MAD and PWA Lab

Name: Sneha Utekar
Class: D15A
Roll no:66

Experiment No.-6

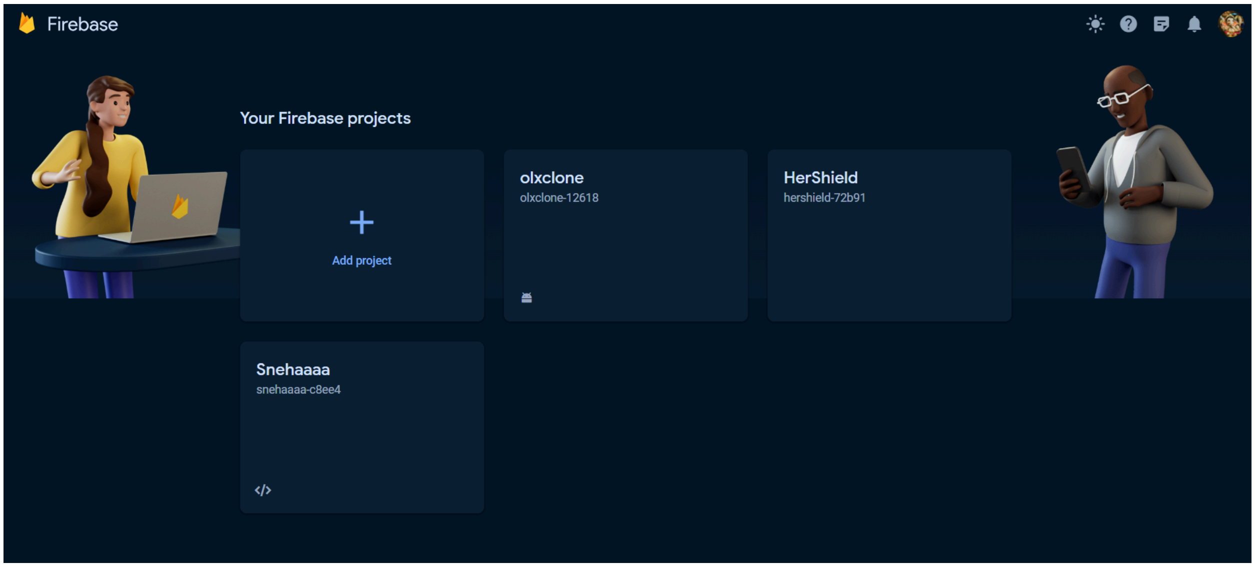
AIM: To connect flutter UI with firebase database.

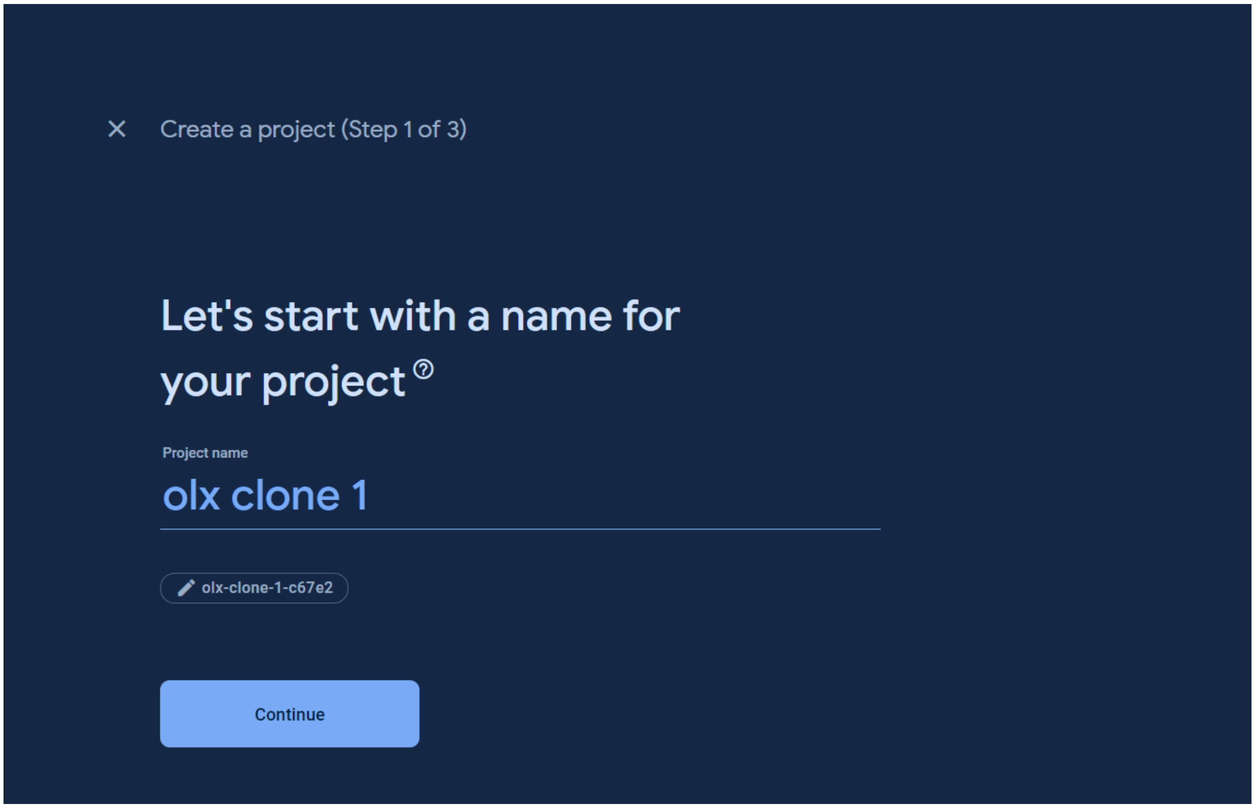
THEORY:

Firebase is a great backend solution for anyone that wants to use authentication, databases, cloud functions, ads, and countless other features within an app.

OUTPUT:

1.Create a new firebase project:





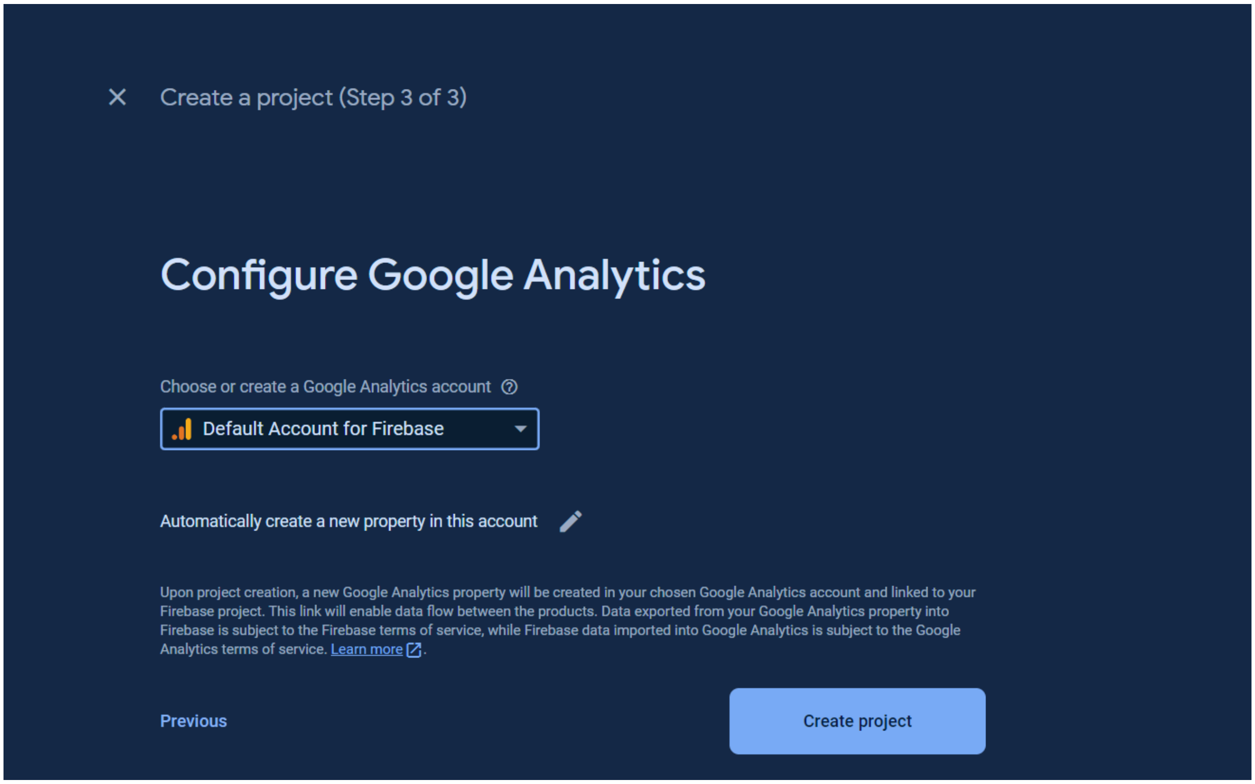
Create a project (Step 1 of 3)

Let's start with a name for your project

Project name

olx clone 1

olx-clone-1-c67e2



Create a project (Step 3 of 3)

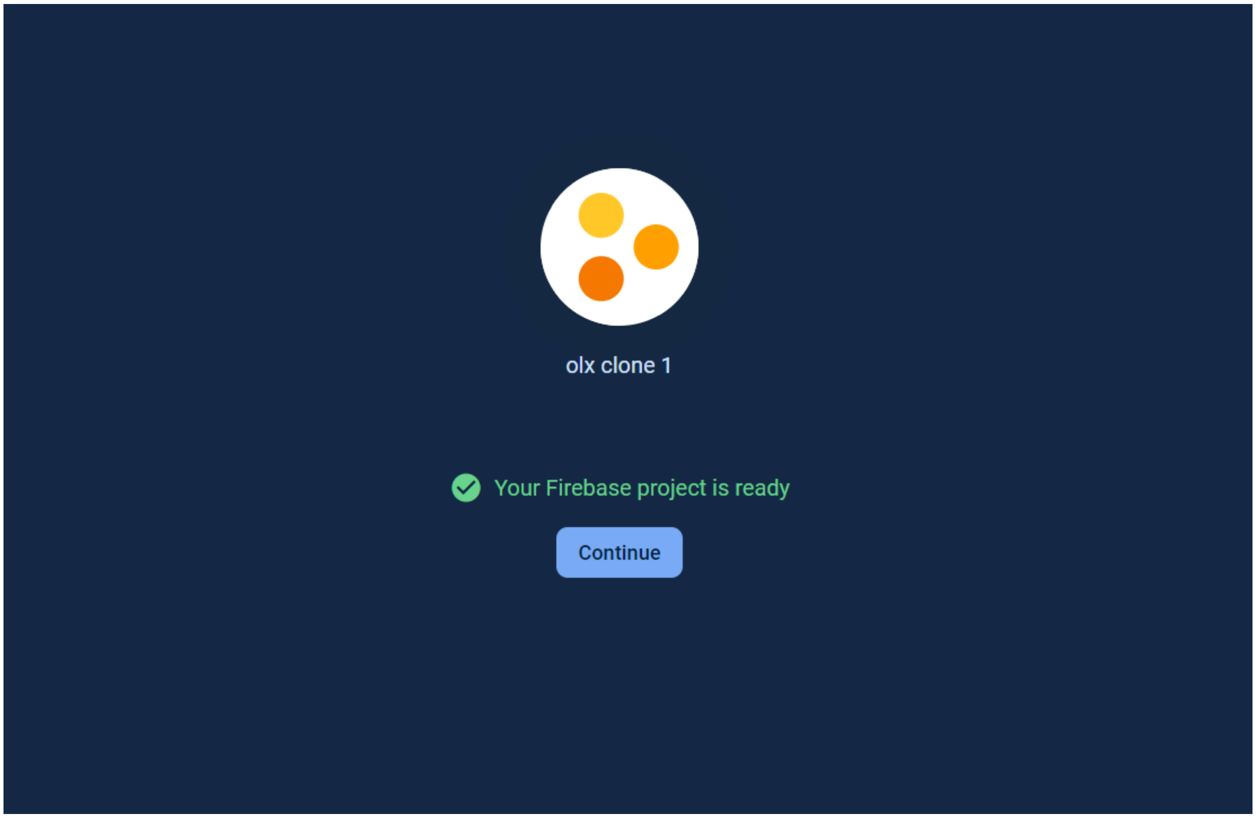
Configure Google Analytics

Choose or create a Google Analytics account

Default Account for Firebase

Automatically create a new property in this account

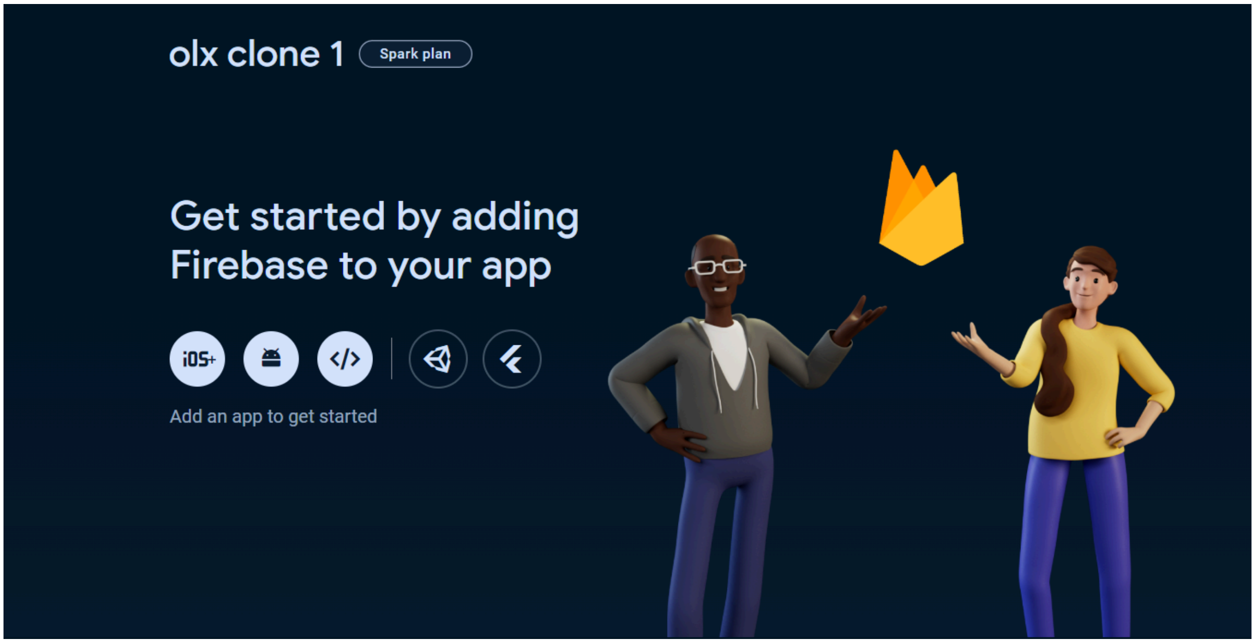
Upon project creation, a new Google Analytics property will be created in your chosen Google Analytics account and linked to your Firebase project. This link will enable data flow between the products. Data exported from your Google Analytics property into Firebase is subject to the Firebase terms of service, while Firebase data imported into Google Analytics is subject to the Google Analytics terms of service. Learn more.



olx clone 1

Your Firebase project is ready

2.Select the Android option:

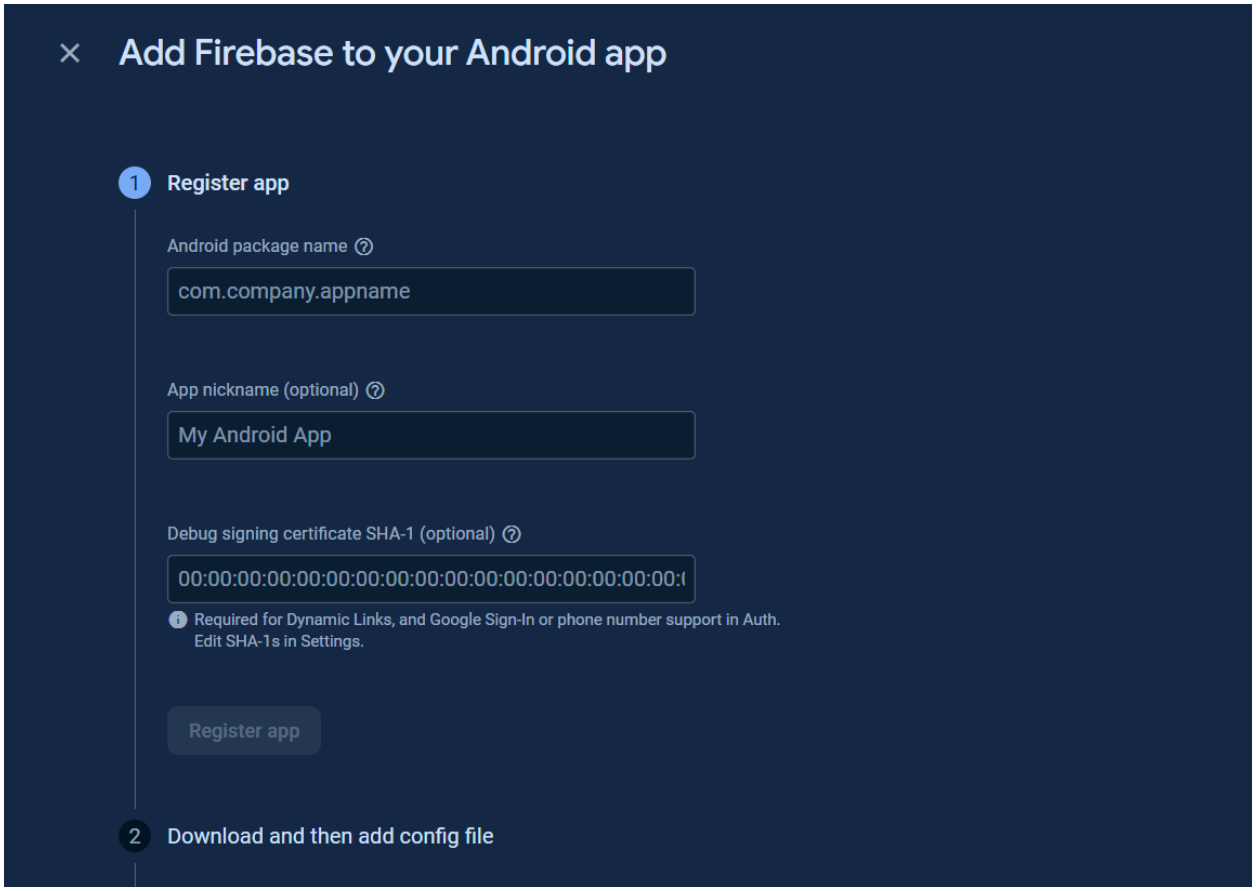


olx clone 1 Spark plan

Get started by adding Firebase to your app

Add an app to get started

3.Add firebase to your app:



× Add Firebase to your Android app

1 Register app

Android package name ⓘ

com.company.appname

App nickname (optional) ⓘ

My Android App

Debug signing certificate SHA-1 (optional) ⓘ

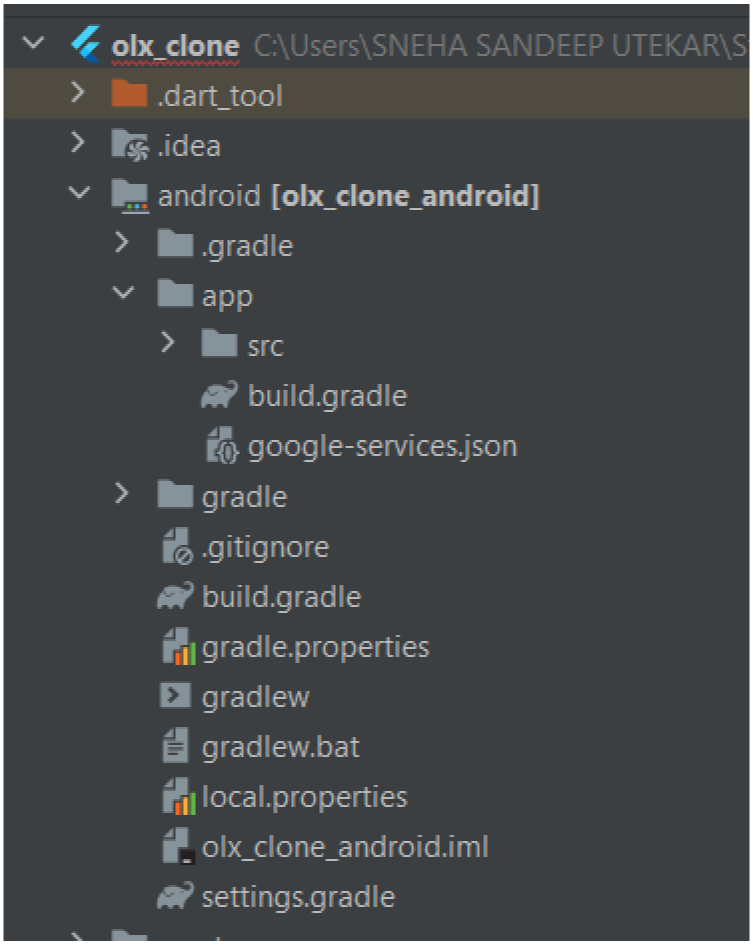
00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:(

ⓘ Required for Dynamic Links, and Google Sign-In or phone number support in Auth. Edit SHA-1s in Settings.

2 Download and then add config file

4.After completing all the steps ,you will get a google-services.json file.Download that file and add it to your Android project:

app→src→google-services.json:



```
olx_clone C:\Users\SNEHA SANDEEP UTEKAR\S
  .dart_tool
  .idea
  android [olx_clone_android]
    .gradle
    app
      src
      build.gradle
      google-services.json
    gradle
    .gitignore
    build.gradle
    gradle.properties
    gradlew
    gradlew.bat
    local.properties
    olx_clone_android.iml
    settings.gradle
```

google-services.json:

```
{
 "project_info": {
  "project_number": "232516563209",
  "project_id": "olxclone-12618",
  "storage_bucket": "olxclone-12618.appspot.com"
 },
 "client": [
  {
   "client_info": {
    "mobilesdk_app_id": "1:232516563209:android:d2dea587c60de17bac63d8",
    "android_client_info": {
     "package_name": "com.example.olx_clone"
    }
   },
   "oauth_client": [],
   "api_key": [
    {
     "current_key": "AIzaSyDuFtAm59795zlrpp16kTJTbd1Lac95KkI"
    }
   ],
   "services": {
    "appinvite_service": {
     "other_platform_oauth_client": []
    }
   }
  }
 ],
 "configuration_version": "1"
}
```

Add dependencies related to firebase in pubsec.yaml:

```
name: olx_clone
description: "A new Flutter project."

publish_to: 'none' # Remove this line if you wish to publish to pub.dev


version: 1.0.0+1

environment:
 sdk: '>=3.2.5 <4.0.0'


dependencies:
 flutter:
  sdk: flutter




 cupertino_icons: ^1.0.2
 cloud_firestore: ^4.15.3
 animated_text_kit: ^4.2.2
 flutter_signin_button: ^2.0.0
 firebase_core: ^2.24.2
 firebase_auth: ^4.16.0
 legacy_progress_dialog: ^0.1.2
 location: ^5.0.3
 flutter_geocoder: ^0.2.2-nullsafety
 google_sign_in: ^6.2.1
 email_validator: ^2.1.17
 flutter_neumorphic_plus: ^3.3.0
 open_mail_app: ^0.4.5
 csc_picker: ^0.2.7
 provider: ^6.1.1
 image_picker: ^0.7.5+3
 firebase_storage: ^11.6.5
 firebase_analytics: ^10.3.0


dev_dependencies:
 flutter_test:
   sdk: flutter



 flutter_lints: ^2.0.0

flutter:

uses-material-design: true

assets:
   - assets/images/cart.png
   - assets/images/map.png
```

CODE:

emailAuth_service:

```
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:olx_clone/screens/authentication/email_verification_screen.dart';

import '../screens/location_screen.dart';

class EmailAuthentication {
 CollectionReference users = FirebaseFirestore.instance.collection('users');

 Future<DocumentSnapshot> getAdminCredential({
   required String email,
   required String password,
   required bool isLog,
   required BuildContext context,
 }) async {
   DocumentSnapshot _result = await users.doc(email).get();
   if (isLog) {
     emailLogin(email, password, context);
   } else {
     if (_result.exists) {
       ScaffoldMessenger.of(context).showSnackBar(
         const SnackBar(
           content: Text('An Account Already Exists with this Email'),
         ),
       );
     } else {
       emailRegister(email, password, context);
     }
   }

   return _result;
 }

 emailLogin(String email, String password, BuildContext context) async {
   try {
     UserCredential userCredential = await FirebaseAuth.instance
         .signInWithEmailAndPassword(
         email: email,
         password: password);
     if(userCredential.user?.uid!=null){
       Navigator.pushReplacementNamed(context,LocationScreen.id);
     }
   } on FirebaseAuthException catch (e) {
     if (e.code == 'user-not-found') {
       ScaffoldMessenger.of(context).showSnackBar(
         const SnackBar(content: Text('No user found for that email')),
       );
     } else if (e.code == 'wrong-password') {
       ScaffoldMessenger.of(context).showSnackBar(
         const SnackBar(
             content: Text('Wrong Password provided for that user')),
       );
     }
   }
 }

 emailRegister(String email, String password, BuildContext context) async {
   try {
     UserCredential userCredential = await FirebaseAuth.instance
         .createUserWithEmailAndPassword(email: email, password: password);
     if (userCredential.user?.uid != null) {
       //login success .Add user data to firestore
       return users.doc(userCredential.user!.uid).set({
         'uid': userCredential.user!.uid,
         'mobile': null,
         'email': userCredential.user!.email,
         'name':null,
       }).then((value)async {
           //before going to location screen will send email verification
           await userCredential.user?.sendEmailVerification().then((value){
             //after sending verification email,screen will move to emailverification screen
             Navigator.pushReplacementNamed(context, EmailVerificationScreen.id);
           });


       }).catchError((onError) {
         ScaffoldMessenger.of(context).showSnackBar(
           const SnackBar(content: Text('Failed to add user')),
         );
       });
     }
   } on FirebaseAuthException catch (e) {
     if (e.code == 'weak-password') {
       ScaffoldMessenger.of(context).showSnackBar(
         const SnackBar(content: Text('The password provided is too weak.')),
       );
     } else if (e.code == 'email-already-in-use') {
       ScaffoldMessenger.of(context).showSnackBar(
         const SnackBar(
             content: Text('The account already exists for that email.')),
       );
     }
   } catch (e) {
     print(e);
     ScaffoldMessenger.of(context).showSnackBar(
       const SnackBar(content: Text('Error occurred')),
     );
   }
 }
}
```

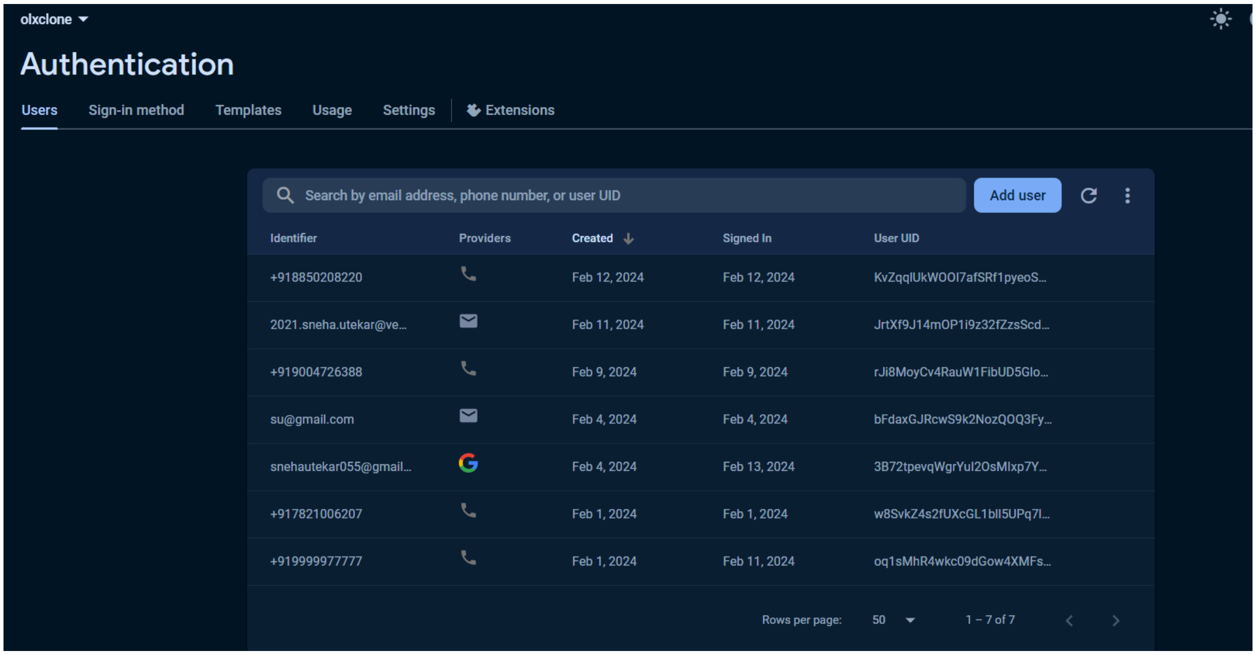
phoneAuth_service:

```
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:olx_clone/screens/location_screen.dart';
import 'package:olx_clone/screens/authentication/otp_screen.dart';

class PhoneAuthService {
 FirebaseAuth auth = FirebaseAuth.instance;
 User? user=FirebaseAuth.instance.currentUser;

 CollectionReference users = FirebaseFirestore.instance.collection('users');

 Future<void> addUser(context,uid)async {
   // Call the user's CollectionReference to add a new user
```

```
  final QuerySnapshot result=await users.where('uid',isEqualTo: uid).get();
  List <DocumentSnapshot> document=result.docs;

  if(document.length>0){
   Navigator.pushReplacementNamed(context,LocationScreen.id);
  }else{
    return users.doc(user?.uid)
        .set({
      'uid': user?.uid, // John Doe
      'mobile': user?.phoneNumber, // Stokes and Sons
      'email': user?.email, // 42
      'name':null
    })
        .then((value) {
      Navigator.pushReplacementNamed(context,LocationScreen.id);
    })
        .catchError((error) => print("Failed to add user: $error"));

  }
  }



Future<void> verifyPhoneNumber(BuildContext context, number) async {
  final PhoneVerificationCompleted verificationCompleted =
      (PhoneAuthCredential credential) async {
    await auth.signInWithCredential(credential);
  };

  final PhoneVerificationFailed verificationFailed =
      (FirebaseAuthException e) {
    if (e.code == 'invalid-phone-number') {
      print('The provided phone number is not valid.');
    }
    print("The error is ${e.code}");
  };

  final PhoneCodeSent codeSent = (String verId, int ?resendToken) async {
    Navigator.push(
      context, MaterialPageRoute(builder: (context) => OTPScreen(number: number,verId:
verId,),),);
  };


  try {
    auth.verifyPhoneNumber(phoneNumber: number,
        verificationCompleted: verificationCompleted,
        verificationFailed: verificationFailed,
        codeSent: codeSent,
        timeout: const Duration(seconds: 60),
        codeAutoRetrievalTimeout: (String verificationId) {
          print(verificationId);
        });
  } catch (e) {
    print('Error ${e.toString()}');
  }
 }
}
```

Firebase Results:

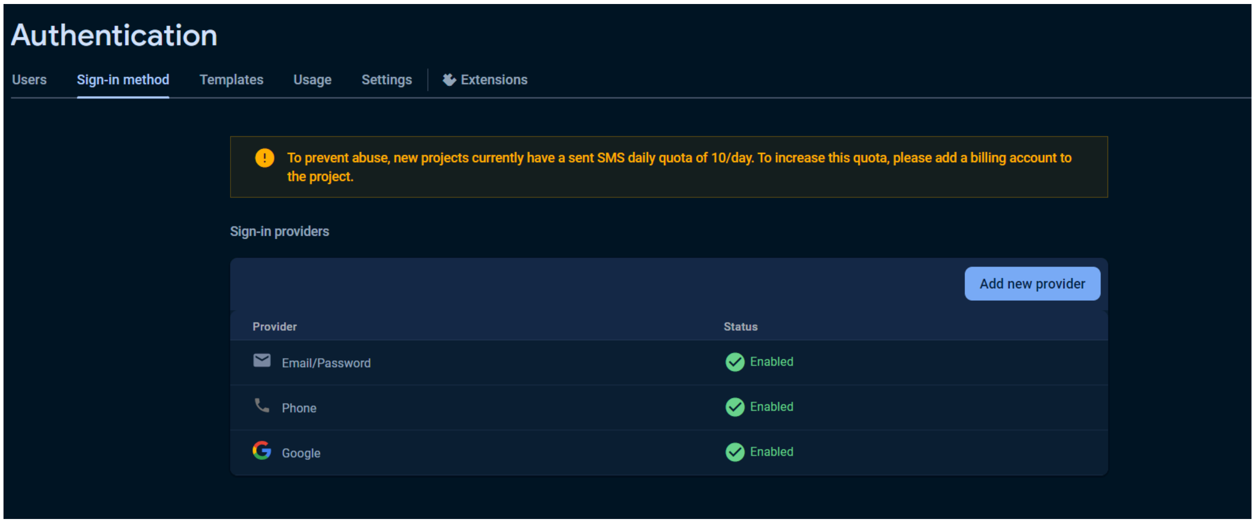
Authentication:



Identifier	Providers	Created	Signed In	User UID
+918850208220	☎	Feb 12, 2024	Feb 12, 2024	KvZqqlUkWOOI7afSRf1pyeoS...
2021.sneha.utekar@ve...	✉	Feb 11, 2024	Feb 11, 2024	JrtXf9J14mOP1i9z32fZzsScd...
+919004726388	☎	Feb 9, 2024	Feb 9, 2024	rJi8MoyCv4RauW1FibUD5Glo...
su@gmail.com	✉	Feb 4, 2024	Feb 4, 2024	bFdaxGJRcwS9k2NozQOQ3Fy...
snehautekar055@gmail...	G	Feb 4, 2024	Feb 13, 2024	3B72tpevqWgrYul2OsMlxp7Y...
+917821006207	☎	Feb 1, 2024	Feb 1, 2024	w8SvkZ4s2fUXcGL1bll5UPq7l...
+919999977777	☎	Feb 1, 2024	Feb 11, 2024	oq1sMhR4wkc09dGow4XMFs...

Rows per page: 50 1 – 7 of 7

Sign-in methods:



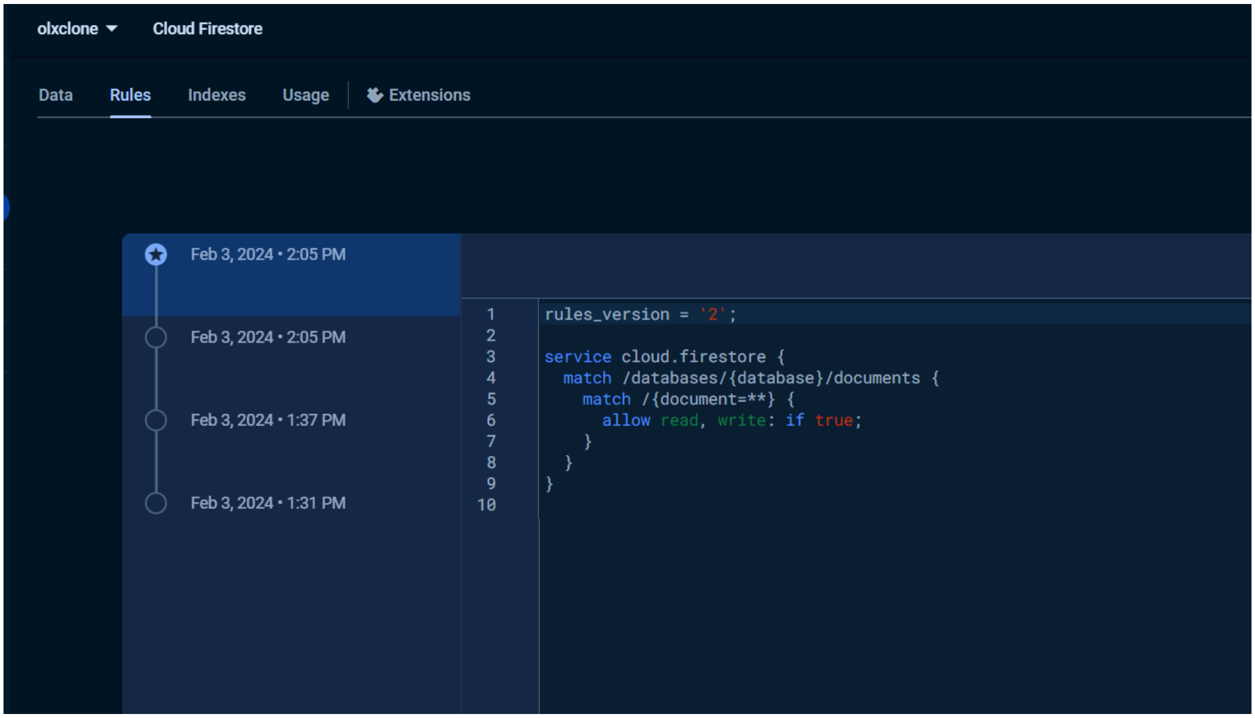
To prevent abuse, new projects currently have a sent SMS daily quota of 10/day. To increase this quota, please add a billing account to the project.

Sign-in providers

Provider	Status
Email/Password	✓ Enabled
Phone	✓ Enabled
Google	✓ Enabled

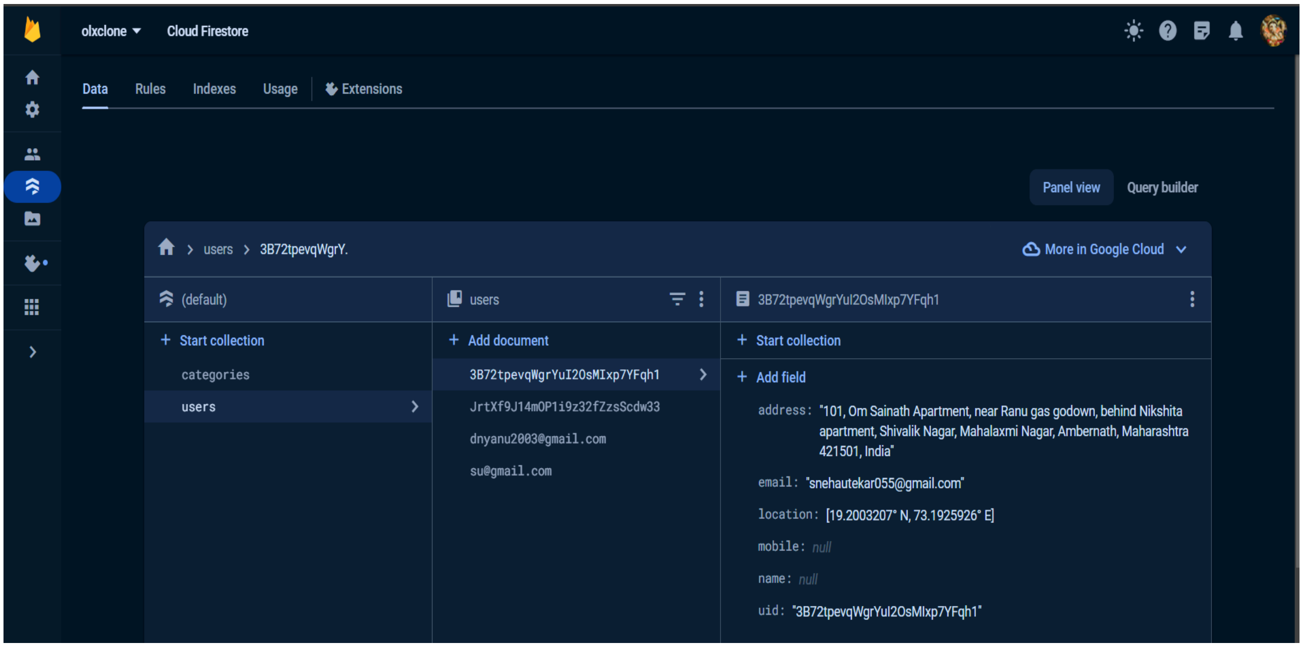
Firebase Database:

Before creating the database add the following rules:



```
rules_version = '2';

service cloud.firestore {
  match /databases/{database}/documents {
    match /{document=**} {
      allow read, write: if true;
    }
  }
}
```

These are the users:



categories
users

3B72tpevqWgrYuI2OsMIxp7YFqh1
JrtXf9J14mOP1i9z32fZzsScdw33
dnyanu2003@gmail.com
su@gmail.com

address: "101, Om Sainath Apartment, near Ranu gas godown, behind Nikshita apartment, Shivalik Nagar, Mahalaxmi Nagar, Ambernath, Maharashtra 421501, India"

email: "snehautekar055@gmail.com"

location: [19.2003207° N, 73.1925926° E]

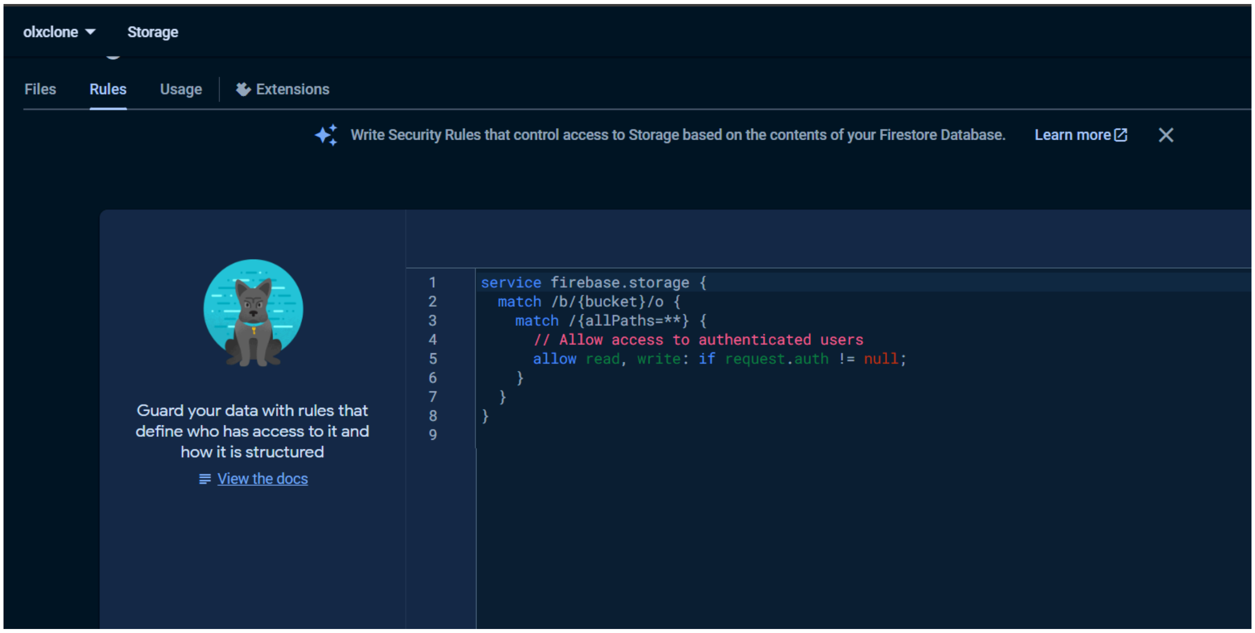
mobile: null

name: null

uid: "3B72tpevqWgrYuI2OsMIxp7YFqh1"

Firebase Storage:

Before using the firebase Storage ,add the following rules:



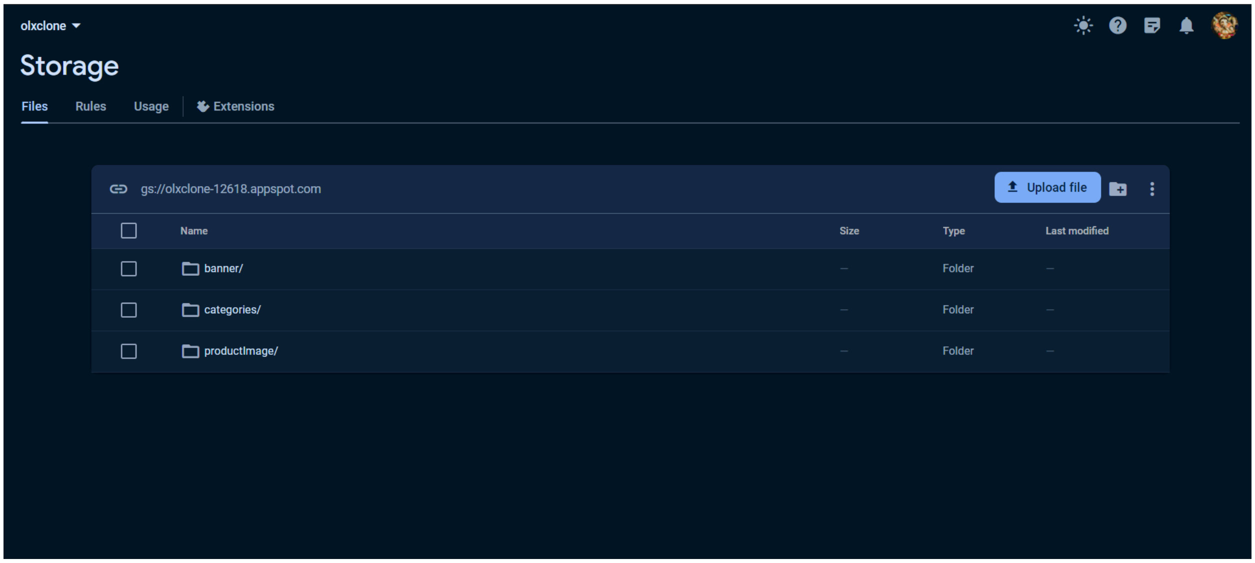
Write Security Rules that control access to Storage based on the contents of your Firestore Database.

Guard your data with rules that define who has access to it and how it is structured

View the docs

```
service firebase.storage {
  match /b/{bucket}/o {
    match /{allPaths=**} {
      // Allow access to authenticated users
      allow read, write: if request.auth != null;
    }
  }
}
```

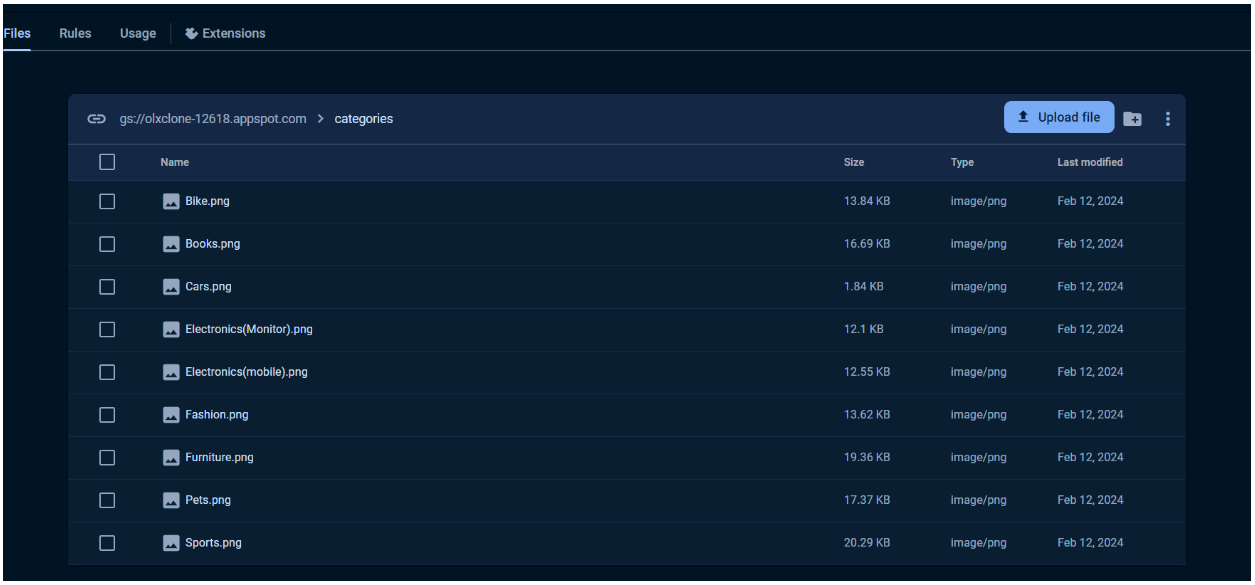
I have used firebase storage to store the images ,icons and categories



gs://olxclone-12618.appspot.com

□	Name	Size	Type	Last modified
□	banner/	–	Folder	–
□	categories/	–	Folder	–
□	productImage/	–	Folder	–

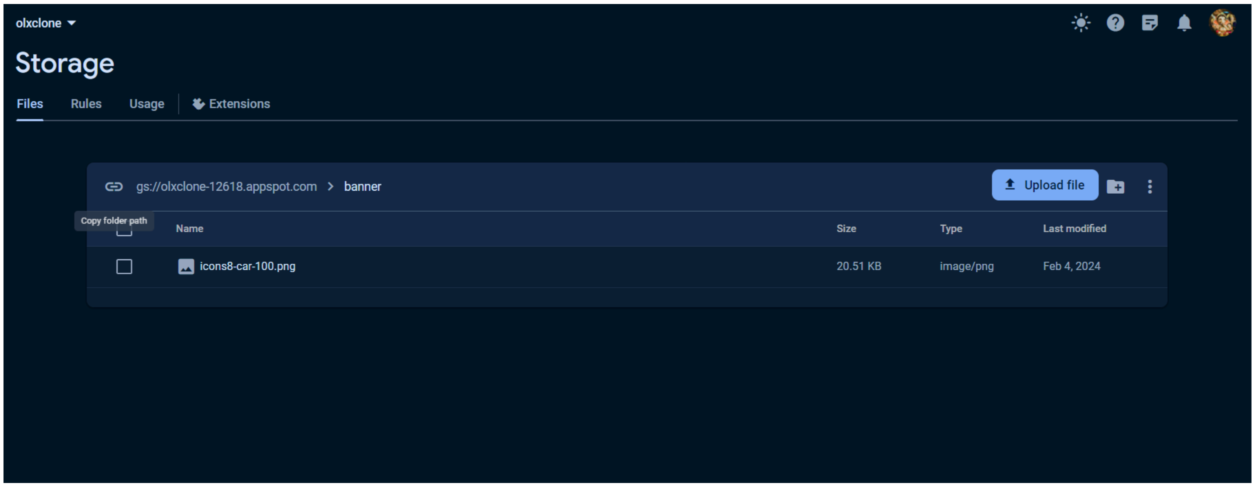
Icons for categories:



gs://olxclone-12618.appspot.com > categories

□	Name	Size	Type	Last modified
□	Bike.png	13.84 KB	image/png	Feb 12, 2024
□	Books.png	16.69 KB	image/png	Feb 12, 2024
□	Cars.png	1.84 KB	image/png	Feb 12, 2024
□	Electronics(Monitor).png	12.1 KB	image/png	Feb 12, 2024
□	Electronics(mobile).png	12.55 KB	image/png	Feb 12, 2024
□	Fashion.png	13.62 KB	image/png	Feb 12, 2024
□	Furniture.png	19.36 KB	image/png	Feb 12, 2024
□	Pets.png	17.37 KB	image/png	Feb 12, 2024
□	Sports.png	20.29 KB	image/png	Feb 12, 2024

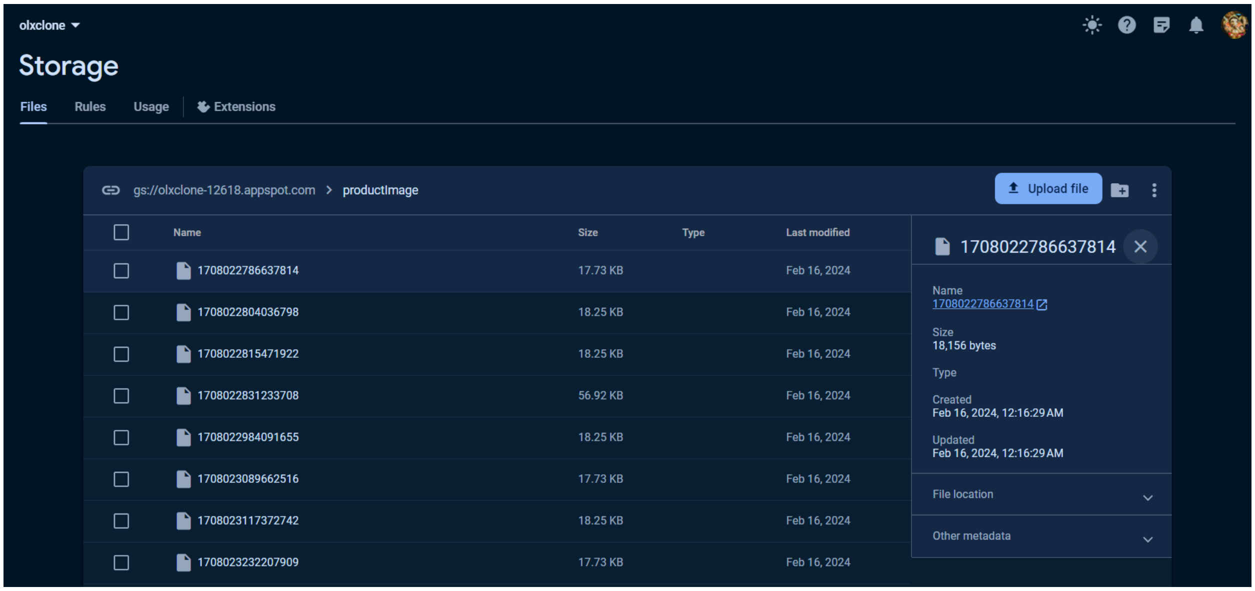
Icons for banner :



gs://olxclone-12618.appspot.com > banner

Name	Size	Type	Last modified
icons8-car-100.png	20.51 KB	image/png	Feb 4, 2024

Images:



gs://olxclone-12618.appspot.com > productImage

Name	Size	Type	Last modified
1708022786637814	17.73 KB		Feb 16, 2024
1708022804036798	18.25 KB		Feb 16, 2024
1708022815471922	18.25 KB		Feb 16, 2024
1708022831233708	56.92 KB		Feb 16, 2024
1708022984091655	18.25 KB		Feb 16, 2024
1708023089662516	17.73 KB		Feb 16, 2024
1708023117372742	18.25 KB		Feb 16, 2024
1708023232207909	17.73 KB		Feb 16, 2024

1708022786637814

Name
1708022786637814

Size
18,156 bytes

Type

Created
Feb 16, 2024, 12:16:29 AM

Updated
Feb 16, 2024, 12:16:29 AM

firebase_service.dart:

```
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:flutter_geocoder/geocoder.dart';
import 'package:olx_clone/screens/home_screen.dart';

class FirebaseService{

 CollectionReference users=FirebaseFirestore.instance.collection('users');
 CollectionReference categories=FirebaseFirestore.instance.collection('categories');
 User? user=FirebaseAuth.instance.currentUser;

 Future<void> updateUser(Map<String,dynamic>data,context) {
 return users
     .doc(user?.uid)
     .update(data)
     .then((value) {
   Navigator.pushNamed(context,HomeScreen.id);
 },)
     .catchError((error) {
       ScaffoldMessenger.of(context).showSnackBar(
         const SnackBar(
             content:Text('Failed to update location'),
         ),
       );
 });

}
Future<String>getAddress(lat,long)async{
 final coordinates = new Coordinates(lat,long);
 var addresses = await Geocoder.local.findAddressesFromCoordinates(coordinates);
 var first = addresses.first;

 return first.addressLine ?? "Address not found";


}

Future<DocumentSnapshot>getUserData()async{
   DocumentSnapshot doc= await users.doc(user?.uid).get();
   return doc;
}
}
```

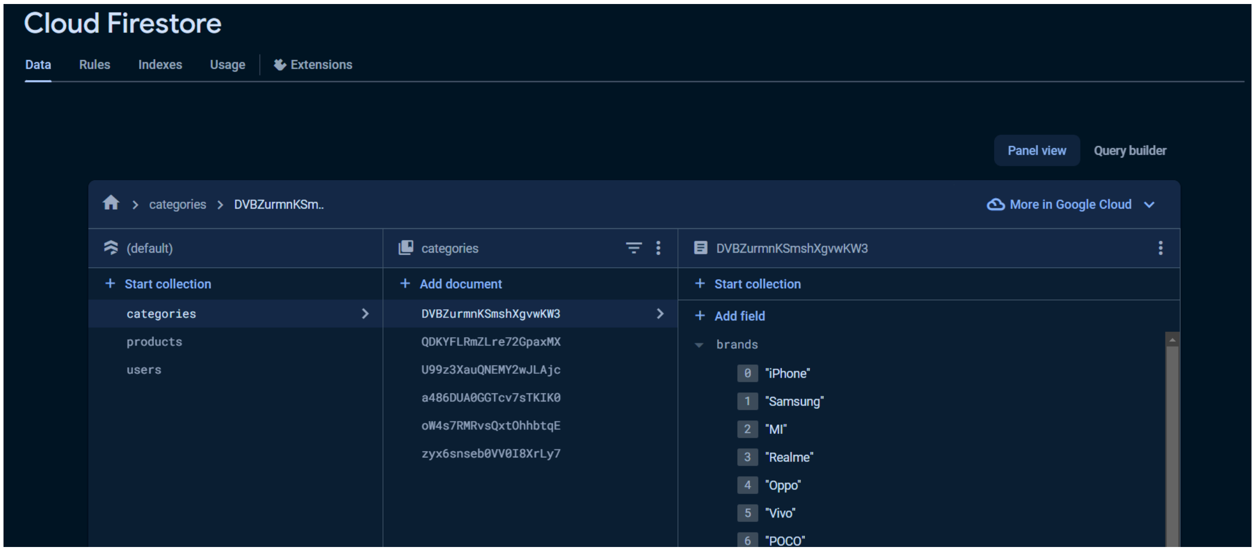
cat_provider.dart:

```
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:flutter/material.dart';
import 'package:olx_clone/services/firebase_service.dart';
```

```
class CategoryProvider with ChangeNotifier{
 FirebaseService _service=FirebaseService();

 late DocumentSnapshot doc;
 late String selectedCategory;
 late String selectedSubCategory= '';
 DocumentSnapshot? userDetails;
 List<String> urlList = [];
Map<String,dynamic> dataToFirestore={}; //this is the data we are going to upload to
firestore

 getCategory(selectedCat){
  this.selectedCategory=selectedCat;
  notifyListeners();
 }

 getSubCategory(selectedsubCat){
  this.selectedSubCategory=selectedsubCat;
  notifyListeners();
 }
 getCatSnapshot(snapshot){
  this.doc=snapshot;
  notifyListeners();
 }

 getImages(List<String> urls) {
  urlList.addAll(urls); // Append the new URLs to the existing list
  notifyListeners();
 }

 getData(data) {
  this.dataToFirestore=data;
  notifyListeners();
 }

  getUserDetails(){
   _service.getUserData().then((value){
    this.userDetails=value;
    notifyListeners();
    });
  }
}
```

Cloud Firestore:

Categories:



categories > DVBZurmnKSm..

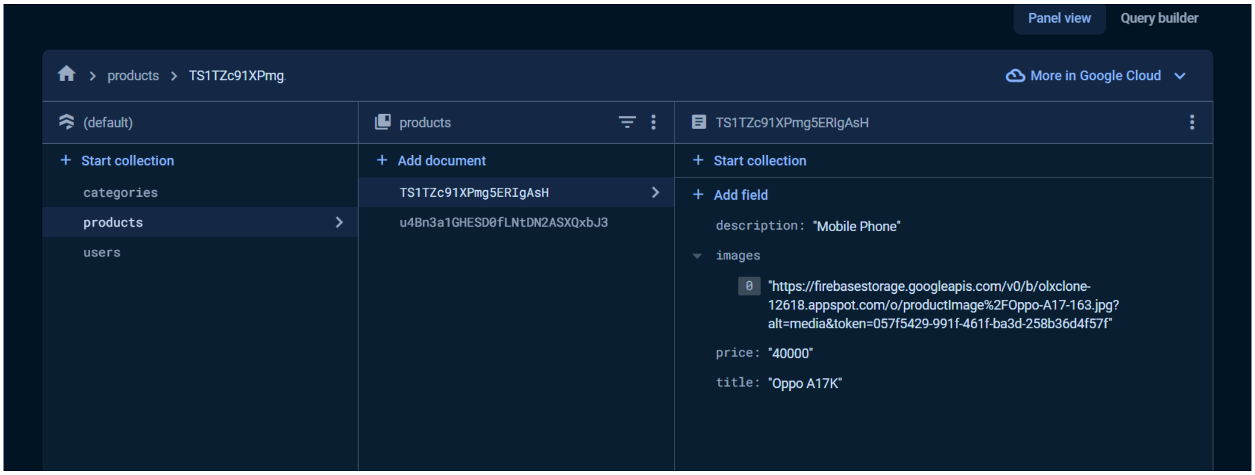
(default)
categories
products
users

categories
DVBZurmnKSmshXgvwKW3
QDKYFLRmZLre72GpaxMX
U99z3XauQNEMY2wJLAjc
a486DUA0GGTcv7sTKIK0
oW4s7RMRvsQxtOhhbtqE
zyx6snseb0VV0I8XrLy7

DVBZurmnKSmshXgvwKW3

```
brands
    0 "iPhone"
    1 "Samsung"
    2 "MI"
    3 "Realme"
    4 "Oppo"
    5 "Vivo"
    6 "POCO"
```

products:



products > TS1TZc91XPmg.

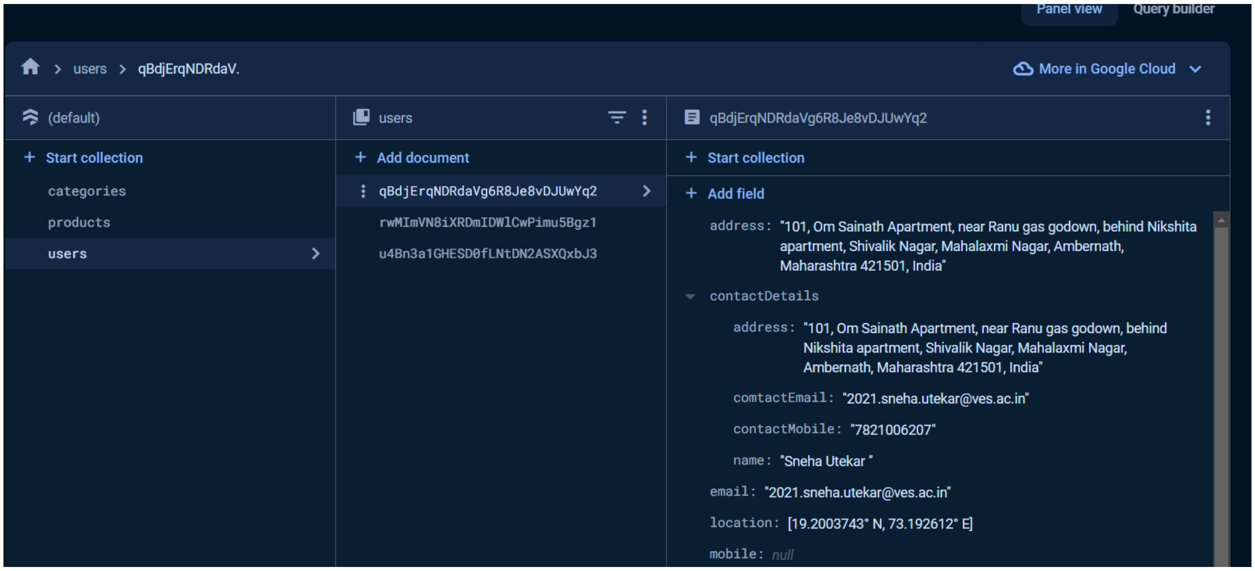
(default)
categories
products
users

products
TS1TZc91XPmg5ERIgAsH
u4Bn3a1GHESD0fLNtDN2ASXQxbJ3

TS1TZc91XPmg5ERIgAsH

```
description: "Mobile Phone"
images
    0 "https://firebasestorage.googleapis.com/v0/b/olxclone-12618.appspot.com/o/productImage%2FOppo-A17-163.jpg?alt=media&token=057f5429-991f-461f-ba3d-258b36d4f57f"
price: "40000"
title: "Oppo A17K"
```

users:



```
address: "101, Om Sainath Apartment, near Ranu gas godown, behind Nikshita
         apartment, Shivalik Nagar, Mahalaxmi Nagar, Ambernath,
         Maharashtra 421501, India"
contactDetails
    address: "101, Om Sainath Apartment, near Ranu gas godown, behind
             Nikshita apartment, Shivalik Nagar, Mahalaxmi Nagar,
             Ambernath, Maharashtra 421501, India"
    comtactEmail: "2021.sneha.utekar@ves.ac.in"
    contactMobile: "7821006207"
    name: "Sneha Utekar "
email: "2021.sneha.utekar@ves.ac.in"
location: [19.2003743° N, 73.192612° E]
mobile: null
```

CONCLUSION:

We have understood how to connect the database with flutter UI.