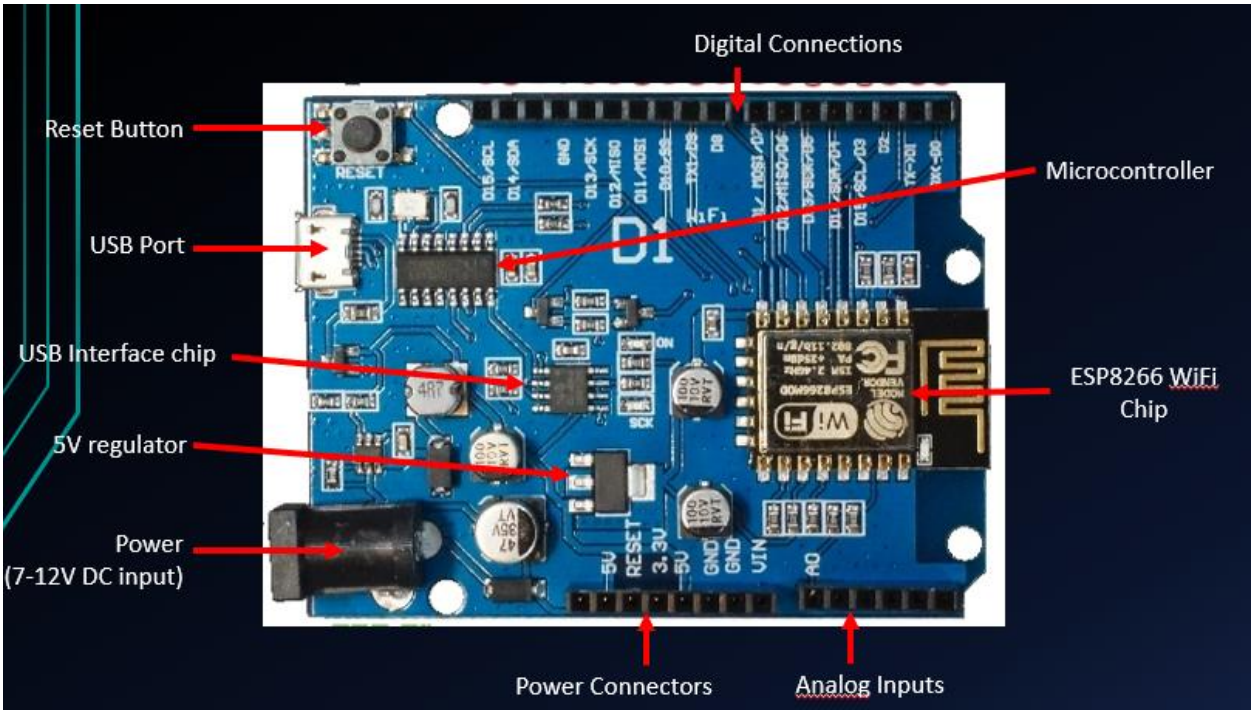
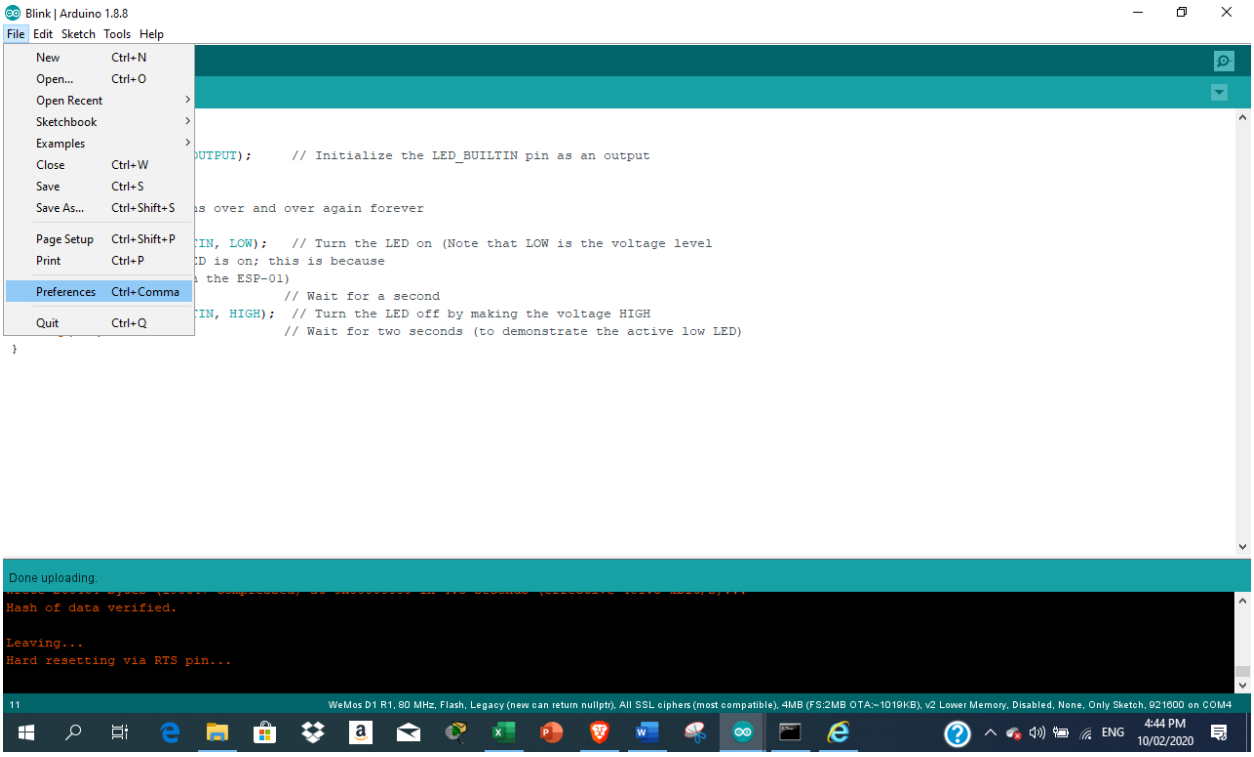
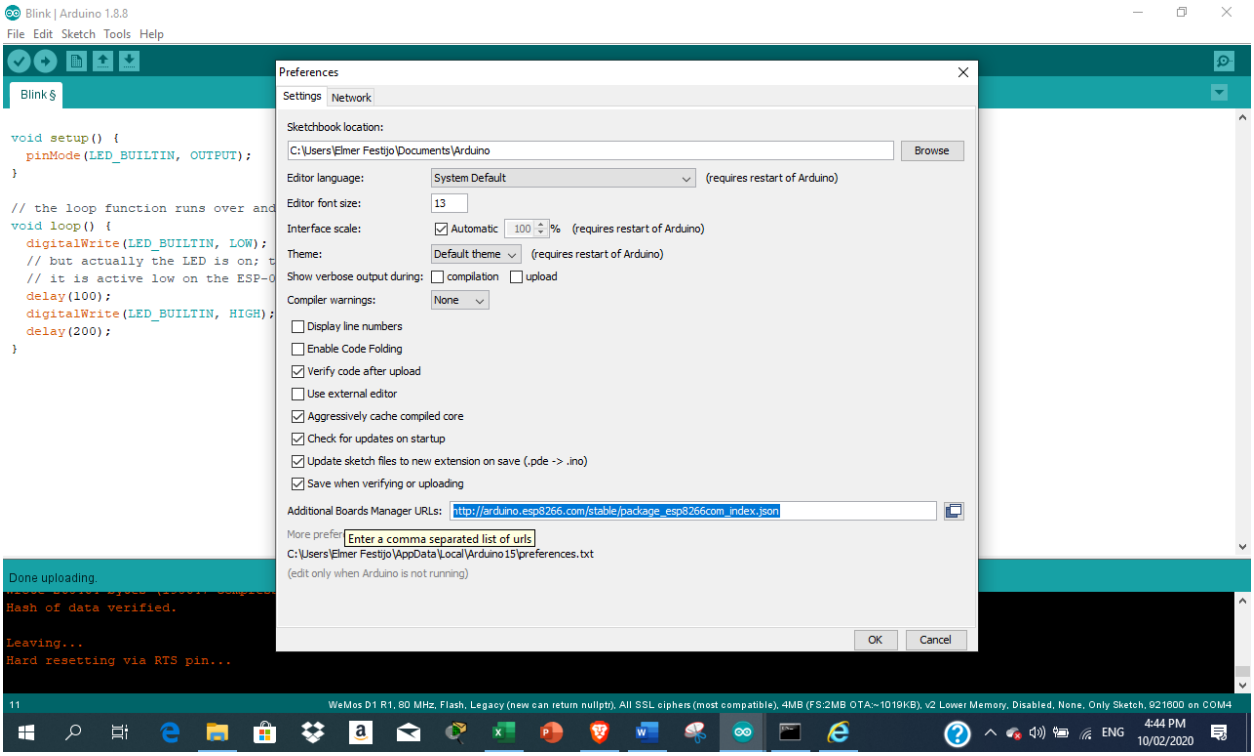
WEMOS BOARD

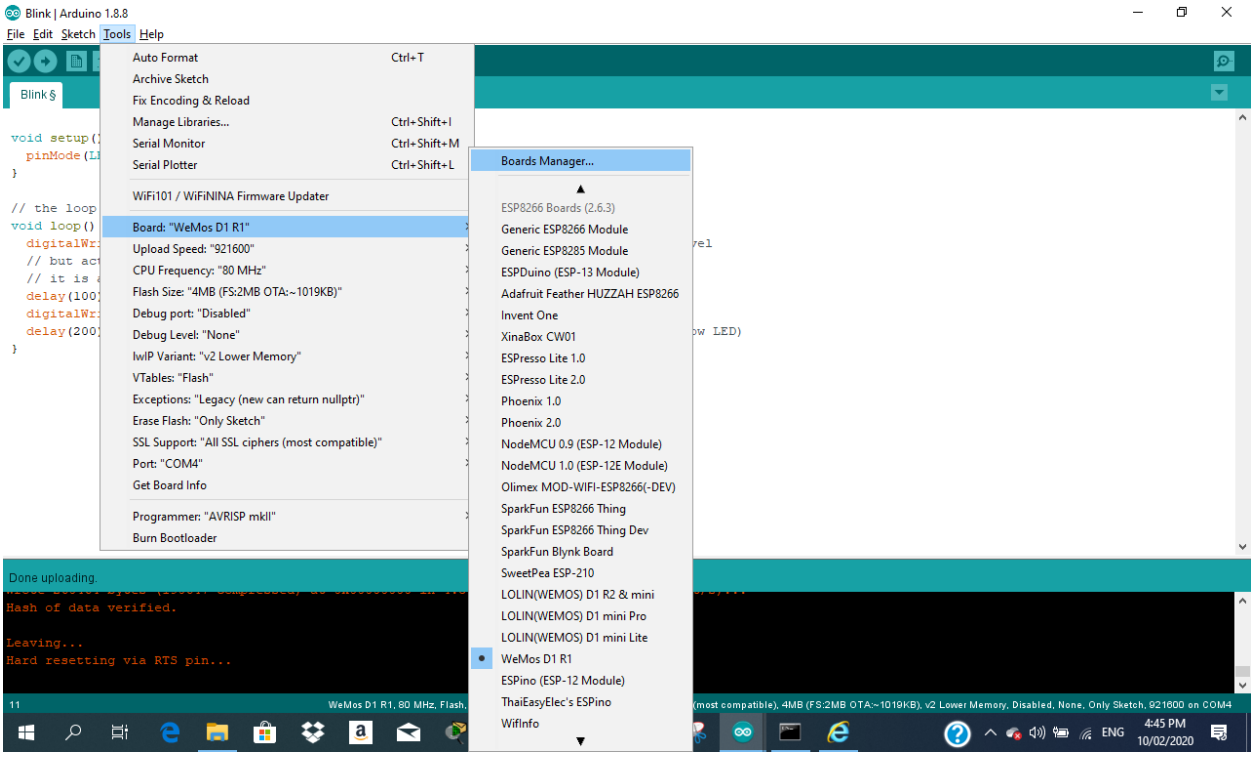


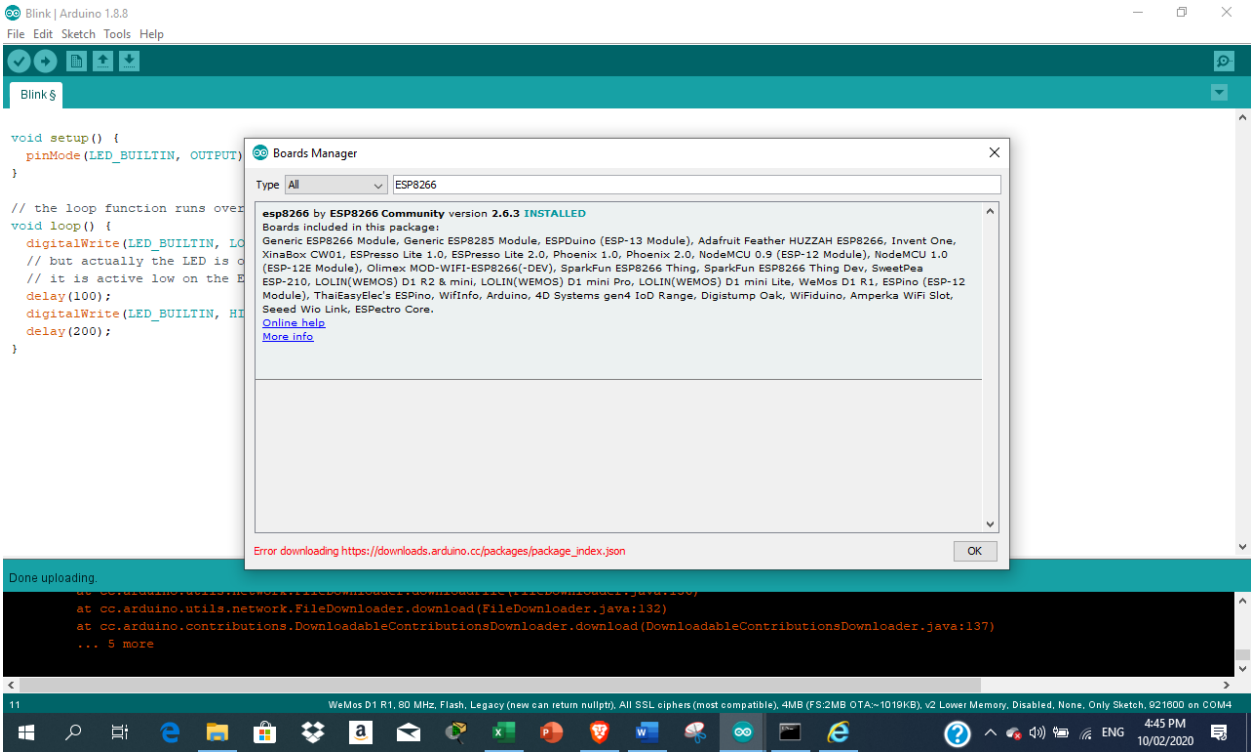
ESP8266 Board Installation





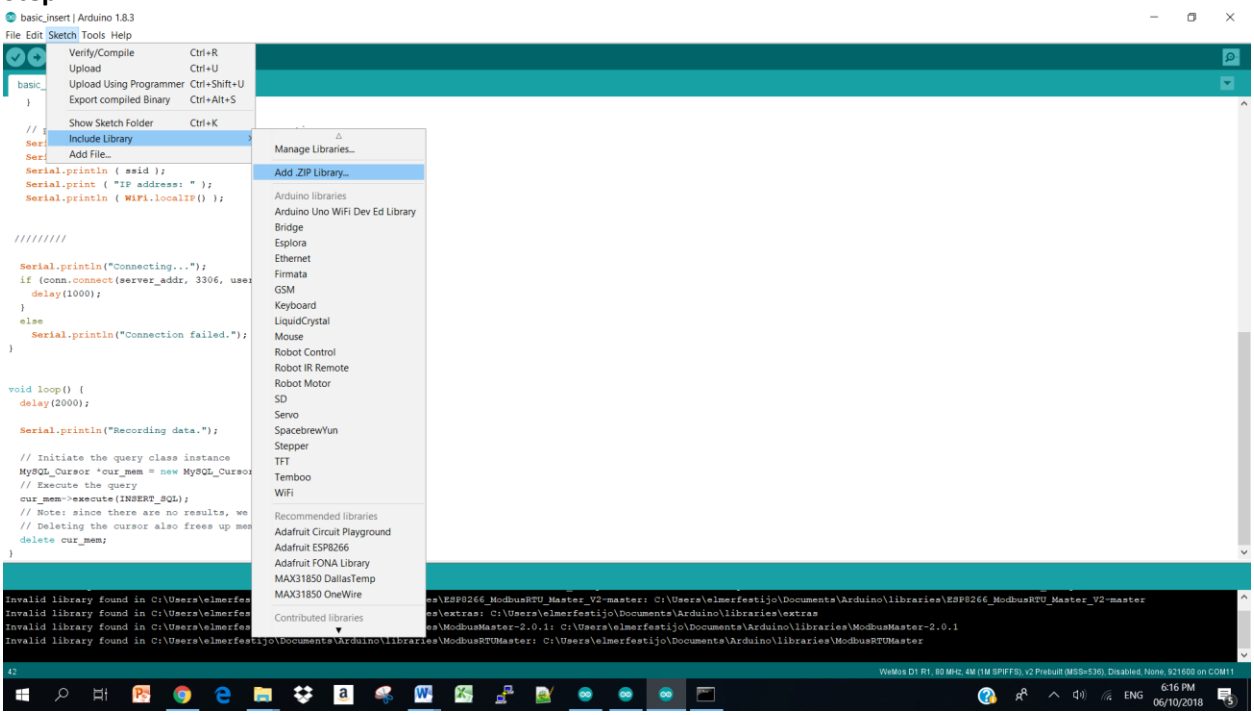
http://arduino.esp8266.com/stable/package_esp8266com_index.json

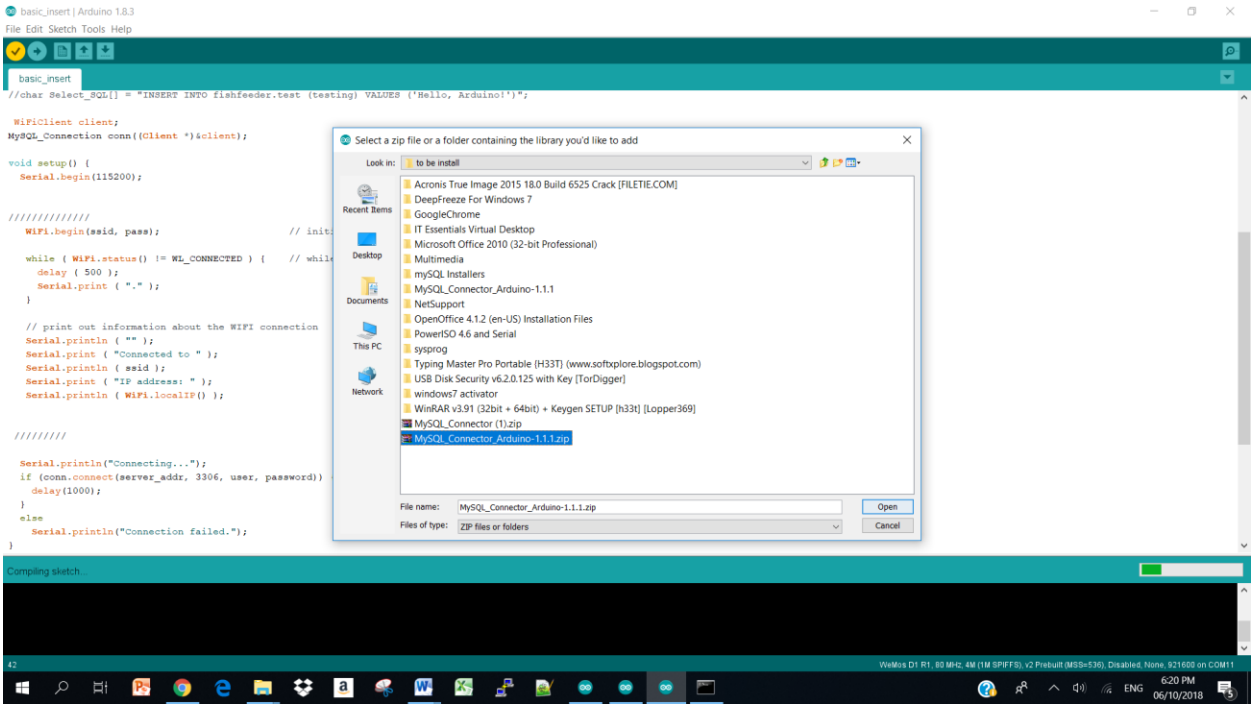




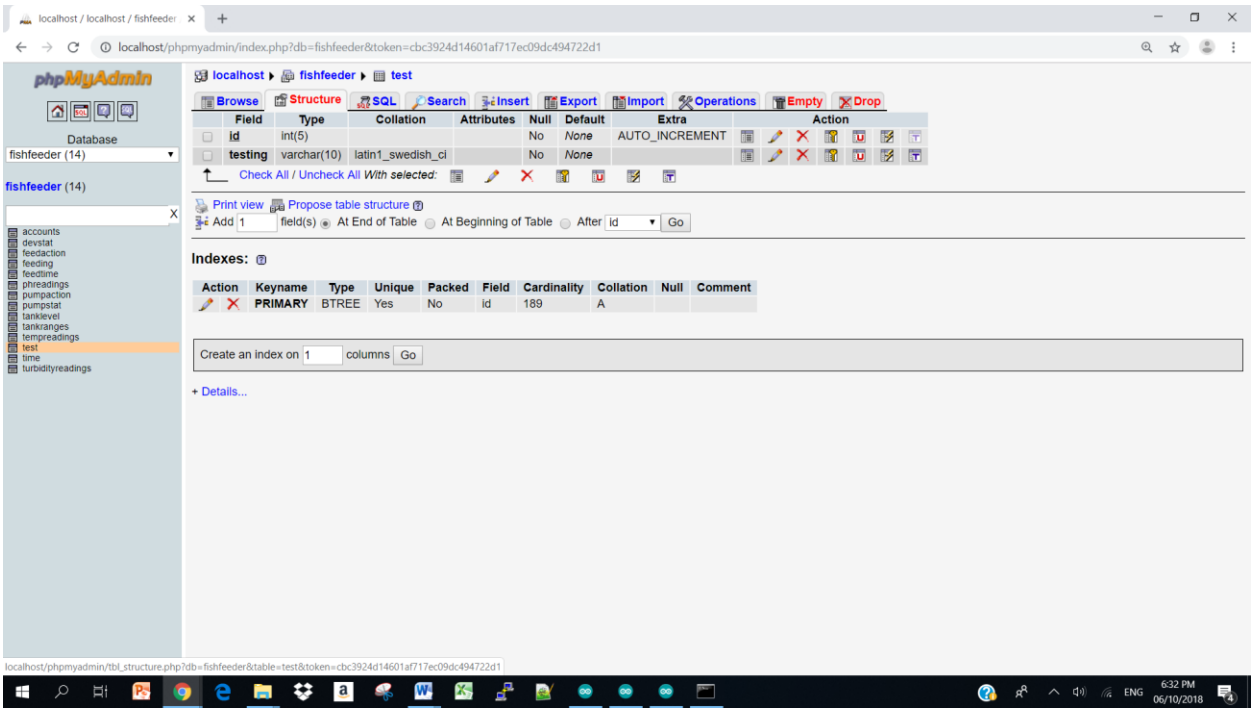
Insert Data to MySQL

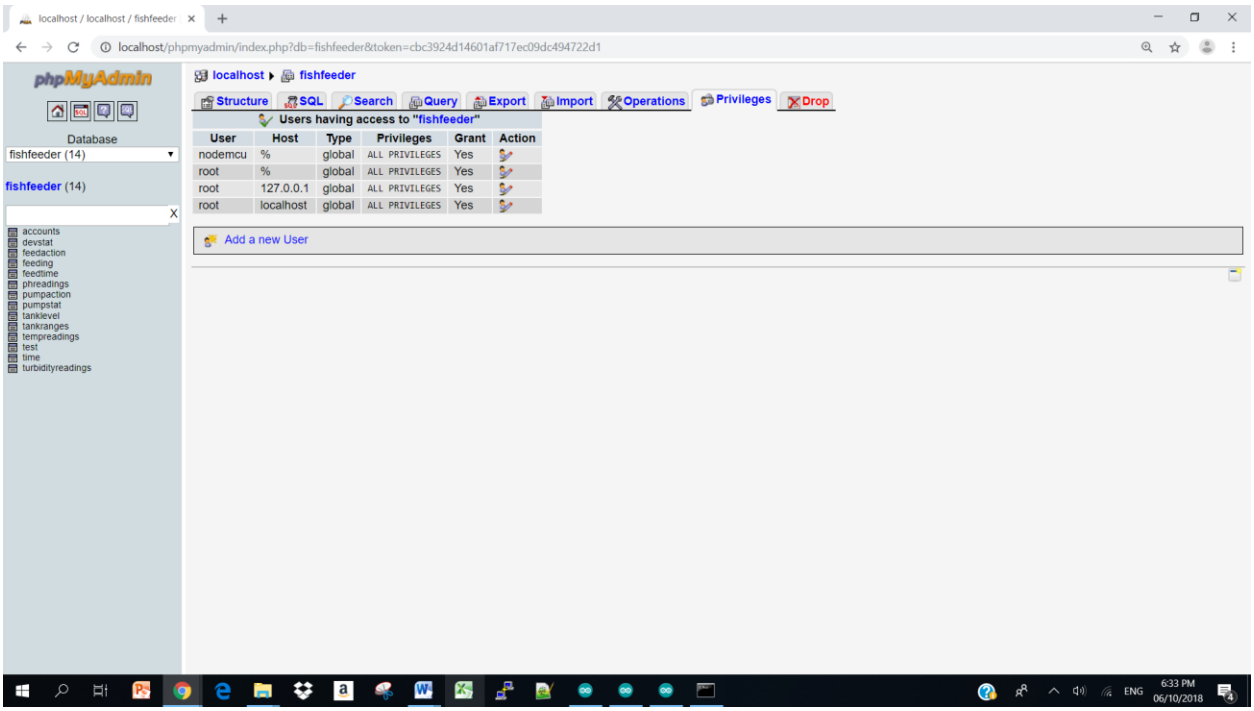
Step 1



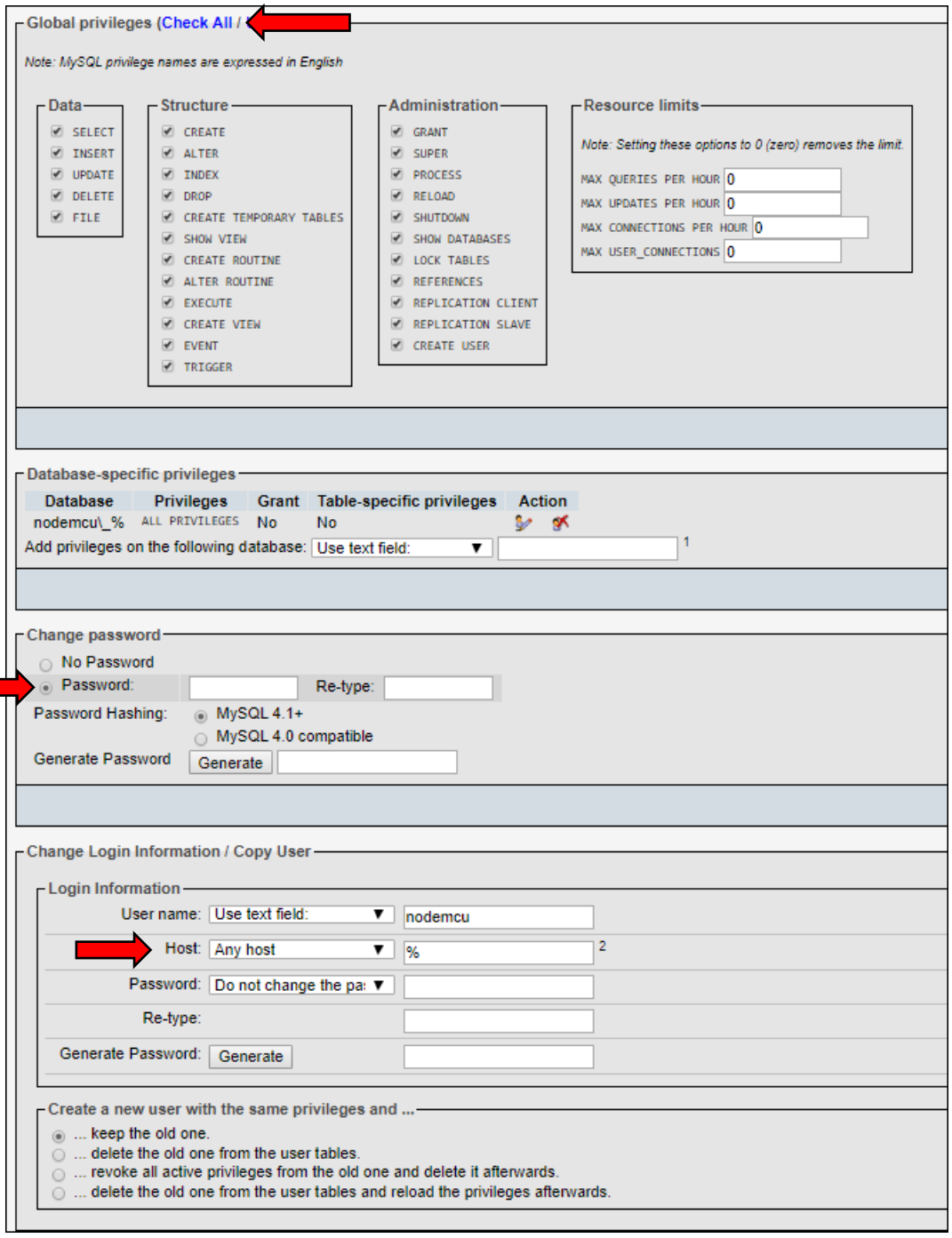


Step 2 - Create your Database.





Add user or change user privileges


Global privileges (Check All / Uncheck All)
Note: MySQL privilege names are expressed in English
Data
SELECT
INSERT
UPDATE
DELETE
FILE
Structure
CREATE
ALTER
INDEX
DROP
CREATE TEMPORARY TABLES
SHOW VIEW
CREATE ROUTINE
ALTER ROUTINE
EXECUTE
CREATE VIEW
EVENT
TRIGGER
Administration
GRANT
SUPER
PROCESS
RELOAD
SHUTDOWN
SHOW DATABASES
LOCK TABLES
REFERENCES
REPLICATION CLIENT
REPLICATION SLAVE
CREATE USER
Resource limits
Note: Setting these options to 0 (zero) removes the limit.
MAX QUERIES PER HOUR 0
MAX UPDATES PER HOUR 0
MAX CONNECTIONS PER HOUR 0
MAX USER_CONNECTIONS 0
Database-specific privileges
Database
Privileges
Grant
Table-specific privileges
Action
nodemcu_%
ALL PRIVILEGES
No
No
Add privileges on the following database:
Use text field:
1
Change password
No Password
Password:
Re-type:
Password Hashing:
MySQL 4.1+
MySQL 4.0 compatible
Generate Password
Generate
Change Login Information / Copy User
Login Information
User name:
Use text field:
nodemcu
Host:
Any host
%
2
Password:
Do not change the pa
Re-type:
Generate Password:
Generate
Create a new user with the same privileges and ...
... keep the old one.
... delete the old one from the user tables.
... revoke all active privileges from the old one and delete it afterwards.
... delete the old one from the user tables and reload the privileges afterwards.

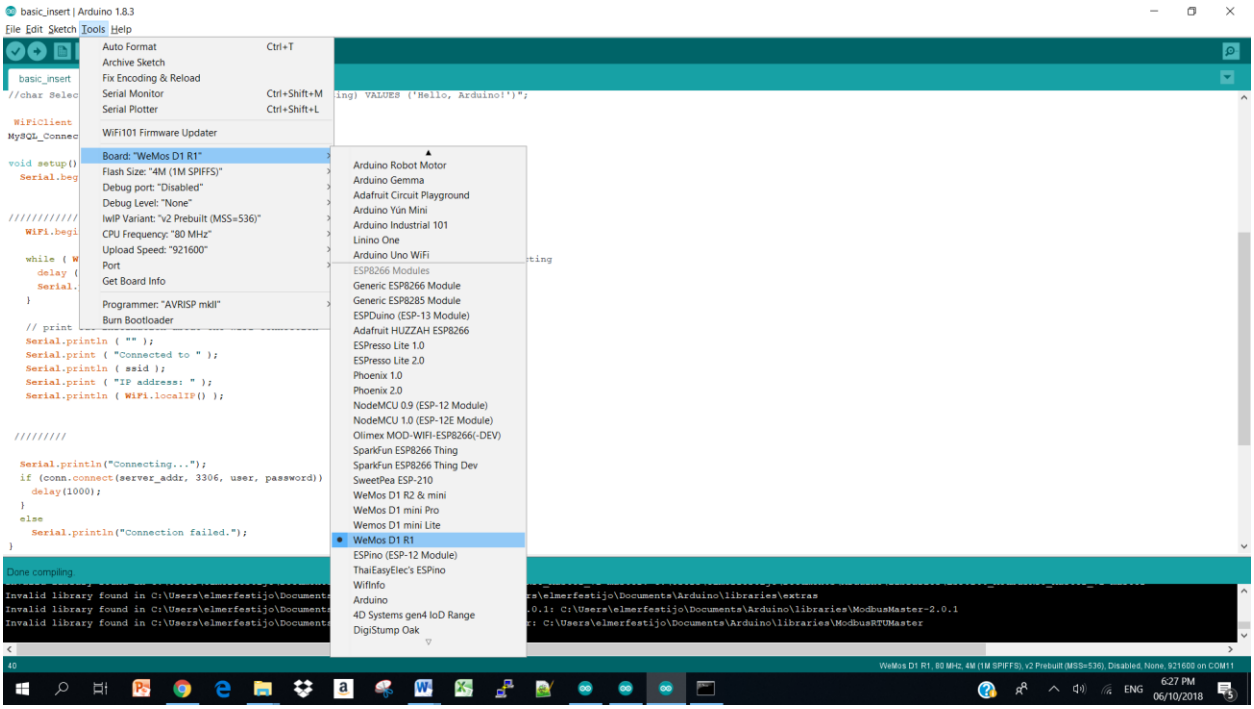
Step 3

In arduino IDE, create New File and paste this code:

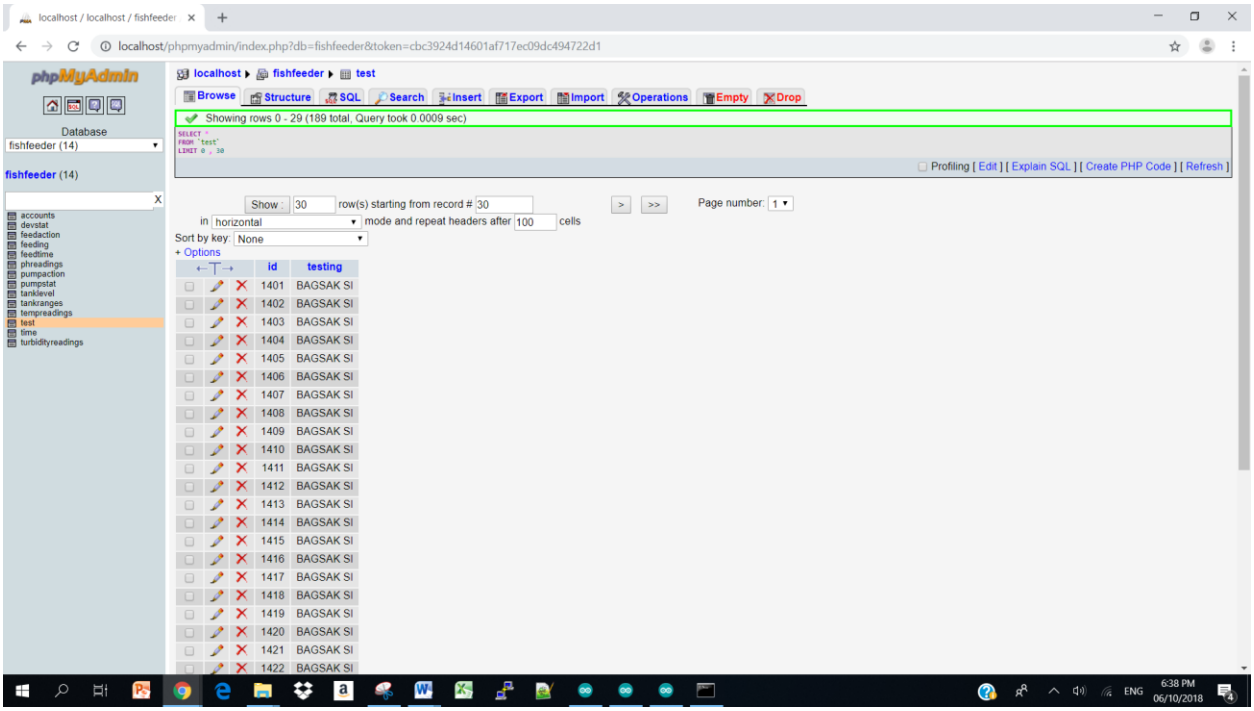
```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <MySQL_Connection.h>
#include <MySQL_Cursor.h>

 IPAddress server_addr(192,168,31,104);   // IP of the MySQL server (change this)
 char user[] = "nodemcu";                 // MySQL user login username (change this)
 char password[] = "nodemcu";             // MySQL user login password (change this)

 // WiFi card example
 char ssid[] = "WEMOS";                   // your SSID (change this)
 char pass[] = "sysprog101";              // your SSID Password (change this)

// Sample query
char INSERT_SQL[] = "INSERT INTO fishfeeder.test (testing) VALUES ('NIGHT SHIFT SI RORO!')";  /// (change this)

 WiFiClient client;
MySQL_Connection conn((Client *)&client);

void setup() {
  Serial.begin(115200);
 WiFi.begin(ssid, pass);                  // initializing the WIFI library
    while ( WiFi.status() != WL_CONNECTED ) {    // while loop to write dots during connecting
     delay ( 500 );
     Serial.print ( "." );
   }
    // print out information about the WIFI connection
   Serial.println ( "" );
   Serial.print ( "Connected to " );
   Serial.println ( ssid );
   Serial.print ( "IP address: " );
   Serial.println ( WiFi.localIP() );
   Serial.println("Connecting...");
 if (conn.connect(server_addr, 3306, user, password)) {
    delay(1000);
 }
 else
    Serial.println("Connection failed.");
}

void loop() {
  delay(2000);
  Serial.println("Recording data.");
  // Initiate the query class instance
  MySQL_Cursor *cur_mem = new MySQL_Cursor(&conn);
  // Execute the query
  cur_mem->execute(INSERT_SQL);
  // Note: since there are no results, we do not need to read any data
  // Deleting the cursor also frees up memory used
  delete cur_mem;
}
```

Step 4

*Change your board to WeMos D1 R1 if you are using wemos as well as the port then upload it.
*Change your board to NodeMCU 1.0 if you are using nodemcu as well as the port then upload it.



Step 5 - Check your database if insertion is successful.



NOTE:

1. **TURN OFF YOUR FIREWALL AND ANTIVIRUS**
2. **CHECK YOUR SERVERS IP ADDRESS (PC)**
3. **WEMOS AND SERVER ARE CONNECTED ON THE SAME NETWORK**
4. **CHECK SERVERS USER PRIVELEGES**
5. **# of user connected to wifi.**

6. BAWAL MAG INTERNET

Read MySQL Data

Code:

```
#include "ESP8266WiFi.h"
#include <Ethernet.h>
#include <MySQL_Connection.h>
#include <MySQL_Cursor.h>

IPAddress server_addr(192,168,31,104);  // IP of the MySQL *server* here
char user[] = "nodemcu";              // MySQL user login username
char password[] = "nodemcu";       // MySQL user login password
const char* SSID = "Xiaomi Mi Max";        // Your network SSID
const char* PASS = "password";                       // Your network password
const char QUERY_POP[] = "SELECT testing FROM fishfeeder.test";    //database query
char query[128];

WiFiClient client;
MySQL_Connection conn((Client *)&client);

void setup() {
  Serial.begin(115200);
  while (!Serial); // wait for serial port to connect
  Serial.print("Connecting to "+*SSID);
  WiFi.begin(SSID, PASS);
  Serial.println("going into wl connect");

  while (WiFi.status() != WL_CONNECTED) //not connected,  ...waiting to connect
    {
      delay(1000);
      Serial.print(".");
    }

  Serial.println("wl connected");
  Serial.println("");
  Serial.println("Credentials accepted! Connected to wifi\n ");
  Serial.println(WiFi.localIP());
  Serial.println("");

  if (conn.connect(server_addr, 3306, user, password)) {
    delay(1000);
  }
  else
     {
```

```
    Serial.println("Connection failed.");
    }
}

void loop() {
 delay(1000);
 Serial.println("> Running SELECT with dynamically supplied parameter");

 // Initiate the query class instance
 MySQL_Cursor *cur_mem = new MySQL_Cursor(&conn);
 // Supply the parameter for the query
 // Here we use the QUERY_POP as the format string and query as the
 // destination. This uses twice the memory so another option would be
 // to allocate one buffer for all formatted queries or allocate the
 // memory as needed (just make sure you allocate enough memory and
 // free it when you're done!).
 sprintf(query, QUERY_POP, 9000000);
 // Execute the query
 cur_mem->execute(query);
 // Fetch the columns and print them
 column_names *cols = cur_mem->get_columns();
 for (int f = 0; f < cols->num_fields; f++) {
  Serial.print(cols->fields[f]->name);
  if (f < cols->num_fields-1) {
   Serial.print(',');
  }
 }
 Serial.println();
 // Read the rows and print them
 row_values *row = NULL;
 do {
  row = cur_mem->get_next_row();
  if (row != NULL) {
   for (int f = 0; f < cols->num_fields; f++) {
    Serial.print(row->values[f]);
    if (f < cols->num_fields-1) {
     Serial.print(',');
    }
   }
   Serial.println();
  }
 } while (row != NULL);
 // Deleting the cursor also frees up memory used
 delete cur_mem;
}
```

WEMOS AS WEBSERVER

Parts

1 x Wemos D1 or D2
1 x USB cable
1 x LED

Code

```
#include <ESP8266WiFi.h>

const char* ssid = "ssid name"; ---------- CHANGE THIS
const char* password = "ssid password"; ---------- CHANGE THIS

int ledPin = D5; ---------- CHANGE THIS
WiFiServer server(80);

void setup() {
  Serial.begin(115200);
  delay(10);


  pinMode(ledPin, OUTPUT);
  digitalWrite(ledPin, LOW);

  // Connect to WiFi network
  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");

  // Start the server
  server.begin();
  Serial.println("Server started");

  // Print the IP address
  Serial.print("Use this URL : ");
  Serial.print("http://");
  Serial.print(WiFi.localIP());
  Serial.println("/");

}

void loop() {

  // Check if a client has connected
  WiFiClient client = server.available();
  if (!client) {
    return;
  }

  // Wait until the client sends some data
  Serial.println("new client");
  while(!client.available()){
    delay(1);
  }
```

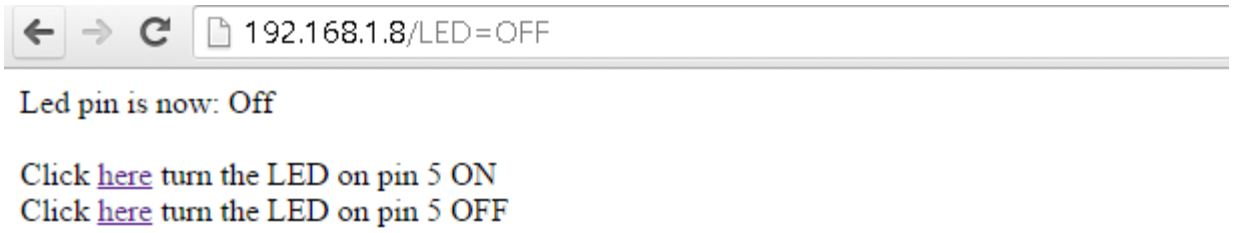
```
// Read the first line of the request
String request = client.readStringUntil('\r');
Serial.println(request);
client.flush();

// Match the request

int value = LOW;
if (request.indexOf("/LED=ON") != -1) {
  digitalWrite(ledPin, HIGH);
  value = HIGH;
}
if (request.indexOf("/LED=OFF") != -1){
  digitalWrite(ledPin, LOW);
  value = LOW;
}



// Return the response
client.println("HTTP/1.1 200 OK");
client.println("Content-Type: text/html");
client.println(""); //  do not forget this one
client.println("<!DOCTYPE HTML>");
client.println("<html>");

client.print("Led pin is now: ");

if(value == HIGH) {
  client.print("On");
} else {
  client.print("Off");
}
client.println("<br><br>");
client.println("Click <a href=\"/LED=ON\">here</a> turn the LED on pin 5 ON<br>");
client.println("Click <a href=\"/LED=OFF\">here</a> turn the LED on pin 5 OFF<br>");
client.println("</html>");

delay(1);
Serial.println("Client disconnected");
Serial.println("");

}
```

Results

Open the serial monitor , all going well and you will see the IP address and messages like the following wemos ip address

```
Connecting to
........
WiFi connected
Server started
Use this URL to connect: http://192.168.1.8/
```

Using your favorite web browser navigate to the IP above

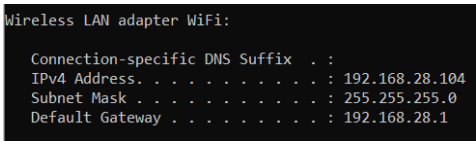


Update

I tried this out and it uses a static IP address

```
//This example will set up a static IP - in this case 192.168.28.105

#include <ESP8266WiFi.h>

const char* ssid = "ssid name";---------- CHANGE THIS
const char* password = "ssid password";---------- CHANGE THIS

int ledPin = D5; ---------- CHANGE THIS BASED ON THE PIN YOU USED
WiFiServer server(80);
IPAddress ip(192, 168, 28, 105); // where xx is the
desired IP Address
IPAddress gateway(192, 168, 28, 1); // set gateway to
match your network

void setup() {

  Serial.begin(115200);
  delay(10);


  pinMode(ledPin, OUTPUT);
  digitalWrite(ledPin, LOW);

  Serial.print(F("Setting static ip to : "));
  Serial.println(ip);

  // Connect to WiFi network
  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  IPAddress subnet(255, 255, 255, 0); // set subnet mask to match your
network
  WiFi.config(ip, gateway, subnet);
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
```

LAPTOP'S IP ADDRESS



```
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");

  // Start the server
  server.begin();
  Serial.println("Server started");

  // Print the IP address
  Serial.print("Use this URL : ");
  Serial.print("http://");
  Serial.print(WiFi.localIP());
  Serial.println("/");

}

void loop() {

  // Check if a client has connected
  WiFiClient client = server.available();
  if (!client) {
    return;
  }

  // Wait until the client sends some data
  Serial.println("new client");
  while(!client.available()){
    delay(1);
  }

  // Read the first line of the request
  String request = client.readStringUntil('\r');
  Serial.println(request);
  client.flush();

  // Match the request

  int value = LOW;
  if (request.indexOf("/LED=ON") != -1) {
    digitalWrite(ledPin, HIGH);
    value = HIGH;
  }
  if (request.indexOf("/LED=OFF") != -1){
    digitalWrite(ledPin, LOW);
    value = LOW;
  }

  // Return the response
  client.println("HTTP/1.1 200 OK");
  client.println("Content-Type: text/html");
  client.println(""); //  do not forget this one
  client.println("<!DOCTYPE HTML>");
  client.println("<html>");

  client.print("Led pin is now: ");
```

```
  if(value == HIGH) {
    client.print("On");
  } else {
    client.print("Off");
  }
  client.println("<br><br>");
  client.println("Click <a href=\"/LED=ON\">here</a> turn the LED on pin 5
ON<br>");
  client.println("Click <a href=\"/LED=OFF\">here</a> turn the LED on pin 5
OFF<br>");
  client.println("</html>");

  delay(1);
  Serial.println("Client disconnected");
  Serial.println("");

}
```

ACTIVITIES TO PERFORM

1. **Installation of wemos board/driver on your laptop.**
2. **Insertion of data to MySQL**
3. **Retrieve data from MySQL**
4. **Wemos as Webserver**
5. **Create Application connected to MySQL that will turn on/off the 3 led light.**