E-Royal Mart [Shopping App]

**A Project Report for Industrial Internship**

**Submitted by**

**Tayan Sinha | Abantika Pandit**

**Indrani Dey | Uddipta Sarkar**

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

**BCA**

In the

Computer Application

**Calcutta Institute Engineering and Management**



at

**Ardent Computech Pvt. Ltd.**





**CERTIFICATE FROM SUPERVISOR**

This is to certify that **“TAYAN SINHA, ABANTIKA PANDIT, INDRANI DEY, UDDIPTA SARKAR**” have successfully completed the project titled **"E-Royal Mart Shopping App**" under my supervision during the period from **“16/11/2021”** to **“29/01/2022”** which is in partial fulfillment of requirements for the award of the **BCA** degree and submitted to the Department of “**BCA**” of **“Calcutta Institute Of Engineering and Management”**.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature of the Supervisor

**Date:**

Name of the Project Supervisor: **PALLABI SAHA**



**ACKNOWLEDGEMENT**

The achievement that is associated with the successful completion of  any task would be incomplete without mentioning the names of those  people whose endless cooperation made it possible. Their constant guidance and encouragement made all our efforts successful.

We take this opportunity to express our deep gratitude towards our project mentor, PALLABI SAHA for giving such valuable suggestions, guidance and encouragement during the development of this project work.

Last but not the least we are grateful to all the faculty members of Ardent Computech Pvt. Ltd. for their support.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **Name of the Topic** | **Page No** |
| 1. | Introduction | 6 |
| 1.1 | Objective | 6 |
| 1.2 | Scope | 7 |
| 2. | System Analysis | 8 |
| 2.1 | Feasibility Study | 7 |
| 2.2 | Workflow | 9 |
| 2.3 | Functional requirement | 9 |
| 2.4 | Non- functional requirement | 10 |
| 2.5 | Hardware and software requirement | 10 |
| 3. | System Design | 11 |
| 3.1 | Use Case | 11 |
| 3.2 | DFD | 12 |
| 3.3 | Database Diagram/Screenshots | 13-14 |
| 4 | Coding | 15 - 68 |
| 4.2 | User interface designs | 69-72 |

|  |  |  |
| --- | --- | --- |
| 5. | Testing and Implementation | 73 |
| 5.1 | Objective of testing | 73 |
| 5.2 | Testing process and strategy | 73-74 |
| 5.3 | Test Cases | 74 |
| 6. | Security Measures | 75 |
| 6.1 | Database Security measures system | 75 |
| 6.2 | Software security measures | 76 |
| 6.3 | Limitations | 76 |
| 7. | Future Scopes | 76 |
| 8. | Conclusion | 77 |
| 9. | Bibliography | 77 |
| 10. | Project Source Code  (Gituhub repository link) | 77 |

**INTRODUCTION**

**About Flutter:**

Flutter is one of the best solutions to develop apps for Android and iOS, without having to write in a different codebase for each platform. The smartphone versions of these apps function as true, native apps on Apple and Android devices and are compiled for the respective platform before publication. They do not need a runtime module or a browser. Using the same codebase, it is also possible to create web apps for browsers as well as native programs for Windows, Linux and macOS.

Google itself uses Flutter for several modules of the Google Assistant and for the user interface of the Google Home hub. Well known e-commerce service providers such as eBay, Groupon or Alibaba Group use Flutter to give their mobile and web apps a uniform look.

**About Our App E-Royal Mart:**

E-Royal Mart is an android application whose main purpose is to shopping in online , basically it is a E-commerce App .This application also helps to buy product (like- Cloth ,Electronic product , Medicine, Grocery etc).

For buying something people can order his product with the help of  this application .

**OBJECTIVE**

1. To help people find the good quality shopping product at the home.

2. It will help the people to buy easily necessary things at home.

3. Online and COD payment features (Google pay, phone pe, paytm).

4. Customer will get the service of online shopping and home delivery from his favorite shop.

5. It helps buying the products in the shop anywhere through internet on any platform.

**SCOPE**

E-Royal Mart is the only sector that frequently deals with necessary product. This project is an attempt to provide the advantages of online shopping to customers of a real shop. It helps buying the products in the shop anywhere through internet by using an android device. Thus the customer will get the service of online shopping and home delivery from his favorite shop. The E-commerce App services are capable of transforming the offline shop industry into a well-design E-commerce App for people.

***SYSTEM ANALYSIS***

**Feasibility Study**

The feasibility study is an evaluation and analysis of the  potential of a proposed project which is based on extensive  investigation and research to support the process of decision  making. Depending on the results of the initial investigation the  survey is now expanded to a more detailed feasibility study.

|  |
| --- |
| ● Feasibility study |
| ● System: E-commerce |
| ● Date: 26/01/2022 |
| ● Partners: TAYAN SINHA , ABANTIKA PANDIT  INDRANI DEY , UDDIPTA SARKAR |
| ● Module: 1 |
| ● Product |
| The project is an E-commerce application. It will help its users to buying the product through internet by using this application. |
| ● Technical Feasibility |
| The E-commerce will be developed using Flutter and Dart. The team is competent in that. |
| ● Social Feasibility |
| Some training for the users/admin is required and all users are Computer Application literate. |

**Workflow:**

In the context of our application we are using the Iterative SDLC Model because the Iterative model is repetition incarnate. Instead of starting with fully known requirements, you implement a set of software requirements, then test, evaluate and pinpoint further requirements.

A new version of the software is produced with each phase, or iteration. Rinse and repeat until the complete system is ready. In short,

Iterative development is a way of breaking down the software development of a large application into smaller pieces.

**Functional requirements:**

1. **Login in:** Users need to enter user name, email address and password which was provided while creating account.

2. **Sign up:** Users need to provide information like name, email address,  phone number then require to press the sign up button to  create an account.

3. **Main:** Users can choose to search for product according to  their requirements.

4. **Types:** This app is going to provide their users the information about  different types of product like cloths , electronic device, shoes, accessories etc.

5. **Product information according to users requirement:** Shows the information regarding the product according to user requirement, selected their favorite product and keep in wish list.  Here the user can order their favorite product.

**Non- functional requirements:**

**1. User information:** User must provide information like name, email  address, phone number, residential address, pin code ,gender.

**2. Internet connection:** An active internet connectivity is required to access this application.

**Software Requirements for Developers:**

Operating System: Windows 7 and above

Software: Android Studio 4.0.1

Toolkit: Flutter Development Toolkit (SDK)

Platform: Flutter and Dart

Backend Server: Firebase

**Hardware Requirements for Developers:**

Computer that has a 1.6Hz or faster processor

8 GB RAM (minimum)

HDD 20 GB Hard Disk Space and above 5400 RPM hard disk drive speed 7.0 minimum version (API 19)

**Software and Hardware Requirements for Users**

Android / IOS device / web platform

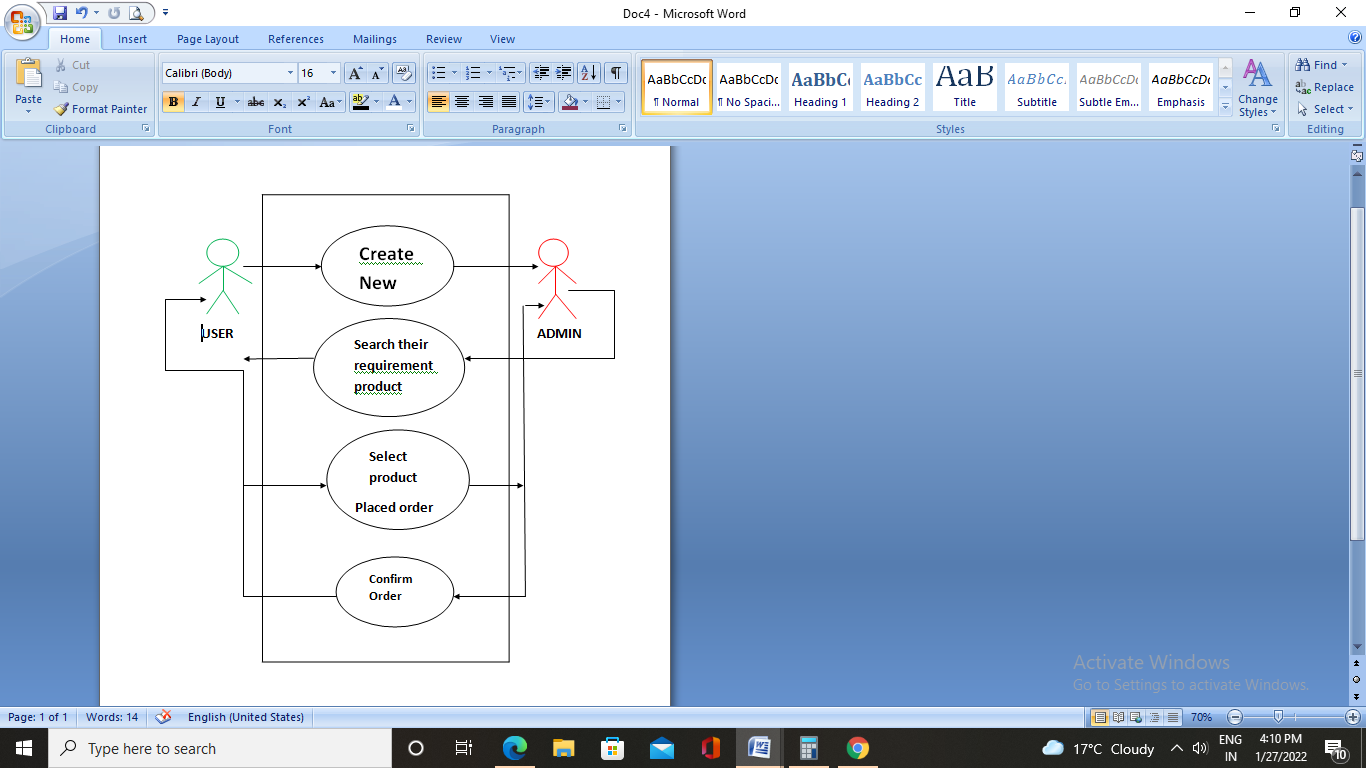
Android Version of 4.4 or above

2 GB RAM

Active Internet

**SYSTEM DESIGN**

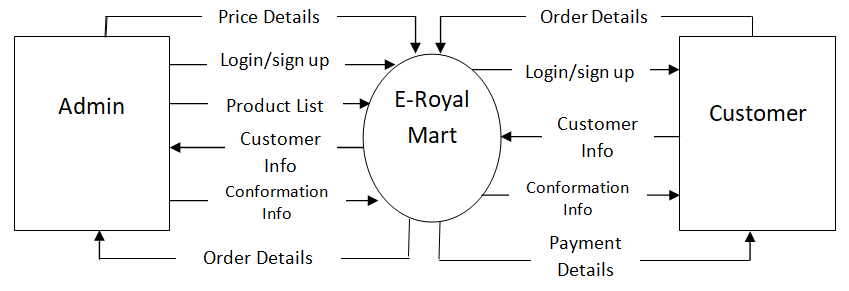
**USE CASE DIAGRAM**



**Data Flow Diagram (DFD) :**

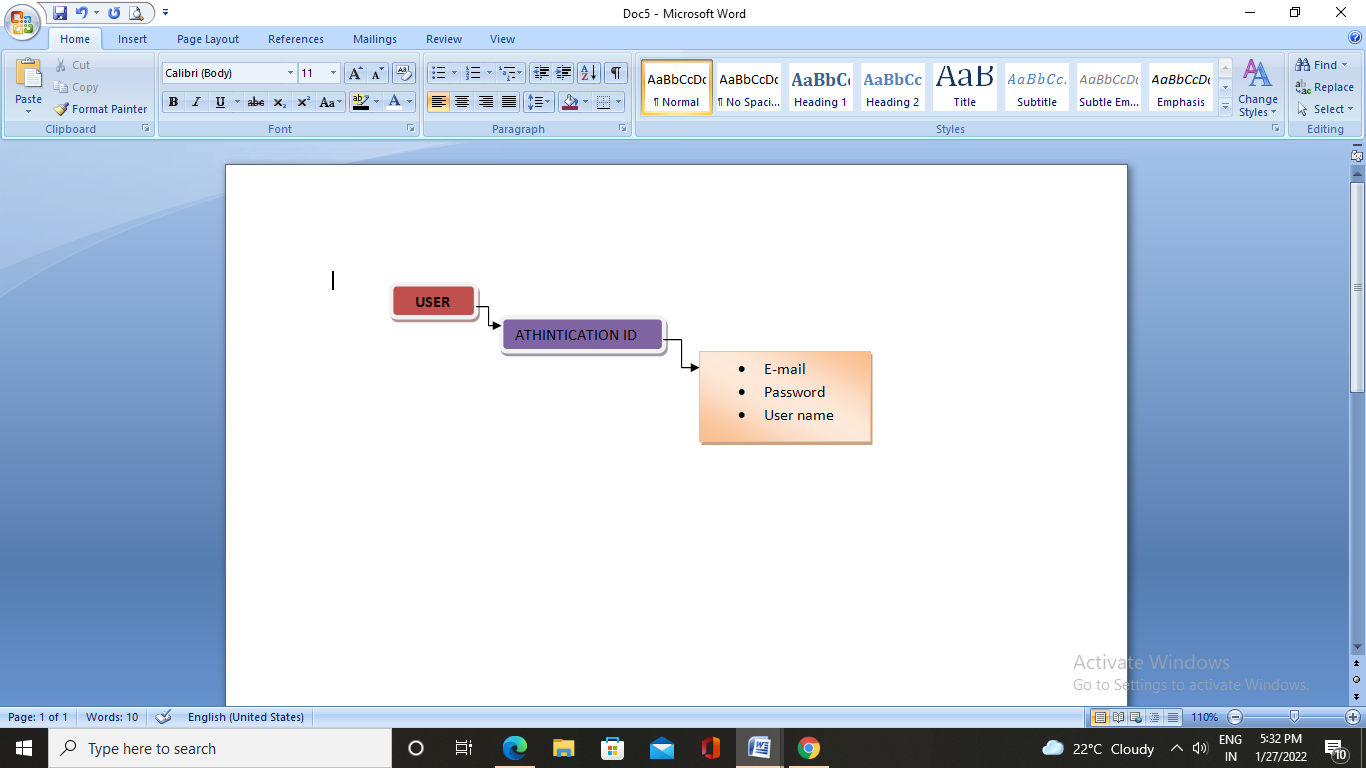
A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. A DFD is often used as a preliminary step to create an overview of the system. DFDs can also be used for the visualization of data processing.

A DFD shows what kind of information will be input to and the output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of process or information about whether processes will operate in sequence or parallel.



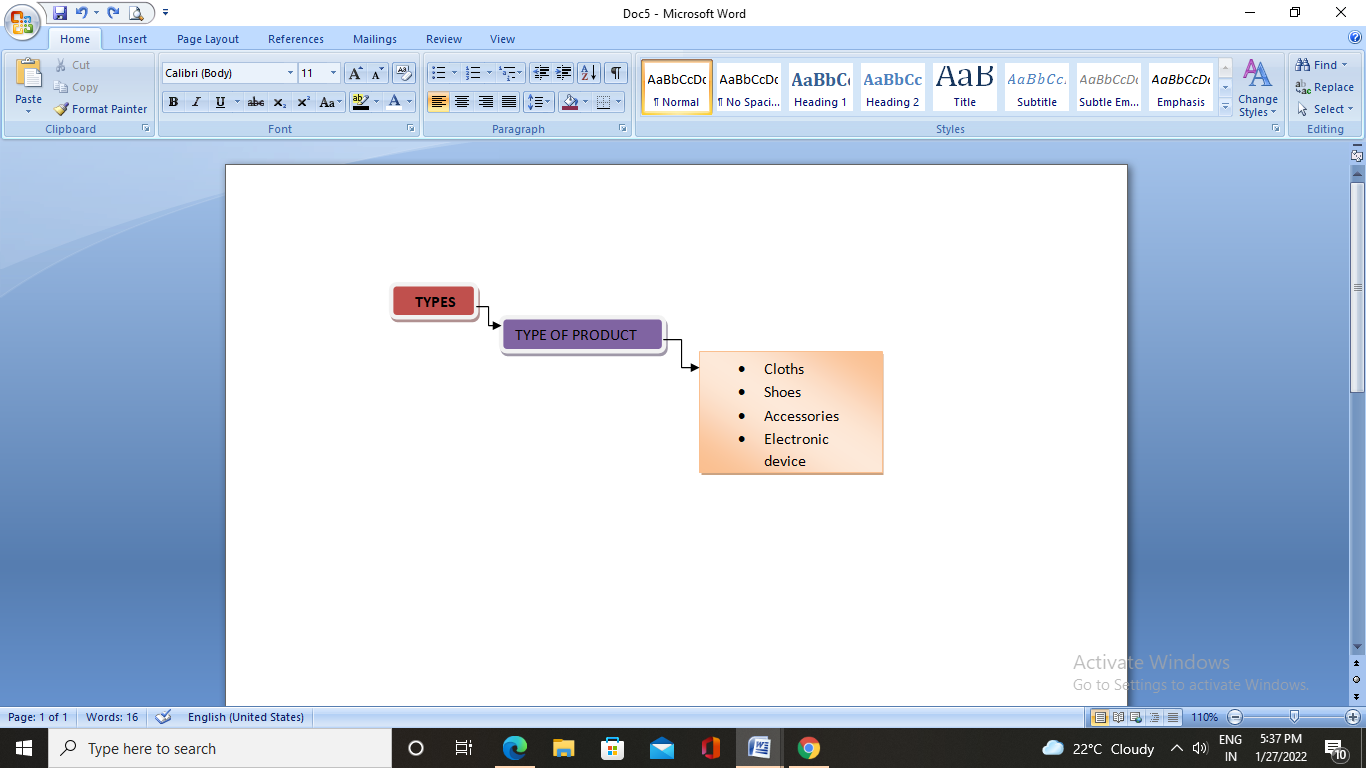
**DATABASE DIAGRAM**

R

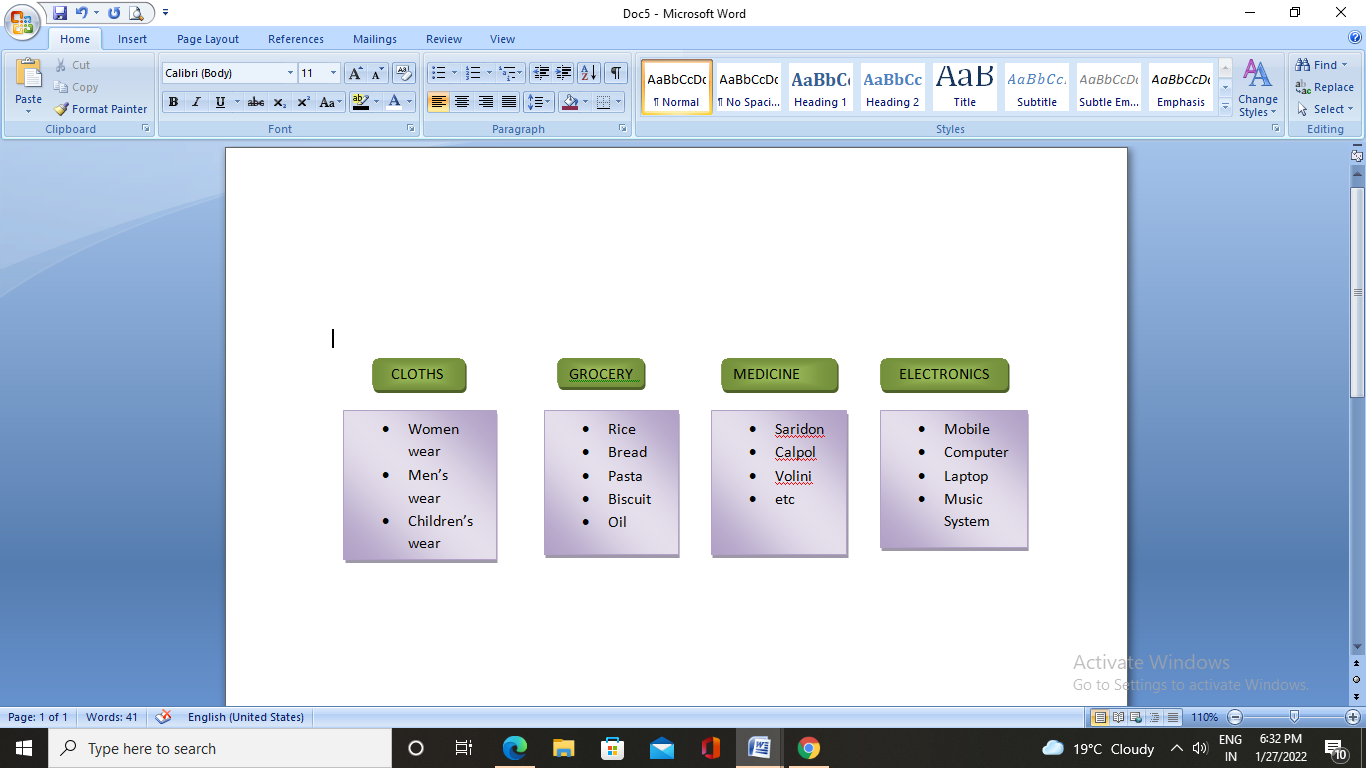


This database tells us the idea about the relationship of this   application with its user.

15

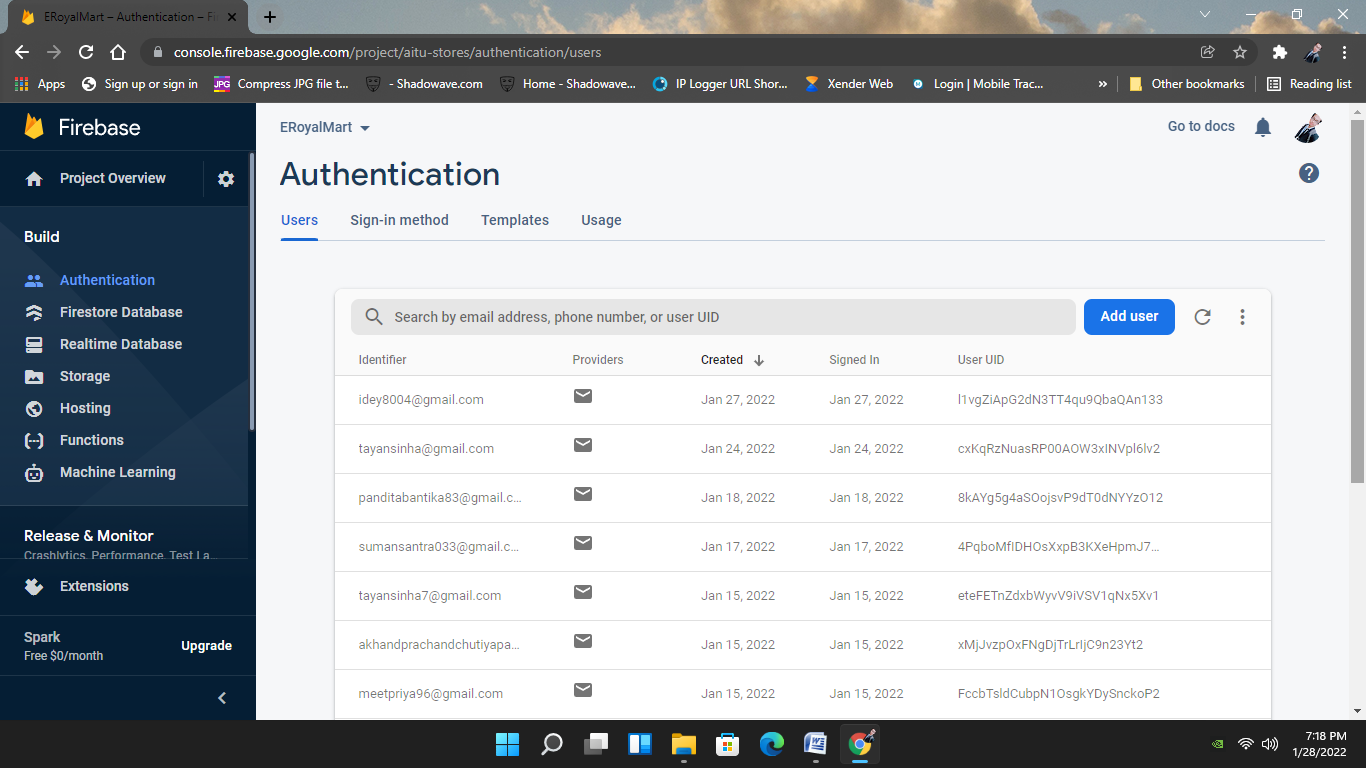


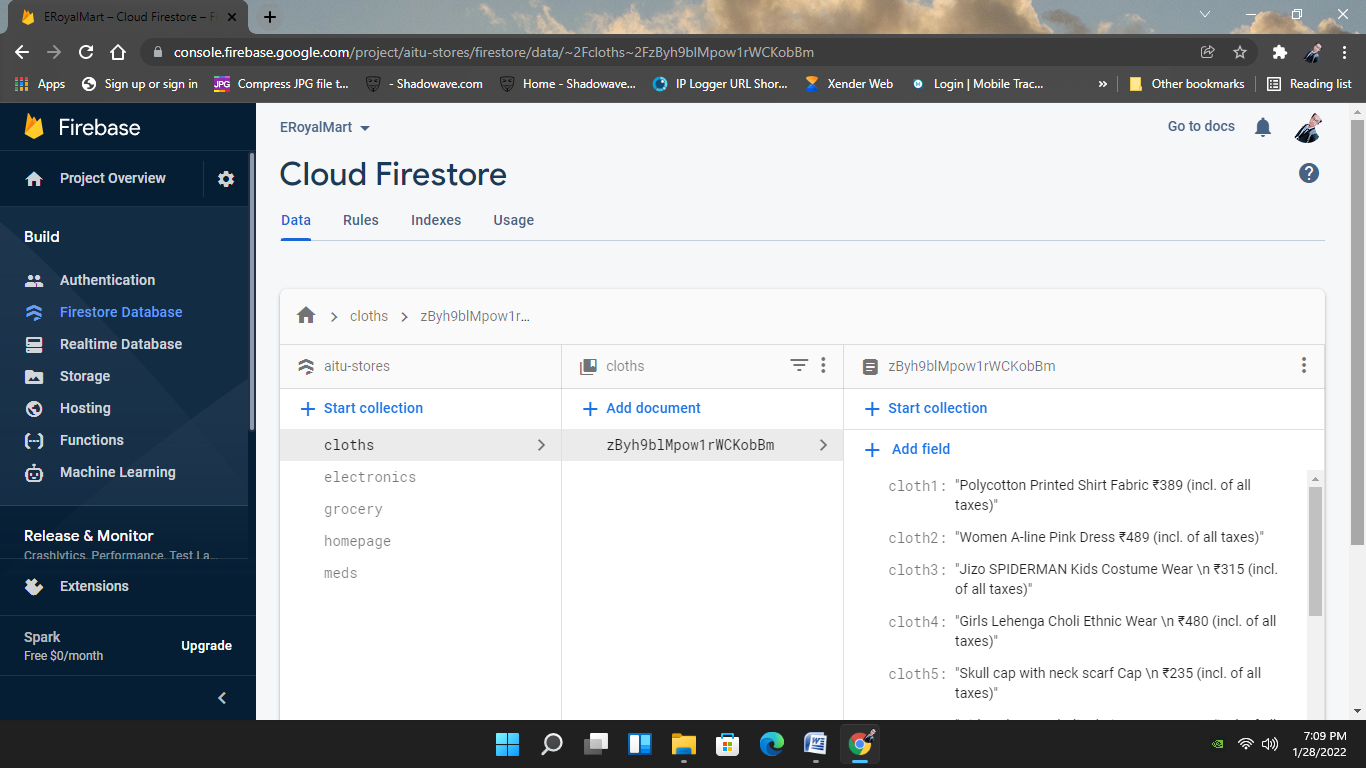
Through this database diagram, we get an idea about Type of things available on it.



Through this database gets the information regarding the   Products.

**DATABASE SCREENSHOTS**

****

****

**Coding**

Codes :

1. Main.dart

// ignore\_for\_file: use\_key\_in\_widget\_constructors, prefer\_const\_constructors, unused\_import, non\_constant\_identifier\_names, avoid\_print

// @dart=2.9

import 'dart:async';

import 'package:animated\_text\_kit/animated\_text\_kit.dart';

import 'package:animations/animations.dart';

import 'package:firebase\_auth/firebase\_auth.dart';

import 'package:firebase\_core/firebase\_core.dart';

import 'package:flutter/material.dart';

import 'package:google\_fonts/google\_fonts.dart';

import 'package:page\_transition/page\_transition.dart';

import 'package:sa/homepage.dart';

import 'package:sa/login.dart';

import 'package:sa/pages/electronics.dart';

import 'package:sa/pages/medicine.dart';

import 'package:sa/register.dart';

import 'package:shared\_preferences/shared\_preferences.dart';

Future<void> main() async {

WidgetsFlutterBinding.ensureInitialized();

WidgetsFlutterBinding.ensureInitialized();

await Firebase.initializeApp();

runApp(MyApp());

}

class MyApp extends StatelessWidget {

// This widget is the root of your application.

@override

Widget build(BuildContext context) {

return MaterialApp(

debugShowCheckedModeBanner: false,

title: 'ERoyalMart',

theme: ThemeData(

primarySwatch: Colors.red,

),

home: SecondClass(),

);

}

}

String user;

class SecondClass extends StatefulWidget {

@override

\_SecondClassState createState() => \_SecondClassState();

}

class \_SecondClassState extends State<SecondClass>

with TickerProviderStateMixin {

AnimationController scaleController;

Animation<double> scaleAnimation;

final FirebaseAuth auth = FirebaseAuth.instance;

@override

void initState() {

super.initState();

user = FirebaseAuth.instance.currentUser.toString();

print(user);

if (user != null) {

MaterialPageRoute(builder: (context) => HomePage());

} else {

MaterialPageRoute(builder: (context) => Login());

}

//hhhhh

scaleController = AnimationController(

vsync: this,

duration: Duration(milliseconds: 600),

)..addStatusListener(

(status) {

if (status == AnimationStatus.completed) {

Navigator.of(context).pushReplacement(

PageTransition(

type: PageTransitionType.bottomToTop,

child: Welcome(),

),

);

Timer(

Duration(milliseconds: 700),

() {

scaleController.reset();

},

);

}

},

);

scaleAnimation =

Tween<double>(begin: 0.0, end: 12).animate(scaleController);

Timer(Duration(seconds: 4), () {

setState(() {

scaleController.forward();

});

});

}

@override

void dispose() {

scaleController.dispose();

super.dispose();

}

@override

Widget build(BuildContext context) {

return Scaffold(

backgroundColor: Colors.black54,

body: Center(

child: DefaultTextStyle(

style: TextStyle(fontSize: 30.0),

child: AnimatedTextKit(

animatedTexts: [

TyperAnimatedText(

'E-Royal',

textStyle: GoogleFonts.indieFlower(

textStyle: TextStyle(fontSize: 50, color: Colors.white)),

speed: Duration(milliseconds: 150),

),

TyperAnimatedText(

' Mart',

textStyle: GoogleFonts.indieFlower(

textStyle: TextStyle(fontSize: 50, color: Colors.yellow)),

speed: Duration(milliseconds: 150),

),

],

isRepeatingAnimation: false,

repeatForever: false,

displayFullTextOnTap: false,

),

),

));

}

}

class Welcome extends StatelessWidget {

@override

Widget build(BuildContext context) {

return MaterialApp(

debugShowCheckedModeBanner: false,

home: Scaffold(

body: Scaffold(

backgroundColor: Colors.black,

body: SingleChildScrollView(

scrollDirection: Axis.vertical,

child: Column(

mainAxisAlignment: MainAxisAlignment.spaceAround,

children: <Widget>[

Image.asset(

"assets/images/welbg.jpg",

fit: BoxFit.contain,

),

SizedBox(

height: 50,

),

Column(children: [

Text("E-Royal ",

style: GoogleFonts.indieFlower(

textStyle: TextStyle(

fontSize: 60, color: Colors.white))),

Text("Mart",

style: GoogleFonts.indieFlower(

textStyle:

TextStyle(fontSize: 60, color: Colors.red)))

]),

Row(

children: [

SizedBox(

width: 150,

height: 200,

),

GestureDetector(

child: Text(

"Get Started",

style:

TextStyle(color: Colors.white, fontSize: 20),

),

onTap: () {

if (user == 'null') {

print("null");

Navigator.of(context).pushAndRemoveUntil(

MaterialPageRoute(

builder: (context) => Login()),

(Route<dynamic> route) => false);

} else {

print("not null");

Navigator.of(context).pushAndRemoveUntil(

MaterialPageRoute(

builder: (context) => HomePage()),

(Route<dynamic> route) => false);

}

},

),

IconButton(

color: Colors.white,

onPressed: () {

if (user == 'null') {

Navigator.push(

context,

MaterialPageRoute(

builder: (context) => Login()));

} else {

Navigator.push(

context,

MaterialPageRoute(

builder: (context) => HomePage()));

}

},

icon: const Icon(

Icons.arrow\_forward,

size: 15,

)),

],

),

],

),

))));

}

}

2. Login Activity

// ignore\_for\_file: prefer\_const\_constructors, unused\_field, non\_constant\_identifier\_names

import 'package:firebase\_auth/firebase\_auth.dart';

import 'package:flutter/material.dart';

import 'package:sa/homepage.dart';

import 'package:sa/pages/forgetpassword.dart';

import 'package:sa/register.dart';

import 'package:sa/utils/color.dart';

import 'package:sa/utils/reusable.dart';

class Login extends StatefulWidget {

const Login({Key? key}) : super(key: key);

@override

LoginState createState() => LoginState();

}

class LoginState extends State<Login> {

final TextEditingController \_emailTextController = TextEditingController();

final TextEditingController \_passwordTextController = TextEditingController();

@override

Widget build(BuildContext context) {

return Scaffold(

body: Container(

width: MediaQuery.of(context).size.width,

height: MediaQuery.of(context).size.height,

decoration: BoxDecoration(

gradient: LinearGradient(colors: [

hexStringColor("000001"),

hexStringColor("4d1521"),

hexStringColor("63133b"),

], begin: Alignment.topCenter, end: Alignment.bottomCenter)),

child: SingleChildScrollView(

child: Padding(

padding: EdgeInsets.fromLTRB(

20, MediaQuery.of(context).size.height \* 0.2, 20, 0),

child: Column(

children: <Widget>[

logoWidget("assets/images/logo.png"),

SizedBox(

height: 30,

),

reusableTextField(

"Enter Email Id",

Icons.person\_outline,

false,

\_emailTextController,

"^[a-zA-Z0-9+\_.-]+@[a-zA-Z0-9.-]+.[a-z]",

"Please Enter Your Email",

"Please Enter a valid email"),

SizedBox(

height: 10,

),

reusableTextField(

"Enter Password",

Icons.lock\_outline,

true,

\_passwordTextController,

"r'^.{6,}\$'",

"Password is required for login",

"Enter Valid Password(Min. 6 Character)"),

SizedBox(

height: 5,

),

forgotPassword(context),

firebaaseButton(context, "Sign In", () async {

FirebaseAuth.instance

.signInWithEmailAndPassword(

email: \_emailTextController.text,

password: \_passwordTextController.text)

.then((value) async {

Navigator.of(context).pushAndRemoveUntil(

MaterialPageRoute(builder: (context) => HomePage()),

(Route<dynamic> route) => false);

Navigator.push(

context,

MaterialPageRoute(builder: (context) => HomePage()),

);

ScaffoldMessenger.of(context).showSnackBar(SnackBar(

content: Text("Signed In Sucessfully",

style: TextStyle(

color: Colors.white,

fontSize: 15,

fontWeight: FontWeight.bold)),

duration: Duration(seconds: 5),

backgroundColor: Colors.red[300],

));

}).onError((error, stackTrace) {

String err = (" Error: ${error.toString()}");

ScaffoldMessenger.of(context).showSnackBar(

SnackBar(

content: Text(err,

style: TextStyle(

color: Colors.white,

fontSize: 15,

fontWeight: FontWeight.bold)),

duration: Duration(seconds: 5),

backgroundColor: Colors.red[300],

),

);

});

}),

signUpOption(),

SizedBox(height: 30)

],

),

),

),

),

);

}

Row signUpOption() {

return Row(

mainAxisAlignment: MainAxisAlignment.center,

children: [

Text(

"Dont have account?",

style: TextStyle(color: Colors.white70),

),

GestureDetector(

onTap: () {

Navigator.push(

context, MaterialPageRoute(builder: (context) => Register()));

},

child: Text(

" Sign Up",

style: TextStyle(color: Colors.white, fontWeight: FontWeight.bold),

),

)

],

);

}

Widget forgotPassword(BuildContext context) {

return Container(

width: MediaQuery.of(context).size.width,

height: 35,

alignment: Alignment.bottomRight,

child: TextButton(

child: Text(

"forget password?",

style: TextStyle(color: Colors.white70),

textAlign: TextAlign.right,

),

onPressed: () => Navigator.push(

context, MaterialPageRoute(builder: (context) => ForgetPassword())),

),

);

}

}

1. Register

// ignore\_for\_file: avoid\_print, prefer\_const\_constructors

// import 'package:firebase\_auth/firebase\_auth.dart';

import 'dart:ui';

import 'package:firebase\_auth/firebase\_auth.dart';

import 'package:sa/homepage.dart';

import 'package:sa/utils/color.dart';

import 'package:flutter/material.dart';

import 'package:sa/utils/reusable.dart';

class Register extends StatefulWidget {

const Register({Key? key}) : super(key: key);

@override

\_RegisterState createState() => \_RegisterState();

}

class \_RegisterState extends State<Register> {

final TextEditingController \_passwordTextController = TextEditingController();

final TextEditingController \_emailTextController = TextEditingController();

final TextEditingController \_userNameTextController = TextEditingController();

@override

Widget build(BuildContext context) {

return Scaffold(

extendBodyBehindAppBar: true,

appBar: AppBar(

backgroundColor: Colors.transparent,

elevation: 0,

title: Row(children: [

const Text(

"Sign Up",

style: TextStyle(fontSize: 24, fontWeight: FontWeight.bold),

),

Padding(

padding: const EdgeInsets.only(left: 170.0),

child: logoWidget2(

"assets/images/logo.png",

))

])),

body: Container(

width: MediaQuery.of(context).size.width,

height: MediaQuery.of(context).size.height,

decoration: BoxDecoration(

gradient: LinearGradient(colors: [

hexStringColor("000001"),

hexStringColor("4d1521"),

hexStringColor("63133b"),

], begin: Alignment.topCenter, end: Alignment.bottomCenter)),

child: SingleChildScrollView(

child: Padding(

padding: const EdgeInsets.fromLTRB(20, 120, 20, 0),

child: Column(

children: <Widget>[

const SizedBox(

height: 20,

),

const SizedBox(

height: 30,

),

reusableTextField(

"Enter UserName",

Icons.person\_outline,

false,

\_userNameTextController,

"",

"Please Enter Your UserName",

"Please Enter a valid UserName"),

const SizedBox(

height: 20,

),

reusableTextField(

"Enter Email Id",

Icons.person\_outline,

false,

\_emailTextController,

"^[a-zA-Z0-9+\_.-]+@[a-zA-Z0-9.-]+.[a-z]",

"Please Enter Your Email",

"Please Enter a valid email"),

const SizedBox(

height: 20,

),

reusableTextField(

"Enter Password (minimum 6 characters)",

Icons.lock\_outlined,

true,

\_passwordTextController,

"r'^.{6,}\$'",

"Password is required for login",

"Enter Valid Password(Min. 6 Character)"),

const SizedBox(

height: 20,

),

firebaaseButton(context, "Sign Up", () {

FirebaseAuth.instance

.createUserWithEmailAndPassword(

email: \_emailTextController.text,

password: \_passwordTextController.text)

.then((value) {

ScaffoldMessenger.of(context).showSnackBar(SnackBar(

content: Text("Registered Sucessfully",

style: TextStyle(

color: Colors.white,

fontSize: 15,

fontWeight: FontWeight.bold)),

duration: Duration(seconds: 5),

backgroundColor: Colors.red[300],

));

Navigator.of(context).pushAndRemoveUntil(

MaterialPageRoute(builder: (context) => HomePage()),

(Route<dynamic> route) => false);

}).onError((error, stackTrace) {

String err = (" Error: ${error.toString()}");

ScaffoldMessenger.of(context).showSnackBar(

SnackBar(

content: Text(err,

style: TextStyle(

color: Colors.white,

fontSize: 15,

fontWeight: FontWeight.bold)),

duration: Duration(seconds: 5),

backgroundColor: Colors.black,

),

);

});

})

],

),

))),

);

}

}

1. Home page

// ignore\_for\_file: prefer\_const\_constructors, prefer\_const\_literals\_to\_create\_immutables,, import\_of\_legacy\_library\_into\_null\_safe

import 'package:carousel\_slider/carousel\_slider.dart';

import 'package:firebase\_auth/firebase\_auth.dart';

import 'package:flutter/material.dart';

import 'package:google\_fonts/google\_fonts.dart';

import 'package:sa/login.dart';

import 'package:sa/pages/aboutus.dart';

import 'package:sa/pages/clothes.dart';

import 'package:sa/pages/electronics.dart';

import 'package:sa/pages/grocery.dart';

import 'package:sa/pages/medicine.dart';

import 'package:sa/pages/search.dart';

import 'package:sa/pages/settings.dart';

import 'package:sa/utils/reusable.dart';

import 'package:cloud\_firestore/cloud\_firestore.dart';

class HomePage extends StatelessWidget {

const HomePage({Key? key}) : super(key: key);

@override

Widget build(BuildContext context) {

final drawerHeader = UserAccountsDrawerHeader(

accountName: reusabletext("E-Royal Mart", 20, Colors.white),

accountEmail:

reusabletext("eroyalmartsupport@yahoo.com", 20, Colors.white),

decoration: BoxDecoration(color: Colors.red[300]),

currentAccountPicture: CircleAvatar(

backgroundColor: Colors.black,

child: logoWidget2("assets/images/logo.png"),

),

);

final drawerItems = ListView(

children: <Widget>[

drawerHeader,

SizedBox(

height: 50,

),

ListTile(

title: Text('My Orders',

style: GoogleFonts.comfortaa(

textStyle: TextStyle(fontSize: 25, color: Colors.black))),

onTap: () {

ScaffoldMessenger.of(context).showSnackBar(SnackBar(

content: Text("Your OrderList is Empty",

style: TextStyle(

color: Colors.white,

fontSize: 15,

fontWeight: FontWeight.bold)),

duration: Duration(seconds: 5),

backgroundColor: Colors.red[300],

));

}),

SizedBox(

height: 20,

),

ListTile(

title: Text('About Us',

style: GoogleFonts.comfortaa(

textStyle: TextStyle(fontSize: 25, color: Colors.black))),

onTap: () => Navigator.push(

context, MaterialPageRoute(builder: (context) => Aboutus()))),

SizedBox(

height: 20,

),

ListTile(

title: Text('Settings',

style: GoogleFonts.comfortaa(

textStyle: TextStyle(fontSize: 25, color: Colors.black))),

onTap: () => Navigator.push(

context, MaterialPageRoute(builder: (context) => Settingss()))),

SizedBox(

height: 20,

),

ListTile(

title: Text('Sign Out',

style: GoogleFonts.comfortaa(

textStyle: TextStyle(fontSize: 25, color: Colors.black))),

onTap: () {

FirebaseAuth.instance.signOut().then(

(value) {

ScaffoldMessenger.of(context).showSnackBar(SnackBar(

content: Text("Signed Out Sucessfully",

style: TextStyle(

color: Colors.white,

fontSize: 15,

fontWeight: FontWeight.bold)),

duration: Duration(seconds: 5),

backgroundColor: Colors.red[300],

));

Navigator.push(context,

MaterialPageRoute(builder: (context) => const Login()));

},

);

}),

],

);

CollectionReference homepage =

FirebaseFirestore.instance.collection('homepage');

return Scaffold(

backgroundColor: Colors.yellow[50],

appBar: AppBar(

backgroundColor: Colors.red[300],

title: Row(

mainAxisAlignment: MainAxisAlignment.spaceAround,

children: [

Text('E-Royal Mart',

style: GoogleFonts.indieFlower(

textStyle:

TextStyle(fontSize: 30, color: Colors.white))),

IconButton(

icon: const Icon(Icons.search),

onPressed: () {

Navigator.push(context,

MaterialPageRoute(builder: (context) => Search()));

},

)

])),

body: SingleChildScrollView(

scrollDirection: Axis.vertical,

child: FutureBuilder(

future: homepage.doc('QMj48L7eoTkeq1UEldUb').get(),

builder: (BuildContext context, AsyncSnapshot snapshot) {

if (snapshot.hasData && !snapshot.data.exists) {

return Text("Document does not exist");

}

if (!snapshot.hasData) {

return Column(

children: const [

Text(

"Loading... Please wait",

style: TextStyle(fontSize: 30),

),

Icon(

Icons.do\_not\_touch,

size: 50,

),

],

);

}

Map<String, dynamic> homepage =

snapshot.data.data() as Map<String, dynamic>;

final List<String> imagesList = [

'${homepage['dealsforyou1']}',

'${homepage['dealsforyou2']}',

'${homepage['dealsforyou3']}',

'${homepage['dealsforyou4']}',

];

return Column(children: [

Row(children: [

SizedBox(

width: 30,

),

Card(

child: SizedBox(

height: 170,

width: 170,

child: Column(

mainAxisAlignment: MainAxisAlignment.spaceAround,

children: [

SizedBox(

height: 10,

),

GestureDetector(

child: Image.asset(

"assets/images/electronic.jpeg",

),

onTap: () {

Navigator.push(

context,

MaterialPageRoute(

builder: (context) => Electronics()));

},

),

SizedBox(

height: 10,

),

Text("ELECTRONICS",

style: GoogleFonts.comfortaa(

textStyle: TextStyle(

fontSize: 15, color: Colors.blue))),

],

)),

),

Card(

child: SizedBox(

height: 170,

width: 170,

child: Column(

mainAxisAlignment: MainAxisAlignment.spaceAround,

children: [

SizedBox(

height: 10,

),

GestureDetector(

child: Image.asset(

"assets/images/cloth.jpg",

),

onTap: () {

Navigator.push(

context,

MaterialPageRoute(

builder: (context) => Clothes()));

},

),

SizedBox(

height: 10,

),

Text("CLOTHES",

style: GoogleFonts.comfortaa(

textStyle: TextStyle(

fontSize: 15, color: Colors.blue))),

],

)),

),

]),

Row(children: [

SizedBox(

width: 30,

),

Card(

child: SizedBox(

height: 170,

width: 170,

child: Column(

mainAxisAlignment: MainAxisAlignment.spaceAround,

children: [

SizedBox(

height: 10,

),

GestureDetector(

child: Image.asset(

"assets/images/grocery.jpg",

),

onTap: () {

Navigator.push(

context,

MaterialPageRoute(

builder: (context) => Grocery()));

},

),

SizedBox(

height: 10,

),

Text("GROCERY",

style: GoogleFonts.comfortaa(

textStyle: TextStyle(

fontSize: 15, color: Colors.blue))),

],

)),

),

Card(

child: SizedBox(

height: 170,

width: 170,

child: Column(

mainAxisAlignment: MainAxisAlignment.spaceAround,

children: [

SizedBox(

height: 10,

),

GestureDetector(

child: Image.asset(

"assets/images/medicine.jpg",

),

onTap: () {

Navigator.push(

context,

MaterialPageRoute(

builder: (context) => Medicine()));

},

),

SizedBox(

height: 10,

),

Text("MEDICINE",

style: GoogleFonts.comfortaa(

textStyle: TextStyle(

fontSize: 15, color: Colors.blue))),

],

)),

),

]),

Card(

child: SizedBox(

height: 400,

child: Column(

children: [

SizedBox(

height: 30,

),

Text(

"Deal of the Day,",

style: GoogleFonts.comfortaa(

textStyle:

TextStyle(fontSize: 27, color: Colors.red)),

),

SizedBox(

height: 0,

),

GestureDetector(

child: Image.network(

("${homepage['dealofthedayimg']}"),

width: 400,

height: 250,

),

onTap: () {

Navigator.push(

context,

MaterialPageRoute(

builder: (context) => Electronics()));

},

),

Text("${homepage['dealoftheday']}",

style: GoogleFonts.comfortaa(

textStyle: TextStyle(

fontSize: 15, color: Colors.black))),

Text("${homepage['dealoftheday2']}",

style: GoogleFonts.comfortaa(

textStyle: TextStyle(

fontSize: 15, color: Colors.blue)))

],

),

),

),

SizedBox(

height: 19,

),

Text(

"Deals For You",

style: GoogleFonts.comfortaa(

textStyle: TextStyle(fontSize: 27, color: Colors.red)),

),

SizedBox(

height: 19,

),

CarouselSlider(

options: CarouselOptions(

viewportFraction: 0.6,

autoPlayAnimationDuration:

const Duration(milliseconds: 100),

enlargeCenterPage: true,

autoPlay: true,

),

items: imagesList

.map(

(item) => Center(

child: Image.network(

item,

fit: BoxFit.cover,

width: 300,

height: 400,

),

),

)

.toList(),

),

SizedBox(

height: 29,

),

Card(

child: SizedBox(

height: 400,

width: 400,

child: Column(

children: [

SizedBox(

height: 25,

),

Text("Up to 42% off | Clothes & appliances",

style: GoogleFonts.comfortaa(

textStyle: TextStyle(

fontSize: 15, color: Colors.red))),

SizedBox(

height: 25,

),

GestureDetector(

child: Image.asset(

"assets/images/cloth2.jpg",

),

onTap: () {

Navigator.push(

context,

MaterialPageRoute(

builder: (context) => Clothes()));

},

),

Row(children: [

SizedBox(

width: 50,

height: 70,

),

Icon(

Icons.card\_giftcard,

color: Colors.green,

),

SizedBox(

width: 10,

),

Text(

"No Cost EMI",

style: GoogleFonts.comfortaa(

textStyle: TextStyle(

fontSize: 15, color: Colors.green)),

),

SizedBox(

width: 10,

),

Icon(

Icons.card\_travel,

color: Colors.green,

),

SizedBox(

width: 10,

),

Text(

"Sheduled Delivery",

style: GoogleFonts.comfortaa(

textStyle: TextStyle(

fontSize: 15, color: Colors.green)),

)

])

],

),

),

),

SizedBox(

height: 20,

),

Text(

"Interesting Finds",

style: GoogleFonts.comfortaa(

textStyle: TextStyle(fontSize: 30, color: Colors.red)),

),

Row(children: [

SizedBox(

height: 10,

width: 20,

),

Card(

child: SizedBox(

height: 200,

width: 180,

child: Column(

mainAxisAlignment: MainAxisAlignment.spaceAround,

children: [

SizedBox(

height: 10,

),

GestureDetector(

child: Image.asset(

"assets/images/if.jpg",

),

onTap: () {

Navigator.push(

context,

MaterialPageRoute(

builder: (context) => Electronics()));

},

),

SizedBox(

height: 10,

),

Text("Mobile Stands",

style: GoogleFonts.comfortaa(

textStyle: TextStyle(

fontSize: 25, color: Colors.black))),

Text(

"From ₹299 ",

style: GoogleFonts.comfortaa(

textStyle: TextStyle(color: Colors.green)),

)

],

)),

),

Card(

child: SizedBox(

height: 200,

width: 180,

child: Column(

mainAxisAlignment: MainAxisAlignment.spaceAround,

children: [

SizedBox(

height: 10,

),

GestureDetector(

child: Image.asset(

"assets/images/if2.jpg",

),

onTap: () {

Navigator.push(

context,

MaterialPageRoute(

builder: (context) => Electronics()));

},

),

SizedBox(

height: 10,

),

Text("Furniture",

style: GoogleFonts.comfortaa(

textStyle: TextStyle(

fontSize: 25, color: Colors.black))),

Text(

"Up to 80% off ",

style: GoogleFonts.comfortaa(

textStyle: TextStyle(color: Colors.green)),

)

],

)),

),

]),

Row(children: [

SizedBox(

height: 10,

width: 20,

),

Card(

child: SizedBox(

height: 200,

width: 180,

child: Column(

mainAxisAlignment: MainAxisAlignment.spaceAround,

children: [

SizedBox(

height: 10,

),

GestureDetector(

child: Image.asset(

"assets/images/if3.jpg",

),

onTap: () {

Navigator.push(

context,

MaterialPageRoute(

builder: (context) => Medicine()));

},

),

SizedBox(

height: 5,

),

Text("Medicines",

style: GoogleFonts.comfortaa(

textStyle: TextStyle(

fontSize: 25, color: Colors.black))),

Text(

"At you doresteps 24x7 ",

style: GoogleFonts.comfortaa(

textStyle: TextStyle(color: Colors.green)),

)

],

)),

),

Card(

child: SizedBox(

height: 200,

width: 180,

child: Column(

mainAxisAlignment: MainAxisAlignment.spaceAround,

children: [

SizedBox(

height: 10,

),

GestureDetector(

child: Image.asset(

"assets/images/if4.jpg",

),

onTap: () {

Navigator.push(

context,

MaterialPageRoute(

builder: (context) => Grocery()));

},

),

SizedBox(

height: 10,

),

Text("Vegitable",

style: GoogleFonts.comfortaa(

textStyle: TextStyle(

fontSize: 25, color: Colors.black))),

Text(

"Delivery Free for first order ",

style: GoogleFonts.comfortaa(

textStyle: TextStyle(color: Colors.green)),

),

],

)),

),

]),

]);

},

),

),

drawer: Drawer(

child: Container(

color: Colors.yellow[50],

child: drawerItems,

)));

}

}

1. Settings

// @dart=2.9

import 'package:firebase\_auth/firebase\_auth.dart';

import 'package:flutter/material.dart';

import 'package:google\_fonts/google\_fonts.dart';

import 'package:sa/login.dart';

import 'package:sa/pages/forgetpassword.dart';

import 'package:sa/utils/reusable.dart';

class Settingss extends StatelessWidget {

const Settingss({Key key}) : super(key: key);

@override

Widget build(BuildContext context) {

String useremail;

bool emailvarify;

useremail = FirebaseAuth.instance.currentUser.email;

emailvarify = FirebaseAuth.instance.currentUser.emailVerified;

return Scaffold(

backgroundColor: Colors.pink[100],

appBar: AppBar(

backgroundColor: Colors.red[300],

title: reusabletext("Settings", 30, Colors.white),

),

body: SingleChildScrollView(

scrollDirection: Axis.vertical,

child: Column(children: [

const SizedBox(

height: 50,

),

Text('Email: $useremail',

style: GoogleFonts.comfortaa(

textStyle:

const TextStyle(fontSize: 20, color: Colors.brown))),

Text('Email varification: $emailvarify',

style: GoogleFonts.comfortaa(

textStyle:

const TextStyle(fontSize: 20, color: Colors.brown))),

const SizedBox(

height: 350,

),

GestureDetector(

child: firebaaseButton(context, "Sign Out", () {

FirebaseAuth.instance.signOut().then(

(value) {

Navigator.push(context,

MaterialPageRoute(builder: (context) => const Login()));

ScaffoldMessenger.of(context).showSnackBar(const SnackBar(

content: Text("Signed Out Sucessfully",

style: TextStyle(

color: Colors.white,

fontSize: 15,

fontWeight: FontWeight.bold)),

duration: Duration(seconds: 5),

backgroundColor: Colors.redAccent,

));

},

);

})),

GestureDetector(

child: firebaaseButton(context, "Delete Account", () {

FirebaseAuth.instance.currentUser.delete().then(

(value) {

Navigator.push(context,

MaterialPageRoute(builder: (context) => const Login()));

ScaffoldMessenger.of(context).showSnackBar(const SnackBar(

content: Text("User Deleted",

style: TextStyle(

color: Colors.white,

fontSize: 15,

fontWeight: FontWeight.bold)),

duration: Duration(seconds: 5),

backgroundColor: Colors.redAccent,

));

},

).onError((error, stackTrace) {

ScaffoldMessenger.of(context).showSnackBar(const SnackBar(

content: Text("User Deleted",

style: TextStyle(

color: Colors.white,

fontSize: 15,

fontWeight: FontWeight.bold)),

duration: Duration(seconds: 5),

backgroundColor: Colors.redAccent,

));

});

})),

firebaaseButton(

context,

"Reset Password",

() {

Navigator.push(

context,

MaterialPageRoute(

builder: (context) => const ForgetPassword()));

},

)

])));

}

}

1. Products

// ignore\_for\_file: prefer\_const\_constructors\_in\_immutables, avoid\_unnecessary\_containers, prefer\_const\_constructors, use\_key\_in\_widget\_constructors

// @dart=2.9

import 'package:cloud\_firestore/cloud\_firestore.dart';

import 'package:flutter/material.dart';

import 'package:sa/pages/checkoutpage.dart';

// ignore: must\_be\_immutable

class Electronics extends StatefulWidget {

FirebaseFirestore firestore = FirebaseFirestore.instance;

@override

\_ElectronicsState createState() => \_ElectronicsState();

}

class \_ElectronicsState extends State<Electronics> {

@override

Widget build(BuildContext context) {

CollectionReference elecdetail =

FirebaseFirestore.instance.collection('electronics');

return Scaffold(

// SliverAppBar is declared in Scaffold.body, in slivers of a

// CustomScrollView.

body: CustomScrollView(

slivers: <Widget>[

SliverAppBar(

expandedHeight: 160.0,

flexibleSpace: FlexibleSpaceBar(

title: const Text("Electronics"),

background: Image.asset(

'assets/images/electronicbackground.jpg',

fit: BoxFit.fill,

),

),

),

SliverFillRemaining(

child: SingleChildScrollView(

scrollDirection: Axis.vertical,

child: FutureBuilder(

future: elecdetail.doc('azv143xeioLShsyIXzUL').get(),

builder: (BuildContext context,

AsyncSnapshot<DocumentSnapshot> snapshot) {

if (snapshot.hasData && !snapshot.data.exists) {

return Text("Document does not exist");

}

if (!snapshot.hasData) {

return Container(

child: Column(

children: const [

Text(

"Loading... Please wait",

style: TextStyle(fontSize: 30),

),

Icon(

Icons.do\_not\_touch,

size: 50,

),

],

));

}

Map<String, dynamic> elecdetail =

snapshot.data.data() as Map<String, dynamic>;

return Column(

children: <Widget>[

SizedBox(

width: 30,

),

Card(

child: SizedBox(

height: 500,

width: 350,

child: Column(

mainAxisAlignment: MainAxisAlignment.spaceAround,

children: [

SizedBox(

height: 10,

),

GestureDetector(

child: Image.network(

("${elecdetail['image1']}")),

onTap: () {

Navigator.push(

context,

MaterialPageRoute(

builder: (context) => Checkout()));

},

),

Text(

"${elecdetail['product1']}",

style: TextStyle(

fontSize: 20, color: Colors.blue),

),

],

)),

),

SizedBox(

width: 30,

),

Card(

child: SizedBox(

height: 500,

width: 350,

child: Column(

mainAxisAlignment: MainAxisAlignment.spaceAround,

children: [

SizedBox(

height: 10,

),

GestureDetector(

child: Image.network(

("${elecdetail['image2']}")),

onTap: () {

Navigator.push(

context,

MaterialPageRoute(

builder: (context) => Checkout()));

},

),

Text(

"${elecdetail['product2']}",

style: TextStyle(

fontSize: 20, color: Colors.blue),

),

],

)),

),

SizedBox(

width: 30,

),

Card(

child: SizedBox(

height: 500,

width: 350,

child: Column(

mainAxisAlignment: MainAxisAlignment.spaceAround,

children: [

SizedBox(

height: 10,

),

GestureDetector(

child: Image.network(

("${elecdetail['image3']}")),

onTap: () {

Navigator.push(

context,

MaterialPageRoute(

builder: (context) => Checkout()));

},

),

Text(

"${elecdetail['product3']}",

style: TextStyle(

fontSize: 20, color: Colors.blue),

),

],

)),

),

SizedBox(

width: 30,

),

Card(

child: SizedBox(

height: 500,

width: 350,

child: Column(

mainAxisAlignment: MainAxisAlignment.spaceAround,

children: [

SizedBox(

height: 10,

),

GestureDetector(

child: Image.network(

("${elecdetail['image4']}")),

onTap: () {

Navigator.push(

context,

MaterialPageRoute(

builder: (context) => Checkout()));

},

),

Text(

"${elecdetail['product4']}",

style: TextStyle(

fontSize: 20, color: Colors.blue),

),

],

)),

),

SizedBox(

width: 30,

),

Card(

child: SizedBox(

height: 500,

width: 350,

child: Column(

mainAxisAlignment: MainAxisAlignment.spaceAround,

children: [

SizedBox(

height: 10,

),

GestureDetector(

child: Image.network(

("${elecdetail['image5']}")),

onTap: () {

Navigator.push(

context,

MaterialPageRoute(

builder: (context) => Checkout()));

},

),

Text(

"${elecdetail['product5']}",

style: TextStyle(

fontSize: 20, color: Colors.blue),

),

],

)),

),

SizedBox(

width: 30,

),

Card(

child: SizedBox(

height: 500,

width: 350,

child: Column(

mainAxisAlignment: MainAxisAlignment.spaceAround,

children: [

SizedBox(

height: 10,

),

GestureDetector(

child: Image.network(

("${elecdetail['image6']}")),

onTap: () {

Navigator.push(

context,

MaterialPageRoute(

builder: (context) => Checkout()));

},

),

Text(

" ${elecdetail['product6']}",

style: TextStyle(

fontSize: 20, color: Colors.blue),

),

],

)),

),

SizedBox(

width: 30,

),

Card(

child: SizedBox(

height: 500,

width: 350,

child: Column(

mainAxisAlignment: MainAxisAlignment.spaceAround,

children: [

SizedBox(

height: 10,

),

GestureDetector(

child: Image.network(

("${elecdetail['image7']}")),

onTap: () {

Navigator.push(

context,

MaterialPageRoute(

builder: (context) => Checkout()));

},

),

Text(

"${elecdetail['product7']}",

style: TextStyle(

fontSize: 20, color: Colors.blue),

),

],

)),

),

SizedBox(

width: 30,

),

Card(

child: SizedBox(

height: 500,

width: 350,

child: Column(

mainAxisAlignment: MainAxisAlignment.spaceAround,

children: [

SizedBox(

height: 10,

),

GestureDetector(

child: Image.network(

("${elecdetail['image8']}")),

onTap: () {

Navigator.push(

context,

MaterialPageRoute(

builder: (context) => Checkout()));

},

),

Text(

"${elecdetail['product8']}",

style: TextStyle(

fontSize: 20, color: Colors.blue),

),

],

)),

),

],

);

},

),

),

),

],

),

);

}

}

1. Checkout

// ignore\_for\_file: prefer\_const\_literals\_to\_create\_immutables, prefer\_const\_constructors, deprecated\_member\_use, prefer\_const\_constructors\_in\_immutables

// import 'dart:html';

import 'package:flutter/material.dart';

class Checkout extends StatefulWidget {

Checkout({Key? key}) : super(key: key);

@override

\_CheckoutState createState() => \_CheckoutState();

}

class \_CheckoutState extends State<Checkout> {

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

backgroundColor: Colors.red[300],

title: Text("Select Your Payment Method"),

),

body: SingleChildScrollView(

scrollDirection: Axis.vertical,

child: Center(

child: Column(

children: [

SizedBox(

height: 50,

),

Text(

"Select a way to pay",

style: TextStyle(fontSize: 30),

),

SizedBox(

height: 50,

),

GestureDetector(

child: SizedBox(

height: 600,

child: Column(

children: [

Card(

child: SizedBox(

height: 50,

width: 350,

child: Row(

mainAxisAlignment:

MainAxisAlignment.spaceEvenly,

children: [

// Icon(Icons.rounded\_corner),

Column(

children: [

Text(

"UPI/NetBanking",

style: TextStyle(

fontSize: 20, color: Colors.blue),

),

Text(

"(Pay direct from bank)",

style: TextStyle(

fontSize: 12, color: Colors.blue),

),

],

)

],

)),

),

Card(

child: SizedBox(

height: 50,

width: 350,

child: Row(

mainAxisAlignment:

MainAxisAlignment.spaceEvenly,

children: [

// Icon(Icons.rounded\_corner),

Column(

children: [

Text(

"Add Debit/Credit/ATM Card",

style: TextStyle(

fontSize: 20, color: Colors.blue),

),

Text(

"(You can save your MasterCard and VISA cards)",

style: TextStyle(

fontSize: 12, color: Colors.blue),

),

],

)

],

)),

),

Card(

child: SizedBox(

height: 50,

width: 350,

child: Row(

mainAxisAlignment:

MainAxisAlignment.spaceEvenly,

children: [

// Icon(Icons.rounded\_corner),

Column(

children: [

Text(

"EMI",

style: TextStyle(

fontSize: 20, color: Colors.blue),

),

Text(

"(Currently Unavailable)",

style: TextStyle(

fontSize: 12, color: Colors.blue),

),

],

)

],

)),

),

Card(

child: SizedBox(

height: 50,

width: 350,

child: Row(

mainAxisAlignment:

MainAxisAlignment.spaceEvenly,

children: [

// Icon(Icons.rounded\_corner),

Column(

children: [

Text(

"Pay on Delivery",

style: TextStyle(

fontSize: 20, color: Colors.blue),

),

Text(

"(Pay digitally with SMS Pay Link.)",

style: TextStyle(

fontSize: 12, color: Colors.blue),

),

],

)

],

)),

),

SizedBox(

height: 50,

),

Card(

child: SizedBox(

height: 30,

width: 350,

child: Row(

mainAxisAlignment:

MainAxisAlignment.spaceEvenly,

children: [

Icon(Icons.arrow\_drop\_down),

Column(

children: [

Text(

"Add Gift Card or Promo Coad",

style: TextStyle(

fontSize: 20,

color: Colors.black),

),

],

)

],

)),

),

SizedBox(

height: 80,

),

SizedBox(

width: 200,

height: 50,

child: ElevatedButton(

style: ElevatedButton.styleFrom(

primary: Colors.red[300],

onPrimary: Colors.white,

),

onPressed: () {

ScaffoldMessenger.of(context)

.showSnackBar(SnackBar(

content: Text("Under Construction",

style: TextStyle(

color: Colors.white,

fontSize: 15,

fontWeight: FontWeight.bold)),

duration: Duration(seconds: 5),

backgroundColor: Colors.red[300],

));

},

child: Text(

'Continue',

style: TextStyle(fontSize: 20),

),

),

)

],

)),

onTap: () {

ScaffoldMessenger.of(context).showSnackBar(SnackBar(

content: Text("Under Construction",

style: TextStyle(

color: Colors.white,

fontSize: 15,

fontWeight: FontWeight.bold)),

duration: Duration(seconds: 5),

backgroundColor: Colors.black,

));

},

)

],

),

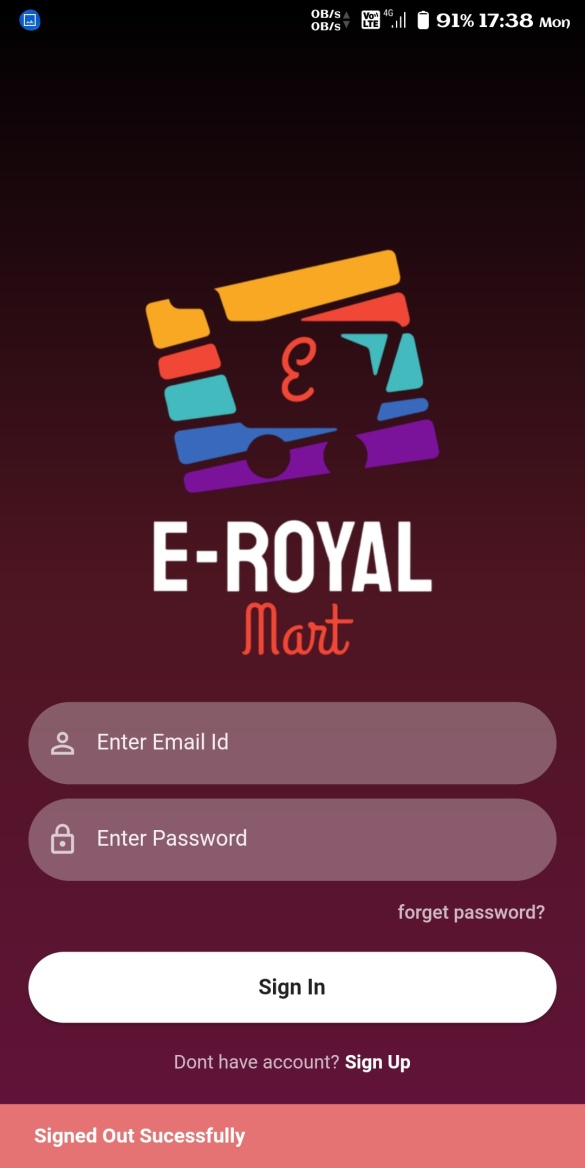
),

));

}}

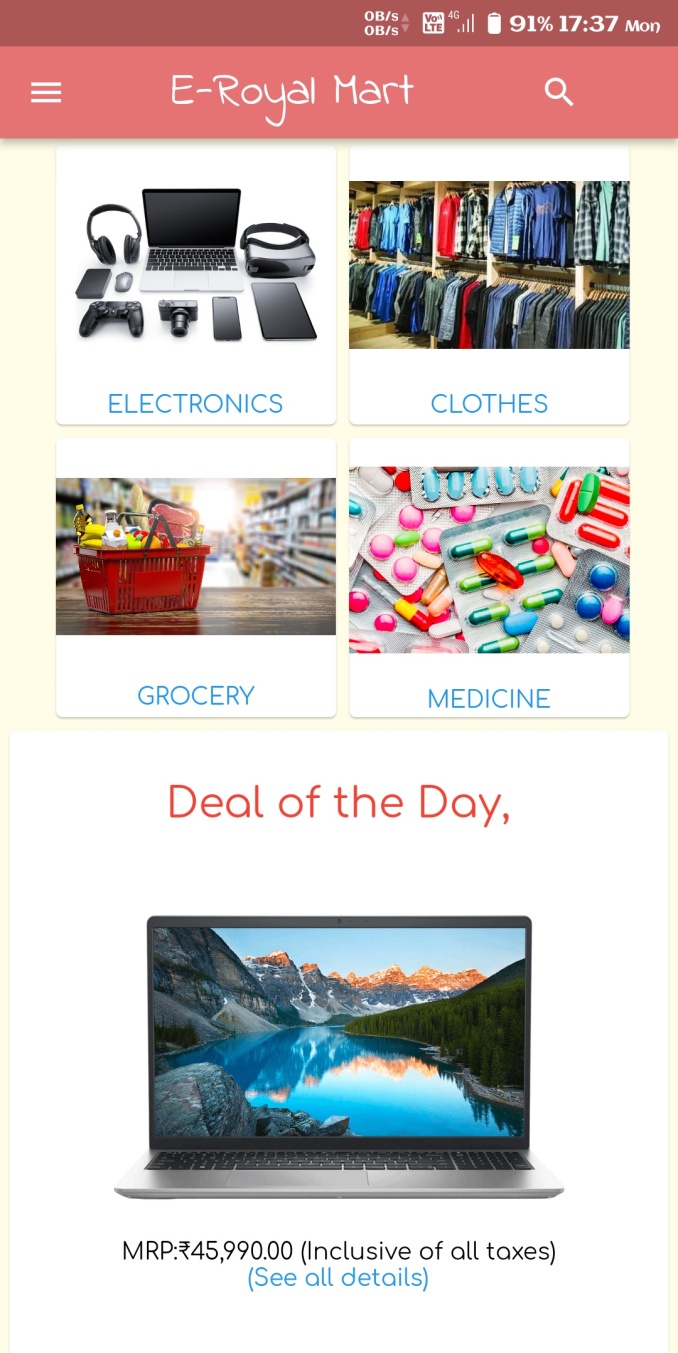
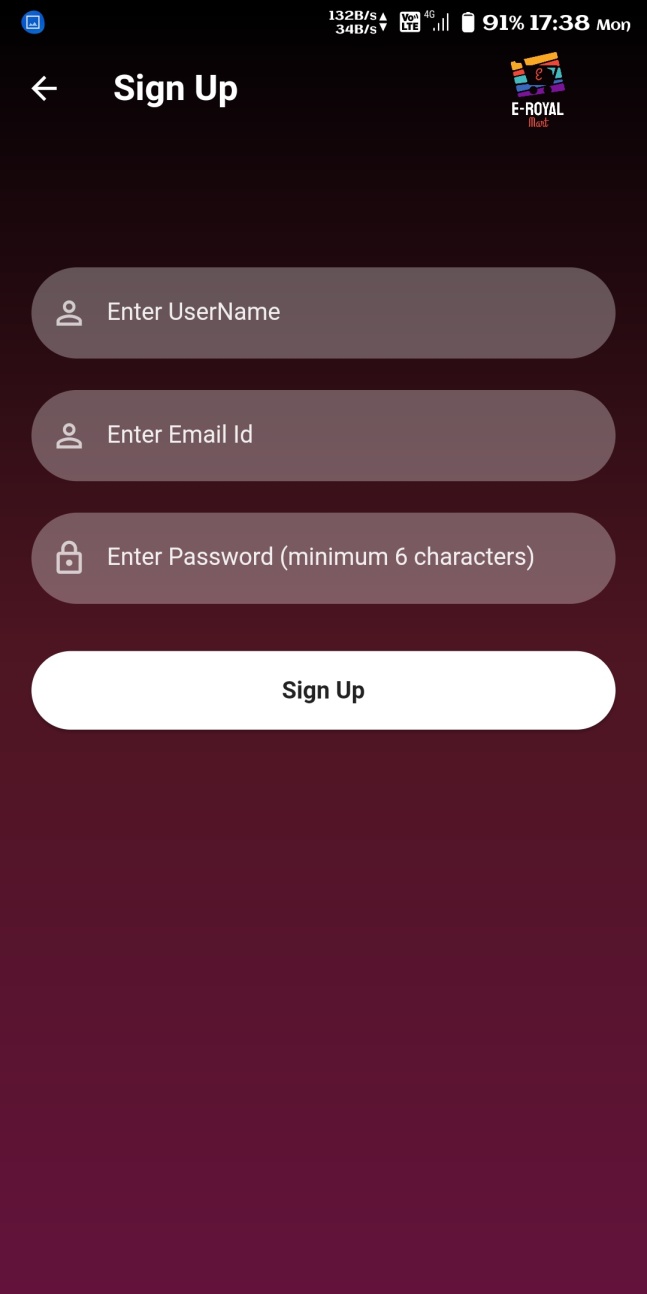
**User Interface Designs**

**WELCOME ACTIVITY: LOGIN ACTIVITY:**

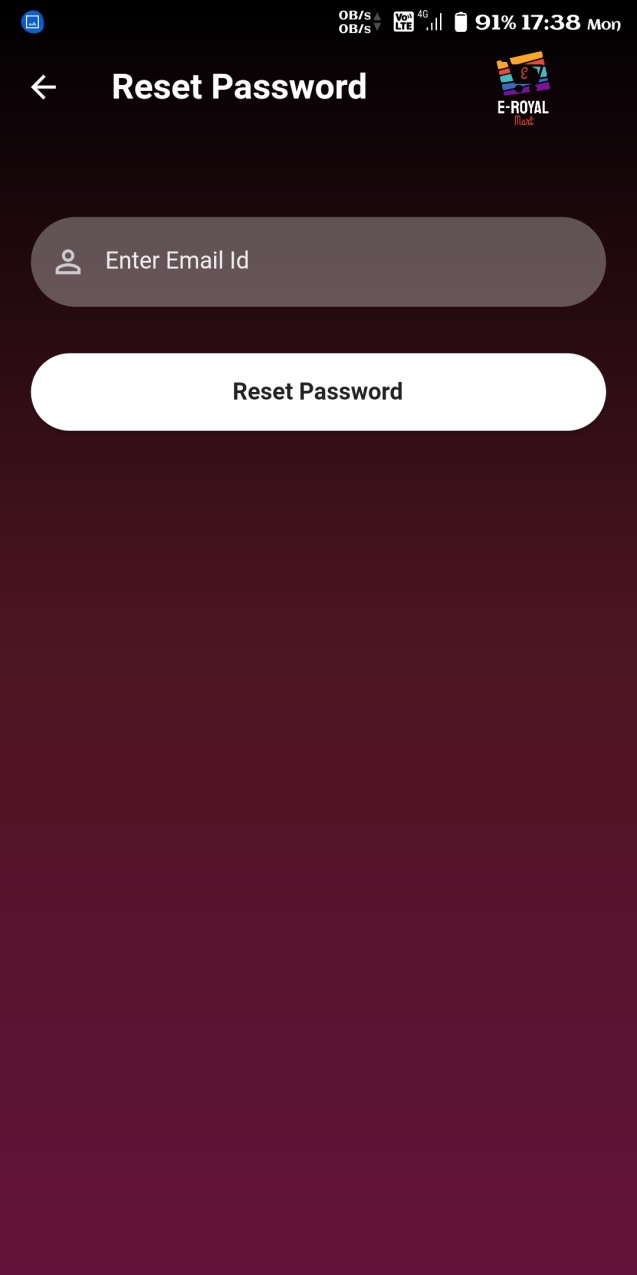
****

**Homepage:**

**Sign up Activity:**

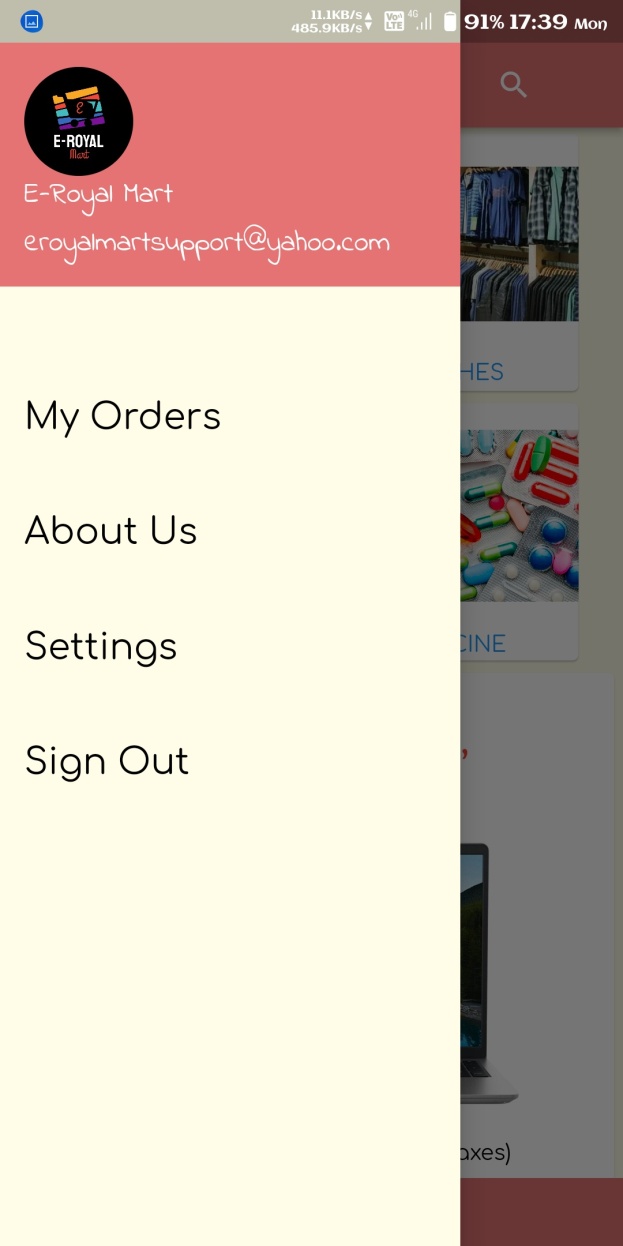
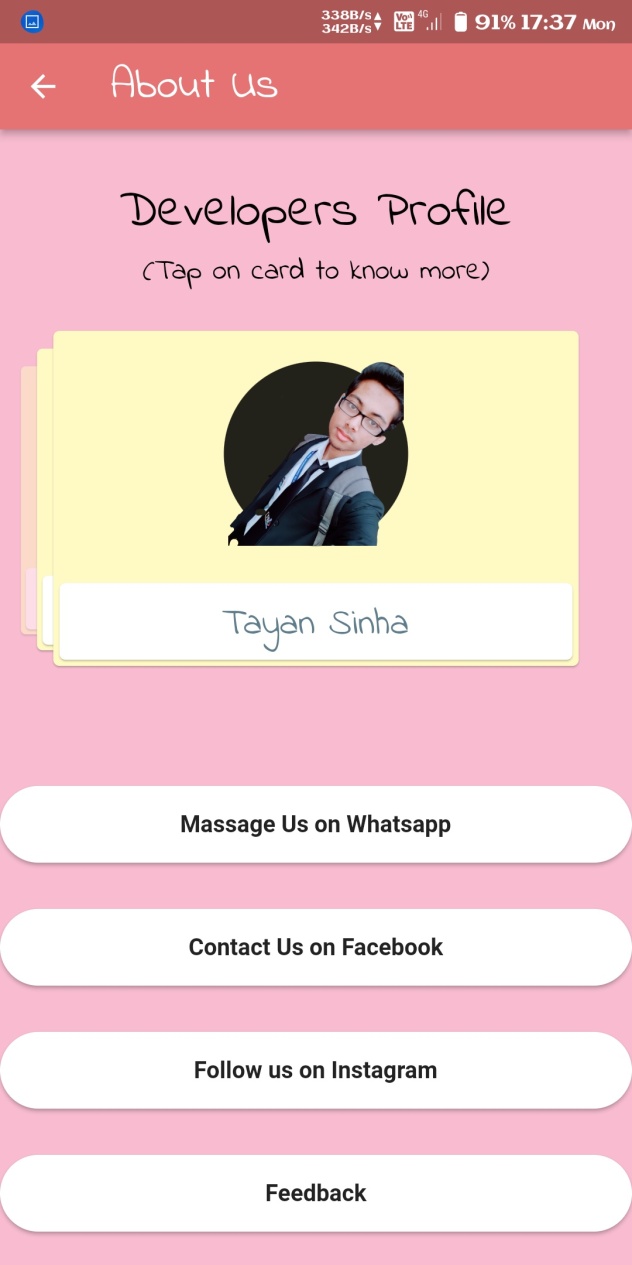
****

**Reset password:**

** Products :**

****

**About us: Drawer:**

****

**Testing:**

**Objective:**

The objective of our test plan is to find and report as many bugs as  possible to improve the integrity of our program. Although exhaustive  testing is not possible, we will exercise a broad range of tests to achieve  our goal. We will also test the user friendliness of our application. The application will be used as an important tool, but we would like to ensure that it could be run on a variety of platforms with little impact on  performance or Usability.

**Testing Process:**

1.Organize Project :It involves creating a System Test Plan, Schedule & Test Approach, and assigning responsibilities.

2.Design/Build System Test: It involves identifying Test Cycles, Test  Cases, Entrance & Exit Criteria, Expected Results, etc. In general, test conditions/expected results will be identified by the Test Team in  conjunction with the Development Team. The Test Team will then identify Test Cases and the Data required. The Test conditions are  derived from the Program Specifications Document.

3.Design/Build Test Procedures: It includes setting up procedures such  as Error Management systems and Status reporting.

4.Build Test Environment:It includes requesting/building hardware,  software and data setups.

5.Execute System Tests :It is identified in the Design/Build Test  Procedures will be executed. All results will be documented and Bug  Report Forms filled out and given to the Development

team as necessary.

6.Signoff : It happens when all pre-defined exit criteria have been  achieved.

**Testing Strategy:**

The following outlines the types of testing that will be

done for unit, integration, and system testing. While it

includes what will be tested, the specific use cases that

determine how the testing is done will be detailed in

the Test Design Document. The test cases that will be

used for designing use cases is shown below.

**TEST CASES:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SL.**  **NO.** | **Requirement**  **Name** | **User** | **Description** | **Successful  end**  **connection** |
| 1 | Sign up  And Order require product | Customer | For sign up, the user has  to provide Email and password and user name. The user  can search their require product . | Sign up mail sent to your email address |
| 2 | Login  And order their product | Customer | Login by giving email  and password. The user can select which product they need and can order their product . | After order placed order confirmed mail sent to your mail |

Security Measures

Data security refers to the process of protecting data from unauthorized  access and data corruption throughout its lifecycle. Data security includes data encryption, hashing, tokenization, and key management practices that  protect data across all applications and platforms.

Organizations around the globe are investing heavily in information  technology (IT) cyber security capabilities to protect their critical assets.  Whether an enterprise needs to protect a brand, intellectual capital, and  customer information or provide controls for critical infrastructure, the  means for incident detection and response to protecting organizational  interests have three common elements: people, processes, and technology

Database Security Measures

● Ensure physical **database security**.

● Use web application and **database** firewalls.

● Harden your **database** to the fullest extent possible.

● Encrypt your **data**.

● Minimize value of **databases**.

● Manage **database** access tightly.

● Audit and monitor **database** activity.

**Software Security Measures**

***1) Establish strong passwords***

This first measure is really easy to put in place. You must put together a combination of capitals,  lower-case letters, numbers, and symbols to create a strong password. The more characters  you put, the better.

***2) Think of antivirus protection***

Antivirus are indispensable to protecting your Data. They are designed to prevent, search for,  detect and remove viruses but also adware, worms, trojans, and so on.

***3) Updating is important***

Your app must be properly patched and updated. Recent updates allow your Data to be more  secured.

***4) Secure mobile phones***

Mobile phones are even more easily stolen than laptops but they are as valuable for companies.  Equally to laptops, phones can be encrypted- you can put a strong password and enable an  automatic lock-out. You can also set up a wiping process if the phone is lost or stolen.

***5) Monitor steadily***

Data, Software, technologies, everything is moving so fast. Keep track of them, keep in touch  with news to see what is new on the market.

***6) Be smart with emails***

Downloading apps or files, opening emails and clicking on links can infect your computer and  your network. Be careful with the sources you find online or you receive.

**Limitations:**

1. This app requires stable internet connection.

2. Without creating an account a person cannot utilize its facilities.

**Future Scopes:**

1. Checkout portal will be updated with various options of payment.
2. Searching option will be added later.
3. More product categories will be added later.

**CONCLUSION:**

Technology has made significant progress over the years to provide consumers a better online shopping experience and will continue to do so for years to come.  With the rapid growth of products and brands, people have speculated that online shopping will overtake in-store shopping.  While this has been the case in some areas, there is still demand for brick and mortar stores in market areas where the consumer feels more comfortable seeing and touching the product being bought.  However, the availability of online shopping has produced a more educated consumer that can shop around with relative ease without having to spend a large amount of time.  In exchange, online shopping has opened up doors to many small retailers that would never be in business if they had to incur the high cost of owning a brick and mortar store.  At the end, it has been a win-win situation for both consumer and sellers.

**BIBLIOGRAPHY**

1. www.stackoverflow.com

2. https://pub.dev/

3. https://docs.flutter.dev/

4. https://firebase.flutter.dev/

Source Code: [TayanSinha/E-commerce\_App: Developed using flutter... (github.com)](https://github.com/TayanSinha/E-commerce_App)