Experiment 6

Name:Umesh artani
Div: D15A
Roll no: 03

Aim: How To Set Up Firebase with Flutter for iOS and Android Apps

Theory:

Setting up Firebase with Flutter for iOS and Android apps involves several steps to integrate Firebase services seamlessly into your mobile application. Here's a high-level overview of the process without providing specific code:

1. Step 1: Create a Firebase Project
- Firebase Console: Go to the Firebase Console (https://console.firebase.google.com/) and create a new project.
- Add Your App: In the Firebase Console, add your Flutter app by providing its package name for Android or bundle identifier for iOS.

2. Step 2: Configure Firebase SDKs
- Download Configuration Files: After adding your app, download the configuration files for both Android (google-services.json) and iOS (GoogleService-Info.plist).
- Add Configuration Files: Place the configuration files in the appropriate directories within your Flutter project.

3. Step 3: Configure Your Project
   1. Android Configuration:
- For Android, place the google-services.json file in the android/app directory of your Flutter project.
- Configure the project-level build.gradle file to include the Google Services plugin.
   2. iOS Configuration:
- For iOS, place the GoogleService-Info.plist file in the root of your iOS project's Runner directory.
- Configure the Info.plist file of your iOS project to include necessary settings for Firebase.
4. Step 4: Integrate Firebase SDKs
- Add Firebase Plugins: Add the necessary Firebase Flutter plugins to your pubspec.yaml file. These plugins enable communication with Firebase services such as Authentication, Cloud Firestore, Cloud Messaging, etc.
- Initialize Firebase: Initialize Firebase in your Flutter app's entry point (usually main.dart) using Firebase.initializeApp() method.

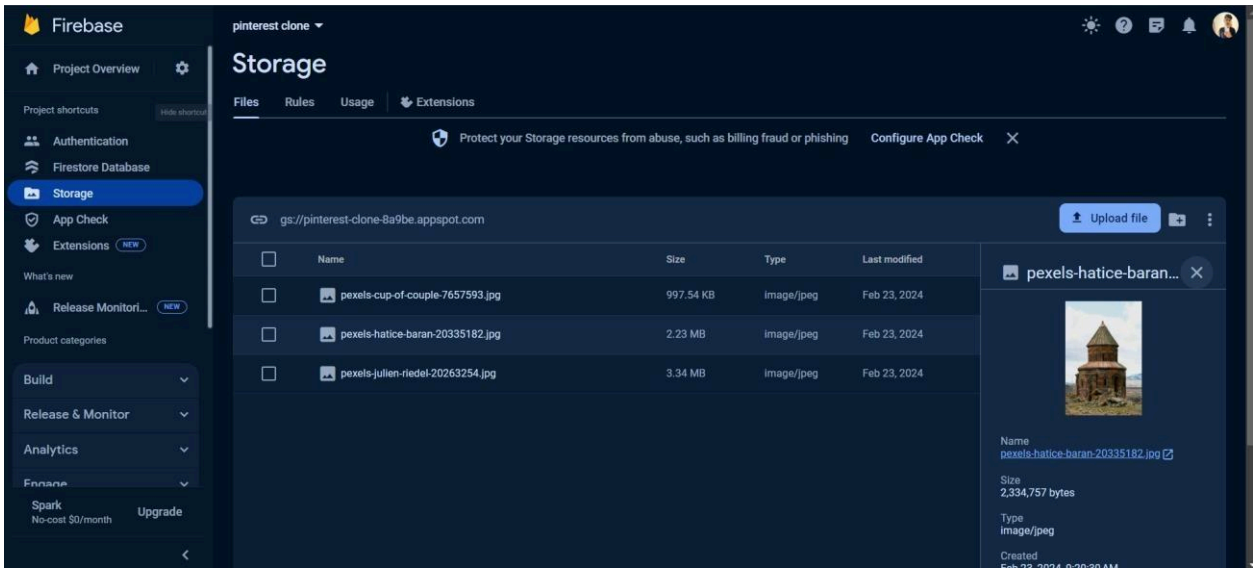
5. Step 5: Use Firebase Services

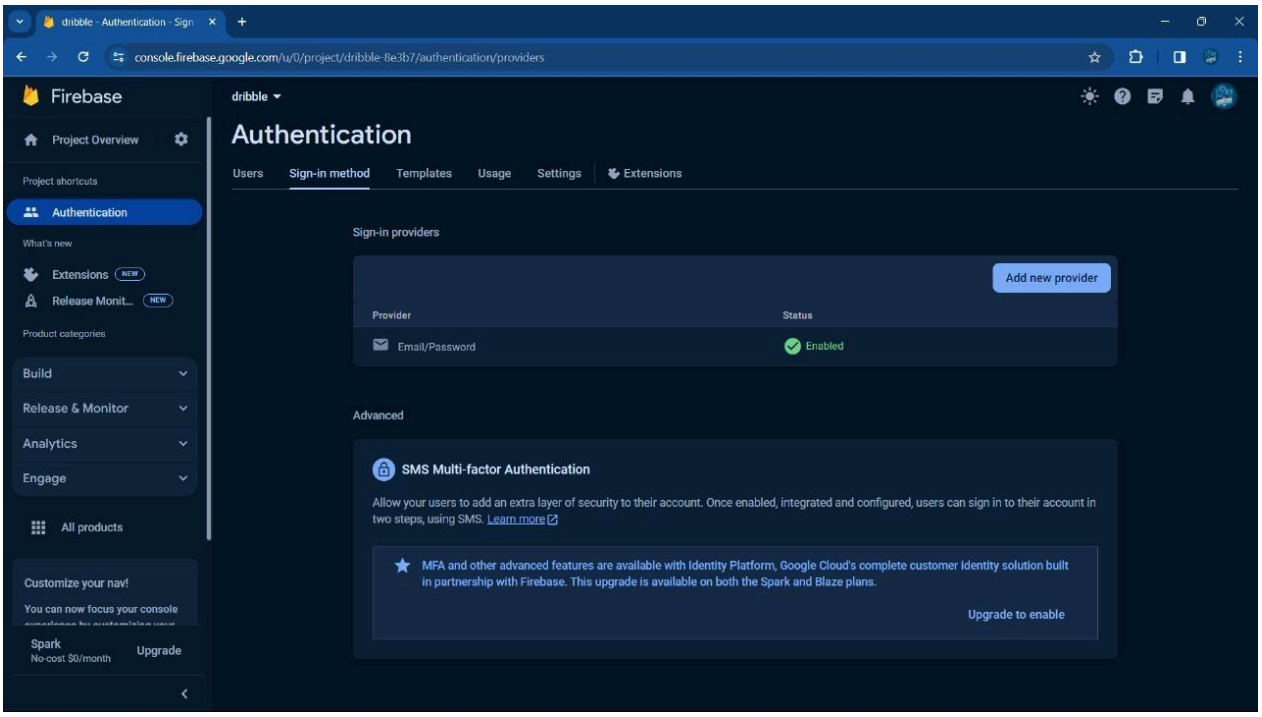
- Authentication: Implement user authentication using Firebase Authentication services. This allows users to sign up, sign in, and manage their accounts.
- Database: Utilize Cloud Firestore or Realtime Database to store and retrieve data from the cloud. Design your data model and interact with the database using Firebase SDK methods.
- Cloud Functions: Implement serverless backend logic using Cloud Functions for Firebase. This allows you to run custom backend code in response to events triggered by Firebase features and HTTPS requests.
- Cloud Messaging: Implement push notifications using Firebase Cloud Messaging (FCM) to engage users with timely and relevant messages.

6. Step 6: Test and Deploy

- Test Your App: Test your Flutter app thoroughly to ensure that Firebase services are integrated correctly and functioning as expected.
- Deploy Your App: Deploy your Flutter app to the Google Play Store for Android or the Apple App Store for iOS, making it available to users.
- By following these steps, you can successfully set up Firebase with Flutter for both iOS and Android apps, enabling powerful features and capabilities to enhance your application.

App UI:





References: https://github.com/MujtabaNasir/Pinterest_Clone_Flutter https://iconscout.com/all-assets/pinterest

Conclusion: Therefore understood setup of firebase and how to use it our flutter app