

# **North western university**

***Department of Computer Science And Engineering (CSE -4 year)***

## **LAB REPORT**

**Course title : Software Engineering Laboratory**

**Course code : CSE-3104**

**Project on : Smart Farmer Application**

<p><b>Submitted by :</b></p> <p>Name :</p> <ul style="list-style-type: none"><li>1. Kamrul golder</li><li>2. Plabon Biswas</li><li>3. Fahim Faisal</li></ul> <p>ID :</p> <ul style="list-style-type: none"><li>1. 20211016010</li><li>2. 20211034010</li><li>3. 20211041010</li></ul> <p>Department of CSE, North Western University ,Khulna</p>	<p><b>Submitted to :</b></p> <p>Name : M. Raihan Senior Lecturer, Department of CSE, North Western University ,Khulna</p>
--	---

**Submission Date :08/04/2023**

## I. Introduction :

The term ‘Smart Farmer’ means a platform that is dedicated to integrate farmer, Merchant/Markets, government and end user and thereby bridge the gap between them. It also let everyone to be updated with the changing market scenario. Bangladeshi farmers faced many challenges and one of them is that to get a good profit for the efforts and investment that they had put in. There exist different reasons like season limitation, crop life due to which farmer get very limited amount of time to study the market conditions. farmer has a limited amount of time to find out nearest market, current stock details. Smart Farmer App can solve these daily problems of farmers. Farmers will be able to get their required fertilizer to crop seedlings and seeds through this app. And they can sell their crops at the right price. And making the right profit. One of the problems faced by farmers is determining the amount of fertilizer applied to the land. As a solution to this problem, the pesticide calculator has been kept as a small part of the app. Through which the farmer will be able to determine the amount of fertilizer according to the amount of his land.

There are many mix e commerce apps in our country. Through which we can buy our daily necessities online. This ‘Smart Farmer’ app will be able to bring this modern trend to the farmers as well.

## II. Objective of the ‘Smart Farmer’ app:

1. Farmers can easily purchase fertilizers and pesticides for their crops at the right price.
2. Farmers will know about the current market place.
3. Farmers can easily purchase medicine for their animals and birds at the right price.
4. With the help of pesticide calculator, they can determine the amount of fertilizer according to their land.
5. They can buy the seedlings or seeds they need for cultivation at the right price.
6. And they can also buy agricultural machinery through this ‘Smart Farmer’ app.

### III. Activity Diagram :

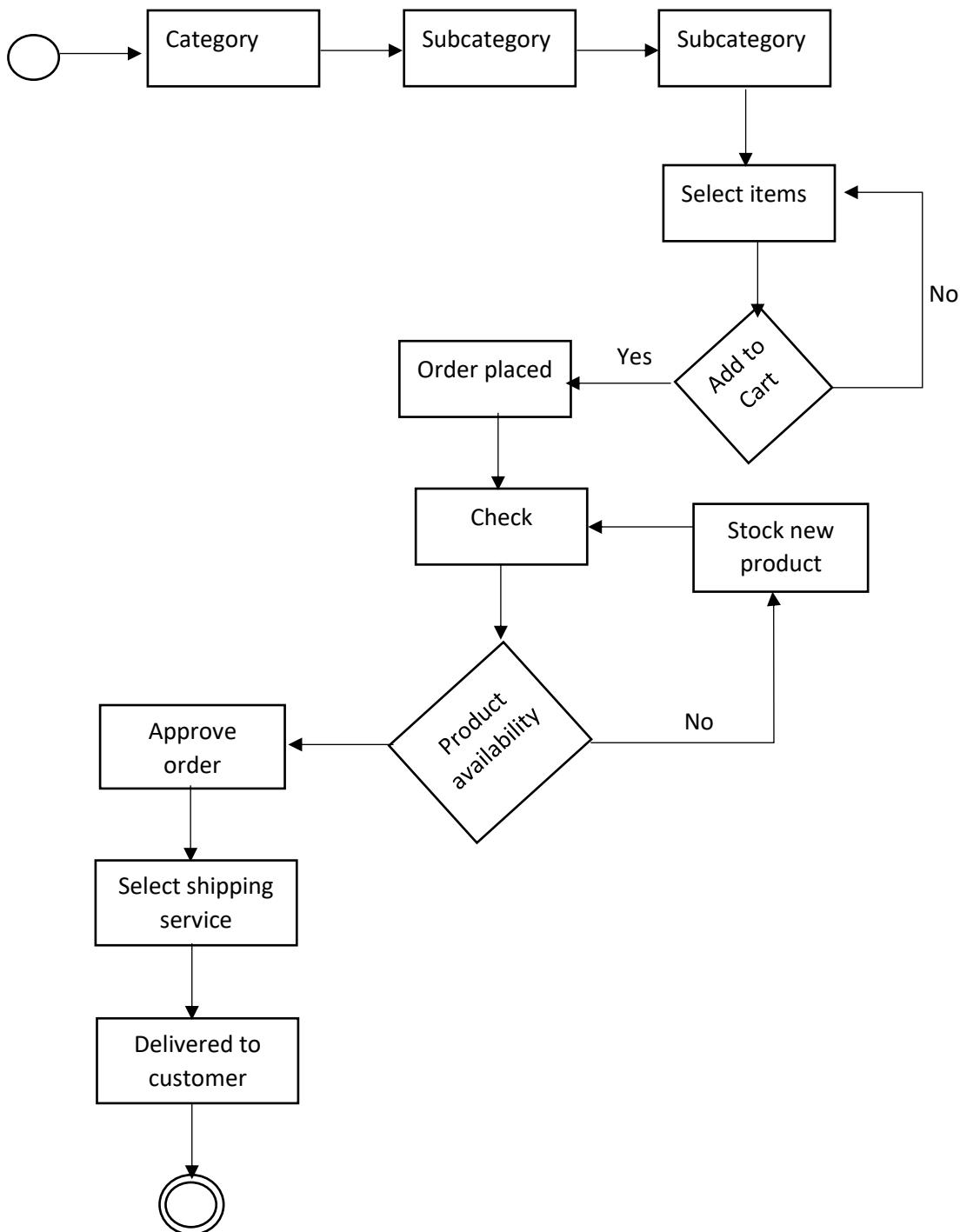


Fig:1.1 activity diagram

To order a product, first select the category. Then we have to select the desired product by selecting the sub category and add it to the cart. If you want to order the product, you have to complete the order process. Once the order process is complete, if there is sufficient quantity on the seller's site, the order will be approved and the shipping method will be selected and delivered to the customer.

#### **IV. Features of ‘Smart Farmer’ app :**

- 1. Splash Screen :** When we open the Smart Farmer app, we will see a splash screen of 2 seconds. Where our app name and logo appear.



Fig:1.2 Splash screen

- 2. Log in and Sign up screen :** At the end of the 2-second splash screen, the login and sign-up screen will appear in front of us. To access the app we must have an account with which account to log in. When logged in, the home page will appear in front of us. If there is no account, the user can sign up with the help of email and password.

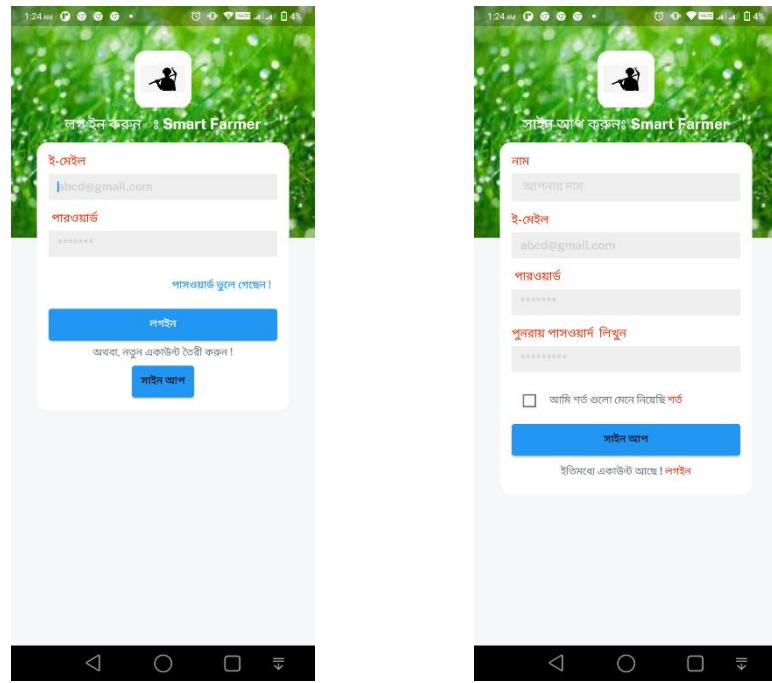


Fig:1.3 Log in and Sign up screen

**3. Home Screen:** At the top of the home screen is the search bar. Through which the user can search and find any product. We can get an idea of what kind of services we can get through this app from the home page. Here is the pesticide meter, popular product group and all product list. User can search from home screen, buy products from popular products list or all products list.

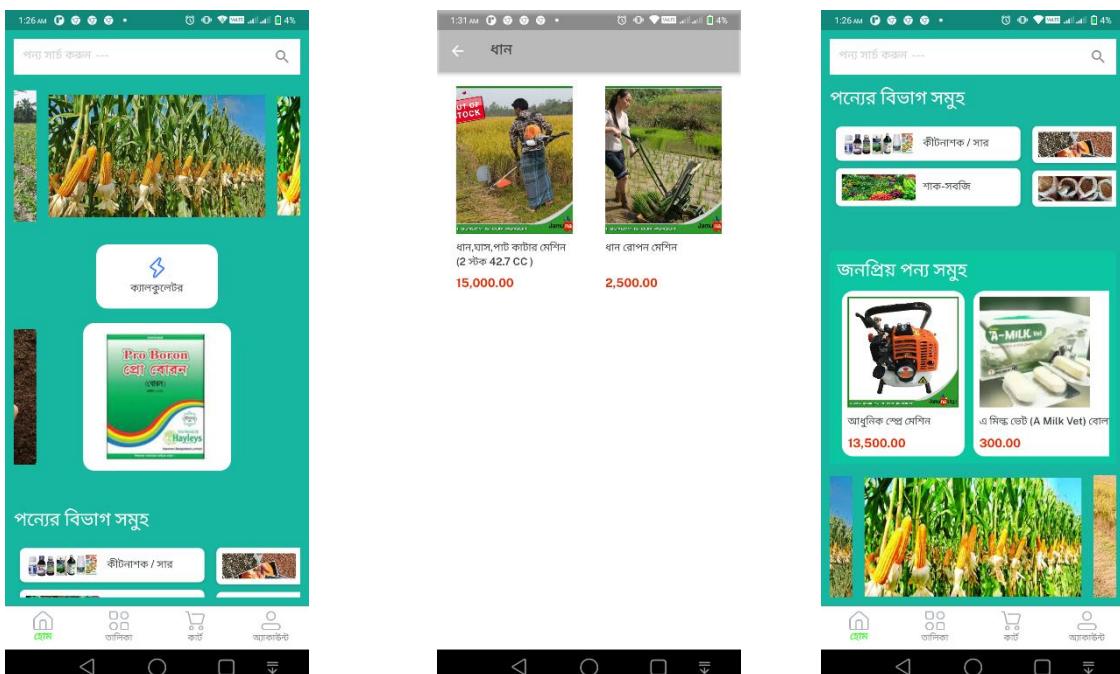


Fig:1.4 Home Screen



Fig:1.4 Home Screen

**4.Pesticide Calculator :** The pesticide calculator is mainly designed for paddy cultivation. Using this calculator, the amount of fertilizer can be determined for paddy cultivation.



Fig:1.5 Pesticide Calculator

**5. Category Screen :** From the category, the user can select the category according to his needs and purchase the product.



Fig:1.6 Category Screen

**6. Sub Category:** The category is divided into some sub categories. With the help of this sub-category, the user can easily select products.

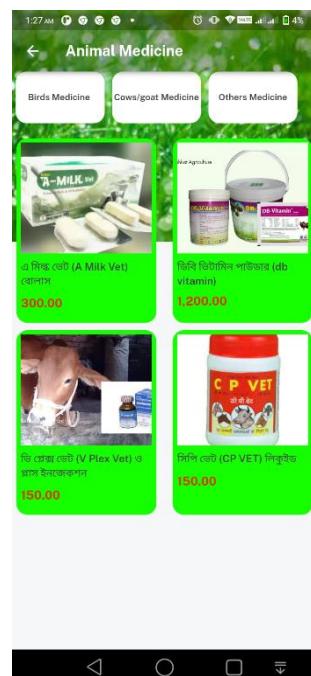


Fig:1.7 Category Details

**7. Items Details:** items Details stores all the information about the product. Protect Price, and information about Protect can be found here. The option to add products to the list of preferred products is available on this screen. The product can be added to the list of favorites by pressing the favorite button. In case of product order, quantity should be selected and added to the cart.



Fig:1.8 Items details

**8.Chat :** For any questions or to know anything about the product. Chat option can be used from item details screen.

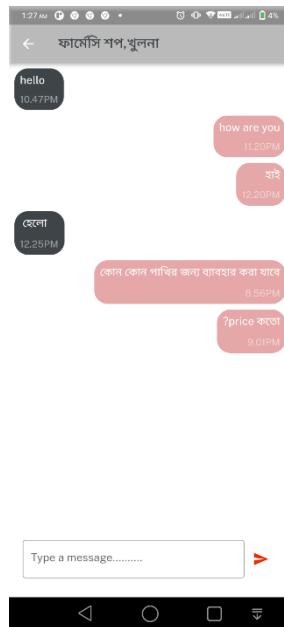


Fig:1.9 Chat Screen

**9. Cart Screen :** In the beginning, there is the cart screen. If the product is selected from the item details, the product will be added to the cart screen. There is a delete option to delete the product from the cart. All order processes are managed from this channel. If user press the order button from the cart screen, a form will appear. By filling this form and selecting the payment method, the user can complete the order process.

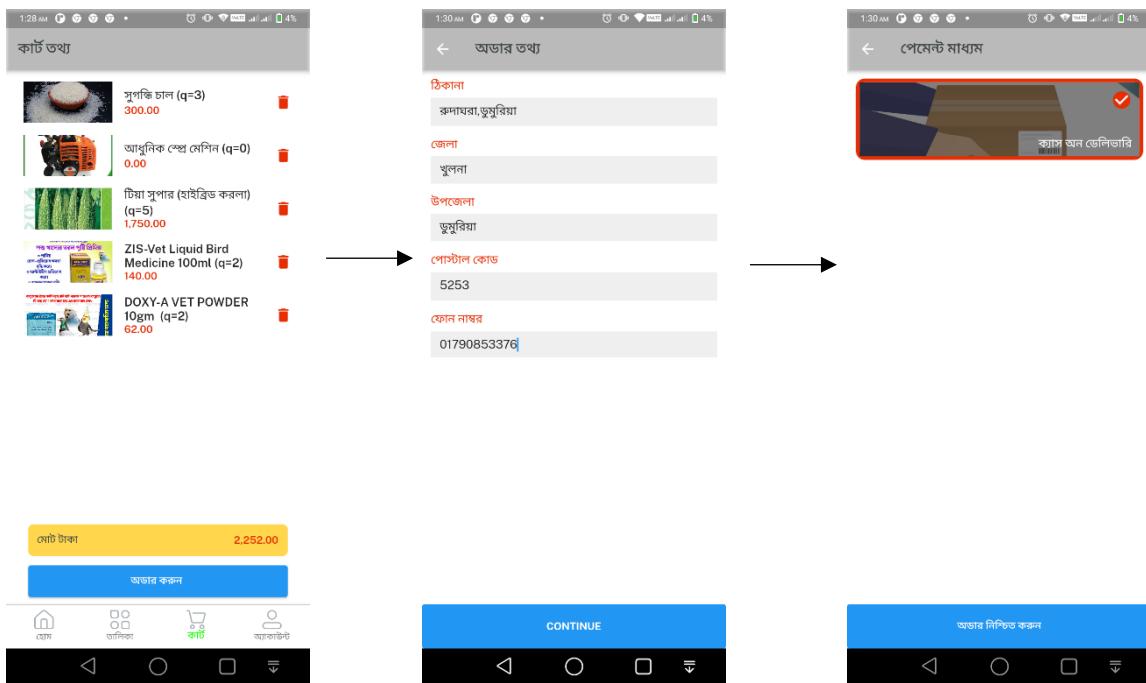


Fig:1.10 Order process

**10. Profile Screen :** The profile screen stores the user's details. This screen has log out button for logging out and user can change his picture and password with the help of edit button. Moreover, he can see the current status of his ordered products and the list of his favorite products.

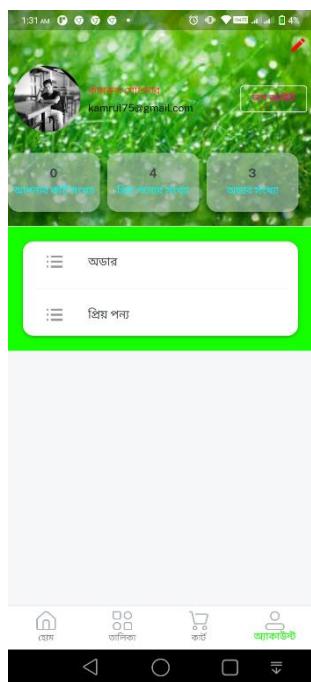


Fig:1.11 Profile Screen

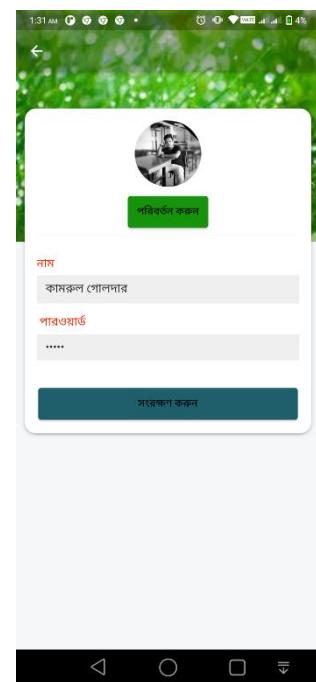


Fig:1.12 Edit profile Screen

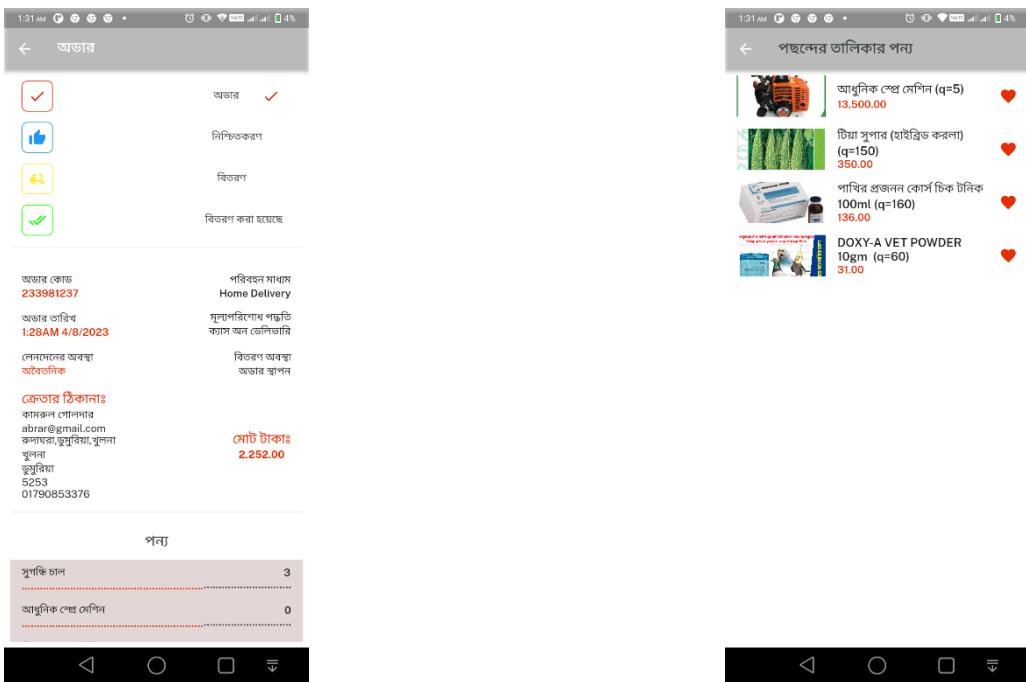


Fig:1.12 order details

Fig:1.13 Favorite list

## V. Data Follow Diagram :

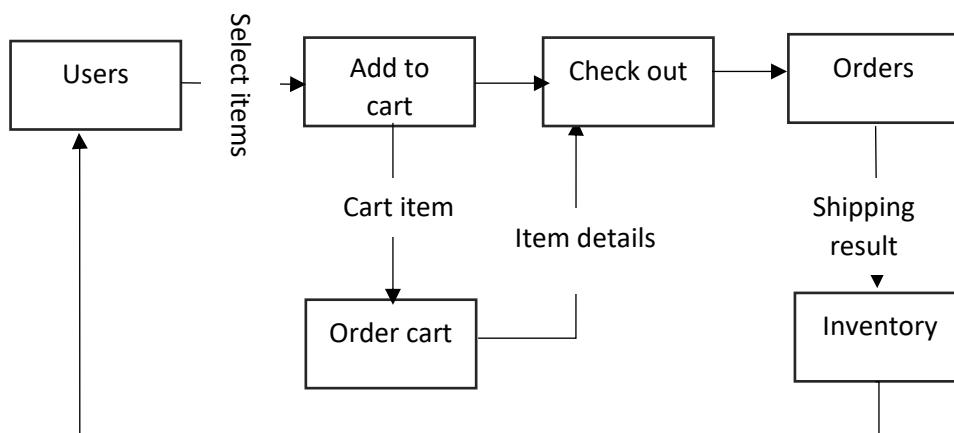
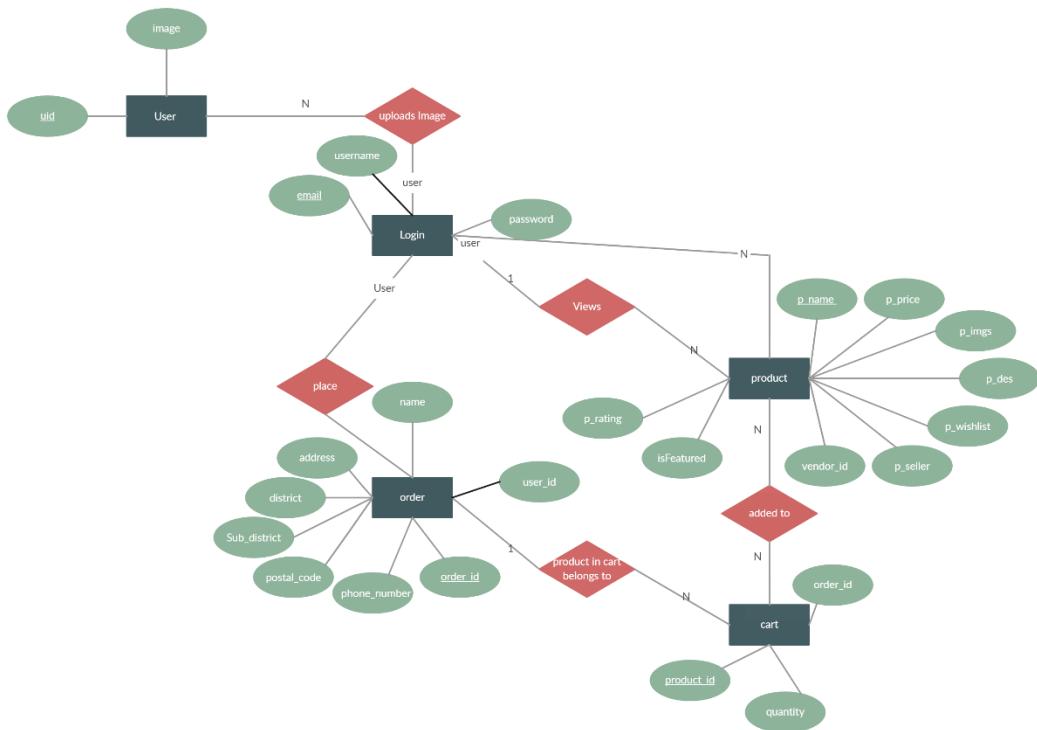


Fig:1.14 data follow diagram

## **VI. ER Diagram**



## **VII. Target Population :**

This app is mainly made targeting farmers. Although there are many favorite e-commerce apps in our country. There is no such platform for farmers. For which ‘Smart Farmer’ app is made for farmers.

## **VIII. Social Benefit :**

Through this app, farmers can buy fertilizers at the right price. As a result production difficulties will be reduced. Moreover, farmers can sell their crops at the right price. As a result, farmers will benefit. The product goes through many stages for sale and finally the product reaches the consumer. Selling products online will not require these steps. So the Product prices will be under control.

## **IX. Conclusion :**

With the help of this system, farmer can optimize his crop profit and launch complaint. Government can set minimum price for minimum quality of crop/product. Government can set and update rules and regulations time to time for better and effective management. Government can deal with different problem with the help of market details and load balancing technique.

## **Reference :**

- [1] Flutter® A HANDS ON GUIDE TO APP DEVELOPMENT, Marco L. Napoli,
- [2] Link : [Flutter Tutorial \(tutorialspoint.com\)](https://www.tutorialspoint.com/flutter/index.htm)
- [3]Link : [Learn \(flutter.dev\)](https://flutter.dev/)
- [4] Link : [Flutter UI Tutorial - Email Client Gmail Clone App UI Design \(Figma to Flutter\) | PART 1 - YouTube](https://www.youtube.com/watch?v=KJLjyfzXWUw&list=PLHqOOGdVQDgkPmIuGZBzvC9MzF0oRzrA&t=10s)

## **Individual GitHub link and screenshot :**

- 1.Kamrul Golder :** [kamrul1027 · GitHub](https://github.com/kamrul1027)

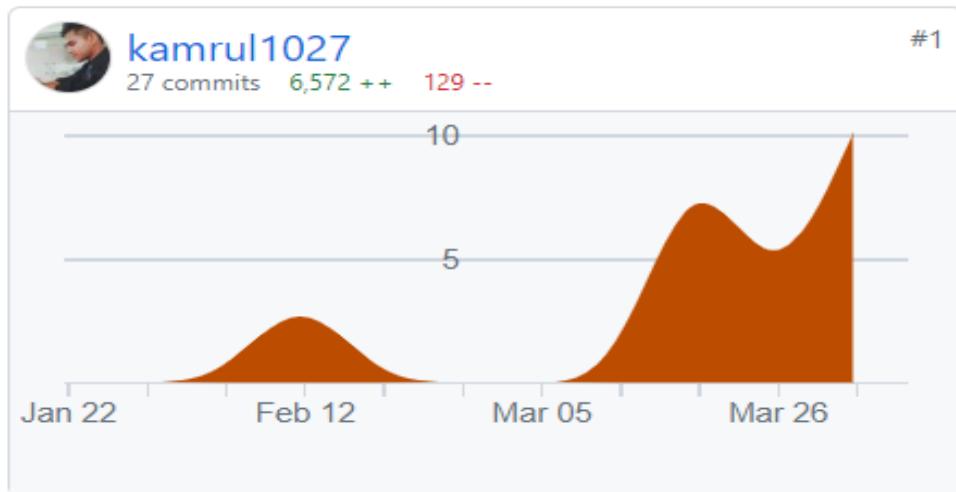


Fig:1.15 Github account link (Kamrul Golder )

## 2.Plabon Biswash : [plabon360 \(Plabon Biswas Shovon\) · GitHub](#)



Fig:1.16 Github account link (Plabon Biswas)

### 3.Fahim Faisal [fahimfaisaltarun \(Fahim Faisal Tarun\) · GitHub](#)

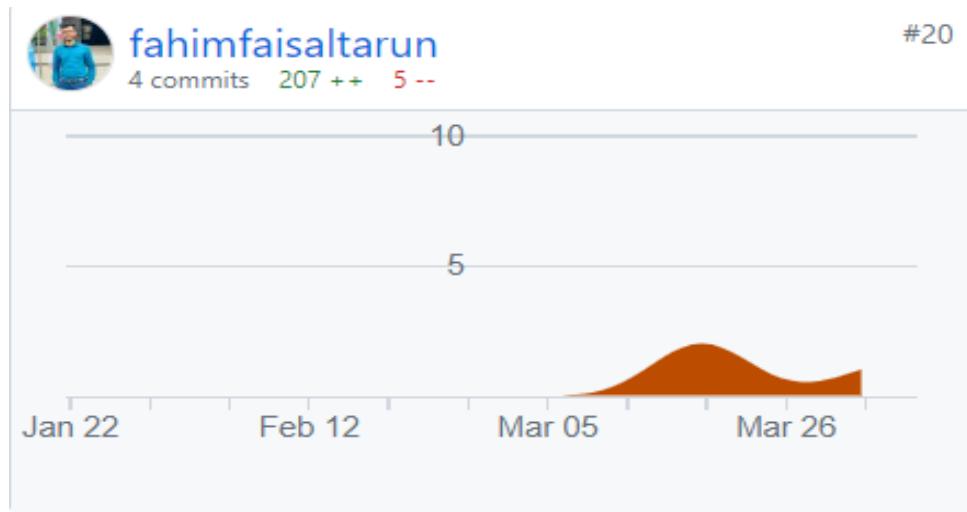


Fig:1.17 Github account link (Fahim Faisal )

### Task Scheduling App :

Board link : [Smart Farmer | Trello](#)

<https://trello.com/invite/b/u6UkzNUr/ATTIbed6d420e5877e76f1beb44eacfcea5e31807CB5/smart-farmer>

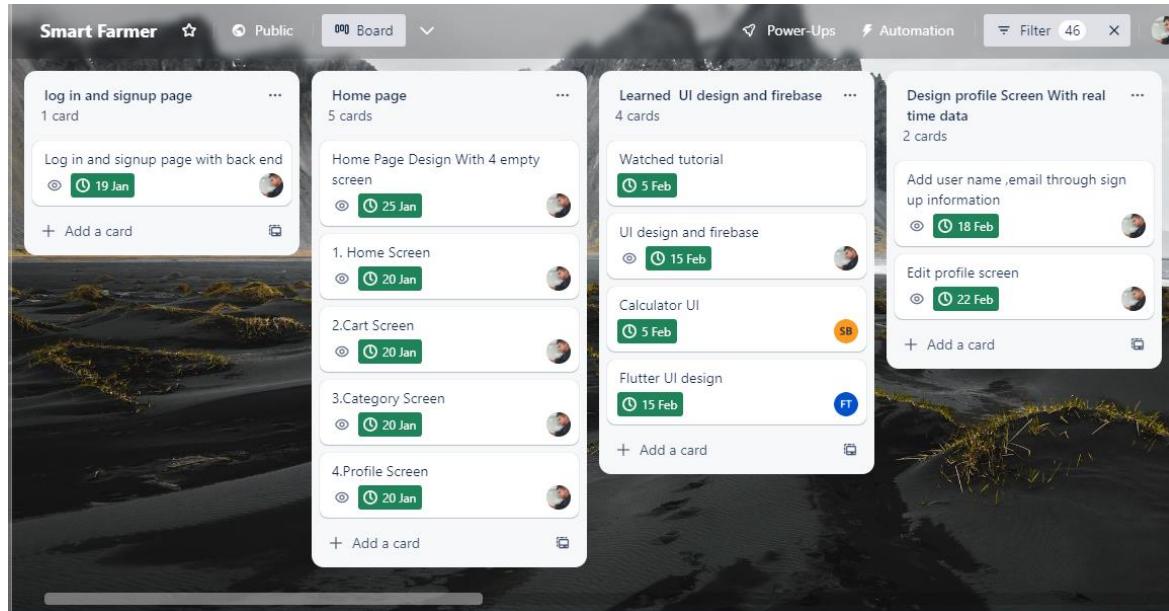


Fig:1.18 Task Scheduling app Screenshot

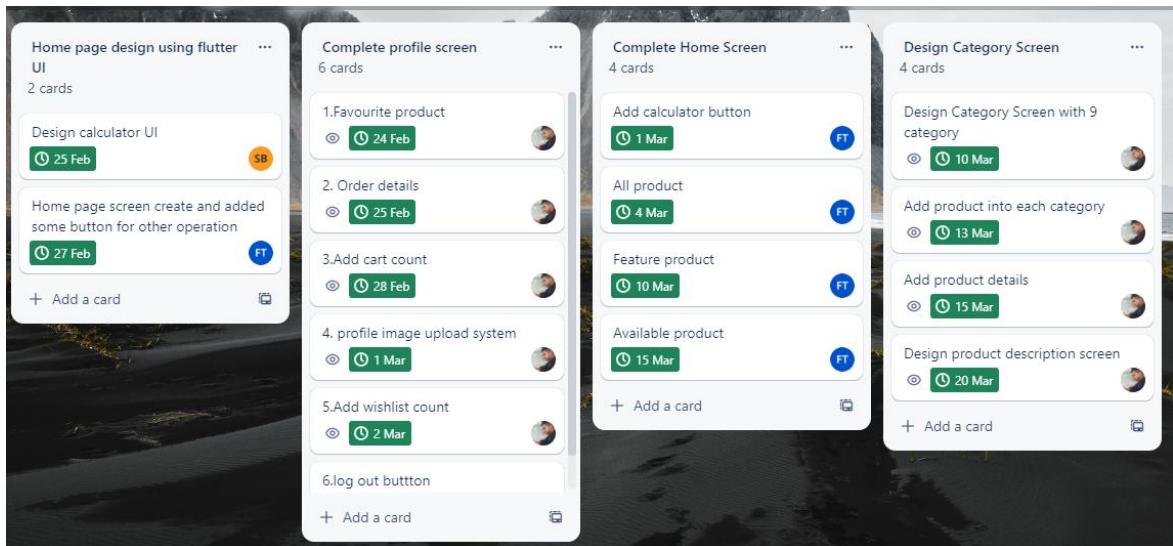


Fig:1.19 Task Scheduling app Screenshot

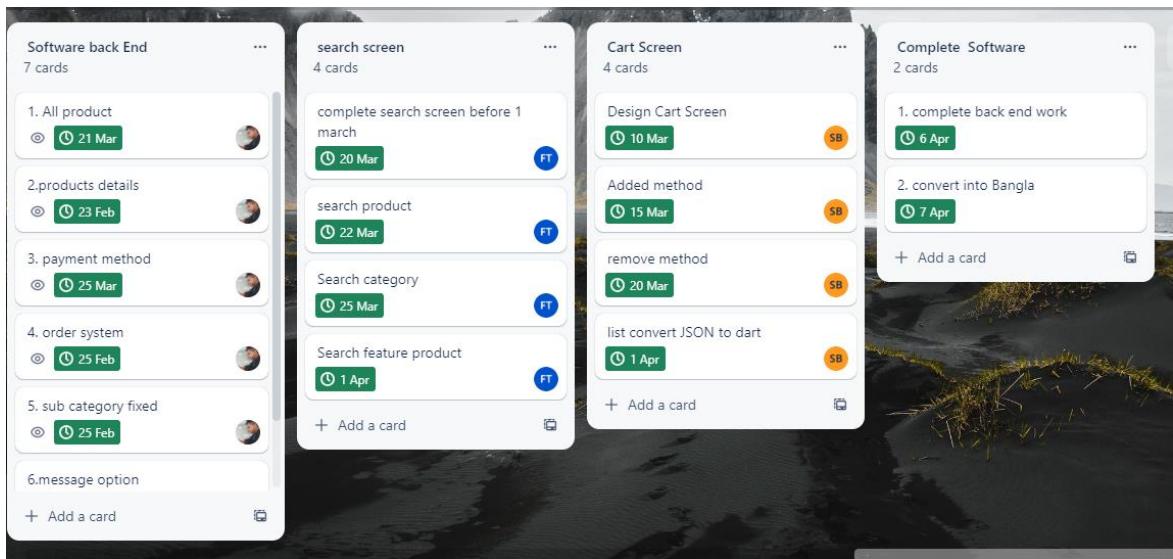


Fig:1.20 Task Scheduling app Screenshot

**Source Code**

DOCUMENT \_HEADING

## 2 files

(file list disabled)

### lib\Account\account.dart

```
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:farmer/Account/components/details_cart.dart';
import 'package:farmer/consts/consts.dart';
import 'package:farmer/consts/list.dart';
import 'package:farmer/controller/auth_controller.dart';
import 'package:farmer/controller/profile_controller.dart';
import 'package:farmer/services/firestore_services.dart';
import 'package:farmer/views/chat_screen/messeging_screen.dart';
import 'package:farmer/views/edit_profile_screen.dart';
import 'package:farmer/views/scren/Wishlistscreen/wishlist_screen.dart';
import 'package:farmer/views/scren/loginscreen.dart';
import 'package:farmer/views/scren/order_screen/order_screen.dart';
import 'package:farmer/widest_common/bg_wid.dart';
import 'package:farmer/widest_common/loading_indicator.dart';
import 'package:flutter/src/widgets/framework.dart';
import 'package:flutter/src/widgets/placeholder.dart';
import 'package:get/get.dart';

class profileScreen extends StatelessWidget {
  const profileScreen({super.key});

  @override
  Widget build(BuildContext context) {
    var controller = Get.put(profileController());

    return bgwidget(
      child: Scaffold(
        body: StreamBuilder(
          stream: FireStoreServices.gtUser(currentUser!.uid),
          builder:
            (BuildContext context, AsyncSnapshot<QuerySnapshot> snapshot) {
              if (!snapshot.hasData) {
                return const Center(
                  child: CircularProgressIndicator(
                    valueColor: AlwaysStoppedAnimation(redColor)),
                );
              } else {
                var data = snapshot.data!.docs[0];
                return SafeArea(
                  child: Column(
                    children: [
                      //edit profile button
                      Padding(

```

```

padding: const EdgeInsets.all(8.0),
child: const Align(
    alignment: Alignment.topRight,
    child: const Icon(
        Icons.edit,
        color: Color.fromARGB(255, 255, 7, 7),
    ),
).onTap(() {
    controller.nameController.text = data['name'];
    controller.passwordController.text = data['password'];

    //edit profile section
    Get.to(() => EditProfileScreen(data: data));
}),
),
),

//user details section

Padding(
    padding: const EdgeInsets.all(8.0),
    child: Row(
        children: [
            data['imageUrl'] == ''
                ? Image.asset(imP1,
                    height: 90, width: 90, fit: BoxFit.cover)
                    .box
                    .roundedFull
                    .clip(Clip.antiAlias)
                    .make()
                : Image.network(data['imageUrl'],
                    height: 90, width: 90, fit: BoxFit.cover)
                    .box
                    .roundedFull
                    .clip(Clip.antiAlias)
                    .make(),
            10.widthBox,
            Expanded(
                child: Column(
                    crossAxisAlignment: CrossAxisAlignment.start,
                    children: [
                        "${data['name']}"
                            .text
                            .fontFamily(semibold)
                            .color(Color.fromARGB(255, 255, 57, 7))
                            .make(),
                        5.heightBox,
                        "${data['email']}"
                            .text
                            .color(Color.fromARGB(242, 0, 0, 0))
                            .make(),
                    ],
            ),
        ],
),
),
),

```

```

        OutlinedButton(
            style: OutlinedButton.styleFrom(
                side: BorderSide(
                    color: whiteColor,
                )),
            onPressed: () async {
                await Get.put(AuthController())
                    .signoutMethod(context);
                Get.offAll(() => LoginScreen());
            },
            child: logout.text
                .fontFamily(semibold)
                .color(Color.fromARGB(255, 245, 8, 138))
                .make(),
        )
    ],
),
),
),
),
20.heightBox,
FutureBuilder(
    future: FireStoreServices.getcount(),
    builder: (BuildContext context, AsyncSnapshot snapshot)
{
    if (!snapshot.hasData) {
        return Center(
            child: loadingIndicator(),
        );
    } else {
        var countdata = snapshot.data;
        return Row(
            children: [
                detailsCart(
                    count: countdata[0].toString(),
                    title: "আপনার কার্ট সংখ্যা",
                    width: context.screenWidth / 3.32),
                10.widthBox,
                detailsCart(
                    count: countdata[1].toString(),
                    title: "প্রিয় পণ্যের সংখ্যা",
                    width: context.screenWidth / 3.32),
                10.widthBox,
                detailsCart(
                    count: countdata[2].toString(),
                    title: "অড়ার সংখ্যা",
                    width: context.screenWidth / 3.32),
                ],
            );
        }
    },
),
// Row(
//     children: [

```

```

//      detailsCart(
//        count: data['cart_count'],
//        title: "আপনার কার্ট সংখ্যা",
//        width: context.screenWidth / 3.32),
//      10.widthBox,
//      detailsCart(
//        count: data['wishlist_count'],
//        title: "প্রিয় পণ্যের সংখ্যা",
//        width: context.screenWidth / 3.32),
//      10.widthBox,
//      detailsCart(
//        count: data['order_count'],
//        title: "অডার সংখ্যা",
//        width: context.screenWidth / 3.32),
//    ],
//  ),
20.heightBox,
//button section
ListView.separated(
  shrinkWrap: true,
  separatorBuilder: (context, index) {
    return const Divider(color: lightGrey);
},
itemCount: profilebuttonlist.length,
itemBuilder: (BuildContext context, int index) {
  return ListTile(
    onTap: () {
      switch (index) {
        case 0:
          Get.to(() => const OrderScreeen());
          break;

        case 1:
          Get.to(() => const wishlistScreen());
          break;
      }
    },
    leading: Image.asset(
      profilebuttononicon[index],
      width: 22,
    ),
    title: profilebuttonlist[index]
      .text
      .fontFamily(bold)
      .color(darkFontGrey)
      .make(),
  );
},
),
.box
.white

```

```

        .margin(EdgeInsets.all(20))
        .shadowSm
        .rounded
        .padding(EdgeInsets.symmetric(horizontal: 16))
        .make()
        .box
        .color(green)
        .make(),
    ],
)
},
),
),
);
}
}
}
}

```

## lib\Account\components\details\_cart.dart

```

import 'package:farmer/consts/consts.dart';

Widget detailsCart({width, String? count, String? title}) {
  return Column(
    mainAxisAlignment: MainAxisAlignment.center,
    children: [
      count!.text.fontFamily(bold).size(16).color(darkFontGrey).make(),
      5.heightBox,
      title!.text.color(Color.fromARGB(255, 7, 247, 255)).make(),
    ],
  )
    .box
    .color(Color.fromARGB(143, 202, 230, 210))
    .rounded
    .shadow3xl
    .width(width)
    .padding(EdgeInsets.all(4))
    .height(80)
    .make();
}

```

DOCUMENT \_HEADING

### 3 files

(file list disabled)

## lib\Cart\Shipping\_screen.dart

```
import 'package:farmer/Cart/payment_method.dart';
import 'package:farmer/consts/consts.dart';
import 'package:farmer/controller/cart_controller.dart';
import 'package:farmer/widest_common/custom_textfield.dart';
import 'package:farmer/widest_common/order_button.dart';
import 'package:flutter/src/widgets/framework.dart';
import 'package:flutter/src/widgets/placeholder.dart';
import 'package:get/get.dart';

class ShippingDetails extends StatelessWidget {
    const ShippingDetails({super.key});

    @override
    Widget build(BuildContext context) {
        var controller = Get.find<cartController>();
        return Scaffold(
            backgroundColor: whiteColor,
            appBar: AppBar(
                title: "অডার
তথ্য".text.fontFamily(semibold).color(darkFontGrey).make(),
            ),
            bottomNavigationBar: SizedBox(
                height: 60,
                child: Continue(onPress: () {
                    if (controller.addressController.text.length > 10) {
                        Get.to(() => const paymentMethods());
                    } else {
                        VxToast.show(context, msg: "Please fill the form");
                    }
                })),
            body: Padding(
                padding: const EdgeInsets.all(12),
                child: Column(
                    children: [
                        customtextfield(
                            hint: "জেলা, উপজেলা, গ্রাম",
                            isPass: false,
                            title: "ঠিকানা",
                            controller: controller.addressController),
                        customtextfield(
                            hint: "জেলা",
                            isPass: false,
                            title: "জেলা",
                            controller: controller.addressController),
                    ],
                ),
            ),
        );
    }
}
```

```

        controller: controller.districtController),
    customtextfield(
        hint: "উপজেলা",
        isPass: false,
        title: "উপজেলা",
        controller: controller.subDistrictController),
    customtextfield(
        hint: "পোস্টল কোড",
        isPass: false,
        title: "পোস্টল কোড",
        controller: controller.postalCodeController),
    customtextfield(
        hint: "+880.....",
        isPass: false,
        title: "ফোন নাম্বর",
        controller: controller.phoneNumberController),
    ],
),
);
);
}
}

```

## lib\Cart\cartScreen.dart

```

import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:farmer/Cart/Shiping_screen.dart';
import 'package:farmer/consts/consts.dart';
import 'package:farmer/controller/cart_controller.dart';
import 'package:farmer/services/firestore_services.dart';
import 'package:farmer/widest_common/cart_button.dart';
import 'package:farmer/widest_common/loading_indicator.dart';
import 'package:flutter/src/widgets/framework.dart';
import 'package:flutter/src/widgets/placeholder.dart';
import 'package:get/get.dart';

class cartScreen extends StatelessWidget {
    const cartScreen({super.key});

    @override
    Widget build(BuildContext context) {
        var controller = Get.put(cartController());
        return Scaffold(
            backgroundColor: whiteColor,
            appBar: AppBar(
                automaticallyImplyLeading: false,
                title:
                    "কার্ট তথ্য".text.color(darkFontGrey).fontFamily(semibold).make(),
            ),
        );
    }
}

```

```
body: StreamBuilder(
  stream: FireStoreServices.getCart(currentUser!.uid),
  builder:
    (BuildContext context, AsyncSnapshot<QuerySnapshot> snapshot) {
      if (!snapshot.hasData) {
        return Center(
          child: loadingIndicator(),
        );
      } else if (snapshot.data!.docs.isEmpty) {
        return Center(
          child: "Cart is Empty".text.color(redColor).make(),
        );
      } else {
        var data = snapshot.data!.docs;
        controller.calculate(data);

        controller.productSnapshot = data;

        return Padding(
          padding: const EdgeInsets.all(8.0),
          child: Column(
            children: [
              Expanded(
                child: ListView.builder(
                  itemCount: data.length,
                  itemBuilder: (BuildContext context, int index) {
                    return ListTile(
                      leading: Image.network(
                        "${data[index]['img']}",
                        width: 120,
                        fit: BoxFit.cover,
                      ),
                      title:
                        "${data[index]['title']} (q=${data[index]
['qty']})"
                          .text
                          .fontFamily(semibold)
                          .size(16)
                          .make(),
                      subtitle: "${data[index]['tprice']}"
                        .numCurrency
                        .text
                        .color(redColor)
                        .fontFamily(semibold)
                        .make(),
                      trailing:
                        Icon(Icons.delete, color: redColor).onTap(() {
                          FireStoreServices.deleteDocument(data[index].id);
                        }),
                    );
                },
              ),
            ],
          ),
        );
      }
    });

```

## lib\Cart\payment\_method.dart

```
import 'package:farmer/consts/consts.dart';
import 'package:farmer/consts/list.dart';
import 'package:farmer/controller/cart_controller.dart';
import 'package:farmer/views/screen/Home.dart';
import 'package:farmer/views/screen/homeScreen.dart';
import 'package:farmer/widest_common/loading_indicator.dart';
import 'package:flutter/src/widgets/framework.dart';
import 'package:flutter/src/widgets/placeholder.dart';
import 'package:get/get.dart';
```

```

import '../widest_common/place_order_button.dart';

class paymentMethods extends StatelessWidget {
  const paymentMethods({super.key});

  @override
  Widget build(BuildContext context) {
    var controller = Get.find<cartController>();
    return Obx(
      () => Scaffold(
        backgroundColor: whiteColor,
        bottomNavigationBar: SizedBox(
          height: 60,
          child: controller.plachingOrder.value
            ? Center(
                child: loadingIndicator(),
              )
            : procedOrder(onPress: () async {
                controller.placemyorder(
                  orderPaymentMethod:
                    paymentmthod[controller.paymentIndex.value],
                  totalAmount: controller.totalP.value);
                await controller.clearCart();
                VxToast.show(context,
                  msg:
                    "আপনার অডারটি সফল হয়েছে ,পর্ন্য হাতে পাওয়ার জন্য
কিছুদিন অপেক্ষা করুন ! ধন্যবাদ");
                Get.offAll(const Home());
              }),
      ),
      appBar: AppBar(
        title: "পেমেন্ট মাধ্যম"
          .text
          .fontFamily(semibold)
          .color(darkFontGrey)
          .make(),
      ),
      body: Padding(
        padding: const EdgeInsets.all(12.0),
        child: Obx(
          () => Column(
            children: List.generate(paymentmthod.length, (index) {
              return GestureDetector(
                onTap: () {
                  controller.changePaymentindex(index);
                },
                child: Container(
                  clipBehavior: Clip.antiAlias,
                  decoration: BoxDecoration(
                    borderRadius: BorderRadius.circular(12),
                    border: Border.all(
                      color: controller.paymentIndex.value == index

```

```
        ? redColor
        : Colors.transparent,
      width: 5,
    )),
margin: const EdgeInsets.only(bottom: 8),
child: Stack(
  alignment: Alignment.topRight,
  children: [
    Image.asset(paymentMethodlist[index],
      width: double.infinity,
      colorBlendMode: BlendMode.darken,
      color: Colors.black.withOpacity(0.5),
      height: 100,
      fit: BoxFit.cover),
    controller.paymentIndex.value == index
      ? Transform.scale(
          scale: 1.3,
          child: Checkbox(
            activeColor: redColor,
            shape: RoundedRectangleBorder(
              borderRadius:
BorderRadius.circular(50)),
            value: true,
            onChanged: (Value) {},
          ),
        )
      : Container(),
    Positioned(
      bottom: 5,
      right: 10,
      child: paymentmethod[index]
        .text
        .white
        .fontFamily(bold)
        .size(16)
        .make(),
    ],
  ),
),
),
);
},
),
),
),
),
),
),
),
),
);
}
}
```

DOCUMENT \_HEADING

## 8 files

(file list disabled)

### lib\consts\AppColors.dart

```
import 'package:flutter/cupertino.dart';

class AppColors {
    static const Color deep_orange = Color(0xFFFF6B6B);
}
```

### lib\consts\colors.dart

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

const Color textfieldGrey = Color.fromRGBO(209, 209, 209, 1);
const Color fontGrey = Color.fromRGBO(107, 115, 119, 1);
const Color darkFontGrey = Color.fromRGBO(62, 68, 71, 1);
const Color whiteColor = Color.fromRGBO(255, 255, 255, 1);
const Color lightGrey = Color.fromRGBO(239, 239, 239, 1);
const Color redColor = Color.fromRGBO(230, 46, 4, 1);
const Color golden = Color.fromRGBO(255, 168, 0, 1);
const Color green = Color.fromRGBO(21, 255, 0, 1);
```

### lib\consts\consts.dart

```
export './colors.dart';
export './colors.dart';
export './images.dart';
export './strings.dart';
export './styles.dart';
export './firebase_const.dart';
export 'package:velocity_x/velocity_x.dart';
export 'package:flutter/material.dart';
```

### lib\consts\firebase\_const.dart

```
import 'package:cloud_firestore/cloud_firestore.dart';
```

```

import 'package:firebase_auth/firebase_auth.dart';

FirebaseAuth auth = FirebaseAuth.instance;
Firestore firestore = FirebaseFirestore.instance;
User? currentUser = auth.currentUser;

//collection

const userCollection = "users";

const productCollection = "products";
const cartCollection = 'cart';
const chatscollection = 'chats';
const messageCollection = 'messages';
const ordersCollection = 'oders';

```

## lib\consts\images.dart

```

//icons
const icAdd = "assets/icons/add.png";
const icAppLogo = "assets/icons/app_logo.png";
const icAppleLogo = "assets/icons/apple_logo.png";
const icArrow = "assets/icons/arrow.png";
const icBell = "assets/icons/bell.png";
const icBlog = "assets/icons/blog.png";
const icBrands = "assets/icons/brands.png";
const icCompaigns = "assets/icons/compaigns.png";
const icCart = "assets/icons/cart.png";
const icCategories = "assets/icons/categories.png";
const icChangePassword = "assets/icons/changed_password.png";
const icChat = "assets/icons/chat.png";
const icClubpoint = "assets/icons/clubpoint.png";
const icCoupon = "assets/icons/coupon.png";
const icDownload = "assets/icons/download.png";
const icEdit = "assets/icons/edit.png";
const icFacebookLogo = "assets/icons/facebook_logo.png";
const icFavoriteSeller = "assets/icons/favoriteseller.png";
const icFlashDeal = "assets/icons/flash_deal.png";
const icGoogleLogo = "assets/icons/google_logo.png";
const icMenu = "assets/icons/o-hamburger.png";
const icHeadphone = "assets/icons/headphone.png";
const icHeart = "assets/icons/heart.png";
const icHome = "assets/icons/home.png";
const icLogin = "assets/icons/login.png";
const icLogout = "assets/icons/logout.png";
const icMessages = "assets/icons/messages.png";
const icMinus = "assets/icons/minus.png";
const icMore = "assets/icons/more.png";
const icOrder = "assets/icons/order.png";
const icOrders = "assets/icons/orders.png";

```

```

const icPlaceholderRect = "assets/icons/placeholder_rectangle.png";
const icPlaceholder = "assets/icons/placeholder.png";
const icPlus = "assets/icons/plus.png";
const icPoints = "assets/icons/points.png";
const icProfile = "assets/icons/profile.png";
const icRefund = "assets/icons/refund.png";
const icSearch = "assets/icons/search.png";
const icShop = "assets/icons/shop.png";
const icSplashBg =
    "assets/icons/splash_login_registration_background_image.png";
const icSquareLogo = "assets/icons/square_logo.png";
const icTodaysDeal = "assets/icons/todays_deal.png";
const icTopCategories = "assets/icons/top_categories.png";
const icTopSeller = "assets/icons/top_sellers.png";
const icTrash = "assets/icons/trash.png";
const icTwitterLogo = "assets/icons/twitter_logo.png";
const icWallet = "assets/icons/wallet.png";
const icWhiteTick = "assets/icons/white_tick.png";
const icWholeSale = "assets/icons/wholesale.png";

//images

const imgBackground = "assets/icons/bg.png";
const imgB1 = "assets/images/b1.jpeg";
const imgB2 = "assets/images/b2.jpeg";
const imgB3 = "assets/images/b3.jpeg";
const imgB4 = "assets/images/b4.jpeg";
const imgB5 = "assets/images/b5.jpeg";
const imgB6 = "assets/images/b6.jpeg";
const imgB7 = "assets/images/b7.jpeg";
const imgB8 = "assets/images/b8.jpeg";
const imgB9 = "assets/images/b9.jpeg";
const imgB10 = "assets/images/b10.jpeg";
const imgCod = "assets/icons/cod.jpg";
const imgFc1 = "assets/images/fc1.jpeg";
const imgFc2 = "assets/images/fc2.jpeg";
const imgFc3 = "assets/images/fc3.jpeg";
const imgFc4 = "assets/images/fc4.jpeg";
const imgFc5 = "assets/images/fc5.jpeg";
const imgFc6 = "assets/images/fc6.jpeg";
const imgFc7 = "assets/images/fc7.jpeg";
const imgFc8 = "assets/images/fc8.jpeg";
const imgFc9 = "assets/images/fc9.jpeg";
const imgFc10 = "assets/images/fc10.jpg";
const imgFd1 = "assets/images/fd1.jpg";
const imgFd2 = "assets/images/fd2.jpg";
const imgFd3 = "assets/images/fd3.jpg";
const imgFlutterwave = "assets/images/flutterwave.png";
const imgInstamojo = "assets/images/instamojo.png";
const imgIyzico = "assets/images/iyzico.png";
const imgMpesa = "assets/images/mpesa.png";
const imgNgenius = "assets/images/ngenius.png";
const imgP1 = "assets/images/p1.jpeg";

```

```
const imgP2 = "assets/images/p2.jpeg";
const imgP3 = "assets/images/p3.jpeg";
const imgP4 = "assets/images/p4.jpeg";
const imgP5 = "assets/images/p5.jpeg";
const imgP6 = "assets/images/p6.jpeg";
const imgP7 = "assets/images/p7.jpeg";
const imgPayfast = "assets/images/payfast.png";
const imgPayhere = "assets/images/payhere.png";
const imgPaypal = "assets/images/paypal.png";
const imgPaystack = "assets/images/paystack.png";
const imgPaytm = "assets/images/paytm.png";
const imgPi1 = "assets/images/pi1.jpg";
const imgPi2 = "assets/images/pi2.jpg";
const imgPi3 = "assets/images/pi3.jpg";
const imgPi4 = "assets/images/pi4.jpg";
const imgPi5 = "assets/images/pi5.jpg";

const imgProfile = "assets/images/profile_image.png";
const imgRavePayment = "assets/images/rave_payment.png";
const imgRozarpay = "assets/images/rozarpay.png";
const imgS1 = "assets/images/s1.jpg";
const imgS2 = "assets/images/s2.jpg";
const imgS3 = "assets/images/s3.jpg";
const imgS4 = "assets/images/s4.jpg";
const imgS5 = "assets/images/s5.jpg";
const imgS6 = "assets/images/s6.jpg";
const imgS7 = "assets/images/s7.jpg";
const imgS8 = "assets/images/s8.jpg";
const imgS9 = "assets/images/s9.jpg";
const imgS10 = "assets/images/s10.jpg";
const imgS11 = "assets/images/s11.jpg";
const imgS12 = "assets/images/s12.jpg";
const imgSs1 = "assets/images/ss1.jpg";
const imgSs2 = "assets/images/ss2.jpg";
const imgSs3 = "assets/images/ss3.jpg";
const imgSs4 = "assets/images/ss4.jpg";
const imgSave = "assets/images/save.png";
const imgSlider1 =
    "assets/fertilizer/26-right-placee-fertilizer-feature-min.jpg";
const imgSlider2 = "assets/fertilizer/download (2).jpg";
const imgSlider3 = "assets/fertilizer/e7780034-800px-wm.jpg";
const imgSlider4 = "assets/fertilizer/399.jpg";
const imgSslcommerz = "assets/images/sslcommerz.png";
const imgStripe = "assets/images/stripe.png";
const imgTwocheckout = "assets/images/twocheckout.png";
const imgVogue = "assets/images/vogue.png";
const imgWallet = "assets/images/wallet.png";
const igSlider1 = "assets/crops/Slide1.jpg";
const igSlider2 = "assets/crops/Slide2.jpg";
const igSlider3 = "assets/crops/Slide3.jpg";
const imSlider4 = "assets/crops/Slide4.jpg";
const f1 = "assets/fertilizer/f1.PNG";
const f2 = "assets/fertilizer/f2.PNG";
```

```

const f3 = "assets/fertilizer/f3.PNG";
const f4 = "assets/fertilizer/f4.PNG";
const f5 = "assets/fertilizer/f5.PNG";
const f6 = "assets/fertilizer/f6.jpg";
const B1 = "assets/fertilizer/B1.jpg";
const B2 = "assets/fertilizer/B2.jpg";
const B3 = "assets/fertilizer/B3.jpg";
const B4 = "assets/fertilizer/B4.jpg";
const B5 = "assets/fertilizer/B5.jpg";

const ac1 = "assets/fertilizer/c1.jpg";
const ac2 = "assets/fertilizer/c2.jpg";
const ac3 = "assets/fertilizer/c3.jpg";
const ac4 = "assets/fertilizer/c4.jpg";
const ac5 = "assets/fertilizer/c5.jpg";
const ac6 = "assets/fertilizer/c6.png";
const ac7 = "assets/fertilizer/c7.jpg";
const ac8 = "assets/fertilizer/c8.jpg";
const ac9 = "assets/fertilizer/c9.jpg";
const imP1 = "assets/fertilizer/p1.jpg";

```

## lib\consts\list.dart

```

import 'package:farmer/Cart/payment_method.dart';
import 'package:farmer/consts/consts.dart';

const Sliderlist = [imgSlider1, imgSlider2, imgSlider3, imgSlider4];

const cropslist = [igSlider1, igSlider2, igSlider3, imSlider4];
const featurelistimg1 = [f1, f2, f5];
const featurelistimg2 = [f3, f4, f6];

const featuretitle1 = [sar, bij, dhn];
const featuretitle2 = [sak, khabr, tel];
const categorilist = [
    animalmedicine,
    agriculturalMachinery,
    organicfertilizer,
    vegitable,
    plants,
    seeds,
    rice,
    feeds,
    Fruits
];
const categoriimg = [ac1, ac2, ac3, ac4, ac5, ac6, ac7, ac8, ac9];
const profilebuttonlist = [order, wishlist];
const profilebuttonicon = [icOrder, icOrder];
const paymentMethodlist = [imgCod];

```

```
const paymentmethod = [cashondelivery];
```

## lib\consts\strings.dart

```
import 'package:farmer/consts/consts.dart';
import 'package:flutter/foundation.dart';

const appname = "Smart Farmer";
const appversion = "Version 1.0.0";
const credits = "@Kamrul,@Fahim,@Plabon";
const email = "ই-মেইল";
const emailhint = "abcd@gmail.com";
const password = " পাসওয়ার্ড";
const passwordhint = "*****";
const login = "লগইন";
const forgetpassword = "পাসওয়ার্ড ভুলে গেছেন !";

const signup = "সাইন আপ";

const createanewaccount = "অথবা, নতুন একাউন্ট তৈরী করুন !";
const loginwith = "log in with";
const namehint = "আপনার নাম ";
const name = "নাম";
const repassword = "পুনরায় পাসওয়ার্ড লিখুন";
const repasswordhint = "*****";
const privacyPolicy = " নিরাপত্তা";
const termsandcondition = " শর্ত ";
const alreadyhaveanaccount = "ইতিমধ্যে একাউন্ট আছে ! ";
const home = "হোম",
    Categories = "তালিকা",
    cart = "কার্ট",
    Account = "অ্যাকাউন্ট";
const searchanything = "পন্য সার্চ করুন ---",
    todaydeal = "Pestiside Calculator",
    FlashSell = "Pestiside Calculator",
    Topsell = "Pestiside Calculator",
    Topcategory = "Top categories";
const agriculturtips = "Tips", morefuture = "পন্যের বিভাগ সমূহ";

//feature
const Fertilizer = "Fertilizer",
    seeds = "Seeds",
    vegetable = "Vegetables",
    feeds = "Feeds",
    rice = "Rice",
    oil = "oil";
const featureproduct = "জনপ্রিয় পন্য সমূহ";

const dryFruits = "Dry Fruits",
    Fruits = "Fruits",
```

```

agriculturalMachinery = "Agricultural Machinery",
organicfertilizer = "Organic Fertilizer",
animalmedicine = "Animal Medicine",
plants = "Plants";

///item details button
///
const addtocard = "কার্ট যুক্ত করুন";
const productmatalsolelike = "অন্যান্য পণ্য : ";
const loginsuccess = "Logged in successfully";
const logOutsuccess = "Logged Out successfully";
const signsucces = "Sign up successfully ";
const logout = "লগ আউট";

//
const wishlist = "প্রিয় পণ্য", order = "অডার", messege = "মেসেজ";
const pleaseEnterCurrect =
    "Wrong password or user name ! if you dont have an account you may
siggned up ";
const whether = "ক্যালকুলেটর";
const cashondelevary = "ক্যাস অন ডেলিভারি";

const sar = "কীটনাশক / সার",
sak = "শাক-সবজি",
bij = "ফসলের বীজ",
khabr = "পশু-পাখির খবার",
dhn = "ধান-চাল",
tel = "তেল";

```

## lib\consts\styles.dart

```

const regular = "sans_regular";
const semibold = "sans_semiBold";
const bold = "sans_bold";

```

DOCUMENT \_HEADING

## 6 files

(file list disabled)

### lib\controller\auth\_controller.dart

```
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:farmer/consts/consts.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:get/get.dart';

class AuthController extends GetxController {
    var isloading = false.obs;
    //text controller

    var emailController = TextEditingController();
    var passwordController = TextEditingController();

    // login method

    Future<UserCredential?> loginMethod({context}) async {
        UserCredential? userCredential;
        try {
            await auth.signInWithEmailAndPassword(
                email: emailController.text, password: passwordController.text);
        } on FirebaseAuthException catch (e) {
            VxToast.show(context, msg: e.toString());
        }
        return userCredential;
    }

    //signup method
    Future<UserCredential?> signupMethod({email, password, context}) async {
        UserCredential? userCredential;

        try {
            await auth.createUserWithEmailAndPassword(
                email: email, password: password);
        } on FirebaseAuthException catch (e) {
            VxToast.show(context, msg: e.toString());
        }
        return userCredential;
    }

    //
    //storing data
    storeusedata({name, password, email}) async {
        DocumentReference store =
            await firestore.collection(userCollection).doc(currentUser!.uid);
```

```

        store.set({
            'name': name,
            'password': password,
            'email': email,
            'imageUrl': '',
            'id': currentUser!.uid,
            'cart_count': "00",
            'order_count': "00",
            'wishlist_count': "00",
        });
    }

//signout method
signoutMethod(context) async {
    try {
        await auth.signOut();
    } catch (e) {
        VxToast.show(context, msg: e.toString());
    }
}
}

```

## lib\controller\cart\_controller.dart

```

import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:farmer/consts/consts.dart';
import 'package:farmer/controller/home_controller.dart';
import 'package:get/get.dart';

class cartController extends GetxController {
    var totalP = 0.obs;

//text controller for shipping details

    var addressController = TextEditingController();
    var districtController = TextEditingController();
    var subDistrictController = TextEditingController();
    var postalCodeController = TextEditingController();
    var phoneNumberController = TextEditingController();
    var paymentIndex = 0.obs;
    late dynamic productSnapshot;
    var Products = [];
    var plachingOrder = false.obs;

    calculate(data) {
        totalP.value = 0;
        for (var i = 0; i < data.length; i++) {
            totalP.value = totalP.value + int.parse(data[i]['tprice'].toString());
        }
    }
}

```

```

    }

    changePaymentindex(index) {
        paymentIndex.value = index;
    }

    placemyorder({required orderPaymentMethod, required totalAmount}) async {
        plachingOrder(true);
        await GetProductDetails();
        await firestore.collection(ordersCollection).doc().set({
            'order_code': '233981237',
            'order_date': FieldValue.serverTimestamp(),
            'order_by': currentUser!.uid,
            'order_by_name': Get.find<HomeController>().username,
            'order_by_email': currentUser!.email,
            'order_by_address': addressController.text,
            'order_by_district': districtController.text,
            'order_by_sub_district': subDistrictController.text,
            'order_by_postal_code': postalCodeController.text,
            'order_by_phone': phoneNumberController.text,
            'shipping_method': "Home Delivery",
            'payment_method': orderPaymentMethod,
            'order_confirmed': false,
            'order_delivered': false,
            'order_on_delivery': false,
            'order_placed': true,
            'total_amount': totalAmount,
            'oders': FieldValue.arrayUnion(Products)
        });
        plachingOrder(false);
    }

    GetProductDetails() {
        Products.clear();
        for (var i = 0; i < productSnapshot.length; i++) {
            Products.add({
                'img': productSnapshot[i]['img'],
                'vendor_id': productSnapshot[i]['vendor_id'],
                'tprice': productSnapshot[i]['tprice'],
                'qty': productSnapshot[i]['qty'],
                'title': productSnapshot[i]['title'],
            });
        }
    }

    clearCart() {
        for (var i = 0; i < productSnapshot.length; i++) {

            firestore.collection(cartCollection).doc(productSnapshot[i].id).delete();
        }
    }
}

```

## lib\controller\chats\_controller.dart

```
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:farmer/consts/consts.dart';
import 'package:farmer/controller/home_controller.dart';
import 'package:get/get.dart';

class chatController extends GetxController {
    @override
    void onInit() {
        getchatId();
        // TODO: implement onInit
        super.onInit();
    }

    var chats = firestore.collection(chatscollection);

    var friendName = Get.arguments[0];
    var friendId = Get.arguments[1];

    var senderName = Get.find<HomeController>().username;
    var currentId = currentUser!.uid;

    var msgController = TextEditingController();

    dynamic chatDocId;
    var isloading = false.obs;

    getchatId() async {
        isloading(true);
        await chats
            .where('users', isEqualTo: {friendId: null, currentId: null})
            .limit(1)
            .get()
            .then((QuerySnapshot snapshot) {
                if (snapshot.docs.isNotEmpty) {
                    chatDocId = snapshot.docs.single.id;
                } else {
                    chats.add({
                        'created_on': null,
                        'last_msg': '',
                        'users': {friendId: null, currentId: null},
                        'toId': '',
                        'fromId': '',
                        'friend_name': friendName,
                        'sender_name': senderName,
                    }).then((value) {
                        chatDocId = value.id;
                    });
                }
            });
    }
}
```

```
        }
    });
}
});
isloading(false);
}

SendMsg(String msg) async {
if (msg.trim().isNotEmpty) {
chats.doc(chatDocId).update({
'created_on': FieldValue.serverTimestamp(),
'last_msg': msg,
'told': friendId,
'formId': currentId,
});

chats.doc(chatDocId).collection(messageCollection).doc().set({
'created_on': FieldValue.serverTimestamp(),
'msg': msg,
'uid': currentId,
});
}
}
}
```

## lib\controller\home\_controller.dart

```
import 'package:farmer/consts/consts.dart';
import 'package:get/get.dart';

class HomeController extends GetxController {
@Override
void onInit() {
getUsername();
// TODO: implement onInit
super.onInit();
}

var currentNavIndex = 0.obs;
var username = '';
var searchController = TextEditingController();

getUsername() async {
var n = await firestore
.collection(userCollection)
.where('id', isEqualTo: currentUser!.uid)
.get()
.then((value) {
if (value.docs.isNotEmpty) {
```

```

        return value.docs.single['name'];
    }
});

username = n;
}
}

```

## lib\controller\product\_controller.dart

```

import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:farmer/Model/category_model.dart';
import 'package:farmer/consts/consts.dart';
import 'package:flutter/services.dart';
import 'package:get/get.dart';
import 'package:get/get_state_manager/src/simple/get_controllers.dart';

class ProductController extends GetxController {
    var quantity = 0.obs;

    var totalprice = 0.obs;

    var subcat = [];
    var isFav = false.obs;

    getSubCategories(title) async {
        subcat.clear();
        var data = await rootBundle.loadString("lib/services/Category_model.json");
        var decoded = categoryModelFromJson(data);
        var s =
            decoded.categories.where((element) => element.name == title).toList();

        for (var e in s[0].subcategory) {
            subcat.add(e);
        }
    }

    increasequantity(totalQuantity) {
        if (quantity.value < totalQuantity) quantity.value++;
    }

    decreasequantity() {
        if (quantity.value > 0) quantity.value--;
    }

    calalculateTotalprice(price) {
        totalprice.value = price * quantity.value;
    }
}

```

```

addtoCart({title, img, sellername, qty, tprice, context, vendorID}) async {
  await firestore.collection(cartCollection).doc().set({
    'title': title,
    'img': img,
    'sellername': sellername,
    'qty': qty,
    'vendor_id': vendorID,
    'tprice': tprice,
    'added_by': currentUser!.uid
  }).catchError((error) {
    VxToast.show(context, msg: error.toString());
  });
}

resetvalue() {
  totalprice.value = 0;
  quantity.value = 0;
}

addtoWishlist(docId, context) async {
  await firestore.collection(productCollection).doc(docId).set({
    'p_wishlist': FieldValue.arrayUnion([currentUser!.uid])
  }, SetOptions(merge: true));
  isFav(true);
  VxToast.show(context, msg: "Added to wishlist");
}

removeFromWishlist(docId, context) async {
  await firestore.collection(productCollection).doc(docId).set({
    'p_wishlist': FieldValue.arrayRemove([currentUser!.uid])
  }, SetOptions(merge: true));

  isFav(false);

  VxToast.show(context, msg: "Remove from wishlist");
}

checkifFav(data) async {
  if (data['p_wishlist'].contains(currentUser!.uid)) {
    isFav(true);
  } else {
    isFav(false);
  }
}
}

```

## lib\controller\profile\_controller.dart

```
import 'dart:io';
```

```
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:farmer/consts/consts.dart';
import 'package:firebase_storage/firebase_storage.dart';
import 'package:flutter/services.dart';
import 'package:get/get.dart';
import 'package:image_picker/image_picker.dart';
import 'package:path/path.dart';

class profileController extends GetxController {
    var profileimgPath = ''.obs;
    var profileImageLink = '';

//text field

var nameController = TextEditingController();
var passwordController = TextEditingController();
var isloading = false.obs;

changeImage(context) async {
    try {
        final img = await ImagePicker()
            .pickImage(source: ImageSource.gallery, imageQuality: 70);
        if (img == null) return;
        profileimgPath.value = img.path;
    } on PlatformException catch (e) {
        VxToast.show(context, msg: e.toString());
    }
}

//image method
uploadprofileimage() async {
    var filename = basename(profileimgPath.value);
    var destination = 'images/${currentUser!.uid}/$filename';
    Reference ref = FirebaseStorage.instance.ref().child(destination);
    await ref.putFile(File(profileimgPath.value));
    profileImageLink = await ref.getDownloadURL();
}

updatProfile({name, password, imgUrl}) async {
    var store = firestore.collection(userCollection).doc(currentUser!.uid);
    await store.set({'name': name, 'password': password, 'imageUrl': imgUrl},
        SetOptions(merge: true));
    isloading(false);
}
}
```

DOCUMENT \_HEADING

## 4 files

(file list disabled)

### lib\ctegori\button\buybutton.dart

```
import 'package:farmer/consts/consts.dart';

Widget button1({onPress, color}) {
    return ElevatedButton(
        style: ElevatedButton.styleFrom(
            backgroundColor: color,
            padding: const EdgeInsets.all(12),
        ),
        onPressed: onPress,
        child: addtocard.text.white.fontFamily(bold).make(),
    );
}
```

### lib\ctegori\catagori\_details.dart

```
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:farmer/consts/consts.dart';
import 'package:farmer/controller/product_controller.dart';
import 'package:farmer/ctegori/items_details.dart';
import 'package:farmer/services/firestore_services.dart';
import 'package:farmer/widest_common/bg_wid.dart';
import 'package:farmer/widest_common/loading_indicator.dart';
import 'package:flutter/src/widgets/framework.dart';
import 'package:flutter/src/widgets/placeholder.dart';
import 'package:get/get.dart';

class categorydetails extends StatefulWidget {
    final String? title;
    const categorydetails({super.key, required this.title});

    @override
    State<categorydetails> createState() => _categorydetailsState();
}

class _categorydetailsState extends State<categorydetails> {
    @override
    void initState() {
        // TODO: implement initState
        super.initState();
    }
}
```

```

        switchcategory(widget.title);
    }

switchcategory(title) {
    if (controller.subcat.contains(title)) {
        productMethod = FireStoreServices.getsubcategory(title);
    } else {
        productMethod = FireStoreServices.getproduct(title);
    }
}

var controller = Get.find<ProductController>();
dynamic productMethod;
@override
Widget build(BuildContext context) {
    return bgwidget(
        child: Scaffold(
            appBar: AppBar(
                title: widget.title!.text.fontFamily(bold).white.make(),
            ),
            body: Column(
                mainAxisAlignment: MainAxisAlignment.start,
                children: [
                    SingleChildScrollView(
                        physics: const BouncingScrollPhysics(),
                        scrollDirection: Axis.horizontal,
                        child: Row(
                            children: List.generate(
                                controller.subcat.length,
                                (index) => "${controller.subcat[index]}"
                                    .text
                                    .size(12)
                                    .fontFamily(semibold)
                                    .color(darkFontGrey)
                                    .makeCentered()
                                    .box
                                    .white
                                    .rounded
                                    .size(120, 70)
                                    .margin(const EdgeInsets.symmetric(horizontal: 5))
                                    .make()
                                    .onTap(() {
                                        switchcategory("${controller.subcat[index]}");
                                        setState(() {});
                                    })),
                            ),
                    ),
                    20.heightBox,
                    StreamBuilder(
                        stream: productMethod,
                        builder:
                            (BuildContext context, AsyncSnapshot<QuerySnapshot> snapshot)
{

```

```

    if (!snapshot.hasData) {
        return Expanded(
            child: Center(
                child: loadingIndicator(),
            ),
        );
    } else if (snapshot.data!.docs.isEmpty) {
        return Expanded(
            child: " দুঃখিত ! এই সাব ক্যাটাগরিতে কোনো পণ্য নাই !"
                .text
                .color(redColor)
                .makeCentered(),
        );
    } else {
        var data = snapshot.data!.docs;
        return

            //item container
            Expanded(
                child: GridView.builder(
                    physics: BouncingScrollPhysics(),
                    shrinkWrap: true,
                    itemCount: data.length,
                    gridDelegate:
                        SliverGridDelegateWithFixedCrossAxisCount(
                            crossAxisCount: 2,
                            mainAxisExtent: 250,
                            mainAxisSpacing: 10,
                            crossAxisSpacing: 10),
                    itemBuilder: (context, index) {
                        return Column(
                            crossAxisAlignment: CrossAxisAlignment.start,
                            children: [
                                Image.network(data[index]['p_imgs'][0],
                                    height: 150,
                                    width: 200,
                                    fit: BoxFit.cover),
                                10.heightBox,
                                "${data[index]['p_name']}"
                                    .text
                                    .fontFamily(semibold)
                                    .color(darkFontGrey)
                                    .make(),
                                10.heightBox,
                                "${data[index]['p_price']}"
                                    .numCurrency
                                    .text
                                    .color(redColor)
                                    .fontFamily(bold)
                                    .size(16)
                                    .make(),
                            ],
                        )
                )
            }
    }
}

```

```
        .box
        .color(green)
        .margin(
            const EdgeInsets.symmetric(horizontal:
6))
        .rounded
        .padding(EdgeInsets.all(6))
        .makeCentered()
        .onTap(() {
    controller.checkifFav(data[index]);
    Get.to(() => itemdetails(
        title: "${data[index]['p_name']}",
        data: data[index]));
});
})
},
),
],
),
));
}
}
```

## lib\ctegori\categories\_screen.dart

```
import 'package:farmer/consts/consts.dart';
import 'package:farmer/consts/list.dart';
import 'package:farmer/controller/product_controller.dart';
import 'package:farmer/ctegori/catagori_details.dart';
import 'package:farmer/widest_common/bg_wid.dart';
import 'package:flutter/src/widgets/framework.dart';
import 'package:flutter/src/widgets/placeholder.dart';
import 'package:get/get.dart';

class categoriesScreen extends StatelessWidget {
    const categoriesScreen({super.key});

    @override
    Widget build(BuildContext context) {
        var controller = Get.put(ProductController());
        return bgwidget(
            child: Scaffold(
                appBar: AppBar(
                    title: Categories.text.fontFamily(bold).color(whiteColor).make(),
                ),
                body: Container(
                    padding: EdgeInsets.all(12),
                    child: GridView.builder(
                        shrinkWrap: true,
```

```

        itemCount: 9,
        gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(
            crossAxisCount: 3,
            mainAxisSpacing: 8,
            crossAxisSpacing: 8,
            mainAxisExtent: 200),
        itemBuilder: (context, index) {
            return Column(
                children: [
                    Image.asset(
                        categoriimg[index],
                        height: 130,
                        width: 200,
                        fit: BoxFit.cover,
                    ),
                    10.heightBox,
                    "${categorilist[index]}"
                        .text
                        .black
                        .align(TextAlign.center)
                        .make()
                ],
            )
            .box
            .color(Color.fromARGB(213, 7, 255, 255))
            .roundedSM
            .clip(Clip.antiAlias)
            .outerShadowSm
            .make()
            .onTap(() {
                controller.getSubCategories(categorilist[index]);
                Get.to(() => categorydetails(title: categorilist[index]));
            });
        },
    )));
}
}

```

## lib\ctegori\items\_details.dart

```

import 'package:farmer/consts/consts.dart';
import 'package:farmer/controller/product_controller.dart';
import 'package:farmer/ctegori/button/buybutton.dart';
import 'package:farmer/views/chat_screen/chat_screen.dart';
import 'package:farmer/widest_common/our_button.dart';
import 'package:flutter/src/widgets/framework.dart';
import 'package:flutter/src/widgets/placeholder.dart';
import 'package:flutter_screenutil/flutter_screenutil.dart';

```

```
import 'package:get/get.dart';
import 'package:get/get_core/src/get_main.dart';

class itemdetails extends StatelessWidget {
    final String? title;
    final dynamic data;
    const itemdetails({super.key, required this.title, this.data});

    @override
    Widget build(BuildContext context) {
        var controller = Get.put(ProductController());
        //var controller = Get.put(ProductController());
        return WillPopScope(
            onWillPop: () async {
                controller.resetvalue();
                return true;
            },
            child: Scaffold(
                backgroundColor: lightGrey,
                appBar: AppBar(
                    leading: IconButton(
                        onPressed: () {
                            controller.resetvalue();
                            Get.back();
                        },
                        icon: Icon(Icons.arrow_back)),
                    title: title!.text.color(darkFontGrey).fontFamily(bold).make(),
                    actions: [
                        IconButton(
                            onPressed: () {},
                            icon: Icon(
                                Icons.share,
                                color: darkFontGrey,
                            )),
                        Obx(
                            () => IconButton(
                                onPressed: () {
                                    if (controller.isFav.value) {
                                        controller.removeFromWishlist(data.id, context);
                                    } else {
                                        controller.addToWishlist(data.id, context);
                                    }
                                },
                                icon: Icon(
                                    Icons.favorite_border_outlined,
                                    color: controller.isFav.value ? redColor : darkFontGrey,
                                )),
                        )
                    ],
                ),
                body: Column(
                    children: [
                        Expanded(
                            child: Padding(
```

```
padding: const EdgeInsets.all(8),
child: SingleChildScrollView(
  physics: BouncingScrollPhysics(),
  child: Column(
    mainAxisAlignment: CrossAxisAlignment.start,
    children: [
      Column(
        children: [
          //swiper section

          VxSwiper.builder(
            autoPlay: true,
            height: 350,
            itemCount: data['p_imgs'].length,
            aspectRatio: 16 / 9,
            viewportFraction: 1.0,
            itemBuilder: (context, index) {
              return Image.network(data["p_imgs"][index],
                width: double.infinity, fit: BoxFit.cover);
            },
          ),

          10.heightBox,
          //title and details selection

          title!.text
            .size(16)
            .color(darkFontGrey)
            .fontFamily(bold)
            .make(),
          10.heightBox,
          //rating

          VxRating(
            isSelectable: false,
            value: double.parse(data['p_rating']),
            onRatingUpdate: (value) {},
            normalColor: textfieldGrey,
            selectionColor: green,
            count: 5,
            maxRating: 5,
            size: 25,
            stepInt: false,
          ),
        ],
      ),
    ],
  ),
  " ${data['p_price']}"
    .numCurrency
    .text
    .color(redColor)
    .fontFamily(bold)
    .size(18)
    .make(),
  10.heightBox,
```

```

Row(
  children: [
    Expanded(
      child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
          "বিক্রেতা".text.blue100.make(),
          5.heightBox,
          "${data['p_seller']}"" .text
            .fontFamily(semibold)
            .color(darkFontGrey)
            .size(16)
            .make(),
        ],
      )),
    CircleAvatar(
      backgroundColor: Colors.white,
      child: Icon(Icons.message_rounded,
        color: darkFontGrey),
      onTap(() {
        Get.to(() => const ChatScreen(), arguments: [
          data['p_seller'],
          data['vendor_id']
        ]);
      })),
    ],
  )
  .box
  .height(60)
  .padding(EdgeInsets.symmetric(horizontal: 16))
  .color(textfieldGrey)
  .make(),
),
),
Column(
  children: [
    Row(
      children: [
        SizedBox(
          width: 100,
          child:
            "চিহ্নিতঃ".text.color(textfieldGrey).make(),
        ),
      ],
    ).box.padding(const EdgeInsets.all(8)).make(),
],
),
).box.white.shadowSm.make(),
//color section
20.heightBox,
Column(
  children: [

```

```

Row(
    children: [
        SizedBox(
            width: 100,
        ),
    ],
).box.padding(const EdgeInsets.all(8)).make(),
// Quantity row
Row(
    children: [
        SizedBox(
            width: 100,
            child:
"পরিমাণ".text.color(textfieldGrey).make(),
        ),
        Obx(
            () => Row(
                children: [
                    IconButton(
                        onPressed: () {
                            controller.decreasequantity();
                            controller.calalculateTotalprice(
                                int.parse(data['p_price']));
                        },
                        icon: Icon(Icons.remove)),
                    controller.quantity.value.text
                        .size(16)
                        .color(darkFontGrey)
                        .fontFamily(bold)
                        .make(),
                    IconButton(
                        onPressed: () {
                            controller.increasequantity(
                                int.parse(data['p_quantity']));
                            controller.calalculateTotalprice(
                                int.parse(data['p_price']));
                        },
                        icon: Icon(Icons.add)),
                    5.widthBox,
                    10.widthBox,
                    " (Available ${data['p_quantity']})"
                        .text
                        .size(18)
                        .color(textfieldGrey)
                        .make(),
                ],
            ),
        ),
    ],
).box.padding(const EdgeInsets.all(8)).make(),

//total section
Row(

```

```

        children: [
          SizedBox(
            width: 100,
            child: "মোট টাকা"
              .text
              .size(20)
              .color(textfieldGrey)
              .make(),
          ),
          Row(
            children: [
              "${controller.totalprice.value}"
                .numCurrency
                .text
                .size(16)
                .color(redColor)
                .fontFamily(bold)
                .size(20)
                .make(),
            ],
          ),
        ],
      ).box.padding(const EdgeInsets.all(8)).make(),
    ],
  ).box.white.shadowSm.make(),
//discription

"Discription : "
  .text
  .color(darkFontGrey)
  .fontFamily(bold)
  .make(),
10.heightBox,
"${data['p_des']}"
  .text
  .color(darkFontGrey)
  .fontFamily(bold)
  .make(),

20.heightBox,
//product may also like

//productmatalso.like.text
// .fontFamily(bold)
//   .size(25)
//   .color(darkFontGrey)
//   .make(),
20.heightBox,
SingleChildScrollView(
  scrollDirection: Axis.horizontal,
  child: Row(),
),
],

```

```
        ),
        ),
    )),
SizedBox(
    height: 60,
    width: double.infinity,
    child: button1(
        color: green,
        onPressed: () {
            controller.addtoCart(
                context: context,
                vendorID: data['vendor_id'],
                img: data['p_imgs'][0],
                qty: controller.quantity.value,
                sellername: data['p_seller'],
                title: data['p_name'],
                tprice: controller.totalprice.value);
            VxToast.show(context, msg: "Added to cart");
        },
        ),
    ],
),
),
);
}
}
```

DOCUMENT \_HEADING

## 1 files

(file list disabled)

### lib\Model\category\_model.dart

```
// To parse this JSON data, do
//
//     final categoryModel = categoryModelFromJson(jsonString);

import 'package:meta/meta.dart';
import 'dart:convert';

CategoryModel categoryModelFromJson(String str) =>
    CategoryModel.fromJson(json.decode(str));

String categoryModelToJson(CategoryModel data) => json.encode(data.toJson());

class CategoryModel {
    CategoryModel({
        required this.categories,
    });

    List<Category> categories;

    factory CategoryModel.fromJson(Map<String, dynamic> json) => CategoryModel(
        categories: List<Category>.from(
            json["categories"].map((x) => Category.fromJson(x))),
    );

    Map<String, dynamic> toJson() => {
        "categories": List<dynamic>.from(categories.map((x) => x.toJson())),
    };
}

class Category {
    Category({
        required this.name,
        required this.subcategory,
    });

    String name;
    List<String> subcategory;

    factory Category.fromJson(Map<String, dynamic> json) => Category(
        name: json["name"],
        subcategory: List<String>.from(json["subcategory"].map((x) => x)),
    );
}
```

```
Map<String, dynamic> toJson() => {
    "name": name,
    "subcategory": List<dynamic>.from(subcategory.map((x) => x)),
};

}
```

DOCUMENT\_HEADING

## 2 files

(file list disabled)

### lib\services\Category\_model.json

```
{
  "categories": [
    {
      "name": "Animal Medicine",
      "subcategory": [
        "Birds Medicine",
        "Cows/goat Medicine",
        "Others Medicine"
      ]
    },
    {
      "name": "Agricultural Machinery",
      "subcategory": [
        "Tractor & Power",
        "Soil Cultivation",
        "Planting"
      ]
    },
    {
      "name": "Organic Fertilizer",
      "subcategory": [
        "Mixed Fertilizer",
        "Home Made"
      ]
    },
    {
      "name": "Vegetables",
      "subcategory": [
        "Vegetables leaves",
        "Fruits Vegetables"
      ]
    },
    {
      "name": "Plants",
      "subcategory": [
        "All Plants"
      ]
    }
  ]
}
```

```

        "name": "Seeds",
        "subcategory": [
            "All Seeds"
        ]
    },
    {
        "name": "Rice",
        "subcategory": [
            "Basmati Rice",
            "Jasmine Rice",
            "Calrose Rice",
            "others"
        ]
    },
    {
        "name": "Feeds",
        "subcategory": [
            "All Feeds for animal"
        ]
    },
    {
        "name": "Fruits",
        "subcategory": [
            "All seasonal Fruits "
        ]
    }
]
}

```

## **lib\services\firestore\_services.dart**

```

import 'package:farmer/Model/category_model.dart';
import 'package:farmer/consts/consts.dart';

class FireStoreServices {
    //get user data
    static gtUser(uid) {
        return firestore
            .collection(userCollection)
            .where('id', isEqualTo: uid)
            .snapshots();
    }
    // Get product according to category
}

```

```

static getproduct(Category) {
    return firestore
        .collection(productCollection)
        .where('p_category', isEqualTo: Category)
        .snapshots();
}

static getsubcategory(title) {
    return firestore
        .collection(productCollection)
        .where('p_subcategory', isEqualTo: title)
        .snapshots();
}

//getcarrt

static getCart(uid) {
    return firestore
        .collection(cartCollection)
        .where('added_by', isEqualTo: uid)
        .snapshots();
}

//delecte document

static deleteDocument(docId) {
    return firestore.collection(cartCollection).doc(docId).delete();
}

//get all chat messege
static getchatMessage(docId) {
    return firestore
        .collection(chatscollection)
        .doc(docId)
        .collection(messageCollection)
        .orderBy('created_on', descending: false)
        .snapshots();
}

static getallOrders() {
    return firestore
        .collection(ordersCollection)
        .where('order_by', isEqualTo: currentUser!.uid)
        .snapshots();
}

static getWishlist() {
    return firestore
        .collection(productCollection)
        .where('p_wishlist', arrayContains: currentUser!.uid)
        .snapshots();
}

```

```

static getallMessage() {
    return firestore
        .collection(chatscollection)
        .where('fromId', isEqualTo: currentUser!.uid)
        .snapshots();
}

static getCount() async {
    var res = await Future.wait([
        firestore
            .collection(cartCollection)
            .where('added_by', isEqualTo: currentUser!.uid)
            .get()
            .then((value) {
                return value.docs.length;
            }),
        firestore
            .collection(productCollection)
            .where('p_wishlist', arrayContains: currentUser!.uid)
            .get()
            .then((value) {
                return value.docs.length;
            }),
        firestore
            .collection(ordersCollection)
            .where('order_by', isEqualTo: currentUser!.uid)
            .get()
            .then((value) {
                return value.docs.length;
            })
    ]);
    return res;
}

static allproduct() {
    return firestore.collection(productCollection).snapshots();
}

static GetFeatureProduct() {
    return firestore
        .collection(productCollection)
        .where('isFeatured', isEqualTo: true)
        .get();
}

static searchproduct(title) {
    return firestore.collection(productCollection).get();
}
}

```

DOCUMENT \_HEADING

## 16 files

(file list disabled)

### lib\views\chat\_screen\chat\_screen.dart

```
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:farmer/consts/consts.dart';
import 'package:farmer/controller/chats_controller.dart';
import 'package:farmer/services/firestore_services.dart';
import 'package:farmer/views/chat_screen/component/sender_bubble.dart';
import 'package:farmer/widest_common/loading_indicator.dart';
import 'package:get/get.dart';

class ChatScreen extends StatelessWidget {
    const ChatScreen({super.key});

    @override
    Widget build(BuildContext context) {
        var controller = Get.put(chatController());
        return Scaffold(
            backgroundColor: whiteColor,
            appBar: AppBar(
                title: "${controller.friendName}"
                    .text
                    .fontFamily(semibold)
                    .color(darkFontGrey)
                    .make(),
            ),
            body: Padding(
                padding: const EdgeInsets.all(8.0),
                child: Column(
                    children: [
                        Obx(
                            () => controller.isloading.value
                                ? Center(
                                    child: loadingindicator(),
                                )
                                : Expanded(
                                    child: StreamBuilder(
                                        stream: FireStoreServices.getchatMessage(
                                            controller.chatDocId.toString()),
                                        builder: (BuildContext context,
                                            AsyncSnapshot<QuerySnapshot> snapshot) {
                                            if (!snapshot.hasData) {
                                                return Center(
                                                    child: loadingindicator(),
                                                );
                                            } else if (snapshot.data!.docs.isEmpty) {

```



```

        .height(80)
        .padding(const EdgeInsets.all(12))
        .margin(const EdgeInsets.only(bottom: 8))
        .make(),
    ],
),
),
);
}
}

```

## lib\views\chat\_screen\component\sender\_bubble.dart

```

import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:farmer/consts/consts.dart';
import 'package:intl/intl.dart' as intl;

Widget senderBubble(DocumentSnapshot data) {
  var t =
    data['created_on'] == null ? DateTime.now() :
  data['created_on'].toDate();
  var time = intl.DateFormat("h.mma").format(t);
  return Directionality(
    textDirection:
      data['uid'] == currentUser!.uid ? TextDirection.rtl :
    TextDirection.ltr,
    child: Container(
      padding: const EdgeInsets.all(8),
      margin: const EdgeInsets.only(bottom: 8),
      decoration: BoxDecoration(
        color: data['uid'] == currentUser!.uid
          ? Color.fromARGB(88, 180, 1, 1)
          : darkFontGrey,
        borderRadius: BorderRadius.only(
          topLeft: Radius.circular(20),
          topRight: Radius.circular(20),
          bottomLeft: Radius.circular(20),
          bottomRight: Radius.circular(20),
        )),
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
          "${data['msg']}".text.size(16).white.make(),
          10.heightBox,
          time.text.color(whiteColor.withOpacity(0.5)).make(),
        ],
      ),
    ),
  );
}

```

}

## lib\views\chat\_screen\messegingscreen.dart

```
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:farmer/consts/consts.dart';
import 'package:farmer/services/firestore_services.dart';
import 'package:farmer/widest_common/loading_indicator.dart';
import 'package:flutter/src/widgets/framework.dart';
import 'package:flutter/src/widgets/placeholder.dart';

class MessegessScreen extends StatelessWidget {
  const MessegessScreen({super.key});

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: whiteColor,
      appBar: AppBar(
        title: "মেসেজ".text.color(darkFontGrey).fontFamily(semibold).make(),
      ),
      body: StreamBuilder(
        stream: FireStoreServices.getallMessage(),
        builder:
            (BuildContext context, AsyncSnapshot<QuerySnapshot> snapshot) {
          if (!snapshot.hasData) {
            return Center(
              child: loadingIndicator(),
            );
          } else if (snapshot.data!.docs.isEmpty) {
            return "কোনো বার্তা নাই!!"
                .text
                .color(darkFontGrey)
                .makeCentered();
          } else {
            var data = snapshot.data!.docs;
            return Column(
              children: [
                Expanded(
                  child: ListView.builder(
                    itemCount: data.length,
                    itemBuilder: (BuildContext context, int index) {
                      return ListTile(
                        title: "${data[index]['friend_name']}"
                            .text
                            .fontFamily(semibold)
                            .color(darkFontGrey)
                            .make(),
                      );
                },
              ],
            );
          }
        }
      );
  }
}
```

```
        ),
        )
    ],
);
}
);
}
}
```

## lib\views\edit\_profile\_screen.dart

```
import 'dart:io';

import 'package:farmer/consts/consts.dart';
import 'package:farmer/controller/profile_controller.dart';
import 'package:farmer/widest_common/bg_wid.dart';
import 'package:farmer/widest_common/custom_testfield.dart';
import 'package:farmer/widest_common/ourEditbutton2.dart';
import 'package:farmer/widest_common/ourEditscreenButton.dart';
import 'package:flutter/src/widgets/framework.dart';
import 'package:flutter/src/widgets/placeholder.dart';
import 'package:get/get.dart';

class EditProfileScreen extends StatelessWidget {
    final dynamic data;

    const EditProfileScreen({super.key, this.data});

    @override
    Widget build(BuildContext context) {
        var controller = Get.find<profileController>();

        return bgwidget(
            child: Scaffold(
                appBar: AppBar(),
                body: Obx(
                    () => Column(
                        mainAxisSize: MainAxisSize.min,
                        children: [
                            //if image url and controller path is empty
                            data['imageUrl'] == '' && controller.profileimgPath.isEmpty
                                ? Image.asset(imP1, height: 90, width: 90, fit: BoxFit.cover)
                                    .box
                                    .roundedFull
                                    .clip(Clip.antiAlias)
                                    .make()
                            //if image url is noy empty but controller is empty
                            : data['imageUrl'] != '' && controller.profileimgPath.isEmpty

```

```
        ? Image.network(
            data['imageUrl'],
            width: 90,
            height: 90,
            fit: BoxFit.cover,
        ).box.roundedFull.clip(Clip.antiAlias).make()
        //both are empty
        : Image.file(
            File(controller.profileImgPath.value),
            width: 90,
            height: 90,
            fit: BoxFit.cover,
        ).box.roundedFull.clip(Clip.antiAlias).make(),
    10.heightBox,
    ourEditbutton(onPress: () {
        controller.changeImage(context);
   }),
    Divider(),
    20.heightBox,
    customtextfield(
        controller: controller.nameController,
        hint: namehint,
        title: name,
        isPass: false),
    customtextfield(
        controller: controller.passwordController,
        hint: passwordhint,
        title: password,
        isPass: true),
    20.heightBox,
    controller.isloading.value
        ? const CircularProgressIndicator(
            valueColor: AlwaysStoppedAnimation(redColor),
        )
        : SizedBox(
            width: context.screenWidth - 60,
            child: ourEditbutton2(onPress: () async {
                controller.isloading(true);
                await controller.uploadprofileimage();
                await controller.updateProfile(
                    imgUrl: controller.profileImageLink,
                    name: controller.nameController.text,
                    password: controller.passwordController.text,
                );
                VxToast.show(context, msg: "Uploaded");
            })),
    ],
)
    .box
    .white
    .shadowSm
    .rounded
    .padding(EdgeInsets.all(16))
```

```

        .margin(EdgeInsets.only(top: 50, left: 12, right: 12))
        .make(),
    ),
));
}
}

```

## lib\views\screen\Home.dart

```

import 'dart:async';

import 'package:farmer/Account/account.dart';
import 'package:farmer/Cart/cartScreen.dart';
import 'package:farmer/consts/consts.dart';
import 'package:farmer/controller/home_controller.dart';
import 'package:farmer/ctegori/categories_screen.dart';
import 'package:farmer/views\scren\homeScreen.dart';
import 'package:farmer/widest_common/exit_dialog.dart';
import 'package:flutter/material.dart';
import 'package:get/get.dart';

class Home extends StatelessWidget {
  const Home({super.key});

  @override
  Widget build(BuildContext context) {
    var controller = Get.put(HomeController());
    var navbaritem = [
      BottomNavigationBarItem(
          icon: Image.asset(icHome, width: 26), label: home),
      BottomNavigationBarItem(
          icon: Image.asset(icCategories, width: 26), label: Categories),
      BottomNavigationBarItem(
          icon: Image.asset(icCart, width: 26), label: cart),
      BottomNavigationBarItem(
          icon: Image.asset(icProfile, width: 26), label: Account)
    ];

    var navBody = [
      HomeScreen(),
      categoriesScreen(),
      cartScreen(),
      profileScreen()
    ];

    return WillPopScope(
      onWillPop: () async {
        showDialog(
          barrierDismissible: false,

```

```
        context: context,
        builder: (context) => exitDialog(context));
    return false;
},
child: Scaffold(
    body: Column(
        children: [
            Obx(() => Expanded(
                child: navBody.elementAt(controller.currentNavIndex.value))),
        ],
),
bottomNavigationBar: Obx(
() => BottomNavigationBar(
    currentIndex: controller.currentNavIndex.value,
    selectedItemColor: green,
    selectedLabelStyle: const TextStyle(fontFamily: semibold),
    type: BottomNavigationBarType.fixed,
    backgroundColor: whiteColor,
    items: navbaritem,
    onTap: (Value) {
        controller.currentNavIndex.value = Value;
    },
),
),
),
),
);
}
}
```

## lib\views\screen\Wishlistscreen\wishlist\_screen.dart

```
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:farmer/consts/consts.dart';
import 'package:farmer/services/firestore_services.dart';
import 'package:farmer/widest_common/loading_indicator.dart';
import 'package:flutter/src/widgets/framework.dart';
import 'package:flutter/src/widgets/placeholder.dart';

class wishlistScreen extends StatelessWidget {
    const wishlistScreen({super.key});

    @override
    Widget build(BuildContext context) {
        return Scaffold(
            backgroundColor: whiteColor,
            appBar: AppBar(
                title: "পছন্দের তালিকার পন্য"
                    .text
                    .color(darkFontGrey)
                    .fontFamily(semibold)
```

```
        .make(),
    ),
body: StreamBuilder(
    stream: FireStoreServices.getWishlist(),
    builder:
        (BuildContext context, AsyncSnapshot<QuerySnapshot> snapshot) {
    if (!snapshot.hasData) {
        return Center(
            child: loadingIndicator(),
        );
    } else if (snapshot.data!.docs.isEmpty) {
        return "পছন্দের তালিকায় পণ্য যুক্ত করুন!".
            .text
            .color(darkFontGrey)
            .makeCentered();
    } else {
        var data = snapshot.data!.docs;

        return Column(
            children: [
                Expanded(
                    child: ListView.builder(
                        shrinkWrap: true,
                        itemCount: data.length,
                        itemBuilder: (BuildContext context, int index) {
                            return ListTile(
                                leading: Image.network(
                                    "${data[index]['p_imgs'][0]}",
                                    width: 120,
                                    fit: BoxFit.cover,
                                ),
                                title:
                                    "${data[index]['p_name']} (q=${data[index]
['p_quantity']})"
                                    .text
                                    .fontFamily(semibold)
                                    .size(16)
                                    .make(),
                                subtitle: "${data[index]['p_price']}"
                                    .numCurrency
                                    .text
                                    .color(redColor)
                                    .fontFamily(semibold)
                                    .make(),
                                trailing:
                                    const Icon(Icons.favorite, color: redColor)
                                    .onTap(() async {
                                        await firestore
                                            .collection(productCollection)
                                            .doc(data[index].id)
                                            .set({
                                                'p_wishlist':
                                                    FieldValue.arrayRemove([currentUser!.uid])
                                            })
                                    });
                            }
                        );
                    );
                }
            );
        );
    }
});
```

```
        },
        SetOptions(merge: true));
    },
}),
),
),
],
);
}
),
);
}
);
}
}
}
```

## lib\views\screen\component\featurebutton.dart

```
import 'package:farmer/consts/consts.dart';

Widget featureButton({String? title, icon}) {
    return Row(
        children: [
            Image.asset(icon, width: 100, fit: BoxFit.fill),
            10.widthBox,
            title!.text.fontFamily(semibold).color(darkFontGrey).make(),
        ],
    )
        .box
        .width(250)
        .margin(EdgeInsets.symmetric(horizontal: 8))
        .white
        .padding(const EdgeInsets.all(8))
        .roundedSM
        .outerShadow
        .make();
}
```

## lib\views\screen\component\order\_status.dart

```
import 'package:farmer/consts/consts.dart';

Widget orderstatus({icon, color, title, showDone}) {
    return ListTile(
        leading: Icon(
            icon,
            color: color,
        ).box.border(color: color).roundedSM.padding(EdgeInsets.all(8)).make(),
        trailing: SizedBox(
            height: 100,
```

```

        width: 120,
        child: Row(
            mainAxisAlignment: MainAxisAlignment.spaceAround,
            children: [
                "$title".text.color(darkFontGrey).make(),
                showDone
                    ? const Icon(
                        Icons.done,
                        color: redColor,
                    )
                    : Container(),
            ],
        ),
    ),
),
);
}
}

```

## lib\views\screen\homeScreen.dart

```

import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:farmer/consts/consts.dart';
import 'package:farmer/consts/list.dart';
import 'package:farmer/controller/home_controller.dart';
import 'package:farmer/ctegori/items_details.dart';
import 'package:farmer/services/firestore_services.dart';
import 'package:farmer/views/screen/component/featurebutton.dart';
import 'package:farmer/views/screen/searchScreen.dart';
import 'package:farmer/widest_common/homebutton.dart';
import 'package:farmer/widest_common/loading_indicator.dart';
import 'package:flutter/src/widgets/framework.dart';
import 'package:flutter/src/widgets/placeholder.dart';
import 'package:get/get.dart';

import '../../../../../widest_common/whetherbutton.dart';

class HomeScreen extends StatelessWidget {
    const HomeScreen({super.key});

    @override
    Widget build(BuildContext context) {
        var controller = Get.find<HomeController>();
        return Container(
            padding: const EdgeInsets.all(12),
            color: Color.fromARGB(255, 23, 182, 161),
            width: context.screenWidth,
            height: context.screenHeight,
            child: SafeArea(
                child: Column(
                    children: [
                        //search bar

```

```
Container(
    color: lightGrey,
    child: TextFormField(
        controller: controller.searchController,
        decoration: InputDecoration(
            border: InputBorder.none,
            suffixIcon: Icon(Icons.search).onTap(() {
                if (controller.searchController.text.isNotEmptyAndNotNull) {
                    Get.to(() => SearchScreem(
                        title: controller.searchController.text,
                    )));
                }
            }),
            filled: true,
            fillColor: whiteColor,
            hintText: searchanything,
            hintStyle: TextStyle(color: textfieldGrey),
        ),
    ),
),
),
),
10.heightBox,
Expanded(
    child: SingleChildScrollView(
        physics: BouncingScrollPhysics(),
        child: Column(
            children: [
                //swiper
                VxSwiper.builder(
                    aspectRatio: 16 / 9,
                    autoPlay: true,
                    height: 200,
                    itemCount: cropslist.length,
                    itemBuilder: (context, index) {
                        return Container(
                            child: Image.asset(
                                cropslist[index],
                                fit: BoxFit.fitWidth,
                            )
                        .box
                        .rounded
                        .clip(Clip.antiAlias)
                        .margin(EdgeInsets.symmetric(horizontal: 8))
                        .make(),
                    );
                }),
                10.heightBox,
                10.heightBox,
                Row(
                    mainAxisAlignment: MainAxisAlignment.spaceEvenly,
                    children: List.generate(
                        1,
                        (index) => homeButton(

```

```
        width: context.screenWidth / 2.5,
        height: context.screenHeight * 0.1,
        icon: icFlashDeal,
        title: whether,
    )),
).onTap(() {
    Get.offAll(() => MyApp());
}),
/* Row(
    mainAxisAlignment: MainAxisAlignment.spaceEvenly,
    children: List.generate(
        2,
        (index) => homeButton(
            width: context.screenWidth / 2.5,
            height: context.screenHeight * 0.1,
            icon: index == 0 ? icTodaysDeal : icFlashDeal,
            title: index == 0 ? todaydeal : FlashSell,
        )),
),*/
20.heightBox,
VxSwiper.builder(
    aspectRatio: 16 / 9,
    autoPlay: true,
    height: 200,
    itemCount: Sliderlist.length,
    itemBuilder: (context, index) {
        return Container(
            child: Image.asset(
                Sliderlist[index],
                fit: BoxFit.fitWidth,
            )
            .box
            .rounded
            .clip(Clip.antiAlias)
            .margin(EdgeInsets.symmetric(horizontal: 8))
            .make(),
        );
    }),
50.heightBox,
Align(
    alignment: Alignment.centerLeft,
    child:
morefeture.text.white.size(25).fontFamily(bold).make(),
),
20.heightBox,
SingleChildScrollView(
    scrollDirection: Axis.horizontal,
    child: Row(
        children: List.generate(
            3,
```





```

        autoPlay: true,
        height: 200,
        itemCount: cropslist.length,
        itemBuilder: (context, index) {
            return Container(
                child: Image.asset(
                    cropslist[index],
                    fit: BoxFit.fitWidth,
                )
            .box
            .rounded
            .clip(Clip.antiAlias)
            .margin(EdgeInsets.symmetric(horizontal: 8))
            .make(),
        );
    )));
}

//all product section

20.heightBox,
"ପନ୍ୟ ସମ୍ମୁଦ୍ର".text.size(25).white.fontFamily(bold).make(),
30.heightBox,
StreamBuilder(
    stream: FireStoreServices.allproduct(),
    builder: (BuildContext context,
        AsyncSnapshot<QuerySnapshot> snapshot) {
        if (!snapshot.hasData) {
            return loadingIndicator();
        } else {
            var allproductdata = snapshot.data!.docs;
            return GridView.builder(
                physics: const NeverScrollableScrollPhysics(),
                shrinkWrap: true,
                itemCount: allproductdata.length,
                gridDelegate:
                    const
                    SliverGridDelegateWithFixedCrossAxisCount(
                        crossAxisCount: 2,
                        mainAxisSpacing: 8,
                        crossAxisSpacing: 8,
                        mainAxisExtent: 300),
                itemBuilder: (context, index) {
                    return Column(
                        crossAxisAlignment:
CrossAxisAlignment.start,
                        children: [
                            Image.network(
                                allproductdata[index]['p_imgs'][0],
                                height: 200,
                                width: 200,
                                fit: BoxFit.cover),
                    10.heightBox,
                    "${allproductdata[index]['p_name']}"
                    .text

```

```
        .fontFamily(semibold)
        .color(darkFontGrey)
        .make(),
    const Spacer(),
    "${allproductdata[index]['p_price']}"
        .numCurrency
        .text
        .color(redColor)
        .fontFamily(bold)
        .size(16)
        .make(),
    ],
)
    .box
    .white
    .margin(const EdgeInsets.symmetric(
        horizontal: 6))
    .rounded
    .padding(EdgeInsets.all(12))
    .make()
    .onTap(() {
Get.to(() => itemdetails(
        title:
            "${allproductdata[index]
['p_name']}",
        data: allproductdata[index],
));
});
})
),
],
),
),
),
),
),
],
),
));
);
}
}
```

## lib\views\scren\loginscreen.dart

```
import 'package:farmer/consts/consts.dart';
import 'package:farmer/controller/auth_controller.dart';
import 'package:farmer/views/scren/Home.dart';
import 'package:farmer/views/scren/signupscren.dart';
import 'package:farmer/widest_common/applogo_wid.dart';
import 'package:farmer/widest_common/bg_wid.dart';
```

```
import 'package:farmer/widest_common/custom_textfield.dart';
import 'package:farmer/widest_common/our_button.dart';
import 'package:farmer/widest_common/re_button.dart';
import 'package:flutter/src/widgets/framework.dart';
import 'package:flutter/src/widgets/placeholder.dart';
import 'package:get/get.dart';
import 'package:get/get_core/src/get_main.dart';

class LoginScreen extends StatelessWidget {
  const LoginScreen({super.key});

  @override
  Widget build(BuildContext context) {
    var controller = Get.put(AuthController());

    return bgwidget(
      child: Scaffold(
        body: Center(
          child: Column(
            children: [
              (context.screenHeight * 0.07).heightBox,
              applogoWidget(),
              10.heightBox,
              "ଲଗ ଇନ୍ କରନ୍ ଓହ
$appname".text.fontFamily(bold).white.size(20).make(),
              10.heightBox,
              Obx(
                () => Column(
                  children: [
                    customtextfield(
                      hint: emailhint,
                      title: email,
                      isPass: false,
                      controller: controller.emailController),
                    customtextfield(
                      hint: passwordhint,
                      title: password,
                      isPass: true,
                      controller: controller.passwordController),
                    Align(
                      alignment: Alignment.centerRight,
                      child: TextButton(
                        onPressed: () {}, child:
forgetpassword.text.make()))),
              5.heightBox,
              controller.isloading.value
                ? const CircularProgressIndicator(
                    valueColor: AlwaysStoppedAnimation(redColor),
                  )
                : ourbutton(
                    onPressed: () async {
                      await controller
                        .loginMethod(context: context)
                        .then((value) {

```

lib\views\screen\order\_screen\order\_details.dart

```
import 'package:farmer/consts/consts.dart';
import 'package:farmer/views/screen/component/order_status.dart';
import 'package:farmer/views/screen/order_screen/order_place_details.dart';
import 'package:flutter/src/widgets/framework.dart';
import 'package:flutter/src/widgets/placeholder.dart';
import 'package:intl/intl.dart' as intl;

class ordersdetails extends StatelessWidget {
    final dynamic data;
    const ordersdetails({super.key, this.data});
```

```
@override
Widget build(BuildContext context) {
    return Scaffold(
        backgroundColor: whiteColor,
        appBar: AppBar(
            title: "অডার".text.fontFamily(semibold).make(),
        ),
        body: Padding(
            padding: const EdgeInsets.all(8.0),
            child: SingleChildScrollView(
                physics: const BouncingScrollPhysics(),
                child: Column(
                    children: [
                        orderstatus(
                            color: redColor,
                            icon: Icons.done,
                            title: "অডার",
                            showDone: data['order_placed'],
                        ),
                        orderstatus(
                            color: Colors.blue,
                            icon: Icons.thumb_up,
                            title: "নিশ্চিতকরণ",
                            showDone: data['order_confirmed'],
                        ),
                        orderstatus(
                            color: Colors.yellow,
                            icon: Icons.delivery_dining,
                            title: "বিতরণ",
                            showDone: data['order_on_delivery'],
                        ),
                        orderstatus(
                            color: green,
                            icon: Icons.done_all,
                            title: "বিতরণ করা হয়েছে",
                            showDone: data['order_delivered'],
                        ),
                        const Divider(),
                        20.heightBox,
                        orderPlaceDetails(
                            d1: data['order_code'],
                            d2: data['shipping_method'],
                            title1: "অডার কোড",
                            title2: "পরিবহন মাধ্যম"),
                        orderPlaceDetails(
                            d1: intl.DateFormat("h:mma")
                                .add_yMd()
                                .format(data['order_date'].toDate()),
                            d2: data['payment_method'],
                            title1: "অডার তারিখ",
                            title2: "মূল্যপরিশোধ পদ্ধতি",
                        ),
                        orderPlaceDetails(
```

```
d1: "অবৈতনিক",
d2: "অড়ার স্থাপন",
title1: "লেনদেনের অবস্থা",
title2: "বিতরণ অবস্থা",
),
Padding(
padding:
  const EdgeInsets.symmetric(horizontal: 16, vertical: 8),
child: Row(
  mainAxisAlignment: MainAxisAlignment.spaceBetween,
  children: [
    Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        "ক্রেতার ঠিকানা:"
        .text
        .size(18)
        .color(redColor)
        .fontFamily(bold)
        .make(),
        "${data['order_by_name']}".text.make(),
        "${data['order_by_email']}".text.make(),
        "${data['order_by_address']}".text.make(),
        "${data['order_by_district']}".text.make(),
        "${data['order_by_sub_district']}".text.make(),
        "${data['order_by_postal_code']}".text.make(),
        "${data['order_by_phone']}".text.make()
      ],
),
Column(
  children: [
    "মোট টাকাস্বত্ত্ব"
    .text
    .size(18)
    .color(redColor)
    .fontFamily(bold)
    .make(),
    "${data['total_amount']}"
    .numCurrency
    .text
    .color(redColor)
    .fontFamily(bold)
    .make(),
  ],
),
],
),
],
),
),
const Divider(),
20.heightBox,
"পন্য"
.text
.color(darkFontGrey)
```

```
        .fontFamily(bold)
        .size(20)
        .makeCentered(),
10.heightBox,
ListView(
    physics: const NeverScrollableScrollPhysics(),
    shrinkWrap: true,
    children: List.generate(data['oders'].length, (index) {
        return orderPlaceDetails(
            title1: data['oders'][index]['title'],
            title2: data['oders'][index]['qty'],
            d1:
".....",
            d2: ".....",
        );
    }).toList(),
).box.color(Color.fromARGB(57, 126, 71, 71)).make(),
20.heightBox,
],
),
),
),
),
);
}
}
```

## lib\views\screen\order\_screen\order\_place\_details.dart

```
import 'package:farmer/consts/consts.dart';

Widget orderPlaceDetails({title1, title2, d1, d2}) {
    return Padding(
        padding: const EdgeInsets.symmetric(horizontal: 16.0, vertical: 8),
        child: Row(
            mainAxisAlignment: MainAxisAlignment.spaceBetween,
            children: [
                Column(
                    crossAxisAlignment: CrossAxisAlignment.start,
                    children: [
                        "$title1".text.fontFamily(semibold).make(),
                        "$d1".text.color(redColor).fontFamily(semibold).make(),
                    ],
                ),
                Column(
                    crossAxisAlignment: CrossAxisAlignment.end,
                    children: [
                        "$title2".text.fontFamily(semibold).make(),
                        "$d2".text.fontFamily(semibold).make(),
                    ],
                ),
            ],
        ),
    );
}
```

```
        ),
        ],
    ),
);
}
```

## lib\views\scren\order\_screen\order\_screen.dart

```
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:farmer/consts/consts.dart';
import 'package:farmer/services/firestore_services.dart';
import 'package:farmer/views/scren/order_screen/order_details.dart';
import 'package:farmer/widest_common/loading_indicator.dart';
import 'package:flutter/src/widgets/framework.dart';
import 'package:flutter/src/widgets/placeholder.dart';
import 'package:get/get.dart';

class OrderScreeen extends StatelessWidget {
    const OrderScreeen({super.key});

    @override
    Widget build(BuildContext context) {
        return Scaffold(
            backgroundColor: whiteColor,
            appBar: AppBar(
                title: "আমার
অডার".text.color(darkFontGrey).fontFamily(semibold).make(),
            ),
            body: StreamBuilder(
                stream: FireStoreServices.getallOrders(),
                builder:
                    (BuildContext context, AsyncSnapshot<QuerySnapshot> snapshot) {
                if (!snapshot.hasData) {
                    return Center(
                        child: loadingindicator(),
                    );
                } else if (snapshot.data!.docs.isEmpty) {
                    return "আপনার কোনো অডার নেই!!"
                        .text
                        .color(darkFontGrey)
                        .makeCentered();
                } else {
                    var data = snapshot.data!.docs;

                    return ListView.builder(
                        itemCount: data.length,
                        itemBuilder: (BuildContext context, int index) {
                            return ListTile(
                                leading: "${index + 1}"
                                    .text
                                    .color(darkFontGrey)
                                    .fontFamily(semibold).make()
                            );
                        }
                    );
                }
            }
        );
    }
}
```

```

        .fontFamily(bold)
        .color(darkFontGrey)
        .xl
        .make(),
      title: data[index]['order_code']
        .toString()
        .text
        .color(redColor)
        .fontFamily(semibold)
        .make(),
      subtitle: data[index]['total_amount']
        .toString()
        .numCurrency
        .text
        .fontFamily(bold)
        .make(),
      trailing: IconButton(
        onPressed: () {
          Get.to(ordersdetails(data: data[index]));
        },
        icon: Icon(Icons.arrow_forward_ios_rounded,
          color: darkFontGrey)),
      );
    });
  });
}

```

## lib\views\scrren\searchScreen.dart

```

import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:farmer/consts/consts.dart';
import 'package:farmer/services/firestore_services.dart';
import 'package:farmer/widest_common/loading_indicator.dart';
import 'package:flutter/src/widgets/framework.dart';
import 'package:flutter/src/widgets/placeholder.dart';
import 'package:get/get.dart';
import 'package:get/get_core/src/get_main.dart';

import '../../../../../ctegori/items_details.dart';

class SearchScreem extends StatelessWidget {
  final String? title;
  const SearchScreem({super.key, this.title});

  @override
  Widget build(BuildContext context) {

```

```

return Scaffold(
  backgroundColor: whiteColor,
  appBar: AppBar(
    title: title!.text.color(darkFontGrey).make(),
  ),
  body: FutureBuilder(
    future: FireStoreServices.searchproduct(title),
    builder:
      (BuildContext context, AsyncSnapshot<QuerySnapshot> snapshot) {
    if (!snapshot.hasData) {
      return Center(
        child: loadingIndicator(),
      );
    } else if (snapshot.data!.docs.isEmpty) {
      return "পন্যটি খুজে পাওয়া যায় নি ! আবার সার্চ করুন "
        .text
        .color(redColor)
        .fontFamily(bold)
        .makeCentered();
    } else {
      var data = snapshot.data!.docs;
      var filter = data
        .where(
          (element) => element['p_name']
            .toString()
            .toLowerCase()
            .contains(title!.toLowerCase()),
        )
        .toList();

      return Padding(
        padding: const EdgeInsets.all(8.0),
        child: GridView(
          gridDelegate: const SliverGridDelegateWithFixedCrossAxisCoun
t(
            crossAxisCount: 2,
            mainAxisSpacing: 8,
            crossAxisSpacing: 8,
            mainAxisExtent: 300),
            children: filter
              .mapIndexed(
                (currentValue, index) => Column(
                  crossAxisAlignment: CrossAxisAlignment.start,
                  children: [
                    Image.network(filter[index]['p_imgs'][0],
                      height: 200, width: 200, fit: BoxFit.cover),
                    10.heightBox,
                    "${filter[index]['p_name']}"
                      .text
                      .fontFamily(semibold)
                      .color(darkFontGrey)
                      .make(),
                    const Spacer(),

```

```
        "${filter[index]['p_price']}"}  
        .numCurrency  
        .text  
        .color(redColor)  
        .fontFamily(bold)  
        .size(16)  
        .make(),  
    ],  
)  
    .box  
    .white  
    .margin(const EdgeInsets.symmetric(horizontal: 6))  
    .rounded  
    .padding(EdgeInsets.all(12))  
    .make()  
    .onTap(() {  
        Get.to(() => itemdetails(  
            title: "${filter[index]['p_name']}",  
            data: filter[index],  
        ));  
    }),  
)  
    .toList(),  
),  
);  
}  
});  
};  
}
```

## lib\views\screen\signupscren.dart

```
import 'package:farmer/consts/consts.dart';
import 'package:farmer/controller/auth_controller.dart';
import 'package:farmer/views/scren/signupscren.dart';
import 'package:farmer/widest_common/applogo_wid.dart';
import 'package:farmer/widest_common/bg_wid.dart';
import 'package:farmer/widest_common/custom_testfield.dart';
import 'package:farmer/widest_common/our_button.dart';
import 'package:farmer/widest_common/re_button.dart';
import 'package:flutter/src/widgets/framework.dart';
import 'package:flutter/src/widgets/placeholder.dart';
import 'package:get/get.dart';
import 'package:get/get_core/src/get_main.dart';

import 'Home.dart';

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

```
@override
State<SignupScreen> createState() => _SignupScreenState();
}

class _SignupScreenState extends State<SignupScreen> {
  bool? isChecked = false;
  var controller = Get.put(AuthController());

  //text controller
  var namecontroller = TextEditingController();
  var emailcontroller = TextEditingController();
  var passwordcontroller = TextEditingController();
  var passwordRetypecontroller = TextEditingController();

  @override
  Widget build(BuildContext context) {
    return bgwidget(
      child: Scaffold(
        body: Center(
          child: Column(
            children: [
              (context.screenHeight * 0.07).heightBox,
              applogoWidget(),
              10.heightBox,
              "সাইন আপ করুনঃ $appname"
                .text
                .fontFamily(bold)
                .white
                .size(20)
                .make(),
              10.heightBox,
              Obx(
                () => Column(
                  children: [
                    customtextfield(
                      hint: namehint,
                      title: name,
                      controller: namecontroller,
                      isPass: false),
                    customtextfield(
                      hint: emailhint,
                      title: email,
                      controller: emailcontroller,
                      isPass: false),
                    customtextfield(
                      hint: passwordhint,
                      title: password,
                      controller: passwordcontroller,
                      isPass: true),
                    customtextfield(
                      hint: repasswordhint,
                      title: repassword,
```

```
controller: passwordRetypecontroller,
isPass: true),
```

```
Row(
  children: [
    Checkbox(
      checkColor: redColor,
      value: isCheck,
      onChanged: (newValue) {
        setState(() {
          isCheck = newValue;
        });
      },
    ),
    10.heightBox,
    RichText(
      text: const TextSpan(
        children: [
          TextSpan(
            text: "আমি শর্ত গুলো মেনে নিয়েছি",
            style: TextStyle(
              fontFamily: bold,
              color: fontGrey,
            )),
          TextSpan(
            text: termsandcondition,
            style: TextStyle(
              fontFamily: bold,
              color: redColor,
            )),
        ],
      )));
  ],
),
5.heightBox,
controller.isloading.value
? const CircularProgressIndicator(
  valueColor: AlwaysStoppedAnimation(redColor),
)
: myButton(onPress: () async {
  try {
    // controller.isloading(true);
    await controller
      .signupMethod(
        context: context,
        email: emailcontroller.text,
        password: passwordcontroller.text)
      .then((value) {
    return controller.storeusedata(
      email: emailcontroller.text,
      password: passwordcontroller.text,
      name: namecontroller.text);
  }).then((value) {
```

```
        VxToast.show(context, msg: signsuccess);
        Get.offAll(() => const Home());
    });
} catch (e) {
    auth.signOut();
    // VxToast.show(context, msg: e.toString());
}
}).box.width(context.screenWidth - 50).make(),
10.heightBox,
RichText(
    text: const TextSpan(
        children: [
            TextSpan(
                text: alreadyhaveanaccount,
                style: TextStyle(
                    fontFamily: bold,
                    color: fontGrey,
                )), TextSpan(
                text: login,
                style: TextStyle(
                    fontFamily: bold,
                    color: redColor,
                )), ],
    ),),
).onTap(() {
    Get.back();
}),
//ourbutton(color: redColor, title: Login, textcolor:
whiteColor)
// .box
// .width(context.screenWidth - 50)
// .make(),

5.heightBox,
// myButton(onPress: () {
// Get.to(() => SignupScreen());
// }), ],
)
.box
.white
.rounded
.padding(const EdgeInsets.all(16))
.width(context.screenWidth - 70)
.make(),
),
],
),
),
));
}
```

```
}
```

## lib\views\screen\splash\_screen.dart

```
import 'dart:async';
import 'dart:typed_data';

import 'package:farmer/consts/colors.dart';
import 'package:farmer/consts/consts.dart';
import 'package:farmer/views/screen/Home.dart';
import 'package:farmer/views/screen/homeScreen.dart';
import 'package:farmer/views/screen/loginscreen.dart';
import 'package:farmer/widest_common/applogo_wid.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:flutter/src/widgets/framework.dart';
import 'package:flutter/src/widgets/placeholder.dart';
import 'package:get/get.dart';

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

    @override
    State<SplashScreen> createState() => _SplashScreenState();
}

class _SplashScreenState extends State<SplashScreen> {
    changeScreen() {
        Future.delayed(Duration(seconds: 2), () {
            //Get.to(() => const LoginScreen());
            auth.authStateChanges().listen((User? user) {
                if (user == null && mounted) {
                    Get.to(() => LoginScreen());
                } else {
                    Get.to(() => const Home());
                }
            });
        });
    }

    @override
    void initState() {
        changeScreen();
        super.initState();
    }

    @override
    Widget build(BuildContext context) {
        return Scaffold(
```

```
backgroundColor: green,
body: Center(
    child: Column(
        children: [
            Align(
                alignment: Alignment.topLeft,
                child: Image.asset(
                    icSplashBg,
                    width: 300,
                )),
            20.heightBox,
            appLogoWidget(),
            10.heightBox,
            appName.text.fontFamily(bold).size(22).white.make(),
            5.heightBox,
            appVersion.text.white.make(),
            const Spacer(),
            credits.text.white.fontFamily(semibold).make(),
            30.heightBox,
        ],
    ),
),
);
}
}
```

DOCUMENT\_HEADING

## 16 files

(file list disabled)

### lib\widest\_common\applogo\_wid.dart

```
import 'package:farmer/consts/consts.dart';

Widget applogoWidget() {
    return Image.asset(icAppLogo)
        .box
        .white
        .size(77, 77)
        .padding(EdgeInsets.all(8))
        .rounded
        .make();
}
```

### lib\widest\_common\bg\_wid.dart

```
import 'package:farmer/consts/consts.dart';

Widget bgwidget({Widget? child}) {
    return Container(
        decoration: const BoxDecoration(
            image: DecorationImage(
                image: AssetImage(imgBackground), fit: BoxFit.fill)),
        child: child,
    );
}
```

### lib\widest\_common\cart\_button.dart

```
import 'package:farmer/consts/consts.dart';

Widget cartbutton({onPress}) {
    return ElevatedButton(
        style: ElevatedButton.styleFrom(
            padding: const EdgeInsets.all(12),
        ),
        onPressed: onPress,
        child: "অডার করুন ".text.white.fontFamily(bold).make(),
    );
}
```

```
    );
}
```

## lib\widest\_common\custom\_testfield.dart

```
import 'package:farmer/consts/consts.dart';

Widget customtextfield({String? title, String? hint, controller, isPass}) {
    return Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
            title!.text.color(redColor).fontFamily(semibold).size(16).make(),
            5.heightBox,
            TextFormField(
                obscureText: isPass,
                controller: controller,
                decoration: InputDecoration(
                    hintStyle: TextStyle(
                        fontFamily: semibold,
                        color: textfieldGrey,
                    ),
                    hintText: hint,
                    isDense: true,
                    fillColor: lightGrey,
                    filled: true,
                    border: InputBorder.none,
                    focusedErrorBorder:
                        OutlineInputBorder(borderSide: BorderSide(color: redColor)),
                ),
            ),
            15.heightBox,
        ],
    );
}
```

## lib\widest\_common\exit\_button2.dart

```
import 'package:farmer/consts/consts.dart';

Widget exitbtnNo({onPress}) {
    return ElevatedButton(
        style: ElevatedButton.styleFrom(
            padding: const EdgeInsets.all(12),
        ),
        onPressed: onPress,
        child: "No".text.white.fontFamily(bold).make(),
    );
}
```

```
}
```

## lib\widest\_common\exit\_dialog.dart

```
import 'package:farmer/consts/consts.dart';
import 'package:farmer/widest_common/exit_button2.dart';
import 'package:farmer/widest_common/exitbutton.dart';
import 'package:flutter/services.dart';
import 'package:get/get.dart';

Widget exitDialog(context) {
    return Dialog(
        child: Column(
            mainAxisSize: MainAxisSize.min,
            children: [
                "conform".text.fontFamily(bold).size(18).color(darkFontGrey).make(),
                Divider(),
                10.heightBox,
                "Are you sure you want to exit?"
                    .text
                    .size(16)
                    .color(darkFontGrey)
                    .make(),
                10.heightBox,
                Row(
                    mainAxisAlignment: MainAxisAlignment.spaceEvenly,
                    children: [
                        exitbutton(onPress: () {
                            SystemNavigator.pop();
                        }),
                        exitbtnNo(onPress: () {
                            Navigator.pop(context);
                        })
                    ],
                ),
                ],
            ).box.color(lightGrey).padding(EdgeInsets.all(16)).roundedSM.make(),
    );
}
```

## lib\widest\_common\exitbutton.dart

```
import 'package:farmer/consts/consts.dart';

Widget exitbutton({onPress}) {
    return ElevatedButton(
        style: ElevatedButton.styleFrom(
```

```
        padding: const EdgeInsets.all(12),
    ),
    onPressed: onPress,
    child: "Yes".text.white.fontFamily(bold).make(),
);
}
```

## lib\widest\_common\homebutton.dart

```
import 'package:farmer/consts/colors.dart';
import 'package:farmer/consts/consts.dart';

Widget homeButton({width, height, icon, String? title, onPress}) {
    return Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
            Image.asset(
                icon,
                width: 28,
            ),
            5.heightBox,
            title!.text.fontFamily(semibold).color(darkFontGrey).make(),
        ],
    ).box.rounded.white.size(width, height).make();
}
```

## lib\widest\_common\loading\_indicator.dart

```
import 'package:farmer/consts/consts.dart';

Widget loadingindicator() {
    return const CircularProgressIndicator(
        valueColor: AlwaysStoppedAnimation(redColor),
    );
}
```

## lib\widest\_common\order\_button.dart

```
import 'package:farmer/consts/consts.dart';

Widget Continue({onPress}) {
    return ElevatedButton(
        style: ElevatedButton.styleFrom(

```

```
        padding: const EdgeInsets.all(12),
    ),
    onPressed: onPress,
    child: "CONTINUE".text.white.fontFamily(bold).make(),
);
}
```

## lib\widest\_common\ourEditbutton2.dart

```
import 'package:farmer/consts/consts.dart';

Widget ourEditbutton2({onPress}) {
    return ElevatedButton(
        style: ElevatedButton.styleFrom(
            backgroundColor: Color.fromARGB(218, 4, 80, 94),
            padding: const EdgeInsets.all(12),
        ),
        onPressed: onPress,
        child: "সংরক্ষণ করুন"
            .text
            .color(Color.fromARGB(255, 5, 3, 3))
            .fontFamily(bold)
            .make(),
    );
}
```

## lib\widest\_common\ourEditscreenButton.dart

```
import 'package:farmer/consts/consts.dart';

Widget ourEditbutton({onPress}) {
    return ElevatedButton(
        style: ElevatedButton.styleFrom(
            backgroundColor: Color.fromARGB(255, 22, 148, 11),
            padding: const EdgeInsets.all(12),
        ),
        onPressed: onPress,
        child: "পরিবর্তন করুন"
            .text
            .color(Color.fromARGB(255, 0, 0, 0))
            .fontFamily(bold)
            .make(),
    );
}
```

## **lib\widest\_common\our\_button.dart**

```
import 'package:farmer/consts/consts.dart';

Widget ourbutton({onPress}) {
    return ElevatedButton(
        style: ElevatedButton.styleFrom(
            padding: const EdgeInsets.all(12),
        ),
        onPressed: onPress,
        child: login.text.white.fontFamily(bold).make(),
    );
}
```

## **lib\widest\_common\place\_order\_button.dart**

```
import 'package:farmer/consts/consts.dart';

Widget procedOrder({onPress}) {
    return ElevatedButton(
        style: ElevatedButton.styleFrom(
            padding: const EdgeInsets.all(12),
        ),
        onPressed: onPress,
        child: "অডার নিশ্চিত করুন".text.white.fontFamily(bold).make(),
    );
}
```

## **lib\widest\_common\re\_button.dart**

```
import 'package:farmer/consts/consts.dart';
import 'package:farmer/consts/strings.dart';

Widget myButton({onPress}) {
    return ElevatedButton(
        style: ElevatedButton.styleFrom(padding: EdgeInsets.all(12)),
        onPressed: onPress,
        child: signup.text.black.fontFamily(bold).make(),
    );
}
```

## **lib\widest\_common\whetherbutton.dart**

```
import 'dart:ffi';

import 'package:farmer consts/consts.dart';
import 'package:farmer/controller/home_controller.dart';
import 'package:flutter/material.dart';
import 'package:get/get.dart';

void main() => runApp(new MyApp());

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return new MaterialApp(
      title: "Calculator App",
      theme: new ThemeData(primarySwatch: Colors.red),
      home: new HomePage(),
    );
  }
}

class HomePage extends StatefulWidget {
  @override
  State createState() => new HomePageState();
}

class HomePageState extends State<HomePage> {
  var num1 = 0, num2 = 0;
  double sum = 0;

  final TextEditingController t1 = new TextEditingController(text: "0");
  final TextEditingController t2 = new TextEditingController(text: "0");

  void doAddition() {
    setState(() {
      num1 = int.parse(t1.text);
      num2 = int.parse(t2.text);
      double num3 = num1.toDouble();
      sum = num3 * 0.8;
    });
  }

  //জৈব স্যার

  void doSub() {
    setState(() {
      num1 = int.parse(t1.text);
      double num3 = num1.toDouble();
      sum = num3 * 2.2;
    });
  }

  //জিন্মাম
```

```

void doMul() {
    setState(() {
        num1 = int.parse(t1.text);
        double num3 = num1.toDouble();
        sum = num3 * 0.454;
    });
}

//দ্বা
void doDiv() {
    setState(() {
        num1 = int.parse(t1.text);
        double num3 = num1.toDouble();
        sum = num3 * 0.045;
    });
}

void Tsp() {
    setState(() {
        num1 = int.parse(t1.text);
        double num3 = num1.toDouble();
        sum = num3 * 0.363;
    });
}

void Mop() {
    setState(() {
        num1 = int.parse(t1.text);
        double num3 = num1.toDouble();
        sum = num3 * 0.8;
    });
}

void doClear() {
    setState(() {
        t1.text = "0";
        t2.text = "0";
    });
}

@Override
Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
            title: Text("ক্যালকুলেটর"),
        ),
        body: new Container(
            padding: const EdgeInsets.all(40.0),
            child: new Column(
                mainAxisAlignment: MainAxisAlignment.center,
                children: <Widget>[
                    new Text(
                        "সারের পরিমান :$sum কেজি",
                    )
                ],
            ),
        ),
    );
}

```

```

        style: new TextStyle(
            fontSize: 20.0,
            fontWeight: FontWeight.bold,
            color: Colors.purple),
    ),
    new TextField(
        keyboardType: TextInputType.number,
        decoration:
            new InputDecoration(hintText: "জমির পরিমাণ শতক অনুযায়ী"),
        controller: t1,
    ),
    // new TextField(
    //   keyboardType: TextInputType.number,
    //   decoration: new InputDecoration(hintText: "Enter Number 2"),
    //   controller: t2,
    // ),
    new Padding(
        padding: const EdgeInsets.only(top: 20.0),
    ),
    new Row(
        mainAxisAlignment: MainAxisAlignment.spaceEvenly,
        children: <Widget>[
            new MaterialButton(
                child: new Text("ইউরিয়া"),
                color: Color.fromRGBO(105, 240, 174, 1),
                onPressed: doAddition,
            ),
            new MaterialButton(
                child: new Text("জৈব সার"),
                color: Colors.greenAccent,
                onPressed: doSub,
            ),
        ],
    ),
    new Padding(
        padding: const EdgeInsets.only(top: 20.0),
    ),
    new Row(
        mainAxisAlignment: MainAxisAlignment.spaceEvenly,
        children: <Widget>[
            new MaterialButton(
                child: new Text("জিপসাপ"),
                color: Colors.greenAccent,
                onPressed: doMul,
            ),
            new MaterialButton(
                child: new Text("দণ্ড"),
                color: Colors.greenAccent,
                onPressed: doDiv,
            ),
        ],
    ),
    20.heightBox,

```

```
new Row(
    mainAxisAlignment: MainAxisAlignment.spaceEvenly,
    children: <Widget>[
        new MaterialButton(
            child: new Text("টি এস পি "),
            color: Colors.greenAccent,
            onPressed: Tsp,
        ),
        new MaterialButton(
            child: new Text("ডি এ পি "),
            color: Colors.greenAccent,
            onPressed: Mop,
        ),
    ],
),
new Padding(
    padding: const EdgeInsets.only(top: 20.0),
),
new Row(
    mainAxisAlignment: MainAxisAlignment.center,
    children: <Widget>[
        new MaterialButton(
            child: new Text("পুনরায় হিসাব করুন"),
            color: Colors.greenAccent,
            onPressed: doClear,
        ),
    ],
),
),
),
),
);
}
}
```

## lib\main.dart

```
import 'package:farmer/consts/consts.dart';
import 'package:farmer/views/screen/splash_screen.dart';
import 'package:firebase_core/firebase_core.dart';
import 'package:flutter/material.dart';
import 'package:get/get_navigation/src/root/get_material_app.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:url_launcher/url_launcher.dart';

void main() async {
    WidgetsFlutterBinding.ensureInitialized();
    await Firebase.initializeApp();
    runApp(const MyApp());
}

class MyApp extends StatelessWidget {
    const MyApp({super.key});

    // This widget is the root of your application.
    @override
    Widget build(BuildContext context) {
        return GetMaterialApp(
            debugShowCheckedModeBanner: false,
            title: 'Flutter Demo',
            theme: ThemeData(
                scaffoldBackgroundColor: Colors.transparent,
                appBarTheme: AppBarTheme(backgroundColor: Colors.transparent),
                fontFamily: regular),
            home: const SplashScreen(),
        );
    }
}
```