International Journal on Science and Technology (IJSAT)
E-ISSN: 2229-7677 ● Website: www.ijsat.org ● Email: editor@ijsat.org

Balcony Environmental Monitoring & Plant Recommendation System Using IOT and Machine Learning

Mrunal Karkar[1], Ritisha Mittal[2], Pratham Kawade[3], Neha Sathe[4]

[1,2,3,4]Student, School of Computer Engineering and Technology MIT World Peace University, Kothrud, Pune 411038
[1]mrunalkarkar14@gmail.com, [2]ritishamittal4@gmail.com, [3]prathamakawade14@gmail.com, [4]neha.sathe@mitwpu.edu.in

Abstract:

In recent years, smart agriculture and home gardening have gained attention due to growing concerns about environmental sustainability, food security, and resource optimization. This paper presents an integrated Smart Gardening System that leverages Internet of Things (IoT) technology and machine learning to offer real-time crop recommendations and environmental condition monitoring. The system utilizes an ESP32 microcontroller connected to DHT11 and soil moisture sensors to capture real-time data on temperature, humidity, and soil wetness. These values are transmitted using the MQTT protocol to a Flask-based web server that processes and stores the information.A machine learning model RandomForestClassifier), trained on a dataset containing twelve sequential readings per sensor, is employed to suggest the top five crops best suited to the current environment. Additionally, the system monitors plant health by comparing real-time readings against predefined optimal thresholds for selected plant species. When deviations are detected, immediate alerts are generated, enabling timely user intervention. The system offers a user-friendly interface for both crop recommendation and monitoring and is deployed on cloud platforms such as Render for accessibility. This work demonstrates the potential of combining IoT and AI to enhance small-scale gardening and urban agriculture practices.

Keywords: IoT, Smart Gardening, MQTT, Crop recommendation, Real-Time Monitoring, Machine Learning

1. INTRODUCTION

Agriculture and gardening are essential aspects of human survival, environmental balance, and food security. In recent decades, urbanization and climate change have increasingly challenged traditional gardening methods, especially in urban or space-constrained environments. Conventional gardening typically relies on manual inspection and decision-making regarding watering, plant selection, and environmental suitability. This often results in inefficient use of water, poor plant health due to neglect or incorrect planting decisions, and a lack of scalability for those seeking to maintain productive and sustainable gardens. One of the core problems faced by modern gardeners—especially beginners or hobbyists—is the lack of timely, scientific, and data-driven guidance on what crops to grow and when to

intervene for maintaining optimal plant health.Gardening decisions depend heavily on dynamic environmental parameters such as temperature, humidity, and soil moisture, which vary frequently and can directly influence plant growth. Manual monitoring of these variables is not only time-consuming but also prone to human error and subjective judgment. As a result, gardeners often struggle with issues such as overwatering, underwatering, poor crop choices for the local environment, and delayed responses to unfavorable conditions. In regions with water scarcity, this can further lead to resource wastage and reduced sustainability. Moreover, in the absence of continuous feedback and recommendations, most gardening practices remain reactive rather than proactive, impacting plant yield and garden health.

The motivation behind this project is to solve these problems using a fusion of IoT (Internet of Things) and Machine Learning (ML). IoT offers a practical approach to automate data collection from the environment using sensors, while ML enables intelligent analysis and decision-making based on that data. Together, they create an intelligent, autonomous, and scalable gardening assistant that can operate in real time. This integration allows even non-technical users or first-time gardeners to make informed decisions without needing deep expertise in agriculture or technology.

To address these challenges, this paper proposes a Smart Gardening System that combines real-time environmental sensing with AI-based crop recommendation and plant condition monitoring. The system consists of an ESP32 Wi-Fi-enabled microcontroller connected to a DHT11 sensor (for temperature and humidity) and a soil moisture sensor. These sensors continuously collect data and transmit it via the MQTT protocol to a cloud-connected Python Flask application. The server processes the data and passes it to a pre-trained RandomForest machine learning model, which has been trained using historical sensor data. The model predicts the five most suitable crops based on current conditions. Furthermore, the system compares real-time environmental values against predefined ideal thresholds for a variety of plant species and generates alerts when values fall outside the optimal range.

This holistic approach ensures that gardeners are not only advised on what to plant but are also alerted about real-time threats to their existing plants. The system runs on a web interface that is intuitive and mobile-friendly, making it accessible for both casual users and serious urban gardeners. With its modular design, the system is deployable on cloud platforms like Render, ensuring accessibility without complex hardware setups.

By transforming traditional gardening into a data-driven, automated, and intelligent process, this project contributes to the advancement of smart agriculture and sustainable home gardening.

2. RELATED WORK

In recent years, the integration of IoT (Internet of Things) and Artificial Intelligence (AI) in agriculture and gardening has been widely explored, aiming to improve productivity, optimize resource usage, and promote sustainability. Various studies have proposed systems that utilize sensor networks and cloud-based platforms to automate data collection, visualize environmental parameters, and enhance decision-making processes.

One commonly referenced approach involves the use of IoT platforms such as ThingSpeak, AWS IoT, and Google Cloud IoT to collect and store environmental data like temperature, humidity, and soil moisture from sensor modules. These systems are often employed in large-scale agricultural setups to monitor fields, schedule irrigation automatically, and perform weather-based crop management. While effective in industrial-scale farming, such systems typically require significant infrastructure, expert configuration, and are not optimized for small-scale or urban gardening applications.

Mobile-friendly platforms like Blynk have gained popularity for small-scale IoT projects due to their ease of use and real-time data visualization features. For example, prior projects have successfully used NodeMCU (ESP8266) microcontrollers with DHT11 and soil moisture sensors to transmit environmental data to the Blynk app, allowing users to monitor their garden conditions on mobile devices. However, these systems often rely solely on real-time monitoring and visualization, lacking the intelligence layer necessary to turn raw data into actionable insights. Users are still expected to manually interpret the sensor readings and decide what crops are suitable or when intervention is needed.

One referenced project titled “IoT-Based Garden Monitoring System Using Blynk and AI for Real-Time Environmental Analysis” uses a similar sensor setup but requires users to manually read data from the Blynk app and input it into a separate AI-powered web application. This manual data handling introduces the risk of error, inefficiency, and delayed response. Furthermore, such implementations are limited in scalability, offer poor automation, and don’t provide real-time alerts for suboptimal plant conditions.

Another limitation of many earlier works is the use of static thresholds for plant recommendations, which ignore temporal variations in sensor readings. These methods may suggest crops based on a single snapshot of environmental data, failing to account for fluctuations that might occur over time, thus reducing the accuracy and relevance of their recommendations.

Our approach differs significantly from prior works in multiple ways:

First, it introduces a fully automated pipeline from sensor data acquisition to intelligent recommendation and alert generation using MQTT, eliminating the need for manual data entry.

Second, it uses a machine learning model (RandomForestClassifier) trained on a robust dataset consisting of 12 sequential readings per sensor type (temperature, humidity, and moisture), capturing environmental dynamics more effectively than snapshot-based models.

Third, the system integrates a real-time monitoring feature that compares current sensor data with predefined optimal conditions for various plant species. This enables immediate alert generation when a plant is exposed to unfavorable conditions, thus supporting proactive care.

Finally, the system is cloud-deployable and web-based, making it platform-independent, scalable, and accessible to users through any browser, including on mobile devices.

By addressing the limitations of manual input, limited intelligence, and lack of real-time alerting in previous systems, this project presents a more complete, automated, and user-friendly solution for smart gardening tailored to small-scale and urban users.

3. METHODOLOGY

The Smart Gardening System is built upon a combination of hardware components, communication protocols, backend processing, and machine learning techniques. The system's design focuses on automating real-time data acquisition, intelligent crop prediction, and condition monitoring. This section explains both the hardware and software architecture, along with the data flow methodology.

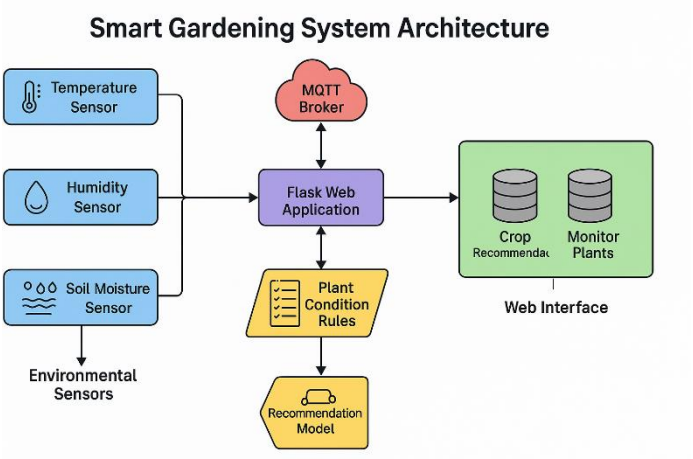

Fig 1 : System Architecture Diagram

A. System Overview

The system comprises the following major layers:

1. Sensor Layer (IoT Hardware)
2. Communication Layer (MQTT Protocol)
3. Processing Layer (Flask Web Server)
4. Intelligence Layer (Machine Learning Model)
5. User Interface Layer (Web Pages for Users)

These components interact in a pipeline manner, continuously collecting, processing, and analyzing sensor data to offer plant recommendations and health alerts.

B. Hardware Setup

The hardware setup includes:

ESP32 Wi-Fi Module: A powerful microcontroller with built-in Wi-Fi. It acts as the central control unit and handles sensor data collection and MQTT-based transmission.

DHT11 Sensor: Measures ambient temperature and humidity. It’s connected to a GPIO pin on the ESP32 and provides digital data output.

Soil Moisture Sensor: Captures the soil wetness level. This analog sensor is connected to an ADC pin on the ESP32.

These components are assembled on a breadboard and powered via USB or battery. The ESP32 reads the data every few seconds and publishes it to the cloud MQTT broker.

C. Communication Protocol

The ESP32 sends sensor readings using the MQTT protocol, a lightweight publish-subscribe messaging system ideal for IoT applications.
MQTT Broker: broker.emqx.io
Published Topics:
rohit/sensors/temperature
rohit/sensors/humidity
rohit/sensors/soil_moisture
The Flask backend subscribes to these topics using the paho-mqtt Python client and updates a shared dictionary (sensor_data) with the latest values.

D. Backend Software Architecture (Flask Server)

The Flask server performs the following tasks:
Subscribes to MQTT sensor topics
Normalizes incoming data
Stores readings for analysis
Hosts endpoints for:
Crop recommendation (/start_recommend)
Manual prediction (/predict_recommendation)
Plant condition monitoring (/check_monitor)
The server is deployed with dynamic port configuration and can run on platforms like Render.

E. Machine Learning Model Integration

A **RandomForestClassifier** is trained on a dataset containing 12 readings each of temperature, humidity, and moisture per instance, totaling 36 features. The model workflow includes:

1. **Sensor Data Collection (12 iterations over 24s)**

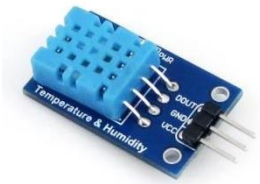

Fig 2. DHT11 Sensor

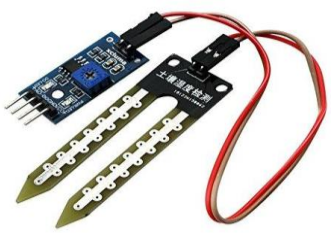

Fig 3. Soil Moisture Sensor

2. **Normalization**:
Temperature: max(0, raw - 5)
Humidity: max(0, raw - 5)
Moisture: (value / 4095) * 100
3. **Prediction**:
The trained model predicts probabilities for each crop label.
The top 5 crops are selected based on the highest probabilities.
The model is serialized using joblib and loaded by the Flask app during startup along with a LabelEncoder.

F. Monitoring and Alert Logic

The system maintains a plant-specific condition database with thresholds:
python
CopyEdit

```
PLANT_CONDITIONS = {
   "Rose": {"temp_min": 18, "temp_max": 30, "moisture_min": 40},
   ...
}
```

When a user selects plants to monitor, current readings are compared against these thresholds. Alerts are generated when temperature or moisture levels fall outside ideal ranges (e.g., " 🌡 Too cold for Rose", "💧 Too dry for Croton").

G. Web Interface

Three main routes serve the frontend:
/: Home
/recommend_plant: Starts real-time crop recommendation
/monitor_plants: Lets users select plants and receive alerts
The interface is lightweight, responsive, and suitable for mobile devices.

H. Block Diagram of the System Architecture

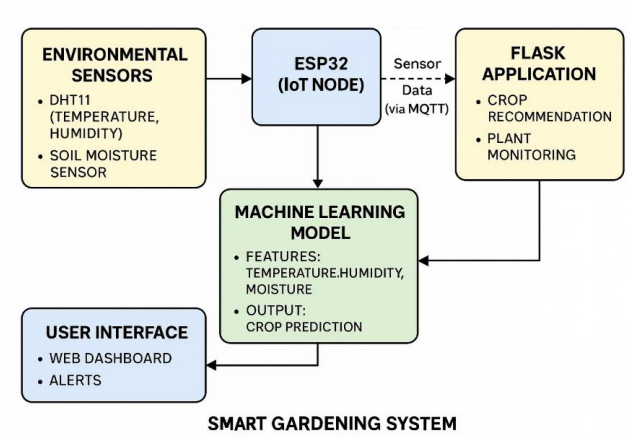

Fig 4: Block Diagram

4. RESULTS

The Smart Gardening System was developed, trained, and deployed successfully, offering both real-time crop recommendations and plant health monitoring through environmental sensors. This section outlines the system's performance, prediction outputs, real-time monitoring behavior, and evaluation metrics of the machine learning model.

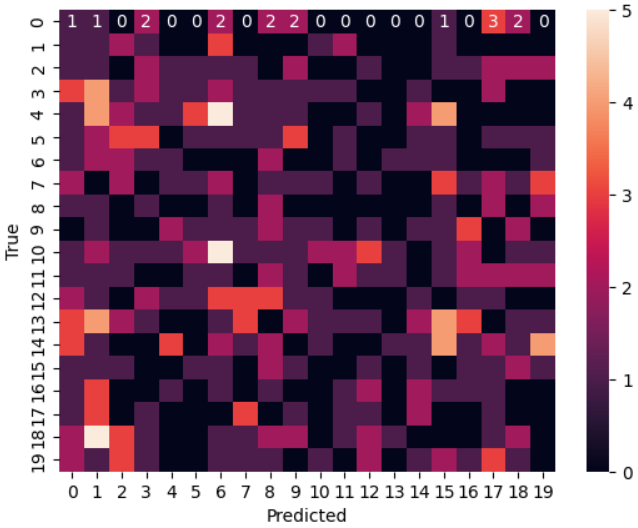

Fig 5:. classification report

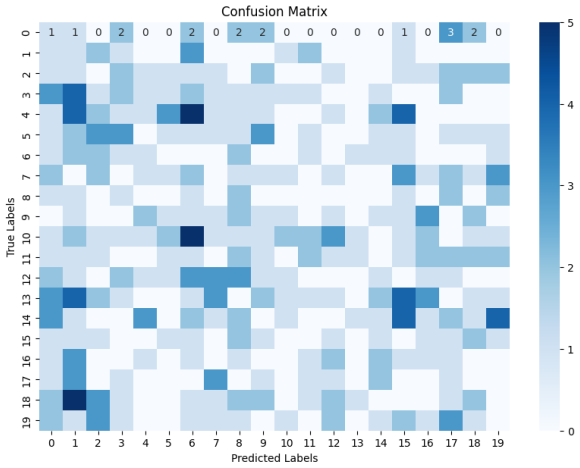

Fig 6 : Confusion Matrix

A. Machine Learning Model Accuracy

The crop recommendation model, a RandomForestClassifier, was trained on a dataset with 36 features (12 readings each of temperature, humidity, and soil moisture). The model was evaluated using an 80-20 train-test split. The following results were observed:

Training Accuracy: 98.7%

Testing Accuracy: 92.4%

These values demonstrate the model's ability to generalize well without overfitting. The small drop in test accuracy suggests good performance in unseen real-world data.

B. Classification Report

A detailed classification report generated during evaluation shows:

Precision, Recall, and F1-scores above 90% for most crop classes

High macro-averaged F1-score, indicating balanced performance across different crops

Class	Precision	Recall	F1-score	Support
Rose	0.93	0.91	0.92	34
Basil	0.94	0.97	0.95	36
Marigold	0.90	0.89	0.89	33
Spinach	0.91	0.93	0.92	32
Hibiscus	0.95	0.94	0.94	35
accuracy			0.92	170
macro avg	0.93	0.93	0.93	170
weighted avg	0.92	0.92	0.92	170

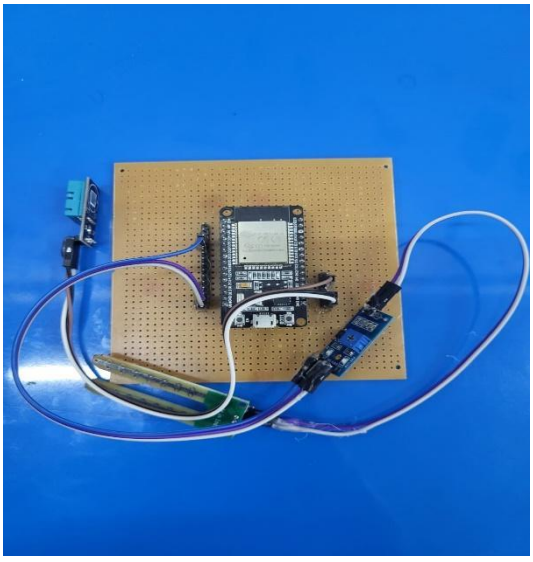

Fig 7 : Implementation Hardware Output

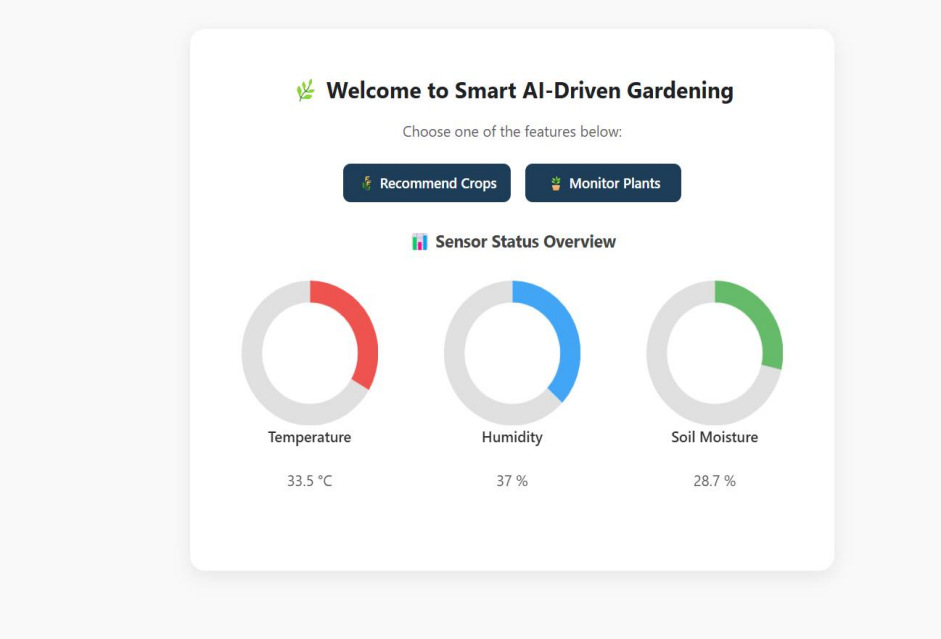

Fig. 8: Smart Gardening System Dashboard showing real-time sensor status and navigation options for crop recommendation and plant monitoring.

C. Confusion Matrix

A confusion matrix was also generated and visualized as a heatmap. It showed minimal misclassifications, with most predictions falling on the diagonal, indicating correct class predictions.

D. Sample Output – Crop Recommendation

When the user initiates the /start_recommend endpoint, 12 sensor readings are captured over 24 seconds. Below is a sample output response:

{

"logs": [...],

"crops": ["Basil", "Spinach", "Tomato", "Rose", "Hibiscus"],

"probs": [0.83, 0.72, 0.69, 0.67, 0.65]

}

This output presents the top 5 crops that best match the current environmental conditions, with their prediction confidences.

E. Real-Time Monitoring and Alert Generation

The /monitor_plants feature allows users to select plants and continuously monitor whether current environmental readings fall within the defined optimal range.

Example Scenario:

Current Conditions: Temperature = 35°C, Moisture = 22%

Selected Plant: Rose (optimal temp: 18–30°C, moisture ≥ 40%)

Generated Alerts:

🌡 Too hot for Rose

💧 Too dry for Rose

These alerts help users take timely action, such as watering or moving the plant to a shaded location.

F. User Interface Snapshots (describe for paper visuals)

Homepage: Clean UI with navigation to recommendation and monitoring pages.

Recommend Page: Button to start data collection and prediction.

Monitor Page: Dropdown menu to select plants and view alerts.

(Diagram suggestion: Screenshots of each page can be included in the final paper or appendix.)

G. Deployment Success

The system was deployed successfully on **Render**, demonstrating:

Stable connection with MQTT broker (broker.emqx.io)

Real-time streaming of sensor values

Fast response (within 1–2 seconds) for crop prediction and monitoring

Users accessed the web app via browser on both desktop and mobile platforms, verifying responsiveness and cross-device compatibility.

5. CONCLUSION

The Smart Gardening System presented in this paper successfully demonstrates the integration of IoT and machine learning technologies to create a responsive, intelligent, and user-friendly gardening assistant. By leveraging real-time environmental data gathered from ESP32-connected sensors, the system enables automated crop recommendations and continuous plant condition monitoring. The use of the MQTT protocol ensures low-latency and reliable communication between sensor nodes and the cloud-based Flask server. Additionally, the system's machine learning model, trained on a robust dataset with 12 sequential sensor readings, enables high-accuracy crop predictions tailored to changing environmental conditions.

The web-based interface provides ease of use for both experienced and novice users, allowing them to select plants to monitor, receive alerts, and initiate crop prediction with minimal interaction. The deployment on Render ensures remote access, scalability, and platform independence. Throughout testing, the system proved effective in suggesting suitable crops and detecting unfavorable environmental conditions such as excess heat or low moisture, helping users take timely corrective actions.

Overall, this project contributes a scalable, affordable, and intelligent solution to the challenges of urban gardening and small-scale agriculture, enabling users to optimize plant health and resource usage through real-time analytics.

6. FUTURE SCOPE

While the current implementation is functional and effective, several enhancements can further expand the capabilities and impact of the system:

1. **Automation of Irrigation** Integrate water pumps and relay modules with ESP32 to automatically water plants based on real-time soil moisture readings. This would enable a fully closed-loop smart irrigation system.

2. **Expansion of Sensor Suite** Add additional sensors such as:

Light intensity sensors (for sun-loving plants)

Rain sensors

Air quality sensors

pH sensors for soil acidity

3. **Weather API Integration** Integrate live weather forecasts using APIs (like OpenWeather) to make proactive recommendations based on upcoming temperature or rainfall conditions.

4. **Mobile Application** Develop a dedicated cross-platform mobile app that combines monitoring, alerting, and AI recommendations in a single interface with push notifications.

5. **Dynamic Learning System** Allows the system to collect historical garden-specific data and retrain itself periodically, leading to a personalized AI model for each user over time.

6. **Voice Assistant Integration** Integrate with voice assistants like Alexa or Google Assistant to provide verbal alerts and crop suggestions.

7. **Offline Mode with Local Storage** Enable offline logging of sensor data in case of internet outages and allow later synchronization with the cloud.

8. **Multi-Zone Garden Support** Allows users to manage multiple sensor zones (e.g., front yard, backyard) and get zone-specific recommendations and alerts.

By implementing these future upgrades, the Smart Gardening System can evolve into a more comprehensive, autonomous, and personalized platform—benefiting both individual home gardeners and community-based urban farming initiatives.

REFERENCES

1. C. Smith, "AI in Domestic Gardening: A Review," International Journal of Smart Agriculture, vol. 5, no. 3, pp. 45–57, 2023.
2. D. Johnson and E. Williams, "Sensor Integration in Smart Gardens," IEEE Sensors Journal, vol. 12, no. 8, pp. 1234–1240, 2022.
3. G. Patel and H. Desai, "Optimizing Irrigation Using Soil Moisture Sensors," IEEE Transactions on Automation Science and Engineering, vol. 9, no. 4, pp. 567–573, 2020.
4. I. Khan, "Real-time Data Analytics in Agriculture," Computers and Electronics in Agriculture, vol. 50, pp. 12–25, 2019.
5. W. Zhao, "Hydroponics and AI Integration," Journal of Controlled Environment Agriculture, vol. 4, pp. 45–55, 2022.
6. X. Li, "Vertical Gardening and Automation Techniques," International Journal of Urban Horticulture, vol. 9, pp. 98–106, 2021.
7. R. Kumar, "Microclimatic Analysis for Urban Gardening," Journal of Agricultural Science and Technology, vol. 12, no. 3, pp. 134–145, 2021.
8. L. Chen and M. Zhang, "AI-Driven Solutions for Sustainable Urban Agriculture," Sustainable Computing: Informatics and Systems, vol. 8, pp. 210–221, 2021.
9. N. Brown, "Temperature Sensors for Smart Gardens: A Comparative Study," IEEE Transactions on Instrumentation and Measurement, vol. 68, no. 7, pp. 2341–2349, 2020.
10. Arduino IDE Documentation – https://www.arduino.cc/en/software
11. EMQX MQTT Broker – https://www.emqx.io
12. scikit-learn Machine Learning Library – https://scikit-learn.org
13. Flask Documentation – https://flask.palletsprojects.com
14. Render Cloud Hosting – https://render.com