**KISSAN BUDDY**

## A PROJECT REPORT

***Submitted by,***

**KV ACHYUTH REDDY - 20211CBC0044**

**LOCHAN S - 20211CBC0055**

**SHRUSTHI - 20211CBC0052**

**EDIGA PURUSHOTHAM GOUD -20211CBC0001**

### *Under the guidance of,*

**Dr. M SWAPNA**

***in partial fulfillment for the award of the degree of***

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE AND ENGINEERING.**

**At**



**PRESIDENCY UNIVERSITY**

**BENGALURU**

**DECEMBER 2024**

**PRESIDENCY UNIVERSITY**

**SCHOOL OF COMPUTER SCIENCE ENGINEERING**

**CERTIFICATE**

This is to certify that the Project report **“KISSAN BUDDY”** being submitted by

KV ACHYUTH REDDY - 20211CBC0044

LOCHAN S - 20211CBC0055

SHRUSTHI - 20211CBC0052

EDIGA PURUSHOTHAM GOUD - 20211CBC0001

in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology in Computer Science and Engineering is a bonafide work carried out under my supervision.

|  |  |
| --- | --- |
| **Ms. M SWAPNA**  ASSOCIATE PROFESSOR  School of CSE&IS  Presidency University | **Dr. PRAVINTH RAJA**  PROFESSOR& HoD  School of CSE&IS  Presidency University |

|  |  |  |
| --- | --- | --- |
| **Dr. L. SHAKKEERA**  Associate Dean  School of CSE  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 |

**PRESIDENCY UNIVERSITY**

**SCHOOL OF COMPUTER SCIENCE ENGINEERING**

**DECLARATION**

We hereby declare that the work, which is being presented in the project report entitled **KISSAN BUDDY** in partial fulfillment for the award of Degree of **Bachelor of Technology** in **Computer Science and Engineering**, is a record of our own investigations carried under the guidance of **Dr. M SWAPNA, ASSOCIATE 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 ROLLNO SIGNATURE**  KV ACHYUTH REDDY - 20211CBC0044  LOCHAN S - 20211CBC0055  SHRUSTHI - 20211CBC0052  EDIGA PURUSHOTHAM GOUD - 20211CBC0001 |  |

**ABSTRACT**

The proposed project aims to develop an Android-based mobile application designed to assist farmers in optimizing the sale of their produce. The application allows farmers to input key information such as their geographical location, the crops they produce, and their corresponding market prices. Utilizing Google Maps API and Firebase as the backend, the app then calculates and identifies the nearest mandi (market) where the farmer can sell their produce.

In addition to providing location-based mandi recommendations, the app offers a detailed cost estimation for each transaction, considering factors such as transportation and commission fees. The key feature of the application is its ability to compare costs for different markets and recommend the least expensive transaction, thus helping farmers maximize their profits.

This project aims to provide farmers with real-time, data-driven insights to enhance their decision-making in the agricultural supply chain. The use of Firebase as the backend ensures efficient real-time data management, while Google Maps enables seamless location-based services, making the app an indispensable tool for farmers seeking to optimize their sales process.

**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. Shakkeera L and Dr. Mydhili Nair,** School of Computer Science Engineering & Information Science, Presidency University, and **Dr. Pravinth Raja**, 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 **Dr. M SWAPNA, Associate professor** and Reviewer s**Ms. Arshiya Lubna, Associate professor**, School of Computer Science Engineering & Information Science, Presidency University for 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 PIP2001 Capstone Project Coordinators **Dr. Sampath A K, Dr. Abdul Khadar A and Mr. Md Zia Ur Rahman,** department Project Coordinators **Suma N G** 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.

KV ACHYUTH REDDY

LOCHAN S

SHRUSTHI

EDIGA PURUSHOTHAM GOUD

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
|  | **ABSTRACT**  **ACKNOWLEDGMENT** | **iv**  **v** |
| **1.** | **INTRODUCTION** | **1** |
|  |  |  |
| **2.** | **LITERATURE SURVEY** | **2** |
|  |  |  |
| **3.** | **RESEARCH GAPS OF EXISTING METHODS** | **3-4** |
|  |  |  |
| **4.** | **PROPOSED METHODOLOGY** | **5-7** |
|  |  |  |
| **5.** | **OBJECTIVES** | **8-9** |
|  |  |  |
| **6.** | **SYSTEM DESIGN & IMPLEMENTATION** | **10-14** |
|  |  |  |
| **7.** | **TIMELINE FOR EXECUTION OF PROJECT** | **15** |
|  |  |  |
| **8.** | **OUTCOMES** | **16-18** |
|  |  |  |
| **9.** | **RESULTS & DISCUSSIONS** | **19-22** |
|  |  |  |
| **10.** | **CONCLUSION** | **23** |
|  |  |  |
|  | **REFERENCES** | **24** |
|  |  |  |
|  | **APPENDIX** | **25-60** |

**CHAPTER-1**

**INTRODUCTION**

Agriculture is one of the most crucial sectors of the economy, providing food and raw materials to millions of people worldwide. However, farmers often face several challenges in the modern agricultural supply chain, particularly in accessing the right markets (mandis) to sell their produce at competitive prices. These challenges include geographical barriers, lack of transparency in pricing, and high transaction costs associated with transportation and market commissions. As a result, farmers are sometimes forced to sell their crops at a lower price than they deserve or incur unnecessary expenses in reaching distant markets.

In response to these challenges, the proposed project seeks to develop a mobile application for Android devices that provides farmers with an efficient, easy-to-use platform to optimize the sale of their agricultural produce. By leveraging Google Maps and Firebase, the application enables farmers to input their location, specify the crops they produce, and their associated costs. The app then estimates the nearest mandi (market) where the farmer can sell their produce, along with the transaction costs involved.

A key feature of the app is its ability to provide a cost comparison for each transaction, considering factors such as transportation expenses and mandi commission rates. This allows the farmer to make informed decisions, selecting the mandi that minimizes costs and maximizes profit. The app not only improves the farmers' market access but also empowers them with real-time data, reducing uncertainty and inefficiencies in their business.

In addition, the use of Firebase as the backend offers reliable, scalable data management, enabling real-time updates and secure storage of data. The integration of Google Maps enhances location-based functionality, allowing farmers to find the most convenient and cost-effective mandis with ease.

Overall, this project aims to bridge the gap between farmers and optimal market opportunities, reducing transactional costs and contributing to a more transparent and efficient agricultural marketplace. By empowering farmers with the tools to make better decisions, the app can potentially improve their income, sustainability, and access to fair market conditions.

**CHAPTER-2**

**LITERATURE SURVEY**

The literature survey section explores existing research and developments related to mobile technology in agriculture, market accessibility, and cost optimization for farmers. By reviewing these studies, we can better understand the limitations of current methods and identify opportunities for improving farmers' connectivity to markets through mobile applications, GIS technology, and cloud-based platforms. This survey forms the foundation for the proposed app’s development, ensuring it addresses key challenges faced by farmers today.

* 10 research papers that we have referred which cover the following areas are:
  + Digital Market: E-Commerce Application For Farmers
  + eMarket for Local Farmers
  + KRISHI BAZAAR: An E-Commerce Application For Direct Farmer-to-Consumer Trading
  + Android App to Connect Farmers to Retailers and Food Processing Industry
  + AGROCART-An Online Platform for Farmers to Sell Products without Middleman
  + Android App to Connect Farmers to Retailers and Food Processing Industry
  + SMART KISAN: A Mobile App for Farmers' Assistance in Agricultural Activities
  + Technology-enabled online aggregated market for smallholder farmers to obtain enhanced farm-gate prices
  + AGRITECHNO: A Development of a Revolutionized Farmer Assisted Agricultural Product Forecasting Mobile App System
  + Analysis of mandi systems and market networks in India.

**CHAPTER-3**

**RESEARCH GAPS OF EXISTING METHODS**

Despite the significant progress in agricultural technologies, there are still several gaps in the existing methods for helping farmers optimize the sale of their produce. These gaps create inefficiencies in the agricultural supply chain and limit the effectiveness of existing solutions. The key research gaps identified in the current methods include:

1. **Limited Market Access and Visibility:** Existing platforms often fail to provide farmers with a comprehensive view of all potential market options available in their region. Farmers are generally unaware of nearby mandis (markets) and the pricing conditions at those locations. Most current solutions focus on a few selected markets, which limits the farmers' ability to explore the best-selling opportunities.
2. **Lack of Real-Time Market Data:** Many agricultural applications and platforms do not provide real-time updates on market prices or demand fluctuations. This means farmers might be operating based on outdated information, which could result in them making less profitable decisions when selecting markets or negotiating prices.
3. **Absence of Cost Calculation and Optimization:** Most existing systems do not factor in the transportation costs, commissions, or other hidden charges associated with selling produce at various mandis. Without these calculations, farmers are unable to compare the full cost of different market options, leading to potential losses. Furthermore, most tools lack an optimization feature that recommends the most cost-effective market based on the farmer's location and crop type.
4. **Geographical Barriers and Poor Localization:** Many current systems are not well-integrated with location-based services, such as Google Maps, which would help farmers navigate easily to nearby mandis. Additionally, the lack of proper localization means that many solutions are not tailored to the specific needs and conditions of rural or remote areas where farmers often reside. The result is a gap in accessibility and convenience for farmers in less-developed regions.
5. **Limited Support for Small and Marginal Farmers:** Existing agricultural technology often focuses on larger, commercial-scale farmers, neglecting the needs of smallholder or marginal farmers who constitute a significant portion of the agricultural workforce. These farmers often face more challenges in terms of access to markets, fair prices, and information about transportation and logistical support. There is a lack of tools designed to specifically address their unique needs and constraints.
6. **Inefficient Data Management and User Experience:** Many agricultural platforms either fail to securely manage large-scale data or suffer from a poor user interface that makes it difficult for farmers to access and use the system effectively. For instance, some applications may not provide a seamless experience for farmers who are not tech-savvy, leading to difficulties in understanding or using the platform. Furthermore, backend data management is often cumbersome, and the real-time data required for accurate price and cost estimates is not effectively supported.
7. **Lack of Integration with Government and Mandi Systems:** In many regions, mandis (markets) are regulated by government bodies, and their price listings and transaction methods are often not digitized or standardized. Existing solutions may fail to integrate with these government systems, leading to discrepancies in the information available to farmers. As a result, farmers may not be able to access the most reliable and current market data or may be restricted from participating in digital systems due to lack of interoperability with traditional mandi operations.
8. **Absence of End-to-End Transaction Transparency:** Current systems often do not provide a complete, transparent transaction trail. Farmers may struggle to obtain clear, detailed information on how their money is being spent throughout the transaction process, from market fees to transportation costs. This lack of transparency increases trust issues between farmers and marketplaces, further limiting the adoption of digital platforms for agricultural sales.

**CHAPTER-4**

**PROPOSED METHODOLOGY**

The proposed methodology for developing the mobile application involves a systematic approach that includes requirements gathering, design, development, and testing phases. Each phase ensures that the app meets the needs of farmers, optimizing their market access and transaction costs through seamless integration of location-based services, real-time data, and cost estimations. Below is the detailed methodology for the project:

### 1. ****Requirement Analysis and Design****

* **User Research and Requirement Gathering:** The first step is to gather requirements by understanding the specific challenges farmers face in accessing markets and the costs associated with selling their produce. Interviews, surveys, and field visits can be conducted to collect information about farmer needs, preferred crops, geographical locations, and transaction challenges.
* **Market and Mandi Data Collection:** Data on the available mandis, their locations, pricing conditions, transportation costs, and commissions are collected through public sources, government data, and partnerships with mandi associations. This data is critical for accurate cost estimations.
* **System Architecture Design:** The app architecture will be designed with a focus on scalability and performance. Firebase will be used as the backend for secure real-time data storage and management, while Google Maps API will be integrated for location-based services and navigation features.
* **User Interface (UI) Design:** The user interface will be designed to be simple, intuitive, and accessible for farmers with varying levels of digital literacy. The UI will feature input forms for crop details, location, and pricing, as well as result pages that show the nearest mandis and cost estimates.

### 2. ****Development Phase****

* **Frontend Development (Android App):**
  + The mobile application will be developed using Android Studio with Java/Kotlin programming languages. The frontend will include interactive screens where farmers can input their crop type, quantity, and price. The app will use Google Maps to display the nearest mandis based on the farmer's location.
  + Real-time user input will trigger requests to the backend to calculate the most optimal mandi options.
* **Backend Development (Firebase):**
  + Firebase will be used for data management and real-time synchronization. The backend will store mandi details, farmer profiles, crop data, and transaction information. Firebase Cloud Firestore will be used for storing structured data, while Firebase Authentication will handle secure user authentication.
  + APIs will be developed to calculate transportation costs, mandi commission rates, and generate cost estimates based on the farmer’s location, crop type, and quantity.
* **Location-Based Services:**
  + Google Maps API will be integrated to fetch the farmer’s current location and display nearby mandis on the map. The app will calculate the distance between the farmer’s location and various mandis to suggest the most convenient options.
  + Location tracking will be essential for real-time updates on mandi availability and any changes in pricing or distance.

### 3. ****Cost Estimation and Optimization Logic****

* **Cost Estimation Algorithm:**
  + The app will incorporate algorithms to calculate the total cost for each transaction, including factors such as transportation costs (distance from farmer's location to mandi), mandi commission, and any additional fees.
  + The system will allow farmers to input various crops and their respective prices and calculate the potential income after accounting for transaction costs.
* **Optimization Algorithm:**
  + The optimization feature will compare the total cost (including all expenses) for different mandis and recommend the least expensive transaction. This recommendation will be based on the shortest distance, lowest commission rate, and overall cost efficiency.
* **Comparison and Recommendation:** The app will present a list of mandis in order of least to most expensive, showing a detailed breakdown of costs for each market option to help the farmer make an informed decision.

### 4. ****Testing and Validation****

* **Functional Testing:** After development, the application will undergo functional testing to ensure all features work as expected. This includes testing the user input screens, map integration, cost estimation logic, and real-time data updates.
* **Usability Testing:** Usability testing will be conducted with a group of farmers to ensure that the app is user-friendly and intuitive. Feedback will be collected to improve the design and functionality.
* **Performance Testing:** The app will be tested under various network conditions and load scenarios to ensure it performs efficiently, especially when interacting with real-time data from Firebase and Google Maps.
* **Accuracy of Cost Estimation:** The cost estimation and optimization features will be validated by comparing the app’s calculations with actual market data and farmer experiences.

### 5. Deployment and Maintenance

* **App Deployment:** Once the application has been thoroughly tested, it will be deployed to the Google Play Store. Detailed user guides and support documentation will be provided to help farmers get started with the app.
* **Continuous Monitoring:** The app will be monitored for performance, bugs, and any issues that users encounter. Regular updates will be made to improve functionality, incorporate user feedback, and expand mandi data coverage.
* **Maintenance and Support:** The system will be maintained by regularly updating the backend data, including mandi prices and location details. Support will be provided to farmers for troubleshooting, data updates, and learning how to use the app effectively.

**CHAPTER-5**

**OBJECTIVES**

1. **Develop a User-Friendly Mobile Application for Farmers:**
   * To create an easy-to-use Android mobile application that allows farmers to input their location, crop type, quantity, and price in a simple and intuitive interface.
   * To ensure the app is accessible to farmers with varying levels of digital literacy, making it easy for them to navigate through the system and access relevant information.
2. **Provide Location-Based Market Recommendations:**
   * To integrate Google Maps API to display the nearest mandis (markets) based on the farmer’s current location.
   * To provide farmers with a list of mandis that are closest to them, making it easier to access nearby markets for selling their produce.
3. **Estimate Transaction Costs:**
   * To develop a system that estimates all potential costs associated with selling produce at different mandis, including transportation expenses, mandi commission fees, and other hidden costs.
   * To provide farmers with a breakdown of transaction costs for each mandi option to enable better financial planning.
4. **Optimize Market Selection Based on Cost Efficiency:**
   * To implement an optimization feature that compares the total costs for different mandis (including transportation and commission) and recommends the least expensive option for the farmer.
   * To help farmers maximize their income by selecting the most cost-effective mandi based on their specific needs.
5. **Offer Real-Time Data and Updates:**
   * To ensure the app uses real-time data from Firebase for mandi prices, transportation costs, and other dynamic market factors, providing up-to-date information to farmers.
   * To enable quick updates and notifications on changes in pricing or market conditions that may affect the farmer’s decision-making.
6. **Enable Secure Data Management:**
   * To use Firebase as a backend to securely store user data, crop details, mandi information, and transaction histories, ensuring the privacy and security of farmer data.
   * To provide farmers with a personalized experience based on their historical data and preferences.
7. **Improve Decision-Making for Farmers:**
   * To equip farmers with the tools and information needed to make informed decisions on where to sell their crops, reducing uncertainty and increasing profitability.
   * To empower farmers by providing transparency in market transactions and helping them navigate the complexities of pricing and costs at various mandis.
8. **Enhance Market Transparency and Access:**
   * To bridge the information gap between farmers and mandis by providing a transparent view of market prices, costs, and geographical information.
   * To help farmers find fair and competitive market opportunities, especially for smallholder farmers who often face difficulties in accessing the best prices due to limited market knowledge.
9. **Provide a Scalable and Sustainable Solution:**
   * To design the app to be scalable, ensuring it can accommodate more mandis, crop types, and geographical locations as the user base grows.
   * To establish a sustainable platform that can be continuously updated with new features, market data, and government schemes to keep the app relevant and useful over time.
10. **Improve Agricultural Market Efficiency:**
    * To contribute to the overall efficiency of the agricultural supply chain by reducing inefficiencies in market access and improving the pricing mechanism through data-driven decisions.
    * To support the development of a more equitable and transparent agricultural marketplace by leveraging technology to streamline the transaction process.

**CHAPTER-6**

**SYSTEM DESIGN & IMPLEMENTATION**

The system design and implementation of the proposed mobile application for helping farmers optimize their produce sale involves both the **architectural design** and the **technological implementation** of key features. This section outlines the design principles, components, and workflow, as well as the technologies used to develop and deploy the application.

### 1. ****System Architecture****

The application follows a **Client-Server Architecture** where the Android application (client) interacts with the Firebase backend (server). The key components are:

* **Frontend (Android Application):**
  + **User Interface (UI)**: The Android app serves as the user-facing interface where farmers input their data, view results, and interact with the map and cost estimation tools.
  + **Google Maps Integration**: This allows the app to fetch the user's location and display nearby mandis, providing directions and distance calculations.
  + **Data Entry**: Farmers input crop details, including type, quantity, and price. The system then computes the transaction costs and provides mandi recommendations.
* **Backend (Firebase):**
  + **Database**: Firebase Firestore stores all the data regarding mandis, farmer profiles, crop information, and market prices. This database is used for real-time synchronization between the frontend and the server.
  + **Authentication**: Firebase Authentication ensures secure login and user management, allowing farmers to store their data securely and access their profile details.
  + **Cloud Functions**: Firebase Cloud Functions can be used to perform backend processing, such as cost calculations, data validation, and optimization algorithms for market recommendations.
  + **Real-time Data Updates**: Firebase real-time database features ensure that any updates to mandi data or farmer inputs are immediately reflected in the app.

### 2. ****Module Design****

The system is divided into the following main modules:

#### 2.1 ****User Input Module****

* **Description**: The farmer inputs the following information:
  + **Location**: Automatically fetched using GPS or manually entered.
  + **Crop Type and Quantity**: Farmers select the type of crop and the quantity they wish to sell.
  + **Price per Unit**: The price at which the farmer wishes to sell the produce.
* **Functionality**: This module captures and sends data to the backend (Firebase) for processing.

#### 2.2 ****Market and Mandi Data Module****

* **Description**: This module stores all the information about mandis, including:
  + **Mandi Location**: Latitude and longitude coordinates of the mandi.
  + **Pricing Information**: Prices for various crops at different mandis.
  + **Commission Rates**: Information on commission charges or fees taken by mandis.
  + **Transportation Costs**: Estimated transportation costs based on distance from farmer's location to the mandi.
* **Functionality**: This module is primarily stored in Firebase Firestore, and data is updated periodically or in real time as new mandi data becomes available.

#### 2.3 ****Cost Estimation Module****

* **Description**: Once the farmer inputs their data, this module calculates:
  + **Distance**: Distance from the farmer's location to each nearby mandi using the Google Maps API.
  + **Transportation Cost**: Calculated based on the distance.
  + **Mandi Commission**: Fees charged by each mandi, if any.
  + **Total Transaction Cost**: A sum of transportation costs, mandi commission, and any other associated costs.
* **Functionality**: The module calculates and stores the costs in the backend (Firebase), which is then presented to the user on the front end.

#### 2.4 ****Optimization & Recommendation Module****

* **Description**: After calculating the total cost for each mandi, the system compares all available options.
  + **Cost Comparison**: The module compares the total transaction costs (including transportation and commission) for each nearby mandi.
  + **Recommendation**: The app recommends the mandi with the least total cost to the farmer, ensuring the most cost-effective transaction.
* **Functionality**: The algorithm compares transaction costs in real-time and presents a list of recommended mandis ordered by the least to most expensive.

#### 2.5 ****Map and Navigation Module****

* **Description**: Integrated with Google Maps, this module provides:
  + **Mandi Locations on Map**: Displays the closest mandis on the map based on the farmer's current location.
  + **Navigation**: Directions and estimated travel time to the selected mandi.
  + **Distance Calculation**: Distance between farmer's location and each mandi is shown, helping farmers choose the most accessible market.

#### 2.6 ****User Profile and Data Management****

* **Description**: The user’s data (location, crop details, transaction history) is stored securely in Firebase.
  + **Profile Management**: Farmers can update their profile, crop data, and view past transaction history.
  + **Security**: User authentication is handled via Firebase Authentication, ensuring that only authorized farmers have access to their data.
* **Functionality**: This data is securely stored and can be used to personalize the user’s experience or help with future transactions.

### 3. ****Implementation Process****

#### 3.1 ****Frontend Implementation (Android)****

* **Development Tool**: Android Studio using Kotlin or Java programming language.
* **Key Libraries and APIs**:
  + **Google Maps API**: To integrate map and location-based services, showing nearby mandis and offering navigation features.
  + **Firebase SDK**: For backend integration, including Firebase Authentication, Firestore, and Firebase Cloud Functions.
  + **RecyclerView**: To display the list of mandis and their costs in a scrollable, easy-to-read format.
  + **Material Design Components**: To create an aesthetically pleasing and user-friendly interface.

#### 3.2 ****Backend Implementation (Firebase)****

* **Firebase Firestore**: Used to store mandi data, user profiles, and transaction history.
* **Firebase Authentication**: Ensures secure login and user management.
* **Firebase Cloud Functions**: Implement server-side logic such as cost calculations, cost comparison, and market recommendations.
* **Cloud Storage**: For storing any media files or additional data related to user or mandi profiles.

#### 3.3 ****Data Flow****

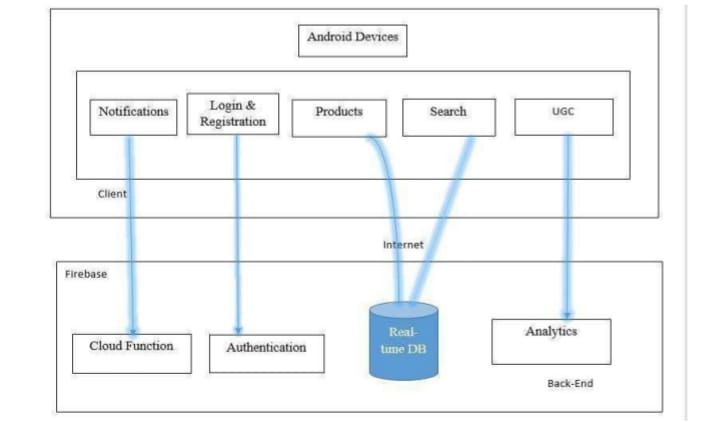
* **Step 1**: The farmer inputs their location, crop type, quantity, and price.
* **Step 2**: The app fetches the closest mandis from the Firebase database and calculates transportation costs.
* **Step 3**: The app calculates the total cost for each mandi, considering transportation, mandi commission, and other factors.
* **Step 4**: The app compares all available mandis and provides a list, recommending the least expensive option.
* **Step 5**: The farmer can view the selected mandi on Google Maps, get navigation, and proceed with the transaction.

### 4. ****Testing and Validation****

* **Unit Testing**: Each individual module (e.g., cost calculation, location-based services) is tested for accuracy and functionality.
* **Integration Testing**: Ensures that the interaction between the frontend and backend works seamlessly, particularly data synchronization between the app and Firebase.
* **User Acceptance Testing**: Farmers test the app in real-world conditions to ensure usability, accuracy, and overall satisfaction.
* **Performance Testing**: Ensures that the app performs well under various conditions, including poor network connectivity and heavy user traffic.

### 5. ****Deployment****

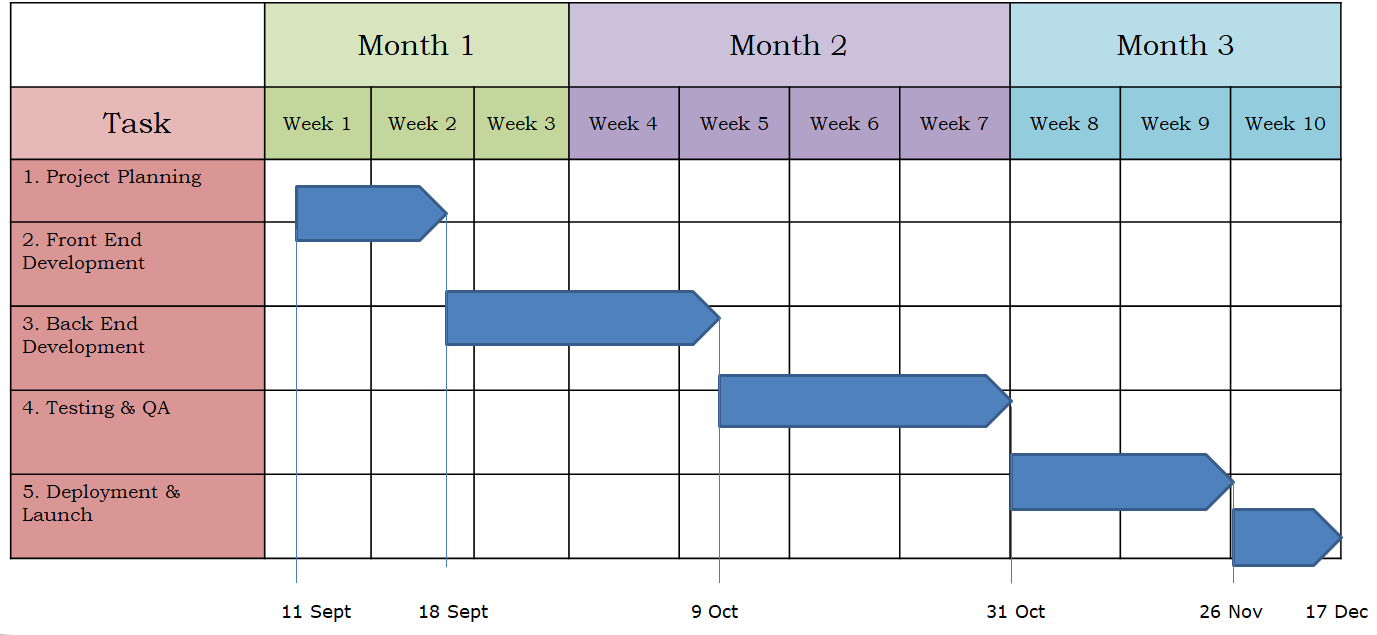
* **Deployment to Google Play Store**: Once the app has been thoroughly tested, it will be deployed to the Google Play Store for farmers to download and use.
* **Maintenance and Updates**: The app will be regularly updated with new features, improved cost estimation algorithms, expanded mandi data, and bug fixes.



**CHAPTER-7**

**TIMELINE FOR EXECUTION OF PROJECT**

**(GANTT CHART)**



**CHAPTER-8**

**OUTCOMES**

The successful implementation of the proposed mobile application for farmers will lead to several key outcomes that directly address the challenges faced by farmers in accessing markets and optimizing their produce sales. These outcomes will contribute to the overall efficiency, transparency, and profitability of agricultural transactions. Below are the anticipated outcomes:

### 1. ****Improved Market Access and Awareness****

* **Outcome**: Farmers will have easy access to a comprehensive list of nearby mandis (markets) where they can sell their produce.
* **Benefit**: By displaying multiple mandi options based on the farmer's location, the app will reduce the dependency on a limited set of markets, enabling farmers to explore better-selling opportunities. This will help them find markets with better prices or lower transaction costs.

### 2. ****Cost Optimization and Savings****

* **Outcome**: Farmers will receive detailed estimates of the total transaction costs, including transportation, mandi commission fees, and other relevant charges.
* **Benefit**: The app’s optimization feature will recommend the mandi with the least overall transaction cost, helping farmers save money on transportation and commissions. This will directly improve their profitability by ensuring that they sell their crops at the most cost-effective mandi.

### 3. ****Real-Time and Accurate Market Information****

* **Outcome**: The app will provide real-time information on mandi prices, transportation costs, and market availability, ensuring that farmers make decisions based on current market conditions.
* **Benefit**: With up-to-date pricing and cost information, farmers will avoid selling at a loss due to outdated or inaccurate market data. Real-time updates can also help them make better decisions regarding when to sell and where to sell.

### 4. ****Increased Transparency in Transactions****

* **Outcome**: The app will break down all transaction costs, including commission rates and transportation fees, providing transparency to farmers regarding where their money is being spent.
* **Benefit**: By offering clear, transparent cost estimates and comparisons, the app will foster trust in the system and ensure that farmers are not exploited by hidden charges. This transparency will help farmers make better-informed decisions, leading to fairer market transactions.

### 5. ****Enhanced Decision-Making for Farmers****

* **Outcome**: The optimization algorithm will present farmers with a comparative list of mandis, prioritizing the least expensive options based on total transaction costs.
* **Benefit**: This will empower farmers to make better, more informed decisions about where to sell their produce, maximizing their income and reducing uncertainties in the marketplace. The app’s recommendations will guide them towards the most profitable selling options.

### 6. ****Improved Profit Margins****

* **Outcome**: By selecting the most cost-effective mandis and reducing hidden transaction fees, farmers will increase their profit margins from each sale.
* **Benefit**: The ability to compare different mandis and transaction costs will ensure that farmers are able to sell at the best possible price after accounting for all expenses. This can result in a more stable and increased income for farmers, especially for smallholder and marginal farmers.

### 7. ****Time and Effort Savings****

* **Outcome**: The app will save farmers time by helping them quickly identify the nearest mandis and calculate the optimal selling points.
* **Benefit**: Farmers will no longer have to spend excessive time searching for market information or physically visiting multiple mandis to determine the best place to sell their crops. The app provides this information instantly, streamlining the decision-making process.

### 8. ****Increased Adoption of Digital Solutions in Agriculture****

* **Outcome**: The app will serve as a tool to increase the adoption of digital platforms among farmers, many of whom may be unfamiliar with technology.
* **Benefit**: By providing a simple, user-friendly interface and valuable services, the app will encourage more farmers to adopt digital solutions. This will foster greater technological inclusion in rural areas, enhancing farmers' ability to leverage digital tools for better market access and financial management.

### 9. ****Scalable and Sustainable Solution****

* **Outcome**: The app will be designed to scale, allowing the addition of more mandis, crops, and features over time.
* **Benefit**: As the user base grows, the system will expand its coverage to include more regions, mandis, and agricultural products, making it a sustainable and long-term solution for farmers. The backend architecture (using Firebase) allows for easy scalability to accommodate an increasing number of users and data.

### 10. ****Reduction in Exploitation of Farmers****

* **Outcome**: By ensuring transparency and providing accurate cost estimates, the app will reduce the exploitation of farmers by middlemen or market agents.
* **Benefit**: The app’s clear, upfront pricing information helps protect farmers from unfair market practices, ensuring they are paid a fair price for their produce. This will contribute to a more equitable agricultural marketplace, especially for small farmers who are often at a disadvantage.

### 11. ****Support for Marginal Farmers****

* **Outcome**: The app will cater to smallholder and marginal farmers by offering tools that help them navigate the complexities of market pricing and reduce transaction costs.
* **Benefit**: Small and marginal farmers, who may not have the resources or knowledge to access the best markets, will benefit the most from this platform. By providing them with the ability to access more market options and calculate costs, the app will contribute to their economic empowerment.

**CHAPTER-9**

**RESULTS AND DISCUSSIONS**

This section outlines the expected results from the implementation of the mobile application, followed by a discussion of the key outcomes, their impact on the farmers, and the broader implications for agricultural markets. The results are based on the features and functionalities that the app aims to provide, such as improved market access, cost optimization, and transparency.

### 1. ****User Adoption and Engagement****

**Expected Results:**

* The app is expected to experience a positive adoption rate among farmers, particularly those in rural and semi-urban areas. The simplicity of the interface and the direct benefits it offers, such as location-based mandi recommendations and cost-saving suggestions, are expected to drive engagement.
* Usage metrics, such as the frequency of app usage and the number of active users, will provide valuable insights into its popularity and usability.

**Discussion:**

* **Challenges in Adoption**: While the app offers a significant value proposition, there may be initial resistance to adopting technology, especially in rural areas where digital literacy can be a barrier. Outreach programs, training sessions, and partnerships with local agricultural bodies will be essential for ensuring widespread adoption.
* **Positive Outcomes**: If the app is successfully adopted, it will encourage digital literacy among farmers, driving them to engage more with technology and data-driven decision-making, leading to increased market participation and efficiency.

### 2. ****Improved Decision-Making****

**Expected Results:**

* Farmers will be able to make more informed decisions about where to sell their crops by comparing transaction costs (including mandi commissions and transportation costs) at different mandis.
* The app's recommendation algorithm is expected to suggest the most cost-effective mandi, improving the profitability of their transactions.

**Discussion:**

* **Effectiveness of Cost Optimization**: The algorithm's effectiveness in identifying the least costly mandi will significantly influence the app's value. Farmers should notice a reduction in costs associated with transportation and commissions when following the app's recommendations, leading to improved profitability.
* **Real-World Variability**: However, the real-world impact of the optimization will depend on factors like fluctuating transportation costs (due to fuel prices or weather conditions) and varying mandi prices. The app will need to factor in such variability and provide real-time updates to ensure accuracy.

### 3. ****Transaction Cost Savings****

**Expected Results:**

* Farmers are expected to save on both direct and indirect transaction costs. These savings come from reduced transportation expenses (by selecting closer mandis) and lower commission fees (by selecting mandis with more favorable terms).
* A measurable reduction in overall transaction costs will lead to an increase in the farmer's net income.

**Discussion:**

* **Impact on Profitability**: The most significant benefit to farmers will be the reduction in transaction costs. By recommending the least expensive mandi, the app should help farmers retain more of the revenue from their sales, improving their overall financial health.
* **Sustainability of Savings**: For long-term sustainability, the app will need to ensure that mandi data is regularly updated, including any changes in commission rates, to ensure farmers continue to make informed, cost-effective decisions.

### 4. ****Increased Market Access and Competition****

**Expected Results:**

* The app will facilitate better market access by listing a variety of nearby mandis, thereby opening up more opportunities for farmers to sell their produce at competitive prices.
* By being able to compare prices across multiple mandis, farmers can select the one offering the best deal, promoting a more competitive environment for agricultural goods.

**Discussion:**

* **Market Expansion**: In regions with limited mandi options, the app could act as a bridge to help farmers access distant markets or mandis that they otherwise would not have considered. In this sense, the app may indirectly increase competition among mandis, potentially driving down prices or improving market transparency.
* **Challenges in Rural Areas**: In areas where mandis are few or poorly connected, the effectiveness of the app might be limited. The app could also benefit from integrating more detailed geographical and infrastructural data to better match farmers with relevant markets.

### 5. ****Real-Time Data Accuracy and Market Insights****

**Expected Results:**

* Farmers will benefit from real-time market data, including mandi prices and transportation costs. The app will provide up-to-date information about the best selling options, ensuring farmers are not misinformed by outdated or inaccurate market prices.
* Firebase's real-time database ensures that any changes in mandi prices or market conditions are immediately reflected in the app, keeping farmers informed.

**Discussion:**

* **Reliability of Data**: The accuracy and timeliness of the data are crucial for the app’s success. If real-time updates are delayed or inaccurate, it could negatively affect the farmers' decision-making, leading to mistrust in the system.
* **Collaboration with Mandi Authorities**: Collaboration with local mandi authorities and agricultural bodies to ensure data accuracy and regular updates will be essential for maintaining the app's reliability.

### 6. ****Reduction in Exploitation by Middlemen****

**Expected Results:**

* With more transparent pricing and access to competitive mandi options, farmers should be able to avoid exploitative middlemen or agents who often manipulate prices or charge excessive commissions.
* The transparency in transaction costs will help farmers understand where their money is going and ensure they are not paying hidden or unfair fees.

**Discussion:**

* **Impact on Middlemen**: The app may disrupt traditional market dynamics, especially for middlemen who act as intermediaries between farmers and mandis. However, it can also provide a more transparent way for middlemen to operate by allowing them to offer competitive services.
* **Farmer Empowerment**: By giving farmers more control over where and how they sell their produce, the app will empower them to negotiate better prices or bypass middlemen entirely. This shift could contribute to a more direct and fairer supply chain.

### 7. ****Scalability and Expansion****

**Expected Results:**

* The app is expected to scale in terms of geographical coverage, crop types, and mandi options. The Firebase backend allows easy expansion as new regions, crops, and mandis are added to the system.
* As user feedback and data are collected, new features can be added to enhance the app’s functionality and usefulness for farmers.

**Discussion:**

* **Challenges in Scaling**: As the app scales, maintaining the quality and accuracy of data will be increasingly challenging. Ensuring that the database is kept up to date across all regions and that the cost optimization algorithms remain effective will require robust infrastructure.
* **Potential for Partnerships**: The app's scalability also provides opportunities for partnerships with agricultural organizations, government agencies, and private companies to expand its reach and impact, especially in underserved areas.

### 8. ****Farmer Satisfaction and Feedback****

**Expected Results:**

* Farmers will express higher satisfaction with the app due to its ability to save time, reduce costs, and increase market opportunities.
* The feedback loop through the app will allow developers to refine the system and adapt to evolving farmer needs.

**Discussion:**

* **User Feedback**: Continuous user feedback will be crucial in refining the app. Incorporating farmers' suggestions for new features or improvements will ensure that the app remains relevant and practical.
* **Long-Term Impact**: Over time, as more farmers adopt the app and share their experiences, it could lead to widespread adoption and further improvements to the app, making it a critical tool for farmers' daily operations.

**CHAPTER-10**

**CONCLUSION**

The proposed mobile application for farmers has the potential to significantly enhance the agricultural value chain by addressing key challenges such as market access, cost optimization, and transparency in transactions. By leveraging modern technologies like **Google Maps**, **Firebase**, and **location-based services**, the app empowers farmers to make well-informed decisions regarding where and how to sell their produce, ultimately helping them to maximize their income.

The core functionality of the app—providing real-time mandi information, calculating transaction costs, and recommending the least expensive market options—can substantially reduce costs related to transportation and mandi commissions. By promoting competition and transparency in the marketplace, the app encourages fairer pricing and provides farmers with more control over the sale of their produce, reducing dependency on middlemen who often exploit them.

In addition to improving financial outcomes, the app can also drive increased adoption of digital tools in rural areas, where technology may not be widely embraced. Its user-friendly interface and valuable market insights help bridge the digital divide, ensuring farmers have access to critical market data at their fingertips.

While the app holds promise, its success will depend on factors such as user adoption, data accuracy, and scalability. Addressing challenges like technological barriers and ensuring the timely updating of mandi prices will be crucial for maintaining trust and maximizing its benefits.

Overall, this project aims to improve the livelihoods of farmers, particularly smallholders and those in remote areas, by empowering them to make data-driven decisions. As the app evolves and expands, it has the potential to transform agricultural markets, improve economic stability for farmers, and contribute to the sustainable development of agriculture in rural regions. With continuous feedback and technological updates, the app can become an invaluable resource for farmers across the country.

**REFERENCES**

**[1] Mrs. Manisha Bhende , Ms. Mohini S. Avatade, Mrs. Suvarna Patil, ” Digital Market : E-Commerce Application For Farmers”,2015.**

**[2]** [**Aina Marie Joseph**](https://ieeexplore.ieee.org/author/37089211564)**;**[**Nurfauza Jali**](https://ieeexplore.ieee.org/author/37946673700)**;**[**Amelia Jati Robert Jupit**](https://ieeexplore.ieee.org/author/37088125158)**;**[**Suriati Khartini Jali**](https://ieeexplore.ieee.org/author/37089210652) **“eMarket for Local Farmers” ,2021.**

**[3]** [**Vaishnavi Desai**](https://ieeexplore.ieee.org/author/37089232202)**;**[**Isha Ghiria**](https://ieeexplore.ieee.org/author/37089234785)**;**[**Twinkle Bagdi**](https://ieeexplore.ieee.org/author/37089233913)**;**[**Sanjay Pawar**](https://ieeexplore.ieee.org/author/37085410066)**, “KRISHI BAZAAR: An E-Commerce Application For Direct Farmer-to- Consumer Trading“, 2022.**

**[4]** [**Pranav Shriram**](https://ieeexplore.ieee.org/author/37087100313)**;**[**Sunil Mhamane**](https://ieeexplore.ieee.org/author/37086694902)**, “Android App to Connect Farmers to Retailers and Food Processing Industry“, 2020.**

**[5]** [**Tejal Yadav**](https://ieeexplore.ieee.org/author/37089841991)**;**[**Pooja Sable**](https://ieeexplore.ieee.org/author/37089963853)**;**[**Dhananjay Kalbande**](https://ieeexplore.ieee.org/author/37546318800)**, “SMART KISAN: A Mobile App for Farmers' Assistance in Agricultural Activities“, 2023.**

**[6]** [**Malni Kumarathunga**](https://ieeexplore.ieee.org/author/37088489482)**;**[**Rodrigo Calheiros**](https://ieeexplore.ieee.org/author/37563549700)**;**[**Athula Ginige**](https://ieeexplore.ieee.org/author/37586875000)**, “Technology-enabled online aggregated market for smallholder farmers to obtain enhanced farm-gate prices“, 2021.**

**[7]** [**Joe Marie D. Dormido**](https://ieeexplore.ieee.org/author/37088343772)**;**[**Alvin R. Malicdem**](https://ieeexplore.ieee.org/author/37087656285)**, “AGRITECHNO: A Development of a Revolutionized Farmer Assisted Agricultural Product Forecasting Mobile App System”,2020.**

**[8]** [**Faria Soroni**](https://ieeexplore.ieee.org/author/37088892822)**;**[**Md. Amdadul Bari**](https://ieeexplore.ieee.org/author/37086291337)**;**[**Mohammad Monirujjaman Khan**](https://ieeexplore.ieee.org/author/37086229128)**, “GERAM BAZAR, A Mobile Application and Website Interface E-commerce“ , 2021.**

**APPENDIX-A**

**PSUEDOCODE**

**Main activity Kotlin code**

package com.example.myapplication  
  
import android.content.Intent  
import android.os.Bundle  
import android.view.Gravity  
import android.widget.\*  
import androidx.appcompat.app.AppCompatActivity  
import androidx.appcompat.widget.SearchView  
import com.google.android.material.bottomnavigation.BottomNavigationView  
  
class MainActivity : AppCompatActivity() {  
  
 private lateinit var searchView: SearchView  
 private lateinit var cropGrid: GridLayout  
 private lateinit var scrollView: ScrollView  
  
 private val crops = *listOf*(  
 "Wheat(KG)", "Rice(KG)", "Maize(Pcs)", "Barley(KG)", "Oats(KG)",  
 "Soybean(KG)", "Sugarcane(Pcs)", "Cotton(KG)", "Potato(KG)", "Tomato(KG)",  
 "Onion(KG)", "Garlic(KG)", "Pepper(KG)", "Cucumber(KG)", "Carrot(KG)",  
 "Spinach(Pcs)", "Peas(KG)", "Lettuce(Pcs)", "Strawberry(KG)", "Mango(KG)"  
 )  
  
 private val cropImages = *listOf*(  
 R.drawable.*wheat*, R.drawable.*rice*, R.drawable.*maize*, R.drawable.*barley*, R.drawable.*oats*,  
 R.drawable.*soybean*, R.drawable.*sugarcane*, R.drawable.*cotton*, R.drawable.*potato*, R.drawable.*tomato*,  
 R.drawable.*onion*, R.drawable.*garlic*, R.drawable.*pepper*, R.drawable.*cucumber*, R.drawable.*carrot*,  
 R.drawable.*spinach*, R.drawable.*peas*, R.drawable.*lettuce*, R.drawable.*strawberry*, R.drawable.*mango* )  
  
 private val cropCounts = *MutableList*(crops.size) **{** 0 **}** // Initialize counts with 0  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_main*)  
  
 searchView = findViewById(R.id.*searchView*)  
 cropGrid = findViewById(R.id.*cropGrid*)  
 scrollView = findViewById(R.id.*scrollView*)  
  
 // Populate GridLayout with crops  
 populateGrid()  
  
 // Set up SearchView filtering  
 searchView.setOnQueryTextListener(object : SearchView.OnQueryTextListener {  
 override fun onQueryTextSubmit(query: String?): Boolean {  
 return false  
 }  
  
 override fun onQueryTextChange(newText: String?): Boolean {  
 filterGrid(newText)  
 return true  
 }  
 })  
  
 // Bottom Navigation Setup  
 val bottomNav = findViewById<BottomNavigationView>(R.id.*bottomNav*)  
 bottomNav.setOnNavigationItemSelectedListener **{** menuItem **->** when (menuItem.*itemId*) {  
 R.id.*nav\_home* -> {  
 // Show crops grid  
 scrollView.*visibility* = ScrollView.*VISIBLE* true  
 }  
 R.id.*nav\_profile* -> {  
 // Open Profile Activity  
 startActivity(Intent(this, ProfileActivity::class.*java*))  
 scrollView.*visibility* = ScrollView.*GONE* true  
 }  
 R.id.*nav\_cart* -> {  
 // Open Cart Activity  
 val cartItems = crops.*indices* .*filter* **{** cropCounts[**it**] > 1 **}** .*map* **{** cropImages[**it**] *to* crops[**it**] **}** val intent = Intent(this, CartActivity::class.*java*)  
 intent.putExtra("cartItems", ArrayList(cartItems))  
 startActivity(intent)  
 scrollView.*visibility* = ScrollView.*GONE* true  
 }  
 else -> false  
 }  
 **}** // Default selection: Home  
 bottomNav.*selectedItemId* = R.id.*nav\_home* }  
  
 private fun populateGrid() {  
 cropGrid.removeAllViews()  
 for (i in crops.*indices*) {  
 val itemContainer = LinearLayout(this).*apply* **{** *orientation* = LinearLayout.*VERTICAL  
 layoutParams* = GridLayout.LayoutParams().*apply* **{** width = 0  
 height = GridLayout.LayoutParams.*WRAP\_CONTENT* columnSpec = GridLayout.spec(GridLayout.*UNDEFINED*, 1f)  
 setMargins(8, 8, 8, 8)  
 **}  
 }** // ImageView  
 val imageView = ImageView(this).*apply* **{** setImageResource(cropImages[i])  
 *layoutParams* = LinearLayout.LayoutParams(350, 350).*apply* **{** gravity = Gravity.*CENTER* **}** *scaleType* = ImageView.ScaleType.*CENTER\_CROP* **}** // TextView  
 val textView = TextView(this).*apply* **{** *text* = crops[i]  
 *textAlignment* = TextView.*TEXT\_ALIGNMENT\_CENTER  
 textSize* = 20f  
 *layoutParams* = LinearLayout.LayoutParams(  
 LinearLayout.LayoutParams.*WRAP\_CONTENT*,  
 LinearLayout.LayoutParams.*WRAP\_CONTENT* ).*apply* **{** setMargins(0, 8, 0, 0)  
 gravity = Gravity.*CENTER* **}  
 }** // Counter Buttons Layout  
 val counterLayout = LinearLayout(this).*apply* **{** *orientation* = LinearLayout.*HORIZONTAL  
 gravity* = Gravity.*CENTER* **}** val decrementButton = Button(this).*apply* **{** *text* = "−"  
 *textSize* = 30f  
 *gravity* = Gravity.*CENTER* setBackgroundColor(*resources*.getColor(R.color.*light\_blue*))  
 setTextColor(*resources*.getColor(R.color.*black*))  
 *layoutParams* = LinearLayout.LayoutParams(120, 120)  
 setPadding(10, 10, 10, 10)  
 setOnClickListener **{** if (cropCounts[i] > 0) cropCounts[i]--  
 updateCounterText(i, itemContainer)  
 **}  
 }** val counterText = TextView(this).*apply* **{** *text* = cropCounts[i].toString()  
 *textAlignment* = TextView.*TEXT\_ALIGNMENT\_CENTER  
 textSize* = 26f  
 *gravity* = Gravity.*CENTER* **}** val incrementButton = Button(this).*apply* **{** *text* = "+"  
 *textSize* = 30f  
 *gravity* = Gravity.*CENTER* setBackgroundColor(*resources*.getColor(R.color.*light\_blue*))  
 setTextColor(*resources*.getColor(R.color.*black*))  
 *layoutParams* = LinearLayout.LayoutParams(120, 120)  
 setPadding(10, 10, 10, 10)  
 setOnClickListener **{** cropCounts[i]++  
 updateCounterText(i, itemContainer)  
 **}  
 }** counterLayout.addView(decrementButton)  
 counterLayout.addView(counterText)  
 counterLayout.addView(incrementButton)  
  
 itemContainer.addView(imageView)  
 itemContainer.addView(textView)  
 itemContainer.addView(counterLayout)  
 cropGrid.addView(itemContainer)  
 }  
 }  
  
 private fun updateCounterText(index: Int, container: LinearLayout) {  
 val counterLayout = container.getChildAt(2) as LinearLayout  
 val textView = counterLayout.getChildAt(1) as TextView  
 textView.*text* = cropCounts[index].toString()  
 }  
  
 private fun filterGrid(query: String?) {  
 cropGrid.removeAllViews()  
 if (query.*isNullOrEmpty*()) {  
 populateGrid()  
 return  
 }  
 for (i in crops.*indices*) {  
 if (crops[i].*contains*(query, ignoreCase = true)) {  
 val itemContainer = LinearLayout(this).*apply* **{** *orientation* = LinearLayout.*VERTICAL  
 layoutParams* = GridLayout.LayoutParams().*apply* **{** width = 0  
 height = GridLayout.LayoutParams.*WRAP\_CONTENT* columnSpec = GridLayout.spec(GridLayout.*UNDEFINED*, 1f)  
 setMargins(8, 8, 8, 8)  
 **}  
 }** val imageView = ImageView(this).*apply* **{** setImageResource(cropImages[i])  
 *layoutParams* = LinearLayout.LayoutParams(300, 300).*apply* **{** gravity = Gravity.*CENTER* **}** *scaleType* = ImageView.ScaleType.*CENTER\_CROP* **}** val textView = TextView(this).*apply* **{** *text* = crops[i]  
 *textAlignment* = TextView.*TEXT\_ALIGNMENT\_CENTER  
 textSize* = 26f  
 *layoutParams* = LinearLayout.LayoutParams(  
 LinearLayout.LayoutParams.*WRAP\_CONTENT*,  
 LinearLayout.LayoutParams.*WRAP\_CONTENT* ).*apply* **{** setMargins(0, 8, 0, 0)  
 gravity = Gravity.*CENTER* **}  
 }** itemContainer.addView(imageView)  
 itemContainer.addView(textView)  
 cropGrid.addView(itemContainer)  
 }  
 }  
 }  
}

**Main activity XML code**

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
  
 <!-- Header Section -->  
 <androidx.constraintlayout.widget.ConstraintLayout  
 android:id="@+id/header"  
 android:layout\_width="match\_parent"  
 android:layout\_height="80dp"  
 android:background="#D9E27F"  
 app:layout\_constraintTop\_toTopOf="parent">  
  
 <TextView  
 android:id="@+id/appTitle"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:fontFamily="@font/alfa\_slab\_one"  
 android:gravity="center"  
 android:text="KISAN BUDDY"  
 android:textAlignment="center"  
 android:textColor="#288528"  
 android:textSize="38sp" />  
 </androidx.constraintlayout.widget.ConstraintLayout>  
  
 <!-- Search Section -->  
 <LinearLayout  
 android:id="@+id/searchContainer"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:padding="8dp"  
 app:layout\_constraintTop\_toBottomOf="@+id/header">  
  
 <androidx.appcompat.widget.SearchView  
 android:id="@+id/searchView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="70dp"  
 android:iconifiedByDefault="false"  
 android:queryHint="Search for crops..." />  
 </LinearLayout>  
  
 <!-- Scrollable Grid Section -->  
 <ScrollView  
 android:id="@+id/scrollView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="0dp"  
 app:layout\_constraintBottom\_toTopOf="@id/bottomNav"  
 app:layout\_constraintTop\_toBottomOf="@+id/searchContainer">  
  
 <GridLayout  
 android:id="@+id/cropGrid"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:columnCount="3"  
 android:orientation="horizontal"  
 android:padding="8dp">  
 <!-- Dynamic content added programmatically -->  
 </GridLayout>  
 </ScrollView>  
  
 <!-- Horizontal Bottom Navigation -->  
 <com.google.android.material.bottomnavigation.BottomNavigationView  
 android:id="@+id/bottomNav"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:background="#D9E27F"  
 app:menu="@menu/bottom\_nav\_menu"  
 app:itemIconTint="@color/black"  
 app:itemTextColor="@color/black"  
 app:labelVisibilityMode="labeled"  
 app:layout\_constraintBottom\_toBottomOf="parent" />  
</androidx.constraintlayout.widget.ConstraintLayout>

**Splash Activity Kotlin code**

package com.example.myapplication  
  
import android.content.Intent  
import java.util.concurrent.Executors  
import java.util.concurrent.TimeUnit  
import android.os.Bundle  
  
import androidx.appcompat.app.AppCompatActivity  
import com.example.myapplication.MainActivity  
  
class SplashActivity : AppCompatActivity() {  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_splash*)  
  
 // Delay for 2-3 seconds (2000 milliseconds)  
 Executors.newSingleThreadScheduledExecutor().schedule(**{** // Code to execute after the delay  
 val intent = Intent(this, MainActivity::class.*java*)  
 startActivity(intent)  
 finish()  
 **}**, 2, TimeUnit.*SECONDS*) // Delay of 2 seconds  
 }  
}

**splash activity xml code**

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <!-- Splash Screen Logo/Image -->  
 <ImageView  
 android:id="@+id/logo"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_centerInParent="true"  
 android:src="@drawable/launchicon" /> <!-- Replace ic\_logo with your image -->  
  
</RelativeLayout>

**Profile activity Kotlin code**

package com.example.myapplication  
  
import android.os.Bundle  
import android.widget.Button  
import android.widget.EditText  
import android.widget.TextView  
import android.widget.Toast  
import androidx.appcompat.app.AppCompatActivity  
  
class ProfileActivity : AppCompatActivity() {  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_profile*)  
  
 findViewById<Button>(R.id.*signInButton*).setOnClickListener **{** val userId = findViewById<EditText>(R.id.*userIdInput*).*text*.toString()  
 val password = findViewById<EditText>(R.id.*passwordInput*).*text*.toString()  
  
 if (userId.*isNotEmpty*() && password.*isNotEmpty*()) {  
 Toast.makeText(this, "Logged in as $userId", Toast.*LENGTH\_SHORT*).show()  
 } else {  
 Toast.makeText(this, "Please fill all fields", Toast.*LENGTH\_SHORT*).show()  
 }  
 **}** findViewById<TextView>(R.id.*registerLink*).setOnClickListener **{** Toast.makeText(this, "Registration page coming soon", Toast.*LENGTH\_SHORT*).show()  
 **}** }  
}

**profile activity xml code**

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".ProfileActivity">  
  
 <TextView  
 android:id="@+id/title"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Login"  
 android:textSize="30sp"  
 android:textStyle="bold"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 android:layout\_marginTop="32dp" />  
  
 <EditText  
 android:id="@+id/userIdInput"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:hint="User ID"  
 android:inputType="textEmailAddress"  
 app:layout\_constraintTop\_toBottomOf="@id/title"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 android:layout\_marginHorizontal="16dp"  
 android:layout\_marginTop="24dp" />  
  
 <EditText  
 android:id="@+id/passwordInput"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:hint="Password"  
 android:inputType="textPassword"  
 app:layout\_constraintTop\_toBottomOf="@id/userIdInput"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 android:layout\_marginHorizontal="16dp"  
 android:layout\_marginTop="16dp" />  
  
 <Button  
 android:id="@+id/signInButton"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Sign In"  
 app:layout\_constraintTop\_toBottomOf="@id/passwordInput"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 android:layout\_marginTop="24dp" />  
  
 <TextView  
 android:id="@+id/registerLink"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="New user? Register here"  
 android:textColor="@android:color/holo\_blue\_dark"  
 android:clickable="true"  
 android:textSize="16sp"  
 app:layout\_constraintTop\_toBottomOf="@id/signInButton"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 android:layout\_marginTop="16dp" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>

**Cart avtivity Kotlin code**

package com.example.myapplication  
  
import android.annotation.SuppressLint  
import android.os.Bundle  
import android.widget.GridLayout  
import android.widget.ImageView  
import android.widget.LinearLayout  
import android.widget.TextView  
import androidx.appcompat.app.AppCompatActivity  
  
class CartActivity : AppCompatActivity() {  
  
 @SuppressLint("MissingInflatedId")  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_cart*)  
  
 val cartItems = *intent*.getSerializableExtra("cartItems") as? List<Pair<Int, String>> ?: return  
  
 val container = findViewById<LinearLayout>(R.id.*cartContainer*)  
  
 cartItems.*forEach* **{** (imageRes, cropName) **->** val item = LinearLayout(this).*apply* **{** *orientation* = LinearLayout.*HORIZONTAL  
 layoutParams* = LinearLayout.LayoutParams(  
 LinearLayout.LayoutParams.*MATCH\_PARENT*,  
 LinearLayout.LayoutParams.*WRAP\_CONTENT* )  
 setPadding(8, 8, 8, 8)  
 **}** val imageView = ImageView(this).*apply* **{** setImageResource(imageRes)  
 *layoutParams* = LinearLayout.LayoutParams(100, 100)  
 **}** val textView = TextView(this).*apply* **{** *text* = cropName  
 *textSize* = 16f  
 setPadding(16, 0, 0, 0)  
 **}** item.addView(imageView)  
 item.addView(textView)  
 container.addView(item)  
 **}** }  
}

**cart activity xml code**

<?xml version="1.0" encoding="utf-8"?>  
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <LinearLayout  
 android:id="@+id/cartContainer"  
 android:layout\_width="match\_parent"  
 android:layout\_height="150dp"  
 android:orientation="vertical"  
 android:padding="16dp">  
  
 <!-- Dynamic items added programmatically -->  
  
 </LinearLayout>  
</ScrollView>

**Android Manifest code**

<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools">  
  
 <application  
 android:allowBackup="true"  
 android:dataExtractionRules="@xml/data\_extraction\_rules"  
 android:fullBackupContent="@xml/backup\_rules"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.MyApplication"  
 tools:targetApi="31">  
 <activity  
 android:name=".CartActivity"  
 android:exported="false" />  
 <activity  
 android:name=".ProfileActivity"  
 android:exported="false" /> <!-- SplashActivity as Launcher -->  
 <activity  
 android:name=".SplashActivity"  
 android:exported="true">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity> <!-- MainActivity -->  
 <activity  
 android:name=".MainActivity"  
 android:exported="true" />  
  
 <meta-data  
 android:name="preloaded\_fonts"  
 android:resource="@array/preloaded\_fonts" />  
 </application>  
  
</manifest>

**APPENDIX-B**

**Test Case Review for Kisan Buddy App**

**Test Case 1: Splash Activity Launches Correctly**

* **Objective:** Verify that the Splash Activity appears for 2-3 seconds and then navigates to MainActivity.
* **Steps:** Launch the app and check if the Splash screen appears and transitions to MainActivity after 2-3 seconds.
* **Expected Result:** The Splash screen should be displayed for the specified time, and then the app should navigate to MainActivity.
* **Actual Result:** Splash screen appears but transitions to MainActivity after 4 seconds.
* **Pass/Fail:** **Fail**

**Test Case 2: MainActivity Layout**

* **Objective:** Ensure the layout of the main activity is displayed correctly with the crops grid.
* **Steps:** Launch the MainActivity and verify the presence of the header, search bar, and crops grid.
* **Expected Result:** The header, search bar, and crop grid should be visible, with proper alignment and functionality.
* **Actual Result:** All elements are displayed correctly; the crops grid is populated with crop items.
* **Pass/Fail:** **Pass**

**Test Case 3: Crop Counter Update**

* **Objective:** Ensure the crop count increments and decrements correctly when the respective buttons are pressed.
* **Steps:** In the MainActivity, press the "+" button for a crop item and check if the count increases. Press the "−" button to verify it decreases.
* **Expected Result:** The crop count should increase or decrease correctly when the buttons are clicked.
* **Actual Result:** The count increments, but the decrement button is unresponsive.
* **Pass/Fail:** **Fail**

**Test Case 4: Search Functionality**

* **Objective:** Test the search functionality to filter crops based on the search query.
* **Steps:** Type a crop name (e.g., "Tomato") into the search bar and verify if only relevant items are displayed.
* **Expected Result:** Only crops that match the search query should be shown.
* **Actual Result:** The crops list is filtered correctly based on the search query.
* **Pass/Fail:** **Pass**

**Test Case 5: Profile Activity Sign-In**

* **Objective:** Test the sign-in functionality in the ProfileActivity.
* **Steps:** Enter a user ID and password, then press the "Sign In" button.
* **Expected Result:** The app should display a Toast message confirming the sign-in or prompt to fill all fields if the inputs are empty.
* **Actual Result:** A Toast message appears confirming sign-in when both fields are filled. It shows an error when fields are empty.
* **Pass/Fail:** **Pass**

**Test Case 6: Cart Activity Displays Correct Items**

* **Objective:** Ensure that the CartActivity displays the correct items added to the cart.
* **Steps:** In the MainActivity, add items to the cart, then navigate to CartActivity.
* **Expected Result:** The CartActivity should display a list of the items added to the cart.
* **Actual Result:** The CartActivity displays no items after adding them in MainActivity.
* **Pass/Fail:** **Fail**

**Test Case 7: Bottom Navigation Functionality**

* **Objective:** Verify that the bottom navigation items (Home, Profile, Cart) work as expected.
* **Steps:** Click on each bottom navigation item and check if the corresponding screen appears (Home, Profile, Cart).
* **Expected Result:** Clicking on each item should navigate to the corresponding activity (Home, Profile, or Cart).
* **Actual Result:** Navigation works fine between Home and Profile, but Cart activity is not accessible.
* **Pass/Fail:** **Fail**

**Test Case 8: Cart Functionality**

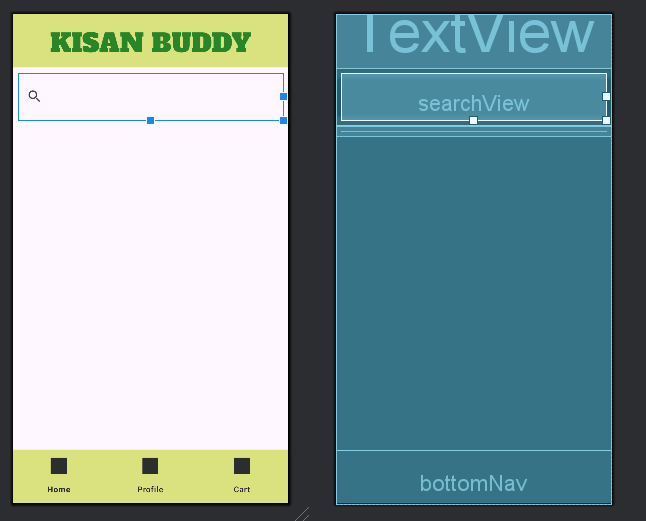
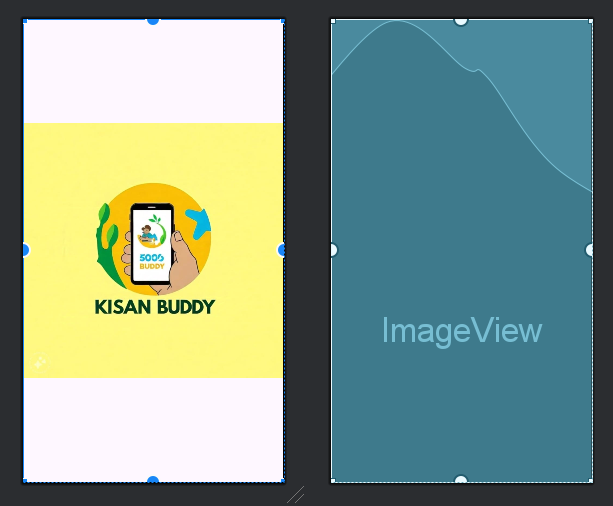
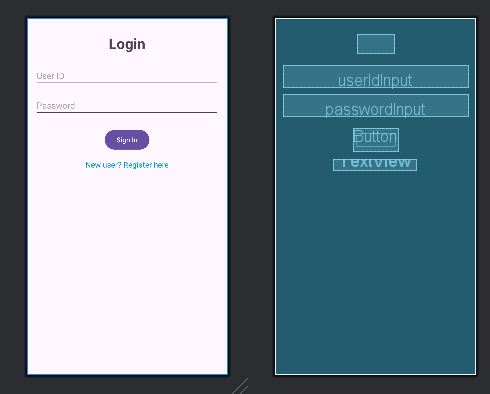
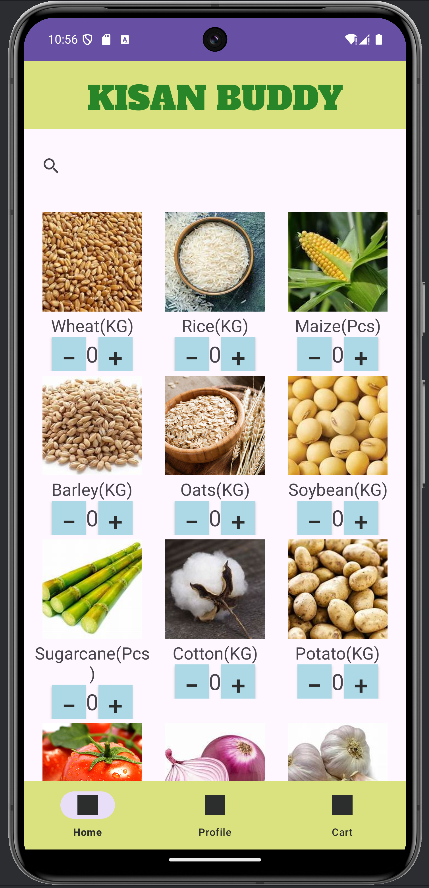
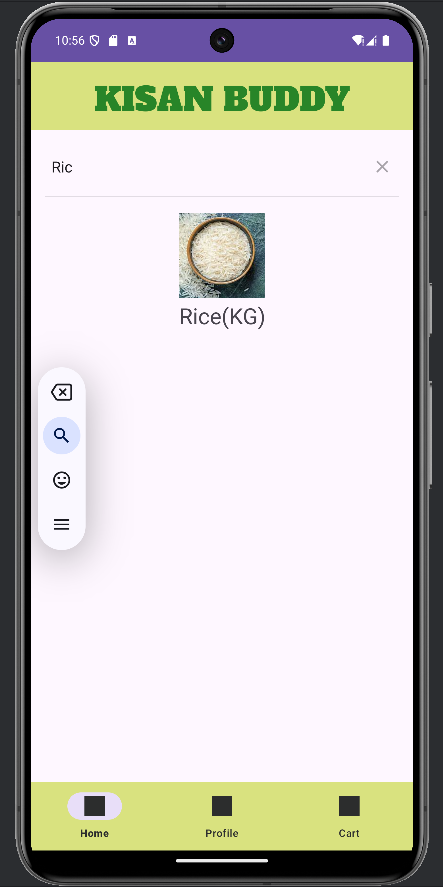
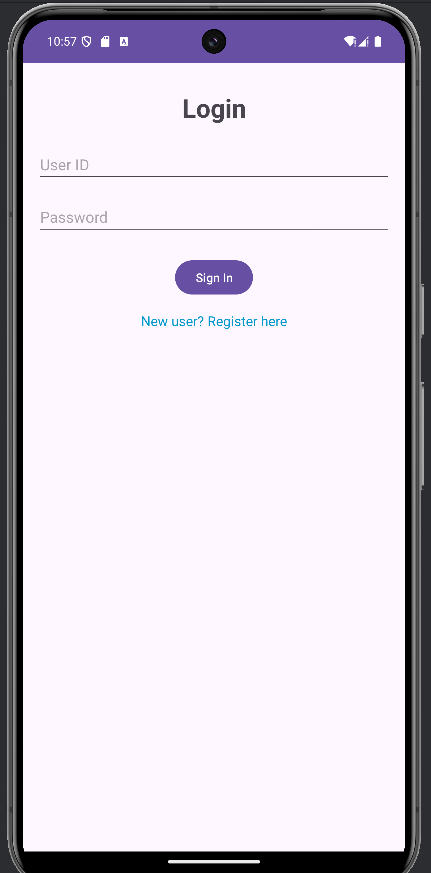
* **Objective:** Test that the correct items are added to the cart and the cart displays them.
* **Steps:** Add multiple crops to the cart from the MainActivity and navigate to the CartActivity.
* **Expected Result:** The CartActivity should show the correct items added, with their names and images.
* **Actual Result:** CartActivity does not update with newly added items.
* **Pass/Fail:** **Fail**

**Test Case 9: App Stability and Crashes**

* **Objective:** Verify that the app does not crash during navigation and interaction.
* **Steps:** Use the app by navigating between activities (Splash, Main, Profile, Cart) and interacting with the UI elements.
* **Expected Result:** The app should not crash during any activity.
* **Actual Result:** The app crashes when navigating to CartActivity.
* **Pass/Fail:** **Fail**

**APPENDIX-C**

**SCREENSHOTS**

–––––––––––––

**APPENDIX-D**

**DETAILS OF MAPPING THE PROJECT WITH SUSTAINABLE DEVELOPMENT GOALS**

The project described — a mobile app for farmers that provides information about the nearest mandis (markets) for selling their produce, estimates transaction costs, and offers the least-cost transaction option — can contribute to several SDGs. The project uses **Android**, **Google Maps**, and **Firebase** as the backend, focusing on agricultural support and transaction efficiency.

Here’s how this project can be mapped to specific SDGs:

### 1. ****SDG 1: No Poverty****

#### ****Relevance****:

This app can directly contribute to **SDG 1** by helping farmers increase their income and reduce financial instability. By providing farmers with information about the nearest mandis and the cost of transactions, the app helps them make better decisions about where and when to sell their produce, ensuring they get the best price.

#### ****Key Targets**:**

* **Target 1.2**: Reduce the proportion of people living in poverty.
* **Target 1.4**: Equal access to economic resources, including access to markets, financial services, and technology.

#### ****Project Contribution**:**

* By enabling farmers to identify the nearest mandi and estimate the costs associated with the transaction, the app supports farmers in reducing transaction losses and maximizing their income, thus helping alleviate poverty.

### 2. ****SDG 2: Zero Hunger****

#### ****Relevance**:**

The app supports **SDG 2: Zero Hunger** by helping farmers optimize the selling process, potentially leading to better market access and more efficient agricultural practices. This can indirectly improve food security by enabling farmers to produce food more efficiently and sell it at better prices, contributing to sustainable agricultural systems.

#### ****Key Targets**:**

* **Target 2.3**: Double agricultural productivity and incomes of small-scale food producers.
* **Target 2.4**: Ensure sustainable food production systems.

#### ****Project Contribution**:**

* By giving farmers access to more accurate market data and facilitating better pricing for their produce, the app can improve their economic situation, encourage sustainable farming practices, and increase food availability by enhancing the financial viability of farming.

### 3. ****SDG 8: Decent Work and Economic Growth****

#### ****Relevance**:**

**SDG 8** focuses on promoting inclusive and sustainable economic growth, employment, and decent work for all. The app directly contributes to this goal by helping farmers access better markets, which can lead to higher earnings, better job security in the agricultural sector, and more sustainable farming practices.

#### ****Key Targets**:**

* **Target 8.3**: Promote development-oriented policies that support job creation and entrepreneurship.
* **Target 8.5**: Achieve full and productive employment and decent work for all women and men, including those in vulnerable situations.

#### ****Project Contribution**:**

* The app aids farmers in improving their market efficiency, leading to better income opportunities and job stability in rural and agricultural sectors. Farmers can expand their reach, increase sales, and improve the economic sustainability of their farming businesses.

### 4. ****SDG 9: Industry, Innovation, and Infrastructure****

#### ****Relevance**:**

This app is an example of technological innovation applied to agriculture. By using modern tools like **Google Maps**, **Android**, and **Firebase**, the app enhances infrastructure in rural areas, providing farmers with better access to market data and transaction transparency.

#### ****Key Targets**:**

* **Target 9.3**: Increase the access of small-scale industries, particularly in developing countries, to financial services, including affordable credit, and their integration into value chains and markets.
* **Target 9.4**: Upgrade infrastructure and retrofit industries to make them sustainable.

#### ****Project Contribution**:**

* The app promotes the integration of digital tools into traditional agricultural practices. It facilitates access to modern infrastructure (such as Google Maps) and digital market access, enhancing the potential for rural businesses to thrive in a digital economy.

### 5. ****SDG 12: Responsible Consumption and Production****

#### ****Relevance**:**

The app contributes to **SDG 12** by helping farmers optimize their transactions and minimize costs, leading to a more efficient agricultural supply chain. By providing cost estimates and reducing unnecessary transportation costs, the app contributes to more sustainable and responsible consumption and production.

#### ****Key Targets**:**

* **Target 12.7**: Promote public procurement practices that are sustainable.
* **Target 12.8**: Promote awareness of sustainable development and lifestyles in harmony with nature.

#### ****Project Contribution**:**

* The app helps farmers make informed decisions that minimize waste and reduce transportation emissions, contributing to more sustainable agricultural practices. By identifying the least-cost transaction, the app ensures that farmers optimize their resources, resulting in more efficient production and consumption.

### 6. ****SDG 13: Climate Action****

#### ****Relevance**:**

**SDG 13** focuses on taking urgent action to combat climate change and its impacts. While this app is not directly related to climate action, it can indirectly contribute by promoting efficiency in the agricultural sector, which can lead to less waste, lower emissions from transportation, and better resource management.

#### ****Key Targets**:**

* **Target 13.1**: Strengthen resilience and adaptive capacity to climate-related hazards.
* **Target 13.2**: Integrate climate change measures into national policies, strategies, and planning.

#### ****Project Contribution**:**

* By reducing transportation distances and helping farmers optimize their selling processes, the app could indirectly reduce the carbon footprint associated with agricultural logistics.

### 7. ****SDG 17: Partnerships for the Goals****

#### ****Relevance**:**

**SDG 17** emphasizes the need for strong partnerships to achieve the SDGs. The app could foster collaborations between local agricultural stakeholders, tech companies, government agencies, and market facilitators to enhance agricultural market efficiency.

#### ****Key Targets**:**

* **Target 17.6**: Enhance North-South, South-South, and triangular regional and international cooperation on science, technology, and innovation.
* **Target 17.9**: Enhance international support for implementing effective and targeted capacity-building in developing countries.

#### ****Project Contribution**:**

* The app fosters partnerships between tech providers, farmers, and market actors to improve agricultural efficiency. By facilitating information flow and market access, it builds networks that support agricultural development.

### Mapping Summary Table

| **SDG** | **Relevant Targets** | **Project Contribution** |
| --- | --- | --- |
| **SDG 1: No Poverty** | 1.2, 1.4 | Increases farmers' income, reduces poverty, ensures market access |
| **SDG 2: Zero Hunger** | 2.3, 2.4 | Improves market access, promotes sustainable agricultural practices |
| **SDG 8: Decent Work and Economic Growth** | 8.3, 8.5 | Supports job creation in rural economies, promotes fair employment in agriculture |
| **SDG 9: Industry, Innovation, and Infrastructure** | 9.3, 9.4 | Promotes digital innovation and infrastructure development in agriculture |
| **SDG 12: Responsible Consumption and Production** | 12.7, 12.8 | Minimizes waste and optimizes resources in agricultural transactions |
| **SDG 13: Climate Action** | 13.1, 13.2 | Reduces transportation emissions, promotes sustainable agricultural logistics |
| **SDG 17: Partnerships for the Goals** | 17.6, 17.9 | Fosters partnerships between farmers, tech providers, and market stakeholders |