need a text editor to write the HTML & CSS code.
2. HTML: We will need HTML to create the structure of the table.
3. CSS: We will use CSS to style the table.

Source Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>

    <link rel="stylesheet" href="style.css">
</head>
<body>
  <table border="1" align="center">
      <caption>
          A test table with merged cells
      </caption>
      <tr>
          <th rowspan="2"></th>
          <th colspan="2">Average</th>
          <th rowspan="2">Red eyes</th>
      </tr>
      <tr>
          <th>
              height
          </th>

          <th>
```

```
            weight
          </th>
        </tr>

        <tr>
          <th>Males</th>
          <td>1.9</td>
          <td>0.003</td>
          <td>40%</td>
        </tr>
        <tr>
          <th>Females</th>
          <td>1.7</td>
          <td>0.002</td>
          <td>43%</td>
        </tr>
    </table>
</body>
</html>
```

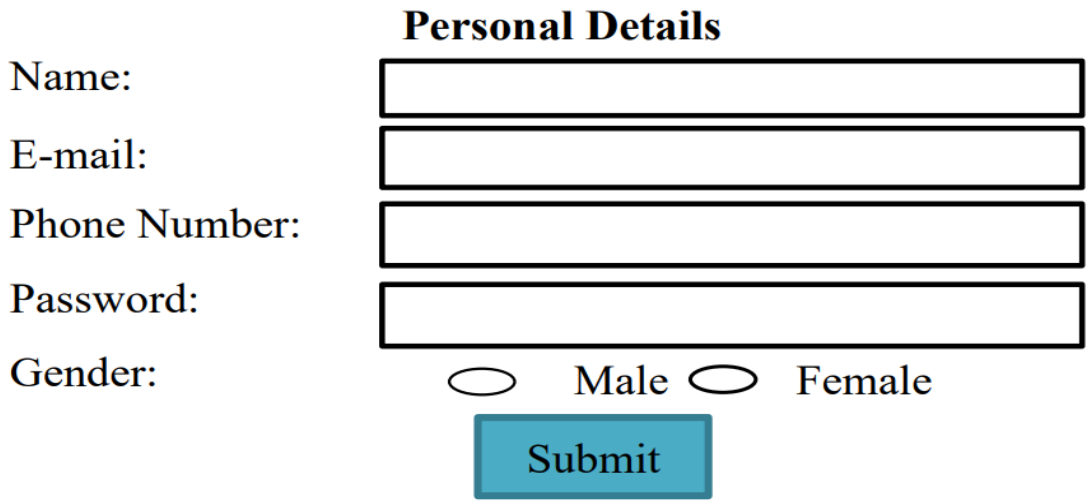
Output:

A test table with merged cells

	Average		Red eyes
	height	weight	
Males	1.9	0.003	40%
Females	1.7	0.002	43%

Problem No:06

Problem Name: Write HTML for the following registration page & use CSS to beautify it as your own choice.



Objective:

The objective of this report is to provide step-by-step guidance on creating a registration page using HTML and CSS.

Theory:

A registration page is a common component of many websites that allow users to create accounts and access specific content or services. A well-designed registration page can attract and retain users, while a poorly designed page can turn them away. HTML and CSS are two programming languages that can be used to create a visually appealing and functional registration page.

Required Components:

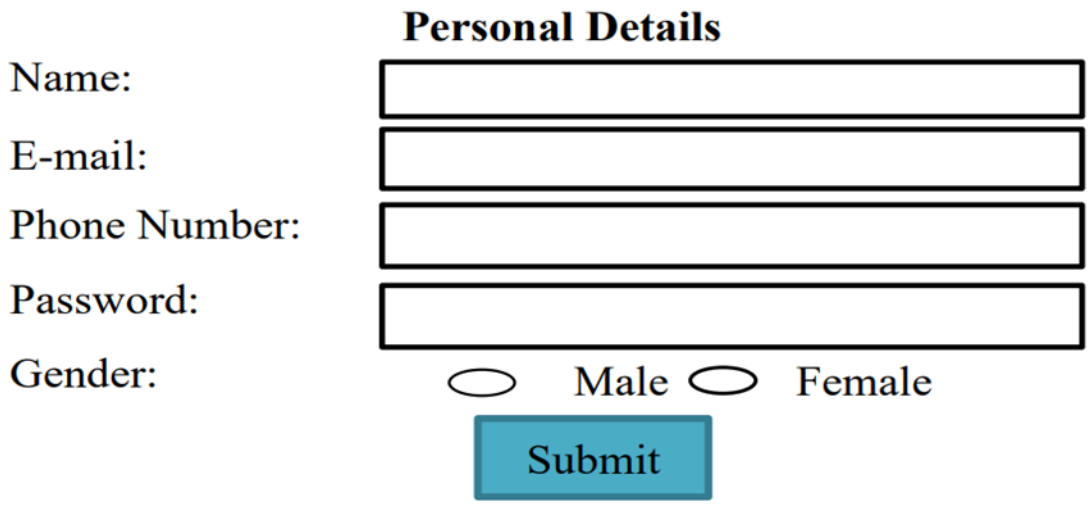
1. Text editor: A text editor is required to write and edit the HTML and CSS code. Popular text editors include Sublime Text, Visual Studio Code, and Notepad++.
2. HTML code: HTML code is used to structure the page and add elements such as text, input fields, and buttons.
3. CSS code: CSS code is used to style the page and make it visually appealing.
4. CSS code: CSS code is used to style the page and make it visually appealing.

Source Code:

```
<!DOCTYPE html>
<html>
<head>
	<title>Registration Form</title>

	<style>
		.error {
			border: 2px solid red;
		}
	</style>
</head>
<body>
	<h1>Registration Form</h1>
	<form id="registration-form" method="post" action="">
		<label for="name">Name:</label>
		<input type="text" id="name" name="name" required>
		<br>
		<label for="password">Password:</label>
		<input type="password" id="password" name="password" required>
		<br>
		<label for="email">Email:</label>
		<input type="email" id="email" name="email" required>
		<br>
		<label for="phone">Phone:</label>
		<input type="tel" id="phone" name="phone" required>
		<br>
		<input type="submit" value="Submit">
```

```
        </form>

        <div class="" id="show_result">

        </div>
        <script src="js1.js"></script>
</body>
</html>
```

Output:



Problem No:07

Problem Name: Write JavaScript to validate the following fields of the Question 06 registration page.

1. Name (Name should contains alphabets and the length should not be less than 6 characters).
2. E-mail (should not contain any invalid and must follow the standard patternname@domain.com).
3. Phone Number (Phone Number should contain 10 digits only).
4. Password (Password should not be less than 6 characters length).

Objective:

The objective of this task is to validate the user inputs for the registration page using JavaScript. The script will ensure that the required fields are not empty, and the input format is correct.

Theory:

JavaScript can be used to add client-side validation to web forms, ensuring that the user enters valid information. In the case of a registration form, this validation can be used to ensure that the user enters valid information in the required fields. There are several ways to perform validation, including using regular expressions and built-in JavaScript functions.

In this case, we will use the built-in JavaScript functions to validate the following fields of the registration form:

1. Name: Ensure that the name field is not empty and contains only letters and spaces.
2. Password: Ensure that the password field is not empty and is at least 6 characters long.
3. Email: Ensure that the email field is not empty and is in the correct format (contains "@" and "." symbols).
4. Phone: Ensure that the phone field is not empty and contains only 10 digits.

Required Components:

To implement this validation, we will need:

1. A registration form with the required fields (name, password, email, and phone).
2. JavaScript code to perform the validation.
3. HTML code to connect the JavaScript code to the registration form.

Source Code:

```
// Javascript Code;

        // Get form inputs
        const nameInput = document.getElementById("name");
        const passwordInput = document.getElementById("password");
        const emailInput = document.getElementById("email");
        const phoneInput = document.getElementById("phone");

        // Define validation functions
        function validateName(name) {
            const regex = /^[a-z A-Z]{6,}$/;
            return regex.test(name);
        }

        function validatePassword(password) {
            return password.length >= 6;
        }

        function validateEmail(email) {
            const regex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
            return regex.test(email);
        }

        function validatePhone(phone) {
            const regex = /^\d{10}$/;
            return regex.test(phone);
        }

        // Define form submit event listener
        const form = document.getElementById("registration-form");
        form.addEventListener("click", function (event) {
            event.preventDefault(); // Prevent form submission

            // Get input values
            const name = nameInput.value;
            const password = passwordInput.value;
            const email = emailInput.value;
            const phone = phoneInput.value;


            let isValid = true;
            if (!validateName(name)) {
                nameInput.classList.add("error");
                isValid = false;
```

```
} else {
    nameInput.classList.remove("error");
}
if (!validatePassword(password)) {
    passwordInput.classList.add("error");
    isValid = false;
} else {
    passwordInput.classList.remove("error");
}
if (!validateEmail(email)) {
    emailInput.classList.add("error");
    isValid = false;
} else {
    emailInput.classList.remove("error");
}
if (!validatePhone(phone)) {
    phoneInput.classList.add("error");
    isValid = false;
} else {
    phoneInput.classList.remove("error");
}

// Submit form if valid
if (isValid) {
    const result = document.getElementById('show_result')

    result.innerHTML = `
     <ul>
                        <li>
                                ${name}
                        </li>
                        <li>
                                ${password}
                        </li>
                        <li>
                                ${email}
                        </li>
                        <li>
                                ${phone}
                        </li>

                </ul>
```

```
        `
        console.log(result);

    }

})

  // Validate inputs
```

Output:

Problem :08

Problem Name: Write a JavaScript program to calculate multiplication and division of two numbers.

Objective: The objective of this program is to allow the user to input two numbers and calculate their multiplication and division using JavaScript. The program should then display the results on the web page**.**

Theory:

JavaScript is a powerful programming language that can be used to perform complex mathematical calculations. In this program, we will use JavaScript to calculate the multiplication and division of two numbers.

The first step in this program is to create an HTML form that will allow the user to input two numbers. We will then use JavaScript to retrieve these numbers and perform the calculations. Once the calculations have been completed, we will display the results on the web page using JavaScript**.**

Required components:

To create this program, you will need the following components

1. An HTML form with two input fields for the numbers
2. JavaScript code to retrieve the values and perform the calculations
3. HTML code to display the results on the web page

Source Code:

```
//Html code
<!DOCTYPE html>
<html>
<head>
    <title>Multiplication and Division</title>
    <script src="script.js"></script>
</head>
<body>
    <form>
        <label for="num1">Enter the first number:</label>
```

```
                    <input type="number" id="num1" name="num1"><br><br>

                    <label for="num2">Enter the second number:</label>
                    <input type="number" id="num2" name="num2"><br><br>

                    <button type="button" onclick="multiply()">Multiply</button>
                    <button type="button" onclick="divide()">Divide</button>
                </form>

                <p id="result"></p>
        </body>
        </html>
//Js code
function multiply() {
  let num1 = parseInt(document.getElementById("num1").value);
  let num2 = parseInt(document.getElementById("num2").value);
  let result = num1 * num2;
  document.getElementById("result").innerHTML = "The result is: " + result;
}

function divide() {
  let num1 = parseInt(document.getElementById("num1").value);
  let num2 = parseInt(document.getElementById("num2").value);
  let result = num1 / num2;
  document.getElementById("result").innerHTML = "The result is: " + result;
}
```

Problem No:09

Problem Name: Write a JavaScript for loop that will iterate from 0 to 15. For each iteration, it will check if the current number is odd or even, and display a message to the screen.

Objective: The objective of this program is to use a for loop in JavaScript to iterate from 0 to 15, check if the current number is odd or even, and display a message to the screen.

Theory: In JavaScript, a for loop is used to iterate over a range of values. The for loop consists of an initialization statement, a condition, and an update statement. The loop will continue to execute as long as the condition is true. In this program, we will use the for loop to iterate from 0 to 15 and check if each number is odd or even using the modulo operator (%).

Required Components:

To write and run this program, we will need a text editor and a web browser.

Source Code:

```
// Iterate from 0 to 15
for (let i = 0; i <= 15; i++) {
  // Check if current number is even
  if (i % 2 === 0) {
    console.log(i + " is even");
  }
  // Check if current number is odd
  else {
    console.log(i + " is odd");
  }
}
```

10 is even.
1 is odd.
2 is even.
3 is odd.
4 is even.
5 is odd.
6 is even.
7 is odd.
8 is even.
9 is odd.
10 is even.
11 is odd.
12 is even.
13 is odd.
14 is even.
15 is odd.

Problem No:10

Problem Name: Write a PHP program to calculate Electricity bill in single page.

Conditions:

1. For units less 50 – Taka. 3.50/unit
2. For units 51 to 100 – Taka. 4.00/unit
3. For units 101 to 200 – Taka. 5.20/unit
4. For units above 250 – Taka. 6.50/unit

Objective: The objective of this program is to calculate the electricity bill based on the number of units consumed using PHP programming language.

Theory:

Electricity billing is an essential aspect of any household or business. This PHP program calculates the electricity bill based on the number of units consumed by the customer. The cost per unit varies based on the number of units consumed. The formula to calculate the electricity bill is:

For units less than or equal to 50, the cost per unit is Taka. 3.50

For units between 51 to 100, the cost per unit is Taka. 4.00

For units between 101 to 200, the cost per unit is Taka. 5.20

For units above 250, the cost per unit is Taka. 6.50

The PHP program takes the number of units consumed as input from the user and calculates the total electricity bill based on the above formula.

Required Components:

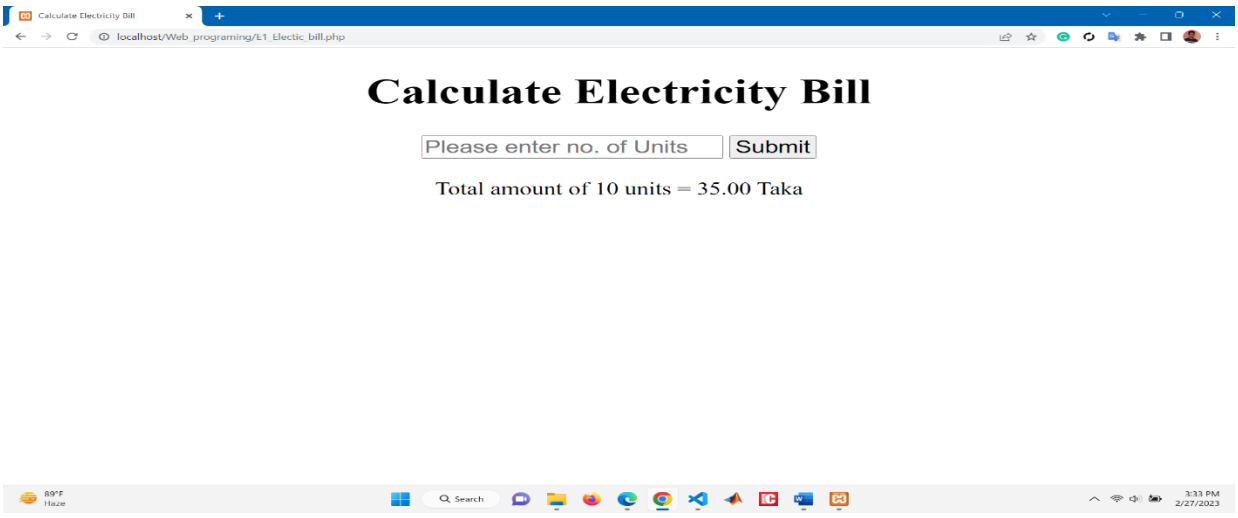
1. PHP programming language
2. HTML form to take user input
3. PHP code to calculate the electricity bill
4. Web server to run the PHP code

Source Code:

```
php
<!DOCTYPE html>
<html>
  <head>
    <title>Electricity Bill Calculator</title>
  </head>
  <body>
    <h1>Electricity Bill Calculator</h1>
    <form method="POST" action="">
      <label for="units">Enter the number of units consumed:</label>
      <input type="text" id="units" name="units"><br><br>
      <input type="submit" value="Calculate">
    </form>
    <?php  if($_POST){
```

```
        $units = $_POST['units'];
        $total_cost = 0;
        if($units <= 50){
          $total_cost = $units * 3.50;
        }
        elseif($units > 50 && $units <= 100){
          $total_cost = 50 * 3.50 + ($units - 50) * 4.00;
        }
        elseif($units > 100 && $units <= 200){
          $total_cost = 50 * 3.50 + 50 * 4.00 + ($units - 100) * 5.20;
        }
        else{
          $total_cost = 50 * 3.50 + 50 * 4.00 + 100 * 5.20 + ($units - 250) * 6.50;
        }
        echo "<br><br><b>Total Electricity Bill:</b> Tk. ".$total_cost;
      }
    ?>
  </body>
</html>
```

Output:



Problem No:11

Problem Name: Write a simple calculator program using PHP in single page.

1. Operations:
2. Addition
3. Subtraction
4. Multiplication
5. Division

Objective: The objective of this program is to create a simple calculator using PHP to perform basic arithmetic operations such as addition, subtraction, multiplication, and division.

Theory: PHP is a popular server-side scripting language that is widely used to create dynamic web pages. It provides many built-in functions that can be used to perform various operations such as arithmetic, string manipulation, and more. In this program, we will use PHP to perform basic arithmetic operations.

Required Components:

1. A web server with PHP installed, such as XAMPP or WAMP.
2. An HTML form to accept input from the user.
3. PHP code to perform the calculations and display the results.

Source Code:

```
<!DOCTYPE html>
<html>
<head>
    <title>Simple Calculator</title>
</head>
<body>
    <h1>Simple Calculator</h1>
    <form action="calculator.php" method="post">
        <label for="num1">Number 1:</label>
        <input type="number" name="num1" id="num1" required><br><br>
```

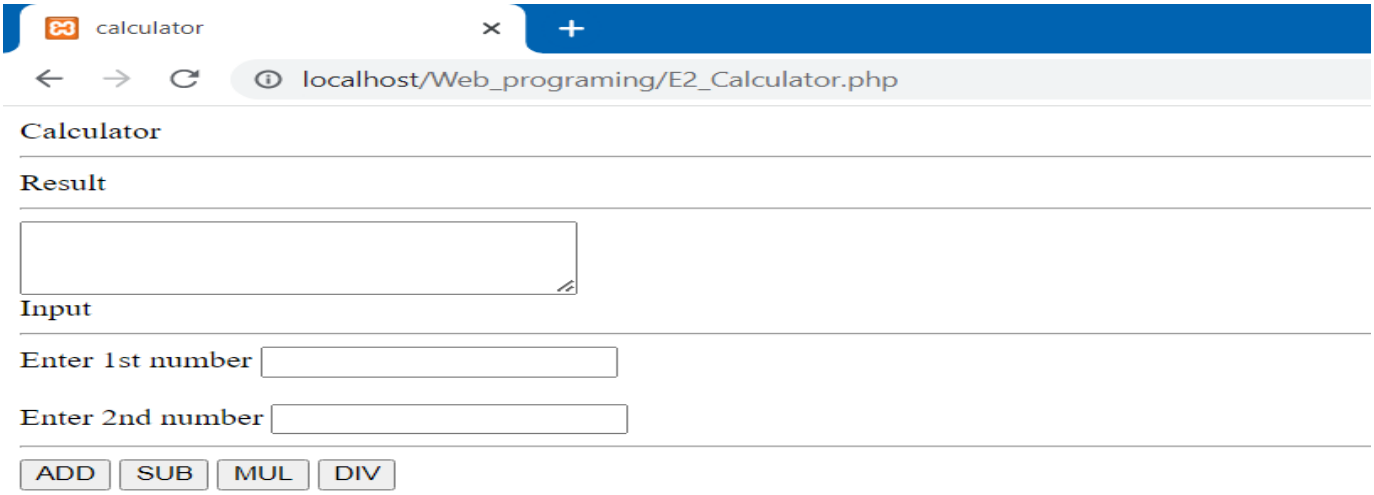
```
            <label for="num2">Number 2:</label>
            <input type="number" name="num2" id="num2" required><br><br>
            <label for="operator">Operator:</label>
            <select name="operator" id="operator">
                <option value="+">Addition</option>
                <option value="-">Subtraction</option>
                <option value="*">Multiplication</option>
                <option value="/">Division</option>
            </select><br><br>
            <input type="submit" value="Calculate">
        </form>
</body>
</html>
```

The PHP code for "calculator.php" is as follows:

```
<?php
if(isset($_POST['num1']) && isset($_POST['num2']) && isset($_POST['operator'])){
    $num1 = $_POST['num1'];
    $num2 = $_POST['num2'];
    $operator = $_POST['operator'];

    switch($operator){
        case "+":
            $result = $num1 + $num2;
            break;
        case "-":
            $result = $num1 - $num2;
```

```
                break;
            case "*":
                $result = $num1 * $num2;
                break;
            case "/":
                if($num2 == 0){
                    $result = "Cannot divide by zero";
                }
                else{
                    $result = $num1 / $num2;
                }
                break;
            default:
                $result = "Invalid operator";
        }

        echo "Result: ".$result;
}
```

Output:



Problem No:12

Problem Name:

A. Solve the following Task-1 and Task-2.

Task-1: Create a database called Student in XAMPP MySQL.

Task-2: Create a table called Semester_Reg in the Student database having the

structure as shown below:

Field name	**Data type**	**Requirement**
ID	Number/Text	Mandatory and primary key
Name	Text	Mandatory
Session	Text	Must follow the format like **2017-2018**
Phone_No	Text	Optional
City	Text	For example **Pabna**
Gender	Text	Only (**Male or Female**)

B. Solve the following marked tasks.

Task 3: Insert some sample data into Semester_Reg table using PHP program.

Task 4: Write a PHP program to show the all records of Semester_Reg table.

Task 5: Delete single sample data from Semester_Reg table using PHP program.

Task 6: Update one sample data of Semester_Reg table using PHP program

Objective:

The objective of this report is to explain the process of creating and managing a MySQL database for a student information system. The report includes the creation of a database, a table within the database, and performing CRUD (Create, Read, Update, and Delete) operations on the table using PHP.

Theory:

MySQL is an open-source relational database management system. It is commonly used in web applications to store and manage data. PHP is a popular server-side scripting language used to create dynamic web pages. The combination of MySQL and PHP provides an efficient way to manage data in web applications.

In this report, we will create a database named Student in XAMPP MySQL and create a table named Semester_Reg in the Student database. The Semester_Reg table will have fields such as ID, Name, Session, Phone_No, City, and Gender.

The CRUD operations will be performed on the Semester_Reg table using PHP. We will insert some sample data into the table, display all the records of the table, delete a single sample data, and update one sample data using PHP.

Required Components:

1. XAMPP Server
2. MySQL
3. PHP

Source Code

```
CREATE TABLE Semester_Reg (
ID INT PRIMARY KEY,
Name VARCHAR(50) NOT NULL,
Session VARCHAR(9) NOT NULL,
Phone_No VARCHAR(15),
City VARCHAR(50),
Gender VARCHAR(10) NOT NULL
);
```

```
<?php
//$connect=mysqli_connect("localhost","root","","Student");
include("connection.php");
//require();
//Insert start
if(isset($_POST["insert"]))
{
        $id =$_POST["id"];
        $name=$_POST["name"];
```

```
	$sess=$_POST["session"];
	$phone=$_POST["ph_number"];
	$city=$_POST["city"];
	$gender=$_POST["gender"];
	$insert="insert into semester_reg(ID,Name,Session,Ph_Num,City, Gender)
values('$id','$name','$sess', '$phone', '$city', '$gender')";
	$result=mysqli_query($connect,$insert);
	if($result==1)
	{
		echo"Successfully insert a record!";
	}
	else
	{
		echo"Unsucess";
	}
}  //Insert end


//Delete start
if(isset($_POST["delete"]))
{
	$id =$_POST["id"];
	$name=$_POST["name"];
	$sess=$_POST["session"];
	$delete="delete from semester_reg where ID='$id'and Name='$name' and
Session='$sess'";
	$result=mysqli_query($connect,$delete);
	if($result==1)
	{
		echo"Successfully delete your record!";
```

```
	}
	else
	{
		echo"Unsucess";
	}
} //Delete end


//update start
if(isset($_POST["update"]))
{
	$id =$_POST["id"];
	$name=$_POST["name"];
	$sess=$_POST["session"];
	$phone=$_POST["ph_number"];
	$city=$_POST["city"];
	$gender=$_POST["gender"];
	$insert="update semester_reg  set Name='$name',Session='$sess',Ph_Num='$phone',
City='$city',Gender='$gender' where ID='$id'";
	$result=mysqli_query($connect,$insert);
	if($result==1)
	{
		echo"Successfully updated your record!";
	}
	else
	{
		echo"Unsucess";
	}
}//update end
```

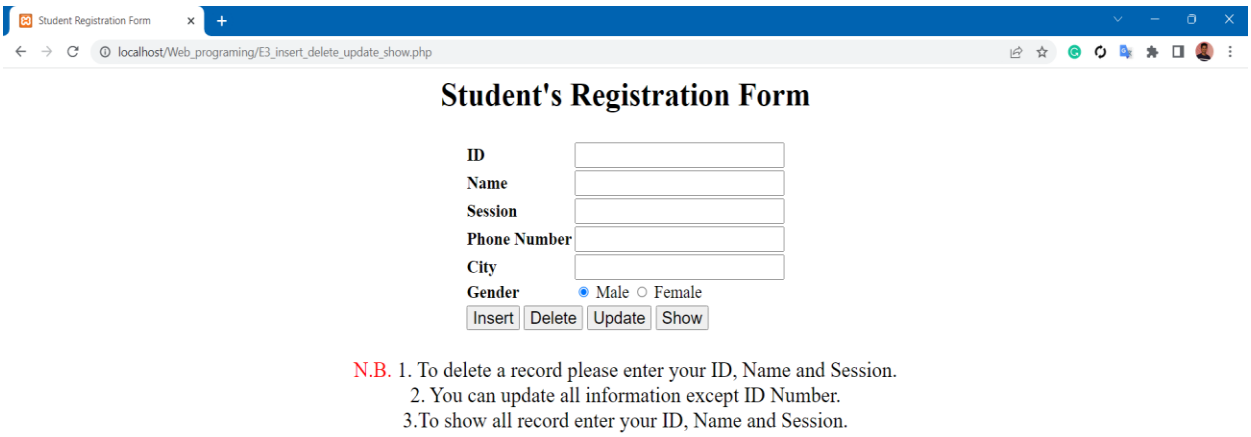
```
//show data start
if(isset($_POST["select"])){

$query="SELECT * FROM semester_reg"; //ORDER BY id ASC";";
$result=mysqli_query($connect,$query);
if($result==true){
        echo "All  Registered Students List <br>";
echo "<table cellpadding=10 border='1'>
<tr>
<th>ID</th>
<th>Name</th>
<th>Session</th>
<th>Phone Number</th>
<th>City</th>
<th>Gender</th>
</tr>";
 if(mysqli_num_rows($result) > 0)
{
while($row = mysqli_fetch_array($result))
{

echo "<tr>";
echo "<td style='color:black'>" . $row['ID'] ."</td>";
echo "<td style='color:black'>" . $row['Name'] . "</td>";
echo "<td style='color:black'>" . $row['Session'] . "</td>";
echo "<td style='color:black'>" . $row['Ph_Num'] . "</td>";
echo "<td style='color:black'>" . $row['City'] . "</td>";
echo "<td style='color:black'>" . $row['Gender'] . "</td>";
```

```
echo "</tr>";
}
echo "</table>";

}
} else
{
        echo "No record found!";
}
}
//end of show data

?>
<html>
<head>
        <title>Student Registration Form</title>
        <style type="text/css">
                body{
                        text-align: center;
        font-size: 25px;
                }
                input{
                        font-size: 20px;
                }
                table
                {
                        margin: auto;
                        font-size: 20px;
```

```
                }
        </style>
</head>
<body>
        <h2>Student's Registration Form </h2>
<form method="post"action="">
<table border="0" style="text-align:left" >
      <tr>
        <th >ID</th>
        <td><input type="text" name="id"  required></td>
       </tr>
                <tr>
        <th>Name</th>
      <td><input type="text" name="name"  required></td>
       </tr>
       <tr>
        <th>Session</th>
        <td><input type="text" name="session" required></td>
       </tr>
       <tr>
        <th >Phone Number</th>
        <td><input type="text" name="ph_number"></td>
       </tr>
                <tr>
        <th >City</th>
        <td><input type="text" name="city" value="" ></td>
                </tr>
                <tr>
```

```
<th>Gender</th>
<td><input type="radio" name="gender" value="Male" checked> Male
    <input type="radio" name="gender" value="Female"> Female
 </td>
            </tr>
            <tr>
                <td colspan="4">
                    <input type="submit" name="insert" value="Insert">
    <input type="submit" name="delete" value="Delete">
    <input type="submit" name="update" value="Update">
    <input type="submit" name="select" value="Show">
                </td>
            </tr>
            </table>
            <br>
    <label style="color:red">N.B. </label> 1. To delete a record please enter your ID, Name and Session.<br>
        2. You can update all information except ID Number. <br>
        3.To show all record enter your ID, Name and Session.
</form></body></html>
```

Output:



Problem No:13

Problem Name: A. Solve the following Task-1 and Task-2.

Task-1: Create a database called Programmer- in XAMPP MySQL.

Task-2: Create a table called Stu_Reg in the Programmer database having the

structure as shown below

Field name	Data type	Requirement
ID	Varchar (30)	Mandatory and primary key
Name	Text	Optional
Image	Varchar(400)	Optional
Password	Number/ Varchar (20)	Mandatory

B. Solve the following marked tasks.

Task 3: Insert some sample data into Stu_Reg table including an encryption algorithm

to secure the password.

Task 4: Write a PHP program to show the all records of Stu_Reg table.

Task 5: Delete single sample record from Stu_Reg table using PHP program

Objective:

The objective of this report is to provide step-by-step instructions to create a MySQL database and table in XAMPP, and implement basic CRUD (Create, Read, Update, and Delete) operations using PHP.

Theory:

XAMPP is an open-source software package that provides a local server environment for developing and testing web applications. It includes Apache, MySQL, PHP, and Perl. MySQL is a popular open-source relational database management system used to manage and store data.

To create a MySQL database in XAMPP, we need to follow the following steps:

Open XAMPP and start Apache and MySQL.

Open a web browser and navigate to http://localhost/phpmyadmin/.

Click on the "New" button in the left sidebar and enter the name of the database in the "Database name" field. Click on the "Create" button.

To create a table in the database, we need to follow these steps:

Select the database from the left sidebar and click on the "SQL" tab.

Enter the SQL query to create the table with the specified fields and data types.

Click on the "Go" button to execute the query.

To implement CRUD operations using PHP, we need to follow these steps:

Establish a connection to the MySQL database using the mysqli_connect() function.

Write the PHP code to implement CRUD operations using MySQL queries (INSERT, SELECT, UPDATE, DELETE).

Required Components:

1. XAMPP server
2. Web browser
3. Text editor
4. MySQL database
5. PHP

Source Code:

```
<?php
$connect=mysqli_connect("localhost","root","","Programmer");

//Insert start

if(isset($_POST["insert"]))
{
        $id =$_POST["id"];
        $name=$_POST["name"];
```

```
        //image
        $img=$_FILES["img"]["name"];
        // $extention=pathinfo($img,PATHINFO_EXTENSION); Use to rename the image
        // $img_new_name=$id.'.'.$extention;
        $password=$_POST["password"];
        //encrypt your password
        $pass = md5($password);
        //echo $password;
        $insert="INSERT INTO Stu_Reg(ID,Name,Image,Password) VALUES
('$id','$name','images/$img', '$pass')";
        $result=mysqli_query($connect,$insert);


        //upload image
      move_uploaded_file($_FILES['img']['tmp_name'], "images/" .
$_FILES['img']['name']);
        if($result==1)
        {
                echo"Successfully insert your record!";
        }
        else
        {
                echo"Unsucess";
        }
} //insert End
//delete start
if(isset($_POST['delete']))
    {
   $id = $_POST['id'];
   $password = $_POST['password'];
```

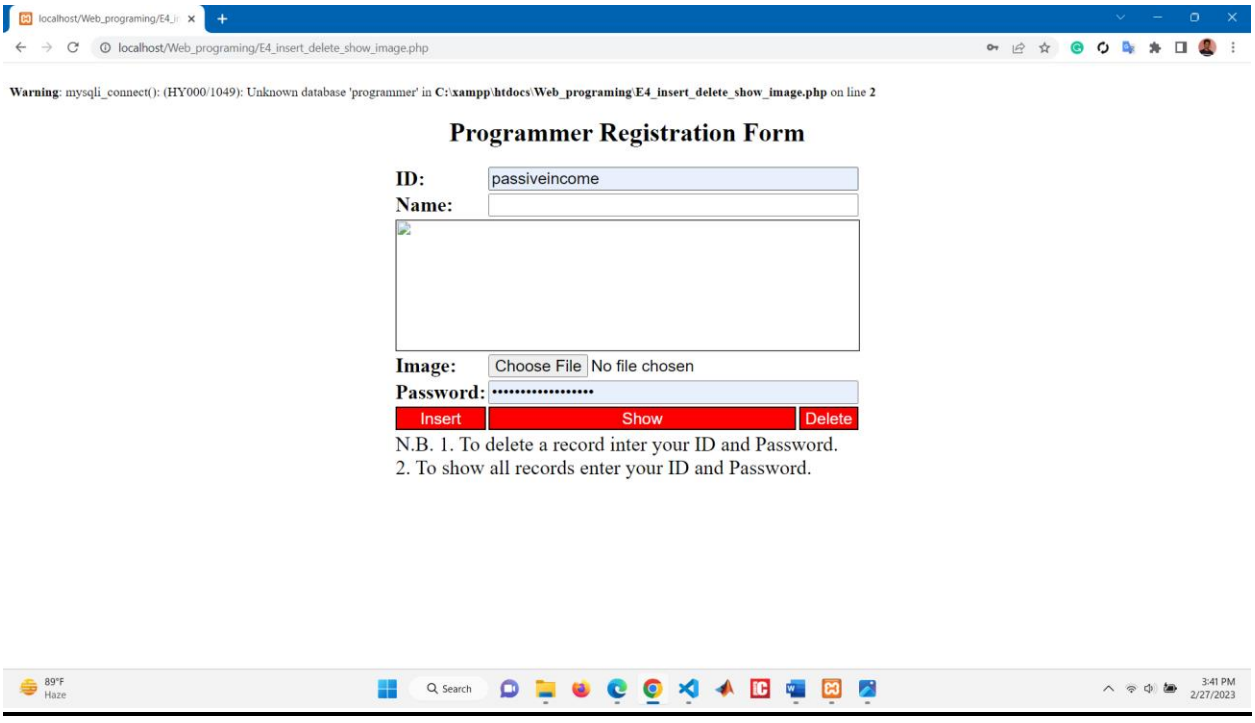
```
    $pass=md5($password);
    $query="SELECT * FROM Stu_Reg where ID = '$id' and Password='$pass'";
        $result=mysqli_query($connect,$query);
        $row = mysqli_fetch_array($result);
   $query = "DELETE FROM Stu_Reg where ID = '$id' and Password='$pass'";
   $execute = mysqli_query($connect,$query);
   if($execute)
    {
          //remove image

                $image=$row['Image'];
            unlink("$image");
           echo "Succesfully deleted your record";
    }
   else
{
 echo "Unsucess";
}
}//delete end


//show data from database
if(isset($_POST["select"])){

$query="SELECT * FROM Stu_Reg"; //ORDER BY id ASC";";
$result=mysqli_query($connect,$query);

 if(mysqli_num_rows($result) > 0)
```

```
{
                    ?>
 <table cellpadding=10 border='1'>
<tr>
<th>ID</th>
<th>Name</th>
<th>Image</th>
</tr>
 <?php
while($row = mysqli_fetch_array($result))
{
?>
<tr>
<td style='color:black'><?php echo $row['ID']?></td>
<td style='color:black'><?php echo $row['Name']?></td>
<td style='color:black'> <img width=100px height=80px src="<?php echo
$row['Image']?>"></td>
</tr>
<?php
}
?>
</table>
<?php
}
else
{
        echo "No Data Found!";
}
}
```

```
//end of show database
?>
<html>
<head>
<script>
function change(event)
{
	var output=document.getElementById('image_change');
	output.src=URL.createObjectURL(event.target.files[0]);
}
</script>

<style type="text/css">
	table
	{
	   margin: auto;
		font-size: 25px;
		text-align: left;
	}
	input
	{
		font-size: 20px;
		width: 100%;
	}
	button
	{
		width: 100%;
```

```
            font-size: 20px;
            background-color: red;
            color: white;
            cursor: pointer;
        }
</style>
</head>
<body>
        <h1 style="text-align:center;">Programmer Registration Form</h1>
<form method="post" action="" enctype="multipart/form-data">
        <table border="0">
            <tr>
                <th>ID:</th>
                <td colspan="2"><input type="text"name="id" required> </td>
            </tr>
            <tr>
                <th>Name:</th>
                <td colspan="2"> <input type="text"name="name"></td>
            </tr>
            <tr >
                <th colspan="3"><img id="image_change"
src="images/man_icon.jpg" height="160px" width="100%" border="1"></th>
            </tr>
            <tr>
                <th >Image:</th>
                <td><input type="file" name="img"
id="img_id"onchange="change(event)"></td>
            </tr>
            <tr>
```

```
                <th>Password:</th>
                <td colspan="2"><input type="password" name="password"
required></td>
            </tr>
            <tr >
                <th><button name="insert">Insert</button></th>
                <th><button name="select">Show</button></th>
                <th><button name="delete">Delete</button></th>
            </tr>
            <tr>
                <td colspan="3">
                    N.B. 1. To delete a record inter your ID and Password.<br>
                    2. To show all records enter your ID and Password.
                </td>
            </tr>
</table>
</form>
```



</body>

</html>