1a)**Develop a PHP program to calculate areas of Triangle and Rectangle**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Area Calculator</title>

</head>

<body>

<h2>Area Calculator</h2>

<form method="post">

<label for="base">Base:</label>

<input type="number" id="base" name="base" required><br><br>

<label for="height">Height:</label>

<input type="number" id="height" name="height" required><br><br>

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

</form>

<?php

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

$base = $\_POST['base'];

$height = $\_POST['height'];

$areaRectangle = $base \* $height;

$areaTriangle = ($base \* $height) / 2;

echo "<br><strong>Results:</strong><br>";

echo "The area of a rectangle with base $base and height $height is $areaRectangle square units.<br>";

echo "The area of a triangle with base $base and height $height is $areaTriangle square units.";

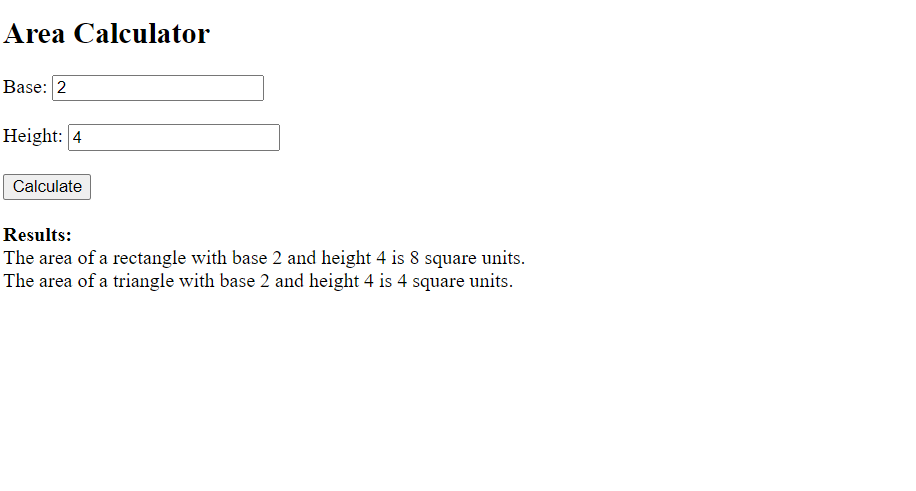
}

?>

</body>

</html>

**Output**



**1b) Develop a PHP program to calculate compound interest .**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Compound Interest Calculator</title>

</head>

<body>

<h2>Compound Interest Calculator</h2>

<form method="post">

<label for="principal">Principal Amount:</label>

<input type="number" id="principal" name="principal" required><br><br>

<label for="rate">Annual Interest Rate (%):</label>

<input type="number" id="rate" name="rate" step="0.01" required><br><br>

<label for="time">Number of Years:</label>

<input type="number" id="time" name="time" required><br><br>

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

</form>

<?php

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

$principal = $\_POST['principal'];

$rate = $\_POST['rate'] / 100; // Convert percentage to decimal

$time = $\_POST['time'];

$compound\_per\_year = 1;

// Calculate compound interest

$compound\_interest = $principal \* (1 + ($rate / $compound\_per\_year)) \*\* ($compound\_per\_year \* $time) - $principal;

echo "<br><strong>Results:</strong><br>";

echo "Principal Amount: $principal<br>";

echo "Annual Interest Rate: $rate<br>";

echo "Number of Years: $time<br>";

echo "Compound Interest: $compound\_interest";

}

?>

</body>

</html>

**Output**



1. **Demonstrating the various forms to concatenate multiple strings Develop program(s) to demonstrate concatenation of strings:**
2. **Strings represented with literals (single quote or double quote).**
3. **Strings as variables.**
4. **Multiple strings represented with literals (single quote or double quote) and variables.**
5. **Strings and string variables containing single quotes as part string contents.**
6. **Strings containing HTML segments having elements with attributes.**
7. **Strings represented with literals (single quote or double quote).**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>String Concatenation</title>

</head>

<body>

<?php

// Concatenation using literals

$string1 = 'Hello, ';

$string2 = "world!";

$result = $string1 . $string2;

echo $result;

?>

</body>

</html>

**Output:** Hello, world!

1. **Strings as variables.**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>String Concatenation</title>

</head>

<body>

<?php

// Concatenation using variables

$name = "John";

$greeting = "Hello, " . $name . "!";

echo $greeting;

?>

</body>

</html>

**Output:** Hello, John!

**(iii)Multiple strings represented with literals (single quote or double quote) and variables.**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>String Concatenation</title>

</head>

<body>

<?php

// Concatenation using literals and variables

$firstName = 'John';

$lastName = "Doe";

$greeting1 = "Hello, " . $firstName . ' ' . $lastName . "!";

echo $greeting1;

?>

</body>

</html>

**Output:** Hello, John Doe!

**(iv)Strings and string variables containing single quotes as part string contents.**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>String Concatenation</title>

</head>

<body>

<?php

// Concatenation with strings containing single quotes

$singleQuotedString = 'It\'s a sunny day.';

$variableWithSingleQuotes = "I said, '$singleQuotedString'";

echo $variableWithSingleQuotes;

?>

</body>

</html>

**Output:** I said, 'It's a sunny day.'

**(v)Strings containing HTML segments having elements with attributes.**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>String Concatenation</title>

</head>

<body>

<?php

// Concatenation with HTML segments

$element = 'div';

$className = 'my-class';

$content = 'This is PHP LAB.';

$htmlString = '<' . $element . ' class="' . $className . '">' . $content . '</' . $element . '>';

echo $htmlString;

?>

</body>

</html>

**Output:** This is PHP LAB.

1. **Develop a PHP Program(s) to check given number is:**
2. **Odd or even**
3. **Divisible by a given number (N)**
4. **Square of a another number**

<!DOCTYPE html>

<html>

<head>

<title>Number Checker</title>

</head>

<body>

<h2>Number Checker</h2>

<form method="post" action="">

Enter a number: <input type="text" name="number" required><br>

Enter a divisor (N): <input type="text" name="divisor" required><br>

Enter a base number: <input type="text" name="baseNumber" required><br>

<input type="submit" value="Check">

</form>

<?php

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

**// Get user input**

$givenNumber = $\_POST["number"];

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

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

**// Function to check if a number is odd or even**

function checkOddEven($number) {

return ($number % 2 == 0) ? "even" : "odd";

}

**// Function to check if a number is divisible by N**

function checkDivisibility($number, $divisor) {

return ($number % $divisor == 0) ? "divisible by $divisor" : "not divisible by $divisor";

}

**// Function to check if a number is the square of another number**

function checkSquare($number, $base) {

$square = $base \* $base;

return ($number == $square) ? "is the square of $base" : "is not the square of $base";

}

// Output the results

echo "Given number $givenNumber is " . checkOddEven($givenNumber) . ".<br>";

echo "It is " . checkDivisibility($givenNumber, $divisor) . ".<br>";

echo "And it " . checkSquare($givenNumber, $baseNumber) . ".<br>";

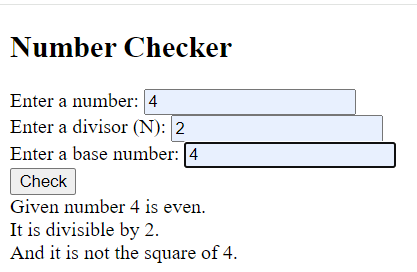
}

?>

</body>

</html>

**Output:**



**3b. Develop a PHP Program to compute the roots of a quadratic equation by accepting the coefficients. Print the appropriate messages.**

<!DOCTYPE html>

<html>

<head>

<title>Quadratic Equation Solver</title>

</head>

<body>

<h2>Quadratic Equation Solver</h2>

<form method="post" action="">

Enter coefficient a: <input type="text" name="a" required><br>

Enter coefficient b: <input type="text" name="b" required><br>

Enter coefficient c: <input type="text" name="c" required><br>

<input type="submit" value="Solve">

</form>

<?php

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

// Get user input

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

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

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

// Function to calculate the discriminant

function calculateDiscriminant($a, $b, $c) {

return $b \* $b - 4 \* $a \* $c;

}

// Function to calculate the roots

function calculateRoots($a, $b, $c) {

$discriminant = calculateDiscriminant($a, $b, $c);

if ($discriminant > 0) {

$root1 = (-$b + sqrt($discriminant)) / (2 \* $a);

$root2 = (-$b - sqrt($discriminant)) / (2 \* $a);

return "Root 1: $root1, Root 2: $root2";

} elseif ($discriminant == 0) {

$root = -$b / (2 \* $a);

return "Root: $root (double root)";

} else {

return "No real roots (complex roots)";

}

}

// Output the results

echo "Quadratic equation: $a\*x^2 + $b\*x + $c = 0<br>";

$roots = calculateRoots($a, $b, $c);

echo "Roots: $roots";

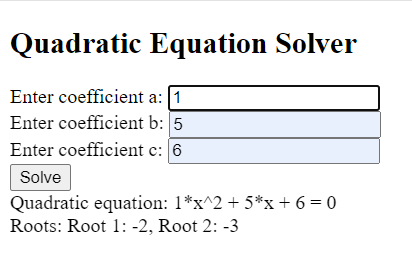
}

?>

</body>

</html>

**Output:**



**4a. Develop a PHP program to find the square root of a number by using the newton’s algorithm.**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Square Root Calculator</title>

</head>

<body>

<h2>Square Root Calculator</h2>

<form action="" method="post">

<label for="number">Enter a number:</label>

<input type="number" step="any" name="number" id="number" required>

<button type="submit">Calculate Square Root</button>

</form>

<?php

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

$number = isset($\_POST["number"]) ? $\_POST["number"] : null;

if (!is\_numeric($number)) {

echo "<p>Please enter a valid number.</p>";

} else {

$result = squareRoot($number);

echo "<p>The square root of $number is approximately $result.</p>";

}

}

function squareRoot($number) {

// Newton's algorithm for square root

$guess = $number / 2;

$epsilon = 1e-10; // small value for precision

while (abs($guess \* $guess - $number) > $epsilon) {

$guess = 0.5 \* ($guess + $number / $guess);

}

return $guess;

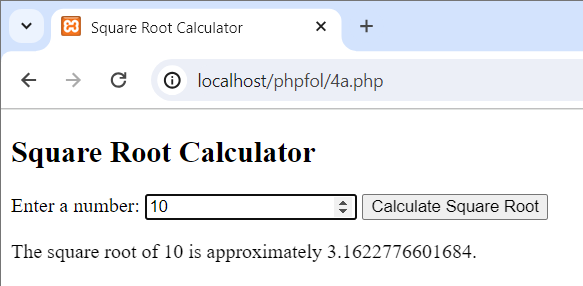
}

?>

</body>

</html>

**Output**



**4b. Develop a PHP program to generate Floyd’s triangle.**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Floyd's Triangle Generator</title>

</head>

<body>

<h2>Floyd's Triangle</h2>

<form action="" method="post">

<label for="rows">Enter the number of rows:</label>

<input type="number" name="rows" id="rows" min="1" required>

<button type="submit">Generate Floyd's Triangle</button>

</form>

<?php

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

$rows = isset($\_POST["rows"]) ? $\_POST["rows"] : null;

if (!is\_numeric($rows) || $rows <= 0) {

echo "<p>Please enter a valid positive integer for the number of rows.</p>";

} else {

generateFloydsTriangle($rows);

}

}

function generateFloydsTriangle($rows) {

$number = 1;

echo "<p>Floyd's Triangle:</p>";

for ($i = 1; $i <= $rows; $i++) {

for ($j = 1; $j <= $i; $j++) {

echo $number . " ";

$number++;

}

echo "<br>";

}

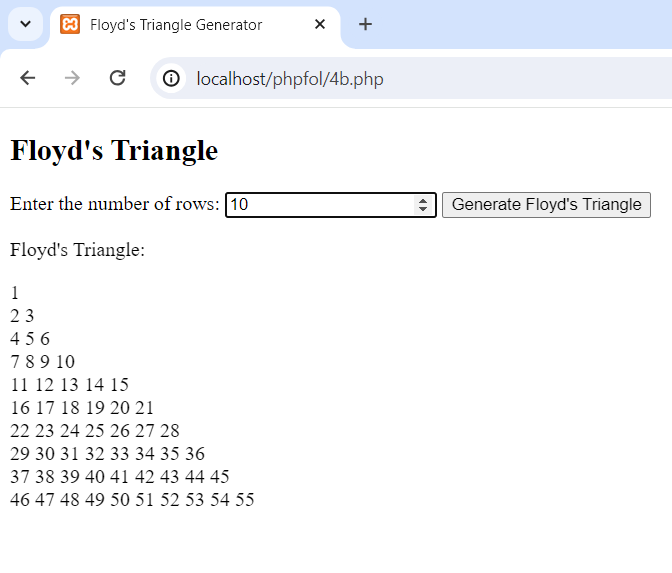
}

?>

</body>

</html>

**Output**



**5a. Develop a PHP application that reads a list of numbers and calculates mean and standard deviation.**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Mean and Standard Deviation Calculator</title>

</head>

<body>

<h2>Mean and Standard Deviation Calculator</h2>

<form action="" method="post">

<label for="numbers">Enter a list of numbers (comma-separated):</label>

<input type="text" name="numbers" id="numbers" required>

<button type="submit">Calculate</button>

</form>

<?php

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

$numbersInput = isset($\_POST["numbers"]) ? $\_POST["numbers"] : null;

if (empty($numbersInput)) {

echo "<p>Please enter a list of numbers.</p>";

} else {

$numbers = array\_map('floatval', explode(',', $numbersInput));

if (count($numbers) < 2) {

echo "<p>Please enter at least two numbers for calculation.</p>";

} else {

$mean = calculateMean($numbers);

$stdDev = calculateStandardDeviation($numbers);

echo "<p>Mean: $mean</p>";

echo "<p>Standard Deviation: $stdDev</p>";

}

}

}

function calculateMean($numbers) {

$sum = array\_sum($numbers);

$count = count($numbers);

return $count > 0 ? $sum / $count : 0;

}

function calculateStandardDeviation($numbers) {

$mean = calculateMean($numbers);

$variance = 0;

foreach ($numbers as $number) {

$variance += pow($number - $mean, 2);

}

$variance /= count($numbers);

return sqrt($variance);

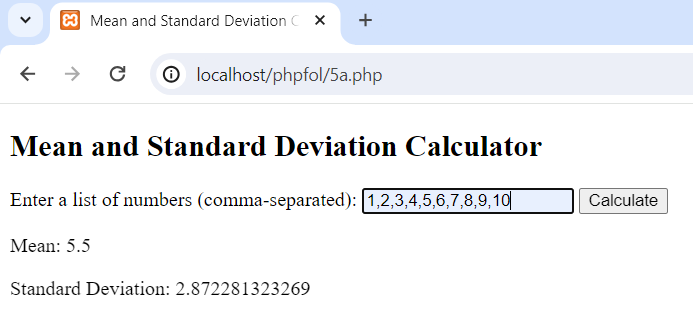
}

?>

</body>

</html>

**Output**



**5b. Develop a PHP application that reads scores between 0 and 100 (possibly including both 0 and 100) and creates a histogram array whose elements contain the number of scores between 0 and 9, 10 and 19, etc. The last “box” in the histogram should include scores between 90 and 100. Use a function to generate the histogram.**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Score Histogram Generator</title>

</head>

<body>

<h2>Score Histogram Generator</h2>

<form action="" method="post">

<label for="scores">Enter scores (comma-separated between 0 and 100):</label>

<input type="text" name="scores" id="scores" required>

<button type="submit">Generate Histogram</button>

</form>

<?php

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

$scoresInput = isset($\_POST["scores"]) ? $\_POST["scores"] : null;

if (empty($scoresInput)) {

echo "<p>Please enter scores.</p>";

} else {

$scores = array\_map('intval', explode(',', $scoresInput));

if (validateScores($scores)) {

$histogram = generateHistogram($scores);

echo "<h3>Score Histogram</h3>";

echo "<pre>";

print\_r($histogram);

echo "</pre>";

} else {

echo "<p>Please enter valid scores between 0 and 100.</p>";

}

}

}

function validateScores($scores) {

foreach ($scores as $score) {

if (!is\_numeric($score) || $score < 0 || $score > 100) {

return false;

}

}

return true;

}

function generateHistogram($scores) {

$histogram = array\_fill(0, 11, 0); // Initialize histogram array with 11 elements

foreach ($scores as $score) {

$index = floor($score / 10); // Determine the index for the histogram array

$histogram[$index]++;

}

return $histogram;

}

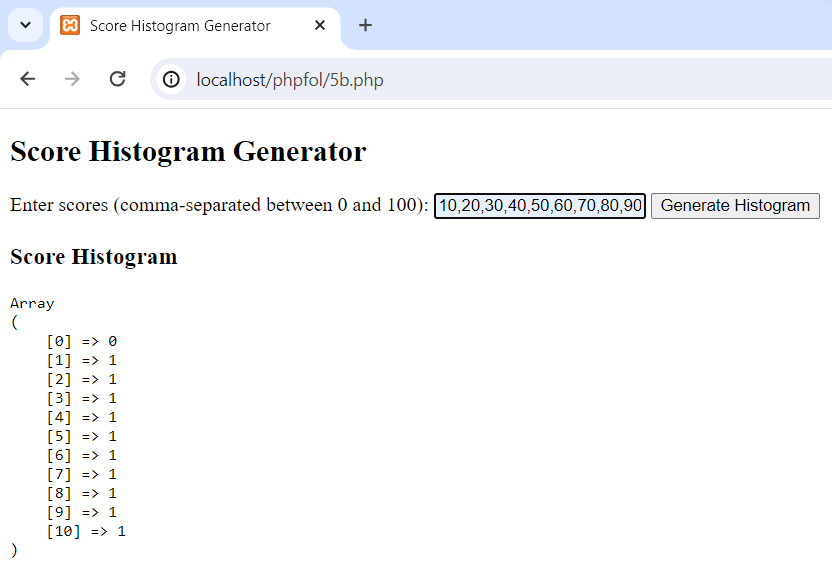
?>

</body>

</html>

This HTML file includes a form with an input field to enter scores (comma-separated) between 0 and 100. When the form is submitted, the PHP code checks if the input is valid, generates a histogram based on the specified ranges, and displays the results on the page. The validateScores function ensures that the entered scores are valid, and the generateHistogram function creates the histogram array.

**Output**



**6a. Develop PHP program to demonstrate the date() with different parameter options.**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Date() Function Demo</title>

</head>

<body>

<h1>Date() Function Demo</h1>

<p>Current Date and Time (Indian Standard Time):</p>

<?php

// Set timezone to Indian Standard Time

date\_default\_timezone\_set('Asia/Kolkata');

?>

<p><?php echo date('Y-m-d H:i:s'); ?></p>

<h2>Formatting Options:</h2>

<ul>

<li>Year-Month-Day: <?php echo date('Y-m-d'); ?></li>

<li>Month-Day-Year: <?php echo date('m-d-Y'); ?></li>

<li>Day-Month-Year: <?php echo date('d-m-Y'); ?></li>

<li>Hour:Minute:Second: <?php echo date('H:i:s'); ?></li>

<li>Day of the Week: <?php echo date('l'); ?></li>

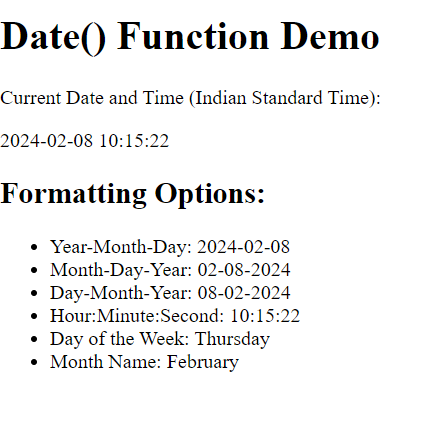
<li>Month Name: <?php echo date('F'); ?></li>

</ul>

</body>

</html>

Output:



**6.b Develop a PHP program to generate the Fibonacci series using a recursive function.**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Fibonacci Series</title>

</head>

<body>

<h1>Fibonacci Series Generator</h1>

<form method="post">

<label for="terms">Enter the number of terms:</label>

<input type="number" id="terms" name="terms">

<button type="submit">Generate Fibonacci Series</button>

</form>

<?php

// Recursive function to generate Fibonacci series

function fibonacci($n) {

if ($n <= 1) {

return $n;

} else {

return (fibonacci($n - 1) + fibonacci($n - 2));

}

}

// Check if the form is submitted

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

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

echo "<h2>Fibonacci Series:</h2>";

echo "<p>";

for ($i = 0; $i < $terms; $i++) {

echo fibonacci($i);

echo ($i < $terms - 1) ? ", " : ""; // Add comma if not the last term

}

echo "</p>";

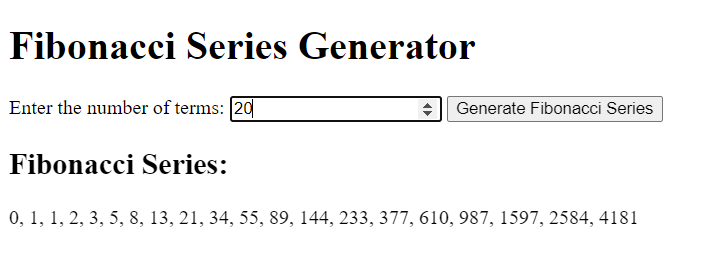
}

?>

</body>

</html>

Output:



**9 Develop a PHP program to filter the elements of an array with key names.**

**Sample Input Data:**

**1st array: ('c1' => 'Red', 'c2' => 'Green', 'c3' => 'White', c4 => 'Black')**

**2nd array: ('c2', 'c4')**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Array Key Filter</title>

</head>

<body>

<h1>Array Key Filter</h1>

<!-- Form to input data -->

<form method="post">

<label for="array1">Enter the 1st array:</label><br>

<input type="text" id="array1" name="array1" placeholder="Enter the 1st array (key-value pairs separated by commas)" style="width: 400px;"><br><br>

<label for="array2">Enter the 2nd array:</label><br>

<input type="text" id="array2" name="array2" placeholder="Enter the 2nd array (comma-separated keys)" style="width: 400px;"><br><br>

<button type="submit" name="submit">Filter Array</button>

</form>

<?php

// Check if form is submitted

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

// Get input arrays from form

$array1\_input = $\_POST['array1'];

$array2\_input = $\_POST['array2'];

// Convert input strings to arrays

$array1 = explode(',', $array1\_input);

$array2 = explode(',', $array2\_input);

// Convert key-value pairs to associative array

$data = [];

foreach ($array1 as $item) {

$key\_value = explode('=>', $item);

$key = trim($key\_value[0]);

$value = trim($key\_value[1]);

$data[$key] = $value;

}

// Filter elements based on keys in array2

$filtered\_array = array\_filter($data, function ($key) use ($array2) {

return in\_array(trim($key), $array2);

}, ARRAY\_FILTER\_USE\_KEY);

// Display the filtered array

echo "<h2>Filtered Array:</h2>";

echo "<pre>";

print\_r($filtered\_array);

echo "</pre>";

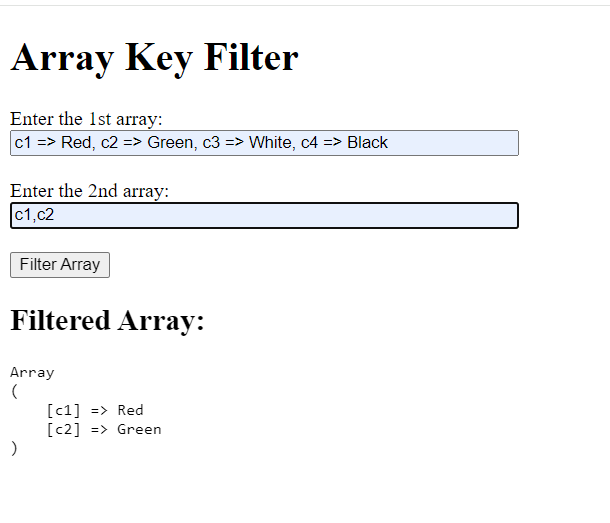
}

?>

</body>

</html>

Output:



**10 . Develop a PHP program that illustrates the concept of classes and objects by reading and printing employee data, including Emp\_Name, Emp\_ID, Emp\_Dept, Emp\_Salary, and Emp\_DOJ.**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Employee Data</title>

</head>

<body>

<h1>Employee Data</h1>

<!-- Form to input employee data -->

<form method="post">

<label for="emp\_name">Employee Name:</label><br>

<input type="text" id="emp\_name" name="emp\_name"><br><br>

<label for="emp\_id">Employee ID:</label><br>

<input type="text" id="emp\_id" name="emp\_id"><br><br>

<label for="emp\_dept">Employee Department:</label><br>

<input type="text" id="emp\_dept" name="emp\_dept"><br><br>

<label for="emp\_salary">Employee Salary:</label><br>

<input type="text" id="emp\_salary" name="emp\_salary"><br><br>

<label for="emp\_doj">Date of Joining:</label><br>

<input type="date" id="emp\_doj" name="emp\_doj"><br><br>

<button type="submit" name="submit">Submit</button>

</form>

<?php

// Define Employee class

class Employee {

public $name;

public $id;

public $dept;

public $salary;

public $doj;

// Constructor to initialize employee data

public function \_\_construct($name, $id, $dept, $salary, $doj) {

$this->name = $name;

$this->id = $id;

$this->dept = $dept;

$this->salary = $salary;

$this->doj = $doj;

}

// Method to display employee data

public function displayData() {

echo "<h2>Employee Details</h2>";

echo "<p>Name: $this->name</p>";

echo "<p>ID: $this->id</p>";

echo "<p>Department: $this->dept</p>";

echo "<p>Salary: $this->salary</p>";

echo "<p>Date of Joining: $this->doj</p>";

}

}

// Check if form is submitted

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

// Retrieve input data from form

$name = $\_POST['emp\_name'];

$id = $\_POST['emp\_id'];

$dept = $\_POST['emp\_dept'];

$salary = $\_POST['emp\_salary'];

$doj = $\_POST['emp\_doj'];

// Create an instance of Employee class

$employee = new Employee($name, $id, $dept, $salary, $doj);

// Display employee data

$employee->displayData();

}

?>

</body>

</html>

Output:



1. **a. a. Develop a PHP program to count the occurrences of Aadhaar numbers present in a text.**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Aadhaar Number Occurrence Counter</title>

</head>

<body>

<h1>Aadhaar Number Occurrence Counter</h1>

<!-- Form to input text -->

<form method="post">

<label for="text">Enter text:</label><br>

<textarea id="text" name="text" rows="4" cols="50"></textarea><br><br>

<button type="submit" name="submit">Count Occurrences</button>

</form>

<?php

// Function to count occurrences of Aadhaar numbers in text

function countAadhaarOccurrences($text) {

// Regular expression to match Aadhaar numbers

$pattern = '/\b[0-9]{4}\s[0-9]{4}\s[0-9]{4}\b/';

// Count occurrences of Aadhaar numbers

preg\_match\_all($pattern, $text, $matches);

// Return count of occurrences

return count($matches[0]);

}

// Check if form is submitted

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

// Retrieve input text from form

$text = $\_POST['text'];

// Count occurrences of Aadhaar numbers in text

$occurrences = countAadhaarOccurrences($text);

// Display the count of occurrences

echo "<h2>Occurrences of Aadhaar Numbers:</h2>";

echo "<p>$occurrences</p>";

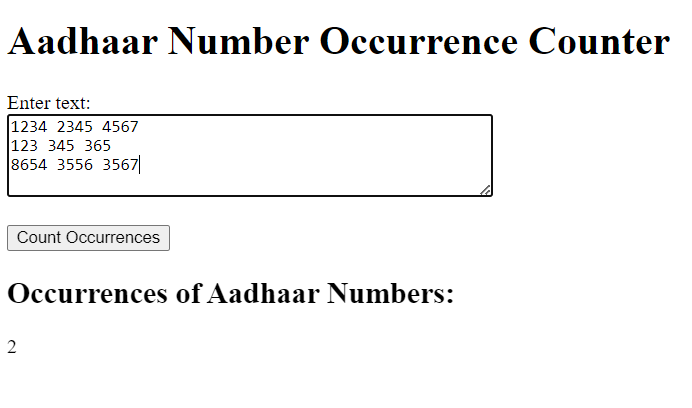
}

?>

</body>

</html>

Output:



**11** **b. Develop a PHP program to find the occurrences of a given pattern and replace them with a text.**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Pattern Occurrence Replacer</title>

</head>

<body>

<h1>Pattern Occurrence Replacer</h1>

<!-- Form to input text and pattern -->

<form method="post">

<label for="text">Enter text:</label><br>

<textarea id="text" name="text" rows="4" cols="50"></textarea><br><br>

<label for="pattern">Enter pattern:</label><br>

<input type="text" id="pattern" name="pattern"><br><br>

<label for="replacement">Enter replacement text:</label><br>

<input type="text" id="replacement" name="replacement"><br><br>

<button type="submit" name="submit">Replace Occurrences</button>

</form>

<?php

// Function to find occurrences of a pattern and replace them with a text

function replaceOccurrences($text, $pattern, $replacement) {

// Perform the replacement

$result = preg\_replace($pattern, $replacement, $text, -1, $count);

// Return the result and the count of replacements

return array($result, $count);

}

// Check if form is submitted

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

// Retrieve input data from form

$text = $\_POST['text'];

$pattern = '/' . preg\_quote($\_POST['pattern'], '/') . '/'; // Convert pattern to regex

$replacement = $\_POST['replacement'];

// Perform the replacement

list($result, $count) = replaceOccurrences($text, $pattern, $replacement);

// Display the result and count of replacements

echo "<h2>Replaced Text:</h2>";

echo "<p>$result</p>";

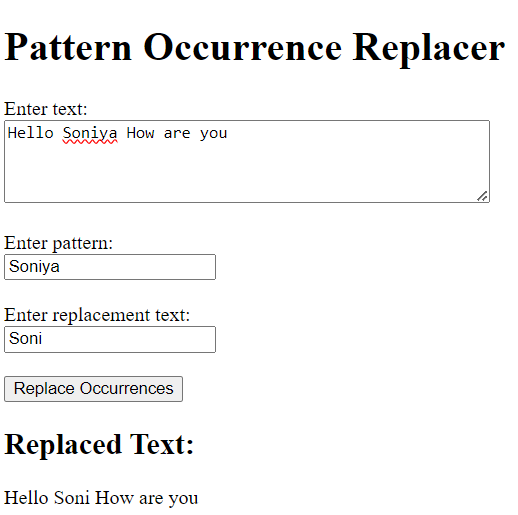
}

?>

</body>

</html>

Output:



1. **Develop a PHP program to read the contents of a HTML form and display the contents on a browser.**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Form Content Display</title>

</head>

<body>

<h1>Form Content Display</h1>

<!-- Form to input data -->

<form method="post">

<label for="name">Enter your name:</label><br>

<input type="text" id="name" name="name"><br><br>

<label for="email">Enter your email:</label><br>

<input type="email" id="email" name="email"><br><br>

<button type="submit" name="submit">Submit</button>

</form>

<?php

// Check if form is submitted

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

// Retrieve input data from form

$name = $\_POST['name'];

$email = $\_POST['email'];

// Display the input data

echo "<h2>Submitted Information:</h2>";

echo "<p>Name: $name</p>";

echo "<p>Email: $email</p>";

}

?>

</body>

</html>

Output:

