Unit 5 Establishing a Database Connection and Working With Database

Database

- A database management system (DBMS) is system software for creating and managing database.
- The DBMS provides users and programmers with a systematic way to create, retrieve, update and manage data.
- Now-a-days, we use relational database management systems (RDBMS) to store and manage huge volume of data. This is called relational database because all the data is stored into different tables and relations are established using primary keys or foreign keys.
- A Relational Database Management System (RDBMS) is a software that:
  - Enables you to implement a database with tables, columns and indexes.
  - Guarantees the Referential Integrity between rows of various tables.
  - Updates the indexes automatically.
  - Interprets an SQL query and combines information from various tables.

RDBMS Terminology

- **Database:** A database is a collection of tables, with related data.
- **Table:** A table is a matrix with data. A table in a database looks like a simple spreadsheet.
- **Column:** In relational table, a column is a set of value of a particular type.
- **Row:** A row (tuple or record) is a group of related data.
- **Redundancy:** Storing data twice, redundantly to make the system faster.
- **Primary Key:** A primary key is used to uniquely identify the table record and cannot contain duplicate data and null value.
- **Foreign Key:** A foreign is used to derive value from primary key of other table.
- **Compound Key:** A compound key is a key that consists of multiple columns, because one column is not sufficiently unique.
- **Index:** A database index is a data structure that improves the speed of operations in a table.

What is phpMyAdmin?

- PhpMyAdmin is one of the most popular applications for MySQL databases management.
- It is a free tool written in PHP.
- Through this software you can create, alter, drop, delete, import and export MySQL database tables.
- You can run MySQL queries, optimize, repair and check tables, change collation and execute other database management commands.

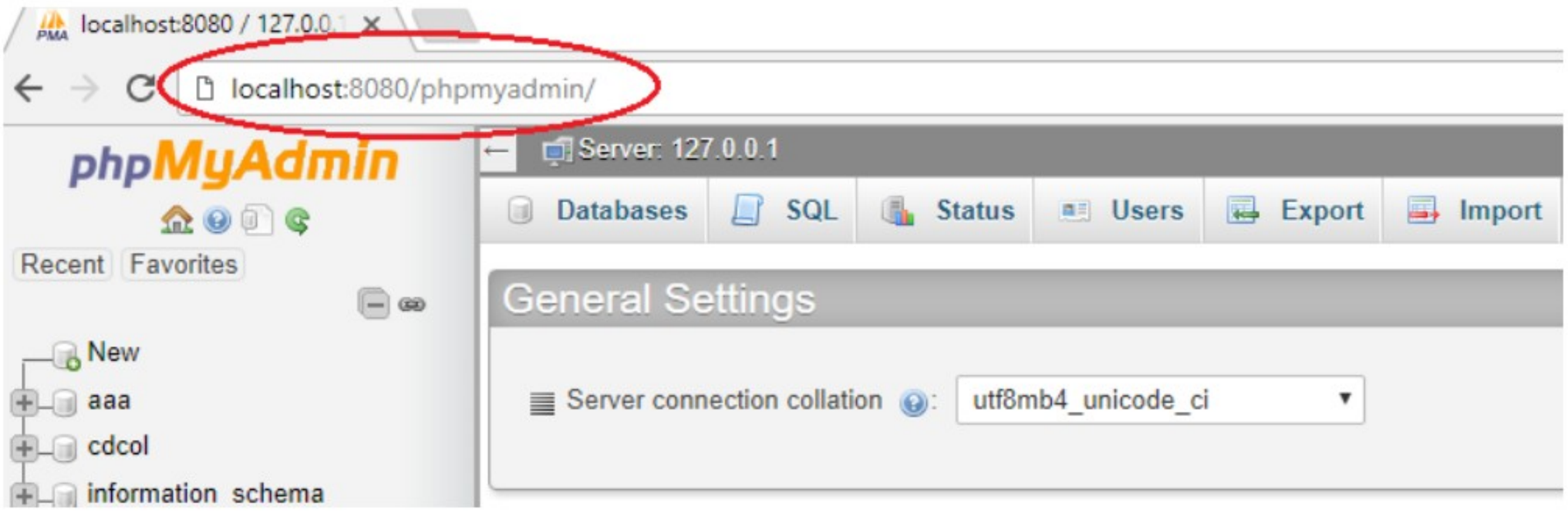
What is MySQL database? OR Introduction of MySQL Database

- MySQL is a freely available open source Relational Database Management System (RDBMS) that uses Structured Query Language (SQL).
- SQL is the most popular language for adding, accessing and managing content in a database.
- Standard SQL commands, such as ADD, DROP, INSERT, and UPDATE can be used with MySQL.
- MySQL can be used for a variety of applications, but is most commonly found on Web servers.

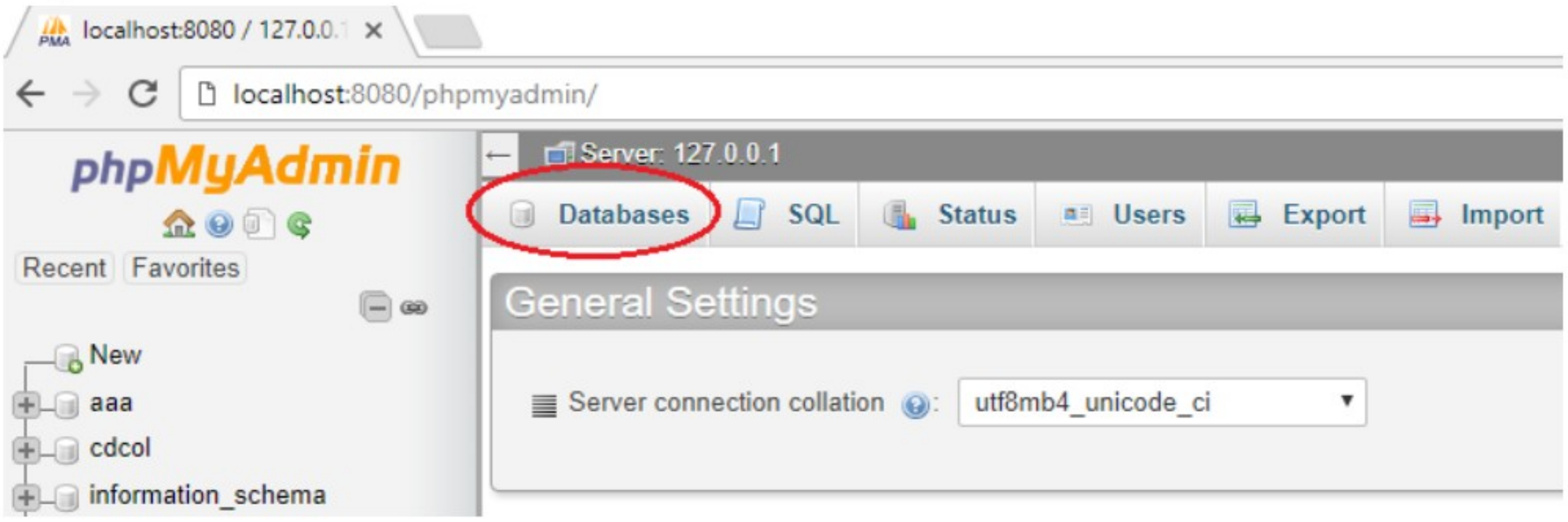
- A website that uses MySQL may include Web pages that access information from a database. These pages are often referred to as "dynamic," meaning the content of each page is generated from a database as the page loads.
- Websites that use dynamic Web pages are often referred to as database-driven websites.
- Many database-driven websites that use MySQL also use a Web scripting language like PHP to access information from the database.
- MySQL commands can be incorporated into the PHP code, allowing part or all of a Web page to be generated from database information. Because both MySQL and PHP are both open source (meaning they are free to download and use), the PHP/MySQL combination has become a popular choice for database-driven websites.

Creating Database using phpmyadmin

Step 1: Go to PHP myadmin: Open your browser and type http://localhost/phpmyadmin/ You will get following window,

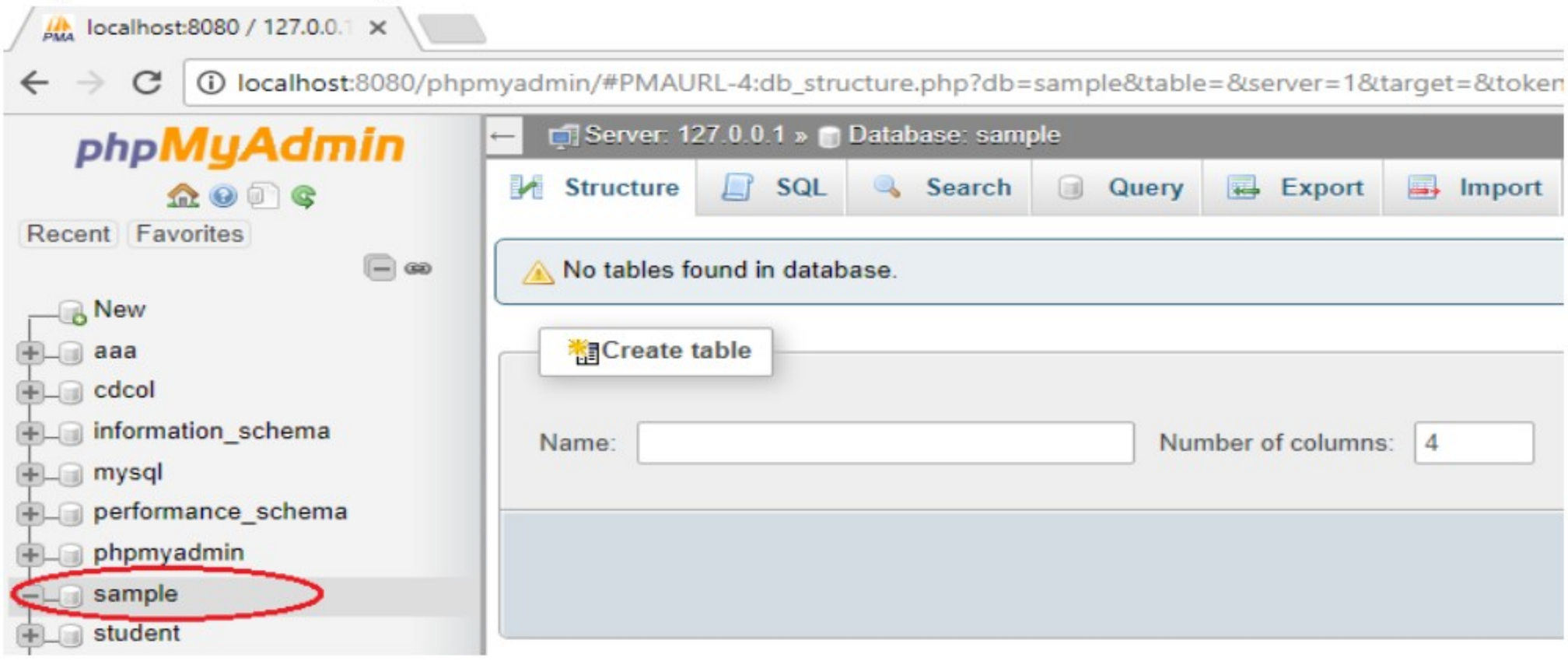


Step 2: Select Database menu item: Click on Database Manu where we marked.



Step 3: Create Database: Type your database name, then click 'create' Button.

Step 4: Database ending: Your database is created. You can find that one on leftside.



Creating Database using Console

- Login to the MySQL server from the command line with the following command:
  mysql -u root -p
- In this case, specified the user root with the -u flag, and then used the -p flag so MySQL prompts for a password. Enter your current password to complete the login.
- You should now be at a MySQL prompt that looks very similar to this:
  mysql>
- To create a database with the name test type the following command:
  CREATE DATABASE test;

Data types (field types) of MySQL

- MySQL uses many different data types broken into three categories:
  - Numeric
  - Date and Time
  - String Types

Numeric

Type	Use For	Size
TINYINT	A very small integer	The signed range is -128 to 127. The unsigned range is 0 to 255.
SMALLINT	A small integer	The signed range is -32768 to 32767. The unsigned range is 0 to 65535.
MEDIUMINT	A medium-sized integer	The signed range is -8388608 to 8388607. The unsigned range is 0 to 16777215.
INT	A normal-sized integer	The signed range is -2147483648 to 2147483647. The unsigned range is 0 to 4294967295.
BIGIN	A large integer	The signed range is -9223372036854775808 to 9223372036854775807. The unsigned range is 0 to 18446744073709551615.

Date and time

Type	Description	Example
DATE	A date in YYYY-MM-DD format, between 1000-01-01 and 9999-12-31.	December 30th, 1973 would be stored as 1973-12-30.
DATETIME	A date and time combination in YYYY-MM-DD HH:MM:SS format, between 1000-01-01 00:00:00 and 9999-12-31 23:59:59.	3:30 in the afternoon on December 30th, 1973 would be stored as 1973-12-30 15:30:00.
TIMESTAMP	A timestamp between midnight, January 1st, 1970 and sometime in 2037.	This looks like the previous DATETIME format, only without the hyphens between numbers; 3:30 in the afternoon on December 30th, 1973 would be stored as 19731230153000 (YYYYMMDDHHMMSS).
TIME	Stores the time in a HH:MM:SS format.	02:20:34.
YEAR	Stores a year in a 2-digit or a 4-digit format.	If the length is specified as 2 (for example YEAR(2)), YEAR can be between 1970 to 2069 (70 to 69). If the

		length is specified as 4, then YEAR can be 1901 to 2155. The default length is 4.

String Type

Type	Description
CHAR(size)	• A fixed-length string between 1 and 255 characters in length (for example CHAR(5). Defining a length is not required, but the default is 1.
VARCHAR(size)	• A variable-length string between 1 and 255 characters in length. For example, VARCHAR(25). You must define a length when creating a VARCHAR field. If you put a greater value than 255 it will be converted to a TEXT type.
TEXT	• A field with a maximum length of 65535 characters. • Fields defined as TEXT also hold large amounts of data. • TEXT is case insensitive when sorting and comparing.
BLOB	• BLOB is equal to a text field. • A field with a maximum length of 65535 characters. • BLOB is case sensitive when sorting and comparing.
ENUM	• It is used for listing. When defining an ENUM, you are creating a list of items from which the value must be selected (or it can be NULL). • For example, if you wanted your field to contain "A" or "B" or "C", you would define your ENUM as ENUM ('A', 'B', 'C') and only those values (or NULL) could ever populate that field.

Integration of PHP with MySQL

- It is possible to execute various commands of MySQL from PHP. PHP provides various built-in functions which you allows you to use MySQL commands from PHP page. Thus you can integrate PHP with MySQL.
- Following are the various PHP functions that allows you the facility of integrating PHP with MySQL.

mysql_connect()

- The mysql_connect() function opens a new connection to the MySQL server.
- Returns an object representing the connection to the MySQL server.

Syntax: mysql_connect(host, username, password)

Parameter	Description
host	Specifies a host name or an IP address
username	Specifies the MySQL username
password	Specifies the MySQL password

Example

```
<?php
    $conn=mysql_connect("localhost","root","");
    if(!$conn)
    {
        echo " Could not connect";
    }
?>
```

mysql_select_db()

- The mysql_select_db() function is used to change the default database for the connection.
- Returns TRUE on success and FALSE on failure.

Syntax: mysql_select_db(dbname)

Parameter	Description
dbname	Required. Specifies the default database to be used

Example

```
<?php
    $conn=mysql_connect("localhost","root","");
    $db=mysql_select_db("student");
    if(!$conn)
    {
        echo " Database error";
    }
?>
```

mysql_query()

- The mysql_query() function executes a query on a MySQL database.

Syntax: mysql_query(query)

Parameter	Description
query	Required. Specifies the query string

Example

```
<?php
    $conn=mysql_connect("localhost","root","");
    $db=mysql_select_db("student");
    $qry="select * from studentinfo";
    $result=mysql_query($qry);
    if(!$result) {
        echo "Database query failed";
    }
?>
```

mysql_num_rows()

- The mysql_num_rows() function returns the number of rows in a result set.

Syntax: mysql_num_rows(result)

Parameter	Description
result	Required. Specifies a result set identifier returned by mysql_query()

Example

```
<?php
    $conn=mysql_connect("localhost","root","");
    $db=mysql_select_db("student");
    $qry="select * from studentinfo";
    $result=mysql_query($qry);
    $rowcount=mysql_num_rows($result);
    echo "Total Row:".$rowcount;
?>
```

mysql_fetch_array()

- The mysql_fetch_array() function fetches a result row as an associative array, a numeric array, or both.
- Returns an array of strings that corresponds to the fetched row.
- NULL if there are no more rows in result-set.

Syntax: mysql_fetch_array(result)

Parameter	Description
result	Required. Specifies a result set identifier returned by mysql_query()

Example

```
<?php
    $conn=mysql_connect("localhost","root","");
    $db=mysql_select_db("student");
    $qry="select StudentID,Name,Email,Gender from studentinfo";
    $result=mysql_query($qry);
    while($row=mysql_fetch_array($result))
    {
        echo $row[0]."<br/>";
        echo $row['Name']."<br/>";
    }
?>
```

- **Note:** Fieldnames returned from this function are case-sensitive.

mysql_fetch_assoc()

- The mysql_fetch_assoc() function fetches a result row as an associative array.
- Returns an associative array of strings representing the fetched row.

- NULL if there are no more rows in result-set.

Syntax: mysql_fetch_assoc(result)

Parameter	Description
result	Required. Specifies a result set identifier returned by mysql_query()

Example

```
<?php
    $conn=mysql_connect("localhost","root","");
    $db=mysql_select_db("student");
    $qry="select StudentID,Name,Email,Gender from studentinfo";
    $result=mysql_query($qry);
    while($row=mysql_fetch_assoc($result))
    {
        echo $row['StudentID']."<br/>";
        echo $row['Name']."<br/>";
    }
?>
```

- **Note:** Fieldnames returned from this function are case-sensitive.

mysql_close()

- The mysql_close() function closes a previously opened database connection.

Parameter	Description
connection	Required. Specifies the MySQL connection to close

```
<?php
    $conn=mysql_connect("localhost","root","");
    if(!$conn)
    {
        echo " Could not connect to database ";
    }
    mysql_close($conn)
?>
```

mysql_error()

- The mysql_error() function used to get the error message from the last MySQL operation.

Parameter	Description
connection	Optional. Before performing any operation on a MySQL database, it is required to set a connection to the mysql database you want to work with. And this is done by mysql_connect() function.

```
<?php
    $conn=mysql_connect("localhost", "root","");
    $db=mysql_select_db("student");
```

```
        if(!$db)
        {
                echo "Something else wrong".mysql_error($conn);
                die;
        }
    ?>
```

Developing a sample web based Application

Config.php

```
<?php
        $Conn=mysqli_connect('localhost', 'root','','workingwithsql');
        if(!$Conn)
        {
                echo "Something else wrong".mysqli_error($Conn);
                die;
        }
?>
```

StudentList.php

```
<?php
        include_once 'config.php';
        if(isset($_GET['Mode']))
        {
                if($_GET['Mode']=='delete')
                {
                        $StudentID=$_GET['Studentid'];
                        $DelSQL="delete from studentdata where StudentID='$StudentID'";
                        $Result=mysqli_query($Conn,$DelSQL);
                        if($Result)
                        {
                                echo "Record Deleted Successfully";
                        }
                        else
                        {
                                echo "Opps!! Something went wrong".mysqli_error($Conn);
                        }
                }
        }
?>
```

```
<html>
	<head>
		<title>Student Registration</title>
	</head>
	<body>
		<div align="center">
			<a href="StudentAddEdit.php">Add New Student</a>
			<table border="1px";>
				<tr>
					<th>StudentName</th>
					<th>StudentEmail</th>
					<th>StudentContact</th>
					<th>StudentGender</th>
					<th>Action</th>
				</tr>
<?php
	$SelectStr="select * from studentdata";
	$Result=mysqli_query($Conn,$SelectStr);
	while($RowData=mysqli_fetch_assoc($Result))
	{
		//print_r($RowData);
		//exit;
?>
	<tr>
		<td><?php echo $RowData['StudentName']; ?></td>
		<td><?php echo $RowData['StudentEmail']; ?></td>
		<td><?php echo $RowData['StudentContact']; ?></td>
		<td><?php echo $RowData['StudentGender']; ?></td>
		<td><a href="StudentAddEdit.php?Mode=edit&Studentid=<?php
		echo $RowData['StudentID']; ?>">Update</a>  | <a href=
		"StudentList.php?Mode=delete&Studentid=<?php echo
		$RowData['StudentID']; ?>">Delete</a></td>
	</tr>
<?php
	}
?>
			</table>
		</div>
	</body>
```

```
</html>
```

StudentAddEdit.php

```
<?php
        include_once 'config.php';
        if(isset($_POST['submit']))
        {
                $StudentID=$_POST['StuID'];
                $Name=$_POST['StudentName'];
                $Email=$_POST['StudentEmail'];
                $Contact=$_POST['StudentContact'];
                $Gender=$_POST['StudentGender'];
                if(empty($Name))
                {
                        $ErrorMsg="Enter Name";
                }
                else if(empty($Email))
                {
                        $ErrorMsg="Enter Email";
                }
                else if(empty($Contact))
                {
                        $ErrorMsg="Enter Contact";
                }
                else if(empty($Gender))
                {
                        $ErrorMsg="Enter Gender";
                }
                else
                {
                        if((isset($StudentID))and !empty($StudentID))
                        {
                                $UpdateStr="Update studentdata set StudentName = '$Name',
                                StudentEmail ='$Email', StudentContact='$Contact', StudentGender=
                                '$Gender' where StudentID=$StudentID";

                                $Result=mysqli_query($Conn,$UpdateStr);
                                if($Result)
                                {
```

```
                echo "Data Updated Seccesfully";
            }
            else
            {
                echo "Error in Data Insertion".mysqli_error($Conn);
            }
    }
    else
    {
        $InsertStr="insert into studentdata (StudentName, StudentEmail,
        StudentContact, StudentGender) values ('$Name','$Email', '$Contact',
        '$Gender')";

        $Result=mysqli_query($Conn,$InsertStr);
        if($Result)
        {
            echo "Data Inserted Seccesfully";
        }
        else
        {
            echo "Error in Data Insertion".mysqli_error($Conn);
        }
            }
        }
    }
?>
```

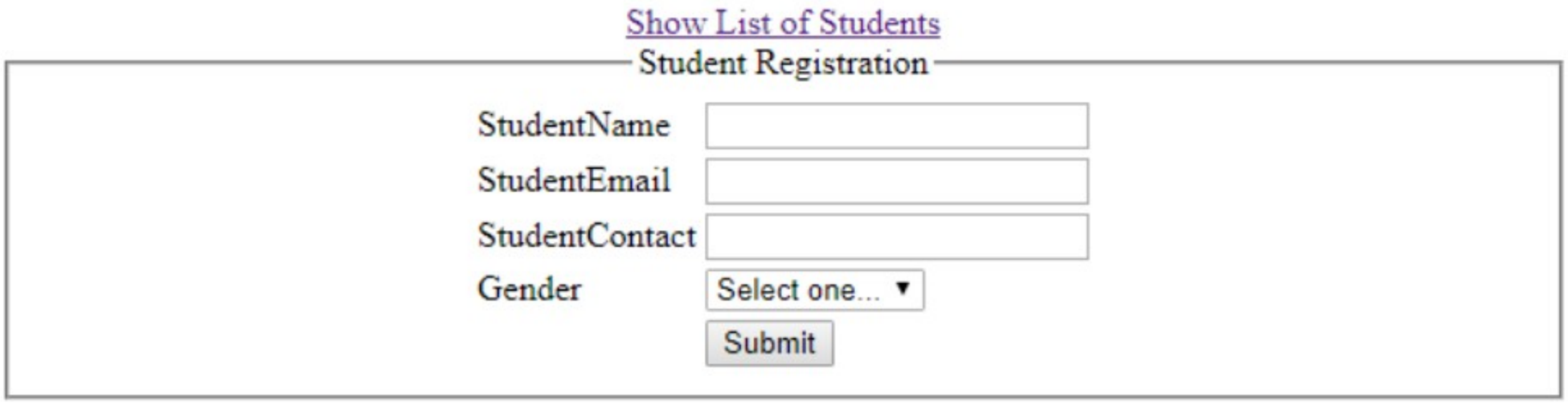
```
<?php
    if(isset($_GET['Mode']))
    {
        if($_GET['Mode']=='edit')
        {
            $SID=$_GET['Studentid'];
            $SelectStr="select * from studentdata where StudentID=$SID";
            $Result=mysqli_query($Conn,$SelectStr);

            if(mysqli_num_rows($Result)>0)
            {
                $ObjData=mysqli_fetch_assoc($Result);
```

```
                        }
                        else
                        {
                                echo "Data not found";
                        }
                }
        }
?>
<html>
        <head>
                <title>Student Registration</title>
        </head>
        <body>
                <form name="StudentDetail" action="" method="POST">
                        <div align="center">
                        <a href="StudentList.php">Show List of Students</a>
                        <fieldset style="width:50%;">
                        <legend>Student Registration</legend>
                        <table>
                                <tr>
                                        <td><label>StudentName</label></td>
                                        <td><input type="text" name="StudentName" id="StudentName"
                                        value="<?php if(isset($ObjData)) echo
                                        $ObjData['StudentName'];?>"/>
                                        <input type="hidden" name="StuID" id="StuID" value="<?php
                                        if(isset($ObjData)) echo $ObjData['StudentID'];?>"/></td>
                                </tr>
                                <tr>
                                        <td><label>StudentEmail</label></td>
                                        <td><input type="text" name="StudentEmail" id="StudentEmail"
                                        value="<?php if(isset($ObjData)) echo
                                        $ObjData['StudentEmail'];?>"/></td>
                                </tr>
                                <tr>
                                        <td><label>StudentContact</label></td>
                                        <td><input type="text" name="StudentContact"
                                        id="StudentContact" value="<?php if(isset($ObjData)) echo
                                        $ObjData['StudentContact'];?>"/></td>
                                </tr>
```

```
<tr>
        <td><label>Gender</label></td>
        <td><select name="StudentGender">
                <option value="" selected>Select one...</option>
                <option value="Male" <?php if((isset($ObjData)) and
                ($ObjData['StudentGender']=='Male')) echo "selected";
                ?>>Male</option>
                <option value="Female" <?php if((isset($ObjData)) and
                ($ObjData['StudentGender']=='Female')) echo "selected";
                ?>>Female</option>
                </select>
                </td>
        </tr>
        <tr>
                <td></td>
                <td><?php /*<input type="submit" name="submit" value="<?php
                if(isset($ObjData)) echo "Update"; else echo "Submit";?>"
                id="submit"/>*/ ?>
                <input type="submit" name="submit" value="Submit"
                id="submit"/>
                </td>
        </tr>
    </table>
    </fieldset>
    </div>
  </form>
</body>
</html>
```

Output:

StudentAddEdit.php



StudentList.php

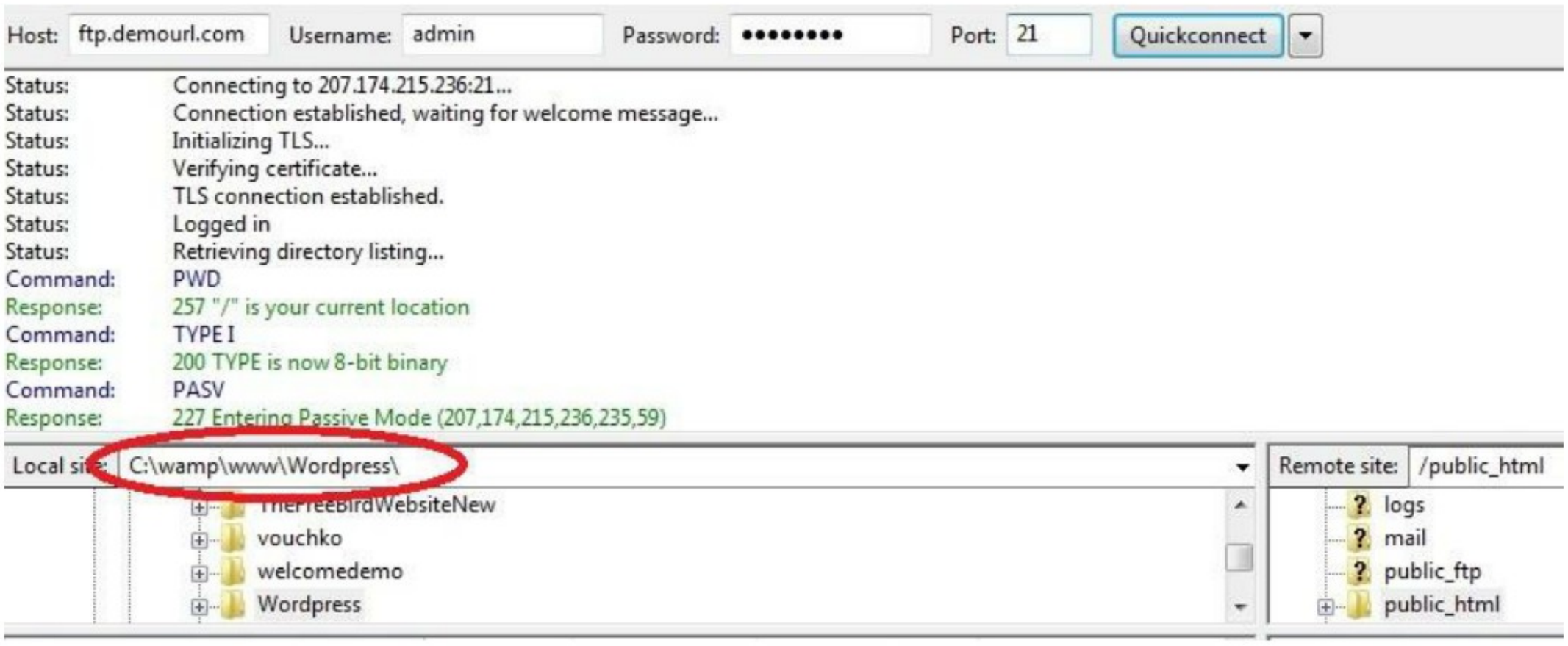
Add New Student

StudentName	StudentEmail	StudentContact	StudentGender	Action
Priya	priya@gmail.com	8888889999	Female	Update | Delete
Sumit	sumitpatel@gmail.com	9977775555	Male	Update | Delete
Kishan	kishan@yahoo.com	9822227777	Male	Update | Delete

Hosting Website Using Filezilla Software

- **FTP** is stands for File Transfer Protocol. It is a system that allows you to log in to a server and upload, download or modify content.
- FileZilla is powerful and free software for transferring files over the Internet.
- Once you have the FileZilla client downloaded and activated on your computer, enter the domain name in the address field (you can also use the server's IP address).
- The username and the password you need to type in are the same as the ones you use to log in to your cPanel.

- Click Quickconnect and the file listing will appear



- Then, select the files to be uploaded and drag-and-drop them under the www folder. Wait for the transfer to be completed.