RATHINAM COLLEGE OF ARTS & SCIENCE

(AUTONOMOUS)

Coimbatore-641021

DEPARTMENT OF COMPUTER SCIENCE

22BCS5DP--(PHP & MySql) Lab Manual

Prepared by:	Approved & Reviewed by:	Issued by:	W.e.f Date:
Dr.M.Kathiresh			

In-charge **HOD** **Principal**

RATHINAM COLLEGE OF ARTS & SCIENCE (AUTONOMOUS)

Rathinam Techzone Campus, Pollachi Road, Eachanari,

Coimbatore - 641021, Tamil Nadu.

http://www.rathinamcollege.com

Department of Computer Science (AI & DS)

Lab Manual for the Academic Year 2020-21

(in accordance with Computer Science syllabus)

SUBJECT : **PHP & MySql**

STREAM : B.Sc Computer Science

Staff Incharge **H.O.D**

INDEX

LAB OBJECTIVE

By the completion of the Web Development with PHP/MySQL course you should be able to:

- Understand the usage of PHP and MySQL in dynamic web development.
- Understand PHP language data types, logic controls, built-in and user-defined functions.
- Be able to setup and configure MySQL, PHP, Apache web server development environment.
- Select, insert, update and delete data using SQL language.
- Understand Object oriented programming paradigm in PHP.
- Build a simple, yet functional web application using PHP/MySQL.

INTRODUCTION ABOUT LAB

There are 60 systems installed in this Lab. Their configurations are as follows:

Processor	:	intel i3 processor
RAM	:	2 GB
Hard Disk	:	500 GB
Mouse	:	Optical Mouse
Network Interface card	:	Present

Software

- Each system will be connect with internet to connect to linux server to update the OS and Java version.
- Systems are provided for students in the 1:1 ratio.
- Systems are assigned numbers and same system is allotted for students when they do the lab.

STANDARD OPERATING PROCEDURE – SOP

a) Explanation on experiment by the concerned faculty covering the following aspects:

1) Name of the experiment/Aim
2) Software/Hardware required
3) Algorithm
4) Source Program
5) Test Data

c) Compiling and execution of the program

Writing of the experiment in the Observation Book:

The students will write the experiment in the Observation book as per the following format:

a) Name of the experiment
b) Aim of the experiment
c) Algorithm
d) Source Program
e) Test Data
f) Results for different data sets
g) Errors observed (if any) during compilation/execution
h) Signature of the Faculty

Guidelines to Students

- Equipment in the lab for the use of student community. Students need to maintain a proper decorum in the computer lab. Students must use the equipment with care. Any damage is caused is punishable.

- Students are required to carry their observation with completed exercises while entering the lab. Those who don't have observation book, are not allowed to enter the lab

- Students are supposed to occupy the machines allotted to them and are not supposed to talk or make noise in the lab.

- Lab can be used in free time / lunch hours by the students who need to use the systems should take prior permission from the lab in-charge.

- Lab records need to be submitted on or before date of submission.

- Students are not supposed to use floppy disks / pen drives.

List of Lab Exercises

Syllabus Programs

S. No	Name of the program
1	Create a php webpage and print “hello world”.
2	Write a PHP program to swap two numbers.
3	Write a PHP Program to demonstrate the variable function: Gettype():
4	Write a PHP Program to demonstrate the variable unction: unset()
5	Give the example of string function: substr()
6	Give the example of string function: strcmp()
7	Write a PHP program to create a database using MySQL.
8	Write a PHP program to create a table in MySQL.

1. Create a php webpage and print “hello world”.

Aim:-

To Create a php webpage and print “hello world”.

Algorithm

Step 1:
Create a PHP File:

Create a new file named hello.php.
Step 2:Write PHP Code:

Open hello.php and enter the following code:

```
<?php
echo "Hello, World!";
?>
```

Step 3:
Save the File:

Save and close hello.php.
Place the File in the Web Server Directory:

Step 4:
Move hello.php to your web server’s root directory (, htdocs for XAMPP).
Access the File in a Browser:

Step 5:
Open your web browser and navigate to http://localhost/hello.php to view "Hello, World!" on the page.

Program:

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

<h1>My first PHP page</h1>
```

```
<?php
echo ("Hello World!");
?>

</body>
</html>
```

Output:

My first PHP page
Hello World!

2.Write a PHP program to swap two numbers.

Aim:
To Create a PHP program to swap two numbers.
Algorithm
Step1:
Initialize Variables:

Set two variables with initial values for the numbers you want to swap.
Step2:
Display Initial Values:

Print the values of the two variables before swapping.

Step3:
Perform the Swap:

Use a temporary variable to hold the value of one variable while swapping.
Step 4:
Display Swapped Values:

Print the values of the two variables after swapping.
Step 5:
End Program:

Finish the PHP script.

Program:

```
<html>
<head>
<title>Swapping of Two numbers</title>
</head>
<body>
<?php
$a = 15;
$b = 27;

// Swap Logic

echo "\nThe number before swapping is:\n";
echo "Number a =".$a." and b=".$b;

$temp = $a;
$a = $b;
$b = $temp;

echo "\nThe number after swapping is: \n";
echo "Number a =".$a." and b=".$b."\n";
?>
</body>
</html>
```

Output:

The number before swapping is:
Number a =15 and b=27
The number after swapping is:

Number a =27 and b=15

3. Write a PHP Program to demonstrate the variable function: Gettype():

Aim:
To create a PHP Program to demonstrate the variable function: Gettype():
Algorithm:

Step1:
Initialize Variables:

Define variables with different types of data (e.g., integer, string, array).
Step2:
Use gettype() Function:

Apply the gettype() function to each variable to determine and retrieve its type.
Step3:
Display Variable Types:

Print the type of each variable using the results from gettype().
Step4:
End Program:

Complete the PHP script.

Program:

```
<html>
<head>
<title>Gettype() Function</title>
</head>
<body>
<?php
echo gettype(102).'<br>';
echo gettype(true).'<br>';
echo gettype(' ').'<br>';
```

```
echo gettype(null).'<br>';
echo gettype(array()).'<br>';
echo gettype(new stdclass());
?>
</body>
</html>
```

Output:

```
integer
boolean
string
NULL
array
object
```

4.Write a PHP Program to demonstrate the variable unction: unset()

Aim:

To create a PHP Program to demonstrate the variable unction: unset()

Algorithm

Step1:

Initialize Variables:

Define variables with some initial values.

Step2:

Display Initial Values:

Print the values of the variables before using unset().

Step3:

Use unset() Function:

Apply the unset() function to one or more variables to remove them.

Step4:
Attempt to Display Unset Variables:

Try to print the values of the variables after they have been unset.
Step5:
End Program:

Complete the PHP script.
Program:

```
<html>
<head>
<title>Unset Function</title>
</head>
<body>
<?php
function unset_val1()
{
global $val1;
echo $val1;
unset($val1);
}
$val1 = "Bipin";
unset_val1();
?>
</body>
</html>
```

Output:

Bipin

5.Give the example of string function: substr()

Aim:
To create a PHP program for string function: substr()
Algorithm
Step1:
Initialize a String:

Define a string from which you want to extract a substring.
Step2:
Use substr() Function:

Apply the substr() function to the string with appropriate parameters to extract a part of it.
Step3:
Display the Extracted Substring:

Print the substring to show the result of the extraction.
Step4:
End Program:

Complete the PHP script.

Program:

```
<html>
<head>
<title>Sub string Function</title>
</head>
<body>
<?php
$string1="Welcome to Rathinamcollege.com";
echo $string1;
echo '<br>';
echo substr($string1,1);
echo '<br>';
echo substr($string1,1,5);
echo '<br>';
echo substr($string1,0,10);
```

```
echo '<br>';
echo substr($string1,-1,1);
echo '<br>';
?>
</body>
</html>
```

Output:

```
Welcome to Rathinamcollege.com
athinamcollege.com
athin
Welcome   to
m
```

6.Give the example of string function: strcmp()

Aim:

To create a PHP program for using the string function: strcmp()

Algorithm

Step1:

Initialize Strings:

Step2:

Define two strings to be compared.

Step3:

Use strcmp() Function:

Apply the strcmp() function to compare the two strings.

Step4:

Display Comparison Result:

Print the result of the comparison to show whether the strings are equal, or which one is greater.

Program:

```
<html>
<head>
<title>String Compare Function</title>
</head>
<body>
<?php
$str1 = 'a';
$str2 = 'b';
echo strcmp($str1, $str2)."<br>";
$str3 = 'b';
$str4 = 'a';
echo strcmp($str3, $str4)."<br>";
$str5 = "Anil";
$str6 = "anil";
echo strcmp($str5, $str6)."<br>";
?>
</body>
</html>
```

OUTPUT

```
-1
1
-1
```

7. Write a PHP program to create a database using MySQL.

Aim:

PHP program to create a database using MySQL.

Algorithm

Step1:
Establish a Connection:

Connect to the MySQL server using PHP's mysqli_connect() function.

Step 2:
Check the Connection:

Verify that the connection is successful.

Step 3:
Create the Database:

Use SQL CREATE DATABASE command to create a new database.
Step4:
Check for Errors:

Verify that the database creation was successful and handle any errors.
Step5:
Close the Connection:

Close the connection to the MySQL server.

Program:

```
<!DOCTYPE html>
<html>
<head>
   <title>Create a Database using MySQL</title>
</head>
<body>
<?php
$dsn = 'mysql:host=localhost';
$username = 'root';
$password = '';

try {
   // Create a new PDO instance
   $pdo = new PDO($dsn, $username, $password);
   // Set the PDO error mode to exception
   $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

   // SQL query to create a database
   $query = "CREATE DATABASE std";

   // Execute the query
   $pdo->exec($query);

   echo "Database created successfully!";
} catch (PDOException $e) {
   echo "Error creating database: " . $e->getMessage();
}
```

```
// Close the connection (PDO doesn't require explicit closing)
?>
</body>
</html>
```

Output:
Database created successfully!

8. Write a PHP program to create a table in MySQL.

Aim:
PHP program to create a table in MySQL.
Algorithm
Stpep1:
Establish a Connection:

Connect to the MySQL server using PHP's mysqli_connect() function.
Step2:
Select the Database:

Choose the database where you want to create the table using mysqli_select_db().
Step3:
Define the SQL Query:

Write the SQL CREATE TABLE statement to define the table structure.
Step 4:
Execute the SQL Query:

Run the query using mysqli_query() to create the table.
Step 5:
Check for Errors:

Verify if the table creation was successful and handle any errors.
Close the Connection:

Close the connection to the MySQL server.

Program:

```
<!DOCTYPE html>
<html>
<head>
   <title>Create a Database and Table</title>
</head>
<body>
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "studinfo";

// Create a new MySQLi connection
$con = new mysqli($servername, $username, $password);

// Check connection
if ($con->connect_error) {
   die("Connection failed: " . $con->connect_error);
}
echo "Connection open" . "<br>";

// Check if the database exists
$result = $con->query("SHOW DATABASES LIKE '$dbname'");

if ($result->num_rows == 0) {
   // Database does not exist, so create it
   $createDbQuery = "CREATE DATABASE $dbname";
   if ($con->query($createDbQuery) === TRUE) {
      echo "Database created successfully" . "<br>";
   } else {
      die("Error creating database: " . $con->error);
   }
} else {
   echo "Database already exists" . "<br>";
}

// Select the database
```

```
$con->select_db($dbname);

// SQL query to create a table
$query = "CREATE TABLE IF NOT EXISTS computer (
   id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
   name VARCHAR(50),
   branch VARCHAR(50)
)";

// Execute the query
if ($con->query($query) === TRUE) {
   echo "Table created successfully!" . "<br>";
} else {
   die("Error creating table: " . $con->error);
}

// Close the connection
$con->close();
?>
</body>
</html>
```

Output:

Connection open
Database created successfully
Table created successfully!