# Documentation for Multipurpose Calculator

## Introduction

The Multipurpose Calculator is a web-based application that performs various mathematical operations, such as addition, subtraction, multiplication, division, exponentiation, percentage calculations, square root computation, and logarithms. The user-friendly application offers additional features such as calculation history and memory functions.

## Features Overview

1. **Basic Mathematical Operations:** Addition, subtraction, multiplication, and division.
2. **Advanced Operations:** Exponentiation, percentage calculation, square root, and logarithms.
3. **Session-Based Memory:** Stores the last result and supports recalling and clearing stored results.
4. **Calculation History:** Tracks and displays previous calculations for user reference.
5. **Interactive UI:** A modern, intuitive interface with form elements and styled components.

## Application Structure

The application consists of the following components:

### 1. index.php

This is the application's main entry point. It includes an HTML form where users can input numbers, choose operations, and submit their requests for calculation.

**Key Features:**

- Number input fields for num1 and num2
- Dropdown selection for mathematical operations
- Buttons for submission and memory functions
- Display of calculation results and history

### 2. calculate.php

This file handles the backend logic for performing calculations based on the user input from index.php.

**Key Functionalities:**

- Retrieval and validation of user input
- Calculation logic implemented using a switch statement for different operations
- Handling errors such as division by zero and invalid logarithms
- Storing the result in the session memory
- Maintaining a calculation history

### 3. memory.php

This file manages session-based memory operations.

**Key Functionalities:**

- Recall the last calculation result from session memory
- Clear the stored result
- Redirect back to the main interface after each action

## User Interface (UI) Design

The UI is designed with simplicity and usability in mind.

**Design Elements:**

- **Typography:** Clean font selection using Arial
- **Color Scheme:** Soft background with contrasting button colours for better user interaction
- **Layout:** Centered container with rounded corners and padding
- **Form Elements:** Intuitive input fields and dropdowns
- **Interactive Elements:** Hover effects for buttons

## CSS Styles

Key styles include:

- **Container Styling:** White background, padding, and box shadow for better visual appeal
- **Button Styling:** Green colour for primary actions, hover effects, and consistent size
- **List Elements:** Display history in styled list items

## Session Management

The application stores and manages calculation history and memory functions using PHP sessions.

### Session Variables

- $_SESSION['last_result']: Stores the last calculated result
- $_SESSION['history']: An array that maintains the calculation history

### Memory Functions

- **Recall:** Fetches and displays the last stored result
- **Clear:** Removes the last result from the session

## Backend Logic

### Calculation Logic (calculate.php)

- User input is validated and parsed as floating-point numbers.
- Operations are performed using a switch statement:
  - **Addition:** $result = $num1 + $num2;
  - **Subtraction:** $result = $num1 - $num2;
  - **Multiplication:** $result = $num1 * $num2;
  - **Division:** $result = ($num2 != 0) ? ($num1 / $num2) : "Error: Division by zero";
  - **Exponentiation:** $result = pow($num1, $num2);
  - **Percentage:** $result = ($num1 / 100) * $num2;

- **Square Root:** $result = ($num1 >= 0) ? sqrt($num1): "Error: Negative square root";
- **Logarithm:** $result = ($num1 > 0) ? log($num1): "Error: Logarithm undefined for non-positive values";

- Results are stored in the session memory and redirected to index.php for display.

**Memory Management (memory.php)**

- The recall action fetches the last result from the session.
- The explicit action removes the last result from the session.

**Error Handling**

The application includes error handling for common scenarios:

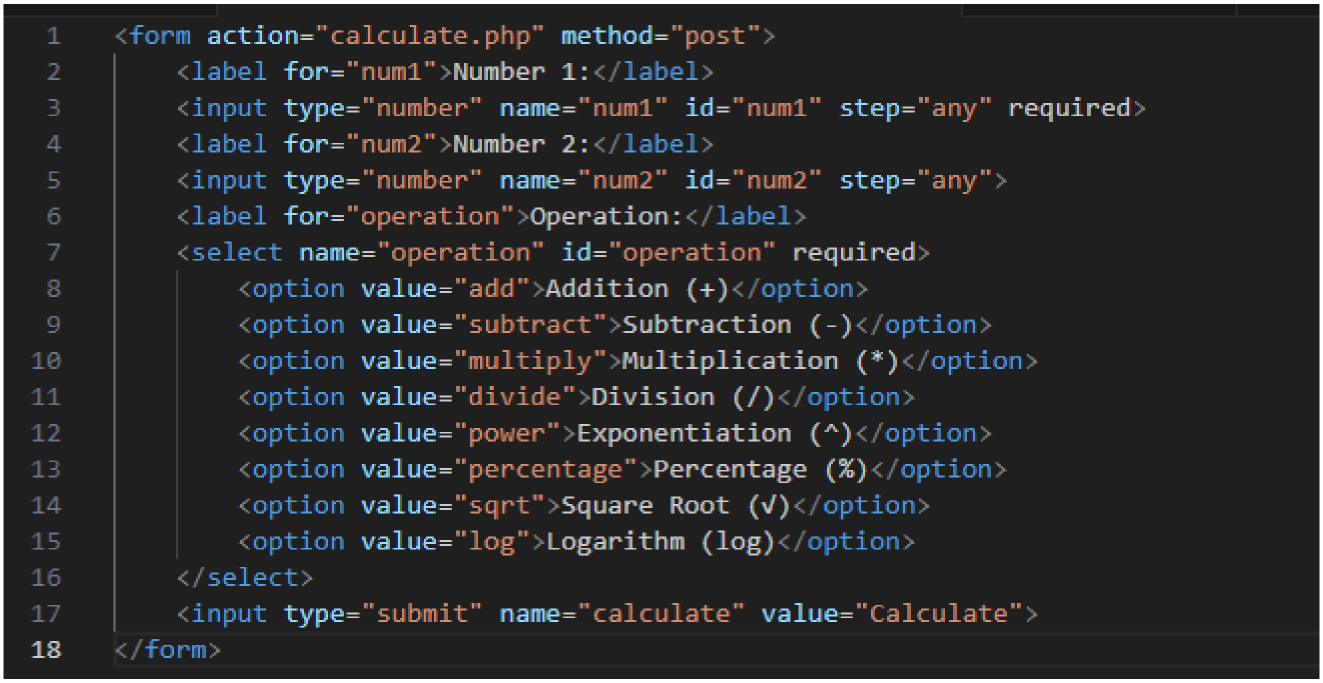
- Division by zero: Displays "Error: Division by zero"
- Negative square root: Displays "Error: Negative square root"
- Invalid logarithms: Displays "Error: Logarithm undefined for non-positive values"

**How to Use the Application**

1. **Launch the Application:** Open index.php in a browser.
2. **Enter Numbers:** Provide values for Number 1 and Number 2 (if applicable).
3. **Select Operation:** Choose a mathematical operation from the dropdown menu.
4. **Calculate:** Click the Calculate button to see the result.
5. **View History:** Check the list of previous calculations.
6. **Memory Functions:** Use the Recall Last Result and Clear Memory buttons as needed.

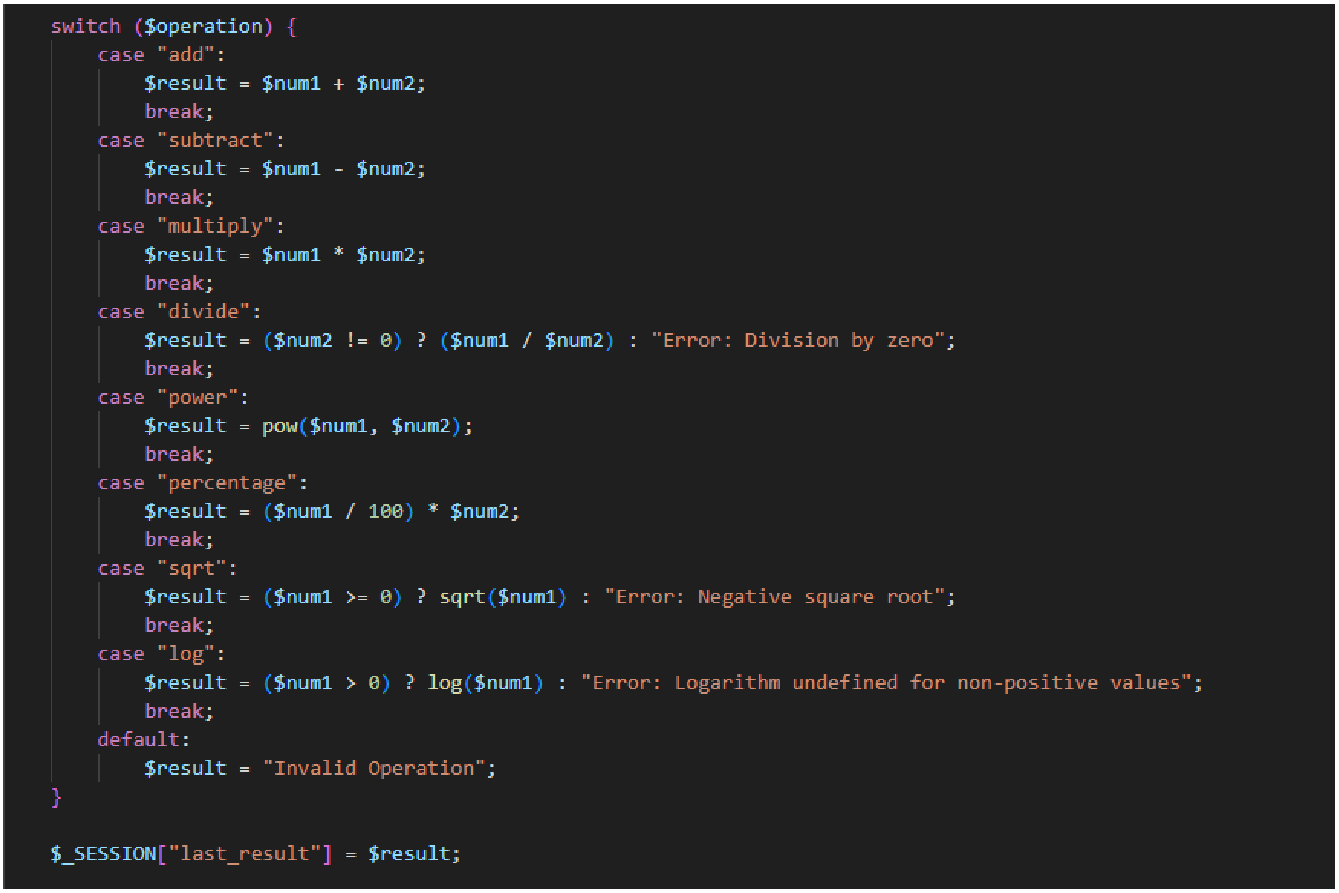
**Code Snippet Highlights**

**Form Example from index.php**



```html
<form action="calculate.php" method="post">
    <label for="num1">Number 1:</label>
    <input type="number" name="num1" id="num1" step="any" required>
    <label for="num2">Number 2:</label>
    <input type="number" name="num2" id="num2" step="any">
    <label for="operation">Operation:</label>
    <select name="operation" id="operation" required>
        <option value="add">Addition (+)</option>
        <option value="subtract">Subtraction (-)</option>
        <option value="multiply">Multiplication (*)</option>
        <option value="divide">Division (/)</option>
        <option value="power">Exponentiation (^)</option>
        <option value="percentage">Percentage (%)</option>
        <option value="sqrt">Square Root (√)</option>
        <option value="log">Logarithm (log)</option>
    </select>
    <input type="submit" name="calculate" value="Calculate">
</form>
```

## Calculation Logic Example from calculate.php



```php
switch ($operation) {
    case "add":
        $result = $num1 + $num2;
        break;
    case "subtract":
        $result = $num1 - $num2;
        break;
    case "multiply":
        $result = $num1 * $num2;
        break;
    case "divide":
        $result = ($num2 != 0) ? ($num1 / $num2) : "Error: Division by zero";
        break;
    case "power":
        $result = pow($num1, $num2);
        break;
    case "percentage":
        $result = ($num1 / 100) * $num2;
        break;
    case "sqrt":
        $result = ($num1 >= 0) ? sqrt($num1) : "Error: Negative square root";
        break;
    case "log":
        $result = ($num1 > 0) ? log($num1) : "Error: Logarithm undefined for non-positive values";
        break;
    default:
        $result = "Invalid Operation";
}

$_SESSION["last_result"] = $result;
```

## Conclusion

The Multipurpose Calculator is a comprehensive and user-friendly web-based application for various mathematical operations. With features such as session-based memory, calculation history, and error handling, it provides an efficient and interactive user experience. The modern UI design and robust backend logic make it a valuable tool for educational and practical purposes.