

<https://www.upsite.com/blog/sealing-rack-need/>

## MODULE 2d

# Centralized Temperature & Humidity Monitoring System for Server Racks

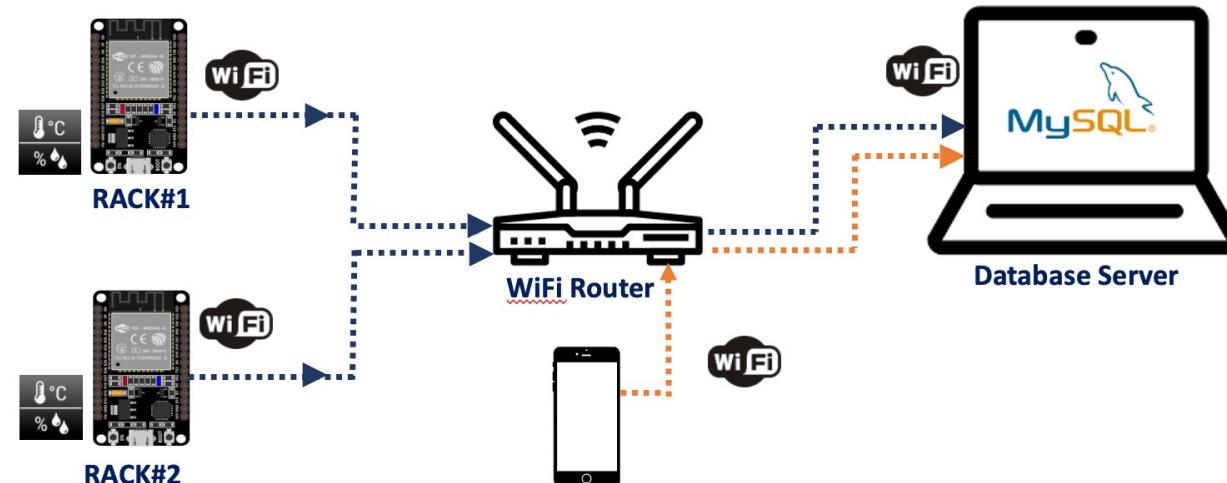
safyzan salim  
019 622 0575

## Project Overview

This tutorial will teach how to save data from ESP32s (or any ethernet microcontrollers) into MySQL.

### **Scenario:**

You are required to monitor 2 server racks in a data center; i.e., **Rack#1** & **Rack#2** by using 2 ESP32s and 2 DHT11 sensors. Both readings need to be saved into a single table. The temperature & humidity of both rooms can be monitored thru webpage.



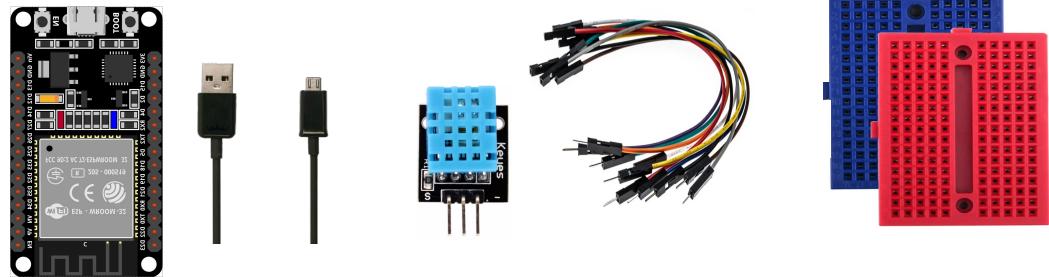
ZXZXZ

## 2d. Centralized Temperature & Humidity Monitoring System for Server Racks

### Components

#### **Hardware:**

- 1 x ESP-32 Wifi+Bluetooth 2-In-1 Development Board for Arduino (30 pin) + Cable.
- 1 x Temperature & Humidity Sensor, DHT11.
- Jumpers (male to female).
- 2 x Mini Breadboard.



#### **Software:**

- XAMPP ver 8.0.1(PHP 8.0.1) for Windows / OS X.



#### **What is XAMPP?**

XAMPP is the most popular PHP development environment

XAMPP is a completely free, easy to install Apache distribution containing MariaDB, PHP, and Perl. The XAMPP open source package has been set up to be incredibly easy to install and to use.

## Solution

→ **Step 1: Propose block diagram & flowcharts**

A story board of the whole project which will give clear picture what you need to do.

→ **Step 2: Install Apache Web Server**

- i. Install XAMPP (once), start web server service
- ii. Start XAMPP service
- iii. Test XAMPP page & PHPMyAdmin page

→ **Step 3: Creating MySQL Database**

- i. Create database + test
  - ii. Create a PHP file for medium of transferring information from ESP32 to MySQL + Test
- \*\*Test with dummy data / manually inject data to database

→ **Step 4: Preparing PHP script to insert data to MySQL database**

- i. Configure \*.php scripts.
- ii. PHP script to display database.
- iii. HTTP GET & HTTP POST.

## Solution

### → Step 5: Upload sketch to ESP32

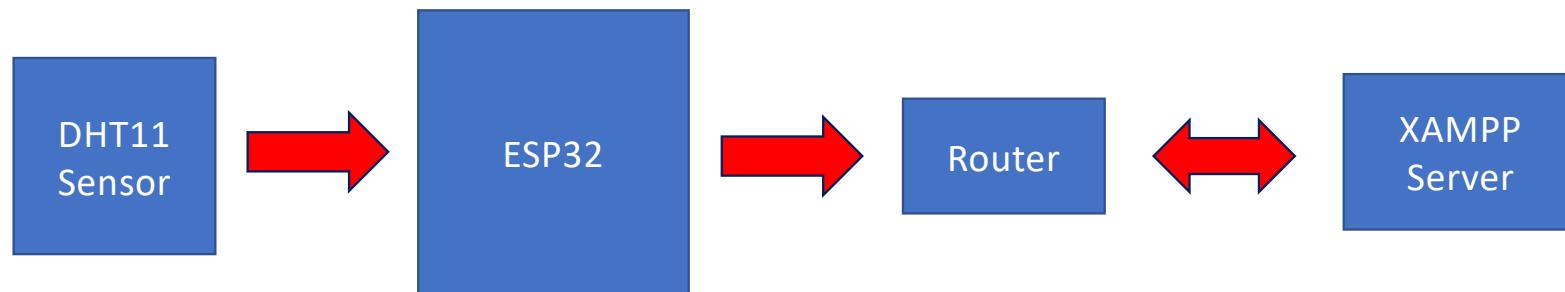
- i. Connect DHT11 to ESP32.
- ii. Configure ***sketch-vii\_esp32-dht11-http-post.ino***.

## 2d. Centralized Temperature & Humidity Monitoring System for Server Racks

### Solution: Step 1

PROPOSED BLOCK DIAGRAM & FLOWCHART

#### Project Block Diagram

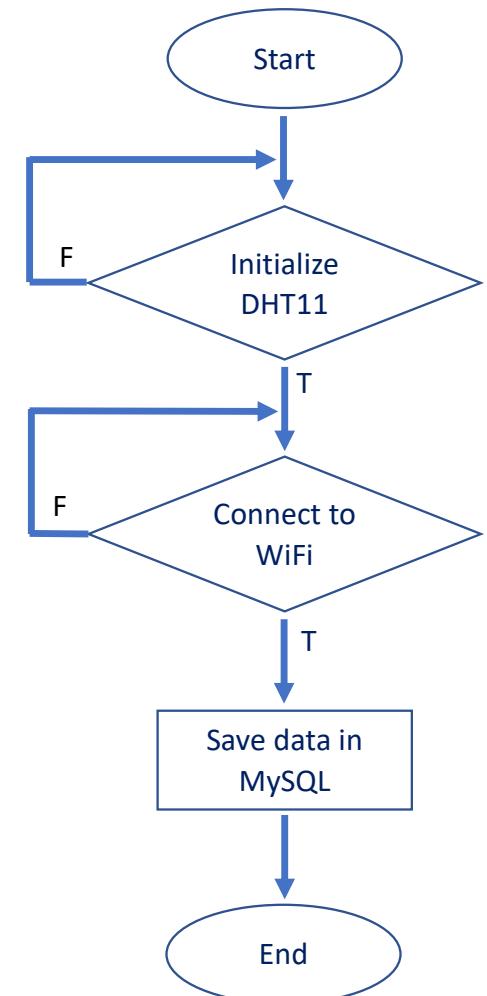
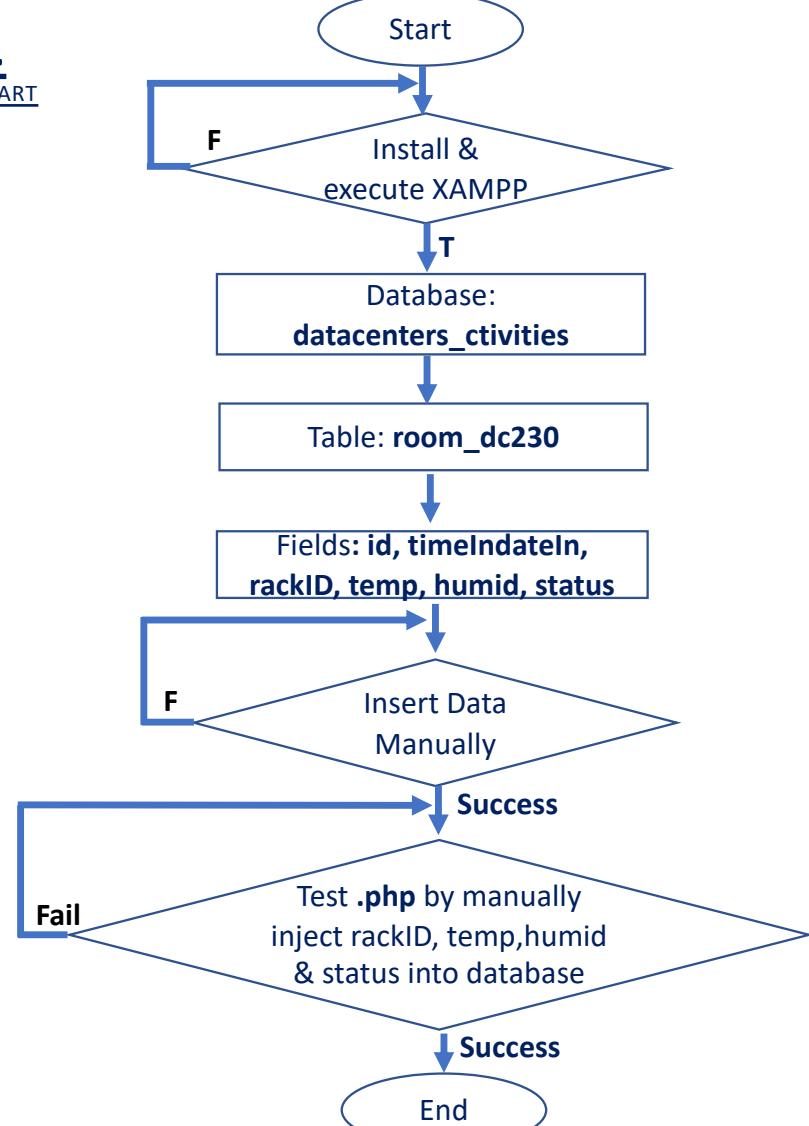


## 2d. Centralized Temperature & Humidity Monitoring System for Server Racks

### Solution: Step 1

PROPOSED BLOCK DIAGRAM & FLOWCHART

#### Flowchart



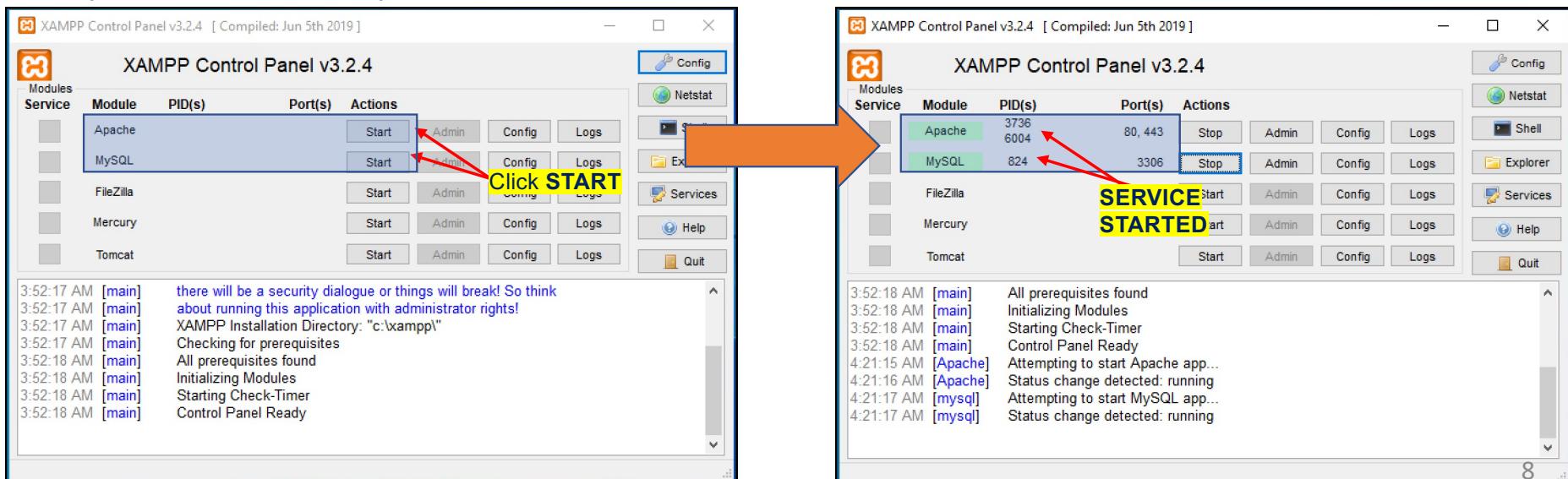
## 2d. Centralized Temperature & Humidity Monitoring System for Server Racks

### Solution: Step 2 INSTALL, START & TEST XAMPP SERVICES

- Download and install latest version of XAMPP from <https://www.apachefriends.org/download.html>
- This link shows full tutorial on installing XAMPP on windows <https://blog.templatetoaster.com/install-xampp-on-windows/>
- Once installed, launch XAMPP Control Panel (7.3.26)

For Windows (v7.3.26):

- to run, go to **c://xampp/control-control.exe** → Start **MySQL Database & Apache Web Service**
- open your browser and type <http://localhost/>

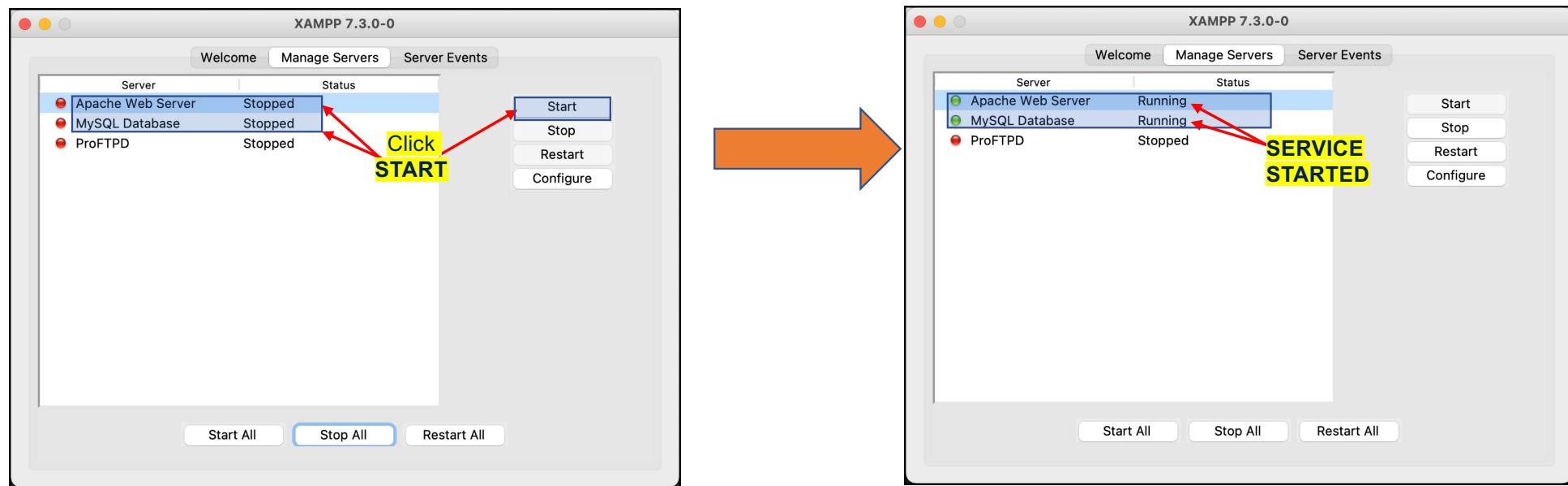


## 2d. Centralized Temperature & Humidity Monitoring System for Server Racks

### Solution: Step 2 INSTALL, START & TEST XAMPP SERVICES

For OS X (v7.3.0):

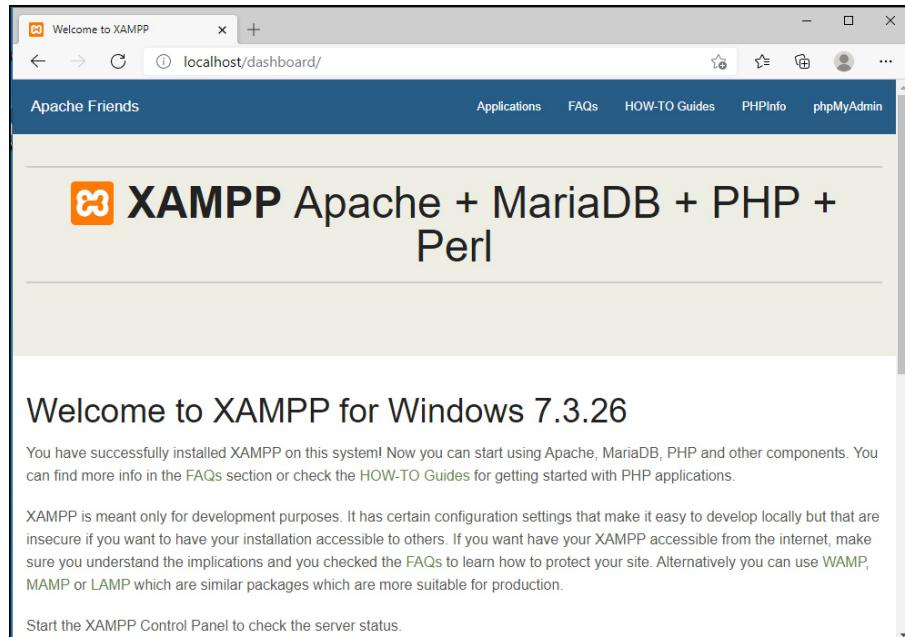
- go to **Applications>XAMPP>manager-osx** → Start **MySQL Database & Apache Web Service**
- open your browser and type <http://localhost/>



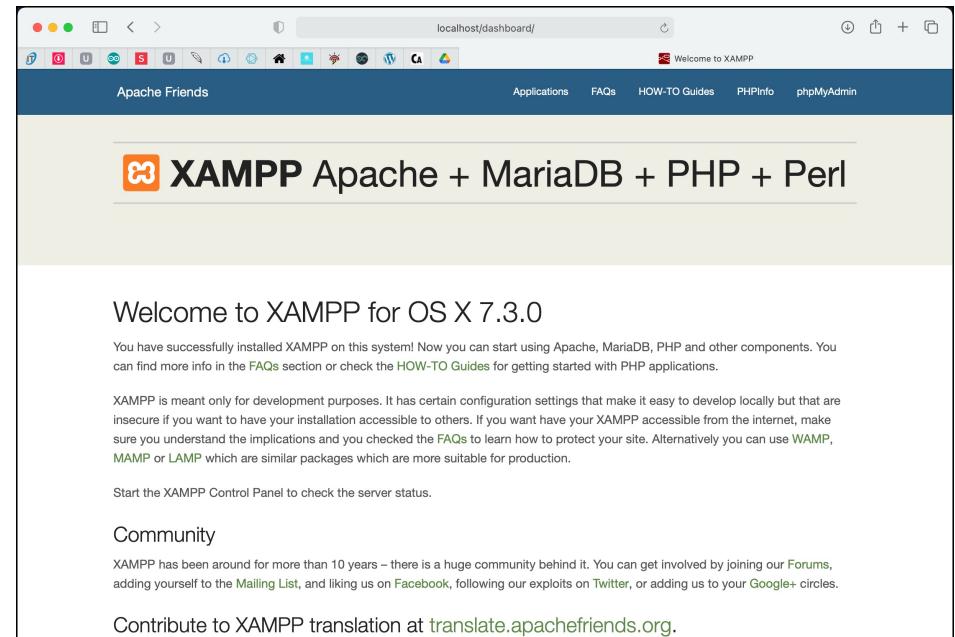
## 2d. Centralized Temperature & Humidity Monitoring System for Server Racks

### Solution: Step 2 INSTALL, START & TEST XAMPP SERVICES

→ XAMPP landing page can be found at <http://localhost/>



Windows system

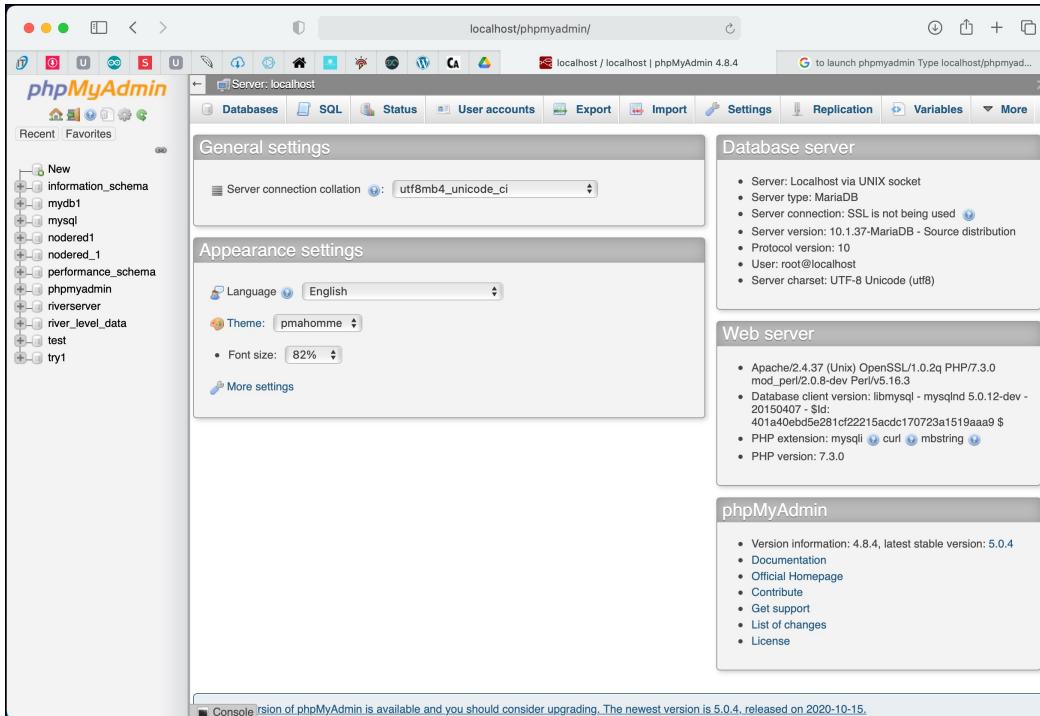


OS X system

## 2d. Centralized Temperature & Humidity Monitoring System for Server Racks

### Solution: Step 2 INSTALL, START & TEST XAMPP SERVICES

- To use MySQL database, point your browser to <http://localhost/phpmyadmin>
- You can also click phpMyAdmin link at XAMPP dashboard.
- PHPMyAdmin is a third party tool to manage data inside the database.



## 2d. Centralized Temperature & Humidity Monitoring System for Server Racks

### Solution: Step 3 CREATING MySQL DATABASE

→ Set up database with the following properties:

Database Name: **datacenter\_activities**

Table Name: **room\_dc230**

Columns: **6**

- The number of characters used should be equal to or less than 64.
- The name should comprise of letters, numbers and underscore.
- The DB name should not start with a number.
- It should be relevant to the topic for which it is being created.

<https://www.javatpoint.com/creating-mysql-database-with-xampp>

NAME	TYPE	LENGTH	ADDITIONAL SETTING
id	INT	11	Index: <b>PRIMARY</b> AI: <b>YES</b>
logDateTime	DATETIME	-	Default: <b>CURRENT_TIMESTAMP</b>
rackID	VARCHAR	255	-
tempValue	VARCHAR	255	-
humidValue	VARCHAR	255	-
rackStatus	VARCHAR	255	-

**AI** = Auto Increment

- = empty

#use appropriate name for variables (dbname, tablename...) that reflect with the task/activity

12

## 2d. Centralized Temperature & Humidity Monitoring System for Server Racks

### Solution: Step 3 CREATING MySQL DATABASE

→ Database Name: **datacenter\_activities**

5

6

The screenshot shows the phpMyAdmin interface for creating a new database. A red arrow points from the 'New' button in the sidebar (labeled 1) to the 'Create database' input field (labeled 2). Another red arrow points from the 'Create' button (labeled 3) to the 'Create' button in the top right corner of the main panel (labeled 5). The database name 'datacenters\_activities' is typed into the input field. The top right corner also has a '6' labeled 6.

localhost/phpmyadmin/server\_databases.php?server=1

localhost / localhost | phpMyAdmin 4.8.4

Server: localhost

Databases SQL Status User accounts Export Import Settings Replication More

New

information\_schema

mydb1

mysql

nodered1

nodered\_1

performance\_schema

phpmyadmin

riverserver

river\_level\_data

test

try1

Existing databases

Create database

datacenters\_activities latin1\_swedish\_ci Create

Filters

Containing the word:

Database	Collation	Action
information_schema	utf8_general_ci	Check privileges
mydb1	latin1_swedish_ci	Check privileges
mysql	latin1_swedish_ci	Check privileges
nodered1	latin1_swedish_ci	Check privileges
nodered_1	latin1_swedish_ci	Check privileges
performance_schema	utf8_general_ci	Check privileges
Console	admin	Check privileges

13

## 2d. Centralized Temperature & Humidity Monitoring System for Server Racks

### Solution: Step 3 CREATING MySQL DATABASE

→ Table Name: **room\_ds230** Number of columns: **6**

The screenshot shows the phpMyAdmin interface. The left sidebar lists databases: New, datacenters\_activities (highlighted with a yellow box labeled 'Just created'), information\_schema, mydb1, mysql, nodered1, nodered\_1, performance\_schema, phpmyadmin, riverserver, river\_level\_data, test, and try1. The main area shows the 'datacenters\_activities' database selected. A message says 'No tables found in database.' Below it, a 'Create table' form is open. The 'Name:' field contains 'room\_dc230'. To its right, a red arrow points to the 'Number of columns:' dropdown, which has the value '6'. Another red arrow points to the dropdown's current value '5'. A third red arrow points to the 'Go' button, which is highlighted with a blue box and has the value '6' next to it. The bottom of the interface shows a 'Console' tab.

## 2d. Centralized Temperature & Humidity Monitoring System for Server Racks

### Solution: Step 3 CREATING MySQL DATABASE

→ Table Name: room\_ds230 Number of columns: 6

The screenshot shows the 'Structure' tab of the phpMyAdmin interface for the 'room\_dc230' table. The table has six columns:

- id**: Type INT, Length/Values 11, Default NULL, Null, Index PRIMARY, A\_I checked, Comments 12.
- logDateTime**: Type DATETIME, Length/Values CURRENT\_TIME, Default NULL.
- rackID**: Type VARCHAR(255), Length/Values 18, Default None.
- tempValue**: Type VARCHAR(255), Length/Values 21, Default None.
- humidValue**: Type VARCHAR(255), Length/Values 24, Default None.
- rackStatus**: Type VARCHAR(255), Length/Values 27, Default None.

A modal window titled 'Add index' is open, showing an index named 'PRIMARY' with 'Column' set to 'id'. A red arrow labeled 11 points from the 'Index' section of the table definition to the 'Column' dropdown in the modal. Another red arrow labeled 10 points from the 'Index' section to the 'A\_I' checkbox. A red arrow labeled 12 points from the 'Comments' field to the 'Comments' field in the modal.

At the bottom right of the table definition, a red arrow labeled 28 points to the 'Save' button.

## 2d. Centralized Temperature & Humidity Monitoring System for Server Racks

### Solution: Step 3 CREATING MySQL DATABASE

→ What you should know on PHPMyAdmin panel (at least).

The screenshot shows the PHPMyAdmin interface with the following details:

- Left sidebar:** Shows the database structure with 'datacenters\_activities' selected. It contains a 'room\_dc230' table.
- Top navigation bar:** Includes tabs for Structure, SQL, Search, Query, Export, Import, Operations, Privileges, Routines, Events, and More. The 'Structure' tab is active.
- Table list:** Shows one table named 'room\_dc230'. The table has two rows of data. The first row is highlighted with orange boxes around the 'Action' column and the 'Sum' button. The second row is also highlighted with orange boxes around the 'Action' column and the 'Sum' button.
- Action buttons:** Below the table list are several buttons: 'Browse', 'Structure', 'Search', 'Insert', 'Empty', and 'Drop'.
- Table structure:** Below the table list, it shows the table's structure with columns: 'Rows', 'Type', 'Collation', 'Size', and 'Overhead'. The table uses InnoDB storage engine, latin1\_swedish\_ci collation, and is 16 KiB in size.
- Create table form:** A 'Create table' form is visible at the bottom left, with 'Name:' and 'Number of columns: 4' fields.
- Bottom buttons:** 'Go' and 'Console' buttons.

Browse → to view / edit / delete data stored in room\_dc230, increment counter will proceed to next number.

Structure → to view / edit / delete / add table structure.

Insert → to manually insert data into room\_dc230.

**Empty → to empty room\_dc230 database, any increment counter will start from 1.**

**Drop → will delete room\_dc230 table.**

## 2d. Centralized Temperature & Humidity Monitoring System for Server Racks

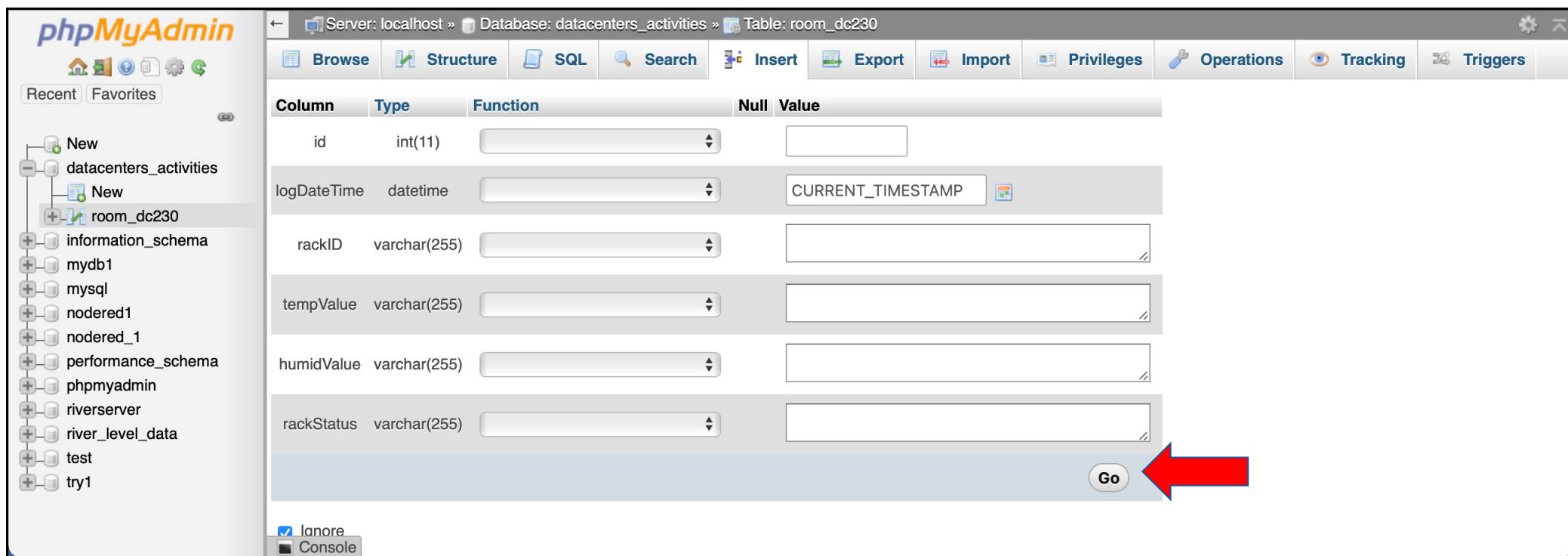
### Solution: Step 3 CREATING MySQL DATABASE

→ Click **a** **Browse** to view room\_dc230 database. It is empty since no data in it.

The screenshot shows the phpMyAdmin interface. On the left, the database tree is visible with the 'datacenters\_activities' database selected. Inside it, the 'room\_dc230' table is selected. The main area shows the table structure with columns: id, logDateTime, rackID, tempValue, humidValue, and rackStatus. A message at the top states: 'MySQL returned an empty result set (i.e. zero rows). (Query took 0.0006 seconds.)'. Below the message is a SQL query: 'SELECT \* FROM `room\_dc230`'. The table body displays the message 'An empty database'. At the bottom, there are options for 'Query results operations' like 'Create view' and 'Bookmark this SQL query'.

## Solution: Step 3 CREATING MySQL DATABASE

- In order to insert data manually into room\_dc230, select  **Insert**.
- For this time, leave **ALL** text box empty. Then click **Go** to save into room\_dc230 database.



The screenshot shows the phpMyAdmin interface for the 'room\_dc230' table. The table structure is as follows:

Column	Type	Function	Null	Value
id	int(11)			
logDateTime	datetime			CURRENT_TIMESTAMP
rackID	varchar(255)			
tempValue	varchar(255)			
humidValue	varchar(255)			
rackStatus	varchar(255)			

A red arrow points to the 'Go' button at the bottom right of the form.

## 2d. Centralized Temperature & Humidity Monitoring System for Server Racks

### Solution: Step 3 CREATING MySQL DATABASE

→ Observe the red arrow. What do you think?

→ a Click Browse link to view the input.

```
INSERT INTO `room_dc230` (`id`, `logDateTime`, `rackID`, `tempValue`, `humidValue`, `rackStatus`) VALUES (NULL, CURRENT_TIMESTAMP, '', '', '', ''');
```

The screenshot shows the phpMyAdmin interface for the 'room\_dc230' table in the 'datacenters\_activities' database. The 'Browse' tab is selected. A success message indicates '1 row inserted.' with 'Inserted row id: 1'. The SQL query entered is:

```
INSERT INTO `room_dc230` (`id`, `logDateTime`, `rackID`, `tempValue`, `humidValue`, `rackStatus`) VALUES (NULL, CURRENT_TIMESTAMP, '', '', '', ''');
```

A yellow box highlights the part of the query where values are being inserted. A red arrow points from this highlighted area to the corresponding row in the SQL query editor below. Another red arrow points to the '[Edit inline]' button. A yellow box also highlights the text 'Blank data inserted into rackID, tempValue, humidValue & rackStatus'.

Below the query editor, the 'Columns' section lists the table's columns: id, logDateTime, rackID, tempValue, humidValue, and rackStatus. At the bottom of the interface, there are buttons for SELECT \*, SELECT, INSERT, UPDATE, DELETE, Clear, and Format.

## 2d. Centralized Temperature & Humidity Monitoring System for Server Racks

### Solution: Step 3 CREATING MySQL DATABASE

- Congrats! You managed to add your first data into room\_dc230 database.
- Click **Edit** link if you wish to edit the data.
- **Delete** will not reset the *id* into **zero**. Next data entered will get the following running number.

The screenshot shows the phpMyAdmin interface for the 'room\_dc230' table. The table has columns: id, logDateTime, rackID, tempValue, humidValue, and rackStatus. There is one row with the following values:

	<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	id	logDateTime	rackID	tempValue	humidValue	rackStatus
				1	2021-02-09 01:21:19				

A red arrow points to the 'Delete' link in the row header. The status bar at the bottom right of the table area also contains a red arrow pointing to the same 'Delete' link.

## 2d. Centralized Temperature & Humidity Monitoring System for Server Racks

### Solution: Step 3 CREATING MySQL DATABASE

→ Go to **Insert** link and add the information as show by red arrow & press **Go** to save the data.

The screenshot shows the phpMyAdmin interface for the 'room\_dc230' table. The table structure is as follows:

Column	Type	Function	Null	Value
id	int(11)			
logDateTime	datetime			CURRENT_TIMESTAMP
rackID	varchar(255)			r1
tempValue	varchar(255)			24
humidValue	varchar(255)			70
rackStatus	varchar(255)			Normal

Red arrows numbered 1 through 5 indicate the following steps:

- Arrow 1 points to the 'rackID' input field containing 'r1'.
- Arrow 2 points to the 'tempValue' input field containing '24'.
- Arrow 3 points to the 'humidValue' input field containing '70'.
- Arrow 4 points to the 'rackStatus' input field containing 'Normal'.
- Arrow 5 points to the 'Go' button at the bottom right of the form.

## 2d. Centralized Temperature & Humidity Monitoring System for Server Racks

### Solution: Step 3 CREATING MySQL DATABASE

→ Red arrow shows the new data that have been entered.

The screenshot shows the phpMyAdmin interface for the 'room\_dc230' table in the 'datacenters\_activities' database. The table has columns: id, logDateTime, rackID, tempValue, humidValue, and rackStatus. There are two rows of data:

	1	2021-02-09 01:59:08	r1	24	70
	2	2021-02-09 02:04:38	r1	24	70

A red arrow points to the second row of data, indicating the newly entered data.

## 2d. Centralized Temperature & Humidity Monitoring System for Server Racks

### Solution: Step 4 PHP SCRIPTS & HTDOCS & CODE EXPLANATION & TEST RUN

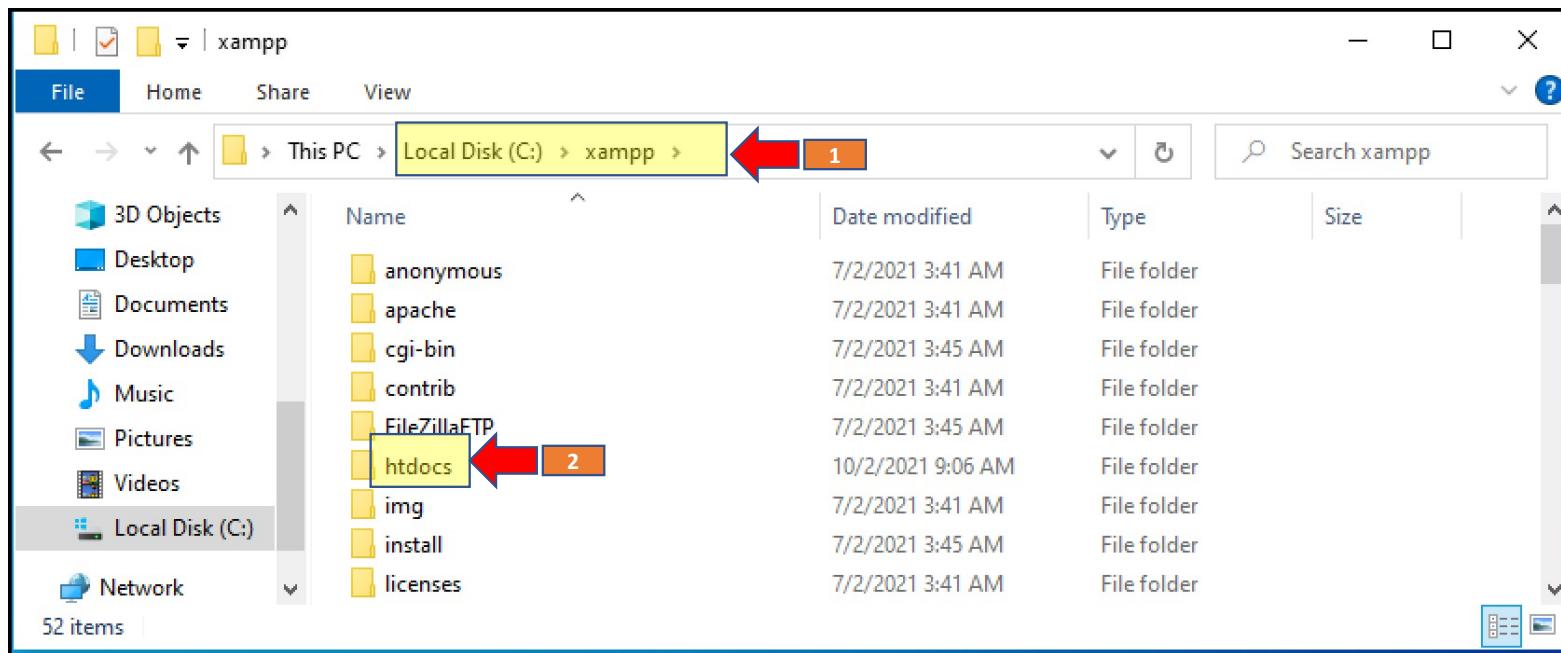
→ Two PHP files are needed; esptodb.php & webview.php.



\*\***esptodb.php** is the medium of ESP32 to save information from racks into database  
 \*\* **webview.php** is to display datacenters activities on web page.

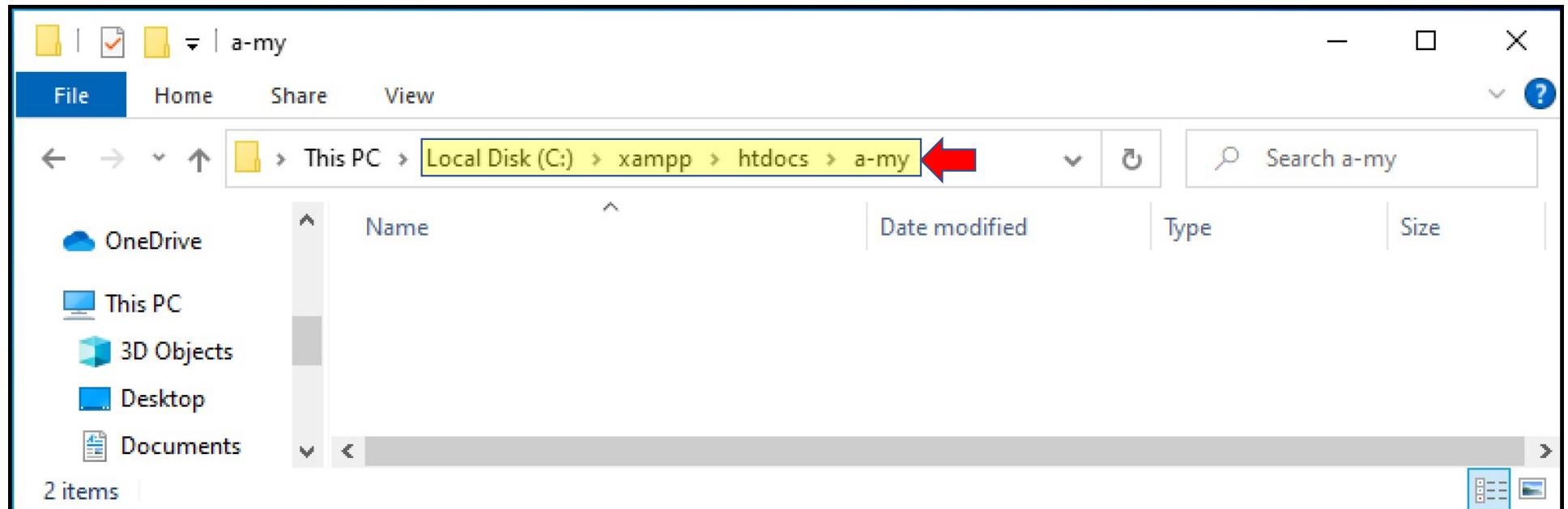
## Solution: Step 4 PHP SCRIPTS & HTDOCS & CODE EXPLANATION & TEST RUN

- In order to publish any website either PHP or HTML, you need to save the files in a **specific folder**, named **htdocs**. It can be found in XAMPP folder.
- It is advised that each project has its own folder. This may avoid you to override files that carries same filename.



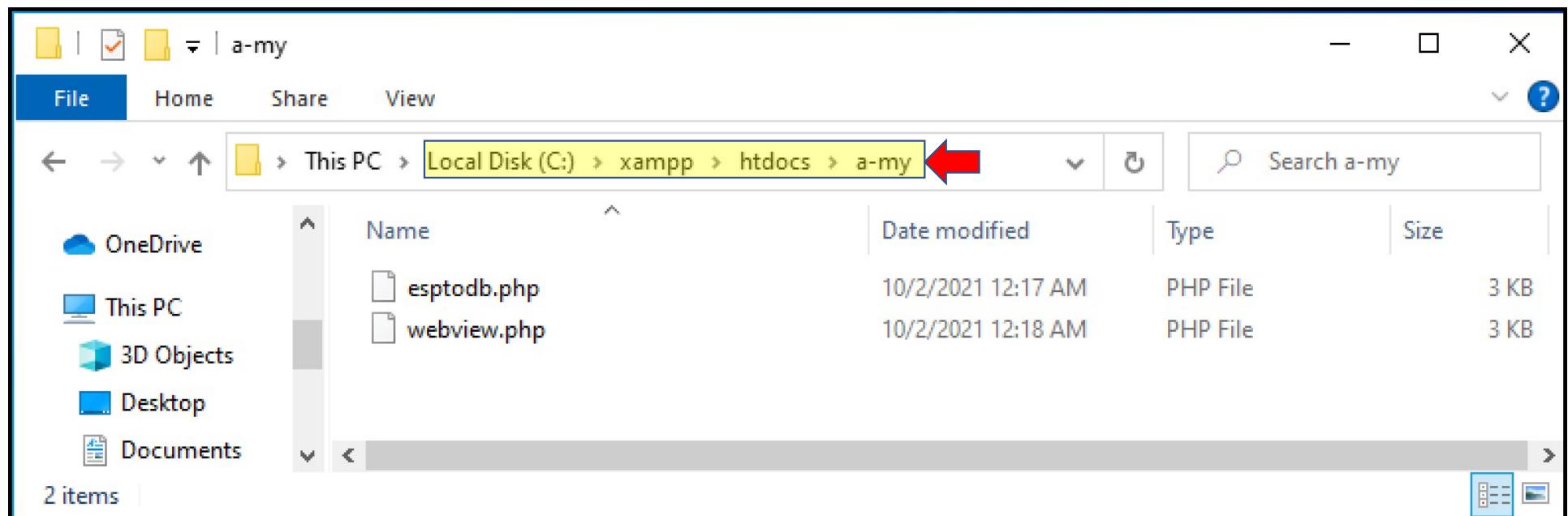
## Solution: Step 4 PHP SCRIPTS & HTDOCS & CODE EXPLANATION & TEST RUN

→ Navigate to **htdocs** & create new folder named **a-my**.



## Solution: Step 4 PHP SCRIPTS & HTDOCS & CODE EXPLANATION & TEST RUN

- Download “**esptodb.php**” & “**webview.php**” from <http://bit.ly/3p2pmNd>.
- Extract & paste the files into **a-my** folder.
- ESP32 will use “**esptodb.php**” as a medium to save information into MySQL while client will use “**webview.php**” to view the information thru web browsers.



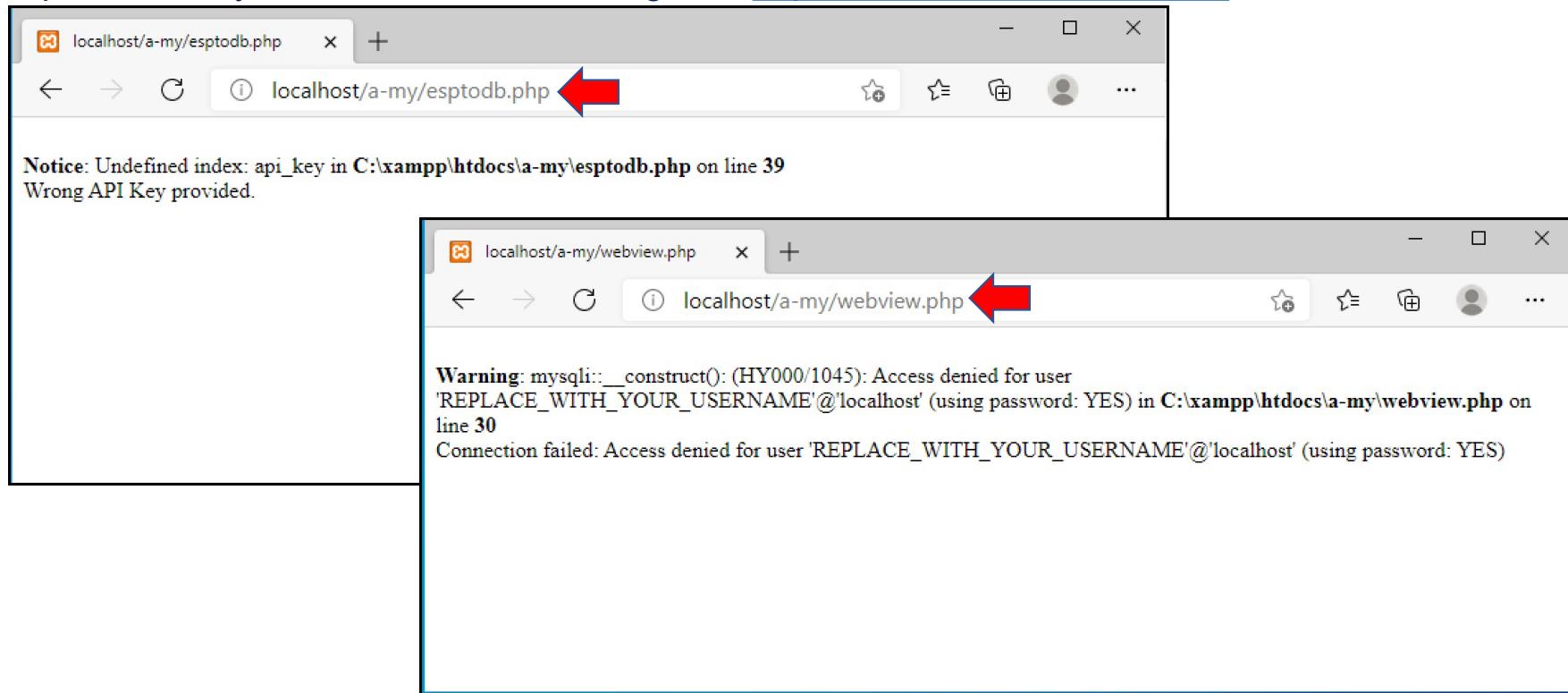
## Solution: Step 4 PHP SCRIPTS & HTDOCS & CODE EXPLANATION & TEST RUN

→ Open up your web browser & type **localhost/a-my/** in the address bar. You should get the similar output. If you failed to see this, go to your XAMPP C-Panel & check whether the server services has been deployed or not.



## Solution: Step 4 PHP SCRIPTS & HTDOCS & CODE EXPLANATION & TEST RUN

- Click at the file one at a time. There are errors/warnings since the files were not properly configured.
- You need a text editor to edit both files & we are going to use Sublime Text as the editor for PHP scripts. You may download it from the following link: <https://www.sublimetext.com/>



## Solution: Step 4 PHP SCRIPTS & HTDOCS & CODE EXPLANATION & TEST RUN

- Open ***webview.php*** with **sublimetext** or any preferable text editor. The credentials are to establish connection to the database.
- By default, **\$servername** is “**localhost**” & **\$username** is “**root**”.

```

15 $servername = "localhost";
16
17 // REPLACE with your Database name
18 $dbname = "REPLACE_WITH_YOUR_DATABASE_NAME";
19
20 // REPLACE with table name
21 $tablename = "REPLACE_WITH_YOUR_TABLE_NAME";
22
23 // REPLACE with Database user
24 $username = "REPLACE_WITH_YOUR_USERNAME";
25
26 // REPLACE with Database user password
27 $password = "REPLACE_WITH_YOUR_PASSWORD";

```

→ Set up database with the following properties:  
 Database Name: **datacenter\_activities**  
 Table Name: **room\_dc230**



```

15 $servername = "localhost";
16
17 // REPLACE with your Database name
18 $dbname = "datacenters_activities";
19
20 // REPLACE with table name
21 $tablename = "room_dc230";
22
23 // REPLACE with Database user
24 $username = "root";
25
26 // REPLACE with Database user password
27 $password = "";

```

## Solution: Step 4 PHP SCRIPTS & HTDOCS & CODE EXPLANATION & TEST RUN

- Before getting access to MySQL database, the client must establish a connection to the server as shown in **Line#30**. If any of the credentials incorrect, the client don't have the permission to access the server. The script stops & will display error message at the browser. (MySQLi = MySQL improved)
- **Line#36** indicates selection of 5 columns from **room\_dc230** table with descending order of ***id***. The syntax is put into a variable called **\$sql**.

```
29 // Create connection
30 $conn = new mysqli($servername, $username, $password, $dbname);
31 // Check connection
32 if ($conn->connect_error) {
33     die("Connection failed: " . $conn->connect_error);
34 }
35
36 $sql = "SELECT logDateTime, rackID, tempValue, humidValue, rackStatus FROM $tablename ORDER BY id DESC";
37
```

## Solution: Step 4 PHP SCRIPTS & HTDOCS & CODE EXPLANATION & TEST RUN

→ A mix of PHP and HTML syntax in creating table.

```

38 echo '<table cellspacing="5" cellpadding="5">
39     <tr>
40         <td>Rack ID</td>
41         <td>Timestamp</td>
42         <td>Temparature (C)</td>
43         <td>Humidity (%)</td>
44         <td>Rack Status</td>
45     </tr>';

```

→ Line#47 – Line#69 is hosting asking, fetch data from database & display them in table manner.

→ "query()" is a function that fetch data that was assigned in \$sql variable. The results were put into a variable called \$results.

→ each data from "\$results" will be split into an array form according to its respected field.

```

47 if ($result = $conn->query($sql)) {
48     while ($row = $result->fetch_assoc()) {
49         $row_id = $row["rackID"];
50         $row_log = $row["logDateTime"];
51         $row_temp = $row["tempValue"];
52         $row_humid = $row["humidValue"];
53         $row_stats = $row["rackStatus"];

```

## Solution: Step 4 PHP SCRIPTS & HTDOCS & CODE EXPLANATION & TEST RUN

- Script to display the information fetched from database in table form. Each row represent 1 set data / reading. New row carries another set of reading/data.

```

60      echo '<tr>
61          <td>' . $row_id . '</td>
62          <td>' . $row_log . '</td>
63          <td>' . $row_temp . '</td>
64          <td>' . $row_humid . '</td>
65          <td>' . $row_stats . '</td>
66      '</tr>';
67  }

```

- The fetched data in **\$result** are emptied, thus the variable carries no information.
- "**close()**" is a function that closes previously open database connection.
- End of HTML script.

```

68      $result->free();
69  }
70
71 $conn->close();
72 ?>
73 </table>
74 </body>
75 </html>

```

## Solution: Step 4 PHP SCRIPTS & HTDOCS & CODE EXPLANATION & TEST RUN

- Open **esptodb.php** with **sublimetext** or any preferable text editor. Fill up with your database credentials.
- If **webview.php** is meant for client to view the temperature & humidity of the racks, **esptodb.php** is a script for EPS32 saving the racks data into the database.
- **\$api\_key\_value** is a kind of password to except information send from the **ESP32**. **ESP32** must send same key & the information will only be entertained when the key is matched. Otherwise, the information will be rejected.

```
28 // Keep this API Key value to be compatible with the ESP32 code provided in the project page.  
29 // If you change this value, the ESP32 sketch needs to match  
30 $api_key_value = "tPmAT5Ab3j7F9";  
31
```

- **Line#33** will empty the content of the handlers. The task of the handlers are to keep for information temporarily when received from the ESP32 thru HTTP POST or HTTP GET request.

```
32 // empty the variables  
33 $api_key= $rackID = $tempVal = $humidVal = $rackStat = "";  
34
```

## Solution: Step 4 PHP SCRIPTS & HTDOCS & CODE EXPLANATION & TEST RUN

- There are 2 types of Hyper Text Transfer Protocol (HTTP) request methods: **POST & GET**.
- What is HTTP? HTTP works as request-response protocol between a client & a server.
- **HTTP GET** is used to request data from a specified resource. It is often used to get values from APIs.
  - eg., ***http://localhost/a-my/esptodb.php?key=tPmAT5Ab3jF9&temp=10***
  - Name = **Key** & value = **tPmAT5Ab3jF9** + name = **temp** & value = **10** are send together in URL
  - With HTTP GET, data is **visible** to everyone
- **HTTP POST** is used to send information to a server to create/update a resource.
  - eg., publish sensor readings to a server
  - The data sent to the server with POST is stored in the request body of the HTTP request
  - ***String serverPath = serverName "?temperature=24.37"; // sample in sketch***
  - With HTTP POST, data is ***not visible*** in the URL request. However, if it's not encrypted, it's still visible in the request body.

## 2d. Centralized Temperature & Humidity Monitoring System for Server Racks

### Solution: Step 4 PHP SCRIPTS & HTDOCS & CODE EXPLANATION & TEST RUN

http://localhost/a-my/esptodbbase.php?api\_key=tPmAT5Ab3j7F9&rID=v112&temp=22&humid=40&currentStat=NORMAL

a

```
35 // DESC TEST FOR PRODUCTION
36
37 if ($_SERVER["REQUEST_METHOD"] == "GET") {           //GET or POST
38     //echo "test";
39     $api_key = test_input($_GET["api_key"]);          a
40     if($api_key == $api_key_value) {
41         $rackID  = test_input($_GET["rID"]);
42         $tempVal = test_input($_GET["temp"]);
43         $humidVal = test_input($_GET["humid"]);
44         $rackStat = test_input($_GET["currentStat"]);
45
46         // Create connection
47         $conn = new mysqli($servername, $username, $password, $dbname);
48         // Check connection
49         if ($conn->connect_error) {
50             die("Connection failed: " . $conn->connect_error);
51         }
52
53         $sql = "INSERT INTO tablename (rackID, tempValue, humidValue, rackStatus)
54             VALUES ('" . $rackID . "', '" . $tempVal . "', '" . $humidVal . "', '" . $rackStat . "')";
55
56         if ($conn->query($sql) === TRUE) {
57             echo "New record created successfully";
58         } else {
59             echo "Error: " . $sql . "<br>" . $conn->error;
60         }
61
62         $conn->close();
63     }
64     else {
65         echo "Wrong API Key provided.";
66     }
67
68 }
69 else {
70     echo "No data posted with HTTP POST.";
71 }
72
73
74 function test_input($data) {
75     $data = trim($data); //remove whitespace
76     $data = stripslashes($data); //removes backslashes
77     $data = htmlspecialchars($data); //to converts special characters
78     //& (ampersand), " (double quote), ' (single quote), < (less than), > (greater than)
79     //to HTML entities ( i.e. & (ampersand) becomes &amp;, ' (single quote) becomes &#039,
80     //< (less than) becomes &lt; (greater than) becomes &gt; ).
81     return $data;
82 }
```

## 2d. Centralized Temperature & Humidity Monitoring System for Server Racks

### Solution: Step 4 PHP SCRIPTS & HTDOCS & CODE EXPLANATION & TEST RUN

- It is now the time to test both PHP scripts. Make sure that the database credentials have set accordingly.
- No information print out since the database is still empty.

The screenshot shows a desktop environment with two windows open. The top window is a web browser displaying a table with columns: Rack ID, Timestamp, Temperature (C), Humidity (%), and Rack Status. The table is currently empty. The bottom window is the phpMyAdmin interface, showing the database structure and a query results page for the 'room\_dc230' table. The query `SELECT \* FROM `room\_dc230`;` returns zero rows. The MySQL status message indicates "MySQL returned an empty result set (i.e. zero rows). (Query took 0.0006 seconds.)".

## 2d. Centralized Temperature & Humidity Monitoring System for Server Racks

### Solution: Step 4 PHP SCRIPTS & HTDOCS & CODE EXPLANATION & TEST RUN

→ Open a browser, then copy & paste the following link & press enter.

[http://localhost/a-my/esptodb.php?api\\_key=tPmAT5Ab3j7F9&rID=v112&temp=22&humid=40&currentStat=NORMAL](http://localhost/a-my/esptodb.php?api_key=tPmAT5Ab3j7F9&rID=v112&temp=22&humid=40&currentStat=NORMAL)

New record created successfully

Rack ID	Timestamp	Temparature (C)	Humidity (%)	Rack Status
v112	2021-02-12 13:11:59	22	40	NORMAL
v112	2021-02-12 13:11:40	22	40	NORMAL
r1	2021-02-09 02:04:38	24	70	Normal
	2021-02-09 01:59:08			

Server: localhost » Database: datacenters\_activities » Table: room\_dc230

	id	logDateTime	rackID	tempValue	humidValue	rackStatus
1	1	2021-02-09 01:59:08				
2	2	2021-02-09 02:04:38	r1	24	70	Normal
4	4	2021-02-12 13:11:40	v112	22	40	NORMAL
5	5	2021-02-12 13:11:59	v112	22	40	NORMAL

→ Try this:

1- <http://localhost/a-my/>

2- [http://localhost/a-my/esptodb.php?api\\_key=tPmAT5Ab3j7F9&rID=v111&temp=22.4&humid=40%&currentStat="NORMAL"](http://localhost/a-my/esptodb.php?api_key=tPmAT5Ab3j7F9&rID=v111&temp=22.4&humid=40%&currentStat='NORMAL')

## Solution: Step 5 ESP32 POST INFORMATION TO DBASE: SKETCH WALKTHROUGH

- Open **sketch-vii\_esp32-dht11-http-post.ino** from <http://bit.ly/3p2pmNd> downloaded in Step 4. Do not upload the sketch since you need to make some changes.
- This will inform the compiler to compile the correct library no matter if what board you are currently uploading into. In this case, either ESP32 or NodeMCU & ESP8266.

```
13 #ifdef ESP32
14 // this will compile for ESP32 board
15 #include <WiFi.h>
16 #include <HTTPClient.h>
17 #else
18 // this will compile for ESP8266, NodeMCU boards
19 #include <ESP8266WiFi.h>
20 #include <ESP8266HTTPClient.h>
21 #include <WiFiClient.h>
22#endif
```

## Solution: Step 5 ESP32 POST INFORMATION TO DBASE: SKETCH WALKTHROUGH

→ Set the followings to the WiFi network connection.

```
24 // Replace with your network credentials  
25 const char* ssid      = "REPLACE_WITH_YOUR_SSID";  
26 const char* password = "REPLACE_WITH_YOUR_PASSWORD";
```

→ Change the “192.168.1.102” with your server’s IP address. How to find IP address of your system ☺?

→ The file path “/a-my/esptodb.php” reflects the “drive/xampp/htdocs/” & set in STEP xx.

```
28 // REPLACE with your Domain name and URL path or server's IP address with path  
29 const char* serverName = "http://192.168.1.102/a-my/esptodb.php";
```

→ Both ESP32 has the same API key with esptodb.php to avoid fake request. If not match, the information will not save into database. The API key is manually generated by the developer/user.

```
31 // Keep this API Key value to be compatible with the PHP code provided in the project page.  
32 // If you change the apiKeyValue value, the PHP file "/esptodb.php" also needs to have the same key  
33 String apiKeyValue = "tPmAT5Ab3j7F9";
```

→ Set the sensorID accordingly – e.g. rackID. This ID must be different from one another & must be documented, easier to detect & locate.

```
35 String sensorID = "a1";
```

## Solution: Step 5 ESP32 POST INFORMATION TO DBASE: SKETCH WALKTHROUGH

→ The board will connect to the WiFi router. If any of the SSID or Password is incorrect, the board will not join the network. Open Serial Monitor to view the process. IP address will be published if the board connected to the router.

```
40 void setup() {  
41   Serial.begin(115200);  
42  
43   WiFi.begin(ssid, password);  
44   Serial.println("Connecting");  
45   while(WiFi.status() != WL_CONNECTED) {  
46     delay(500);  
47     Serial.print(".");  
48   }  
49   Serial.println("");  
50   Serial.print("Connected to WiFi network with IP Address: ");  
51   Serial.println(WiFi.localIP());  
52 }
```

→ Fail to connect to router.

```
Connecting  
.....
```

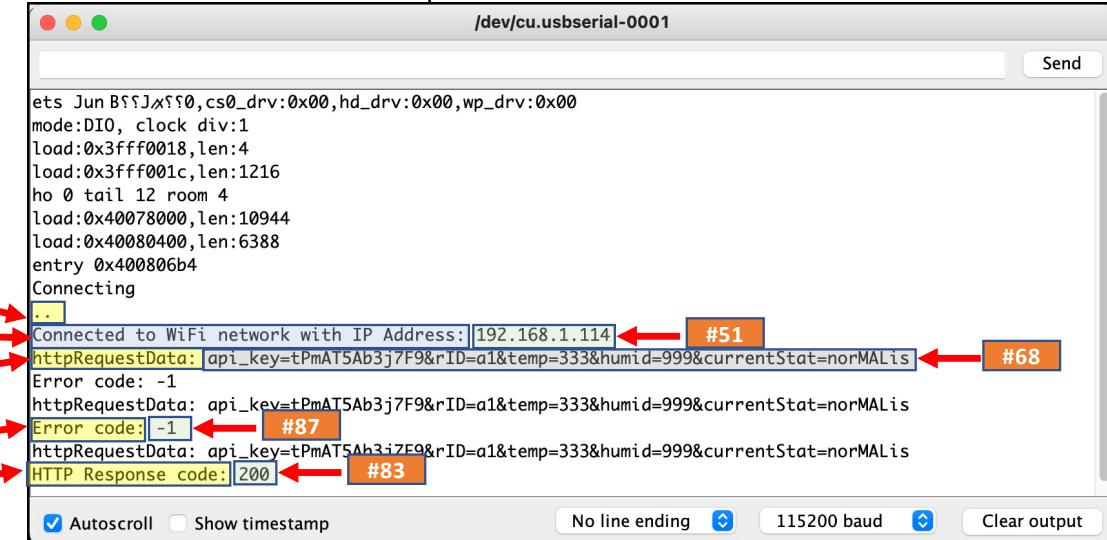
## 2d. Centralized Temperature & Humidity Monitoring System for Server Racks

### Solution: Step 5 ESP32 POST INFORMATION TO DBASE: SKETCH WALKTHROUGH

```

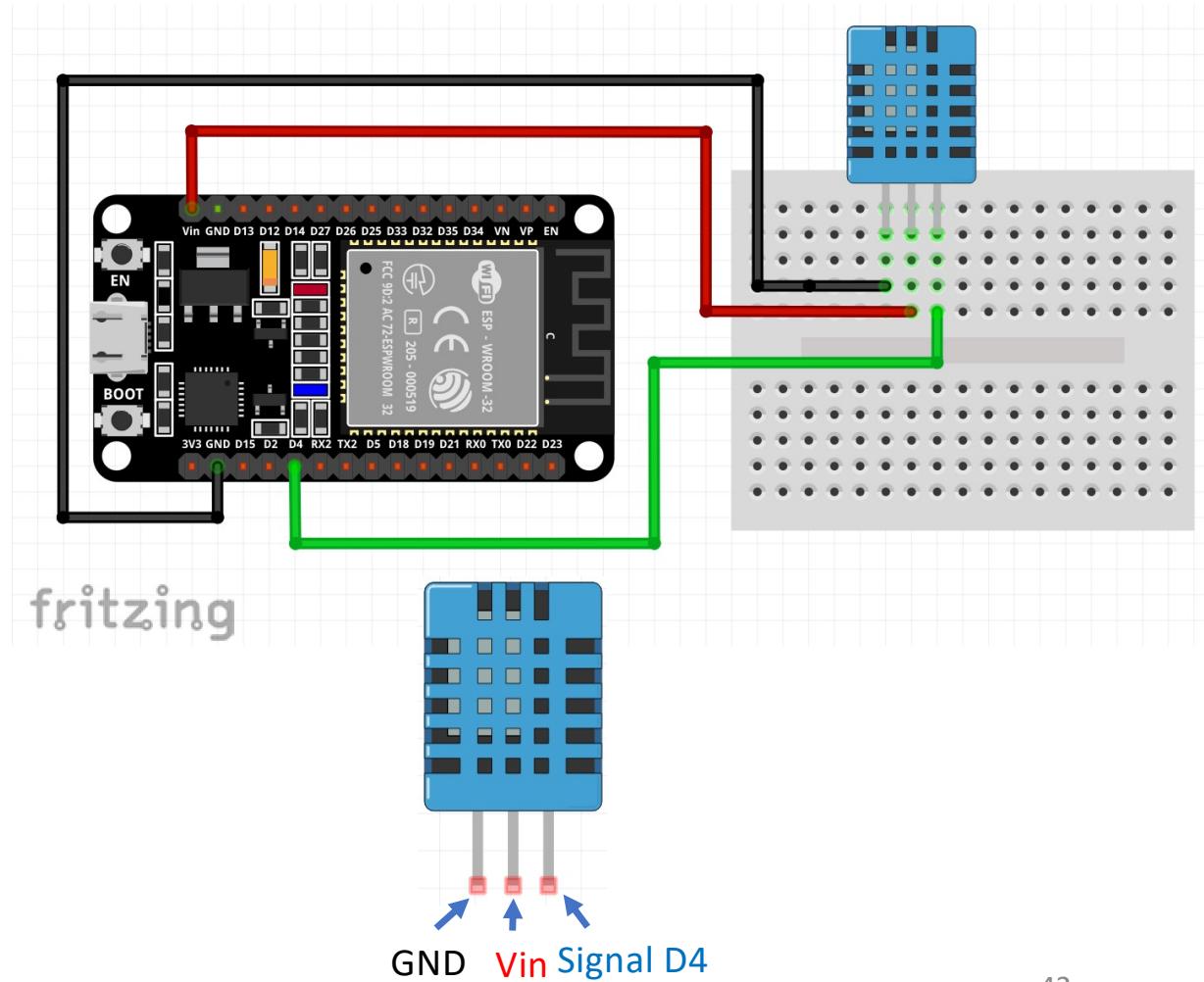
54 void loop() {
55 //Check WiFi connection status
56 if(WiFi.status()== WL_CONNECTED){
57   HTTPClient http;
58
59   // Your Domain name with URL path or IP address with path
60   http.begin(serverName);
61
62   // Specify content-type header
63   http.addHeader("Content-Type", "application/x-www-form-urlencoded");
64
65   // Prepare your HTTP POST request data
66   String httpRequestData = "api_key=" + apiKeyValue + "&rID=" + sensorID + "&temp=" + 333 + "&humid=" + 999 + "&currentStat=" + "norMALis";
67   Serial.print("httpRequestData: ");
68   Serial.println(httpRequestData);
69
70   // You can comment the httpRequestData variable above
71   // then, use the httpRequestData variable below (for testing purposes without the DHT11 sensor) -->Similar with Line#66
72   //String httpRequestData = "api_key=tPmAT5Ab3j7F9&rID=abc&temp=123&humid=24.75&currentStat=norMALis";
73
74   // Send HTTP POST request
75   int httpResponseCode = http.POST(httpRequestData);
76
77   // If you need an HTTP request with a content type: text/plain
78   //http.addHeader("Content-Type", "text/plain");
79   //int httpResponseCode = http.POST("Hello, World!");
80
81   if (httpResponseCode>0) {
82     Serial.print("HTTP Response code: ");
83     Serial.println(httpResponseCode);
84   }
85   else {
86     Serial.print("Error code: ");
87     Serial.println(httpResponseCode);
88   }
89   // Free resources
90   http.end();
91
92   else {
93     Serial.println("WiFi Disconnected");
94   }
95   //Send an HTTP POST request every 30 seconds
96   delay(30000);
97 }
```

- #66: Test send data to esptodb.php (IP add at #29)
- #81: Success to communicate with server
- #85: Unable to communicate with server --> not in the same network
- #96: Update every 30 seconds



## Solution: Step 6 SCHEMATIC DIAGRAM

- After success tested the sketch, its time to combine the DHT part with ESP32.
- Before we go any further, construct the circuit as in the figure.
- Check a few times, the connections that have been made before plugging ESP32 to USB port. This might prevent short circuit to DHT11. Watch out on sensors' pin. Same type, doesn't mean same pin configurations.



## Solution: Step 7 THE WORKING SKETCH

→ Open **sketch-vii\_esp32-dht11-http-post.ino** from <http://bit.ly/3p2pmNd> downloaded in Step 4. Do not upload the sketch since you need to make some changes.

```

13 #ifdef ESP32
14 // this will compile for ESP32 board
15 #include <WiFi.h>
16 #include <HTTPClient.h>
17 #else
18 // this will compile for ESP8266, NodeMCU boards
19 #include <ESP8266WiFi.h>
20 #include <ESP8266HTTPClient.h>
21 #include <WiFiClient.h>
22 #endif
23
24 #include <dht11.h> ←
25 dht11 DHT;
26 #define DHT11_PIN 4
27
28 // Replace with your network credentials
29 const char* ssid      = "air24";//REPLACE_WITH_YOUR_SSID";
30 const char* password = "polis12345";//REPLACE_WITH_YOUR_PASSWORD";
31
32 // REPLACE with your Domain name and URL path or server's IP address with path
33 const char* serverName = "http://192.168.1.102/a-my/esptodb.php";
34
35 // Keep this API Key value to be compatible with the PHP code provided in the project page.
36 // If you change the apiKeyValue value, the PHP file "/esptodb.php" also needs to have the same key
37 String apiKeyValue = "tPmAT5Ab3j7F9";
38
39 String sensorID = "a1";
40 //// latihan
41 ////String sensorLocation = "Office";

```

## Solution: Step 7 THE WORKING SKETCH

```
44 void setup() {  
45   Serial.begin(115200);  
46   WiFi.begin(ssid, password);  
47   Serial.println("Connecting");  
48   while(WiFi.status() != WL_CONNECTED) {  
49     delay(500);  
50     Serial.print(".");  
51   }  
52   Serial.println("");  
53   Serial.print("Connected to WiFi network with IP Address: ");  
54   Serial.println(WiFi.localIP());  
55 }
```

## 2d. Centralized Temperature & Humidity Monitoring System for Server Racks

### Solution: Step 7 THE WORKING SKETCH

```
57 void loop() {  
58     int chk = DHT.read(DHT11_PIN);    //Read humid & temp ←  
59     //Check WiFi connection status  
60     if(WiFi.status()== WL_CONNECTED){  
61         HTTPClient http;  
62  
63         // Your Domain name with URL path or IP address with path  
64         http.begin(serverName);  
65  
66         // Specify content-type header  
67         http.addHeader("Content-Type", "application/x-www-form-urlencoded");  
68  
69         // Prepare your HTTP POST request data  
70         String httpRequestData = "api_key=" + apiKeyValue + "&rID=" + sensorID  
71             + "&temp=" + DHT.temperature + "&humid=" + DHT.humidity ←  
72             + "&currentStat=" + norMALis;  
73         Serial.print("httpRequestData: ");  
74         Serial.println(httpRequestData);  
75  
76         // You can comment the httpRequestData variable above  
77         // then, use the httpRequestData variable below (for testing purposes without the DHT11 sensor) -->Similar with Line#66  
78         //String httpRequestData = "api_key=tPmAT5Ab3j7F9&rID=abc&temp=123&humid=24.75&currentStat=norMALis";  
79  
80         // Send HTTP POST request  
81         int httpResponseCode = http.POST(httpRequestData);  
82  
83         // If you need an HTTP request with a content type: text/plain  
84         //http.addHeader("Content-Type", "text/plain");  
85         //int httpResponseCode = http.POST("Hello, World!");  
86         if (httpResponseCode>0) {  
87             Serial.print("HTTP Response code: ");  
88             Serial.println(httpResponseCode);  
89         }  
90         else {  
91             Serial.print("Error code: ");  
92             Serial.println(httpResponseCode);  
93         }  
94         // Free resources  
95         http.end();  
96     }  
97     else {  
98         Serial.println("WiFi Disconnected");  
99     }  
100    //Send an HTTP POST request every 30 seconds  
101    delay(30000);  
102 }
```

## 2d. Centralized Temperature & Humidity Monitoring System for Server Racks

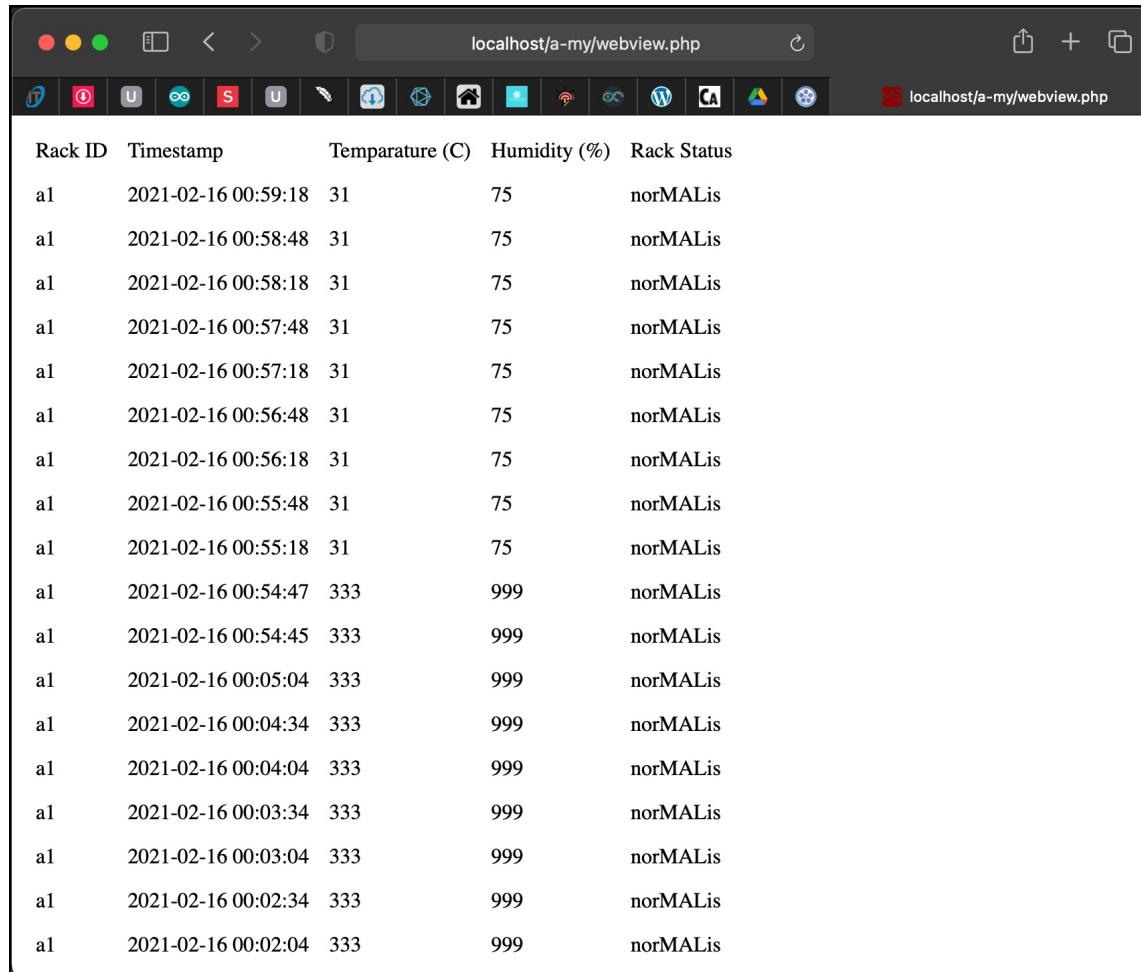
### Solution: Step 7 THE WORKING SKETCH

The image displays three windows illustrating the working sketch of a centralized monitoring system:

- Terminal Log:** Shows a series of log entries from a USB serial port. The entries indicate requests to an API endpoint with parameters like `api_key=tPmAT5Ab3j7F9&rID=a1&temp=31&humid=75&currentStat=normal` and responses with code 200.
- MySQL Database:** A screenshot of a MySQL database table showing log entries. The columns are `+ Options`, `id`, `logDateTime`, `rackID`, `tempValue`, `humidValue`, and `rackStatus`. The data shows multiple entries for rack ID `a1` at various timestamps, with `tempValue` and `humidValue` both set to 333, and `rackStatus` consistently showing "norMALis".
- Web Browser:** A screenshot of a web browser window titled "localhost/a-my/webview.php". It displays a table with columns: `Rack ID`, `Timestamp`, `Temparature (C)`, `Humidity (%)`, and `Rack Status`. The data is identical to the MySQL table, showing values for rack ID `a1` over time.

## 2d. Centralized Temperature & Humidity Monitoring System for Server Racks

### Solution: Step 7 THE WORKING SKETCH



A screenshot of a web browser window titled "localhost/a-my/webview.php". The browser has a dark theme with a toolbar at the top featuring various icons. The main content area displays a table with the following data:

Rack ID	Timestamp	Temparature (C)	Humidity (%)	Rack Status
a1	2021-02-16 00:59:18	31	75	norMALis
a1	2021-02-16 00:58:48	31	75	norMALis
a1	2021-02-16 00:58:18	31	75	norMALis
a1	2021-02-16 00:57:48	31	75	norMALis
a1	2021-02-16 00:57:18	31	75	norMALis
a1	2021-02-16 00:56:48	31	75	norMALis
a1	2021-02-16 00:56:18	31	75	norMALis
a1	2021-02-16 00:55:48	31	75	norMALis
a1	2021-02-16 00:55:18	31	75	norMALis
a1	2021-02-16 00:54:47	333	999	norMALis
a1	2021-02-16 00:54:45	333	999	norMALis
a1	2021-02-16 00:05:04	333	999	norMALis
a1	2021-02-16 00:04:34	333	999	norMALis
a1	2021-02-16 00:04:04	333	999	norMALis
a1	2021-02-16 00:03:34	333	999	norMALis
a1	2021-02-16 00:03:04	333	999	norMALis
a1	2021-02-16 00:02:34	333	999	norMALis
a1	2021-02-16 00:02:04	333	999	norMALis

## **QUESTIONS?**

**Questions?**

## EXERCISE

Modify this project by doing the followings:

- a. Add LED statement in ESP32 sketch – if connected to WiFi, built in led (GPIO2) will switch on.
- b. Write condition statements at PHP files:
  - i. If temp > 30°C status is ALERT
  - ii. If temp > 40°C status is DANGER

The status need to be saved in the database together with other information.

HINT: you need to omit the status information from ESP32 sketch.