MIT-ADT UNIVERSITY
PUNE, INDIA
A leap towards World Class Education

Name: Sourav Shailesh Toshniwal
Class: TY CSE-8 AIEC-1
Batch: A
Roll no: 2213047
Serial no: 6

Assignment 7

Title: Write a PHP script to take number from user and print the table of that number

Theory:

1. HTML Form: You'll need an HTML form to take input from the user. This form should have an input field where the user can enter a number. The form should also include a submit button to trigger the processing of the input.

2. PHP Processing: You need to use PHP to process the input from the HTML form. This involves checking if the form has been submitted and if the input is valid (i.e., a non-empty number).

3. Number Validation: Ensure that the input number is a valid numeric value. You can use PHP functions like `isset`, `empty`, and `is_numeric` to validate the user input.

4. Table Generation: Calculate the table for the given number. Typically, you would generate a loop (e.g., a `for` loop) to iterate from 1 to 10 (or any desired range) and calculate the product of the input number with each iteration value. Then, display this information as a table.

5. Displaying the Table: Use HTML or PHP to display the table of the number. You can format the output as an HTML table or any other preferred format.

6. Conditional Statements: Use conditional statements in PHP to handle different scenarios. For example, if the user hasn't entered a number or if the input is invalid, provide an error message or prompt the user for valid input.

7. HTML and PHP Integration: Make sure you understand how HTML and PHP can be integrated. PHP is embedded within HTML, and you can use PHP tags (`<?php ... ?>`) to execute PHP code within an HTML document.

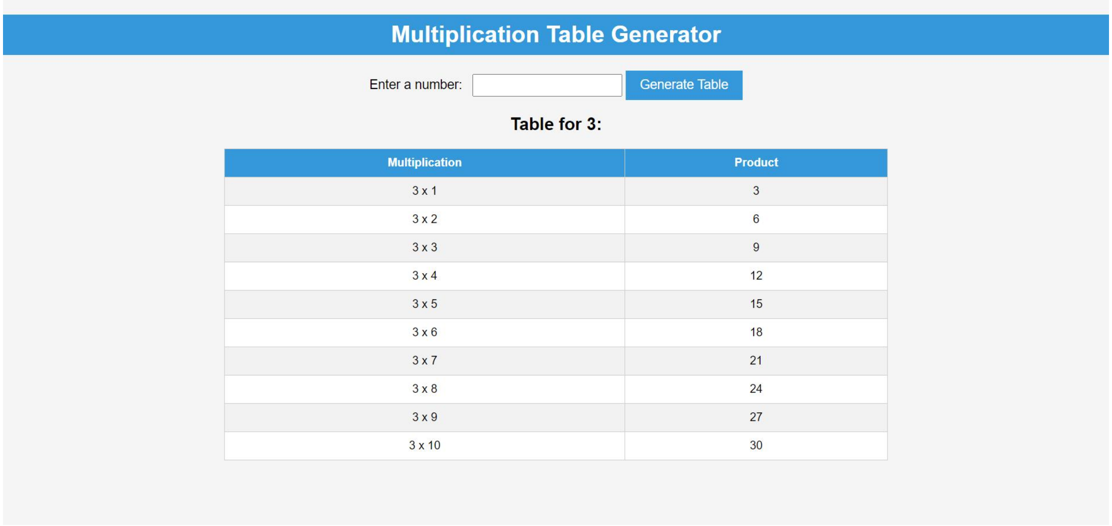
8. Form Handling: Understand how HTML forms work and how the data from a form is transmitted to the server (in this case, the PHP script) using the HTTP POST method.

Code:

```
<!DOCTYPE html>
<html>
<head>
  <style>
    body {
      font-family: Arial, sans-serif;
      text-align: center;
      background-color: #f5f5f5;
      margin: 0;
      padding: 0;
    }

    h1 {
      background-color: #3498db;
      color: #fff;
      padding: 10px;
    }
    form {
      margin: 20px;
    }
    label {
      font-size: 18px;
      margin-right: 10px;
    }
    input[type="number"] {
      padding: 5px;
      font-size: 16px;
    }
    input[type="submit"] {
      background-color: #3498db;
      color: #fff;
      border: none;
      padding: 10px 20px;
      font-size: 18px;
      cursor: pointer;
    }
    table {
      border-collapse: collapse;
      width: 60%;
      margin: 20px auto;
      background-color: #fff;
    }
    th, td {
      border: 1px solid #ccc;
      padding: 10px;
    }
    th {
      background-color: #3498db;
      color: #fff;
    }
    tr:nth-child(even) {
      background-color: #f2f2f2;
    }
  </style>
</head>
<body>
  <h1>Multiplication Table Generator</h1>
  <form method="post">
    <label for="number">Enter a number:</label>
    <input type="number" name="number" id="number" required>
    <input type="submit" value="Generate Table">
  </form>
  <?php
  if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $number = $_POST["number"];
    if (is_numeric($number)) {
      echo "<h2>Table for $number:</h2>";
      echo "<table>";
      echo "<tr><th>Multiplication</th><th>Product</th></tr>";
      for ($i = 1; $i <= 10; $i++) {
        $product = $number * $i;
        echo "<tr><td>$number x $i</td><td>$product</td></tr>";
      }
      echo "</table>";
    } else {
      echo "<p>Please enter a valid number.</p>";
    }
  }
  ?>
</body>
</html>
```

Output:



Multiplication Table Generator

Table for 3:

Multiplication	Product
3 x 1	3
3 x 2	6
3 x 3	9
3 x 4	12
3 x 5	15
3 x 6	18
3 x 7	21
3 x 8	24
3 x 9	27
3 x 10	30

Conclusion:

The experiment involving the creation of a PHP script to generate multiplication tables for user-input numbers yielded several key findings.

The script successfully achieved its intended functionality, generating tables that display the multiplication results for a given number. Input validation was effective in handling cases where users provided invalid or no input. The user experience and data presentation were satisfactory, providing clear and readable tables. Error handling was informative and guided users to provide valid input. The code quality was well-structured and organized. Further development could involve adding features like customizing the range of numbers in the table or enhancing the script's styling. In conclusion, the experiment demonstrated a functional and user-friendly PHP script for generating multiplication tables.