RV COLLEGE OF ENGINEERING
Bengaluru-560 059

REPORT ON

EXPERIENTIAL LEARNING

ACY 2024-25

THEME

Environment

Title of the Project

Real Time Soil Detector

Student(s)

USN	Name
1RV24IS005	AAYUSHI PRIYA
1RV24EC018	AINESH MALLICK
1RV24CH029	RISHABH GUPTA
1RV24AS042	RISHIT KUMAR

Mentor

Name, Designation & Department
Dr. Bhuvaneswara Babu T Professor Department of Physics

Project Evaluators

SL. No.	Name of the Evaluators (Enter the Names in Capitals)
1.	DR. BHUVANESWARA BABU T (PHYSICS)
2.	DR SOUMYA A (CSE)
3.	DR NANJUNDARADHYA N V (ME)
4.	DR. VEENA DIVYA KRISHNAPPA (EC)

1.Introduction

Agriculture is a cornerstone of the Indian economy, providing livelihoods to a large portion of the population. However, many farmers still rely on traditional practices that lack real-time feedback mechanisms, resulting in inefficient resource usage, inconsistent yields, and avoidable crop failures. To modernize this sector, there is an urgent need for systems that can monitor soil conditions continuously and automatically manage irrigation.

Smart agriculture, powered by embedded systems and IoT technologies, presents a transformative opportunity to address these limitations. Among various environmental factors, soil moisture and pH are two crucial parameters that determine the health and productivity of crops. Inappropriate levels of either can lead to suboptimal plant growth, nutrient deficiency, and reduced yields.

This project introduces a Real-Time Soil Detector—a compact, battery-operated device that continuously measures soil moisture and pH levels using low-cost sensors. The system is built around the ESP32 microcontroller and includes a relay-controlled water pump for automatic irrigation, an LCD screen for real-time monitoring, and a Blynk IoT interface for remote access. It has been carefully designed to be portable, affordable, and scalable.

The project's development process included planning, hardware integration, software programming, calibration, and rigorous testing. Various tools such as Arduino IDE, Wi-Fi modules, LCD displays, and analog sensors were used. Real-world testing validated its effectiveness, with the device reliably triggering irrigation below a preset moisture threshold and delivering accurate pH values.

By supporting smart farming goals, this project serves as a scalable model for deploying low-cost precision agriculture tools in small and medium-scale farms. Its successful implementation also sets the stage for future enhancements, such as cloud data logging, solar power integration, and additional sensor inputs. Overall, the Real-Time Soil Detector demonstrates how embedded systems can empower farmers with timely, actionable insights while promoting sustainable agricultural practices.

2.Problem Definition

2.1. Problem Statement:

In many parts of India, including urban and peri-urban agricultural regions like Bangalore, farmers face the challenge of making critical irrigation and soil treatment decisions without access to real-time data on soil conditions. Two essential parameters—soil moisture and soil pH—directly influence plant growth, nutrient absorption, and overall yield quality. The absence of real-time monitoring leads to inefficient water usage, delayed intervention, poor crop health, and increased resource wastage.

Most existing solutions are either too expensive, infrastructure-heavy, or require technical expertise, making them unsuitable for small and marginal farmers. Manual testing techniques like using moisture probes or pH strips are not only time-consuming and inconsistent but also unsuitable for frequent or continuous monitoring.

Thus, there is a pressing need for a low-cost, compact, and automated solution that enables continuous monitoring of soil conditions and intelligent irrigation management. A solution that is reliable, user-friendly, energy-efficient, and deployable in both urban rooftop gardens and remote rural farmlands can significantly benefit sustainable agriculture practices.

2.2. Background Information: (literature review)

SL NO.	**Topic/ Focus Area**	**Key Insights from Literature**	**Source/ Author**
1	Importance of Soil Moisture Monitoring	Soil moisture impacts nutrient transport, root health, and plant growth. Real-time monitoring improves yield.	Li et al., IEEE Access, 2021 [1]
2	Role of Soil pH in Agriculture	Soil pH determines nutrient availability and absorption. Imbalanced pH causes poor crop performance.	Patel & Deshmukh, Agri Research Bulletin, 2020 [3]
3	Limitations of Traditional Methods	Manual testing (probes/strips) is slow, inconsistent, and not	Singh et al., IJAB, 2022 [2]

		scalable for frequent monitoring.	
4	Use of ESP32 and IoT in Smart Farming	ESP32-based systems offer cost-effective, wireless-enabled solutions for field sensing and automation.	Kumar, Int. J. Embedded Systems, 2021 [4]
5	Precision Agriculture & IoT Integration	IoT platforms like Blynk enhance accessibility of real-time data via smartphones, improving farm management.	Li et al., IEEE Access, 2021 [1]
6	Need for Low-Cost, Scalable Systems	Most commercial systems are expensive or complex. Open-source alternatives are preferred for rural deployment.	Review of multiple case studies and experimental projects

3.Objectives

3.1. Primary Objectives:

The primary objectives of this project are designed to address key challenges in the agriculture sector, especially in regions with limited access to smart farming technologies. The system aims to:

i. **Develop a real-time soil detection system:** Enable the continuous monitoring of soil moisture and pH levels using sensors interfaced with an ESP32 microcontroller.

ii. **Automate irrigation processes:** Use real-time soil moisture data to control a water pump through a relay, thereby conserving water and reducing manual intervention.

iii. **Provide real-time data display:** Display critical soil parameters such as moisture percentage and pH level on a 16x2 LCD screen, enabling farmers to take immediate action when necessary.

iv. **Ensure accuracy through calibration:** Employ mapping and constraint techniques to convert raw analog values into meaningful environmental readings.

v. **Establish robust system performance:** Design a stable embedded system with reliable sensor integration, actuator control, and visual/auditory feedback mechanisms.

These objectives collectively aim to promote precision farming, reduce resource wastage, and support environmentally sustainable agricultural practices.

3.2. Secondary Objectives:

In addition to the core functionalities, the project also sets forth supplementary goals to enhance usability, adaptability, and future development potential:

i. **Portability and Self-Powered Operation:** Implement a battery-operated power system using rechargeable 18650 cells, with voltage regulation via TP4056 and MT3608 modules for portability in off-grid rural areas.

ii. **Remote Monitoring Capability:** Integrate Wi-Fi functionality and Blynk IoT platform to allow farmers and users to remotely view live sensor data and pump status from a smartphone.

iii. **User-Friendly Interface:** Include features such as a buzzer for alerts and LED indicators to offer instant status updates in low-visibility conditions.

iv. **Scalability and Modularity:** Design the system architecture to be modular, allowing easy expansion to include additional sensors (e.g., temperature, humidity, NPK) or functionalities (e.g., automated fertilization).

v. **Foundation for IoT Expansion:** Create a foundation for future integration with cloud services, enabling data logging, predictive analytics, and dashboard visualization tools for farm management.

Together, these objectives ensure that the system is not only functional but also practical, scalable, and future-ready for the growing demands of smart agriculture.

- Make the system portable and battery-powered.
- Lay groundwork for future cloud/IoT integration.

4. Methodology

4.1 Approach:

The system is designed to sense soil moisture and pH levels using analog sensors connected to an ESP32 microcontroller. Based on moisture threshold values, a relay module is used to control a small water pump that irrigates the soil. The entire system is powered by rechargeable 18650 batteries regulated by a 7805 and MT3608 module, making it suitable for remote field use. Sensor values are displayed in real time on a 16x2 LCD interfaced via I2C.

i. **Start** – Power up the ESP32 and initialize all components.
ii. **Sensor Initialization** – Configure the ADC pins and I2C communication for LCD.
iii. **Data Acquisition** – Continuously read values from the soil moisture and pH sensors.
iv. **Display Update** – Output real-time sensor readings to the LCD display.
v. **Decision Making** – If moisture < threshold, activate relay and turn ON pump.
vi. **Looping** – Keep monitoring in real time, updating display and pump status.

4.2 Procedures:

1. **Component Selection:** Selected ESP32 for its dual-core processing, low power use, and inbuilt Wi-Fi (for future scope). Soil moisture and pH sensors were chosen based on local availability and sensitivity.
2. **Circuit Design:** Designed using breadboard initially for easy reconfiguration. Used TP4056 for battery charging and MT3608 for voltage boosting.
3. **Hardware Assembly:** Interfaced all components on a breadboard. Carefully managed power routing to avoid voltage mismatch.
4. **Software Development:** Programmed ESP32 in Arduino IDE. Created modular code for reading sensor data, controlling output, and displaying on LCD.
5. **Threshold Calibration:** Conducted field simulation to determine suitable moisture and pH thresholds based on sample data.
6. **Power Integration:** Designed a power system capable of sustaining the ESP32, pump, and LCD using 18650 Li-ion batteries with voltage regulation.

7. **Testing:** Dry and wet soil samples were tested for repeatability. Sensor responses were validated against known benchmarks.
8. **Iteration:** Improved pump control logic and LCD readability after initial test results. Final prototype was made compact and portable for real field use.

.

5. Project Execution

5.1 Planning and Design:

The planning phase began with extensive literature review and brainstorming sessions where the problem statement, expected outcomes, and user requirements were clearly outlined. During this phase, the team identified the core functionalities: soil moisture and pH sensing, real-time display, automated irrigation, and portable power. A detailed Gantt chart was prepared to define the weekly deliverables and responsibilities.

Initial design drafts were created using hand-drawn schematics, and later refined using digital circuit design tools. Each component was selected based on cost, availability, and compatibility with the ESP32. The use of modular blocks (sensor module, power module, display module, actuator module) was finalized to allow ease of expansion in future iterations.

5.2 Implementation:

Implementation started with sensor interfacing on a breadboard. The team first calibrated the soil moisture sensor using wet and dry soil samples, adjusting for analog output variations. The pH sensor required additional filtering in code to stabilize the readings.

The LCD was connected using an I2C interface for fewer GPIO pin requirements. Once sensor readings were displayed successfully, the relay was introduced to switch the water pump based on soil moisture levels.

The battery management system was carefully designed using the TP4056 charging module and MT3608 booster, which enabled the circuit to be run off 3.7V batteries and regulate to 5V as needed.

In parallel, code development progressed in the Arduino IDE. The firmware was developed modularly: one function each for sensor reading, display updates, decision logic, and motor control.

Testing was conducted in phases. Initial lab-based testing confirmed sensor accuracy and system stability. Field testing was simulated using garden soil. The system reliably triggered the pump when soil was dry and turned it off once the threshold was met. The LCD effectively displayed live sensor readings during the test conditions.

Post-testing, the components were soldered onto a PCB to increase durability, and the entire unit was enclosed in a plastic casing for protection. All modules were labelled, and a final checklist of functionality was cross-verified with the project requirements.

This phase concluded with a fully functional, portable prototype capable of monitoring and responding to soil conditions autonomously.

6. Tools and Techniques Used

6.1 Tools:

i. **Arduino IDE:** Primary environment for coding and uploading firmware to the ESP32.

ii. **ESP32 Dev Module:** Chosen for its performance, multiple GPIO pins, Wi-Fi capabilities, and low-power consumption.

iii. **Soil Moisture Sensor (Capacitive):** Measures water content in soil and provides analog output.

iv. **pH Sensor Module:** Analog pH probe interfaced to monitor soil acidity or alkalinity.

v. **LiquidCrystal_I2C Library + LCD (16x2):** Simplifies real-time feedback display using minimal pins.

vi. **TP4056 Charging Module:** Used for safe USB-based battery charging with protection features.

vii. **7805 Voltage Regulator:** Maintains stable 5V output from a higher-voltage source.

viii. **MT3608 Boost Converter:** Boosts battery output to ensure consistent voltage to ESP32 and peripherals.

ix. **Relay Module:** Controls high-current devices like the water pump via ESP32 logic signals.

x. **Buzzer and LEDs:** Used for status indication and alerts.

xi. **Wi-Fi Router and Blynk Cloud:** Facilitated remote monitoring through a smartphone dashboard.

6.2 Techniques:

i. **Sensor Mapping and Calibration:** Used analogRead() to fetch raw values and map () + constrain () functions to convert to readable formats (e.g., 0–100% moisture).

ii. **Relay Switching Logic:** Relay was triggered based on moisture threshold; code ensured safety and toggling logic.

iii. **LCD Display Handling:** I2C LCD updates provided structured user feedback in two lines, efficiently showing sensor values and pump status.

iv. **Wi-Fi + Blynk Integration:** ESP32 was configured to connect via Wi-Fi and send real-time data to Blynk virtual pins (V0 for moisture, V1 for pH, V2 for pump).

v. **Modular Code Structure:** Code broken into sections in setup() and loop() for clarity; includes sensor reading, logic decision, display update, and Blynk communication.

vi. **Power Optimization:** Low-power components were selected and logic was implemented to avoid continuous pump activation.

vii. **Real-time Feedback System:** Combines LCD, Blynk app, buzzer, and LEDs to notify users of critical states in the soil environment.

7. Results and Discussion

7.1 Final Results:

The final prototype successfully integrated all key hardware components and software functionalities. The ESP32 microcontroller effectively acquired real-time data from the soil moisture and pH sensors, processed it, and displayed it clearly on the 16x2 LCD screen using I2C communication. The pump activation through the relay was accurate and consistent, functioning as intended when soil moisture dropped below the predefined threshold. LED indicators provided intuitive visual status signals—green for pump ON and red for pump OFF—while the buzzer served as an audible alert during low-moisture events.

The system also demonstrated reliable remote communication with the Blynk cloud platform. Through a Wi-Fi connection, sensor data was transmitted and visualized in the Blynk mobile app in real time. This enabled wireless monitoring of soil parameters from anywhere with internet access.

Power supply management was stable throughout testing. The 18650 batteries, charged via a TP4056 module and boosted through an MT3608 converter, provided consistent voltage output for extended periods, validating the system's suitability for off-grid or rural applications. Sensor calibration efforts resulted in accurate readings, with soil moisture values reflecting known wet/dry conditions and pH readings aligning with reference testing kits.

7.2 Discussion:

This project met all its primary objectives and demonstrated the feasibility of implementing a real-time soil health monitoring and irrigation control system using affordable, accessible components. The results validate the system's ability to automate irrigation based on live soil data, thereby reducing manual intervention and promoting water conservation.

The LCD displayed real-time feedback effectively, and the use of visual and audible indicators helped improve system usability, especially in outdoor or low-light environments. The pH and moisture readings were consistent over multiple tests, confirming the reliability of the sensors and calibration techniques. The logic for pump activation based on moisture thresholds worked flawlessly during testing, and the inclusion of a relay provided safe and effective switching of the water pump.

One of the most impactful outcomes was the system’s potential for scalability and future integration. The Wi-Fi and Blynk integration pave the way for IoT expansion, allowing cloud storage of sensor data and integration with farm management platforms. Field deployment tests are expected to offer more data for performance tuning across diverse soil types and environmental conditions.

Moreover, the modular hardware design ensures easy upgrades. Future iterations can include temperature/humidity sensors, solar charging modules, or additional soil health parameters such as NPK detection. The system architecture supports these additions without significant redesign.

In conclusion, the system is robust, portable, cost-effective, and applicable to a variety of small-scale agriculture use cases, especially in rural or remote regions where traditional irrigation monitoring is lacking.

8. Prototype (Hardware/Software)

8.1 Prototype Description:

The developed prototype is a compact and modular system designed for portability and real-world soil monitoring. It includes:

i. **ESP32 Dev Module:** Acts as the central processing unit. Reads analog inputs from sensors, executes logic for pump activation, and manages Blynk communication over Wi-Fi.

ii. **Soil Moisture Sensor (Capacitive):** Continuously senses soil moisture content. Connected to analog input pin VP (GPIO34).

iii. **pH Sensor Module:** Measures soil acidity or alkalinity, connected to analog pin VN (GPIO35). Values are calibrated and interpreted using a mapping formula.

iv. **Relay Module:** Connected to GPIO27, it switches the water pump ON/OFF based on soil moisture.

v. **Water Pump:** A mini 5V motor pump that activates during low moisture conditions.

vi. **LCD Display (16x2 with I2C):** Shows real-time moisture percentage, pH value, and pump status. Connected via I2C protocol to reduce pin usage.

vii. **Power Supply:** Two 18650 Li-ion batteries managed by a TP4056 charging module and MT3608 booster. This enables rechargeable, portable use with regulated 5V output.

viii. **Buzzer (GPIO26):** Gives audible alert when the soil is dry.

ix. **LED Indicators:** Red (GPIO33) and Green (GPIO25) LEDs indicate the pump status clearly for quick visual feedback

8.2 Development Process:

The prototype was developed through an incremental process:

i. **Initial Assembly:** All components were first connected on a breadboard. The ESP32 was programmed to read sensor values and verify signal stability.

ii. **Sensor Testing:** Soil moisture was tested using various samples—dry, semi-wet, and wet. Calibration was done to convert raw analog readings into accurate moisture percentages.

iii. **pH Sensor Adjustment:** pH readings were compared with commercial pH test kits. Raw analog values were mapped to pH scale (0–14) using a custom formula.

iv. **LCD Display Configuration:** Using the LiquidCrystal_I2C library, the display was initialized to show key information with readability for outdoor conditions.

v. **Relay Integration:** Relay and pump were added next. Controlled by digital logic from ESP32, the pump turned ON only when soil moisture dropped below 30%.

vi. **Wi-Fi + Blynk Integration:** Wi-Fi credentials and Blynk Auth Token were configured. Real-time sensor values were pushed to Blynk virtual pins (V0: Moisture, V1: pH, V2: Pump).

vii. **Power Module Design:** TP4056 ensured safe charging of batteries. MT3608 boosted the 3.7V battery output to 5V needed by the ESP32 and sensors.

viii. **Soldering and PCB Mounting:** After successful breadboard testing, components were transferred onto a perfboard. Wires were carefully routed and soldered to minimize short circuits.

ix. **Enclosure:** The final circuit was housed inside a plastic casing for protection during field testing. Cut-outs were made for the LCD and USB charging port.

8.3 Testing and Validation:

Comprehensive testing was carried out in various conditions:

i. **Moisture Test:** Dry and wet soils were used to trigger and deactivate the pump. The system consistently activated below 30% moisture.

ii. **pH Test:** Solutions of known pH were tested. The displayed values closely matched manual pH paper tests, confirming sensor accuracy.

iii. **Field Simulation:** Prototype was placed in a potted plant. Over several hours, it was observed to activate and deactivate the pump autonomously based on soil dryness.

iv. **Power Test:** The battery module powered the system for over 6 hours continuously. No voltage drops were observed that would affect ESP32 performance.

v. **LCD and Alerts:** Display remained readable in daylight. Buzzer provided clear alerts, and LED indicators gave quick visual feedback.

vi. **Remote Monitoring:** Blynk dashboard accurately reflected real-time changes in soil condition. Users were able to monitor pump status from a mobile device.

This rigorous process resulted in a stable, functional prototype tailored for agricultural field use, meeting all design goals and ready for deployment.

9. Conclusion

9.1 Summary:

This project effectively addressed the problem of inefficient irrigation and soil health monitoring by developing a functional, real-time soil detector system. The integration of an ESP32 microcontroller with analog soil moisture and pH sensors enabled automated control of a water pump using relay logic, thereby minimizing human intervention. The use of a 16x2 LCD for real-time data display, LED indicators for quick status identification, and a buzzer for alerts added practical usability for field deployment.

The system was also integrated with the Blynk IoT platform, enabling remote monitoring and real-time data visualization via a smartphone. This advancement marks a significant step toward smart agriculture. The power system, designed around rechargeable 18650 batteries with a TP4056 charger and MT3608 booster, confirmed the potential for use in rural or off-grid environments.

From initial planning to final testing, the project progressed through structured design, modular implementation, and rigorous validation. Calibration ensured sensor accuracy, while the compact PCB layout and protective casing made the prototype robust and field-ready.

This system can serve as a valuable tool for farmers, agritech researchers, and educational institutions aiming to implement IoT solutions in agriculture. Its modular design allows for scalability—future upgrades may include wireless communication protocols like LoRa, addition of temperature/humidity sensors, cloud storage, or machine learning-based predictive analytics for irrigation scheduling.

The project not only achieved its core objectives but also demonstrated a working solution that is low-cost, efficient, easy to use, and adaptable to real-world needs. It contributes to the broader goal of sustainable agriculture and water conservation through data-driven farming.

9.2 Personal Reflection:

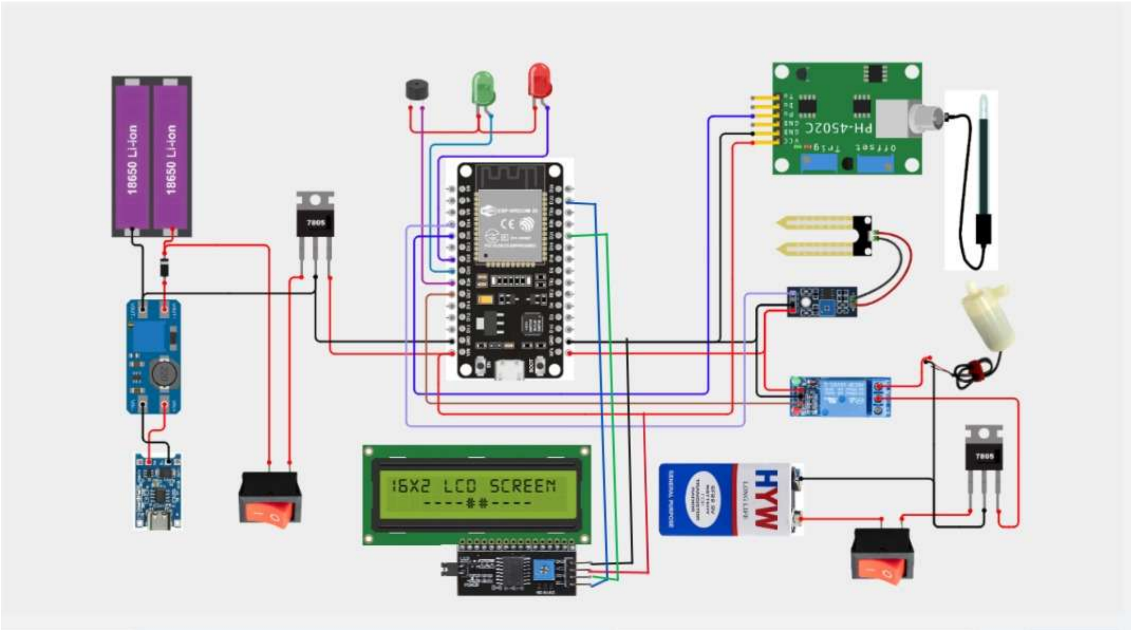
AAYUSHI PRIYA: Gained deep insight into IoT hardware and power management.

AINESH MALLICK: Learned integration and calibration of analog sensors.
RISHABH GUPTA: Developed skills in prototyping and debugging embedded systems.

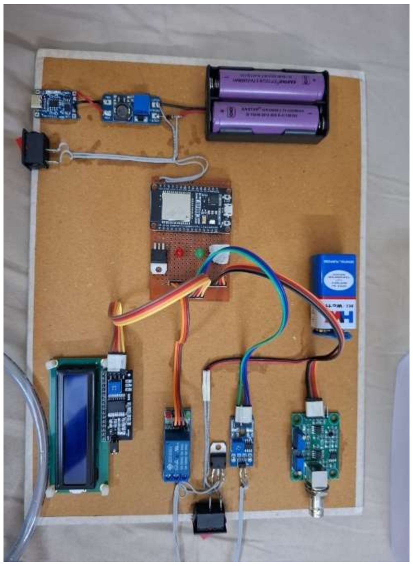
RISHIT KUMAR: Improved understanding of sustainable tech in agriculture.

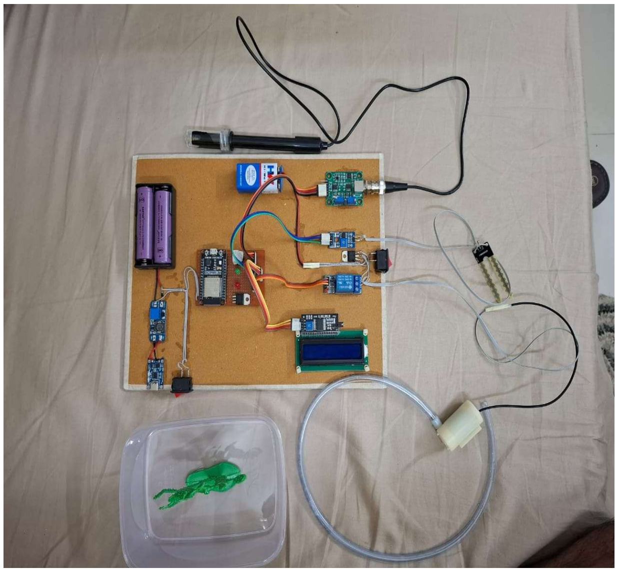
Visuals:

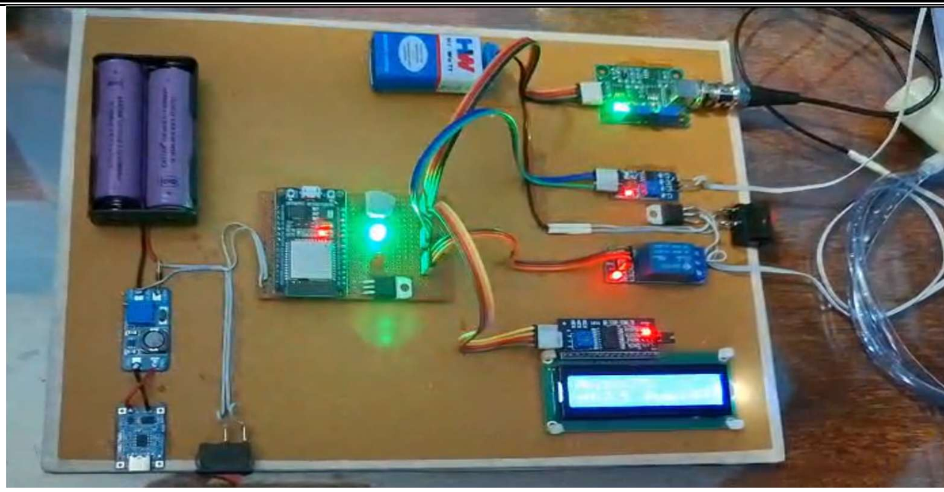
CIRCUIT DIAGRAM:



PICTURES OF PROJECT:







SOURCE CODE:

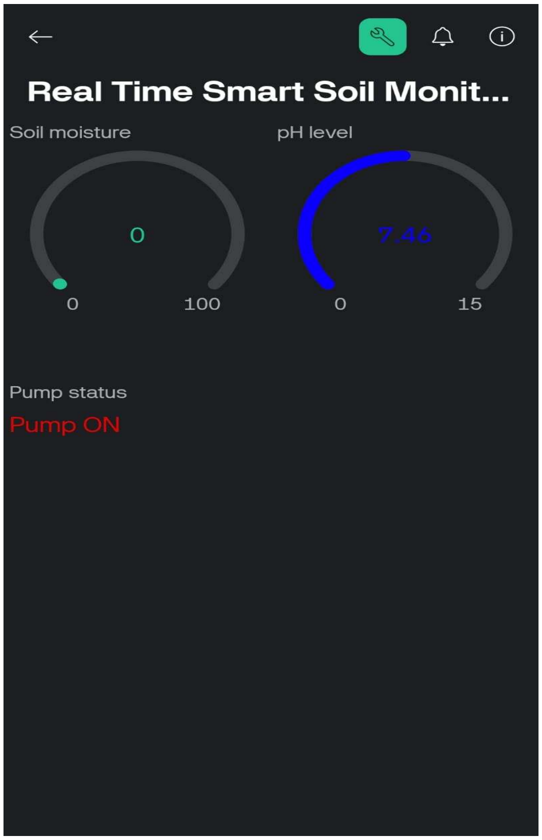
Agriculture_with_esp32_pH_sensor_moisture_sensor_blynk.ino

```
#define BLYNK_TEMPLATE_ID          "..."
#define BLYNK_TEMPLATE_NAME        "..."
#define BLYNK_AUTH_TOKEN           "..."
#include <WiFi.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <BlynkSimpleEsp32.h>

// WiFi credentials
char ssid[] = "Aayushi Oppo";
char pass[] = "anayanusha";

// Blynk Auth Token
char auth[] = "qRfJZKOWlBpCwvcjOYjcs06aV7-XwGdg";

// Pins
#define SOIL_PIN 34   // VP
#define PH_PIN 35     // VN
#define RELAY_PIN 27
#define BUZZER_PIN 26
#define GREEN_LED 25
#define RED_LED 33
```

```
// LCD
LiquidCrystal_I2C lcd(0x27, 16, 2);

// Variables
int soilValue;
float moisturePercent;
int phRaw;
float phValue;
bool pumpStatus = false;

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

  pinMode(RELAY_PIN, OUTPUT);
  pinMode(BUZZER_PIN, OUTPUT);
  pinMode(GREEN_LED, OUTPUT);
  pinMode(RED_LED, OUTPUT);

  digitalWrite(RELAY_PIN, LOW);
  digitalWrite(BUZZER_PIN, LOW);
  digitalWrite(GREEN_LED, LOW);
  digitalWrite(RED_LED, HIGH);

  lcd.init();
  lcd.backlight();

  WiFi.begin(ssid, pass);
  Blynk.begin(auth, ssid, pass);

  lcd.setCursor(0, 0);
  lcd.print("WELCOME TO");
  lcd.setCursor(0, 1);
  lcd.print("AGRI PROJECT");
  delay(3000);
  lcd.clear();
}

void loop() {
  Blynk.run();

  // Read sensors
  soilValue = analogRead(SOIL_PIN);
  moisturePercent = map(soilValue, 4095, 0, 0, 100);
  moisturePercent = constrain(moisturePercent, 0, 100);

  phRaw = analogRead(PH_PIN);
  phValue = map(phRaw, 0, 6955, 0, 140) / 11.0;  // Adjust if needed
```

```
  // Check and control pump
  if (moisturePercent < 30) {
    digitalWrite(RELAY_PIN, HIGH);  // Pump ON
    digitalWrite(BUZZER_PIN, HIGH);
    digitalWrite(GREEN_LED, HIGH);
    digitalWrite(RED_LED, LOW);
    pumpStatus = true;
    delay(500);
    digitalWrite(BUZZER_PIN, LOW);
  } else {
    digitalWrite(RELAY_PIN, LOW);   // Pump OFF
    digitalWrite(GREEN_LED, LOW);
    digitalWrite(RED_LED, HIGH);
    pumpStatus = false;
  }

  // Display on LCD
  lcd.setCursor(0, 0);
  lcd.print("Moist:");
  lcd.print((int)moisturePercent);
  lcd.print("% ");

  lcd.setCursor(0, 1);
  lcd.print("pH:");
  lcd.print(phValue, 1);
  lcd.print(" ");

  lcd.setCursor(8, 1);
  if (pumpStatus)
    lcd.print("Pump:ON ");
  else
    lcd.print("Pump:OFF");

  // Send to Blynk
  Blynk.virtualWrite(V0, (int)moisturePercent);
  Blynk.virtualWrite(V1, phValue);
  Blynk.virtualWrite(V2, pumpStatus ? "Pump ON" : "Pump OFF");

  delay(500);
}
```

REAL TIME APP DISPLAY



11. QR Code of Demonstration Video