VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belagavi – 590 018.

A PROJECT REPORT

on

"LITE HARVEST: Wick-Based Hydroponics System With Mobile Application"

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

Bachelor of Engineering

in

Computer Science and Engineering

by

HASINA PARVIN	**:**	**1VI20CS047**
PRERANA B	**:**	**1VI20CS085**
PRIANKA D	**:**	**1VI20CS086**
RAJASHREE GAUDA	**:**	**1VI20CS088**

Under the supervision of

Dr. Shilpa G V
Associate Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VEMANA INSTITUTE OF TECHNOLOGY

BENGALURU – 560 034

2023 - 24

Karnataka ReddyJana Sangha®

VEMANA INSTITUTE OF TECHNOLOGY

(Affiliated to Visvesvaraya Technological University, Belagavi)

Koramangala, Bengaluru-34.

Department of Computer Science and Engineering

Certificate

Certified that the project work entitled "**LITE HARVEST: Wick-Based Hydroponics System With Mobile Application**" carried out jointly by **Hasina Parvin (1VI20CS047), Prerana B (1VI20CS085), Prianka D (1VI20CS086), Rajashree Gauda (1VI20CS088),** are bonafide students **of Vemana Institute of Technology** in partial fulfillment for the award of **Bachelor of Engineering** in **Computer Science and Engineering** of the **Visvesvaraya Technological University, Belagavi** during the year 2023-24. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report. The project report has been approved as it satisfies the academic requirements in respect of the project work prescribed for the said degree.

Supervisor	**HOD**	**Principal**
Dr. Shilpa G V	Dr. M. Ramakrishna	Dr. Vijayasimha Reddy. B. G.

Submitted for the university examination (viva-voce) held on ……………………….

External Viva

Internal Examiner **External Examiner**

1.

2.

DECLARATION BY THE CANDIDATES

We the undersigned solemnly declare that the project report “LITE HARVEST: Wick-Based Hydroponics System With Mobile Application” is based on our own work carried out during the course of our study under the supervision of ‘Dr. SHILPA G V’.

We assert the statements made and conclusions drawn are an outcome of our project work. We further certify that,

a. The work contained in the report is original and has been done by us under the general supervision of my supervisor.

b. The work has not been submitted to any other Institution for any other degree/diploma/certificate in this university or any other University of India or abroad.

c. We have followed the guidelines provided by the university in writing the report.

d. Whenever we have used materials (data, theoretical analysis, and text) from other sources, we have given due credit to them in the text of the report and their details are provided in the references.

Date:

Place: Bengaluru

Project Team Members:

Hasina Parvin	1VI20CS047
Prerana B	1VI20CS085
Prianka D	1VI20CS086
Rajashree Gauda	1VI20CS088

ACKNOWLEDGEMENT

First we would like to thank our parents for their kind help, encouragement and moral support.

We thank **Dr. Vijayasimha Reddy. B. G,** Principal, Vemana Institute of Technology, Bengaluru for providing the necessary support.

We would like to place on record our regards to **Dr. M. Ramakrishna,** Professor and Head, Department of Computer Science and Engineering for his continued support.

We would like to thank our project coordinators **Mrs. Mary Vidya John,** Assistant Professor and **Mrs. Vijayashree HP,** Assistant Professor, Dept. of CSE for their support and coordination.

We would like to thank our project guide **Dr. Shilpa G V,** Associate Professor, Dept. of CSE for her/his continuous support, valuable guidance and supervision towards successful completion of the project work.

We also thank all the Teaching and Non-teaching staff of Computer Science and Engineering Department, who have helped us to complete the project in time.

Project Team Members:

HASINA PARVIN	**1VI20CS047**
PRERANA B	**1VI20CS085**
PRIANKA D	**1VI20CS086**
RAJASHREE GAUDA	**1VI20CS088**

CONTENTS

ABSTRACT

This project focuses on the development of a wick-based hydroponics system integrated with sensors and a mobile application for remote monitoring and control. The objectives were to design a hardware setup capable of monitoring temperature, TDS, and EC levels in water, automate water flow control, and provide real-time data visualization and notifications through a mobile app. The hardware comprises temperature, and TDS sensors placed in water bucket which is connected to a water motor controlling water flow to cups of plant in the hydroponic setup. A water level sensor ensures optimal water levels. Data from the sensors are transmitted to a cloud database via Wi-Fi. The mobile application enables users to access setup instructions, view real-time temperature, TDS, and EC values, and receive notifications if thresholds are exceeded. Admins have additional privileges to manage user accounts. In the findings, the hardware successfully measures and controls environmental parameters, ensuring optimal growing conditions for the plants. The mobile application provides intuitive user interface and effective data visualization, enhancing user experience and facilitating remote monitoring. Threshold notifications enable timely intervention, contributing to improved plant health and yield.

Keywords: Wick-based hydroponics system, Sensors, Mobile application, Remote monitoring, Data Visualization, Notification.

LIST OF ABBREVIATIONS

Abbreviation	**Description**
AI	Artificial network
CM	Cotton cloth material
CNW	Cotton bonded non-woven wick
CWIS	Capillary wick irrigation system
DLCNN	Deep Learning Convolutional Neural Network
EC	Electrical conductivity
GBM	Gunny bad material
IDE	Integrated development environment
IOT	Internet of things
KNN	K-Nearest Neighbors
LMIC	Low- and Middle-Income Countries
NFT	Nutrient film technique
NSC	Nutrient Solution Concentrations
TDS	Total Dissolved Solids
WSN	Wireless Sensor network

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

1.1 Introduction

Agriculture is an important sector of the Indian economy as it contributes about 17% to the total GDP and provides employment to over 60% of the population. However, despite its significance, the agricultural sector faces formidable challenges that hinder the prosperity of farmers. Issues such as unpredictable weather patterns, limited knowledge regarding crop planning, insufficient access to modern farming techniques, and fluctuating market prices pose significant hurdles for the agricultural community. In recent years, modern vertical farming has emerged as a transformative solution to the challenges faced by traditional agriculture in India. With a rapidly growing population and limited arable land, the need for sustainable and efficient food production methods has become paramount. Vertical farming can produce two to three times as many crops as conventional farming in a much smaller area.

Hydroponics is a cultivation method that uses only water mixed with nutrient solution to grow plants without using soil. This method helps farmers to grow crops efficiently and more productively using less water, labor, and time. Soil is not required for the plant to grow but it acts as a base to provide all the nutrients for the growth of the plant. By providing these nutrients through water externally with external support to hold the plant, water can take over the soil. In hydroponics, we externally provide nutrients that plants require to grow and develop. Nutrients required by the plant are mixed within water thus directly providing the nutrients to plant roots.

All plants will get an equal share of nutrients which leads all plants to grow equally all the time. It has a better nutrient value compared to the soil because of no waste of nutrients and water in soil-less cultivation methods. Hydroponics can be used to cultivate various types of plants like leafy crops, lettuce, and fruit-bearing plants like tomatoes, cucumbers, herbs, peppers, etc. The hydroponic cultivation method requires to have controlled temperature and humidity surroundings, and TDS and EC values of the nutrient solution are to be well maintained in the required range. Many commercial hydroponic farmers use the controlled greenhouse environment to get the highest quality product on a year-round basis.

1.2 Scope

Our project seeks to revolutionize traditional farming practices through the transformative potential of hydroponics. Hydroponic farming, as a soil-less cultivation method, transcends geographical constraints, allowing for adaptable and efficient crop production. Soil is not required for the plant to grow but it acts as a base to provide all the nutrients for the growth of the plant. The wick system, known for its simplicity and efficiency, becomes a cornerstone in our endeavor to create a Smart Hydroponics System that not only facilitates year-round crop production but also caters to a diverse range of crops. Resource efficiency is a cornerstone of our vision, where we aim to optimize water consumption, reduce reliance on arable land, and maximize nutrient use efficiency within the hydroponic environment. This approach seeks to enable not just year-round crop cultivation but to tailor the hydroponic environment to the unique requirements of a diverse array of crops. Moreover, the project delves into the diversity of crops that can thrive in hydroponic systems, breaking away from the constraints imposed by traditional agriculture. The integration of cutting-edge technologies, such as water level, Temperature and TDS sensors, coupled with advanced data analytics and automation, forms a crucial aspect of our comprehensive scope. This integration not only enhances the precision of the wick system but transforms our hydroponic endeavor into an intelligent, adaptable, and sustainable solution. This blend of the wick system and hydroponics is set to reshape the world of precision farming, paving the way for a future where farming isn't just smart but also mindful of the environment and resilient in the face of challenges.

1.3 Objective

Our objective is to build a hydroponics wick system for herbs that can be controlled using a mobile app. First, we focus on integrating water level, TDS, and Temperature sensors to provide real-time insights into the hydroponic environment. Next, data analysis techniques help distill meaningful patterns and trends from the collected data. The implementation of a responsive notification system ensures timely alerts for any deviations from optimal conditions. Automation takes center stage as we integrate sensor data to regulate water circulation and nutrient distribution, optimizing resource usage. This heralds a transition from a manual to a productive resource management system. Finally, an intuitive app is developed for user-friendly control and monitoring, making system maintenance accessible whether on-site or remotely. Together, these objectives lay the foundation for an intelligent, efficient, and sustainable hydroponic farming system.

1.4 Organization of the project work

1.4.1 Completion timeline

The completion timeline shown in Table 1.1 shows the step-by-step phases and timeline of the project.

Table 1.1 Completion timeline

No.	Start	Finish	Activity
1	01-Feb-24	24-Feb-24	Creating the mobile app
2	26-Feb-24	09-Mar-24	Setup of hardware
3	11-Mar-24	16-Mar-24	Analysis of data from sensors and testing
4	18-Mar-24	26-Mar-24	Creating environment for mobile and hardware communication
5	28-Mar-24	03-Apr-24	Connection of mobile app and hardware
6	05-Apr-24	11-Apr-24	Testing
7	12-Apr-24	17-Apr-24	Setting up the wick system and hardware
8	18-Apr-24	19-Apr-24	Setting up Notification module
9	20-Apr-24	22-Apr-24	System Testing

1.4.2 Outline of the Chapters

1. **Chapter 1 Introduction:** This chapter gives a brief introduction to the project Lite Harvest. The introduction contains the subsections company profile, scope, objectives of the project, organization of the project which contains completion timeline, and outline of the chapters.
2. **Chapter 2 Literature Survey:** This chapter provides an explanation about the literature survey that was carried out regarding the project. The chapter provides a detailed analysis regarding the methodology utilized in each paper. The individual paper analysis is followed by a tabular comparative analysis of the performance metrics of each paper.
3. **Chapter 3 System Analysis:** This chapter contains existing system, and their drawbacks. It also provides a brief explanation regarding the proposed system and analyses its feasibility.
4. **Chapter 4 System Specification:** This chapter consists of the functional and non-functional requirements of this project. It also includes the hardware requirements used in the proposed system and software requirements that contain the software, IDEs, and tools used to implement the system.
5. **Chapter 5 Project Description:** This chapter provides a description about the project including the system architecture, data-flow diagram and module description.

6. **Chapter 6 System Implementation:** This chapter provides an explanation about the implementation of the Lite Harvest Project. And snapshots of its working.
7. **Chapter 7 System Testing:** This chapter specifies the test cases that were used to determine the working efficiency of the project.
8. **Chapter 8 Conclusions and Future Enhancements:** This chapter provides concluding remarks.

Chapter 2

Literature Survey

2.1 Survey 1

Title: Development of Automated Monitoring System for Hydroponics Vertical Farming by G W Michael, F S Tay, and Y L Then.

Description: This study employed a comprehensive methodology. The initial focus was on designing an architecture for the automated monitoring system, ensuring its scalability, real-time monitoring capabilities, and seamless integration with hydroponics systems. Careful sensor selection and integration protocols were implemented to capture key variables like nutrient levels, pH, and environmental conditions. The study incorporated automation mechanisms, including actuators and controllers, to optimize farming processes through intelligent responses to sensor data. Communication protocols facilitated effective data transmission, while data storage and analysis provided meaningful insights for crop optimization. A user-friendly interface was developed for accessibility, and rigorous testing, validation, and real-world deployment were conducted to assess the system's functionality, reliability, and practical applicability in hydroponics vertical farming setups.

2.2 Survey 2

Title: Design and Implementation of Smart Hydroponics Farming Using IoT-Based AI Controller with Mobile Application System by S. V. S. Ramakrishnam Raju , Bhasker Dappuri , P. Ravi Kiran Varma , Murali Yachamaneni , D. Marlene Grace Verghese , and Manoj Kumar Mishra.

Description: The study commenced with the design and architectural planning of a smart hydroponics farming system integrated with Internet of Things (IoT) and artificial intelligence (AI) technologies. The team focused on selecting appropriate sensors for monitoring crucial parameters in hydroponics, and an AI controller was implemented to intelligently manage and optimize the farming environment. The integration of a mobile application system allowed for remote monitoring and control. The researchers conducted extensive testing and validation to ensure the efficiency and reliability of the system, considering factors such as data accuracy, responsiveness of the AI controller, and the user interface's functionality. Real-world implementation and user feedback were crucial components of the methodology, validating the practical applicability of their smart hydroponics farming system in agricultural settings.

2.3 Survey 3

Title: IoT-Based Mobile Application for Monitoring of Hydroponic Vertical Farming by Gaganjot Kaur ,Prashant Upadhyaya , and Paras Chawla.

Description: In this study a meticulous methodology was employed to develop and implement an Internet of Things (IoT)-based mobile application for the monitoring of hydroponic vertical farming. The research initiated with the conceptualization of the mobile application system and the identification of key parameters crucial for hydroponic farming. The team integrated IoT technologies, selecting and deploying appropriate sensors to capture real-time data on factors such as nutrient levels, pH, and environmental conditions. The mobile application was designed to provide farmers with a user-friendly interface for remote monitoring and control of the hydroponic system. Extensive testing and validation procedures were implemented to ensure the accuracy and reliability of the data collected and the functionality of the mobile application. The researchers also considered factors like scalability, responsiveness, and the overall user experience, contributing to a comprehensive methodology that aimed at enhancing the efficiency and accessibility of hydroponic vertical farming through IoT-based mobile technology.

2.4 Survey 4

Title: Hydroponic Nutrient Control System based on Internet of Things and K-Nearest Neighbors by Demi Adidrana, Nico Surantha.

Description: In this research, a robust methodology was implemented to develop and assess the efficacy of the proposed system. The study began with the design and integration of an Internet of Things (IoT) infrastructure tailored for hydroponic nutrient control. Special attention was given to the selection and deployment of sensors to capture essential data related to nutrient levels in the hydroponic system. The integration of the K-Nearest Neighbors (KNN) algorithm played a pivotal role in the development of an intelligent nutrient control system, leveraging machine learning techniques to optimize nutrient delivery. Following the system's design, a comprehensive testing phase was conducted to evaluate the accuracy and responsiveness of the IoT-based nutrient control system. The KNN algorithm's performance in predicting optimal nutrient levels was rigorously assessed, considering factors such as precision, recall, and overall model accuracy. Real-world implementation in hydroponic setups allowed for practical validation and fine-tuning of the system's parameters. The methodology encompassed user feedback and system adjustments to ensure that the IoT and KNN-based nutrient control system could effectively contribute to enhanced efficiency and productivity in hydroponic farming.

2.5 Survey 5

Title: Wireless sensor networks for agriculture: The state-of-the-art in practice and future challenges by Tamoghna Ojha, Sudip Misra, Narendra Singh Raghuwanshi.

Description: This study employed a comprehensive methodology to assess the current landscape and potential challenges in implementing wireless sensor networks (WSNs) in agriculture. The study have commenced with an extensive literature review, summarizing existing research and applications of WSNs in agricultural practices. The researchers might have employed systematic data collection methods, possibly through surveys, case studies, or experimental deployments, to gather information on the practical use of WSNs in real-world agricultural scenarios. The methodology would likely include an analysis of the collected data to identify trends, challenges, and opportunities in the adoption of WSNs, offering insights into the current state-of-the-art and paving the way for addressing future challenges in the domain.

2.6 Survey 6

Title: AI-Based Crop Recommendations for Intensive Farming using WSN by G. Cema, Dr. E. Kaliappan.

Description: In this study, the methodology employed is to integrate artificial intelligence (AI) with wireless sensor networks (WSN) for optimized crop recommendations in intensive farming. The research have commenced with the deployment of a WSN infrastructure strategically placed across the farming area, equipped with sensors to capture crucial agricultural parameters. Data collected by the WSN would likely include information on soil moisture, temperature, and other environmental factors. The researchers might have utilized machine learning algorithms, possibly employing supervised learning techniques, to train an AI model for crop recommendations based on historical data from the WSN and known successful crop yields. The methodology would also encompass thorough testing and validation of the AI model, assessing its accuracy and reliability in generating personalized crop recommendations. Real-world deployment and farmer feedback would be crucial aspects, validating the practical applicability of the AI-based recommendations in intensive farming scenarios.

2.7 Survey 7

Title: Analysis of Wick Irrigation in Vegetable Planting by Abdul Rahim Junejo, Shoukat Ali Soomro, Komal Jamil Gujjar, Indian Raj N, Jamshed Ali Channa, Jahangeer Dahri, Yasmin Junejo, Salman Ali Qureshi, Muhammad Ibrahim Junejo, Khadija Urooj.

Description: In this a comprehensive methodology was likely employed to assess the effectiveness of wick irrigation in vegetable cultivation. The study have initiated with the selection of an experimental site for vegetable planting, where wick irrigation systems were implemented. The researchers would likely have measured and recorded key parameters such as soil moisture levels, plant growth, and yield in both the wick irrigation and control groups. The methodology would include a detailed analysis of the data collected, possibly using statistical methods to compare the performance of wick irrigation against traditional irrigation methods. Moreover, considerations for factors like water efficiency, cost-effectiveness, and the overall feasibility of wick irrigation in vegetable planting would contribute to a holistic understanding of its applicability and potential benefits.

2.8 Survey 8

Title: Modeling capillary wick irrigation system for greenhouse crop production by Shaheen Javed Roonjho, Rowshon Md Kamal, Abdul Rehman Roonjho.

Description: In this study methodology employed is to develop and analyze a capillary wick irrigation system tailored for greenhouse crop production. The study have commenced with the conceptualization and design of the capillary wick irrigation model, specifying parameters such as wick material, spacing, and irrigation frequency. The researchers would likely have conducted experimental trials in a controlled greenhouse environment, implementing the capillary wick irrigation system alongside traditional irrigation methods. Key metrics, including soil moisture levels, crop growth, and yield, would have been measured and recorded for both the capillary wick irrigation and control groups. The methodology would encompass the development of a mathematical model to simulate and predict the performance of the capillary wick irrigation system, integrating factors such as water uptake by plants and capillary flow within the wick. The analysis of the model's predictions against observed data would provide insights into the effectiveness and efficiency of capillary wick irrigation in greenhouse crop production.

2.9 Survey 9

Title: Effect of Water pH on Yield and Nutritional Status of Greenhouse Cucumber Grown in Recirculating Hydroponics by R. V. Tyson, E. H. Simonne, D. D. Treadwell, M. Davis & J. M. White.

Description: In the implemented a study to analyze the impact of water pH on cucumber cultivation in recirculating hydroponics systems. The research has begun with the establishment of controlled greenhouse conditions, ensuring consistency in environmental variables. A range of water pH levels would likely have been applied to the hydroponic system, and relevant parameters such as cucumber yield, plant growth, and nutritional content would have been measured and recorded. The methodology would likely include the implementation of a randomized experimental design to account for variability, and the researchers have employed statistical analyses to determine the significance of observed differences. Nutritional status could involve the analysis of key elements in cucumber tissues. Additionally, considerations for factors like water uptake, nutrient availability, and potential nutrient imbalances due to varied pH levels would contribute to a comprehensive understanding of the interplay between water pH and cucumber performance in recirculating hydroponics. The methodology also includes a discussion of potential limitations and practical implications, providing a robust foundation for concluding the influence of water pH on cucumber yield and nutritional status in the specific hydroponic context.

2.10 Survey 10

Title: Effect of Nutrient Solution Concentration on the Growth of hydroponics Sweet potato by Masaru Sakamoto and Takahiro Suzuki.

Description: This study investigates the impact of varying nutrient solution concentrations on the growth of sweet potatoes in a hydroponic system. The study initiated the selection of a controlled experimental environment conducive to sweet potato cultivation in hydroponics. Different nutrient solution concentrations would have been applied to distinct groups of sweet potato plants, encompassing a range of concentrations to observe potential dose-response relationships. The researchers monitored key growth parameters such as plant height, leaf development, tuber formation, and overall plant health throughout the experiment. It includes details on the specific nutrient formulations used, irrigation schedules, and environmental conditions maintained to ensure consistency. Statistical analyses, such as analysis of variance or regression analysis, have been applied to discern significant differences in growth metrics among the different nutrient solution concentrations. Additionally, considerations for factors like nutrient uptake efficiency and potential nutrient-related stress on the plants would contribute to a comprehensive understanding of the optimal nutrient solution concentration for hydroponic sweet potato cultivation. The methodology conclude with a discussion of practical implications and potential applications in hydroponic agriculture.

2.11 Comparative Analysis

Table 2.1: Comparative Analysis

Reference	Algorithm/ Technique	Platform Used	Performance Metrics/ Based On	Advantages	Disadvantages
[1]	SolidWorks-designed three-level hardware setup	SolidWorks	Efficiency	Nutrient management, electrical efficiency, drainage.	Slow plant growth, daily maintenance, sensitivity.
[2]	DLCNN	Raspberry Pi controller	Accuracy	Enhanced nutrient prediction, user-friendly interface.	Initial setup costs, stable internet connectivity, continuous monitoring.
[3]	Tashi Home Pind fresh system	Arduino, Raspberry Pi, and a mobile application for monitoring.	Real-time monitoring	Live data display and alerts, and the automated system.	Does not incorporate farming assistance features, support for vertical farming operations.
[4]	KNN	Data measurement in NFT, and	Accuracy	Real-time monitoring, and control the	Accuracy can be improved for more precise

		the KNN algorithm operates within IoT.		nutrition controller simultaneously.	classification of nutrient conditions.
[5]	WSN	Terrestrial WSN and Wireless Underground Sensor Networks	Efficiency	Resource Optimization, enhances crop quality, and enables prediction and mitigation of adverse environmental effects.	Cost reduction, scalability in LMICs, faults in WSN, requirement for simplified user interfaces.
[6]	AI and WSN	AI algorithms and WSN for data collection and analysis.	Agriculture metrics	Improved efficiency in crop selection, optimized resource utilization, increased crop yield.	Need for reliable and continuous connectivity for WSN, and the accuracy depends on quality and quantity of input data collected.
[7]	Wick irrigation	Wick irrigation setup with modified plastic bottle.	Efficiency	Significant water savings and increased crop yields, efficient.	Lack of generalizability, Variations in water-saving effects.
[8]	GBM, CM, and CNW	CWIS	Wick discharge	Low-cost, efficiency in saving absorbent materials, and enhancement of plant growth.	Discharge of the selected wick material higher in buried wick compared to hanging wick configurations.
[9]	Aquaponic systems	Recirculating aquaponic systems.	Responsivenes s	Enhanced nitrification, higher fish	Higher pH levels affect early marketable

				stocking and nutrient loads, improved ammonia biofiltration.	cucumber yield. Visual deficiency symptoms by foliar nutritional sprays.
[10]	NSC	Hydroponic sweet potato cultivation.	Root Yield	Resource efficiency, precise control over nutrient components, flexible, and enhances plant growth.	Cost-effectiveness and sustainability.

2.12 Summary of Literature Survey

The studies outlined a diverse range of methodologies and applications in the field of hydroponics farming, with a common emphasis on leveraging technology for enhanced monitoring, control, and optimization. One notable endeavor focused on the development of an automated monitoring system tailored for hydroponics vertical farming. This involved meticulous architecture design, sensor selection, and integration of automation mechanisms to ensure real-time monitoring capabilities and seamless integration with hydroponic setups. Through rigorous testing and real-world deployment, the system's functionality and practical applicability were thoroughly validated, highlighting its potential to revolutionize farming practices.

Another significant study delved into the integration of Internet of Things (IoT) and artificial intelligence (AI) technologies into hydroponics farming. By meticulously selecting sensors and implementing an AI controller, the researchers aimed to intelligently manage and optimize the farming environment. The integration of a mobile application system allowed for remote monitoring and control, further enhancing accessibility and user experience. Extensive testing, validation, and user feedback were pivotal in affirming the efficiency and reliability of the system, underscoring its potential to drive innovation in agricultural practices.

Furthermore, research efforts were directed towards the development of IoT-based mobile applications tailored specifically for monitoring hydroponic vertical farming. Through careful selection and deployment of sensors, real-time data on crucial parameters such as nutrient levels and environmental conditions were captured and made accessible to farmers via user-friendly interfaces. The studies prioritized scalability, responsiveness, and overall user experience, reflecting a holistic approach towards enhancing efficiency and accessibility in hydroponic farming through IoT-based technologies.

Moreover, a focus on nutrient control systems leveraging IoT infrastructure and machine learning techniques underscored the importance of precision agriculture in hydroponics. By integrating algorithms like K-Nearest Neighbors (KNN), researchers aimed to optimize nutrient delivery and enhance crop productivity. Rigorous testing, validation, and real-world implementation validated the efficacy of these systems, offering insights into their potential to revolutionize nutrient management practices in hydroponic setups.

Additionally, studies exploring the effectiveness of irrigation methods such as wick irrigation and capillary wick systems provided valuable insights into their applicability and potential benefits in hydroponic farming. Through controlled experiments, data analysis, and comparison with traditional methods, researchers aimed to elucidate the impact of these techniques on crop growth, yield, and overall efficiency, contributing to a comprehensive understanding of irrigation practices in hydroponics.

Chapter 3

System Analysis

3.1 Existing System

The existing wick-based hydroponics system lacks advanced monitoring and control features, making it challenging for growers to maintain optimal growing conditions. Typically, these systems rely on manual monitoring, where growers must regularly check parameters such as temperature, nutrient levels, and water levels, leading to inefficiencies and potential inconsistencies in plant growth. Without automated sensors, growers may struggle to accurately assess the environmental conditions within the hydroponic setup. This can result in suboptimal nutrient management, leading to nutrient imbalances that may impact plant health and yield. Additionally, manual monitoring makes it difficult for growers to detect and respond promptly to fluctuations in temperature or nutrient levels, increasing the risk of plant stress or damage. Furthermore, the lack of a centralized monitoring and control system limits growers' ability to remotely manage the hydroponic setup. Without access to real-time data and notifications, growers may miss critical changes in the growing environment, leading to missed opportunities for intervention and optimization. Overall, the existing wick-based hydroponics systems suffer from limitations in monitoring, control, and management capabilities, hindering growers' ability to maximize plant growth and yield effectively.

3.1.1 Drawbacks

- **Scalability:** The existing system may have limitations in scaling up to accommodate larger hydroponic setups or expanding operations.
- **Technical Dependency:** The system relies heavily on sensor technology and automation, making it vulnerable to technical malfunctions or failures.
- **Data Security:** As the system collects and processes sensitive data, such as plant health and environmental conditions, ensuring data security and protection against cyber threats is essential but may pose challenges.
- **Maintenance:** Regular maintenance and calibration of sensors and automation components are required to ensure the system's reliability and accuracy, which can be time-consuming and labor-intensive.

- **User Interface:** While the App-Based Maintenance Module offers a user-friendly interface, improvements may be needed to enhance usability and accessibility for a broader range of users, including those with limited technical expertise.
- **Reliability:** The system's reliance on technology and automation introduces the risk of system failures or malfunctions, which could disrupt plant growth and crop production if not promptly addressed.

3.2 Proposed System

The proposed system aims to enhance the existing hydroponic setup by integrating automated sensors for monitoring temperature, TDS (Total Dissolved Solids), and water levels in the nutrient solution. These sensors provide real-time data on environmental conditions, enabling growers to maintain optimal growing conditions more effectively. In addition to sensor integration, the proposed system includes a mobile application that enables remote monitoring and control of the hydroponic system. Through the app, users can access real-time data on temperature, TDS, and water levels, as well as receive notifications if any parameters deviate from preset thresholds. This mobile interface enhances accessibility and allows growers to monitor and manage their crops from anywhere, at any time. Furthermore, the proposed system incorporates an automated irrigation system that controls the flow of nutrient-rich water to the plants based on sensor readings. This automation reduces the need for manual intervention and ensures consistent and precise delivery of nutrients to the plants, optimizing growth and yield. Another key feature of the proposed system is the integration of a data logging and analysis module. This module collects and stores sensor data over time, allowing growers to track trends, identify patterns, and make data-driven decisions about nutrient management and environmental control. By analyzing historical data, growers can gain valuable insights into the performance of their hydroponic system and optimize its operation for maximum productivity. Overall, the proposed system represents a comprehensive solution for hydroponic farming, combining automated monitoring, remote control, and data analysis capabilities to enable growers to achieve consistent and high-yield crop production with minimal effort.

3.3 Feasibility Study

A feasibility study is necessary to determine whether the proposed system for Liteharvest with IoT is feasible in terms of technical, economic, and operational aspects.

3.3.1 Technical Feasibility

- **Temperature Sensor**: This sensor measures the temperature of the hydroponic solution to ensure it remains within the optimal range for plant growth, preventing stress and potential damage to the plants.
- **TDS (Total Dissolved Solids) Sensor:** The TDS sensor measures the concentration of dissolved solids in the hydroponic solution, providing an indication of nutrient levels. It helps growers ensure that the nutrient solution is properly balanced to meet the plants' needs.
- **Water Level Sensor:** This sensor detects the level of water in the reservoir or growing containers, ensuring a sufficient water supply for the plants while preventing overfilling or low water levels.

3.3.2 Operational Feasibility

Operational feasibility for our project involves assessing the ease of integration into existing hydroponic setups, the user-friendliness of the interface, and the system's efficiency in monitoring and controlling environmental conditions. Additionally, compatibility with existing hardware and software components is crucial for smooth integration and minimal disruption to operations, ensuring effective utilization by hydroponic growers.

Seamless integration: The system seamlessly integrates with existing hydroponic setups and processes, ensuring compatibility and minimal disruption to operations.

Compatibility: It is compatible with commonly used hardware and software components in hydroponic farming, facilitating smooth data exchange and communication.

User-friendly interface: With intuitive controls and graphical interfaces, the system is easy to operate and manage, even for users with limited technical knowledge.

Efficient monitoring and control: Through sensors and automation technology, the system enables efficient monitoring and control of environmental parameters, ensuring optimal growing conditions.

Minimal training required: Clear instructions and intuitive interfaces require minimal training for effective use, facilitating smooth adoption of the technology by users.

3.3.3 Economical Feasibility

Economic feasibility evaluates the financial viability and potential benefits of implementing the proposed system.

- **Cost-effectiveness:** The system's initial investment is offset by long-term savings in labor costs and resource utilization due to automation and efficient monitoring.
- **Return on investment (ROI):** The projected ROI from increased crop yields and reduced resource wastage justifies the initial investment in the system.
- **Scalability:** The system is designed to be scalable, allowing for gradual expansion as needed without significant additional costs.
- **Market competitiveness:** By improving crop yields and reducing operational costs, the system enhances the competitiveness of hydroponic farming operations in the market.
- **Cost-benefit analysis:** A thorough cost-benefit analysis demonstrates that the benefits of implementing the system outweigh the associated costs over its lifespan.

Chapter 4

System Specification

4.1 Hardware Requirements

- Water level Sensor
- EC Sensor
- TDS Sensor
- Temperature Sensor
- Water Pump
- L293D Motor Driver Module
- Arduino Uno
- Power Supply
- LCD Display
- I2C Module
- ESP8266

4.2 Software Requirements

- Arduino IDE
- Arduino Cloud
- Android Platform
- Flutter
- Firebase

4.3 Functional Requirements

- The system is capable of reading data from water sensors, including information related to water level.
- The system facilitates the transmission of collected data to Firebase Cloud for further analysis.
- The cloud-based system is able to analyze the received data, presumably for water level, TDS, and EC.
- The system includes a mechanism to monitor water levels continuously.
- It compares the water level against a predefined threshold value.

- If the water level falls below the specified threshold, the system generates an alert to notify relevant parties.
- The system collects data from TDS and Temperature sensors.
- The system analyzes the TDS and EC values obtained from the sensors.
- If TDS or EC values are below optimal levels, the system generates an alert to indicate potential issues.
- The system presents the analyzed data, particularly water levels, Temperature TDS, and EC, in a graphical format for easy interpretation.
- The system has a mobile application for maintenance purposes.
- The application facilitates manual adjustments or interventions based on the analyzed data.

4.4 Non-Functional Requirements

Non-functional requirements define the desired qualities of the system to be developed and often influence the system architecture more than functional requirements do. A non-functional requirement corresponds to a set of restrictions imposed on the system to be developed, establishing, how attractive, useful, fast, or reliable it is. This category of requirement is a set of required overall attributes of the system, including portability, reliability, efficiency, human engineering, testability, understand ability, and modifiability.

The Non-Functional requirement of the project is to meet the accuracy and execution speed expected by the users. It is built to be reliable and does not consist of any sensitive information of a user, thus making it more secure. The project is to be compatible with different machine learning models. As the user interface for the project will be a web application it is built to be user friendly, and the operations can be well understood by the user.

The non-functional requirements for the proposed system are as follows:

Performance: Real-time monitoring and analysis with rapid alert response.

Reliability: Consistent, accurate data with minimal false alerts.

Scalability: Accommodate a growing number of sensors/devices.

Usability: User-friendly graphical display and intuitive mobile app.

Security: Encrypted data transmission, and access restrictions.

Availability: High uptime, and redundancy for continuous operation.

Maintainability: Easy updates, and configuration changes.

Compatibility: Works with various water sensors, and common mobile devices.

Data Storage: Secure, efficient storage for historical data.

Interoperability: Integration capability with other IoT devices/systems.

Chapter 5

Project Description

5.1 Problem Definition

Hydroponics, as an innovative method of soilless farming, offers numerous advantages such as efficient use of space, water, and nutrients, along with the potential for higher yields. However, one of the challenges faced by hydroponic farmers is the need for continuous monitoring and management of environmental parameters such as temperature, Total Dissolved Solids (TDS), Electrical Conductivity (EC), and water levels. Fluctuations in these variables can significantly impact plant health and productivity. Traditional hydroponic systems often lack automated monitoring and control mechanisms, requiring farmers to manually oversee and adjust environmental conditions. This manual approach is labor-intensive, time-consuming, and prone to human error, leading to suboptimal growing conditions and reduced yields. Additionally, the lack of real-time data visualization and notifications further complicates the management process, hindering timely intervention in case of adverse conditions.

To address these challenges, there is a need for an integrated solution that combines hardware sensors, automated control mechanisms, and a user-friendly mobile application for remote monitoring and management of hydroponic systems. Such a system would enable farmers to monitor key environmental parameters in real-time, automate water flow control based on sensor readings, and receive timely notifications in case of deviations from optimal conditions. Furthermore, administrators should have the ability to manage user accounts and access system data for analysis and decision-making purposes.

5.2 Overview of the Project

The project focuses on the development of an integrated wick-based hydroponics system equipped with sensors and a mobile application for remote monitoring and control. This comprehensive solution aims to address the challenges associated with manual management of hydroponic systems by providing automated monitoring, control, and data visualization capabilities.

Hardware Setup

The hardware setup includes temperature, and TDS sensors integrated into the hydroponic system. These sensors continuously monitor key environmental parameters such as temperature and nutrient levels in the water. A water motor controls the flow of water to cups containing plants, ensuring optimal hydration. Additionally, a water level sensor maintains the desired water level in the system, preventing overwatering or underwatering.

Mobile Application

The mobile application serves as the interface for users to remotely monitor and control the hydroponic system. Users can access setup instructions, view real-time data on temperature, TDS, and EC levels, and receive notifications if any parameter exceeds predefined thresholds. The application provides intuitive data visualization tools, allowing users to track trends and make informed decisions about system management.

Cloud Database Integration

Data collected from the sensors are transmitted to a cloud database via Wi-Fi connectivity. This cloud database serves as a centralized repository for storing and analysing sensor data. Administrators can access the database to manage user accounts and perform data analysis for system optimization and decision-making purposes.

5.3 System Architecture

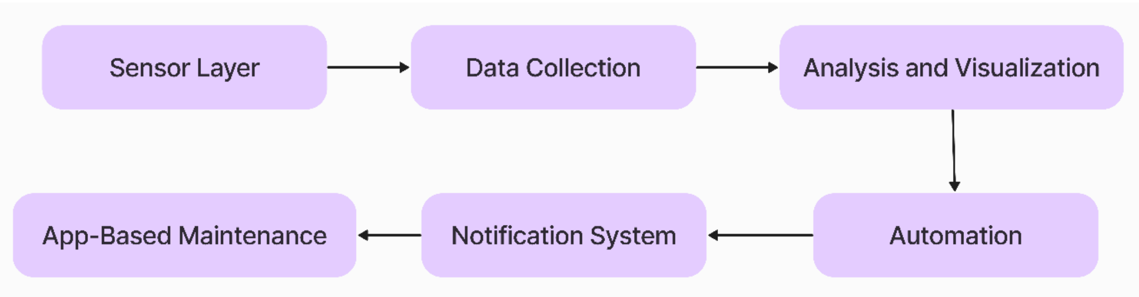


Fig. 5.1: System Architecture

The system architecture in Fig. 5.1 contains of the following:

Sensor Layer: The initial phase involves deploying sensors to capture data. These sensors could be diverse, measuring variables like temperature, humidity, pressure, etc.

Data Collection: Once deployed, the sensors gather data, creating a stream of information related to the monitored environment or system.

Analysis and Preprocessing: The collected data undergoes analysis and preprocessing to extract meaningful insights. This stage involve cleaning, filtering, or aggregating the data to make it suitable for further processing.

App-Based Maintenance: An application (app) is used to manage and maintain the system. This could involve monitoring the health of the sensors, checking for anomalies, and ensuring proper functionality.

Notification System: A notification system is implemented to alert relevant parties or users about significant events or issues detected during the analysis phase. This could include warnings about abnormal sensor readings or system malfunctions.

Automation: The system incorporate automation based on the analyzed data and maintenance needs. This could involve adjusting settings, triggering responses, or initiating specific actions to optimize or correct the system based on the information gathered by the sensors.

5.4 Module Description

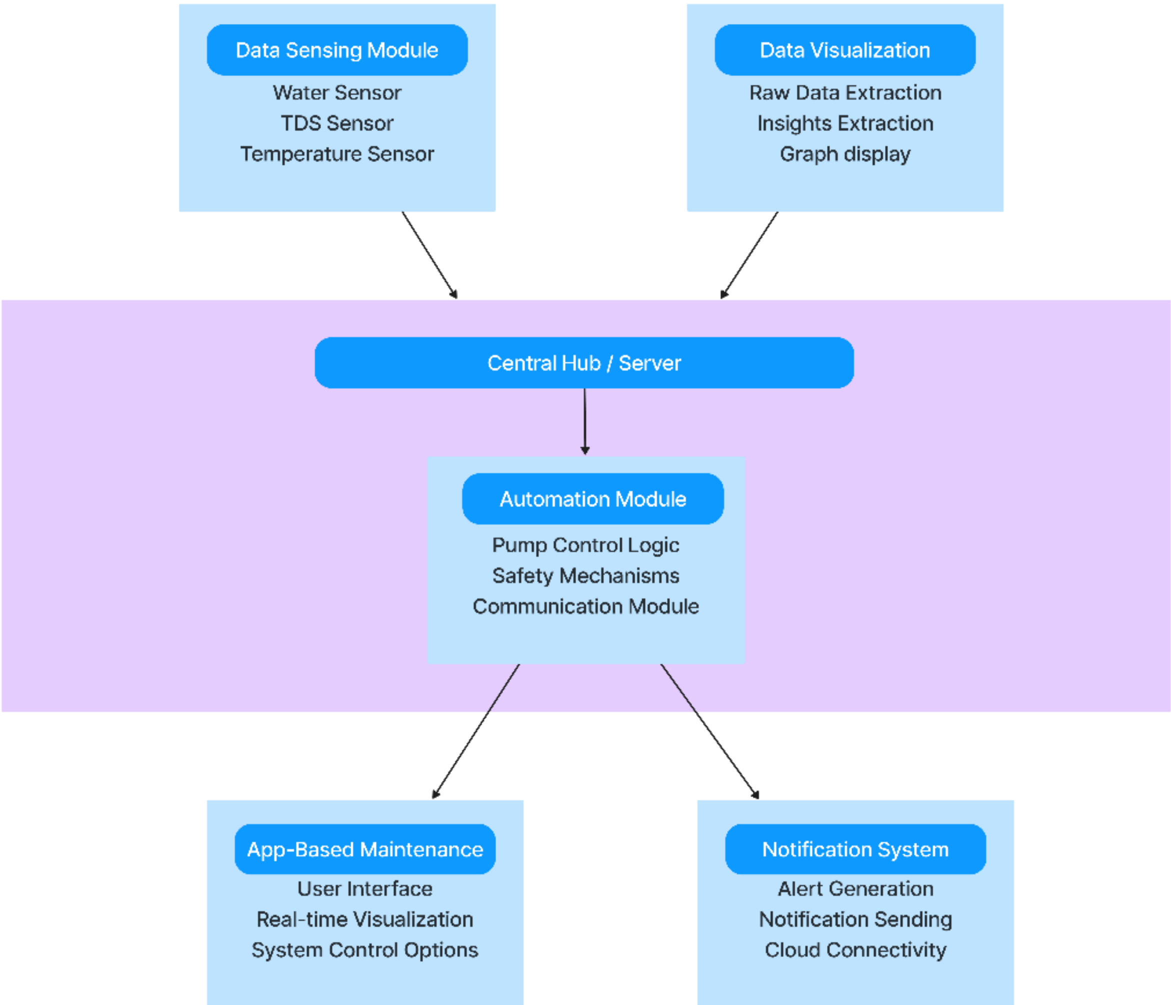


Fig. 5.2: Modules of Lite Harvest

5.4.1 Reading Sensor Data Module

In a wick-based hydroponics system, the Reading Sensor Data Module in Fig. 5.2 plays a pivotal role in maintaining optimal growing conditions by monitoring crucial parameters such as electrical conductivity (EC), Total Dissolved Solids (TDS), and water levels. This module integrates advanced sensors dedicated to each parameter, ensuring precise and real-time data acquisition. The TDS sensor measures the nutrient concentration in the water, allowing for accurate adjustments to the hydroponic solution's composition. Simultaneously, the TDS sensor monitors acidity levels, enabling growers to fine-tune the TDS balance for optimal nutrient absorption by the plants. Additionally, water level sensors provide insights into the reservoir's status, preventing issues like overfilling or low water levels. The comprehensive data gathered by the Sensor Data Module empowers hydroponic cultivators with the information needed to regulate water conditions effectively, fostering a controlled and flourishing environment for plant growth.

5.4.2 Data Visualization Module

The Data Visualization Module in Fig. 5.2 of the wick-based hydroponics system, featuring EC, TDS, and water level sensors, stands as a critical layer of intelligence essential for enhancing plant growth optimization. This module employs sophisticated data visualization techniques, including trend analysis and graphical representation, to derive actionable insights from sensor data. Through trend analysis, the module identifies and visualizes patterns and fluctuations in EC, TDS, and water level readings over time, providing growers with a comprehensive understanding of environmental dynamics. Graphical representation enables intuitive visualization of sensor data, facilitating quick identification of trends, anomalies, and correlations. By leveraging these visualization techniques, the module not only ensures the accuracy and reliability of sensor data but also empowers growers to make informed decisions regarding nutrient adjustments, TDS balancing, and water management. The graphical representation of sensor data enables growers to monitor environmental conditions in real-time and detect any deviations from optimal levels promptly. Additionally, trend analysis facilitates the identification of recurring patterns and correlations between different variables, enhancing growers' understanding of how changes in one parameter affect others. The integration of the Data Visualization Module in the wick-based hydroponics system contributes significantly to its overall efficiency and success. By providing growers with intuitive visualizations of sensor data, the module enables them to maintain an optimal and customized environment for plant cultivation, ultimately leading to improved plant health and yield.

5.4.3 Automation Module

The Automation Module in Fig. 5.2 in a wick-based hydroponics system, featuring EC, TDS, and water level sensors, represents a sophisticated layer of control and efficiency. Central to this module is the pump control logic, which interprets data from the sensors to automate the delivery of nutrient-rich water to the plants. Through meticulous programming, the pump control logic adjusts the nutrient solution's flow rate based on real-time EC and TDS readings, ensuring that the hydroponic system maintains optimal conditions for plant growth. Additionally, the Automation Module incorporates safety mechanisms to prevent potential issues, such as over-flooding or extreme variations in nutrient concentration. These safeguards enhance the reliability and stability of the system, minimizing the risk of damage to plants. Moreover, the module incorporates a communication module that facilitates real-time interaction and feedback. This enables remote monitoring and control, allowing growers to access critical information and make adjustments to the hydroponic system from a distance, thereby optimizing resource management and ensuring the overall health and productivity of the cultivated plants.

5.4.4 Notification System Module

The Notification System Module in Fig. 5.2 in a wick-based hydroponics system, integrated with EC, TDS and water level sensors, plays a pivotal role in keeping growers informed and proactive in managing their cultivation environment. This module is designed to generate alerts and send notifications in real-time based on the data received from the sensors. If irregularities or critical conditions are detected, such as fluctuations in EC or TDS levels, or if the water level deviates from the optimal range, the Notification System Module promptly generates alerts. These notifications can be sent through various channels, such as mobile apps, email, or SMS, ensuring that growers are instantly aware of any issues that require attention. Additionally, the module often incorporates cloud connectivity, allowing growers to access historical data, and trends, and receive notifications remotely. This cloud integration enhances the accessibility and convenience of monitoring the hydroponic system, enabling growers to make informed decisions promptly and proactively address any challenges, ultimately contributing to the efficiency and success of the cultivation process.

5.4.5 App-Based Maintenance Module

The App-Based Maintenance Module in Fig. 5.2 in a wick-based hydroponics system, equipped with EC, TDS, and water level sensors, revolutionizes the user experience by offering a sophisticated interface for monitoring and controlling the cultivation environment.

This module provides a user-friendly interface accessible through a mobile or web application, allowing growers to have real-time visualization of key parameters such as EC, TDS, and water levels. The intuitive design of the user interface simplifies data interpretation and enables growers to make informed decisions about nutrient adjustments and system maintenance. Moreover, the app-based module offers system control options, allowing users to remotely manage and fine-tune parameters to optimize plant growth conditions. Whether adjusting nutrient concentrations, and TDS levels or monitoring water levels, this module grants growers unprecedented control over their hydroponic system, fostering a seamless and efficient cultivation experience. The integration of user-friendly interfaces and real-time visualization capabilities in the App-Based Maintenance Module enhances accessibility, making it a valuable tool for both novice and experienced hydroponic cultivation.

5.5 Data Flow Diagram

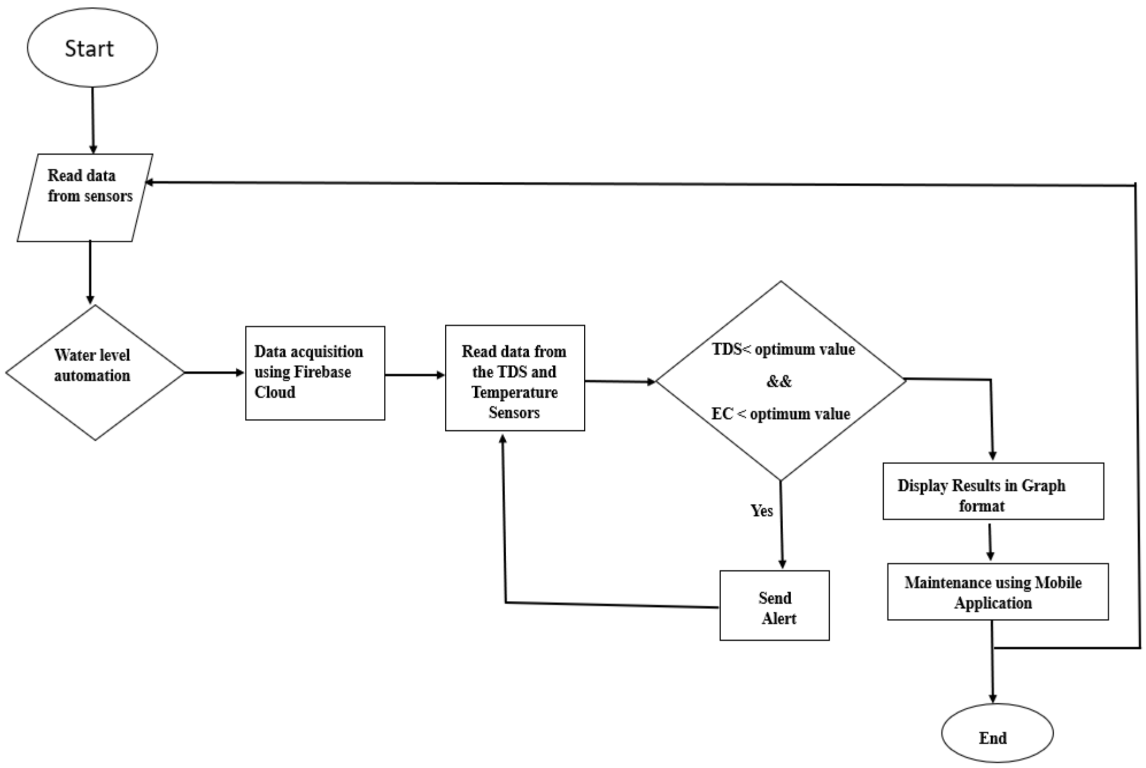


Fig. 5.3 Data Flow Diagram

A water level simulation system is a computer program that can predict and monitor the health of water in various environments. The system data flow is shown in Fig. 5.3.

The system first gathers information through sensors that track the water's level, temperature, and the total dissolved solids (TDS) content. This data is then uploaded to a cloud storage service, often using Firebase Cloud by Google, for easy access and analysis.

The core function of the system lies in water level automation. It continuously monitors the water level data. Additionally, it checks the TDS levels and electrical conductivity (EC) against a predetermined optimal value. If both TDS and EC fall within the acceptable range, signifying good water quality, the system simply displays the collected data in a clear graph format for users to visualize trends.

However, if the TDS or EC readings surpass the optimum value, indicating potential water quality issues, the system takes corrective actions. It triggers an alert notification that's sent directly to a mobile application. This alert informs the user that maintenance is required to restore the water quality to its ideal state.

In essence, this water level simulation system acts as a vigilant guardian for various water bodies. By constantly monitoring TDS and EC, it efficiently detects water quality deterioration and prompts users to take necessary maintenance measures, ensuring the health of the water.

5.6 Sequence Diagram

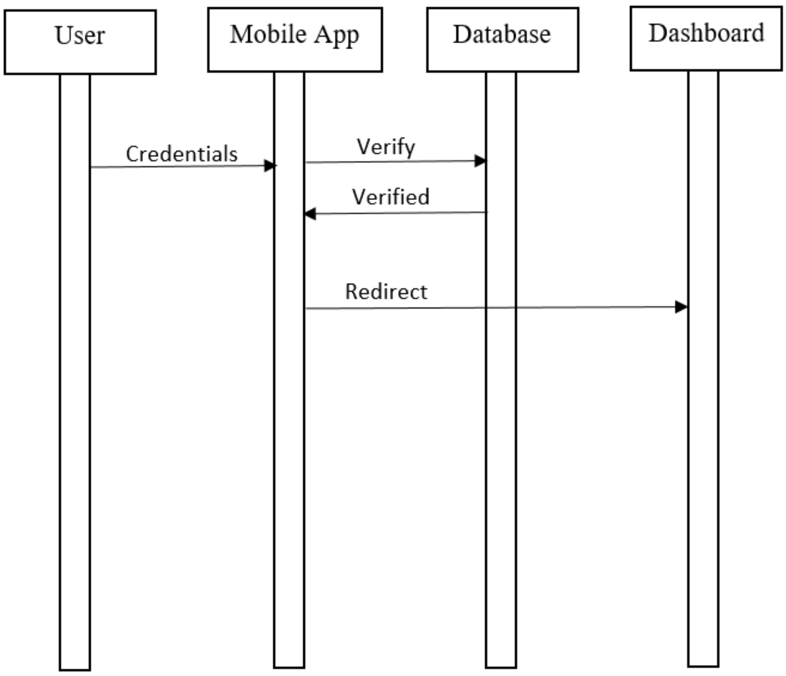


Fig. 5.4 Sequence Diagram for Login

Logging into a mobile app as shown in sequence diagram in Fig. 5.4 typically involves a dance between your device, an app database, and the app itself. You enter your username and password, which the app then sends to the database for verification. If everything checks out, the database sends a green light back to the app, granting you access to the app's main dashboard. This is a streamlined explanation, and real-world apps often have extra security measures like password encryption to keep your information safe.

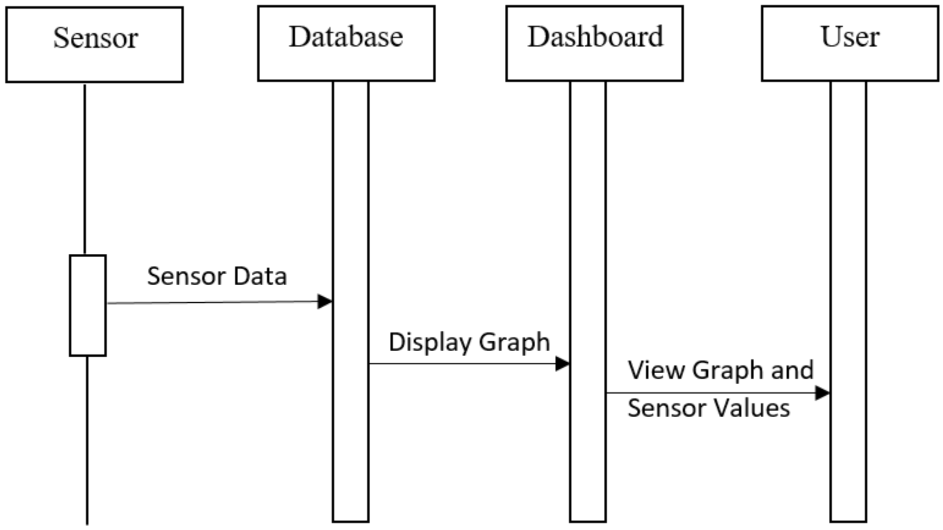


Fig. 5.5 Sequence Diagram for Displaying Graph and Sensor Values

Sensor data monitoring systems keep tabs on the world around us. Sensors, like tiny spies, collect info on temperature, pressure, or anything you can measure. This data is then beamed to a central database for storage. Think of it as a giant filing cabinet for the environment. A user-friendly dashboard acts as a translator, turning this data into charts and graphs. Users can then easily see trends and make informed decisions based on what the sensors are picking up. These systems are like super-powered assistants, helping us understand everything from greenhouse health to building energy use.

Chapter 6

System Implementation

6.1 Introduction

The implementation of the wick-based hydroponics system integrated with sensors and a mobile application marks a significant advancement in agricultural technology. With the hardware setup designed and deployed, and the mobile application fully functional, the project enters the implementation phase. This phase involves putting the system into operation, conducting real-world testing, and ensuring its seamless functionality in a hydroponic farming environment.

The hardware components, including temperature, TDS sensors, water motor, and water level sensor, have been assembled and installed within the hydroponic setup. Each sensor is strategically placed to accurately monitor environmental parameters, while the water motor and level sensor ensure precise water management. The system is powered and connected to WiFi for data transmission to the cloud database.

The data collected by the sensors are transmitted to the cloud database in real-time via WiFi connectivity. The cloud database serves as a centralized repository for storing and analysing sensor data. This integration enables seamless access to data from the mobile application and provides administrators with the necessary tools for user management and data analysis.

The mobile application, designed to provide users with remote monitoring and control capabilities, has been developed and deployed. Users can access setup instructions, view real-time data on temperature, TDS, and EC levels, and receive notifications if thresholds are exceeded. The intuitive user interface and effective data visualization tools enhance the user experience, facilitating remote monitoring and timely intervention.

As the implementation phase progresses, the wick-based hydroponics system with integrated sensors and a mobile application demonstrates its effectiveness in monitoring, controlling, and optimizing hydroponic farming processes. The seamless integration of hardware components, cloud database, and mobile application contributes to improved plant health, productivity, and overall efficiency in hydroponic farming practices.

6.2 Hardware Implementation

The hardware implementation for the proposed project involves several components as shown in Fig. 6.1. The temperature sensor and TDS sensor is responsible to sense the water quality. The data is read from the sensor through Arduino Uno and displayed onto the LCD

Display. The water pump is responsible to send water from the water container to the wick-based hydroponics setup. The water flow is made automatic for the user, where with the help of water level sensor it fills the setup till certain point.

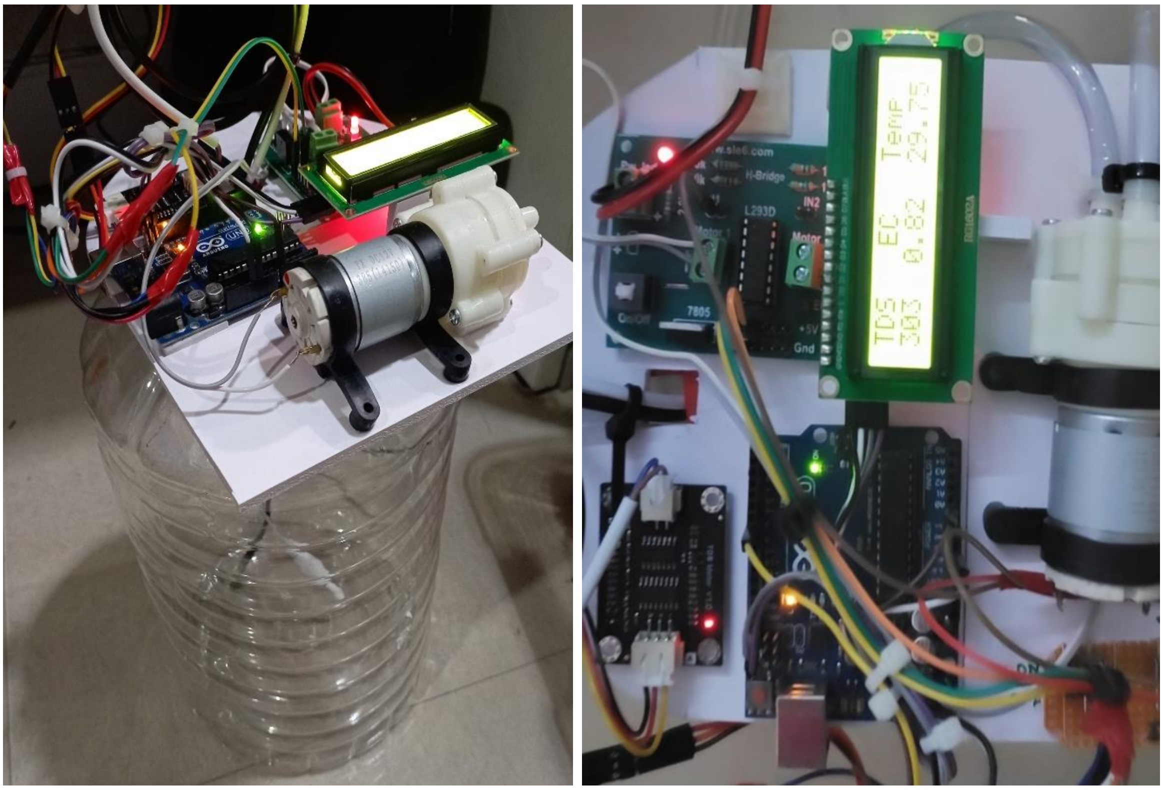


Fig. 6.1 Reading Sensor Data

Fig. 6.2 Wick-Based Hydroponics Setup

The wick-based hydroponics setup as shown in Fig. 6.2 contains small cups which holds the plant. The cups contain small holes in the bottom for the wick material to be placed. The wick material is adjusted inside the cup such that it is able to give sufficient water to the plant. Small clay balls are used to hold the plant in straight position inside the cup.

6.3 Software Implementation

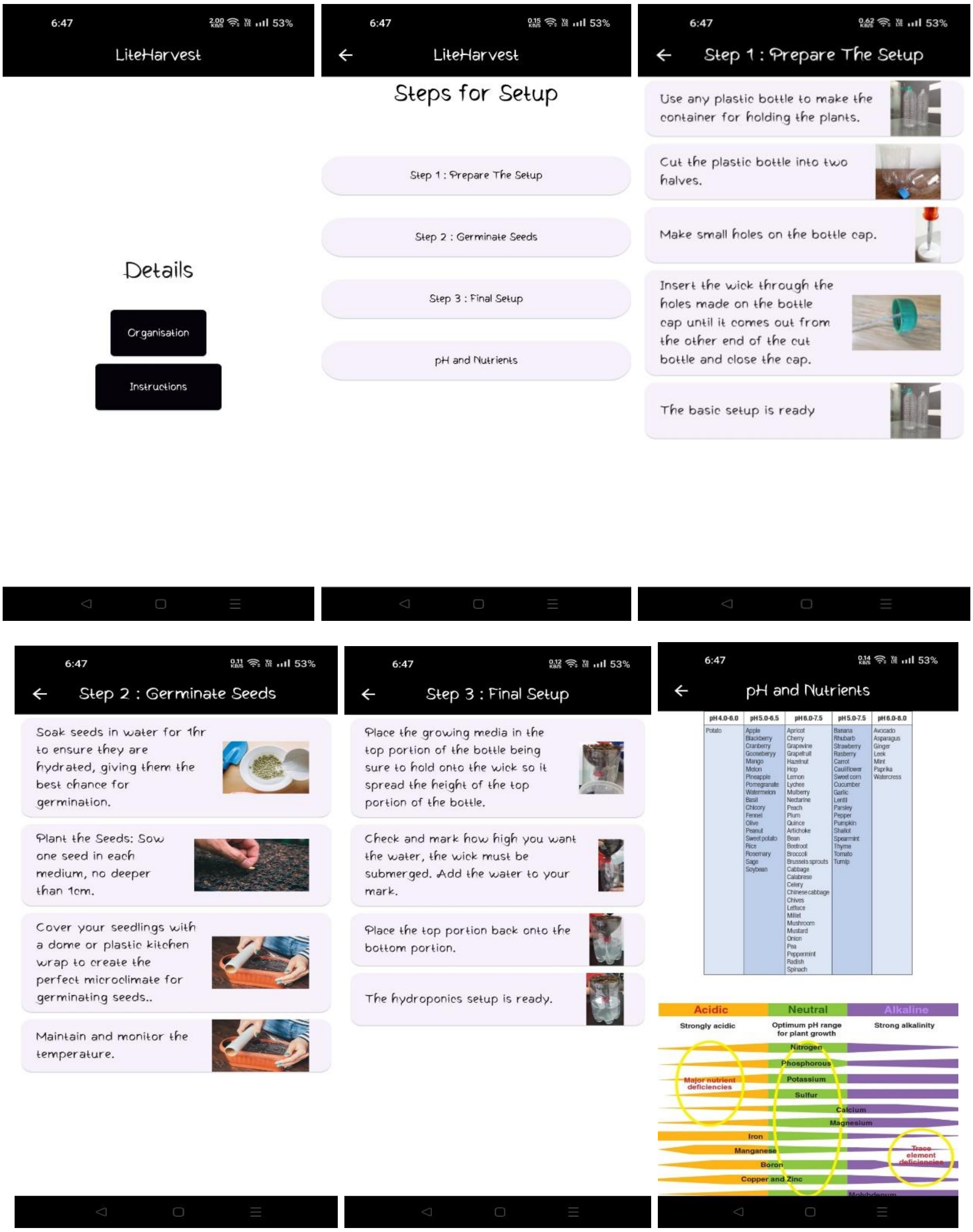


Fig. 6.3 Home Page and User Instructions

The home page of Lite Harvest app as shown in Fig. 6.3 presents the buttons to go to instructions page and and organisation login page. The intructions button allows user to read and apply the steps to build a wick-based hydroponics setup in low cost using recyclable materials. The buttons goes to a page which presents multiple buttons to which instructs on that particular step. The steps are prepares the setup, germinate the seeds, final setup and details of pH and nutrients.

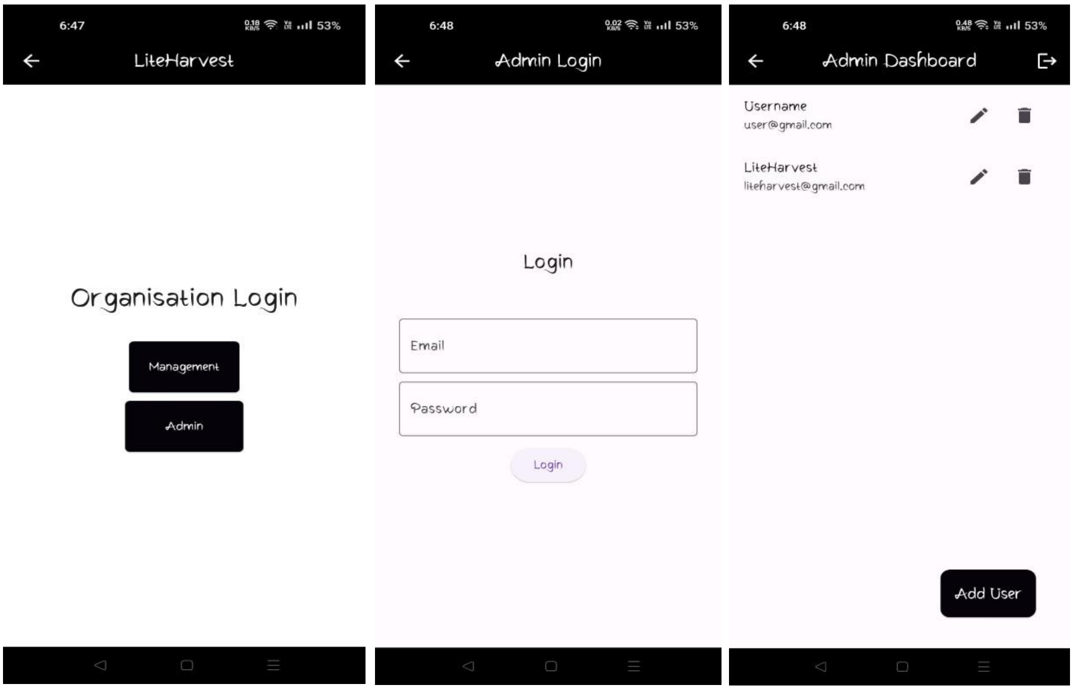
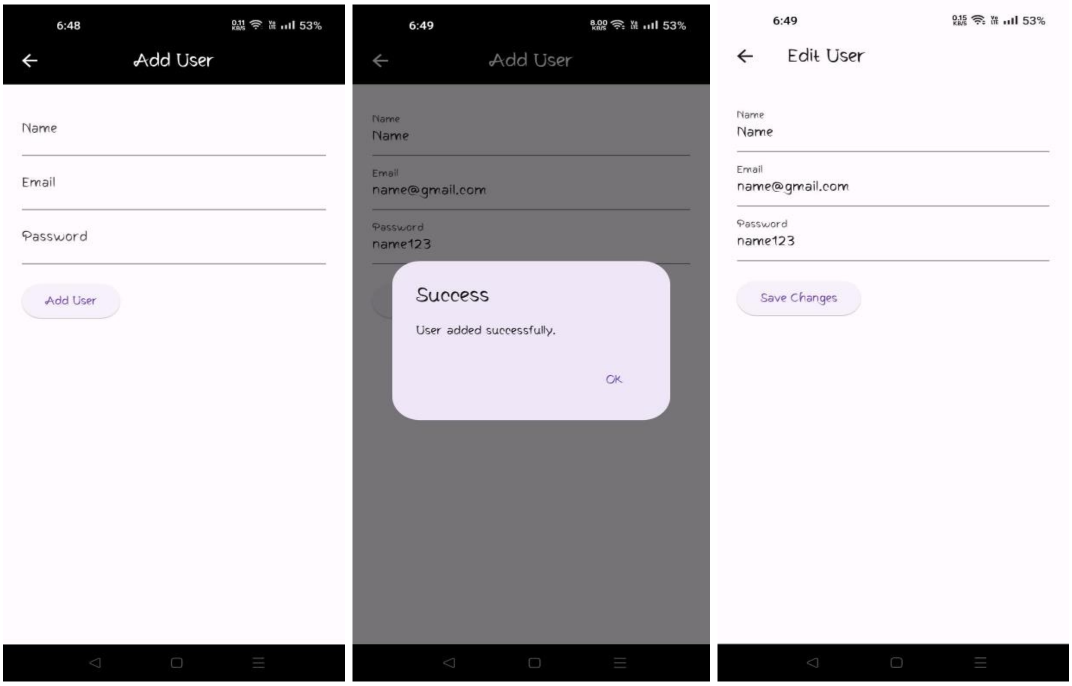


Fig. 6.4 Organisation- Admin Login

The Organisation page as shown in Fig. 6.4 is for organisation usage to maintain and track the plant growth. The organisation button leads to a page which contains two buttons leading to admin login and management login. The admin login allows only the admin to login and manage user details for management.



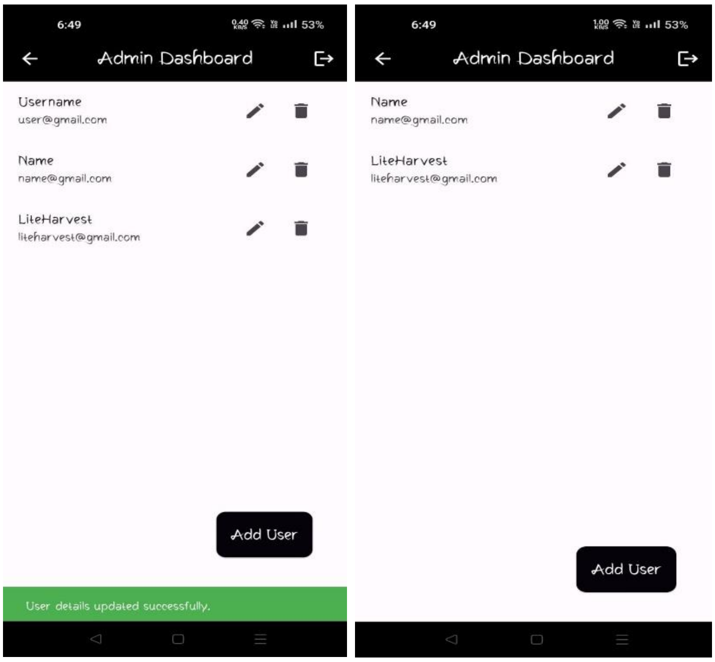


Fig. 6.5 Organisation – Admin

The admin can add user as shown in Fig. 6.5 by clicking the button Add User which leads to a page which allows the admin to add user details containing name, email and password. The admin can change the user details by clicking the editing icon. The user details can be deleted by clicking the delete icon.

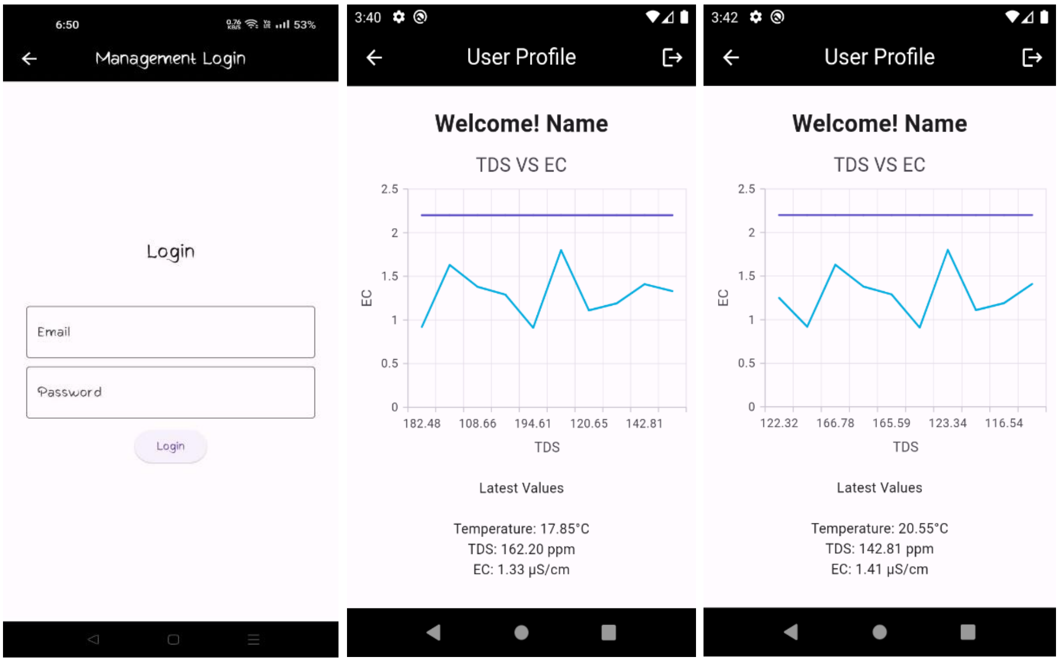


Fig. 6.6 Organisation - Management

Management login as shown in Fig. 6.6 allows the management user to login and view the plant growth. The user profile displays a graph comparison of TDS and EC. The latest Temperature, TDS and EC values are also displayed.

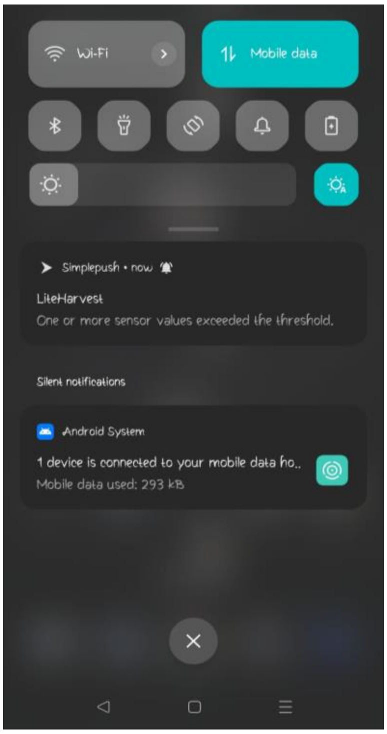


Fig. 6.7 Notification

The notification is sent to user as shown in Fig. 6.7 when the sensor values changes. The notification is sent when threshold is reached for any of the Temperature, TDS, or EC values from esp8266 using an app simple push. The user open the flutter app to view he values and take action accordingly.

Chapter 7

System Testing

System testing is a crucial part of any project to ensure that it meets the desired requirements and functions correctly. For the Lite Harvest project, there are several components that need to be tested to ensure their proper functioning.

7.1 Tests Conducted

7.1.1 Hardware Testing

Hardware testing for the Lite Harvest Project involves testing the individual components and the system to ensure that they are functioning correctly and as intended. The testing process involves the following steps:

Testing the Temperature sensor: Testing a temperature sensor involves validating its accuracy and reliability across a range of temperatures. This typically includes placing the sensor in controlled environments with known temperatures and comparing its readings against calibrated reference instruments.

Testing the TDS sensor: Testing a TDS (Total Dissolved Solids) sensor involves verifying its accuracy and reliability in measuring the concentration of dissolved solids in water. This typically includes calibrating the sensor using standard calibration solutions of known TDS values, conducting validation tests in various water samples to ensure consistent readings, and assessing the sensor's response time and stability over time.

Testing the water level sensor: Testing the water level sensor involves verifying its accuracy and reliability in detecting varying water levels. This typically includes immersing the sensor in water to simulate different levels and observing its response. Additionally, calibrating the sensor to ensure precise readings across the entire range of water levels is essential.

Testing the LED display: Testing the LED display to show sensor values involves ensuring proper connectivity and functionality of the display unit with the microcontroller. This includes verifying the wiring connections and uploading the appropriate code to the microcontroller to fetch and display sensor data. The test procedure typically involves simulating sensor inputs or using actual sensors to validate the accuracy of the displayed values on the LED screen.

7.1.2 Software Testing

Software testing is an important part of the development process to ensure that the software functions as expected and meets the user requirements. In the case of the Lite Harvest system, the software includes the code that runs on the Arduino microcontroller and the application that the user interacts with.
The software testing process typically involves the following steps:

Unit Testing: Unit testing for a Lite Harvest system with a mobile app involves testing individual components and functions to ensure they perform as expected. This includes testing sensor data collection, analysis algorithms, automation logic, and user interface interactions. By simulating various scenarios and inputs, developers can verify the accuracy of sensor readings, the effectiveness of data analysis algorithms, and the responsiveness of the mobile app interface. Unit tests help identify bugs and ensure the reliability and robustness of the system, ultimately contributing to its overall effectiveness and user satisfaction.

Integration Testing: This involves testing how different units of code work together. In the case of the Lite Harvest system, this would involve testing how the code that handles the signals from the sensor's data interacts with the code that controls the System and the mobile application.

System Testing: System testing for the Lite Harvest system with a mobile app involves rigorous evaluation of various components and functionalities. This includes testing the accuracy and reliability of sensor readings for parameters such as pH, EC, humidity, and water level. Additionally, the mobile app undergoes testing to ensure seamless data transmission, real-time monitoring, and user-friendly interface navigation. Integration testing verifies the interaction between the hardware components, sensors, and the mobile app, while performance testing assesses system responsiveness and efficiency under different conditions. Overall, thorough system testing is essential to validate the system's functionality, usability, and reliability in real-world scenarios.

User Acceptance Testing: System testing for the Lite harvest system with a mobile app involves rigorous evaluation of various components and functionalities. This includes testing the accuracy and reliability of sensor readings for parameters such as pH, EC, humidity, and water level. Additionally, the mobile app undergoes testing to ensure seamless data transmission, real-time monitoring, and user-friendly interface navigation. Integration testing verifies the interaction between the hardware components, sensors, and the mobile app, while performance testing assesses system responsiveness and efficiency under different conditions. Overall, thorough system testing is essential to validate the system's functionality, usability, and reliability in real-world scenarios.

7.2 Test Cases

Table 7.1 Test Cases for Lite Harvest

Test Case Id	Test Case Description	Input Data	Expected Output	Actual Output	Pass/Fail
1	Instruction Display	Steps for setup	Steps displayed	Steps displayed	Pass
2	Add user by admin	user login creation	User profile created	User profile created	Pass
3	Edit user details	Profile edit	Updated details	Updated details	Pass
4	Delete User	Profile deletion	Profile deleted	Profile deleted	Pass
5	Management login	User login	Logged in by user	Logged in by user	Pass
6	Intruder Login	Wrong credentials	Display error message	Display error message	Pass
7	Graph based display for sensor values	Sensor data	Read sensor data and plot graph	Read sensor data and plot graph	Pass
8	Changes in sensor value	Sensor data	Update values and graph in display	Update values and graph in display	Pass

Table 7.1 provide the analysis regarding the test cases. It provides a comparison between the actual and expected outputs. It can be said that the Lite Harvest system performed reasonably well on the test cases.

Chapter 8

Conclusions and Future Enhancements

8.1 Conclusions

In conclusion, the fusion of wick-based hydroponic systems with IoT technologies marks a significant transformation in contemporary agriculture, offering numerous advantages that surpass conventional farming methods. By leveraging capillary action and passive wicking systems, cultivators can effectively tackle major challenges like water scarcity, soil degradation, and environmental harm, thereby fostering sustainable and productive plant cultivation. One of the key merits of wick-based hydroponic systems is their ability to minimize water wastage while ensuring optimal delivery of nutrients to plant roots. This not only conserves vital resources but also enhances water efficiency, proving especially beneficial in areas prone to drought or water shortages. Furthermore, by eliminating the requirement for soil, these systems mitigate soil erosion and depletion, safeguarding fertile land and nurturing soil health for future generations. The integration of sophisticated sensors and IoT technologies further elevates the efficiency and productivity of wick-based hydroponic setups. Continuous monitoring of essential parameters such as TDS, temperature, and humidity empowers growers to maintain precise control over growing conditions, enhance nutrient absorption, and pre-empt potential issues before they escalate. The capability to remotely supervise and manage hydroponic systems through mobile applications adds a layer of convenience and accessibility, democratizing the adoption of modern farming techniques and enabling cultivators from diverse backgrounds to engage in sustainable agriculture. The proposed approach offers a systematic framework for implementing and overseeing wick-based hydroponic systems with IoT technologies. By streamlining data acquisition, pre-processing, automation, visualization, and maintenance processes, growers can optimize resource utilization, minimize operational burdens, and maximize yields. The integration of automation technology and cloud connectivity facilitates seamless coordination and communication among various components of the hydroponic system, enabling efficient operation and scalability in an economically viable manner.

8.2 Future Enhancements

In future endeavours focusing on advanced water management for vertical farming, there are several promising avenues to explore for enhancing productivity and sustainability. Broadening the sensor network to encompass additional environmental parameters such as light intensity would provide deeper insights into crop health and growth conditions. Refining machine learning algorithms for predictive analytics and anomaly detection could facilitate proactive management and risk mitigation. Incorporating smart irrigation systems equipped with dynamic scheduling algorithms customized to plant requirements could optimize yields while conserving water usage. Leveraging advanced data analytics and visualization tools, alongside precision agriculture technologies, could equip farmers with actionable insights to refine farming practices and enhance profitability.

References

1. G W Michael, F S Tay, Y L Then. “Development of Automated Monitoring System for Hydroponics Vertical Farming”. In IOP, 28 Nov 2020.
2. Gaganjot Kaur, Prashant Upadhyaya, Paras Chawla. “IoT Based Mobile Application for Monitoring of Hydroponic Vertical Farming”. In IEEE, October 2022.
3. G. Cema, Dr. E. Kaliappan. “AI Based Crop Recommendations for Intensive Farming using WSN”. August 2022.
4. Tamoghna Ojha, Sudip Misra, Narendra Singh Raghuwanshi. “Wireless sensor networks for agriculture: The state-of-the-art in practice and future challenges”. In Elsevier, 5 Sept 2015.
5. Demi Adidrana, Nico Surantha. “Hydroponic Nutrient Control System based on Internet of Things and K-Nearest Neighbors”. In IEEE, October 2019.
6. S. V. S. Ramakrishnam Raju, Bhasker Dappuri, P. Ravi Kiran Varma, Murali Yachamaneni, D. Marlene Grace Verghese, and Manoj Kumar Mishra. “Design and Implementation of Smart Hydroponics Farming Using IoT-Based AI Controller with Mobile Application System”. In Hindawi 12 July 2022.
7. Abdul Rahim Junejo, Shoukat Ali Soomro, Komal Jamil Gujjar, Indian Raj N, "Analysis of Wick Irrigation in Vegetable Planting”. May 2023.
8. Shaheen Javed Roonjho, Rowshon Md Kamal, Abdul Rehman Roonjho, "Modeling capillary wick irrigation system for greenhouse crop production”. In Elsevier, April 2022.
9. D. D. Treadwell, M. Davis and J. M. White, “Effect of Water pH on Yield and Nutritional Status of Greenhouse Cucumber Grown in Recirculating Hydroponics” Oct 2008.
10. Masaru Sakamoto and Takahiro Suzuki, "Effect of Nutrient Solution Concentration on the Growth of hydroponics Sweet potato”. Sept 2020.

Appendix A

A.1 Source Code

// code that displays the first page

main.dart

```
import 'package:flutter/material.dart';
import 'package:flutter/services.dart';
import 'package:firebase_messaging/firebase_messaging.dart';
import 'package:firebase_core/firebase_core.dart';
import 'firebase_options.dart';
import './details.dart';
import './push_notifications.dart';

//function to listen to bg changes
Future _firebaseBAckgroundMessage(RemoteMessage message) async {
  if (message.notification != null) {
    print("Some notification received");
  }
}

void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp(
  options: DefaultFirebaseOptions.currentPlatform,
  );
  PushNotification.init();
  //Listen to bg notifications
  FirebaseMessaging.onBackgroundMessage(_firebaseBAckgroundMessage);
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  // This widget is the root of your application.
```

```
 @override
 Widget build(BuildContext context) {
  SystemChrome.setSystemUIOverlayStyle(
    SystemUiOverlayStyle(statusBarColor: Colors.transparent));
  return MaterialApp(
    debugShowCheckedModeBanner: false,
    title: 'LiteHarvest',
    theme: ThemeData(
     colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
     useMaterial3: true,
    ),
    home: Scaffold(
      backgroundColor: Colors.white,
      appBar: AppBar(
       title: Text('LiteHarvest'),
       backgroundColor: Colors.black,
       centerTitle: true,
       foregroundColor: Colors.white,
      ),
      body: Center(
       child: Details(),
      )));
 }
}

//code to display button onto first page
```

detail.dart

```
import 'package:flutter/material.dart';
import 'organisation.dart';
import 'Instructions/home.dart';

class Details extends StatelessWidget{
 const Details ({Key? key}) : super(key: key);

 @override
```

```
Widget build(BuildContext context) {
  return Container(
    child: Column(
      mainAxisAlignment: MainAxisAlignment.center,
      children: [
        const Text(
          'Details',
          style: TextStyle(
            fontSize: 30,
            fontWeight: FontWeight.bold,
          ),
        ),
        const SizedBox(height: 30.0),
        ElevatedButton(
          style: ElevatedButton.styleFrom(
            foregroundColor: Colors.white,
            backgroundColor: Colors.black,
            padding: const EdgeInsets.symmetric(horizontal: 20, vertical: 20),
            shape: RoundedRectangleBorder(
              borderRadius: BorderRadius.circular(5),
            ),
          ),
          onPressed: () {
            Navigator.of(context).push(
              MaterialPageRoute(
                builder: (BuildContext context) => const Organisation(),
              ),
            );
          },
          child: const Text('Organisation'),
        ),
        const SizedBox(height: 10),
        ElevatedButton(
          style: ElevatedButton.styleFrom(
            foregroundColor: Colors.white,
```

```
              backgroundColor: Colors.black,
              padding: const EdgeInsets.symmetric(horizontal: 40, vertical: 20),
              shape: RoundedRectangleBorder(
                borderRadius: BorderRadius.circular(5),
              ),
            ),
            onPressed: () {
              Navigator.of(context).push(
                MaterialPageRoute(
                  builder: (BuildContext context) => const House(),
                ),
              );
            },
            child: const Text('Instructions'),
          ),
        ],
      ),
    );
  }
}
```

//code to display the organisation page buttons

organisation.dart

```
import 'package:flutter/material.dart';
import 'package:lite_harvest/Admin/login.dart';
import 'package:lite_harvest/Management/UserProfilePage.dart';
// import 'package:lite_harvest/Management/login.dart';

class Organisation extends StatelessWidget{
  const Organisation ({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Scaffold(
        backgroundColor: Colors.white,
```

```
    appBar: AppBar(
      title: Text('LiteHarvest'),
      backgroundColor: Colors.black,
      centerTitle: true,
      foregroundColor: Colors.white,
    ),
  body: Container(
    alignment: Alignment.center,
    child: Column(
      mainAxisAlignment: MainAxisAlignment.center,
      children: [
        const Text(
          'Organisation Login',
          style: TextStyle(
            fontSize: 30,
            fontWeight: FontWeight.bold,
          ),
        ),
        const SizedBox(height: 30.0),
        ElevatedButton(
          style: ElevatedButton.styleFrom(
            foregroundColor: Colors.white,
            backgroundColor: Colors.black,
            padding: const EdgeInsets.symmetric(horizontal: 20, vertical: 20),
            shape: RoundedRectangleBorder(
              borderRadius: BorderRadius.circular(5),
            ),
          ),
          onPressed: () {
            Navigator.of(context).push(
              MaterialPageRoute(
                builder: (BuildContext context) => const UserProfilePage(),
              ),
            );
          },
```

```
          child: const Text('Management'),
        ),
        const SizedBox(height: 10),
        ElevatedButton(
          style: ElevatedButton.styleFrom(
            foregroundColor: Colors.white,
            backgroundColor: Colors.black,
            padding: const EdgeInsets.symmetric(horizontal: 40, vertical: 20),
            shape: RoundedRectangleBorder(
              borderRadius: BorderRadius.circular(5),
            ),
          ),
          onPressed: () {
            Navigator.of(context).push(
              MaterialPageRoute(
                builder: (BuildContext context) => const LoginAdmin(),
              ),
            );
          },
          child: const Text('Admin'),
        ),
      ],
    ),
  ),
  );
 }
}
```

//inside instructions folder

//code to navigate to instructions page

home.dart

```
import 'package:flutter/material.dart';
import 'select.dart';
import 'btn.dart';
```

```
class House extends StatelessWidget{
  const House ({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Scaffold(
        backgroundColor: Colors.white,
        appBar: AppBar(
          title: Text('LiteHarvest'),
          backgroundColor: Colors.black,
          centerTitle: true,
          foregroundColor: Colors.white,
        ),
        body: Column(
          children: [
            Select(),
            Btn(),
          ],
        ),
    );
  }
}
```

//code to display all the buttons that navigate to steps

btn.dart

```
import 'package:flutter/material.dart';
import 'Steps/step1.dart';
import 'Steps/step2.dart';
import 'Steps/step3.dart';
import 'Steps/nutrient.dart';

class Btn extends StatelessWidget{
  const Btn({Key? key}) : super(key: key);
```

```
@override
Widget build(BuildContext context) {
  return Container(
    alignment: Alignment.center,
    child: Column(
      children: [
        const Padding(padding: EdgeInsets.only(top: 60)),
        ElevatedButton(
          onPressed: (){
            Navigator.push(
              context,
              MaterialPageRoute(
                builder: (context) => Step1()
              )
            );
          },
          child: Text(
            "Step 1 : Prepare The Setup",
            style: TextStyle(
              color: Colors.black,
              fontSize: 14
            ),
          ),
          style: ElevatedButton.styleFrom(fixedSize: const Size(450, 50)),
        ),
        const Padding(padding: EdgeInsets.only(top: 30)),
        ElevatedButton(
          onPressed: () {
            Navigator.push(
              context,
              MaterialPageRoute(
                builder: (context) => Step2()
              )
            );
```

```
        },
        child: Text(
          "Step 2 : Germinate Seeds",
          style: TextStyle(
            color: Colors.black,
            fontSize: 14
          ),
        ),
        style: ElevatedButton.styleFrom(fixedSize: const Size(450, 50)),
      ),
      const Padding(padding: EdgeInsets.only(top: 30)),
      ElevatedButton(
        onPressed: () {
          Navigator.push(
            context,
            MaterialPageRoute(
              builder: (context) => Step3()
            )
          );
        },
        child: Text(
          "Step 3 : Final Setup",
          style: TextStyle(
            color: Colors.black,
            fontSize: 14
          ),
        ),
        style: ElevatedButton.styleFrom(fixedSize: const Size(450, 50)),
      ),
      const Padding(padding: EdgeInsets.only(top: 30)),
      ElevatedButton(
        onPressed: () {
          Navigator.push(
            context,
            MaterialPageRoute(
```

```
                builder: (context) => Nutrient()
              )
            );
          },
          child: Text(
            "pH and Nutrients",
            style: TextStyle(
              color: Colors.black,
              fontSize: 14
            ),
          ),
          style: ElevatedButton.styleFrom(fixedSize: const Size(450, 50)),
        ),
      ],
    ),
  );
 }
}
//inside steps folder
```

//code to display instructions for step 1

step1.dart

```
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import '../tile.dart';

class Step1 extends StatelessWidget{
  final List<Tile> tiles = [
    Tile(name: 'Use any plastic bottle to make the container for holding the plants.',img: 'assets/images/bottle.jpeg'),
    Tile(name: 'Cut the plastic bottle into two halves.',img: 'assets/images/cutBottle.jpg'),
    Tile(name: 'Make small holes on the bottle cap.',img: 'assets/images/capholes.jpg'),
    Tile(name: 'Insert the wick through the holes made on the bottle cap until it comes out from the other end of the cut bottle and close the cap.',img: 'assets/images/wickCap.jpeg'),
    Tile(name: 'The basic setup is ready',img: 'assets/images/bottle.jpeg'),
```

```
 ];
 @override
 Widget build(BuildContext context) {
   return Scaffold(
       backgroundColor: Colors.white,
       appBar: AppBar(
         title: Text('Step 1 : Prepare The Setup'),
         backgroundColor: Colors.black,
         centerTitle: true,
         foregroundColor: Colors.white,
       ),
     body:ListView.builder(
          itemCount: tiles.length,
          itemBuilder: (context,index){
            return Padding(
              padding: const EdgeInsets.symmetric(vertical:1.0,horizontal: 4.0 ),
              child: Card(
                child: ListTile(
                  title: Text(tiles[index].name),
                  visualDensity: VisualDensity(vertical: 4),
                  trailing: Image.asset(tiles[index].img),
                ),
            ),
            );
          })
   );
 }
}
```

//code to display instructions for step 2

step2.dart

```
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import '../tile.dart';
```

```
class Step2 extends StatelessWidget{
  final List<Tile> tiles = [
    Tile(name: 'Soak seeds in water for 1hr to ensure they are hydrated, giving them the best chance for germination.',img: 'assets/images/soakSeeds.jpeg'),
    Tile(name: 'Plant the Seeds: Sow one seed in each medium, no deeper than 1cm.',img: 'assets/images/plantSeeds.jpeg'),
    Tile(name: 'Cover your seedlings with a dome or plastic kitchen wrap to create the perfect microclimate for germinating seeds..',img: 'assets/images/coverSeeds.jpeg'),
    Tile(name: 'Maintain and monitor the temperature.',img: 'assets/images/coverSeeds.jpeg'),
  ];
  @override
  Widget build(BuildContext context) {
    return Scaffold(
        backgroundColor: Colors.white,
        appBar: AppBar(
          title: Text('Step 2 : Germinate Seeds'),
          backgroundColor: Colors.black,
          centerTitle: true,
          foregroundColor: Colors.white,
        ),
      body:ListView.builder(
            itemCount: tiles.length,
            itemBuilder: (context,index){
              return Padding(
                padding: const EdgeInsets.symmetric(vertical:1.0,horizontal: 4.0 ),
                child: Card(
                  child: ListTile(
                    title: Text(tiles[index].name),
                    visualDensity: VisualDensity(vertical: 4),
                    trailing: Image.asset(tiles[index].img),
                  ),
              ),
              );
            })
```

```
    );
  }
}
```

//code to display instructions for step 3

step3.dart

```
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import '../tile.dart';

class Step3 extends StatelessWidget{
  final List<Tile> tiles = [
    Tile(name: 'Place the growing media in the top portion of the bottle being sure to hold onto the wick so it spread the height of the top portion of the bottle.',img: 'assets/images/growMedia.jpeg'),
    Tile(name: 'Check and mark how high you want the water, the wick must be submerged. Add the water to your mark. ',img: 'assets/images/waterMark.jpeg'),
        Tile(name: 'Place the top portion back onto the bottom portion.',img: 'assets/images/topBottom.jpeg'),
    Tile(name: 'The hydroponics setup is ready.',img: 'assets/images/system.jpeg'),
  ];
  @override
  Widget build(BuildContext context) {
    return Scaffold(
        backgroundColor: Colors.white,
        appBar: AppBar(
          title: Text('Step 3 : Final Setup'),
          backgroundColor: Colors.black,
          centerTitle: true,
          foregroundColor: Colors.white,
        ),
      body:ListView.builder(
          itemCount: tiles.length,
          itemBuilder: (context,index){
```

```
        return Padding(
          padding: const EdgeInsets.symmetric(vertical:1.0,horizontal: 4.0 ),
          child: Card(
            child: ListTile(
              title: Text(tiles[index].name),
              visualDensity: VisualDensity(vertical: 4),
              trailing: Image.asset(tiles[index].img),
            ),
        ),
        );
      })
  );
 }
}
```

//code to display instructions for nutrient

nutrient.dart

```
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';

class Nutrient extends StatelessWidget {
  final List<Img> _imgs = [
    Img(img: 'assets/images/ph.jpeg'),
    Img(img: 'assets/images/nutrient.jpeg'),
  ];

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.white,
      appBar: AppBar(
        title: Text('pH and Nutrients'),
        backgroundColor: Colors.black,
        centerTitle: true,
```

```
        foregroundColor: Colors.white,
      ),
      body: Expanded(
          child: GridView.builder(
              itemCount: _imgs.length,
              gridDelegate: const SliverGridDelegateWithFixedCrossAxisCount(
                  crossAxisCount: 1),
              itemBuilder: (BuildContext context, int index) {
                return Center(
                  child: Image.asset(_imgs[index].img),
                );
              })),
    );
  }
}

class Img {
  final String img;

  Img({required this.img});
}
```

```
//include the dependencies inside the pubspec.yaml according to new version
firebase_core: ^2.29.0
  firebase_auth: ^4.19.1
  cloud_firestore: ^4.16.1
  firebase_database: ^10.5.2
  firebase_messaging: ^14.8.2
  syncfusion_flutter_charts: ^25.1.40
  http: ^1.2.1
  flutter_local_notifications: ^17.1.0

//assests folder contains the images used, included in the pubspec.yaml

 assets:
   - assets/images/

//run flutter pub get to install the dependencies
```

A.2 Installation Steps

System requirements

To install and run Flutter, your development environment must meet these minimum requirements:

- **Operating Systems**: Windows 10 or later (64-bit), x86-64 based.
- **Disk Space**: 1.64 GB (does not include disk space for IDE/tools).
- **Tools**: Flutter depends on these tools being available in your environment.
  - Windows PowerShell 5.0 or newer (this is pre-installed with Windows 10)
  - Git for Windows 2.x, with the **Use Git from the Windows Command Prompt** option.

    If Git for Windows is already installed, make sure you can run git commands from the command prompt or PowerShell.

Get the Flutter SDK

1. Download the flutter installation bundle to get the latest stable release of the Flutter SDK
2. Extract the zip file and place the contained flutter in the desired installation location for the Flutter SDK (for example, C:\src\flutter).

Warning: Do not install Flutter to a path that contains special characters or spaces.

Warning: Do not install Flutter in a directory like C:\Program Files\ that requires elevated privileges.

If you don't want to install a fixed version of the installation bundle, you can skip steps 1 and 2. Instead, get the source code from the Flutter repo on GitHub, and change branches or tags as needed. For example:

```
C:\src>git clone https://github.com/flutter/flutter.git -b stable
```

You are now ready to run Flutter commands in the Flutter Console.

Update your path

If you wish to run Flutter commands in the regular Windows console, take these steps to add Flutter to the PATH environment variable:

- From the Start search bar, enter 'env' and select **Edit environment variables for your account**.

- Under **User variables** check if there is an entry called **Path**:
  - If the entry exists, append the full path to flutter\bin using ; as a separator from existing values.
  - If the entry doesn't exist, create a new user variable named Path with the full path to flutter\bin as its value.

You have to close and reopen any existing console windows for these changes to take effect.

Note: As of Flutter's 1.19.0 dev release, the Flutter SDK contains the dart command alongside the flutter command so that you can more easily run Dart command-line programs. Downloading the Flutter SDK also downloads the compatible version of Dart, but if you've downloaded the Dart SDK separately, make sure that the Flutter version of dart is first in your path, as the two versions might not be compatible. The following command tells you whether the flutter and dart commands originate from the same bin directory and are therefore compatible.

```
C:\>where flutter dart
C:\path-to-flutter-sdk\bin\flutter
C:\path-to-flutter-sdk\bin\flutter.bat
C:\path-to-dart-sdk\bin\dart.exe        :: this should go after `C:\path-to-flutter-sdk\bin\` commands
C:\path-to-flutter-sdk\bin\dart
C:\path-to-flutter-sdk\bin\dart.bat
```

As shown above, the command dart from the Flutter SDK doesn't come first. Update your path to use commands from C:\path-to-flutter-sdk\bin\ before commands from C:\path-to-dart-sdk\bin\ (in this case). After restarting your shell for the change to take effect, running the where command again should show that the flutter and dart commands from the same directory now come first.

```
C:\>where flutter dart
C:\dev\src\flutter\bin\flutter
C:\dev\src\flutter\bin\flutter.bat
C:\dev\src\flutter\bin\dart
C:\dev\src\flutter\bin\dart.bat
C:\dev\src\dart-sdk\bin\dart.exe
```

However, if you are using PowerShell, in it where is an alias of Where-Object command, so you need to use where.exe instead.

```
PS C:\> where.exe flutter dart
```

To learn more about the dart command, run dart -h from the command line, or see the dart tool page.

Run flutter doctor

From a console window that has the Flutter directory in the path (see above), run the following command to see if there are any platform dependencies you need to complete the setup:

```
C:\src\flutter>flutter doctor
```

This command checks your environment and displays a report of the status of your Flutter installation. Check the output carefully for other software you might need to install or further tasks to perform (shown in **bold** text).

For example:

```
[-] Android toolchain - develop for Android devices
    • Android SDK at D:\Android\sdk
    ✗ Android SDK is missing command line tools; download from https://goo.gl/XxQghQ
    • Try re-installing or updating your Android SDK,
      visit https://docs.flutter.dev/setup/#android-setup for detailed instructions.
```

The following sections describe how to perform these tasks and finish the setup process. Once you have installed any missing dependencies, you can run the flutter doctor command again to verify that you’ve set everything up correctly.

Note: If flutter doctor returns that either the Flutter plugin or Dart plugin of Android Studio are not installed, move on to Set up an editor to resolve this issue.

Warning: The Flutter tool may occasionally download resources from Google servers. By downloading or using the Flutter SDK you agree to the Google Terms of Service.

For example, when installed from GitHub (as opposed to from a prepackaged archive), the Flutter tool will download the Dart SDK from Google servers immediately when first run, as it is used to execute the flutter tool itself. This will also occur when Flutter is upgraded (e.g. by running the flutter upgrade command).

The flutter tool uses Google Analytics to report feature usage statistics and send crash reports. This data is used to help improve Flutter tools over time.

Flutter tool analytics are not sent on the very first run. To disable reporting, run flutter config --no-analytics. To display the current setting, use flutter config. If you opt out of analytics, an opt-out event is sent, and then no further information is sent by the Flutter tool.

Dart tools may also send usage metrics and crash reports to Google. To control the submission of these metrics, use the following options on the dart tool:

- --enable-analytics: Enables anonymous analytics.
- --disable-analytics: Disables anonymous analytics.

The Google Privacy Policy describes how data is handled by these services.

Android setup

Note: Flutter relies on a full installation of Android Studio to supply its Android platform dependencies. However, you can write your Flutter apps in a number of editors; a later step discusses that.

Install Android Studio

1. Download and install Android Studio.
2. Start Android Studio, and go through the 'Android Studio Setup Wizard'. This installs the latest Android SDK, Android SDK Command-line Tools, and Android SDK Build-Tools, which are required by Flutter when developing for Android.
3. Run flutter doctor to confirm that Flutter has located your installation of Android Studio. If Flutter cannot locate it, run flutter config --android-studio-dir <directory> to set the directory that Android Studio is installed to.

Set up your Android device

To prepare to run and test your Flutter app on an Android device, you need an Android device running Android 4.1 (API level 16) or higher.

1. Enable **Developer options** and **USB debugging** on your device. Detailed instructions are available in the Android documentation.
2. Windows-only: Install the Google USB Driver.
3. Using a USB cable, plug your phone into your computer. If prompted on your device, authorize your computer to access your device.
4. In the terminal, run the flutter devices command to verify that Flutter recognizes your connected Android device. By default, Flutter uses the version of the Android SDK where your adb tool is based. If you want Flutter to use a different installation of the Android SDK, you must set the ANDROID_SDK_ROOT environment variable to that installation directory.

Set up the Android emulator

To prepare to run and test your Flutter app on the Android emulator, follow these steps:

1. Enable VM acceleration on your machine.
2. Launch **Android Studio**, click the **AVD Manager** icon, and select **Create Virtual Device…**
    - In older versions of Android Studio, you should instead launch **Android Studio > Tools > Android > AVD Manager** and select **Create Virtual Device…**. (The **Android** submenu is only present when inside an Android project.)
    - If you do not have a project open, you can choose **Configure > AVD Manager** and select **Create Virtual Device…**
3. Choose a device definition and select **Next**.
4. Select one or more system images for the Android versions you want to emulate, and select **Next**. An *x86* or *x86_64* image is recommended.
5. Under Emulated Performance, select **Hardware - GLES 2.0** to enable hardware acceleration.
6. Verify the AVD configuration is correct, and select **Finish**.

    For details on the above steps, see Managing AVDs.

7. In Android Virtual Device Manager, click **Run** in the toolbar. The emulator starts up and displays the default canvas for your selected OS version and device.

Agree to Android Licenses

Before you can use Flutter, you must agree to the licenses of the Android SDK platform. This step should be done after you have installed the tools listed above.

1. Make sure that you have a version of Java 8 installed and that your JAVA_HOME environment variable is set to the JDK's folder.

    Android Studio versions 2.2 and higher come with a JDK, so this should already be done.

2. Open an elevated console window and run the following command to begin signing licenses.

```
$ flutter doctor --android-licenses
```

3. Review the terms of each license carefully before agreeing to them.
4. Once you are done agreeing with licenses, run flutter doctor again to confirm that you are ready to use Flutter.