

Laboratory Activity

Web Systems and Technology

Midterm

Procedure:

Step 1: Install a local server environment

Since phpMyAdmin requires a web server (Apache), a database server (MariaDB/MySQL), and PHP to run, you will need to install a bundled software package.

Popular options include:

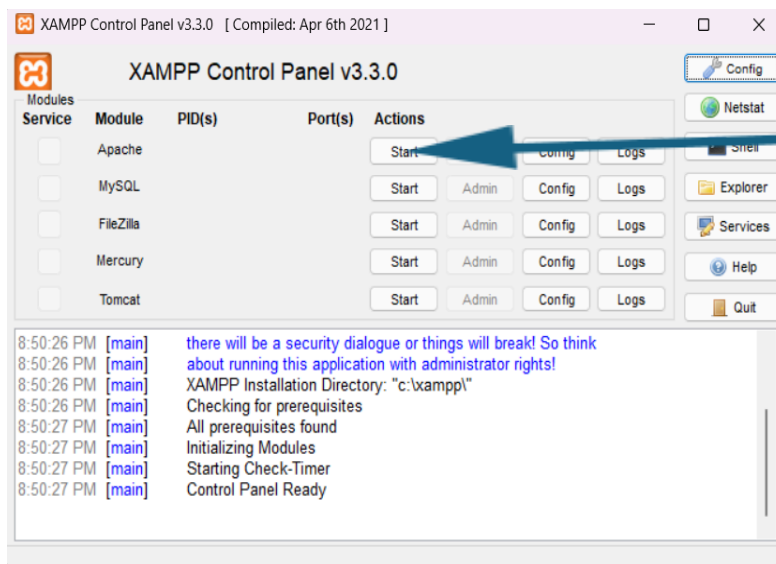
- **XAMPP:** Works on Windows, macOS, and Linux.
- **WAMP:** For Windows users.
- **MAMP:** For macOS users.

Step 2: Start the required services

After installing your server environment, you need to open its control panel and manually start the services.

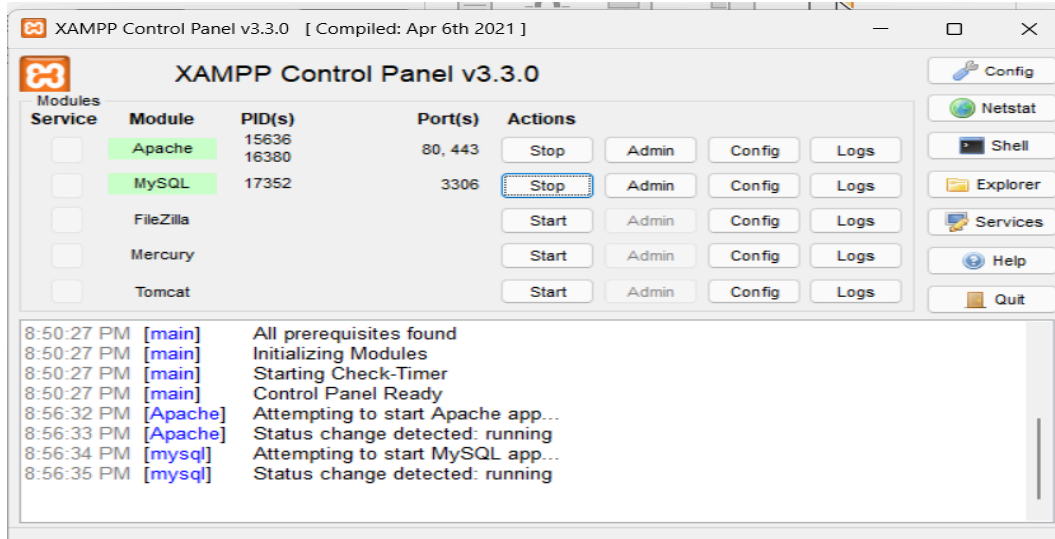
For XAMPP, this means:

1. Open the **XAMPP Control Panel**.
2. Click the **Start** button next to **Apache**.
3. Click the **Start** button next to **MySQL**.
4. Ensure that both modules are running successfully.



Make sure to click the Start button to open the Apache and MySQL

- Once it green your server is now running



- Make sure to check the ports of Apache and MySQL as follows

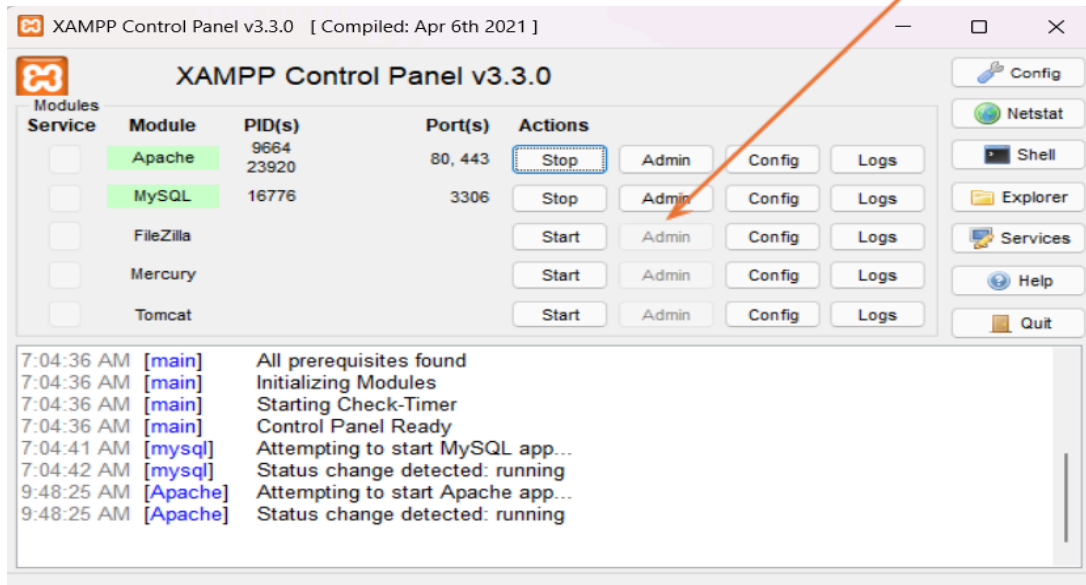
Module	PID(s)	Port(s)
Apache	26368 3028	80, 443
MySQL	25340	3306

- You are now ready to test your server

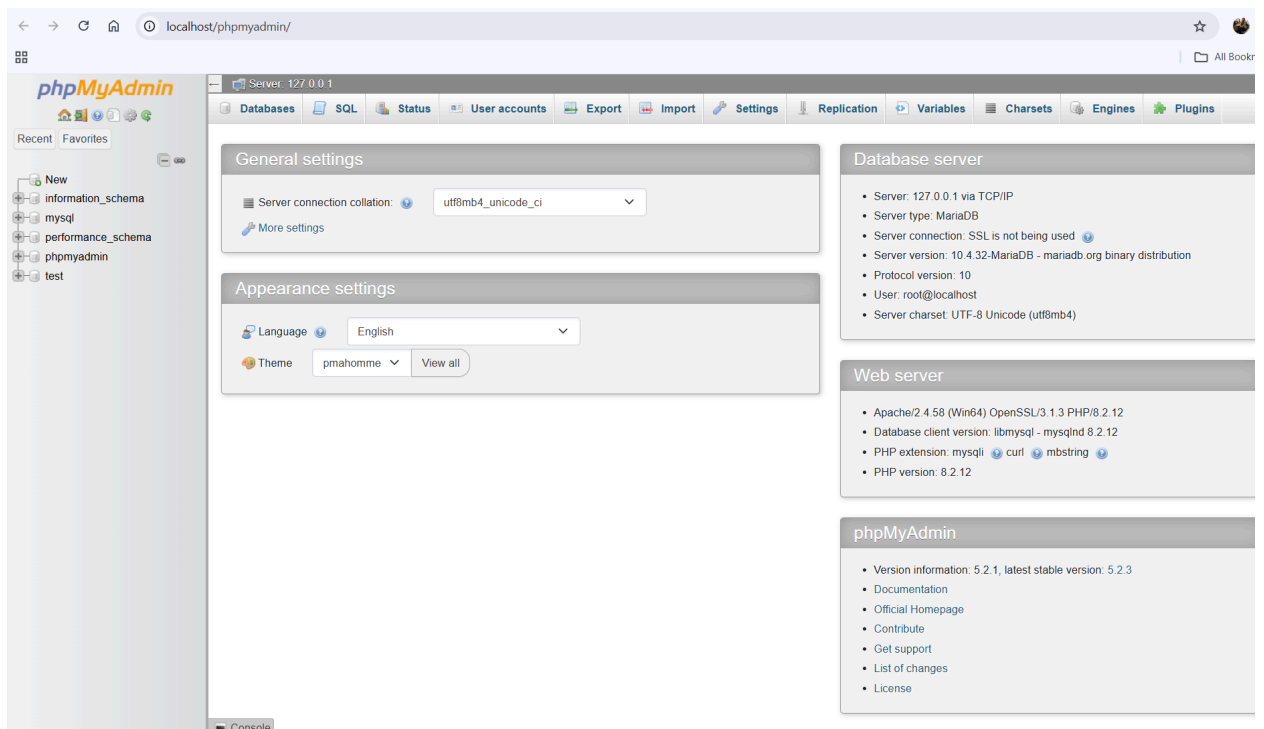
Step 3: Access phpMyAdmin

Once your servers are running, you can access the phpMyAdmin interface through your web browser.

1. Open your web browser.
2. Type `http://localhost/phpmyadmin` in the address bar and press Enter.
3. Log in with your credentials. If you are using a new installation, the default username is often **root** with **no password**.



Then it will redirect you to the admin panel



Step 4: Plan your database schema

Before creating your database, consider what information you need to store and how it should be organized. Proper planning helps avoid structural problems later on.

A simple planning process involves:

- **Identifying entities:** Figure out the main "things" your database needs to track, like Users, Products, or Orders.
- **Defining attributes:** Decide on the specific data points for each entity. For a Users table, this might include first_name, last_name, and email.
- **Specifying primary keys:** Choose a unique identifier for each table, such as a user ID. This is typically an auto-incrementing integer.
- **Establishing relationships:** Determine how your tables will connect to each other. For example, the Orders table might link to a Users table.

This time:

- I want you to explore or try to create a sample database as your activity for today without connecting it to php (or to your website)
- Name it on the desired table name.
- Screenshot all your works and output in this document and sent it through your github (filename: Activity 3 -10/30) this is a different filename from your Activity 2.

The screenshot shows the phpMyAdmin interface for a database named 'ewalletdb'. The selected table is 'storing_notifications'. The 'Table structure' tab is active, displaying the following table structure:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	NotificationID	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	UserID	int(11)			No	None			Change Drop More
3	NotificationType	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
4	Content	text	utf8mb4_general_ci		No	None			Change Drop More
5	Timestamp	timestamp			No	current_timestamp()			Change Drop More
6	Status	varchar(20)	utf8mb4_general_ci		No	None			Change Drop More

Below the table structure, there are options to 'Check all', 'With selected:', 'Browse', 'Change', 'Drop', 'Primary', 'Unique', 'Index', 'Spatial', and 'Fulltext'. There is also a 'Print' button and a 'Propose table structure' button. A 'Move columns' section shows '1' column(s) moved 'after Status'. The 'Indexes' tab is active, showing a primary index on 'NotificationID'.

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Rename Drop	PRIMARY	BTREE	Yes	No	NotificationID	0	A	No	

At the bottom, there is a 'Create an index on' section with '1' column(s) and a 'Go' button. The 'Partitions' tab is also visible at the bottom.