SVKM's NMIMS University

Mukesh Patel School of Technology Management & Engineering

PROGRAM: BTech IT/MBATECH **SEMESTER IV**

COURSE: Web Programming **Practical Experiment: 8**

Part A (To be referred by students)

Roll No.: K041	**Name: Anish Sudhan Nair**
Prog/Yr/Sem:B.Tech CSE Cybersecurity	**Batch: K2**
Date of Experiment: 22/03/2022	**Date of Submission: 22/03/2022**

Topic covered: working with database

Learning Objective:
Learner would be able to

1. Create Database and tables.
2. Employ server side scripting using PHP with MySQL for Database Connectivity.

Prerequisites: -

- PHP BASICS

Outcomes:-

- Student will learn to created database and working with it
- Students will learn to develop PHP page to take data and insert into database.
- Students will learn to develop PHP page to take data and update into database.
- Students will learn to develop PHP page to choose the data and delete from database.
-

Theory:-

PHP Database connection

The collection of related data is called a database. XAMPP stands for cross-platform, Apache, MySQL, PHP, and Perl. It is among the simple light-weight local servers for website development.

Requirements: XAMPP web server procedure:

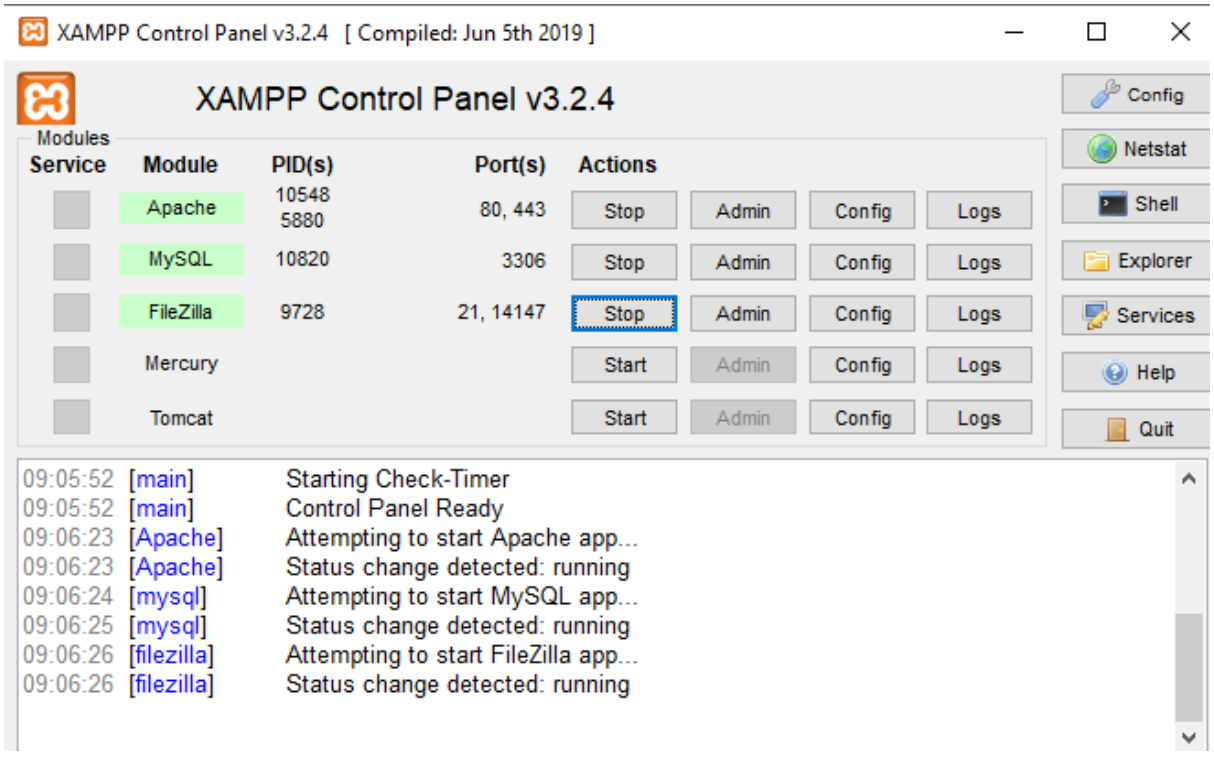
- Start XAMPP server by starting Apache and MySQL.
- Write PHP script for connecting to XAMPP.
- Run it in the local browser.
- Database is successfully created which is based on the PHP code.

In PHP, we can connect to the database using XAMPP web server by using the following path.

`"localhost/phpmyadmin"`

Steps in Detail:

- Open XAMPP and start running Apache, MySQL and FileZilla



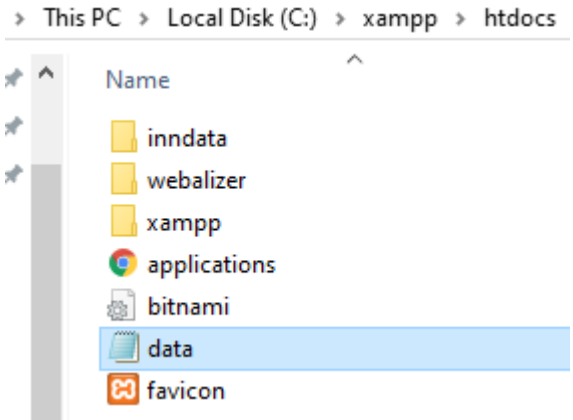
- Now open your PHP file and write your PHP code to create a database and a table in your database.
  PHP code to create a database:

```
- PHP
```

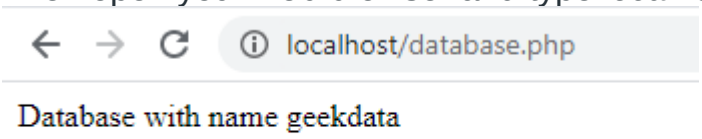
```php
<?php

// Server name must be localhost
$servername = "localhost";

// In my case, user name will be root
$username = "root";

// Password is empty
$password = "";

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

// Check connection
if ($conn->connect_error) {
    die("Connection failure: "
        . $conn->connect_error);
}
```

```
// Creating a database named geekdata

$sql = "CREATE DATABASE geekdata";

if ($conn->query($sql) === TRUE) {

    echo "Database with name geekdata";

} else {

    echo "Error: " . $conn->error;

}



// Closing connection

$conn->close();

?>
```

- Save the file as “data.php” in *htdocs* folder under XAMPP folder.

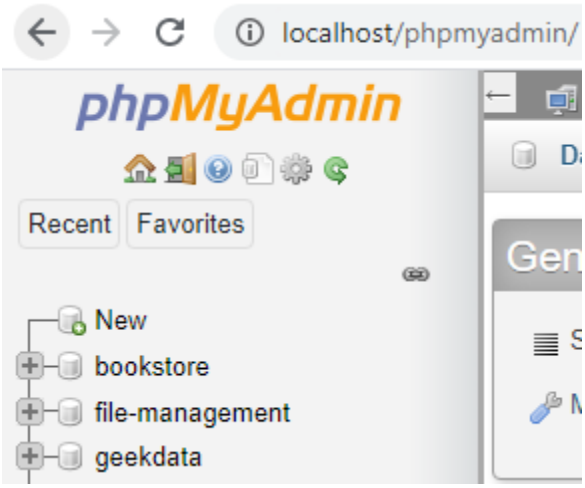


- Then open your web browser and type *localhost/data.php*



Finally, the database is created and connected to PHP.

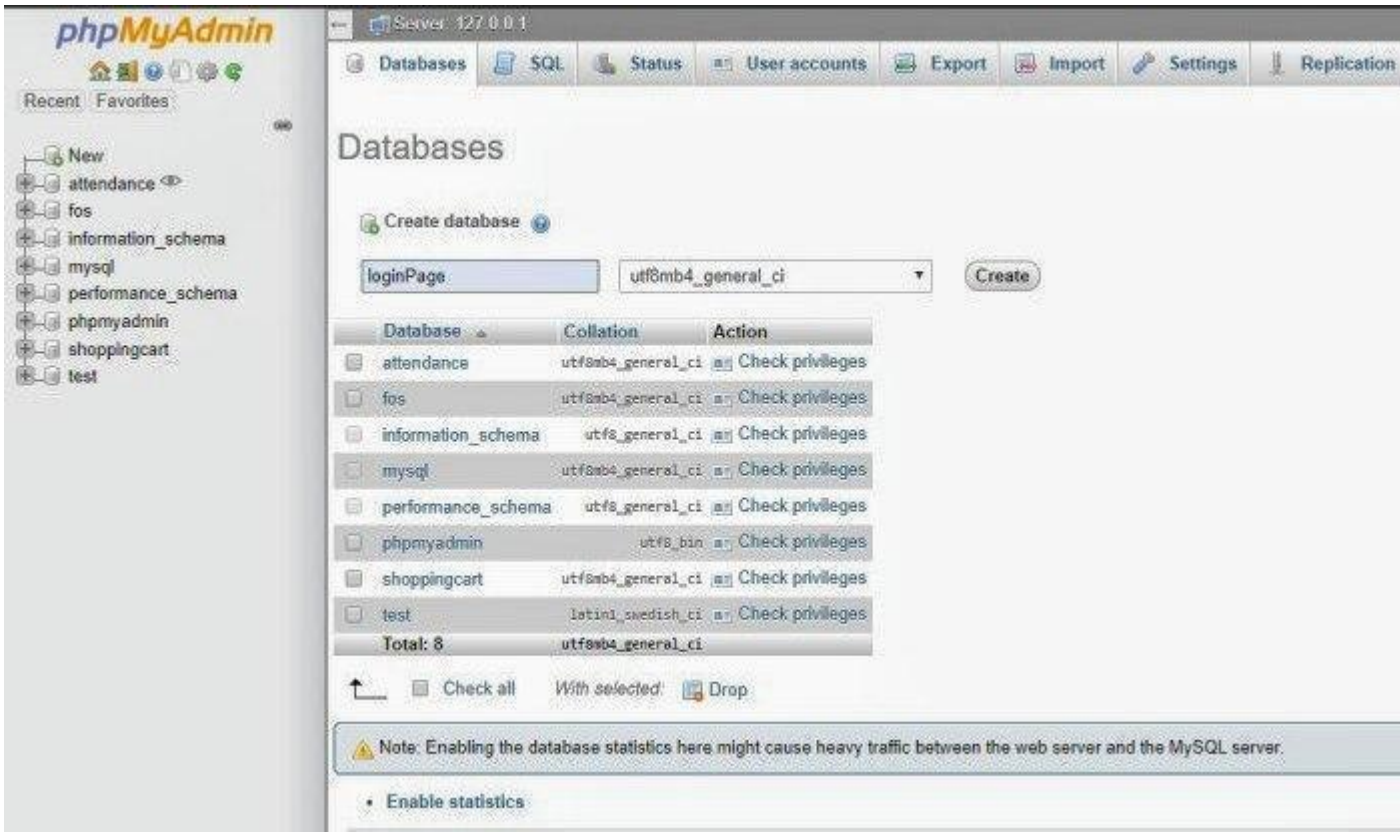
If you want to see your database, just type *localhost/phpmyadmin* in the web browser and the database can be found.



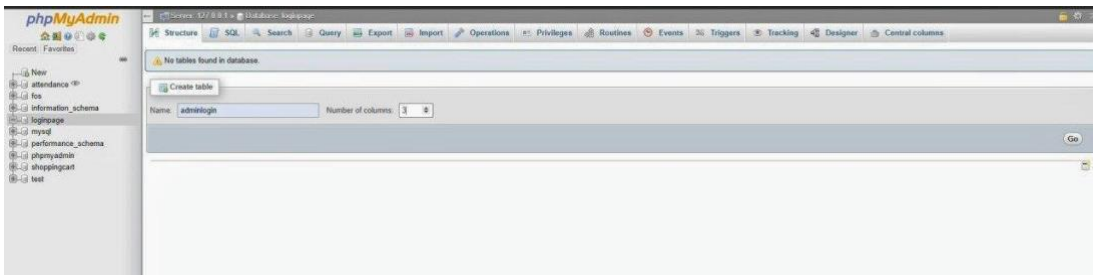
How to create an admin login page using PHP?

- Difficulty Level : Hard
- Last Updated : 31 Jul, 2021

Follow the steps to create an **admin login page using PHP:**
Here, we have created a login page of the admin, connected with the database, or whose information to log in to the page is already stored in our database.
1. Create Database: Create a database using XAMPP, the database is named “loginpage” here. You can give any name to your database.



2. Create Table: Create a table named “adminlogin”, inside “loginpage” database.



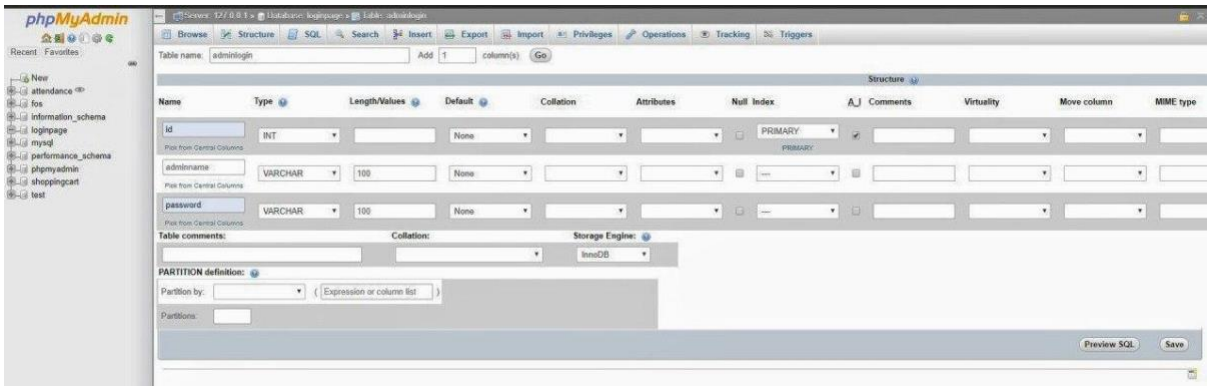
3. Create Table Structure: The table “adminlogin” should contain three fields.

- id – primary key – auto increment

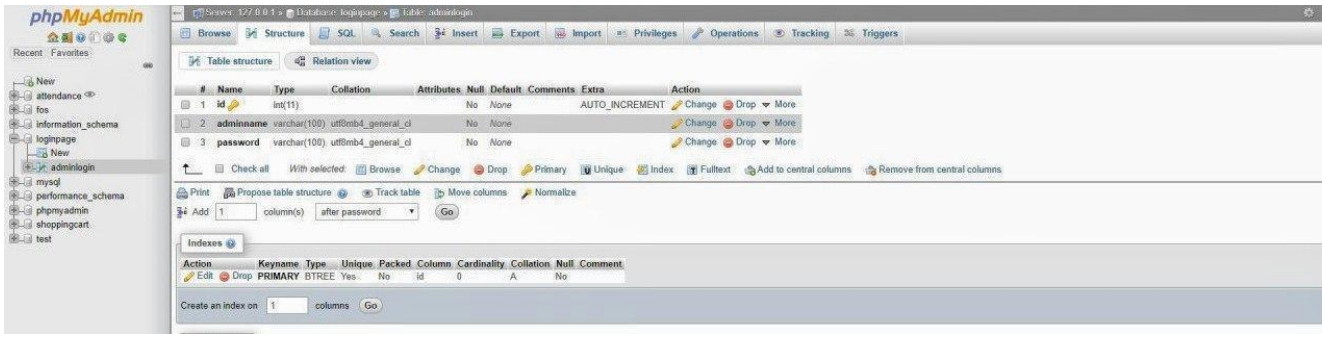
- adminname – varchar(100)
- password – varchar(100)

The datatype for adminname and password is **varchar**. The size can be altered as per the requirement. However, 100 is sufficient, and the datatype for “id” is **int** and it is a **primary key**.

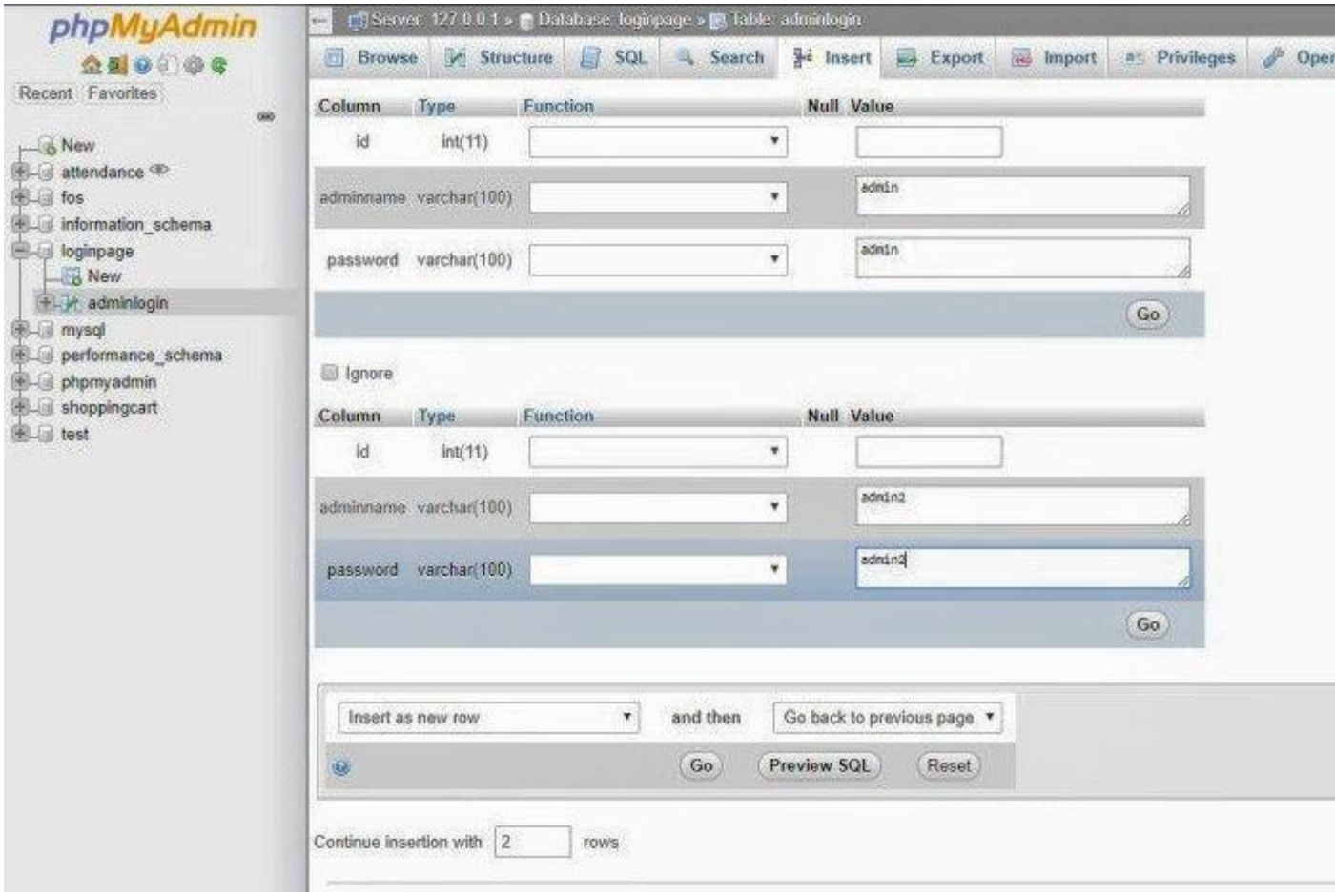
A **primary key** also called a primary keyword is a key in a relational database that is unique for each record. It is a unique identifier, such as a driver’s license number, telephone number (including area code), or vehicle identification number (VIN).



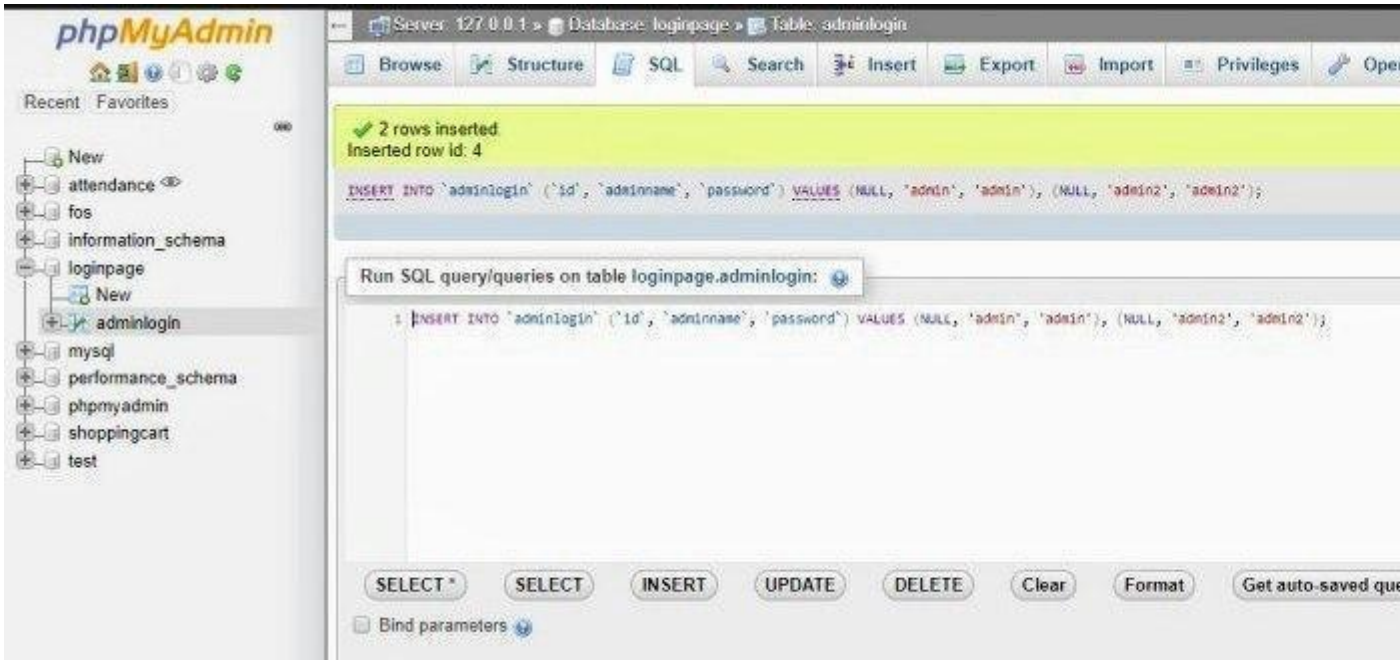
The structure of the table will look like this



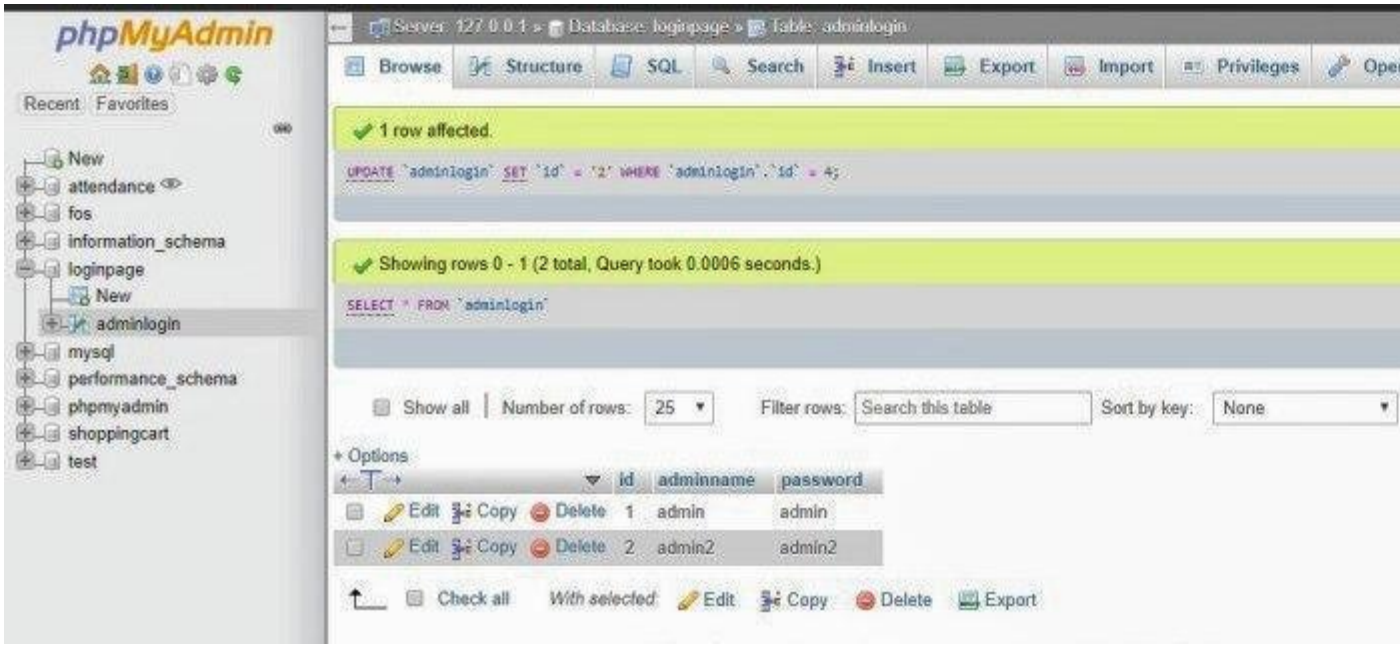
4. Insert admin login information: Here, the information of 2 admin are inserted. You can add as many as you want.



Or you can write an SQL query to insert the values.



After inserting the values, the table will look like this.



5. Create a folder that **includes** the **following files:** The folder should be in **"D:\xampp\htdocs\"** (or where your XAMPP is installed). On Linux **"/opt/lampp/htdocs"**.

- **Filename: index.php**
  SOURCE;
  https://www.geeksforgeeks.org/how-to-create-admin-login-page-using-php/?ref=lbp

- html

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <link rel="stylesheet" href=
"https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <link rel="stylesheet" href="login.css">
    <title>Login Page</title>
</head>

<body>
    <form action="validate.php" method="post">
        <div class="login-box">
            <h1>Login</h1>

            <div class="textbox">
```

```
                <i class="fa fa-user" aria-hidden="true"></i>
                <input type="text" placeholder="Adminname"
                        name="adminname" value="">
            </div>

            <div class="textbox">
                <i class="fa fa-lock" aria-hidden="true"></i>
                <input type="password" placeholder="Password"
                        name="password" value="">
            </div>

            <input class="button" type="submit"
                    name="login" value="Sign In">
        </div>
    </form>
</body>

</html>
```

- **Filename: connection.php**

- php

```
<?php

$conn = "";

try {
    $servername = "localhost:3306";
    $dbname = "loginPage";
    $username = "root";
    $password = "";

    $conn = new PDO(
        "mysql:host=$servername; dbname=loginPage",
        $username, $password
    );

   $conn->setAttribute(PDO::ATTR_ERRMODE,
                    PDO::ERRMODE_EXCEPTION);
}
catch(PDOException $e) {
    echo "Connection failed: " . $e->getMessage();
}

?>
```

- **Filename: login.css**

- css **Filename: validate.php**

- php

```
<?php

include_once('connection.php');

function test_input($data) {

    $data = trim($data);
    $data = stripslashes($data);
    $data = htmlspecialchars($data);
    return $data;
}

if ($_SERVER["REQUEST_METHOD"]== "POST") {

    $adminname = test_input($_POST["adminname"]);
    $password = test_input($_POST["password"]);
    $stmt = $conn->prepare("SELECT * FROM adminlogin");
    $stmt->execute();
    $users = $stmt->fetchAll();

    foreach($users as $user) {

        if(($user['adminname'] == $adminname) &&
            ($user['password'] == $password)) {
                header("Location: adminpage.php");
        }
        else {
            echo "<script language='javascript'>";
            echo "alert('WRONG INFORMATION')";
            echo "</script>";
            die();
        }
    }
}

?>
```

  - **Filename: adminpage.php** Add anything that you want to display to the admin page.

- html

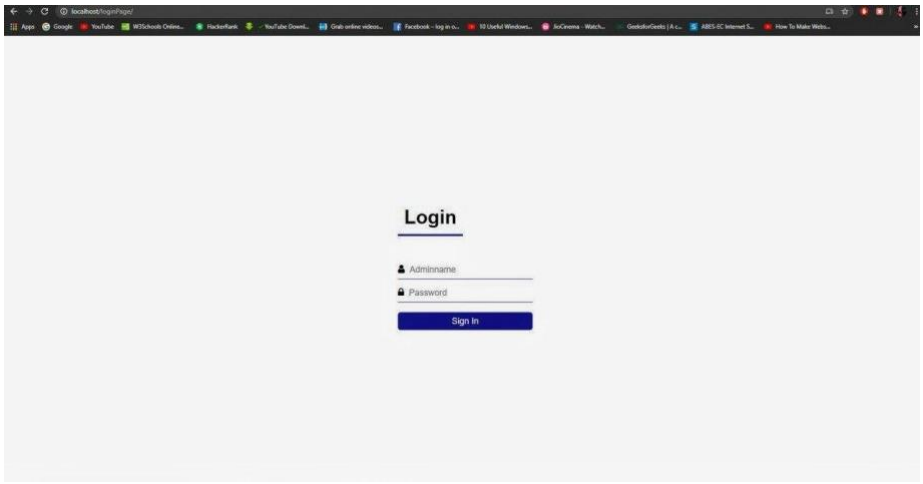
```
<h2>Hello Admin</h2>
```

6. After completing all the **above steps, now follow the steps:**

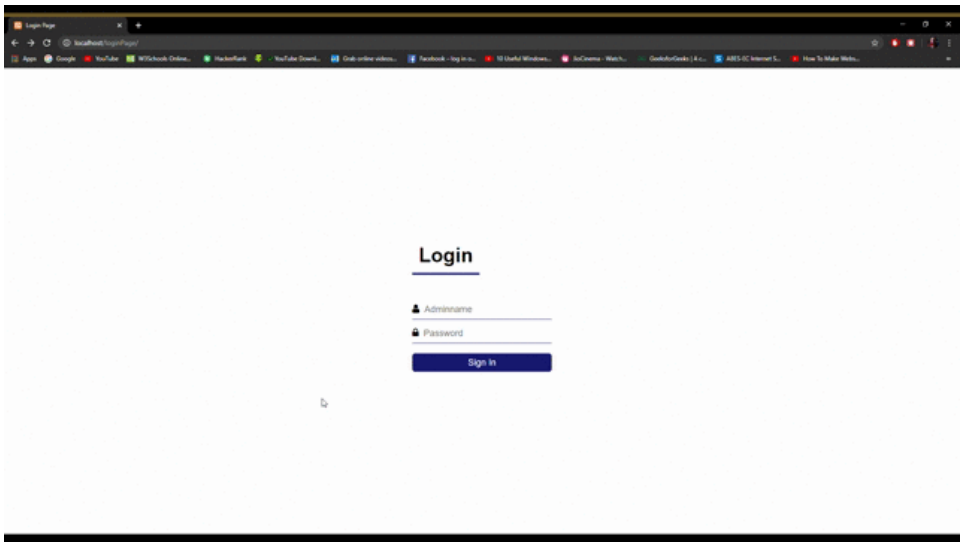
- Run XAMPP
- Start Apache and MySQL server

- Type *http://localhost/loginPage/* in your browser.

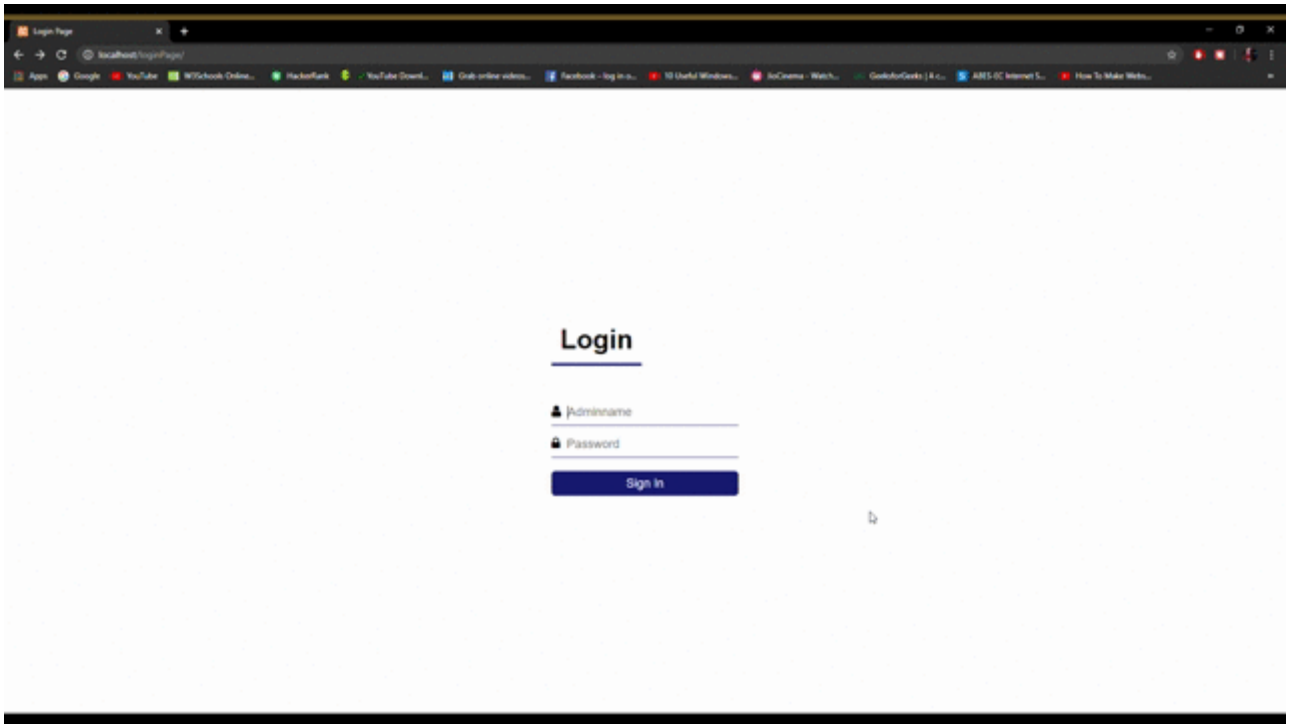
This login page will appear.



If you enter the correct credentials i.e. admin name and password, then you will be logged-in to the “admin.php” page.



else, you get an error pop-up alert.



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
- MySQL is named after co-founder Monty Widenius's daughter: My

The data in a MySQL database are stored in tables. A table is a collection of related data, 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)

Database Queries

A query is a question or a request.

We can query a database for specific information and have a recordset returned.

Look at the following query (using standard SQL):

```
SELECT LastName FROM Employees
```

The query above selects all the data in the "LastName" column from the "Employees" table.

To learn more about SQL, please visit our SQL tutorial.

Download MySQL Database

If you don't have a PHP server with a MySQL Database, you can download it for free here: http://www.mysql.com

---

Facts About MySQL Database

MySQL is the de-facto standard database system for web sites with HUGE volumes of both data and end-users (like Facebook, Twitter, and Wikipedia).

Another great thing about MySQL is that it can be scaled down to support embedded database applications.

Look at http://www.mysql.com/customers/ for an overview of companies using MySQL.

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";
```

```php
$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";
?>
```

Close the Connection

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

MySQLi Object-Oriented:

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

MySQLi Procedural:

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

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 and PDO

Example (MySQLi Procedural)

```php
<?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());
}

// Create database
$sql = "CREATE DATABASE myDB";
if (mysqli_query($conn, $sql)) {
  echo "Database created successfully";
} else {
  echo "Error creating database: " . mysqli_error($conn);
}

mysqli_close($conn);
?>
```

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
<?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
```

```php
$sql = "CREATE DATABASE myDB";
if ($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 MySQL Create Table

Create a MySQL Table Using MySQLi and PDO

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) NOT NULL,
email VARCHAR(50),
```

```
reg_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_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, go to our Data Types reference.

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 Procedural)

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

// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
  die("Connection failed: " . mysqli_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 DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP
)";

if (mysqli_query($conn, $sql)) {
  echo "Table MyGuests created successfully";
} else {
  echo "Error creating table: " . mysqli_error($conn);
}

mysqli_close($conn);
?>
```

PHP MySQL Insert Data

Insert Data Into MySQL Using MySQLi and PDO

After a database and a table have been created, we can start adding data in them.

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,...)
```

To learn more about SQL, please visit our SQL tutorial.

In the previous chapter we created an empty table named "MyGuests" with five columns: "id", "firstname", "lastname", "email" and "reg_date". Now, let us fill the table with data.

Note: If a column is AUTO_INCREMENT (like the "id" column) or TIMESTAMP with default update of current_timesamp (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 Procedural)

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

// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
  die("Connection failed: " . mysqli_connect_error());
}

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

if (mysqli_query($conn, $sql)) {
  echo "New record created successfully";
} else {
  echo "Error: " . $sql . "<br>" . mysqli_error($conn);
}

mysqli_close($conn);
?>
```

Select Data From a MySQL Database

The SELECT statement is used to select data from one or more tables:

```
SELECT column_name(s) FROM table_name
```

or we can use the * character to select ALL columns from a table:

```
SELECT * FROM table_name
```

To learn more about SQL, please visit our SQL tutorial.

Example (MySQLi Procedural)

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

// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
  die("Connection failed: " . mysqli_connect_error());
}

$sql = "SELECT id, firstname, lastname FROM MyGuests";
$result = mysqli_query($conn, $sql);

if (mysqli_num_rows($result) > 0) {
  // output data of each row
  while($row = mysqli_fetch_assoc($result)) {
    echo "id: " . $row["id"]. " - Name: " . $row["firstname"]. " " .
$row["lastname"]. "<br>";
  }
} else {
  echo "0 results";
}

mysqli_close($conn);
?>
```

PHP MySQL Use The WHERE Clause

❮ PreviousNext ❯

Select and Filter Data From a MySQL Database

The WHERE clause is used to filter records.

The WHERE clause is used to extract only those records that fulfill a specified condition.

```
SELECT column_name(s) FROM table_name WHERE column_name operator value
```

To learn more about SQL, please visit our SQL tutorial.

Example (MySQLi Procedural)

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

// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
  die("Connection failed: " . mysqli_connect_error());
}

$sql = "SELECT id, firstname, last name FROM MyGuests WHERE
lastname='Doe'";
$result = mysqli_query($conn, $sql);

if (mysqli_num_rows($result) > 0) {
  // output data of each row
  while($row = mysqli_fetch_assoc($result)) {
    echo "id: " . $row["id"]. " - Name: " . $row["firstname"]. " " .
$row["lastname"]. "<br>";
  }
} else {
  echo "0 results";
}

mysqli_close($conn);
?>
```

PHP MySQL Use The ORDER BY Clause

Select and Order Data From a MySQL Database

The ORDER BY clause is used to sort the result-set in ascending or descending order.

The ORDER BY clause sorts the records in ascending order by default. To sort the records in descending order, use the DESC keyword.

```
SELECT column_name(s) FROM table_name ORDER BY column_name(s) ASC|DESC
```

To learn more about SQL, please visit our SQL tutorial.

Select and Order Data With MySQLi

Example (MySQLi Procedural)

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

// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
  die("Connection failed: " . mysqli_connect_error());
}

$sql = "SELECT id, firstname, lastname FROM MyGuests ORDER BY
lastname";
$result = mysqli_query($conn, $sql);

if (mysqli_num_rows($result) > 0) {
  // output data of each row
  while($row = mysqli_fetch_assoc($result)) {
    echo "id: " . $row["id"]. " - Name: " . $row["firstname"]. " " .
```

```
$row["lastname"]. "<br>";
  }
} else {
  echo "0 results";
}

mysqli_close($conn);
?>
```

Delete Data From a MySQL Table Using MySQLi and PDO

The DELETE statement is used to delete records from a table:

```
DELETE FROM table_name
WHERE some_column = some_value
```

Notice the WHERE clause in the DELETE syntax: The WHERE clause specifies which record or records that should be deleted. If you omit the WHERE clause, all records will be deleted!

To learn more about SQL, please visit our SQL tutorial.

Let's look at the "MyGuests" table:

id	firstname	lastname	email	reg_date
1	John	Doe	john@example.com	2014-10-22 14:26:15
2	Mary	Moe	mary@example.com	2014-10-23 10:22:30
3	Julie	Dooley	julie@example.com	2014-10-26 10:48:23

Example (MySQLi Procedural)

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

// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
  die("Connection failed: " . mysqli_connect_error());
}

// sql to delete a record
$sql = "DELETE FROM MyGuests WHERE id=3";

if (mysqli_query($conn, $sql)) {
  echo "Record deleted successfully";
} else {
  echo "Error deleting record: " . mysqli_error($conn);
}

mysqli_close($conn);
?>
```

Update Data In a MySQL Table Using MySQLi and PDO

The UPDATE statement is used to update existing records in a table:

```
UPDATE table_name
SET column1=value, column2=value2,...
WHERE some_column=some_value
```

Example (MySQLi Procedural)

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

// Create connection
```

```php
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
  die("Connection failed: " . mysqli_connect_error());
}

$sql = "UPDATE MyGuests SET lastname='Doe' WHERE id=2";

if (mysqli_query($conn, $sql)) {
  echo "Record updated successfully";
} else {
  echo "Error updating record: " . mysqli_error($conn);
}

mysqli_close($conn);
?>
```

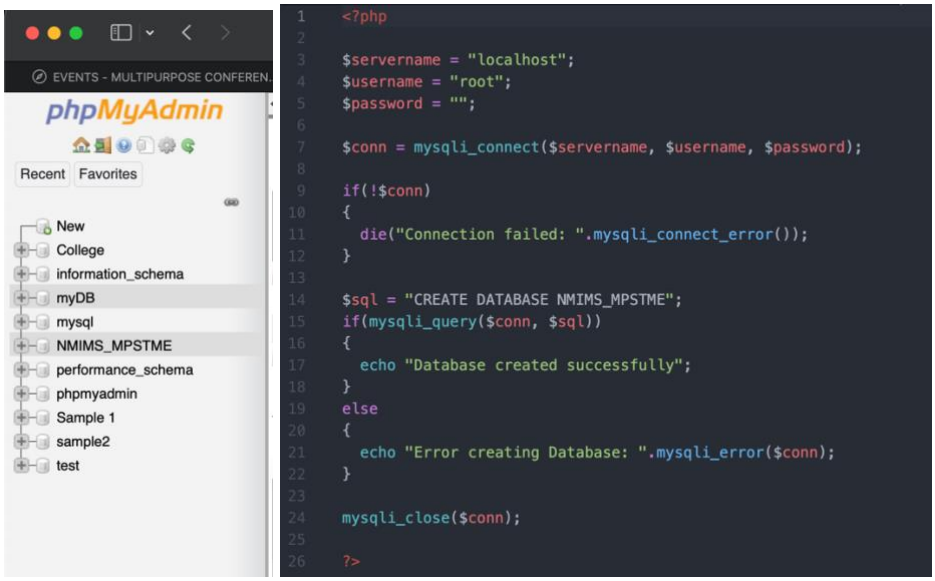
Part B (to be completed by students)

(Students must submit the soft copy as per the following segments. A soft copy containing Part A and Part B must be uploaded on the platform specified by the Practical Teacher. The filename should be **RollNo_Name_Exp1**)

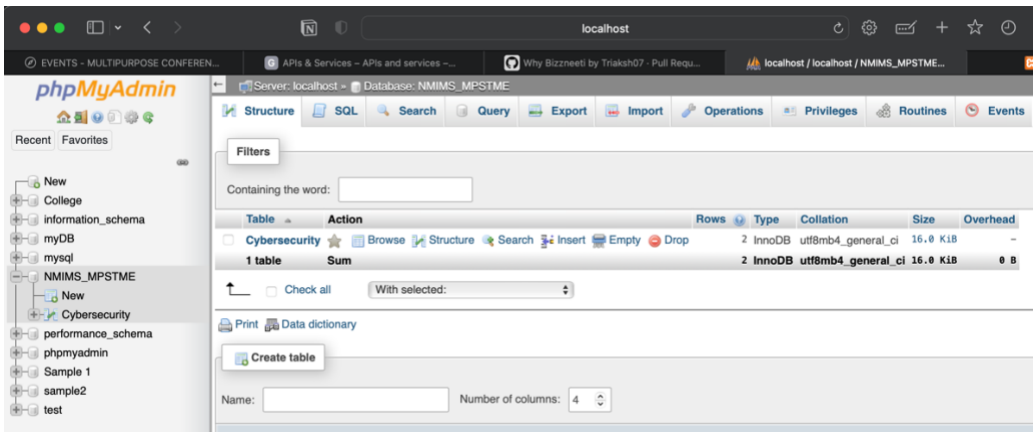
Roll No.: K041	**Name: Anish Sudhan Nair**
Prog/Yr/Sem:B.Tech CSE Cybersecurity	**Batch: K2**
Date of Experiment: 22/03/2022	**Date of Submission: 22/03/2022**

Q.1 . Create a database for educational institutes with table names as class names like Extc,IT, Computers. Also Create Table with ROLL NO , NAME , MOBILE NUMBER, MARKS1 ,MARKS2,MARKS3 AND PERCENTAGE column.

Database



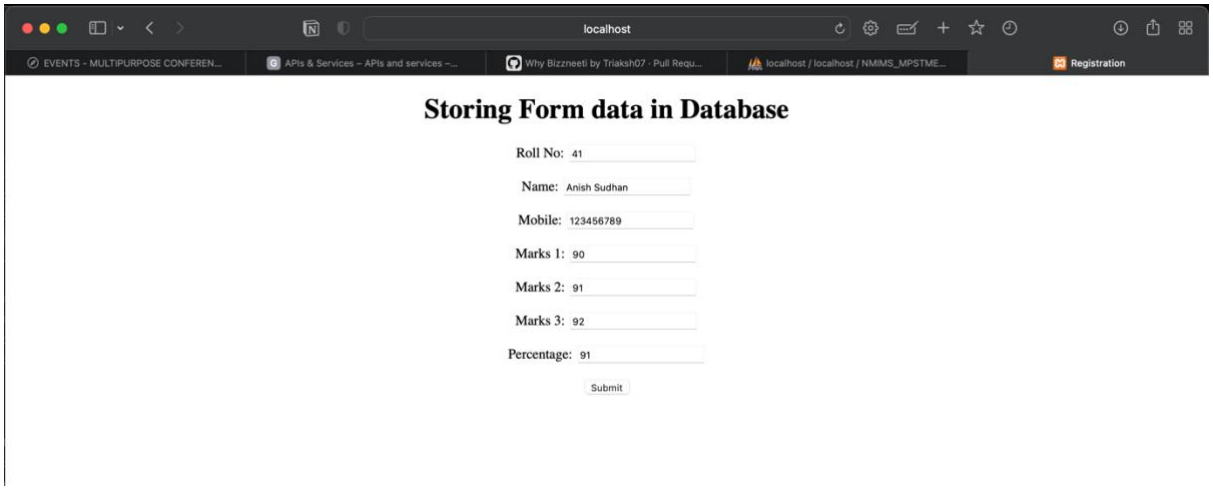
Table



```php
<?php

$servername = "localhost";
$username = "root";
$password = "";
$dbname = "NMIMS_MPSTME";

$conn = mysqli_connect($servername, $username, $password, $dbname);

if(!$conn)
{
  die("Connection failed: ".mysqli_connect_error());
}

$sql2 = "CREATE TABLE Cybersecurity (ROLL_NO INT(10) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
                  NAME VARCHAR(255),
                  MOBILE_NUMBER VARCHAR(255),
                  MARKS1 VARCHAR(255),
                  MARKS2 VARCHAR(255),
                  MARKS3 VARCHAR(255),
                  PERCENTAGE VARCHAR(255)
                  )";

if(mysqli_query($conn, $sql2))
{
  echo "Table created successfully";
}
else
{
  echo "Error creating Table: ".mysqli_error($conn);
}


mysqli_close($conn);

?>
```

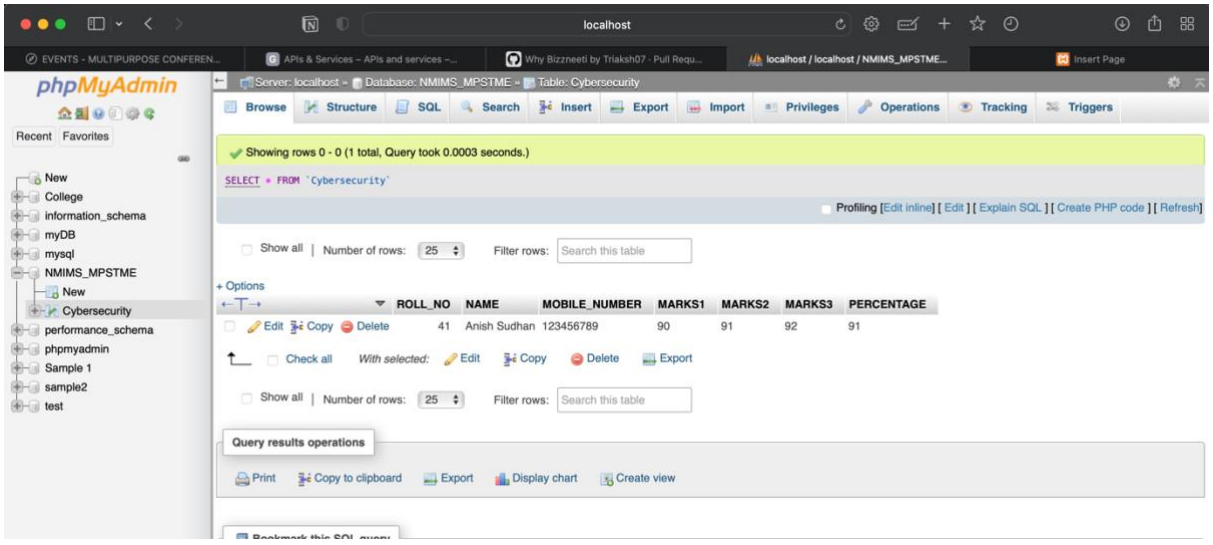
Q.2. Design PHP form to take data input from user and insert into above created table.

Form Input



```html
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Registration</title>
</head>

<body>
    <center>
        <h1>Storing Form data in Database</h1>

        <form action="insert.php" method="post">
<p>
<label for="roll">Roll No:</label>
<input type="text" name="ROLL_NO" id="roll">
</p>

<p>
<label for="name">Name:</label>
<input type="text" name="NAME" id="name">
</p>

<p>
<label for="mobile">Mobile:</label>
<input type="text" name="MOBILE_NUMBER" id="mobile">
</p>

<p>
<label for="marks1">Marks 1:</label>
<input type="text" name="MARKS1" id="marks1">
</p>

<p>
<label for="marks2">Marks 2:</label>
<input type="text" name="MARKS2" id="marks2">
</p>

<p>
<label for="marks3">Marks 3:</label>
<input type="text" name="MARKS3" id="marks3">
</p>

<p>
<label for="percentage">Percentage:</label>
<input type="text" name="PERCENTAGE" id="percentage">
</p>

<input type="submit" value="Submit">
</form>
    </center>
</body>
</html>
```

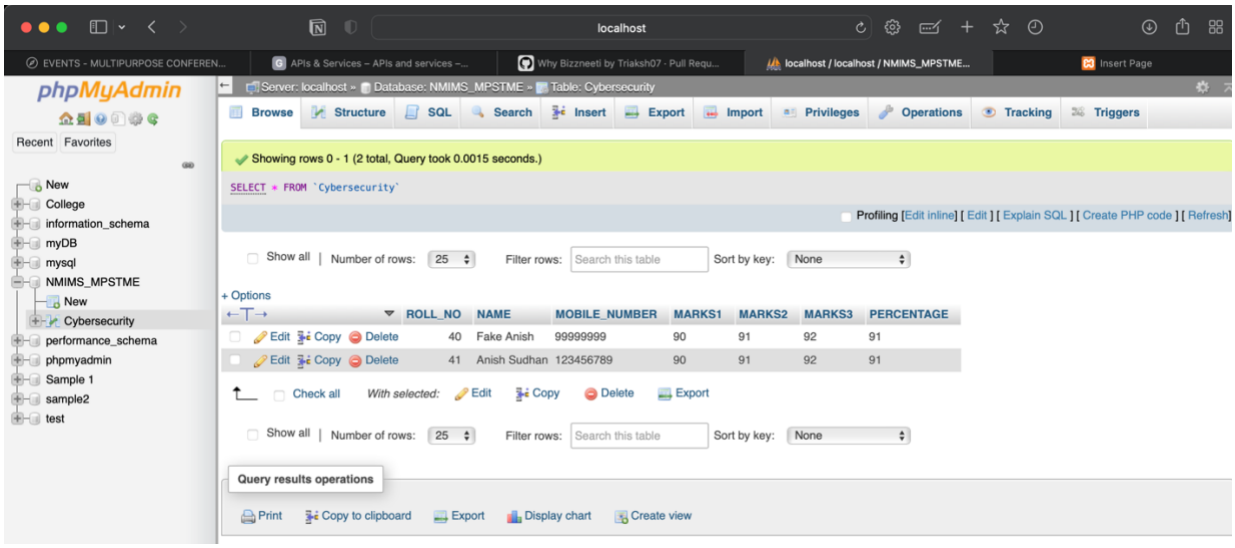
Data inserted into table



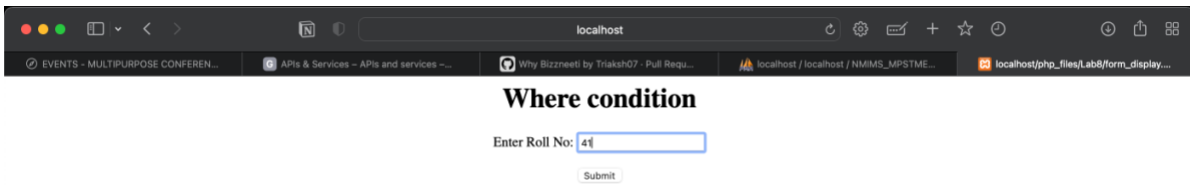
```php
<!DOCTYPE html>
<html>

<head>
    <title>Insert Page </title>
</head>

<body>
    <center>
        <?php

        $conn = mysqli_connect("localhost", "root", "", "NMIMS_MPSTME");

        if($conn === false){
            die("ERROR: Could not connect. ". mysqli_connect_error());
        }

        $ROLL_NO = $_REQUEST['ROLL_NO'];
        $NAME =  $_REQUEST['NAME'];
        $MOBILE_NUMBER = $_REQUEST['MOBILE_NUMBER'];
        $MARKS1 =  $_REQUEST['MARKS1'];
        $MARKS2 =  $_REQUEST['MARKS2'];
        $MARKS3 =  $_REQUEST['MARKS3'];
        $PERCENTAGE = $_REQUEST['PERCENTAGE'];

        $sql = "INSERT INTO Cybersecurity
        VALUES('$ROLL_NO','$NAME','$MOBILE_NUMBER','$MARKS1','$MARKS2','$MARKS3','$PERCENTAGE')";

        if(mysqli_query($conn, $sql)){
            echo "data stored in a database successfully.";

        } else{
            echo "ERROR". mysqli_error($conn);
        }

        mysqli_close($conn);
        ?>
    </center>
</body>

</html>
```

Q.3 Design PHP form to display the data based on conditions/options selected in the PHP form.

Database Data

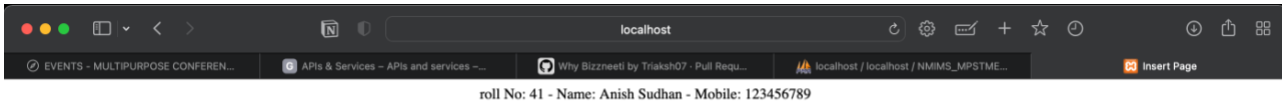


Form to input desired Roll No



```
<body>
    <center>
        <h1>Where condition</h1>

        <form action="form_condition.php" method="post">

<p>
<label for="condition">Enter Roll No:</label>
<input type="text" name="condition" id="condition">
</p>

<input type="submit" value="Submit">
</form>
    </center>
</body>
</html>
```

Fetching the particular data from database

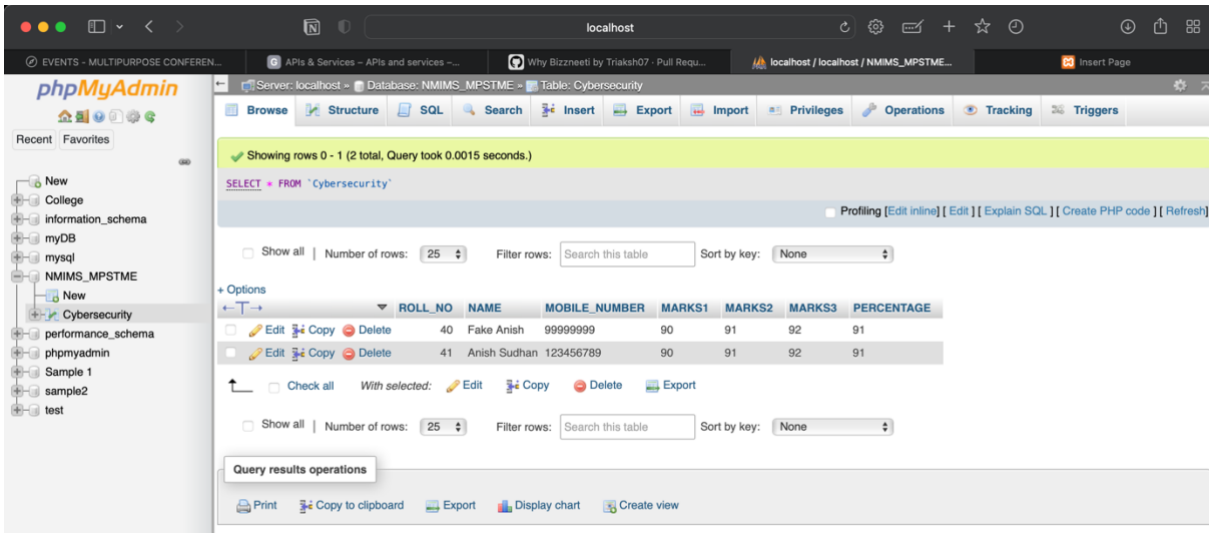


roll No: 41 - Name: Anish Sudhan - Mobile: 123456789

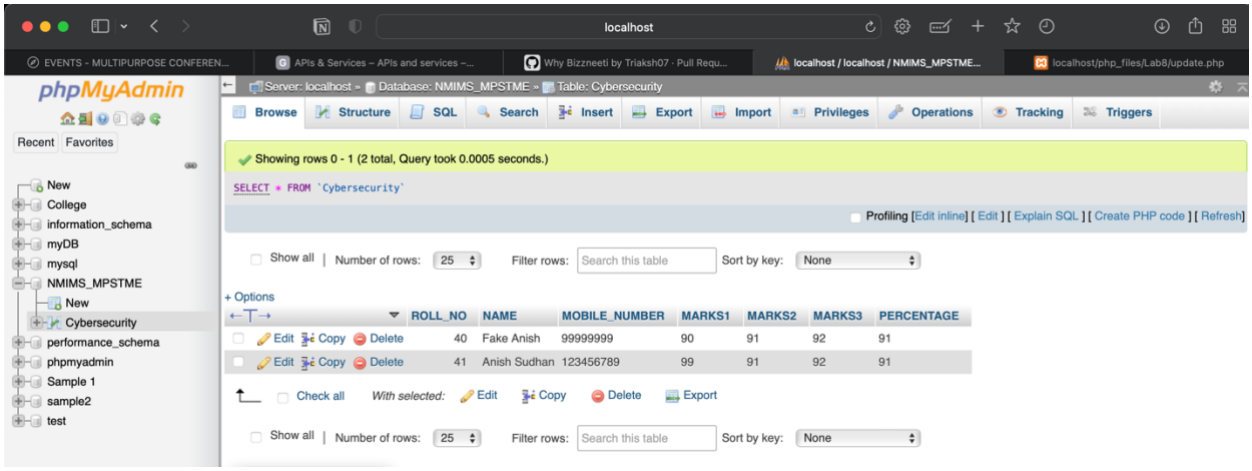
```
<!DOCTYPE html>
<html>

<head>
    <title>Insert Page </title>
</head>

<body>
    <center>
        <?php

        $conn = mysqli_connect("localhost", "root", "", "NMIMS_MPSTME");

        if($conn === false){
            die("ERROR: Could not connect. ". mysqli_connect_error());
        }

        $conditon = $_REQUEST['condition'];

        $sql = "SELECT ROLL_NO, NAME, MOBILE_NUMBER FROM Cybersecurity WHERE ROLL_NO=$conditon";
        $result = mysqli_query($conn, $sql);

        if (mysqli_num_rows($result) > 0) {
          while($row = mysqli_fetch_assoc($result)) {
            echo "roll No: " . $row["ROLL_NO"]. " - Name: " . $row["NAME"]. " - Mobile: " .
            $row["MOBILE_NUMBER"]. "<br>";
          }
        } else {
          echo "0 results";
        }

        mysqli_close($conn);
        ?>
    </center>
</body>

</html>
```

Q.4 Write PHP code to update and delete the data of a given table.

Database Data

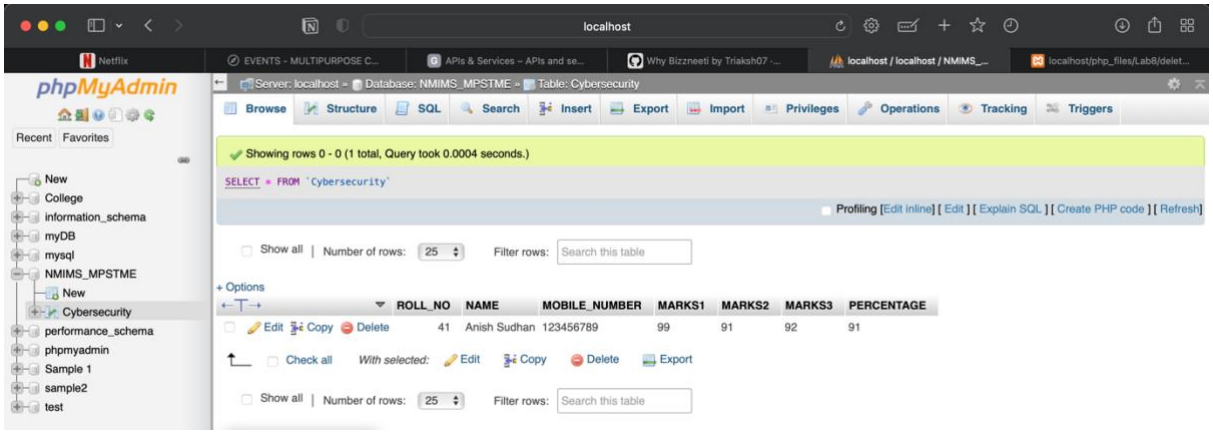


Updating




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


$conn = mysqli_connect($servername, $username, $password, $dbname);

if (!$conn) {
  die("Connection failed: " . mysqli_connect_error());
}

$sql = "UPDATE Cybersecurity SET MARKS1='99' WHERE ROLL_NO=41";

if (mysqli_query($conn, $sql)) {
  echo "Record updated successfully";
} else {
  echo "Error updating record: " . mysqli_error($conn);
}

mysqli_close($conn);
?>
```

Deleting




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

$conn = mysqli_connect($servername, $username, $password, $dbname);

if (!$conn) {
  die("Connection failed: " . mysqli_connect_error());
}

$sql = "DELETE FROM Cybersecurity WHERE ROLL_NO=40";

if (mysqli_query($conn, $sql)) {
  echo "Record deleted successfully";
} else {
  echo "Error deleting record: " . mysqli_error($conn);
}

mysqli_close($conn);
?>
```