

## **Thank You for your purchase**

Thank you for choosing our code. We appreciate your purchase and aim to provide exceptional service. If you need help with the code or documentation, contact our [Support Team](#).

## **Astroway Partner App**

Explore the official Flutter website at <https://docs.flutter.dev/get-started/install> for a complete installation guide.

Prefer video tutorials? Check out the playlist: [link](#)

## **Steps to run this project successfully:**

1. Begin by extracting the downloaded code, resulting in the creation of the Partner App - Flutter Code zip folder.
2. Next, extract the contents of the Partner App - Flutter Code folder and launch it in either Android Studio or Visual Studio Code.
3. Access the IDE terminal and enter the command "flutter pub get" to fetch the necessary dependencies.
4. If you want to run the app in IOS simulator (otherwise skip this step), then execute the following commands in the terminal:  
`cd ios  
pod install  
cd ..`
5. Finally, open the terminal and input "flutter run" to run the application. This will initiate the execution of the project.

## **Change app package name**

1. If you are using Android Studio, you can press **CTRL+SHIFT+R** to open Replace in Files tool.  
then search our package name **com.astroway.partner**
2. and replace it with your package name in all places.

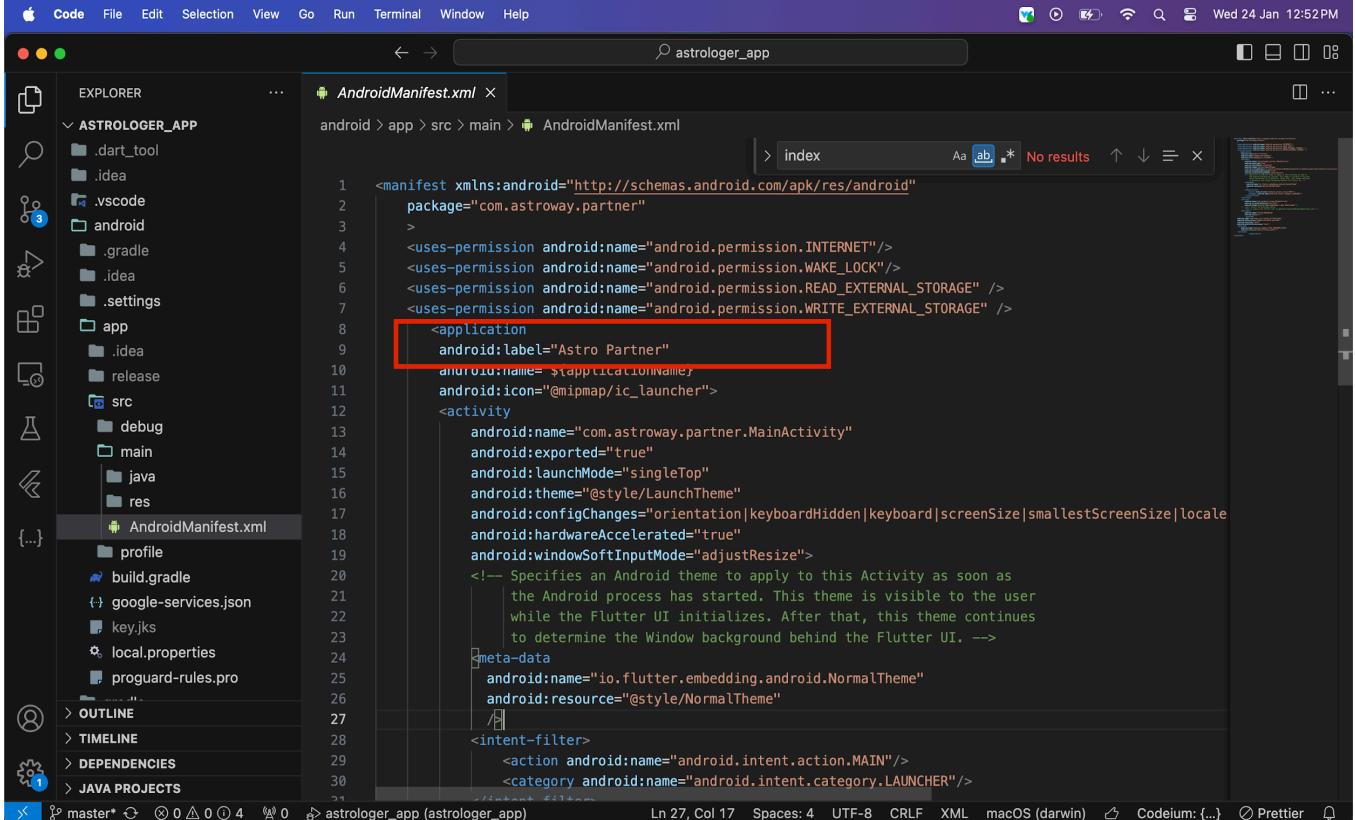
<https://drive.google.com/file/d/1qewHCPg1EaWeYPxiOPM4TmuuXS6qRP2o/view?usp=sharing>

3. And if you are using VS Code, you can press **CTRL+SHIFT+F** to open Find in Files Tool. then search our package name **com.astroway.partner** and replace it with your package name in all places.

<https://drive.google.com/file/d/1sMHNhmHzvYW0tMmHV8VRtmGbt9TIRjtM/view?usp=sharing>

## Change application name

Similarly you can change the app name by finding and replacing in all the places.



The screenshot shows the VS Code interface with the following details:

- File Explorer:** Shows the project structure under "ASTROLOGER\_APP".
- Editor:** Displays the "AndroidManifest.xml" file. The line containing the application label is highlighted with a red rectangle.
- Search Bar:** Shows the search term "astrologer\_app".
- Status Bar:** Shows the file path "astrologer\_app (astrologer\_app)", line count "Ln 27, Col 17", and other settings like "Spaces: 4", "UTF-8", "CRLF", "XML", "macOS (darwin)".

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.astroway.partner"
    >
    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.WAKE_LOCK"/>
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <application
        android:label="Astro Partner"
        android:name="$applicationName"
        android:icon="@mipmap/ic_launcher">
        <activity
            android:name=".MainActivity"
            android:exported="true"
            android:launchMode="singleTop"
            android:theme="@style/LaunchTheme"
            android:configChanges="orientation|keyboardHidden|keyboard|screenSize|smallestScreenSize|locale"
            android:hardwareAccelerated="true"
            android:windowSoftInputMode="adjustResize">
            <!-- Specifies an Android theme to apply to this Activity as soon as
                the Android process has started. This theme is visible to the user
                while the Flutter UI initializes. After that, this theme continues
                to determine the Window background behind the Flutter UI. -->
            <meta-data
                android:name="io.flutter.embedding.android.NormalTheme"
                android:resource="@style/NormalTheme"
            />
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
                <category android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
```

## Integrate firebase with your Project

- First, Create a new Firebase Project

← → ⌛ console.firebaseio.google.com ☆ ⓘ New Chrome available :

Create a project(Step 1 of 3)

Let's start with a name for your project<sup>②</sup>

Project name

[partner-26790](#) [Select parent resource](#)

**Continue**



This screenshot shows the first step of creating a Firebase project. The user has entered 'partner' as the project name. A red box highlights the 'Continue' button. The background features a 3D illustration of two people working on a laptop, with large orange gears visible behind them.

← → ⌛ console.firebaseio.google.com ☆ ⓘ New Chrome available :

Create a project(Step 2 of 2)

## Google Analytics for your Firebase project

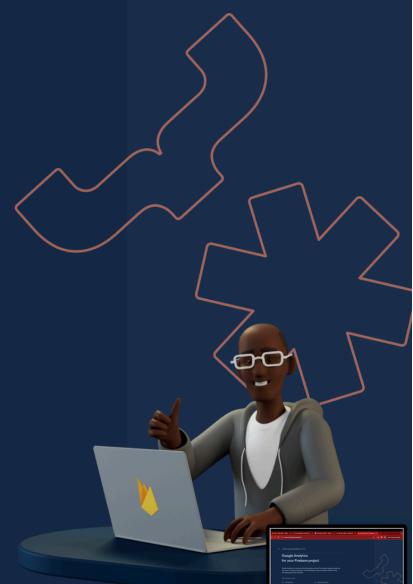
Google Analytics is a free and unlimited analytics solution that enables targeting, reporting and more in Firebase Crashlytics, Cloud Messaging, in-app messaging, Remote Config, A/B Testing and Cloud Functions.

Google Analytics enables:

- ✗ A/B testing ②
- ✗ User segmentation and targeting across Firebase products ②
- ✗ Crash-free users ②
- ✗ Event-based Cloud Functions triggers ②
- ✗ Free unlimited reporting ②

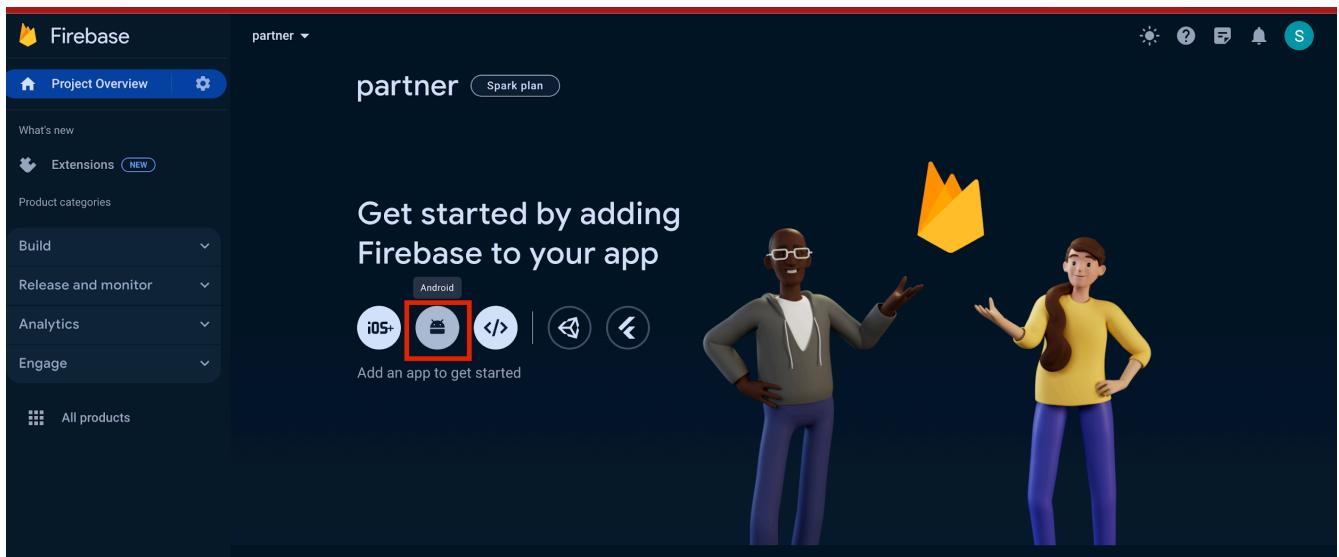
**Enable Google Analytics for this project**  
Recommended

Previous **Create project**



This screenshot shows the second step of creating a Firebase project, specifically enabling Google Analytics. A red box highlights the 'Enable Google Analytics for this project' checkbox, which is checked and labeled 'Recommended'. Another red box highlights the 'Create project' button at the bottom right. The background features a 3D illustration of a person working on a laptop, with large orange gears visible behind them.

- After creating your Firebase project, you'll now connect your Android application to it



[Go to docs](#)

### Add Firebase to your Android app

**1 Register app**

Android package name  **Add Your Package Name**

App nickname (optional)

Debug signing certificate SHA-1 (optional)  **add Your SHA-1 Key here**

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

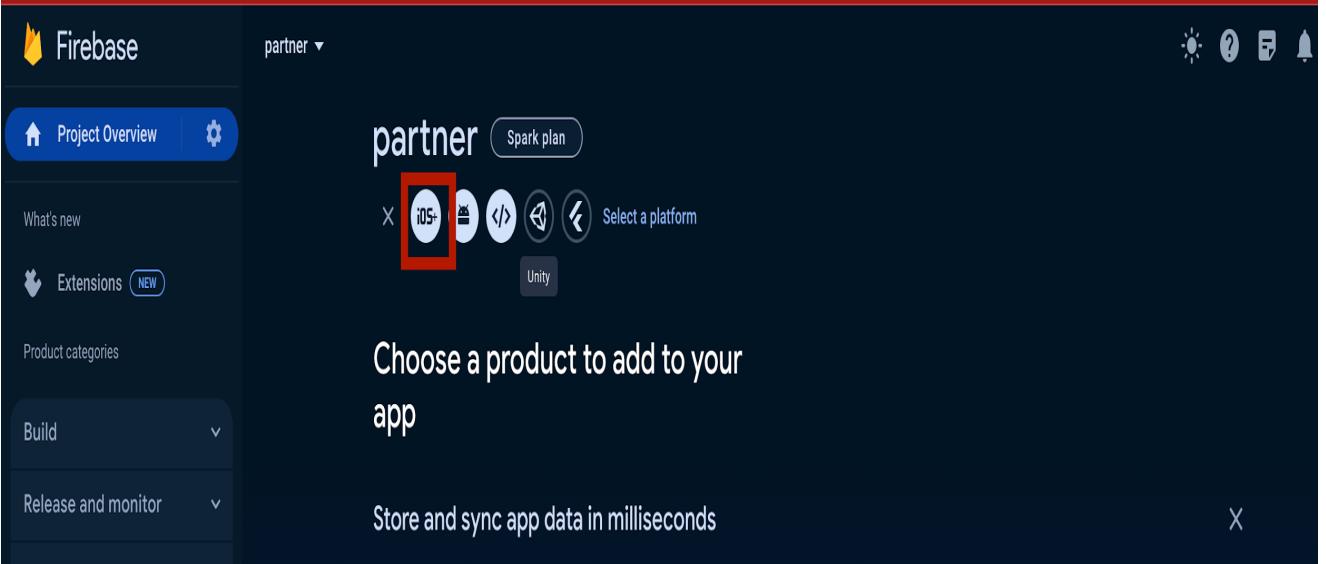
**Register app**

**2 Download and then add config file**

**3 Add Firebase SDK**

**4 Next steps**

## Now it's time for connecting IOS to your Firebase Project.



The screenshot shows the Firebase Project Overview page for a project named "partner". The "iOS" icon in the "Select a platform" section is highlighted with a red box. Below the header, the text "Choose a product to add to your app" is displayed, followed by "Store and sync app data in milliseconds".

**Add Firebase to your Apple app**

1 Register app

Apple bundle ID: com.partnerapp.org **Enter Your Bundle id here**

App nickname (optional): Partner

App Store ID (optional): 234325343

**Register app**

2 Download config file

3 Add Firebase SDK

4 Add initialisation code

5 Next steps

A blue laptop and smartphone icon are shown on the right side of the form.

## Add Firebase to your Apple app

- ✓ Register app  
Apple bundle ID: com.partnerapp.org, app nickname: Partner, App Store ID: 234325343

2 Download config file

[Download GoogleService-Info.plist](#)

Instructions for Xcode below | [Unity](#) | [C++](#)

Move the GoogleService-Info.plist file that you just downloaded into the root of your Xcode project and add it to all targets.



[Next](#)

3 Add Firebase SDK

4 Add initialisation code



## Skip this step

- ✓ Register app  
Apple bundle ID: com.partnerapp.org, app nickname: Partner, App Store ID: 234325343

Download config file

3 Add Firebase SDK

[CocoaPods](#) | [Download ZIP](#) | [Unity](#) | [C++](#)

Use [Swift Package Manager](#) to install and manage Firebase dependencies.

1. In Xcode, with your app project open, navigate to File > Add packages
2. When prompted, enter the Firebase iOS SDK repository URL:

[Copy](#)

3. Select the SDK version that you want to use.  
We recommend using the default (latest) SDK version, but you can use an older version, if needed.

4. Choose the Firebase libraries that you want to use.

After you click Finish, Xcode will automatically begin resolving and downloading your dependencies in the background.

[Previous](#)

[Next](#)

4 Add initialisation code

5 Next steps



To connect Firebase when your app starts up, add the initialisation code below to your app's main entry point.

SwiftUI    Swift    Objective-C

```
import SwiftUI
import FirebaseCore

class AppDelegate: NSObject, UIApplicationDelegate {
    func application(_ application: UIApplication,
                     didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey] = [:]) {
        FirebaseApp.configure()
        return true
    }
}

@main
struct YourApp: App {
    // register app delegate for Firebase setup
    @UIApplicationDelegateAdaptor(AppDelegate.self) var delegate

    var body: some Scene {
        WindowGroup {
            NavigationView {
                ContentView()
            }
        }
    }
}
```

Previous   [Next](#)

5 Next steps



### Add Firebase to your Apple app

- ✓ Register app  
Apple bundle ID: com.partnerapp.org, app nickname: Partner, App Store ID: 234325343
- ✍ Download config file
- ✍ Add Firebase SDK
- ✍ Add initialisation code

5 Next steps

You're ready!

Make sure that you take a look at the [documentation](#) to learn how to get started with each Firebase product that you want to use in your app.

You can also explore [sample Firebase apps](#).

Or, continue to the console to explore Firebase.

Previous   [Continue to the console](#)



Congrats! You have successfully connected your android & IOS app to your firebase project.

## 1. Enable OTPless Authentication

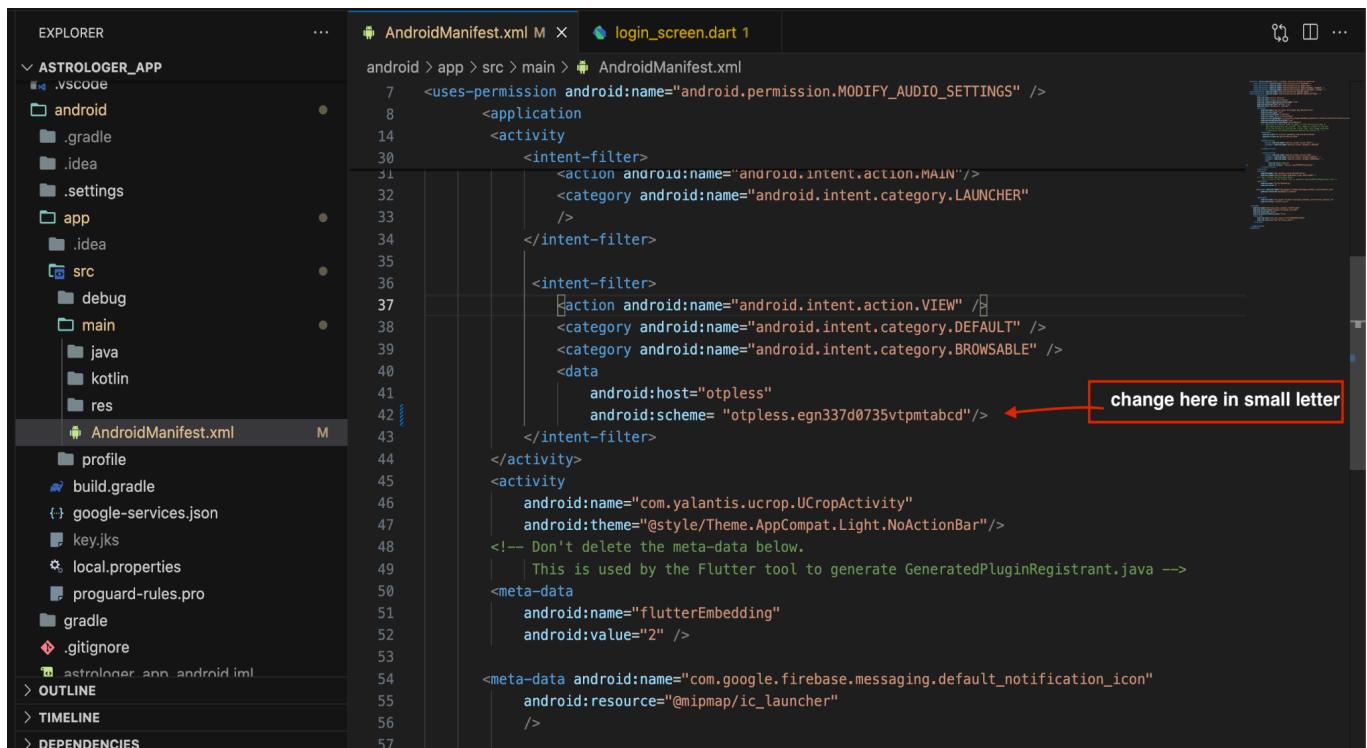
Get your otpless app id from this [otpless link](#)

Now Go to **android/app/src/main/AndroidManifest.xml**

And paste your otpless app id in small letter

**For example:-** `otpless.egn337d0735vtpmtabcd`  
(`otpless.your-otpless-appid`)

**NOTE :-** The otpless keys we are showing here are dummy keys and it will not work with your project so please get your otpless key and change them



```
EXPLORER          ...          AndroidManifest.xml M X      login_screen.dart 1
ASTROLOGER_APP
  .vscode
    android
      .gradle
      .idea
      .settings
    app
      .idea
    src
      debug
      main
        java
        kotlin
      res
    AndroidManifest.xml M
  profile
  build.gradle
  google-services.json
  key.jks
  local.properties
  proguard-rules.pro
  gradle
  .gitignore
  astrologer_app.android.iml
> OUTLINE
> TIMELINE
> DEPENDENCIES

Android > app > src > main > AndroidManifest.xml
7   <uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS" />
8     <application
14       <activity
30         <intent-filter>
31           <action android:name="android.intent.action.MAIN" />
32           <category android:name="android.intent.category.LAUNCHER" />
34         </intent-filter>
35         <intent-filter>
36           <action android:name="android.intent.action.VIEW" />
37           <category android:name="android.intent.category.DEFAULT" />
38           <category android:name="android.intent.category.BROWSABLE" />
39           <data
40             android:host="otpless"
41             android:scheme="otpless.egn337d0735vtpmtabcd"/> ← change here in small letter
42         </intent-filter>
43       </activity>
44     <activity
45       android:name="com.yalantis.ucrop.UCropActivity"
46       android:theme="@style/Theme.AppCompat.Light.NoActionBar" />
47     <!-- Don't delete the meta-data below.
48         This is used by the Flutter tool to generate GeneratedPluginRegistrant.java -->
49     <meta-data
50       android:name="flutterEmbedding"
51       android:value="2" />
52
53   <meta-data android:name="com.google.firebase.messaging.default_notification_icon"
54     android:resource="@mipmap/ic_launcher" />
55
56
57
```

Now go to **lib/utils/config.dart**

And find `OtplessappId` and paste your otpless app id here in capital letter

```
lib > utils > config.dart > appParameters
1 // ignore_for_file: non_constant_identifier_names, dangling_library_doc_comments
2
3 /**
4  * =====
5  * ***** BASE URL FOR API *****
6  */
7 String contactsupportEmail = "support@gmail.com";
8 String imgBaseUrl = "https://astroway.deploy.in/";
9 String pdfBaseUrl = "https://astroway.deploy.in/public";
10 String appMode = "LIVE";
11 String OtplessappId = "EGN337D0735VTPMTABCD"; ← add your otpless key here
12
13 String storiesIcon = "assets/images/stories_icon.png";
14
15 Map<String, dynamic> appParameters = {
16   "LIVE": {
17     "apiUrl": "https://astroway.deploy.in/api/",
18   },
19   "DEV": {
20     "apiUrl": "https://astroway.deploy.in/api/",
21   }
22};
```

**IMPORTANT -** you need to change your folders name inside kotlin folder  
goto

android/app/src/main/kotlin/com/astroway/astrologer\_app/MainActivity.kt

For example:-**com.astroway.astrologer\_app**

Then first folder name:-**com**

Second folder name:-**astroway**

Third folder name:- **astrologer\_app**

replace this package name With your own package name

And change that in manifest too like i have done below

```

<manifest xmlns:android="http://schemas.android.com/apk/res/android">
    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.WAKE_LOCK"/>
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.RECORD_AUDIO" />
    <uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS" />
    <application
        android:label="Astro Partner"
        android:name="${applicationName}"
        android:requestLegacyExternalStorage="true"
        android:usesCleartextTraffic="true"
        android:icon="@mipmap/ic_launcher">
        <activity
            android:name="com.astroway.astrologer_app.MainActivity" ←
            android:exported="true"
            android:launchMode="singleTop"
            android:theme="@style/LaunchTheme"
            android:configChanges="orientation|keyboardHidden|keyboard|screenSize|smallestScreenSize|locale"
            android:hardwareAccelerated="true"
            android:windowSoftInputMode="adjustResize">
            <!-- Specifies an Android theme to apply to this Activity as soon as
                the Android process has started. This theme is visible to the user.
            </activity>
    </application>
</manifest>

```

change here with new package name before .MainActivity

### 3.Create Firestore Database

Firebase

partner ▾

Project Overview

Project shortcuts

- Authentication
- Firestore Database**
- Realtime Database

What's new

Extensions (NEW)

Product categories

- Build
- Release and monitor
- Analytics
- Engage

All products

Cloud Firestore

Real-time updates, powerful queries and automatic scaling

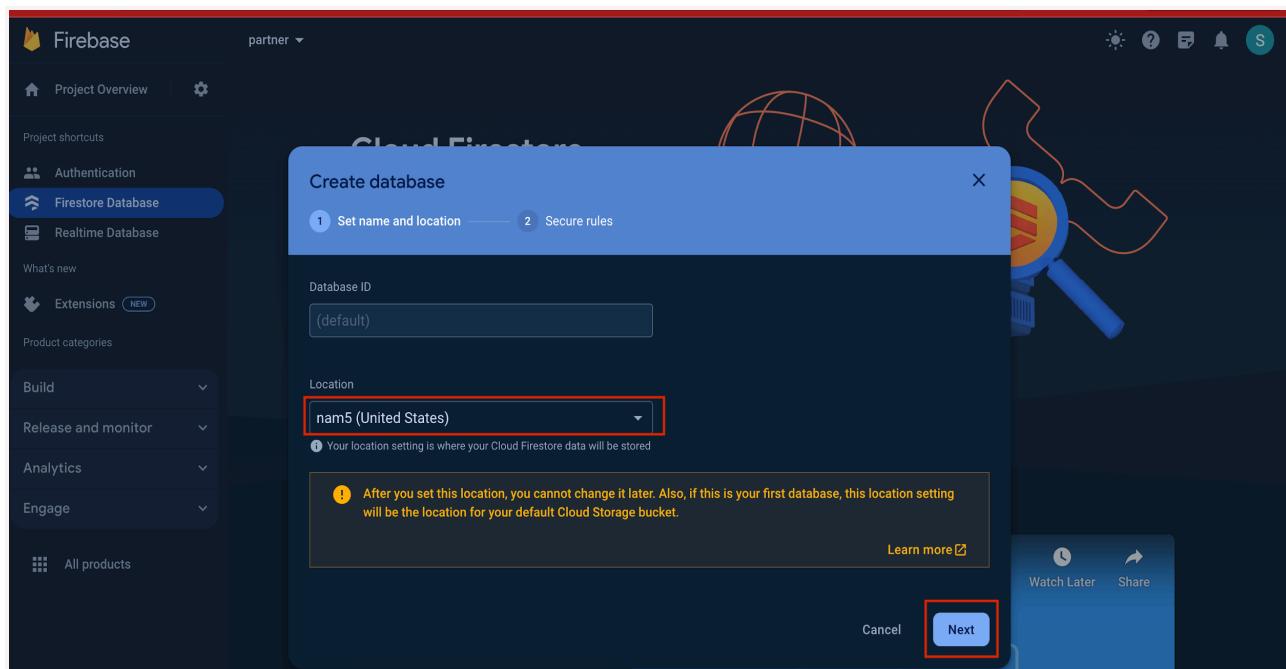
Create database

Learn more

How do I get started? View the docs

Introducing Cloud Firestore

Watch Later Share



Make Sure to add these rules in your firestore, otherwise it won't work.

```
rules_version = '2';
rules_version = '2';

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

    }
  }
}
```

Your security rules are defined as public, so anyone can steal, modify or delete data in your database

```
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {
    match /{document=**} {
      allow read, write: if true;
    }
  }
}
```

#### 4. Enable Firebase storage to support chat with images/pdf

In rules section paste the below code

```
service firebase.storage {
  match /b/{bucket}/o {
    match /{allPaths=**} {
      allow read, write
    }
  }
}
```

The screenshot shows the Firebase Storage interface. On the left sidebar, the 'Storage' option is selected and highlighted with a red box. The main area is titled 'Storage' and has tabs for 'Files', 'Rules', 'Usage', and 'Extensions'. The 'Files' tab is active. A sub-header says 'Protect your Storage resources from abuse, such as billing fraud or phishing' with a shield icon. Below it is a table with columns 'Name', 'Size', 'Type', and 'Last modified'. Two folders are listed: '101\_609/' and '104\_609/'. An 'Upload file' button is at the top right of the table.

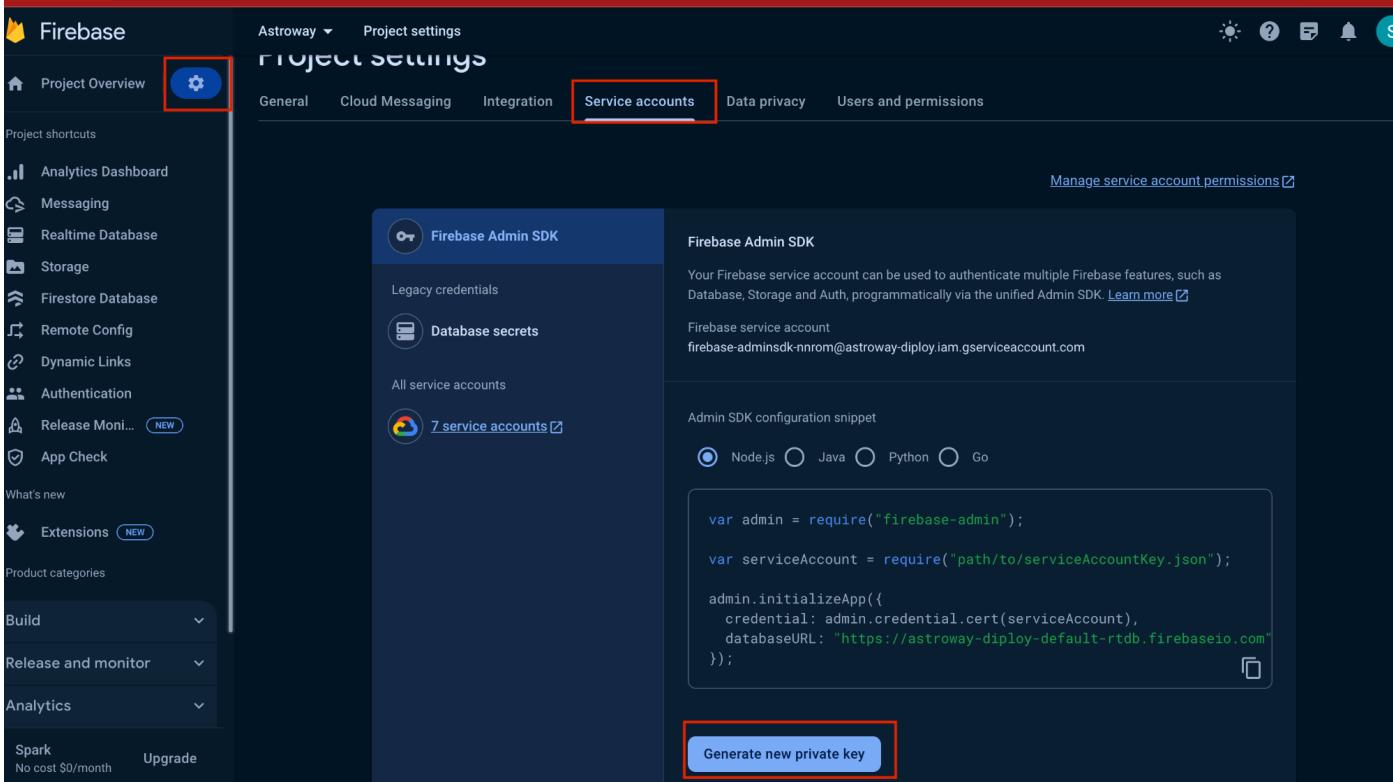
The screenshot shows the Firebase Storage interface. The 'Rules' tab is active. A central message says 'Write security rules that control access to Storage based on the contents of your Firestore Database.' with a star icon. Below this is a large text area containing a sample security rule:

```
1 service firebase.storage {
2   match /b/{bucket}/o {
3     match /{allPaths=**} {
4       allow read, write
5     }
6   }
7 }
8 }
```

To the left of the code editor, there's a small circular icon with a dog's face and the text 'Guard your data with rules that define who has access to it and how it is structured' followed by a 'View the docs' link.

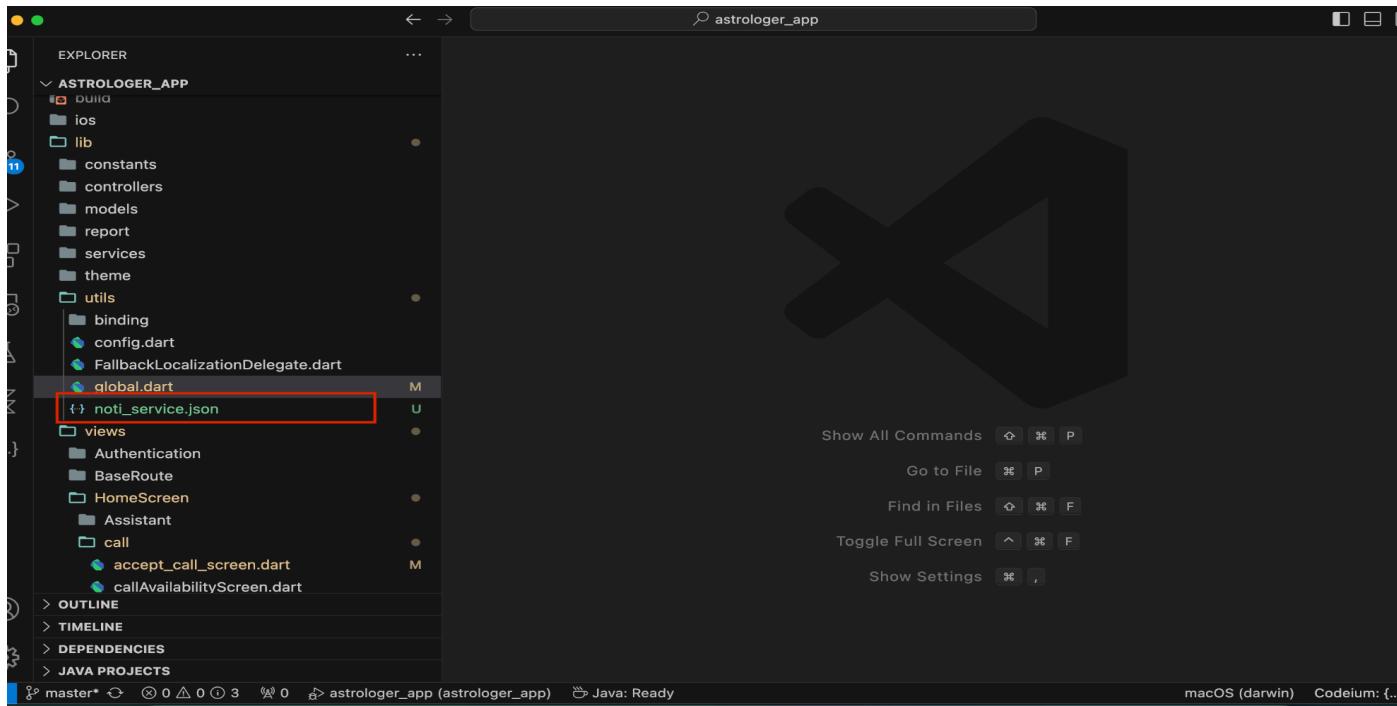
## Enable Notifications for App

Access your Firebase project and locate "Project Settings" > "Service account TAB" and click on [Generate new Private key](#)



The screenshot shows the Firebase Project Settings interface. On the left, there's a sidebar with various project management options like Analytics Dashboard, Messaging, Realtime Database, Storage, Firestore Database, Remote Config, Dynamic Links, Authentication, Release Monitor, App Check, Extensions, Build, Release and monitor, and Analytics. The main area is titled 'PROJECT SETTINGS' and has tabs for General, Cloud Messaging, Integration, Service accounts (which is highlighted with a red box), Data privacy, and Users and permissions. Under the Service accounts tab, it says 'Firebase Admin SDK' and lists Legacy credentials, Database secrets, All service accounts, and a link to 7 service accounts. It also includes an 'Admin SDK configuration snippet' for Node.js, Java, Python, and Go. At the bottom right of this section, there's a red box highlighting the 'Generate new private key' button.

## 1. Download the file.json and save it into lib/utils/noti\_service.json

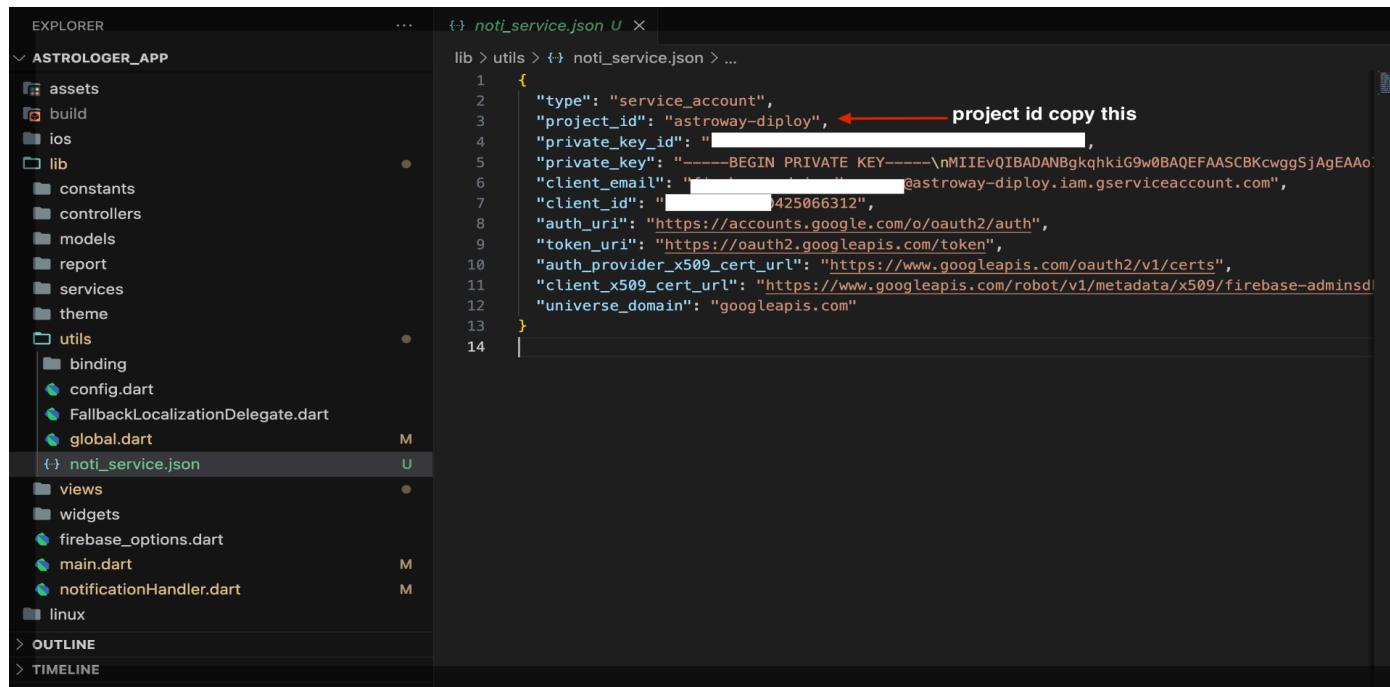


## 2. Now go to lib/utils/global.dart

```
lib > utils > global.dart > sendNotification
197 Future<void> sendNotification({
238   "requestType": requestType,
239   "id": id,
240   "charge": charge,
241   "fcmToken": fcmToken,
242   "astroProfile": astroProfile,
243   "videoCallCharge": videoCallCharge
244 },
245 "android": {
246   "notification": {"click_action": "android.intent.action.MAIN"}
247 }
248 );
249
250 final url = Uri.parse([
251   'https://fcm.googleapis.com/v1/projects/astroway-diploy/messages:send'];
252
253 final response = await http.post(
254   url,
255   headers: headers,
256   body: json.encode(data),
257 );
258 log('noti response ${response.body}');
259 if (response.statusCode == 200) {
260   log('Notification sent successfully');
261 } else {
262   log('Failed to send notification: ${response.body}');
263 }
264 } catch (e) {
265   print('Error sending notification: $e');
266 } finally {
267   client.close();
268 }
```

Replace this project ID with your own ID

And replace the projectId from your project id which is in the .json file you downloaded earlier

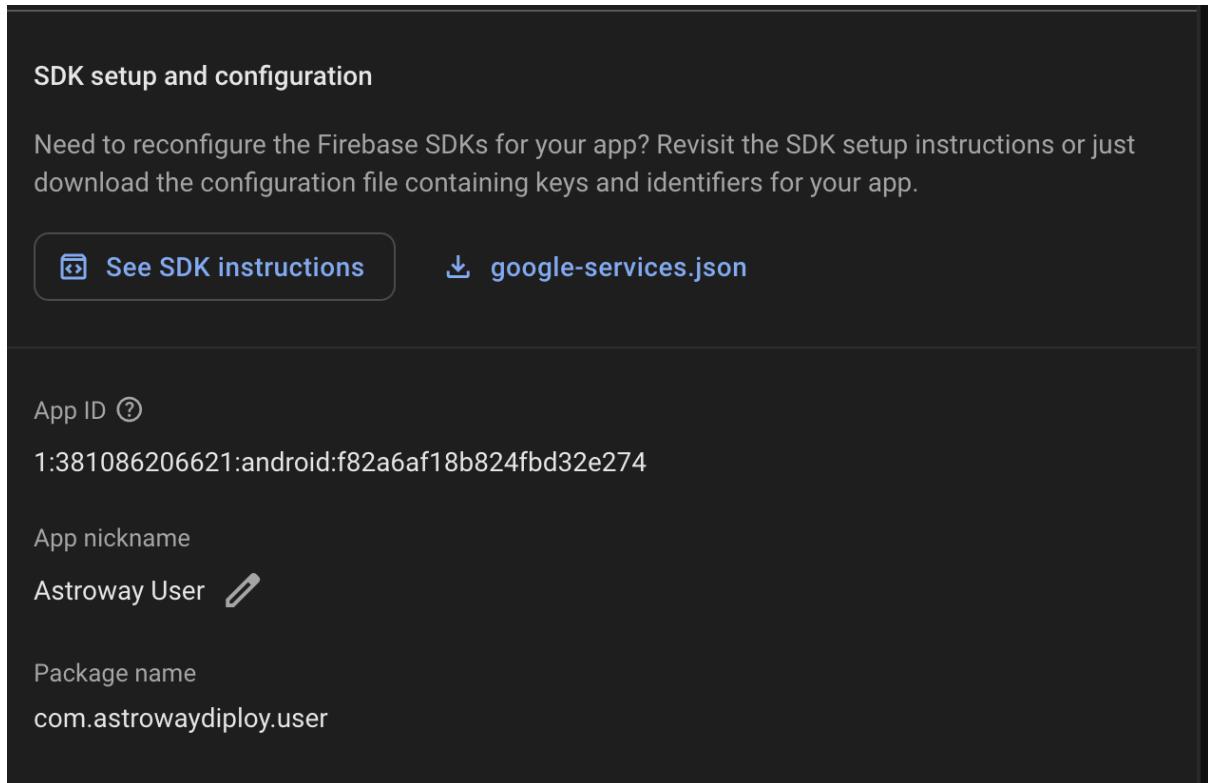


```
lib > utils > noti_service.json > ...
1  {
2    "type": "service_account",
3    "project_id": "astroway-diploy", ← project id copy this
4    "private_key_id": "████████████████████████████████████████",
5    "private_key": "-----BEGIN PRIVATE KEY-----\nMIIEvQIBADANBgkqhkiG9w0BAQEFAASCBKcwggSjAgEAAo",
6    "client_email": "████████████████@astroway-diploy.firebaseio.gserviceaccount.com",
7    "client_id": "████████████████425066312",
8    "auth_uri": "https://accounts.google.com/o/oauth2/auth",
9    "token_uri": "https://oauth2.googleapis.com/token",
10   "auth_provider_x509_cert_url": "https://www.googleapis.com/oauth2/v1/certs",
11   "client_x509_cert_url": "https://www.googleapis.com/robot/v1/metadata/x509.firebaseio-adminsdk-universe_domain": "googleapis.com"
12
13
14 }
```

## Download the google-services.json file and replace it in code

You will get the google-services.json file from **project setting in firebase>general and on your app section**

Url:- <https://console.firebaseio.google.com/project>



**Note:- This image is only for reference**

And you need to replace it from **google-services.json** file in your code

Path:- **project>android>app>google-services.json** file

## Changes in firebase\_options.dart file

Now you need to make changes in firebase\_options.dart file

File path:- **project>lib>firebase\_options.dart**

For this file you can get data from the noti\_service.json and google-services.json

A screenshot of the Android Studio interface showing the firebase\_options.dart file. The file contains Dart code defining two static const variables: android and ios. The android variable is initialized with FirebaseOptions, which includes fields for apiKey, authDomain, projectId, storageBucket, messagingSenderId, appId, and measurementId. The ios variable is also initialized with FirebaseOptions, including fields for apikey, appId, messagingSenderId, projectId, storageBucket, iosBundleId, and measurementId. All field values are redacted with [REDACTED]. The code editor shows lines 15 through 41.

Note:- This image is only for reference

## Change application version and app launcher icon

Go to `pubspec.yaml` Update the `version:A.B.C+X` as shown in image below.

**Note :-** Make sure execute `flutter packages get`, `flutter build` or `flutter run` after changing the version.

The screenshot shows the VS Code interface with the pubspec.yaml file open. The file contains the following configuration:

```
name: astrowaypartner
description: A new Flutter project.

# The following line prevents the package from being accidentally published to
# pub.dev using `flutter pub publish`. This is preferred for private packages.
publish_to: "none" # Remove this line if you wish to publish to pub.dev

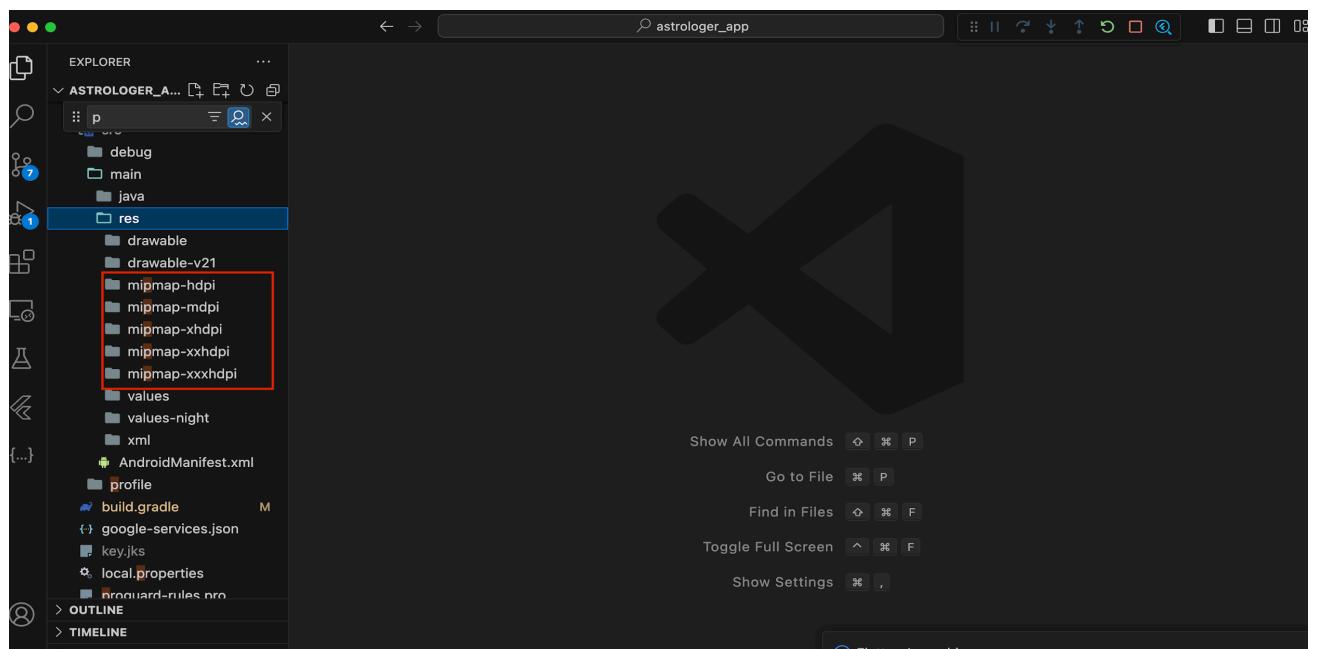
version: 1.0.0+1

environment:
  sdk: ">=2.19.0 <3.0.0"
dependencies:
  flutter:
    sdk: flutter
  #UI DESIGN
  cupertino_icons: ^1.0.6
  intl_phone_field: ^3.2.0
  flutter_otp_text_field: ^1.1.1
  percent_indicator: ^4.2.3
  font_awesome_flutter: ^10.6.0
```

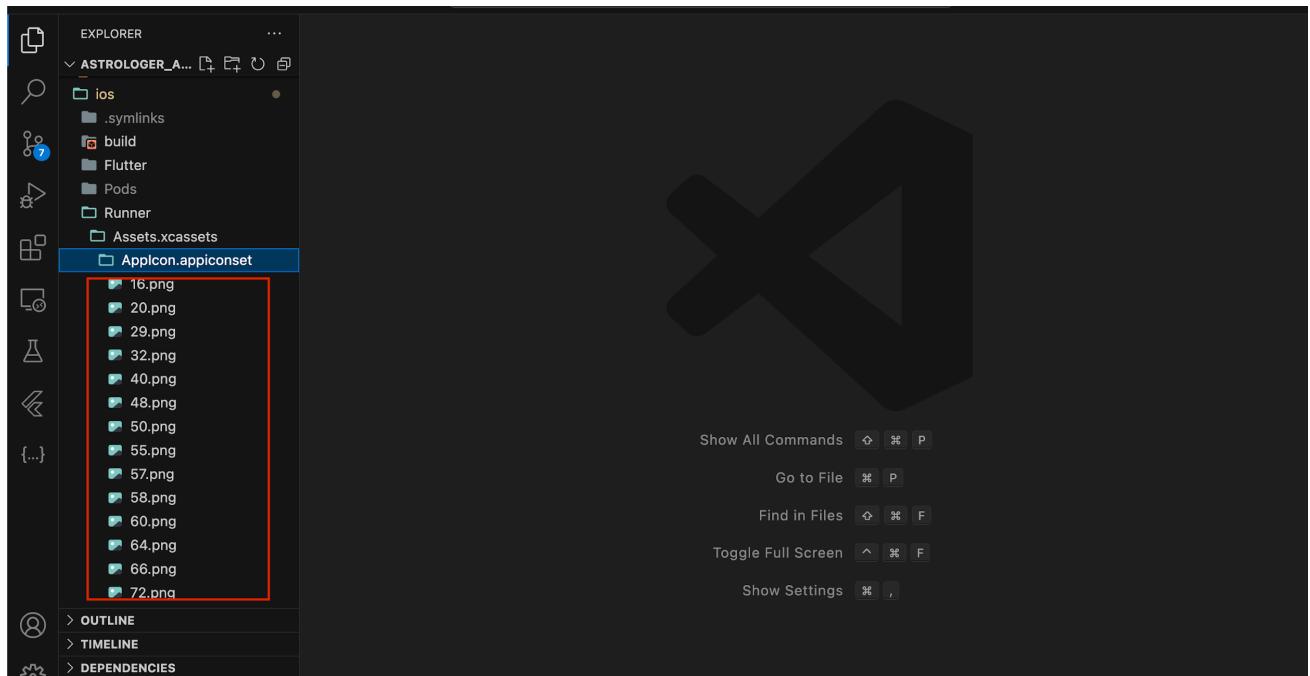
The 'version' field is highlighted with a red rectangle.

## Change App Logo

1. For Android, open android > app > src > main > res and add here your logo according to device screen size



2. For IOS open, ios > Runner > Assets.xcassets > AppIcon.appiconset here and add your logo according to different size.



## Set app's default language

The App's default language is English, but you can easily modify it. To do so, navigate to the `lib/main.dart` file and locate the section displaying the language codes, as illustrated in the image below.

You can obtain your desired language code from the following link:

<https://developers.google.com/admin-sdk/directory/v1/languages>.

Just ensure that the language code you select is available in the aforementioned link for proper functionality.

```
runApp(  
    EasyLocalization(  
        supportedLocales: const [  
            Locale('en', 'US'),  
            Locale('hi', 'IN'),  
        ],  
        path: 'assets/translations',  
        fallbackLocale: const Locale('en', 'US'), //Fallback language  
        startLocale: const Locale('en', 'US'), //Default language set  
        child: const MyApp(),  
    ),  
);  
}
```

The screenshot shows the VS Code interface with the following details:

- EXPLORER** view: Shows the project structure. A red box highlights the `translations` folder under `ASTROLOGER_A...`, which contains multiple language files: `bn-IN.json`, `en-US.json` (selected), `es-ES.json`, `gu-IN.json`, `hi-IN.json`, `kn-IN.json`, `ml-IN.json`, `mr-IN.json`, `sa-IN.json`, `ta-IN.json`, and `te-IN.json`.
- CODE** view: Displays the content of `en-US.json`. The JSON file defines various terms related to astrology and compatibility, such as Place of Birth, Kundli Matching, Open Kundli, New Matching, Match Horoscope, Compatibility Score, Share, Details, Compatibility (Varna), Varna, Bhaukti, Mental Compatibility (Maitri), Maitri, Health (Nadi), Nadi, Dominance (Vashya), Vashya, Temperament (Gana), Gana, Destiny (Tara), Tara, Physical compatibility (Yoni), Yoni, Manglik Report, and Non Manglik.

```
1 {  
2   "Place of Birth": "Place of Birth",  
3   "Kundli Matching": "Kundli Matching",  
4   "Open Kundli": "Open Kundli",  
5   "New Matching": "New Matching",  
6   "Match Horoscope": "Match Horoscope",  
7   "Compatibility Score": "Compatibility Score",  
8   "Share": "Share",  
9   "Details": "Details",  
10  "Compatibility (Varna)": "Compatibility (Varna)",  
11  "Varna refers to the mental compatibility of the two persons involved. It holds nominal effect in the mat",  
12  "Love (Bhakut)": "Love (Bhakut)",  
13  "Bhaukti is related to the couple's joys and sorrows together and assesses the wealth and health after the",  
14  "Mental Compatibility (Maitri)": "Mental Compatibility (Maitri)",  
15  "Maitri assesses the mental compatibility and mutual love between the partners to be married.": "Maitri a",  
16  "Health (Nadi)": "Health (Nadi)",  
17  "Nadi is related to the health compatibility of the couple. Matters of childbirth and progeny are also de",  
18  "Dominance (Vashya)": "Dominance (Vashya)",  
19  "Vashya indicates the bride and the groom's tendency to dominate or influence each other in marriage.": "",  
20  "Temperament (Gana)": "Temperament (Gana)",  
21  "Gana is the indicator of the behavior, character, and temperament of the potential bride and groom toward",  
22  "Destiny (Tara)": "Destiny (Tara)",  
23  "Tara is the indicator of the birth star compatibility of the bride and the groom. It also indicates the",  
24  "Physical compatibility (Yoni)": "Physical compatibility (Yoni)",  
25  "Yoni is the indicator of the sexual or physical compatibility between the bride and the groom in question",  
26  "Manglik Report": "Manglik Report",  
27  "Non Manglik": "Non Manglik",  
28  
29  
30}
```

If you are adding a new language to the application , make sure to create a separate file inside `assets/translations/` with the appropriate language code (e.g., `en-US.json` for English with the United States country code). if your desired language is not available. Copy all labels from `en-US.json` and translate the values accordingly.

And add Your language code in main.dart file

```
lib > main.dart > main
183 void main() async {
184   WidgetsFlutterBinding.ensureInitialized();
185   await Firebase.initializeApp();
186   HttpOverrides.global = MyHttpOverrides();
187
188   runApp(
189     EasyLocalization(
190       supportedLocales: const [
191         Locale('en', 'US'),
192         Locale('hi', 'IN'),
193         Locale('bn', 'IN'),
194         Locale('es', 'ES'),
195         Locale('gu', 'IN'),
196         Locale('kn', 'IN'),
197         Locale('ml', 'IN'),
198         Locale('mr', 'IN'), //marathi
199         Locale('sa', 'IN'),
200         Locale('ta', 'IN'),
201         Locale('te', 'IN') ←
202       ],
203       path: 'assets/translations',
204       fallbackLocale: const Locale('en', 'US'),
205       startLocale: const Locale('en', 'US'),
206       child: const MyApp(),
207     ), // EasyLocalization
208   );
209
210   class MyApp extends StatelessWidget {
211     const MyApp({super.key});
212
213     @override
214     Widget build(BuildContext context) {
215       return MaterialApp(
216         title: 'Astrologer App',
217         theme: ThemeData(
218           primarySwatch: Colors.indigo,
219           scaffoldBackgroundColor: Colors.white,
220           colorScheme: ColorScheme.fromSwatch()
221             .copyWith(secondary: Colors.indigo),
222           appBarTheme: AppBarTheme(
223             titleTextStyle: TextStyle(
224               color: Colors.indigo,
225               fontWeight: FontWeight.bold,
226             ),
227             iconTheme: IconTheme(
228               color: Colors.indigo,
229             ),
230             centerTitle: true,
231             elevation: 0,
232           ),
233           bottomSheetTheme: BottomSheetThemeData(
234             color: Colors.indigo,
235             shape: RoundedRectangleBorder(
236               borderRadius: BorderRadius.circular(10),
237             ),
238             builder: (context, scrollController) {
239               return Container(
240                 padding: EdgeInsets.all(10),
241                 child: Column(
242                   children: [
243                     Text('Astrologer App'),
244                     Text('Version 1.0'),
245                     Text('Developed by Astrologer'),
246                     Text('Copyright © 2023'),
247                     Text('All rights reserved.'),
```

## Update Homescreen.dart with new language code case 11

```
lib > views > HomeScreen > home_screen.dart > _HomeScreenState > build
61   HomeScreen>
62   {
63     ...
64     if (newLocale == null) {
65       return Text('Language Not Found');
66     }
67     context.setLocale(newLocale);
68     Get.updateLocale(newLocale);
69     refreshIt(homeController);
70   }
71
72   switch (language) {
73     case 'en':
74       var newLocale = const Locale('en', 'US'); //ENGLISH
75       context.setLocale(newLocale);
76       Get.updateLocale(newLocale);
77       refreshIt(homeController);
78       break;
79
80     case 'hi':
81       var newLocale = const Locale('hi', 'IN'); //HINDI
82       context.setLocale(newLocale);
83       Get.updateLocale(newLocale);
84       refreshIt(homeController);
85       break;
86
87     case 'bn':
88       var newLocale = const Locale('bn', 'IN'); //BENGALI
89       context.setLocale(newLocale);
90       Get.updateLocale(newLocale);
91       refreshIt(homeController);
92       break;
93
94     case 'es':
95       var newLocale = const Locale('es', 'ES'); //SPANISH
96       context.setLocale(newLocale);
97       Get.updateLocale(newLocale);
98       refreshIt(homeController);
99       break;
100
101    case 10: ←
102       var newLocale = const Locale('te', 'IN'); //TELGU
103       context.setLocale(newLocale);
104       Get.updateLocale(newLocale);
105       refreshIt(homeController);
106       break;
107
108    default:
109      return Text('Language Not Found');
110   }
111
112   return Scaffold(
113     appBar: AppBar(
114       title: Text('Astrologer App'),
115       centerTitle: true,
```

Now Update Language.dart file add new language in list

```
language.dart M ...
lib > models > language.dart > Language > Language.fromJson
1  class Language {
14    factory Language.fromJson(Map<String, dynamic> json) => Language(
18      subTitle: json['language_sign'] ?? "",
19    );
20
21    Map<String, dynamic> toJson() => {
22      "id": id,
23      "languageName": title,
24      "languageCode": lanCode,
25      "language_sign": subTitle,
26    };
27  }
28
29  List<Language> staticLanguageList = [
30    Language(title: 'English', lanCode: 'en', subTitle: ''),
31    Language(title: 'Gujarati', lanCode: 'gu', subTitle: 'ગુજરાતી'),
32    Language(title: 'Hindi', lanCode: 'hi', subTitle: 'હિન્ડી'),
33    Language(title: 'Spanish', lanCode: 'es', subTitle: 'Española'),
34    Language(title: 'Marathi', lanCode: 'mr', subTitle: 'મરાಠી'),
35    Language(title: 'Sanskrit', lanCode: 'sa', subTitle: 'સંસ્કૃત'),
36    Language(title: 'Bengali', lanCode: 'bn', subTitle: 'বাঙালি'),
37    Language(title: 'Kannada', lanCode: 'kn', subTitle: 'ಕನ್ನಡ'),
38    Language(title: 'Malayalam', lanCode: 'ml', subTitle: 'മലയാളം'),
39    Language(title: 'Tamil', lanCode: 'ta', subTitle: 'தமிழ்'),
40    Language(title: 'Telugu', lanCode: 'te', subTitle: 'తెలుగు'), ←
41  ];
42
```

## Change App Base Url

You need to update keys such as `CONTACT_SUPPORT` , `IMAGE_BASE_URL`, `PDF_BASE_URL`, `GOOGLE_PLACES_API_KEY`, `SERVER_NOTIFICATION_KEY`, or `API_URL`, simply open the `config.dart` file and paste your keys there.

```
lib > utils > config.dart > appParameters
1 // ignore_for_file: non_constant_identifier_names, dangling_library_doc_comments
2
3 /**
4  * =====
5  * ***** GOOGLE PLACES KEY AND NOTIFICATION SERVER KEY *****
6  */
7 String GOOGLE_PLACES_API_KEY = 'VT8NWYqyfIfDI'; ←
8 /**
9  * =====
10 /**
11  * ***** BASE URL FOR API *****
12 /**
13 String contactsupportEmail = "support@gmail.com";
14 String imgBaseurl = "https://astroway.deploy.in/";
15 String pdfBaseurl = "https://astroway.deploy.in/public";
16 String appMode = "LIVE";
17
18 Map<String, dynamic> appParameters = {
19   "LIVE": {
20     "apiUrl": "https://astroway.deploy.in/api/",
21   },
22   "DEV": {
23     "apiUrl": "https://astroway.deploy.in/api/",
24   }
25 };
26
```

## Change Google Places API Key

To change the API key for Google Places, please follow these steps:

1. Visit the following link to create an API key: [places key generated](#)
2. Once you have obtained the API key, paste it into the **config.dart** file above the existing configuration.

## Change Agora Credentials

For changing the agora credentials you need to change to channel name inside the global.dart file

**lib>utils>global.dart**

The screenshot shows an IDE interface with the project 'ASTROLOGER\_APP' selected in the Explorer panel. The current file is 'global.dart'. A red box highlights the following code block, which defines various Agora API variables:

```
136 IosDeviceInfo? iosInfo;
137 DeviceInfoPlugin deviceInfo = DeviceInfoPlugin();
138 String? deviceId;
139 String? fcmToken;
140 String? deviceLocation;
141 String? deviceManufacturer;
142 String? deviceModel;
143 var appVersion;
144 int? currentUserId;
145
146 String agoraAppId = "832f8b58443247a2b8b74677198bbc82";
147 String agoraAppCertificate = "36a8fdae33b447e8a928e108a7f36bd9";
148
149 String agoraChannelName = "";
150 String agoraToken = "";
151 String agoraLiveToken = "";
152 String channelName = "astrowayGuruLive";
153 String agoraLiveChannelName = "";
154 String liveChannelName = "astrowayGuruLive";
155 String agoraChatToken = "";
156 String agoraChatUserId = "astrowayGuruLive";
157 String chatChannelName = "astrowayGuruLive";
158 String encodedString = "&&";
159 String appName = "";
160 var nativeAndroidPlatform = const MethodChannel('nativeAndroid');
161 SystemFlagNameList systemFlagNameList = SystemFlagNameList();
162
163 //Get app version
164 String getAppVersion() {
165   PackageInfo.fromPlatform().then((PackageInfo packageInfo) {
166     appVersion = packageInfo.version;
167   });
}
```

## Change app colors

can change colors of the app from `lib/theme/nativeTheme.dart`.

The screenshot shows the Visual Studio Code interface with the following details:

- Explorer View:** Shows the project structure for "ASTROLOGER\_APP". The "theme" folder is selected and highlighted with a red box. It contains "nativeTheme.dart" and "themeService.dart".
- Editor View:** Displays the file "nativeTheme.dart". A specific line of code, `primaryColor: const Color(0xfffffc10),`, is highlighted with a red box.
- Side Bar:** Shows various icons for file operations like Find, Replace, and Save.

```
nativeTheme.dart M X
lib > theme > nativeTheme.dart > Themes > light
21
22 class Themes {
23 static final light = ThemeData.light().copyWith(
24   appBarTheme: AppBarTheme(
25     systemOverlayStyle:
26       const SystemUiOverlayStyle(statusBarColor: Colors.transparent),
27     backgroundColor: COLOR(.primaryColor,
28     foregroundColor: COLOR(.blackColor,
29     elevation: 0.5,
30   ), // AppBarTheme
31
32   primaryColor: const Color(0xfffffc10),
33
34   textTheme: const TextTheme(
35     displayLarge: TextStyle(color: Colors.black),
36     displayMedium: TextStyle(color: Colors.black),
37     displaySmall: TextStyle(color: Colors.black, fontSize: 14),
38     headlineMedium: TextStyle(color: Colors.black),
39     headlineSmall: TextStyle(color: Colors.black), //
40     titleLarge: TextStyle(color: Colors.black), //
41     bodySmall: TextStyle(color: Colors.black),
42     titleMedium: TextStyle(color: Colors.black, fontSize: 14),
43     titleSmall: TextStyle(color: Colors.black),
44     bodyLarge: TextStyle(color: Colors.black),
45     bodyMedium: TextStyle(color: Colors.black),
46   ), // TextTheme
47   primaryTextTheme: TextTheme(
48     displayLarge: const TextStyle(
49       color: Colors.black,
50       fontWeight: FontWeight.w300,
51       fontSize: 19), // APPBAR // TextStyle
52     displayMedium: const TextStyle(
```

# Path of the Screens

- loginScreen:- lib>views>Authentication>login\_screen.dart
  - signupScreen:- lib>views>Authentication>signup\_screen.dart
  - successRegistration:- lib>views>Authentication>success\_registration\_screen.dart
  - login\_otp\_screen:- lib>views>Authentication>OtpScreen> login\_otp\_screen.dart
  - signup\_otp\_screen:- lib>views>Authentication>OtpScreen> signup\_otp\_screen.dart
  - splashScreen:- lib>views>splash>splash\_screen.dart
  - AddAssistanScreen:- lib>views>HomeScreen>Assistant>  
add\_or\_edit\_assistant\_screen.dart
  - assistantChatRequestScreen:- lib>views>HomeScreen> Assistant>  
assistant\_chat\_request\_screen.dart
  - assistantChatScreen:- lib>views>HomeScreen> Assistant>  
assistant\_chat\_screen.dart
  - assistantScreen:- lib>views>HomeScreen> Assistant> assistant\_screen.dart
  - AcceptCallScreen:- lib>views>HomeScreen> call> accept\_call\_screen.dart
  - callAvailabilityScreen:- lib>views>HomeScreen> call> callAvailabilityScreen.dart
  - app\_review\_Screen:-lib/views/HomeScreen/Drawer/AppReview/app\_review\_screen.dart
  - privacy\_policy\_screen:-lib/views/HomeScreen/Drawer/Setting/privacy\_policy\_screen.dart

- terms\_and\_condition\_screen:-lib/views/HomeScreen/Drawer/Setting/term\_and\_condition\_screen.dart
- setting\_list\_screen:-lib/views/HomeScreen/Drawer/Setting/setting\_list\_screen.dart
- add\_amout\_screen:-lib/views/HomeScreen/Drawer/Wallet/add\_amount\_screen.dart
- wallet\_screen:-lib/views/HomeScreen/Drawer/Wallet/Wallet\_screen.dart
- customer\_review\_Screen:-  
lib/views/HomeScreen/Drawer/customer\_review\_screen.dart
- drawer\_screen:- lib/views/HomeScreen/Drawer/drawer\_screen.dart
- astrology\_blog\_detil\_screen:-lib/views/HomeScreen/FloatingButton/AstroBlog/astrology\_blog\_detil\_screen.dart
- astrology\_blog\_screen:-lib/views/HomeScreen/FloatingButton/AstroBlog/astrology\_blog\_screen.dart
- dailyHoroScopeDetailScreen:-lib/views/HomeScreen/FloatingButton/DailyHoroscope/dailyHoroScopeDetailScreen.dart
- dailyHoroscopeVedic:-  
lib/views/HomeScreen/FloatingButton/DailyHoroscope/dailyHoroscopeVedic.dart
- chatAvailabilityScreen:-lib/views/HomeScreen/FloatingButton/FreeKundli/Tabs/report  
Tabs/chatAvailabilityScreen.dart
- basicKundliScreen:-lib/views/HomeScreen/FloatingButton/FreeKundli/Tabs/basicKun  
dliScreen.dart
- Pdf\_Screen:-lib/views/HomeScreen/FloatingButton/FreeKundli/Tabs/Pdf\_Screen.dart
- createNewKundli:-lib/views/HomeScreen/FloatingButton/FreeKundli/createNewKundli  
.dart
- editKundliScreen:-lib/views/HomeScreen/FloatingButton/FreeKundli/editKundliScreen  
.dart
- kundliDetailsScreen:-lib/views/HomeScreen/FloatingButton/FreeKundli/kundliDetails  
Screen.dart
- kundliScreen:-lib/views/HomeScreen/FloatingButton/FreeKundli/kundliScreen.dart
- New\_matching\_screen:-  
lib/views/HomeScreen/FloatingButton/KundliMatching/Tabs/new\_matching\_screen.d  
art
- open\_kundli\_screen:-lib/views/HomeScreen/FloatingButton/KundliMatching/Tabs/ope  
n\_kundli\_screen.dart
- kundli\_matching\_screen:-lib/views/HomeScreen/FloatingButton/KundliMatching/kund  
li\_matching\_screen.dart
- place\_of\_birth\_screen:-lib/views/HomeScreen/FloatingButton/KundliMatching/place\_  
of\_birth\_screen.dart
- north\_kundli\_match\_result\_screen:-lib/views/HomeScreen/FloatingButton/KundliMatc  
hing/north\_kundli\_match\_result\_screen.dart
- south\_kundali\_matching\_screen:-lib/views/HomeScreen/FloatingButton/KundliMatchi  
ng/south\_kundali\_matching\_screen.dart
- onetooneVideo:-lib/views/HomeScreen/live/onetooone\_video/onetoooneVideo.dart
- live\_screen:- lib/views/HomeScreen/live/live\_screen.dart
- assignment\_detail\_screen:-  
lib/views/HomeScreen/Profile/ProfileDetailScreen/assignment\_detail\_screen.dart
- availability\_Detail\_screen:-  
lib/views/HomeScreen/Profile/ProfileDetailScreen/availability\_Detail\_scree.dart

- other\_detail\_screen:-  
lib/views/HomeScreen/Profile/ProfileDetailScreen/other\_detail\_screen.dart
- personal\_detail\_screen :-  
lib/views/HomeScreen/Profile/ProfileDetailScreen/personal\_detail\_screen.dart
- skill\_detail\_screen :-  
lib/views/HomeScreen/Profile/ProfileDetailScreen/skill\_detail\_screen.dart
- edit\_profile\_screen:- lib/views/HomeScreen/Profile/edit\_profile\_screen.dart
- follower\_list\_screen:- lib/views/HomeScreen/Profile/follower\_list\_screen.dart
- Profile\_screen:- lib/views/HomeScreen/Profile/profile\_screen.dart
- report\_detail\_screen :-  
lib/views/HomeScreen/Report\_Module/report\_detail\_screen.dart
- report\_history\_details\_screen: -  
lib/views/HomeScreen/Report\_Module/report\_history\_details\_screen.dart
- Report\_history\_list\_screen:-  
lib/views/HomeScreen/Report\_Module/report\_history\_list\_screen.dart
- report\_request\_screen:-  
lib/views/HomeScreen/Report\_Module/report\_request\_screen.dart
- view\_report\_pdf\_scree:-  
lib/views/HomeScreen/Report\_Module/view\_report\_pdf\_scree.dart
- call\_detail\_screen:- lib/views/HomeScreen/call\_detail\_screen.dart
- chat\_screen:- lib/views/HomeScreen/chat\_screen.dart
- home\_screen:- lib/views/HomeScreen/home\_screen.dart
- notification\_screen:- lib/views/HomeScreen/notification\_screen.dart
- search\_blog\_screen:- lib/views/HomeScreen/search\_blog\_screen.dart

## Contact Support

Delighted to serve and support you! Contact our support team. We're available Mon-Fri, 9:00 am - 6:00 pm IST (GMT +5.30),

India - Asia. Expect prompt responses within 24 hours via comments, forum, or email.

---

## Reach Us

Unlock the potential of diploy, your go-to hub for top-notch Flutter app development and web applications. Reach out to us today for quality solutions and exciting freelance opportunities.

Visit us at [diploy.in](https://diploy.in) or drop an email to [start@diploy.in](mailto:start@diploy.in)

Thank you for considering us!

