

Build and release an Android app

(<https://docs.flutter.dev/deployment/android>)

1. Adding a Launcher Icon

- Generate Icon- <https://romannurik.github.io/AndroidAssetStudio/icons-launcher.html>
- Download generated Icon
- Create assets folder and paste the generated icon and change the name if required.
- Add flutter_launcher_icons plugin from pub.dev - https://pub.dev/packages/flutter_launcher_icons
- Copy and paste config in dev dependencies in pubspec.yaml

```
flutter_launcher_icons:  
  android: "launcher_icon"  
  ios: true  
  image_path: "assets/icon/icon.png"  
  min_sdk_android: 21 # android min sdk min:16,  
    default 21
```

- Run the package,
- Flutter pub get
- Flutter pub run flutter_launcher_icons

2. App name and BundleID:

- Get rename package from pub.dev - <https://pub.dev/packages/rename>
- Activate rename using command "flutter pub global activate rename"
- flutter pub global run rename setAppName --targets ios,android --value "Management Dashboard" to rename the app
- Change bundleid using flutter pub global run rename setBundleId --targets android --value com.itcsg.management_dashboard '

3. Signing the app:

- Create new file in the \android\key.properties path
- Past keystore credentials
storePrassword=Welcome@123
keyPassword=Welcome@123
keyAlias=upload
storeFile=<keystore-file-location>

- create keystore

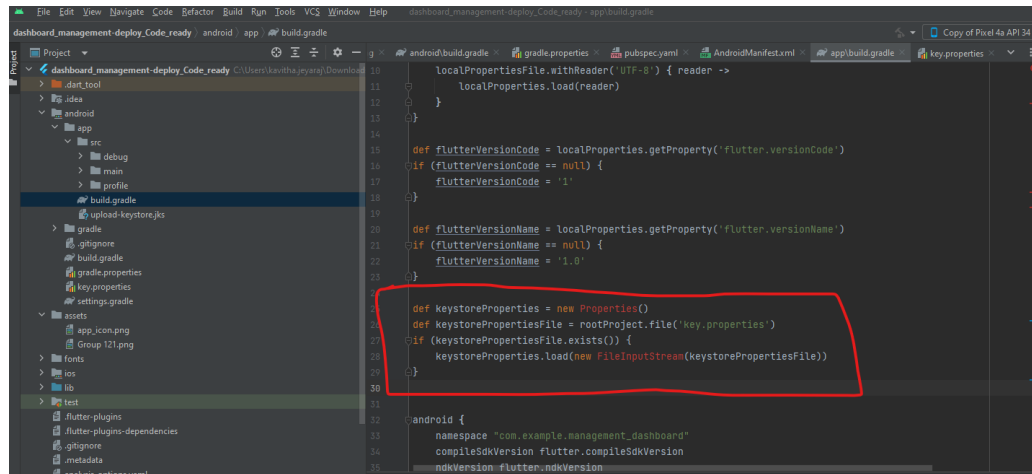
```
keytool -genkey -v -keystore ~/upload-keystore.jks -keyalg RSA \
        -keysize 2048 -validity 10000 -alias upload
```

4. Configure signing in gradle

- Paste config details in [project]/android/app/build.gradle before android block,

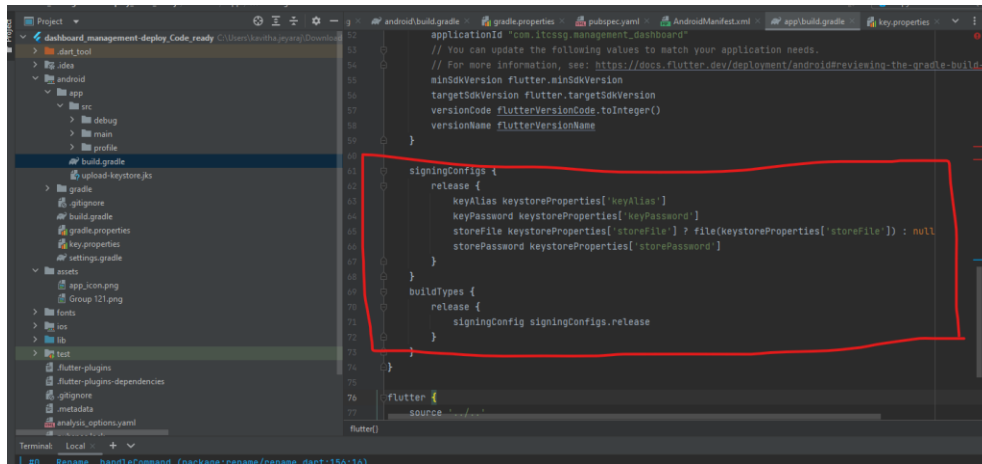
```
def keystoreProperties = new Properties()
def keystorePropertiesFile = rootProject.file('key.properties')
if (keystorePropertiesFile.exists()) {
    keystoreProperties.load(new FileInputStream(keystorePropertiesFile))
}

android {
    ...
}
```



- Change the signingconfig

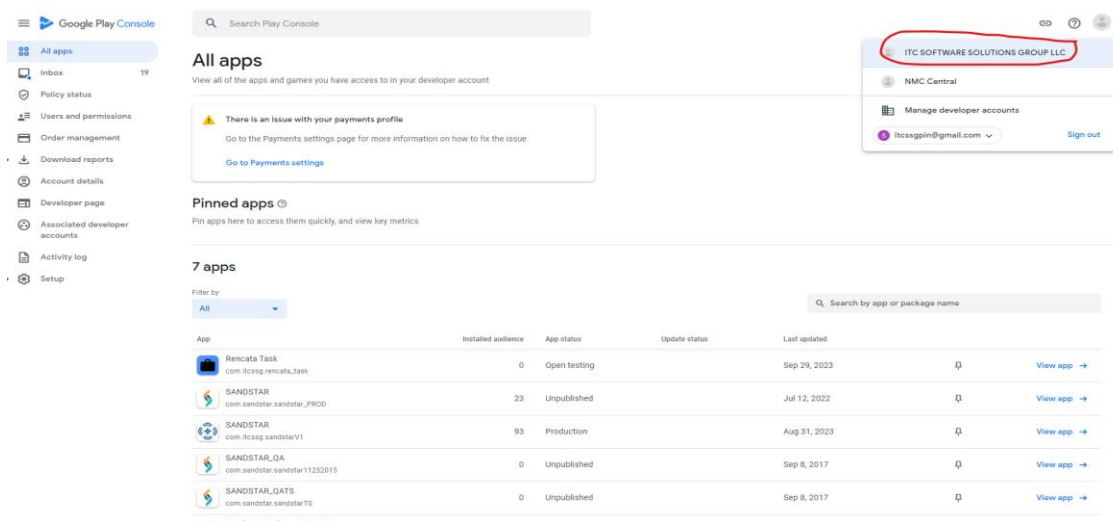
```
signingConfigs {
    release {
        keyAlias keystoreProperties['keyAlias']
        keyPassword keystoreProperties['keyPassword']
        storeFile keystoreProperties['storeFile'] ?
file(keystoreProperties['storeFile']) : null
        storePassword keystoreProperties['storePassword']
    }
}
buildTypes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



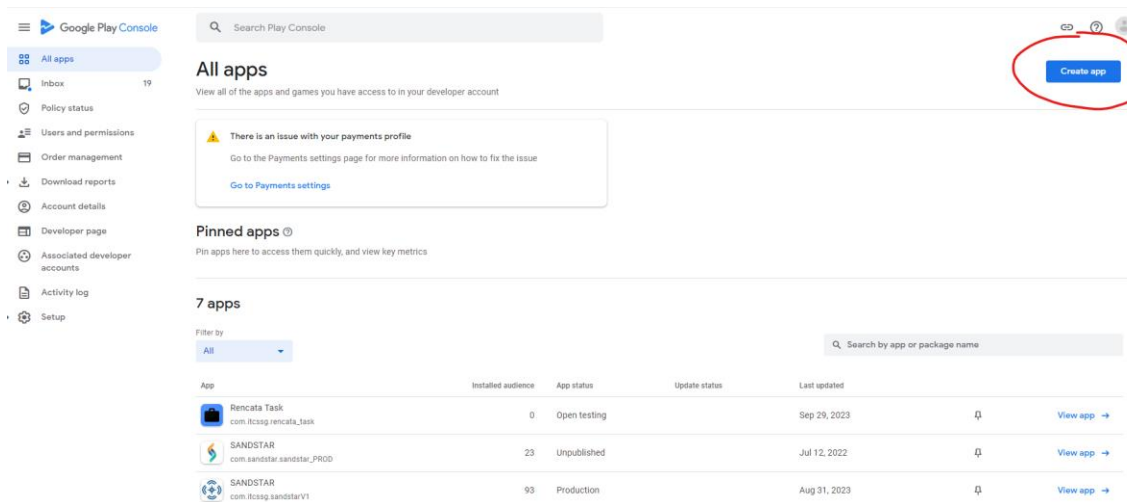
5. Build App Bundle

- Run the 'flutter build appbundle' to create appbundle
- Run the 'flutter build apk' for apk release

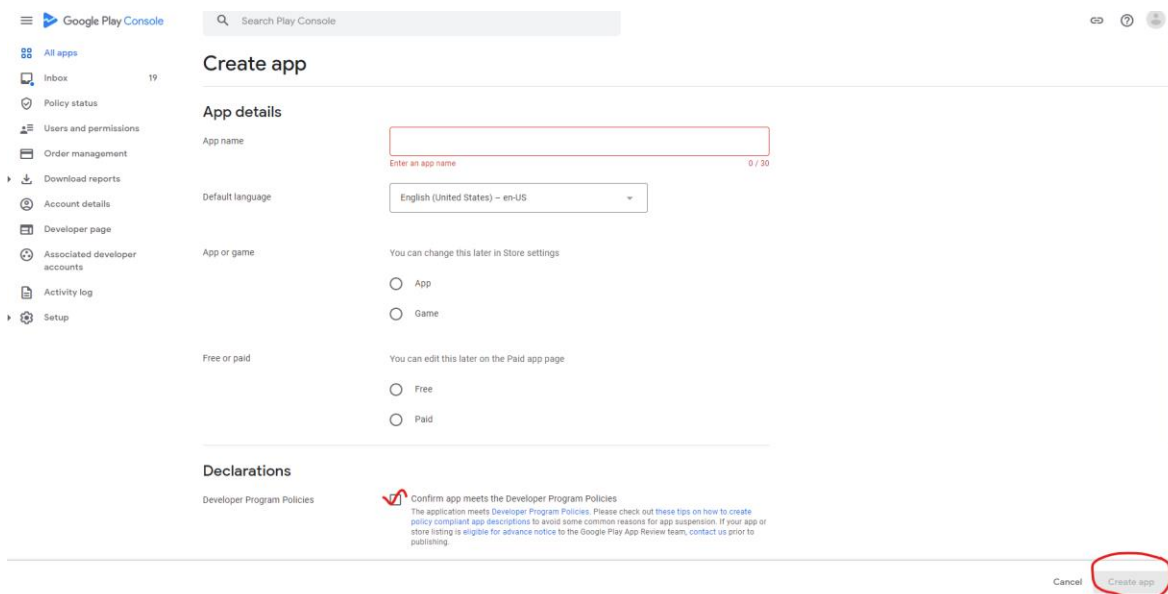
6. Log into the Google play console as ITC SOFTWARE SOLUTION GROUP LLC account



7. click on create App to create new app



8. Fill all the required field and check the policy checkbox and click on create app as like below img



Google Play Console

Search Play Console

Create app

App details

App name: test app (8 / 30)

Default language: English (United States) - en-US

App or game: You can change this later in Store settings.
☒ App
☐ Game

Free or paid: You can edit this later on the Paid app page.
☒ Free
☐ Paid

You can edit this until you publish your app. Once you've published, you can't change a free app to paid.

Declarations

Developer Program Policies: ☒ Confirm app meets the Developer Program Policies

Cancel **Create app**

9. Finish all the dashboard task as per your app required

Google Play Console

Search Play Console

test app

Dashboard

Start testing now

Release your app early for internal testing without review
 Share your app with up to 100 internal testers to identify issues and get early feedback

Hide tasks ^

- Select testers > ✓

CREATE AND ROLL OUT A RELEASE

- Create a new release > ✓
- Preview and confirm the release ✓

Set up your app

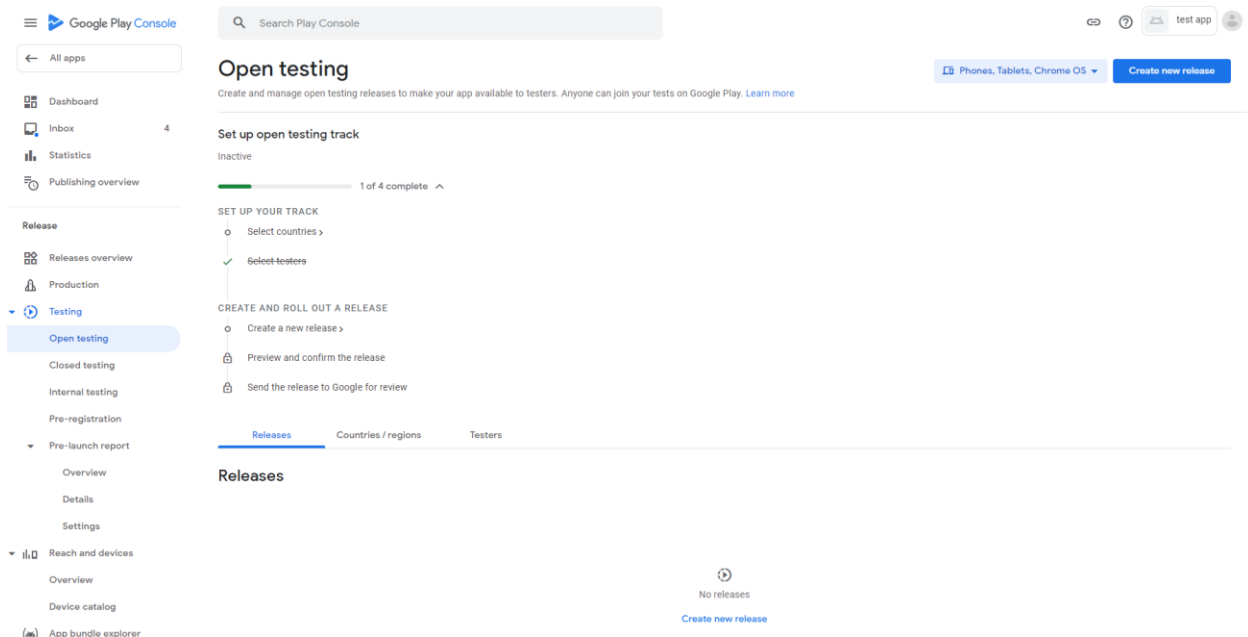
Provide information about your app and set up your store listing ✓
 Let us know about the content of your app, and manage how it is organized and presented on Google Play

Hide tasks ^

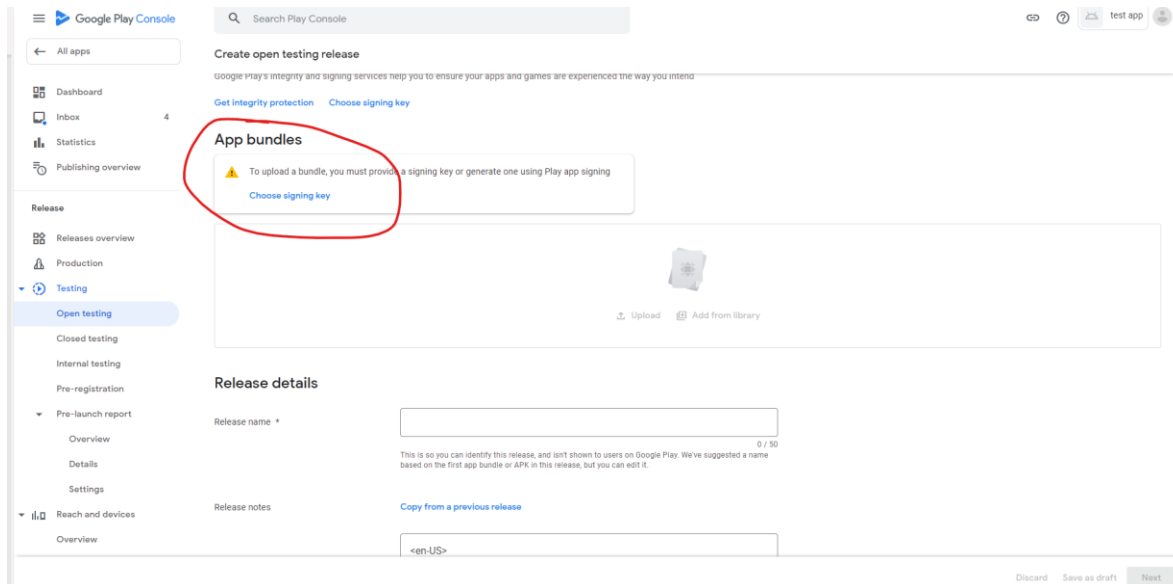
LET US KNOW ABOUT THE CONTENT OF YOUR APP

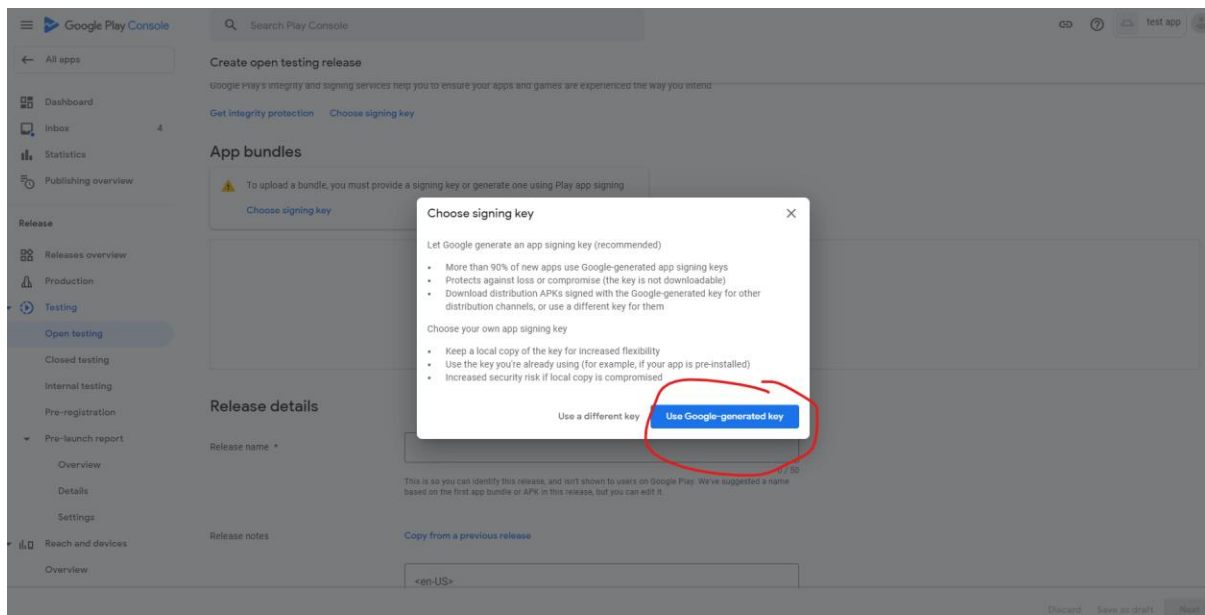
- Set privacy policy >
- App access >
- Ads >

10. Once all the task has been completed click on Open testing(left side under testing tab) and click on create new release

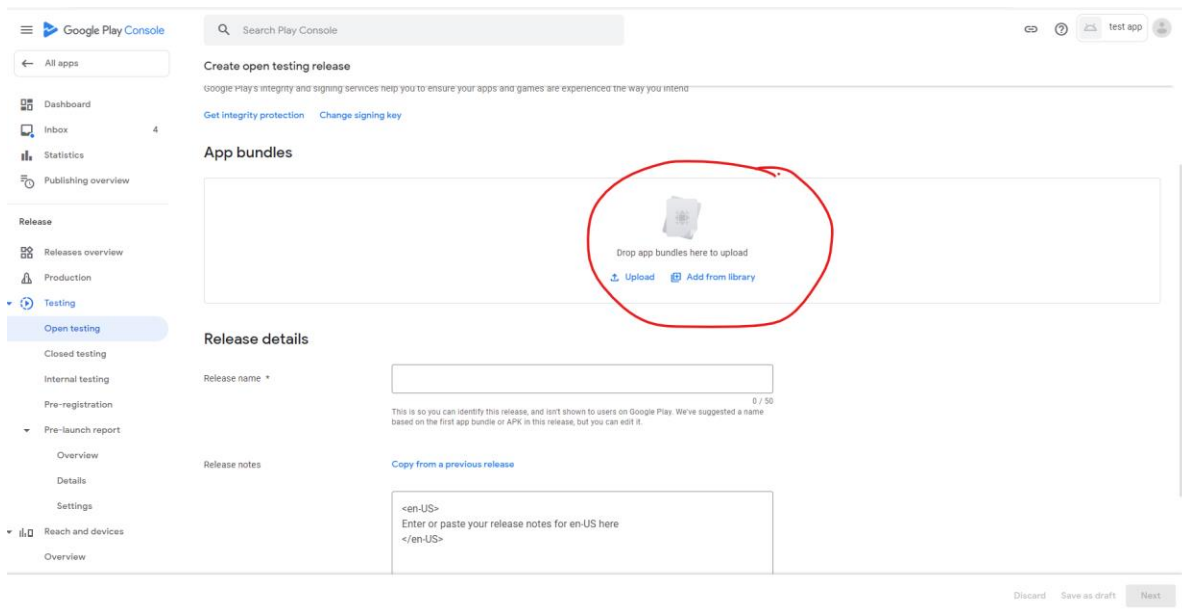


11. once created the new release, you need to choose signing key





12. Once generated, need to upload the .aab file from local drive



13. once that is uploaded, we need to give proper name and proper release note here and click on next

The screenshot shows the 'Create open testing release' page in the Google Play Console. The left sidebar contains navigation options: All apps, Dashboard, Inbox (4), Statistics, Publishing overview, Release, Releases overview, Production, Testing (selected), Open testing (selected), Closed testing, Internal testing, Pre-registration, Pre-launch report, Overview, Details, Settings, Reach and devices, and Overview. The main content area is titled 'Create open testing release' and includes a 'Release details' section. Under 'Release details', there is a 'Release name' field with a character count of 0 / 50. Below it, a note states: 'This is so you can identify this release, and isn't shown to users on Google Play. We've suggested a name based on the first app bundle or APK in this release, but you can edit it.' There is also a 'Release notes' section with a 'Copy from a previous release' link and a text area containing HTML tags: `<en-US>`, 'Enter or paste your release notes for en-US here', and `</en-US>`. Below the text area, it says 'Release notes provided for 0 language' and 'Let users know what's in your release. Enter release notes for each language within the language tags.' At the bottom right, there are buttons for 'Discard', 'Save as draft', and 'Next'.

14. after that, app name and app details we need provide properly

The screenshot shows the 'Main store listing' page in the Google Play Console. The left sidebar contains navigation options: Setup (App signing, Internal app sharing, Advanced settings), Grow (Store presence (selected), Main store listing (selected), Custom store listings, Store listing experiments, Store settings, Translation service), Store performance (Store analysis, Listing conversion analysis), Deep links, Quality, Ratings and reviews, and Android vitals (Overview). The main content area is titled 'Main store listing' and includes a 'Listing assets' section. Under 'Listing assets', there is a note about checking the Metadata policy and Help Center guidance. Below this, there are three text areas: 'App name' with the value 'Rencata Task' and a character count of 12 / 30; 'Short description' with the value 'Task Management App' and a character count of 19 / 80; and 'Full description' with a multi-line text area containing the following text: 'The user can be able to add their task or Todo list in this app. They can be able to mark as Todo list or task list has completed. The User can be able to search from task list. Delete feature also we can use, if needed.' and a character count of 221 / 4000. At the bottom right, there are buttons for 'Discard' and 'Save'.

15. Add App icon and feature Graghic, phone screenshot, 7-inch tablet screenshots, 10-inch tablet screenshots, Chromebook screenshots(optinal) and click on save.

Google Play Console

Search Play Console

Setup

App signing

Internal app sharing

Advanced settings

Grow

Store presence

Main store listing

Custom store listings

Store listing experiments

Store settings

Translation service

Store performance

Store analysis

Listing conversion analysis

Deep links

Quality

Ratings and reviews

Android vitals

Overview


Rencata Task

Main store listing

Default - English (United States) - en-US Manage translations

Manage your app icon, screenshots, and videos to promote your app on Google Play. Review the [content guidelines](#) before uploading new graphics. If you add translations for your store listing without localized graphics, we will use the graphics from your default language.


App icon *



Replace

Your app icon must be a PNG or JPEG, up to 1 MB, 512 px by 512 px, and meet our [design specifications](#) and [metadata policy](#).

Feature graphic *



Replace

Discard

Save

Google Play Console

Search Play Console

Setup

App signing

Internal app sharing

Advanced settings

Grow

Store presence

Main store listing

Custom store listings

Store listing experiments

Store settings

Translation service

Store performance

Store analysis

Listing conversion analysis

Deep links

Quality

Ratings and reviews

Android vitals

Overview

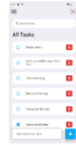
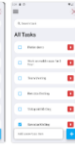
Rencata Task

Main store listing

Default - English (United States) - en-US Manage translations

PHONE

Phone screenshots *

2 / 8

Upload



Upload 2-8 phone screenshots. Screenshots must be PNG or JPEG, up to 8 MB each, 16:9 or 9:16 aspect ratio, with each side between 320 px and 3,840 px.

To be eligible for promotion, include at least 4 screenshots at a minimum of 1080 px on each side.

[View content guidelines](#)

Tablet

7-inch tablet screenshots *

Discard

Save

16. need to give app content as per original proper content and give proper policy URL(note: if anything is wrong your app will reject by google)

Google Play Console

Search Play Console

GDPR ? Rencata Task

Monetize

Products

App pricing

In-app products

Subscriptions

Price experiments

Promo codes

Financial reports

Overview

Subscriptions

Revenue

Buyers

Conversions

Monetization setup

Policy and programs

Policy status

App content

Teacher Approved

App content

Declaration	Last edited	
Privacy policy ⓘ	Sep 29, 2023	Manage
Target audience and content ⓘ	Sep 22, 2023	Manage
Content ratings ⓘ	Sep 22, 2023	Manage
Financial features ⓘ	Sep 22, 2023	Manage
Government apps ⓘ	Sep 22, 2023	Manage
Data safety ⓘ	Sep 22, 2023	Manage
COVID-19 contact tracing and status apps ⓘ	Sep 22, 2023	Manage
News apps ⓘ	Sep 22, 2023	Manage
Ads ⓘ	Sep 22, 2023	Manage
App access ⓘ	Sep 22, 2023	Manage
Advertising ID ⓘ	Sep 22, 2023	Manage

17. then click on open testing and you can see send 11 changes to google, once that is clicked your app will send for open testing. If they approved, app will update/published into the play store.