CHAPTER 1

INTRODUCTION

In the dynamic landscape of modern agriculture, farmers often face numerous challenges in accessing fair markets for their produce. Traditional methods of selling crops frequently involve intermediaries, leading to reduced profit margins and limited control over pricing. These issues are compounded by a lack of real-time market information, logistical inefficiencies, and the complexities of navigating supply chains. To address these challenges, we propose the development of a mobile application designed to facilitate Direct Market Access (DMA) for farmers. This innovative platform will empower farmers by connecting them directly with consumers, retailers, and other buyers, thereby eliminating intermediaries and optimizing the selling process.

The core objective of this mobile app is to create a transparent, efficient, and equitable marketplace for agricultural products. By leveraging the power of mobile technology, the app will provide farmers with tools to manage their sales, access market information, and improve their overall profitability. The app will enable farmers to list their products, set prices, and negotiate directly with buyers. It will also provide features for managing inventory, tracking sales, and receiving payments securely. Furthermore, the app will integrate features for providing real-time market data, weather updates, and best practices for farming, empowering farmers with the knowledge they need to make informed decisions.

The development of this mobile app is rooted in the belief that technology can play a transformative role in uplifting the agricultural sector. By providing farmers with direct access to markets, the app will contribute to several key benefits. Firstly, it will increase farmers' income by eliminating intermediaries and allowing them to capture a larger share of the market value. Secondly, it will enhance market transparency by providing real-time pricing information and reducing information asymmetry. Thirdly, it will improve efficiency by streamlining the sales process and reducing the time and effort required to sell crops. Finally, it will promote sustainability by reducing food waste and supporting local economy.

The app will be designed with a user-centric approach, ensuring that it is accessible and easy to use for farmers with varying levels of technological literacy. The user interface will be intuitive and multilingual, supporting local languages to cater to a diverse user base. The app will also incorporate features for offline access, enabling farmers to use the app even in areas with limited internet connectivity. Furthermore, the app will be integrated with secure payment gateways and logistics providers to facilitate seamless transactions and deliveries.

The target users of the app will include smallholder farmers, farmer producer organizations (FPOs), and other agricultural stakeholders. The app will be designed to cater to the specific needs of each user group, providing tailored features and functionalities. For example, the app will provide FPOs with tools to manage their members' produce, coordinate sales, and access bulk buyers. For retailers and consumers, the app will provide a platform to source fresh, high-quality produce directly from farmers, supporting local agriculture and reducing carbon emissions.

The development process will involve several key phases, including requirements gathering, app design, development, testing, and deployment. The app will be developed using a cross-platform framework to ensure compatibility with both Android and iOS devices. The development team will consist of experienced software developers, UI/UX designers, and agricultural experts. The project will follow an agile development methodology, allowing for iterative development and continuous feedback.

In conclusion, the mobile app for Direct Market Access for Farmers represents a significant opportunity to transform the agricultural sector. By empowering farmers with direct access to markets, the app will contribute to increased incomes, enhanced market transparency, improved efficiency, and sustainable agriculture practices. This project has the potential to create a more equitable and resilient food system, benefiting farmers, consumers, and the environment.

1.1 PURPOSE

The main goal of this mobile app is to empower farmers by giving them direct access to markets. This helps them sell their produce directly to consumers, retailers, or other buyers. By cutting out the middlemen, farmers can potentially increase their income, get better prices for their crops, and have more control over their sales.

1.2 PROJECT SCOPE

The project will cover the development of a mobile app with these key features:

- User Registration and Profiles: Farmers and buyers will be able to create accounts and set up profiles with their information.
- Product Listings: Farmers can list their products with details like the type of crop, quantity, price, and pictures.
- Search and Filtering: Buyers can search for products based on different criteria like crop type, location, and price.
- Direct Communication: The app will allow farmers and buyers to communicate directly to negotiate prices and arrange sales.
- Order Management: There will be features for managing orders, including tracking the status of sales and deliveries.
- Payment Integration: Secure payment options will be included to handle transactions.
- Reviews and Ratings: Buyers and farmers can rate each other to build trust and transparency.
- Notifications: The app will send notifications for new listings, messages, order updates, and payment confirmations.

1.3 PRODUCT FEATURES.

For Farmers:

- Easy-to-use interface for listing products: This means a clean, intuitive design with large buttons, clear instructions, and minimal steps to create a listing. The app should guide farmers through the process, perhaps with templates or pre-filled options for common crops.

- Tools for managing inventory and tracking sales: The app should allow farmers to easily update their available stock, mark items as sold, and generate reports on their sales. This could include features like automatic deduction of inventory when a sale is made.

- Real-time price updates and market information: This is crucial. The app needs to pull data from various sources to provide up-to-the-minute price information for different crops in their local markets. This could also include information on demand, weather forecasts, and competitor pricing.

- Direct messaging with buyers: This feature should allow farmers to communicate directly with potential buyers, answer questions, negotiate prices, and arrange for delivery. The messaging system should be reliable and user-friendly, with features like read receipts.

- Payment tracking and transaction history: The app should provide a clear record of all transactions, including payment status, amounts, and dates. Farmers should be able to easily see their earnings, track outstanding payments, and generate reports for their records.

For Buyers:

- Advanced search and filtering options: Buyers should be able to search for specific crops, filter by location, price, quality, and other relevant criteria. The app should provide a range of filtering options to help buyers find exactly what they need.

- Ability to compare prices and products: Buyers should be able to compare prices from different farmers and view details about the products. This could include

features like a side-by-side comparison of prices, quality, and other relevant information.

- Direct messaging with farmers: This feature allows buyers to communicate directly with farmers, ask questions, negotiate prices, and arrange for delivery. The messaging system should be reliable and user-friendly.

- Secure payment options: The app should offer secure payment options, such as mobile money, bank transfers, or credit card payments. The payment system should be reliable and protect both buyers and farmers from fraud.

- Reviews and ratings of farmers: This feature allows buyers to rate and review farmers based on their experience. This helps build trust and allows other buyers to make informed decisions.

Additional Features:

- Multilingual support: The app should be available in multiple languages to cater to a wider audience. This includes the user interface, instructions, and customer support.

- Offline access to basic information: The app should allow users to access basic information, such as crop listings and market information, even without an internet connection. This is particularly important in areas with limited connectivity.

- Customer support and FAQs: The app should provide customer support and a comprehensive FAQ section to help users with any questions or issues they may have. This could include features like in-app chat support or a phone number.

This project aims to create a comprehensive and user-friendly platform that empowers farmers and buyers, fostering economic growth and improving market efficiency.

CHAPTER 2

WORKS DONE IN THE RELATED AREA

1. e-NAM (Electronic National Agriculture Market):

The e-NAM initiative, launched by the Government of India, aims to create a unified national market for agricultural commodities. By integrating existing agricultural markets across states, e-NAM facilitates online trading of farm produce. The platform allows farmers to register and list their products for sale, enabling them to reach a larger audience of buyers, including wholesalers and retailers.

Farmers can view real-time prices for various commodities and make informed decisions about when and where to sell their produce. The app also ensures transparency in pricing, reducing the chances of exploitation by middlemen. Additionally, e-NAM provides logistical support, helping farmers with transportation and storage solutions. Overall, this initiative empowers farmers by giving them more control over their sales and improving their income potential.

2. Kisan Suvidha:

Kisan Suvidha is a comprehensive mobile app developed to assist farmers in various aspects of agriculture. The app provides critical information such as weather forecasts, market prices, and expert advice on crop management. By receiving timely updates on weather conditions, farmers can plan their sowing and harvesting activities more effectively, minimizing losses due to unforeseen weather changes.

The market price feature allows farmers to compare prices of their produce across different markets, helping them choose the most profitable options. Furthermore, the app includes a section for expert advice, where farmers can consult agronomists and agricultural scientists for guidance on best practices.

Kisan Suvidha also promotes government schemes and subsidies available to farmers, ensuring they are aware of all support systems in place.

3. IFFCO Kisan Agriculture App:

The IFFCO Kisan Agriculture App is designed to provide farmers with a wealth of information related to agriculture. It covers various topics, including crop management, soil health, pest control, and market prices. The app aims to educate farmers on sustainable farming practices while also providing them with the tools to enhance productivity.

One of the standout features of the IFFCO Kisan app is its advisory services. Farmers can access expert opinions and recommendations tailored to their specific needs, whether they are dealing with crop diseases or seeking advice on fertilization. The app also includes a marketplace where farmers can connect with buyers directly, reducing their reliance on intermediaries and ensuring they receive a fair price for their produce.

4. Reuters Market Light:

Reuters Market Light is a mobile app that delivers real-time information to farmers, helping them make informed decisions in a rapidly changing market environment. The app provides updates on market prices for various crops, allowing farmers to track price fluctuations and identify the best times to sell.

In addition to market prices, the app offers weather updates, which are crucial for farmers in planning their agricultural activities. Reuters Market Light also features news articles and insights related to agriculture, keeping farmers informed about industry trends and developments. The combination of market data and timely information helps farmers reduce risks and maximize their profits.

5. FarmGhar:

FarmGhar is a unique mobile app that connects farmers directly with consumers and businesses looking to purchase agricultural products. By eliminating intermediaries, FarmGhar enables farmers to sell their produce at competitive prices, ultimately improving their profit margins.

The app provides a user-friendly interface for farmers to list their products, including details such as quantity, price, and quality. Consumers can browse through available products and make purchases directly from farmers. This direct connection fosters a sense of community and trust between farmers and consumers, encouraging sustainable practices and supporting local economies. Additionally, FarmGhar includes features for farmers to manage their sales, track inventory, and receive payments securely.

6. e-Mandi:

e-Mandi is designed to revolutionize the way farmers sell their produce by connecting them directly to buyers, such as wholesalers, retailers, and consumers. This app eliminates the need for intermediaries, which often take a significant cut from the profits. Farmers can register on the platform, list their products, and set prices based on real-time market data. The app features a user-friendly interface that allows farmers to easily navigate through various options, including listing their produce, viewing demand trends, and receiving notifications about buyer inquiries. Additionally, e-Mandi can include features like secure payment options, logistics support for delivery, and a rating system to build trust between farmers and buyers. By providing transparency in pricing and reducing transaction costs, e-Mandi aims to increase farmers' income while ensuring consumers have access to fresh produce.

7. Farm Connect:

Farm Connect focuses on creating a marketplace where farmers can directly sell their products to consumers. This app allows farmers to create profiles, showcase their produce, and manage orders efficiently. Buyers can browse through different categories, filter products based on availability, and place orders directly through the app. The platform can also facilitate bulk purchases for restaurants and retailers, ensuring that farmers can reach larger markets. To enhance user engagement, Farm Connect can incorporate features like customer reviews, recipe suggestions using

the purchased produce, and seasonal promotions. The app may also provide educational content for farmers on best practices for marketing their products effectively. By fostering a direct relationship between farmers and consumers, Farm Connect aims to empower farmers and promote local food systems.

8. Krishi Bazaar:

Krishi Bazaar is a comprehensive platform that not only enables direct sales but also offers valuable insights to farmers. The app can provide market analysis tools that help farmers understand demand trends, pricing fluctuations, and competitor analysis. Farmers can access data on which crops are currently in demand and adjust their planting strategies accordingly. The app can also feature a forum for farmers to share experiences, ask questions, and learn from each other. Additionally, Krishi Bazaar can include educational resources, such as videos and articles on sustainable farming practices, pest management, and crop rotation techniques. By combining direct market access with educational support, Krishi Bazaar aims to enhance the overall productivity and profitability of farmers.

9. Kisan Suvidha:

Kisan Suvidha is designed to be a one-stop solution for farmers, providing them with essential information to make informed decisions. The app can feature weather forecasts tailored to specific regions, helping farmers plan their planting and harvesting schedules effectively. In addition to weather updates, Kisan Suvidha can provide real-time market prices for various crops, enabling farmers to choose the right time to sell their produce. The app can also include expert advice from agricultural specialists, covering topics such as soil health, crop diseases, and irrigation techniques. By integrating these features, Kisan Suvidha empowers farmers with knowledge and tools to enhance their productivity and profitability.

10. AgriTrade:

   AgriTrade focuses on creating a seamless trading experience for farmers looking to sell their produce directly to consumers or businesses. The app can facilitate user-friendly transactions, allowing farmers to list their products, negotiate prices, and finalize sales all within the platform. To ensure transparency, AgriTrade can implement a secure payment system that protects both parties. The app may also offer features like inventory management, sales analytics, and customer relationship management tools to help farmers grow their businesses. Additionally, AgriTrade can encourage local sourcing by promoting farm-to-table initiatives and connecting farmers with local restaurants and grocery stores. By simplifying the trading process, AgriTrade aims to enhance farmers' market access and foster a sustainable agricultural ecosystem.

CHAPTER 3

SYSTEM ANALYSIS

By clearly defining these system requirements, we can create a app that not only meets the needs of farmers but also convenient for consumers. This structured approach will facilitate effective development, testing, and deployment, ultimately leading to a successful product that enhances the experience.

3.1 USER REQUIREMENTS

1. User-Friendly Interface:

A user-friendly interface is crucial for ensuring that users can navigate the app easily. The design should be intuitive, with clearly labelled buttons and a straightforward layout. This will help users, including those who may not be tech-savvy, to quickly learn how to use the app. Features such as a tutorial or help section can further assist users in understanding how to utilize the app effectively.

2. Market Information:

Providing real-time market information is essential for users to make informed decisions. The app should display current prices of various crops and products, along with historical data to help users identify trends. Additionally, information about the demand for specific products in different markets can guide farmers in planning their production and sales strategies.

3. Direct Communication:

Enabling direct communication between farmers and buyers is a key feature. The app should include a messaging system or chat feature that allows users to negotiate prices,

discuss product details, and finalize sales without intermediaries. This direct line of communication can help build trust and foster better relationships between farmers and buyers.

4. Transaction Management:

A secure transaction management system is vital for facilitating sales. The app should support various payment methods, including mobile wallets and bank transfers, ensuring that transactions are safe and efficient. Users should also be able to track their orders, view transaction history, and receive confirmations for completed sales.

5. Logistics Support:

To aid in the delivery of products, the app should offer logistics support. This could include partnerships with local transport services or features that allow users to arrange for transportation directly through the app. Providing information on delivery options and costs can help farmers plan their sales more effectively.

6. Multilingual Support:

Given the diversity in language among users, the app should support multiple languages. This feature will make the app accessible to a broader audience, ensuring that language barriers do not hinder farmers from using the app effectively.

7. Offline Access:

Many farmers may not have consistent internet access, so the app should allow users to access essential information offline. This could include downloadable market data or the ability to view previously loaded content without an internet connection.

8. Notifications and Alerts:

Real-time notifications about market updates, price changes, and transaction statuses are important for keeping users informed. The app should allow users to customize their notification preferences to receive alerts that are most relevant to their interests and needs.

9. Data Security:

Ensuring the security and privacy of user data is paramount. The app should implement strong encryption methods and secure login processes to protect sensitive information. Users should also be informed about how their data will be used and have the option to control their privacy settings.

10. Feedback and Support:

A feedback mechanism is essential for continuous improvement. The app should provide users with an easy way to submit feedback or report issues. Additionally, a customer support feature, such as a help desk or FAQ section, can assist users in resolving any problems they encounter while using the app.

11. Profile Management:

Farmers should be able to manage their profiles easily, including updating contact information and banking details for transactions. A straightforward profile management system can help ensure that all information is current and accurate.

3.2 HARDWARE REQUIREMENTS

1. Laptop

    - Processor - AMD Ryzen 5 4600H
    - RAM - 4GB
    - Hard Disk - 100GB
    - Dedicated GPU
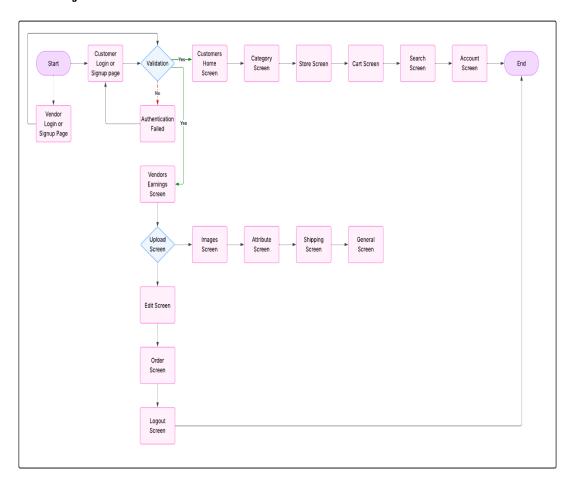
2. Android Phone

3.3 SOFTWARE REQUIREMENTS

- OS - Windows 11
- IDE - Android Studio
- Frontend – Flutter Framework
- Backend – Dart
- Database – Firebase Realtime Database
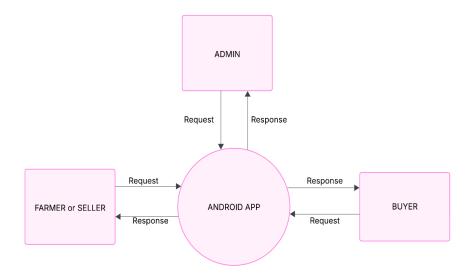
CHAPTER 4

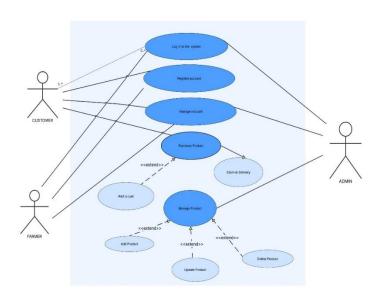# SYSTEM DESIGN AND SPECIFICATIONS

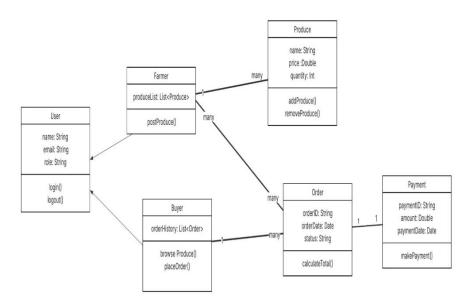## 4.1 HIGH LEVEL DESIGN

4.1.1 Project Model

4.1.2 DFD

(Level 0)

4.1.3 UML Diagram

1. Use Case Diagram

2. Class Diagram

4.2 LOW LEVEL DESIGN

4.2.1   Process Specification

1. User Registration and Authentication:

    - Farmers and buyers register with verified details.

    - Secure login with password and optional two-factor authentication.

2. Market Information:

    - Real-time price updates for various crops.

    - Market location and contact details.

    - Historical price trends and analysis.

3. Produce Listing and Management:

    - Farmers list produce with details (quantity, quality, price).

    - Ability to update or remove listings.

    - Categorization and search functionality.

4. Buyer Interaction:

    - Buyers can search for produce.

    - Direct messaging and negotiation features.

    - Option to place orders.

5. Order Management:

    - Farmers and buyers can track order status.

    - Notifications for order updates.

6. Payment Gateway:

- Integration with secure payment gateways.
- Transaction history and reports.

7. Communication:

- In-app messaging for communication.
- Push notifications for updates and alerts.

8. Location Services:

- Geolocation to find nearby markets and buyers.
- Mapping of market locations.

9. User Interface:

- Intuitive and user-friendly design.
- Multilingual support.
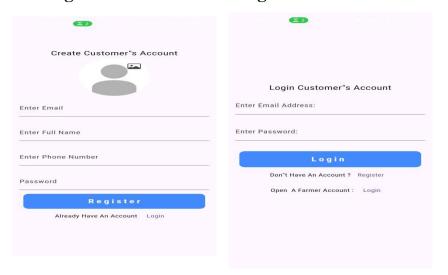- Accessibility features.

10. Data Security:

- Secure data storage and transmission.
- Compliance with privacy regulations.
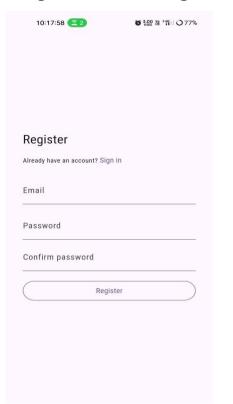
11. Feedback and Support:

- Feedback mechanisms for users.
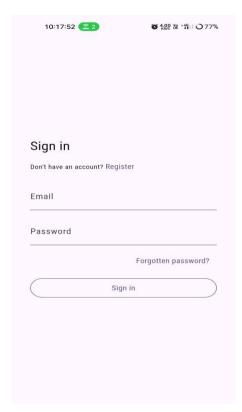- Customer support channels.

4.2.2 Screenshot Diagram

1.   Registration   and Login Screen   of   customer:-

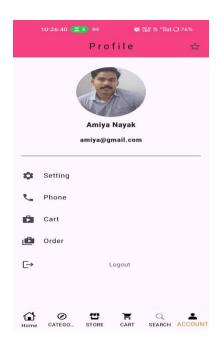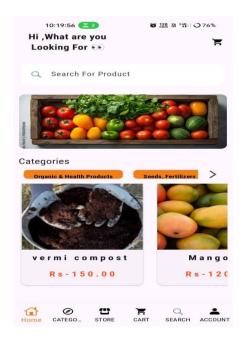2. . Registration and Login Screen of farmer : -

3. Item Screen : -

4. Profile Screen: -

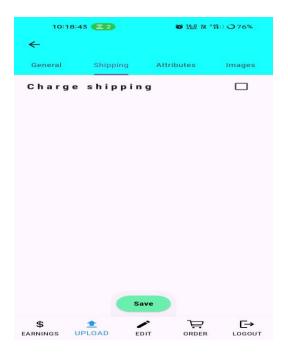## 5. Cart Screen: -

6. Home Screen: -

7. Category Screen:-
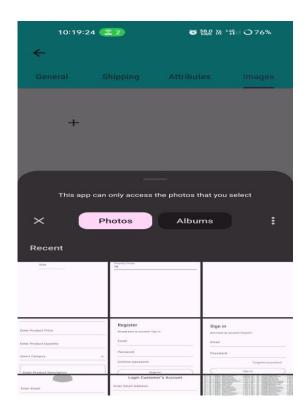

8. Store Screen:-

9. Upload Screen:-

10. Uploading images from gallery:-

24

CHAPTER 5

CODING

1.Admin Panel Code

```dart
import 'package:farmer_connect_web_admin/views/screens/side_bar_screens.dart/categories_screen.dart';
import 'package:farmer_connect_web_admin/views/screens/side_bar_screens.dart/dashboard_screen.dart';
import 'package:farmer_connect_web_admin/views/screens/side_bar_screens.dart/order_screen.dart';
import 'package:farmer_connect_web_admin/views/screens/side_bar_screens.dart/product_screen.dart';
import 'package:farmer_connect_web_admin/views/screens/side_bar_screens.dart/upload_banner_screen.dart';
import 'package:farmer_connect_web_admin/views/screens/side_bar_screens.dart/vensdors_screen.dart';
import 'package:farmer_connect_web_admin/views/screens/side_bar_screens.dart/withdrawl_screen.dart';
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:flutter_admin_scaffold/admin_scaffold.dart';

class MainScreen extends StatefulWidget {
  const MainScreen({super.key});

  @override
  State<MainScreen> createState() => _MainScreenState();
}

class _MainScreenState extends State<MainScreen> {
  Widget _selectedItem = DashboardScreen();

  screenSelector(item) {
    switch (item.route) {
      case DashboardScreen.routeName:
        setState(() {
          _selectedItem = DashboardScreen();
        });

        break;
      case VensdorsScreen.routeName:
        setState(() {
          _selectedItem = VensdorsScreen();
        });
        break;
      case WithdrawlScreen.routeName:
        setState(() {
          _selectedItem = WithdrawlScreen();
        });
        break;

      case OrderScreen.routeName:
        setState(() {
          _selectedItem = OrderScreen();
        });
        break;

      case CategoriesScreen.routeName:
        setState(() {
          _selectedItem = CategoriesScreen();
        });
        break;

      case ProductScreen.routeName:
        setState(() {
          _selectedItem = ProductScreen();
        });
        break;

      case UploadBannerScreen.routeName:
        setState(() {
          _selectedItem = UploadBannerScreen();
        });
        break;
    }
  }

  @override
  Widget build(BuildContext context) {
    return AdminScaffold(
      appBar: AppBar(
        title: Text(
          'Management',
        ), // Text
        backgroundColor: Colors.amber,
      ), // AppBar
      sideBar: SideBar(
        items: [
          AdminMenuItem(
            title: 'Dashoard',
            icon: Icons.dashboard,
            route: DashboardScreen.routeName,
          ), // AdminMenuItem
          AdminMenuItem(
            title: 'Vendors',
            icon: CupertinoIcons.person_3,
            route: VensdorsScreen.routeName,
          ), // AdminMenuItem
```

```
95          icon: CupertinoIcons.shopping_cart,
96          route: OrderScreen.routeName,
97        ), // AdminMenuItem
98        AdminMenuItem(
99          title: 'Categories',
00          icon: Icons.category,
01          route: CategoriesScreen.routeName,
02        ), // AdminMenuItem
03        AdminMenuItem(
04          title: 'Product',
05          icon: Icons.shop,
06          route: ProductScreen.routeName,
07        ), // AdminMenuItem
08        AdminMenuItem(
09          title: 'Upload Banner',
10          icon: CupertinoIcons.add,
11          route: UploadBannerScreen.routeName,
12        ), // AdminMenuItem
13      ],
14      selectedRoute: '/',
15      onSelected: (item) {
16        screenSelector(item);
17      },
18      header: Container(
19        height: 50,
20        width: double.infinity,
21        color: const Color(0xff444444),
22        child: const Center(
23          child: Text(
24            'Farmer-Connect AdminPanel',
25            style: TextStyle(
26              color: Colors.white,
27            ), // TextStyle
28          ), // Text
29        ), // Center
30      ), // Container
31      footer: Container(
32        height: 50,
33        width: double.infinity,
34        color: const Color(0xff444444),
35        child: const Center(
36          child: Text(
118      header: Container(
119        height: 50,
120        width: double.infinity,
121        color: const Color(0xff444444),
122        child: const Center(
123          child: Text(
124            'Farmer-Connect AdminPanel',
125            style: TextStyle(
126              color: Colors.white,
127            ), // TextStyle
128          ), // Text
129        ), // Center
130      ), // Container
131      footer: Container(
132        height: 50,
133        width: double.infinity,
134        color: const Color(0xff444444),
135        child: const Center(
136          child: Text(
137            'footer',
138            style: TextStyle(
139              color: Colors.white,
140            ), // TextStyle
141          ), // Text
142        ), // Center
143      ), // Container
144    ), // SideBar
145    body: _selectedItem
146  ); // AdminScaffold
147  }
148 }
149
```

26

2. Mobile app Main Screen code: -

```dart
1   import 'package:farmer_connect/views/buyers/nav_screens/account_screen.dart';
2   import 'package:farmer_connect/views/buyers/nav_screens/cart_screen.dart';
3   import 'package:farmer_connect/views/buyers/nav_screens/category_screen.dart';
4   import 'package:farmer_connect/views/buyers/nav_screens/home_screen.dart';
5   import 'package:farmer_connect/views/buyers/nav_screens/search_screen.dart';
6   import 'package:farmer_connect/views/buyers/nav_screens/store_screen.dart';
7   import 'package:flutter/cupertino.dart';
8   import 'package:flutter/material.dart';
9   import 'package:flutter_svg/flutter_svg.dart';
10
11  class MainScreen extends StatefulWidget {
12    const MainScreen({super.key});
13
14    @override
15    State<MainScreen> createState() => _MainScreenState();
16  }
17
18  class _MainScreenState extends State<MainScreen> {
19    int _pageIndex = 0;
20
21    // ignore: prefer_final_fields
22    List<Widget> _pages = [
23      HomeScreen(),
24      CategoryScreen(),
25      StoreScreen(),
26      CartScreen(),
27      SearchScreen(),
28      AccountScreen(),
29
30    ];
31    @override
32    Widget build(BuildContext context) {
33      return Scaffold(
34        bottomNavigationBar: BottomNavigationBar(
35          type: BottomNavigationBarType.fixed,
36          currentIndex: _pageIndex,
37          onTap: (value) {
38            setState(() {
39              _pageIndex = value;
40            });
41          },
42          unselectedItemColor: Colors.black,
43          selectedItemColor: Colors.yellow.shade900,
44          items: [
45            BottomNavigationBarItem(

45            BottomNavigationBarItem(
46                icon: Icon(CupertinoIcons.home), label: 'Home'), // BottomNavigationBarItem
47            BottomNavigationBarItem(
48                icon: SvgPicture.asset(
49                  'assets/icons/explore.svg',
50                  width: 20,
51                ), // SvgPicture.asset
52                label: 'CATEGORIES'), // BottomNavigationBarItem
53            BottomNavigationBarItem(
54                icon: SvgPicture.asset(
55                  'assets/icons/shop.svg',
56                  width: 20,
57                ), // SvgPicture.asset
58                label: 'STORE'), // BottomNavigationBarItem
59            BottomNavigationBarItem(
60                icon: SvgPicture.asset(
61                  'assets/icons/cart.svg',
62                  width: 20,
63                ), // SvgPicture.asset
64                label: 'CART'), // BottomNavigationBarItem
65            BottomNavigationBarItem(
66                icon: SvgPicture.asset(
67                  'assets/icons/search.svg',
68                  width: 20,
69                ), // SvgPicture.asset
70                label: 'SEARCH'), // BottomNavigationBarItem
71            BottomNavigationBarItem(
72                icon: SvgPicture.asset(
73                  'assets/icons/account.svg',
74                  width: 20,
75                ), // SvgPicture.asset
76                label: 'ACCOUNT'), // BottomNavigationBarItem
77
78          ]), // BottomNavigationBar
79
80          body: _pages[_pageIndex],
81        ); // Scaffold
82    }
83  }
84
```

3. Product Screen Code:-

```
1    import 'package:farmer_connect/provider/cart_provider.dart';
2    import 'package:flutter/cupertino.dart';
3    import 'package:flutter/material.dart';
4    import 'package:photo_view/photo_view.dart';
5    import 'package:intl/intl.dart';
6    import 'package:provider/provider.dart';
7
8    class ProductDetailScreen extends StatefulWidget {
9      final Map<String, dynamic>? productData;
10
11     const ProductDetailScreen({super.key, this.productData});
12
13     @override
14     State<ProductDetailScreen> createState() => _ProductDetailScreenState();
15   }
16
17   class _ProductDetailScreenState extends State<ProductDetailScreen> {
18     String formatedDate(date) {
19       final outPutDateFormat = DateFormat('dd/MM/yyyy');
20       final outPutDate = outPutDateFormat.format(date);
21       return outPutDate;
22     }
23
24     int _imageIndex = 0;
25     String? _selectedSize;
26
27     @override
28     Widget build(BuildContext context) {
29       final CartProvider _cartProvider = Provider.of<CartProvider>(context);
30       if (widget.productData == null) {
31         return Scaffold(
32           appBar: AppBar(title: Text('Product Detail')),
33           body: Center(child: Text('No product data available.')),
34         );
35       }
36
37       final imageUrls = widget.productData!['imageUrl'] as List<dynamic>? ?? [];
38       final sizeList = widget.productData!['sizeList'] as List<dynamic>? ?? [];
39
40       return Scaffold(
41         appBar: AppBar(
42           title: Text(
43             widget.productData!['productName'] ?? 'Product',
44             style: TextStyle(
45               fontSize: 20,
46               color: Colors.black,
47               fontWeight: FontWeight.bold,
48               letterSpacing: 4,
49             ),
50           ),
51         ),
52         body: SingleChildScrollView(
53           child: Column(
54             children: [
55               Stack(
56                 children: [
57                   Container(
58                     height: 300,
59                     width: double.infinity,
60                     child: PhotoView(
61                       imageProvider: NetworkImage(
62                         imageUrls.isNotEmpty && _imageIndex < imageUrls.length
63                           ? imageUrls[_imageIndex]
64                           : '')),
65                   ),
66                   Positioned(
67                     bottom: 0,
68                     child: Container(
69                       height: 50,
70                       width: MediaQuery.of(context).size.width,
71                       child: ListView.builder(
72                         scrollDirection: Axis.horizontal,
73                         itemCount: imageUrls.length,
74                         itemBuilder: (context, index) {
75                           return InkWell(
76                             onTap: () {
77                               setState(() {
78                                 _imageIndex = index;
79                               });
80                             },
81                             child: Padding(
82                               padding: const EdgeInsets.all(8.0),
83                               child: Container(
84                                 decoration: BoxDecoration(
85                                   border: Border.all(
86                                     color: Colors.yellow.shade900),
87                                 ),
88                                 height: 60,
89                                 width: 60,
90                                 child: Image.network(imageUrls.isNotEmpty
91                                   ? imageUrls[index]
```

\pikun\Desktop\farmer_connect\lib\views\buyers\productDetail

```
 92                                    | :  ''),
 93                                  ),
 94                                ),
 95                              );
 96                            }),
 97                        ))
 98                    ],
 99                  ),
100                  SizedBox(height: 20),
101                  Padding(
102                    padding: const EdgeInsets.all(13.0),
103                    child: Text(
104                      widget.productData!['productName'] ?? 'Product Name',
105                      style: TextStyle(
106                        fontSize: 22,
107                        color: const Color.fromARGB(255, 3, 1, 12),
108                        fontWeight: FontWeight.bold,
109                      ),
110                    ),
111                  ),
112                  Text(
113                    'Rs-' +
114                        (widget.productData!['productPrice']?.toStringAsFixed(2) ??
115                            '0.00'),
116                    style: TextStyle(
117                      fontSize: 22,
118                      color: const Color.fromARGB(255, 227, 5, 34),
119                      fontWeight: FontWeight.bold,
120                    ),
121                  ),
122                  ExpansionTile(
123                    title: Row(
124                      mainAxisAlignment: MainAxisAlignment.spaceBetween,
125                      children: [
126                        Text(
127                          'Description:',
128                          style: TextStyle(color: Colors.yellow.shade900),
129                        ),
130                        Text(
131                          'ViewMore',
132                          style: TextStyle(color: Colors.yellow.shade900),
133                        ),
134                      ],
135                    ),
136                    children: [
137                      Padding(
138                        padding: const EdgeInsets.all(8.0),
139                        child: Text(
140                          widget.productData!['description'] ??
141                              'No description available.',
142                          style: TextStyle(fontSize: 17),
143                          textAlign: TextAlign.center,
144                        ),
145                      )
146                    ],
147                  ),
148                  Padding(
149                    padding: const EdgeInsets.all(8.0),
150                    child: Row(
151                      mainAxisAlignment: MainAxisAlignment.spaceEvenly,
152                      children: [
153                        Text(
154                          'This Product Will be Shipping on ',
155                          style: TextStyle(fontWeight: FontWeight.bold, fontSize: 18),
156                        ),
157                        Text(
158                          widget.productData!['scheduleDate'] != null
159                              ? formatedDate(
160                                  widget.productData!['scheduleDate'].toDate())
161                              : 'No shipping date available.',
162                          style: TextStyle(
163                            fontSize: 18,
164                            fontWeight: FontWeight.bold,
165                            color: Colors.blueAccent,
166                          ),
167                        ),
168                      ],
169                    ),
170                  ),
171                  ExpansionTile(
172                    title: Text(
173                      'Available Size',
174                      style: TextStyle(
175                        fontSize: 18,
176                        fontWeight: FontWeight.bold,
177                      ),
```

29

```
220                    widget.productData!['productPrice'],
221                    widget.productData!['vendorId'],
222                    _selectedSize!,
223                    widget.productData!['scheduleDate']);
224              } else {
225                // Handle case where no size is selected
226                ScaffoldMessenger.of(context).showSnackBar(
227                  SnackBar(content: Text('Please select a size.')),
228                );
229              }
230            },
231            child: Container(
232              height: 50,
233              width: MediaQuery.of(context).size.width,
234              decoration: BoxDecoration(
235                color: Colors.deepOrange,
236                borderRadius: BorderRadius.circular(30),
237              ), // BoxDecoration
238              child: Row(
239                mainAxisAlignment: MainAxisAlignment.center,
240                children: [
241                  Padding(
242                    padding: const EdgeInsets.all(8.0),
243                    child: Icon(
244                      CupertinoIcons.cart,
245                      color: Colors.white,
246                      size: 25,
247                    ), // Icon
248                  ), // Padding
249                  Text(
250                    'Add To Cart',
251                    style: TextStyle(
252                      fontSize: 18,
253                      color: Colors.white,
254                      fontWeight: FontWeight.bold,
255                    ), // TextStyle
256                  ), // Text
257                ],
258              ), // Row
259            ), // Container
260          ), // InkWell
261        ), // Padding
262      ); // Scaffold
263    }
264  }
265
```
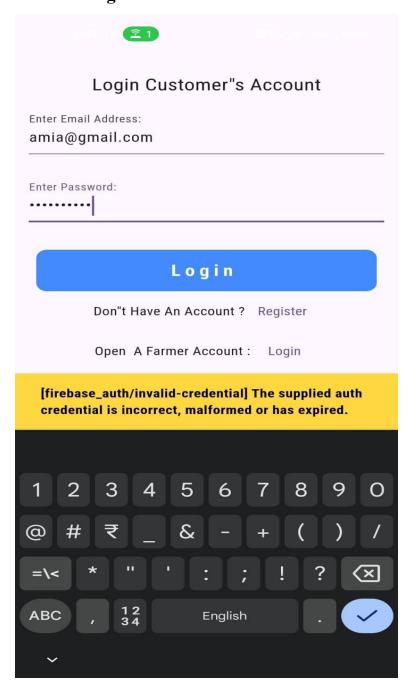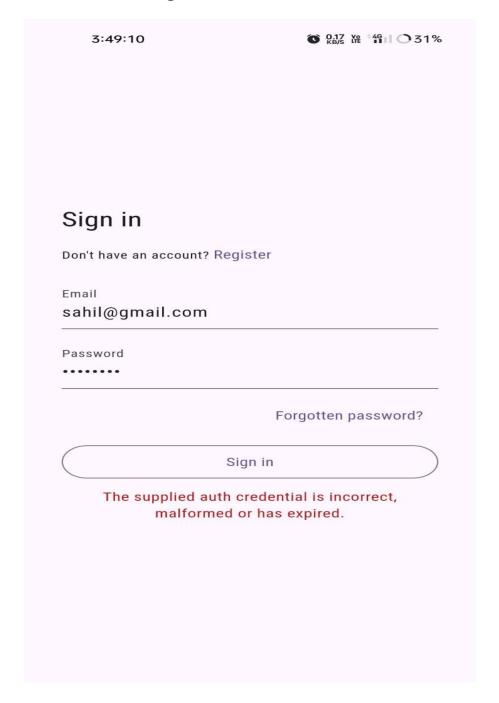
CHAPTER 6

TESTING

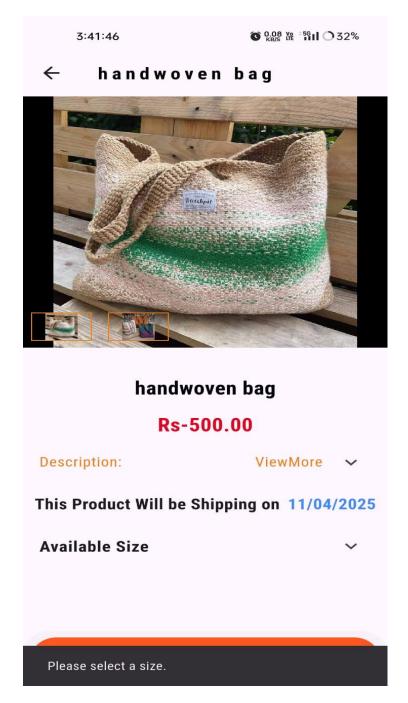## 1. When the username or password field is empty: -

2. When user gives invalid credentials: -

3. When Farmer gives invalid credentials: -

4. When user does not select the correct size:-

CHAPTER 7

CONCLUSIONS AND LIMITATIONS

7.1 LIMITATIONS

While a mobile app for Direct Market Access (DMA) offers numerous benefits to farmers, it's essential to acknowledge its limitations. These can include issues like digital literacy and access to smartphones and reliable internet, which can create a digital divide. Additionally, the app's effectiveness may be hampered by the need for robust data security to protect sensitive information and the potential for price manipulation. Furthermore, the app's success hinges on consistent updates, user-friendly interfaces, and ongoing support to address technical glitches and user queries. Understanding these constraints is crucial for the app's developers, users, and stakeholders to ensure its effective and equitable implementation.

1. Digital Literacy and Infrastructure:

One of the primary limitations of a mobile app for Direct Market Access for farmers revolves around digital literacy and the availability of reliable infrastructure. While smartphones have become increasingly common, especially in rural areas, a significant portion of farmers may lack the necessary digital skills to effectively use the app. This includes basic smartphone operation, understanding app navigation, and the ability to interpret data and information presented on the app. Moreover, the app's usability hinges on the availability of consistent and affordable internet access. Many rural areas, particularly in developing countries, suffer from poor network coverage and unstable

internet connectivity. This can lead to delayed updates, difficulty in accessing real-time market information, and problems in completing transactions.

Furthermore, the cost of data and smartphones can be a barrier for some farmers. While the price of smartphones has decreased, they still represent a significant investment for low-income farmers. Similarly, the cost of data plans can be prohibitive, especially if the app requires frequent updates or the transmission of large amounts of data, such as images of produce. Without adequate digital literacy training and affordable access to both devices and internet, the app's potential benefits may not be realized by a large segment of the target audience. Addressing this limitation requires a multi-pronged approach, including providing digital literacy training programs tailored to farmers' needs, partnering with telecom providers to offer affordable data plans, and ensuring the app is designed with a user-friendly interface that minimizes the need for advanced digital skills. Additionally, offline functionality, where farmers can access some information even without an internet connection, can be a crucial feature.

2. Trust, Security, and Payment Systems:

Building trust and ensuring the security of transactions are critical limitations. Farmers may be hesitant to use an app if they do not trust the platform or are concerned about the security of their personal and financial information. This lack of trust can stem from various factors, including past experiences with scams, a general lack of trust in digital platforms, and concerns about data privacy. The app must therefore implement robust security measures to protect user data and prevent fraudulent activities. This includes secure authentication protocols, encryption of sensitive information, and regular security audits.

Another significant challenge is integrating secure and reliable payment systems. Farmers need a convenient and trustworthy way to receive payments for their produce. This requires the app to support various payment options, such as mobile money, bank transfers, and potentially even cash-on-delivery, depending on the local context. However, integrating these payment systems can be complex, involving partnerships with financial institutions and ensuring compliance with relevant regulations. Furthermore, there is a need

to educate farmers about the different payment methods and how to use them securely. Building trust also involves transparency in the app's operations. Farmers need to understand how the app works, how their data is used, and how disputes are resolved. Clear communication, accessible customer support, and a strong reputation are essential for building trust and encouraging adoption of the app. The app should also have mechanisms for dispute resolution, such as a dedicated customer service team or an automated dispute resolution system, to address any issues that may arise during transactions.

3. Market Volatility and Price Discovery:

The dynamic nature of agricultural markets and the challenge of providing accurate and timely price information represent another significant limitation. Agricultural prices are subject to various factors, including seasonality, weather conditions, supply and demand dynamics, and market speculation. Providing farmers with up-to-date and accurate price information is crucial for enabling them to make informed decisions about when and where to sell their produce. However, this is a complex task. Market prices can change rapidly, and the app needs to be able to capture and disseminate this information in real-time. This requires data feeds from various sources, including wholesale markets, commodity exchanges, and other relevant market participants.

The app also needs to address the challenge of price discovery. Farmers may not always be aware of the best prices available for their produce. The app can help by providing price comparisons, market analysis, and information about potential buyers. However, the accuracy and reliability of this information are crucial. If the price information is inaccurate or outdated, farmers may make poor decisions, leading to financial losses. Furthermore, the app should consider the heterogeneity of agricultural products. Different varieties of the same crop may

command different prices, and the app needs to be able to differentiate between them. This may involve providing detailed information about product quality, grading

standards, and other relevant factors. In addition, the app should also consider the logistical challenges of transporting produce to different markets. Transportation costs can significantly impact the profitability of a sale, and the app should provide information about transportation options and costs.

Expanding on the limitations, it's important to consider infrastructure challenges such as the availability of electricity for charging devices and the physical distance to markets, which can impact the app's practicality. Moreover, the app's functionality may be limited by the types of crops and market demands it supports, potentially excluding certain farmers. The app's success also relies on the integration with existing agricultural practices and the willingness of farmers to adopt new technologies. Finally, the app's impact on existing market dynamics, including potential displacement of traditional intermediaries, needs careful consideration to avoid unintended consequences.

7.2 CONCLUSION

The proposed smart mobile application offers a transformative solution to the challenges faced by farmers in accessing markets and optimizing profits. By facilitating direct transactions between farmers, consumers, and retailers, the platform eliminates intermediaries and ensures greater control, transparency, and efficiency within the agricultural supply chain. Using GPS, machine learning, and secure payment systems, the application not only boosts farmers' earnings and reduces wastage but also enhances market accessibility. With promising results from the pilot study, the application holds significant potential for revolutionizing the agricultural marketplace. Future developments, such as blockchain integration and advanced AI-driven demand forecasting, will further enhance the system's effectiveness, making it a powerful tool for sustainable agricultural growth world.

BIBLIOGRAPHY

- "Small Farmers, Big Impact: A Guide to Direct Market Access" - Food and Agriculture Organization of the United Nations (FAO) - FAO publications often provide detailed insights on agricultural practices, including the impact of direct market access on farmers' livelihoods.

- "Direct Marketing Strategies for Farmers and Food Entrepreneurs" - Sustainable Agriculture Research & Education (SARE) - This guide discusses effective direct marketing strategies for small farmers, focusing on reaching consumers and increasing profit margins.

- "Direct Marketing Channels and the Development of Farmers' Markets" - Journal of Food Distribution Research - Research papers in this journal often analyse the economic impact of direct marketing strategies and the success factors of farmers' markets.

- "Empowering Farmers through Direct Market Access: Case Studies from India" - National Institute of Agricultural Extension Management (MANAGE) - This report provides case studies on how direct access is transforming small farming communities in India.

- "Farmers Markets as a Strategy for Increasing Access to Healthy Food for Low- Income Communities and Supporting Farmers" - Centres for Disease Control and Prevention (CDC) - This resource explains how direct market access promotes both public health and economic opportunities for farmers.

"Farm Fresh: Direct Marketing and Agricultural Development" - World Bank Reports - World Bank reports provide case studies and insights into how direct market access affects agricultural productivity in developing economies.