BSIT-S25-007

03-135221-007 Bilal Rasheed

03-135221-003 Abeesha Shahnawaz

Harvest Direct: AI-Powered Freshness &Transactions

In partial fulfilment of the requirements for the degree of

Bachelor of Science in Information Technology

Supervisor: Zunnurain Hussain

Department of Computer Sciences

Bahria University, Lahore Campus

January 2026

C e r t i f i c a t e

We accept the work contained in the report titled

"Harvest Direct: AI-Powered Freshness & Transactions"

written by

Bilal Rasheed

Abeesha Shahnawaz

as a confirmation to the required standard for the partial fulfilment of the degree of Bachelor of Science in Information Technology.

Approved by:

Supervisor: ZUNNURAIN HUSSAIN

_____ (Signature)

January 05, 2026

DECLARATION

We hereby declare that this project report is based on our original work except for citations and quotations which have been duly acknowledged. We also declare that it has not been previously and concurrently submitted for any other degree or award at Bahria University or other institutions.

Enrolment	Name	Signature
03-135221-007	Bilal Rasheed	
03-135221-003	Abeesha Shahnawaz	

Date : January 04, 2026

Specially dedicated to

my beloved grandmother, mother and father

(Bilal Rasheed)

my beloved grandmother, mother and father

(Abeesha Shahnawaz)

(this dedication page is optional)

ACKNOWLEDGEMENTS

We would like to thank everyone who had contributed to the successful completion of this project. We would like to express my/our gratitude to my research supervisor, Mr Zunnurain Hussain for his invaluable advice, guidance and his enormous patience throughout the development of the research.

In addition, we would also like to express my gratitude to our loving parent who had helped and given me encouragement.

Bilal Rasheed

Abeesha Shahnawaz

Harvest Direct: AI-Powered Freshness & Transactions

ABSTRACT

The "Harvest Direct" project is a mobile application aimed at transforming the agricultural marketplace by directly connecting farmers with buyers, eliminating the need for intermediaries and ensuring fairer prices for both parties. The app features an AI-powered freshness detection system, using a MobileNetV2-based machine learning model to evaluate the quality of apples based on visual attributes, such as color and texture, ensuring that consumers can make informed purchasing decisions. Designed with user accessibility in mind, the app is particularly suited for rural farmers with limited technological experience, offering an intuitive interface, a 24/7 AI chatbot for real-time assistance, and secure transactions powered by Firebase for ease of use and data security. The platform allows farmers to upload product listings, set their own prices based on product quality, and manage orders without third-party involvement, thus promoting fair trade and improved profitability. It also supports bulk orders for businesses, such as restaurants and wholesalers, providing a streamlined approach to acquiring fresh produce. This project integrates essential features such as user authentication, a real-time database, and AI-powered tools, aiming to bridge the gap between farmers and consumers, create a more efficient supply chain, reduce food waste, and support sustainable farming practices. Future developments will expand the app's functionality to include additional produce categories, enhanced freshness detection, and more advanced AI features ensuring continued innovation in the agricultural sector.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS / ABBREVIATIONS

AI	Artificial Intelligence
CNN	Convolutional Neural Network
Firebase	A platform for building and managing mobile and web applications.
NLP	Natural Language Processing
TensorFlow Lite	A lightweight version of TensorFlow for mobile and embedded devices.
UI	User Interface
UX	User Experience
CRUD	Create, Read, Update, Delete
TFLite	TensorFlow Lite
SDK	Software Development Kit
JSON	JavaScript Object Notation
REST	Representational State Transfer

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Background

In E-farmers app which enables direct rice sales from farmers to consumers bypassing middlemen while addressing challenges like user registration and technological literacy among farmers[1]. The farmers' perceptions of mobile apps in India, highlighting how these tools provide essential farming information but also pointing out barriers such as limited access to technology and low digital literacy, which affect adoption[2]. Android-based agricultural apps offering features like weather forecasts and pest management. It emphasizes the need for a more comprehensive and integrated platform due to the challenges of using multiple apps[3].

A study examines various Android-based mobile apps designed for farmers including features like weather forecasting, soil health monitoring, and market prices and discusses the integration of Arduino systems to provide real-time data for farm management[4].

Another study highlights the significance of mobile apps like Plantix, Kisan Suvidha, and Farm-o-Pedia in the agricultural sector. These apps equip farmers with essential information on pest control, crop management, and market prices, effectively bridging the information gap and enabling farmers to make more informed decisions. [5]. A paper highlights apps like Pusa Krishi and Kisan Suvidha which help farmers improve productivity, market access and financial management by providing real-time data on weather, crop management, and market prices[6].

Harvest Direct is a mobile app that connects farmers directly with buyers eliminating middlemen for better prices and fresher produce. It features an AI-powered freshness detection system for apples, a 24/7 chatbot for support, and secure transactions through Firebase. The app also supports bulk purchases for businesses, aiming to create a more efficient and transparent marketplace.

1.2 Problem Statements

The core issue lies in how disconnected farmers and buyers still are. A farmer may harvest high-quality produce but has no way to reach buyers directly, so they end up selling to wholesalers at whatever rate is offered. Once the produce reaches retail shops or online vendors, the price increases significantly but yet the farmer still receives the same low amount. This imbalance has existed for years and leaves both farmers and buyers frustrated.

Another problem is the lack of transparency regarding freshness. People often buy fruits or vegetables that look fine from the outside but turn out to be old or spoiled inside. There is no dependable way for buyers to verify freshness, and many online stores simply mix newer and older stock together. This leads to mistrust, complaints, and food waste.

Existing agricultural apps in Pakistan mostly share information such as weather updates, crop guidance, or government schemes. Very few provide an actual marketplace where farmers can sell directly. And almost none offer any AI-based feature to verify product quality, which has become increasingly important in digital commerce.

In short, there is a clear need for a system that connects farmers and buyers directly, provides a dependable way to assess freshness, ensures secure transactions, and remains simple enough for rural users to adopt without difficulty.

1.3 Aims and Objectives

Aim:

To create a practical, easy-to-use mobile application that strengthens the link between farmers and buyers, allowing them to trade fairly, verify quality confidently, and communicate without unnecessary middlemen.

Objectives:

I) Build a user-friendly mobile app using Flutter where farmers can upload their products with photos, basic details, and prices, and buyers can browse and purchase without needing a third party.

II) Introduce an AI-based freshness detection system (starting with apples) to help buyers make informed decisions and avoid spoiled or low-quality products. This adds a layer of trust that current platforms lack.

III) Integrate a 24/7 AI chatbot that can guide users through common issues like account setup, product listing, or troubleshooting. This helps users who may not be very tech-savvy.

IV) Use Firebase for backend services, ensuring fast performance, secure data storage, and reliable authentication without the complexity of managing physical servers.

V) Encourage fair trade practices by eliminating middlemen and giving farmers the ability to set their own prices based on the actual quality of their produce.

VI) Enable bulk purchases for restaurants, wholesalers, and businesses that prefer to deal directly with farmers for fresh supplies.

1.4 Scope of Project

The scope of Harvest Direct focuses on developing a fully functional mobile application that provides a direct marketplace for farmers and buyers while integrating intelligent features that improve trust and usability.

In Scope:

- Designing and developing an Android mobile app using Flutter.
- Creating separate interfaces for farmers and buyers with features tailored to their needs.
- Enabling product listings, search options, and order placement.
- Implementing an AI model that analyzes apple images to classify whether they appear fresh or spoiled.
- Adding an AI chatbot capable of answering common questions and guiding users step-by-step.
- Integrating Firebase for real-time database operations, user authentication, and cloud storage.
- Providing notifications for orders, updates, and important alerts through Firebase Cloud Messaging.
- Supporting both individual buyers and businesses that may require bulk orders.

Out of Scope (but possible future enhancements):

- Creating an iOS version of the application.
- Adding payment gateways such as Easypaisa, JazzCash, or bank transfer options.
- Expanding AI freshness detection to include more fruits and vegetables.
- Building a full web-based dashboard for administrators or suppliers
- Developing logistics modules like delivery tracking or route optimization.

CHAPTER 2

LITERATURE REVIEW (and/or SRS)

2.1 Technological Constraints and Digital Literacy

Many studies highlight significant technological constraints affecting agricultural mobile apps, particularly related to farmers' low digital literacy. In rural areas, many farmers struggle to navigate apps due to limited experience with smartphones, leading to low adoption rates and underutilization of app functionalities [1]. The dependency on internet access exacerbates this issue, as many rural farmers lack reliable or affordable connectivity, limiting the usability of these apps [1][5]. Additionally, awareness and training are key factors, as many farmers do not have the necessary knowledge to fully utilize mobile technology [2].

There is a clear need for improved digital literacy through targeted training and user-friendly app design to make mobile apps accessible for farmers with varying technological skills.

2.2 Access to Technology and Infrastructure Limitations

Access to smartphones and internet infrastructure remains a significant barrier for farmers in rural areas, limiting their ability to benefit from agricultural mobile apps [2]. The digital divide prevents many farmers from using mobile solutions, as not all have access to modern smartphones or the necessary network connectivity [5]. Inadequate infrastructure in rural areas such as poor irrigation systems, unreliable storage facilities, and limited access to electricity further complicates the adoption of mobile apps [8].

To address these limitations, future work could focus on providing affordable, low-cost smartphones and enhancing rural connectivity through alternative networks or satellite-based solutions.

2.3 Lack of Integration and Fragmented Solutions

Several agricultural apps are designed to serve specific functions, such as weather updates, pest control, or market price information. However, these apps often lack integration, forcing farmers to use multiple apps for different tasks, which is inefficient and time-consuming [3][7]. This fragmentation limits the overall utility of these apps for farmers who require a comprehensive solution for managing their agricultural activities. To address this, future work should focus on developing all-in-one platforms that combine these functionalities into a single app, simplifying the experience and improving usability [4].

2.4 Connectivity Issues and Offline Functionality

A major challenge highlighted across studies is the reliance on continuous internet access, which is often unavailable in rural areas. Poor network coverage and high data costs make it difficult for farmers to access necessary information via apps [1][6]. To make apps more accessible, future developments should incorporate offline functionalities, allowing farmers to continue accessing essential features even without an internet connection [6][7]. Reducing data consumption and designing apps for low-end smartphones will be essential in overcoming these barriers and improving adoption in remote areas [6].

2.5 Language and Literacy Barriers

Language barriers and varying literacy levels significantly hinder the effectiveness of agricultural apps, especially in regions where farmers may not be proficient in dominant languages such as English [3]. Many farmers in rural areas struggle with illiteracy, which limits their ability to use apps that require reading and comprehension. Localization of apps into regional languages and simplifying content for illiterate farmers is critical to making these technologies more accessible. Future work should

focus on creating apps in multiple languages and ensuring that they cater to different literacy levels, thus broadening their user base [4].

2.6 Market Access and Supply Chain Challenges

Another limitation noted is the difficulty farmers face in accessing markets directly due to reliance on intermediaries and middlemen in the agricultural supply chain. This reduces their ability to secure fair prices for their produce and increases inefficiencies [9]. Many farmers also lack access to extension services and struggle with poor connectivity, further hindering their ability to utilize mobile apps effectively. Future research should focus on creating mobile platforms that connect farmers directly to buyers and eliminate the need for intermediaries, thus improving market access and profitability. Additionally, integrating real-time market data and payment solutions will help facilitate direct transactions between farmers and buyers [9].

2.7 AI and Machine Learning for Crop and Disease Management

The use of AI and machine learning for crop management and disease detection has shown potential but also presents challenges. Many apps rely on complex algorithms that require extensive data and computational power, which may not be feasible for farmers with limited resources [22]. Furthermore, many machine learning models face limitations in accuracy due to issues like limited datasets, overfitting, and hardware constraints [17][22]. Future work should focus on improving the accuracy of these models by expanding datasets, improving real-world applicability, and enhancing the processing power of devices. Developing lightweight, optimized AI models for handheld devices will be crucial for enabling more farmers to access these advanced tools for crop management and disease detection [14][22].

2.8 Area of Study

The study focuses on this project, a mobile application designed to improve the agricultural supply chain by connecting farmers directly with buyers eliminating intermediaries and ensuring fair pricing and fresh produce. The app integrates AI-powered freshness detection for quality verification, particularly for apples, and

features a 24/7 AI chatbot for user support. Built using Flutter and Firebase, it offers secure, real-time transactions with AES encryption and supports multiple regional languages to enhance accessibility. Additionally, it facilitates bulk purchases for businesses and restaurants, promoting sustainable farming and reducing food waste, while empowering farmers with better profit opportunities and consumers with high-quality, fresh produce.

Table 2.1 SLR (Systematic Literature Review)

Year	Paper Title	Authors	Methodology	Performance Metrics	Contributions	Future Gap	Dataset
2017	Designing mobile farmer application using object oriented analysis and design.	Mufti, A., Novianti, D., & Anjani, D.	Research and Development (R&D); Object-Oriented Analysis and Design	System usability, application functionality after testing	Design of a mobile app integrating existing system, improves access for farmers	Focus on better system integration and testing on a larger scale	Not specified
2024	A study of farmers perception about agricultural mobile apps.	Singh, D., Shehrawat, P. S., Godara, A. K., & Kumari, N	Survey-based study; focus group discussions with farmers	User feedback, usability testing, adoption	Highlights the challenges and opportunities in mobile applications for farmers	Need for more user-friendly apps for rural farmers with limited	Not specified

						digital literacy	
2016	Survey of android apps for agriculture sector.	Patel, H., & Patel, D.	Survey and analysis of existing mobile	Performance of apps in terms of user interface and functionalities	Identifies gaps in current mobile applications and suggests improvements	Integration of multiple farming functionalities in a single app, offline capabilities	Not specified
2017	Comparative study of android-based M-Apps for farmers.	Kaur, S., & Dhindsa, K. S	Survey-based study, comparison of M-apps features	User satisfaction, app ratings, and downloads	Identifies the most useful agricultural apps and their benefits	Need for better integration and user-friendly design	Not specified

2024	Review on Use of Mobile Applications in Digital Agriculture.	Khan, Sheema & Parihar, Poonam.	Descriptive study, secondary data from Google Play and app reviews	App downloads, ratings, and user feedback	Review of mobile apps for agriculture, highlighting user adoption barriers	Addressing technological literacy and internet connectivity issues	Not specified
2023	Mobile applications for agricultural transformation: Types, impacts, case studies, and recommendations.	Yadav, A., Thilagam, P. and Singh, S	Literature review, secondary data collection from agricultural apps	App performance, farmer satisfaction, usage statistics	Discusses impact of mobile apps on smallholder farmers and their productivity	Improving data privacy, security, and enhancing user experience	Not specified
2024	Mobile Technology for Farmers: An Overview	Kambale, Parashuram & M, Dharma & Patil, Dayananda	Descriptive research, seconda	Popularity, app ratings	Highlights how mobile apps help improve	Future focus on better data	None

	of Agricultural Apps.	& N R, Ganavi	ry data analysis		agricultural practices	integration	
2022	Smart Farming Applications on Agriculture.	Babashli, Bakhtiyar.	Review of smart farming applications using IoT for data collection and integration.	Yield forecasting, smart irrigation effectiveness, and soil quality prediction.	Focuses on the application of smart farming technologies for precision agriculture and increased crop yield.	Need for better evaluation of cost-effectiveness and scalable solutions.	Not specified
2024	A Survey Paper On: Uplifting Farmers with Mobile Applications: An Overview of Modern Agricultural Utilities.	Firame, S., Dhamale, A., Yerme, A., Kote, K.,& Shinde, S.	Android-based application design and development	Usability, market linkage effectiveness	Develops an app to link farmers with buyers and suppliers	Lack of infrastructure and knowledge for new methods	Not specified

2022	An Android App for Farm: A Survey.	Krishna, R. V. C., & Adaickalam,	Survey-based, with Android SDK and Xampp Server used for testing	Farmer satisfaction, app usability, feature testing.	Develops an Android-based system to assist Indian farmers in managing weather updates, market prices, and other agricultural needs.	More focus on enhancing mobile network access for farmers, and scaling the app's functionality.	Not specified
2018	Android app to connect farmers to retailers and food processing industry.	Shriram, P., & Mhamane, S	Mobile app with API integration, Firebase server for storage	User registration, product search, price filtering	Facilitates direct sales between farmers, retailers, and food processing industries, removing	Expand database for more products, improve language support for regional farmers	Not specified

					middle men		
2024	Fruit classification based on freshness. In 2024 International Conference on Emerging Techniques in Computational Intelligence (ICETCI) (pp. 373-381).	Muttaqin, I. F., & Arifudin, R	Image processing and machine learning (CNN), Android app development	95% accuracy, fast classification	System classifies fruits based on freshness using CNN, reducing waste	Expand dataset, enhance real-world utility	2000 images of fruits (apples, bananas, oranges) and vegetables (tomatoes, capsicum, bitter gourd)
2025	Smart Harvesting Solutions: Robotics in Fruit and Vegetable Harvestin	Sathiyasuntharam, V., Navaneethan, S	Robotic harvesting system using CNN, depth sensors, reinforc	95% accuracy in detecting mature fruits, 50% time reduction, 5%	System aims to reduce labor dependency, increase efficiency, and	Expand application to various crops, improve adaptability	Not specified

	g for Reduced Labor Dependency.		ement learning	damage rate	reduce damage during harvesting		
2024	Fruit classification based on freshness.	Mallegowda, M., Sanskar, R. G., VISHVESHWARA, N., Safwan, G. A., & Vivek, J.	AI-based classification using CNN and transfer learning, Agile methodology	Accuracy >90%, fast classification	Classification of fruits based on freshness using visual features (color, texture)	Expand dataset, improve real-time monitoring	2000 images of fruits and vegetables
2022	Real-time quality assurance of fruits and vegetables with artificial intelligence.	Tata, J. S., Kalidindi, N. K. V., Katherapaka, H., Julakal, S. K., & Banothu, M.	AI-based classification using CNN, transfer learning, and data augmentation	Over 90% accuracy in freshness detection	AI system for classifying fruit and vegetable freshness using visual attributes	Expand dataset, improve real-time monitoring	2000 images of fruits and vegetables

2018	In field fruit sizing using a smart phone application.	Wang, Z., Koirala, A., Walsh, K., Anderson, N., & Verma, B	Image analysis, allometric modeling	RMSE for distance estimation, accuracy 95%	Fruit size estimation using camera	Improve frame design, real-time data accuracy	Mango, apple, avocado, mandarin datasets
2020	Pear and apple recognition using deep learning and mobile.	Kodors, S.	Deep Learning for pear/apple classification	Not specified	Fruit recognition for quality analysis	Improve system performance, real-time deployment	Fruit images dataset
2022	Prediction of fruit maturity, quality, and its life using deep learning algorithms.	Aherwadi, N., Mittal, U., Singla, J., Jhanjhi, N. Z., Yassine, A., & Hossain, M. S	Case study of mobile applications for agriculture	User feedback, app usability, download rates	Explores agricultural apps' impact on farming practices	Expand integration, improve user education	Not specified

2022	Implementation of a Fruit Quality Classification Application Using an Artificial Intelligence Algorithm.	Chen, M. C., Cheng, Y. T., & Liu, C. Y.	YOLO-based object detection and CNN for fruit quality classification	88% classification accuracy, image-based tracking	Develops a fruit quality classification system using YOLO and CNN for fruit defects detection	Expand dataset, improve real-time tracking capabilities	6000 images of apples, oranges, lemons, defective fruits
2024	Development of a cross-platform mobile application for fruit yield estimation	Duncan, B., Bulanon, D. M., Bulanon, J. I., & Nelson	Color ratio-based image-segmentation algorithm using OpenCV in a cross-platform mobile app	MAPE 8.52%, Pearson Correlation 0.6, iOS 0.14s, Android 1.16s	Developed Fruit Harvest Helper app for apple yield estimation	Improve algorithm, expand to other fruit types, conduct broader testing	73 images from Symm's Fruit Ranch of Pink Lady apple trees.

2024	Development of apple fruit classification system using convolutional neural network (cnn) mobilenet architecture on android platform.	Masparudin, M., Fitri, I., & Sumijan	Comparative study of existing mobile apps for agriculture, focusing on usability and features	App usage rate, user feedback, functionality effectiveness	Identifies gaps in current agricultural apps and suggests improvements for better farmer adoption	Addressing infrastructure and education gaps for rural farmers	Not specified
2024	Apple Detection: A CNN Approach for Diseases Detection.	Manzoor, E. S., Malhotra, R., Bhat, R., & Shekhar	CNN-based approach for detecting apple diseases using image analysis	Accuracy of 92%, precision 89%, recall 87%	Uses deep learning (CNN) to detect diseases in apple fruits with high accuracy	Further enhancement of model generalization and real-time disease detection	Apple fruit images with labeled disease types

CHAPTER 3

DESIGN AND METHODOLOGY

3.1 Development Methodology

3.1.1 Agile Scrum Framework

To manage the complexity of the project, which included app development, AI model integration, and backend management, the team used the Agile Scrum framework. The work was divided into short, manageable sprints, each focusing on completing a specific module or component of the system. This approach allowed the team to quickly adapt to new requirements, implement feedback from users, and deliver a functional product at the end of each sprint.

3.1.2 Sprint Breakdown

Each sprint involved focused tasks designed to achieve a particular milestone. The breakdown of the sprints is as follows:

- Sprint 1: Requirement gathering, defining user personas, and outlining the basic app flow.
- Sprint 2: UI/UX design using Figma, followed by setting up the Flutter project for mobile development.
- Sprint 3: Firebase authentication setup and database structure configuration.
- Sprint 4: Development of the farmer and buyer modules, including product listing, browsing, and chat functionalities.
- Sprint 5: AI model training using Google Colab, along with dataset preparation.
- Sprint 6: Integration of the API for the AI-powered freshness detection system.
- Sprint 7: Final testing, bug fixing, and deployment for the first release.

This iterative process ensured that feedback was incorporated into the development, allowing continuous improvement.

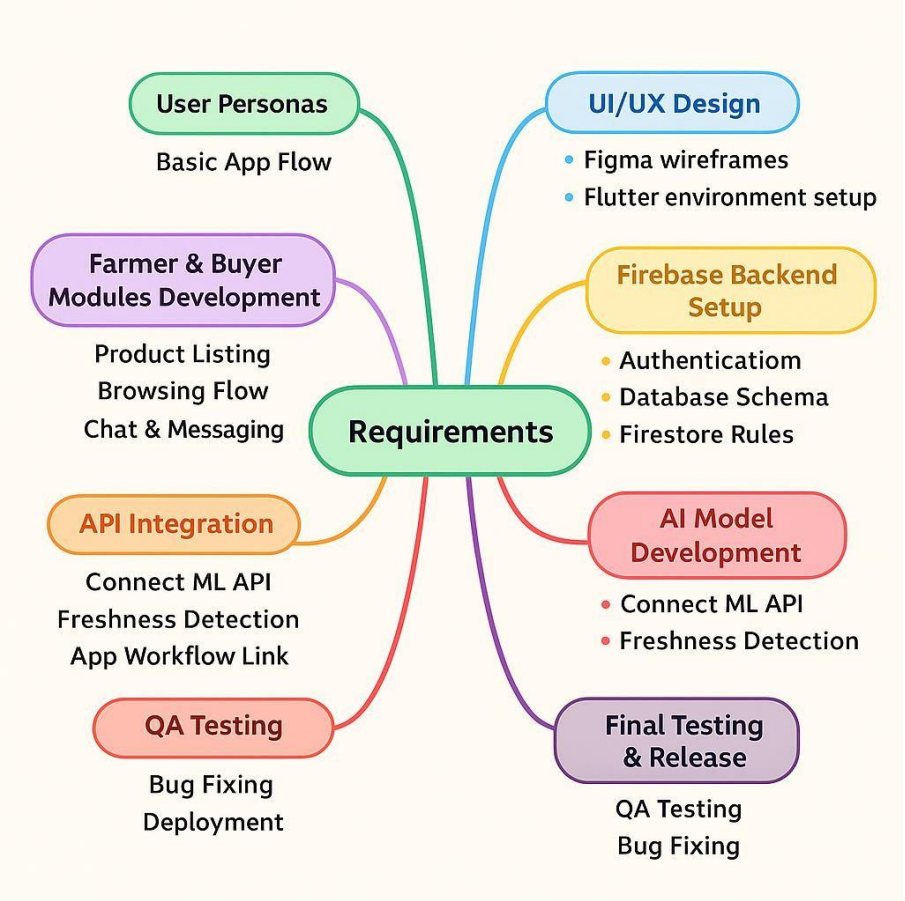


Figure 0.1: Sprint Breakdown Tasks

3.2 System Architecture

The Harvest Direct system is composed of several interconnected components, each with a specific role. These components work together to provide a seamless experience for both farmers and buyers.

3.2.1 High-Level Architecture

The system consists of four main layers:

- Frontend Layer: The mobile application built using Flutter for cross-platform development.

- Backend Layer: Powered by Firebase, which includes Firebase Authentication, Firestore for database management, Firebase Storage for image handling, and Firebase Cloud Messaging for real-time notifications.
- AI Model Layer: A MobileNetV2 model, converted into TensorFlow Lite for efficiency on mobile devices.
- Chatbot Layer: A Natural Language Processing (NLP)-based conversational support system to assist users.

Each layer communicates only where necessary, minimizing complexity and improving the system's performance.

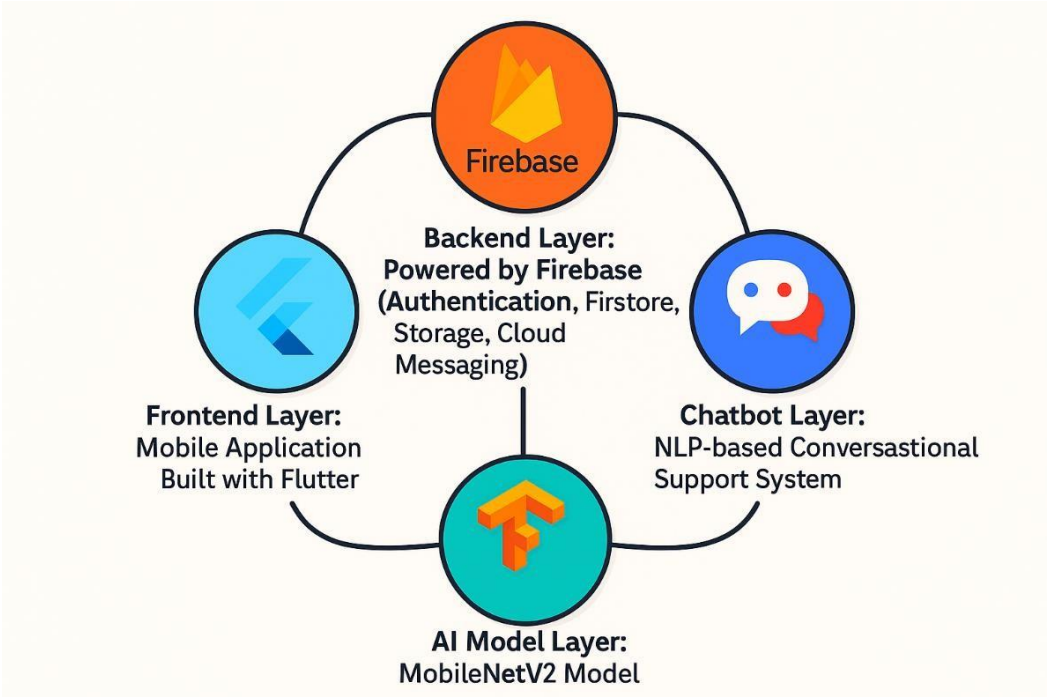


Figure 0.2: High-Level Architecture

3.3 Mobile Application Design

3.3.1 User Interface (UI) Structure

The UI of the app was designed with simplicity and ease of use in mind, as many farmers may not be familiar with complex mobile applications. The app layout features large text, clear navigation, and minimal clutter.

Main UI Modules:

Table 0.1: Main UI Modules of Harvest Direct

Module	Description
Farmer Module	Allows farmers to add product details, upload images, set prices, and manage listings.
Buyer Module	Enables buyers to browse produce, apply search filters, chat with farmers, and place orders.
Chat Module	Provides one-to-one messaging with push notifications for real-time communication.
Freshness Detection	Allows buyers to upload apple images and receive a freshness status.
Chatbot	Offers FAQ and step-by-step guidance through conversational prompts.

3.3.2 User Experience Design

- **Big Buttons for Easy Tapping:** Ensures ease of use, especially for users with limited mobile experience.
- **Minimal Typing:** Reduces the need for text input, making the app more user-friendly.
- **Icons:** Icons are used to assist farmers who might not be fluent in English.

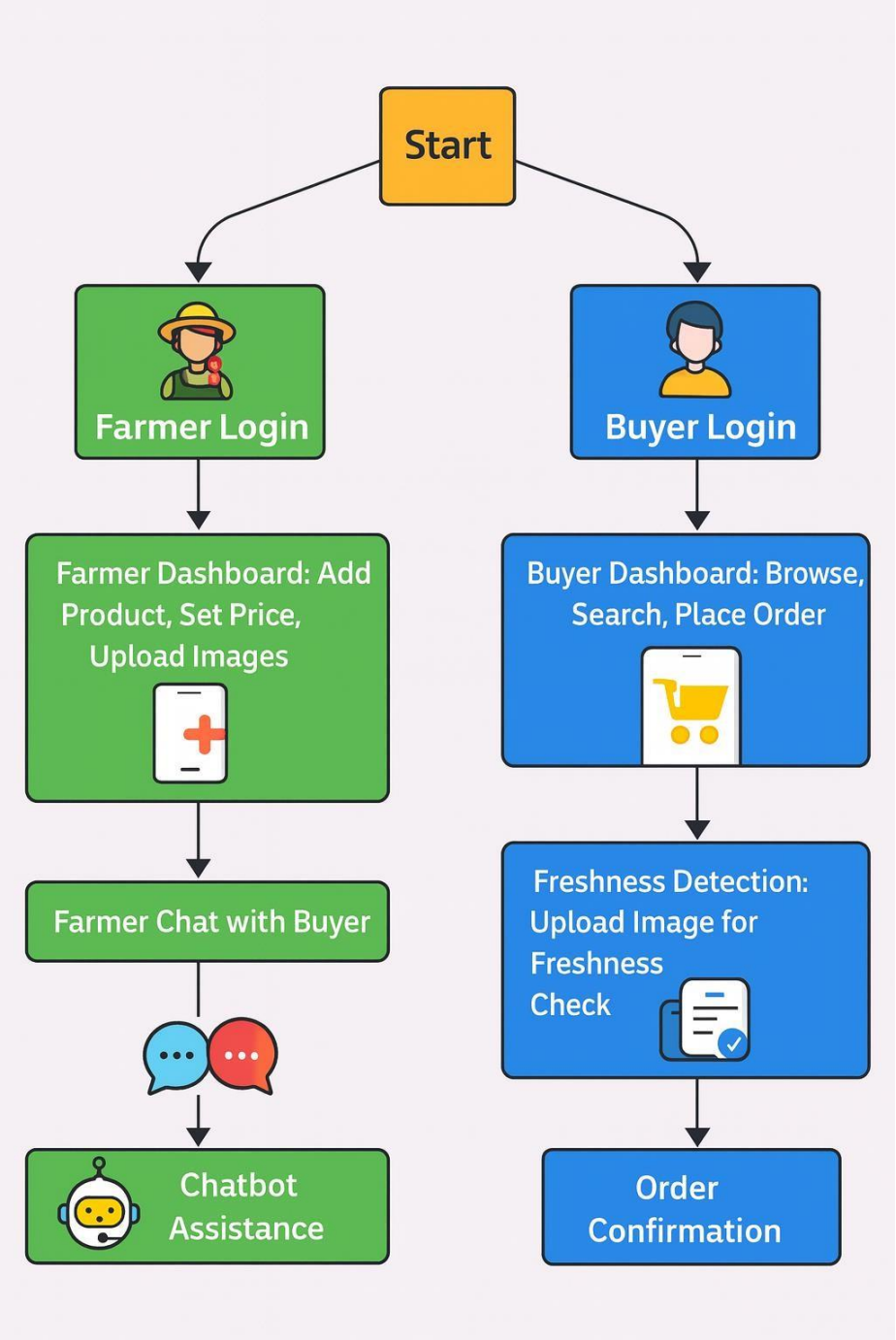


Figure 0.3: Users Flow Diagram

3.4 Backend Design

3.4.1 Firebase Authentication

Firebase Authentication provides a robust solution for managing user authentication and secure logins within the Harvest Direct app. It allows both farmers and buyers to

access the app with secure credentials. The Firebase Authentication service is easy to integrate and provides various authentication methods. Here's how the authentication system works for different types of users:

Buyer Login:

- Buyers can create an account or log in with their email and password.
- After logging in, buyers can browse through the products listed by farmers, view details, and place orders.

Farmer Login:

- Farmers create their own account or log in securely using their credentials to manage their products.
- Once logged in, farmers can upload their products, set prices, and manage product details such as images, descriptions, and quantity.

Password Reset:

If a user forgets their password, Firebase Authentication provides a secure password reset flow. This ensures that only the rightful user can reset the password through their registered email.

Google Sign-In :

To simplify the login process, users can log in using their Google account. This provides a quicker and more convenient authentication method. Firebase Authentication makes it easy to integrate Google Sign-In for both iOS and Android platforms.

3.4.2 Firestore Database Structure

Firestore is a real-time NoSQL database that automatically syncs data across all devices, which makes it perfect for this app, especially with features that require real-time updates like product listings, orders, and chat systems. Below is the Firestore structure that will be used to store various types of data for Harvest Direct.

Users Collection:

The Users Collection stores user data, including personal information and their role (whether they are a farmer or buyer). Each user has a unique userId document.

Products Collection:

The Products Collection stores information about products listed by farmers, such as the product name, price, and image URL. Each product is stored in a document identified by productId.

Orders Collection:

The Orders Collection holds order data, including the buyer and farmer IDs, as well as the product ordered. This collection links buyers, farmers, and products in one record.

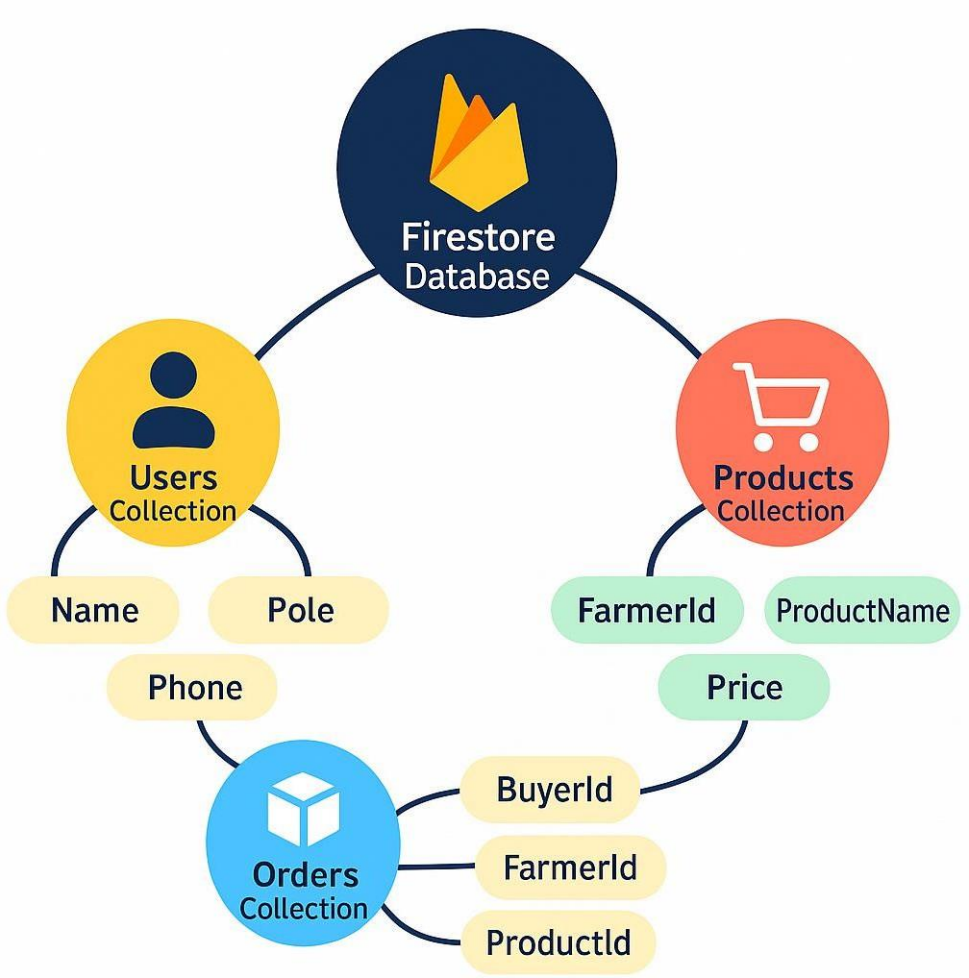


Figure 0.4: Firestore Database Structure

3.4.3 Firebase Storage

Firebase Storage is used to store and retrieve files such as images or videos. It is designed to be scalable and can handle large files efficiently. In the context of this app, Firebase Storage is used for two main purposes:

Product Images:

Farmers upload images of their products (e.g., apples) to Firebase Storage, which are then linked to the product listings in Firestore. These images are stored securely and can be easily retrieved when buyers browse products.

Freshness Detection Images:

When a buyer uploads an image of an apple to check its freshness, the image is sent to Firebase Storage. It is then processed by the AI model to determine if the apple is fresh or spoiled. The image is stored temporarily until the analysis is complete.

3.5 AI Model Design

3.5.1 Overview of Freshness Detection Model

We used MobileNetV2 model which is a lightweight Convolutional Neural Network (CNN) specifically designed for mobile devices. It excels at running efficiently on mobile platforms while maintaining high accuracy for image classification tasks. Here's why MobileNetV2 is ideal for the Harvest Direct app:

Table 0.2: Features of MobileNetV2

Feature	Description
Model Type	MobileNetV2 (Lightweight Convolutional Neural Network for mobile devices)
Speed	Optimized for mobile devices, ensuring fast and efficient execution even on devices with limited resources.
Compactness	Smaller in size compared to traditional CNNs, making it ideal for mobile environments where resources are limited.
Accuracy	Achieves high accuracy for image classification tasks, making it reliable for freshness detection.
Tenserflow Lite Compatibility	Can be converted into TensorFlow Lite (TFLite) format, which optimizes the model for mobile deployment and ensures faster inference times.

3.5.2 Dataset Description

The dataset used for training the model is crucial for its ability to detect apple freshness accurately. The dataset consists of two main categories:

Fresh Apple Images:

These images contain high-quality apples that are fresh and ready for sale. They serve as the positive class for classification.

Spoiled Apple Images:

These images contain apples that are rotten, bruised, or spoiled. They serve as the negative class for classification.

Preprocessing Steps:

Before training, the dataset undergoes several preprocessing steps to ensure consistency and improve the model's performance:

- **Resizing:** All images are resized to 224x224 pixels, a standard input size for MobileNetV2, to ensure uniformity across the dataset.
- **Normalizing:** Pixel values are scaled to a range of 0-1 by dividing by 255. This helps in faster convergence during training.
- **Data Augmentation:** Data augmentation techniques such as rotation, flipping, and zooming are used to artificially expand the dataset, introducing variety and helping the model generalize better on unseen images.

3.5.3 Model Training Steps

Here’s a step-by-step breakdown of how the model is trained:

Table 0.3: Model Training Steps

Step	Description
Load Pre-trained MobileNetV2	Use pre-trained weights from ImageNet to reduce training time and leverage previously learned features.
Remove Classifier Head	Remove the top layer (classifier) of MobileNetV2 and replace it with a custom classification layer.

Add Custom Dense Layer	Add a dense layer for classification, with a softmax output to predict either "Fresh" or "Rotten".
Train the model	Train the model on the apple dataset (fresh and spoiled images) to learn distinguishing features.
Evaluate Performance	Measure model accuracy and loss on test data to assess its generalization ability.
Convert to TensorFlow Lite	Convert the trained model into TensorFlow Lite for efficient mobile deployment.
Integrate with Flutter App	Integrate the TensorFlow Lite model into the Flutter app to provide real-time predictions for users.

3.5.4 Model Output

The output of the trained model is a classification label that determines the freshness of the apple. The two categories are:

Fresh (Class 0): If the model predicts that the apple is fresh, the app will display a message such as:

“This apple appears fresh.”

Rotten (Class 1): If the model predicts that the apple is rotten or spoiled, the app will display a message such as:

“This apple is likely spoiled.”

This real-time prediction helps buyers make more informed decisions when purchasing produce.

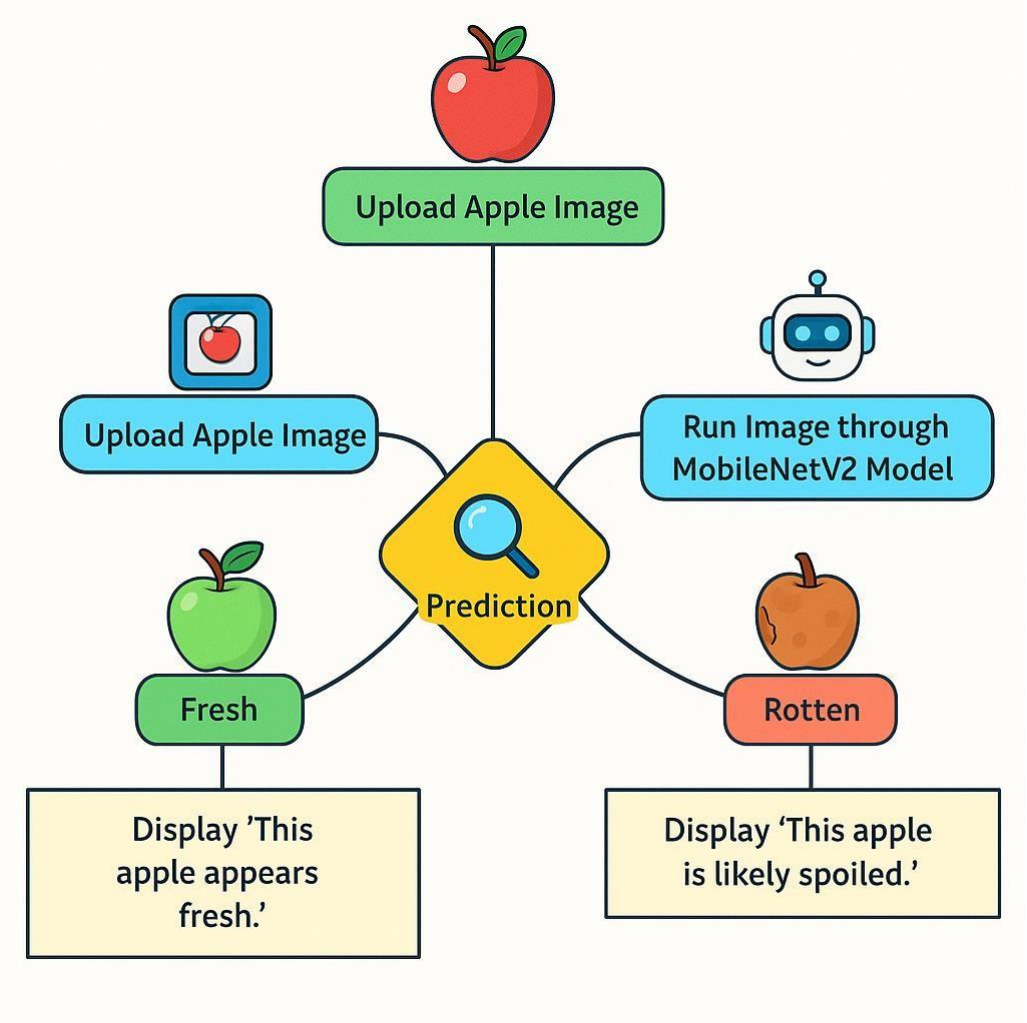


Figure 0.5: Model Output Flow Diagram

3.6 Chatbot Design

The Harvest Direct app integrates a Natural Language Processing (NLP)-based chatbot designed to assist both farmers and buyers by answering common queries, resolving issues, and guiding them through various processes. The chatbot enhances the user experience by providing instant support.

3.6.1 NLP-based Chatbot

An NLP-based chatbot uses algorithms that allow it to understand and process human language. Here are the key features of the chatbot:

- **FAQs (Frequently Asked Questions):**

The chatbot provides quick answers to common questions, reducing the need for users to search through the app or wait for support.
Examples of FAQs might include:
“How can I upload a product?”
“How do I place an order?”
“What is the freshness check process?”

- **Complaint Handling:**

If a buyer or farmer faces an issue (e.g., a problem with an order or the product), the chatbot helps resolve the dispute by providing troubleshooting steps or forwarding the complaint to the appropriate person.
Example:
“I received a spoiled apple. What should I do?”
The chatbot can either resolve the issue through the system or escalate it to a human support team.

- **Marketplace Guidance:**

The chatbot guides users through the marketplace, helping them understand how to browse products, apply filters, add products to their cart, and place an order.
It can also suggest products based on preferences or past behavior.

- **Step-by-Step Instructions:**

The chatbot assists with tasks by providing step-by-step instructions on how to use the app's features, such as setting up an account, uploading products, or checking the freshness of an apple.
Example:
"To upload a product, click on the 'Add Product' button, enter the product details, and upload an image."

3.6.2 Conversation Flow

The conversation flow describes the steps that the chatbot follows when interacting with users. Here's how the chatbot process works:

- **User Sends a Message:**

The user types a message in the chatbot interface. For example, “How do I place an order?” or “What is the price of apples?”

- **Bot Identifies the User's Intent:**

Using NLP algorithms, the bot processes the message and tries to understand the intent. Intent recognition is the process of determining the user's goal (e.g., asking for a price, requesting help, etc.).

- **Fetch the Appropriate Response:**

Once the intent is identified, the chatbot fetches the appropriate response from a predefined database or knowledge base. This could involve searching through FAQ data or triggering an action like escalating the issue to customer support.

- **Response Displayed in Real-Time:**

The chatbot sends the response back to the user, usually within seconds. This response can be a text message, a suggestion, or even an action trigger (like navigating to a particular screen in the app).

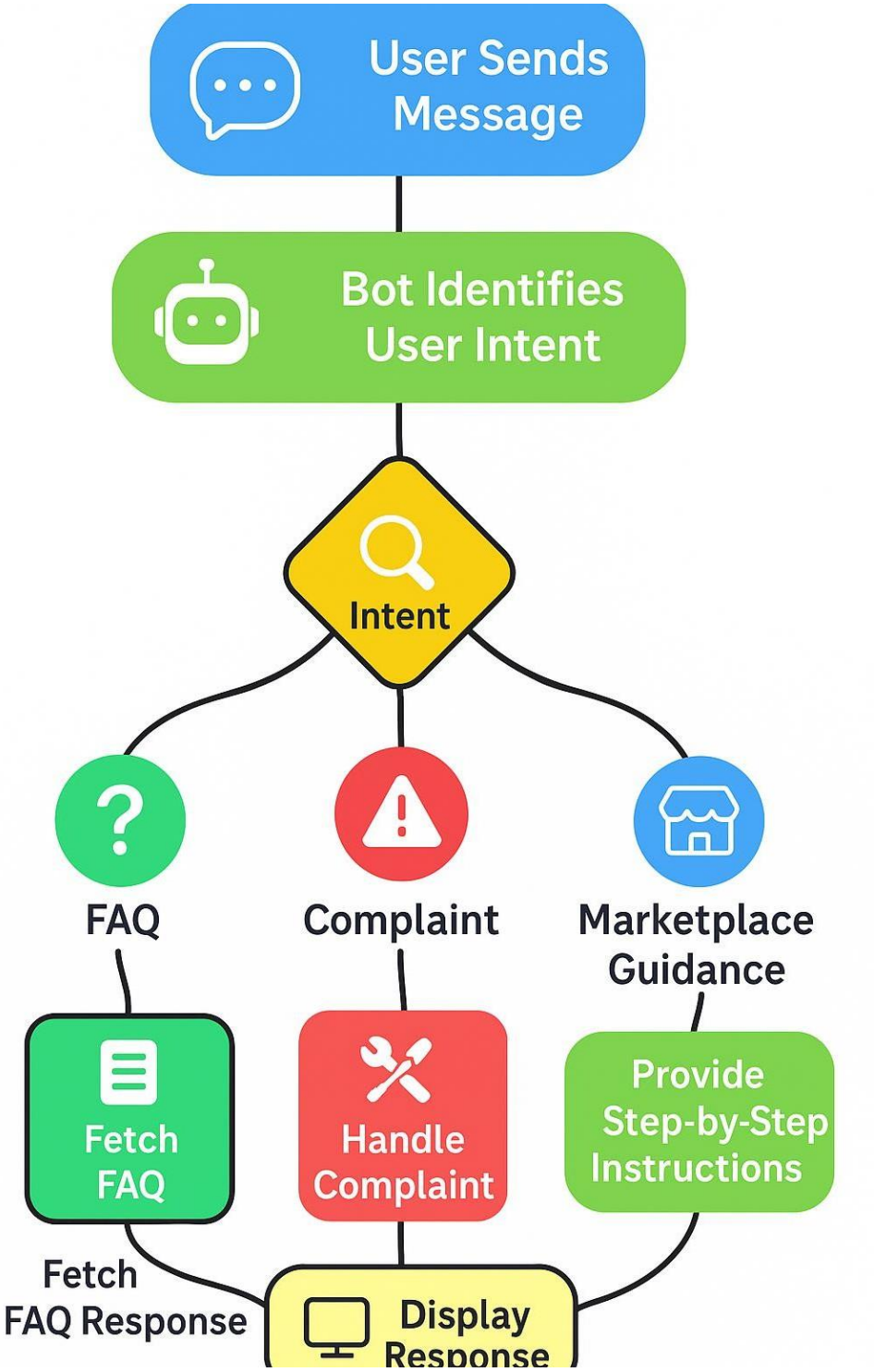


Figure 0.6: Chatbot Flow Diagram

3.7 Testing Strategy

The Testing Strategy is crucial for ensuring that the Harvest Direct app works correctly and efficiently. The table format provides clarity on the different testing types.

Table 0.4: Types of Testing Strategy with their Key Tests

Testing Type	Description	Key Tests
Functional Testing	Ensures the core functionality of the app works as expected.	**Product Upload:** Ensure farmers can upload products correctly.
		Order Placement: Validate the buyer's ability to place orders.
		App Navigation: Verify smooth transitions between screens.
AI Testing	Focuses on testing the AI model for accuracy, generalization, and handling edge cases.	**Unseen Images:** Test the model's ability to generalize to new images.
		Prediction Accuracy: Check the model's accuracy with different apple images.
		Edge Case Testing: Evaluate performance under conditions like low-light or blurry images.
User Testing	Involves feedback from real users (farmers and buyers) to improve usability and interface.	**Farmer Usability:** Test how easily farmers can manage listings and orders.
		Buyer Usability: Test how easily buyers can browse and place orders.
		Feedback Refinement: Collect feedback from both groups to improve the interface.

CHAPTER 4

DATA AND EXPERIMENTS (and/or IMPLEMENTATION)

4.1 Dataset

4.1.1 Overview of Dataset

For this project, we used the Fruit and Vegetable Disease - Healthy vs Rotten dataset from Kaggle, which contains images of various fruits and vegetables, including apples. The dataset is labeled with two categories: fresh (healthy) apples and spoiled (rotten) apples. These images serve as the foundation for training the AI model to classify the freshness of apples accurately in real-world applications.
Dataset Link: Fruit and Vegetable Disease - Healthy vs Rotten

4.1.2 Dataset Description

Fresh Apple Images: Images of high-quality, fresh apples.
Spoiled Apple Images: Images of apples that are spoiled, bruised, or rotten.
Image Types: The images are taken under different conditions, with varying light, angles, and backgrounds. This adds to the challenge of the model in distinguishing fresh apples from spoiled ones.

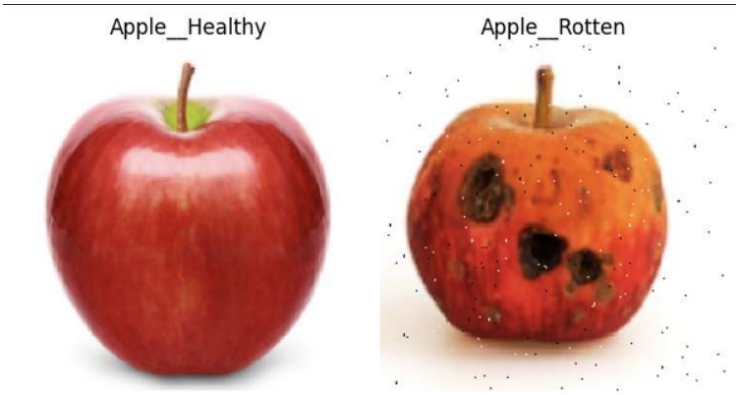


Figure 4.1: Random Images from both classes

4.1.3 Dataset Class Distribution

The class distribution of the dataset indicates the number of images for each class, which is crucial to ensure that the model is not biased towards one class due to data imbalance. A balanced dataset allows the model to learn the characteristics of both classes equally well.

For this dataset, the distribution is as follows:

Fresh Apples as Class 0

Spoiled Apples as Class 1

This class distribution can influence model performance, and strategies like data augmentation can be used to address any imbalance.

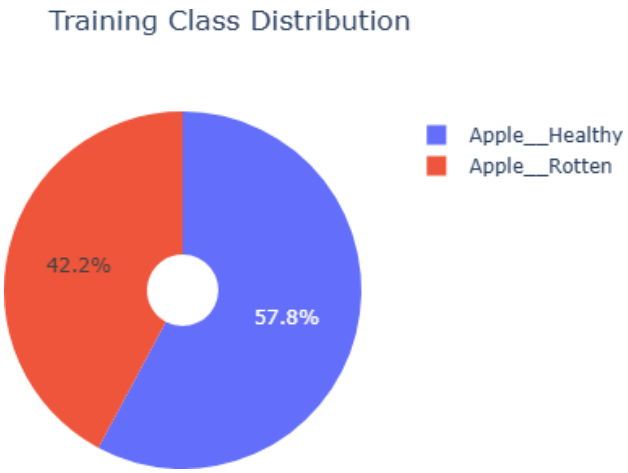


Figure 4.2: Classes Distribution

4.2 Preprocessing

4.2.1 Image Resizing and Normalization

All images were resized to 224x224 pixels, the input size required by the MobileNetV2 model. This standardization ensures that the model receives uniform input dimensions. After resizing, the pixel values were normalized by scaling them to a range of 0-1 by dividing by 255 to improve model convergence during training.

4.2.2 Data Augmentation

To address the issue of limited data and improve generalization, several data augmentation techniques were applied:

- Rotation: Randomly rotate images by up to 30 degrees.
- Flipping: Flip images horizontally.
- Zooming: Introduce zooming effects on the images.
- Brightness Adjustment: Randomly adjust the brightness of the images to simulate varying lighting conditions.

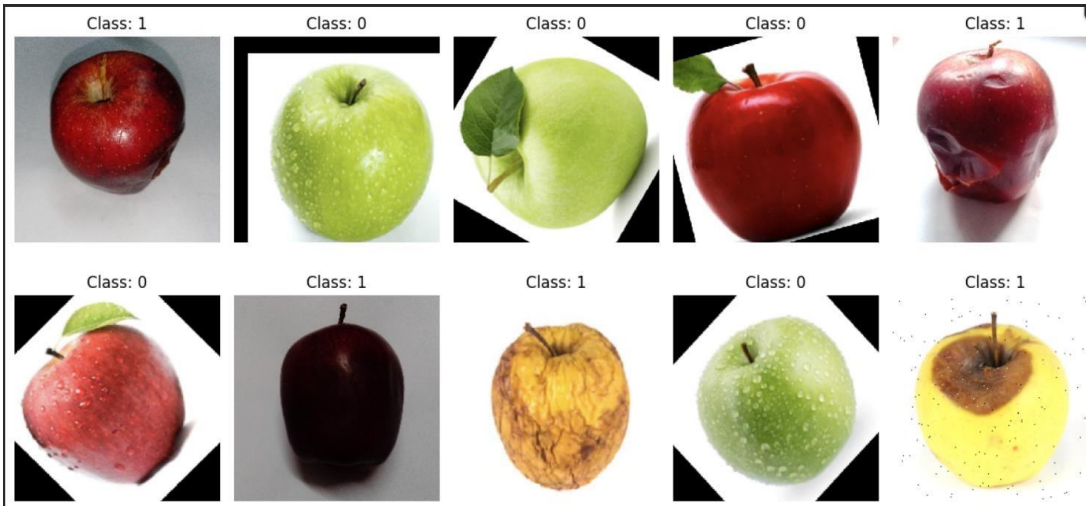


Figure 4.3: Visualization of the first 10 training images with their corresponding class labels to verify correct data loading and preprocessing.

4.3 Model Training

4.3.1 MobileNetV2 Architecture

The model chosen for this project is **MobileNetV2**, which is optimized for mobile applications. It uses depth-wise separable convolutions to reduce the model size while maintaining high accuracy, making it ideal for mobile deployment.

- **Model Type:** Convolutional Neural Network (CNN)
- **Efficiency:** Designed to work well with limited computational resources, especially on mobile devices.
- **Accuracy:** High accuracy for image classification tasks.
- **TensorFlow Lite Compatibility:** Can be converted into TensorFlow Lite format, ensuring faster and more efficient execution on mobile devices.

Layer (type)	Output Shape	Param #
input_layer_1 (InputLayer)	(None, None, None, 3)	0
resizing (Resizing)	(None, 224, 224, 3)	0
mobilenetv2_1.00_224 (Functional)	(None, 1280)	2,257,984
dense (Dense)	(None, 512)	655,872
dropout (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 2)	1,026

Total params: 2,914,882 (11.12 MB)
Trainable params: 656,898 (2.51 MB)
Non-trainable params: 2,257,984 (8.61 MB)

Figure 4.4: Model Summary

4.3.2 Transfer Learning and Fine-tuning

A **pre-trained MobileNetV2** model was used, fine-tuned on the dataset of apple images. This approach leverages pre-learned features from a large-scale dataset (ImageNet) to reduce training time and improve the performance of the model. The top layers of the pre-trained model were replaced with a custom dense layer to perform binary classification (Fresh vs Rotten).

4.3.3 Model Training Process

- **Optimizer:** Adam optimizer was chosen for efficient convergence.
- **Loss Function:** Binary Cross-Entropy, suitable for binary classification.
- **Metrics:** Accuracy and F1-score were used to measure model performance.
- **Epochs:** The model was trained for **20 epochs** with a **batch size of 32**.

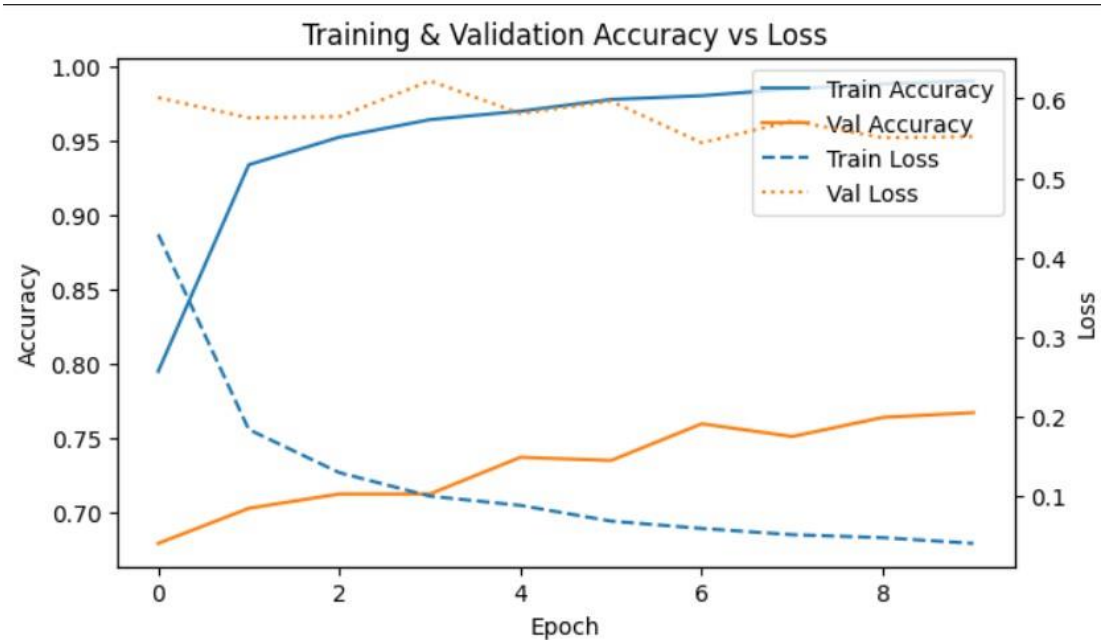


Figure 4.5: Training & Validation Acuuracy vs Loss

4.4 Model Evaluation

4.4.1 Testing the Model

The model was tested on a separate test set to evaluate its performance. The goal was to determine how well the model generalized to new, unseen data. The evaluation metrics used were accuracy, precision, recall, and F1-score.

- Test Accuracy: The percentage of correctly classified images in the test set.
- Precision: The ratio of true positive predictions to all positive predictions.
- Recall: The ratio of true positive predictions to all actual positives in the dataset.
- F1-Score: The harmonic mean of precision and recall, used to balance both metrics.

4.4.2 Model Performance Metrics

Confusion Matrix: The confusion matrix helped us visualize the true positives, true negatives, false positives, and false negatives.

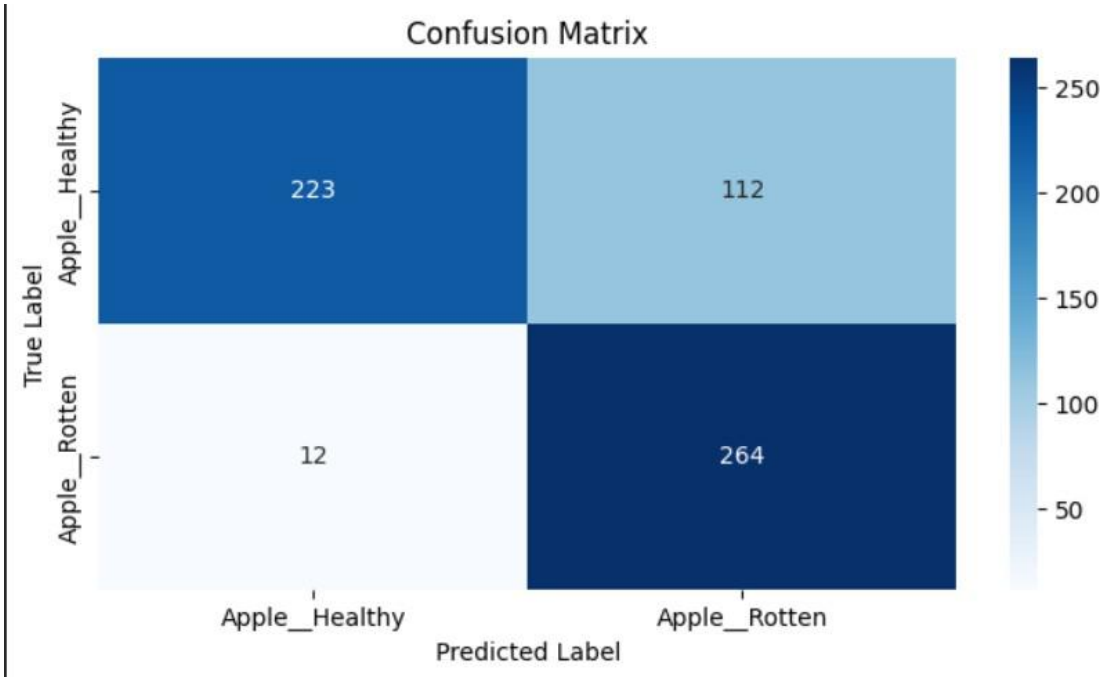


Figure 4.6: Confusion Matrix

Precision and Recall: These metrics helped assess how well the model correctly identifies fresh vs. rotten apples and avoids misclassifications.

F1-Score: An essential metric that balances both precision and recall, ensuring that the model is not biased towards one class.

Table 4.1: Model Evalution Table

Metric	Apple_Healthy	Apple_Rotten	Macro Avg	Weighted Avg	Overall Accuracy
Precision	0.95	0.70	0.83	0.84	-
Recall	0.67	0.96	0.81	0.80	-
F1-Score	0.78	0.81	0.80	0.79	-
Support	335	276	611	611	-
Accuracy	-	-	-	-	0.80 (80%)

4.5 Integration with Flutter App

4.5.1 TensorFlow Lite Model Conversion

After training, the MobileNetV2 model was converted into TensorFlow Lite format, making it compatible for deployment on mobile devices. This conversion reduces the size of the model and optimizes it for faster performance on smartphones.

4.5.2 Real-time Integration in Flutter App

The TensorFlow Lite model was integrated into the Harvest Direct Flutter app to provide real-time predictions. Users can upload images of apples, and the app uses the trained model to classify whether the apple is fresh or rotten.
Real-Time Classification: The model provides results in real-time, showing whether the apple is fresh or spoiled, allowing buyers to make informed decisions before purchasing.

4.6 Backend Development with Firebase

4.6.1 Firebase Authentication

Firebase Authentication was used to handle user authentication in the app. Users can sign up and log in using email/password or Google Sign-In for ease of use. The authentication process ensures that only registered users can list products or place orders, providing a secure environment.

4.6.2 Firestore Database

We used Firebase Firestore for real-time data storage, which handles user data, product listings, orders, and freshness checks. Firestore allows seamless synchronization across all devices, ensuring that users see the most up-to-date information.

- Users Collection: Stores user information (farmers and buyers).
- Products Collection: Contains data about the products listed by farmers.
- Orders Collection: Tracks orders placed by buyers.

4.6.3 Firebase Storage

Firebase Storage was used to store product images and freshness detection images uploaded by users. This scalable storage solution ensures quick retrieval of images when browsing products or analyzing apple freshness.

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

The Harvest Direct mobile app was developed using Flutter, a cross-platform development framework. Flutter was chosen for this project because it allows for the development of apps for both Android and iOS from a single codebase, making it easier and faster to deploy across multiple platforms. The app's User Interface (UI) was designed with simplicity and ease of use in mind, particularly targeting farmers who may not be familiar with complex mobile applications. The UI follows a minimalistic design philosophy, making it easy to navigate and ensuring that even users with limited technical experience can operate the app without difficulty.

Table 4.2: UI and UX Design Table

Feature	Description	Target User
Role Selection	Users select either "Buyer" or "Farmer" roles, which tailors the app's features accordingly.	Both Buyers & Farmers
Farmer Dashboard	A central hub for farmers to manage products, track orders, and view sales analytics.	Farmers
Product Listings	Buyers can browse products, filter by freshness, price, and other criteria.	Buyers
Product Detail Page	Detailed product information with freshness analysis and pricing. Includes the option to add to cart.	Buyers
AI Freshness Detection	AI-powered freshness score for products based on image analysis.	Both Buyers & Farmers
Order Management	A comprehensive view for farmers to track and manage their orders (processing, shipped, delivered).	Both Buyers & Farmers
Chatbot Assistance	An AI-based chatbot that provides real-time assistance with common queries and order issues.	Both Buyers & Farmers

CHAPTER 5

RESULTS AND DISCUSSIONS (or USER MANUAL)

5.1 Front-End (Flutter)

The front-end of the Harvest Direct mobile application was developed using Flutter, a cross-platform development framework. Flutter's ability to compile to both Android and iOS ensures broad accessibility and a consistent user experience across platforms. The UI was designed to be user-friendly, with minimalistic elements to assist farmers who may not have prior experience using complex mobile applications. The interface includes large text, easy navigation, and intuitive icons. The app provides several key screens, such as onboarding, sign-up/login, role selection, and product browsing, all designed to ensure a smooth flow of operations for both farmers and buyers. The AI Freshness Detection Screen utilizes a machine learning model to analyze apple images, providing a freshness score that is easily understandable. The app's user interface also features seamless integration with a chatbot for real-time support, and a dashboard to manage orders and product listings, empowering farmers to efficiently manage their business. Flutter ensured that the development process was streamlined, providing a high-quality, responsive, and visually appealing experience.

5.1.1 Splash Screen

Harvest Direct seamlessly connects local apple farmers with quality-conscious buyers. With our AI-powered freshness detection and secure transactions, we ensure the freshest produce and a smooth, trusted shopping experience. Get ready to explore a new way of buying fresh, directly from the source.

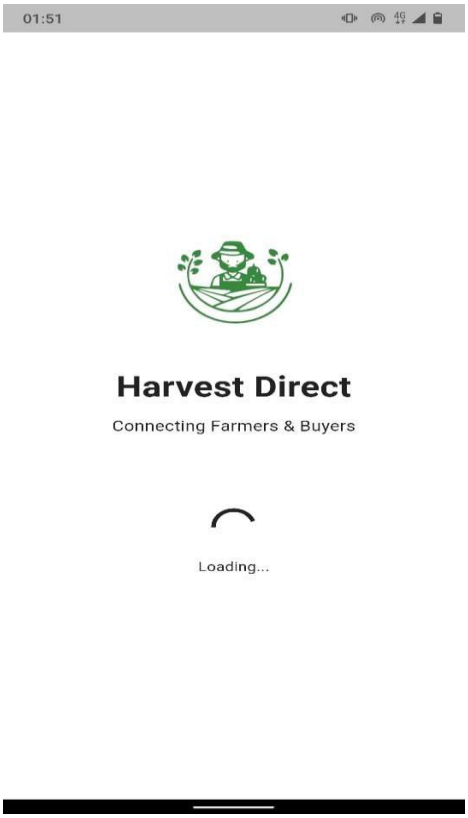


Figure 5.1: Splash Screen

5.1.2 On-Boarding Screens

Harvest Direct bridges the gap between fresh apple farmers and quality-conscious buyers through a direct marketplace. Find local farmers, communicate directly with sellers, and build trusted relationships for a seamless buying experience. Shop with confidence thanks to secure transactions, bank-grade security, and 24/7 multilingual support. Our AI-powered freshness detection guarantees you get the freshest apples every time, with real-time quality scanning and instant freshness scoring for peace of mind.

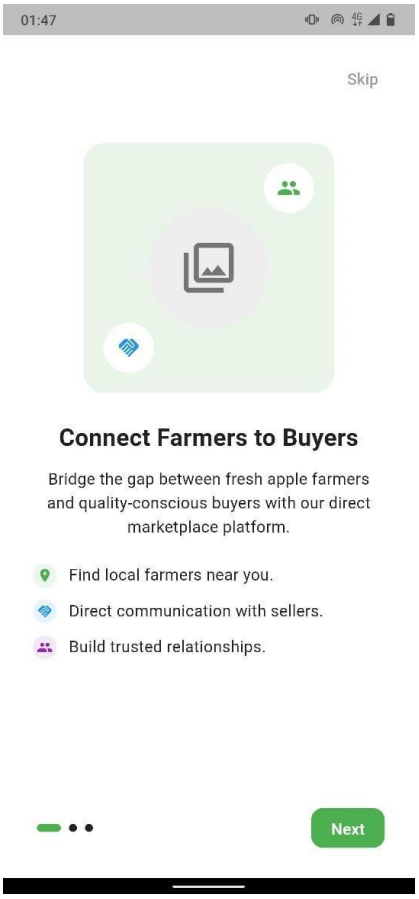


Figure 5.2: On-Boarding Screen 1

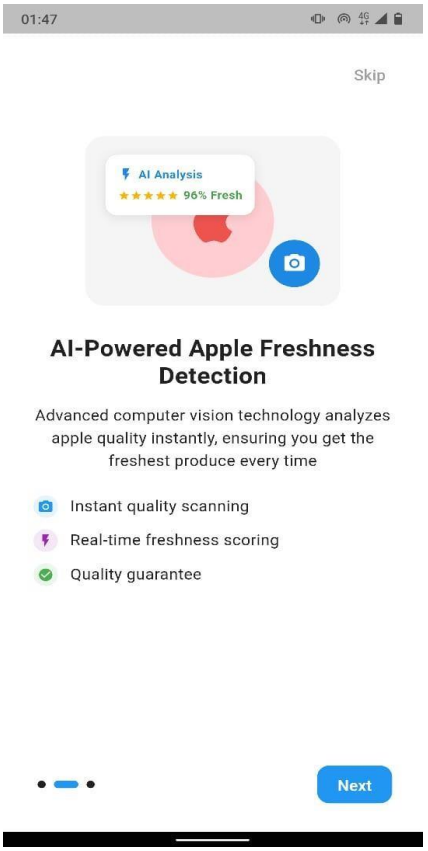


Figure 5.3:On-Boarding Screen 2

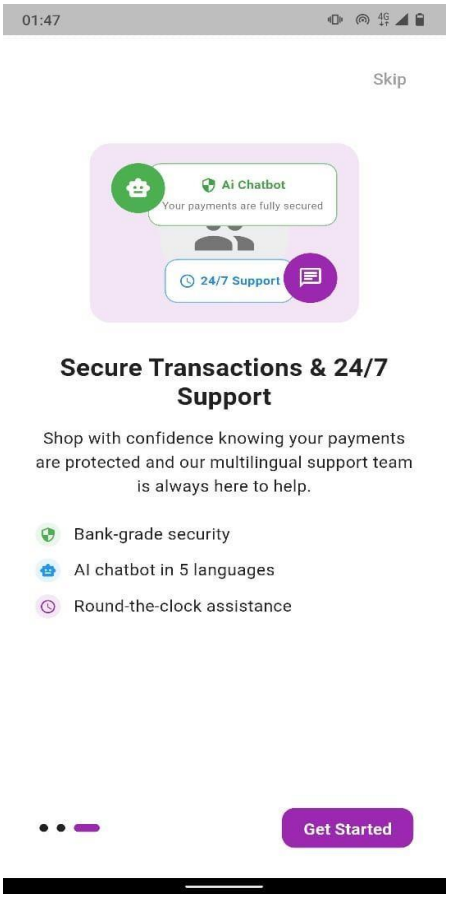


Figure 5.4: On-Boarding Screen 3

5.1.3 Sign Up And Sign In Screen

Welcome to Harvest Direct! To get started, sign in with your existing account or easily create a new one by entering your full name, email, and password. With secure sign-in options, you can also sign up or log in using your Google account. Once logged in, you can explore fresh apples directly from farmers ensuring a seamless and secure shopping experience. Simply enter your details and start your journey with us today!

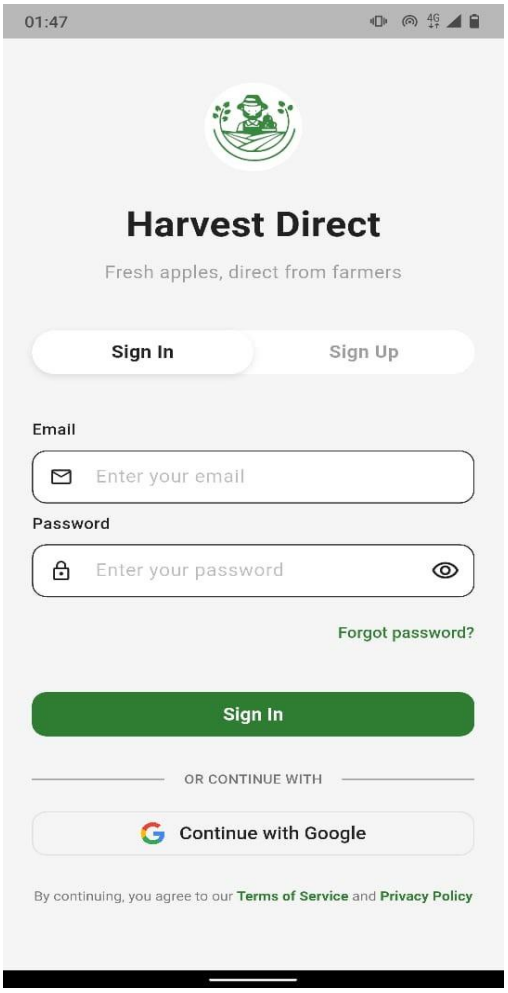


Figure 5.5: Sign In Screen

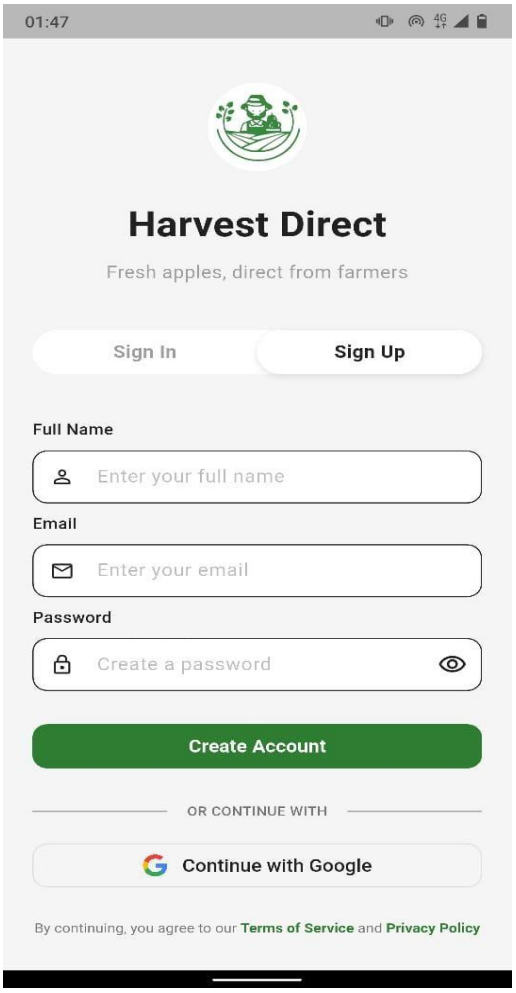


Figure 5.6: Sign Up Screen

5.1.4 Forget Password Screen

If you've forgotten your password, don't worry! Simply enter the email address associated with your account, and we'll send you a link to reset your password. Be sure to check your spam folder if you don't see the email. The reset link will expire in 1 hour, so make sure to act quickly. If you continue to have trouble, feel free to contact our support team for assistance.

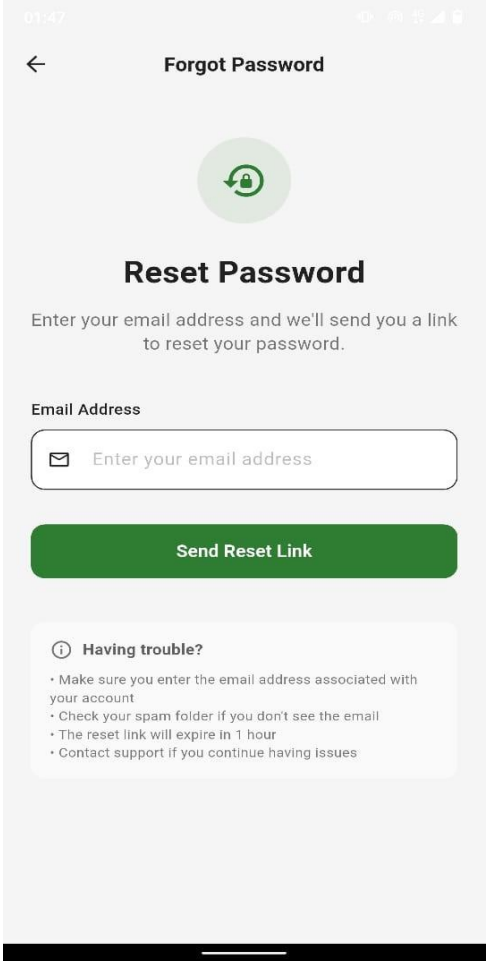


Figure 5.7 Forget Password Screen

5.1.5 Role Selection Screen

Select how you would like to use Harvest Direct! As a Buyer, you can browse fresh apple varieties directly from local farmers, check AI-verified freshness scores, communicate directly with farmers, and enjoy secure payment and delivery. As a Farmer, you can list your apple varieties, access AI quality verification, use order management tools, and track sales analytics to maximize profits and build lasting customer relationships. Choose your role and start exploring.

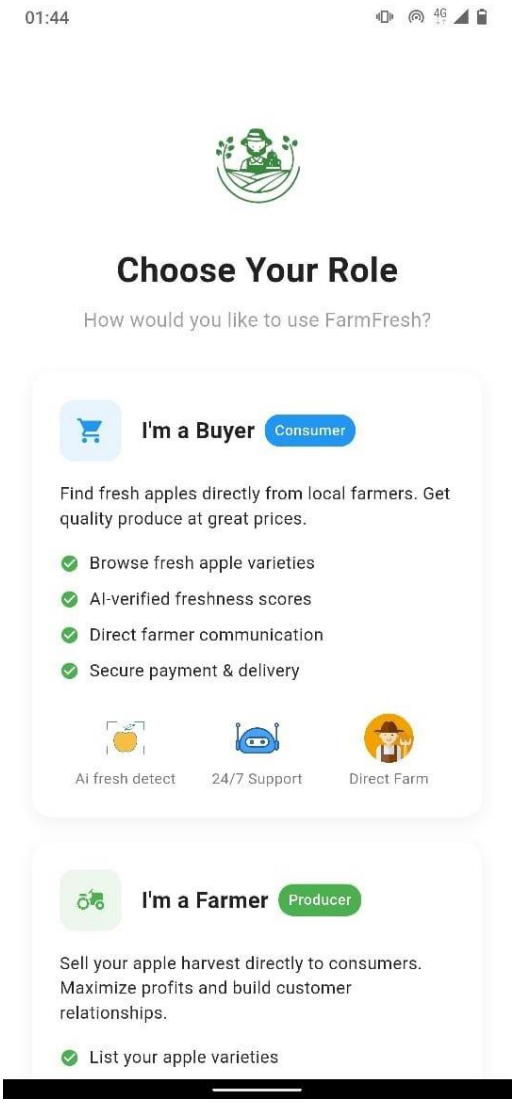


Figure 5.5: Role Selection Screen

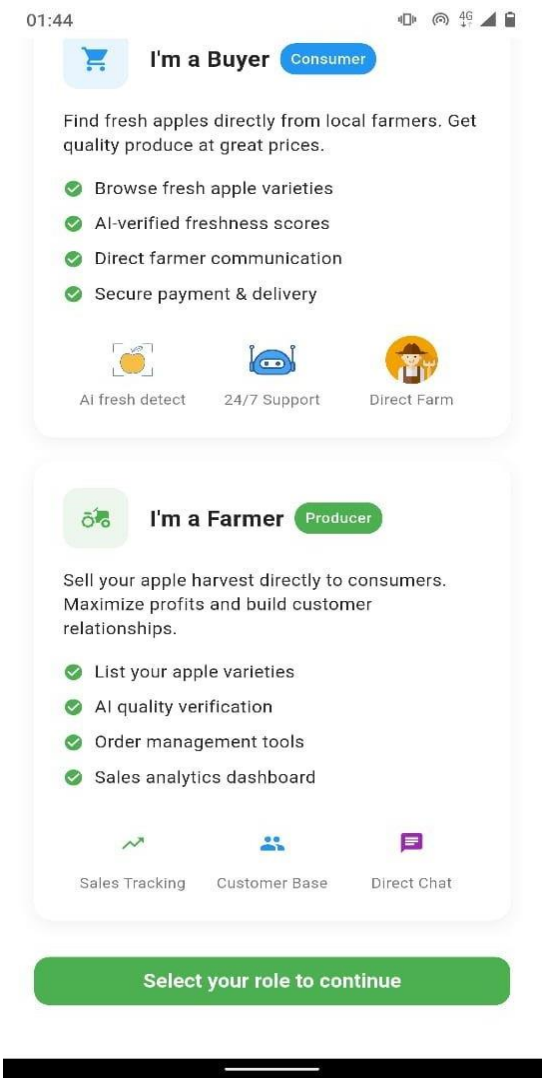


Figure 5.6: Role Selection Screen

5.1.6 Farmer Dashboard Screen

The Farmer Dashboard provides you with all the essential tools to manage your apple sales efficiently. Track your total revenue and monitor active orders in real-time. Easily manage and edit your top-selling products, such as fresh apples and other produce, with details on pricing and sales. You can also view and manage draft products, ensuring you're always ready to list new items for sale. The dashboard is designed to help farmers stay on top of their business and grow their customer base seamlessly.

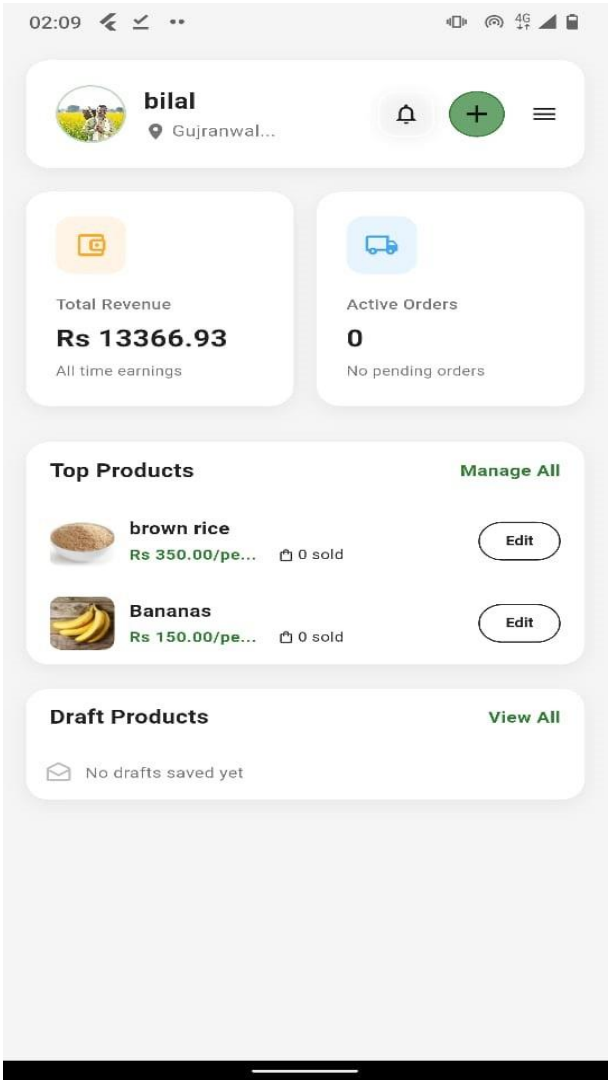


Figure 5.6: Farmer DashBoard Screen

5.1.7 Published Product Screen

The Published Product Screen allows you to easily add new products to your farm's listing. You can provide a detailed description of the product, select the farming method (Organic or Conventional), and specify the farm location. Add certifications like USDA Organic or Non-GMO to increase your product’s credibility. After reviewing the product details and price, you can either publish the product immediately or save it as a draft for later editing, giving you full control over your product listings.

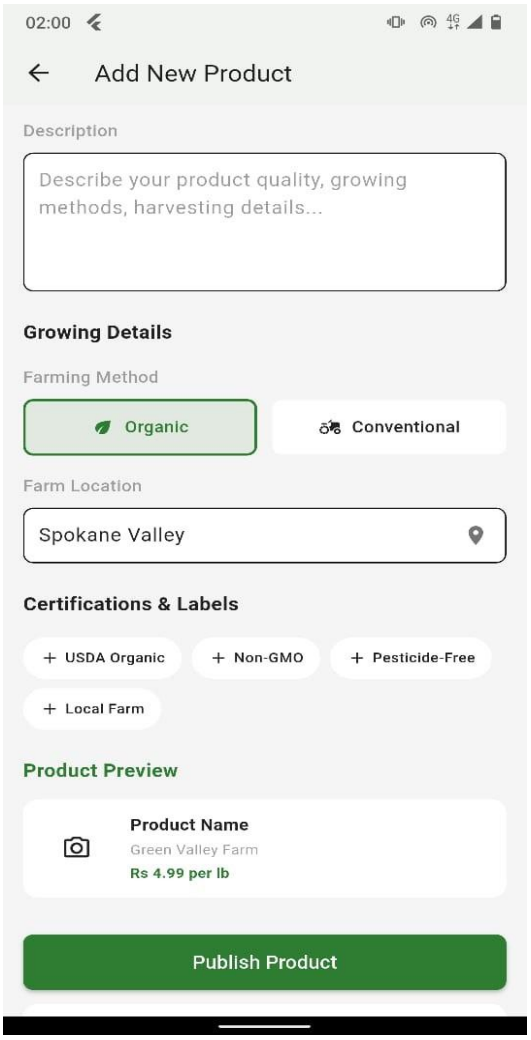


Figure 5.11: Product Upload Screen

5.1.8 AI Freshness Detection Screen

The AI Freshness Detection feature allows you to assess the quality of your apples using advanced computer vision. By capturing a clear image of the apple, the system performs two types of analysis: Visual Analysis, checking for color, shape, and spots, and Surface Quality, detecting texture and blemishes. Once the scan is complete, the app provides a quality grade and detailed analysis, helping you determine whether the apple meets your desired freshness standards before listing it for sale.

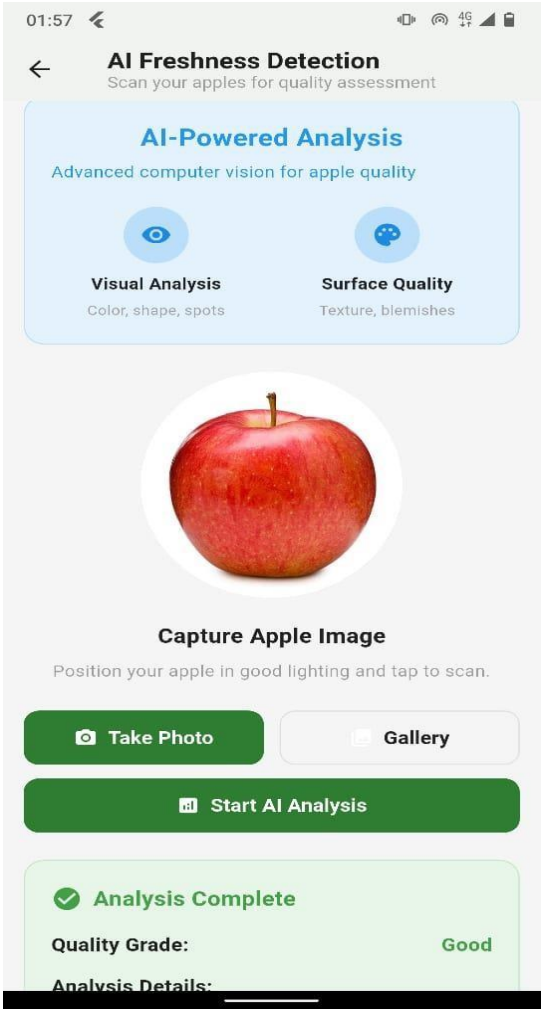


Figure 5.12: AI Freshness Detection Screen

5.1.9 AI Chatbot Screen

The AI Chatbot is available 24/7 to assist you with various needs. Whether you need help tracking your order, understanding apple quality, resolving payment issues, or getting delivery support, the AI chatbot offers a quick solution. It can also provide farming tips and help you report issues, ensuring you receive timely assistance in any situation. Simply choose a topic and start chatting with our AI assistant to get the support you need.

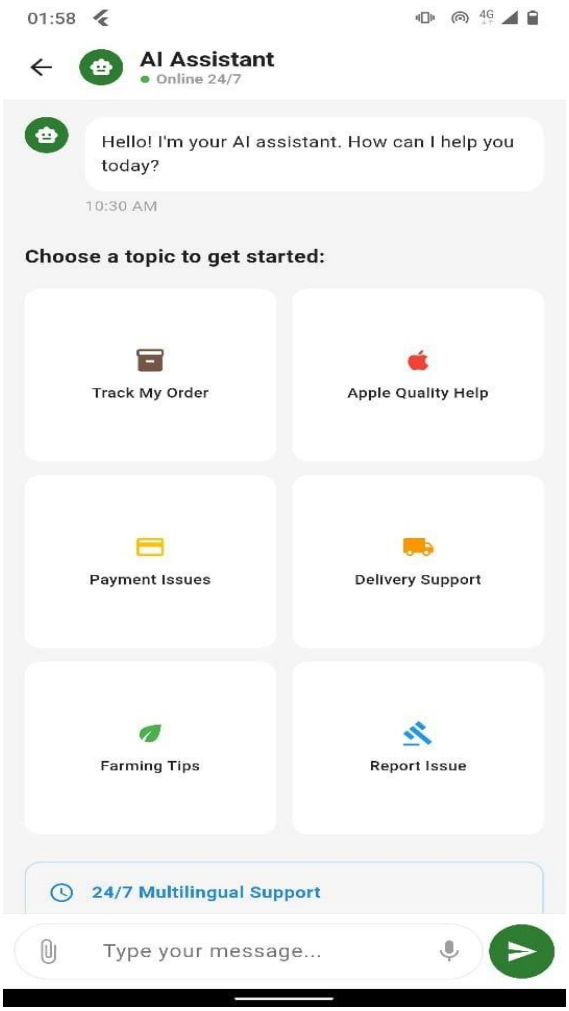


Figure 5.13: AI Chatbot Screen

5.1.10 Order Updates Screen

The Order Updates Screen provides a clear overview of your recent orders. You can track the status of each order, whether it has been Delivered or Cancelled, along with the customer details and the total amount. The screen makes it easy to monitor each transaction, ensuring that you stay up to date on the status of your products. With visual indicators for delivered and cancelled orders, it helps you manage and fulfill orders effectively.

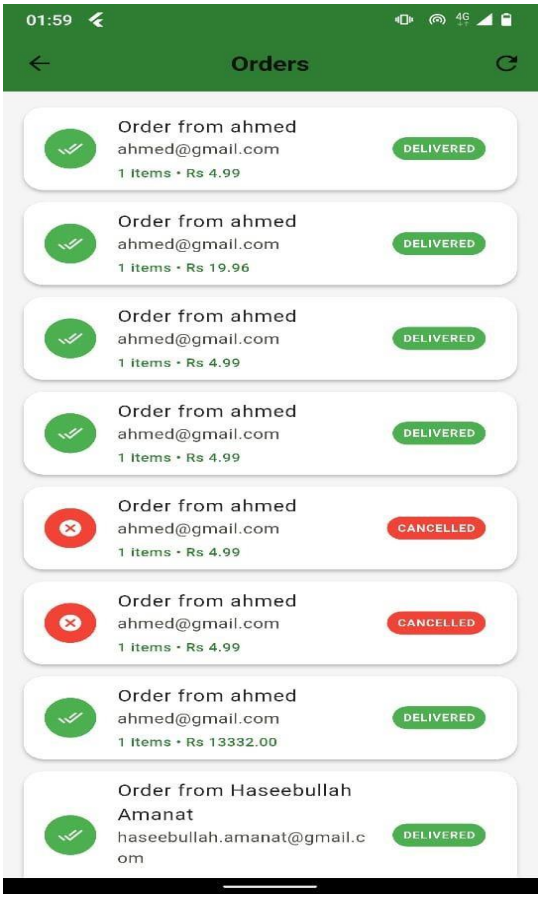


Figure 5.14: Order Updates Screen

5.1.11 Notification Screen

The Notification Screen keeps you updated with real-time alerts on new orders and other important updates. You'll receive notifications whenever a new order is placed, including details like the customer's name, the product ordered, and the total amount. You can easily view and manage your notifications, ensuring you're always informed about your business activities. With the option to mark all notifications as read, this screen helps you stay organized and responsive.

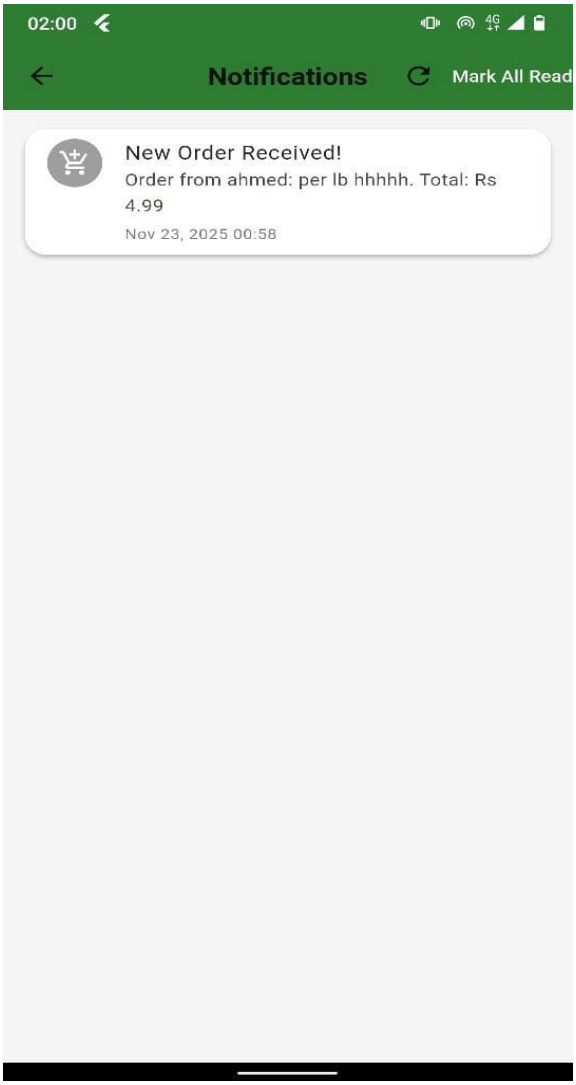


Figure 5.15: Notification Screen

5.1.12 Draft Products Screen

The Draft Products Screen allows you to save products before officially listing them. You can easily view and manage saved draft products, including details like pricing and product name. If you wish to make changes, simply tap the Edit button, or delete the product if it’s no longer needed. Once finalized, you can publish the product, making it available for sale on the platform.

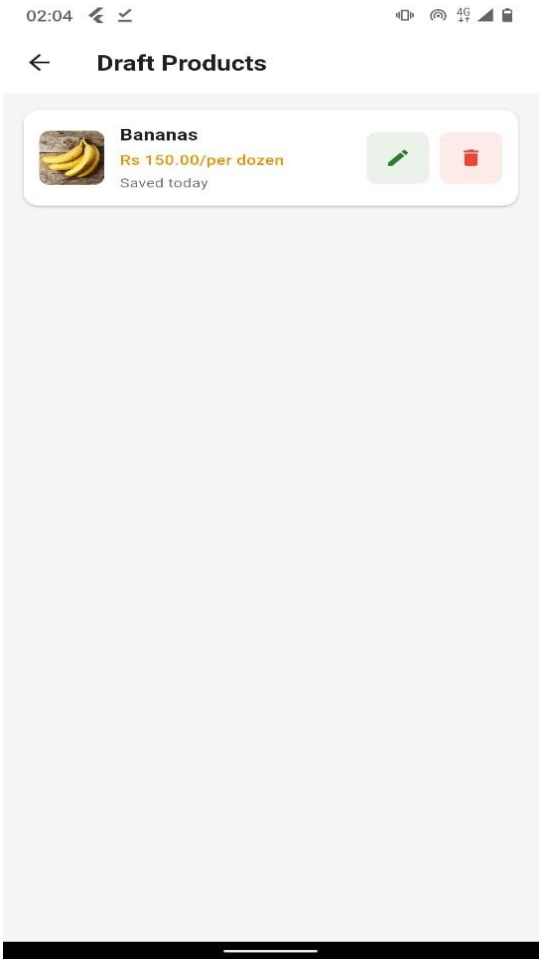


Figure 5.16: Draft Product Screen

5.1.13 Farmer Profile And Setting Screen

The Farmer Profile & Settings Screen allows you to manage your personal and account information. You can view and edit your full name, phone number, and location, ensuring your details are up to date. The screen also displays your user role (e.g., Farmer), which determines the features available to you. Additionally, you can update your role and explore app information, such as version, build, and package name, to stay informed about your app settings.

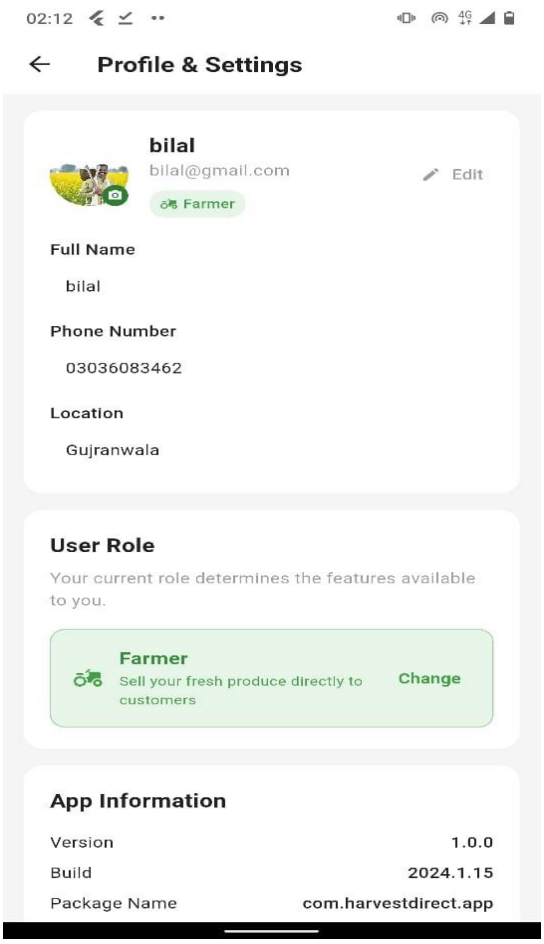


Figure 5.17: Farmer Profile and Setting Screen

5.1.14 User dashboard screen

The User Dashboard allows you to explore fresh produce available for purchase. You can easily browse products by category, including vegetables, fruits, grains, and herbs. Featured products are highlighted for quick access, and you can see details like pricing and freshness status. With the Fresh Today section, you can access farm-to-table produce within 24 hours. The dashboard makes it simple to search for your desired items and add them to your cart for a smooth shopping experience.

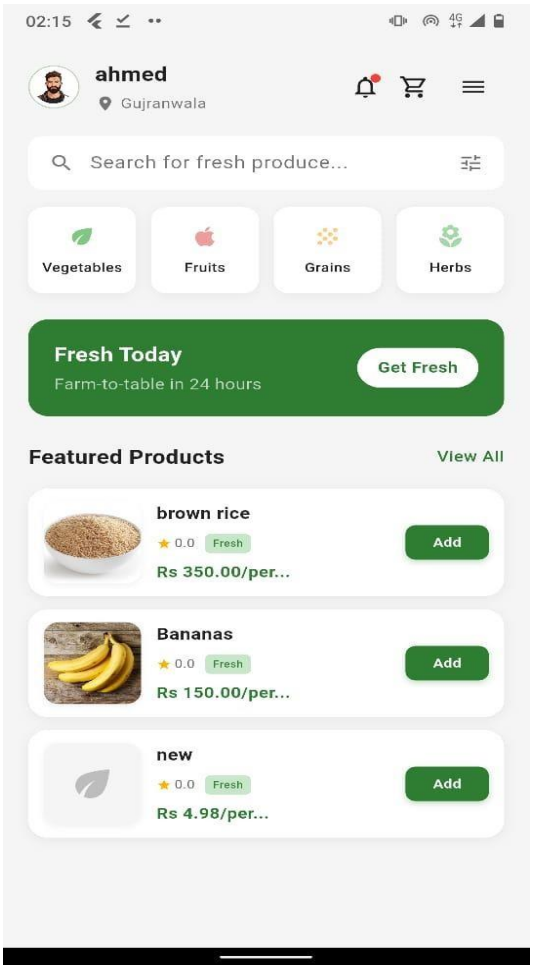


Figure 5.18: User DashBoard Screen

5.1.15 User profile and setting screen

The User Profile & Settings Screen allows you to view and update your personal information, including your full name, phone number, and location. You can also manage your user role, which determines the features available to you on the platform. For example, as a Buyer, you can purchase fresh produce directly from local farmers. Additionally, the screen displays app version details and other technical information to keep you informed about your app's status.

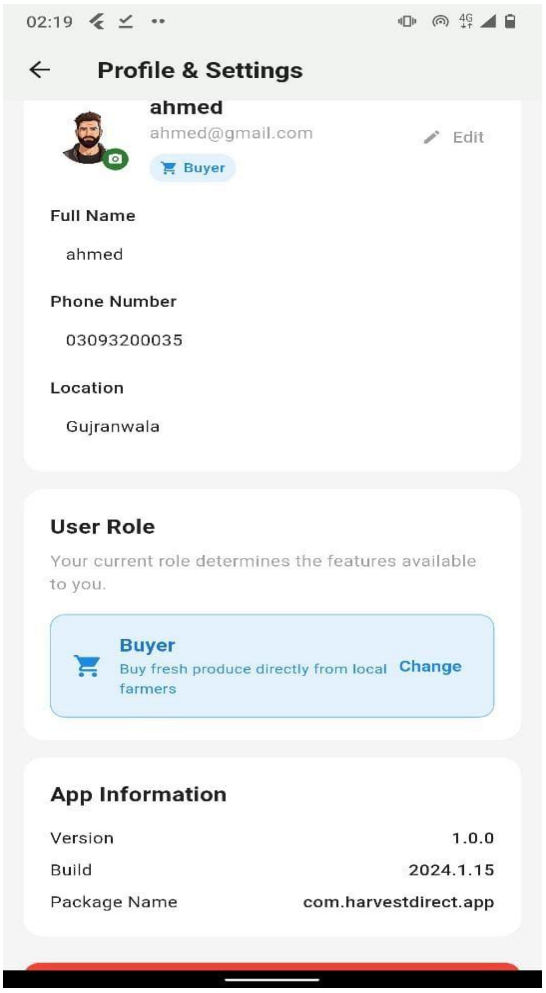


Figure 5.19: User Profile and Setting Screen

5.1.16 Showing all products screen

The All Products Screen displays a comprehensive list of all available products, allowing users to browse items by category, such as grains, fruits, herbs, and vegetables. Each product is showcased with essential details like the seller's name, product price, and freshness percentage. You can easily add products to your cart for purchase, making it simple to find fresh and local produce. The screen also includes filters to help narrow down your search, ensuring you find exactly what you need.

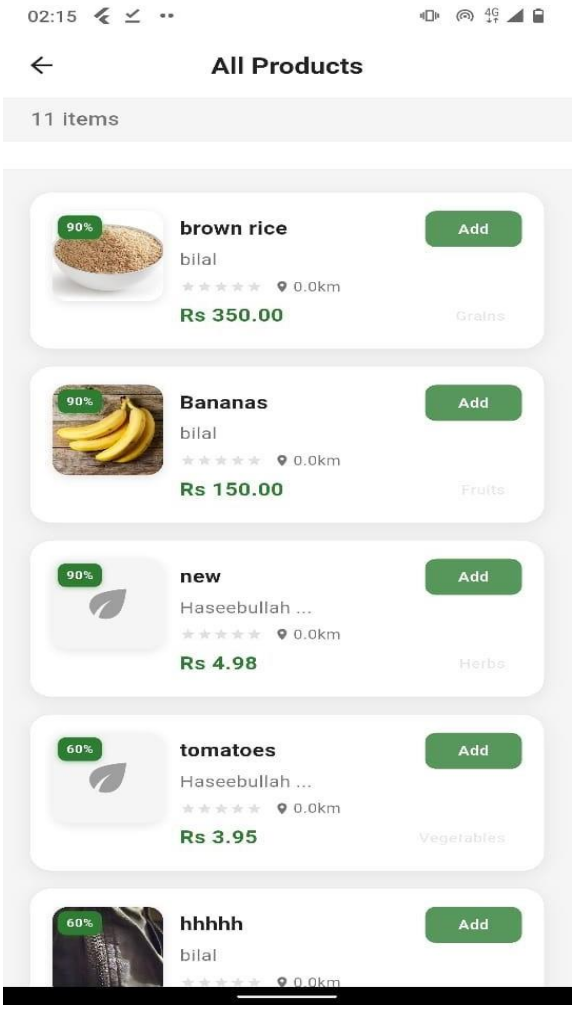


Figure 5.20: Showing all Products Screen

5.1.17 Shopping cart screen

The Shopping Cart Screen provides an overview of all the items you've added to your cart. You can adjust the quantity of each product by tapping the "+" or "-" buttons, and the total cost is updated in real-time. The screen shows the item details, such as price per unit, and the total for all products. Once you've reviewed your selections, you can proceed to checkout to complete your purchase.

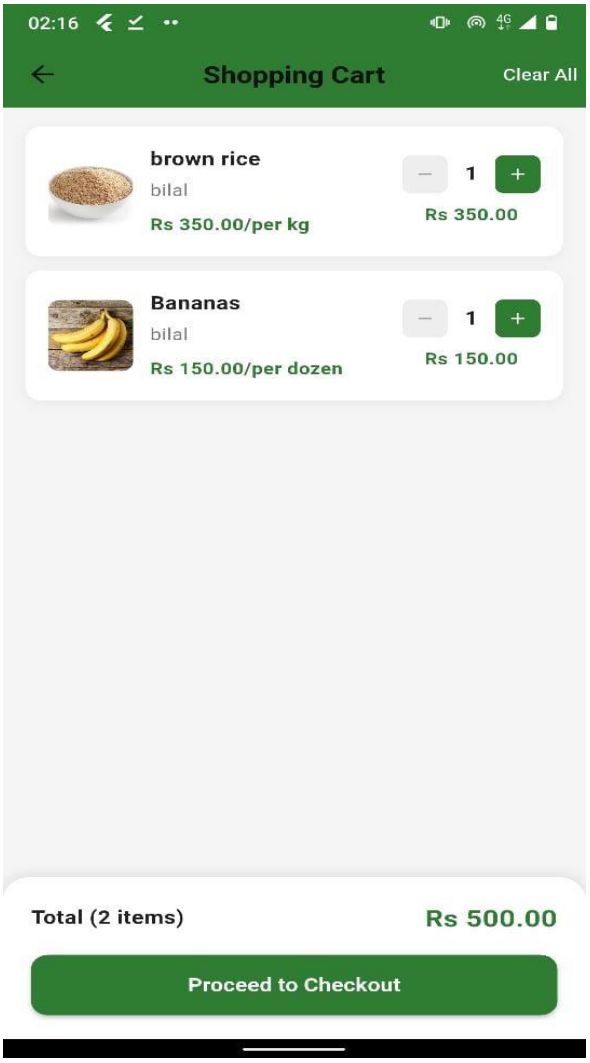


Figure 5.21: Shopping Cart Screen

5.1.18 Product Checkout Screen

The Product Checkout Screen provides a detailed order summary, including the items purchased, their quantities, and individual prices. The subtotal is calculated along with applicable taxes and shipping charges, giving a clear view of the total amount due. Below, users can enter shipping information such as name, email, phone number, and address to complete the purchase. Once everything is confirmed, users can proceed to finalize the order, ensuring a seamless checkout experience.

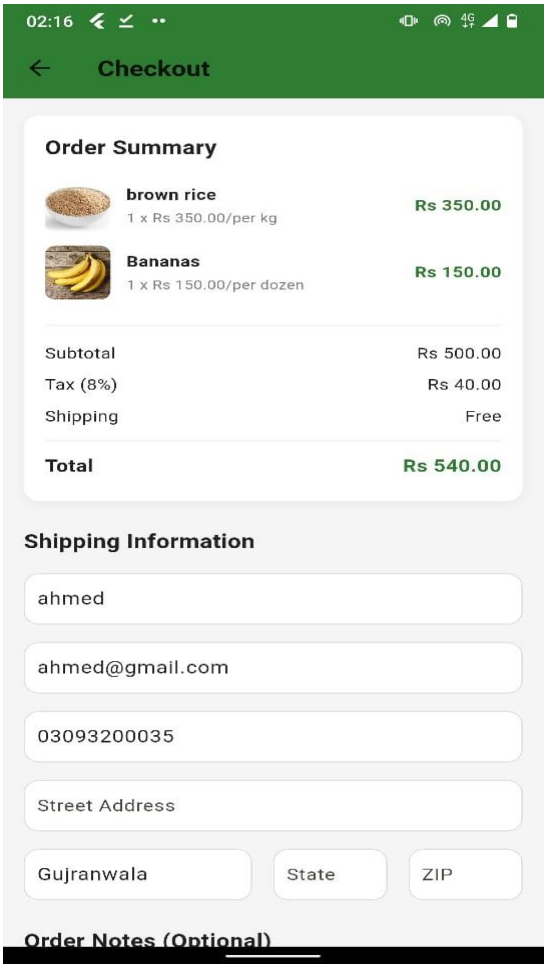


Figure 5.22:Product Checkout Screen

5.2 Back-End Development

The backend of Harvest Direct is powered by Firebase, providing a robust solution for authentication, real-time data management, and cloud storage. Firebase Authentication handles secure logins for both farmers and buyers, allowing for multiple authentication methods, including email/password and Google Sign-In. The real-time database solution, Firestore, allows for seamless synchronization of data, ensuring that product listings, orders, and user interactions are instantly reflected across all devices. Farmers can upload product images, which are stored in Firebase Storage, and linked to their product listings. Additionally, the backend integrates an AI-powered freshness detection system, utilizing a MobileNetV2 model for real-time image analysis of apple freshness. The backend is designed for scalability, handling an increasing number of users and transactions. Firebase Cloud Messaging enables real-time notifications for

updates on orders, product listings, and other important actions, keeping users informed and engaged. The integration of Firebase ensures the app's functionality remains secure, fast, and responsive, offering a seamless experience for both buyers and farmers.

CHAPTER 6

CONCLUSION AND RECOMMENDATIONS

6.1 Conclusion

The "Harvest Direct" mobile application has successfully tackled several pressing challenges in the agricultural sector, particularly the inefficiencies in the supply chain and the disconnect between farmers and buyers. By eliminating intermediaries, the platform ensures that farmers receive fairer prices for their produce, while buyers benefit from fresher products at lower costs. The integration of an AI-powered freshness detection system, specifically designed to assess the quality of apples, sets the app apart by offering buyers an innovative way to verify product freshness before purchase. This technology, coupled with a user-friendly interface, has made the app accessible to farmers with limited technological literacy, empowering them to list products, set prices based on quality, and manage their orders with ease.

The secure transaction system powered by Firebase and the integration of a 24/7 AI chatbot ensure that both farmers and buyers have continuous support, enhancing trust and reliability. The app's flexibility, including its ability to cater to both individual and business buyers, further strengthens its value proposition. Overall, "Harvest Direct" is a pivotal solution that supports fair trade practices, improves the livelihoods of farmers, reduces food waste, and promotes sustainable farming. The application serves as a critical step toward modernizing the agricultural marketplace and can serve as a foundation for further development and expansion within the sector.

6.2 Recommendations And Future Work

Building on the current success of "Harvest Direct," we recommend the following strategic improvements and developments for the future growth of the application:

1. **Enhance AI Freshness Detection:**

Expand the AI-powered freshness detection to include a broader range of produce, such as vegetables and other fruits. Improve the model's ability to detect spoilage in various conditions, including low-quality images, poor lighting, and different angles.

2. **Integrate Additional Payment Methods:**

Add more payment gateways, such as mobile wallets (e.g., Easypaisa, JazzCash) and direct bank transfers, to provide users with flexible and secure payment options.

3. **Develop Offline Functionality:**

Implement offline capabilities for areas with limited internet access, enabling farmers to manage their listings, orders, and product details even without stable connectivity.

4. **Implement Logistics and Delivery Tracking:**

Introduce a logistics module that includes real-time order tracking and route optimization to streamline delivery and improve the order fulfillment process for both farmers and buyers.

5. **Support Multilingual Capabilities:**

Expand the app's language support to cater to a more diverse user base, enabling farmers and buyers from various linguistic backgrounds to fully utilize the platform.

REFERENCES

[1] Mufti, A., Novianti, D., & Anjani, D. (2017). Designing mobile farmer application using object oriented analysis and design. Register: Jurnal Ilmiah Teknologi Sistem Informasi, 3(2), 105-113.

[2] Singh, D., Shehrawat, P. S., Godara, A. K., & Kumari, N. (2024). A study of farmers perception about agricultural mobile apps.

[3] Patel, H., & Patel, D. (2016). Survey of android apps for agriculture sector. International Journal of Information Sciences and Techniques, 6(1-2), 61-67.

[4] Kaur, S., & Dhindsa, K. S. (2017, December). Comparative study of android-based M-Apps for farmers. In International Conference on Intelligent Computing and Applications: ICICA 2016 (pp. 173-183). Singapore: Springer Singapore.

[5] Khan, Sheema & Parihar, Poonam. (2024). Review on Use of Mobile Applications in Digital Agriculture. Journal of Community Mobilization and Sustainable Development. Volume 1. 253-258.

[6] Yadav, A., Thilagam, P. and Singh, S., 2023. Mobile applications for agricultural transformation: Types, impacts, case studies, and recommendations.

[7] Kambale, Parashuram & M, Dharma & Patil, Dayananda & N R, Ganavi. (2024). Mobile Technology for Farmers: An Overview of Agricultural Apps. Asian Journal of Agricultural Extension, Economics & Sociology. 42. 75-81. 10.9734/ajaees/2024/v42i92543.

[8] Babashli, Bakhtiyar. (2022). Smart Farming Applications on Agriculture. 2.

[9] Firame, S., Dhamale, A., Yerme, A., Kote, K., & Shinde, S. (2024). A Survey Paper On: Uplifting Farmers with Mobile Applications: An Overview of Modern Agricultural Utilities. International Journal of Innovative Science and Research Technology, 9 (3), 368-371.

[10] Krishna, R. V. C., & Adaickalam, V. (2022). An Android App for Farm: A Survey. * International Journal of Science, Engineering and Technology, 10 (2), 1-5.

[11] Das, B., Hoque, A., Roy, S., Kumar, K., Laskar, A. A., & Mazumder, A. S. (2025). Post-Harvest Technologies and Automation: Al-Driven Innovations in Food Processing and Supply Chains. Int. J. Sci. Res. Sci. Technol, 12(1), 183-205.

[12] Mallegowda, M., Sanskar, R. G., VISHVESHWARA, N., Safwan, G. A., & Vivek, J. (2024, August). Fruit classification based on freshness. In 2024 International Conference on Emerging Techniques in Computational Intelligence (ICETCI) (pp. 373-381). IEEE.

[13] Sathiyasuntharam, V., Navaneethan, S., Grewal, A., Sandhu, I., & Mangai, R. A. (2025, March). Smart Harvesting Solutions: Robotics in Fruit and Vegetable Harvesting for Reduced Labor Dependency. In 2025 International Conference on Intelligent Computing and Control Systems (ICICCS) (pp. 31-36). IEEE.

[14] Muttaqin, I. F., & Arifudin, R. (2024). Fruit Freshness Detection Using Android-Based Transfer Learning MobileNetV2. Recursive Journal of Informatics, 2(1), 8-17.

[15] Tata, J. S., Kalidindi, N. K. V., Katherapaka, H., Julakal, S. K., & Banothu, M. (2022, August). Real-time quality assurance of fruits and vegetables with artificial intelligence. In Journal of Physics: Conference Series (Vol. 2325, No. 1, p. 012055). IOP Publishing.

[16] Shriram, P., & Mhamane, S. (2018, November). Android app to connect farmers to retailers and food processing industry. In 2018 3rd international conference on inventive computation technologies (ICICT) (pp. 284-287). IEEE.

[17] Mallegowda, M., Sanskar, R. G., VISHVESHWARA, N., Safwan, G. A., & Vivek, J. (2024, August). Fruit classification based on freshness. In 2024 International Conference on Emerging Techniques in Computational Intelligence (ICETCI) (pp. 373-381). IEEE.

[18] Wang, Z., Koirala, A., Walsh, K., Anderson, N., & Verma, B. (2018). In field fruit sizing using a smart phone application. Sensors, 18(10), 3331.

[19] Kodors, S. (2020, January). Pear and apple recognition using deep learning and mobile. In 19th International Scientific Conference Engineering for Rural Development Proceedings.

[20] Aherwadi, N., Mittal, U., Singla, J., Jhanjhi, N. Z., Yassine, A., & Hossain, M. S. (2022). Prediction of fruit maturity, quality, and its life using deep learning algorithms. Electronics, 11(24), 4100.

[21] Aksoy, S., Demircioglu, P., & Bogrekci, I. (2025). Web-Based AI System for Detecting Apple Leaf and Fruit Diseases. AgriEngineering, 7(3).

[22] Patidar, A., & Chakravorty, A. (2024). Using machine learning to identify diseases and perform sorting in apple fruit. International Journal of Innovative Science and Research Technology.

[23] Chen, M. C., Cheng, Y. T., & Liu, C. Y. (2022). Implementation of a Fruit Quality Classification Application Using an Artificial Intelligence Algorithm. Sensors & Materials, 34.

[24] Singh, P., Kumar, A., & Raj, G. (2023, May). Detection of Apple Fruit Diseases for On-Tree Apples Using Machine Learning. In Proceedings of the KILBY 100 7th International Conference on Computing Sciences.

[25] S SM, S. B., Kanavi, A., Kuradagi, A., & Pendem, S. (2024). Improving apple fruit quality detection with ai and machine vision. Int. J. Multidiscip. Res, 6, 1-27.

[26] Masparudin, M., Fitri, I., & Sumijan, S. (2024). Development of apple fruit classification system using convolutional neural network (cnn) mobilenet architecture on android platform. Sistemasi: Jurnal Sistem Informasi, 13(1), 230-243.

[27] Leonid, T. T., & Nanthine, T. (2023, April). Fruit quality detection and classification using Computer Vision Techniques. In 2023 Eighth International Conference on Science Technology Engineering and Mathematics (ICONSTEM) (pp. 1-6). IEEE.

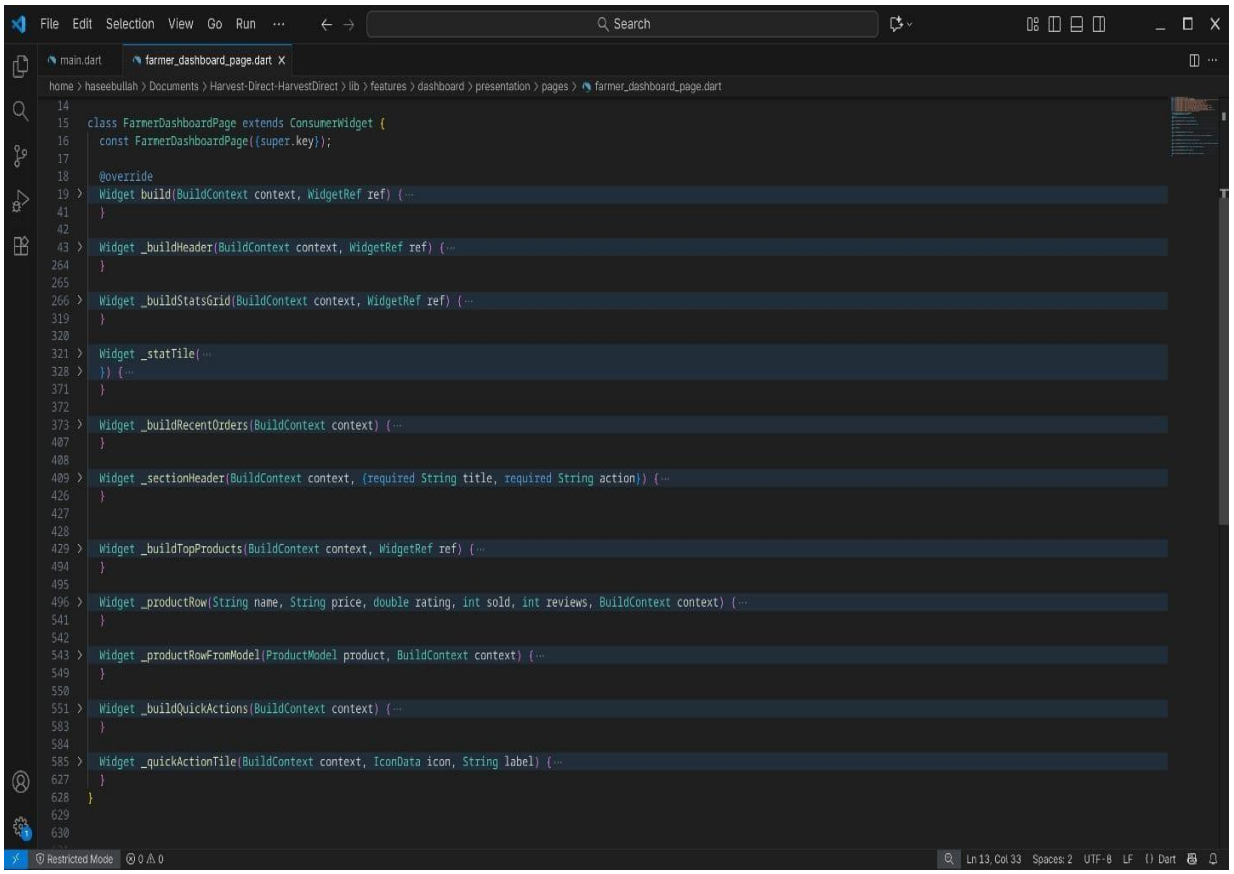
[28] Tata, J. S., Kalidindi, N. K. V., Katherapaka, H., Julakal, S. K., & Banothu, M. (2022, August). Real-time quality assurance of fruits and vegetables with artificial intelligence. In Journal of Physics: Conference Series (Vol. 2325, No. 1, p. 012055). IOP Publishing.

[29] Manzoor, E. S., Malhotra, R., Bhat, R., & Shekhar, S. (2024, July). Apple Detection: A CNN Approach for Diseases Detection. In 2024 Second International Conference on Advances in Information Technology (ICAIT) (Vol. 1, pp. 1-5). IEEE.

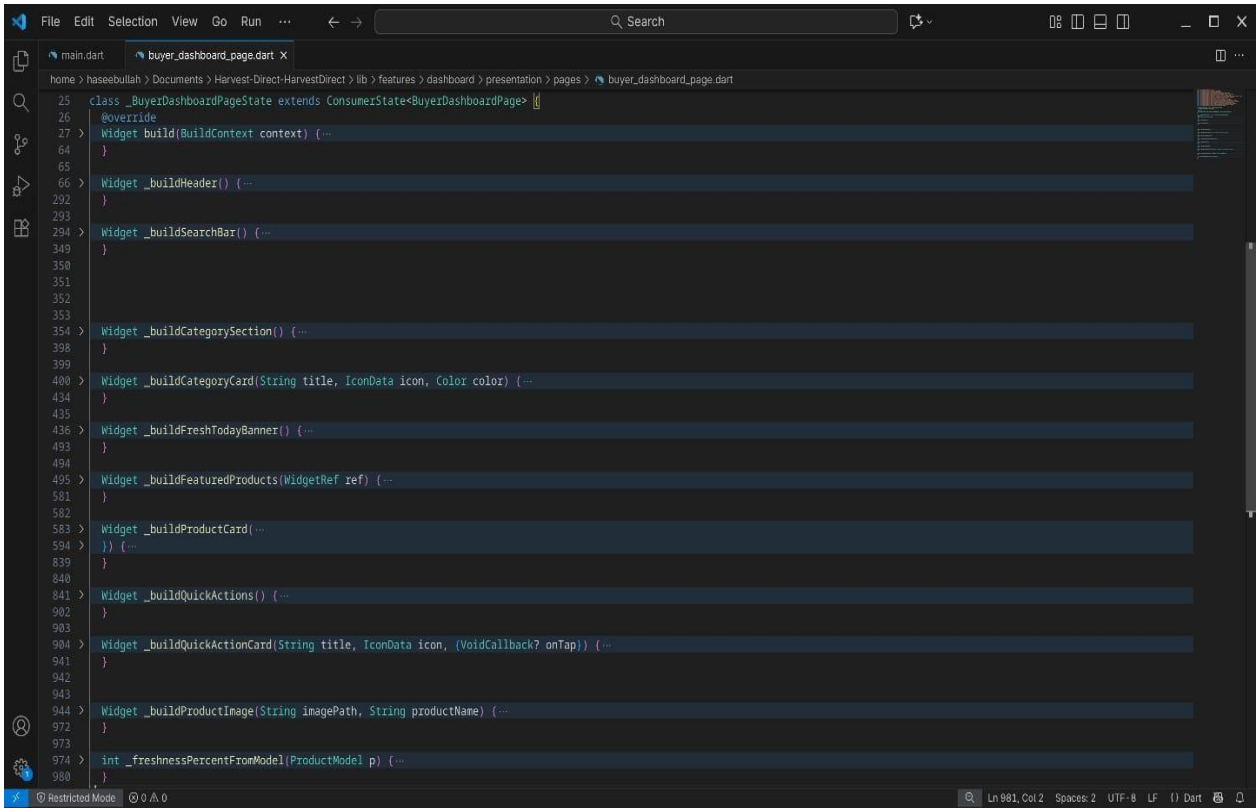
[30] Ramya, G., & Dinesh, S. (2017). A Computer Vision Based Diseases Detection and Classification in Apple Fruits. IJERT, 6, 161-164.

APPENDICES

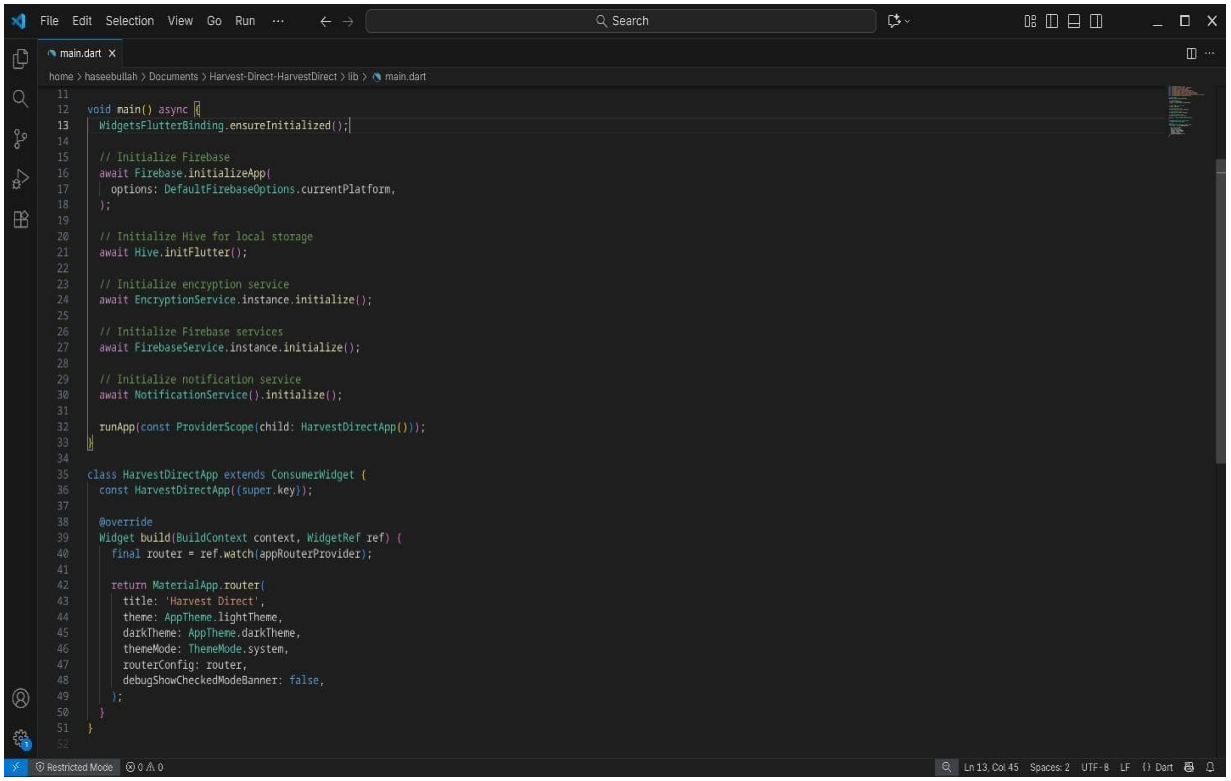
APPENDIX A: Code snippet of Farmer Dashboard



APPENDIX B: Code snippet of Buyer Dashboard



APPENDIX C: Code snippet of MAIN.DART



APPENDIX D: Code snippet of Firebase