**DOCTORHARISINGHGOUR**

**VISHWAVIDYALAYA**

**DEPARTMENTOFCOMPUTER APPLICATIONS**



**CSA-CC-3206 : WEB APPLICATION LAB**

**SESSION: 2024-2025**

**SUBMITTED BY SAKSHI PANDEY MCAIIISEMESTER Y23271024**

**SUBMITTEDTO**

**ER.KAMALKANTSIR**

# QUESTION 1 => WRITE INSTALLATION STEPS OF XAMPP

# Here are the steps to install XAMPP on a Windows machine:

# Step 1: Download XAMPP

# 1. Visit the [official XAMPP website](https://www.apachefriends.org/index.html).

# 2. Choose the version of XAMPP based on your operating system (Windows, Linux, or macOS).

# 3. Click the "Download" button for the latest version.

# Step 2: Run the Installer

# 1. Once the download is complete, locate the downloaded file (e.g., `xampp-windows-x.x.x-installer.exe` for Windows).

# 2. Double-click the installer to run it.

# 3. If prompted by User Account Control (UAC), click "Yes" to allow the installer to make changes to your device.

# Step 3: Select Components

# 1. The installer will open a setup wizard.

# 2. Click "Next" on the initial welcome screen.

# 3. You'll be asked to select the components you want to install (e.g., Apache, MySQL, PHP, Perl). By default, all components are selected. You can leave them checked or uncheck any components you don’t need.

# 4. Click "Next."

# Step 4: Choose Installation Folder

# 1. Choose the folder where XAMPP should be installed. By default, it installs in `C:\xampp`.

# 2. Click "Next."

# Step 5: Language Selection

# 1. Select the language for XAMPP's interface (English or your preferred language).

# 2. Click "Next."

# Step 6: Begin Installation

# 1. Click "Next" to start the installation process.

# 2. Wait for the installation to complete. This may take a few minutes depending on your system.

# Step 7: Start the XAMPP Control Panel

# 1. Once the installation is complete, check the box to launch the XAMPP Control Panel.

# 2. Click "Finish."

# Step 8: Run Apache and MySQL

# 1. In the XAMPP Control Panel, you’ll see a list of services (Apache, MySQL, FileZilla, etc.).

# 2. Click the "Start" button next to Apache and MySQL to start the web server and database server.

# 3. If the services start successfully, the module names will turn green.

# Step 9: Test the Installation

# 1. Open your web browser.

# 2. Type `http://localhost` or `http://127.0.0.1` in the address bar and hit Enter.

# 3. If XAMPP is installed correctly, the XAMPP dashboard will appear, confirming a successful setup.

# Optional: Configure Services to Start Automatically

# 1. If you want Apache and MySQL to start automatically when you launch XAMPP, check the "Svc" checkbox next to each service in the Control Panel.

# QUESTION 2 => WAP TO PRINT HELLO MESSAGE

<!DOCTYPE html>

<html lang="en">

<head>

<metacharset="UTF-8">

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

<title>PHP-Hello,World!</title>

<style>

body{

display:flex;

justify-content:center; align-items: center; height: 100vh;

background-color: #f0f0f0; font-family:Arial,sans-serif;

}

h1{

color:#333;

}

</style>

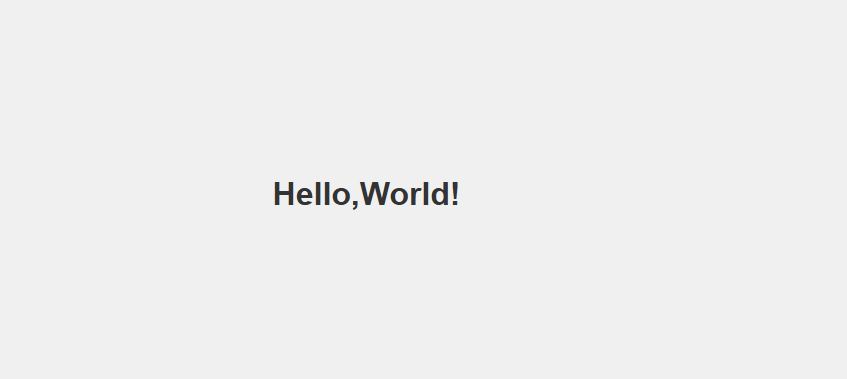
</head>

<body>

<h1><?phpecho'Hello,World!';?></h1>

</body>

</html>



# QUESTION 3 => WAP TO USE OF DATATYPES

<!DOCTYPE html>

<html lang="en">

<head>

<metacharset="UTF-8">

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

<title>PHPDataTypes</title>

<style>

body{

display:flex;

flex-direction:column; align-items: center; justify-content:center; height: 100vh;

background-color: #f0f0f0; font-family:Arial,sans-serif;

}

.datatype { margin: 10px; padding:10px;

border: 1px solid #ccc; background-color:#fff; width: 300px;

text-align:center;

}

</style>

</head>

<body>

<divclass="datatype">

<h2>String</h2>

<?php

$string="Hello,World!"; echo $string;

?>

</div>

<divclass="datatype">

<h2>Integer</h2>

<?php

$integer=123; echo $integer;

?>

</div>

<divclass="datatype">

<h2>Float</h2>

<?php

$float=3.14; echo $float;

?>

</div>

<divclass="datatype">

<h2>Boolean</h2>

<?php

$boolean= true;

echo$boolean?'True':'False';

?>

</div>

<divclass="datatype">

<h2>Array</h2>

<?php

$array=array("Apple","Banana","Cherry"); echo implode(", ", $array);

?>

</div>

<divclass="datatype">

<h2>Object</h2>

<?php

class Car {public$make;

public$model;

public function construct($make,$model){

$this->make=$make;

$this->model=$model;

}

}

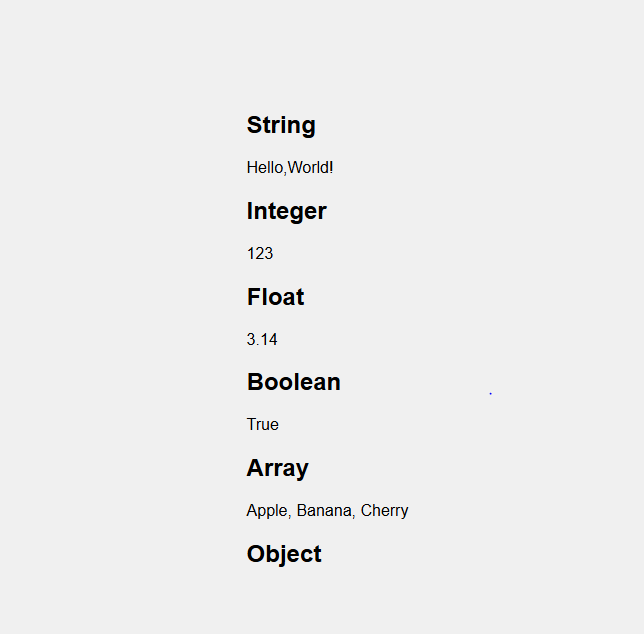
$car = new Car("Toyota", "Corolla"); echo$car->make."".$car->model;

?>

</div>

</body>

</html>



# QUESTION 4 => WAP TO SHOW USE OF OPERATORS .

<!DOCTYPE html>

<html lang="en">

<head>

<metacharset="UTF-8">

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

<title>PHPOperators</title>

<style>

body{

display:flex;

flex-direction:column; align-items: center; justify-content:center; height: 150vh;

background-color: #f0f0f0; font-family:Arial,sans-serif;

}

.operator { margin: 10px; padding:10px;

border: 1px solid #ccc; background-color:#fff; width: 300px;

text-align:center;

}

</style>

</head>

<body>

<divclass="operator">

<h2>Addition</h2>

<?php

$a= 10;

$b =20;

$sum=$a+$b;

echo"$a +$b = $sum";

?>

</div>

<divclass="operator">

<h2>Subtraction</h2>

<?php

$sub=$a -$b;

echo"$a -$b =$sub";

?>

</div>

<divclass="operator">

<h2>Multiplication</h2>

<?php

$mul=$a\* $b;

echo"$a \*$b = $mul";

?>

</div>

<divclass="operator">

<h2>Division</h2>

<?php

$div=$a/$b;

echo"$a /$b= $div";

?>

</div>

<divclass="operator">

<h2>Modulus</h2>

<?php

$mod=$a%$b;

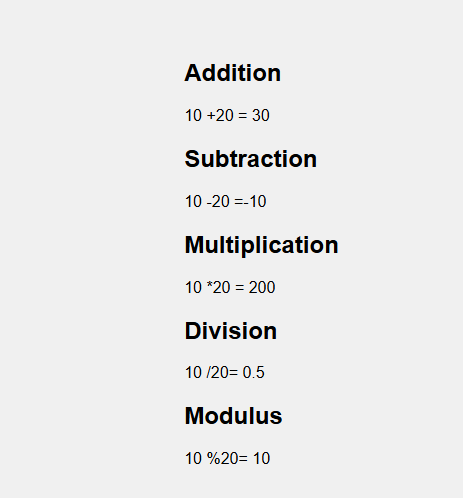
echo"$a %$b= $mod";

?>

</div>

</body>

</html>



# QUESTION 5 => WAP TO SHOW USE OF IF ELSE STATEMENT.

<!DOCTYPE html>

<html lang="en">

<head>

<metacharset="UTF-8">

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

<title>PHPIfElseStatement</title>

<style>

body{

display:flex;

flex-direction:column; align-items: center; justify-content:center;

height:100vh;

background-color: #f0f0f0; font-family:Arial,sans-serif;

}

.result{

margin: 10px; padding: 10px;border:1pxsolid#ccc;

background-color:#fff; width: 300px;

text-align:center;

}

</style>

</head>

<body>

<divclass="result">

<h2>CheckNumber</h2>

<?php

$number=10;

 if ($number > 0) {

echo"$number is a positive number.";

}elseif($number<0) {

echo"$numberis a negative number.";

}else{

echo"$numberis zero.";

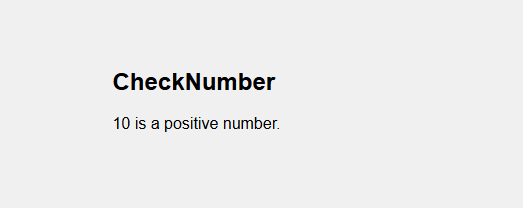
}

?>

</div>

</body>

</html>



# QUESTION 6 => WAP TO SHOW USE OF SWITCH STATEMENT .

<!DOCTYPE html>

<html lang="en">

<head>

<metacharset="UTF-8">

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

<title>PHPSwitchStatement</title>

<style>

body{

display:flex;

flex-direction:column; align-items: center; justify-content:center; height: 100vh;

background-color: #f0f0f0; font-family:Arial,sans-serif;

}

.result{

margin: 10px; padding: 10px;border:1pxsolid#ccc;

background-color:#fff; width: 300px;

text-align:center;

}

</style>

</head>

<body>

<divclass="result">

<h2>Day of the  Week</h2>

<?php

$day=date("l"); switch ($day) {

case"Monday":

echo"TodayisMonday.Startoftheworkweek!"; break;

case"Tuesday":

echo"TodayisTuesday.Keepgoing!"; break;

case"Wednesday":

echo"TodayisWednesday.Halfwaythrough!"; break;

case"Thursday":

echo"TodayisThursday.Almostthere!"; break;

case"Friday":

    echo"TodayisFriday.Weekendisnear!"; break;

    case"Saturday":

    echo"TodayisSaturday.Enjoyyourweekend!"; break;

    case"Sunday":

    echo"TodayisSunday.Relaxandrecharge!"; break;

    default:

    echo"Unknownday!"; break;

}

?>

</div>

</body>

</html>



# QUESTION 7 => WAP TO SHOW USE OF WHILE LOOP.

<!DOCTYPE html>

<html>

<head>

<title>WhileLoopExample</title>

<style>

body{

font-family:Arial,sans-serif; text-align: center;

margin-top:50px;

}

.number{

font-size:24px; margin: 10px;

}

</style>

</head>

<body>

<h1>Numbersfrom1to 10</h1>

<?php

$count= 1;

while($count<=10){

echo"<divclass='number'>$count</div>";

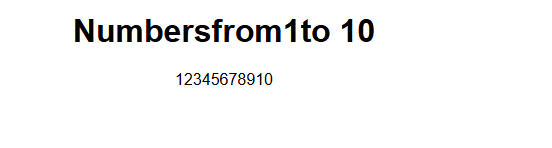
$count++;

}

?>

</body>

</html>



# QUESTION 8 => WAP TO SHOW USE OF DO WHILE LOOP.

<!DOCTYPE html>

<html>

<head>

<title>DoWhileLoopExample</title>

<style>

body{

font-family:Arial,sans-serif; text-align: center;

margin-top:50px;

}

.number{

font-size:24px; margin: 10px;

}

</style>

</head>

<body>

<h1>Numbersfrom1to 10</h1>

<?php

$count=1; do {

echo"<divclass='number'>$count</div>";

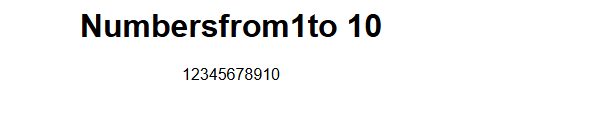
$count++;

}while($count<=10);

?>

</body>

</html>



# QUESTION 9 => WAP TO SHOW USE OF FOR LOOP.

<!DOCTYPE html>

<html>

<head>

<title>ForLoopExample</title>

<style>

body{

font-family:Arial,sans-serif; text-align: center;

margin-top:50px;

}

.number{

font-size:24px; margin: 10px;

}

</style>

</head>

<body>

<h1>Numbersfrom1to 10</h1>

<?php

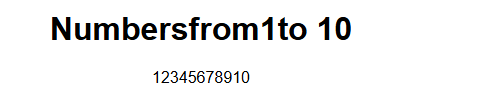
for ($count = 1; $count <= 10; $count++) { echo"<divclass='number'>$count</div>";

}

?>

</body>

</html>



# QUESTION 10 => WAP IN TO SHOW USE OF FOR EACH LOOP .

<!DOCTYPE html>

<html>

<head>

<title>ForeachLoopExample</title>

<style>

body{

font-family:Arial,sans-serif; text-align: center;

margin-top:50px;

}

.color{

font-size:24px;

margin:10px;

}

</style>

</head>

<body>

<h1>ListofColors</h1>

<?php

$colors=array("Red","Green","Blue","Yellow","Purple"); foreach ($colors as $color) {

echo"<divclass='color'>$color</div>";

}

?>

</body>

</html>

# 

# QUESTION 11 => WAP TO CHECK EVEN OR ODD NUMBER .

?php

// Input number

$number = 7;  // You can change this value to test with different numbers

// Check if the number is even or odd

if ($number % 2 == 0) {

    echo "$number is even.";

} else {

    echo "$number is odd.";

}

?>



# QUESTION 12 => WAP TO CHECK PRIME NUMBER OR NOT.

<?php

// Function to check if a number is prime

function isPrime($number) {

    // Prime numbers are greater than 1

    if ($number <= 1) {

        return false;

    }

    // Check divisibility from 2 to the square root of the number

    for ($i = 2; $i <= sqrt($number); $i++) {

        if ($number % $i == 0) {

            return false; // Divisible by another number, not prime

        }

    }

    return true; // Prime if no divisors found

}

// Input number

$number = 29;  // Change this value to test with other numbers

// Check if the number is prime

if (isPrime($number)) {

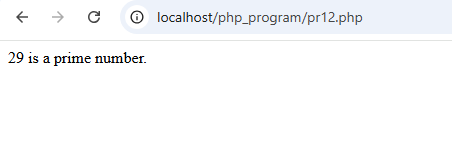
    echo "$number is a prime number.";

} else {

    echo "$number is not a prime number.";

}

?>



# QUESTION 13 => WAP TO FIND FACTORIAL OF A NUMBER .

<?php

// Function to calculate factorial

function factorial($number) {

    $fact = 1;  // Start with 1 because factorial starts at 1

    // Multiply from 1 to the given number

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

        $fact \*= $i; // Multiply the current result by i

    }

    return $fact;

}

// Input number

$number = 5;  // You can change this number to test

// Check if the number is non-negative

if ($number < 0) {

    echo "Factorial is not defined for negative numbers.";

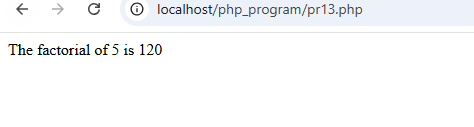
} else {

    // Display the factorial of the number

    echo "The factorial of $number is " . factorial($number);

}

?>



# QUESTION 14 => WAP TO CHECK STRONG NUMBER OR NOT.

<?php

// Function to calculate factorial of a number

function factorial($n) {

    $fact = 1;

    for ($i = 1; $i <= $n; $i++) {

        $fact \*= $i;

    }

    return $fact;

}

// Function to check if a number is a Strong number

function isStrongNumber($number) {

    $sum = 0;

    $temp = $number;

    // Loop through each digit of the number

    while ($temp > 0) {

        // Get the last digit

        $digit = $temp % 10;

        // Add the factorial of the digit to the sum

        $sum += factorial($digit);

        // Remove the last digit from the number

        $temp = (int)($temp / 10);

    }

    // Check if the sum of the factorials of the digits is equal to the number

    if ($sum == $number) {

        return true;

    } else {

        return false;

    }

}

// Input number

$number = 145;  // You can change this number to test

// Check if the number is a Strong number

if (isStrongNumber($number)) {

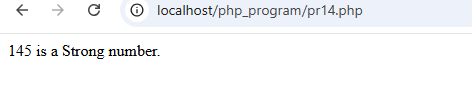
    echo "$number is a Strong number.";

} else {

    echo "$number is not a Strong number.";

}

?>

****

# QUESTION 15 => WAP TO CHECK PALLINDROME NUMBER OR NOT.

<?php

// Function to check if a number is palindrome

function isPalindrome($number) {

    // Convert the number to a string

    $strNumber = (string)$number;

    // Reverse the string and compare it with the original

    if ($strNumber == strrev($strNumber)) {

        return true;  // The number is a palindrome

    } else {

        return false; // The number is not a palindrome

    }

}

// Input number

$number = 121;  // You can change this number to test

// Check if the number is a palindrome

if (isPalindrome($number)) {

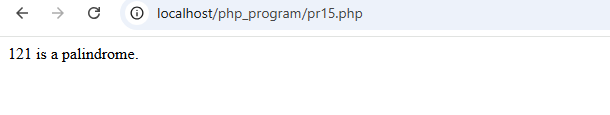
    echo "$number is a palindrome.";

} else {

    echo "$number is not a palindrome.";

}

?>



# QUESTION 16=> WAP TO SHOW USER DEFINED FUNCTION.

<?php

// User-defined function to calculate the sum of two numbers

function addNumbers($a, $b) {

    $sum = $a + $b;

    return $sum;

}

// Calling the function and storing the result

$result = addNumbers(5, 10);

// Displaying the result

echo "The sum of 5 and 10 is: " . $result . "<br>";

?>

<?php

// User-defined function without parameters

function greetUser() {

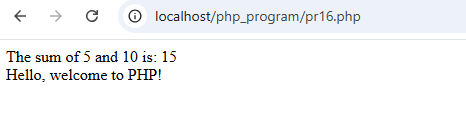
    echo "Hello, welcome to PHP!";

}

// Calling the function

greetUser(); // This will display the greeting message

?>



# QUESTION 17 => WAP TO SHOW SET() FUNCTION AND UNSET() FUNCTION

<?php

// Declaring a variable

$name = "John";

// Display the variable value

echo "Name before unset: " . $name . "<br>";

// Unset the variable

unset($name);

// Try to display the variable after unset

if (isset($name)) {

    echo "Name after unset: " . $name . "<br>";

} else {

    echo "Name has been unset and no longer exists.";

}

?>

<?php

// Start the session

session\_start();

// Set a session variable

$\_SESSION['username'] = "Alice";

// Display session variable value

echo "Session username is: " . $\_SESSION['username'] . "<br>";

// Unset the session variable

unset($\_SESSION['username']);

// Try to access the session variable after unset

if (isset($\_SESSION['username'])) {

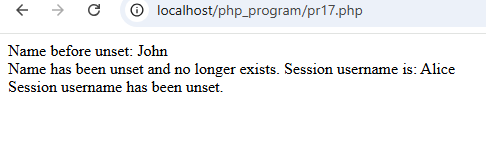
    echo "Session username is: " . $\_SESSION['username'];

} else {

    echo "Session username has been unset.";

}

?>



# QUESTION 18 => WAP TO REVERSE STRING USING SIMPLE FUNCTION AND USING PARAMETRIZED FUNCTION.

<?php

// Simple function to reverse a fixed string

function reverse\_String() {

    $str = "Hello, PHP!";  // Fixed string

    $reversedStr = strrev($str);  // Use PHP's built-in strrev() function

    echo "Original String: " . $str . "<br>";

    echo "Reversed String: " . $reversedStr . "<br>";

}

// Call the function to reverse the string

reverse\_String();

?>

<?php

// Parameterized function to reverse a string

function reverseString($str) {

    $reversedStr = strrev($str);  // Use PHP's built-in strrev() function

    return $reversedStr;

}

// Input string

$inputString = "Welcome to PHP!";

// Call the function and pass the string as an argument

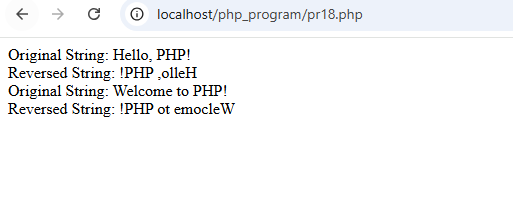
$reversedString = reverseString($inputString);

// Display the results

echo "Original String: " . $inputString . "<br>";

echo "Reversed String: " . $reversedString . "<br>";

?>



# QUESTION 19 => WAP TO SHOW USE OF STRING FUNCTIONS.

<?php

// Original string for demonstration

$string = "Hello, PHP String Functions!";

// 1. strlen() - Get the length of a string

echo "Length of string: " . strlen($string) . "<br>";

// 2. strtoupper() - Convert string to uppercase

echo "Uppercase string: " . strtoupper($string) . "<br>";

// 3. strtolower() - Convert string to lowercase

echo "Lowercase string: " . strtolower($string) . "<br>";

// 4. ucfirst() - Convert the first character of the string to uppercase

echo "First letter uppercase: " . ucfirst($string) . "<br>";

// 5. ucwords() - Convert the first character of each word to uppercase

echo "Each word capitalized: " . ucwords($string) . "<br>";

// 6. substr() - Extract a part of the string

echo "Substring (7 to 11): " . substr($string, 7, 5) . "<br>";

// 7. strrev() - Reverse the string

echo "Reversed string: " . strrev($string) . "<br>";

// 8. strpos() - Find the position of the first occurrence of a substring

echo "Position of 'PHP': " . strpos($string, "PHP") . "<br>";

// 9. str\_replace() - Replace all occurrences of a substring with another substring

echo "Replace 'PHP' with 'Java': " . str\_replace("PHP", "Java", $string) . "<br>";

// 10. trim() - Remove whitespace (or other characters) from the beginning and end of a string

echo "Trimmed string (spaces removed): '" . trim("   Hello World!   ") . "'<br>";

?>

# 

# QUESTION 20=> WAP TO CREATE CONSTRUCT OR AND INITALIZE A OBJECT OF CLASS BY USING OBJECT ORIENTED FUNCTION .

<?php

// Define a class called "Person"

class Person {

    // Properties of the class

    public $name;

    public $age;

    // Constructor to initialize properties

    function \_\_construct($name, $age) {

        $this->name = $name;  // Set the name property

        $this->age = $age;    // Set the age property

    }

    // Method to display information about the person

    function displayInfo() {

        echo "Name: " . $this->name . "<br>";

        echo "Age: " . $this->age . "<br>";

    }

}

// Create an object of the class "Person" and pass parameters to the constructor

$person1 = new Person("John Doe", 25);

// Call the method to display the object's information

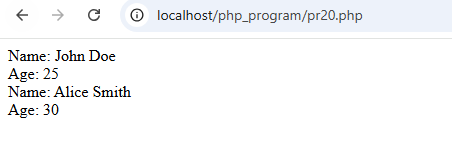
$person1->displayInfo();

// Another object initialization

$person2 = new Person("Alice Smith", 30);

$person2->displayInfo();

?>



# QUESTION 21 => WAP TO INHERIT MEMBERS OF A SUPER CLASS IN SUBCLASS.

<?php

// Superclass (Parent Class)

class Animal {

    // Property of the superclass

    public $name;

    // Constructor of the superclass

    function \_\_construct($name) {

        $this->name = $name;

    }

    // Method of the superclass

    function speak() {

        echo $this->name . " makes a sound.<br>";

    }

}

// Subclass (Child Class) that inherits from Animal

class Dog extends Animal {

    // Additional method in the subclass

    function speak() {

        echo $this->name . " barks.<br>";

    }

}

// Creating an object of the subclass (Dog)

$dog = new Dog("Buddy");

// Calling the inherited constructor (from the parent class)

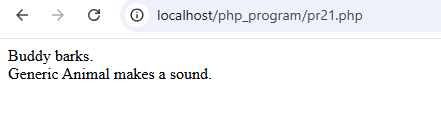
$dog->speak(); // This will call the overridden method in the Dog class

// Creating an object of the superclass (Animal)

$animal = new Animal("Generic Animal");

$animal->speak(); // This will call the method in the Animal class

?>



# QUESTION 22 => WAP TO SHOW USES OF NAMESPACES.

<?php

// Define a namespace called "Animals"

namespace Animals {

    class Dog {

        public function speak() {

            echo "Woof! Woof!<br>";

        }

    }

    class Cat {

        public function speak() {

            echo "Meow! Meow!<br>";

        }

    }

}

// Define a namespace called "Vehicles"

namespace Vehicles {

    class Car {

        public function start() {

            echo "Vroom! Vroom!<br>";

        }

    }

    class Bike {

        public function start() {

            echo "Zoom! Zoom!<br>";

        }

    }

}

// Define another namespace "Animals1" for the alias example

namespace Animals1 {

    class Dog1 {

        public function speak() {

            echo "Woof! Woof!<br>";

        }

    }

}

// Define the "MathOperations" namespace

namespace MathOperations {

    // Define a constant

    const PI = 3.14159;

    // Define functions

    function add($a, $b) {

        return $a + $b;

    }

    function multiply($a, $b) {

        return $a \* $b;

    }

}

// Using the classes from the "Animals" namespace

namespace {

    use Animals\Dog;

    use Animals\Cat;

    $dog = new Dog();

    $dog->speak();  // Output: Woof! Woof!

    $cat = new Cat();

    $cat->speak();  // Output: Meow! Meow!

}

// Using the classes from the "Vehicles" namespace

namespace {

    use Vehicles\Car;

    use Vehicles\Bike;

    $car = new Car();

    $car->start();  // Output: Vroom! Vroom!

    $bike = new Bike();

    $bike->start();  // Output: Zoom! Zoom!

}

// Using an alias for a class from the "Animals1" namespace

namespace {

    use Animals1\Dog1 as Puppy;

    $myPuppy = new Puppy();

    $myPuppy->speak();  // Output: Woof! Woof!

}

// Using functions and constants from the "MathOperations" namespace

namespace {

    // Fully qualify the function name from the MathOperations namespace

    echo "Addition of 5 and 10: " . \MathOperations\add(5, 10) . "<br>";  // Output: 15

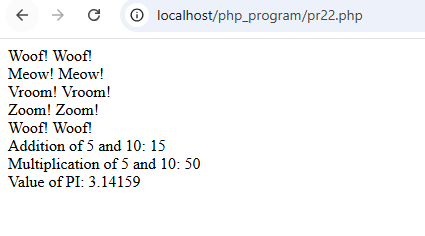
    echo "Multiplication of 5 and 10: " . \MathOperations\multiply(5, 10) . "<br>";  // Output: 50

    // Directly use the constant by referencing its full namespace

    echo "Value of PI: " . \MathOperations\PI . "<br>";  // Output: 3.14159

}

?>



# QUESTION 23 => WAP TO IMPLEMENT CALCULATOR.

<?php

// Function to perform calculation

function calculate($num1, $num2, $operator) {

    switch($operator) {

        case 'add':

            return $num1 + $num2;

        case 'subtract':

            return $num1 - $num2;

        case 'multiply':

            return $num1 \* $num2;

        case 'divide':

            if ($num2 == 0) {

                return "Error! Division by zero.";

            } else {

                return $num1 / $num2;

            }

        default:

            return "Invalid operator!";

    }

}

// Take user input

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

    $num1 = $\_POST['num1'];

    $num2 = $\_POST['num2'];

    $operator = $\_POST['operator'];

    $result = calculate($num1, $num2, $operator);

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>PHP Calculator</title>

</head>

<body>

    <h1>PHP Calculator</h1>

    <form method="post">

        <input type="number" name="num1" placeholder="Enter first number" required><br><br>

        <input type="number" name="num2" placeholder="Enter second number" required><br><br>

        <select name="operator">

            <option value="add">Add</option>

            <option value="subtract">Subtract</option>

            <option value="multiply">Multiply</option>

            <option value="divide">Divide</option>

        </select><br><br>

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

    </form>

    <?php

    if (isset($result)) {

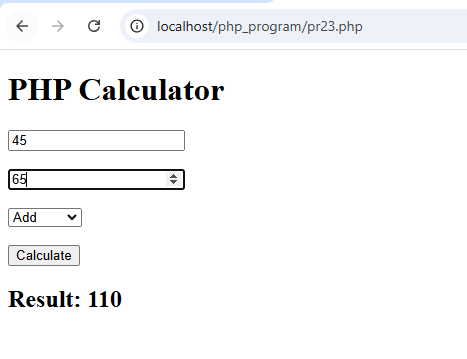
        echo "<h2>Result: $result</h2>";

    }

    ?>

</body>

</html>



# QUESTION 24 => WAP TO CALCULATE SIMPLE INTEREST.

<?php

// Function to calculate Simple Interest

function calculate\_simple\_interest($principal, $rate, $time) {

    return ($principal \* $rate \* $time) / 100;

}

// Take user input

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

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

    $rate = $\_POST['rate'];

    $time = $\_POST['time'];

    $interest = calculate\_simple\_interest($principal, $rate, $time);

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Simple Interest Calculator</title>

</head>

<body>

    <h1>Simple Interest Calculator</h1>

    <form method="post">

        <input type="number" name="principal" placeholder="Enter principal" required><br><br>

        <input type="number" name="rate" placeholder="Enter rate" required><br><br>

        <input type="number" name="time" placeholder="Enter time in years" required><br><br>

        <input type="submit" value="Calculate Interest">

    </form>

    <?php

    if (isset($interest)) {

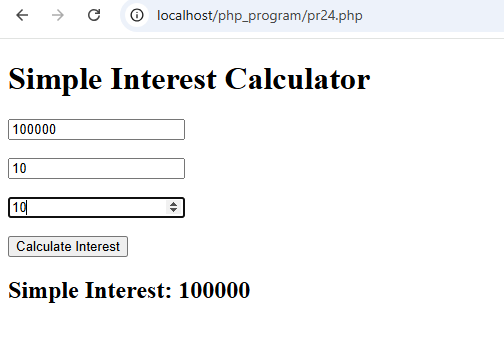
        echo "<h2>Simple Interest: $interest</h2>";

    }

    ?>

</body>

</html>



# QUESTION 25 => WAP TO CALCULATE COMPOUND INTEREST.

<?php

// Function to calculate Compound Interest

function calculate\_compound\_interest($principal, $rate, $time, $compoundings) {

    return $principal \* pow((1 + $rate / $compoundings), $compoundings \* $time) - $principal;

}

// Take user input

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

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

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

    $time = $\_POST['time'];

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

    $interest = calculate\_compound\_interest($principal, $rate, $time, $compoundings);

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Compound Interest Calculator</title>

</head>

<body>

    <h1>Compound Interest Calculator</h1>

    <form method="post">

        <input type="number" name="principal" placeholder="Enter principal" required><br><br>

        <input type="number" name="rate" placeholder="Enter annual interest rate (%)" required><br><br>

        <input type="number" name="time" placeholder="Enter time in years" required><br><br>

        <input type="number" name="compoundings" placeholder="Enter number of compoundings per year" required><br><br>

        <input type="submit" value="Calculate Interest">

    </form>

    <?php

    if (isset($interest)) {

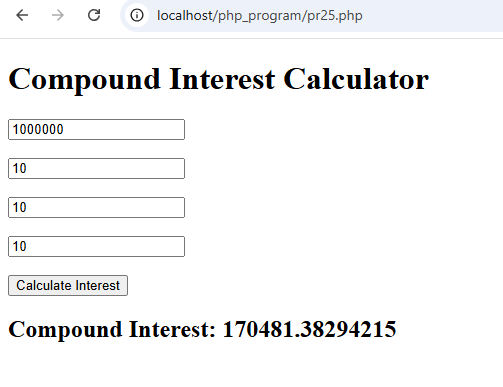
        echo "<h2>Compound Interest: $interest</h2>";

    }

    ?>

</body>

</html>



# QUESTION 26 => WAP TO SHOW USES OF FILE HANDLING.

<?php

// Writing to a file

$file = fopen("example.txt", "w"); // Open file for writing

if ($file) {

    fwrite($file, "Hello, this is a file handling demo in PHP.\n");

    fwrite($file, "This file was created and written using PHP.");

    fclose($file);  // Close the file

    echo "Data has been written to the file.<br>";

}

// Reading from the file

$file = fopen("example.txt", "r"); // Open file for reading

if ($file) {

    echo "<h2>Contents of the file:</h2>";

    while (($line = fgets($file)) !== false) {

        echo nl2br($line); // nl2br() converts newlines to HTML <br> tags

    }

    fclose($file);  // Close the file

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>PHP File Handling Demo</title>

</head>

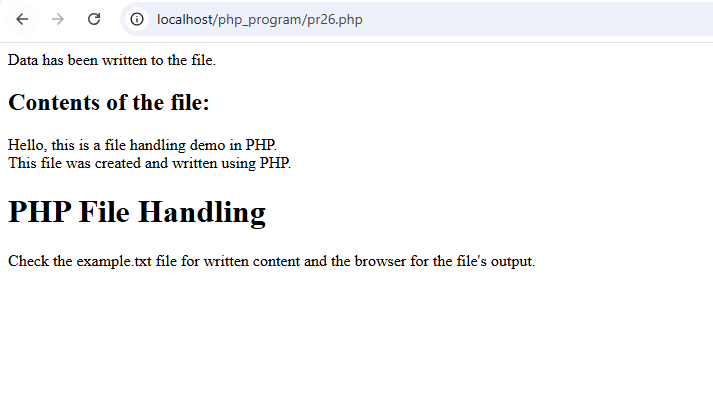
<body>

    <h1>PHP File Handling</h1>

    <p>Check the example.txt file for written content and the browser for the file's output.</p>

</body>

</html>



**QUESTION 27 => WAP TO VALIDATE EMAIL ADDRESS**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Document</title>

</head>

<body>

    <h2>27. WAP IN PHP TO VALIDATE AN EMAIL ADDRESS ENTERED BY USER

</h2>

</body>

</html>

<?php

// PHP program to validate email

// Function to validate email using regular expression

function email\_validation($str) {

    return (!preg\_match(

    "^[\_a-z0-9-]+(\.[\_a-z0-9-]+)\*@[a-z0-9-]+(\.[a-z0-9-]+)\*(\.[a-z]{2,3})$^", $str))

    ? FALSE : TRUE;

}

// Function call

if(!email\_validation("author@phpprogram.com")) {

    echo "Invalid email address.";

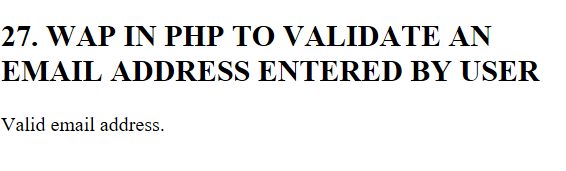
}

else {

    echo "Valid email address.";

}

?>



**QUESTION 28 => WAP IN PHP THAT REPLACES ALL SPACES IN STRING WITH UNDERSCORE**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Document</title>

</head>

<body>

    <h2>28. WAP IN PHP THAT REPLACES ALL SPACES IN STRING WITH UNDERSCORE

</h2>

</body>

</html>

<?php

echo str\_replace("world","Peter","Hello world!");

?>

****

**QUESTION 29=>WAP IN PHP TO CHECK IF A STRING IS A VALID URL OR NOT**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Document</title>

</head>

<body>

    <h2>29. WAP IN PHP TO CHECK IF A STRING IS A VALID URL OR NOT</h2>

</body>

</html>

<?php

$url = "https://www.w3schools.com";

if (filter\_var($url, FILTER\_VALIDATE\_URL)) {

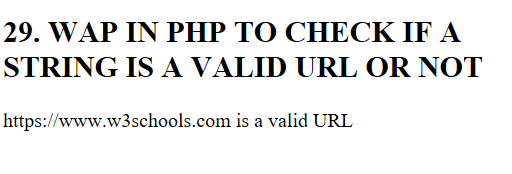
    echo("$url is a valid URL");

} else {

    echo("$url is not a valid URL");

}

?>

****

**QUESTION 30 => WAP IN PHP TO VALIDATE A PASSWORD MUST CONTAIN AT LEAST 8 CHARACTERS, ONE UPPERCASE LETTER, ONE LOWERCASE LETTER , ONE NUMBER AND ONE SPECIAL CHARACTER**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Document</title>

</head>

<body>

    <h1>30. WAP IN PHP TO VALIDATE A PASSWORD MUST CONTAIN AT LEAST 8 CHARACTERS, ONE UPPERCASE LETTER, ONE LOWERCASE LETTER , ONE NUMBER AND ONE SPECIAL CHARACTER

</h1>

</body>

</html>

<?php

$password = "PhpProgram@123";

$pattern = '/^(?=.\*[a-z])(?=.\*[A-Z])(?=.\*\d)(?=.\*[\W\_]).{8,}$/';

if (preg\_match($pattern, $password)) {

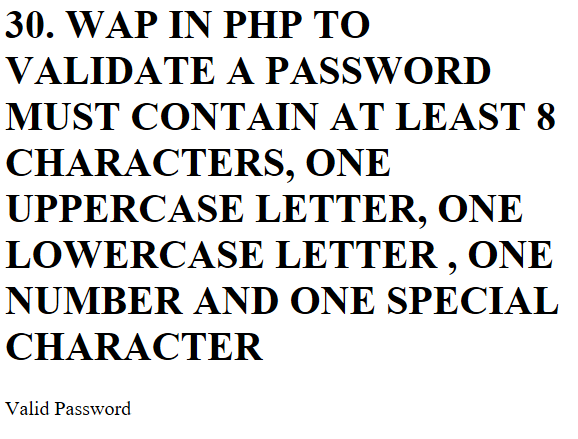
    echo "Valid Password";

} else {

    echo "Invalid Password";

}

?>

****

**QUESTION 31 => WAP IN PHP TO EXTRACT DOMAIN NAME FROM URL**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Document</title>

</head>

<body>

    <h1>31. WAP IN PHP TO EXTRACT DOMAIN NAME FROM URL</h1>

</body>

</html>

<?php

$full\_url = "http://google.com/search/console";

// 👇 parse the url and print the host part

$array = parse\_url($full\_url);

print $array["host"];

****

**QUESTION 32 => WAP IN PHP TO FIND AND REPLACE SPECIFIC WORD IN A STRING**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Document</title>

</head>

<body>

    <h1>WAP IN PHP TO FIND AND REPLACE SPECIFIC WORD IN A STRING</h1>

</body>

</html>

<?php

$arr = array("blue","red","green","yellow");

print\_r(str\_replace("red","pink",$arr,$i));

echo "<br>" . "Replacements: $i";

?>



**QUESTION 33 => WAP IN PHP TO MATCH AND COUNT THE NUMBER OF VOWELS IN ASTRING**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Document</title>

</head>

<body>

    <h1>WAP IN PHP TO MATCH AND COUNT THE NUMBER OF VOWELS IN ASTRING</h1>

</body>

</html>

<?php

function countVowels($str)

{

    // Case-insensitive match for vowels

    preg\_match\_all("/[aeiou]/i", $str, $matches);

    return count($matches[0]);

}

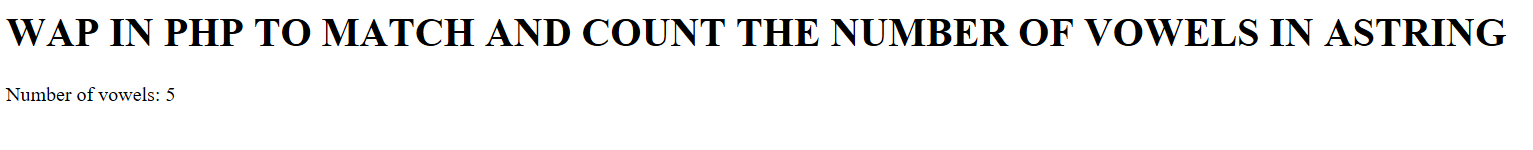
// Driver code

$str = "GeeksforGeeks";

$vowelsCount = countVowels($str);

echo "Number of vowels: " . $vowelsCount;

?>

****

**QUESTION 34 => WRITE A PHP SCRIPT TO UPLOAD AN IMAGE FILE, VALIDATE ITS TYPE AND DISPLAY IT ON THE PAGE**

<?php

/\* Write a PHP script to upload an image file, validate its type and display it on the page\*/

// Check if form is submitted

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

    // Define allowed file types

    $allowedTypes = ['image/jpeg', 'image/png', 'image/gif'];

    $uploadDir = 'uploads/';  // Directory to store uploaded images

    // Check if the upload directory exists, if not, create it

    if (!is\_dir($uploadDir)) {

        mkdir($uploadDir, 0777, true);

    }

    // Get file info

    $fileTmpName = $\_FILES['image']['tmp\_name'];

    $fileName = $\_FILES['image']['name'];

    $fileSize = $\_FILES['image']['size'];

    $fileType = $\_FILES['image']['type'];

    $fileError = $\_FILES['image']['error'];

    // Check for errors

    if ($fileError !== UPLOAD\_ERR\_OK) {

        echo "Error during file upload.";

        exit;

    }

    // Validate file type

    if (!in\_array($fileType, $allowedTypes)) {

        echo "Invalid file type. Only JPG, PNG, and GIF are allowed.";

        exit;

    }

    // Generate a unique name for the uploaded file

    $fileExtension = pathinfo($fileName, PATHINFO\_EXTENSION);

    $uniqueFileName = uniqid('img\_', true) . '.' . $fileExtension;

    // Move uploaded file to the designated folder

    if (move\_uploaded\_file($fileTmpName, $uploadDir . $uniqueFileName)) {

        // File uploaded successfully

        echo "File uploaded successfully. <br>";

        echo "Image preview: <br>";

        echo "<img src='" . $uploadDir . $uniqueFileName . "' alt='Uploaded Image' width='300px' />";

    } else {

        echo "Error moving the uploaded file.";

    }

} else {

    // Display the image upload form

    echo '<form action="" method="POST" enctype="multipart/form-data">

            <label for="image">Choose an image to upload:</label>

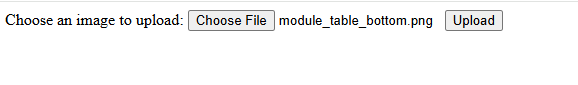
            <input type="file" name="image" id="image" required>

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

          </form>';

}

?>

****

**QUESTION 35 => WRITE A PHP SCRIPT TO CREATE A BASIC LOGIN SYSTEM USING SESSIONS**

<?php

/\* Write  a PHP script to create a basic login system using sessions \*/

session\_start();  // Start the session

// Check if the user is already logged in, if so redirect to home page

if (isset($\_SESSION['username'])) {

    header("Location: home.php");

    exit;

}

// Define a simple hard-coded username and password for demonstration purposes

$validUsername = "user";

$validPassword = "password123";

// Check if the form is submitted

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

    $username = $\_POST['username'];

    $password = $\_POST['password'];

    // Check if the credentials are correct

    if ($username === $validUsername && $password === $validPassword) {

        // Set session variables

        $\_SESSION['username'] = $username;

        // Redirect to home page

        header("Location: home.php");

        exit;

    } else {

        // Invalid credentials

        $error = "Invalid username or password.";

    }

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Login</title>

</head>

<body>

    <h2>Login</h2>

    <?php

    // Display error message if credentials are invalid

    if (isset($error)) {

        echo "<p style='color: red;'>$error</p>";

    }

    ?>

    <form action="login.php" method="POST">

        <label for="username">Username:</label>

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

        <label for="password">Password:</label>

        <input type="password" name="password" id="password" required><br><br>

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

    </form>

</body>

</html>

<?php

session\_start();  // Start the session

// Check if the user is logged in

if (!isset($\_SESSION['username'])) {

    header("Location: login.php");

    exit;

}

// Retrieve the username from the session

$username = $\_SESSION['username'];

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Home</title>

</head>

<body>

    <h2>Welcome, <?php echo htmlspecialchars($username); ?>!</h2>

    <p>You are logged in.</p>

    <a href="logout.php">Logout</a>

</body>

</html>

<?php

session\_start();  // Start the session

// Destroy all session data

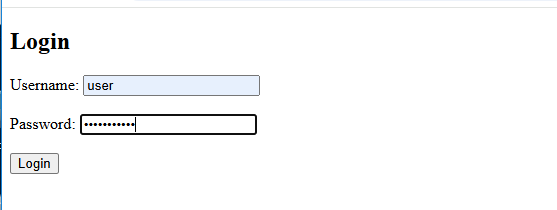
session\_unset();

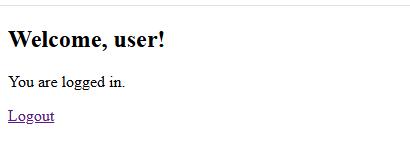
session\_destroy();

// Redirect to login page after logout

header("Location: login.php");

exit;

****

****

**QUESTION 36 => WRITE A PHP SCRIPT TO CRAETE A FROM WITH FILEDS FOR “FIRST NAME “ AND “LAST NAME” . WHEN FORM IS SUBMITTED, DISPLAY A GTREETING MESSAGE THAT INCLUDES USER’S FIRST AND LAST NAME**

<?php

/\* Write a PHp script to create a form with fields for "First Name " and " Last Name " .

When the form is submitted , display a greeting message that includes the user's first and last name \*/

// Check if the form is submitted via POST method

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

    // Get the first name and last name from the form

    $firstName = htmlspecialchars($\_POST['first\_name']);

    $lastName = htmlspecialchars($\_POST['last\_name']);

    // Display the greeting message with the user's first and last name

    echo "<h2>Hello, $firstName $lastName! Welcome to our site.</h2>";

} else {

    // If the form is not submitted yet, show the form

    ?>

    <!DOCTYPE html>

    <html lang="en">

    <head>

        <meta charset="UTF-8">

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

        <title>Greeting Form</title>

    </head>

    <body>

        <h2>Please enter your details:</h2>

        <form action="pn3.php" method="POST">

            <label for="first\_name">First Name:</label>

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

            <label for="last\_name">Last Name:</label>

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

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

        </form>

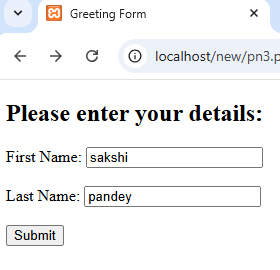
    </body>

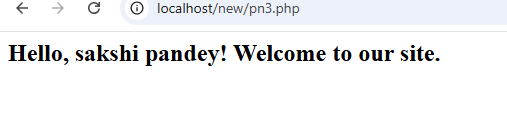
    </html>

    <?php

}

?>

****

****

**QUESTION 37 => WRITE A PHP SCRIPT TO CREATE A SIMPLE CACULATOR WITH FORM AND DISPLAY THE RESULT**

<?php

/\* Write a PHp script to create a simple calculator form with two fields

for " Number1 " and  "Number2 " and a drop-down to select an operation

(+, -, \*, /) . when the form is submitted, display the result of the

selected operation \*/

// Initialize result variable

$result = null;

// Check if the form is submitted

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

    // Get the input values from the form

    $number1 = $\_POST['number1'];

    $number2 = $\_POST['number2'];

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

    // Validate that inputs are numeric

    if (is\_numeric($number1) && is\_numeric($number2)) {

        // Perform the selected operation

        switch ($operation) {

            case '+':

                $result = $number1 + $number2;

                break;

            case '-':

                $result = $number1 - $number2;

                break;

            case '\*':

                $result = $number1 \* $number2;

                break;

            case '/':

                if ($number2 != 0) {

                    $result = $number1 / $number2;

                } else {

                    $result = "Error: Division by zero.";

                }

                break;

            default:

                $result = "Invalid operation.";

                break;

        }

    } else {

        $result = "Please enter valid numbers.";

    }

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Simple Calculator</title>

</head>

<body>

    <h2>Simple Calculator</h2>

    <form action="pn4.php" method="POST">

        <label for="number1">Number 1:</label>

        <input type="text" name="number1" id="number1" required><br><br>

        <label for="number2">Number 2:</label>

        <input type="text" name="number2" id="number2" required><br><br>

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

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

            <option value="+">+</option>

            <option value="-">-</option>

            <option value="\*">\*</option>

            <option value="/">/</option>

        </select><br><br>

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

    </form>

    <?php

    // Display the result after form submission

    if ($result !== null) {

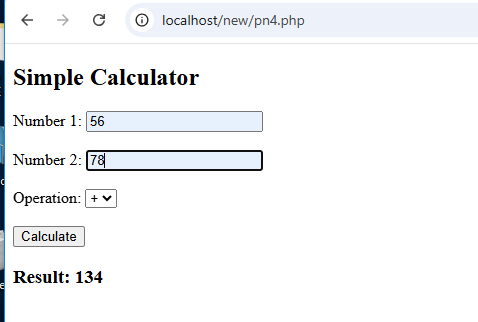
        echo "<h3>Result: $result</h3>";

    }

    ?>

</body>

</html>

****

**QUESTION 38 => WRITE A PHP SCRIPT TO CREATE A USER REGISTRATION**

<?php

/\* Write  a PHp script to create a user registration form with fields for

" Username ", "Password ", " Confirm Password ".

When the form is submitted , check that the password and confirm password match.

If they match , display a success message, otherwise display an error  \*/

// Initialize the result message

$message = "";

// Check if the form is submitted

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

    // Get the form values

    $username = $\_POST['username'];

    $password = $\_POST['password'];

    $confirmPassword = $\_POST['confirm\_password'];

    // Check if password and confirm password match

    if ($password === $confirmPassword) {

        // Passwords match, display success message

        $message = "Registration successful! Welcome, $username.";

    } else {

        // Passwords don't match, display error message

        $message = "Error: Password and Confirm Password do not match.";

    }

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>User Registration</title>

</head>

<body>

    <h2>User Registration</h2>

    <!-- Display message after form submission -->

    <?php

    if ($message !== "") {

        echo "<p style='color: " . (strpos($message, "Error") === false ? "green" : "red") . ";'>$message</p>";

    }

    ?>

    <!-- Registration Form -->

    <form action="pn5.php" method="POST">

        <label for="username">Username:</label>

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

        <label for="password">Password:</label>

        <input type="password" name="password" id="password" required><br><br>

        <label for="confirm\_password">Confirm Password:</label>

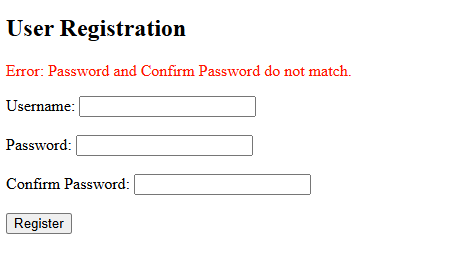
        <input type="password" name="confirm\_password" id="confirm\_password" required><br><br>

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

    </form>

</body>

</html>

****

**QUESTION 39 => WRITE A PHP SCRIPT TO HANDLE DIVIDE BY ZERO EXCEPTION**

<?php

/\* Write a PHP script to create a simple calculator that divides two numbers

entered by the user. If the user tries to divide by zero, catch the exception

and display an error message \*/

// Initialize result variable and error message

$result = null;

$errorMessage = "";

// Check if the form is submitted

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

    // Get the input values from the form

    $number1 = $\_POST['number1'];

    $number2 = $\_POST['number2'];

    // Check if the input values are numeric

    if (is\_numeric($number1) && is\_numeric($number2)) {

        try {

            // Attempt to perform the division

            if ($number2 == 0) {

                throw new Exception("Error: Division by zero is not allowed.");

            } else {

                $result = $number1 / $number2;

            }

        } catch (Exception $e) {

            // Catch the exception and store the error message

            $errorMessage = $e->getMessage();

        }

    } else {

        $errorMessage = "Please enter valid numbers.";

    }

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Simple Calculator with Exception Handling</title>

</head>

<body>

    <h2>Simple Division Calculator</h2>

    <!-- Display error message if there's any -->

    <?php

    if ($errorMessage !== "") {

        echo "<p style='color: red;'>$errorMessage</p>";

    } elseif ($result !== null) {

        echo "<h3>Result: $result</h3>";

    }

    ?>

    <!-- Calculator Form -->

    <form action="pn6.php" method="POST">

        <label for="number1">Number 1:</label>

        <input type="text" name="number1" id="number1" required><br><br>

        <label for="number2">Number 2:</label>

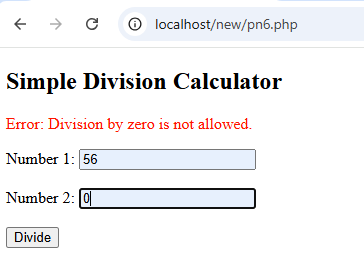
        <input type="text" name="number2" id="number2" required><br><br>

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

    </form>

</body>

</html>

****

**QUESTION 40 => WRITE A PHP SCRIPT FOR CUSTOM EXCEPTION HANDLING**

<?php

/\* Write a PHp script that accepts an age input from user and throws a

custom exception if the age is less than 18. Catch the exception

and dsiplay an error message saying " Age  must be 18 or older " \*/

// Define a custom exception class

class AgeException extends Exception {

    // Custom message for the exception

    public function errorMessage() {

        return "Error: Age must be 18 or older.";

    }

}

// Initialize variables

$age = null;

$errorMessage = "";

// Check if the form is submitted

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

    // Get the age from the form

    $age = $\_POST['age'];

    try {

        // Check if the age is numeric

        if (!is\_numeric($age)) {

            throw new Exception("Error: Please enter a valid number for age.");

        }

        // Throw an exception if age is less than 18

        if ($age < 18) {

            throw new AgeException(); // Custom exception is thrown

        } else {

            $errorMessage = "You are eligible, age is: $age.";

        }

    } catch (AgeException $e) {

        // Catch the custom exception

        $errorMessage = $e->errorMessage();

    } catch (Exception $e) {

        // Catch general exceptions (like invalid number input)

        $errorMessage = $e->getMessage();

    }

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Age Check</title>

</head>

<body>

    <h2>Enter Your Age</h2>

    <!-- Display error or success message -->

    <?php

    if ($errorMessage !== "") {

        echo "<p style='color: red;'>$errorMessage</p>";

    }

    ?>

    <!-- Form to input age -->

    <form action="pn7.php" method="POST">

        <label for="age">Age:</label>

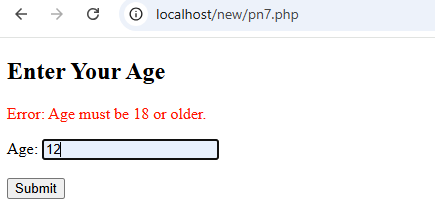
        <input type="text" name="age" id="age" required><br><br>

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

    </form>

</body>

</html>

****

**QUESTION 41 => WRITE A PHP SCRIPT TO CONNECT TO MYSQL DATABASE**

**QUESTION 42 => CREATE DATABSE COMPANY AND TABLE EMPLOYEE**

**QUESTION 43 => PERFORM INSERTION AND DELETION OPERTATION**

<?php

/\* Write a PHP script that connects to a MySQL database named company .

The script should use localhost as the server. root as the username , and an empty password.

display a message indicating whether the  conection was sucess ful. \*/

$servername ="localhost";

$username = "root";

$password = "";

$dbname ="company";

$conn = new mysqli($servername,$username,$password,$dbname);

if($conn->connect\_error){

    die("Connection failed: ".$conn->connect\_error);

}

echo "connected successfully!";

/\*Write php script to inser a new  employee record into the table named employees in the company

The table has the following fields

Insert a new record \*/

$sql = "INSERT INTO employees (id, name\_emp, email,salary) VALUES ('102','user4','user4@example.com',40000)";

if($conn->query($sql) == TRUE){

    echo "New record created successfully";

}

else{

    echo "Error: ".$sql."<br>".$conn->error;

}

/\* display all record of table with column name  \*/

/\* write php script to delete an employee data with id E102 \*/

$sql2 = "DELETE FROM employees  WHERE id ='102' ";

if($conn->query($sql2) == TRUE){

    echo "Record deleted  successfully";

}

else{

    echo "Error: ".$sql."<br>".$conn->error;

}

$sql3 = " SELECT \* FROM employees ORDER BY id";

$result = $conn->query($sql3);

$conn->close();

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Employee details</title>

    <!-- CSS FOR STYLING THE PAGE -->

    <style>

        table {

            margin: 0 auto;

            font-size: large;

            border: 1px solid black;

        }

        h1 {

            text-align: center;

            color: #006600;

            font-size: xx-large;

            font-family: 'Gill Sans', 'Gill Sans MT',

            ' Calibri', 'Trebuchet MS', 'sans-serif';

        }

        td {

            background-color: #E4F5D4;

            border: 1px solid black;

        }

        th,

        td {

            font-weight: bold;

            border: 1px solid black;

            padding: 10px;

            text-align: center;

        }

        td {

            font-weight: lighter;

        }

    </style>

</head>

<body>

    <section>

        <h1>Company</h1>

        <!-- TABLE CONSTRUCTION -->

        <table>

            <tr>

                <th>Employee Id </th>

                <th>Employee Name</th>

                <th>Employee Email</th>

                <th>GFG Articles</th>

            </tr>

            <!-- PHP CODE TO FETCH DATA FROM ROWS -->

            <?php

                // LOOP TILL END OF DATA

                while($rows=$result->fetch\_assoc())

                {

            ?>

            <tr>

                <!-- FETCHING DATA FROM EACH

                    ROW OF EVERY COLUMN -->

                <td><?php echo $rows['id'];?></td>

                <td><?php echo $rows['name\_emp'];?></td>

                <td><?php echo $rows['email'];?></td>

                <td><?php echo $rows['salary'];?></td>

            </tr>

            <?php

                }

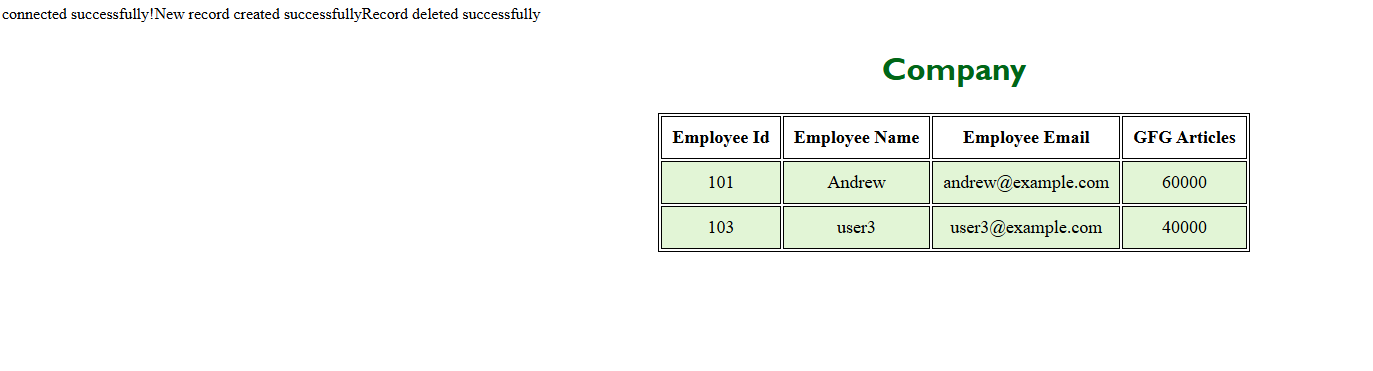
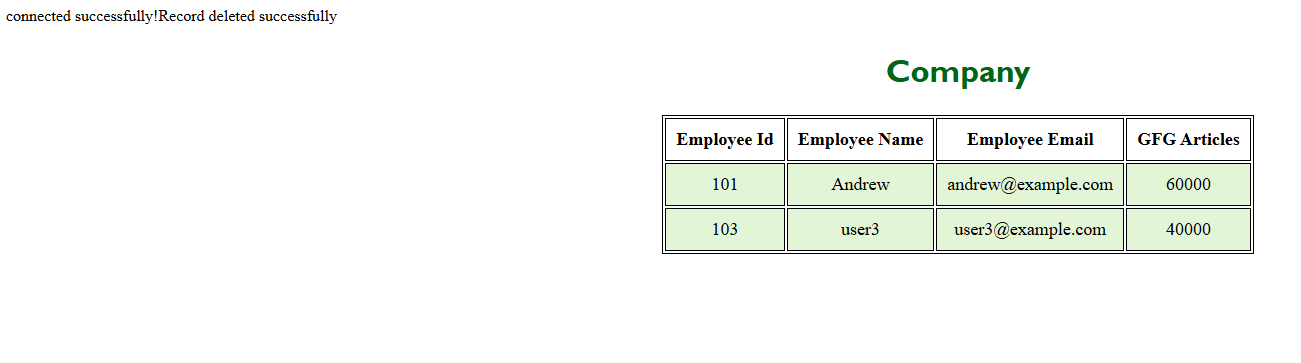
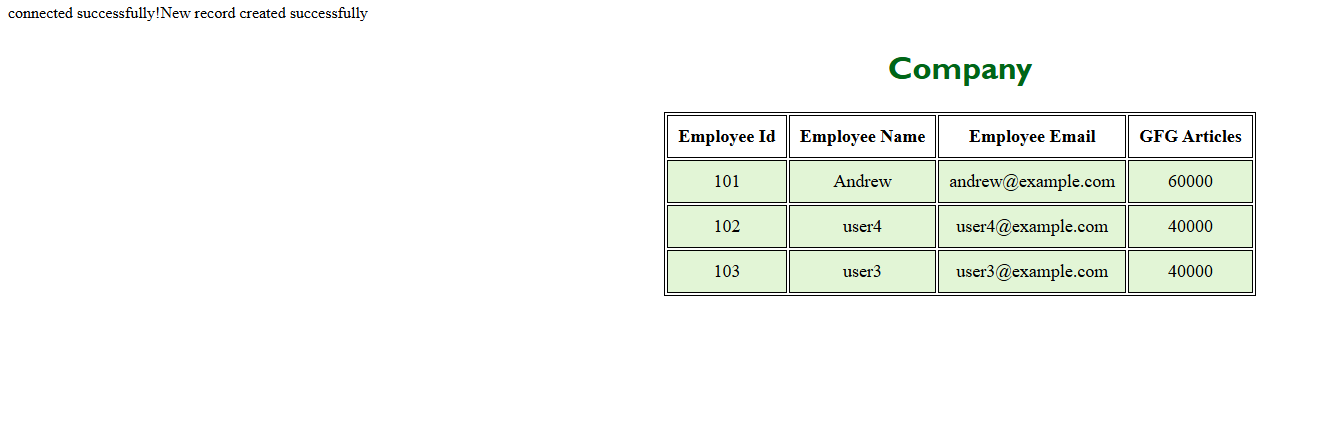
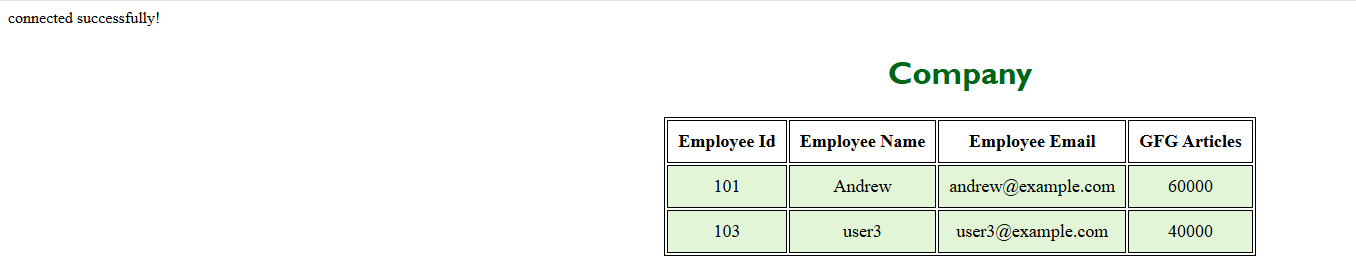
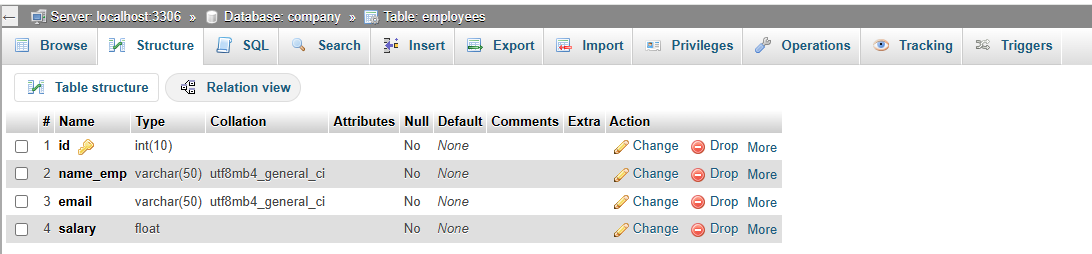
            ?>

        </table>

    </section>

</body>

</html>

****

**QUESTION 44 => WRITE A PHP SCRIPT TO DEMONSTRATE POST AND GET METHOD**

<?php

/\* Explain the difference between  the POST and GET methods in PHP and write a PHP script to demonstrate the use of both methods \*/

// Check if the form was submitted using POST

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

    // Collect POST data

    $name = $\_POST['name'];

    $age = $\_POST['age'];

    echo "<h3>Form submitted using POST:</h3>";

    echo "Name: $name<br>";

    echo "Age: $age<br>";

}

// Check if the form was submitted using GET

if ($\_SERVER["REQUEST\_METHOD"] == "GET" && isset($\_GET['search'])) {

    // Collect GET data

    $search = $\_GET['search'];

    echo "<h3>Search submitted using GET:</h3>";

    echo "Search term: $search<br>";

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>POST and GET Example</title>

</head>

<body>

    <h2>POST Method Example</h2>

    <form method="POST" action="">

        Name: <input type="text" name="name"><br>

        Age: <input type="text" name="age"><br>

        <input type="submit" value="Submit via POST">

    </form>

    <h2>GET Method Example</h2>

    <form method="GET" action="">

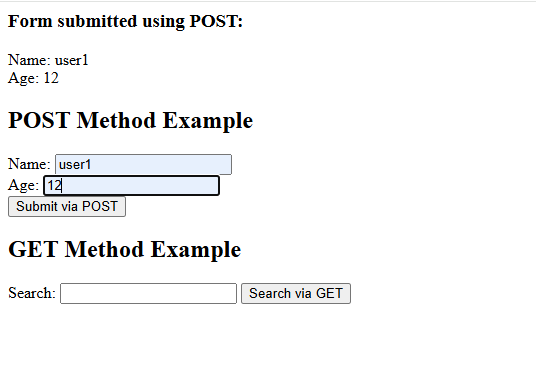
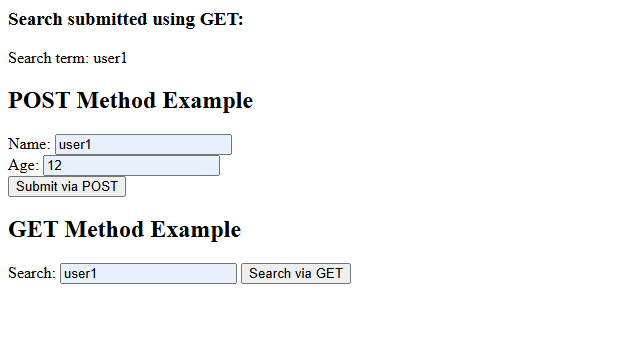
        Search: <input type="text" name="search">

        <input type="submit" value="Search via GET">

    </form>

</body>

</html>



**QUESTION 45 => WRITE PHP SCRIPT TO DEMONSTRATE SYMMETRIC ENCRYPTION AND DECRYPTION**

<?php

/\* Write a PHP script to demonstrate symmetric encryption and decryption \*/

// Encryption function

function encrypt($data, $key) {

    // Generate a random initialization vector (IV) for encryption

    $iv = openssl\_random\_pseudo\_bytes(openssl\_cipher\_iv\_length('aes-256-cbc'));

    // Encrypt the data using AES-256-CBC encryption method

    $encryptedData = openssl\_encrypt($data, 'aes-256-cbc', $key, 0, $iv);

    // Return both the encrypted data and the IV (IV is required for decryption)

    return base64\_encode($encryptedData . '::' . $iv);

}

// Decryption function

function decrypt($data, $key) {

    // Decode the data from base64

    list($encryptedData, $iv) = explode('::', base64\_decode($data), 2);

    // Decrypt the data using AES-256-CBC decryption method

    return openssl\_decrypt($encryptedData, 'aes-256-cbc', $key, 0, $iv);

}

// Example data to be encrypted

$data = "This is a secret message!";

// Secret key for symmetric encryption (it should be 32 bytes for AES-256)

$key = "12345678901234567890123456789012";  // 32 bytes key

// Encrypt the data

$encryptedData = encrypt($data, $key);

echo "Encrypted Data: " . $encryptedData . "\n";

// Decrypt the data

$decryptedData = decrypt($encryptedData, $key);

echo "Decrypted Data: " . $decryptedData . "\n";

?>

