

INTRODUCTION TO PHP PROGRAMMING

PHP, originally an acronym for “*Personal Home Page*” (*Hypertext Preprocessor*) is a widely-used open-source server-side scripting language that is particularly suited for web development and can be embedded into **HTML**. It was originally created by **Danish-Canadian** programmer **Rasmus Lerdorf** in 1994 and has since been continually developed by a large community of contributors.

The significance of PHP lies in its role as one of the most widely used languages for building dynamic web applications. Its simplicity, flexibility, and integration with various databases make it a preferred choice for developers worldwide.

PHP's history is marked by several milestones, including the release of PHP 3 in 1998, which introduced a full-fledged language with support for object-oriented programming. Subsequent versions, such as PHP 4 and PHP 5, brought significant improvements and enhancements, including better performance and security features.

In terms of functions, PHP offers a vast array of built-in functions for various tasks, such as string manipulation, file handling, database interaction, and more. Additionally, PHP supports user-defined functions, allowing developers to create custom functionality to suit their specific needs.

PHP supports various data types, including integers, floats, strings, booleans, arrays, and objects. These data types enable developers to work with different kinds of information efficiently.

Operations in PHP encompass a wide range of tasks, including arithmetic operations, string concatenation, comparison operations, logical operations, and more. These operations allow developers to manipulate data and control the flow of their applications effectively.

Applications of PHP span across numerous domains, including:

1. **Web Development:** PHP is extensively used for creating dynamic websites, content management systems (CMS), e-commerce platforms, and web applications.
2. **Server-Side Scripting:** PHP's server-side scripting capabilities enable developers to generate dynamic content, interact with databases, handle form submissions, and perform other server-side tasks.
3. **Command-Line Scripting:** PHP can also be used for command-line scripting, allowing developers to automate tasks and perform system administration tasks.
4. **Frameworks and CMS:** PHP frameworks like Laravel, Symfony, and CodeIgniter, as well as CMS platforms like WordPress, Drupal, and Joomla, leverage PHP's capabilities to streamline web development processes.

Overall, PHP's versatility, ease of use, and extensive community support have solidified its position as a cornerstone of web development for over two decades.

CALCULATOR

OUR PHP MINI PROJECT IS ON SIMULATING A CALCULATOR!!

The program on calculator using PHP, HTML , CSS and JS goes here..,

```
<!DOCTYPE html>

<html>

<?php

    ini_set('resplay_errors',0);

    if( isset( $_REQUEST['calculator'] ))
    {
        $op=$_REQUEST['operator'];
        $num1 = $_REQUEST['firstnum'];
        $num2 = $_REQUEST['secondnum'];

        if($op=="+")
        {
            $res= $num1+$num2;
        }
        if($op=="-")
        {
            $res= $num1-$num2;
        }
        if($op=="*")
        {
            $res =$num1*$num2;
        }
        if($op=="/")
        {

```

```
$res= $num1/$num2;

}

if($_REQUEST['firstnum']==NULL || $_REQUEST['secondnum']==NULL)
{
    echo "<script language=javascript> alert(\"Enter values.\");</script>";
}
}
?>

<head>

    <script src=
"https://cdnjs.cloudflare.com/ajax/libs/mathjs/10.6.4/math
.min.js"
        integrity=
"sha512-
iphNRh6dPbeuPGIrQbCdbBF/qcqadKWLla35YPVfMZMHBSI6PLJh1om2xCT
WhpVpmUyb4IvVS9iYnnYMkleVXLA=="
        crossorigin="anonymous"
        referrerpolicy="no-referrer">

    </script>

    <style>

        table {

            border: 1px solid white;

            margin-left: auto;

            margin-right: auto;

        }

        input[type="button"] {

            width: 88%;

            padding: 20px 45px;
```

```
        background-color: grey;
        color: black;
        font-size: 25px;
        font-weight: bold;
        border: none;
        border-radius: 5px;
    }
    input[type="text"] {
        padding: 20px 120px;
        font-size: 26px;
        font-weight: bold;
        border: none;
        border-radius: 8px;
        border: 3px solid black;
    }
</style>
</head>
<body>
    <table id="calculator">
        <tr>
            <td colspan="3">
                <input type="text" id="answer">
            </td>
        </tr>
        <tr>
            <td>
                <input type="button"
                    value="AC"
```

```
        onclick="clear_input()"
        onkeydown="ans(event)">
</td>
<td>
    <input type="button"
        value="()"
        onclick="res('(')"
        onkeydown="ans(event)">
</td>
<td>
    <input type="button"
        value="%"
        onclick="res('%')"
        onkeydown="ans(event)">
</td>
<td>
    <input type="button"
        value="/"
        onclick="res('/')"
        onkeydown="ans(event)">
</td>
</tr>
<tr>
<td>
    <input type="button"
        value="7"
        onclick="res('7')"
        onkeydown="ans(event)">
```

```
</td>

<td>

    <input type="button"
        value="8"
        onclick="res('8')"
        onkeydown="ans(event)">

</td>

<td>

    <input type="button"
        value="9"
        onclick="res('9')"
        onkeydown="ans(event)">

</td>

<td>

    <input type="button"
        value="*"
        onclick="res('*)"
        onkeydown="ans(event)">

</td>
</tr>

<tr>

<td>

    <input type="button"
        value="4"
        onclick="res('4)"
        onkeydown="ans(event)">

</td>
```

```
<td>

    <input type="button"
        value="5"
        onclick="res('5')"
        onkeydown="ans(event)">

</td>

<td>

    <input type="button"
        value="6"
        onclick="res('6')"
        onkeydown="ans(event)">

</td>

<td>

    <input type="button"
        value="-"
        onclick="res('-)"
        onkeydown="ans(event)">

</td>
</tr>
<tr>
<td>

    <input type="button"
        value="1"
        onclick="res('1)"
        onkeydown="ans(event)">

</td>
<td>

    <input type="button"
```

```
        value="2"
        onclick="res('2')"
        onkeydown="ans(event)">
    </td>
    <td>
        <input type="button"
            value="3"
            onclick="res('3')"
            onkeydown="ans(event)">
    </td>
    <td>
        <input type="button"
            value="+"
            onclick="res('+)"
            onkeydown="ans(event)">
    </td>
</tr>
<tr>
    <td>
        <input type="button"
            value="0"
            onclick="res('0)"
            onkeydown="ans(event)">
    </td>
    <td>
        <input type="button"
            value="."
            onclick="res('.')"
```



```

        onkeydown="ans(event)">

</td>

<td>

    <input type="button"
        value="clear"
        onclick="clearfield()"
        onkeydown="ans(event)">

</td>

<!-- calculate function call function calculate to evaluate value -->

<td>

    <input type="button"
        value="="
        onclick="calculate()">

</td>

</tr>

</table>

<script>

    function res(val) {
        document.getElementById("answer").value += val
    }

    function clearfield() {
        let st = document.getElementById("answer").value
        document.getElementById("answer").value =
            st.substring(0, st.length - 1);
    }

    function ans(event) {
        if (event.key == '0' || event.key == '1'
            || event.key == '2' || event.key == '3'

```

```
    || event.key == '4' || event.key == '5'
    || event.key == '6' || event.key == '7'
    || event.key == '8' || event.key == '9'
    || event.key == '+' || event.key == '-'
    || event.key == '*' || event.key == '/')
    document.getElementById("answer").value += event.key;
}
var cal = document.getElementById("calculator");
cal.onkeyup = function (event) {
    if (event.keyCode === 13) {
        console.log("Enter");
        let a = document.getElementById("answer").value
        console.log(a);
        calculate();
    }
}
function calculate() {
    let a = document.getElementById("answer").value
    let b = math.evaluate(a)
    document.getElementById("answer").value = b
}
function clear_input() {
    document.getElementById("answer").value = ""
}
</script>
</body>
</html>
```

IMPLEMENTATION OF CALCULATOR

It works using the following inputs:

HTML Structure:

The HTML structure consists of a table element (<table>) with various rows and columns for organizing the calculator buttons.

An input field (<input type="text">) is provided to display the result.

CSS:

Some basic CSS styling is applied to the calculator buttons and the input field to enhance the visual appearance.

JavaScript Functions:

res(val): This function is called when a number or operator button is clicked. It appends the value of the clicked button to the input field.

clearfield(): This function is called when the "clear" button is clicked. It removes the last character from the input field.

ans(event): This function is called when a key is pressed. It allows input of numbers and operators through keyboard keys.

calculate(): This function evaluates the expression entered in the input field using the `math.evaluate()` function provided by the `Math.js` library. It then displays the result in the input field.

clear_input(): This function clears the input field when the "AC" (all clear) button is clicked.

Event Handling:

Each button has an `onclick` attribute that triggers specific functions when clicked.

The calculator table has an `onkeyup` event listener that detects the "Enter" keypress. When the "Enter" key is pressed, the `calculate()` function is called to evaluate the expression.

PHP:

PHP code is embedded within the HTML to handle form submission. It checks if the form containing calculator inputs has been submitted (`isset($_REQUEST['calculator'])`).

Error Handling:

If the user tries to perform a calculation without entering both numbers, an alert message is displayed using JavaScript (`alert("Enter values.")`).

External Library:

The program utilizes the `Math.js` library to evaluate mathematical expressions. This library allows for more complex mathematical operations and evaluations.

Line by line with explanations of the program:

```
<!DOCTYPE html>
<html>
```

This declares the document type and starts the HTML document.

```
<?php
    ini_set('resplay_errors',0);
```

This PHP code sets the error reporting level to suppress error messages.

```
if( isset( $_REQUEST['calculator'] ))
{
    $op=$_REQUEST['operator'];
    $num1 = $_REQUEST['firstnum'];
    $num2 = $_REQUEST['secondnum'];
```

This PHP block checks if the form with the name "calculator" has been submitted. If it has, it retrieves the values of the operator and two numbers from the form.

```
if($op=="+")
{
    $res= $num1+$num2;
}
if($op=="-")
{
    $res= $num1-$num2;
}
if($op=="*")
{
    $res = $num1*$num2;
}
if($op=="/")
{
    $res= $num1/$num2;
}
```

This block calculates the result based on the operator selected (+, -, *, /) and assigns it to the variable \$res.

```

if($_REQUEST['firstnum']==NULL || $_REQUEST['secondnum']==NULL)
{
    echo "<script language=javascript> alert(\"Enter values.\");</script>";
}
}
?>

```

This part checks if both numbers are entered. If not, it displays an alert prompting the user to enter values.

```

head>
<script src=
"https://cdnjs.cloudflare.com/ajax/libs/mathjs/10.6.4/math.min.js"
    integrity=
"sha512-
iphNRh6dPbeuPGIrQbCdbBF/qcqadKWL35YPVfMZMHBSI6PLJh1om2xCTWhpVpmUy
b4IvVS9iYnnYMkleVXLA=="
    crossorigin="anonymous"
    referrerpolicy="no-referrer">
</script>

```

This includes the Math.js library, which is used for evaluating mathematical expressions.

```

<style>
table {
    border: 1px solid white;
    margin-left: auto;
    margin-right: auto;
}

```

This defines some CSS styles for the calculator table, setting a border and centering it on the page.

```

input[type="button"] {
    width: 88%;
    padding: 20px 45px;
    background-color: grey;
    color: black;
    font-size: 25px;
    font-weight: bold;
    border: none;
    border-radius: 5px;
}

```

This sets the style for the calculator buttons, specifying their width, padding, background color, font size, and other properties.

```
input[type="text"] {
    padding: 20px 120px;
    font-size: 26px;
    font-weight: bold;
    border: none;
    border-radius: 8px;
    border: 3px solid black;
}
</style>
</head>
```

This sets the style for the input field, specifying its padding, font size, border, and other properties.

```
<body>
  <table id="calculator">
```

This starts the table for the calculator.

```
<tr>
  <td colspan="3">
    <input type="text" id="answer">
  </td>
</tr>
<tr>
```

This creates a row with one cell spanning three columns to display the input and output.

```
<td>
  <input type="button"
    value="AC"
    onclick="clear_input()"
    onkeydown="ans(event)">
</td>
<td>
```

This creates buttons for clearing the input, adding parentheses, calculating the modulus, and division.

```
<td>
  <input type="button"
    value="()"
    onclick="res('(')'"
    onkeydown="ans(event)">
</td>
```

This creates buttons for parentheses.

```

<td>
  <input type="button"
    value="%"
    onclick="res('%')"
    onkeydown="ans(event)">
</td>

```

This creates buttons for modulus calculation.

```

<td>
  <input type="button"
    value="/"
    onclick="res('/')"
    onkeydown="ans(event)">
</td>
</tr>
<tr>

```

This creates buttons for division.

```

<td>
  <input type="button"
    value="7"
    onclick="res('7')"
    onkeydown="ans(event)">
</td>
<td>
  <input type="button"
    value="8"
    onclick="res('8')"
    onkeydown="ans(event)">
</td>
<td>
  <input type="button"
    value="9"
    onclick="res('9')"
    onkeydown="ans(event)">
</td>
<td>
  <input type="button"
    value="*"
    onclick="res('*')"
    onkeydown="ans(event)">
</td>
</tr>

```

This creates buttons for numbers 7, 8, 9, and multiplication.

```
<tr>
  <td>
    <input type="button"
      value="4"
      onclick="res('4')"
      onkeydown="ans(event)">
  </td>
  <td>
    <input type="button"
      value="5"
      onclick="res('5')"
      onkeydown="ans(event)">
  </td>
  <td>
    <input type="button"
      value="6"
      onclick="res('6')"
      onkeydown="ans(event)">
  </td>
  <td>
    <input type="button"
      value="-"
      onclick="res('-)"
      onkeydown="ans(event)">
  </td>
</tr>
```

This creates buttons for numbers 4, 5, 6, and subtraction.

```
<tr>
  <td>
    <input type="button"
      value="1"
      onclick="res('1)"
      onkeydown="ans(event)">
  </td>
  <td>
    <input type="button"
      value="2"
      onclick="res('2)"
      onkeydown="ans(event)">
  </td>
  <td>
    <input type="button"
      value="3"
      onclick="res('3)"
      onkeydown="ans(event)">
  </td>
```



```

        <td>
            <input type="button"
                value="+"
                onclick="res('+)"
                onkeydown="ans(event)">
        </td>
    </tr>

```

This creates buttons for numbers 1, 2, 3, and addition.

```

<tr>
    <td>
        <input type="button"
            value="0"
            onclick="res('0)"
            onkeydown="ans(event)">
    </td>
    <td>
        <input type="button"
            value="."
            onclick="res('.')"
            onkeydown="ans(event)">
    </td>
    <td>
        <input type="button"
            value="clear"
            onclick="clearfield()"
            onkeydown="ans(event)">
    </td>

```

This creates a button for 0, decimal point and clearing the input.

```

    <td>
        <input type="button"
            value="="
            onclick="calculate()">
    </td>
</tr>
</table>

```

This creates a button for calculation and closes the table for the calculator.

```
<script>
```

This starts the JavaScript code block.

```
function res(val) {  
    document.getElementById("answer").value += val  
}
```

This defines a function res() that appends the passed value val to the input field with the ID "answer".

```
function clearfield() {  
    let st = document.getElementById("answer").value  
    document.getElementById("answer").value =  
    st.substring(0, st.length - 1);  
}
```

This defines a function clearfield() that removes the last character from the input field.

```
function ans(event) {  
    if (event.key == '0' || event.key == '1'  
        || event.key == '2' || event.key == '3'  
        || event.key == '4' || event.key == '5'  
        || event.key == '6' || event.key == '7'  
        || event.key == '8' || event.key == '9'  
        || event.key == '+' || event.key == '-'  
        || event.key == '*' || event.key == '/')  
        document.getElementById("answer").value += event.key;  
}
```

This defines a function ans() that allows input of numbers and operators through keyboard keys.

```
var cal = document.getElementById("calculator");
```

This selects the calculator table element.

```
cal.onkeyup = function (event) {  
    if (event.keyCode === 13) {  
        console.log("Enter");  
        let a = document.getElementById("answer").value  
        console.log(a);  
        calculate();  
    }  
}
```

This adds an event listener to the calculator table that triggers the calculate() function when the "Enter" key is pressed.

```
function calculate() {  
    let a = document.getElementById("answer").value  
    let b = math.evaluate(a)  
    document.getElementById("answer").value = b  
}
```

This defines a function `calculate()` that evaluates the expression entered in the input field using the `math.evaluate()` function provided by the `Math.js` library and displays the result in the input field.

```
function clear_input() {  
    document.getElementById("answer").value = ""  
}  
</script>
```

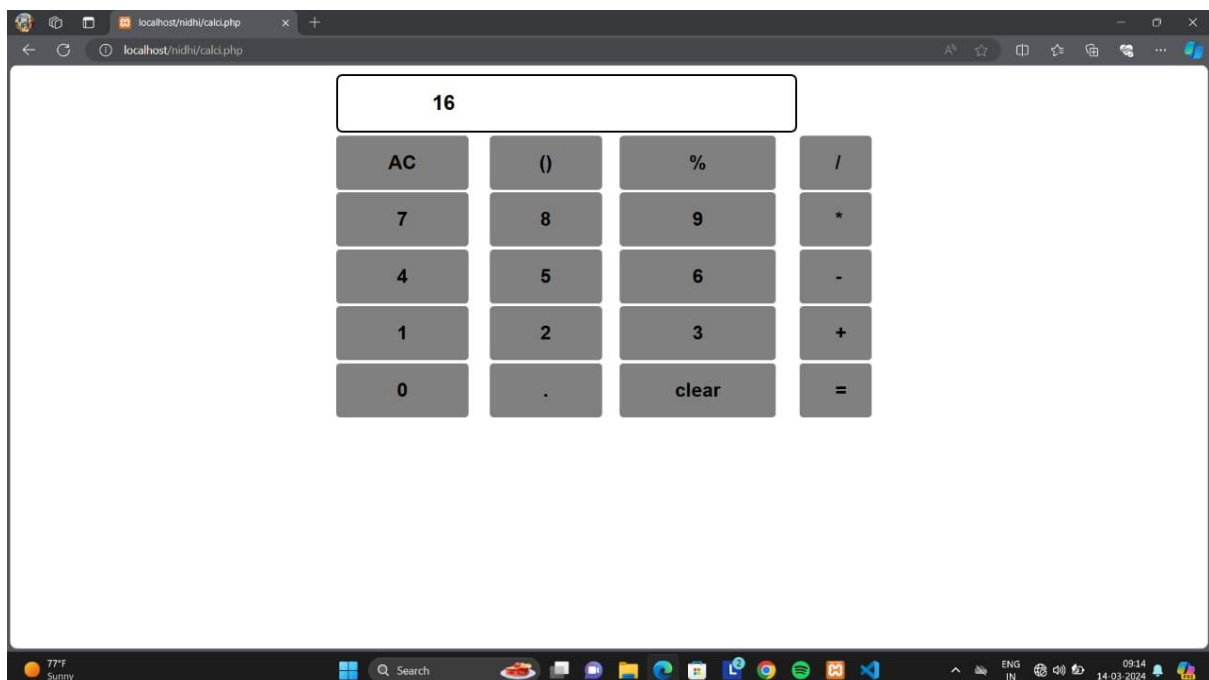
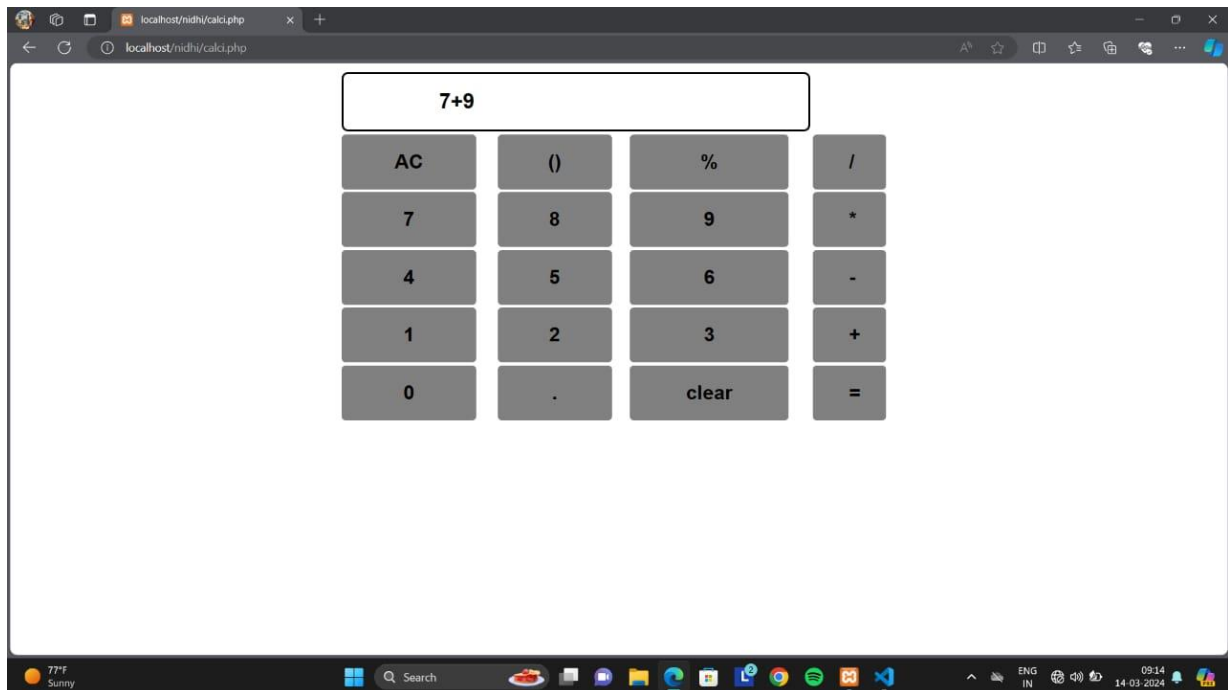
This defines a function `clear_input()` that clears the input field.

```
</body>  
</html>
```

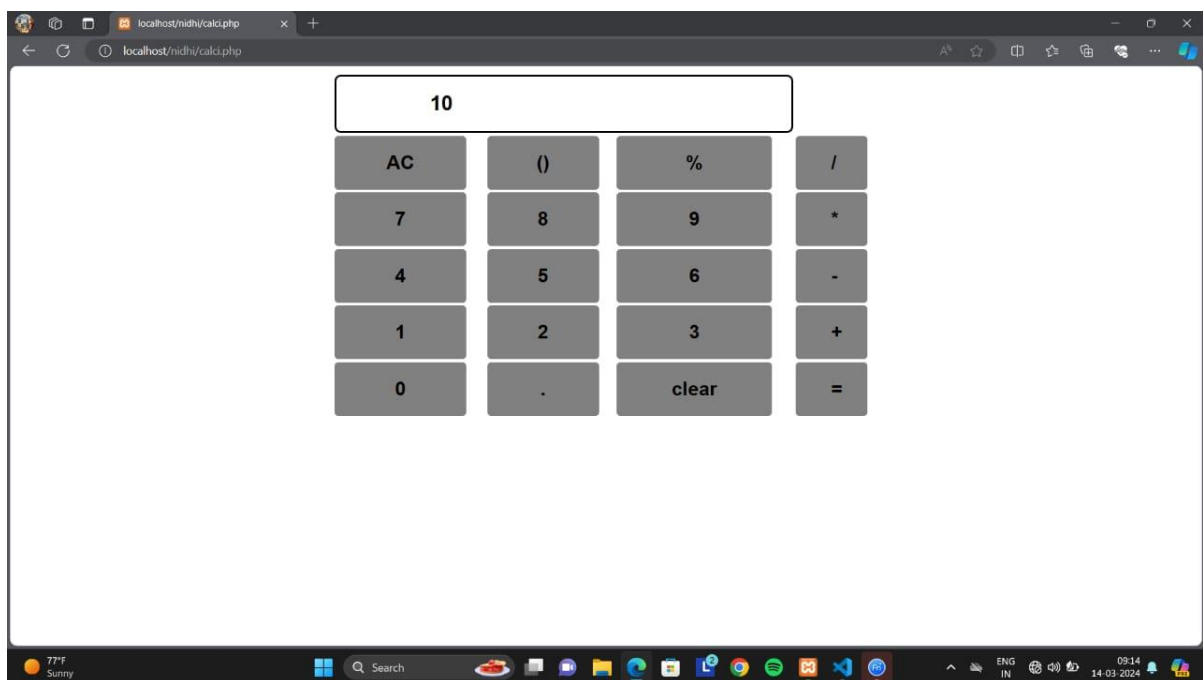
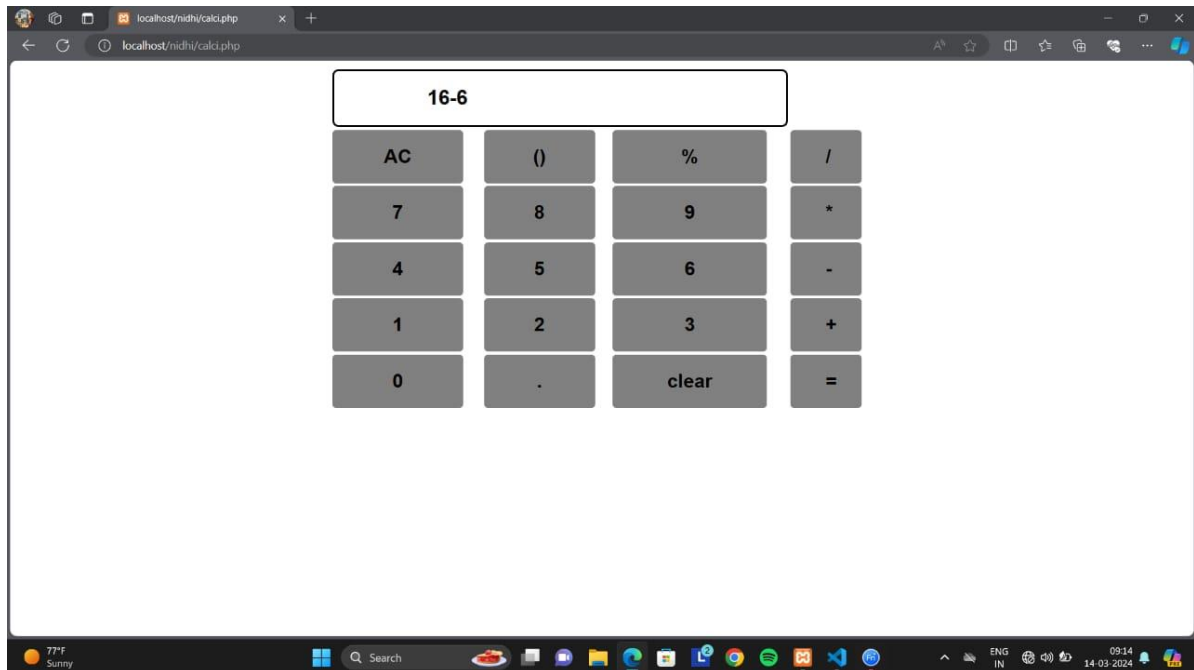
This closes the HTML document.

Outputs:

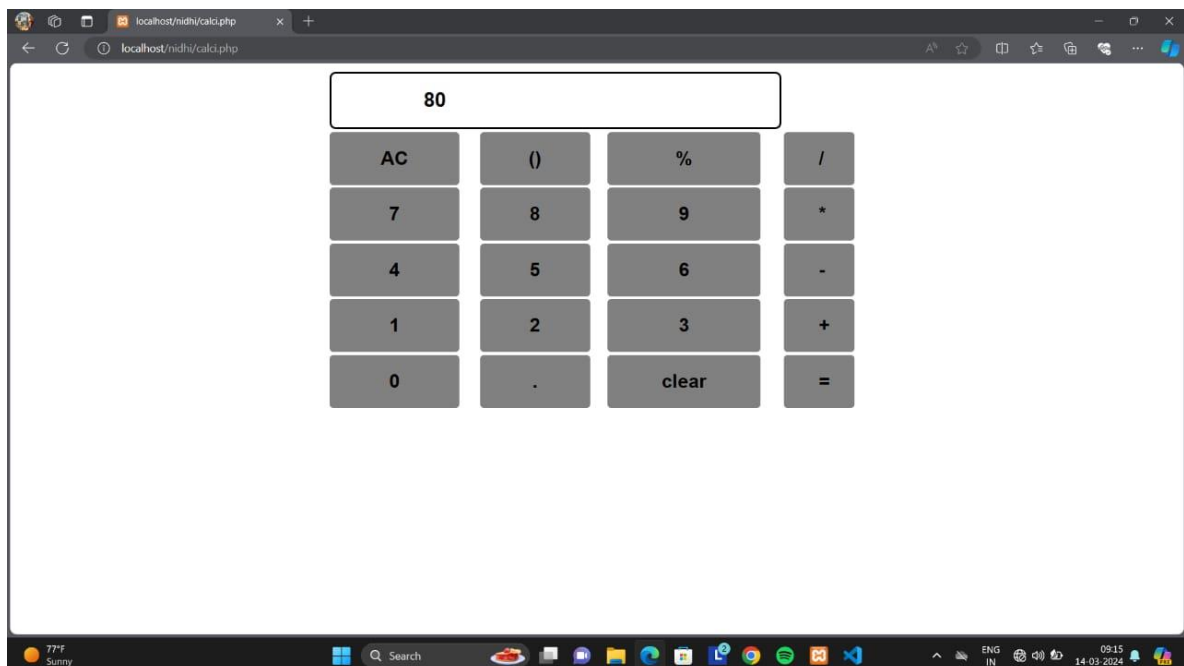
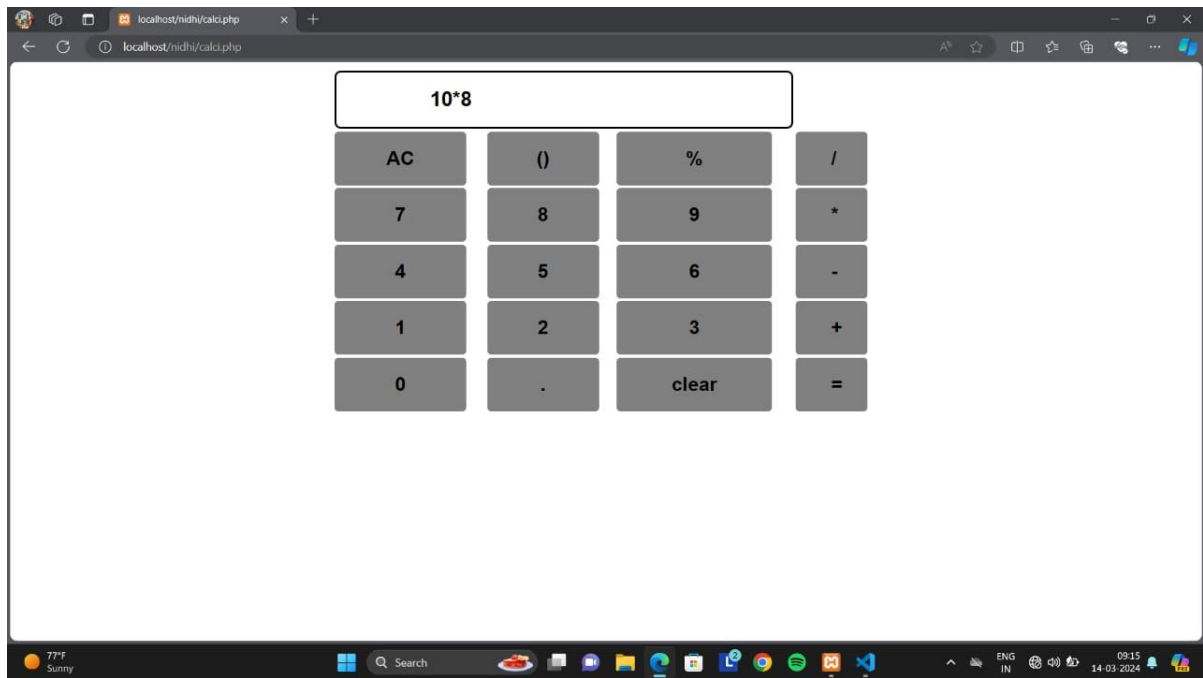
➤ ADDITION OPERATION:



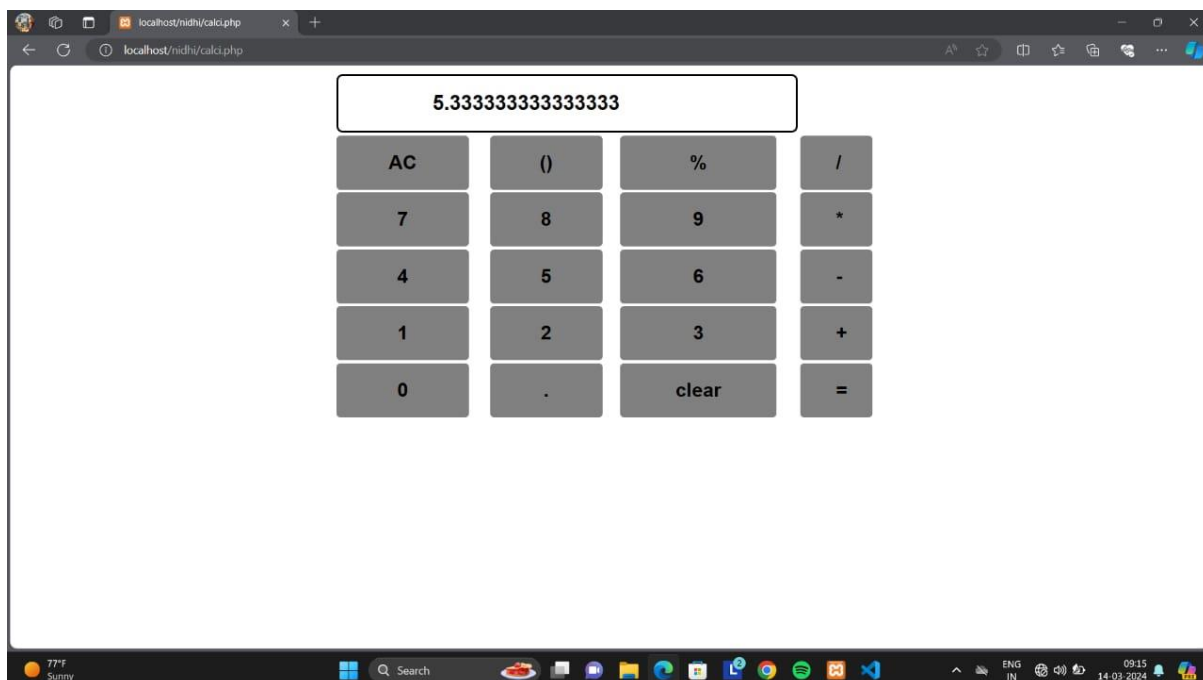
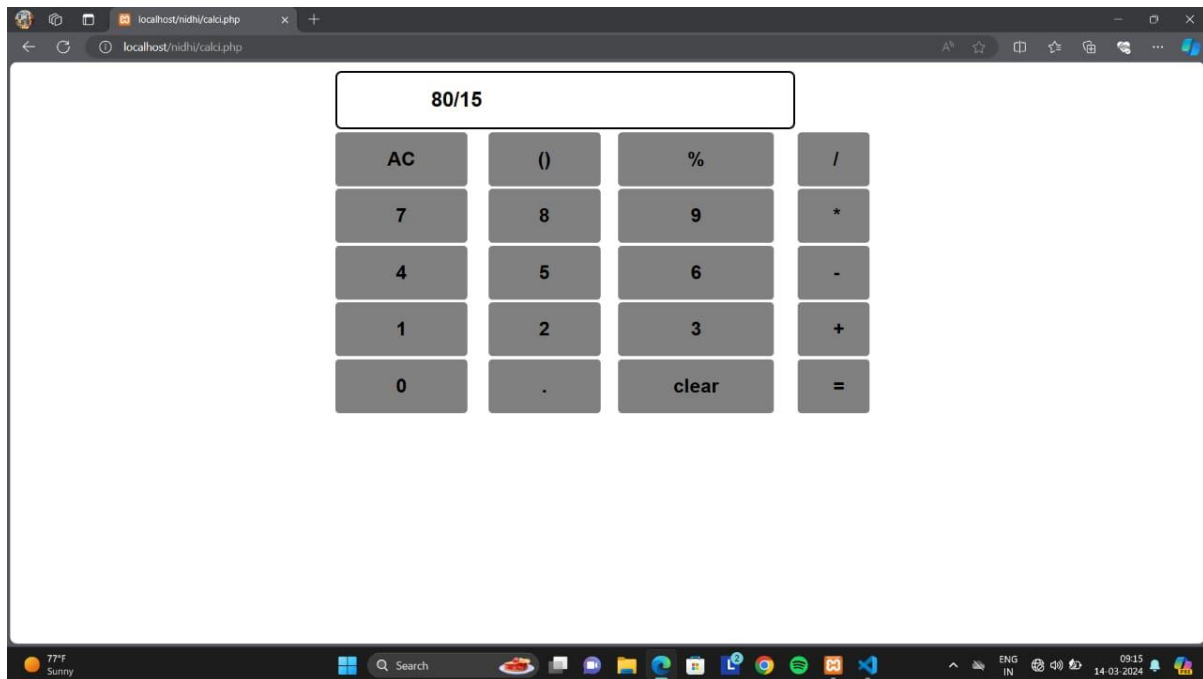
➤ **SUBTRACTION OPERATION:**



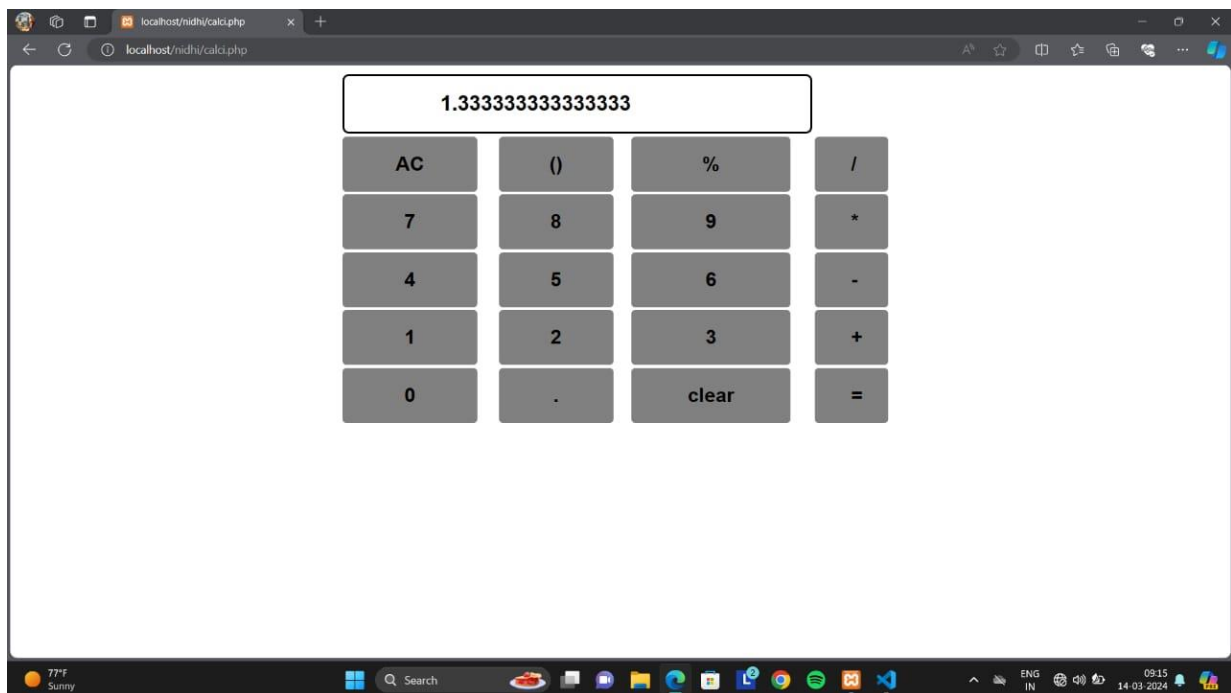
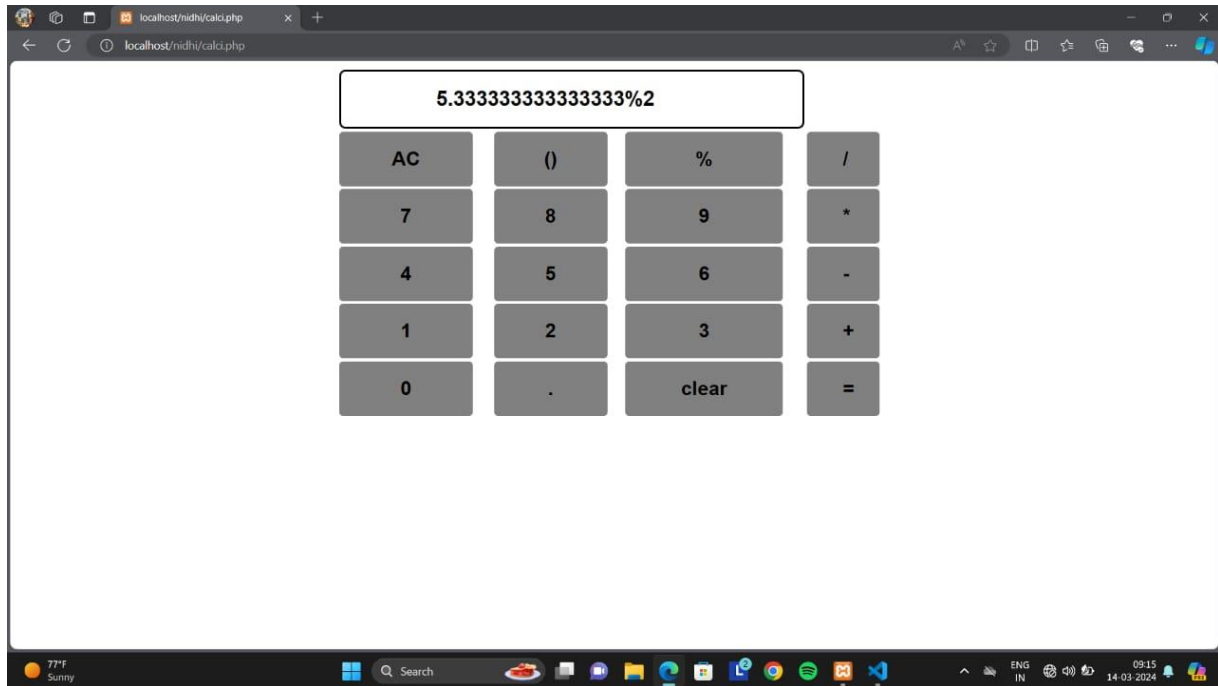
➤ MULTIPLICATION OPERATION:



➤ **DIVISION OPERATION:**



➤ MODULUS OPERATION:



CONCLUSION

In conclusion, this code creates a fully functional web-based calculator application. It combines HTML for the structure, CSS for styling, JavaScript for interactive functionality, and PHP for server-side processing. Users can perform basic arithmetic operations such as addition, subtraction, multiplication, and division either by clicking the buttons or by typing on the keyboard. The application also handles error cases, such as when users attempt to perform calculations without entering both numbers, by displaying an alert message. Overall, it provides a convenient and intuitive way for users to perform mathematical calculations directly within their web browser.