Experiment Number: 01

Name of the Experiment: To write HTML code to create a table on a Web page and design the table with CSS.

Objectives:

1. To create a table which can show via web browser.
2. To design the table with CSS.

Theory:

HTML (Hypertext Markup Language) is the code that's used to create web pages. It uses special instructions, called tags, to tell web browsers how to display content. With HTML, we can create headings, paragraphs, links, images, and more. It's like the backbone of websites, providing structure for text, images, and multimedia. When we visit a website, our browser reads the HTML and shows us the web page. It's a basic language that, when combined with other technologies like CSS and JavaScript, makes websites look good and work smoothly. HTML is the starting point for building anything we see on the internet. CSS (Cascading Style Sheets) is a design language for web pages. It styles HTML elements, controlling layout, colors, and appearance. The HTML `<table>` tag is used to create tables on web pages. It organizes data into rows and columns, with additional tags like `<tr>` for rows, `<th>` for headers, and `<td>` for data cells. It's essential for displaying tabular data in a structured format on websites.

Required Components:

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

HTML and CSS Code:

```
<html>
<head>
<style>
  table {
    border-collapse: collapse;
    width: 50%;
  }
  th, td {
    padding: 8px;
    text-align: center;
  }
  th {
    background-color: #c4c4c4;
  }
  td {
    background-color: #b7e2f5;
  }caption {
    text-align: center;
    font-weight: bold;
  }
</style>
</head>
<body>
<table border="1">
```

```
    <caption>A test table with marges cells</caption>
  <tr>
    <th rowspan="2"> </th>
    <th colspan="2" align="center">Average</th>
    <th rowspan="2" align="center">Red Eyes</th>
  </tr>
  <tr>
    <th align="center">Height</th>
    <th align="center">Weight</th>
  </tr>
  <tr>
    <td align="center">Males</td>
    <td align="center">1.9</td>
    <td align="center">0.003</td>
    <td align="center">40%</td>
  </tr>
  <tr>
    <td align="center">Females</td>
    <td align="center">1.7</td>
    <td align="center">0.002</td>
    <td align="center">43%</td>
  </tr>
</table>
</body>
</html>
```

Output:

A test table with marges cells

	Average		Red Eyes
	Height	Weight	
Males	1.9	0.003	40%
Females	1.7	0.002	43%

Results and Discussion:

We can see that a table has shown in our web browser with filled color and proper table borders. So, we can say that the code is correct.

Precautions:

While writing the HTML code we must close the tags with their closing tags and maintain sequence correctly.

Experiment Number: 02

Name of the Experiment: To 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.

Objectives:

- To create a web page with multiple sections.
- To visit multiple section of the page with the help of Internal Link.

Theory:

Internal links, also known as anchor links, bookmarks, or jump links, are a fundamental component of web navigation. They serve the purpose of guiding users within the same webpage, enhancing interactivity and improving the overall user experience. These links are invaluable for long webpages, documentation, and content-rich articles.

To create an internal link, we utilize an anchor (`<a>`) tag and set the `href` attribute to the unique ID of the section we intend to reference. This ID is typically defined with the `id` attribute within the target section. For instance, to link to a section identified as "section1," we'd construct an anchor tag with `href="#section1"`.

Internal links simplify and expedite navigation by allowing users to effortlessly jump to specific content sections. They are widely employed in table of contents, FAQs, and articles with multiple sections, enhancing the website's usability. Moreover, they reduce the need for scrolling and encourage visitors to explore content with greater ease and efficiency, ultimately contributing to a more pleasant and intuitive web experience.

Required Components:

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

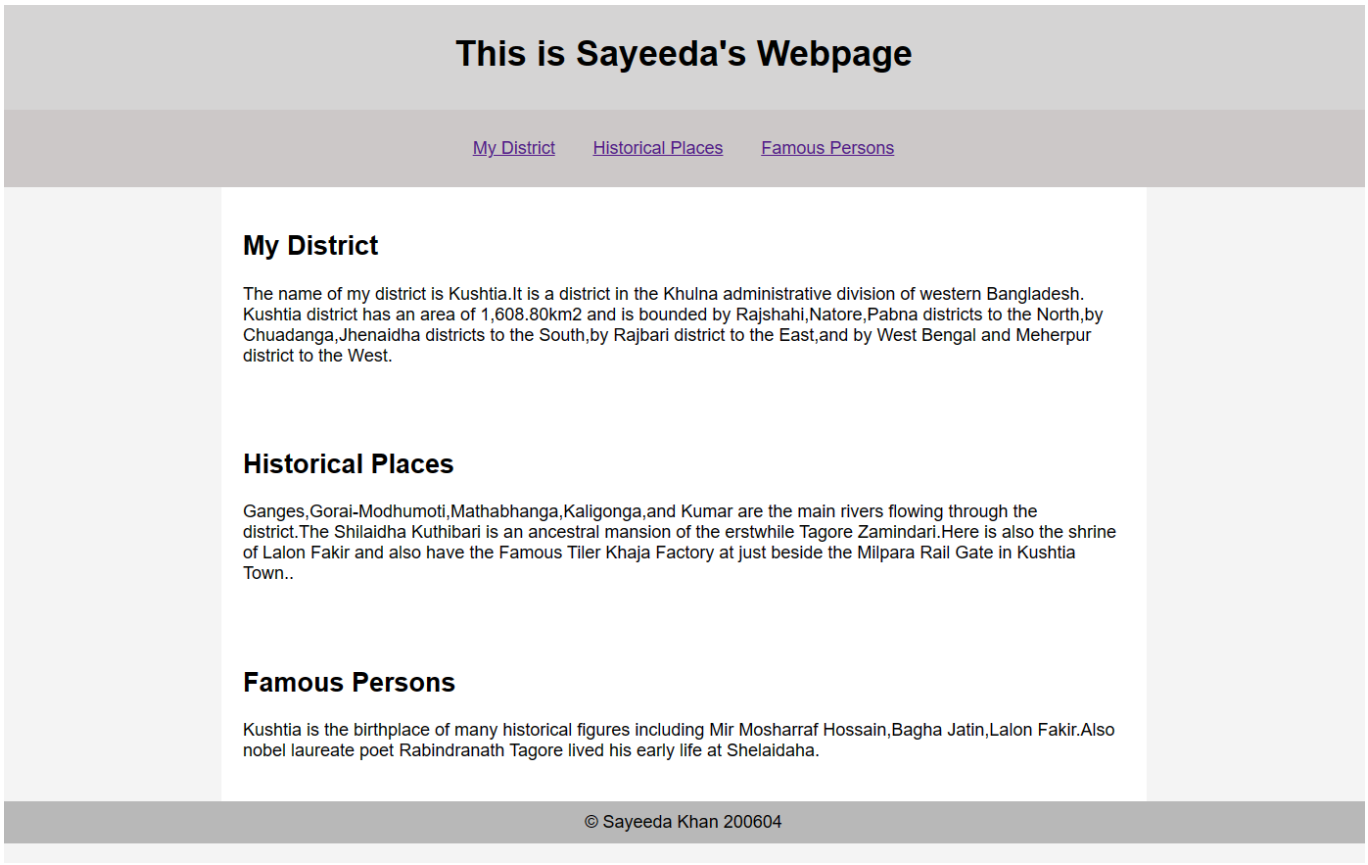
HTML and CSS Code:

```
<html >
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>My Standard Web Page</title>
    <style>
        /* CSS for styling */
        body {
            font-family: Arial, sans-serif;
            background-color: #f4f4f4;
            margin: 0;
            padding: 0;
        }
        header {
            background-color: #d5d4d4;
            color: #000000;
            text-align: center;
            padding: 10px;
```

```
        }
        nav {
            background-color: #ccc8c8;
            color: #000000;
            text-align: center;
            padding: 10px;
        }
        nav ul {
            list-style: none;
            padding: 0;
        }
        nav li {
            display: inline;
            margin: 0 15px;
        }
        section {
            max-width: 800px;
            margin: 0 auto;
            padding: 20px;
            background-color: #fff;
        }
        footer {
            background-color: #b8b8b8;
            color: #000000;
            text-align: center;
            padding: 10px;
        }
    </style>
</head>
<body>
    <header>
        <h1>This is Sayeeda's Webpage</h1>
    </header>
    <nav>
        <ul>
            <li><a href="#section1">My District</a></li>
            <li><a href="#section2">Historical Places</a></li>
            <li><a href="#section3">Famous Persons</a></li>
        </ul>
    </nav>
    <section id="section1">
        <h2>My District</h2>
        <p>About My District</p>
    </section>
    <section id="section2">
        <h2>Historical Places</h2>
        <p>Historical places of my District</p>
    </section>
    <section id="section3">
        <h2>Famous Persons</h2>
        <p>The Famous Persons</p>
    </section>
    <footer>
```

```
        &copy; Sayeeda Khan 200604
    </footer>
</body>
</html>
```

Output:



Results and Discussion:

We can see that the sections are working properly while clicking on the hyperlink this is opening the section. It works similar to a bookmark on a page.

Precautions:

While writing the HTML code we must close the tags with their closing tags and maintain sequence correctly.

Experiment Number: 03

Name of the Experiment: To write HTML code to create a form and connect the form with a database with the help of PHP.

Objectives:

1. To create a registration form using HTML and CSS
2. To connect the form to the database with the help of php.

Theory:

HTML forms are essential for user interaction on the web. They enable users to input data, make selections, and submit information. PHP, a server-side scripting language, is commonly used to process form submissions. When a user submits a form, PHP scripts can validate, sanitize, and save the data into a database. This data can be anything from user registration details to feedback forms. Databases, such as MySQL or PostgreSQL, store and manage this information efficiently. This combination of HTML forms, PHP processing, and database storage is the backbone of dynamic web applications, facilitating everything from e-commerce transactions to user registrations.

Required Components:

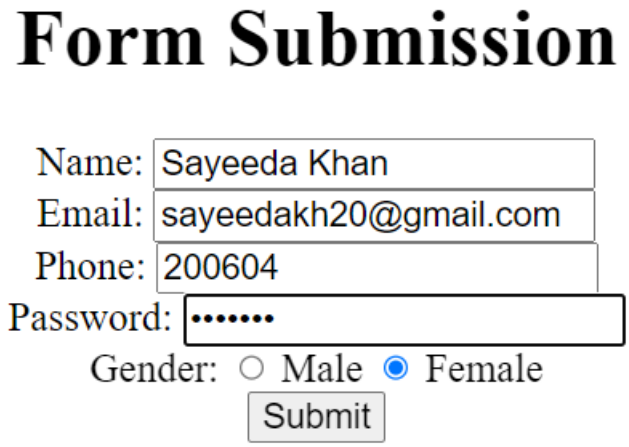
- HTML Editor (e.g., Visual Studio Code, Sublime Text, Atom, etc.)
- Web Browser (e.g., Google Chrome, Mozilla Firefox, Safari, etc.)
- XAMPP server

HTML and CSS Code:

```
<html>
<head>
    <title>Form Submission</title>
</head>
<body>

    <h1 align="center">Form Submission</h1>
    <form method="post" action="connection.php" align="center">
        <label for="name">Name:</label>
        <input type="text" id="name" name="name" 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>
        <label for="password">Password:</label>
        <input type="password" id="password" name="password" required><br>
        <label for="gender">Gender:</label>
        <input type="radio" id="male" name="gender" value="Male"> Male
        <input type="radio" id="female" name="gender" value="Female"> Female
        <br>
        <input type="submit" value="Submit">
    </form>
</body>
</html>
```

```php
<?php
$connect=mysqli_connect("localhost","root","","problem3");
    if ($_SERVER["REQUEST_METHOD"] == "POST") {
        $name = $_POST['name'];
        $email = $_POST['email'];
        $phone = $_POST['phone'];
        $password = $_POST['password'];
        $gender = $_POST['gender'];
        /*echo "Name: " . $name . "<br>";
        echo "Email: " . $email . "<br>";
        echo "Phone: " . $phone . "<br>";
        echo "Password: " . $password . "<br>";
        echo "Gender: " . $gender . "<br>";*/

    }
if($connect){
    $insert="insert into `information table` (name, email, phone, password,
gender)
     values('$name', '$email', '$phone', '$password','$gender')";
    $result=mysqli_query($connect,$insert);
    if($result){
        echo "Data Inserted Successfully";
    }
}else{
    echo "Error Occured";
}

?>
```

Output:



Results and Discussion: We can see that an Information form has been created successfully, and we can submit information in that form which is stored in the database.

Precautions: While writing the HTML code we must close the tags with their closing tags and maintain sequence correctly.

Experiment Number: 04

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

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

Objectives:

- To check that the submitted information in the form is valid.

Theory:

JavaScript is a valuable tool for implementing client-side validation in web forms, guaranteeing the entry of valid user information. For registration forms, this validation ensures that essential fields are filled correctly. Using built-in JavaScript functions, we can validate the following fields:

- Name: It's required, and it should consist of only letters and spaces. JavaScript checks for empty inputs and verifies the character content.
- Password: A non-empty field with a minimum length of 6 characters. JavaScript enforces these conditions.
- Email: The email field must not be empty and should adhere to the correct format, containing "@" and "." symbols. JavaScript examines the input against these criteria.
- Phone: A non-empty field with precisely 10 digits is necessary. JavaScript ensures that the input contains only numeric characters.

By implementing these validations, JavaScript enhances the user experience by preventing the submission of invalid data, reducing errors, and enhancing the overall functionality of web forms.

Required Components: To implement this validation, we will need:

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

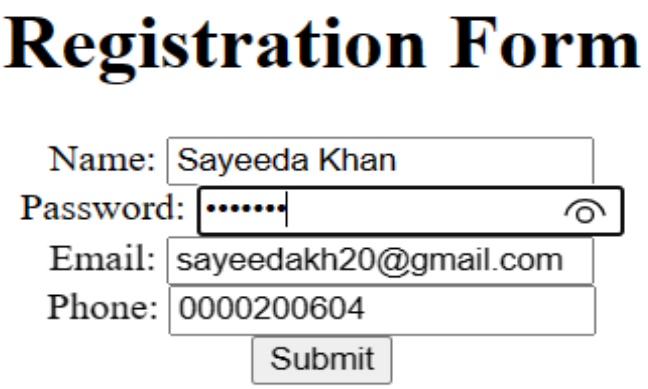
HTML, CSS and JavaScript Code:

```
<!DOCTYPE html>
<html>
<head>
    <title>Registration Form</title>
    <style>
        .error {
            border: 2px solid red;
        }
    </style>
</head>
<body>
    <h1 align="center">Registration Form</h1>
```

```
    <form align="center" id="registration-form"
method="post"  onsubmit="return validateForm()">
        <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 id="show_result"></div>
    <script>
        function validateForm() {
            var name = document.getElementById('name').value;
            var password = document.getElementById('password').value;
            var email = document.getElementById('email').value;
            var phone = document.getElementById('phone').value;

            // Name should contain alphabets and be at least 6 characters
long.
            if (!/^[a-zA-Z ]{6,}$/.test(name)) {
                alert("Invalid name. It should contain at least 6 alphabetic
characters.");
                return false;
            }

            // Password should be at least 6 characters long.
            if (password.length < 6) {
                alert("Password should be at least 6 characters long.");
                return false;
            }

            // Email validation (a basic pattern, not comprehensive).
            if (!/^[a-zA-Z0-9._-]+@[a-zA-Z0-9.-]+\.[a-zA-
Z]{2,4}$/.test(email)) {
                alert("Invalid email address.");
                return false;
            }

            // Phone number should contain exactly 10 digits.
            if (!/^\d{10}$/.test(phone)) {
                alert("Invalid phone number. It should contain 10 digits
only.");
                return false;
            }
```

```
            return true; // Form submission is allowed if all validations
pass.
        }
    </script>
</body>
</html>
```

Output:



Results and Discussion:

We can see that the Registration form works correctly, and it contains Name contain alphabets and the length is not less than 6 characters. Password is not less than 6 characters length. E-mail id follows the standard patternname@domain.com. Phone numbers contain 10 digits.

Precautions:

While writing the HTML code we must close the tags with their closing tags and maintain sequence correctly

Experiment Number: 05

Name of the Experiment:

To write HTML code to create a frameset with two vertical frames: the first frame is 250 pixels wide. Fill the first frame (left _vertical) with links of *ice.html* and *it.html.* The 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*

Objectives:

1. To create a webpage containing multiple sections.
2. To know the concept of section and how it works.

Theory:

HTML framesets were once used for creating complex layouts by dividing a webpage into multiple frames. In the provided example, a frameset is used to create a layout with two vertical frames. The left frame, 250 pixels wide, contains links to "ice.html" and "it.html." Meanwhile, the right frame is further divided into two horizontal frames, with the top frame displaying "ice.html" and the bottom frame showing "it.html." However, it's important to note that framesets have fallen out of favor in modern web development due to accessibility, SEO, and responsiveness concerns. Today, CSS and other layout techniques are preferred for creating flexible and user-friendly web designs.

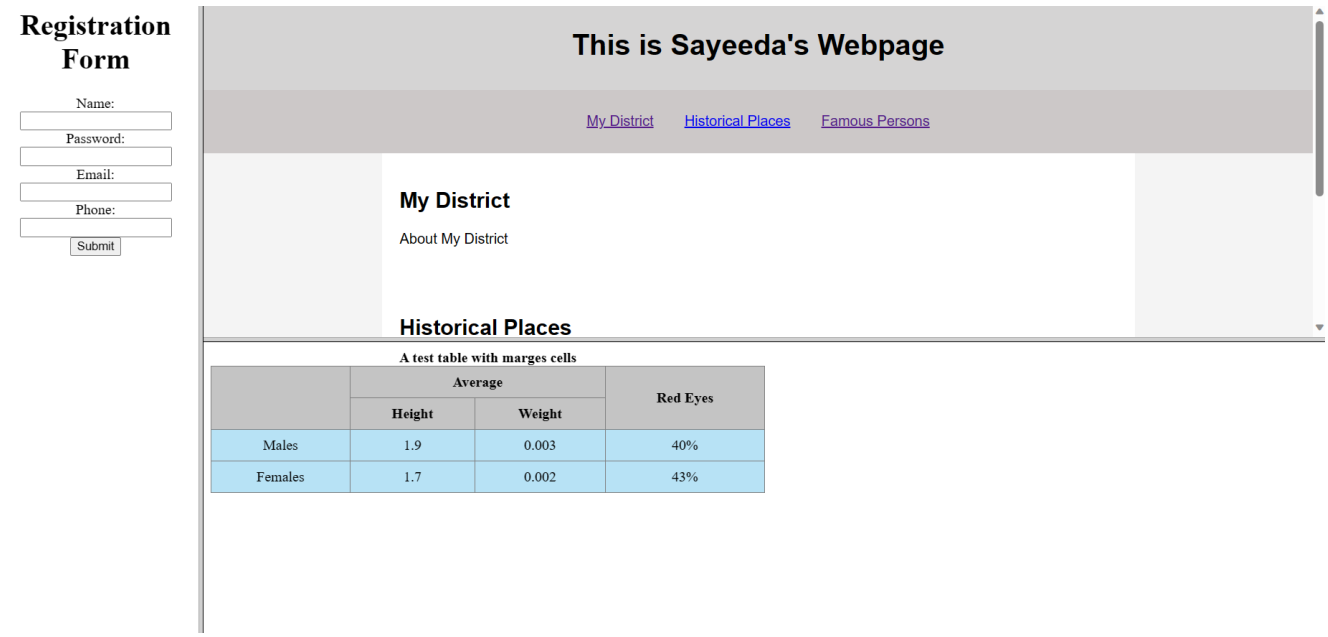
Required Components:

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

HTML and CSS Code:

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

Output:



Results and Discussion:

We can see that the sections are working properly and there are three frames in a single webpage and the sections of the right top working properly and the registration form on the left side also working. So, we can say that the code is correct, and the experiment went successfully.

Precautions:

While writing the HTML code we must close the tags with their closing tags and maintain sequence correctly

Experiment Number: 06

Name of the Experiment: To 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.

Objectives: To print the even and odd numbers using JavaScript.

Theory: JavaScript is a versatile and widely used programming language primarily employed for web development. Developed by Netscape in the mid-1990s, it's an essential component of modern web applications, enabling dynamic and interactive content. Its adaptability extends beyond the web, as it can be used in server-side development (Node.js) and even for creating desktop applications. JavaScript empowers developers to manipulate the Document Object Model (DOM), enabling real-time updates, form validation, animations, and more**.** In JavaScript, a `for` loop is used for repeating a block of code a specific number of times. It consists of three parts: initialization, condition, and increment/decrement, enclosed in parentheses and separated by semicolons.

Required Components:

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

JavaScript Code:

```
for (let i = 0; i <= 30; i++) {
  if (i % 2 === 0) {
    console.log(i + " is even.");
  } else {
    console.log(i + " is odd.");
  }
}
```

Output:

```
0 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.
16 is even.
17 is odd.
18 is even.
19 is odd.
20 is even.
21 is odd.
22 is even.
23 is odd.
24 is even.
25 is odd.
26 is even.
27 is odd.
28 is even.
29 is odd.
30 is even.
```

Experiment Number: 07

Name of the Experiment: 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

Objectives:

1. To calculate electricity bill in a webpage.
2. To know the logical and calculating part of php within a webpage.

Theory: PHP is commonly used to create dynamic web applications, and one practical use case is an electricity bill calculator. With PHP, you can take user input for the number of consumed units and apply specific rate rules to calculate the total bill. In the provided example, PHP is used to determine the cost based on different unit ranges and then display the result to the user. This showcases the versatility of PHP in processing data and presenting it in a user-friendly manner. Such PHP scripts are valuable for utility companies and users alike, simplifying the process of calculating electricity bills accurately and efficiently on web platforms.

Required Components:

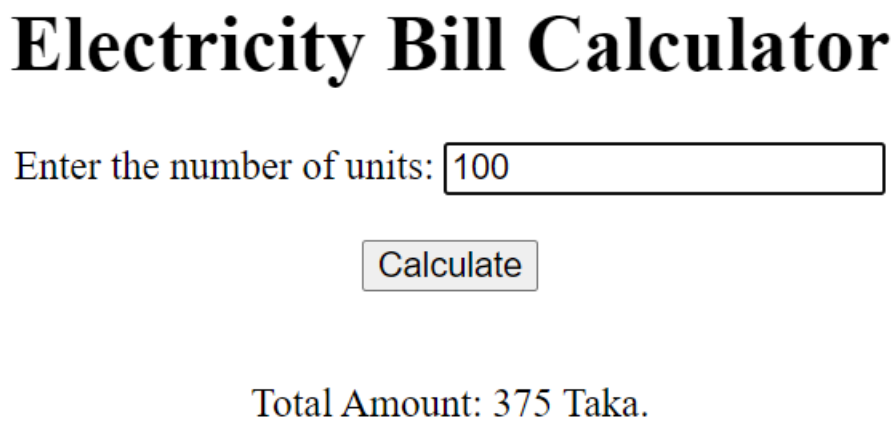
- HTML Editor (e.g., Visual Studio Code, Sublime Text, Atom, etc.)
- Web Browser (e.g., Google Chrome, Mozilla Firefox, Safari, etc.)

HTML and CSS Code:

```
<!DOCTYPE html>
<html>
<head>
    <title>Electricity Bill Calculator</title>
    <style>
        .center {
            text-align: center;
        }
    </style>
</head>
<body>
    <h1 align="Center">Electricity Bill Calculator</h1>
    <form class="center" method="post" action="<?php echo
$_SERVER['PHP_SELF']; ?>">
        Enter the number of units: <input type="text" name="units"
required><br>
        <br>
        <input type="submit" value="Calculate">
    </form>
```

```
    <?php
    if ($_SERVER["REQUEST_METHOD"] == "POST") {
        $units = $_POST['units'];
        $totalAmount = 0;
        if ($units <= 50) {
            $totalAmount = $units * 3.50;
        } elseif ($units <= 100) {
            $totalAmount = 50 * 3.50 + ($units - 50) * 4.00;
        } elseif ($units <= 200) {
            $totalAmount = 50 * 3.50 + 50 * 4.00 + ($units - 100) * 5.20;
        } else {
            $totalAmount = 50 * 3.50 + 50 * 4.00 + 100 * 5.20 + ($units -
200) * 6.50;
        }

        echo "<br><br><div class='center'>Total Amount: " . $totalAmount ."
Taka.  </div>";
    }
    ?>
</body>
</html>
```

Output:



Results and Discussion: We can see that this webpage calculates the electricity bill as per given conditions.

Precautions: While writing the HTML code we must close the tags with their closing tags and maintain sequence correctly.

Experiment Number: 08

Name of the Experiment: To write a simple calculator program using PHP on a single page. Operations:

- ❖ Addition
- ❖ Subtraction
- ❖ Multiplication
- ❖ Division

Objectives:

- To calculate Addition, Subtraction, and Multiplication Division of two numbers in a webpage.
- To know the logical and calculating part of php within a webpage.

Theory: PHP is commonly used to create dynamic web applications, and one practical use case is an electricity bill calculator. With PHP, you can take user input for the number of consumed units and apply specific rate rules to calculate the total bill. In the provided example, PHP is used to determine the cost based on different unit ranges and then display the result to the user. This showcases the versatility of PHP in processing data and presenting it in a user-friendly manner. Such PHP scripts are valuable for utility companies and users alike, simplifying the process of calculating electricity bills accurately and efficiently on web platforms.

Required Components:

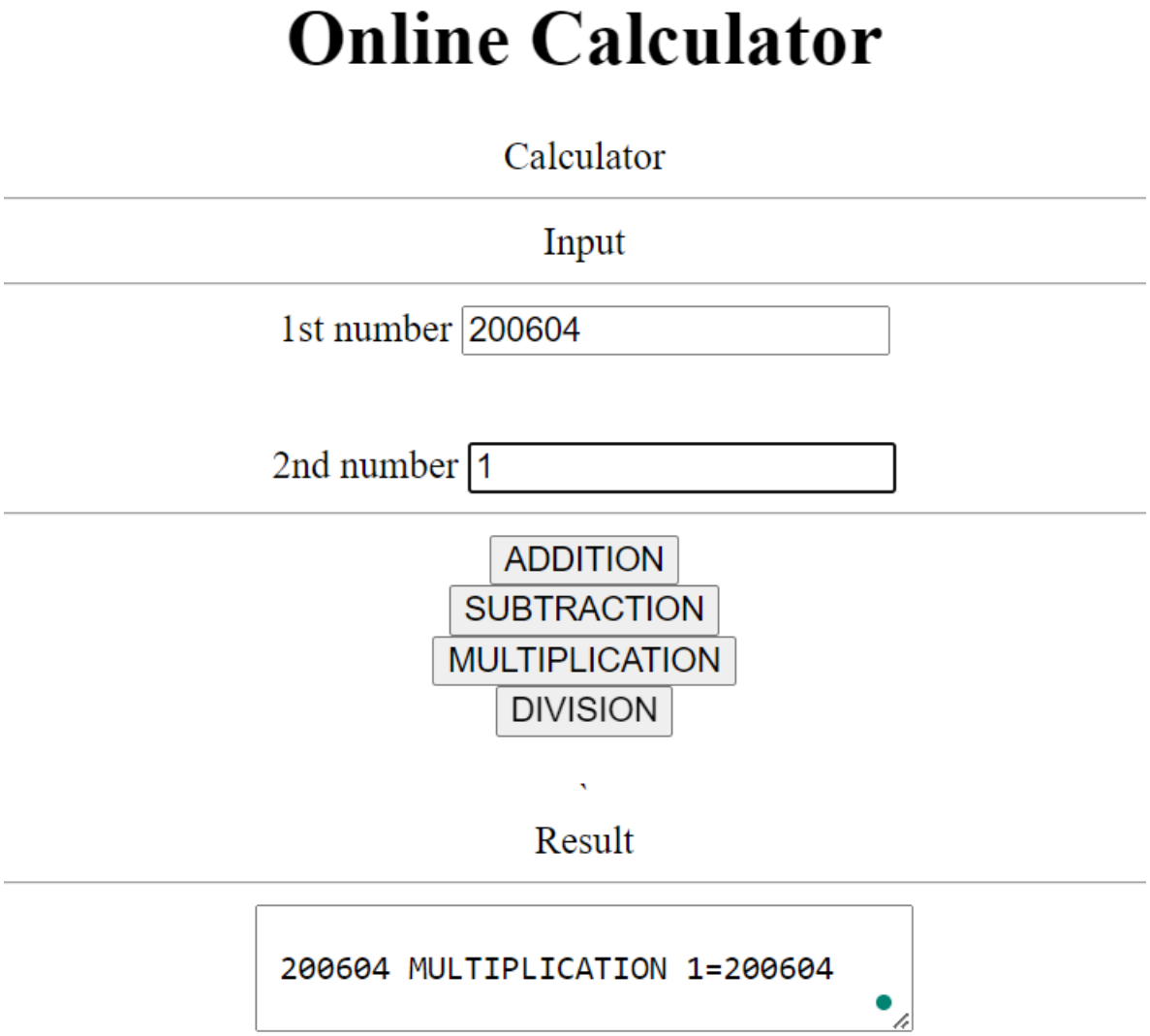
- PHP Editor (e.g., Visual Studio Code, Sublime Text, Atom, etc.)
- Web Browser (e.g., Google Chrome, Mozilla Firefox, Safari, etc.)

HTML and CSS Code:

```php
<?php
$sum=null;
$opa=null;
$x=0;
$y=0;

if(isset($_POST["ADD"]))
{
$x=$_POST['fnum'];
$y=$_POST['snum'];
$opa=$_POST["ADD"];
$sum=$x+$y;
}
else if(isset($_POST["SUB"]))
{
$x=$_POST['fnum'];
$y=$_POST['snum'];
$opa=$_POST["SUB"];
$sum=$x-$y;
}
```

```
else if(isset($_POST["MUL"]))
{
$x=$_POST['fnum'];
$y=$_POST['snum'];
$opa=$_POST["MUL"];
$sum=$x*$y;
}
else if(isset($_POST["DIV"]))
{
$x=$_POST['fnum'];
$y=$_POST['snum'];
$opa=$_POST["DIV"];
$sum=$x/$y;
$sum=number_format($sum,3);
}
?>
<html>
    <head>
        <title>This is an Online Calculator</title>
    </head>
    <body>
        <h1 align="center">Online Calculator</h1>
        <div align="center">
            Calculator
            <hr/>
        </div>
        <div align="center">
            <div align="center">
                    <form align="center" method="post" action="">
                        <label>Input </label> <hr/>
                        1st number <input type="text" name="fnum" required/><br/>
                        2nd number <input type="text" name="snum" required/><hr/>
                        <input type="submit" name="ADD" value="ADDITION"/><br/>
                        <input type="submit" name="SUB" value="SUBTRACTION"/><br/>
                        <input type="submit" name="MUL" value="MULTIPLICATION"/><br/>
                        <input type="submit" name="DIV" value="DIVISION"/><br/>
                    </form>`
            </div>
        </div>
        <div align="center">
            <label>Result </label> <hr/>
            <textarea rows="3"cols="33">
                <?php
                if($sum==null){
                    echo "$sum";
                }else{
                    echo " $x $opa $y=$sum";
                }?>
            </textarea>
        </div>
    </body>
</html>
```

Output:



Results and Discussion: We can see that this webpage calculates Addition, Subtraction, and Multiplication Division of two numbers in a webpage and generates an appropriate result.

Precautions: While writing the HTML code we must close the tags with their closing tags and maintain sequence correctly.

Experiment Number: 9

Name of the Experiment:

A. To 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. To 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 all the 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.

Objectives:

- To create a STUDENT database in XAMPP.
- To create a Semester_Reg table in that database.
- To insert, update and delete elements from the table.

Theory: MySQL is a widely used relational database management system, offering essential operations for data management. "Insert" allows adding new records to a database table, facilitating data entry. "Update" enables modifying existing data, crucial for keeping information current. "Delete" removes records, aiding data cleanup and maintenance. "Create" creates new database tables or schemas, forming the structural foundation for data storage. These operations are fundamental in database-driven applications, ensuring data integrity, accuracy, and adaptability. MySQL's ease of use and versatility make it a popular choice for handling these operations, empowering developers and businesses to manage data effectively in various applications.

Required Components: PHP Editor (e.g., Visual Studio Code, Sublime Text, Atom, etc.)**.** Web Browser (e.g., Google Chrome, Mozilla Firefox, Safari, etc.)**.** XAMPP software.

CODE:

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

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";}
}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";}
} 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";}
}if(isset($_POST["select"])){
    $query="SELECT * FROM semester_reg";
    $result=mysqli_query($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:

Student's Registration Form

ID	200604
Name	Sayeeda Khan
Session	2019-2020
Phone Number	1065369
City	Kushtia
Gender	○ Male ◉ Female

Insert Delete Update Show

N.B. 1. To delete a record please enter your ID, Name and Session.
2. You can update all information except ID Number.
3.To show all record enter your ID, Name and Session.

Experiment Number: 10

Name of the Experiment:

A. To 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. To 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 all the records of **Stu_Reg** table.
Task 5: Delete single sample record from **Stu_Reg** table using PHP program

Objectives:

- To create a **Programmer** database in XAMPP.
- To create a Stu_Reg table in that database.
- To insert, update and delete elements from the table.

Theory: MySQL is a widely used relational database management system, offering essential operations for data management. "Insert" allows adding new records to a database table, facilitating data entry. "Update" enables modifying existing data, crucial for keeping information current. "Delete" removes records, aiding data cleanup and maintenance. "Create" creates new database tables or schemas, forming the structural foundation for data storage. These operations are fundamental in database-driven applications, ensuring data integrity, accuracy, and adaptability. MySQL's ease of use and versatility make it a popular choice for handling these operations, empowering developers and businesses to manage data effectively in various applications.

Required Components: PHP Editor (e.g., Visual Studio Code, Sublime Text, Atom, etc.)**.** Web Browser (e.g., Google Chrome, Mozilla Firefox, Safari, etc.)**.** XAMPP software.

CODE:

```php
<?php
    $connect=mysqli_connect("localhost","root","","Programmer");
    if(isset($_POST["insert"])){
        $id =$_POST["id"];
        $name=$_POST["name"];e
        $img=$_FILES["img"]["name"];
        $password=$_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";
        }
    }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("$image");
        echo "Succesfully deleted your record";
        }else{
            echo "Unsucess";}
    }if(isset($_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: