

Software Requirements Specification

Farmalink

Prepared By:

Ahammad Ullah (201014077)

Morium Begum (201014072)

Prepared For:

Shakib Mahmud Diplo

Lecturer

Department of Computer Science and Engineering

Date:

25 December, 2023

Table of Contents

Revision History

1. Introduction	1
1.1 Purpose	
1.2 Intended Audience and Reading Suggestions	
1.3 Product Scope	
1.4 References	
2. Overall Description	1-2
2.1 Product Perspective	
2.2 Product Functions	
2.3 User Classes and Characteristics	
2.4 Operating Environment	
2.5 Design and Implementation Constraints	
2.6 User Documentation	
2.7 Assumptions and Dependencies	
3. External Interface Requirements	2-3
3.1 Farmers Interfaces	
3.2 Suppliers Interfaces	
3.3 Consumers Interfaces	
3.4 Admins Interfaces	
4. System Features	4-5
4.1 System Features 1	
5. Other Nonfunctional Requirements	5
5.1 Performance Requirements	
5.2 Safety Requirements	
5.3 Security Requirements	
6. Other Requirements	6
Appendix A: Class Diagram	

1.Introduction

1.1.Purpose

The purpose of this document is to provide a list of software requirements for the development of the "FarmLink" app, an agricultural platform that connects farmers with suppliers and consumers to enhance the efficiency, productivity and sustainability of farming operations.

1.2.Intended Audience and Reading Suggestions

This document is intended for project managers, developers, testers and stakeholders involved in the app development. It should be read in the context of comprehending the features, limitations, and functionality of the app.

1.3.Product Scope

The "FarmLink" app's scope encompasses user registration, crop management, communication between farmers, suppliers, and consumers. It serves as a bridge among farmers and others like consumers and suppliers so that farmers can get their proper prices for their products as well as sell their produced crops at a fair price. For the consumers, they get the chance to buy fresh farm products from farmers directly at a reasonable price. Moreover the suppliers can advertise their products like seeds, pesticides, insecticides and set a reasonable price for the farmers.

1.4.References

- a. https://agrimart.in/index.php/home/about_us
- b. <https://www.cropin.com/about>

2.Overall Description

2.1.Product Perspective

The "FarmLink" app exists within the larger context of the agricultural ecosystem. It connects farmers, suppliers, and consumers, streamlining the process of agricultural inputs supply and crop sales.

2.2.Product Functions

- User (farmers, suppliers, consumer) registration and authentication
- Farmer and supplier profile management
- Agricultural inputs listing and management
- Agricultural inputs search and browsing

- Agricultural inputs purchase and sales
- Crop listing and management
- Crop search and browsing
- Crop purchase and sales
- Messaging and notifications
- Admin panel for app management

2.3.User Classes and Characteristics

- Farmers: Registered users who list and manage crops.
- Suppliers: Registered users who supply agricultural inputs.
- Consumers: Registered users who purchase crops.
- Admins: App administrators responsible for managing users and content.

2.4.Operating Environment

The app is designed for both Android and iOS platforms, ensuring that any mobile device with internet connection can use it.

2.5.Design and Implementation Constraints

- The project must adhere to a defined budget and timeframe.
- The app will be developed using a specific technology stack (Figma).
- Third-party APIs may be used for functionalities like payment processing.

2.6.User Documentation

User documentation, including user manuals and help guides, will be provided to ensure users can effectively navigate and use the app.

2.7.Assumptions and Dependencies

Assumptions include reasonable user behavior, system availability, and data accuracy. Dependencies may include external APIs for weather information and internet connectivity.

3.External Interface Requirements

3.1.Farmers Interfaces

- Registration and profile management screens.
- Crop listing and management screens.
- Product browsing and purchasing interfaces.
- Messaging and notifications interfaces.

3.2.Suppliers Interfaces

- Registration and profile management screens.
- Product listing and management interfaces.
- Messaging and notifications interfaces.

3.3.Consumers Interfaces

- Registration and profile management screens.
- Crop browsing and purchasing interfaces.
- Messaging and notifications interfaces.

3.4.Admins Interfaces

- User management screens.
- Content moderation tools.

4.System Features

4.1.System Features 1

4.1.1.User Registration and Authentication

- User registration with options for email or phone number.
- Secure user authentication.

4.1.2.Farmer and Supplier Profile Management

- Profile creation and Profile management.
- Update and edit personal and farm information.

4.1.3.Agricultural Inputs Listing and Management

- Suppliers can list Agricultural inputs, including type, quantity, pricing, and images.
- Inputs editing and removal functionalities.

4.1.4.Agricultural Inputs Search and Browsing

- Farmers can search, filter, and browse Agricultural Inputs listings based on various criteria.

4.1.5.Agricultural Inputs Purchase and Sales

- Farmers can purchase Agricultural Inputs
- Suppliers can confirm and manage sales.

4.1.6.Crop Listing and Management

- Farmers can list crops, including crop type, quantity, pricing, and images.
- Crop editing and removal functionalities.

4.1.7.Crop Search and Browsing

- Users can search, filter, and browse crop listings based on various criteria.

4.1.8.Crop Purchase and Sales

- Consumers can purchase crops; Farmers can confirm and manage sales.

4.1.9.Messaging and Notifications

- In-app messaging for user communication.
- Notifications for new listings, inquiries, and updates.

4.1.10.Admin Panel for App Management

- Functions and permissions for app administrators, including user management and content moderation.

5.Other Nonfunctional Requirements

5.1.Performance Requirements

- There must be a specific timeframe to provide the responses.
- High possibilities of availability of services when needed and minimal downtime are required.

5.2.Safety Requirements

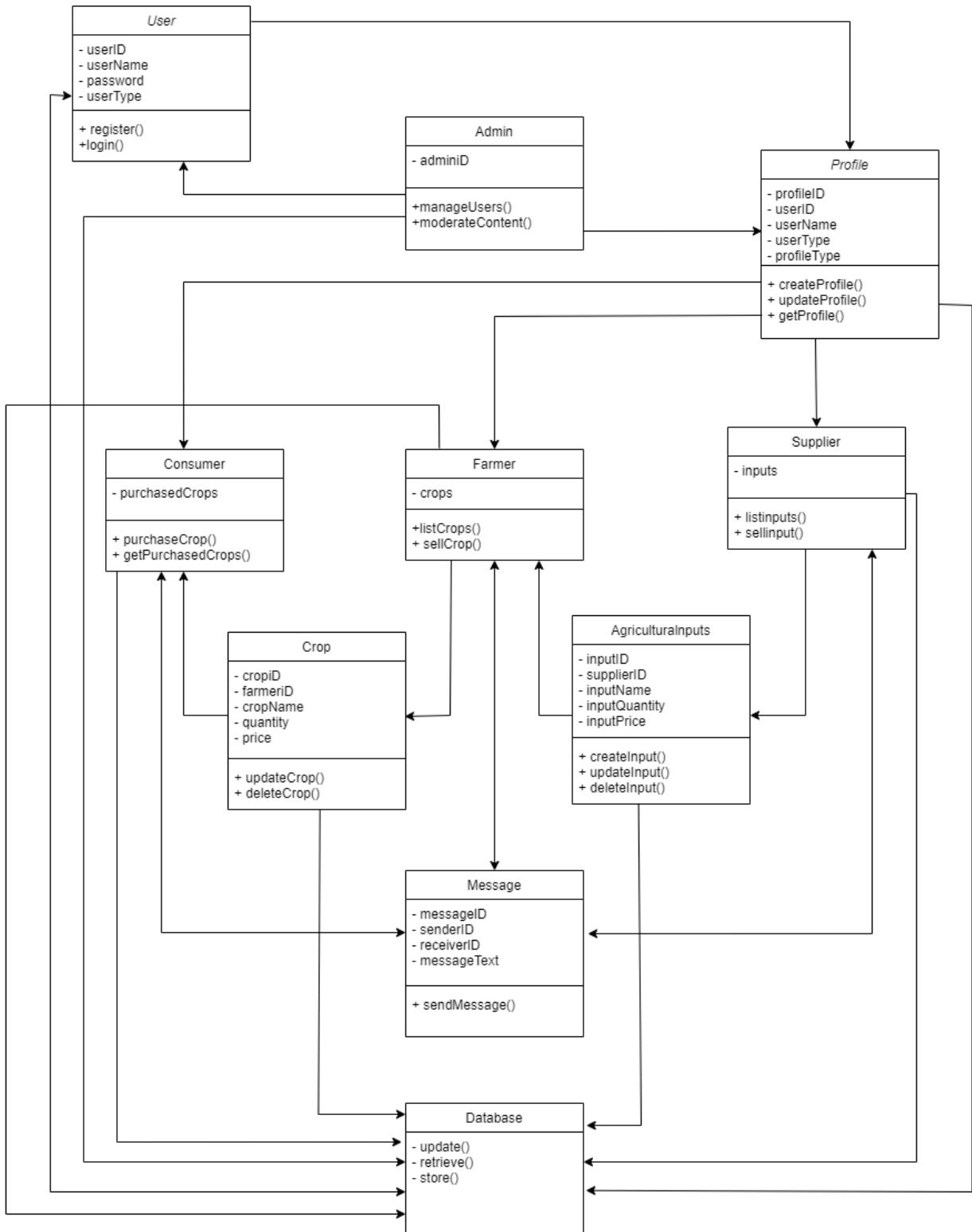
- Personal information of users' needs to be kept secure; that's how the user's privacy must be ensured.
- A set of rules and requirements must be followed by the users and defined as well as monitored by the admins.

5.3.Security Requirements

- Any kind of information of users must be encrypted and protected.
- Authentication process must be checked by the admins.

6. Other Requirements

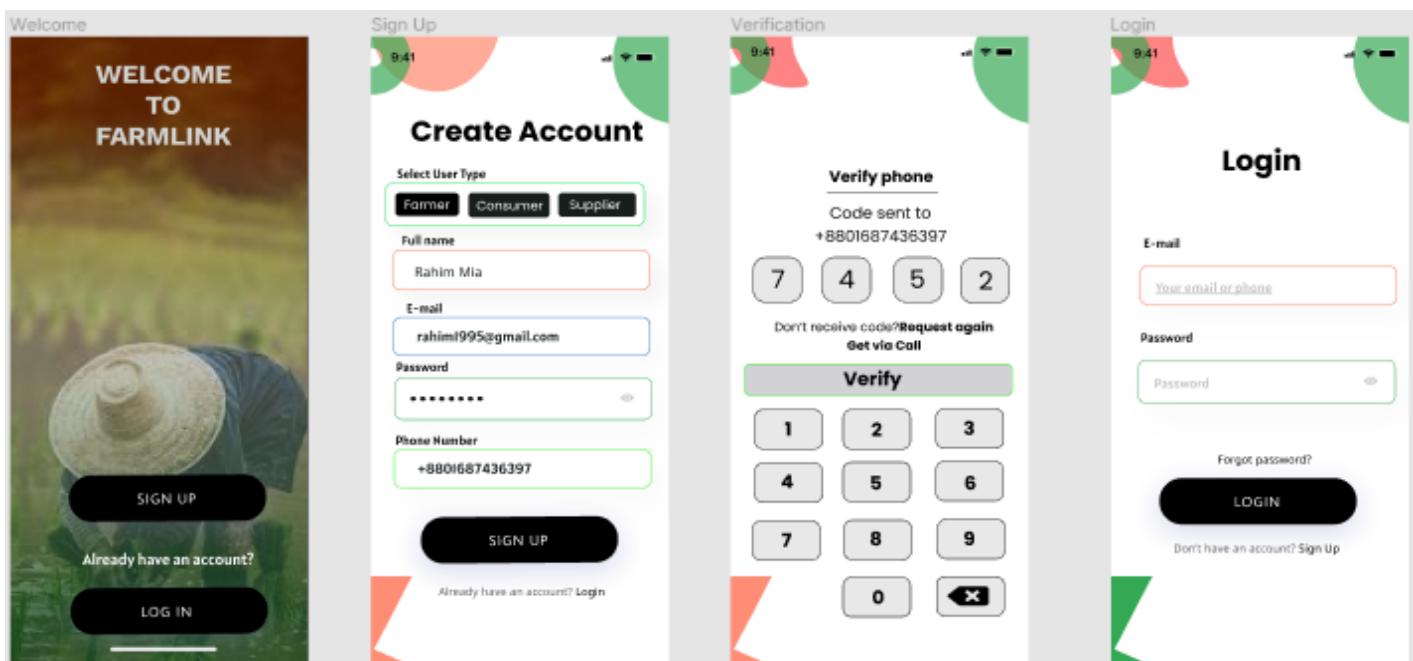
Appendix A: Class Diagram



Timeline:

Oct				Nov				Dec				Jan				Feb				March				
W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4	
Proposal	Idea Selection																							
		SRS and Class Diagram				Welcome & Signup Page						Login, Homepage & Add to Cart		Profile			Consumers Interface		Suppliers Interface			Testing		
																							Software Release	

Outline:



Home Screen Farmer

Farmer's Cart

Farmers Orders Screen

Farmer Profile

Home Screen Consumer

Consumer's Cart

Consumer Profile

Code:

SnapShot of main.dart file:

```
import 'package:flutter/material.dart';
import 'register.dart';
import 'login.dart'
void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: WelcomePage(),
    );
  }
}

class WelcomePage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Stack(
        children: [
          // Background Image
          Image.asset(
            'assets/images/background_image.jpg', // Replace with your image asset
            fit: BoxFit.cover,
            width: double.infinity,
            height: double.infinity,
          ),
          // Blur Overlay
          Container(
            color: Colors.black.withOpacity(0.3), // Adjust the opacity as needed
            width: double.infinity,
            height: double.infinity,
          ),
          // Welcome Text
          Positioned(
            top: 100.0, // Adjust the top position as needed
            left: 0,
            right: 0,
            child: Center(
              child: Text(
                'Welcome to farm LINK',
                style: TextStyle(
                  color: Colors.white,
                  fontSize: 32.0,
                  fontWeight: FontWeight.bold,
                  fontFamily: 'Average Sans', // Replace with your font family
                ),
              ),
            ),
          ),
        ],
      ),
    );
  }
}
```

```
// Sign-up and Login Buttons
Positioned(
    bottom: 250.0, // Adjust the bottom position as needed
    left: 0,
    right: 0,
    child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        crossAxisAlignment: CrossAxisAlignment.center,
        children: [
            ElevatedButton(
                onPressed: () {
                    Navigator.push(
                        context,
                        MaterialPageRoute(builder: (context) => RegistrationPage()),
                    );
                },
                style: ElevatedButton.styleFrom(
                    primary: Colors.black,
                    padding: EdgeInsets.symmetric(vertical: 16.0, horizontal: 60.0),
                ),
                child: Text('Sign Up', style: TextStyle(color: Colors.white)),
            ),
            ElevatedButton(
                onPressed: () {
                    Navigator.push(
                        context,
                        MaterialPageRoute(builder: (context) => LoginPage()),
                    );
                },
                style: ElevatedButton.styleFrom(
                    primary: Colors.black,
                    padding: EdgeInsets.symmetric(vertical: 16.0, horizontal: 60.0),
                ),
                child: Text('Log In', style: TextStyle(color: Colors.white)),
            ),
        ],
    ),
);
```

SnapShot of register.dart file:

```
import 'package:flutter/material.dart';
import 'home.dart';

class RegistrationPage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Stack(
        children: [
          // Background Image
          Image.asset(
            'assets/images/background_image.jpg', // Replace with your image asset
            fit: BoxFit.cover,
            width: double.infinity,
            height: double.infinity,
          ),
          // Blur Overlay
          Container(
            color: Colors.black.withOpacity(0.3), // Adjust the opacity as needed
            width: double.infinity,
            height: double.infinity,
          ),
          // Your Registration Page Content
          Positioned(
            top: 50.0,
            left: 20.0,
            right: 20.0,
            child: Column(
              crossAxisAlignment: CrossAxisAlignment.start,
              children: [
                Text(
                  "FARMLINK",
                  style: TextStyle(
                    fontFamily: 'Poppins',
                    fontSize: 40,
                    color: Colors.white,
                  ),
                ),
                Text(
                  "Registration",
                  style: TextStyle(
                    fontFamily: 'Aoboshi One',
                    fontSize: 20,
                    color: Colors.white,
                  ),
                ),
                SizedBox(
                  height: 20,
                  width: 150,
                ),
                Padding(
                  padding: const EdgeInsets.symmetric(horizontal: 20, vertical: 10),
                  child: TextField(
                    decoration: InputDecoration(
                      prefixIcon: Icon(Icons.location_on),
                      hintText: 'Address',
                      filled: true,
                      fillColor: Colors.white.withOpacity(0.7),
                    ),
                ),
                ),
                Padding(
                  padding: const EdgeInsets.symmetric(horizontal: 20, vertical: 10),
                  child: TextField(
                    obscureText: true,
                    decoration: InputDecoration(
                      prefixIcon: Icon(Icons.lock),
                      hintText: 'Password',
                      filled: true,
                      fillColor: Colors.white.withOpacity(0.7),
                    ),
                ),
                ),
              ],
            ),
            child: ElevatedButton(
              onPressed: () {
                Navigator.push(
                  context,
                  MaterialPageRoute(builder: (context) => HomePage()),
                );
              },
              style: ElevatedButton.styleFrom(
                primary: Colors.white70.withOpacity(0.7), // Adjust opacity
                onPrimary: Colors.white, // Change the text color
                padding: EdgeInsets.symmetric(vertical: 10.0, horizontal: 32.0),
              ),
              child: Text('Register', style: TextStyle(color: Colors.black87)),
            ),
          ),
        ],
      ),
    );
  }
}
```

SnapShot of login.dart file:

```

1 import 'package:flutter/material.dart';
2 import 'home.dart';
3
4 class LoginPage extends StatelessWidget {
5   @override
6   Widget build(BuildContext context) {
7     return Scaffold(
8       body: Stack(
9         children: [
10           // Background Image
11           Image.asset(
12             'assets/images/background_image.jpg', // Replace
13             fit: BoxFit.cover,
14             width: double.infinity,
15             height: double.infinity,
16           ), // Image.asset
17           // Blur Overlay
18           Container(
19             color: Colors.black.withOpacity(0.3), // Adjust t
20             width: double.infinity,
21             height: double.infinity,
22           ), // Container
23           // Your Login Page Content
24           Positioned(
25             top: 50.0,
26             left: 20.0,
27             right: 20.0,
28             child: Column(
29               crossAxisAlignment: CrossAxisAlignment.start,
30               children: [
31                 Text(
32                   "FARMLINK",
33                   style: TextStyle(
34                     fontFamily: 'Poppins',
35                   ),
36               ],
37             ),
38           ),
39         ],
40       ),
41     );
42   }
43 }

```

```

1 import 'package:flutter/material.dart';
2 import 'home.dart';
3
4 class LoginPage extends StatelessWidget {
5   @override
6   Widget build(BuildContext context) {
7     return Scaffold(
8       body: Stack(
9         children: [
10           // Background Image
11           Image.asset(
12             'assets/images/background_image.jpg', // Replace
13             fit: BoxFit.cover,
14             width: double.infinity,
15             height: double.infinity,
16           ), // Image.asset
17           // Blur Overlay
18           Container(
19             color: Colors.black.withOpacity(0.3), // Adjust t
20             width: double.infinity,
21             height: double.infinity,
22           ), // Container
23           // Your Login Page Content
24           Positioned(
25             top: 50.0,
26             left: 20.0,
27             right: 20.0,
28             child: Column(
29               crossAxisAlignment: CrossAxisAlignment.start,
30               children: [
31                 Text(
32                   "FARMLINK",
33                   style: TextStyle(
34                     fontFamily: 'Poppins',
35                   ),
36               ],
37             ),
38           ),
39         ],
40       ),
41     );
42   }
43 }

```

```

66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99

```

```

padding: const EdgeInsets.symmetric(horizontal: 20, vertical: 10),
child: TextField(
  obscureText: true,
decoration: InputDecoration(
  prefixIcon: Icon(Icons.lock),
  hintText: 'Password',
  filled: true,
  fillColor: Colors.white.withOpacity(0.7),
), // InputDecoration
), // Padding
SizedBox(height: 20),
Center(
  child: ElevatedButton(
    onPressed: () {
      Navigator.push(
        context,
        MaterialPageRoute(builder: (context) => HomePage()),
      );
    },
    style: ElevatedButton.styleFrom(
      primary: Colors.white70.withOpacity(0.7), // Adjust opacity
      onPrimary: Colors.white, // Change the text color
      padding: EdgeInsets.symmetric(vertical: 16.0, horizontal: 32.0),
    ),
    child: Text('Login', style: TextStyle(color: Colors.black87)),
  ), // ElevatedButton
), // Center
], // Column
), // Positioned
], // Stack
); // Scaffold

```

SnapShot of home.dart file:

```
import 'package:flutter/material.dart';
import 'pages/seeds.dart';
import 'pages/equipment.dart';
import 'pages/fertilizers.dart';
import 'pages/irrigation.dart';
void main() => runApp(MyApp());
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: HomePage(),
    );
  }
}
class HomePage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        backgroundColor: Colors.deepOrangeAccent,
        title: Text('Categories', style: TextStyle(color: Colors.white)),
        centerTitle: true,
        toolbarHeight: 80.0,
      ),
      body: GridView.count(

```

```
        crossAxisCount: 2,
        children: [
          _buildGridItem(context, 'Seeds', Icons.grass, 'assets/images/seeds.jpg', SeedListPage()),
          _buildGridItem(context, 'Equipment', Icons.build, 'assets/images/equipment.jpg', EquipmentListPage()),
          _buildGridItem(context, 'Irrigation', Icons.water, 'assets/images/irrigation.jpg', IrrigationListPage()),
          _buildGridItem(context, 'Fertilizer', Icons.emoji_nature, 'assets/images/fertilizer.jpg', FertilizerListPage())
        ],
      ),
      bottomNavigationBar: BottomNavigationBar(
        items: [
          BottomNavigationBarItem(
            icon: Icon(Icons.home, color: Colors.deepOrangeAccent),
            label: 'Home',
          ),
          BottomNavigationBarItem(
            icon: Icon(Icons.notifications, color: Colors.deepOrangeAccent),
            label: 'Alerts',
          ),
          BottomNavigationBarItem(
            icon: Icon(Icons.person, color: Colors.deepOrangeAccent),
            label: 'Profile',
          ),
        ],
      ),

```

```
        BottomNavigationBarItem(
          icon: Icon(Icons.view_list, color: Colors.deepOrangeAccent),
          label: 'Tasks',
        ),
      ],
    );
}
Widget _buildGridItem(BuildContext context, String title, IconData icon, String backgroundImage, Widget page) {
  return GestureDetector(
    onTap: () {
      // Navigate to the corresponding page when the grid item is tapped
      Navigator.push(context, MaterialPageRoute(builder: (context) => page));
    },
    child: Card(
      elevation: 2.0,
      margin: EdgeInsets.all(8.0),
      shape: RoundedRectangleBorder(
        borderRadius: BorderRadius.circular(16.0),
      ),
      child: ClipRRect(
        borderRadius: BorderRadius.circular(16.0),
        child: Stack(
          fit: StackFit.expand,
          children: [

```

```
    Image.asset(
      backgroundImage,
      fit: BoxFit.cover,
    ),
    Positioned.fill(
      child: Container(
        color: Colors.black12.withOpacity(0.7),
      ),
    ),
    Positioned(
      bottom: 16.0,
      left: 16.0,
      right: 16.0,
      child: Column(
        mainAxisAlignment: MainAxisAlignment.end,
        crossAxisAlignment: CrossAxisAlignment.end,
        children: [
          Icon(
            icon,
            size: 50.0,
            color: Colors.blue,
          ),
          SizedBox(height: 8.0),
          Text(
            title,
            style: TextStyle(
              color: Colors.white,
              fontSize: 16.0,
            ),
          ),
        ],
      ),
    ),
  ],
)
```

SnapShot of seeds.dart file:

```
import 'package:flutter/material.dart';
class Seed {
  final String name;
  final double price;
  final String imageUrl;
  Seed({required this.name, required this.price, required this.imageUrl});
}
class SeedListPage extends StatefulWidget {
  @override
  _SeedListPageState createState() => _SeedListPageState();
}
class _SeedListPageState extends State<SeedListPage> {
  List<Seed> seeds = [
    Seed(name: 'Tomato Seeds', price: 150, imageUrl: 'assets/images/tomato-seed.jpg'),
    Seed(name: 'Carrot Seeds', price: 300, imageUrl: 'assets/images/carrot-seeds.jpg'),
    Seed(name: 'Lettuce Seeds', price: 100, imageUrl: 'assets/images/lettuce-seeds.jpg')
  ];
  Map<Seed, int> cart = {};
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        backgroundColor: Colors.deepOrangeAccent,
        title: Text('Seeds', style: TextStyle(color: Colors.white)),
        centerTitle: true,
        toolbarHeight: 80.0,
      ),
      body: ListView.builder(
        itemCount: seeds.length,
        itemBuilder: (context, index) {
          final seed = seeds[index];
          final itemCount = cart.containsKey(seed) ? cart[seed] : 0;
          return Card(
            elevation: 3,
            margin: EdgeInsets.symmetric(vertical: 8, horizontal: 16),
            child: ListTile(
              contentPadding: EdgeInsets.all(16),
              leading: Image.asset(seed.imageUrl),
              title: Column(
                crossAxisAlignment: CrossAxisAlignment.start,
                children: [
                  Text(
                    seed.name,
                    style: TextStyle(fontSize: 18, fontWeight: FontWeight.bold),
                  ),
                  SizedBox(height: 8),
                  Text(
                    '\$${seed.price.toStringAsFixed(2)}',
                    style: TextStyle(fontSize: 16),
                  ),
                ],
              ),
            ),
            child: Text(
              'Cart Item Count: $itemCount',
              style: TextStyle(fontSize: 16),
            ),
          );
        }
      ),
    );
  }
}
```

```
        ),
        trailing: Row(
            mainAxisAlignment: MainAxisAlignment.end,
            children: [
                IconButton(
                    icon: Icon(Icons.remove),
                    onPressed: () {
                        removeFromCart(seed);
                    },
                ),
                Text(itemCount.toString()),
                IconButton(
                    icon: Icon(Icons.add),
                    onPressed: () {
                        addToCart(seed);
                    },
                ),
            ],
        ),
    );
}
);
```

```
bottomNavigationBar: BottomNavigationBar(
  items: [
    BottomNavigationBarItem(
      icon: Icon(Icons.home, color: Colors.deepOrangeAccent),
      label: 'Home',
    ),
    BottomNavigationBarItem(
      icon: Icon(Icons.notifications, color: Colors.deepOrangeAccent),
      label: 'Alerts',
    ),
    BottomNavigationBarItem(
      icon: Icon(Icons.person, color: Colors.deepOrangeAccent),
      label: 'Profile',
    ),
    BottomNavigationBarItem(
      icon: Icon(Icons.view_list, color: Colors.deepOrangeAccent),
      label: 'Tasks',
    ),
  ],
);
```

```
void addToCart(Seed seed) {
    setState(() {
        if (cart.containsKey(seed)) {
            cart[seed] = cart[seed]! + 1;
        } else {
            cart[seed] = 1;
        }
    });
}

void removeFromCart(Seed seed) {
    setState(() {
        if (cart.containsKey(seed)) {
            if (cart[seed]! > 1) {
                cart[seed] = cart[seed]! - 1;
            } else {
                cart.remove(seed);
            }
        }
    });
}
```

SnapShot of equipment.dart file:

```
import 'package:flutter/material.dart';
class Equipment {
  final String name;
  final double price;
  final String imageUrl;
  Equipment({required this.name, required this.price, required this.imageUrl});
}
class EquipmentListPage extends StatefulWidget {
  @override
  _EquipmentListPageState createState() => _EquipmentListPageState();
}
class _EquipmentListPageState extends State<EquipmentListPage> {
  List<Equipment> equipments = [
    Equipment(name: 'Disc Harrow', price: 15000, imageUrl: 'assets/images/Disc-Harrow.jpg'),
    Equipment(name: 'Tractors', price: 300000, imageUrl: 'assets/images/trucktor.jpg'),
    Equipment(name: 'Combine', price: 1000000, imageUrl: 'assets/images/combine.jpg'),
  ];
  Map<Equipment, int> cart = {};
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        backgroundColor: Colors.deepOrangeAccent,
        title: Text('Equipments', style: TextStyle(color: Colors.white)),
        centerTitle: true,
      ),

```

```
      toolbarHeight: 80.0,
    ),
    body: ListView.builder(
      itemCount: equipments.length,
      itemBuilder: (context, index) {
        final seed = equipments[index];
        final itemCount = cart.containsKey(seed) ? cart[seed] : 0;
        return Card(
          elevation: 3,
          margin: EdgeInsets.symmetric(vertical: 8, horizontal: 16),
          child: ListTile(
            contentPadding: EdgeInsets.all(16),
            leading: Image.asset(seed.imageUrl),
            title: Column(
              crossAxisAlignment: CrossAxisAlignment.start,
              children: [
                Text(
                  seed.name,
                  style: TextStyle(fontSize: 18, fontWeight: FontWeight.bold),
                ),
                SizedBox(height: 8),
                Text(
                  '\$${seed.price.toStringAsFixed(2)}',
                  style: TextStyle(fontSize: 16),
                ),
              ],

```

```
        ],
    ),
    trailing: Row(
        mainAxisAlignment: MainAxisAlignment.end,
        children: [
            IconButton(
                icon: Icon(Icons.remove),
                onPressed: () {
                    removeFromCart(seed);
                },
            ),
            Text(itemCount.toString()),
            IconButton(
                icon: Icon(Icons.add),
                onPressed: () {
                    addToCart(seed);
                },
            ),
        ],
    ),
),
);
});
```

```
bottomNavigationBar: BottomNavigationBar(
  items: [
    BottomNavigationBarItem(
      icon: Icon(Icons.home, color: Colors.deepOrangeAccent),
      label: 'Home',
    ),
    BottomNavigationBarItem(
      icon: Icon(Icons.notifications, color: Colors.deepOrangeAccent),
      label: 'Alerts',
    ),
    BottomNavigationBarItem(
      icon: Icon(Icons.person, color: Colors.deepOrangeAccent),
      label: 'Profile',
    ),
    BottomNavigationBarItem(
      icon: Icon(Icons.view_list, color: Colors.deepOrangeAccent),
      label: 'Tasks',
    ),
  ],
),
);
```

```
void addToCart(Equipment equip) {
    setState(() {
        if (cart.containsKey(equip)) {
            cart[equip] = cart[equip]! + 1;
        } else {
            cart[equip] = 1;
        }
    });
}

void removeFromCart(Equipment equip) {
    setState(() {
        if (cart.containsKey(equip)) {
            if (cart[equip]! > 1) {
                cart[equip] = cart[equip]! - 1;
            } else {
                cart.remove(equip);
            }
        }
    });
}
```

SnapShot of irrigation.dart file:

```
import 'package:flutter/material.dart';
class IrrigationSystem {
  final String name;
  final double price;
  final String imageUrl;
  IrrigationSystem({required this.name, required this.price, required this.imageUrl});
}
class IrrigationListPage extends StatefulWidget {
  @override
  _IrrigationListPageState createState() => _IrrigationListPageState();
}
class _IrrigationListPageState extends State<IrrigationListPage> {
  List<IrrigationSystem> irrigationSystems = [
    IrrigationSystem(name: 'Drip Irrigation', price: 2000, imageUrl: 'assets/images/drip_irrigation.jpg'),
    IrrigationSystem(name: 'Sprinkler System', price: 1500, imageUrl: 'assets/images/sprinkler_system.jpg'),
    IrrigationSystem(name: 'Flood Irrigation', price: 1000, imageUrl: 'assets/images/flood_irrigation.jpeg'),
  ];
  Map<IrrigationSystem, int> cart = {};
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        backgroundColor: Colors.blue,
        title: Text('Irrigation Systems', style: TextStyle(color: Colors.white)),
        centerTitle: true,
      ),
      body: ListView.builder(
        itemCount: irrigationSystems.length,
        itemBuilder: (context, index) {
          final irrigationSystem = irrigationSystems[index];
          final itemCount = cart.containsKey(irrigationSystem) ? cart[irrigationSystem] : 0;
          return Card(
            elevation: 3,
            margin: EdgeInsets.symmetric(vertical: 8, horizontal: 16),
            child: ListTile(
              contentPadding: EdgeInsets.all(16),
              leading: Image.asset(irrigationSystem.imageUrl),
              title: Column(
                crossAxisAlignment: CrossAxisAlignment.start,
                children: [
                  Text(
                    irrigationSystem.name,
                    style: TextStyle(fontSize: 18, fontWeight: FontWeight.bold),
                  ),
                  SizedBox(height: 8),
                  Text(
                    '\t${irrigationSystem.price.toStringAsFixed(2)}',
                    style: TextStyle(fontSize: 16),
                  ),
                ],
              ),
            ),
          );
        },
      ),
    );
  }
}
```

```
        ],
    ),
    trailing: Row(
        mainAxisAlignment: MainAxisAlignment.end,
        children: [
            IconButton(
                icon: Icon(Icons.remove),
                onPressed: () {
                    removeFromCart(irrigationSystem);
                },
            ),
            Text(itemCount.toString()),
            IconButton(
                icon: Icon(Icons.add),
                onPressed: () {
                    addToCart(irrigationSystem);
                },
            ),
        ],
    ),
);
});
```

```
bottomNavigationBar: BottomNavigationBar(
    items: [
        BottomNavigationBarItem(
            icon: Icon(Icons.home, color: Colors.blue),
            label: 'Home',
        ),
        BottomNavigationBarItem(
            icon: Icon(Icons.notifications, color: Colors.blue),
            label: 'Alerts',
        ),
        BottomNavigationBarItem(
            icon: Icon(Icons.person, color: Colors.blue),
            label: 'Profile',
        ),
        BottomNavigationBarItem(
            icon: Icon(Icons.view_list, color: Colors.blue),
            label: 'Tasks',
        ),
    ],
),
);
```

```
void addToCart(IrrigationSystem irrigationSystem) {
    setState(() {
        if (cart.containsKey(irrigationSystem)) {
            cart[irrigationSystem] = cart[irrigationSystem]! + 1;
        } else {
            cart[irrigationSystem] = 1;
        }
    });
}

void removeFromCart(IrrigationSystem irrigationSystem) {
    setState(() {
        if (cart.containsKey(irrigationSystem)) {
            if (cart[irrigationSystem]! > 1) {
                cart[irrigationSystem] = cart[irrigationSystem]! - 1;
            } else {
                cart.remove(irrigationSystem);
            }
        }
    });
}
```

SnapShot of fertilizer.dart file:

```
import 'package:flutter/material.dart';
class Fertilizer {
  final String name;
  final double price;
  final String imageUrl;
  Fertilizer({required this.name, required this.price, required this.imageUrl});
}
class FertilizerListPage extends StatefulWidget {
  @override
  _FertilizerListPageState createState() => _FertilizerListPageState();
}
class _FertilizerListPageState extends State<FertilizerListPage> {
  List<Fertilizer> fertilizers = [
    Fertilizer(name: 'Nitrogen Fertilizer', price: 500, imageUrl: 'assets/images/nitrogen.jpg'),
    Fertilizer(name: 'Phosphorus Fertilizer', price: 400, imageUrl: 'assets/images/phosphet.png'),
    Fertilizer(name: 'Potassium Fertilizer', price: 350, imageUrl: 'assets/images/potassium.jpg'),
  ];
  Map<Fertilizer, int> cart = {};
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        backgroundColor: Colors.green,
        title: Text('Fertilizers', style: TextStyle(color: Colors.white)),
        centerTitle: true,
```

```
        toolbarHeight: 80.0,
),
body: ListView.builder(
  itemCount: fertilizers.length,
  itemBuilder: (context, index) {
    final fertilizer = fertilizers[index];
    final itemCount = cart.containsKey(fertilizer) ? cart[fertilizer] : 0;
    return Card(
      elevation: 3,
      margin: EdgeInsets.symmetric(vertical: 8, horizontal: 16),
      child: ListTile(
        contentPadding: EdgeInsets.all(16),
        leading: Image.asset(fertilizer.imageUrl),
        title: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            Text(
              fertilizer.name,
              style: TextStyle(fontSize: 18, fontWeight: FontWeight.bold),
            ),
            SizedBox(height: 8),
            Text(
              '\$${fertilizer.price.toStringAsFixed(2)}',
              style: TextStyle(fontSize: 16),
            ),
          ],

```

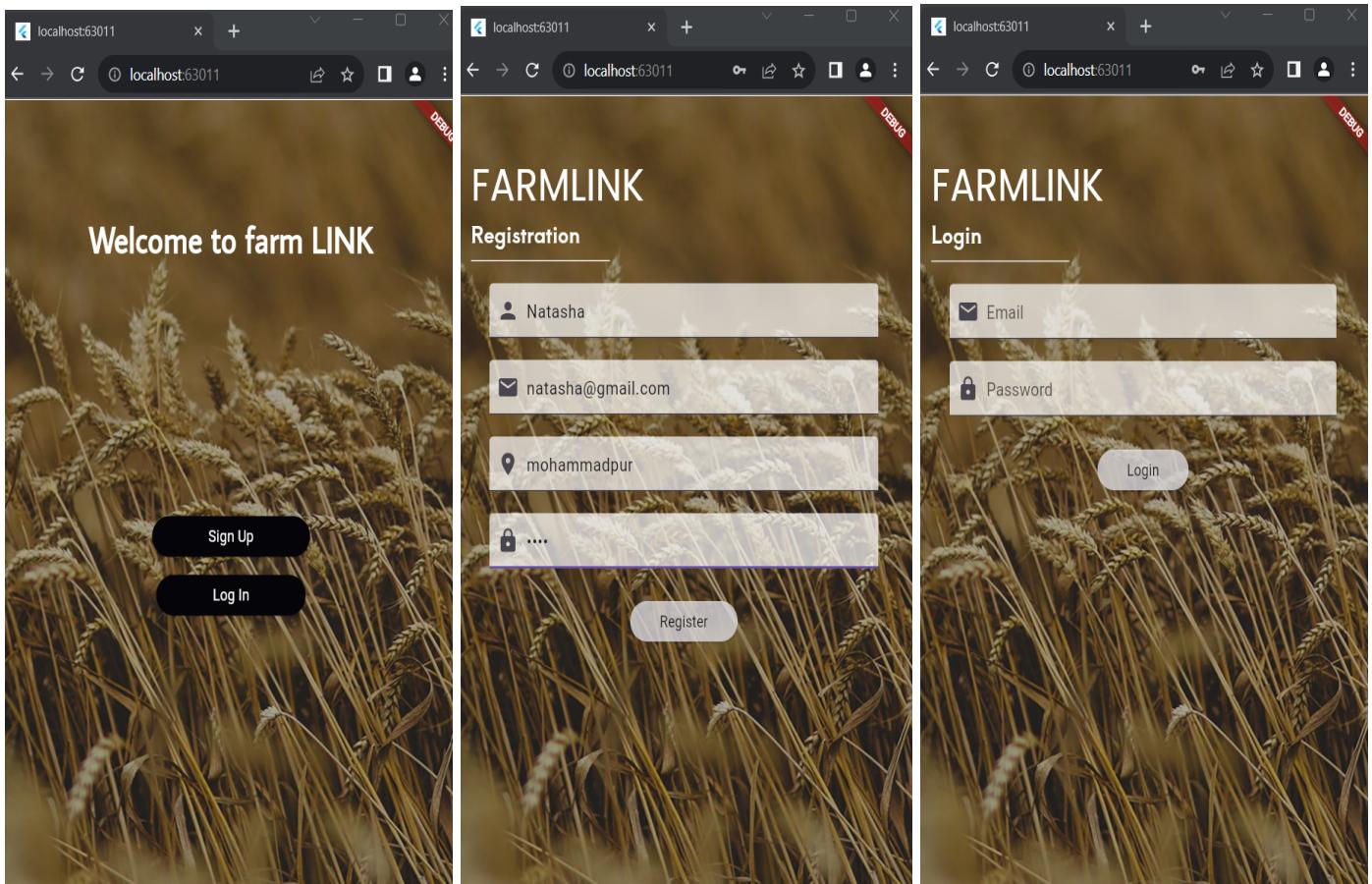
```
        ],
      ),
      trailing: Row(
        mainAxisAlignment: MainAxisAlignment.end,
        children: [
          IconButton(
            icon: Icon(Icons.remove),
            onPressed: () {
              removeFromCart(fertilizer);
            },
          ),
          Text(itemCount.toString()),
          IconButton(
            icon: Icon(Icons.add),
            onPressed: () {
              addToCart(fertilizer);
            },
          ),
        ],
      ),
    ),
  );
},
);
```

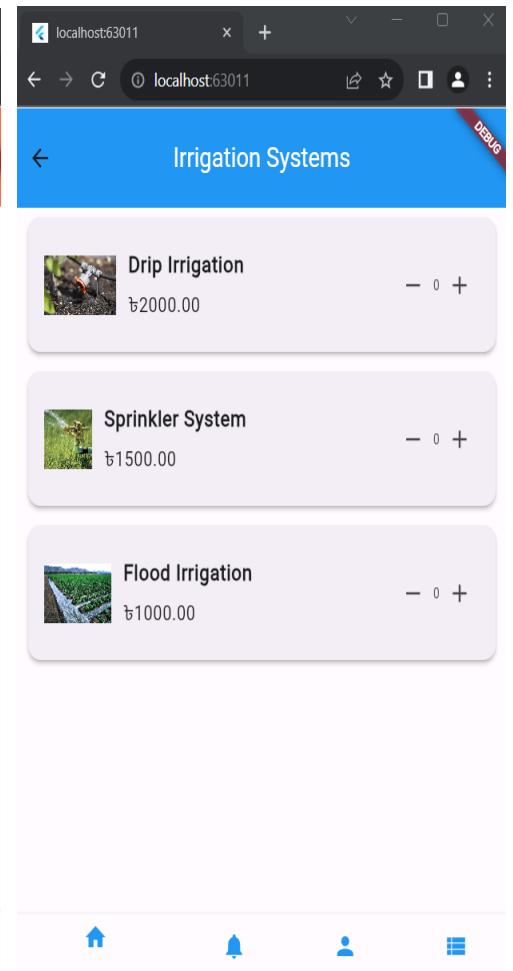
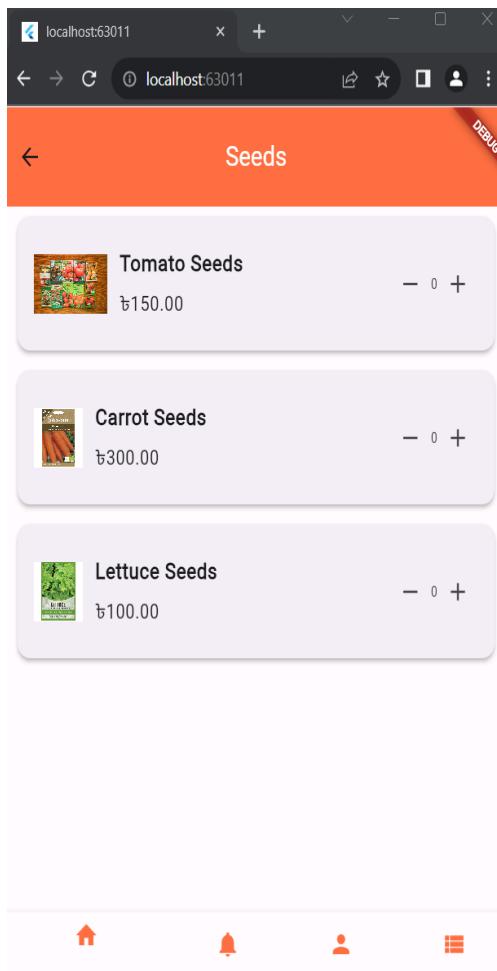
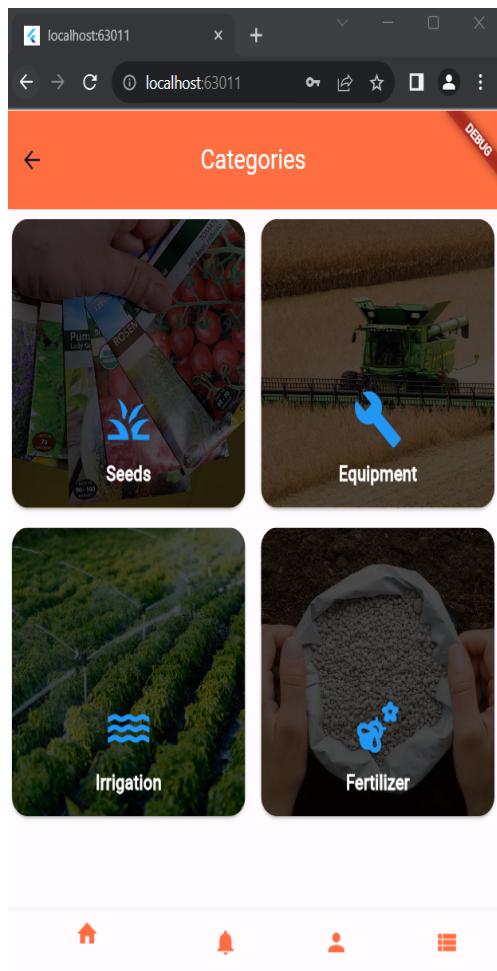
```
bottomNavigationBar: BottomNavigationBar(
  items: [
    BottomNavigationBarItem(
      icon: Icon(Icons.home, color: Colors.green),
      label: 'Home',
    ),
    BottomNavigationBarItem(
      icon: Icon(Icons.notifications, color: Colors.green),
      label: 'Alerts',
    ),
    BottomNavigationBarItem(
      icon: Icon(Icons.person, color: Colors.green),
      label: 'Profile',
    ),
    BottomNavigationBarItem(
      icon: Icon(Icons.view_list, color: Colors.green),
      label: 'Tasks',
    ),
  ],
),
);
```

```
void addToCart(Fertilizer fertilizer) {
  setState(() {
    if (cart.containsKey(fertilizer)) {
      cart[fertilizer] = cart[fertilizer]! + 1;
    } else {
      cart[fertilizer] = 1;
    }
  });
}

void removeFromCart(Fertilizer fertilizer) {
  setState(() {
    if (cart.containsKey(fertilizer)) {
      if (cart[fertilizer]! > 1) {
        cart[fertilizer] = cart[fertilizer]! - 1;
      } else {
        cart.remove(fertilizer);
      }
    }
  });
}
```

Output:





localhost:63011

Equipments

Disc Harrow ₦15000.00 - 3 +

Tractors ₦300000.00 - 3 +

Combine ₦100000.00 - 0 +

localhost:63011

Fertilizers

Nitrogen Fertilizer ₦500.00 - 0 +

Phosphorus Fertilizer ₦400.00 - 0 +

Potassium Fertilizer ₦350.00 - 0 +

localhost:63011

Profile Page



John Doe
123 Main Street, Cityville
Farm Types:

- Vegetables
- Fruits
- Livestock

Home Notifications Profile Grid