

Last Updated: 24<sup>th</sup> November 2021. [ **NEW** Point 28, 32]

Please follow these bellow steps in the mentioned sequence for quick & easy installation:

## BASIC SETUP ON YOUR COMPUTER

1. Make sure you have installed any flutter 2.5+ version. (App Built on **flutter 2.5.3** & **dart2.14.4**).
2. Get ready with your unique **app name** & **app package name** *com.yourcompanyname.appname*
3. **Download** the Fiberchat User app **source code**. Open the source code (**fiberchat** folder) in your preferred Code Editor. We recommend to use **VS Code** especially for flutter apps.
4. Replace the default package name **com.fiberchat.fiberchat** with your app package name- ***com.yourcompanyname.appname*** inside the *source code/android* folder. ) as per point **23**.
5. **Replace** the **fiberchat** label wherever available in the source code ( *except in pubspec.yaml & import statements inside every .dart file in the lib folder. No need to change there*) as per point **23**.

## FIREBASE SETUP

6. Go to firebase official website : <https://firebase.google.com/>
7. **Login** using your desired google account.
8. **Create a new project** with your **app name** as *Project name*. Select the **analytics location** country from where you are launching the app (country from where your targeted user base belongs to)
9. Go to firebase console and **upgrade the billing account** to **PAY AS YOU GO** located at the bottom right in the console. Provide your card information & verify the details to enable billing account (*Billing account is required so that cloud functions can be uploaded in later stages*)
10. Go to **Google Cloud Platform**, login using same google account you have used for firebase then head over to Dashboard/**Console** and in the **API section** search for **Android Device Verification API** and enable the API from the **Enable API** located there. Make sure you have same **Selected your firebase project** from the top left while enabling this API (*Required to SKIP captcha in the phone authentication page in app*)
11. Again go to Firebase Console,  
**Register your android & iOS apps** using your app package name by clicking on the android & ios icon in the Firebase console and simply proceeding by clicking on next>next>done button.
12. Once both apps are displayed in Firebase console. Go to **project settings** at the top left and select the **Android user app** section: **SHA1** & **SHA256** fingerprint fields.
13. You need to have your **debug** & **release key** generated from your **computer** for the app to connect with firebase. Simply copy and paste those **SHA1** & **SHA256** fingerprints code in the previous step for both debug & release mode .
14. Go to **Authentication** section in Firebase console from the left menu, **enable** the **Phone authentication** (for users) and **anonymous** authentication (for admin – if applicable).

15. Go to **Cloud Firestore** Section from the left-menu. And select the **server location** from the popup (we recommend to select **nam5-uscentral** as it is the fastest multi-region server located in US and serve the whole world with strong capability & uptime.) if you want the app to be limited to a particular country, then you can choose the nearest regional server located to that country.
16. Once the cloud firestore is ready, Go to **Rules section** in the **Cloud firestore** window. Copy the whole codes rules provided in the [FirestoreSecurityRules.md](#) file in the source code folder & paste these rules in the Rules section and click on **Publish**.
17. Similarly, Go to **Firebase Storage** section & open the **Rules** window. Copy the whole codes rules provided in the [StorageSecurityRules.md](#) file in the source code folder & paste these rules in the Rules section and click on **Publish**.
18. Firebase account is *now ready* to host your app.
19. Head over to the terminal in the code editor, and login to your Firebase account using **Firebase CLI** (to install Firebase CLI tools, refer this: <https://www.npmjs.com/package/firebase-tools>) so that you can upload the Cloud Functions to make the notification feature work. You can follow these steps: Add firebase project id to fiberchat/.firebaserc file. Run below commands:
  1. **firebase login**
  2. **firebase deploy --only functions**
 Wait for the process to complete & You will see a *successfully deployed* message in the terminal. Push notification feature is successfully integrated.

### 3<sup>RD</sup> PARTY API INTEGRATION

20. Get **Agora App ID** (for Audio & Video calls)
  - a) Refer the *Agora\_Setup\_Guide.pdf* attached within the source code folder.
  - b) Paste the Agora App ID generated in the > fiberchat/lib/Configs/app\_constant.dart file.
21. Get **Giphy API Key** (for GIF Gallery in chat)
  - a) Refer the *Giphy\_Setup\_Guide.pdf* attached within the source code folder.
  - b) Paste the Giphy API key generated in the > fiberchat/lib/Configs/app\_constant.dart file.
22. Enable **Firebase Crashlytics** :
 

Go to Firebase Dashboard Crashlytics > Click on “Enable Crashlytics”.

## ANDROID APP SETUP

### 23. Change Package Name in Following Files:

1. android/app/src/debug/**AndroidManifest.xml**
2. android/app/src/main/kotlin/com/fiberchat/fiberchat/**Application.kt**
3. android/app/src/main/kotlin/com/fiberchat/fiberchat/**MainActivity.kt**
4. android/app/src/main/res/values/**strings.xml**
5. android/app/src/main/**AndroidManifest.xml**
6. android/app/src/profile/**AndroidManifest.xml**
7. android/app/**build.gradle**

### Change following Folder names with your App name:

1. android/app/src/main/kotlin/**com/fiberchat/fiberchat**
2. android/**fiberchat\_android.iml**
3. *fiberchat*/**fiberchat.iml**

Execute the command:

***flutter clean***

***flutter pub get***

24. Download the **google-services.json** file from the *FirebaseConsole>ProjectSettings>Android User App*. Paste the downloaded file inside *fiberchat/android/app/google-services.json*
25. Replace the assets located in the *fiberchat/assets* folder with your reskin assets. Make sure file name & extension should remain the same as it is provided.
26. Set the **app version** number in *fiberchat/pubspec.yaml* & *android/local.properties* file.
27. **Set the information** required in the *fiberchat/lib/Configs/app\_constant.dart* file.
28. Execute the command to set the app icon for Android & iOS automatically from *assets/appicon/appicon.jpg*.

***flutter pub get***

***flutter pub run flutter\_launcher\_icons:main***

**NEW** Also replace the custom notification icon in the following path with the same file name: ***android/app/src/main/res/drawable/transparent.png***

Also change color value of the notification icon in  
***android/app/src/main/res/values/color.xml***

29. Android app is ready for production release.
30. For testing/debugging the app, use command:

*flutter clean*

*flutter run --debug*

31. Make sure security rules is already setup as per Point 16 & 17.
32. **NEW** Create **Firestore Queries** exactly as shown in the attached FIRESTORE\_QUERY\_EXAMPLE.jpg provided in the source code bundle.
33. For building the release.apk file, use command:

*flutter clean*

*flutter run --release*

Please note: .APK file size will be more than 300MB but once .aab gets publish in Playstore app size will be between 30 to 70MB only.

34. For generating the app bundle package for Publishing in Google Playstore, use command:

*flutter clean*

*flutter build appbundle --release*

## IOS SETUP

35. In the terminal, run following command:

*flutter clean*

*flutter pub get*

*cd ios*

*pod install*

*pod update repo* (run only if the above commands throws any error)

36. Open the ios folder present in the source code in Xcode.
37. Now, Download the Google Services file from the iOS User App. Paste the downloaded file inside *fiberchat/assets/ios\_assets/GoogleServices-Info.plist*
38. Open Xcode, right click on **runner** > **Add files to runner**, then select the **GoogleServices-Info.plist** & **ringtone.caf** & **blank.caf** present in the path *fiberchat/assets/ios\_assets/* and click **Add**.
39. Check if all the app name, package name, version is correct in the **General** Section & Make sure your *bundle name* is showing correct in **Info** section in the Xcode.
40. Go to **Signing & Capabilities**, select or add the *team* & *bundle identifier*. Also make sure, *background fetch* & *Remote notification* is ticked in the Background Modes & Push notification is also enabled.
41. Run on any iOS device (better if real device)
42. Got to > **product** and click these one after another:

*Clean build folder*

### ***Build***

### ***Archive***

43. You are almost ready to go. Please refer this official link for detailed integration of iOS app in Firebase & publishing it to Apple store connect.

<https://flutter.dev/docs/deployment/ios>

For more detailed information of any particular, you can also refer the documentation file provided in the Source Code package.