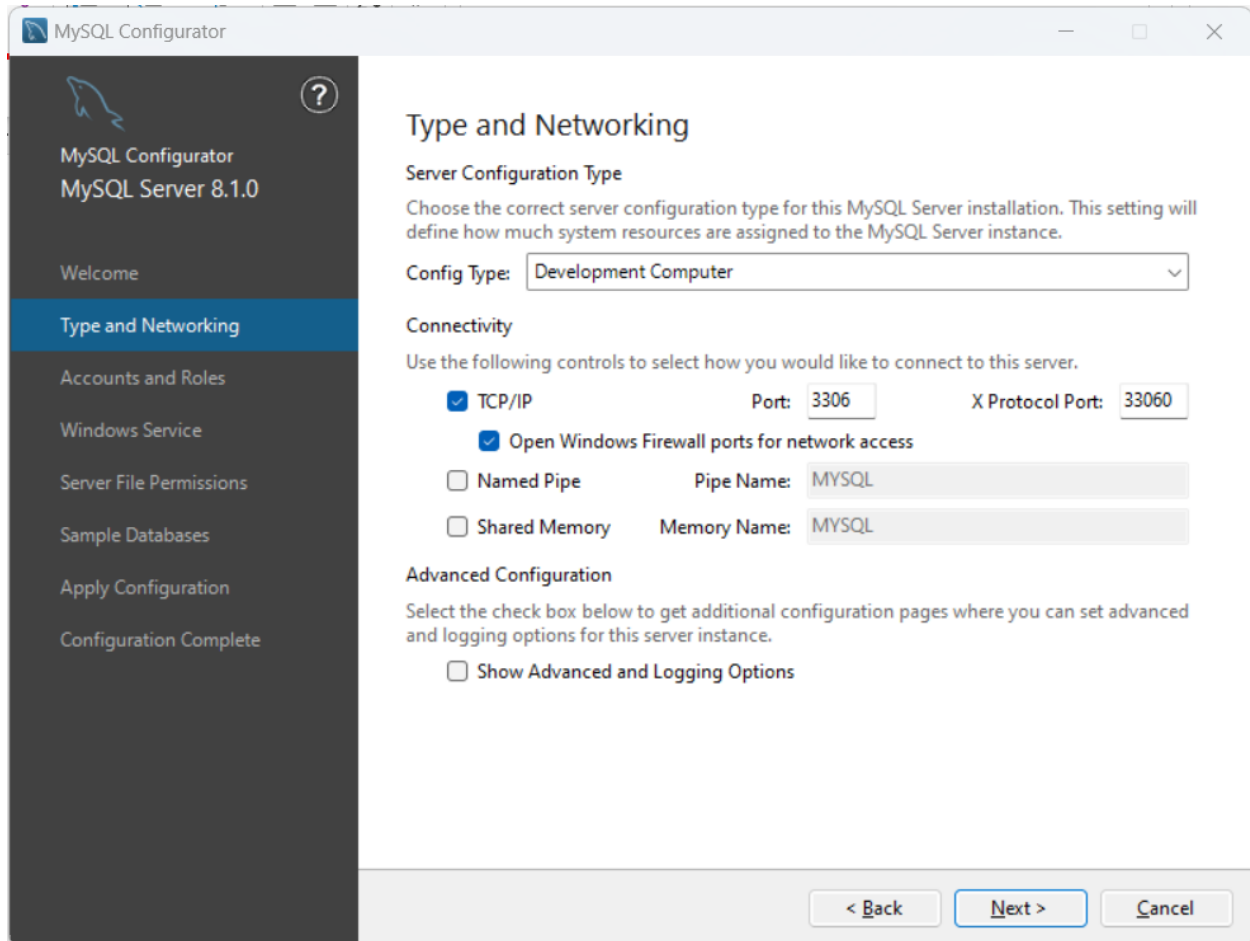


## WAD Lab 09: PHP

### I. Install MySQL

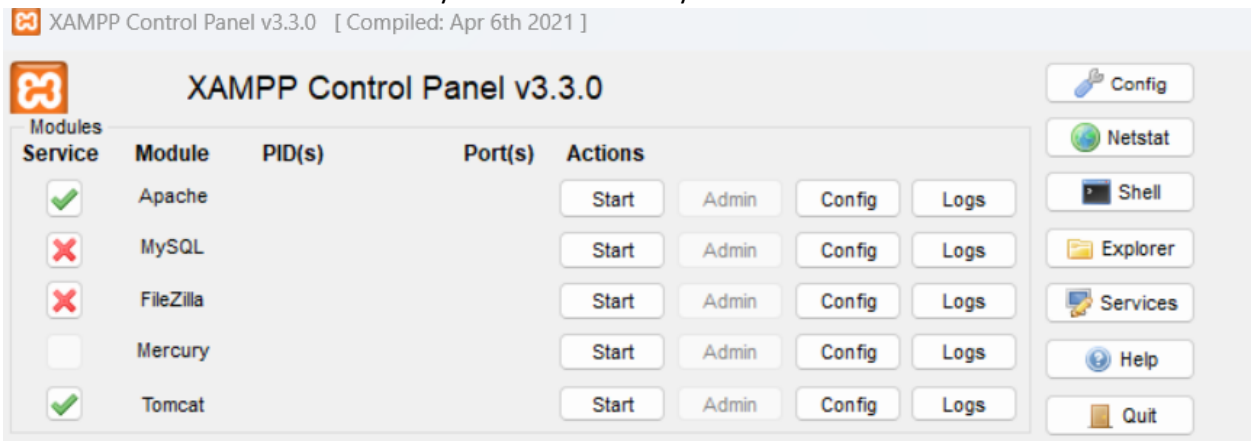
- Download and install **MySQL** server at the link: <https://dev.mysql.com/downloads/mysql/>
- After installing **MySQL** server, run MySQL configuration and set up your **MySQL** server, you can let everything as its default to install. Set up password for your root account of **MySQL** (this account has already been created with username '**root**'; you just need to set up password for it).



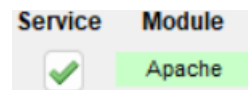
- After setting up the password, the next steps just leave them as default and press "Next" button and execute the configuration process.
- To make it easier to manage **MySQL** database and server you can download and install **MySQL Workbench** at this link: <https://dev.mysql.com/downloads/workbench/>
- After finishing setting up **MySQL Workbench** you can see your **MySQL** server on the workbench. The information of **MySQL** can be extracted from **MySQL Workbench**, and it is necessary for **NodeJS** to interact with **MySQL**.

### II. Install PHP

- To make it easy to install in every operating system, we will use **XAMPP** which is an easy to install **Apache** distribution containing **MariaDB**, **PHP**, and **Perl**. Just download and start the installer. It's that easy. The version used in this tutorial is 8.2.4
- Install **XAMPP** at this link: <https://www.apachefriends.org/download.html>
- MacOS: open Security & Privacy → Open any way
- Follow the installation instructions in the installer, in case you have not installed **MySQL** Server you can choose to install it in this installer.
- After finishing the installation, you can now open the control panel of **XAMPP** (Note: You must run this as **Administrator** on **Windows** or else it will not have the right to write files) and see the list of servers which you can start to run your **PHP** code.



- In this tutorial we will use **Apache** as local server to run **PHP** files.
- In **XAMPP** control panel, first of all please make sure that **Apache** has been installed by checking the Service column of Apache, if it is a **Tick** sign then **Apache** has been installed, on the other hand if it is an **X** sign you must click the **X** button to install **Apache**.



- After installing **Apache**, click the Config button and choose “**Apache (httpd.conf)**”, a text file will appear, scroll down to line “**Listen 80**” and change to “**Listen 1000**” or any port number you want (**Apache’s** HTTP port number in this tutorial will be changed to 1000), because some device deny to change port **80 (HTTP)** and **443 (HTTPS)** to point to others service, then we will change the https port number by choosing Config button of Apache and choose “**Apache (httpd-ssl.conf)**” and change the line “**Listen 443**” to “**Listen 1001**” or any other port number you want (**Apache’s** HTTPS port number in this tutorial will be changed to 1001),. After finishing configuration for port number of **Apache** you can now start Apache server by clicking on Start button and wait until the log said “**Status change detected: running**”

12:54:23 PM [Apache] Status change detected: running

- You can also check if the server has successfully started by access **localhost:1000** the UI of **Apache** default page will appear as in the figure below:

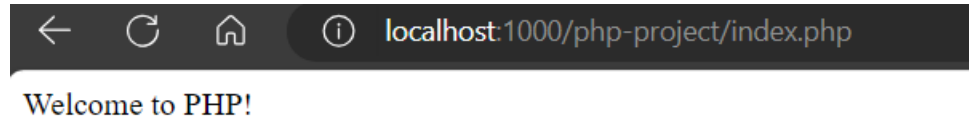


### III. Create PHP project:

- There are 2 ways to run a **PHP** project on **Apache** server:
  - o In your **XAMPP** stored location find folder "**htdocs**" and create your new **PHP** project folder in it, you can now run your **PHP** project code by access **localhost:1000/{your project folder name in htdocs folder}/{PHP file that you want to run}.php** (This way will be used in this tutorial).
  - o Create your **PHP** project folder somewhere on your computer and then after finish the coding, copy the folder to folder "**htdocs**" in your XAMPP stored location and you can now run your **PHP** project code by access **localhost:1000/{your project folder name in htdocs folder}/{PHP file that you want to run}.php**.
- Create new folder name "**php-project**" (you can change the name of project folder but remember to change it in the link whenever you want to run your PHP files) in "**htdocs**" folder in your **XAMPP** stored location and create **index.php** file in this project and copy the below code block to the file.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <?php
    echo "Welcome to PHP!"
  ?>
</body>
</html>
```

- Now try to access the link **localhost:1000/php-project/index.php**, the result will be shown as below figure:



- Now we will create a simple page for students to enter their personal information, **PHP** will help us to send entered information to **MySQL** as well as display back the information gotten from **MySQL**.
- Firstly, we must create a new database in **MySQL** server, the below **SQL** code will help you to create the new database as well as a new table name “**students**”.

```
-- Create new Database
CREATE DATABASE IF NOT EXISTS `mydb`

-- Create students table
CREATE TABLE IF NOT EXISTS students (
  id VARCHAR(255) PRIMARY KEY,
  `name` VARCHAR(255) NOT NULL,
  email VARCHAR(255) NOT NULL,
  gender int2 NOT NULL,
  department VARCHAR(255) NOT NULL
)
```

- Then, we will create an **HTML** UI for the page. The below code will help you to create a simple form for students to enter their information. The information here includes **student's ID, full name, department, email, gender**.

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <h1>Enter Student Information</h1>
  <form action="student.php" method="POST">
    <div style="margin-bottom: 10px">
      <input
        type="text"
        id="studentId"
        placeholder='Student ID'
        name="studentid"
      />
    </div>
    <div style="margin-bottom: 10px">
      <input
        type="text"
        id="name"
        placeholder='Fullname'
        name="fullname"
      />
    </div>
    <div style="margin-bottom: 10px">
      <input
        type="text"
        id="email"
        placeholder='Email'
        name="email"
      />
    </div>
    <div style="margin-bottom: 10px">
      <input
        type="text"
        id="department"
        placeholder='Department'
        name="department"
      />
    </div>
    <div style="margin-bottom: 10px">
      <input type="radio" value="1" name="gender" /> Male
      <input type="radio" value="0" name="gender" /> Female
    </div>
    <input type="submit">
  </form>
</body>
</html>

```

- Now we will create a new file name **student.php** to handle receiving submitted data and insert it into **MySQL** database and get the information from **MySQL** to display for users.
  - o First of all, we must create connection to our created database in **MySQL** server, in this tutorial we will use **MySQLi** extension which has already been installed along with **PHP**. Using this **PHP** code block in **student.php** file to create connection to **MySQL** server with **PHP**.

```
<?php
    $host = "localhost";
    $user = "root";
    $password = "root";
    $database = "mydb";

    // Create connection
    $conn = mysqli_connect($host, $user, $password, $database);

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

If the sentence “**Connected successfully**” appears when you access **localhost:1000/php-project/student.php** then you have successfully created a connection to database in **MySQL** server.

- Next, we will catch the submitted data and insert it to database using **PHP**, because we are using method **POST** so we will use “**\$\_POST**” to collect form-data. Add the below code block after you have successfully created connection to database.



```

$studentId = $_POST["studentid"];
$fullname = $_POST["fullname"];
$email = $_POST["email"];
$department = $_POST["department"];
$gender = $_POST["gender"];

# Insert into database
$sql = "INSERT INTO students (id, name, email, gender, department)
VALUES (
    '" . $studentId . "',
    '" . $fullname . "',
    '" . $email . "',
    '" . $gender . "',
    '" . $department . "'
)";

if (mysqli_query($conn, $sql)) {
    echo "<br/> Inserted successfully!";
} else {
    echo "<br/> Error: " . $sql . "<br/>" . mysqli_error($conn);
}

```

This code block helps you to catch the data which is sent through form to **students.php** by **POST** method, after that we create a variable name **\$sql** to contain the **INSERT SQL** statement and pass the data we just caught to the statement, the statement “**Inserted successfully!**” will display if there is no error while inserting data to database.

- Now we will get the inserted data based on student’s id and display to student after they have entered their information, the below code will help you to get the job done, please put it below the code block for inserting new data:

```

# Get student's information based on his/her id
$sql = "SELECT * FROM students WHERE id LIKE '%" . $studentId . "%'";
$result = mysqli_query($conn, $sql);

echo "
    <table border='1'>
        <tr>
            <th>Student ID</th>
            <th>Fullname</th>
            <th>Email</th>
            <th>Department</th>
            <th>Gender</th>
        </tr>
        <tr>
";
if(mysqli_num_rows($result) > 0){
    // Output data
    while($row = mysqli_fetch_assoc($result)){
        $gender_word = $row["gender"] == 0 ? "female" : "male";
        echo "
            <td>". $row["id"] . "</td>
            <td>". $row["name"] . "</td>
            <td>". $row["email"] . "</td>
            <td>". $row["department"] . "</td>
            <td>". $gender_word . "</td>
        ";
    }
}

echo "
    </tr>
</table>
";

mysqli_close($conn);

```

- And this is the whole file **student.php**, you can now try to enter the information and submit the form to see the result.



```

<?php
    $host = "localhost";
    $user = "root";
    $password = "root";
    $database = "mydb";

    // Create connection
    $conn = mysqli_connect($host, $user, $password, $database);

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

    $studentId = $_POST["studentid"];
    $fullname = $_POST["fullname"];
    $email = $_POST["email"];
    $department = $_POST["department"];
    $gender = $_POST["gender"];

    # Insert into database
    $sql = "INSERT INTO students (id, name, email, gender, department)
        VALUES (
            '" . $studentId . "',
            '" . $fullname . "',
            '" . $email . "',
            '" . $gender . "',
            '" . $department . "'
        )";

    if (mysqli_query($conn, $sql)) {
        echo "<br/> Inserted successfully!";
    } else {
        echo "<br/> Error: " . $sql . "<br/>" . mysqli_error($conn);
    }

    # Get student's information based on his/her id
    $sql = "SELECT * FROM students WHERE id LIKE '%" . $studentId . "%'";
    $result = mysqli_query($conn, $sql);

    echo "
        <table border='1'>
            <tr>
                <th>Student ID</th>
                <th>Fullname</th>
                <th>Email</th>
                <th>Department</th>
                <th>Gender</th>
            </tr>
            <tr>
";
    if(mysqli_num_rows($result) > 0){
        // Output data
        while($row = mysqli_fetch_assoc($result)){
            $gender_word = $row["gender"] == 0 ? "female" : "male";
            echo "
                <td>". $row["id"] . "</td>
                <td>". $row["name"] . "</td>
                <td>". $row["email"] . "</td>
                <td>". $row["department"] . "</td>
                <td>". $gender_word . "</td>
            ";
        }
    }

    echo "
        </tr>
    </table>
    ";

    mysqli_close($conn);
?>

```