

Thank you for your purchase! To ensure seamless after-sales support, please activate your product using the link below:

[Activate Now](#)

Your activation is essential for us to provide you with the best assistance. We appreciate your cooperation.

Technology Used

- **Flutter** with Dart Language for Mobile App
- **DB Browser for SQLite** for Database

Please follow the below steps to set up the App.

1. Setup Prerequisites (If not available)

- a. Install and set up Flutter from [this link](#)
- b. Install DB Browser for SQLite from [this link](#)

2. Setup Mobile App (Technology Flutter)

- a. Initial steps to set up and run mobile app
 - i. Open the **App** folder in the VSCode
 - ii. Run the following commands in the VSCode Terminal

```
flutter clean  
flutter pub get
```

- iii. Additional steps to set up for iOS (You can skip these steps if you don't want to set up for iOS)
 1. In the VSCode terminal, go to the ios directory
(using the command **cd ios**)
 2. Run the following command to install pods

```
pod install
```

- iv. Connect your Android or iOS device with your machine

(To run on an Apple device, you must have an Apple computer)

- v. Run the following command to run on an Android or iOS device

```
flutter run
```

- vi. To upgrade the Flutter version run the following command in the VSCode Terminal

(Only if your Flutter version is lower than mentioned in this document)

```
flutter upgrade
```

b. Changes for About screen

You have to change your about us screen URLs, for that go to the file located at **lib\screens\aboutScreen.dart**

1. Search **<YOUR EMAIL>** and replace it with your email
2. Search **<YOUR WEBSITE>** and replace it with your website
3. Search **<YOUR FB PAGE URL>** and replace it with your facebook page URL
4. Search **<YOUR INSTAGRAM PAGE URL>** and replace it with your instagram page URL
5. Search **<YOUR LINKEDIN PAGE URL>** and replace it with your linkedin page URL
6. Search **<YOUR TWITTER PAGE URL>** and replace it with your twitter page URL
7. Search **<YOUR WEB SITE URL>** and replace it with your website URL

c. Changes for App store and Play store

You have to change your app store and play store URLs, for that go to the file located at **lib\utils\global.dart**

1. Search **<PLAY STORE URL>** and replace it with your play store URL
2. Search **<APP STORE URL>** and replace it with your app store URL

d. Changes for Privacy policy

You have to change your privacy policy URL, for that go to the file located at **lib\screens\privacyPolicyScreen.dart**

1. Search **<PRIVACY POLICY URL>** and replace it with your privacy policy URL

e. Change Package Name/Bundle ID

An app's package name is a unique identifier that is automatically created when you create an app. The term used for iOS apps is "**bundle ID**" and for Android apps, it is "**package name**".

i. Set package name for Android App

1. Change the package name in the file located at **android/app/src/main/AndroidManifest.xml**
2. Change the package name in the file located at **android/app/src/debug/AndroidManifest.xml**
3. Change Package Name in file which is located at **android/app/src/Profile/AndroidManifest.xml**
4. Change the folder structure for the below path as per your package name.

```
android\app\src\main\java\com\demo\accountingapp
```

ii. Set Bundle ID for iOS App

1. In VsCode Go to **ios/Runner/info.plist**
 2. In Xcode Change your Target Runner's General Tab
- For more information please refer to **Accounting App Setup Document v2.1.pdf** point no **5.c**.

f. Create and set Keystore file for Android

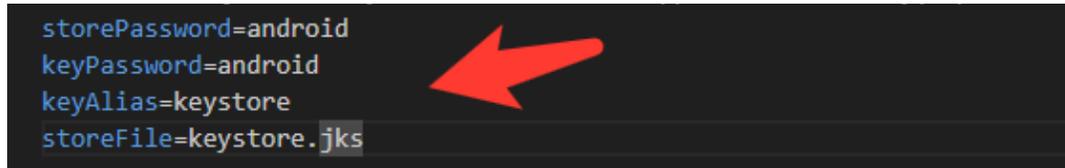
- i. Create a keystore.jks file, if not exist, use the below command in the terminal

```
keytool -genkey -v -keystore "path\keystore.jks" -storetype JKS  
-keyalg RSA -keysize 2048 -validity 10000 -alias keystore
```

- ii. Fill in all the details asked while executing the above command

- iii. Recommended. After creating your keystore.jks file, please put it in the **android/app** folder
- iv. Create a key.properties file in the **android** folder and add the details in the file as per the below screenshot.

```
storePassword=android
keyPassword=android
keyAlias=keystore
storeFile=keystore.jks
```



NOTE:

- If you have changed any default value for any of these keys (storePassword, keyPassword, keyAlias, storeFile) while creating the keystore.jks file, then please also change them to key.properties file.
- If you place your keystore.jks file somewhere else in the project than mentioned in step 5.c.iii then please change storeFile key value accordingly.
- For more details please refer to [this link](#)

g. Create a Firebase Account & Project Setup

1. To create a Firebase project go to this [Firebase console](#).
2. Set up Firebase for Your Flutter App
 - Go to your Firebase console dashboard.
 - Click on the “Add app” button (Android/iOS) and follow the instructions to register your app.
 - **For Android:**
 - Enter your Android package name(com.example.myapp) and click on “Register app”.
 - Download the ‘**google-services.json**’ file and place it inside your Flutter app’s ‘**android/app**’ directory.
 - Add the Firebase SDK to your app by following the instructions provided on the Firebase console.
 - **For iOS:**
 - Enter your iOS bundle ID and click on “Register app”.
 - Download the ‘**GoogleServices-Info.plist**’ file and add it to your Flutter app’s ‘**ios/Runner**’ directory.

- Follow the additional setup instructions provided on the Firebase console.
- If you need any help please refer to our **Accounting App Setup Document v1.4.pdf** point no **5.f, 5.g and 5.h**.

h. Change App Icon

- To change the app icon in a Flutter app for both Android and iOS platforms, you can follow these steps:s
- **For Android:**
 1. **Prepare Your App Icon:** Create your app icon in the required sizes. Android app icons typically come in various sizes (e.g., 48x48, 72x72, 96x96, 144x144, and 192x192 pixels) and densities (mdpi, hdpi, xhdpi, xxhdpi, xxxhdpi).
 2. **Replace the Default Icon:** In your Flutter project, navigate to the android/app/src/main/res directory.
 3. **Replace Icons:** Replace the default icons in the respective density folders with your new app icons. Make sure to maintain the correct folder structure and file naming conventions. For example, ic_launcher.png for the launcher icon.
 4. **Update Android Manifest:** Open android/app/src/main/AndroidManifest.xml and ensure that the android:icon attribute of the <application> tag points to the correct icon resource. It should be something like @mipmap/ic_launcher.
- **For iOS:**
 1. **Prepare Your App Icon:** Create your app icon in the required sizes. iOS app icons typically come in various sizes (e.g., 60x60, 120x120, 180x180, and so on) for different device resolutions.
 2. **Replace the Default Icon:** In your Flutter project, navigate to the ios/Runner/Assets.xcassets directory.
 3. **Replace Icons:** Replace the default icons in the Applcon.appiconset folder with your new app icons. Make sure to maintain the correct size and file format (usually PNG).
 4. **Update Info.plist:** Open ios/Runner/Info.plist and ensure that the CFBundleIconFile property refers to your app icon filename without the extension. For example, Applcon.
 5. **Build and Run:** Build and run your Flutter app on an iOS device or simulator to see the updated app icon
- If you need any help with any of the above points please refer to our **Accounting App Setup Document v1.4.pdf** point no **5.i**.

i. Build Release for Android

- i. Open Project in VS Code
- ii. In Terminal Execute the below commands

```
flutter clean
flutter pub get
flutter build apk --release
```

- iii. After making the release, to generate the release bundle Execute the below command

```
flutter build appbundle --release
```

- iv. Get the APK from the below path

Build\app\outputs\flutter-apk\app-release.apk

j. Build Release for iOS

- i. Open Project in XCode
- ii. Select **Archive** from the **Product Menu**
- iii. After successfully archiving select the **Organizer** option from the **Windows menu**
- iv. After clicking on it opens one popup for Archive, Click on the **Distribute App** Button
- v. After successfully done, you can upload this app to your Apple developer account in the TestFlight
- vi. To publish your app from TestFlight please follow [this link](#)

NOTE: While running/building the app, if you get any error **A Firebase App name “[DEFAULT]” already exists**, then add the param name when initializing the Firebase in the **lib/main.dart** file.

USEFUL LINKS

- For more information on iOS refer to [this link](#)

This document was last updated on 30 May 2024.