International Journal of Scientific Engineering and Research (IJSER)
ISSN (Online): 2347-3878
SJIF (2024): 6.623

# Smart Irrigation System for Precision Farming

**Dr. P. Ganesh Kumar[1], Santhos K S[2], Suriya N[3], Nithesh M[4], Vimal Adithan K[5]**

[1]Department of Information Technology, K.L.N. College of Engineering, Sivagangai, India
Email: *itz.ganeshkumar[at]gmail.com*

[2]Department of Information Technology, K.L.N. College of Engineering, Sivagangai, India
Email: *kssanthos2004[at]gmail.com*

[3]Department of Information Technology, K.L.N. College of Engineering, Sivagangai, India
Email: *suriyanagalingam13[at]gmail.com*

[4]Department of Information Technology, K.L.N. College of Engineering, Sivagangai, India
Email: *nitheshkrishna2003[at]gamil.com*

[5]Department of Information Technology, K.L.N. College of Engineering, Sivagangai, India
Email: *vmlbtechit[at]gmail.com*

**Abstract:** ***This project proposes two innovative digital solutions addressing challenges in agriculture and recreational infrastructure. First, a Smart Irrigation System for Precision Farming utilizes IoT sensors, data analytics, and a Power BI dashboard to monitor soil conditions and optimize water usage. It enables real-time decisions for irrigation, fertilization, and crop health monitoring, thus improving resource efficiency, reducing environmental impact, and increasing crop yield. The system is modular, cost-effective, and suitable for small-scale farmers, supporting sustainable agricultural practices. Secondly, the Intelligent Field Reservation System is a mobile application designed for real-time booking of sports facilities. It includes user authentication, location-based search, availability tracking, booking history, and secure payment gateways. The admin interface allows for schedule management and maintenance tracking, enhancing overall facility management. Both systems exemplify SDG 9: Industry, Innovation, and Infrastructure, by integrating technology into everyday challenges. These projects satisfy Program Outcomes (PO 1–8) and Program Specific Outcomes (PSO 1 & 2), demonstrating both technical competency and societal impact.***

**Keywords:** Precision Farming, Smart Irrigation, IoT Sensors, Power BI, Mobile Application, Real-Time Booking, Sports Facility Booking, Field Reservation System, User Authentication, Availability Tracking

## 1. Introduction

Agriculture is the backbone of the Indian economy, employing more than half of the country's population. However, the sector still relies heavily on traditional irrigation methods that lead to excessive water usage, uneven crop growth, and reduced productivity. The advancement of IoT (Internet of Things), data analytics, and automation has paved the way for Smart Irrigation Systems that can revolutionize traditional farming practices.

The proposed system, Smart Irrigation System for Precision Farming, integrates IoT sensors, cloud analytics, and Power BI dashboards to monitor environmental conditions such as soil moisture, humidity, and temperature in real time. By automating irrigation decisions based on live data, the system ensures optimal water usage, improved crop yield, and sustainable resource management.

## 2. Problem Statement

Traditional irrigation systems in India often operate without considering real-time soil and weather conditions. As a result Farmers face water scarcity and crop damage due to over-irrigation or under-irrigation. Lack of real-time monitoring tools leads to inefficient water and fertilizer management. Manual methods are time-consuming and not scalable for large farmlands. There is a need for a cost-effective, multilingual, and user-friendly system that helps farmers make data-driven decisions regarding irrigation and crop management. Therefore, a Smart Irrigation System is essential to address these challenges and promote sustainable agriculture.

## 3. Problem Objectives

The main objectives of the Smart Irrigation System for Precision Farming are: To automate irrigation using IoT sensors and Raspberry Pi based on real-time soil and weather data. To enable farmers to monitor and control irrigation remotely using a web or mobile application. To develop interactive Power BI dashboards for real-time visualization of farm conditions. Support multi-language interfaces (Tamil, Hindi, English) to make the system accessible to all farmers. Promote efficient water management and sustainable agriculture practices through data analytics. Reduce human intervention and operational costs while increasing agricultural productivity.

## 4. Scope of the Project

***a) Exsisting System***

The existing irrigation system in many Indian farmlands is manual or time-based, which leads to: Excessive use of water resources and energy. Lack of automation and real-time feedback. No integration with data analytics or remote monitoring. Difficulty for farmers to make informed irrigation decisions. Poor adaptability to changing climatic conditions.

***b) Proposed System***

The proposed Smart Irrigation System overcomes these limitations by Using IoT sensors (for soil moisture, humidity, temperature) to collect real-time data. Employing a Raspberry Pi as the central IoT hub for processing and controlling irrigation valves automatically. Connecting to the cloud for secure data storage and real-time analytics visualization via Power BI dashboards. Providing multi-language interfaces and mobile/web access for remote control. Enabling data-driven decision making, promoting resource optimization, and reducing environmental impact. This makes the system scalable, modular, and ideal for small and medium-scale farmers.

## 5. Implementation

**a) *Model Architecture***

Layered design: Sensor Layer, Processing Layer, Cloud Layer, and User Interface. Sensors collect soil moisture, temperature, and humidity data. Raspberry Pi/ESP32 processes data and controls irrigation. Cloud stores data and displays insights via Power BI dashboard. Farmers monitor and control irrigation through web or mobile app.

***b) Dataset Used***

Real-time data from IoT sensors (soil moisture, temperature, humidity). Data stored in ThingSpeak/Firebase/AWS IoT cloud. Used for analysis, visualization, and decision-making. Can integrate external datasets like weather forecast data.

***c) Code Flow***

Sensors initialize and collect environmental data. Data processed by Raspberry Pi using Python. Pump activates when soil moisture is below threshold. Data uploaded to cloud for visualization in Power BI. Farmer can view data or control irrigation manually via dashboard.

***d) Deployment***

Hardware installed in the field (sensors, Raspberry Pi, pump). Software hosted on cloud for analytics and control. Communication via Wi-Fi/GSM for real-time monitoring. Farmers can access and control the system remotely. Ensures efficient, automated, and sustainable irrigation.

## 6. Testing

***a) Testing Objectives***

The main objective of testing the Smart Irrigation System is to ensure that all components hardware, software, and communication modules work together accurately and reliably. It verifies that sensor readings, cloud updates, and irrigation control operations function as intended. Testing also aims to detect and fix bugs early, validate automation logic, and confirm that the system performs efficiently under various environmental conditions.

***b) Testing***

Testing was carried out throughout the development process to validate both hardware and software functionality. The sensors were tested for accurate data collection, and the microcontroller's response was verified for correct pump activation. Cloud communication and dashboard updates were checked for consistency. Different scenarios, such as low moisture or connection failure, were tested to ensure stable and reliable operation.

***c) Software Testing***

Software testing ensures that the Smart Irrigation System's application logic and data flow are correct. The Python code running on the Raspberry Pi and the cloud integration scripts were tested for errors and performance. The testing process included validating data transmission, dashboard visualization, and real-time response accuracy. Automated and manual tests were used to confirm that the software meets functional and non-functional requirements.

***d) Basic types of Testing***

1) Black Box Testing
2) White Box Testing
3) FunctionalTesting
4) Integration Testing
5) System Testing
6) Performance Testing
7) User Interface Testing
8) Unit testing
9) Verification and Validation Testing

***e) Testing techniques used***

*1) Unit Testing*

Unit testing focuses on verifying individual modules of the system. Each component sensor module, pump control logic, and cloud communication code was tested separately using simulated data. For example, soil moisture sensor output was checked for accuracy, and the pump control function was tested for correct ON/OFF response. This ensured that each unit performed correctly before system integration.

*Advantages*

- Easy to locate the bug.
- Exhaustive testing up to some extent.
- Interaction of multiple errors can be avoided.

*2) Integration Testing*

Integration testing was done to ensure all components of the Smart Irrigation System work together seamlessly. The main focus was on communication between the microcontroller, sensors, cloud platform, and mobile interface.

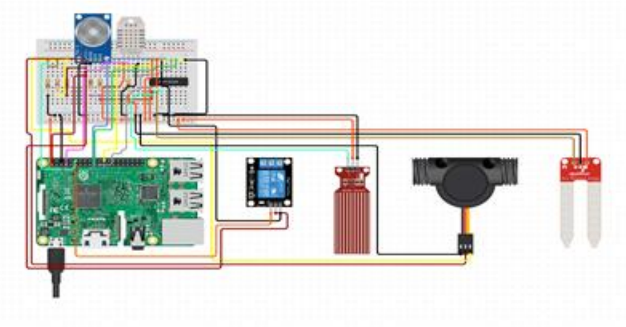
*3) Approach*

The bottom-up approach was used, starting with sensor data validation and gradually integrating other modules like the microcontroller, data processing code, and the IoT dashboard. Each integration step was verified for correct data transfer and response.

*4) Typical Tests*

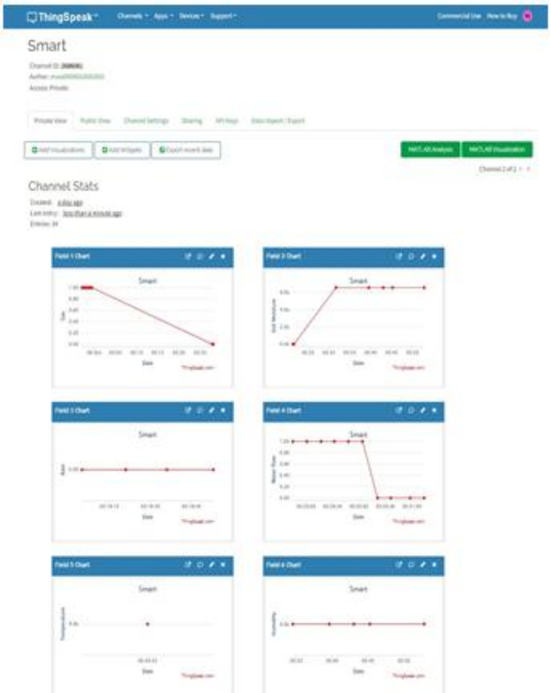
Tested sensor data transfer from ESP32/Raspberry Pi to the cloud (via Wi-Fi). Verified correct irrigation commands based on moisture levels. Checked cloud data synchronization with Power BI dashboard. Monitored system performance during continuous irrigation cycles. Ensured that alerts and notifications were triggered correctly on mobile devices.

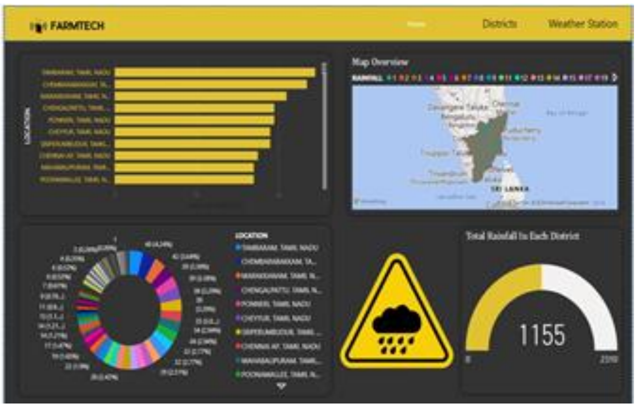
## 7. Screenshots

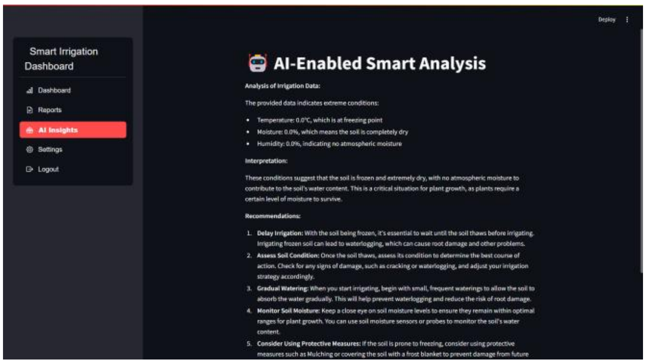


**Circuit diagram**

***Power bi dashboard***





**Smart Irrigation System diagram**

## 8. Literature Review

1) Leh et al. (2019) — “Smart Irrigation System Using Internet of Things” (IEEE ICSET) This paper presents a low-cost IoT-based irrigation system that utilizes sensors to measure soil moisture and temperature in real time. The collected data is processed by a microcontroller that controls water flow based on threshold levels.
2) Laksiri et al. (2019) — “Design and Optimization of IoT-Based Smart Irrigation System in Sri Lanka” (IEEE ICIIS) This research focuses on optimizing irrigation scheduling through IoT-based sensor networks to improve water utilization. The study demonstrates how soil moisture data can be used for dynamic irrigation decisions.
3) Shufian et al. (2019) — “Smart Irrigation System with Solar Power and GSM Technology” (IEEE ICAEE) The study integrates solar power and GSM communication for irrigation in remote areas without grid access. The GSM module sends messages to control irrigation pumps. Strengths: Sustainable and suitable for off-grid farms.
4) Karpagam et al. (2020) — “Smart Irrigation System Using IoT” (IEEE ICACCS) This research highlights the use of IoT sensors and cloud technology to automate irrigation based on soil moisture and temperature readings.
5) Kulkarni (2020) — “AgriTalk: IoT for Precision Soil Farming of Turmeric Cultivation” (IEEE IoT Journal) This study introduces a real-world IoT-based precision agriculture platform for turmeric cultivation. It uses multiple sensors to collect soil data and make decisions for irrigation and fertilization.

## 9. System Analysis

### A. *Functional requirement*

Collect real-time data from soil moisture, temperature, and humidity sensors. Automatically control water pumps based on sensor readings. Store and visualize data through Power BI dashboards. Provide remote monitoring and multilingual user access. Send alerts when soil moisture is too low or high.

### B. *Non-Functional Requirement*

System must be reliable, secure, and scalable. Real-time data updates with minimal delay. Easy to use for farmers with simple interface design. Accurate sensor readings and low maintenance.

### C. *Hardware Requirement*

1) Raspberry Pi (Controller)
2) Soil Moisture, DHT11 Sensors
3) Relay Module and Water Pump
4) Power Supply, Wi-Fi Module
5) Jumper Wires, Breadboard

### D. *Software Requirement*

1) Raspberry Pi OS, Python
2) Power BI for Data Visualization
3) Cloud Platform (ThingSpeak / Firebase)
4) HTML, CSS, JavaScript for Web Dashboard
5) SQL/NoSQL Database

### E. *Module Description*

1) **Sensor Module:** Collects real-time soil moisture, temperature, and humidity data using IoT sensors.
2) **Processing & Control Module:** Raspberry Pi/ESP32 processes sensor data and automatically controls the water pump through a relay.
3) **Cloud & Database Module:** Stores sensor data on cloud platforms like ThingSpeak or Firebase for analysis and remote access.
4) **Dashboard & Visualization Module:** Displays live sensor readings and irrigation status using Power BI dashboards for easy monitoring.
5) **User Interface Module:** Allows farmers to monitor and control irrigation via a multilingual web or mobile app.
6) **Communication Module:** Handles data transfer between sensors, controller, and cloud using Wi-Fi or GSM.
7) **Alert & Notification Module:** Sends alerts to farmers when soil moisture is low or system faults occur.
8) **Data Analytics Module:** Analyzes stored data to optimize waterusage and predict irrigation needs

## 10. Future Enhancement

1) **AI-Based Prediction:** Integrate machine learning models to predict optimal irrigation schedules based on soil type, crop growth stage, and weather conditions.
2) **Weather Forecast Integration:** Connect the system with real-time weather APIs to automatically adjust irrigation during rainfall or high humidity periods.
3) **Fertilizer Automation:** Extend the system to control fertilizer mixing and distribution using IoT-enabled nutrient sensors.
4) **Mobile Application Development:** Create a dedicated mobile app for farmers to monitor live data, control irrigation remotely, and receive instant alerts.
5) **Voice Command Control:** Implement voice assistant features in local languages for easy use by farmers with limited technical knowledge.
6) **Solar Energy Integration:** Use solar panels to power sensors and pumps, reducing dependency on electricity and promoting renewable energy use.
7) **Drone-Based Field Monitoring:** Add drone technology to capture aerial images, assess crop health, and detect irrigation coverage.
8) **Blockchain for Data Security:** Employ blockchain technology to securely record sensor data and ensure data transparency and reliability.
9) **Advanced Analytics Dashboard:** Enhance Power BI or cloud dashboards with predictive insights, charts, and historical trend analysis.

**Scalability for Large Farms:** Expand the system to manage multiple irrigation zones and large-scale agricultural fields with centralized monitoring and control.

## 11. Conclusion

The Smart Irrigation System for Precision Farming successfully demonstrates how IoT and automation can revolutionize traditional agricultural practices. By integrating soil moisture, temperature, and humidity sensors with microcontrollers and cloud platforms, the system enables real-time monitoring and automatic irrigation based on crop and soil conditions. This not only reduces human effort but also ensures optimal water usage, leading to increased crop yield and sustainability. The system's dashboard allows farmers to view real-time data, control irrigation remotely, and receive instant alerts. Overall, the project achieves its objective of promoting efficient resource utilization and supports the vision of smart and sustainable agriculture.

The implementation of this Smart Irrigation System proves that automation and IoT can play a vital role in solving agricultural challenges. It helps farmers make data-driven decisions rather than relying on manual observation. The system's ability to automatically control irrigation reduces water wastage and ensures crops get the right amount of moisture at the right time. This project highlights how technology can empower farmers to increase efficiency, save resources, and move toward sustainable farming practices.

**Acknowledgment**

The authors would like to thank the Department of Information Technology, K.L.N. College of Engineering, Sivaganga, for providing the necessary facilities, guidance, and support to carry out this project successfully. The authors also express their gratitude to the faculty members and technical staff of the department for their valuable suggestions and assistance during the course of this work.

## References

[1] Leh et al. (2019) — "Smart irrigation system using Internet of Things" (ICSET): Discusses a low-cost IoT-based irrigation approach that emphasizes energy-

efficient sensor networks and real-time monitoring. (IEEE ICSET, 2019) IJCA

[2] Laksiri et al. (2019) — "Design and optimization of IoT based smart irrigation system in Sri Lanka": Presents an optimized IoT deployment tailored for local conditions and cost-effectiveness. (IEEE ICIIS, 2019) IJCA

[3] Shufian et al. (2019) — "Smart irrigation system with solar power and GSM technology": Integrates sustainable energy with IoT and GSM communication, ideal for off-grid agriculture. (IEEE ICAEE, 2019) IJCA

[4] Karpagam et al. (2020) — "Smart irrigation system using IoT": Introduces sensor-based irrigation automation using IoT infrastructure, targeting irrigation efficiency improvements. (IEEE ICACCS, 2020) IJCA

[5] Kulkarni (2020) — "AgriTalk: IoT for Precision Soil Farming of Turmeric Cultivation": A real-world IoT deployment detailed in IEEE Internet of Things Journal, focusing on precision monitoring for crop-specific farming.
(IEEE IoT Journal, 2019) AIP Publishing

[6] Roy et al. (2021) — "AgriSens: IoT-Based Dynamic Irrigation Scheduling System for Water Management of Irrigated Crops": Proposes dynamic, sensor-based irrigation schedules with IoT for optimal water use. (IEEE IoT Journal, 2021)

## APPENDIX

### *A. About the Hardware and Software*

Appendix provides supplementary information related to the design, tools, technologies, and additional references used in developing the Smart Irrigation System for Precision Farming. It includes technical details, component specifications, code snippets, and reference links that support the overall understanding of the project. This section serves as a resource for future improvements, replication of the project, and academic documentation.

The Smart Irrigation System integrates IoT, embedded systems, and data analytics to create a sustainable and automated irrigation model. The combination of Raspberry Pi/ESP32, soil moisture sensors, DHT11/DHT22 sensors, and relay modules ensures efficient irrigation based on real-time soil and environmental conditions. The project is enhanced by Power BI dashboards, which visualize live data collected from the field, allowing farmers to monitor and control irrigation remotely through mobile or web interfaces.

The implementation process involved hardware setup, coding in Python, and integration with a cloud platform such as ThingSpeak, Firebase, or AWS IoT for real-time data storage and visualization. The Power BI tool was used to analyze and display the data in a user-friendly format. The project successfully demonstrates how IoT and automation can solve agricultural problems and promote smart farming practices.

Below are key reference details, components, and additional notes that support the project's development:

1) *Hardware Components*
- Raspberry Pi or ESP32 Microcontroller
- Soil Moisture Sensor (Analog/Digital)
- DHT11/DHT22 Temperature and Humidity Sensor
- Relay Module (5V)
- Submersible Water Pump
- Breadboard and Jumper Wires
- Power Adapter and USB Cable
- Wi-Fi Module (Built-in ESP32 or Raspberry Pi Wi-Fi)

2) *Software and Tools Used*
- Programming Language: Python / Arduino IDE
- Operating System: Raspberry Pi OS / Windows 10
- Cloud Platform: ThingSpeak / Firebase / AWS IoT
- Dashboard: Microsoft Power BI
- Database: SQL or NoSQL (for data logging)
- Libraries: Adafruit_DHT, RPi.GPIO, Requests, Flask
- Development Tools: Visual Studio Code, Thonny IDE

3) *Key Features Implemented*
- Real-time monitoring of soil moisture, humidity, and temperature.
- Automatic control of water pump based on moisture levels.
- Live data visualization through Power BI dashboard.
- Multi-language user interface for accessibility.
- Cloud-based data storage for remote access and analysis.

4) *Testing and Validation*
- Sensor readings validated using calibrated devices.
- System tested under different soil and weather conditions.
- Cloud connectivity and dashboard updates verified.
- Performance tested for reliability and response speed.

5) *Reference Links and Resources*
- https://thingspeak.com – IoT Cloud Platform
- https://powerbi.microsoft.com – Data Visualization Tool
- https://www.raspberrypi.org – Raspberry Pi Documentation
- https://firebase.google.com – Cloud Database Platform

6) *Outcomes and Benefits*
- Promotes efficient water management and conservation.
- Reduces manual labor through automation.
- Provides real-time insights for better decision-making.
- Supports sustainable and precision farming practices.
- Can be expanded for larger farms and multi-crop systems.

7) *Team Contributions*
- Hardware Setup: Sensor integration and circuit connections.
- Software Development: Python coding, cloud integration, and data flow.
- Dashboard Design: Power BI visualization and real-time data analytics.
- Testing & Validation: Field trials and performance evaluation.