

**PABNA UNIVERSITY OF SCIENCE AND TECHNOLOGY**

**DATE OF SUBMISSION: 01/11/2023**

**SIGNATURE**

**SUBMITTED to**

**MD. Imran Hossain**

**Associate Professor**

**Department of Information and Communication Engineering, Pabna University of Science and Technology.**

**SUBMITTED BY**

**Sajeeb Kumar Ray**

**Roll: 200622**

**Session: 2019-20**

**3rd year and 1st Semester**

**Department of Information and Communication Engineering, Pabna University of Science and Technology.**

**COURSE CODE: ICE-3104**

**COURSE TITLE: Web Programming Sessional**

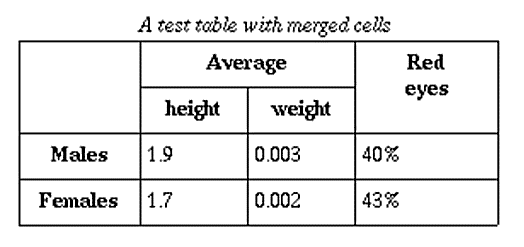
**DEPARTMENT OF INFORMATION AND COMMUNICATION ENGINEERING**

**FACULTY OF ENGINEERING AND TECHNOLOGY**

**INDEX**

|  |  |  |
| --- | --- | --- |
| **SL** | **Experiment Name** | **Page** |
| 01 | Write HTML code for following table and design it your own choice using CSS.  C:\Users\shoai\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot_4.png | 03 |
| 02 | 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 and display different order list. | 05 |
| 03 | Write HTML code for the following picture: i) Alignment text level and text box using CSS and ii) Connect this form into database using PHP.  Screenshot_12 | 08 |
| 04 | Write *JavaScript*to validate the following fields of the above registration page.  i) Name (Name should contains alphabets and the length should not be less than 6characters).  ii) Password (Password should not be less than 6 characters length).  iii) E-mail id (should not contain any invalid and must follow the standard patternname@domain.com)  iv) Phone number (Phone number should contain 10 digits only). | 11 |
| 05 | Write HTML code to create a frameset with two vertical frames: the first frame is 250 pixelswide. Fill the first frame (left \_vertical) with links of *ice.html* and *it.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* | 14 |
| 06 | Write JavaScript code using a for loop that will iterate from 0 to 30. For each iteration, it will check if the current number is odd or even and display a message on the screen. | 16 |
| 07 | Write a PHP program to calculate Electricity bill in single page for units.  **Conditions:**  For units less 50 – Taka. 3.50/unit  For units 51 to 100 – Taka. 4.00/unit  For units 101 to 200 – Taka. 5.20/unit  For units above 250 – Taka. 6.50/unit | 17 |
| 08 | Write a simple calculator program using PHP in single page.  **Operations:**  Addition  Subtraction  Multiplication  Division | 19 |
| 09 | 1. 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**) |  1. Solve the following **P** 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. | 22 |
| 10 | 1. 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 |  1. Solve the following **P** 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. | 25 |

**Experiment-01:** Write HTML code for following table and design it your own choice using CSS.



**Objective(s):**

1. To know how to construct a table using <table> tag in HTML.
2. To know about merged cells in table using HTML tag and CSS.

**Theory:**

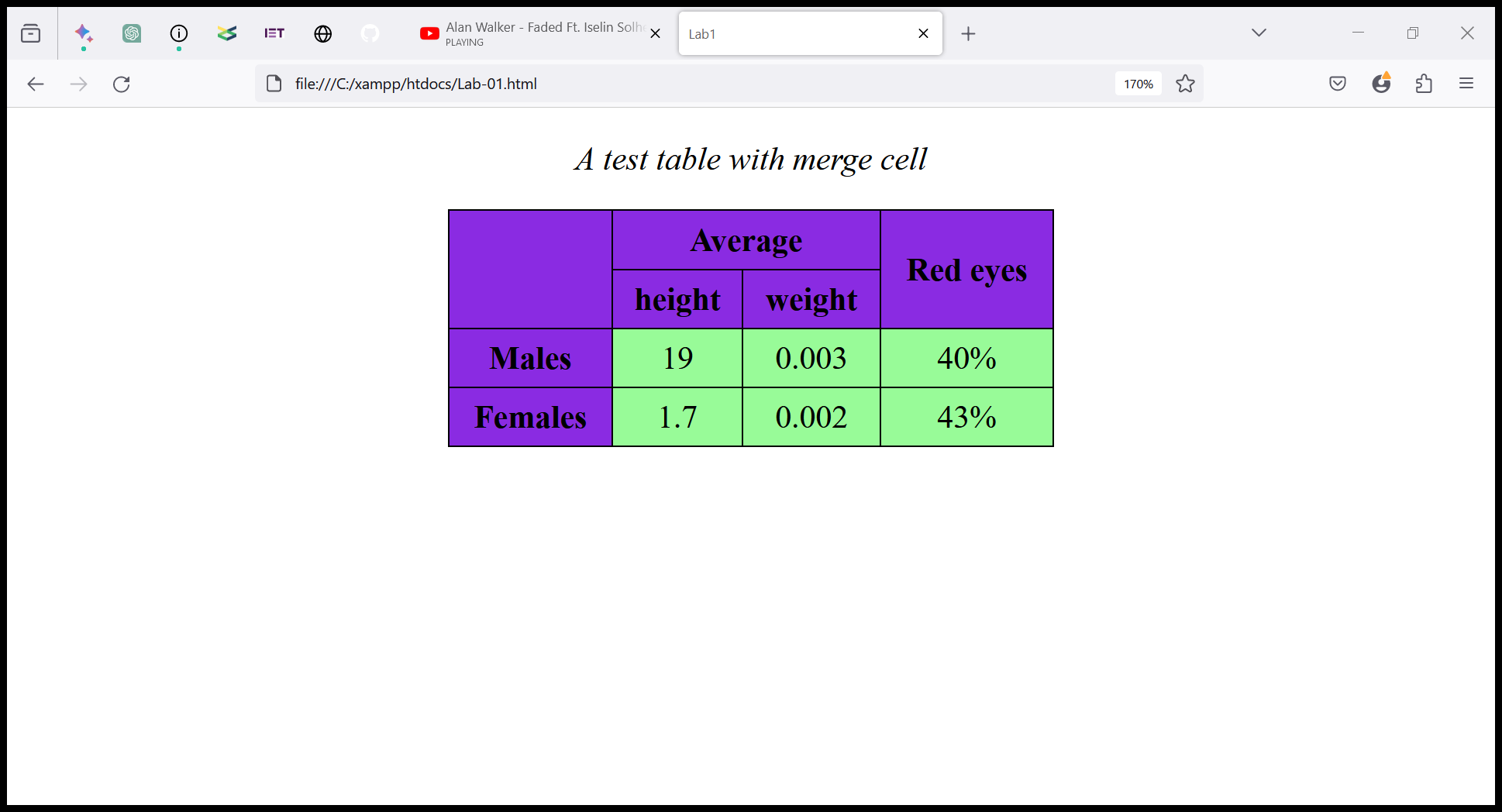
In this Experiment we have used these HTML tags and CSS given bellow:

1. The <table> element creates a table with a border, and specifies that the table should take up 100% of the width of its container and be centered using the margin and auto properties.
2. The first <tr> element within the table defines the first row of the table, which contains a single cell with a rowspan attribute of 2 to merge it with the next row. This cell is left empty, so the first column of the table is effectively merged with the second column in the next row.
3. The second <tr> element defines the second row of the table, which contains three cells with the headings "Average height", "Average weight", and "Red Eyes". The first two cells are merged using the colspan attribute.
4. The third and fourth <tr> elements define the third and fourth rows of the table, which contain data for males and females, respectively. Each row contains four cells, with the first cell displaying the gender ("Males" or "Females"), and the remaining cells displaying the average height, average weight, and red eye percentage for that gender.
5. The <th> and <td> elements arc used to define table headers and data cells, respectively. The <th> elements have a light gray background color, while the <td> elements have a white background color.
6. The CSS code within the <style> element defines styles for the various elements in the document. The border-collapse property is used to collapse the border between cells in the table, and the text-align property is used to center the text within cells. The background-color property is used to set the background color of the table headers, and the margin and auto properties are used to center the table horizontally on the page.

**Code:**

|  |  |
| --- | --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <title>Lab1</title>  <style>  table {  border-collapse: collapse;  width: 100%;  }  th, td {  border: 1px solid black;  padding: 5px;  }  th {  background-color: blueviolet;  }  td {  background-color: palegreen;  }  </style>  </head>  <body>  <p style ="text-align: center;"> <i> A test table with merge cell </i> </p>  <div style="text-align: center;"> | <table style = "width: 300px; margin: 0 auto;">  <thead>  <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>  </thead>  <tbody>  <tr>  <th>Males</th>  <td>19</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>  </tbody>  </table>  </div>  </body>  </html> |

**Output:**

****

**Experiment-02:** 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 and display different order list.

**Objective(s):**

1. To know about internal links
2. To know how internal links works to locate appropriate locations or sections
3. To know various types of lists
4. To demonstrates HTML Skills

**Theory:** In this Experiment we use:

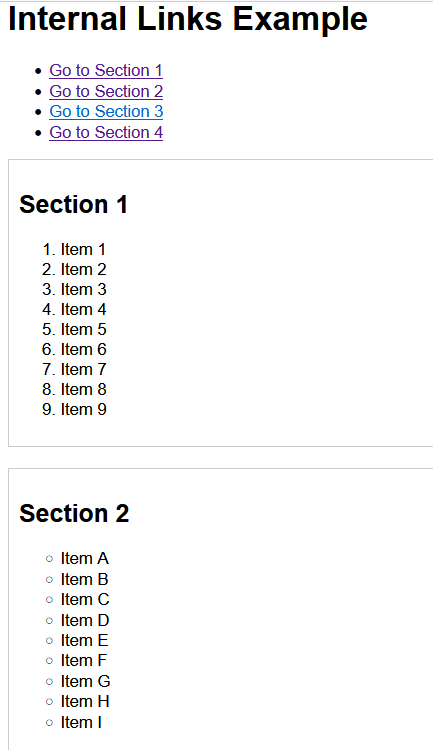
1. **Unordered list (<ul>) -** a bulleted list where each item is preceded by a bullet point. The individual items are enclosed within **<li>** tags.
2. **Ordered** **list (<ol>) -** a numbered list where each item is preceded by a number. The individual items are also enclosed within **<li>** tags.
3. **<a>** - creates a hyperlink.
4. **href -** sets the URL of the hyperlink. We have used a hashtag (#) followed by the ID of the target section to create an internal link.
5. **<ul> and <Ii> -** create an unordered list with three list items containing the links to the four sections.
6. **<hr> -** creates a horizontal rule to separate the sections.
7. **<h2> -** creates subheadings for each section.
8. **id -** sets a unique identifier for each section. We have used these identifiers as targets for the internal links.

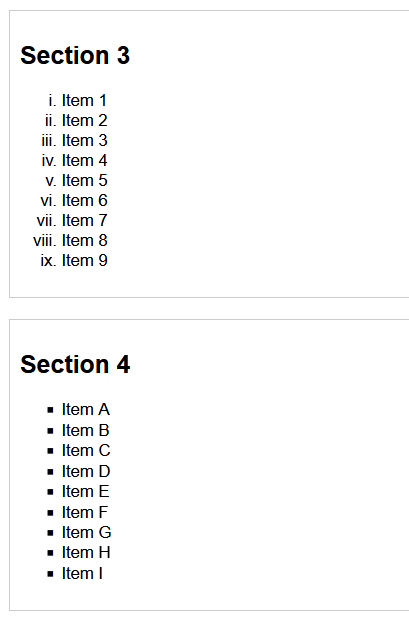
When the user clicks on one of the links in the list, the page will scroll down to the corresponding section. When the user clicks on the "Top" link, the page will scroll back up to the top of thepage. Note that we have used the same ID ("top") for the link to go back to the top and the target for the links in the list. This creates a circular navigation where **the user can jump** between sections and back to the top of the page.

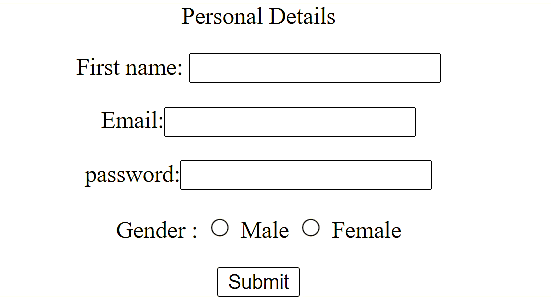
**Code:**

|  |  |
| --- | --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <title>Lab-02 Internal Links</title>  <style>  body {  font-family: Arial, sans-serif;  margin: 20px;  }  section {  margin-bottom: 20px;  padding: 10px;  border: 1px solid #ccc;  }  #back-to-top {  position: fixed;  bottom: 20px;  right: 30px;  background-color: #007bff;  color: #fff;  font-size: 24px;  cursor: pointer;  }  </style>  </head>  <body>  <button id="back-to-top" title="Go to top"> <a href = "#Top" > Top </a></button>  <h1>Internal Links Example</h1>  <ul>  <li><a href="#section1">Go to Section 1</a></li>  <li><a href="#section2">Go to Section 2</a></li>  <li><a href="#section3">Go to Section 3</a></li>  <li><a href="#section4">Go to Section 4</a></li>  </ul>  <section id="section1">  <h2>Section 1</h2>  <ol type="1">  <li>Item 1</li>  <li>Item 2</li>  <li>Item 3</li>  <li>Item 4</li>  <li>Item 5</li>  <li>Item 6</li>  <li>Item 7</li>  <li>Item 8</li>  <li>Item 9</li>  </ol>  </section> | <section id="section2">  <h2>Section 2</h2>  <ul type = "circle">  <li>Item A</li>  <li>Item B</li>  <li>Item C</li>  <li>Item D</li>  <li>Item E</li>  <li>Item F</li>  <li>Item G</li>  <li>Item H</li>  <li>Item I</li>  </ul>  </section>    <section id="section3">  <h2>Section 3</h2>  <ol type="i">  <li>Item 1</li>  <li>Item 2</li>  <li>Item 3</li>  <li>Item 4</li>  <li>Item 5</li>  <li>Item 6</li>  <li>Item 7</li>  <li>Item 8</li>  <li>Item 9</li>  </ol>  </section>  <section id="section4">  <h2>Section 4</h2>  <ul type = "square">  <li>Item A</li>  <li>Item B</li>  <li>Item C</li>  <li>Item D</li>  <li>Item E</li>  <li>Item F</li>  <li>Item G</li>  <li>Item H</li>  <li>Item I</li>  </ul>  </section>  </body>  </html> |

**Output:**

****

**Experiment-03:** Write HTML code for the following picture: i) Alignment text level and text box using CSS and ii) Connect this form into database using PHP.



**Objective(s):**

1. To know how construct a form using HTML and CSS
2. To know how design a form using CSS
3. To know about submit button
4. To know how connect form with database server using PHP

**Theory:**

1. We have used HTML and CSS code for a form that captures personal details such as name, email, phone number, password, and gender. The CSS code sets the styles for the different elements on the form, such as the text alignment, font weight, width, margin, padding, and border radius, among others.
2. The form is wrapped inside a div with a class "form", which sets the width, display, and margin of the form. The form contains input fields for capturing the user's name, email and password. The input fields have the "required" attribute to ensure that the user fills in all the necessary details before submitting the form.
3. The form also has a radio button for selecting the user's gender. The radio buttons are styled to have a smaller width and height using CSS.
4. Finally, there is a submit button at the bottom of the form.
5. In the PHP section, It establishes a connection to a MySQL database on a local server using provided credentials (server, username, password, and database).
6. It checks if the database connection was successful and outputs a corresponding message.
7. It listens for a form submission (with the 'submit' button) and, if triggered, collects user data from the form (name, email, password and gender).
8. It constructs an SQL query to insert the collected data into the 'personal\_details' table.
9. It executes the SQL query and displays a success or failure message based on the query's execution status.

In summary, this PHP code connects to a database, handles form data submission, and inserts user information into a MySQL table while providing status messages for each step.

**Code:**

|  |
| --- |
| <!doctype html>  <html>  <head>  <title>Lab-3 Registration Form</title>  <style>  body {  border: 6px solid rgb(153, 141, 141);  padding: 10px;  width: 324px;  margin: 0 auto;  }  form {  background-color: yellow;  }  .inputForm {  padding-top: 5px;  padding-bottom: 8px;  width: 98%;  }  #submit{  width = 100%;  }  </style>    </head>  <body>  <div class = "formdiv" >  <div id="form" align="center">  <form align="center" action="Lab-03.php" method="POST" onsubmit="return validateForm()">  <table>  <tr>  <td colspan = "2"> </td><h1 name="signup">Personal Details</h1> </td>  </tr>  <tr>  <td> <b> First Name: </b> </td>  <td> <input class = "inputForm" type="text" name="user" id="user"placeholder="Enter the firstname"required> </td>  </tr>  <tr>  <td> <b> Email: </b> </td>  <td> <input class = "inputForm" type="email" name="email" id="email"placeholder="Email"required> </td>  </tr>  <tr>  <td> <b> Password: </b> </td>  <td> <input class = "inputForm" type="password" name="pass" id="pass"placeholder="Set password"required> </td>  </tr>    <tr>  <td> <b>Gender: </b> </td>  <td> <input type="radio" id="gender" name="gender" value="male"> Male  <input type="radio" id="gender" name="gender" value="female"> Female </td>  </tr>  <tr>  <td colspan = "2"> <div align = "center"> <input type="submit" name="submit" id="submit" value="Submit"> </div> </td>  </tr>  </table>  </form>  </div>  </div>  <?php  $server="localhost";  $username="root";  $password="";  $db="lab3";  $conn=mysqli\_connect($server,$username,$password,$db);    if($conn){  echo "Database Connected Successfully!";  }  else{  echo "Database Have some error!";  }  if(isset($\_POST['submit'])) {  $firstname=$\_POST['user'];  $email=$\_POST['email'];  $pass=$\_POST['pass'];  $cont = $\_POST['phone'];  $gender=$\_POST['gender'];  $insertquery = "insert into personal\_details(Name,Email,Password,Contact,Gender) values('$firstname','$email','$pass', '$cont', '$gender')";  $res = mysqli\_query($conn,$insertquery);  if($res) {  echo " Data inserted Successfully!";  }  else {  echo " Data is not inserted Successfully!";  }  }  ?>  </body>    </html> |

**Output:**

****

**Experiment-04:** Write *JavaScript*to validate the following fields of the above registration page.

i) Name (Name should contains alphabets and the length should not be less than 6characters).

ii) Password (Password should not be less than 6 characters length).

iii) E-mail id (should not contain any invalid and must follow the standard patternname@domain.com)

iv) Phone number (Phone number should contain 10 digits only).

**Objective(s):**

1. To know about JavaScript function
2. To know about validation of name, email, number and password
3. To know about condition operator in JavaScript

**Theory:** This JavaScript code defines a function named validateForm() that is typically used for client-side form validation. Here's a description of what this code does:The function starts by retrieving values from different form fields by their id attributes. These fields include a "name" input, a "password" input, an "email" input, and a "phone" input.It then performs a series of validation checks on each input value:

1. **Name Validation:** It checks if the name contains only alphabetic characters and is at least 6 characters long. If the name doesn't meet these criteria, it displays an alert message and returns false, indicating that the form is not valid.
2. **Password Validation:** It checks if the password is at least 6 characters long. If not, it displays an alert message and returns false.
3. **Email Validation:** It checks if the email address matches a pattern that resembles a typical email format. If the email address doesn't match the pattern, it displays an alert message and returns false.
4. **Phone Number Validation:** It checks if the phone number contains exactly 10 digits. If not, it displays an alert message and returns false.

If all the validation checks pass, the function returns true, indicating that the form is valid. In essence, this script is meant to be used as an onsubmit handler for a form, and it checks whether the input values meet specific criteria, displaying alert messages for any validation failures. If all validations pass, the form submission is allowed.

**Code:**

|  |
| --- |
| <!doctype html>  <html>  <head>  <title>Lab-4 Registration Form</title>  <script type="text/javascript">  function validateForm() {  var name = document.getElementById('user').value;  if (!/^[a-zA-Z ]{6,}$/.test(name)) {  alert('Name should contain alphabets and be at least 6 characters long.');  return false;  }  var password = document.getElementById('pass').value;  if (password.length < 6) {  alert('Password should be at least 6 characters long.');  return false;  }  var email = document.getElementById('email').value;  if (!/^\S+@\S+\.\S{2,}$/.test(email)) {  alert('Invalid email address.');  return false;  }  var phoneNumber = document.getElementById('phone').value;  if (!/^\d{10}$/.test(phoneNumber)) {  alert('Phone number should contain 10 digits only.');  return false;  }  return true;  }  </script>  <style>  body {  border: 6px solid rgb(153, 141, 141);  padding: 10px;  width: 324px;  margin: 0 auto;  }  .inputForm {  padding-top: 5px;  padding-bottom: 8px;  width: 98%;  }  #submit{  width = 100%;  }  </style>    </head>  <body>  <div class = "formdiv" >  <div id="form" align="center">  <form align="center" action="Lab-03.php" method="POST" onsubmit="return validateForm()">  <table>  <tr>  <td colspan = "2"> </td><h1 name="signup">Personal Details</h1> </td>  </tr>  <tr>  <td> <b> First Name: </b> </td>  <td> <input class = "inputForm" type="text" name="user" id="user"placeholder="Enter the firstname"required> </td>  </tr>  <tr>  <td> <b> Email: </b> </td>  <td> <input class = "inputForm" type="email" name="email" id="email"placeholder="Email"required> </td>  </tr>  <tr>  <td> <b> Password: </b> </td>  <td> <input class = "inputForm" type="password" name="pass" id="pass"placeholder="Set password"required> </td>  </tr>  <tr>  <td> <b> Phone NO: </b> </td>  <td> <input class = "inputForm" type="text" name="phone" id="phone"placeholder="Phone number"required> </td>  </tr>  <tr>  <td> <b>Gender: </b> </td>  <td> <input type="radio" id="gender" name="gender" value="male" required> Male  <input type="radio" id="gender" name="gender" value="female"> Female  <input type="radio" id="gender" name="gender" value="other"> Other </td>  </tr>  <tr>  <td colspan = "2"> <div align = "center"> <input type="submit" name="submit" id="submit" value="Submit"> </div> </td>  </tr>  </table>  </form>  </div>  </div>  <?php  $server="localhost";  $username="root";  $password="";  $db="lab3";  $conn=mysqli\_connect($server,$username,$password,$db);    if($conn){  echo "Database Connected Successfully!";  }  else{  echo "Database Have some error!";  }  if(isset($\_POST['submit'])) {  $firstname=$\_POST['user'];  $email=$\_POST['email'];  $pass=$\_POST['pass'];  $cont = $\_POST['phone'];  $gender=$\_POST['gender'];  $insertquery = "insert into personal\_details(Name,Email,Password,Contact,Gender) values('$firstname','$email','$pass', '$cont', '$gender')";  $res = mysqli\_query($conn,$insertquery);  if($res) {  echo " Data inserted Successfully!";  }  else {  echo " Data is not inserted Successfully!";  }  }  ?>  </body>    </html> |

**Output:**

****

**Experiment-05:** Write HTML code to create a frameset with two vertical frames: the first frame is 250 pixelswide. Fill the first frame (left \_vertical) with links of *ice.html* and *it.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):**

1. To divide webpage into multiple sections
2. To display multiple web pages simultaneously

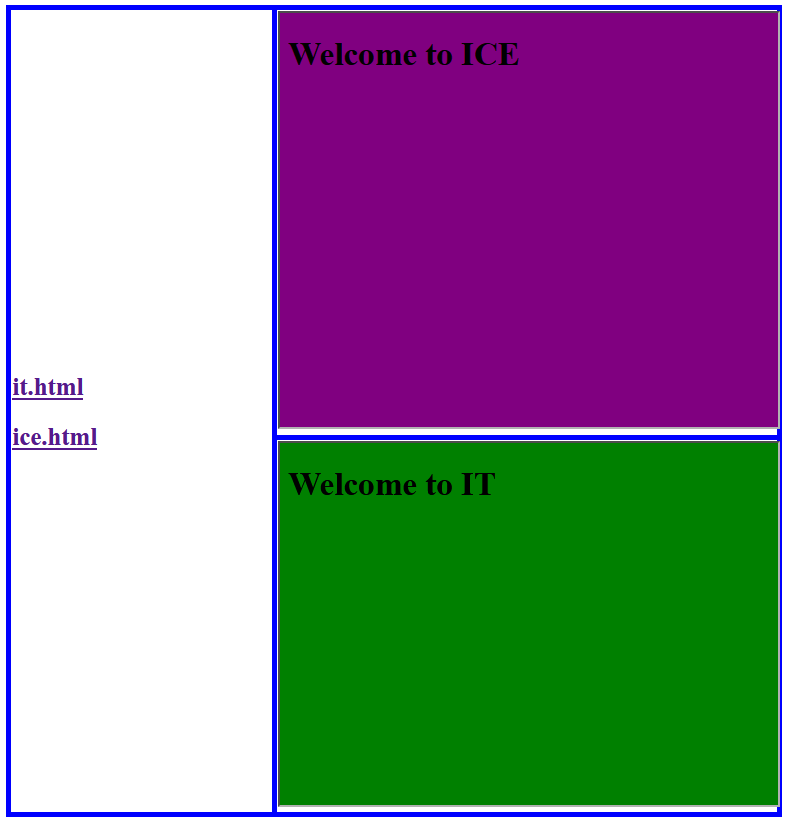
**Theory:** This HTML document represents a webpage that includes the following features:

1. **Document Declaration:** It begins with the standard HTML document declaration (<!DOCTYPE html>).
2. **Head Section:** In the <head> section, it sets the title of the webpage to "Lab-05" and defines a simple internal CSS style that specifies the appearance of the table and table cells.
3. **Body Content:** The <body> section contains the main content of the webpage.
4. **Table Layout:** Inside the body, it creates an HTML table element with a fixed width of 750 pixels (width="750px"). The border-collapse style is applied to collapse cell borders, and border style is added to table cells (<td>) to create a 5-pixel blue border around each cell.
5. **Table Rows and Cells:** The table is divided into two rows using the <tr> element, and each row contains two cells (<td>). The first cell in the first row has a rowspan attribute set to "2," which means it spans two rows.
6. **Links:** Inside the first cell of the table, there are two hyperlinks (HTML files), "it.html" and "ice.html," both wrapped in <h2> headings. These links allow users to navigate to other pages.
7. **IFrames:** Inside the second cell of the table, there are two <iframe> elements. Each iframe is used to embed content from external web pages. The "src" attribute of the iframes specifies the source HTML pages ("ice.html" and "it.html"). Iframes are set to have a height of 100% and width of 100%, and they display a message ("Your Browser Does Not Support Frame") if the browser doesn't support iframes.

**Code:**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Lab-05</title>  <style>  table{  border-collapse: collapse;  }  td{  border: 5px solid blue;  }  </style>  </head>  <body>  <table width = 750px>  <tr>  <td rowspan="2" width = 250px>  <h2> <a href = "it.html"> it.html</a> <br> </h2>  <h2> <a href = "ice.html"> ice.html </a> </h2>  </td>  <td height = 400px>  <iframe src = "ice.html" height=100% width = 100%>  Your Browser Not support Frame  </iframe>  </td>  </tr>  <tr>  <td height = 350px>  <iframe src = "it.html" height=100% width = 100%>  Your Browser Not support Frame  </iframe>  </td>  </tr>  </table>  </body>  </html> |

**Output:**

****

**Experiment-06:** Write JavaScript code using a for loop that will iterate from 0 to 30. For each iteration, it will check if the current number is odd or even and display a message on the screen.

**Objective(s):**

1. To know how using a loop and conditional operator, we can determine a number is even or odd within a range.
2. To know how a loop working in JavaScript

**Theory:**

1. It obtains a reference to an HTML element with the id "output" using document.getElementById("output"). This element will be used to display the output.
2. It enters a loop that iterates from 0 to 30 (inclusive) using a for loop.
3. Within the loop, it checks if the current value of i is even or odd using the condition if (i % 2==0).
4. If i is even, it appends the message "i is even; " to the content of the HTML element referenced by outputElement. If i is odd, it appends the message "i is odd; ".
5. The final result is that the HTML element with the id "output" will display a string of numbers from 0 to 30, with each number labeled as either "even" or "odd" based on its parity. The numbers and their labels are all concatenated together into a single string in the output element.

**Code:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="border: 2px solid black; width=device-width, initial-scale=1.0">

<title>Lab-06</title>

</head>

<body>

<div id="output" style="border: 6px solid black; background-color: #55f904; width: 400px; height: 300px; margin: 0 auto; padding: 10px; font-family: 'Times New Roman', Times, serif; color: rgb(146, 3, 255)">The result is: <br></div>

<script>

var outputElement = document.getElementById("output");

for (var i = 0; i <= 30; i++) {

if (i % 2 === 0) {

outputElement.innerHTML += i + " is even; ";

} else {

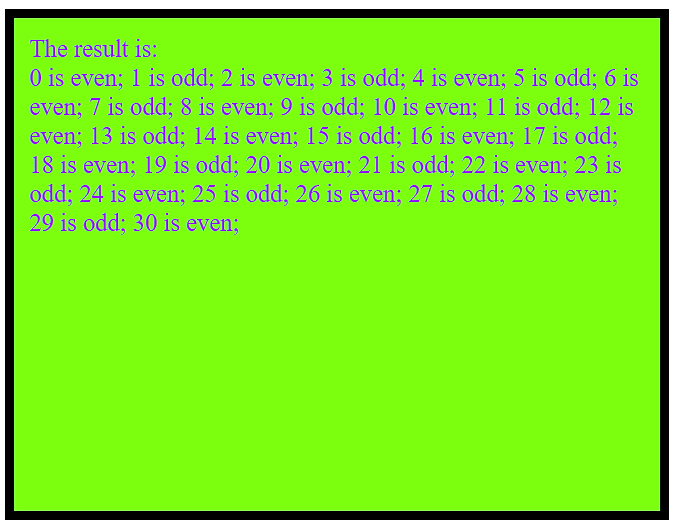
outputElement.innerHTML += i + " is odd; "; }}

</script>

</body>

</html>

**Output:**

****

**Experiment-07:** Write a PHP program to calculate Electricity bill in single page for units.

**Conditions:**

For units less 50 – Taka. 3.50/unit

For units 51 to 100 – Taka. 4.00/unit

For units 101 to 200 – Taka. 5.20/unit

For units above 250 – Taka. 6.50/unit

**Objective(s):**

1. To know how to calculate electricity bill in PHP
2. To know how to show calculated value in webpage

**Theory:** A PHP script for calculating an electricity bill based on the number of units consumed. Here's a brief description:

1. An HTML <form> is provided with an input field for users to enter the number of units consumed.
2. When the user submits the form by clicking the "Calculate" button, the form data is sent to a PHP script named "Lab-07.php" via the HTTP POST method.
3. In the PHP script, it checks if the "submit" button was pressed, indicated by the presence of the $\_POST['submit'] value, using if(isset($\_POST['submit'])).
4. It retrieves the number of units entered by the user from the form using $units = $\_POST["units"].
5. The script calculates the electricity bill according to conditions.
6. The calculated electricity bill is then displayed with a message in the HTML using echo.

**Code:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Lab-07</title>

<style>

body {

border: 6px solid rgb(153, 141, 141);

padding: 10px;

width: 430px;

margin: 0 auto;

}

</style>

</head>

<body>

<h2>Electricity Bill Calculator</h2>

<form action = "Lab-07.php" method="POST">

Enter the number of units: <input type="text" id = "units" name="units">

<input type="submit" name = "submit" id = "submit" value="Calculate">

</form>

<?php

if(isset($\_POST['submit'])) {

// Get the units from the form

$units = $\_POST["units"];

if ($units <= 50) {

$bill = $units \* 3.50;

} elseif ($units <= 100) {

$bill = 50 \* 3.50 + ($units - 50) \* 4.00;

} elseif ($units <= 200) {

$bill = 50 \* 3.50 + 50 \* 4.00 + ($units - 100) \* 5.20;

} else {

$bill = 50 \* 3.50 + 50 \* 4.00 + 100 \* 5.20 + ($units - 200) \* 6.50;

}

// Display the calculated bill

echo "Your electricity bill for $units units is Taka. " . number\_format($bill, 2);

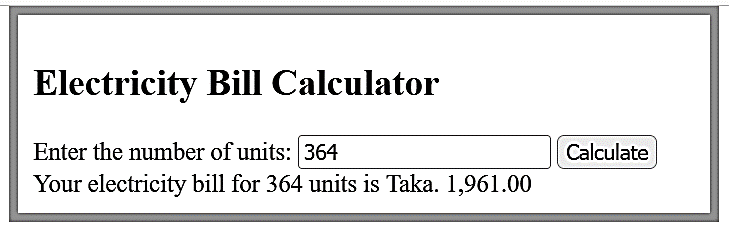
}

?>

</body>

</html>

**Output:**



**Experiment-08:** Write a simple calculator program using PHP in single page.

**Operations:**

Addition

Subtraction

Multiplication

Division

**Objectivc(s):**

1. To know how to calculate addition in PHP
2. To know how to calculate subtraction in PHP
3. To know how to calculate multiplication in PHP
4. To know how to calculate division in PHP

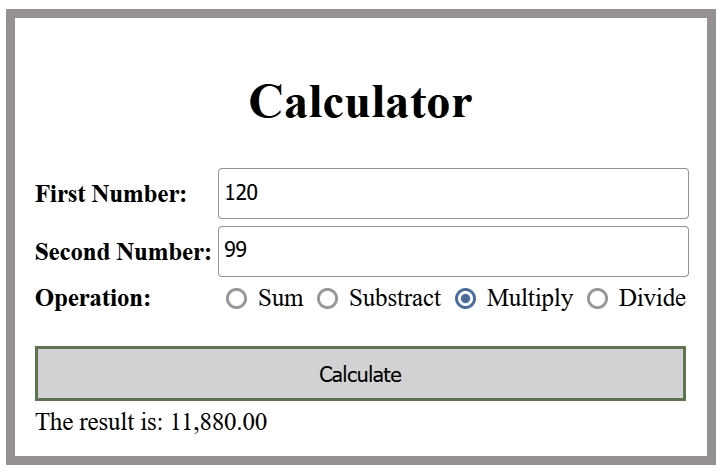
**Theory:**

1. We checks if the "submit" button was pressed, indicated by the presence of the $\_POST['submit'] value, using if(isset($\_POST['submit'])).
2. If the "submit" button was pressed, it retrieves the values of two numbers ($num1 and $num2) and an operation type ($type) from a form submitted via POST request.
3. It then performs the requested arithmetic operation based on the selected operation type:
4. If the operation type is 'Sum', it adds the two numbers and stores the result in the variable $res.
5. If the operation type is 'Subs', it subtracts the second number from the first and stores the result in $res.
6. If the operation type is 'Multi', it multiplies the two numbers and stores the result in $res.
7. If the operation type is 'Divide', it checks if the second number is zero. If it's zero, it prints "Divided by zero" to indicate division by zero. Otherwise, it performs division and stores the result in $res.

**Code:**

|  |
| --- |
| <!doctype html>  <html>  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <title>Lab-8 Form</title>  <script type="text/javascript">  function validateForm() {  // Name validation (should contain alphabets and be at least 6 characters long)  var name = document.getElementById('number1').value;  if (!/^[0-9]{1,}$/.test(name)) {  alert('Number should contains at least 1 digit');  return false;  }  var name2 = document.getElementById('number2').value;  if (!/^[0-9]{1,}$/.test(name2)) {  alert('Number should contains at least 1 digit');  return false;  }  return true;  }  </script>  <style>  #form {  border: 6px solid rgb(153, 141, 141);  padding: 10px;  width: 430px;  margin: 0 auto;  }  .inputForm {  padding-top: 5px;  padding-bottom: 8px;  width: 98%;  }  #submit {  width: 100%;  border: 2px solid #598f37;  padding: 8px 5px;  margin-top: 18px;  }  </style>    </head>  <?php  if(isset($\_POST['submit']))  {  $num1=$\_POST['number1'];  $num2=$\_POST['number2'];  $type=$\_POST['calc'];  if ($type == 'Sum') {  $res = $num1 + $num2;  } elseif ($type == 'Subs') {  $res = $num1 - $num2;  } elseif ($type == 'Multi') {  $res = $num1 \* $num2;  } else {  if ($num2 == 0) {  echo "Divided by zero";  } else {  $res = $num1 / $num2;  }  }  }  ?>  <body>  <div class = "formdiv" >  <div id="form">  <form action="" method="POST" onsubmit="return validateForm()">  <table>  <tr>  <th colspan = "2"> </td><h1 name="signup">Calculator</h1> </th>  </tr>  <tr>  <td> <b> First Number: </b> </td>  <td> <input class = "inputForm" type="text" name="number1" id="number1" required> </td>  </tr>  <tr>  <td> <b> Second Number: </b> </td>  <td> <input class = "inputForm" type="text" name="number2" id="number2" required> </td>  </tr>  <tr>  <td> <b>Operation: </b> </td>  <td> <input type="radio" id="calc" name="calc" value="Sum" required> Sum  <input type="radio" id="calc" name="calc" value="Subs"> Substract  <input type="radio" id="calc" name="calc" value="Multi"> Multiply  <input type="radio" id="calc" name="calc" value="Div"> Divide </td>  </tr>  <tr>  <td colspan = "2"> <input class="btn btn-primary" type="submit" name="submit" id="submit" value="Calculate"> </td>  </tr>  <tr>  <td colspan = "2" width = 400px> <?php echo "The result is: " . number\_format($res, 2); ?></td>  </tr>  </table>  </form>    </div>  </div>  </body>    </html> |

**Output:**

****

**Experiment-09:**

1. 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) |

1. Solve the following P 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(s):**

1. To know how update data in database using PHP
2. To know how delete data in database using PHP
3. To know how insert data in database using PHP
4. To know how create a database

**Theory:** This PHP script is designed for a student registration form with various functionalities:

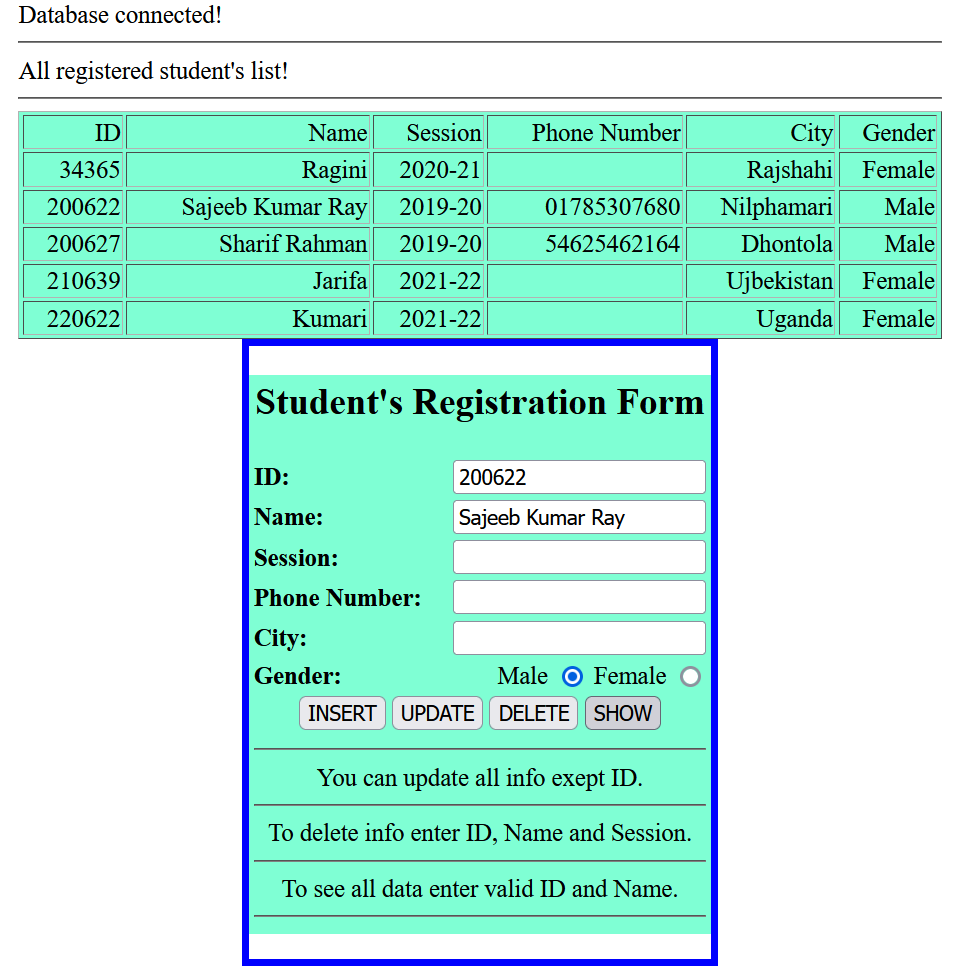
1. It establishes a connection to a MySQL database with a table named "semester\_reg" for storing student records.
2. It handles the following operations:
   1. Insert: Inserts a new student record into the database when the user submits the form.
   2. Delete: Deletes a student record from the database when the user submits the form with the appropriate ID, Name, and Session.
   3. Update: Updates an existing student record in the database when the user submits the form with the appropriate ID.
   4. Show: Retrieves and displays all registered student records from the database.
3. The form in the HTML section allows users to input student information, including ID, Name, Session, Phone Number, City, and Gender.
4. Users can perform operations such as inserting, deleting, updating, and showing records by clicking the respective buttons on the form.
5. It displays the results of these operations, such as success messages or the retrieved student records.
6. The script includes CSS styles to format the form and results.

This script provides a basic web-based interface for managing student records, including registration, deletion, updating, and viewing of records in a MySQL database.

**Code:**

|  |  |
| --- | --- |
| <?php  $conn = mysqli\_connect("localhost","root","","student1");  if($conn) {  echo "Database connected! <hr>";  }  else {  echo "Check database connection! <hr>";  }  if(isset($\_POST['insert'])){  $id = $\_POST['id'];  $name = $\_POST['name'];  $ses = $\_POST['session'];  $phone = $\_POST['phone'];  $city = $\_POST['city'];  $gender = $\_POST['gender'];  $insert = "INSERT INTO semester\_reg(ID, Name, Session, Phone\_No, City, Gender) VALUES ('$id','$name','$ses','$phone','$city','$gender')";  $result = mysqli\_query($conn, $insert);  if($result){  echo "Data inserted successfully! <hr>";  }  else {  echo "Check insertion query! <hr>";  }  }  if(isset($\_POST['update'])){  $id = $\_POST['id'];  $name = $\_POST['name'];  $ses = $\_POST['session'];  $phone = $\_POST['phone'];  $city = $\_POST['city'];  $gender = $\_POST['gender'];  $update = "UPDATE semester\_reg SET Name='$name',Session='$ses',Phone\_No='$phone',City='$city',Gender='$gender' WHERE ID='$id'";  $result = mysqli\_query($conn, $update);  if($result){  echo "Data updated successfully! <hr>";  }  else {  echo "Check update query! <hr>";  }  }  if(isset($\_POST['delete'])){  $id = $\_POST['id'];  $name = $\_POST['name'];  $ses = $\_POST['session'];  $delete = "DELETE FROM semester\_reg WHERE ID='$id' and Name='$name' and Session='$ses'";  $result = mysqli\_query($conn, $delete);  if($result){  echo "Data deleted successfully! <hr>";  }  else {  echo "Check delete query! <hr>";  }  }  if(isset($\_POST['select'])){  $id = $\_POST['id'];  $name = $\_POST['name'];  $select = "SELECT \* FROM semester\_reg";  $result = mysqli\_query($conn, $select);  if($result){  echo "All registered student's list! <hr>";  echo "<table border='1'>  <tr>  <td>ID</td>  <td>Name</td>  <td>Session</td>  <td>Phone Number</td>  <td>City</td>  <td>Gender</td>  </tr>";  if(mysqli\_num\_rows($result) ){  while($row = mysqli\_fetch\_array($result) ){  echo "<tr>  <td> ".$row['ID']." </td>  <td> ".$row['Name']." </td>  <td> ".$row['Session']." </td>  <td> ".$row['Phone\_No']."</td>  <td> ".$row['City']." </td>  <td> ".$row['Gender']." </td>  </tr>";  }  }  echo "</table>";  }  else {  echo "Check query or no data! <hr>";  }  }  ?> | <html>  <head>  <title>Lab-09</title>  <style>  body, div {  margin: 0 auto;  width: 600px;  }  div {  border: 5px solid blue;  width: 300px;  }  table {  background-color: aquamarine;  width: 100%;  margin: 0 auto;  text-align: right;  }  th {  text-align: left;  }  #mine {  text-align: center;  }  submit {  cursor: pointer;  }  </style>  </head>  <body>  <div>  <form action="Practice.php" method="POST">  <table>  <tr> <td colspan="2" id="mine"> <h2> Student's Registration Form <h2> </td></tr>  <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"> </td>  </tr>  <tr>  <th>Phone Number: </th>  <td> <input type="text" name="phone" > </td>  </tr>  <tr>  <th>City: </th>  <td> <input type="text" name="city"> </td>  </tr>  <tr>  <th>Gender: </th>  <td>  Male <input type="radio" name="gender" value="Male">  Female <input type="radio" name="gender" value="Female">  </td>  </tr>  <tr>  <td colspan="2" id="mine">  <input type="submit" name="insert" value="INSERT">  <input type="submit" name="update" value="UPDATE">  <input type="submit" name="delete" value="DELETE">  <input type="submit" name="select" value="SHOW">  </td>  </tr>  <tr>  <td colspan="2" id="mine"> <hr>  You can update all info exept ID. <hr>  To delete info enter ID, Name and Session. <hr>  To see all data enter valid ID and Name. <hr>  </td>  </tr>  <br>  </table>  </form>  </div>  </body>  </html> |

**Output:**

****

**Experiment-10:**

1. 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 |

1. Solve the following P 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

**Objective(s):**

1. To know how to create table in database using PHP
2. To know how password encrypt in database using PHP
3. To know how insert data in database using PHP

**Theory:** The PHP script creates a registration form for programmers and handles the following operations:

1. Insert: Allows users to insert new programmer records into a database. It collects information such as ID, Name, Image, and Password. The image is uploaded and stored in a folder, and the password is securely hashed (MD5) before insertion.
2. Delete: Enables users to delete programmer records from the database by providing the ID and Password. It also deletes the associated image file.
3. Show: Retrieves and displays all registered programmer records, including the ID, Name, and Image, from the database.

The HTML form allows users to input the programmer's ID, Name, Image, and Password. It includes an image preview feature when selecting an image file.

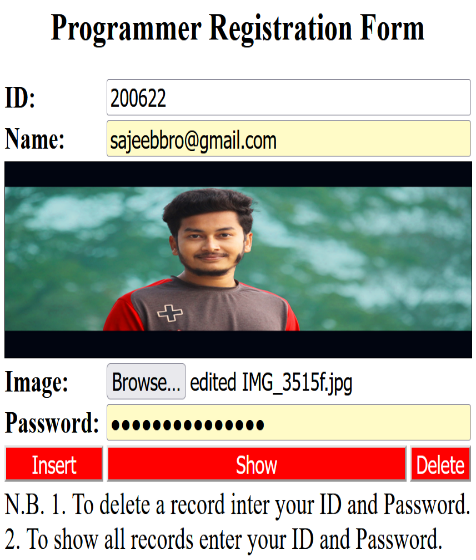
The script also includes error handling and conditional logic for each operation and displays appropriate success or failure messages.

Overall, this script provides basic functionality for registering, viewing, and deleting programmer records with secure password hashing and image handling.

**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"];  $pass = md5($password);  $insert="INSERT INTO Stu\_Reg(ID,Name,Image,Password) VALUES ('$id','$name','images/$img', '$pass')";  $result=mysqli\_query($connect,$insert);  move\_uploaded\_file($\_FILES['img']['tmp\_name'], "images/" . $\_FILES['img']['name']);  if($result==1) {  echo"Successfully insert your record!";  }  else {  echo"Unsucess";  }  } //insert End  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) {  $image=$row['Image'];  unlink("$image");  echo "Succesfully deleted your record";  }  else  {  echo "Unsucess";  }  }//delete end  if(isset($\_POST["select"])){  $query="SELECT \* FROM Stu\_Reg";  $result=mysqli\_query($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!"; }  }  ?> | <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> |

**Output:**

****