

Aim - To connect Flutter UI with firebase database.

Theory -

- What is firebase ?

- 1. Firebase is a Backend-as-a-Service (BaaS) platform that provides developers with a cloud-based infrastructure for storing and managing data and building real-time, scalable apps.
- 2. It offers a wide range of services such as authentication, database storage, hosting and more, which can be easily integrated with other web and mobile applications.

- What are the steps to connect firebase with flutter ui ?

- 1. Set up a Firebase project in the Firebase console.
- 2. Install the firebase-core and firebase-auth packages in your Flutter project.
- 3. Import firebase-core and firebase-auth into your Flutter code.
- 4. Add your Firebase configuration information to your flutter app.
- 5. Initialize firebase in your flutter app.
- 6. Use the firebase services in your flutter UI by creating instances of the relevant firebase classes and calling their methods.

**This application was built and tested in this environment**

**Android Studio - Version 2022.1**

**Flutter SDK - Version 3.7.0-30.0.pre.1**

**JDK - Version - 11.0.16.1**

**Code -**

**google-services.json**

```
{
  "project_info": {
    "project_number":
"121295635615",
    "project_id": "clone-46d99",
    "storage_bucket":
"clone-46d99.appspot.com"
  },
  "client": [
    {
      "client_info": {
        "mobilesdk_app_id":
"1:121295635615:android:f03e560b7
79a2852f10eaa",
        "android_client_info": {
          "package_name":
"com.igor.usersapp"
        }
      },
      "oauth_client": [
        {
          "client_id":
"121295635615-qkuvsm85c21568v87
pcn72n98c42fumg.apps.googleuserco
ntent.com",
          "client_type": 1,
          "android_info": {
            "package_name":
"com.igor.usersapp",
            "certificate_hash":
"1a94eabdbeb7cfefb57a2069bc4c5c8
4e500c034"
          }
        },
        {
          "client_id":
"121295635615-ma85auc6tsbas0vpo8
inq4vjv704kpap.apps.googleuserconte
nt.com",
          "client_type": 3
        }
      ],
      "api_key": [
        {
          "current_key":
"A1zaSyBhsQ1GefLe9s9CGaIaK9ezP
05A-5p_Ans"
        }
      ],
      "services": {
        "appinvite_service": {
          "other_platform_oauth_client":
[
            {
```

```

        "client_id":
"121295635615-ma85auc6tsbas0vpo8
inq4vjv704kpap.apps.googleuserconte
nt.com",
        "client_type": 3
    }
]

```

```

    }
}
},
"configuration_version": "1"
}

```

## main.dart

```

import
'package:firebase_core/firebase_core.
dart';
import 'package:flutter/material.dart';
import
'package:provider/provider.dart';
import
'package:shared_preferences/shared_p
references.dart';

import
'package:users_app/assistantMethods/
address_changer.dart';
import
'package:users_app/assistantMethods/
cart_item_counter.dart';
import
'package:users_app/assistantMethods/
total_amount.dart';

import
'package:users_app/global/global.dart'
;

```

```

import
'package:users_app/splashScreen/my_
splash_screen.dart';

Future<void> main() async {

WidgetsFlutterBinding.ensureInitializ
ed();

    sharedPreferences = await
SharedPreferences.getInstance();

    await Firebase.initializeApp();
    runApp(const MyApp());
}

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

    // This widget is the root of your
    application.
    @override

```

```

Widget build(BuildContext context)
{
  return MultiProvider(
    providers: [
      ChangeNotifierProvider(create:
(c) => CartItemCount()),
      ChangeNotifierProvider(create:
(c) => TotalAmount()),
      ChangeNotifierProvider(create:
(c) => AddressChanger()),
    ],
    child: MaterialApp(

```

```

debugShowCheckedModeBanner:
false,
    title: 'Users App',
    theme: ThemeData(
      primarySwatch: Colors.purple,
    ),
    home: const MySplashScreen(),
  ),
);
}
}

```

### **android/build.gradle**

```

buildscript {
  ext.kotlin_version = '1.6.10'
  repositories {
    google()
    mavenCentral()
  }

  dependencies {
    classpath
'com.android.tools.build:gradle:7.1.2'
    classpath
"org.jetbrains.kotlin:kotlin-gradle-plu
gin:$kotlin_version"
    classpath
'com.google.gms:google-services:4.3.
13'
  }
}

allprojects {

```

```

repositories {
  google()
  mavenCentral()
}

rootProject.buildDir = '../build'
subprojects {
  project.buildDir =
"${rootProject.buildDir}/${project.na
me}"
}
subprojects {

  project.evaluationDependsOn(':app')
}

task clean(type: Delete) {
  delete rootProject.buildDir
}

```

## android/app/build.gradle

```
def localProperties = new Properties()
def localPropertiesFile =
rootProject.file('local.properties')
if (localPropertiesFile.exists()) {

localPropertiesFile.withReader('UTF-
8') { reader ->
    localProperties.load(reader)
}
}

def flutterRoot =
localProperties.getProperty('flutter.sd
k')
if (flutterRoot == null) {
    throw new
GradleException("Flutter SDK not
found. Define location with flutter.sdk
in the local.properties file.")
}

def flutterVersionCode =
localProperties.getProperty('flutter.ver
sionCode')
if (flutterVersionCode == null) {
    flutterVersionCode = '1'
}

def flutterVersionName =
localProperties.getProperty('flutter.ver
sionName')
if (flutterVersionName == null) {
```

```
    flutterVersionName = '1.0'
}

apply plugin:
'com.android.application'
apply plugin: 'kotlin-android'
apply from:
"$flutterRoot/packages/flutter_tools/g
radle/flutter.gradle"
apply plugin:
'com.google.gms.google-services'

android {
    compileSdkVersion 33
    ndkVersion flutter.ndkVersion

    compileOptions {
        sourceCompatibility
JavaVersion.VERSION_1_8
        targetCompatibility
JavaVersion.VERSION_1_8
    }

    kotlinOptions {
        jvmTarget = '1.8'
    }

    sourceSets {
        main.java.srcDirs +=
'src/main/kotlin'
    }
```

```

defaultConfig {
    // TODO: Specify your own
    unique Application ID
    (https://developer.android.com/studio/
    build/application-id.html).
    applicationId
    "com.igor.usersapp"
    // You can update the following
    values to match your application
    needs.
    // For more information, see:
    https://docs.flutter.dev/deployment/an
    droid#reviewing-the-build-configurati
    on.
    minSdkVersion 21
    targetSdkVersion
    flutter.targetSdkVersion
    versionCode
    flutterVersionCode.toInteger()
    versionName
    flutterVersionName
    multiDexEnabled true
}

```

```

buildTypes {
    release {
        // TODO: Add your own
        signing config for the release build.
        // Signing with the debug keys
        for now, so `flutter run --release`
        works.
        signingConfig
        signingConfigs.debug
    }
}

flutter {
    source '../..'
}

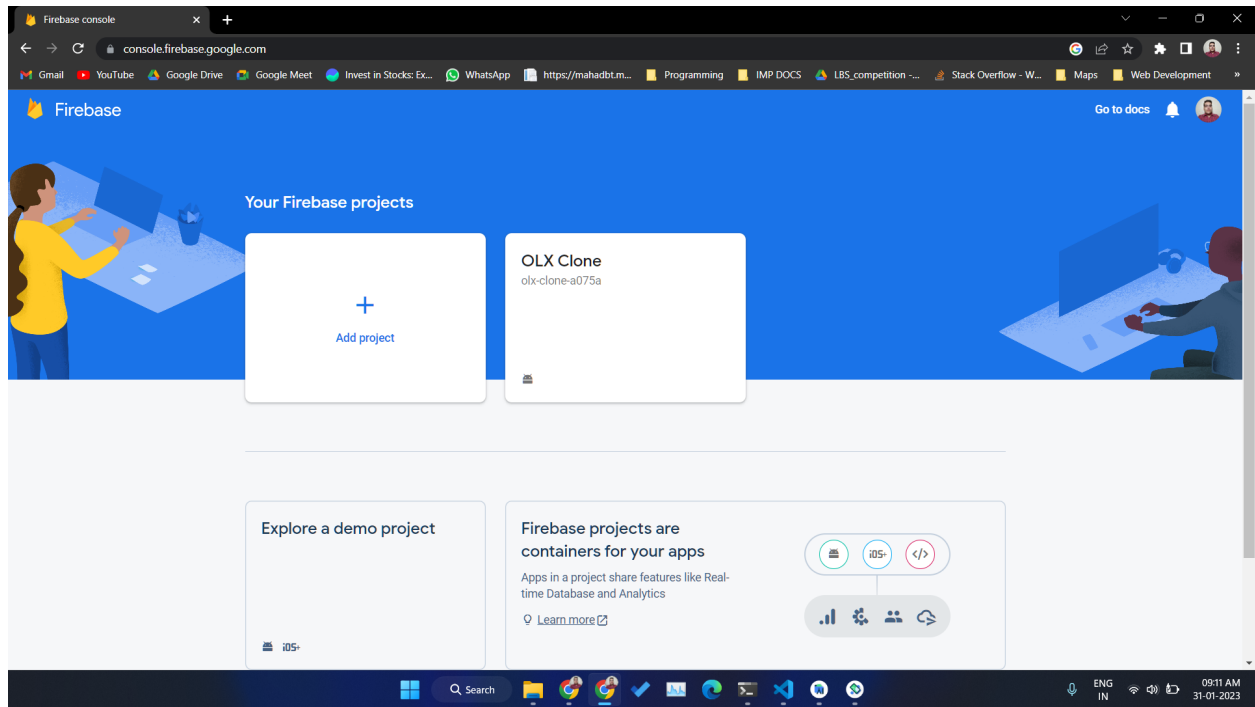
dependencies {
    implementation
    "org.jetbrains.kotlin:kotlin-stdlib-jdk7
    :$kotlin_version"
}

```

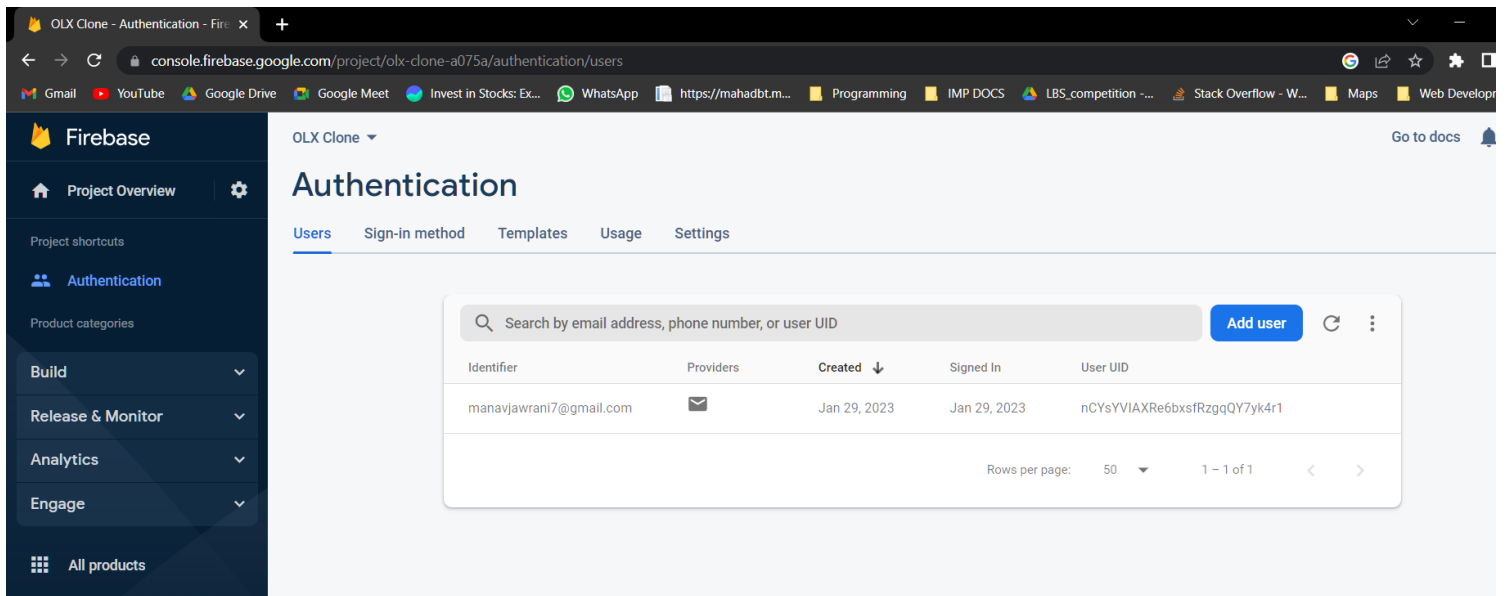


## Screenshots -

### 1. Firebase Console



### 2. Firebase console with one user (manavjawrani7@gmail.com)



### 3. Firebase console by adding one more user (manavjawrani700@gmail.com)

OLX Clone

Go to docs

Authentication

[Users](#)[Sign-in method](#)[Templates](#)[Usage](#)[Settings](#)

Search by email address, phone number, or user UID

Add user

Identifier	Providers	Created	Signed In	User UID
manavjawrani700@gmail.c...		Jan 31, 2023		AUvlqetA8mhPrFbttLrsvq5GtjH2
manavjawrani7@gmail.com		Jan 29, 2023	Jan 29, 2023	nCYsYVIAXRe6bxsRzgqQY7yk4r1

Rows per page: 501 - 2 of 2