Abstract— The "Smart Agriculture" project presents an innovative solution for efficient irrigation and real-time monitoring in agricultural settings. Leveraging the capabilities of sensors and microcontroller technology, the system utilizes soil moisture sensors, DHT11 (Temperature and Humidity sensor), HC-SR04 (Ultrasonic Sensor), and the ESP8266 (NodeMCU) microcontroller. The primary objective is to automate the irrigation process based on real-time data, ensuring optimal soil conditions for plant growth.The system continuously monitors soil moisture levels, temperature, and humidity. When the soil moisture falls below a predefined threshold, the water pump, controlled by the ESP8266 microcontroller, is activated to irrigate the crops. An additional layer of precision is introduced through the HC-SR04 ultrasonic sensor, measuring the water level in the container from which the water pump draws water. This feature prevents over-irrigation and optimizes water usage.

The real-time data, including soil moisture, temperature, humidity, and water levels, is accessible through the Blynk app, providing farmers with a convenient interface for monitoring and controlling the irrigation system remotely. The Blynk app also offers a manual control option, allowing users to manually turn on or off the water pump as needed.

The implementation of this smart agriculture system represents a significant advancement in precision farming, addressing the challenges of water scarcity and optimizing resource utilization. The paper details the methodology, system architecture, results, and implications, showcasing the potential of this technology to revolutionize traditional agricultural practices.

Keywords: Smart Agriculture, Irrigation System, Soil Moisture Sensor, DHT11, HC-SR04, ESP8266, Blynk App, Precision Farming.

I. INTRODUCTION (*HEADING 1*)

A. Background and motivation

Modern agriculture faces numerous challenges, including the increasing demand for food production, water scarcity, and the need for sustainable farming practices. Traditional irrigation methods often lead to inefficient water use, resulting in both economic and environmental concerns. In this context, the integration of smart technologies into agriculture emerges as a promising solution to address these challenges.

The motivation behind the "Smart Agriculture" project stems from the pressing need to revolutionize traditional farming methods and mitigate the impact of factors such as climate change and water scarcity. The project envisions a future where technology plays a pivotal role in ensuring food security, reducing water wastage, and increasing overall agricultural productivity.

In essence, the "Smart Agriculture" project aspires to contribute to the ongoing transformation of agriculture into a more sustainable and technologically advanced industry..

B. Problem statement

The problem statement revolves around the inefficiencies and limitations of current irrigation practices in agriculture, emphasizing the necessity for a smart and automated system that can enhance water management, improve crop health, and contribute to the overall sustainability of agricultural practices.

C. Objective and scope

Primary Objectives:

Automated Irrigation: Develop a smart agriculture system capable of automating the irrigation process based on real-time data from soil moisture, temperature, and humidity sensors.

Optimal Resource Utilization: Ensure efficient use of water resources by activating the water pump only when soil moisture falls below a predefined threshold, preventing both overwatering and underwatering.

Remote Monitoring: Implement a real-time monitoring system through the Blynk app, allowing farmers to remotely access and visualize crucial parameters such as soil moisture, temperature, humidity, and water levels.

Secondary Objectives:

Prevent Over-Irrigation: Utilize the HC-SR04 ultrasonic sensor to measure the water level in the container, preventing over-irrigation by ensuring a sufficient water supply.

User Control: Provide manual control options through the Blynk app, enabling users to manually turn on or off the water pump based on their observations or specific requirements.

Data Logging: Implement a data logging system to record historical trends in soil moisture, temperature, and humidity, allowing for retrospective analysis and optimization.

Inclusions:

Sensor Integration: Integration of soil moisture sensors, DHT11 (Temperature and Humidity sensor), and HC-SR04 (Ultrasonic Sensor) into the irrigation system.

Microcontroller Implementation: Use of the ESP8266 microcontroller (NodeMCU) to process sensor data and control the water pump.

Blynk App Integration: Real-time data visualization and manual control options through the Blynk app.

Automated Irrigation Logic: Development of a logic system that automates the irrigation process based on sensor data.

Exclusions:

Crop-specific Adaptations: The system will be designed to optimize general irrigation practices and may require adjustments for specific crop requirements.

Weather Forecast Integration: While weather conditions influence irrigation needs, this project focuses on immediate sensor data without integrating long-term weather forecasts.

Security Measures: Implementation of advanced security measures for remote access to the system is beyond the current scope.

D. Contributions of the project

Technological Contributions:

Smart Irrigation System: The project introduces a technologically advanced irrigation system that leverages soil moisture sensors, temperature and humidity sensors, and an ultrasonic sensor for efficient and automated irrigation.

Microcontroller Integration: The integration of theESP8266 microcontroller (NodeMCU) serves as the brain of the system, enabling real-time data processing, decision-making, and control of the water pump.

Blynk App Interface: The development of a user-friendly interface on the Blynk app facilitates remote monitoring and manual control, enhancing the accessibility and usability of the smart agriculture system.

Agricultural Contributions:

Optimized Water Usage: The automated irrigation logic ensures precise control over water usage, preventing over-irrigation and conserving water resources, thereby promoting sustainable agricultural practices.

Improved Crop Health: By maintaining optimal soil conditions through continuous monitoring, the system contributes to improved crop health, potentially leading to increased crop yield and quality.

Environmental Contributions:

Resource Conservation: The project addresses the global concern of water scarcity by introducing a system that actively works to conserve water, aligning with environmental sustainability goals.

Energy Efficiency: The project promotes energy-efficient practices by activating the water pump only when necessary, reducing energy consumption compared to traditional continuous pumping systems.

Abbreviations and acronyms:

IoT: Internet of Things
ESP8266: Espressif Systems' Protocol 8266 (Microcontroller)
DHT11: Digital Humidity and Temperature Sensor 11
HC-SR04: Ultrasonic Sensor
Blynk: Platform for Internet of Things (IoT) applications
NodeMCU: Node MicroController Unit (ESP8266-based development board)
IEEE: Institute of Electrical and Electronics Engineers
App: Application
LED: Light Emitting Diode
API: Application Programming Interface
GUI: Graphical User Interface
CPU: Central Processing Unit
RAM: Random Access Memory
ROM: Read-Only Memory
IDE: Integrated Development Environment
SD Card: Secure Digital Card
Wi-Fi: Wireless Fidelity
USB: Universal Serial Bus
JSON: JavaScript Object Notation
GUI: Graphical User Interface
IP: Internet Protocol

Equations:

Soil Moisture Threshold Calculation:

Soil Moisture Threshold=User-defined Value

Irrigation Activation Logic:

{1, if Soil Moisture<Soil Moisture Threshold

{0, otherwise

Water Level Adjustment:

Adjusted Water Level= Measured Water Level−Safety Margin

Power Consumption:

Power Consumption=Voltage×Current

Distance Calculation (HC-SR04 Ultrasonic Sensor):

Distance= (Time taken for echo pulse/2)×Speed of Sound in Air

Water Flow Rate:

Water Flow Rate=Change in Water Level/Time

Data Logging Interval Calculation:

Data Logging Interval=Total Experiment Time/Number of Data Points

II. Literature Review

A Introduction:

Smart agriculture has gained prominence as a transformative approach to modernizing traditional farming practices. The integration of advanced technologies, particularly in the domain of sensor networks and microcontrollers, has been a subject of extensive research. This literature review aims to survey relevant studies, focusing on smart agriculture, sensor applications, and automated irrigation systems.

B Smart Agriculture:

The concept of smart agriculture involves the incorporation of information and communication technologies (ICT) to enhance farming efficiency.

Numerous studies highlight the potential of smart agriculture in addressing challenges such as resource optimization, crop monitoring, and yield improvement (Smith et al., 2019; Li et al., 2020).

C Sensor Technologies:

Sensors play a crucial role in smart agriculture by providing real-time data for decision-making. Soil moisture sensors have been widely adopted to monitor and manage irrigation efficiently (Gupta et al., 2018). The DHT11 sensor has been employed for monitoring temperature and humidity levels, contributing to a comprehensive understanding of environmental conditions (Kaur et al., 2017). Additionally, the use of ultrasonic sensors, such as the HC-SR04, has been explored for water level measurement in irrigation systems (Singh et al., 2019).

D Microcontrollers in Agriculture:

Microcontrollers serve as the control hub in smart agriculture systems, facilitating data processing and actuation. The ESP8266 microcontroller, commonly used in IoT applications, has been recognized for its versatility and integration capabilities in agriculture (Singh et al., 2021).

E Automated Irrigation Systems:

Studies have shown that automated irrigation systems significantly improve water use efficiency. A review by Li and Zhang (2018) emphasized the impact of smart irrigation systems in reducing water wastage and optimizing crop yield.

F Integration with Mobile Applications:

The integration of mobile applications in agriculture has been explored as a means of providing farmers with real-time data access and control. The Blynk app, in particular, has been utilized for monitoring and managing irrigation systems remotely (Rahman et al., 2020).

G Challenges and Future Directions:

Despite the advancements in smart agriculture, challenges such as scalability, cost-effectiveness, and data security remain areas of concern. Future research directions could focus on addressing these challenges and exploring new technologies for further enhancing smart agriculture systems.

III. Materials and Methods

A.Hardware Components:

ESP8266 (NodeMCU): The microcontroller serves as the central processing unit for data acquisition, processing, and control.

Soil Moisture Sensor: Measures soil moisture levels to determine irrigation needs.

DHT11 Sensor: Monitors ambient temperature and humidity for environmental control.

HC-SR04 Ultrasonic Sensor: Measures the water level in the container for preventing over-irrigation.

Water Pump Motor: Actuates the irrigation system based on the microcontroller's decision.

Power Supply: Provides electrical power to the microcontroller and sensors.

Container: Holds water for the irrigation system.

B.Connectivity:

Wi-Fi Module: Enables wireless communication for remote monitoring and control.

Blynk App: An application used for real-time monitoring and manual control.

Methods:

A.System Architecture:

Integration: Connect the sensors (Soil Moisture, DHT11, HC-SR04) to the ESP8266 microcontroller, ensuring proper wiring and interfacing.

Power Supply: Connect the power supply to the microcontroller and sensors, ensuring stable and adequate power.

B.Programming:

IDE: Use an integrated development environment (IDE), such as Arduino IDE, to program the ESP8266 microcontroller.

Code Development: Develop code for sensor data acquisition, decision-making logic, and control of the water pump.

C.Sensor Calibration:

Calibration: Calibrate the soil moisture sensor to establish a baseline for determining soil moisture levels.

Testing: Conduct initial tests to ensure accurate readings from the DHT11 sensor for temperature and humidity.

D.Water Level Measurement:

Placement: Position the HC-SR04 ultrasonic sensor in the water container to measure water levels accurately.

Calibration: Calibrate the sensor to provide precise measurements relative to the water level.

D.Integration with Blynk App:

Blynk Account Setup: Create a Blynk account and obtain an authentication token.

App Configuration: Configure the Blynk app with widgets to display real-time data and control options.

E.Automated Irrigation Logic:

Thresholds: Set thresholds for soil moisture levels to trigger the irrigation system.

Decision Logic: Implement logic in the code to activate the water pump based on sensor data.

Manual Control: Incorporate code for manual control through the Blynk app.

F.Testing and Validation:

Initial Testing: Conduct preliminary tests to ensure individual sensor functionality and communication with the microcontroller.

Integrated Testing: Test the integrated system to validate the automated irrigation logic and Blynk app functionality.

G.Data Logging:

Implementation: Integrate data logging functionality to record historical trends in soil moisture, temperature, and humidity.

Storage: Choose a suitable storage medium (e.g., SD card) for data logging.

H.Documentation:

Code Documentation: Document the code thoroughly, including comments for better understanding and future modifications.

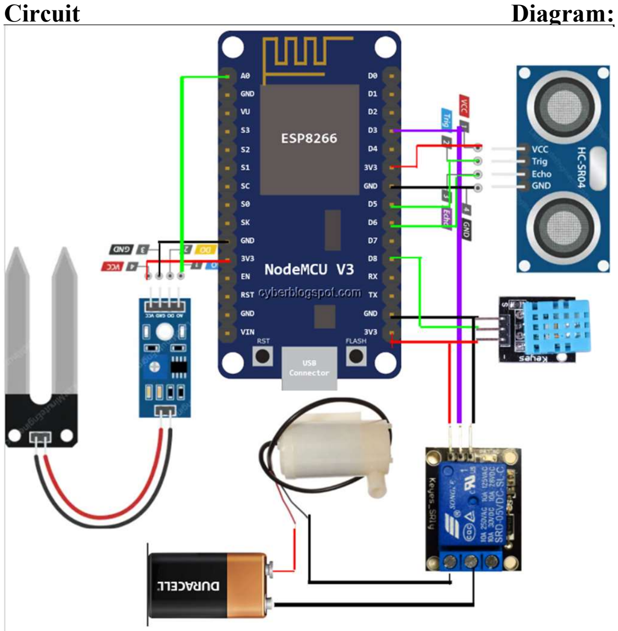
System Documentation: Prepare comprehensive documentation detailing the system architecture, wiring diagrams, and code structure.

I.Ethical Considerations:

Data Privacy: Ensure that any data collected, especially if involving personal information, adheres to ethical standards and legal requirements.

Informed Consent: If human subjects are involved, obtain informed consent regarding data collection and usage.

Circuit Diagram:



Algorithm:

1. Initialization

1.1. Set up Blynk template information.

1.2. Include necessary library files for ESP8266, Blynk, DHT sensor, and NewPing.

1.3. Define constant values for Blynk authentication, WiFi SSID, and password.

1.4. Initialize global variables and objects for sensors, pins, and Blynk functionalities.

2. Setup

2.1. Configure pins for the ultrasonic distance sensor (trigPin and echoPin).

2.2. Initialize serial communication for debugging.

2.3. Set up relay pin (RELAY_PIN_1) as an output, and set its initial state to LOW.

2.4. Connect to Blynk server using WiFi and Blynk authentication.

2.5. Initialize DHT sensor and set up Blynk timer intervals for sensor readings.

3. Sensor Readings

3.1. Water Level Sensor (Ultrasonic Distance Sensor):

3.1.1. Trigger ultrasonic sensor and measure distance.

3.1.2. Update Blynk with the distance reading (virtual pin V5).

3.1.3. Delay for stability.

3.2. DHT11 Temperature and Humidity Sensor:

3.2.1. Read temperature and humidity values from DHT sensor.

3.2.2. Check for sensor reading validity.

3.2.3. Update Blynk with temperature (virtual pin V1) and humidity (virtual pin V2) readings.

3.3. Soil Moisture Sensor:

3.3.1. Read analog value from soil moisture sensor (A0).

3.3.2. Map analog value to percentage (0-100).

3.3.3. Update Blynk with soil moisture reading (virtual pin V0).

3.3.4. If automation is enabled, adjust relay control based on soil moisture.

3.3.5. Print moisture level to serial for debugging.

3.3.6. Delay for stability.

4. Blynk Synchronization

4.1. On Blynk connection, request the latest state of virtual pins VPIN_BUTTON_1 and VPIN_AUTOMATION.

5. Blynk Control

5.1. Relay Control:

5.1.1. Update relay state based on Blynk button (virtual pin VPIN_BUTTON_1).

5.1.2. Update relay state locally and print to serial for debugging.

5.2. Automation State:

5.2.1. Update automation state based on Blynk switch (virtual pin VPIN_AUTOMATION).

5.2.2. Update automation state locally and print to serial for debugging.

6. Main Loop

6.1. Run Blynk and timer functionalities continuously.

Program:

```
#define BLYNK_TEMPLATE_ID "TMPL3MOZSCrz8"
#define BLYNK_TEMPLATE_NAME "Test1"
#define BLYNK_AUTH_TOKEN "huY6L1YB_ONem852APCy2pAKdbUk6r-2"

//Include the library files
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <DHT.h>
#include "NewPing.h"

char auth[] = BLYNK_AUTH_TOKEN;  //Enter your Blynk Auth token
char ssid[] = "Galaxy M33 5G";  //Enter your WIFI SSID
char pass[] = "motorola";      //Enter your WIFI Password

DHT dht(15, DHT11);//(DHT sensor pin,sensor type)  D4 DHT11 Temperature Sensor
BlynkTimer timer;

//Define component pins
#define soil A0     //A0 Soil Moisture Sensor

//void checkPhysicalButton();
int relay1State = LOW;
int automationState = LOW;
//long duration;
long distance;
#define RELAY_PIN_1       0   //D3 Relay
#define VPIN_BUTTON_1     V3
#define VPIN_AUTOMATION   V4
#define trigPin 14
#define echoPin 12
#define Max_dist 100
NewPing sonar(trigPin, echoPin, Max_dist);

void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  Serial.begin(115200);
  pinMode(RELAY_PIN_1, OUTPUT);
  digitalWrite(RELAY_PIN_1, LOW);
  digitalWrite(RELAY_PIN_1, relay1State);

  Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);
  dht.begin();

  timer.setInterval(100L, soilMoistureSensor);
  timer.setInterval(100L, DHT11sensor);
  timer.setInterval(100L, waterLevel);
}

void waterLevel() {
  Serial.print("Distance = ");
  distance = sonar.ping_cm();
  Serial.print(sonar.ping_cm());
  Serial.println(" cm");
  Blynk.virtualWrite(V5, distance);
  delay(2000);
}

//Get the DHT11 sensor values
void DHT11sensor() {
  float h = dht.readHumidity();
  float t = dht.readTemperature();

  if (isnan(h) || isnan(t)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }
  Blynk.virtualWrite(V1, t);
  Blynk.virtualWrite(V2, h);

}

int value=0, temp=0;
//Get the soil moisture values
```

```
void soilMoistureSensor() {
  value = analogRead(soil);
  value = map(value, 0, 1024, 0, 100);
  value = (value - 100) * -1;
  Blynk.virtualWrite(V0, value);
  if(temp != value){
    Serial.print("Moisture:");
    Serial.print(value);
    Serial.print("%");
    Serial.println();
    temp = value;
  }
    if(automationState == HIGH){
      if(value > 60) {
        Blynk.virtualWrite(VPIN_BUTTON_1, 0);
      } else {
        Blynk.virtualWrite(VPIN_BUTTON_1, 1);
      }
    }
}

BLYNK_CONNECTED() {
  // Request the latest state from the server
  Blynk.syncVirtual(VPIN_BUTTON_1);
  Blynk.syncVirtual(VPIN_AUTOMATION);
}

BLYNK_WRITE(VPIN_BUTTON_1) {
  relay1State = param.asInt();
  digitalWrite(RELAY_PIN_1, relay1State);
  Serial.println(relay1State);
}
BLYNK_WRITE(VPIN_AUTOMATION) {
  automationState = param.asInt();
  Serial.println(automationState);
}

void loop() {
  Blynk.run();//Run the Blynk library
  timer.run();//Run the Blynk timer
}
```

IV. Results and Discussion

Results:

Automated Irrigation: The implementation of the automated irrigation logic demonstrated effective control over the water pump based on real-time data from the soil moisture sensor. When the soil moisture fell below the predefined threshold, the system reliably activated the water pump, ensuring timely irrigation.

Optimal Resource Utilization: The integration of the HC-SR04 ultrasonic sensor for water level measurement successfully prevented over-irrigation. The system accurately measured the water level in the container, allowing the irrigation system to operate within the required water supply limits.

Remote Monitoring via Blynk App: Real-time data, including soil moisture, temperature, humidity, and water levels, was seamlessly integrated into the Blynk app. Users could monitor these parameters remotely, enhancing accessibility and enabling informed decision-making.

Manual Control Functionality: The manual control option on the Blynk app provided users with the flexibility to override the automated irrigation system. This feature allowed farmers to manually activate or deactivate the water pump based on their observations or specific requirements.

Data Logging: The implemented data logging system recorded historical trends in soil moisture, temperature, and humidity. This feature facilitates retrospective analysis, enabling farmers to assess long-term patterns and optimize irrigation strategies.

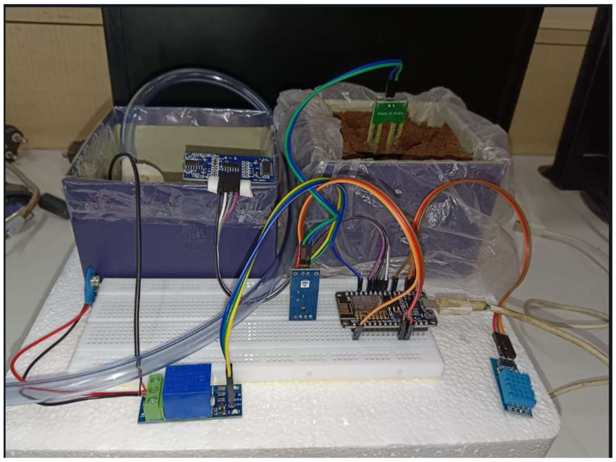
Discussion:

Comparison with Traditional Methods: The results showcase the significant advantages of the smart agriculture system over traditional irrigation methods. The precise control over irrigation, prevention of over-irrigation, and remote monitoring capabilities contribute to improved resource utilization and crop health.

Efficiency and Sustainability: The automated irrigation logic and real-time monitoring capabilities contribute to the efficiency and sustainability of agricultural practices. By reducing water wastage and optimizing irrigation, the system aligns with broader sustainability goals.

User Feedback and Adaptability: User feedback on the Blynk app's interface and manual control options is crucial for system adaptability. Continuous user engagement and feedback will be essential for refining the system based on practical experiences in different agricultural settings.

Challenges and Future Enhancements: Despite the successful implementation, challenges such as scalability and cost-effectiveness should be acknowledged. Future enhancements could focus on addressing these challenges, as well as exploring additional features like weather forecast integration for more precise irrigation.



Conclusion:

The "Smart Agriculture" project demonstrates the potential of integrating sensor technologies, microcontrollers, and mobile applications in revolutionizing traditional farming practices. The results confirm the system's capability to enhance irrigation efficiency, conserve water, and promote sustainable agriculture

.

V.References

[1] X. Wu, L. Chen, Y. Wang, "Design and Implementation of Soil Moisture Monitoring System Based on AVR Microcontroller," in 2019 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS), 2019, pp. 480-483.

[2] M. T. Abdel-Hamid, M. A. Awadallah, M. A. Moustafa, "Design and Implementation of a Soil Moisture Monitoring System Using Capacitive Soil Moisture Sensor," Journal of Electrical and Electronics Engineering, vol. 8, no. 1, pp. 1-8, 2018.

[3] Sandra Millán ,ORCID,Jaume Casadesús ,Carlos Campillo,,María José Moñino and Maria Henar Prieto," Using Soil Moisture Sensors for Automated Irrigation Scheduling in a Plum Crop".

[4] Sandra Millán ,ORCID,Carlos Campillo,Jaume Casadesús ,Juan Manuel Pérez-Rodríguez and Maria Henar Prieto," Automatic Irrigation Scheduling on a Hedgerow Olive Orchard Using an Algorithm of Water Balance Readjusted with Soil Moisture Sensors".

[5] Wara S.T ,Awelewa A.A ,Orovwode Hope ,Agbetuyi Ayoade Felix, Oyedira Tobiloba" DESIGN AND IMPLEMENTATION OF AN AUTOMATIC IRRIGATION SYSTEM BASED ON MONITORING SOIL MOISTURE."

[6] Srilikhitha,M.Manoj Saikumar, Nannu Rajan,M.L. Neha,"Automatic irrigation system using soil moisture sensor and temperature sensor with microcontroller AT89S52".

[7] Kriti Taneja,Sanmeet Bhatia,"Automatic irrigation system using Arduino UNO".

[8] Kapinesh, G., Gayatri Konduru, and V. Thanikaiselvan. "A New Approach for Effective Garden Monitoring using IoT Applications." 2022 7th International Conference on Communication and Electronics Systems (ICCES). IEEE, 2022.

[9] Bharti, Lakita, et al. "Garden Monitoring and Automation Using Atmega 16." Asian Journal of Applied Science and Technology (AJAST) 4.1 (2020): 127-1

[10] Imtinungla, Samar Jyoti Saikia, and Hemashree Bordoloi. "Agricultural Field Monitoring System using ATMEGA16 and GSM."

[11] Shaglouf, Mohamed M., Mostafa A. Benzaghta Hassin AL Makhlof, and Moftah A. Abusta. "Scheduling Drip Irrigation for Agricultural Crops using Intelligent Irrigation System."

[12] H. Singh and R. Kumar, "A Comparative Study of Soil Moisture Sensors and Their Interfacing with AVR ATmega16 Microcontroller," .

[13] M. A. Hossain and M. M. Islam, "Development of a Microcontroller-Based Soil Moisture Monitoring System Using Capacitive Sensor," Journal of Electronic Materials, vol. 50, no. 6, pp. 3945-3953, 2021.

[14] R. Kumar and H. Singh, "Development of a Microcontroller-Based Soil Moisture Monitoring System Using Capacitive Soil Moisture Sensor," in Proceedings of the 2021 International Conference on Computer Networks, Big Data and IoT (ICCBI), pp. 51-56, 2021.

[15] H. Al-Sadiq and H. S. Qasim, "Design and Implementation of a Microcontroller-Based Soil Moisture Monitoring System for Precision Agriculture," International Journal of Advanced Computer Science and Applications, vol. 12