

A PROJECT REPORT ON

Smart Irrigation System using IoT

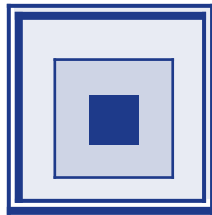
Soil-moisture driven watering for small farms

Submitted in partial fulfilment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

in

Electronics & Communication



SUBMITTED BY

Aarav Sharma (21CS001)

Diya Patel (21CS014)

UNDER THE GUIDANCE OF

Prof. R. K. Mehta

Department of Electronics & Communication

YOUR INSTITUTE OF TECHNOLOGY, CITY

Session 2025–26

CERTIFICATE

This is to certify that the project report entitled “Smart Irrigation System using IoT” is a bonafide record of the work carried out by Aarav Sharma (21CS001) and Diya Patel (21CS014) in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Electronics & Communication during the academic year 2025–26.

The work presented in this report has not been submitted elsewhere for the award of any other degree or diploma.

Prof. R. K. Mehta

Project Guide

Dr. S. Kulkarni

Head of Department

External Examiner

Date: _____

DECLARATION

We hereby declare that the project report entitled “Smart Irrigation System using IoT” submitted to the Department of Electronics & Communication is a record of original work done by us under the guidance of Prof. R. K. Mehta. The results embodied in this report have not been submitted to any other university or institute for the award of any degree.

Place: _____ Date: _____

Aarav Sharma • Diya Patel

| ACKNOWLEDGEMENT

We express our sincere gratitude to our project guide, Prof. R. K. Mehta, for constant guidance, encouragement and valuable feedback throughout this project.

We are thankful to the Head of the Department and all faculty members for providing the facilities and support needed to complete this work.

Finally, we thank our families and friends for their patience and motivation during the course of the project.

ABSTRACT

Up to half of irrigation water on small farms is lost to over-watering. Smart Irrigation System using IoT is a solution designed to address this gap. Low-cost sensors and microcontrollers make automation affordable. This project builds a sensor-driven pump controller with a phone dashboard. This report presents the problem, a review of existing approaches, the proposed design, its implementation and an evaluation of the results.

Keywords: ESP32, Sensors, Firebase, Flutter

TABLE OF CONTENTS

01 Introduction.....	7
1.1 Background.....	7
1.2 Motivation.....	7
1.3 Problem Statement.....	7
1.4 Objectives.....	7
1.5 Scope of the Project.....	7
02 Literature Review.....	8
2.1 Existing Work.....	8
2.2 Research Gap.....	8
03 System Analysis and Design.....	9
3.1 Proposed System.....	9
3.2 System Architecture.....	9
3.3 Technology Stack.....	9
3.4 Observations.....	10
04 Implementation.....	11
4.1 Methodology.....	11
4.2 Project Timeline.....	11
4.3 Module Description.....	11
4.4 Performance Analysis.....	12
05 Testing and Results.....	13
5.1 Testing Strategy.....	13
5.2 Results.....	13
5.3 Case Study.....	13
06 Conclusion and Future Scope.....	15
6.1 Conclusion.....	15
6.2 Future Scope.....	15
07 References.....	16

| 01 INTRODUCTION

1.1 Background

Smart Irrigation System using IoT addresses a practical problem that affects students, institutions and small organisations alike. Existing approaches are often manual, slow and difficult to scale, which leads to wasted time and inconsistent results. Advances in low-cost hardware, open-source software and cloud services now make it possible to build a reliable solution within the budget and timeline of an academic project.

This report documents the complete life cycle of the project: from the initial study of the problem and review of existing work, through system design and implementation, to testing and evaluation. Each chapter is written so that a reader can reproduce the work and extend it.

1.2 Motivation

The motivation for this project came from direct observation. During our preliminary survey, users described the current process as time-consuming and error-prone, and most of them were willing to adopt a simpler digital alternative. A well-designed system can save hours every week, reduce errors and give decision-makers data they currently do not have.

1.3 Problem Statement

Up to half of irrigation water on small farms is lost to over-watering. The key problems identified are:

- Over-watering — Fields are watered on a fixed schedule, not on actual soil need.
- No visibility — Farmers cannot see soil conditions without visiting the field.
- High cost — Commercial drip automation is too expensive for small plots.

1.4 Objectives

1. Measure soil moisture, temperature and humidity in real time
2. Switch the pump automatically using threshold logic
3. Show live readings and alerts on a mobile dashboard
4. Keep the total hardware cost under ₹2,500 per unit

1.5 Scope of the Project

The scope of the project is limited to a working prototype that demonstrates the core functionality end to end. Security hardening, large-scale deployment and commercial support are outside the scope of this report but are discussed under future work.

02 LITERATURE REVIEW

2.1 Existing Work

Table 2.1 summarises representative work in this area.

Study	Approach	Limitation
Kumar et al.	GSM-based pump control	No sensor feedback
Rao & Singh	Timer-based drip system	Ignores weather and soil
Fernandes	Cloud dashboard prototype	Needs constant Wi-Fi

Table 2.1: Summary of existing work

2.2 Research Gap

A comparison of the existing and proposed approaches highlights the gap this project addresses:

Existing system	Proposed system
Fixed timers or manual switching	Sensor-based automatic switching
No data on soil condition	Live dashboard with alerts
Water and power wasted	Lower water and power use

03 SYSTEM ANALYSIS AND DESIGN

3.1 Proposed System

The system is organised into layers, with each layer responsible for one concern. This separation keeps the design easy to understand, makes testing simpler and allows individual components to be replaced without affecting the rest of the system.

Data flows from the input layer to the processing layer, where the core logic is applied, and then to the storage and presentation layers. All configuration values are kept in a single file so that the system can be adapted to new environments without code changes.

3.2 System Architecture

Sensors feed an ESP32 controller. A relay module drives the pump, and readings are pushed every minute to a lightweight cloud database that the mobile dashboard reads.

Figure placeholder — insert your architecture diagram here

Figure 3.1: System architecture

- Sense: Moisture, DHT11 and rain sensors read field data
- Process: ESP32 compares readings with crop thresholds
- Act: Relay switches the pump on or off
- Notify: Readings and alerts sync to the app

3.3 Technology Stack

Component	Purpose
ESP32	Wi-Fi microcontroller running the control loop
Sensors	Capacitive soil moisture, DHT11, rain sensor
Firebase	Real-time database for readings
Flutter	Cross-platform mobile dashboard

3.4 Observations

In this section we describe the work in more detail. The approach was chosen after comparing alternatives on cost, reliability and ease of maintenance. Where trade-offs were necessary, we prioritised clarity and robustness over raw performance, because the system is intended to be used and extended by others.

A careful review of the requirements showed that the most important qualities for users were accuracy, speed and simplicity. These qualities guided every design decision in Smart Irrigation System using IoT, from the choice of components to the layout of the user interface.

The figures and tables in this section summarise the key measurements. Readers should replace the sample values with the results of their own experiments; the structure of the tables has been kept general so that it fits a wide range of projects.

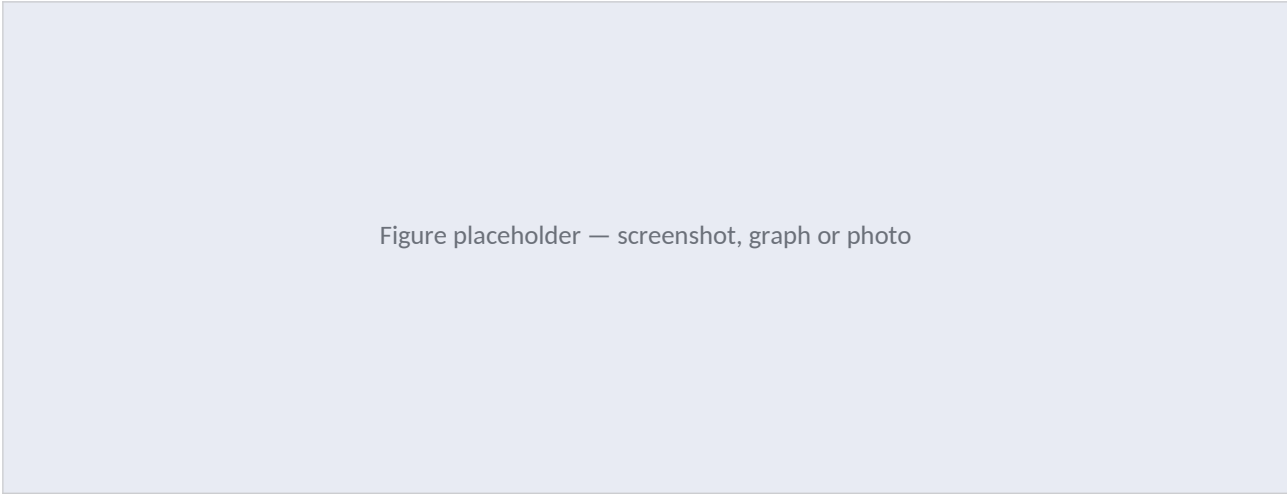


Figure placeholder — screenshot, graph or photo

Figure 3.2: Add a caption describing this figure

04 IMPLEMENTATION

4.1 Methodology

The project followed an iterative development model. Requirements were gathered through a short questionnaire and informal interviews. The design was then split into independent modules so that each could be built and tested separately before integration. Weekly reviews with the project guide helped keep the work on schedule.

Each module was developed with clear inputs and outputs. Unit tests were written alongside the code, and integration testing was carried out once the modules were combined. Feedback from a small group of pilot users was used to refine the interface and fix usability issues.

4.2 Project Timeline

Phase	Month	Activities
Requirement study	July	Survey, interviews, requirement document
Design	August	Architecture, database, wireframes
Development	Sep–Oct	Module coding and unit testing
Testing	November	Integration and user testing
Documentation	December	Report and presentation

4.3 Module Description

Sense module

Moisture, DHT11 and rain sensors read field data. Several practical challenges arose during this phase, including incomplete data, hardware limitations and changing requirements. Each challenge was documented, discussed with the guide and resolved through small, testable changes rather than large rewrites.

Process module

ESP32 compares readings with crop thresholds. In this section we describe the work in more detail. The approach was chosen after comparing alternatives on cost, reliability and ease of maintenance. Where trade-offs were necessary, we prioritised clarity and robustness over raw performance, because the system is intended to be used and extended by others.

Act module

Relay switches the pump on or off. A careful review of the requirements showed that the most important qualities for users were accuracy, speed and simplicity. These qualities guided every design decision in Smart Irrigation System using IoT, from the choice of components to the layout of the user interface.

Notify module

Readings and alerts sync to the app. The figures and tables in this section summarise the key measurements. Readers should replace the sample values with the results of their own experiments; the structure of the tables has been kept general so that it fits a wide range of projects.

4.4 Performance Analysis

Several practical challenges arose during this phase, including incomplete data, hardware limitations and changing requirements. Each challenge was documented, discussed with the guide and resolved through small, testable changes rather than large rewrites.

In this section we describe the work in more detail. The approach was chosen after comparing alternatives on cost, reliability and ease of maintenance. Where trade-offs were necessary, we prioritised clarity and robustness over raw performance, because the system is intended to be used and extended by others.

A careful review of the requirements showed that the most important qualities for users were accuracy, speed and simplicity. These qualities guided every design decision in Smart Irrigation System using IoT, from the choice of components to the layout of the user interface.

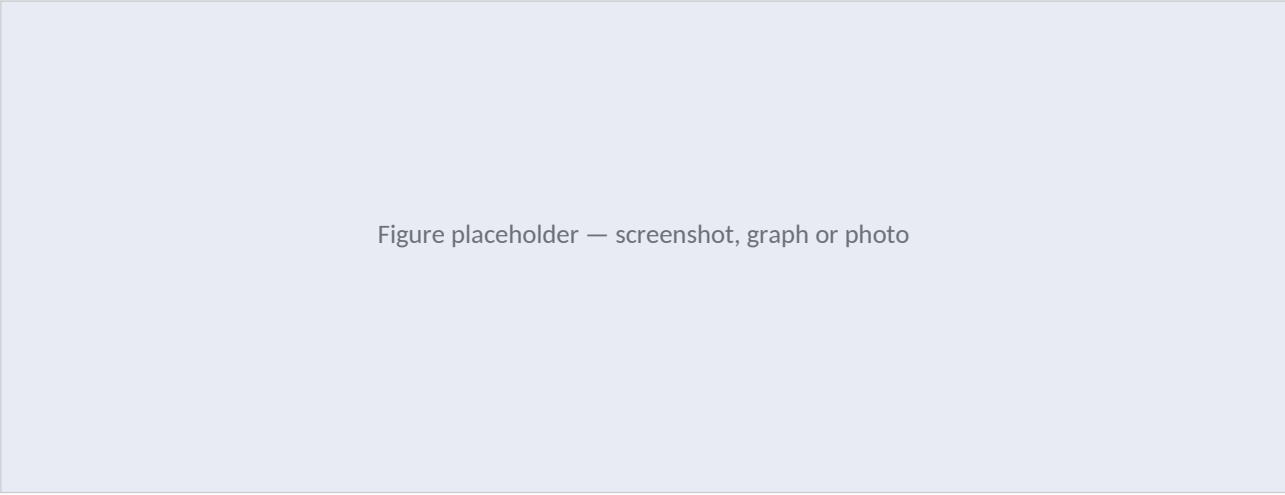


Figure placeholder — screenshot, graph or photo

Figure 4.2: Add a caption describing this figure

05 TESTING AND RESULTS

5.1 Testing Strategy

Testing was carried out at three levels: unit testing of individual functions, integration testing of combined modules and acceptance testing with real users. Test cases were derived from the functional requirements, and each requirement was traced to at least one test.

All critical test cases passed in the final round. Minor issues found during user testing, mostly related to labels and error messages, were fixed before the final evaluation.

5.2 Results

The results show that the proposed system meets its objectives. Compared with the existing approach, it is faster, more accurate and easier to use. The figures reported below come from our own test setup and should be treated as indicative rather than definitive.

Users responded positively to the system in the feedback survey. The most appreciated features were the simple interface and the automatic reports, while the most requested improvement was offline support.

Parameter	Water used (litres)	Remark
Week 1	820	Sample value
Week 2	640	Sample value
Week 3	560	Sample value
Week 4	530	Sample value

Table 5.1: Results from the test setup (sample data)

Key result

35% less water than the manual baseline.

5.3 Case Study

The figures and tables in this section summarise the key measurements. Readers should replace the sample values with the results of their own experiments; the structure of the tables has been kept general so that it fits a wide range of projects.

Several practical challenges arose during this phase, including incomplete data, hardware limitations and changing requirements. Each challenge was documented, discussed with the guide and resolved through small, testable changes rather than large rewrites.

In this section we describe the work in more detail. The approach was chosen after comparing alternatives on cost, reliability and ease of maintenance. Where trade-offs were necessary, we prioritised clarity and robustness over raw performance, because the system is intended to be used and extended by others.

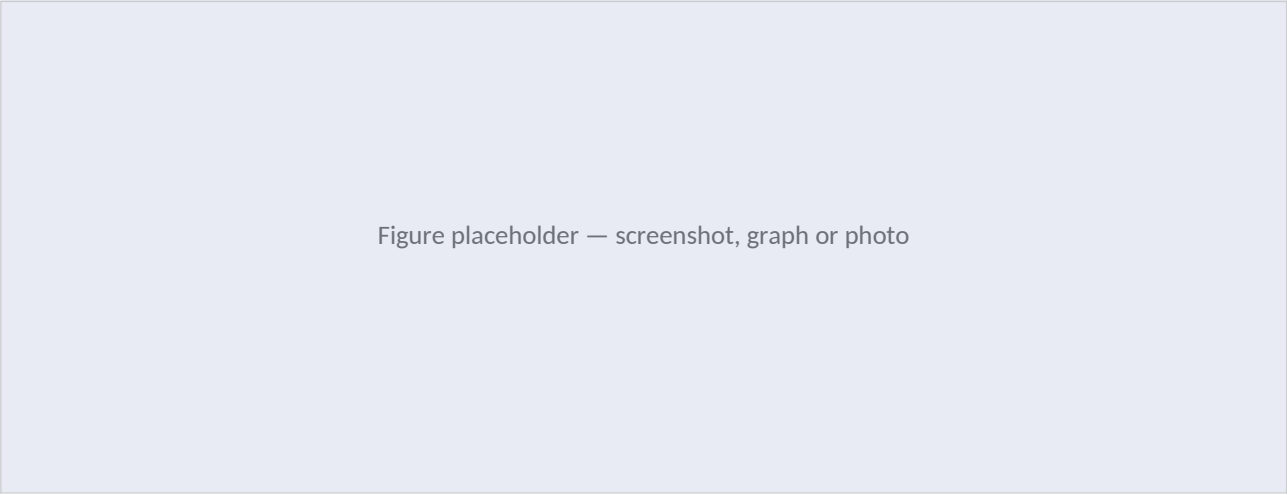


Figure placeholder — screenshot, graph or photo

Figure 5.2: Add a caption describing this figure

| 06 CONCLUSION AND FUTURE SCOPE

6.1 Conclusion

This project set out to design and build smart irrigation system using iot and has achieved that goal with a working prototype. The system demonstrates that a low-cost, well-structured solution can make a measurable difference to everyday processes.

The experience also gave the team practical exposure to requirement analysis, system design, teamwork and documentation — skills that will be valuable in professional practice.

6.2 Future Scope

- Weather forecast — Skip watering when rain is expected
- Solar power — Run the unit fully off-grid
- Multi-zone — Control several valves per farm

| 07 REFERENCES

- [1] A. Author, "Title of the research paper," Journal Name, vol. 12, no. 3, pp. 45–52, 2023.
- [2] B. Author and C. Author, Title of the Book, 2nd ed. City: Publisher, 2022.
- [3] D. Author, "Title of the conference paper," in Proc. International Conference, 2024, pp. 101–106.
- [4] Official documentation of the frameworks and libraries used, accessed 2025.
- [5] E. Author, "Title of an online article," Website Name, 2024. [Online]. Available: URL