

An IoT Technology as Smart Water Management System for Family Medicinal Plants

Muhammad Akmal Ar Rasid¹, Dendy K Pramudito², Amali³

^{1,2,3}Informatics Engineering Department, Universitas Pelita Bangsa

papers.upb@gmail.com

Abstract

Watering of Family Medicinal Plants (FMP) located in government institutions should not follow a regular interval. As they require adequate water to grow, over-watering can lead to wastage of water and can decrease efficiency in plant maintenance. The plants are typically watered by employees manually based on a schedule or by checking up on the plants. But, at times soil moisture can vary at any hour of the day and irrigation might not happen if a person is not available to check. There are chances of overwatering or underwatering the plants. In this paper we propose a system which uses Internet of Things to monitor the soil moisture level and make irrigation automated. NodeMCU ESP8,266 acts as the main controller of the system. A soil moisture sensor is used to sense the amount of water present in the medium where the plants are growing. A relay module is used to control the pump, a water pump is used to water the plants and finally an I2C LCD is used to display the real time information. We also use IoT Watering platform to display the monitoring data and control the system over internet. We performed field study, literature survey, requirement study, system design (hardware and software), implementation and testing of the prototype. The results showed that the system sensed the soil moisture levels and send the data to monitoring platform and display the data in real time. The water pump is used to water the plants automatically. If the moisture level falls below the threshold which is set in the system, the pump will switch on. If the moisture level is enough for the plants to survive, then the pump will switch off. This system allows us to have better control over watering of plants, measure the efficiency and make the process faster.

Keywords: Internet of Things, Soil Moisture, Automatic Watering, Nodemcu ESP8266, Water Efficiency.

INFEK is licensed under a Creative Commons 4.0 International License.



1. Introduction

Family Medicinal Plants (FMP) are plants cultivated for simple herbal use and health support. In government institutions, FMP can be planted in yards, green areas, or demonstration gardens. The maintenance of these plants requires sufficient water because soil moisture affects root condition, nutrient absorption, and plant growth. Previous studies on FMP cultivation, IoT-based plant monitoring, and smart irrigation show that stable soil moisture is important for maintaining plant quality and avoiding drought stress or excess water [1] [2] [10].

In the observed environment, watering is commonly performed manually by maintenance staff. The staff checks the plant condition visually or by simple experience, then waters the plants according to a routine schedule. This practice is easy to perform, but it does not provide accurate soil moisture information. Manual watering may waste water when the soil is already moist and may delay watering when the soil becomes dry outside the routine schedule. Research on smart water management, IoT irrigation, and irrigation water-use efficiency indicates that routine-based watering should be improved through measured and data-supported decisions [3] [9] [11].

Real-time monitoring is important because watering decisions should be based on current soil condition. A soil moisture sensor can convert the water content of

the soil into a measurable value. When this value is connected to a microcontroller and a web dashboard, users can monitor the planting media from a distance. Wireless sensor networks, energy-efficient sensing nodes, and in-field IoT systems have been widely used to support agricultural monitoring and precision irrigation decisions [12] [14] [18].

Internet of Things (IoT) technology provides the ability to connect physical devices to digital platforms. In this research, a capacitive soil moisture sensor is connected to NodeMCU ESP8266. The microcontroller reads the sensor value, sends the data through Wi-Fi, and controls a relay connected to a 5V water pump. The web dashboard displays moisture information, provides pump control, and stores schedule configuration. Similar IoT agriculture studies show that sensor integration, data communication, and dashboard-based control can support smart farming implementation and reduce dependence on manual inspection [4] [5] [6]. The smart park monitoring and control system using NodeMCU ESP8266 by Gunawan, Anshor, and Amali also supports the relevance of using ESP8266 for real-time environmental monitoring and remote control in park maintenance [21] [22] [23].

Previous studies also show that IoT-based irrigation systems can combine soil moisture sensors, wireless communication, cloud dashboards, data analytics, and automatic controllers to improve monitoring accuracy and water-use efficiency. Reviews on Agriculture 4.0,

arable farming IoT, smart irrigation energy management, data-driven precision agriculture, and precision irrigation technology emphasize that irrigation control becomes more effective when sensor data and decision rules are integrated in one system [7] [8] [13] [15] [16].

Based on these conditions, this paper presents a thesis-based implementation of an IoT-based soil moisture monitoring system for efficient FMP watering in a government institution. The research problem is focused on two questions: how to handle real-time soil moisture monitoring to optimize plant water needs, and how the system can reduce water waste and improve the efficiency of water use. The contribution of this study is the implementation of a prototype and web-based dashboard that connects monitoring, scheduling, and control functions, which is aligned with recent studies on advanced IoT sensing, precision irrigation management, IoT-driven water efficiency, and smart irrigation concepts [17] [18] [19] [20].

2. Research Method

The study used the Waterfall development method because its stages are sequential and suitable for a system that requires clear requirement analysis before implementation. The stages consist of requirement analysis, system design, implementation, testing, and evaluation. At the analysis stage, the researcher identified the current watering process, the problems of manual monitoring, and the features required by users. At the design stage, the block diagram, flowchart, hardware arrangement, and application interface were prepared. Implementation was carried out by building the prototype and web application. Testing was then conducted to observe the relation between sensor reading, dashboard display, manual command, and schedule control.

Data were collected through observation and literature study. Observation was carried out in the Ministry of Home Affairs environment, especially related to the maintenance of FMP plants and the manual watering activity performed by gardeners. Literature study was used to understand smart irrigation, IoT-based monitoring, sensor-based watering, and the use of microcontrollers in agriculture. These data sources became the basis for defining system requirements and validating the relevance of the proposed system.

The user requirements include easy access to soil moisture information, real-time dashboard display through smartphone or computer, automatic watering support, and warning or status information when moisture is below the threshold. The hardware requirements include NodeMCU ESP8266, capacitive soil moisture sensor, LCD I2C 16x2, relay module, jumper cables, 5V pump, hose, and a device for accessing the dashboard. The software requirements include Arduino IDE for programming, IoT Watering platform for monitoring and control, and Fritzing for circuit design documentation. Next Block diagram of the IoT watering system on Figure 1.

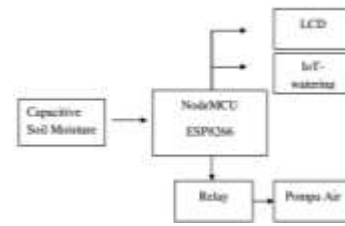


Figure 1. Block Diagram of the IoT Watering System

Figure 1 presents the block diagram of the proposed system. The capacitive soil moisture sensor functions as the input unit because it measures water content in the soil. NodeMCU ESP8266 acts as the main processing unit that reads sensor data, executes the programmed logic, and connects the device to the internet using its internal Wi-Fi capability. The LCD I2C is used for local display, while the IoT Watering dashboard is used for remote monitoring and control. The relay and pump become the output unit that realizes the watering command.

The system architecture separates several responsibilities. The sensor layer collects soil condition data. The controller layer processes the data and decides whether the pump should be activated. The interface layer presents information to the user through the dashboard and LCD. The actuator layer performs physical watering through the relay and pump. This separation makes troubleshooting easier because each part can be checked independently. In automatic mode, the system compares the moisture value with a threshold. If the value is lower than the threshold, the soil is considered dry. However, the pump is not immediately activated unless the active schedule also permits watering. This means the schedule is a time boundary, while the threshold is a soil-condition boundary. The combination prevents unnecessary watering outside the planned maintenance time. Next Device workflow and watering decision process on figure 2.

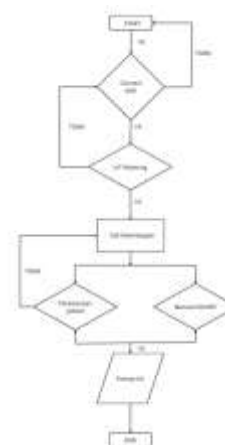


Figure 2. Device Workflow and Watering Decision Process

Figure 2 shows the device workflow. The process starts when the microcontroller is powered on. The device attempts to connect to Wi-Fi. If the connection fails, the device repeats the connection process until communication is available. After connection, the

ESP8266 communicates with the IoT Watering platform, reads the soil moisture sensor, and sends the measured value to the dashboard. The system then checks the watering control logic.

The watering decision has two control alternatives. In manual control, the user can press the pump ON or OFF button from the dashboard. This mode is useful for testing, maintenance, or special watering needs. In automatic control, the system uses the moisture threshold and schedule configuration. The pump is activated only when soil moisture is below the threshold and the current time is within the active schedule. After watering, the pump is stopped to avoid excess water use.

The prototype arrangement is shown in Figure 3. The soil moisture sensor is connected to the ESP8266 analog input to read the moisture value. The relay is connected to a digital output pin and works as an electronic switch for the pump. The LCD I2C receives data from the microcontroller and displays local information, while the web dashboard provides remote information and control. This arrangement demonstrates that the physical prototype and the application are integrated into a single IoT watering system. Next Prototype Circuit Arrangement on Figure 3.

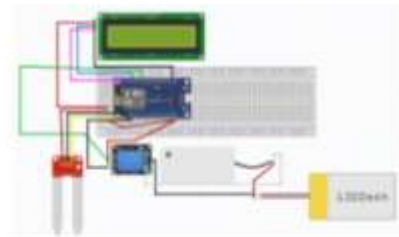


Figure 3. Prototype Circuit Arrangement

3. Result and Discussion

The implementation resulted in a web-based IoT Watering application and a prototype device. The application includes login, registration, password recovery, dashboard monitoring, manual pump control, and watering schedule management. The prototype includes ESP8266, capacitive soil moisture sensor, relay, LCD I2C, and 5V water pump. The relationship between both parts is essential because the dashboard receives data from the device and the device receives configuration from the dashboard.

Authentication is provided through login and registration pages. The registration page records a new user's identity, username, email, password, and password confirmation. The login page verifies the account before allowing access to the dashboard. Password recovery provides a reset mechanism when users forget their password. These features are important because the dashboard contains pump control and schedule configuration, so unauthorized access could result in incorrect watering operation.

The authentication interface shown in Figure 4 also supports usability. The login page clearly provides

username or email input, password input, login button, forgot password link, and registration link. The registration page provides complete account fields. The password recovery page provides a simple input for email and a reset link request button. These pages make the system easier to operate by users who are not familiar with IoT devices.



Figure 4. Login, Registration, and Password Recovery Interfaces

The dashboard interface shown in Figure 5 is the main page for monitoring and control. It displays soil moisture value, last update information, soil status, system mode, threshold value, pump duration, and manual control buttons. The dashboard also contains a schedule summary so users can understand whether watering has been configured for specific days and times. The dashboard refresh interval supports near real-time monitoring so the user can observe the latest system condition.

The manual control buttons consist of POMPA ON and POMPA OFF. When POMPA ON is pressed, the system sends a command that can be read by the device and converted into a relay signal. When POMPA OFF is pressed, the system sends a command to stop the pump. This feature is necessary because maintenance officers may need direct control during device inspection or when the automatic rule needs to be overridden temporarily.

The threshold value is used as the boundary between dry and sufficient soil condition. When the sensor reading is below the threshold, the system interprets that the soil needs water. When the value is above the threshold, watering is not required. This logic prevents watering based only on fixed routines and encourages watering based on actual soil condition. Therefore, threshold configuration becomes one of the key elements for water efficiency. Next IoT Watering Dashboard Interface on Figure 5.



Figure 5. IoT Watering Dashboard Interface

The schedule configuration page shown in Figure 6 allows users to set the watering day, start time, end time, and active status. The user can update a schedule, reset form input, or create a default schedule for all days. Existing schedules are displayed in a table with actions such as edit, disable, and delete. This schedule feature is useful for institutional maintenance because watering can be limited to working hours or selected daily periods.

For example, if the schedule is active from 06:00 to 18:00, the pump is allowed to operate only during that period. If the soil becomes dry at 10:00, the pump can activate because the condition is dry and the time is allowed. If the soil becomes dry at 22:00, the pump remains inactive because the time is outside the active schedule. This mechanism prevents the pump from working continuously and reduces the risk of unnecessary water consumption.

The functional testing was described through five scenarios. First, when the sensor reads soil moisture, the value should appear on the dashboard. Second, when automatic mode is active and the soil is below the threshold within the active schedule, the pump should turn on. Third, when manual mode is used, the dashboard ON and OFF buttons should directly control the pump. Fourth, when schedule data are updated, the new values should be stored and displayed. Fifth, when password recovery is requested using registered email, the system should provide the reset process. Next Watering Schedule Configuration Page on Figure 6.



Figure 6. Watering Schedule Configuration Page

The system was tested using four main test groups: soil

moisture sensor reading, ESP8266 data delivery, water pump response, and comparison between the previous manual watering process and the proposed IoT-based system. These tests were used to confirm that the prototype can read soil condition, send monitoring data, control the actuator, and support more efficient watering decisions.

Table 1. Soil Moisture Sensor Test Results

No.	Soil Condition	Sensor Value (%)	Category
1	Very dry	18	Dry
2	Very dry	22	Dry
3	Dry	30	Dry
4	Dry	38	Dry
5	Slightly damp	47	Medium
6	Slightly damp	53	Medium
7	Humid	65	Humid
8	Humid	71	Humid
9	Very humid	82	Very humid
10	Very humid	88	Very humid

Table 1 shows that the sensor reading changes according to the soil condition. Very dry soil produces low moisture values, while humid and very humid soil produces higher values. This result indicates that the soil moisture sensor can be used as the main input for determining whether the plant media requires watering. Next Data Delivery Test Results on Table 2.

Table 2. Data Delivery Test Results

No.	Test	Connection Status	Data Delivery
1	Testing 1	Connected	Successful
2	Testing 2	Connected	Successful
3	Testing 3	Connected	Successful
4	Testing 4	Connected	Successful
5	Testing 5	Connected	Successful

Table 2 explains the data delivery test between NodeMCU ESP8266 and the IoT Watering dashboard. The test was conducted five times, and each trial showed a connected status with successful data delivery. This result demonstrates that the device can transmit soil moisture data through the internet connection so that users can monitor the system remotely. Next Water Pump Test Results on Table 3.

Table 3. Water Pump Test Results

Moisture Value (%)	Pump Status
20	ON
28	ON
35	ON
42	OFF
50	OFF
65	OFF
78	OFF

Table 3 presents the water pump response based on several moisture values. The pump turns on when the moisture value is below the defined threshold and turns off when the soil condition is sufficient. This result confirms that the relay and pump control can follow the automatic watering logic designed in the system. Next Comparison of Manual Systems and IoT Systems on Table 4.

Table 4. Comparison of Manual Systems and IoT Systems

Aspect	Manual System	IoT System
Humidity monitoring	Visual inspection	Real-time sensor reading
Watering decision	Based on estimation	Based on data
Water usage	Unmeasured	Measurable
Remote monitoring	Not available	Available
Time efficiency	Low	High
Water efficiency	Low	High

Table 4 compares the previous manual watering system and the proposed IoT-based system. The manual system depends on visual inspection and officer estimation, while the IoT system uses sensor data, real-time monitoring, and remote access. Therefore, the proposed system provides better support for time efficiency, water efficiency, and more objective watering decisions. The discussion shows that the system's efficiency is produced by combining sensor-based measurement, threshold rules, and schedule restrictions. The sensor provides actual soil condition, the threshold determines whether watering is needed, and the schedule determines when watering is allowed. Without the sensor, watering would depend only on routine time. Without the schedule, the pump could operate whenever the soil is dry, including at inappropriate times. Therefore, the three elements must work together.

From the perspective of institutional maintenance, the system can reduce dependence on manual inspection. Officers do not need to repeatedly check the soil by hand because the dashboard provides the latest moisture information. The system also provides control from a web interface, so the pump can be managed from a smartphone or computer. This supports faster decision making and can make the watering process more accountable because the configuration and monitoring information are displayed in one place.

Security is also part of the discussion because the system has the ability to activate a physical pump. Login and account management ensure that only registered users can enter the dashboard. Password strength, password recovery, and periodic account review should be considered in the maintenance plan. If the system is used by a government institution, role management may be added in future development to separate administrator access and ordinary operator access.

Hardware maintenance is required to ensure stable operation. The soil moisture sensor should be cleaned periodically because soil residue and corrosion can affect reading accuracy. The ESP8266 must receive stable power and Wi-Fi signal. The relay wiring should be inspected to ensure safe switching, and the pump hose should be checked for blockage or leakage. Software maintenance includes backing up sensor and schedule data, reviewing error logs, and updating code when features need improvement.

The system still has limitations. It focuses on soil moisture and pump control, but it does not measure

other conditions such as soil pH, nutrient content, rainfall, temperature, or weather prediction. It also depends on Wi-Fi connection for real-time dashboard update. When the connection is interrupted, the dashboard may not show the latest condition. Future research can add offline control logic, notification through WhatsApp or email, water-flow measurement, and historical reports to quantify actual water savings.

For deployment in a government institution, the system should be installed in a protected box to reduce exposure to rain, direct sunlight, and accidental cable movement. The sensor probe should be placed near the plant root zone because the purpose of monitoring is to represent the actual moisture condition experienced by the plant. If the sensor is placed too far from the root area or too close to the water outlet, the reading may not represent the general soil condition.

Sensor calibration should be carried out before routine operation. Calibration can be performed by observing the sensor value in dry soil and wet soil, then defining an appropriate conversion range or threshold. This step is important because different soil textures may produce different readings. Sandy soil, clay soil, and compost-rich media can retain water differently. Therefore, the threshold should not be copied without considering the planting media used in the FMP area.

Operational procedures should also be prepared for users. The operator should log in, check the dashboard status, confirm that the device is connected, review the latest moisture value, and verify the active schedule. If manual watering is needed, the operator can use the pump ON button and then stop it using the pump OFF button. If automatic watering is preferred, the operator should ensure that the mode, threshold, and schedule have been configured correctly.

The system can be evaluated by comparing the watering pattern before and after the implementation. In manual watering, the amount of water is usually determined by habit or estimation. In the proposed system, watering is triggered by soil condition and limited by schedule. Although this paper does not present a measured water-volume comparison, the logic of the system supports water saving because unnecessary watering events can be reduced. Future testing can add a water-flow sensor to calculate exact water consumption.

The database and application should be maintained regularly. User accounts, sensor records, and schedule data should be checked to avoid duplication or outdated configuration. Backup is necessary when the system is used for long-term monitoring because historical data can be used to evaluate plant maintenance patterns. Error logs should also be reviewed when the dashboard fails to receive new data or the pump command is not executed.

From the sustainability perspective, this system supports the efficient use of resources by using data as the basis of watering decisions. The system does not only automate the pump, but also provides information

that helps staff understand the condition of the planting media. This is relevant for public institutions because resource efficiency, cost awareness, and environmental responsibility are part of daily operational concerns.

4. Conclusion

The developed IoT-based soil moisture monitoring system can support real-time observation of soil conditions by reading data from a capacitive soil moisture sensor, processing it through NodeMCU ESP8266, and presenting the result on a web dashboard. This allows users to understand the current planting condition and perform watering based on actual moisture needs rather than manual estimation. The system can also reduce water waste by combining threshold-based automatic watering with a configurable watering schedule. The pump operates only when the soil is dry and the active schedule permits watering, so the watering process becomes more controlled, measurable, and suitable for institutional plant maintenance. Future development should include notification features, historical water-use reports, more durable sensors, and additional environmental parameters to improve reliability and evaluation.

Acknowledgements

The author thanks the supervisors, the Informatics Study Program of Universitas Pelita Bangsa, and the Ministry of Home Affairs environment that supported observation and data collection for this research.

References

[1] Aisyah, N., Ulhaq, E. D., Dharmawan, A., & Purbakawaca, R. (2025). Design of an IoT-Based Smart Irrigation System Using Soil Moisture Sensors for Water Efficiency. *Journal Online of Physics*, 11(1), 89–97. DOI: <https://doi.org/10.22437/jop.v11i1.48928> .

[2] García, L., Parra, L., Jimenez, J. M., Lloret, J., & Lorenz, P. (2020). IoT-Based Smart Irrigation Systems: An Overview on the Recent Trends on Sensors and IoT Systems for Irrigation in Precision Agriculture. *Sensors*, 20(4), 1042. DOI: <https://doi.org/10.3390/s20041042> .

[3] Kamiński, C., et al. (2019). Smart Water Management Platform: IoT-Based Precision Irrigation for Agriculture. *Sensors*, 19(2), 276. DOI: <https://doi.org/10.3390/s19020276> .

[4] Muangprathub, J., Boonnam, N., Kajornkasirat, S., Lekbangpong, N., Wanichsombat, A., & Nillaor, P. (2019). IoT and Agriculture Data Analysis for Smart Farm. *Computers and Electronics in Agriculture*, 156, 467–474. DOI: <https://doi.org/10.1016/j.compag.2018.12.011> .

[5] Ayaz, M., Ammad-Uddin, M., Sharif, Z., Mansour, A., & Aggoune, E.-H. M. (2019). Internet-of-Things (IoT) Based Smart Agriculture: Towards Making the Fields Talk. *IEEE Access*, 7, 129551–129583. DOI: <https://doi.org/10.1109/ACCESS.2019.2932609> .

[6] Farooq, M. S., Riaz, S., Abid, A., Abid, K., & Naeem, M. A. (2019). A Survey on the Role of IoT in Agriculture for the Implementation of Smart Farming. *IEEE Access*, 7, 156237–156271. DOI: <https://doi.org/10.1109/ACCESS.2019.2949703> .

[7] Liu, Y., Ma, X., Shu, L., Hancke, G. P., & Abu-Mahfouz, A. M. (2021). From Industry 4.0 to Agriculture 4.0: Current Status, Enabling Technologies, and Research Challenges. *IEEE Transactions on Industrial Informatics*, 17(6), 4322–4334. DOI: <https://doi.org/10.1109/TII.2020.3003910> .

[8] Villa-Henriksen, A., Edwards, G. T. C., Pesonen, L. A., Green, O., & Sørensen, C. A. G. (2020). Internet of Things in Arable Farming: Implementation, Applications, Challenges and Potential. *Biosystems Engineering*, 191, 60–84. DOI: <https://doi.org/10.1016/j.biosystemseng.2019.12.013> .

[9] Obaideen, K. (2022). An Overview of Smart Irrigation Systems Using IoT. *Energy Nexus*, 7, 100124. DOI: <https://doi.org/10.1016/j.nexus.2022.100124> .

[10] Vallejo-Gómez, D., Osorio, M., & Hincapié, C. A. (2023). Smart Irrigation Systems in Agriculture: A Systematic Review. *Agronomy*, 13(2), 342. DOI: <https://doi.org/10.3390/agronomy13020342> .

[11] Koeh, R., & Langat, P. (2018). Improving Irrigation Water Use Efficiency: A Review of Advances, Challenges and Opportunities in the Australian Context. *Water*, 10(12), 1771. DOI: <https://doi.org/10.3390/w10121771> .

[12] Koeh, R., & Langat, P. (2018). Improving Irrigation Water Use Efficiency: A Review of Advances, Challenges and Opportunities in the Australian Context. *Water*, 10(12), 1771. DOI: <https://doi.org/10.3390/w10121771> .

[13] Nasiakou, A., Vavalis, M., & Zimeris, D. (2016). Smart Energy for Smart Irrigation. *Computers and Electronics in Agriculture*, 129,74-83. DOI: <https://doi.org/10.1016/j.compag.2016.09.008> .

[14] Jawad, H., Nordin, R., Gharghan, S., Jawad, A., & Ismail, M. (2017). Energy-Efficient Wireless Sensor Networks for Precision Agriculture: A Review. *Sensors*, 17(8), 1781. DOI: <https://doi.org/10.3390/s17081781> .

[15] Tantalaki, N., Souravlas, S., & Roumeliotis, M. (2019). Data-Driven Decision Making in Precision Agriculture: The Rise of Big Data in Agricultural Systems. *Journal of Agricultural & Food Information*, 20(4), 344–380. DOI: <https://doi.org/10.1080/10496505.2019.1638264> .

[16] Lakhari, I. A., et al. (2024). A Review of Precision Irrigation Water-Saving Technology Under Changing Climate for Enhancing Water Use Efficiency, Crop Yield, and Environmental Footprints. *Agriculture*, 14(7), 1141. DOI: <https://doi.org/10.3390/agriculture14071141> .

[17] Abdelmoneim, A. A., Kimaita, H. N., Al Kalaany, C. M., Derardja, B., Dragonetti, G., & Khadra, R. (2025). IoT Sensing for Advanced Irrigation Management: A Systematic Review of Trends, Challenges, and Future Prospects. *Sensors*, 25(7), 2291. DOI: <https://doi.org/10.3390/s25072291> .

[18] Dong, Y., Werling, B., Cao, Z., & Li, G. (2024). Implementation of an In-Field IoT System for Precision Irrigation Management. *Frontiers in Water*, 6. DOI: <https://doi.org/10.3389/frwa.2024.1353597> .

[19] Mohamed, Z. E., Afify, M. K., Badr, M. M., & Omar, O. A. (2026). IoT-Driven Smart Irrigation System to Improve Water Use Efficiency. *Scientific Reports*, 16(1). DOI: <https://doi.org/10.1038/s41598-025-33826-6> .

[20] Morchid, A., Khalid, H. M., El Alami, R., Sitaramanan, S. R., & Muyeen, S. M. (2024). High-Technology Agriculture System to Enhance Food Security: A Concept of Smart Irrigation System Using Internet of Things and Cloud Computing. *Journal of the Saudi Society of Agricultural Sciences*. DOI: <https://doi.org/10.1016/j.jssas.2024.02.001> .

[21] Gunawan, S., Anshor, A. H., & Amali, A. (2023). Sistem Monitoring dan Kontrol Taman Pintar Berbasis IoT (Internet of Things) dengan NodeMCU ESP8266. *Bulletin of Computer Science Research*, 3(4), 283–288. DOI: <https://doi.org/10.47065/bulletincsr.v3i4.270> .

[22] Kasendra, M., Pramudito, D. K., & Afandi, D. (2025). The Soil Moisture Monitoring Utilize IoT System to Improve Urban Farming Productivity and Enhance Food Security. *Journal of Information and Digital Technology*, 1–12. DOI: <https://doi.org/10.60083/jidt.vi0.609> .

[23] Agtyaputra, I. M., Raharjo, T., & Pujiono, I. (2024). Influencing

