Smart Plant Monitoring System

Group Members:

- 23-ntu-cs-1257: Laiba Fatima
- 23-ntu-cs-1259: Mahnoor Sajjad

Course: BSAI 5th

Date: 1-Jan-2026

1. Project Overview

The **Smart Plant Monitoring System** is an IoT-based system designed to **monitor and care for plants automatically and manually**. It measures environmental parameters like **temperature, humidity, and soil moisture**, displays them on an **OLED screen**, and can **automatically water plants** when needed.

It also sends real-time data to:

- **Blynk mobile app** → Remote monitoring and manual control.
- **Node-RED dashboard** → Web-based visualization using MQTT protocol.

2. Main Features:

- Real-time sensor monitoring.
- Automatic irrigation based on soil moisture.
- Remote monitoring via Blynk and Node-RED.
- Manual control of the motor through Blynk.
- Integration with MQTT for IoT communication.

3. Hardware Components:

- ESP32 Dev Board
- DHT11 Sensor (Temperature & Humidity)
- Soil Moisture Sensor

- Relay Module
- Water Pump
- OLED Display (128x64)
- Jumper Wires
- Power Supply (7.4V)

4. Pin Configuration:

Component	ESP32 Pin	Mode
DHT11	27	Digital Input
Soil Moisture Sensor	34	Analog Input
Relay	26	Digital Output
OLED Display SDA/SCL	21 / 22	I2C

5. Software and Libraries

- VSCode
- PlatformIO IDE
- Blynk / Blynk Library
- Adafruit SSD1306
- Adafruit GFX Library
- DHT Sensor Library
- PubSubClient

6. System Workflow

1. ESP32 Starts

- Connects to WiFi.
- Connects to **Blynk app** and **MQTT broker**.
- Initializes sensors and OLED display.

2. Read Sensors

- DHT11 measures temperature and humidity.
- Soil sensor measures soil moisture.

3. Display on OLED

- Shows temperature, humidity, soil value, soil status (DRY/WET), and motor state.

4. Automatic Watering

- If soil < threshold → Motor turns ON → Water pump starts.
- If soil ≥ threshold → Motor turns OFF → Pump stops.
- Updates motor state in **Blynk app** automatically.

5. Send Data to Blynk

- Virtual pins:
  - V0 → Soil moisture
  - V1 → Temperature
  - V2 → Humidity
  - V3 → Motor ON/OFF switch
  - V4 → Soil status LED
- Users can manually control motor using Blynk.

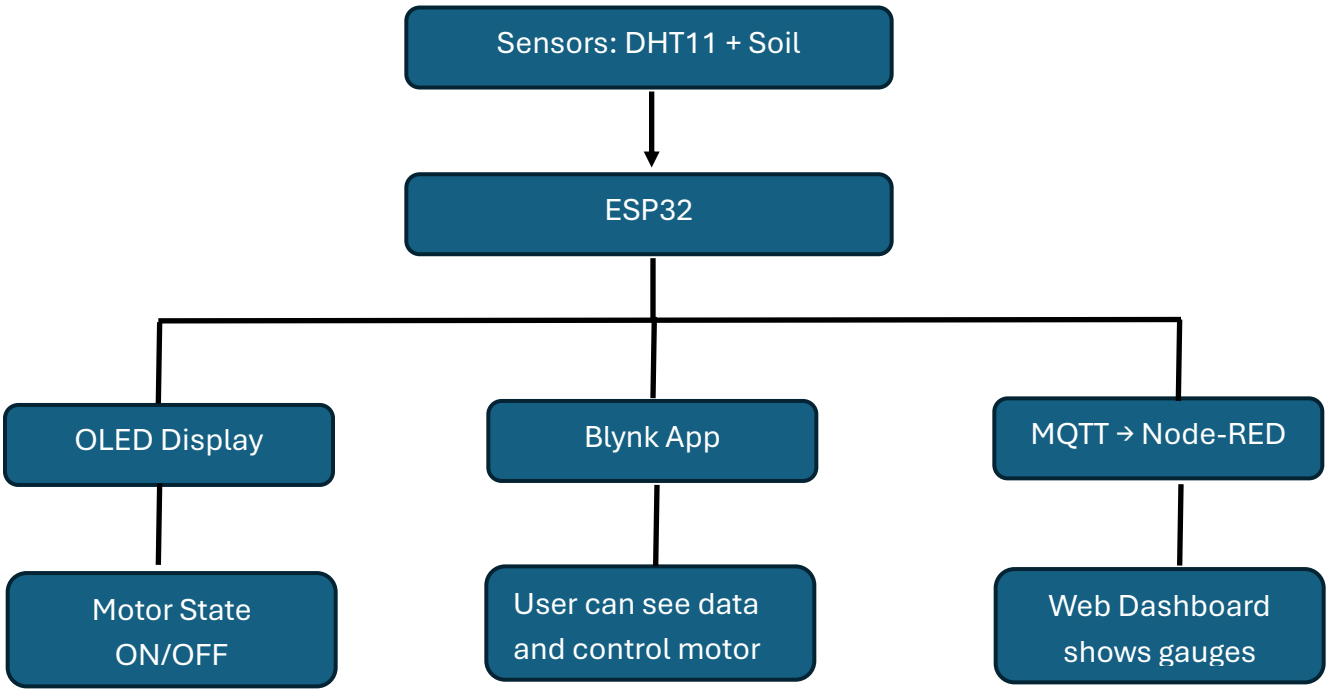
6. Send Data to Node-RED via MQTT

- ESP32 publishes to MQTT topics:
  - smartplant/temp
  - smartplant/hum
  - smartplant/soil
  - smartplant/motor
- Node-RED visualizes the data in **gauges, and LED**.

7. Repeat Automatically

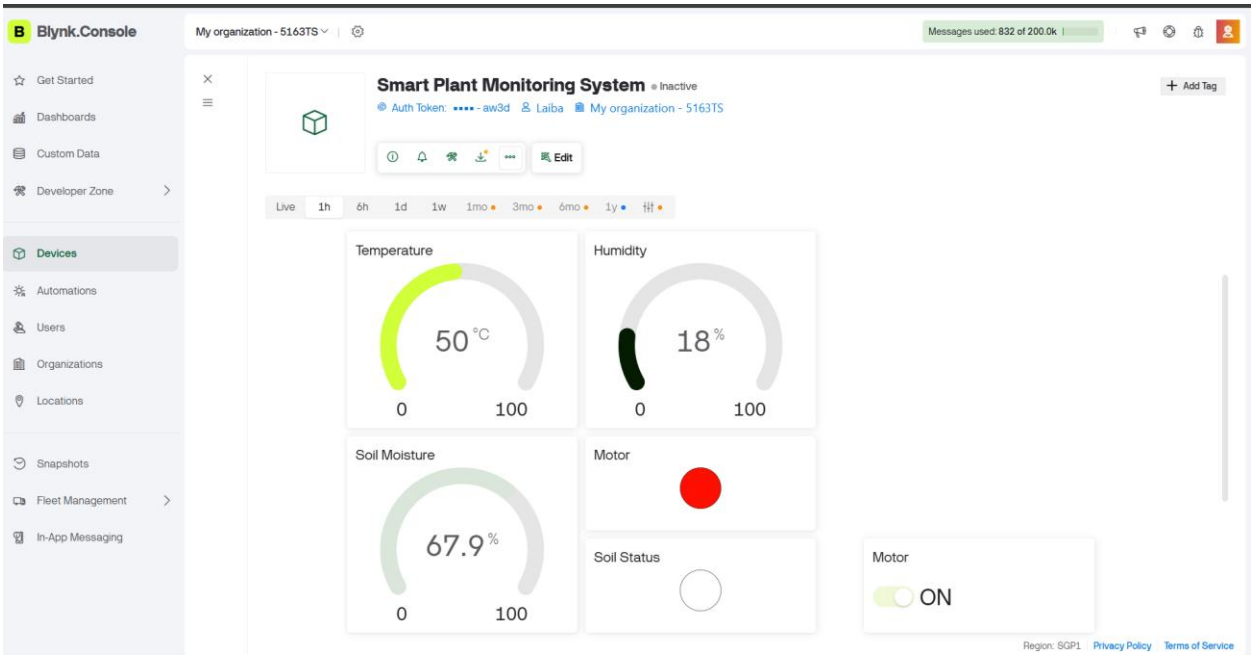
- Every 3 seconds, ESP32 reads sensors, updates displays, and checks soil for watering.

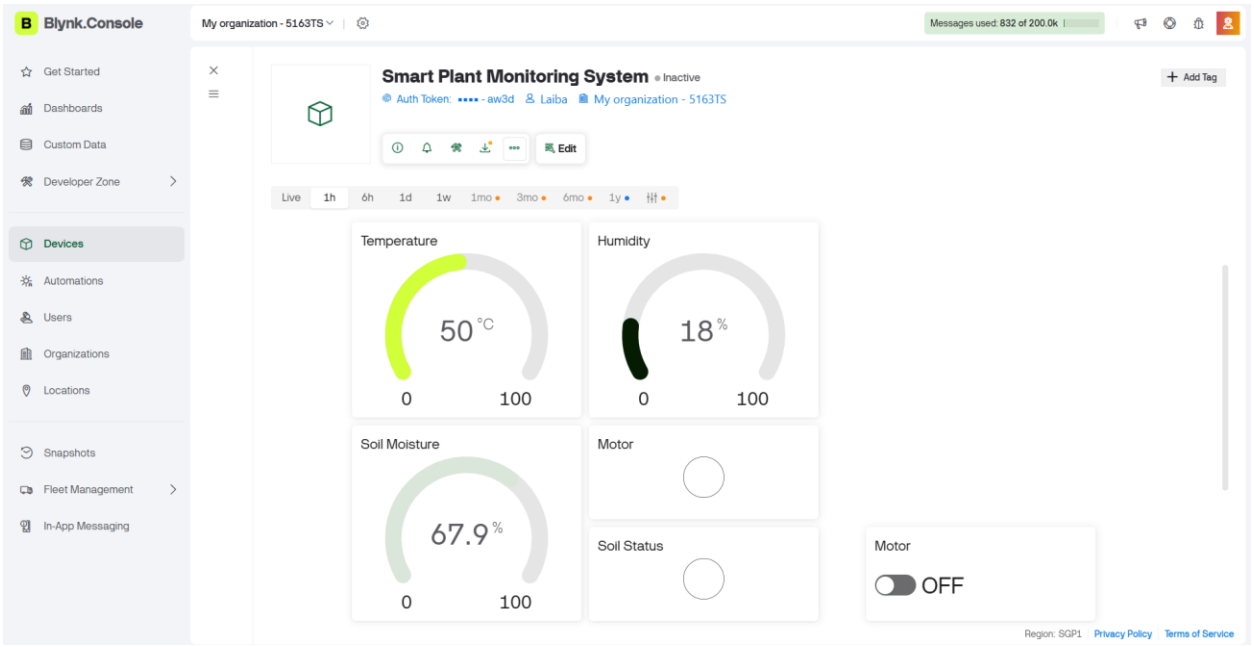
7. Flowchart of Workflow:

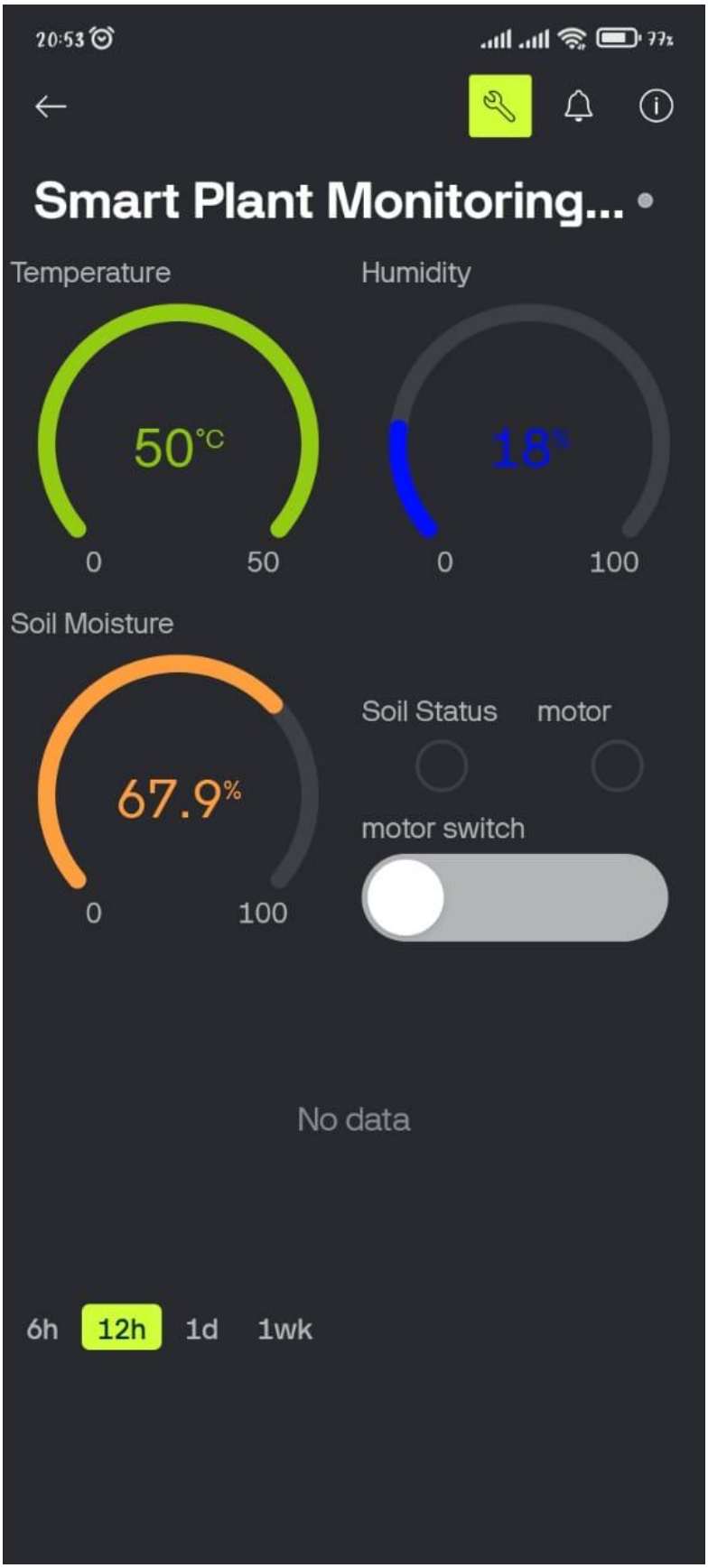


8. Blynk App Integration

- Mobile app for remote monitoring.
- Displays sensor readings and motor state in **real-time**.
- Allows **manual motor control**.
- Virtual pins mapping:
    - V0 → Temperature
    - V1 → Humidity
    - V2 → Soil Moisture
    - V3 → Motor ON/OFF switch
    - V4 → Soil Status LED


Blynk.Console
My organization - 5163TS
Messages used: 832 of 200.0k
Get Started
Dashboards
Custom Data
Developer Zone
Devices
Automations
Users
Organizations
Locations
Snapshots
Fleet Management
In-App Messaging
Smart Plant Monitoring System
Inactive
Auth Token: •••• - aw3d
Laiba
My organization - 5163TS
Add Tag
Edit
Live
1h
6h
1d
1w
1mo
3mo
6mo
1y
Temperature
50 °C
0
100
Humidity
18 %
0
100
Soil Moisture
67.9 %
0
100
Motor
Soil Status
Motor
ON
Region: SGP1
Privacy Policy
Terms of Service

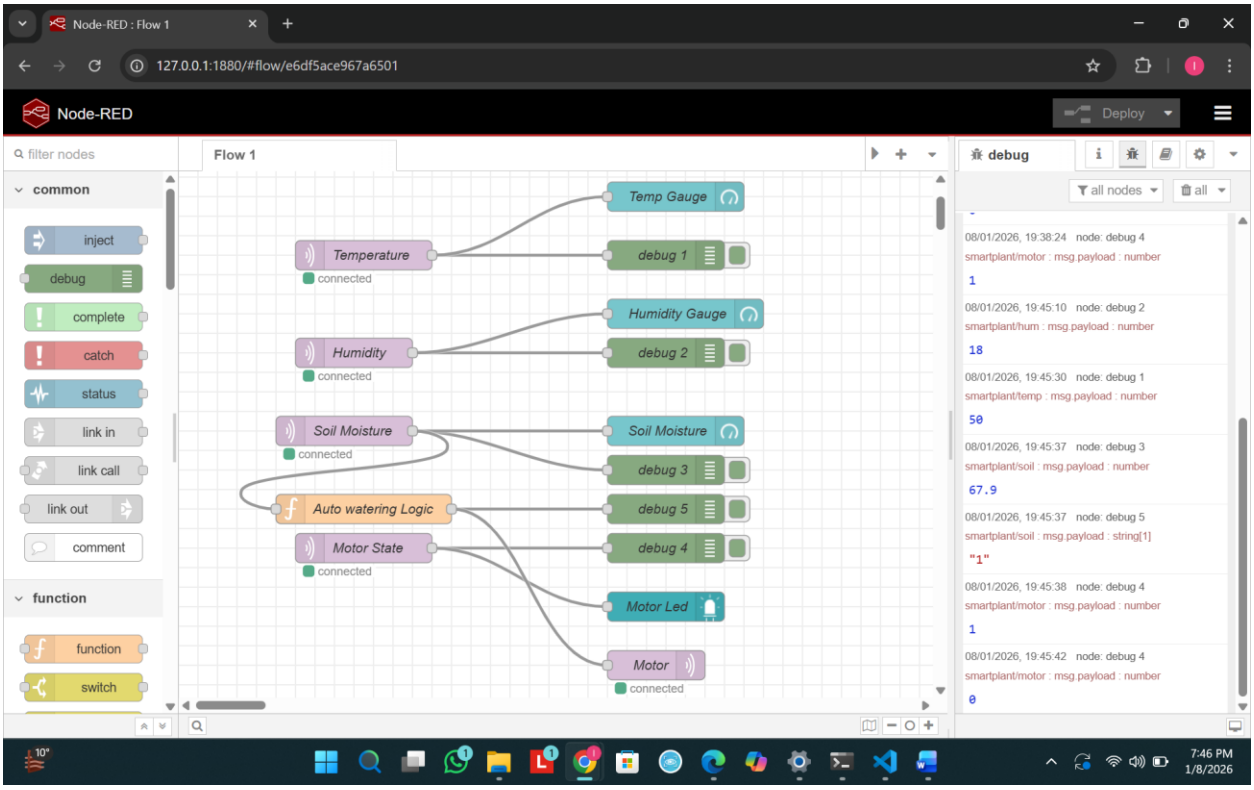

Blynk.Console
My organization - 5163TS
Messages used: 832 of 200.0k
Get Started
Dashboards
Custom Data
Developer Zone
Devices
Automations
Users
Organizations
Locations
Snapshots
Fleet Management
In-App Messaging
Smart Plant Monitoring System
Inactive
Auth Token: •••• - aw3d
Laiba
My organization - 5163TS
Add Tag
Edit
Live
1h
6h
1d
1w
1mo
3mo
6mo
1y
Temperature
50 °C
0
100
Humidity
18 %
0
100
Soil Moisture
67.9 %
0
100
Motor
Soil Status
Motor
OFF
Region: SGP1
Privacy Policy
Terms of Service

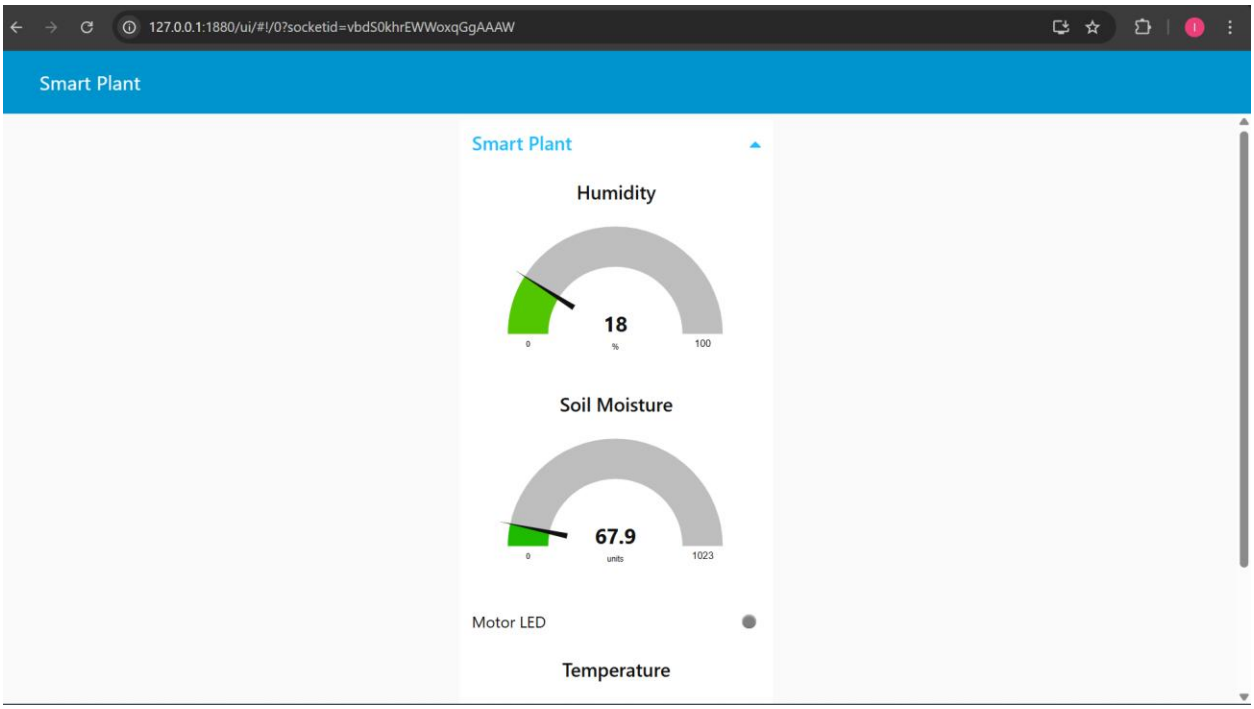

20:53
77%
Smart Plant Monitoring...
Temperature
50°C
0
50
Humidity
18%
0
100
Soil Moisture
67.9%
0
100
Soil Status
motor
motor switch
No data
6h
12h
1d
1wk

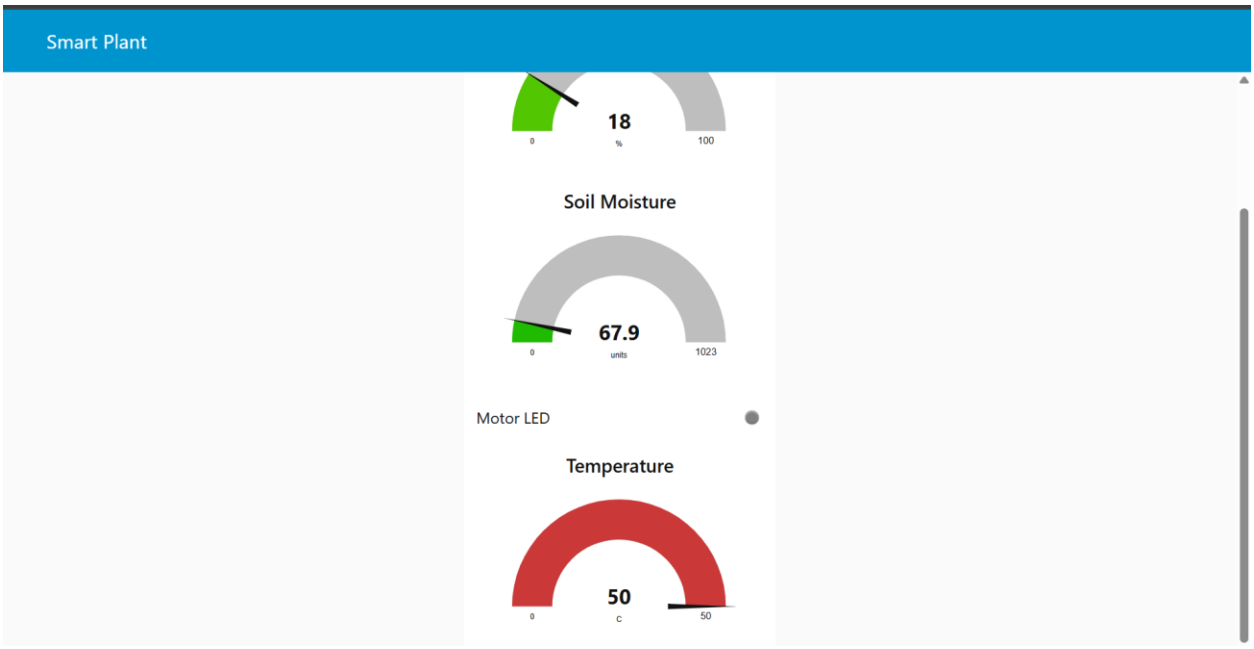
9. Node-RED Dashboard

- Web-based dashboard for monitoring IoT data.
- Receives data from ESP32 via **MQTT broker**.
- Visualizes:
    - Temperature (gauge)
    - Humidity (gauge)
    - Soil Moisture (gauge)
    - Motor Status (LED)

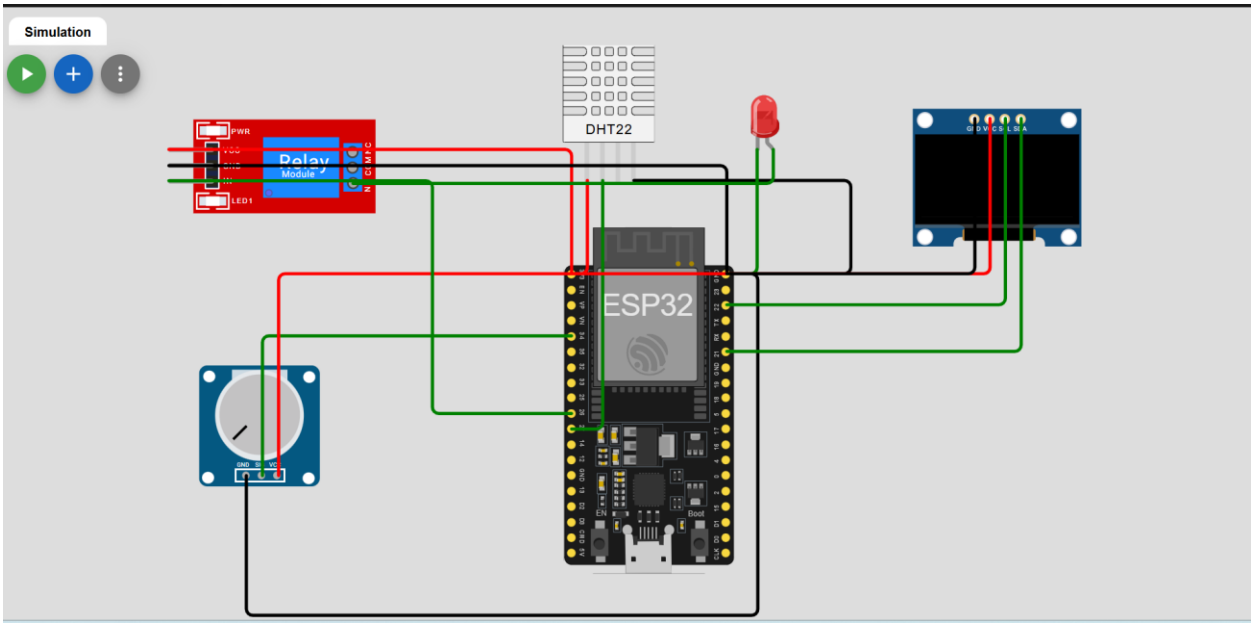
Screenshot Placeholder:







10. Circuit Diagram



11. Automatic Watering Logic

1. Read soil moisture.
2. Compare with threshold (500).
3. If soil < threshold → Pump ON → Motor ON.
4. If soil ≥ threshold → Pump OFF → Motor OFF.
5. Update **OLED**, **Blynk**, and **Node-RED**.

12. Data Transmission via MQTT

- Lightweight IoT protocol for real-time communication.
- ESP32 publishes sensor values to MQTT broker.
- Node-RED subscribes to topics to visualize the data.

Topics:

- smartplant/temp → Temperature
- smartplant/hum → Humidity
- smartplant/soil → Soil Moisture

- smartplant/motor → Motor state

13. Potential Machine Learning Integration

- Currently, system uses **fixed threshold** for soil moisture.
- **ML Integration Idea:**
    - Collect historical sensor data.
    - Train a **predictive model** to determine when watering is needed.
    - Update Node-RED/Blynk to water intelligently.

14. Code:

```
/*************************************************
 * Smart Plant Monitoring System (MQTT + Node-RED + Blynk)
 * ESP32 + DHT11 + Soil Sensor + OLED + Relay + MQTT
 * Group Members:
 *            23-ntu-cs-1257(Laiba Fatima)
 *            23-ntu-cs-1259(Mahnoor Sajjad)
 * BSAI_5th
 *************************************************/

#define BLYNK_TEMPLATE_ID   "TMPL6Wj9hIWLz"
#define BLYNK_TEMPLATE_NAME "Smart Plant Monitoring System"
#define BLYNK_AUTH_TOKEN    "Y1586IR5zmLZyro0jc1Tvgs7lx07aw3d"

#define BLYNK_PRINT Serial

#include <Arduino.h>
```

```
#include <WiFi.h>
#include <BlynkSimpleEsp32.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include "DHT.h"
#include <PubSubClient.h>

// ---------------- WiFi ----------------
char ssid[] = "Umar";
char pass[] = "17337960";

// ---------------- Pins ----------------
#define DHTPIN     27
#define DHTTYPE    DHT11
#define SOIL_PIN   34
#define RELAY_PIN  26

// ---------------- OLED ----------------
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_RESET -1
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);

// ---------------- Objects ----------------
DHT dht(DHTPIN, DHTTYPE);
```

```
BlynkTimer timer;

// ---------------- MQTT ----------------
const char* mqtt_server = "192.168.0.107"; // PC IP with Mosquitto
const int   mqtt_port   = 1883;

const char* TOPIC_TEMP  = "smartplant/temp";
const char* TOPIC_HUM   = "smartplant/hum";
const char* TOPIC_SOIL  = "smartplant/soil";
const char* TOPIC_MOTOR = "smartplant/motor";

WiFiClient espClient;
PubSubClient mqtt(espClient);

// ---------------- Variables ----------------
int soilValue = 0;
int threshold = 500;      // Soil dry limit
bool motorState = false;

// ---------------- Blynk Motor Control ----------------
BLYNK_WRITE(V3) {   // Button widget (Switch Mode)
  motorState = param.asInt();
  digitalWrite(RELAY_PIN, motorState ? HIGH : LOW);
}

// ---------------- MQTT Connect ----------------
```

```
void connectMQTT() {
  while (!mqtt.connected()) {
    Serial.println("Connecting to MQTT...");
    if (mqtt.connect("ESP32_SmartPlant")) {
      Serial.println("MQTT connected");
    } else {
      Serial.print("Failed, rc=");
      Serial.println(mqtt.state());
      delay(2000);
    }
  }
}

// ---------------- Read Sensors ----------------
void readAndSendData() {

  float temp = dht.readTemperature();
  float hum  = dht.readHumidity();
  soilValue  = analogRead(SOIL_PIN);

  if (isnan(temp) || isnan(hum)) {
    Serial.println("DHT Error");
    return;
  }

  // ---------- OLED Display ----------
```

```
display.clearDisplay();
display.setCursor(0,0);
display.println("Smart Plant System");
display.println("------------------");
display.print("Temp: "); display.print(temp); display.println(" C");
display.print("Hum : "); display.print(hum); display.println(" %");
display.print("Soil: "); display.println(soilValue);

if (soilValue < threshold)
  display.println("Soil: DRY");
else
  display.println("Soil: WET");

display.display();

// ---------- Auto Watering ----------
if (soilValue < threshold && !motorState) {
  digitalWrite(RELAY_PIN, HIGH);
  Blynk.virtualWrite(V3, 1);
  motorState = true;
}
else if (soilValue >= threshold && !motorState) {
  digitalWrite(RELAY_PIN, LOW);
  Blynk.virtualWrite(V3, 0);
  motorState = false;
}
```

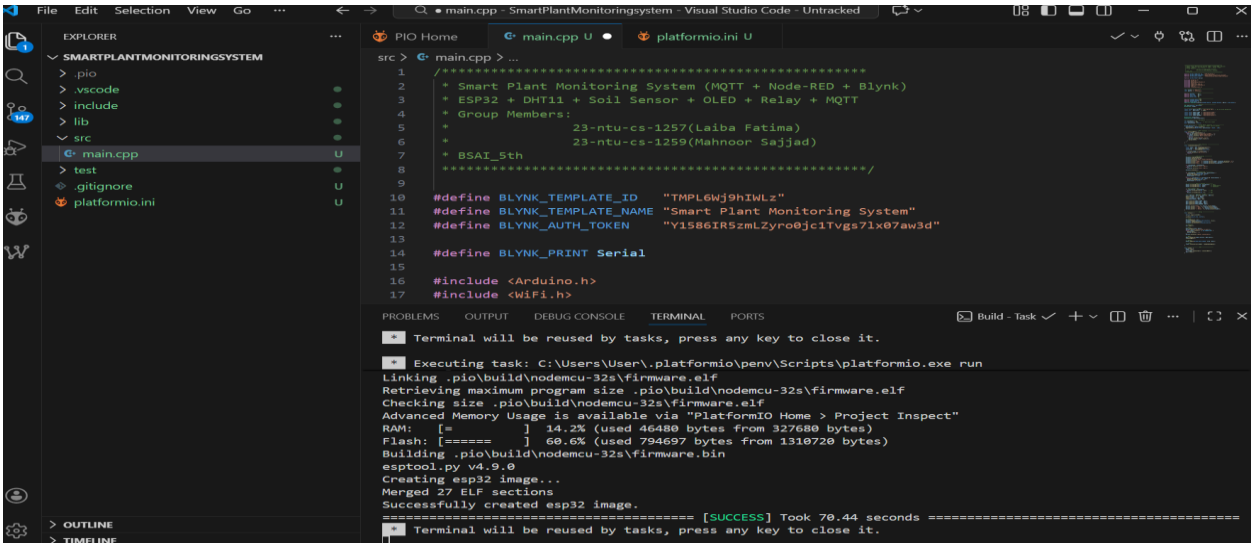
```
// ---------- Send to Blynk ----------
Blynk.virtualWrite(V0, soilValue); // Soil
Blynk.virtualWrite(V1, temp);      // Temperature
Blynk.virtualWrite(V2, hum);       // Humidity

// Soil Status LED
if (soilValue < threshold)
  Blynk.virtualWrite(V4, 255);  // DRY
else
  Blynk.virtualWrite(V4, 0);    // WET

// ---------- Publish to MQTT ----------
char tBuf[8], hBuf[8], sBuf[8], mBuf[2];
dtostrf(temp, 4, 2, tBuf);
dtostrf(hum,  4, 2, hBuf);
itoa(soilValue, sBuf, 10);
itoa(motorState, mBuf, 10);

mqtt.publish(TOPIC_TEMP, tBuf);
mqtt.publish(TOPIC_HUM,  hBuf);
mqtt.publish(TOPIC_SOIL, sBuf);
mqtt.publish(TOPIC_MOTOR, mBuf);

Serial.print("Temp: "); Serial.print(tBuf);
Serial.print(" | Hum: "); Serial.print(hBuf);
```

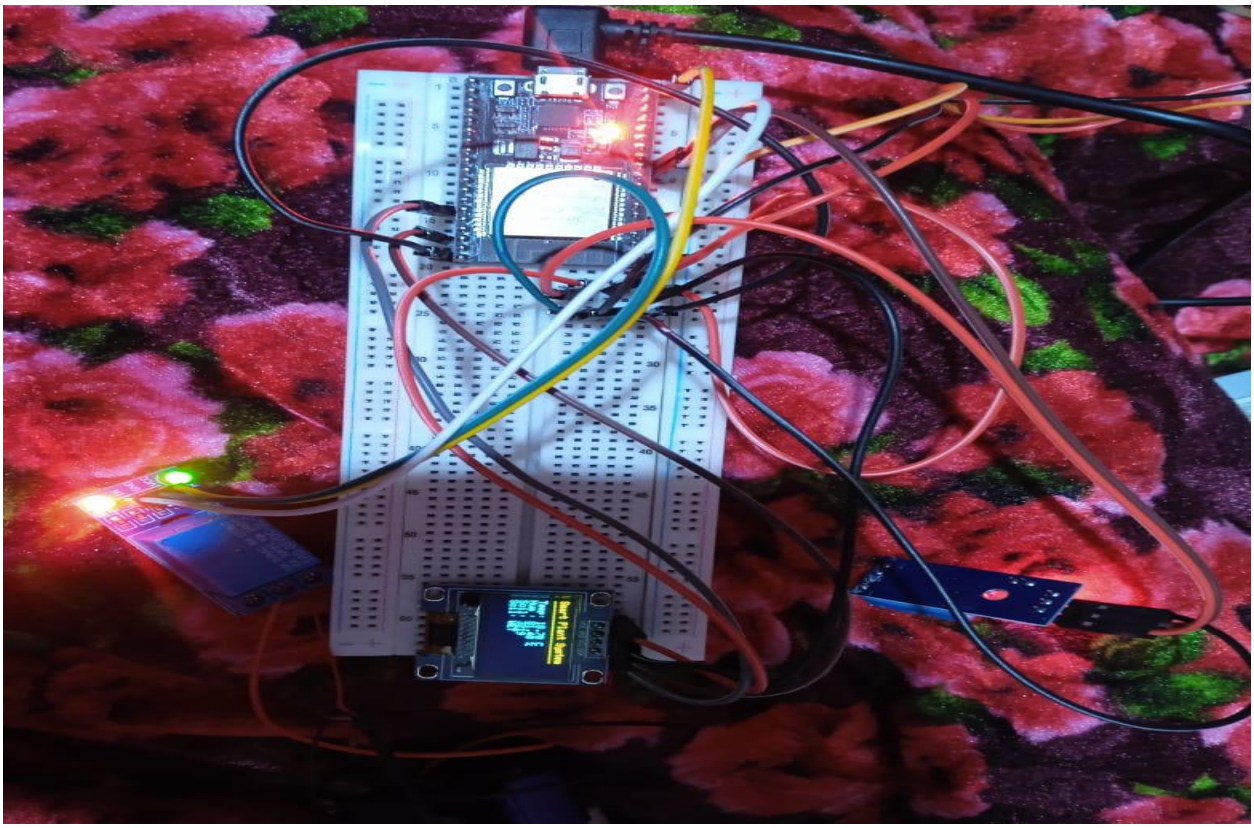
```
  Serial.print(" | Soil: "); Serial.print(sBuf);
  Serial.print(" | Motor: "); Serial.println(mBuf);
}

// ---------------- Setup ----------------
void setup() {

  Serial.begin(115200);

  pinMode(RELAY_PIN, OUTPUT);

  // OLED
  Wire.begin(21, 22);
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);

  // Sensors
  dht.begin();

  // WiFi
  WiFi.begin(ssid, pass);
  Serial.print("Connecting to WiFi");
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
```

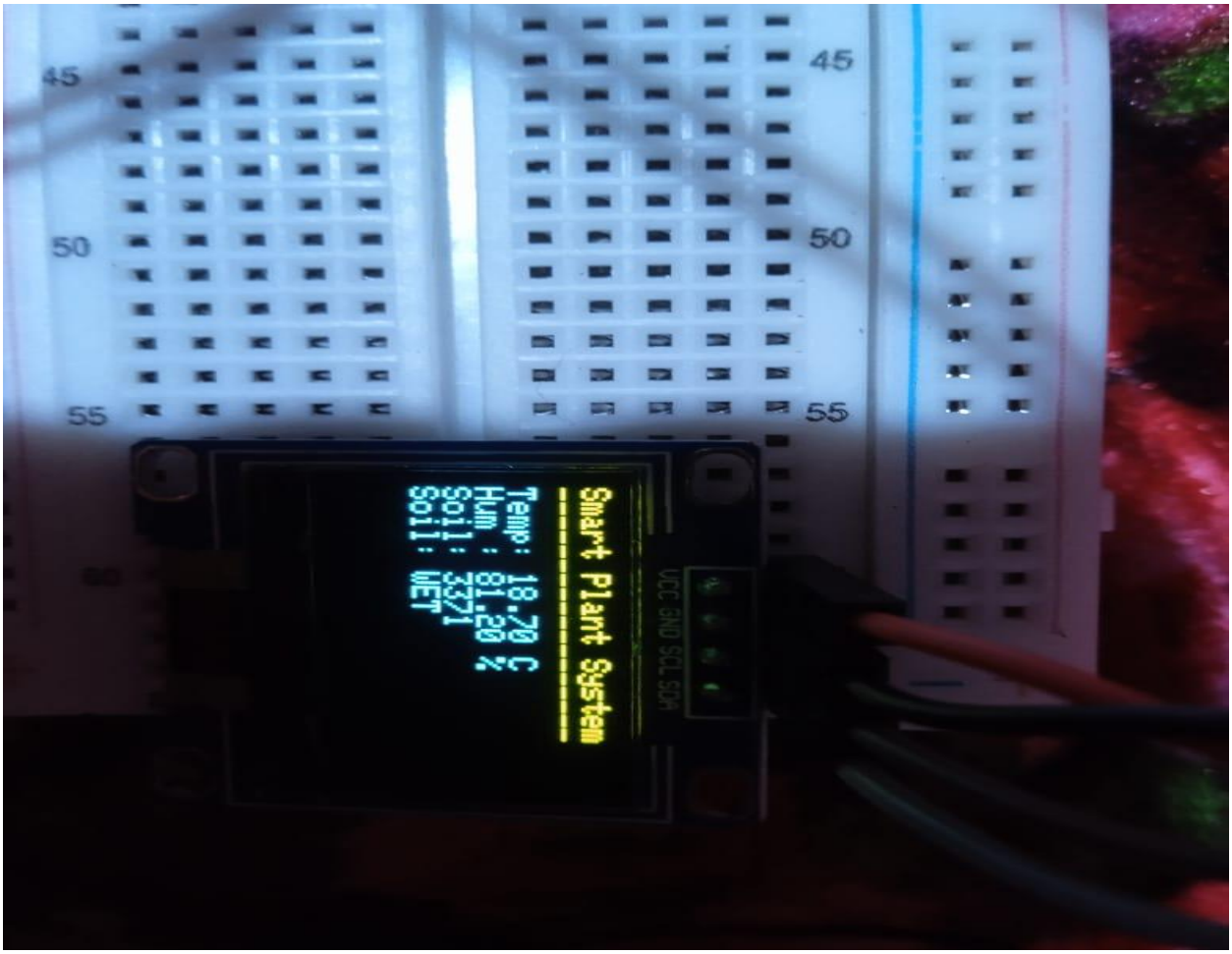
```
  }
  Serial.println("\nWiFi connected");

  // MQTT
  mqtt.setServer(mqtt_server, mqtt_port);
  connectMQTT();

  // Blynk
  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);

  // Timer
  timer.setInterval(3000L, readAndSendData);
}

// ---------------- Loop ----------------
void loop() {
  Blynk.run();
  timer.run();
  if (!mqtt.connected()) connectMQTT();
  mqtt.loop();
}
```

15. Output

Build success:


File Edit Selection View Go
main.cpp - SmartPlantMonitoringsystem - Visual Studio Code - Untracked
EXPLORER
SMARTPLANTMONITORINGSYSTEM
.pio
.vscode
include
lib
src
main.cpp
test
.gitignore
platformio.ini
PIO Home
main.cpp U
platformio.ini U
src > main.cpp > ...
/***
* Smart Plant Monitoring System (MQTT + Node-RED + Blynk)
* ESP32 + DHT11 + Soil Sensor + OLED + Relay + MQTT
* Group Members:
* 23-ntu-cs-1257(Laiba Fatima)
* 23-ntu-cs-1259(Mahnoor Sajjad)
* BSAI_5th
***/
#define BLYNK_TEMPLATE_ID "TMPL6Wj9hIWLz"
#define BLYNK_TEMPLATE_NAME "Smart Plant Monitoring System"
#define BLYNK_AUTH_TOKEN "Y1586IR5zmLZyro0jc1Tvgs71x07aw3d"
#define BLYNK_PRINT Serial
#include <Arduino.h>
#include <WiFi.h>
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Build - Task
Terminal will be reused by tasks, press any key to close it.
Executing task: C:\Users\User\.platformio\penv\Scripts\platformio.exe run
Linking .pio\build\nodemcu-32s\firmware.elf
Retrieving maximum program size .pio\build\nodemcu-32s\firmware.elf
Checking size .pio\build\nodemcu-32s\firmware.elf
Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"
RAM: [=] 14.2% (used 46480 bytes from 327680 bytes)
Flash: [======] 60.6% (used 794697 bytes from 1310720 bytes)
Building .pio\build\nodemcu-32s\firmware.bin
esptool.py v4.9.0
Creating esp32 image...
Merged 27 ELF sections
Successfully created esp32 image.
========== [SUCCESS] Took 70.44 seconds ==========
Terminal will be reused by tasks, press any key to close it.
OUTLINE
TIMELINE





16. Summary

The **Smart Plant Monitoring System** is a complete **IoT-based plant care solution**:

- Monitors **temperature, humidity, and soil moisture**.
- Performs **automatic watering** using relay-controlled pump.
- Sends real-time data to **Blynk app** and **Node-RED dashboard**.
- Can be extended for **Machine Learning-based predictive watering**.