# CHAPTER 1

## Hardware Interface

- **Gas Sensor module** - To sense or detect gas. Ex. Gas detector, Smoke detector
- **LDR Sensor module** - To sense or detect color on the surface. Ex. Line Following Robot(LFR), Solar Optimizer
- **IR Sensor module** - To sense or detect color on the surface. Ex. Line Following Robot(LFR)
- **Current Sensor module** - To sense current in the power line.
- **LM35 Sensor** - To sense temperature.
- **PIR Sensor** - To sense or detect motion.
- **DHT11** - To sense temperature and humidity
- **Sonar Sensor** - To measure distance.
- **RDM6300 RFID sensor module** - To sense RFID cards or tags. Ex. Access controller
- **I2C LCD** - To show messages.
- **Keypad** - To give input to the controller.
- **RTC module** - To get real time.
- **Relay module** - works as an electronic switch.
- **DC motor** - Direct current drive motor.
- **Push button** - To take input to the controller from the button.

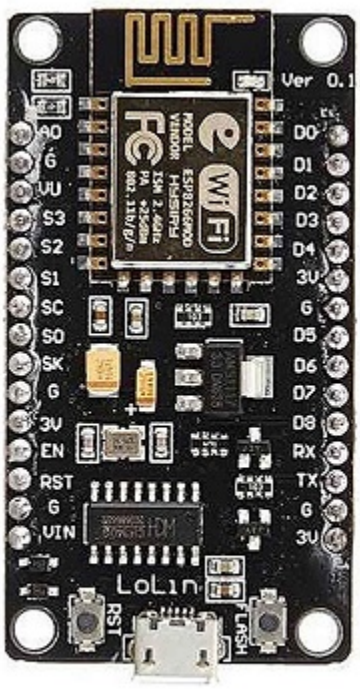
Additional Hardware Required -

1. Breadboard
2. ESP8266 Node MCU
3. USB Mini cable
4. Jumper Cables

Software Requirements -

1. Arduino IDE (version below 1.8.12)

## Client Server Model HTTP




**Client**

**Server**

Software Requirements -

1. Visual Studio Code
2. Arduino IDE

Software Packages -

1. Flask (Python)

## Database Interface

Software Requirements -

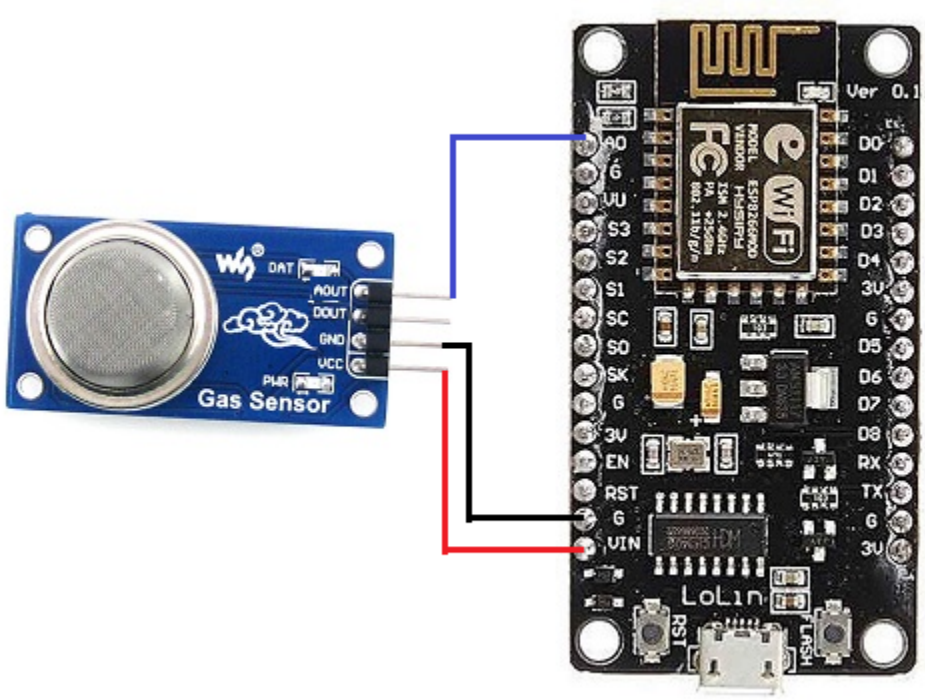
1. Visual Studio Code
2. DB Browser for SQLite or SQLite Studio

Database - SQLite

# CHAPTER 2

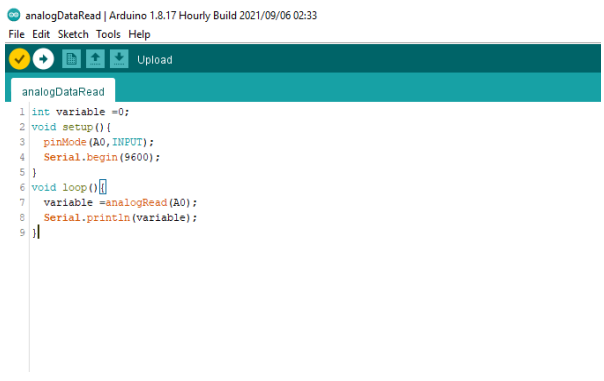
## Hardware Interface

1. Gas sensor module

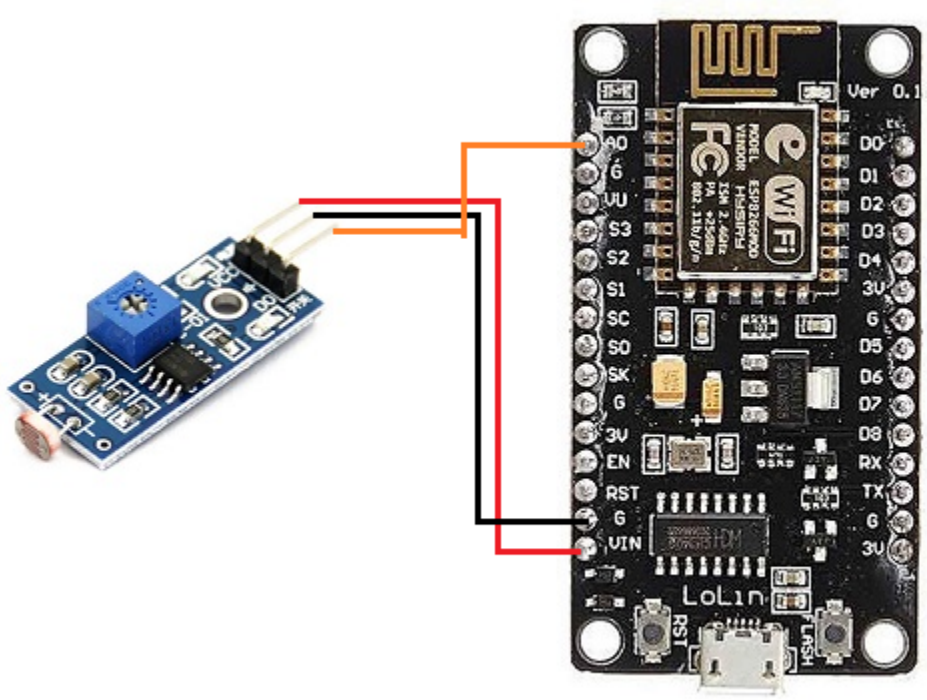



Program

```
int variable =0;
void setup(){
  pinMode(A0,INPUT);
  Serial.begin(9600);
}
void loop(){
  variable =analogRead(A0);
  Serial.println(variable);
}
```

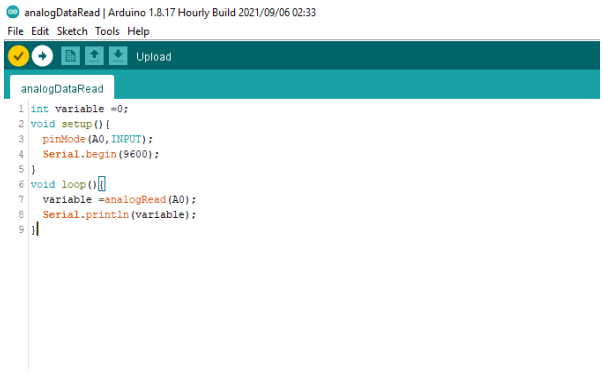


2. LDR sensor module

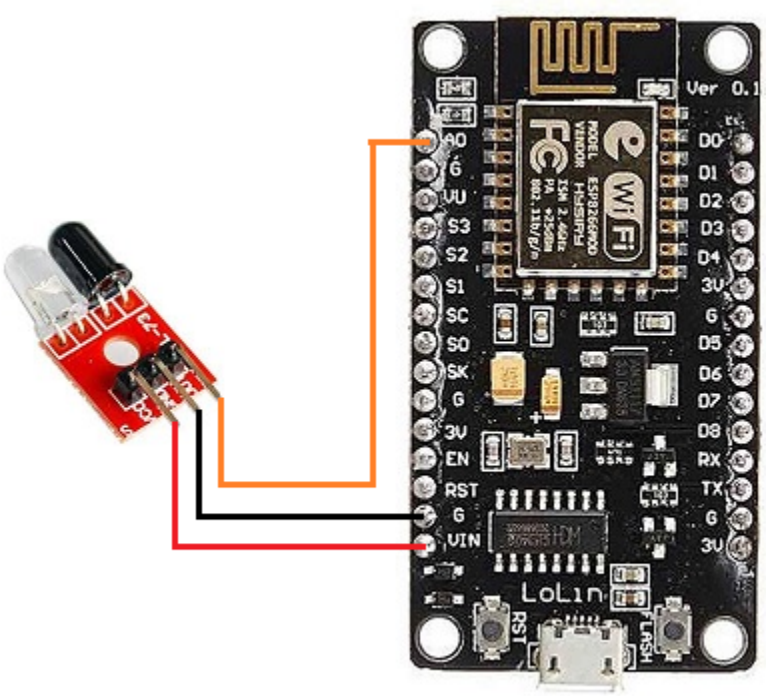



Program

```
int variable =0;
void setup(){
  pinMode(A0,INPUT);
  Serial.begin(9600);
}
void loop(){
  variable =analogRead(A0);
  Serial.println(variable);
}
```

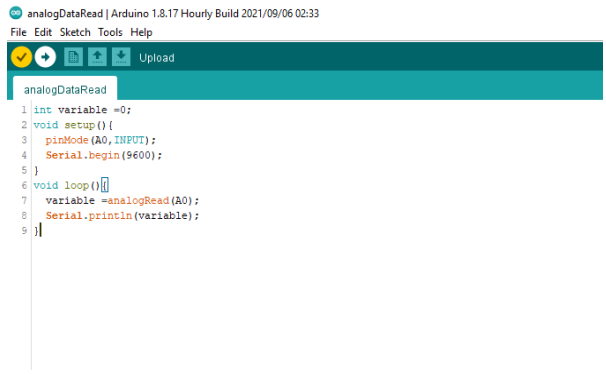


3. IR sensor module

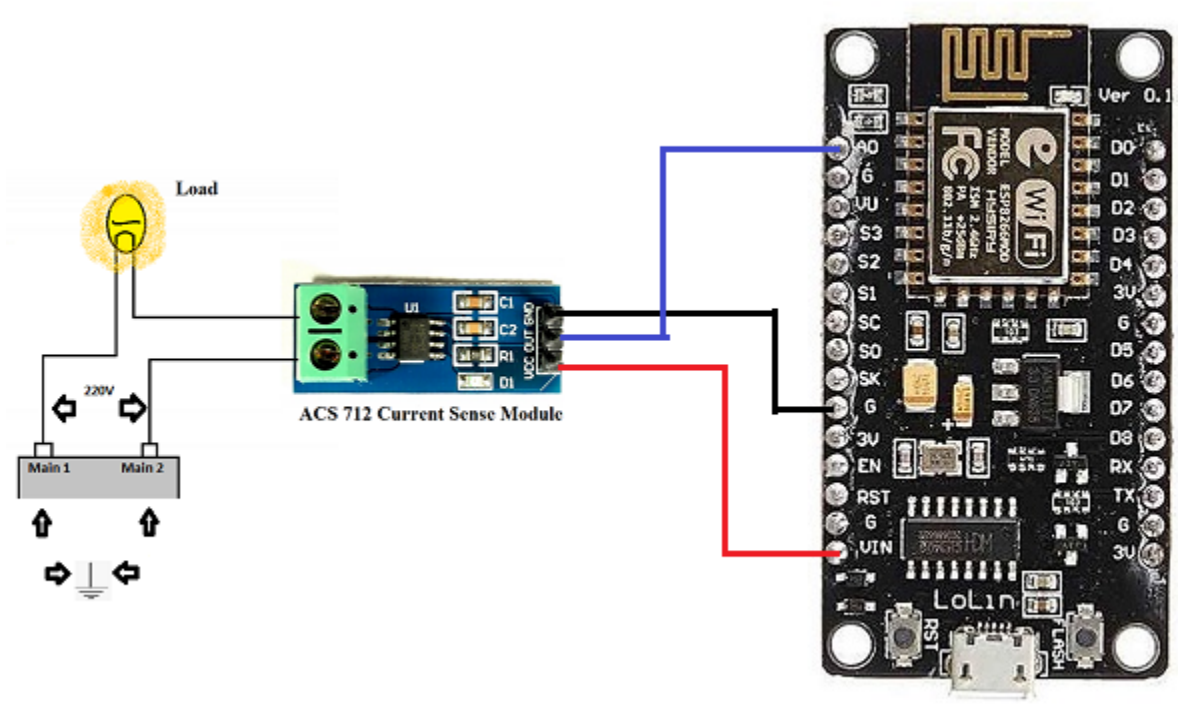



Program

```
int variable =0;
void setup(){
  pinMode(A0,INPUT);
  Serial.begin(9600);
}
void loop(){
  variable =analogRead(A0);
  Serial.println(variable);
}
```

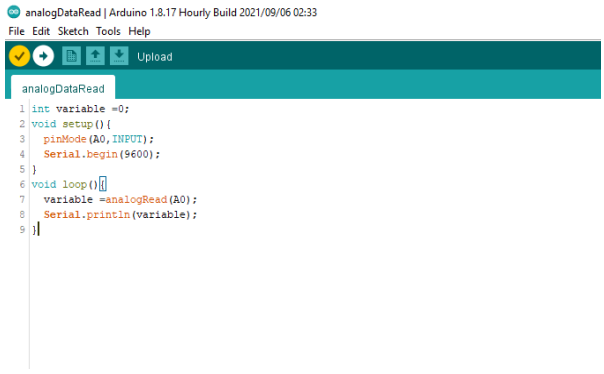


4. Current sensor module (ACS 712)

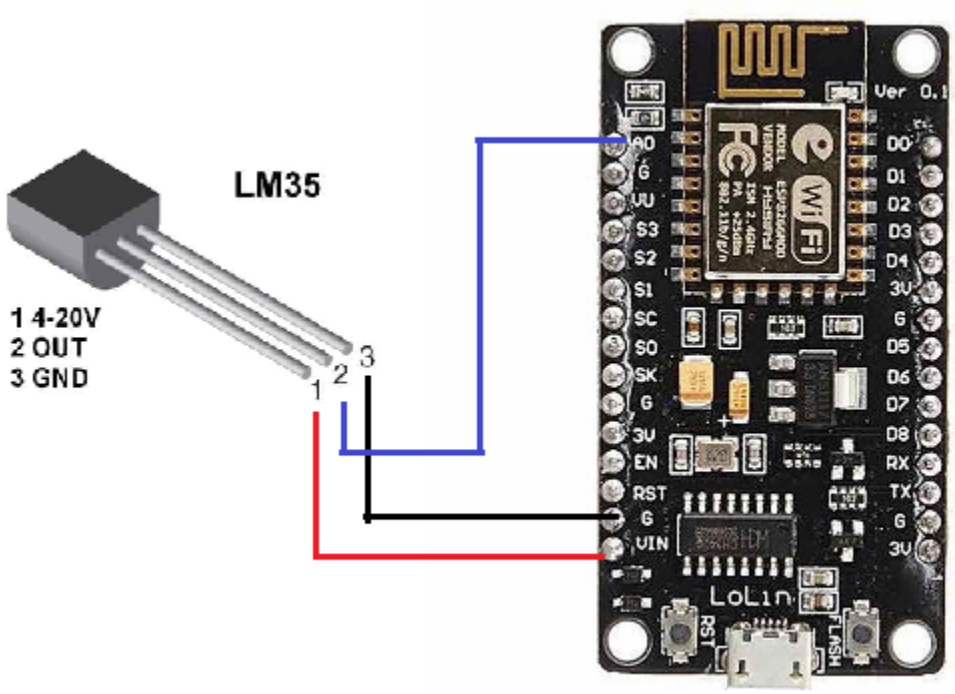



Program

```
int variable =0;
void setup(){
  pinMode(A0,INPUT);
  Serial.begin(9600);
}
void loop(){
  variable =analogRead(A0);
  Serial.println(variable);
}
```

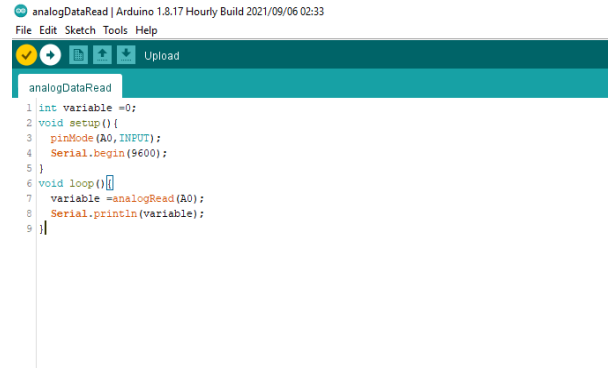


5. LM35 sensor

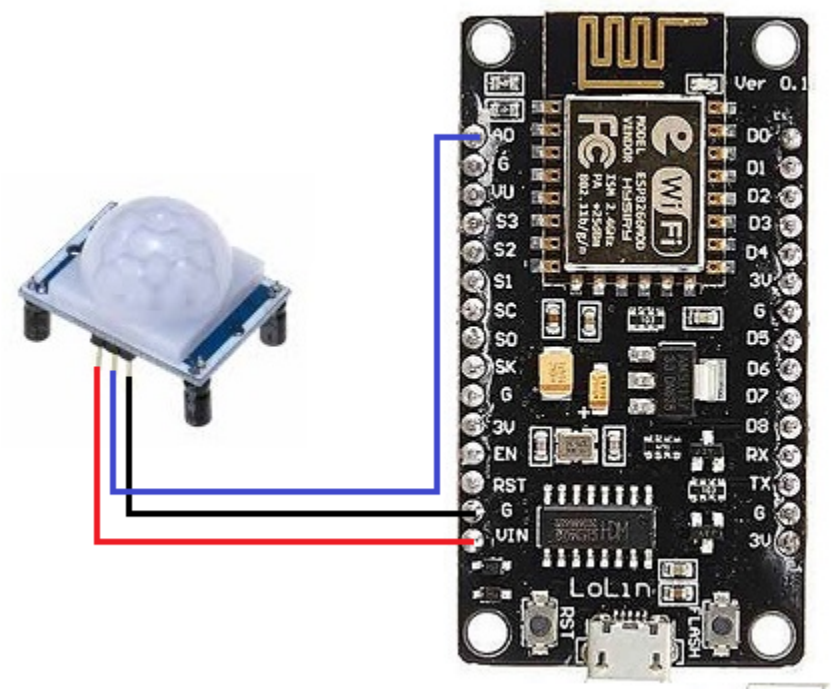



Program

```
int variable =0;
void setup(){
  pinMode(A0,INPUT);
  Serial.begin(9600);
}
void loop(){
  variable =analogRead(A0);
  Serial.println(variable);
}
```

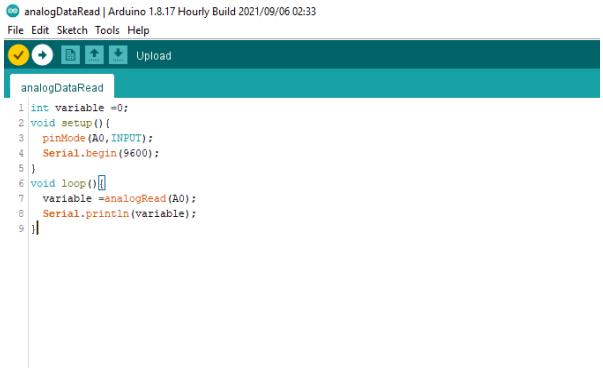


6. PIR sensor module

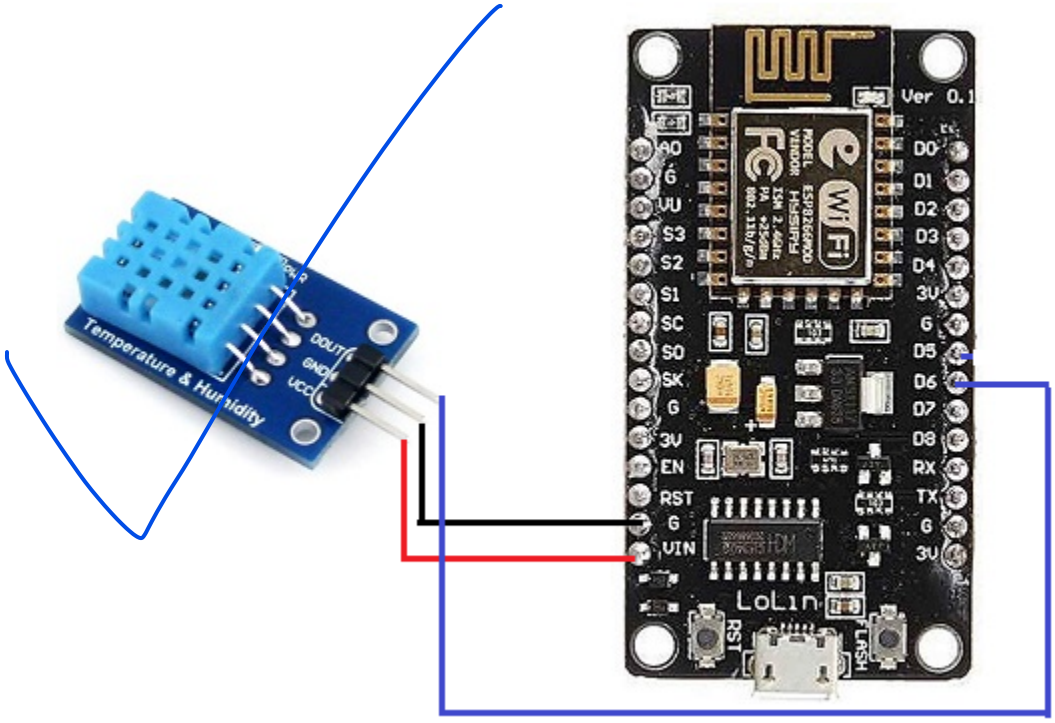



Program

```
int variable =0;
void setup(){
  pinMode(A0,INPUT);
  Serial.begin(9600);
}
void loop(){
  variable =analogRead(A0);
  Serial.println(variable);
}
```



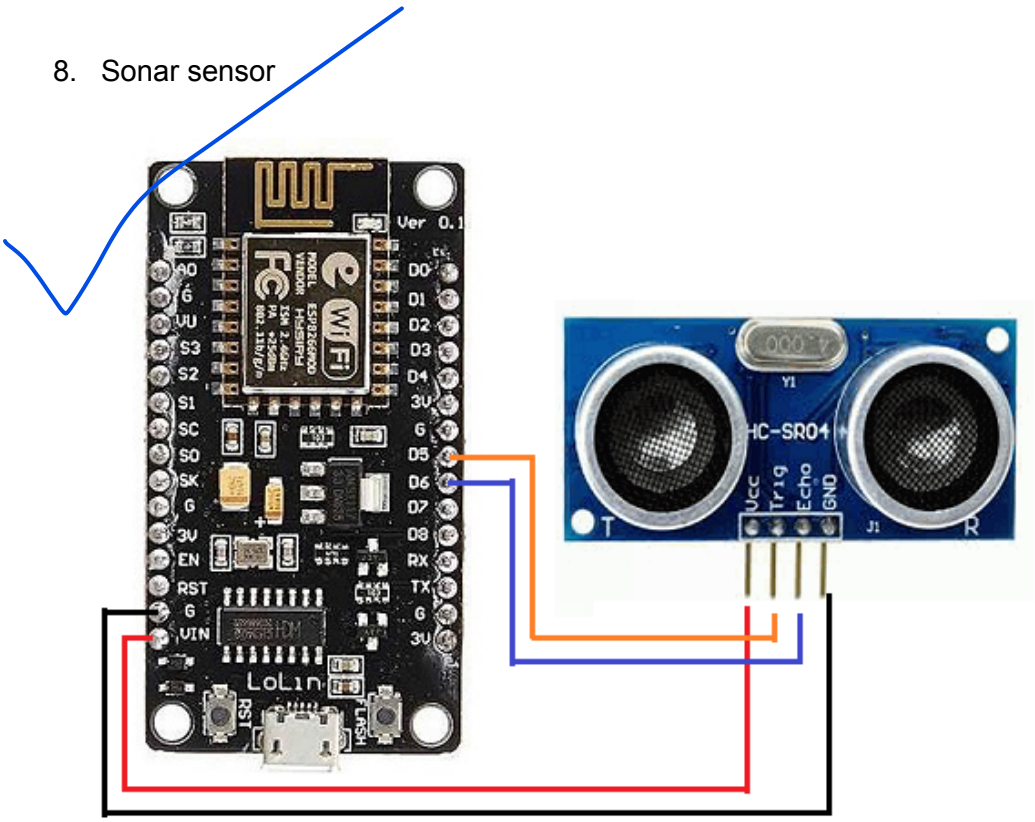
7. DHT11 sensor module




Program

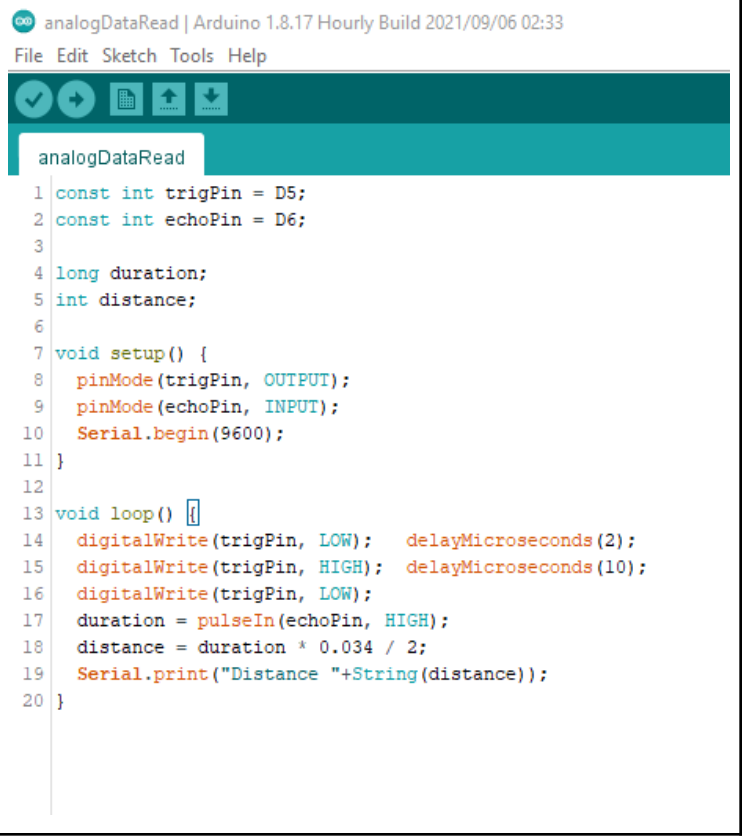
```
#include <DHT.h>

DHT dht;

void setup()
{
  Serial.begin(9600);
  dht.setup(D6);
}

void loop()
{
  delay(dht.getMinimumSamplingPeriod());

  float humidity = dht.getHumidity();
  float temperature = dht.getTemperature();

  Serial.print("Humidity "+String(humidity, 1));
  Serial.print(" ");
  Serial.print("Temperature "+String(temperature, 1));
  Serial.println();
}
```

8. Sonar sensor

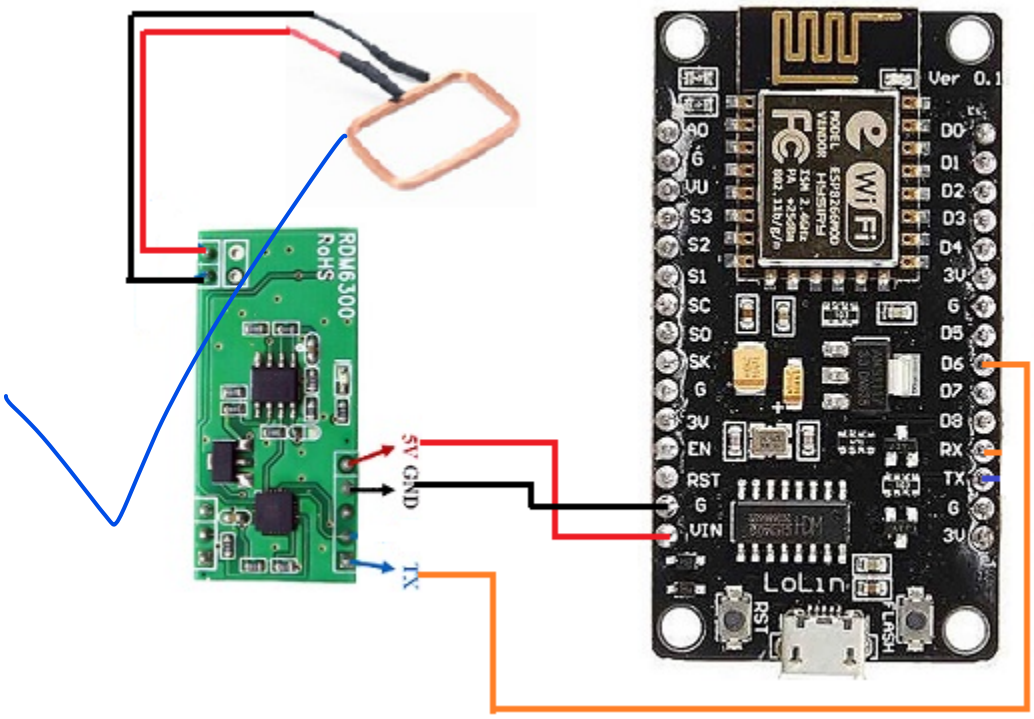



Program

```
const int trigPin = D5;
const int echoPin = D6;

long duration;
int distance;

void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  Serial.begin(9600);
}

void loop() {
  digitalWrite(trigPin, LOW);   delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = duration * 0.034 / 2;
  Serial.print("Distance "+String(distance));
}
```



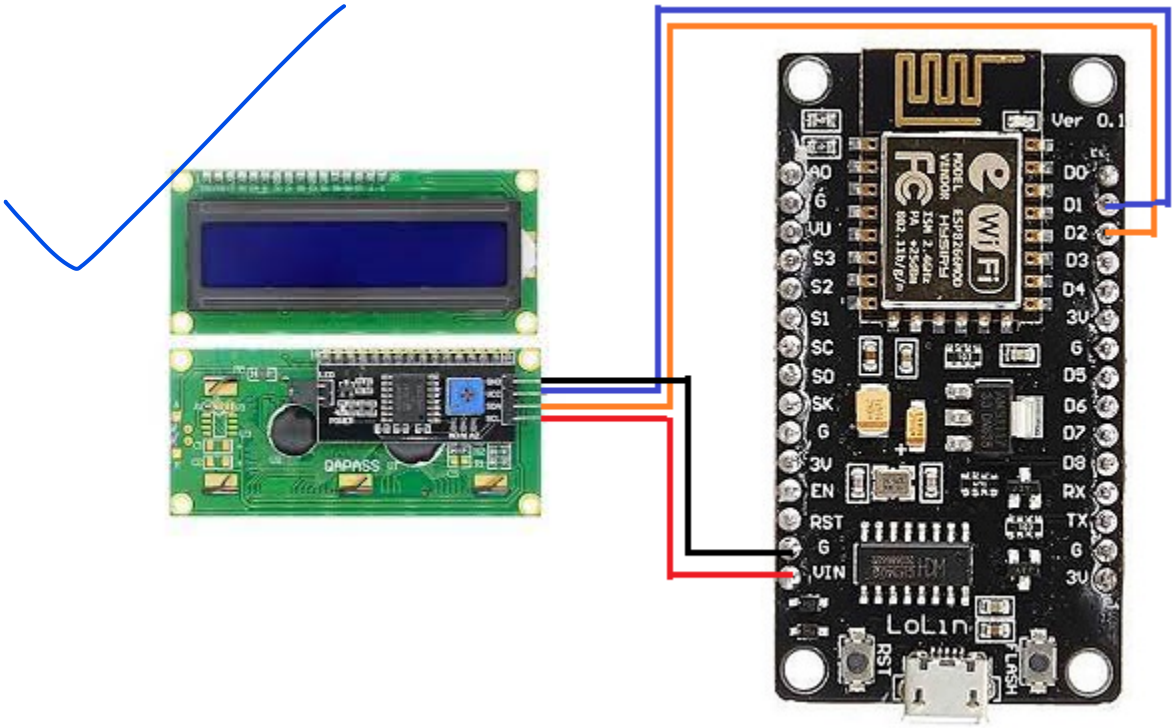
9. RFID module RDM6300




Program

```
#include <rdm6300.h>

Rdm6300 rdm6300;
int32_t cardID = 0;

void setup()
{
  Serial.begin(9600);
  rdm6300.begin(D6);
}

void loop()
{
  if (rdm6300.is_listening())
    cardID = rdm6300.get_tag_id();
    Serial.println(cardID);
    delay(2000);
}
```

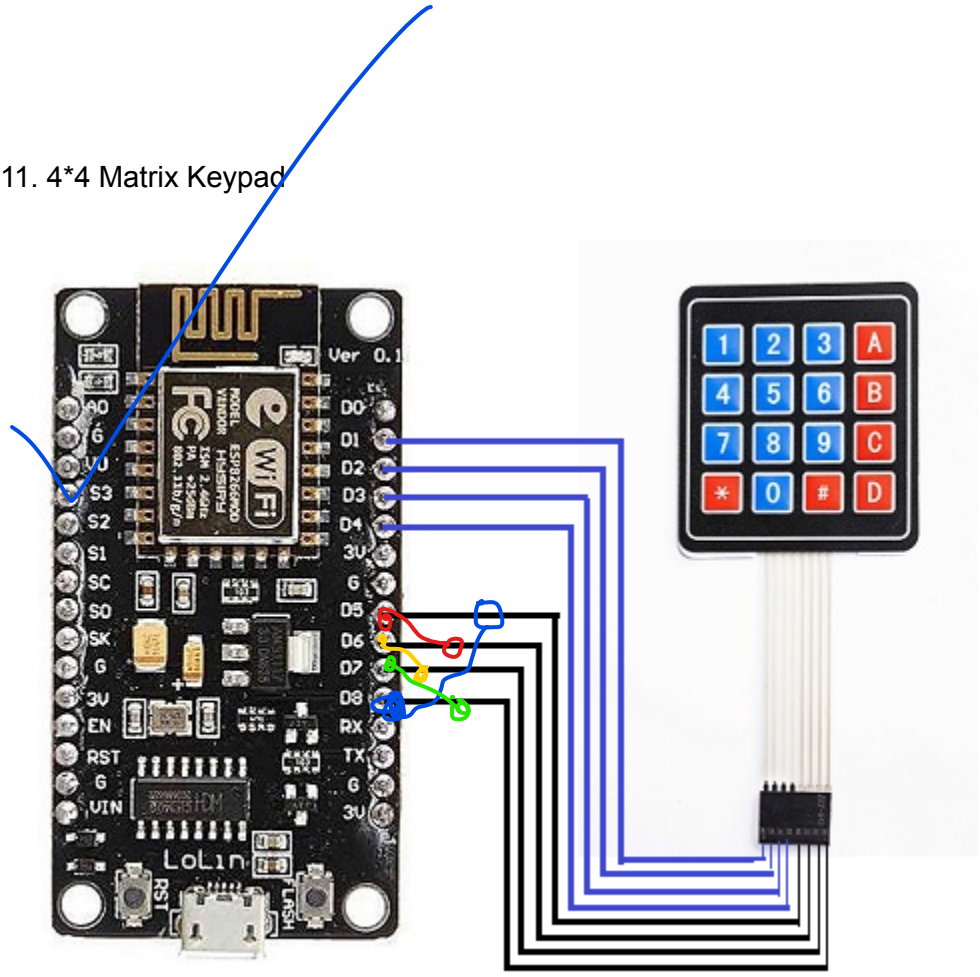
10. I2C LCD Display




Program

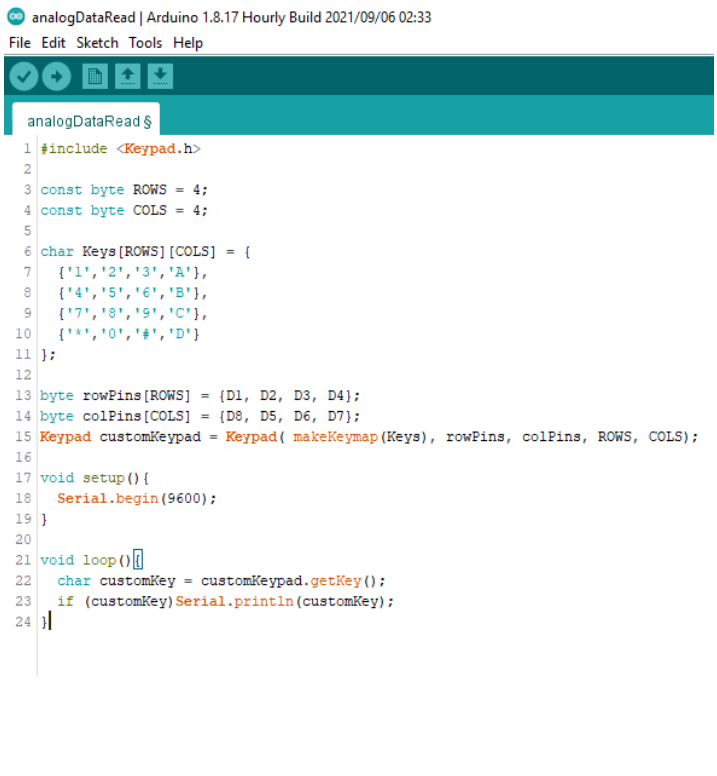
```
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);

void setup(){
lcd.init();
lcd.backlight();
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("LCD Interface");
}

void loop()
{

}
```

11. 4*4 Matrix Keypad

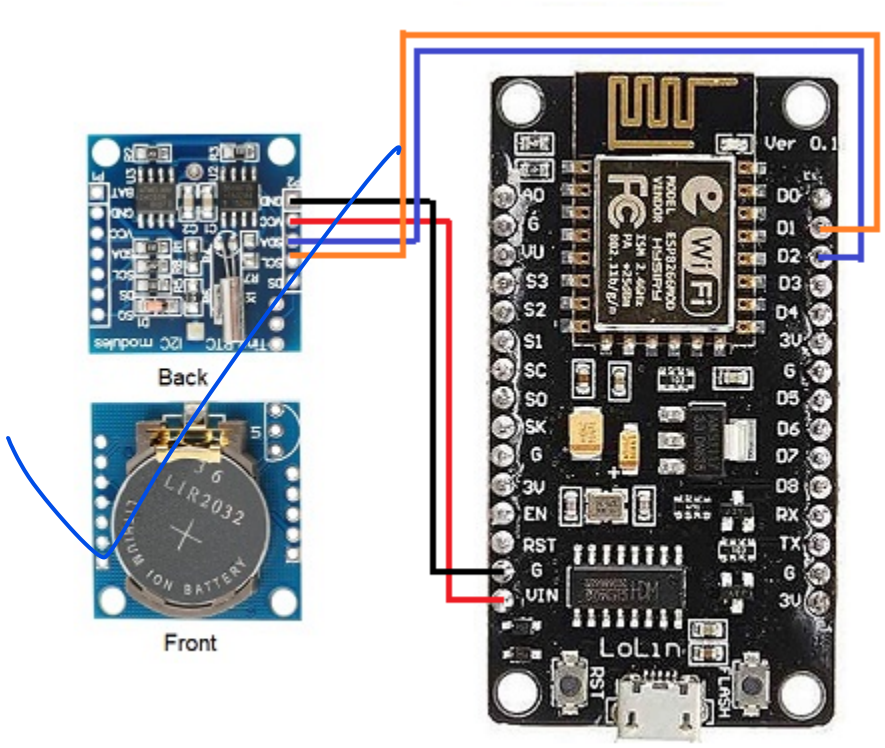



Program

```
#include <Keypad.h>

const byte ROWS = 4;
const byte COLS = 4;

char Keys[ROWS][COLS] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};

byte rowPins[ROWS] = {D1, D2, D3, D4};
byte colPins[COLS] = {D8, D5, D6, D7};
Keypad customKeypad = Keypad( makeKeymap(Keys),
rowPins, colPins, ROWS, COLS);

void setup(){
  Serial.begin(9600);
}

void loop(){
  char customKey = customKeypad.getKey();
  if (customKey)Serial.println(customKey);
}
```



12. RTC module

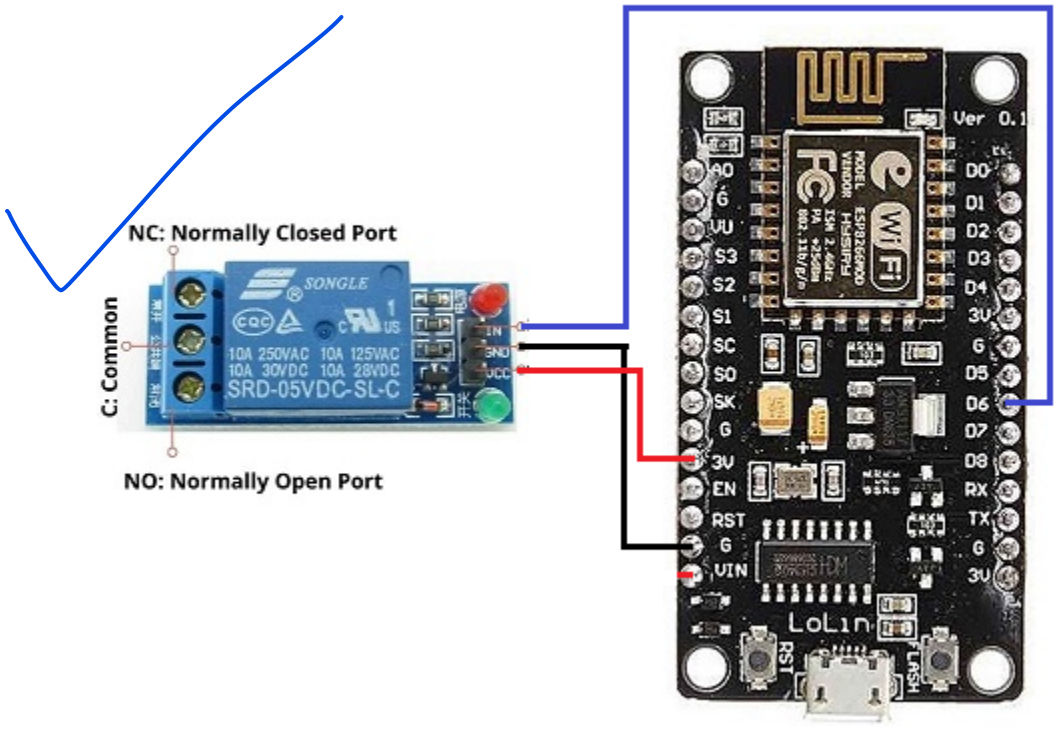



Program

```
#include <Wire.h>
#include "RTClib.h"

RTC_DS1307 rtc;
char Day[7][12] = {"Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"};

void setup()
{
 Serial.begin(9600);
 while(!rtc.begin());
 if (!rtc.isrunning())rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
}

void loop()
{
 DateTime now = rtc.now();
 Serial.print(String(now.year(), DEC)+'/'+String(now.month(), DEC)+'/'+String(now.day(), DEC));
 Serial.print(" ");
 Serial.print(String(now.hour(), DEC)+':'+String(now.minute(), DEC)+':'+String(now.second(), DEC));
 Serial.print(" ");
 Serial.print(Day[now.dayOfTheWeek()]);
 Serial.println();
}
```


analogDataRead | Arduino 1.8.17 Hourly Build 2021/09/06 02:33
File Edit Sketch Tools Help
Verify
analogDataRead §
#include <Wire.h>
#include "RTClib.h"
RTC_DS1307 rtc;
char Day[7][12] = {"Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"};
void setup()
{
Serial.begin(9600);
while(!rtc.begin());
if (!rtc.isrunning())rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
}
void loop()
{
DateTime now = rtc.now();
Serial.print(String(now.year(), DEC)+'/'+String(now.month(), DEC)+'/'+String(now.day(), DEC));
Serial.print(" ");
Serial.print(String(now.hour(), DEC)+':'+String(now.minute(), DEC)+':'+String(now.second(), DEC));
Serial.print(" ");
Serial.print(Day[now.dayOfTheWeek()]);
Serial.println();
}

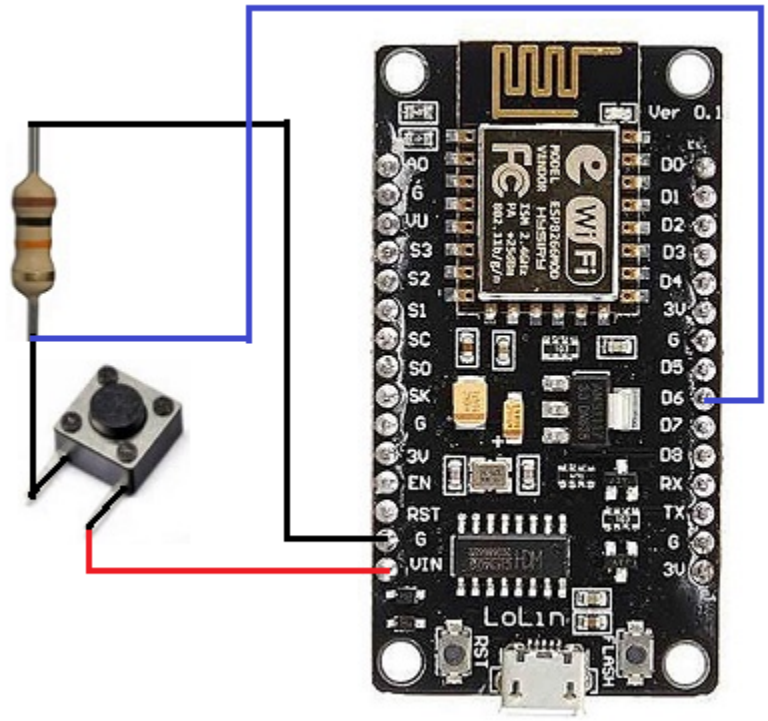
13. Relay Module




Program

```
void setup()
{
  pinMode(D6, OUTPUT);
  digitalWrite(D6, HIGH);
}
void loop()
{
  digitalWrite(D6,LOW);
  delay(2000);
  digitalWrite(D6,HIGH);
  delay(5000);
}
```

15. Push button interface




Program

```
void setup() {
  Serial.begin(9600);
  pinMode(D6, INPUT);
}

void loop() {
  int pinState = digitalRead(D6);
  Serial.println("pinState "+String(pinState));
  delay(200);
}
```

# CHAPTER 3

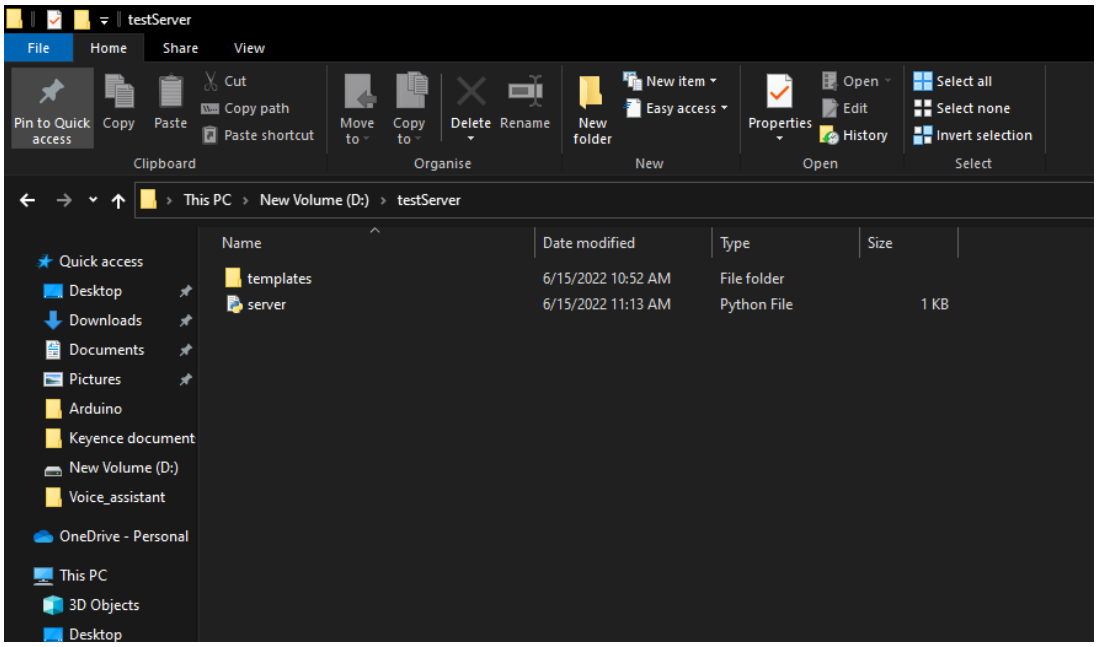
## Client Server Model HTTP

**Practice - Send a message from client to server & show the message in a webpage**

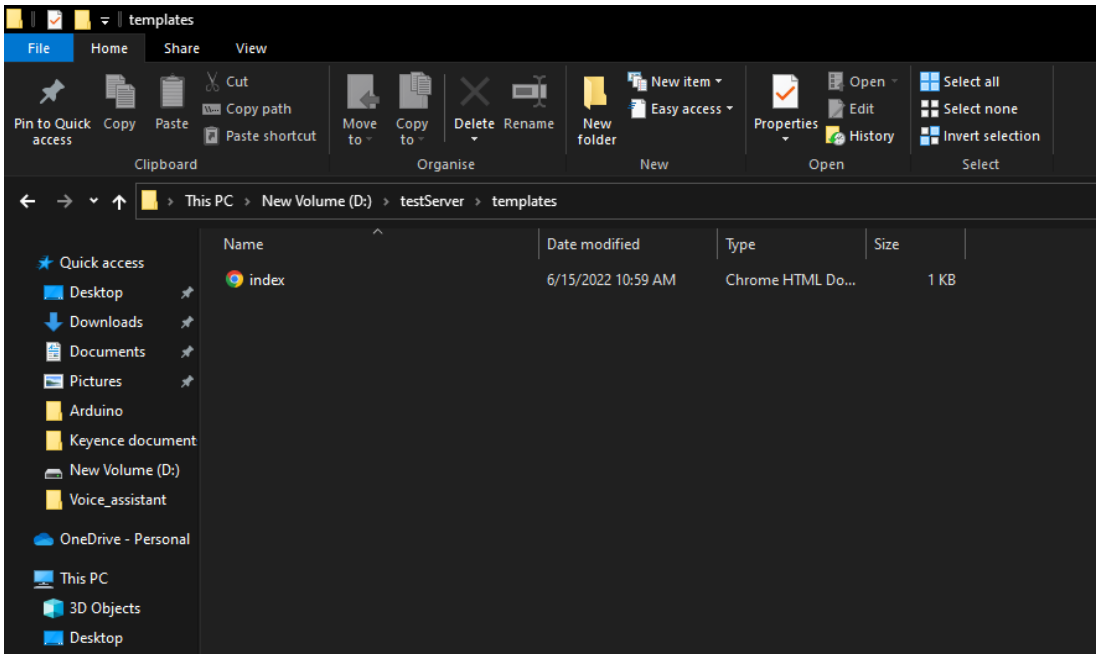
3 programs -

- Server
- Webpage
- Client

Server Project directory



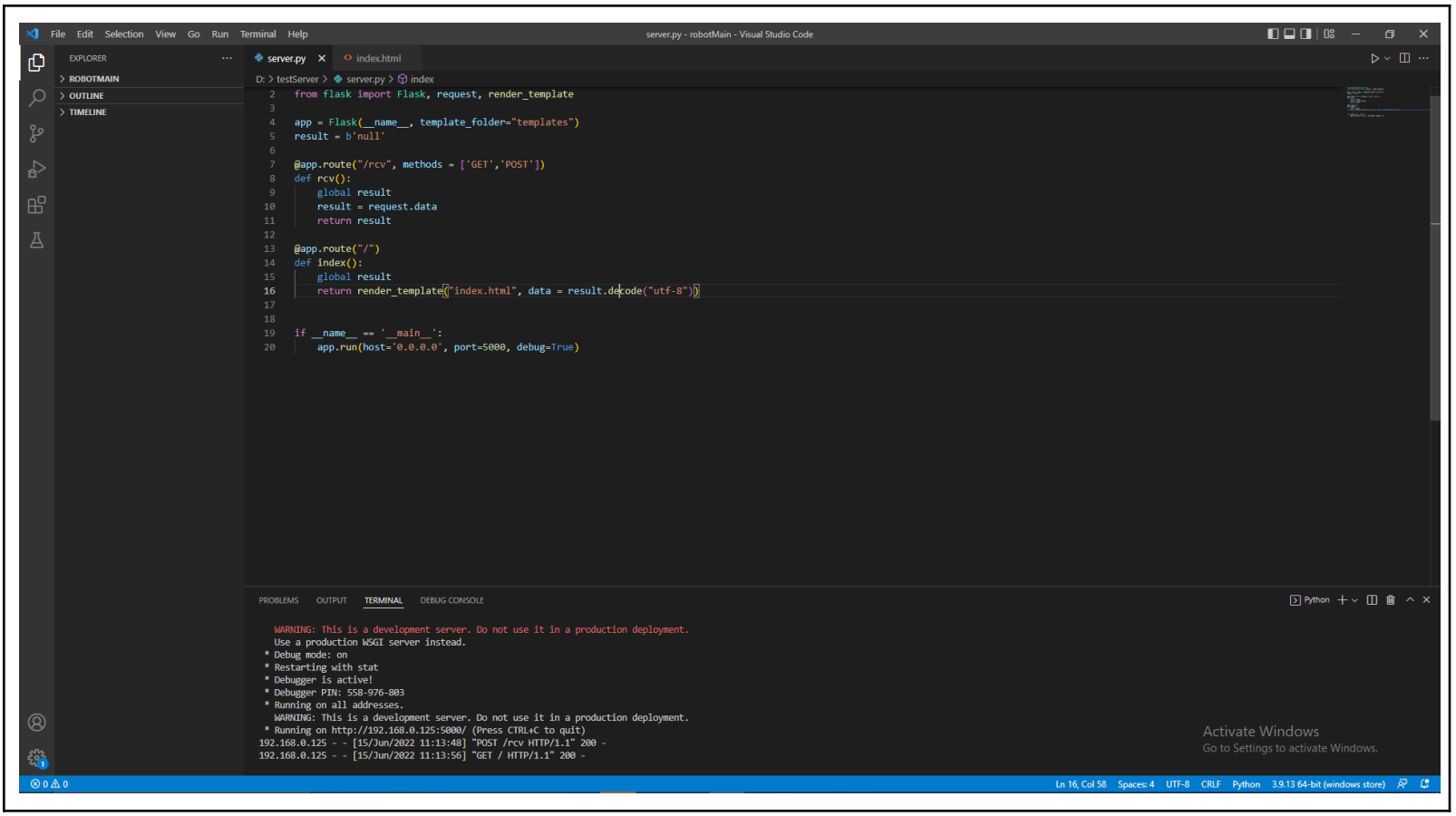
Webpage project directory




## Project programs

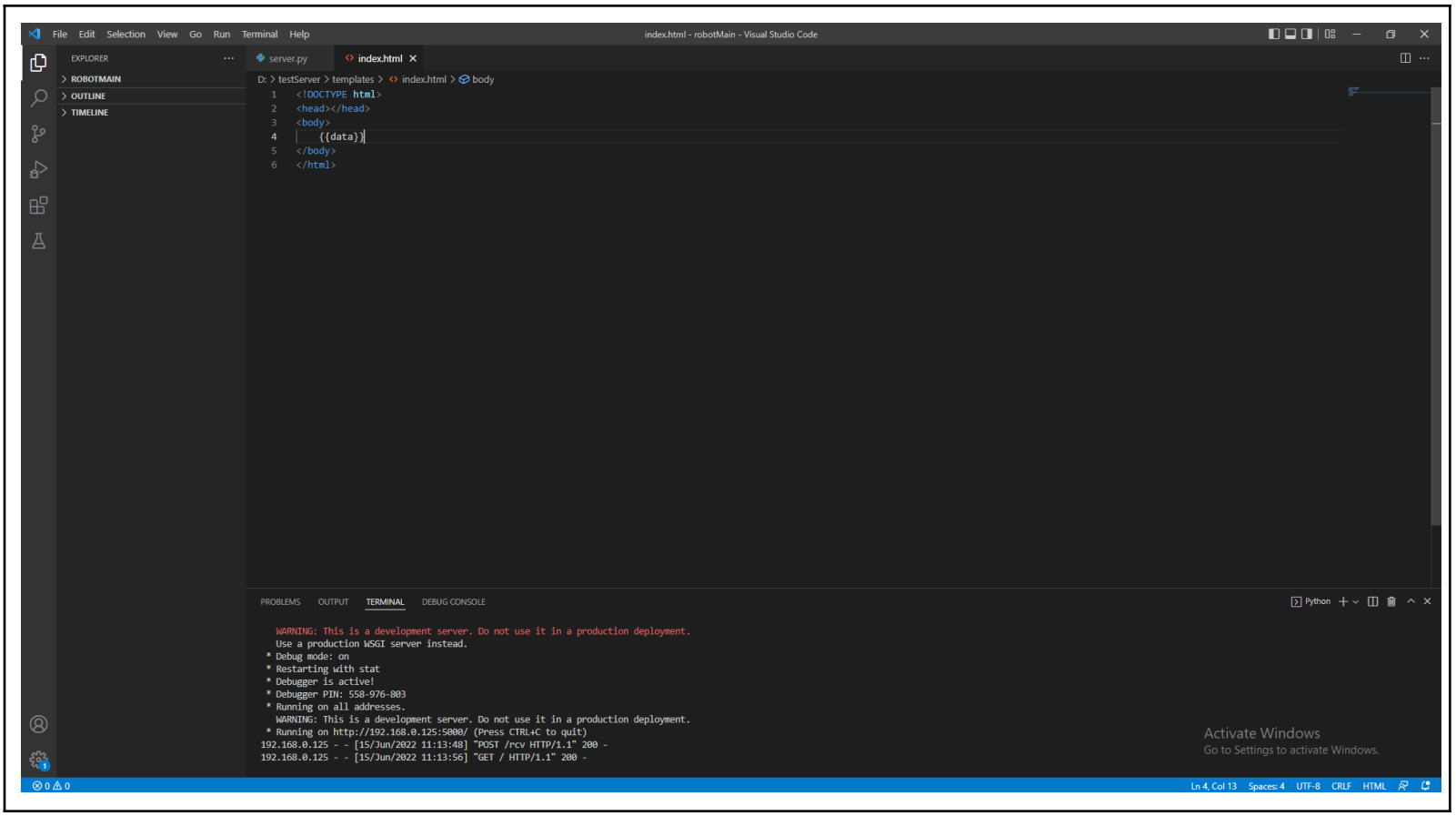
### server.py (server program)

```
from flask import Flask, request, render_template

app = Flask(__name__, template_folder="templates")
result = b'null'

@app.route("/rcv", methods = ['GET','POST'])
def rcv():
    global result
    result = request.data
    return result

@app.route("/")
def index():
    global result
    return render_template("index.html", data = result.decode("utf-8"))


if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000, debug=True)
```




**index.html (webpage program)**

```
<!DOCTYPE html>
<head></head>
<body>
    {{data}}
</body>
</html>
```


File Edit Selection View Go Run Terminal Help
index.html - robotMain - Visual Studio Code
EXPLORER
ROBOTMAIN
OUTLINE
TIMELINE
server.py
index.html
D: > testServer > templates > index.html > body
<!DOCTYPE html>
<head></head>
<body>
{{data}}
</body>
</html>
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
Python
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: on
* Restarting with stat
* Debugger is active!
* Debugger PIN: 558-976-803
* Running on all addresses.
WARNING: This is a development server. Do not use it in a production deployment.
* Running on http://192.168.0.125:5000/ (Press CTRL+C to quit)
192.168.0.125 - - [15/Jun/2022 11:13:48] "POST /rcv HTTP/1.1" 200 -
192.168.0.125 - - [15/Jun/2022 11:13:56] "GET / HTTP/1.1" 200 -
Activate Windows
Go to Settings to activate Windows.
Ln 4, Col 13 Spaces: 4 UTF-8 CRLF HTML

**Client = ESP8266 Node MCU**

```
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClient.h>

const char* ssid = "ABRAR2";
const char* password = "abrar96009";
const char* server = "192.168.43.2";

int httpResponseCode = 0;
WiFiClient clientOne;
HTTPClient http;

void setup()
{
  Serial.begin(9600);
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println();
  Serial.println("WiFi connected");
  Serial.println(WiFi.localIP());
  int httpStatus = http.begin(clientOne, server, 5000, "/rcv");
}

void loop()
{
  httpResponseCode = http.POST("UGV");
  Serial.println(httpResponseCode);
  if (httpResponseCode > 0)
  {
    String payload = http.getString();
    Serial.println(payload);
  }
  else Serial.println("Error occurred");
  delay(5000);
}
```

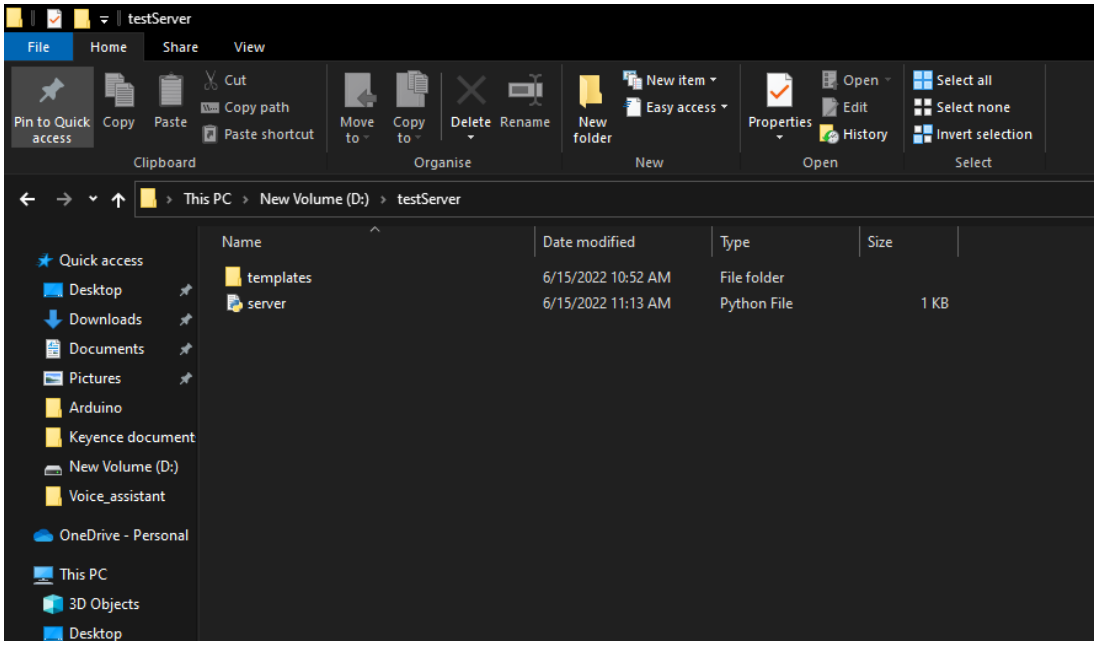

analogDataRead | Arduino 1.8.17 Hourly Build 2021/09/06 02:33
File Edit Sketch Tools Help
analogDataRead
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClient.h>
const char* ssid = "ABRAR2";
const char* password = "abrar96009";
int httpResponseCode = 0;
WiFiClient clientOne;
HTTPClient http;
void setup()
{
Serial.begin(9600);
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
delay(500);
Serial.print(".");
}
Serial.println();
Serial.println("WiFi connected");
Serial.println(WiFi.localIP());
int httpStatus = http.begin(clientOne, "192.168.43.2", 5000, "/rcv");
}
void loop()
{
httpResponseCode = http.POST("UGV");
Hard resetting via RTS pin...
Invalid library found in C:\Users\MTI-PC\Documents\Arduino\libraries\freertos: no headers
dule), 80 MHz, Flash, Disabled (new aborts on oom), Disabled, All SSL ciphers (most compatible), 32KB cache + 32KB IRAM (balan

# Practice - Connect RFID with ESP8266. Send card ID from client to server. Then show the card ID into a webpage

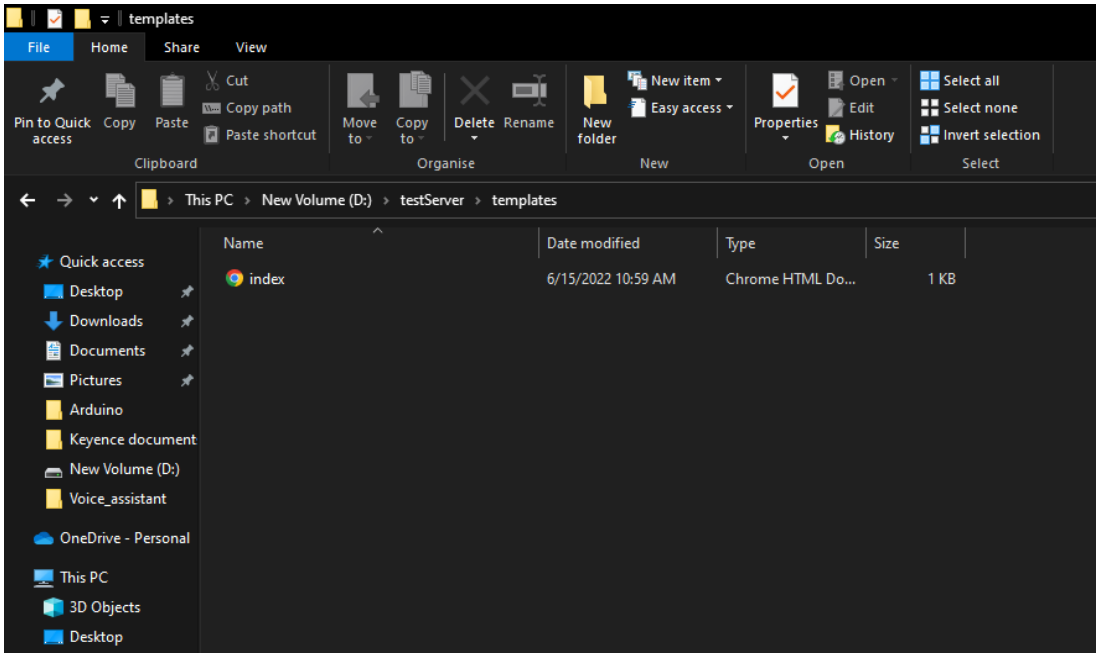
3 programs -

- Server
- Webpage
- Client

Server Project directory



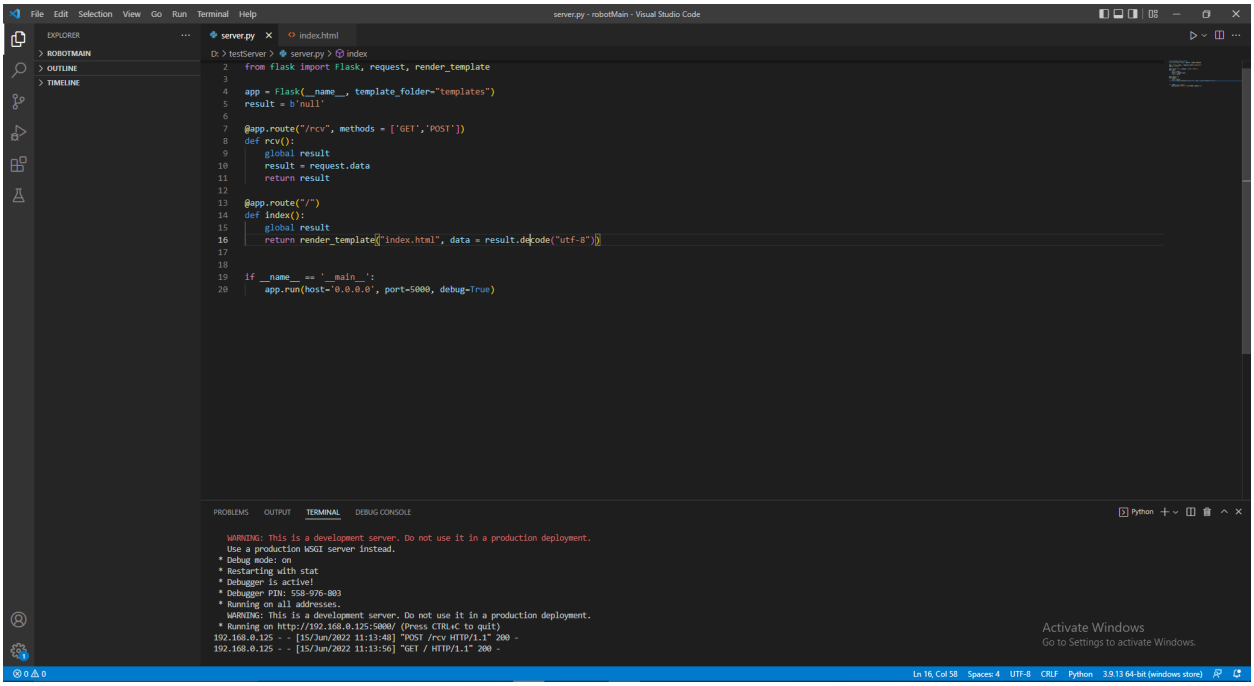
Webpage project directory




## Project programs

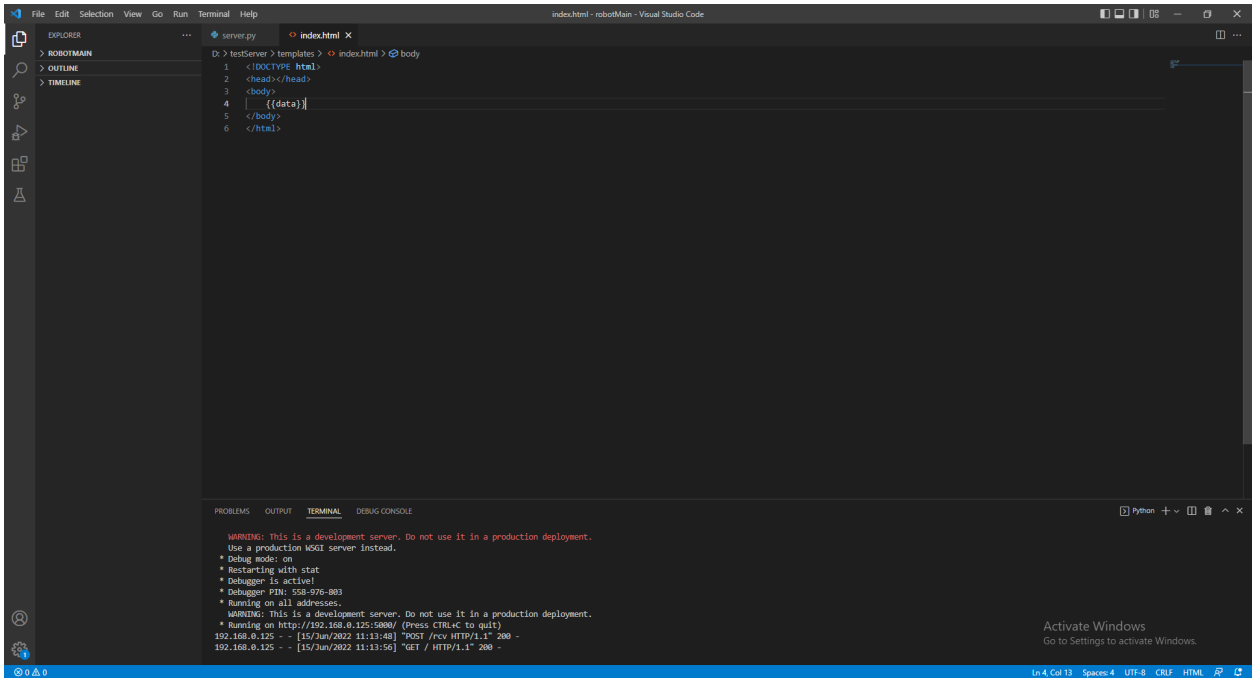
### server.py (server program)

```
from flask import Flask, request, render_template

app = Flask(__name__, template_folder="templates")
result = b'null'

@app.route("/rcv", methods = ['GET','POST'])
def rcv():
    global result
    result = request.data
    return result

@app.route("/")
def index():
    global result
    return render_template("index.html", data = result.decode("utf-8"))


if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000, debug=True)
```




## index.html (webpage program)

```
<!DOCTYPE html>
<head></head>
<body>
    {{data}}
</body>
</html>
```


File Edit Selection View Go Run Terminal Help
index.html - robotMain - Visual Studio Code
EXPLORER
ROBOTMAIN
OUTLINE
TIMELINE
server.py
index.html
D: > testServer > templates > index.html > body
<!DOCTYPE html>
<head></head>
<body>
{{data}}
</body>
</html>
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
Python
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: on
* Restarting with stat
* Debugger is active!
* Debugger PIN: 558-976-803
* Running on all addresses.
WARNING: This is a development server. Do not use it in a production deployment.
* Running on http://192.168.0.125:5000/ (Press CTRL+C to quit)
192.168.0.125 - - [15/Jun/2022 11:13:48] "POST /rcv HTTP/1.1" 200 -
192.168.0.125 - - [15/Jun/2022 11:13:56] "GET / HTTP/1.1" 200 -
Activate Windows
Go to Settings to activate Windows.
Ln 4, Col 13 Spaces: 4 UTF-8 CRLF HTML

**Client = ESP8266 Node MCU**

```
#include <rdm6300.h>
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClient.h>
#include <String.h>

#define RDM6300_RX_PIN D6
Rdm6300 rdm6300;

const char* ssid = "ABRAR";
const char* password = "abrar960";
const char* server = "192.168.43.2";

char cardNo[15];
int32_t cardID = 0;
int httpResponseCode = 0;
WiFiClient clientOne;
HTTPClient http;

void setup()
{
  Serial.begin(9600);
  rdm6300.begin(RDM6300_RX_PIN);
  WiFi.begin(ssid, password);

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

  Serial.println();
  Serial.println("WiFi connected");
  Serial.println(WiFi.localIP());
  int httpStatus = http.begin(clientOne, server, 5000, "/rfidVerify");
}

void loop()
{
  if (rdm6300.is_listening())cardID = rdm6300.get_tag_id();
  if (cardID != 0)
  {
    sprintf(cardNo, "%d", cardID);
    Serial.println(cardNo);
    httpResponseCode = http.POST(cardNo);

    Serial.println(httpResponseCode);
    if (httpResponseCode > 0)
    {
      String payload = http.getString();
```

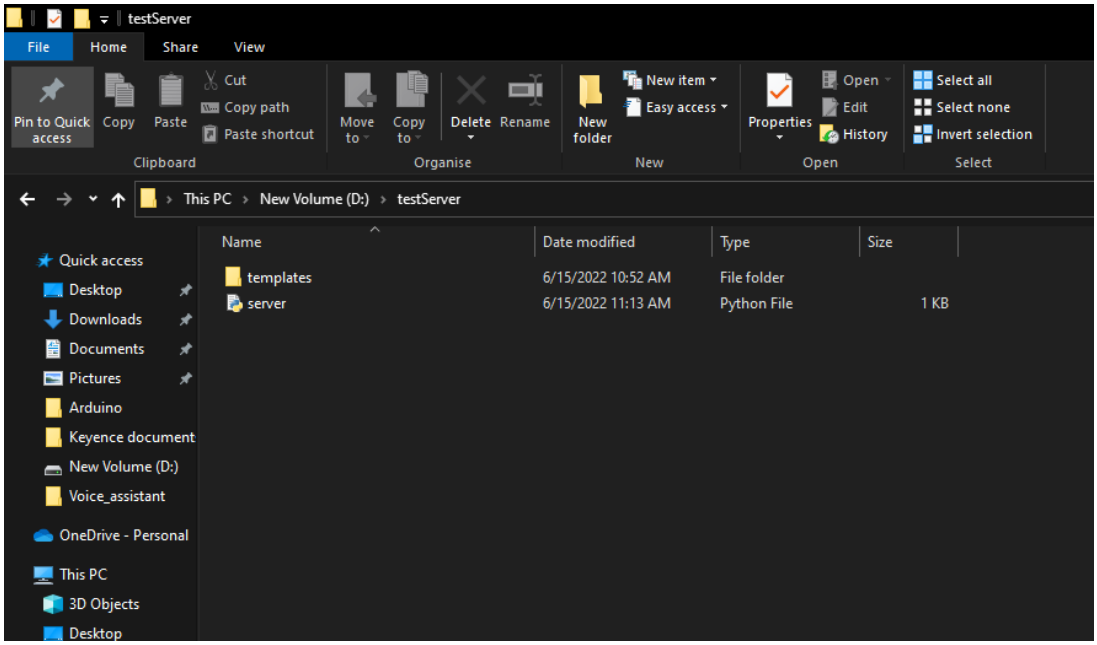
```
    Serial.println(payload);

   }
   else Serial.println("Error occurred");
  }
  delay(1000);
}
```

# Practice - Connect LM35 with ESP8266. Send data from client to server. Then show the data into a webpage

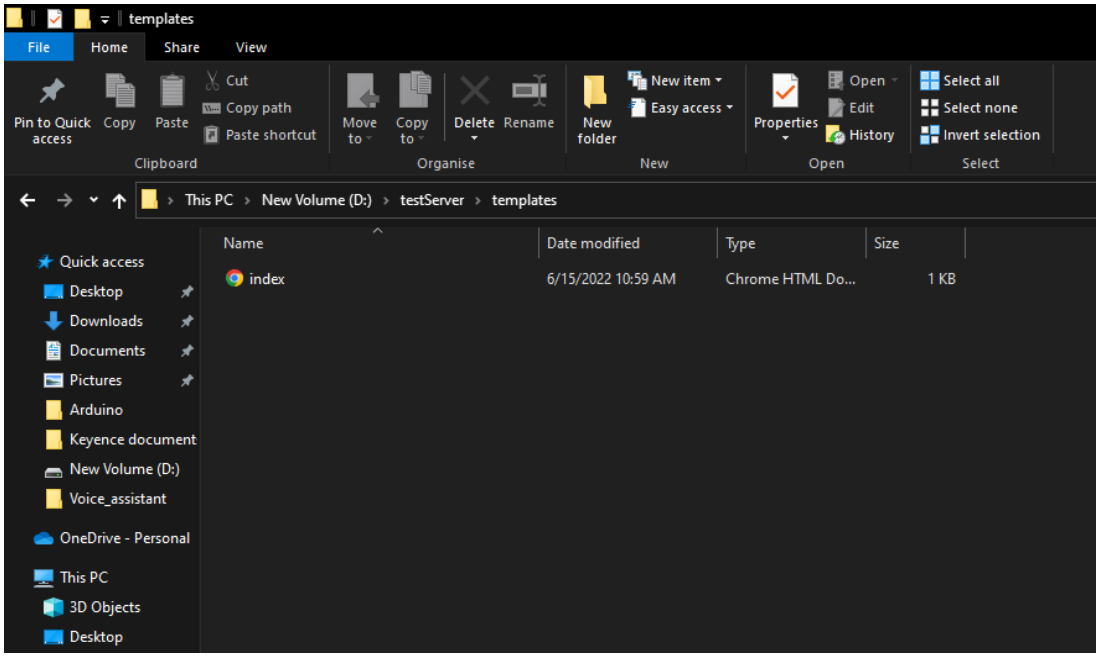
3 programs -

- Server
- Webpage
- Client

Server Project directory



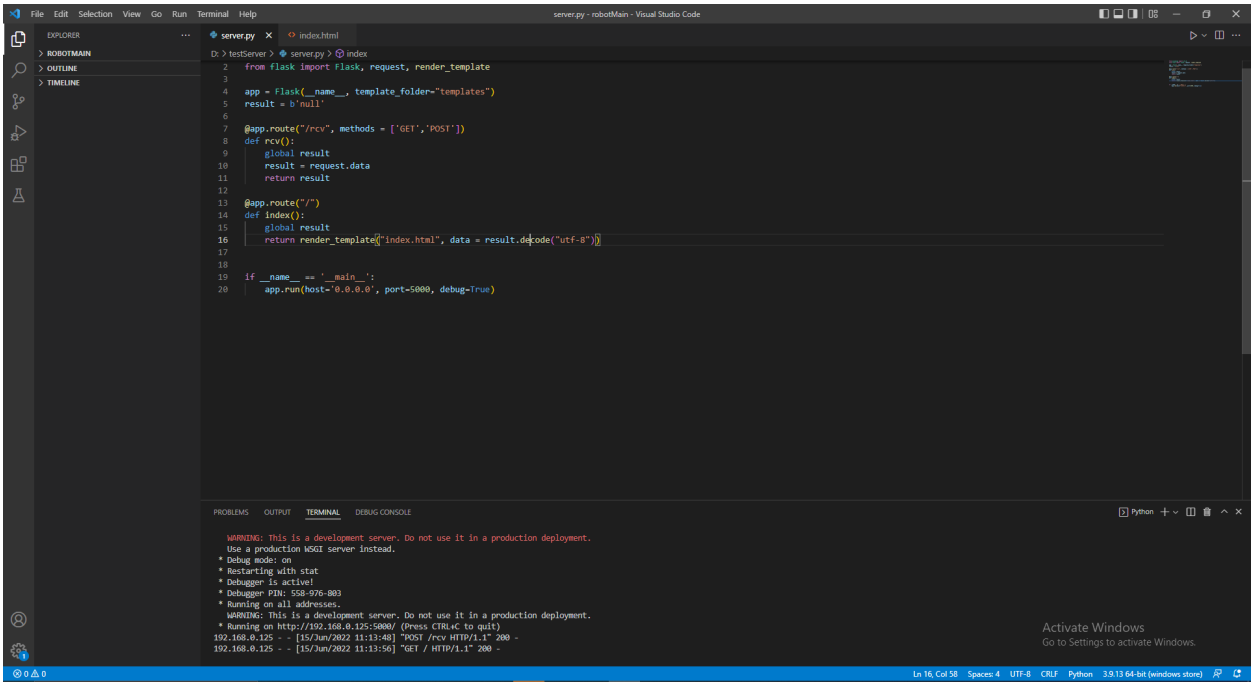
Webpage project directory




## Project programs

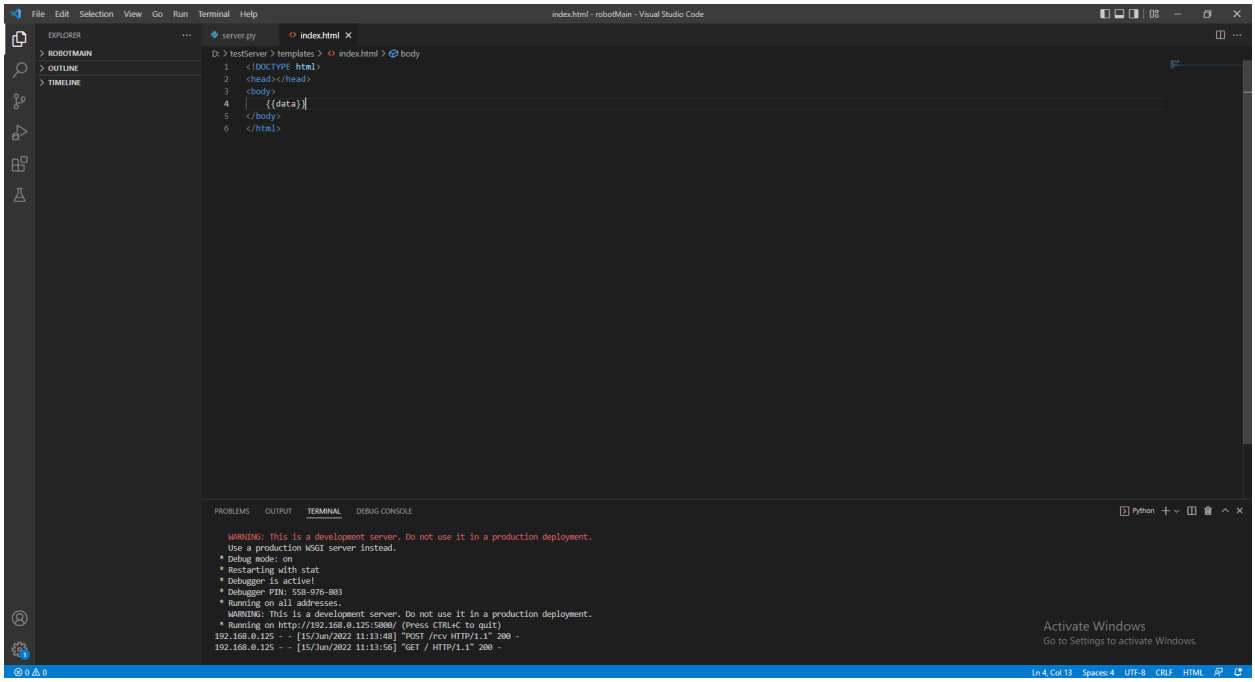
### server.py (server program)

```
from flask import Flask, request, render_template

app = Flask(__name__, template_folder="templates")
result = b'null'

@app.route("/rcv", methods = ['GET','POST'])
def rcv():
    global result
    result = request.data
    return result

@app.route("/")
def index():
    global result
    return render_template("index.html", data = result.decode("utf-8"))


if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000, debug=True)
```




## index.html (webpage program)

```
<!DOCTYPE html>
<head></head>
<body>
    {{data}}
</body>
</html>
```


File Edit Selection View Go Run Terminal Help
index.html - robotMain - Visual Studio Code
EXPLORER
ROBOTMAIN
OUTLINE
TIMELINE
server.py
index.html
D: > testServer > templates > index.html > body
<!DOCTYPE html>
<head></head>
<body>
{{data}}
</body>
</html>
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
Python
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: on
* Restarting with stat
* Debugger is active!
* Debugger PIN: 558-976-803
* Running on all addresses.
WARNING: This is a development server. Do not use it in a production deployment.
* Running on http://192.168.0.125:5000/ (Press CTRL+C to quit)
192.168.0.125 - - [15/Jun/2022 11:13:48] "POST /rcv HTTP/1.1" 200 -
192.168.0.125 - - [15/Jun/2022 11:13:56] "GET / HTTP/1.1" 200 -
Activate Windows
Go to Settings to activate Windows.
Ln 4, Col 13 Spaces: 4 UTF-8 CRLF HTML

Client program

```
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClient.h>
#include <String.h>

const char* ssid = "ABRAR2";
const char* password = "abrar96009";
const char* server = "192.168.43.2";

int httpResponseCode = 0;
WiFiClient clientOne;
HTTPClient http;

void setup()
{
  Serial.begin(9600);
  pinMode(A0, INPUT);
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println();
  Serial.println("WiFi connected");
  Serial.println(WiFi.localIP());
  int httpStatus = http.begin(clientOne, server, 5000, "/rcv");
}

void loop()
{
  char result[10];
  int data = analogRead(A0);
  sprintf(result,"%d",data);
  httpResponseCode = http.POST(result);
  Serial.println(httpResponseCode);
  if (httpResponseCode > 0)
  {
    String payload = http.getString();
    Serial.println(payload);
  }
  else Serial.println("Error occurred");
  delay(5000);
}
```


LM35Client | Arduino 1.8.17 Hourly Build 2021/09/06 02:33
File Edit Sketch Tools Help
LM35Client
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClient.h>
#include <String.h>
const char* ssid = "ABRAR2";
const char* password = "abrar96009";
const char* server = "192.168.43.2";
int httpResponseCode = 0;
WiFiClient clientOne;
HTTPClient http;
void setup()
{
Serial.begin(9600);
pinMode(A0, INPUT);
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
delay(500);
Serial.print(".");
}
Serial.println();
Serial.println("WiFi connected");
Serial.println(WiFi.localIP());
int httpStatus = http.begin(clientOne, server, 5000, "/rcv");
}
Done compiling.
Sketch uses 283505 bytes (27%) of program storage space. Maximum is 1044464 bytes.
Global variables use 28368 bytes (34%) of dynamic memory, leaving 53552 bytes for loca
Module), 80 MHz, Flash, Disabled (new aborts on oom), Disabled, All SSL ciphers (most compatible), 32KB cache + 32KB IRAM

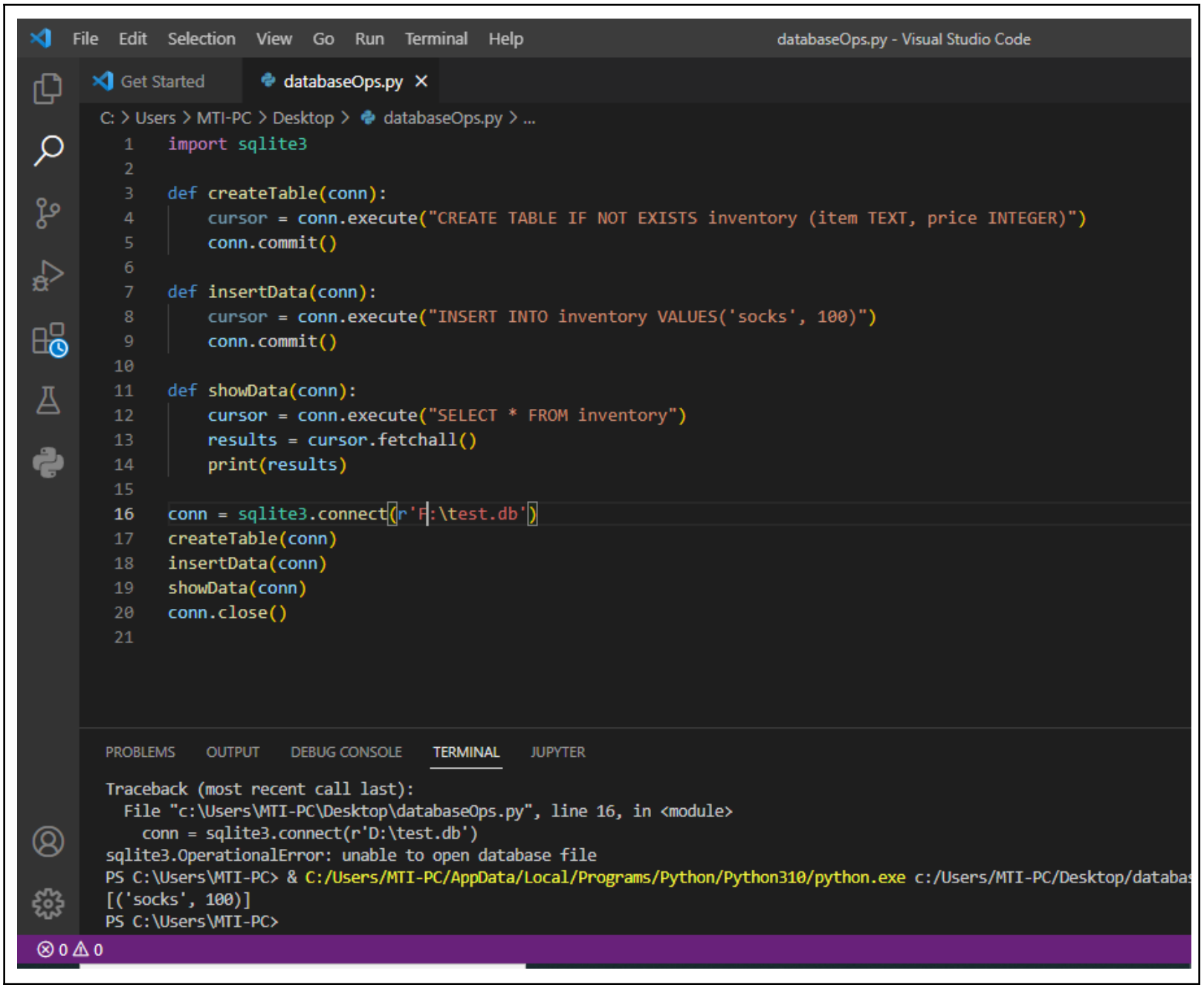
# CHAPTER 4

Database Tasks -

- Create database
- Create table
- Insert data
- Show data

Sample Program

```
import sqlite3

def createTable(conn):
    cursor = conn.execute("CREATE TABLE IF NOT EXISTS inventory (item TEXT, price INTEGER)")
    conn.commit()

def insertData(conn):
    cursor = conn.execute("INSERT INTO inventory VALUES('socks', 100)")
    conn.commit()

def showData(conn):
    cursor = conn.execute("SELECT * FROM inventory")
    results = cursor.fetchall()
    print(results)

conn = sqlite3.connect(r'D:\test.db')
createTable(conn)
insertData(conn)
showData(conn)
conn.close()
```


File Edit Selection View Go Run Terminal Help
databaseOps.py - Visual Studio Code
Get Started
databaseOps.py
C: > Users > MTI-PC > Desktop > databaseOps.py > ...
1 import sqlite3
2
3 def createTable(conn):
4 cursor = conn.execute("CREATE TABLE IF NOT EXISTS inventory (item TEXT, price INTEGER)")
5 conn.commit()
6
7 def insertData(conn):
8 cursor = conn.execute("INSERT INTO inventory VALUES('socks', 100)")
9 conn.commit()
10
11 def showData(conn):
12 cursor = conn.execute("SELECT * FROM inventory")
13 results = cursor.fetchall()
14 print(results)
15
16 conn = sqlite3.connect(r'D:\test.db')
17 createTable(conn)
18 insertData(conn)
19 showData(conn)
20 conn.close()
21
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
Traceback (most recent call last):
File "c:\Users\MTI-PC\Desktop\databaseOps.py", line 16, in <module>
conn = sqlite3.connect(r'D:\test.db')
sqlite3.OperationalError: unable to open database file
PS C:\Users\MTI-PC> & C:/Users/MTI-PC/AppData/Local/Programs/Python/Python310/python.exe c:/Users/MTI-PC/Desktop/databas
[('socks', 100)]
PS C:\Users\MTI-PC>
0 0