MOBILE APP FOR DIRECT MARKET ACCESS FOR FARMERS

A PROJECT REPORT

Submitted by,

Ms. LAKKI REDDY VARSHITHA - 20211CIT0049

Mr. ERANTI SAI KISHAN - 20211CIT0052

Mr. ERANTI SAI DINESH - 20211CIT0053

Ms. PAVANI M - 20211CIT0067

Under the guidance of,

Ms. BHAVYA.B

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING (INTERNET OF THINGS)

At

PRESIDENCY UNIVERSITY

BENGALURU

MAY 2025

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE ENGINEERING

CERTIFICATE

This is to certify that the Project report **MOBILE APP FOR DIRECT MARKET ACCESS FOR FARMERS** being submitted by LAKKI REDDY VARSHITHA, ERANTI SAI KISHAN, ERANTI SAI DINESH, PAVANI M bearing roll number(s) 20211CIT0049, 20211CIT0052, 20211CIT0053, 20211CIT0067 in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology in Computer Science and Engineering(Internet Of Things) is a bonafide work carried out under my supervision.

Ms. Bhavya.B
ASSISTANT PROFESSOR
School of CSE&IS
Presidency University

Dr. Shanthi Pichandi Anandaraj
PROFESSOR & HoD
School of CSE&IS
Presidency University

Dr. MYDHILI NAIR
Associate Dean
School of CSE
Presidency University

Dr. SAMEERUDDIN KHAN
Pro-Vc School of Engineering
Dean -School of CSE&IS
Presidency University

ii

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE ENGINEERING

DECLARATION

We hereby declare that the work, which is being presented in the project report entitled **MOBILE APP FOR DIRECT MARKET ACCESS FOR FARMERS** in partial fulfillment for the award of Degree of **Bachelor of Technology** in **Computer Science and Engineering (Internet Of Things)**, is a record of our own investigations carried under the guidance of **BHAVYA.B, ASSISTANT PROFESSOR, School of Computer Science Engineering & Information Science, Presidency University, Bengaluru.**

We have not submitted the matter presented in this report anywhere for the award of any other Degree.

NAME	ROLL NO	SIGNATURE
LAKKI REDDY VARSHITHA	20211CIT0049	
ERANTI SAI KISHAN	20211CIT0052	
ERANTI SAI DINESH	20211CIT0053	
PAVANI M	20211CIT0067	

ABSTRACT

Agriculture is a source of income for most economies, but farmers themselves struggle when it comes to selling their crops. From having to go through various middlemen to receiving unfair prices, it's a challenging journey. In response to this, the "Mobile App for Direct Market Access for Farmers" was developed—a intelligent, user-friendly solution developed with Android Studio that allows farmers to directly link with buyers, retailers, and consumers.

The objective of the app is straightforward: eliminate middlemen who are not needed, enhance transparency, and make farmers get what they truly deserve for their efforts. With a minimalistic, non-cluttered interface, the app is very easy to use even for people who are not very fond of technology. The app is multilingual and comes with smooth navigation, making it simple for a larger number of farmers to use.

Some of the key features include real-time updates on market prices, a direct chat platform between buyers and sellers, and order management, inventory, and delivery management tools. The app also facilitates secure digital payments, making transactions smoother and more reliable. In addition to that, farmers can browse useful financial tools—such as details on loans, government subsidies, and savings options—all within the app.

In order to establish trust on the platform, there is a review and rating system so that farmers and buyers can post their experiences. Because it is built with Android Studio, the app is compatible with a broad spectrum of Android devices, so it is easily accessible to many. Its architecture also makes it easy to update and upgrade in the future, so it can develop and adapt with new technology and shifting user demands.

ACKNOWLEDGEMENT

First of all, we indebted to the **GOD ALMIGHTY** for giving me an opportunity to excel in our efforts to complete this project on time.

We express our sincere thanks to our respected dean **Dr. Md. Sameeruddin Khan**, Pro-VC, School of Engineering and Dean, School of Computer Science Engineering & Information Science, Presidency University for getting us permission to undergo the project.

We express our heartfelt gratitude to our beloved Associate Deans **Dr. Mydhili Nair,** School of Computer Science Engineering & Information Science, Presidency University, and Dr. **Anandaraj SP**, Head of the Department, School of Computer Science Engineering & Information Science, Presidency University, for rendering timely help in completing this project successfully.

We are greatly indebted to our guide **Ms. Bhavya B, Assistant Professor** and Reviewer **Dr. Sharmasth Vali Y, Associate Professor**, School of Computer Science Engineering & Information Science, Presidency University for his/her inspirational guidance, and valuable suggestions and for providing us a chance to express our technical capabilities in every respect for the completion of the project work.

We would like to convey our gratitude and heartfelt thanks to the CSE7301 University Project Coordinators **Dr. Sampath A K and Mr. Md Zia Ur Rahman,** department Project Coordinators **Dr. Sharmasth Vali Y** and Git hub coordinator **Mr. Muthuraj.**

We thank our family and friends for the strong support and inspiration they have provided us in bringing out this project.

LAKKI REDDY VARSHITHA - 20211CIT0049

ERANTI SAI KISHAN - 20211CIT0052

ERANTI SAI DINESH - 20211CIT0053

PAVANI M - 20211CIT0067

LIST OF TABLES

LIST OF FIGURES

TABLE OF CONTENTS

CHAPTER-1
INTRODUCTION

Agriculture is a crucial sector in sustaining economies and livelihoods for millions of individuals globally. The conventional agricultural market system, though, has too many middlemen. This not only increases the cost but also eats into the farmers' income, leaving them with very little profit. Most small farmers find it difficult to access the market directly, which makes it more challenging for them to compete on an equal basis. These include issues of delayed payments, poor information about the market, and limited provision of financial and transport services.

To assist in resolving these issues, this project proposes a mobile application that links farmers directly with customers—either consumers, wholesalers, or retailers. By cutting out the middlemen, farmers receive higher prices for their produce. The app provides functionalities such as real-time tracking of inventory, secure digital payment, and intelligent market insights with AI, enabling farmers to make better-informed decisions and maximize their revenues.

Simplicity and ease of use are a major focus in the app, remembering that not all farmers are technology literate. It can work with many languages and voice commands, so it is accessible to a larger group. The app also has beneficial sections such as information about government schemes, financial assistance, and advice from experts to assist the farmers in boosting productivity and making their farms more sustainable.

In addition to market access, the app assists with transport by providing car hire services to transport produce. Other upcoming updates can extend to fertilizer tracking tools, AI-based crop recommendations, and land hire services, further increasing the utility of the app.

By employing state-of-the-art technologies such as blockchain for secure payments, AI for price forecasting and weather forecast, and GPS for logistics tracing, this initiative plans to disrupt agricultural trade. It not just plans to increase farmers' revenue but also to provide consumers fresh, quality products at reasonable prices. Overall, the app serves to enable an open, efficient, and sustainable system that enhances both rural growth and economic progress.

1.1 **Background**

Agriculture is significant to the economy, but middlemen and lack of market access leave farmers making little profit. Direct Market Access for Farmers Android application, developed through the use of Android Studio, makes it possible for farmers to go directly to customers, wholesalers, and retailers. It's set to eliminate intermediaries and, therefore, enables farmers to get fair returns from their crops as well as increasing transparency in selling and buying agriculture products. By tapping into the use of smartphones, the app allows farmers to post produce, price, and talk to consumers, wholesalers, or retailers. Facilities such as real-time pricing, safe payment, and logistical support boost effectiveness. This initiative fosters agricultural digitalization, empowering farmers and enhancing market availability.

1.1.1 Evolution of Direct Market Access App for Farmers

Farmers have long depended on intermediaries to market their crops, which usually restricted their incomes and access to larger markets. But with the advent of digital technology, mobile applications have become a powerful tool to enable farmers to link directly with customers. In the early stages, these systems were primarily based on SMS to send market price information. But with smartphones becoming increasingly ubiquitous, more sophisticated apps emerged that provide real-time pricing, secure payment systems, and even facilitate transportation and logistics. Platforms such as Android Studio and cloud-based services have gone a long way in simplifying these apps to use, speeding them up, and making them scalable. The Direct Market Access App for Farmers leverages these innovations to facilitate fairer trade and enhance farmers' means of selling their goods.

1.1.2 Challenges in Traditional Agricultural Markets

- Presence of Middlemen
- Lack of Direct Consumer Access
- Price Manipulation & Unfair Trade Practices
- Delayed Payments & Financial Insecurity
- Limited Awareness of Government Schemes & Support
- Logistical & Transportation Issues
- Low Digital Literacy & Technology Barriers

1.2 Project overview

The Farmers' Direct Market Access Mobile App, created with Android Studio, aims to fill the gap between farmers and consumers by providing a straightforward, transparent, and efficient market. It empowers farmers by enabling them to list their products, determine their own prices, and link directly with consumers, retailers, and wholesalers without the middlemen. Some of the most prominent features include real-time price updates, secure payments online, demand prediction, and integrated logistics support to assist with smooth transactions and timely delivery. The application is user-friendly, with a simple layout and multi-lingual support, making it accessible even for farmers who are not very tech-friendly. A part from providing trading tools, the app also offers valuable insights about the market and weather conditions to enable farmers to make more informed decisions and reap higher benefits. Leveraging the capabilities of smartphones, the app facilitates the digital revolution of agriculture, enabling farmers to get remunerative prices, increase their income, and enhance the efficiency of the market as a whole.

1.3 Requirements and Specifications

Category	**Specification**
Platform	Android (Developed using Android Studio)
Operating System	Android 7.0 (Nougat) and Above versions
Programming Language	Java, Kotlin
Database	MySQL / SQLite (Via REST API)
Backend	Node.js with Express
UI Framework	XML (Android UI Design), Jetpack Compose(Optional)
Development Tool	Android Studio(Latest Stable Version)
Cloud Services	Firebase Authentication, Firebase Storage (if needed)

APIs Used	Google Maps API (for location services), Payment Gateway API (e.g., Razorpay, Stripe)
Security	User authentication via Firebase Auth or OAuth
Storage Requirements	Minimum 1GB of device storage
RAM Requirement	Minimum 4GB RAM for smooth performance
Internet Requirement	Required for online transactions and updates
Additional Features	Push Notifications (Firebase Cloud Messaging), Chat Support (if needed)

Table 1.1: System Requirements and Technical Specifications

1.4 Problem Statement

Farmers tend to find it difficult to receive a fair price for their produce because of middleman control, insufficient current market prices, and restricted access to customers. Conventional modes of selling tend to leave them with low margins, late payments, and plenty of inefficiencies. Farmers also lack adequate support for transport, are unable to monitor what is in demand, and lack a digital platform that suits their requirements. On the other hand, customers such as retailers and wholesalers also encounter problems in availing fresh fruits and vegetables directly from farmers. There is no direct and open market that results in price variations, unequal bargains, and negligible bargaining power on the part of the farmers. A Direct Market Access mobile application, built with Android Studio, can assist in addressing these issues. It will provide farmers and consumers with an easy means of connecting, with functionalities such as real-time market price updates, secure transactions, direct messaging, and logistics assistance. This type of platform eliminates middlemen, making the whole process more efficient and transparent. With the increasing popularity of smartphones, this application can provide farmers with greater control over their sales, increase their income, and facilitate fairer trade. It also streamlines the supply chain and brings the farm market into the digital era.

CHAPTER-2

LITERATURE SURVEY

Title of Paper	Author (s)	Year	Method Used	Result Obtained	Drawbacks of the Method
Mobile based agricultural Apps and portals for farmers' welfare in India	Manobharathi K., Ph.D. Scholar, Department of Agricultural Extension	2021	Mobile applications for agricultural information dissemination, such as Kisan Suvidha, providing weather, market prices, and pest control updates.	Improved decision-making: Farmers get real-time weather and market updates, helping them make informed choices.	Limited awareness and digital literacy: Many farmers are unaware of these apps or struggle to use them effectively.
Mobile Based Agricultural Apps and Portals for Farmers' Welfare in India	N. Anandaraja, Programme Coordinator, ICAR-Krishi Vigyan Kendra	2021	The Paper discusses the development of mobile applications and portals to provide farmers with real-time agricultural information, market intelligence, weather forecasts, and advisory services.	Improved trust and traceability in transactions, reducing fraud.	High computational costs and lack of farmer familiarity with block chain technology.
E-Mandi: Digital Marketplace for Farmers	Prof. Tushar Phadtare, Nisha Yanbhar, Komal Nimbalkar,Rushikesh Wakode, and Prasana	2022	Digital platform to connect farmers with buyers via a bidding system for agricultural produce.	Enhanced competition leading to better price discovery for farmers.	Requires high internet penetration; limited adoption in areas with low digital literacy.

Agro Connect: AI-Driven Marketplace for Farmers	Aditya Shinde.	2022	AI-powered app recommending optimal prices and connecting farmers to nearby buyers based on demand.	Reduced time to market and better price optimization.	High processing demands and challenges in adapting AI models to regional variations.
Farmers' Direct Selling Mobile App	Hossein Abbasi, Maysam Or0uskhani, Samaneh Asgari, Sara Shomal Zadeh	2023	Mobile application enabling farmers to list products and directly negotiate prices with consumers.	Simplified selling process and higher profit margins for farmers.	Limited support for logistics and delivery, restricting usability for large-scale operations.
Mobile Based Agricultural Apps and Portals for Farmers' Welfare in India	Manobharathi K. and N. Anandaraja.	2023	Blockchain-based supply chain for tracking agricultural produce from farm to market.	Increased transparency, reduced middlemen exploitation.	High energy consumption and complex implementation costs.
Direct Market Access for Farmers	Sribharathi B., Parthasarathy G., Harini R., and Surya C.	2024	The research proposes a digital platform that allows farmers to sell their produce directly to buyers, eliminating intermediaries.	The adoption of these mobile applications has improved farmers' access to information, helping them make informed decisions about crop management, pest control, and market prices.	Despite the benefits, limited digital literacy and lack of internet connectivity in rural areas hinder widespread adoption. Additionally, farmers' reliance on traditional farming practices and middlemen remains a challenge

Mobile App for Direct Market Access for Farmers	A. Kumar and B. Patel.	2024	The research proposes the development of a mobile application, FarmConnect, to provide farmers with direct market access by connecting them with consumers and retailers.	During alpha testing, the system successfully allowed farmers to list and manage their produce while enabling real-time updates	While the system effectively connected farmers and consumers, it lacks real-time transaction monitoring
Mobile App For Direct Market Access For Farmers	Sheetal Phatangare, Sandhyarani Lavhare, Sneha Ingle, Nirmal Chaudhari.	2024	GPS-Based Direct Market Access App: A mobile application that connects farmers directly with consumers, wholesalers, and retailers, eliminating middlemen and allowing real-time price negotiation.	Improved Market Accessibility: Farmers gain direct access to buyers, increasing profits and reducing dependency on intermediaries.	Limited Digital Infrastructure: Many rural areas lack proper internet connectivity and smartphone access, making adoption challenging.
Farmers E-Commerce Mobile Application	S. R. Patil, P. S. Patil, and S. S. Patil.	2024	"HarvestHub" E-Commerce Platform: A mobile application that allows farmers to buy and sell agricultural products like fruits, vegetables, seeds, and fertilizers in their local language, enhancing market accessibility.	Enhanced Market Access: Farmers can now directly connect with buyers, eliminating middlemen and improving their profits.	Digital Divide: Many farmers still lack access to smartphones and internet connectivity, limiting adoption.

Table 2.1: Literature Survey

2.1 Historical Development

Mobile apps and digital tools over the years have been central to changing the way farmers access markets. The most significant developments enumerated below are the critical steps taken to develop a Direct Market Access App for Farmers based on Android Studio.

2.1.1 Traditional Agricultural Marketing

Previously, farmers used to sell their produce mostly through local markets, wholesalers, or middlemen. This usually resulted in unequal pricing since the farmers did not have any control over how much they were paid. Without direct access to consumers, they could not negotiate higher rates and had to take whatever price the middlemen were willing to give. Often, they also did not have up-to-the-minute data concerning the prices of goods in the market, and thus they were compelled to sell their products at prices lower than their value. To make matters worse, problems such as bad roads and late payments compounded their woes. Despite the fact that this old system had existed for generations, modern farming no longer required it. There is now a clear demand for direct market access, where farmers sell directly to consumers, receive reasonable prices, and have greater financial stability.

2.1.2 Introduction of Government Marketplaces

Governments implemented controlled farm markets such as India's Agricultural Produce Market Committees (APMCs) to enable farmers to receive fair prices, provide transparency, and introduce consistency in trading procedures. These markets were intended to shield farmers from exploitation by middlemen and to provide them with a proper place to sell their produce. Still, due to these efforts, issues such as excessive bureaucracy, rigorous rules, and the persistence of middlemen rendered the system ineffective. These issues frequently hindered farmers' direct access to buyers. To make the situation better, initiatives in the digital space such as eNAM (National Agriculture Market) were initiated. These platforms seek to enhance the trading process through modernization, enabling the online sale of agricultural products by farmers at favorable prices while cutting their reliance on conventional market systems.

2.1.3 Emergence of E-Agriculture Platforms

With the advent of the internet in the early 2000s, e-agriculture websites came into being, introducing technology-driven innovations to agriculture and enhancing market access.

These sites gave farmers critical information like market prices, weather, agricultural advice, and information regarding government schemes, through websites, SMS, and mobile services. Online trading platforms enabled farmers to deal directly with wholesalers and retailers, bypassing middlemen and enhancing profit margins. Both governments and non-governmental organizations created online marketplaces to enable direct sales and equitable pricing. Although these platforms enhanced transparency and efficiency, the mass adoption was initially hampered by low internet connectivity and limited digital literacy in rural communities.

2.1.4 Growth of Mobile Technology in Agriculture

The advent of mobile technology in agriculture has revolutionized the way farmers access markets, obtain information, and receive financial services. With the extensive use of smartphones, mobile applications now enable farmers to directly access consumers, view real-time prices, and make secure online transactions. GPS-based weather forecasts, pest identification, and AI-based recommendations have also increased productivity. Moreover, mobile banking and electronic payment systems have fastened up and secured financial transactions, lessening the use of cash-based systems. These technologies have empowered farmers with more control over their produce, enhanced market efficiency, and softened the grip of middlemen.

2.1.5 Advancements in Android App Development

Android Studio has also simplified and eased mobile app development, enabling developers to develop effective and user-friendly apps for sectors such as agriculture. With support such as Firebase for real-time databases, safe payment methods, GPS tracking, and push messages, apps have improved functionality and ease of accessibility, making it simpler for farmers to reach out to customers. Furthermore, the implementation of tools such as Jetpack Compose, machine learning APIs, image recognition APIs, and data analytics has enhanced the general user experience and offered effective decision-making support. Owing to these improvements, Android-based agricultural apps have grown to be more scalable, secure, and efficient, making it possible for farmers to access markets directly and up-to-date information.

2.2 Techniques for Mobile App for Direct Market

2.2.1 Android Studio for Development

Android Studio is the primary application for developers to design Android applications. It is a robust and adaptable environment that simplifies and speeds up building applications. It includes inbuilt facilities for user interface design, debugging and resolving bugs, and optimization of the app's performance. Code can be written either in Kotlin or Java, and the backend of the app can be integrated with Firebase or a bespoke server to manage data. With frequent updates and a massive pool of users, Android Studio is a trusted platform for app development that can scale and evolve in the future.

Key Features of Android Studio:

- Code Editor and Languages Support
- User Interface (UI) Design Tools
- Built-in Emulator and Testing Tools
- Integration with Backend Services
- Performance Optimization Tools
- Version Control and Deployment

2.2.2 Firebase for Database and Authentication

Firebase would be a solid option for the app's backend since it offers capabilities such as a real-time database, user authentication, and cloud storage. Through Firebase, farmers can sign up in a straightforward manner, post their products, and process transactions securely. Firestore, its database product, synchronizes data in real-time—so items such as market prices and stock levels are always current. Firebase Authentication simplifies and secures logging in, whether users like to log in via email, phone number, or social media. And Firebase can scale up easily, so even as more and more users come aboard, the app will still be running smoothly.

2.2.3 Secure Payment Gateway Integration

We have introduced secure payment methods such as Razorpay, Phonepay, and UPI to enable users to make transactions with ease, security, and convenience. This helps buyers to pay directly to the farmers in an instant, minimizing the use of cash transactions. Encryption

methods safeguard user information and deter fraud. Various payment methods such as credit/debit cards and digital wallets provide ease of use. A secure refund and dispute resolution process fosters user trust.

2.2.4 Push Notifications for Updates

Push notifications update users on price fluctuations, new listings, and order status. Farmers are notified of inquiries from buyers and payment receipt to enhance response time. Buyers are notified of seasonal fruit and vegetable availability and discounts. Firebase Cloud Messaging (FCM) provides an efficient and affordable way to deliver notifications. Regular updates keep users engaged and informed to make the best decisions.

2.2.5 Data Analytics for Market Insights

Data analysis enables farmers and consumers to make informed decisions using trends and demand patterns. The app can apply AI-based tools to forecast price fluctuations and recommend the optimal times for selling. Farmers receive insights about consumer tastes, allowing them to cultivate in-demand crops. Consumers can monitor price fluctuations and purchase at low costs. Graphs and charts act as visualization tools to enhance user comprehension of trends.

2.2.7 Multi-Language and User-Friendly UI

A simple multilingual interface assures farmers from varied regions access. The application can be designed to support two or more languages so that farmers who communicate in different regional languages can use it easily. Including voice assistance and chatbots can assist those who are not so tech-savvy, and it will become easier for them to learn and operate the application. A neat and simple design with big icons and buttons will make it user-friendly, even for those who are not habitual of smartphones. Accessibility features that enhance the user experience will also enable more people to use the app, and wider adoption and usage will ensue.

2.3 Future Directions and Research Opportunities

2.3.1 Integration of AI for Price Prediction

Artificial Intelligence (AI) can be used to forecast crop prices based on historical trends, demand in the market, and external factors such as weather and inflation. Based on these,

farmers can decide when and where to sell their crops. More research can be done on more sophisticated machine learning methods to enhance the accuracy of these forecasts.

2.3.2 Blockchain for Secure Transactions

Blockchain technology can secure and make transactions more transparent by maintaining an unalterable, permanent history of all transactions. This makes trading fair, minimizes the possibility of fraud, and establishes trust between buyers and farmers. Nonetheless, for blockchain technology to be applied extensively in the future, researchers must identify mechanisms to make it serve many users without hindrance and lower the cost per transaction.

2.3.3 IoT-based Smart Farming Integration

By employing the Internet of Things (IoT) in agriculture, farmers are able to monitor vital aspects such as soil conditions, weather conditions, and crop health — in real time. Smart sensors placed in the fields can gather such information and provide farmers with insightful suggestions to enable them to make informed decisions, increase crop output, and save waste. One of the most important areas of research is how to make these IoT devices more accessible and feasible for small farmers who might not have large budgets.

2.3.4 Expansion to International Markets

The app would be able to expand to cater to international markets, enabling farmers to participate in cross-border agriculture trade. In order to facilitate this, it would have to support various currencies, integrate with international shipping providers, and adhere to other countries' rules and regulations. Strategies to improve global trade accessibility for farmers are areas that can be researched further.

2.3.5 AI-powered Chartbots for Farmer Support

AI-powered chatbots can offer real-time assistance to farmers by responding to questions on crop prices, weather forecasts, pest management, and optimal farming techniques. Chatbots can be made available in local languages for accessibility. Studies can be aimed at enhancing chatbot accuracy and voice-based AI support.

2.3.6 Data Analytics for Market Trends and Insights

Advanced data analytics can be used to determine market demand, demand patterns, and consumer tastes, and farmers can change their production accordingly. Analytics powered by AI can also tell farmers what crops are in greatest demand and during which periods. There

are research opportunities to build predictive models that can optimize the supply chain efficiency.

2.3.7 Sustainable and Eco-friendly Farming Practices

The app may support sustainable agriculture by sharing information on organic cultivation practices, water harvesting techniques, and environmentally friendly pest management. Including measures of sustainability in the app can motivate farmers to practice sustainable agriculture. Subsequent research can investigate the options for rewarding sustainable agriculture through policies of the government and reward systems through apps.

CHAPTER-3
RESEARCH GAPS OF EXISTING METHODS

Existing agricultural market structures tend not to offer farmers direct access to buyers, resulting in reliance on intermediaries, price gouging, and inefficiency. Other digital solutions have been created, but they are hindered by usability, accessibility, and technological embeddedness limitations. This section discusses the essential research gaps of current farmer direct market access methods.

3.1 Deficiency of Immediate Farmer-to-Consumer Connectivity

3.1.1 Absence of a Special Digital Platform

One of the main factors hindering the establishment of a robust farmer-to-consumer relationship is the lack of an inclusive and easy-to-use digital platform. Although there are some websites and mobile applications that offer agricultural data, hardly any offer direct transactions between the consumer and the farmer. Due to the absence of a structured digital marketplace, farmers turn to traditional supply chains, which incorporate intermediaries that claim a large portion of profits.

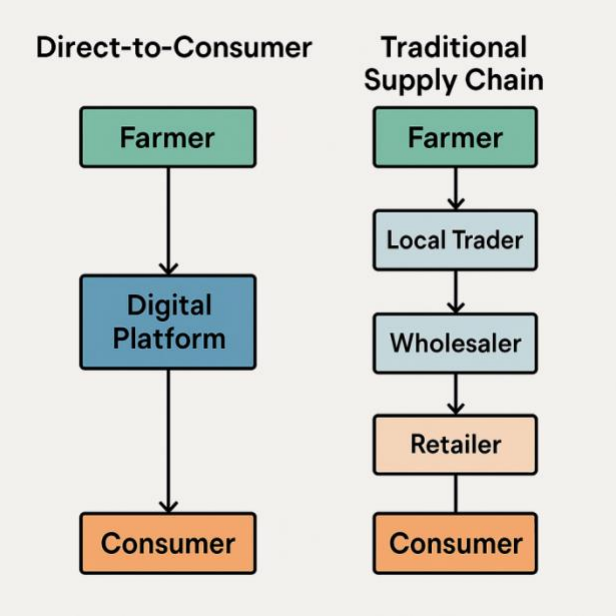


Figure 3.1.1 Direct-to-Consumer Vs Traditional Supply Chain

3.1.2 Farmers' Poor Digital Literacy

Most farmers, particularly in rural regions, lack digital literacy and are not aware of utilizing mobile apps for commercial transactions. Lack of training and awareness does not allow them to make use of technology to access consumers directly. Even if mobile apps exist, most farmers are unable to overcome issues related to language, intricate interfaces, and apprehension towards digital transactions.

3.1.3 Middleman Dependence in Sales

Historically, farmers have relied on middlemen like wholesalers and distributors to market their produce. Although these intermediaries facilitate logistics and distribution, they tend to play games with prices, and farmers end up with reduced profits. The lack of a direct-to-consumer model keeps farmers from receiving the actual value of their crops, which further discourages them from adopting digital platforms.

3.1.4 Unstable Internet and Technology Infrastructure

Many rural areas still suffer from unreliable internet connectivity and poor technological infrastructure; hence, it is hard for farmers to continually access online platforms. Sometimes frequent network disconnections, low smartphone penetration, and unaffordable data all present barriers to reaching consumers directly through such applications. Even if such apps exist, they often fail to work properly because of networks getting disconnected, which leads to defects in real-time communication and transactions.

3.1.5 Limited Knowledge of Market Prices and Consumer Demand

Without direct market access, farmers do not get immediate information about the current market prices and consumer demand. They either end up underpricing or overpricing their crops, causing financial losses or stockpiles. A efficient digital platform would have to include price updates and demand trends and enable farmers to make data-driven decisions.

3.1.6 Logistical and Delivery Issues

Even if farmers are able to reach consumers, efficient delivery of their products is still a challenge. In the absence of an integrated supply chain or logistical support, farmers are unable to ensure timely delivery, product freshness, and order fulfillment. The absence of integrated transportation solutions in farmer-to-consumer models deters both parties from making direct transactions.

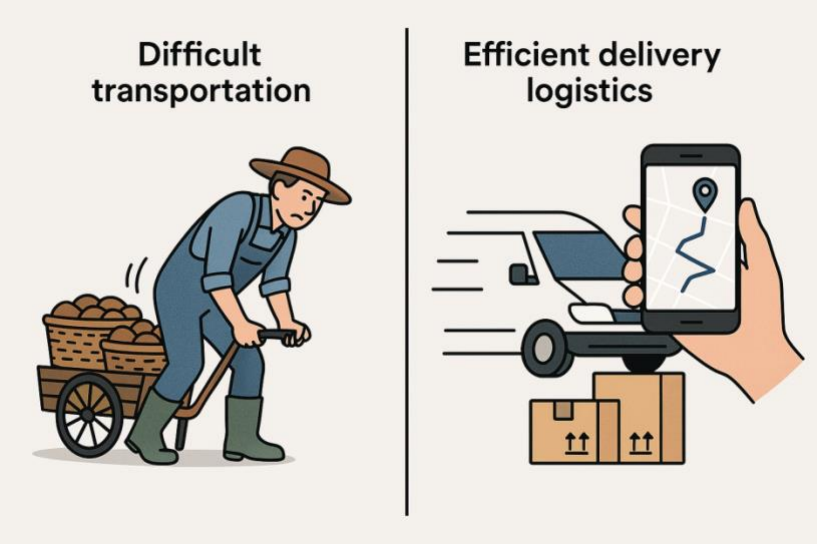


Figure 3.1.6 Traditional transportation Vs Modern transportation

3.2 Market Fluctuations and Price Volatility

3.2.1 Imbalances in Supply and Demand

One of the most basic causes of price volatility in agricultural markets is supply-demand imbalance. When demand is less than supply, prices go down, resulting in economic losses for the farmers. In contrast, when demand outstrips supply as a result of bad harvests, bad weather, or unanticipated rises in consumption, prices escalate. Such unpredictability makes it hard for farmers to organize their production and revenues in an efficient manner.

3.2.2 Crop Seasonal Patterns of Production

Crop production is mostly subject to seasonal patterns. Some crops are cultivated in certain seasons, so there is surplus during harvesting times and less supply during off-seasons. Prices fall when there is a large harvest and increase when there is a shortage. Farmers who depend on produce from a single season usually experience volatile incomes due to these fluctuations. Proper storage facilities and improved market access can mitigate these risks.

3.2.3 Effects of Weather and Climate Change

Extreme weather occurrences like floods, droughts, hurricanes, and unexpected rainfall may significantly affect agricultural production, leading to unpredictable shifts in prices. When there is low production of crops due to unfavorable weather conditions, prices quickly increase, but when there are plenty of crops from good weather conditions, prices decrease. Increasing impacts of climate change render this price volatility even more unpredictable, as

it becomes more difficult for farmers to plan and operate their operations in the long term.

3.2.4 Government Policies and Market Regulations

Government policies, including minimum support prices (MSP), import/export controls, and subsidies, can have a considerable impact on price volatility. Although most of these policies are intended to stabilize prices and benefit farmers, any abrupt policy shifts can lead to uncertainty. For instance, if a government issues an export ban on specific crops, it could cause over-supply in the domestic market, reducing prices. At the same time, eliminating subsidies increases the cost of production, requiring farmers to push their products upwards.

3.2.5 Global Market Influences

Farm prices are not only affected by domestic market conditions but also by international forces like foreign trade policies, exchange rates, and political developments. For instance, if a key wheat-exporting nation faces a drought, the world price of wheat goes up, and that, in turn, impacts domestic market prices. Likewise, shifts in fuel prices can raise the cost of transportation, which subsequently has a backdoor impact on farm product prices.

Figure 3.2.5 Global Agricultural Trade Routes

3.2.6 Middlemen's Role and Market Speculations

Middlemen and traders are also important in determining prices. At times, they stockpile commodities in order to cause artificial shortages and push up prices. Speculations in the market and investments in the stock market in agri-commodities also lead to price fluctuations. Therefore, it is important that farmers have direct access to market to prevent unnecessary price manipulation and earn more from their commodities.

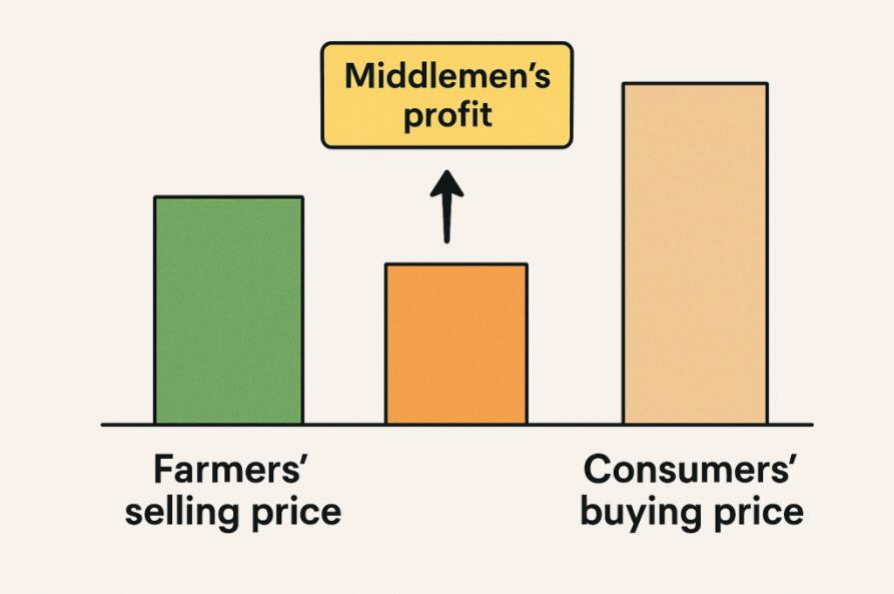


Figure 3.2.6 Price Gap in Agriculture: Farmers' Sales vs Consumers' Costs

3.3 Organic Farming and Sustainable Practice Challenges

3.3.1 High Upfront Costs and Investment

Switching to organic agriculture involves a substantial initial cost. Organic seeds, natural fertilizers, and biological pest control need to be bought by farmers. Moreover, gaining organic certification involves a high price tag and much time, which demands strict guidelines to be followed. This monetary cost tends to deter small-scale farmers from practicing organic farming.

3.3.2 Limited Market Access and Price Volatility

In spite of the growing demand for organic products, farmers face limited market access. Most organic farmers base their markets on local sites because there are no appropriate distribution channels. In addition, prices of organic produce fluctuate based on seasonal supply changes and consumer demand, posing a challenge for farmers to provide consistent income.

3.3.3 Pest and Disease Control

Organic agriculture limits the application of synthetic pesticides, and pest and disease control are more difficult. The farmer has to use crop rotation, biological control, and organic pesticides, which are less potent than synthetic ones. An unexpected infestation or

plant disease can cause extensive damage to the yield, and the farmer incurs economic losses.

3.3.4 Insufficient Awareness and Training

Most farmers do not have the right training and information to implement organic and sustainable agriculture. Organic farming is different from conventional farming as it demands expertise in soil management, composting, and natural pest control. Farmers will be unable to change successfully without adequate guidance, which will result in reduced productivity and inefficiency.

3.3.5 Regulatory and Certification Barriers

Organic certification requires intricate documentation, strict following of requirements, and regular checks. Small farmers are commonly challenged by such regulatory demands because they have limited financial and administrative capabilities. Moreover, the certification process takes time, holding their entry into organic markets at bay.

3.3.6 Soil Fertility and Productivity Problems

In organic agriculture, soil fertility is difficult to maintain without using synthetic fertilizers. Farmers have to adopt green manure, compost, and crop rotation methods to enrich soil nutrients. These practices take time to become effective, and yields during the initial years might be less than in traditional agriculture. This lag period dissuades farmers from switching over to organic practices.

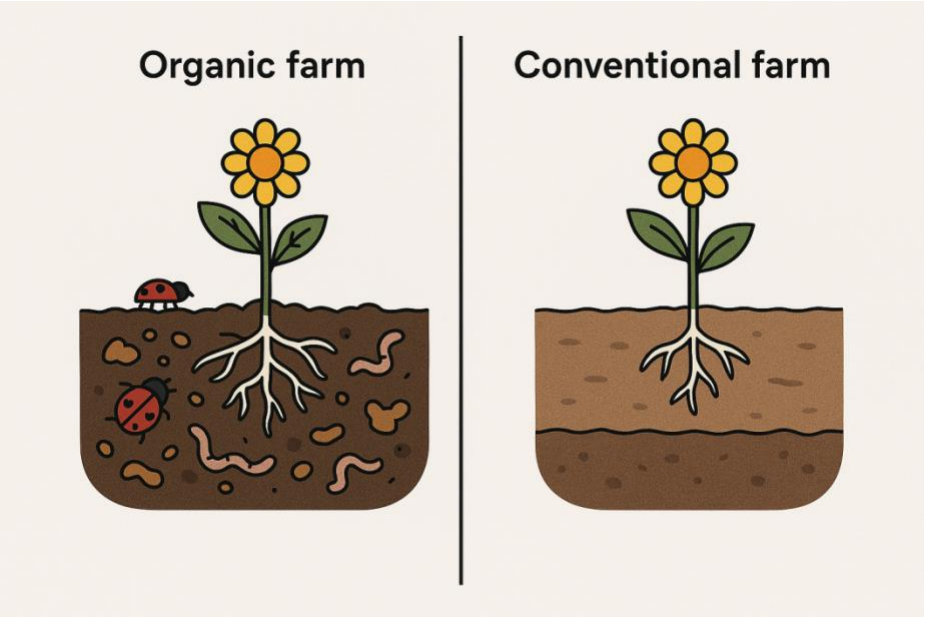


Figure 3.3.6 Organic Farm Vs Conventional Farm

3.4 Farmer Cooperatives and Support Systems Lack

3.4.1 Weak Bargaining Power

Farmers as individuals do not have bargaining power to negotiate favorable prices for their produce in the absence of cooperatives. Middlemen and traders tend to exploit this weakness, purchasing goods at low prices and selling them at much higher prices. This drains farmers' profits and discourages them from expanding their production.

3.4.2 Inadequate Access to Market Information

Farmers who lack a support system find it difficult to access current market prices, demand patterns, and government programs. This absence of information results in uninformed selling decisions, which in turn lead to monetary losses. A cooperative can offer collective resources to keep farmers informed about the most favorable market opportunities.

3.4.3 Poor Infrastructure and Logistics

Transport, storage, and distribution are still issues of concern for independent farmers. In the absence of cooperative networks, farmers struggle to transport goods collectively, resulting in high costs and post-harvest losses. Cooperatives can assist in establishing storage facilities, cold chains, and transport networks to enable direct market access.

3.4.4 Inadequate Financial Support and Credit Facilities

Farmer cooperatives can make it easier to access loans, subsidies, and government grants. In their absence, small farmers struggle to access financial assistance because of high interest or lack of collateral. Most farmers are compelled to borrow from informal lenders, resulting in debt traps.

Figure 3.4.4 Formal Vs Informal Financial Aid for Farmers

3.4.5 Challenge in Embracing Contemporary Farming Methods

Support networks usually offer training and technical support for new farming techniques, including organic farming, precision agriculture, and sustainable farming. Without such networks, farmers still use old and inefficient methods, resulting in decreased productivity and lower profits.

3.4.6 Failure to Respond to Bulk Demand by Major Buyers

A large volume of the produce required by supermarkets, food processing companies, and exporters is in bulk, which the individual farmers cannot supply. Cooperatives facilitate group marketing, which enables the farmers to supply in bulk and even obtain long-term contracts.

3.4.7 Inadequate Legal and Administrative Support

Most independent farmers are unaware of legal rules, taxation policies, and export standards. Farmer cooperatives usually possess legal advisors or legal departments that facilitate the complex process. Without access to such assistance, individual farmers struggle with bureaucratic obstacles that hinder them from expanding their businesses.

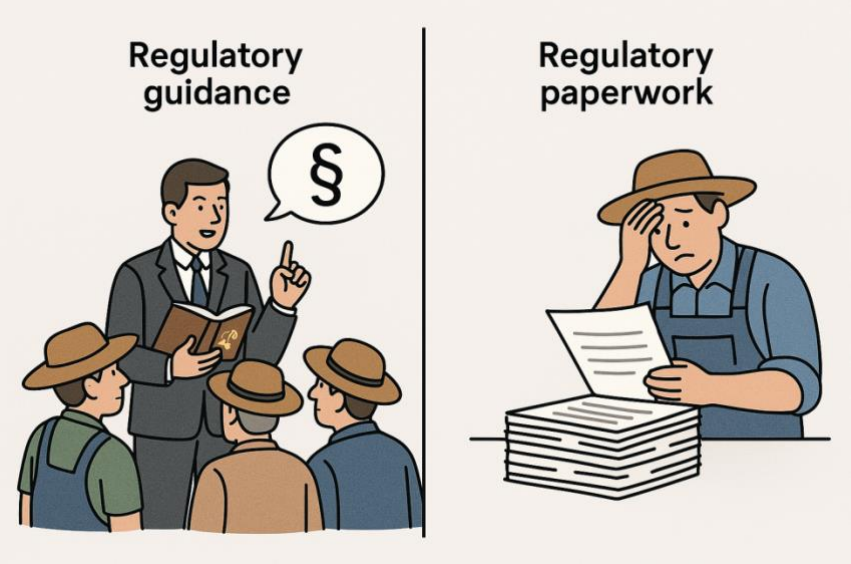


Figure 3.4.7 Legal and Administrative guidance

CHAPTER-4
PROPOSED METHODOLOGY

4.1 Requirement Analysis

Requirement analysis is an important stage in the lifecycle of development wherein the exact user needs and system expectations are properly established. In the case of a mobile app providing direct access to markets, the objective is to remove intermediaries, provide improved prices for farmers, and make the purchase and sale process easier for agricultural produce. The section below establishes the root problem, goals, and the requirements that fuel the process of developing the app.

4.1.1 Identification of the Problem

Farmers in conventional farm markets usually do not receive the appropriate prices for their crops as there are a series of intermediaries involved. Without direct connection with buyers and market information, farmers become subject to exploitation and suffer economic loss. Additionally, insufficient digital awareness and poor infrastructural facilities create an added hassle.

Problems Identified:

- Reliance on middlemen who shave the farmer's margin.
- Absence of instant price updates and information about the buyers.
- Lack of proper access to market trends and demand information.
- Inability to manage sales and logistics without an organized platform.

4.1.2 Purpose of the App

The main purpose of the mobile app is to create an immediate digital platform whereby farmers can link directly with customers, including consumers, retailers, and wholesalers. The app will bridge the gap between the producers of agriculture and the market through a straightforward, easy-to-use interface.

Major Objectives:

- Empower farmers to post and sell crops directly to customers.
- Offer live updates of market prices and demand.
- Enable free flow of communication between purchasers and vendors.
- Enable transparency in payments and transactions.

- Enable multi-language access to make it more useful for rural consumers.

4.1.3 Identification of Stakeholders

Identification of stakeholders is vital to know whom the app will be interacting with and how the app will address their requirements. The most prominent stakeholders in this project are:

- Farmers: Primary stakeholders who will list and sell their crops.
- Buyers: Consumers, retailers, wholesalers who will directly purchase from farmers.
- Administrators: Manage the functionality of the app, authenticate listings, monitor users, and settle disputes.
- Government/NGOs (optional): Can integrate for subsidies, training, or awareness.
- Developers: They are responsible for designing, coding, and maintaining the application.

4.1.4 Functional Requirements

Functional requirements outline the core functions and operations of the application:

- User Registration and Login: Farmers and buyers must be able to register and login with mobile numbers or email.
- User Profile Management: Users are able to manage their personal and business details.
- Product Listing and Browsing: Farmers can add their products; buyers can browse available produce.
- Search and Filter Functionality: Buyers are able to search produce by category, price, area, etc.
- Order Placement and Confirmation: Customers can place orders and monitor their status.
- In-app Chat or Call Feature: For immediate negotiation and communication.
- Payment Gateway Integration: To facilitate online payments.
- Real-time Notifications: To notify users of new offers, orders, or updates.
- Admin Panel: To control listings, user inquiries, and app content.

4.1.5 Non-Functional Requirements

Non-functional requirements guarantee the performance, usability, and reliability of the app:

- Scalability: The application shall be able to handle an increasing number of users and data without performance loss.

- Usability: The application interface should be easy, intuitive, and support several local languages.
- Reliability: The system shall not fail to work, even in places with low connectivity.
- Performance: Quick loading times and efficient use of resources.
- Security: All user information and transactions shall be encrypted securely.
- Availability: The application shall be available 24/7 with minimal downtime.

4.1.6 Platform Requirements

In order to make it widely accessible and compatible, the following platform requirements are taken into account:

- Operating System: Android (minimum version 8.0 and up).
- Development Environment: Android Studio with Java or Kotlin.
- Backend Server: Firebase (or alternative like Node.js with MongoDB) for authentication, database, and storage.
- Device Support: To run on smartphones with minimal hardware configurations (RAM 2GB+, internet-enabled).
- API Services: REST APIs for market price updates, payment gateway, and real-time chat.

4.2 System Design

System design is an essential activity in shaping functional requirements into an organized solution that guarantees usability, scalability, and performance. For the "Direct Market Access for Farmers" mobile application, system design comprises outlining the total architecture, developing user-friendly layout interfaces, and implementing a powerful and scalable database. This chapter details the three principal components: architecture overview, user interface, and database.

4.2.1 Architecture Overview

Architecture of the mobile app is based on client-server. Android app behaves as a client, while the cloud-based backend system serves as the server. There is data exchange between the mobile client and server using RESTful APIs across the internet. The architecture supports seamless interaction, data transfer, and real-time updates.

Components:

Client Side (Android App):

Developed in Android Studio.

Handles user input, data entry, and shows pertinent market information.

Uses APIs to send/receive data to/from the backend.

Server Side (Backend):

Done using technologies such as Firebase or Node.js with Express.

Handles authentication, data storage, business logic, and push notifications.

Database:

Cloud-based NoSQL/SQL database (e.g., Firebase Firestore, MySQL, or MongoDB) to store user profiles, market listings, and transaction information.

4.2.2 User Interface Design

The user interface (UI) is made simple and easy to use for farmers, who are particularly targeted, many being smartphone and app users for the first time. The UI adheres to Material Design Guidelines for Android to maintain consistency and ease of understanding.

Key UI Elements:

- Home Screen: Shows market trends, new products, and buyer/seller selection.
- Login/Register Screen: Safe login and registration through phone number or email.
- Product Upload Form: Enables adding crop information, quantity, and price by the farmer.
- Marketplace Feed: Buyers can browse and filter the crops/products offered by farmers.
- Chat/Contact Feature: Allows direct communication between buyer and farmer.
- Notifications: Gives alerts on new listings, orders, or price changes.

4.2.3 Database Design

The database is the core of the application, holding all the necessary data securely and efficiently. Depending on the backend option, either a relational database (e.g., MySQL) or a NoSQL database (e.g., Firebase Firestore) can be utilized.

Entities and Tables/Collections:

Users Table:

User ID (Primary Key)

Name

Role (Farmer/Buyer)

Contact Info

Location

Products Table:

Product ID (Primary Key)

Farmer ID (Foreign Key)

Crop Name

Quantity

Price

Availability Status

Date Listed

Orders Table:

Order ID (Primary Key)

Product ID (Foreign Key)

Buyer ID (Foreign Key)

Order Status

Date of Transaction

4.3 Module Description

The suggested mobile app is divided into various modules so that it becomes modular, scalable, and easy to implement. Every module has been framed so that it carries out some functionalities so that it becomes easy for communication among farmers, buyers, and admin.

4.3.1 Farmer Module

The farmer module is designed especially for farmers so that they are able to update their profiles, post their products, and even directly communicate with buyers.

Main Features:

- Registration/Login: Farmers can register/login securely using credentials or mobile OTP authentication.
- Profile Management: Enables farmers to update their personal and farm information.
- Product Upload: Farmers can upload images and product information (type,

quantity, price, availability) of products.

- View Orders: Farmers are able to view orders from customers and change the order status.
- Chat with Buyers: Individual contact with potential customers via chat.
- Notification Alerts: Instant alerts regarding new orders, customer messages, or admin messages.

4.3.2 Buyer Module

This module is specifically focused on the buyers interested in buying the produce directly from farmers.

Key Features:

- Registration/Login: Secure sign-up and sign-in facilities.
- Search and Browse: Bidders can browse products on sale by category, price, and location.
- Product Details View: View detailed information regarding the produce such as price, farmer details, and availability.
- Place Orders: Placing buy and payment through available means.
- Order Tracking: Order status tracking and shipping alerts.
- Chat with Farmers: Enables direct negotiation or question with farmers.

4.3.3 Admin Module

Admin module provides system administrators a way to administer and monitor the entire application ecosystem.

Major Features:

- User Management: Admin may administer buyer and farmer accounts (approve, deactivate, authenticate).
- Product Tracking: Track all listings of products regarding quality and regulation.
- Order and Payment Tracking: Keep order and payment history private and for records.
- Content Management: Moderate and manage objectionable content, complaints from users, or disputes.
- Analytics Dashboard: Graphical depiction of user action, sales, and product movement.

4.3.4 Communication Module

It allows farmers and buyers to communicate and keep everyone updated.

Key Features:

- In-App Messaging: Realtime chat using Firebase or equivalent backend.
- Notifications System: Push notifications for order status, new listings, messages, and announcements.
- Support Messaging: Enables users to contact admin for feedback or support.

4.3.5 Order and Payment Module

This module allows easy ordering and secure payment transactions between farmers and buyers.

Key Features:

- Cart and Checkout: Users can add items to cart and go for checkout.
- Order Summary and Confirmation: Users get order summaries in detail prior to confirmation.
- Payment Integration: Payment gateway integration (such as Razorpay, Google Pay, UPI) for online payments.
- Order History: Customers and farmers are able to view past orders and their status.
- Invoice Generation: Invoices automatically generated on successful payment.

4.4 Workflow Diagram

A workflow diagram will illustrate how the user (buyer or farmer) will interact with the system. It will consist of steps such as registration → login → listing of products (farmer) / browsing products (buyer) → placing orders → payments → feedback.

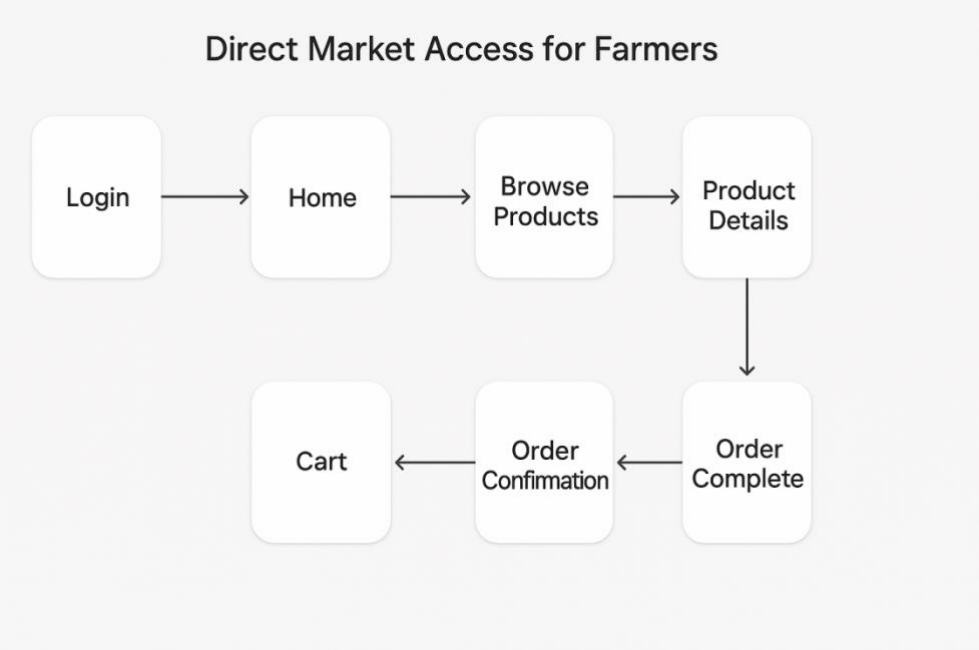


Figure 4.4.1 Workflow Diagram

4.5 Test Methodology

Testing is one of the pivotal phases of the mobile application life cycle, guaranteeing that the end product will be operational, trustworthy, and easy to use. In case of the targeted mobile application allowing direct access by farmers, test methodology is aggressively mapped to inspect both the back-end functionality as well as the front-end user interface. Following are the test types performed:

4.5.1 Unit Testing

Unit testing is all about testing individual parts or functions of the application separately. Unit tests were utilized in this project to test specific functionality like:

- User authentication and validation
- Input and form validation
- APIs to retrieve market price updates or crop information
- Local database operations (e.g., SQLite or Room DB)

Tools Used:

- JUnit for Java/Kotlin-based logic testing
- Mockito for dependency and service mocking
- Robolectric for Android-specific unit testing
- Unit tests catch bugs early by ensuring every module of the application works as expected regardless of other modules.

4.5.2 Integration Testing

Integration testing is done to ensure various modules of the application work in harmony together. In farmers' app, this involves testing interactions between:

- The user interface and the database
- The API services and UI components
- Authentication and session management
- Data transfer from input screens to display and storage modules

Tested Scenarios

- During login, checking if the user profile information is retrieved and displayed as expected
- Issuing a new query or request and checking if it gets cached and can be retrieved
- Checking data consistency between screens (e.g., list of crops, market price)

Tools Used:

- Espresso for UI checking and validation
- Android Instrumentation Tests for end-to-end module interaction
- These kinds of tests aid in discovering bugs that might occur due to the interaction among different components.

4.5.3 User Acceptance Testing (UAT)

User Acceptance Testing ensures that the application is meeting the needs and requirements of actual users — farmers and agriculture officers in this case, who would be using the application. UAT included real-world scenarios, usability testing, and observation of performance during this stage.

UAT Process:

- A panel of local farmers was chosen to pilot the app under a simulated live environment.
- Test cases were derived from typical user flows, including:
- Logging in and navigating available crop support services
- Viewing market prices for particular crops
- Sending a complaint or support inquiry
- Alerts for receiving agricultural update messages

Feedback Focused On:

- Navigability

- Clarity of language and imagery
- Speed of data loading and responsiveness
- Error handling (for example, what if the internet does not work)
- Minor adjustments to the design and easier wording were included as a result of feedback to enable a more user-friendly experience, particularly for users who might not necessarily be tech-savvy.

CHAPTER-5
OBJECTIVES

5.1 To Empower Farmers Through Technology

5.1.1 Encourage Digital Literacy Among Farmers

Digital literacy is key to inclusion of farmers in the digital ecosystem. Farmers, especially in rural settings, are not well exposed to smartphones, mobile apps, and web-based tools. The app has easy-to-use interfaces with local language support and step-by-step tutorials that enable farmers to understand basic app features like registration, product uploads, and market price checking. Digital literacy empowers farmers to be standalone users of technology, paving the way for broader opportunities and government programs.

5.1.2 Provide Access to Real-Time Market Information

Historically, farmers have relied on intermediaries or local merchants for price discovery, thus being exploited most of the time. This application gives real-time information on prices in the markets, trends in demand, as well as needs of buyers across different regions. It enables farmers to make accurate decisions on where, when, and at what price to sell their crops. Access to real-time markets diminishes reliance on intermediaries and enables farmers to maximize profits.

5.1.3 Enable Direct Communication Between Farmers and Buyers

The app allows direct communication between farmers and prospective buyers like wholesalers, retailers, exporters, and consumers. In-app messaging, chat, or sharing of contacts feature eliminates the conventional obstacles in the supply chain. Transparency, improved negotiation, and customized offers are the advantages of direct communication. This fosters trust and long-term partnerships among producers and consumers.

5.1.4 Enhance Farmer Profitability

Through the elimination of intermediaries and providing instant market access, the app guarantees farmers a just price for their produce. Dynamic pricing, bulk selling capabilities, and marketing tools assist farmers in generating more revenue. The app can also recommend the optimal markets using price patterns and past data to facilitate wiser choices.

5.1.5 Encourage Self-Sufficiency and Entrepreneurship

The app also seeks to instill self-sufficiency and agri-entrepreneurship in farmers. Through market intelligence, logistics help, and financial enablement, farmers can run their farm produce as miniature businesses. It pushes them to diversify crops according to demand, venture into value addition (e.g., organic farming, processed foods), and even venture into direct-to-consumer platforms.

5.2 To Design an Easy-to-Use Mobile Application

5.2.1 Easy and User-Friendly Interface

- Make the application easy to use a top priority.
- Implement Material Design for Android to make it uniform and easy to understand visually.
- Buttons and icons must be large and properly labeled.
- Prevent cluttered navigation and ensure that it is easy to move around. (e.g., bottom navigation menu or hamburger menu)
- Visual indicators like images and tooltips can be used to inform low-literacy users about features.

5.2.2 Multi-Language Support

- As the app is intended for farmers across various locations, multilingual support

(e.g., Hindi, Telugu, Tamil, etc.) is essential.

- Utilize Android strings.xml resource files for localization.
- Offer language choice on initial setup and enable setup in settings.

Advantages:

- Enhances user uptake and knowledge of the app.
- Reduces misunderstanding of app capabilities.

5.2.3 Fast Registration and Login

- Keep onboarding quick, easy, and convenient.
- Implement mobile number verification through OTP for easy sign-up.
- Supports optional fingerprint biometric authentication for those devices that support it.
- Provide third-party login capabilities such as Google Sign-In for users who are already accustomed to them.

Objective: Reduce friction and enable users to use the app without technical complications.

5.2.4 Generate Listings Dashboard

There must be a central dashboard for farmers to oversee their produce within the app.

Dashboard features:

- Add/edit/delete produce listings.
- Upload produce photos.
- Set price and quantity.
- View interest or buy orders from buyers.

Design Tip: Apply cards or lists with visual markers (i.e., green = available, red = out of stock).

5.2.5 Offline Availability and Syncing

There are certain farmers in remote or poorly connected areas, and offline availability is essential in such a scenario.

Users must be able to:

- Add or modify listings offline.
- Access offline pre-loaded data.
- Apply Room Database for local storage and WorkManager for background syncing in case of internet connectivity.

Sync Features

- Automatically upload any changes on device reconnection.

- Notify users when sync has completed successfully.

5.3 In Order to Facilitate Safe and Transparent Transactions

5.3.1 Integrated Payment Gateway

The application must utilize secure and common payment gateways like Razorpay, PayPal, or UPI services (Google Pay, PhonePe, etc.) to provide seamless transactions. The gateway should provide various payment modes like credit/debit cards, net banking, and wallets.

Implementation in Android Studio: Use SDKs offered by the payment services that are PCI DSS compliant and securely encrypted.

5.3.2 Transaction History and Receipts

Users must be able to view a complete history of transactions, including date, time, product description, buyer/seller name, and payment method. Digital receipts must be automatically created and downloadable for each transaction.

Implementation Tip: Save transaction information in a secure Firebase Firestore/Realtime Database or a local SQLite database with backup sync.

5.3.3 Buyer and Seller Ratings

After every transaction, users should be asked to review and rate their experience. This provides accountability, trust, and credibility of users on the platform. Ratings can affect other users' decisions.

Implementation Tip: Use a 5-star rating system with optional text reviews, and keep ratings in user profiles.

5.3.4 Secure Login and Authentication

Use secure login features like two-factor authentication (2FA), biometric login (fingerprint, face recognition), and phone number OTP verification.

Implementation Tip: Add Firebase Authentication to enable secure user management with phone, email/password, and social login support.

5.3.5 Fraud Detection and Reporting

Add a reporting feature for users to flag suspicious behavior or activity. Supplement this with basic fraud detection algorithms that alert on anomalies such as listings that are copied, duplicate accounts, or abnormal transaction patterns.

Implementation Hint: Track user activity with back-end validation rules and admin

alerting. Provide in-app reporting with screenshots and commentary.

5.4 To Enhance Agricultural Market Access

5.4.1 Geo-Location Based Search for Markets

This allows utilizing the device GPS facility for locating local markets or buyers in a specified distance. Farmers will be able to:

- See interactive maps showing locations of markets in real-time.
- Filter markets by type of market (e.g., wholesalers, retailers, cooperatives).
- Use maps integrated (e.g., Google Maps API) for navigation.
- Find new markets by location and product category.
- Implementation: Apply Android's Location Services and Google Maps SDK to identify and render proper market places dynamically.

5.4.2 New Order Notification Alerts

To ensure on-time response and effective trade, the app shall issue push alerts for:

- New orders from buyers.
- Changes in order status (pending, accepted, fulfilled).
- Quotations or bidding opportunities.

Deployment: Utilize Firebase Cloud Messaging (FCM) for reliable and scalable delivery of push notifications, even when the app is in the background.

5.4.3 Delivery and Logistics Integration

The application will allow farmers to become connected with logistic providers or cooperatives that have transport services available. Some of the features can be:

- Order pickup scheduling.
- Real-time tracking of delivery.
- Displaying logistics partner ratings and prices.
- Sharing delivery status with buyers.

Implementation: Integrate third-party logistics APIs or develop a local logistics partner module that supports order-to-delivery workflows in the app.

5.4.4 Digital Market Calendar

This feature reminds farmers of:

- Market days, fairs, and trade events upcoming.

- Seasonal demand trends and promotions.
- Order cycles and payment dates.

Implementation: Design an interactive calendar with event labels and reminders. Events can be dynamically added by market authorities or app administrators using a backend dashboard.

5.4.5 Bulk Order Management

For enabling large-volume buyers or cooperatives, this feature shall:

- Permit farmers to see and accept bulk orders.
- Manage produce gathering and volume pricing.
- Provide coordination of multiple farmers to collectively satisfy large orders.
- Support functionality such as quantity tracking, estimated delivery schedule, and divided payment.

Implementation: Design a module for bulk orders with inventory tracking, farmer collaboration features, and status management (accepted, processing, delivered, etc.).

5.5 In order to Extend Support Services to Agriculture

5.5.1 Weather Forecast and Crop Advisory

Weather Forecast: Incorporate authentic weather data feeds (e.g., IMD APIs) to offer localized, real-time weather information like temperature, rain prediction, humidity, and wind speed.

Crop Advisory: Provide timely advice on sowing, irrigation, pest management, and harvesting depending on weather and soil health. This can be rule-based or AI-powered, assisting farmers in achieving maximum yield and minimizing losses.

5.5.2 Association with Government Schemes and Subsidies

Offer a separate section that provides government schemes, subsidies, and grants to farmers, state-wise and crop-wise.

Offer eligibility criteria, benefits, documents needed, and a direct link or phone number for application.

Push notifications may also notify users of new or soon-to-expire opportunities.

5.5.3 Expert Helpline and Chat Support

Add a live chat option or chatbot that puts farmers in touch with agricultural experts for instant problem-solving.

Insert call-back request or toll-free helpline number in low internet connectivity areas.

Cover general questions such as disease/pest identification, organic farming practices, fertilizer use, etc.

5.5.4 Crop Insurance Information

Insert information about crop insurance schemes offered, both government and private.
Premium rate, cover period, claim process, and eligibility.
Facility to calculate premium of insurance and reminder for deadline submission.

5.5.5 Knowledge Bank (Videos, FAQs, Blogs)

A knowledge bank with:
Educational farm videos on mechanization, techniques, and success stories.
FAQs to common farmer problems in simple language.
Blog articles and posts on new best practices, case studies, market trends, and sustainability.
Content must be downloadable off-line and categorized by crop type, region, and language.

5.6 To Promote Sustainable Agric Practices

5.6.1 Promote Organic Farming

Educational Materials: Add tutorials, articles, and video tutorials on organic farming techniques like composting, crop rotation, natural pest control, and the application of no synthetic chemicals.

- Guidance for Certification: Offer step-by-step guidance and assistance to farmers who are willing to become certified organic by national or local government.
- Sharing Community: Enable farmers to exchange success stories and tips within the app to inspire and educate other farmers.

5.6.2 Environmentally Friendly Packaging Tips

- Best Practices: Recommend the use of biodegradable, recyclable, or reusable packaging materials for fruits and vegetables.
- Cost-Effective Alternatives: Recommend low-cost eco-packaging materials available locally.
- Supplier Directory: Include sustainable packaging suppliers in the region to simplify procurement.

5.6.3 Reducing Waste and Maximizing Resource Efficiency

- Smart Planning Tools: Include crop planning and forecasting tools to minimize

overproduction.

- Composting Guidance: Educate farmers on how to transform waste into organic fertilizer.
- Water and Soil Conservation Tips: Provide tips on efficient water use (e.g., drip irrigation) and soil conservation techniques.

5.6.4 Local Marketplace Promotion

- Geo-Targeted Marketplace: Enable farmers to directly connect with local consumers, retailers, and local businesses via a geotargeted buying platform.
- Local Events & Fairs: Inform farmers that the next area agricultural fair or local farmer's market is available for produce to be sold.
- Cultural Branding: Engage farmers in marketing heritage, culturally appropriate, environmentally friendly local varieties.

CHAPTER-6
SYSTEM DESIGN & IMPLEMENTATION

6.1 System Architecture Overview

6.1.1. Three-Tier Architecture

The system is a three-tier architecture built on:

Presentation Layer (Frontend):

- Simple and intuitive mobile user interface developed using Flutter for both Android and iOS. This layer is utilized to leverage the native system capabilities like marketplace, advisory, and access to finance.

Application Layer (Business Logic):

- Built using Node.js or Django, this layer does all processing, validations, and logics such as matching farmers with buyers and generating personalized advisories.

Data Layer (Backend & Database):

- A Firebase or MongoDB database securely keeps all the relevant information like user profiles, market transactions, advisory messages, and alerts.

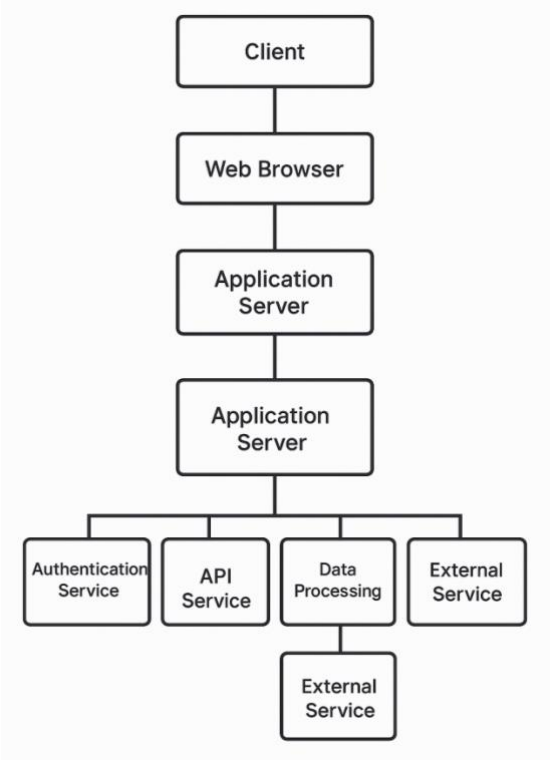


Figure 6.1.1 Architecture Diagram

6.1.2. Cloud-Based Hosting Infrastructure

The system is hosted on a cloud infrastructure like AWS, Google Cloud, or Microsoft Azure with the following:

- Docker containers for scalable deployments.
- Load balancers for performance tuning.
- API gateway for secure integration of modules.

6.2. Core System Modules and Components

6.2.1 User Management Module

- OTP-based login via mobile number.
- Role-based user profiles (Farmer, Buyer, Expert, Admin).
- Support for registration and profile setup in different languages.

6.2.2 Marketplace Module

- Farmers can add crops with fields like price, quantity, harvest date.
- Buyers can search and contact sellers location-wise.
- Real-time order management system integrated with it.

6.2.3 Advisory & Alert System

- Personalized crop advice depending on soil, region, and weather details.
- Push alerts for disease alerts, rainfall forecast, and price trends.
- Integrations with IMD and Agmarknet APIs for real-time data.

6.2.4 Input Access Module

- Farmers can search and order seeds, fertilizers, and equipment.
- Dealer interface to upload genuine products.
- Reviews and ratings for transparency.

6.2.5 Financial Services Interface

- Loan and subsidy portal.
- Facility for KCC, crop insurance, and PM-KISAN schemes applications.
- Financial literacy advice and eligibility calculators.

6.2.6 Admin and Analytics Dashboard

- Usage metrics and transaction monitoring.
- Tracking of most active users and geographies.
- Aggregation of feedback and complaint management.

6.3 User Interface (UI) and User Experience (UX) Design

6.3.1 Design Principles

- Simple, intuitive interface for ease of use.
- High-contrast color and icon support for visibility.
- Standard screen navigation patterns with minimal complexity.

6.3.2 Features for Accessibility

- Multiple language support (local languages as well).
- Voice inputs and audio instructions.
- Icon-based inputs for low-literacy users.

6.4 Data Flow Design

6.4.1 Input Layer

- User-manual inputs (text, voice, images).
- Sensor or GPS-based inputs depending on location and environment.

6.4.2 Processing Layer

- Buyer-seller matching algorithms.
- Decision trees and ML models to give personalized recommendations.

6.4.3 Output Layer

- Summary on dashboard, advisory messages, warnings, order summary.
- Record and export options for historical data.

6.5 Implementation Plan and Phases

6.5.1 Phase 1: Prototype Development

- MVP with login, crop list, and advisory.
- Field testing with small farmer groups to gather feedback.

6.5.2 Phase 2: Backend Integration

- Firebase database synchronization.
- Low-connectivity area offline storage using SQLite.

6.5.3 Phase 3: Testing and Feedback

- Pilot in a couple of districts.
- User feedback integration and iterative design fixes.

6.5.4 Phase 4: Full Feature Integration

- Integrate finance, input access, and expert advisory modules.
- Database configuration and analytics integration are complete.

6.5.5 Phase 5: Deployment

- Publish on Google Play Store with regional marketing.
- Onboarding training workshops and tutorial videos.

6.6 Security and Data Protection

6.6.1 Encryption

- All in-transit data encrypted (HTTPS) and data at rest.
- Sensitive data stored securely through Firebase rules or custom encryption.

6.6.2 Role-Based Access Control

- Special admin rights, farmer rights, and buyer rights.
- Restricted visibility per user role and action.

6.6.3 Legal Compliance

- GDPR-driven privacy paradigm.
- Consent management and user data deletion management.

CHAPTER-7
TIMELINE FOR EXECUTION OF PROJECT
(GANTT CHART)

7.1 Gantt Chart Breakdown

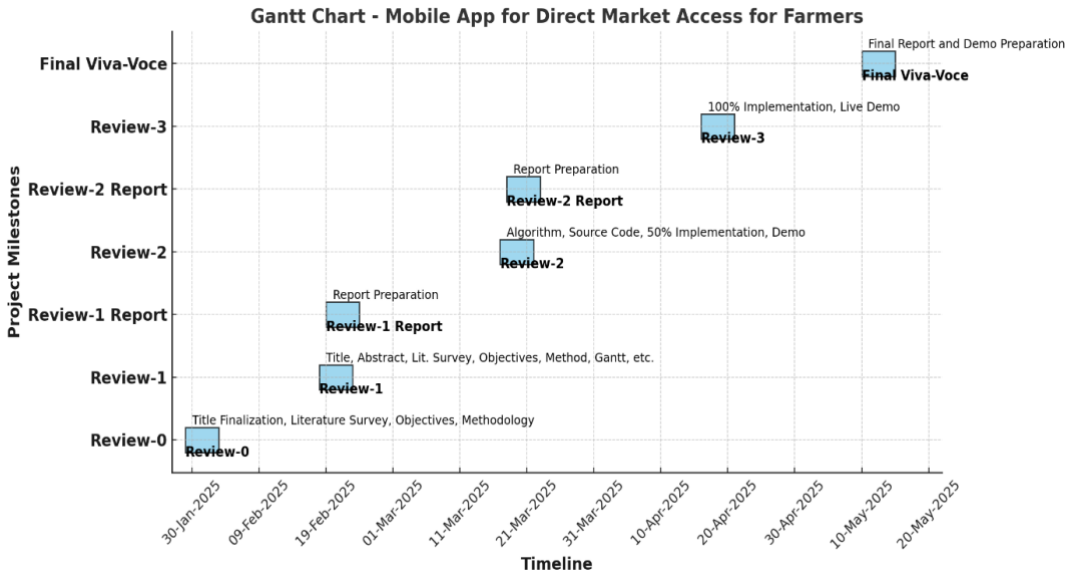


Figure 7.1.1 Gantt Chart

7.2 Key Milestones

7.2.1 Phase Completion Milestones

- Week 1: Requirement Analysis
- Week 2: System Design
- Week 4: UI/UX Design
- Week 6: Backend Development
- Week 7: Frontend Development
- Week 8: Database Integration
- Week 9: Testing & Debugging
- Week 10: Final Review & Documentation
- Week 11: Project Submission & Presentation

7.2.2 Critical Deliverables

Week	Deliverable	Details
Week 1	Requirement Specification Document	Detailed outline of app goals, User roles, and features.
Week 2	Design Mockups	User interface designs using tools like Figma or XML

		layouts.
Week 4	Functional Database & API Setup	Firebase or MySQL setup with RESTful API endpoints.
Week 6	Fully Functional App Prototype	Core features working with test data.
Week 8	Deployed Application	Working APk submitted or uploades for internal testing.
Week 11	Final Project Report & Presentation Slides	Complete report and visuals for project defene.

Table 7.2 Critical deliverables

CHAPTER-8
OUTCOMES

8.1 Overview of Outcomes

Development of the Farmers' Mobile App for Direct Market Access has produced a number

of worthwhile outcomes aligning with the project's fundamental objectives of empowering the farmer, simplifying agricultural trade, and utilizing digital technology in enhancing transparency. Not only does the app offer a virtual platform through which the farmer can engage directly with consumers and retailers, but it also increases their capacity to make informed decisions through access to real-time market information and streamlined transaction processes.

8.2 Technical Results

8.2.1 Successful Development with Android Studio

The mobile app was successfully developed and designed with the official IDE for Android application development, Android Studio. This provided a robust, responsive, and scalable app that can be easily updated and maintained in the future.

8.2.2 User-Friendly Interface

Among the top achievements was the design of an easy-to-use interface for users with little technical expertise. Navigation simplicity, visual simplicity, and the ability to support regional languages were given considerations while developing.

8.2.3 Core Functionalities Integration

These main features, which include listings, pricing, the communication of the sellers to buyers, and safety in making payment, were harmonized and undergone testing. Together, they had the aim to improve the usability of users to facilitate hassle-free transactions.

8.2.4 Backend Connectivity and Data Handling

Firebase was used for backend operations, offering real-time database functionality, user authentication, and cloud storage. This improved the performance and reliability of the app significantly.

8.3 Functional Outcomes

8.3.1 Direct Communication Between Farmers and Buyers

The application allows farmers and buyers to talk directly without the middlemen. This makes farmers earn more money while lowering expenses for buyers.

8.3.2 Real-Time Market Updates

Farmers get real-time access to prices of the market, buyers' demand, and stock availability,

thereby making correct choices regarding prices and inventory control.

8.3.3 Increased Transaction Transparency

Each sale is recorded and monitored, hence a system which is open for all, one that earns credibility between sellers and buyers. That openness plays an important role in avoiding exploitation as well as fairness in trading practices.

8.3.4 Cost and Time Efficiency

The mobile platform also mitigates the costs and time involved in conventional market practices by facilitating electronic transactions, remote product viewing, and online negotiations.

8.4 Societal Outcomes

8.4.1 Empowerment of Farmers

The greatest effect is the empowerment of farmers, particularly those in rural areas. Through market information and direct selling tools, farmers have greater control over their produce and earnings.

8.4.2 Encouragement of Digital Literacy

Moreover, as farmers utilize the app, they acquire basic digital skills that lead to opportunities to discover other digital platforms, services, and agricultural resources that can enhance their farming activities.

8.4.3 Community-Level Benefits

If used at a community level, the app can improve local agricultural economies, encourage sustainable agriculture, and minimize post-harvest losses by balancing supply and demand better.

CHAPTER-9
RESULTS AND DISCUSSIONS

The mobile application created for farmers was pilot tested at various levels, beginning from the simulation to field tests with small batches of farmers. The app was primarily designed to enhance farmers' access to markets, expert guidance, quality inputs, and government

schemes and decrease their dependence on middlemen and increase the transparency of the farming process. In pilot phase, the app was noted for its simplicity, usability, and ability to connect with the farmer community.

The best-case outcome was the users' uptake. A majority of farmers who used the trial could easily use the app on their own after a basic orientation, and this demonstrated that the user interface was user-friendly and responsive to different levels of digital proficiency among users. Such features as voice support and local language support made smartphone use easier for individuals with minimal or no experience so that they might readily access related information and services.

The marketplace feature, through which the farmers could get current prices and negotiate directly with the buyers, was much cherished. The majority reported getting a better price for their produce, even 15% more in some cases, compared to the previous system. It not only raised revenues but also brought in price transparency into local markets.

The crop advice module had an equally strong impact. With the guidance of their region, pest advisories, and weather, farmers were better equipped to make good decisions in their daily work. Better crop quality and reduced input costs, particularly through the application of pest advisories to prevent excessive use of pesticides, were some of the benefits that most enjoyed. The main purchase module for inputs facilitated the matching of buyers with genuine sellers, minimizing the risk of purchasing counterfeit or substandard fertilizers and seeds.

Adding information about government schemes further enhanced the utility of the app. Farmers appreciated the reminders and eligibility details of loans, subsidies, and insurance on crops. It was the first time most of them could make sense of these schemes without having to go to government offices or depending upon third-party agents.

Overall, the app was well on technical and user interface fronts. There was a reference to performance lagging on outdated gear and network connectivity issues being encountered sporadically in rural areas but not affecting consumer satisfaction to any large extent. It is possible to imagine from the findings that well-designed and maintained mobile applications had the potential to be a significant driver of making rural farmers increase productivity and make inclusive digital agriculture transformation a reality.

CHAPTER-10
CONCLUSION

The development of the mobile app for direct access by the farmers is a step in the direction of agriculture digitalization. With one and easy-to-use interface,

the project aimed to address the prevailing challenges for marginal and small farmers such as poor market access, uncertain sources of farm inputs, poor access to timely expert guidance, and weak knowledge regarding government schemes. It was developed to empower the farmers by equipping them with real-time information, direct interaction with buyers and suppliers, and location- and season-based crop advisory.

Throughout the project, the focus was on delivering a usable, intuitive solution. The app's interface was specifically designed to include support for users with varying degrees of digital literacy, including regional language support, voice accessibility, and offline usage. Field test and response showed that the application boosted farmers' confidence and independence in farm and marketing decisions to a great extent. The majority of the users enjoyed better market prices, easier access to certified inputs, and prompt responses to crop issues through expert reminders and weather information.

While the app achieved its fundamental goals, there were some problems which were encountered, such as rural connectivity problems and resistance on the part of older farmers to adopt digital innovation. These were, however, offset to a significant degree through sensitization workshops among users and outreach efforts at community levels. Overall, the project has proven that technology, if created with empathy and inclusiveness, can be beneficial for the agricultural system. This smartphone app is not just an imaginary utility but a gateway between farmers and prospects—pioneering more sustainable, efficient, and profitable agriculture in the future.

REFERENCES

[1] Chang, H.-H., & Meyerhoefer, C.D. (2020). COVID-19 and the demand for online food shopping services: Empirical evidence from Taiwan. American Journal of Agricultural

Economics, 103(2), 448–465. 2020– "Android App to Connect Farmers to Retailers and Food Processing Industry." IEEE Xplore. Available at: https://ieeexplore.ieee.org/document/9034434.

[2] IEEE Xplore. (2020). Android App to Connect Farmers to Retailers and Food Processing Industry. Available at: https://ieeexplore.ieee.org/document/9034434.

[3] Saxena, S., & Limbad, A. (2021). Consumption change of household food habits pre and post lockdown during COVID-19: A perspective study of Gujarat and Maharashtra. Global Journal of Interdisciplinary Studies, 4(1), 16–36.

[4] Zhao, L., Zhang, Y., & Zhang, H. (2022). Research on the impact of digital literacy on farmer households' green cooking energy consumption: Evidence from rural China. International Journal of Environmental Research and Public Health, 19(13464).

[5] Manocha, S., Bhullar, P.S., & Sachdeva, T. (2023). Factors determining the investment behaviour of farmers—The moderating role of socioeconomic demographics. Journal of Indian Business Research.

[6] IEEE Xplore. (2023). Smart Kisan: A Mobile App for Farmers' Assistance in Agricultural Activities. Available at: https://ieeexplore.ieee.org/document/10199471.

[7] Hinojosa, C., Sanchez, K., Camacho, A., & Arguello, H. (2023). AgroTIC: Bridging the gap between farmers, agronomists, and merchants through smartphones and machine learning. arXiv preprint, arXiv:2305.12418. Available at: https://arxiv.org/abs/2305.12418.

[8] Kumar, R. (2023). Farmers' use of the mobile phone for accessing agricultural information in Haryana: An analytical study. Open Information Science. Available at: https://www.degruyter.com/document/doi/10.1515/opis-2023-0031/html.

[9] Agriculture & Food Security. (2023). Is agricultural digitization a reality among smallholder farmers in Sub-Saharan Africa? Unpacking the digital agricultural paradox.

[10] Mdoda, L., Christian, M., & Agbugba, I. (2024). Use of information systems (mobile phone app) for enhancing smallholder farmers' productivity in Eastern Cape Province, South Africa: Implications on food security. Journal of the Knowledge Economy, 15, 1993–2009. Available at: https://link.springer.com/article/10.1007/s13132-024-01478-x.

APPENDIX-A

PSUEDOCODE

A.1 AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.example.pickfresh">

    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.SEND_SMS"/>
    <application
        android:allowBackup="true"
        android:allowClearUserData="true"
        android:allowClearUserDataOnFailedRestore="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@drawable/fresh"
        android:label="@string/app_name"
        android:roundIcon="@drawable/fresh"
        android:supportsRtl="true"
        android:theme="@style/Theme.PickFresh"
        android:usesCleartextTraffic="true"
        tools:targetApi="31">
        <activity
            android:name=".Buyer.ProflieActivity"
            android:exported="false" />
        <activity
            android:name=".Buyer.CompletedActivity"
            android:exported="false" />
        <activity
            android:name=".Seller.Userdeatils"
            android:exported="false" />
        <activity
            android:name=".Seller.ViewAccepted"
            android:exported="false" />
        <activity
            android:name=".Seller.ViewBuyerItems"
            android:exported="false" />
        <activity
            android:name=".Seller.RequestsActivity"
            android:exported="false" />
        <activity
            android:name=".Seller.SettingsActivity"
            android:exported="false" />
        <activity
```

```
    android:name=".Buyer.ViewOrderitems"
    android:exported="false" />
<activity
    android:name=".Buyer.View_Pendings"
    android:exported="false" />
<activity
    android:name=".Buyer.ViewItems"
    android:exported="false" />

<meta-data
    android:name="com.google.android.geo.API_KEY"
    android:value="@string/apikey" />

<activity
    android:name=".Buyer.MapsActivity"
    android:exported="false"
    android:label="@string/title_activity_maps" />
<activity
    android:name=".Seller.OrderupdateActivity"
    android:exported="false" />
<activity
    android:name=".Seller.Viewitems"
    android:exported="false" />
<activity
    android:name=".Seller.SchemesActivity"
    android:exported="false" />
<activity
    android:name=".Seller.WebActivity"
    android:exported="false" />

<activity
    android:name=".Buyer.SettingsForUser"
    android:exported="false" />
<activity
    android:name=".Seller.Additems"
    android:exported="false" />
<activity
    android:name=".Admin.AddSeller"
    android:exported="false" />
<activity
    android:name=".Buyer.BuyerMainActivity"
    android:exported="false" />
<activity
    android:name=".Seller.SellerMainActivity"
    android:exported="false" />
<activity
    android:name=".Admin.AdminActivity"
    android:exported="false"
    android:theme="@style/mytheme" />
<activity
    android:name=".LoginActivity"
    android:exported="false" />
<activity
    android:name=".Signup"
    android:exported="false" />
```

```
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <receiver android:name=".Buyer.BroadcastReceiver"
            android:exported="true"
            android:enabled="true"
            tools:ignore="ExportedReceiver" />
    </application>

</manifest>
```

A2. Kotlin+java (com.example.pickfresh)

Admin

(AdapterForSellerList)

```
package com.example.pickfresh.Admin


import android.content.Context
import android.net.Uri
import android.text.Html
import android.text.Layout
import android.view.ContextMenu
import android.view.LayoutInflater
import android.view.ViewGroup
import androidx.core.text.HtmlCompat
import androidx.recyclerview.widget.RecyclerView
import coil.load
import com.example.pickfresh.Model.User
import com.example.pickfresh.databinding.ActivitySellerMainBinding


class AdapterForSellerList(val context: Context,val data:
ArrayList<User>):RecyclerView.Adapter<AdapterForSellerList.Viewed>() {
    class Viewed(val item :ActivitySellerMainBinding):RecyclerView.ViewHolder(item.root)

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int)=
```

```
    Viewed(ActivitySellerMainBinding.inflate(LayoutInflater.from(context),parent,false))

  override fun onBindViewHolder(holder: Viewed, position: Int) {
val k=data[position]
    with(holder.item){
  val string="<b>Name : </b>${k.name}<br></br>" +
        "<b>Joined in:</b>${k.date}"
  details.text=HtmlCompat.fromHtml(string, HtmlCompat.FROM_HTML_MODE_LEGACY)
        shapeimage.load(Uri.parse(k.image))
}
  }

  override fun getItemCount()=data.size

}
```

Buyer

(AdapterForPendings)

```
package com.example.pickfresh.Buyer

import android.content.Context
import android.content.Intent
import android.view.LayoutInflater
import android.view.ViewGroup
import androidx.recyclerview.widget.RecyclerView
import com.example.pickfresh.Model.Orderid
import com.example.pickfresh.Seller.ViewAccepted
import com.example.pickfresh.Seller.ViewBuyerItems
import com.example.pickfresh.databinding.IdviewBinding

class AdapterForPendings(val context:Context,val  data: ArrayList<Orderid>,var string:String) :
RecyclerView.Adapter<AdapterForPendings.Holder>() {
  class Holder(val item :IdviewBinding):RecyclerView.ViewHolder(item.root)

  override fun onCreateViewHolder(parent: ViewGroup, viewType: Int)=
Holder(IdviewBinding.inflate(LayoutInflater.from(context),parent,false))
```

```
    override fun onBindViewHolder(holder: Holder, position: Int) {
val id=data[position]
        with(holder.item){
    details5.text=id.orderid
}
        holder.itemView.setOnClickListener {
            if(string=="seller") {
                Intent(context, ViewBuyerItems::class.java).apply {
                    putExtra("id",id.orderid)
                    context.startActivity(this)
                }
            } else if(string=="buyer"){
                Intent(context, ViewOrderitems::class.java).apply {
                    putExtra("id", id.orderid)
                    context.startActivity(this)
                }
            }else if(string=="completed"){
                Intent(context, CompletedActivity::class.java).apply {
                    putExtra("id", id.orderid)
                    context.startActivity(this)
                }
            }
            else if(string=="accepted"){
Intent(context, ViewAccepted::class.java).apply {
    putExtra("id",id.orderid)
    context.startActivity(this)
}
            }

        }
    }

    override fun getItemCount()=data.size

}
```

Model

(Dummy)

```
package com.example.pickfresh.Model

data class Dummy (var one:String,
var second:String)
```

Responses

(Api)

```
package com.example.pickfresh.Responses



import retrofit2.Call
import retrofit2.Response
import retrofit2.http.Field
import retrofit2.http.FormUrlEncoded
import retrofit2.http.GET
import retrofit2.http.POST
import retrofit2.http.Query

interface Api {
    @FormUrlEncoded
    @POST("user.php")
    fun signup(
        @Field("name") name: String,
        @Field("email") email: String,
        @Field("mobile") mobile: String,
        @Field("password") password: String,
        @Field("location") location: String,
    ): Call<CommonReponse>

    @FormUrlEncoded
    @POST("user.php")
    fun login(
```

```
    @Field("condition") condition: String,
    @Field("email") email: String,
    @Field("password") password: String,
): Call<LoginResponse>

@FormUrlEncoded
@POST("addseller.php")
fun addseller(
    @Field("name") name: String,
    @Field("email") email: String,
    @Field("password") password: String,
    @Field("mobilenumber") mobilenumber: String,
    @Field("encoded") encoded: String,
    @Field("date") date: String,
    @Field("location") location: String,
): Call<CommonReponse>

@GET("getdata.php")
fun getSellers(): Call<LoginResponse>

@FormUrlEncoded
@POST("additems.php")
fun additems(
    @Field("itemname") itemname: String,
    @Field("sellerid") sellerid: String,
    @Field("itemphoto") itemphoto: String,
    @Field("price") price: String,
    @Field("type") type: String,

    ): Call<CommonReponse>

@FormUrlEncoded
@POST("getdata.php")
fun getitems(
    @Field("condition") condition: String,
    @Field("id") id: String,
): Call<ProductResponse>
```

```
@FormUrlEncoded
@POST("updatefun.php")
fun update(
    @Field("id") id: String,
    @Field("state") state: String,
    @Field("cost") cost: String,
): Call<CommonReponse>

@FormUrlEncoded
@POST("updatefun.php")
fun stateupdate(
    @Field("id") id: String,
    @Field("condition") condition: String,
    @Field("state") state: String,
): Call<CommonReponse>

@FormUrlEncoded
@POST("getdata.php")
fun getuser(
    @Field("id") id: String,
    @Field("condition") condition: String,
): Call<LoginResponse>

@FormUrlEncoded
@POST("getdata.php")
fun getData(
    @Field("condition") condition: String,
    @Field("search") search: String,
): Call<LoginResponse>

@FormUrlEncoded
@POST("getdata.php")
fun getitemsbyid(
    @Field("id") id: String,
    @Field("condition") condition: String,
): Call<GoodResponse>
```

```
@FormUrlEncoded
@POST("insertdata.php")
fun orderitem(

    @Field("userid") userid: String,
    @Field("orderid") orderid: String,
    @Field("sellerid") sellerid: String,
    @Field("status") status: String,
    @Field("itemphoto") itemphoto: String,
    @Field("itemname") itemname: String,
    @Field("qty") qty: String,
    @Field("price") price: String,
    @Field("date") date: String,
): Call<CommonReponse>

@FormUrlEncoded
@POST("getdata.php")
fun getpendings(
    @Field("condition") condition: String,
    @Field("id") id: String,
): Call<OrderResponseonlyid>

@FormUrlEncoded
@POST("getdata.php")
fun getorderitems(
    @Field("condition") condition: String,
    @Field("id") id: String,
): Call<OrderResponse>

@FormUrlEncoded
@POST("getdata.php")
fun getorderbyseing(
    @Field("condition") condition: String,
    @Field("id") id: String,
): Call<OrderResponseonlyid>
```

```
@FormUrlEncoded
@POST("getdata.php")
fun getbuyerdetails(
    @Field("id") id: String,
    @Field("condition") condition: String,
): Call<OrderResponse>

@FormUrlEncoded
@POST("updatefun.php")
fun updaterequest(
    @Field("condition") condition: String,
    @Field("id") id: String,
    @Field("state") state: String,
): Call<CommonReponse>

@FormUrlEncoded
@POST("getdata.php")
fun getorderview(
    @Field("condition") condition: String,
    @Field("id") id: String,
): Call<OrderResponseonlyid>

@FormUrlEncoded
@POST("getdata.php")
fun getuserdeatils(
    @Field("condition") condition: String,
    @Field("id") id: String,
): Call<LoginResponse>

@FormUrlEncoded
@POST("updatefun.php")
fun updatefun(
    @Field("condition") condition: String,
    @Field("id") id: String,
    @Field("state") state: String,
): Call<CommonReponse>
```

```
@FormUrlEncoded
@POST("getdata.php")
fun getaccepted(
    @Field("condition") condition: String,
    @Field("id") id: String,
): Call<OrderResponseonlyid>

@FormUrlEncoded
@POST("getdata.php")
fun getorderdetails(
    @Field("condition") condition: String,
    @Field("id") id: String,
): Call<CustomeResponse>

@FormUrlEncoded
@POST("orderreview.php")
fun addreview(
    @Field("rating") rating: String,
    @Field("review") review: String,
    @Field("buyerid") buyerid: String,
    @Field("productid") productid: String,
): Call<CommonReponse>

@FormUrlEncoded
@POST("ratings.php")
fun getrating(
    @Field("id") id: String,
): Call<Reviewresponse>

@FormUrlEncoded
@POST("getsmsdata.php")
fun getmobilenumber(
    @Field("id") id: String,
): Call<MessagingResponse>

@FormUrlEncoded
@POST("getuser.php")
```

```
    fun getuser(
        @Field("id") id: String,
    ): Call<LoginResponse>

    @FormUrlEncoded
    @POST("update.php")
    fun updateorder(@Field("id") id: String): Call<CommonReponse>

    @GET("updateLocations.php")
    suspend fun updateLocations(
        @Query("id") id: String,
        @Query("location") location: String,
        @Query("address") address: String,
    ): Response<CommonReponse>


}
```

Seller

(AdapterForViewItems)

```
package com.example.pickfresh.Seller

import android.annotation.SuppressLint
import android.content.Context
import android.content.Intent
import android.net.Uri
import android.os.Bundle
import android.view.LayoutInflater
import android.view.ViewGroup
import androidx.core.text.HtmlCompat
import androidx.recyclerview.widget.RecyclerView
import coil.load
import com.example.pickfresh.Model.Items
import com.example.pickfresh.databinding.ViewitemsBinding

class AdapterForViewItems(val context: Context, val  data: ArrayList<Items>, val string:String) :
```

```
RecyclerView.Adapter<AdapterForViewItems.Holder>() {
   class Holder(val item:ViewitemsBinding):RecyclerView.ViewHolder(item.root)

   override fun onCreateViewHolder(parent:ViewGroup, viewType: Int)
   =Holder(ViewitemsBinding.inflate(LayoutInflater.from(context),parent,false))
   @SuppressLint("SetTextI18n")
   override fun onBindViewHolder(holder: Holder, position: Int) {
val k=data[position]
      with(holder.item){
   shapable.load(Uri.parse(k.itemphoto))

         if("not change"==string){
            details2.text= HtmlCompat.fromHtml("<b>Name : </b>${k.itemname}<br></br><b>Price
: </b>₹${k.price}/-<br></br><b>Status : </b>${k.status}<br></br>"
            ,HtmlCompat.FROM_HTML_MODE_LEGACY)
            k.itemname="Name : ${k.itemname}"
            k.price="Price :₹${k.price}/-"
            k.status="Status :${k.status}"
         }else{
            details2.text="${k.itemname}\n${k.price}\n${k.status}"
            }
      }
      holder.itemView.setOnClickListener {
         Intent(context,OrderupdateActivity::class.java).apply {
            putExtra("data",k)
            context.startActivity(this)
         }
      }
   }

   override fun getItemCount()=data.size

}
```

Functions.kt

(LoginActivity)

```
package com.example.pickfresh

import android.annotation.SuppressLint
import android.app.*
import android.content.Intent
import android.graphics.Color
import android.graphics.drawable.ColorDrawable
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.View
import android.widget.AdapterView
import android.widget.ArrayAdapter
import android.widget.Toast
import androidx.lifecycle.ViewModelProvider
import com.example.pickfresh.Admin.AddSeller
import com.example.pickfresh.Admin.AdminActivity
import com.example.pickfresh.Buyer.BuyerMainActivity
import com.example.pickfresh.Model.Onewordchange
import com.example.pickfresh.Responses.LoginResponse
import com.example.pickfresh.Responses.Retrofit
import com.example.pickfresh.Seller.SellerMainActivity
import com.example.pickfresh.databinding.ActivityLoginBinding
import com.google.mlkit.common.model.DownloadConditions
import com.google.mlkit.nl.translate.TranslateLanguage
import com.google.mlkit.nl.translate.Translation
import com.google.mlkit.nl.translate.TranslatorOptions
import kotlinx.coroutines.CoroutineScope
import kotlinx.coroutines.Dispatchers.IO
import kotlinx.coroutines.launch
import retrofit2.Call
import retrofit2.Callback
import retrofit2.Response

class LoginActivity : AppCompatActivity() {
   private lateinit var bind:ActivityLoginBinding
```

```
lateinit var dialog:Dialog
lateinit var onewordchange: Onewordchange
var realString=ArrayList<String>()
var kk=arrayOf("English","Tamil","Telugu","Kannada","Hindi")
   @SuppressLint("UnspecifiedImmutableFlag")
   override fun onCreate(savedInstanceState: Bundle?) {
      super.onCreate(savedInstanceState)
      bind=ActivityLoginBinding.inflate(layoutInflater)
      setContentView(bind.root)
      onewordchange=ViewModelProvider(this)[Onewordchange::class.java]
      realString.add("${bind.titlew.text}")
      realString.add("${bind.email.text}")
      realString.add("${bind.email2.hint}")
      realString.add("${bind.password.text}")
      realString.add("${bind.password2.hint}")
      realString.add("${bind.btn.text}")
      realString.add("${bind.dont.text}")
      realString.add("${bind.create.text}")
      realString.add("${bind.create2.text}")
      realString.add("${bind.dont2.text}")
      dialog=Dialog(this).apply {
         setContentView(R.layout.progressdi)
         setCancelable(false)
         window!!.setBackgroundDrawable(ColorDrawable(Color.TRANSPARENT))
      }
bind.create2.setOnClickListener {
startActivity(Intent(this, AddSeller::class.java))

}

bind.btn.setOnClickListener {

   val mail=bind.email2.text.toString().trim()
   val password=bind.password2.text.toString().trim()

 if(!mail.contains("@gmail.com")) {
it.toast("Please enter a valid email")
```

```
    }else if(password.isEmpty()){
        it.toast("Please enter your password")
    }else if(password.lowercase()=="admin"&&mail.lowercase()=="admin@gmail.com") {
        getSharedPreferences("user", MODE_PRIVATE).edit().putString("type","admin").apply()
        startActivity(Intent(this,AdminActivity::class.java))
    finishAffinity()
        }else{
    dialog.show()
        CoroutineScope(IO).launch {
            Retrofit.instance.login(condition = "login",email = mail, password=password).enqueue(object
:Callback<LoginResponse>{
                override fun onResponse(
                    call: Call<LoginResponse>,
                    response: Response<LoginResponse>
                ) {
                    dialog.dismiss()
                    response.body().apply {
                        if(this!=null) {
                            if (data.isNotEmpty()) {
                                val k = data[0]
                                getSharedPreferences("user", MODE_PRIVATE).edit().apply {
                                    putString("id", k.id)
                                    putString("name", k.name)
                                    putString("mail", k.mail)
                                    putString("mobile", k.mobile)
                                    putString("password", k.password)
                                    putString("location", k.location)
                                    putString("type", k.type)
                                    putString("state", k.state)
                                    putString("language","English")
                                    apply()
                                }
                                finishAffinity()
                                if (k.type == "user") {
                                    startActivity(
                                        Intent(
                                            this@LoginActivity,
```

```
                        BuyerMainActivity::class.java
                    )
                )
            } else if (k.type == "seller") {
                startActivity(
                    Intent(
                        this@LoginActivity,
                        SellerMainActivity::class.java
                    )
                )
            }

        } else {
            if (message == "failed") {
                it.toast("Invalid user")
            }
        }
    }else{
        it.toast(response.body()!!)
    }
  }

}

override fun onFailure(call: Call<LoginResponse>, t: Throwable) {
it.toast(t.message!!)
   dialog.dismiss()
}
  })
 }
}
}
```

```
bind.create.setOnClickListener {
Intent(this,Signup::class.java).apply {
```

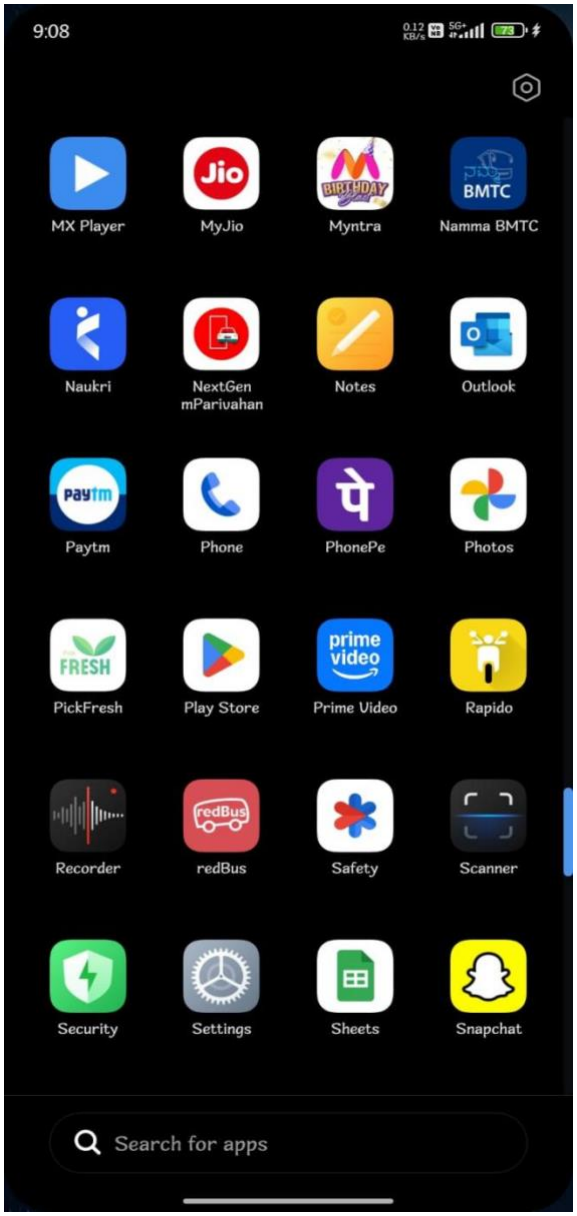
```
            putExtra("language",bind.spinner.selectedItem.toString())
            startActivity(this)
        }
        }

ArrayAdapter(this,android.R.layout.simple_dropdown_item_1line,kk)
    .apply {
        bind.spinner.adapter=this
    }

        bind.spinner.onItemSelectedListener=object :AdapterView.OnItemSelectedListener{
            override fun onItemSelected(p0: AdapterView<*>?, p1: View?, p2: Int, p3: Long) {
if(kk[p2]=="Tamil"){
    translate(TranslateLanguage.TAMIL)
}else if(kk[p2]=="English"){
    transulated(realString)
}else if(kk[p2]=="Telugu"){
    translate(TranslateLanguage.TELUGU)
}else if(kk[p2]=="Kannada"){
    translate(TranslateLanguage.KANNADA)
}else if(kk[p2]=="Hindi"){
    translate(TranslateLanguage.HINDI)
}
            }
            override fun onNothingSelected(p0: AdapterView<*>?) {
            }
        }
    }
    private fun transulated(it: ArrayList<String>) {
        if(realString.size==it.size){
            bind.titlew.text=it[0]
            bind.email.text=it[1]
            bind.email2.hint=it[2]
            bind.password.text=it[3]
            bind.password2.hint=it[4]
            bind.btn.text=it[5]
            bind.dont.text=it[6]
```

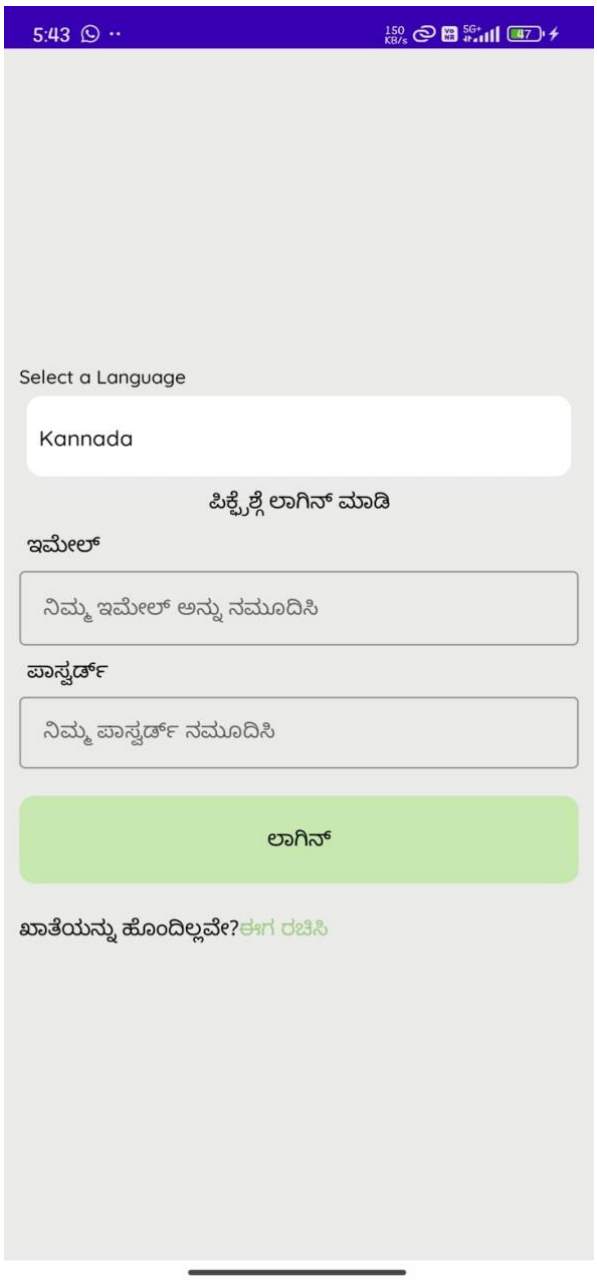
```
            bind.create.text=it[7]
            bind.create2.text=it[8]
            bind.dont2.text=it[9]

        }
        bind.titlew.textSize=16f
        bind.email.textSize=16f
        bind.email2.textSize=16f
        bind.password.textSize=16f
        bind.password2.textSize=16f
        bind.btn.textSize=16f
        bind.dont.textSize=16f
        bind.create.textSize=16f
    }
    private fun translate(langauage: String) {

        if(TranslateLanguage.TAMIL==langauage){
            bind.titlew.textSize=14f
            bind.email.textSize=14f
            bind.email2.textSize=14f
            bind.password.textSize=14f
            bind.password2.textSize=14f
            bind.btn.textSize=14f
            bind.dont.textSize=14f
            bind.create.textSize=14f
            bind.create2.textSize=14f
            bind.dont2.textSize=14f
        }else{
            bind.titlew.textSize=16f
            bind.email.textSize=16f
            bind.email2.textSize=16f
            bind.password.textSize=16f
            bind.password2.textSize=16f
            bind.btn.textSize=16f
            bind.dont.textSize=16f
            bind.create.textSize=16f
            bind.create2.textSize=16f
```

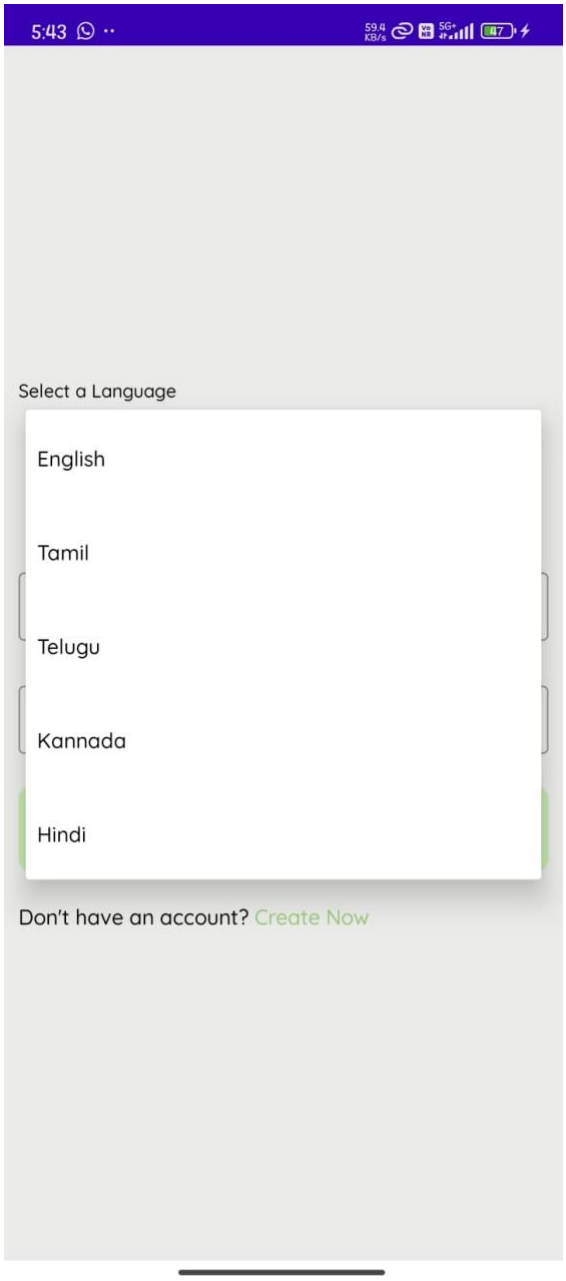
```
            bind.dont2.textSize=16f
        }
        dialog.show()
val option= TranslatorOptions.Builder()
    .setSourceLanguage(TranslateLanguage.ENGLISH)
    .setTargetLanguage(langauage)
    .build()
        val condition=DownloadConditions.Builder().build()
        val translation= Translation.getClient(option)
        translation.downloadModelIfNeeded(condition)
            .addOnSuccessListener {
                onewordchange.option(option, realString)
                onewordchange.observer().observe(this){
                    if(realString.size==it.size){
                        bind.titlew.text=it[0]
                        bind.email.text=it[1]
                        bind.email2.hint=it[2]
                        bind.password.text=it[3]
                        bind.password2.hint=it[4]
                        bind.btn.text=it[5]
                        bind.dont.text=it[6]
                        bind.create.text=it[7]
                        bind.create2.text=it[7]
                        bind.dont2.text=it[7]
                    }
                }
                dialog.dismiss()
            }
            .addOnFailureListener {
                Toast.makeText(this, "1->${it.message}", Toast.LENGTH_SHORT).show()
                dialog.dismiss()
            }
        }


    }
```

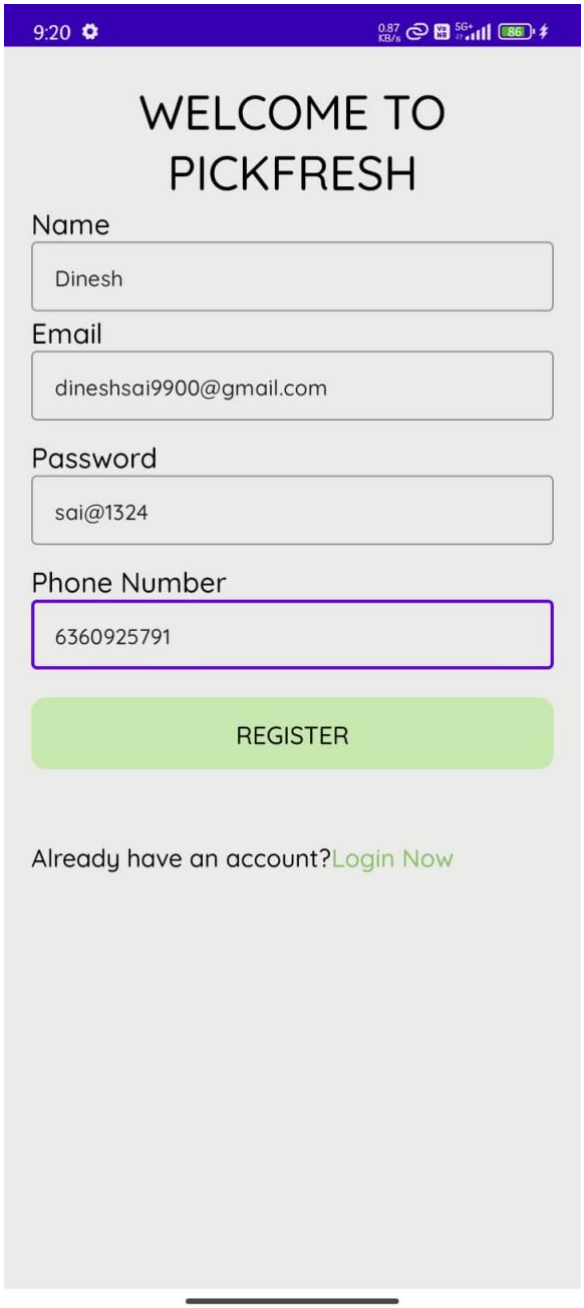
APPENDIX-B

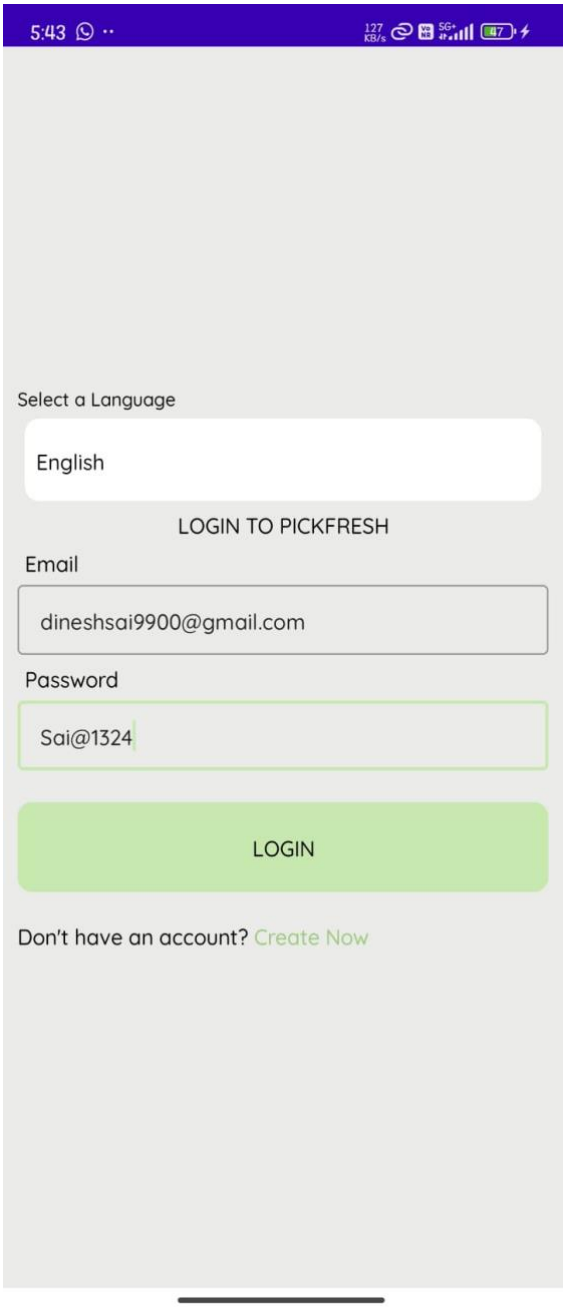
SCREENSHOTS

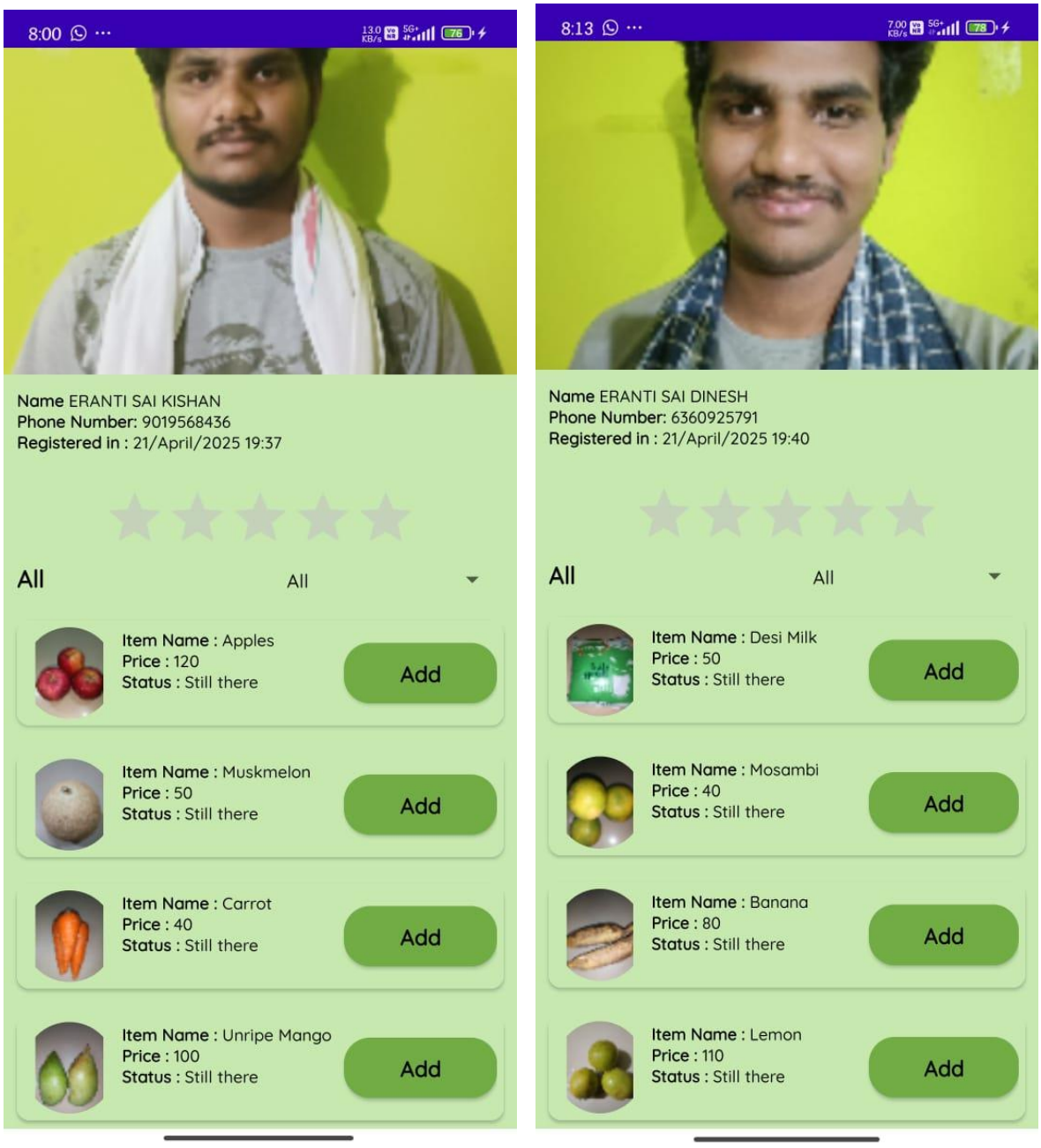


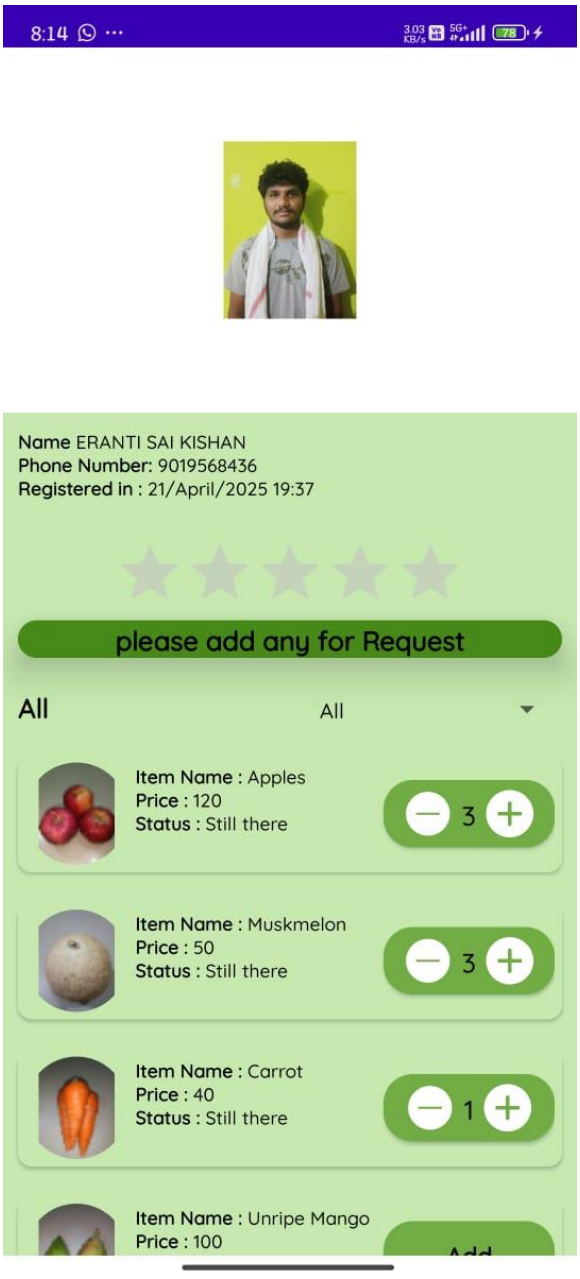

5:43
Select a Language
Kannada
ಪಿಕ್ಚೆಶ್ಗೆ ಲಾಗಿನ್ ಮಾಡಿ
ಇಮೇಲ್
ನಿಮ್ಮ ಇಮೇಲ್ ಅನ್ನು ನಮೂದಿಸಿ
ಪಾಸ್ವರ್ಡ್
ನಿಮ್ಮ ಪಾಸ್ವರ್ಡ್ ನಮೂದಿಸಿ
ಲಾಗಿನ್
ಖಾತೆಯನ್ನು ಹೊಂದಿಲ್ಲವೇ?ಈಗ ರಚಿಸಿ

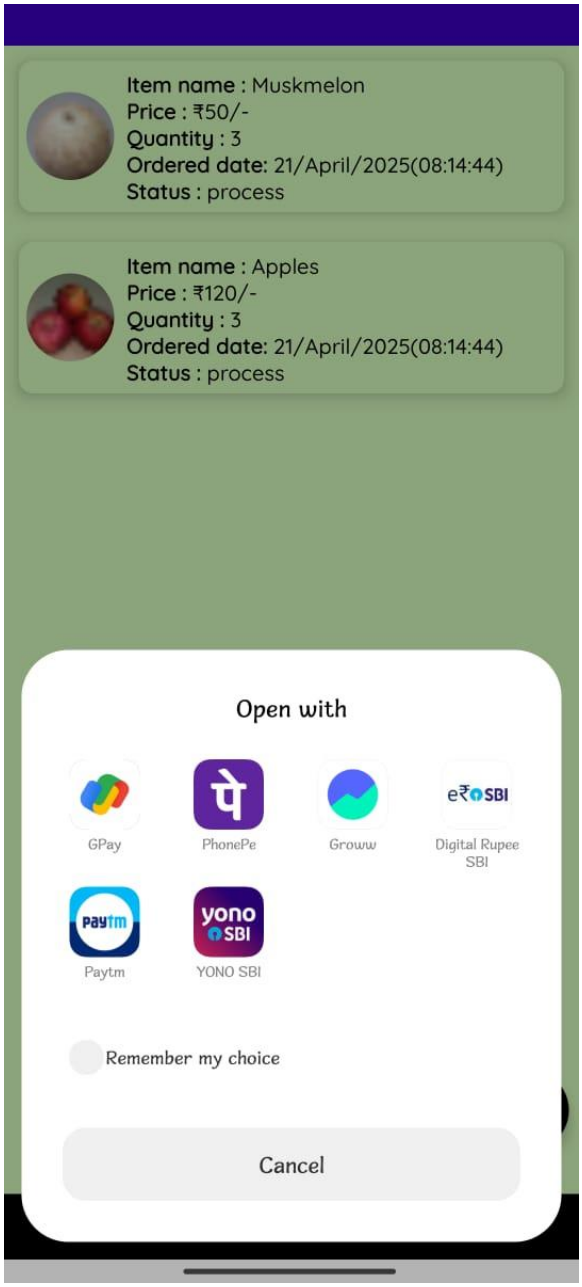

5:43
Select a Language
English
Tamil
Telugu
Kannada
Hindi
Don't have an account? Create Now


9:20
WELCOME TO PICKFRESH
Name
Dinesh
Email
dineshsai9900@gmail.com
Password
sai@1324
Phone Number
6360925791
REGISTER
Already have an account?Login Now


5:43
Select a Language
English
LOGIN TO PICKFRESH
Email
dineshsai9900@gmail.com
Password
Sai@1324
LOGIN
Don't have an account? Create Now


8:00
Name ERANTI SAI KISHAN
Phone Number: 9019568436
Registered in : 21/April/2025 19:37
All
All
Item Name : Apples
Price : 120
Status : Still there
Add
Item Name : Muskmelon
Price : 50
Status : Still there
Add
Item Name : Carrot
Price : 40
Status : Still there
Add
Item Name : Unripe Mango
Price : 100
Status : Still there
Add
8:13
Name ERANTI SAI DINESH
Phone Number: 6360925791
Registered in : 21/April/2025 19:40
All
All
Item Name : Desi Milk
Price : 50
Status : Still there
Add
Item Name : Mosambi
Price : 40
Status : Still there
Add
Item Name : Banana
Price : 80
Status : Still there
Add
Item Name : Lemon
Price : 110
Status : Still there
Add


8:14
Name ERANTI SAI KISHAN
Phone Number: 9019568436
Registered in : 21/April/2025 19:37
please add any for Request
All
All
Item Name : Apples
Price : 120
Status : Still there
3
Item Name : Muskmelon
Price : 50
Status : Still there
3
Item Name : Carrot
Price : 40
Status : Still there
1
Item Name : Unripe Mango
Price : 100


Item name : Muskmelon
Price : ₹50/-
Quantity : 3
Ordered date: 21/April/2025(08:14:44)
Status : process
Item name : Apples
Price : ₹120/-
Quantity : 3
Ordered date: 21/April/2025(08:14:44)
Status : process
Open with
GPay
PhonePe
Groww
Digital Rupee SBI
Paytm
YONO SBI
Remember my choice
Cancel


8:15
E
Paying ERANTI SAI KISHAN
Banking name: Eranti Sai Kishan
PhonePe • UPI ID: 9019568436@ybl
₹510.00
DummyPayment
Choose account to pay with
State Bank of India ••••8989
Balance: Check now
Pay ₹510
AXIS BANK
UPI

APPENDIX-C

ENCLOSURES

© April 2025 | IJIRT | Volume 11 Issue 11 | ISSN: 2349-6002

Mobile App for Direct Market Access for Farmers

Eranti Sai Kishan[1], Eranti Sai Dinesh[2], Lakki Reddy Varshitha[3], Pavani M[4], Battula Bhavya[5]
[1,2,3,4] *Department of CSE, SoE, Presidency university, Yelahanka, Bengaluru,KA.*
[5]*Battula Bhavya, Assistant Professor, Presidency university, Yelahanka, Bengaluru,KA.*

***Abstract*—Agriculture is the backbone of most economies, but farmers tend to face restricted market access, thin profit margins, and reliance on intermediaries. This study introduces the creation of a Mobile App for Direct Market Access for Farmers, developed with Android Studio to close the gap between farmers and consumers. The suggested application empowers farmers by facilitating direct sales, cutting out middlemen, and guaranteeing fair prices for their produce. The application combines capabilities like live pricing, order management, secure payment processors, GPS-enabled buyer-seller mapping, and multi-language support, providing accessibility to various users. System architecture relies on Firebase for real-time database management and RESTful APIs for easy communication between buyers and sellers. By using this mobile app, farmers can post their produce, bargain over price, and finish transactions with ease, generating better financial returns and minimizing post-harvest losses. Pilot testing and user reaction suggest that the application increases market efficiency and farmer incomes. The study documents the potential of mobile technology to revolutionize the agricultural supply chain, facilitate sustainability, and contribute to rural economic development.**

***Index Terms*—Mobile App, Direct Market Access, Farmers, Android Studio, E-commerce, Digital Marketplace, Agriculture.**

I. INTRODUCTION

Agriculture is an important sector in the world economy, but farmers usually struggle to get a fair market for their crops. Conventional agricultural value chains have numerous intermediaries, which result in lower returns for farmers and higher prices for consumers. To fill this gap, technology-based solutions like mobile apps can empower farmers by giving them direct access to markets, cutting out middlemen, and ensuring improved price realization.

This study aims to develop a Mobile App for Direct Market Access for Farmers based on Android Studio, which allows farmers to directly interact with buyers, wholesalers, and retailers. The app supports real-time price discovery, easy transaction management, and direct communication between farmers and consumers. Through the utilization of new mobile technologies, this solution targets increased efficiency, transparency, and profitability in the supply chain of agriculture.

The envisioned mobile app incorporates product listing, real-time price updates, geolocation-based market access, secure payment gateways, and a feedback system to promote trust among users. It provides ease of use with a simple yet robust user interface that is specifically designed for farmers with low digital literacy. Additionally, the app caters to regional languages to increase accessibility and adoption.

This research paper outlines the design, development, and deployment of the mobile application, emphasizing the major technologies employed, challenges faced, and the potential effect on agricultural trade. By offering farmers a direct market access tool, this solution has the potential to revolutionize the agricultural industry, promoting financial growth and sustainability for rural communities.

II. LITRATURE REVIEW

The evolution of mobile technology has greatly impacted the agricultural industry, with farmers being offered improved market access, price information, and trading opportunities. Mobile applications have been studied as a means of enhancing agricultural marketing and minimizing intermediary dependence. Direct market access solutions using mobile bases enable farmers to be empowered as they are brought directly into contact with buyers without the involvement of intermediaries, guaranteeing fairness in prices as well as transparent transactions. It has

© April 2025 | IJIRT | Volume 11 Issue 11 | ISSN: 2349-6002

been found through research that conventional market structures tend to create inefficiencies, wherein farmers get lesser returns because there are multiple levels of intermediaries. To overcome this, various digital platforms have been created using Android-based mobile apps, based on their large-scale usage among users in rural regions.

Research has brought to light the fact that Android Studio, as an open-source development platform, is used extensively for mobile application development because of its strong features, ease of integration with cloud services, and real-time data update support. A number of mobile applications used in agriculture, like eNAM and Kisan Suvidha in India, have proved the success of digital solutions in enhancing farmers' access to markets. The incorporation of features like real-time price updates, direct communication between buyers and sellers, and secure digital transactions has increased the efficiency of agricultural trade. The existing literature also highlights the importance of easy-to-use interfaces, multilingual interfaces, and offline availability in mobile applications to support farmers with different backgrounds.

In addition, research on the adoption of digital agriculture solutions has shown that mobile apps can help mitigate post-harvest losses, improve profit margins, and ensure sustainable agriculture. Researchers have investigated how artificial intelligence (AI) and machine learning (ML) can aid in the best market forecast optimization to enable farmers to make informed decisions regarding the sale of their crops. Yet, digital literacy, internet connectivity in rural locations, and trust in online transactions are still a long way from being overcome as obstacles to adoption. Training programs, government assistance, and technological innovation can help overcome these obstacles further to improve the performance of mobile applications in enabling direct market access. This review points out the potential of Android Studio-based apps to revolutionize the agricultural marketing industry and emphasizes the necessity for further research on their application and influence on rural economies.

III. METHODOLOGY

The process of developing the Mobile App for Direct Market Access for Farmers is systematic and involves requirement gathering, system design, application development, testing, and deployment. The process guarantees that the mobile app is user-friendly, efficient, and scalable to cater to farmers in need of direct market access without the involvement of intermediaries.

1. Requirement Analysis:

The initial stage consists of gathering and analyzing farmers', buyers', and other stakeholders' requirements. Interviews and questionnaires were organized with farmers to realize their difficulty in reaching the market, concerns related to price, and level of digital literacy. Likewise, feedback was collected from buyers and agriculture experts to ascertain the important features required by the application. The results facilitated identification of core functionalities, i.e., listing products, managing orders, price updates in real-time, and secure payment processing.

2. System Design and Architecture:

The app was developed using Android Studio with a well-organized Model-View-Controller (MVC) structure to make it maintainable and scalable. The UI components were designed using XML, and the back end was developed using Java/Kotlin. A Firebase database was used for real-time data storage and retrieval, enabling farmers to update their product listings and interact with buyers smoothly. The flow of the application was planned with intuitive interface navigation, so it was simple to use even for farmers who do not have much technical background.

3. Application Development:

User Interface (UI) Development: XML-based design tools in Android Studio were employed to design a simple and interactive UI. Icons, buttons, and menus were made easy to access.

Back-End Development: Java/Kotlin was employed to realize fundamental functionalities, such as user authentication, product listing, price updating, and notifications. Firebase Authentication handled user registration and login.

Database Management: Firebase Realtime Database was chosen to hold farmer profiles, product information, and transaction data. The database schema was structured to allow rapid and effective data retrieval.

API Integration: Third-party APIs, like Google Maps API, were integrated in order to enable farmers to

© April 2025 | IJIRT | Volume 11 Issue 11 | ISSN: 2349-6002

identify markets and logistics near them. Integration with the payment gateway was also done for safe transactions.

4. Testing and Evaluation:

The app was tested through various phases of testing to validate functionality and usability. Unit testing was conducted using JUnit and Espresso to check the correctness of the code. Usability testing was done with a set of farmers and buyers selected to test the ease of use of the app. Their comments resulted in UI improvements and feature additions. Performance testing was also done to make load times and database queries as fast as possible for a seamless user experience.

5. Deployment and Future Improvements

Following successful testing, the application was rolled out on the Google Play Store for public use. Training sessions and digital literacy classes were organized to enable farmers to use the app effectively. Enhancements in the future include the integration of AI-based price forecasting, multilingual support, and blockchain for secure transactions.

6. Future Enhancements

While the app currently facilitates direct market access, future improvements include AI-based price prediction, blockchain-based transactions for enhanced security, and multilingual voice assistants for better accessibility. Continuous monitoring and updates will be implemented based on user feedback and evolving market trends.

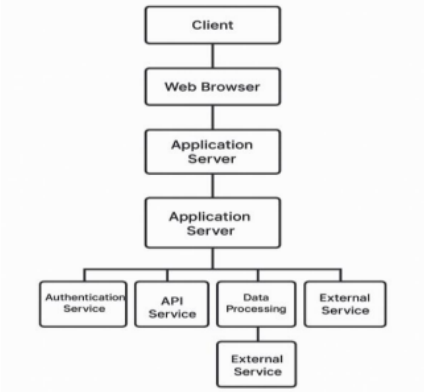


Fig.1 Architecture diagram

IV. RELATED WORK

The creation of a mobile app for direct market access by farmers based on Android Studio is an advancement on current technology progress in agriculture, mobile trading platforms, and online marketplaces. There have been various research studies and initiatives aimed at enhancing farmers' market access, the removal of intermediaries, and fair pricing via mobile apps. Our project builds on these concepts by incorporating an easy-to-use Android app specifically tailored to equip farmers with real-time market information, direct buyer-seller communication, and a hassle-free transaction process.

One of the major sources of inspiration for our work is existing research on mobile apps for agricultural e-commerce. Research has indicated that mobile-based platforms greatly enhance farmers' revenues by offering direct access to buyers, minimizing logistical expenses, and enhancing market transparency. A few established initiatives, like India's eNAM (Electronic National Agriculture Market), have tried to close the gap between buyers and farmers using digital platforms. Yet, most of these systems continue to be plagued by issues like cumbersome user interfaces, non-support for local languages, and poor integration with mobile technologies. Our app overcomes such issues by presenting an easy-to-use interface, multilingual assistance, and minimalistic navigation specially designed for rural farmers.

Also, mobile apps using IoT and AI-driven models of price prediction have been considered in the past to increase the accessibility of the market for farmers. AI algorithms have been utilized in some research to forecast prices of crops with the help of historical data, weather, and demand in the market. Although such methods provide useful information, they are computationally demanding and require technical know-how, hence not practical for small-scale farmers. Our application, on the other hand, relies on real-time updates of prices, communication between farmers and buyers, and secure payment facilitation to ensure that it is accessible without the need for technical expertise.

Another project that impacted our research is mobile supply chain management for agriculture. Several systems have been created to maximize the agricultural supply chain by linking farmers, wholesalers, and retailers on digital platforms.

© April 2025 | IJIRT | Volume 11 Issue 11 | ISSN: 2349-6002

Nevertheless, most of the available solutions tend to serve large-scale agribusinesses and not small and marginal farmers. Our project is targeting small farmers in particular by offering a user-friendly platform on which they can post their produce, negotiate directly with the buyers, and get paid instantly without any intermediaries.

Also, blockchain technology's use in agricultural commerce has been the subject of study over the last few years. Some mobile apps have tried to leverage blockchain for safe and open transactions in agriculture markets. Although blockchain has security advantages, it has intricated implementation and may not find much traction in rural communities with poor digital literacy. Our platform, on the other hand, includes secure payment gateways and transaction monitoring to provide a safe and effective market platform for farmers and consumers.

In short, our research and development of a mobile app for direct market access to farmers draw inspiration from available mobile-based agricultural platforms, AI-fueled market forecasting models, supply chain management apps, and blockchain-based solutions. Our solution, though, uniquely confronts the issues of small and marginal farmers by offering an easy, efficient, and user-friendly mobile app that enables direct trade, increases price visibility, and improves general market access.

V. CONCLUSION

The Mobile App for Direct Market Access for Farmers built on Android Studio is a revolutionary solution that closes the gap between market and farmers by doing away with intermediaries and providing fair prices for farm produce. With real-time market price feeds, direct communication between buyer and seller, and secure payment arrangements, the app makes the marketplace transparent and empowers farmers to make better decisions. The ease of use of the platform, its multi-language capability, and offline capability further facilitate use by farmers in rural locations with poor internet connectivity. The use of data analytics also assists farmers in maximizing sales of crops on the basis of demand patterns and seasonal price volatility. The research identifies the app's promise to revolutionize farm trade through an enhanced, equitable, efficient, and technology-based system that will ultimately result in greater financial stability for farmers and a better agricultural industry. Future developments would involve AI-based price forecasting, blockchain-supported security of transactions, and IoT support for intelligent farming data, further enhancing the influence of this technology.

REFERENCES

[1] Chang, H.-H., & Meyerhoefer, C.D. (2020). COVID-19 and the demand for online food shopping services: Empirical evidence from Taiwan. American Journal of Agricultural Economics, 103(2), 448–465. 2020– "Android App to Connect Farmers to Retailers and Food Processing Industry." IEEE Xplore. Available at: https://ieeexplore.ieee.org/document/9034434.

[2] IEEE Xplore. (2020). Android App to Connect Farmers to Retailers and Food Processing Industry. Available at: https://ieeexplore.ieee.org/document/9034434.

[3] Saxena, S., & Limbad, A. (2021). Consumption changes of household food habits pre and post lockdown during COVID-19: A perspective study of Gujarat and Maharashtra. Global Journal of Interdisciplinary Studies, 4(1), 16–36.

[4] Zhao, L., Zhang, Y., & Zhang, H. (2022). Research on the impact of digital literacy on farmer households' green cooking energy consumption: Evidence from rural China. International Journal of Environmental Research and Public Health, 19(13464).

[5] Manocha, S., Bhullar, P.S., & Sachdeva, T. (2023). Factors determining the investment behaviour of farmers—The moderating role of socioeconomic demographics. Journal of Indian Business Research.

[6] IEEE Xplore. (2023). Smart Kisan: A Mobile App for Farmers' Assistance in Agricultural Activities. Available at: https://ieeexplore.ieee.org/document/10199471.

[7] Hinojosa, C., Sanchez, K., Camacho, A., & Arguello, H. (2023). AgroTIC: Bridging the gap between farmers, agronomists, and merchants through smartphones and machine learning. arXiv preprint, arXiv:2305.12418. Available at: https://arxiv.org/abs/2305.12418.

[8] Kumar, R. (2023). Farmers' use of the mobile phone for accessing agricultural information in

© April 2025 | IJIRT | Volume 11 Issue 11 | ISSN: 2349-6002

Haryana: An analytical study. Open Information Science. Available at: https://www.degruyter.com/document/doi/10.1515/opis-2023-0031/html.

[9] Agriculture & Food Security. (2023). Is agricultural digitization a reality among smallholder farmers in Sub-Saharan Africa? Unpacking the digital agricultural paradox.

[10] Mdoda, L., Christian, M., & Agbugba, I. (2024). Use of information systems (mobile phone app) for enhancing smallholder farmers' productivity in Eastern Cape Province, South Africa: Implications on food security. Journal of the Knowledge Economy, 15, 1993–2009. Available at: https://link.springer.com/article/10.1007/s13132-024-01478-x.

[11] Sustainability. (2024). Bridging the digital divide: Empowering rural women farmers through mobile technology. Sustainability, 16(21), 9188. Available at: https://www.mdpi.com/journal/sustainability.

PLAGARISM REPORT OF PUBLISHED RESEARCH PAPER

Mobile App for Direct Market-CITG01_Paper_Publication

ORIGINALITY REPORT

2%	0%	2%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Presidency University Student Paper	1%
2	Debasis Chaudhuri, Jan Harm C Pretorius, Debashis Das, Sauvik Bal. "International Conference on Security, Surveillance and Artificial Intelligence (ICSSAI-2023) - Proceedings of the International Conference on Security, Surveillance and Artificial Intelligence (ICSSAI-2023), Dec 1–2, 2023, Kolkata, India", CRC Press, 2024 Publication	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		

s

PLAGARISM REPORT OF PROJECT REPORT

Bhavya B

B.BHAVYA - Mobile App for Direct Market Access for Farmers_CITG01_Report (3)

Quick Submit

Quick Submit

Presidency University

Document Details

Submission ID
trn:oid:::1:3219661904

Submission Date
Apr 17, 2025, 12:27 PM GMT+5:30

Download Date
Apr 17, 2025, 12:32 PM GMT+5:30

File Name
le_App_for_Direct_Market_Access_for_Farmers_CITG01_Report_3.pdf

File Size
1.4 MB

41 Pages

9,293 Words

54,808 Characters

*% detected as AI

AI detection includes the possibility of false positives. Although some text in this submission is likely AI generated, scores below the 20% threshold are not surfaced because they have a higher likelihood of false positives.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Disclaimer

Our AI writing assessment is designed to help educators identify text that might be prepared by a generative AI tool. Our AI writing assessment may not always be accurate (it may misidentify writing that is likely AI generated as AI generated and AI paraphrased or likely AI generated and AI paraphrased writing as only AI generated) so it should not be used as the sole basis for adverse actions against a student. It takes further scrutiny and human judgment in conjunction with an organization's application of its specific academic policies to determine whether any academic misconduct has occurred.

Frequently Asked Questions

How should I interpret Turnitin's AI writing percentage and false positives?

The percentage shown in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was either likely AI-generated text from a large-language model or likely AI-generated text that was likely revised using an AI-paraphrase tool or word spinner.

False positives (incorrectly flagging human-written text as AI-generated) are a possibility in AI models.

AI detection scores under 20%, which we do not surface in new reports, have a higher likelihood of false positives. To reduce the likelihood of misinterpretation, no score or highlights are attributed and are indicated with an asterisk in the report (*%).

The AI writing percentage should not be the sole basis to determine whether misconduct has occurred. The reviewer/instructor should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in accordance with their school's policies.

What does 'qualifying text' mean?

Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be likely AI-generated will be highlighted in cyan in the submission, and likely AI-generated and then likely AI-paraphrased will be highlighted purple.

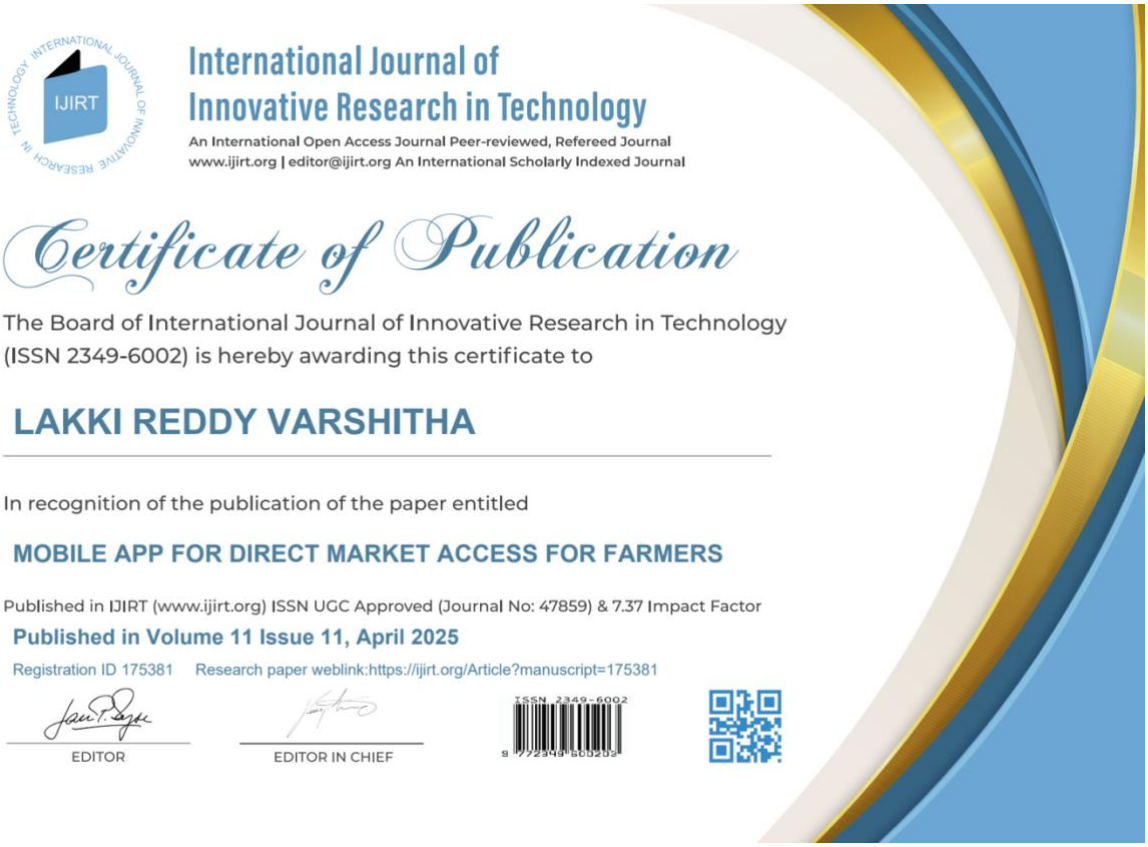
Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.

MAPPING THE PROJECT TO SDG MAPPING

The Project work carried out here is mapped to SDG-3 Good Health and Well-Being.

The main objective of the Mobile App for Direct Market Access for Farmers through Android Studio is to enhance agricultural trade efficiency, minimize reliance on middlemen, and enhance farmers' returns by offering a more transparent, direct, and accessible market.

PUBLICATION CERTIFICATES



International Journal of Innovative Research in Technology

An International Open Access Journal Peer-reviewed, Refereed Journal

www.ijirt.org | editor@ijirt.org An International Scholarly Indexed Journal

Certificate of Publication

The Board of International Journal of Innovative Research in Technology (ISSN 2349-6002) is hereby awarding this certificate to

LAKKI REDDY VARSHITHA

In recognition of the publication of the paper entitled

MOBILE APP FOR DIRECT MARKET ACCESS FOR FARMERS

Published in IJIRT (www.ijirt.org) ISSN UGC Approved (Journal No: 47859) & 7.37 Impact Factor

Published in Volume 11 Issue 11, April 2025

Registration ID 175381 Research paper weblink:https://ijirt.org/Article?manuscript=175381

EDITOR EDITOR IN CHIEF

International Journal of Innovative Research in Technology

An International Open Access Journal Peer-reviewed, Refereed Journal

www.ijirt.org | editor@ijirt.org An International Scholarly Indexed Journal

Certificate of Publication

The Board of International Journal of Innovative Research in Technology (ISSN 2349-6002) is hereby awarding this certificate to

ERANTI SAI KISHAN

In recognition of the publication of the paper entitled

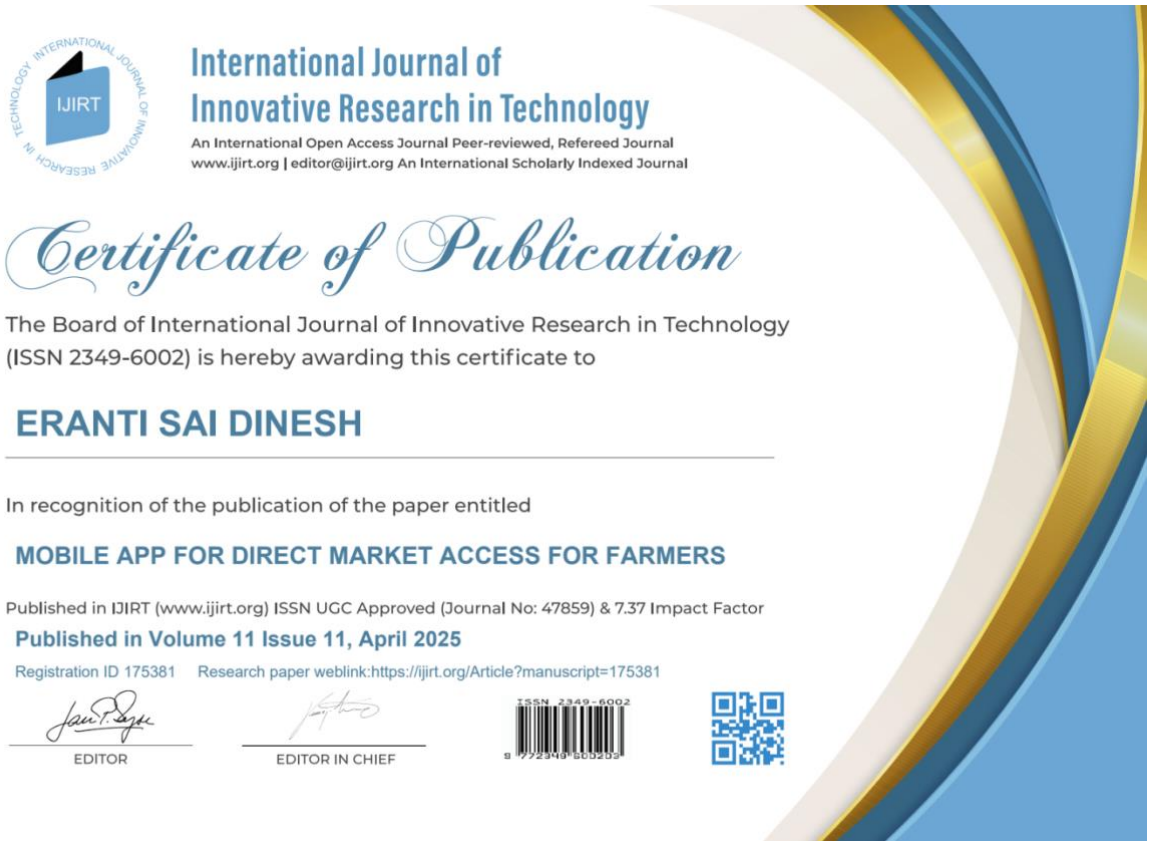
MOBILE APP FOR DIRECT MARKET ACCESS FOR FARMERS

Published in IJIRT (www.ijirt.org) ISSN UGC Approved (Journal No: 47859) & 7.37 Impact Factor

Published in Volume 11 Issue 11, April 2025

Registration ID 175381 Research paper weblink:https://ijirt.org/Article?manuscript=175381

EDITOR EDITOR IN CHIEF



International Journal of Innovative Research in Technology

An International Open Access Journal Peer-reviewed, Refereed Journal

www.ijirt.org | editor@ijirt.org An International Scholarly Indexed Journal

Certificate of Publication

The Board of International Journal of Innovative Research in Technology (ISSN 2349-6002) is hereby awarding this certificate to

ERANTI SAI DINESH

In recognition of the publication of the paper entitled

MOBILE APP FOR DIRECT MARKET ACCESS FOR FARMERS

Published in IJIRT (www.ijirt.org) ISSN UGC Approved (Journal No: 47859) & 7.37 Impact Factor

Published in Volume 11 Issue 11, April 2025

Registration ID 175381 Research paper weblink:https://ijirt.org/Article?manuscript=175381

EDITOR EDITOR IN CHIEF

International Journal of Innovative Research in Technology

An International Open Access Journal Peer-reviewed, Refereed Journal

www.ijirt.org | editor@ijirt.org An International Scholarly Indexed Journal

Certificate of Publication

The Board of International Journal of Innovative Research in Technology (ISSN 2349-6002) is hereby awarding this certificate to

PAVANI M

In recognition of the publication of the paper entitled

MOBILE APP FOR DIRECT MARKET ACCESS FOR FARMERS

Published in IJIRT (www.ijirt.org) ISSN UGC Approved (Journal No: 47859) & 7.37 Impact Factor

Published in Volume 11 Issue 11, April 2025

Registration ID 175381 Research paper weblink:https://ijirt.org/Article?manuscript=175381

EDITOR EDITOR IN CHIEF

International Journal of Innovative Research in Technology

An International Open Access Journal Peer-reviewed, Refereed Journal

www.ijirt.org | editor@ijirt.org An International Scholarly Indexed Journal

Certificate of Publication

The Board of International Journal of Innovative Research in Technology (ISSN 2349-6002) is hereby awarding this certificate to

BATTULA BHAVYA

In recognition of the publication of the paper entitled

MOBILE APP FOR DIRECT MARKET ACCESS FOR FARMERS

Published in IJIRT (www.ijirt.org) ISSN UGC Approved (Journal No: 47859) & 7.37 Impact Factor

Published in Volume 11 Issue 11, April 2025

Registration ID 175381 Research paper weblink:https://ijirt.org/Article?manuscript=175381

EDITOR EDITOR IN CHIEF

International Journal of Innovative Research in Technology

An International Open Access Journal Peer-reviewed, Refereed Journal

www.ijirt.org | editor@ijirt.org An International Scholarly Indexed Journal

Certificate of Publication

The Board of International Journal of Innovative Research in Technology (ISSN 2349-6002) is hereby awarding this certificate to

SHARMASTH VALI

In recognition of the publication of the paper entitled

MOBILE APP FOR DIRECT MARKET ACCESS FOR FARMERS

Published in IJIRT (www.ijirt.org) ISSN UGC Approved (Journal No: 47859) & 7.37 Impact Factor

Published in Volume 11 Issue 11, April 2025

Registration ID 175381 Research paper weblink:https://ijirt.org/Article?manuscript=175381

EDITOR EDITOR IN CHIEF