In this project, we will send the temperature (using a temperature sensor LM35) to a MySQL table.

Hardware requirements:

1. NodeMCU
2. Temperature sensor (LM35)
3. Prototype breadboard
4. Jumper Wires
5. Micro USB Cable (not provided please use your phone charger cable with data transfer capability)

Software requirements:

1. Xampp
2. Arduino IDE. In this case I am using Arduino 1.8.8 (Windows Installer). The installer can be downloaded through https://www.arduino.cc/en/Main/OldSoftwareReleases#previous)

Code Downloads:

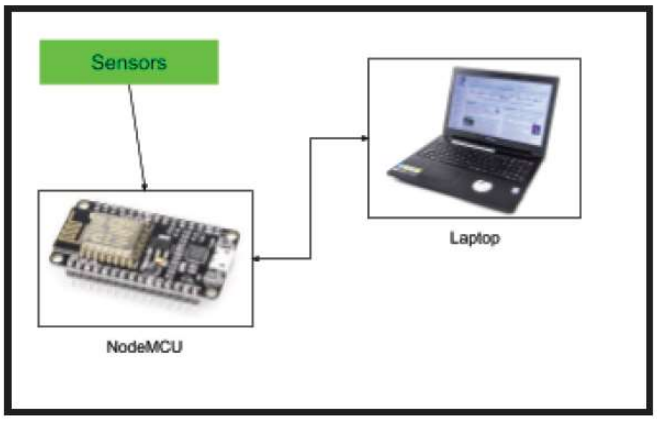
Go to: https://github.com/anon-drfd/temp_to_db_esp8266 and download all the codes from the repository

1. connNodeMCUdb.ino
2. dbInsert.php

Step 1: Setup Arduino IDE to be compatible with NodeMCU

We will be using Arduino IDE to send the codes to NodeMCU through the micro USB cable (which wil also be reposible to power up the NodeMCU). Once the codes are uploaded to the NodeMCU, you can remove the micro USB cable and connect through independent power source such as a power bank since the codes are now stored in the NodeMCU memory and thus, be run independently without using PC.

If there is changes in codes, you need to repeat the process by sending codes from Arduino IDE in PC to the board using micro USB cable. In this example, I will use the same laptop to upload the codes to the board and as my server to run my MySQL database.



Once you finished the installation process of Arduino IDE, we need to setup NodeMCU compatibility in Arduino IDE.

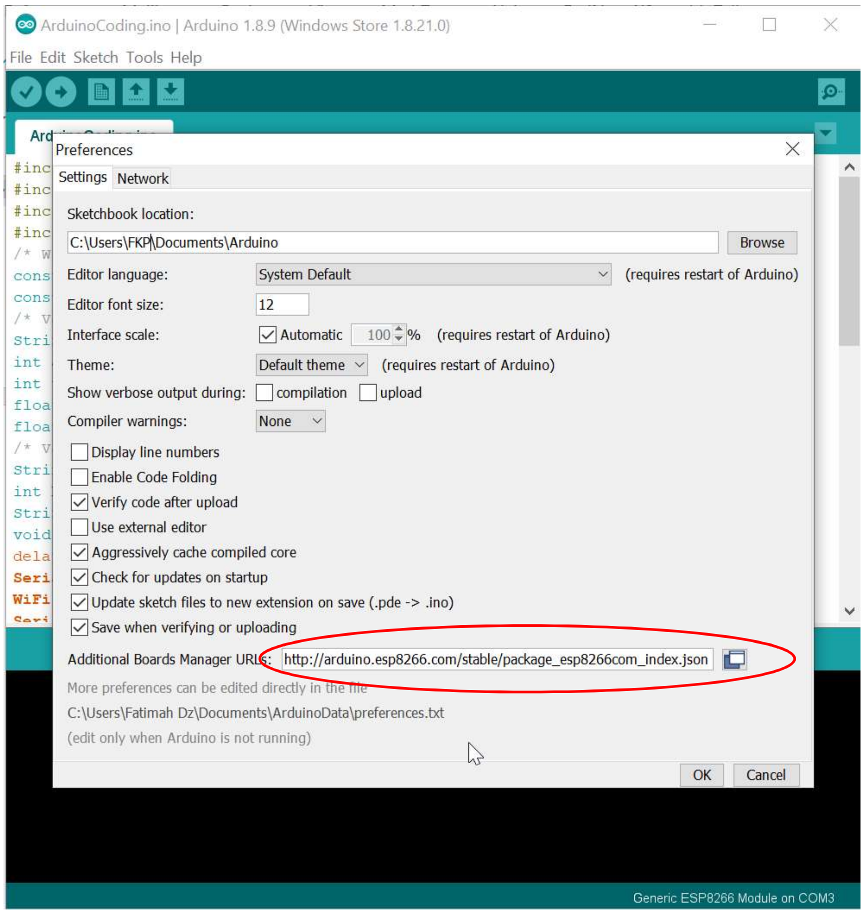
USB driver installation:

1. Go to the link for driver installation: https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers

   Pick the right on for your Operating System. In this project we used "Download for Windows 10 Universal (v10.1.7)"

ESP8266 library Installation:

1. In Arduino IDE, Go to Files -> Preference. Copy the code in the Additional boards Manager: (http://arduino.esp8266.com/stable/package_esp8266com_index.json)


ArduinoCoding.ino | Arduino 1.8.9 (Windows Store 1.8.21.0)
File Edit Sketch Tools Help
Preferences
Settings Network
Browse
OK Cancel
Generic ESP8266 Module on COM3

Sketchbook location: C:\Users\FKP\Documents\Arduino

Editor language: System Default (requires restart of Arduino)

Editor font size: 12

Interface scale: ✓ Automatic 100 % (requires restart of Arduino)

Theme: Default theme (requires restart of Arduino)

Show verbose output during: ☐ compilation ☐ upload

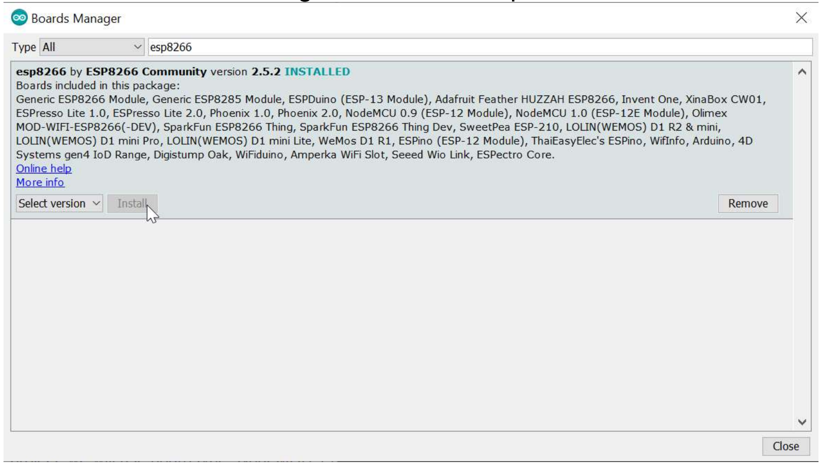
Compiler warnings: None

☐ Display line numbers
☐ Enable Code Folding
✓ Verify code after upload
☐ Use external editor
✓ Aggressively cache compiled core
✓ Check for updates on startup
✓ Update sketch files to new extension on save (.pde -> .ino)
✓ Save when verifying or uploading

Additional Boards Manager URLs: http://arduino.esp8266.com/stable/package_esp8266com_index.json

More preferences can be edited directly in the file
C:\Users\Fatimah Dz\Documents\ArduinoData\preferences.txt
(edit only when Arduino is not running)

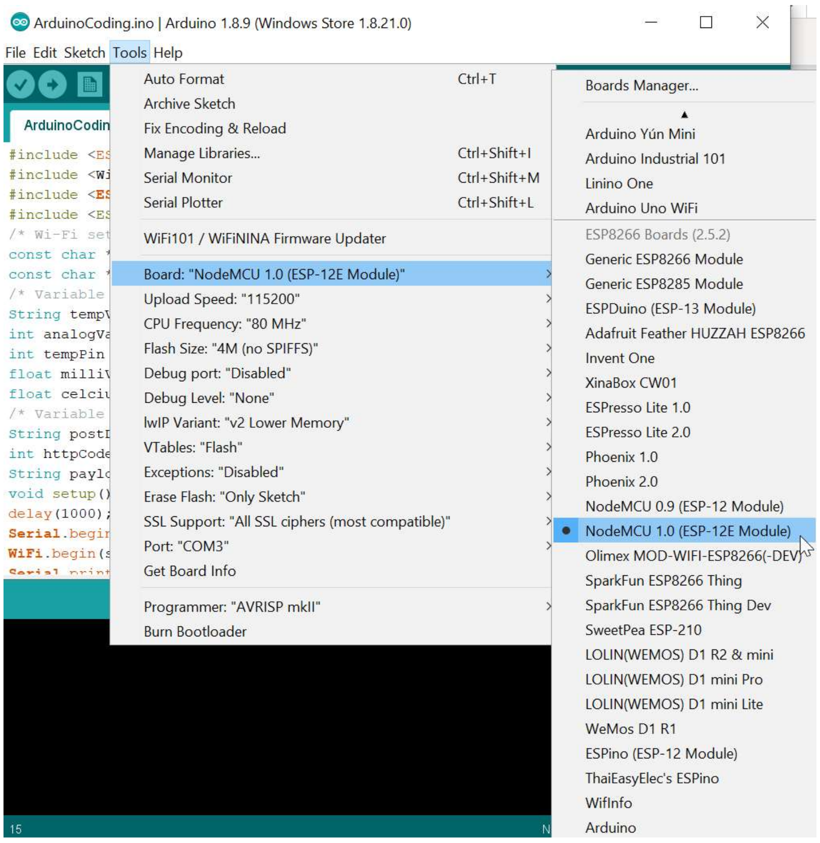
Then go to Tools -> board -> board Manager. Search for esp8266 and install the software for Arduino.



Type All esp8266

esp8266 by **ESP8266 Community** version **2.5.2** INSTALLED
Boards included in this package:
Generic ESP8266 Module, Generic ESP8285 Module, ESPDuino (ESP-13 Module), Adafruit Feather HUZZAH ESP8266, Invent One, XinaBox CW01, ESPresso Lite 1.0, ESPresso Lite 2.0, Phoenix 1.0, Phoenix 2.0, NodeMCU 0.9 (ESP-12 Module), NodeMCU 1.0 (ESP-12E Module), Olimex MOD-WIFI-ESP8266(-DEV), SparkFun ESP8266 Thing, SparkFun ESP8266 Thing Dev, SweetPea ESP-210, LOLIN(WEMOS) D1 R2 & mini, LOLIN(WEMOS) D1 mini Pro, LOLIN(WEMOS) D1 mini Lite, WeMos D1 R1, ESPino (ESP-12 Module), ThaiEasyElec's ESPino, WifInfo, Arduino, 4D Systems gen4 IoD Range, Digistump Oak, WiFiduino, Amperka WiFi Slot, Seeed Wio Link, ESPectro Core.
Online help
More info

After installation, you should see that your Arduino IDE has NodeMCU 0.9 and NodeMCU 1.0 in board selection. For this project, we will use board type "NodeMCU 1.0".

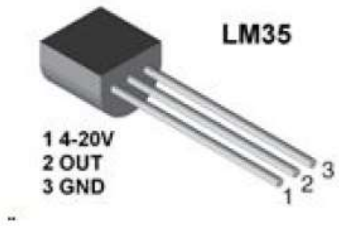


```
Board: "NodeMCU 1.0 (ESP-12E Module)"
Upload Speed: "115200"
CPU Frequency: "80 MHz"
Flash Size: "4M (no SPIFFS)"
Debug port: "Disabled"
Debug Level: "None"
lwIP Variant: "v2 Lower Memory"
VTables: "Flash"
Exceptions: "Disabled"
Erase Flash: "Only Sketch"
SSL Support: "All SSL ciphers (most compatible)"
Port: "COM3"
Programmer: "AVRISP mkII"
```

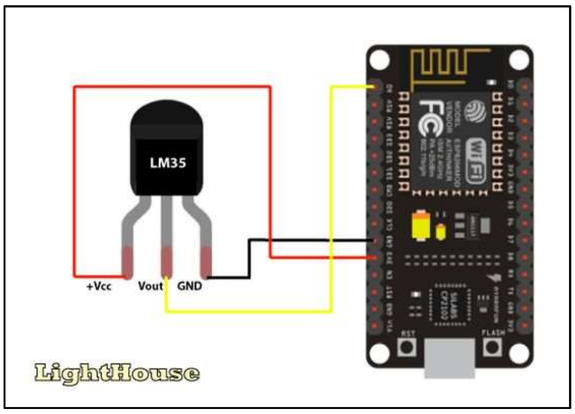
Step 2: Setup LM35 electrical connections

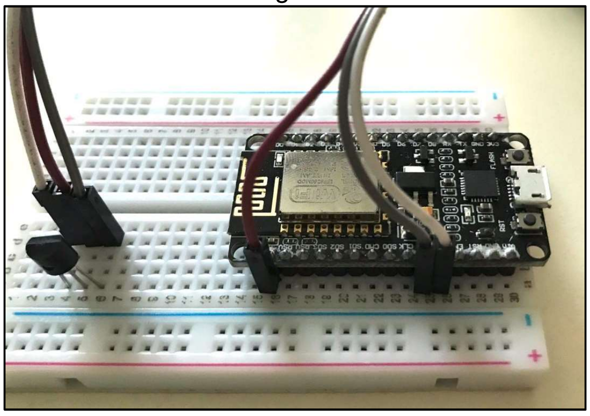
The circuit connections are setup as follows (see below figure better understanding):

1. Pin 1 of the LM35 goes into +3.3v of the NodeMCU.
2. Pin 2 of the LM35 goes into Analog Pin A0 of the NodeMCU.
3. Pin 3 of the LM35 goes into Ground Pin (GND) of the NodeMCU



Create electrical connection of NodeMCU and LM35 as illustrated in the figure below:





Step 3: Setup the database

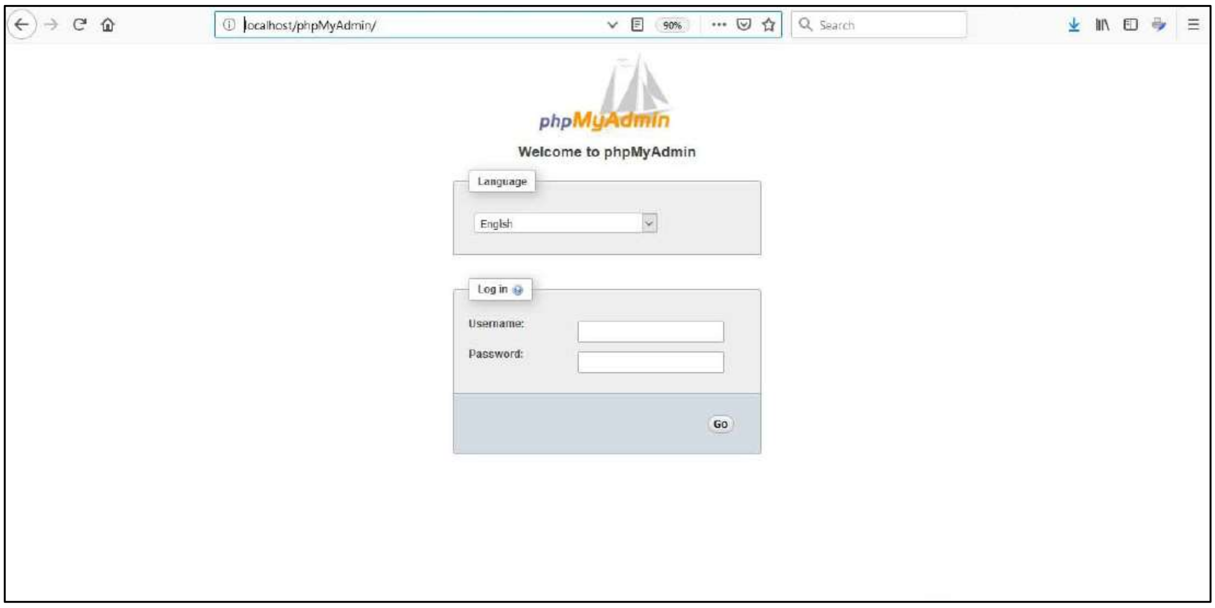
Before we start coding the NodeMCU, we need to setup the database first. We need to prepare the place for data placing in MySQL. We will be using Xampp that already has Apache, PHP, MySQL.

- Apache is responsible to make your computer behave like server
- PHP is allows your computer to read PHP scripts
- MySQL is enable your computer to have a database management system.

Before you open phpMyAdmin interface, you need to run apache server on your PC. Make sure you open Xampp and switch these settings on.

Once Apache is running, open your internet browser (Firefox, chrome, IE, etc.) and type http://localhost/phpMyAdmin/

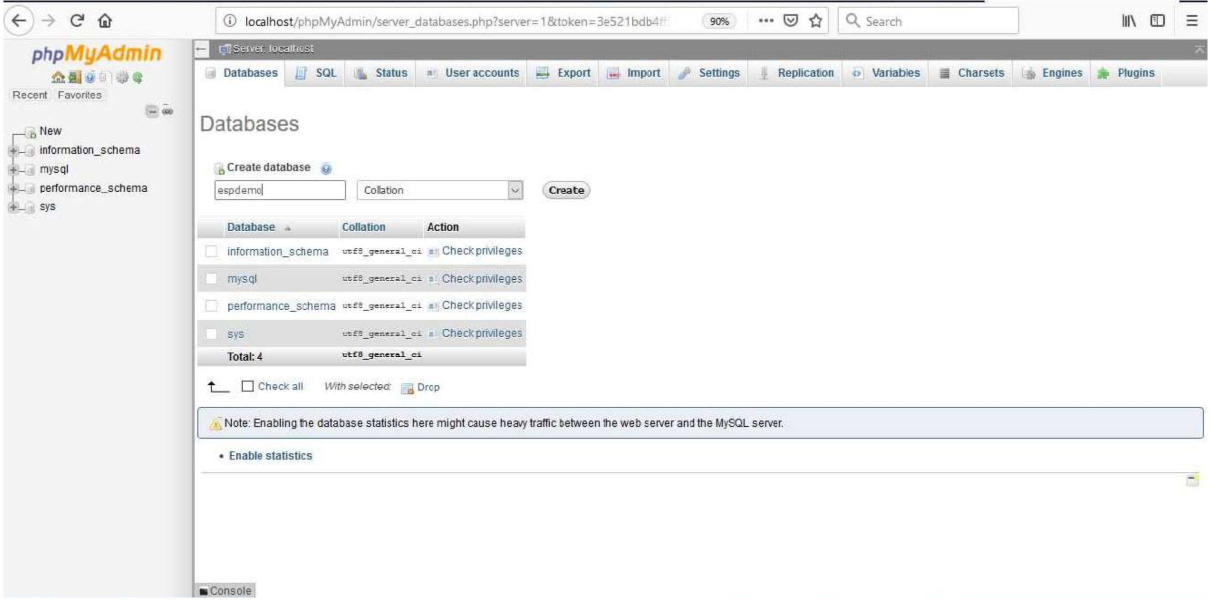
Make sure you remember the username and password for this login. In my case, username is "root" and password is "12345678".



After login, create a new database by clicking "New" at your left panel.

- Database name: espdemo

Click "Create" button and you will see your "espdemo" database is added at your left panel.


localhost/phpMyAdmin/server_databases.php?server=1&token=3e521bdb4f
phpMyAdmin
Recent Favorites
New
information_schema
mysql
performance_schema
sys
Server: localhost
Databases SQL Status User accounts Export Import Settings Replication Variables Charsets Engines Plugins
Create
Console

Databases

Create database

espdemo | Collation

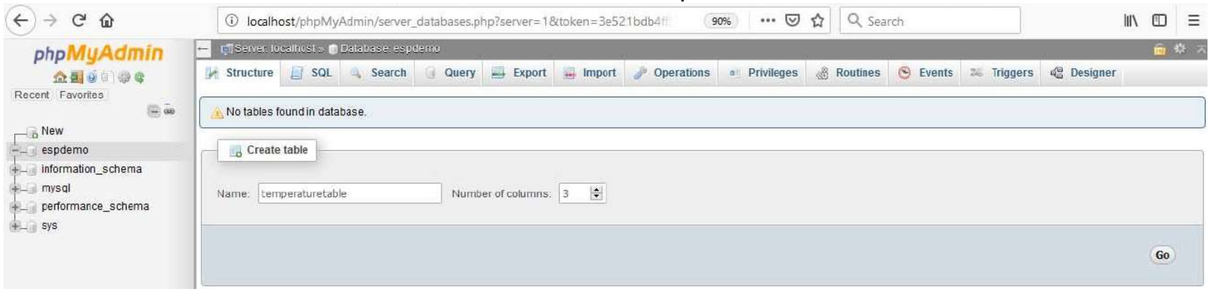
Database	Collation	Action
information_schema	utf8_general_ci	Check privileges
mysql	utf8_general_ci	Check privileges
performance_schema	utf8_general_ci	Check privileges
sys	utf8_general_ci	Check privileges
Total: 4	utf8_general_ci	

Check all With selected: Drop

Note: Enabling the database statistics here might cause heavy traffic between the web server and the MySQL server.

- Enable statistics

The next step is to create a table. Here, I want to 3 values: date, time, and temperature (celcius). So we need to have 3 columns. We will name this table "temperaturetable".


localhost/phpMyAdmin/server_databases.php?server=1&token=3e521bdb4f
phpMyAdmin
Recent Favorites
New
espdemo
information_schema
mysql
performance_schema
sys
Server: localhost » Database: espdemo
Structure SQL Search Query Export Import Operations Privileges Routines Events Triggers Designer
Create table
Go

No tables found in database.

Name: temperaturetable Number of columns: 3

Click "Go" button and you will redirected to the column setup. Setup all those columns with type and lengths/values which as follows:

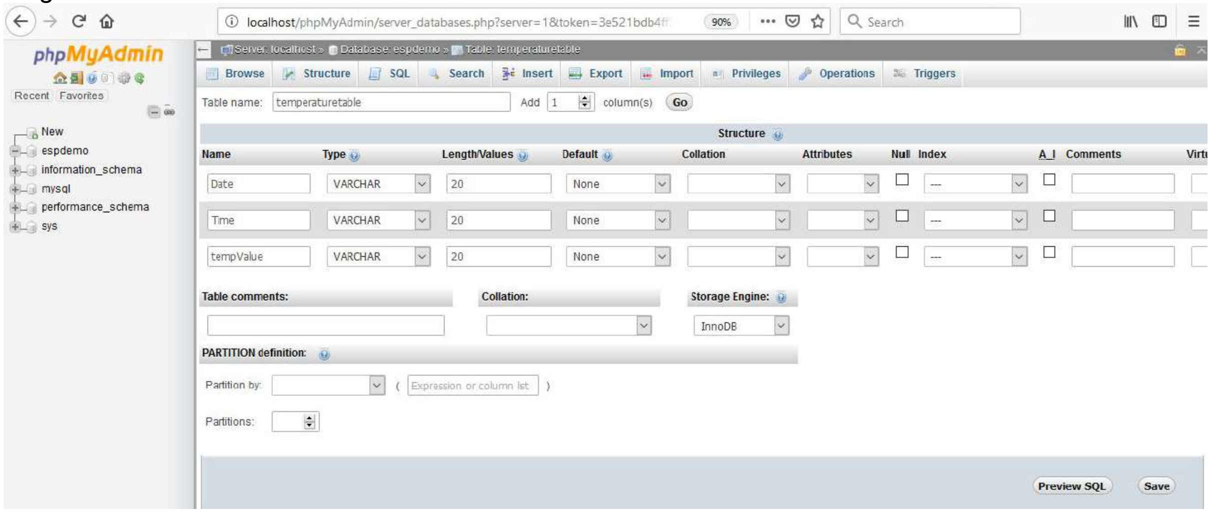
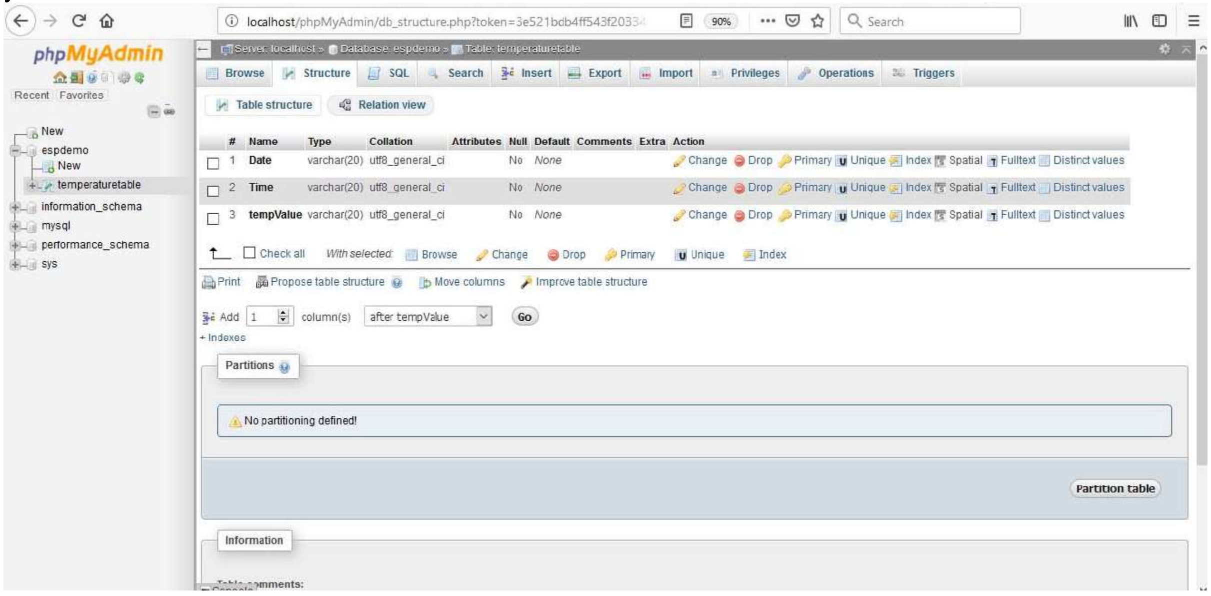


Table name: temperaturetable

Name	Type	Length/Values	Default
Date	VARCHAR	20	None
Time	VARCHAR	20	None
tempValue	VARCHAR	20	None

Storage Engine: InnoDB

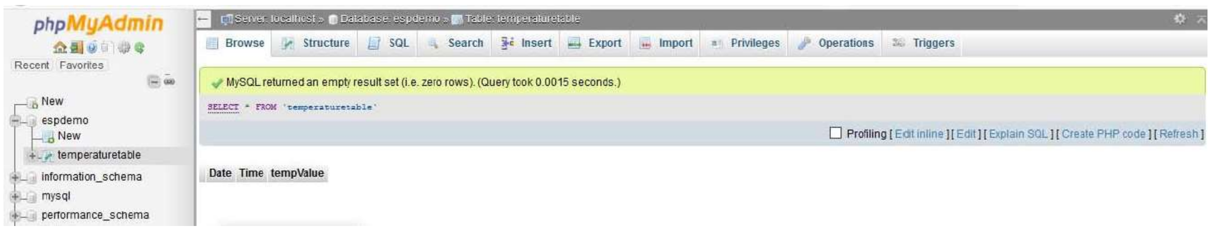
Click the "Save" button on the bottom right of the screen and you will see your table is created. Now you have established the database.



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	Date	varchar(20)	utf8_general_ci		No	None		
2	Time	varchar(20)	utf8_general_ci		No	None		
3	tempValue	varchar(20)	utf8_general_ci		No	None		

No partitioning defined!

Click the Browse tab to visualise your table. Later, when data is sent from your NodeMCU, you will see the data being updated here.



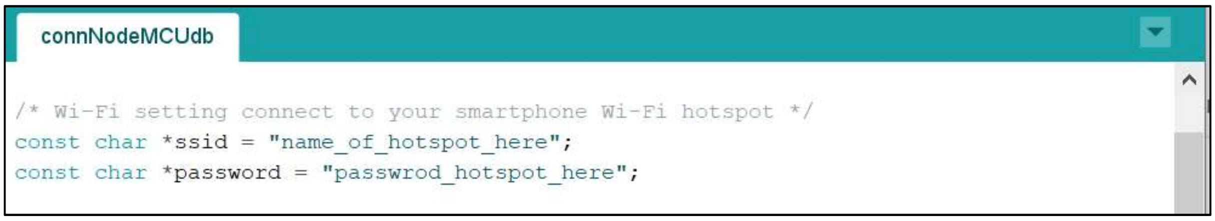
```
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0015 seconds.)
SELECT * FROM `temperaturetable`
```

Date	Time	tempValue

Step 4: Prepare code for NodeMCU

Copy the code in connNodeMCUdb.ino that you downloaded from https://github.com/anon-drfd/temp_to_db_esp8266 into Arduino IDE.

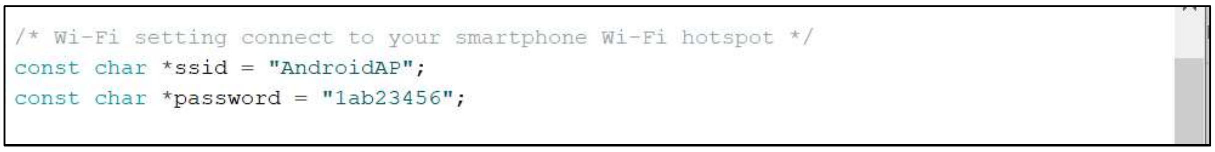
Edit the details to your hotspot:



connNodeMCUdb

```
/* Wi-Fi setting connect to your smartphone Wi-Fi hotspot */
const char *ssid = "name_of_hotspot_here";
const char *password = "passwrod_hotspot_here";
```

For example, if my hotspot name is AndroidAP and the password is 1ab23456:



```
/* Wi-Fi setting connect to your smartphone Wi-Fi hotspot */
const char *ssid = "AndroidAP";
const char *password = "1ab23456";
```

Notice that in connNodeMCUdb.ino, there is a line of code:

```
http.begin(" http://192.168.43.71/dbInsert.php");
```

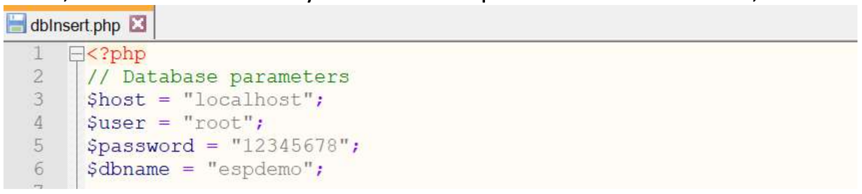
we will be setting this file up in the next step.

Step 5: Setup dbInsert.php file in server computer

Save the dbInsert.php you downloaded from https://github.com/anon-drfd/temp_to_db_esp8266 in the Xampp directory.

This dbInsert.php file is created in order to hold temporary data sending from NodeMCU and store this it in the MySQL database.

Then, edit the details for your database parameters: hostname, username and password.



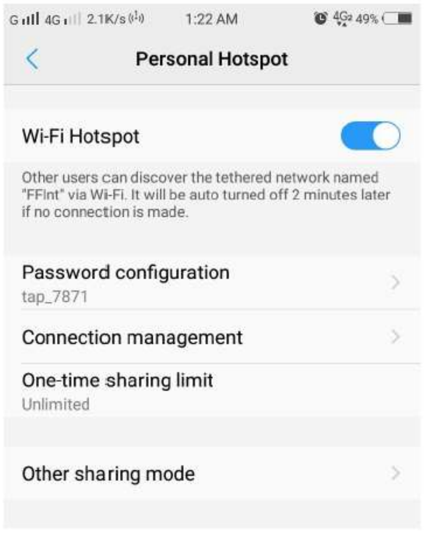
```php
<?php
// Database parameters
$host = "localhost";
$user = "root";
$password = "12345678";
$dbname = "espdemo";
```

Step 6: Enabling Wi-Fi hotspots in your smartphone

1. Turn on Wifi Hotspot on your smartphone.
2. Connect your laptop (server) to this Wifi Hotspot.

Your smartphone now become as a router/switch.

The following figure is an example of a successful hotspot connection:



Personal Hotspot

Wi-Fi Hotspot

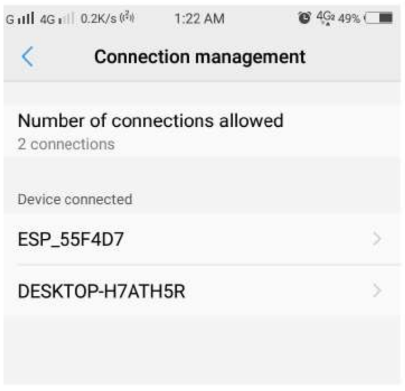
Other users can discover the tethered network named "FFInt" via Wi-Fi. It will be auto turned off 2 minutes later if no connection is made.

Password configuration
tap_7871

Connection management

One-time sharing limit
Unlimited

Other sharing mode



Connection management

Number of connections allowed
2 connections

Device connected

ESP_55F4D7

DESKTOP-H7ATH5R

You can view the IP given to your PC by clicking the device connected.

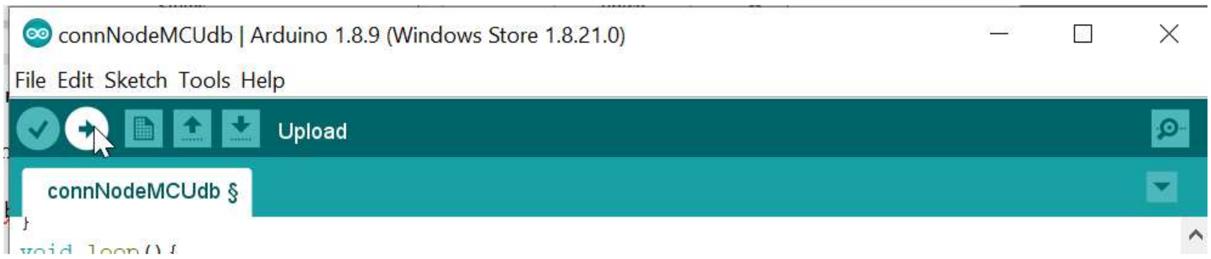
In my case, since I have found that my server IP address is 192.168.43.71, I now need to update connNodeMCUdb.ino, so that the line of code includes this IP address:

```
http.begin(" http://192.168.43.71/dbInsert.php");
```

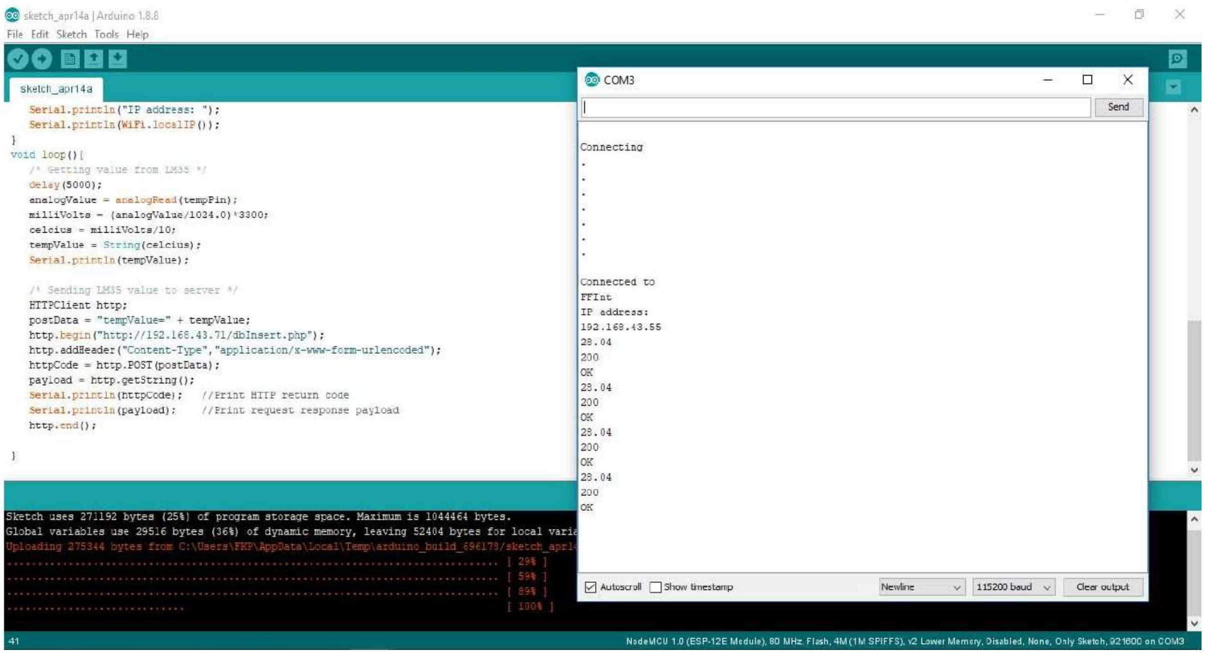
Find your IP address and update the line of code accordingly in connNodeMCUdb.ino. For example, if your IP address is 172.20.10.5, your code will now look like this:

```
/* Sending LM35 value to server */
HTTPClient http;
postData = "tempValue=" + tempValue;
http.begin("http://172.20.10.5/dbInsert.php");
```

After that, upload the connNodeMCUdb.ino to your nodeMCU:


connNodeMCUdb | Arduino 1.8.9 (Windows Store 1.8.21.0)
File Edit Sketch Tools Help
Upload
connNodeMCUdb §

You should get the following serial monitor output (Tools -> Serial Monitor):



```
    Serial.println("IP address: ");
    Serial.println(WiFi.localIP());
}
void loop(){
    /* Getting value from LM35 */
    delay(5000);
    analogValue = analogRead(tempPin);
    milliVolts = (analogValue/1024.0)*3300;
    celcius = milliVolts/10;
    tempValue = String(celcius);
    Serial.println(tempValue);

    /* Sending LM35 value to server */
    HTTPClient http;
    postData = "tempValue=" + tempValue;
    http.begin("http://192.168.43.71/dbInsert.php");
    http.addHeader("Content-Type","application/x-www-form-urlencoded");
    httpCode = http.POST(postData);
    payload = http.getString();
    Serial.println(httpCode);    //Print HTTP return code
    Serial.println(payload);     //Print request response payload
    http.end();

}
```

```
Sketch uses 271192 bytes (25%) of program storage space. Maximum is 1044464 bytes.
Global variables use 29516 bytes (36%) of dynamic memory, leaving 52404 bytes for local varia
Uploading 275344 bytes from C:\Users\FKP\AppData\Local\Temp\arduino_build_696178/sketch_apr1
................................................................................ [ 29% ]
................................................................................ [ 59% ]
................................................................................ [ 89% ]
............................                                                     [ 100% ]
```

```
Connecting
.
.
.
.
.
.
.


Connected to
FFInt
IP address:
192.168.43.55
28.04
200
OK
28.04
200
OK
28.04
200
OK
28.04
200
OK
```

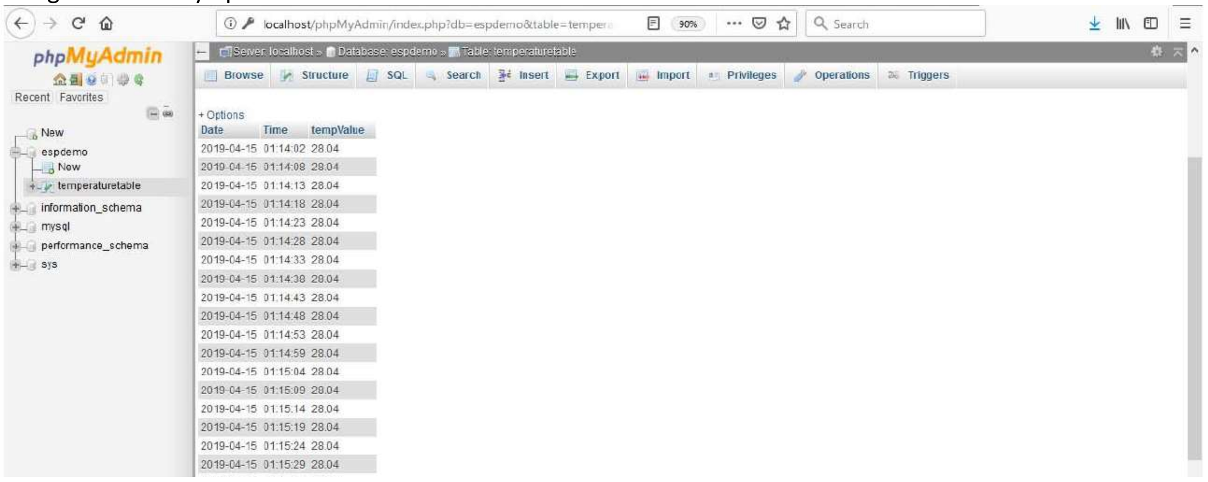
Autoscroll ✓ Show timestamp Newline 115200 baud Clear output

41 NodeMCU 1.0 (ESP-12E Module), 80 MHz, Flash, 4M (1M SPIFFS), v2 Lower Memory, Disabled, None, Only Sketch, 921600 on COM3

From the figure above, you can see that my LM35 read a 28.04 reading. is a value from LM35. After this value, you should see 200.

If you get a value of -1 and not 200, please disable your windows firewall as the firewall is blocking port 80 (normal port for http post).

Once finish, open again your MySQL database that you setup in Step 3). You should see "temperaturetable" being continuously updated.



Date	Time	tempValue
2019-04-15	01:14:02	28.04
2019-04-15	01:14:08	28.04
2019-04-15	01:14:13	28.04
2019-04-15	01:14:18	28.04
2019-04-15	01:14:23	28.04
2019-04-15	01:14:28	28.04
2019-04-15	01:14:33	28.04
2019-04-15	01:14:38	28.04
2019-04-15	01:14:43	28.04
2019-04-15	01:14:48	28.04
2019-04-15	01:14:53	28.04
2019-04-15	01:14:59	28.04
2019-04-15	01:15:04	28.04
2019-04-15	01:15:09	28.04
2019-04-15	01:15:14	28.04
2019-04-15	01:15:19	28.04
2019-04-15	01:15:24	28.04
2019-04-15	01:15:29	28.04