**Steps**

1.Change App name, bundle Id and logo

2.sign app and download aab file

3.publish app to google play store

**Step1:**

* Take reference of this link [Android | Flutter](https://docs.flutter.dev/deployment/android) -> “Add a launcher icon”
* Go to pubspec.yaml file then add [flutter\_launcher\_icons](https://pub.dev/packages/flutter_launcher_icons) package:

dev\_dependencies:

flutter\_test:

sdk: flutter

[**flutter\_launcher\_icons**](https://pub.dev/packages/flutter_launcher_icons)**: 0.14.1 (latest version)**

* Take the logo and place it inside ” icons folder”
* Now add this

dev\_dependencies:

flutter\_test:

sdk: flutter

[flutter\_launcher\_icons](https://pub.dev/packages/flutter_launcher_icons): 0.14.1

**flutter\_icons:**

**image\_path: “icons/app\_logo.png”**

**android: true**

**ios: true**

* Now go to Terminal

PS C:\Users\name\app\_name>**flutter pub get**

PS C:\Users\name\app\_name>**flutter pub run flutter\_launcher\_icons** (changes icon for android and ios)

* Now go to **AndroidManifest.xml** file to change App Name(android>app>src>main>AndroidManifest.xml)

<application

android:label=**”Change App Name”** (ex:- cjn\_app)

* Go to **build.gradle** (android>app>build.gradle)

defaultConfig {

applicationId “com.admin.cjn\_app” (ex:-com.cjn.cjnott) (change to particular domain)

* Run the code to see the logo and name would have changed

**Step 2:**

* Sign the App and use the document [Android | Flutter](https://docs.flutter.dev/deployment/android)
* Go to **Reference the keystore from the app ->** go to project-> android folder -> create a file name **key.properties** then add the below code

**storePassword=<password-from-previous-step> (fill this)**

**keyPassword=<password-from-previous-step> (fill this)**

**keyAlias=upload**

**storeFile=<keystore-file-location>**

* Now go again to flutter documentsearch for -> **Create an upload keystore**

According to the OS System place the code

**{ On macOS or Linux, use the following command:**

|  |
| --- |
| keytool -genkey -v -keystore ~/upload-keystore.jks -keyalg RSA \  -keysize 2048 -validity 10000 -alias upload |

**On Windows, use the following command in PowerShell:**

|  |
| --- |
| keytool -genkey -v -keystore $env:USERPROFILE\upload-keystore.jks `  -storetype JKS -keyalg RSA -keysize 2048 -validity 10000 `  -alias upload |

* Go to Command Prompt then Paste the above code for window -> Enter -> and then that will ask for password (the password which created before) -> **ReEnter -> type first and last name ->** then press **Enter** for all other prompts -> yes -> now we can see **upload-keystore.jks** downloaded -> see the file manually in the given project location -> then drag and drop the keystore file into the **app** folder
* Now place the location of the keystore into this file name **key.properties**

storeFile=<**keystore-file-location**> eg: storeFile=../app/upload-keystore.jks

* Go to flutter document search for **Configure signing in gradle** copy code
* Go to project -> build.gradle -> search for android

|  |
| --- |
| def keystoreProperties = new Properties()  def keystorePropertiesFile = rootProject.file('key.properties')  if (keystorePropertiesFile.exists()) {  keystoreProperties.load(new FileInputStream(keystorePropertiesFile))  } |

**Paste this above android row(align from start of page no space needed)**

* Now again go to flutter document search for -> **And replace it with the following signing configuration info: ->** copy the code

|  |
| --- |
| signingConfigs {  release {  keyAlias = keystoreProperties['keyAlias']  keyPassword = keystoreProperties['keyPassword']  storeFile = keystoreProperties['storeFile'] ? file(keystoreProperties['storeFile']) : null  storePassword = keystoreProperties['storePassword']  }  }  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  signingConfig = signingConfigs.release  }  } |

* Go to build.gradle replace the **buildTypes** code into the above code (make ready to app release)

**buildTypes{**

**release{**

**signingConfig signingConfigs.debug**

**}**

**}**

* Go to Project IDE Terminal then type

PS C:\Users\name\app\_name>**flutter build appbundle –release**

Enter -> Release app will be ready (aab file)

* Go to google play console -> <https://play.google.com/console/about/?hl=en-gb> -> Click the button “Go to Play Console” -> signin with google account -> if there is no googleaccount then click below button to create one
* Now if you have clicked login to go to Google Play Console page where you will find all the user details
* Click on **Create app -> fill the details -> privacy policies -> Click Create App**
* Now on side bar go to **Grow -> Store presence ->** Click on **Main store listing**
* **Fill the form** about the app description(short then a full description)
* Now scroll down upload the app\_logo image (or drag and drop)
* Also upload a Feature graphic image or drag and drop
* Upload the app feature screenshots (atleast 2)
* Optional (upload project video, and more images)
* On right corner three dot(**Click Save**)
* Now go to **Grow -> Store presence ->** Click on **Store settings -> Edit -> fill the option for App or Game and then category**
* Add some tags again in the same path
* Store Listing contact details -> edit -> about email address, phone number, website(https://)
* Now switch to **Release -> Production -> Click on Create new release -> click choose signin key -> click use google generated key -> App bundle(Click Upload) -> drag and drop app-release.aab -> once its done then look into Release notes which will be filled automatically then edit as per your requirement -> Click Next**
* **Now solve errors one by one**
* Go to **Policy and programmes -> App content -> click start declaration -> Add privacy policy URL(ex:- app.termly.io) -> Generate it**
* **Sign In to github and create new repository and place a file “privacy-policy” and paste all the details**
* **And now copy the repository url link -> paste it in the declaration -> Save**
* **Now declare Ads -> choose your option**
* **App access -> choose your option**
* **Content ratings -> fill email address,category -> Next -> fill form -> Save file**
* **Target audience and content -> choose -> save**
* **News apps -> choose -> save -> Covid -19 -> choose ->save ->data safety -> choose->save**
* **Fill the other options -> choose -> Save**
* **Now back to Production -> Go to Countries/regions -> select -> save -> Release overview -> save all changes done -> send changes to overview ->Start rollout**
* Take 1 or 2 days to be published -> Click All apps to see the app information