A Project Report on

SoilStream - A Smart Solar-Powered IoT System for Real-Time Soil Moisture Monitoring and Automated Irrigation

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

Bachelor of Engineering

in

Computer Science and Engineering (Data Science)

by

Raj Choudhary(22107044)
Siddesh Patil(22107031)
Varun Lad(22107043)
Devesh Patil(22107038)

Under the Guidance of

Ms. Rajashri Chaudhari
Ms. Hardiki Patil

Department of Computer Science and Engineering (Data Science)
NBA Accredited
A.P. Shah Institute of Technology
G.B.Road,Kasarvadavli, Thane(W)-400615
UNIVERSITY OF MUMBAI
Academic Year 2025-2026

Approval Sheet

This Project Report entitled ***SoilStream - A Smart Solar-Powered IoT System for Real-Time Soil Moisture Monitoring and Automated Irrigation***, submitted by ***Raj Choudhary (22107044), Siddesh Patil (22107031), Varun Lad (22107043), and Devesh Patil (22107038)***, is approved for the partial fulfillment of the requirements for the award of the degree of ***Bachelor of Engineering*** in ***Computer Science and Engineering (Data Science)*** from ***University of Mumbai***.

(Ms. Hardiki Patil)
Co-Guide

(Ms. Rajashri Chaudhari)
Guide

Dr. Pravin Adivarekar
HOD, Computer Science and Engineering(Data Science)

Place:A.P.Shah Institute of Technology, Thane
Date:

CERTIFICATE

This is to certify that the project entitled ***"SoilStream - A Smart Solar-Powered IoT System for Real-Time Soil Moisture Monitoring and Automated Irrigation"*** submitted by ***"Raj Choudhary" (22107044),"Siddesh Patil" (22107031), "Varun Lad" (22107043),"Devesh Patil" (22107038)*** for the partial fulfillment of the requirement for award of a degree ***Bachelor of Engineering*** in ***Computer Science and Engineering (Data Science)***,to the University of Mumbai,is a bonafide work carried out during academic year 2025-2026.

(Ms. Hardiki Patil)
Co-Guide

(Ms. Rajashri Chaudhari)
Guide

Dr. Pravin Adivarekar
HOD, Computer Science and Engineering(Data Science)

Dr. Uttam D.Kolekar
Principal

External Examiner(s)

1.

2.

Internal Examiner(s)

1.

2.

Place:A.P.Shah Institute of Technology, Thane
Date:

Acknowledgement

We have great pleasure in presenting the report on **SoilStream - A Smart Solar-Powered IoT System for Real-Time Soil Moisture Monitoring and Automated Irrigation.** We take this opportunity to express our sincere thanks towards our guide **Ms.Rajashri Chaudhari** & Co-Guide **Ms.Hardiki Patil** for providing the technical guidelines and suggestions regarding line of work. We would like to express our gratitude towards his constant encouragement, support and guidance through the development of project.

We thank **Dr.Pravin Adivarekar** Head of Department for his encouragement during the progress meeting and for providing guidelines to write this report.

We express our gratitude towards BE project co-ordinator **Dr.Veena Trivedi**, for being encouraging throughout the course and for their guidance. We also thank the entire staff of APSIT for their invaluable help rendered during the course of this work. We wish to express our deep gratitude towards all our colleagues of APSIT for their encouragement.

Raj Choudhary
(22107044)

Siddesh Patil
(22107031)

Varun Lad
(22107043)

Devesh Patil
(22107038)

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(Signature)
Raj Choudhary(22107044)

(Signature)
Siddesh Patil(22107031)

(Signature)
Varun Lad(22107043)

(Signature)
Devesh Patil(22107038)

Date:

Abstract

SoilStream - A Smart Solar-Powered IoT System for Real-Time Soil Moisture Monitoring and Automated Irrigation is an innovative, smart, solar-powered IoT-based agricultural system designed to address some of the most pressing challenges in modern farming, including inefficient water management, unpredictable weather conditions, and the growing threat of crop diseases. The system leverages IoT sensors to continuously monitor real-time parameters such as soil moisture, temperature, and humidity, automating irrigation through precision control to ensure crops receive the exact amount of water they need. This not only reduces water wastage but also promotes soil health and sustainable farming practices. To enhance decision-making, the platform integrates a machine learning model that predicts rainfall probability by analyzing weather patterns and historical data, enabling farmers to optimize irrigation schedules and reduce dependency on external forecasts. Additionally, SoilStream features an api based recommendation engine that provides crop-specific soil improvement suggestions, including fertilizer requirements, nutrient balance, and pH adjustments, helping farmers make informed decisions for higher productivity and improved crop quality. A key highlight of SoilStream is its diagnostic tool, which allows farmers to upload crop images for automated analysis using advanced api to detect pests, diseases, and nutrient deficiencies. The system then suggests eco-friendly treatments and preventive measures, reducing crop losses and minimizing reliance on chemical pesticides. By seamlessly integrating IoT, predictive analytics, and AI-driven diagnostics, SoilStream creates a holistic smart farming ecosystem that delivers accurate, actionable insights, minimizes the misuse of critical resources, and enhances crop yield, resilience, and overall farm profitability.

Index Terms— AIoT, Smart Irrigation, Precision Agriculture, Internet of Things (IoT), Machine Learning, Automated Irrigation, Rainfall Prediction, Crop Identification, Sustainable Farming, Solar Powered Systems.

Contents

1 Introduction **1**
- 1.1 Motivation . . . 1
- 1.2 Problem Statement . . . 2
- 1.3 Objectives . . . 3
- 1.4 Scope . . . 4

2 Literature Review **5**

3 Project Design **7**
- 3.1 Existing System . . . 8
- 3.2 Proposed System . . . 8
  - 3.2.1 Critical Components of System Architecture . . . 10
- 3.3 System Diagrams . . . 11

4 Project Implementation **16**
- 4.1 Code Snippets . . . 16
- 4.2 Steps to Access the System . . . 25
- 4.3 Implementation Images . . . 28
- 4.4 Timeline Chart . . . 32

5 Result and Discussions **34**

6 Testing **37**
- 6.1 Software Testing . . . 37
- 6.2 Functional Testing . . . 39

7 Conclusion **41**

8 Future Scope **42**

Bibliography **43**

Publication **48**

List of Figures

3.1 UML Class Diagram of AIoT Smart Irrigation & Crop Management System 12
3.2 Activity Diagram showing automated irrigation workflow 13
3.3 Use Case Diagram depicting farmer interactions and system functionalities . 14
3.4 Sequence Diagram representing interaction flow between user, app, and backend 15

4.1 Library Inclusions and Pin Configuration 16
4.2 Setup – Sensor and Wi-Fi Initialization 17
4.3 Loop - Sensor Reading and Real-Time Monitoring 18
4.4 Firebase Cloud Function for Rainfall Prediction 19
4.5 Rainfall Prediction Logic 20
4.6 Smart Irrigation Decision Logic 21
4.7 Manual Pump Override from Frontend 22
4.8 Open Camera or Gallery 23
4.9 Show Image Picker Option 23
4.10 Analyze Crop via Flask Backend 24
4.11 AI Crop Identification Interface in the mobile app. 25
4.12 Pest and Disease Diagnosis Interface, showing the photo upload screen. . . . 26
4.13 Smart Irrigation and Pump Control Interface showing real-time pump status and manual override option. 27
4.14 System Interface : Dashboard and Monitoring Views 28
4.15 Mobile Application – Plant Diagnosis 29
4.16 Mobile Application – Crop Analysis 30
4.17 Hardware Implementation Setup : 31
4.18 Project Timeline – Gantt Chart showing project phases and their completion schedule. 33

List of Tables

5.1 Comparison of Model Performance Metrics 34
5.2 Rainfall occurrence prediction performance on Dataset-1 (Mumbai + Ratnagiri) 35
5.3 Rainfall occurrence prediction performance on Dataset-2 (Mumbai + Ratnagiri + Thane + Pune) 35

6.1 Functional Testing Table for SoilStream System 39

List of Abbreviations

IoT:	Internet of Things
AI:	Artificial Intelligence
AIoT:	Artificial Intelligence of Things
ML:	Machine Learning
API:	Application Programming Interface
CNN:	Convolutional Neural Network
VGG16:	Visual Geometry Group, 16 layers
ResNet50:	Residual Network with 50 layers
EfficientNetB0:	Efficient Network Baseline

Chapter 1

Introduction

Agriculture today faces multiple critical challenges such as inefficient water management, unpredictable rainfall, and recurring crop diseases that directly impact productivity and resource sustainability. Traditional farming methods, including manual irrigation, basic weather updates, and conventional pest control, often rely on human judgment and experience rather than real-time data. These practices are not only time-consuming and labor-intensive but also prone to inaccuracies, leading to over-irrigation, water wastage, and delayed responses to plant health issues.

To address these issues, we conceptualized and developed SoilStream - A Smart Solar-Powered IoT System for Real-Time Soil Moisture Monitoring and Automated Irrigation, an innovative, solar-powered smart agriculture system that integrates Internet of Things (IoT), Machine Learning (ML), and Artificial Intelligence (AI) technologies. The system continuously monitors real-time soil parameters such as moisture, temperature, and humidity through IoT-based sensors deployed in the field. This data is transmitted to a cloud platform, where it is analyzed to automate irrigation intelligently, ensuring optimal water utilization and energy efficiency.

Additionally, SoilStream incorporates rainfall prediction using machine learning algorithms, allowing farmers to make proactive decisions about irrigation scheduling and water conservation. The system further employs a api based recommendation engine that provides crop-specific soil management advice, helping farmers maintain the right soil conditions for maximum yield. Beyond irrigation management, SoilStream offers a api based module capable of diagnosing pests and plant diseases from farmer-uploaded images. By leveraging this api, the system identifies potential infections early and suggests suitable remedies, reducing crop loss and improving farm health.

In essence, SoilStream transforms traditional, reactive farming practices into a data-driven, automated, and intelligent agricultural ecosystem. It empowers farmers with real-time insights, predictive analytics, and AI-based decision support, promoting sustainability, efficiency, and resilience in modern agriculture.

1.1 Motivation

Our motivation stems from the urgent need to combat the persistent challenges of water scarcity, unpredictable weather, and crop vulnerability faced by farmers in the Thane region. Agriculture in this area, much like in many parts of India, continues to depend heavily on traditional irrigation methods and manual observation, which often result in inefficient

water use, delayed decision-making, and reduced crop yields. These challenges are further aggravated by climate change, leading to irregular rainfall patterns and increased susceptibility to pests and diseases. Small and medium-scale farmers, who form the backbone of local agriculture, are often unable to afford sophisticated systems or access timely agricultural insights.

To address these issues, we are driven by the vision of empowering local agriculturists through an affordable, intelligent, and sustainable technological solution. Our goal is to develop a low-cost AIoT-based smart farming system that makes precision agriculture practical and accessible for every farmer, regardless of scale. The system aims to transform raw environmental data—such as soil moisture, humidity, and temperature—into actionable insights that guide irrigation scheduling, water usage, and crop management decisions.

By combining Artificial Intelligence (AI), Machine Learning (ML), and Internet of Things (IoT) technologies, the system delivers real-time intelligence directly to the farmer's fingertips. It predicts localized rainfall, provides crop-specific soil health recommendations, and instantly diagnoses pests and diseases through api-based image analysis. This empowers farmers to take proactive and data-driven actions, reducing reliance on intuition or manual guesswork.

Ultimately, our motivation lies in bridging the technological divide between modern precision farming and traditional agriculture. We aspire to ensure that advanced, data-driven agricultural practices are not a luxury but a necessity accessible to all farmers, helping them conserve water, increase productivity, and build resilience against climate variability. Through this initiative, we aim to contribute to a future where technology and sustainability coexist, securing both farmer livelihoods and environmental well-being.

1.2 Problem Statement

SoilStream - A Smart Solar-Powered IoT System for Real-Time Soil Moisture Monitoring and Automated Irrigation project aims to tackle some of the most pressing issues in modern agriculture—inefficient irrigation, unpredictable weather conditions, and delayed pest or disease detection—which collectively result in significant water wastage and reduced crop productivity. Traditional farming practices often rely on manual decision-making and irregular observation, leading to over-irrigation or under-irrigation, both of which negatively impact crop health and soil quality. Moreover, climate variability and inconsistent rainfall patterns make it increasingly difficult for farmers to plan irrigation schedules and optimize resource usage effectively.

To address these challenges, the goal of the SoilStream project is to develop an intelligent, solar-powered IoT-based agricultural system that brings automation, precision, and data-driven intelligence into the farming process. The system integrates real-time soil moisture monitoring using IoT sensors to ensure that irrigation decisions are based on actual field conditions rather than estimations. Through machine learning-based rainfall prediction, the system helps farmers anticipate weather patterns and plan irrigation cycles accordingly, minimizing water wastage and improving crop resilience.

In addition, SoilStream incorporates automated irrigation control, allowing the system to adjust water distribution autonomously based on sensor inputs and predicted weather data. This not only enhances water efficiency but also reduces manual effort and dependency on constant human supervision. Furthermore, an api-driven crop disease diagnosis module enables the system to analyze farmer-uploaded images and identify potential pest infestations

or plant diseases at an early stage, thereby supporting timely intervention and preventing large-scale yield losses.

Ultimately, the SoilStream project addresses the critical intersection of sustainability and agricultural productivity. By combining IoT, Machine Learning, and AI technologies, the system empowers farmers with real-time insights and automated decision support, promoting smarter resource utilization, improving yield quality, and fostering environmentally responsible farming practices.

1.3 Objectives

The primary objective of the SoilStream system is to revolutionize modern farming practices by leveraging Artificial Intelligence (AI) and the Internet of Things (IoT) to create a smart, self-sustaining agricultural ecosystem. The project aims to empower farmers—particularly in water-scarce and climate-sensitive regions—by providing data-driven insights, automated irrigation, and AI-powered crop management. Through the integration of real-time sensing, predictive analytics, and intelligent automation, SoilStream strives to enhance productivity, optimize water usage, and ensure sustainable agricultural practices. The following objectives outline the key features and technological components that collectively contribute to achieving these goals:

- **Real-Time Soil Moisture Monitoring through IoT Sensors:**
  The system continuously monitors soil moisture levels using IoT-based sensors deployed across the farmland. These sensors collect and transmit live data to the cloud, enabling precise tracking of soil health and irrigation needs. By eliminating manual observation, this objective ensures optimal watering cycles, prevents over-irrigation, and maintains consistent soil conditions conducive to plant growth. Farmers can access this real-time data through the mobile app interface, allowing them to make informed irrigation decisions anytime, anywhere.

- **Machine Learning-Based Rainfall Prediction for Efficient Water Management:**
  To minimize water wastage and improve irrigation planning, SoilStream integrates a machine learning model trained on historical and live weather data to predict rainfall probability. These predictive insights allow the system to automatically adjust irrigation schedules—delaying watering if rain is imminent or triggering irrigation during dry spells. This smart forecasting not only conserves water but also reduces dependency on unreliable weather forecasts, contributing to sustainable resource management.

- **Api-Based Crop-Specific Soil-Water Recommendations:**
  To generate intelligent, crop-specific recommendations—such as optimal watering frequency, soil nutrient management, and irrigation duration—to enhance crop productivity. By leveraging Gemini's advanced reasoning capabilities, the platform will offer context-aware and data-driven insights tailored to each crop's requirements.

- **AI-Powered Pest and Disease Diagnosis using Kindwise api:**
  To employ the Kindwise API for automated pest and disease diagnosis through image analysis of farmer-uploaded crop photos. The system will use AI-based image recognition to identify visible signs of pest infestation or disease infection and provide precise treatment recommendations. These include suitable pesticides, organic remedies, and preventive actions, all delivered directly within the application, helping farmers take quick and effective measures to protect their crops.

1.4 Scope

- Can be applied in various agricultural environments such as small-scale farms, organic cultivation, and research plots, providing comprehensive precision irrigation and crop monitoring solutions that promote sustainable farming practices.

- Caters to small-scale farmers, agricultural students, and organic farming enthusiasts, offering tailored soil monitoring and irrigation automation that adapts to different expertise levels, climatic conditions, and crop types. The system's modular design allows it to be easily integrated into existing farm setups without the need for complex technical infrastructure.

- With real-time monitoring, predictive analytics, and AI diagnostics, SoilStream reduces water wastage and makes precision agriculture accessible for farmers seeking sustainable yield optimization. Its scalability enables future enhancements such as soil nutrient analysis, pH detection, and automated fertilizer recommendation systems, further broadening its utility across diverse agricultural domains.

- Additionally, the platform supports data-driven decision-making by enabling farmers to track long-term soil health and climatic patterns, ensuring better crop planning and resource management over time.

Chapter 2

Literature Review

The integration of Artificial Intelligence (AI) and Internet of Things (IoT) technologies into agriculture has gained significant attention in recent years, primarily to address challenges related to water management, crop health, and sustainability. The literature demonstrates a consistent evolution from basic sensor-based automation to the intelligent, AI-driven IoT frameworks seen today. This survey explores this progression by examining past and present trends, from foundational prototypes to modern, fully-integrated systems.

The evolution of smart irrigation began with foundational systems focused on basic automation and remote monitoring. Early prototypes, such as those by Rajalakshmi and Mahalakshmi [15], demonstrated the use of IoT for basic crop field monitoring and irrigation automation. Rawal [14] furthered this by developing a system that automated water distribution based on soil moisture levels, though it remained dependent on external power. Similarly, Casado et al. [12] explored Arduino-based systems for automating water delivery. These initial systems were critical in proving the concept of automated irrigation but were generally limited to prototype-level implementations. They often lacked wireless IoT integration, energy efficiency, and any form of predictive analytics, which restricted their scalability and real-world applicability.

A significant subsequent trend was the integration of renewable energy to address the power dependency and sustainability of these systems. Rout and Mishra [13] pioneered one of the first solar-powered smart irrigation systems using IoT, marking an early step toward energy-efficient agriculture. This focus on sustainability was continued by Devan et al. [10], who presented a proof-of-concept IoT-based solar model, and Ramli and Jabbar [9], who introduced a portable, solar-powered prototype designed to improve flexibility for small farms. Ali et al. [6] also designed an IoT and solar-based system focused on affordability, though its validation was limited to a small-scale setup. Further explorations by Maskara et al. and Al-Zahrani and Al-Baity also emphasized solar-powered IoT solutions but highlighted the persistent challenges of poor connectivity and a lack of real-time monitoring capabilities in many rural regions.

The next major evolutionary leap was the integration of Artificial Intelligence (AI) and Machine Learning (ML) to move from simple automation to predictive, intelligent control. Talaviya et al. [11] discussed the potential of AI for optimizing both irrigation and pesticide applications, noting that high operational costs and the need for skilled operators were significant barriers. This potential was practically demonstrated by Tace et al. [8], who used machine learning for predictive water management based on IoT sensor data. Esmail et al. [7] also implemented ML models for predictive irrigation control, revealing that model

accuracy was heavily dependent on the quality of the dataset. A common theme in these studies was that while AI integration showed great promise, most systems were tested in controlled environments, leaving scalability and adaptability to diverse climatic conditions as open challenges.

Current, state-of-the-art research focuses on creating fully-integrated platforms that combine all three pillars: IoT sensing, solar power, and advanced AI. For instance, Balamurali et al. [1] presented a solar-powered IoT system that integrates rainfall forecasts supported by aerosol data to optimize water usage. Liu, Zhao, and Rezaeipanah [2] developed an intelligent framework using fuzzy control technology for enhanced precision. Al Mamun et al. [3] also proposed an IoT-enabled, solar-powered system, though it faced limitations with battery backup. Capcha-Ochoa et al. [4] extended this concept by integrating ML (modified YOLOv3-tiny) for real-time plant health detection alongside IoT and solar power. Kunt [5] also developed an AI-based model for intelligent decision-making. While these modern systems are highly innovative, they introduce new challenges. The drawbacks most frequently noted are high system complexity and cost [4], significant computational demands [2], [5], and a critical dependency on stable network connectivity and high-quality data for reliable AI predictions [1], [3].

Overall, the literature demonstrates a consistent evolution from basic sensor-based automation to intelligent, AI-driven IoT frameworks. While significant progress has been made in improving irrigation precision and sustainability, challenges such as scalability, affordability, real-world adaptability, and energy efficiency persist. The proposed system in this project builds upon these research foundations by integrating solar power, IoT sensing, and AI-based predictive models into a single, cost-effective platform tailored for local farmers in the Thane region. This approach bridges the gap between technological innovation and regional applicability, ensuring both practicality and accessibility.

Chapter 3

Project Design

This chapter presents a comprehensive overview of the system design for the SoilStream-project — an intelligent, solar-powered IoT-based agricultural solution developed to enhance efficiency, sustainability, and precision in farming practices. The proposed system addresses the limitations of traditional agricultural methods prevalent in the Thane region, where irrigation is primarily based on fixed schedules or manual judgment, often resulting in excessive water consumption or inconsistent watering patterns.

The SoilStream architecture integrates multiple advanced technologies, including IoT-based soil moisture sensing, machine learning-driven rainfall prediction, automated irrigation control, and AI-powered pest and disease diagnosis. These components work collaboratively to create a smart agricultural ecosystem that enables real-time monitoring, automated decision-making, and data-driven recommendations.

This chapter elaborates on the interaction between the hardware components—such as soil sensors, microcontrollers, solar power modules, and pumps—and the software modules that handle data collection, cloud storage, and analytics. The integration ensures that moisture readings, weather data, and crop health inputs are continuously processed to deliver actionable insights directly to farmers through an intuitive user interface.

Furthermore, the communication framework between IoT devices, cloud servers, and the user application is discussed in detail, illustrating how data flows seamlessly through the system. This includes the acquisition of sensor data, transmission via secure network protocols, machine learning-based rainfall analysis, and automated irrigation activation. Such interconnected design ensures minimal manual intervention while maximizing efficiency and sustainability.

To provide a clearer understanding of the system's operation, this chapter also includes visual representations such as Data Flow Diagrams (DFDs), Activity Diagrams, and Use Case Diagrams. These models demonstrate how data is captured, processed, and utilized for intelligent irrigation, soil condition assessment, and pest detection.

Overall, the system design of SoilStream represents a transformative step toward smart and sustainable agriculture. By combining IoT, AI, and renewable energy, the system empowers farmers to make informed decisions, conserve water, and enhance productivity, ultimately contributing to the broader goal of modernizing agriculture in the Thane region.

3.1 Existing System

The existing agricultural practices in the Thane region largely depend on traditional and manual methods that lack automation and data-driven intelligence. Irrigation is primarily carried out on a fixed schedule or based on the farmer's visual assessment of soil and crop conditions. This conventional approach often leads to inefficient water management, causing either excessive irrigation and wastage of valuable water resources or inadequate watering that affects plant growth and yield quality. Such manual practices fail to adapt to changing weather conditions and varying soil moisture levels, resulting in inconsistent crop productivity across different seasons.

Moreover, the process of crop management in the current system is reactive rather than preventive. Farmers usually identify pests and diseases only after visible damage has occurred, at which point yield loss is already significant. Diagnosis and treatment in such cases rely on manual consultation with agricultural experts or local experience, which may not always provide accurate or timely solutions. The absence of predictive systems or automated monitoring tools delays necessary interventions, exacerbating the problem and leading to potential economic losses.

Another major limitation of the existing system is the lack of real-time data collection and remote monitoring. Most farms operate without IoT-based sensing or cloud-connected infrastructure, preventing continuous observation of crucial parameters such as soil moisture, temperature, humidity, and rainfall patterns. Consequently, decision-making is based on intuition rather than data analysis, reducing efficiency in both irrigation and pest control management.

Additionally, the unpredictable nature of weather patterns in the Thane region poses a significant challenge. Without access to localized rainfall prediction or weather forecasting models, farmers struggle to plan irrigation schedules effectively. This unpredictability often results in either overuse of water resources or crop dehydration during dry spells. Furthermore, manual irrigation and crop care increase labor dependency and energy costs, adding to the financial burden of small-scale farmers.

Overall, the existing agricultural ecosystem lacks the integration of modern technologies such as IoT, machine learning, and artificial intelligence that could automate processes and enable intelligent decision-making. The current system's dependency on manual judgment, delayed response to crop health issues, and inability to adapt to dynamic environmental conditions underscore the urgent need for an advanced, automated, and data-driven solution. An intelligent system capable of real-time monitoring, predictive analytics, and automated control can significantly enhance productivity, conserve resources, and promote sustainable agricultural practices in the Thane region.

3.2 Proposed System

The system architecture of the AIoT Smart Irrigation Crop Management System is designed to offer a comprehensive precision agriculture solution by seamlessly integrating both hardware and software components. This architecture facilitates real-time farm monitoring, proactive resource management, and intelligent, automated crop care. It incorporates a blend of modern technologies, including the Internet of Things (IoT) for data collection, Artificial Intelligence (AI) for predictive analytics, and solar power for sustainable, off-grid

operation, creating a robust system capable of functioning effectively in the local conditions of the Thane region.

The architecture consists of several core modules, each performing a specific function while working in synchronization to optimize farming outcomes. The hardware layer serves as the on-field data collection and actuation point. It includes an ESP32-WROOM-32 microcontroller, a Capacitive Soil Moisture Sensor, and a BME280 sensor for temperature, humidity, and atmospheric pressure. This hardware node is powered by a 6V solar panel, a TP4056 charge controller, and an 18650 Li-ion battery, ensuring continuous operation. The sensors gather critical data—soil water content and atmospheric conditions—which are crucial for making intelligent irrigation decisions, while a 5V Relay Module controls a submersible water pump to execute the system's commands.

On the software side, the architecture integrates a suite of advanced machine learning algorithms to provide predictive insights. The Rainfall Prediction Model analyzes atmospheric data from the BME280 sensor to forecast the likelihood of rain, preventing unnecessary watering. The Crop Identification Model, a Convolutional Neural Network (CNN), analyzes a photo of a plant leaf uploaded by the farmer to automatically identify the crop and tailor watering thresholds accordingly. A third AI model, the Pest/Disease Diagnosis Model, uses a similar CNN architecture to identify signs of disease from a leaf image, providing an early warning and recommended remedies. This proactive system ensures that resources are used efficiently and threats to crop health are identified before they escalate.

Data collected from the hardware is transmitted via Wi-Fi and processed in real-time, with the system relying on the Google Firebase platform for scalability, storage, and serverless computing. Firestore is used as the NoSQL database to store all sensor readings, user data, and system states, while Firebase Cloud Functions host the backend logic and execute the AI models. When new sensor data arrives, a cloud function is triggered, which runs the appropriate analysis and updates the device's control commands in the database. This data flow allows for swift, automated interventions based on real-time conditions.

The system also employs secure user authentication to ensure that only the owner can access and control their farm's data. Furthermore, the architecture is designed to be resilient. The solar power system with battery backup ensures the hardware remains operational even during cloudy days or at night. The firmware on the ESP32 can be programmed with a fail-safe mode; in case of poor network connectivity, it can revert to a basic irrigation schedule based on the last known soil moisture reading, ensuring the system remains functional even in areas with unreliable internet.

The proposed system creates a closed-loop ecosystem designed to bring precision agriculture to local farmers. The hardware, consisting of the ESP32 and associated sensors, captures real-time environmental data. On the software side, a React Native mobile application provides a user-friendly interface for monitoring and control. The backend is powered by Google Firebase, which stores data in Firestore and runs serverless logic in Cloud Functions (written in Node.js). The intelligence layer consists of Python-based machine learning models (using libraries like TensorFlow and LightGBM) deployed on the cloud. This transforms traditional farming from a manual, reactive process into an automated, data-driven, and proactive one.

3.2.1 Critical Components of System Architecture

The system architecture is designed to provide a seamless interaction between the farmer, on-field sensors, cloud AI services, and the mobile application, creating a comprehensive precision agriculture solution. This architecture integrates Artificial Intelligence (AI), the Internet of Things (IoT), solar power, and cloud computing to optimize water usage, monitor crop health, and automate farm management. These technologies work in harmony to analyze real-time environmental data, provide predictive insights, and generate automated irrigation responses, making the system robust, sustainable, and highly efficient. Visual representations in the form of UML diagrams such as Class, Activity, Use Case, and Sequence Diagrams are critical in the design phase because they provide a theoretical blueprint, ensure clarity, and facilitate communication among developers, stakeholders, and users.

At the core of the system lies the AI-powered analytics module, which processes real-time sensor data and user-provided images to make intelligent decisions. This module leverages a suite of machine learning models to perform distinct tasks. The Rainfall Prediction Model uses atmospheric data (temperature, humidity, pressure) from on-field sensors to forecast the likelihood of rain, allowing the system to proactively skip unnecessary irrigation cycles. The Crop Identification Model utilize deep learning (CNNs) to analyze images uploaded by the farmer, accurately identifying the crop to tailor watering schedules or detecting signs of distress to provide early warnings and remedial advice. These models can be continuously improved with new data to enhance their localized accuracy for the Thane region.

The on-field IoT sensing module is the system's primary source of real-world data, continuously monitoring the farm's micro-climate. This solar-powered hardware node, built around an ESP32 microcontroller, uses a Capacitive Soil Moisture Sensor to get precise readings of water content in the soil and a BME280 sensor to gather atmospheric data. This tracking system is designed to function reliably in diverse farm environments, with the integrated solar panel and battery backup ensuring uninterrupted operation. All collected data is encrypted and securely transmitted over Wi-Fi to the cloud backend, ensuring data integrity and privacy.

Additionally, the automated irrigation control module triggers the water pump based on the intelligent insights generated by the cloud backend. When the system determines that watering is necessary (i.e., the soil is dry below the crop-specific threshold and no rain is predicted), it sends an "ON" command to the device's document in the Firestore database. The on-field hardware, listening for real-time changes, receives this command and activates a relay to start the water pump. This automated response ensures that crops receive the precise amount of water exactly when needed, eliminating guesswork and conserving resources.

To enhance user interaction, the system is equipped with a user-friendly mobile application that provides real-time farm monitoring, remote control, and access to AI-powered tools. The application features an intuitive dashboard displaying live data visualizations, the pump's operational status, weather forecasts, and actionable system recommendations. Farmers can use the app to customize settings, view historical data trends, and access the AI features for crop and pest analysis, effectively turning their smartphone into a powerful farm management tool.

Furthermore, a cloud-based backend using Google Firebase securely maintains all user data, historical sensor logs, and analytical results. The cloud infrastructure is designed to handle large-scale data processing from numerous IoT devices while ensuring high availability and reliability. Firebase Cloud Functions provide serverless compute resources that execute

the system's core logic and AI model inferences, allowing the system to scale seamlessly without the need for managing dedicated servers. Advanced authentication mechanisms safeguard sensitive farm data, ensuring only the authorized farmer can access and manage their system.

By combining these critical components, the AIoT system ensures a proactive and intelligent agricultural mechanism that not only reacts to current conditions but also anticipates future needs. The system is designed to continuously evolve through AI-driven insights and the potential addition of new sensors and features. This makes it an indispensable tool for modernizing local agriculture and promoting sustainable farming practices.

3.3 System Diagrams

The architecture of SoilStream - A Smart Solar-Powered IoT System for Real-Time Soil Moisture Monitoring and Automated Irrigation is visually articulated through a comprehensive set of system diagrams. These diagrams serve as the formal blueprints that translate the conceptual framework of the project into a tangible and understandable design. Given the system's complexity—spanning physical hardware, embedded firmware, cloud infrastructure, machine learning models, and a user-facing mobile application—these visual representations are indispensable. They are crucial for managing this complexity by providing a clear, standardized language that bridges the gap between the project's technical implementation and its agricultural objectives, ensuring that developers, stakeholders, and end-users share a unified understanding of the system's structure and behavior.

To capture a complete and holistic view of the system, the architecture is documented from multiple perspectives, distinguishing between its static structure and its dynamic behavior. The static view, represented by the UML Class Diagram, acts as the system's floor plan; it defines the core components, their attributes, and the permanent relationships between them, illustrating what the system is. In contrast, the dynamic view illustrates what the system does. This is captured through several diagrams: the Use Case Diagram models the system's functionality from the farmer's perspective; the Activity Diagram details the step-by-step internal workflow of a specific process like automated irrigation; and the Sequence Diagram meticulously maps the timed message exchanges between components for a given interaction.

This multi-faceted approach is fundamental to a robust design process. By creating these diagrams before implementation, potential design flaws, logical inconsistencies, and performance bottlenecks can be identified and resolved early, saving significant time and resources during the development phase. Furthermore, these diagrams are not merely preliminary design artifacts; they form the core of the system's technical documentation. They provide an invaluable guide for the implementation of the hardware and software, facilitate effective debugging, and ensure that the final product is scalable, maintainable, and aligned with the initial project goals. Collectively, the following diagrams provide a complete architectural overview of the AIoT system.

3.3.1 UML Class Diagram

The Unified Modeling Language (UML) diagram for the SoilStream - A Smart Solar-Powered IoT System for Real-Time Soil Moisture Monitoring and Automated Irrigation provides a detailed structural representation of the system's core components and their

interactions. This diagram helps visualize the system's architecture and the relationships between various entities, including the farmer (user), the on-field hardware unit, the cloud backend, the mobile application, and the integrated AI models.

UML diagrams offer a standardized approach to understanding system architecture, making it easier for developers and agricultural technologists to efficiently implement and maintain the platform. These diagrams also assist in streamlining the engineering process by breaking down the complex system into manageable components—the physical IoT device, the cloud services, and the user-facing application—and illustrating how they interconnect to create a cohesive solution.

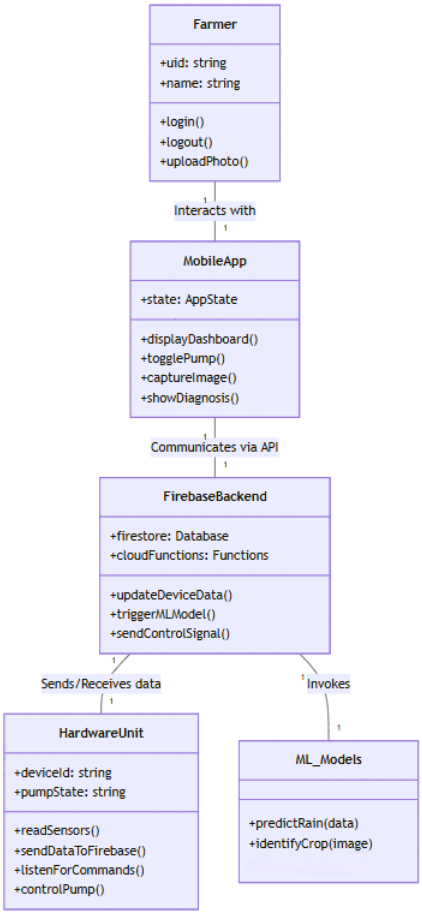


Figure 3.1: UML Class Diagram of AIoT Smart Irrigation & Crop Management System

The above Figure 3.1 consists of the following key entities.The Farmer (User) acts as the primary entity interacting with the mobile application for real-time farm monitoring, manual irrigation control, and accessing AI-driven insights. The Hardware Unit is a solar-powered IoT device deployed in the field, responsible for capturing environmental parameters such as soil moisture, temperature, and humidity through sensors, and controlling the water pump using a relay mechanism. The Cloud Backend, developed on Google Firebase, serves as the central hub that receives and processes sensor data, executes automated irrigation logic, and triggers the integrated machine learning models. These AI/ML Models are designed

for key functionalities such as rainfall prediction and crop identification, enabling data-driven decision-making for farmers. The Database (Firestore) component securely stores user profiles, device data, historical sensor readings, and system-generated recommendations, ensuring reliable data access and management. Finally, the Mobile Application serves as the user-facing dashboard, offering intuitive data visualization, manual control features, and AI-based tools to assist farmers in efficient farm operation and resource optimization.

3.3.2 Activity Diagram

The Activity Diagram for SoilStream - A Smart Solar - Powered IoT System for Real-Time Soil Moisture Monitoring and Automated Irrigation illustrates the sequence of operations for the core automated workflow, detailing the step-by-step process from environmental sensing to intelligent water management. This diagram helps visualize how data flows between the on-field hardware and the cloud backend, and how the AI models interact in real-time to ensure seamless coordination and efficient resource allocation.

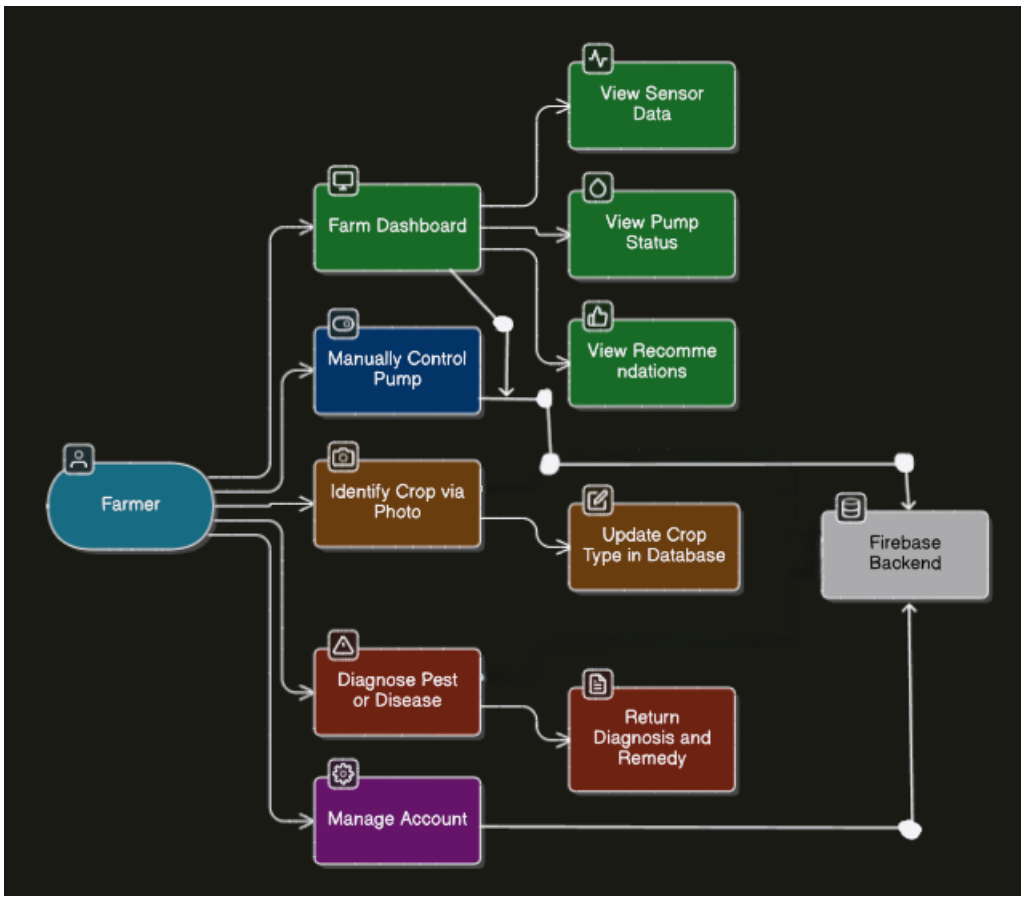


Figure 3.2: Activity Diagram showing automated irrigation workflow

The above Figure 3.2 follows a structured sequence of operations to enable intelligent and automated irrigation management. In the Data Sensing phase, the ESP32 microcontroller collects real-time data from environmental sensors such as the soil moisture sensor and the BME280 module, which measures temperature and humidity levels. During the Cloud Transmission stage, this sensor data is transmitted securely to the Firebase Firestore

database for processing and storage. The AI-Powered Weather Analysis phase utilizes a machine learning model to predict rainfall probability and dynamically adjust irrigation cycles based on weather conditions and soil parameters. In the Actuation Command stage, the system determines the appropriate pump state ("ON" or "OFF") depending on the detected soil dryness level, ensuring optimal water usage. Finally, the Physical Response phase involves the immediate activation or deactivation of the water pump, thereby completing the automated irrigation loop and maintaining ideal soil moisture conditions efficiently.

3.3.3 Use Case Diagram

Use Case Diagrams visualize how the user (farmer) interacts with the system and identify the system's core functionalities. For SoilStream - A Smart Solar - Powered IoT System for Real-Time Soil Moisture Monitoring and Automated Irrigation, the Use Case Diagram outlines all major interactions between the farmer and backend services.

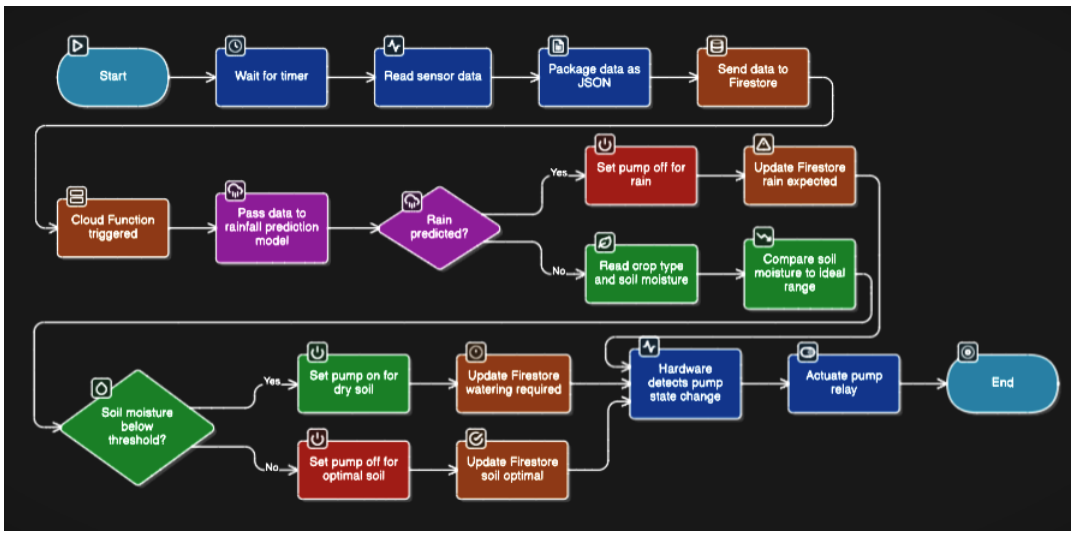


Figure 3.3: Use Case Diagram depicting farmer interactions and system functionalities

The above Figure 3.3 provides a range of interactive and intelligent features to enhance farm management and decision-making. Users can monitor real-time farm data through an intuitive dashboard that visualizes key environmental parameters such as soil moisture, temperature, and humidity. They can also manually control the irrigation pump, allowing flexibility to override automated operations when necessary. The application further enables users to identify crops through photo uploads, utilizing AI-based image recognition to determine crop types accurately. Additionally, the system can diagnose pests or diseases using AI analysis, where uploaded images are processed by advanced models to detect potential issues and suggest appropriate treatments, thereby supporting farmers in maintaining healthy and productive crops.

3.3.4 Sequence Diagram

A sequence diagram represents the order of interactions between components of the system over time. In the SoilStream - A Smart Solar-Powered IoT System for Real-Time Soil Moisture Monitoring and Automated Irrigation, this diagram showcases how the mobile app, cloud backend, AI models, and database interact for tasks such as pest or plant diagnosis.

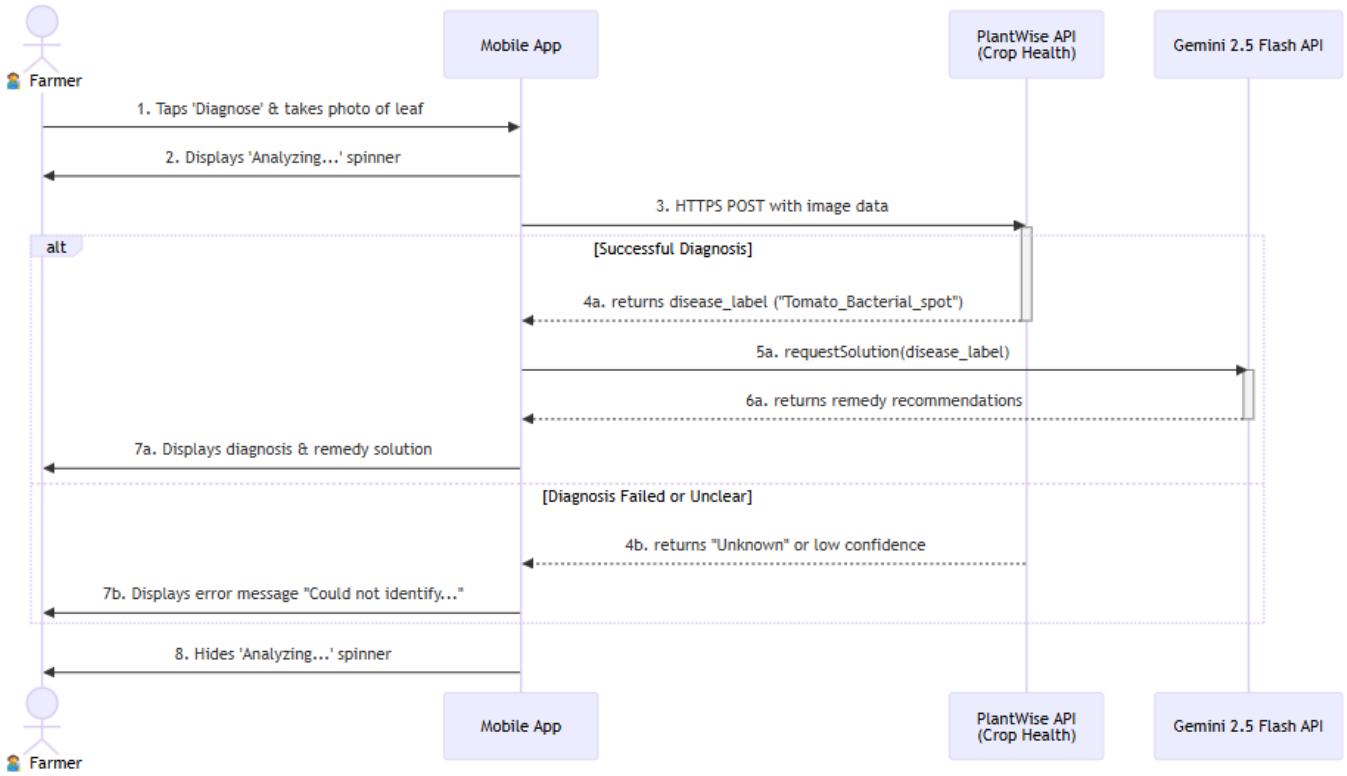


Figure 3.4: Sequence Diagram representing interaction flow between user, app, and backend

The above Figure 3.4 follows a well-defined sequence to ensure seamless interaction between the farmer, the application, and the backend services. It begins when the farmer triggers a diagnosis request through the mobile application by uploading a photo of the affected crop area. Upon submission, the backend invokes the API responsible for analyzing the image using advanced AI algorithms to detect potential pests or diseases. During this process, there is a structured sequence of data exchange between the app and the server, where the image and related metadata are securely transmitted to the backend, processed, and analyzed. Finally, the system ensures response delivery containing the diagnosis and recommendation, which is sent back to the mobile application, providing the farmer with accurate insights and actionable treatment suggestions directly within the app interface.

Chapter 4

Project Implementation

This chapter provides a comprehensive overview of the implementation phase of the project, detailing the technical execution and development process. It includes key code snippets that illustrate core functionalities, step-by-step instructions for accessing and interacting with the system, and a timeline that outlines the progression of development milestones. This chapter aims to connect the conceptual framework with its practical execution, illustrating how the proposed design was methodically developed into a working system.

4.1 Code Snippets

This section presents key code snippets implemented in the SoilStream mobile application.

I]Hardware Implementation – IoT-Based Environmental Monitoring

1) Library Inclusions and Pin Configuration This section defines the required libraries, Wi-Fi credentials, Firebase configuration, and hardware pin mappings used by the ESP32 microcontroller to interface with the soil moisture sensor and BME280 environmental sensor.

```
#include <WiFi.h>
#include <Firebase_ESP_Client.h>
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BME280.h>
#include "addons/TokenHelper.h"
#include "addons/RTDBHelper.h"

// ========================================
// CONFIGURATION
// ========================================
#define WIFI_SSID "YOUR_WIFI_SSID"
#define WIFI_PASSWORD "YOUR_WIFI_PASSWORD"
#define API_KEY "YOUR_FIREBASE_API_KEY"
#define DATABASE_URL "YOUR_FIREBASE_DATABASE_URL" // e.g. https://your-project.firebaseio.com/
#define DEVICE_ID "DEVICE_001"

// ========================================
// PIN CONFIGURATION
// ========================================
const int sensorPin = 34;     // Soil moisture sensor (analog)
const int pump = 5;           // Pump relay pin
const int dryValue = 4000;
const int wetValue = 3000;

// BME280 I2C: SDA = GPIO 21, SCL = GPIO 22
```

Figure 4.1: Library Inclusions and Pin Configuration

Explanation: In above figure 4.1, the ESP32 uses GPIO 34 as the analog input for the capacitive soil moisture sensor, while GPIO 5 controls the relay module connected to the water pump. The dryValue and wetValue constants define the raw ADC range used to map sensor readings to a 0–100.

2) Setup – Sensor and Wi-Fi Initialization The setup() function initializes all hardware peripherals, establishes Wi-Fi connectivity, and authenticates with Firebase to enable cloud communication.

```
void setup() {
  Serial.begin(115200);
  delay(1000);

  Serial.println("\n================================");
  Serial.println("ESP32 Smart Irrigation System");
  Serial.println("================================\n");

  // Initialize pump
  pinMode(pump, OUTPUT);
  digitalWrite(pump, LOW); // Pump OFF initially

  // Initialize I2C for BME280
  Wire.begin(21, 22);

  // Initialize BME280
  Serial.println("Initializing BME280 sensor...");
  if (!bme.begin(0x77)) {  // Try 0x77, change to 0x76 if needed
    Serial.println("❌ Could not find BME280 sensor!");
    Serial.println("Check wiring: SDA=GPIO21, SCL=GPIO22, VCC=3.3V");
    // Continue without BME280 (only soil moisture will work)
  } else {
    Serial.println("✓ BME280 initialized successfully\n");
  }

  // Connect to WiFi
  connectWiFi();
```

Figure 4.2: Setup – Sensor and Wi-Fi Initialization

Explanation: In above figure 4.2, the pump relay is initialized to LOW (OFF) to prevent unintended activation on startup. The BME280 is initialized over I2C at address 0x77. The system attempts Wi-Fi connection and upon success, performs an anonymous Firebase sign-up to obtain an authentication token, enabling secure read/write access to the Realtime Database.

3)Loop - Sensor Reading and Real-Time Monitoring The loop() function continuously reads soil moisture and atmospheric data every 5 seconds, processes the raw values into meaningful metrics, and prints them to the Serial Monitor for real-time observation

```
void loop() {
  // Send sensor data every 5 seconds
  if (Firebase.ready() && signupOK && (millis() - sendDataPrevMillis > 5000 || sendDataPrevMillis == 0)) {
    sendDataPrevMillis = millis();

    // Read soil moisture
    int soilRaw = analogRead(sensorPin);
    int moisturePercent = map(soilRaw, dryValue, wetValue, 0, 100);
    moisturePercent = constrain(moisturePercent, 0, 100);

    // Read BME280 data
    float temperature = bme.readTemperature();
    float humidity = bme.readHumidity();
    float pressure = bme.readPressure() / 100.0F; // Convert to hPa

    // Determine soil status
    String status;
    if (moisturePercent >= 70) {
      status = "Very Wet";
    } else if (moisturePercent <= 30) {
      status = "Dry - Needs Water";
    } else {
      status = "Moist (Good)";
    }
```

Figure 4.3: Loop - Sensor Reading and Real-Time Monitoring

Explanation: In above figure 4.3, the analogRead() function captures the raw ADC value from the capacitive soil moisture sensor. The Arduino map() function converts this raw value to a percentage based on the pre-calibrated dryValue and wetValue thresholds, while constrain() clamps the result within the valid 0–100 range. Simultaneously, the BME280 sensor provides temperature (°C), relative humidity , and atmospheric pressure (hPa). A conditional status string "Dry - Needs Water", "Moist (Good)", or "Very Wet" is derived from the moisture percentage to provide a human-readable soil condition assessment, which is displayed on the Serial Monitor and later used to drive irrigation decisions.

II]Rainfall Prediction and Smart Irrigation Logic – Firebase & Frontend Integration

1) Backend Logic : Firebase Cloud Function for Rainfall Prediction

This section implements the server-side logic using Firebase Cloud Functions to perform rainfall prediction and automated irrigation control based on real-time sensor data.

```
exports.predictRainfall = onValueWritten(
    "/devices/{deviceId}/predictionTrigger",
    async (event) => {
      try {
        const deviceId = event.params.deviceId;
        const newData = event.data.after.val();

        if (!newData) {
          console.log("No data to process");
          return null;
        }

        const {temperature, humidity, pressure, soilMoisture} = newData;

        // Validate data
        if (soilMoisture === undefined || soilMoisture === null) {
          console.error("❌ No soil moisture data in trigger!");
          return null;
        }

        const currentDate = new Date();
        const month = currentDate.getMonth() + 1;

        console.log(`🌧️ Rainfall Prediction for ${deviceId}`);
        console.log(
            `📊 DATA: Temp=${temperature}°C, Hum=${humidity}%, ` +
          `Pres=${pressure}hPa, Soil=${soilMoisture}%, Month=${month}`,
        );
```

Figure 4.4: Firebase Cloud Function for Rainfall Prediction

Explanation: In above figure 4.4,this Firebase Cloud Function is triggered whenever new sensor data is written to the database. It extracts environmental parameters such as temperature, humidity, pressure, and soil moisture. These values are passed to the rainfall prediction logic, which determines whether rainfall is expected. Based on the prediction and soil conditions, the system automatically decides whether the irrigation pump should be turned ON or OFF. The results are stored back in Firebase, enabling real-time synchronization with the frontend.

2) Rainfall Prediction Logic (ML-Based Rule Engine)

This function represents the machine learning logic used for rainfall prediction based on environmental parameters.

```
function predictRainfallKonkan(temperature, humidity, pressure, month) {
  const monsoonMonths = [6, 7, 8, 9];
  const dryMonths = [12, 1, 2, 3];

  const isMonsoonSeason = monsoonMonths.includes(month);
  const isDrySeason = dryMonths.includes(month);

  let willRain = false;
  let confidence = "low";
  let matchedRule = "default";

  // SIMPLIFIED BINARY RULES - Will it rain? YES or NO

  // Rule 1: Monsoon + High humidity + Low pressure = YES
  if (isMonsoonSeason && humidity >= 80 && pressure <= 1010) {
    willRain = true;
    confidence = "high";
    matchedRule = "Monsoon + High Humidity + Low Pressure";
  }
  // Rule 2: Monsoon + Moderate humidity = YES
  else if (isMonsoonSeason && humidity >= 70 && pressure <= 1012) {
    willRain = true;
    confidence = "medium";
    matchedRule = "Monsoon + Moderate Humidity";
  }
  // Rule 3: High humidity + Low pressure (any season) = YES
  else if (humidity >= 80 && pressure <= 1011) {
    willRain = true;
    confidence = "medium";
    matchedRule = "High Humidity + Low Pressure";
  }
```

Figure 4.5: Rainfall Prediction Logic

Explanation: In above figure 4.5, this function acts as a lightweight machine learning model using rule-based classification. It evaluates key environmental features such as humidity, pressure, and seasonal context to determine rainfall probability. The output includes a binary prediction (rain or no rain) along with a confidence level. This approach ensures fast and efficient execution suitable for real-time cloud processing.

3) Smart Irrigation Decision Logic

This function determines the irrigation control strategy based on rainfall prediction and soil moisture levels.

```
function determinePumpControl(soilMoisture, willRain, currentPumpState) {
  let shouldPumpBeOn = false;
  let reason = "";

  // SIMPLE LOGIC:
  // 1. If RAIN = YES → Pump OFF (save water)
  // 2. If RAIN = NO and SOIL DRY → Pump ON
  // 3. If RAIN = NO and SOIL WET → Pump OFF
  // 4. If RAIN = NO and SOIL MODERATE → Keep current state

  if (willRain) {
    // Rain expected - Turn OFF pump
    shouldPumpBeOn = false;
    reason = "Rain expected. Pump OFF to save water.";
  } else if (soilMoisture < 30) {
    // No rain + Dry soil - Turn ON pump
    shouldPumpBeOn = true;
    const soil = Math.round(soilMoisture);
    reason = `No rain expected and soil is dry (${soil}%). Pump ON.`;
  } else if (soilMoisture > 70) {
    // No rain + Wet soil - Keep OFF
    shouldPumpBeOn = false;
    const soil = Math.round(soilMoisture);
    reason = `Soil moisture is sufficient (${soil}%). Pump OFF.`;
  } else {
    // No rain + Moderate soil - Maintain current state
    shouldPumpBeOn = currentPumpState;
    const soil = Math.round(soilMoisture);
    reason = `No rain, moderate soil (${soil}%). Maintaining pump state.`;
  }

  return {shouldPumpBeOn, reason};
}
```

Figure 4.6: Smart Irrigation Decision Logic

Explanation: In above figure 4.6, the irrigation logic follows a simple decision-making process. If rainfall is expected, the pump is turned OFF to conserve water. If no rain is expected and the soil is dry, the pump is activated. If the soil is already wet, irrigation is avoided. Otherwise, the system maintains the current pump state. This ensures efficient water utilization and automated farm management.

4) Manual Pump Override from Frontend

This feature allows users to manually control the irrigation system from the mobile application.

```
const togglePump = async () => {
  if (!pumpControl) return;

  const newState = !pumpControl.status;

  try {
    // Update manual override in Firebase
    const manualOverrideRef = ref(database, `devices/${DEVICE_ID}/pumpControl/manualOverride`);
    await set(manualOverrideRef, newState);

    Alert.alert(
      'Pump Control',
      `Pump turned ${newState ? 'ON' : 'OFF'} manually`,
      [{ text: 'OK' }]
    );
  } catch (error) {
    console.error('Error toggling pump:', error);
    Alert.alert('Error', 'Failed to control pump');
  }
};
```

Figure 4.7: Manual Pump Override from Frontend

Explanation: From the above figure 4.7, we can see that in addition to automated control, the system provides manual override functionality. Users can directly control the pump state through the application. When triggered, the command is written to Firebase, which updates the system and activates the pump accordingly. This ensures flexibility and user control in critical situations.

III]Crop Analysis

1)Open Camera or Gallery This function allows users to capture an image using the camera or choose an existing one from the gallery. It uses the `Expo ImagePicker` library to handle media access and updates the selected image state for further processing.

```
// ✅ Open camera or gallery
const pickImage = async (from: "camera" | "gallery") => {
  let result;
  if (from === "camera") {
    await ImagePicker.requestCameraPermissionsAsync();
    result = await ImagePicker.launchCameraAsync({
      mediaTypes: ImagePicker.MediaTypeOptions.Images,
      quality: 1,
    });
  } else {
    await ImagePicker.requestMediaLibraryPermissionsAsync();
    result = await ImagePicker.launchImageLibraryAsync({
      mediaTypes: ImagePicker.MediaTypeOptions.Images,
      quality: 1,
    });
  }

  if (!result.canceled) {
    const asset = result.assets[0];
    setSelectedImage({
      uri: asset.uri,
      fileName: `crop_${Date.now()}.jpg`,
      type: "image/jpeg",
    });
    setAnalysis(null);
  }
};
```

Figure 4.8: Open Camera or Gallery

Explanation: In above figure 4.8,this function first requests the required permissions to access the device's camera or gallery. Depending on the user's choice, it launches the camera or image picker and updates the app state with the selected image data (URI, filename, and type). The `setAnalysis(null)` resets previous results to prepare for a new analysis.

2)Show Image Picker Option This function displays an alert box allowing users to choose between capturing a photo using the camera or selecting one from the gallery.

```
// ✅ Show options
const showImagePickerOptions = () => {
  Alert.alert("Select Image", "Choose how you want to select an image", [
    { text: "Camera", onPress: () => pickImage("camera") },
    { text: "Gallery", onPress: () => pickImage("gallery") },
    { text: "Cancel", style: "cancel" },
  ]);
};
```

Figure 4.9: Show Image Picker Option

Explanation: The above figure 4.9, this function enhances user interaction by presenting three options in an alert dialog — "Camera," "Gallery," or "Cancel." When the user selects an option, the appropriate image selection method is triggered through the `pickImage()` function defined earlier.

3)Analyze Crop via Flask Backend This function sends the selected image to the Flask backend for AI-based crop analysis. The backend processes the image and returns predictions such as crop type and confidence level.

```
const analyzeCrop = async () => {
  if (!selectedImage) {
    Alert.alert("Error", "Please select an image first");
    return;
  }

  setIsAnalyzing(true);

  try {
    const formData = new FormData();
    formData.append("leaf", {
      uri: selectedImage.uri,
      type: selectedImage.type,
      name: selectedImage.fileName,
    } as any);

    const response = await fetch("http://10.99.243.64:5000/predict", {
      method: "POST",
      body: formData,
      headers: { "Content-Type": "multipart/form-data" },
    });

    const data = await response.json();
    console.log("Response data:", data);

    if (response.ok) {
      setAnalysis(data);
    } else {
      Alert.alert("Error", data.error || "Failed to analyze crop");
    }
  } catch (error) {
    console.error("❌ Analysis error:", error);
    Alert.alert("Error", "Failed to connect to server");
  } finally {
    setIsAnalyzing(false);
```

Figure 4.10: Analyze Crop via Flask Backend

Explanation: From above figure 4.10, this function checks if an image has been selected, then sends it to the Flask backend endpoint (`/predict`) using a POST request with multipart form data. The backend processes the image through a machine learning model and returns results in JSON format. If successful, the crop analysis results are displayed in the app; otherwise, an appropriate error message is shown to the user.

4.2 Steps to Access the System

To effectively utilize the SoilStream - A Smart Solar-Powered IoT System for Real-Time Soil Moisture Monitoring and Automated Irrigation, the user follows a structured process to access its core features. Each section in the mobile app is designed to be intuitive, guiding the user from initial setup to advanced AI-powered diagnostics. The following steps detail the process for accessing the system's intelligent crop management tools.

Step 1: Accessing the AI Crop Identification Feature - As shown in Figure 4.1, the user navigates to the "Crop Analysis" section within the app. From here, they can select an option to update their crop type, which opens the device's camera or photo gallery. The user uploads a clear photo of their crop. This image is sent to the AI model, and the identified crop type (e.g., "Cashew") is displayed, automatically calibrating the system's watering rules.

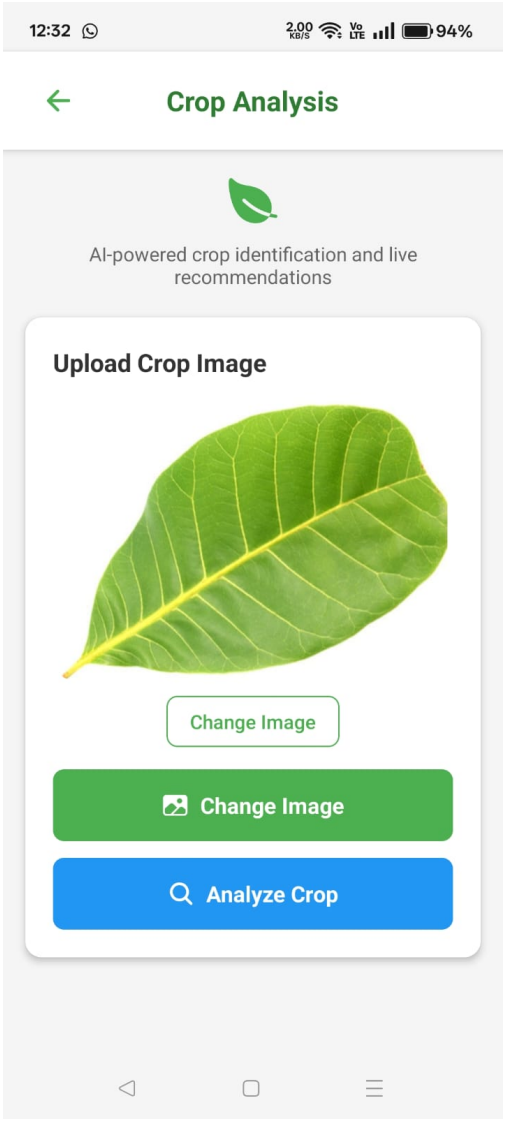


Figure 4.11: AI Crop Identification Interface in the mobile app.

Step 2: Accessing the Pest & Disease Diagnosis Feature - As shown in Figure 4.2, the user selects the "Plant Diagnosis" feature from the app. This action opens an interface to take or upload a photo of an affected plant leaf. Upon submission, the image is sent to the Kindwise API for analysis. Within seconds, the app receives the AI-generated diagnosis and a detailed list of actionable remedies, which are then displayed on the user's screen.

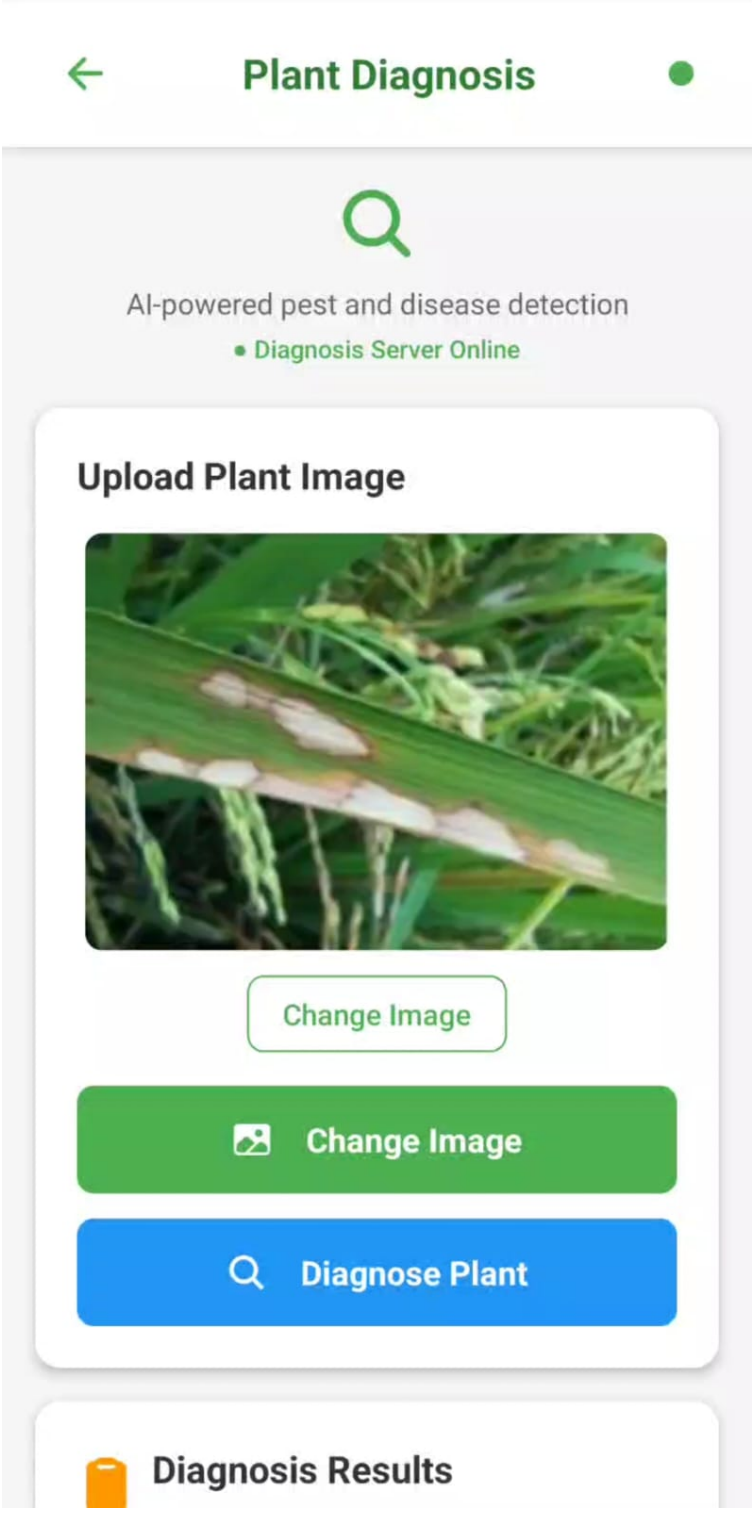


Figure 4.12: Pest and Disease Diagnosis Interface, showing the photo upload screen.

Step 3: Accessing the Smart Irrigation and Pump Control Feature - As shown in Figure 4.3, the user can access the irrigation control system directly from the dashboard interface of the mobile application. The dashboard displays real-time soil moisture levels, rainfall prediction status, and the current state of the water pump (ON/OFF). Based on this data, the system automatically decides whether irrigation is required. In automated mode, if the soil moisture is below the defined threshold and no rainfall is predicted, the system activates the pump. Conversely, if sufficient moisture is present or rainfall is expected, the pump remains OFF to conserve water. This ensures efficient water management without requiring constant user intervention.Additionally, the application provides a **manual override feature** through a toggle switch. By interacting with this control, the user can manually turn the pump ON or OFF as needed. Once activated, the command is sent to the Firebase database, which updates the pump state in real-time, and the IoT device executes the corresponding action.

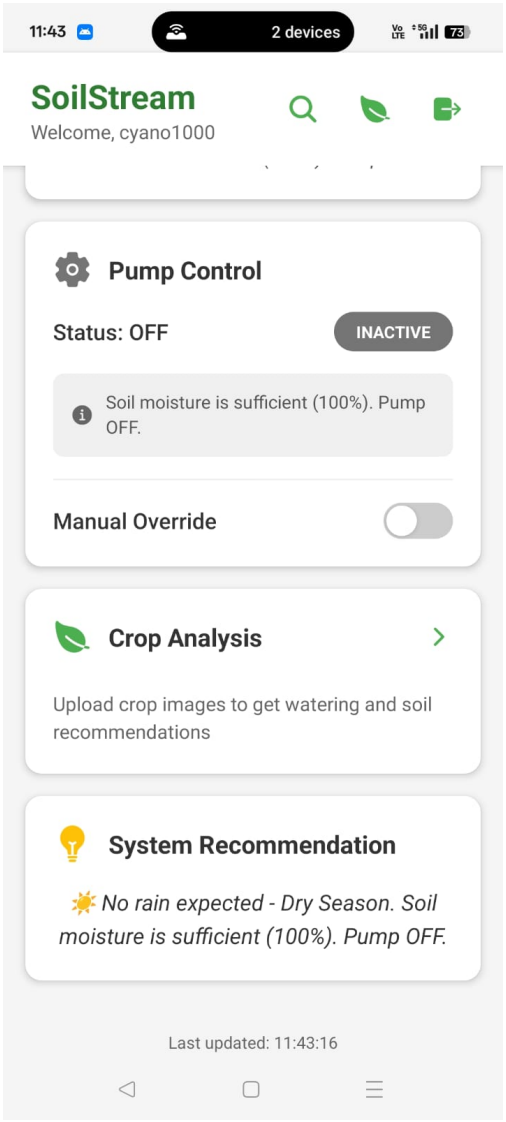


Figure 4.13: Smart Irrigation and Pump Control Interface showing real-time pump status and manual override option.

4.3 Implementation Images

The following section presents the implementation images of the SoilStream system, illustrating the practical realization of the proposed solution. These images highlight the key features of the mobile application interface, including real-time soil moisture monitoring, weather condition tracking, AI-based rainfall prediction, crop analysis, and automated pump control. The visual representations provide a comprehensive understanding of how the system operates in real-time, showcasing the integration of IoT sensors, cloud-based processing, and user interaction through an intuitive interface. The following figures illustrate the major stages and components of the system implementation, including frontend interfaces, backend configuration, and system workflow.

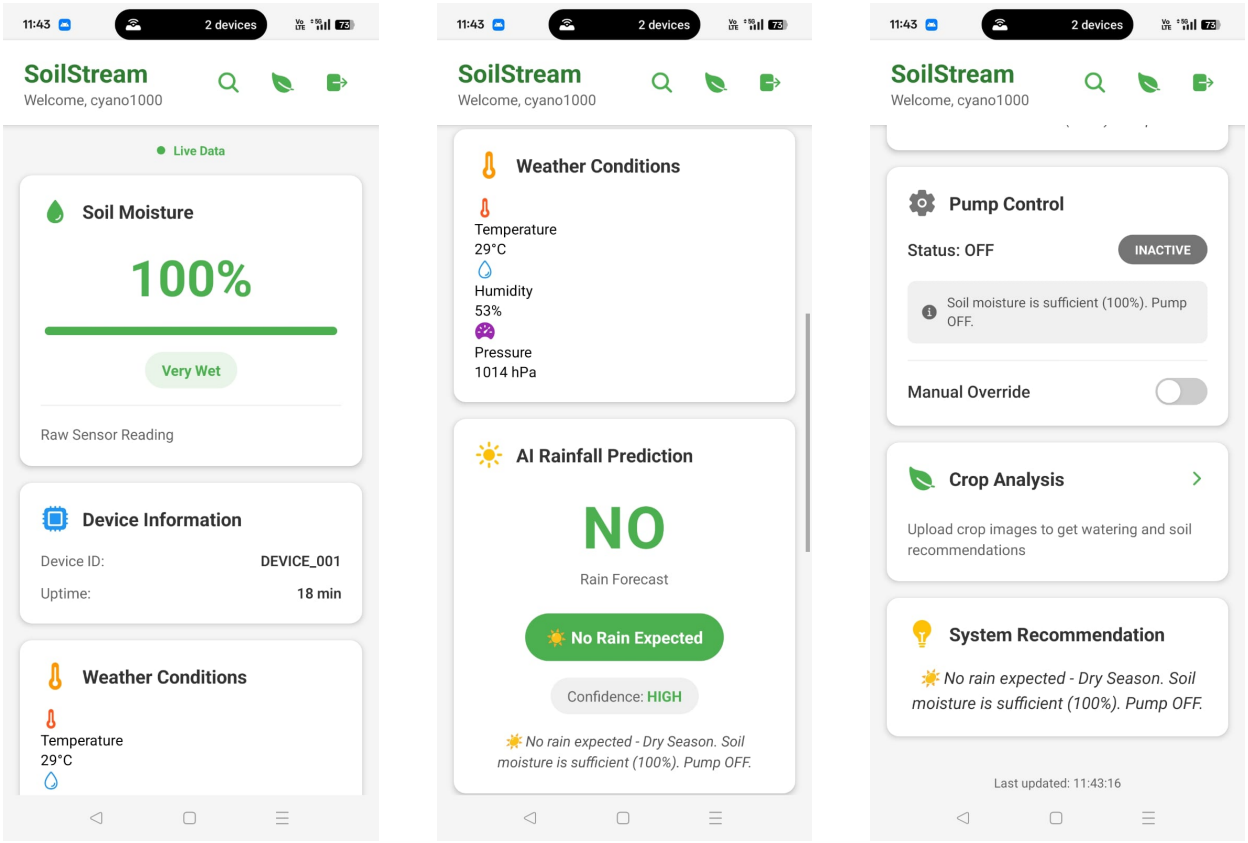


Figure 4.14: System Interface : Dashboard and Monitoring Views

The above figure 4.4 shows the interface that shows real-time soil moisture percentage, weather conditions (temperature, humidity, and pressure), AI-based rainfall prediction, and automated pump control status along with system recommendations.

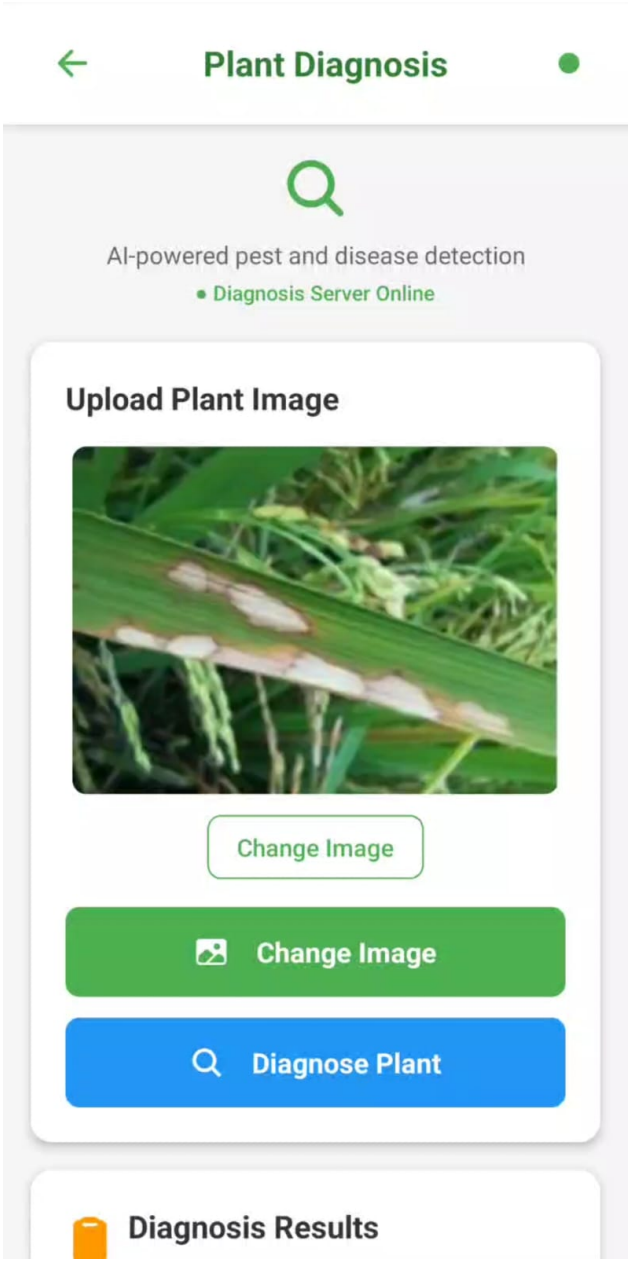


Figure 4.15: Mobile Application – Plant Diagnosis

The above Figure 4.5 illustrates the working of the AI-powered plant diagnosis feature, which analyzes uploaded leaf images to accurately detect pests and diseases, providing timely insights for effective crop management.

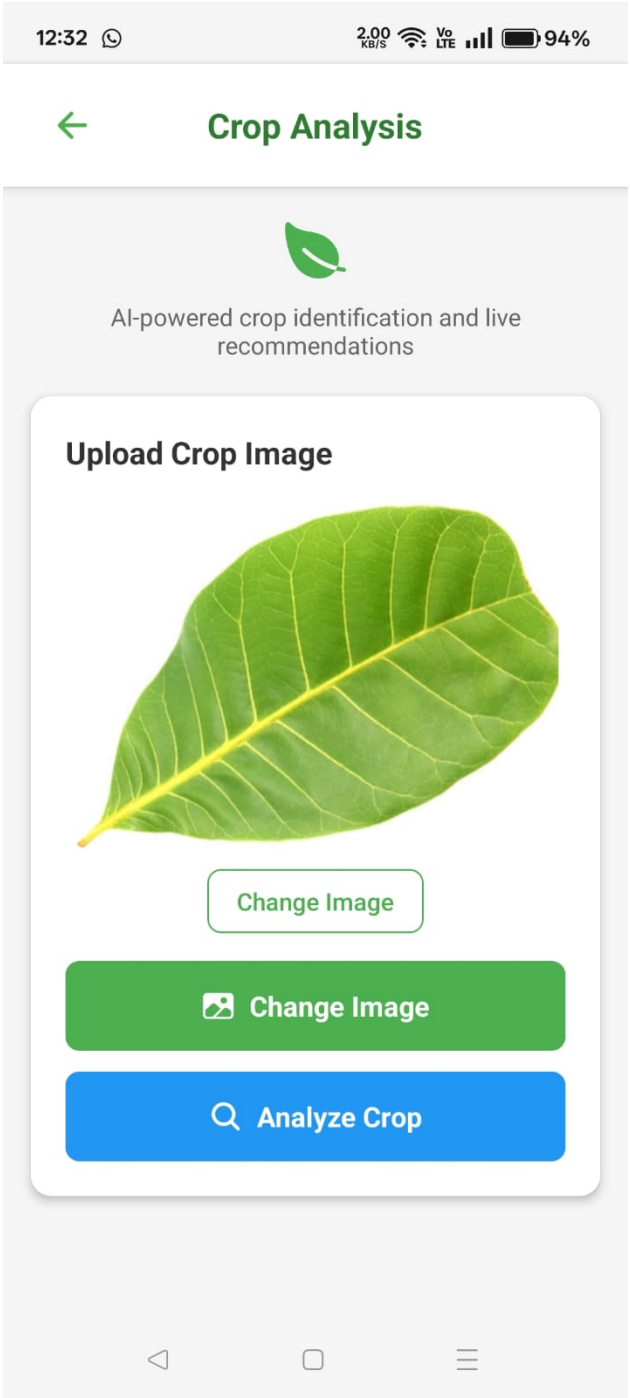


Figure 4.16: Mobile Application – Crop Analysis

The above Figure 4.6 demonstrates how, after identifying the crop, the system intelligently generates customized irrigation recommendations based on crop requirements and environmental conditions..

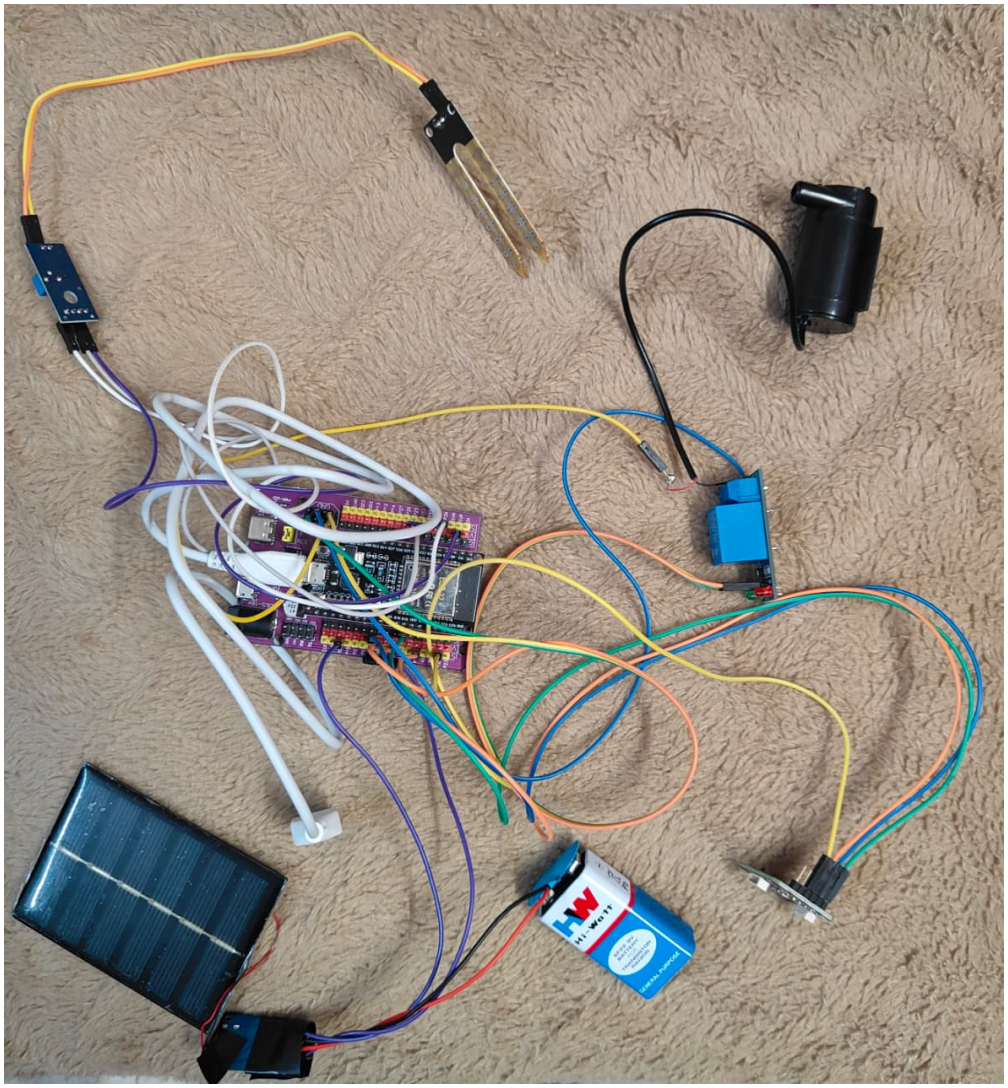


Figure 4.17: Hardware Implementation Setup :

The figure 4.7 shows the complete hardware setup of the smart irrigation system. The ESP32 microcontroller acts as the main controller, collecting data and controlling operations. A capacitive soil moisture sensor measures soil moisture, while the BME280 sensor monitors temperature and humidity.A relay module is used to control the DC water pump, which turns ON/OFF based on soil moisture levels. The system is powered using a solar panel and battery, making it suitable for remote and energy-efficient applications.Overall, this setup enables real-time monitoring and automatic irrigation.

4.4 Timeline Chart

The project timeline played a vital role in ensuring systematic planning and timely execution of all phases of development. The Gantt chart (Figure 4.7) presents a structured visualization of the project lifecycle, highlighting key milestones from initial ideation to final deployment and presentation. It reflects a well-organized progression of tasks, enabling effective monitoring and coordination throughout the development process.The project began with foundational activities such as topic selection, problem identification, and literature review, establishing a strong conceptual and research base. This phase ensured that the proposed solution was both technically feasible and aligned with real-world agricultural challenges. Following this, a detailed project plan was formulated, including task allocation, timeline definition, and system architecture design to guide subsequent development stages.The core development phase focused on implementing the system components in a modular manner. This included IoT hardware setup, sensor integration, and data acquisition, followed by backend development using Firebase for real-time data processing and storage. Parallelly, the frontend mobile application was developed using React Native, ensuring a user-friendly interface for monitoring and control. Machine learning logic for rainfall prediction and decision-making was also incorporated during this stage.As the project progressed, emphasis shifted towards system integration, where all components—hardware, backend, and frontend—were combined to form a cohesive smart irrigation system. This phase involved rigorous testing, debugging, and performance optimization to ensure reliability and accuracy under real-world conditions.The final phase included documentation, result analysis, and project presentation. Feedback from evaluations was incorporated to refine system performance and improve usability. Overall, the timeline demonstrates a disciplined, milestone-driven approach, ensuring steady progress, efficient resource utilization, and successful completion of the project objectives.

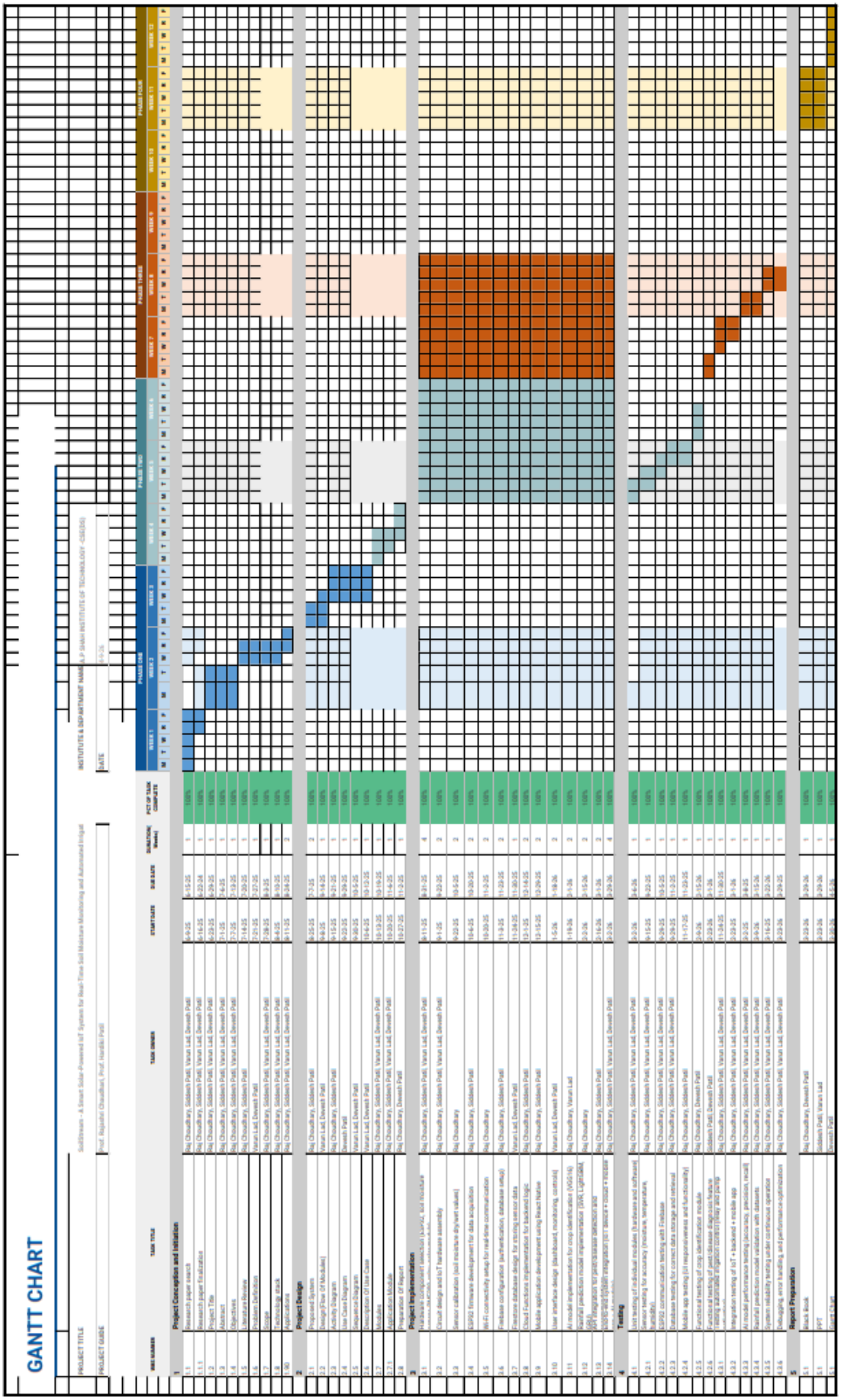


Figure 4.18: Project Timeline – Gantt Chart showing project phases and their completion schedule.

Chapter 5

Result and Discussions

This chapter presents the results obtained from the implementation of the SoilStream system and provides a detailed discussion on its performance and effectiveness. The evaluation focuses on key aspects such as soil moisture monitoring accuracy, rainfall prediction outcomes, and the efficiency of the automated irrigation mechanism. The results are analyzed using appropriate performance metrics and visual representations to assess the reliability and practical applicability of the system. Furthermore, this chapter discusses how the integration of IoT, machine learning, and cloud technologies contributes to improved decision-making, resource optimization, and overall system performance in real-world agricultural scenarios.

1.Crop Identification Performance

Table 5.1: Comparison of Model Performance Metrics

Metrics	**VGG16**	**ResNet50**	**EfficientNetB0**
Accuracy	0.956	0.997	0.1601
Validation Accuracy	0.9404	0.9648	0.9976
Loss	0.2350	0.0139	7826.8145
Validation Loss	0.2206	0.1225	0.0406
Macro Avg. Precision	0.95	0.97	0.05
Macro Avg. Recall	0.95	0.96	0.11
Macro Avg. F1-Score	0.95	0.96	0.05
Weighted Avg. Precision	0.96	0.97	0.06
Weighted Avg. Recall	0.95	0.97	0.16
Weighted Avg. F1-Score	0.95	0.97	0.07

The penultimate chapter of this report, "Results and Discussion," is dedicated to the thorough evaluation of the preliminary investigations carried out for the core AI component of the SoilStream system—the crop identification model. Table 5.1: Comparison of Model Performance Metrics summarizes the performance of the three Convolutional Neural Networks (CNNs) benchmarked for crop classification: VGG16, ResNet50, and EfficientNetB0.

The results indicate that ResNet50 achieved the highest overall performance on the test set, boasting a near-perfect Accuracy of 0.997 and the lowest Loss of 0.0139. Its high Macro and Weighted Average F1-Scores (0.96 and 0.97, respectively) suggest excellent classification ability across all crop classes, indicating minimal issues with class imbalance. The VGG16

architecture also demonstrated strong, reliable performance, achieving an Accuracy of 0.956 and a Weighted Avg. F1-Score of 0.95. While VGG16's metrics are slightly lower than ResNet50's, its performance is highly consistent between training and validation sets (Accuracy of 0.956 vs. Validation Accuracy of 0.9404), suggesting it is a robust model suitable for deployment, particularly if computational resources are a constraint.

In contrast, EfficientNetB0 exhibited poor performance on this dataset. Despite an unusually high Validation Accuracy of 0.9976, its standard Accuracy was only 0.1601, and its Loss was drastically high (7826.8145). Furthermore, its extremely low F1-Scores (Macro Avg. F1-Score of 0.05) clearly indicate a major failure, likely caused by a training anomaly, data mismatch, or severe overfitting to the validation set, rendering it unsuitable for production.

These preliminary findings confirm the feasibility of integrating deep learning into the system. The selection of VGG16 for the current prototype was based on its strong, reliable performance and favorable balance between accuracy and computational requirements. The established high accuracy of VGG16 provides a crucial proof-of-concept for accurately identifying crops before the subsequent API calls to Gemini for recommendations and Kindwise for diagnosis. These metrics serve as a vital baseline, and the full and conclusive evaluation, along with any necessary future model optimization (potentially moving to a model like ResNet50 after thorough resource profiling), will be finalized upon the successful completion of the end-to-end system integration and user acceptance testing phases.

2.Rainfall Prediction Performance

Table 5.2: Rainfall occurrence prediction performance on Dataset-1 (Mumbai + Ratnagiri)

Algorithm	**Accuracy**	**Precision**	**Recall**	**F1-Score**
SVR	0.9286	0.9257	0.8954	0.9103
LightGBM	0.9232	0.9087	0.9007	0.9047
GRU	0.9230	0.9195	0.8873	0.9031

Table 5.3: Rainfall occurrence prediction performance on Dataset-2 (Mumbai + Ratnagiri + Thane + Pune)

Algorithm	**Accuracy**	**Precision**	**Recall**	**F1-Score**
SVR	0.9147	0.9310	0.8725	0.9008
LightGBM	0.9155	0.9232	0.8832	0.9027
GRU	0.9138	0.9166	0.8863	0.9012

Tables 5.2 and 5.3 show the performance of SVR, LightGBM, and GRU models in rainfall prediction based on the two datasets used: Dataset-1, comprising Mumbai and Ratnagiri, and Dataset-2, comprising Mumbai, Ratnagiri, Thane, and Pune cities. In the case of Dataset-1, SVR performed better than the other models, with the highest accuracy of 0.9286 and an F1-score of 0.9103. Although both LightGBM and GRU performed equally, SVR showed a slight edge over the other models, indicating its ability to deal with the nonlinear relationships

found in rainfall data. In the case of Dataset-2, SVR showed a slight decline in performance, with an accuracy of 0.9147 and an F1-score of 0.9008, while both LightGBM and GRU performed equally but showed a slight inferiority compared to SVR.

Chapter 6

Testing

This chapter presents the testing and validation of the SoilStream system to ensure its reliability, accuracy, and overall performance under real-world conditions. The system was evaluated across multiple components, including sensor data acquisition, rainfall prediction logic, cloud communication, and automated pump control. Various test cases were designed to verify the correctness of input-output behavior, system responsiveness, and fault handling. The results obtained from testing demonstrate the effectiveness of the integrated AIoT architecture in delivering accurate predictions, efficient irrigation decisions, and seamless interaction between hardware and software components.

6.1 Software Testing

Software testing constitutes a foundational and non-negotiable phase in the development lifecycle of the Soilstream - A Smart Solar-Powered IoT System for Real-Time Soil Moisture Monitoring and Automated Irrigation, ensuring the accuracy, reliability, and utility of its AI-driven agricultural solutions. This process involves the systematic verification and validation of all software components—from image processing models to API integrations—to confirm they function precisely as intended, meet the specific needs of modern farming, and remain robust under diverse operating conditions. Testing is generally classified into various methodologies, including static and dynamic analysis, with standard methods comprising unit testing, integration testing, system testing, regression testing, and user acceptance testing. Each method plays a critical role in minimizing defects and validating functionality before the system is deployed to users.

For the proposed rainfall prediction and smart monitoring system, adopting a comprehensive testing methodology is essential due to the integration of both advanced software models and real-time hardware components. The system utilizes machine learning models such as Support Vector Regression (SVR), LightGBM, and GRU for accurate rainfall occurrence prediction, while also incorporating hardware modules including the ESP32 microcontroller, BME280 sensor for temperature, humidity, and pressure monitoring, capacitive soil moisture sensor for soil condition analysis, and a relay-controlled water pump for automated irrigation. Any inconsistencies in sensor data acquisition, hardware-software communication, or model predictions could lead to inaccurate forecasts and improper irrigation decisions. Therefore, rigorous testing procedures are implemented to validate the reliability, accuracy, and responsiveness of the entire system. This includes verifying the precision of sensor readings, ensuring stable data transmission from the ESP32 to the prediction system, testing the re-

sponsiveness of the relay and pump mechanism, and evaluating model performance across multiple datasets. Such an integrated validation approach ensures that the complete pipeline, from real-time environmental sensing to intelligent prediction and automated actuation, operates efficiently and delivers dependable results.Selecting a rigorous testing methodology is paramount due to the integration of complex Artificial Intelligence (AI) models and external APIs. The system relies on the VGG16 model for accurate crop identification and the Kindwise API for precise disease diagnosis. Failures in these components could lead to incorrect recommendations, impacting crop health and yield. Therefore, comprehensive validation strategies are necessary to confirm the accuracy, responsiveness, and reliability of the entire information pipeline, ensuring that every feature, from real-time crop classification to personalized treatment plans, performs optimally.

6.2 Functional Testing

Functional testing is one of the most effective and applicable methodologies for the Soilstream as it specifically verifies that the system's features operate in accordance with defined project requirements. This method focuses on testing each function of the software by simulating a user's action—such as uploading a photo—and evaluating the resultant output against the expected results. Functional testing primarily emphasizes the end-to-end user experience, ensuring that all data inputs, model inferences, and API responses are handled correctly.

Test Case ID	Module	Description	Steps	Expected Result	Status
TC-01	Crop Recognition & Recommendation (Objective 3)	Verify the system correctly identifies a crop from an image (VGG16) and provides relevant soil-water recommendations using the Gemini API.	1. User uploads a clear image of a specific crop (e.g., Cashew plant). 2. Verify the VGG16 model output identifies the plant correctly. 3. Input identified crop and real-time soil-moisture data into the Gemini API. 4. Display the resulting crop-specific soil-water condition recommendation to the user.	The system accurately identifies the crop (e.g., "Cashew"). A coherent, relevant, and actionable soil-water-condition recommendation is displayed based on the crop and input moisture data.	Passed
TC-02	Pest/Disease Diagnosis & Treatment (Objective 4)	Verify the system receives a disease diagnosis via Kindwise API and generates tailored treatment suggestions using the Gemini API.	1. User uploads an image of a crop showing signs of a disease (e.g., fungal spots). 2. Verify the image is sent to the Kindwise API and the specific diagnosis is received. 3. Input the received disease diagnosis (e.g., "Late Blight") into the Gemini API. 4. Display the disease name and the Gemini-generated tailored treatment suggestions in the app.	The display shows the specific disease/pest (e.g., "Late Blight"). A coherent, relevant, and actionable treatment plan, generated by the Gemini API, is displayed based on the diagnosis.	Passed
TC-03	IoT-Based Environmental Monitoring & Automated Irrigation	Verify that the system accurately collects real-time environmental data using sensors and triggers automated irrigation using relay and pump.	1. Initialize the ESP32 microcontroller and connect BME280 and capacitive soil moisture sensor. 2. Capture real-time data (temperature, humidity, pressure, soil moisture). 3. Transmit sensor data to the system for processing. 4. Check if soil moisture is below threshold. 5. Verify that relay activates the water pump automatically. 6. Display sensor readings and irrigation status in the application.	The system accurately displays real-time environmental data. When soil moisture falls below threshold, the relay activates the pump automatically, ensuring proper irrigation. The process is seamless and responsive.	Passed

Table 6.1: Functional Testing Table for SoilStream System

The choice of functional testing for this project is justified by the need for precise validation of the two core, AI-integrated features as shown in above Table 6.1:

Crop Recommendation (Objective 3): Verification that the image is correctly processed by VGG16, the crop identity is accurately relayed to the Gemini API, and the resulting soil/water recommendations are relevant and coherent.

Disease Diagnosis and Treatment (Objective 4): Confirmation that the image is successfully sent to the Kindwise API for diagnosis, that the diagnosis is correctly captured, and that the subsequent treatment plan generated by the Gemini API is relevant and actionable.

IoT-Based Monitoring and Automation(objective 1): Validation that the ESP32 microcontroller accurately collects real-time environmental data from the BME280 sensor and

capacitive soil moisture sensor, and that the relay-controlled water pump is triggered appropriately based on predefined thresholds.

Rainfall Prediction and Smart Irrigation (Objective 2): Validation that the system accurately predicts rainfall occurrence using machine learning models (SVR, LightGBM, GRU) based on environmental inputs, and integrates this prediction with the hardware system to control irrigation. It ensures that when low rainfall probability and low soil moisture conditions are detected, the ESP32 triggers the relay-controlled water pump appropriately, while preventing unnecessary irrigation during predicted rainfall, thereby optimizing water usage and system efficiency.

By simulating real-life scenarios, functional testing ensures that the system performs consistently across different environments, providing farmers with a reliable and effective advisory tool. The functional testing of this system is summarized in Table 5.1 (Functional Testing Table for Soilstream), covering the core agricultural advisory features and critical AI/API modules of the application.

Chapter 7

Conclusion

In conclusion, this project presents the comprehensive design of SoilStream - A Smart Solar-Powered IoT System for Real-Time Soil Moisture Monitoring and Automated Irrigation is tailored to address the agricultural challenges of the Thane region. The system integrates a low-cost, solar-powered on-field hardware unit with a Google Firebase backend and an intuitive React Native mobile application, creating a seamless bridge between real-time data collection and intelligent decision-making. By employing machine learning algorithms for rainfall prediction, crop identification, it advances the concept of data-driven precision agriculture, transforming environmental data into actionable insights that enable optimized irrigation, early pest detection, and efficient resource use. The system directly addresses issues such as water scarcity, unpredictable weather, and crop vulnerability, empowering farmers with affordable, adaptive technology that promotes proactive farming decisions. Its implementation promises higher yields, reduced wastage, and sustainable water management, contributing to long-term food security and economic stability. While the prototype demonstrates a robust and scalable framework, future improvements could include additional sensors, wider AI model coverage, and predictive analytics for yield forecasting. Ultimately, SoilStream - A Smart Solar-Powered IoT System for Real-Time Soil Moisture Monitoring and Automated Irrigation represents a step towards intelligent, sustainable, and inclusive agriculture, ensuring that farming evolves through precision, efficiency, and resilience.

Chapter 8

Future Scope

Looking ahead, the future scope of this project is substantial and holds the potential to transform SoilStream - A Smart Solar-Powered IoT System for Real-Time Soil Moisture Monitoring and Automated Irrigation into a fully-fledged, intelligent, and scalable farm management platform. The on-field hardware can be further enhanced by integrating a wider array of advanced sensors capable of monitoring essential soil parameters such as NPK (Nitrogen, Phosphorus, and Potassium) levels, pH balance, and electrical conductivity. These additional data points would enable the system to perform precise, automated nutrient management, ensuring optimal soil fertility and supporting higher-quality crop growth. Moreover, this enhanced sensor suite could be deployed in the form of a distributed sensor network across large farmlands, facilitating zonal control—where irrigation, fertilization, and treatment schedules can be customized for different sections of the farm based on specific soil and crop conditions.

The scalability of the system can be further strengthened by integrating it with LoRaWAN-based communication networks or edge computing nodes, allowing real-time data transmission even in remote rural areas with limited connectivity. The data gathered over time would serve as a valuable resource for developing more advanced predictive analytics models, including AI-driven yield forecasting, crop rotation planning, and early anomaly detection. These insights could empower farmers to plan their agricultural cycles with greater precision and reduce losses due to unforeseen climatic or pest-related issues. Additionally, the system could evolve into a comprehensive decision-support tool by incorporating external data sources such as real-time market price APIs, satellite or drone-based aerial imagery, and climate prediction models. This integration would enable the generation of holistic, data-driven insights covering both production and economic aspects of farming. For instance, farmers could be guided on when to plant, irrigate, harvest, and sell their crops to maximize both yield and profitability.

Finally, future iterations of the SoilStream platform could include features such as voice-based AI assistants for farmers, multilingual mobile interfaces, and community-driven knowledge sharing platforms where farmers can exchange experiences and insights. By combining these advancements, SoilStream can evolve from a smart irrigation system into a truly intelligent and inclusive agricultural ecosystem—one that supports data-informed decision-making, promotes sustainability, and enhances the resilience of agriculture in regions like Thane and beyond.

Bibliography

[1] R. Rajalakshmi and P. Mahalakshmi, "IoT based crop field monitoring and irrigation automation system," *2016 10th International Conference on Intelligent Systems and Control (ISCO)*, Coimbatore, India, 2016, pp. 1-6.

[2] K. Rawal, "IoT based smart irrigation system," *International Journal of Computer Applications*, vol. 159, no. 8, pp. 7-11, 2017.

[3] Á. G. Casado, F. A. Cureses, L. A. J. Laria, J. L. P. Riesgo, and F. S. J. T. de Lera, "Intelligent irrigation system based on Arduino," *arXiv preprint arXiv:1803.00097*, 2018.

[4] H. S. Rout and P. K. Mishra, "Solar powered smart irrigation system using IoT," *2018 2nd International Conference on Inventive Systems and Control (ICISC)*, Coimbatore, India, 2018, pp. 586-590.

[5] C. S. Devan, B. T. M. Prathyusha, T. Keerthana, and P. S. B. S. S. N. Neha, "A proof-of-concept: IoT based solar-powered smart irrigation system," *Journal of Physics: Conference Series*, vol. 2054, no. 1, p. 012074, 2021.

[6] M. H. Ramli and W. A. Jabbar, "Portable solar powered IoT-enabled smart irrigation system," *Sustainable Energy Technologies and Assessments*, vol. 52, p. 102127, 2022.

[7] S. S. Ali, M. A. Ali, M. Z. Khan, and M. F. N. Khan, "IoT and Solar Based Smart Irrigation System," *International Research Journal of Modernization in Engineering Technology and Science*, vol. 7, no. 6, pp. 1245-1250, 2025.

[8] G. Talaviya, D. Shah, N. Patel, H. Yagnik, and M. Shah, "Implementation of artificial intelligence in agriculture for optimisation of irrigation and application of pesticides and herbicides," *Artificial Intelligence in Agriculture*, vol. 4, pp. 58-73, 2020.

[9] A. Tace, E. K. Tace, and M. T. Tace, "Machine learning based smart irrigation system using IoT," *Procedia Computer Science*, vol. 204, pp. 696-703, 2022.

[10] T. T. Esmail, S. H. E. A. E. Atty, and H. A. Ibrahim, "Smart irrigation system using IoT and machine learning," *2023 5th International Conference on Smart Systems and Inventive Technology (ICSSIT)*, Tirunelveli, India, 2023, pp. 1157-1162.

[11] S. Balamurali, P. M. S. Kumar, G. U. Devi, and P. S. Venkateswaran, "A solar-powered IoT-based irrigation system with integrated rainfall forecasts using aerosol data," *Environmental Science and Pollution Research*, 2025.

[12] Y. Liu, S. Zhao, and A. Rezaeipanah, "An intelligent and automatic irrigation system based on IoT and fuzzy control technology," *Scientific Reports*, vol. 15, no. 1, p. 12345, 2025.

[13] M. R. Al Mamun, A. K. Ahmed, and S. M. Upoma, "IoT-enabled solar-powered smart irrigation for precision agriculture," *Heliyon*, vol. 11, no. 2, p. e37007, 2025.

[14] J. M. A. Capcha-Ochoa, D. A. R. O. C. P. G. Ochoa, K. L. M. Flores, and D. A. J. S. Poma, "Smart Irrigation System with IoT, Machine Learning, and Solar Power for Efficient Plant Care," *Engineering Science Journal*, vol. 6, no. 1, pp. 1-13, 2025.

[15] A. Kunt, "IoT and AI-based smart irrigation model," *arXiv preprint arXiv:2506.11835*, 2025.

Appendices

This section provides the detailed system setup and deployment guide for the SoilStream project. It outlines the step-by-step procedure to configure the hardware, backend, frontend, and cloud components required to successfully run the system.

Appendix A: Hardware Setup

Connect the ESP32 to your PC via USB. Open `hardware/soilstream.ino` in Arduino IDE and install the following libraries via the Library Manager:

- Firebase ESP32 Client (by Mobizt)
- Adafruit BME280 Library
- Adafruit Unified Sensor

Update the credentials in the `.ino` file before flashing:

```
#define WIFI_SSID      "your_wifi_name"
#define WIFI_PASSWORD  "your_wifi_password"
#define API_KEY        "your_firebase_api_key"
#define DATABASE_URL   "https://your-project.firebaseio.com/"
#define DEVICE_ID      "DEVICE_001"
```

Flash the code to the ESP32 using Arduino IDE. Once flashed, the device will automatically connect to Wi-Fi and begin transmitting sensor data to Firebase.

Appendix B: Backend Setup

Ensure Python 3.9+ is installed on your system:

```
python --version
pip --version
```

Clone the Repository:

```
git clone https://github.com/RajChoudhary99/SoilStream.git
cd SoilStream
```

Download Model Weights — the VGG16 model file exceeds GitHub's file size limit and must be downloaded separately. Download `model_vgg16.h5` from the Google Drive link provided in the README and place it at: `software/backend/model/model_vgg16.h5`

Navigate to the backend directory and install dependencies:

```
cd software/backend
pip install -r requirements.txt
```

Run the Flask backend server:

```
python app.py
```

The backend will start at `http://localhost:5000` and expose the `/predict` endpoint for crop analysis.

Appendix C: Frontend Setup

Ensure Node.js 18+ is installed:

```
node --version
npm --version
```

Install Expo CLI globally:

```
npm install -g expo-cli
```

Navigate to the software directory and install dependencies:

```
cd software
npm install
```

Configure environment variables by creating a `.env` file inside `software/`:

```
FIREBASE_API_KEY=your_firebase_api_key
GEMINI_API_KEY=your_gemini_api_key
KINDWISE_API_KEY=your_kindwise_api_key
```

Run the mobile application:

```
npx expo start
```

Scan the QR code using the **Expo Go** app on your smartphone, or press `a` for Android emulator.

Appendix D: Firebase Cloud Functions Setup

Ensure Firebase CLI is installed:

```
npm install -g firebase-tools
firebase login
```

Navigate to cloud functions and install dependencies:

```
cd cloud_functions/functions
npm install
```

Deploy to Firebase:

```
firebase deploy --only functions
```

Appendix E: Running the Complete System

To run the full SoilStream system, follow these steps in order:

1. Connect the ESP32 hardware to power (USB or solar)
2. Open **Terminal 1** and run the backend:

   ```
   cd software/backend
   python app.py
   ```

3. Open **Terminal 2** and run the frontend:

   ```
   cd software
   npx expo start
   ```

4. Open the **Expo Go** app on your phone and scan the QR code
5. The dashboard will display live sensor data from Firebase within seconds

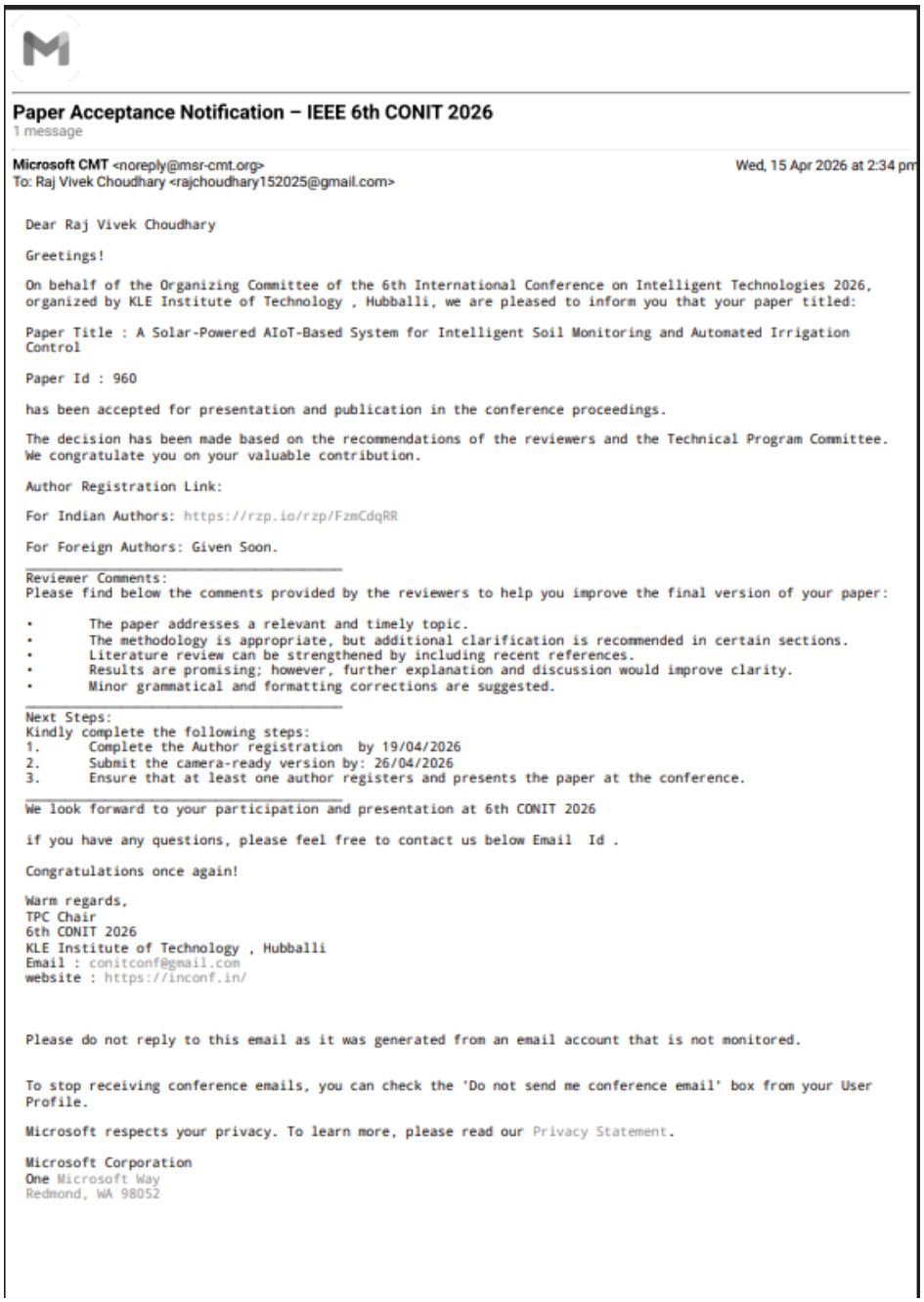
Publication

Paper entitled **"SoilStream: A Solar-Powered AIoT-Based System for Intelligent Soil Monitoring and Automated Irrigation Control"** is presented at **"IEEE 6th International Conference on Intelligent Technologies"** by **"Raj Choudhary"**, **"Siddesh Patil"**, **"Varun Lad"** and **"Devesh Patil"**.

Patent:
Patent entitled **"SoilStream - A Smart Solar-Powered IoT System for Real-Time Soil Moisture Monitoring and Automated Irrigation"** has been registered with **"Intellectual Property India,Government of India"** ,under under **Application Number:** 202621042187

Copyright:
Copyright entitled **"SoilStream - A Smart Solar-Powered IoT System for Real-Time Soil Moisture Monitoring and Automated Irrigation"** has been registered with the **"Copyright Office,Government of India"**, under **"Registration Number SW-13522/2026-CO"**

Research Paper Acceptance



Paper Acceptance Notification – IEEE 6th CONIT 2026
1 message

Microsoft CMT <noreply@msr-cmt.org> Wed, 15 Apr 2026 at 2:34 pm
To: Raj Vivek Choudhary <rajchoudhary152025@gmail.com>

Dear Raj Vivek Choudhary

Greetings!

On behalf of the Organizing Committee of the 6th International Conference on Intelligent Technologies 2026, organized by KLE Institute of Technology , Hubballi, we are pleased to inform you that your paper titled:

Paper Title : A Solar-Powered AIoT-Based System for Intelligent Soil Monitoring and Automated Irrigation Control

Paper Id : 960

has been accepted for presentation and publication in the conference proceedings.

The decision has been made based on the recommendations of the reviewers and the Technical Program Committee. We congratulate you on your valuable contribution.

Author Registration Link:

For Indian Authors: https://rzp.io/rzp/FzmCdqRR

For Foreign Authors: Given Soon.

Reviewer Comments:
Please find below the comments provided by the reviewers to help you improve the final version of your paper:

- The paper addresses a relevant and timely topic.
- The methodology is appropriate, but additional clarification is recommended in certain sections.
- Literature review can be strengthened by including recent references.
- Results are promising; however, further explanation and discussion would improve clarity.
- Minor grammatical and formatting corrections are suggested.

Next Steps:
Kindly complete the following steps:
1. Complete the Author registration by 19/04/2026
2. Submit the camera-ready version by: 26/04/2026
3. Ensure that at least one author registers and presents the paper at the conference.

We look forward to your participation and presentation at 6th CONIT 2026

if you have any questions, please feel free to contact us below Email Id .

Congratulations once again!

Warm regards,
TPC Chair
6th CONIT 2026
KLE Institute of Technology , Hubballi
Email : conitconf@gmail.com
website : https://inconf.in/

Please do not reply to this email as it was generated from an email account that is not monitored.

To stop receiving conference emails, you can check the 'Do not send me conference email' box from your User Profile.

Microsoft respects your privacy. To learn more, please read our Privacy Statement.

Microsoft Corporation
One Microsoft Way
Redmond, WA 98052