LAB-1

Write a PHP Program to

1.To print “hello world”.

2.Keyword and Variable name sensitivity.

3.Insert a single line and multi-line comments.

4.Define different data types(int, float, string, boolean, array) and get it’s type using var_dump() function.

5.Demonstrate the PHP variables scopes(local, global, static)

6.Demonstrate the implicit and explicit data type conversion.

7.Use the error control operator @ to suppress error messages.

CODING

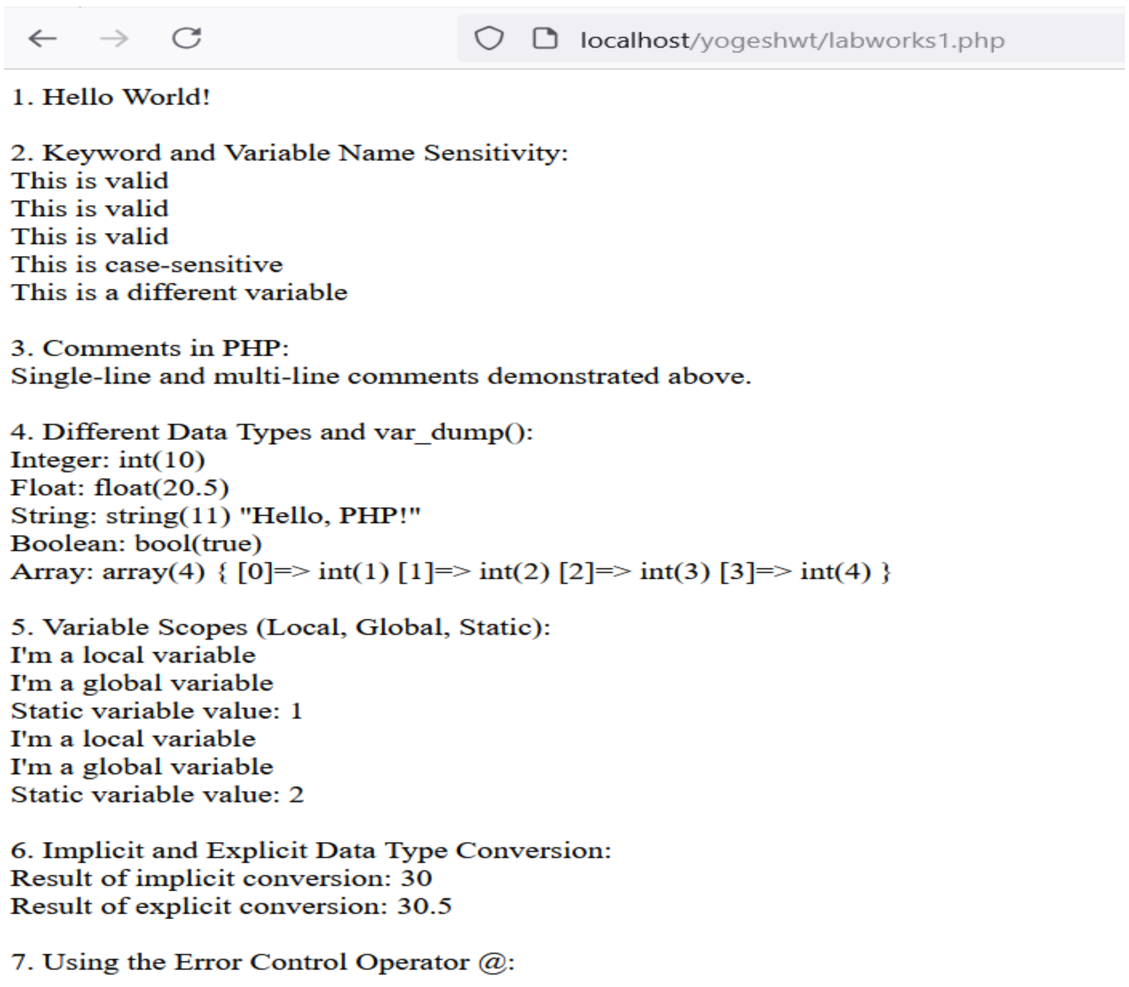
```
<?php

// 1. To print "Hello World"

echo "1. Hello World!<br><br>";

// 2. Keyword and Variable Name Sensitivity

echo "2. Keyword and Variable Name Sensitivity:<br>";

ECHO "This is valid<br>";  // Will print: This is valid

echo "This is valid<br>";  // Will print: This is valid

EcHo "This is valid<br>";  // Will print: This is valid

$Variable = "This is case-sensitive";

$variable = "This is a different variable";

echo $Variable . "<br>";  // Will print: This is case-sensitive

echo $variable . "<br><br>";  // Will print: This is a different variable

// 3. Single-Line and Multi-Line Comments

echo "3. Comments in PHP:<br>";

// This is a single-line comment
```

```
# This is also a single-line comment
/*
This is a multi-line comment.
*/
echo "Single-line and multi-line comments demonstrated above.<br><br>";
// 4. Different Data Types and var_dump()
echo "4. Different Data Types and var_dump():<br>";
$integer = 10;
$float = 20.5;
$string = "Hello, PHP!";
$boolean = true;
$array = array(1, 2, 3, 4);
echo "Integer: ";
var_dump($integer);
echo "<br>";
echo "Float: ";
var_dump($float);
echo "<br>";
echo "String: ";
var_dump($string);
echo "<br>";
echo "Boolean: ";
var_dump($boolean);
echo "<br>";
echo "Array: ";
var_dump($array);
```

```
echo "<br><br>";
// 5. Variable Scopes (Local, Global, Static)
echo "5. Variable Scopes (Local, Global, Static):<br>";
$globalVar = "I'm a global variable";
function testScopes() {
   $localVar = "I'm a local variable";
   echo $localVar . "<br>";
   global $globalVar;
   echo $globalVar . "<br>";
   static $staticVar = 0;
   $staticVar++;
   echo "Static variable value: $staticVar<br>";
}
testScopes();
testScopes();
echo "<br>";
// 6. Implicit and Explicit Data Type Conversion
echo "6. Implicit and Explicit Data Type Conversion:<br>";
$var1 = "10";
$var2 = 20;
$result = $var1 + $var2; // Implicit conversion
echo "Result of implicit conversion: $result<br>"; // Output: 30
$var3 = "30.5";
$convertedVar = (float)$var3; // Explicit conversion
echo "Result of explicit conversion: $convertedVar<br><br>"; // Output: 30.5
// 7. Error Control Operator
```

```
echo "7. Using the Error Control Operator @:<br>";
echo @ $undefinedVariable?>
```

OUTPUT



localhost/yogeshwt/labworks1.php

```
1. Hello World!

2. Keyword and Variable Name Sensitivity:
This is valid
This is valid
This is valid
This is case-sensitive
This is a different variable

3. Comments in PHP:
Single-line and multi-line comments demonstrated above.

4. Different Data Types and var_dump():
Integer: int(10)
Float: float(20.5)
String: string(11) "Hello, PHP!"
Boolean: bool(true)
Array: array(4) { [0]=> int(1) [1]=> int(2) [2]=> int(3) [3]=> int(4) }

5. Variable Scopes (Local, Global, Static):
I'm a local variable
I'm a global variable
Static variable value: 1
I'm a local variable
I'm a global variable
Static variable value: 2

6. Implicit and Explicit Data Type Conversion:
Result of implicit conversion: 30
Result of explicit conversion: 30.5

7. Using the Error Control Operator @:
```

LAB-2

Write a PHP Program to

8.Check if a given number is odd or even using if statement.

9.Check which number is greater among three using nested if else statement.

10.Starting from Sunday=1, Monday=2, … , Saturday =7, use elseif ladder to echo the days of week.

11.Starting from Sunday=1, Monday=2, … , Saturday =7, use switch-case to echo the days of week.

12.Write a switch-case program to determine the season(winter, spring, summer, fall) according to the months using common code block.

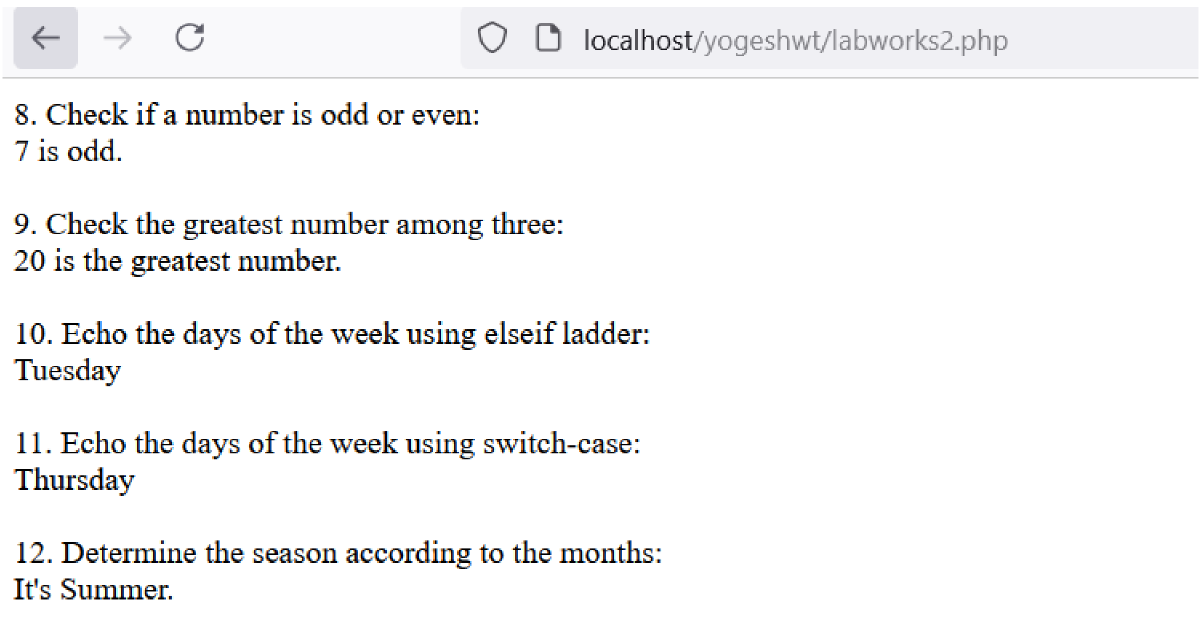
CODING

```php
<?php
// 8. Check if a given number is odd or even using if
statement
echo "8. Check if a number is odd or even:<br>";
$number = 7;
if ($number % 2 == 0) {
    echo "$number is even.<br><br>";
} else {
    echo "$number is odd.<br><br>";
}

// 9. Check which number is greater among three using
nested if else statement
echo "9. Check the greatest number among three:<br>";
$num1 = 10;
$num2 = 20;
$num3 = 15;

if ($num1 > $num2 && $num1 > $num3) {
    echo "$num1 is the greatest number.<br><br>";
} elseif ($num2 > $num1 && $num2 > $num3) {
    echo "$num2 is the greatest number.<br><br>";
```

```
} else {
    echo "$num3 is the greatest number.<br><br>";
}

// 10. Use elseif ladder to echo the days of the week
echo "10. Echo the days of the week using elseif
ladder:<br>";
$day = "Tuesday";
if ($day == "Sunday") {
    echo "Sunday<br><br>";
} elseif ($day == "Monday") {
    echo "Monday<br><br>";
} elseif ($day == "Tuesday") {
    echo "Tuesday<br><br>";
} elseif ($day == "Wednesday") {
    echo "Wednesday<br><br>";
} elseif ($day == "Thursday") {
    echo "Thursday<br><br>";
} elseif ($day == "Friday") {
    echo "Friday<br><br>";
} elseif ($day == "Saturday") {
    echo "Saturday<br><br>";
} else {
    echo "Invalid day name!<br><br>";
}

// 11. Use switch-case to echo the days of the week
echo "11. Echo the days of the week using switch-
case:<br>";
$day = "Thursday";
switch ($day) {
    case "Sunday":
        echo "Sunday<br><br>";
        break;
    case "Monday":
        echo "Monday<br><br>";
        break;
    case "Tuesday":
        echo "Tuesday<br><br>";
        break;
    case "Wednesday":
        echo "Wednesday<br><br>";
        break;
    case "Thursday":
        echo "Thursday<br><br>";
        break;
    case "Friday":
        echo "Friday<br><br>";
        break;
    case "Saturday":
        echo "Saturday<br><br>";
        break;
    default:
        echo "Invalid day name!<br><br>";
}

// 12. Switch-case program to determine the season
according to the months using a common code block
echo "12. Determine the season according to the
months:<br>";
$month = "August";
switch ($month) {
    case "December":
    case "January":
    case "February":
        echo "It's Winter.<br><br>";
        break;
    case "March":
    case "April":
    case "May":
        echo "It's Spring.<br><br>";
        break;
    case "June":
    case "July":
    case "August":
        echo "It's Summer.<br><br>";
        break;
    case "September":
    case "October":
    case "November":
        echo "It's Fall.<br><br>";
        break;
    default:
        echo "Invalid month name!<br><br>";
```

```
}
?>
```

OUTPUT



```
8. Check if a number is odd or even:
7 is odd.

9. Check the greatest number among three:
20 is the greatest number.

10. Echo the days of the week using elseif ladder:
Tuesday

11. Echo the days of the week using switch-case:
Thursday

12. Determine the season according to the months:
It's Summer.
```

LAB-3

Write a PHP Program to

13.Use For Loop to find

- a.Factorial of a Number
- b.Sum of odd numbers between 0-100.
- c.Generate a Fibonacci series up to 15th Terms.
- d.Program to create a table based on the number of rows and columns given. Values 1,2,3,4… are shown on the table cell.

14.Use While and do-while loop to find

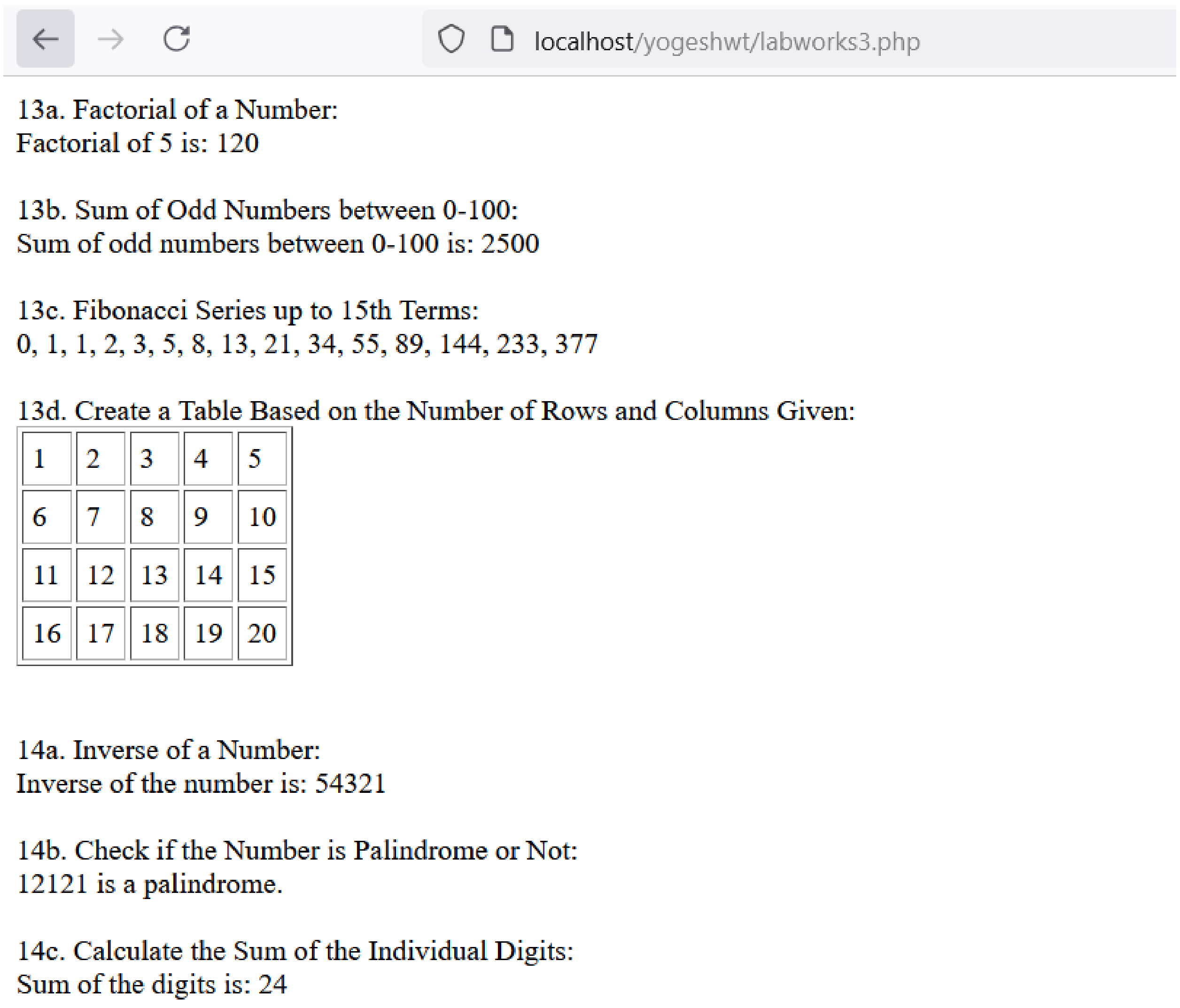
- a.Inverse of a number
- b.Check if the number is palindrome or not
- c.Calculate the sum of the individual digit

CODING

```php
<?php
// 13. Use For Loop to find:

// a. Factorial of a Number
echo "13a. Factorial of a Number:<br>";
$number = 5;
$factorial = 1;
for ($i = 1; $i <= $number; $i++) {
    $factorial *= $i;
}
echo "Factorial of $number is: $factorial<br><br>";

// b. Sum of odd numbers between 0-100
echo "13b. Sum of Odd Numbers between 0-100:<br>";
$sum = 0;
for ($i = 1; $i <= 100; $i += 2) {
    $sum += $i;
}
echo "Sum of odd numbers between 0-100 is: $sum<br><br>";
```

```php
// c. Generate a Fibonacci series up to 15th terms
echo "13c. Fibonacci Series up to 15th Terms:<br>";
$n1 = 0;
$n2 = 1;
echo "$n1, $n2";
for ($i = 3; $i <= 15; $i++) {
    $n3 = $n1 + $n2;
    echo ", $n3";
    $n1 = $n2;
    $n2 = $n3;
}
echo "<br><br>";

// d. Program to create a table based on the number of
rows and columns given
echo "13d. Create a Table Based on the Number of Rows and
Columns Given:<br>";
$rows = 4;
$columns = 5;
$count = 1;

echo "<table border='1' cellpadding='5'>";
for ($i = 1; $i <= $rows; $i++) {
    echo "<tr>";
    for ($j = 1; $j <= $columns; $j++) {
        echo "<td>$count</td>";
        $count++;
    }
    echo "</tr>";
}
echo "</table><br><br>";

// 14. Use While and do-while loop to find:

// a. Inverse of a number
echo "14a. Inverse of a Number:<br>";
$number = 12345;
$inverse = 0;
while ($number > 0) {
    $remainder = $number % 10;
    $inverse = ($inverse * 10) + $remainder;
    $number = (int)($number / 10);
}
echo "Inverse of the number is: $inverse<br><br>";

// b. Check if the number is palindrome or not
echo "14b. Check if the Number is Palindrome or Not:<br>";
$originalNumber = 12121;
$number = $originalNumber;
$reverse = 0;

while ($number > 0) {
    $remainder = $number % 10;
    $reverse = ($reverse * 10) + $remainder;
    $number = (int)($number / 10);
}

if ($originalNumber == $reverse) {
    echo "$originalNumber is a palindrome.<br><br>";
} else {
    echo "$originalNumber is not a palindrome.<br><br>";
}

// c. Calculate the sum of the individual digits
echo "14c. Calculate the Sum of the Individual
Digits:<br>";
$number = 987;
$sum = 0;
do {
    $sum += $number % 10;
    $number = (int)($number / 10);
} while ($number > 0);

echo "Sum of the digits is: $sum<br><br>";
?>
```

OUTPUT



```
13a. Factorial of a Number:
Factorial of 5 is: 120

13b. Sum of Odd Numbers between 0-100:
Sum of odd numbers between 0-100 is: 2500

13c. Fibonacci Series up to 15th Terms:
0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377

13d. Create a Table Based on the Number of Rows and Columns Given:
```

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20

```
14a. Inverse of a Number:
Inverse of the number is: 54321

14b. Check if the Number is Palindrome or Not:
12121 is a palindrome.

14c. Calculate the Sum of the Individual Digits:
Sum of the digits is: 24
```

LAB-4

Write a PHP Program to

15.Use a foreach loop to parse the element of an array and display their values.

16.Use a goto operator to defined label and print it's output as *"the label is reached".*

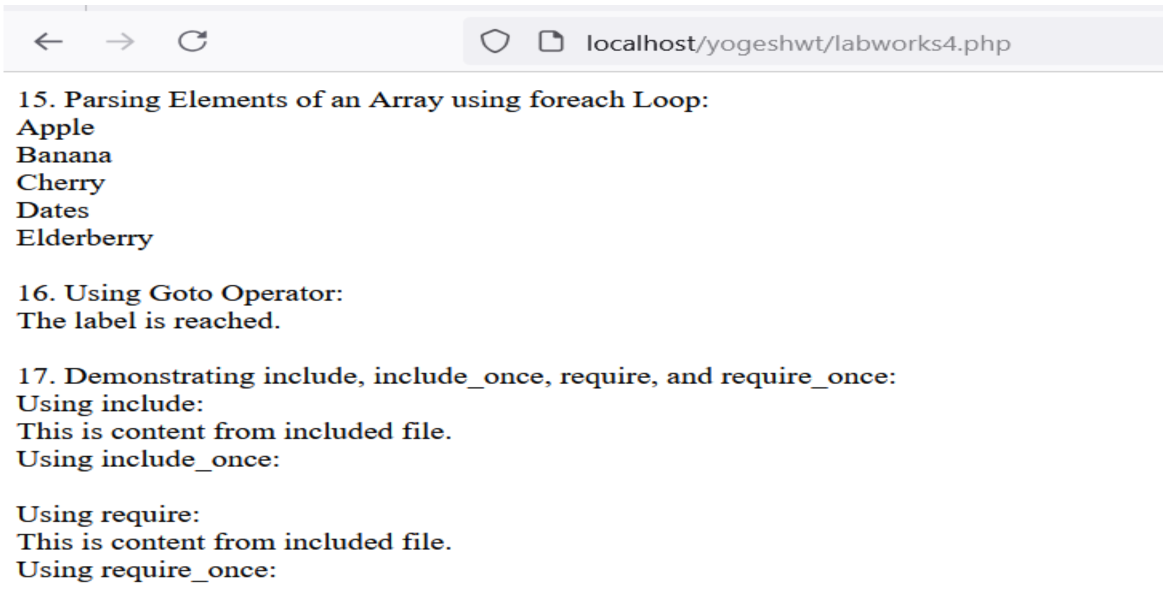
17.Use including files and display that content. Try include, include_one, require, require_once and their corresponding output.

CODING

```
<?php
// 15. Use a foreach loop to parse the elements of an
array and display their values
echo "15. Parsing Elements of an Array using foreach
Loop:<br>";
$array = array("Apple", "Banana", "Cherry", "Dates",
"Elderberry");
foreach ($array as $fruit) {
    echo "$fruit<br>";
}
echo "<br>";
// 16. Use a goto operator to define a label and print its
output as "the label is reached"
echo "16. Using Goto Operator:<br>";
goto label;
echo "This will be skipped.<br>";
label:
echo "The label is reached.<br><br>";
// 17. Use include, include_once, require, require_once
and display the content
echo "17. Demonstrating include, include_once, require,
and require_once:<br>";
// Create content for the included file (normally, this
file would be separate)
```

```php
$filename = "sample.php";
$content = "<?php echo 'This is content from included
file.'; ?>";
file_put_contents($filename, $content);

// Using include
echo "Using include:<br>";
include 'sample.php';
echo "<br>";

// Using include_once
echo "Using include_once:<br>";
include_once 'sample.php';
echo "<br>";

// Using require
echo "Using require:<br>";
require 'sample.php';
echo "<br>";

// Using require_once
echo "Using require_once:<br>";
require_once 'sample.php';
echo "<br><br>";

// Clean up the file (optional)
unlink($filename);
?>
```

OUTPUT



```
15. Parsing Elements of an Array using foreach Loop:
Apple
Banana
Cherry
Dates
Elderberry

16. Using Goto Operator:
The label is reached.

17. Demonstrating include, include_once, require, and require_once:
Using include:
This is content from included file.
Using include_once:

Using require:
This is content from included file.
Using require_once:
```

LAB-5

Write a PHP Program to

18.Create a user-defined function to

- **a.Echo "user defined function".**
- **b.Pass a two numbers as an argument and display it's sum.**
- **c.Call by reference which appends the message to the referenced variable.**
- **d.Pass a default argument to a function.**
- **e.Check if the number is prime or not.**
- **f.Calculate the area of a circle, rectangle.**

19.Function that accepts variable length arguments as a parameter and returns sums all the number passed as arguments.
20.Create a dynamic function call to find the *gcd and factorial* of a number recursively.

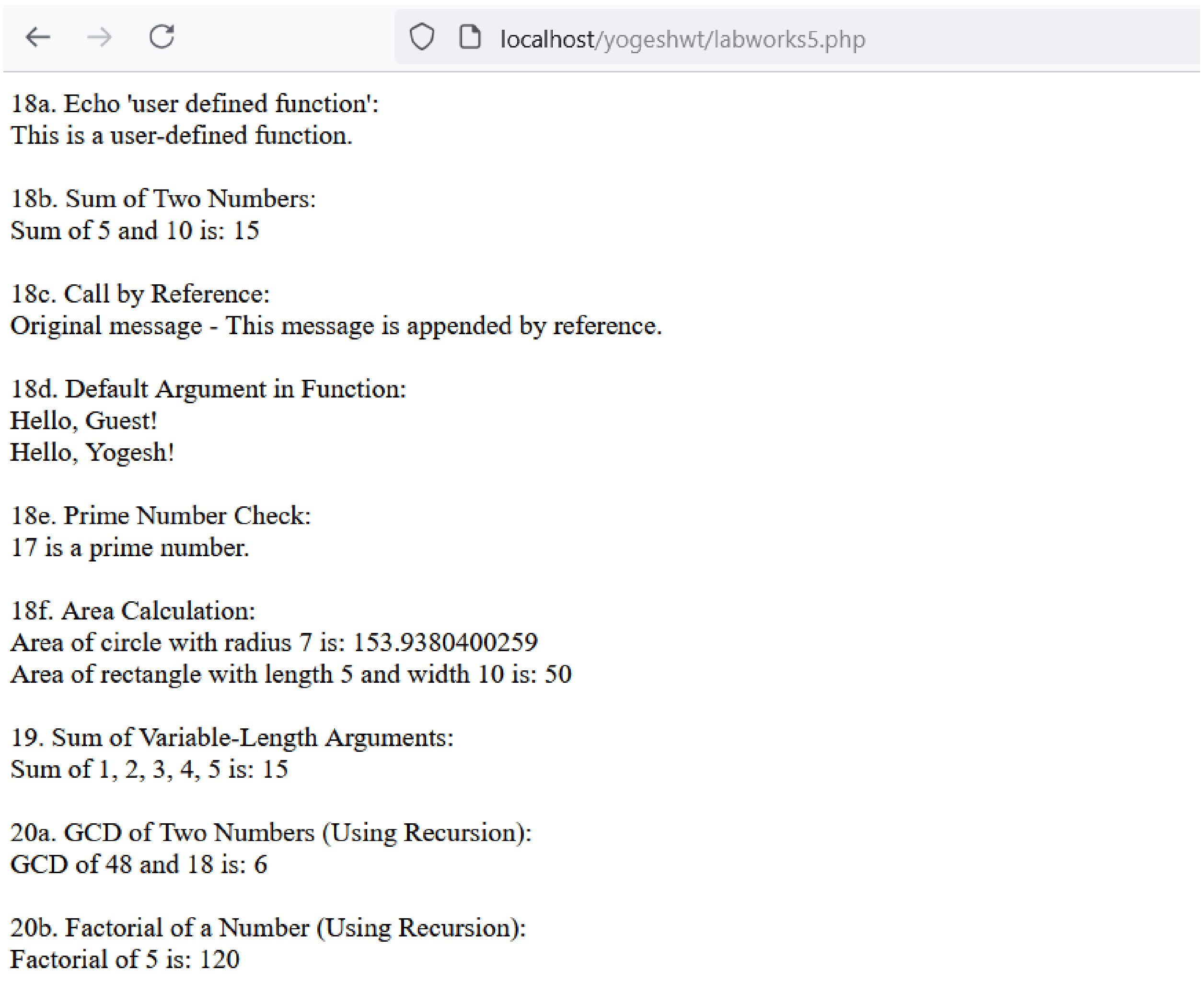
CODING

```php
<?php
// 18. Create a user-defined function:

// a. Echo "user defined function".
echo "18a. Echo 'user defined function':<br>";
function myFunction() {
    echo "This is a user-defined function.<br><br>";
}
myFunction();

// b. Pass two numbers as arguments and display their sum.
echo "18b. Sum of Two Numbers:<br>";
function addNumbers($a, $b) {
    return $a + $b;
}
echo "Sum of 5 and 10 is: " . addNumbers(5, 10) .
"<br><br>";

// c. Call by reference which appends the message to the
referenced variable.
echo "18c. Call by Reference:<br>";
```

```php
function appendMessage(&$message) {
    $message .= " - This message is appended by 
reference.";
}
$msg = "Original message";
appendMessage($msg);
echo $msg . "<br><br>";

// d. Pass a default argument to a function.
echo "18d. Default Argument in Function:<br>";
function greet($name = "Guest") {
    echo "Hello, $name!<br>";
}
greet(); // Uses default value
greet("Yogesh"); // Passes an explicit value
echo "<br>";

// e. Check if the number is prime or not.
echo "18e. Prime Number Check:<br>";
function isPrime($number) {
    if ($number < 2) {
        return false;
    }
    for ($i = 2; $i <= sqrt($number); $i++) {
        if ($number % $i == 0) {
            return false;
        }
    }
    return true;
}
$number = 17;
if (isPrime($number)) {
    echo "$number is a prime number.<br><br>";
} else {
    echo "$number is not a prime number.<br><br>";
}

// f. Calculate the area of a circle, rectangle.
echo "18f. Area Calculation:<br>";
function areaOfCircle($radius) {
    return pi() * $radius * $radius;
}
```

```
function areaOfRectangle($length, $width) {
    return $length * $width;
}
echo "Area of circle with radius 7 is: " . areaOfCircle(7)
. "<br>";
echo "Area of rectangle with length 5 and width 10 is: " .
areaOfRectangle(5, 10) . "<br><br>";

// 19. Function that accepts variable-length arguments as
a parameter and returns the sum of all numbers passed as
arguments.
echo "19. Sum of Variable-Length Arguments:<br>";
function sumOfNumbers(...$numbers) {
    return array_sum($numbers);
}
echo "Sum of 1, 2, 3, 4, 5 is: " . sumOfNumbers(1, 2, 3,
4, 5) . "<br><br>";

// 20. Create a dynamic function call to find the GCD and
factorial of a number recursively:

// a. Recursive function to find GCD of two numbers
echo "20a. GCD of Two Numbers (Using Recursion):<br>";
function gcd($a, $b) {
    if ($b == 0) {
        return $a;
    }
    return gcd($b, $a % $b);
}
echo "GCD of 48 and 18 is: " . gcd(48, 18) . "<br><br>";

// b. Recursive function to find factorial of a number
echo "20b. Factorial of a Number (Using Recursion):<br>";
function factorial($n) {
    if ($n <= 1) {
        return 1;
    }
    return $n * factorial($n - 1);
}
echo "Factorial of 5 is: " . factorial(5) . "<br><br>";

?>
```

OUTPUT



```
18a. Echo 'user defined function':
This is a user-defined function.

18b. Sum of Two Numbers:
Sum of 5 and 10 is: 15

18c. Call by Reference:
Original message - This message is appended by reference.

18d. Default Argument in Function:
Hello, Guest!
Hello, Yogesh!

18e. Prime Number Check:
17 is a prime number.

18f. Area Calculation:
Area of circle with radius 7 is: 153.9380400259
Area of rectangle with length 5 and width 10 is: 50

19. Sum of Variable-Length Arguments:
Sum of 1, 2, 3, 4, 5 is: 15

20a. GCD of Two Numbers (Using Recursion):
GCD of 48 and 18 is: 6

20b. Factorial of a Number (Using Recursion):
Factorial of 5 is: 120
```

LAB-6

Write a PHP Program to

18.Create a user-defined function to

- a.Echo “user defined function”.
- b.Pass a two numbers as an argument and display it’s sum.
- c.Call by reference which appends the message to the referenced variable.
- d.Pass a default argument to a function.
- e.Check if the number is prime or not.
- f.Calculate the area of a circle, rectangle.

19.Function that accepts variable length arguments as a parameter and returns sums all the number passed as arguments.
20.Create a dynamic function call to find the *gcd and factorial* of a number recursively.

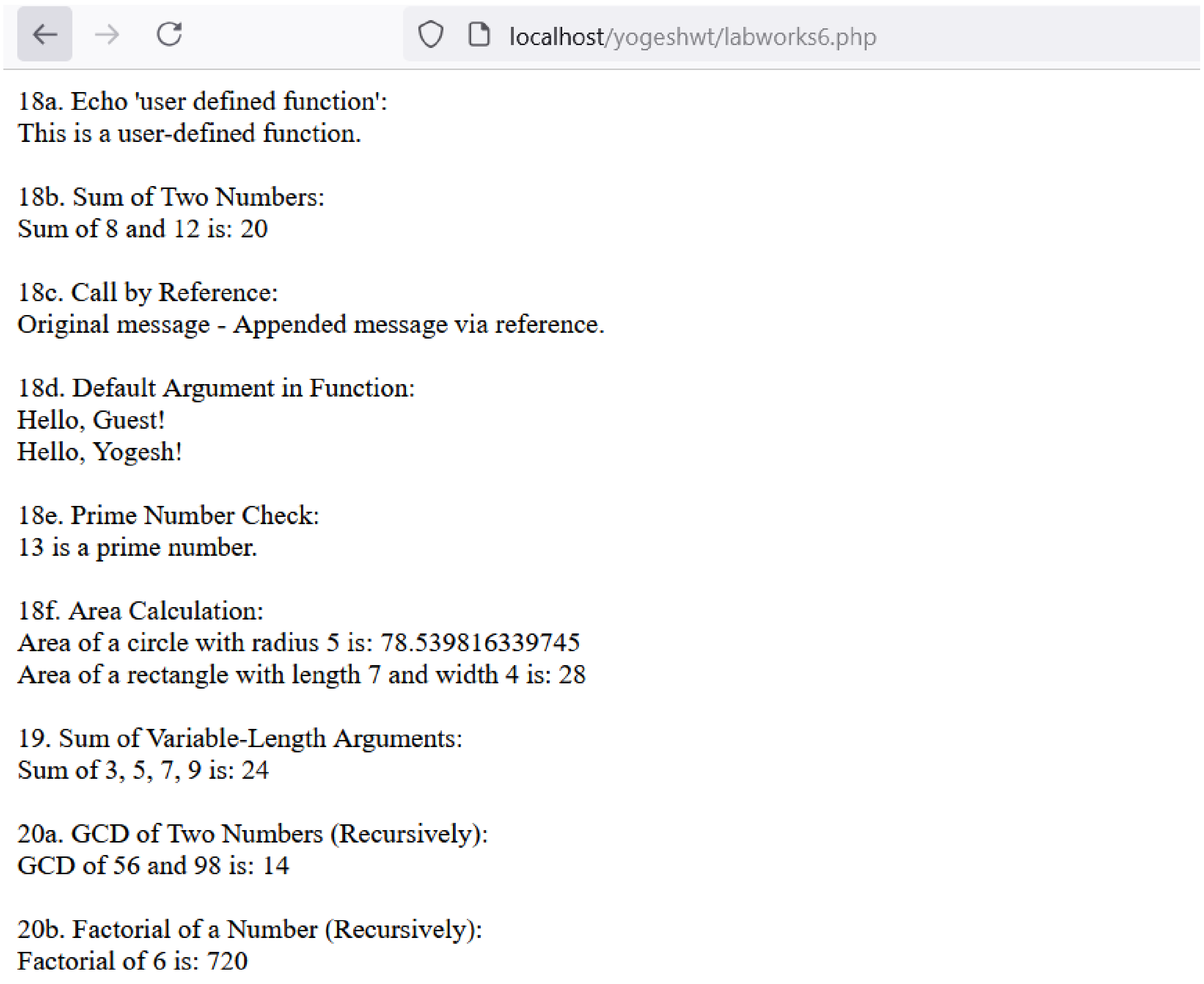
CODING

```
<?php
// 18. Create a user-defined function:
// a. Echo "user defined function".
echo "18a. Echo 'user defined function':<br>";
function displayMessage() {
    echo "This is a user-defined function.<br><br>";
}
displayMessage();
// b. Pass two numbers as arguments and display their sum.
echo "18b. Sum of Two Numbers:<br>";
function add($num1, $num2) {
    return $num1 + $num2;
}
echo "Sum of 8 and 12 is: " . add(8, 12) . "<br><br>";
// c. Call by reference which appends the message to the
referenced variable.
echo "18c. Call by Reference:<br>";
function appendToMessage(&$message) {
    $message .= " - Appended message via reference.";
}
```

```
$text = "Original message";
appendToMessage($text);
echo $text . "<br><br>";
// d. Pass a default argument to a function.
echo "18d. Default Argument in Function:<br>";
function greet($name = "Guest") {
    echo "Hello, $name!<br>";
}
greet(); // Uses default argument
greet("Yogesh"); // Passes explicit value
echo "<br>";
// e. Check if the number is prime or not.
echo "18e. Prime Number Check:<br>";
function isPrime($number) {
    if ($number < 2) return false;
    for ($i = 2; $i <= sqrt($number); $i++) {
        if ($number % $i == 0) return false;
    }
    return true;
}
$testNumber = 13;
if (isPrime($testNumber)) {
    echo "$testNumber is a prime number.<br><br>";
} else {
    echo "$testNumber is not a prime number.<br><br>";
}

// f. Calculate the area of a circle, rectangle.
echo "18f. Area Calculation:<br>";
function areaOfCircle($radius) {
    return pi() * $radius * $radius;
}
function areaOfRectangle($length, $width) {
    return $length * $width;
}
echo "Area of a circle with radius 5 is: " .
areaOfCircle(5) . "<br>";
echo "Area of a rectangle with length 7 and width 4 is: "
. areaOfRectangle(7, 4) . "<br><br>";
```

```
// 19. Function that accepts variable-length arguments as
a parameter and returns the sum of all numbers passed as
arguments.
echo "19. Sum of Variable-Length Arguments:<br>";
function sumOfNumbers(...$numbers) {
    return array_sum($numbers);
}
echo "Sum of 3, 5, 7, 9 is: " . sumOfNumbers(3, 5, 7, 9) .
"<br><br>";

// 20. Dynamic function call to find GCD and factorial of
a number recursively:

// a. Recursive function to find GCD of two numbers
echo "20a. GCD of Two Numbers (Recursively):<br>";
function gcd($a, $b) {
    if ($b == 0) return $a;
    return gcd($b, $a % $b);
}
echo "GCD of 56 and 98 is: " . gcd(56, 98) . "<br><br>";

// b. Recursive function to find factorial of a number
echo "20b. Factorial of a Number (Recursively):<br>";
function factorial($n) {
    if ($n <= 1) return 1;
    return $n * factorial($n - 1);
}
echo "Factorial of 6 is: " . factorial(6) . "<br><br>";
?>
```

OUTPUT



localhost/yogeshwt/labworks6.php

```
18a. Echo 'user defined function':
This is a user-defined function.

18b. Sum of Two Numbers:
Sum of 8 and 12 is: 20

18c. Call by Reference:
Original message - Appended message via reference.

18d. Default Argument in Function:
Hello, Guest!
Hello, Yogesh!

18e. Prime Number Check:
13 is a prime number.

18f. Area Calculation:
Area of a circle with radius 5 is: 78.539816339745
Area of a rectangle with length 7 and width 4 is: 28

19. Sum of Variable-Length Arguments:
Sum of 3, 5, 7, 9 is: 24

20a. GCD of Two Numbers (Recursively):
GCD of 56 and 98 is: 14

20b. Factorial of a Number (Recursively):
Factorial of 6 is: 720
```

LAB-7

Write a PHP Program to

21.Write a function

a.to check if the number is palindrome or not?
b.Armstrong number or not?
c.Reverse or not?
d.Positive or negative?
e.Sum of individual digit
f.Root of a quadratic equation.
g.Check if a given year is leap year or not?

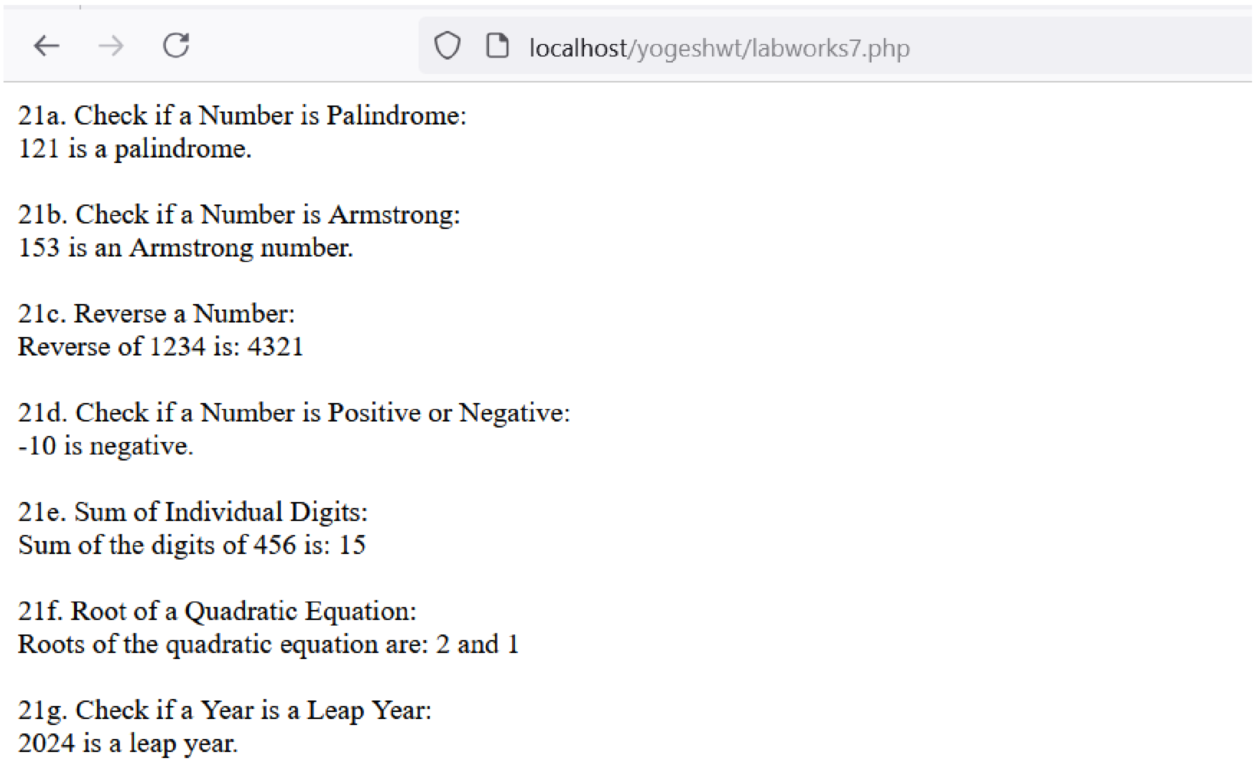
CODING

```
<?php
// 21. Functions to perform various tasks
// a. Check if a number is palindrome or not
echo "21a. Check if a Number is Palindrome:<br>";
function isPalindrome($number) {
   $originalNumber = $number;
   $reversedNumber = 0;
   while ($number > 0) {
      $remainder = $number % 10;
      $reversedNumber = ($reversedNumber * 10) + $remainder;
      $number = (int)($number / 10);
   }
   return $originalNumber == $reversedNumber;
}
$testNumber = 121;
if (isPalindrome($testNumber)) {
    echo "$testNumber is a palindrome.<br><br>";
} else {
    echo "$testNumber is not a palindrome.<br><br>";
}
// b. Check if a number is Armstrong or not
echo "21b. Check if a Number is Armstrong:<br>";
function isArmstrong($number) {
    $sum = 0;
    $digits = strlen($number);
    $temp = $number;
    while ($temp > 0) {
        $digit = $temp % 10;
        $sum += pow($digit, $digits);
        $temp = (int)($temp / 10);
    }
    return $sum == $number;
}
$testNumber = 153;
if (isArmstrong($testNumber)) {
    echo "$testNumber is an Armstrong number.<br><br>";
} else {
    echo "$testNumber is not an Armstrong number.<br><br>";
}
```

```
// c. Reverse a number
echo "21c. Reverse a Number:<br>";
function reverseNumber($number) {
    $reversedNumber = 0;
    while ($number > 0) {
        $remainder = $number % 10;
        $reversedNumber = ($reversedNumber * 10) + $remainder;
        $number = (int)($number / 10);
    }
    return $reversedNumber;
}
$testNumber = 1234;
echo "Reverse of $testNumber is: " . reverseNumber($testNumber) .
"<br><br>";
// d. Check if a number is positive or negative
echo "21d. Check if a Number is Positive or Negative:<br>";
function checkSign($number) {
    if ($number > 0) {
        return "positive";
    } elseif ($number < 0) {
        return "negative";
    } else {
        return "zero";
    }
}
$testNumber = -10;
```

```
echo "$testNumber is " . checkSign($testNumber) . ".<br><br>";
// e. Sum of individual digits
echo "21e. Sum of Individual Digits:<br>";
function sumOfDigits($number) {
    $sum = 0;
    while ($number > 0) {
        $sum += $number % 10;
        $number = (int)($number / 10);
    }
    return $sum;
}
$testNumber = 456;
echo "Sum of the digits of $testNumber is: " . sumOfDigits($testNumber) .
"<br><br>";
// f. Root of a quadratic equation
echo "21f. Root of a Quadratic Equation:<br>";
function quadraticRoots($a, $b, $c) {
    $discriminant = ($b * $b) - (4 * $a * $c);
    $root1 = $root2 = null;
    if ($discriminant > 0) {
        $root1 = (-$b + sqrt($discriminant)) / (2 * $a);
        $root2 = (-$b - sqrt($discriminant)) / (2 * $a);
        return array($root1, $root2);
    } elseif ($discriminant == 0) {
        $root1 = $root2 = -$b / (2 * $a);
        return array($root1);
    } else {
        return "No real roots";
    }
}
list($root1, $root2) = quadraticRoots(1, -3, 2);
echo "Roots of the quadratic equation are: $root1 and $root2<br><br>";
// g. Check if a given year is a leap year or not
echo "21g. Check if a Year is a Leap Year:<br>";
function isLeapYear($year) {
    return ($year % 4 == 0 && $year % 100 != 0) || ($year % 400 == 0);
}
$testYear = 2024;
if (isLeapYear($testYear)) {
    echo "$testYear is a leap year.<br><br>";
} else {
    echo "$testYear is not a leap year.<br><br>";
}
?>
```

<u>**OUTPUT**</u>



```
21a. Check if a Number is Palindrome:
121 is a palindrome.

21b. Check if a Number is Armstrong:
153 is an Armstrong number.

21c. Reverse a Number:
Reverse of 1234 is: 4321

21d. Check if a Number is Positive or Negative:
-10 is negative.

21e. Sum of Individual Digits:
Sum of the digits of 456 is: 15

21f. Root of a Quadratic Equation:
Roots of the quadratic equation are: 2 and 1

21g. Check if a Year is a Leap Year:
2024 is a leap year.
```

LAB-8

23. Display the following output:

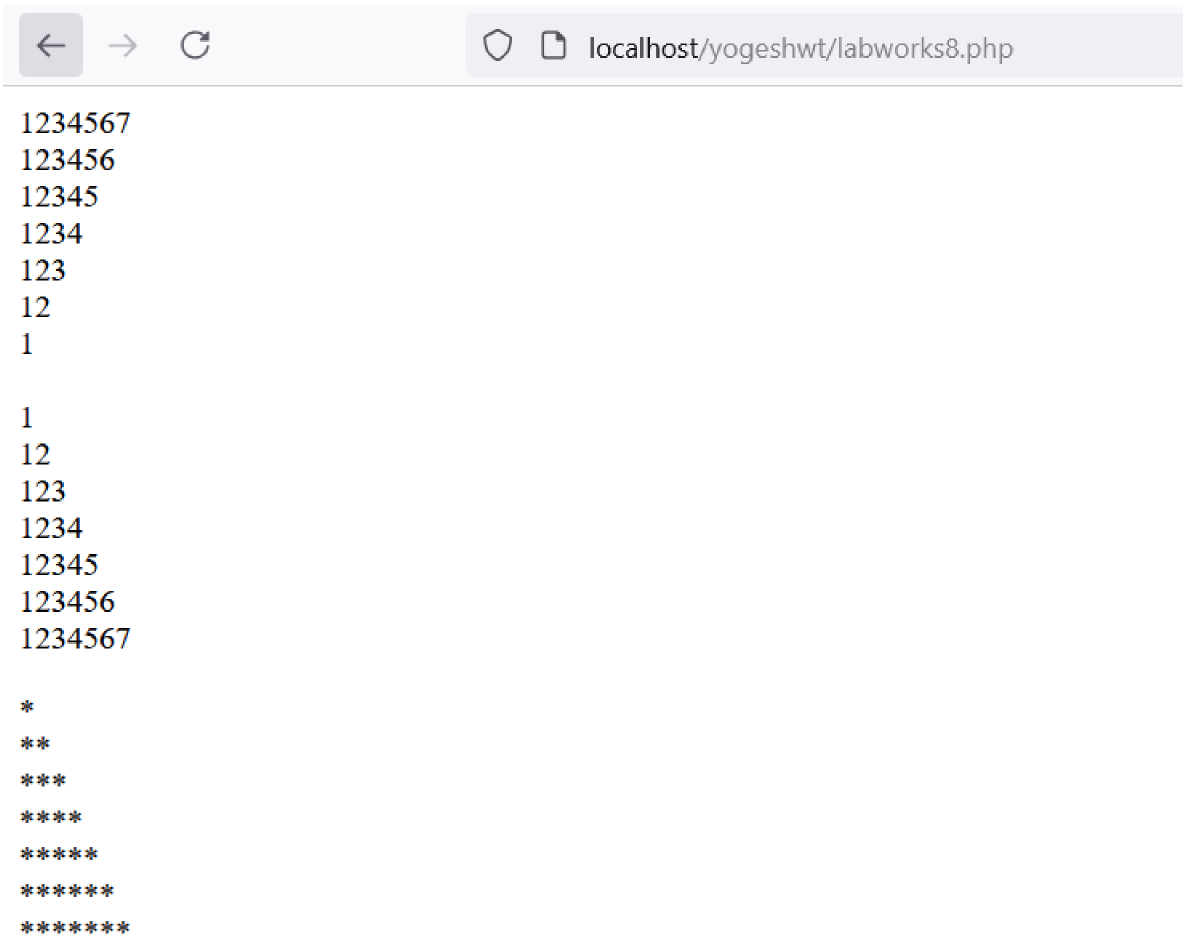
```
1234567         1           *
123456          12          **
12345           123         ***
1234            1234        ****
123             12345       *****
12              123456      ******
1               1234567     *******
```

CODING

```
<?php
// First column pattern
for ($i = 7; $i >= 1; $i--) {
    for ($j = 1; $j <= $i; $j++) {
        echo $j;
    }
    echo "<br>";
}
echo "<br>";
// Second column pattern
for ($i = 1; $i <= 7; $i++) {
    for ($j = 1; $j <= $i; $j++) {
        echo $j;
    }
    echo "<br>";
}
```

```php
echo "<br>";
// Third column pattern
for ($i = 1; $i <= 7; $i++) {
    for ($j = 1; $j <= $i; $j++) {
        echo "*";
    }
    echo "<br>";
}
?>
```

OUTPUT



```
1234567
123456
12345
1234
123
12
1

1
12
123
1234
12345
123456
1234567

*
**
***
****
*****
******
*******
```

LAB-9

1.Declare any string and echo it’s output.

2.Echo single quoted and double quoted strings.

3.Take a user’s name as input and display a greeting message using string concatenation.

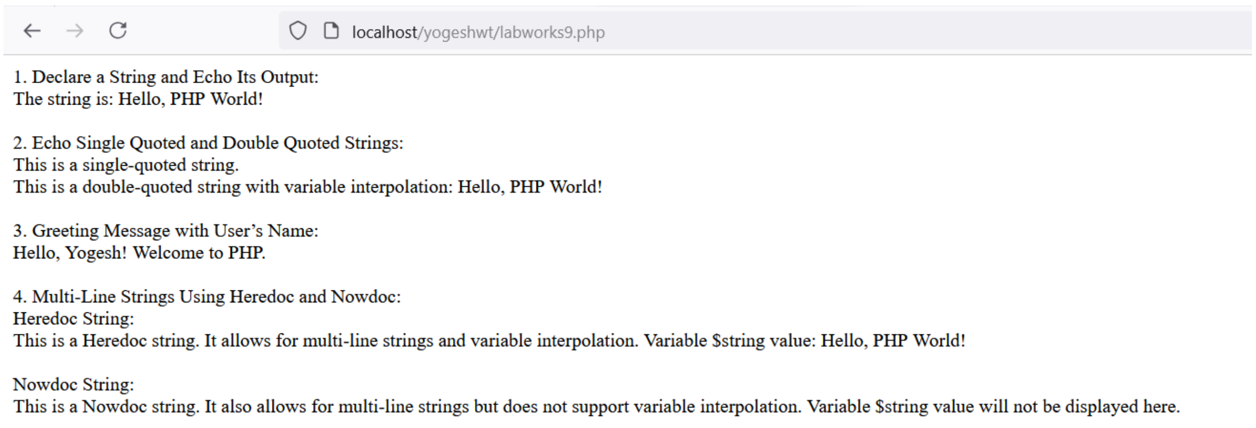
4.Define a multi-line strings using a *heredoc* and *nowdoc* and also mention the variable interpolation in them.

CODING

```
<?php
// 1. Declare a string and echo its output.
echo "1. Declare a String and Echo Its Output:<br>";
$string = "Hello, PHP World!";
echo "The string is: $string<br><br>";

// 2. Echo single quoted and double quoted strings.
echo "2. Echo Single Quoted and Double Quoted
Strings:<br>";
$singleQuoted = 'This is a single-quoted string.';
$doubleQuoted = "This is a double-quoted string with
variable interpolation: $string";
echo "$singleQuoted<br>";
echo "$doubleQuoted<br><br>";

// 3. Take a user’s name as input and display a greeting
message using string concatenation.
echo "3. Greeting Message with User’s Name:<br>";
// Simulate user input for demonstration
$userName = "Yogesh";
$greeting = "Hello, " . $userName . "! Welcome to PHP.";
echo "$greeting<br><br>";
```

```php
// 4. Define multi-line strings using heredoc and nowdoc,
and mention the variable interpolation.
echo "4. Multi-Line Strings Using Heredoc and
Nowdoc:<br>";

// Heredoc syntax (variable interpolation)
$heredocString = <<<EOD
This is a Heredoc string.
It allows for multi-line strings and variable
interpolation.
Variable \$string value: $string
EOD;

echo "Heredoc String:<br>$heredocString<br><br>";

// Nowdoc syntax (no variable interpolation)
$nowdocString = <<<'EOD'
This is a Nowdoc string.
It also allows for multi-line strings but does not support
variable interpolation.
Variable $string value will not be displayed here.
EOD;

echo "Nowdoc String:<br>$nowdocString<br><br>";
?>
```

OUTPUT



```
1. Declare a String and Echo Its Output:
The string is: Hello, PHP World!

2. Echo Single Quoted and Double Quoted Strings:
This is a single-quoted string.
This is a double-quoted string with variable interpolation: Hello, PHP World!

3. Greeting Message with User’s Name:
Hello, Yogesh! Welcome to PHP.

4. Multi-Line Strings Using Heredoc and Nowdoc:
Heredoc String:
This is a Heredoc string. It allows for multi-line strings and variable interpolation. Variable $string value: Hello, PHP World!

Nowdoc String:
This is a Nowdoc string. It also allows for multi-line strings but does not support variable interpolation. Variable $string value will not be displayed here.
```

LAB-10

5.Define a string and use the following string functions.

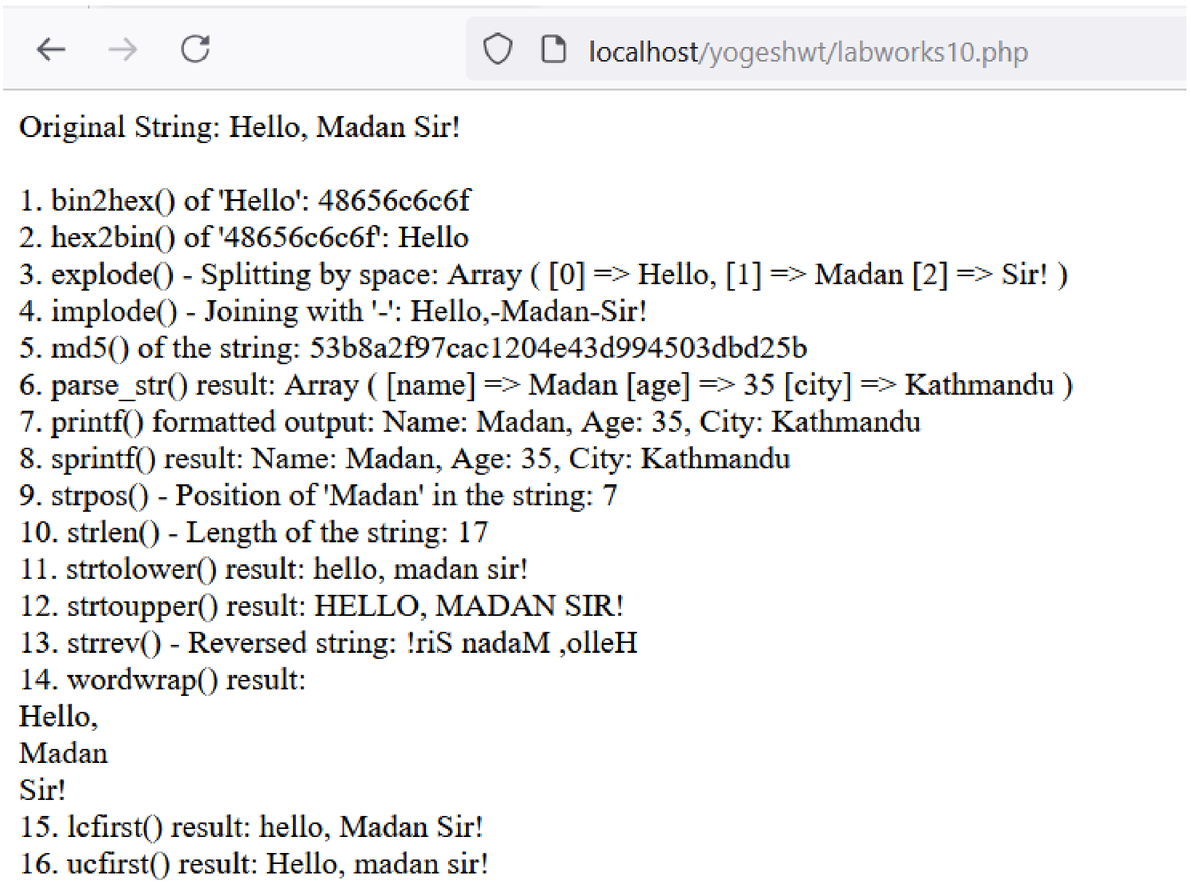
1. bin2hex()
2. hex2bin()
3. explode()
4. implode()
5. md5()
6. parse_str()
7. printf()
8. sprintf()
9. strpos()
10. strlen()
11. strtolower()
12. strtoupper()
13. strrev()
14. wordwrap()
15. lcfirst()
16. ucfirst()

CODING

```
<?php
$string = "Hello, Madan Sir!";
echo "Original String: $string<br><br>";
$binaryData = "Hello";
$hex = bin2hex($binaryData);
echo "1. bin2hex() of 'Hello': $hex<br>";
$binary = hex2bin($hex);
echo "2. hex2bin() of '$hex': $binary<br>";
$array = explode(" ", $string);
echo "3. explode() - Splitting by space: ";
print_r($array);
echo "<br>";
$joinedString = implode("-", $array);
echo "4. implode() - Joining with '-': $joinedString<br>";
$md5Hash = md5($string);
echo "5. md5() of the string: $md5Hash<br>";
$queryString = "name=Madan&age=35&city=Kathmandu";
parse_str($queryString, $outputArray);
```

```php
echo "6. parse_str() result: ";
print_r($outputArray);
echo "<br>";
echo "7. printf() formatted output: ";
printf("Name: %s, Age: %d, City: %s<br>",
$outputArray['name'], $outputArray['age'],
$outputArray['city']);
$formattedString = sprintf("Name: %s, Age: %d, City: %s",
$outputArray['name'], $outputArray['age'],
$outputArray['city']);
echo "8. sprintf() result: $formattedString<br>";
$position = strpos($string, "Madan");
echo "9. strpos() - Position of 'Madan' in the string:
$position<br>";
$length = strlen($string);
echo "10. strlen() - Length of the string: $length<br>";
$lowercaseString = strtolower($string);
echo "11. strtolower() result: $lowercaseString<br>";
$uppercaseString = strtoupper($string);
echo "12. strtoupper() result: $uppercaseString<br>";
$reversedString = strrev($string);
echo "13. strrev() - Reversed string:
$reversedString<br>";
$wrappedString = wordwrap($string, 8, "<br>");
echo "14. wordwrap() result:<br>$wrappedString<br>";
$lowerFirst = lcfirst($string);
echo "15. lcfirst() result: $lowerFirst<br>";
$upperFirst = ucfirst(strtolower($string));
echo "16. ucfirst() result: $upperFirst<br>";

?>
```

OUTPUT



```
Original String: Hello, Madan Sir!

1. bin2hex() of 'Hello': 48656c6c6f
2. hex2bin() of '48656c6c6f': Hello
3. explode() - Splitting by space: Array ( [0] => Hello, [1] => Madan [2] => Sir! )
4. implode() - Joining with '-': Hello,-Madan-Sir!
5. md5() of the string: 53b8a2f97cac1204e43d994503dbd25b
6. parse_str() result: Array ( [name] => Madan [age] => 35 [city] => Kathmandu )
7. printf() formatted output: Name: Madan, Age: 35, City: Kathmandu
8. sprintf() result: Name: Madan, Age: 35, City: Kathmandu
9. strpos() - Position of 'Madan' in the string: 7
10. strlen() - Length of the string: 17
11. strtolower() result: hello, madan sir!
12. strtoupper() result: HELLO, MADAN SIR!
13. strrev() - Reversed string: !riS nadaM ,olleH
14. wordwrap() result:
Hello,
Madan
Sir!
15. lcfirst() result: hello, Madan Sir!
16. ucfirst() result: Hello, madan sir!
```

LAB-11

6.To make a indexed array of your favorite fruits and display them using foreach loop.

7.Insert a person details using associative array to store the information’s (name, age, email) and display them using foreach loop.

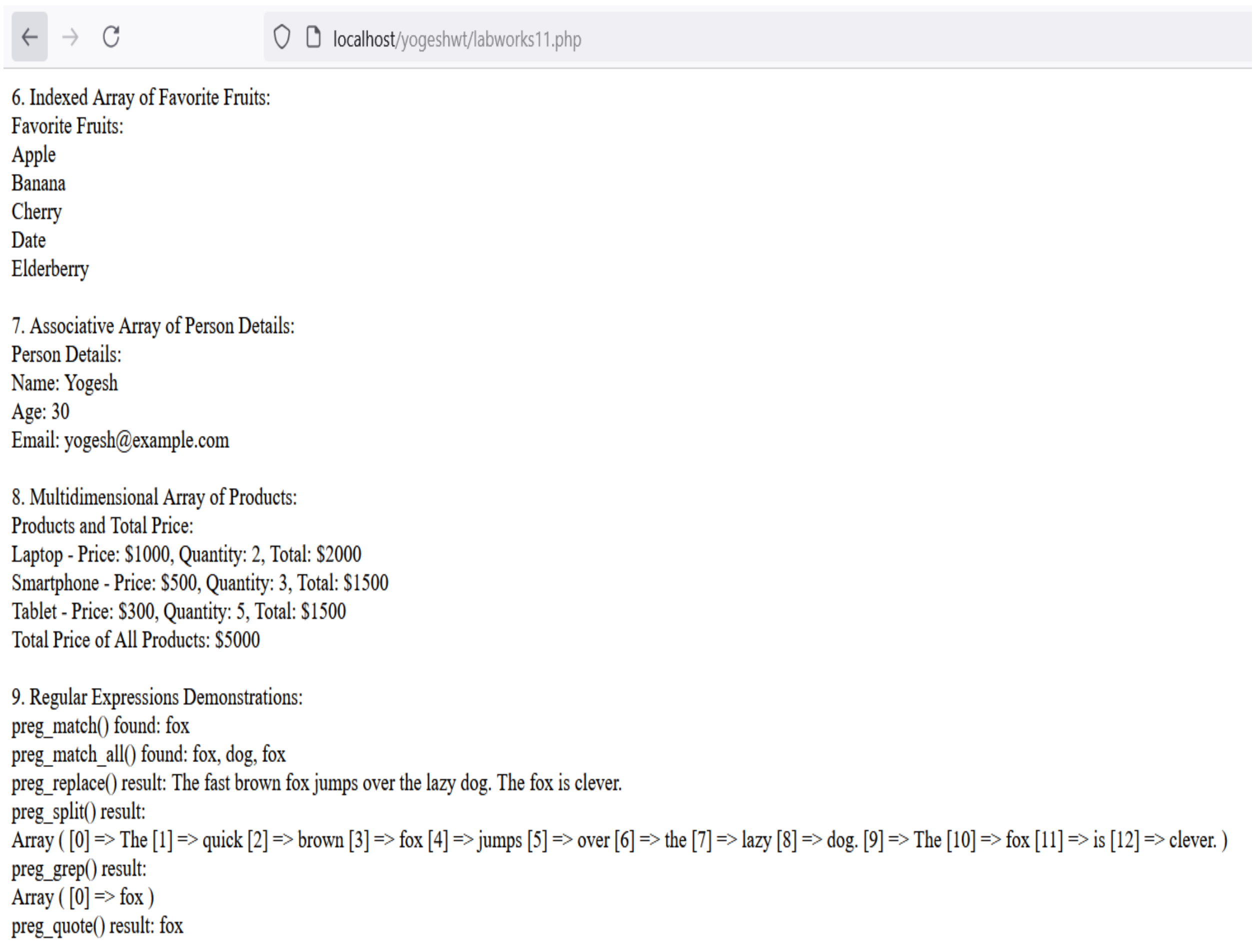
8.Create a multidimensional array to store product prices and quantities and calculate the total prices.
9.To demonstrate the regular expressions preg_match(), preg_match_all(), preg_replace(), preg_split(), preg_grep(), preg_quote()

CODING

```
<?php
// 6. Indexed Array of Favorite Fruits and Display Using
Foreach Loop
echo "6. Indexed Array of Favorite Fruits:<br>";
$favoriteFruits = array("Apple", "Banana", "Cherry",
"Date", "Elderberry");

echo "Favorite Fruits:<br>";
foreach ($favoriteFruits as $fruit) {
    echo "$fruit<br>";
}
echo "<br>";

// 7. Associative Array to Store Person Details and
Display Using Foreach Loop
echo "7. Associative Array of Person Details:<br>";
$personDetails = array(
    "name" => "Yogesh",
    "age" => 30,
    "email" => "yogesh@example.com"
);

echo "Person Details:<br>";
foreach ($personDetails as $key => $value) {
    echo ucfirst($key) . ": $value<br>";
}
echo "<br>";

// 8. Multidimensional Array to Store Product Prices and
Quantities and Calculate Total Prices
echo "8. Multidimensional Array of Products:<br>";
$products = array(
    array("name" => "Laptop", "price" => 1000, "quantity"
=> 2),
    array("name" => "Smartphone", "price" => 500,
"quantity" => 3),
    array("name" => "Tablet", "price" => 300, "quantity"
=> 5)
);

$totalPrice = 0;

echo "Products and Total Price:<br>";
foreach ($products as $product) {
    $productTotal = $product["price"] *
$product["quantity"];
    $totalPrice += $productTotal;
    echo "{$product['name']} - Price:
\${$product['price']}, Quantity: {$product['quantity']},
Total: \$$productTotal<br>";
}

echo "Total Price of All Products: \$$totalPrice<br><br>";

// 9. Regular Expressions Demonstrations
echo "9. Regular Expressions Demonstrations:<br>";

$subject = "The quick brown fox jumps over the lazy dog.
The fox is clever.";
```

```
$pattern = "/fox/";
$patternAll = "/fox|dog/";
$patternReplace = "/quick/";
$replacement = "fast";
$patternSplit = "/\s/";
$patternGrep = "/\bfox\b/";
$patternQuote = preg_quote("fox", "/");
if (preg_match($pattern, $subject, $matches)) {
    echo "preg_match() found: " . implode(', ', $matches)
. "<br>";
} else {
    echo "preg_match() found nothing.<br>";
}
if (preg_match_all($patternAll, $subject, $matchesAll)) {
    echo "preg_match_all() found: " . implode(', ',
$matchesAll[0]) . "<br>";
} else {
    echo "preg_match_all() found nothing.<br>";
}
$replacedString = preg_replace($patternReplace,
$replacement, $subject);
echo "preg_replace() result: $replacedString<br>";
$splittedArray = preg_split($patternSplit, $subject);
echo "preg_split() result:<br>";
print_r($splittedArray);
echo "<br>";
$strings = array("fox", "dog", "cat", "foxes");
$filteredStrings = preg_grep($patternGrep, $strings);
echo "preg_grep() result:<br>";
print_r($filteredStrings);
echo "<br>";
echo "preg_quote() result: " . $patternQuote . "<br><br>";
?>
```

OUTPUT


localhost/yogeshwt/labworks11.php

```
6. Indexed Array of Favorite Fruits:
Favorite Fruits:
Apple
Banana
Cherry
Date
Elderberry

7. Associative Array of Person Details:
Person Details:
Name: Yogesh
Age: 30
Email: yogesh@example.com

8. Multidimensional Array of Products:
Products and Total Price:
Laptop - Price: $1000, Quantity: 2, Total: $2000
Smartphone - Price: $500, Quantity: 3, Total: $1500
Tablet - Price: $300, Quantity: 5, Total: $1500
Total Price of All Products: $5000

9. Regular Expressions Demonstrations:
preg_match() found: fox
preg_match_all() found: fox, dog, fox
preg_replace() result: The fast brown fox jumps over the lazy dog. The fox is clever.
preg_split() result:
Array ( [0] => The [1] => quick [2] => brown [3] => fox [4] => jumps [5] => over [6] => the [7] => lazy [8] => dog. [9] => The [10] => fox [11] => is [12] => clever. )
preg_grep() result:
Array ( [0] => fox )
preg_quote() result: fox
```

LAB-12

10.To match the email type regular expressions.

11.Match the following password format

- Passwords should contain alphabets,
- numbers,
- at least one special character and
- At least 8 digit long.

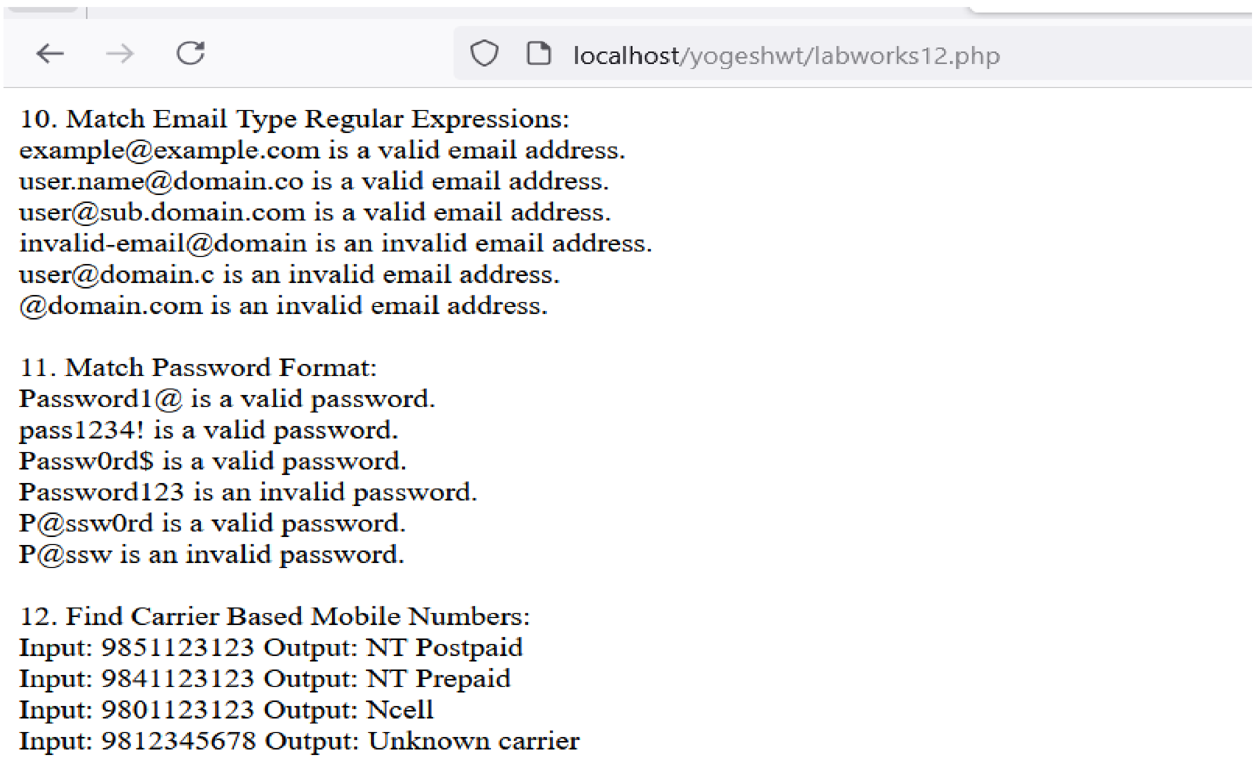
12.To find out the carrier based mobile numbers.

- Input: 9851123123 output: NT Postpaid
- Input: 9841123123 output: NT Prepaid
- Input: 9801123123 output: Ncell

CODING

```php
<?php
// 10. Match Email Type Regular Expressions
echo "10. Match Email Type Regular Expressions:<br>";
$emailPatterns = array(
    "example@example.com",
    "user.name@domain.co",
    "user@sub.domain.com",
    "invalid-email@domain",
    "user@domain.c",
    "@domain.com"
);
foreach ($emailPatterns as $email) {
    if (preg_match("/^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-
]+\.[a-zA-Z]{2,}$/", $email)) {
        echo "$email is a valid email address.<br>";
    } else {
        echo "$email is an invalid email address.<br>";
    }
}
echo "<br>";
// 11. Match Password Format
echo "11. Match Password Format:<br>";
$passwordPatterns = array(
    "Password1@",
    "pass1234!",
    "Passw0rd$",
    "Password123",
    "P@ssw0rd",
    "P@ssw"
);
foreach ($passwordPatterns as $password) {
    if (preg_match("/^(?=.*[A-Za-
z])(?=.*\d)(?=.*[@$!%*?&])[A-Za-z\d@$!%*?&]{8,}$/",
$password)) {
        echo "$password is a valid password.<br>";
    } else {
        echo "$password is an invalid password.<br>";
    }
}
echo "<br>";

// 12. Find Carrier Based Mobile Numbers
echo "12. Find Carrier Based Mobile Numbers:<br>";
function identifyCarrier($mobileNumber) {
    if (preg_match("/^985[0-9]{7}$/", $mobileNumber)) {
        return "NT Postpaid";
    } elseif (preg_match("/^984[0-9]{7}$/",
$mobileNumber)) {
        return "NT Prepaid";
    } elseif (preg_match("/^980[0-9]{7}$/",
$mobileNumber)) {
        return "Ncell";
    } else {
        return "Unknown carrier";
    }
}
$mobileNumbers = array(
    "9851123123",
    "9841123123",
    "9801123123",
    "9812345678"
);
foreach ($mobileNumbers as $number) {
    echo "Input: $number Output: " .
identifyCarrier($number) . "<br>";
}
?>
```

OUTPUT



```
10. Match Email Type Regular Expressions:
example@example.com is a valid email address.
user.name@domain.co is a valid email address.
user@sub.domain.com is a valid email address.
invalid-email@domain is an invalid email address.
user@domain.c is an invalid email address.
@domain.com is an invalid email address.

11. Match Password Format:
Password1@ is a valid password.
pass1234! is a valid password.
Passw0rd$ is a valid password.
Password123 is an invalid password.
P@ssw0rd is a valid password.
P@ssw is an invalid password.

12. Find Carrier Based Mobile Numbers:
Input: 9851123123 Output: NT Postpaid
Input: 9841123123 Output: NT Prepaid
Input: 9801123123 Output: Ncell
Input: 9812345678 Output: Unknown carrier
```

LAB-13

13.To extract all the email address present in an array using regular expression.

14.Remove all the white space and non-numeric characters except comma and dots.

Eg. Rs 12,123.123   op: 12,123.123

15.Reverse a string without using a built-in function.

16.Sort and array using sort(), and asort() function and also find the 3rd highest number in that array.

17.Merge array of temperature of two cities and store all the recorded temperature in a single array and find the average temperature and also find the 3 highest and 3 lowest temperature from that array.

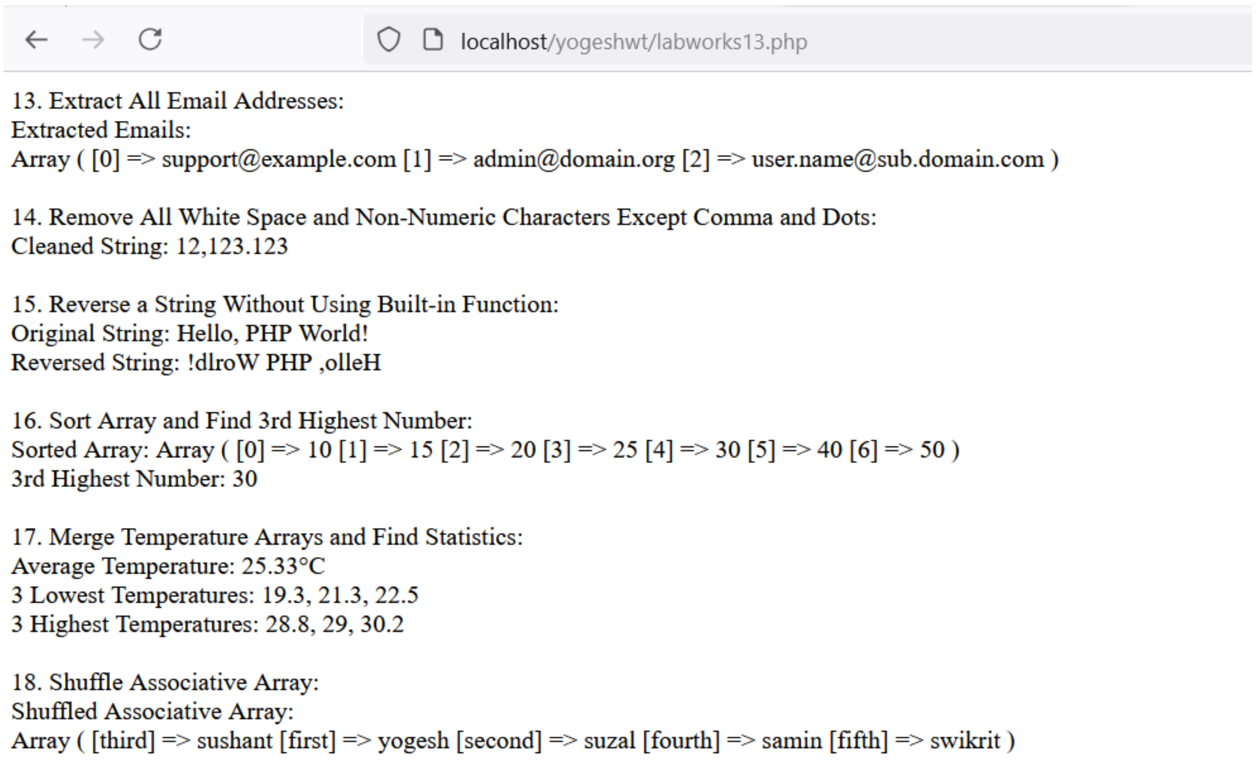
18.Shuffle the associative array preserving it's key and value pairs and display the shuffled result.

CODING

```
<?php
// 13. Extract All Email Addresses from an Array
echo "13. Extract All Email Addresses:<br>";
$emailArray = [
    "Contact us at support@example.com.",
    "Reach admin@domain.org.",
    "Invalid email: user@domain",
    "Valid email: user.name@sub.domain.com",
    "No email here!"
];

$emails = [];
foreach ($emailArray as $text) {
    preg_match_all("/[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-
zA-Z]{2,}/", $text, $matches);
    $emails = array_merge($emails, $matches[0]);
}
$emails = array_unique($emails);
echo "Extracted Emails:<br>";
print_r($emails);
echo "<br><br>";

// 14. Remove All White Space and Non-Numeric Characters
Except Comma and Dots
echo "14. Remove All White Space and Non-Numeric
Characters Except Comma and Dots:<br>";
$inputString = "Rs 12,123.123";
$cleanedString = preg_replace("/[^\d.,]/", "",
$inputString);
echo "Cleaned String: $cleanedString<br><br>";

// 15. Reverse a String Without Using Built-in Function
echo "15. Reverse a String Without Using Built-in
Function:<br>";
function reverseString($str) {
    $reversed = '';
    for ($i = strlen($str) - 1; $i >= 0; $i--) {
        $reversed .= $str[$i];
    }
    return $reversed;
}
$originalString = "Hello, PHP World!";
echo "Original String: $originalString<br>";
echo "Reversed String: " . reverseString($originalString)
. "<br><br>";

// 16. Sort Array and Find 3rd Highest Number
echo "16. Sort Array and Find 3rd Highest Number:<br>";
$array = [10, 20, 15, 30, 50, 40, 25];
sort($array);
echo "Sorted Array: ";
print_r($array);
$uniqueArray = array_unique($array);
rsort($uniqueArray);
echo "<br>3rd Highest Number: " . $uniqueArray[2] .
"<br><br>";
```

```
// 17. Merge Temperature Arrays, Find Average, and 3
Highest & 3 Lowest Temperatures
echo "17. Merge Temperature Arrays and Find
Statistics:<br>";
$city1Temps = [22.5, 25.0, 19.3, 28.8, 24.1];
$city2Temps = [30.2, 27.5, 29.0, 21.3, 25.6];
$mergedTemps = array_merge($city1Temps, $city2Temps);
$averageTemp = array_sum($mergedTemps) /
count($mergedTemps);
sort($mergedTemps);
echo "Average Temperature: " . number_format($averageTemp,
2) . "°C<br>";
echo "3 Lowest Temperatures: " . implode(", ",
array_slice($mergedTemps, 0, 3)) . "<br>";
echo "3 Highest Temperatures: " . implode(", ",
array_slice($mergedTemps, -3)) . "<br><br>";

// 18. Shuffle Associative Array Preserving Key-Value
Pairs
echo "18. Shuffle Associative Array:<br>";
$assocArray = [
    "first" => "yogesh",
    "second" => "suzal",
    "third" => "sushant",
    "fourth" => "samin",
    "fifth" => "swikrit"
];
$keys = array_keys($assocArray);
shuffle($keys);
$shuffledArray = [];
foreach ($keys as $key) {
    $shuffledArray[$key] = $assocArray[$key];
}
echo "Shuffled Associative Array:<br>";
print_r($shuffledArray);
echo "<br>";
?>
```

OUTPUT



```
13. Extract All Email Addresses:
Extracted Emails:
Array ( [0] => support@example.com [1] => admin@domain.org [2] => user.name@sub.domain.com )

14. Remove All White Space and Non-Numeric Characters Except Comma and Dots:
Cleaned String: 12,123.123

15. Reverse a String Without Using Built-in Function:
Original String: Hello, PHP World!
Reversed String: !dlroW PHP ,olleH

16. Sort Array and Find 3rd Highest Number:
Sorted Array: Array ( [0] => 10 [1] => 15 [2] => 20 [3] => 25 [4] => 30 [5] => 40 [6] => 50 )
3rd Highest Number: 30

17. Merge Temperature Arrays and Find Statistics:
Average Temperature: 25.33°C
3 Lowest Temperatures: 19.3, 21.3, 22.5
3 Highest Temperatures: 28.8, 29, 30.2

18. Shuffle Associative Array:
Shuffled Associative Array:
Array ( [third] => sushant [first] => yogesh [second] => suzal [fourth] => samin [fifth] => swikrit )
```

LAB-14

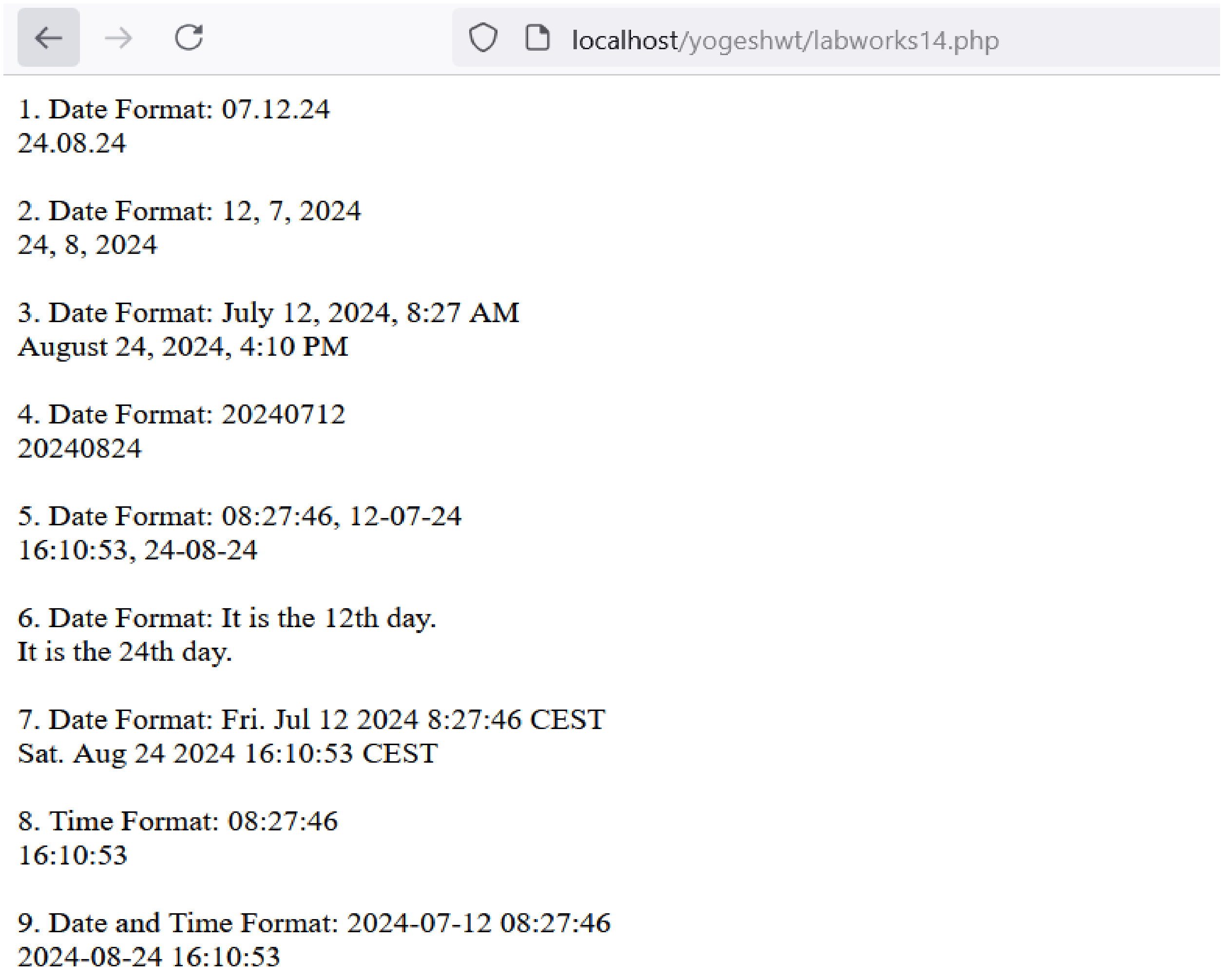
- **Display the following date format:**

  07.12.24
  12, 7, 2024
  July 12, 2024, 8:27 AM
  20240712
  08:27:46, 12-07-24
  It is the 12th day.
  Fri. Jul 12 2024 8:27:46 CEST
  08:27:46
  2024-07-12 08:27:46

CODING

```php
<?php
$date = new DateTime();
// 1. Display date format: 07.12.24
echo "1. Date Format: 07.12.24<br>";
echo $date->format('d.m.y') . "<br><br>";
// 2. Display date format: 12, 7, 2024
echo "2. Date Format: 12, 7, 2024<br>";
echo $date->format('j, n, Y') . "<br><br>";
// 3. Display date format: July 12, 2024, 8:27 AM
echo "3. Date Format: July 12, 2024, 8:27 AM<br>";
echo $date->format('F j, Y, g:i A') . "<br><br>";
// 4. Display date format: 20240712
echo "4. Date Format: 20240712<br>";
echo $date->format('Ymd') . "<br><br>";
// 5. Display date format: 08:27:46, 12-07-24
echo "5. Date Format: 08:27:46, 12-07-24<br>";
echo $date->format('H:i:s, d-m-y') . "<br><br>";
// 6. Display date format: It is the 12th day.
echo "6. Date Format: It is the 12th day.<br>";
echo "It is the " . $date->format('j') . "th
day.<br><br>";
// 7. Display date format: Fri. Jul 12 2024 8:27:46 CEST
echo "7. Date Format: Fri. Jul 12 2024 8:27:46 CEST<br>";
echo $date->format('D. M d Y H:i:s T') . "<br><br>";
// 8. Display time format: 08:27:46
echo "8. Time Format: 08:27:46<br>";
echo $date->format('H:i:s') . "<br><br>";
// 9. Display date and time format: 2024-07-12 08:27:46
echo "9. Date and Time Format: 2024-07-12 08:27:46<br>";
echo $date->format('Y-m-d H:i:s') . "<br>";
?>
```

OUTPUT



```
1. Date Format: 07.12.24
24.08.24

2. Date Format: 12, 7, 2024
24, 8, 2024

3. Date Format: July 12, 2024, 8:27 AM
August 24, 2024, 4:10 PM

4. Date Format: 20240712
20240824

5. Date Format: 08:27:46, 12-07-24
16:10:53, 24-08-24

6. Date Format: It is the 12th day.
It is the 24th day.

7. Date Format: Fri. Jul 12 2024 8:27:46 CEST
Sat. Aug 24 2024 16:10:53 CEST

8. Time Format: 08:27:46
16:10:53

9. Date and Time Format: 2024-07-12 08:27:46
2024-08-24 16:10:53
```

LAB-15

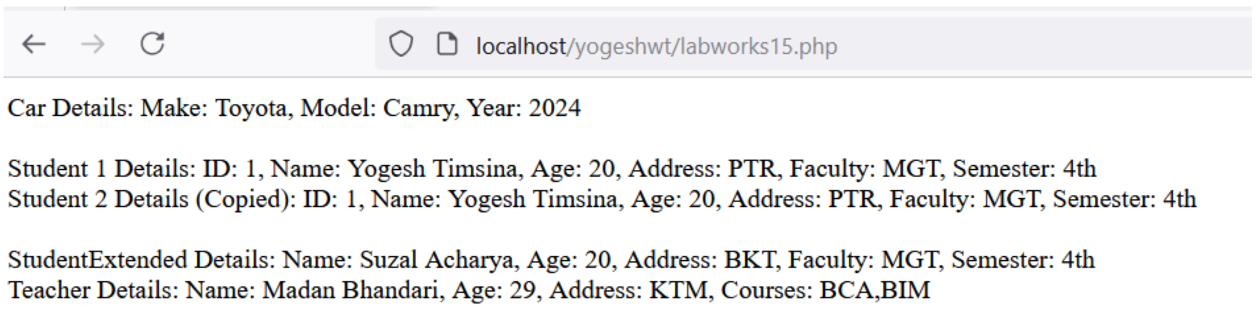
1. **Define a class Car with properties make, model, year and initialize the default constructor and set all the properties.**
    a. **Define a getter method to get the details and display them.**
2. **Define a parameterized constructor for a Student Class with id, name, age, address, faculty, semester.**
3. **Define a copy constructor to pass the above class object as a parameter and display them.**
4. **Define a single level inheritance for the base class Person(name, age, address) with Student(faculty, sem) and Teacher(courses) as child classes to inherit the properties and methods defined in a parent class.**

CODING

```php
<?php
// Define Car Class
class Car {
    private $make;
    private $model;
    private $year;

    // Default Constructor
    public function __construct($make = "Unknown", $model
= "Unknown", $year = 0) {
        $this->make = $make;
        $this->model = $model;
        $this->year = $year;
    }

    // Getter Method
    public function getDetails() {
        return "Make: " . $this->make . ", Model: " .
$this->model . ", Year: " . $this->year;
    }
}
```

```
// Define Student Class with Parameterized Constructor and
Copy Constructor
class Student {
    private $id;
    private $name;
    private $age;
    private $address;
    private $faculty;
    private $semester;

    // Parameterized Constructor
    public function __construct($id, $name, $age,
$address, $faculty, $semester) {
        $this->id = $id;
        $this->name = $name;
        $this->age = $age;
        $this->address = $address;
        $this->faculty = $faculty;
        $this->semester = $semester;
    }

    // Copy Constructor
    public function __clone() {
        // PHP's __clone() is used for copying objects.
    }

    // Display Details
    public function displayDetails() {
        return "ID: " . $this->id . ", Name: " . $this-
>name . ", Age: " . $this->age . ", Address: " . $this-
>address . ", Faculty: " . $this->faculty . ", Semester: "
. $this->semester;
    }
}

// Define Person Class (Base Class)
class Person {
    protected $name;
    protected $age;
    protected $address;

    // Constructor
    public function __construct($name, $age, $address) {
        $this->name = $name;
        $this->age = $age;
        $this->address = $address;
    }
}

// Define Student Class that Inherits Person
class StudentExtended extends Person {
    private $faculty;
    private $semester;

    // Constructor
    public function __construct($name, $age, $address,
$faculty, $semester) {
        parent::__construct($name, $age, $address);
        $this->faculty = $faculty;
        $this->semester = $semester;
    }

    // Display Details
    public function displayDetails() {
        return "Name: " . $this->name . ", Age: " . $this-
>age . ", Address: " . $this->address . ", Faculty: " .
$this->faculty . ", Semester: " . $this->semester;
    }
}

// Define Teacher Class that Inherits Person
class Teacher extends Person {
    private $courses;

    // Constructor
    public function __construct($name, $age, $address,
$courses) {
        parent::__construct($name, $age, $address);
        $this->courses = $courses;
    }

    // Display Details
    public function displayDetails() {
        return "Name: " . $this->name . ", Age: " . $this-
>age . ", Address: " . $this->address . ", Courses: " .
$this->courses;
    }
}

// Testing the Car Class
$car = new Car("Toyota", "Camry", 2024);
echo "Car Details: " . $car->getDetails() . "<br><br>";

// Testing the Student Class
$student1 = new Student(1, "Yogesh Timsina", 20, "PTR",
"MGT", "4th");
echo "Student 1 Details: " . $student1->displayDetails() .
"<br>";

// Copy Constructor Example
$student2 = clone $student1;
echo "Student 2 Details (Copied): " . $student2-
>displayDetails() . "<br><br>";

// Testing Inheritance
$studentExtended = new StudentExtended("Suzal Acharya",
20, "BKT", "MGT", "4th");
echo "StudentExtended Details: " . $studentExtended-
>displayDetails() . "<br>";

$teacher = new Teacher("Madan Bhandari", 29, "KTM",
"BCA,BIM");
echo "Teacher Details: " . $teacher->displayDetails() .
"<br>";
?>
```

OUTPUT



```
Car Details: Make: Toyota, Model: Camry, Year: 2024

Student 1 Details: ID: 1, Name: Yogesh Timsina, Age: 20, Address: PTR, Faculty: MGT, Semester: 4th
Student 2 Details (Copied): ID: 1, Name: Yogesh Timsina, Age: 20, Address: PTR, Faculty: MGT, Semester: 4th

StudentExtended Details: Name: Suzal Acharya, Age: 20, Address: BKT, Faculty: MGT, Semester: 4th
Teacher Details: Name: Madan Bhandari, Age: 29, Address: KTM, Courses: BCA,BIM
```

LAB-16

5. Define a abstract class and inherit it in a child class.

6.Define a interface “paymentGateway “ with methods of processing payments. Implement this interface in two different classes to simulate different payment gateways (eg. eSewa and Khalti)

7.Design a child class that extends parent class and also implements the *interface1*, *interface2*.
8.Design a class to implement function overloading(compile time polymorphism) [*Using Default Arguments, Using Variable-Length Argument Lists, Using Type Checking*]

9.Design a class to implement a function overriding(runtime polymorphism)

10.Design a php traits and use it in a class and access it’s methods.

CODING

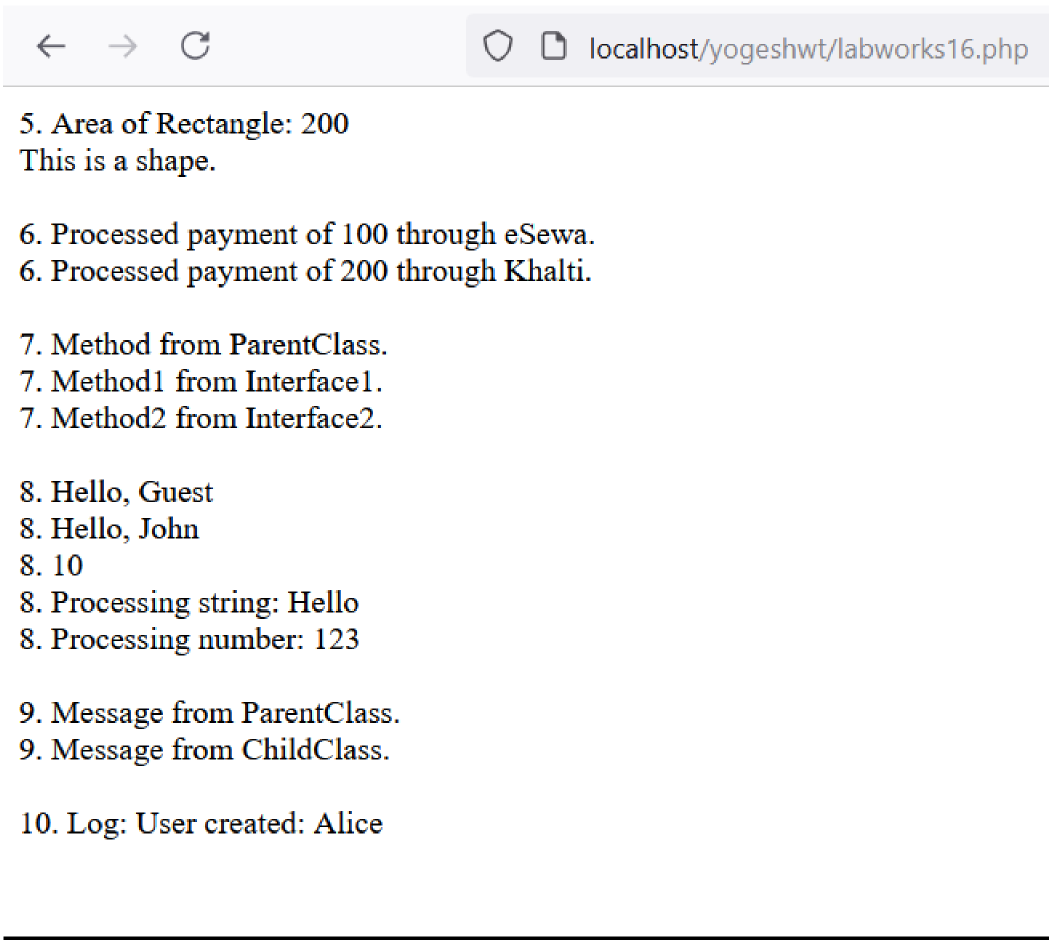
```
<?php
// 5. Define an Abstract Class and Inherit It in a Child
Class
abstract class Shape {
    // Abstract method must be public or protected
    abstract public function getArea();

    public function describe() {
        return "This is a shape.";
    }
}

class Rectangle extends Shape {
    private $width;
    private $height;

    // Constructor
    public function __construct($width, $height) {
        $this->width = $width;
        $this->height = $height;
    }

    // Implement Abstract Method
    public function getArea() {
        return $this->width * $this->height;
    }
}

// Testing Abstract Class and Inheritance
$rectangle = new Rectangle(10, 20);
echo "5. Area of Rectangle: " . $rectangle->getArea() .
"<br>";
echo $rectangle->describe() . "<br><br>";

// 6. Define an Interface and Implement It in Two
Different Classes
interface PaymentGateway {
    public function processPayment($amount);
}

class eSewa implements PaymentGateway {
    public function processPayment($amount) {
        return "Processed payment of $amount through
eSewa.";
    }
}

class Khalti implements PaymentGateway {
    public function processPayment($amount) {
        return "Processed payment of $amount through
Khalti.";
    }
}

// Testing Interface Implementation
$esewa = new eSewa();
echo "6. " . $esewa->processPayment(100) . "<br>";

$khalti = new Khalti();
```

```
echo "6. " . $khalti->processPayment(200) . "<br><br>";

// 7. Design a Child Class that Extends Parent Class and
Implements Interfaces
interface Interface1 {
    public function method1();
}

interface Interface2 {
    public function method2();
}

class ParentClass {
    public function parentMethod() {
        return "Method from ParentClass.";
    }
}

class ChildClass extends ParentClass implements
Interface1, Interface2 {
    public function method1() {
        return "Method1 from Interface1.";
    }

    public function method2() {
        return "Method2 from Interface2.";
    }
}

// Testing Inheritance and Interface Implementation
$child = new ChildClass();
echo "7. " . $child->parentMethod() . "<br>";
echo "7. " . $child->method1() . "<br>";
echo "7. " . $child->method2() . "<br><br>";

// 8. Design a Class to Implement Function Overloading
class OverloadDemo {
    // Using Default Arguments
    public function greet($name = "Guest") {
        return "Hello, " . $name;
    }
```

```php
    // Using Variable-Length Argument Lists
    public function addNumbers(...$numbers) {
        return array_sum($numbers);
    }

    // Using Type Checking
    public function process($input) {
        if (is_string($input)) {
            return "Processing string: $input";
        } elseif (is_numeric($input)) {
            return "Processing number: $input";
        } else {
            return "Processing unknown type.";
        }
    }
}

// Testing Function Overloading
$overload = new OverloadDemo();
echo "8. " . $overload->greet() . "<br>";
echo "8. " . $overload->greet("John") . "<br>";
echo "8. " . $overload->addNumbers(1, 2, 3, 4) . "<br>";
echo "8. " . $overload->process("Hello") . "<br>";
echo "8. " . $overload->process(123) . "<br><br>";

// 9. Design a Class to Implement Function Overriding
class ParentClassForOverride {
    public function showMessage() {
        return "Message from ParentClass.";
    }
}

class ChildClassForOverride extends ParentClassForOverride
{
    public function showMessage() {
        return "Message from ChildClass.";
    }
}

// Testing Function Overriding
$parent = new ParentClassForOverride();
echo "9. " . $parent->showMessage() . "<br>";
```

```php
$childForOverride = new ChildClassForOverride();
echo "9. " . $childForOverride->showMessage() .
"<br><br>";

// 10. Design a PHP Trait and Use It in a Class
trait Logger {
    public function log($message) {
        echo "Log: " . $message . "<br>";
    }
}

class User {
    use Logger;

    public function createUser($name) {
        $this->log("User created: $name");
    }
}

// Testing Traits
$user = new User();
echo "10. ";
$user->createUser("Alice");
?>
```

OUTPUT



```
5. Area of Rectangle: 200
This is a shape.

6. Processed payment of 100 through eSewa.
6. Processed payment of 200 through Khalti.

7. Method from ParentClass.
7. Method1 from Interface1.
7. Method2 from Interface2.

8. Hello, Guest
8. Hello, John
8. 10
8. Processing string: Hello
8. Processing number: 123

9. Message from ParentClass.
9. Message from ChildClass.

10. Log: User created: Alice
```

LAB-17

11.Define a static class and access it's identifiers and methods.

12.Use a php namespace to find the diameter and area of a circle.

/Src

……. /Math

……………../Geometry

……………………../Circle.php

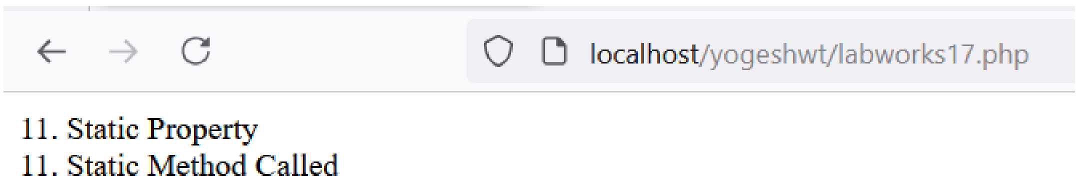
……………../Constants

13.Use PHP magic methods __set, __get, __call, __callStatic methods inside a class and use it to set and get information.

14.Define a class that extends a exception class to handle divide by zero and divison by a negative number and handles the exception. Also display the error as "divide by zero occurred.", "divide by negative number occurred."

CODING

```
<?php
// 11. Define a Static Class and Access Its Identifiers
and Methods
class StaticClass {
    public static $value = "Static Property";

    public static function staticMethod() {
        return "Static Method Called";
    }
}

// Accessing Static Property and Method
echo "11. " . StaticClass::$value . "<br>";
echo "11. " . StaticClass::staticMethod() . "<br><br>";
?>
```


localhost/yogeshwt/labworks17.php

```
11. Static Property
11. Static Method Called
```

Directory structure

```
C:\xampp\htdocs\yogeshwt\
   Src\
      Math\
         Geometry\
            Circle.php
   labworks17.php
```

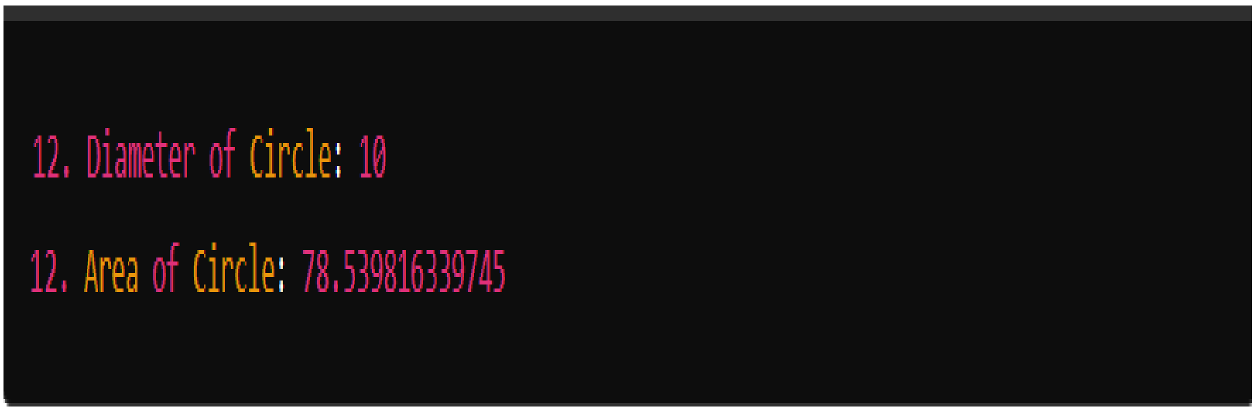
Circle.php

```php
<?php
namespace Src\Math\Geometry;
class Circle {
   private $radius;
   public function __construct($radius) {
      $this->radius = $radius;
   }
   public function getDiameter() {
      return 2 * $this->radius;
   }
   public function getArea() {
        return pi() * ($this->radius ** 2);
    }
}
?>
```

Labworks17.php

```
<?php
require 'Src/Math/Geometry/Circle.php';
use Src\Math\Geometry\Circle;
$circle = new Circle(5);
echo "12. Diameter of Circle: " . $circle->getDiameter() .
"<br>";
echo "12. Area of Circle: " . $circle->getArea() . "<br>";
?>
```

OUTPUT



```
12. Diameter of Circle: 10
12. Area of Circle: 78.539816339745
```

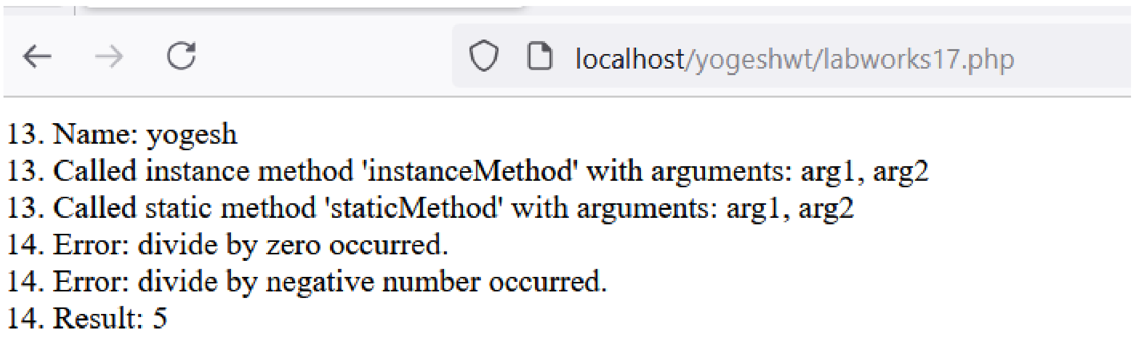
CODING

```
<?php
// 13. PHP Magic Methods __set, __get, __call, __callStatic
```

```
class MagicMethods {
   private $data = [];
   // __set handles setting properties dynamically
   public function __set($name, $value) {
      $this->data[$name] = $value;
   }
   // __get handles getting properties dynamically
   public function __get($name) {
      return isset($this->data[$name]) ? $this->data[$name] : null;
   }
   // __call handles calling instance methods dynamically
   public function __call($name, $arguments) {
      return "Called instance method '$name' with arguments: " . implode(', ',
$arguments);
   }
   // __callStatic handles calling static methods dynamically
   public static function __callStatic($name, $arguments) {
      return "Called static method '$name' with arguments: " . implode(', ',
$arguments);
   }
}

// Testing Magic Methods
$magic = new MagicMethods();
$magic->name = "John";
echo "13. Name: " . $magic->name . "<br>";
echo "13. " . $magic->instanceMethod("arg1", "arg2") . "<br>";
echo "13. " . MagicMethods::staticMethod("arg1", "arg2") . "<br>";
// 14. Custom Exception Handling for Division
class CustomException extends Exception {}
class DivisionCalculator {
  public function divide($numerator, $denominator) {
    if ($denominator == 0) {
      throw new CustomException("divide by zero occurred.");
    }
    if ($denominator < 0) {
      throw new CustomException("divide by negative number occurred.");
    }
    return $numerator / $denominator;
  }
}
// Testing Division Handling
$calculator = new DivisionCalculator();
try {
  $result = $calculator->divide(10, 0);
} catch (CustomException $e) {
  echo "14. Error: " . $e->getMessage() . "<br>";
}
try {
  $result = $calculator->divide(10, -2);
} catch (CustomException $e) {
  echo "14. Error: " . $e->getMessage() . "<br>";
}
```

```
try {
    $result = $calculator->divide(10, 2);
    echo "14. Result: " . $result . "<br>";
} catch (CustomException $e) {
    echo "14. Error: " . $e->getMessage() . "<br>";
}

?>
```

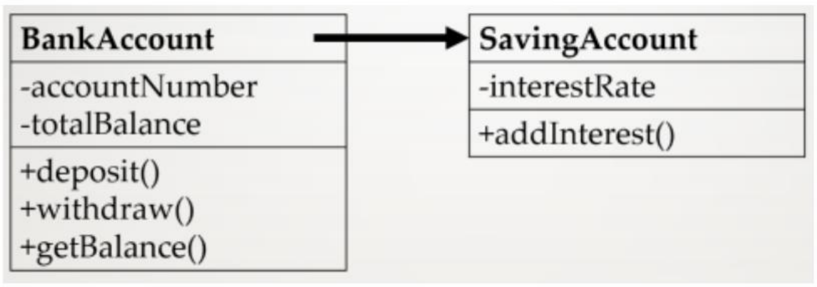
OUTPUT



```
13. Name: yogesh
13. Called instance method 'instanceMethod' with arguments: arg1, arg2
13. Called static method 'staticMethod' with arguments: arg1, arg2
14. Error: divide by zero occurred.
14. Error: divide by negative number occurred.
14. Result: 5
```

LAB-18

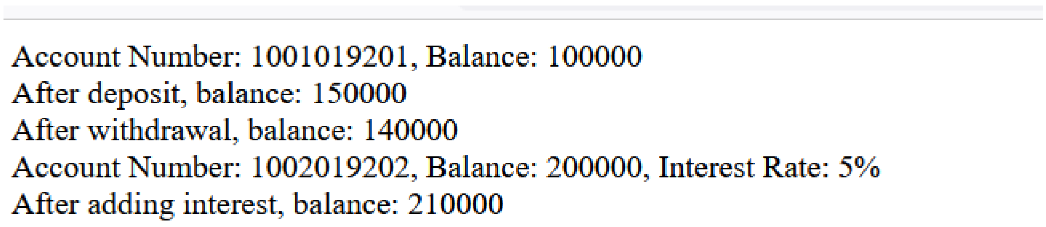
15.Implement the concept of inheritance considering the above diagram also handle a exception for withdraw amount exceeding the balance amount.



CODING

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Bank Account Example</title>
</head>
<body>
    <?php
    class BankAccount {
        private $accountNumber;
        private $totalBalance;

        public function __construct($accountNumber,
$initialBalance) {
            $this->accountNumber = $accountNumber;
            $this->totalBalance = $initialBalance;
        }

        public function deposit($amount) {
            $this->totalBalance += $amount;
            return $this->totalBalance;
        }

        public function getBalance() {
            return $this->totalBalance;
        }

        public function withdraw($amount) {
            if ($amount <= $this->totalBalance) {
                $this->totalBalance -= $amount;
                return $this->totalBalance;
            } else {
                return "Insufficient funds to withdraw
$amount.";
            }
         }

        public function getDetails() {
            return "Account Number: $this->accountNumber,
Balance: $this->totalBalance";
        }
    }

    class SavingAccount {
        private $accountNumber;
        private $totalBalance;
        private $interestRate;

        public function __construct($accountNumber,
$initialBalance, $interestRate) {
            $this->accountNumber = $accountNumber;
            $this->totalBalance = $initialBalance;
            $this->interestRate = $interestRate;
        }

        public function deposit($amount) {
            $this->totalBalance += $amount;
            return $this->totalBalance;
        }

        public function getBalance() {
            return $this->totalBalance;
        }

        public function withdraw($amount) {
            if ($amount <= $this->totalBalance) {
                $this->totalBalance -= $amount;
                return $this->totalBalance;
            } else {
                return "Insufficient funds to withdraw
$amount.";
            }
        }

        public function addInterest() {
            $interest = ($this->totalBalance * $this-
>interestRate) / 100;
            $this->totalBalance += $interest;
            return $this->totalBalance;
        }

        public function getDetails() {
            return "Account Number: $this->accountNumber,
Balance: $this->totalBalance, Interest Rate: $this-
>interestRate%";
        }
    }
    $account1 = new BankAccount(1001019201, 100000);
    echo $account1->getDetails() . "<br>";

    echo "After deposit, balance: " . $account1-
>deposit(50000) . "<br>";

    echo "After withdrawal, balance: " . $account1-
>withdraw(10000) . "<br>";

    $account2 = new SavingAccount(1002019202, 200000, 5);
    echo $account2->getDetails() . "<br>";

    echo "After adding interest, balance: " . $account2-
>addInterest() . "<br>";
    ?>
</body>
</html>
```

OUTPUT



```
Account Number: 1001019201, Balance: 100000
After deposit, balance: 150000
After withdrawal, balance: 140000
Account Number: 1002019202, Balance: 200000, Interest Rate: 5%
After adding interest, balance: 210000
```

LAB-19

1.Read a single line by line with fgets() from a opened text file and display until the end of the file.

2.Read a single character with fgetc() from a opened text file and display it until the end of the file.

3.Copy a content from one text *(source.txt)* file to another text *(destination.txt)* file and again open (*destination.txt)* file and append some content to it.

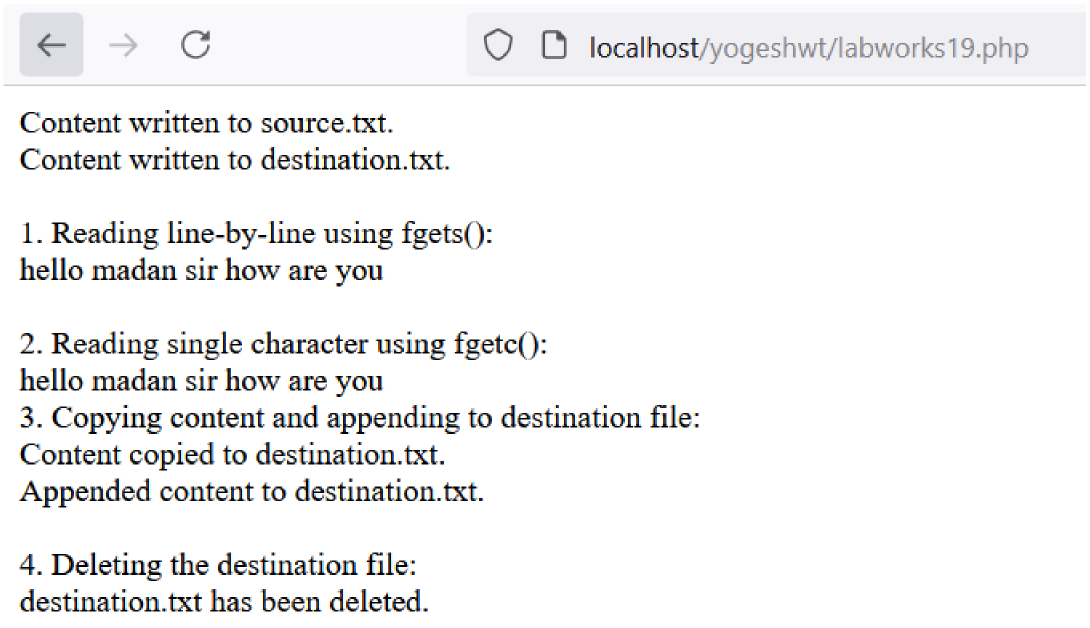
4.Delete the created text file.

CODING

```php
<?php

// Define file paths
$sourceFilePath = 'source.txt';
$destinationFilePath = 'destination.txt';

// 1. Create and write initial content to source.txt
$sourceFile = fopen($sourceFilePath, "w+");
$sourceContent = 'hello madan sir how are you';
fwrite($sourceFile, $sourceContent);
fclose($sourceFile);
echo "Content written to source.txt.<br>";

// 2. Create and write initial content to destination.txt
$destinationFile = fopen($destinationFilePath, "w+");
$destinationContent = 'lab report of webtech2';
fwrite($destinationFile, $destinationContent);
fclose($destinationFile);
echo "Content written to destination.txt.<br>";
```

```
// 3. Read the source file line-by-line using fgets()
echo "<br>1. Reading line-by-line using fgets():<br>";
$sourceFile = fopen($sourceFilePath, 'r');
if ($sourceFile) {
    while (($line = fgets($sourceFile)) !== false) {
        echo htmlspecialchars($line) . "<br>";
    }
    fclose($sourceFile);
} else {
    echo "Unable to open the source file.<br>";
}

// 4. Read a single character using fgetc()
echo "<br>2. Reading single character using fgetc():<br>";
$sourceFile = fopen($sourceFilePath, 'r');
if ($sourceFile) {
    while (($char = fgetc($sourceFile)) !== false) {
        echo htmlspecialchars($char);
    }
    fclose($sourceFile);
} else {
    echo "Unable to open the source file.<br>";
}

// 5. Copy content from source file to destination file
and append content
echo "<br>3. Copying content and appending to destination
file:<br>";
if (file_exists($sourceFilePath)) {
    if (copy($sourceFilePath, $destinationFilePath)) {
        echo "Content copied to
$destinationFilePath.<br>";

        // Append content
        $additionalContent = "Appended content.\n";
        file_put_contents($destinationFilePath,
$additionalContent, FILE_APPEND);
        echo "Appended content to
$destinationFilePath.<br>";
    } else {
        echo "Failed to copy content.<br>";
    }
} else {
    echo "Source file does not exist.<br>";
}

// 6. Delete the destination file
echo "<br>4. Deleting the destination file:<br>";
if (file_exists($destinationFilePath)) {
    if (unlink($destinationFilePath)) {
        echo "$destinationFilePath has been deleted.<br>";
    } else {
        echo "Failed to delete $destinationFilePath.<br>";
    }
} else {
    echo "$destinationFilePath does not exist.<br>";
}

?>
```

OUTPUT



```
Content written to source.txt.
Content written to destination.txt.

1. Reading line-by-line using fgets():
hello madan sir how are you

2. Reading single character using fgetc():
hello madan sir how are you
3. Copying content and appending to destination file:
Content copied to destination.txt.
Appended content to destination.txt.

4. Deleting the destination file:
destination.txt has been deleted.
```

LAB-20

5.Create a Ticket Issuing System for a slow internet connection where a user can create a ticket for his/her problem. The user can submit the form including his/her username, email, phone, problem type (***poor internet connection, fiber breakage)***, comments and screenshots of a problem in a .png file format ***(don't accept any other than .png file format)***.

The submitted file should be in a directory order.

Ticket/

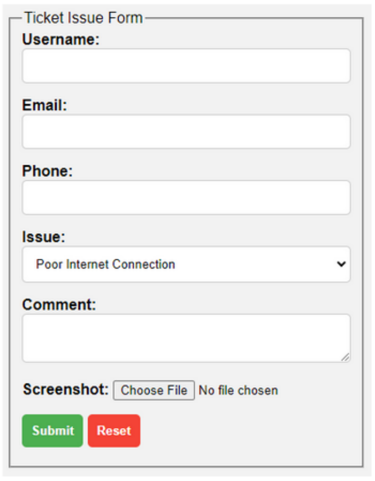
…………/customer/

……………………/comments/abcComments.txt

……………………/screenshots/abcScreenshots.png

Store the person details in a class person.

*If possible embed customer name to comment and screenshot.


Ticket Issue Form

Username:

Email:

Phone:

Issue:

Poor Internet Connection

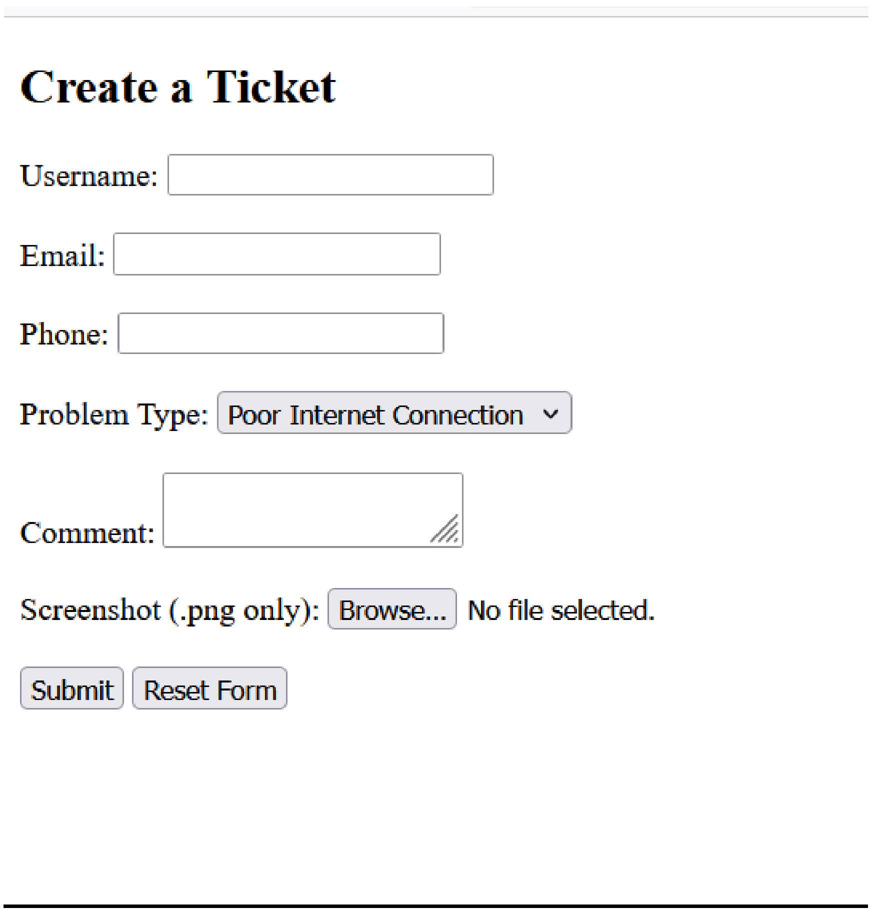
Comment:

Screenshot: Choose File No file chosen

CODING

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Ticket Issuing System</title>
</head>
<body>
    <h2>Create a Ticket</h2>
    <form action="fTicket.php" method="post"
enctype="multipart/form-data">
        <label for="username">Username:</label>
        <input type="text" id="username" name="username"
required><br><br>

        <label for="email">Email:</label>
        <input type="email" id="email" name="email"
required><br><br>

        <label for="phone">Phone:</label>
        <input type="text" id="phone" name="phone"
required><br><br>

        <label for="problem_type">Problem Type:</label>
        <select id="problem_type" name="problem_type"
required>
            <option value="poor_internet_connection">Poor
Internet Connection</option>
            <option value="fiber_breakage">Fiber
Breakage</option>
        </select><br><br>

        <label for="comment">Comment:</label>
        <textarea id="comment" name="comment"
required></textarea><br><br>

        <label for="screenshot">Screenshot (.png
only):</label>
        <input type="file" id="screenshot"
name="screenshot" accept="image/png" required><br><br>

        <button type="submit">Submit</button>
        <button type="reset">Reset Form</button>
    </form>
</body>
</html>
```

OUTPUT



Create a Ticket

Username:

Email:

Phone:

Problem Type: Poor Internet Connection

Comment:

Screenshot (.png only): Browse... No file selected.

LAB-21

6..Design the Travel Booking Form for your own travel company. Also write a PHP program to sanitize and prevent SQL injection in a form data. Handle the form data in ***confirm.php*** file and display the form input data in ***confirm.php*** file.

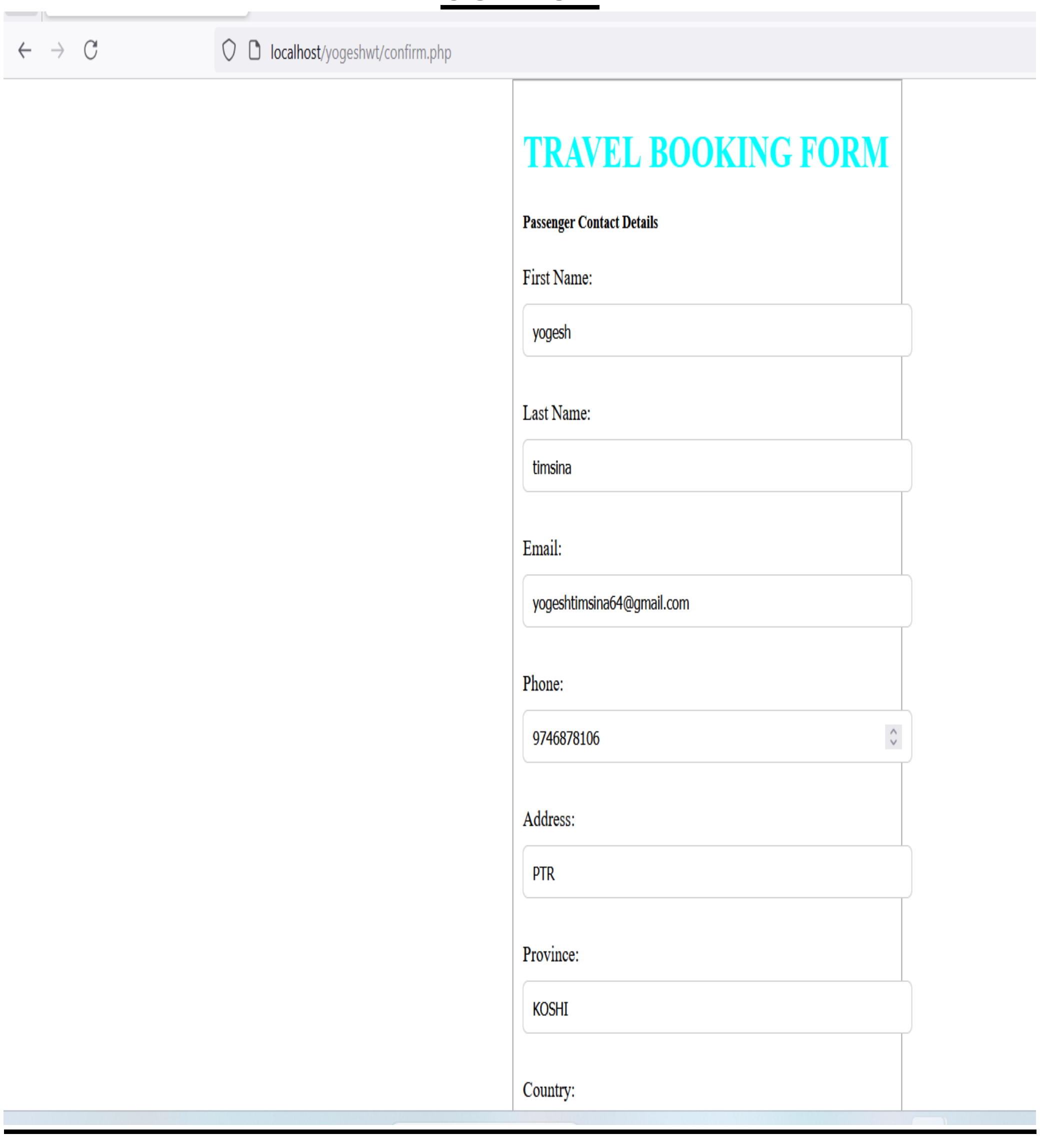
[Hint: use PHP Superglobals]

CODING

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Traveling Form</title>
    <style>
        body {
            display: grid;
            justify-content: center;
            align-items: center;
             height: 100vh;
            margin: 0;
        }
        fieldset {
            background-color: rgba(255, 255, 255, 0.8);
            padding: 10px;
            max-width: 600px;
            width: 100%;
            box-sizing: border-box;
        }
        label {
            display: block;
            margin: 10px 0 5px;
        }
        input, select, textarea {
            width: 100%;
            padding: 10px;
            margin: 5px 0 20px;
            border: 1px solid #ccc;
            border-radius: 5px;
        }
        input[type="submit"], input[type="reset"] {
            background-color: aqua;
            border: none;
            color: white;
            cursor: pointer;
            padding: 10px 20px;
            margin-right: 10px;
        }
        input[type="reset"] {
            background-color: gray;
        }
        @media (max-width: 768px) {
            fieldset {
                max-width: 90%;
            }
        }
        @media (max-width: 480px) {
            fieldset {
                max-width: 95%;
            }
        }
    </style>
</head>
<body>
    <form action="" method="post">
        <fieldset>
            <div name="head">
                <h1 style="color: aqua;">TRAVEL BOOKING
FORM</h1>
            </div>
            <h5>Passenger Contact Details</h5>
            <label for="fname">First Name:</label>
            <input type="text" id="fname" name="fname"
required>
            <label for="lname">Last Name:</label>
            <input type="text" id="lname" name="lname"
required>
            <label for="email">Email:</label>
            <input type="email" id="email" name="email"
required>
            <label for="phone">Phone:</label>
            <input type="number" id="phone" name="phone">
            <label for="address">Address:</label>
            <input type="text" id="address" name="address"
required>
            <label for="province">Province:</label>
            <input type="text" id="province"
name="province">
            <label for="country">Country:</label>
            <input type="text" id="country"
name="country">
            <h5>Traveling Details</h5>
            <label for="Tpassenger">Number Of
Passengers:</label>
            <input type="number" id="Tpassenger"
name="Tpassenger">
            <label for="male">No. Of Males:</label>
            <input type="number" id="male" name="male">
            <label for="female">No. Of Females:</label>
            <input type="number" id="female"
name="female">
            <label for="children">No. Of Children:</label>
            <input type="number" id="children"
name="children">
            <div>Traveling Time:</div>
            <label for="day">Day</label>
            <input type="radio" id="day" value="day"
name="ttime">
            <label for="night">Night</label>
            <input type="radio" id="night" value="night"
name="ttime">
```

```
            <label for="destination">Travel
Destination:</label>
            <select id="destination" name="destination">
                <optgroup label="PURBA">
                    <option value="KTM-BTM">KTM-
BTM</option>
                    <option value="KTM-BTR">KTM-
BTR</option>
                </optgroup>
                <optgroup label="PASCHIM">
                    <option value="KTM-NPJ">KTM-
NPJ</option>
                </optgroup>
            </select>
            <label for="pcode">Postal code:</label>
            <input type="text"><br>
            <label for="Ttype">Reason Of
Traveling:</label>
            <textarea id="Ttype" name="Ttype"></textarea>
            <input type="submit" name="submit"
value="Submit">
            <input type="reset" value="Reset">
        </fieldset>
    </form>
</body>
</html>
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $fname = $_POST['fname'];
    $lname = $_POST['lname'];
    $email = $_POST['email'];
    $phone = $_POST['phone'];
    $address = $_POST['address'];
    $province = $_POST['province'];
    $country = $_POST['country'];
    $Tpassenger = $_POST['Tpassenger'];
    $male = $_POST['male'];
    $female = $_POST['female'];
    $children = $_POST['children'];
    $ttime = $_POST['ttime'];
    $destination = $_POST['destination'];
    $Ttype = $_POST['Ttype'];
    echo "<h1>Traveling Form Submission</h1>";
    echo "First Name: " .htmlspecialchars($fname) .
"<br>";
    echo "Last Name: " .htmlspecialchars ($lname) .
"<br>";
    echo "Email: " . htmlspecialchars($email) . "<br>";
    echo "Phone: " . htmlspecialchars($phone) . "<br>";
    echo "Address: " .htmlspecialchars ($address) .
"<br>";
    echo "Province: " . htmlspecialchars($province) .
"<br>";
    echo "Country: " .htmlspecialchars ($country) .
"<br>";
    echo "Number Of Passengers: " .htmlspecialchars
($Tpassenger) . "<br>";
    echo "No. Of Males: " . htmlspecialchars($male) .
"<br>";
    echo "No. Of Females: " . htmlspecialchars($female) .
"<br>";
    echo "No. Of Children: " .htmlspecialchars ($children)
. "<br>";
    echo "Traveling Time: " .htmlspecialchars($ttime) .
"<br>";
    echo "Travel Destination: " .htmlspecialchars
($destination) . "<br>";
    echo "Reason Of Traveling: " .
htmlspecialchars($Ttype) . "<br>";

}
?>
```

OUTPUT



TRAVEL BOOKING FORM

Passenger Contact Details

First Name:

yogesh

Last Name:

timsina

Email:

yogeshtimsina64@gmail.com

Phone:

9746878106

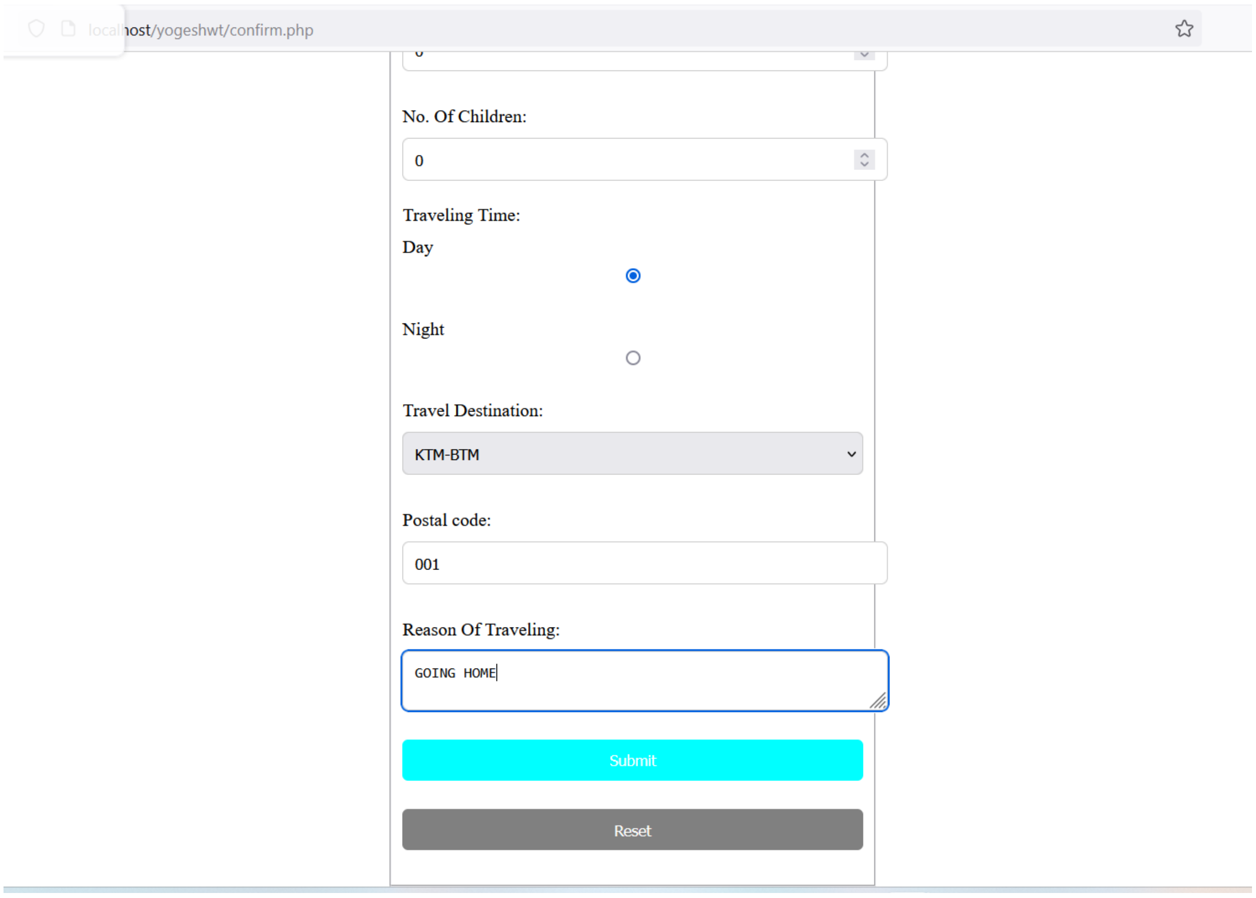
Address:

PTR

Province:

KOSHI

Country:



localhost/yogeshwt/confirm.php

No. Of Children:

0

Traveling Time:

Day [✓ ●]

Night ○

Travel Destination:

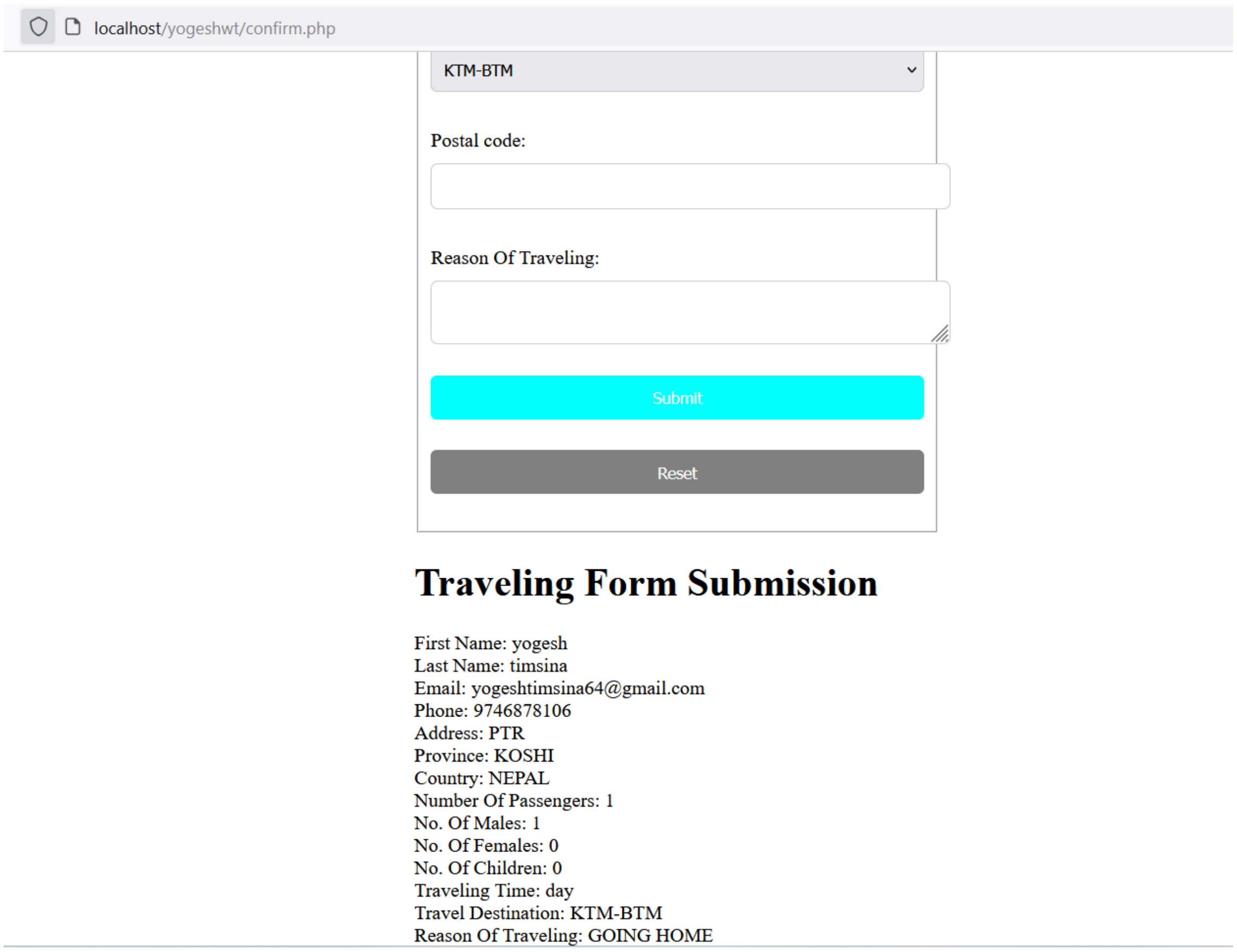
KTM-BTM

Postal code:

001

Reason Of Traveling:

GOING HOME



Traveling Form Submission

First Name: yogesh
Last Name: timsina
Email: yogeshtimsina64@gmail.com
Phone: 9746878106
Address: PTR
Province: KOSHI
Country: NEPAL
Number Of Passengers: 1
No. Of Males: 1
No. Of Females: 0
No. Of Children: 0
Traveling Time: day
Travel Destination: KTM-BTM
Reason Of Traveling: GOING HOME

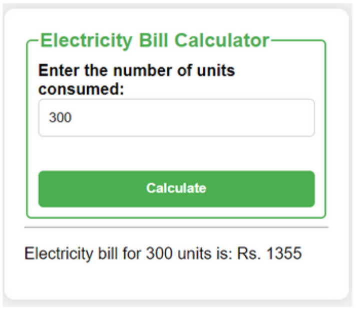
LAB-22

7.User-defined functions:

To calculate the electricity bill as per the given criteria mentioned below:

Conditions:

- For the first 50 units – Rs. 3.50/unit
- For the next 100 units – Rs. 4.00/unit
- For the next 150 units – Rs. 5.20/unit
- For units above 250 – Rs. 6.50/unit


Electricity Bill Calculator
Enter the number of units consumed:
300
Calculate

Electricity bill for 300 units is: Rs. 1355

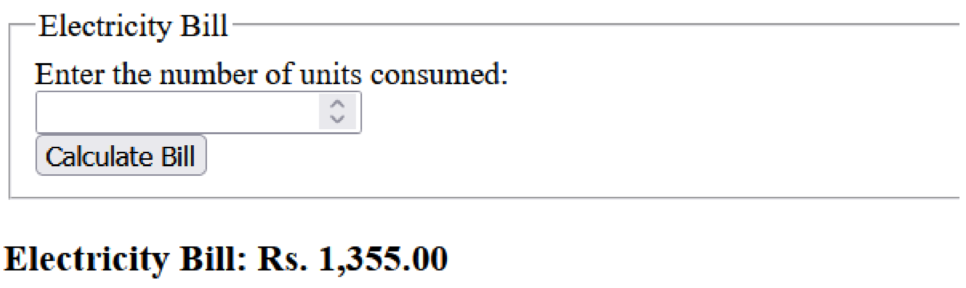
Note: The number of units consumed should be provided using form inputs. Use a single page form Handling.

CODING

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Electricity Bill Calculator</title>
</head>
<body>

    <form method="post">
        <fieldset>
        <legend>Electricity Bill</legend>
        <label for="units">Enter the number of units
consumed:</label><br>
        <input type="number" name="units" id="units"
required><br>
        <input type="submit" name="calculate"
value="Calculate Bill">
```

```
        </fieldset>
    </form>
    <?php
    function calculateElectricityBill($units) {
        $bill = 0;

        if ($units <= 50) {
            $bill = $units * 3.50;
        } elseif ($units <= 150) {
            $bill = (50 * 3.50) + (($units - 50) * 4.00);
        } elseif ($units <= 300) {
            $bill = (50 * 3.50) + (100 * 4.00) + (($units
- 150) * 5.20);
        } else {
            $bill = (50 * 3.50) + (100 * 4.00) + (150 *
5.20) + (($units - 300) * 6.50);
        }

        return number_format($bill, 2);
    }

    if (isset($_POST['calculate'])) {
        $units = $_POST['units'];

        if (is_numeric($units) && $units > 0) {
            $billAmount =
calculateElectricityBill($units);
            echo "<h3>Electricity Bill: Rs.
{$billAmount}</h3>";
        } else {
            echo "<h3>Please enter a valid number of
units.</h3>";
        }
    }
    ?>

</body>
</html>
```

OUTPUT


Electricity Bill

Enter the number of units consumed:

```
Electricity Bill: Rs. 1,355.00
```