**MULTIPURPOSE CALCULATOR APPLICATION**

**An Overview of the Program**

The web-based multipurpose calculator application will enable users to carry out a range of mathematical computations, such as:

* Basic multiplication, division, addition, and subtraction
* Exponentiation
* Calculations of percentages
* Computations of square roots
* Computations using logarithms

**Advantages**

1. Versatility: Able to do a broad range of computations, from simple arithmetic to complex mathematical procedures.
2. User-Friendly Interface: An easy-to-use web form that allows for simple input and results presentation for all users.
3. Calculations in Real Time: quick response on computations without requiring page loads.
4. Extensible: It is simple to add new functions.

**Disadvantages**

1. Server Dependency: Needs a web server capable of supporting PHP.
2. Restricted to Single-User Sessions: The basic solution is incapable of supporting the administration of multiple users' sessions.
3. Security: To stop injection attacks, input must be verified.

**Implementation Approach**

1. HTML Form: Create a web form to capture user input.
2. PHP Script: Develop PHP scripts to handle various mathematical operations.
3. Result Display: Display results dynamically on the web page.

**First, create the HTML form.**  
First, we'll create an HTML form that asks customers to choose an operation and enter two integers. The data entered into the form will be sent to a PHP script for processing.

**HTML CODE FORM**

**<!DOCTYPE html>**

**<html lang="en">**

**<head>**

**<meta charset="UTF-8">**

**<title>Multipurpose Calculator</title>**

**<style>**

**body { font-family: Arial, sans-serif; }**

**.calculator { max-width: 400px; margin: 50px auto; padding: 20px; border: 1px solid #ccc; border-radius: 10px; }**

**.input-group { margin-bottom: 15px; }**

**.input-group label { display: block; margin-bottom: 5px; }**

**.input-group input { width: 100%; padding: 8px; box-sizing: border-box; }**

**.input-group select { width: 100%; padding: 8px; box-sizing: border-box; }**

**.result { margin-top: 20px; padding: 10px; border: 1px solid #ccc; border-radius: 5px; background-color: #f9f9f9; }**

**</style>**

**</head>**

**<body>**

**<div class="calculator">**

**<h2>Multipurpose Calculator</h2>**

**<form method="post" action="calculator.php">**

**<div class="input-group">**

**<label for="num1">Number 1:</label>**

**<input type="number" step="any" id="num1" name="num1" required>**

**</div>**

**<div class="input-group">**

**<label for="num2">Number 2:</label>**

**<input type="number" step="any" id="num2" name="num2" required>**

**</div>**

**<div class="input-group">**

**<label for="operation">Operation:</label>**

**<select id="operation" name="operation" required>**

**<option value="add">Addition</option>**

**<option value="subtract">Subtraction</option>**

**<option value="multiply">Multiplication</option>**

**<option value="divide">Division</option>**

**<option value="exponentiate">Exponentiation</option>**

**<option value="percentage">Percentage</option>**

**<option value="sqrt">Square Root</option>**

**<option value="logarithm">Logarithm</option>**

**</select>**

**</div>**

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

**</form>**

**<?php if (isset($\_GET['result'])): ?>**

**<div class="result">**

**<strong>Result:</strong> <?= htmlspecialchars($\_GET['result']) ?>**

**</div>**

**<?php endif; ?>**

**</div>**

**</body>**

**</html>**

**Step Two Developing the PHP Script**   
In order to process the form input and carry out the chosen procedure, we will now develop the calculator.php script.

**PHP SCRIPT**

**<?php**

**function add($a, $b) {**

**return $a + $b;**

**}**

**function subtract($a, $b) {**

**return $a - $b;**

**}**

**function multiply($a, $b) {**

**return $a \* $b;**

**}**

**function divide($a, $b) {**

**if ($b == 0) {**

**return "Division by zero error!";**

**}**

**return $a / $b;**

**}**

**function exponentiate($a, $b) {**

**return pow($a, $b);**

**}**

**function percentage($a, $b) {**

**return ($a / $b) \* 100;**

**}**

**function sqrt\_calc($a) {**

**return sqrt($a);**

**}**

**function logarithm($a) {**

**return log($a);**

**}**

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

**$num1 = isset($\_POST['num1']) ? floatval($\_POST['num1']) : 0;**

**$num2 = isset($\_POST['num2']) ? floatval($\_POST['num2']) : 0;**

**$operation = isset($\_POST['operation']) ? $\_POST['operation'] : '';**

**$result = '';**

**switch ($operation) {**

**case 'add':**

**$result = add($num1, $num2);**

**break;**

**case 'subtract':**

**$result = subtract($num1, $num2);**

**break;**

**case 'multiply':**

**$result = multiply($num1, $num2);**

**break;**

**case 'divide':**

**$result = divide($num1, $num2);**

**break;**

**case 'exponentiate':**

**$result = exponentiate($num1, $num2);**

**break;**

**case 'percentage':**

**$result = percentage($num1, $num2);**

**break;**

**case 'sqrt':**

**$result = sqrt\_calc($num1);**

**break;**

**case 'logarithm':**

**$result = logarithm($num1);**

**break;**

**default:**

**$result = 'Invalid operation selected';**

**break;**

**}**

**header("Location: index.php?result=$result");**

**exit();**

**}**

**?>**

**Guidelines for Testing the Application:  
  
Setting Up the Environment**1. Make sure your web server supports PHP (such as Apache or Nginx with PHP-FPM).  
2. Put the PHP script (calculator.php) and HTML file (index.html) in the root directory of the web server.  
3. Open the Application:

**Access the Application:**  
  
1. Launch a web browser and go to the index.html file's hosted URL.

**Perform Calculations:**  
  
1. Fill up the input fields with numbers.  
2. From the dropdown menu, choose the desired procedure.  
3. To view the outcome, use the "Calculate" button.

**Step 3: Examining the Program**  
Take these actions to test the application:  
Make a calculator-themed folder on your web server.  
Within the calculator folder, save the HTML form code in a file called index.php.  
Within the calculator folder, save the PHP script code in a file called calculator.php.  
Launch a web browser and go to the folder (http://localhost/calculator, for example).  
For the calculator to be tested, enter numbers and choose an operation.

**The HTML Form for Documentation (index.php)**- Provides a user interface to input two numbers and select the desired mathematical operation.  
- Sends data to the calculator.php PHP script using the POST protocol.  
- Shows the outcome, if it is available.

**Data from the HTML form is received by the calculator.php PHP script.**  
- It uses the chosen operation to validate and process the input.  
- It carries out logarithmic, square root, exponentiation, multiplication, division, addition, and subtraction operations.  
- It returns to the form after redirecting.

**Comprehensive Documentation Features Employed**:

* Add($a, $b): Adds two numbers.
* Subtract($a, $b): Subtracts the second number from the first.
* Multiply($a, $b):- Multiplies two numbers.
* Divide($a, $b):- Divides the first number by the second (handles division by zero).
* Exponentiate($a, $b):- Raises the first number to the power of the second.
* Percentage($a, $b):- Calculates the percentage of the first number relative to the second.
* Sqrt\_calc($a):- Calculates the square root of a number.
* Logarithm($a):- Calculates the natural logarithm of a number.

**Notes on Implementation**  
- Basic error instances like division by zero and incorrect operations are handled by the calculator.  
- Based on user input, it determines the right operation using a straightforward switch statement.  
- For display, the outcome is sent back to the form via a URL parameter.

**Error Handling**

Basic error handling for division by zero and incorrect operations is included in the application. It is possible to enhance security and user experience by adding more validation.

This solution offers a **strong framework for a versatile calculator**, **application base for a PHP multifunctional calculator** application that may be expanded upon as needed to include more functionality and enhance error handling.