DESIGN AND IMPLEMENTATION OF IOT BASED IRRIGATION SYSTEM

BY

Nwabueze Onyedikachi Mark-Anthony
GOU\U22\CSC\805

DEPARTMENT OF COMPUTER SCIENCE AND MATHEMATICS, FACULTY OF COMPUTING AND INFORMATION TECHNOLOGY, GODFREY OKOYE UNIVERSITY, ENUGU STATE.

IN PARTIAL FULFILLMENT OF THE AWARD OF BACHELOR OF SCIENCE (B.SC), DEGREE IN COMPUTER SCIENCE.

SUPERVISOR: Engr. Camillus Njoku

MAY, 2026

APPROVAL PAGE

This research project, titled " DESIGN AND IMPLEMENTATION OF IOT BASED IRRIGATION SYSTEM," has been examined and approved by the Department of Computer Science and Mathematics, Faculty of Natural Sciences and Environmental Studies, Godfrey Okoye University, Enugu, Nigeria.

	....	....
Engr. Camillus Njoku	Signature	Date
Project Supervisor		
External Examiner	....	....
External Examiner	Signature	Date
Dr. Frank Okebanama	....	....
HOD, Department of Computer Science and Mathematics	Signature	Date
Prof.Dr. I A Ajah	....	....
Dean, Faculty of Computing	Signature	Date

And Information Technology.

CERTIFICATION

I certify that this project was carried out by me and that all sources of information used have been properly acknowledged. I understand that the university will take action if it is found that I submitted work that is not entirely mine.

____

Nwabueze Onyedikachi Mark-Anthony
GOU\U22\CSC\805

DEDICATION

This project is dedicated to my parents, who worked hard to make sure I got an education. Their love, kindness and support throughout my time in this university kept me going, especially during the difficult moments. I also dedicate this work to every private farmer out there who has had to struggle with having successful harvest period. I hope this system makes things a little easier for those who come after me.

ACKNOWLEDGEMENTS

First, I thank the Lord for granting me the grace to write this thesis successfully. Everything is made possible by him.

Secondly, I wish to thank my tutor, Mr. Camillus Njoku, for his assistance, corrections, and patience throughout this project. Anytime I had doubts or felt lost and challenges, he would always take time out to show me the way forward.

Thirdly, I wish to thank my family members, particularly my parents and brothers, for their support throughout my academic career. Your faith in me is a great encouragement to me. I thank my friends and other students for their contributions towards this work.

ABSTRACT

In an increasingly water-challenged world, agriculture is still one of the primary sources of food, economic activity, and sustainable development and accordingly, efficient water management has become of greater and greater relevance. Traditional irrigation practices employed by smallholders often involve labor-intensive and time-intensive operations, leading to water loss, labor reliance, and sub-optimal usage of water. To overcome these drawbacks, this research work has an objective of designing and developing an IoT based irrigation system using ESP32 microcontroller, which is an automatic water dispenser as per the actual soil moisture level. In particular, the goals included the development of a soil moisture sensing system, the possibility of automatic soil moisture control, remote monitoring, the incorporation of weather-based decision support, and testing the performance of the prototype developed. System Design and Implementation were carried out with the help of a prototype-based Development Methodology and an Object-Oriented Analysis and Design (OOAD) approach. The hardware architecture included an ESP32 microcontroller, a soil moisture sensor, a DHT22 temperature and humidity sensor, a water level sensor, a relay module, a water pump, LEDs, and other power supply components, while the software architecture involved embedded C++ programming for the ESP32, the use of Python and Flask for backend services, React for the user interface, and cloud-based IoT communication technologies. The developed system monitored soil moisture conditions continuously and automatically turned on irrigation in case the soil moisture is below the pre-described threshold. Furthermore, a web-based dashboard was developed to facilitate the real-time visualization of data collected from the sensors, as well as the status of the irrigation system, environmental conditions and weather forecast data. In addition, the system has a human-in-the-loop capability to allow the manual override of the automated irrigation control when and if needed, and weather forecast data was provided to assist in intelligent irrigation planning and to avoid over-irrigation prior to the arrival of rain. The weather-aware decision support function proved beneficial to irrigation planning and also played an important role in conserving water; the human-in-the loop feature in turn provided added flexibility in water use decisions. In conclusion, the proposed system was able to overcome the major drawbacks of traditional irrigation methods, such as water loss, late decision-making and high labour demand.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE

INTRODUCTION

1.1 Background of the Study

Agriculture plays a critical role in the economic development of many countries worldwide, including Nigeria, as it involves the cultivation of crops and livestock to support the growing human population. It provides nourishment for human beings, helping eliminate extreme poverty and boost the economies of countries such as the Netherlands, Nigeria, Thailand, and many more. According to the World Bank, in 2022, agriculture accounted for 4% of the world's gross domestic product (GDP), with projections indicating that this agricultural production will feed the expected increase in the world's population to 9.7 billion (Pereira et al., 2023). Agriculture has always relied on technology from the beginning, spanning from the use of extremely simple tools made of wood and bones for farming through many ages. And as time went on, mankind developed better tools for farming to increase efficiency and reduce energy loss. Consequently, the second agricultural revolution in the United States of America brought the advent of tractors, which were commonly seen in farmlands (Bellis, 2022).

The recent rise in population growth, food shortage, and water scarcity are major challenges affecting the rate of global sustainable development. Currently, the world's population is around 8 billion, and approximately 10% of this population is chronically hungry (Zhang et al., 2025), necessitating the enhancement of agricultural productivity and the ensuring of food security. To achieve the Sustainable Development Goal (SDG) of "Zero Hunger", the Food and Agriculture Organisation of the United Nations (FAO) identified the adoption of smart irrigation technology as one of six key technological enhancements for global food security (Gao et al., 2023). According to Duguma and Bai (2024), about 72% of the globe's freshwater resources are being used for all kinds of farming worldwide, and irrigation accounts for about 41% of this. In Asian countries such as China, the situation is much higher, as irrigation water accounts for roughly 86% of total water use in agriculture, despite not being effectively utilised. Additionally, a higher percentage of this agricultural production is carried out by small-holder farmers, and their application of irrigation is based on their personal experience rather than a smart and precise approach, resulting in inefficient water use (Duguma & Bai, 2024). The adoption of agriculture 4.0 technologies like Internet of Things (IoT), Artificial Intelligence (AI), and edge computing in irrigation will transform how water is managed in the farms, as traditional irrigation systems rely heavily on manual control, which often leads to water wastage, inefficient distribution, and increased labour costs (Froiz-Míguez et al., 2020).

Internet of Things (IoT) is used in circumstances that requires data collection, automation, or control. As IoT becoming more popular, there has been a rise in smart agricultural technology ideas for enhancing farm productivity and also ensuring food security (Verma et al., 2022). Zhang et al. (2025) designed an IoT-based smart irrigation system using LoRa and edge computing to improve automation, data integration, and decision-making, achieving remarkable communication results, but the system requires high-quality data to integrate ML. In a similar study, Munir et al. (2018) developed an IoT-based smart irrigation and energy-saving system for tunnel farming with results demonstrating better water saving than previous approaches and manual watering. However, the system was tested on a small area. An IoT-based system for monitoring soil water status, including soil water content, matric potential, and hydraulic conductivity, was designed by Comegna et al. (2024) and achieved an $R^2 = 0.99$ for estimating soil water content, demonstrating robust long-term data acquisition. Their study is limited because capacitive sensor readings can be affected by soil texture, temperature, salinity, and soil type. There is a need to design an automatic irrigation system for smart farming that improves on the limitations of this existing system and likewise upgrades the manual irrigation method still widely utilised in Sub-Saharan Africa, especially Nigeria.

The proposed system utilizes ESP32 microcontroller-based intelligent irrigation system that incorporates multiple sensors and controllers such as a soil moisture sensor, ambient temperature sensor, relative humidity sensor, water flow sensor, and an electronically controlled solenoid valve. The developed solution is different from traditional automated irrigation systems by incorporating a human-in-the-loop approach, allowing the irrigation user to keep a close watch on irrigation activities, while also providing them with a set of irrigation decision support functionalities supported by weather information for better irrigation water management.

A web-based application was designed with a unified dashboard for easy monitoring and operation of the irrigation system, giving a real-time view of the irrigation system. It displays sensor data, water consumption, and can switch off or on automatic irrigation, manually run the irrigation valve if needed, and show graphical trends of soil moisture, temperature, and atmospheric humidity. In the normal operation of the system, when the solenoid valve is activated, water can be pumped directly to the root zone of the crop. The soil moisture sensor permanently monitors the moisture level in the soil and the controller decides if irrigation is necessary or not. If the moisture amount that is measured is below the set value and the temperature is suitable, the ESP32 automatically starts the irrigation system by turning on the solenoid valve. Moreover, users can use the automatic irrigation mode or not, depending on the humidity level or operational needs, enabling intelligent automation that allows for user-driven decision-making.

1.2 Statement of Problem

Conventional irrigation methods, such as manual, non-automated watering based on fixed schedules or farmers' judgment, have several limitations that affect food security. Currently, most low- and middle-income countries practice this method, and smallholder farmers in Nigeria, in particular, rely on manual irrigation.

1. Over-irrigation or under-irrigation: This arises when water is applied without considering real-time soil moisture and rainfall possibilities, as seen in conventional irrigation, which leads towards waterlogging or insufficient water supply for crop growth.

2. Lack of real-time monitoring: The existing manual system of irrigation lacks the facilities of presenting continuous information on soil condition, pump status, and weather changes, thereby making timely irrigation decisions difficult.

3. High labour dependency: Traditional irrigation systems require human supervision and monitoring to inspect soil dryness, operate the pumps, and respond to changing weather conditions.

4. Water wastage: The existing system of Irrigation without sensor-based control and a weather forecast system causes waste of water, especially when watering continues shortly before rainfall or immediately before rainfall as a result of a lack of predictive capabilities.

In summary, these problems in a big way limit productivity and sustainability in farming as a result of the increased loss of water, weakening crop health, and also the lack of effective and efficient management of farm resources, which leads to waste of potential materials that would have aided more productivity.

1.3 Aim and Objective of the Study

The aim of this study is to design and implement an IoT-based irrigation system using an ESP32 to automate water dispensing based on soil moisture levels. And the specific objectives are to:

1. Design a soil moisture sensing system using an ESP32 microcontroller for real-time data acquisition.
2. Develop an automated irrigation control mechanism based on soil moisture thresholds.
3. Integrate IoT-based remote monitoring to enable real-time visualisation and control of irrigation parameters.
4. To implement a real-time decision-making system that incorporates a human-in-the-loop mechanism and weather predictions.
5. To evaluate system performance in real time using the designed prototype.

1.4 Significance of the Study

The impact of this study will be significant for the global agricultural system and its stakeholders, including farmers; government officials; companies; researchers, by researching and implementing newer technologies like the Internet of Things (IoT), Sensor Technologies, and Weather Forecasts into a fully automated irrigation system that will generate more food from farms. Some important outcomes resulting from this study include:

1) Supporting the implementation of intelligent agriculture through showcasing the benefits associated with utilising new technologies for agriculture, in particular, irrigation.

2) The implementation of this system will reduce water wasted by using real-time soil data collected by sensors together with data collected on weather patterns to allow sufficient irrigation only when required, thus conserving our limited supply of water resources, as well as eliminating crop loss due to oversaturation.

3) By ensuring that crops grown in Enugu State, Nigeria receive sufficient water only when required; therefore, promoting maximum crop production and maintaining healthy soil conditions.

4) Integration with an intelligent human-in-the-loop function; therefore, users will only need to provide intervention when absolutely necessary, thus reducing manual labor on an ongoing basis while continuing to maintain intelligent supervision.

1.5 Scope of the Study

The focus of the project is to develop a smart irrigation system that will monitor the real-time moisture levels in all of the soil and then use that data to automatically deliver or stop delivering water to your plants based on pre-set thresholds. Additionally, the entire system can be managed and monitored remotely via an ESP32 microcontroller embedded within an IoT platform. A weather aware decision layer will also be developed within a web application so that a farmer or any user can incorporate short-term rainfall forecasts to make irrigation decisions, such as suspending watering if a forecast of rain occurs within the near future as determined by the application, thus reducing the chances of waterlogging while having efficient farm management.

1.6 Operational Definition of Terms

Automated Irrigation System: A system that automatically irrigates plants based on sensor readings or conditions, without requiring manual intervention to turn the water on or off.

Soil Moisture Sensor: This sensor was used in this study to measure the moisture content of the soil and determine when to irrigate the system.

ESP32 Microcontroller: The primary control component of the project, responsible for collecting sensor data and communicating with the internet for monitoring and control.

IoT (Internet of Things): IoT for me is just putting the irrigation system online using the ESP32 functionality, allowing for the remote monitoring and control of the system.

Automatic Irrigation Control: This is the process in which the system activates the water pump automatically when it detects that the soil moisture is low.

Human in the Loop: Even though the system is automated, a human user can still intervene and make decisions, when necessary, particularly due to the weather.

Weather Prediction Integration: This is where the weather data will be used in the web app to help determine if irrigation should be postponed or avoided if rainfall is imminent.

Waterlogging: Excessive water in the soil that may have a detrimental effect on crop growth and yield.

Web-Based Monitoring System: An easy-to-use web-based interface created in this project to remotely monitor the system, sensor data, and begin and end irrigation.

Real-Time Monitoring: It means that the system is constantly updated and displays real-time data, not after a certain period of time (such as soil moisture, system status, etc.).

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

This chapter reviews existing work on irrigation systems, IoT in agriculture, and sensor-based automation. It discusses the traditional irrigation methods, such as flood irrigation and manual watering, which lack efficiency and automation and can lead to poor management. Furthermore, the chapter examines smart irrigation systems, including their operational methods, such as the use of sensors, microcontrollers, and automated controls. IoT was discussed, as it enables real-time monitoring, remote control, and data-driven farming in order to effectively realise crop monitoring, weather tracking and smart irrigation. Finally, the soil moisture sensors and the ESP32 in the IoT system were discussed in detail.

2.2 Theoretical Background

2.2.1 Traditional Irrigation Systems

Traditional irrigation practices such as manual surface watering and flooding are still being used in many middle and low-income agricultural countries due to low initial infrastructure costs and simplicity of implementation (FloraFlex, 2023). However, these old methods are still characterised by their inefficiencies, which is seen as a significant volume of water wasted through high rates of evaporation, runoff, and deep percolation (RJ Wave, 2025). Manual watering might be the affordable and accessible option for small -scale farmers, but it is plagued by labour intensity, human error in application timing, and inconsistent water distribution patterns that can lead to intense waterlogging or nutrient leaching (FloraFlex, 2023; RJ Wave, 2025). These traditional systems we are talking about lack integrated feedback mechanisms that will enable farmers or users to control actions, such as stop irrigation when the soil has moisture or even prevent waterlogging, and the lack of this feature renders them unresponsive to real-time changes in soil moisture levels and also makes them unreliable for crops that require a certain level of water. This problem with the existing traditional system ultimately prevents optimal agricultural productivity, which might contribute to the over-extraction of finite water resources (RJ Wave, 2025).

2.2.2 Smart Irrigation Systems

The concept of smart Irrigation refers to the adoption of technology and artificial intelligence in executing agricultural or farming practices. It represents a critical technological shift toward precision agriculture by integrating environmental sensors, microcontrollers, and automated control logic to optimise water application in farmlands or other agricultural practices based on real-time data. With the adoption of smart irrigation systems that involve continuous monitoring of soil moisture levels, these systems facilitate dynamic, responsive scheduling that ensures crops receive water only when necessary, thereby significantly reducing waste compared to traditional methods that lack control and monitoring (Institute for Irrigation Research, 2025). Figure 2.1 presents a diagram illustrating smart irrigation in action.

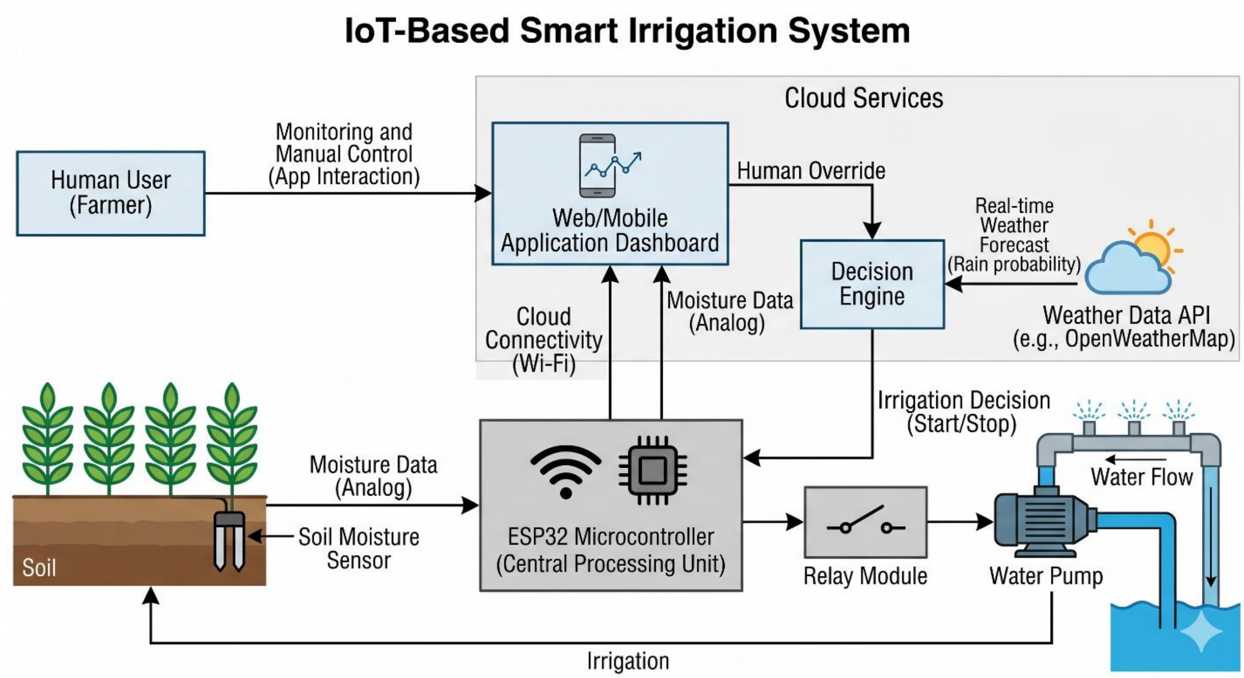

Figure 2.1 IoT-Based Smart Irrigation System

Figure 2.1 is an automated IoT loop in which an ESP32 microcontroller processes soil moisture readings from a soil sensor to automatically irrigate the farmland, while also showing cloud-based weather data that enables the human in the loop to intelligently manage irrigation via a relay-controlled pump. The architecture of this smart irrigation system features a human-in-the-loop design, allowing the farmer to monitor real-time data and exercise manual override via a web or mobile dashboard.

2.2.3 Internet of Things (IoT) in Agriculture

The Internet of Things (IoT) which refers to an interconnected ecosystem of physical objects, including household appliances, vehicles, industrial equipment, and other smart devices, which are equipped with embedded sensors, software applications, and communication technologies. These capabilities give them the allowance in order to gather, process, and exchange information over a network, enabling seamless interaction and automated data-driven operations. It has fundamentally transformed modern agriculture by enabling the easy integration of physical farming resources with digital monitoring networks, and also leverages a complex architecture of sensors, wireless communication protocols, and cloud-based data analytics, so that farmers can achieve real-time visibility into their crop environments from any remote location (Al-Turjman, 2020). This shift in farming patterns from conventional to data-driven farming enables the precise collection of environmental variables that enhance decision-making and reduce reliance on traditional, guesswork-based management practices used in most farming activities in Sub-Saharan African countries. Ultimately, the adoption of IoT infrastructure serves as a foundational element that will help sustain agriculture, provide effective control that is necessary to optimise resource allocation and improve overall operational transparency.

The applications of IoT in the agricultural sector are diverse. It ranges from advanced crop health monitoring, automated irrigation scheduling, to complex environmental tracking, such as integrated sensor networks that can continuously monitor variables such as humidity, temperature, and solar radiation, so as to provide comprehensive insights that were previously difficult to capture manually (Babu et al., 2021). Automated actuators, such as solenoid valves and water pumps, are sometimes added to the systems to enable the system to irrigate efficiently according to the actual soil requirements. These implementations aren't just about simple automation that the system carries out, but also about feeding the data into a remote dashboard, which will enable the farmers and users to analyze historical trends and be able to predict the farmers' future needs. This helps to significantly boost agricultural productivity and adaptation to climate change.

2.2.4 Soil Moisture Sensors

In agriculture, soil moisture sensors are important to precision irrigation and are significant analytical tools to know the soil moisture state of a soil medium, especially in western countries. These soil moisture detection devices generally rely on measurement of either dielectric properties or electrical conductivity of the soil, providing insights and patterns that indicate variations in soil moisture levels associated with changes in soil's capacitance or resistance output (Petropoulos et al., 2022).

It works by measuring the change in the electrical properties of the soil when it dries up, triggering the sensor to output analogue or digital signals which depend on the change and which can be read by a microcontroller, such as the ESP32. Because of their quick response and high reliability, these sensors are the main feedback component in today's irrigation systems to ensure accurate measurements and that they are representative of the plant's active root zone.

Soil moisture sensors are an integral part of achieving actualization of smart or precision agriculture, as their integration is essential to irrigation scheduling decisions, and helps to prevent overwatering, a frequent problem in agriculture that results in wasted resources and damage to the root system of plants. To achieve this, an automated system can use set moisture thresholds to ensure that water is only applied when it is needed, reducing the amount of water used and ensuring efficient application (Zhang et al., 2023). In general, keeping soil water in the optimal range can avoid common problems of the existing irrigation system based on conventional irrigation technologies including waterlogging, nutrient leaching and static nature of manual, flooding based and timer based irrigation with limited predictive water knowledge. In farming, it has been proven that water savings are significant when using sensor-based decision making, and that it creates a better growing environment for healthy roots and longer crop life. The soil moisture sensor circuit is shown in figure 2.2 with ESP32 microcontroller.

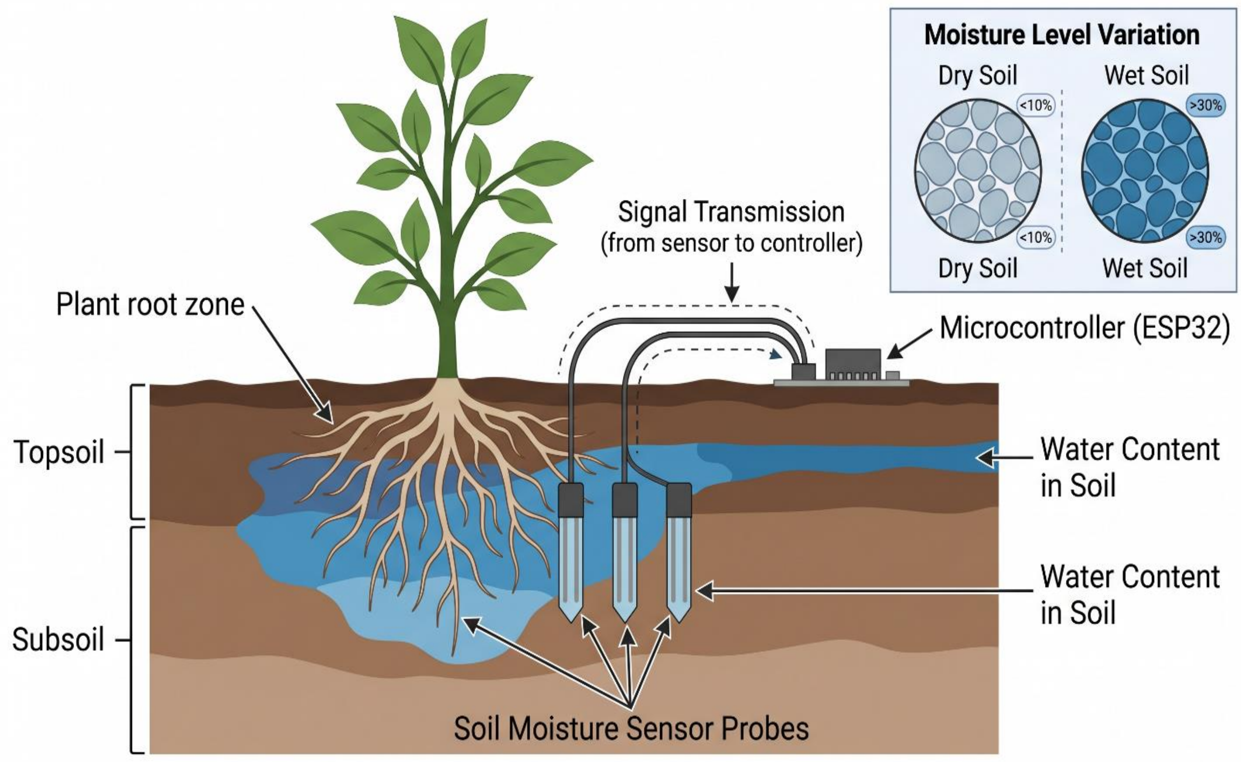

Figure 2.2: Soil Moisture Sensor connected to ESP32

2.2.5 ESP32 in IoT Systems

The ESP32 is equipped with integrated Wi-Fi and Bluetooth capabilities. A study by Koubaa et al. (2021) states that these features that are contained in the ESP32 microcontroller enable it to seamlessly communicate between local sensor networks and external cloud platforms, making it exceptionally well-suited for remote monitoring and control applications in smart irrigation farming. The ESP32 contains a powerful dual-core processor that provides sufficient computational capacity to handle multiple sensor inputs in a simultaneous approach while at the same time managing complex irrigation logic and web-based data transmission. Additionally, the availability of comprehensive development environments and low power consumption make the ESP32 an affordable and highly adaptable solution for both small-scale garden automation and larger, distributed agricultural networks.

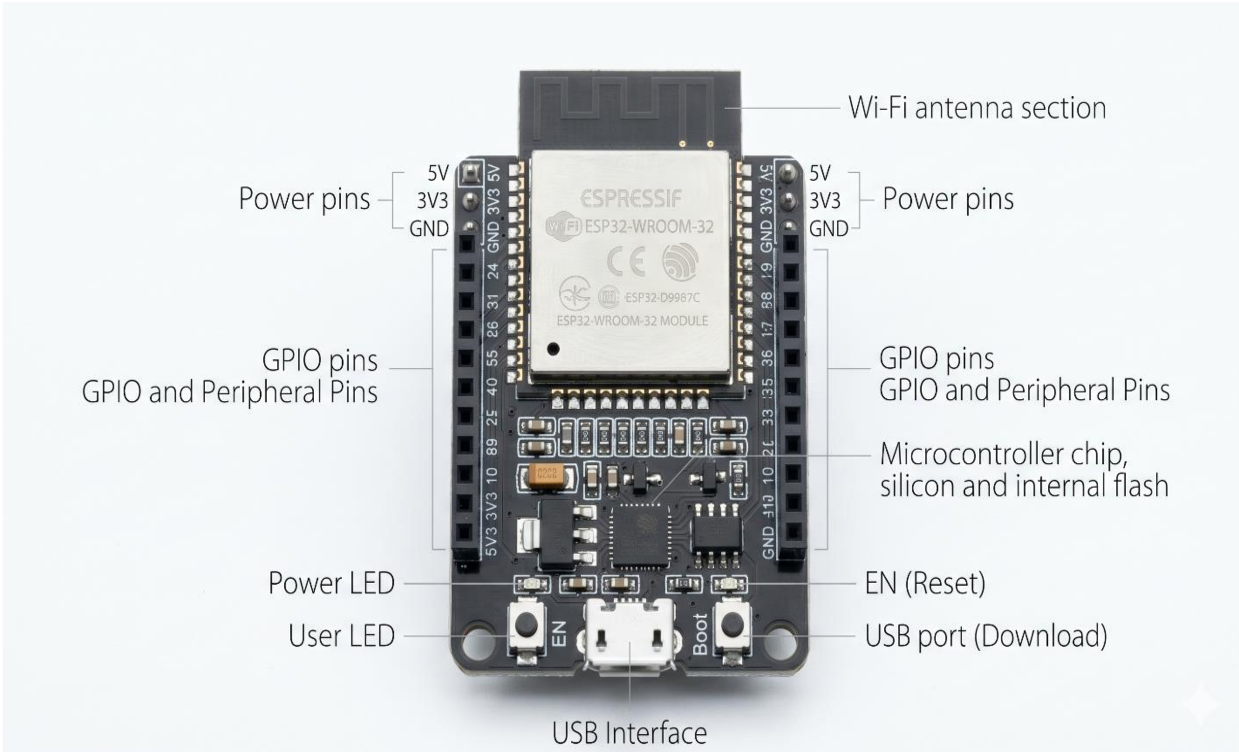

Figure 2.3 ESP32 microcontroller

Figure 2.3 shows an ESP32 microcontroller, which is currently adopted in building smart automated irrigation systems, and its increasing adoption is largely attributed to its balance of high-performance processing and seamless integration with standard peripheral components. In many reported system designs, the ESP32 acts as the central control hub or the brain, which aggregates data from soil moisture sensors and executes command signals for relay-controlled pumps based on real-time environmental conditions obtained (Gupta et al., 2022). Many studies have frequently stated that the ESP32 has the ability to facilitate user-friendly interaction, such as sending alerts or status updates to mobile applications, which simplifies the management of irrigation cycles for non-technical users. Because of these significant features and its flexibility, reliability, and simple deployment protocol, the ESP32 continues to be a preferred choice for building scalable, responsive, and efficient automated irrigation solutions.

2.3 Review of Related Literature

Padiachy et al. (2025) developed a smart IoT irrigation system to improve water efficiency and agricultural productivity in a city facing water scarcity, employing a prototype-based system that integrates sensors, a microcontroller, and automation with threshold-based control logic. The technique adopted by the study for this system involves rule-based threshold control that uses only sensor data, including soil, temperature, and humidity. Results demonstrate water savings and improved crop yield. However, there are problems with data security, system integration, and adoption.

Zhang et al. (2025) designed an IoT-based smart irrigation system using LoRa and edge computing to improve automation, data integration, and decision-making. The study consists of hardware-software co-design using an EC-IoT architecture, LoRa communication, an edge computing gateway, an irrigation algorithm, and multimodal environmental data (soil and weather). The study achieved remarkable communication results, achieving a distance of about 4km and a transmission time of 0.9–6 s, while soil moisture remained within an optimal range. However, the system will require high-quality data to integrate ML, and there is also overreliance on meteorological data.

Habib et al. (2023) designed an IoT-based automated wastewater irrigation system to optimise water usage in water-scarce regions of the Middle East, using a methodology comprising cloud-based processing, an IoT framework with a wireless sensor network, and an HTTP dashboard for monitoring and control. The study made use of real-time multimodal data, that is, soil, weather, evapotranspiration, and forecast, to test and evaluate the designed system. Pilot test confirms the system's feasibility, with a cloud-based monitoring dashboard. The efficiency of this system is marred by water scarcity, dependence on infrastructure, and environmental constraints.

An IoT-based smart irrigation and energy-saving system was developed by Munir et al. (2018) for tunnel farming due to water scarcity and inefficient manual irrigation. The method adopted by the study involves an experimental prototype that uses IoT sensors, fuzzy-logic decision support, a plant water-need database, server processing, and mobile app monitoring for control and potential analysis. Experimental sensor data from a small tunnel home garden setup were used to evaluate water saving, productivity improvement, and energy efficiency, with results demonstrating better water saving than previous approaches and manual watering. However, the system was tested on a small area, and scalability to large fields was suggested but not fully validated.

Purnama et al. (2024) developed an ESP32-based IoT irrigation control system for sustainable agriculture in order to address drought, flooding, unpredictable weather, and poor irrigation management. The study modelled the system using a prototype-based IoT approach with ESP32, a soil moisture sensor, a solenoid valve, a wireless network, Blynk IoT, and a mobile-based client-server interface. Real-time sensor readings were captured from the developed ESP32 irrigation prototype to access the ADC value, humidity record, assessed humidity, and error percentage. Results from the test showed an ADC moisture value of 17.74%, a humidity record of 90% RH, an assessed humidity of 90% RH, and a 0% error.

An IoT-based system for monitoring soil water status, including soil water content, matric potential, and hydraulic conductivity, was designed by Comegna et al. (2024) using sensor data: soil water content, matric potential, and temperature, combined for soil hydraulic status estimation. Soil profile monitoring with SHYPROM and cloud-based acquisition was used to gather sensor readings to assess system performance, using the Coefficient of determination (R^2), sensor robustness, and a comparison of TDR devices. The results showed that the system achieved an R^2 = 0.99 for estimating soil moisture content, and has shown to be robust in terms of long-term data acquisition. However, one of the main challenges for the system will be that the capacitive sensor readings may be influenced by the following: the texture of the soil, temperature, salinity, and type of soil.

Table 2.3 Summary of Literature

Author	Technique	Work done	Limitations
Padiachy et al. (2025)	Rule-based threshold and IoT	developed a smart IoT irrigation system to improve water efficiency and agricultural, and results demonstrate water savings and improved crop yield.	there are problems with data security, system integration, and adoption.
Zhang et al. (2025)	IoT and Multimodal data	Designed an IoT-based smart irrigation system using LoRa and edge computing to improve automation, data integration, and decision-making which achieved remarkable communication results.	The system will require high-quality data to integrate ML.
Habib et al. (2023)	IoT framework; cloud-based processing	They designed an IoT-based automated wastewater irrigation system to optimise water usage in water-scarce regions of the Middle East, and a pilot test confirmed the	The efficiency of the system is marred by water scarcity, dependence on infrastructure, and

		system's feasibility, including a cloud-based monitoring dashboard.	environmental constraints.
Munir et al. (2018)	Fuzzy logic, IoT sensors	An IoT-based smart irrigation and energy-saving system for tunnel farming due to water scarcity and inefficient manual irrigation, with results demonstrating greater water savings than previous approaches and manual watering.	The system was tested on a small area and might not be scalable to large fields
Purnama et al. (2024).	IoT	They developed an ESP32-based IoT irrigation control system for sustainable agriculture by modelling it using a prototype-based IoT approach with an ESP32, a soil moisture sensor, a solenoid valve, a wireless network, Blynk IoT, and a mobile-based client-server interface. Results from the test showed an ADC moisture value of 17.74%, a humidity record of 90% RH, an assessed humidity of 90% RH, and a 0% error.	No human in the loop integration.

Comegna et al. (2024)	IoT, SHYPROM	An IoT-based system for monitoring soil water status, including soil water content, matric potential, and hydraulic conductivity, that achieved an $R^2 = 0.99$ for estimating soil water content.	The capacitive sensor readings can be affected by soil texture, temperature, salinity, and soil type.

2.4 Research Gap

From the literature reviewed, most existing IoT systems for large-scale farming were found to lack scalability, fail to optimise water usage effectively, and lack efficient real-time IoT dashboards, both for traditional manual and current systems. A study by Purnama et al. (2024) developed an ESP32-based IoT irrigation control system for sustainable agriculture, but did not incorporate human-in-the-loop. The proposed system will design an IoT-based irrigation system using ESP32 for automated water control based on soil moisture levels, integrating a human-in-the-loop mechanism that allows users to intervene only when necessary.

CHAPTER THREE

SYSTEM ANALYSIS AND DESIGN

3.1 Introduction

This chapter describes the system analysis and design of my proposed IoT-based irrigation system, which involves designing and implementing it using ESP32. It includes system design, hardware and software components, and implementation procedures. It reviews the existing conventional irrigation system and identifies challenges affecting accuracy, speed, and client convenience. The chapter also presents the methodology, system requirements, and design implementation used in the development of the proposed system.

3.2 System Analysis

This section involves understanding the problem domain. It identifies what the current system does and outlines the issues to be solved through conducting an analysis of the existing system and highlighting its weaknesses.

3.1.1 Analysis of the Existing System

The existing system is a conventional irrigation system that employs manual or flooding methods for dispensing water to crops in the farmlands. This system is static and can be erroneous, such that it waters the farm without analysing the current soil moisture content, which can lead to waterlogging or an insufficient water supply for crop growth. However, there are current systems that employ automatic means of irrigation, for instance, a study by Purnama et al. (2024) that developed an ESP32-based IoT irrigation control system for sustainable agriculture. This system was able to perform irrigation activities incorporating soil moisture data for effective and efficient crop watering. But the system lacked features such as human-in-the-loop and weather forecast data, making it unreliable and leading to water losses.

3.1.2 Limitations of the Existing System

i. The traditional system of manual or flooding irrigation lacks an automatic control or relay system.

ii. The manual system of irrigation is physically demanding, as it requires human monitoring and sometimes the manual dispensing of water to crops.

iii. The existing automated irrigation system applies water without considering real-time soil moisture and rainfall conditions, which can lead to waterlogging or insufficient water supply for crop growth.

iv. The existing manual irrigation system does not present continuous information on soil conditions, pump status, and weather changes, thereby making timely irrigation decisions difficult.

v. The existing system of Irrigation without sensor-based control and a weather forecast system can lead to a waste of water

3.1.3 Analysis of the Proposed System

The proposed system is a smart, automated irrigation system for farming, designed using an ESP32 microcontroller, other necessary hardware components, and IoT devices. The new system is complemented with a user-friendly web application that integrates real-time weather forecast updates to the dashboard of the smart automated irrigation system. This weather forecast enables the human-in-the-loop feature to effectively override an irrigation attempt when there is a prediction of possible rainfall in the immediate future. Figure 3.1 presents the circuit diagram of the smart irrigation system, with its components listed and their functions described appropriately in Table 3.1. Figure 3.2 illustrates the implementation diagram of the prototype of this proposed system, showing the listed components being embedded in a breadboard.

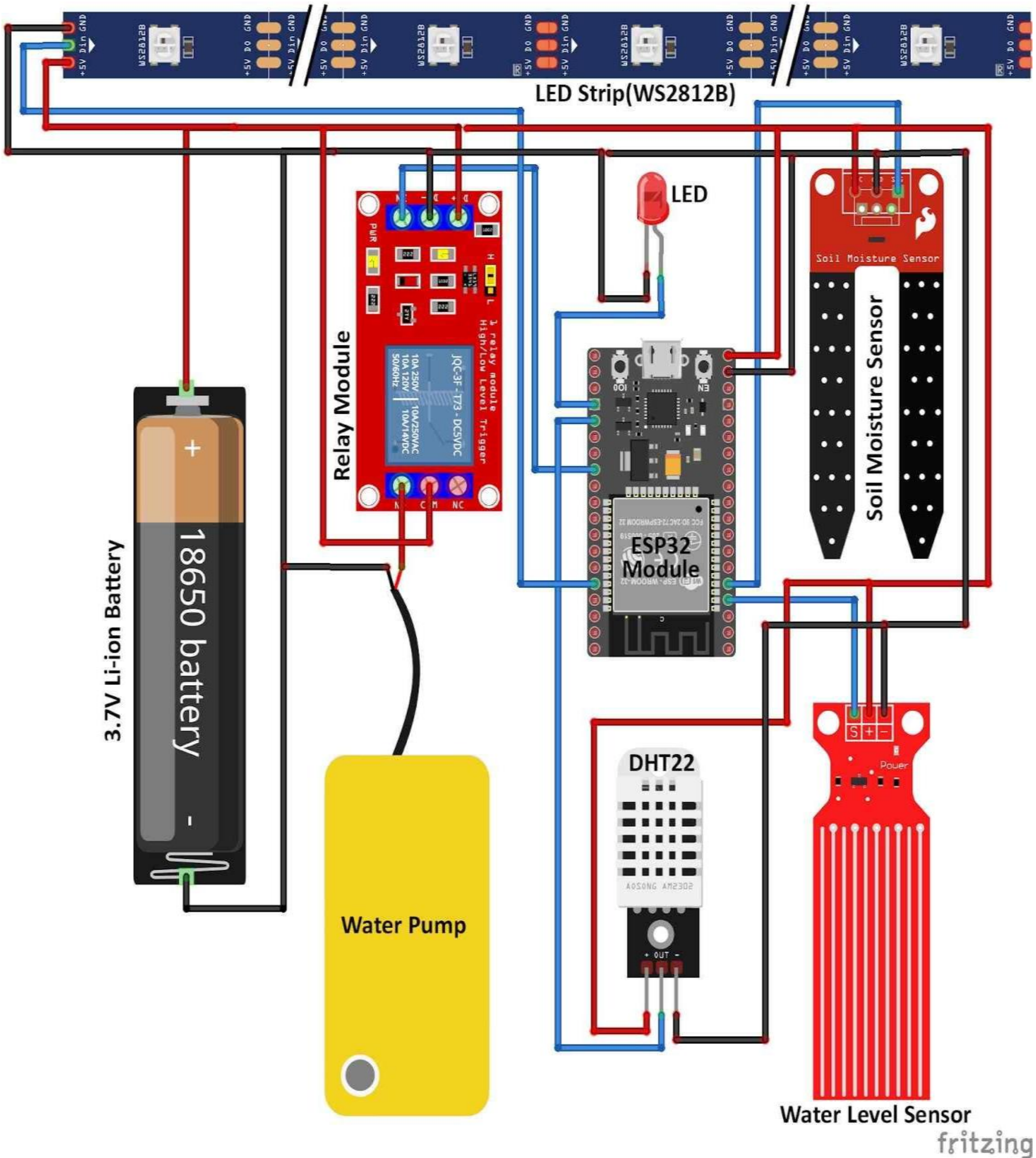

Figure 3.1: Circuit diagram of the smart irrigation system

Table 3.1 Components of the proposed system

Component	Function in the System
ESP32 Module	Acts as the central controller that processes sensor data, makes irrigation decisions, and enables wireless communication via Wi-Fi/Bluetooth for IoT monitoring.
Soil Moisture Sensor	Measures the water content in the soil and sends data to the ESP32 to determine whether irrigation is required.
Water Level Sensor	Detects the level of water available in the tank or reservoir to prevent dry running of the pump and ensure sufficient water supply.
DHT22 Sensor	Measures ambient temperature and humidity, which can influence irrigation decisions and environmental monitoring.
Relay Module	Serves as an electronic switch that allows the low-power ESP32 to control the high-power water pump safely.
Water Pump	Pumps water from the source to the irrigation area when activated by the relay module.
LED (Indicator)	Provides visual feedback indicating system status, such as pump activation or system operation.
WS2812B LED Strip	Used for advanced visual indication (e.g., system status, moisture levels, alerts) through programmable lighting effects.
3.7V Li-ion Battery (18650)	Supplies portable power to the system, enabling it to operate independently of a fixed power source.
Connecting Wires	Establish electrical connections between all components, allowing power flow and signal communication within the system.

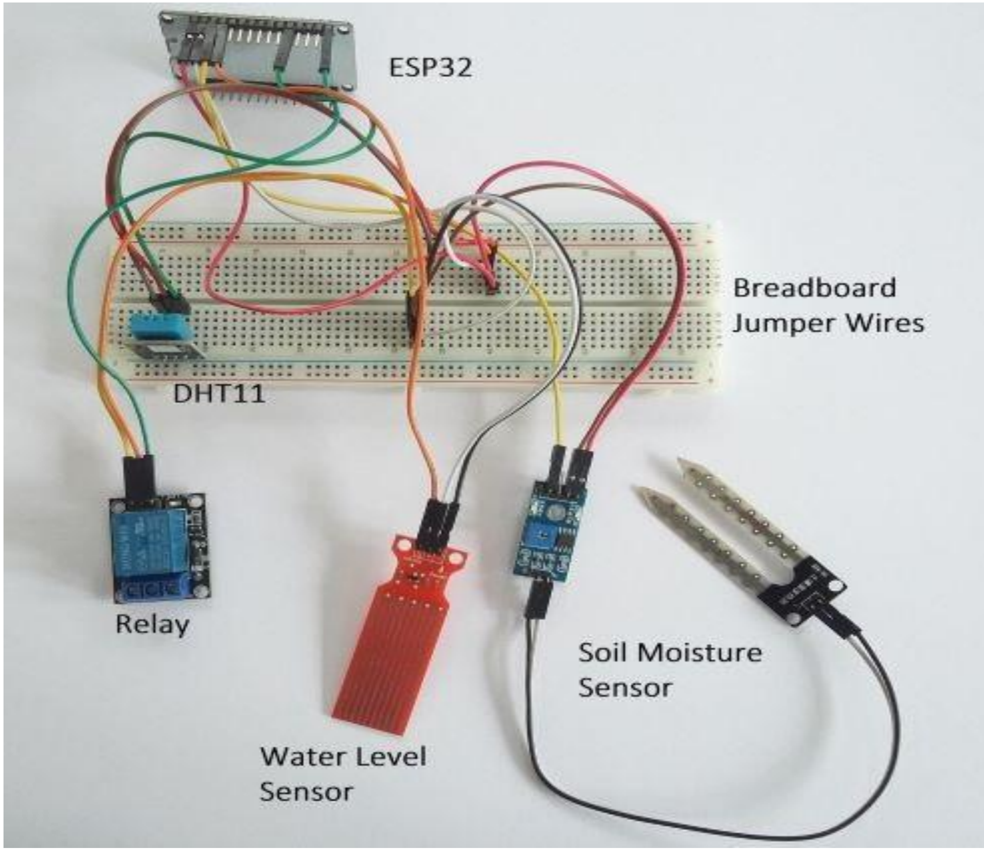

Figure 3.2: Implementation stage of a prototype of the proposed system,

3.2 System Requirements Specification (SRS)

System requirements highlight the specifications necessary for the system to function effectively and meet user needs. These are categorised into functional and non-functional requirements.

Functional Requirements

i. The new system should be able to monitor soil moisture levels in real time using the soil moisture sensor

ii. The system should automatically control the dispensation of water based on the data of the soil moisture conditions.

iii. The new system should allow the user to monitor irrigation status and sensor readings through the dashboard of the web-based interface.

iv. The system should allow the user to manually override or regulate irrigation based on weather updates, which is the human-in-the-loop control feature.
v. The system should be able to activate or deactivate the water pump through the ESP32 controller.
vi. The system should provide clear and readable system status updates, like irrigation ON/OFF and soil conditions, to the user.
vii. The system should prevent irrigation when rainfall is expected within a short time to avoid waterlogging.

Nonfunctional Requirements

i. The system should operate reliably with little or no delays in sensor readings and control actions.
ii. The system should provide near-real-time updates for monitoring and decision-making
iii. The system must be scalable and capable of supporting multiple concurrent users.
iv. The system should be energy-efficient to support steady operation, especially if it is powered by battery sources.
v. The system must provide a responsive user interface compatible with mobile and desktop devices.
vi. The system should ensure high availability and minimal downtime.

3.2.1 System Components

These are the materials and components required to build the system; they may also include components needed for developing the prototype and software to enable system deployment testing. It is categorised into hardware and software components.

A. Hardware Components

i. ESP32
ii. Soil moisture sensor
iii. Relay module
iv. Water pump
v. Power supply

B. Software Components

i. Arduino IDE
ii. Embedded C/C++
iii. IoT platform (Blynk/ThingSpeak)
iv. Python IDE
v. React

3.3 System Design Methodology

A system design methodology is a structured approach or set of guidelines for planning, designing, and organising how a system will be built and how its components will work together. It shapes the researcher's mind and plan by answering questions such as how the system will be structured, what components will be utilised, and how these components will interact in a bid to achieve the purpose of the system.

3.3.1 Proposed Methodology

The methodology for this project, Design and Implementation of a Smart Irrigation System for Farming, is a prototype-based method and Object-Oriented Analysis and Design (OOAD). The study uses a prototype-based experimental design whereby the whole irrigation system design is developed but tested under experimental conditions, including user-accessible high-level structures like the sensors, the ESP32 controller, and a web application/cloud. This methodology describes how the necessary hardware components are selected and incorporated.

In software design, OOAD is used to capture the structure and the behaviour of real-world objects and their interactions in the design of the software system. The OOAD approach improves the modularity, scalability, and code reusability attributes crucial for the development of complex data-driven and AI-integrated systems.

The development process uses Unified Modelling Language (UML) for illustration and Object-Oriented Programming (OOP) principles for implementation for a clear, structured and user-centred system. Such a method can also help to manage the dynamic programming logic, communication with IoT devices, designing web applications, and the user's interactions with the system to control the irrigation process based on weather forecasts. In summary, it presents the data flow design of how data from the sensor flows to the ESP32, then to the cloud-based server and finally to the user interface.

The OOAD methodology consists of the following major phases:

A. **Requirement Analysis:** In this phase, the system features are collected from end users such as landowners, investors, surveyors, and developers. The identified requirements stemmed from the analysis of the gaps in the existing system as well as the expectations from intelligent land valuation aids.

B. **Design of System Architecture:** During this phase, UML diagrams such as use case diagrams, class diagrams, activity diagrams, and sequence diagrams are created to capture the system's static and dynamic aspects.

C. **System Implementation:** The design is executed in OOP languages and respective frameworks. The backend is written in Python (Flask), where weather prediction models will be integrated, while the responsive user interface is designed in React Native for the frontend.

D. **Testing:** Each system module was subjected to both separate and collective tests in order to confirm the would carry out their functionality. Each function undergoes unit testing, and modules undergo integration testing.

E. **Deployment:** Following the testing phase, the system is set up on a live server and is accessible to users through web or mobile applications, and for this project, we will make use of a local host. This step also involves the deployment of the application and the services of the databases onto the secure cloud infrastructure, which ensures accessibility, security and confidentiality

F. **Maintenance:** The system will be continually maintained and updated so that new information, updates to the model, and the latest security measures will be available. The prediction accuracy can be kept at optimal levels by retraining the models regularly.

3.3.5 Unified Modelling Language (UML) diagrams

UML diagrams are employed to conceptualise, document, and plan the architecture and the interaction of the system's components. Using these diagrams, users and researchers will understand the flow of the processes within the system. The Smart IoT-based irrigation system using an ESP32 to automate water dispensing based on soil moisture levels is modelled with the following components:

A. Use Case Diagram for the Proposed System

The use case diagram represents the system's functional requirements and the interactions between actors and system processes. The major actors are the Users, the AI Model, and the Administrator. There are 8 use cases in this system, which include signing in, logging in, input of land details, uploading image, viewing the predicted price and map representation for the user, predict the price and map visualization by the AI model and then the admin manages the user and the predictions to make sure the model is working properly.

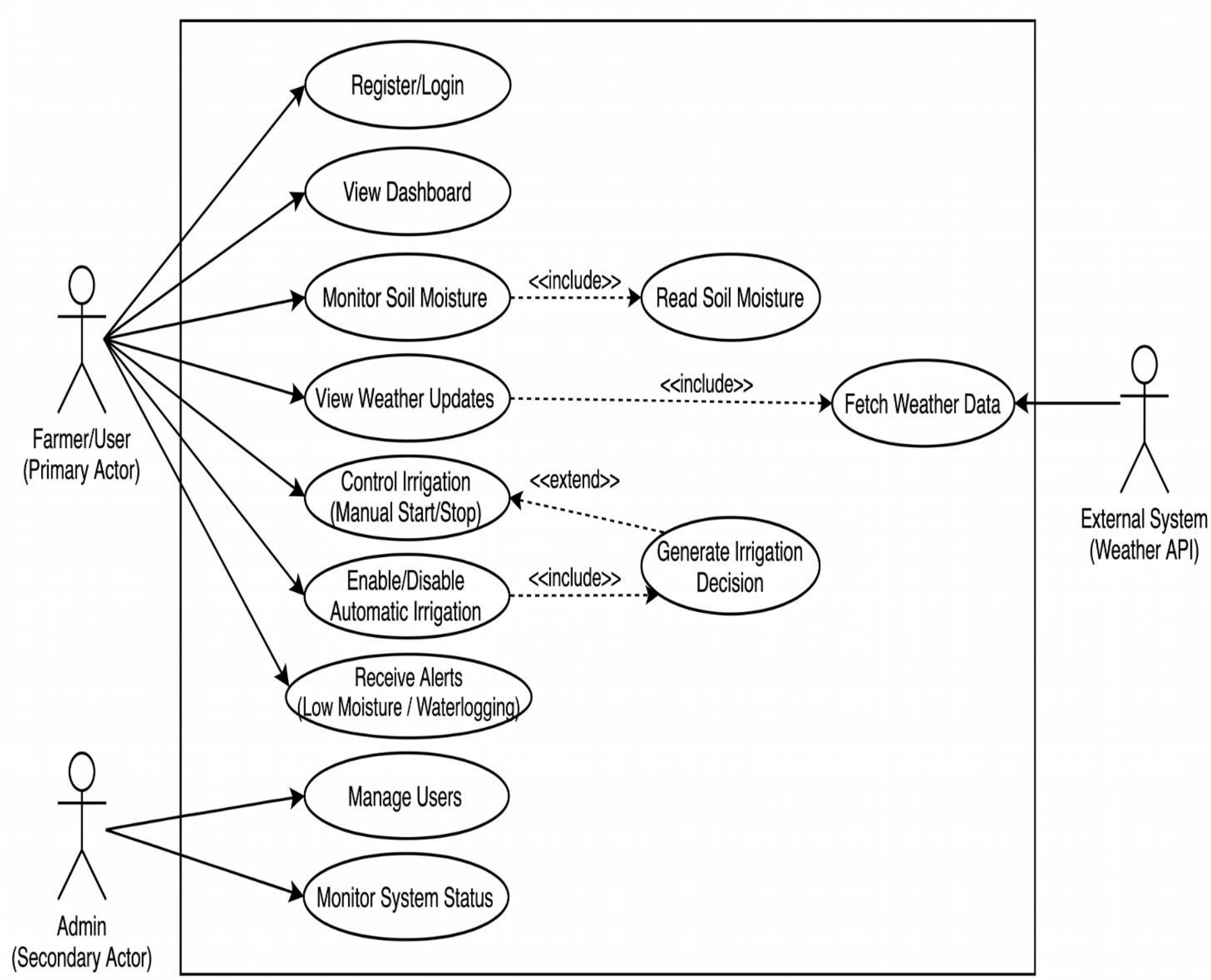

Figure 3.3 Use Case Diagram

The Illustration above shows the proposed system's use case, where the user can engage in several activities that contribute to achieving the goal of the smart irrigation system. The user can perform all the actions in the app, apart from the activities being carried out by the admin and the external system, which is the weather API. The user logs in or signs up, then navigates to the dashboard to monitor soil moisture, view weather updates to enable it to control irrigation (human-in-the-loop), and also receives updates. The admin ensures the app is working properly by monitoring system status and managing and monitoring users' actions within the app.

B. Class Diagram

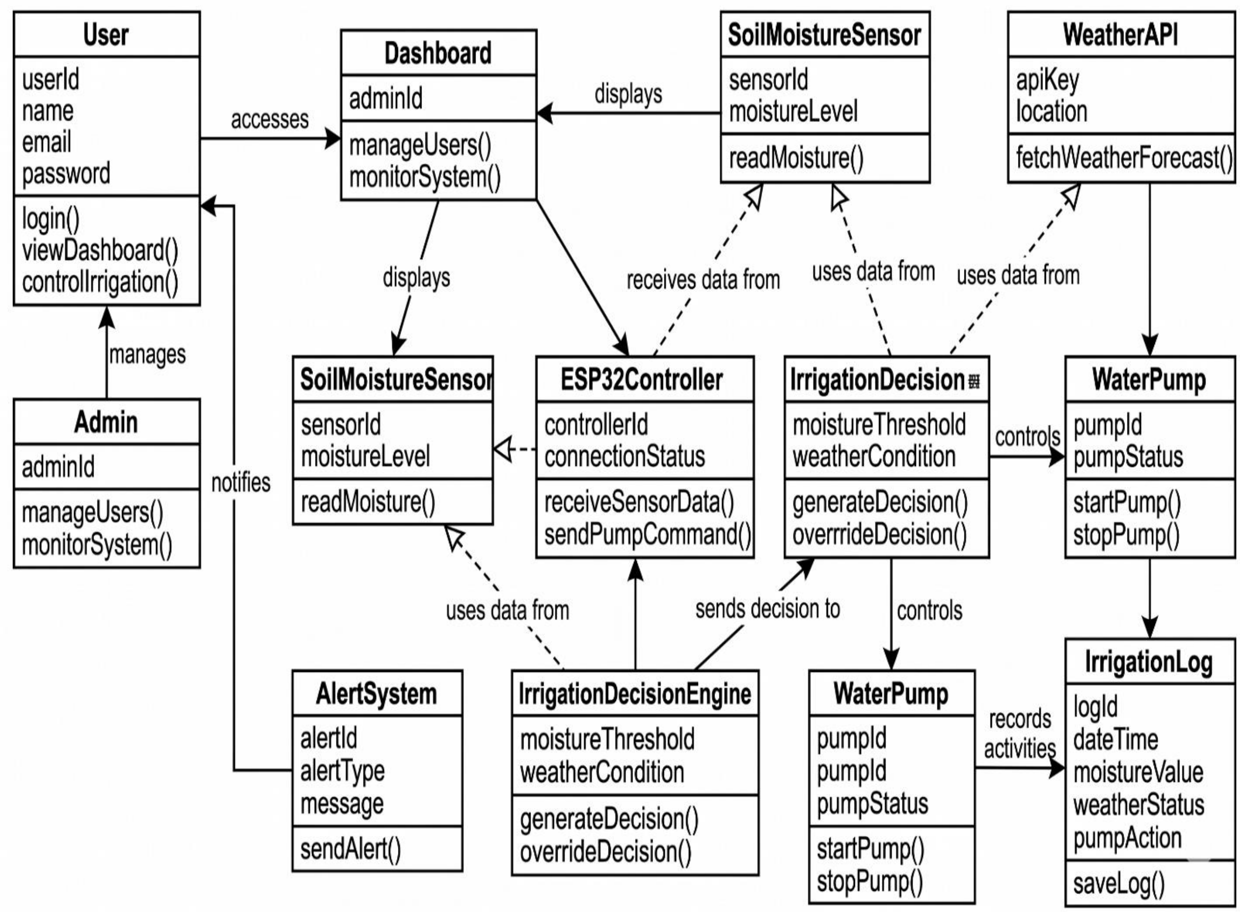

Figure 3.4: Class Diagram

Figure 3.4 presents the class diagram, which includes various classes with attributes and methods. This system consists of 10 classes: users, admin, dashboard, soil moisture sensor, weather API, ESP32 controller, irrigation engine, irrigation log, alert system, and water pump. The User class has several attributes such as user ID, Name, Email, and Password. It also allows users to log in or sign up, view the dashboard to read weather updates and manage the dispensation process. The

Admin Class, representing system administrators, has similar attributes but can also manage users and monitor system performance.

C. Sequence Diagram for the Smart Irrigation App

Figure 3.5 presents the sequence diagram of the smart irrigation app, which is a type of Unified Modelling Language diagram that shows how different parts of a system interact with each other over time. It illustrates who talks to whom, in what order, and at what time during a process. In this sequence diagram, a user logs in to access the dashboard and can request sensor data, prompting the ESP32 controller to read soil moisture. The user can also check the system for weather reports to guide decision-making when necessary. The engine decides on irrigation based on sensor updates, instructing the pump to start or stop, while also incorporating a human-in-the-loop for effective decision-making.

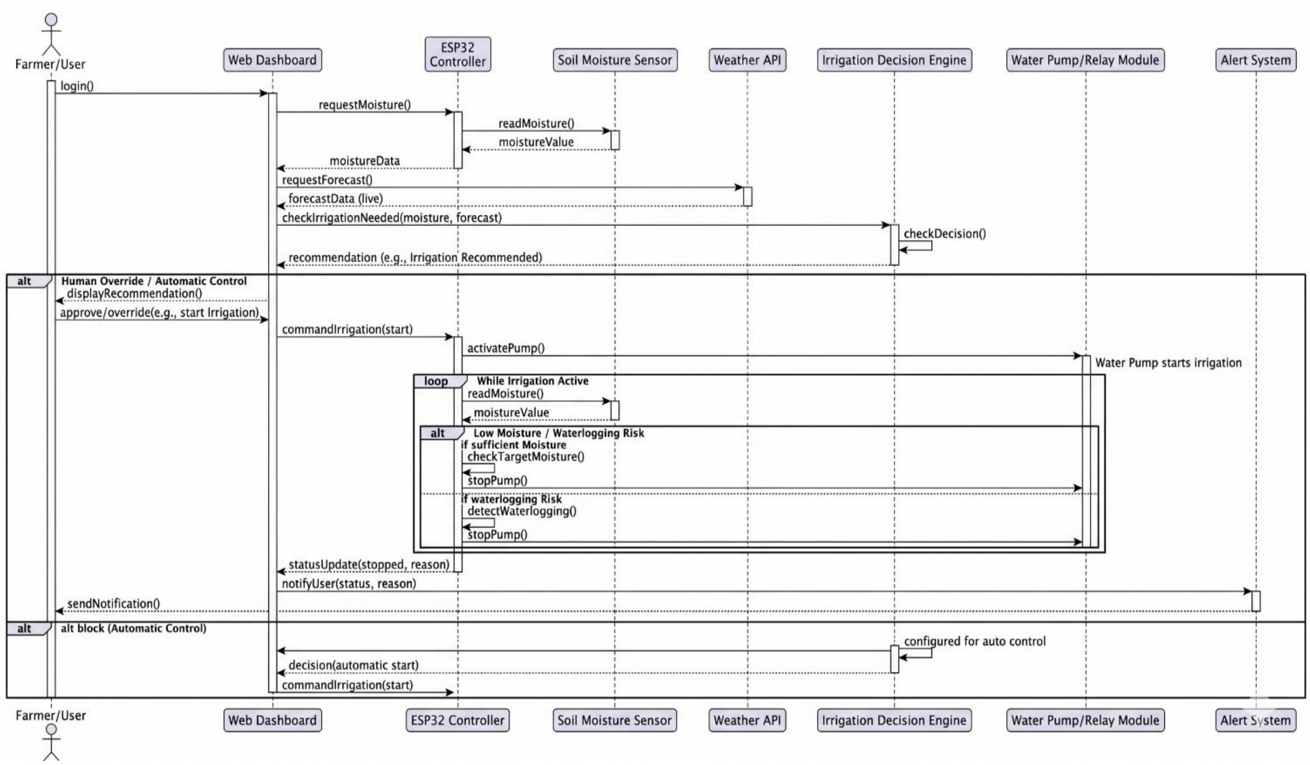

Figure 3.5 Sequence Diagram

3.4 System Flowchart

Figure 3.6 presents the flowchart of the smart automated irrigation system for farming, with the user or farmer needing to sign up or log in to access the dashboard. The system begins by sensing soil moisture, processes the data using ESP32, and triggers irrigation automatically based on predefined conditions.

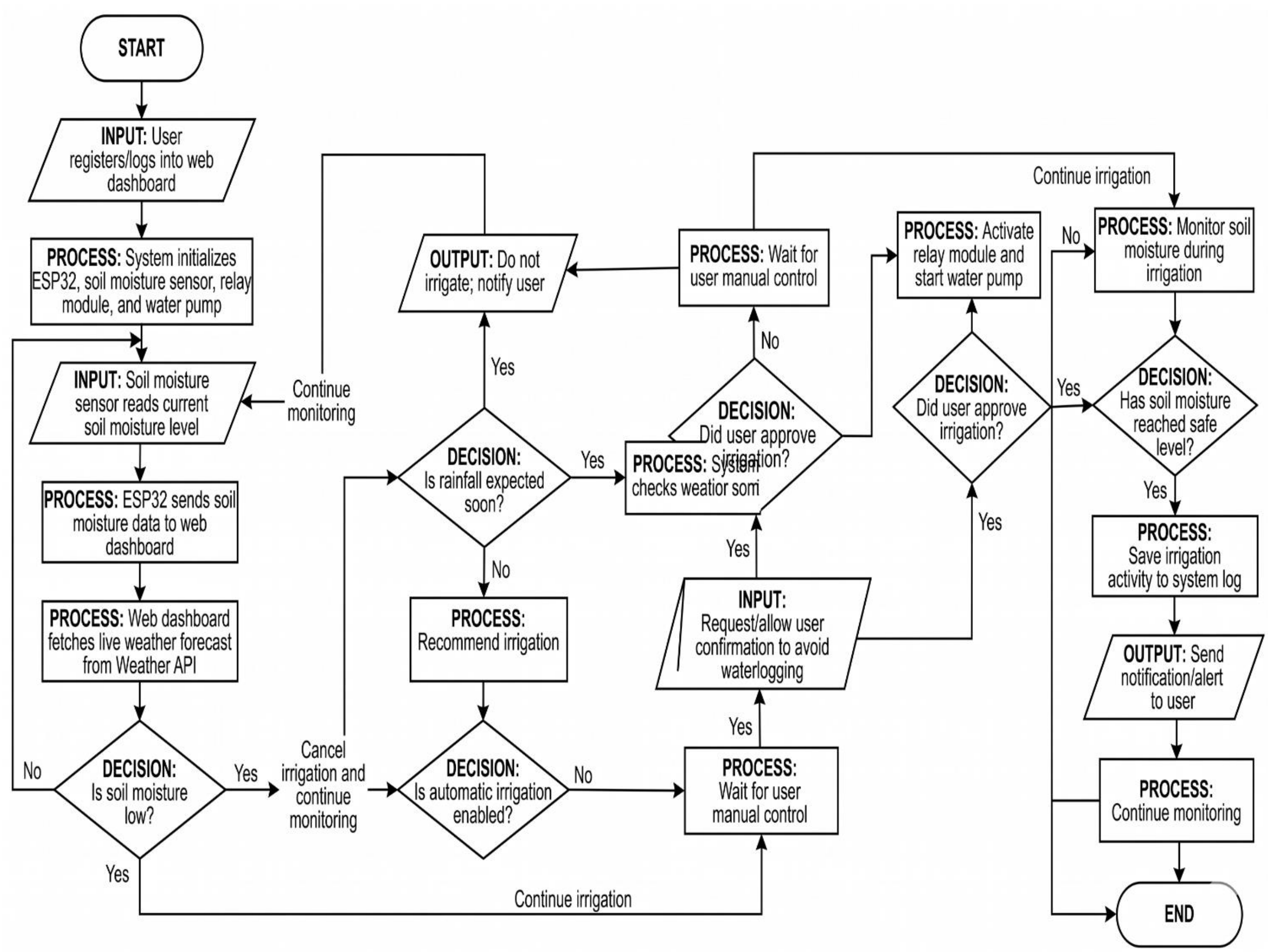

Figure 3.6 System Flowchart

3.5 Database Design

The database is designed to manage users, irrigation activities, sensor readings, weather data, and system decisions, which will ensure efficient storage, retrieval, and monitoring of irrigation operations and, at the same time, support a human-in-the-loop control approach.

User Table

Table 3.2 User Table for Smart Irrigation System

Attribute	Data Type	Description
id	Int	Unique identifier for each user
name	Varchar	Full name of the user
username	Varchar	Username chosen by the user
email	Varchar	User’s email address
password	Varchar	Hashed password for security
role	Enum	Role of the user (Farmer/Admin)
farm_location	Varchar	Location of the farm
profile_image	Varchar	Path to user profile image
is_verified	Boolean	Indicates if the user is verified
date_joined	Timestamp	Account creation time
updated_at	Timestamp	Last profile update time

Soil Moisture Data

Table 3.3: Soil Moisture Table

Attribute	Data Type	Description
id	Int	Unique identifier for each record
sensor_id	Varchar	Identifier for the soil sensor
moisture_level	Float	Soil moisture value (%)
timestamp	Timestamp	Time of data capture

Weather Data

Table 3.4 Weather Data Table

Attribute	Data Type	Description
id	Int	Unique identifier
location	Varchar	Farm location
temperature	Float	Temperature value
humidity	Float	Humidity level
rainfall_prediction	Boolean	Indicates expected rainfall
weather_status	Varchar	Weather condition (e.g., Sunny, Rainy)
timestamp	Timestamp	Time of weather data retrieval

Irrigation Activity

Table 3.5: Irrigation Log Table

Attribute	Data Type	Description
id	Int	Unique identifier
user_id	Int	Reference to the user
moisture_level	Float	Soil moisture at the time of action
weather_condition	Varchar	Weather status during irrigation
irrigation_status	Varchar	Status (Started/Stopped)
mode	Enum	Manual or Automatic
decision_source	Varchar	AI/System or User Override
start_time	Timestamp	Irrigation start time

stop_time	Timestamp	Irrigation stop time

Irrigation Decision Table

Table 3.6: Decision Table

Attribute	**Data Type**	**Description**
id	Int	Unique identifier
moisture_level	Float	Input soil moisture value
weather_forecast	Varchar	Weather condition used
decision	Varchar	Irrigate / Do Not Irrigate
reason	Varchar	Explanation for the decision
user_override	Boolean	Indicates if the user overrode the decision
timestamp	Timestamp	Time of decision

Alert/Notification

Table 3.6: Alert Table

Attribute	**Data Type**	**Description**
id	Int	Unique identifier
user_id	Int	Reference to user
alert_type	Varchar	Type (Low Moisture / Waterlogging)
message	Varchar	Notification message
status	Boolean	Read/Unread status
timestamp	Timestamp	Time alert was sent

CHAPTER FOUR

SYSTEM IMPLEMENTATION

4.1 Introduction

This chapter presents the implementation of the proposed IoT-based smart irrigation system, which uses ESP32 and sensor technologies in its development. It describes the development environment, software tools, hardware components, and implementation procedures adopted by the study. The chapter explains how the hardware and software modules were integrated in order to achieve automated irrigation. Furthermore, it will discuss the testing procedures carried out to validate system functionality and reliability, and then it will present the system outputs, user interfaces, and analysis of the obtained results. In this section, our findings are evaluated against the study objectives to determine the effectiveness of the developed system.

4.2 Development Environment and Tools

During the design of the smart irrigation system, embedded hardware, web technologies and IoT communication tools were employed, and software and hardware resources were also combined to provide real-time sensing and monitoring, decision-making, and irrigation control. The implementation environment was chosen to be both affordable and scalable, and easy to deploy and compatible with IoT devices. To make the project a reality there were significant steps that needed to be taken such as creating software for the device via embedded programming, developing a web application, integrating with a database, and communicating via cloud technologies. All of the selected software tools provided an opportunity to test quickly and build prototypes quickly to demonstrate the proposed irrigation solution.

4.2.1 Programming Language and Libraries

We chose Python as the primary language for this project because of its ease of use and adaptability to the different needs of developing IoT applications. Python is a powerful programming language with many different API and database integration options, which are essential to developing an irrigation system. Python is also easy to maintain and enables developers to quickly build applications. Python also provided the ability to monitor the communication flow between the dashboard, weather services and irrigation control logic.

We developed the software for the ESP32 microcontroller using embedded C/C++. This programming language will give us the ability to efficiently control the hardware directly and to communicate with the sensors connected to the ESP32 microcontroller. The usage of the Embedded C/C++ programming language will allow the microcontroller to execute the irrigation system commands and sensor data quickly with minimal resource usage. Also, using Embedded C/C++ will allow our system to process the sensor data and the irrigation system commands in real-time.

To create the user interface for this system, we used React to create a responsive user interface for the irrigation dashboard. React utilizes components to create a dynamic visualisation of real-time sensor data. The React user interface creates an efficient rendering of real-time sensor data and provides a higher interaction level for the users of the irrigation dashboard than other technologies.

Libraries and Frameworks

A range of libraries were included to develop the system and included Flask, Request, JSON, a Wi-Fi Library, and a number of Sensor Libraries. While Flask served as the backend web framework for API development and as a means of communication between the different parts of the system, it was also utilized to facilitate efficient routing and data exchange. To avoid the hassle of communicating with weather services, the request library was used to get the weather forecast data from the weather service's external API. The data was transmitted and received from the ESP32, backend services, and the dashboard using JSON, which enabled the lightweight and efficient transfer of sensor data. The ESP32 Wi-Fi libraries are used to connect the irrigation system to the internet to enable remote monitoring and control of the irrigation system using a wireless connection. For acquiring data from these sensors (temperature, humidity), the DHT Sensor Libraries were used to calibrate and store the data captured from the DHT sensors. The water level sensor libraries enabled monitoring of the water level of an area.

4.2.2 Development Platform and Hardware Requirements

The development environment was selected in order to provide efficient integration between embedded hardware, IoT communication, and web-based monitoring functionalities for us to effectively implement our system. Arduino IDE and VS Code facilitated firmware programming, backend development, and user interface implementation.

Table 4.1: Development Platforms

Development Platform	Purpose	Justification
Arduino IDE	Programming and uploading firmware to the ESP32 microcontroller	Provides a simple environment for embedded system development, debugging, and serial monitoring of sensor data and irrigation processes
Visual Studio Code (VS Code)	Backend and frontend application development	Supports Python, React, and database management through extensions, while offering debugging and version control

		features that improve development efficiency
Web Browser	Testing and validation of the irrigation dashboard	Enables evaluation of dashboard responsiveness, functionality, and user interaction across different devices and screen sizes

Table 4.2: Hardware Requirements and Components of the Proposed System

Hardware Component	**Function in the System**	**Justification**
Intel Core i5 Processor (or Higher)	Supports software development and testing activities	Provides sufficient computational power for executing development tools, simulation environments, and web services
8 GB RAM (Minimum)	Supports multitasking during development and testing	Ensures smooth execution of IDEs, databases, web servers, and browser-based testing environments
SSD Storage	Stores project files and software resources	Improves system responsiveness, software loading speed, and data access performance

ESP32 Microcontroller	Acts as the central processing and communication unit	Processes sensor readings, executes irrigation logic, and enables Wi-Fi-based IoT communication
Soil Moisture Sensor	Measures soil moisture content	Provides real-time soil condition data required for automated irrigation decisions
DHT22 Sensor	Measures environmental temperature and humidity	Supplies climatic information for environmental monitoring and decision support
Water Level Sensor	Monitors water availability in the reservoir	Prevents irrigation operation when water levels are insufficient
Relay Module	Controls switching of the water pump	Enables safe interfacing between the low-voltage ESP32 and the high-power pump
Water Pump	Delivers water to crops during irrigation	Executes irrigation actions based on automated or manual system decisions

Table 4.3: Summary of Development Environment Specifications

Category	**Specification**

Programming Environment	Arduino IDE, Visual Studio Code
Web Testing Environment	Modern Web Browsers (Chrome, Edge, Firefox)
Processor	Intel Core i5 or Higher
Memory (RAM)	Minimum 8 GB
Storage	SSD Storage
Microcontroller	ESP32
Sensors	Soil Moisture Sensor, DHT22 Sensor, Water Level Sensor
Control Device	Relay Module
Actuator	Water Pump
Connectivity	Wi-Fi (ESP32 Integrated Module)

4.3 System Implementation

The implementation phase was hardware integration with software modules, which use the ESP32; the acquisition of sensor data was then configured, and, through the use of the communication protocols of the IoT, we have created a real-time monitoring function. Moreover, integration of the weather forecast into the web dashboard was done and automatic irrigation decisions were made based on the thresholds set for the soil moisture. The wireless connectivity was via Wi-Fi and the dashboard showed the status of irrigation, weather and environmental data. In addition, a Human-in-the-Loop (HiL) functionality was added to enable manual intervention to avoid waterlogging or underwatering on the farm, and, finally, the relay-controlled pump could be automatically controlled based on irrigation commands issued by the system.

4.3.1 Hardware Implementation

During the hardware implementation, the soil moisture sensor was connected to the ESP32 analogue input pins, while the DHT22 sensor was connected to the ESP32 in order to gather environmental data. The water level sensor was installed within the water reservoir, likewise the relay module, which was connected between the ESP32 and the water pump. The ESP32 communicates with the frontend dashboard through Wi-Fi medium, and a battery power source supplies energy to the prototype. Additionally, LEDs were integrated in order to indicate system operational status, while we made sure that all components were assembled according to the designed circuit architecture.

4.3.2 Software Implementation

For the backend implementation, which is the main communication engine of the system, the services were developed using Python and Flask. APIs were created in order to receive and process sensor readings; likewise, the integration of weather data services through external APIs. The database structures for the project were implemented according to the proposed schema designed conceptually earlier in Chapter Two, and user authentication and dashboard functionalities were developed. The irrigation decision algorithms were implemented based on moisture thresholds, and override functions were added in order to support human-in-the-loop decision-making, while real-time data visualisation modules were integrated into the dashboard.

4.4 System Testing

During and after the training, we performed some subsystem testing in order to affirm the functionality of the different modules. In order to evaluate the performance of the soil moisture module, we conducted soil moisture tests on various types of soils. We compared the output readings from the sensors to the expected moisture levels and confirmed that both wet and dry soil can be detected by this module. Specifically, we conducted multiple requests to verify that the weather module could accurately retrieve the weather forecast; the results obtained included valid predictions of temperature, humidity, and amount of rainfall. Likewise, both the relay and pump portions of the system were tested separately, and both were confirmed to correctly provide the capability of turning irrigation operations on/off. Finally, we evaluated the responsiveness and usability of the dashboard; results strongly show that the dashboard conducted real-time displays of the sensor readings correctly and effectively.

Another phase of testing, integration testing, was conducted where data from the ESP32 controller was successfully transmitted to the sensor system, allowing for uninterrupted communication between the ESP32 and individual sensors. The testing results that were gathered confirmed the successful delivery of all data to the back end and results of these tests showed that real-time updates were successful. In addition, user actions were correctly reflected in the system as soon as those actions were performed.

When all modules were assembled together, the overall system operated successfully; in particular, when the moisture threshold was reached, automatic irrigation began functioning. User override features were also tested successfully. The human override actions successfully helped us to modify irrigation decisions, and weather information effectively influenced irrigation recommendations.

4.5 Results Presentation and Analysis

4.5.1 Sample Output and Interface Screens

A. Login Interface: The login interface authenticated users before granting dashboard access by validating their credentials successfully.

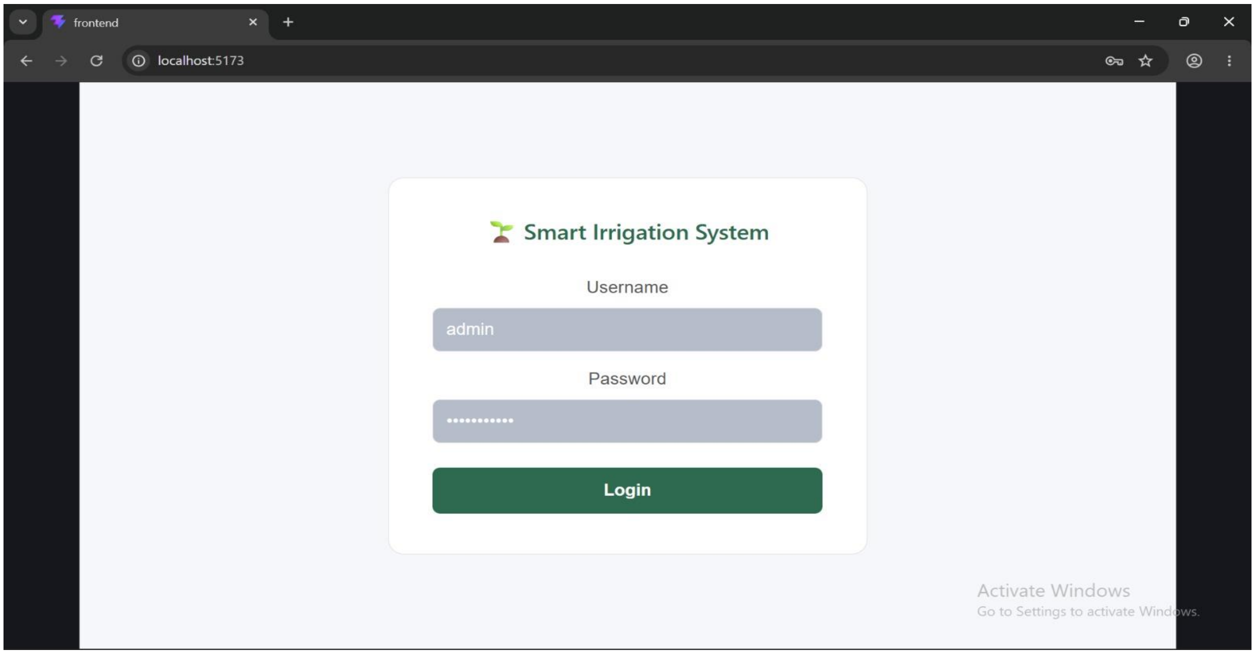

Figure 4.1 Login Page

B. Dashboard Interface: The dashboard displayed soil moisture, temperature, humidity, and irrigation status, helping farmers and other users to monitor environmental conditions remotely.

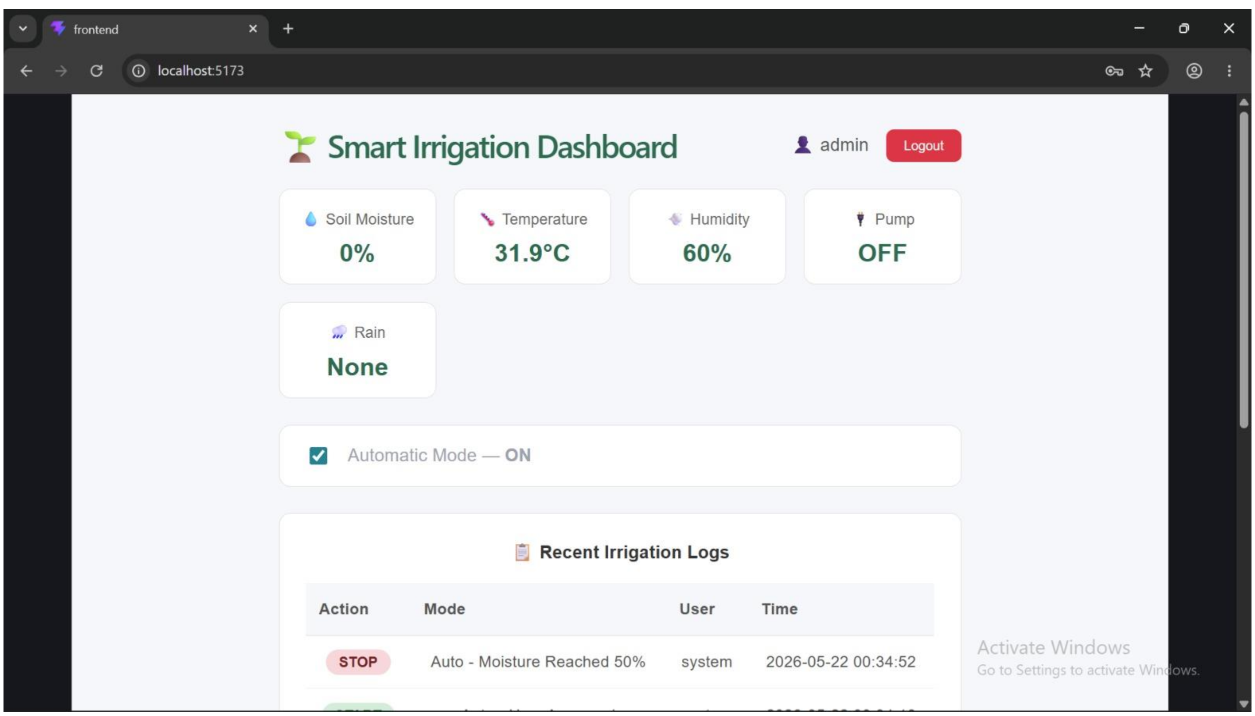

Figure 4.2 User dashboard

C. Human-in-the-Loop Interface: Users could manually enable or disable irrigation operations and override controls, improving flexibility and decision accuracy.

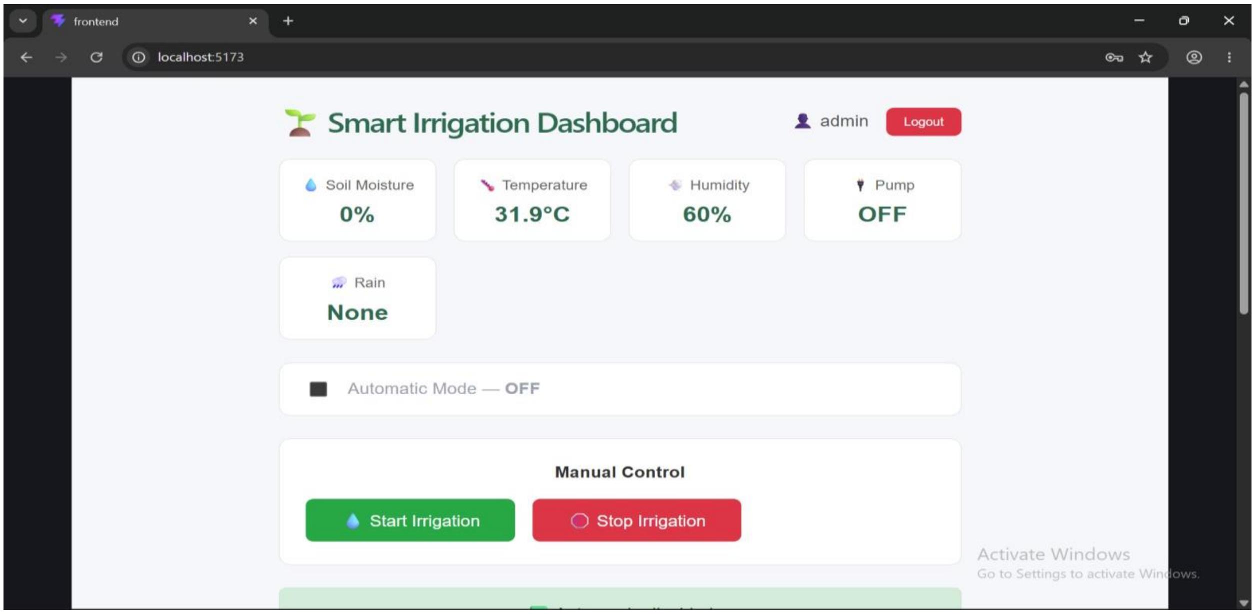

Figure 4.3: Human-in the -loop control

D. History Page: Shows past results or logs of what has been executed before

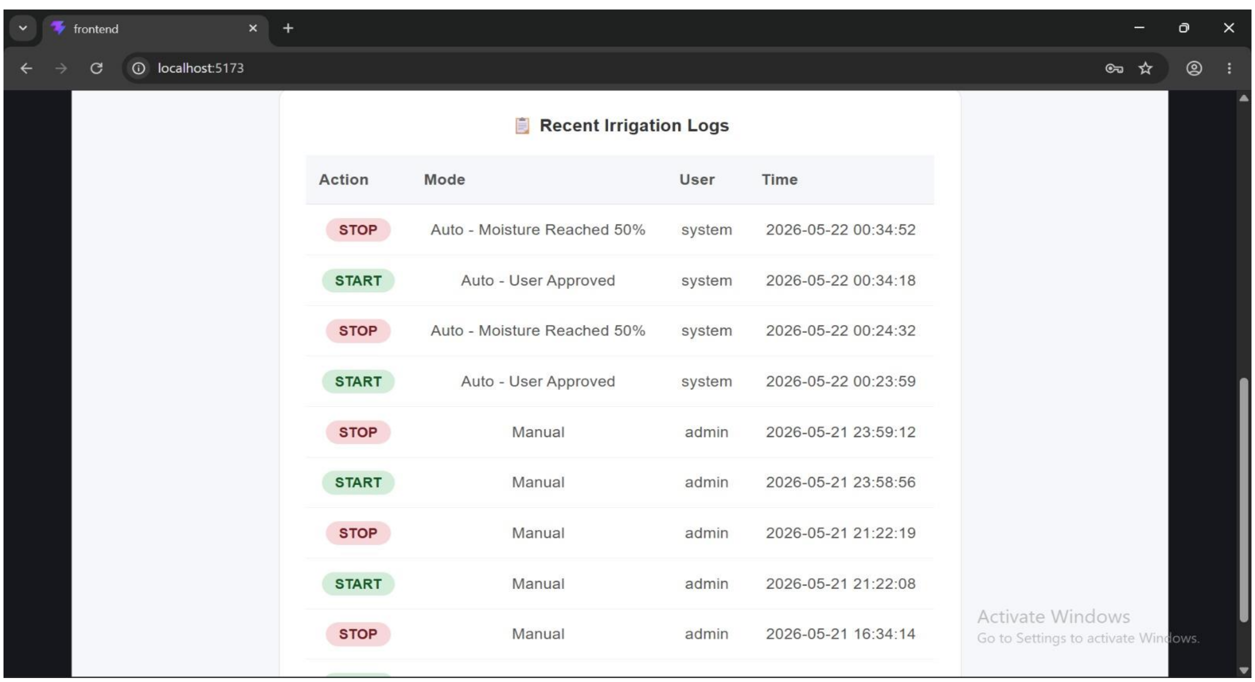

Figure 4.3: History page

4.5.2 Discussion of Results

The system developed is capable of measuring soil moisture in real-time, and has allowed for the automatic irrigation of crops and, therefore, reduces farmers' reliance on manual watering methods. The use of ESP32 microcontrollers provided robust wireless communications. The decision support system, which is aware of weather conditions, improved farmers' ability to plan for irrigation. The addition of human in-the-loop capability allows for further flexibility and control of the system. The visualisation capabilities of the dashboard presented new opportunities for farmers to use the data available to them when making subsequent decisions regarding their farm. The automation of the dispensing of water was successful, and the required amount of water is more efficient through sensor-based irrigation control than what would have been done before. Our proposed solution addressed the significant limitations of the previous irrigation system while also achieving our unique concept of automating irrigation based on soil moisture readings and involving humans. The work completed to date has shown the potential for this technology to support the development of smart agriculture. There are still some limitations to the technology, such as the need for reliable internet connectivity and the need for regular calibration and maintenance of the sensors. In addition, large-scale implementation would require additional field validation and improvement in infrastructure.

CHAPTER FIVE

SUMMARY, CONCLUSION AND RECOMMENDATIONS

5.1 Summary of the Study

The research started with a detailed analysis of the inefficiencies of traditional irrigation methods, focusing on water loss, labor-intensive operations, and limited real-time monitoring. Following this, the objectives and aim of developing an irrigation system using IoT were set, and the importance and scope of the study were clearly determined. A thorough literature review on irrigation systems, IoT technologies, ESP32 microcontroller and soil moisture sensing were performed. The installed smart irrigation systems were then reviewed and compared, and research gaps in current systems were found. This paper review, among others, demonstrated the requirements for a more sophisticated and powerful irrigation system than the current method, which will be weather-aware and human-in-the-loop.

During the design phase, in-depth study was conducted on the existing systems and proposed systems, and functional and non-functional requirements were carefully specified. As a result, hardware and software architectures were designed to satisfy these requirements, UML diagrams were created, and database structures were developed to aid implementation. Prototype-based methodology and Object-Oriented Analysis and Design (OOAD) were used to guide the overall development process. The system has been successfully designed and deployed with the ESP32 microcontroller, sensors, and contemporary web technologies, achieving a comprehensive and integrated system. Following this testing was conducted to confirm system functionality and reliability and proved that monitoring, automation and user control was effective. The developed system met all the aims of the project.

5.3 Conclusion

In this study, an automatic water dispenser system was successfully designed and implemented, which utilized the ESP32 microcontroller with an IoT concept to control irrigation water, continuously monitoring the soil water level in real-time and automatically deciding on the need for irrigation. A dedicated web-based dashboard was used to provide remote monitoring capabilities and the incorporation of weather forecasts further improved irrigation decision making. Moreover, the human-in-the-loop feature allowed users to step in whenever they needed to, adding to the flexibility and control. Consequently, the system was successful in minimizing the constraints that are typical in traditional irrigation systems. The results of the tests verified the feasibility of the solution and proved to be effective as well as completion of the study's purpose and objectives. The prototype developed showcased the feasibility of the IoT technologies in precision agriculture and while there are challenges ahead (such as dependence on internet connectivity and sensor calibration), it offers a solid and reliable backbone to be expanded in the field of smart irrigation.

5.4 Recommendations

i. To enhance the capabilities of the developed system, future work should incorporate machine learning algorithms for predictive irrigation scheduling, while additional sensors, such as rainfall and pH sensors, should also be integrated for more comprehensive environmental monitoring.

ii. In parallel, solar energy systems should be incorporated to improve the overall sustainability of the solution

iii. Future studies should consider adding mobile application support, which should be developed alongside the existing web dashboard, and SMS and email notification functionalities should be added to ensure timely alerts for users.

iv. Cloud-based analytics should additionally be implemented to facilitate long-term farm monitoring, and large-scale field deployment should be conducted to evaluate system performance under varying agricultural conditions.

References

Al-Turjman, F. (2020). *Artificial intelligence in IoT: Smart agriculture and harvesting*. Springer Nature. https://doi.org/10.1007/978-3-030-36939-2

Babu, A. S., Anitha, V., & Sravan, K. (2021). IoT-based smart irrigation system using soil moisture sensor and ESP32. *International Journal of Advanced Research in Engineering and Technology, 12*(5), 458–466.

Bellis, M. (2022). *Farm machinery and technology changes from 1776 to 1990*. https://www.thoughtco.com/americanfarm-tech-development-4083328

Comegna, A., Hassan, S. B. M., & Coppola, A. (2024). Development and application of an IoT-based system for soil water status monitoring in a soil profile. *Sensors, 24*(9), 2725. https://doi.org/10.3390/s24092725

Duguma, A. L., & Bai, X. (2024). Contribution of Internet of Things (IoT) in improving agricultural systems. *International Journal of Environmental Science and Technology, 21*, 2195–2208. https://doi.org/10.1007/s13762-023-05162-7

FloraFlex. (2023). *Irrigation methods: Traditional vs. modern approaches*. https://www.floraflex.com/blogs/floraflex-media/irrigation-methods-traditional-vs-modern-approaches

Froiz-Míguez, I., Lopez-Iturri, P., Fraga-Lamas, P., Celaya-Echarri, M., Blanco-Novoa, Ó., Azpilicueta, L., Falcone, F., & Fernández-Caramés, T. M. (2020). Design, implementation, and empirical validation of an IoT smart irrigation system for fog computing applications based on LoRa and LoRaWAN sensor nodes. *Sensors, 20*(23), 6865. https://doi.org/10.3390/s20236865

Gao, H. R., Zhangzhong, L. L., Zheng, W. A., & Chen, G. F. (2023). How can agricultural water production be promoted? A review on machine learning for irrigation. *Journal of Cleaner Production, 414*, 137687. https://doi.org/10.1016/j.jclepro.2023.137687

Gupta, R., Singh, A., & Verma, S. (2022). Smart irrigation system using ESP32: An IoT approach for water conservation. *IEEE Transactions on Instrumentation and Measurement, 71*, 1–10. https://doi.org/10.1109/TIM.2022.3168853

Habib, S., Alyahya, S., Islam, M., Alnajim, A. M., Alabdulatif, A., & Alabdulatif, A. (2023). Design and implementation: An IoT-framework-based automated wastewater irrigation system. *Electronics, 12*(1), 28. https://doi.org/10.3390/electronics12010028

Institute for Irrigation Research. (2025). *Efficiency comparison between irrigation systems and manual watering*. https://eureka.patsnap.com/report-efficiency-comparison-between-irrigation-systems-and-manual-watering

Koubaa, A., Ahmed, B. M., & Abid, M. (2021). Smart IoT-based irrigation system using ESP32: Architecture and performance analysis. *Sensors, 21*(14), 4820. https://doi.org/10.3390/s21144820

Munir, M. S., Bajwa, I. S., Naeem, M. A., & Ramzan, B. (2018). Design and implementation of an IoT system for smart energy consumption and smart irrigation in tunnel farming. *Energies, 11*(12), 3427. https://doi.org/10.3390/en11123427

Padiachy, V., Prakasan, M., Prasad, K. A., Kumar, E., Nand, A., Kumar, P., & Chand, A. A. (2025). An IoT-based smart irrigation system for sustainable agriculture. In M. K. Nallapaneni & H. Baby (Eds.), *Proceedings of the 2nd International Conference on Smart Innovations in Electrical Engineering (ICSIEE 2025)* (Advances in Engineering Research, Vol. 276). Atlantis Press. https://doi.org/10.2991/978-94-6463-870-7_16

Petropoulos, G. P., Ireland, G., & Lunt, I. (2022). Soil moisture sensors: Technology, selection, and application for precision agriculture. *Agricultural Water Management, 260*, 107285. https://doi.org/10.1016/j.agwat.2021.107285

Purnama, N. A., Abyakto, R., Rosyid, M. G. N., Irfawan, D. R., Soli, M., & Aribowo, D. (2024). IoT-based irrigation control system with ESP32 for sustainable agriculture. *Mars: Jurnal Teknik Mesin, Industri, Elektro dan Ilmu Komputer, 2*(6), 211–224.

RJ Wave. (2025). A comparative study of the traditional irrigation method. *Journal of Agricultural and Food Research, 25*(1), 137–141. https://rjwave.org/jaafr/papers/JAAFR2511371.pdf

Verma, D. K., Mishra, A., & Mishra, K. (2022). Role of IoT in introducing smart agriculture. *International Research Journal of Engineering and Technology, 9*, 883–887.

Zhang, L., Wang, Y., & Zhao, Y. (2023). Intelligent irrigation control based on soil moisture monitoring: A review of recent advancements. *Computers and Electronics in Agriculture, 204*, 107532. https://doi.org/10.1016/j.compag.2022.107532

Zhang, Y., Wang, X., Jin, L., Ni, J., Zhu, Y., Cao, W., & Jiang, X. (2025). Research and development of an IoT smart irrigation system for farmland based on LoRa and edge computing. *Agronomy, 15*(2), 366. https://doi.org/10.3390/agronomy15020366