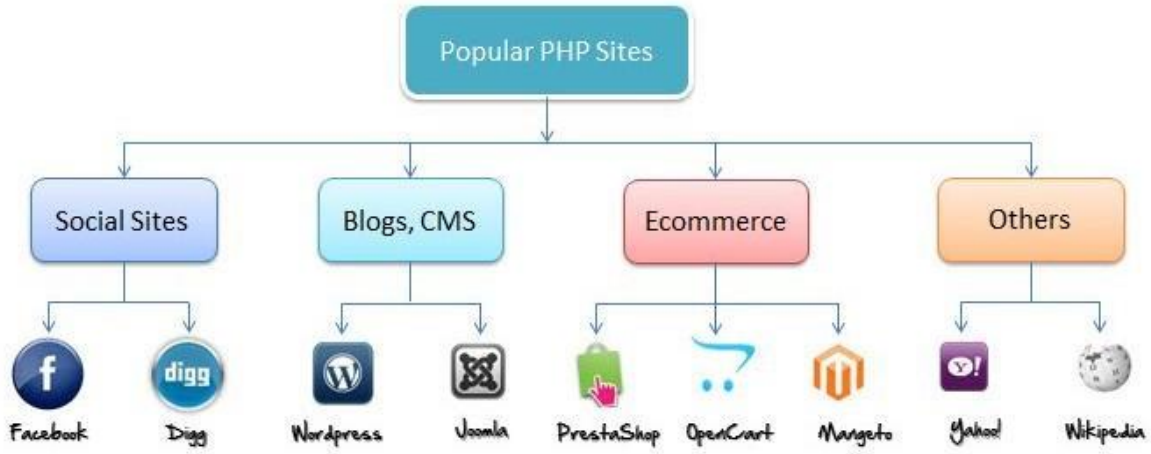
Lab Session #10

(*Connectivity: PHP with MySQL*)

PHP:

PHP is the most popular scripting language (i-e A scripting language is a language that interprets scripts at runtime) for web development. It is free, open source and server-side (the code is executed on the server).



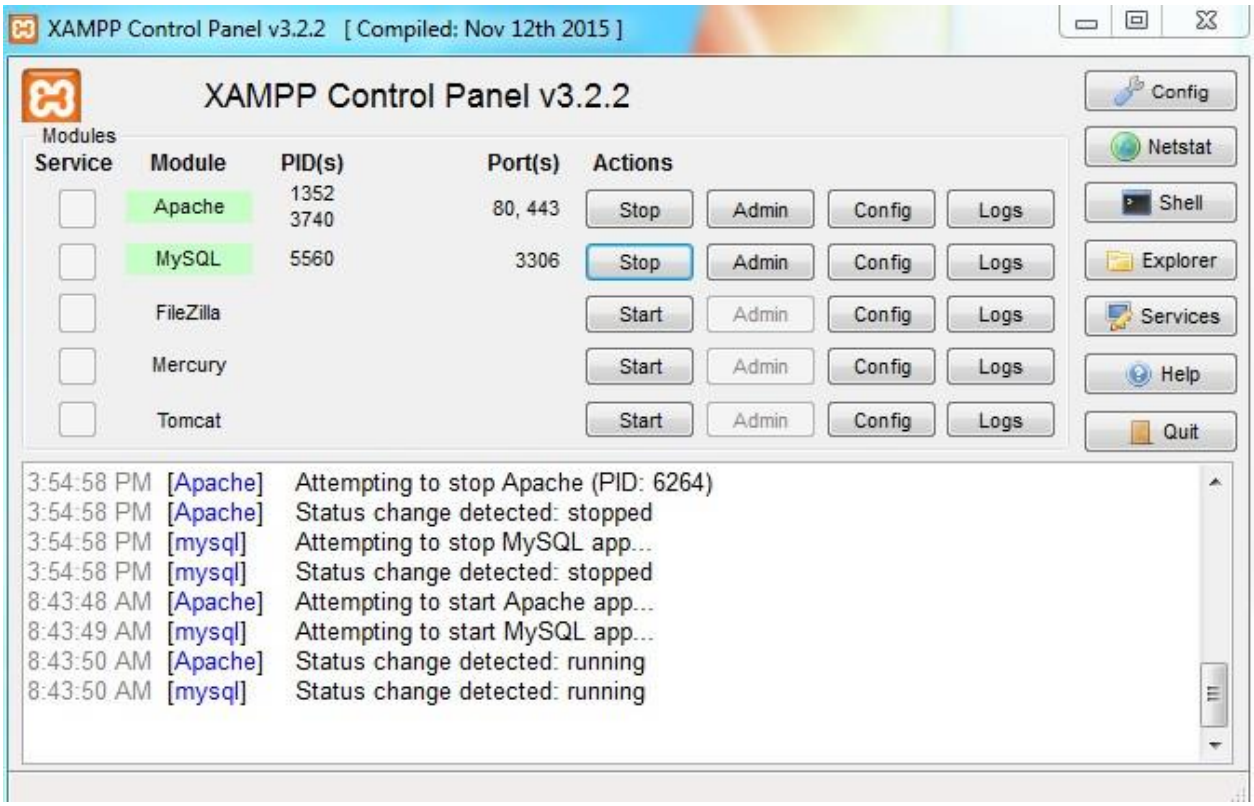
MySQL:

MySQL is a Relational Database Management System (RDBMS) that uses Structured Query Language (SQL). It is also free and open source. The combination of PHP and MySQL gives unmet options to create just about any kind of website - from small contact form to large corporate portal.

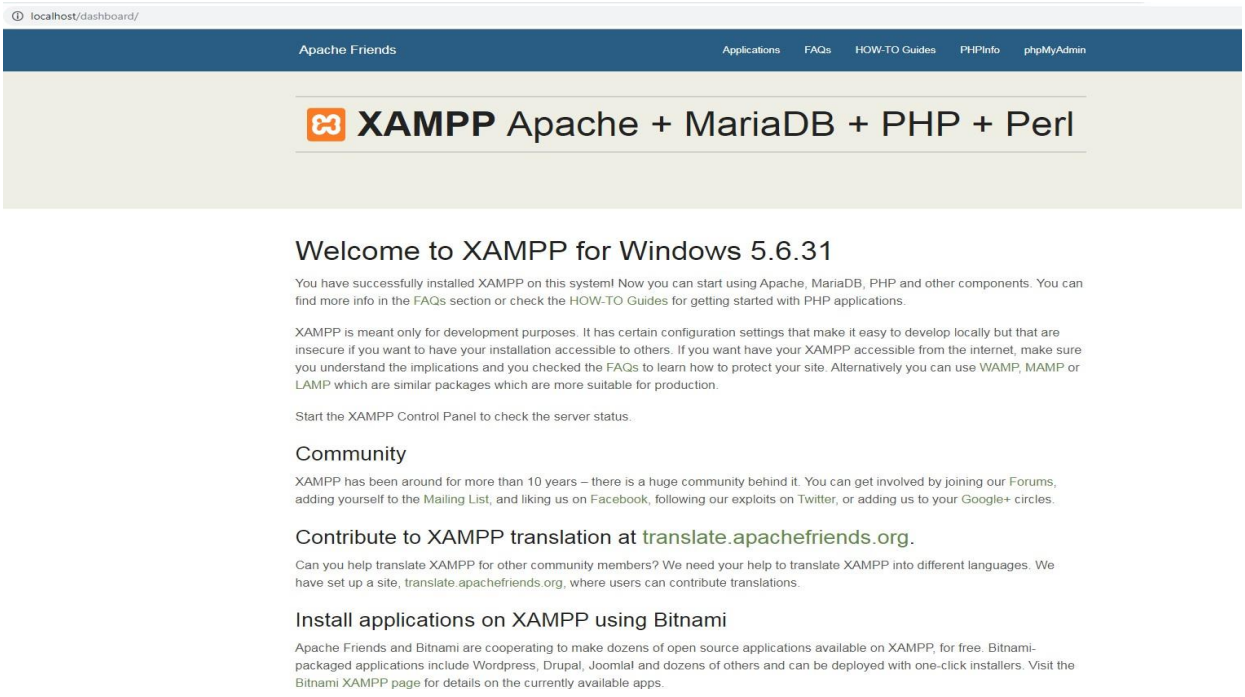
What is required?

1. Text Editor (i.e., Notepad++, Brackets, Dreamweaver etc.)
2. Wamp or Xampp Server
3. Web browser (Firefox, Chrome etc.)

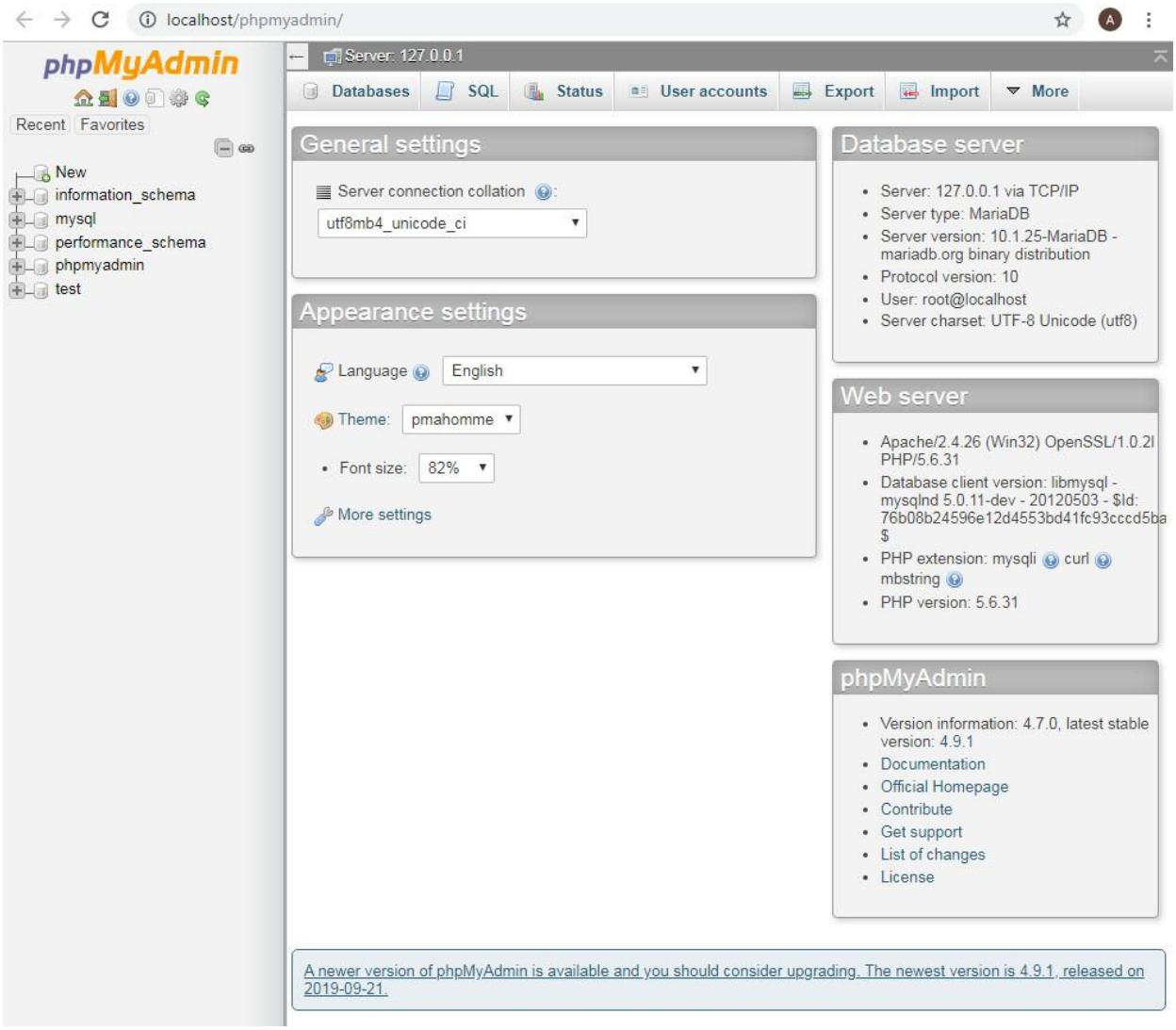
Xampp_Control Panel:



Apache _Admin Console:



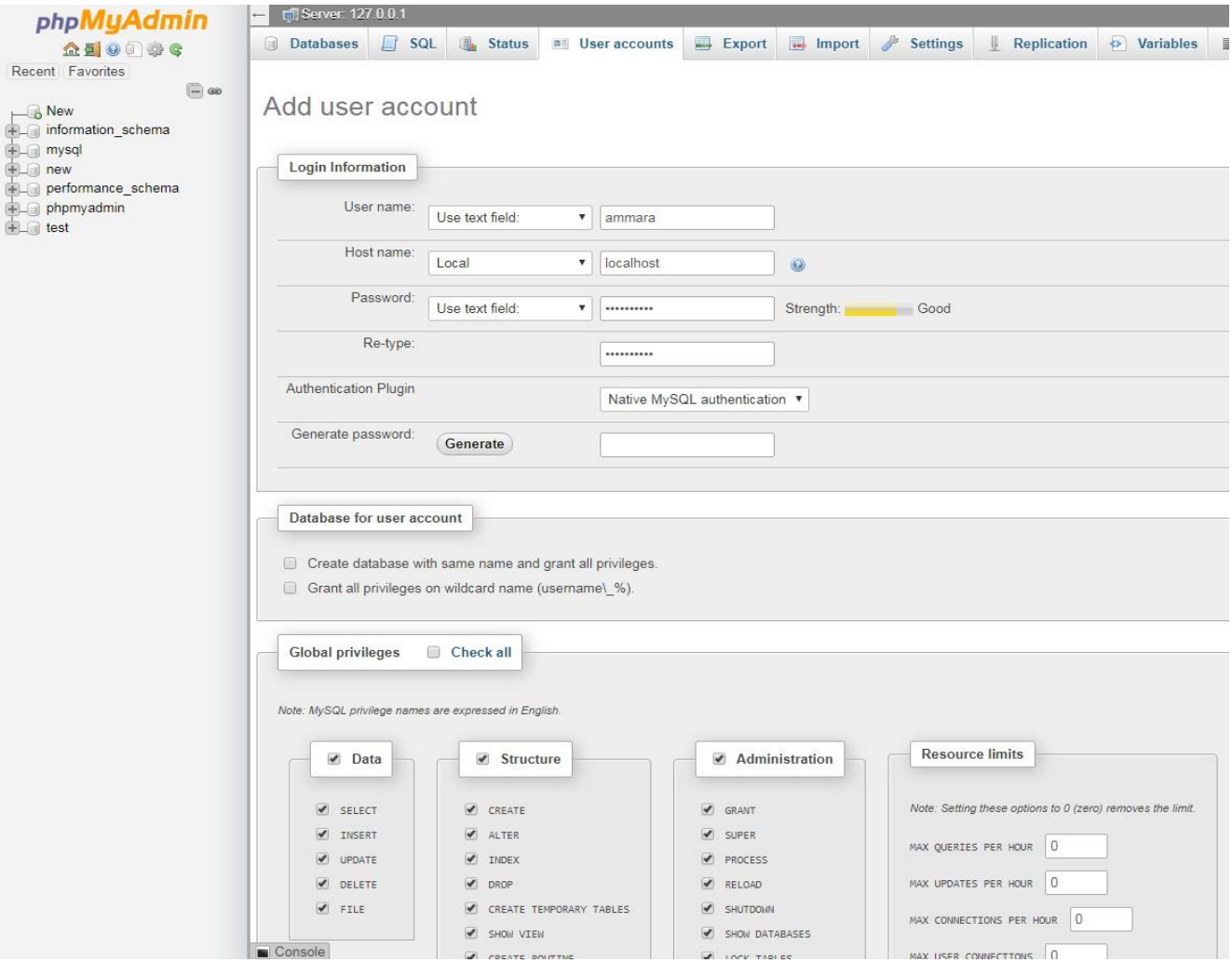
MySQL-phpMyAdmin Panel:



Creating Users:

Follow these Steps:

1. Go to User Accounts in phpMyAdmin.
2. Click Add user account.
3. Set username, hostname, password and privileges for this user.
4. Click Go.



PHP Connection to MySQL:

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

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

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

Save this file in C:\xampp\htdocs folder.

To run the script type in the browser localhost/filename.php

Creating Database:

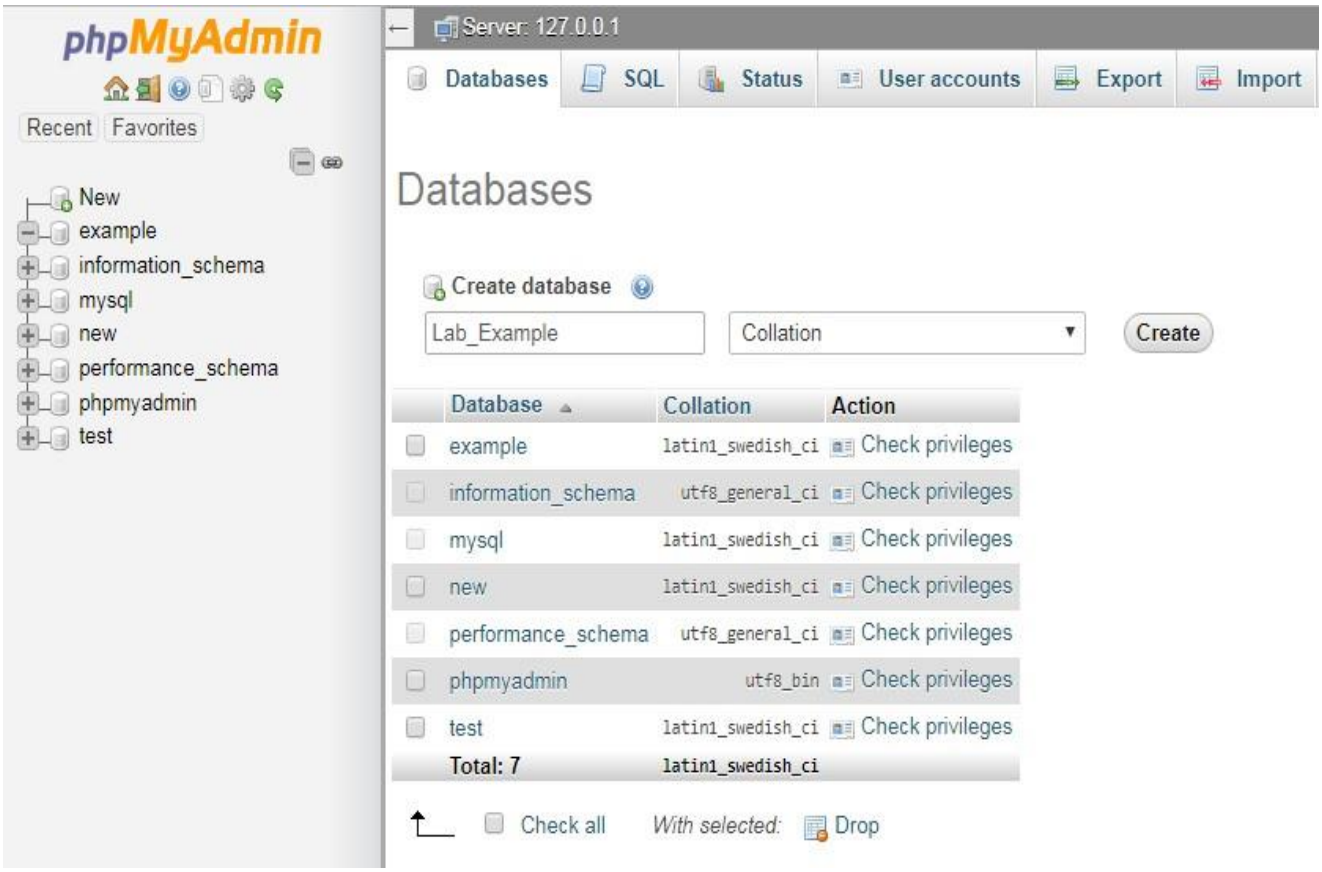
Two ways

1. Directly on phpMyAdmin
2. Front end (Text Editor)

1. Direct on phpMyAdmin:

Follow these steps:

1. Go to Databases in phpMyAdmin.
2. Fill the Database name field then Click Create.



2. Front End (Text Editor)

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

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

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

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

Creating Tables:

Two ways

1. Directly on phpMyAdmin
2. Front end (Text Editor)

1. Direct on phpMyAdmin:

Follow these steps:

1. Select the database in which you want to create table.
2. Fill out the table name and quantity of fields then click Go.
3. Give every field a proper name ,data type, size and Constraint (if any).
4. Click Go.

2.Front end (Text Editor)

```php
<?php

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

// 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();
?>
```

Inserting Data into Table:

```php
<?php

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

// 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 = "INSERT INTO myguests(`id`, `firstname`, `lastname`, `email`) VALUES ('1','john','dalton','john@gmail.com')";

if ($conn->query($sql) == TRUE) {
    echo "data inserted successfully";
} else {
    echo "Error inserting in  table: " . $conn->error;
}

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

Inserting multiple records:

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

// 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');";
$sql .= "INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('Mary', 'Moe', 'mary@example.com');";
$sql .= "INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('Julie', 'Dooley', 'julie@example.com')";

if ($conn->multi_query($sql) == TRUE) {
    echo "New records created successfully";
} else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}

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

Prepared Statements and Bound Parameters:

- A prepared statement is a feature used to execute the same (or similar) SQL statements repeatedly with high efficiency.
- Prepared statements basically work like this:
    - ✓ Prepare: An SQL statement template is created and sent to the database. Certain values are left unspecified, called parameters (labeled "?"). Example: INSERT INTO MyGuests VALUES(?, ?, ?)
    - ✓ The database parses, compiles, and performs query optimization on the SQL statement template, and stores the result without executing it
    - ✓ Execute: At a later time, the application binds the values to the parameters, and the database executes the statement. The application may execute the statement as many times as it wants with different values

Advantages:

Compared to executing SQL statements directly, prepared statements have three main advantages:

- Prepared statements reduces parsing time as the preparation on the query is done only once (although the statement is executed multiple times)
- Bound parameters minimize bandwidth to the server as you need send only the parameters each time, and not the whole query
- Prepared statements are very useful against SQL injections, because parameter values, which are transmitted later using a different protocol, need not be correctly escaped. If the original statement template is not derived from external input, SQL injection cannot occur.

```php
<?php
$servername="localhost";
$username="root";
$password="";
$dbname="new";
//create connection
$conn=new mysqli($servername,$username,$password,$dbname);
//check connection
if($conn->connect_error)
{
die("connection failed:".$conn->connect_error);
}
//prepare and bind
$stmt=$conn->prepare("INSERT INTO myguests(firstname,lastname,email) values(?,?,?)");
$stmt->bind_param("sss",$firstname,$lastname,$email);

$firstname="ammara";
$lastname="yaseen";
$email="ammara@gmail.com";
$stmt->execute();

$firstname="ayesha";
$lastname="farhan";
$email="ayesha@gmail.com";
$stmt->execute();

echo "new record added";
$stmt->close();
$conn->close();
?>
```

Selecting Data:

```php
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "new";
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
$sql = "SELECT id, firstname, lastname FROM MyGuests";
$result = $conn->query($sql);
if ($result->num_rows > 0) {
    // output data of each row
    while($row = $result->fetch_assoc()) {
        echo "id: " . $row["id"]. " - Name: " . $row["firstname"]. " " . $row["lastname"]. "<br>";
    }
} else {
    echo "0 results";
}
$conn->close();
```

Deleting Data:

```php
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "new";
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
// sql to delete a record
$sql = "DELETE FROM MyGuests WHERE id=3";
if ($conn->query($sql) ==TRUE) {
    echo "Record deleted successfully";
} else {
    echo "Error deleting record: " . $conn->error;
}
$conn->close();
?>
```

Updating Data:

```php
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "new";
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
$sql = "UPDATE MyGuests SET lastname='Doe' WHERE id=2";
if ($conn->query($sql) === TRUE) {
    echo "Record updated successfully";
} else {
    echo "Error updating record: " . $conn->error;
}
$conn->close();
?>
```