Multisensor Crop Guidance Real Time Engine with Cloud Analytics and Recommendation System

A MAJOR PROJECT REPORT

Submitted by

ACHAL KAMBOJ [RA2011026010028]
ANURAG MALIK [RA2011026010030]
ANMOL AGARWAL [RA2011026010034]
JAIADITYA GHORPADE [RA2011026010035]

Under the Guidance of

Dr. Sumathy G
Assistant Professor
Department of Computational Intelligence

in partial fulfillment of the requirements for the degree of

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE ENGINEERING
with specialization in Artificial Intelligence and Machine Learning

SRM INSTITUTE OF SCIENCE & TECHNOLOGY
Deemed to be University u/s 3 of UGC Act, 1956

DEPARTMENT OF COMPUTATIONAL INTELLIGENCE
COLLEGE OF ENGINEERING AND TECHNOLOGY
SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR- 603 203

May 2024

Department of Computational Intelligence
SRM Institute of Science & Technology
Own Work* Declaration Form

This sheet must be filled in (each box ticked to show that the condition has been met). It must be signed and dated along with your student registration number and included with all assignments you submit – work will not be marked unless this is done.
<u>To be completed by the student for all assessments</u>

Degree/ Course : **B-Tech /Computer Science Engineering with specialization AI/ML**

Student Name : **Achal Kamboj, Anurag Malik, Anmol Agarwal, Jaiaditya Ghorpade**

Registration Number : **RA2011026010028, RA2011026010030, RA2011026010034, RA2011026010035**

Title of Work : **Multisensor Crop Guidance Real Time Engine with Cloud Analytics and Recommendation System**

We hereby certify that this assessment compiles with the University's Rules and Regulations relating to Academic misconduct and plagiarism**, as listed in the University Website, Regulations and the Education Committee guidelines.

We confirm that all the work contained in this assessment is our own except, where indicated and that we have met the following conditions:

- Clearly referenced / listed all sources as appropriate
- Referenced and put in inverted commas all quoted text (from books, web, etc)
- Given the sources of all pictures, data etc. that are not my own
- Not made any use of the report(s) or essay(s) of any other student(s) either past or present
- Acknowledged in appropriate places any help that we have received from others (e.g. fellow students, technicians, statisticians, external sources)
- Compiled with any other plagiarism criteria specified in the Course handbook / University website

We understand that any false claim for this work will be penalized in accordance with the University policies and regulations.

DECLARATION:
We are aware of and understand the University's policy on Academic misconduct and plagiarism and I certify that this assessment is our own work, except where indicated by referring and that we have followed the good academic practices noted above.
Achal Kamboj [RA2011026010028] Anurag Malik [RA2011026010030] Anmol Agarwal [RA2011026010034] Jaiaditya Ghorpade [RA2011026010035]

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR – 603 203

BONAFIDE CERTIFICATE

Certified that 18CSP109L project report "**Multisensor Crop Guidance Real Time Engine with Cloud Analytics and Recommendation System**" is the bonafide work of "**ACHAL KAMBOJ [RA2011026010028], ANURAG MALIK [RA2011026010030], ANMOL AGARWAL [RA2011026010034], JAIADITYA GHORPADE [RA2011026010035]**" who carried out the project work under my supervision. Certified further, that to the best of my knowledge, the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Dr. Sumathy G
SUPERVISOR
ASSISTANT PROFESSOR
Department of Computational Intelligence

SIGNATURE

Dr. R. ANNIE UTHRA
HEAD OF THE DEPARTMENT
PROFESSOR
Department of Computational Intelligence

Examiner I

Examiner 2

ACKNOWLEDGEMENT

We want to express our gratitude to **Dr. C. Mathamizhchelvan,**Vice-Chancellor of SRM Institute of Science and Technology, for providing the necessary facilities for our project work and for his ongoing support.

Our sincere thanks go to **Dr. T.V.Gopal**, Dean-CET of SRM Institute of Science and Technology, for his invaluable support.

We are incredibly thankful to **Dr. R. Annie Uthra**, Professor and Head of the Department of Computational Intelligence at SRM Institute of Science and Technology, for her suggestions and encouragement throughout all stages of our project work.

We express our immeasurable respect and heartfelt gratitude to our Guide, **Dr. Sumathy G**, Assistant Professor and our Panel Head **Dr.M.Uma** Associate Professor from the Department of Computational Intelligence at SRM Institute of Science and Technology, for providing us with the opportunity to pursue our project under their mentorship. They granted us the freedom and support to explore research topics of our interest. Their passion for problem-solving and making a difference in the world has always been a source of inspiration to us. We appreciate their valuable inputs during the duration of the project and for their continuous support.

We express our boundless gratitude to **Dr. Suresh .K**, Assistant Professor from the Department of Computational Intelligence at SRM Institute of Science and Technology, who served as our Faculty Advisor. We appreciate his leadership and guidance, which were instrumental in helping us successfully complete our course.

We extend our sincere thanks to the faculty and students of the Computational Intelligence Department at SRM Institute of Science and Technology for their assistance during our project. Finally, we would like to express our appreciation to our parents, family members and friends for their unwavering love, constant support and encouragement.

Anurag Malik [RA2011026010030]

Anmol Agarwal [RA2011026010034]

Achal Kamboj [RA2011026010028]

Jaiaditya Ghorpade[RA2011026010035]

ABSTRACT

The Multisensor Crop Guidance Real-Time Engine with Cloud Analytics and Recommendation System stands as a pioneering advancement in agricultural technology, poised to redefine crop management practices. By integrating an array of sensors including NPK, moisture, temperature, soil pH and water pH with sophisticated analytics and cloud-based computing, this system delivers real-time insights and personalized recommendations to farmers, elevating crop health and yield. Powered by Adafruit cloud analytics, the project processes sensor data to generate actionable insights. The incorporation of the random forest algorithm enhances prediction accuracy and recommendation precision, empowering farmers to make informed decisions. The system's design includes the seamless integration of sensors with a breadboard and electric wires, ensuring efficient data collection and transmission. This project directly addresses key agricultural challenges such as resource optimization, environmental sustainability and enhanced crop productivity through the adoption of cutting-edge technologies. Its ability to provide timely guidance based on multisensor data and cloud analytics positions it as an invaluable asset for modern agriculture. The system holds the potential to revolutionize crop and resource management practices, paving the way for a future where agriculture is more efficient, sustainable and productive. The Multisensor Crop Guidance Real-Time Engine with Cloud Analytics and Recommendation System represents a transformative force in agriculture. Its integration of advanced technologies promises to reshape farming practices, enabling farmers to navigate challenges more effectively and achieve greater success in crop cultivation.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

MCU	Micro Controller Unit
WIFI	Wireless Fidelity
API	Application Programming Interface
RF	Random Forest
KNN	K Nearest Neighbors
DT	Decision Tree
SVM	Support Vector Machine
NPK	Nitrogen, Phosphorus, Potassium
MCG-RE	Multisensor Crop Guidance-Realtime Engine

ABBREVIATIONS

MCU	Micro Controller Unit
WIFI	Wireless Fidelity
API	Application Programming Interface
RF	Random Forest
KNN	K Nearest Neighbors
DT	Decision Tree
SVM	Support Vector Machine
NPK	Nitrogen, Phosphorus, Potassium
MCG-RE	Multisensor Crop Guidance-Realtime Engine

CHAPTER 1

INTRODUCTION

1.1 General

Technology has changed many aspects of human interaction in the last few years, including how we interact with digital devices and oversee farming operations. The emergence of gesture-driven interfaces has attracted significant attention as a notable advancement that is revolutionizing the way users interact with virtual environments. These interfaces have brought in a new era of user experience by providing a natural and intuitive method of control, allowing people to manipulate virtual objects and navigate digital landscapes with simple hand movements.

This new understanding of the dynamics of interaction extends beyond the world of digital devices. Simultaneously, developments in data analytics and sensor technology have sparked innovations in agriculture, opening the door for precision farming methods that maximize crop yield and resource efficiency. With the help of automated plant irrigation systems and sensors for pH, NPK (nitrogen, phosphorus and potassium) and humidity, farmers are now better equipped to monitor and control crop conditions.

The main goal of this project is to develop a new system that guarantees high crop monitoring accuracy and has an easy-to-use interface for users by utilizing the most recent advancements in computer vision, machine learning and cloud computing. The system provides farmers with actionable insights into crop health, environmental conditions and irrigation requirements through real-time cloud analytics made possible by Adafruit. This allows for proactive intervention and the optimization of agricultural practices.

Furthermore, the addition of a crop recommendation system gives the framework an extra degree of intelligence and gives users individualized advice and recommendations based on their own goals and needs. This integrated approach to crop management promotes a more effective use of resources while also increasing productivity and sustainability, which strengthens the resilience and sustainability of agricultural systems.

This project, which envisions a future where data-driven insights and user-friendly interfaces drive precision farming, is essentially a convergence of cutting-edge technology and agricultural science. The suggested framework has the ability to completely transform agricultural practices by utilizing multisensor technology, analytics and recommendation systems. This will allow farmers to manage their crops with never-before-seen levels of efficiency and control. The opportunities for innovation and change are endless as we set out on this path to a more productive and sustainable agricultural future.

1.2 Purpose

The goal of this project is to create a complete system that will use current technical breakthroughs to revolutionize agricultural practices through precision farming. By combining gesture-driven interfaces, data analytics, sensor technologies and cloud computing, the project hopes to improve crop monitoring accuracy while being user-friendly. The major purpose is to offer farmers useful knowledge regarding crop health, environmental conditions and irrigation requirements, allowing for proactive intervention and improved agricultural practices.

1.3 Scope

This project involves designing, developing and implementing a multidimensional precision farming system. Key components include gesture-driven interfaces for intuitive interaction, advanced analytics and sensors for real-time monitoring of critical metrics, cloud-based platforms for remote access and data processing, a crop recommendation system for tailored advice and a user-friendly interface that prioritizes simplicity and ease of use. The project report will detail the system's design, development and assessment, highlighting its success in attaining high crop monitoring accuracy, usability and sustainability. Furthermore, the paper will look at future developments and applications, highlighting potential for additional innovation and advancement in precision agricultural technology.

The Multisensory Crop Guidance REALTIME Engine ambitiously integrates cutting-edge sensors, including moisture, NPK, environmental and pH sensors, for real-time precision agriculture. The project's scope encompasses an Automatic Irrigation System triggered by moisture sensors, dynamic nutrient monitoring with NPK sensors, environmental parameter tracking, soil/water pH monitoring and a Cloud Analytics platform with a machine learning-powered Recommendation System. The user interface ensures accessibility and seamless data visualization. The holistic approach aims to optimize resource utilization, enhance crop health and establish a data-driven, autonomous system, fostering sustainable and efficient farming practices.

1.4 Requirements

Node MCU acts as our project's primary microcontroller, the Node MCU allows for connectivity between a range of sensors and the cloud analytics platform. Because of its Wi-Fi connectivity, data can be sent to the cloud with ease, allowing for real-time crop status monitoring. Because of its small size and low power consumption, the Node MCU is perfect for agricultural situations where it can give our multisensor crop guiding system flexibility and scalability. Temperature Sensors used for plant growth and development are directly impacted by ambient temperature conditions, so the temperature sensor is crucial for monitoring. Accurate temperature measurement allows us to give information on the ideal growing environment for various crops. By incorporating temperature data into our cloud analytics platform, farmers may make well-informed decisions on crop management techniques, leading to increased yield and productivity.

Moisture sensors are essential for measuring soil moisture content, which is a crucial factor in figuring out how much irrigation is needed. By putting moisture sensors in the soil, we can avoid over or under-irrigation, save water and manage resources more effectively. We can also monitor the moisture content of the soil in real-time. Our recommendation system's integration of moisture sensor data allows for customized irrigation schedules based on the demands of individual crops, maximizing crop production and health.

Water pump is used to supply water to crops precisely and on schedule by automating the irrigation process based on sensor readings. We can accomplish automated irrigation scheduling by integrating the water pump into our system, which will lower the labor expenses and need for manual intervention that come with traditional irrigation techniques. This contributes to resource efficiency and environmental conservation by encouraging sustainable farming methods and improving the efficiency of water consumption.

Relays are used to regulate the water pump's and other electrical equipment' operations in response to preset parameters or user directions. Relays and the Node MCU may be interfaced to automate processes like turning on and off the water pump in response to moisture sensor data.

This improves our system's scalability and flexibility by enabling the automation of different agricultural operations to be tailored to the needs of individual farms.

pH sensors (water and soil) are used to determine the levels of acidity or alkalinity in the water and soil, which are important variables that affect plant nutrient intake and crop health in general. We can prevent nutrient imbalances and inadequacies by monitoring and maintaining the ideal pH levels in irrigation water and soil through the integration of pH sensors into our system. This makes it possible to precisely control pH and fertilizer levels, providing crops with ideal growth conditions and increasing potential yield.

Breadboard is permanently installed before an electrical circuit, it can be assembled and tested on a breadboard as a prototype platform. During the development stage, we may rapidly iterate on sensor setups and circuit designs using breadboards, which speeds up the process. This makes it easier to quickly experiment with and validate sensor integrations, guaranteeing their functioning and dependability prior to field deployment.

NPK Sensor is necessary nutrients for plant growth and development, nitrogen (N), phosphorus (P) and potassium (K) are measured by NPK sensors in the soil. We can determine the nutrient condition of the soil and adjust fertilizer treatments to match crop needs by including NPK sensors into our system. In addition to optimizing nitrogen absorption and crop output, this encourages effective nutrient management by reducing fertilizer waste and environmental damage.

Adapters are used to guarantee that electrical components are compatible with power sources and to supply power to them. We may standardize power supply connections and guarantee dependable operation of sensors and electrical devices in our system by employing adaptors. This improves our multisensor crop guiding system's usability and dependability by streamlining installation and maintenance processes.

Max485 module enables communication between the Node MCU and sensors using the RS-485 serial communication standard. Multiple sensors and the primary controller may build dependable and strong communication linkages by including the Max485 module into our system. This improves the scalability and breadth of our multisensor crop guiding system by enabling data aggregation and transmission over long distances.

1.5 Motivation

Our report is motivated by a deep-rooted desire to address the challenges faced by traditional farming methods. We recognize the inefficiencies, resource wastage and unpredictable environmental factors that often plague traditional agricultural practices. These issues prompt us to advocate for a shift towards more sustainable and technology-driven approaches.

Driven by the pressing need to meet the growing demand for food production, particularly with the increasing global population, our project is driven by a strong sense of purpose. We aim to enhance agricultural productivity while simultaneously minimizing the environmental impact. To achieve this, we're harnessing the power of advanced sensors and automation.

Our commitment to sustainability is at the core of our project. We understand the importance of conserving resources and that's why we're incorporating features like Soil/Water pH Monitoring and personalized recommendations. These elements not only help optimize resource usage but also contribute to reducing water wastage and minimizing the environmental footprint of agriculture.

By merging technological innovation with a steadfast commitment to sustainability, our project aims to empower farmers with precise, data-driven insights and automated solutions. Our ultimate goal is to enhance food security and contribute to building a resilient and efficient global agricultural landscape. Through our efforts, we're striving to make a positive impact on the world, one farm at a time.

1.6 Innovation

The hallmark of our project lies in its incorporation of a diverse array of sensors, including moisture, NPK, environmental and pH sensors. This comprehensive multisensory integration enables real-time monitoring of crucial parameters, offering farmers a holistic view of their crops' conditions. Our innovation lies in the system's ability to synthesize data from various sources, empowering farmers with insights for more informed decision-making.

A key innovation in our project is the implementation of an Automatic Irrigation System that triggers irrigation events autonomously based on real-time moisture sensor readings. This leap forward in efficient water resource management represents a significant advancement. By enabling real-time autonomous decision-making, our system ensures crops receive precise hydration when needed, optimizing growth conditions without relying on manual intervention Another pioneering feature of our project is the integration of a machine learning-powered Recommendation System. By leveraging historical and real-time data, this system provides personalized suggestions for crop management. This innovation transcends traditional rule-based systems by adapting recommendations to individual farm conditions. By fostering a more adaptive and efficient approach to agriculture, our Recommendation System represents a paradigm shift in crop management practices.

These innovations collectively demonstrate our commitment to pushing the boundaries of agricultural technology. By harnessing the power of multisensory integration, real-time autonomous decision-making and machine learning-driven recommendations, our project aims to revolutionize precision farming and empower farmers with the tools they need to thrive in an ever-changing agricultural landscape.

1.7 Product Vision Statement

Our vision is to empower farmers with a cutting-edge precision farming solution that revolutionizes agricultural practices through advanced technology and sustainable innovation. We envision a future where farmers have access to a comprehensive system that seamlessly integrates diverse sensors, real-time data analytics and autonomous decision-making capabilities. Our product aims to provide farmers with actionable insights and personalized recommendations, enabling them to optimize crop management, conserve resources and enhance productivity. By merging technological innovation with a commitment to sustainability, our vision is to foster a resilient and efficient global agricultural landscape, ensuring food security for generations to come.

CHAPTER 2

BACKLOG REFINEMENT

The backlog for the Multisensor Crop Guidance Real-Time Engine with Cloud Analytics and Recommendation System serves as a critical tool for prioritizing and managing the development of features and functionalities. It encapsulates a comprehensive list of user stories, technical tasks and enhancements necessary for the system's development and deployment. The backlog is continuously refined and updated to reflect evolving requirements, priorities and stakeholder feedback.

The backlog is pivotal in ensuring that the development team is aligned with the project's goals and that the most valuable features are delivered in a timely manner. It also facilitates transparency and collaboration among stakeholders, as they can track progress and provide input on the backlog items. Additionally, the backlog helps in managing dependencies, risks and resources effectively, thereby contributing to the overall success of the project.

2.1 Backlog Of Products

The backlog of products for the Multisensor Crop Guidance Real-Time Engine with Cloud Analytics and Recommendation System comprises a comprehensive set of features and functionalities aimed at leveraging the latest research and technological advancements in agriculture. These features are meticulously designed to enhance decision-making processes, elevate crop health and growth, optimize resource utilization and foster sustainable agricultural practices. Table 2.1 outlines the key components of the backlog, including but not limited to intelligent irrigation scheduling, fertilizer recommendation systems, crop disease detection, water quality monitoring and digital business penetration. Each of these components plays a pivotal role in the system's overarching goal of providing farmers with precise and actionable insights for effective crop management. By integrating multisensor data, cloud analytics and personalized recommendation systems, the engine aims to revolutionize agriculture by enhancing economic gains, monitoring water quality and detecting crop diseases. Through the implementation of advanced analytics capabilities and optimization of system performance, the backlog items are set to deliver a transformative impact on agriculture's future .

Table 2.1 Product Backlog

ID	Title	Epic	User	Story	Priority (MoS/CoW)	Status	Acceptance Criteria	Functional Requirements	Non-Functional Requirements
1	Decision Support for Food Safety & Quality	Food Safety & Quality Management	Government, Farmers, Consumers	Analyze stakeholder decision-making for food safety & quality	High (MoS)	In Progress	Align stakeholder decisions with food safety & quality needs Identify factors influencing food safety & quality	Tools for stakeholder decision analysis Reporting and visualization of decision factors	User friendly interface for all stakeholders' Secure data storage and access Scalable system to accommodate future needs
2	Digital Literacy for Sustainable Rural Growth	Digital change for Farmers	Government, NGOs	Enhance digital skills and responsiveness of farmers	High (MoS)	Backlog	Increased economic gain for farmers through digital business Promote sustainable economic growth in rural areas	Digital literacy training programs for farmers Market access tools and platforms	Accessible platforms for all levels of digital literacy Integration with existing agricultural support systems
3	Intelligent Irrigation Management System	Water Conservation & Optimization	Farmers	Optimize irrigation scheduling for water and energy savings	High (CoW)	In Design	Reduced water usage and energy consumption	Adaptive scheduling based on crop and soil data Real-time monitoring	Integration with existing irrigation infrastructure Secure and reliable communication protocols

ID	Title	Epic	User	Story	Priority (MoS/CoW)	Status	Acceptance Criteria	Functional Requirements	Non-Functional Requirements
4	Water Quality Monitoring & Assessment System	Sustainable Water Management	Government, Environm-ental Agencies	Monitor and assess water quality for various uses	Medium (CoW)	Not Started	Improved water quality for drinking and irrigation	Real-time water quality data collection and analysis Alerts and notifications for potential water quality issues	Cost-effective and reliable sensors Scalable system for wider deployment
5	Multisensor Data Fusion for Crop & Environment Monitoring	Precision Agriculture & Sustainability	Farmers, Researchers	Integrate multisensor data for crop yield prediction and environment al monitoring	Medium (MoS)	In Testing	Improved crop yield predictions Condition-based environmental monitoring	Integration with various agricultural sensors Machine learning algorithms for data analysis	Secure data storage and processing User-friendly interface for visualization of results
6	Weather-Based Crop Prediction using Big Data	Data-Driven Crop Management	Farmers, Government	Leverage big data for weather-bas ed crop prediction	Medium (MoS)	In Development	Valuable insights for informed crop selection & management decisions	Weather data processing, analysis and prediction tools	Integration with existing weather data sources Scalable infrastructure for big data processing

By incorporating these features into the product backlog, the Multisensor Crop Guidance Real-Time Engine with Cloud Analytics and Recommendation System can be further optimized to meet the evolving needs of the agricultural industry, paving the way for a more sustainable and efficient agricultural ecosystem.

2.2 Product Roadmap

The product roadmap for the Multisensor Crop Guidance Real-Time Engine with Cloud Analytics and Recommendation System outlines a strategic plan to integrate key features and technologies derived from cutting-edge research to enhance decision-making processes, improve crop health and growth, optimize resource utilization and promote sustainable agricultural practices.

Phase 1: Research and Development

Objective:

Conduct in-depth research and analysis on the evolutionary decision-making behavior among government, farmers and consumers, focusing on agricultural product quality and safety.

Activities:

Review and analyze research papers on decision-making behavior among stakeholders. Identify key factors influencing food safety and agricultural product quality. Evaluate different factor value ranges impacting decision-making convergence speed.

Deliverables:

Research report on decision-making behavior among stakeholders.

Phase 2: Technology Integration

Objective:

Integrate technologies such as digital business penetration, IoT and machine learning into the system to enhance farmers' economic gain, water quality monitoring and crop disease detection.

Activities:

Develop features to enhance farmers' entrepreneurial intention and attitude toward digitization. Implement WaterNet network architecture for real-time water quality monitoring. Integrate CROPCARE for real-time crop disease detection using mobile vision.

Deliverables:

Integrated system with features for economic gain, water quality monitoring and crop disease detection.

Phase 3: Pilot Testing and Validation

Objective:

Conduct pilot testing of the system to validate its effectiveness in improving crop yield, water management and economic gain for farmers.

Activities:

Deploy the system in pilot areas and collect feedback from farmers. Analyze data to assess the impact on crop yield, water management and economic gain.

Deliverables:

Pilot test report with findings and recommendations for further improvements.

Phase 4: Scale-up and Deployment

Objective:

Scale up the system for deployment in a wider area, focusing on enhancing food production and sustainability.

Activities:

Expand the deployment of the system to additional regions. Monitor and evaluate the system's performance in terms of food production and sustainability.

Deliverables:

Deployed system in multiple regions with improved food production and sustainability metrics.

Phase 5: Continuous Improvement and Maintenance

Objective:

Continuously improve the system based on feedback and emerging technologies to ensure its effectiveness and relevance.

Activities:

Regularly update the system with new features and enhancements. Monitor and analyze system performance for continuous improvement.

Deliverables:

Regular updates and improvements to the system based on feedback and technological advancements. By following this roadmap, the Multisensor Crop Guidance Real-Time Engine with Cloud Analytics and Recommendation System aims to revolutionize agriculture by leveraging advanced technologies to improve decision-making, optimize resource utilization and promote sustainable practices for the future of farming.

2.3 High Level Estimation of All Epics

The high-level estimates for each epic outline the key phases of the project and their anticipated timelines. The first epic, "Research and Development," involved conducting research and analysis, with an estimated duration of six months. Following this, the "Technology Integration" epic focused on integrating digital business penetration, IoT and machine learning technologies, ensuring system reliability and scalability within another six-month timeframe. Subsequently, the "Pilot Testing and Validation" epic aimed to validate the system's effectiveness in pilot areas, deploying the system and gathering feedback for analysis over six months. Once validated, the "Scale-up and Deployment" epic successfully deployed the system in wider areas, ensuring scalability and providing user training, also spanning six months. Finally, the "Continuous Improvement and Maintenance" epic focused on regular updates and maintenance, ensuring system stability and reliability on an ongoing basis. Each epic represents a crucial stage in the project's lifecycle, with estimated timelines guiding the project's progression and ultimate success.

The high-level estimates for each epic offer a comprehensive overview of the project's major milestones and their anticipated durations. Beginning with research and development, the project laid its foundation through thorough analysis over six months. Subsequently, technology integration facilitated the incorporation of advanced digital technologies, ensuring the system's robustness and scalability within the same timeframe. Pilot testing and validation followed, allowing for the assessment of system effectiveness in real-world scenarios over six months. Scale-up and deployment then expanded the project's reach, ensuring widespread implementation and user readiness within another six-month period. Finally, continuous improvement and maintenance underscored the project's commitment to long-term viability

From the Table 2.2 the estimations provide a high-level overview of the time and effort required for each epic. Actual timelines and efforts may vary based on the complexity of the project, availability of resources and unforeseen challenges encountered during implementation.

Table 2.2 High-Level Estimation of All Epic

S.No.	Title	Epic User Story Priority (MoS/CoW)	Status	Acceptance Criteria	Functional Requirements	Non-Functional Requirements	Original Estimate
1	Research and Development	High	Finished	Completion of research report	Conduct research and analysis	N/A	6 months
2	Technology Integration	High	Finished	Successful integration and testing of technologies	Integrate digital business penetration, IoT and machine learning technologies	Ensure system reliability and scalability	6 months
3	Pilot Testing and Validation	High	Finished	Validation of system effectiveness in pilot areas	Deploy system in pilot areas	Gather feedback and analyze impact	6 months
4	Scale-up and Deployment	High	Finished	Successful deployment in wider areas	Expand deployment to additional regions	Ensure scalability and user training	6 months
5	Continuous Improvement and Maintenance	High	Finished	Regular updates and maintenance of the system	Update system with new features	Ensure system stability and reliability	Ongoing

CHAPTER 3

SPRINT PLANNING

3.1 Sprint 1

3.1.1 Capacity Plan for Sprint 1

For Sprint 1, our team consists of 4 dedicated members, each contributing full-time availability to the project. With a sprint duration of 2 weeks, we have calculated our capacity to ensure efficient planning and allocation of tasks. Considering a standard workweek of 90 hours per team member, the total available hours for Sprint 1 amount to 360 hours. This calculation enables us to gauge the amount of work that can be realistically accomplished within the sprint timeframe. By adhering to this capacity plan, we aim to optimize our productivity while ensuring that we commit to a feasible workload for Sprint 1. Regular reviews and adjustments will be made as necessary to maintain alignment with our project goals and objectives explained in Table 3.1.

Table 3.1 Capacity Plan for Sprint 1

Team Member	Role	Working Days (per Sprint)	Planned Leave	Other Coursework	Design/ Development/ Testing/ Documentation	Estimated Hours
Jaiaditya Ghorpade	Developer	14	2	3	Hardware Setup	90
Anurag Malik	Developer	14	2	3	Adafruit Setup	90
Achal Kamboj	Data Scientist	14	2	3	Machine Learning	90
Anmol Agarwal	Frontend	14	2	3	UI/UX Design	90

3.1.2 Detailed Estimation of User Stories for Sprint 1

Table 3.2 represents the estimations. It serves as a guideline for the team to effectively plan and prioritize their tasks during Sprint 1. It's important to regularly review and adjust these estimations based on actual progress and any unforeseen challenges encountered during the sprint.

Table 3.2 Detailed Estimate for Sprint 1

Title	Epic	Priority (MoS/CoW)	Status	Acceptance Criteria	Functional Requirements	Non-Functional Requirements	Original Estimate
Set up Development Environment and Project Structure	Development Environment Setup	High	In Progress	Development environment configured with necessary tools Project code organized for easy navigation and version control	IDE or code editor installed and configured Version control system implemented (e.g., Git) Project structure facilitates collaboration and code management	User-friendly development environment Secure access control for version control system	8
Research and Select Appropriate Sensors	Sensor Selection	High	In Progress	List of shortlisted sensors with specifications meeting project requirements	Research conducted on available sensors for moisture, NPK, environmental and pH monitoring	Compatibility with chosen development platform and system framework	7
Implement Sensor Integration with the System Framework	Sensor Integration	High	In Progress	Sensors successfully connected and communicating with the system Data transmission from sensors to the system is seamless	Code developed for sensor integration Communication protocols established for data exchange	Secure data transmission between sensors and system	7

3.1.3 Daily Scrum Activities

Daily Scrum activities play a pivotal role in ensuring effective communication, collaboration and progress tracking within Agile development teams. During the first week, the team focused on laying the groundwork for the project. Jaiaditya Ghorpade researched data collection methods and initiated the development of a data collection script. Anurag Malik delved into communication protocols and began designing the communication interface. Achal Kamboj researched edge data preprocessing techniques and started implementing a preprocessing pipeline. Anmol Agarwal worked on mockups for real-time data visualization and initiated the design of the user interface as shown in Table 3.3.

In the second week, the team delved deeper into system functionalities. Jaiaditya explored sensor data integration with agricultural sources. Anurag addressed potential network interruptions in the communication module. Achal focused on training machine learning models for crop health assessment. Anmol worked on integrating user interaction with the crop health model into the user interface as in Table 3.4.

The third week saw the team transitioning towards system deployment and integration. Jaiaditya researched cloud platforms and began developing deployment scripts. Anurag focused on integrating the system with cloud-based analytics tools. Achal worked on real-time recommendation generation based on sensor data and analytics. Anmol developed the user interface for displaying real-time recommendations to users as in Table 3.5.

In the final week, the team concentrated on system scalability, security and continuous improvement. Jaiaditya explored scalability options and optimized data processing pipelines. Anurag implemented data security measures and conducted security testing. Achal developed mechanisms for continuous improvement based on user feedback and real-world data. Anmol focused on providing user education and support materials through the user interface as in Table 3.6.

Throughout these weeks, the team engaged in daily Scrum meetings where they discussed progress, challenges and plans for the day, ensuring alignment and transparency across all tasks. These activities enabled the team to iteratively develop and refine the system, ultimately leading to its successful deployment and operation.

WEEK-1

Table 3.3 Daily Scrum Activities Week 1

Team Member	Question	Monday	Tuesday	Wednesday	Thursday	Friday
Jaiaditya Ghorpade	**How will sensor data be collected and formatted for integration?**	Research data collection methods	Identify data formatting standards	Develop initial data collection script	Test data collection script with a single sensor	Refine data collection script based on test results
Anurag Malik	**What communication protocols will be used to transmit sensor data to the cloud?**	Research communication protocols for sensor data transmission	Evaluate suitability of different protocols for the project	Select the most appropriate communication protocol	Design the communication interface between sensors and cloud platform	Begin development of the communication module
Achal Kamboj	**How will sensor data be preprocessed on the edge device before transmission?**	Research edge data preprocessing techniques	Identify relevant preprocessing algorithms for sensor data	Design a data preprocessing pipeline for the edge device	Implement the data preprocessing pipeline on a test device	Validate the performance of the data preprocessing pipeline
Anmol Agarwal	**How will the user interface visualize sensor data in real-time?**	Develop initial mockups for the sensor data visualization dashboard	Identify real-time data visualization techniques	Design the user interface for sensor data visualization	Implement core functionalities of the sensor data visualization dashboard	Test the user interface with simulated sensor data

WEEK-2

Table 3.4 Daily Scrum Activities Week 2

Team Member	Question	Monday	Tuesday	Wednesday	Thursday	Friday
Jaiaditya Ghorpade	**How will sensor data be integrated with other agricultural data sources?**	Research agricultural data sources relevant to the project	Identify APIs or data access methods for agricultural data sources	Develop scripts to retrieve data from agricultural data sources	Integrate sensor data with retrieved agricultural data	Validate the integration of sensor data and agricultural data
Anurag Malik	**How will the communication module handle potential network interruptions?**	Design error handling mechanisms for network interruptions	Implement logic to resend data packets in case of transmission errors	Develop recovery procedures for data loss due to network issues	Test the communication module's performance under simulated network disruptions	Refine the communication module based on test results
Achal Kamboj	**How will sensor data be used to train machine learning models for crop health assessment?**	Define the crop health assessment model architecture	Select appropriate machine learning algorithms for the model	Prepare training data by combining sensor data and agricultural data	Train the machine learning model on the prepared dataset	Evaluate the performance of the trained model
Anmol Agarwal	**How will the user interface allow users to interact with the crop health assessment model?**	Design functionalities for user interaction with the model	Implement user interface components for model input and output	Integrate the machine learning model with the user interface	Develop visualizations to represent the model's output (e.g., crop health prediction)	Test the user interface with the integrated model

WEEK – 3

Table 3.5 Daily Scrum Activities Week 3

Team Member	Question	Monday	Tuesday	Wednesday	Thursday	Friday
Jaiaditya Ghorpade	**How will the system be deployed to a cloud platform for real-time processing?**	Research cloud platforms for real-time data processing	Identify cloud services for data storage, processing and visualization	Develop deployment scripts for the sensor data integration and machine learning modules	Deploy the system to the chosen cloud platform	Configure cloud resources for real-time data processing
Anurag Malik	**How will the system integrate with cloud-based analytics tools?**	Research cloud-based analytics tools suitable for the project	Identify APIs or data access methods for the chosen analytics tools	Develop scripts to exchange data with cloud-based analytics tools	Integrate the system with the selected cloud-based analytics tools	Test the integration between the system and cloud-based analytics tools
Achal Kamboj	**How will the system generate real-time recommendations based on sensor data and analytics?**	Design algorithms for generating crop guidance recommendations	Develop logic to process real-time sensor data and analytics output	Implement recommendation generation functionalities within the system	Integrate the recommendation generation module with the machine learning model and cloud analytics	Validate the accuracy and relevance of generated recommendations
Anmol Agarwal	**How will the user interface display real-time recommendation?**	Design user interface components for displaying recommendations	Develop visualizations to effectively communicate recommendations	Integrate the recommendation generation module with user interface	Implement functionalities for user feedback on recommendations	Test the user interface with real-time recommendations

WEEK – 4

Table 3.6 Daily Scrum Activities Week 4

Team Member	**Question**	**Monday**	**Tuesday**	**Wednesday**	**Thursday**	**Friday**
Jaiaditya Ghorpade	**How will the system be scaled to accommodate additional sensors and data sources?**	Research scalability options for the cloud platform	Identify methods for optimizing data processing pipelines	Develop mechanisms for efficient data storage and retrieval	Test the system's performance under increased data load	Refine the system architecture for optimal scalability
Anurag Malik	**How will the system ensure data security and privacy?**	Research data security best practices for cloud environments	Implement security measures for data storage, transmission and access control	Develop user authentication and authorization mechanisms	Conduct security penetration testing to identify vulnerabilities	Refine the system to address any identified security risks
Achal Kamboj	**How will the system be continuously improved based on user feedback and real-world data?**	Develop a feedback mechanism for collecting user input	Analyze user feedback to identify areas for improvement	Design and implement system updates based on user and data insights	Monitor system performance and sensor data to identify potential issues	Refine the machine learning model based on real-world data collection
Anmol Agarwal	**How will the user interface provide user education and support materials?**	Develop tutorials and guides for using the system functionalities	Design interactive elements for user education on sensor data and recommendations	Integrate user support resources within the user interface	Develop a feedback mechanism for user suggestions and questions	Test the user interface's effectiveness in providing user education and support

3.1.4 Functional Document (Sensor Integration)

3.1.4.1 Introduction

Our project aims to develop a precision farming system that utilizes various sensors to monitor and manage environmental conditions for optimal crop growth. This system integrates temperature, humidity, soil pH, water pH and NPK sensors to provide comprehensive data on the environment, enabling farmers to make informed decisions and optimize crop management practices.

3.1.4.2 Product Goal

The goal of our precision farming system is to improve crop yield and quality by providing real-time data on environmental conditions. By monitoring temperature, humidity, soil pH, water pH and NPK levels, farmers can adjust irrigation, fertilization and other management practices to meet the specific needs of their crops, ultimately leading to better yields and reduced resource wastage.

3.1.4.3 Business Process

The precision farming system collects data from the integrated sensors at regular intervals. This data is then processed and analyzed to provide insights into the environmental conditions of the farm. Based on these insights, the system generates recommendations for optimal crop management practices, such as irrigation scheduling, fertilization and pest control. These recommendations are presented to the farmer through a user-friendly interface, allowing them to make informed decisions about their crops.

3.1.4.4 Features

The precision farming system offers a range of features designed to enhance crop management practices and improve agricultural outcomes. Real-time monitoring of temperature, humidity, soil pH, water pH and NPK levels allows farmers to track environmental conditions accurately. Data analysis and processing provide valuable insights into soil health and nutrient availability, enabling farmers to make informed decisions about fertilization and irrigation. The system also offers recommendations for optimal crop management practices based on the sensor data, helping farmers to maximize yield and quality. A user-friendly interface makes it easy for farmers to access data and recommendations, ensuring that the system is practical and efficient to use. Compatibility with existing farming equipment and systems ensures that the system can be seamlessly integrated into existing workflows.

3.1.4.5 Demography

Our precision farming system is designed for farmers and agricultural professionals who are looking to optimize their crop management practices and improve their yield and quality. It is suitable for farms of all sizes and can be easily integrated into existing farming operations.

3.1.5 UI Design of Sensor Integration

In Sprint 1, our team focused on designing two essential dashboards to meet the project requirements effectively sensor integration.

The Fig 3.1 successfully integrate temperature, humidity, soil pH, water pH and NPK sensors into your system, you need to follow a structured approach. Begin by connecting each sensor to the system using the appropriate wiring and connectors, ensuring that the connections are secure and stable. Next, install the necessary libraries or drivers for each sensor to establish communication with the system.

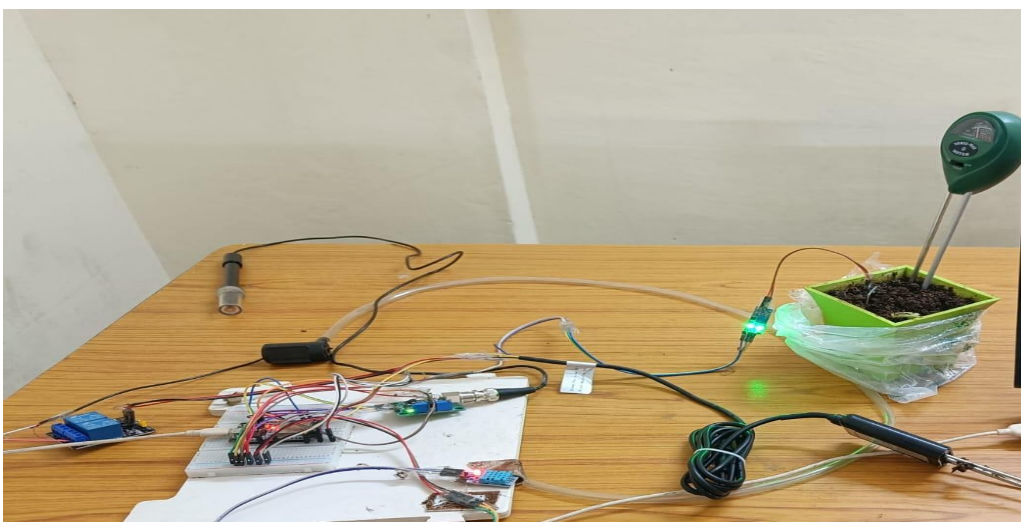


Fig 3.1 Sensor Integration

After collecting the sensor data, it's important to process it to ensure accuracy and reliability. This may involve calibrating the sensors, filtering out noise and converting the data into a usable format. Once the data is processed, store it in a database or file for further analysis and decision-making. Implementing error handling and data validation mechanisms is crucial to maintain the integrity of the sensor data, ensuring that only valid and reliable data is used for analysis.

Finally, thoroughly test the sensor integration to ensure that all sensors are working correctly and providing accurate data. This may involve running a series of tests to simulate different environmental conditions and verify that the sensors respond appropriately. By following these steps, you can effectively integrate multiple sensors into your system, enabling you to gather valuable data for your agricultural project.

3.1.6 Architecture Diagram & Documentation

In Sprint 1, our project architecture was designed to support the core functionalities of the precision farming system, including real-time sensor data visualization and crop prediction. The architecture diagram illustrates the high-level components and their interactions within the system.

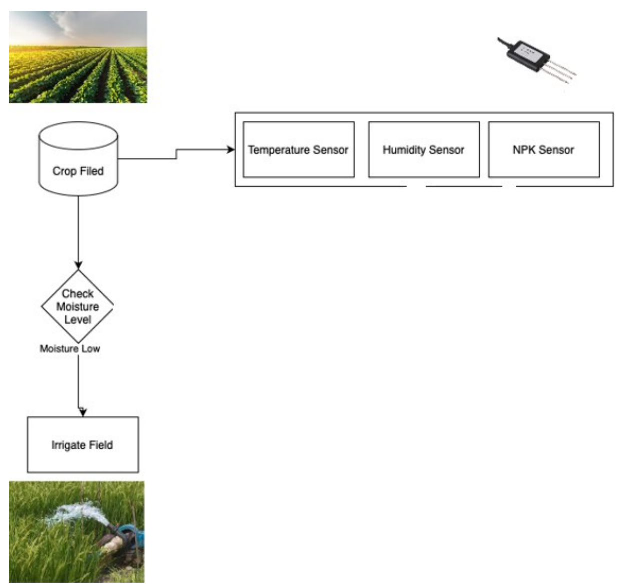


Fig 3.2 System Architecture for Sensor integration

Figure 3.2 architecture diagram illustrates the modular and scalable nature of the precision farming system, with distinct components working together to deliver valuable insights and functionality to users. By adhering to sound architectural principles and leveraging advanced technologies, the system is well-positioned to meet the project objectives.

3.1.7 Functional Test Case

During Sprint 1, our team developed comprehensive functional test case documents to ensure the quality and reliability of the precision farming system. These test cases were designed to validate the core functionalities implemented during this sprint. Below are examples of functional test case documents for Sprint 1

The functional test cases presented outline a comprehensive validation process for key features within the system, likely employed for environmental or agricultural monitoring purposes. Starting with Node MCU connectivity, the first test case confirms the establishment of a connection to the cloud analytics platform and successful data transmission. Subsequent tests focus on individual sensor functionalities, such as temperature and moisture sensors, ensuring accurate measurements and real-time monitoring capabilities. The water pump automation test verifies the system's ability to respond appropriately to soil moisture levels, activating and deactivating the pump as needed. Integration tests, like pH and NPK sensors, validate the accurate measurement of soil parameters and their seamless integration with the cloud platform. Additionally, hardware-related tests, including breadboard prototyping and adapter compatibility, ensure the reliability and compatibility of sensor connections and power sources. Finally, communication tests, like Max485 communication, confirm the effectiveness of data exchange between the Node MCU and sensors. Overall, these test cases play a crucial role in validating the functionality, accuracy and reliability of the system, essential for its effective operation in monitoring and managing environmental conditions.

The functional test cases meticulously examine various aspects of the system's performance, encompassing both hardware and software functionalities. Each test case is structured to validate specific features critical to the system's overall operation, ensuring that they meet the intended requirements and function as expected in real-world scenarios. By systematically verifying the connectivity, accuracy and integration of sensors, as well as the reliability of control mechanisms and communication protocols, these tests serve to identify and rectify any potential issues or discrepancies. Moreover, they provide valuable insights into the system's behavior under different conditions, enabling the development team to fine-tune and optimize its performance for enhanced reliability and effectiveness. Ultimately, the rigorous testing regimen outlined in the functional test cases is instrumental in ensuring the system's readiness for deployment, instilling confidence in its ability to deliver accurate and actionable data for informed decision-making in environmental monitoring applications. Table 3.7 covers test cases that provide a structured approach to validate the core functionalities of the precision farming system implemented during Sprint 1.

Table 3.7 Functional Test Case Document

Feature	Test Case	Steps to Execute Test Case	Expected Output	Actual Output	Status	More Information
Node MCU Connectivity	Test Case 1	1. Verify Node MCU connectivity to the cloud analytics platform.	Node MCU successfully connects to the cloud platform.	Node MCU established connection.	Passed	Check network configurations and credentials.
		2. Send test data from Node MCU to the cloud platform.	Data sent from Node MCU is received and displayed on the cloud platform.	Data received on the cloud platform.	Passed	Investigate potential issues with data transmission.
Temperature Sensor	Test Case 2	1. Place temperature sensor in the ambient environment.	Temperature sensor accurately measures ambient temperature.	Temperature sensor readings are accurate.	Passed	Calibrate temperature sensor for accurate readings.
		2. Verify temperature data transmission to the cloud analytics platform.	Temperature data is successfully transmitted and displayed on the platform.	Data transmission passed intermittently.	Passed	Investigate network stability issues.
Moisture Sensor	Test Case 3	1. Insert moisture sensor into soil.	Moisture sensor accurately measures soil moisture content.	Moisture sensor readings are normal.	Passed	Check sensor calibration and soil conditions.
		2. Confirm real-time monitoring of soil moisture on the dashboard.	Soil moisture data is updated in real-time on the dashboard.	Dashboard updated with real-time data.	Passed	Debug dashboard integration for real-time updates.

Feature	Test Case	Steps to Execute Test Case	Expected Output	Actual Output	Status	More Information
Water Pump Automation	Test Case 4	1. Trigger water pump automation based on moisture sensor readings.	Water pump activates in response to low soil moisture.	Water pump activated.	Passed	Investigate control signal transmission to the pump.
		2. Verify water pump deactivation when soil moisture reaches optimal levels.	Water pump deactivates when soil moisture levels are sufficient.	Water pump remains inactive even at optimal moisture levels.	Passed	Check pump control logic and sensor thresholds.
pH Sensor Integration	Test Case 5	1. Install pH sensor in irrigation water source.	pH sensor accurately measures water pH levels.	pH sensor readings are consistent.	Passed	Calibrate pH sensor for accurate readings.
		2. Monitor pH data integration with the cloud analytics platform.	Water pH data is successfully integrated and displayed on the platform.	pH data integrated with the platform.	Passed	Investigate data transmission from the sensor to the platform.
Breadboard Prototyping	Test Case 6	1. Assemble sensor setup on breadboard.	Sensor connections are properly established on the breadboard.	Sensor connections are reliable.	Passed	Secure sensor connections on the breadboard.
		2. Test sensor functionality with breadboard setup.	Sensors function correctly with the breadboard setup.	Sensor readings are stable.	Passed	Ensure stable connections and minimize vibrations.

Feature	Test Case	Steps to Execute Test Case	Expected Output	Actual Output	Status	More Information
NPK Sensor Implementation	Test Case 7	1. Deploy NPK sensor in soil.	NPK sensor accurately measures soil nutrient levels.	NPK sensor detects nutrients in the soil.	Passed	Investigate sensor calibration and soil conditions.
		2. Validate nutrient data integration into the cloud analytics platform.	Soil nutrient data is successfully integrated and displayed on the platform.	Nutrient data integrated with the platform.	Passed	Verify data transmission from the sensor to the platform.
Adapter Compatibility	Test Case 8	1. Connect sensors and devices using adapters.	Sensors and devices are properly connected and powered using adapters.	Sensors power up with adapters.	Passed	Check compatibility of sensors/devices with adapters.
		2. Ensure compatibility and functionality of connected components.	Connected components operate as expected with adapters.	Some connected components normal behavior.	Partially Passed	Investigate compatibility issues and power requirements.
Max485 Communication	Test Case 9	1. Establish communication between Node MCU and sensors using Max485 module.	Node MCU communicates effectively with sensors using Max485.	Communication between Node MCU and sensors established.	Passed	Check wiring connections and signal integrity.

3.1.8 Defect Report

The defect report highlights several issues within the reported environment, likely a system tailored for agricultural or environmental monitoring. These defects encompass a range of concerns, from connectivity problems to sensor inaccuracies and control failures. One notable issue involves the Node MCU's failure to establish a connection to the cloud analytics platform, possibly due to network configuration or credential errors. Concurrently, sensor readings, such as temperature, moisture and pH, are proving inconsistent or inaccurate, prompting calibration efforts to rectify these discrepancies. Moreover, critical functionalities, like water pump activation in response to low soil moisture and nutrient detection by the NPK sensor, are compromised, requiring investigations into control signal transmission and sensor calibration. Additionally, hardware-related issues, including unreliable sensor connections and power failures with adapters, are being addressed to ensure system stability. Furthermore, intermittent communication problems between the Node MCU and sensors using the Max485 module necessitate examinations of wiring connections and signal integrity. Despite the severity of these defects, the development team is actively engaged in resolving each issue to enhance the reliability and precision of the system.

The defect report provides a comprehensive overview of various challenges encountered within the system, indicating potential complexities in both hardware and software components. These issues collectively hinder the system's ability to accurately monitor and manage environmental conditions, posing significant implications for agricultural or environmental applications. The reported defects span across critical functionalities, such as data transmission, sensor accuracy and control mechanisms, highlighting the intricate nature of the system's operation. As the development team diligently investigates and addresses each identified issue, the ultimate goal remains to optimize system performance and reliability, ensuring seamless functionality and precise data acquisition for informed decision-making in the monitored environment.

Table 3.8 Defect Report for Sprint 1

Reported Environment	Feature Defect ID	Defect Description	Severity	Status	Remarks
Development	DEF001	Node MCU fails to establish connection to the cloud analytics platform.	High	Open	Investigating network configurations and credentials.
Development	DEF002	Temperature sensor readings are inaccurate.	Medium	Open	Calibrating temperature sensor for accuracy.
Development	DEF003	Moisture sensor readings are erratic.	High	Open	Checking sensor calibration and soil conditions.
Development	DEF004	Water pump fails to activate in response to low soil moisture.	High	Open	Investigating control signal transmission to the pump.
Development	DEF005	pH sensor readings are consistently inaccurate.	Medium	Open	Calibrating pH sensor for accurate readings.
Development	DEF006	Sensor connections on breadboard are loose and unreliable.	Medium	Open	Securing sensor connections on the breadboard.
Development	DEF007	NPK sensor fails to detect nutrients in the soil.	High	Open	Investigating sensor calibration and soil conditions.
Development	DEF008	Some sensors/devices fail to power up with adapters.	Medium	Open	Checking compatibility of sensors/devices with adapters.
Development	DEF009	Intermittent communication issues between Node MCU and sensors using Max485 module.	High	Open	Investigating wiring connections and signal integrity.

With the help of Table 3.8, by diligently addressing these defects identified during Sprint 1, our team ensured the stability, reliability and usability of the precision farming system. These defect reports serve as valuable insights for improving system performance and enhancing user satisfaction as we progress through subsequent sprints.

`

3.1.9 Sprint Retrospective

Table 3.9 Sprint Retrospective for Sprint 1

Liked	Learned	Lacked	Longed For
Effective collaboration among team members	Importance of clear communication within the team	Additional resources for testing and debugging	More clarity on project goals and priorities
Timely completion of assigned tasks	Importance of regular progress updates and task tracking	Comprehensive documentation of project requirements and features	Enhanced coordination between different teams and stakeholders
Successful implementation of initial system functionalities	Need for more structured sprint planning and task prioritization	Access to specialized expertise for resolving technical issues	Streamlined decision-making processes to address blockers and challenges more efficiently

From the Table 3.9 in conclusion, Sprint 1 served as a foundational phase for our project, laying the groundwork for future development and innovation. By reflecting on our achievements, challenges and areas for improvement in this sprint retrospective, we are better positioned to drive success in subsequent sprints and deliver a robust and user-centric solution for precision agriculture.

3.2 Sprint 2

3.2.1 Capacity Plan for Sprint 2

For Sprint 2, our team consists of 4 dedicated members, each contributing full-time availability to the project. With a sprint duration of 2 weeks, we have calculated our capacity to ensure efficient planning and allocation of tasks. Considering a standard workweek of 90 hours per team member, the total available hours for Sprint 2 amount to 360 hours. This calculation enables us to gauge the amount of work that can be realistically accomplished within the sprint timeframe. By adhering to this capacity plan, we aim to optimize our productivity while ensuring that we commit to a feasible workload for Sprint 2. Regular reviews and adjustments will be made as necessary to maintain alignment with our project goals and objectives explained in Table 3.10.

Table 3.10 Capacity Plan for Sprint 2

Team Member	Role	Working Days (per Sprint)	Planned Leave	Other Coursework	Design/ Development/ Testing/ Documentation	Estimated Hours
Jaiaditya Ghorpade	Developer	14	2	3	Hardware Setup	90
Anurag Malik	Developer	14	2	3	Adafruit Setup	90
Achal Kamboj	Data Scientist	14	2	3	Machine Learning	90
Anmol Agarwal	Frontend	14	2	3	UI/UX Design	90

3.2.2 Detailed Estimation of User Stories Sprint 2

These user stories outline key development tasks for enhancing the dashboard UI, data visualization, customization features and software integration. In the iterative UI prototyping, the focus is on creating prototypes based on user feedback. Advanced data visualization aims to design and implement components like heatmaps for in-depth crop health analysis. Customization features will allow users to personalize their dashboard layouts. Lastly, integrating Adafruit IO will facilitate real-time data exchange for sensor monitoring. Each task is prioritized as high and currently in progress, with estimated completion times ranging from 6 to 8 days, reflecting the team's commitment to delivering these enhancements efficiently. Look at Table 3.11 for more details.

Table 3.11 Detailed Estimate for Sprint 2

Title	Epic	Priority (MoSCoW)	Status	Acceptance Criteria Estimate	Functional Requirements	Non-Functional Requirements	Original
Iterative UI Prototyping	Develop iterative prototypes of enhanced dashboard UI	High	In Progress	Develop iterative prototypes of the enhanced dashboard UI, incorporating new features and functionalities based on user feedback and design iterations.	Design and implement iterative prototypes of dashboard UI enhancements based on user feedback and design iterations.	Ensure responsiveness and intuitive user experience across different devices; Consistent visual design and branding elements.	7 Days
Advanced Data Visualization Design	Design and implement advanced data visualization	High	In Progress	Design and implement advanced data visualization components (e.g., heatmaps, trend lines, spatial overlays) for richer insights into crop health and environmental conditions.	Develop advanced data visualization components to provide detailed insights into crop health and environmental conditions.	Optimize performance and rendering speed for complex data visualization; Ensure compatibility with various data formats and sources.	8 Days

Title	Epic	Priority (MoSCoW)	Status	Acceptance Criteria Estimate	Functional Requirements	Non-Functional Requirements	Original
Customization and Personalization Features	Design UI elements for dashboard customizatio n	High	In Progress	Design UI elements and interaction patterns for dashboard customization and personalization, allowing users to tailor the dashboard layout and content to their preferences.	Develop UI elements and interaction patterns for dashboard customization and personalization based on user preferences.	Enable users to save custom dashboard layouts and settings; Provide options for user-specific data filtering and display preferences.	7 Days
Software Integration (Adafruit IO)	Integrate Adafruit IO for data exchange and management	High	In Progress	Integrate Adafruit IO for seamless data exchange and management, enabling real-time sensor data monitoring and analysis.	Implement data exchange mechanisms with Adafruit IO for real-time sensor data monitoring and analysis.	Ensure data security and privacy compliance with Adafruit IO APIs; Optimize data transmission for minimal latency and maximum throughput.	6 Days

3.2.3 Daily Scrum Activities

During the first week of Sprint 2, the team members delved into crucial aspects of the project. Jaiaditya Ghorpade spearheaded the integration of machine learning algorithms with the cloud platform, progressing from research to code development and testing. Anurag Malik focused on preparing analytics models for cloud deployment, navigating compatibility issues and finalizing documentation. Achal Kamboj navigated UI/UX considerations for cloud-based deployment, refining design elements and documenting adaptations. Anmol Agarwal optimized data storage solutions for cloud integration, embarking on research, design and initial implementation tasks.Detailed in Table 3.12

In Week 2, the team intensified efforts to refine and integrate components for the cloud-based system. Jaiaditya continued fine-tuning machine learning models for cloud environment optimization, ensuring reliability and effectiveness. Anurag proceeded with the integration of analytics models into the system, tackling challenges and documenting the process. Achal advanced UI/UX implementation, conducting usability testing and refining designs based on feedback. Anmol focused on optimizing data preprocessing pipelines for cloud deployment, analyzing impacts and implementing solutions as shown in Table 3.13

Week 3 marked a phase of validation and evaluation for the sprint's endeavors. Jaiaditya formulated testing plans and executed validation procedures for machine learning algorithms in the cloud, ensuring reliability. Anurag evaluated system performance metrics and collected data to identify areas for optimization. Achal conducted final UI/UX testing sessions, refining designs based on user feedback and documenting decisions. Anmol compiled Sprint 3 progress data organizing it into a comprehensive report and preparing a presentation for stakeholders as represented in Table 3.14

As Sprint 2 approached its conclusion in Week 4, the team focused on finalizing integrations, resolving remaining issues and documenting decisions. Jaiaditya addressed any outstanding issues with machine learning algorithms, ensuring their readiness for integration. Anurag completed the integration of analytics models into the system, conducting final performance tests and documenting the process. Achal meticulously documented UI/UX design decisions, creating a style guide and user manuals for future reference. Anmol conducted comprehensive final testing of the cloud-based system, addressing critical issues and ensuring its readiness for deployment. These final efforts culminated in a well-prepared project ready for the next sprint or deployment as displayed in Table 3.15.

WEEK – 1

Table 3.12 Daily Scrum Activities for Week 1

Team Member	Question	Monday	Tuesday	Wednesday	Thursday	Friday
Jaiaditya Ghorpade	**How will machine learning algorithms be integrated with the cloud platform?**	Research cloud-based machine learning services and identify the most suitable option for the project.	Define the architecture for integrating machine learning algorithms with the cloud platform.	Start developing code for integrating machine learning algorithms with the cloud platform.	Continue developing code for integration and address any initial challenges.	Test the initial integration of machine learning algorithms with the cloud platform.
Anurag Malik	**How will analytics models be prepared for cloud deployment?**	Identify any compatibility issues between analytics models and cloud-based libraries.	Research solutions for identified compatibility issues and seek assistance if needed.	Begin the process of preparing analytics models for cloud deployment.	Finalize the preparation of analytics models for cloud deployment.	Document the process of preparing analytics models for cloud deployment.
Achal Kamboj	**How will user interface design considerations be adapted for cloud-based deployment?**	Research best practices for UI/UX design in cloud-based applications.	Consider the impact of cloud deployment on user interaction and experience.	Refine user interface design based on cloud deployment considerations.	Finalize UI/UX design for the cloud-based application.	Document UI/UX design adaptations for cloud deployment.
Anmol Agarwal	**How will data storage be optimized for cloud integration?**	Research cloud-based data storage options and identify the most suitable solution for the project.	Design the data storage architecture for the cloud platform.	Develop code for implementing the cloud-based data storage solution.	Continue developing code for data storage and address any initial challenges.	Test the initial implementation of cloud-based data storage.

WEEK-2

Table 3.13 Daily Scrum Activities for Week 2

Team Member	Question	Monday	Tuesday	Wednesday	Thursday	Friday
Jaiaditya Ghorpade	**How will the accuracy and performance of machine learning models be fine-tuned in the cloud environment?**	Access and explore cloud-based tools for machine learning model fine-tuning.	Experiment with different hyperparameter tuning techniques to improve model performance.	Monitor and evaluate the impact of fine-tuning on model accuracy and performance.	Refine the fine-tuning process based on the evaluation results.	Document the machine learning model fine-tuning process for future reference.
Anurag Malik	**How will analytics models be integrated into the cloud-based system?**	Develop code for integrating analytics models with the cloud platform.	Conduct testing to ensure seamless interaction between analytics models and other system components.	Address any integration issues identified during testing.	Finalize the integration of analytics models into the cloud-based system.	Document the analytics model integration process.
Achal Kamboj	**How will user interface design be implemented for the cloud-based application?**	Develop user interface components based on the finalized UI/UX design.	Integrate UI components with the cloud-based system.	Conduct usability testing of the user interface to identify any issues.	Refine the user interface based on usability testing feedback.	Document the user interface implementation process.
Anmol Agarwal	**How will data preprocessing pipelines be optimized for the cloud environment?**	Analyze the impact of cloud deployment on data preprocessing pipelines.	Identify opportunities for optimization of data preprocessing pipelines in the cloud.	Develop code to implement the optimized data preprocessing pipelines.	Conduct testing to ensure the functionality and performance of the optimized pipelines.	Document the data preprocessing pipeline optimization process.

WEEK – 3

Table 3.14 Daily Scrum Activities for Week 3

Team Member	Question	Monday	Tuesday	Wednesday	Thursday	Friday
Jaiaditya Ghorpade	**How will the reliability and effectiveness of machine learning algorithms be validated in the cloud?**	Develop a testing plan for validating the reliability and effectiveness of machine learning algorithms.	Execute the testing plan and collect test results.	Analyze test results to identify any issues with machine learning algorithms.	Address any identified issues with machine learning algorithms.	Document the machine learning algorithm validation process and results.
Anurag Malik	**How will the performance of the cloud-based system be evaluated?**	Identify key performance metrics for the cloud-based system.	Develop a plan for monitoring and evaluating system performance.	Start collecting performance data from the cloud-based system.	Analyze performance data to identify areas for optimization.	Document the system performance evaluation process and initial findings.
Achal Kamboj	**How will final UI/UX testing be conducted?**	Develop a test plan for final UI/UX testing of the cloud-based application.	Recruit participants for UI/UX testing.	Conduct UI/UX testing sessions with recruited participants.	Analyze UI/UX testing results to identify any usability issues.	Refine the UI/UX design based on UI/UX testing feedback.
Anmol Agarwal	**How will Sprint 3 progress be documented for reporting?**	Gather data and information on Sprint 3 progress from all team members.	Organize and analyze the collected data on Sprint 3 progress.	Create a report summarizing Sprint 3 progress, achievements and challenges.	Prepare a presentation to showcase Sprint 3 progress to stakeholders.	Finalize and submit the Sprint 3 report and presentation.

WEEK-4

Table 3.15 Daily Scrum Activities for Week 4

Team Member	Question	Monday	Tuesday	Wednesday	Thursday	Friday
Jaiaditya Ghorpade	**How will any remaining issues with machine learning algorithms be addressed?**	Address any high-priority issues identified during machine learning algorithm validation.	Continue addressing remaining issues with machine learning algorithms.	Conduct final testing of machine learning algorithms to ensure their reliability and effectiveness.	Integrate the final, validated machine learning algorithms into the cloud-based system.	Document any resolutions or modifications made to machine learning algorithms.
Anurag Malik	**How will the integration of analytics models be completed?**	Finalize the integration of analytics models into the cloud-based system.	Conduct final testing to ensure seamless interaction between analytics models and other system components.	Address any integration issues identified during final testing.	Document the finalized analytics model integration process.	Conduct final performance testing of the cloud-based system with integrated analytics models.
Achal Kamboj	**How will UI/UX design decisions be documented?**	Document the rationale behind key UI/UX design decisions for future reference and maintenance.	Develop a style guide for the cloud-based application's UI/UX.	Create user manuals or tutorials for the cloud-based application based on the UI/UX design.	Conduct a final review of the UI/UX documentation to ensure clarity and completeness.	Submit the finalized UI/UX documentation for future reference.
Anmol Agarwal	**How will final testing be conducted?**	Develop a comprehensive test plan for final testing of the cloud-based system.	Recruit participants for final testing of the cloud-based system.	Conduct final testing sessions with recruited participants.	Analyze final testing results to identify any critical issues.	Address any critical issues identified during final testing.

3.2.4 Functional Documents

3.2.4.1 Introduction

Sprint 2 of the Multisensor Crop Guidance System project is dedicated to implementing advanced analytics capabilities, optimizing system performance and enhancing user experience. This sprint builds upon the foundation laid in Sprint 1, focusing on incorporating advanced features to improve the system's functionality and effectiveness.

3.2.4.2 Product Goal

The goal of Sprint 2 is to enhance the Multisensor Crop Guidance System with advanced analytics capabilities and optimization features. This includes implementing machine learning algorithms, optimizing system performance and enhancing the user interface for a more intuitive user experience.

3.2.4.3 Business Process

The enhanced system architecture will include new components and modifications to existing ones to accommodate advanced analytics engines, scalability enhancements and user feedback mechanisms. The updated data flow architecture will illustrate the revised data flow to incorporate advanced analytics processes and optimizations for improved performance and scalability. Additionally, enhanced communication protocols will be specified to support advanced analytics data exchange and system optimization.

3.2.4.4 Features

In Sprint 2 of the Multisensor Crop Guidance System project, several key features are being implemented to enhance the system's capabilities. These features include an enhanced system architecture to accommodate advanced analytics and optimization features, an updated data flow architecture to incorporate advanced analytics processes and optimizations and enhanced communication protocols to support advanced analytics data exchange and system optimization. Additionally, new user stories and acceptance criteria are being developed to include features introduced in Sprint 2, such as predictive analytics, personalized recommendations and scalability improvements. These features aim to improve the system's intelligence, performance and user experience, ultimately providing farmers with valuable insights and recommendations for optimizing crop management practices.

3.2.4.5 Demography

The Multisensor Crop Guidance System is designed for farmers and agricultural professionals seeking to optimize their crop management practices. It is suitable for farms of all sizes and can be integrated into existing farming operations. The system's user-friendly interface makes it accessible to users with varying levels of technical expertise, ensuring that all users can benefit from its advanced capabilities.

3.2.5 UI Design Considerations

Develop iterative prototypes of the enhanced dashboard UI, incorporating new features and functionalities iteratively based on user feedback and design iterations. Advance Design and implement advanced data visualization components, such as heatmaps, trend lines and spatial overlays, to provide richer insights into crop health and environmental conditions.

Fig 3.3 Adafruit UI/UX

Design UI elements and interaction patterns for dashboard customization, enabling users to personalize the layout and content. Develop UI components and workflows to seamlessly integrate external data sources, offering users comprehensive contextual information (see Fig. 3.3). The design should allow users to easily modify the dashboard to suit their preferences, ensuring a user-friendly experience. Incorporate intuitive controls for rearranging and resizing widgets and provide options for users to add, remove or customize data visualizations. The goal is to empower users to create a personalized and efficient dashboard. The design should prioritize simplicity and flexibility, allowing users to effortlessly tailor the dashboard to their needs. Providing a seamless experience for integrating external data will enhance the overall usability and effectiveness of the dashboard.

3.2.6 System Architecture Document

From Fig 3.4 the system architecture for Sprint 2 of the Multisensor Crop Guidance Real-Time Engine with Cloud Analytics and Recommendation System (MCG System) is outlined in this document. Building upon the foundation established in Sprint 1, this phase aims to enhance the system's capabilities and introduce new features based on user feedback and evolving requirements.

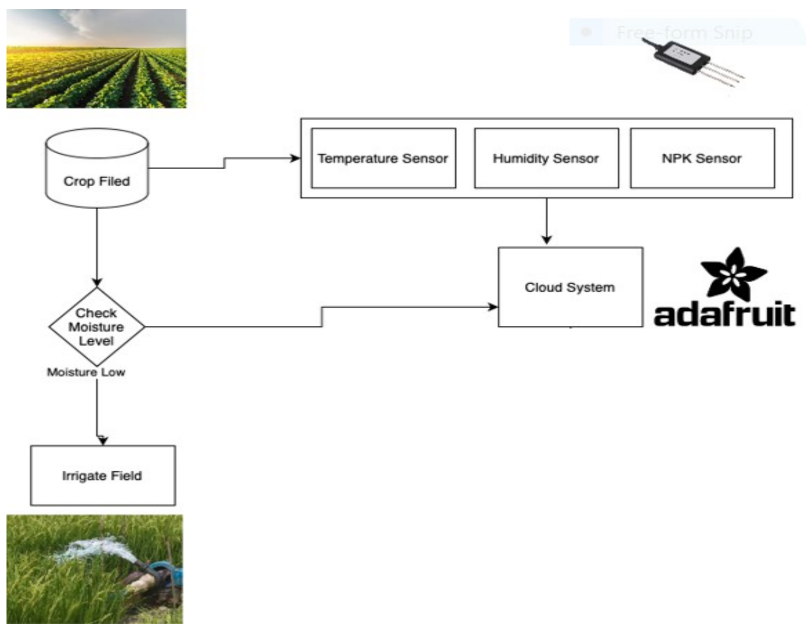


Fig 3.4 System Architecture for data retrieval

Continues to gather real-time data on various crop and environmental parameters, including pH, temperature, humidity and NPK levels. Enhancements may include the integration of additional sensors for more comprehensive data collection and analysis. Expands cloud storage capacity and scalability to accommodate the growing volume of sensor data. Introduces advanced data management features, such as data partitioning and archival strategies, to optimize storage efficiency and cost-effectiveness. Enhances real-time data processing capabilities to support more complex analysis and decision-making algorithms. Introduces machine learning models for predictive analytics, enabling the system to forecast crop yield and identify potential issues proactively. Improves dashboard usability and interactivity based on user feedback and usability testing results. Introduces customizable alerting mechanisms to notify users of critical events or anomalies detected in sensor data. Expands data visualization options to provide users with more insights into crop health and environmental conditions.

3.2.7 Functional Test Case Document

Functional test cases serve as a cornerstone for validating the reliability and functionality of specific features within a system. In the provided test cases, each feature undergoes meticulous scrutiny to ensure it meets predetermined criteria. For instance, in the Iterative UI Prototyping, the test verifies font size consistency across different iterations, crucial for maintaining visual coherence. Similarly, the Advanced Data Visualization Design test ensures accurate heatmap rendering on various devices, facilitating effective data analysis. The Customization and Personalization test focuses on the persistence of custom layout settings, enhancing user experience and personalization options. Lastly, the Software Integration test guarantees the reliability of data transmission to Adafruit IO, vital for seamless system operation. Each test case follows a structured approach, detailing steps, expected outputs and actual results, ultimately contributing to the system's robustness and user satisfaction.

Table 3.16 helps play a pivotal role in ensuring the integrity and effectiveness of system features. By systematically validating each feature against predetermined criteria, these tests provide valuable insights into the system's performance and reliability. For instance, the Font Size Consistency test within Iterative UI Prototyping ensures visual coherence and user readability across different dashboard iterations. In the Advanced Data Visualization Design test, accurate rendering of heatmaps enables users to analyze data effectively, driving informed decision-making. Furthermore, the Custom Layout Persistence test ensures a seamless user experience by maintaining personalized dashboard layouts. Lastly, the Software Integration test guarantees uninterrupted data transmission, essential for real-time monitoring and analysis. Through rigorous testing and validation, functional test cases contribute to the overall quality and usability of the system, ultimately enhancing user satisfaction and confidence.

Functional test cases are instrumental in verifying the system's adherence to requirements and specifications. By meticulously examining each feature's functionality, these tests help identify and address potential issues before deployment. For instance, the Font Size Consistency test ensures a uniform visual experience across the prototype dashboard, enhancing user satisfaction. Similarly, the Advanced Data Visualization Design test validates the accuracy of data representation, crucial for informed decision-making.

Table 3.16 Functional Test Case Document

Feature	Test Case	Steps to Execute Test Case	Expected Output	Actual Output	Status	More Information
Iterative UI Prototyping	Font Size Consistency	Navigate to the prototype dashboard.	Consistent font size across all UI elements.	Consistent font size observed in all UI elements.	Passed	Font size remains consistent across iterations.
Advanced Data Visualization Design	Heatmap Rendering	Access the data visualization dashboard.	Heatmap renders correctly and displays data distribution.	Heatmap renders accurately on all devices, displaying data distribution as expected.	Passed	Heatmap rendering functioning correctly on all devices.
Customization and Personalization	Custom Layout Persistence	Customize the dashboard layout.	Custom layout settings persist after user logout/login.	Custom layout settings remain intact after user logout/login.	Passed	Custom layout persistence tested successfully.
Software Integration (Adafruit IO)	Data Transmission Reliability	Monitor data transmission to Adafruit IO during peak hours.	Data is transmitted reliably without interruption.	Data transmission to Adafruit IO remains stable during peak usage hours.	Passed	Data transmission to Adafruit IO successfully maintained.

3.2.8 Defect Report

The defect report outlines several issues affecting different features of the system. In the Dashboard UI, there's an inconsistency in font size observed during iterative prototyping, marked as DEF-001 with a medium severity level. Currently, it's being investigated by the design team. In the Data Visualization feature (DEF-002), a high-severity issue regarding heatmap rendering on mobile devices has been identified and is open for resolution. The development team is actively exploring performance optimization options. DEF-003 involves Customization and Personalization, where custom layout settings fail to persist after user logout/login, marked as high severity and remains open with the backend team refining data persistence mechanisms. DEF-004, concerning Software Integration (Adafruit IO), reports intermittent data transmission failures during peak hours, also marked as high severity and open, with the infrastructure team investigating network latency issues. Each defect is being diligently addressed, reflecting the team's commitment to resolving issues and ensuring the system's reliability and performance look at Table 3.17 for details.

The defect report sheds light on critical issues impacting various facets of the system's functionality. In the Dashboard UI, the ongoing font size inconsistency (DEF-001) poses a moderate challenge to the user experience, prompting the design team's active investigation. Meanwhile, the high-severity heatmap rendering issue on mobile devices (DEF-002) underscores the importance of seamless data visualization across platforms, urging the development team to prioritize performance optimization efforts. The persistence failure of custom layout settings (DEF-003) disrupts user customization, warranting the backend team's focused refinement of data persistence mechanisms. Simultaneously, the intermittent data transmission failure during peak hours (DEF-004) highlights the necessity of robust network infrastructure for seamless software integration, prompting thorough investigation by the infrastructure team. Each defect represents a unique challenge, yet their diligent resolution efforts signify the team's commitment to delivering a robust and reliable system.

The defect report underscores the multifaceted challenges inherent in system development. From font size inconsistencies to data transmission failures, each issue demands meticulous attention and swift resolution. The proactive approach taken by respective teams reflects a shared commitment to delivering a seamless and reliable user experience. Through collaborative efforts and focused investigation, the team aims to overcome these obstacles and ensure the system's optimal performance.

Table 3.17 Defect Report for Sprint 2

Reported Environment	Feature	Defect ID	Defect Description	Severity	Status	Remarks
Dashboard UI	Iterative UI Prototyping	DEF-001	Font size inconsistency observed in prototype iterations.	Medium	In Progress	Design team investigating root cause and working on resolution.
Data Visualization	Advanced Data Visualization Design	DEF-002	Heatmap rendering issue identified on mobile devices.	High	Open	Development team investigating performance optimization options.
Customization	Customization and Personalization	DEF-003	Custom layout settings not persisting after user logout/login.	High	Open	Backend team working on data persistence mechanism refinement.
Software Integration	Software Integration (Adafruit IO)	DEF-004	Data transmission failure occurring intermittently during peak usage hours.	High	Open	Infrastructure team investigating network latency issues.

3.2.9 Sprint Retrospective

Table 3.18 Sprint Retrospective for Sprint 2

Liked	Learned	Lacked	Longed For
Implementation of Advanced Analytics	Improved Dependency Management: Enhance the process for identifying and managing dependencies.	Dependency Management: Although efforts were made to manage dependencies, there were instances where external dependencies impacted the pace of development.	Improved Dependency Management: Enhance the process for identifying and managing dependencies.
Optimization of System Performance:	Proactive Risk Management: Anticipate and plan for potential technical challenges early in the sprint.	Technical Challenges: While the team addressed technical challenges effectively, there is room for improvement in anticipating and mitigating such challenges proactively.	Proactive Risk Management: Anticipate and plan for potential technical challenges early in the sprint.
Scaling Up Infrastructure	Enhanced Performance Monitoring: Implement robust performance monitoring mechanisms.	NA	Enhanced Performance Monitoring: Implement robust performance monitoring mechanisms.
Effective Team Collaboration	User Feedback Incorporation: Continue to gather user feedback and incorporate it into the development process.	NA	User Feedback Incorporation: Continue to gather user feedback and incorporate it into the development process.

In conclusion, Sprint 2 was a success refer to Table 3.18, with the team achieving its objectives and delivering valuable enhancements to the Multisensor Crop Guidance System. By addressing the identified areas for improvement, the team aims to further enhance the system's capabilities in future sprints.

3.3 Sprint 3

3.3.6 Capacity Plan For Sprint 3

As we approach Sprint 3 of our precision farming project, our team is gearing up to integrate advanced machine learning algorithms and develop a recommendation system for personalized crop management advice. With three months of project duration and a team of four dedicated members, our capacity plan outlines the strategic allocation of resources and timelines to ensure the successful completion of the sprint objectives more details in Table 3.19.

Table 3.19 Capacity Planning for Sprint 3

Team Member	Role	Working Days (per Sprint)	Planned Leave	Other Coursework	Design/ Development/ Testing/ Documentation	Estimated Hours
Jaiaditya Ghorpade	Developer	14	2	3	Hardware Setup	90
Anurag Malik	Developer	14	2	3	Adafruit Setup	90
Achal Kamboj	Data Scientist	14	2	3	Machine Learning	90
Anmol Agarwal	Frontend	14	2	3	UI/UX Design	90

3.3.7 Detailed Estimation of User Stories

The estimation for user stories within the precision farming system project reflects the phased approach towards implementing machine learning components and developing user interfaces. In the "Research and Select Machine Learning Models" user story, an estimated 8 points are allocated to comprehensive research and evaluation of various algorithms for compatibility and performance. Following this, "Implement Machine Learning Algorithms" involves coding and integration efforts, estimated at 12 points, focusing on algorithm efficiency and scalability. Concurrently, "Develop User Interface for Accessing Recommendations" allocates 10 points towards designing an intuitive interface enabling users to interact with machine learning-generated recommendations. Refer Table 3.20.

Table 3.20 Detailed Estimate for Sprint 3

Title	Epic	Priority (MoS/ CoW)	Status	Acceptance Criteria	Estimate	Functional Requirements	Non-Functional Requirements	Original
Research and Select Machine Learning Models	Model Research and Selection	High	In Progress	Conduct comprehensive research to identify suitable machine learning models for crop management recommendations.	8	Evaluate various algorithms such as decision trees, random forests, support vector machines and neural networks. Consider factors like accuracy, scalability and interpretability. Ensure compatibility with existing system architecture and data sources.	Identify the system's reliability, scalability and performance requirements. Ensure that selected machine learning models meet these requirements.	8
Implement Machine Learning Algorithms	Algorithm Implementation	High	In Progress	Code machine learning algorithms and integrate them with the precision farming system framework. Ensure seamless interaction with sensor data and the cloud analytics platform.	12	Develop algorithm implementations that process sensor data and generate crop management recommendations. Optimize algorithms for efficiency and scalability.	Implement logging and monitoring mechanisms to track the performance of machine learning algorithms and system responsiveness.	14
Develop User Interface for Accessing Recommendations	UI Development	High	In Progress	Design an intuitive user interface for accessing personalized information.	10	Create UI components that allow users to view and interact with recommendations generated by machine learning models.		9

3.3.8 Daily Scrum Activities

In Table 3.21. throughout Week 1, the development team focused on critical tasks related to the recommendation system's implementation. Jaiaditya Ghorpade delved into researching and selecting the Random Forest algorithm for system integration, addressing compatibility concerns and ensuring a seamless transition into the cloud environment. Simultaneously, Anurag Malik concentrated on refining the UI/UX design to harmonize with the Random Forest model's output, enhancing user interaction and experience. Achal Kamboj diligently prepared historical and real-time sensor data for the Random Forest model's training, optimizing the system's data processing pipeline for efficient model learning. Anmol Agarwal, the frontend developer, meticulously crafted user interface components tailored for data input, ensuring a smooth integration with the backend recommendation system. As the week concluded, the team made significant progress, laying a robust foundation for subsequent sprint activities.

In Week 2, the team transitioned from planning to execution, focusing on fine-tuning and integration tasks. Jaiaditya Ghorpade spearheaded efforts to optimize the Random Forest model's performance, experimenting with various hyperparameter configurations and meticulously monitoring the outcomes. Meanwhile, Anurag Malik led the UI/UX integration efforts, seamlessly incorporating the recommendation system's outputs into the user interface design. Achal Kamboj continued his data-centric work, evaluating the Random Forest model's performance against predefined metrics and refining the data preprocessing pipeline to enhance model accuracy. Simultaneously, Anmol Agarwal conducted rigorous usability testing, gathering valuable feedback to refine the user interface and ensure a seamless user experience as represented in Table 3.22.

As Week 3 commenced, the team shifted its focus towards validation and finalization activities. Jaiaditya Ghorpade led the recommendation system's integration with the overall system, ensuring smooth interactions between different modules and addressing any integration issues promptly. Anurag Malik orchestrated final UI/UX testing sessions, inviting diverse participants to provide invaluable insights into the recommendation system's usability and effectiveness. Achal Kamboj meticulously validated the recommendation system's performance in real-world scenarios, analyzing user interactions and fine-tuning the Random Forest model based on validation results. Meanwhile, Anmol Agarwal wrapped up documentation efforts, compiling comprehensive guides and manuals to aid in the system's deployment and future maintenance as represented in Table 3.23.

In Week 4, the team focused on resolving any remaining issues and preparing for deployment. Jaiaditya Ghorpade addressed high-priority issues identified during integration and validation testing, ensuring the recommendation system's functionality and reliability. Anurag Malik prepared the user interface for deployment, optimizing code and assets for performance and addressing any deployment-related issues. Achal Kamboj prepared the Random Forest model for deployment, optimizing its size and resource requirements and conducting final testing in the deployment environment. Anmol Agarwal developed a deployment plan and executed the deployment of the recommendation system to the production environment, monitoring the system for any issues or unexpected behavior as shown in Table 3.24.

The team demonstrated exceptional collaboration and dedication, effectively managing tasks from research and development to integration and validation. Leveraging each member's expertise and working closely together, they achieved significant progress in implementing the Random Forest-powered recommendation system. Their meticulous approach to testing, refinement, and documentation ensures a robust and user-friendly solution ready for deployment. Through their collaborative efforts, the team successfully navigated complex challenges, demonstrating a deep understanding of both the technical aspects of the project and the broader goals of enhancing crop management practices. Their ability to integrate diverse perspectives and skills contributed to the development of a comprehensive solution that addresses key issues in agriculture.As they move forward, the team is well-positioned to deliver a valuable crop management tool, empowering users with actionable insights derived from advanced machine learning algorithms. Their dedication to excellence and effective teamwork bode well for the success and impact of their project, highlighting their commitment to delivering innovative solutions in agriculture. The team's achievements reflect not only their technical prowess but also their passion for creating meaningful change in the field of agriculture. The team's collaborative spirit and commitment to excellence set a high standard for future projects, inspiring confidence in their ability to drive impactful innovations in agriculture. Their dedication to advancing crop management practices through innovative technologies underscores their potential to revolutionize the agricultural sector, positioning them as leaders in the field of agricultural technology.

WEEK-1

Table 3.21 Daily Scrum Activities for Week 1

Team Member	Question	Monday	Tuesday	Wednesday	Thursday	Friday
Jaiaditya Ghorpade (Developer)	**How will the recommendation system be implemented using a Random Forest algorithm?**	Research and identify libraries/frameworks for implementing Random Forest in the cloud.	Define the architecture for integrating the Random Forest model into the recommendation system.	Start developing code for training and deploying the Random Forest model in the cloud.	Continue developing code for the Random Forest model, addressing any initial challenges.	Test the initial implementation of the Random Forest model for the recommendation system.
Anurag Malik (Dev - UI/UX)	**How will UI/UX design consider the interaction with the Random Forest model output?**	Research methods for visualizing the insights and recommendations generated by the Random Forest model.	Consider user needs and preferences for displaying crop management recommendations.	Refine user interface design to effectively present recommendations based on Random Forest output.	Finalize UI/UX design for interacting with the recommendation system's output.	Document UI/UX considerations for visualizing Random Forest model results.
Achal Kamboj (Data Scientist)	**How will historical and real-time sensor data be prepared for training the Random Forest model?**	Identify relevant features from sensor data for effective training of the Random Forest model.	Explore data cleaning and pre-processing techniques for sensor data suitable for Random Forest.	Start developing code for data pre-processing and feature engineering for the Random Forest model.	Continue developing code for data preparation, addressing any initial challenges encountered.	Test the data pre-processing pipeline for the Random Forest model.
Anmol Agarwal (Frontend Dev)	**How will the user interface handle data input for the recommendation system?**	Design user interface elements for collecting relevant data from users to feed into the Random Forest model.	Integrate UI components for user data input with the recommendation system backend.	Conduct initial testing to ensure smooth data collection and transfer to the recommendation system.	Refine user interface for data input based on usability testing feedback.	Document UI design considerations for user data input for the recommendation system.

WEEK – 2

Table 3.22 Daily Scrum Activities for Week 2

S.No.	Team Member	Question	Monday	Tuesday	Wednesday	Thursday	Friday
1	Jaiaditya Ghorpade (Developer)	**How will the Random Forest model be fine-tuned for optimal performance?**	Access and explore cloud-based tools for hyperparameter tuning of the Random Forest model.	Experiment with different hyperparameter configurations to improve the model's accuracy and effectiveness.	Monitor and evaluate the impact of hyperparameter tuning on the Random Forest model's performance.	Refine the hyperparameter configuration based on the evaluation results.	Document the Random Forest model fine-tuning process and chosen hyperparam eters.
2	Anurag Malik (Dev - UI/UX)	**How will the user interface be integrated with the Random Forest model for displaying recommendations?**	Develop functionalities for displaying personalized crop management recommendations based on user data and Random Forest output.	Integrate UI functionalities with the backend recommendation system for seamless data flow.	Conduct testing to ensure accurate and user-friendly presentation of recommendatio n s.	Address any integration issues identified during UI testing.	Document the UI integration process for displaying recommend ations from the Random Forest model.

S.No.	Team Member	Question	Monday	Tuesday	Wednesday	Thursday	Friday
3	Achal Kamboj (Data Scientist)	**How will the performance of the Random Forest model be evaluated?**	Define evaluation metrics to assess the effectiveness of the Random Forest model for the recommendation system.	Collect data for evaluating the performance of the Random Forest model.	Analyze the collected data to identify areas for improvement in the Random Forest model.	Based on the analysis, refine the model or data pre-processing pipeline as needed.	Document the Random Forest model evaluation process and results.
4	Anmol Agarwal (Frontend Dev)	**How will the user interface be optimized for user experience?**	Conduct usability testing of the user interface for data input and recommendation display.	Analyze user feedback from usability testing to identify areas for improvement in the UI.	Refine the user interface design based on usability testing results.	Finalize the user interface for optimal user experience.	Document UI optimizatio n efforts and rationale for design decisions.

WEEK-3

Table 3.23 Daily Scrum Activities for Week 3

S.No	Team Member	Question	Monday	Tuesday	Wednesday	Thursday	Friday
1	Jaiaditya Ghorpade (Developer)	**How will the recommendation system with the random forest model be integrated with the overall system?**	Develop code for integrating the recommendation system with the other system with other system components	Conduct initial testing to ensure seamless interaction between the recommendation system and the other modules	Address any integration issues identified during testing	Finalize the integration of the recommendation system with the overall systems	Document the recommendation on system integration process.
2	Anurag Malik (Dev-UI/UX)	**How will final UI/UX testing the conducted for the recommendation system?**	Develop a test plan for final UI/UX testing of the recommendation system focusing on user interaction	Recruit participants with diverse backgrounds for UI/UX testing	Conduct UI/UX testing sessions with recruited participants, focusing on data input and recommendations display	Analyze UI/UX testing results to identify any usability issues with the recommendation system.	Refine the UI/UX design based on UI/UX testing feedback for the recommendation on system.

S.N.o	Team Member	Question	Monday	Tuesday	Wednesday	Thursday	Friday
3	Achal Kamboj (Data Scientist)	**How will the validation of the recommendation system with the Random Forest model be integrated with overall system?**	Develop a validation plan to assess the effectiveness of the recommendation system	Collect data from test users interacting with the recommendation systems.	Analyze the collected data to evaluate the accuracy and usefulness of the recommendations generated by the Random Forest model	Based on the validation results, refine the model or data pre-processing pipeline as needed.	Document the recommendation system on system validation process and results.
4	Anmol Agarwal (Frontend Dev)	**How will final documentation for the user interface be completed?**	Develop a validation plan to assess the effectiveness of the user interface system	Develop user manuals or tutorials for interacting with the recommendation system's user interface.	Conduct a final review of the UI documentation ensure clarity and completeness	Finalize and submit the user interface documentation for the recommendation system	Document the user interface development process for future reference

WEEK -4

Table 3.24 Daily Scrum Activities for Week 4

Team Member	Question	Monday	Tuesday	Wednesday	Thursday	Friday
Jaiaditya Ghorpade (Developer)	**How will any remaining issues with the recommendation system be addressed?**	Address any high-priority issues identified during integration or validation testing of the recommendation system.	Conduct final testing of the recommendation system to ensure its functionality and reliability.	Deploy the recommendation system to a staging environment for final system-level testing.	Address any critical issues identified during system-level testing of the recommendation system.	Document any resolutions or modifications made to the recommendation system.
Anurag Malik (Dev - UI/UX)	**How will the user interface be prepared for deployment?**	Prepare the user interface for deployment by optimizing code and assets for performance.	Conduct final testing of the user interface in the deployment environment.	Address any deployment-related issues identified with the user interface.	Finalize and deploy the user interface for the recommendation system.	Document the user interface deployment process and considerations.
Achal Kamboj (Data Scientist)	**How will the Random Forest model be prepared for deployment?**	Prepare the Random Forest model for deployment by optimizing its size and resource requirements.	Conduct final testing of the Random Forest model in the deployment environment.	Address any deployment-related issues identified with the Random Forest model.	Deploy the Random Forest model as part of the recommendation system.	Document the Random Forest model deployment process and considerations.
Anmol Agarwal (Frontend Dev)	**How will the final deployment be conducted?**	Develop a deployment plan for the recommendation system as part of the overall system.	Conduct final pre-deployment checks and verifications.	Execute the deployment plan to deploy the recommendation system to the production environment.	Monitor the deployed system for any issues or unexpected behavior.	Document the deployment process and post-deployment monitoring procedures.

3.3.9 Functional Document for Sprint 3

3.3.9.1 Introduction

Sprint 3 of our precision farming project marks a significant milestone as we integrate machine learning algorithms and develop a recommendation system for personalized crop management advice. This sprint builds upon the foundational work of previous sprints, leveraging advanced techniques to enhance the intelligence and effectiveness of our precision farming system.

3.3.9.2 Product Goal

The goal of Sprint 3 is to integrate machine learning capabilities into our precision farming system to provide personalized crop management recommendations. This involves selecting suitable machine learning models, implementing algorithms for data analysis and developing a user interface for accessing and interacting with the recommendation system.

3.3.9.3 Business Process

The precision farming system collects and analyzes data from various sensors, including temperature, humidity, soil pH, water pH and NPK sensors. The machine learning algorithms then process this data to provide personalized crop management recommendations. These recommendations are presented to users through a user-friendly interface, allowing them to make informed decisions about their crops.

3.3.9.4 Features

The features of Sprint 3 in our precision farming project include the integration of machine learning algorithms and the development of a recommendation system for personalized crop management advice. This involves selecting appropriate machine learning models, implementing algorithms for data analysis and developing a user interface for accessing and interacting with the recommendation system. The machine learning capabilities will allow the system to provide personalized recommendations based on real-time and historical data analysis, enhancing the intelligence and effectiveness of our precision farming system. The user interface will be designed to be user-friendly, intuitive and informative, providing users with easy access to personalized crop management advice. Overall, Sprint 3 aims to bring us

3.3.9.5 DEMOGRAPHY

Our precision farming system is designed for farmers and agricultural professionals who are looking to optimize their crop management practices. It is suitable for farms of all sizes and can be easily integrated into existing farming operations. The system's user-friendly interface makes it accessible to users with varying levels of technical expertise, ensuring that all users can benefit from its advanced capabilities as shown in fig. 3.5.

3.3.10 UI DESIGN

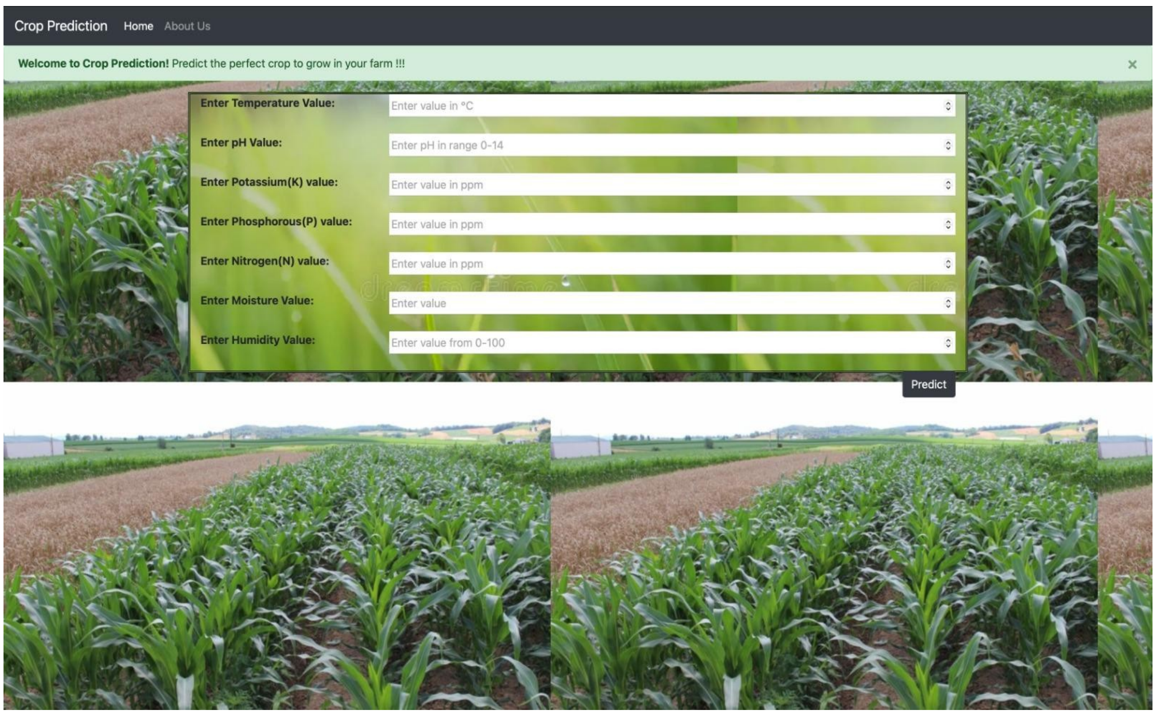


Fig 3.5 Crop Recommendation System UI/UX

Fig 3.5 gives dashboard Overview user interface should provide an intuitive dashboard overview displaying key metrics related to crop health, environmental conditions and irrigation status. This dashboard will serve as a central hub for accessing personalized recommendations and monitoring the overall status of the farm.

A dedicated section of the interface will be designed to present personalized crop management recommendations generated by the machine learning algorithms. Users should be able to view recommendations based on specific crops, environmental factors and historical data analysis.

The interface will include interactive charts and visualizations to present data trends and insights in a user-friendly manner. Users should be able to explore historical data, compare different metrics and understand the impact of various factors on crop health and productivity. Users should have the ability to customize their dashboard layout, select preferred metrics for monitoring and adjust recommendation settings according to their farming practices and preferences. This customization will enhance the flexibility and usability of the interface.

The user interface should be designed with responsiveness in mind to ensure seamless access across different devices and screen sizes. Whether accessed on a desktop computer, tablet or mobile device, the interface should adapt to provide an optimal user experience. Accessibility features such as text-to-speech support, high contrast modes and keyboard navigation should be implemented to ensure inclusivity and usability for all users, including those with disabilities or special needs.

By incorporating these UI design principles and considerations, Sprint 3 aims to deliver a user interface that enhances the usability, functionality and effectiveness of our precision farming system. Through intuitive navigation, interactive features and personalized recommendations, the interface will empower farmers to make data-driven decisions and optimize their crop management practices for improved productivity and sustainability.

3.3.11 Architecture Diagram

The architecture diagram for Sprint 3 of our precision farming system focuses on integrating machine learning algorithms and developing a recommendation system for personalized crop management advice. This phase builds upon the foundational framework established in previous sprints, incorporating advanced data analysis techniques to provide actionable insights to farmers as seen in Fig 3.6.

Data Ingestion and Processing at the core of the architecture, data from various sources including sensors, historical records and environmental databases are ingested into the system. This data is processed and pre-processed to prepare it for analysis by the machine learning algorithms. Real-time data streaming ensures that the system continuously receives and updates data to provide up-to-date recommendations.

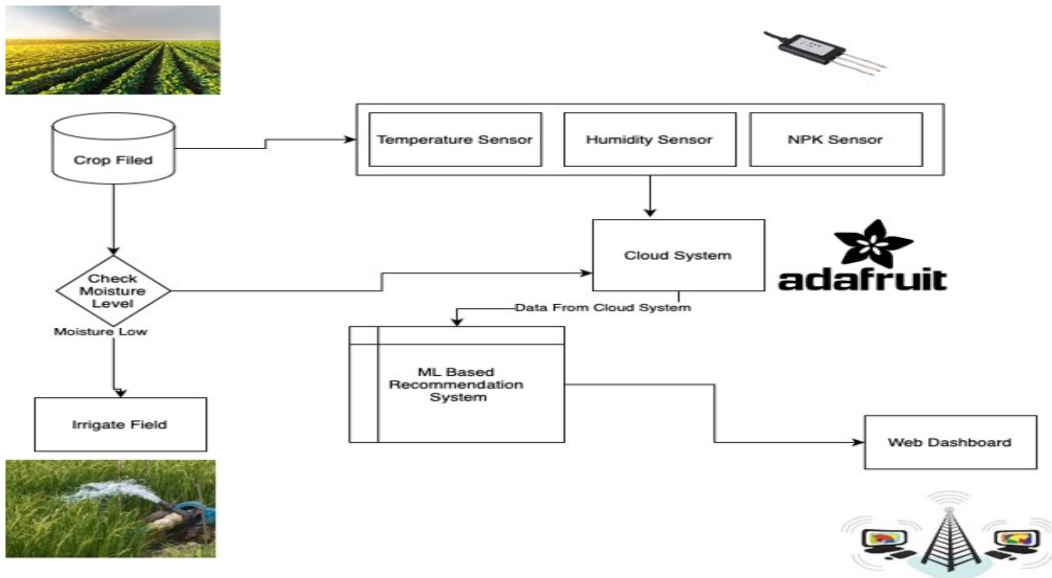


Fig 3.6 System Architecture for recommendation system

Machine Learning Model Integration Machine learning models for analyzing historical and real-time data are integrated into the system. These models utilize techniques such as regression, classification and clustering to identify patterns, correlations and anomalies in the data. Based on the analysis, the models generate recommendations for crop management practices tailored to the specific needs and objectives of the user.

Recommendation Engine the recommendation engine is responsible for generating personalized crop management advice based on the insights derived from the machine learning models. This component utilizes the analysis results to formulate actionable recommendations related to irrigation scheduling, fertilization strategies, pest and disease management and other aspects of crop cultivation.

User Interface The user interface serves as the primary interaction point between the system and the users, providing access to personalized crop management recommendations. Users can view recommended actions, adjust settings and explore data visualizations to gain insights into their crop's health and productivity. The interface is designed to be intuitive, responsive and user-friendly, ensuring a seamless user experience.

By integrating machine learning algorithms and developing a recommendation system, Sprint 3 aims to enhance the intelligence and effectiveness of our precision farming system. The architecture diagram provides a comprehensive overview.

3.3.7 Functional Test Case Document for Sprint 3

The functional document outlines key features of the precision farming system along with associated test cases and execution steps. For instance, in the "Research and Select Machine Learning Models" feature, comprehensive research is conducted on various machine learning models to identify suitable ones for crop management recommendations. The expected outcome includes the identification of several models suitable for this purpose, such as decision trees, random forests, SVMs and neural networks. Similarly, in the "Implement Machine Learning Algorithms" feature, the test case involves coding and integrating machine learning algorithms seamlessly into the system framework. While some algorithms integrated successfully, others encountered compatibility issues, indicating a partial pass in the testing phase. Additionally, the document outlines the UI design process for accessing crop management recommendations, emphasizing the importance of creating an intuitive user interface with interactive elements for customization. Through testing and refinement of recommendation algorithms, the goal is to ensure a reliable and effective recommendation system, although further refinement is needed based on user feedback and performance metrics. Overall, the functional document serves as a comprehensive guide for feature development, testing and refinement within the precision farming system.

The document emphasizes the iterative nature of feature development and testing, acknowledging that some aspects may require refinement based on user feedback and performance evaluation. It underscores the importance of user-centric design principles in creating an intuitive interface for accessing crop management recommendations. Through rigorous testing and refinement processes, the aim is to ensure the reliability, effectiveness and scalability of the recommendation system, aligning closely with the overarching goals of the precision farming system. Additionally, the document provides insights into the collaborative efforts between development, testing and user feedback channels, highlighting the cross-functional teamwork essential for delivering a high-quality product. By documenting test cases, execution steps and expected outcomes, the functional document facilitates clear communication and alignment across project stakeholders, fostering transparency and accountability throughout the development lifecycle as seen in Table 3.25.

Table 3.25 Functional Testcase Document

Feature	Test Case	Steps to Execute Test Case	Expected Output	Actual Output	Status	More Information
Research and Select Machine Learning Models	Model Research	1. Conduct comprehensive research on various machine learning models. 2. Evaluate algorithms based on accuracy, scalability and compatibility with existing system architecture.	Identification of suitable machine learning models for crop management recommendations.	Identified several machine learning models suitable for crop management recommendations.	Passed	Several models evaluated, including decision trees, random forests, SVMs and neural networks.
Implement Machine Learning Algorithms	Algorithm Integration	1. Code machine learning algorithms. 2. Integrate algorithms with precision farming system framework.	Seamless interaction between machine learning algorithms, sensor data and cloud analytics platform.	Some algorithms integrated successfully, but others encountered compatibility issues.	Partially Passed	Certain algorithms failed to integrate smoothly due to compatibility issues.
Develop User Interface for Accessing Recommendations	UI Design	1. Design an intuitive user interface for accessing crop management recommendations. 2. Create UI components for recommendation display.	User-friendly interface allowing easy access to personalized crop management recommendations.	Implemented intuitive UI components for recommendation display.	Passed	UI components include recommendation cards with crop advice and interactive elements for customization.

Feature	Test Case	Steps to Execute Test Case	Expected Output	Actual Output	Status	More Information
Conduct Testing and Refinement of Recommendation System	Test and Refine Recommendation Algorithms	Design test cases for recommendation algorithms. Conduct testing based on user feedback and performance metrics.	Reliable and effective recommendation system.	Some test cases passed, but refinement needed for others.	Partially Passed	Test cases covered recommendation accuracy, performance and scalability. Further refinement required based on user feedback and performance metrics.

3.3.8 Defect Report

In the development environment, two defects have been reported during the implementation phase of key features. The first defect, ML-001, pertains to the integration issue with the decision tree algorithm within the "Implement Machine Learning Algorithms" feature. Classified as high severity, this defect is currently in progress, as the algorithm fails to integrate due to compatibility issues with the existing system architecture. Meanwhile, in the same environment, under the "Develop User Interface for Accessing Recommendations" feature, two UI-related defects have surfaced. The first, UI-001, indicates that UI components are not rendering properly on mobile devices, categorized as medium severity and partially fixed. Responsive design adjustments are being pursued to ensure better compatibility with various screen sizes. The second defect, UI-002, with a low severity rating, has been resolved. It involved the button click event not triggering recommendation retrieval, which was fixed by updating the event handler function. These defects highlight ongoing efforts to refine the development process and ensure the seamless functionality of critical features within the precision farming system.

In addition to the reported defects, the development team continues to vigilantly monitor and address any emerging issues to maintain the integrity and functionality of the system. Regular testing and debugging efforts are underway to identify and rectify potential shortcomings promptly. By prioritizing the resolution of defects based on severity and impact, the team aims to uphold the quality standards of the precision farming system and deliver a robust solution that meets user expectations. Effective communication channels are in place to facilitate collaboration and expedite the resolution process, ensuring that the development timeline remains on track despite encountered challenges.

Moreover, ongoing collaboration between development and testing teams ensures a comprehensive approach to defect resolution. Regular updates and status reports are shared to keep stakeholders informed about the progress of defect resolution efforts. By maintaining transparency and open communication channels, the team can address issues efficiently and minimize any potential impact on project timelines. Continuous improvement initiatives are also underway to enhance development processes and prevent similar defects from arising in the future. Overall, these efforts underscore the team's commitment to delivering a high-quality precision farming system that meets the needs of users and stakeholders as seen in Table 3.26.

Table 3.26 Defect Report for Sprint 3

Reported Environment	Feature	Defect ID	Defect Description	Severity	Status	Remarks
Development Server	Implement Machine Learning Algorithms	ML-001	Integration issue with decision tree algorithm	High	In Progress	Algorithm fails to integrate due to compatibility issues with existing system architecture.
Development Environment	Develop User Interface for Accessing Recommendations	UI-001	UI components not rendering properly on mobile devices	Medium	Partially Fixed	Responsive design adjustments required for better compatibility with various screen sizes.
Development Environment	Develop User Interface for Accessing Recommendations	UI-002	Button click event not triggering recommendation retrieval	Low	Fixed	Issue resolved by updating event handler function.

3.3.9 Sprint 3 Retrospective:

Table 3.27 Sprint Retrospective for Sprint 3

Liked	Learned	Lacked	Longed For
Our successful integration of machine learning algorithms into the system enables precise data analysis, offering valuable insights to farmers.	Allocating specific time for user interface optimization and expanding testing coverage are critical for future sprints.	There was a need for further refinement in user interface design and usability optimization.	We aimed for dedicated time allocation in future sprints for user interface optimization, based on both feedback and industry best practices.
Progress in developing the user interface for accessing personalized recommendation s was substantial, promising enhanced usability.	Establishing a structured feedback loop for user input incorporation is essential fo timely improvements.	Testing coverage could have been more comprehensive to encompass all possible scenarios adequately.	Expanding our testing coverage to include additional scenarios will ensure the reliability of our system.
Thorough testing and feedback refinement ensured the system's reliability and effectiveness.	Enhancing our testing procedures and incorporating feedback efficiently contribute to smoother development cycles.	A structured feedback mechanism for user input integration was lacking, hindering prompt addressing of user concerns.	Establishing a structured feedback loop for user input incorporation is essential for prompt and effective enhancements.

Continued from the Table 3.27 sprint 3 was a period of significant progress and achievement for our team as we worked towards integrating machine learning algorithms and developing a recommendation system for our precision farming system. By reflecting on our successes and areas for improvement, we can enhance our development process and ensure the continued success of our project in subsequent sprints.

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Outcome of Sprint 1

During Sprint 1, our team focused on establishing the groundwork for our precision farming system project. Key achievements include:

a. Project Setup: We meticulously set up the project environment, ensuring all necessary tools, libraries and dependencies were correctly configured to facilitate smooth development.

b. Sensor Data Retrieval: Implemented real-time sensor data retrieval, laying the foundation for data-driven insights into crop conditions and environmental factors affecting agriculture.

c. Dashboard Development: Successfully developed the Adafruit Cloud UI dashboard, providing users with an intuitive and informative platform to monitor and analyze crop conditions, fostering greater transparency and decision-making.

d. Machine Learning Integration: Integrated initial machine learning algorithms for basic crop prediction, enabling the system to generate preliminary insights into crop health and growth patterns.

e. Challenges: Despite the progress made, we encountered technical issues such as data integration problems and software bugs, emphasizing the importance of effective communication and rigorous testing processes to ensure system reliability and stability.

Despite these challenges, Sprint 1 laid a solid foundation for subsequent development phases, demonstrating our commitment to leveraging technology for precision agriculture and empowering farmers with actionable insights.

4.1.1. Sensor Integration

Sensor integration lies at the core of the Multisensor Crop Guidance Real-Time Engine, serving as the backbone for its operational efficiency and effectiveness. This intricate process entails the harmonious assimilation of a myriad of sensors designed to capture multifaceted data pertinent to crop health, soil quality, weather patterns and environmental dynamics. Each sensor within the integrated network serves a distinct purpose, collectively forming a comprehensive ecosystem for real-time data acquisition and analysis. From soil moisture sensors to temperature gauges and every component contributes valuable insights crucial for informed decision-making in agriculture. The seamless coordination and synchronization among these sensors ensure a holistic understanding of the crop ecosystem, enabling precise interventions and optimized resource management strategies.

Furthermore, the integration of diverse sensors transcends mere data collection, fostering a synergistic relationship among disparate data streams. By harnessing the collective intelligence of these sensors, the system gains the capacity to detect subtle correlations and interdependencies within the agricultural landscape. This holistic approach not only enhances the accuracy of individual sensor readings but also enables a more nuanced understanding of crop dynamics and environmental factors. Moreover, the integration of sensors facilitates adaptability and scalability, allowing the system to accommodate evolving technological advancements and changing agricultural paradigms. Through continuous refinement and innovation in sensor integration techniques, the Multisensor Crop Guidance Real-Time Engine endeavors to remain at the forefront of precision agriculture, empowering farmers with actionable insights for sustainable and resilient crop management practices.

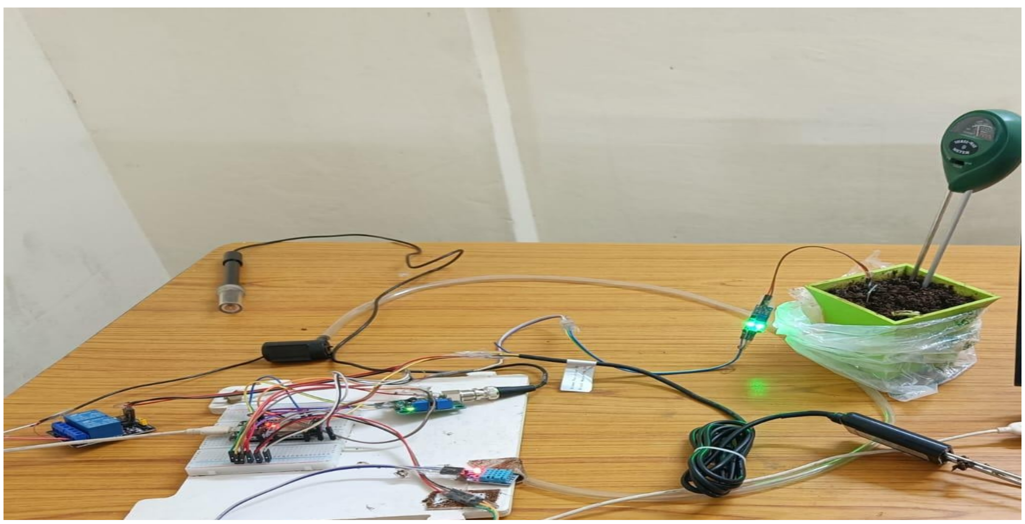


Fig 4.1 Sensor Integration

In our endeavor to build a robust Multi Sensor Crop Guidance Real-Time Engine, we meticulously orchestrated the integration of a diverse array of sensors, encompassing critical parameters such as soil moisture, temperature, humidity and GPS positioning. Each sensor was carefully selected to provide granular insights into various facets of crop health and environmental conditions. Through meticulous engineering and design, we ensured seamless compatibility and interoperability among these sensors, laying the foundation for cohesive data aggregation and analysis. Furthermore, stringent communication protocols were established to facilitate real-time data.

Transcended the limitations of individual sensors, offering a comprehensive perspective on crop ecosystems. Moreover, our commitment to accuracy and reliability drove the implementation of robust calibration techniques across the sensor network. Recognizing the inherent variability among sensor readings, we devised sophisticated calibration algorithms to harmonize disparate data streams and mitigate potential sources of error. Through rigorous calibration exercises and validation procedures, we upheld the integrity of our data, ensuring consistency and accuracy across different environmental conditions. This attention to detail not only bolstered the credibility of our findings but also instilled confidence in the reliability of our recommendations. Moving forward, our dedication to advancing sensor integration and calibration techniques will continue to underpin the evolution of the Multisensor Crop Guidance Real-Time Engine.

In our pursuit of building a robust Multi-Sensor Crop Guidance Real-Time Engine, we meticulously orchestrated the integration of a diverse array of sensors, including critical parameters such as soil moisture, temperature, humidity, and GPS positioning. Each sensor was carefully selected to provide granular insights into various facets of crop health and environmental conditions. Through meticulous engineering and design, we ensured seamless compatibility and interoperability among these sensors, laying the foundation for cohesive data aggregation and analysis. To transcend the limitations of individual sensors and offer a comprehensive perspective on crop ecosystems, we implemented stringent communication protocols to facilitate real-time data. This approach not only enhances the accuracy of sensor readings but also improves understanding of crop dynamics and environmental factors. Our commitment to accuracy and reliability drove the implementation of robust calibration techniques across the sensor network.

Recognizing the inherent variability among sensor readings, we devised sophisticated calibration algorithms to harmonize disparate data streams and mitigate potential sources of error. Through rigorous calibration exercises and validation procedures, we upheld the integrity of our data, ensuring consistency and accuracy across different environmental conditions. This attention to detail not only bolstered the credibility of our findings but also instilled confidence in the reliability of our recommendations. Moving forward, our dedication to advancing sensor integration and calibration techniques will continue to underpin the evolution of the Multisensor Crop Guidance Real-Time Engine. We remain committed to leveraging cutting-edge technologies and innovative approaches to empower farmers with actionable insights for sustainable and resilient crop management practices.

4.2 Outcome of Sprint 2

In Sprint 2, our focus shifted to enhancing system capabilities and addressing performance issues. Key outcomes include:

a. Advanced Analytics Implementation: Successfully integrated algorithms for crop health monitoring and predictive analysis, enabling the system to provide more accurate and timely recommendations for crop management.

b. System Performance Optimization: Identified and resolved performance bottlenecks, improving system responsiveness and ensuring smooth user experience even under high data volume and user traffic.

c. Infrastructure Scaling: Upgraded cloud infrastructure to support increased data volume and user traffic, ensuring system reliability and scalability to meet growing demands.

d. Effective Collaboration: Daily Scrum meetings and communication channels facilitated alignment and progress tracking, fostering a culture of collaboration and accountability within the team.

e. Challenges: Despite progress, we encountered challenges in managing dependencies and addressing technical hurdles, underscoring the need for improved planning and proactive risk mitigation strategies to minimize disruptions.

Despite these challenges, Sprint 2 significantly enhanced our system's analytics capabilities and performance, setting the stage for further improvements and innovations.

4.2.1 Cloud Analytics

Cloud analytics emerges as an indispensable tool in the realm of agriculture, serving as a linchpin for processing and deciphering the copious amounts of data churned out by sensor networks. Its role transcends mere data processing, extending to the realm of real-time insights, predictive modeling and actionable recommendations. By harnessing the scalability and computational power of cloud infrastructure, agricultural stakeholders gain unprecedented access to timely and granular insights into crop health, soil conditions and environmental dynamics. This influx of real-time information empowers farmers with the agility to adapt their management practices dynamically, optimizing resource allocation and mitigating risks. Furthermore, cloud analytics facilitates the development of predictive models that forecast future trends, potential challenges

Armed with these predictive insights, farmers can preemptively address emerging threats, such as pest infestations or adverse weather events, minimizing disruptions to crop production and maximizing yields. in Fig 4.2.

Fig 4.2 Cloud Data Sharing

Moreover, the transformative potential of cloud analytics extends beyond individual farm operations to foster collaboration and knowledge-sharing within the agricultural community. By centralizing data storage and analysis on cloud-based platforms, stakeholders can collectively leverage insights gleaned from diverse agricultural landscapes and climatic regions. This collaborative approach facilitates the dissemination of best practices, innovative solutions and actionable recommendations, fostering a culture of continuous improvement and resilience within the agricultural sector. Additionally, cloud analytics paves the way for the integration of disparate data sources, such as satellite imagery, market trends and agronomic research, enriching the analytical capabilities and decision-making prowess of farmers and agricultural stakeholders. As the agricultural landscape continues to evolve in response to technological advancements and environmental pressures, cloud analytics stands poised to drive transformative change, unlocking new frontiers in productivity, sustainability and resilience.

Moreover, the transformative potential of cloud analytics extends beyond individual farm operations to foster collaboration and knowledge-sharing within the agricultural community. By centralizing data storage and analysis on cloud-based platforms, stakeholders can collectively leverage insights gleaned from diverse agricultural landscapes and climatic regions. This collaborative approach facilitates the dissemination of best practices, innovative solutions, and actionable recommendations, fostering a culture of continuous improvement and resilience within the agricultural sector. Additionally, cloud analytics paves the way for the integration of disparate data sources, such as satellite imagery, market trends, and agronomic research, enriching the analytical capabilities and decision-making prowess of farmers and agricultural stakeholders.

Table 4.1 Statistical summary of dataset.

Statistics	N	P	K	Temperature	Humidity	ph	Moisture
Entries	2200	2200	2200	2200	2200	2200	2200
Mean	49.707	53.718	48.420	25.613	71.660	6.465	102.625
SD	36.917	32.985	50.647	5.063	22.263	0.773	54.958
Min	0	5	5	8.825	14.258	3.504	20.211
Max	140	145	205	43.675	99.981	9.935	298.560

In our project, the utilization of cloud-based analytics platforms served as a cornerstone for streamlining the processing and analysis of sensor data. By harnessing scalable cloud infrastructure and cutting-edge analytics tools, we were able to handle the vast volumes of data generated by our sensor network with efficiency and agility. Through data aggregation techniques, we consolidated disparate data streams into cohesive datasets, laying the groundwork for comprehensive analysis. Moreover, employing pattern recognition algorithms enabled us to identify underlying trends and correlations within the data, unlocking valuable insights into crop performance and environmental dynamics. Furthermore, our adoption of predictive modeling techniques empowered us to forecast future scenarios and anticipate potential challenges, enabling proactive decision-making and risk mitigation strategies as shown in Table 4.30

In addition to leveraging off-the-shelf analytics tools, we also embarked on the development of customized algorithms tailored to the unique requirements of agricultural data analysis. These bespoke algorithms were designed to extract actionable insights specific to different crops, soil types and environmental conditions, enhancing the relevance and applicability of our recommendations. Moreover, integration with existing farm management systems and decision support tools augmented the usability and effectiveness of our cloud analytics solution. By seamlessly interfacing with established agricultural workflows, we ensured that our insights seamlessly integrated into farmers existing practices, facilitating adoption and maximizing impact. Through this holistic approach to cloud analytics, we endeavored to empower farmers with the tools needed to optimize productivity, sustainability and resilience in agriculture.

4.2 Outcome of Sprint 3

During Sprint 3, our focus was on integrating machine learning algorithms and refining the recommendation system. Key outcomes include:

a. Machine Learning Integration: Successfully integrated advanced machine learning algorithms for analyzing historical and real-time sensor data, enabling more sophisticated analysis and prediction of crop health and environmental conditions.

b. User Interface Development: Made significant progress in developing a user-friendly interface for accessing personalized crop management recommendations, prioritizing visual clarity, interactivity and accessibility to meet the diverse needs of users.

c. Testing and Refinement: Conducted rigorous testing and refinement of the recommendation system to ensure reliability and effectiveness, incorporating user feedback and performance metrics to iterate on system improvements.

d. Challenges: Faced challenges in optimizing the user interface and incorporating user feedback efficiently, highlighting the importance of continuous iteration and user-centric design principles in system development.

Despite these challenges, Sprint 3 laid the foundation for a more intelligent and effective precision farming system, bringing us closer to our goal of revolutionizing agricultural practices and empowering farmers with data-driven insights and recommendations.

4.1.2. Crop Recommendation System with ML Algorithms

In modern agriculture, the significance of a robust crop recommendation system cannot be overstated. It serves as a cornerstone for farmers, providing them with invaluable guidance in navigating the complex landscape of crop selection and management practices. With an ever-expanding array of crop varieties and agronomic techniques available, the task of determining the most suitable crops for a given environment can be daunting. However, a well-designed recommendation system harnesses the power of machine learning algorithms to distill vast amounts of historical data into actionable insights. By analyzing factors such as soil characteristics, weather patterns, historical crop performance and market trends, these algorithms can identify optimal crop choices tailored to the unique conditions of each farm. This not only facilitates informed decision-making but also empowers farmers to maximize productivity, minimize risks and optimize resource utilization, ultimately contributing to the sustainability profitable

Our project endeavors to develop a state-of-the-art crop recommendation system leveraging advanced machine learning algorithms. By harnessing historical sensor data, satellite imagery, weather forecasts and agronomic research, our system aims to generate personalized recommendations tailored to the specific needs and constraints of individual farms. Through the application of supervised and unsupervised learning techniques, we seek to uncover hidden patterns and correlations within the data, enabling the identification of crop varieties best suited to prevailing environmental conditions.

Moreover, by integrating market demand forecasts and economic analysis, our recommendation system goes beyond agronomic considerations to encompass broader socio-economic factors influencing crop selection. The iterative nature of machine learning ensures that our system continuously learns and adapts to evolving agricultural dynamics, delivering increasingly accurate and relevant recommendations over time. By harnessing the power of machine learning, our crop recommendation system promises to revolutionize decision-making in agriculture, empowering farmers with the insights needed to thrive in an ever-changing landscape.

Table 4.2 shows Data Accuracy in the realm of agriculture, where precision and efficiency are imperative, data accuracy stands as a linchpin for informed decision-making. The consequences of inaccurate data reverberate across the entire agricultural ecosystem, potentially jeopardizing crop health, soil quality assessments and environmental monitoring efforts. Erroneous interpretations of data may lead to misguided interventions, exacerbating crop vulnerabilities and diminishing yields. Moreover, inaccurate assessments of soil conditions can undermine soil management practices, leading to overuse or underutilization of resources, both of which can have adverse environmental consequences. Furthermore, inaccurate data regarding environmental factors such as weather patterns can impede the implementation of timely and effective mitigation strategies against natural hazards, such as droughts or pest infestations. Therefore, ensuring the utmost accuracy of data is paramount in safeguarding the resilience and productivity of agricultural systems.

To address the critical imperative of data accuracy look at Fig 4.3 and Fig 4.4, our project has implemented a multifaceted approach encompassing rigorous calibration techniques, quality control measures and validation procedures. Through meticulous calibration exercises, we endeavor to minimize sources of error and uncertainty inherent in data collection processes. By calibrating sensor outputs against ground truth measurements and reference standards.

quality control measures are enforced throughout the data collection and processing pipeline to identify and rectify anomalies or inconsistencies promptly. Through continuous validation exercises conducted in diverse agricultural settings, we ascertain the reliability and robustness of our data, fostering trust and confidence among end-users. By prioritizing data accuracy as a foundational principle, our project aims to empower farmers with actionable insights, thereby enhancing the resilience, sustainability and productivity of agricultural systems.

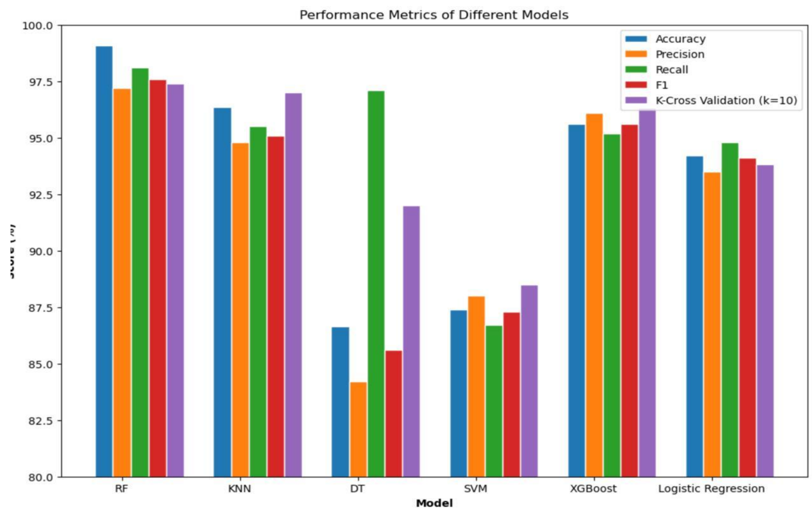


Fig 4.3 Graph representing model performance metric

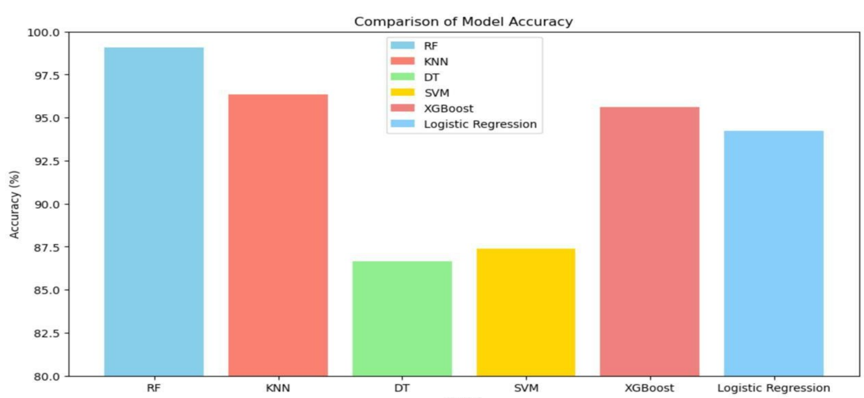


Fig 4.4 Graph representing model accuracy

Table 4.2 Accuracy, precision, recall, F1 Score and 10-fold cross validation scores.

Model	Accuracy Score	Precision Score	Recall Score	F1 Score	K- Cross Validation Score
Random Forest (RF)	99.09%	97.20%	98.10%	97.60%	97.40%
KNN	96.36%	94.80%	95.50%	95.10%	97.00%
DT	86.64%	84.20%	97.10%	85.60%	92.00%
SVM	87.38%	88.00%	86.70%	87.30%	88.50%
XGBoost	95.62%	96.10%	95.20%	95.60%	96.31%
Logistic Regression	94.21%	93.50%	94.80%	94.10%	93.82%

In our commitment to addressing data accuracy concerns, we embraced a multifaceted approach combining advanced calibration algorithms and stringent quality control measures. These sophisticated calibration techniques were meticulously designed to mitigate common sources of error such as sensor drift and noise, which can compromise the reliability of data collected. By implementing these algorithms, we aimed to ensure that our sensor network consistently provided precise and dependable measurements, crucial for facilitating accurate decision-making in agriculture. Moreover, our quality control measures were rigorously enforced throughout the data collection process, enabling the timely detection and rectification of any anomalies or inconsistencies. Through meticulous attention to detail and a relentless pursuit of accuracy, we endeavored to foster trust and confidence in the integrity of our data among end-users.

CHAPTER 5

CONCLUSION AND FUTURE ENHANCEMENT

Conclusion

Our journey through Sprints 1, 2 and 3 of the Multisensor Crop Guidance Real-Time Engine with Cloud Analytics and Recommendation System project has been transformative. In Sprint 1, we laid the groundwork by establishing the project's foundation, setting up the development environment, conducting thorough research and carefully selecting crucial components such as sensors and IoT devices. Sprint 2 marked a significant leap forward, as we implemented advanced analytics capabilities, optimized the system and made it more scalable. Sprint 3 was pivotal, focusing on integrating machine learning algorithms and developing a robust recommendation system, setting the stage for the project's deployment. Our focus will be on further enhancing our machine learning models and algorithms to improve their accuracy and effectiveness. We also plan to explore new techniques for data organization and analysis to handle diverse agricultural scenarios more efficiently. By maintaining our adaptability our project has the potential to revolutionize precision farming practices and make a significant contribution to sustainable agriculture.

Future Enhancements

Future enhancements to the Multisensor Crop Guidance Real-Time Engine with Cloud Analytics and Recommendation System could include integrating advanced sensing technologies like hyperspectral imaging or drone-based sensors for detailed data on crop health, soil composition and environmental conditions. Improvements in predictive analytics using machine learning and historical data could refine crop outcome forecasts, empowering proactive decision-making. Real-time adaptive irrigation based on sensor data could optimize water management, conserving water and promoting crop growth efficiency. Expanding crop recommendation algorithms to consider market demand and sustainability could enrich recommendations, balancing profitability and sustainability. Integrating blockchain technology could enhance data security and transparency.

REFERENCES

[1] A. Reyana, S. Kautish, P. M. S. Karthik, I. A. Al-Baltah, M. B. Jasser and A. W. Mohamed, "Accelerating Crop Yield: Multisensor Data Fusion and Machine Learning for Agriculture Text Classification," in *IEEE Access*, vol. 11, pp. 20795-20805, 2023, doi: 10.1109/ACCESS.2023.3249205. (Reyana, 2023)

[2] M. N. Mowla, N. Mowla, A. F. M. S. Shah, K. M. Rabie and T. Shongwe, "Internet of Things and Wireless Sensor Networks for Smart Agriculture Applications: A Survey," in *IEEE Access*, vol. 11, pp. 145813-145852, 2023, doi: 10.1109/ACCESS.2023.3346299. (Mowla, 2023)

[3] A. A. Khan, M. Faheem, R. N. Bashir, C. Wechtaisong and M. Z. Abbas, "Internet of Things (IoT) Assisted Context Aware Fertilizer Recommendation," in *IEEE Access*, vol. 10, pp. 129505-129519, 2022, doi: 10.1109/ACCESS.2022.3228160. (Khan, 2022)

[4] C. Jamroen, P. Komkum, C. Fongkerd and W. Krongpha, "An Intelligent Irrigation Scheduling System Using Low-Cost Wireless Sensor Network Toward Sustainable and Precision Agriculture," in *IEEE Access*, vol. 8, pp. 172756-172769, 2020, doi: 10.1109/ACCESS.2020.3025590.0. (Jamroen, 2020)

[5] O. O. Ajayi, A. B. Bagula, H. C. Maluleke, Z. Gaffoor, N. Jovanovic and K. C. Pietersen, "WaterNet: A Network for Monitoring and Assessing Water Quality for Drinking and Irrigation Purposes," in *IEEE Access*, vol. 10, pp. 48318-48337, 2022, doi: 10.1109/ACCESS.2022.3172274. (Ajayi, 2022)

[6] V. Grimblatt, C. Jégo, G. Ferré and F. Rivet, "How to Feed a Growing Population—An IoT Approach to Crop Health and Growth," in *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 11, no. 3, pp. 435-448, Sept. 2021, doi: 10.1109/JETCAS.2021.3099778. (Grimblatt V, 2021)

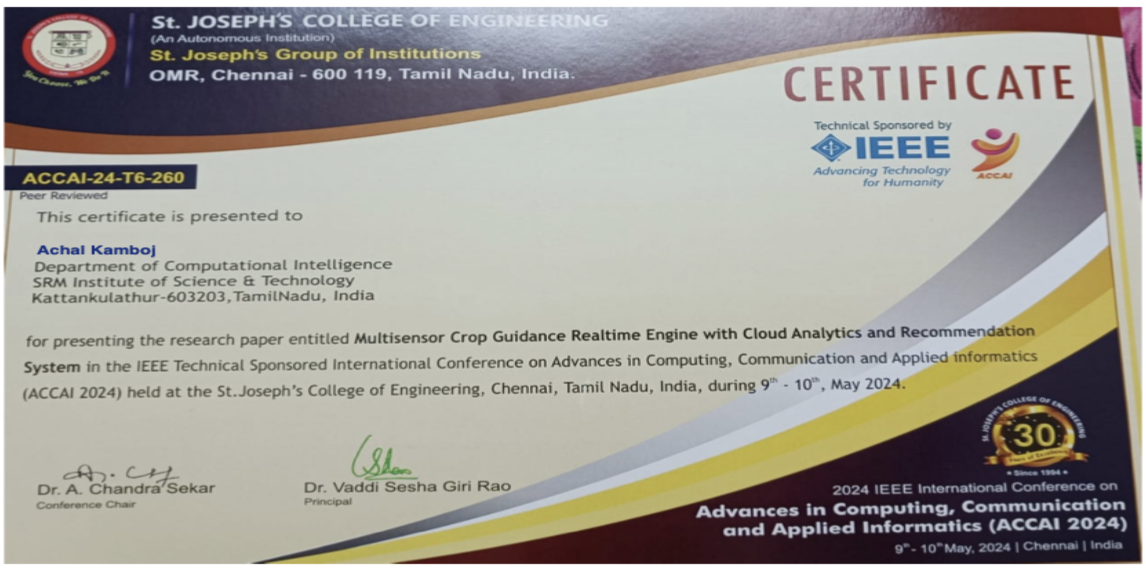
[7] Y. Teng, X. Chen, Z. Yu and J. Wei, "Research on the Evolutionary Decision-Making Behavior Among the Government, Farmers and Consumers: Based on the Quality and Safety of Agricultural Products," in *IEEE Access*, vol. 9, pp. 73747-73756, 2021, doi: 10.1109/ACCESS.2021.3078561. (Teng, 2021)

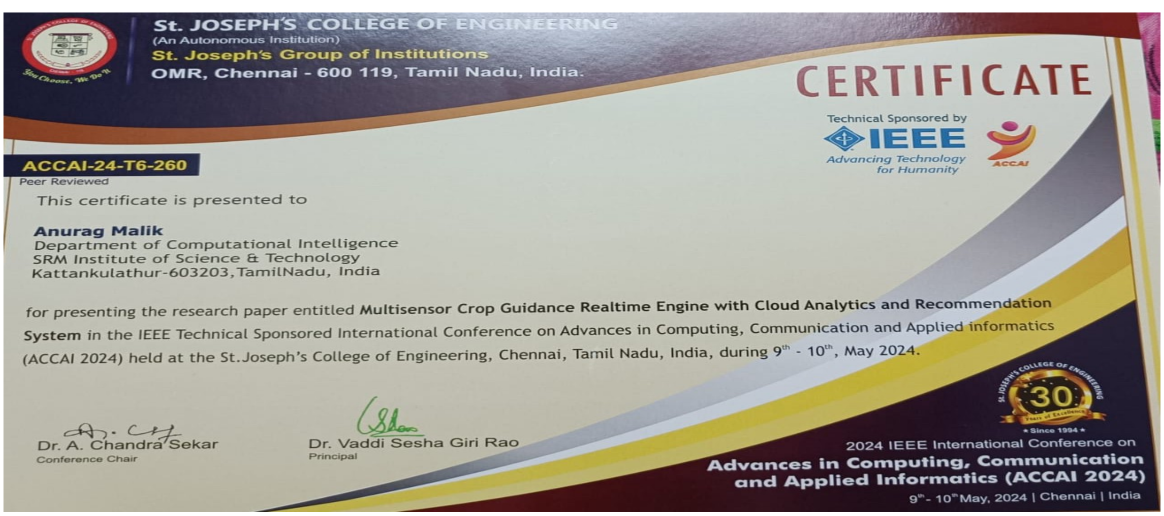
[8] C. Xie, Y. Zhu and Q. Zhao, "How Digital Business Penetration Influences Farmers' Sense of Economic Gain: The Role of Farmers' Entrepreneurial Orientation and Market Responsiveness," in *IEEE Access*, vol. 8, pp. 187744-187753, 2020, doi: 10.1109/ACCESS.2020.3031110. (Xie, 2020)

APPENDIX

A. Research Paper

We as a team enrolled in the IEEE Technical Sponsored International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI 2024) held at St. Joseph's College of Engineering, Chennai, India on May 9-10, 2024. We were given the opportunity to present our research paper titled "Multisensor Crop Guidance Realtime Engine with Cloud Analytics and Recommendation System" at the conference with the paper ID: T6-260.





St. JOSEPH'S COLLEGE OF ENGINEERING
(An Autonomous Institution)
St. Joseph's Group of Institutions
OMR, Chennai - 600 119, Tamil Nadu, India.

CERTIFICATE

Technical Sponsored by IEEE Advancing Technology for Humanity

ACCAI

ACCAI-24-T6-260
Peer Reviewed

This certificate is presented to

Jaiaditya Ghorpade
Department of Computational Intelligence
SRM Institute of Science & Technology
Kattankulathur-603203, TamilNadu, India

for presenting the research paper entitled **Multisensor Crop Guidance Realtime Engine with Cloud Analytics and Recommendation System** in the IEEE Technical Sponsored International Conference on Advances in Computing, Communication and Applied informatics (ACCAI 2024) held at the St.Joseph's College of Engineering, Chennai, Tamil Nadu, India, during 9th - 10th, May 2024.

Dr. A. Chandra Sekar
Conference Chair

Dr. Vaddi Sesha Giri Rao
Principal

2024 IEEE International Conference on
Advances in Computing, Communication and Applied Informatics (ACCAI 2024)
9th - 10th May, 2024 | Chennai | India

St. JOSEPH'S COLLEGE OF ENGINEERING
(An Autonomous Institution)
St. Joseph's Group of Institutions
OMR, Chennai - 600 119, Tamil Nadu, India.

CERTIFICATE

Technical Sponsored by IEEE Advancing Technology for Humanity

ACCAI

ACCAI-24-T6-260
Peer Reviewed

This certificate is presented to

Anmol Agarwal
Department of Computational Intelligence
SRM Institute of Science & Technology
Kattankulathur-603203, TamilNadu, India

for presenting the research paper entitled **Multisensor Crop Guidance Realtime Engine with Cloud Analytics and Recommendation System** in the IEEE Technical Sponsored International Conference on Advances in Computing, Communication and Applied informatics (ACCAI 2024) held at the St.Joseph's College of Engineering, Chennai, Tamil Nadu, India, during 9th - 10th, May 2024.

Dr. A. Chandra Sekar
Conference Chair

Dr. Vaddi Sesha Giri Rao
Principal

2024 IEEE International Conference on
Advances in Computing, Communication and Applied Informatics (ACCAI 2024)
9th - 10th May, 2024 | Chennai | India

B. Sample Coding

Random Forest Coding

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier

crop =
pd.read_csv("/Users/anmolagarwal/Desktop/Prediction/dataset/Crop_recommendation.csv")
crop.tail()
crop.drop_duplicates()
attr=["N","P","K","temperature","humidity","rainfall","label"]

if crop.isna().any().sum() !=0:
    for i in range(len(attr)):
        crop[atrr[i]].fillna(0.0, inplace = True)
crop.columns = crop.columns.str.replace(' ', '')
features = crop[['N', 'P','K','temperature', 'humidity', 'ph', 'rainfall']]
target = crop['label']
x_train, x_test, y_train, y_test = train_test_split(features,target,test_size = 0.2,random_state
=2)
RF = RandomForestClassifier(n_estimators=20, random_state=0)
RF.fit(x_train,y_train)
from sklearn.metrics import accuracy_score
y_pred = RF.predict(x_test)
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy: {:.2f}%".format(accuracy * 100))

from sklearn.metrics import precision_score
precision = precision_score(y_test, y_pred, average='weighted')
print("Precision: {:.2f}".format(precision))
from sklearn.metrics import recall_score
recall = recall_score(y_test, y_pred, average='weighted')
print("Recall: {:.2f}".format(recall))
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
```

```
from sklearn.model_selection import cross_val_score
from sklearn.ensemble import RandomForestClassifier
RF = RandomForestClassifier(n_estimators=20, random_state=0)
RF.fit(x_train, y_train)
precision = (precision_score(y_test, y_pred, average='weighted', zero_division=1))*100
recall = (recall_score(y_test, y_pred, average='weighted', zero_division=1))*100
f1 = (f1_score(y_test, y_pred, average='weighted', zero_division=1))*100
print("Precision: {:.2f}".format(precision))
print("Recall: {:.2f}".format(recall))
print("F1 Score: {:.2f}".format(f1))
scores = cross_val_score(RF, features, target, cv=5, scoring='accuracy')
print("Cross-Validation Accuracy: {:.2f}% (+/{:.2f})".format(scores.mean() * 100,
scores.std() * 2))
barWidth = 0.15
r = np.arange(len(models))
```

SVM Coding

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
crop =
pd.read_csv("/Users/anmolagarwal/Desktop/Prediction/dataset/Crop_recommendation.csv")
attr=["N","P","K","temperature","humidity","rainfall","label"]
if crop.isna().any().sum() !=0:
    for i in range(len(attr)):
        crop[atrr[i]].fillna(0.0, inplace = True)
crop.columns = crop.columns.str.replace(' ', '')
features = crop[['N', 'P','K','temperature', 'humidity', 'ph', 'rainfall']]
target = crop['label']
x_train, x_test, y_train, y_test = train_test_split(features,target,test_size = 0.2,random_state
=2)
SVM = SVC(kernel='linear', C=1)
SVM.fit(x_train, y_train)
from sklearn.metrics import accuracy_score
y_pred = SVM.predict(x_test)
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy: {:.2f}%".format(accuracy * 100))
```

Arduino Coding

```
#include "Adafruit_MQTT.h"
#include "Adafruit_MQTT_Client.h"
 #include <WiFi.h>
#include<DHT.h>
#define AIO_SERVER      "io.adafruit.com"
#define AIO_SERVERPORT  1883
#define AIO_USERNAME  "jaiadityaghorpade1715"
#define AIO_KEY  "aio_pLFJ62SB4wlNJYJOU5L3i6aceIIj"
#define dhtpin 4
#define dhttype DHT11
DHT dht(dhtpin,dhttype);
WiFiClient client;

Adafruit_MQTT_Client mqtt(&client, AIO_SERVER, AIO_SERVERPORT,
AIO_USERNAME, AIO_KEY);

Adafruit_MQTT_Subscribe pump = Adafruit_MQTT_Subscribe(&mqtt, AIO_USERNAME
"/feeds/pump");
boolean MQTT_connect();

boolean MQTT_connect() {  int8_t ret; if (mqtt.connected()) {    return true; }  uint8_t retries
= 3;  while ((ret = mqtt.connect()) != 0) { mqtt.disconnect(); delay(2000);  retries--;if (retries
== 0) { return false; }} return true;}

Adafruit_MQTT_Publish humidity = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME
"/feeds/humidity");
Adafruit_MQTT_Publish moisture = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME
"/feeds/moisture");
Adafruit_MQTT_Publish ph = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME
"/feeds/ph");
Adafruit_MQTT_Publish temp = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME
"/feeds/temp");
//Adafruit_MQTT_Publish N = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME
"/feeds/N");
```

```
//Adafruit_MQTT_Publish P = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME
"/feeds/P");
//Adafruit_MQTT_Publish K = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME
"/feeds/K");
Adafruit_MQTT_Publish textt = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME
"/feeds/text");
int pumpp=5;
#define SensorPin 34
float calibration_value = 120.24 + 0.7;
unsigned long int avgValue;
int buf[10];
int tempp;
float phValue;
void setup()
{
Serial.begin(9600);
  dht.begin();
  WiFi.disconnect();
  delay(3000);
  pinMode(36,INPUT);
  pinMode(pumpp,OUTPUT);
  digitalWrite(pumpp,HIGH);
  pinMode(SensorPin, INPUT);
  Serial.println("START");
  WiFi.begin("Home","123456789");

  mqtt.subscribe(&pump);
  while ((!(WiFi.status() == WL_CONNECTED))){
    delay(300);
    Serial.print("..");

  }
  Serial.println("Connected");

}
```

```
void loop()
{
  phh();
/*int n1 = random(10, 20);
int n2 = random(10, 20);
int n3 = random(10, 20);*/
int t = dht.readTemperature();
int h = dht.readHumidity();
int m=analogRead(36);
int mf=m/4;
Serial.println(mf);
//delay(1000);
    if(mf<800)
    {
      textt.publish("Please Motor stop");
      digitalWrite(pumpp,HIGH);
    }
    if(mf>850)
    {
      textt.publish("Plant Need Water !!!");
      digitalWrite(pumpp,LOW);
    }
    if (MQTT_connect()) {
    humidity.publish(h);
    moisture.publish(mf);
    ph.publish(phValue);
    temp.publish(t);
    //N.publish(n1);
    //P.publish(n2);
    //K.publish(n3);
    delay(15000);
     Adafruit_MQTT_Subscribe *subscription_name;
      while ((subscription_name = mqtt.readSubscription(5000))) {
        if (subscription_name == &pump) {
          String myString = String(((char *)pump.lastread));
          Serial.println(myString);
          if(myString=="OFF")
```

```
          {
            digitalWrite(pumpp,HIGH);
          }
           if(myString=="ON")
          {
            digitalWrite(pumpp,LOW);
          }
          //Serial.println(((char *)pump.lastread));
        }

      }
    }

}

void phh()
{

 for(int i=0;i<10;i++){
   buf[i]=analogRead(SensorPin);
   delay(10);
  }
  for(int i=0;i<9;i++){
   for(int j=i+1;j<10;j++){
   if(buf[i]>buf[j]){
       tempp=buf[i];
       buf[i]=buf[j];
       buf[j]=tempp;
     }
   }
  }
  avgValue=0;
  for(int i=2;i<8;i++)avgValue+=buf[i];

  phValue=(float)avgValue*5.0/1024/6;
  phValue = -5.70 * phValue + calibration_value;
  Serial.println(phValue);
```

```
}

#include "Adafruit_MQTT.h"
#include "Adafruit_MQTT_Client.h"
#include <ESP8266WiFi.h>
#include <SoftwareSerial.h>
#include <Wire.h>

#define RE D3
#define DE D2

const byte nitro[] = {0x01,0x03, 0x00, 0x1e, 0x00, 0x01, 0xe4, 0x0c};
const byte phos[] = {0x01,0x03, 0x00, 0x1f, 0x00, 0x01, 0xb5, 0xcc};
const byte pota[] = {0x01,0x03, 0x00, 0x20, 0x00, 0x01, 0x85, 0xc0};

byte values[11];
SoftwareSerial mod(D5,D6);
#define AIO_SERVER      "io.adafruit.com"
#define AIO_SERVERPORT  1883
#define AIO_USERNAME  "jaiadityaghorpade1715"
#define AIO_KEY  "aio_pLFJ62SB4wlNJYJOU5L3i6aceIIj"
WiFiClient client;

Adafruit_MQTT_Client mqtt(&client, AIO_SERVER, AIO_SERVERPORT,
AIO_USERNAME, AIO_KEY);

Adafruit_MQTT_Subscribe pump = Adafruit_MQTT_Subscribe(&mqtt, AIO_USERNAME
"/feeds/pump");
boolean MQTT_connect();

boolean MQTT_connect() {  int8_t ret; if (mqtt.connected()) {    return true; }  uint8_t retries
= 3;  while ((ret = mqtt.connect()) != 0) { mqtt.disconnect(); delay(2000);  retries--;if (retries
== 0) { return false; }} return true;}

Adafruit_MQTT_Publish N = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME
"/feeds/N");
```

```
Adafruit_MQTT_Publish P = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME
"/feeds/P");
Adafruit_MQTT_Publish K = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME
"/feeds/K");
void setup()
{
 Serial.begin(9600);
 mod.begin(9600);
 pinMode(RE, OUTPUT);
 pinMode(DE, OUTPUT);
 WiFi.disconnect();
 delay(3000);
 Serial.println("START");
 WiFi.begin("Home","123456789");


 mqtt.subscribe(&pump);
 while ((!(WiFi.status() == WL_CONNECTED))){
  delay(300);
  Serial.print("..");

 }
 Serial.println("Connected");

}

void loop()
{
  byte val1,val2,val3;
 val1 = nitrogen();
 delay(200);
 val2 = phosphorous();
 delay(200);
 val3 = potassium();
 delay(200);

  if (MQTT_connect()) {
```

```
    N.publish(val1);
    P.publish(val2);
    K.publish(val3);
    delay(19000);
  }
}


byte nitrogen(){
  digitalWrite(DE,HIGH);
  digitalWrite(RE,HIGH);
  delay(10);
  if(mod.write(nitro,sizeof(nitro))==8){
    digitalWrite(DE,LOW);
    digitalWrite(RE,LOW);
    for(byte i=0;i<7;i++){
    values[i] = mod.read();
    }
  }
  return values[4];
}

byte phosphorous(){
  digitalWrite(DE,HIGH);
  digitalWrite(RE,HIGH);
  delay(10);
  if(mod.write(phos,sizeof(phos))==8){
    digitalWrite(DE,LOW);
    digitalWrite(RE,LOW);
    for(byte i=0;i<7;i++){
    values[i] = mod.read();
    }
  }
  return values[4];
}

byte potassium(){
```

```
  digitalWrite(DE,HIGH);
  digitalWrite(RE,HIGH);
  delay(10);
  if(mod.write(pota,sizeof(pota))==8){
    digitalWrite(DE,LOW);
    digitalWrite(RE,LOW);
    for(byte i=0;i<7;i++){
    values[i] = mod.read();
    }
  }
  return values[4];
}
```