UNIVERSITI MALAYSIA TERENGGANU

FACULTY OF COMPUTER SCIENCE AND MATHEMATICS

CSM3313 INTERNET OF THINGS

GROUP 5

WATERING PLANT SENSOR

PREPARED BY:

PUTRI BALQIS BT HANAFI (S65630)

AERISHA FASHAHIRA BT MOHD PUAD (S66081)

HAZRIEN NUR QISTINA BT HASWADI (S67017)

PREPARED FOR:

DR FARIZAH YUNUS

BACHELOR OF COMPUTER SCIENCE WITH MARITIME INFORMATICS

SEMESTER I 2024/2025

EXECUTIVE SUMMARY 3

1. INTRODUCTION........ 4

1.1 BACKGROUND 4

1.2 PROBLEM STATEMENT 5

1.3 PROBLEM SOLUTIONS........ 6

1.4 OBJECTIVES 7

1.5 SCOPE 7

2.LITERATURE REVIEW........ 8

2.1 OVERVIEW OF IOT 8

2.2 RELATED WORK 8

2.3 GAP........ 9

3. SYSTEM DESIGN 10

3.1 SYSTEM ARCHITECTURE........ 10

3.2 HARDWARE COMPONENT 11

3.3 SOFTWARE COMPONENT........ 12

4.IMPLEMENTATION 13

4.1 SETUP AND CONFIGURATION 13

4.2 INTEGRATION OF COMPONENTS 15

4.3 DATA FLOW (FLOW CHART) 16

5.RESULTS AND ANALYSIS........ 18

5.1 DATA COLLECTED........ 18

6.CHALLENGES AND LIMITATIONS 20

6.1 TECHNICAL CHALLENGES 20

6.2 CONSTRAINT FACED........ 21

7.FUTURE WORK........ 22

7.1 POTENTIAL ENHANCEMENT........ 22

8.CONCLUSION 23

8.1 SUMMARY OF FINDINGS 23

8.2 FIND REMARKS........ 23

9. REFERENCES 24

10.APPENDIXES........ 25

EXECUTIVE SUMMARY

Watering Plant Sensor

This document outlines the use of Internet of Thing (IoT) in agriculture for watering plants. The sensor that uses in this project are detected soil moisture, temperature and humidity, and weather forecast will help to solve a lot of problems and improve the garden watering plant. A project focused on agriculture used soil moisture sensors and water pumps, with software UpyCraft running the coding and FavorIoT website receiving outputs. The project also integrated with the OpenWeatherMap API for temperature, humidity, and weather data. It will reduce over watering for plants during raining time. This project demonstrates the way for people to use IoT technology in their daily life for future improvement in gardening.

1. INTRODUCTION

1.1 BACKGROUND

For some individuals who are looking for a work-life balance, home gardening is a hobby that allows them to be a pastime to beautify their yard and have a pleasant evening after long hours of working. However, maintaining the home garden to be always in good condition such as watering it frequently can be exhausting especially during hot weather. Home garden owners face difficulties when they are away from their residences because of work, travel, or other reasons. This absence can leave their plants without access to any water resource for survival.

Malaysia's tropical climate poses a risk of moisture loss, causing plant wilting and death. This can lead to dry seasons, affecting plants health. Some plants can withstand drought, but flowers and vegetables like roses and orchids need daily watering. Traditional gardening methods rely on manual labour, which is insufficient for home gardeners who are often away. Automated watering solutions are needed to address plant loss, reduced goods, and increased worry, highlighting the need for improved home gardening experiences.

The emerging topic that includes of technical, computer is the Internet of Things (IoT). According to Mouha (2021), many people used products such as cars, industrial components and facilities, sensor and other objects with internet connectivity and capabilities of powerful data analysis to transform the way we live in this world. IoT is the any device that have ability to collect data embedded with any types of sensors that enables interaction between Machine-to-Machine (M2M) for communication.

For the project that conducted under the theme agriculture, watering plant sensor were chosen as title. The types of sensors that used are soil moisture sensor, and water pump using relay module, breadboard, jumper wires and power source. Software UpyCraft were used to run the coding when connected with the Esp32. FavorIoT website were connected to receive the output that were executed by the sensors. It is also integrated to the OpenWeatherMap API to get the temperature, humidity and weather description to record the data.

1.2 PROBLEM STATEMENT

This section outlines the identified problems about watering plant that happen in daily life.

1. **Insufficient knowledge about watering needs:**

Some people enjoy gardening but lack sufficient knowledge, particularly about the appropriate amount of water needed for their plants. This often leads to the withering of plants.

2. **Overwatering plant:**

The plant died because of overwatering by the individuals. Overwatering is the common mistakes that happens to the gardener. It happens because of lack of gardening knowledge and forgetting the watering schedule.

3. **People always forget to watering the plant:**

The people always forget to watering the plant. Many gardeners struggle to maintain a consistent schedule, leading to neglected plants and poor health.

These problem statements highlight the need for an application or smart system to help gardeners maintain healthier gardens and improve their gardening experience.

1.3 PROBLEM SOLUTIONS

Presented in this section is the solution that has been designed to address the identified problems and achieve the project objectives.

1. **Utilization of Soil Moisture Sensors:**

To address the issue of insufficient knowledge about watering needs, the system incorporates soil moisture sensors. To ensure sufficient hydration, these sensors will determine the soil's moisture levels and the optimal amount of water needed for plants.

2. **Integration with Weather Forecast APIs:**

To minimize overwatering, the system uses the OpenWeatherMap API to receive real-time weather forecasts so it can reduce water waste and ensuring the plants are not overwatered if rain is expected.

3. **Automated Watering Activation:**

To assist gardeners in maintaining a regular watering schedule, this technology automates the whole process. The water pump will be switched on when the soil moisture content falls below a certain threshold to effectively hydrate the plants. This maintains the best possible growth of plants without requiring manual intervention.

1.4 OBJECTIVES

The primary goal of this project is to create an efficient and automated plant watering system. The specific objectives are as follows:

i. To automate the plant watering process using soil moisture sensors to ensure plants receive adequate hydration.

ii. To integrate real-time weather data to adjust watering schedules, preventing overwatering during periods of expected rainfall.

iii. To encourage sustainable water usage by optimizing irrigation practices.

1.5 SCOPE

The scopes for this IOT project are as follows:

i. The water plant sensor aims to provide efficient solution for monitoring and managing soil moisture to ensure plants receive adequate hydration.

ii. It integrates real-time weather data to prevent overwatering and promotes sustainable water usage through optimized irrigation practices.

iii. The system is user-friendly and designed to meet the needs of plant owners while supporting efficient resource management.

2.LITERATURE REVIEW

2.1 OVERVIEW OF IOT

Aerisha's mother enjoys spending time outside and frequently goes on outings, often leaving the house for extended periods. During these absences, the numerous flower plants around the house are left unattended, resulting in dehydration due to the lack of proper watering. This lack of care leads to the plants wilting or dying, especially in her mother's absence.

The situation highlights a significant challenge in ensuring the well-being of the plants when no one is around to tend to them. To address this issue, a reliable and automated system for watering the plants is necessary, ensuring that they receive the proper care and attention even when no one is home. Implementing such a system would help maintain the health of the plants and prevent them from suffering due to dehydration.

2.2 RELATED WORK

Several studies have explored IoT-based automatic plant watering systems:

i. **IoT-Based Automatic Plant Watering System with the Blynk Application**:

This study developed a system using Arduino Uno, ESP8266, DHT11, and soil moisture sensors, controlled via the Blynk application. The system automated plant watering based on real-time soil moisture data, enhancing irrigation efficiency.

ii. **An Experimental Comparison of IoT-Based and Traditional Irrigation Systems**:

This research compared water usage between IoT-based and traditional irrigation methods over nine months on a 2-acre lemon farm. The IoT system utilized real-time sensor data and a decision support system, resulting in optimized water utilization.

iii. **IoT-Based Plant Watering System**:

This paper presented a system equipped with soil moisture, temperature, and humidity sensors to monitor environmental conditions. When soil moisture fell below a preset threshold, the system activated a water pump, ensuring precise watering.

2.3 GAP

Gap analysis was conducted to compare the proposed IoT-based plant watering system with existing similar systems. The comparison highlights several advantages of our system over current solutions:

1. Integration with FavorIot Platform

Existing Systems: Many existing IoT-based plant watering systems rely on local storage or standalone microcontrollers, limiting data accessibility and remote monitoring capabilities.

Proposed System: Our system leverages the FavorIot platform, a cloud-based IoT platform, to store, process, and manage sensor data in real time. This integration enables users to monitor soil moisture levels and control the watering system remotely from any location.

2. Cost Efficiency

Existing Systems: Some systems use high-end microcontrollers and sensors, which increase the overall cost, making them less accessible for small-scale farmers or home gardeners.

Proposed System: By using the cost-effective ESP32 microcontroller and essential components, our system achieves the same functionality while reducing costs, making it more affordable and accessible for individual users.

3. SYSTEM DESIGN

3.1 SYSTEM ARCHITECTURE

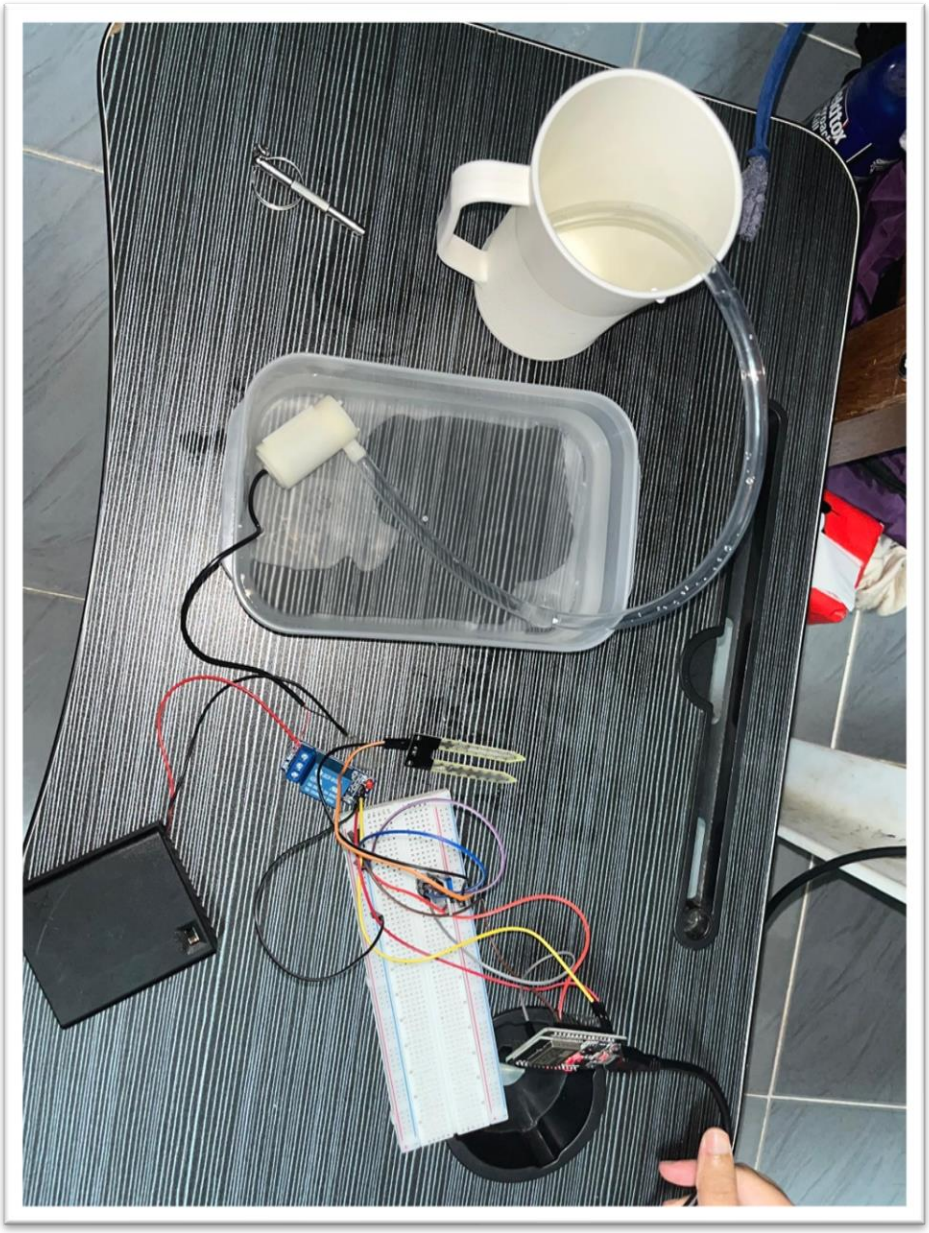


Figure 3.1

3.2 HARDWARE COMPONENTS

No	Hardware Component	Functions
1.	Soil Moisture Detector	i. Measures the moisture level in the soil. ii. Sends data to the controller (ESP32) to determine if watering is needed.
2.	Relay Module Control	i. Acts like a switch
3.	Water Pump	i. Pump water when activated by the relay module.
4.	Power Source	i. supplies electrical power
5.	Jumper wires	i. Creates electrical connections between the components
6.	0.5m Vinyl Tubing	i. Directs the flow of water from the pump to the plants
7.	USB power cable	i. Powers the ESP32 or other connected components via a USB power source.
8.	ESP32	i. Serves as the central controller

Table 3.2 Hardware component of the project.

3.3 SOFTWARE COMPONENT

No	Software Component	Functions
1.	UpyCraft	i. A Python-based IDE (Integrated Development Environment) used for ESP32 programming. ii. Run the coding when connected with the ESP32
2.	FavorIot	i. A cloud-based IoT platform for data management. ii. Receives and stores sensor data (e.g., soil moisture, temperature, weather).
3.	Weather API Integration	i. Provide real-time weather data for the system's location.

Table 3.2 Software component of the project.

4. IMPLEMENTATION

4.1 SETUP AND CONFIGURATION

This section describes the preparation and system setup, including hardware components, connections and the initialization of software settings.

1. **Soil Moisture Sensor Setup**

   The plant's soil has a soil moisture sensor inserted to measure moisture levels.
   It has two pins, digital output (DO) and VCC (Voltage Common Collector), which connect to the ESP32 microcontroller for signal transmission and power supply.

2. **ESP32 Configuration**

   The ESP32 microcontroller is programmed to process input from the soil moisture sensor. It acts as the central control unit, managing the relay module and facilitating optional remote monitoring through a Wi-Fi connection.

3. **Weather API Integration**

   A weather API is utilized to retrieve real-time weather data. This information is processed by the ESP32 to optimize watering schedules based on external environmental conditions.

4. **UpyCraft for ESP32 Programming**

   The UpyCraft IDE is used to code and upload the ESP32 firmware. It allows for efficient implementation of control logic and integration of sensor data processing with the weather API.

5. **FavorIOT Platform**

   The FavorIOT platform is utilized to receive and monitor sensor data. It can remote access to real-time and historical information about soil moisture levels and system operations, providing enhanced control and analytics.

6. **Relay Module Setup**

   The relay module acts like a switch to control the water pump and connected to the ESP32. When triggered by the soil moisture sensor and its input pin receives signals from ESP32, it allows power to flow to the water pump.

7. **Power Source**

   This completes the system, which is powered with an external battery or a power supply unit connected to the ESP32, relay module. Connections are made via VCC and GND wires.

4.2 INTEGRATION OF COMPONENTS

The following describes how each component is interconnected:

1. **Soil Moisture Sensor Integration:**

   The soil moisture sensor is inserted into the soil to monitor moisture levels. Its digital output (DO) pin connects to the ESP32 microcontroller. The sensor sends moisture data to the ESP32 for further processing.

2. **Signal Processing via ESP32:**

   The ESP32 serves as the central controller analysed data received from the soil moisture sensor. When the soil moisture level falls below the preset threshold, the ESP32 triggers the relay module by sending a digital signal to its input pin.

3. **Relay Module Control:**

   The relay module acts as an electronic switch. Upon receiving a signal from the ESP32, it closes the circuit, allowing current to flow to the water pump.

4. **Water Pump Activation:**

   The water pump receives power through the relay module and begins pumping water. Vinyl tubing directs the water from the pump to the soil to ensure targeted irrigation of plants.

5. **Automatic Shutdown:** The soil moisture sensor detects adequate hydration level, it stops sending a signal to the ESP32. Then, the ESP32 deactivates the relay module and cuts power to the water pump, stopping water flow.

6. **Power Source Integration:**
   The system is powered by an external power source, such as a battery or USB connection, which supplies electricity to the components.

4.3 DATA FLOW

The diagram below shows the watering plant sensor data flow.

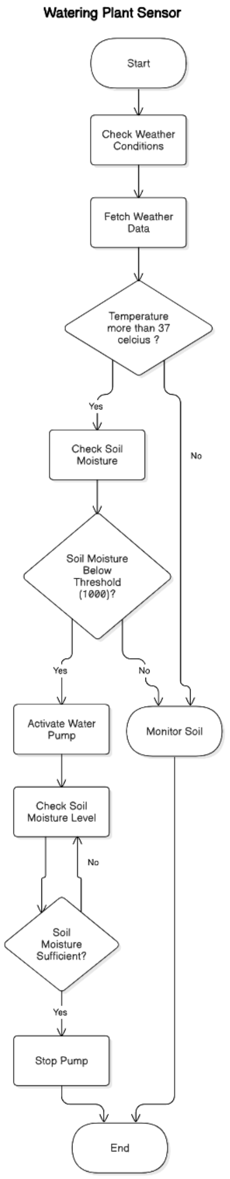


Figure 4.3 Data flow diagram.

Explanation for data flow diagram:

1. Start
2. Check Weather Conditions
   - Use the Weather API to fetch real-time weather data (example, rainfall, humidity).
   - If temperature higher than 37 Celsius, proceed to the next step.
   - If temperature below than 37 Celsius, monitor the soil and end.
3. Check Soil Moisture using sensor
   - If the soil moisture level is below the threshold (1000), activate the water pump.
   - If the soil moisture level is above the threshold (1000), monitor the soil and end.
4. Activate water pump.
5. Check the soil moisture level

   If soil moisture sufficient, stop pump, if no, check soil moisture level.
6. Stop pump when the soil moisture is sufficient.
7. End.

5.RESULTS AND ANALYSIS

5.1 DATA COLLECTED

Data collected from watering plant sensor project are sent to Favoriot website for monitoring and analysis. The system uses this data to activate the pump only when both the soil is dry and the temperature exceeds a specified threshold, ensuring efficient water usage:

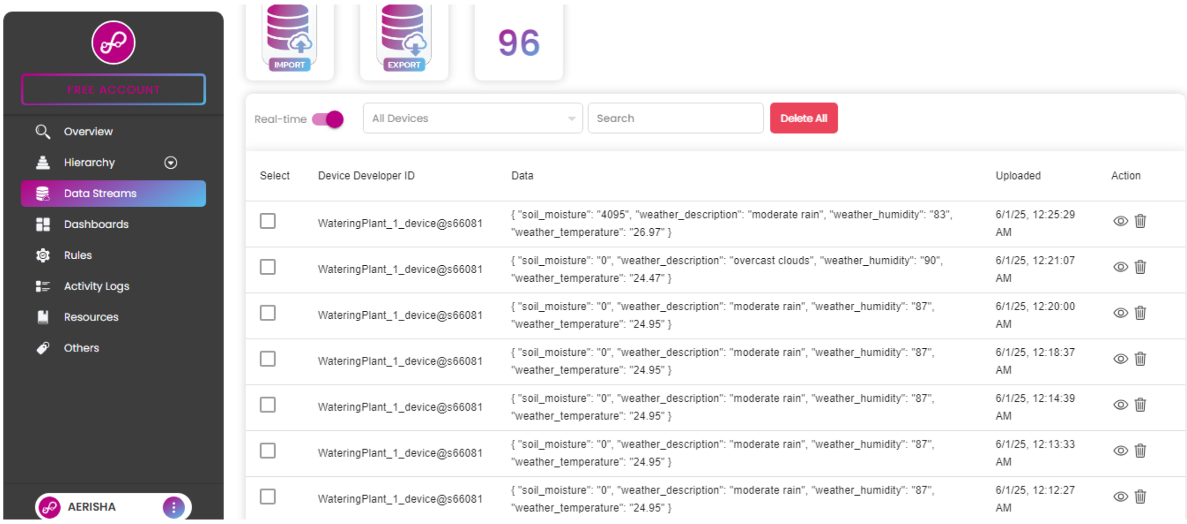
1.Soil Moisture Value: The value is collected by using the soil moisture sensor to indicate how dry or wet the soil is. It is can help to determine whether the plants need watering or not.

2.Temperature Information: The project retrieved the weather temperature from OpenWeatherMap API. The temperature value is in Degree Celsius unit.

3.Humidity: Humidity value also retrieved from OpenWeatherMap API.

4.Weather Description: The description will display like “moderate rain”, “light rain”, “overcast cloud”, “clear sky”. It is useful for understanding the weather context in word.

Analyzing the data show a valuable insight for plant care. The soil moisture value will be compared to moisture threshold and temperature to activate the water pump. If the moisture value below than 1000 and temperature higher than 37 Celsius, the relay module will execute message to turn on the water pump. Soil moisture values help identify periods of dryness and the frequency of watering needs for the plants. After one minute, another data sent to Favoriot to show the soil moisture value has increase.



Select	Device Developer ID	Data	Uploaded	Action
□	WateringPlant_1_device@s66081	{ "soil_moisture": "4095", "weather_description": "moderate rain", "weather_humidity": "83", "weather_temperature": "26.97" }	6/1/25, 12:25:29 AM	
□	WateringPlant_1_device@s66081	{ "soil_moisture": "0", "weather_description": "overcast clouds", "weather_humidity": "90", "weather_temperature": "24.47" }	6/1/25, 12:21:07 AM	
□	WateringPlant_1_device@s66081	{ "soil_moisture": "0", "weather_description": "moderate rain", "weather_humidity": "87", "weather_temperature": "24.95" }	6/1/25, 12:20:00 AM	
□	WateringPlant_1_device@s66081	{ "soil_moisture": "0", "weather_description": "moderate rain", "weather_humidity": "87", "weather_temperature": "24.95" }	6/1/25, 12:18:37 AM	
□	WateringPlant_1_device@s66081	{ "soil_moisture": "0", "weather_description": "moderate rain", "weather_humidity": "87", "weather_temperature": "24.95" }	6/1/25, 12:14:39 AM	
□	WateringPlant_1_device@s66081	{ "soil_moisture": "0", "weather_description": "moderate rain", "weather_humidity": "87", "weather_temperature": "24.95" }	6/1/25, 12:13:33 AM	
□	WateringPlant_1_device@s66081	{ "soil_moisture": "0", "weather_description": "moderate rain", "weather_humidity": "87", "weather_temperature": "24.95" }	6/1/25, 12:12:27 AM	

Figure5.1: Data in Favoriot

Picture above show the data that enter to the Favoriot platform. All data that sent can be viewed in real time or store for analysis. The soil moisture value was records to know if the plants already receive water or not. Weather description, humidity and temperature were display to ensure the water pump start and stop on the right time. At 12:18 A.M. until 12.21 A.M. data shown soil moisture value is zero (0).

The water pump release water to the plant and at 12:25 A.M the data recorded show moisture value is four thousand ninety-five (4095). It means the soil moisture are working properly and soil has received water. This allows user remote monitoring of plant condition when they are away from home.

6.CHALLENGES AND LIMITATIONS

6.1 TECHNICAL CHALLENGES

i. **Sensor Calibration:**

Soil moisture sensors may require frequent calibration to maintain accuracy.
Variations in soil type and composition can impact sensor readings.

ii. **Connectivity Issues:**

Stable Wi-Fi connectivity is crucial for real-time data transmission to Internet of Things (IoT) platforms like FavorIoT. However, poor network conditions can disrupt data transfer and impact the overall performance of the system.

iii. **Data Accuracy:**

Environmental factors, such as temperature fluctuations and sensor wear, pose challenges to maintaining the accuracy of data readings. These influences can lead to errors in measurements, potentially compromising system reliability.

6.2 CONSTRAINT FACED

i. **Hardware Limitations**:
Hardware limitations presented challenges to the system's performance. The ESP32 microcontroller, while cost-effective and versatile, is restricted by its limited memory and processing power.

ii. **Environmental Factors**:
Environmental factors posed a significant risk to the durability and functionality of the system. Components such as soil moisture sensors and water pumps are susceptible to damage under harsh weather conditions, including heavy rainfall. Such environmental stresses can reduce the lifespan and reliability of the hardware, necessitating frequent maintenance or replacement.

7.FUTURE WORK

7.1 POTENTIAL ENHANCEMENT

i. **Solar Power Integration**:
To enhance sustainability and enable operation in remote areas, it is recommended to integrate solar panels as an alternative power source. Solar energy is a renewable resource that reduces dependency on conventional electricity while ensuring uninterrupted functionality in off-grid locations.

ii. **Multiple Sensor Support**:
Expanding the system to include additional sensors, such as pH sensors and nutrient detectors, will provide more comprehensive monitoring capabilities. This improvement will enable the system to collect critical data for precise environmental management, contributing to better decision-making and resource optimization.

iii. **Machine Learning Integration**:
Integrating machine learning algorithms into the system can facilitate the analysis of historical data to optimize watering schedules. This approach enhances efficiency by adapting irrigation patterns to the specific needs of the environment, thereby reducing water wastage and improving overall productivity.

8.CONCLUSION

8.1 SUMMARY OF FINDINGS

The Internet of Things (IoT)-based plant watering system has proven to be a transformative solution for automating irrigation processes. By leveraging sensors to monitor soil moisture and environmental conditions, the system ensures precise water delivery tailored to plant needs. This automation significantly enhances resource efficiency, reduces the need for manual labour, and fosters optimal conditions for plant growth.

The integration of the FavorIoT platform further amplifies the system's capabilities by providing real-time data management and monitoring functionalities. This integration has enabled a robust and reliable framework for smart agriculture applications, showcasing its potential for improving agricultural productivity and sustainability.

8.2 FIND REMARKS

To overcome these challenges, future developments should prioritize the integration of renewable energy sources like solar power to ensure uninterrupted operation in remote areas. Additionally, the incorporation of advanced analytics and machine learning algorithms can enhance decision-making processes, while upgraded hardware can improve system durability and accuracy.

With these advancements, the IoT-based plant watering system has the potential to play a pivotal role in the evolution of smart farming technologies. It offers a pathway toward promoting sustainable agricultural practices, addressing global food security challenges, and contributing to the broader goals of environmental conservation and resource optimization.

9. REFERENCES

1. Mouha, R. a. R. A. (2021). Internet of Things (IoT). *Journal of Data Analysis and Information Processing*, *09*(02), 77–101. https://doi.org/10.4236/jdaip.2021.92006

2. Siskandar, R., Fadhil, M. A., Kusumah, B. R., Irmansyah, I., & Irzaman, I. (2020). INTERNET OF THINGS: AUTOMATIC PLANT WATERING SYSTEM USING ANDROID. Jurnal Teknik Pertanian Lampung (Journal of Agricultural Engineering), 9(4), 297. https://doi.org/10.23960/jtep-l.v9i4.297-310

3. Subhajit. (2023, February 20). Indoor Plant Watering System project using ESP32 - 2025. *IotCircuitHub*. https://iotcircuithub.com/indoor-plant-watering-system-esp32/

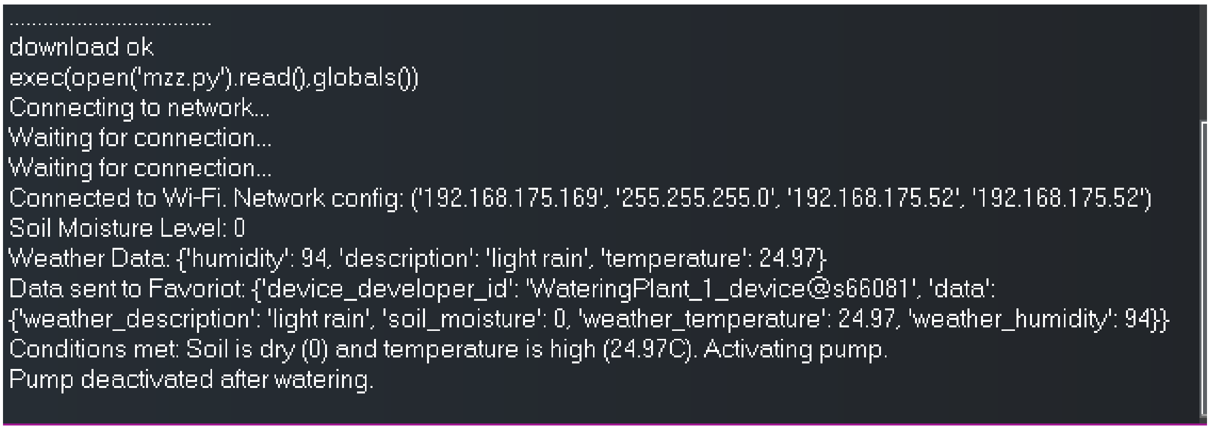
10.APPENDIXES

Appendix 1: Source Code

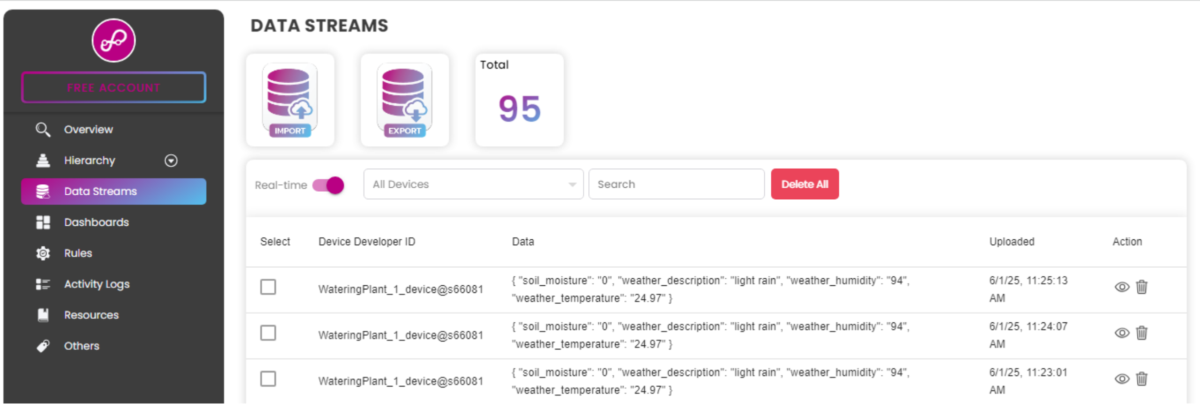
```
import network

import time

from machine import Pin, ADC

from umqtt.simple import MQTTClient

import urequests  # For HTTP requests



# Wi-Fi credentials

SSID = "Babababa"

PASSWORD = "nananana789"



# Favoriot MQTT settings

SERVER = "mqtt.favoriot.com"

DEVICE_ACCESS_TOKEN = "iJmjTgfOMMscTCOgKj4727gUHsAQhTfR"

TOPIC = f"{DEVICE_ACCESS_TOKEN}/v2/streams"



# OpenWeatherMap API settings

WEATHER_API_KEY = "0d898d27813b427d7abcdb960b6883c1"

CITY_NAME = "Kuala Terengganu"
```

```
WEATHER_API_URL =
f"http://api.openweathermap.org/data/2.5/weather?q=Kuala%20Terengganu&appid=0
d898d27813b427d7abcdb960b6883c1&units=metric"

# Thresholds
TEMPERATURE_THRESHOLD = 37  # Temperature in °C
MOISTURE_THRESHOLD = 1000    # Dry soil threshold (higher value = wetter soil)
WATERING_DURATION = 5        # Time to activate pump (in seconds)

# Initialize soil moisture sensor
soil_sensor = ADC(Pin(34))
soil_sensor.atten(ADC.ATTN_11DB)

# Initialize the relay to control the water pump
relay = Pin(18, Pin.OUT)
relay.value(0)  # Ensure the pump is off initially

# Connect to Wi-Fi
def connect_to_wifi():
    wlan = network.WLAN(network.STA_IF)
    wlan.active(True)
    if not wlan.isconnected():
        print("Connecting to network...")
        wlan.connect(SSID, PASSWORD)
        while not wlan.isconnected():
            time.sleep(1)
            print("Waiting for connection...")
    print("Connected to Wi-Fi. Network config:", wlan.ifconfig())


# Get weather data
def get_weather():
    try:
        response = urequests.get(WEATHER_API_URL)
        if response.status_code == 200:
            data = response.json()
            temp = data["main"]["temp"]
            humidity = data["main"]["humidity"]
            weather_desc = data["weather"][0]["description"]
            return {"temperature": temp, "humidity": humidity, "description": weather_desc}
        else:
            print("Failed to fetch weather data:", response.text)
            return None
    except Exception as e:
        print("Error fetching weather data:", e)
        return None


# Main loop
def main():
    # Connect to Wi-Fi
    connect_to_wifi()

    # Set up MQTT client
    client = MQTTClient("esp32_client", SERVER, user=DEVICE_ACCESS_TOKEN,
password=DEVICE_ACCESS_TOKEN)

    while True:
        try:
            # Read soil moisture value
            moisture_value = soil_sensor.read()
            print("Soil Moisture Level:", moisture_value)

            # Fetch weather data
            weather_data = get_weather()
            if weather_data:
                print("Weather Data:", weather_data)

            # Prepare payload for Favoriot
            payload = {
                "device_developer_id": "WateringPlant_1_device@s66081",
                "data": {
                    "soil_moisture": moisture_value,
                    "weather_temperature": weather_data["temperature"] if weather_data else
None,
                    "weather_humidity": weather_data["humidity"] if weather_data else None,
                    "weather_description": weather_data["description"] if weather_data else None,
                }
            }

            # Publish data to Favoriot
            client.connect()
            client.publish(TOPIC, str(payload).replace("'", '"'))
            print("Data sent to Favoriot:", payload)
            client.disconnect()

            # Control the water pump based on soil moisture and temperature
            if (moisture_value < MOISTURE_THRESHOLD) and (weather_data and weather_data["temperature"] > TEMPERATURE_THRESHOLD):
                print(f"Conditions met: Soil is dry ({moisture_value}) and temperature is high ({weather_data['temperature']}°C). Activating pump.")
                relay.value(1)  # Turn on the pump
                time.sleep(WATERING_DURATION)  # Keep the pump on for the specified duration
                relay.value(0)  # Turn off the pump
                print("Pump deactivated after watering.")
            else:
                print("Conditions not met. Pump remains off.")
                relay.value(0)  # Ensure the pump stays off
                except Exception as e:
                    print("Error:", e)
                    relay.value(0)  # Turn off the pump in case of error
                    client.disconnect()  # Disconnect MQTT client
                    time.sleep(5)  # Wait before retrying

                # Wait for 60 seconds before the next iteration
                time.sleep(60)

# Run the main loop
main()
```

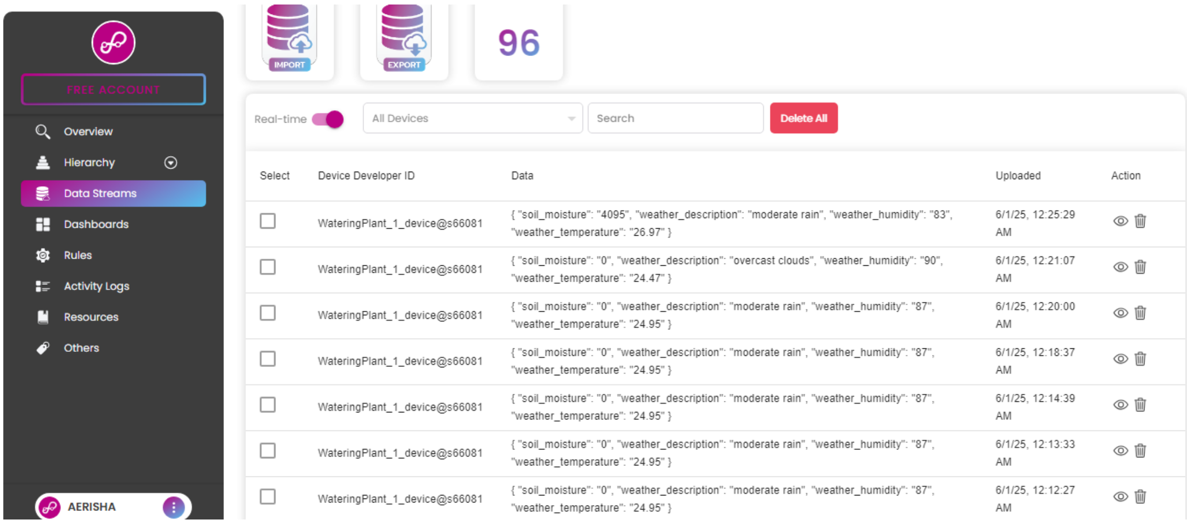
Appendix 2: Screenshots of output



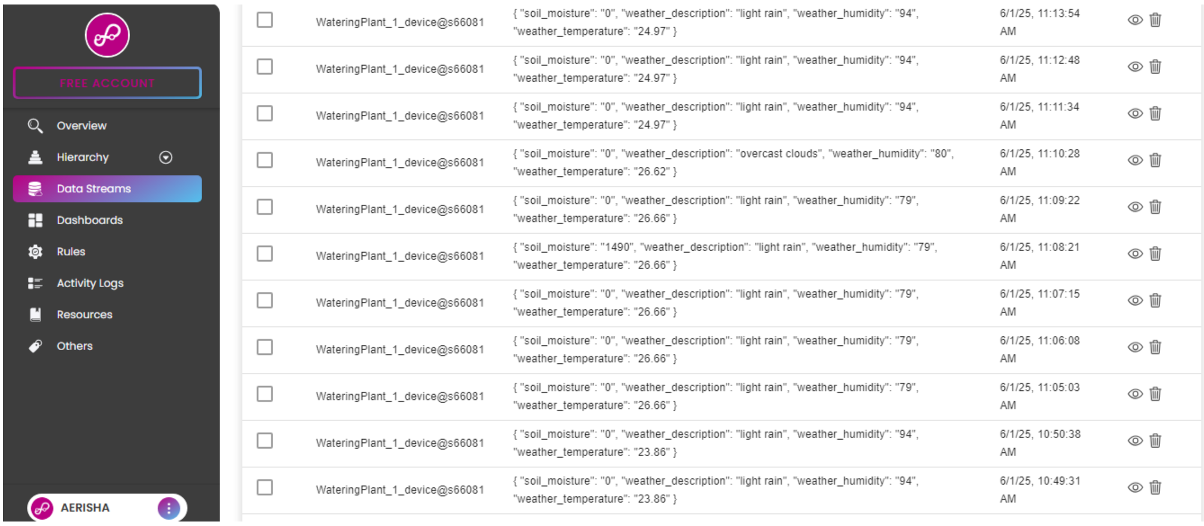
```
..................................
download ok
exec(open('mzz.py').read(),globals())
Connecting to network...
Waiting for connection...
Waiting for connection...
Connected to Wi-Fi. Network config: ('192.168.175.169', '255.255.255.0', '192.168.175.52', '192.168.175.52')
Soil Moisture Level: 0
Weather Data: {'humidity': 94, 'description': 'light rain', 'temperature': 24.97}
Data sent to Favoriot: {'device_developer_id': 'WateringPlant_1_device@s66081', 'data':
{'weather_description': 'light rain', 'soil_moisture': 0, 'weather_temperature': 24.97, 'weather_humidity': 94}}
Conditions met: Soil is dry (0) and temperature is high (24.97C). Activating pump.
Pump deactivated after watering.
```



Select	Device Developer ID	Data	Uploaded	Action
□	WateringPlant_1_device@s66081	{ "soil_moisture": "0", "weather_description": "light rain", "weather_humidity": "94", "weather_temperature": "24.97" }	6/1/25, 11:25:13 AM	
□	WateringPlant_1_device@s66081	{ "soil_moisture": "0", "weather_description": "light rain", "weather_humidity": "94", "weather_temperature": "24.97" }	6/1/25, 11:24:07 AM	
□	WateringPlant_1_device@s66081	{ "soil_moisture": "0", "weather_description": "light rain", "weather_humidity": "94", "weather_temperature": "24.97" }	6/1/25, 11:23:01 AM	



Select	Device Developer ID	Data	Uploaded	Action
□	WateringPlant_1_device@s66081	{ "soil_moisture": "4095", "weather_description": "moderate rain", "weather_humidity": "83", "weather_temperature": "26.97" }	6/1/25, 12:25:29 AM	
□	WateringPlant_1_device@s66081	{ "soil_moisture": "0", "weather_description": "overcast clouds", "weather_humidity": "90", "weather_temperature": "24.47" }	6/1/25, 12:21:07 AM	
□	WateringPlant_1_device@s66081	{ "soil_moisture": "0", "weather_description": "moderate rain", "weather_humidity": "87", "weather_temperature": "24.95" }	6/1/25, 12:20:00 AM	
□	WateringPlant_1_device@s66081	{ "soil_moisture": "0", "weather_description": "moderate rain", "weather_humidity": "87", "weather_temperature": "24.95" }	6/1/25, 12:18:37 AM	
□	WateringPlant_1_device@s66081	{ "soil_moisture": "0", "weather_description": "moderate rain", "weather_humidity": "87", "weather_temperature": "24.95" }	6/1/25, 12:14:39 AM	
□	WateringPlant_1_device@s66081	{ "soil_moisture": "0", "weather_description": "moderate rain", "weather_humidity": "87", "weather_temperature": "24.95" }	6/1/25, 12:13:33 AM	
□	WateringPlant_1_device@s66081	{ "soil_moisture": "0", "weather_description": "moderate rain", "weather_humidity": "87", "weather_temperature": "24.95" }	6/1/25, 12:12:27 AM	



□	WateringPlant_1_device@s66081	{ "soil_moisture": "0", "weather_description": "light rain", "weather_humidity": "94", "weather_temperature": "24.97" }	6/1/25, 11:13:54 AM
□	WateringPlant_1_device@s66081	{ "soil_moisture": "0", "weather_description": "light rain", "weather_humidity": "94", "weather_temperature": "24.97" }	6/1/25, 11:12:48 AM
□	WateringPlant_1_device@s66081	{ "soil_moisture": "0", "weather_description": "light rain", "weather_humidity": "94", "weather_temperature": "24.97" }	6/1/25, 11:11:34 AM
□	WateringPlant_1_device@s66081	{ "soil_moisture": "0", "weather_description": "overcast clouds", "weather_humidity": "80", "weather_temperature": "26.62" }	6/1/25, 11:10:28 AM
□	WateringPlant_1_device@s66081	{ "soil_moisture": "0", "weather_description": "light rain", "weather_humidity": "79", "weather_temperature": "26.66" }	6/1/25, 11:09:22 AM
□	WateringPlant_1_device@s66081	{ "soil_moisture": "1490", "weather_description": "light rain", "weather_humidity": "79", "weather_temperature": "26.66" }	6/1/25, 11:08:21 AM
□	WateringPlant_1_device@s66081	{ "soil_moisture": "0", "weather_description": "light rain", "weather_humidity": "79", "weather_temperature": "26.66" }	6/1/25, 11:07:15 AM
□	WateringPlant_1_device@s66081	{ "soil_moisture": "0", "weather_description": "light rain", "weather_humidity": "79", "weather_temperature": "26.66" }	6/1/25, 11:06:08 AM
□	WateringPlant_1_device@s66081	{ "soil_moisture": "0", "weather_description": "light rain", "weather_humidity": "79", "weather_temperature": "26.66" }	6/1/25, 11:05:03 AM
□	WateringPlant_1_device@s66081	{ "soil_moisture": "0", "weather_description": "light rain", "weather_humidity": "94", "weather_temperature": "23.86" }	6/1/25, 10:50:38 AM
□	WateringPlant_1_device@s66081	{ "soil_moisture": "0", "weather_description": "light rain", "weather_humidity": "94", "weather_temperature": "23.86" }	6/1/25, 10:49:31 AM

"

Appendix 3: Task Distribution among group members

NAME	TASK
AERISHA FASHAHIRA BINTI MOHD PUAD	1. Intro 1.1 Background 1.2 Problem Statement 1.3 Objectives 1.4 scope of the project 5. Result and analysis 5.1 data collected 7. future work 7.1 potential enhancement
HAZRIEN NUR QISTINA BINTI HASWADI	2. Literature Review 2.1 Overview of IoT 2.2 Related work 2.3 Gap 6. challenges and limitations 6.1 Technical challenge 6.2 constraints faced * Slide Presentation
PUTRI BALQIS BINTI HANAFI	3. System Design 3.1 System Architecture 3.2 Hardware components 3.3 Software components 4. implementation 4.1 Setup and configuration 4.2 integration of components 4.3 Data Flow

Appendix 4: PowerPoint presentation Slide

GROUP 5

WATERING PLANT SENSOR

OUR TEAM

S66081

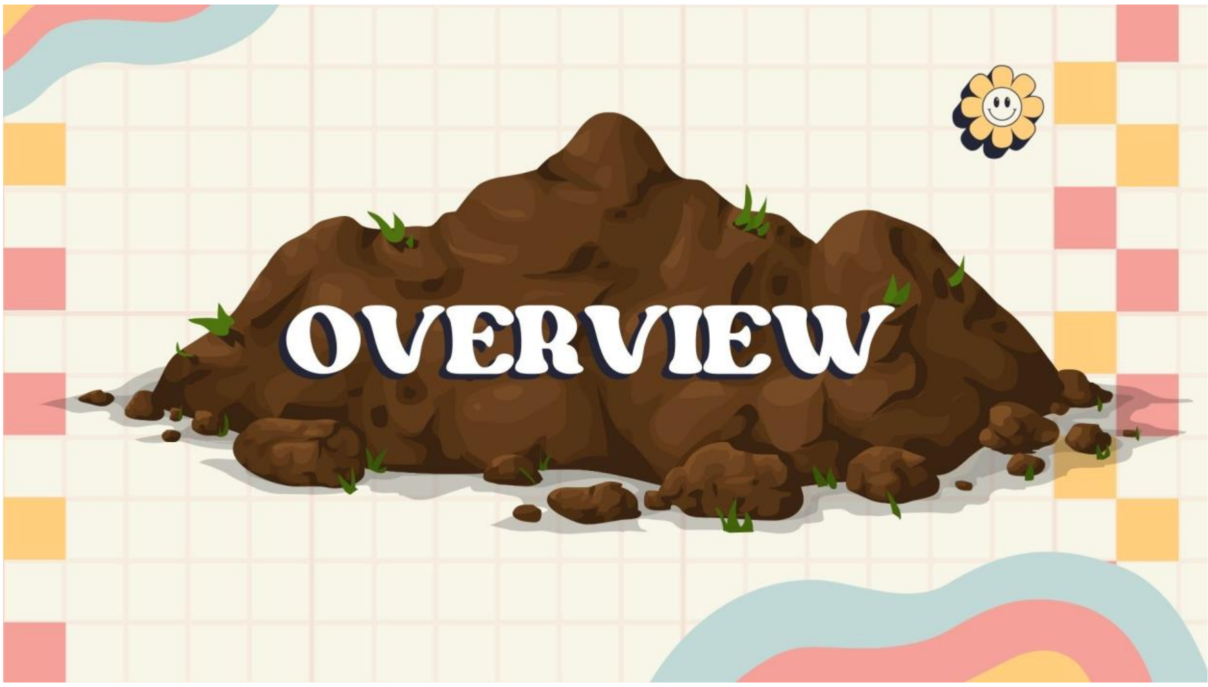
AERISHA FASHAHIRA BINTI MOHD PUAD

S67017

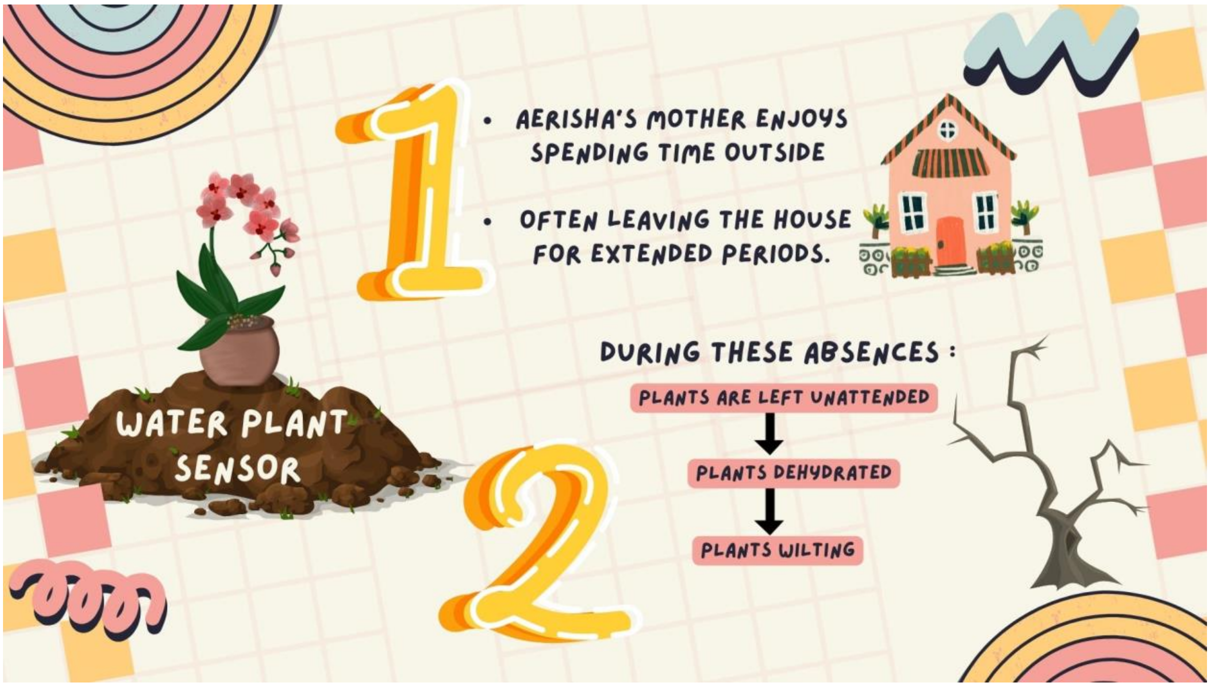
HAZRIEN NUR QISTINA BINTI HASWADI

S65630

PUTRI BALQIS BINTI HANAFI



OVERVIEW



WATER PLANT SENSOR

1

- AERISHA'S MOTHER ENJOYS SPENDING TIME OUTSIDE
- OFTEN LEAVING THE HOUSE FOR EXTENDED PERIODS.

2

DURING THESE ABSENCES :

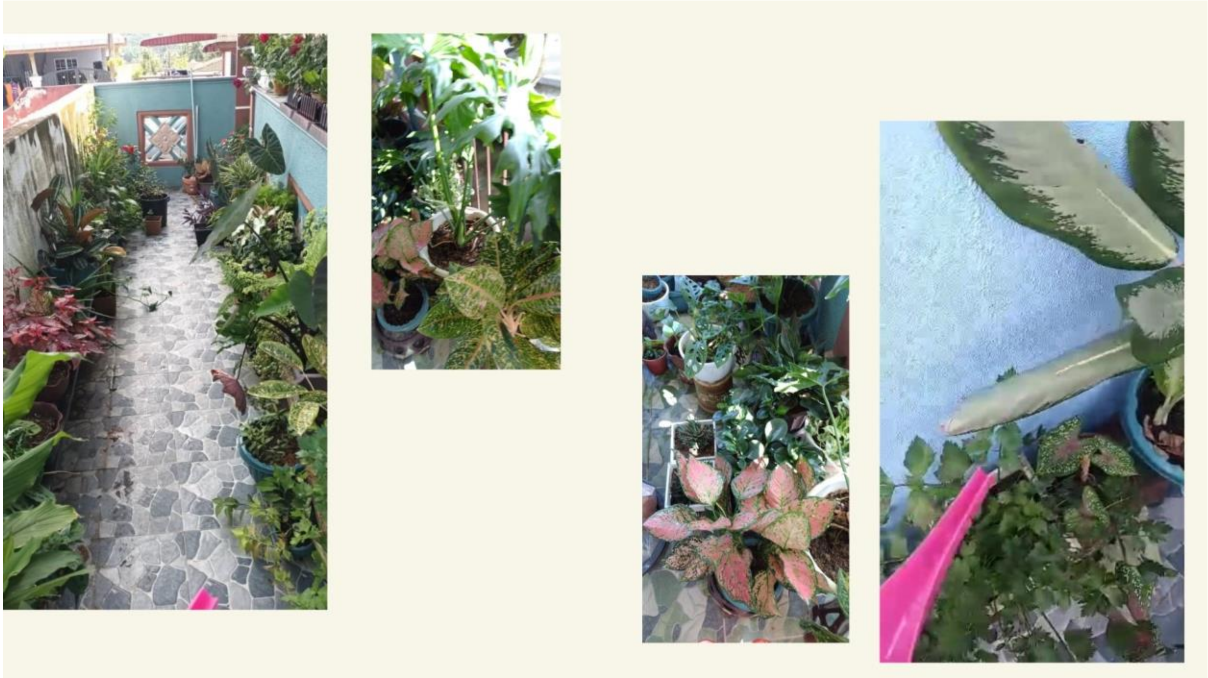
PLANTS ARE LEFT UNATTENDED

↓

PLANTS DEHYDRATED

↓

PLANTS WILTING

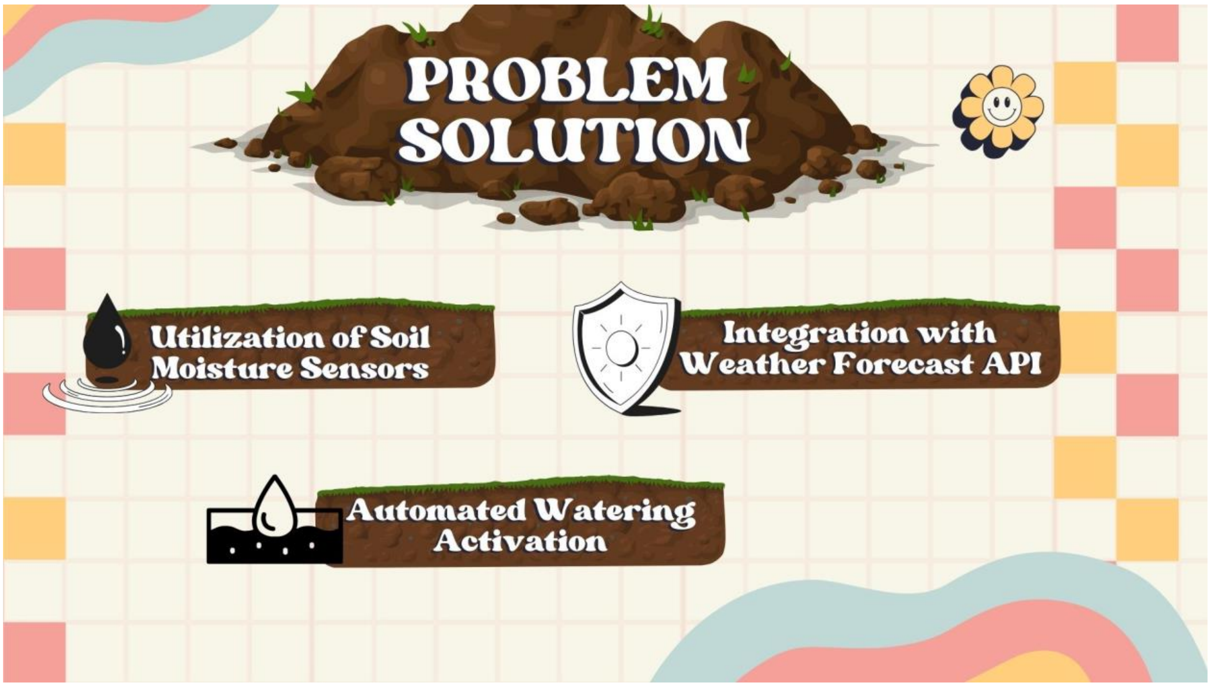


PROBLEM STATEMENT

Insufficient knowledge about watering needs

Overwatering plant

People always forget to watering the plant

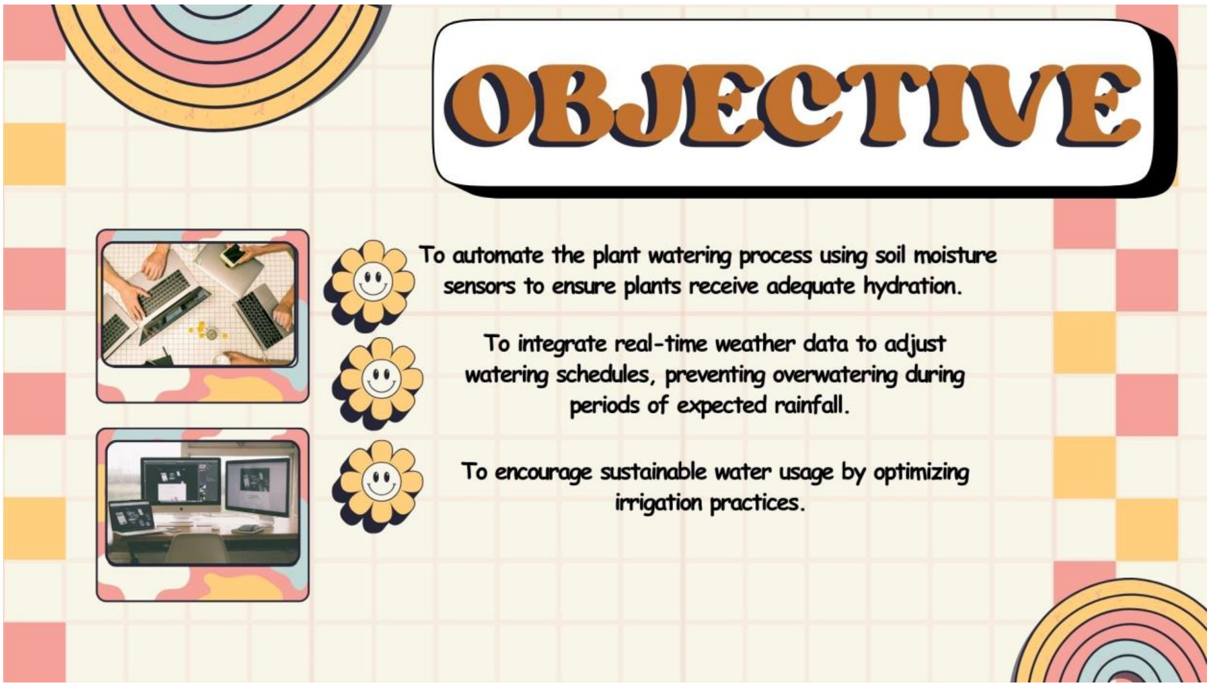


PROBLEM SOLUTION

Utilization of Soil Moisture Sensors

Integration with Weather Forecast API

Automated Watering Activation

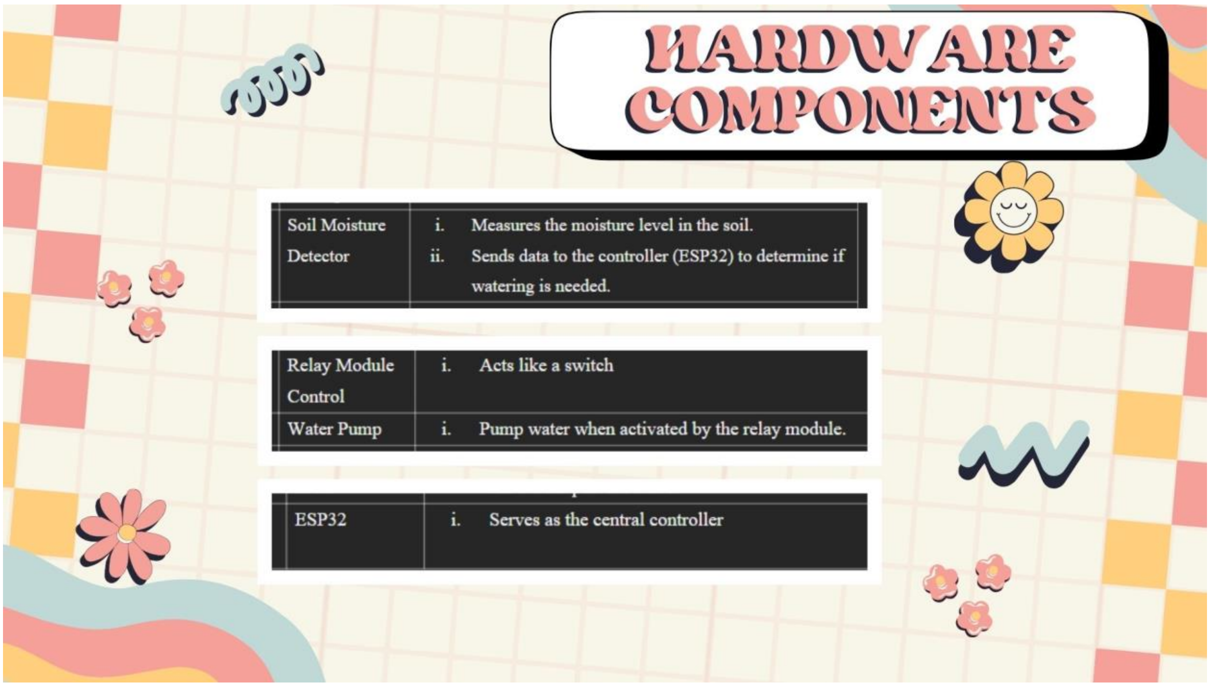


OBJECTIVE

To automate the plant watering process using soil moisture sensors to ensure plants receive adequate hydration.

To integrate real-time weather data to adjust watering schedules, preventing overwatering during periods of expected rainfall.

To encourage sustainable water usage by optimizing irrigation practices.

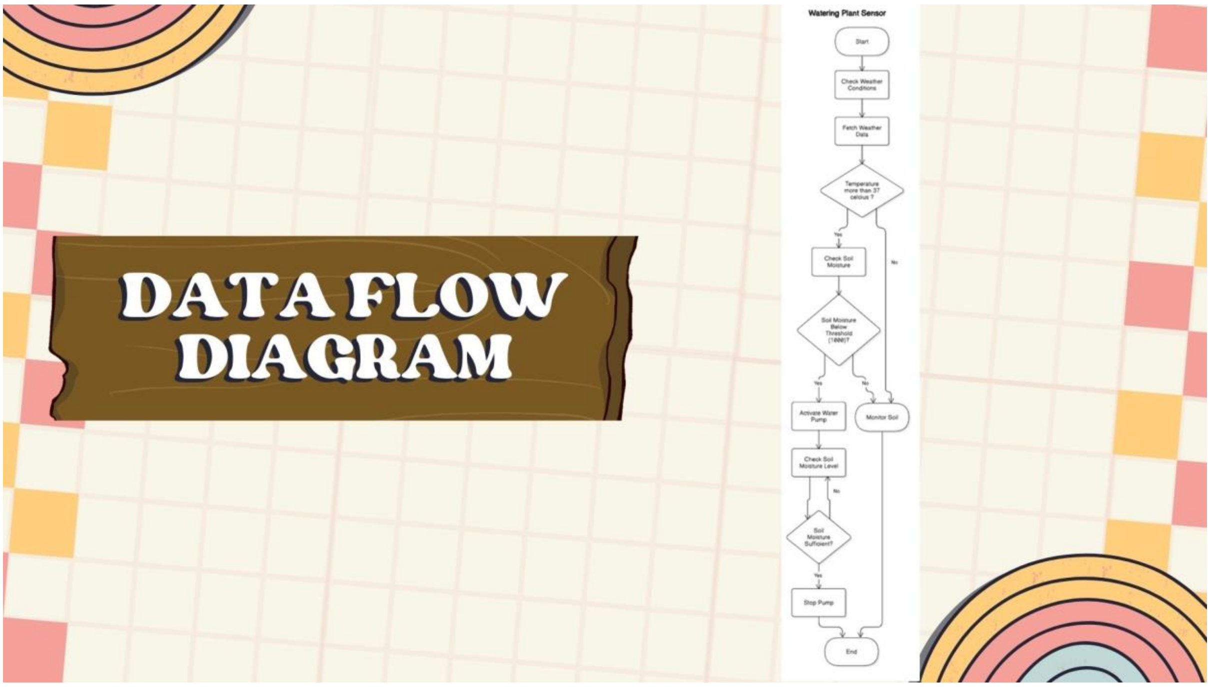


HARDWARE COMPONENTS

Soil Moisture Detector	i. Measures the moisture level in the soil. ii. Sends data to the controller (ESP32) to determine if watering is needed.

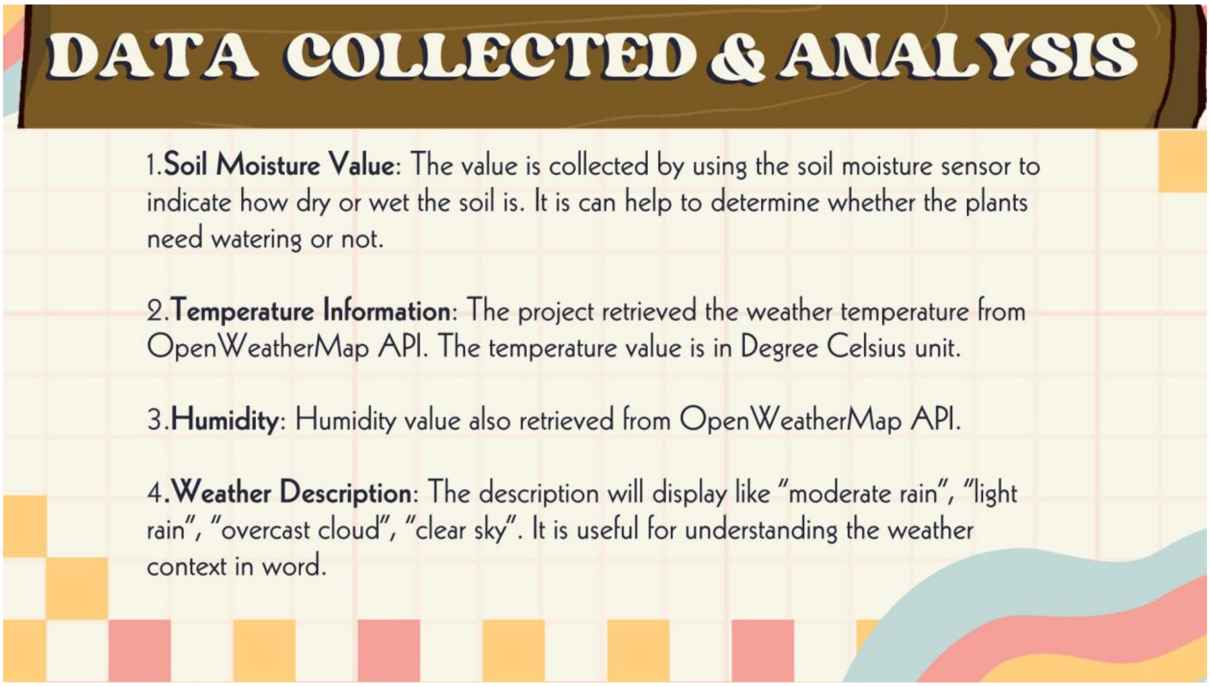
Relay Module Control	i. Acts like a switch
Water Pump	i. Pump water when activated by the relay module.

ESP32	i. Serves as the central controller



DATA FLOW DIAGRAM

RESULT & ANALYSIS



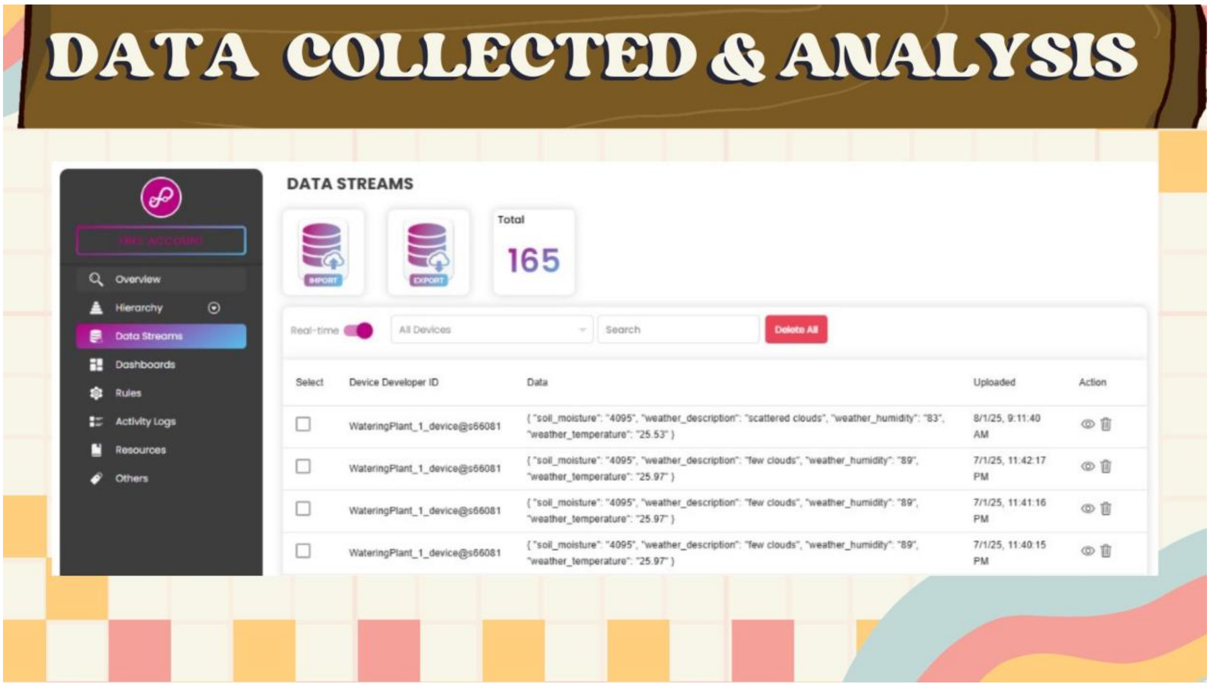
DATA COLLECTED & ANALYSIS

1.**Soil Moisture Value**: The value is collected by using the soil moisture sensor to indicate how dry or wet the soil is. It is can help to determine whether the plants need watering or not.

2.**Temperature Information**: The project retrieved the weather temperature from OpenWeatherMap API. The temperature value is in Degree Celsius unit.

3.**Humidity**: Humidity value also retrieved from OpenWeatherMap API.

4.**Weather Description**: The description will display like "moderate rain", "light rain", "overcast cloud", "clear sky". It is useful for understanding the weather context in word.



DATA COLLECTED & ANALYSIS

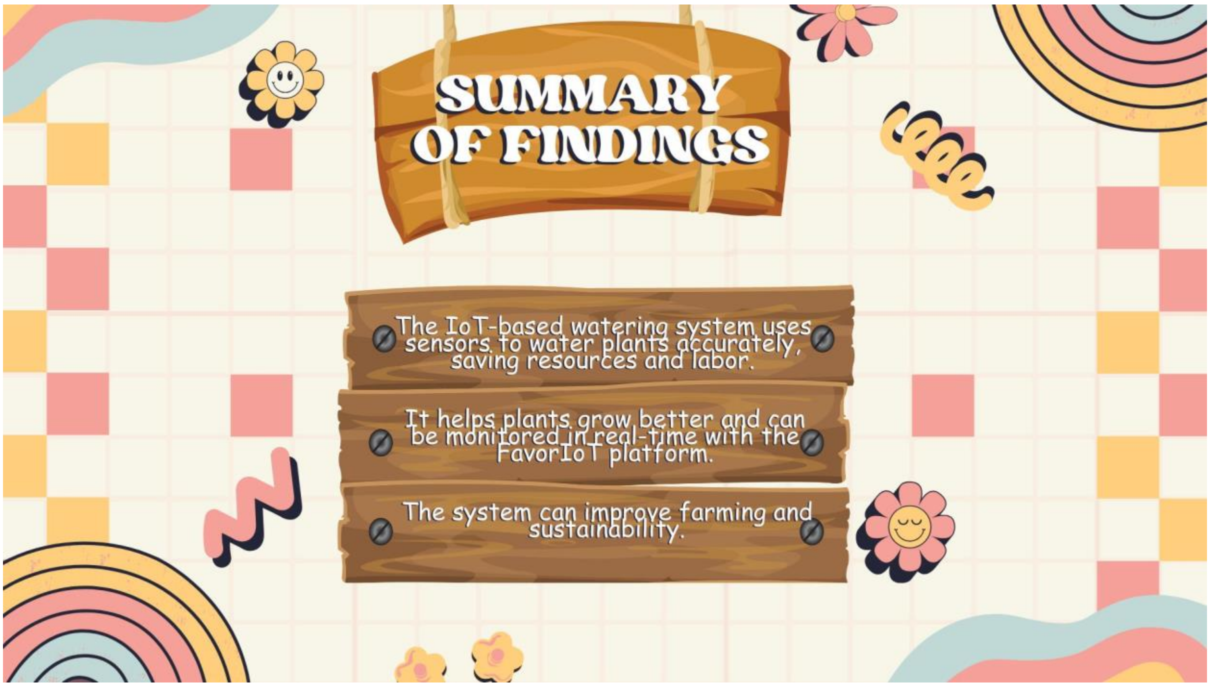
DATA STREAMS

Total

165

Select	Device Developer ID	Data	Uploaded	Action
□	WateringPlant_1_device@s66081	{ "soil_moisture": "4095", "weather_description": "scattered clouds", "weather_humidity": "83", "weather_temperature": "25.53" }	8/1/25, 9:11:40 AM	
□	WateringPlant_1_device@s66081	{ "soil_moisture": "4095", "weather_description": "few clouds", "weather_humidity": "89", "weather_temperature": "25.97" }	7/1/25, 11:42:17 PM	
□	WateringPlant_1_device@s66081	{ "soil_moisture": "4095", "weather_description": "few clouds", "weather_humidity": "89", "weather_temperature": "25.97" }	7/1/25, 11:41:16 PM	
□	WateringPlant_1_device@s66081	{ "soil_moisture": "4095", "weather_description": "few clouds", "weather_humidity": "89", "weather_temperature": "25.97" }	7/1/25, 11:40:15 PM	

CONCLUSION

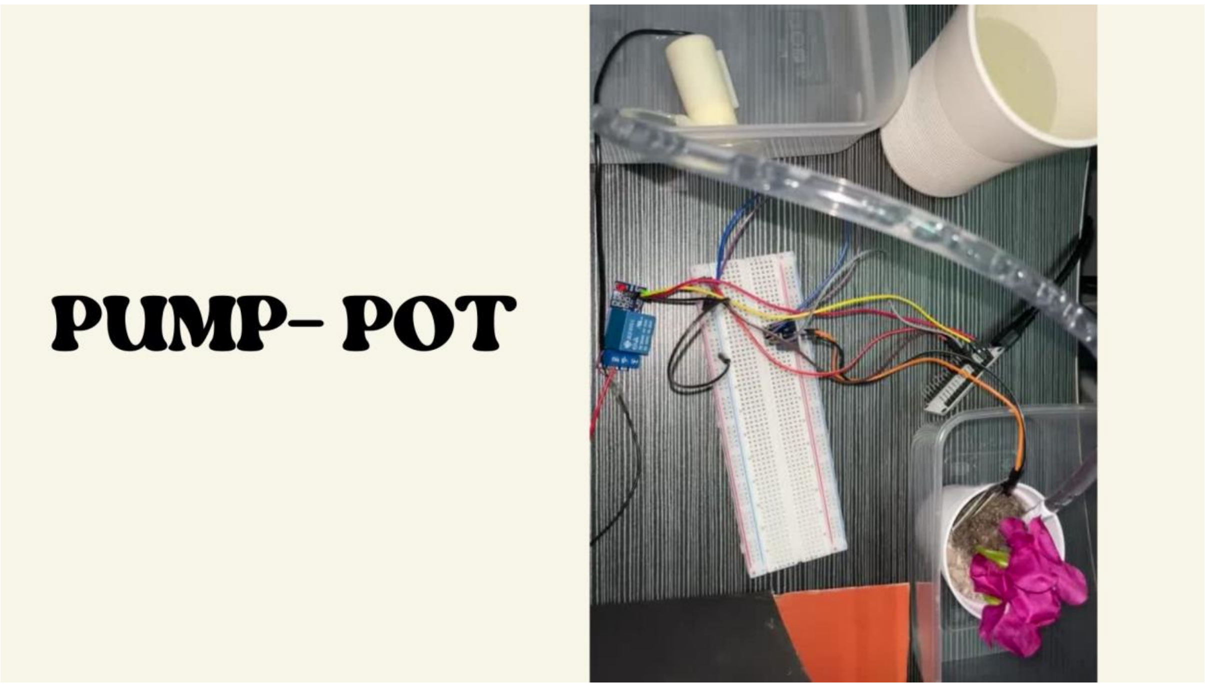


SUMMARY OF FINDINGS

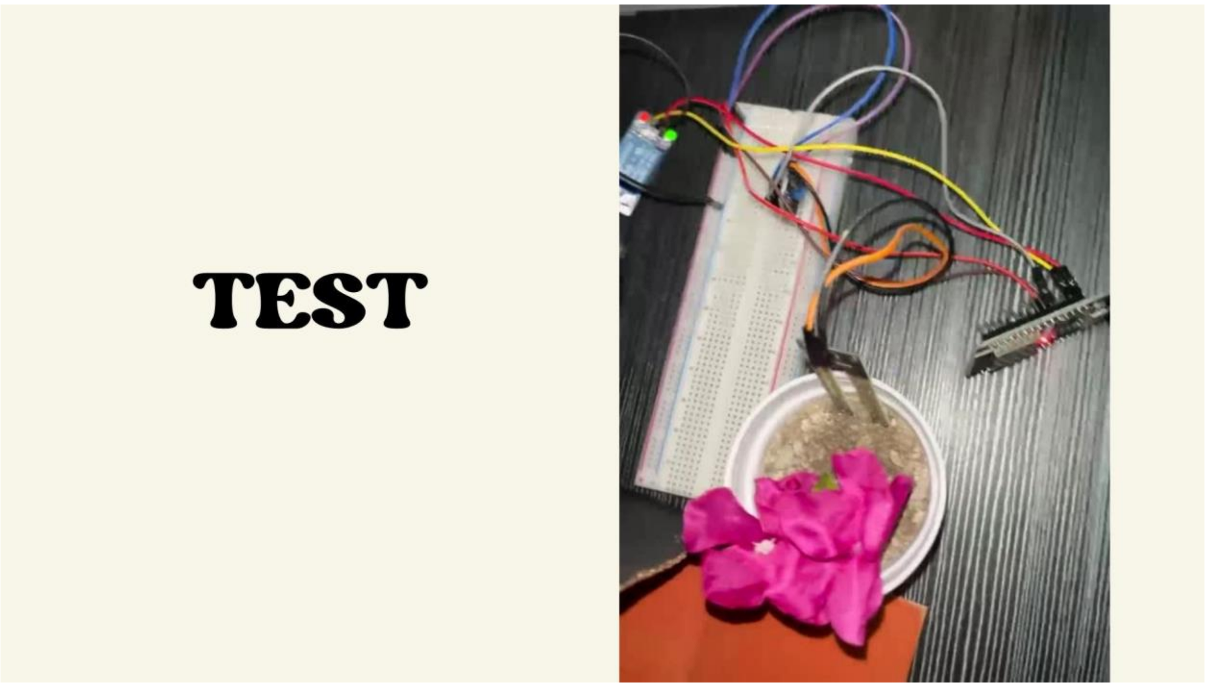
The IoT-based watering system uses sensors to water plants accurately, saving resources and labor.

It helps plants grow better and can be monitored in real-time with the FavorIoT platform.

The system can improve farming and sustainability.



PUMP- POT



TEST

THANK YOU