A Project Report On

**MUMBAI CHI MASOLI**

Submitted by

|  |  |
| --- | --- |
| **Tanmay Wani** | **FS18CO040** |
| **Shreeraj Sangle** | **FS18CO032** |
| **Atharva Waghdhare** | **FS18CO033** |
| **Pratik Kadam** | **FS18CO037** |
| **Parth kadam** | **FS18CO038** |

**In partial fulfillment of the requirements for**

**Diploma in Computer Engineering**

**2020-2021**

Under the guidance of

**Prof. V. S. Lokhande**



**Department of Computer Engineering**

**Government Polytechnic, Mumbai**

**Mumbai-400051**



**Department of Computer Engineering**

**Government Polytechnic, Mumbai**

Mumbai-400051

CERTIFICATE

This is to certify that following students have successfully and satisfactorily completed the project on **MUMBAI CHI MASOLI** and presented their report in the partial fulfillment of requirement for Diploma in Computer Engineering from Government Polytechnic, Mumbai under the guidance of Prof. V. S. Lokhande in the academic year 2020-2021.

|  |  |
| --- | --- |
| **Tanmay Wani**  **Shreeraj Sangle** | **FS18CO040**  **FS18CO032** |
| **Atharva Waghdhare** | **FS18CO033** |
| **Pratik Kadam** | **FS18CO037** |
| **Parth Kadam** | **FS18CO038** |
|  |  |

|  |  |  |
| --- | --- | --- |
| Project Guide |  | External Examiner |
| Head Of the Department |  | Principal |

ACKNOWLEDGEMENT

This project required a lot of guidance and assistance from many people and we are extremely privileged to have got this all along the journey of our project. All that we have done is only due to such supervision and assistance and we would not forget to thank them.

We would first and foremost like to thank our Principal, Prof. Swati Deshpande, for supporting our project idea from the beginning and motivating us to work on our project without any worries.

We owe our deep gratitude to our Head Of Department, Mrs. V. S. Lokhande, who took keen interest on our project work and guided us all along, by providing all the necessary information for developing a good system.

We respect and thank Prof. V. S. Lokhande, for providing us an opportunity to do the project work in Government Polytechnic Mumbai and giving us all support and guidance which made us complete the project duly. We are extremely thankful to her for providing such a nice support and guidance.

We heartily thank Mr.Smit Pusalkar, owner of SP-Tech for their encouragement and more over for their timely support and guidance till the completion of our project work.

We are thankful to and fortunate enough to get constant encouragement, support and guidance from all Teaching staffs of Computer department, GPM which helped us in successfully progressing with our project work.

INDEX

|  |  |  |
| --- | --- | --- |
| **SR. NO.** | **CONTENTS** | **PAGE NO.** |
| **1** | **Introduction** | 6-7 |
| **2** | **Feasibilty Study** | 8-10 |
| **3** | **Project Planning** | 11-14 |
| **4** | **Requirement Analysis** | 15-16 |
| **5** | **System Design** | 17-21 |
| **6** | **Database Structure** | 22-23 |
| **7** | **Coding** | 24-128 |
| **8** | **Testing** | 129-146 |
| **9** | **Outputs/Screenshots** | 147-168 |
| **10** | **Bibliography** | 169-170 |

CHAPTER 1

Introduction

Introduction

## OBJECTIVES

The key object of current project was to allow the vendors to order the products more efficiently, at anytime from anywhere. This application will allow the user to sell the product more easily & will allow the vendors to browse & purchase the product. The general objective of this application was to build a platform for user to sell the products more reliably & for user to purchase that product easily.

Specific Objectives

1. User Friendly.
2. To lessen the measure of time to purchase the product.
3. To enhance the correspondence between the vendor and the user and improve the shopping experience.

CHAPTER 2

Feasibility Study

1. Feasibility Study

A feasibility study is an analysis that takes all of a project's relevant factors into account—including economic, technical, legal, and scheduling considerations— to ascertain the likelihood of completing the project successfully. Project managers use feasibility studies to discern the pros and cons of undertaking a project before they invest a lot of time and money into it.

**2.1 Go native:**

The best user experience can be achieved only when using the native development environment. Apps built using cross-platform development tools will not cut it.

**2.2 Remember form factors:**

Your app will be run on phones from different manufacturers, with different size of displays and resolutions, on portrait and landscape as well as on tablets and possibly readers. Don’t hard-code your screen layouts with a specific form factor or phone in mind.

**2.3 Support as many versions as possible:**

Android OS versions are coming out thick and fast, as are the phones. At any time, expect your target user base to be spread across at least 3 different OS versions.

**2.4 Be prepared for missing components:**

Not all devices have all the bells and whistles such as front cameras, accelerometers, and gyroscopes. Use them in your apps, by all means, but don’t depend on their presence either.

**2.5 Stick to the basics:**

Most people understand standard android app behavior – so follow the same standards. For example, if your app has a menu, it has to be accessible from the menu button.

**2.6 Be prepared for lack of connectivity:**

Connections on mobile phones will drop. Program defensively for such situations.

**2.7 No time-consuming processing in the UI thread:**

Use multi-threading wisely and avoid holding up the UI/main thread. Even a sub 2 second frozen screen is enough the turn a customer away Student Placement Application 5

**2.8 Don’t hog the resources:**

The Android phone is thin on resources and it has to be shared between all the apps. Plus, Android provides users with easy access to data such as how much memory and battery your app is consuming. As it is, most of the Android phones have to be recharged every day. If your app is seen to be draining a lot of the battery, it will be the first to be un-installed.

**2.9 Don’t force users to login every time:**

Typing is never easy, even with Swype or Swipe Pad installed. If the user has to login to access the app functionality every time, your app will never get used. Avoid login altogether if you can. If you can’t, then provide an explicit logout and leave it to the user to decide whether she wants to be permanently logged in or not. Remember, a phone is a personal device and most often not shared with anybody else. How many times have you logged into google mail from your Android? Imagine if you had to go through the login process every time!

CHAPTER 3

Project Planning

# Project Planning

Project planning is part of project management, which refers to use of schedules such as Gantt chart to plan and subsequently report progress within the project environment. Project planning can be done manually. The project plan clearly defines how the project is created, monitored and closed.

**SCOPE SCHEDULE AND COST**

Now we are going to discuss about what are the scope of the project, the size of project and in how much time it will be completed. For that we have to schedule the time and dates from which date it will start and the date it will be completed.

**SCHEDULING OF THE PROJECT GIVEN BELOW:**

|  |  |  |
| --- | --- | --- |
| **Date** | **Day** | **Subject** |
| 05/10/2020 | Monday | Introduction to Third Year project |
| 12/10/2020 | Monday | Group Selection |
| 24/10/2020 | Monday | Abstract Submission |
| 02/11/2020 | Monday | Guide Allocation |
| 03/11/2020 | Tuesday | Project selected |
| 12/11/2020 | Thursday | Virtual meeting with SP-Tech associates |
| 18/11/2020 | Wednesday | Designing of class diagrams and  work flow of project began |
| 03/12/2020 | Thursday | Face to face meeting with SP-Tech associates |
| 04/12/2020 | Friday | Designing of front-end began |
| 20/12/2020 | Sunday | Face to face meeting with SP-Tech associates, regarding database |
| 21/12/2020 | Monday | Working on the database began |
| 06/01/2021 | Wednesday | Working on android app modules began |
| 31/01/2021 | Sunday | Front-end/ user interface completedtill authentication model |
| 03/02/2021 | Wednesday | Working on android database structure started |
| 26/02/2021 | Friday | Database completed and successfully linked with front-end |
| 15/03/2021 | Monday | Android app completed halfway |
| 05/04/2021 | Monday | Android app database completed |
| 14/04/2021 | Wednesday | Integration of all android app modules and error solving |
| 12/05/2021 | Wednesday | Integrated android app modules and linked with database |
| 30/06/2021 | Wednesday | Fixed minor errors |

## COMMUNICATION, RESOURCES AND RISKS

Communication is key in project management. For a successful project execution, effective communication to all stakeholders is essential. Many projects fail because of a lack of communication or an ineffective one. Communication is best defined as the exchange of information and the expression of ideas, thoughts and feelings by using words and other methods. In the project management context this means the exchange of knowledge, skills and experience.

A resource is a necessary asset whose main role is to help carry out a certain task or project. A resource can be a person, a team, a tool, finances, and time. Most projects require many different resources in order to be completed. Resources should be assessed and allocated before a project begins. Poor resource planning can result in running out of resources midway through a project, delaying deadlines, and delivery of the final product or service.

Risk is the possibility of loss or injury. Project risk is an uncertain event or condition that, if it occurs, has an effect on at least one project objective. Risk management focuses on identifying and assessing the risks to the project and managing those risks to minimize the impact on the project. There are no risk-free projects because there are an infinite number of events that can have a negative effect on the project.

## CHANGE IN PLAN

A change management plan defines activities and roles to manage and control change during the execute and control stage of the project. Change is measured against the project baseline, which is a detailed description of the project’s scope, budget, schedule, and plans to manage quality, risk, issues, and change. During the execute and control stage, changes may require one or more revised project baselines to be issued.

CHAPTER 4

Requirement Analysis

# Requirement Analysis

**4.1. Hardware Components:**

The hardware used for the development of the project:

1. Processor: Intel i5 8th Gen

2. RAM: 8 GB, minimum 4 GB

3. Monitor:15 inches color

4. Keyboard: Optical

5. Mouse: Optical

**4.2. Software Requirements:**

The software used for the development of the project:

1. Operating system: Windows, Android

2. Programming Language: Dart

3. IDE: VS Code, Android studios

4. SDK: Flutter SDK

5. Emulators: AVD

6. Tools used: Android SDK Tools, Android Developer Tools

7. Database used: Firebase Firestore Database and Firebase Authentication

CHAPTER 5

System Design

# System Design

**System Features**

**Attractive Presentation:**

* The Menu is organized in an attractive way.
* There are images of every item which will make the view of customer clear about how the food will look like after delivery.
* Here is an attractive use of Various themes and color schemes.

**Customer Feedback:**

* Customer can enter the feedback about the product.
* This helps the owner to analyze the service and make necessary changes if needed.
* This also helps the Customer’s to decide a particular item with a positive feedback.

**Cart:**

* Customer can add & remove a particular product in cart.
* Customer can set quantity of product in cart.
* Customer can view total amount in cart.
* Customer can place order in cart.

**Wish list:**

* Customer can add a particular product in wish list.
* This saves a lot of time of customer to view saved products.

**Searching Item:**

* Customer can search a particular item according to name.
* This saves a lot of time of customer to order an item.

**User Profile:**

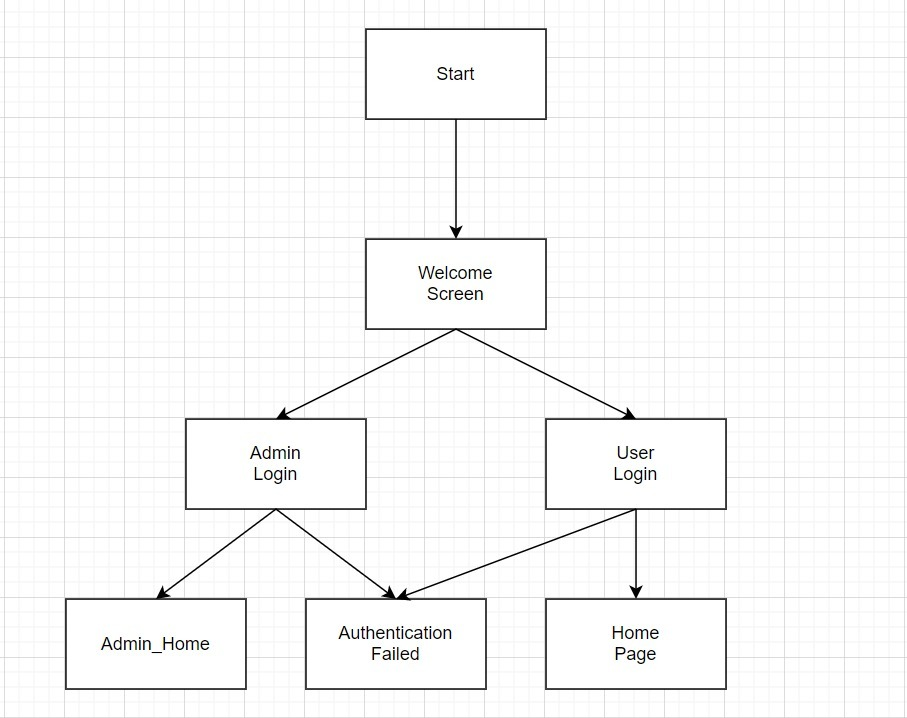
* Customer can view their profile.
* This saves a lot of time of customer to order an item.

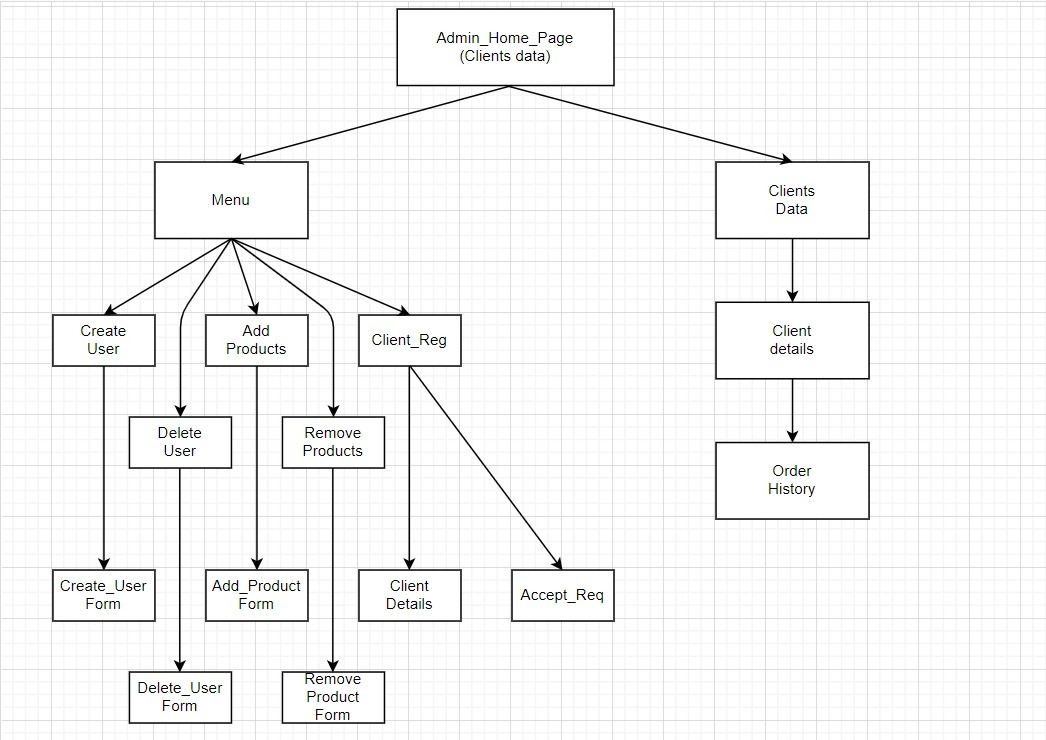
**Add Products:**

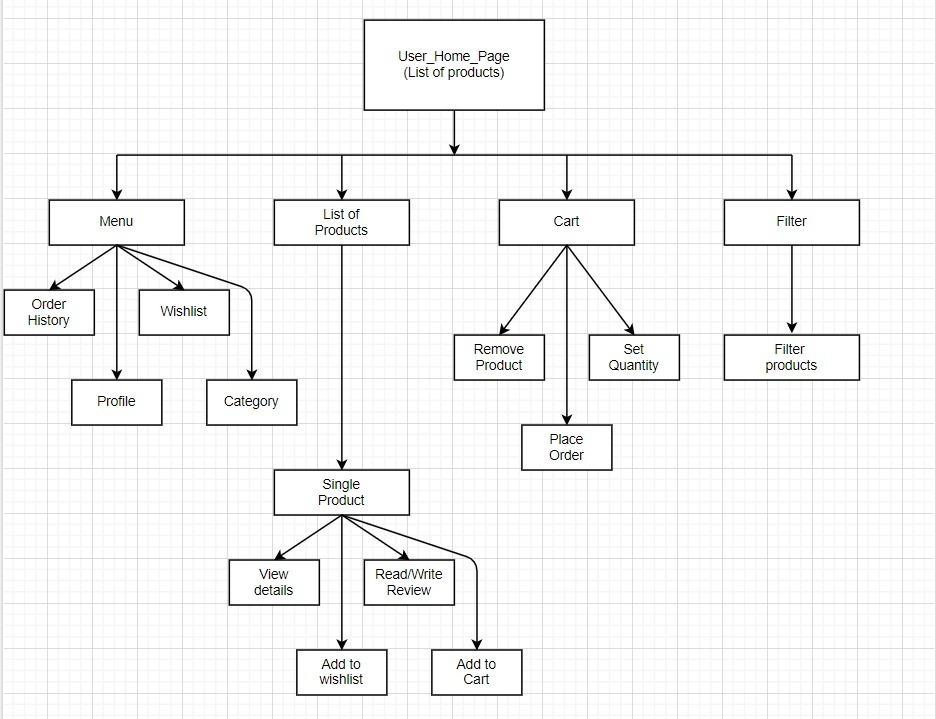
* The products can be modified by theadmin.
* The items which are not available in a particular time period are not displayed.

**Contact us & About us section :-**

* The about us section will include information.
* The contact us will contain phone or telephone.
* Any queries or enquires regarding service, reservation, etc will be done on the phone number.
* A similar feedback section will be included customers may provide feedback.



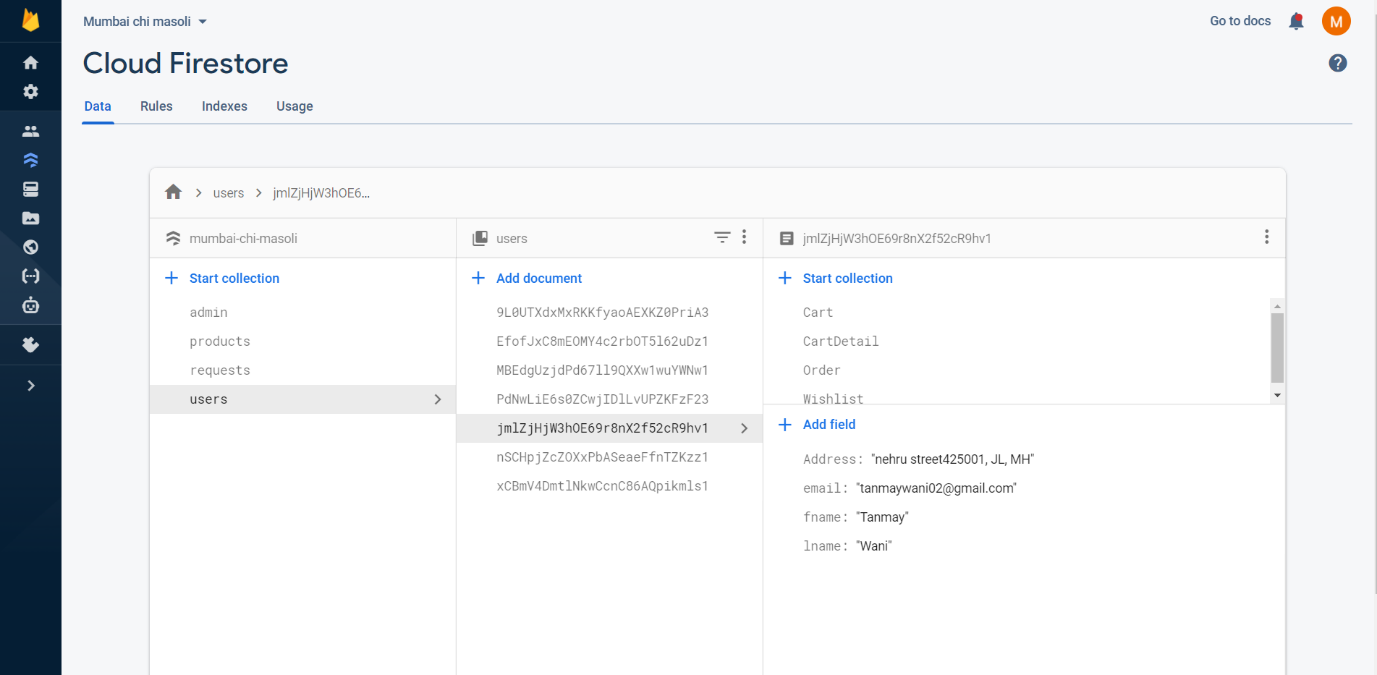




CHAPTER 6

Database Structure

**6.Database Structure**

* Cloud Firestore is a cloud-hosted, NoSQL database that your iOS, Android, and web apps can access directly via native SDKs.
* Cloud Firestore is also available in native Node.js, Java, Python, Unity, C++ and Go SDKs, in addition to REST and RPC APIs.
* Following Cloud Firestore's NoSQL data model, you store data in documents that contain fields mapping to values.
* These documents are stored in collections, which are containers for your documents that you can use to organize your data and build queries.
* 
* Documents support many different data types, from simple strings and numbers, to complex, nested objects.
* You can also create subcollections within documents and build hierarchical data structures that scale as your database grows.
* The Cloud Firestore data model supports whatever data structure works best for your app.
* Additionally, querying in Cloud Firestore is expressive, efficient, and flexible. Create shallow queries to retrieve data at the document level without needing to retrieve the entire collection, or any nested subcollections.
* Add sorting, filtering, and limits to your queries or cursors to paginate your results. To keep data in your apps current, without retrieving your entire database each time an update happens, add real-time listeners.
* Adding real-time listeners to your app notifies you with a data snapshot whenever the data your client apps are listening to changes, retrieving only the new changes.
* Protect access to your data in Cloud Firestore with Firebase Authentication and Cloud Firestore Security Rules for Android, iOS, and JavaScript, or Identity and Access Management (IAM) for server-side languages.

CHAPTER 7

Coding

**Main.dart**

import 'package:firebase\_auth/firebase\_auth.dart';  
import 'package:flutter/services.dart';  
import 'package:flutter/material.dart';  
import 'package:firebase\_core/firebase\_core.dart';  
import 'package:mumbaichimasoli/config/config.dart';  
import 'package:mumbaichimasoli/models/user.dart';  
import 'package:mumbaichimasoli/screen/splash\_screen.dart';  
// import 'package:mumbaichimasoli/screen/wrapper.dart';  
import 'package:mumbaichimasoli/services/auth.dart';  
import 'package:provider/provider.dart';  
  
void main() async {  
WidgetsFlutterBinding.*ensureInitialized*();  
SystemChrome.*setEnabledSystemUIOverlays*([]);  
 await Firebase.*initializeApp*();  
ConstantVariable.*auth*= FirebaseAuth.*instance*;  
runApp(MyApp());  
}  
  
class MyAppextends StatelessWidget {  
// This widget is the root of your application.  
@override  
Widget build(BuildContext context) {  
// ignore: missing\_required\_param  
return StreamProvider<AppUser>.value(  
 value: AuthService().user,  
child: MaterialApp(  
debugShowCheckedModeBanner: false,  
title: 'MUMBAI CHI MASOLI',  
theme: ThemeData(),  
home: SplashScreenWidget(),  
),  
);   
}  
}

**pubspec.yaml:**

name: mumbaichimasoli  
description: A new Flutter project.  
  
*# The following line prevents the package from being accidentally published to  
# pub.dev using `pub publish`. This is preferred for private packages.*publish\_to: 'none' *# Remove this line if you wish to publish to pub.dev  
  
# The following defines the version and build number for your application.  
# A version number is three numbers separated by dots, like 1.2.43  
# followed by an optional build number separated by a +.  
# Both the version and the builder number may be overridden in flutter  
# build by specifying --build-name and --build-number, respectively.  
# In Android, build-name is used as versionName while build-number used as versionCode.  
# Read more about Android versioning at https://developer.android.com/studio/publish/versioning  
# In iOS, build-name is used as CFBundleShortVersionString while build-number used as CFBundleVersion.  
# Read more about iOS versioning at  
# https://developer.apple.com/library/archive/documentation/General/Reference/InfoPlistKeyReference/Articles/CoreFoundationKeys.html*version: 1.0.0+1  
  
environment:  
sdk: ">=2.7.0 <3.0.0"  
  
dependencies:  
flutter:  
sdk: flutter  
  
  
*# The following adds the Cupertino Icons font to your application.  
 # Use with the CupertinoIcons class for iOS style icons.*cupertino\_icons: ^1.0.2  
google\_fonts: ^2.0.0  
flutter\_spinkit: ^5.0.0  
provider: ^5.0.0  
auto\_size\_text: ^2.1.0  
email\_auth: ^0.0.1+3  
fluttertoast: ^8.0.6  
uuid: ^3.0.4  
splashscreen: ^1.3.5  
image\_picker: ^0.8.1+3  
intl: ^0.17.0  
  
*# Firebase Dependencies*firebase\_core: ^1.2.1  
firebase\_auth: ^1.3.0  
cloud\_firestore: ^2.2.1  
firebase\_storage: ^9.0.0  
  
dev\_dependencies:  
flutter\_test:  
sdk: flutter  
  
*# For information on the generic Dart part of this file, see the  
# following page: https://dart.dev/tools/pub/pubspec  
  
# The following section is specific to Flutter.*flutter:  
  
*# The following line ensures that the Material Icons font is  
 # included with your application, so that you can use the icons in  
 # the material Icons class.*uses-material-design: true  
  
*# To add assets to your application, add an assets section, like this:*assets:  
 - assets/images/  
 - assets/images2/  
  
*# An image asset can refer to one or more resolution-specific "variants", see  
 # https://flutter.dev/assets-and-images/#resolution-aware.  
  
 # For details regarding adding assets from package dependencies, see  
 # https://flutter.dev/assets-and-images/#from-packages  
  
 # To add custom fonts to your application, add a fonts section here,  
 # in this "flutter" section. Each entry in this list should have a  
 # "family" key with the font family name, and a "fonts" key with a  
 # list giving the asset and other descriptors for the font. For  
 # example:  
 # fonts:  
 # - family: Schyler  
 # fonts:  
 # - asset: fonts/Schyler-Regular.ttf  
 # - asset: fonts/Schyler-Italic.ttf  
 # style: italic  
 # - family: Trajan Pro  
 # fonts:  
 # - asset: fonts/TrajanPro.ttf  
 # - asset: fonts/TrajanPro\_Bold.ttf  
 # weight: 700  
 #  
 # For details regarding fonts from package dependencies,  
 # see https://flutter.dev/custom-fonts/#from-packages*

**config.dart:**

import 'package:firebase\_auth/firebase\_auth.dart';  
  
class ConstantVariable {  
  
static FirebaseAuth*auth*;  
 static String *aid*;  
  
}

**size\_config.dart:**

import "package:flutter/widgets.dart";  
  
class SizeConfig {  
static MediaQueryData*\_mediaQueryData*;  
 static double *screenWidth*;  
 static double *screenHeight*;  
 static double *blockSizeHorizontal*;  
 static double *blockSizeVertical*;  
 static double *\_safeAreaHorizontal*;  
 static double *\_safeAreaVertical*;  
 static double *safeBlockHorizontal*;  
 static double *safeBlockVertical*;  
 static double *fontSize*;  
  
 void init(BuildContext context) {  
*\_mediaQueryData*= MediaQuery.*of*(context);  
*fontSize*= *\_mediaQueryData*.textScaleFactor;  
*screenWidth*= *\_mediaQueryData*.size.width;  
*screenHeight*= *\_mediaQueryData*.size.height;  
*blockSizeHorizontal*= *screenWidth*/ 100;  
*blockSizeVertical*= *screenHeight*/ 100;  
  
*\_safeAreaHorizontal*=  
*\_mediaQueryData*.padding.left+ *\_mediaQueryData*.padding.right;  
*\_safeAreaVertical*=  
*\_mediaQueryData*.padding.top+ *\_mediaQueryData*.padding.bottom;  
*safeBlockHorizontal*= (*screenWidth*- *\_safeAreaHorizontal*) / 100;  
*safeBlockVertical*= (*screenHeight*- *\_safeAreaVertical*) / 100;  
}  
}

**categories.dart:**

import "package:flutter/widgets.dart";  
  
class SizeConfig {  
static MediaQueryData*\_mediaQueryData*;  
 static double *screenWidth*;  
 static double *screenHeight*;  
 static double *blockSizeHorizontal*;  
 static double *blockSizeVertical*;  
 static double *\_safeAreaHorizontal*;  
 static double *\_safeAreaVertical*;  
 static double *safeBlockHorizontal*;  
 static double *safeBlockVertical*;  
 static double *fontSize*;  
  
 void init(BuildContext context) {  
*\_mediaQueryData*= MediaQuery.*of*(context);  
*fontSize*= *\_mediaQueryData*.textScaleFactor;  
*screenWidth*= *\_mediaQueryData*.size.width;  
*screenHeight*= *\_mediaQueryData*.size.height;  
*blockSizeHorizontal*= *screenWidth*/ 100;  
*blockSizeVertical*= *screenHeight*/ 100;  
  
*\_safeAreaHorizontal*=  
*\_mediaQueryData*.padding.left+ *\_mediaQueryData*.padding.right;  
*\_safeAreaVertical*=  
*\_mediaQueryData*.padding.top+ *\_mediaQueryData*.padding.bottom;  
*safeBlockHorizontal*= (*screenWidth*- *\_safeAreaHorizontal*) / 100;  
*safeBlockVertical*= (*screenHeight*- *\_safeAreaVertical*) / 100;  
}  
}

**users.dart:**

import 'package:flutter/material.dart';  
  
class Users extends StatelessWidget{  
@override  
Widget build(BuildContext context) {  
return Scaffold  
(  
appBar: AppBar  
(  
 title: Text('User List'),  
backgroundColor: Colors.*blueAccent*,  
),  
body: ListView(children: <Widget>[  
Center(  
 child: Text(  
'Users-Chart',  
style: TextStyle(fontSize: 35, fontWeight: FontWeight.*bold*, color: Colors.*blueAccent*),  
)),  
DataTable(  
 columns: const<DataColumn>[  
DataColumn(  
 label: Text(  
'FirstName',  
style: TextStyle(fontStyle: FontStyle.normal),  
),  
),  
DataColumn(  
 label: Text(  
'LastName',  
style: TextStyle(fontStyle: FontStyle.normal),  
),  
),  
DataColumn(  
 label: Text(  
'Email',  
style: TextStyle(fontStyle: FontStyle.normal),  
),  
),  
],  
rows: const<DataRow>[  
DataRow(  
 cells: <DataCell>[  
DataCell(Text('Ravi')),  
DataCell(Text('kumar')),  
DataCell(Text('ravi@gmail.com')),  
],  
),  
DataRow(  
 cells: <DataCell>[  
DataCell(Text('Jane')),  
DataCell(Text('William')),  
DataCell(Text('jane@gmail.com')),  
],  
),  
DataRow(  
 cells: <DataCell>[  
DataCell(Text('William')),  
DataCell(Text('Jane')),  
DataCell(Text('willam@gmail.com')),  
],  
),  
]  
 ),  
],  
)  
 );  
}  
}  
  
 Future navigateToUserPage(context) async {  
Navigator.*push*(context, MaterialPageRoute(builder: (context) =>Users()));  
}

**brand.dart:**

import 'package:uuid/uuid.dart';  
  
class BrandService{  
void createBrand(String text) {  
var id = Uuid();  
String brandId = id.v1();  
}  
  
}

**category.dart:**

import 'package:uuid/uuid.dart';  
  
class CategoryService{  
void createCategory(String text) {  
var id = Uuid();  
String categoryId = id.v1();  
}  
  
}

**users.dart:**

class AppUser {  
  
final String uid;  
  
AppUser({ this.uid});  
  
}  
  
class UserData {  
  
final String uid;  
 final String fname;  
 final String lname;  
 final String email;  
  
UserData({ this.uid, this.fname, this.lname, this.email});  
  
}

**addproduct.dart:**

import 'package:cloud\_firestore/cloud\_firestore.dart';  
import 'package:flutter/cupertino.dart';  
import 'package:flutter/material.dart';  
import 'dart:io';  
import 'package:firebase\_storage/firebase\_storage.dart' as firebase\_storage;  
import 'package:path/path.dart' as Path;  
  
import 'package:image\_picker/image\_picker.dart';  
  
class Addproductextends StatefulWidget {  
 List searchString;  
  
@override  
\_AddproductStatecreateState() =>\_AddproductState();  
}  
  
class \_AddproductStateextends State<Addproduct> {  
  
  
TextEditingControllersampledata1 = new TextEditingController();  
TextEditingControllersampledata2 = new TextEditingController();  
TextEditingControllersampledata3 = new TextEditingController();  
TextEditingControllersampledata4 = new TextEditingController();  
bool uploading = false;  
double val= 0;  
String value ="";  
  
  
  
  
  
  
CollectionReferenceimgRef;  
firebase\_storage.Referenceref;  
  
List<File>\_image = [];  
 final picker = ImagePicker();  
  
Future uploadFile() async {  
 int i = 1;  
 for (var imgin \_image) {  
setState(() {  
val= i / \_image.length;  
});  
ref = firebase\_storage.FirebaseStorage.*instance*.ref()  
 .child('${Path.basename(img.path)}');  
 await ref.putFile(img).whenComplete(() async {  
await ref.getDownloadURL().then((value) {  
imgRef.add({'images': value});  
i++;  
});  
});  
}  
 }  
  
@override  
Widget build(BuildContext context) {  
return Scaffold(  
appBar: AppBar(  
 title: Text("Add Products"),  
),  
body: SingleChildScrollView(  
 child: Center(  
 child: Column(  
 children: <Widget>[  
Container(  
color: Colors.*lightBlue*[50],  
padding: EdgeInsets.all(4.0),  
child: TextFormField(  
 controller: sampledata1,  
decoration: InputDecoration(  
hintText: "Name",  
labelText: "Name",  
border: OutlineInputBorder(),  
),),  
),  
SizedBox(  
 height: 7.0,  
),  
Container(  
color: Colors.*lightBlue*[50],  
padding: EdgeInsets.all(4.0),  
child: TextFormField(  
 controller: sampledata2,  
decoration: InputDecoration(  
hintText: "Discription",  
labelText: "Discription",  
border: OutlineInputBorder(),  
),),  
),  
SizedBox(  
 height: 7.0,  
),  
Container(  
color: Colors.*lightBlue*[50],  
padding: EdgeInsets.all(4.0),  
child: TextFormField(  
 controller: sampledata3,  
decoration: InputDecoration(  
hintText: "Price",  
labelText: "Price",  
border: OutlineInputBorder(),  
),  
keyboardType: TextInputType.*number*,  
),  
),  
SizedBox(  
 height: 7.0,  
),  
Container(  
color: Colors.*lightBlue*[50],  
padding: EdgeInsets.all(4.0),  
child: TextFormField(  
 controller: sampledata4,  
decoration: InputDecoration(  
hintText: "Uniqe string",  
labelText: "Search String",  
border: OutlineInputBorder(),  
),),  
),  
SizedBox(  
 height: 25.0),  
Container(  
color: Colors.*cyan*[50],  
height: 135.0,  
width: 370.0,  
child: Stack(  
 children: [  
Container(  
 child: GridView.builder(  
itemCount: \_image.length+ 1,  
  
gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(  
crossAxisCount: 3),  
itemBuilder: (context, index) {  
return index == 0  
? Center(child: IconButton(  
 icon: Icon(Icons.*add*),  
alignment: Alignment.*centerLeft*,  
onPressed: () =>  
 !uploading ? chooseImage() : null),  
)  
 : Container(margin: EdgeInsets.all(3),  
decoration: BoxDecoration(image: DecorationImage(  
 image: FileImage(\_image[index - 1]),  
fit: BoxFit.cover)),  
);  
}),  
),  
uploading  
? Center(  
 child: Column(  
mainAxisSize: MainAxisSize.min,  
children: [  
Container(  
 child: Text(  
'uploading...',  
style: TextStyle(fontSize: 20),  
),  
),  
SizedBox(  
 height: 10,  
),  
CircularProgressIndicator(  
 value: val,  
valueColor: AlwaysStoppedAnimation<Color>(Colors.*green*),  
)  
 ],  
))  
 : Container(),  
  
  
  
]),  
),  
SizedBox(  
 height: 30.0),  
Container(  
  
 child: Center(  
child:Row(  
 children:<Widget>[  
 \_button('70'),  
\_button('10'),  
\_button('15'),  
\_button('20'),  
],  
),  
  
),  
),  
SizedBox(  
 height: 80.0),  
  
  
Container(  
 child: FlatButton(  
onPressed: () {  
  
 Map <String,dynamic>  
 data={"name":sampledata1.text,"descr":sampledata2.text,"price":sampledata3.text,"searchString":sampledata4.text,};  
FirebaseFirestore.*instance*.collection("products").add(data);  
setState(() {  
uploading = true;  
});  
uploadFile().whenComplete(() =>Navigator.*of*(context).pop());  
},  
child: Text("submit",  
style: TextStyle(fontSize: 20),),  
textColor: Colors.*white*,  
color: Colors.*blue*[600],  
),  
),  
  
  
],  
),  
),  
),  
);  
}  
 Widget \_button (event)=>FlatButton(  
 child: Text(event),  
onPressed: ()async{  
DocumentReferencedocRef = FirebaseFirestore.*instance*.collection("products").doc('Fish');  
DocumentSnapshotdoct = await docRef.get();  
List Quantity = doct['Quantity'];  
 if(Quantity.contains(event)==true){  
docRef.update(  
 {  
'tags':FieldValue.*arrayRemove*([event])  
 }  
 );  
}  
else{  
docRef.update(  
 {'tags':FieldValue.*arrayUnion*([event])}  
 );  
}  
  
 },  
);  
  
chooseImage() async {  
final pickedFile = await picker.getImage(source: ImageSource.gallery);  
setState(() {  
\_image.add(File(pickedFile?.path));  
});  
 if (pickedFile.path== null) retrieveLostData();  
}  
  
 Future<void>retrieveLostData() async {  
final LostData response = await picker.getLostData();  
 if (response.file!= null) {  
setState(() {  
\_image.add(File(response.file.path));  
});  
} else {  
 print(response.file);  
}  
 }  
  
@override  
void initState() {  
super.initState();  
Map <String,dynamic> data1={'images':\_image};  
imgRef= FirebaseFirestore.*instance*.collection('products');  
}  
  
 }

**admin.dart:**

import 'package:auto\_size\_text/auto\_size\_text.dart';  
import 'package:mumbaichimasoli/constants.dart';  
import 'package:mumbaichimasoli/dashboard\_screen/categories.dart';  
import 'package:mumbaichimasoli/dashboard\_screen/users.dart';  
import 'package:flutter/cupertino.dart';  
import 'package:flutter/material.dart';  
import 'package:mumbaichimasoli/screen/SignIn.dart';  
import 'package:mumbaichimasoli/screen/SignUp.dart';  
import 'package:mumbaichimasoli/screen/addproduct.dart';  
import 'package:mumbaichimasoli/screen/product\_list.dart';  
import 'package:mumbaichimasoli/screen/req\_accounts.dart';  
import 'package:mumbaichimasoli/services/auth.dart';  
import '../db/category.dart';  
import '../db/brand.dart';  
import 'package:fluttertoast/fluttertoast.dart';  
  
import 'client\_list.dart';  
  
enumPage { dashboard, manage }  
  
class Admin extends StatefulWidget {  
@override  
\_AdminStatecreateState() =>\_AdminState();  
}  
  
class \_AdminStateextends State<Admin> {  
 Size screenSize() {  
return MediaQuery.*of*(context).size;  
}  
final AuthService\_auth = AuthService();  
Page \_selectedPage= Page.dashboard;  
//MaterialColor active = Colors.white;  
 //MaterialColornotActive = Colors.black;  
TextEditingControllercategoryController= TextEditingController();  
TextEditingControllerbrandController= TextEditingController();  
GlobalKey<FormState>\_categoryFormKey= GlobalKey();  
GlobalKey<FormState>\_brandFormKey= GlobalKey();  
BrandService\_brandService= BrandService();  
CategoryService\_categoryService= CategoryService();  
  
Widget build(BuildContext context) {  
  
var active ;  
 var notActive;  
 return Scaffold(  
appBar: AppBar(  
 title: Row(  
 children: <Widget>[  
Expanded(  
 child: FlatButton.icon(  
onPressed: () {  
setState(() =>\_selectedPage= Page.dashboard);  
},  
icon: Icon(  
Icons.*dashboard*,  
color: \_selectedPage== Page.dashboard  
? active = Colors.*white*: notActive,  
),  
label: Text('Dashboard'))),  
Expanded(  
 child: FlatButton.icon(  
onPressed: () {  
setState(() =>\_selectedPage= Page.manage);  
},  
icon: Icon(  
Icons.*sort*,  
color:  
\_selectedPage== Page.manage? active = Colors.*white*: notActive,  
),  
label: Text('Manage'))),  
],  
),  
elevation: 0.0,  
backgroundColor: Colors.*blueAccent*,  
),  
body: \_loadScreen());  
}  
  
 Widget \_loadScreen() {  
switch (\_selectedPage) {  
case Page.dashboard:  
return Column(  
 children: <Widget>[  
ListTile(  
  
 subtitle: FlatButton.icon(  
onPressed: () {},  
icon: Icon(  
Icons.*person*,  
size: 30.0,  
color: Colors.*blueAccent*,  
),  
label: Text('ADMIN',  
textAlign: TextAlign.center,  
style: TextStyle(fontSize: 30.0, color: Colors.*blueAccent*)),  
),  
  
  
  
),  
GestureDetector(  
onTap: (){  
Navigator.*push*(context, MaterialPageRoute(builder: (context)=>UserList()));  
},  
child: Container(  
margin:EdgeInsets.symmetric(vertical: 20,horizontal: 20),  
child: Padding(  
 padding: constEdgeInsets.symmetric(vertical: 19),  
child: Column(children: [  
Text("CLIENTS",style:regHead),  
Icon(Icons.*people\_outlined*,size: 100),  
],),  
),  
  
decoration: BoxDecoration(  
color: Colors.*blue*[100],  
borderRadius: BorderRadius.circular(18)  
 ),  
height: 160,  
width: screenSize().width,  
  
),  
),  
GestureDetector(  
onTap: (){  
Navigator.*push*(context, MaterialPageRoute(builder: (context)=>Requsetedaccount()));  
},  
child: Container(  
 margin: EdgeInsets.symmetric(vertical: 20,horizontal: 20),  
decoration: BoxDecoration(  
color: Colors.*blue*[100],  
borderRadius: BorderRadius.circular(18)  
 ),  
height: 160,  
width: screenSize().width,  
  
  
  
  
  
child: Padding(  
 padding: constEdgeInsets.symmetric(vertical: 19),  
child: Column(  
 children: [  
Text("Requested Account",style: regHead,),  
Icon(Icons.*announcement\_outlined*,size: 100,)  
 ],  
),  
)  
 ),  
),  
  
  
  
  
GestureDetector(  
onTap: (){  
Navigator.*push*(context, MaterialPageRoute(builder: (context)=>ProductList()));  
},  
child: Container(  
 margin: EdgeInsets.symmetric(vertical: 20,horizontal: 20),  
decoration: BoxDecoration(  
color: Colors.*blue*[100],  
borderRadius: BorderRadius.circular(18)  
 ),  
height: 160,  
width: screenSize().width,  
  
  
child: Padding(  
 padding: constEdgeInsets.symmetric(vertical: 19),  
child: Column(  
 children: [  
Text("Products",style: regHead,),  
Icon(Icons.*track\_changes*,size: 100,)  
 ],  
),  
)  
 ),  
),  
  
  
],  
);  
 break;  
 case Page.manage:  
return ListView(  
 children: <Widget>[  
  
Divider(),  
ListTile(  
 leading: Icon(Icons.*change\_history*),  
title: Text("Products list"),  
onTap: () {  
Navigator.*push*(context, MaterialPageRoute(  
 builder: (context) =>ProductList(),  
));  
} ),  
  
Divider(),  
ListTile(  
 leading: Icon(Icons.*people\_outline*),  
title: Text("Add users"),  
onTap: () {  
Navigator.*push*(context, MaterialPageRoute(  
 builder: (context) =>Signup(),  
),  
);  
  
},  
),  
Divider(),  
ListTile(  
 leading: Icon(Icons.*account\_circle*),  
title: Text("Requested accounts"),  
onTap: () {  
Navigator.*push*(context, MaterialPageRoute(  
 builder: (context) =>Requsetedaccount(),  
));  
} ),  
  
Divider(),  
ListTile(  
 leading: Icon(Icons.*library\_books*),  
title: Text("Users list"),  
onTap: () {  
// navigateToUserPage(context);  
Navigator.*push*(context, MaterialPageRoute(  
 builder: (context) =>UserList(),  
)  
 );  
},  
),  
Divider(),  
ListTile(  
 leading: Icon(Icons.*logout*),  
title: Text("Logout"),  
onTap: () {  
Navigator.*pushReplacement*(context, MaterialPageRoute(  
 builder: (context) =>SignIn(),  
));  
},  
)  
 ],  
);  
 break;  
 default:  
return Container();  
}  
 }  
  
void \_categoryAlert() {  
var alert = new AlertDialog(  
 content: Form(  
 key: \_categoryFormKey,  
child: TextFormField(  
 controller: categoryController,  
validator: (value) {  
if (value.isEmpty) {  
return 'category cannot be empty';  
}  
 },  
decoration: InputDecoration(hintText: "add category"),  
),  
),  
actions: <Widget>[  
FlatButton(  
onPressed: () {  
if (categoryController.text!= () {}) {  
\_categoryService.createCategory(categoryController.text);  
}  
Fluttertoast.*showToast*(msg: 'category created');  
Navigator.*pop*(context);  
},  
child: Text('ADD')),  
FlatButton(  
onPressed: () {  
Navigator.*pop*(context);  
},  
child: Text('CANCEL')),  
],  
);  
  
showDialog(context: context, builder: (\_) => alert);  
}  
  
void \_brandAlert() {  
var alert = new AlertDialog(  
 content: Form(  
 key: \_brandFormKey,  
child: TextFormField(  
 controller: brandController,  
validator: (value) {  
if (value.isEmpty) {  
return 'category cannot be empty';  
}  
 },  
decoration: InputDecoration(hintText: "add brand"),  
),  
),  
actions: <Widget>[  
FlatButton(  
onPressed: () {  
if (brandController.text!= () {}) {  
\_brandService.createBrand(brandController.text);  
}  
Fluttertoast.*showToast*(msg: 'brand added');  
Navigator.*pop*(context);  
},  
child: Text('ADD')),  
FlatButton(  
onPressed: () {  
Navigator.*pop*(context);  
},  
child: Text('CANCEL')),  
],  
);  
  
showDialog(context: context, builder: (\_) => alert);  
}  
  
 Future navigateToUserPage(context) async {  
Navigator.*push*(context, MaterialPageRoute(builder: (context) =>Users()));  
}  
  
 Future navigateToAdminPage(context) async {  
Navigator.*push*(context, MaterialPageRoute(builder: (context) =>Admin()));  
}  
  
 Future navigateToCategoryListPage(context) async {  
Navigator.*push*(context, MaterialPageRoute(builder: (context) =>CategoryList()));  
}  
}

**adnin\_Changepass.dart:**

import 'package:flutter/material.dart';  
import 'package:flutter/services.dart';  
import 'package:google\_fonts/google\_fonts.dart';  
import 'package:auto\_size\_text/auto\_size\_text.dart';  
import 'package:mumbaichimasoli/screen/admin\_login.dart';  
import 'package:mumbaichimasoli/services/auth.dart';  
import 'package:mumbaichimasoli/services/database.dart';  
  
class NewPasswordPageextends StatefulWidget {  
@override  
\_NewPasswordPageStatecreateState() =>\_NewPasswordPageState();  
}  
  
class \_NewPasswordPageStateextends State<NewPasswordPage> {  
  
void \_togglePasswordView() {  
setState(() {  
isHiddenPassword= !isHiddenPassword;  
});  
}  
  
void \_toggleConfirmPasswordView() {  
setState(() {  
isHiddenConfirmPassword= !isHiddenConfirmPassword;  
});  
}  
  
var \_changePassformKey= GlobalKey<FormState>();  
  
bool isHiddenPassword= true;  
bool isHiddenConfirmPassword= true;  
String newPass;  
String confirmPass;  
String error;  
  
TextEditingControllernewPasswordcon= new TextEditingController();  
TextEditingControllerconfirmPasscon= new TextEditingController();  
  
confirmPassword() {  
if(newPass== confirmPass) {  
return null;  
} else {  
setState(() {  
error = "Password doesn't match";  
});  
}  
 }  
  
 Widget showAlert() {  
if (error != null) {  
return Container(  
color: Colors.*amberAccent*,  
width: double.*infinity*,  
padding: EdgeInsets.all(8.0),  
child: Row(  
 children: <Widget>[  
Padding(  
 padding: constEdgeInsets.only(right: 8.0),  
child: Icon(Icons.*error*),  
),  
Expanded(  
 child: AutoSizeText(  
error,  
maxLines: 3,  
)  
 ),  
Padding(  
 padding: constEdgeInsets.only(left: 8.0),  
child: IconButton(  
 icon: Icon(Icons.*close*),  
onPressed: () {  
setState(() {  
error = null;  
});  
}  
 ),  
)  
 ],  
),  
);  
}  
return SizedBox(  
 height: 8,  
);  
}  
  
@override  
Widget build(BuildContext context) {  
return Scaffold(  
 body: AnnotatedRegion<SystemUiOverlayStyle>(  
 value: SystemUiOverlayStyle.*light*,  
child: GestureDetector(  
onTap: () =>FocusScope.*of*(context).unfocus(),  
child: Stack(  
 children: <Widget>[  
Container(  
 height: double.*infinity*,  
width: double.*infinity*,  
decoration: BoxDecoration(  
 gradient: LinearGradient(  
 begin: Alignment.*topCenter*,  
end: Alignment.*bottomCenter*,  
colors: [  
Color(0xFF73AEF5),  
Color(0xFF61A4F1),  
Color(0xFF478DE0),  
Color(0xFF398AE5),  
],  
stops: [0.1, 0.4, 0.7, 0.9],  
),  
),  
child: SingleChildScrollView(  
 physics: AlwaysScrollableScrollPhysics(),  
padding: EdgeInsets.symmetric(  
 horizontal: 40,  
vertical: 110,  
),  
child: Form(  
 key: \_changePassformKey,  
child: Column(  
mainAxisAlignment: MainAxisAlignment.center,  
children: <Widget>[  
SizedBox(height: 10,),  
showAlert(),  
SizedBox(height: 20,),  
Align(  
 alignment: Alignment.*topLeft*,  
child: Text(  
'Change Password',  
style: GoogleFonts.*openSans*(  
fontSize: 50,  
color: Colors.*white*,  
fontWeight: FontWeight.*w600*,  
),  
),  
),  
SizedBox(  
 height: 50,  
),  
Column(  
crossAxisAlignment: CrossAxisAlignment.start,  
children: <Widget>[   
SizedBox(  
 height: 10,  
),  
Container(  
 alignment: Alignment.*centerLeft*,  
decoration: BoxDecoration(  
color: Color(0xFF6CA8F1),  
borderRadius: BorderRadius.circular(10),  
boxShadow: [  
BoxShadow(  
color: Colors.*black12*,  
blurRadius: 6,  
offset: Offset(0, 2)  
 )  
 ]  
 ),  
height: 60,  
child: TextFormField(  
 validator: PasswordValidator.*validate*,  
controller: newPasswordcon,  
obscureText: isHiddenPassword,  
style: TextStyle(  
color: Colors.*white*,  
),  
decoration: InputDecoration(  
 border: InputBorder.*none*,  
contentPadding: EdgeInsets.only(top: 14),  
prefixIcon: Icon(Icons.*lock*, color: Colors.*white*),  
hintText: "New Password",  
hintStyle:  
TextStyle(color: Colors.*grey*[300]),  
suffixIcon: InkWell(  
onTap: \_togglePasswordView,  
child: Icon(  
isHiddenPassword? Icons.*visibility*: Icons.*visibility\_off*,  
color: Colors.*white*,  
),  
)  
 ),  
)  
 )  
 ],  
),  
SizedBox(  
 height: 30,  
),  
Column(  
crossAxisAlignment: CrossAxisAlignment.start,  
children: <Widget>[  
SizedBox(  
 height: 10,  
),  
Container(  
 alignment: Alignment.*centerLeft*,  
decoration: BoxDecoration(  
color: Color(0xFF6CA8F1),  
borderRadius: BorderRadius.circular(10),  
boxShadow: [  
BoxShadow(  
color: Colors.*black12*,  
blurRadius: 6,  
offset: Offset(0, 2),  
)  
 ]  
 ),  
height: 60,  
child: TextFormField(  
 validator: confirmPassword(),  
controller: confirmPasscon,  
obscureText: isHiddenConfirmPassword,  
style: TextStyle(  
color: Colors.*white*,  
),  
decoration: InputDecoration(  
 border: InputBorder.*none*,  
contentPadding: EdgeInsets.only(top: 14),  
prefixIcon: Icon(Icons.*lock*, color: Colors.*white*),  
hintText: "Confirm Password",  
hintStyle:  
TextStyle(color: Colors.*grey*[300]),  
suffixIcon: InkWell(  
onTap: \_toggleConfirmPasswordView,  
child: Icon(  
isHiddenConfirmPassword? Icons.*visibility*: Icons.*visibility\_off*,  
color: Colors.*white*,  
),  
)  
 ),  
)  
 )  
 ],  
),  
SizedBox(  
 height: 12,  
),  
Container(  
 padding: EdgeInsets.symmetric(vertical: 25),  
width: double.*infinity*,  
child: ElevatedButton(  
 child: Text(  
'Update Password',  
style: GoogleFonts.*openSans*(  
color: Color(0xFF527DAA),  
fontSize: 20,  
fontWeight: FontWeight.*bold*,  
letterSpacing: 1.5,  
),  
),  
onPressed: () async {  
setState(() {  
newPass= newPasswordcon.text;  
confirmPass= confirmPasscon.text;  
});  
 if(confirmPassword() == null) {  
 print("inside confirm password loop");  
 dynamic result = await DatabaseService().updateAdminPassword(newPass);  
 if(result == null){  
 print("Password successfully change");  
Navigator.*of*(context).pop();  
}  
 }  
 },  
style: ElevatedButton.*styleFrom*(  
 padding: EdgeInsets.all(15),  
primary: Colors.*white*,  
shape: RoundedRectangleBorder(  
borderRadius: BorderRadius.circular(30)  
 )  
 ),  
)  
 ),  
],  
),  
),  
),  
)  
 ],  
),  
),  
),  
);   
}  
}

**admin\_login.dart**

import 'package:auto\_size\_text/auto\_size\_text.dart';  
import 'package:flutter/material.dart';  
import 'package:flutter/services.dart';  
import 'package:mumbaichimasoli/screen/admin.dart';  
import 'package:mumbaichimasoli/screen/SignUp.dart';  
import 'package:mumbaichimasoli/screen/SignIn.dart';  
import 'package:mumbaichimasoli/screen/VerifyEmail.dart';  
import 'package:mumbaichimasoli/services/auth.dart';  
import 'package:mumbaichimasoli/shared/loading.dart';  
import 'package:cloud\_firestore/cloud\_firestore.dart';  
import '../constants.dart';  
  
class AdminLoginScreenextends StatefulWidget {  
@override  
\_AdminLoginScreenStatecreateState() =>\_AdminLoginScreenState();  
}  
  
class \_AdminLoginScreenStateextends State<AdminLoginScreen> {  
  
void \_togglePasswordView() {  
setState(() {  
isHiddenPassword= !isHiddenPassword;  
});  
}  
  
 bool loading = false;  
bool isHiddenPassword= true;  
String adminUsername;  
String adminpass;  
String error;  
  
TextEditingControlleradminUsernamecon= new TextEditingController();  
TextEditingControlleradminpasscon= new TextEditingController();  
  
Widget showAlert() {  
if (error != null) {  
return Container(  
color: Colors.*amberAccent*,  
width: double.*infinity*,  
padding: EdgeInsets.all(8.0),  
child: Row(  
 children: <Widget>[  
Padding(  
 padding: constEdgeInsets.only(right: 8.0),  
child: Icon(Icons.*error*),  
),  
Expanded(  
 child: AutoSizeText(  
error,  
maxLines: 3,  
)  
 ),  
Padding(  
 padding: constEdgeInsets.only(left: 8.0),  
child: IconButton(  
 icon: Icon(Icons.*close*),  
onPressed: () {  
setState(() {  
error = null;  
});  
}),  
)  
 ],  
),  
);  
}  
return SizedBox(  
 height: 8,  
);  
}  
  
var \_adminformKey= GlobalKey<FormState>();  
Widget \_buildEmailTF() {  
return Column(  
crossAxisAlignment: CrossAxisAlignment.start,  
children: <Widget>[  
SizedBox(height: 10.0),  
Container(  
 alignment: Alignment.*centerLeft*,  
decoration: kBoxDecorationStyle,  
height: 60.0,  
child: TextFormField(  
 controller: adminUsernamecon,  
validator: UsernameValidator.*validate*,  
keyboardType: TextInputType.*emailAddress*,  
style: TextStyle(  
color: Colors.*white*,  
fontFamily: 'OpenSans',  
),  
decoration: InputDecoration(  
 border: InputBorder.*none*,  
contentPadding: EdgeInsets.only(top: 14.0),  
prefixIcon: Icon(  
Icons.*person*,  
color: Colors.*white*,  
),  
hintText: 'Username',  
hintStyle: kHintTextStyle,  
),  
),  
),  
],  
);  
}  
  
 Widget \_buildPasswordTF() {  
return Column(  
crossAxisAlignment: CrossAxisAlignment.start,  
children: <Widget>[  
SizedBox(height: 10.0),  
Container(  
 alignment: Alignment.*centerLeft*,  
decoration: kBoxDecorationStyle,  
height: 60.0,  
child: TextFormField(  
 validator: PasswordValidator.*validate*,  
controller: adminpasscon,  
obscureText: isHiddenPassword,  
style: TextStyle(  
color: Colors.*white*,  
fontFamily: 'OpenSans',  
),  
decoration: InputDecoration(  
 border: InputBorder.*none*,  
contentPadding: EdgeInsets.only(top: 14.0),  
prefixIcon: Icon(  
Icons.*lock*,  
color: Colors.*white*,  
),  
hintText: 'Password',  
hintStyle: kHintTextStyle,  
suffixIcon: InkWell(  
onTap: \_togglePasswordView,  
child: Icon(  
isHiddenPassword? Icons.*visibility*: Icons.*visibility\_off*,  
color: Colors.*white*,  
)  
 )  
 ),  
),  
),  
],  
);  
}  
  
 Widget \_buildForgotPasswordBtn() {  
return Container(  
 alignment: Alignment.*centerRight*,  
child: TextButton(  
onPressed: () =>Navigator.*push*(  
context,   
MaterialPageRoute(builder: (context) =>VerifyEmailPage())  
 ),  
child: Text(  
'Forgot Password?',  
style: kLabelStyle,  
),  
style: TextButton.*styleFrom*(  
 padding: EdgeInsets.only(right: 0.0),  
),  
),  
);  
}  
  
 Widget \_buildLoginBtn() {  
return Container(  
 padding: EdgeInsets.symmetric(vertical: 25.0),  
width: double.*infinity*,  
child: ElevatedButton(  
onPressed: () {  
if(\_adminformKey.currentState.validate()) {  
setState(() {  
adminUsername= adminUsernamecon.text;  
adminpass= adminpasscon.text;  
loading = true;  
});  
 if(adminSignin(adminUsername, adminpass) == null) {  
setState(() {  
loading = false;  
});  
}  
 }  
 },  
child: Text(  
'LOGIN',  
style: TextStyle(  
color: Color(0xFF527DAA),  
letterSpacing: 1.5,  
fontSize: 18.0,  
fontWeight: FontWeight.*bold*,  
fontFamily: 'OpenSans',  
),  
),  
style: ElevatedButton.*styleFrom*(  
 padding: EdgeInsets.all(15.0),  
shape: RoundedRectangleBorder(  
borderRadius: BorderRadius.circular(30.0),  
),  
primary: Colors.*white*,  
),  
),  
);  
}  
  
adminSignin(String username, String password) {  
 FirebaseFirestore.*instance*.collection('admin').get().then((snapshot) {  
snapshot.docs.forEach((result) {  
if(username.isEmpty){  
setState(() {  
error = "Username can't be empty";  
});  
 return null;  
}   
else if(password.isEmpty) {  
setState(() {  
error = "Password can't be empty";  
});  
 return null;  
}   
else if (result.data()["username"] != username) {  
setState(() {  
error = "Username doesn't match";  
});  
 return null;  
}   
else if (result.data()["password"] != password) {  
setState(() {  
error = "Password doesn't match";  
});  
 return null;  
}   
else {  
 Route route = MaterialPageRoute(builder: (c) =>Admin());  
Navigator.*pushReplacement*(context, route);  
}  
 });  
});  
}  
  
@override  
Widget build(BuildContext context) {  
return loading ? Loading() : new WillPopScope(  
onWillPop: () async =>false,  
child: Scaffold(  
 body: AnnotatedRegion<SystemUiOverlayStyle>(  
 value: SystemUiOverlayStyle.*light*,  
child: GestureDetector(  
onTap: () =>FocusScope.*of*(context).unfocus(),  
child: Stack(  
 children: <Widget>[  
Container(  
 height: double.*infinity*,  
width: double.*infinity*,  
decoration: BoxDecoration(  
 gradient: LinearGradient(  
 begin: Alignment.*topCenter*,  
end: Alignment.*bottomCenter*,  
colors: [  
Color(0xFF73AEF5),  
Color(0xFF61A4F1),  
Color(0xFF478DE0),  
Color(0xFF398AE5),  
],  
stops: [0.1, 0.4, 0.7, 0.9],  
),  
),  
),  
Container(  
 height: double.*infinity*,  
child: SingleChildScrollView(  
 physics: AlwaysScrollableScrollPhysics(),  
padding: EdgeInsets.symmetric(  
 horizontal: 40.0,  
vertical: 90.0,  
),  
child: Column(  
mainAxisAlignment: MainAxisAlignment.center,  
children: <Widget>[  
SizedBox(  
 height: 10,  
),  
showAlert(),  
SizedBox(height: 20),  
SizedBox(  
 height: 22,  
),  
Form(  
 key: \_adminformKey,  
child: Text(  
'Admin Sign In',  
style: TextStyle(  
color: Colors.*white*,  
fontFamily: 'OpenSans',  
fontSize: 30.0,  
fontWeight: FontWeight.*bold*,  
),  
),  
),  
SizedBox(height: 30.0),  
\_buildEmailTF(),  
SizedBox(height: 30.0),  
\_buildPasswordTF(),  
\_buildForgotPasswordBtn(),  
\_buildLoginBtn(),  
SizedBox(height: 80),  
Align(  
 alignment: Alignment.*centerLeft*,  
child: GestureDetector(  
onTap: () {  
Navigator.*pushReplacement*(context, MaterialPageRoute(builder: (context) =>SignIn()));  
},  
child: Row(  
 children: [  
Icon(  
Icons.*block*,  
color: Colors.*white*,  
),  
SizedBox(width: 3,),  
Text(  
"I ain't admin",  
style: kLabelStyle,  
),  
],  
),  
),  
)  
 ],  
),  
),  
)  
 ],  
),  
),  
),  
)  
 );  
}  
}

**cartpage.dart:**

import 'package:auto\_size\_text/auto\_size\_text.dart';  
import 'package:cloud\_firestore/cloud\_firestore.dart';  
import 'package:flutter/material.dart';  
import 'package:mumbaichimasoli/config/size\_config.dart';  
import 'package:mumbaichimasoli/services/auth.dart';  
import 'package:mumbaichimasoli/services/database.dart';  
import 'package:mumbaichimasoli/shared/loading.dart';  
import 'package:mumbaichimasoli/widgets/cartbottom.dart';  
import 'package:mumbaichimasoli/widgets/enterqty.dart';  
import 'package:mumbaichimasoli/widgets/SearchInput.dart';  
import 'package:mumbaichimasoli/widgets/topBar.dart';  
import 'package:mumbaichimasoli/widgets/custom\_Stepper.dart';  
import 'package:provider/provider.dart';  
import '../constants.dart';  
import 'package:mumbaichimasoli/screen/detailProdAdmin.dart';  
  
class CartPageextends StatefulWidget {  
@override  
\_CartPageStatecreateState() =>\_CartPageState();  
}  
  
class \_CartPageStateextends State<CartPage> {  
AuthService\_auth = AuthService();  
DatabaseService\_databaseService= DatabaseService();  
  
bool loading = false;  
int \_quantity = 10;  
int amount = 0;  
// dynamic total = 10;  
  
calculateAmount(String docid, int quantity, int price) async {  
amount = quantity \* price;  
 await \_databaseService.updateProductAmount(docid, amount);  
calculateTotal();  
}  
  
calculateTotal() async {  
// double total;  
double temp;  
\_databaseService.userCollection.doc(\_auth.getUserId()).collection('Cart').snapshots().listen((snapshot) {  
 temp = snapshot.docs.fold(0, (tot, doc) => tot + doc['Amount']);  
// print('temp $temp');  
\_databaseService.updateCartTotal(temp);  
});  
}  
  
@override  
Widget build(BuildContext context) {  
SizeConfig().init(context);  
  
  
  
  
Size screenSize() {  
return MediaQuery.*of*(context).size;  
}  
  
return Scaffold(  
 body: Stack(  
 children: [  
StreamBuilder(  
 stream: \_databaseService.userCollection  
.doc(\_auth.getUserId())  
 .collection("Cart")  
 .snapshots(),  
builder: (context, snapshot) {  
if (snapshot.hasError) {  
return Scaffold(  
 body: Center(  
 child: Text("Error: ${snapshot.error}"),  
),  
);  
}  
  
  
if (snapshot.connectionState== ConnectionState.active) {  
return ListView(  
 padding: EdgeInsets.only(top: 90.0, bottom: 80.0),  
children: snapshot.data.docs.map<Widget>((document) {  
  
calculateTotal();  
 return GestureDetector(  
  
 child: StreamBuilder(  
 stream: \_databaseService.productsReference  
.doc(document.id)  
 .snapshots(),  
builder: (context, productSnap) {  
if (productSnap.hasError) {  
return Container(  
 child: Center(  
 child: Text("${productSnap.error}"),  
),  
);  
}  
if (productSnap.connectionState==  
ConnectionState.active) {  
 Map \_productMap = productSnap.data.data();  
  
 return Stack(  
 children: [  
SingleChildScrollView(  
 child: Container(  
 decoration: BoxDecoration(  
borderRadius:  
BorderRadius.circular(10),  
color: Colors.*white*,  
boxShadow: [  
BoxShadow(  
color: Colors.*grey*,  
offset: Offset(0.0, 1.0), //(x,y)  
blurRadius: 2.0,  
),  
],  
),  
// margin: EdgeInsets.symmetric(  
 // horizontal: 10, vertical: 6),  
child: Padding(  
 padding: constEdgeInsets.symmetric(  
 vertical: 16.0,  
horizontal: 24.0,  
),  
child: Row(  
mainAxisAlignment:  
MainAxisAlignment.spaceEvenly,  
children: [  
Container(  
width:SizeConfig.*blockSizeHorizontal*\* 25,  
height: SizeConfig.*blockSizeVertical*\*12,  
child: ClipRRect(  
borderRadius:  
BorderRadius.circular(8.0),  
child: Image.network(  
"${\_productMap['images'][0]}",  
fit: BoxFit.cover,  
height: 50,  
width: 20,  
),  
),  
),  
Expanded(  
 child:   
Container(  
 padding: EdgeInsets.only(  
 left: 10.0,  
// right: 10.0,  
),  
height: 100,  
child: Column(  
mainAxisAlignment:  
MainAxisAlignment.start,  
crossAxisAlignment:  
CrossAxisAlignment.start,  
children: [  
Row(  
 children: <Widget>[  
Container(  
 height: SizeConfig  
 .*blockSizeHorizontal*\*  
10,  
width: SizeConfig  
 .*blockSizeVertical*\*  
10,  
// color: Colors.blue,  
child: AutoSizeText(  
"${\_productMap['name']}",  
maxLines: 1,  
style: TextStyle(  
fontWeight:  
FontWeight  
 .*bold*),  
)),  
Padding(  
 padding: constEdgeInsets.only(left: 21),  
child: CustomStepper(  
lowerLimit: 1,  
upperLimit: 100,  
stepValue: 1,  
iconSize: 18,  
value: document['Quantity'],  
onChanged:  
 (value) {  
\_databaseService.updateProductQuantity(document.id, value);  
calculateAmount(document.id, value, \_productMap['price']);  
  
print("Amount: ${document['Amount']}");  
}),  
),  
  
],  
),  
  
SizedBox(  
 height: 11,  
),  
Row(  
 children: [  
Text(  
"Price:",  
style: TextStyle(  
fontSize: 14.0,  
fontWeight: FontWeight.*w600*,  
),  
),  
SizedBox(  
 width: 10,  
),  
Text(  
"\₹${\_productMap['price']}",  
style: TextStyle(  
fontSize: 14.0,  
color: Theme.*of*(  
 context)  
 .accentColor,  
fontWeight:  
FontWeight  
 .*w600*),  
),  
],  
),  
Row(  
 children: [  
Text(  
"Amount:",  
style: TextStyle(  
fontSize: 14.0,  
fontWeight: FontWeight.*w600*,  
),  
),  
SizedBox(  
 width: 10,  
),  
Container(  
 height: SizeConfig  
 .*blockSizeHorizontal*\*  
5,  
width: 70,  
// color: Colors.blue,  
  
child: AutoSizeText(  
"\₹${document['Amount']}",  
maxLines: 1,  
style: TextStyle(  
fontSize: 14.0,  
color: Theme.*of*(context).accentColor,  
fontWeight:  
FontWeight  
 .*w600*),  
)),  
Padding(  
 padding: constEdgeInsets.only(left: 89),  
child: GestureDetector(  
onTap: () async {  
await \_databaseService.removeProductFromCart(document.id);  
calculateTotal();  
},  
child: Icon(Icons.*delete\_forever\_outlined*),  
),  
)  
 ],  
),  
  
  
],  
),  
),  
),  
  
],  
),  
),  
),  
  
),  
],  
);  
  
  
}  
return Container(  
 child: Center(  
 child: CircularProgressIndicator(),  
),  
);  
}));  
}).toList(),  
);  
}  
  
return Scaffold(  
 body: Center(  
 child: CircularProgressIndicator(),  
),  
);  
},  
),  
TopBar(  
 title: "Cart",  
backArrow: true,  
hasCart: false,  
),  
Align(alignment: Alignment.*bottomCenter*, child: Cartbottom())  
 ],  
));  
  
}  
}

**client\_list.dart:**

import 'package:auto\_size\_text/auto\_size\_text.dart';  
import 'package:cloud\_firestore/cloud\_firestore.dart';  
import 'package:flutter/material.dart';  
import 'package:flutter/rendering.dart';  
import 'package:mumbaichimasoli/services/database.dart';  
import 'package:mumbaichimasoli/constants.dart';  
  
class UserListextends StatefulWidget {  
  
  
@override  
  
\_UserListStatecreateState() =>\_UserListState();  
  
}  
  
class \_UserListStateextends State<UserList> {  
DatabaseService\_databaseService= DatabaseService();  
@override  
Widget build(BuildContext context) {  
 Size screenSize() {  
return MediaQuery.*of*(context).size;  
}  
return Scaffold(  
appBar: AppBar(  
 title: Text('User list'),  
centerTitle: true,  
),  
body: Stack(  
 children: <Widget>[  
FutureBuilder<QuerySnapshot>(  
 future: \_databaseService.userCollection.get(),  
// ignore: missing\_return  
builder: (context, snapshot) {  
if (snapshot.hasError) {  
return Scaffold(  
 body: Center(  
 child: Text("Error : ${snapshot.error}"),  
),  
);  
}  
if (snapshot.connectionState== ConnectionState.done) {  
return ListView(  
 children: snapshot.data.docs.map((document) {  
  
  
return Container(  
 height: 110,  
width: 110,  
color: Colors.*blueGrey*[50],  
child:Stack(  
 children: [  
Column(  
mainAxisAlignment: MainAxisAlignment.start,  
children:[  
Padding(  
 padding: constEdgeInsets.symmetric(vertical:15,horizontal: 18),  
child: Row(  
 children:[  
Text("Name:",style: regHead,),  
Text("${document["fname"]}",style: regtext,),  
SizedBox(width:10),  
Text("${document["lname"]}",style: regtext,)  
  
  
 ]  
 ),  
),  
  
Padding(  
 padding: constEdgeInsets.symmetric(horizontal:18.0),  
child: Container(  
 child: Row(  
  
 children: [  
Text("Email:",style: regHead),  
Container(  
 width: 300,  
child: AutoSizeText("${document["email"]}", maxLines: 1,style: regtext,)),  
],  
),  
),  
),  
]  
 ),  
],  
  
  
),  
);  
  
}).toList()  
 );  
}  
 }  
 )  
 ]  
 ),  
);  
}  
}

**detailProd.dart:**

// import 'package:cloud\_firestore/cloud\_firestore.dart';  
// import 'package:firebase\_auth/firebase\_auth.dart';  
import 'package:flutter/material.dart';  
import 'package:google\_fonts/google\_fonts.dart';  
import 'package:mumbaichimasoli/constants.dart';  
import 'package:mumbaichimasoli/services/auth.dart';  
import 'package:mumbaichimasoli/services/database.dart';  
import 'package:mumbaichimasoli/widgets/Fishquantity.dart';  
import 'package:mumbaichimasoli/widgets/imageSwipe.dart';  
import 'package:mumbaichimasoli/widgets/topBar.dart';  
import 'package:cloud\_firestore/cloud\_firestore.dart';  
import 'package:flutter/material.dart';  
import 'package:mumbaichimasoli/constants.dart';  
import 'package:mumbaichimasoli/services/database.dart';  
import 'package:mumbaichimasoli/widgets/imageSwipe.dart';  
import 'package:mumbaichimasoli/widgets/topBar.dart';  
import 'package:mumbaichimasoli/config/size\_config.dart';  
import 'package:fluttertoast/fluttertoast.dart';  
  
class DetailProdextends StatefulWidget {  
final String productId;  
DetailProd({this.productId});  
@override  
\_DetailProdStatecreateState() =>\_DetailProdState();  
}  
  
class \_DetailProdStateextends State<DetailProd> {  
TextEditingControllerfeedback = new TextEditingController();  
AuthService\_auth = AuthService();  
DatabaseService\_databaseService= DatabaseService();  
  
  
int \_selQuantity= 0;  
int index = 0;  
int amount = 0;  
  
calculateAmount(String docid, int quantity, int price) {  
 int result = quantity \* price;  
setState(() {  
amount = result;  
});  
}  
  
 Future \_add2Cart() {  
return \_databaseService.userCollection  
.doc(\_auth.getUserId())  
 .collection("Cart")  
 .doc(widget.productId)  
 .set({  
"Quantity": \_selQuantity,  
"Amount": amount,  
"id": widget.productId,  
});  
}  
  
 Future \_add2wishlist() {  
return \_databaseService.userCollection  
.doc(\_auth.getUserId())  
 .collection("Wishlist")  
 .doc(widget.productId)  
 .set({  
"Quantity": \_selQuantity,  
});  
}  
  
final SnackBar\_add2cartmsg = SnackBar(  
 duration: constDuration(seconds: 02),  
  
// padding: EdgeInsets.only(bottom: 24),  
backgroundColor: Colors.*green*,  
content: Text("Fish added to cart ",  
style: GoogleFonts.*openSans*(  
fontWeight: FontWeight.*w600*, color: Colors.*white*)));  
  
 final SnackBar\_add2wishlistmsg = SnackBar(  
 duration: constDuration(seconds: 2),  
backgroundColor: Colors.*green*,  
content: Text(  
"Fish added to wishlist",  
style: GoogleFonts.*openSans*(  
fontWeight: FontWeight.*w600*, color: Colors.*white*),  
));  
  
@override  
Widget build(BuildContext context) {  
return Scaffold(  
resizeToAvoidBottomInset: false,  
body: Stack(  
 children: <Widget>[  
FutureBuilder(  
 future: \_databaseService.productsReference  
.doc(widget.productId)  
 .get(),  
builder: (context, snapshot) {  
if (snapshot.hasError) {  
return Scaffold(  
 body: Center(  
 child: Text("Error:${snapshot.hasError}"),  
),  
);  
}  
if (snapshot.connectionState== ConnectionState.done) {  
 Map<String, dynamic>documentData = snapshot.data.data();  
  
List album = documentData['images'];  
  
List quantiy = snapshot.data['quantity'];  
  
  
 return Expanded(  
 child: ListView(  
 padding: EdgeInsets.all(0),  
children: [  
ImageSwipe(  
 album: album,  
),  
SizedBox(  
 height: 20,  
),  
Padding(  
 padding: constEdgeInsets.symmetric(  
 vertical: 4, horizontal: 24),  
child: Text(  
"${documentData['name']}",  
style: xlboldhead,  
),  
),  
Padding(  
 padding: constEdgeInsets.symmetric(horizontal: 24),  
child: Text(  
"₹${documentData['price']}",  
style: TextStyle(  
fontSize: 30.0,  
color: Theme.*of*(context).accentColor,  
fontWeight: FontWeight.*w700*),  
),  
),  
Padding(  
 padding: constEdgeInsets.symmetric(  
 vertical: 8, horizontal: 24),  
child: Text(  
"${documentData['descr']}",  
style: TextStyle(fontSize: 16),  
),  
),  
Padding(  
 padding: constEdgeInsets.symmetric(  
 vertical: 25, horizontal: 24),  
child: Text(  
"Select Quantity",  
style: regtext,  
),  
),  
SelQuan(  
 quantity: quantiy,  
onSelQuan: (quantiy) {  
\_selQuantity= quantiy;  
calculateAmount(widget.productId, \_selQuantity,  
documentData['price']);  
},  
),  
Padding(  
 padding: constEdgeInsets.symmetric(  
 horizontal: 24, vertical: 20),  
child: Container(  
 height: 96,  
decoration: BoxDecoration(  
color: Colors.*grey*[100],  
borderRadius: BorderRadius.circular(18)),  
child: Column(  
 children: [  
Padding(  
 padding: constEdgeInsets.only(  
 left: 10, right: 10),  
child: TextField(  
 controller: feedback,  
decoration: InputDecoration(  
hintText: "Provide Feedback",  
border: InputBorder.*none*),  
),  
),  
Align(  
 alignment: Alignment.*bottomRight*,  
child: TextButton(  
onPressed: () {  
\_databaseService.submitFeedback(feedback.text, widget.productId);  
},  
child: Text("Submit")),  
)  
 ],  
),  
)),  
Padding(  
 padding: constEdgeInsets.symmetric(  
 horizontal: 16.0, vertical: 10),  
child: Row(  
mainAxisAlignment: MainAxisAlignment.start,  
children: <Widget>[  
ElevatedButton(  
onPressed: () async {  
await \_add2wishlist();  
ScaffoldMessenger.*of*(context)  
 .showSnackBar(\_add2wishlistmsg);  
},  
child: Row(  
 children: [  
Image.asset(  
"assets/images/tab\_saved.png",  
height: 22,  
),  
SizedBox(  
 width: 5,  
),  
Text(  
"Wishlist",  
style: TextStyle(  
fontSize: 15,  
fontWeight: FontWeight.*w500*,  
color: Colors.*black*),  
)  
 ],  
),  
style: ElevatedButton.*styleFrom*(  
 padding: EdgeInsets.all(15),  
primary: Colors.*grey*[300],  
shape: RoundedRectangleBorder(  
borderRadius:  
BorderRadius.circular(10))),  
),  
SizedBox(  
 width: 13,  
),  
Expanded(  
 child: ElevatedButton(  
onPressed: () async {  
await \_add2Cart();  
ScaffoldMessenger.*of*(context)  
 .showSnackBar(\_add2cartmsg);  
},  
child: Row(  
 children: [  
Icon(Icons.*shopping\_cart*),  
  
SizedBox(  
 width: 10,  
),  
Text(  
"Add to cart",  
style: TextStyle(  
fontSize: 20,  
fontWeight: FontWeight.*w600*,  
color: Colors.*white*),  
)  
 ],  
),  
style: ElevatedButton.*styleFrom*(  
 padding: EdgeInsets.all(15),  
shape: RoundedRectangleBorder(  
borderRadius:  
BorderRadius.circular(10))),  
),  
),  
SizedBox(  
 width: 5,  
),  
],  
),  
)  
 ],  
),  
);  
}  
return Scaffold(  
 body: Center(  
 child: CircularProgressIndicator(),  
),  
);  
},  
),  
TopBar(  
backArrow: true,  
hasTitle: false,  
hasBg: false,  
)  
 ],  
));  
}  
}

**detailProdAdmin.dart:**

import 'package:cloud\_firestore/cloud\_firestore.dart';  
import 'package:flutter/material.dart';  
import 'package:mumbaichimasoli/constants.dart';  
import 'package:mumbaichimasoli/services/database.dart';  
import 'package:mumbaichimasoli/widgets/imageSwipe.dart';  
import 'package:mumbaichimasoli/widgets/topBar.dart';  
import 'package:mumbaichimasoli/config/size\_config.dart';  
import 'package:fluttertoast/fluttertoast.dart';  
  
class DetailProdAdminextends StatefulWidget {  
final String productId;  
DetailProdAdmin({this.productId});  
@override  
\_DetailProdAdminStatecreateState() =>\_DetailProdAdminState();  
}  
  
class \_DetailProdAdminStateextends State<DetailProdAdmin> {  
  
DatabaseService\_databaseService= DatabaseService();  
  
showToast() {  
Fluttertoast.*showToast*(  
msg: "Data has been Updated",  
toastLength: Toast.LENGTH\_SHORT,  
gravity: ToastGravity.CENTER,  
);  
}  
  
@override  
Widget build(BuildContext context) {  
return Scaffold(  
 body: Stack(  
 children: <Widget>[  
FutureBuilder(  
 future: \_databaseService.productsReference.doc(widget.productId).get(),  
builder: (context, snapshot) {  
if (snapshot.hasError) {  
return Scaffold(  
 body: Center(  
 child: Text("Error:${snapshot.hasError}"),  
),  
);  
}  
if (snapshot.connectionState== ConnectionState.done) {  
 Map<String, dynamic>documentData = snapshot.data.data();  
  
List album = documentData['images'];  
 return ListView(  
 padding: EdgeInsets.all(0),  
children: [  
ImageSwipe(  
 album: album,  
),  
SizedBox(  
 height: 20,  
),  
Padding(  
 padding:  
constEdgeInsets.symmetric(vertical: 4, horizontal: 24),  
child: Text(  
"${documentData['name']}",  
style: xlboldhead,  
),  
),  
Padding(  
 padding: constEdgeInsets.symmetric(horizontal: 24),  
child: Text(  
"₹${documentData['price']}",  
style: TextStyle(  
fontSize: 20.0,  
color: Theme.*of*(context).accentColor,  
fontWeight: FontWeight.*w700*),  
),  
),  
Padding(  
 padding:  
constEdgeInsets.symmetric(vertical: 8, horizontal: 24),  
child: Text(  
"${documentData['descr']}",  
style: TextStyle(fontSize: 16),  
),  
),  
SizedBox(  
 height: 50,  
),  
Container(  
 margin: EdgeInsets.symmetric(vertical: 0, horizontal: 10),  
width: SizeConfig.*blockSizeHorizontal*\* 80,  
child: TextButton(  
onPressed: () {  
\_databaseService.removeProduct(widget.productId);  
showToast();  
},  
child: Text(  
"Remove Product",  
style: regtext,  
  
),  
style: TextButton.*styleFrom*(  
 padding: EdgeInsets.all(15),  
shape: RoundedRectangleBorder(  
borderRadius: BorderRadius.circular(18)),  
backgroundColor: Colors.*red*),  
),  
),  
],  
  
);  
}  
return Scaffold(  
 body: Center(  
 child: CircularProgressIndicator(),  
),  
);  
},  
),  
TopBar(  
backArrow: true,  
hasTitle: false,  
hasBg: false,  
)  
 ],  
));  
}  
}

**home\_page.dart:**

import 'package:flutter/material.dart';  
import 'package:mumbaichimasoli/tabs/HomeTab.dart';  
import 'package:mumbaichimasoli/tabs/ProfileTab.dart';  
import 'package:mumbaichimasoli/tabs/SavedTab.dart';  
import 'package:mumbaichimasoli/tabs/SearchTab.dart';  
import 'package:mumbaichimasoli/widgets/bottomTabs.dart';  
  
class HomePageextends StatefulWidget {  
@override  
\_HomePageStatecreateState() =>\_HomePageState();  
}  
  
class \_HomePageStateextends State<HomePage> {  
PageController\_bottmtbsPageController;  
int \_selectedTb= 0;  
  
@override  
void initState() {  
\_bottmtbsPageController= PageController();  
super.initState();  
}  
  
@override  
void dispose() {  
\_bottmtbsPageController.dispose();  
super.dispose();  
}  
  
@override  
Widget build(BuildContext context) {  
return Scaffold(  
 body: Column(  
mainAxisAlignment: MainAxisAlignment.spaceBetween,  
children: <Widget>[  
Expanded(  
 child: PageView(  
 controller: \_bottmtbsPageController,  
onPageChanged: (num) {  
setState(() {  
\_selectedTb= num;  
});  
},  
children: <Widget>[  
HomeTb(),  
SearchTb(),  
SavedTb(),  
ProfileTb(),  
],  
),  
),  
BottomTabs(  
selectedTab: \_selectedTb,  
taBPressed: (num) {  
\_bottmtbsPageController.animateToPage(num,  
duration: Duration(milliseconds: 400),  
curve: Curves.*easeOutCubic*);  
},  
)  
 ],  
),  
);  
}  
}

**product\_list.dart:**

import 'package:cloud\_firestore/cloud\_firestore.dart';  
import 'package:flutter/material.dart';  
import 'package:mumbaichimasoli/constants.dart';  
import 'package:mumbaichimasoli/screen/detailProdAdmin.dart';  
import 'package:mumbaichimasoli/widgets/productCard.dart';  
import 'package:mumbaichimasoli/widgets/topBar.dart';  
  
class ProductListextends StatelessWidget {  
final CollectionReference\_productsRefrence=  
FirebaseFirestore.*instance*.collection('products');  
@override  
Widget build(BuildContext context) {  
 Size screenSize() {  
return MediaQuery.*of*(context).size;  
}  
  
return Scaffold(  
appBar: AppBar(  
 title: Text('Product list'),  
centerTitle: true,  
),  
body: Stack(  
 children: <Widget>[  
FutureBuilder<QuerySnapshot>(  
 future: \_productsRefrence.get(),  
builder: (context, snapshot) {  
if (snapshot.hasError) {  
return Scaffold(  
 body: Center(  
 child: Text("Error: ${snapshot.error}"),  
),  
);  
}  
  
if (snapshot.connectionState== ConnectionState.done) {  
return ListView(  
 padding: EdgeInsets.only(top: 10.0, bottom: 20.0),  
children: snapshot.data.docs.map((document) {  
return ProductCard(  
 title: document["name"],  
imageUrl: document['images'][0],  
price: "\₹${document["price"]}",  
onPressed: () {  
Navigator.*push*(  
 context,  
MaterialPageRoute(  
 builder: (context) =>DetailProdAdmin(  
productId: document.id,  
)));  
},  
);  
}).toList());  
}  
  
return Scaffold(  
 body: Center(  
 child: CircularProgressIndicator(),  
),  
);  
},  
),  
  
],  
),  
);  
}  
}

**Register\_screen.dart:**

import 'package:flutter/material.dart';  
import 'package:flutter/services.dart';  
import 'package:google\_fonts/google\_fonts.dart';  
import 'package:mumbaichimasoli/constants.dart';  
import 'package:mumbaichimasoli/screen/SignIn.dart';  
import 'package:mumbaichimasoli/services/auth.dart';  
  
class Register extends StatefulWidget {  
@override  
\_RegisterStatecreateState() =>\_RegisterState();  
}  
  
class \_RegisterStateextends State<Register> {  
 String requestAccEmail;  
String requestfname;  
String requestlname;  
  
TextEditingControllerrequestAccCon= new TextEditingController();  
TextEditingControllerfnamecon= new TextEditingController();  
TextEditingControllerlnamecon= new TextEditingController();  
  
 final AuthService\_auth = AuthService();  
 var \_registerformKey= GlobalKey<FormState>();  
  
@override  
Widget build(BuildContext context) {  
return  
Scaffold(  
 body: AnnotatedRegion<SystemUiOverlayStyle>(  
 value: SystemUiOverlayStyle.*light*,  
child: GestureDetector(  
onTap: () =>FocusScope.*of*(context).unfocus(),  
child: Stack(  
 children: <Widget>[  
Container(  
 height: double.*infinity*,  
width: double.*infinity*,  
decoration: BoxDecoration(  
 gradient: LinearGradient(  
 begin: Alignment.*topCenter*,  
end: Alignment.*bottomCenter*,  
colors: [  
Color(0xFF73AEF5),  
Color(0xFF61A4F1),  
Color(0xFF478DE0),  
Color(0xFF398AE5),  
],  
stops: [0.1, 0.4, 0.7, 0.9],  
),  
),  
child: SingleChildScrollView(  
 physics: AlwaysScrollableScrollPhysics(),  
padding: EdgeInsets.symmetric(  
 horizontal: 35,  
vertical: 110,  
),  
child: Form(  
 key: \_registerformKey,  
child: Column(  
mainAxisAlignment: MainAxisAlignment.center,  
children: <Widget>[  
Text(  
'Request Account',  
style: GoogleFonts.*openSans*(  
fontSize: 50,  
color: Colors.*white*,  
fontWeight: FontWeight.*w600*,  
),  
),  
SizedBox(  
 height: 40,  
),  
Column(  
crossAxisAlignment: CrossAxisAlignment.start,  
children: <Widget>[  
  
SizedBox(  
 height: 10,  
),  
Container(  
 alignment: Alignment.*centerLeft*,  
decoration: BoxDecoration(  
color: Color(0xFF6CA8F1),  
borderRadius: BorderRadius.circular(10),  
boxShadow: [  
BoxShadow(  
color: Colors.*black12*,  
blurRadius: 6,  
offset: Offset(0, 2)  
 )  
 ]  
 ),  
height: 60,  
child: TextFormField(  
// textCapitalization:  
 // TextCapitalization.sentences,  
controller: fnamecon,  
validator: FirstNameValidator.*validate*,  
keyboardType: TextInputType.*name*,  
style: TextStyle(  
color: Colors.*white*,  
),  
decoration: InputDecoration(  
 border: InputBorder.*none*,  
contentPadding: EdgeInsets.only(top: 14),  
prefixIcon: Icon(Icons.*account\_circle*, color: Colors.*white*),  
hintText: "First Name",  
hintStyle: kHintTextStyle,  
),  
)  
 )  
 ],  
),  
SizedBox(height: 0),  
Column(  
crossAxisAlignment: CrossAxisAlignment.start,  
children: <Widget>[  
  
SizedBox(height: 10,),  
Container(  
 alignment: Alignment.*centerLeft*,  
decoration: BoxDecoration(  
color: Color(0xFF6CA8F1),  
borderRadius: BorderRadius.circular(10),  
boxShadow: [  
BoxShadow(  
color: Colors.*black12*,  
blurRadius: 6,  
offset: Offset(0, 2),  
)  
 ]  
 ),  
height: 60,  
child: TextFormField(  
 controller: lnamecon,  
validator: LastNameValidator.*validate*,  
keyboardType: TextInputType.*name*,  
style: TextStyle(  
color: Colors.*white*,  
),  
decoration: InputDecoration(  
 border: InputBorder.*none*,  
contentPadding:  
EdgeInsets.only(top: 14),  
prefixIcon: Icon(Icons.*account\_circle*,  
color: Colors.*white*),  
hintText: "Last name",  
hintStyle: kHintTextStyle),  
))  
 ],  
),  
Column(  
crossAxisAlignment: CrossAxisAlignment.start,  
children: <Widget>[  
SizedBox(  
 height: 10,  
),  
Container(  
 alignment: Alignment.*centerLeft*,  
decoration: BoxDecoration(  
color: Color(0xFF6CA8F1),  
borderRadius: BorderRadius.circular(10),  
boxShadow: [  
BoxShadow(  
color: Colors.*black12*,  
blurRadius: 6,  
offset: Offset(0, 2))  
 ]),  
height: 60,  
child: TextFormField(  
 controller: requestAccCon,  
validator: EmailValidator.*validate*,  
keyboardType: TextInputType.*emailAddress*,  
style: TextStyle(  
color: Colors.*white*,  
),  
decoration: InputDecoration(  
errorStyle: TextStyle(  
fontWeight: FontWeight.*w400*,  
),  
border: InputBorder.*none*,  
contentPadding:  
EdgeInsets.only(top: 14),  
prefixIcon: Icon(Icons.*mail*,  
color: Colors.*white*),  
hintText: "Email",  
hintStyle: kHintTextStyle),  
))  
 ],  
),  
SizedBox(  
 height: 10,  
),  
Container(  
 alignment: Alignment.*center*,  
child: GestureDetector(  
onTap: () {  
Navigator.*pushReplacement*(  
 context,  
MaterialPageRoute(  
 builder: (context) =>SignIn()));  
},  
child: Text(  
"Already have an account? Sign in",  
style: TextStyle(color: Colors.*white*),  
),  
)),  
SizedBox(  
 height: 12,  
),  
Container(  
 padding: EdgeInsets.symmetric(vertical: 25),  
width: double.*infinity*,  
child: ElevatedButton(  
 child: Text(  
'REQUEST',  
style: GoogleFonts.*openSans*(  
color: Color(0xFF527DAA),  
fontSize: 20,  
fontWeight: FontWeight.*bold*,  
letterSpacing: 1.5),  
),  
onPressed: () async {  
if (\_registerformKey.currentState.validate()) {  
setState(() {  
requestAccEmail= requestAccCon.text;  
requestfname= fnamecon.text;  
requestlname= lnamecon.text;  
});  
// print("request Pressed");  
print('ReqEmail :$requestAccEmail');  
print('Reqfname :$requestfname');  
print('ReqEmail :$requestlname');  
 dynamic result = await \_auth.request(requestfname, requestlname, requestAccEmail);  
 if(result != null){  
 print("error occur during requesting account");  
} else {  
 Route route = MaterialPageRoute(builder: (c) =>SignIn());  
Navigator.*pushReplacement*(context, route);  
print("Request has been registered.");  
}  
 }  
 },  
style: ElevatedButton.*styleFrom*(  
 padding: EdgeInsets.all(15),  
primary: Colors.*white*,  
shape: RoundedRectangleBorder(  
borderRadius:  
BorderRadius.circular(25))),  
))  
 ],  
),  
),  
),  
)  
 ],  
),  
),  
),  
);  
// );  
}  
}

**req\_account.dart:**

import 'package:auto\_size\_text/auto\_size\_text.dart';  
import 'package:cloud\_firestore/cloud\_firestore.dart';  
import 'package:flutter/cupertino.dart';  
import 'package:flutter/material.dart';  
import 'package:mumbaichimasoli/constants.dart';  
import 'package:mumbaichimasoli/services/auth.dart';  
import 'package:mumbaichimasoli/services/database.dart';  
import 'package:mumbaichimasoli/widgets/topBar.dart';  
import 'package:fluttertoast/fluttertoast.dart';  
  
  
class Requsetedaccountextends StatefulWidget {  
  
  
@override  
\_State createState() =>\_State();  
}  
  
class \_State extends State<Requsetedaccount> {  
DatabaseService\_databaseService= DatabaseService();  
AuthService\_auth = AuthService();  
  
 final String DefaultPass= "pass123";  
  
showToast() {  
Fluttertoast.*showToast*(  
msg: "Data has been Updated",  
toastLength: Toast.LENGTH\_SHORT,  
gravity: ToastGravity.CENTER,  
);  
}  
  
@override  
Widget build(BuildContext context) {  
return Scaffold(  
appBar: AppBar(  
 title: Text("Requested Accounts"),  
),  
  
body: Stack(  
 children: <Widget>[  
FutureBuilder<QuerySnapshot>(  
 future: \_databaseService.requestCollection.get(),  
// ignore: missing\_return  
builder: (context, snapshot) {  
if (snapshot.hasError) {  
return Scaffold(  
  
 body: Center(  
 child: Text("Error : ${snapshot.error}"),  
),  
);  
}  
if(!snapshot.hasData){  
return Center(  
 child: Container(child: Text("No requested Accounts"),),  
);  
}  
else if (snapshot.connectionState== ConnectionState.done) {  
return ListView(  
 padding: EdgeInsets.only(top: 50.0, bottom: 80.0),  
children: snapshot.data.docs.map((document) {  
return Padding(  
 padding: EdgeInsets.symmetric(vertical:10,horizontal: 10),  
child: Container(  
// height: 10,  
decoration: BoxDecoration(  
color:Colors.*blue*,  
borderRadius: BorderRadius.circular(11)  
 ),  
child: Stack(  
 children: [  
Column(  
mainAxisAlignment: MainAxisAlignment.start,  
children:[  
Padding(  
 padding: constEdgeInsets.symmetric(vertical:15,horizontal: 18),  
child: Row(  
  
 children:[  
  
Text("${document["fname"]}",style: TextStyle(fontSize: 18, fontWeight: FontWeight.*bold*, color: Colors.*white*),),  
SizedBox(width:10),  
Text("${document["lname"]}",style: TextStyle(fontSize: 18, fontWeight: FontWeight.*bold*, color: Colors.*white*),)  
  
  
 ]  
 ),  
),  
  
Padding(  
 padding: constEdgeInsets.symmetric(horizontal:18.0),  
child: Row(  
  
 children: [  
Text("${document["email"]}",style: TextStyle(fontSize: 18, fontWeight: FontWeight.*bold*, color: Colors.*white*),),  
],  
),  
)  
 ]  
 ),  
Padding(  
 padding: constEdgeInsets.symmetric(horizontal:15,vertical:15),  
child: Align(  
 alignment: Alignment.*topRight*,  
child: Column(  
 children: [  
IconButton(  
onPressed: () {  
\_databaseService.cancelRequest(document.id);  
showToast();  
},  
icon: constIcon(Icons.*delete*),  
color: Colors.*white*,  
),  
SizedBox(  
 height:25  
),  
Container(  
 height: 20,  
width: 90,  
  
decoration: BoxDecoration(  
  
borderRadius: BorderRadius.circular(5),  
color: Colors.*blue*,  
boxShadow: [  
BoxShadow(  
color: Colors.*black*,  
offset: Offset(0.0, 1.0), //(x,y)  
blurRadius: 1.0,  
),  
],  
),  
child: GestureDetector(  
onTap: () {  
\_auth.register(document['fname'], document['lname'], document['email'], DefaultPass);  
\_databaseService.cancelRequest(document.id);  
showToast();  
},  
child: AutoSizeText(  
"Set Password",style: TextStyle(color:Colors.*white*), maxLines: 1,),  
),  
)  
 ],  
),  
),  
)  
 ],  
),  
),  
);  
  
  
}).toList()  
 );  
}  
 }  
 ),  
  
]  
  
 ),  
);  
}  
}

**reset\_password.dart:**

import 'package:flutter/material.dart';  
import 'package:flutter/services.dart';  
import 'package:google\_fonts/google\_fonts.dart';  
import 'package:mumbaichimasoli/constants.dart';  
import 'package:mumbaichimasoli/services/auth.dart';  
  
class ResetPassextends StatefulWidget {  
@override  
\_ResetPassStatecreateState() =>\_ResetPassState();  
}  
  
class \_ResetPassStateextends State<ResetPass> {  
  
 String email;  
  
 var \_forgetformKey= GlobalKey<FormState>();  
  
 final AuthService\_auth = AuthService();  
TextEditingControllerforgetPassEmailcon= new TextEditingController();  
  
@override  
Widget build(BuildContext context) {  
return  
Scaffold(  
 body: AnnotatedRegion<SystemUiOverlayStyle>(  
 value: SystemUiOverlayStyle.*light*,  
child: GestureDetector(  
onTap: () =>FocusScope.*of*(context).unfocus(),  
child: Stack(  
 children: <Widget>[  
Container(  
 height: double.*infinity*,  
width: double.*infinity*,  
decoration: BoxDecoration(  
 gradient: LinearGradient(  
 begin: Alignment.*topCenter*,  
end: Alignment.*bottomCenter*,  
colors: [  
Color(0xFF73AEF5),  
Color(0xFF61A4F1),  
Color(0xFF478DE0),  
Color(0xFF398AE5),  
],  
stops: [0.1, 0.4, 0.7, 0.9],  
)),  
),  
Container(  
 height: double.*infinity*,  
child: SingleChildScrollView(  
 physics: AlwaysScrollableScrollPhysics(),  
padding: EdgeInsets.symmetric(horizontal: 20, vertical: 90),  
child: Column(  
mainAxisAlignment: MainAxisAlignment.start,  
children: <Widget>[  
Text(  
"Forgot Password, Don't Worry",  
style: GoogleFonts.*openSans*(  
fontSize: 35,  
color: Colors.*white*,  
fontWeight: FontWeight.*w600*,  
),  
),  
Align(  
 alignment: Alignment.*topLeft*,  
child: Text(  
"Just Enter your email below",  
style: TextStyle(  
color: Colors.*white60*, fontSize: 13,  
),  
)  
 ),  
SizedBox(  
 height: 20,  
),  
Form(  
 key: \_forgetformKey,  
child: Column(  
crossAxisAlignment: CrossAxisAlignment.start,  
children: <Widget>[  
SizedBox(  
 height: 10,  
),  
Container(  
 alignment: Alignment.*centerLeft*,  
decoration: BoxDecoration(  
color: Color(0xFF6CA8F1),  
borderRadius: BorderRadius.circular(10),  
boxShadow: [  
BoxShadow(  
color: Colors.*black12*,  
blurRadius: 6,  
offset: Offset(0, 2))  
 ]),  
height: 60,  
child: TextFormField(  
 controller: forgetPassEmailcon,  
validator: EmailValidator.*validate*,  
keyboardType: TextInputType.*emailAddress*,  
style: TextStyle(  
color: Colors.*white*,  
),  
decoration: InputDecoration(  
 border: InputBorder.*none*,  
contentPadding:  
EdgeInsets.only(top: 14),  
prefixIcon: Icon(Icons.*mail*,  
color: Colors.*white*),  
hintText: "Email",  
hintStyle: kHintTextStyle),  
))  
 ],  
),  
),  
Container(  
 padding: EdgeInsets.symmetric(vertical: 25),  
width: double.*infinity*,  
child: ElevatedButton(  
 child: Text(  
"Reset Password",  
style: GoogleFonts.*openSans*(  
color: Color(0xFF527DAA),  
fontSize: 20,  
fontWeight: FontWeight.*bold*,  
),  
),  
style: ElevatedButton.*styleFrom*(  
 padding: EdgeInsets.all(15),  
primary: Colors.*white*,  
shape: RoundedRectangleBorder(  
borderRadius: BorderRadius.circular(30))),  
onPressed: () {  
if(\_forgetformKey.currentState.validate()){  
setState(() {  
email = forgetPassEmailcon.text;  
});  
\_auth.sendPasswordResetEmail(email);  
Navigator.*of*(context).pop();  
}   
 }   
 ),  
)  
 ],  
),  
),  
),  
],  
),  
),  
),  
);  
// );  
}  
}

**shippingAdd.dart:**

import 'package:flutter/material.dart';  
import 'package:mumbaichimasoli/config/size\_config.dart';  
import 'package:mumbaichimasoli/constants.dart';  
import 'package:mumbaichimasoli/services/database.dart';  
import 'package:mumbaichimasoli/widgets/topBar.dart';  
  
class ShippingAddextends StatelessWidget {  
 String shippingAdd;  
  
TextEditingControlleraddress = new TextEditingController();  
TextEditingControllerpincode= new TextEditingController();  
TextEditingControllercity = new TextEditingController();  
TextEditingControllerstate = new TextEditingController();  
  
DatabaseService\_databaseService= DatabaseService();  
  
@override  
Widget build(BuildContext context) {  
SizeConfig().init(context);  
  
 return Scaffold(  
 body: SingleChildScrollView(  
 child: Column(  
 children: [  
TopBar(  
 title: "Shipping Address",  
hasCart: false,  
),  
Padding(  
 padding: constEdgeInsets.symmetric(horizontal: 20, vertical: 30),  
child: Column(  
 children: [  
Center(  
 child: Container(  
 height: 100,  
decoration: BoxDecoration(  
color: Colors.*blue*[100],  
borderRadius: BorderRadius.circular(18)),  
width: SizeConfig.*blockSizeHorizontal*\* 80,  
child: TextField(  
 controller: address,  
maxLines: null,  
keyboardType: TextInputType.*multiline*,  
decoration: InputDecoration(  
 border: InputBorder.*none*,  
hintText: "Address",  
prefixIcon: Icon(  
Icons.*landscape*,  
)),  
obscureText: false,  
// keyboardType: TextInputType.text,  
),  
)),  
SizedBox(  
 height: 10,  
),  
Center(  
 child: Container(  
 decoration: BoxDecoration(  
color: Colors.*blue*[100],  
borderRadius: BorderRadius.circular(18)),  
width: SizeConfig.*blockSizeHorizontal*\* 80,  
child: Center(  
 child: TextField(  
 controller: pincode,  
decoration: InputDecoration(  
 border: InputBorder.*none*,  
hintText: "Pincode",  
prefixIcon: Icon(Icons.*location\_city*)),  
obscureText: false,  
keyboardType: TextInputType.*number*,  
),  
),  
),  
),  
SizedBox(  
 height: 10,  
),  
Center(  
 child: Container(  
 decoration: BoxDecoration(  
color: Colors.*blue*[100],  
borderRadius: BorderRadius.circular(18)),  
width: SizeConfig.*blockSizeHorizontal*\* 80,  
child: Center(  
 child: TextField(  
 controller: city,  
decoration: InputDecoration(  
 border: InputBorder.*none*,  
hintText: "City",  
prefixIcon: Icon(Icons.*location\_city*)),  
obscureText: false,  
keyboardType: TextInputType.*text*,  
),  
),  
),  
),  
SizedBox(  
 height: 10,  
),  
Center(  
 child: Container(  
 decoration: BoxDecoration(  
color: Colors.*blue*[100],  
borderRadius: BorderRadius.circular(18)),  
width: SizeConfig.*blockSizeHorizontal*\* 80,  
child: Center(  
 child: TextField(  
 controller: state,  
decoration: InputDecoration(  
 border: InputBorder.*none*,  
hintText: "State",  
prefixIcon: Icon(Icons.*map*)),  
obscureText: false,  
keyboardType: TextInputType.*text*,  
),  
),  
),  
),  
SizedBox(  
 height: 50,  
),  
Container(  
 width: SizeConfig.*blockSizeHorizontal*\* 80,  
child: TextButton(  
onPressed: () {  
shippingAdd= (address.text+ ", " +  
pincode.text+ ", " +  
city.text+ ", " +  
state.text);  
print(shippingAdd);  
\_databaseService.updateAddress(shippingAdd);  
Navigator.*pop*(context);  
},  
child: Text(  
"Submit",  
style: regtext,  
),  
style: TextButton.*styleFrom*(  
 padding: EdgeInsets.all(15),  
shape: RoundedRectangleBorder(  
borderRadius: BorderRadius.circular(18)),  
backgroundColor: Colors.*blue*[50]),  
),  
),  
SizedBox(  
 height: 10,  
  
),  
  
Container(  
 width: SizeConfig.*blockSizeHorizontal*\* 80,  
child: TextButton(  
onPressed: () {  
Navigator.*pop*(context);  
},  
child: Text(  
"Cancel",  
style: regtext,  
),  
style: TextButton.*styleFrom*(  
 padding: EdgeInsets.all(15),  
shape: RoundedRectangleBorder(  
borderRadius: BorderRadius.circular(18)),  
backgroundColor: Colors.*blue*[50]),  
),  
),  
],  
),  
),  
],  
),  
));  
}  
}

**SignIn.dart:**

import 'package:auto\_size\_text/auto\_size\_text.dart';  
import 'package:flutter/material.dart';  
import 'package:flutter/services.dart';  
import 'package:google\_fonts/google\_fonts.dart';  
import 'package:mumbaichimasoli/screen/Register\_screen.dart';  
import 'package:mumbaichimasoli/screen/admin\_login.dart';  
import 'package:mumbaichimasoli/screen/home\_page.dart';  
import 'package:mumbaichimasoli/screen/reset\_password.dart';  
import 'package:mumbaichimasoli/services/auth.dart';  
import 'package:mumbaichimasoli/shared/loading.dart';  
import '../constants.dart';  
  
class SignInextends StatefulWidget {  
@override  
\_SignInStatecreateState() =>\_SignInState();  
}  
  
class \_SignInStateextends State<SignIn> {  
  
void \_togglePasswordView() {  
setState(() {  
isHiddenPassword= !isHiddenPassword;  
});  
}  
  
 bool isHiddenPassword= true;  
bool loading = false;  
String signinEmail;  
String signinPass;  
String error;  
  
 final AuthService\_auth = AuthService();  
 var \_signinformKey= GlobalKey<FormState>();  
  
TextEditingControllersignInEmailcon= new TextEditingController();  
TextEditingControllersignInPasscon= new TextEditingController();  
  
  
Widget showAlert() {  
if (error != null) {  
 print("inside if statement");  
 return Container(  
color: Colors.*amberAccent*,  
width: double.*infinity*,  
padding: EdgeInsets.all(8.0),  
child: Row(  
 children: <Widget>[  
Padding(  
 padding: constEdgeInsets.only(right: 8.0),  
child: Icon(Icons.*error*),  
),  
Expanded(  
 child: AutoSizeText(  
error,  
maxLines: 3,  
)  
 ),  
Padding(  
 padding: constEdgeInsets.only(left: 8.0),  
child: IconButton(  
 icon: Icon(Icons.*close*),  
onPressed: () {  
setState(() {  
error = null;  
});  
}  
 ),  
)  
 ],  
),  
);  
}  
return SizedBox(height: 8,);  
}  
  
@override  
Widget build(BuildContext context) {  
return loading ? Loading() : /\*new WillPopScope(  
onWillPop: () async => false,  
 child: \*/  
Scaffold(  
 body: AnnotatedRegion<SystemUiOverlayStyle>(  
 value: SystemUiOverlayStyle.*light*,  
sized: false,  
child: GestureDetector(  
onTap: () =>FocusScope.*of*(context).unfocus(),  
child: Stack(  
 children: <Widget>[  
Container(  
 height: double.*infinity*,  
width: double.*infinity*,  
decoration: BoxDecoration(  
 gradient: LinearGradient(  
 begin: Alignment.*topCenter*,  
end: Alignment.*bottomCenter*,  
colors: [  
Color(0xFF73AEF5),  
Color(0xFF61A4F1),  
Color(0xFF478DE0),  
Color(0xFF398AE5),  
],  
stops: [0.1, 0.4, 0.7, 0.9],  
),  
),  
child: SingleChildScrollView(  
 physics: AlwaysScrollableScrollPhysics(),  
padding: EdgeInsets.symmetric(  
 horizontal: 40,  
vertical: 110,  
),  
child: Form(  
 key: \_signinformKey,  
child: Column(  
mainAxisAlignment: MainAxisAlignment.center,  
children: <Widget>[  
SizedBox(  
 height: 10,  
),  
showAlert(),  
SizedBox(height: 20),  
Align(  
 alignment: Alignment.*topLeft*,  
child: Text(  
'Sign In',  
style: GoogleFonts.*openSans*(  
fontSize: 50,  
color: Colors.*white*,  
fontWeight: FontWeight.*w600*),  
),  
),  
SizedBox(  
 height: 50,  
),  
Column(  
crossAxisAlignment: CrossAxisAlignment.start,  
children: <Widget>[  
SizedBox(  
 height: 10,  
),  
Container(  
 alignment: Alignment.*centerLeft*,  
decoration: BoxDecoration(  
color: Color(0xFF6CA8F1),  
borderRadius: BorderRadius.circular(10),  
boxShadow: [  
BoxShadow(  
color: Colors.*black12*,  
blurRadius: 6,  
offset: Offset(0, 2)  
 )  
 ]  
 ),  
height: 60,  
child: TextFormField(  
 controller: signInEmailcon,  
validator: EmailValidator.*validate*,  
keyboardType:  
TextInputType.*emailAddress*,  
style: TextStyle(  
color: Colors.*white*,  
),  
decoration: InputDecoration(  
 border: InputBorder.*none*,  
contentPadding: EdgeInsets.only(top: 14),  
prefixIcon: Icon(  
Icons.*mail*,  
color: Colors.*white*),  
hintText: "Email",  
hintStyle: kHintTextStyle  
 ),  
)  
 )  
 ],  
),  
SizedBox(  
 height: 30,  
),  
Column(  
crossAxisAlignment: CrossAxisAlignment.start,  
children: <Widget>[  
SizedBox(  
 height: 10,  
),  
Container(  
 alignment: Alignment.*centerLeft*,  
decoration: BoxDecoration(  
color: Color(0xFF6CA8F1),  
borderRadius:  
BorderRadius.circular(10),  
boxShadow: [  
BoxShadow(  
color: Colors.*black12*,  
blurRadius: 6,  
offset: Offset(0, 2)  
 )  
 ]  
 ),  
height: 60,  
child: TextFormField(  
 controller: signInPasscon,  
obscureText: isHiddenPassword,  
style: TextStyle(  
color: Colors.*white*,  
),  
decoration: InputDecoration(  
 border: InputBorder.*none*,  
contentPadding:  
EdgeInsets.only(top: 14),  
prefixIcon: Icon(Icons.*lock*,  
color: Colors.*white*),  
hintText: "Password",  
hintStyle: kHintTextStyle,  
suffixIcon: InkWell(  
onTap: \_togglePasswordView,  
child: Icon(  
isHiddenPassword? Icons.*visibility*: Icons.*visibility\_off*,  
color: Colors.*white*,  
),  
)  
 ),  
)  
 )  
 ],  
),  
SizedBox(  
 height: 12,  
),  
Container(  
 alignment: Alignment.*centerRight*,  
child: TextButton(  
onPressed: () =>Navigator.*push*(  
 context,  
MaterialPageRoute(builder: (context) =>ResetPass())  
 ),  
child: Text(  
"Forgot Password?",  
style: kLabelStyle,  
),  
style: TextButton.*styleFrom*(  
 padding: EdgeInsets.only(right: 0)  
 )  
 ),  
),  
Container(  
 padding: EdgeInsets.symmetric(vertical: 25),  
width: double.*infinity*,  
child: ElevatedButton(  
 child: Text(  
'LOGIN',  
style: GoogleFonts.*openSans*(  
color: Color(0xFF527DAA),  
fontSize: 20,  
fontWeight: FontWeight.*bold*,  
letterSpacing: 1.5,  
),  
),  
onPressed: () async {  
if (\_signinformKey.currentState.validate()) {  
setState(() {  
signinEmail= signInEmailcon.text;  
signinPass= signInPasscon.text;  
loading = true;  
print('SignInEmail: $signinEmail');  
print('SignInPass: $signinPass');  
});  
 dynamic result = await \_auth.signInWithEmailAndPassword(signinEmail, signinPass);  
 if (result != null) {  
Navigator.*pushReplacement*(  
 context,  
MaterialPageRoute(builder: (context) =>HomePage()),  
);  
  
} else {  
 result = await \_auth.returnError();  
setState(() {  
error = result;  
loading = false;  
});  
print('error: ' + error);  
}  
 }  
 },  
style: ElevatedButton.*styleFrom*(  
 padding: EdgeInsets.all(15),  
primary: Colors.*white*,  
shape: RoundedRectangleBorder(  
borderRadius: BorderRadius.circular(30)  
 )  
 ),  
)  
 ),  
GestureDetector(  
onTap: () {  
Navigator.*push*(  
 context,  
MaterialPageRoute(builder: (context) =>Register())  
 );  
},   
child: RichText(  
 text: TextSpan(children: [  
TextSpan(  
 text: 'Don\'t have an Account?',  
style: TextStyle(  
color: Colors.*white*,  
fontSize: 15,  
fontWeight: FontWeight.*w500*,  
)  
 ),  
TextSpan(  
 text: ' Request Account',  
style: TextStyle(  
color: Colors.*white*,  
fontSize: 14,  
fontWeight: FontWeight.*w500*)  
 )  
 ])),  
),  
SizedBox(height: 240,),  
Container(  
 child: Row(  
 children: [  
Icon(  
Icons.*admin\_panel\_settings*,  
color: Colors.*white*,  
),  
GestureDetector(  
onTap: () {  
Navigator.*pushReplacement*(  
 context,  
MaterialPageRoute(builder: (context) =>AdminLoginScreen())  
 );  
},  
child: Text(  
'I am Admin',  
style: kLabelStyle,  
),  
),  
],  
),  
)  
 ],  
),  
),  
),  
),  
],  
),  
),  
),  
);  
// );  
}  
}

**SignUp.dart:**

import 'package:flutter/material.dart';  
import 'package:flutter/services.dart';  
import 'package:google\_fonts/google\_fonts.dart';  
import 'package:mumbaichimasoli/constants.dart';  
import 'package:mumbaichimasoli/screen/SignIn.dart';  
import 'package:mumbaichimasoli/services/auth.dart';  
  
class Signup extends StatefulWidget {  
@override  
\_SignupStatecreateState() =>\_SignupState();  
}  
  
class \_SignupStateextends State<Signup> {  
final AuthService\_auth = AuthService();  
bool isHiddenPassword= true;  
 var \_signupformKey= GlobalKey<FormState>();  
  
 void \_togglePasswordView() {  
setState(() {  
isHiddenPassword= !isHiddenPassword;  
});  
}  
  
 String signupEmail;  
String signupPass;  
String confirmPass;  
String firstName;  
String lastName;  
String error = "";  
  
TextEditingControllersignUpEmailcon= new TextEditingController();  
TextEditingControllersignUpPasscon= new TextEditingController();  
TextEditingControllerconfirmPasscon= new TextEditingController();  
TextEditingControllernamecon= new TextEditingController();  
TextEditingControllerlastnamecon= new TextEditingController();  
  
Widget showAlert() {  
if (error != null) {  
return Container(  
color: Colors.*amberAccent*,  
width: double.*infinity*,  
padding: EdgeInsets.all(8.0),  
child: Row(  
 children: <Widget>[  
Padding(  
 padding: constEdgeInsets.only(right: 8.0),  
child: Icon(Icons.*error\_outline*),  
),  
Expanded(child: Text(error)),  
Padding(  
 padding: constEdgeInsets.only(left: 8.0),  
child: IconButton(  
 icon: Icon(Icons.*close*),  
onPressed: () {  
setState(() {  
error = null;  
});  
},  
),  
),  
],  
),  
);  
}  
return Container(  
color: Colors.*amberAccent*,  
width: double.*infinity*,  
padding: EdgeInsets.all(8.0),  
);  
}  
  
@override  
Widget build(BuildContext context) {  
return Scaffold(  
 body: AnnotatedRegion<SystemUiOverlayStyle>(  
 value: SystemUiOverlayStyle.*light*,  
child: GestureDetector(  
onTap: () =>FocusScope.*of*(context).unfocus(),  
child: Stack(  
 children: <Widget>[  
SizedBox(height: 20),  
showAlert(),  
SizedBox(height: 20),  
Container(  
 height: double.*infinity*,  
width: double.*infinity*,  
decoration: BoxDecoration(  
 gradient: LinearGradient(  
 begin: Alignment.*topCenter*,  
end: Alignment.*bottomCenter*,  
colors: [  
Color(0xFF73AEF5),  
Color(0xFF61A4F1),  
Color(0xFF478DE0),  
Color(0xFF398AE5),  
],  
stops: [0.1, 0.4, 0.7, 0.9],  
),  
),  
child: SingleChildScrollView(  
 physics: AlwaysScrollableScrollPhysics(),  
padding: EdgeInsets.symmetric(  
 horizontal: 40,  
vertical: 75,  
),  
child: Form(  
 key: \_signupformKey,  
child: Column(  
mainAxisAlignment: MainAxisAlignment.center,  
children: <Widget>[  
Align(  
 alignment: Alignment.*topLeft*,  
child: Text(  
'CREATE ACCOUNT',  
style: GoogleFonts.*openSans*(  
fontSize: 45,  
color: Colors.*white*,  
fontWeight: FontWeight.*w600*),  
),  
),  
SizedBox(  
 height: 30,  
),  
Column(  
crossAxisAlignment: CrossAxisAlignment.start,  
children: <Widget>[  
  
SizedBox(  
 height: 10,  
),  
Container(  
 alignment: Alignment.*centerLeft*,  
decoration: BoxDecoration(  
color: Color(0xFF6CA8F1),  
borderRadius: BorderRadius.circular(10),  
boxShadow: [  
BoxShadow(  
color: Colors.*black12*,  
blurRadius: 6,  
offset: Offset(0, 2))  
 ]),  
height: 60,  
child: TextFormField(  
 controller: namecon,  
validator: FirstNameValidator.*validate*,  
  
keyboardType: TextInputType.*emailAddress*,  
style: TextStyle(  
color: Colors.*white*,  
),  
decoration: InputDecoration(  
 border: InputBorder.*none*,  
contentPadding:  
EdgeInsets.only(top: 14),  
prefixIcon: Icon(Icons.*account\_circle*,  
color: Colors.*white*),  
hintText: "First Name",  
hintStyle: kHintTextStyle),  
))  
 ],  
),  
SizedBox(  
 height: 20,  
),  
Column(  
crossAxisAlignment: CrossAxisAlignment.start,  
children: <Widget>[  
SizedBox(  
 height: 10,  
),  
Container(  
 alignment: Alignment.*centerLeft*,  
decoration: BoxDecoration(  
color: Color(0xFF6CA8F1),  
borderRadius: BorderRadius.circular(10),  
boxShadow: [  
BoxShadow(  
color: Colors.*black12*,  
blurRadius: 6,  
offset: Offset(0, 2))  
 ]),  
height: 60,  
child: TextFormField(  
 controller: lastnamecon,  
validator: LastNameValidator.*validate*,  
  
keyboardType: TextInputType.*emailAddress*,  
style: TextStyle(  
color: Colors.*white*,  
),  
decoration: InputDecoration(  
 border: InputBorder.*none*,  
contentPadding:  
EdgeInsets.only(top: 14),  
prefixIcon: Icon(Icons.*account\_circle*,  
color: Colors.*white*),  
hintText: "Last name",  
hintStyle: kHintTextStyle),  
))  
 ],  
),  
SizedBox(  
 height: 20,  
),  
Column(  
crossAxisAlignment: CrossAxisAlignment.start,  
children: <Widget>[  
SizedBox(  
 height: 10,  
),  
Container(  
 alignment: Alignment.*centerLeft*,  
decoration: BoxDecoration(  
color: Color(0xFF6CA8F1),  
borderRadius: BorderRadius.circular(10),  
boxShadow: [  
BoxShadow(  
color: Colors.*black12*,  
blurRadius: 6,  
offset: Offset(0, 2))  
 ]),  
height: 60,  
child: TextFormField(  
 controller: signUpEmailcon,  
validator: EmailValidator.*validate*,  
  
keyboardType: TextInputType.*emailAddress*,  
style: TextStyle(  
color: Colors.*white*,  
),  
decoration: InputDecoration(  
 border: InputBorder.*none*,  
contentPadding:  
EdgeInsets.only(top: 14),  
prefixIcon: Icon(Icons.*mail*,  
color: Colors.*white*),  
hintText: "Email",  
hintStyle: kHintTextStyle),  
))  
 ],  
),  
SizedBox(  
 height: 20,  
),  
Column(  
crossAxisAlignment: CrossAxisAlignment.start,  
children: <Widget>[  
SizedBox(  
 height: 10,  
),  
Container(  
 alignment: Alignment.*centerLeft*,  
decoration: BoxDecoration(  
color: Color(0xFF6CA8F1),  
borderRadius: BorderRadius.circular(10),  
boxShadow: [  
BoxShadow(  
color: Colors.*black12*,  
blurRadius: 6,  
offset: Offset(0, 2))  
 ]),  
height: 60,  
child: TextFormField(  
 controller: signUpPasscon,  
validator: PasswordValidator.*validate*,  
obscureText: isHiddenPassword,  
style: TextStyle(  
color: Colors.*white*,  
),  
decoration: InputDecoration(  
 border: InputBorder.*none*,  
contentPadding:  
EdgeInsets.only(top: 14),  
prefixIcon: Icon(Icons.*lock*,  
color: Colors.*white*),  
hintText: "Password",  
hintStyle: kHintTextStyle,  
suffixIcon: InkWell(  
onTap: \_togglePasswordView,  
child: Icon(  
isHiddenPassword  
? Icons.*visibility*: Icons.*visibility\_off*,  
color: Colors.*white*,  
),  
),  
),  
))  
 ],  
),  
SizedBox(  
 height: 10,  
),  
Container(  
 padding: EdgeInsets.symmetric(vertical: 25),  
width: double.*infinity*,  
child: ElevatedButton(  
 child: Text(  
'SIGN UP',  
style: GoogleFonts.*openSans*(  
color: Color(0xFF527DAA),  
fontSize: 20,  
fontWeight: FontWeight.*bold*,  
letterSpacing: 1.5,  
),  
),  
onPressed: () async {  
if (\_signupformKey.currentState.validate()) {  
// print("Pressed");  
setState(() {  
signupEmail= signUpEmailcon.text;  
signupPass= signUpPasscon.text;  
confirmPass= confirmPasscon.text;  
firstName= namecon.text;  
lastName= lastnamecon.text;  
  
print('SignUpEmail: $signupEmail');  
print('SignUpPass: $signupPass');  
print('ConfirmPass: $confirmPass');  
print('First Name: $firstName');  
print('Last Name: $lastName');  
});  
 dynamic result = await \_auth.register(firstName, lastName, signupEmail, signupPass);  
 if (result == null) {  
 result = await \_auth.returnError();  
setState(() {  
error = result;  
});  
}  
 print("error: " + error);  
}  
 },  
style: ElevatedButton.*styleFrom*(  
 padding: EdgeInsets.all(15),  
primary: Colors.*white*,  
shape: RoundedRectangleBorder(  
borderRadius:  
BorderRadius.circular(30))),  
)),  
GestureDetector(  
onTap: () {  
Navigator.*pushReplacement*(  
 context,  
MaterialPageRoute(  
 builder: (context) =>SignIn()));  
print('Sign up pressed');  
},  
child: RichText(  
 text: TextSpan(children: [  
TextSpan(  
 text: 'Already have an Account?',  
style: kLabelStyle),  
TextSpan(text: ' Sign in', style: kLabelStyle)  
 ])),  
)  
 ],  
),  
),  
),  
)  
 ],  
),  
),  
),  
);  
}  
}

**splash\_screen.dart:**

import 'package:flutter/material.dart';  
  
import 'package:mumbaichimasoli/screen/wrapper.dart';  
import 'package:splashscreen/splashscreen.dart';  
  
  
class SplashScreenWidgetextends StatelessWidget {  
@override  
Widget build(BuildContext context) {  
return SplashScreen(  
 seconds: 4,  
navigateAfterSeconds: Wrapper(),  
title: Text(  
'मुंबईचीमासोळी',  
style: TextStyle(  
color: Colors.*blue*,  
fontSize: 35,  
fontWeight: FontWeight.*bold*),  
),  
image: Image.asset(  
'assets/images2/masoli.png',  
),  
photoSize: 100.0,  
loaderColor: Colors.*blue*,  
);  
}  
}

**VerifyEmail.dart:**

import 'package:auto\_size\_text/auto\_size\_text.dart';  
import 'package:cloud\_firestore/cloud\_firestore.dart';  
import 'package:email\_auth/email\_auth.dart';  
import 'package:flutter/material.dart';  
import 'package:flutter/services.dart';  
import 'package:fluttertoast/fluttertoast.dart';  
import 'package:google\_fonts/google\_fonts.dart';  
import 'admin\_changePass.dart';  
  
  
class VerifyEmailPageextends StatefulWidget {  
@override  
\_VerifyEmailPageStatecreateState() =>\_VerifyEmailPageState();  
}  
  
class \_VerifyEmailPageStateextends State<VerifyEmailPage> {  
  
 String error;  
bool verified = false;  
  
TextEditingController\_emailController= new TextEditingController();  
TextEditingController\_otpController= new TextEditingController();  
  
 var \_verifyformKey= GlobalKey<FormState>();  
  
Widget showAlert() {  
if (error != null) {  
return Container(  
color: Colors.*amberAccent*,  
width: double.*infinity*,  
padding: EdgeInsets.all(8.0),  
child: Row(  
 children: <Widget>[  
Padding(  
 padding: constEdgeInsets.only(right: 8.0),  
child: Icon(Icons.*error*),  
),  
Expanded(  
 child: AutoSizeText(  
error,  
maxLines: 3,  
)  
 ),  
Padding(  
 padding: constEdgeInsets.only(left: 8.0),  
child: IconButton(  
 icon: Icon(Icons.*close*),  
onPressed: () {  
setState(() {  
error = null;  
});  
}  
 ),  
)  
 ],  
),  
);  
}  
return SizedBox(  
 height: 8,  
);  
}  
  
showToast() {  
Fluttertoast.*showToast*(  
msg: "OTP has been sent successfully",  
toastLength: Toast.LENGTH\_SHORT,  
gravity: ToastGravity.CENTER,  
);  
}  
  
sendOTP() async {  
EmailAuth.*sessionName*="Test Session";  
 var response = await EmailAuth.*sendOtp*(receiverMail: \_emailController.text);  
 if (response){  
 print("OTP Sent");  
showToast();  
 return true;  
}else {  
setState(() {  
error = "An error occured. Try again later.";  
});  
print("An error occured. Try again later.");  
}  
 }  
  
  
verifyOTP() async {  
var response = EmailAuth.*validate*(receiverMail: \_emailController.text, userOTP: \_otpController.text);  
 if (response){  
 print("OTP Verified");  
setState(() {  
verified = true;  
});  
}else {  
setState(() {  
error = "Invalid OTP";  
});  
print("Invalid OTP");  
}  
 }  
  
checkEmail(email) {  
 FirebaseFirestore.*instance*.collection("admin").get().then((snapshot) {  
snapshot.docs.forEach((result) {   
if(result.data()["email"] != email) {  
setState(() {  
error = "This email is not the registered email";  
});  
 return false;  
}else {  
return true;  
}  
 });  
});  
}  
  
static String *emailValidate*(value) {  
return RegExp(r"^[a-zA-Z0-9.a-zA-Z0-9.!#$%&'\*+-/=?^\_`{|}~]+@[a-zA-Z0-9]+\.[a-zA-Z]+").hasMatch(value) ?   
null :   
" Enter a valid Email";  
}  
  
@override  
Widget build(BuildContext context) {  
return Scaffold(  
 body: AnnotatedRegion<SystemUiOverlayStyle>(  
 value: SystemUiOverlayStyle.*light*,  
child: GestureDetector(  
onTap: () =>FocusScope.*of*(context).unfocus(),  
child: Stack(  
 children: <Widget>[  
Container(  
 height: double.*infinity*,  
width: double.*infinity*,  
decoration: BoxDecoration(  
 gradient: LinearGradient(  
 begin: Alignment.*topCenter*,  
end: Alignment.*bottomCenter*,  
colors: [  
Color(0xFF73AEF5),  
Color(0xFF61A4F1),  
Color(0xFF478DE0),  
Color(0xFF398AE5),  
],  
stops: [0.1, 0.4, 0.7, 0.9],  
),  
),  
child: SingleChildScrollView(  
 physics: AlwaysScrollableScrollPhysics(),  
padding: EdgeInsets.symmetric(  
 horizontal: 40,  
vertical: 110,  
),  
child: Form(  
 key: \_verifyformKey,  
child: Column(  
mainAxisAlignment: MainAxisAlignment.center,  
children: <Widget>[  
SizedBox(height: 10,),  
showAlert(),  
SizedBox(height: 20,),  
Align(  
 alignment: Alignment.*topLeft*,  
child: Text(  
'Verify Email',  
style: GoogleFonts.*openSans*(  
fontSize: 50,  
color: Colors.*white*,  
fontWeight: FontWeight.*w600*),  
),  
),  
Align(  
 alignment: Alignment.*topLeft*,  
child: Text(  
"Enter your registered email",  
style: TextStyle(  
color: Colors.*white60*, fontSize: 13),  
)  
 ),  
SizedBox(  
 height: 50,  
),  
Column(  
crossAxisAlignment: CrossAxisAlignment.start,  
children: <Widget>[  
SizedBox(  
 height: 10,  
),  
Container(  
 alignment: Alignment.*centerLeft*,  
decoration: BoxDecoration(  
color: Color(0xFF6CA8F1),  
borderRadius: BorderRadius.circular(10),  
boxShadow: [  
BoxShadow(  
color: Colors.*black12*,  
blurRadius: 6,  
offset: Offset(0, 2)  
 )  
 ]  
 ),  
height: 60,  
child: TextFormField(  
 controller: \_emailController,  
validator: *emailValidate*,  
keyboardType: TextInputType.*emailAddress*,  
style: TextStyle(  
color: Colors.*white*,  
),  
decoration: InputDecoration(  
 border: InputBorder.*none*,  
contentPadding: EdgeInsets.only(top: 14),  
prefixIcon:  
Icon(Icons.*mail*, color: Colors.*white*),  
hintText: "Email",  
hintStyle:  
TextStyle(color: Colors.*grey*[300]),  
suffixIcon: TextButton(  
 child: Text(  
"Send OTP",  
style: TextStyle(color: Colors.*white*),),  
onPressed: () {  
if(\_verifyformKey.currentState.validate()) {  
if(error != null) {  
return 'ERROR';  
}  
else if(checkEmail(\_emailController.text) == true){  
 print("checkEmail : "+checkEmail(\_emailController.text));  
sendOTP();  
}  
 }  
 },  
)  
 ),  
)  
 )  
 ],  
),  
SizedBox(  
 height: 30,  
),  
Column(  
crossAxisAlignment: CrossAxisAlignment.start,  
children: <Widget>[  
SizedBox(  
 height: 10,  
),  
Container(  
 alignment: Alignment.*centerLeft*,  
decoration: BoxDecoration(  
color: Color(0xFF6CA8F1),  
borderRadius: BorderRadius.circular(10),  
boxShadow: [  
BoxShadow(  
color: Colors.*black12*,  
blurRadius: 6,  
offset: Offset(0, 2)  
 )  
 ]  
 ),  
height: 60,  
child: TextFormField(  
 controller: \_otpController,  
style: TextStyle(  
color: Colors.*white*,  
),  
decoration: InputDecoration(  
 border: InputBorder.*none*,  
contentPadding: EdgeInsets.only(top: 14),  
prefixIcon:  
Icon(Icons.*lock*, color: Colors.*white*),  
hintText: "Enter OTP",  
hintStyle:  
TextStyle(color: Colors.*grey*[300]),  
),  
)  
 )  
 ],  
),  
SizedBox(  
 height: 12,  
),  
Container(  
 padding: EdgeInsets.symmetric(vertical: 25),  
width: double.*infinity*,  
child: ElevatedButton(  
 child: Text(  
'Verify OTP',  
style: GoogleFonts.*openSans*(  
color: Color(0xFF527DAA),  
fontSize: 20,  
fontWeight: FontWeight.*bold*,  
letterSpacing: 1.5,  
),  
),  
onPressed: () {  
if(\_otpController.text.isEmpty){  
setState( () {  
error = "OTP can't be empty";  
});  
} else{  
verifyOTP();  
 if(verified == true) {  
Navigator.*pushReplacement*(  
 context,   
MaterialPageRoute(builder: (context) =>NewPasswordPage())  
 );  
}  
 }  
 },  
style: ElevatedButton.*styleFrom*(  
 padding: EdgeInsets.all(15),  
primary: Colors.*white*,  
shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(30))  
 ),  
)  
 ),  
],  
),  
),  
),  
)  
 ],  
),  
),  
),  
);  
}  
}

**wrapper.dart:**

import 'package:mumbaichimasoli/screen/SignIn.dart';  
import 'package:mumbaichimasoli/screen/home\_page.dart';  
import 'package:flutter/material.dart';  
import 'package:provider/provider.dart';  
import 'package:mumbaichimasoli/models/user.dart';  
  
class Wrapper extends StatelessWidget {  
@override  
Widget build(BuildContext context) {  
  
final user = Provider.*of*<AppUser>(context);  
  
 if(user == null) {  
return SignIn();  
} else {  
return HomePage();  
}  
  
 }  
}

**Auth.dart:**

import 'package:cloud\_firestore/cloud\_firestore.dart';  
import 'package:mumbaichimasoli/models/user.dart';  
import 'package:mumbaichimasoli/config/config.dart';  
import 'package:firebase\_auth/firebase\_auth.dart';  
import 'package:mumbaichimasoli/services/database.dart';  
  
class AuthService {  
FirebaseAuth\_auth = ConstantVariable.*auth*;  
String error;  
  
String getUserId() {  
return \_auth.currentUser.uid;  
}  
  
// create user object based on FirebaseUser  
AppUser\_userFromFirebaseUser(User user) {  
return user != null ? AppUser(uid: user.uid) : null;  
}  
  
// auth change user stream  
Stream<AppUser>get user {  
return \_auth.authStateChanges().map(\_userFromFirebaseUser);  
}  
  
  
// sign up with email and password  
Future register(String fname, String lname, String email, String password) async {  
try {  
UserCredential result = await \_auth.createUserWithEmailAndPassword(email: email, password: password);  
User user = result.user;  
  
// create a new document for the user with uid  
await DatabaseService(uid: user.uid).updateUserData(fname, lname, email);  
 return \_userFromFirebaseUser(user);  
} catch (e) {  
error = e.message;  
print(e.toString());  
 return null;  
}  
 }  
  
// request an account  
Future request(String fname, String lname, String email) async {  
try{  
await DatabaseService().updateRequestData(fname, lname, email);  
} catch(e){  
 print(e.toString());  
error = e.toString();  
 return e.toString();  
}  
 }  
  
// sign in with email and password  
Future signInWithEmailAndPassword(String email, String password) async {  
try {  
UserCredential result = await \_auth.signInWithEmailAndPassword(email: email, password: password);  
User user = result.user;  
 return \_userFromFirebaseUser(user);  
} catch (e) {  
error = e.message;  
print(e.toString());  
 return null;  
}  
 }  
  
// sign out  
Future signOut() async {  
try {  
return await \_auth.signOut();  
  
} catch (e) {  
 print(e.toString());  
 return null;  
}  
 }  
  
// Reset Password  
Future sendPasswordResetEmail(String email) async {  
error = "A Password reset link has been sent to "+email;  
\_auth.sendPasswordResetEmail(email: email);  
 return true;  
}  
  
// check if user is logged in or not  
Future isAvailable() async {  
if (\_auth.currentUser!= null) {  
return true;  
}  
return false;  
}  
  
 Future returnError() async {  
if(error != null){  
return error;  
}  
return null;  
}  
}  
  
class EmailValidator {  
static String *validate*(String value) {  
if (value.isEmpty) {  
return " Email can't be empty";  
}  
return null;  
}  
}  
  
class PasswordValidator {  
static String *validate*(String value) {  
if (value.isEmpty) {  
return " Password can't be empty";  
}   
if (value.length<6) {  
return " Password must have more than 6 characters";  
}  
return null;  
}  
}  
  
class FirstNameValidator {  
static String *validate*(String value) {  
if (value.isEmpty) {  
return " First Name can't be empty";  
}  
if (value.length<2) {  
return " First Name must be at least 2 characters long";  
}  
if (value.length>50) {  
return " First Name must be less than 50 characters long";  
}  
return null;  
}  
}  
  
class LastNameValidator {  
static String *validate*(String value) {  
if (value.isEmpty) {  
return " Last Name can't be empty";  
}  
if (value.length<2) {  
return " Last Name must be at least 2 characters long";  
}  
if (value.length>50) {  
return " Last Name must be less than 50 characters long";  
}  
return null;  
}  
}  
  
class UsernameValidator {  
static String *validate*(String value) {  
if (value.isEmpty) {  
 print("IN USERNAME VALIDATOR");  
 return " Username can't be empty";  
}  
return null;  
}  
}

**database.dart:**

import 'package:cloud\_firestore/cloud\_firestore.dart';  
import 'package:firebase\_auth/firebase\_auth.dart';  
  
import 'package:mumbaichimasoli/services/auth.dart';  
import 'package:intl/intl.dart';  
  
class DatabaseService {  
  
final AuthService\_auth = AuthService();  
  
 final String uid;  
 final String aid;  
 final String docid;  
DatabaseService({ this.uid, this.aid, this.docid});  
  
 final String CartDetails= "CartDetails";  
  
 final CollectionReferenceuserCollection= FirebaseFirestore.*instance*.collection('users');  
 final CollectionReferencerequestCollection= FirebaseFirestore.*instance*.collection('requests');  
 final CollectionReferenceadminCollection= FirebaseFirestore.*instance*.collection("admin");  
 final CollectionReferenceproductsReference= FirebaseFirestore.*instance*.collection('products');  
  
getDate() {  
DateTime now = DateTime.now();  
String formattedDate = DateFormat('yyyy-MM-dd - kk:mm').format(now);  
 return formattedDate;  
}  
  
 Future updateUserData(String fname, String lname, String email) async {  
await userCollection.doc(uid).set({  
'fname': fname,  
'lname': lname,  
'email': email,  
'Address': null,  
});  
 return await userCollection.doc(\_auth.getUserId()).collection('CartDetail').doc(CartDetails).set({  
'Total': 0.0,  
});  
}  
  
  
  
updateUserName(String fname, String lname) {  
FirebaseFirestore.*instance*.runTransaction((transaction) async {  
transaction.update(  
userCollection.doc(\_auth.getUserId()), {"fname": fname, "lname": lname});  
});  
}  
  
 Future updateRequestData(String fname, String lname, String email) async {  
return await requestCollection.doc().set({  
'fname': fname,  
'lname': lname,  
'email': email,  
});  
}  
  
// admin password update  
updateAdminPassword(String password) {  
FirebaseFirestore.*instance*.runTransaction((transaction) async {  
transaction.update(  
adminCollection.doc("hZCPSR7OpSrwWqteIjRN"), {"password": password});  
});  
}  
  
submitFeedback(String feedback, String docid) async {  
return await productsReference.doc(docid).collection('Feedback').doc(\_auth.getUserId()).set({  
'Feedback': feedback  
 });  
}  
  
updateAddress(String address) {  
FirebaseFirestore.*instance*.runTransaction((transaction) async {  
transaction.update(  
userCollection.doc(\_auth.getUserId()), {'Address' : address}  
 );  
});  
}  
  
removeProduct(String docid) {  
FirebaseFirestore.*instance*.runTransaction((transaction) async {  
transaction.delete(  
productsReference.doc(docid)  
 );  
});  
}  
  
checkout(double total) async {  
 String checkoutTime = getDate();  
  
// Place Order  
print('INSIDE PLACEORDER');  
 await userCollection.doc(\_auth.getUserId()).collection('Order').doc(checkoutTime).set({  
'Date': DateTime.now(),  
'Total': total,   
});  
  
// Copy Collection  
print('INSIDE COPYCOLLECTION');  
userCollection.doc(\_auth.getUserId()).collection('Cart').get().then((value) => {  
value.docs.forEach((doc) {  
var promise = userCollection.doc(\_auth.getUserId()).collection('Order').doc(checkoutTime).collection('Products').doc(doc.id).set(doc.data());  
})  
 });  
  
// removeCart  
print('INSIDE REMOVECART');  
userCollection.doc(\_auth.getUserId()).collection('Cart').get().then((snapshot) {  
for(DocumentSnapshot ds in snapshot.docs){  
ds.reference.delete();  
}  
 });  
FirebaseFirestore.*instance*.runTransaction((transaction) async {  
 transaction.update(userCollection.doc(\_auth.getUserId()).collection('CartDetail').doc(CartDetails), {"Total":0.0});  
});   
  
}  
  
removeProductFromCart(String docid) {  
FirebaseFirestore.*instance*.runTransaction((transaction) async {  
transaction.delete(  
userCollection.doc(\_auth.getUserId()).collection('Cart').doc(docid)  
 );  
});  
}  
  
cancelRequest(String docid) {  
FirebaseFirestore.*instance*.runTransaction((transaction) async {  
transaction.delete(requestCollection.doc(docid));  
});  
}  
  
  
getQuantity(docid) async {  
return await userCollection.doc(\_auth.getUserId()).collection('Cart').doc(docid).get().then((document) {  
document.get('Quantity');  
});  
}  
  
updateCartTotal(double total) {  
FirebaseFirestore.*instance*.runTransaction((transaction) async {  
 transaction.update(userCollection.doc(\_auth.getUserId()).collection("CartDetail").doc(CartDetails), {"Total": total});  
});  
}  
  
updateProductAmount(String docid, int amount) {  
FirebaseFirestore.*instance*.runTransaction((transaction) async {  
transaction.update(  
userCollection.doc(\_auth.getUserId()).collection('Cart').doc(docid), {"Amount": amount} );  
});  
}  
  
updateProductQuantity(String docid, int quantity) {  
FirebaseFirestore.*instance*.runTransaction((transaction) async {  
transaction.update(  
userCollection.doc(\_auth.getUserId()).collection('Cart').doc(docid), {"Quantity": quantity} );  
});  
}  
  
}

**loading.dart:**

import 'package:flutter/material.dart';  
import 'package:flutter\_spinkit/flutter\_spinkit.dart';  
  
class Loading extends StatelessWidget {  
@override  
Widget build(BuildContext context) {  
return Container(  
color: Colors.*white*,  
child: Center(  
 child: SpinKitChasingDots(  
color: Colors.*blue*,  
size: 50.0,  
),  
),  
);  
}  
}

**HomeTab.dart:**

import 'package:cloud\_firestore/cloud\_firestore.dart';  
import 'package:flutter/material.dart';  
import 'package:mumbaichimasoli/constants.dart';  
import 'package:mumbaichimasoli/screen/detailProd.dart';  
  
import 'package:mumbaichimasoli/widgets/productCard.dart';  
import 'package:mumbaichimasoli/widgets/topBar.dart';  
  
class HomeTbextends StatelessWidget {  
final CollectionReference\_productsRefrence=  
FirebaseFirestore.*instance*.collection('products');  
@override  
Widget build(BuildContext context) {  
 Size screenSize() {  
return MediaQuery.*of*(context).size;  
}  
  
return Container(  
 child: Stack(  
 children: <Widget>[  
FutureBuilder<QuerySnapshot>(  
 future: \_productsRefrence.get(),  
builder: (context, snapshot) {  
if (snapshot.hasError) {  
return Scaffold(  
 body: Center(  
 child: Text("Error: ${snapshot.error}"),  
),  
);  
}  
  
if (snapshot.connectionState== ConnectionState.done) {  
return ListView(  
 padding: EdgeInsets.only(top: 80.0, bottom: 20.0),  
children: snapshot.data.docs.map((document) {  
return ProductCard(  
 title: document["name"],  
imageUrl: document['images'][0],  
price: "\₹${document["price"]}",  
onPressed: () {  
try {  
Navigator.*push*(  
 context,  
MaterialPageRoute(  
 builder: (context) =>  
DetailProd(  
productId: document.id,  
)));  
}catch(e){  
 print(e.toString());  
}  
 },  
);  
}).toList());  
}  
  
return Scaffold(  
 body: Center(  
 child: CircularProgressIndicator(),  
),  
);  
},  
),  
TopBar(  
 title: "Mumbai chi Masoli",  
),  
],  
),  
);  
}  
}

**ProfileTab.dart:**

import 'package:cloud\_firestore/cloud\_firestore.dart';  
import 'package:flutter/material.dart';  
import 'package:google\_fonts/google\_fonts.dart';  
import 'package:mumbaichimasoli/constants.dart';  
import 'package:mumbaichimasoli/services/auth.dart';  
import 'package:mumbaichimasoli/services/database.dart';  
import 'package:mumbaichimasoli/screen/ShippingAdd.dart';  
import 'package:mumbaichimasoli/screen/SignIn.dart';  
  
class ProfileTbextends StatefulWidget {  
@override  
\_ProfileTbStatecreateState() =>\_ProfileTbState();  
  
}  
  
class \_ProfileTbStateextends State<ProfileTb> {  
  
final DatabaseService\_databaseService= DatabaseService();  
  
Future<void>editFieldsDailog(BuildContext context) async {  
final TextEditingControllerfnameController = TextEditingController();  
 final TextEditingControllerlnameController = TextEditingController();  
 return await showDialog(  
 context: context,  
builder: (context) {  
return AlertDialog(  
 shape: RoundedRectangleBorder(  
borderRadius: BorderRadius.all(Radius.circular(12.0))),  
contentPadding: EdgeInsets.symmetric(horizontal: 12, vertical: 20),  
content: Stack(  
 children: <Widget>[  
Form(  
 child: Column(  
mainAxisSize: MainAxisSize.min,  
children: <Widget>[  
Text(  
"ENTER NEW DETAILS",  
style: boldHead,  
),  
Padding(  
 padding: EdgeInsets.all(8.0),  
child: TextFormField(  
 controller: fnameController,  
decoration: InputDecoration(hintText: "First Name"),  
),  
),  
Padding(  
 padding: EdgeInsets.all(8.0),  
child: TextFormField(  
 controller: lnameController,  
decoration: InputDecoration(hintText: "Last Name"),  
),  
),  
],  
),  
),  
],  
),  
actions: [  
TextButton(onPressed: () {  
Navigator.*of*(context).pop();  
}, child: Text("CANCEL")),  
TextButton(onPressed: () {  
\_databaseService.updateUserName(fnameController.text, lnameController.text);  
Navigator.*pop*(context);  
}, child: Text("SUBMIT"))  
 ],  
);  
});  
}  
  
 Size screenSize() {  
  
return MediaQuery.*of*(context).size;  
}  
  
final AuthService\_auth = AuthService();  
 final CollectionReferenceuserCollection=  
FirebaseFirestore.*instance*.collection('users');  
// String uid = "${\_auth.getUserId()}";  
  
@override  
Widget build(BuildContext context) {  
// String uid = \_auth.getUserId();  
  
return Scaffold(  
appBar: PreferredSize(  
preferredSize: Size.fromHeight(80),  
  
child: AppBar(  
 elevation: 0.5,  
title: Padding(  
 padding: constEdgeInsets.only(  
 top: 55, left: 24, right: 24.0, bottom: 24.0),  
child: Text(  
"Profile",  
style: GoogleFonts.*poppins*(  
color: Colors.*black*,  
fontSize: 24,  
fontWeight: FontWeight.*bold*),  
),  
),  
backgroundColor: Colors.*white*),  
),  
body: SingleChildScrollView(  
  
  
 child: Stack(  
  
 children: [  
Padding(  
 padding: constEdgeInsets.symmetric(vertical: 50, horizontal: 15),  
child: StreamBuilder(  
 stream: userCollection.doc(\_auth.getUserId()).snapshots(),  
builder: (context, snapshot) {  
if (snapshot.hasError) {  
return Scaffold(  
 body: Center(  
 child: Text("Error: ${snapshot.error}"),  
),  
);  
}  
if (snapshot.connectionState== ConnectionState.active) {  
 Map \_currentuser = snapshot.data.data();  
  
 return Container(  
 child: Padding(  
 padding: constEdgeInsets.symmetric(  
 vertical: 10, horizontal: 10),  
child: Stack(  
 children: [  
Container(  
 child: Column(  
mainAxisAlignment: MainAxisAlignment.start,  
children: [  
Row(  
 children: [  
Text(  
"${\_currentuser['fname']}",  
style: regHead,  
),  
SizedBox(  
 width: 6,  
),  
Text(  
"${\_currentuser['lname']}",  
style: regHead,  
),  
SizedBox(  
 width: 10,  
),  
InkWell(  
onTap: () async {  
await editFieldsDailog(context);  
},  
child: Icon(  
Icons.*edit*,  
size: 20,  
color: Colors.*grey*,  
),  
)  
 ],  
),  
Row(  
 children: [  
Text(  
"${\_currentuser['email']}",  
),  
],  
),  
SizedBox(  
 height: 50,  
),  
TextButton(  
onPressed: () {  
Navigator.*push*(context, MaterialPageRoute(builder:(context)=>ShippingAdd()));  
},  
child: Row(  
 children: [  
Expanded(  
 child: Text(  
"Shipping Address",  
style: TextStyle(color: Colors.*black*),  
)),  
Icon(Icons.*arrow\_forward\_ios*)  
 ],  
),  
style: TextButton.*styleFrom*(  
 padding: EdgeInsets.all(15),  
shape: RoundedRectangleBorder(  
borderRadius:  
BorderRadius.circular(18)),  
backgroundColor: Colors.*blue*[50]),  
),  
SizedBox(height: 10),  
TextButton(  
onPressed: () async {  
await \_auth.signOut();  
Navigator.*pushReplacement*(context, MaterialPageRoute(builder: (context)=>SignIn()));  
},  
child: Row(  
 children: [  
Icon(Icons.*logout*),  
SizedBox(  
 width: 5,  
),  
Expanded(  
 child: Text(  
"LOG OUT",  
style: TextStyle(color: Colors.*black*),  
)),  
Icon(Icons.*arrow\_forward\_ios*)  
 ],  
  
),  
style: TextButton.*styleFrom*(  
 padding: EdgeInsets.all(15),  
shape: RoundedRectangleBorder(  
borderRadius:  
BorderRadius.circular(18)),  
backgroundColor: Colors.*blue*[50]),  
),  
]),  
),  
],  
),  
));  
}  
  
return Center(  
  
 child: CircularProgressIndicator(),  
);  
  
  
  
},  
),  
),  
],  
),  
),  
);  
}  
}

**SavedTab.dart:**

import 'package:cloud\_firestore/cloud\_firestore.dart';  
import 'package:flutter/material.dart';  
import 'package:mumbaichimasoli/screen/detailProdAdmin.dart';  
  
import 'package:mumbaichimasoli/services/auth.dart';  
import 'package:mumbaichimasoli/services/database.dart';  
import 'package:mumbaichimasoli/widgets/topBar.dart';  
  
class SavedTbextends StatelessWidget {  
  
AuthService\_auth = AuthService();  
DatabaseService\_databaseService= DatabaseService();  
  
  
@override  
Widget build(BuildContext context) {  
return Container(  
 child: Stack(  
 children: [  
FutureBuilder<QuerySnapshot>(  
 future: \_databaseService.userCollection  
.doc(\_auth.getUserId())  
 .collection("Wishlist")  
 .get(),  
builder: (context, snapshot) {  
if (snapshot.hasError) {  
return Scaffold(  
 body: Center(  
 child: Text("Error: ${snapshot.error}"),  
),  
);  
}  
  
  
// Collection Data ready to display  
if (snapshot.connectionState== ConnectionState.done) {  
// Display the data inside a list view  
return ListView(  
 padding: EdgeInsets.only(  
 top: 90.0,  
bottom: 12.0,  
),  
children: snapshot.data.docs.map((document) {  
return GestureDetector(  
onTap: () {  
Navigator.*push*(  
 context,  
MaterialPageRoute(  
 builder: (context) =>DetailProdAdmin(  
productId: document.id,  
),  
));  
},  
child: FutureBuilder(  
 future: \_databaseService.productsReference  
.doc(document.id)  
 .get(),  
builder: (context, productSnap) {  
if (productSnap.hasError) {  
return Container(  
 child: Center(  
 child: Text("${productSnap.error}"),  
),  
);  
}  
  
if (productSnap.connectionState==  
ConnectionState.done) {  
 Map \_productMap = productSnap.data.data();  
  
 return Padding(  
 padding: constEdgeInsets.symmetric(  
 vertical: 16.0,  
horizontal: 24.0,  
),  
child: Row(  
mainAxisAlignment: MainAxisAlignment.start,  
children: [  
Container(  
 width: 90,  
height: 90,  
child: ClipRRect(  
borderRadius: BorderRadius.circular(50.0),  
child: Image.network(  
"${\_productMap['images'][0]}",  
fit: BoxFit.cover,  
),  
),  
),  
Expanded(  
 child: Container(  
 padding: EdgeInsets.only(  
 left: 16.0,  
),  
child: Column(  
mainAxisAlignment:  
MainAxisAlignment.start,  
crossAxisAlignment:  
CrossAxisAlignment.start,  
children: [  
Text(  
"${\_productMap['name']}",  
style: TextStyle(  
fontSize: 18.0,  
color: Colors.*black*,  
fontWeight: FontWeight.*w600*),  
),  
Padding(  
 padding: constEdgeInsets.symmetric(  
 vertical: 4.0,  
),  
child: Text(  
"\₹${\_productMap['price']}",  
style: TextStyle(  
fontSize: 16.0,  
color: Theme.*of*(context)  
 .accentColor,  
fontWeight: FontWeight.*w600*),  
),  
),  
  
],  
),  
),  
),  
],  
),  
);  
}  
  
return Container(  
 child: Center(  
 child: CircularProgressIndicator(),  
),  
);  
},  
),  
);  
}).toList(),  
);  
}  
  
// Loading State  
return Scaffold(  
 body: Center(  
 child: CircularProgressIndicator(),  
),  
);  
},  
),  
TopBar(  
 title: "Saved",  
backArrow: false,  
),  
],  
),  
);  
}  
}

**SearchTab.dart:**

import 'package:cloud\_firestore/cloud\_firestore.dart';  
import 'package:flutter/material.dart';  
import 'package:mumbaichimasoli/screen/detailProdAdmin.dart';  
import 'package:mumbaichimasoli/services/database.dart';  
import 'package:mumbaichimasoli/widgets/SearchInput.dart';  
import 'package:mumbaichimasoli/widgets/productCard.dart';  
import '../constants.dart';  
  
class SearchTbextends StatefulWidget {  
@override  
\_SearchTbStatecreateState() =>\_SearchTbState();  
}  
  
class \_SearchTbStateextends State<SearchTb> {  
 String \_searchString= "";  
  
DatabaseService\_databaseService= DatabaseService();  
  
@override  
Widget build(BuildContext context) {  
return Container(  
 child: Stack(  
 children: [  
if (\_searchString.isEmpty)  
Container(  
 alignment: Alignment.*center*,  
child: Text(  
"Search Results",  
style: regHead,  
))  
else  
Padding(  
 padding: EdgeInsets.only(top: 20),  
child: FutureBuilder<QuerySnapshot>(  
 future: \_databaseService.productsReference  
.orderBy("searchString")  
 .startAt([\_searchString]).endAt(  
 ["$\_searchString\uf8ff"]).get(),  
builder: (context, snapshot) {  
if (snapshot.hasError) {  
return Scaffold(  
 body: Center(  
 child: Text("Error: ${snapshot.error}"),  
),  
);  
}  
  
if (snapshot.connectionState== ConnectionState.done) {  
return ListView(  
 padding: EdgeInsets.only(top: 80.0, bottom: 20.0),  
children: snapshot.data.docs.map((document) {  
return ProductCard(  
 title: document["name"],  
imageUrl: document['images'][0],  
price: "\₹${document["price"]}",  
onPressed: () {  
Navigator.*push*(  
 context,  
MaterialPageRoute(  
 builder: (context) =>DetailProdAdmin(  
productId: document.id,  
)));  
},  
);  
}).toList());  
}  
  
return Scaffold(  
 body: Center(  
 child: CircularProgressIndicator(),  
),  
);  
},  
),  
),  
SearchInput(  
keyboardType: TextInputType.*name*,  
hintText: "Search..",  
onSubmitted: (value) {  
setState(() {  
\_searchString= value.toLowerCase();  
});  
},  
),  
],  
),  
);  
}  
}

**bottomtabs.dart:**

import 'package:flutter/cupertino.dart';  
import 'package:flutter/material.dart';  
  
class BottomTabsextends StatefulWidget {  
final int selectedTab;  
 final Function(int) taBPressed;  
BottomTabs({this.selectedTab, this.taBPressed});  
  
@override  
\_BottomTabsStatecreateState() =>\_BottomTabsState();  
}  
  
class \_BottomTabsStateextends State<BottomTabs> {  
 int \_selectedTab= 0;  
  
@override  
Widget build(BuildContext context) {  
\_selectedTab= widget.selectedTab?? 0;  
  
 return Container(  
 decoration: BoxDecoration(  
color: Colors.*white*,  
borderRadius: BorderRadius.only(  
topLeft: Radius.circular(12.0),  
topRight: Radius.circular(12.0),  
),  
boxShadow: [  
BoxShadow(  
color: Colors.*black*.withOpacity(0.06),  
spreadRadius: 1.0,  
blurRadius: 22.0)  
 ]),  
child: Row(  
mainAxisAlignment: MainAxisAlignment.spaceAround,  
children: [  
BottomTabsBtn(  
imagePath: "assets/images/tab\_home.png",  
selected: \_selectedTab== 0 ? true : false,  
onPressed: () {  
widget.taBPressed(0);  
},  
),  
BottomTabsBtn(  
imagePath: "assets/images/tab\_search.png",  
selected: \_selectedTab== 1 ? true : false,  
onPressed: () {  
widget.taBPressed(1);  
},  
),  
BottomTabsBtn(  
imagePath: "assets/images/tab\_saved.png",  
selected: \_selectedTab== 2 ? true : false,  
onPressed: () {  
widget.taBPressed(2);  
},  
),  
BottomTabsBtn(  
imagePath: "assets/images/outline\_account.png",  
selected: \_selectedTab== 3 ? true : false,  
onPressed: () {  
widget.taBPressed(3);  
}),  
],  
));  
}  
}  
  
class BottomTabsBtnextends StatelessWidget {  
final String imagePath;  
 final bool selected;  
 final Function onPressed;  
BottomTabsBtn({this.imagePath, this.selected, this.onPressed});  
  
@override  
Widget build(BuildContext context) {  
 bool \_selected = selected ?? false;  
  
 return GestureDetector(  
onTap: onPressed,  
child: Container(  
 padding: EdgeInsets.symmetric(horizontal: 24, vertical: 24),  
decoration: BoxDecoration(  
 border: Border(  
 top: BorderSide(  
color: \_selected  
 ? Theme.*of*(context).accentColor  
: Colors.*transparent*,  
width: 2.0))),  
child: Image(  
 image: AssetImage(imagePath?? "assets/images/tab\_home.png"),  
width: 22,  
height: 22,  
color: \_selected ?Theme.*of*(context).accentColor: Colors.*black*,  
),  
),  
);  
}  
}

**cartbottom.dart:**

import 'package:auto\_size\_text/auto\_size\_text.dart';  
import 'package:cloud\_firestore/cloud\_firestore.dart';  
import 'package:flutter/material.dart';  
import 'package:mumbaichimasoli/config/size\_config.dart';  
import 'package:mumbaichimasoli/constants.dart';  
import 'package:mumbaichimasoli/services/auth.dart';  
import 'package:mumbaichimasoli/services/database.dart';  
import 'package:provider/provider.dart';  
import 'package:mumbaichimasoli/widgets/topBar.dart';  
  
class Cartbottomextends StatefulWidget {  
@override  
\_CartbottomStatecreateState() =>\_CartbottomState();  
  
}  
  
class \_CartbottomStateextends State<Cartbottom> {  
DatabaseService\_databaseService= DatabaseService();  
AuthService\_auth = AuthService();  
TopBar\_topBar= TopBar();  
  
int total;  
bool isCartEmpty;  
bool cartLength;  
  
checkCartLength() async {  
// double total;  
  
\_databaseService.userCollection.doc(\_auth.getUserId()).collection('Cart').snapshots().listen((snapshot) {  
cartLength= snapshot.docs.isEmpty;  
 return cartLength.toString();  
});  
}  
  
@override  
Widget build(BuildContext context) {  
SizeConfig().init(context);  
  
  
  
 return Container(  
 child: Row(  
 children: [  
StreamBuilder(  
 stream: \_databaseService.userCollection.doc(\_auth.getUserId()).collection("CartDetail").doc("CartDetails").snapshots(),  
builder: (context, snapshot) {  
if (snapshot.hasError) {  
return Container(  
 child: Center(  
 child: Text("Error: ${snapshot.error}"),  
),  
);  
}  
  
if (snapshot.connectionState== ConnectionState.active) {  
 Map \_cartMap = snapshot.data.data();  
  
 return Container(  
 child: Row(  
 children: <Widget>[  
Padding(  
 padding: EdgeInsets.only(left: 20),  
child: Text(  
"TOTAL :",  
style: regHead,  
),  
),  
SizedBox(  
 width: 10,  
),  
Container(  
 decoration: BoxDecoration(  
color: Colors.*blue*[100],  
borderRadius: BorderRadius.circular(10)),  
width: 100,  
height: 50,  
child: Center(  
 child: AutoSizeText(  
// "0",  
"${\_cartMap['Total']}",  
maxLines: 1,  
style: regHead,  
),  
),  
  
),  
SizedBox(  
 width: 05,  
),  
GestureDetector(  
onTap: () {  
checkCartLength();  
 if(!cartLength?? true){  
\_databaseService.checkout(\_cartMap['Total']);  
}  
// print('cartlengthfrom on tap $cartLength');  
  
},  
  
  
child: Container(  
 height: 50,  
width: SizeConfig.*blockSizeHorizontal*\*45,  
decoration: BoxDecoration(  
borderRadius: BorderRadius.circular(10), color: Colors.*red*),  
child: Center(  
 child: Text(  
"Checkout",  
style: regHead,  
),  
),  
))  
 ],  
),  
height: 60,  
decoration: BoxDecoration(  
borderRadius: BorderRadius.circular(20), color: Colors.*white10*),  
);  
}  
  
return Container(  
 child: Center(  
 child: CircularProgressIndicator(),  
),  
);  
}  
 ),  
],  
),  
);  
  
}  
}

**custom\_stepper.dart:**

import 'package:flutter/material.dart';  
import 'package:mumbaichimasoli/constants.dart';  
  
class CustomStepperextends StatefulWidget {  
CustomStepper({  
@required this.lowerLimit,  
@required this.upperLimit,  
@required this.stepValue,  
@required this.iconSize,  
@required this.value,  
@required this.onChanged,  
});  
  
 final int lowerLimit;  
 final int upperLimit;  
 final int stepValue;  
 final double iconSize;  
int value;  
 final ValueChanged<dynamic>onChanged;  
  
@override  
\_CustomStepperStatecreateState() =>\_CustomStepperState();  
}  
  
class \_CustomStepperStateextends State<CustomStepper> {  
@override  
Widget build(BuildContext context) {  
return Container(  
//width: 100,  
 //height: 50,  
decoration: BoxDecoration(  
color: Colors.*grey*[100],  
borderRadius: BorderRadius.all(Radius.circular(20))),  
child: Row(  
//mainAxisAlignment: MainAxisAlignment.center,  
children: [  
IconButton(icon: Icon(Icons.*remove*,size: 15,), onPressed: () {  
setState(() {  
widget.value= widget.value== widget.lowerLimit? widget.lowerLimit: widget.value-= widget.stepValue;  
this.widget.onChanged(widget.value);  
});  
}),  
Container(  
 width: this.widget.iconSize,  
child: Text(  
"${widget.value}",  
style: TextStyle(fontSize: widget.iconSize\* 0.8),  
textAlign: TextAlign.center,  
),  
),  
IconButton(icon: Icon(Icons.*add*,size: 15,), onPressed: () {  
setState(() {  
widget.value= widget.value== widget.upperLimit? widget.upperLimit: widget.value+= widget.stepValue;  
this.widget.onChanged(widget.value);  
});  
}),  
Container(  
 height: 40,  
width: 40,  
decoration: BoxDecoration(color: Colors.*blue*[100],borderRadius: BorderRadius.only(bottomRight: Radius.circular(18),topRight: Radius.circular(18))),  
child: Center(child: Text("Kg",style: regtext,)),  
)  
 ],  
),  
);  
}  
}

**enterqty.dart:**

import 'package:flutter/material.dart';  
import 'package:mumbaichimasoli/constants.dart';  
  
class Enterqtyextends StatelessWidget {  
  
  
final quantityController= TextEditingController();  
  
String returnQuantity() {  
return quantityController.text;  
}  
  
@override  
Widget build(BuildContext context) {  
return Container(  
 width: 110,  
height: 50,  
decoration: BoxDecoration(  
borderRadius: BorderRadius.circular(18), color: Colors.*white*),  
child: Row(  
 children: <Widget>[  
Container(  
 height: 40,  
width: 70,  
child: Center(  
 child: TextField(  
 controller: quantityController,  
decoration: InputDecoration(  
hintText: "Quantity",  
hintStyle: TextStyle(fontSize: 12),  
border: InputBorder.*none*,  
contentPadding:  
EdgeInsets.symmetric(horizontal: 12, vertical: 17)),  
)),  
decoration: BoxDecoration(  
color: Colors.*blue*[100],  
borderRadius: BorderRadius.only(  
topLeft: Radius.circular(11),  
bottomLeft: Radius.circular(11))),  
),  
Container(  
 height: 40,  
width: 40,  
decoration: BoxDecoration(  
color: Colors.*grey*[200],  
borderRadius: BorderRadius.only(  
topRight: Radius.circular(11),  
bottomRight: Radius.circular(11))),  
child: Center(  
 child: Text(  
"Kg",  
style: regtext,  
)),  
)  
 ],  
));  
}  
}

**Fishquantity.dart:**

import 'package:flutter/material.dart';  
  
class SelQuanextends StatefulWidget {  
final List quantity;  
 final Function(int) onSelQuan;  
  
SelQuan({this.quantity, this.onSelQuan});  
@override  
\_SelQuanStatecreateState() =>\_SelQuanState();  
}  
  
class \_SelQuanStateextends State<SelQuan> {  
 int selquantity= 0;  
  
  
@override  
Widget build(BuildContext context) {  
  
  
  
return Padding(  
 padding: constEdgeInsets.only(left: 16.0),  
child: Row(  
 children: [  
for (var i = 0; i<widget.quantity.length; i++)  
GestureDetector(  
onTap: () {  
  
widget.onSelQuan(widget.quantity[i]);  
setState(() {  
selquantity= i;  
  
});  
},  
child: Container(  
 margin: EdgeInsets.symmetric(horizontal: 6),  
height: 40,  
width: 42,  
decoration: BoxDecoration(  
color: selquantity== i  
 ? Theme.*of*(context).accentColor  
: Color(0xffdcdcdc),  
borderRadius: BorderRadius.circular(30)),  
alignment: Alignment.*center*,  
child: Text(  
'${widget.quantity[i]}',  
style: TextStyle(  
fontWeight: FontWeight.*w600*,  
color: selquantity== i ? Colors.*white*:Colors.*blue*),  
),  
),  
)  
 ],  
),  
);  
}  
}

**imageSwipe.dart:**

import 'package:flutter/material.dart';  
import 'package:mumbaichimasoli/config/size\_config.dart';  
  
class ImageSwipeextends StatefulWidget {  
final List album;  
ImageSwipe({this.album});  
  
@override  
\_ImageSwipeStatecreateState() =>\_ImageSwipeState();  
}  
  
class \_ImageSwipeStateextends State<ImageSwipe> {  
 int \_selectedimage= 0;  
@override  
Widget build(BuildContext context) {  
return Container(  
 height: SizeConfig.*blockSizeVertical*\* 40,  
child: Stack(  
 children: [  
PageView(  
onPageChanged: (num) {  
setState(() {  
\_selectedimage= num;  
});  
},  
children: [  
for (var i = 0; i<widget.album.length; i++)  
Container(  
 padding: EdgeInsets.only(top: 80),  
child: Image.network(  
"${widget.album[i]}",  
),  
)  
 ],  
),  
Positioned(  
 bottom: 10,  
left: 0,  
right: 0,  
child: Row(  
mainAxisAlignment: MainAxisAlignment.center,  
children: [  
for (var i = 0; i<widget.album.length; i++)  
AnimatedContainer(  
 duration: Duration(milliseconds: 300),  
curve: Curves.*easeInOutCubic*,  
margin: EdgeInsets.symmetric(horizontal: 4.0),  
width: \_selectedimage== i ? 25.0 :10,  
height: 10,  
decoration: BoxDecoration(  
borderRadius: BorderRadius.circular(10),  
color: Colors.*indigoAccent*.withOpacity(0.6)),  
)  
 ],  
),  
)  
 ],  
),  
// child: Image.network("${documentData['images'][1]}")  
);  
}  
}

**productCard.dart**

import 'package:flutter/material.dart';  
import 'package:mumbaichimasoli/config/size\_config.dart';  
import 'package:mumbaichimasoli/screen/detailProdAdmin.dart';  
  
import '../constants.dart';  
  
class ProductCardextends StatelessWidget {  
final String productID;  
 final Function onPressed;  
 final String imageUrl;  
 final String title;  
 final String price;  
  
ProductCard(  
 {this.imageUrl, this.onPressed, this.price, this.title, this.productID});  
  
@override  
Widget build(BuildContext context) {  
SizeConfig().init(context);  
Size screenSize() {  
return MediaQuery.*of*(context).size;  
}  
  
return GestureDetector(  
onTap: onPressed,  
child: Container(  
 decoration: BoxDecoration(borderRadius: BorderRadius.circular(12.0)),  
height: 200,  
margin: EdgeInsets.symmetric(vertical: 12.0, horizontal: 24.0),  
child: Stack(  
 children: [  
Container(  
 width: screenSize().width,  
height:SizeConfig.*blockSizeVertical*\*200,  
child: ClipRRect(  
borderRadius: BorderRadius.circular(12.0),  
child: Image.network(  
"$imageUrl",  
fit: BoxFit.cover,  
),  
),  
),  
Positioned(  
 bottom: 0,  
left: 0,  
right: 0,  
child: Padding(  
 padding: constEdgeInsets.all(24),  
child: Row(  
mainAxisAlignment: MainAxisAlignment.spaceBetween,  
children: <Widget>[  
Text(  
title,  
style: regHead,  
),  
Text(  
price,  
style: TextStyle(  
fontSize: 18.0,  
color: Colors.*red*,  
fontWeight: FontWeight.*bold*),  
)  
 ],  
),  
),  
)  
 ],  
),  
),  
);  
}  
}

**SearchInput.dart:**

import 'package:flutter/material.dart';  
import 'package:mumbaichimasoli/constants.dart';  
  
class SearchInputextends StatelessWidget {  
final String hintText;  
//final Function(String) onChanged;  
final Function(String) onSubmitted;  
 final FocusNodefocusNode;  
 final TextInputActiontextInputAction;  
 final TextInputTypekeyboardType;  
//final bool isPasswordField;  
SearchInput(  
 {this.hintText, this.onSubmitted, this.focusNode, this.textInputAction, this.keyboardType});  
  
@override  
Widget build(BuildContext context) {  
return Padding(  
 padding: constEdgeInsets.only(top: 30),  
child: Container(  
 margin: EdgeInsets.symmetric(  
 vertical: 8.0,  
horizontal: 24.0,  
),  
decoration: BoxDecoration(  
color: Colors.*blue*[100], borderRadius: BorderRadius.circular(12.0)),  
child: TextField(  
  
onSubmitted: onSubmitted,  
textInputAction: textInputAction,  
keyboardType: keyboardType,  
decoration: InputDecoration(  
prefixIcon: Icon(Icons.*search\_rounded*),  
border: InputBorder.*none*,  
hintText: hintText,  
contentPadding: EdgeInsets.symmetric(  
 horizontal: 24.0,  
vertical: 20.0,  
)),  
style: regHead),  
),  
);  
}  
}

**topBar.dart:**

import 'package:cloud\_firestore/cloud\_firestore.dart';  
import 'package:firebase\_auth/firebase\_auth.dart';  
import 'package:flutter/material.dart';  
import 'package:google\_fonts/google\_fonts.dart';  
import 'package:mumbaichimasoli/screen/cartPage.dart';  
import 'package:mumbaichimasoli/services/auth.dart';  
import 'package:mumbaichimasoli/services/database.dart';  
  
class TopBarextends StatelessWidget {  
final bool backArrow;  
 final String title;  
 final bool hasTitle;  
 final bool hasBg;  
 final bool hasCart;  
TopBar({this.backArrow, this.title, this.hasTitle, this.hasBg, this.hasCart});  
  
AuthService\_auth = AuthService();  
DatabaseService\_databaseService= DatabaseService();  
  
  
 final CollectionReference\_users =  
FirebaseFirestore.*instance*.collection('users');  
  
  
@override  
Widget build(BuildContext context) {  
 bool \_backArrow = backArrow?? false;  
bool \_hasTitle = hasTitle?? true;  
bool \_hasBg = hasBg?? true;  
bool \_hasCart = hasCart?? true;  
  
 return Container(  
 decoration: BoxDecoration(  
 gradient: \_hasBg  
 ? LinearGradient(  
colors: [Colors.*white*, Colors.*white*],  
begin: Alignment(0, 0),  
end: Alignment(0, 1),  
)  
 : null),  
padding: EdgeInsets.only(top: 26, left: 24, right: 24.0, bottom: 24.0),  
child: Row(  
mainAxisAlignment: MainAxisAlignment.spaceBetween,  
children: [  
if (\_backArrow)  
GestureDetector(  
onTap: () {  
Navigator.*pop*(context);  
},  
child: Container(  
 child: Icon(  
Icons.*arrow\_back\_ios*,  
size: 30,  
),  
),  
),  
 if (\_hasTitle)  
Text(title ?? "TopBar",  
style: GoogleFonts.*poppins*(  
fontSize: 24, fontWeight: FontWeight.*bold*)  
 ),  
 if(\_hasCart)   
GestureDetector(  
onTap: () {  
Navigator.*push*(  
 context, MaterialPageRoute(builder: (context) =>CartPage()));  
},  
child: Container(  
 width: 42,  
height: 42,  
decoration: BoxDecoration(  
color: Colors.*blue*,  
borderRadius: BorderRadius.circular(20)),  
alignment: Alignment.*center*,  
child: StreamBuilder(  
 stream: \_databaseService.userCollection  
.doc(\_auth.getUserId())  
 .collection("Cart")  
 .snapshots(),  
builder: (context, snapshot) {  
 int \_totalItems = 0;  
  
 if (snapshot.connectionState== ConnectionState.active) {  
 List \_documents = snapshot.data.docs;  
\_totalItems = \_documents.length;  
}  
  
return Text(  
"$\_totalItems" ?? "0",  
style: TextStyle(  
fontSize: 16.0,  
fontWeight: FontWeight.*bold*,  
color: Colors.*white*,  
),  
);  
},  
)),  
)  
 ],  
),  
);  
}  
}

**constants.dart:**

import 'package:flutter/material.dart';  
import 'package:google\_fonts/google\_fonts.dart';  
  
final kHintTextStyle = TextStyle(  
color: Colors.*white54*,  
fontFamily: 'OpenSans',  
);  
  
final kLabelStyle = TextStyle(  
color: Colors.*white*,  
fontWeight: FontWeight.*bold*,  
fontFamily: 'OpenSans',  
);  
  
final kBoxDecorationStyle = BoxDecoration(  
color: Color(0xFF6CA8F1),  
borderRadius: BorderRadius.circular(10.0),  
boxShadow: [  
BoxShadow(  
color: Colors.*black12*,  
blurRadius: 6.0,  
offset: Offset(0, 2),  
),  
],  
);  
  
bool isHiddenPassword = true;  
  
final regHead =  
TextStyle(fontSize: 18, fontWeight: FontWeight.*bold*, color: Colors.*black*);  
  
final boldHead =  
GoogleFonts.*openSans*(fontSize: 22, fontWeight: FontWeight.*bold*);  
  
final regtext =  
TextStyle(fontSize: 14, fontWeight: FontWeight.*w600*, color: Colors.*black*);  
  
final xlboldhead = GoogleFonts.*openSans*(  
fontSize: 30, fontWeight: FontWeight.*bold*, color: Colors.*black*);

**generated\_plgin\_registrant.dart:**

//  
// Generated file. Do not edit.  
//  
  
// ignore\_for\_file: lines\_longer\_than\_80\_chars  
  
import 'package:cloud\_firestore\_web/cloud\_firestore\_web.dart';  
import 'package:firebase\_auth\_web/firebase\_auth\_web.dart';  
import 'package:firebase\_core\_web/firebase\_core\_web.dart';  
import 'package:firebase\_storage\_web/firebase\_storage\_web.dart';  
import 'package:fluttertoast/fluttertoast\_web.dart';  
import 'package:image\_picker\_for\_web/image\_picker\_for\_web.dart';  
  
import 'package:flutter\_web\_plugins/flutter\_web\_plugins.dart';  
  
// ignore: public\_member\_api\_docs  
void registerPlugins(Registrar registrar) {  
FirebaseFirestoreWeb.*registerWith*(registrar);  
FirebaseAuthWeb.*registerWith*(registrar);  
FirebaseCoreWeb.*registerWith*(registrar);  
FirebaseStorageWeb.*registerWith*(registrar);  
FluttertoastWebPlugin.*registerWith*(registrar);  
ImagePickerPlugin.*registerWith*(registrar);  
registrar.registerMessageHandler();  
}

CHAPTER 8

TESTING

**8. Testing**

**User Module:**

**Pre-requisite:**

1.Internet should be available.

2.App should be installed and open.

**SIGN IN PAGE TEST CASE**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case**  **Id** | **Test Case Objectives** | **Input Data** | **Expected Result** | **Actual Result** | **Status** |
| TC1 | Sign-in Page | Email id (Correct) and Password(Correct) | It should be accepted | It Accept | Pass |
| TC2 | Sign-in Page | Email id (Correct) and Password(Wrong) | It should not accept | The password is invalid notice shown | Pass |
| TC3 | Sign-in Page | Email id (Wrong) and Password(Wrong) | It should not accept | There is no USER record notice shown | Pass |
| TC4 | Sign-in Page | Email id (Wrong) and Password(Correct) | It should not accept | There is no USER record notice shown | Pass |
| TC5 | Sign-in Page | Email id (Empty) and Password(Empty) | It should not accept | Email cannot be empty notice shown | Pass |
| TC6 | Sign-in Page | Email id (Correct) and Password(Empty) | It should not accept | Given string is empty or NULL (  Notice shown) | Pass |

**FORGOT PASSWORD Page:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TC1 | Forgot Password | Email id (Empty) | It should not accept | Email cannot be empty(  Notice shown) | Pass |
| TC2 | Forgot Password | Email id (Wrong) | It should not accept | Given string is empty or NULL(  Notice shown) | Pass |
| TC3 | Forgot Password | Email id (Correct) | Mail should receive | Received the mail | Pass |

**REQUEST ACCOUNT PAGE:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TC1 | Request Account | First Name (Empty)  Last Name (Empty)  Email id (Empty) | It should not accept | First Name cannot be empty (Notice shown).  Last Name cannot be empty (Notice shown).  Email cannot be empty (Notice shown) | Pass |
| TC2 | Request Account | First Name (ABC)  Last Name (Empty)  Email id (Empty) | It should not accept | Last Name cannot be empty (Notice shown).  Email cannot be empty (Notice shown) | Pass |
| TC3 | Request Account | First Name (ABC)  Last Name (XYZ)  Email id (Empty) | It should not accept | Email cannot be empty (Notice shown) | Pass |
| TC4 | Request Account | First Name (ABC)  Last Name (XYZ)  Email id (Invalid Email) | It should not send request | It sends request | Fail |
| TC5 | Request Account | First Name (ABC)  Last Name (XYZ)  Email id (valid Email) | It should send request | It sends request | Pass |

**ADMIN SIGN-UP PAGE:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case**  **Id** | **Test Case Objectives** | **Input Data** | **Expected Result** | **Actual Result** | **Status** |
| TC1 | Create Account  (Admin) | First Name (Empty)  Last Name (Empty)  Email id (Empty)  Password(Empty) | It should not accept | First Name cannot be empty (Notice shown).  Last Name cannot be empty (Notice shown).  Email  cannot be empty (Notice shown).  Password  cannot be empty (Notice shown). | Pass |
| TC2 | Create Account  (Admin) | First Name (ABCD)  Last Name (Empty)  Email id (Empty)  Password(Empty) | It should not accept | Last Name cannot be empty (Notice shown).  Email  cannot be empty (Notice shown).  Password  cannot be empty (Notice shown). | Pass |
| TC3 | Create Account  (Admin) | First Name (ABCD)  Last Name (XYZ)  Email id (Empty)  Password(Empty) | It should not accept | Email  cannot be empty (Notice shown).  Password  cannot be empty (Notice shown). | Pass |
| TC4 | Create Account  (Admin) | First Name (ABCD)  Last Name (XYZ)  Email id (Valid Email)  Password(Empty) | It should not accept | Password  cannot be empty (Notice shown). | Pass |
| TC5 | Create Account  (Admin) | First Name (ABCD)  Last Name (XYZ)  Email id (Valid Email)  Password(Only 4 characters) | It should  not accept | Password  Should have more than 6 characters (Notice shown). | Pass |
| TC6 | Create Account  (Admin) | First Name (ABCD)  Last Name (XYZ)  Email id (Valid Email)  Password(6 characters) | It should  accept | Accept | Pass |

**ADMIN SIGN-IN PAGE:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case**  **Id** | **Test Case Objectives** | **Input Data** | **Expected Result** | **Actual Result** | **Status** |
| TC1 | Admin Sign-In | Username (Empty)  Password (Empty) | It should not accept | Username cannot be empty  (Notice shown). | Pass |
| TC2 | Admin Sign-In | Username (Invalid)  Password (Empty) | It should not accept | Password cannot be empty  (Notice shown). | Pass |
| TC3 | Admin Sign-In | Username (Valid)  Password (Empty) | It should not accept | Password cannot be  empty  (Notice shown). | Pass |
| TC4 | Admin Sign-In | Username (Valid)  Password (Invalid) | It should not accept | Password does not match  (Notice shown). | Pass |
| TC5 | Admin Sign-In | Username (Valid)  Password (Valid) | It should accept | Login Successful | Pass |

**ADMIN SIDE FORGOT PASSWORD:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case**  **Id** | **Test Case Objectives** | **Input Data** | **Expected Result** | **Actual Result** | **Status** |
| TC1 | Admin Side Forgot Password  (Verify Email) | Email(Unregistered Email) | It should not Verify | This email is not a registered email  (Notice Received) | Pass |
| TC2 | Admin Side Forgot Password  (Verify Email) | Email(Registered Email) | OTP should receive on registered email | OTP receive | Pass |
| TC3 | Admin Side Forgot Password  (Verify Email) | Email(Registered Email)  OTP(Correct) | It should Verify | Verifies OTP | Pass |
| TC4 | Admin Side Forgot Password  (Verify Email) | Email(Registered Email)  OTP(Incorrect) | It should  not Verify | Incorrect OTP  (Notice Shown) | Pass |

**HOME PAGE TEST CASE:-**

**SEARCH TEST CASE:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case**  **Id** | **Test Case Objectives** | **Input Data** | **Expected Result** | **Actual Result** | **Status** |
| TC1 | SEARCH | Click on search icon | Search page should open | Search opened | Pass |
| TC2 | SEARCH | Search product with first letter “C” | Products  with first letter “C”, | 2 products shown  (“Clams”,  “Crabs”) | Pass |
| TC3 | SEARCH | Search product with first letter “P” | Products with first letter “P”, | 2 products shown  (“Pomfret”,  “Prawns”) | Pass |
| TC4 | SEARCH | Search Products with invalid name(ABC/numbers/  special characters) | Blank page  (Nothing should be shown) | Blank Page  (Nothingshown) | Pass |

**WISHLIST TEST CASE:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case**  **Id** | **Test Case Objectives** | **Input Data** | **Expected Result** | **Actual Result** | **Status** |
| TC1 | Add product to wish list | Click on Wish list button | User can add product to wish list | Products added to wish list | Pass |
| TC2 | View wish list | click wish list icon to view wish list | User can view wish list | Wishlist is visible | Pass |
| TC3 | Remove product wish list | Click again on wish list button | Product should be removed from wish list | Product did not remove from wish list | Failed |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case**  **Id** | **Test Case Objectives** | **Input Data** | **Expected Result** | **Actual Result** | **Status** |
| TC1 | Add to cart | Click on add to cart button | User can add product in cart | Products are added in cart | Pass |
| TC2 | View cart | Click on cart icon | User can view cart | Cart is visible | Pass |
| TC3 | Delete product from cart | Click on delete icon | Product should be removed from the cart | Product did remove from the cart | Pass |
| TC4 | Add Quantity of product | Click on add icon | Product quantity should be increased | Product Quantity does increase | Pass |
| TC5 | Decrease quantity of the product | Click on decrease icon | Product quantity should be decreased | Product quantity does decrease | Pass |
| TC6 | Check out | Click on check out button | Check out screen should be visible | Check out Successful | Pass |

**ADD TO CART TEST CASE:-**

**PROFILE TEST CASE:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case**  **Id** | **Test Case Objectives** | **Input Data** | **Expected Result** | **Actual Result** | **Status** |
| TC1 | Log out | Click on Log out button | User should be logout | User logout | Pass |

CHAPTER 9

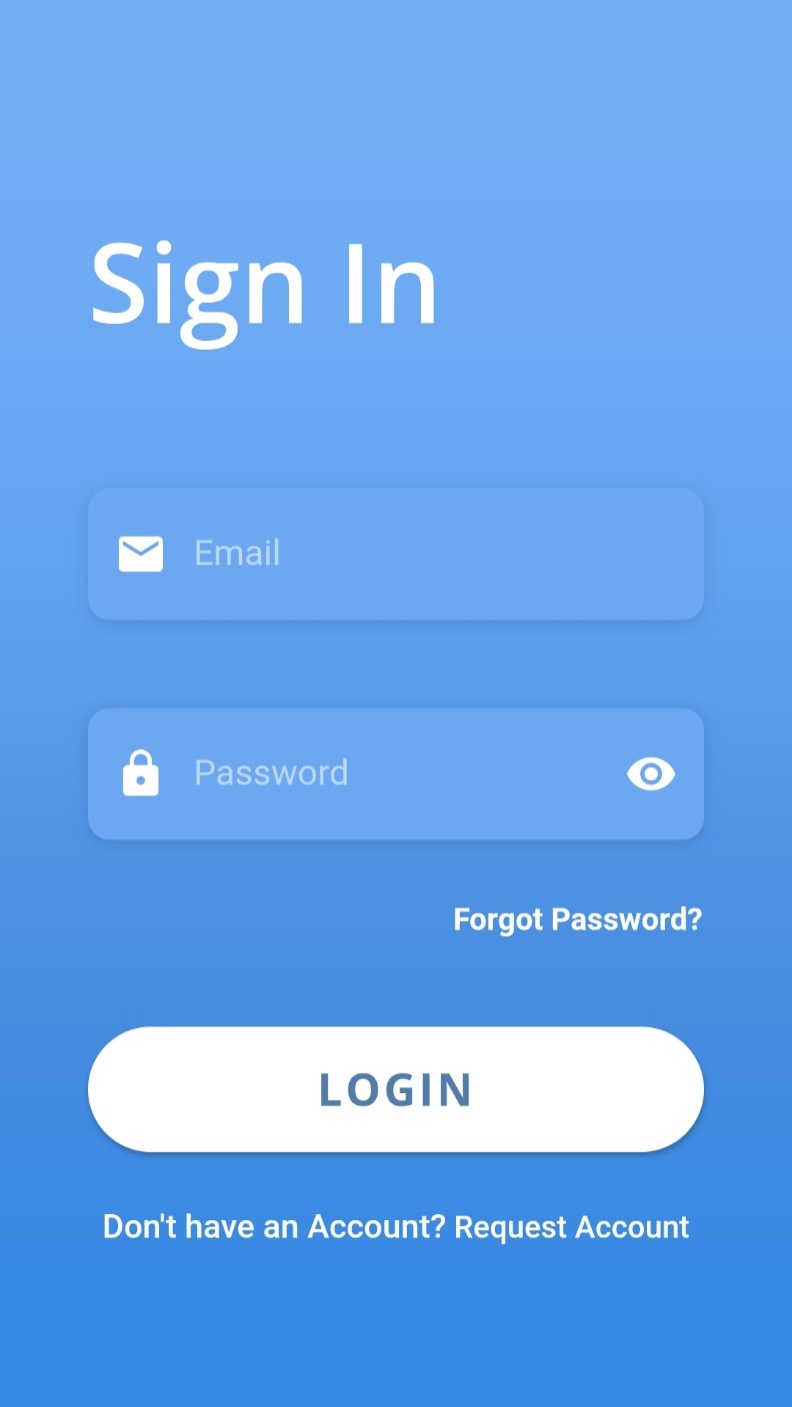
OUTPUTS/ SCREENSHOTS

**User Side**

**WELCOME SCREEN**

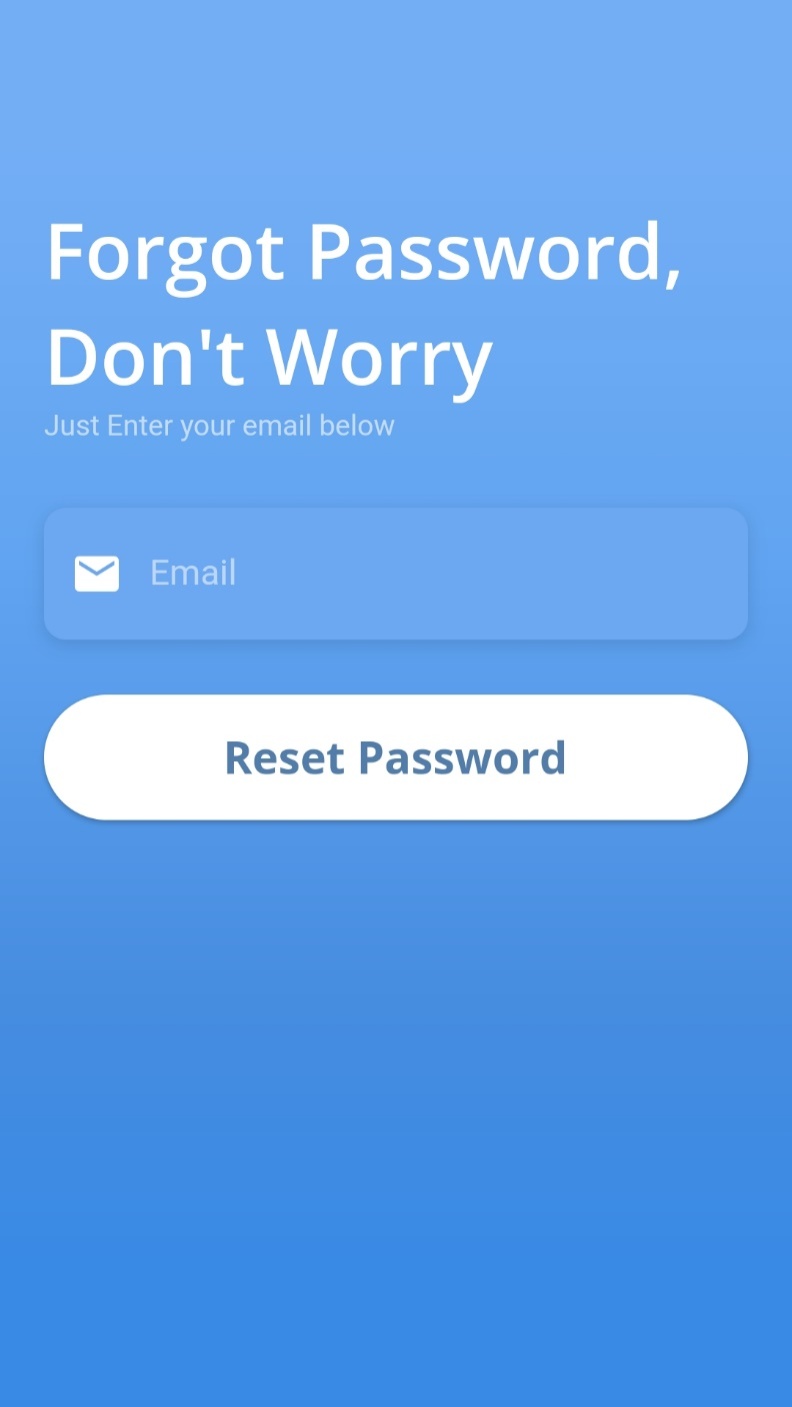
* **This is the first screen to be appeared at the time of launching the application.**



****

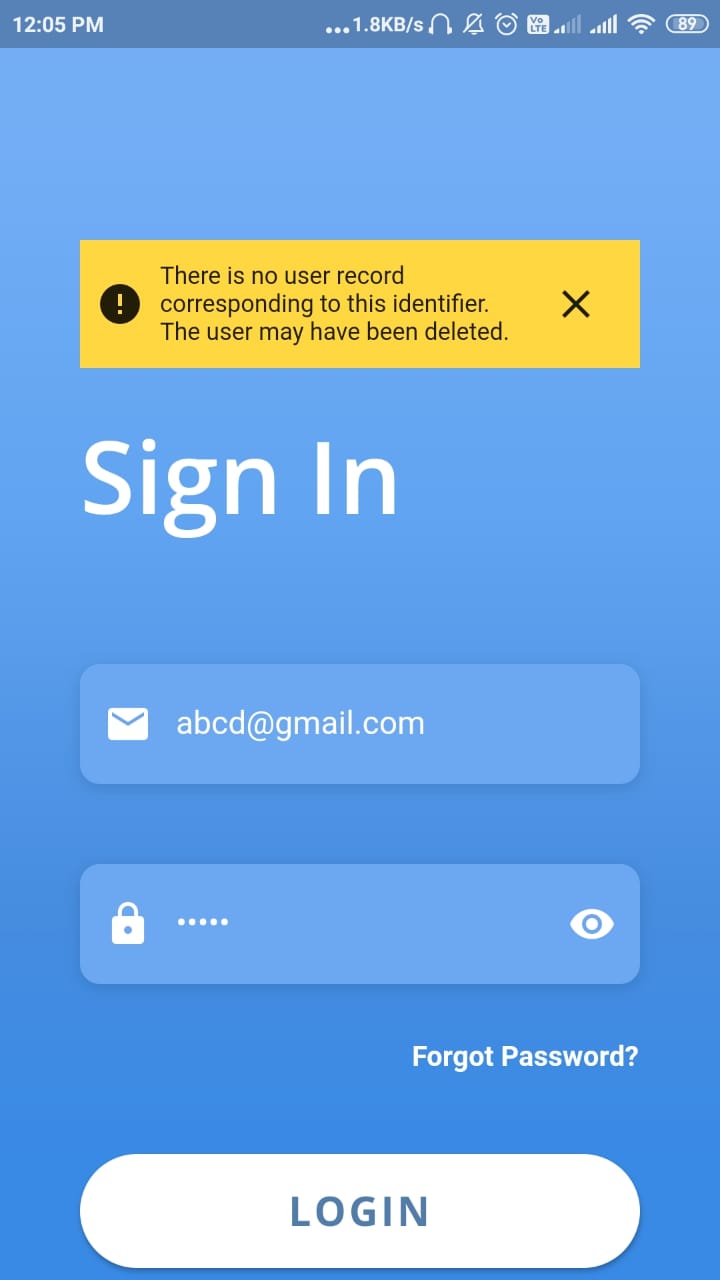
**SIGN IN SCREEN**

* **This screen will appear to customer for signing in the application after registration of an account.**
* **The user will have to enter email id and password to sign into the account.**
* **After entering valid credentials the user will get access to its account.**



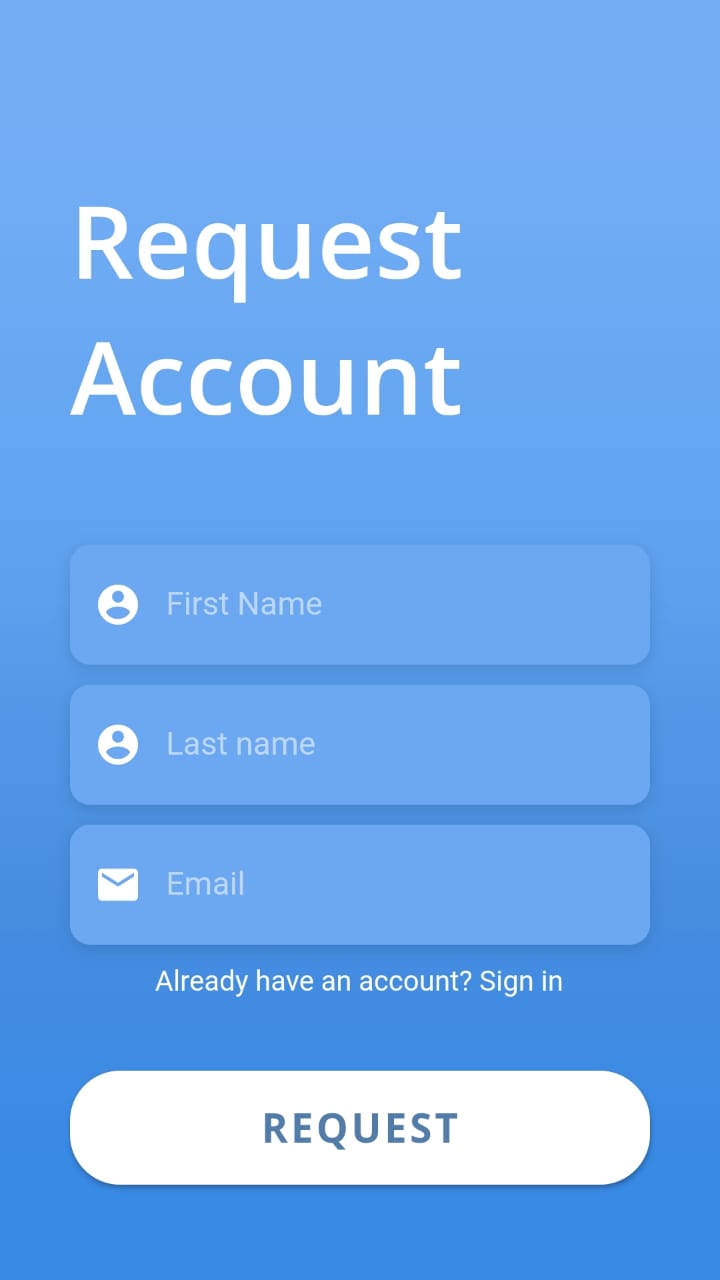
**FORGOT PASSWORD PAGE**

* **If user forgot his password he can reset the password by using his previously registerd email id.**

****

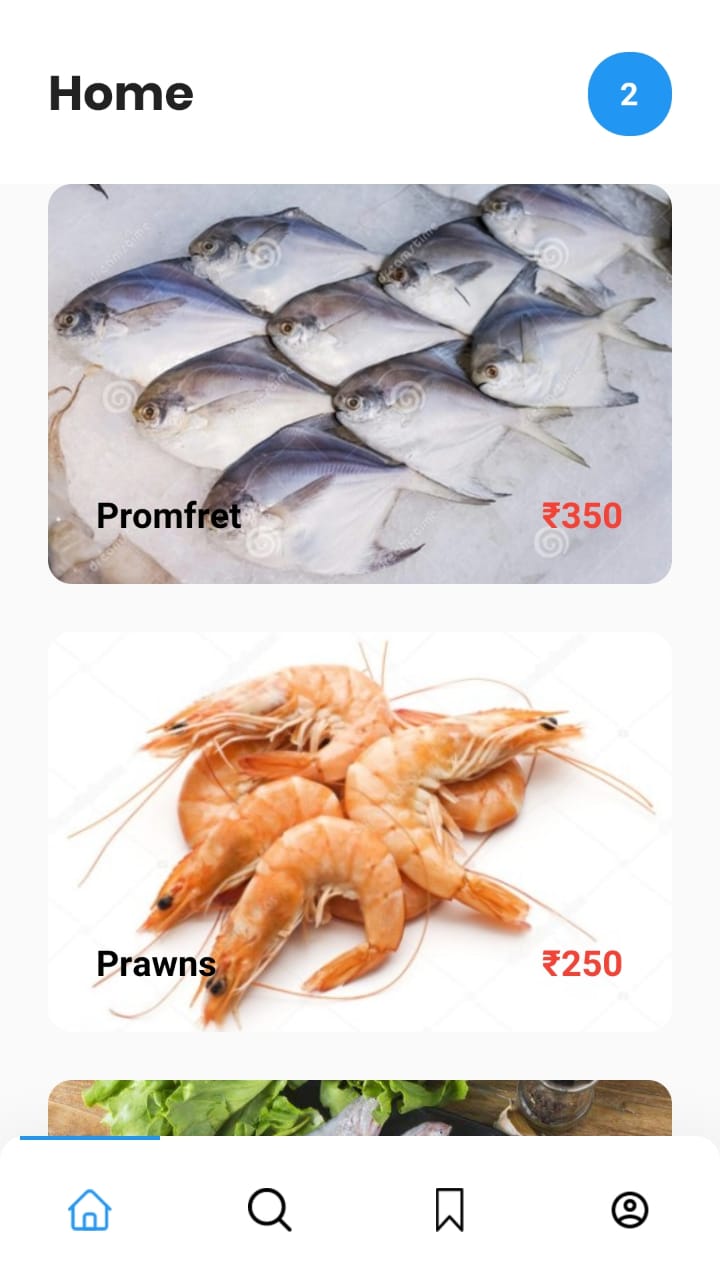
**SIGN IN ERROR SCREEN**

* **This screen will appear to customer if wrong email id & password is inserted.**
* **The user will have to enter coorect email id and password to sign into the account.**
* **After entering valid credentials the user will get access to its account.**

****

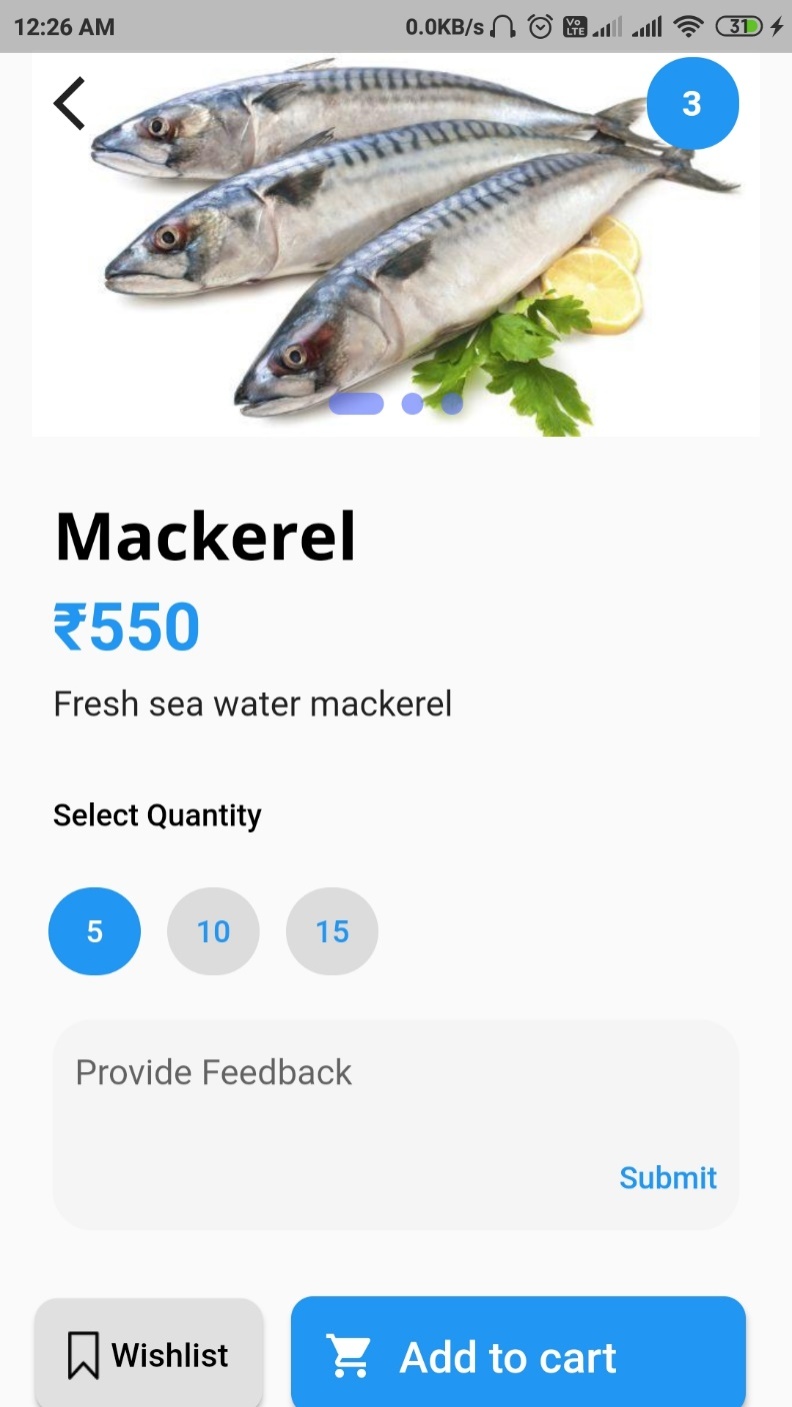
**REQUEST ACCOUNT SCREEN:**

* **This is the screen in which customer have to register an account.**
* **Through this account customer will be able to access application.**
* **The account details will be secured by a password which will be set by customer**

****

**HOME PAGE**

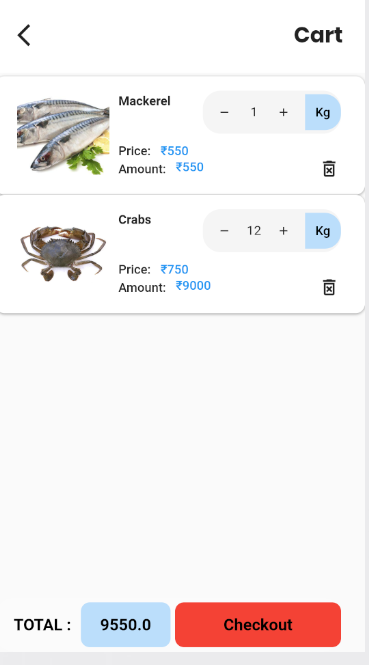
* **This is the screen in where customer can see products.**

****

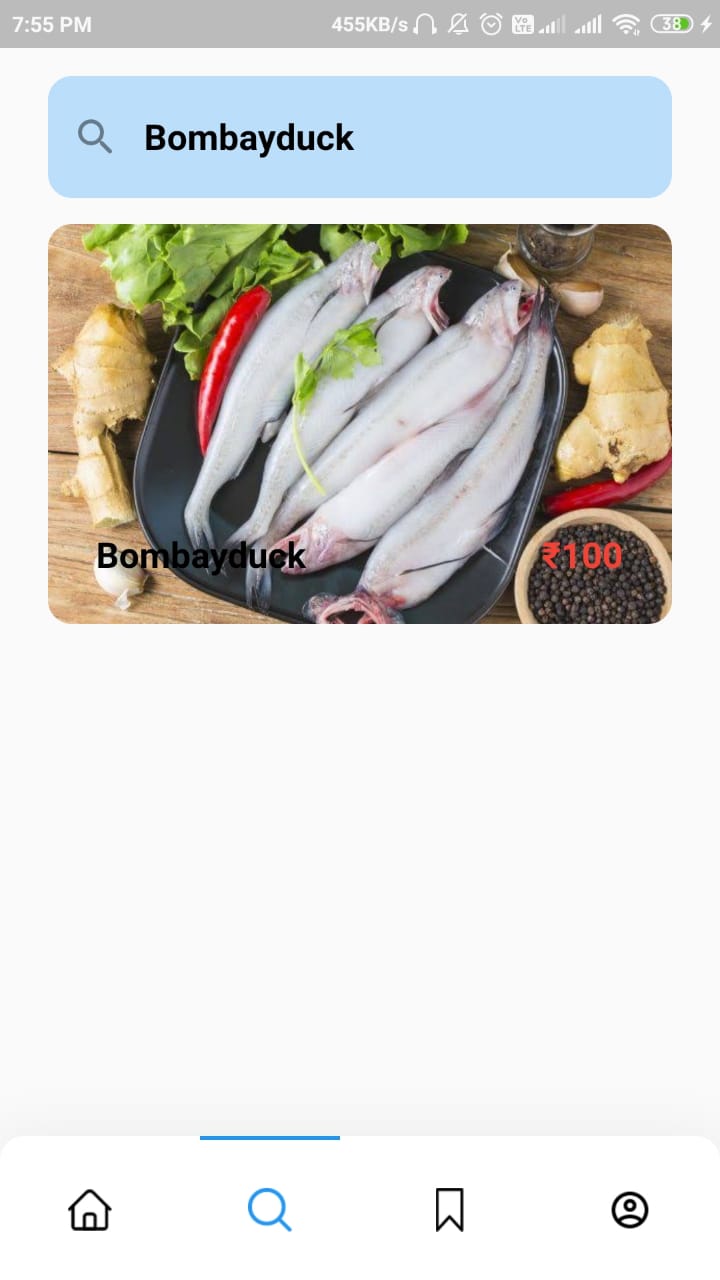
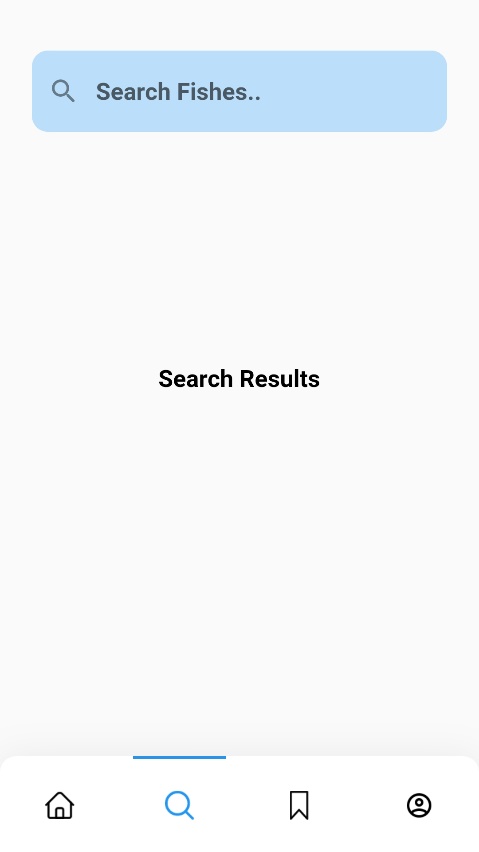
**PRODUCT DETAILS**

* **This is the screen in where customer can see products DETAILS.**

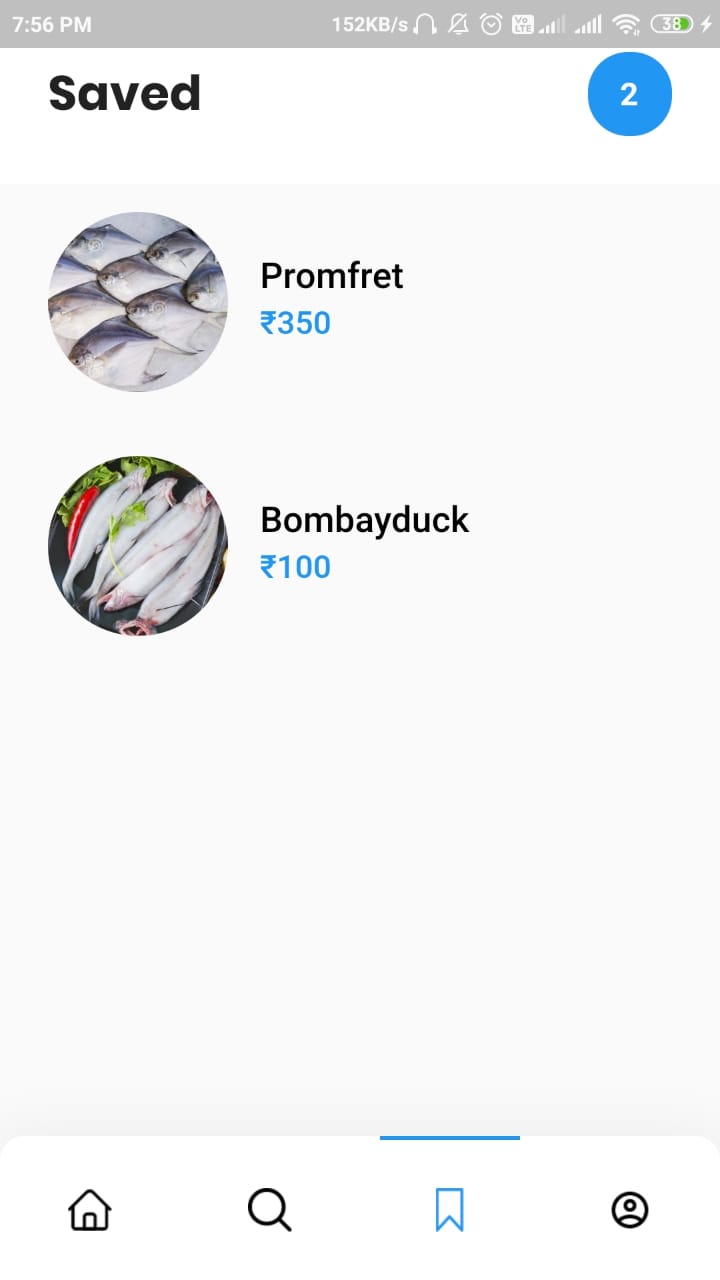
**CART**

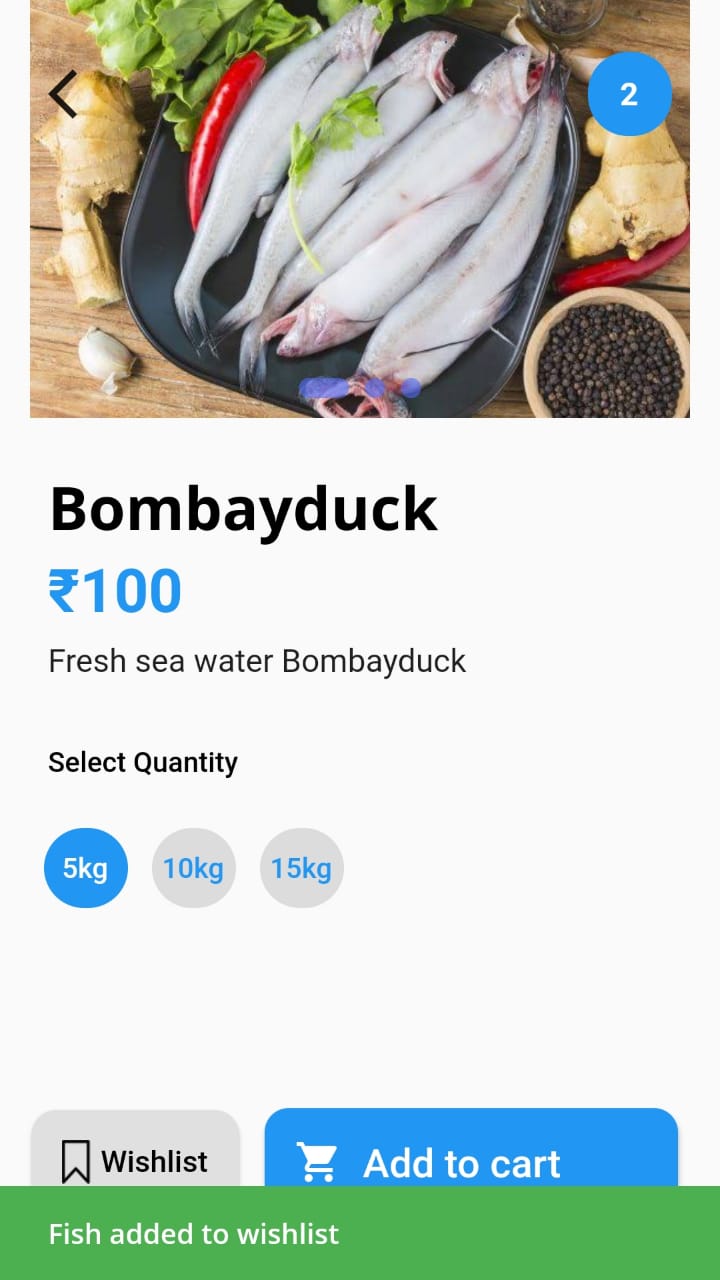


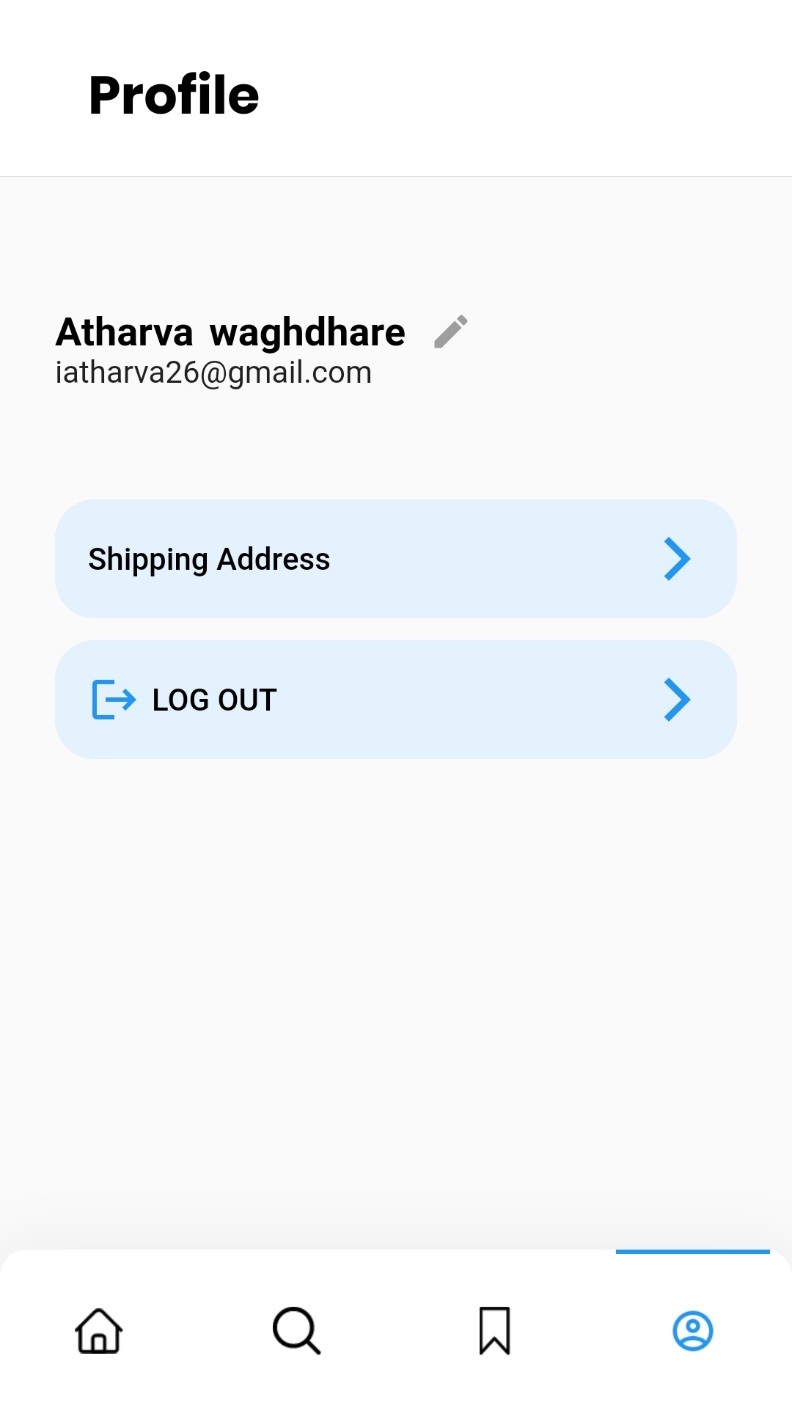
**SEARCH MENU**

****

**WISH LIST**

****

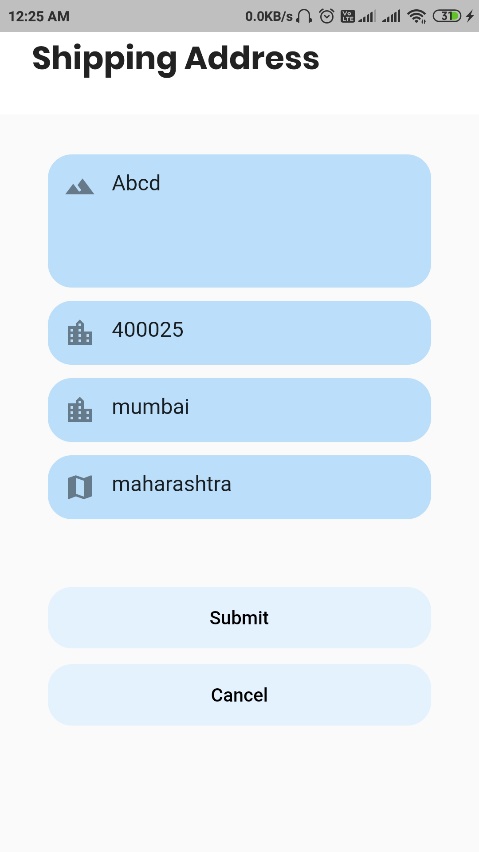
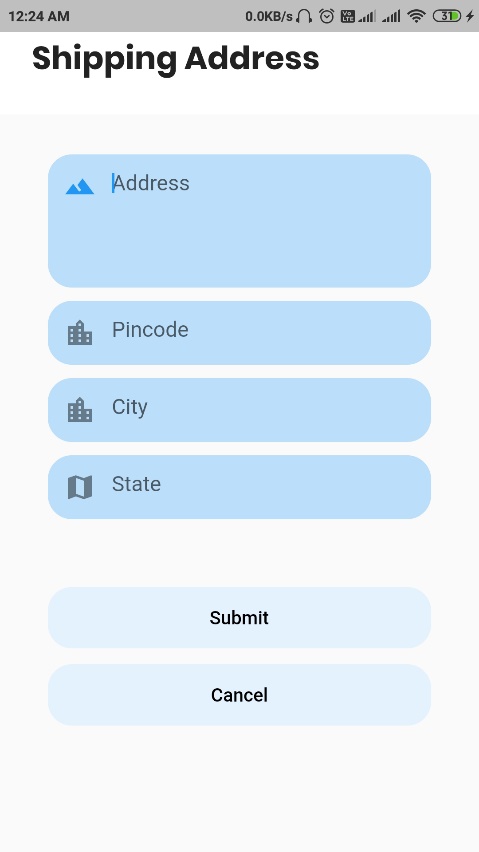
****

****

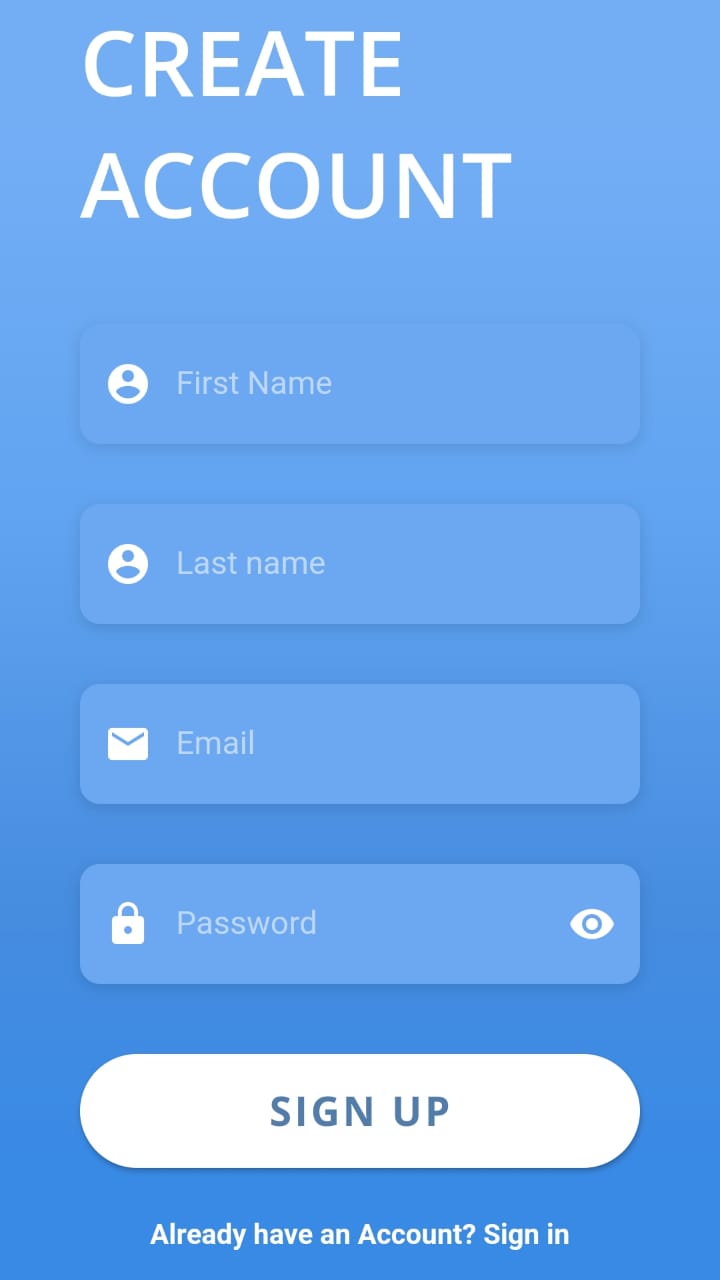
**PROFILE PAGE**

* **The user can see his profile in this page.**

**PROFILE PAGE -ADDRESS**

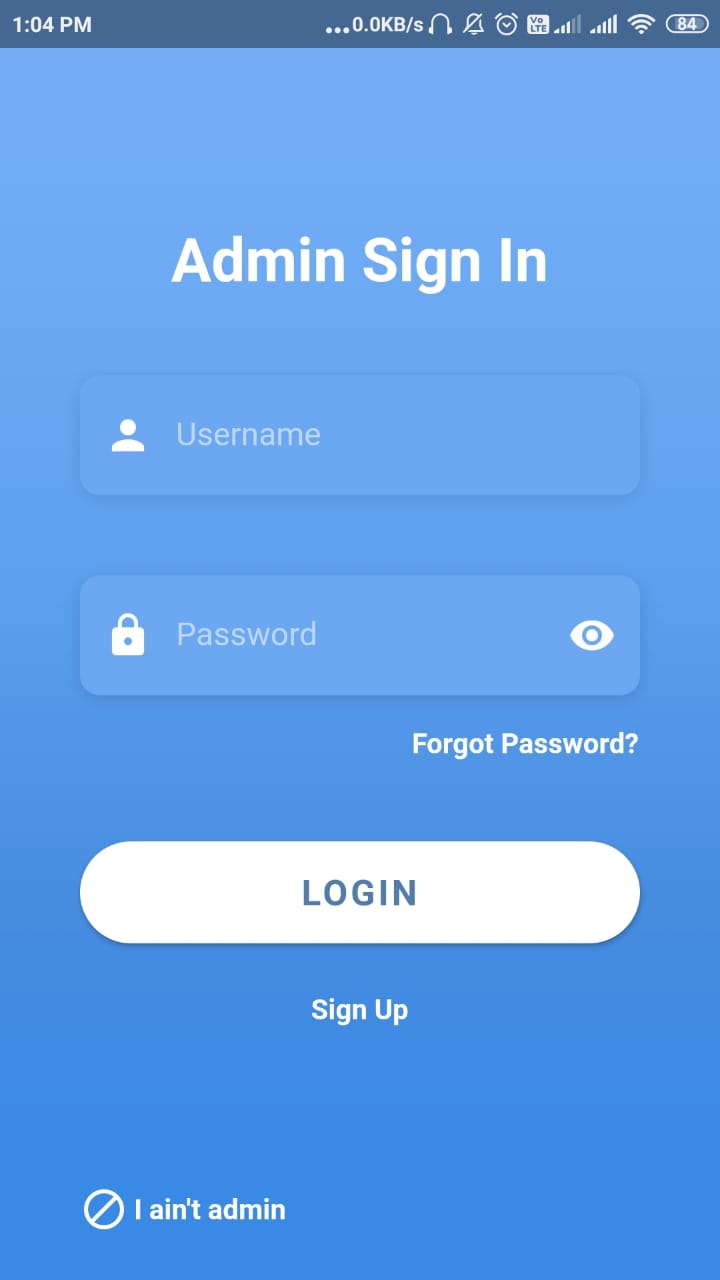
****

**ADMIN SIDE**



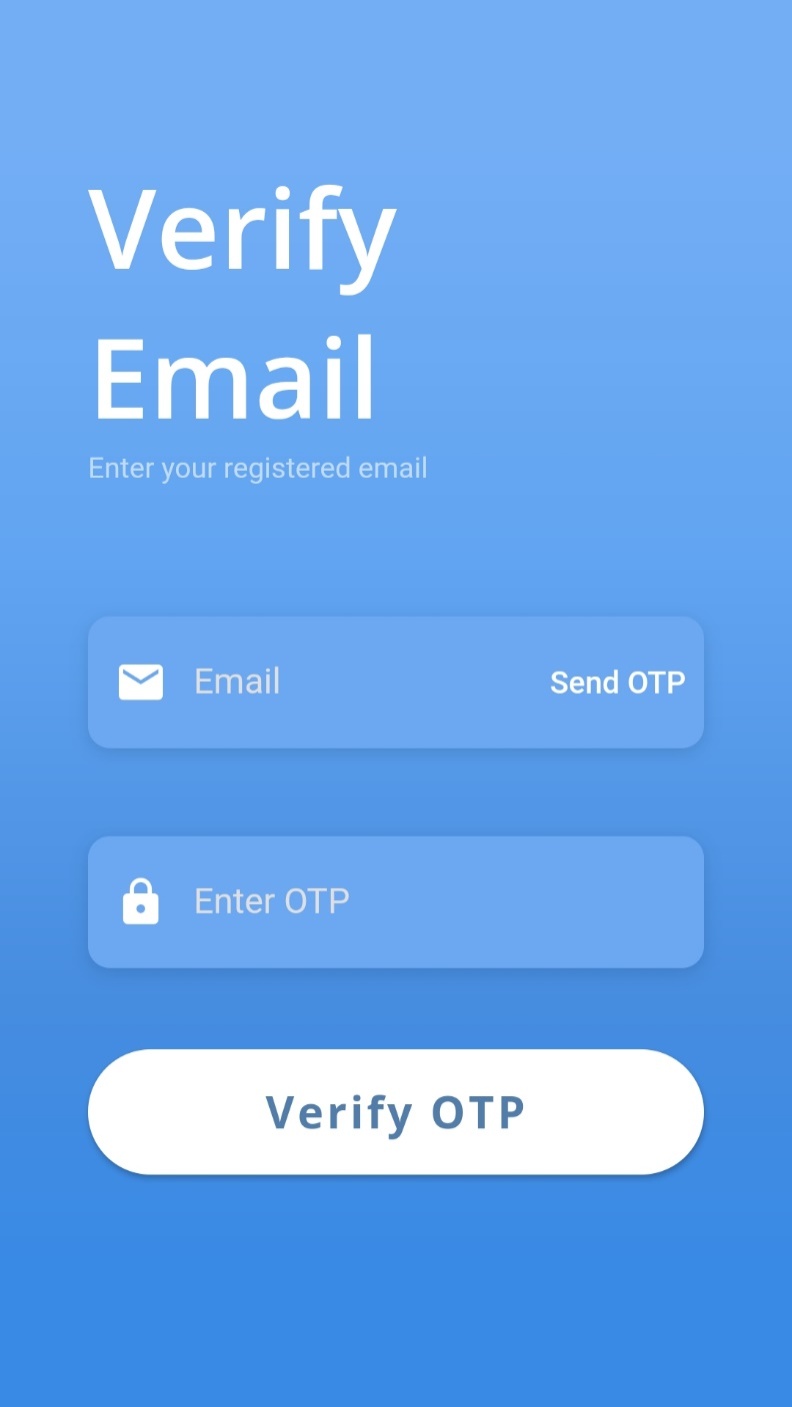
**ADMIN SIGN UP**

* **The Admin can create his account using this page.**
* **After entering valid credentials the admin can sign up.**

****

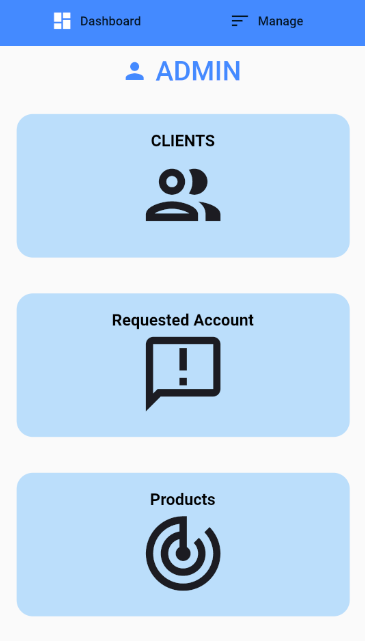
**ADMIN SIGN IN**

* **The Admin will have to enter email id and password to sign into the account.**
* **After entering valid credentials the admin will get access to its admin side.**

****

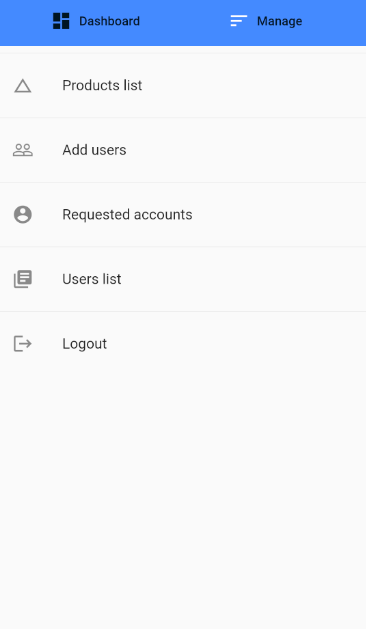
**ADMIN FORGOT PASSWORD**

* **If the admin forgots his password then admin can login using OTP(one time password), sent to registered email id.**
* **After entering valid credentials the admin will get to choose new Password.**

****

**ADMIN HOME PAGE**

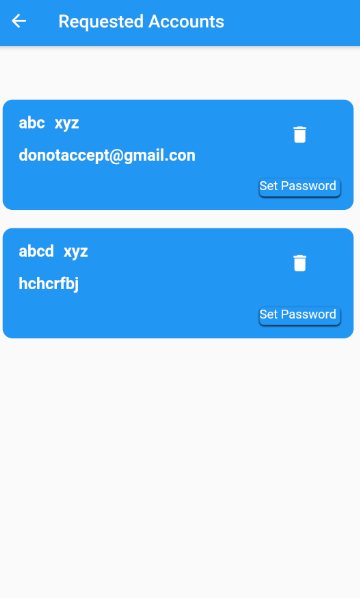
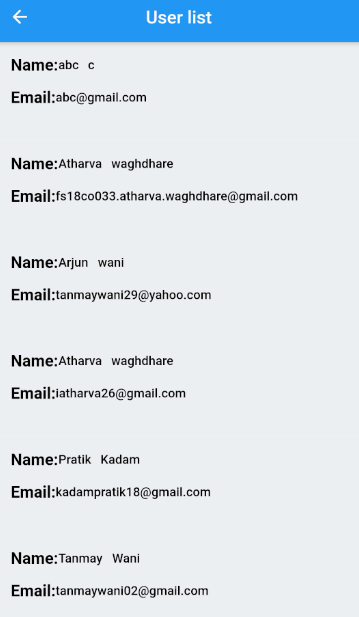
* **The admin home page will help the admin to control the application.**
* **This page contains users information, categories, products, sold products, order details & history of returned products.**

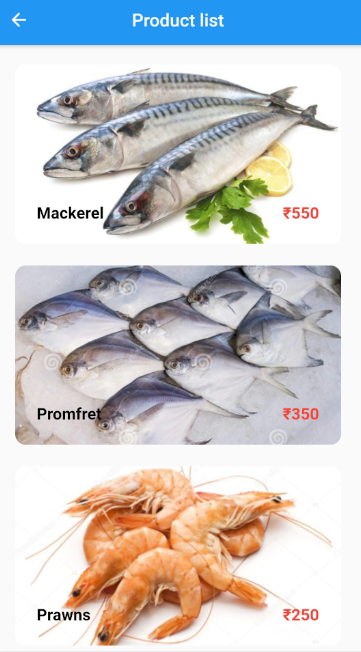
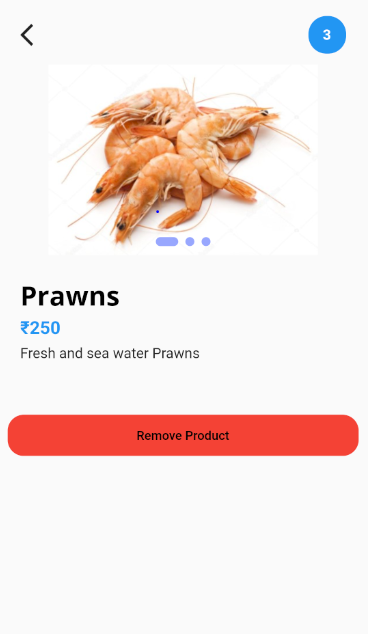
****

**MANAGE PAGE**

* **This page will help the admin to manage the application.**
* **This page contains add products, products list.**
* **Category list, add category, add users, user list & logout.**

**MANAGE PAGE**

****



CHAPTER 10

Bibliography

# Bibliography

1. Android Programming: The Big Nerd Ranch Guide (3rd Edition) (Big Nerd Ranch Guides) 3rd Edition
2. Ashutosh Bhargave , Prof. Mr. S. R Lahane Department of Computer Engineering, GES’s RHSCOE
3. Android Cookbook: Problems and Solutions for Android Developers 2nd Edition
4. Firebase Tutorial: Getting Started - <https://www.raywenderlich.com/3-firebase-tutorial-getting-started>
5. Android App Development In Android Studio: Create visually appealing applications – by by Earl Triggs
6. [ eBook ] The Ultimate Guide to Launch your Food Ordering & Delivery App Based Restaurant Business with SpotnEats – by UberEats Clone Solution

\*\*