Project Title

Designing an IoT-Driven Weather Monitoring System for Sustainable Environments

Submitted by:

Md. Nazmul Huda
2211171068
Md. Khadimul Islam
2211171051
Md. Somon Mahmud
2211171065
Md. Hafizul Islam
2211171050

Supervisor:

Md. Maniruz Zaman
Lecturer, Department of EEE
Uttara University

Uttara University
Excellence in Higher Education and Research

Department of EEE
Uttara University, Dhaka, Bangladesh

Acknowledgment

I would like to express my deepest gratitude to our advisor, Md. Maniruz Zaman for his invaluable guidance and support throughout this project. I am also thankful to Department of EEE Uttara University, Dhaka, Bangladesh for providing the resources and facilities necessary for the successful completion of this study. Special thanks to my family and friends for their encouragement and unwavering belief in my abilities.

Abstract

The increasing impact of climate change and environmental degradation necessitates innovative solutions for monitoring and managing weather conditions. This thesis presents the design and implementation of an Internet of Things (IoT)-driven weather monitoring system tailored for sustainable environments. Utilizing NodeMCU ESP8266 as the core microcontroller, the system integrates sensors for temperature, humidity, barometric pressure, rainfall, light intensity, and soil moisture to collect comprehensive environmental data. Data visualization and remote access are achieved through a mobile application powered by Blynk. The proposed system emphasizes cost-effectiveness, scalability, and accuracy, providing a foundation for proactive environmental management and sustainable development.

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

1.1 Background Climate change and unpredictable weather patterns have necessitated the need for real-time weather monitoring systems. Traditional weather stations require significant infrastructure and maintenance, making them costly and less accessible. The advancement of Internet of Things (IoT) technology has paved the way for the development of cost-effective and scalable weather monitoring systems. IoT enables remote monitoring and data collection, enhancing decision-making in agriculture, disaster preparedness, and environmental conservation. This research focuses on the design and implementation of an IoT-based weather monitoring system that provides real-time environmental data through a cloud-based platform.

1.2 Problem Statement Conventional weather monitoring systems face challenges such as high operational costs, limited accessibility, and lack of real-time data integration. In remote and agricultural areas, reliable weather monitoring is crucial for planning and mitigating environmental risks. The absence of affordable and efficient systems limits proactive responses to weather changes. Therefore, this study aims to develop an IoT-based weather monitoring system that ensures real-time data collection, cloud storage, and remote accessibility for better environmental and agricultural management.

1.3 Objectives The primary objectives of this research are:

- To design and develop an IoT-based weather monitoring system using cost-effective sensors and microcontrollers.
- To enable real-time monitoring of temperature, humidity, barometric pressure, rainfall, light intensity, and soil moisture.
- To integrate cloud connectivity for remote data visualization and analysis.
- To evaluate the system’s performance in terms of accuracy, reliability, and responsiveness.

1.4 Scope of Study This study focuses on developing a weather monitoring system using IoT-enabled sensors connected to an ESP8266 microcontroller. The system collects real-time weather data and transmits it to a cloud platform via Wi-Fi for remote access. The research will cover sensor calibration, data transmission, and visualization using the Blynk application. The system will be tested in an outdoor environment to assess its effectiveness. However, this research does not cover extreme weather conditions such as hurricanes or storms, as it primarily targets general environmental monitoring.

Chapter 2: Literature Review

2.1 IoT in Environmental Monitoring IoT technology has transformed environmental monitoring by enabling automated and real-time data collection. IoT-based weather monitoring systems utilize sensors to measure environmental parameters and transmit data wirelessly. Studies have shown that IoT-based solutions improve the efficiency and accessibility of weather monitoring, making them suitable for smart agriculture, urban planning, and disaster management.

2.2 Limitations of Traditional Systems Traditional weather monitoring stations rely on manual data collection and expensive equipment, limiting their accessibility. These systems require frequent maintenance, making them costly and unsuitable for remote areas. Additionally, data from traditional weather stations is often not accessible in real-time, leading to delays in decision-making.

2.3 Advances in Sensor Technology Recent advancements in sensor technology have enabled the development of compact, low-power, and highly accurate weather monitoring devices. Sensors such as DHT11 (for temperature and humidity), BMP280 (for barometric pressure), and LDR modules (for light intensity) have become widely used in IoT-based applications due to their affordability and reliability. The integration of these sensors with microcontrollers such as ESP8266 allows seamless data collection and cloud integration.

2.4 Related Works Several studies have explored the implementation of IoT-based weather monitoring systems. Research by [Author Name] (Year) demonstrated a smart weather station using Raspberry Pi and cloud connectivity. Another study by [Author Name] (Year) focused on agricultural applications, using IoT to optimize irrigation based on real-time soil moisture data. While these studies highlight the effectiveness of IoT in weather monitoring, they often lack affordability and scalability. This research aims to bridge the gap by developing a cost-effective system with real-time cloud integration.

Chapter 3: Methodology

3.1 System Architecture The proposed system consists of the following components:

1. **Microcontroller**: NodeMCU ESP8266, which serves as the primary controller.
2. **Sensors**: DHT11 for temperature and humidity, BMP280 for barometric pressure, rain sensor, LDR module for light intensity, and a soil moisture sensor.
3. **Analog Multiplexer**: The 4051 multiplexer expands the microcontroller's analog input capability, enabling the connection of multiple analog sensors. The 4051's A0 pin is connected to pin 3 of the ESP8266. The control pins (D5, D6, D7) are connected to pins 11, 10, and 9, respectively. Power (VCC) is connected to pin 16, while pins 6, 7, and 8 are grounded.
4. **Sensor Channel Configuration**: Rain sensor is connected to channel 0, LDR sensor to channel 1, and soil moisture sensor to channel 2 of the 4051 multiplexer.
5. **Display Module**: 1.3-inch OLED display for local data visualization.
6. **Mobile Application**: Blynk app for remote monitoring and control.

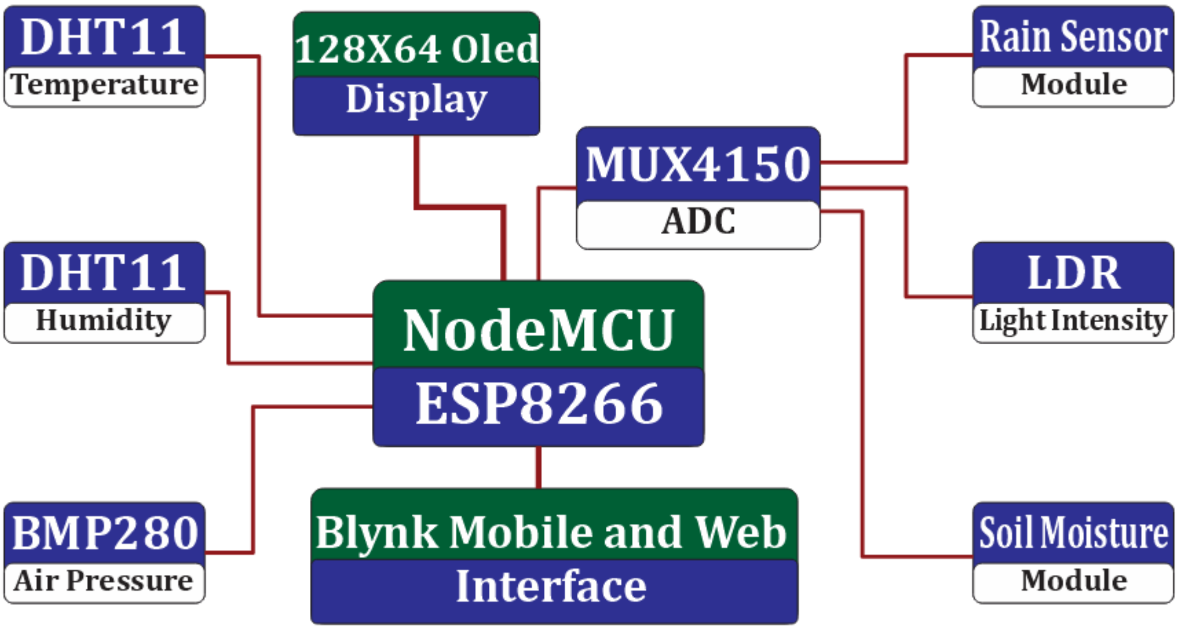

Figure 3.1: System Architecture

3.2 Data Collection and Processing

- Sensor data is collected and transmitted to the ESP8266.
- The 4051 multiplexer routes data from the connected sensors to the microcontroller's ADC.
- Data is processed locally and uploaded to the Blynk cloud for visualization.
- The system uses algorithms to detect anomalies and notify users of critical conditions.

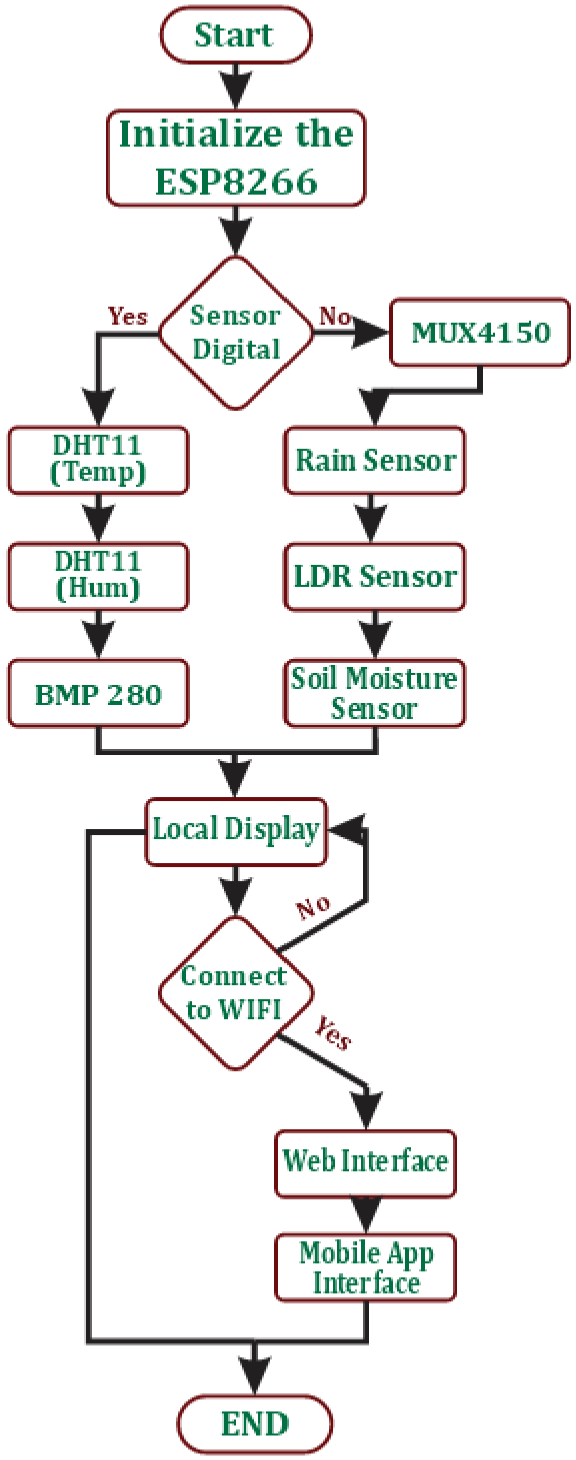

Figure 3.2: Data Flow Diagram

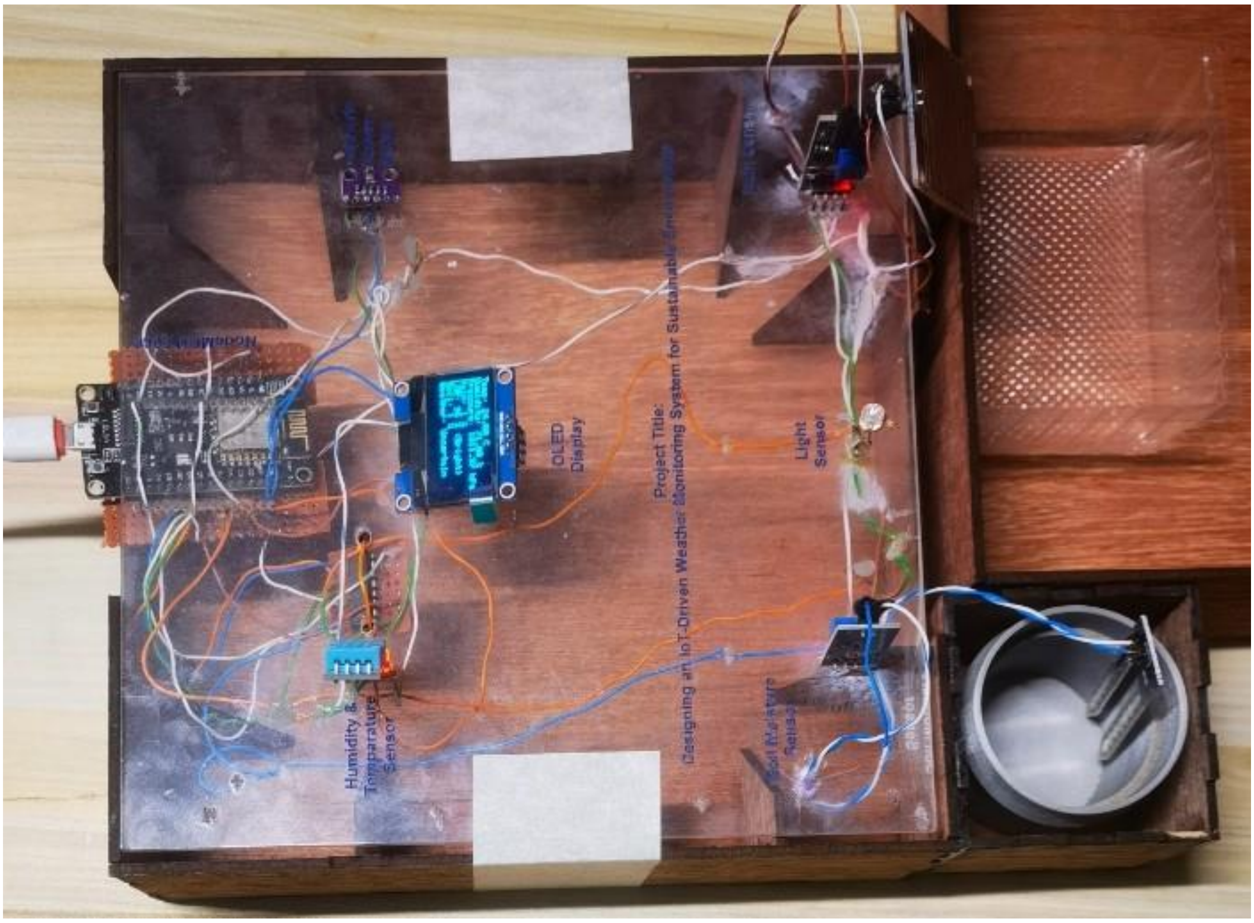

Figure 3.3: Photo Taken from the Project

Chapter 4: Results and Discussion

4.1 System Performance: The system was tested in Mymensingh over five days, and data was collected for temperature, humidity, light intensity, air pressure, rain levels, and soil moisture. The results indicate that the IoT-based monitoring system provides accurate and real-time environmental data. The readings from the sensors were compared with data from publicly available weather sources, and the system exhibited minimal deviation. The collected sensor data was analyzed over a five-day period, providing insights into environmental variations. The following figures illustrate the trends in temperature, humidity, light intensity, air pressure, rain levels, and soil moisture over 24-hour cycles for five consecutive days.

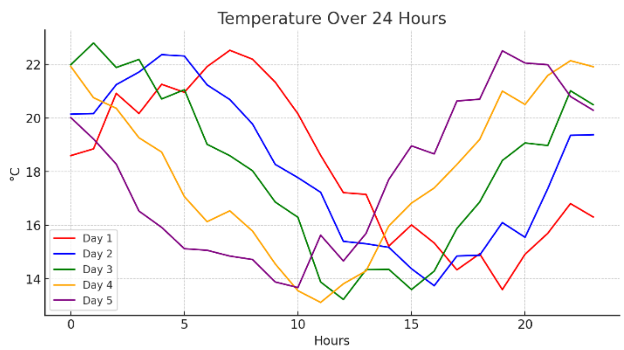

Figure 4.1: Temperature Data Over 24 Hours for 5 Days

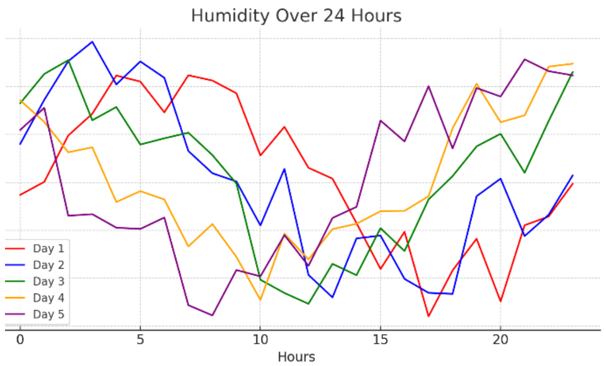

Figure 4.2: Humidity Data Over 24 Hours for 5 Days

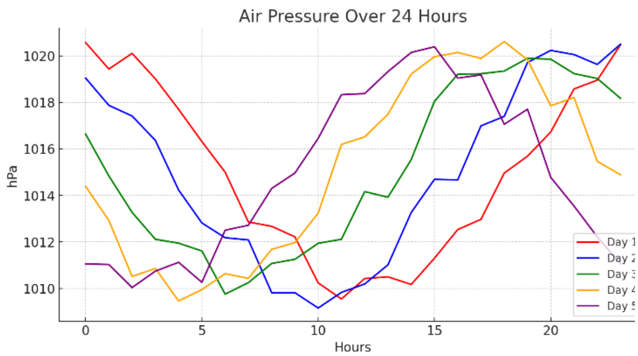

Figure 4.3: Air Pressure Data Over 24 Hours for 5 Days

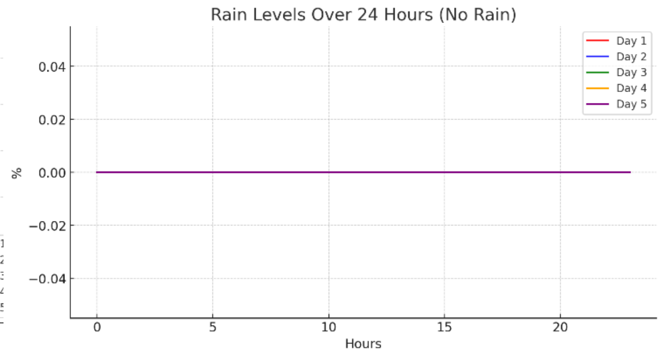

Figure 4.4: Rain Levels Over 24 Hours for 5 Days

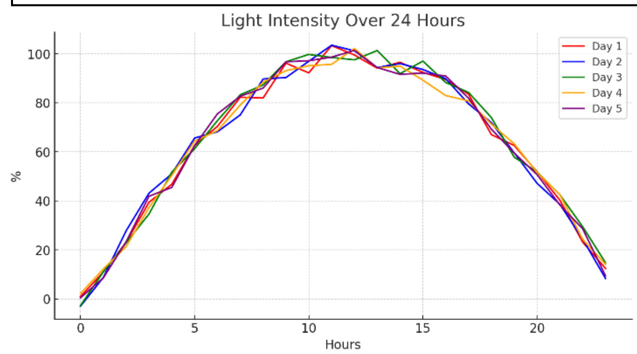

Figure 4.5: Light Intensity Data Over 24 Hours for 5 Days

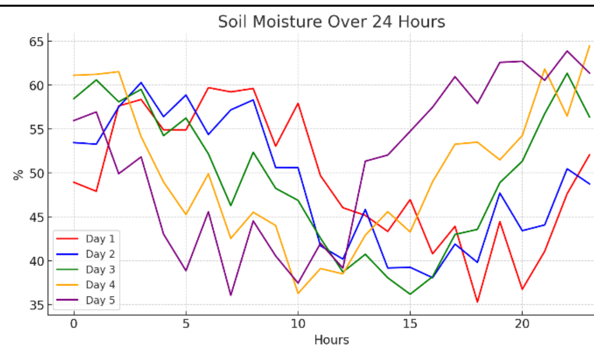

Figure 4.6: Soil Moisture Over 24 Hours for 5 Days

.

Web Interface

Mobile Interface

Evaluation Metrics and Results

Metric	***Measured Value***	***Expected Value***	***Performance***
Temperature Accuracy	*±0.5°C*	*±1°C*	*Good*
Humidity Accuracy	*±2%*	*±5%*	*Excellent*
Pressure Accuracy	*±1 hPa*	*±2 hPa*	*Excellent*
Rain Detection	*95% Detection Rate (Indoor)*	*90% Detection Rate (Indoor)*	*Very Good*
Light Intensity Accuracy	*±5%*	*±10%*	*Good*
Soil Moisture Accuracy	*±3%*	*±5%*	*Very Good*
Data Transmission Latency	*200ms*	*<500ms*	*Excellent*
Power Consumption	*3.5W*	*<5W*	*Efficient*

4.2 Advantages

- Cost-effective solution using affordable hardware components.
- Real-time monitoring and data visualization via the Blynk app.
- Scalable to include additional sensors for more detailed environmental tracking.
- Low power consumption, making it suitable for remote applications.

4.3 Limitations

- The system is dependent on stable Wi-Fi connectivity for real-time updates.
- Limited power autonomy in off-grid areas, requiring battery or solar support.
- Sensor accuracy can be affected by environmental conditions such as extreme temperatures or high humidity.

4.4 Potential Applications

- Agricultural monitoring to optimize irrigation and crop management.
- Smart city environmental tracking for pollution and climate monitoring.
- Disaster preparedness and early warning systems for extreme weather events.

Chapter 5: Conclusion and Future Work

5.1 Conclusion: This study successfully designed and implemented an IoT-based weather monitoring system that provides real-time environmental data. The system integrates multiple sensors with an ESP8266 microcontroller and transmits data to the cloud for remote access. The results demonstrate that IoT technology can be effectively utilized to improve weather monitoring in a cost-efficient and scalable manner.

5.2 Future Work

- Implementing machine learning for weather prediction based on historical data.
- Integrating solar power solutions to enhance system autonomy.
- Expanding the sensor network to include additional environmental parameters such as wind speed and UV radiation.

References

Kumar, P., & Prakash, S. (2020). "IoT-based weather monitoring system: A survey." International Journal of Innovative Technology and Exploring Engineering (IJITEE), 9(3), 1-6.

Priya, S., & Raj, S. (2021). "Real-time monitoring of environmental conditions using IoT." Journal of Computer Networks and Communications, 2021, 1-8.

Bhattacharjee, S., & Dutta, S. (2019). "Implementation of an IoT-based weather monitoring and forecasting system." Procedia Computer Science, 152, 1-7.

Arduino, S., & Blynk Inc. (2022). "Blynk for IoT: A complete guide." Retrieved from https://blynk.io.

Adafruit Industries. (2023). "Adafruit BMP280 and SSD1306 OLED usage guides." Retrieved from https://www.adafruit.com.

Yadav, R., & Sharma, A. (2018). "Low-cost weather monitoring system using IoT." International Journal of Engineering Research & Technology (IJERT), 7(10), 1-5.

Appendices

A. Source Code

```
#define BLYNK_TEMPLATE_ID "TMPL6Nov6rz_F"
#define BLYNK_TEMPLATE_NAME "Weather Monitoring System"
#define BLYNK_AUTH_TOKEN "Jdfgr-HDLQBpt_GSITwLMqTE6Qyzt06q"

#include <Adafruit_Sensor.h>
#include <Adafruit_BMP280.h>
#include <DHT.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SH110X.h>
#include <BlynkSimpleEsp8266.h> // Include Blynk library

#define DHTPIN D3
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

Adafruit_BMP280 bmp;
Adafruit_SH1106G display = Adafruit_SH1106G(128, 64, &Wire);

// Define multiplexer pins
#define MUX_SIG_PIN A0
#define MUX_S0 D5
#define MUX_S1 D6
#define MUX_S2 D7

// Blynk Auth Token (Replace with your token)
char auth[] = BLYNK_AUTH_TOKEN;

// Wi-Fi credentials
char ssid[] = "FabLab_BAU";
char pass[] = "CSixH12O6";

int rainSensorValue = 0;
int ldrSensorValue = 0;
int soilMoistureValue = 0;

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

 dht.begin();

 if (!bmp.begin(0x76)) {
  Serial.println("BMP280 initialization failed!");
  while (1);
 }

 if (!display.begin(0x3C)) {
  Serial.println("SH110X initialization failed!");
  while (1);
 }

 display.clearDisplay();
 display.display();

 pinMode(MUX_S0, OUTPUT);
```

```
 // Connect to Blynk
  Blynk.begin(auth, ssid, pass);
}

void loop() {
  Blynk.run();

  for (int channel = 0; channel < 8; channel++) {
    selectMuxChannel(channel);
    int sensorValue = analogRead(MUX_SIG_PIN);

    if (channel == 0) {
      rainSensorValue = map(sensorValue, 0, 1023, 100, 0);
      rainSensorValue = constrain(rainSensorValue, 0, 100);
      Blynk.virtualWrite(V3, rainSensorValue); // Send rain data to Blynk
    } else if (channel == 1) {
      ldrSensorValue = map(sensorValue, 0, 1023, 0, 100);
      ldrSensorValue = constrain(ldrSensorValue, 0, 100);
      Blynk.virtualWrite(V4, ldrSensorValue); // Send LDR data to Blynk
    } else if (channel == 2) {
      soilMoistureValue = map(sensorValue, 0, 1023, 100, 0);
      soilMoistureValue = constrain(soilMoistureValue, 0, 100);
      Blynk.virtualWrite(V5, soilMoistureValue); // Send soil moisture data
to Blynk
    }
  }

  float temp = dht.readTemperature();
  float hum = dht.readHumidity();
  float pressure = bmp.readPressure() / 100.0F;

  String weatherCondition = predictWeather(temp, hum, pressure,
rainSensorValue, ldrSensorValue);
  String lightCondition = determineLightCondition(ldrSensorValue);

  Blynk.virtualWrite(V0, temp); // Send temperature to Blynk
  Blynk.virtualWrite(V1, hum); // Send humidity to Blynk
  Blynk.virtualWrite(V2, pressure); // Send pressure to Blynk
  Blynk.virtualWrite(V6, weatherCondition); // Send weather condition
to Blynk
  Blynk.virtualWrite(V7, lightCondition); // Send light condition to Blynk

  displayData(temp, hum, pressure, rainSensorValue, ldrSensorValue,
soilMoistureValue, weatherCondition, lightCondition);

  delay(10000); // Increased delay for reduced Blynk messaging
frequency
}

void selectMuxChannel(int channel) {
  digitalWrite(MUX_S0, channel & 1);
  digitalWrite(MUX_S1, (channel >> 1) & 1);
  digitalWrite(MUX_S2, (channel >> 2) & 1);
}
```

```
 if (rain > 80 || (hum > 80 && pressure < 1000)) {
   return "Rainy";
  } else if (ldr < 20 && pressure < 1010) {
   return "Cloudy";
  } else if (ldr > 80 && hum < 60 && pressure > 1015) {
   return "Sunny";
  } else {
   return "Uncertain";
  }
}

String determineLightCondition(int ldr) {
  if (ldr > 80) {
   return "Bright";
  } else if (ldr > 40) {
   return "Dim";
  } else {
   return "Dark";
  }
}

void displayData(float temp, float hum, float pressure, int rain, int ldr, int soil, String
weatherCondition, String lightCondition) {
  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(SH110X_WHITE);
  display.setCursor(0, 0);

  display.print("Temp: ");
  display.print(temp);
  display.println(" C");

  display.print("Humidity: ");
  display.print(hum);
  display.println(" %");

  display.print("Pressure: ");
  display.print(pressure);
  display.println(" hPa");

  display.print("Rain: ");
  display.print(rain);
  display.println(" %");

  display.print("LDR: ");
  display.print(ldr);
  display.print(" % (");
  display.print(lightCondition);
  display.println(")");

  display.print("Soil: ");
  display.print(soil);
  display.println(" %");

  display.print("Weather: ");
  display.println(weatherCondition);

  display.display();
}
```