**PSG COLLEGE OF TECHNOLOGY, COIMBATORE – 641 004**

**Department of Applied Mathematics and Computational Sciences**

**MSc SOFTWARE SYSTEMS – Semester IV**

**20XW48 – Web Designing Lab**

**PROBLEM SHEET – PHP & MySQL**

1. Write a PHP script to get the PHP version and configuration information.

<!DOCTYPE html>

<html>

<body>

<?php

phpinfo()

?>

</body>

</html>

1. Create a PHP script that displays 1-2-3-4-5-6-7-8-9-10 on one line. There will be no hyphen(-) at starting and ending position.

<!DOCTYPE html>

<html>

<body>

<?php

print "1 2-3-4-5-6-7-8-9 10"

?>

</body>

</html>

1. Write a PHP program to keep track of the number of visitors visiting the web page and display the count of visitors with proper headings.

print "**REFRESH PAGE** ";

$name="counter.txt";

$file = fopen($name,"r");

$hits= fscanf($file,"%d");

fclose($file);

$hits[0]++;

$file = fopen($name,"w");

fprintf($file,"%d",$hits[0]);

fclose($file);

print "Total number of views: ".$hits[0];

?>

1. Write a PHP program to display a digital clock which displays the current time of the server.

<!DOCTYPE html>

<html>

<body>

<?php

echo "Current date and time is :";

$myDate = date("d-m-y h:i:s");

echo $myDate

?>

</body>

</html>

1. Write a PHP script to calculate and display average temperature, five lowest and highest temperatures using functions.

$month\_temp = "78, 60, 62, 68, 71, 68, 73, 85, 66, 64, 76, 63, 81, 76, 73,

68, 72, 73, 75, 65, 74, 63, 67, 65, 64, 68, 73, 75, 79, 73";

$temp\_array = explode(',', $month\_temp);

$tot\_temp = 0;

$temp\_array\_length = count($temp\_array);

foreach($temp\_array as $temp)

{

$tot\_temp += $temp;

}

$avg\_high\_temp = $tot\_temp/$temp\_array\_length;

echo "Average Temperature is : ".$avg\_high\_temp."

";

sort($temp\_array);

echo " List of five lowest temperatures :";

for ($i=0; $i< 5; $i++)

{

echo $temp\_array[$i].", ";

}

echo "List of five highest temperatures :";

for ($i=($temp\_array\_length-5); $i< ($temp\_array\_length); $i++)

{

echo $temp\_array[$i].", ";

}

1. Write a PHP script that removes the whitespaces from a string.

*Sample String:* 'The quick " " brown fox'

*Expected Output:* Thequick""brownfox

*<!DOCTYPE html>*

*<html>*

*<body>*

*<?php*

*$str = "The quick brown fox ";*

*$str = str\_replace(' ', '', $str);*

*print "$str"*

*?>*

*</body>*

*</html>*

1. Write a PHP function that checks whether a passed string is palindrome or not? A palindrome is word, phrase, or sequence that reads the same backward as forward, e.g., madam or nurses run.

<!DOCTYPE html>

<html>

<body>

<?php

function Palindrome($string){

if (strrev($string) == $string){

return 1;

}

else{

return 0;

}

}

$original = "DAD";

if(Palindrome($original)){

echo "Palindrome";

}

else {

echo "Not a Palindrome";

}

?>

</body>

</html>

1. Write a PHP script to sort the following associative array using functions: array("Sophia"=>"31","Jacob"=>"41","William"=>"39","Ramesh"=>"40") in

a) ascending order sort by value

b) ascending order sort by key

c) descending order sorting by value

d) descending order sorting by key

<?php

echo "

Associative array : Ascending order sort by value

";

$array2=array("Sophia"=>"31","Jacob"=>"41","William"=>"39","Ramesh"=>"40"); asort($array2);

foreach($array2 as $y=>$y\_value)

{

echo "Age of ".$y." is : ".$y\_value."

";

}

echo "

Associative array : Ascending order sort by Key

";

$array3=array("Sophia"=>"31","Jacob"=>"41","William"=>"39","Ramesh"=>"40"); ksort($array3);

foreach($array3 as $y=>$y\_value)

{

echo "Age of ".$y." is : ".$y\_value."

";

}

echo "

Associative array : Descending order sorting by Value

";

$age=array("Sophia"=>"31","Jacob"=>"41","William"=>"39","Ramesh"=>"40");

arsort($age);

foreach($age as $y=>$y\_value)

{

echo "Age of ".$y." is : ".$y\_value."

";

}

echo "

Associative array : Descending order sorting by Key

";

$array4=array("Sophia"=>"31","Jacob"=>"41","William"=>"39","Ramesh"=>"40"); krsort($array4);

foreach($array4 as $y=>$y\_value)

{

echo "Age of ".$y." is : ".$y\_value."

";

}

?>

1. Write a PHP script to store and retrieve persistent data across a client session.
2. Write a PHP script to simulate the Shopping Cart, allow users to select items from a catalog and save them for later access using session.
3. Write a PHP script to store and retrieve cookies in a web page.
4. Write a PHP script named states.php that creates a variable $states with the value *"Mississippi Texas Massachusetts Kansas"*. The script should perform the following tasks:

a) Search for a word in *$states* that ends in *xas*. Store this word in element 0 of an array

named *$statesArray*.

b) Search for a word in *$states* that begins with *k* and ends in *s*. Perform a case-insensitive

comparison. Store this word in element 1 of *$statesArray*.

c) Search for a word in *$states* that begins with *M* and ends in *s*. Store this element in

element 2 of the array.

d) Search for a word in *$states* that ends in a. Store this word in element 3 of the array.

e) Search for a word in *$states* at the beginning of the string that starts with *M*. Store this

word in element 4 of the array.

f) Output the array *$statesArray* to the screen.

1. Write a PHP script that tests whether an e-mail address is input correctly. Verify that the

input begins with series of characters, followed by the @ character, another series of characters, a period (.) and a final series of characters. Test your program, using both valid and invalid e-mail addresses.

1. Write a PHP program to sort the student records which are stored in the database using selection sort.
2. Write a PHP script that obtains a URL and its description from a user and stores the information into a database using MySQL. Create and run a SQL script with a database named *URL* and a table named *URLTable*. The first field of the table should contain an actual *URL*, and the second, which is named Description, should contain a description of the *URL*. Use www.deitel.com as the first *URL*, and input *‘Cool site!’* as its description. The second *URL* should be www.php.net, and the description should be ‘*The official PHP’* site. After each new *URL* is submitted, print the contents of the database in a table.
3. Develop a PHP program to authenticate users with a valid *User Id* and *Password* before granting access to a protected resource.