Dr.SNS RAJALAKSHMI COLLEGE OF ARTS AND SCIENCE
(Autonomous)
Coimbatore -641049
Accredited by NAAC (Cycle- III)with 'Á+' Grade
(Recognised by UGC, Approved by AICTE, New Delhi and
Affiliated to Bharathiar University , Coimbatore

PHP LECTURE NOTES: UNIT-III

PREPARED BY

Dr.A.DEVI

Associate Professor

Department of Computer Applications

DR.SNSRACS

➢ Files

Working with Files

- File handling is an important part of any web application. You often need to open and process a file for different tasks.

PHP Manipulating Files

PHP has several functions for creating, reading, uploading, and editing files.

Be careful when manipulating files!

- When you are manipulating files you must be very careful.
- You can do a lot of damage if you do something wrong. Common errors are: editing the wrong file, filling a hard-drive with garbage data, and deleting the content of a file by accident.

PHP readfile() Function

- The readfile() function reads a file and writes it to the output buffer.
- Assume we have a text file called **"webdictionary.txt",** stored on the server, that looks like this:

AJAX = Asynchronous JavaScript and XML
CSS = Cascading Style Sheets
HTML = Hyper Text Markup Language
PHP = PHP Hypertext Preprocessor SQL = Structured Query Language SVG = Scalable Vector Graphics
XML = EXtensible Markup Language

- The PHP code to read the file and write it to the output buffer is as follows (the readfile() function returns the number of bytes read on success):

Example

```
<html>
<body>
<?php
echo readfile("webdictionary.txt");
?>
</body>
</html>
```

OUTPUT:

AJAX = Asynchronous JavaScript and XML
CSS = Cascading Style Sheets
HTML = Hyper Text Markup Language
PHP = PHP Hypertext Preprocessor
SQL = Structured Query Language

SVG = Scalable Vector Graphics
XML = EXtensible Markup Language236.

- The readfile() function is useful if all you want to do is open up a file and read itscontents.

File Open/Read/Close
PHP Open File - fopen()

- A better method to open files is with the fopen() function. This function gives youmore options than the readfile() function.

We will use the text file, **"webdictionary.txt",** during the lessons:
AJAX = Asynchronous JavaScript and XML
CSS = Cascading Style Sheets
HTML = Hyper Text Markup Language
PHP = PHP Hypertext Preprocessor
SQL = Structured Query Language
SVG = Scalable Vector Graphics
XML = EXtensible Markup Language

The first parameter of fopen() contains the name of the file to be opened and the second parameter specifies in which mode the file should be opened. The following example also generates a message if the fopen() function is unable to open the specified file:

Example

```
<?php
$myfile = fopen("webdictionary.txt", "r") or die("Unable to open file!");
echo fread($myfile,filesize("webdictionary.txt"));
fclose($myfile);
?>
```

OUTPUT:
AJAX = Asynchronous JavaScript and XML CSS = Cascading Style Sheets HTML = Hyper Text Markup Language PHP = PHP Hypertext Preprocessor SQL = Structured Query Language SVG = Scalable Vector Graphics XML = EXtensible Markup Language

The file may be opened in one of the following modes:

Modes	Description
r	Open a file for read only. File pointer starts at the beginning of the file
w	Open a file for write only. Erases the contents of the file or creates a new file if it doesn't exist. File pointer starts at the beginning of the file
a	Open a file for write only. The existing data in file is preserved. File pointer starts at the end of the file. Creates a new file if the file doesn't exist
x	Creates a new file for write only. Returns FALSE and an error if file already exists

r+	Open a file for read/write. File pointer starts at the beginning of the file
w+	Open a file for read/write. Erases the contents of the file or creates a new file if it doesn't exist. File pointer starts at the beginning of the file
a+	Open a file for read/write. The existing data in file is preserved. File pointer starts at the end of the file. Creates a new file if the file doesn't exist
x+	Creates a new file for read/write. Returns FALSE and an error if file already Exists

PHP Read File - fread()

- The fread() function reads from an open file.
- The first parameter of fread() contains the name of the file to read from and thesecond parameter specifies the maximum number of bytes to read.
- The following PHP code reads the "webdictionary.txt" file to the end:

```
fread($myfile,filesize("webdictionary.txt"));
```

PHP Close File - fclose()

- The fclose() function is used to close an open file.
- It's a good programming practice to close all files after you have finished withthem. You don't want an open file running around on your server taking up resources!
- The fclose() requires the name of the file (or a variable that holds the filename) we want to close:

```
<?php
$myfile = fopen("webdictionary.txt","r"); // some code to be executed....
fclose($myfile);
?>
```

PHP Read Single Line - fgets()

- The fgets() function is used to read a single line from a file.
- The example below outputs the first line of the "webdictionary.txt" file:

Example

```
<?php
$myfile = fopen("webdictionary.txt", "r") or die("Unable to open file!");
echo fgets($myfile);
fclose($myfile);
?>
```

OUTPUT:

AJAX = Asynchronous JavaScript and XML

After a call to the fgets() function, the file pointer has moved to the next line.

PHP Check End-Of-File - feof()

- The feof() function checks if the "end-of-file" (EOF) has been reached.
- The feof() function is useful for looping through data of unknown length.
- The example below reads the "webdictionary.txt" file line by line, until end-of-file is reached:

Example

```
<?php
$myfile = fopen("webdictionary.txt", "r") or die("Unable to open file!");
// Output one line until end-of-file
while(!feof($myfile))
{
  echo fgets($myfile) . "<br>";
}
fclose($myfile);
?>
```

OUTPUT:

AJAX = Asynchronous JavaScript and XML
CSS = Cascading Style Sheets
HTML = Hyper Text Markup Language
PHP = PHP Hypertext Preprocessor SQL = Structured Query Language SVG = Scalable Vector Graphics
XML = EXtensible Markup Language

PHP Read Single Character - fgetc()

- The fgetc() function is used to read a single character from a file.
- The example below reads the "webdictionary.txt" file character by character, untilend-of-file is reached:

Example

```
<?php
$myfile = fopen("webdictionary.txt", "r") or die("Unable to open file!");
// Output one character until end-of-file
while(!feof($myfile))
  {                    echo
  fgetc($myfile);
}
fclose($myfile);
?>
```

OUTPUT:

AJAX = Asynchronous JavaScript and XML CSS = Cascading Style Sheets HTML = Hyper Text Markup Language PHP = PHP Hypertext Preprocessor SQL = Structured Query Language SVG = Scalable Vector Graphics XML = EXtensible Markup Language

- After a call to the fgetc() function, the file pointer moves to the next character.

File Create/Write

PHP Create File - fopen()

- The fopen() function is also used to create a file. Maybe a little confusing, but inPHP, a file is created using the same function used to open files.
- If you use fopen() on a file that does not exist, it will create it, given that the file is opened for writing (w) or appending (a).

- The example below creates a new file called "testfile.txt". The file will be createdin the same directory where the PHP code resides:

Example

```
$myfile = fopen("testfile.txt", "w")
```

PHP File Permissions

If you are having errors when trying to get this code to run, check that you havegranted your PHP file access to write information to the hard drive.

PHP Write to File - fwrite()

- The fwrite() function is used to write to a file.
- The first parameter of fwrite() contains the name of the file to write to and thesecond parameter is the string to be written.
- The example below writes a couple of names into a new file called "newfile.txt":

Example

```
<?php
$myfile = fopen("newfile.txt", "w") or die("Unable to open
file!"); $txt = "John Doe\n";
fwrite($myfile, $txt);
$txt = "Jane Doe\n";
fwrite($myfile, $txt);
fclose($myfile);
?>
```

Notice that we wrote to the file "newfile.txt" twice. Each time we wrote to the file we sent the string $txt that first contained "John Doe" and second contained "Jane Doe". Afterwe finished writing, we closed the file using the fclose() function.

If we open the "newfile.txt" file it would look like this:

John Doe
Jane Doe

PHP Overwriting

- Now that "newfile.txt" contains some data we can show what happens when we open an existing file for writing. All the existing data will be ERASED and we startwith an empty file.
- In the example below we open our existing file "newfile.txt", and write some new data into it:

Example

```
<?php
$myfile = fopen("newfile.txt", "w") or die("Unable to open
file!"); $txt = "Mickey Mouse\n";
fwrite($myfile, $txt);
```

```
$txt = "Minnie Mouse\n";
fwrite($myfile, $txt);
fclose($myfile);
?>
```

If we now open the "newfile.txt" file, both John and Jane have vanished, and only thedata we just wrote is present:

OUTPUT:

Mickey Mouse
Minnie Mouse

PHP ftp_fput() Function

Example

- Open local file, and upload it to a file on the FTP server:

```
<?php
// connect and login to FTP server
$ftp_server = "ftp.example.com";
$ftp_conn = ftp_connect($ftp_server) or die("Could not connect to $ftp_server"); $login = ftp_login($ftp_conn, $ftp_username, $ftp_userpass);

// open file for reading
$file = "test.txt";
$fp = fopen($file,"r");

// upload file
if (ftp_fput($ftp_conn, "somefile.txt", $fp, FTP_ASCII))
  {
  echo "Successfully uploaded $file.";
  }
else
{
  echo "Error uploading $file.";
  }

// close this connection and file handler
ftp_close($ftp_conn);
fclose($fp);
?>
```

Definition and Usage

- The ftp_fput() function uploads from an open file and saves it to a file on theFTP server.

Syntax

ftp_fput(*ftp_connection,remote_file,open_file,mode,startpos*);

Parameter	Description
ftp_connection	Required. Specifies the FTP connection to use
remote_file	Required. Specifies the file path to upload to

open_file	Required. Specifies an open local file. Reading stops at end of file
mode	Required. Specifies the transfer mode. Possible values: FTP_ASCII or FTP_BINARY
startpos	Optional. Specifies the position in the remote file to start uploading to

Working with Databases

PHP MySQL Database

- With PHP, you can connect to and manipulate databases.
- MySQL is the most popular database system used with PHP.

What is MySQL?

- MySQL is a database system used on the web
- MySQL is a database system that runs on a server
- MySQL is ideal for both small and large applications
- MySQL is very fast, reliable, and easy to use
- MySQL uses standard SQL
- MySQL compiles on a number of platforms
- MySQL is free to download and use
- MySQL is developed, distributed, and supported by Oracle Corporation

The data in a MySQL database are stored in tables. A table is a collection of relateddata, and it consists of columns and rows.

Databases are useful for storing information categorically. A company may have a database with the following tables:

- Employees
- Products
- Customers
- Orders

PHP + MySQL Database System

- PHP combined with MySQL are cross-platform (you can develop in Windows and serve on a Unix platform)

PHP Connect to MySQL

PHP 5 and later can work with a MySQL database using:

- **MySQLi extension** (the "i" stands for improved)
- **PDO (PHP Data Objects)**

Earlier versions of PHP used the MySQL extension. However, this extension was deprecated in 2012.

Should I Use MySQLi or PDO?

If you need a short answer, it would be "Whatever you like".Both MySQLi and PDO have their advantages:

- PDO will work on 12 different database systems, whereas MySQLi will only work with MySQL databases.
- So, if you have to switch your project to use another database, PDO makes the process easy. You only have to change the connection string and a few queries. With MySQLi, you will need to rewrite the entire code - queries included.
- Both are object-oriented, but MySQLi also offers a procedural API.
- Both support Prepared Statements. Prepared Statements protect from SQL injection, and are very important for web application security.

MySQL Examples in Both MySQLi and PDO

Syntax

In this, and in the following chapters we demonstrate three ways of working with PHP and MySQL:

- MySQLi (object-oriented)
- MySQLi (procedural)
- PDO

MySQLi Installation

- For Linux and Windows: The MySQLi extension is automatically installed in most cases, when php5 mysql package is installed.
- For installation details, go to: http://php.net/manual/en/mysqli.installation.php
- PDO Installation
- For installation details, go to: http://php.net/manual/en/pdo.installation.php

Open a Connection to MySQL

- Before we can access data in the MySQL database, we need to be able to connect to the server:

Example (MySQLi Object-Oriented)

```
<?php
$servername = "localhost";
$username  = "username";
$password = "password";

// Create connection
$conn = new mysqli($servername, $username, $password);

// Check connection
if ($conn->connect_error) {
   die("Connection failed: " . $conn->connect_error);
}
echo    "Connected
successfully"; ?>
```

Example (MySQLi Procedural)

```
<?php
$servername = "localhost";
$username  = "username";
$password = "password";

// Create connection
$conn = mysqli_connect($servername, $username, $password);

//  Check  connection
if (!$conn) {
   die("Connection failed: " . mysqli_connect_error());
}
echo    "Connected
successfully"; ?>
```

Example (PDO)

```
<?php
$servername = "localhost";
$username  = "username";
$password = "password";

try {
   $conn = new PDO("mysql:host=$servername;dbname=myDB",
$username, $password);
   // set the PDO error mode to exception $conn-
   >setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION); echo
   "Connected successfully";
   }
catch(PDOException $e)
```

```
    {
    echo "Connection failed: " . $e->getMessage();
    }
?>
```

Tip: A great benefit of PDO is that it has an exception class to handle any problems that may occur in our database queries. If an exception is thrown within the try{ } block, the script stops executing and flows directly to the first catch(){ } block.

Close the Connection

The connection will be closed automatically when the script ends. To close the connection before, use the following:

Example (MySQLi Object-Oriented)

```
$conn->close();
```

Example (MySQLi Procedural)

```
mysqli_close($conn);
```

Example (PDO)

```
$conn = null;
```

PHP Create a MySQL Database

- A database consists of one or more tables.
- You will need special CREATE privileges to create or to delete a MySQL database.

Create a MySQL Database Using MySQLi

- The CREATE DATABASE statement is used to create a database in MySQL.
- The following examples create a database named "myDB":

Example (MySQLi Object-oriented)

```
<?php
$servername = "localhost";
$username  = "username";
$password = "password";

// Create connection
$conn = new mysqli($servername, $username,
$password);  //  Check  connection if
($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Create database
$sql = "CREATE DATABASE myDB";
if ($conn->query($sql) === TRUE) {
    echo    "Database    created
successfully"; } else {
    echo "Error creating database: " . $conn->error;
}
$conn->close();
?>
```

Note: When you create a new database, you must only specify the first three arguments to the mysqli object (servername, username and password).

Tip: If you have to use a specific port, add an empty string for the database-name argument, like this: new mysqli("localhost", "username", "password", "", port)

PHP Create MySQL Table

- A database table has its own unique name and consists of columns and rows.
- Create a MySQL Table Using MySQLi
- The CREATE TABLE statement is used to create a table in MySQL.
- We will create a table named "MyGuests", with five columns: "id", "firstname", "lastname", "email" and "reg_date":

CREATE TABLE MyGuests (id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY, firstname VARCHAR(30) NOT NULL,lastname VARCHAR(30) NOTNULL, email VARCHAR(50), reg_date TIMESTAMP)

Notes on the table above:

- The data type specifies what type of data the column can hold. For a complete reference of all the available data types.

After the data type, you can specify other optional attributes for each column:

- NOT NULL - Each row must contain a value for that column, null values are not allowed
- DEFAULT value - Set a default value that is added when no other value is passed
- UNSIGNED - Used for number types, limits the stored data to positive numbers and zero
- AUTO INCREMENT - MySQL automatically increases the value of the field by 1 each time a new record is added
- PRIMARY KEY - Used to uniquely identify the rows in a table. The column with PRIMARY KEY setting is often an ID number, and is often used with AUTO_INCREMENT

Each table should have a primary key column (in this case: the "id" column). Its value must be unique for each record in the table.

The following examples shows how to create the table in PHP:

Example (MySQLi Object-oriented)

```
<?php
$servername = "localhost";
$username = "username";
```

```
$password = "password";
$dbname = "myDB";

// Create connection
$conn = new mysqli($servername, $username, $password,
$dbname);  //  Check  connection if
($conn->connect_error) {
   die("Connection failed: " . $conn->connect_error);
}

// sql to create table
$sql = "CREATE TABLE MyGuests (id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
firstname VARCHAR(30) NOT NULL,lastname VARCHAR(30) NOT NULL, email
VARCHAR(50),reg_date TIMESTAMP)";

if ($conn->query($sql) == TRUE)
{
echo "Table MyGuests created successfully";
}
else
{
   echo "Error creating table: " . $conn->error;
}

$conn->close(); ?>
```

PHP Insert Data Into MySQL

Insert Data Into MySQL Using MySQLi

- After a database and a table have been created, we can start adding data inthem.

Here are some syntax rules to follow:

- The SQL query must be quoted in PHP
- String values inside the SQL query must be quoted
- Numeric values must not be quoted
- The word NULL must not be quoted

The INSERT INTO statement is used to add new records to a MySQL table:
INSERT INTO table_name (column1, column2, column3,...)
VALUES (value1, value2, value3,...)

Note: If a column is AUTO_INCREMENT (like the "id" column) or TIMESTAMP (like the"reg_date" column), it is no need to be specified in the SQL query; MySQL will automatically add the value.

The following examples add a new record to the "MyGuests" table:

Example (MySQLi Object-oriented)

```php
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";

// Create connection
$conn = new mysqli($servername, $username, $password,$dbname); // Check connection
if ($conn->connect_error)
{
   die("Connection failed: " . $conn->connect_error);
}

$sql = "INSERT INTO MyGuests (firstname, lastname, email) VALUES
('John', 'Doe', 'john@example.com')";

if ($conn->query($sql) === TRUE)
{echo "New record created
successfully";
}
Else
 {
   echo "Error: " . $sql . "<br>" . $conn->error;
}
$conn->close(); ?>
```

Example:

Student Application using PHP and Mysql

aa.html:

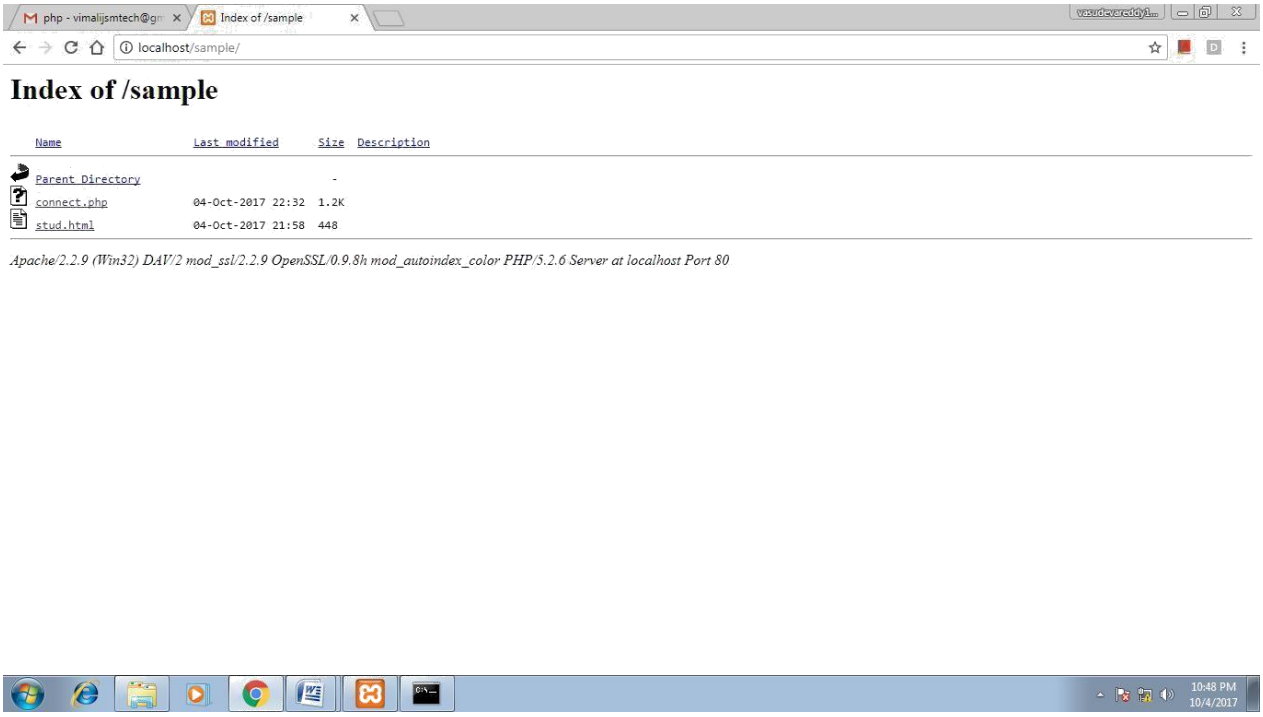
```
<!DOCTYPE HTML>
<html>

<head>

<title>Student
Table</title> </head>
<body>

<div id="dept"> <h3 align=center>Student table entry</h3>
<form method="POST" action="connect.php">
SName <br><input type="text" name="sname"><br>
Reg.No <br><input type="text" name="regno"><br>
Mark1<br><input type="text" name="m1"><br>
Mark2<br><input type="text" name="m2"><br>
<inputtype="submit" value="ok">
</form>
</div>
</body>
</html>
```

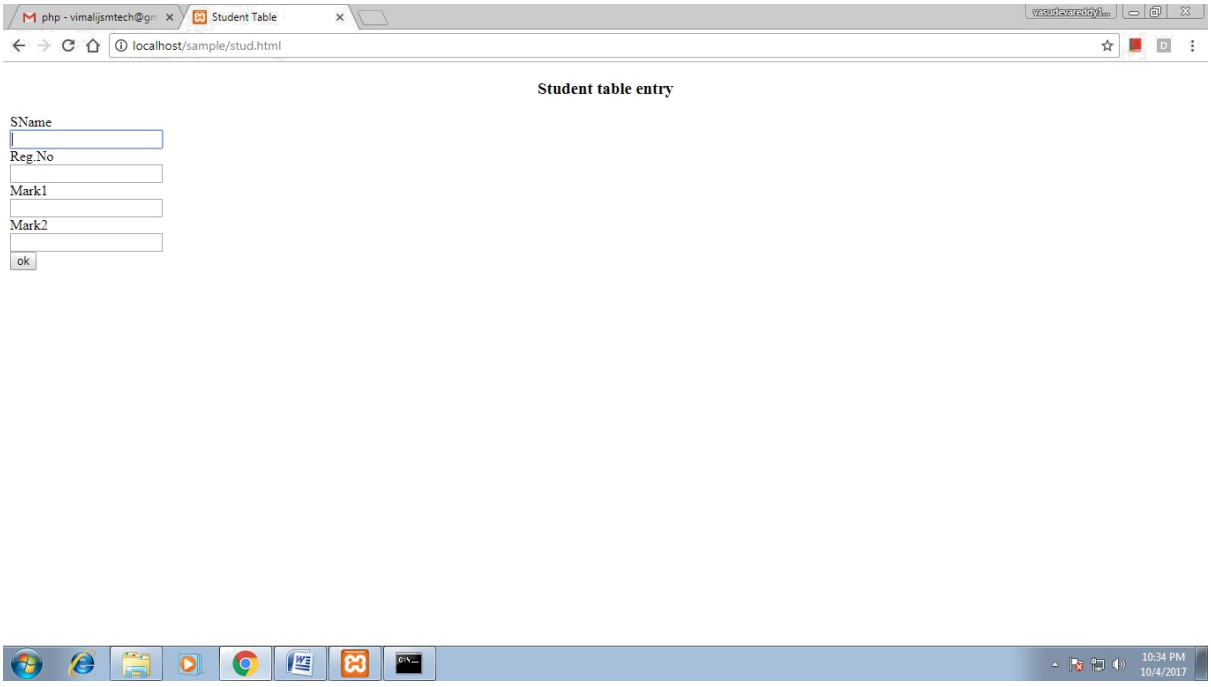
Conn.php:

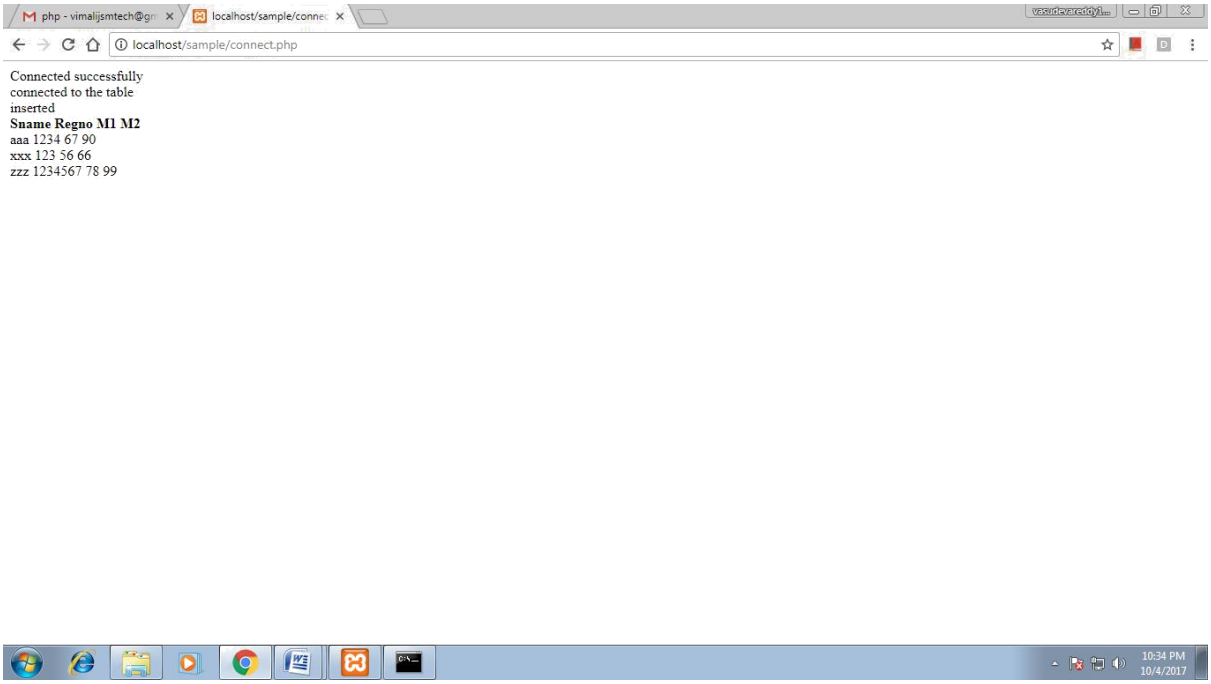
```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "sample";
// Create connection
$conn = new mysqli($servername, $username, $password,$dbname);
// Check connection
if ($conn->connect_error)

{

    die("Connection failed: " . $conn->connect_error);

}

echo "Connected successfully";
//connect table
$sql="desc student"; if($conn->query($sql)==TRUE)

{

echo "<br>";

 echo "connected to the table";

}

else

{

echo "error";

}

//Inserting the contents
echo "<br>";
//insertion from html

$sname=$_POST['sname'];

$regno=$_POST['regno'];

$m1=$_POST['m1'];

$m2=$_POST['m2'];

$sql11="insert into student values('$sname',$regno,$m1,$m2)";
if($conn->query($sql11)==TRUE){
```

```
echo "inserted";

}

else

{echo  "error";}
echo "<br>";
$sql1="select * from student";

$result  =  $conn->query($sql1);
if ($result->num_rows > 0) {
    // output data of each row

echo  "<b>Sname  Regno  M1  M2</b><br>";
    while($row = $result->fetch_assoc()) {
        echo $row["sname"]." ". $row["regno"]." ".$row["m1"]." ".$row["m2"]."<br>";
    }

} else {

    echo "empty table";

}

$conn->close(); ?>
```

OUTPUT:


php - vimalijsmtech@gm
Index of /sample
localhost/sample/
Index of /sample
Name
Last modified
Size
Description
Parent Directory
-
connect.php
04-Oct-2017 22:32
1.2K
stud.html
04-Oct-2017 21:58
448
Apache/2.2.9 (Win32) DAV/2 mod_ssl/2.2.9 OpenSSL/0.9.8h mod_autoindex_color PHP/5.2.6 Server at localhost Port 80
10:48 PM
10/4/2017


php - vimalijsmtech@gm
Student Table
localhost/sample/stud.html
Student table entry
SName
Reg.No
Mark1
Mark2
ok
10:34 PM
10/4/2017


php - vimalijsmtech@gm
localhost/sample/connec
localhost/sample/connect.php
Connected successfully
connected to the table
inserted
Sname Regno M1 M2
aaa 1234 67 90
xxx 123 56 66
zzz 1234567 78 99
10:34 PM
10/4/2017