NaiHydro: A Mobile Application for Environment Condition Monitoring in IoT-Enabled Hydroponic Farms for Lettuce Using Random Forest and XGBoost

Student Name: Nancy Mungai

Admission Number: 150912

Supervisor: Tiberius Tabulu

Submitted to the School of Computing and Engineering Sciences in partial fulfillment of the requirements for the completion of a Bachelor of Science degree in Informatics and Computer Science

School of Computing and Engineering Sciences

Strathmore University

Nairobi, Kenya

November 2025

Declaration and Approval

I declare that this project has not been presented to Strathmore University or any other Institution for the fulfilment of a Degree in a Bachelor of Science in Informatics and Computer Science or any other academic Degree. To the best of my knowledge, this research documentation does not include any content that has been previously published or authored by someone else, except where proper citations and acknowledgments have been provided within the document.

Student Admission Number: 150912

Sign: 

Date: 28/11/2025

Supervisor's Signature:

Sign: 

Date: 28/11/2025

Acknowledgment

I am deeply grateful to God for granting me the strength and clarity to carry out this research. I wish to extend my heartfelt appreciation to my supervisor, Mr. Tiberius Tabulu, for his unwavering support, thoughtful guidance, and invaluable feedback throughout the course of this project. I am particularly grateful to my lecturer, Dr. Esther Khataka, for her commitment to excellence in teaching and for providing me with the skills and insight that have greatly contributed to the success of this project. Their expertise and patience have been instrumental in bringing this project to fruition.

Abstract

Climate change and rising food demand are making traditional farming less reliable. Hydroponic systems grow plants in nutrient-rich water without soil but require continuous monitoring of parameters such as temperature, pH and nutrient levels to avoid setbacks that reduce yields. Manual checks are slow, error-prone and hard to scale, creating a gap in reliable crop production.

This project presents a smart condition-monitoring system for lettuce growing in hydroponic farms, which combines IoT sensors, an ESP32 prototype, an ensemble machine model built with XGBoost and Random Forest and a mobile application built with flutter framework connected to Firebase for storage. The ensemble model was trained on a hydroponic dataset containing sensor readings, enabling it to detect anomalies and predict anomalous conditions. Farmers get real-time dashboards, alerts and actionable recommendations on their phones to support timely decisions.

The system was developed using an agile prototyping approach, which supported continuous refinement as new components were tested and integrated. The implementation achieved all intended objectives. The IoT prototype reliably synchronised sensor data with the cloud database, and the mobile application consistently displayed accurate real-time readings. In addition, the system generated prompt push notifications whenever the machine learning model detected abnormal conditions, demonstrating successful coordination across all modules.

Overall, the solution emphasises affordability, scalability, and suitability for resource-limited farming environments. By automating environmental monitoring and providing intelligent alerts, the system strengthens decision-making for smallholder farmers and enhances the reliability and sustainability of lettuce production.

Keywords: Hydroponics, IoT, Random Forest, Anomaly Detection, ESP32 microcontroller, **Sustainable Agriculture, XGBoost**

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AI – Artificial Intelligence

EC – Electrical Conductivity

GPIO – General-Purpose Input/Output

IoT – Internet of Things

ML – Machine Learning

MVP – Minimum Viable Product

OOAD – Object-Oriented Analysis and Design

PVC – Poly Vinyl Chloride

UAT – User Acceptance Testing

UI – User Interface

XGBoost – Extreme Gradient Boost

Chapter 1: Introduction

1.1 Background of Study

Hydroponic farming is a soilless method of crop production that offers a sustainable alternative to traditional agriculture. This type of especially fitting in regions affected by unpredictable rainfall, prolonged droughts, poor soil fertility, and limited arable land. By relying on nutrient-rich water solutions under controlled conditions, hydroponics reduces dependence on natural soil and mitigates the risks of crop failures, food shortages, and rising production costs that have become common in soil-based farming. These challenges are further intensified by climate change, which disrupts weather patterns and increases uncertainty in conventional practices (Croft et al., 2017).

Urbanization is also contributing to the problem by reducing the amount of land available for cultivation. In some areas, productive farmland is now occupied by buildings, leaving farmers with fewer options to grow food (Mosaad et al., 2023a). Moreover, soil degradation, caused by the overuse of fertilizers and harmful chemicals, has lowered soil fertility and increased dependency on costly inputs. These factors combined have highlighted the urgent need for more sustainable and efficient farming practices.

Hydroponics presents a promising alternative by accelerating crop development and significantly reducing resource consumption, using up to 90% less water and 75% less land than traditional methods (Croft et al., 2017). The absence of soil also reduces risks of pest infestations and soil-borne diseases (Kundal et al., 2024). Its adaptability makes hydroponics ideal for space-constrained or non-arable locations such as rooftops, balconies, and controlled greenhouses.

In Kenya, hydroponics has shown potential not just for improving food security but also for boosting nutrition. For instance, crops like Amaranthus cruentus (vegetable amaranth), when grown hydroponically, have been found to accumulate more essential nutrients such as iron, zinc, and carotenoids which are nutrients often missing from Kenyan diets (Croft et al., 2017). Despite these benefits, managing a hydroponic system effectively can be challenging. It requires constant monitoring of variables like pH, temperature, water levels, and nutrient concentration, which can be labor intensive and susceptible to human errors.

The integration of Machine Learning and the Internet of Things (IoT) into hydroponic farming presents a promising technological solution, enabling more precise monitoring and control of crop growth conditions by farmers. Sensors collect real-time data, while ML algorithms analyze the information to detect problems early, suggest corrections, and even predict outcomes (Mamatha et al., 2023). Such smart systems can reduce labour, optimize utilization of resources, and improve crop yields.

1.2 Problem Statement

Hydroponic farming depends on strict control of conditions such as pH, temperature, nutrient concentration, and water levels. Small deviations in these parameters can quickly lead to stunted growth, disease, or crop failure (Bhandari et al., 2024). The main challenge is that most smallholder farmers still rely on manual monitoring, which is slow, error-prone, and ineffective for detecting early signs of imbalance. Although advanced automated systems exist, they are designed for large-scale farms and require high costs and technical expertise, making them unsuitable for small operations (Mamatha et al., 2023; Bua et al., 2024).

As a result, anomalies such as pH drift, nutrient fluctuations, temperature stress, water shortages, and equipment failures often go unnoticed until they severely reduce yields (Anusha et al., 2025). This lack of timely and accurate monitoring undermines the potential of hydroponic farming to provide reliable and sustainable food production.

To address this gap, this project implements a smart condition-monitoring system in growing lettuce hydroponically. It integrates low-cost IoT sensors with an ensemble ml model that uses Random Forest and XGBoost algorithms to analyze real-time data and detect anomalies such as low water level, temperature and unsuitable pH early. The system is also integrated with a Flutter-based mobile application that is connected to Firebase and delivers instant alerts and visual dashboards, offering farmers accessible and actionable insights.

1.3 General Objective

To design and develop a smart hydroponic monitoring system that combines IoT technologies with an ensemble machine learning (ML) model built with XGBoost and

Random Forest Algorithms, providing farmers with a mobile-based platform for real-time tracking and effective management of essential growing conditions.

1.3.1 Specific Objectives

i. To investigate hydroponic farming principles and assess the current landscape of smart hydroponic practices, identifying key challenges in their adoption.

ii. To analyze how XGBoost, Random Forest, and IoT technologies can be applied to monitor and detect anomalies in hydroponic farming environments.

iii. To design a smart hydroponic monitoring system architecture that integrates IoT with an ensemble ML model built using XGBoost and Random Forest algorithms for effective data handling and decision support.

iv. To develop a functional smart hydroponic monitoring system that enables farmers to track and manage essential growing conditions through a mobile application supported by IoT and ensemble ML capabilities.

v. To test and evaluate the performance of the integrated system using data collected from a the IoT based hydroponic setup.

1.3.2 Research Questions

i. What are the core principles of hydroponic farming, and what challenges exist in the current adoption of smart hydroponic practices?

ii. How can XGBoost, Random Forest and IoT technologies be effectively applied to monitor and detect anomalies in hydroponic systems?

iii. How can a smart hydroponic monitoring system architecture be designed to integrate IoT technologies with an ensemble ML model using XGBoost and Random Forest algorithms for efficient data management and decision support?

iv. How can an integrated smart hydroponic monitoring system be developed to help farmers track and manage essential growing conditions through a mobile application?

v. How effectively does the integrated system perform when tested using data collected from the IoT based hydroponic setup?

1.4 Justification

Hydroponic farming is gaining importance as farmers face rising challenges such as unpredictable weather patterns, water scarcity, and declining arable land. These pressures make traditional soil-based farming increasingly unsustainable in the context of climate change (Gumisiriza et al., 2022; Nalwade et al., 2017). In response to these challenges, hydroponic farming offers a favorable alternative that uses less water and space while producing higher yields (Gumisiriza et al., 2022). However, hydroponic systems require close monitoring and precise control of factors like pH, temperature, water level, and nutrient concentration. When these factors shift outside the optimal range, crops are highly vulnerable to rapid failure, which poses a major risk for small-scale farmers who cannot absorb such losses (Suresh et al., 2025; Mamatha et al., 2023).

This project bridged that gap by developing a smart hydroponic monitoring system that integrates IoT with ML, and mobile technology to provide real-time feedback and guidance to farmers. By using an ensemble ML model built with XGBoost and Random Forest algorithms, the system detects anomalies in sensor data to help farmers identify potential issues early and improve decision-making (Mosaad et al., 2023; Suresh et al., 2025). IoT components collect real-time environmental data, while a Flutter-based mobile application served as the user interface, allowing farmers to easily view system data, receive alerts, and manage the setup remotely (Mamatha et al., 2023). Firebase was used for data storage, synchronization, and notification delivery, strengthening the system's responsiveness and ensuring timely updates to the user. The solution offers a practical, cost-effective, and accessible approach to supporting sustainable hydroponic farming practices through modern technology.

1.5 Scope and Limitations

This section outlines the extent of the project's coverage and the constraints encountered during its development. clarity on what the system achieves and the boundaries within which it operates.

1.5.1 Scope of the Project

This project focuses on designing and developing a smart hydroponic monitoring system that integrates IoT technologies with ML to support the cultivation of lettuce in a controlled hydroponic environment. The system monitored key parameters such as pH, temperature, water level, and nutrient concentration using basic sensors. A mobile application built with Flutter allowed users to view sensor readings, receive alerts, and access recommendations.

The ML model, built using an ensemble of XGBoost and Random Forest algorithms, processed tabular sensor data to detect anomalies and provide actionable insights. An IoT testbed was used to validate the system's functionality and ensure consistency of sensor-data collection and communication. Firebase served as the cloud database to enable real-time data exchange between components.

The scope of the project is limited to monitoring lettuce crops and allowing the user to control the water pump; it does not include automatic adjustment of pH, nutrient levels, or support for other plant types. The system was only tested in a controlled environment as an initial implementation.

1.5.2 Limitations in the Project

One of the main limitations of this project is the restricted capabilities of the available advanced IoT hardware. Since the project is being developed as a prototype, only a small-scale demo system with basic sensors was built. This may not fully reflect the complexity of real-world hydroponic setups. In addition, only the water pump was automated in this prototype, meaning other control functions were not fully automated, limiting the system's ability to represent a complete end-to-end automation process.

Another limitation is the simulated environment for testing. The system was tested under controlled conditions, which may not capture all the variables found in actual farming settings, such as outdoor temperature changes, power interruptions, leakages or large-scale nutrient management.

The ML model was trained on a dataset with limited diversity in operational scenarios. This reduced the model's exposure to a wide range of real-world variations, which may

affect its ability to generalize effectively in more complex or unpredictable hydroponic environments.

1.5.3 Delimitations in the Project

This project monitored only lettuce grown in a hydroponic setup. Other plant types such as fruits or root crops are excluded due to their differing environmental and nutrient needs.

The system was tested only in a lab-based hydroponic environment rather than in field or commercial farm conditions. This choice ensures controlled evaluation but limits real-world applicability.

Random Forest and XGboost were deliberately selected over more complex models such as deep learning autoencoders. While autoencoders can be effective for anomaly detection, they are computationally intensive and less suited for deployment on lightweight hardware like the ESP32 microcontroller.

Chapter 2: Literature Review

2.1 Introduction

This chapter reviews literature on smart hydroponic systems, focusing on the challenges faced by farmers. It examines existing smart farming solutions, highlighting their strengths and weaknesses, and identifies gaps in current technologies. The chapter ends with a conceptual framework showing how the IoT-based system, integrated with Random Forest and XGBoost, aims to address these gaps.

2.2 Hydroponic Farming: A Global and Kenyan Perspective

Hydroponic farming is a method of growing crops that does not involve the use of soil but instead uses nutrient-rich water solutions to provide essential minerals directly to plant roots. This approach allows for efficient resource use, faster crop growth, and the ability to farm in areas with limited land or poor soil quality. As the world faces increasing challenges related to climate change, urbanization, and food insecurity, hydroponics has emerged as a sustainable solution for food production.

Globally, countries have adopted hydroponic farming and enhanced it with smart technologies. In South Korea, AI-based systems have been developed to automate greenhouse management by optimizing lighting, temperature, and nutrient distribution, helping boost crop efficiency and reduce human labor (Anjali et al., 2025). In China, large-scale vertical farms utilize IoT and automation to manage lettuce production with minimal environmental footprint (Anusha et al., 2025). In Europe, smart greenhouses are common in countries like Italy and the Netherlands, where they support year-round tomato and leafy green cultivation under controlled environments (Olajide et al., 2024).

In Asia, nations such as India and Malaysia are applying IoT-based systems to monitor pH, temperature, and humidity in hydroponic environments. These systems enable farmers to receive real-time alerts and automate irrigation schedules, especially in peri-urban regions where land and water are scarce (Anusha et al., 2025). In Southeast Asia, temperature-sensitive crops like strawberries are now being grown in controlled environments using smart vertical hydroponics, improving yield quality and reducing crop failures(Anjali et al., 2025).

Across Africa, hydroponics is emerging as a viable solution for sustainable agriculture, particularly in Kenya, Nigeria, and Ghana. In Nairobi and Kisumu, for example, hydroponic kits are being used to grow spinach and lettuce in urban estates, addressing the challenge of space and food insecurity. Private agribusinesses and innovation hubs are increasingly offering training and modular systems to support this transition (Olajide et al., 2024). These systems are often solar-powered and leverage mobile applications for monitoring.

Consumer trends are also supporting hydroponic farming. A study in the Caribbean found that consumers preferred hydroponically grown lettuce due to its consistent appearance, shelf life, and lower pesticide use (Anusha et al., 2025). In Kenya, similar trends are observed among health-conscious urban consumers who seek safer and more sustainable vegetables.

Despite growing interest, the adoption of hydroponic systems still faces barriers such as high setup costs, limited technical training, and inconsistent government support. However, with increasing digitization and support from innovation-driven policies, hydroponics is poised to play a larger role in Africa's food systems (Olajide et al., 2024).

2.2.1 Challenges in Hydroponic Farming

Hydroponics offers a promising solution to the limitations of traditional agriculture by enabling faster crop cycles, efficient water use, and urban adaptability. However, several challenges limit its adoption, particularly in low-resource settings.

A major obstacle is the high setup cost. Essential components like nutrient pumps, sensors, lighting, and automation systems make the initial investment unaffordable for many small-scale farmers. Even basic cloud-connected systems can incur operational costs of up to $75 monthly, which adds to the financial burden (Ali et al., 2024; Rosca et al., 2025).

Technical knowledge is also critical. Farmers must understand how to manage water pH, nutrient levels, and system calibration. Inadequate training often results in poor yields or system failure, especially in areas with limited access to agricultural support services (Ali et al., 2024).

Infrastructure limitations further complicate implementation. Access to affordable materials and stable internet or electricity is limited in many rural areas, making it difficult to maintain automated hydroponic systems (Rosca et al., 2025).

Lastly, energy consumption is a serious concern. Most systems depend on artificial lighting and environmental control, which increases power usage. In regions with high electricity costs, this reduces sustainability. Solar-powered alternatives and low-energy designs are recommended to support off-grid or low-income users (Dhal et al., 2024).

2.3 Integration of XGboost, Random Forest and IoT for Anomaly Detection in Hydroponics

Smart hydroponic farming increasingly utilizes IoT technologies to monitor critical environmental conditions such as pH, humidity, temperature, and nutrient levels. Devices such as microcontrollers, sensors, and cloud gateways form a responsive ecosystem capable of real-time monitoring and control. These systems enable constant data flow from the hydroponic environment to cloud-based applications, offering unprecedented visibility and responsiveness in agricultural operations (Rahman et al., 2024). However, issues such as high setup costs, intermittent connectivity in rural zones, and energy demands of greenhouse lighting systems often limit the widespread adoption of these solutions, particularly among small-scale farmers (Bakirov et al., 2025).

Machine learning plays a key role in analyzing the data collected from IoT devices to detect patterns, predict growth, and identify anomalies in crop behavior. Random Forest is widely used due to its ability to handle noisy agricultural datasets, provide fast predictions, and maintain high accuracy across classification tasks. It has been applied effectively in detecting water stress, nutrient imbalance, and potential disease outbreaks in crop environments (Shalash Métwalli et al., 2025). Its nature makes it stable and resistant to fluctuations in sensor readings.

XGBoost complements Random Forest by offering a gradient-boosted approach optimized for structured and time-dependent sensor data commonly observed in hydroponic setups. Its strength lies in reducing overfitting, handling high-dimensional inputs, and delivering

efficient performance even with complex datasets. Studies report that integrating XGBoost with continuous sensor streams can achieve above 95% accuracy in identifying abnormal growth patterns in common hydroponic crops such as lettuce and tomatoes (Rajendiran, 2024). However, deploying these models, especially Random Forest on constrained edge devices like the ESP32, may require hybrid edge cloud architectures to manage computational load.

Several recent projects also demonstrate the combined use of IoT and XGBoost in hydroponic and aeroponic systems. A study by Métwalli et al. (2025) implemented an AIoT system using XGBoost for nutritional optimization and anomaly detection, showing improvements in yield and resource conservation. Similarly, Rahman et al. (2024) applied an IoT-to-XGBoost pipeline for crop recommendation and monitoring. While these integrations have proven beneficial, they still face practical challenges including sensor calibration, scalability of deployments, and ensuring secure data exchange across IoT networks.

2.4 Related Works

This section reviews prior studies and technological solutions relevant to smart hydroponic systems. It highlights existing approaches, their contributions, and the gaps that inform the direction of this project.

2.4.1 Growee: Nutrient Automation and Real-Time Monitoring

Growee is a commercial hydroponic automation platform designed to simplify water-based plant growing for home and small-scale growers. It includes smart dosing controllers that automatically manage pH and nutrient levels. The system is managed through a dedicated mobile application that allows real-time monitoring and remote operation of multiple grow units. Key features include modular setup, cloud syncing, and environmental logging, all targeted at non-technical users (Peskett, 2023).

Despite being IoT-enabled, Growee does not currently implement advanced AI models like deep learning. It follows a rules-based automation framework with basic adaptive scheduling, adjusting nutrient delivery based on previous usage data. This restricts its use in more dynamic agricultural settings that require real-time predictive analytics or anomaly

detection. (Nitsan, 2022). An image of the Growee user interface is provided in the appendix.

2.4.2 Niwa Grow Hub: App-Based AI Grow Recipes

Niwa is a smart grow controller that provides users with app-controlled automation for lights, water, and ventilation systems in indoor grow environments. Through the Niwa Grow Hub and its mobile app, users can select plant-specific "grow recipes" which adjust environmental settings over time. The app offers remote control, preset cultivation modes, and real-time monitoring dashboards, making it suitable for hobbyists and early adopters of home-based hydroponics (Hemlata, 2023).

While Niwa provides a form of intelligent environmental adaptation through recipe-based logic, it does not include advanced AI features like machine learning or visual diagnostics. Its primary value lies in accessibility, user interface simplicity, and consistent app-based system feedback (Hemlata, 2023). An image of the Niwa Grow Hub user interface is included in the appendix.

2.4.3 Agrihydroponics: AI-Based Disease Detection in Controlled Environments

Agrihydroponics is an academic prototype that combines environmental sensing with computer vision techniques to detect plant diseases such as leaf curl and nutrient deficiencies in hydroponic lettuce systems. The platform integrates pH, EC, and temperature sensors with an ESP32 processor and a convolutional neural network (CNN) to assess leaf health through captured images. The accompanying mobile application displays real-time environmental data and issues alerts when deviations or anomalies are detected (Ramakrishnam et al., 2022).

This system distinguishes itself by incorporating image recognition, enabling it to detect visual symptoms such as chlorosis and leaf curl before they escalate. Agrihydroponics allows for more complex decisions by correlating image data with sensor readings. However, it remains a simulation-based study and has not been field-tested in long-term deployments, limiting its generalizability and reliability for real-world farming (Dennison, Kumar, et al., 2025). An image of the Agrihydroponics mobile interface is included in the appendix.

2.5 Gaps in Existing Solutions

A comparative review of Growee, Niwa, and Agrihydroponics systems reveals several key limitations that remain unaddressed in current smart hydroponics solutions. One major gap lies in the integration of comprehensive machine learning models. While Growee offers pattern-based automation and Niwa includes adaptive scheduling, both systems rely on rule-based logic rather than true learning algorithms. Agrihydroponics, by contrast, includes a CNN for image analysis but lacks real-world deployment, which questions its scalability and usability beyond lab environments(Hemlata, 2023; Ramakrishnam et al., 2022).

Another critical gap is the lack of multimodal decision-making in existing hydroponic systems. None of the evaluated platforms combine sensor data analysis with a feedback loop that supports real-time anomaly detection and intervention. Growee tracks pH and EC values, and Niwa automates routines, but both systems lack predictive analytics. Agrihydroponics explores deep learning simulations but does not provide a mobile interface or scalable deployment (Dennison et al., 2025; Shareef et al., 2024).

User inclusivity and scalability also remain underdeveloped. Niwa and Growee cater primarily to small or hobbyist setups, with limited expansion for larger installations. Agrihydroponics presents promising technical depth but lacks usability for non-technical users. In all cases, there is no robust validation mechanism for ensuring accuracy across diverse crop types or environmental conditions (Dhal et al., 2022)

2.6 Conceptual Framework

The conceptual framework shows how the smart hydroponics monitoring system works using IoT, machine learning, and a mobile application. The system is designed to help farmers track important growing conditions and respond quickly when problems are detected.

The process begins with IoT sensors that collect data such as pH, temperature, humidity, water and nutrient levels. This data is sent to a cloud server, where it is processed using an ensemble (XGBoost and Random Forest) machine learning model to check for anomalous conditions. Once the model finishes processing, the results are stored in a central database.

If the system detects a problem, the mobile application retrieves the processed results from the database and sends an alert to the user through firebase cloud messaging. This allows the user to act in real time.

This framework supports the goal of building a smart, easy-to-use solution that helps monitor crop conditions and reduce the need for manual checks.

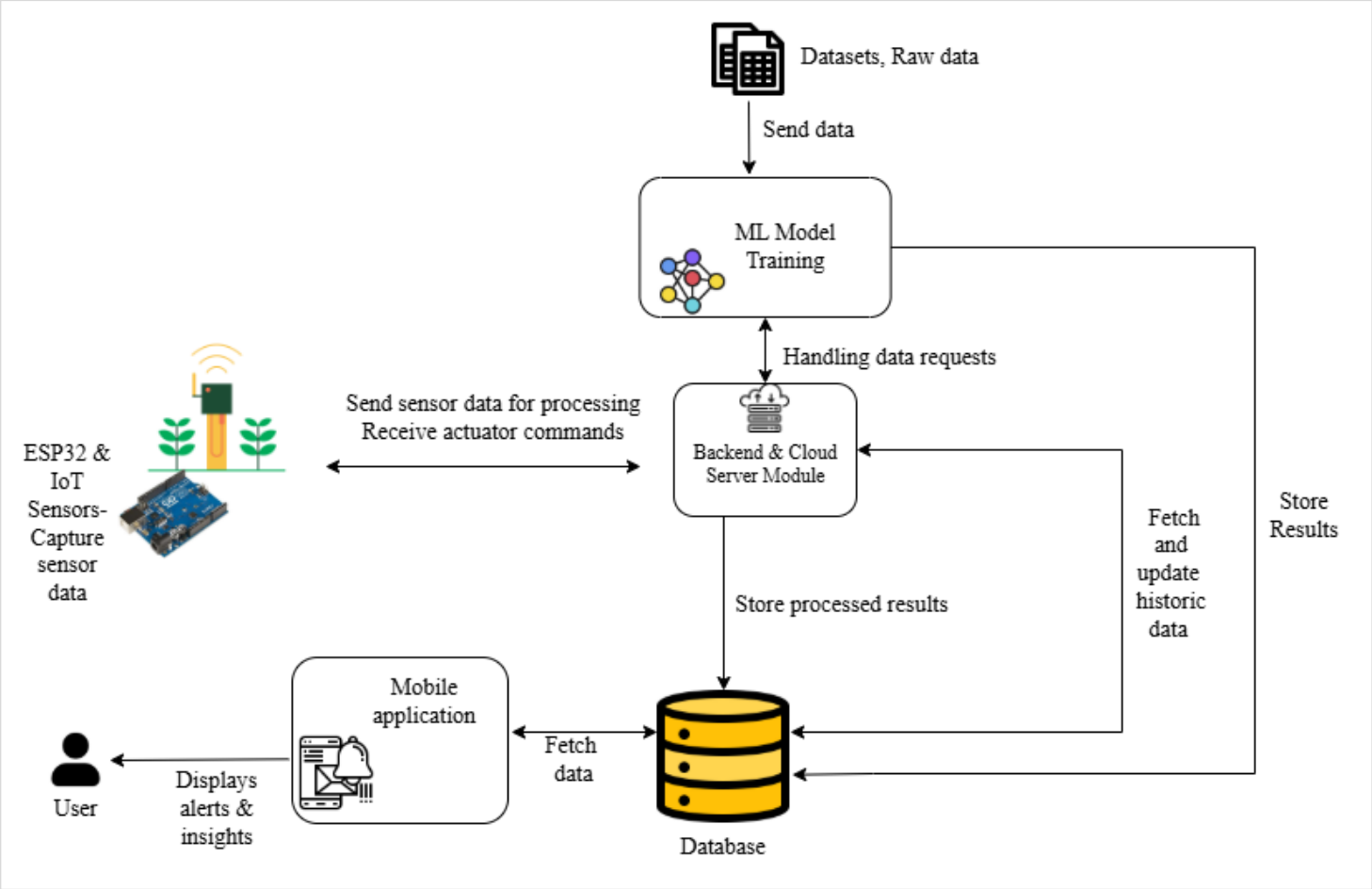


Figure 2.1 Conceptual Framework

Chapter 3: Methodology

3.1 Introduction

This chapter outlines the methodology used in the development of the smart hydroponics monitoring system. It describes the software development approach, the research paradigm, and the various steps taken to collect, process, and analyze the data. The project follows the Agile prototyping methodology, which supports iterative development and continuous testing. Within this framework, Object-Oriented Analysis and Design (OOAD) is applied to structure and model the system's components. Some of the tools and technologies used include Python, Random Forest, XGBoost, Firebase, Flutter, and IoT hardware components.

3.2 Research Paradigm

This project follows an experimental research paradigm, which is suitable for solutions involving machine learning and IoT integration. The focus is on building a working prototype that can detect anomalies in hydroponic farming conditions using live data collected from sensors. Experiments are conducted to test the accuracy and reliability of the model's predictions when integrated with a mobile-based monitoring application.

The experimental setup involves the simulation of a hydroponic system using a publicly available dataset representing real-world parameters. These include temperature, humidity, pH, and EC, which are critical for growing leafy vegetables in soilless environments(Mamatha et al., 2023). The mobile app receives the results from the trained ensemble model, providing feedback and alerts to the user when an anomaly is detected.

3.2.1 Data Acquisition

Data acquisition refers to the process of collecting raw data from various sources to analyze and train intelligent systems. In IoT and machine learning applications, this involves capturing structured or unstructured data using sensors or other data sources and storing it for further processing(Syafrudin et al., 2018).

For this project, data is obtained from a publicly available dataset on Kaggle the "Hydroponic IoT Sensor and Actuator Logs (Raw and Processed)" which contains raw real-time telemetry records for hydroponic systems. The dataset includes environmental

parameters like temperature, pH level, TDS (Total Dissolved Solids), humidity and different actuators for adding water and nutrients, all of which are critical in hydroponic farming. These parameters were recorded through IoT sensors in a controlled environment simulating a real-world hydroponic farm. The dataset supported the training of the ensemble ML model which was used to detect anomalies in the system and issue early alerts to users.

3.2.2 Data Preprocessing

Data preprocessing is the technique used to clean, organize, and transform raw data into a usable format before feeding it into a machine learning model.(Rosero-Montalvo et al., 2022). This step is particularly essential in IoT-based systems where sensor noise and inconsistencies are common.

For this project, preprocessing began with inspecting the dataset for missing entries and outliers. Missing values were handled using imputation techniques or removed if necessary. Numerical features like temperature and pH were normalized to ensure consistent scale, while categorical entries were encoded into binary or one-hot format as needed. Any text-based labels were cleaned and standardized. This cleaned data was then split into training and testing sets, with 75% used for training and 25% reserved for testing to enable the ensemble model to learn from a diverse set of inputs and generalize effectively to unseen cases.

3.2.3 Model Training

The model used in this project utilized an ensemble of XGBoost and Random Forest, which are powerful machine learning algorithms known for their efficiency and ability to handle complex data. It is well-suited for applications that involve sensor data and anomaly detection due to its ability to manage non-linear relationships and reduce overfitting through regularization (Le et al., 2022).

For this study, Random Forest is trained using 75% of the dataset obtained from the hydroponics system. The training process involves stratified k-fold cross-validation to ensure that the model is evaluated fairly across different data segments. During training, hyperparameters such as learning rate, maximum depth, and the number of estimators is fine-tuned using grid search to improve the model’s performance. Feature analysis was

also conducted to identify which environmental factors most significantly influence crop conditions. The trained model learnt patterns that differentiate between healthy and potentially problematic conditions in the hydroponic system.

3.2.4 Model Validation and Testing

Model validation is performed using the remaining 25% of the dataset, which was not seen by the model during training. Evaluation metrics include Precision, Recall, F1-score, and Accuracy with F1-score being especially important to balance between false positives and false negatives in anomaly detection.

Once validated, the model is utilized in a controlled environment where sensor inputs are passed through the IoT prototype which was implemented using ESP32 Microcontroller and model predictions are returned in real time. These predictions are stored in Firebase, and the mobile application displays them to the user through alerts. These alerts are sent if values deviate from optimal ranges. This integrated setup helps test the end-to-end flow of data and verifies the practical utility of the system (Mosaad et al., 2023).

3.3 System Development Methodology

This project adopts prototyping methodology within agile development as the guiding software development approach. This methodology supports rapid development and continuous improvement, which is especially important in projects involving hardware, specifically IoT in this case, machine learning models, and mobile applications. The prototype was developed in stages, starting with a basic functional version and then refined based on testing and feedback. The use of OOAD ensured that each system component is modular, scalable, and easy to maintain. This flexible and iterative structure aligns well with the project's goals of building a smart hydroponics system that is responsive, adaptable, and user-friendly.

3.3.1 Justification of Methodology

The decision to use agile prototyping for this project is based on its strong ability to support iterative learning, user involvement, and system evolution, all of which are critical in building a smart hydroponics monitoring system. Prototyping enables the developer to create a minimal working version of the system early, test its functionality, and refine it based on real-world conditions and feedback. This is particularly effective when dealing

with real-time sensor data, mobile integration, and machine learning, where flexibility and frequent testing are required (Gupta et al., 2022).

Moreover, agile methodologies like prototyping promote continuous improvement and adaptability, which is vital when building systems that involve both hardware and software components. Research has shown that agile based prototyping improves user satisfaction and system success in IoT applications by allowing developers to respond quickly to changes and reduce risks during development (Kinast et al., 2021). Combined with the OOAD (Object Oriented Analysis Design) paradigm, this approach ensures a structured yet flexible design process where system features are developed as discrete objects and improved continuously throughout the development lifecycle (Sommerville, 2016).

3.4 Methodology Diagram

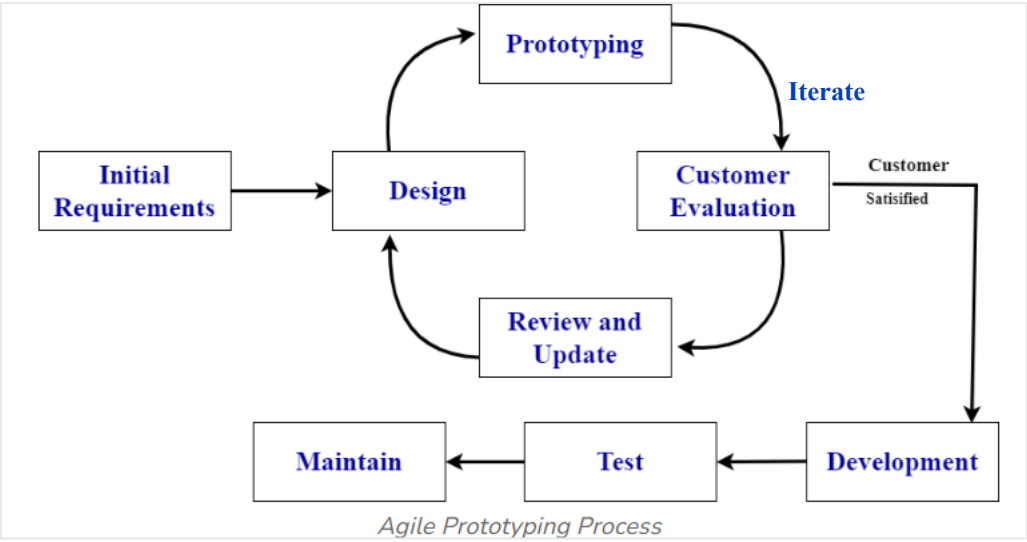


Figure 3.1 Agile prototyping Methodology

3.4.1 Prototype Planning

This is the first stage in Agile Prototyping; it involves identifying user needs and defining what the initial system demonstrated. In this project, user requirements were derived from secondary sources, such as existing smart farming systems, academic literature on hydroponics, and public datasets. The core focus is on key parameters that affect lettuce growth namely temperature, humidity, pH, and EC. These variables are essential in

determining optimal plant health in hydroponic setups and must be monitored continuously.

The prototype was intended to demonstrate real-time monitoring of hydroponic conditions using IoT sensors connected to an ESP32 microcontroller, the application of an ensemble machine learning model for anomaly detection, and a mobile interface for user feedback. An ESP32 acts as the central processing unit, collecting data from sensors and transmitting it for analysis. This demonstration reflects the overall goal of building a practical, scalable solution for small-scale farmers who lack advanced monitoring tools. By beginning with clear definitions of scope and functionality, the planning phase ensures that the prototype aligns with both project goals and stakeholder expectations.

3.4.2 Initial Prototype Development

The second phase involves the development of the initial prototype, referred to as a Minimum Viable Product (MVP). This prototype integrates three components: the IoT sensor system, a backend with machine learning logic, which used an ensemble of XGBoost and Random Forest algorithms, and a mobile application. The system followed an iterative and incremental development approach, where small, functional parts are developed, tested, and improved over multiple cycles.

This phase was guided by the Object-Oriented Analysis and Design (OOAD) methodology, which facilitates modularity and clear separation of concerns. Through this approach, various diagrams such as Use Case Diagrams, Sequence Diagrams, and Class Diagrams were developed to map system functionality, sensor interactions, and data flow. The development tools included Python for sensor integration and machine learning model training, Flutter for mobile application development, Firebase for real-time data storage and synchronization, Random Forest and XGBoost for anomaly detection.

3.4.3 Iterative Refinement

The stage represents the core of Agile Prototyping. After building the MVP, it was tested and refined across multiple sprint cycle. At the end of each sprint, the prototype was reviewed against specific goals, such as achieving at least 90% prediction accuracy from the ensemble model and ensuring reliable sensor communication.

Feedback was gathered through testing different simulated farming scenarios, after which the prototype was adjusted to address identified issues. For instance, if the sensors return inaccurate readings, calibration algorithms or hardware adjustments were implemented. Similarly, improvements to the machine learning model's capability for detecting anomalies involved hyperparameter tuning and retraining on the cleaned dataset. Throughout these cycles, updated OOAD diagrams were produced to reflect changes in the system's structure and interaction logic. This continuous improvement approach is what makes Agile Prototyping especially effective for hardware-integrated, user-facing systems.

3.4.4 Testing

Once the prototype reaches a reliable state, testing was conducted to ensure functionality, performance, and usability. The project used several levels of testing. Unit testing was used to verify individual components like sensor modules, database connectors, and the ML model's ability to detect anomalies.

Integration testing ensured that these components work together, for example, whether the data from the pH sensor correctly flows into the ML model and the result updates in the mobile interface.

System testing evaluated the prototype as a whole, assessing it under real and simulated conditions to ensure it can reliably monitor parameters and issue alerts. Additionally, performance testing focused on the accuracy of anomalous detections, latency in sensor data transmission, and mobile app response times.

3.4.5 Final Prototype Evaluation

In the final stage, the prototype was evaluated to determine whether it met the system's goals and was ready for deployment or further development. The evaluation involved both system analysis and testing against predefined success criteria such as model detection accuracy, sensor stability, user satisfaction, and system responsiveness. System analysis focused on examining the interaction between hardware, software, and users to ensure that all components functioned cohesively and supported the intended objectives.

If the system consistently detected anomalies, provided timely alerts, and demonstrated good usability, it was considered successful. The results of this evaluation and analysis

informed whether the prototype should be scaled into a full system. This includes expanding the mobile application's features, integrating more crop types, or deploying them in larger hydroponic setups. Feedback from all testing stages was documented and used to create recommendations for future iterations. The final evaluation ensures that the prototype is not just functional but also relevant, usable, and scalable in real-world hydroponic farming contexts.

3.5 System Design Diagrams

For this smart hydroponic crop monitoring system, various diagrams were used to provide a visual representation of interactions, components, and data flows. These include the Use Case Diagram for modeling system-user interactions, Class Diagram for defining system structure, Entity Relationship Diagram (ERD) and Database Schema for data organization, as well as System Architecture, Sequence, Activity Diagrams and Wireframe to support both logic planning and UI development.

3.5.1 Use Case Diagram

A use case diagram is a behavioral model that shows how different actors interact with the system and outlines its functional boundaries (Brahmia et al., 2024). In this project, it included actors such as users (farmers) and system administrators capturing activities like monitoring crops, managing users, and exchanging sensor data. This diagram is key for identifying user needs and ensuring the system aligns with real-world interaction scenarios.

3.5.2 Class Diagram

Class diagrams represent the static architecture of a system by defining classes, their attributes, methods, and relationships (Bashir et al., 2021). The diagram captured components such as sensor module, user management module, farm data processing each responsible for specific tasks in data handling and decision-making. This approach supports modularity and allows for easier updates and code reuse in future development stages (Bashir et al., 2021).

3.5.3 Entity Relationship Diagram (ERD)

An ERD models data relationships within a relational database by showing entities, attributes, and their associations (Salem et al., 2024). For this system, it included entities like admin, users, farms and sensor data, showing how data flows and is linked in the

database. The ERD supports the planning of a normalized, scalable database essential for Firebase integration and reliable data retrieval.

3.5.4 Database Schema

A database schema defines how data is structured at the implementation level, including tables, fields, keys, and constraints (Brahmia et al., 2024). It stored critical information such as user profiles, sensor logs and farm management. This structure ensures efficient data flow between IoT sensors, the ML model, and the mobile application, supporting both system speed and data consistency.

3.5.5 System Architecture Diagram

A system architecture diagram provides a high-level overview of the system's main components and their communication flow (Muccini, 2021). It mapped out the integration between IoT sensors, an ESP32 microcontroller, ml model for anomaly detection, Firebase for cloud storage, and the Flutter mobile application. This diagram is essential for understanding hardware-software interaction and aligning component design with performance requirements.

3.5.6 Sequence Diagram

A sequence diagram illustrates the flow of messages between components in a specific interaction sequence(Jha et al., 2023). In this project, it showed how the environmental conditions detected by sensors, are sent to the ml model for processing and depending on the processed results the user gets an alert if an anomaly is present or viewed normal farm status if conditions are optimal. This helps validate the system's logic over time and ensures real-time processes occur in the correct order (Jha et al., 2023).

3.5.7 Activity Diagram

Activity diagrams model workflows by showing the sequence of actions, conditions, and decisions involved in a process (Jha et al., 2023). This diagram mapped the steps from sensor input to ML based decision-making and alert notification. It clarifies how the system processes data and supports logical flow in task execution, helping to anticipate delays. (Jha et al., 2023).

3.5.8 Wireframes

Wireframes are simplified screen layouts that show the structure of a user interface without detailed design elements (Staiano, 2022). In this system, wireframes were created using Figma. They comprised of screens for the dashboard, alerts, and sensor readings, guiding the Flutter-based application development. They play a crucial role in aligning the interface with user needs and ensuring ease of use for all users.

3.6 Tools and techniques

This section outlines the tools and techniques selected for the development of the smart hydroponic monitoring system. Each tool has been chosen based on its functionality and ease of integration with IoT and ML based system.

3.6.1 Flutter and Dart

Flutter, with Dart as its programming language, was used to develop mobile applications for real-time monitoring. It supports cross-platform development and is well-suited for creating responsive and interactive UIs. (Flutter Documentation, 2025).

3.6.2 Visual Studio Code

Visual Studio Code was used as the main development environment for writing and organizing the project's codebase. It supported the development of the backend logic, integration scripts, and mobile-related configurations, making it suitable for managing the overall project structure (Pattanayak, 2019).

3.6.3 Google Colab

Google Colab was used for training and evaluating the ensemble machine learning model built with XGBoost and Random Forest. Its cloud-based computational resources allowed efficient processing of the dataset, model experimentation, and performance testing without the limitations of local hardware.

3.6.4 Firebase

This was used as the backend platform for real-time data storage, synchronization, and communication between the IoT setup and the mobile application. It supported seamless integration with Flutter, delivered real-time messaging for alerts and notifications, and

hosted the trained ensemble model through firebase functions, and sent timely feedback (GnagnarellaDirector et al., 2025).

3.6.5 ESP32 Microcontroller

This functioned as the IoT gateway, collecting sensor data and relaying it to the cloud. Its GPIO support and Python compatibility make it suitable for small-scale sensor-based applications. As a compact and affordable computing device, it also supports basic programming tasks in Python, making it ideal for embedded ML projects.(Haji et al., 2021).

3.6.6 IoT Sensors and actuators

Sensors such as the DHT22 for temperature and humidity, the TDS sensor for nutrient concentration, the pH sensor for acidity and alkalinity, and an ultrasonic sensor for water level were used alongside actuators including a relay and a mini submersible pump. These components worked together to monitor key environmental conditions within the hydroponic system. Real-time sensor data is crucial for supporting automated feedback loops and maintaining optimal growth conditions. The sensors integrate seamlessly with the ESP32 microcontroller, allowing efficient data collection and real-time processing through Python (Haji et al., 2021).

3.6.7 Git and GitHub

Git was used for version control while GitHub hosted the code repository. This enables better collaboration, change tracking, and secure cloud storage (GitHub Documentation, 2025).

3.6.8 Ensemble Algorithms: XGBoost and Random Forest

The ensemble model, built using both XGBoost and Random Forest, was applied to detect anomalies in hydroponic growing conditions based on sensor data. These algorithms are known for their high accuracy, robustness, and effectiveness in agricultural monitoring tasks, especially when handling complex environmental datasets (Sangeetha et al., 2025).

3.6.9 Real-Time Data Synchronization

The system relied on Firebase's real-time capabilities to ensure instant updates between sensors and the user interface. This enhances responsiveness and improves the reliability of alerts (GnagnarellaDirector et al., 2025).

3.6.10 Figma

Figma was used to design wireframes for the Flutter-based mobile application. It allows for cloud-based UI design, making it ideal for planning the app's user interface before development (Staiano, 2022).

3.7 System Deliverables

This section outlines all the components and outputs expected from the development of the smart hydroponic monitoring system.

3.7.1 Project Proposal

This is the formal document outlining the project's background, objectives, scope, and methodology. It provides direction and ensures all stakeholders understand the project's goals and deliverables.

3.7.2 Test Cases

Test cases were created to verify that each module of the system performs as expected. They are essential for validating system reliability and ensuring functional accuracy.

3.7.3 IoT Prototype

This was a small-scale demonstration of the IoT setup using sensors, actuators and an ESP32 microcontroller. It proves the feasibility of the system design and supports testing in a controlled environment.

3.7.4 Mobile Application

The mobile application allowed users to monitor crop conditions, view alerts, and manage settings. It enables real-time interaction and offers a user-friendly interface for visualization.

Authentication Module

This module managed secure user login and access control. This ensures only authorized users can access or modify system data, enhancing overall security.

Alerts and Anomaly Detection Module

This module delivered alerts generated from the ensemble machine learning model, which used XGBoost and Random Forest to detect abnormal environmental conditions. Notifications were sent through Firebase, enabling users to respond quickly to potential risks.

Data Management and Synchronization Module

The application integrated with Firebase for data storage, real-time updates, and seamless synchronization with the IoT setup. This ensured users always received up-to-date sensor readings and system status, supporting timely decision-making and remote management such as controlling the water pump when the need arose.

3.7.5 System Documentation

Comprehensive documentation included technical specifications, user manuals and system architecture diagrams. This supports system maintenance, scalability, and smooth knowledge transfer for future developers or users.

Chapter 4: System Analysis and Design

4.1 Introduction

This chapter presents the system analysis and design of the smart hydroponic monitoring system. It outlines the functional and non-functional requirements of the system and details how different modules were structured to achieve the objectives of the project. The chapter also highlights the design methodology applied, Object-Oriented Analysis and Design (OOAD), which emphasizes modularity and clear separation of concerns. Various diagrams such as a Use Case Diagram, Sequence Diagram, Entity-Relationship Diagram (ERD), Activity Diagram and System Architecture diagram are used to represent the system's behavior, structure, and interactions.

4.2 System Requirements

The system requirements define the configuration and specifications necessary to ensure smooth and efficient functioning. These requirements were drawn from the objectives and scope of the project and are divided into functional and non-functional requirements.

4.2.1 Functional Requirements

The system was structured into distinct processes to ensure modular development and maintainability:

i. **Data Collection** -This process involved the continuous collection of environmental parameters from the hydroponic setup. The ESP32 microcontroller retrieved readings from the sensors measuring temperature, humidity, pH, and Total Dissolved Solids (TDS). Each data point was time-stamped and validated before being transmitted to ensure accuracy and reliability.

ii. **Data Transmission and Storage** - After sensor data collection, the sensor readings are transmitted wirelessly to the cloud through Firebase. Firebase's real-time database ensures synchronization between the IoT testbed and the mobile application, allowing continuous monitoring and data persistence even after temporary network interruptions.

iii. **Anomaly Detection** - This process applied the trained ensemble model that used XGBoost and Random Forest algorithms, to incoming sensor data to identify anomalous conditions. Data was preprocessed cleaned, scaled, and structured

before model inference. The system classifies each reading as “normal” or “anomalous” supporting timely responses to irregular environmental changes.

iv. **Notification and Alert** - When an anomaly is detected, the system automatically generates alerts and notifies the user via Firebase Cloud Messaging. These alerts were designed to appear in real-time on the mobile application, prompting corrective action to restore optimal growing conditions.

4.2.2 Non-Functional Requirements

i. **System Performance**- The system is optimized for low-latency operation by maintaining efficient communication between the ESP32, Firebase, and the mobile application. Sensor readings are updated in near real-time, with end-to-end delays maintained under five seconds. This was achieved by pushing sensor values to Firebase immediately using asynchronous writes, while the mobile application uses real-time listeners that fetch updates constantly. The machine learning model runs on an optimized Python backend where preprocessing and classification are executed in under two seconds by caching loaded models and minimizing I/O operations.

ii. **System Reliability**- This is ensured through automated reconnection protocols and periodic data validation. When connectivity issues occur, the ESP32 attempts to reconnect within 30 seconds, ensuring continuous data flow. Firebase’s redundancy and synchronization features prevent data loss, maintaining system consistency during recovery.

iii. **System Security**- This is implemented through Firebase Authentication and HTTPS encryption. Only authorized users can access the application, all data transfers between the IoT devices, backend, and mobile application occur over encrypted channels. Access control ensures that only verified users can retrieve or modify system data.

iv. **System Scalability** - The architecture supports horizontal scaling by allowing new sensors and hydroponic units to be added without altering the system design. Each farm or device writes to a unique Firebase node, enabling independent data streams that do not affect each other’s performance. Firebase automatically scales its

storage and throughput to handle additional data, while the backend uses modular APIs that treat every new device as an independent client. This approach allows the platform to accommodate hundreds of farms or sensor nodes with minimal configuration changes.

v. **System Usability** - The mobile application emphasizes usability through intuitive interfaces, consistent design, and clear navigation. Contextual help features and a logical information hierarchy reduce training needs, ensuring ease of adoption by end users.

4.3 System Narrative

The objective of this project was to develop a smart hydroponic monitoring system focused on lettuce production. Manual monitoring of pH, nutrient concentration, and environmental conditions is often time-consuming and prone to error, creating risks for crop health.

The system uses IoT sensors connected to an ESP32 microcontroller to automatically collect real-time data on key parameters. This data is stored in Firebase and analyzed by an ensemble model trained with XGBoost and Random Forest algorithms, to detect anomalies that may indicate unfavorable conditions.

A Flutter-based mobile application provides an interface for farmers, enabling them to view live readings, access historical data, receive alerts, and control the water pump remotely. By integrating IoT, machine learning, and mobile technologies, the system delivers timely insights that help prevent crop failure and improve efficiency in farming.

4.4 System Analysis Diagrams

4.4.1 Use case Diagram

The use case diagram presents the system's key interactions between two primary actors, admin and user. Both actors have access to login and registration functionalities. The User can register farm details, monitor sensor data, receive alerts, evaluate farm conditions, and manage farm activities. The admin oversees system-wide operations, including managing users, viewing all farm data, configuring telemetry thresholds, and managing farm devices. This diagram provides a structured overview of the functional requirements and roles within the smart hydroponic system.

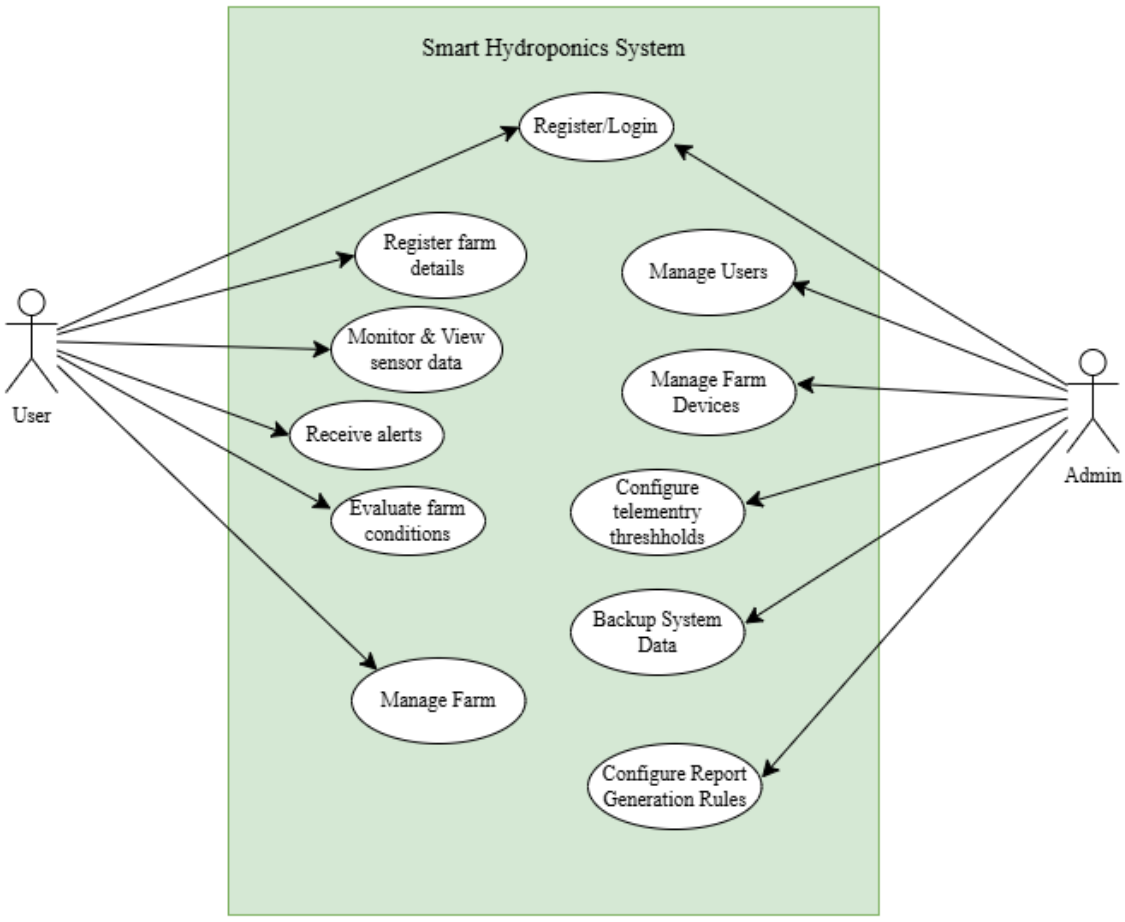


Figure 4.1 Use case diagram

4.4.2 Class Diagram

The class diagram highlights the relationship between the system's main entities. The admin class maintains a one-to-many relationship with the User class, reflecting administrative control over multiple users. A User is linked to one or more farms, while each farm is associated with one or more devices. Devices are further connected to one or more sensors. Reports are then generated for analysis using the data from the farms. Attributes such as integer, string, and relevant methods are included to define properties and operations for each class. This diagram demonstrates the modular structure of the system, clarifying both entity interactions and object-oriented design principles.

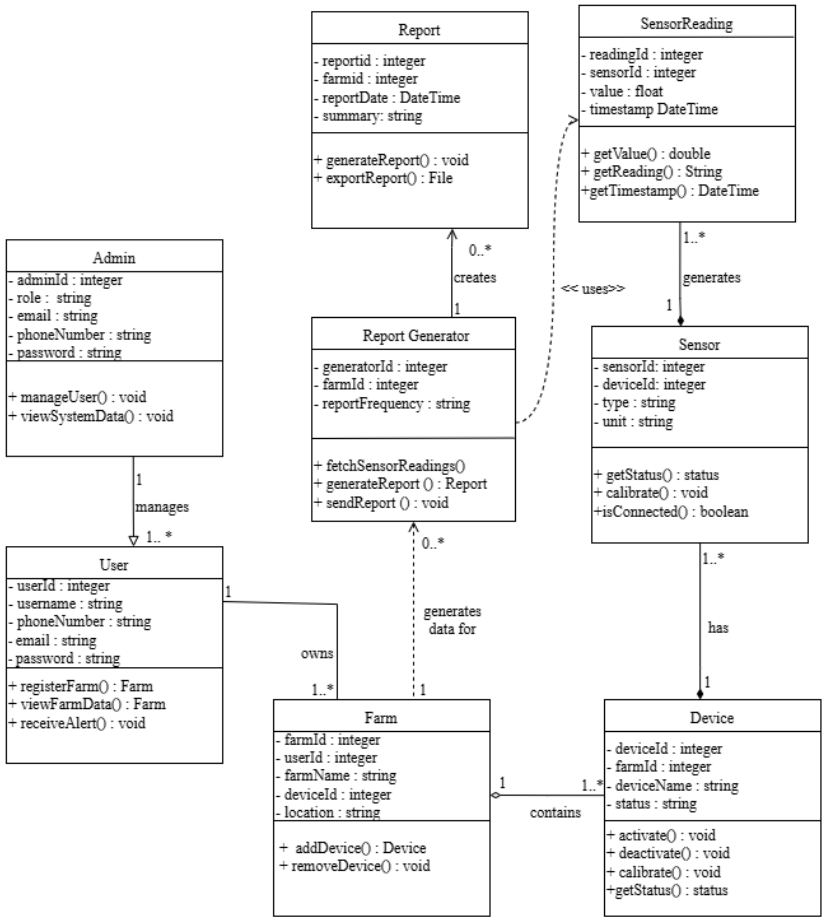


Figure 4.2 Class Diagram

4.4.3 Entity Relationship diagram

The Entity Relationship Diagram (ERD) presents the logical structure of the smart hydroponic system by illustrating how major entities are connected. The diagram shows how administrators are responsible for overseeing user accounts, while users are associated with specific farms. Each farm is linked to devices, and every device contains multiple sensors. Reports are then generated from the farm data. These relationships outlined the ownership chain from administrator to sensor level, providing a clear foundation for tracking data flow and management responsibilities. By defining these links, the ERD supported the development of a well-structured and consistent data model.

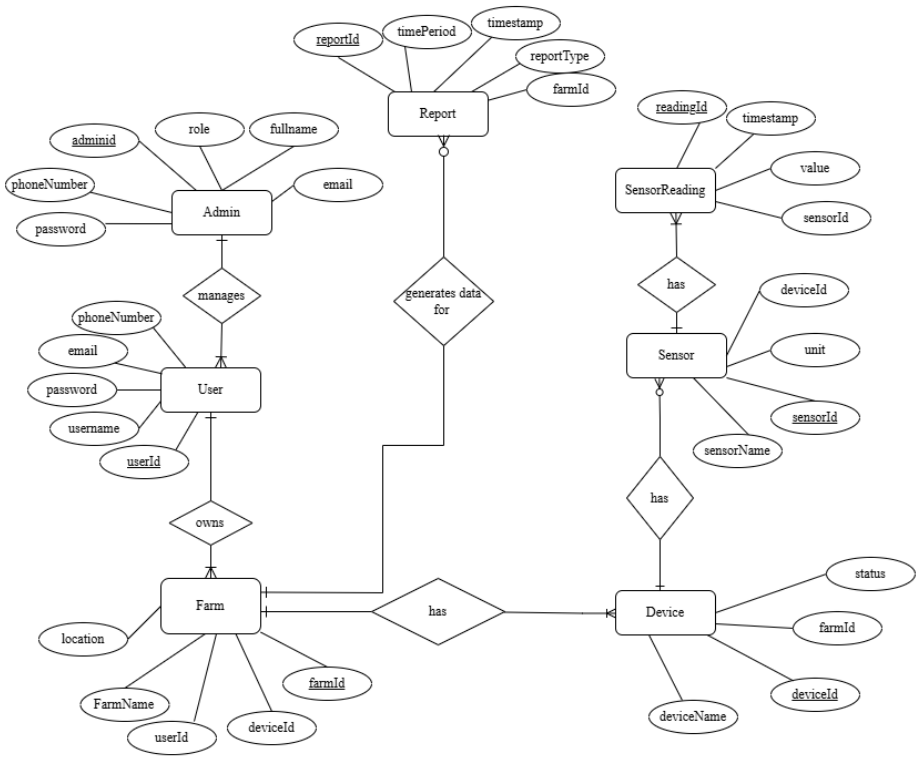


Figure 4.3 Entity Relationship Diagram

4.4.4 Sequence diagram

The sequence diagram depicts the flow of information between system components during operation. After logging in, the user initiates a farm status request, which triggers data collection from the sensors. This data is processed by the ensemble ML model, to detect anomalies. The results are then concurrently stored in the database and transmitted to the mobile application, where the user receives either normal farm status or alert notifications. This diagram illustrates the dynamic, real-time interactions between system components.

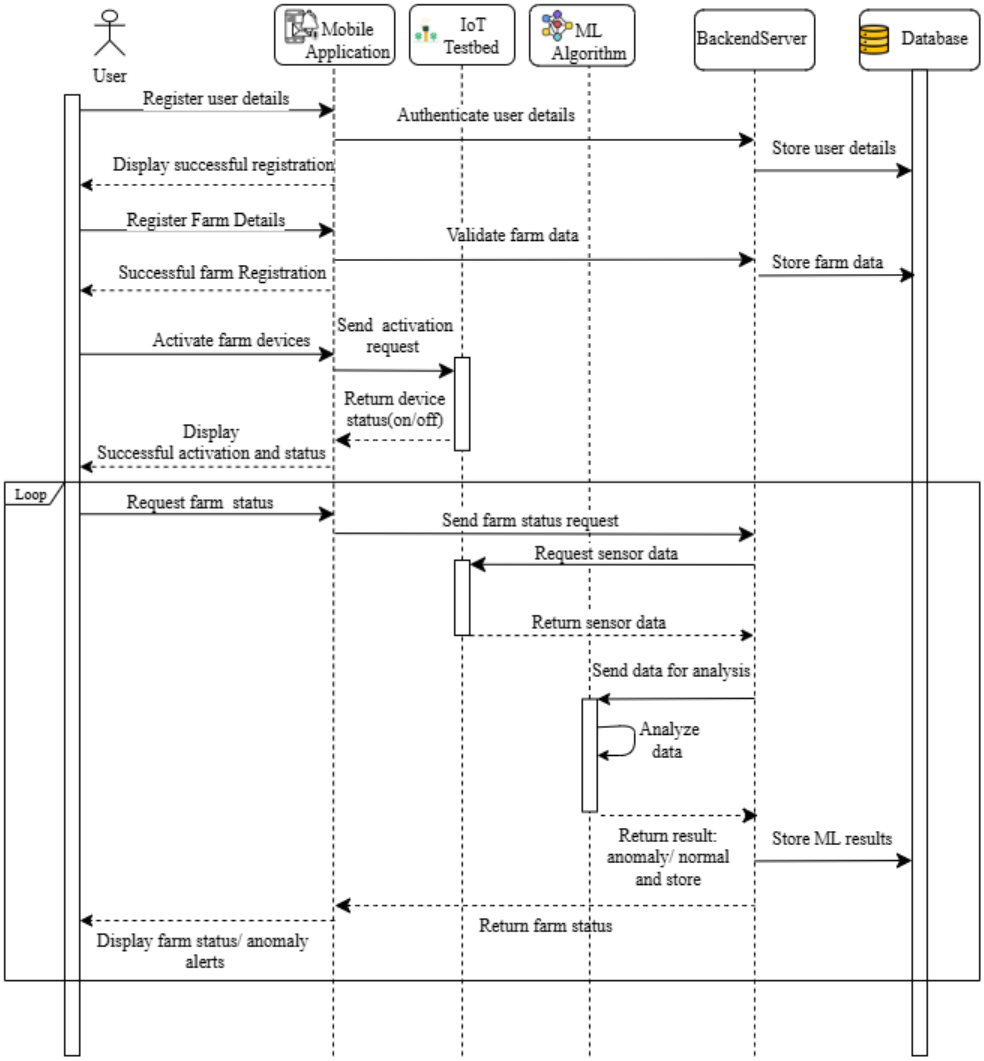


Figure 4.4 Sequence diagram

4.4.5 System Architecture

The system architecture is organized into four tiers. The first tier is the application layer, which provides a mobile interface where users monitor farm conditions and receive insights. The second tier, the data processing layer, handles machine learning analysis and decision-making based on sensor inputs. The third tier is the sensing layer, consisting of IoT devices such as the ESP32 microcontroller, pH sensor, TDS sensor and DHT22(temperature and humidity) sensor. Finally, we have the storage layer, where collected data is securely stored for retrieval and historical analysis. Data flows continuously across all tiers, with two-way communication supported by cloud services. This structure ensures real-time monitoring, analysis, and control of hydroponic operations in a scalable manner.

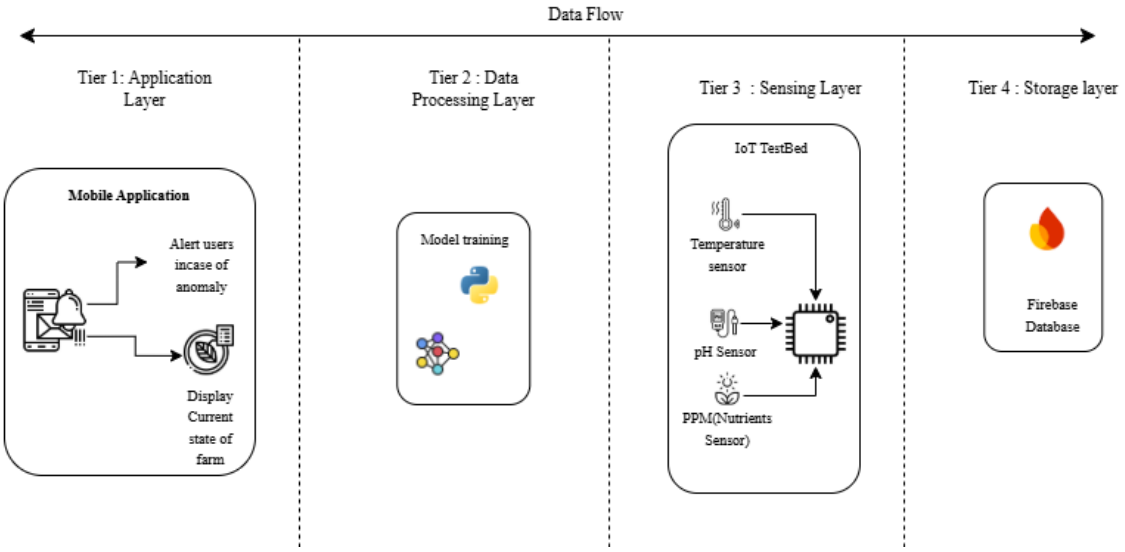


Figure 4.5 System Architecture Diagram

4.4.6 Database Schema

The database schema presents the blueprint for how data is organized and stored within the system. It shows tables for entities such as users, farms, devices, sensors and reports each with fields that capture identifiers, readings, and timestamps. The relationships between the tables ensured consistency, for instance by linking farms to their devices and devices to their associated sensors. This structure outlined how data integrity was maintained and supported efficient retrieval, storage, and long-term management of operational information.

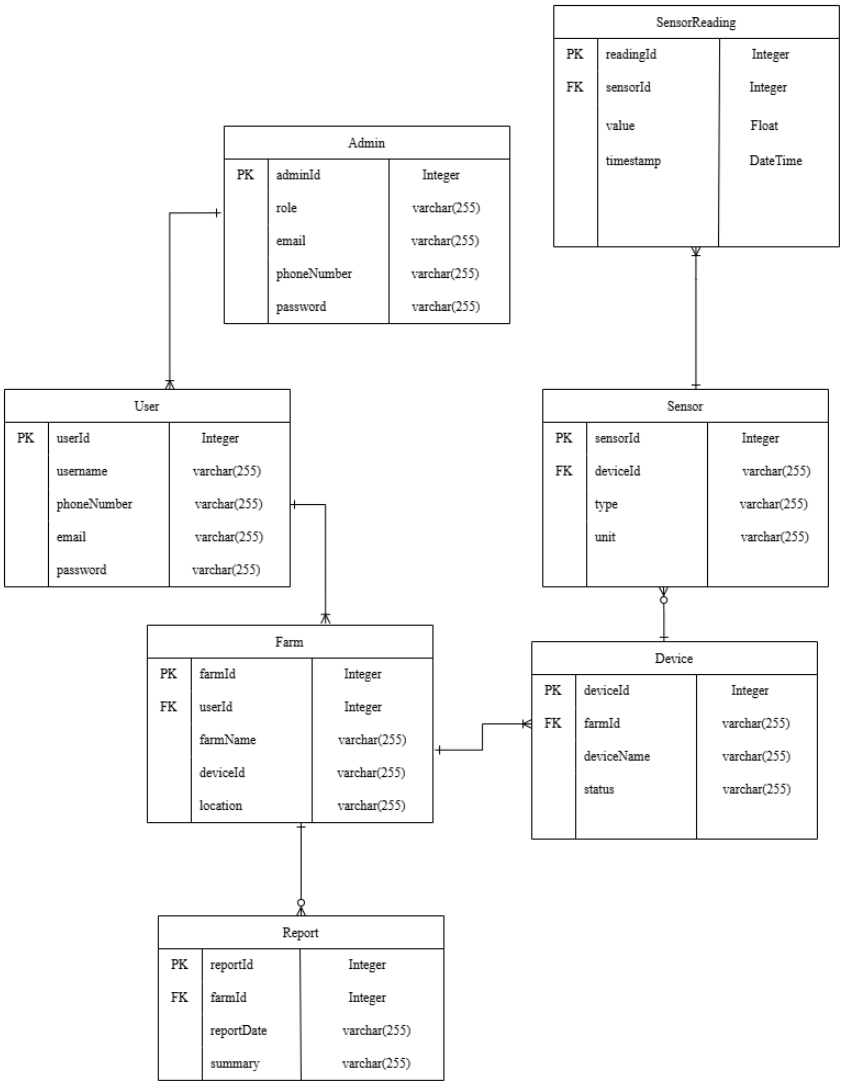


Figure 4.6 Database Schema

4.4.7 Activity Diagram

The activity diagram models the workflow of farm monitoring and management. The process begins when the user opens the mobile app and activates the sensors. Sensor data is collected and sent to the ensemble ML model for analysis. If the environment is within optimal conditions, the system sends a normal status update to the user; otherwise, alerts or recommendations are generated. The user may also issue control commands, such as starting or stopping the water pump. At the end of each cycle, all data is stored for future reference. This diagram captures the continuous feedback loop between the user, IoT testbed, and the machine learning model.

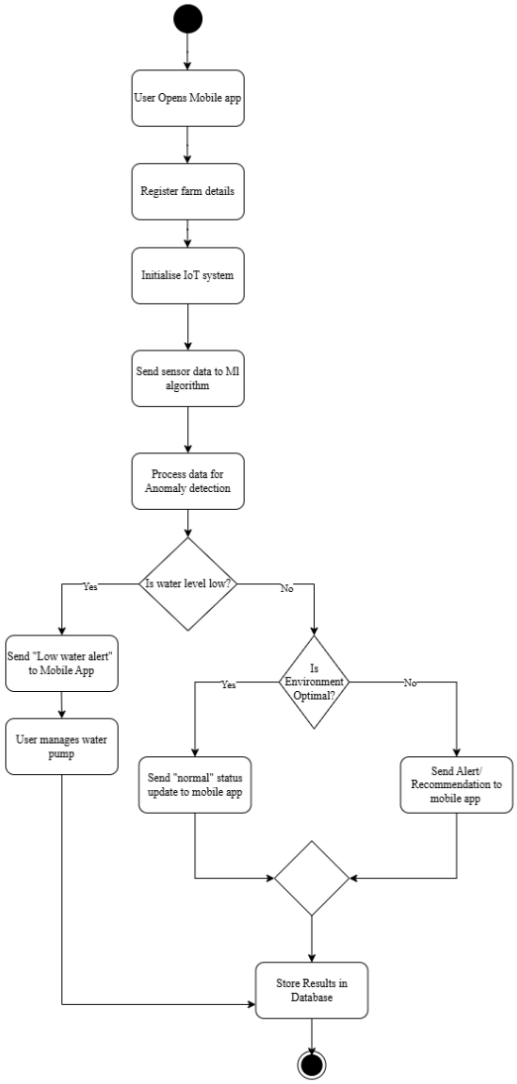


Figure 4.7 Activity Diagram

4.4.8 IoT Block Diagram

The IoT block diagram illustrates the physical components of the hydroponic monitoring system. The ESP32 microcontroller acts as the central hub, connected to a pH sensor, TDS sensor, DHT22 (temperature and humidity), and a relay-controlled mini submersible water pump. These components communicate wirelessly with the cloud server, where machine learning processing takes place. The processed data is then relayed to the mobile application, enabling users to monitor farm status and receive alerts in real time.

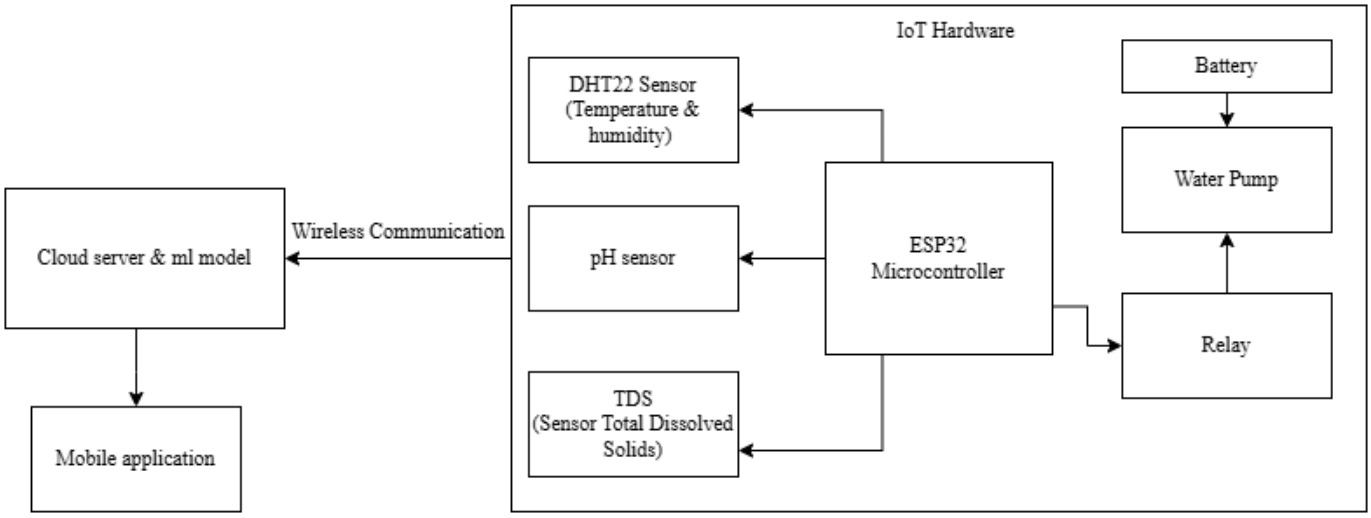


Figure 4.8 IoT Block Diagram

4.4.9 Wireframes

The wireframes provide a visual blueprint of the mobile application interface. Key screens include the Login and Signup page for user authentication, the Home page for quick access to farm data, and the Analytics page for detailed insights into sensor readings and farm conditions. These wireframes emphasize intuitive navigation and user-friendly interaction, ensuring accessibility for both novice and experienced users.

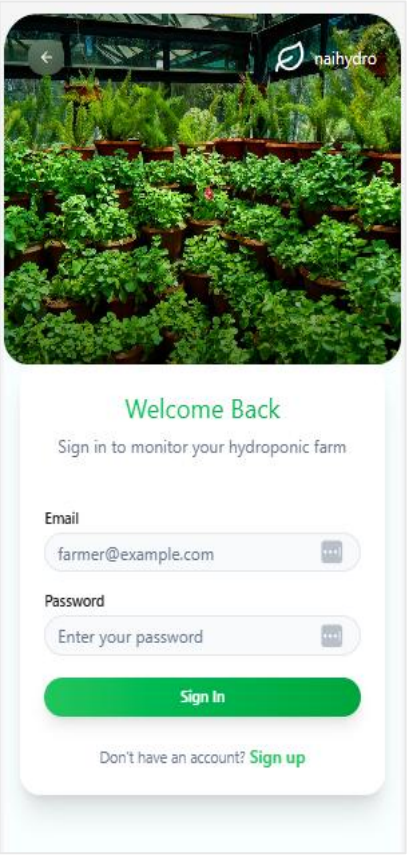


Figure 4.9 Sign Up Page

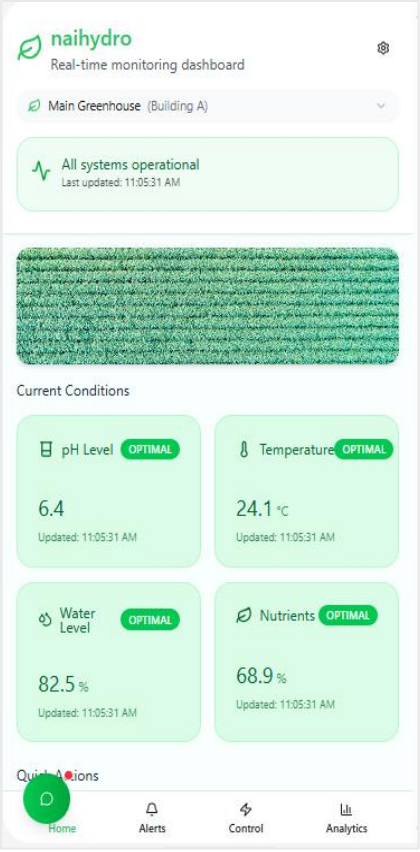


Figure 4.10: Home Page

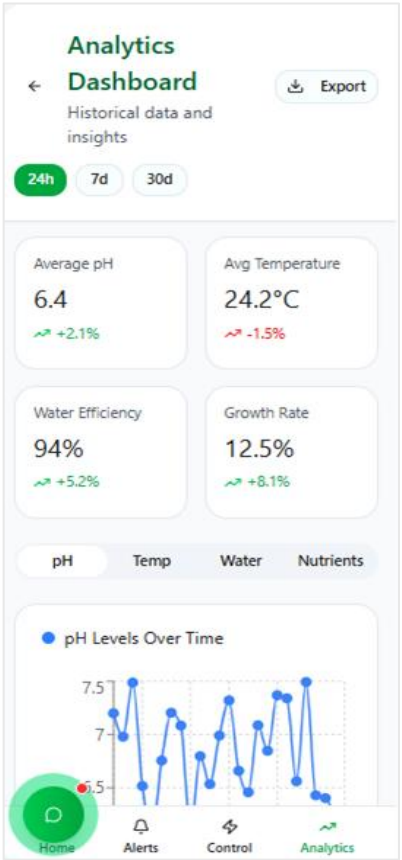


Figure 4.11: Analytics Dashoard Page

Chapter 5: System Implementation and Testing

5.1 Introduction

This chapter presents the implementation and evaluation of the Smart Hydroponic Monitoring System. It explains the hardware and software environments used, the dataset applied in model training and outlines how the machine learning model was developed and evaluated. The section also details the major components of the system and how each module was implemented. Finally, the testing process and outcomes are discussed to validate system functionality and performance.

5.2 The Implementation Environment

This section describes the computing environment required for developing, deploying, and testing the system. The implementation relied on moderate hardware and compatible software tools to ensure smooth model execution and real-time data handling.

5.2.1 Hardware Specifications

Table 5.1 Hardware Specifications

Hardware	**Justification**
Android 11 and later	Ensures compatibility with modern APIs and provides a stable environment for deploying the mobile application.
Processor, Octa-core 2.0 GHz or higher	Provides sufficient processing capability for real-time data handling, ML model inference, and user interface responsiveness.
Network, Stable 4G/LTE or Wi-Fi (≥100 Mbps)	Enables fast and reliable data transmission between the IoT testbed, Firebase cloud services, and the mobile application.
Storage, Minimum 32 GB internal memory	Accommodates mobile app files, cached sensor data, and machine learning model weights without degrading performance.
Memory (RAM), Minimum 8 GB	Ensures smooth background operations, efficient data processing, and consistent app performance under continuous sensor communication.

Microcontroller (ESP32, Wi-Fi 802.11 b/g/n, Bluetooth)	Provides built-in Wi-Fi and Bluetooth connectivity, and low-power operation, allowing efficient multi-sensor data handling and seamless cloud communication.
pH Sensor (Analog, 0–14 range)	Gives accurate pH readings and works smoothly with the microcontroller's analog input to track nutrient solution balance
DHT22 Temperature and Humidity Sensor	Offers reliable digital readings with low error rates, ensuring accurate monitoring of environmental conditions.
TDS Sensor (Analog, 0–1000 ppm range)	Measures nutrient concentration levels using simple voltage signals that are easy for the controller to interpret.
Ultrasonic Sensor (HC-SR04)	Employs echo measurement for non-contact water level detection, minimizing corrosion risk and providing high-resolution distance data via digital I/O pins.
Water Pump (Mini DC Submersible)	Operates at low DC voltage with efficient nutrient flow and can be easily automated and controlled using the relay module for consistent system operation.
Relay Module (5V)	Acts as a safe electronic switch that lets the controller turn pumps on or off.

5.2.2 Software Specifications

Table 5.2 Software Specifications

Software Name	**Justification**
Python 3.10 - 3.12	Used for data preprocessing, model training, and evaluation of the ensemble model for anomaly detection.
Node.js	Serves as the backend environment, managing API endpoints for data exchange between IoT devices, the ML model, and the mobile app..

Flutter SDK	Enables the development of a cross-platform mobile application that integrates real-time data, alerts, and visualizations for user interaction.
Firebase	Offers real-time database, cloud storage, cloud messaging and user authentication services to synchronize IoT data with the mobile application.
Arduino IDE	Used to program the ESP32 microcontroller, configure sensor connections, and upload firmware for data acquisition.
Google Colab	Provides a cloud-based platform for executing Python scripts, training the model and testing performance without local resource limitations.

5.3 Description of the Dataset

The dataset used in this project was the "Hydroponic IoT Sensor and Actuator Logs" Dataset, which contained over 50,000 records collected from an IoT-based hydroponic system. The dataset included 13 features representing both sensor readings and actuator states, such as pH, Total Dissolved Solids (TDS), humidity, temperature, and water level. These measurements reflected variations in nutrient solution quality and environmental conditions, offering an authentic representation of real-world hydroponic operations.

The dataset contained irregular time intervals and missing values, characteristics common in IoT-based agricultural data. This made it particularly suitable for training anomaly detection models, as it reflected real sensor fluctuations that could indicate potential system faults. The data was cleaned and formatted into structured input for the machine learning model, which then learned to classify normal and abnormal environmental conditions. The dataset played a vital role in simulating realistic sensor behaviors and validating the predictive capabilities of the developed system.

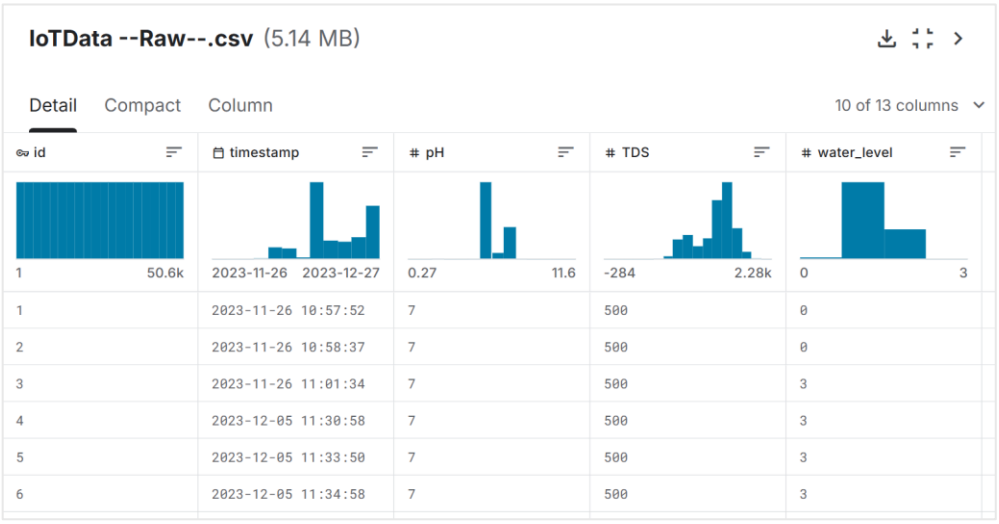


Figure 5.1 Sample of Hydroponic System Dataset

5.3.1 Dataset Normalization and Balancing

The dataset exhibited significant class imbalance, with 50,462 normal observations and only 558 anomalous observations. This imbalance required special handling during model training to prevent bias toward the majority class. To address the severe class imbalance, an under-sampling strategy was implemented. The majority class, normal observations, was randomly sampled to maintain a 5:1 ratio with the minority class which was the anomaly class. This approach preserved all real anomaly patterns while reducing computational overhead. The balanced dataset was split using stratified sampling to maintain class proportions across subsets. A 75-25 split was applied, allocating 75% of samples to the training set and 25% of the samples to the test set. Stratified sampling ensured that both training and test sets contained representative proportions of normal and anomalous cases. Model validation was performed using 5-fold stratified cross-validation during training. This technique divided the training data into five equal subsets, training

the model five times while using a different subset for validation in each iteration. This approach provided robust performance estimates and helped detect overfitting.

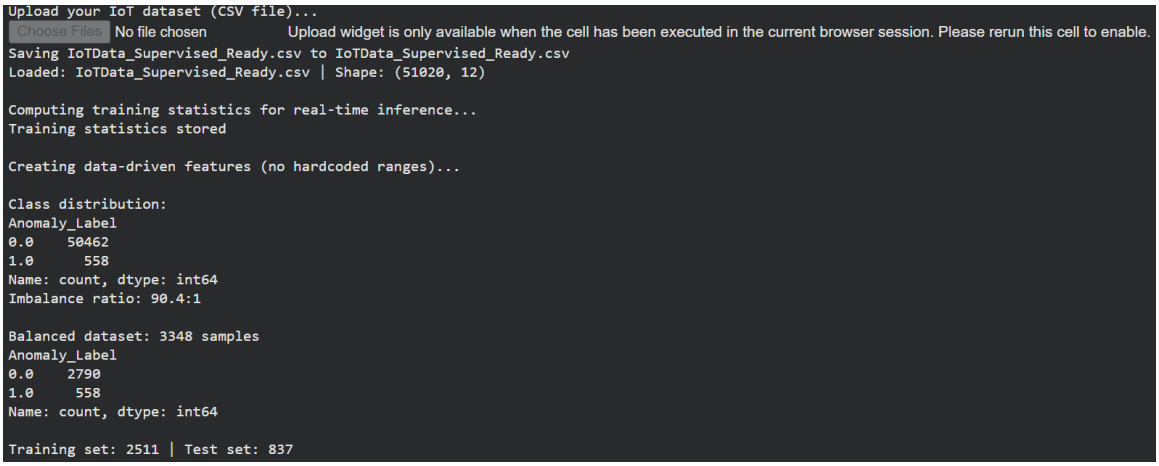


Figure 5.2 Training, Testing and Validating the Dataset

5.4 Model Training and Evaluation

Prior to model training, several data preparation steps were undertaken. Training statistics such as mean, standard deviation, median, and quartiles were computed for each sensor feature and stored for subsequent real-time inference. These statistics enabled the calculation of standardized z-scores during both training and deployment phases.

Advanced feature engineering techniques were applied to enhance model performance. For each base sensor reading, three derived features were created: z-score (standardized distance from mean), percentile rank, and median absolute deviation. Additionally, interaction features capturing relationships between sensors were computed, including pH-TDS product, pH-TDS ratio, temperature-humidity product, and temperature-humidity ratio. Polynomial features, pH squared and TDS squared, were also generated to capture non-linear patterns. A composite feature representing the sum of absolute z-scores across all sensors was created as an overall anomaly indicator. This process expanded the feature space from 5 base features to 27 engineered features.

Following feature engineering, RobustScaler was applied to normalize the feature values. RobustScaler was chosen over StandardScaler because it is less sensitive to outliers, using median and interquartile range rather than mean and standard deviation for scaling. This

preprocessing step ensured all features contributed equally to model training regardless of their original measurement scales.

5.4.1 Random Forest and XGBoost Ensemble Model

The Random Forest classifier was applied to detect anomalies in sensor data collected from the hydroponic environment, including pH, TDS, temperature, humidity, and water level. The model operates as a bagging ensemble by constructing multiple decision trees on randomly sampled subsets of the training data. Each tree produces an independent classification, and the final output is determined through majority voting across all trees. This structure enables the model to capture complex, non-linear relationships between environmental variables while reducing overfitting and noise sensitivity. The prediction $\widehat{y_i}$ for a given observation x_i is mathematically represented as:

$$\widehat{y_i} = \text{model}\ \{h_1(x_i), h_2(x_i), \dots, h_K(x_i)\}$$

Equation 5.1 Random Forest Algorithm

where $h_K$denotes the K^{th}decision tree in the forest. The Random Forest component was configured with 200 trees, each limited to a depth of 10, ensuring a balance between accuracy and computational efficiency.

The XGBoost classifier complemented the Random Forest algorithm by applying a gradient boosting strategy to iteratively refine prediction errors left by previous trees. Unlike Random Forest, which trains trees independently, XGBoost builds each successive tree to correct the residuals of the ensemble, progressively improving anomaly detection accuracy. In this project, it was used to capture subtle sensor deviations that Random Forest might overlook, particularly under fluctuating water quality or temperature conditions. The model was trained with 200 boosting rounds, a learning rate of 0.05, and a maximum tree depth of 7. Its prediction process can be summarized as:

$$\widehat{y_i} = \sum_{t=1}^{T} f_t(x_i)$$

Equation 5.2 XGboost Algorithm

where each $f_t$represents a weak learner contributing to the final anomaly probability. Regularization terms $L_1$and L_2were included to minimize overfitting and enhance model generalization to unseen data from different growth phases. To improve robustness, the outputs of both models were integrated into a hybrid ensemble, where the predicted anomaly probabilities from Random Forest (P_{RF}) and XGBoost (P_{XGB}) were combined through weighted averaging:

$$\text{Pensemble}(x_i) = 0.5 \times P_{RF}(x_i) + 0.5 \times P_{XGB}(x_i)$$

Equation 5.3 Ensemble Algorithm

This balanced fusion allowed the system to leverage Random Forest's stability with XGBoost's precision, enhancing anomaly detection across sensor readings. A classification threshold of 0.45 was applied to flag data points as anomalous when the ensemble probability exceeded this value, enabling reliable identification of nutrient, pH, or water-level abnormalities in the hydroponic setup.

5.4.2 Model Evaluation Metrics

The trained ensemble model was evaluated using multiple performance metrics to comprehensively assess its effectiveness. The primary metrics included accuracy, precision, recall, F1-score, and ROC-AUC. On the held-out samples, the ensemble achieved the following results: accuracy of 97.25%, precision of 91.43%, recall of 92.09%, F1-score of 91.76%, and ROC-AUC of 0.9931. These metrics demonstrated strong overall performance with balanced precision-recall tradeoffs. The confusion matrix revealed the model's classification breakdown: 686 true negatives which were correctly identified normal conditions, 12 false positives showing normal conditions incorrectly flagged as anomalies, 11 false negatives which showed the missed anomalies, and 128 true positives indicating the correctly detected anomalies. The low false negative rate (7.9%) was particularly important for a safety-critical system where missing true anomalies could harm plant health.

Five-fold stratified cross-validation provided additional validation for model robustness. Random Forest achieved 98.12% accuracy and ±0.63% standard deviation, while XGBoost achieved 97.94% accuracy and ±0.58% standard deviation. The low variance across folds

and alignment with test set performance confirmed that the model generalized well and was not overfitted to training data.

The classification threshold was evaluated across multiple values (0.35 to 0.55). The selected threshold of 0.45 balanced precision and recall effectively, achieving 91.43% precision and 92.09% recall. Higher thresholds increased precision but reduced recall, while lower thresholds had the opposite effect.

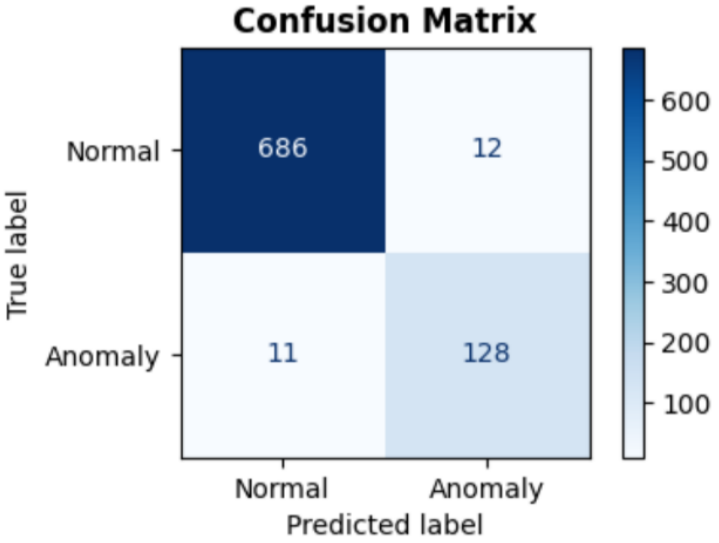


Figure 5.3 Confusion Matrix

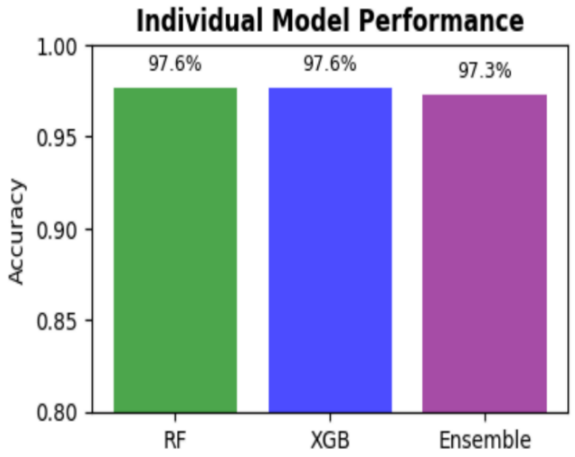


Figure 5.4 Individual Model Performance Comparisons

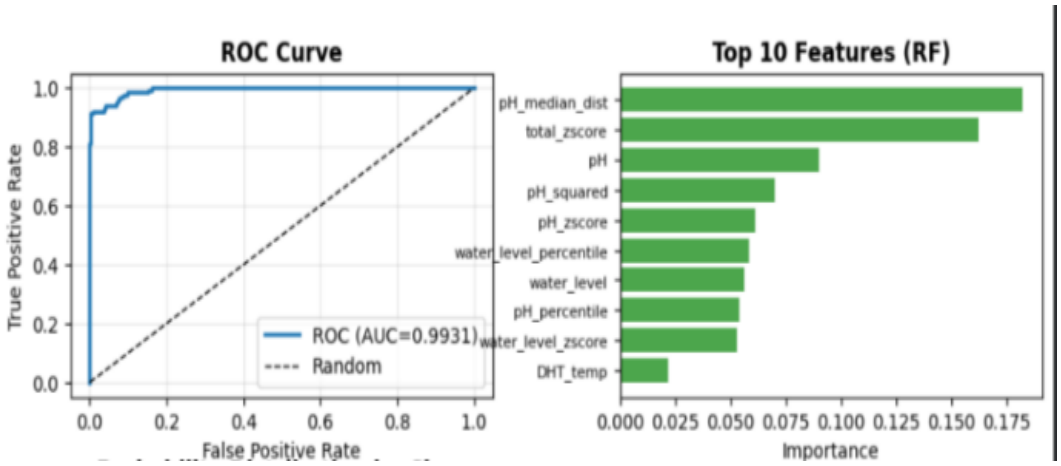


Figure 5.3 ROC Curve Ensemble Model

5.5 IoT System Implementation

The physical hydroponics monitoring system consisted of several interconnected components. The ESP32 microcontroller served as the central processing unit, handling sensor data acquisition and wireless communication. The sensor suite included a TDS sensor for nutrient concentration, a pH probe for monitoring acidity and alkalinity, a DHT22 sensor for ambient temperature and humidity, and an HC-SR04 ultrasonic sensor for water level detection. The actuator setup comprised a submersible pump for nutrient circulation and a relay module to control pump operation.

All sensors were installed within a small-scale lettuce setup. The pH and TDS sensors were placed directly in the nutrient reservoir, the ultrasonic sensor was positioned above the solution to measure water depth through distance calculation, and the DHT22 sensor was mounted near the lettuce to capture atmospheric conditions.

Each sensor was first tested individually to confirm stable performance across its expected measurement range. The pH probe was tested with solutions of different acidity levels to verify calibration. The TDS sensor was assessed using solutions ranging from 300 ppm to 1500 ppm. The ultrasonic sensor was evaluated at distances between 5 cm and 15 cm to confirm accurate detection of water height.

To assess the system's behaviour under abnormal conditions, controlled anomalies were introduced. The nutrient solution was adjusted to extreme pH values (3.0 and 8.5), and TDS levels were altered by diluting the water to 200 ppm and concentrating it to 1800 ppm. These tests demonstrated that the sensors reliably detected values outside the normal range, supporting effective anomaly detection.

After validating each component independently, full system integration was carried out. During testing, the ESP32 successfully captured data from all sensors simultaneously and transmitted the readings to Firebase Realtime Database using a temporary Wi-Fi connection. This method was used strictly for prototyping; for real-world deployment, GSM connectivity would be more suitable due to its broader coverage and improved reliability in agricultural environments. Data accuracy was confirmed by cross-checking Firebase entries with serial monitor outputs. The relay-controlled pump responded correctly to both manual inputs and automated triggers, confirming proper actuator performance.

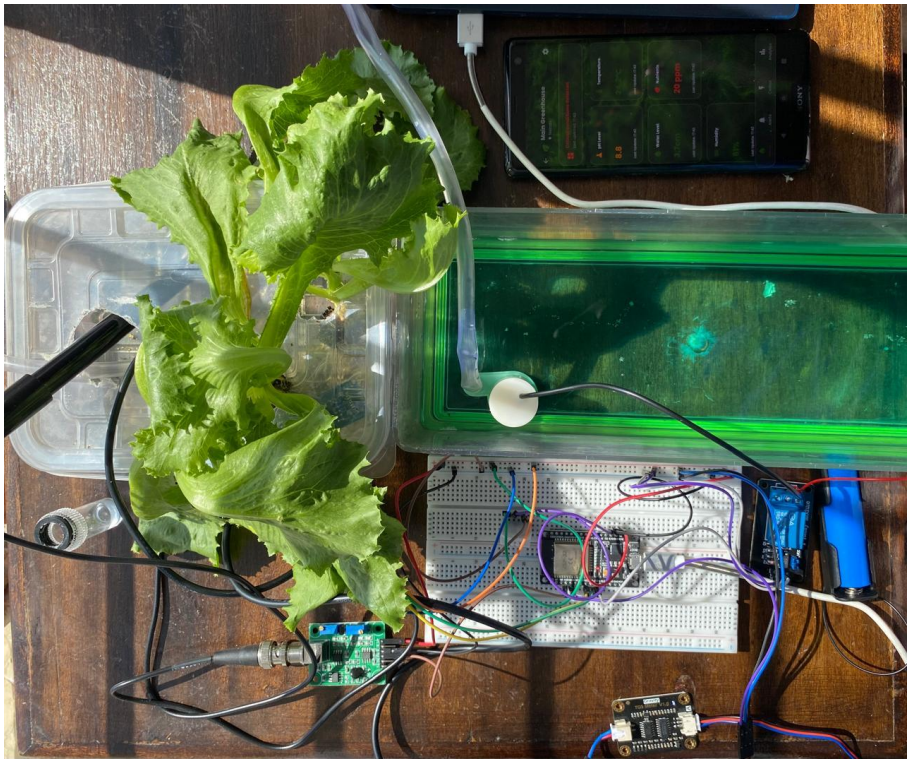


Figure 5.4 IoT System Implementation

The Firebase dashboard showed consistent real-time data updates, with timestamps and all sensor values correctly logged. Refer to Appendix 12 for the system's data flow representation.

5.6 System Implementation

System implementation involved the integration of multiple interconnected modules to create a functional and intelligent hydroponic monitoring platform. Each module was designed to perform a specific role, ranging from sensor data collection and anomaly detection to backend coordination and user interaction through a mobile application. Collectively, these modules enabled automated, real-time monitoring and control of the hydroponic system.

5.6.1 Sensor and Actuator Control Module

This module handled both the collection of sensor readings and the control of actuators such as the relay and water pump. Using the ESP32 microcontroller, environmental parameters including pH, Total Dissolved Solids (TDS), temperature, and humidity were continuously monitored. The microcontroller was programmed through the Arduino IDE to read sensor values at intervals of ten seconds each and then transmitted them to the backend via Wi-Fi. Additionally, actuator logic was incorporated to respond to abnormal readings such as activating the pump as per the user's instruction from the mobile app, when water levels dropped below the optimal range. The module maintained reliable operation through efficient data buffering and Wi-Fi reconnection routines, ensuring uninterrupted data flow. This setup simulated the real-time hydroponic environment and formed the foundation for the system's intelligent decision-making.

5.6.2 Machine Learning Analysis Module

This module integrated the ensemble learning approach using Random Forest and XGBoost algorithms to detect anomalies within sensor data streams. Each incoming data point underwent preprocessing consistent with the training phase before being evaluated by the models. The Random Forest classifier provided stable baseline predictions, while the XGBoost model refined these outcomes through gradient boosting, focusing on correcting residual errors. Anomalous conditions such as unusual pH or temperature fluctuations were flagged based on ensemble probability thresholds. When an anomaly was

detected, the module-initiated responses, such as issuing alerts and recommendations to the user through the mobile app. The analysis results, whether normal or anomalous, were logged in Firebase for real-time tracking and historical analysis. This module served as the system's decision engine, ensuring proactive management of hydroponic conditions.

5.6.3 Backend and Cloud Module

The backend module, developed using Node.js, functioned as the central communication hub connecting the IoT hardware, ML model, and mobile application. It facilitated data exchange through RESTful APIs, receiving sensor data from the ESP32 microcontroller, processing it, and relaying it to the ML module for analysis. Once predictions were made, the backend updated Firebase's Real-Time Database with the latest status, ensuring instantaneous reflection of readings on the user's dashboard. Firebase also handled user authentication and cloud functions for management of ML model requests, notification triggers, and maintaining data integrity. This design ensured scalability, security, and smooth synchronization between all system components.

5.6.4 Mobile Interface Module

The mobile interface was developed using Flutter to provide an intuitive and responsive user experience. Through Firebase authentication, users could create accounts, register farms, and securely access their system data. The application displayed real-time readings for pH, TDS, temperature, humidity and water levels, along with historical charts for trend analysis. When the ML module detected anomalies, push notifications were instantly delivered to the user, accompanied by recommended corrective actions. A lightweight chatbot was integrated to assist users by responding to quick questions and offering basic guidance. The interface also included visual cues such as color-coded readings on the dashboard for quick interpretation of system health. Designed for both simplicity and performance, the app ensured that even users with limited technical expertise could effectively monitor and control their hydroponic systems remotely. Refer to Appendix 9 and Appendix 10 for visuals of the screens.

5.6.5 Admin Dashboard Module

The admin dashboard provided a centralised platform for managing all system operations. Through this interface, the administrator could oversee all registered farms, user accounts, and connected devices while accessing detailed analytics to monitor overall system performance. The dashboard also included tools for generating reports, enabling the creation of exportable summaries for review and documentation. In addition, it supported full create, read, update, and delete operations for farms, users, and device records, ensuring smooth management across the system. Designed with clarity and ease of use, the dashboard allowed the administrator to work efficiently without needing advanced technical skills. Refer to Appendix 11 for dashboard screen visuals.

5.7 System Testing

The section evaluates whether all components of the smart hydroponic monitoring system function as intended under real and controlled conditions. It examines the performance of the mobile application, IoT integration, machine learning model, and alert mechanisms. The tests aim to validate system reliability, responsiveness, and overall operational accuracy.

5.7.1 Testing Results

The system was tested across multiple modules to confirm functionality, responsiveness, and integration accuracy.

Table 5.3 User Login Validation

Test ID	**Test Case Description**	**Test Data**	**Expected Results**	**Actual Results**	**Pass/Fail**
TC001	Validate user login using Firebase Authentication	User credentials (email, password)	User successfully authenticated and redirected to home page	As expected, refer to Appendix 5	Pass

Table 5.4 IoT Data Synchronization Test Case

Test ID	Test Case Description	Test Data	Expected Results	Actual Results	Pass/Fail
TC002	Verify real-time IoT data upload and synchronization with Firebase	Sensor readings from IoT setup (Temp=25°C, pH=6.5, TDS=850)	Sensor values displayed correctly on mobile dashboard	As expected, refer to Appendix 6	Pass

Table 5.5 Anomaly Detection

Test ID	Test Case Description	Test Data	Expected Results	Actual Results	Pass/Fail
TC003	Test anomaly detection using machine learning model	Sensor readings from IoT setup (low water level)	Abnormal readings correctly flagged	As expected, refer to Appendix 9	Pass

Table 5.6 Pump control test

Test ID	Test Case Description	Test Data	Expected Results	Actual Results	Pass/Fail
TC004	Verify activation of water pump through relay from user input	Low water level (distance < 10 cm)	Pump activates after user turns on pump from mobile app	As expected, refer to Appendix 8	Pass

Table 5.7 Mobile Alert Notification Test Case

Test ID	Test Case Description	Test Data	Expected Results	Actual Results	Pass/Fail
TC005	Validate alert notification trigger on abnormal readings	Firebase data indicating anomaly	Mobile app displays push notification	As expected, refer to Appendix 7	Pass

5.8 GitHub Documentation

This project was managed using GitHub, which served as the central platform for tracking progress, organising tasks, and maintaining version control throughout development. GitHub provided a structured environment where all work could be planned and executed systematically (GitHub link)

To guide development, the work was organised into milestones, each containing its own set of issues. These milestones aligned with the project's major phases. For example, Sprint 1 focused on user authentication, Sprint 2 handled the implementation of basic CRUD operations and API connections. Sprint 3 dealt with advanced data processing, Sprint 4 covered analytics and user interaction, and Sprint 5 handled exportable reports and documentation. All issues within each sprint were resolved before moving forward, ensuring consistent progress and clear visibility of completed tasks.

Version control was supported through multiple branches, created to separate new features, UI changes, and backend improvements. This helped maintain stability and avoid disruptions in the primary codebase. Pull requests were used to merge updates from feature branches into the main branch once the work was tested and confirmed to be stable. This process ensured that the main branch remained reliable and production ready throughout development.

Figure 5.5 GitHub Insights

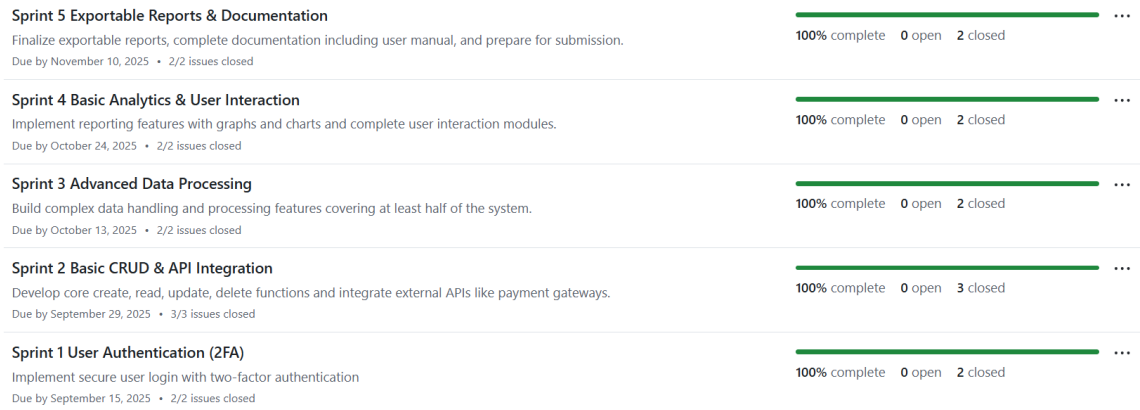


Figure 5.6: GitHub Milestones

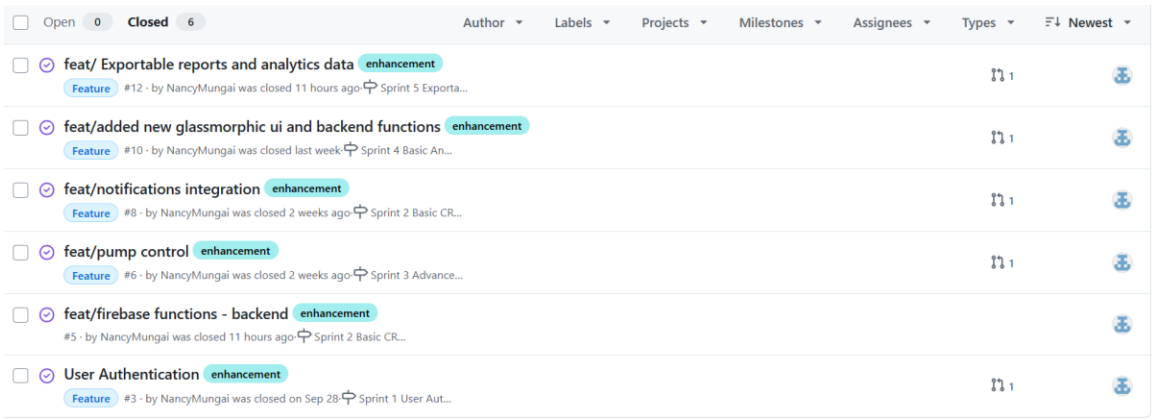


Figure 5.7: GitHub Issues

Chapter 6: Conclusions, Recommendations and Future Works

6.1 Conclusion

In conclusion, the development of the smart hydroponics monitoring system successfully addressed the challenge of real-time crop environment monitoring and management. By integrating IoT sensors, a cloud-based backend, and machine learning models (Random Forest and XGBoost), the system enabled accurate detection of anomalies in temperature, humidity, pH, and nutrient levels. The mobile application further enhanced accessibility by allowing users to receive instant alerts and view live data remotely. This innovation contributes to efficient resource utilization, improved crop yield, and reduced manual supervision in hydroponic farming. Overall, the project demonstrates how machine learning and IoT can be combined to create practical, data-driven agricultural solutions that benefit the farmers and Tech community at large.

6.2 Recommendations

Based on the project findings, it is recommended that future implementations focus on improving system reliability and data handling. Hosting the backend on a dedicated cloud server would ensure faster response times and uninterrupted data flow between sensors, the model, and the mobile app. Since the results showed that the accuracy of anomaly detection depends heavily on data quality, regular sensor calibration and data cleaning procedures should be maintained. The study also found that mobile accessibility improves system usability; therefore, optimizing the Flutter app for offline mode and using efficient Firebase queries would enhance performance. To support large-scale deployments, horizontal scalability should be emphasized by designing the system so multiple farms and sensor clusters can operate concurrently without performance issues. Additionally, careful selection of accurate sensors and clustering crops with similar environmental needs can improve monitoring efficiency and reduce unnecessary variations in readings. Finally, integrating advanced visualization tools would help farmers interpret environmental patterns more easily and make timely adjustments.

6.3 Future Works

Future work could focus on expanding the model to include predictive algorithms capable of forecasting plant growth trends or detecting early signs of disease. Incorporating computer vision through CNNs for image-based crop health monitoring could improve accuracy. A key area for advancement is establishing seamless data exchange across cloud, fog, and on-farm IoT layers. Implementing a cloud–fog–edge architecture would allow computationally heavy tasks to run in the cloud, real-time tasks to execute at the fog layer, and sensor operations to remain local on the farm, enabling more efficient and scalable interactions. Integration with automated nutrient dosing systems and energy-efficient IoT hardware would further enhance system performance and sustainability in commercial-scale hydroponic farms.

References

Ali, A., Niu, G., Masabni, J., Ferrante, A., & Cocetta, G. (2024). Integrated Nutrient Management of Fruits, Vegetables, and Crops through the Use of Biostimulants, Soilless Cultivation, and Traditional and Modern Approaches—A Mini Review. Agriculture, 14(8), Article 8. https://doi.org/10.3390/agriculture14081330

Anjali, ., Kopal, S., Rohan Raju, T., Shubham, S., Sahil, K., & Shilpa, K. (2025). Hydroponics: An Innovative Approach to Sustainable High-Value Crop Cultivation. Journal of Scientific Research and Reports. https://hal.science/hal-05032591

Anusha, M., Soni, A., Mohapatra, R., Kumar, V., Bhanusree, M., Giri, D., & Gupta, S. (2025). Exploring the Role of IoT in Transforming Agriculture: Current Applications and Future Prospects. Archives of Current Research International, 25(4), 85–105. https://doi.org/10.9734/acri/2025/v25i41139

Asaduzzaman, M., Niu, G., & Asao, T. (2022). Editorial: Nutrients Recycling in Hydroponics: Opportunities and Challenges Toward Sustainable Crop Production Under Controlled Environment Agriculture. Frontiers in Plant Science, 13. https://doi.org/10.3389/fpls.2022.845472

Bakirov, K., Tussupov, J., Tussupov, A., Shayea, I., & Shoman, A. (2025). Application of a Hybrid Model for Data Analysis in Hydroponic Systems. Technologies, 13(5), Article 5. https://doi.org/10.3390/technologies13050166

Bashir, N., Bilal, M., Liaqat, M., Marjani, M., Malik, N., & Ali, M. (2021). Modeling Class Diagram using NLP in Object-Oriented Designing. 2021 National Computing Colleges Conference (NCCC), 1–6. https://doi.org/10.1109/NCCC49330.2021.9428817

Bhandari, N. S., Bhandari, N., Agarwal, R., & Sharma, P. K. (2024). An Insight on Artificial Intelligence (AI) and Internet of Things (IoT) driven Hydroponics Farming. 2024 5th International Conference on Image Processing and Capsule Networks (ICIPCN), 496–501. https://doi.org/10.1109/ICIPCN63822.2024.00087

Brahmia, Z., Grandi, F., & Oliboni, B. (2024). Schema Versioning in Databases: A Literature Review. Computing Open, 02, 2430002. https://doi.org/10.1142/S2972370124300024

Bua, C., Adami, D., & Giordano, S. (2024). GymHydro: An Innovative Modular Small-Scale Smart Agriculture System for Hydroponic Greenhouses. Electronics, 13(7), Article 7. https://doi.org/10.3390/electronics13071366

Dennison, M. S., Kumar, P. S., Wamyil, F., Meji, M. A., & Ganapathy, T. (2025). The role of automation and robotics in transforming hydroponics and aquaponics to large scale. Discover Sustainability, 6(1), 105. https://doi.org/10.1007/s43621-025-00908-4

Dennison, M.S., Kumar, P.S, & Wamyil. (2025). The role of automation and robotics in transforming hydroponics and aquaponics to large scale | Discover Sustainability. https://link.springer.com/article/10.1007/s43621-025-00908-4

Dhal, S. B., & Kar, D. (2024). Transforming Agricultural Productivity with AI-Driven Forecasting: Innovations in Food Security and Supply Chain Optimization. Forecasting, 6(4), 925–951. https://doi.org/10.3390/forecast6040046

Donat, M., Geistert, J., Grahmann, K., Bloch, R., & Bellingrath-Kimura, S. D. (2022). Patch cropping- a new methodological approach to determine new field arrangements that increase the multifunctionality of agricultural landscapes. Computers and Electronics in Agriculture, 197, 106894. https://doi.org/10.1016/j.compag.2022.106894

Flutter documentation. (2025). https://docs.flutter.dev

GeeksforGeeks. (2024, October). Agile Prototyping. GeeksforGeeks. https://www.geeksforgeeks.org/agile-prototyping/

Get started with GitHub documentation. (2025). GitHub Docs. https://docs-internal.github.com/_next/data/lR633-ZTexqIO_0vnHRFJ/en/free-pro-team%40latest/get-started.json?versionId=free-pro-team%40latest&productId=get-started

GnagnarellaDirector, A. H. A., Frameworks, & ServicesFirebase. (2025). What's new in Firebase at I/O 2025. The Firebase Blog. https://firebase.blog/

Haji, S. H., & Sallow, A. B. (2021). IoT for Smart Environment Monitoring Based on Python: A Review. Asian Journal of Research in Computer Science, 57–70. https://doi.org/10.9734/ajrcos/2021/v9i130215

Hemlata. (2023, May 16). Automated Smart Gardening & Monitoring System With Grow Room |. https://www.getniwa.com/

Jacob, K. (2017). Factors Influencing Adoption of Urban Hydroponic Farming. A Case of Meru Town, Meru County, Kenya.

Jha, P., Sahu, M., & Isobe, T. (2023). A UML Activity Flow Graph-Based Regression Testing Approach. Applied Sciences, 13(9), Article 9. https://doi.org/10.3390/app13095379

Kinast, A., Doerner, K. F., & Rinderle-Ma, S. (2021). Biased random-key genetic algorithm for cobot assignment in an assembly/disassembly job shop scheduling problem. Procedia Computer Science, 180, 328–337. https://doi.org/10.1016/j.procs.2021.01.170

Le, T.-T.-H., Oktian, Y. E., & Kim, H. (2022). XGBoost for Imbalanced Multiclass Classification-Based Industrial Internet of Things Intrusion Detection Systems. Sustainability, 14(14), Article 14. https://doi.org/10.3390/su14148707

Liakos, K. G., Busato, P., Moshou, D., Pearson, S., & Bochtis, D. (2018). Machine Learning in Agriculture: A Review. Sensors, 18(8), Article 8. https://doi.org/10.3390/s18082674

Mamatha, V., & Kavitha, J. C. (2023). Remotely monitored Web based Smart Hydroponics System for Crop Yield Prediction using IoT. 2023 IEEE 8th International Conference for Convergence in Technology (I2CT), 1–6. https://doi.org/10.1109/I2CT57861.2023.10126337

Mohmed, G., Hasanaliyeva, G., O'Mahony, R., & Lu, C. (2025). Optimising Nutrient Formulations through Artificial Intelligence Model to Reduce Excessive Fertigation in Lettuce grown in Hydroponic Systems. IEEE Access, 1–1. https://doi.org/10.1109/ACCESS.2025.3571730

Mosaad, B., Abdulla, R., & Rana, M. E. (2023a). Recommendations for a Vertical Farming System Using Hydroponics, Machine Learning and IoT. 2023 4th International

Conference on Data Analytics for Business and Industry (ICDABI), 373–378. https://doi.org/10.1109/ICDABI60145.2023.10629318

Mosaad, B., Abdulla, R., & Rana, M. E. (2023b). Recommendations for a Vertical Farming System Using Hydroponics, Machine Learning and IoT. 2023 4th International Conference on Data Analytics for Business and Industry (ICDABI), 373–378. https://doi.org/10.1109/ICDABI60145.2023.10629318

Muccini, H., & Vaidhyanathan, K. (2021). Software Architecture for ML-based Systems: What Exists and What Lies Ahead. 2021 IEEE/ACM 1st Workshop on AI Engineering - Software Engineering for AI (WAIN), 121–128. https://doi.org/10.1109/WAIN52551.2021.00026

nitsan. (2022, December 28). What is Ebb and Flow In Hydroponics Systems? | Growee. Growee Smart Hydroponics. https://getgrowee.com/hydroponic-ebb-and-flow-system/

Olajide, A. I., Bamiro, O. M., Adeyonu, A. G., & Faronbi, O. A. (2024). Technological innovations and Potential Adaptation Strategies for Agriculture in developing countries; A Case study of Nigeria. 2024 IEEE 5th International Conference on Electro-Computing Technologies for Humanity (NIGERCON), 1–5. https://doi.org/10.1109/NIGERCON62786.2024.10927284

Pattanayak, S. (2019). Intelligent Projects Using Python: 9 real-world AI projects leveraging machine learning and deep learning with TensorFlow and Keras. Packt Publishing Ltd.

Peskett, M. (2023, June 8). Funding for Israel's urban hydroponics specialist—Growee. Vertical Farming Today. https://www.verticalfarmingtoday.com/news/hydroponics/funding-for-israels-urban-hydroponics-specialist-growee.html

Puengsungwan, S., & Jirasereeamornkul, K. (2019). Internet of Things (IoTs) based hydroponic lettuce farming with solar panels. 2019 International Conference on Power, Energy and Innovations (ICPEI), 86–89. https://doi.org/10.1109/ICPEI47862.2019.8944986

Rahman, M. A., Chakraborty, N. R., Sufiun, A., Banshal, S. K., & Tajnin, F. R. (2024a). An AIoT-based hydroponic system for crop recommendation and nutrient

parameter monitorization. Smart Agricultural Technology, 8, 100472. https://doi.org/10.1016/j.atech.2024.100472

Rahman, M. A., Chakraborty, N. R., Sufiun, A., Banshal, S. K., & Tajnin, F. R. (2024b). An AIoT-based hydroponic system for crop recommendation and nutrient parameter monitorization. Smart Agricultural Technology, 8, 100472. https://doi.org/10.1016/j.atech.2024.100472

Rajendiran, G. & R. (2024). Optimizing Lettuce Crop Yield Prediction in an Indoor Aeroponic Vertical Farming System Using IoT-Integrated Machine Learning Regression Models. ResearchGate. https://www.researchgate.net/publication/381638477_Optimizing_Lettuce_Crop_Yield_Prediction_in_an_Indoor_Aeroponic_Vertical_Farming_System_Using_IoT-Integrated_Machine_Learning_Regression_Models

Ramakrishnam Raju, S. V. S., Dappuri, B., Ravi Kiran Varma, P., Yachamaneni, M., Verghese, D. M. G., & Mishra, M. K. (2022). Design and Implementation of Smart Hydroponics Farming Using IoT-Based AI Controller with Mobile Application System. Journal of Nanomaterials, 2022(1), 4435591. https://doi.org/10.1155/2022/4435591

Rosca, C., Stancu, A., & Popescu. (2025). The Impact of Cloud Versus Local Infrastructure on Automatic IoT-Driven Hydroponic Systems. ResearchGate. https://www.researchgate.net/publication/390568262_The_Impact_of_Cloud_Versus_Local_Infrastructure_on_Automatic_IoT-Driven_Hydroponic_Systems

Rosero-Montalvo, P. D., López-Batista, V. F., & Peluffo-Ordóñez, D. H. (2022). A New Data-Preprocessing-Related Taxonomy of Sensors for IoT Applications. Information, 13(5), Article 5. https://doi.org/10.3390/info13050241

Salem, N., Al-Tarawneh, K., Hudaib, A., Salem, H., Tareef, A., Salloum, H., & Mazzara, M. (2024). Generating database schema from requirement specification based on natural language processing and large language model. Computer Research and Modeling, 16(7), 1703–1713. https://doi.org/10.20537/2076-7633-2024-16-7-1703-1713

Sangeetha, T., & Ezhumalai, P. (2025). Optimizing plant health monitoring: Improved accuracy and the computational efficiency with stacked machine learning models

and feature filtering. Bulletin of Electrical Engineering and Informatics, 14(2), Article 2. https://doi.org/10.11591/eei.v14i2.8809

Shalash, O., Hassan, N., Métwalli, A., & Elhefny, A. (2025). HydroGrowNet of Batavia Dataset. 4. https://doi.org/10.17632/g6cm3v3wdp.4

Shalash, O., Métwalli, A., Elhefny, A., Rezk, N., El Gohary, F., El Hennawy, O., Akrab, F., Shawky, A., Mohamed, Z., Hassan, N., & Hassanen, M. (2025). Enhancing Hydroponic Farming with Machine Learning: Growth Prediction and Anomaly Detection (SSRN Scholarly Paper No. 5079228). Social Science Research Network. https://doi.org/10.2139/ssrn.5079228

Shareef, U., Rehman, A. U., & Ahmad, R. (2024). A Systematic Literature Review on Parameters Optimization for Smart Hydroponic Systems. AI, 5(3), Article 3. https://doi.org/10.3390/ai5030073

Staiano, F. (2022). Designing and Prototyping Interfaces with Figma: Learn essential UX/UI design principles by creating interactive prototypes for mobile, tablet, and desktop. Packt Publishing Ltd.

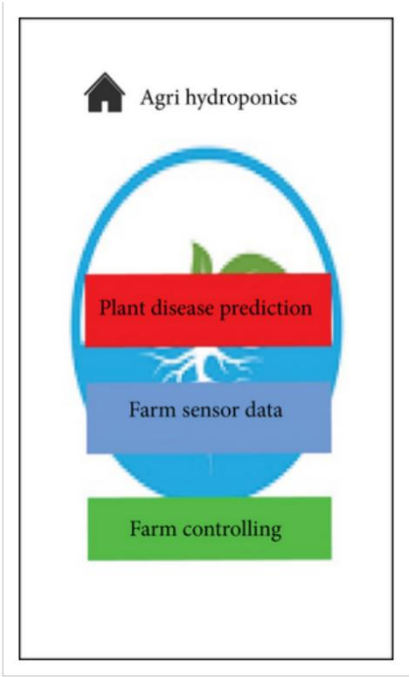
Suresh, C., & S, S. (2025). Enhancing Hydroponic Farming Using IoT Integration and Machine Learning Techniques. 2025 International Conference on Multi-Agent Systems for Collaborative Intelligence (ICMSCI), 439–444. https://doi.org/10.1109/ICMSCI62561.2025.10894677

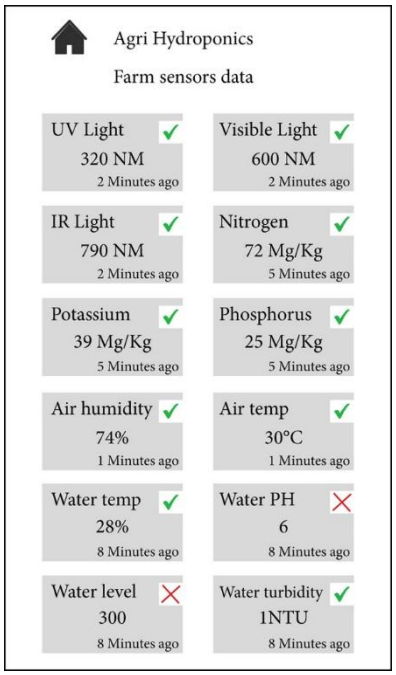
Syafrudin, M., Alfian, G., Fitriyani, N. L., & Rhee, J. (2018). Performance Analysis of IoT-Based Sensor, Big Data Processing, and Machine Learning Model for Real-Time Monitoring System in Automotive Manufacturing. Sensors, 18(9), Article 9. https://doi.org/10.3390/s18092946

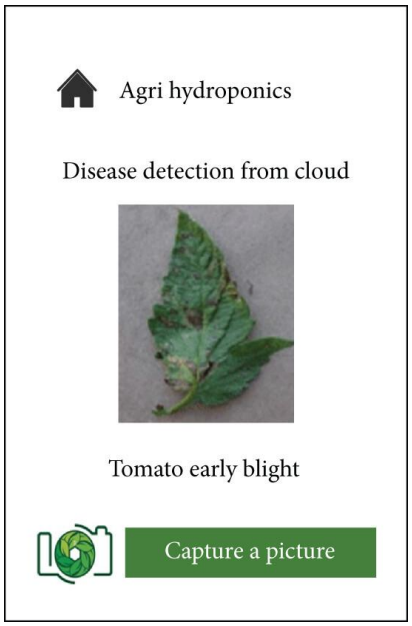
Tegbaru, A., Fitzsimons, J. G., & Gondwe, T. (2021). Womens empowerment: A gender outcome of an improved agriculture health and nutrition project in Zambia and Malawi. Journal of Agricultural Extension and Rural Development, 13(2), 125–137. https://doi.org/10.5897/JAERD2021.1234

Appendix

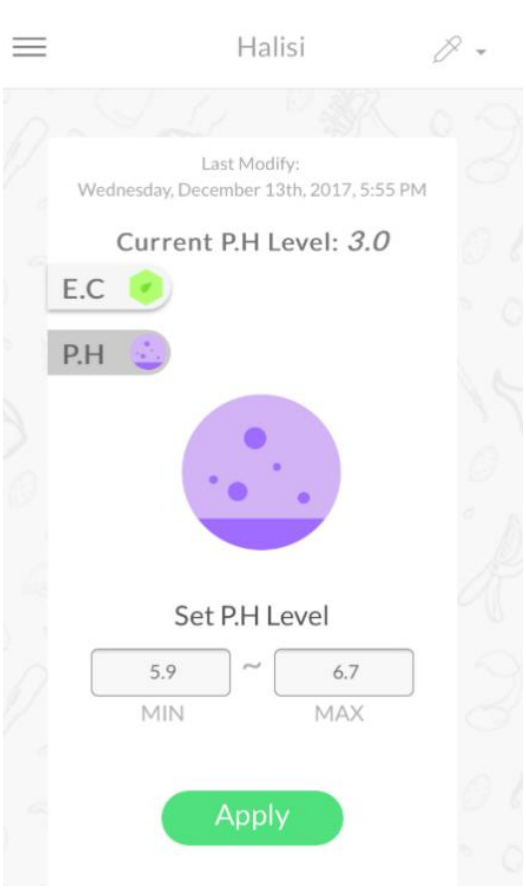
Appendix 1: Related Works: Agrihydroponics Mobile Application UI

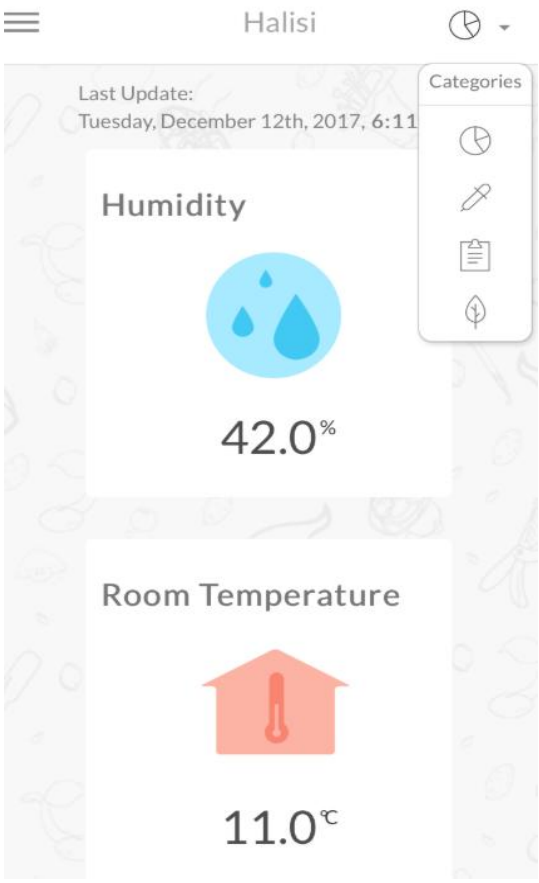




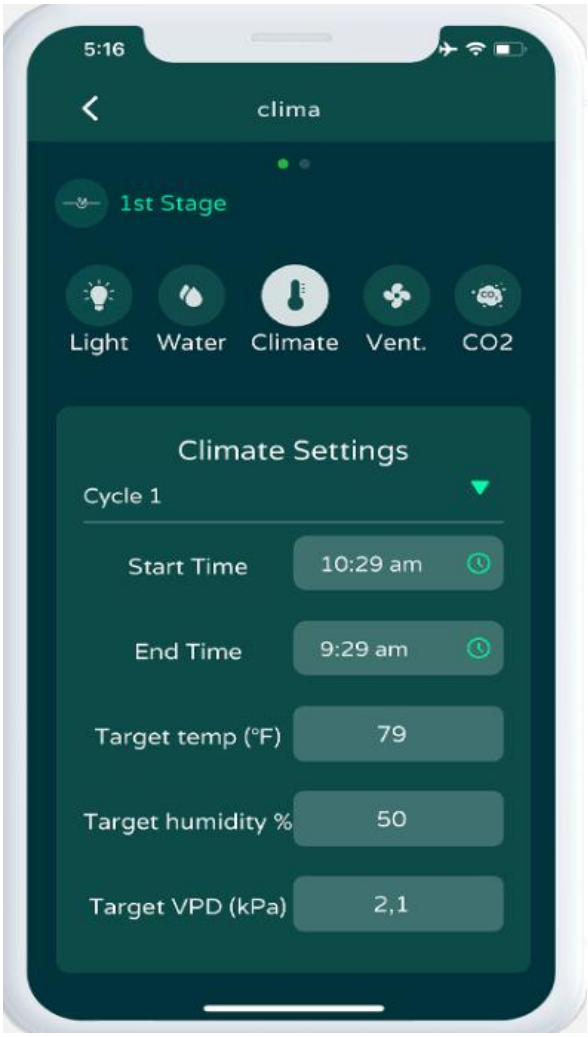


Appendix 2: Related Works: Growee Mobile application

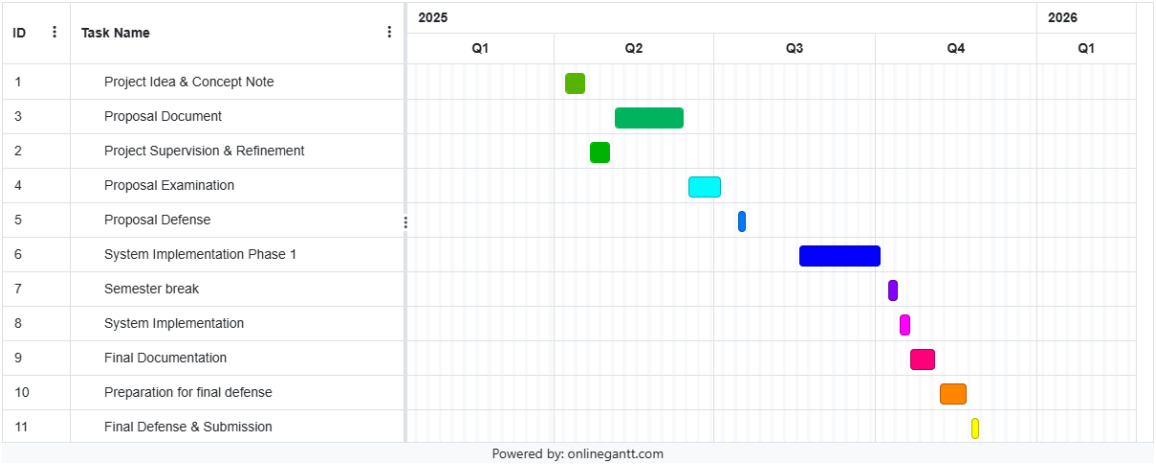




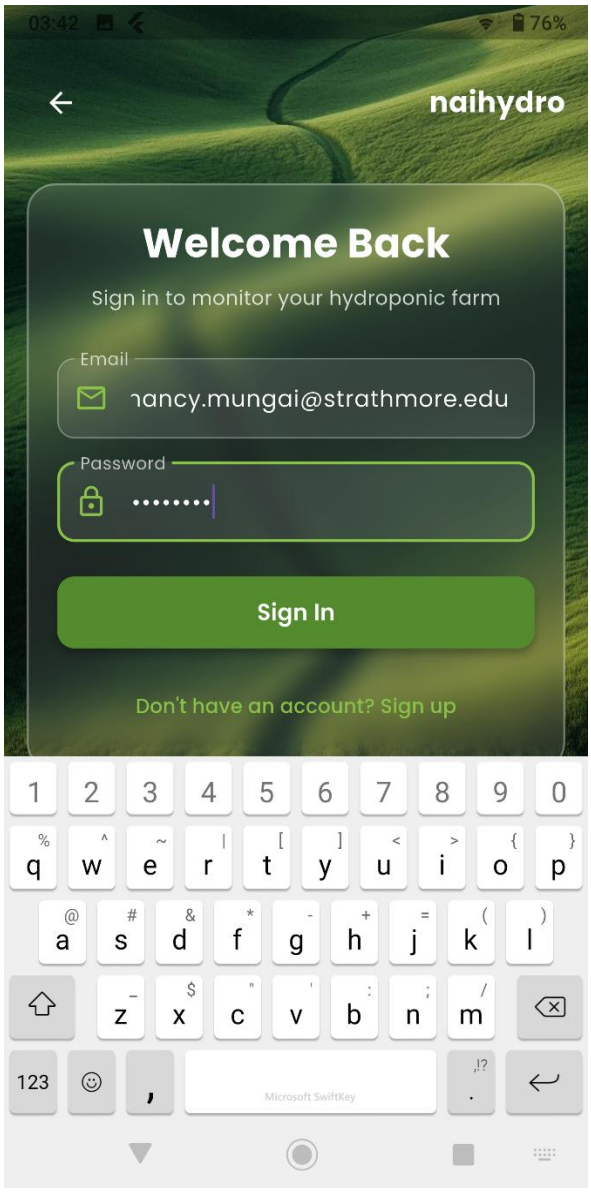
Appendix 3: Related Works: Niwa Mobile application

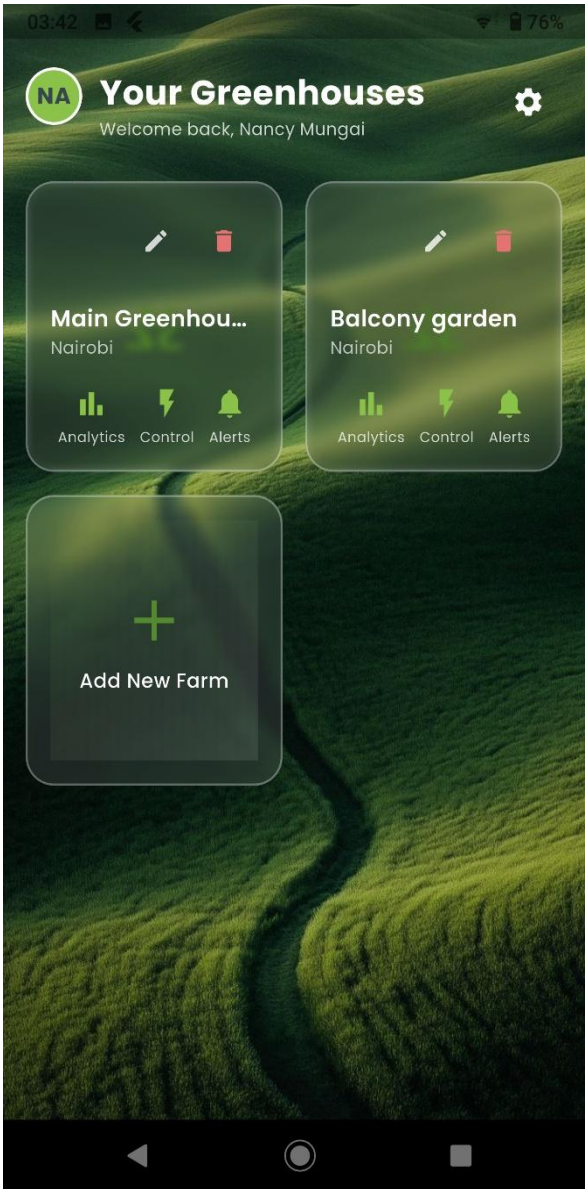


Appendix 4: Gantt Chart

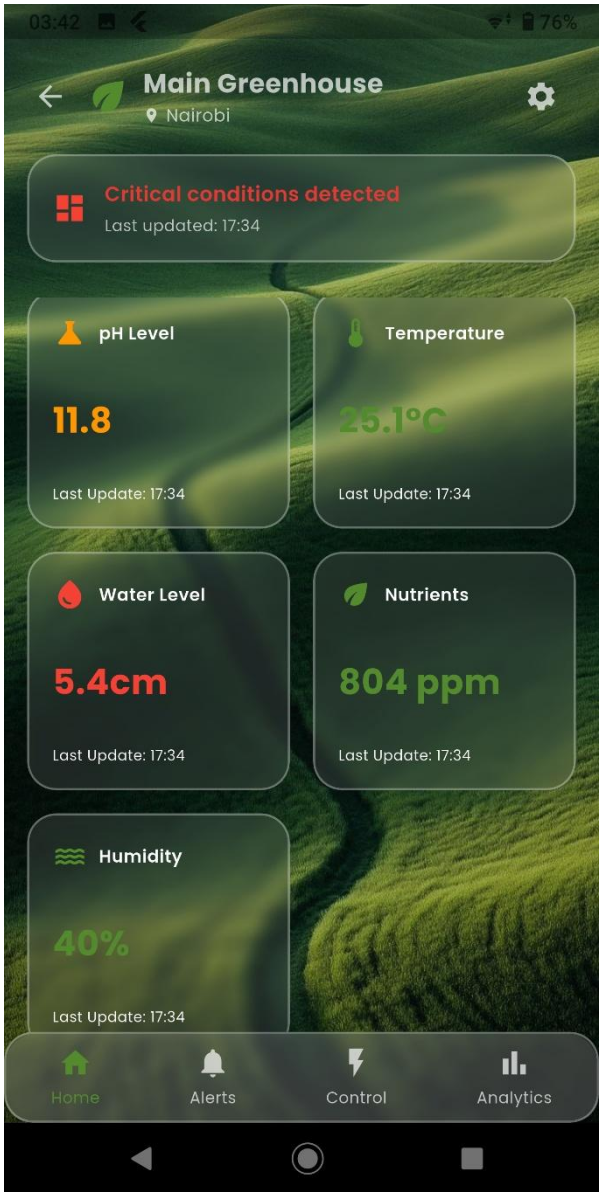


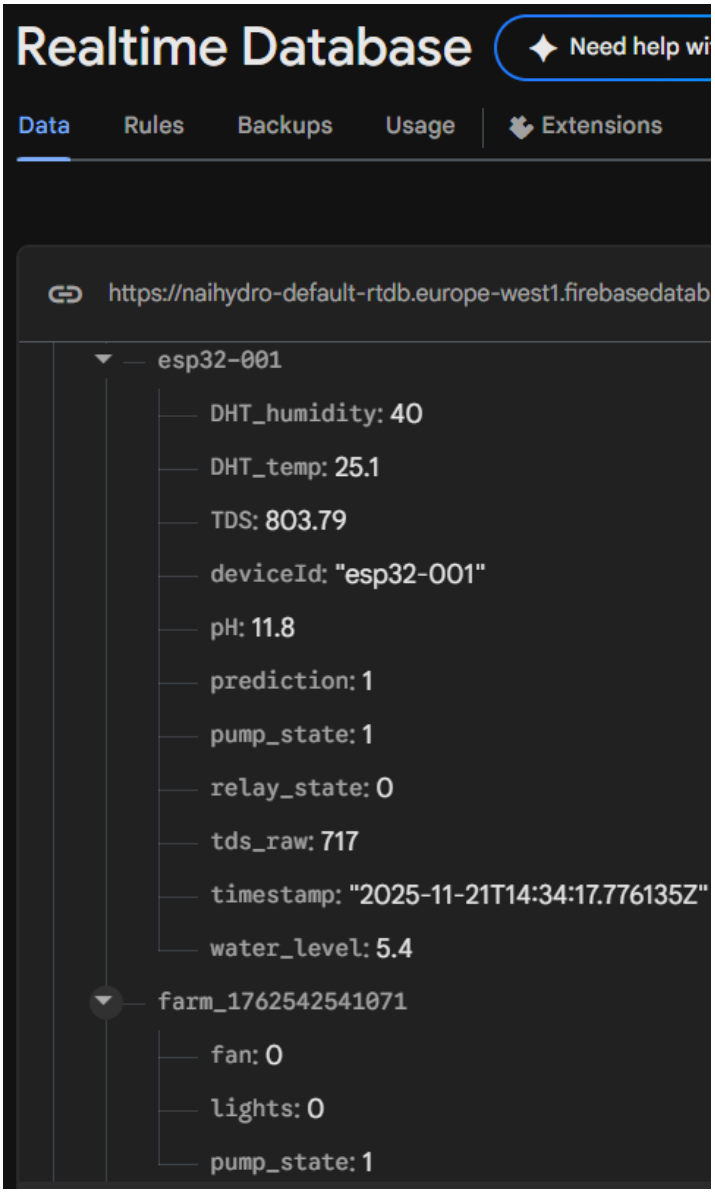
Appendix 5: Login Test Case



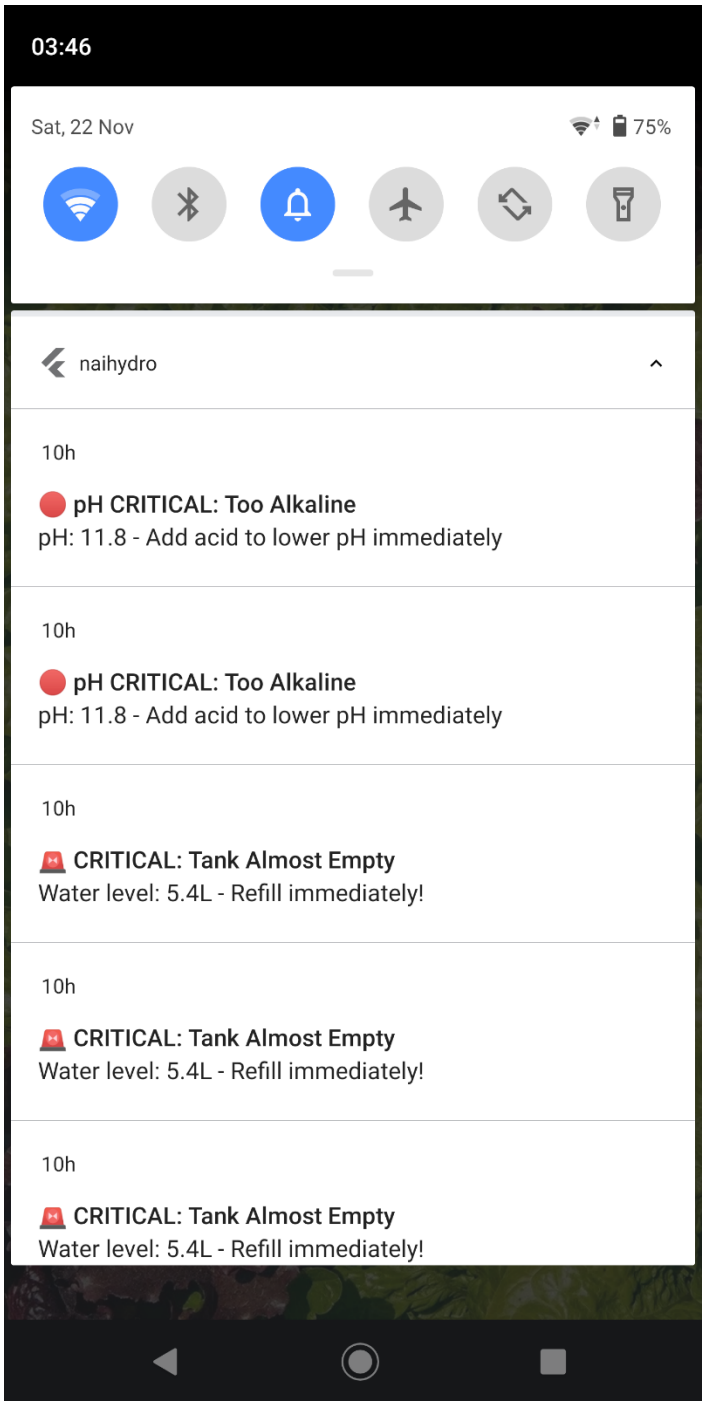


Appendix 6: Data Synchronization Test Case

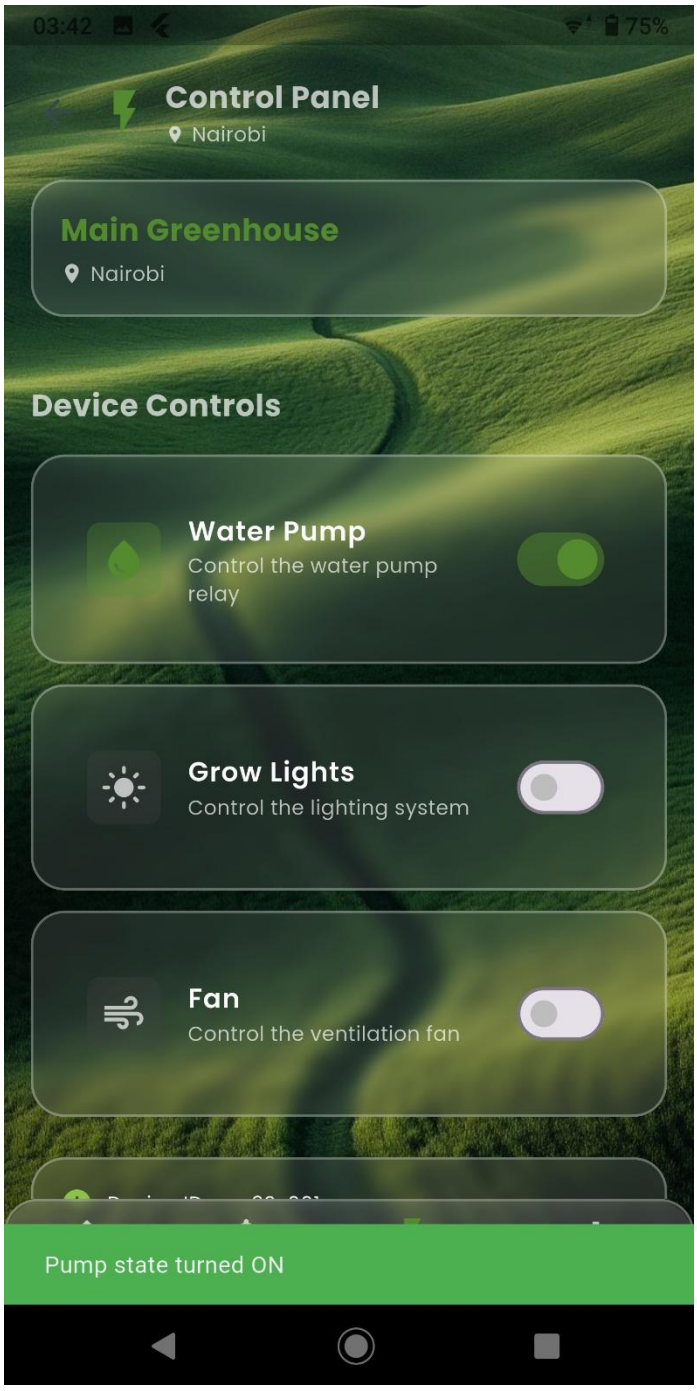


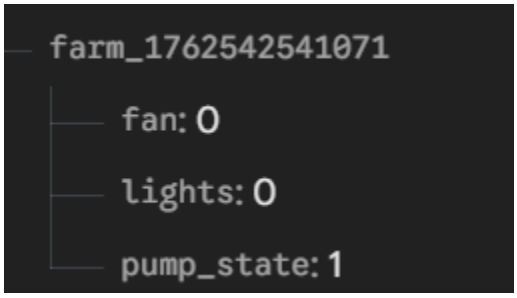


Appendix 7: Mobile Alert Notification Test Case

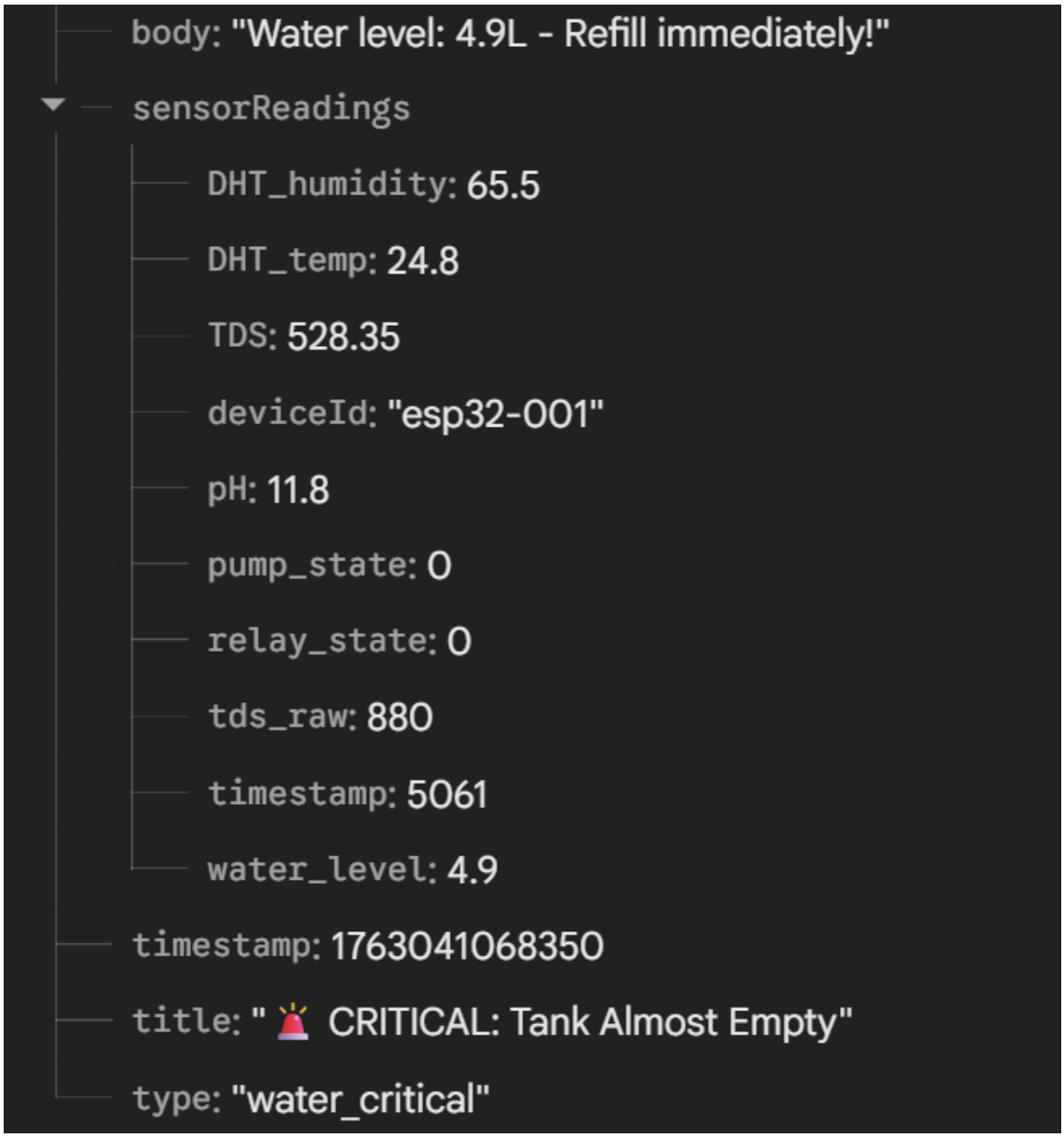


Appendix 8: Automated Pump Control Test Case

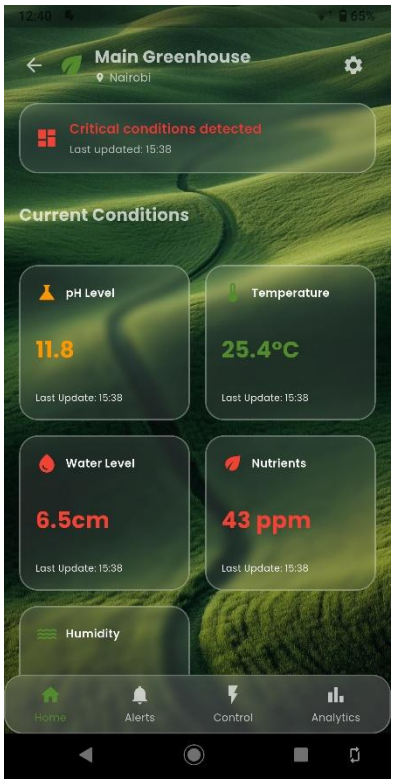


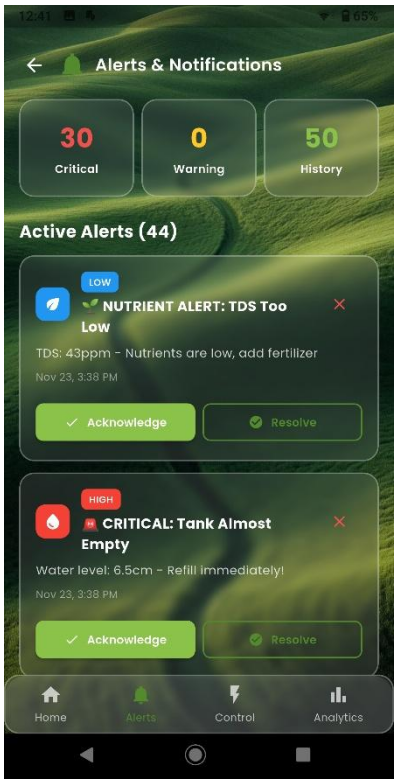


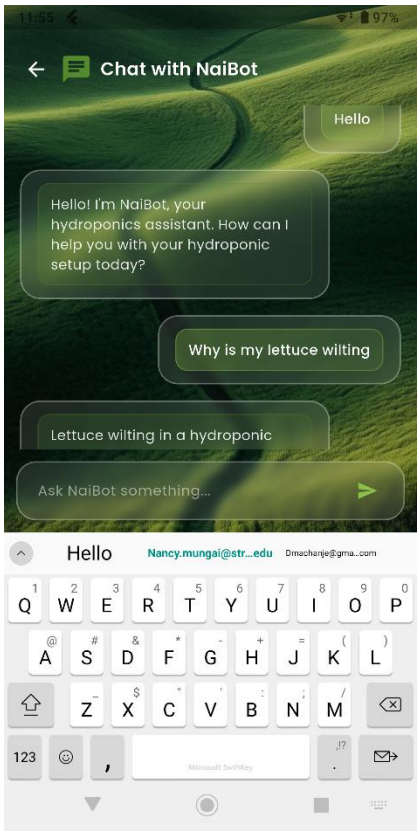
Appendix 9: Anomaly Detection Test Case

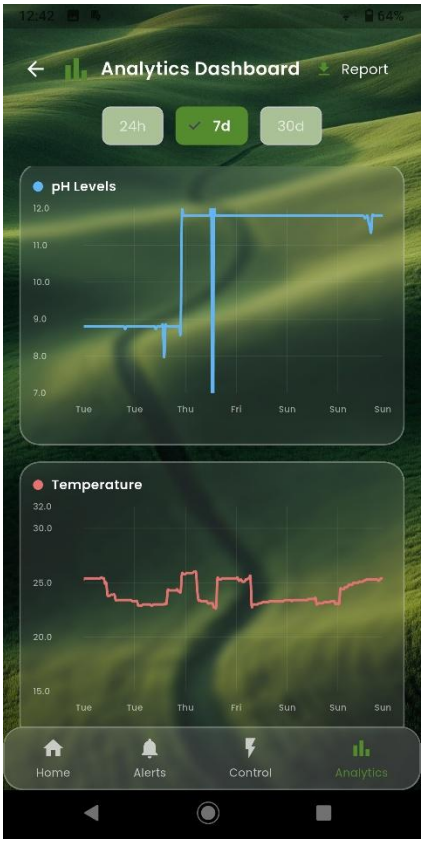


Appendix 10: Mobile Application Screens

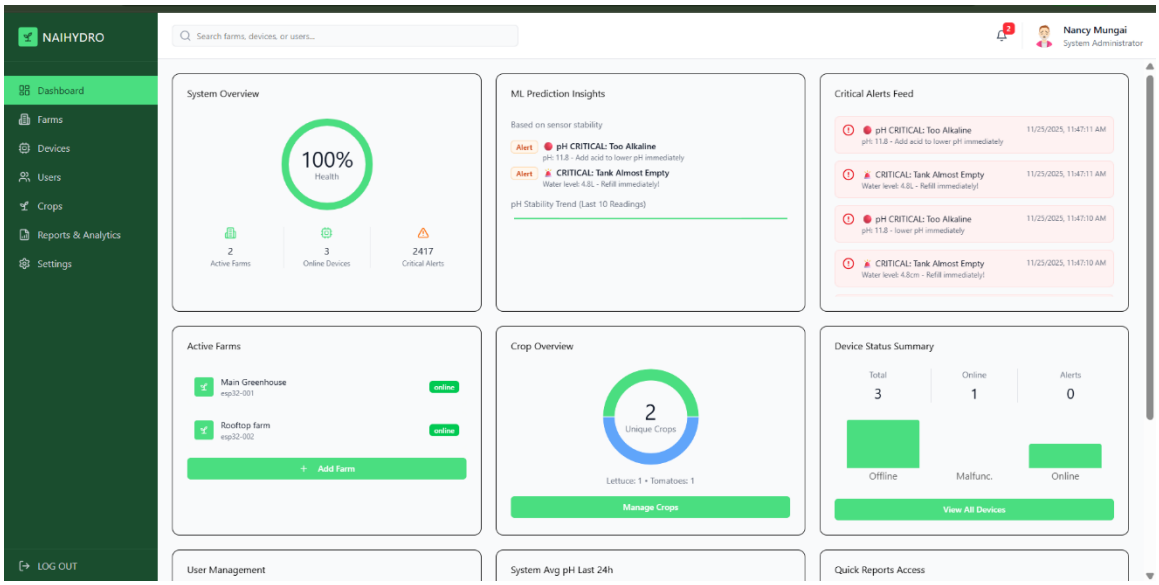


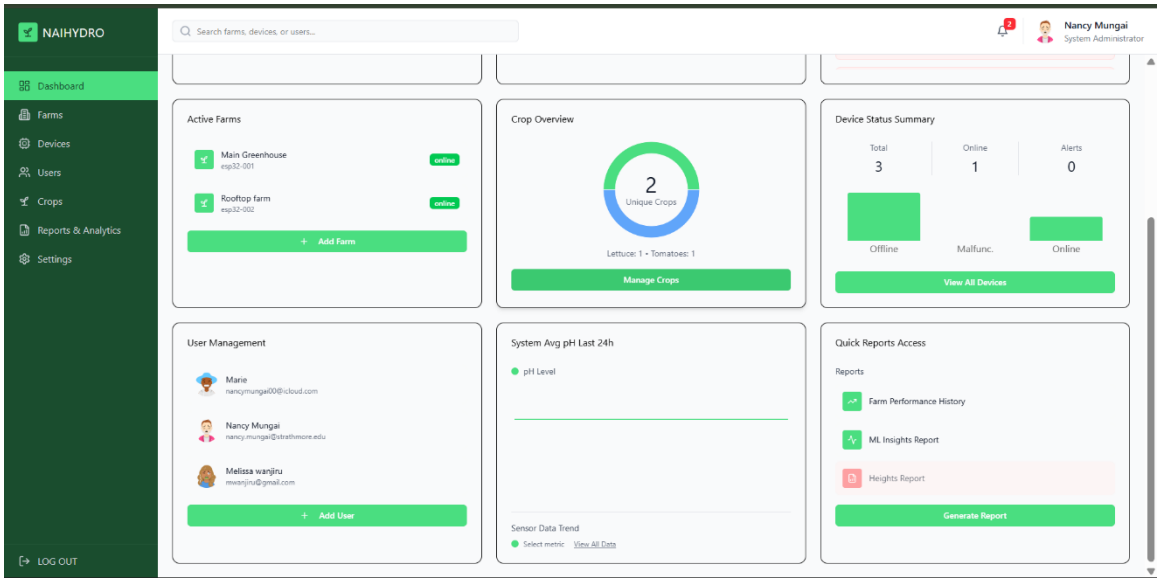






Appendix 11: Admin Dashboard UI





Appendix 12: Data Flow from IoT to Firebase

```
11:45:38.765 -> 📤 Publishing sensor data:
11:45:38.765 -> {"deviceId":"esp32-001","timestamp":40169,"DHT_temp":24.30,"DHT_humidity":58.70,"pH":11.80,"TDS":859.66,"water_level":4.82}
11:45:38.811 ->    → /devices/esp32-001/latest
11:45:38.811 ->    → Cloud Function will process this
11:45:38.998 -> ✅ Data uploaded successfully (code 200)
11:45:38.998 ->    Cloud Function will process and save to /processed and /history
11:45:41.565 -> 📡 Firebase pump_state: 0 | Current relay: 0
11:45:44.640 -> 📡 Firebase pump_state: 0 | Current relay: 0
11:45:47.570 -> 📡 Firebase pump_state: 0 | Current relay: 0
11:45:48.796 ->
11:45:48.796 -> 📤 Publishing sensor data:
11:45:48.796 -> {"deviceId":"esp32-001","timestamp":50188,"DHT_temp":24.30,"DHT_humidity":58.40,"pH":11.80,"TDS":851.15,"water_level":4.80}
11:45:48.796 ->    → /devices/esp32-001/latest
11:45:48.796 ->    → Cloud Function will process this
11:45:49.004 -> ✅ Data uploaded successfully (code 200)
11:45:49.046 ->    Cloud Function will process and save to /processed and /history
11:45:50.685 -> 📡 Firebase pump_state: 0 | Current relay: 0
11:45:53.746 -> 📡 Firebase pump_state: 0 | Current relay: 0
11:45:56.825 -> 📡 Firebase pump_state: 1 | Current relay: 0
11:45:56.825 -> 🔄 Command changed from 0 to 1
11:45:56.825 -> ✓ PUMP ON (command from app)
11:45:56.825 -> ✓ Relay pin is now: 0
11:45:58.819 ->
```