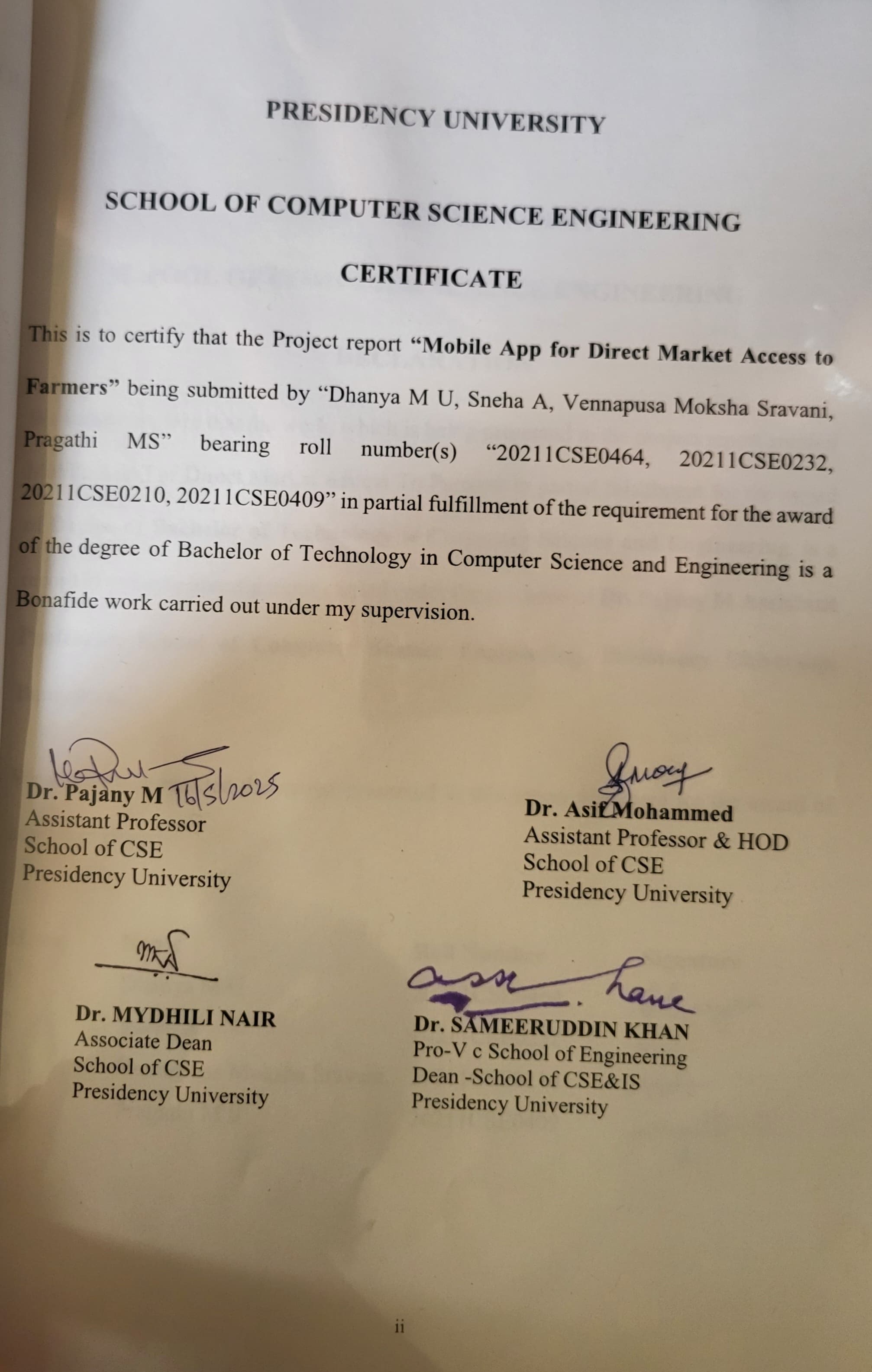
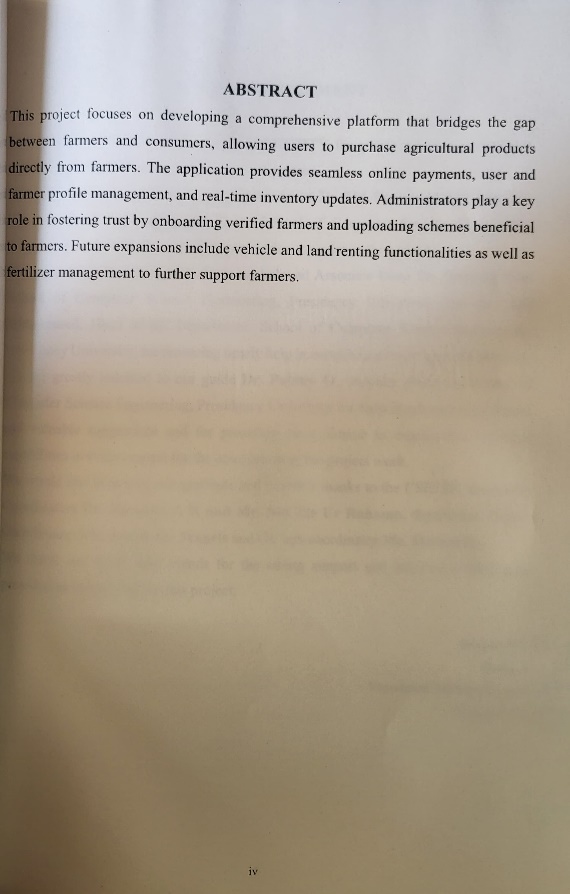
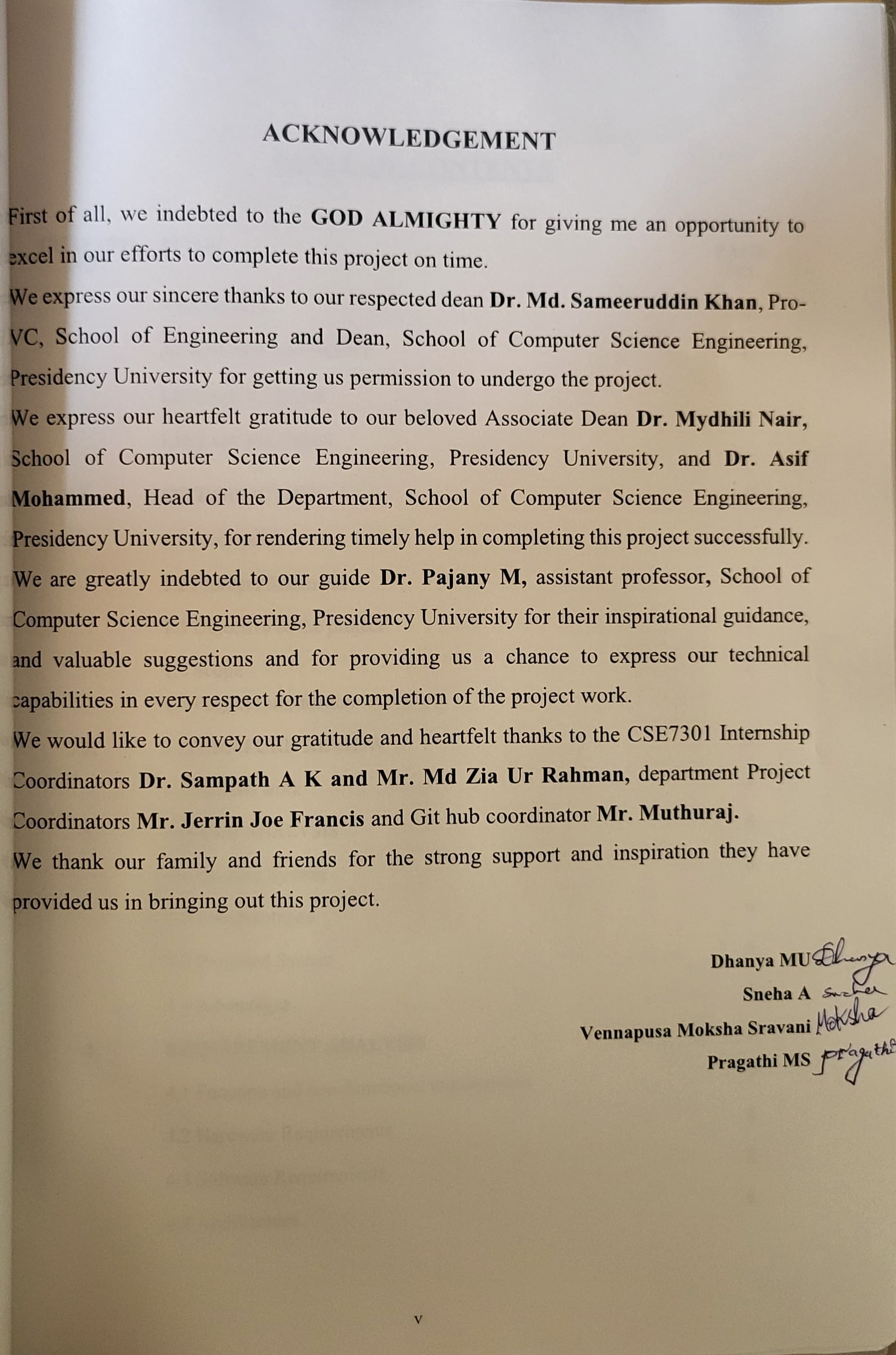


****

****



****

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
|  | **ABSTRACT**  **ACKNOWLEDGMENT**  **LIST OF TABLES**  **LIST OF FIGURES** | **iv**  **v** |
| **1.** | **INTRODUCTION** | **1** |
|  | 1.1 Motivation | 1 |
|  | 1.2 Problem Statement | 1 |
|  | 1.3 Objective of the project | 1 |
|  | 1.4 Scope  1.5 Project Introduction | 2 |
| **2.** | **LITERATURE REVIEW** | **3** |
|  | 2.1 Related work | 3 |
| **3.** | **SYSTEM ANALYSIS** | 6 |
|  | 3.1 Existing System | 6 |
|  | 3.2 Disadvantages | 6 |
|  | 3.3 Proposed System | 6 |
|  | 3.4 Advantages | 6 |
| **4.** | **REQUIREMENT ANALYSIS** | **7** |
|  | 4.1Function and non-functional requirements | 7 |
|  | 4.2 Hardware Requirements | 8 |
|  | 4.3 Software Requirements | 8 |
|  | 4.4 Architecture | 8 |
| **5.** | **SYSTEM DESIGN** | **9** |
|  | 5.1 Introduction of Input design | 9 |
|  | 5.2 UML Diagram | 10 |
|  | 5.3 Goals | 11 |
| **6.** | **IMPLEMENTATION AND RESULTS** | **18** |
|  | 6.1 Modules | 18 |
| **7.** | **SYSTEM STUDY AND TESTING** | **19** |
|  | 7.1 Feasibility study | 19 |
|  | 7.2 Types of Test & Test Cases | 20 |
|  | 7.3 Test strategy and approach | 21 |
|  | 7.4 Integration Testing | 22 |
|  | 7.5 Acceptance Testing | 22 |
|  | 7.6 Testing Cases | 22 |
| **8.** | **TIMELINE OF THE PROJECT** | **23** |
| **9.** | **OUTCOMES** | **24** |
| **10.** | **RESULTS AND DISCUSSIONS** | **25** |
| **11.** | **CONCULSION** | **26** |
| **12.** | **REFERENCES** | **27** |
| **13.** | **APPENDICES** | **28** |
|  | 1. **Pseudocode** | **28** |
|  | 1. **Screenshots** | **32** |
|  | Fig B.1 - Fig B.2 | 32 |
|  | Fig B.3 - Fig B.4 | 33 |
|  | Fig B.5 - Fig B.6 | 34 |
|  | Fig B.7 - Fig B.8 | 35 |
|  | Fig B.9 - Fig B.10 | 36 |
|  | Fig B.11 - Fig B.12 | 37 |
|  | Fig B.13 - Fig B.14 | 38 |
|  | Fig B.15 - Fig B.16 | 39 |
|  | Fig B.17 - Fig B.18 | 40 |
|  | Fig B.19 - Fig B.20 | 41 |
|  | Fig B.21 - Fig B.22 | 42 |
|  | Fig B.23 - Fig B.24 | 43 |
|  | 1. **Enclosures** | **44** |
|  | 1. Conference Paper Presented Certificates | 44 |
|  | 1. Plagiarism report of the Report | 45 |
|  | 1. Plagiarism Report of Research Paper | 46 |
|  | 1. Sustainable Development Goals | 47 |

**LIST OF TABLES**

|  |  |  |  |
| --- | --- | --- | --- |
| **SI.no** | **Table Name** | **Table Caption** | **Page No.** |
| 1 | Table 7.1 | Testing Cases | 22 |
| 2 | Table 8.1 | Project Timeline and Activities | 23 |

**LIST OF FIGURES**

|  |  |  |  |
| --- | --- | --- | --- |
| **SI.no** | **Figure Name** | **Table Caption** | **Page No.** |
|  | Fig 4.1 | Architecture | 8 |
|  | Fig 5.1 | Use case Diagram | 11 |
|  | Fig 5.2 | Class Diagram | 12 |
|  | Fig 5.3 | Sequence Diagram | 13 |
|  | Fig 5.4 | Collaboration Diagram | 14 |
|  | Fig 5.5 | Activity Diagram | 15 |
|  | Fig 5.6 | Component Diagram | 16 |
|  | Fig 5.7 | Deployment Diagram | 16 |
|  | Fig 5.8 | ER Diagram | 17 |
|  | Fig 5.9 | Data Flow Diagram | 17 |
|  | Fig 8.1 | TimeLine of Project Execution | 23 |

**CHAPTER 1**

**INTRODUCTION**

**1.1 Motivation**

The agricultural sector has long faced challenges with transparency, efficiency, and market accessibility, leaving farmers without direct avenues to reach consumers and maximize their profits. The motivation behind this project is to empower farmers by bridging the gap between them and consumers, creating an ecosystem that ensures fairness, trust, and accessibility. By integrating technology with agriculture, this platform provides farmers with tools to manage their products, streamline payments, and access beneficial schemes. This will not only uplift farmers' livelihoods but also foster a sustainable farm-to-consumer relationship, enhancing the overall agricultural economy.

**1.2 Problem Statement**

Traditional agricultural marketplaces often suffer from a lack of transparency and inefficient payment processes. Farmers are often isolated from direct market access, and consumers have limited opportunities to purchase fresh products directly from the source. This project addresses these gaps by creating a platform that connects farmers directly with consumers, ensuring transparency in product quality and prices, along with efficient payment processing.

**1.3 Objective of the Project**

The objective is to develop an application that connects consumers directly with farmers, offering a seamless and secure payment gateway, profile management, and inventory system. The platform will allow users to view and purchase products from verified farmers. Provide farmers with tools to manage their product listings, receive payments, and stay informed about government schemes. Enable administrators to maintain trust within the system by onboarding verified farmers and providing them access to beneficial schemes.

**1.4 Scope**

The project focuses on creating a user-friendly mobile application that facilitates transactions between farmers and consumers. Users can browse, purchase, and make payments for farm products, while farmers can manage their inventory and receive payments directly via UPI. Admins will manage farmer onboarding and upload relevant agricultural schemes. Future extensions include vehicle renting for transporting produce, land renting, and fertilizer management systems.

**1.5 Project Introduction**

Agriculture is the backbone of many economies, yet farmers often struggle with limited access to markets and inefficient processes that hinder their financial growth. This project,

**Mobile App for Direct Market Access for Farmers**, addresses these challenges by offering a comprehensive platform where farmers can sell their products directly to consumers. The system integrates user-friendly features such as UPI payments, real-time inventory management, and profile handling for both users and farmers. By fostering direct connections between consumers and farmers, along with offering farmers access to government schemes, this project ensures transparency and efficiency, promoting a fair and sustainable agricultural marketplace.

**CHAPTER 2**

**LITERATURE SURVEY**

**2.1 Related Work**

1. **Pranav Shriram; Sunil Mhamane | Android App to Connect Farmers to Retailers and Food Processing Industry | 15-16 November 2018**

Mobile internet is revolutionizing the way farmers can connect directly with consumers and food processing industries, allowing them to sell their products efficiently. This paper introduces a mobile application designed to deliver real-time, updated market information to farmers through an easy-to-use interface. The app includes native language support to simplify transactions for farmers. It allows farmers to act as both sellers and buyers, facilitating the purchase and sale of agricultural goods. Market prices, provided by data.gov.in, help maintain control over buying and selling prices. To streamline browsing, the system offers filters for product selection, making it easier for users to navigate a wide variety of products. The system addresses the challenges farmers face in selling their goods by offering a transparent and fair platform. Consumers can browse a broad selection of products, apply price filters, and select items based on their needs. Additionally, location-based features allow users to find products close to them. The core objective of the system is to meet the needs of both farmers and consumers, fostering a transparent and equitable agricultural marketplace.

1. **L.A. Imalka; K.G.A. Gunawardana; K.M.S.K. Kodithuwakku; H.K.E. Arachchi; S.M.B. Harshanath | Farming Through Technology Driven Solutions For Agriculture Industry Ceylon E-Agro mobile application-find technology based solutions for agricultural problems | 16-18 September 2022**

Agriculture is the backbone of many developing countries, with over 60% of the population relying on this sector. This project focuses on addressing the challenges faced by maize farmers in Sri Lanka. Farmers in this region are currently grappling with various issues related to maize cultivation. The proposed mobile application aims to assist maize farmers by offering solutions to problems such as pests, diseases, and fire threats on the farm. The app features an AI-based Agri Agent that provides real-time solutions, connecting farmers directly with buyers on a single platform. It also includes a price forecasting and price index feature to help farmers make informed decisions. Additionally, the app integrates IoT-based smart farming capabilities to monitor soil moisture and quality, ensuring optimal conditions for maize cultivation.

1. **R. Ranjana; T. Subha; Pravin Kumar P; Sneka L; Varsha S; Jothishree N |Integrated App for Farmers - Agreliance | 16-17 December 2021**

In today's rapidly advancing world, telecommunication technology has bridged the gap, allowing people to stay connected even over vast distances. An integrated mobile app has been designed to improve both the financial and mental well-being of farmers. During the COVID-19 pandemic, telehealth care systems have proven to be an effective alternative to traditional in-person counseling, offering mental health support to farmers through video conferencing. This method has been shown to be just as effective as face-to-face consultations. In addition to addressing emotional concerns, experts can also provide guidance on crop diseases, offering consultations to farmers through pre-booked sessions. Given the economic challenges faced due to the global pandemic, selling farm products through online platforms can significantly boost economic growth, reduce costs, and eliminate the role of middlemen. Furthermore, the app helps farmers become aware of nearby loan facilities, which can play a crucial role in their survival and sustainability in agriculture.

1. **Niket Chauhan; M. Krishnakanth; G. Praneeth Kumar; Prerna Jotwani; Utkarsh Tandon; | Crop Shop – An application to maximize profit for farmers | 30-31 March 2019**

For many years, farmers in India have had limited freedom in selecting markets and buyers for their produce. Except for three states, all others mandate that farm produce be marketed through state-owned **mandis**, where middlemen dominate, driving up margins and exploiting farmers. According to Goldman Sachs, these middlemen control the agricultural market, siphoning off the majority of the profits, leaving farmers with very little. Farmers work tirelessly, borrowing money to buy fertilizers and seeds, and yet they receive minimal returns on their hard work. To address this issue, we propose a solution that directly connects farmers with retailers and consumers, bypassing the middlemen. Our **mobile application** will create a platform for farmers to sell their products directly to retailers or consumers, ensuring they receive a fair price for their goods. This system enables farmers to gain a larger share of the profits, while retailers and consumers can access products at lower prices, fostering a more equitable agricultural market.

1. **Aina Marie Joseph; Nurfauza Jali; Amelia Jati Robert Jupit; Suriati Khartini Jali | eMarket for Local Farmers | 23-25 November 2021**

The COVID-19 pandemic has significantly impacted local farmers, limiting their ability to generate income outside of selling crops to customers. Additionally, consumers have faced difficulties accessing fresh, quality produce other than from supermarkets. This paper has two main objectives: first, to evaluate the existing manual systems used by consumers to acquire fresh produce during the pandemic, and second, to design and implement a mobile application for local farmers to sell their fresh produce directly to consumers via a delivery system. The paper proposes the development of an online marketplace for local farmers, leveraging the widespread use of mobile applications that have reshaped how people access essential goods during the pandemic. To build this platform, Rapid Application Development (RAD) methodology was used. The pre-development stage included a survey with 20 local farmers and 20 potential customers to gather their opinions on the app. Afterward, usability testing was conducted with 10 local farmers and 10 university students to evaluate the app’s functionality.The outcome of these tests led to the development of a farmer’s market application, which has been positively received by both farmers and consumers, offering an effective solution to the challenges faced by both groups during the pandemic.

The application provides a seamless platform for farmers to sell their fresh produce directly to consumers, cutting out intermediaries and ensuring fair prices for both parties. Additionally, the app offers convenient delivery options, making it easier for consumers to access high-quality local produce. By fostering a direct connection between farmers and consumers, this solution helps stimulate the local agricultural economy while providing a convenient and reliable source of fresh food for the community.

**CHAPTER 3**

**SYSTEM ANALYSIS**

**3.1 Existing System:**

Current platforms in agriculture-based commerce either offer limited direct interaction between farmers and consumers or are complicated by lengthy payment processes. Farmers often face challenges in listing their products and managing payments. Additionally, government schemes that benefit farmers are often underutilized due to a lack of awareness and access.

**3.2 Disadvantages**

* Fragmented processes lead to inefficiencies in connecting farmers with consumers.
* Lack of seamless payment solutions hinders transaction completion.
* Limited visibility of government schemes results in missed opportunities for farmers.

**3.3 Proposed System**

The proposed system integrates a unified platform for farmers, users, and administrators. Users can browse and purchase products while completing payments through UPI-based gateways. Farmers can manage inventory, update product listings, and receive payments directly into their accounts. Administrators will onboard trusted farmers, upload helpful schemes, and manage the platform's ecosystem.

**3.4 Advantages**

* A cohesive system for purchasing agricultural products directly from farmers.
* Payments are processed securely through UPI-like gateways, reducing delays.
* Farmers can easily add and update their products.
* Admins can upload schemes beneficial to farmers, ensuring they are accessible.
* Future updates will allow for vehicle and land renting, as well as fertilizer management

**CHAPTER 4**

**REQUIREMENT ANALYSIS**

**4.1 Function and non-functional requirements**

Requirement’s analysis is very critical process that enables the success of a system or software project to be assessed. Requirements are generally split into two types: Functional and non-functional requirements.

Functional Requirements: These are the requirements that the end user specifically demands as basic facilities that the system should offer. All these functionalities need to be necessarily incorporated into the system as a part of the contract. These are represented or stated in the form of input to be given to the system, the operation performed and the output expected. They are basically the requirements stated by the user which one can see directly in the final product, unlike the non-functional requirements.

Examples of functional requirements:

1) Authentication of user whenever he/she logs into the system

2) System shutdown in case of a cyber-attack

**Non-functional requirements:** These are basically the quality constraints that the system must satisfy according to the project contract. The priority or extent to which these factors are implemented varies from one project to other. They are also called non-behavioral requirements.

They basically deal with issues like:

• Portability

• Security

• Maintainability

• Reliability

• Scalability

• Performance

• Reusability

• Flexibility

Examples of non-functional requirements:

1) Emails should be sent with a latency of no greater than 12 hours from such an activity.

2) The processing of each request should be done within 10 seconds

3) The site should load in 3 seconds whenever of simultaneous users are > 10000

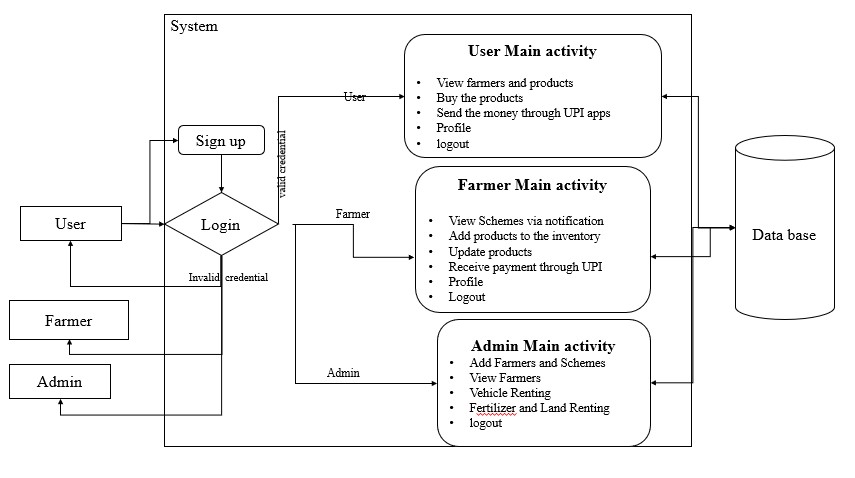
**4.2 Hardware Requirements**

* Processor - I3/Intel Processor
* RAM - 8 GB
* Hard Disk - 1TB

**4.3 Software Requirements**

* Operating System - Windows 10
* JDK - java
* Plugin -Kotlin
* SDK - Android
* IDE -Android studio
* Database` - server script, my sql

**4.4 Architecture**

****

**Fig 4.1:** The image illustrates a system architecture for the rural commerce application, which connects users, farmers, and administrators.

**CHAPTER 5**

**SYSTEM DESIGN**

**5.1 Introduction of Input design**

**1) INPUT DESIGN**

Input design serves as the link between the information system and the user. It involves developing specifications and procedures for data preparation, ensuring transaction data is in a usable form for processing. Data can be entered by the system reading from written or printed documents, or directly through manual entry by users. The focus of input design is on controlling the volume of input, minimizing errors, avoiding delays, reducing unnecessary steps, and keeping the process as simple as possible. The design should prioritize security, ease of use, and maintain privacy.

Input design considers the following factors:

* **What data should be provided as input?**
* **How should the data be arranged or coded?**
* **The dialog or guidance provided to assist users in inputting data.**
* **Methods for validating input and the steps to follow when errors occur.**

**2) OBJECTIVES**

**1.** Input design is the process of transforming a user-oriented description of data into a format that can be processed by a computer-based system. This design is crucial for minimizing errors during data entry and guiding management in extracting accurate information from the computerized system.

**2.** The design is achieved by creating user-friendly data entry screens capable of handling large volumes of data. The goal is to simplify data entry and ensure it is error-free. The input screens are structured to allow all necessary data manipulations and to provide record viewing capabilities.

**3.** When data is entered, its validity is checked. Data is entered via screens, with appropriate messages displayed as needed to guide the user. This helps prevent confusion and ensures the system is intuitive to use. Therefore, the objective of input design is to create an easy-to-follow input layout that enhances user experience and ensures accurate data entry.

**3) OUTPUT DESIGN**

A high-quality output is one that satisfies the needs of the end user and presents information in a clear and understandable manner. In any system, the results of processing are communicated to users and other systems through outputs. Output design focuses on determining how information will be displayed for immediate use and how hard copy outputs will be provided. It is a crucial and direct means of delivering information to the user. Efficient and well-thought-out output design improves the system’s ability to aid in user decision-making.

1. Designing computer outputs should follow a structured, thoughtful approach, ensuring that the right outputs are created. Each output element must be designed in a way that users can easily understand and utilize. When analyzing output design, the specific outputs needed to meet user requirements should be identified.
2. Choose appropriate methods for presenting information.
3. Develop documents, reports, or other formats to display information generated by the system.

The output format in an information system should serve one or more of the following purposes:

* Provide information about past activities, current status, or future projections.
* Highlight significant events, opportunities, problems, or warnings.
* Initiate an action.
* Confirm an action.

**5.2 UML Diagram**

UML, or Unified Modeling Language, is a standardized general-purpose modeling language used in object-oriented software engineering. It was developed and is maintained by the Object Management Group (OMG).

The main goal of UML is to provide a common language for modeling object-oriented software systems. It currently consists of two core parts: a meta-model and a notation. Future enhancements may include methods or processes associated with UML.

UML is used to specify, visualize, construct, and document elements of software systems, as well as for business and non-software system modeling. It incorporates proven engineering practices for modeling large, complex systems.

**5.3 GOALS:**

The main objectives behind the design of UML are:

1. To offer a user-friendly and expressive visual modeling language for creating and sharing meaningful system models.
2. To include mechanisms for extending and customizing core concepts.
3. To remain independent of any specific programming language or development methodology.
4. To establish a formal foundation for understanding the modeling language.
5. To promote the expansion of the object-oriented tools market.
6. To support advanced development concepts like collaborations, frameworks, design patterns, and components.
7. To incorporate industry best practices into modeling.

**5.4 USE CASE DIAGRAM:**

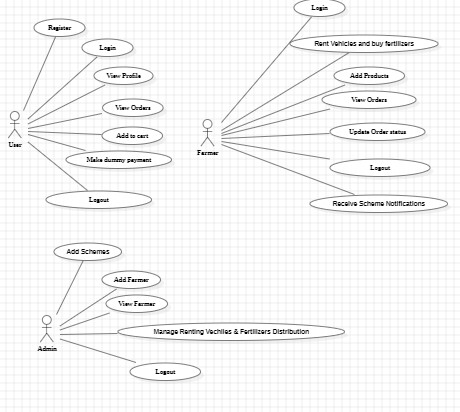
****A use case diagram in UML is a type of behavioral diagram created through use-case analysis.  
It models the functional aspects of a system by showing interactions between users (actors) and system use cases.

Fig 5.1: This diagram Shows interactions between users, farmers, admins, and system features like login, order placement, and scheme viewing.

A use case diagram provides a graphical overview of a system's functionality by illustrating actors, their goals (use cases), and the relationships between them.  
Its main purpose is to show which system functions are performed for each actor.  
It also highlights the roles actors play and how they interact with the system.

**5.5 CLASS DIAGRAM:**

In software engineering, a class diagram in UML is a static structure diagram that illustrates a system’s classes, along with their attributes, methods, and relationships.

It shows how classes are connected and what information each class holds.

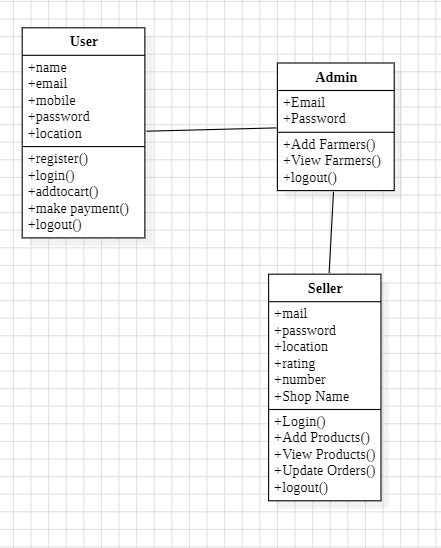
Class diagrams help in understanding the overall architecture and design of the system.

Fig 5.2: This diagram displays system classes, their attributes, methods, and relationships, representing the app's structure.

**5.6 SEQUENCE DIAGRAM:**

A sequence diagram in UML is an interaction diagram that shows the order of message exchanges between objects over time. It's also known as an event diagram or timing diagram.

It helps visualize the flow of logic in a system.

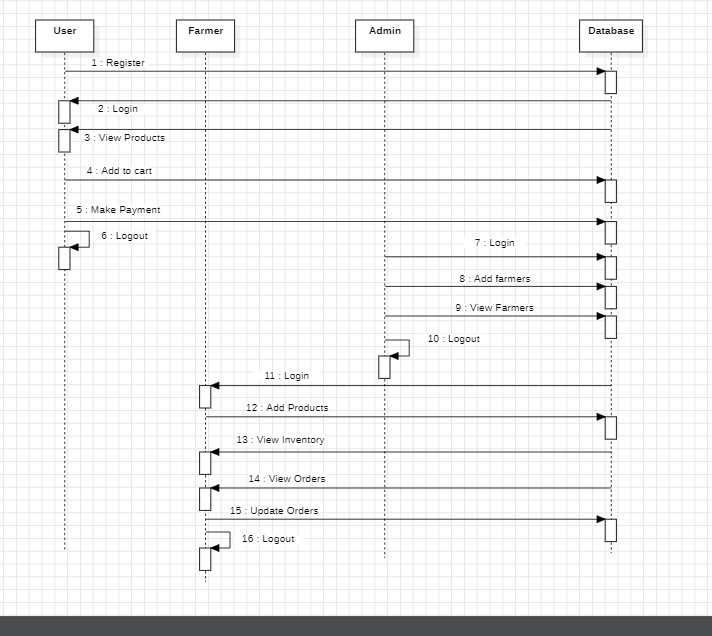
Objects are represented horizontally, and interactions are shown vertically in sequence.

Fig 5.3: This diagram Illustrates the order of messages exchanged during processes like login and order handling.

The sequence diagram demonstrates how users, farmers, and admins interact with the system over time. It highlights the order of operations during key processes such as login, placing orders, updating inventory, and scheme access. The interaction begins with the actor sending a request and continues through a series of method calls between system objects. This helps visualize the flow of control and ensures that the system functions as expected in real-time scenarios.

**5.7 COLLABORATION DIAGRAM:**

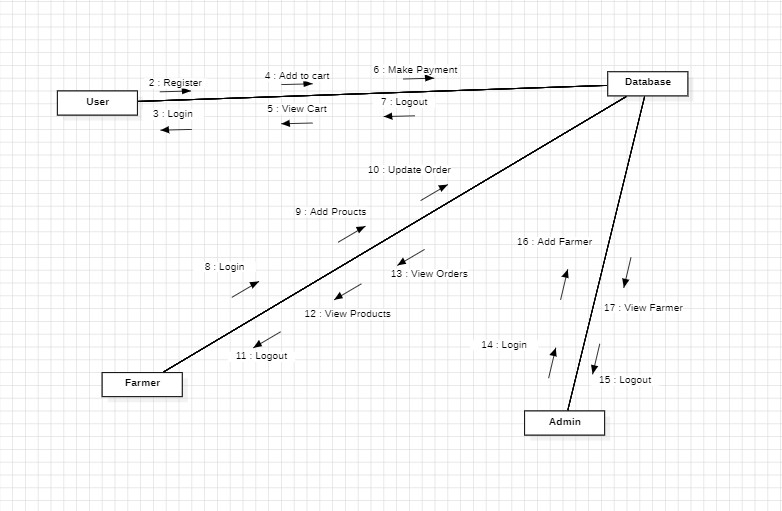
In a collaboration diagram, the sequence in which methods are called is represented using a specific numbering system. These numbers reflect the order of method invocations, similar to what is shown in a sequence diagram. However, unlike sequence diagrams, which primarily focus on the chronological flow of messages, collaboration diagrams also illustrate the structural relationships and organization among objects. Using the order management system as an example, the diagram highlights how objects are linked and how they interact through method calls in a defined sequence.

Fig 5.4: This diagram shows object interactions and message flow for tasks like product ordering and scheme approval.

The Collaboration Diagram further highlights how system objects coordinate with each other to fulfill specific functionalities such as placing an order or updating inventory. Each object in the diagram is connected using links that represent relationships, and numbered arrows indicate the sequence of method calls. For example, when a user places an order, the system invokes methods on the product, order, and payment handler objects in a specific order. This visualization helps developers understand how responsibilities are distributed among components. By modeling these interactions clearly, the diagram supports robust and maintainable system development.

**5.8 ACTIVITY DIAGRAM:**

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modelling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

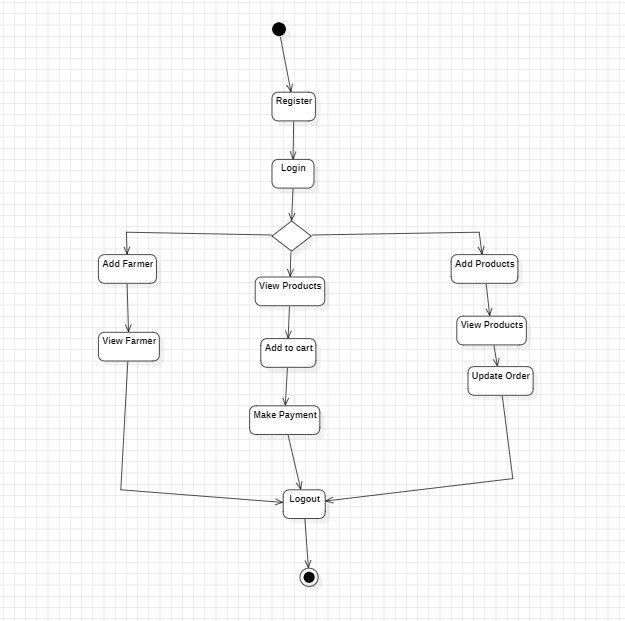


Fig 5.5: This diagram represents the step-by-step workflow for actions such as ordering products and managing inventory.

The activity diagram outlines the dynamic flow of actions within the system, such as user login, product selection, and order placement. It helps visualize the process sequence and decision points involved in user and farmer interactions.

**5.9 COMPONENT DIAGRAM:**

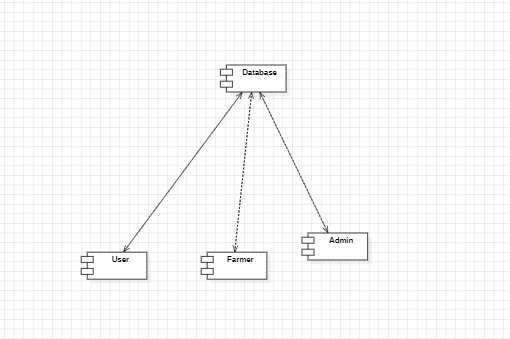
****

Fig 5.6: This diagram depicts software components (UI, backend, database) and how they connect in the system.

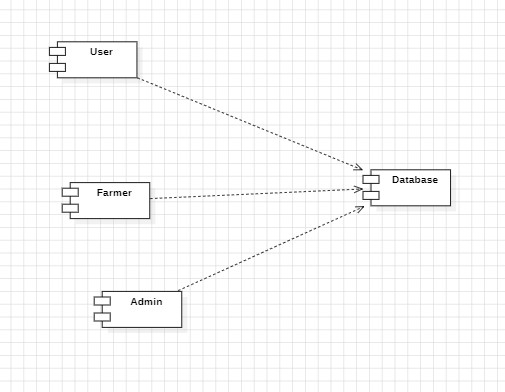
* 1.  **DEPLOYMENT DIAGRAM:**

Fig 5.7: This diagram shows the physical setup of the app across devices like mobile phones and servers.

**5.11 ER Diagram:**

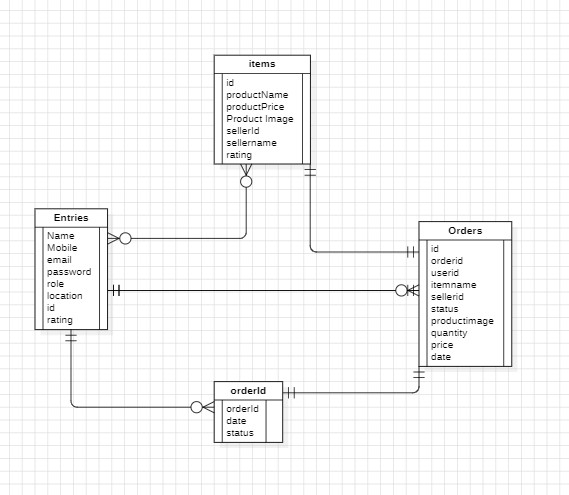
****

Fig 5.8: This diagram defines entities like User, Farmer, and Order, and how they relate in the system’s database.

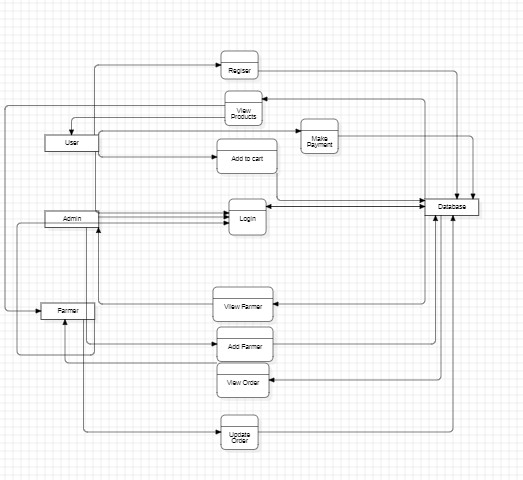
* 1. **Data Flow Diagram:** 

Fig5.9: This diagram illustrates the flow of data between users, farmers, admins, and system processes like login, ordering, and scheme management.

**CHAPTER 6**

**IMPLEMENTATION AND RESULTS**

**6.1 Modules**

1. **User Module:**

Users will sign up, log in, and browse a list of farmers and their products. After selecting products, users can purchase them and complete payments through a UPI-like dummy payment gateway. Users can manage their profiles, track orders, and view their order history.

1. **Farmer Module:**

Farmers can log in, manage their profiles, and add or update products in the inventory. Farmers will receive notifications for new orders and can fulfil them. Payments will be credited directly to their UPI accounts. Farmers can view schemes uploaded by the admin, with notifications available in the app.

1. **Admin Module:**

Admins will onboard trusted farmers, creating a verified environment for users. They will upload and manage schemes that are beneficial for farmers. Admins will also be responsible for suggesting and providing references for applying to these schemes, fostering an efficient farming ecosystem. They can also monitor platform performance and manage system operations.

**CHAPTER 7**

**SYSTEM STUDY AND TESTING**

**7.1 Feasibility study**

**1. FEASIBILITY STUDY**

During this phase, the project's viability is assessed, and a preliminary business proposal is developed, outlining a basic plan along with estimated costs. The goal of system analysis at this stage is to determine whether the proposed solution is practical and manageable for the organization. Conducting a feasibility study helps confirm that the system will not impose unnecessary strain on resources. To perform this evaluation effectively, a general understanding of the core system requirements is necessary.

The feasibility analysis focuses on three primary aspects:

* Economic Feasibility
* Technical Feasibility
* Social Feasibility

**2. ECONOMICAL FEASIBILITY**

This analysis evaluates the financial implications the system may have on the organization. Since the company has a limited budget for research and development, all expenses need to be justified. The project stayed within the allocated budget, primarily because most of the technologies utilized were open-source or free. Only specific customized components required purchase.

**3.** **TECHNICAL FEASIBILITY**

This analysis focuses on determining the technical feasibility by examining the system’s technical requirements. It is essential that the system does not heavily rely on the organization’s existing technical infrastructure, as this could place unnecessary strain on the client. The proposed system is designed to function with basic technical support, requiring little to no adjustments for successful implementation.

1. **SOCIAL FEASIBILITY**

This part of the study evaluates how well the users are likely to accept and adapt to the new system. It involves training users to operate the system effectively and ensuring they view it as a helpful tool rather than a threat. User acceptance largely depends on how well they are introduced to and educated about the system. Building their confidence is crucial, enabling them to provide valuable feedback, which is encouraged since they are the primary end-users.

1. **SYSTEM TESTING**

The primary goal of testing is to identify errors within a system. It involves examining the product thoroughly to uncover any potential faults or weaknesses. Testing helps verify the functionality of individual components, integrated modules, and the complete system. It ensures that the software performs according to specified requirements and meets user expectations without critical failures. Different types of testing are conducted, each serving a unique purpose in the overall validation process.

**7.2 Types of test & Test Cases**

**1. UNIT TESTING**

Unit testing focuses on creating test cases to verify that the internal logic of a program works as intended and that the inputs produce correct outputs. It checks all code paths and decision branches to ensure proper functionality. This type of testing is performed on individual components after their development and before they are integrated with other units. As a form of structural and intrusive testing, it relies on understanding the code's internal structure. Unit tests validate specific business processes or configurations, ensuring that each path functions correctly according to defined requirements, with clear inputs and expected outcomes.

**2. INTEGRATION TESTING**

Integration testing focuses on verifying that combined software modules work together as a unified system. It is event-driven and emphasizes the functionality and output of user interfaces such as screens and fields. While unit testing confirms that individual components function correctly, integration testing ensures that their interaction is smooth and consistent. Its primary goal is to identify issues that may emerge when separate components are integrated.

1. **FUNCTIONAL TEST**

Functional testing involves systematically verifying that all specified functions operate as expected, based on business requirements, technical specifications, user documentation, and system manuals. This type of testing primarily focuses on the following aspects:

* 1. **Valid Input**: The system must correctly accept and handle defined categories of acceptable input.
  2. **Invalid Input**: The system must appropriately reject and manage predefined categories of unacceptable input.
  3. **Core Functions**: All key functionalities identified in the system must be tested to ensure they perform correctly.
  4. **Outputs**: The system must produce correct and expected outputs for each class of processed data.
  5. **System Interfaces and Procedures**: Any external systems or internal procedures that interact with the application must be successfully triggered and evaluated.

Functional test planning emphasizes critical requirements, significant features, and unique test scenarios. It also ensures comprehensive testing by covering business workflows, data fields, standard operations, and sequences of related processes. Throughout the testing phase, new test cases may emerge, and the effectiveness of existing ones is regularly reviewed and updated to maintain test coverage and relevance.

1. **SYSTEM TEST**

System testing verifies that the fully integrated software system functions according to its specified requirements. It evaluates the complete configuration to confirm it produces expected and consistent outcomes. A typical example of this is a system integration test focused on verifying system configurations. This level of testing relies on documented process flows and descriptions, with particular attention given to predefined process connections and integration points between components.

1. **WHITE BOX TESTING**

White Box Testing is a method where the tester has insight into the internal structure, logic, and code of the software being tested. This approach allows the tester to design test cases based on the software's internal workings rather than just its outputs. It is especially useful for examining components or paths that are not accessible through black box testing methods.

1. **BLACK BOX TESTING**

Black Box Testing is a technique where the software is tested without any knowledge of its internal code, structure, or implementation details. The tester interacts with the application by providing inputs and observing outputs, without knowing how those outputs are generated. This type of testing is based on specifications or requirement documents and focuses solely on verifying that the software behaves as expected from an external user's perspective. The system is treated as a "black box," meaning its internal workings remain unseen.

1. **UNIT TESTING:**

Unit testing is typically performed during the combined coding and unit testing phase of the software development lifecycle. However, in some cases, these activities may be carried out separately, with unit testing occurring after the coding phase is complete. This type of testing focuses on verifying the functionality of individual components or units of code in isolation.

**7.3 Test Strategy and Approach**  
Manual field testing will be conducted, and detailed functional test cases will be developed.

**a) Test Objectives**

* All form fields must function as intended.
* Navigation through links must correctly lead to the respective pages.
* Input screens, system messages, and response times should not experience delays.

**b) Features to be Tested**

* Ensure inputs conform to the required format.
* Duplicate entries must be prevented.
* Each hyperlink should correctly redirect to the designated page.

**7.4 Integration Testing**

Integration testing involves the step-by-step testing of multiple combined software modules on a single platform, aimed at identifying issues arising from interface mismatches.  
The goal is to validate that all software components or applications—whether within a system or across systems—interact correctly without generating errors.

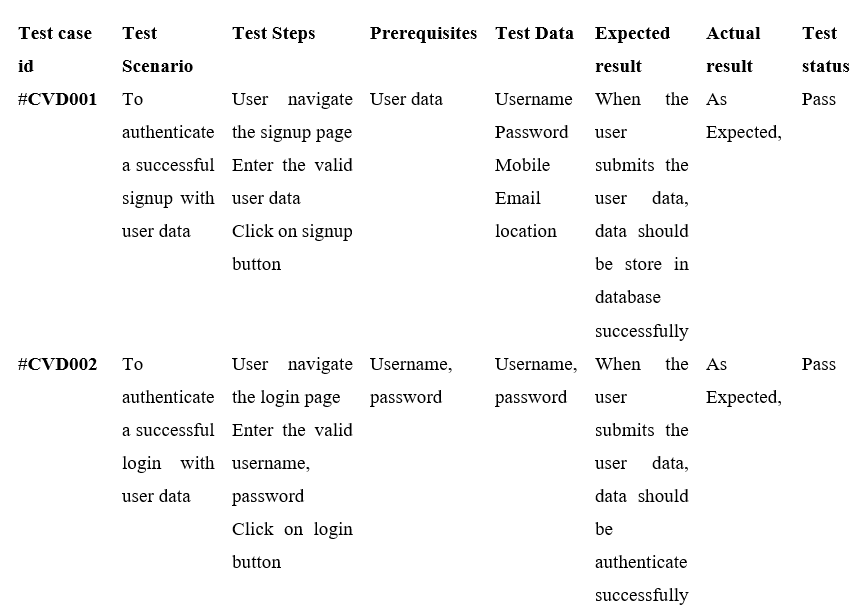
**Test Outcome**: All outlined test cases executed successfully, with no errors or defects found.

**7.5 Acceptance Testing**  
User Acceptance Testing (UAT) represents a crucial phase where end users are actively involved to confirm that the system adheres to the defined functional specifications.  
It serves as a final validation to ensure the solution is ready for deployment.

**Test Outcome**: All defined acceptance test cases were completed successfully, and no defects were observed.

**7.6 Testing Cases**

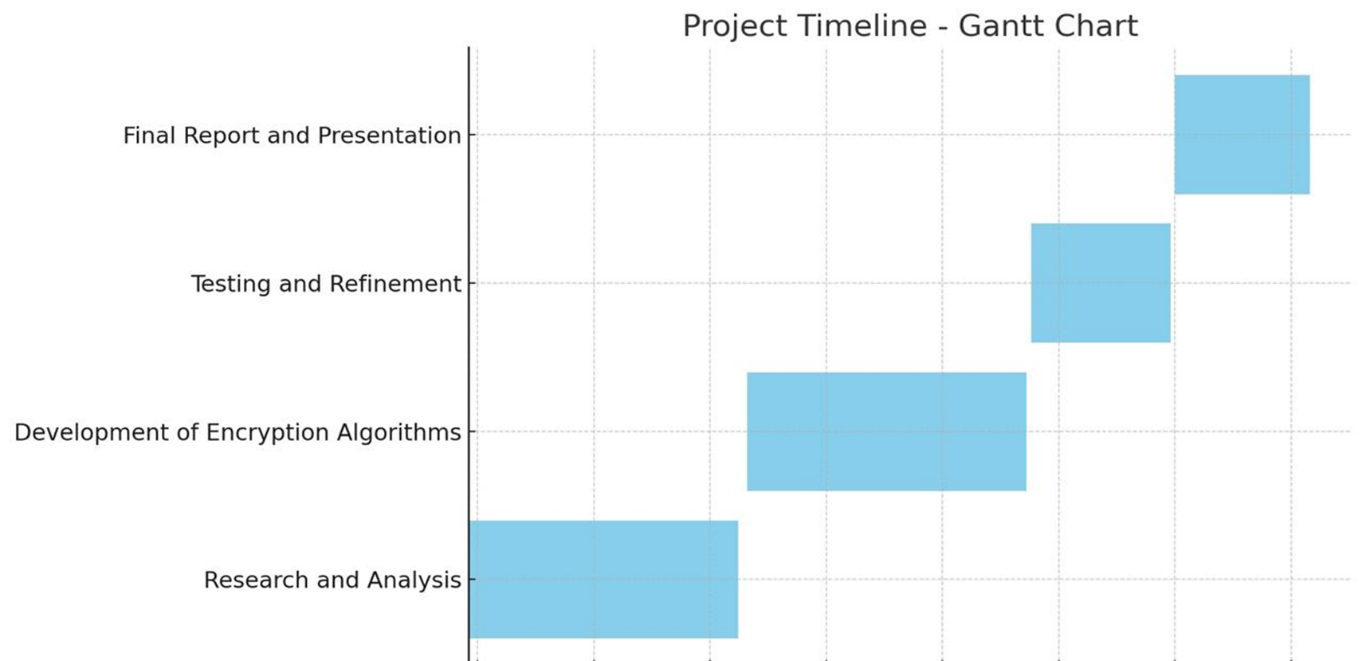
Table 7.1: Displays various scenarios, steps, expected outcomes, and actual results.



**CHAPTER 8**

**TIMELINE FOR EXECUTION OF PROJECT**

**(GANTT CHART)**



**Fig 8.1 TimeLine for the Execution of the Project**

**Table 8.1: Project Timeline and Activities**

|  |  |  |
| --- | --- | --- |
| **Phase** | **Dates** | **Activity** |
| Review – 0 | 29 – Jan – 2025 To  31 – Jan – 2025 | Research and Analysis |
| Review – 1 | 18 – Feb – 2025 To  21 – Feb – 2025 | Applying Algorithms & working on project |
| Review – 2 | 17 – Mar – 2025 To  21 – Mar – 2025 | Development |
| Review – 3 | 16 – Apr – 2025 To  19 – Apr – 2025 | Testing and Refinement |
| Final Viva – voce | 10 – May – 2025 To  17 – May – 2025 | Final Report and Presentation |

**CHAPTER 9**

**OUTCOMES**

1. **Direct Market Access**:

Farmers can sell produce directly to consumers, eliminating middlemen and increasing profits.

1. **User-Friendly Platform**:

The mobile app provides easy navigation, product filtering, and multilingual support.

1. **Improved Market Efficiency**:

Real-time inventory management and price transparency enhance market operations.

1. **Seamless Transactions**:

UPI-based payment integration ensures secure and instant payments.

1. **Scalability & Future Expansion**:

The system can integrate IoT-based smart farming and additional marketplace features.

1. **Government Scheme Awareness**:

Farmers can access relevant schemes, maximizing available benefits.

**CHAPTER 10**

**RESULTS AND DISCUSSIONS**

1. **Direct Farmer-to-Consumer Sales:**

The system enables farmers to sell their produce directly, removing middlemen and ensuring better profits.

1. **Seamless Payment Processing:**

UPI-based transactions provide secure, fast, and hassle-free payments.

1. **Efficient Inventory Management:**

Farmers can easily update product listings, track sales, and manage stock in real-time.

1. **Admin Control & Scheme Access:**

Administrators can onboard farmers, update government schemes, and monitor platform security.

1. **Enhanced Payment Security:**

The integration of UPI-based payments has improved transaction reliability and efficiency.

1. **Improved Accessibility:**

Multilingual support and a user-friendly interface have made the platform accessible to a larger farmer base.

1. **Challenges in Implementation:**

Internet connectivity in rural areas and limited digital literacy among farmers remain key obstacles.

1. **Impact on Agriculture Market:** The system increases transparency, efficiency, and sustainability, ultimately benefiting both farmers and consumers.

The proposed system successfully bridges the gap between farmers and consumers, ensuring fair pricing, secure transactions, and improved market access

**CHAPTER 11**

**CONCLUSION**

Agriculture remains a crucial sector, yet farmers often struggle with unfair pricing, market inefficiencies, and limited access to direct consumers. This project addresses these challenges by providing a **digital marketplace** where farmers can sell their produce directly, ensuring **fair trade and transparency**. With **UPI-based payments, real-time inventory management, and government scheme integration**, the platform simplifies transactions and boosts financial security for farmers. The user-friendly mobile interface, coupled with **multilingual support**, enhances accessibility, making it easier for farmers to adopt digital solutions. While challenges like **internet connectivity and digital literacy** remain, they can be mitigated through awareness programs and infrastructure improvements. Future enhancements, such as **IoT-based smart farming and logistics support**, will further optimize the system. Overall, this initiative **empowers farmers, strengthens rural economies, and promotes sustainable agricultural practices**, paving the way for a more efficient and equitable marketplace.

Overall, the implementation of this project empowers farmers by enabling **direct market access, fair pricing, and efficient transactions** through a digital platform

**REFERENCES**

1. Pranav Shriram; Sunil Mhamane | *Android App to Connect Farmers to Retailers and Food Processing Industry | 15-16 November 2018*
2. L.A. Imalka; K.G.A. Gunawardana; K.M.S.K. Kodithuwakku; H.K.E. Arachchi; S.M.B. Harshanath | *Farming Through Technology Driven Solutions For Agriculture Industry Ceylon E-Agro mobile application-find technology-based solutions for agricultural problems | 16-18 September 2022*
3. R. Ranjana; T. Subha; Pravin Kumar P; Sneka L; Varsha S; Jothishree N | *Integrated App for Farmers - Agreliance | 16-17 December 2021*
4. Niket Chauhan; M. Krishnakanth; G. Praneeth Kumar; Prerna Jotwani; Utkarsh Tandon | *Crop Shop – An application to maximize profit for farmers | 30-31 March 2019*
5. Aina Marie Joseph; Nurfauza Jali; Amelia Jati Robert Jupit; Suriati Khartini Jali | *eMarket for Local Farmers | 23-25 November 2021*

**APPENDIX-A**

**PSEUDOCODE**

START

// Initialization

Initialize database

Initialize UPI payment gateway

Initialize user interface (Android App)

Initialize roles: USER, FARMER, ADMIN

// Login Page

Display "Login as: User / Farmer / Admin"

Input login\_credentials

Authenticate user

IF login is successful THEN

Redirect to respective dashboard

ELSE

Display "Invalid credentials"

// Admin Module

IF role is ADMIN THEN

Display "Admin Dashboard"

WHILE admin is logged in DO

Display options:

1. Add Farmer

2. View Farmers

3. Add/View Schemes

4. View Status Reports

5. Logout

SWITCH selected option:

CASE "Add Farmer":

Input farmer\_details

Verify details

Add to verified\_farmers\_list

CASE "View Farmers":

Display all\_verified\_farmers

CASE "Add Scheme":

Input scheme\_details

Upload to system

CASE "View Schemes":

Display uploaded\_schemes

CASE "View Status Reports":

Fetch reports from farmers

Display reports

CASE "Logout":

Exit admin session

// Farmer Module

ELSE IF role is FARMER THEN

Display "Farmer Dashboard"

WHILE farmer is logged in DO

Display options:

1. View Profile

2. Manage Inventory

3. View Orders

4. View Schemes

5. Make Requests

6. Logout

SWITCH selected option:

CASE "View Profile":

Display farmer\_profile

CASE "Manage Inventory":

Display current\_inventory

Input to add or update products

Update product\_list in database

CASE "View Orders":

Display:

- New Orders

- Accepted Orders

- Pending Orders

- Completed Orders

Update status if required

CASE "View Schemes":

Fetch and display admin\_uploaded\_schemes

CASE "Make Requests":

Input request\_details

Submit to admin

CASE "Logout":

Exit farmer session

// User Module

ELSE IF role is USER THEN

Display "User Dashboard"

WHILE user is logged in DO

Display options:

1. Browse Products

2. View Profile

3. Place Order

4. View Order History

5. Logout

SWITCH selected option:

CASE "Browse Products":

Fetch available\_products from farmers

Display with filters (category, location, price)

CASE "View Profile":

Display user\_profile

CASE "Place Order":

Select products

Confirm order

Initiate UPI payment

IF payment is successful THEN

Place order and notify respective farmer

ELSE

Display "Payment Failed"

CASE "View Order History":

Display past\_orders

CASE "Logout":

Exit user session

// Notifications & Order Tracking

Send notifications to:

- Farmers on new orders and schemes

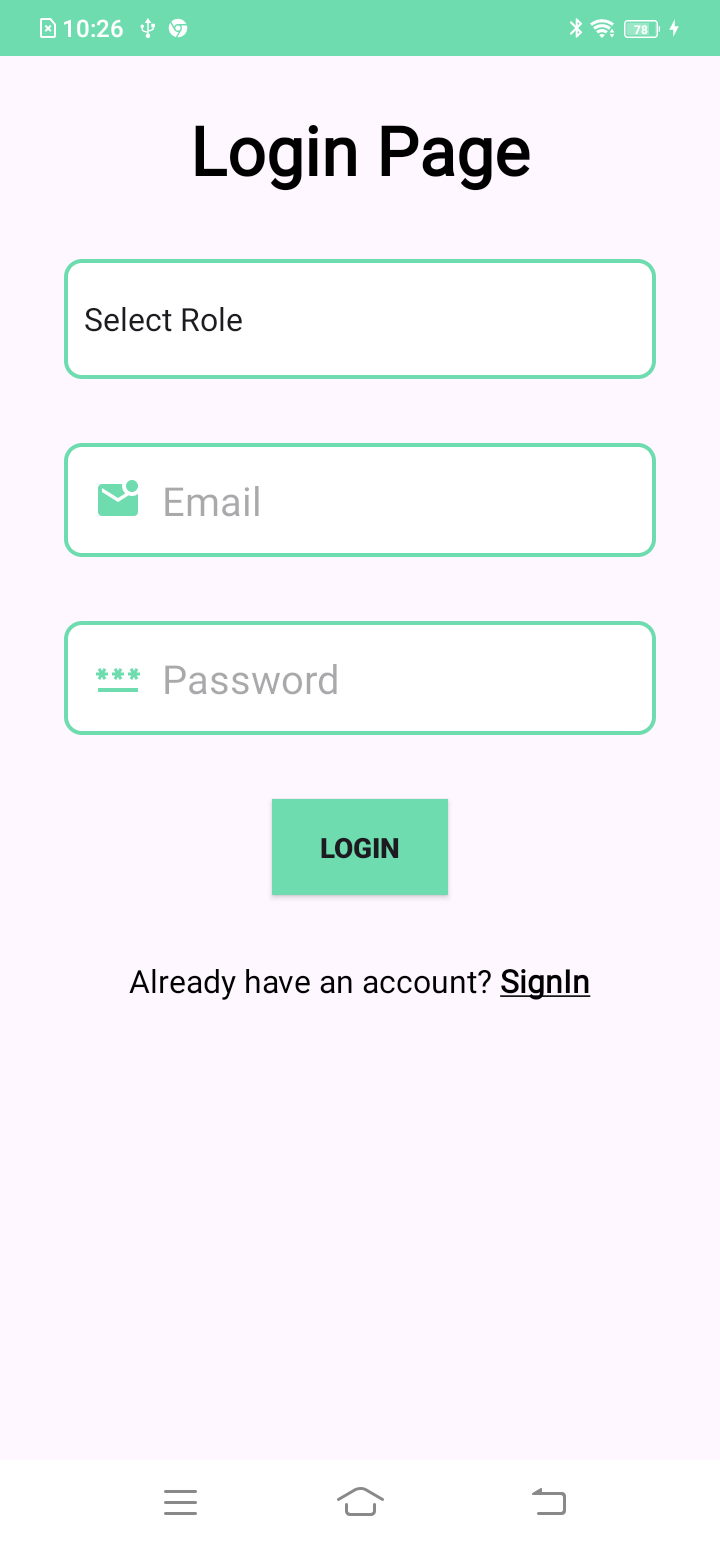
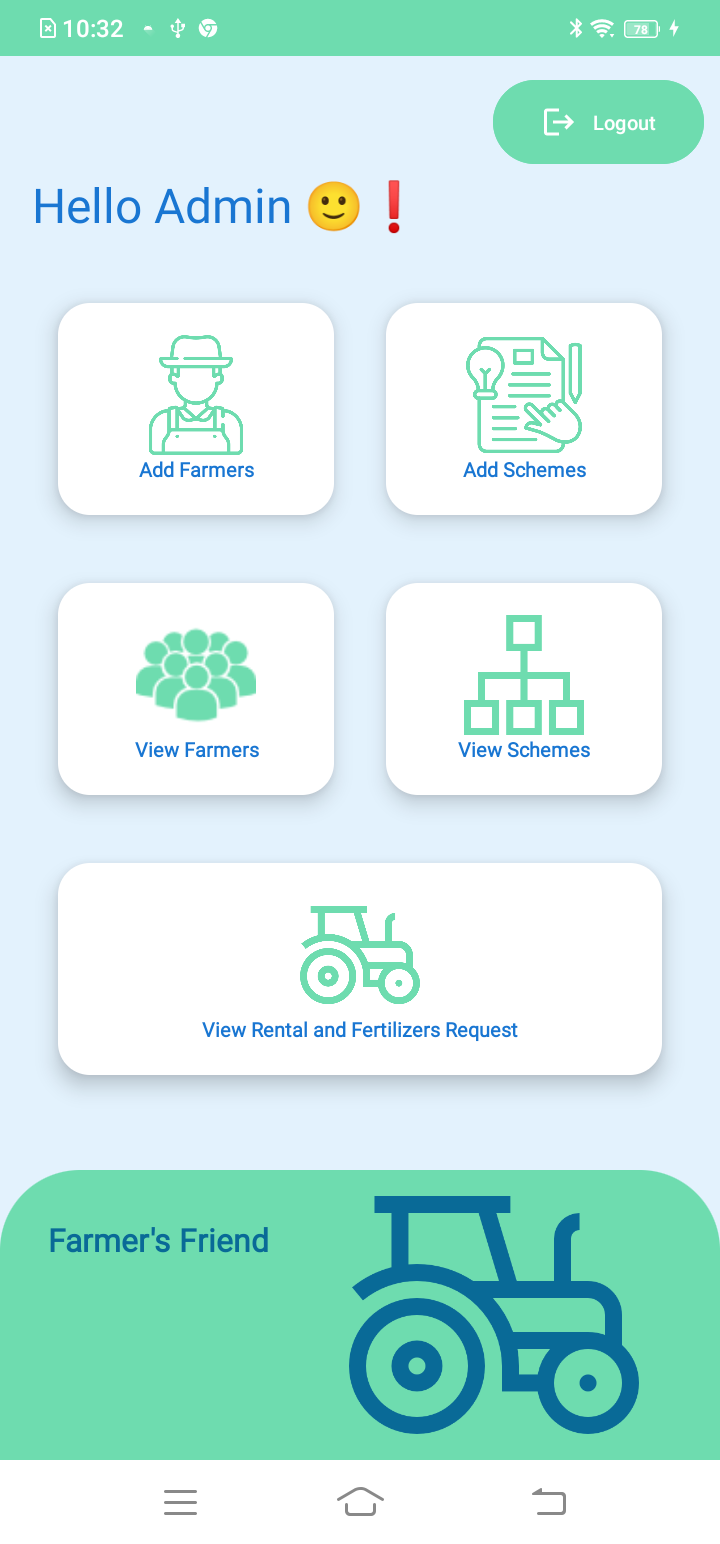
- Users on order status updates

Allow users to track placed orders

END

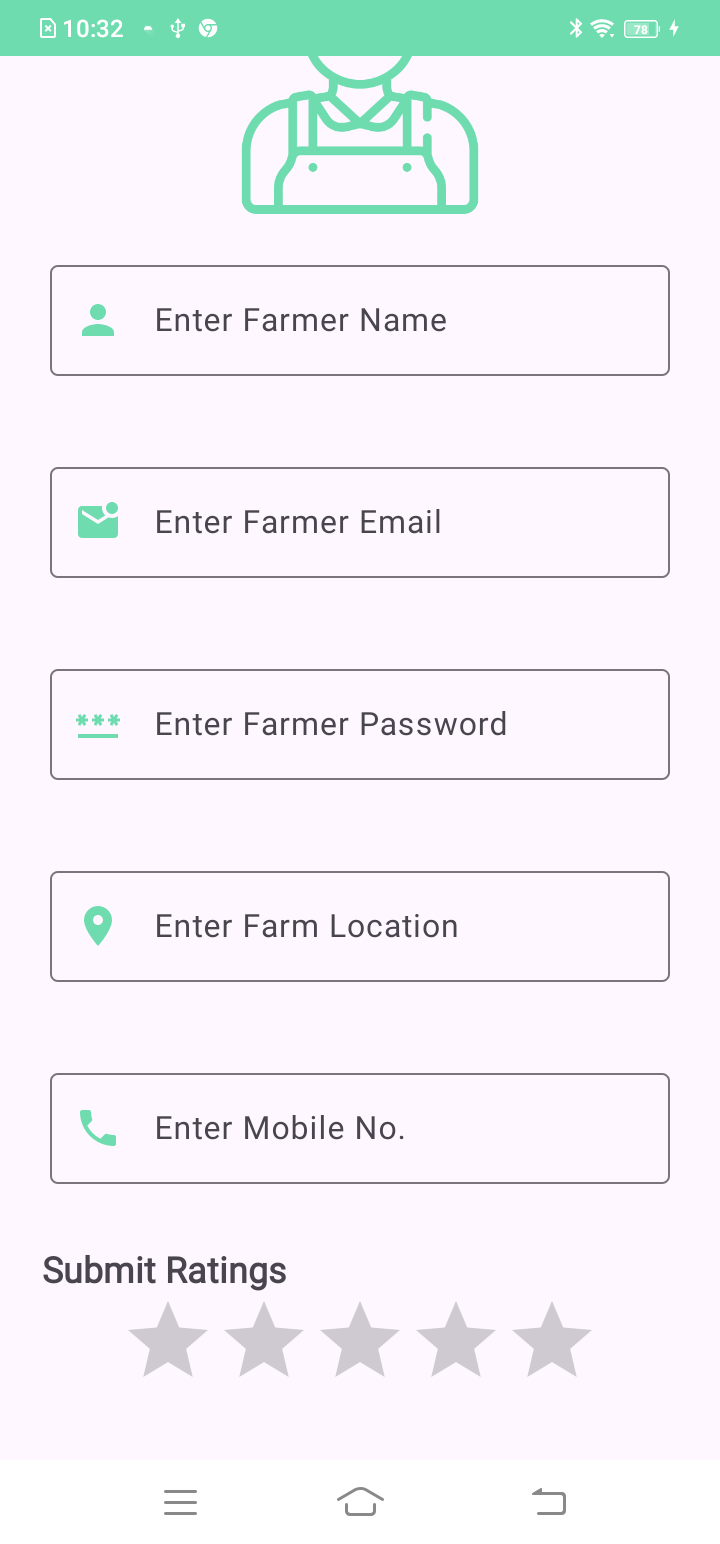
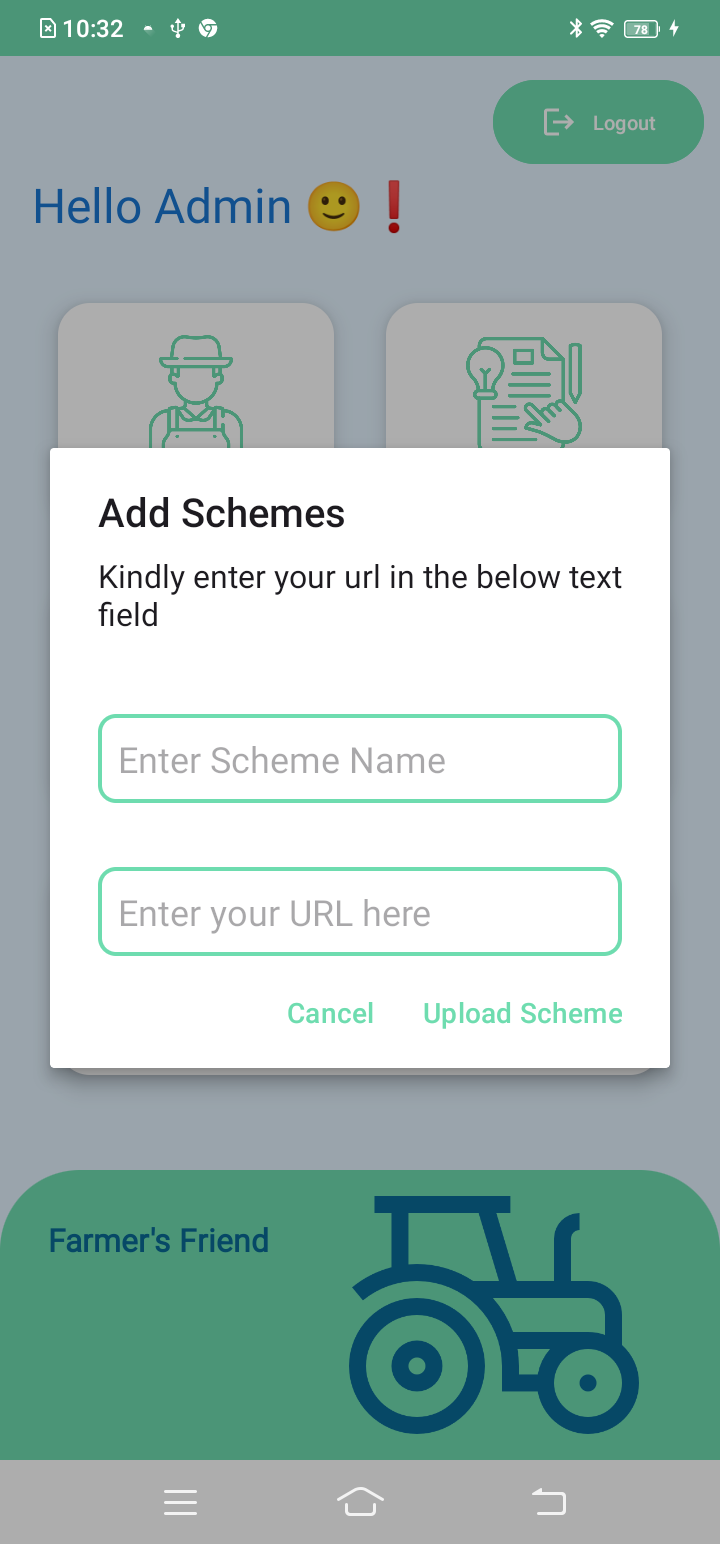
**APPENDIX-B**

**SCREENSHOTS**

****Fig B.1 Fig B.2

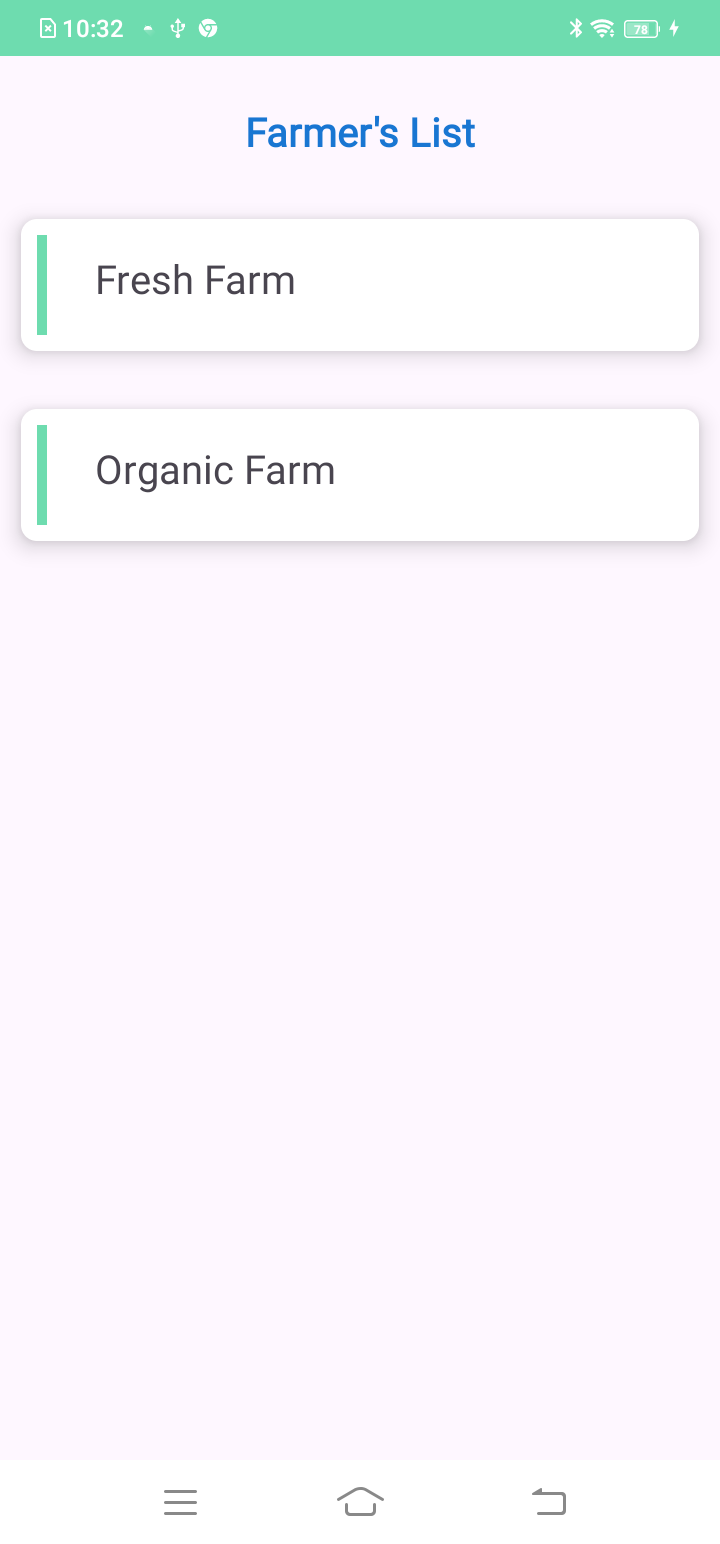
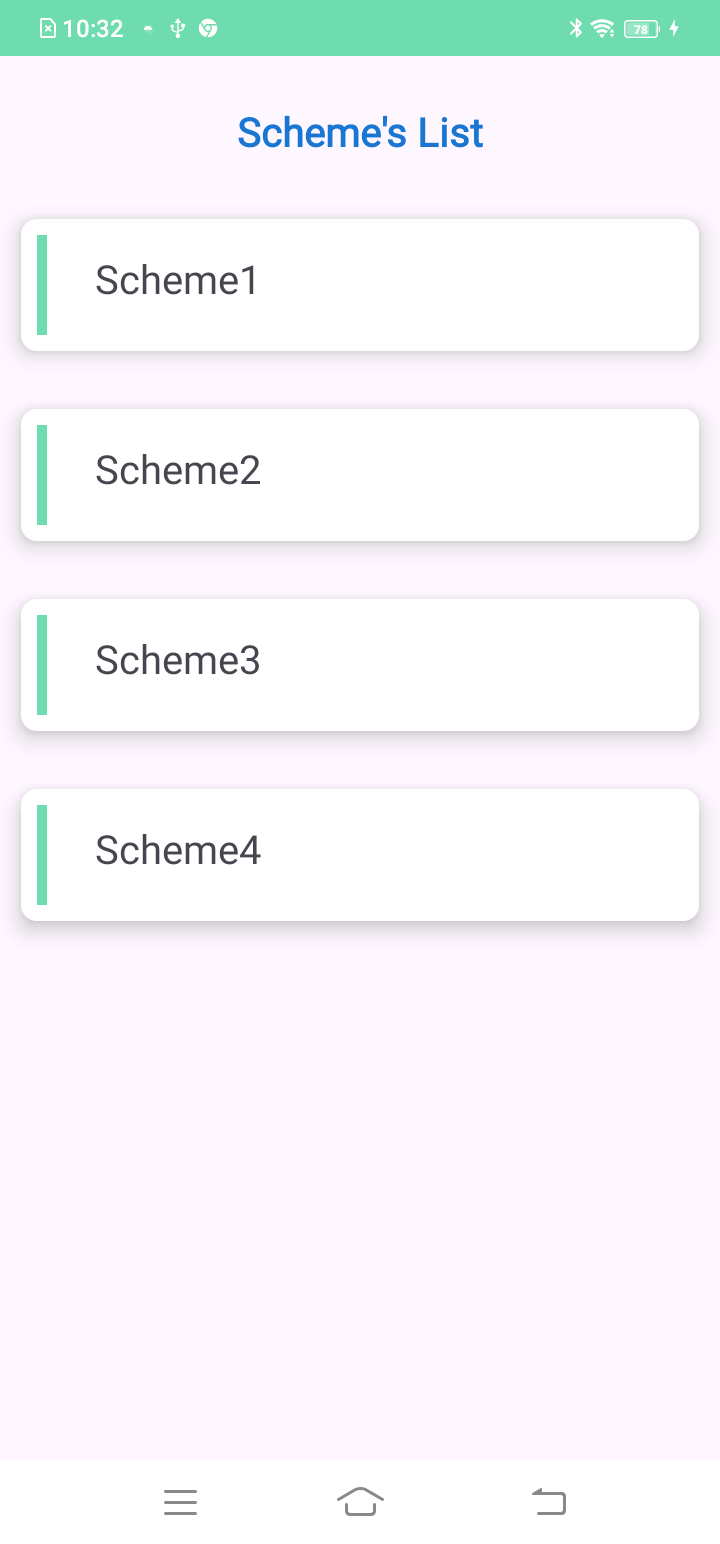
**Figure 2 Admin will navigate to his respective dashboard after successful login**

**Figure 1 User can login in the into the app using this page along with farmer and admin**

****Fig B.3 Fig B.4

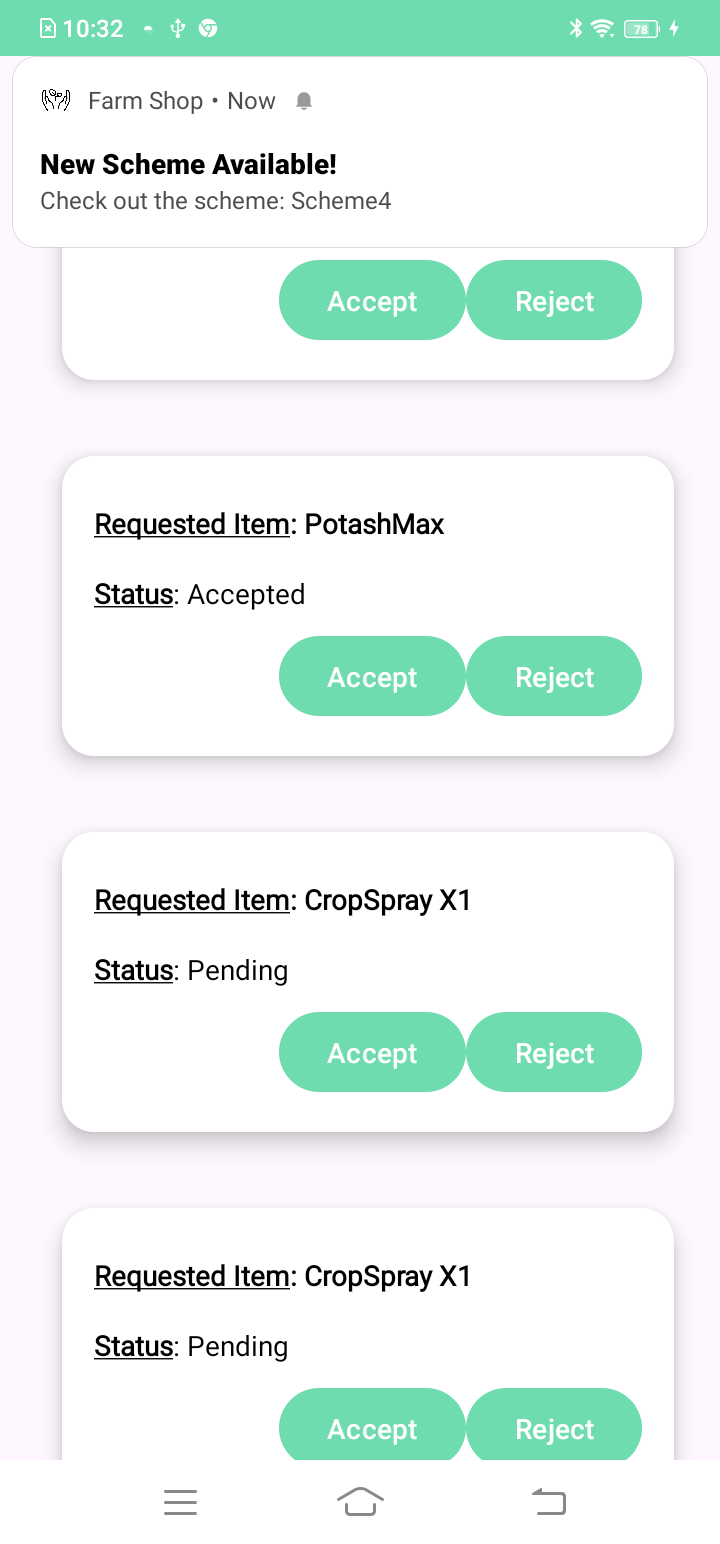
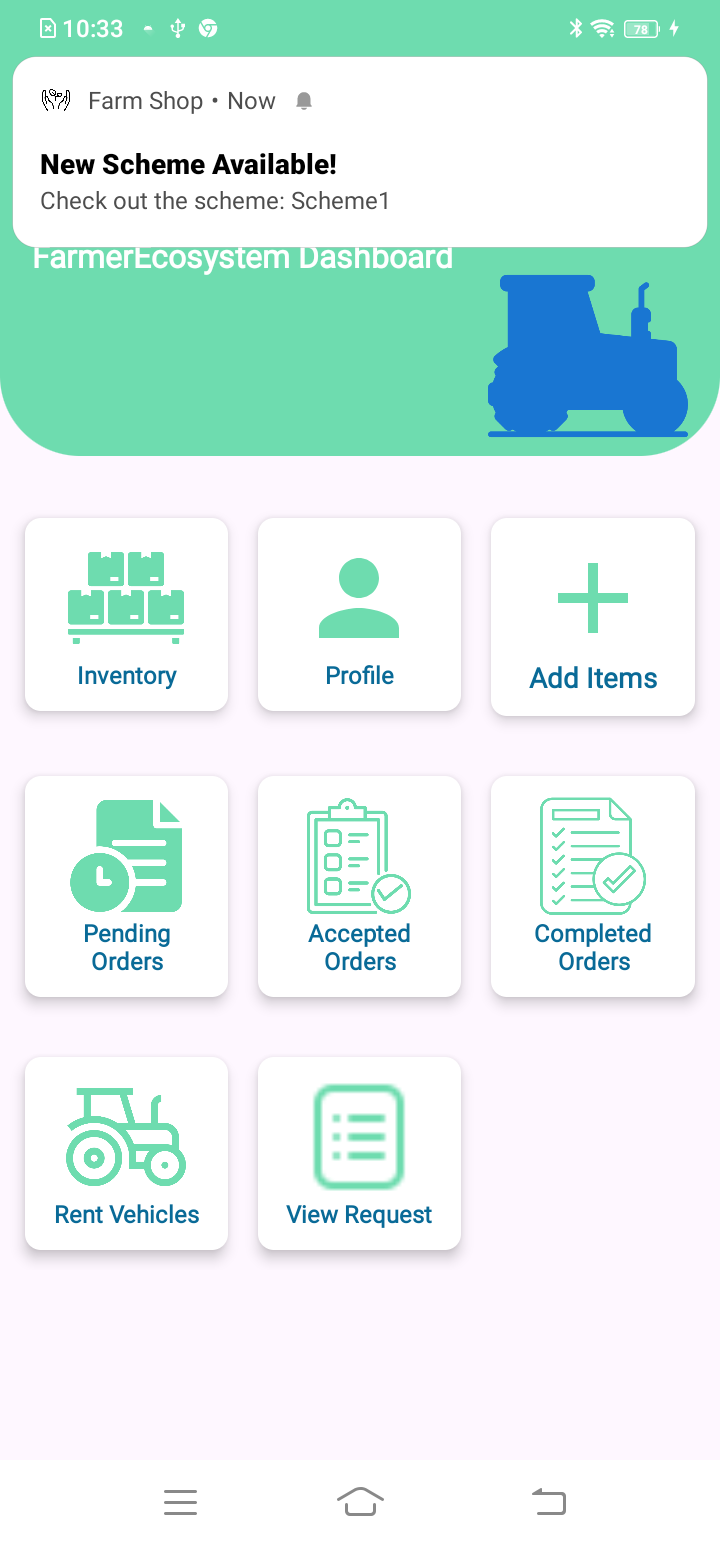
**Figure 4 Admin will be able to add Farmer schemes here**

**Figure 3 Admin can add the trusted farmers here**

****Fig B.5 Fig B.6

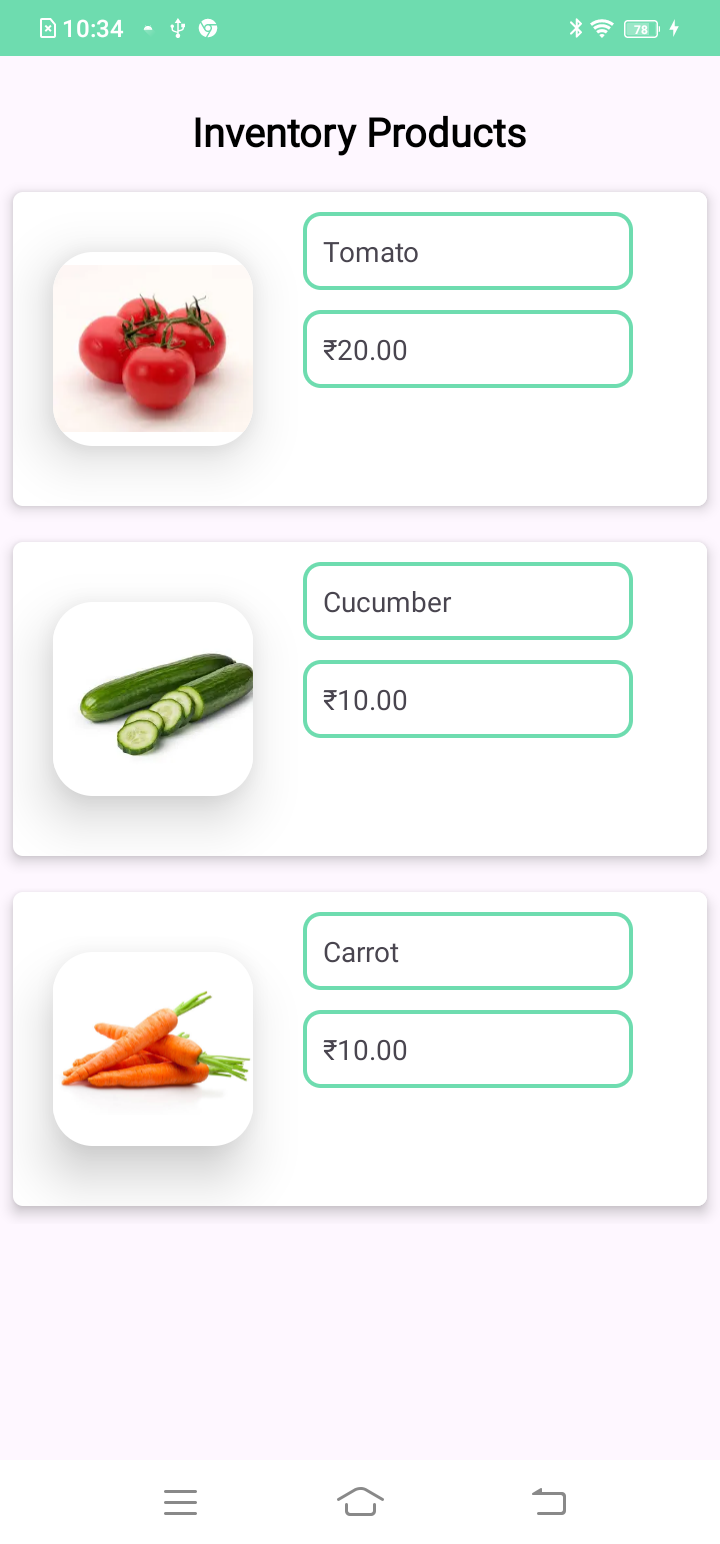
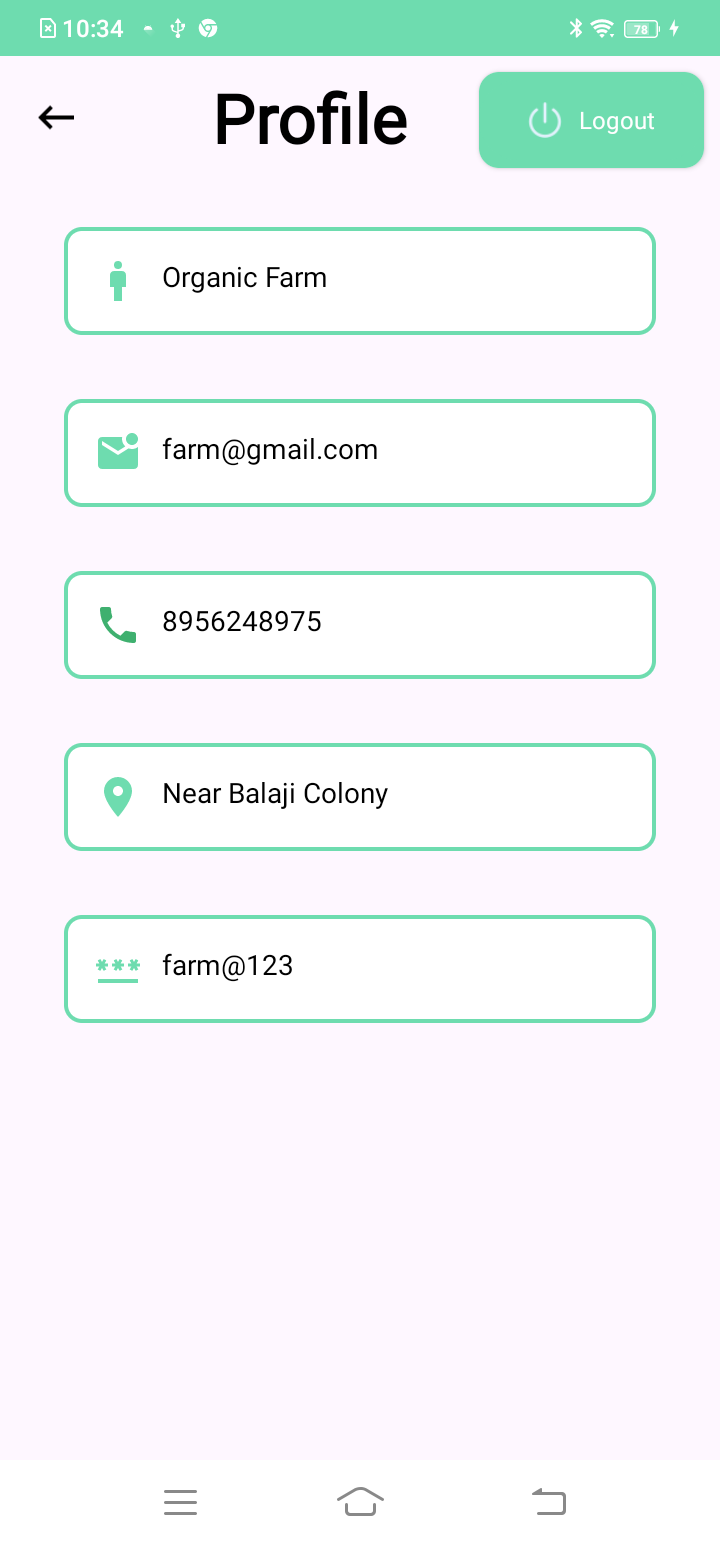
**Figure 6 Admin can view schemes here**

**Figure 5 Admin can view the added farmers here**

****Fig B.7 Fig B.8

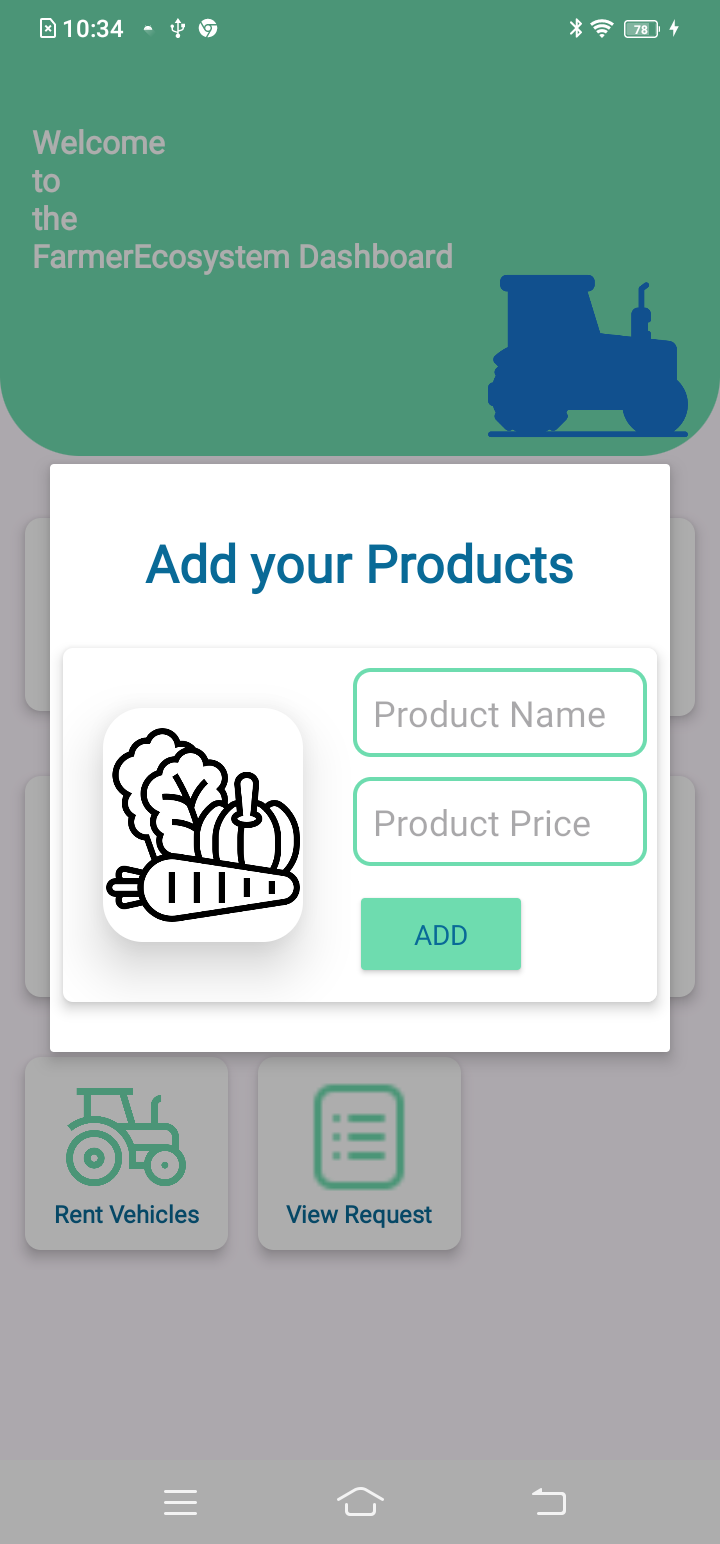
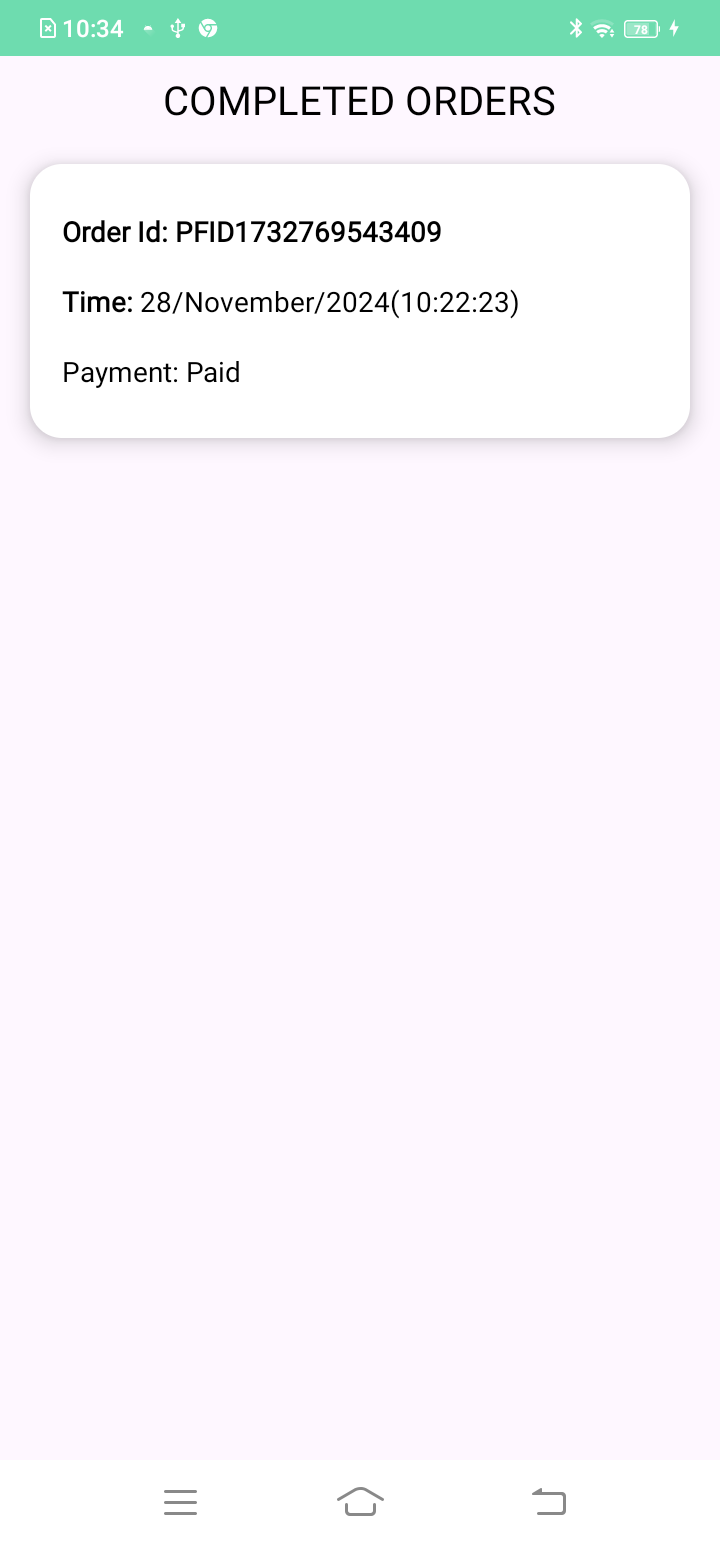
**Figure 8 Farmer will be navigating to this dashboard after successful login**

**Figure 7 Admin will update the status report from farmer’s here**

****Fig B.9 Fig B.10

**Figure 9 Farmers can view inventory products here**

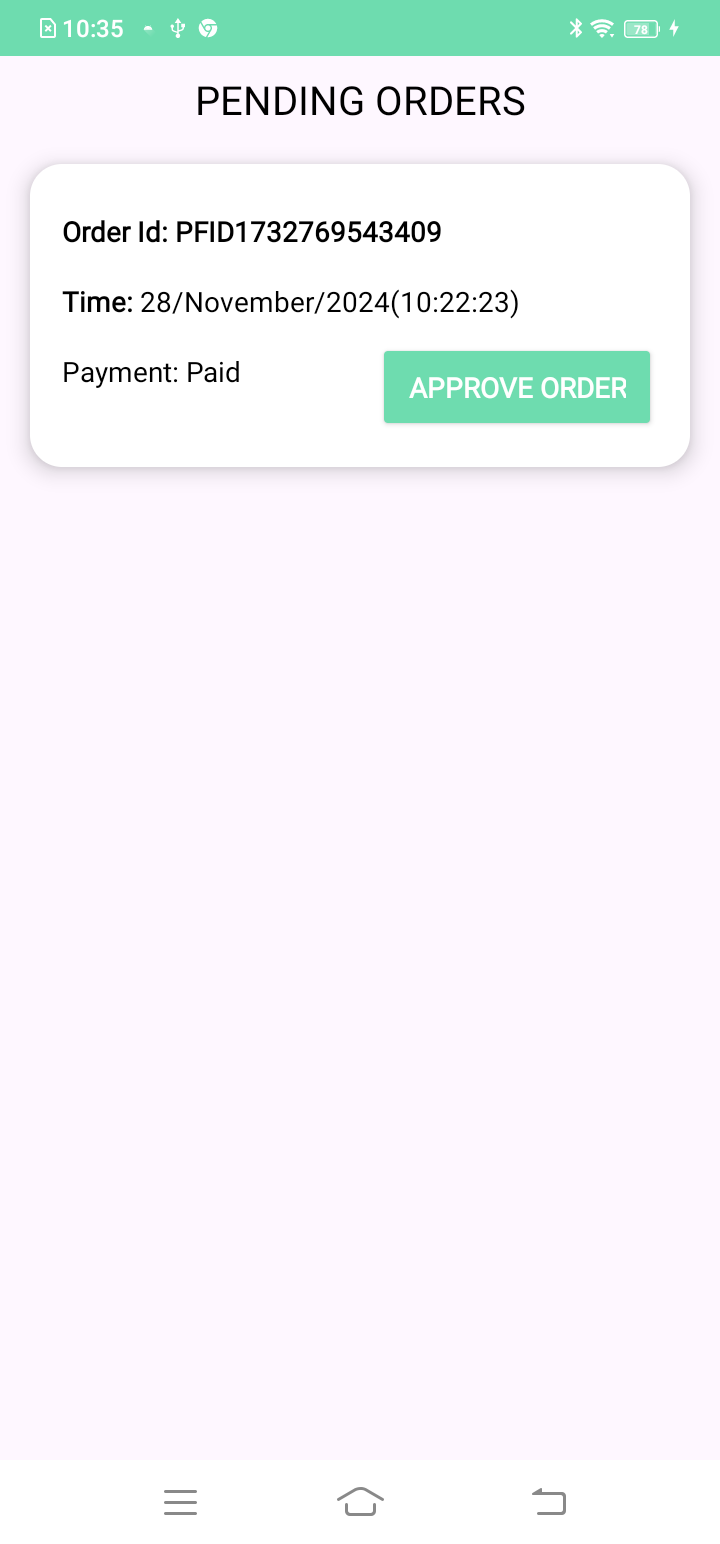
**Figure 10 Farmer can view his profile here**

Fig B.11 Fig B.1****2

**Figure 12 farmer will be able view completed orders here**

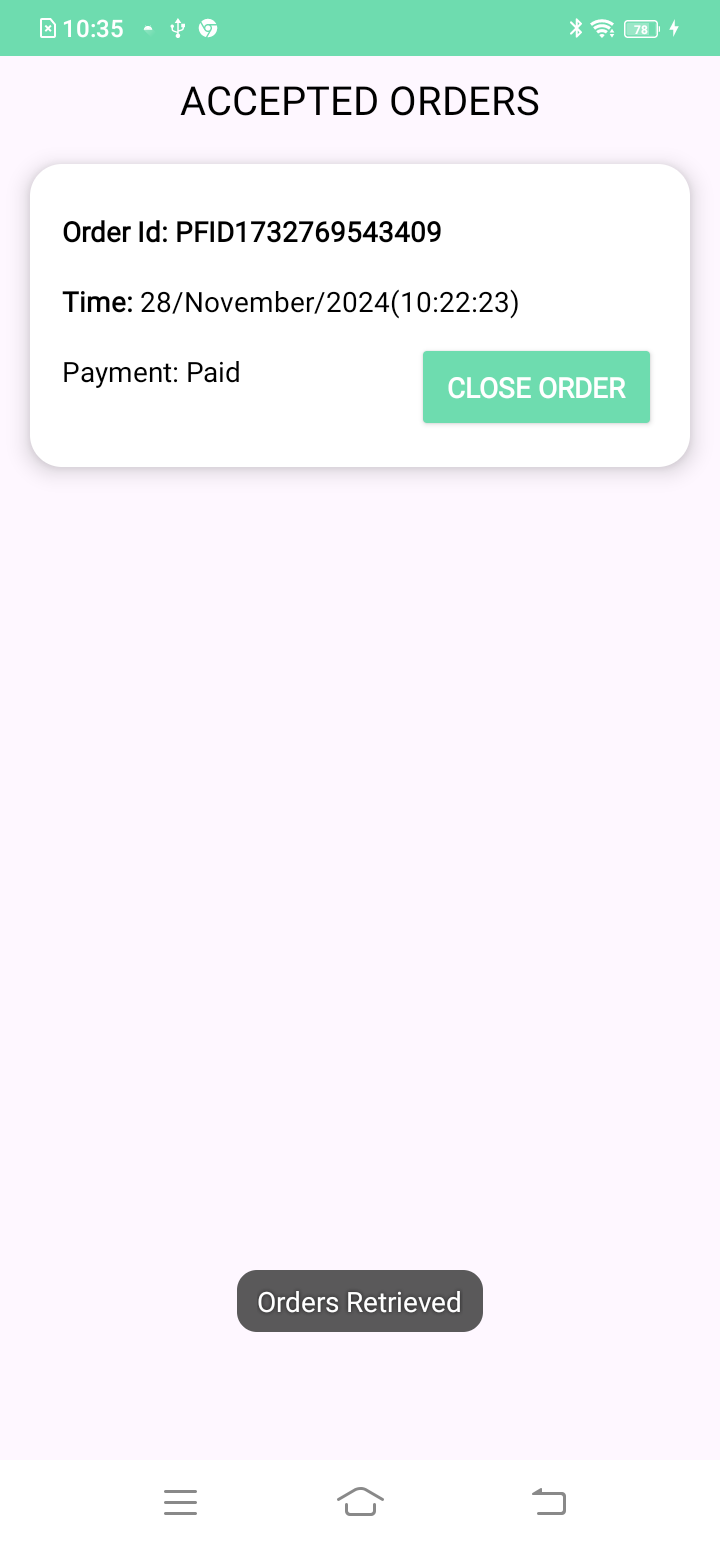
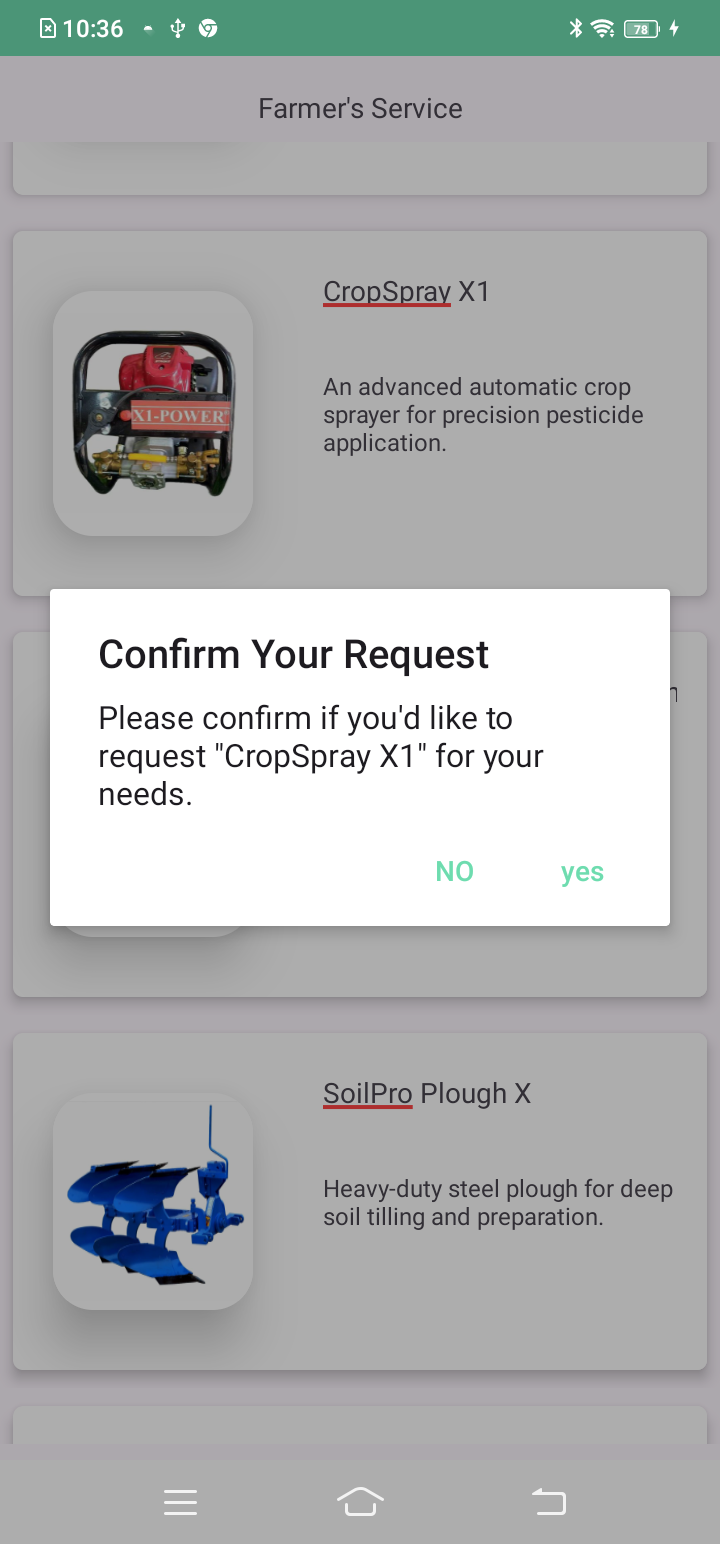
**Figure 11 Farmer can add his products here**

****

Fig B.13 Fig B.14 ****

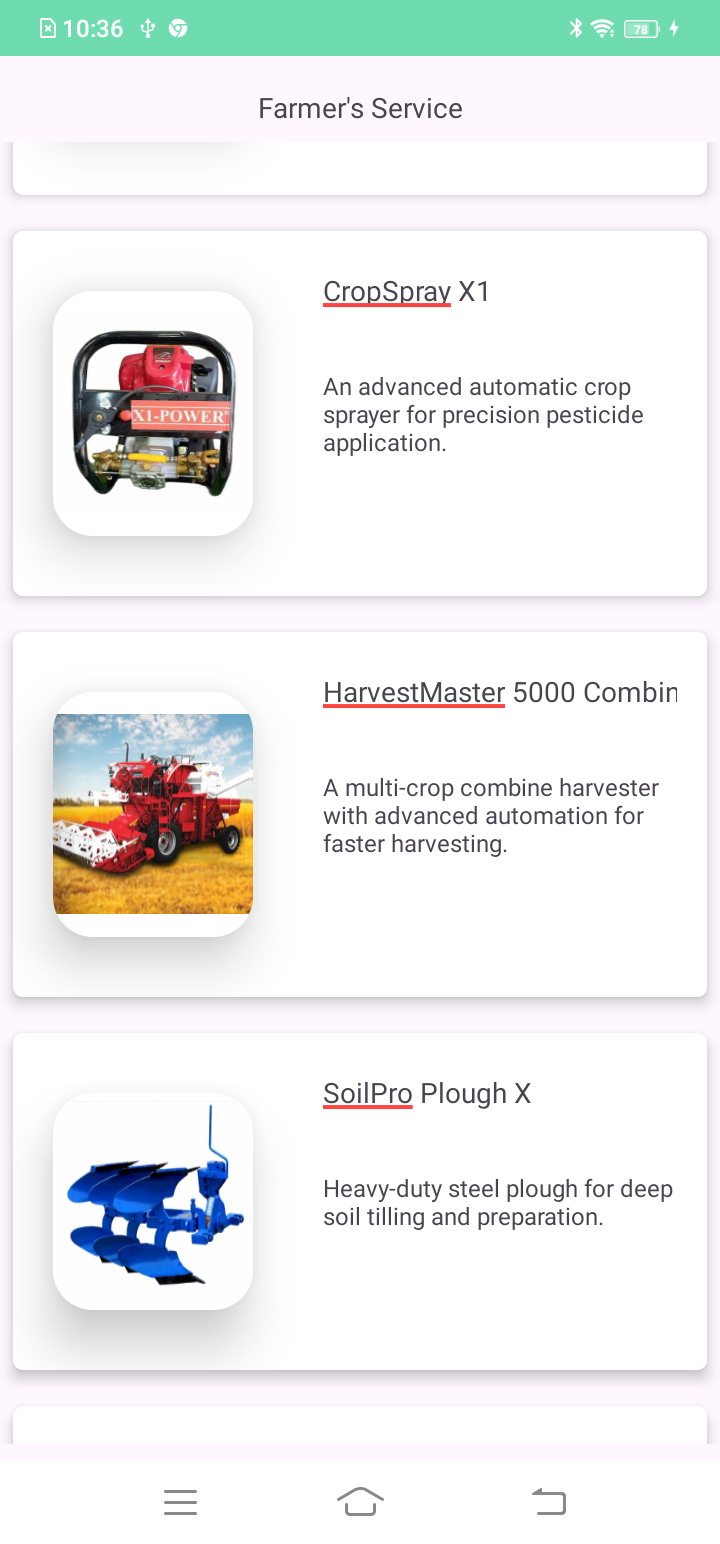
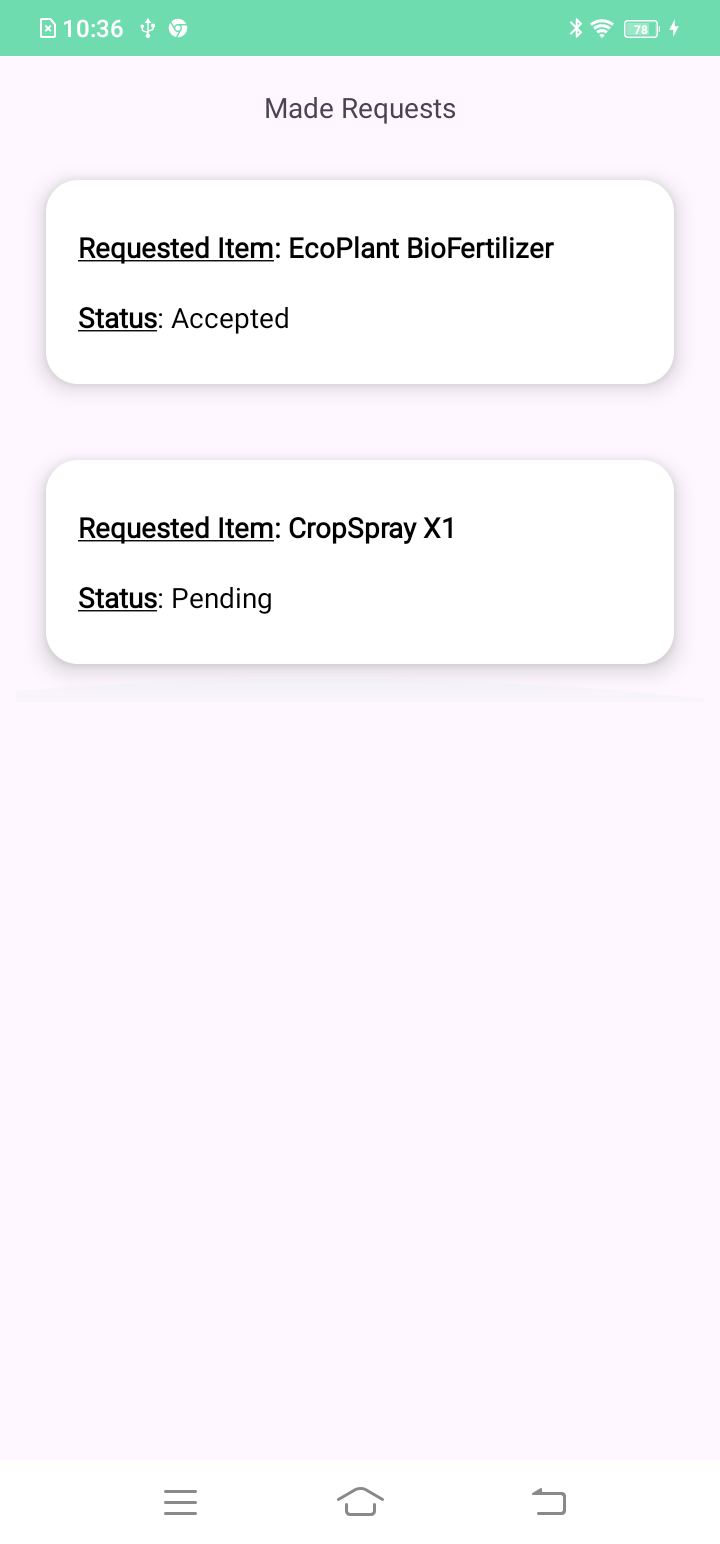
**Figure 14 Farmer will be able to view pending orders here and update them as well**

**Figure 13 Farmer will be to view user info here**

**** Fig B.15 Fig B.16

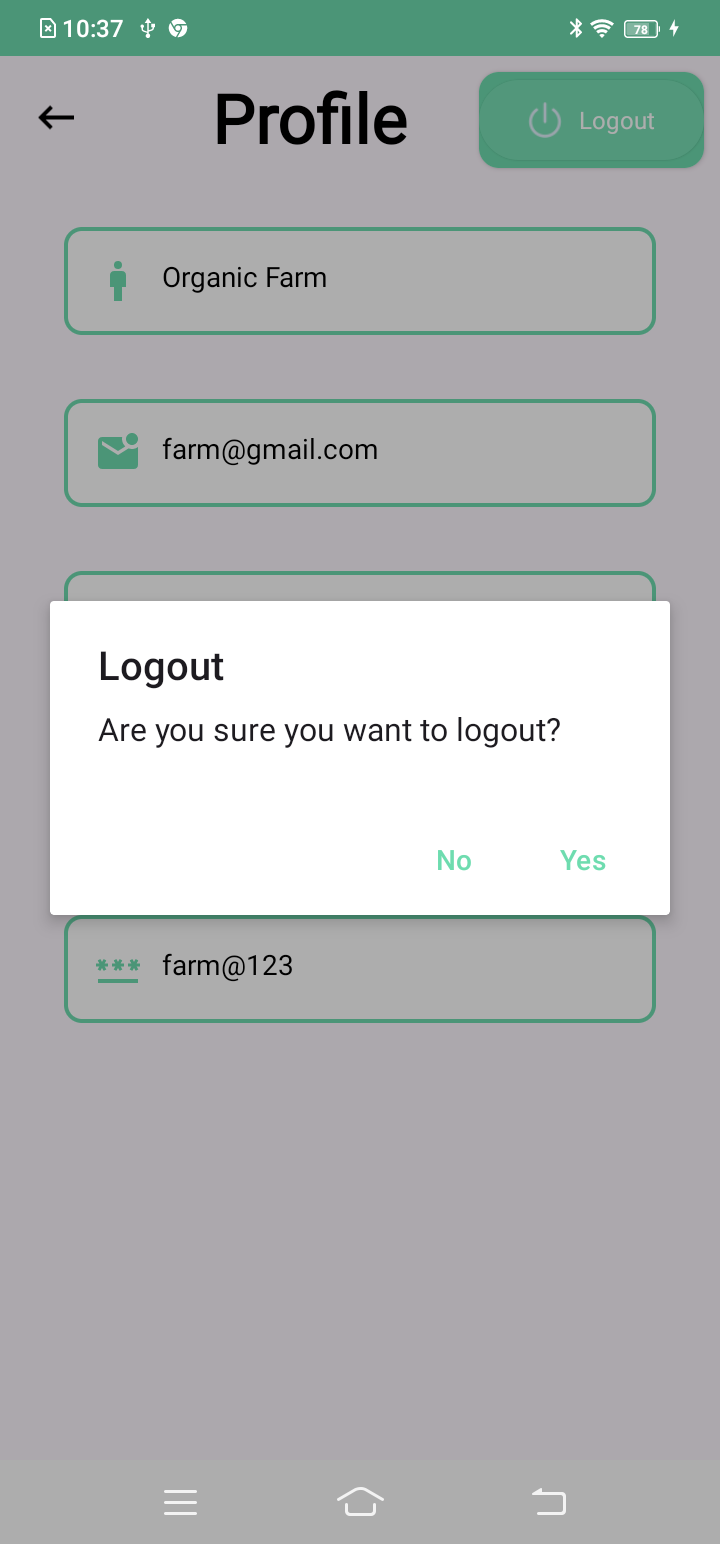
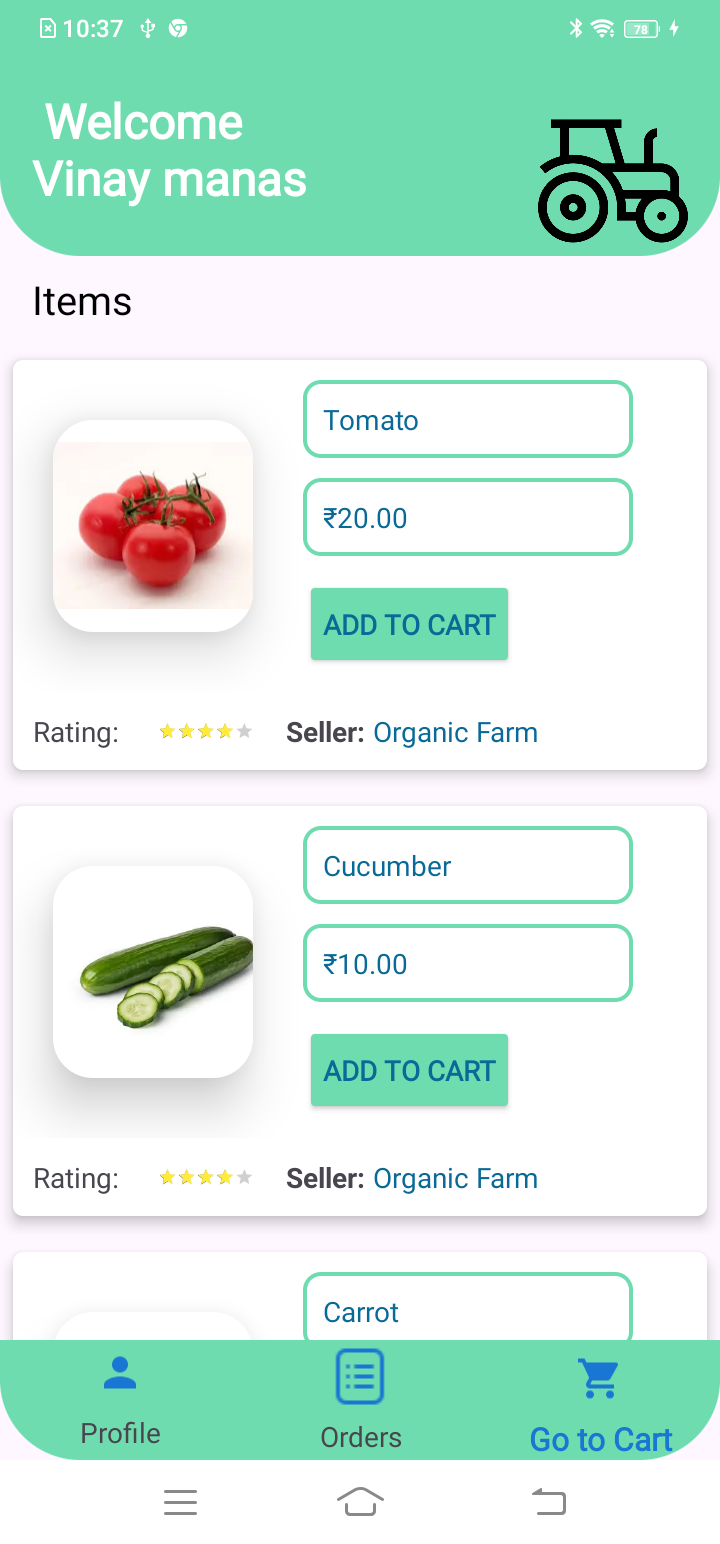
**Figure 16 Farmer can make request for the products to the admin**

**Figure 15 Farmer will be able to view accepted orders here and update them as well**

Fig B.17 Fig B.1****8

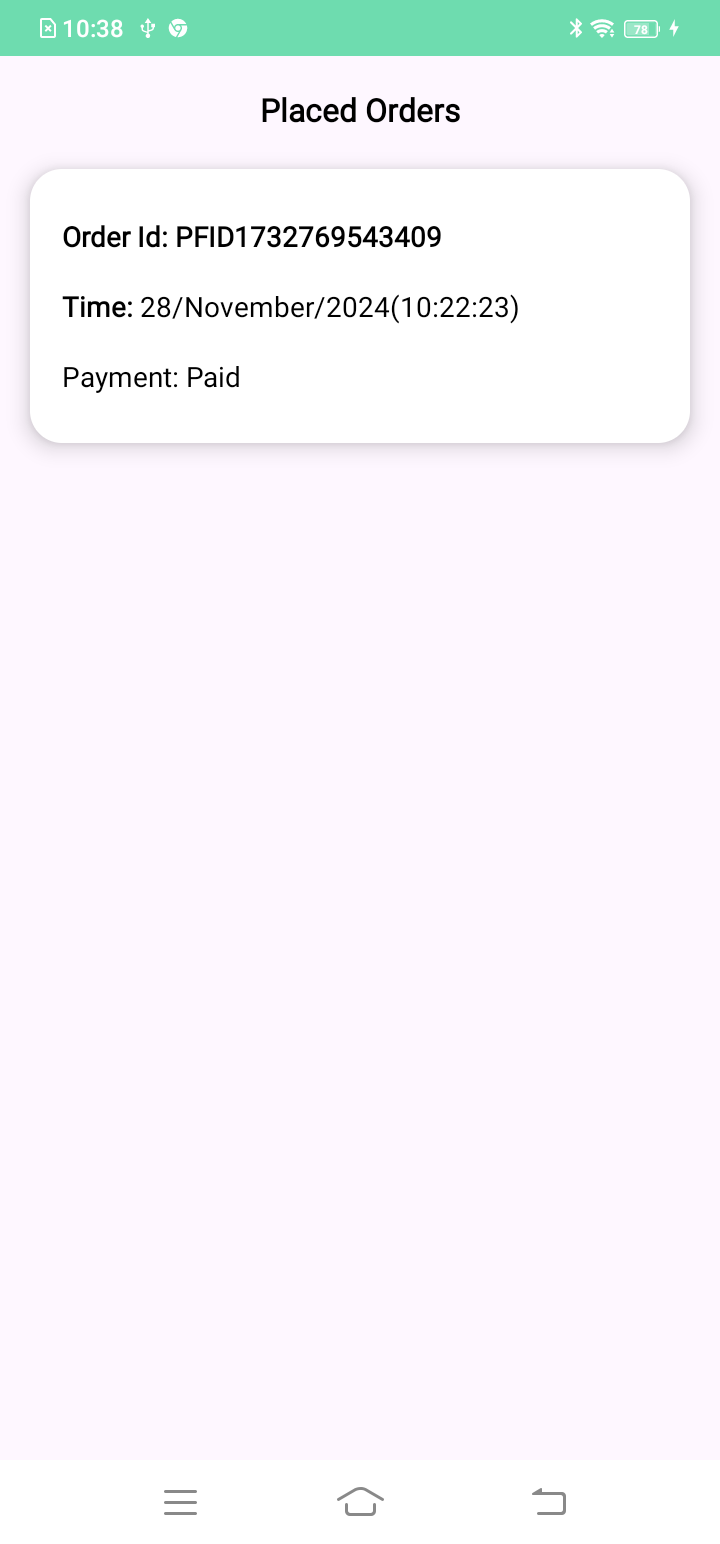
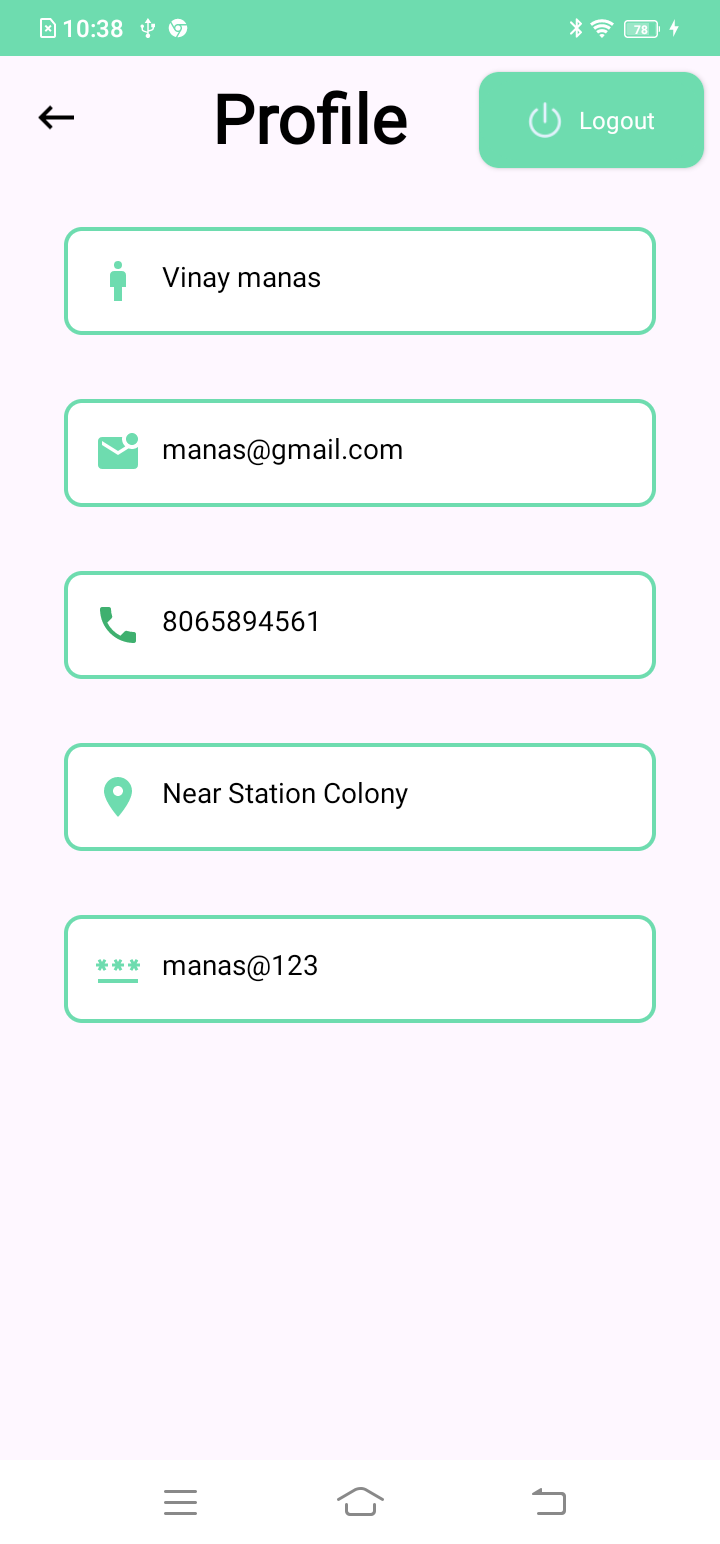
**Figure 18 Farmers can view the status of the made requests here**

**Figure 17 Farmer will be making request as per his needs**

****Fig B.19 Fig B.20

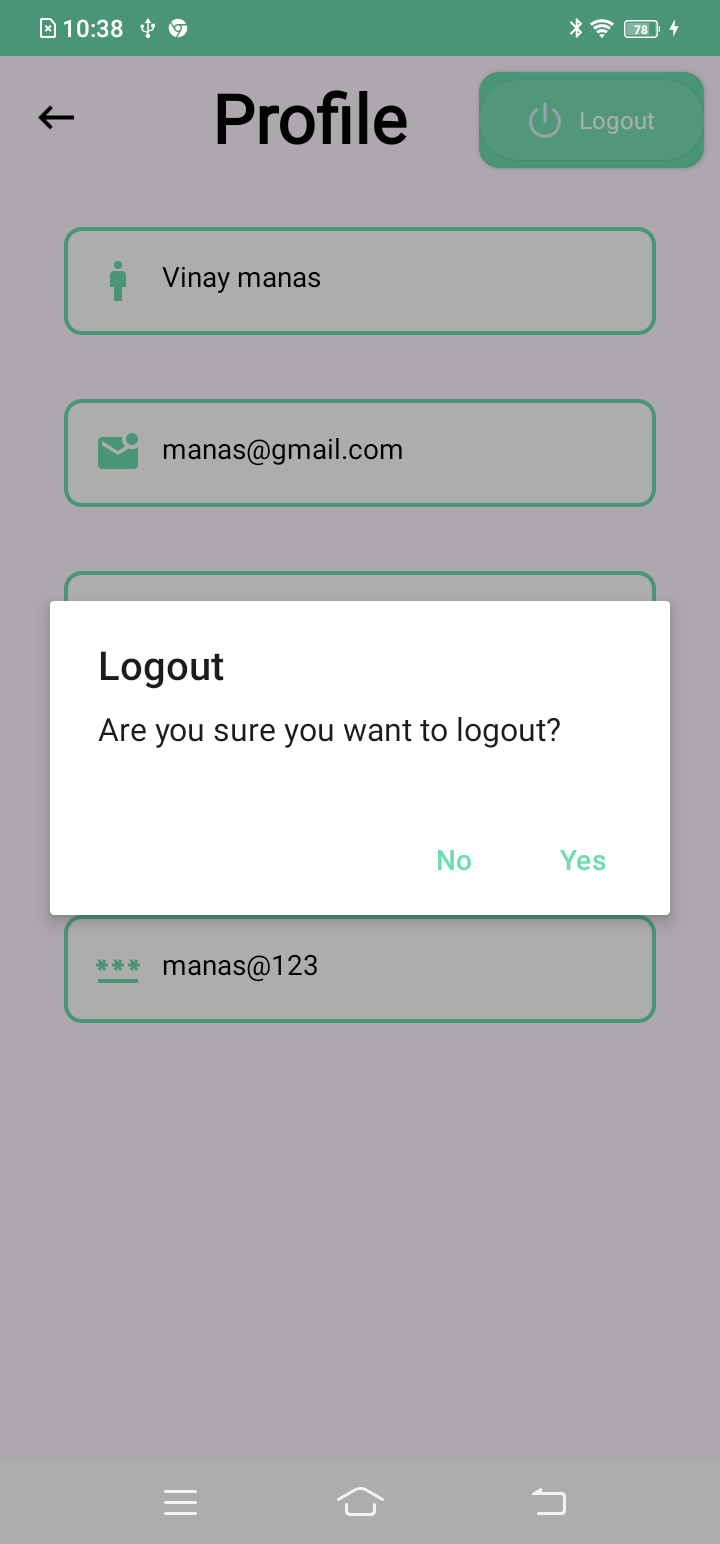
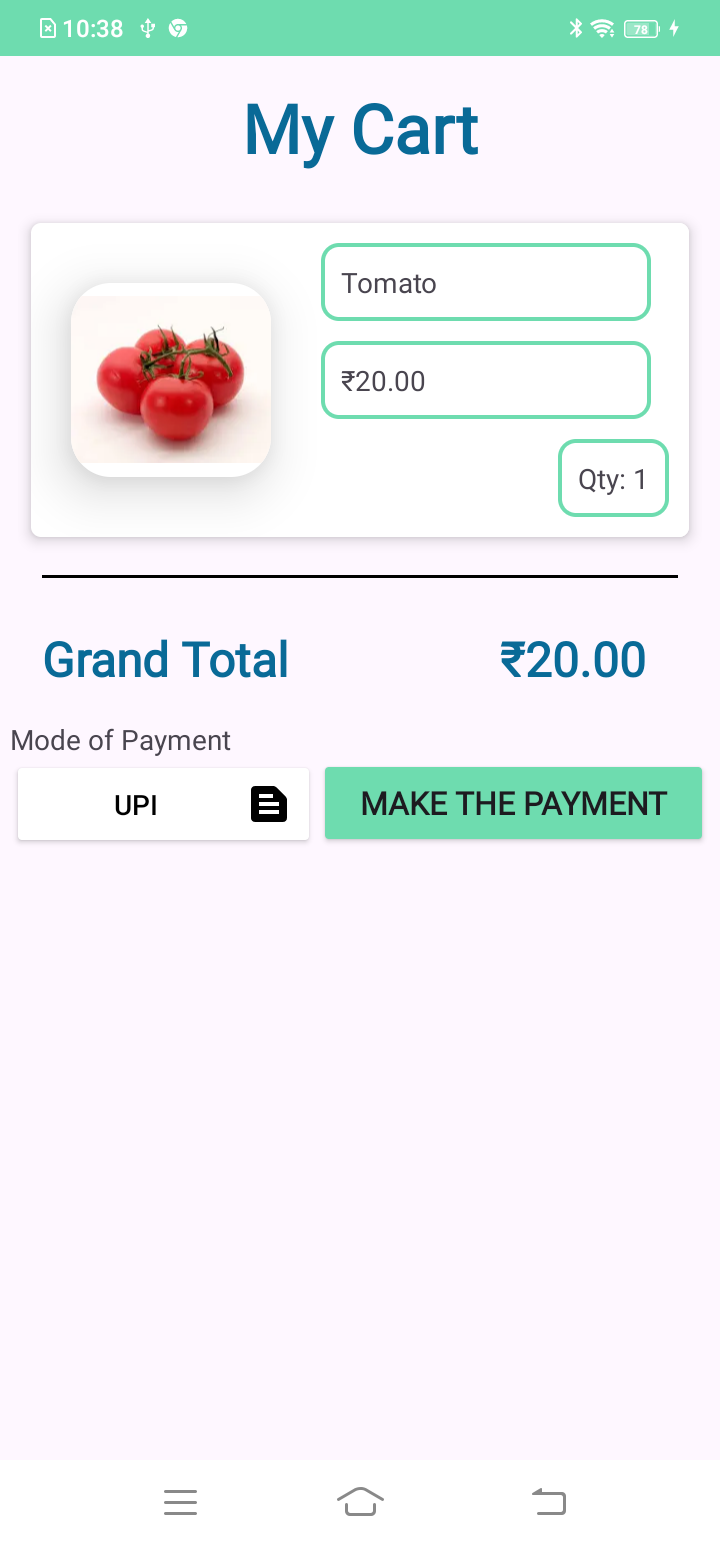
**Figure 20 User will be navigate to this page after successful login**

**Figure 19 Farmers will be logout using this dialog**

****Fig B.21 Fig B.22

**Figure 22 User will be able to see his profile here**

**Figure 21 User will be able to see place orders here**

****Fig B.23 Fig B.24

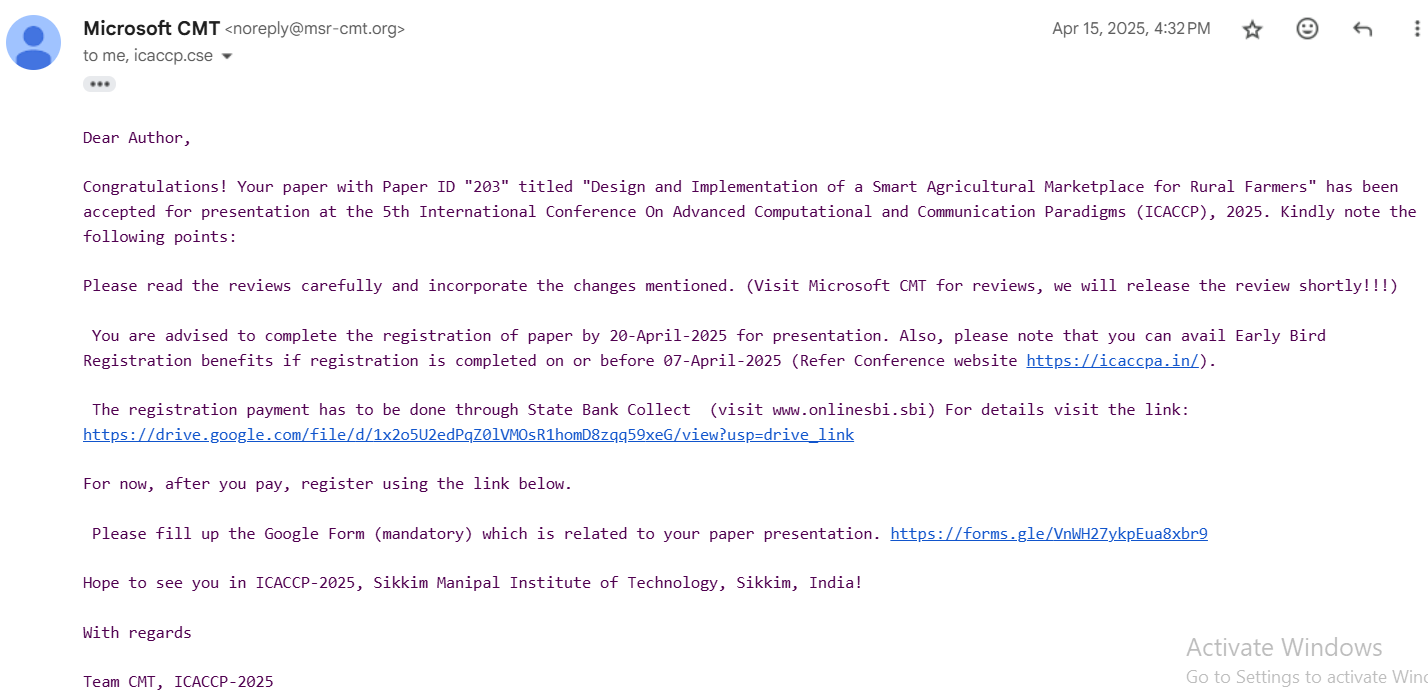
**Figure 24 User can place the order here**

**Figure 23 User will be logout here**

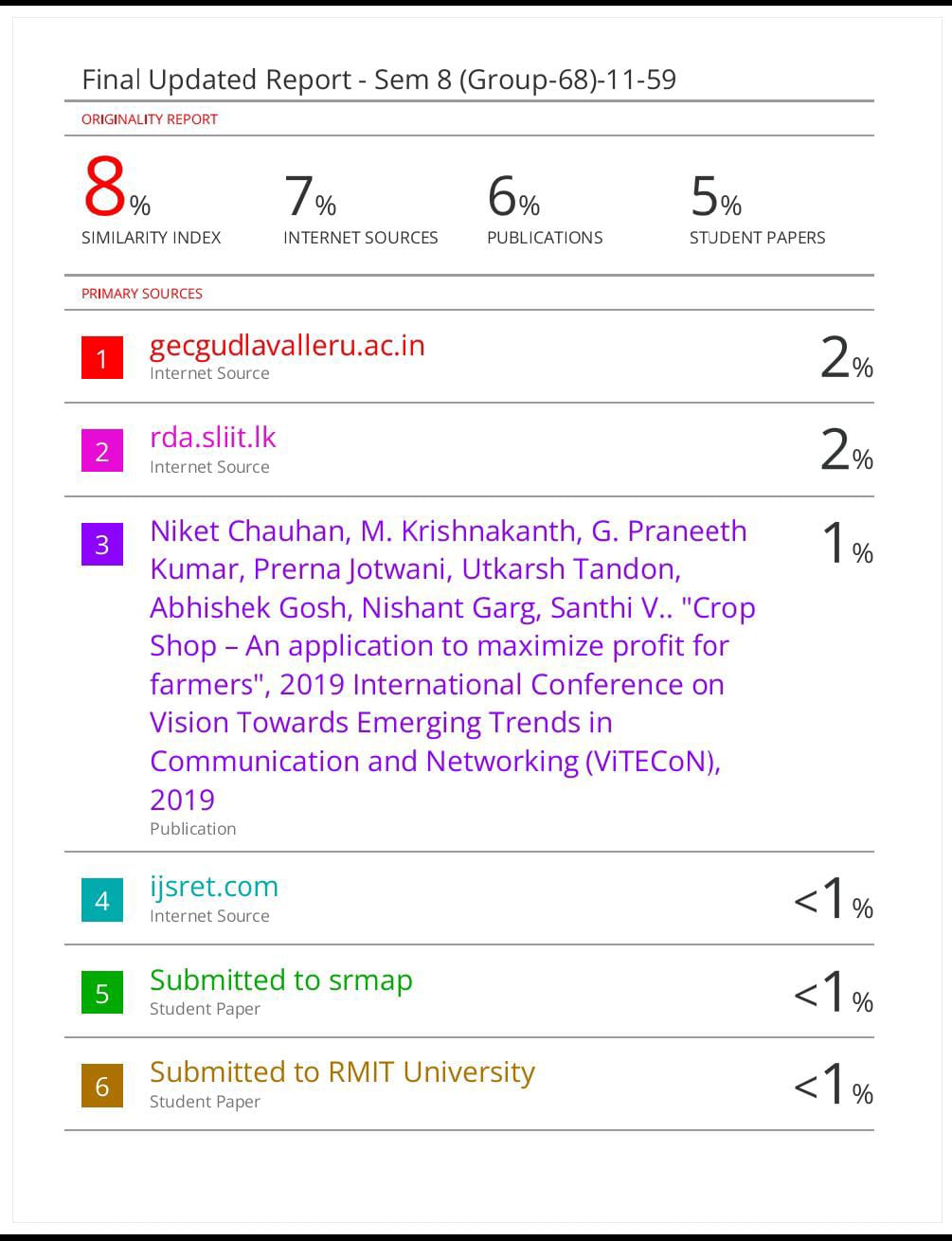
**APPENDIX-C**

**ENCLOSURES**

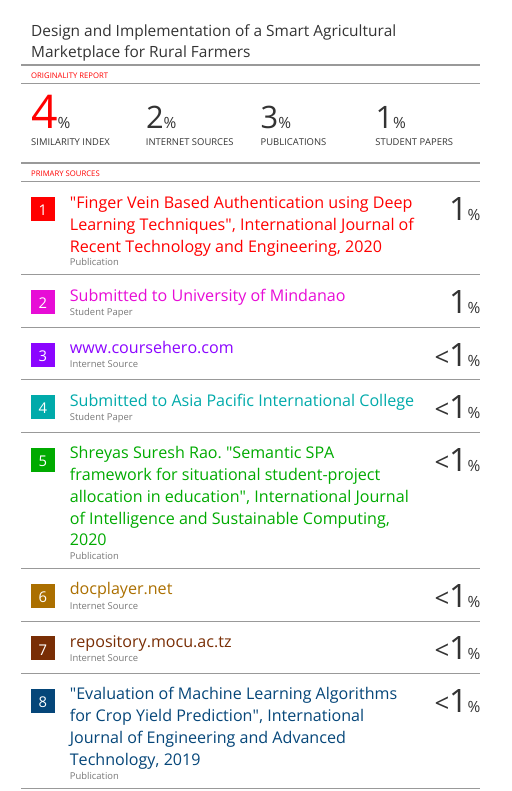
* 1. **CONFERENCE PAPER PRESENTED CERTIFICATES**
     + **Screenshot of acceptance message**

****

* 1. **PLAGIARISM REPORT OF THE REPORT**

****

**3.PLAGIARISM REPORT OF RESEARCH PAPER**

****

1. **SUSTAINABLE DEVELOPMENT GOALS**



The "Uplifting the Farmer through a Connected Ecosystem" project contributes significantly to multiple Sustainable Development Goals (SDGs) by utilizing technology to transform traditional agricultural commerce. This mobile-based platform empowers farmers by enabling direct consumer access, facilitating real-time transactions, promoting awareness of government schemes, and supporting future integrations like land renting and fertilizer management. Through this, the project fosters sustainable development by increasing economic opportunities, enhancing agricultural efficiency, and supporting environmentally sound practices in the rural economy.

**The Project Work Carried out here is mapped to the following goals:**

**1. Goal 1: No Poverty**

* The platform increases income opportunities by removing intermediaries and allowing farmers to sell directly to consumers.
* Verified farmer onboarding and access to government schemes help ensure financial inclusion for rural communities.
* By improving market reach and enabling faster transactions, it supports income stability for small-scale farmers.

**2. Goal 2: Zero Hunger**

* Facilitating better crop distribution through digital channels enhances food accessibility and reduces wastage.
* Integration of fertilizer management and future crop advisory services promotes better yield and sustainable agriculture.
* By connecting producers directly with consumers, it ensures fresher, more affordable food for buyers and better returns for farmers.

**3. Goal 8: Decent Work and Economic Growth**

* Supports inclusive rural development by empowering farmers with modern digital tools.
* Reduces dependency on exploitative middlemen, increasing profit margins and boosting economic participation.
* Encourages agricultural entrepreneurship, especially among youth and small landholders.

**4. Goal 9: Industry, Innovation, and Infrastructure**

* Leverages Android technology, Kotlin, and UPI-based systems to modernize the agri-market ecosystem.
* Builds a robust digital platform that serves as rural infrastructure for commerce, logistics, and scheme delivery.
* Demonstrates scalable innovation by integrating mobile technology in underserved regions.

**5. Goal 10: Reduced Inequalities**

* Empowers marginalized farmers by providing equal access to digital markets and information.
* Scheme management ensures that even remote users are informed of and can benefit from government assistance.
* Promotes equitable trade practices by standardizing pricing and improving transparency.

**6. Goal 12: Responsible Consumption and Production**

* Encourages local sourcing by enabling consumers to buy directly from nearby farmers, reducing food miles.
* Real-time inventory and order tracking minimize overproduction and wastage.
* Supports efficient agricultural logistics through planned future modules like vehicle renting.

**In Conclusion**

The "Uplifting the Farmer through a Connected Ecosystem" project directly addresses critical UN SDGs by fostering economic resilience, food security, and technological innovation in agriculture. It supports **Goal 1 (No Poverty)** and **Goal 2 (Zero Hunger)** by increasing farmer incomes and ensuring better food access. By building a digital marketplace, the project aligns with **Goal 8 (Decent Work and Economic Growth)** and **Goal 9 (Industry, Innovation, and Infrastructure)**, modernizing a traditionally under-resourced sector. Moreover, its inclusive design and equal opportunity access promote **Goal 10 (Reduced Inequalities)**, while its emphasis on localized and efficient agricultural transactions contributes to **Goal 12 (Responsible Consumption and Production)**. This initiative showcases how tech-driven platforms can transform rural economies and promote sustainable development holistically.