SMART IRRIGATION SYSTEM USING INTERNET OF THINGS (IoT)

By

KEERTHANA S

(Registration Number: 23352030)

Project report submitted in partial fulfilment of the requirements for the award of the degree of

MASTER OF COMPUTER APPLICATIONS

DEPARTMENT OF COMPUTER SCIENCE

SCHOOL OF ENGINEERING & TECHNOLOGY

PONDICHERRYUNIVERSITY

NOVEMBER -2024

BONAFIDE CERTIFICATE

This is to certify that this project work entitled "**SMART IRRIGATION SYSTEM USING INTERNET OF THINGS (IoT)**" is a bonafide record of work done by **MISS. KEERTHANA S** (Reg. Number **23352030**) in the partial fulfilment for the Degree of **Master of Computer Applications** in the **Department of Computer Science, School of Engineering and Technology** of Pondicherry University.

This work has not been submitted elsewhere for the award of any other degree to the best of our knowledge.

INTERNAL GUIDE	**HEAD OF THE DEPARTMENT**
Dr. POTHULA SUJATHA	**Dr. S. K. V. JAYAKUMAR**
Assistant Professor	Professor/HOD
Department of Computer Science School of Engineering & Technology Pondicherry University	Department of Computer Science School of Engineering & Technology Pondicherry University
Pondicherry – 605 014	Pondicherry – 605 014

Submitted for the Viva-Voce Examination held on:

INTERNAL EXAMINER	**EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

The completion of this work is undoubtedly ensured by the support of lot of people and it would be impossible without people who supported and believed in me. First and foremost, I am very happy to show my immense gratitude and grateful acknowledgement to my beloved guide **Dr. POTHULA SUJATHA,** Assistant Professor, Department of Computer Science, School of Engineering and Technology, Pondicherry University, Pondicherry, for her excellent guidance, constant encouragement and centering efforts, which helped me to complete my project work.

I express my sincere thanks to **Dr. S.K.V. JAYAKUMAR,** Professor and Head, Department of Computer Science, School of Engineering and Technology, Pondicherry University, Pondicherry, for his kind advises and constant encouragement during the course of this project work.

My gratitude also extends to all the **faculty members and non-teaching staffs** of the Department of Computer Science, for all the assistance they rendered so willingly, to help me in completing my project report.

Finally I am extremely thankful to my parents and friends for giving me the moral support in doing all things.

(KEERTHANA S)

TABLE OF CONTENT

ABSTRACT

The *Smart Irrigation System Using IoT* leverages the power of Internet of Things (IoT) technology to automate and optimize the irrigation process in agriculture, addressing key challenges such as water wastage and labor costs. This system employs the NodeMCU ESP8266 microcontroller, which is connected to soil moisture sensors embedded in the soil to continuously monitor the moisture levels. When the moisture level falls below a predefined threshold, the system automatically triggers an irrigation process by activating a connected water pump via a relay module.

The integration of the Blynk IoT platform enhances the functionality of the system by providing a mobile app interface for remote monitoring and control. Through Blynk, farmers can track real-time soil moisture data, view irrigation status, and manually activate or deactivate the water pump, all from their smartphones. This feature ensures that the system is user-friendly and accessible to individuals with minimal technical expertise. Additionally, the Blynk platform supports push notifications, alerting the user when soil moisture is critically low or when the irrigation process is complete.

The system's key benefits include reduced water consumption through precise irrigation, minimized human intervention, and better management of water resources. It also allows for significant cost savings by automating the irrigation process and optimizing water usage. The project is designed to be scalable, offering the possibility of expanding the system to larger agricultural areas with multiple zones, each independently monitored and controlled.

By combining real-time data collection, IoT connectivity, and automated irrigation, this project exemplifies the potential of smart technologies to transform agricultural practices. The *Smart Irrigation System* can play a critical role in sustainable farming by improving resource efficiency, enhancing crop health, and reducing environmental impact.

1. INTRODUCTION

1.1 ABOUT THE PROJECT

The *Smart Irrigation System Using IoT* is designed to address the challenges faced in traditional irrigation systems, such as water wastage, high labor costs, and inefficient water distribution. The project leverages Internet of Things (IoT) technology to create an automated, efficient irrigation system. By using soil moisture sensors connected to a NodeMCU microcontroller, the system continuously monitors the soil's moisture content and determines when irrigation is necessary. When the soil moisture level drops below a predefined threshold, the system automatically triggers a relay to activate the water pump, thus ensuring precise water delivery to the crops.

The system is further enhanced by the integration of the Blynk platform, which enables users to monitor and control the irrigation process remotely through a smartphone app. This allows farmers or users to track soil moisture data in real-time, receive notifications when irrigation is required, and manually control the pump if necessary. The main goal of this project is to conserve water, reduce human intervention, and optimize agricultural practices by making irrigation more efficient and responsive to the actual needs of the crops. Through automation, this system not only saves time but also ensures that crops receive the optimal amount of water, resulting in better crop yields and overall farm management.

1.2 PROJECT PLAN

The *Smart Irrigation System Using IoT* follows a structured and phased approach to ensure the project is effectively developed, tested, and deployed:

1. Planning:
   The first phase involves identifying the project's objectives and gathering the necessary resources. This includes selecting the appropriate hardware (NodeMCU, soil moisture sensors, relays) and software (Blynk app, Arduino IDE for programming). During this phase, goals for the system are set, such as optimizing water usage, automating irrigation,

and providing remote control and monitoring features. The feasibility of the project is evaluated, considering both technical and economic factors.

2. Design:

   In the design phase, the system architecture and diagrams are created to provide a blueprint for the system's implementation. This includes defining the connections between the NodeMCU, sensors, and actuators, as well as designing the user interface in the Blynk app. Flowcharts and diagrams, such as the system flowchart, use case diagrams, and sequence diagrams, are prepared to visually represent the system's working and interaction between components.

3. Implementation:

   This phase focuses on building and integrating both the hardware and software components. The NodeMCU is programmed to read data from the soil moisture sensor and control the irrigation pump through the relay. The Blynk app is configured to display real-time data from the sensors and to enable users to monitor and control the system remotely. The components are physically assembled, and the system is tested in a controlled environment.

4. Testing:

   The testing phase involves ensuring that all components are functioning as expected. The system's performance is tested by simulating different soil moisture conditions to check if the pump activates correctly. The Blynk app is tested for real-time updates, notification accuracy, and remote-control functionality. Any bugs or issues identified during this phase are fixed, and the system is optimized for better performance.

5. Deployment:

   After successful testing, the system is deployed in a real-world agricultural setting. The setup is installed on a farm or a designated agricultural area, and the system is monitored over time to ensure its effectiveness in real-world conditions. During deployment, adjustments may be made to improve system performance or add additional features based on user feedback.

1.3 ABOUT THE ORGANIZATION

The *Smart Irrigation System Using IoT* is a project supported by [Organization Name], an institution committed to advancing technological innovation and sustainability in agriculture. [Organization Name] is at the forefront of research and development in IoT applications, particularly in the context of improving agricultural practices. Through its state-of-the-art facilities, the organization offers access to cutting-edge technologies and equipment, allowing students and researchers to work on real-world problems and create practical solutions.

The institution actively promotes interdisciplinary collaboration by providing mentorship and guidance from experts in various fields, such as electronics, software engineering, and environmental science. This collaborative environment is crucial for developing innovative projects that bridge the gap between technology and agriculture. Additionally, [Organization Name] offers funding opportunities, research grants, and access to agricultural labs, encouraging the development of projects that enhance productivity and sustainability in farming.

By supporting this project, [Organization Name] demonstrates its commitment to fostering technological advancements that not only improve agricultural efficiency but also contribute to environmental conservation and sustainable farming practices. The organization’s role is pivotal in ensuring that the project receives the necessary resources and guidance to successfully tackle global agricultural challenges using IoT technology.

2. PROBLEM DEFINITION & FEASIBILITY ANALYSIS

2.1 PROBLEM DEFINITION

Traditional irrigation methods, such as flood irrigation and manual watering, are inefficient and often lead to significant water wastage. These methods typically do not account for real-time soil moisture conditions, resulting in either over-irrigation or under-irrigation. Over-irrigation wastes valuable water resources and increases energy consumption, while under-irrigation can lead to poor crop health and reduced yields. Additionally, traditional irrigation systems require substantial manual labor, increasing operational costs and reducing efficiency.

This project aims to solve these issues by implementing an IoT-based irrigation system. The proposed system uses soil moisture sensors to monitor real-time soil conditions and triggers irrigation automatically when moisture levels fall below a preset threshold. By using an automated, data-driven approach, this system ensures precise water usage, reduces manual intervention, and optimizes the water distribution process. The goal is to enhance agricultural productivity while conserving water and minimizing labor costs, contributing to more sustainable farming practices.

2.2 FEASIBILITY STUDY

2.2.1 TECHNICAL FEASIBILITY

The technical feasibility of the *Smart Irrigation System Using IoT* is supported by the use of the NodeMCU ESP8266, a widely used and affordable microcontroller that enables seamless integration with IoT applications. NodeMCU is equipped with Wi-Fi capabilities, allowing it to connect to the internet for remote control and monitoring via the Blynk app. The sensors used in this project, such as soil moisture sensors, are simple to interface with NodeMCU and provide accurate real-time data on soil conditions.

Additionally, relays are employed to control the water pump based on sensor readings, ensuring that irrigation is activated only when required. The integration with Blynk ensures that all components work together effectively. The app allows users to visualize real-time data and control the system remotely, making it user-friendly and efficient. The ease of setting up the system with readily available hardware and the flexibility of the software platform make this solution technically feasible, even for individuals with limited technical expertise.

2.2.2 OPERATIONAL FEASIBILITY

Operationally, the *Smart Irrigation System* is designed to be simple and user-friendly. The system is controlled primarily via the Blynk app, which can be installed on smartphones, making it accessible to farmers without requiring advanced technical knowledge. The Blynk app provides an intuitive interface, where users can monitor the soil moisture levels, check the status of irrigation, and manually control the water pump if needed. Additionally, the app can send notifications to alert users when irrigation is required or if there are any issues with the system.

The system’s automation minimizes the need for constant manual intervention, allowing farmers to focus on other tasks while the system ensures that crops are receiving the proper amount of water. With its straightforward setup and easy-to-use interface, the system can be quickly adopted by farmers in rural or remote areas, making it operationally feasible for a wide range of agricultural users, from small-scale farms to larger commercial operations.

2.2.3 ECONOMIC FEASIBILITY

The economic feasibility of the project is one of its most attractive aspects. The cost of building the *Smart Irrigation System* is relatively low compared to traditional irrigation systems. The primary components, including the NodeMCU, soil moisture sensors, relay modules, and water pumps, are affordable and widely available in the market. This makes the system accessible to a broad range of farmers, particularly those in developing countries, where water conservation and cost reduction are critical.

In addition to the low initial cost, the system provides significant long-term savings. By ensuring that irrigation is only activated when necessary, the system reduces water wastage, conserving both water and energy. This not only results in lower utility bills but also minimizes the environmental impact of excessive water use. Over time, the cost savings from reduced water and energy consumption, as well as the decreased need for manual labor, will offset the initial investment, making the system a cost-effective solution for sustainable agriculture.

3. SOFTWARE REQUIREMENT

3.1 HARDWARE REQUIREMENTS

The hardware components required for the *Smart Irrigation System Using IoT* are crucial to the system’s operation and functionality. They include:

- NodeMCU ESP8266 Microcontroller: The core component of the system, the NodeMCU ESP8266, is a low-cost Wi-Fi-enabled microcontroller that is ideal for IoT applications. It processes the data from the soil moisture sensors and controls the relay module that activates the irrigation system. The ESP8266 ensures seamless connectivity with the Blynk app, allowing remote monitoring and control.

- Soil Moisture Sensors: These sensors are responsible for measuring the moisture content in the soil. Based on the data they provide, the system determines whether irrigation is required. The sensors typically consist of two probes that detect electrical resistance in the soil, with lower resistance indicating drier soil.
- Relay Module and Water Pump: The relay module acts as a switch, turning the water pump on or off based on the signal from the NodeMCU. The water pump is connected to the irrigation system and delivers water to the crops when needed. The relay module allows for safe and controlled operation of the pump, which is powered by the system.
- Power Supply and Irrigation Tubing: A stable power supply is essential to run the NodeMCU and the water pump. A suitable power adapter (usually 5V DC for NodeMCU and higher for the pump) is required. Additionally, the irrigation tubing ensures that water is distributed to the crops efficiently. This can be designed based on the farm's layout and needs.

3.2 SOFTWARE REQUIREMENTS

The software used in this project is vital for programming the NodeMCU and integrating IoT functionalities. The main software components are:

- Arduino IDE: The Arduino Integrated Development Environment (IDE) is used to write and upload the code to the NodeMCU microcontroller. It supports the C/C++ programming language and allows for easy interfacing with various sensors and modules. The Arduino IDE is used to program the logic for reading soil moisture levels, controlling the relay module, and sending data to the Blynk app.
- Blynk App: The Blynk platform is a mobile application used to monitor and control the irrigation system remotely. It connects to the NodeMCU via Wi-Fi and provides a user-friendly interface on a smartphone, allowing users to track soil moisture levels, control the water pump, and receive notifications. The Blynk app simplifies IoT integration without requiring extensive coding skills, making it an ideal choice for this project.
- Optional: Firebase: Firebase is an optional cloud platform that can be integrated for data storage and analytics. It can store real-time data from the soil moisture sensors, enabling users to track historical data trends and optimize irrigation schedules. Firebase also supports user authentication and can be used for generating reports and insights.

3.3 SYSTEM REQUIREMENTS

For the *Smart Irrigation System Using IoT* to operate efficiently, the following system requirements must be met:

- Wi-Fi Connectivity: Since the system relies on IoT capabilities, a stable Wi-Fi network is essential for data transmission between the NodeMCU and the Blynk app. Wi-Fi allows the NodeMCU to send real-time data to the Blynk app, enabling remote control and monitoring of the system.
- Smartphone with Blynk App: The user must have a smartphone with the Blynk app installed to interact with the system. The app provides a graphical interface for monitoring soil moisture levels, controlling irrigation, and receiving push notifications about the system's status. The app is available for both iOS and Android devices.
- Internet Access for Cloud Services (Optional): If Firebase or any other cloud service is used for data storage and analysis, a stable internet connection is required to upload and retrieve data from the cloud. This allows users to access historical data and receive insights into the system's performance over time.

4. SYSTEM DESIGN

4.1 MODULE DESCRIPTION

The system is divided into several modules, each responsible for a specific function in the *Smart Irrigation System Using IoT*. Below is a description of each module:

- **Sensor Module**

  The sensor module includes the soil moisture sensors that monitor the moisture levels in the soil. These sensors are placed in the ground at various locations to gather accurate data about the soil's current condition. They work by measuring the electrical resistance of the soil, which changes based on its moisture content. The data collected is then sent to the controller module for further processing. This module is crucial for detecting when the soil is dry and irrigation is required.

- **Controller Module**

  The controller module is based on the NodeMCU ESP8266, which processes the sensor data. When the moisture level falls below a predefined threshold, the controller determines that irrigation is necessary. It processes the data from the sensors and sends a signal to the actuator module to activate the irrigation system. The controller module also communicates with the Blynk app to provide real-time data updates to the user and enables remote control.

- **Actuator Module**

  The actuator module consists of a relay and a water pump. The relay is controlled by the controller module, which switches the water pump on or off based on the moisture levels detected by the sensor. If the soil moisture is below the threshold, the relay activates the pump to start watering the crops. Once the moisture level reaches the desired level, the relay turns off the pump. This module ensures the automation of the irrigation process.

- **User Interface**

  The user interface is built using the Blynk app, which acts as the front-end platform for controlling the system. Through the Blynk app, users can view real-time data such as soil moisture levels and the status of the irrigation system. The app provides buttons to manually turn the irrigation system on or off and sends push notifications for status updates, such as when irrigation is triggered or when the soil is too dry. This module makes the system accessible from anywhere, enhancing the user experience and enabling remote management.

4.2 USE CASE DIAGRAM

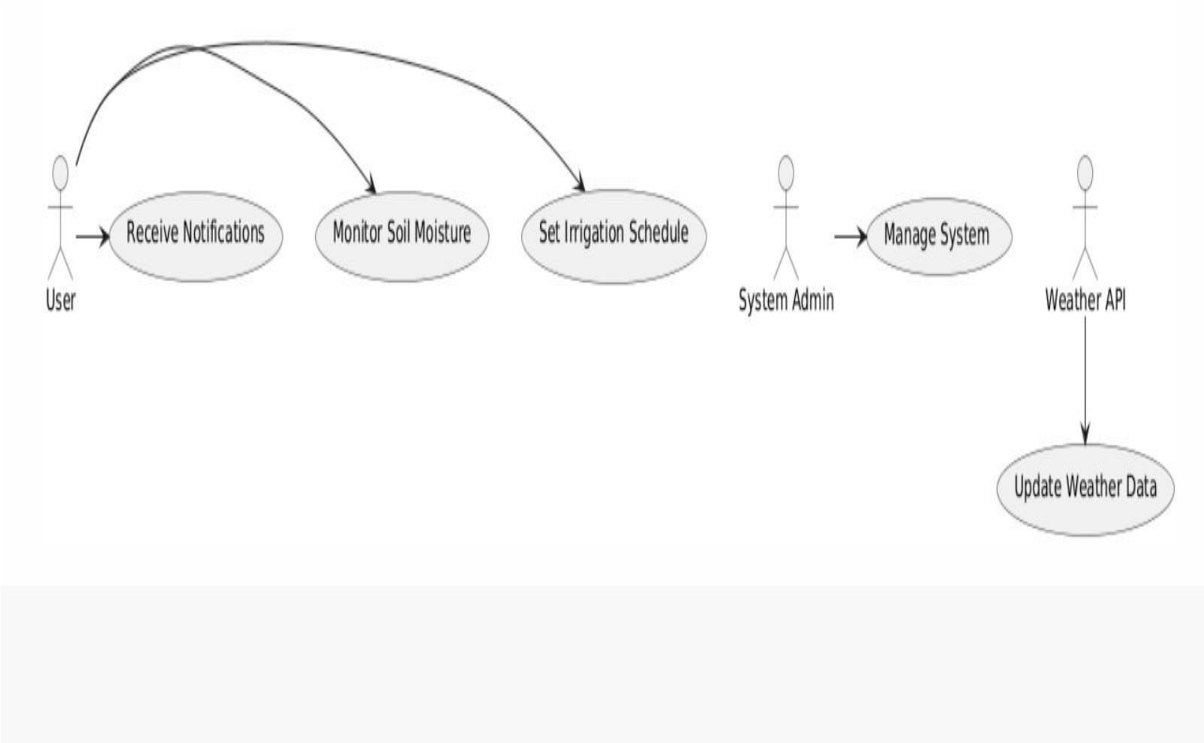
The use case diagram illustrates the interaction between the system components and the external environment. The primary actors in the system include the **User**, the **System**, and the **External Environment** (e.g., the field, sensors, and Wi-Fi network). The diagram will show the following use cases:

- **User Actions**: The user interacts with the system through the Blynk app. The user can:

- Monitor soil moisture levels.
- View irrigation status.
- Manually start or stop the irrigation system.
- Receive notifications regarding soil conditions or system status.

- **System Actions**: The system performs tasks such as:
  - Collecting soil moisture data from the sensors.
  - Sending data to the user interface (Blynk app).
  - Activating or deactivating the irrigation pump based on sensor input.
- **External Environment**: The field and Wi-Fi network act as the environment in which the system operates. The soil moisture sensor module collects data from the field, and the Wi-Fi module of the NodeMCU facilitates communication with the Blynk app.

  This diagram helps visualize how the different actors interact with the system and outlines the primary actions that take place during the operation of the irrigation system.

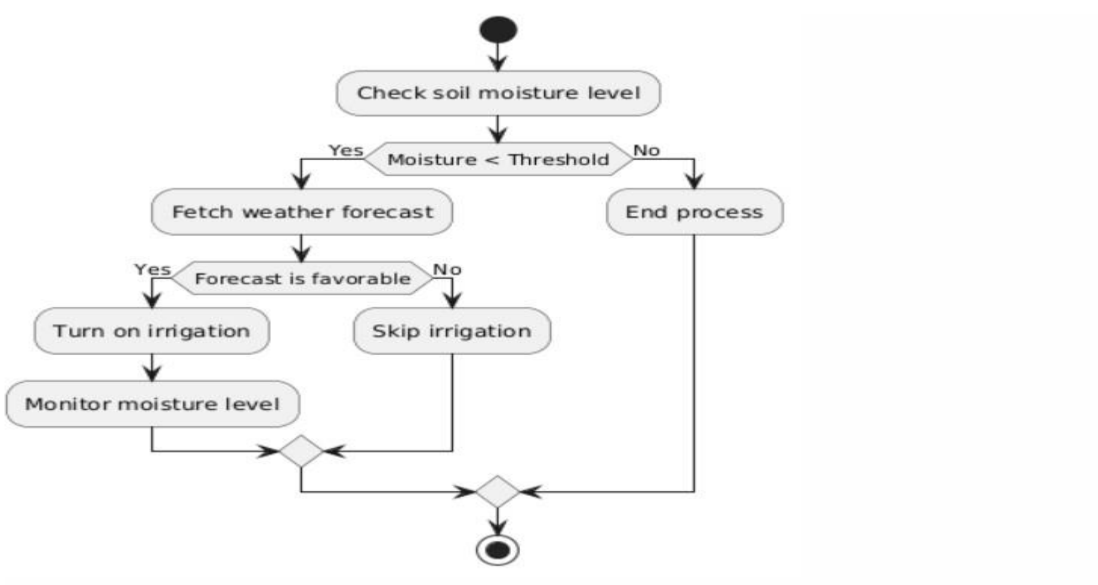


4.3 ACTIVITY DIAGRAM

The activity diagram provides a step-by-step overview of the system's workflow. It starts with the soil moisture sensors detecting the current moisture level in the soil. If the moisture level falls below the threshold, the system proceeds with the following actions:

1. **Sensor Detection**: The sensor reads the soil moisture level and sends the data to the controller module (NodeMCU).
2. **Decision Making**: The controller checks if the moisture level is below the defined threshold.
3. **Activate Irrigation**: If the soil is dry, the controller sends a signal to the relay module to activate the water pump.
4. **Watering Process**: The pump activates and water is delivered to the crops.
5. **Moisture Monitoring**: The sensor continuously monitors the soil moisture. Once the soil reaches the desired moisture level, the pump is turned off by the relay.
6. **Feedback to User**: The Blynk app provides feedback to the user, displaying the current soil moisture level and system status (e.g., whether irrigation is active or not).

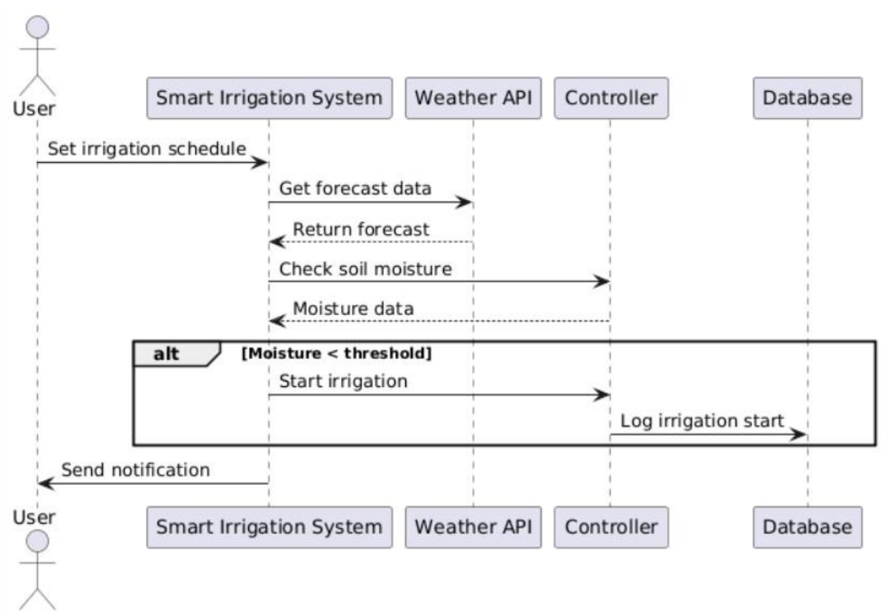
This diagram ensures a clear understanding of how the system automates the irrigation process and how different actions and decisions flow through the system.



4.4 SEQUENCE DIAGRAM

The sequence diagram illustrates the order in which actions occur within the system, starting with data collection and moving through decision-making and action execution:

1. **Sensor Data Collection**: The soil moisture sensors send data to the NodeMCU microcontroller.
2. **Data Processing**: The NodeMCU processes the moisture data and compares it to the predefined threshold.
3. **Decision Point**: Based on the moisture level:
    - If the moisture level is low, the NodeMCU triggers the relay module to activate the water pump.
    - If the moisture level is sufficient, no action is taken.
4. **Activation/Deactivation**: The relay module switches the water pump on or off, depending on the moisture level.
5. **Feedback to User**: The NodeMCU sends real-time updates to the Blynk app, where the user can monitor the soil moisture and pump status. The system can send notifications if irrigation is triggered or if the system detects any issues.



5. IMPLEMENTATION

5.1 BLYNK CONFIGURATION

Blynk is a powerful IoT platform that allows remote monitoring and control of IoT devices. The implementation begins with setting up a Blynk account, creating a new project, and obtaining the unique *Auth Token*. The Auth Token is integrated into the NodeMCU code to enable communication with the Blynk server.

Steps:

1. Download the Blynk app from the Google Play Store or Apple App Store.
2. Create a new project and add widgets like a gauge (to display soil moisture) and buttons (to control the water pump).
3. Note the Auth Token sent via email, as it is required for programming the NodeMCU.

5.2 SENSOR AND NODEMCU CONFIGURATION *(0.5 Page)*

The soil moisture sensor is connected to the NodeMCU to measure real-time moisture levels. The sensor's analog output is read using the ADC pin of the NodeMCU. The data is processed and sent to the Blynk app for visualization.

Connections:

- **VCC (Sensor):** Connected to the 3.3V pin of NodeMCU.
- **GND (Sensor):** Connected to the GND pin.
- **OUT (Sensor):** Connected to the A0 pin of NodeMCU.

5.3 RELAY MODULE CONFIGURATION

The relay module controls the water pump. When the moisture level falls below the threshold, the NodeMCU sends a signal to activate the relay, turning on the pump.

Connections:

- **VCC (Relay):** Connected to 3.3V of NodeMCU.

- **GND (Relay):** Connected to GND.
- **IN (Relay):** Connected to a digital GPIO pin (e.g., D1).

5.4 CODE IMPLEMENTATION IN ARDUINO IDE

The NodeMCU is programmed using the Arduino IDE. The key components of the code include:

1. **Libraries:** Installing and including the Blynk and ESP8266WiFi libraries.
2. **Auth Token & Wi-Fi Credentials:** Configuring the system to connect to the Blynk server using the Auth Token and local Wi-Fi.
3. **Logic:** Writing a logic to read sensor data and trigger the relay based on a pre-set moisture threshold.
4. **Blynk Virtual Pins:** Using virtual pins for sending data to the app and receiving user inputs.

```
#define BLYNK_TEMPLATE_NAME "SMART IRRIGATION"

#define BLYNK_TEMPLATE_ID "TMPL3EXctvFV9"

#define BLYNK_AUTH_TOKEN "LFb4BbVUSzDRl_YPQldcqdCmGNT6NoIY"


#include <DHT.h>

#include <DHT_U.h>

#include <BlynkSimpleEsp8266.h>  // Include Blynk library for ESP8266


// WiFi credentials

char ssid[] = "vivo Y100A";         // Replace with your WiFi SSID

char pass[] = "kiran@2002";     // Replace with your WiFi password
```

```
#define DHTPIN D4          // DHT sensor pin
#define DHTTYPE DHT11      // DHT 11 sensor type
DHT dht(DHTPIN, DHTTYPE);  // Initialize DHT sensor

// Relay Pin
#define RELAY_PIN D0

// Virtual pins in Blynk
#define VIRTUAL_PIN_TEMP V1
#define VIRTUAL_PIN_HUMIDITY V2
#define VIRTUAL_PIN_SOIL_MOISTURE V3
#define VIRTUAL_PIN_RELAY V4

// Threshold values for alerts
const float TEMP_THRESHOLD = 20.0;   // Temperature threshold for low alert (adjust as needed)
const int MOISTURE_THRESHOLD = 30;   // Moisture threshold for low alert (adjust as needed)

// Notification flags
bool tempAlertSent = false;
bool moistureAlertSent = false;

void setup() {
```

```
  // Initialize Serial Monitor
  Serial.begin(9600);
  delay(1500);  // Delay to wait for Serial Monitor

  // Initialize Blynk connection
  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass, "blynk.cloud", 8080);

  // Initialize DHT sensor
  dht.begin();

  // Initialize Relay pin
  pinMode(RELAY_PIN, OUTPUT);
  digitalWrite(RELAY_PIN, HIGH);  // Turn relay off initially
}

void loop() {
  // Run Blynk
  Blynk.run();

  // Read temperature and humidity from DHT11 sensor
  float humidity = dht.readHumidity();
  float temperature = dht.readTemperature();
```

```
  // Read soil moisture value and map it
  int soil_Moisture = analogRead(A0);
  soil_Moisture = map(soil_Moisture, 500, 1023, 100, 1);

  // Send values to Blynk app
  Blynk.virtualWrite(VIRTUAL_PIN_TEMP, temperature);
  Blynk.virtualWrite(VIRTUAL_PIN_HUMIDITY, humidity);
  Blynk.virtualWrite(VIRTUAL_PIN_SOIL_MOISTURE, soil_Moisture);

  // Check for low temperature alert
if (temperature < TEMP_THRESHOLD && !tempAlertSent) {
    Blynk.logEvent("low_temperature_alert", "Warning: Temperature is too low!");
    tempAlertSent = true;  // Set flag to avoid repeated alerts
} else if (temperature >= TEMP_THRESHOLD) {
    tempAlertSent = false;  // Reset flag if temperature goes back to normal
}

// Check for low soil moisture alert
if (soil_Moisture < MOISTURE_THRESHOLD && !moistureAlertSent) {
    Blynk.logEvent("low_moisture_alert", "Warning: Soil moisture is too low!");
    moistureAlertSent = true;  // Set flag to avoid repeated alerts
} else if (soil_Moisture >= MOISTURE_THRESHOLD) {
    moistureAlertSent = false;  // Reset flag if moisture goes back to normal
```

```
}

  // Add a delay for the sampling period
  delay(300);
}

// Blynk function to handle relay control
BLYNK_WRITE(VIRTUAL_PIN_RELAY) {
  int pinValue = param.asInt();  // Get value from Blynk (0 or 1)

  // Control relay based on Blynk input
  if (pinValue) {
    digitalWrite(RELAY_PIN, LOW);  // Turn ON relay
  } else {
    digitalWrite(RELAY_PIN, HIGH);  // Turn OFF relay
  }
}
```

5.5 TESTING THE BLYNK APP

After uploading the code, the system is tested using the Blynk app:

1. Monitor real-time soil moisture values via a gauge widget.
2. Use a button widget to manually control the pump.
3. Set notifications for critical conditions, such as very low soil moisture levels.

5.6 INTEGRATING AUTOMATION

Automation rules are implemented to make the irrigation system independent:

1. The moisture level is continuously monitored.
2. When the value drops below the threshold, the pump is activated automatically.
3. The Blynk app also logs the sensor data for historical analysis.

5.7 CONNECTING MULTIPLE ZONES

For larger fields, multiple sensors and relays are added, with each pair assigned to a specific zone. The Blynk app is configured to display data and control options for each zone, enabling efficient irrigation management.

5.8 FINAL DEPLOYMENT

The system is deployed in a real field setting with all components tested for reliability and scalability. The user is trained to use the Blynk app for monitoring and troubleshooting.

6. SYSTEM TESTING

6.1 SYSTEM IMPLEMENTATION

The implementation of the *Smart Irrigation System Using IoT* involves integrating the hardware and software components to create a fully functional prototype. This process begins with setting up the hardware, including the NodeMCU ESP8266, soil moisture sensors, relay module, water pump, and the necessary power supply. Once the hardware is assembled and connected, the software development begins using the Arduino IDE to program the NodeMCU.

The system is programmed to read data from the soil moisture sensors and, based on predefined thresholds, send commands to the relay to activate or deactivate the water pump. The Blynk app is also configured to allow remote monitoring and control of the irrigation system. Once the integration of hardware and software is complete, the system is tested for basic functionality by ensuring that data is collected from the sensors, processed by the NodeMCU, and used to control the water pump effectively. The system is then connected to Wi-Fi to facilitate communication with the Blynk app, enabling real-time monitoring of the system's status. After the integration is complete, the system is ready for testing in real-world conditions.

6.2 TESTING

Testing is a crucial part of the development process, ensuring that each component of the system operates as expected and that the entire system functions reliably in real-world conditions. The testing process can be broken down into three main categories: unit testing, validation testing, and functional testing.

6.2.1 UNIT TESTING

Unit testing focuses on testing individual components of the system to ensure that each part works correctly in isolation. In the case of the *Smart Irrigation System*, the primary components tested include:

- **Soil Moisture Sensors**: The sensors are tested to ensure they provide accurate moisture readings. This involves measuring the sensor's output under different soil moisture conditions to verify its reliability and response to changes in moisture levels.
- **Relay Module**: The relay module is tested to confirm that it correctly switches the water pump on and off based on the signals from the NodeMCU. This testing ensures that the relay is responsive and operates within the required voltage and current limits.
- **Water Pump**: The water pump is tested independently to ensure it functions correctly when activated by the relay. The pump should deliver water when instructed and stop when the relay turns it off.

These individual tests help identify any issues with the components before integrating them into the larger system.

6.2.2 VALIDATION TESTING

Validation testing ensures that the system as a whole operates accurately and reliably according to the project requirements. In this phase, the following aspects are validated:

- **Sensor Accuracy**: The soil moisture sensors are calibrated and tested in various soil conditions to confirm their accuracy in detecting moisture levels. The system must react appropriately, starting irrigation when the soil is too dry and stopping when the desired moisture level is reached.
- **System Response**: The controller (NodeMCU) processes the sensor data accurately and triggers the water pump at the correct times. Validation testing ensures that the system doesn't overwater or underwater, maintaining optimal soil moisture levels for plant health.
- **Blynk App Integration**: The Blynk app is validated by ensuring that it displays real-time data, sends notifications, and allows the user to control the irrigation system remotely without delays. This testing confirms that the app communicates efficiently with the system.

Validation testing guarantees that the system performs as intended and meets user expectations.

6.2.3 FUNCTIONAL TESTING

Functional testing evaluates the overall performance of the system under various real-world conditions. It ensures that the system can handle a variety of environmental factors, such as:

- **Environmental Conditions**: The system is tested in different weather conditions to simulate real agricultural environments. For instance, the system may be tested during periods of drought or heavy rainfall to see how well it adjusts to changing soil moisture levels.
- **System Load**: Functional testing also ensures that the system can manage multiple sensors or operate in larger irrigation setups. The system's ability to handle different soil moisture levels across multiple zones and react accordingly is evaluated.
- **Reliability and Durability**: Long-term testing is conducted to ensure that the system is reliable and durable under continuous operation. This includes testing the sensors, relay modules, and pumps over extended periods to assess wear and tear, as well as verifying the system's stability over time.

- **User Interface Testing**: The Blynk app is tested under various conditions (e.g., mobile network connectivity, low signal) to confirm that it remains functional and responsive, allowing users to monitor and control the system as needed.

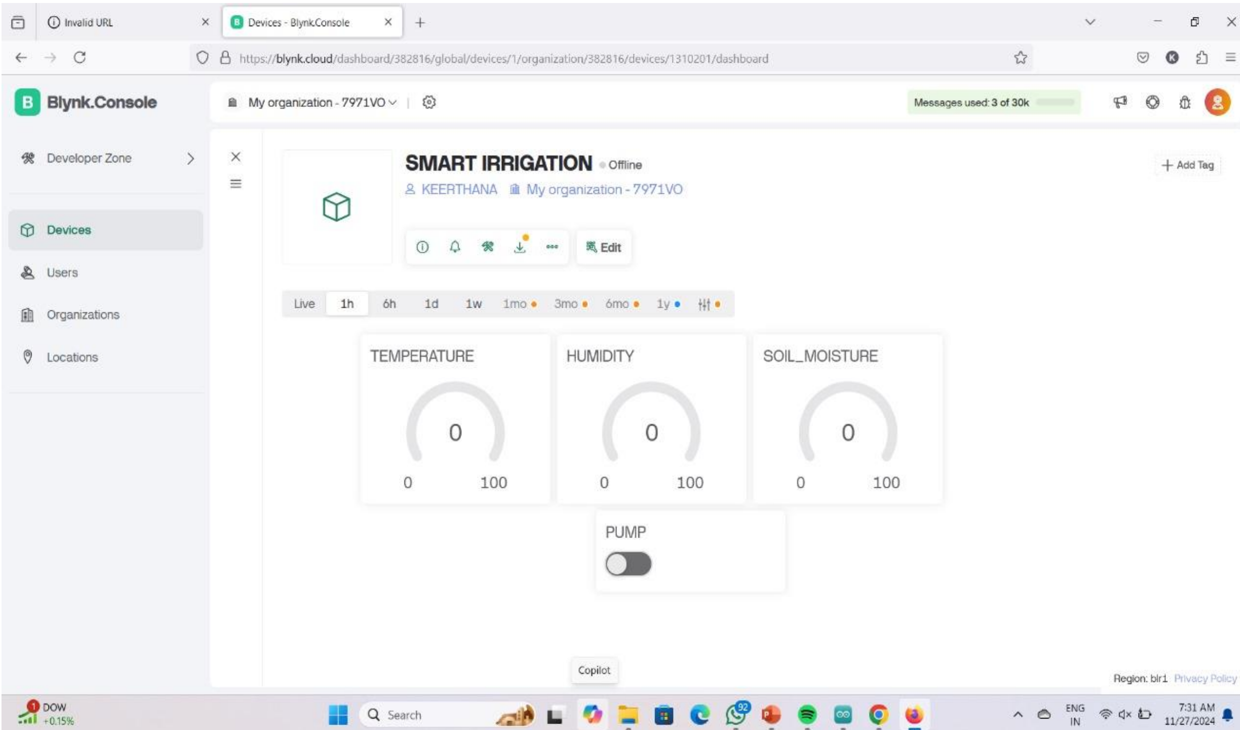
7. CONCLUSION

7.1 FUTURE ENHANCEMENTS

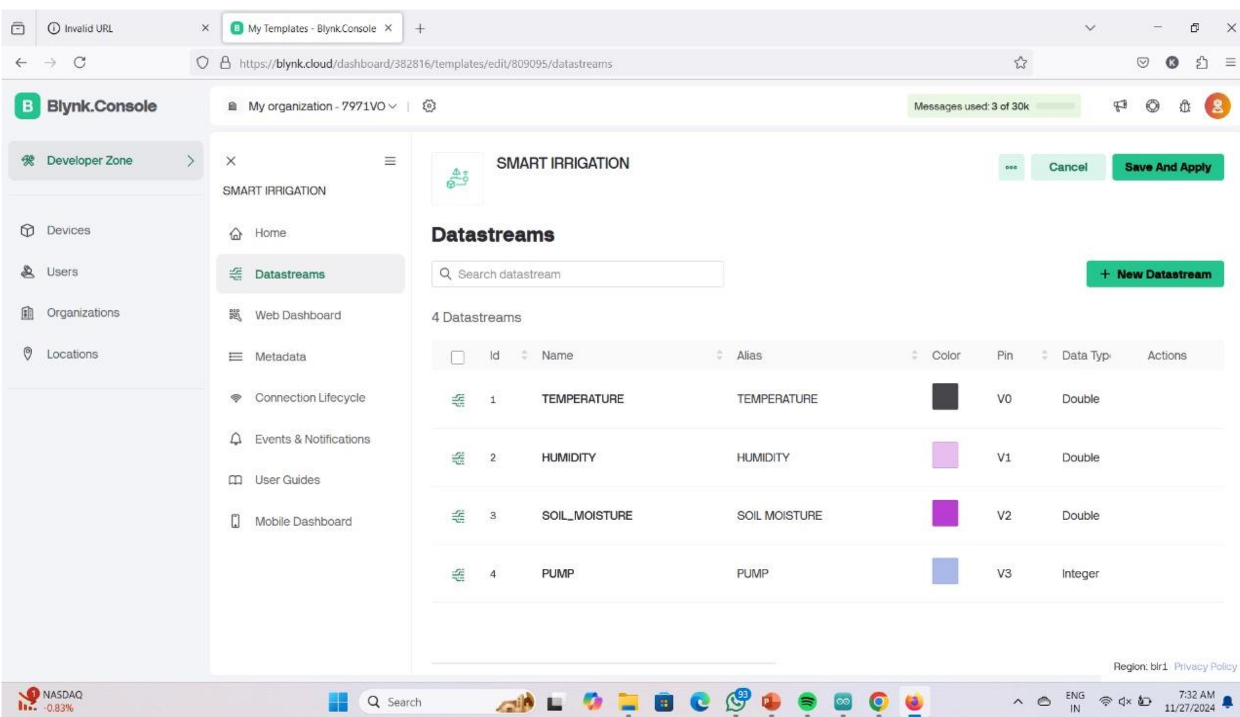
Future enhancements for the *Smart Irrigation System Using IoT* include integrating weather APIs for predictive irrigation, allowing the system to adjust watering schedules based on weather forecasts. Additionally, solar power integration could make the system more sustainable and reduce dependency on external power sources. The system can also be expanded to handle multi-zone irrigation, enabling more precise control over water distribution for different crops. These improvements would further optimize resource usage, lower costs, and increase the system's efficiency in various agricultural settings.

APPENDIX I : SCREEN SHOTS

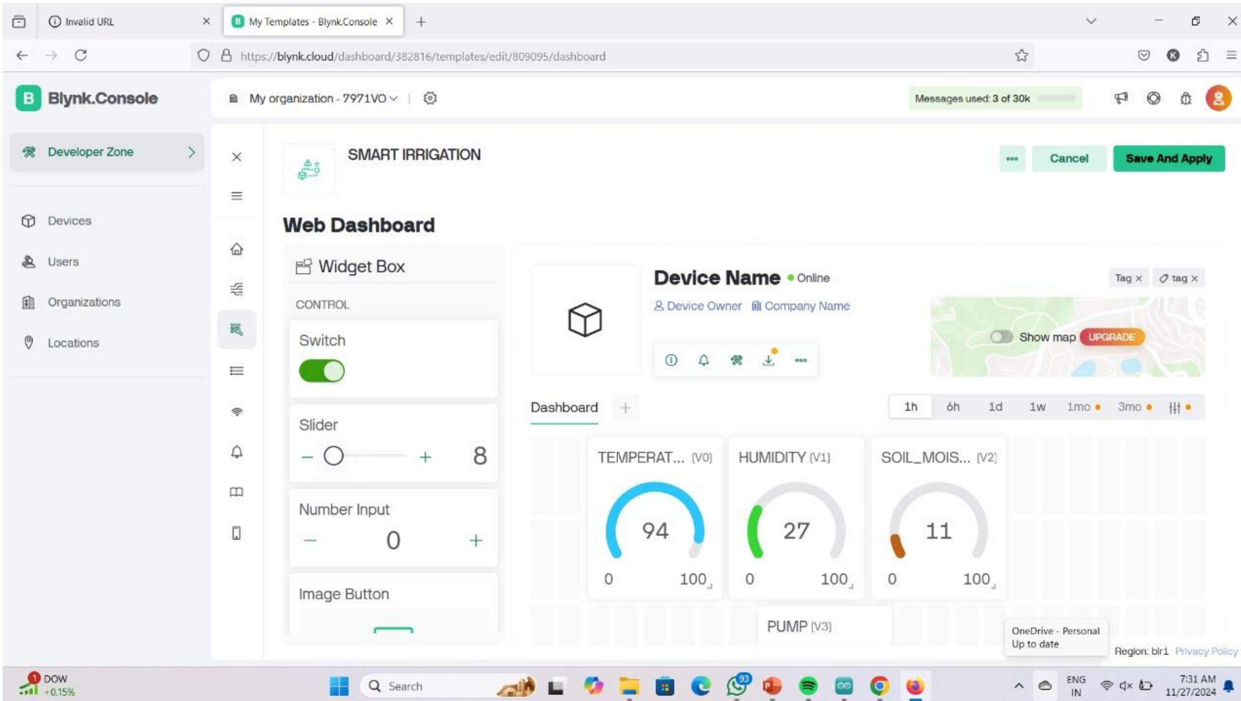
1. BLYNK DASHBOARD



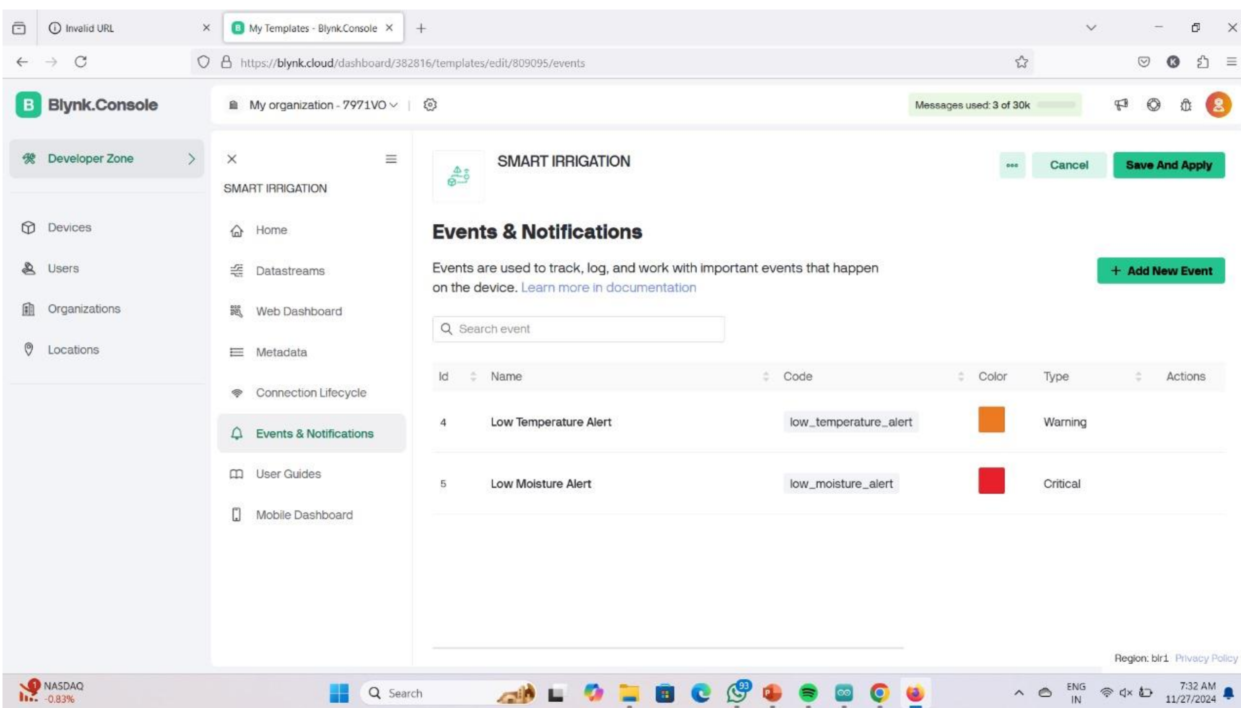
2. DATASTREAM PAGE



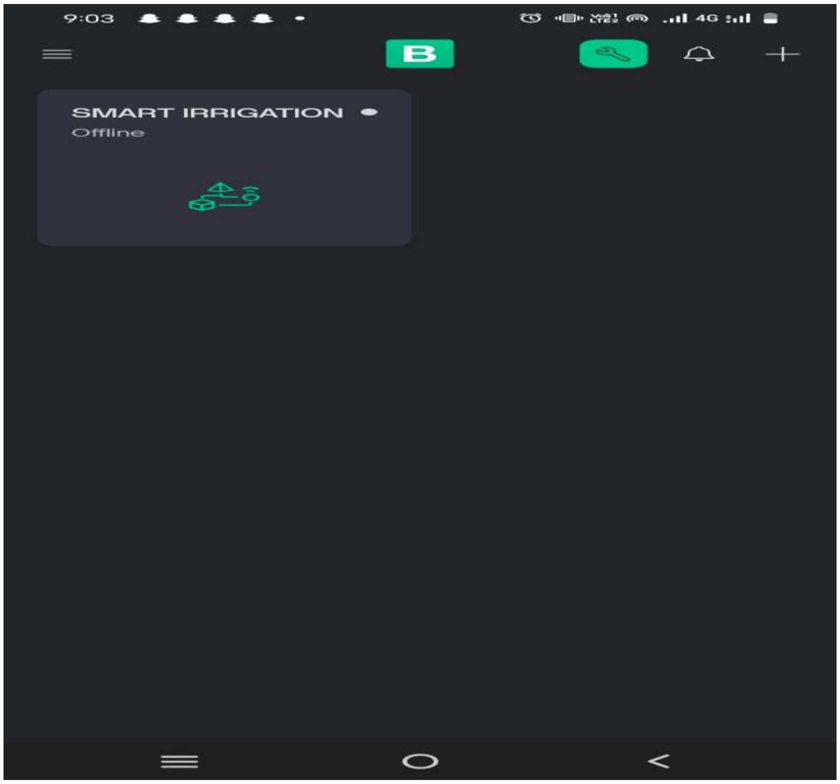
3. WEB DASHBOARD



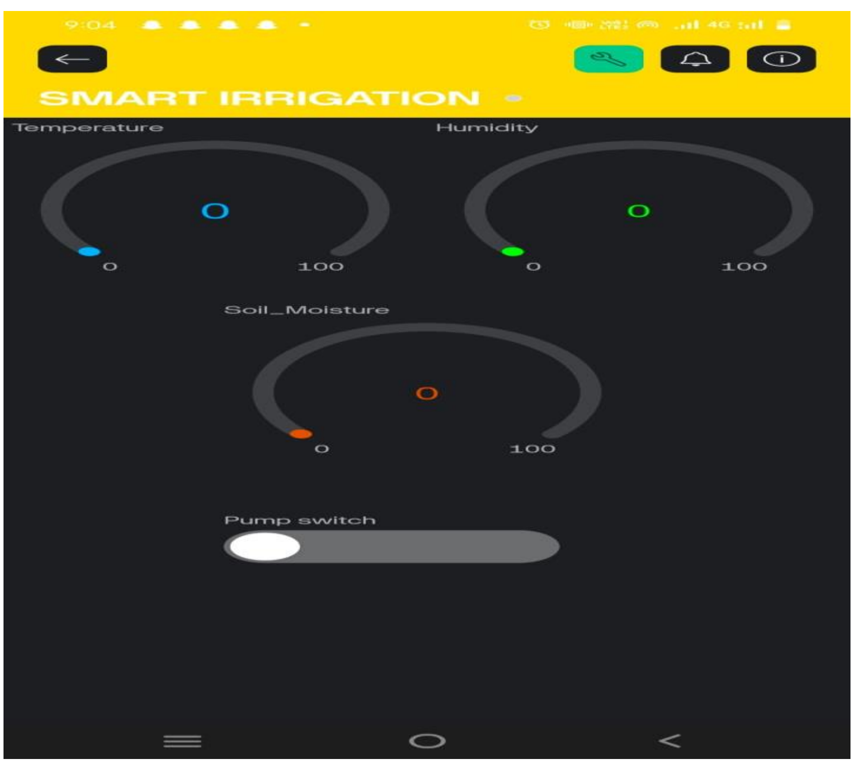
4. EVENT AND NOTIFICATION



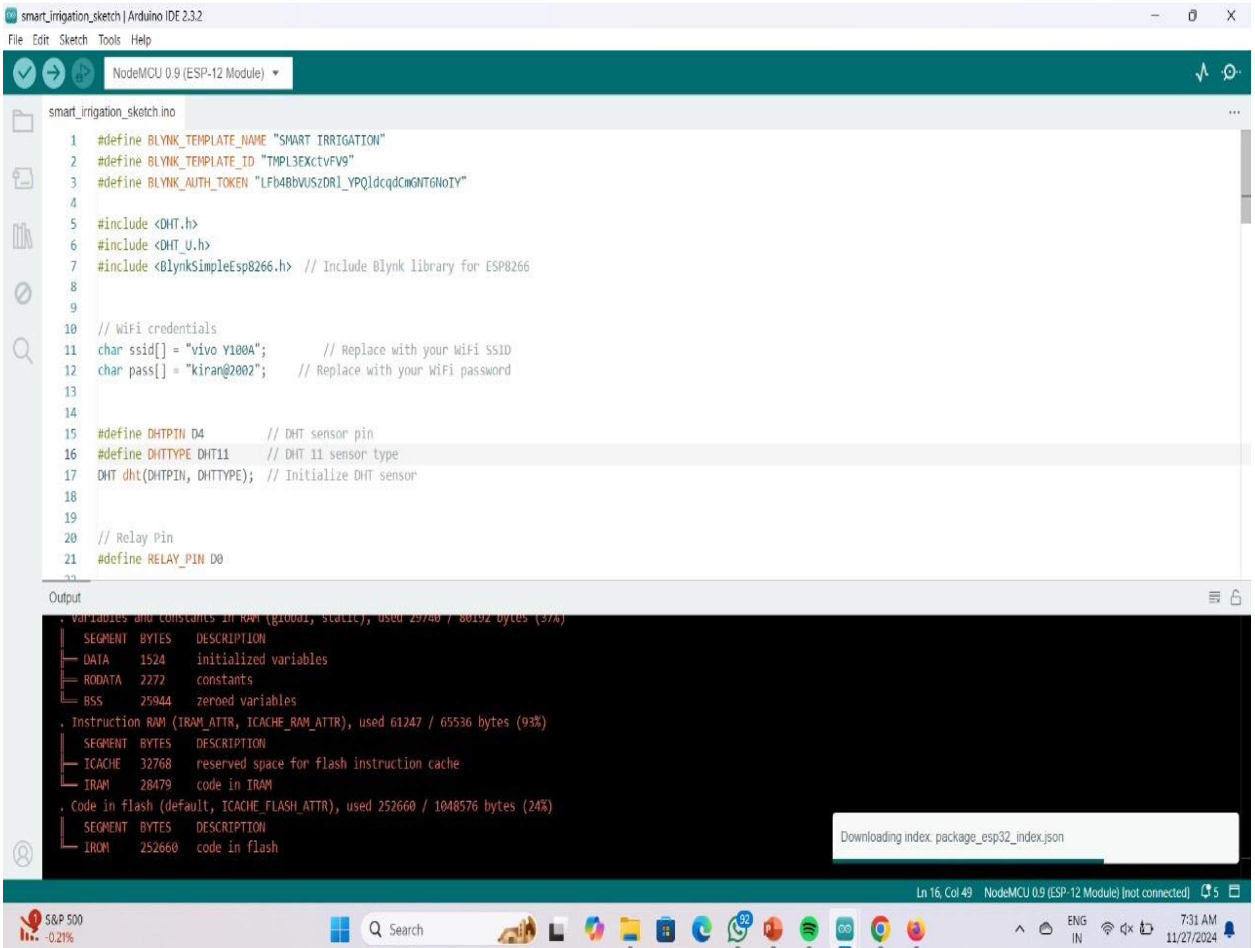
5. MOBILE BLYNK APP



6. MOBILE DASHBOARD



APPENDIX II : OUTPUT IN BACKEND (ARDIUNO IDE)



8. BIBLIOGRAPHY:

1. **YouTube Channel: Sruti Hobby Projects. (2022).** ***Smart Irrigation System Using IoT and Blynk*****. Retrieved from** **https://youtu.be/rmOg9qA_iV4?si=XVNoxwAo8lliHIZf**

    - **This video provides an in-depth tutorial on building a smart irrigation system using IoT, covering the setup of the NodeMCU, sensors, and Blynk app integration for real-time monitoring and control.**

2. **YouTube Channel: The Engineering Projects. (2021).** ***IoT Based Smart Irrigation System with NodeMCU ESP8266*****. Retrieved from** **https://youtu.be/QIdERbxDVuM?si=r1BpG488viD2ACe0**

- **This video guides the viewer through the process of constructing an IoT-based smart irrigation system using the NodeMCU ESP8266 and explains how to interface sensors and control the system remotely with the Blynk app.**

3. **YouTube Channel: IoT Tutorials. (2023). *Quick Guide to Smart Irrigation System Using NodeMCU and Blynk App*. Retrieved from https://youtube.com/shorts/oHwV6ortkWg?si=pM7rFdVF_qOPaPSj**
    - **A short tutorial demonstrating how to quickly set up a smart irrigation system using NodeMCU and Blynk, showing the key steps for beginners.**
4. **YouTube Channel: Learn Robotics. (2021). *Complete IoT Smart Irrigation System with NodeMCU and Blynk*. Retrieved from https://youtu.be/trvACRD9OFo?si=U2vb9i0bM4jCKzNa**
    - **This video explores a complete IoT-based smart irrigation system project, explaining the entire process from hardware setup to programming and integrating with the Blynk app for efficient irrigation control.**