**Steps to Upload Android App to Google Play Store**

This guide is to upload an app to Google Play Store

**Step 1:** Make Developer Account

**Step 2:** Create the App

**Step 3:** Set up App Content

**Step 4:** Store Presence

**Step 5:** Create App Release

**Step 6:** Requirement for Azure DevOps Pipeline

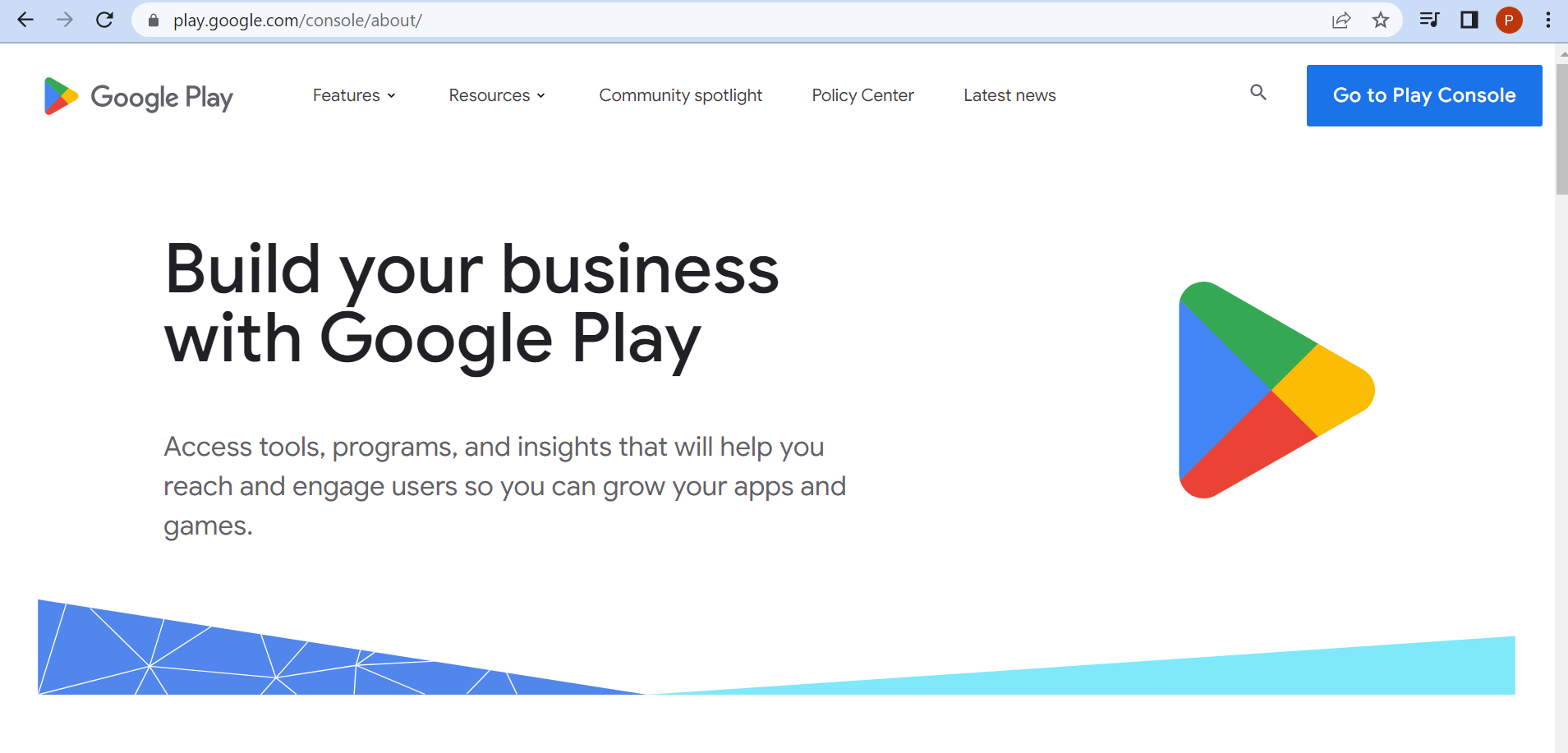
**Step 1: Make Developer Account:**

Developer Account is required to publish app into Google Play Store. The account is created in four simple steps:

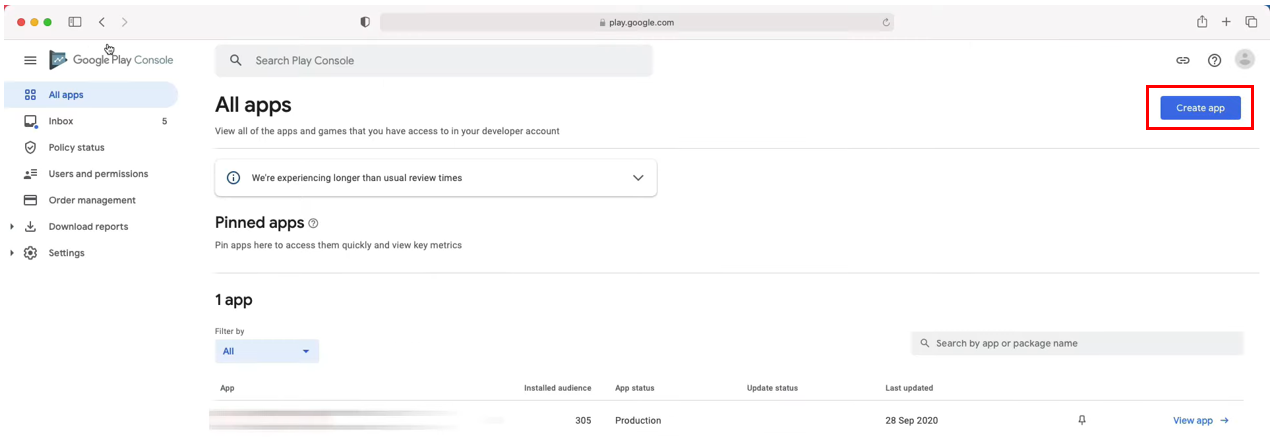
* Sign-In with Your Google Account
* Accept Developer Agreement
* Pay Registration Fee of $25.
* Complete Your Account Details

**Step 2: Create the App:**

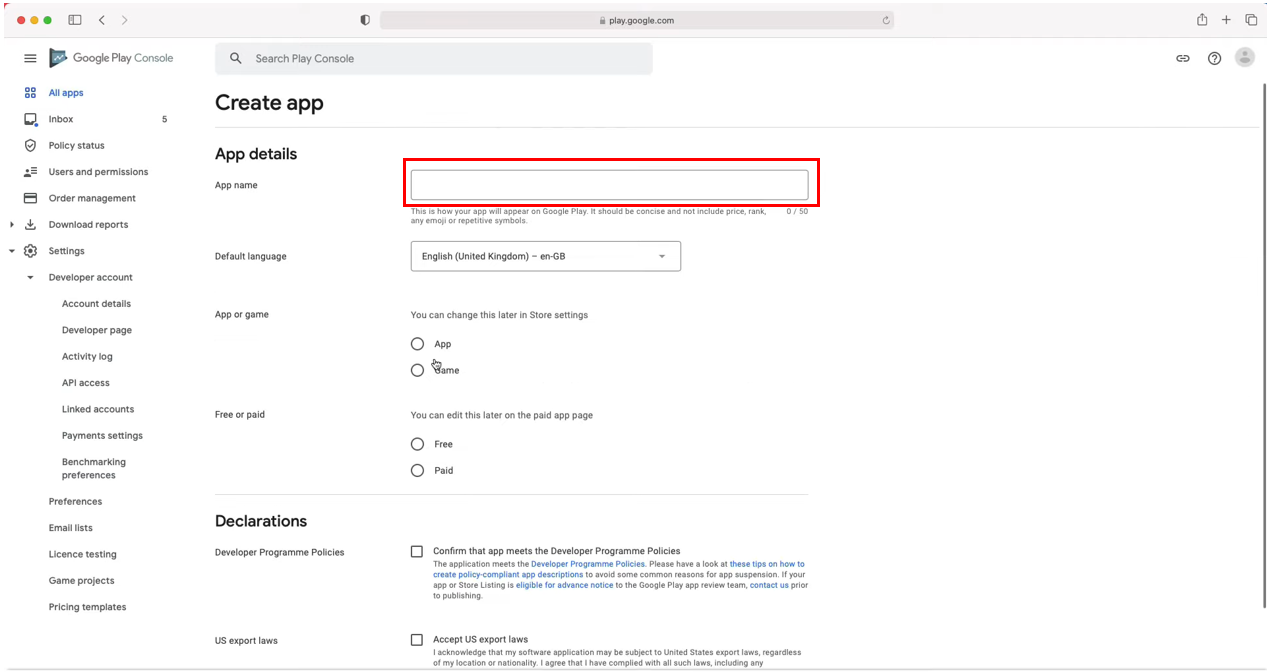
* After Creating Google Developer Account login to <https://play.google.com/console/signup> (Google Play Console)



* Click on **Go to Play Console** It will navigate to **Google Play Console**

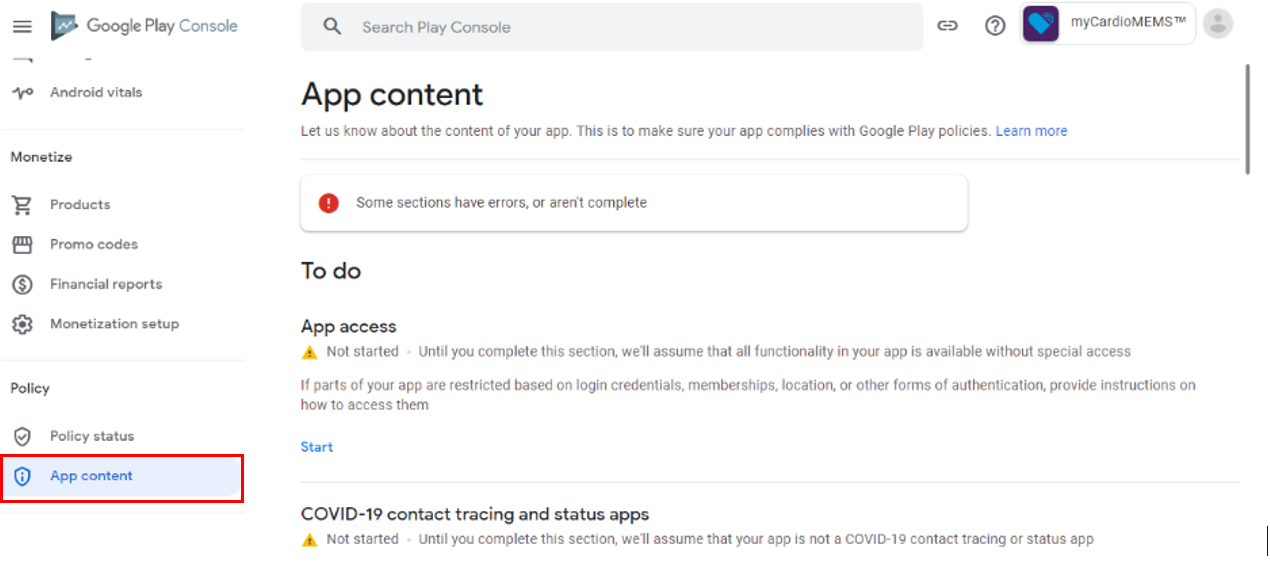


* Click on **Create app** icon
* Provide **App name** and provide below details, finally click on **Create app**
* It will navigate to App **Dashboard**

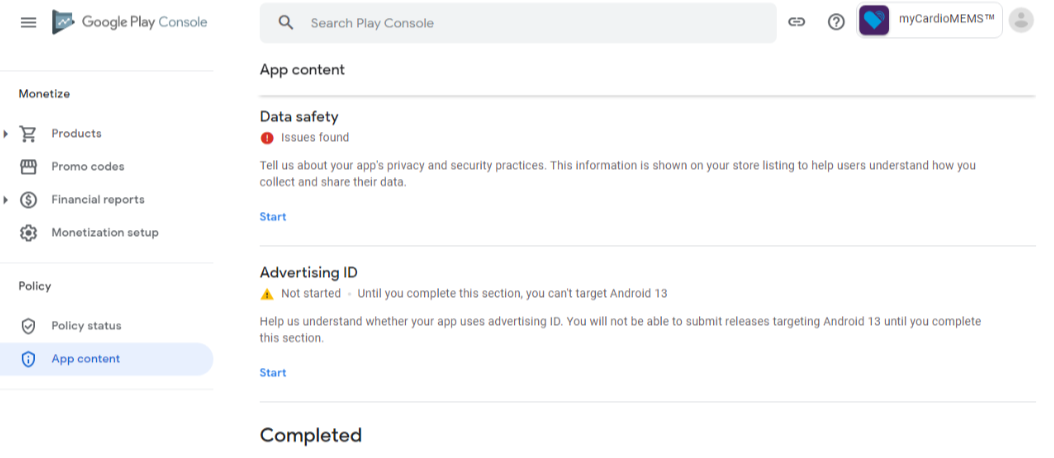


**Step 3: Set up App Content:**

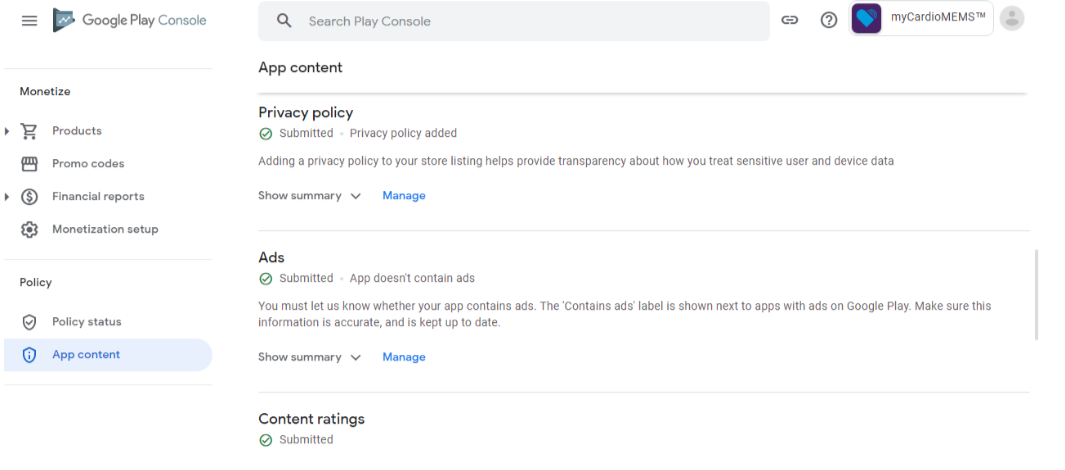
* Click on the App name, go inside the app
* Click on **App Content Button** shown like below
* Provide **App access** details



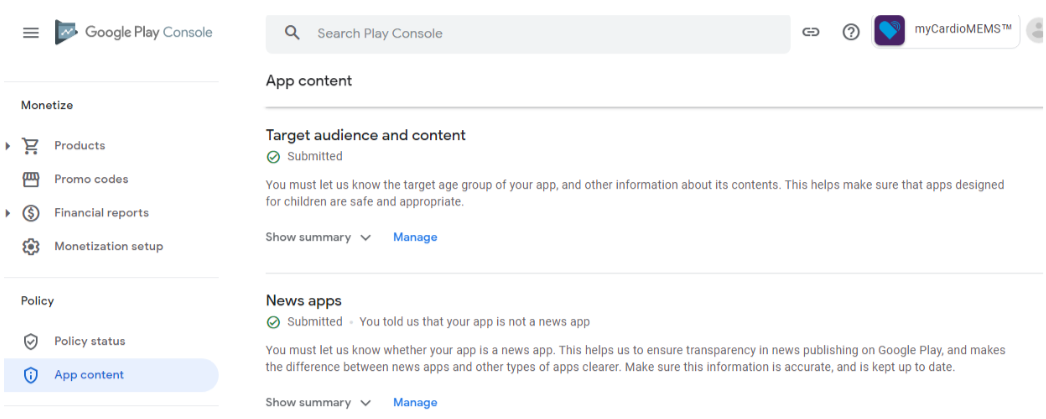
* Provide **Data safety** and **Advertising ID**



* Provide details for **Privacy policy, Ads** and for **Content ratings** will have **IARC** questions for app rating

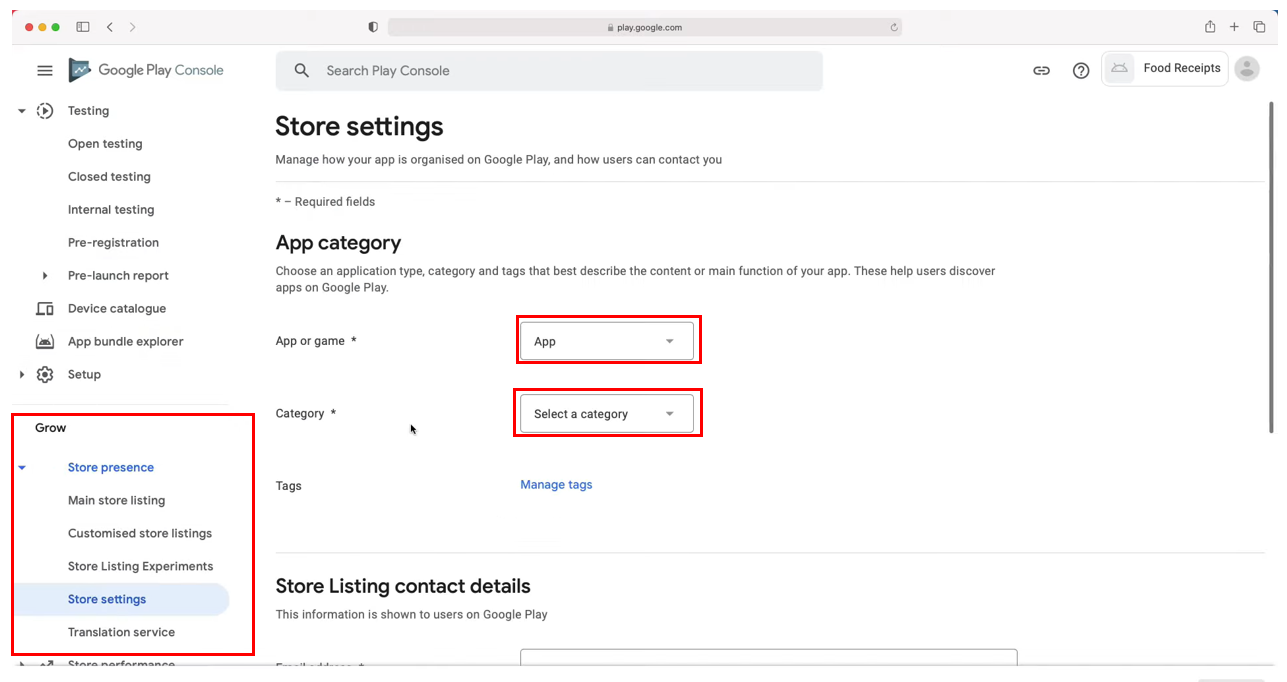


* Provide details for **Target audience and content** and **News apps**

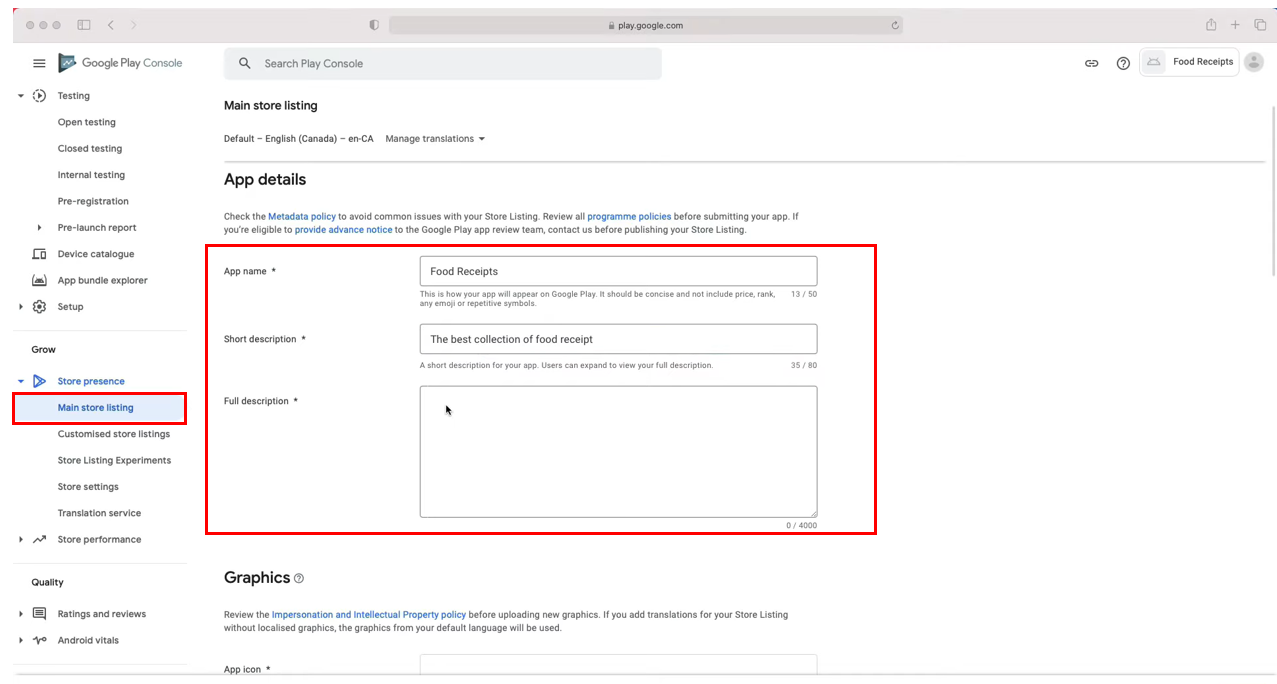


**Step 4: Store Presence:**

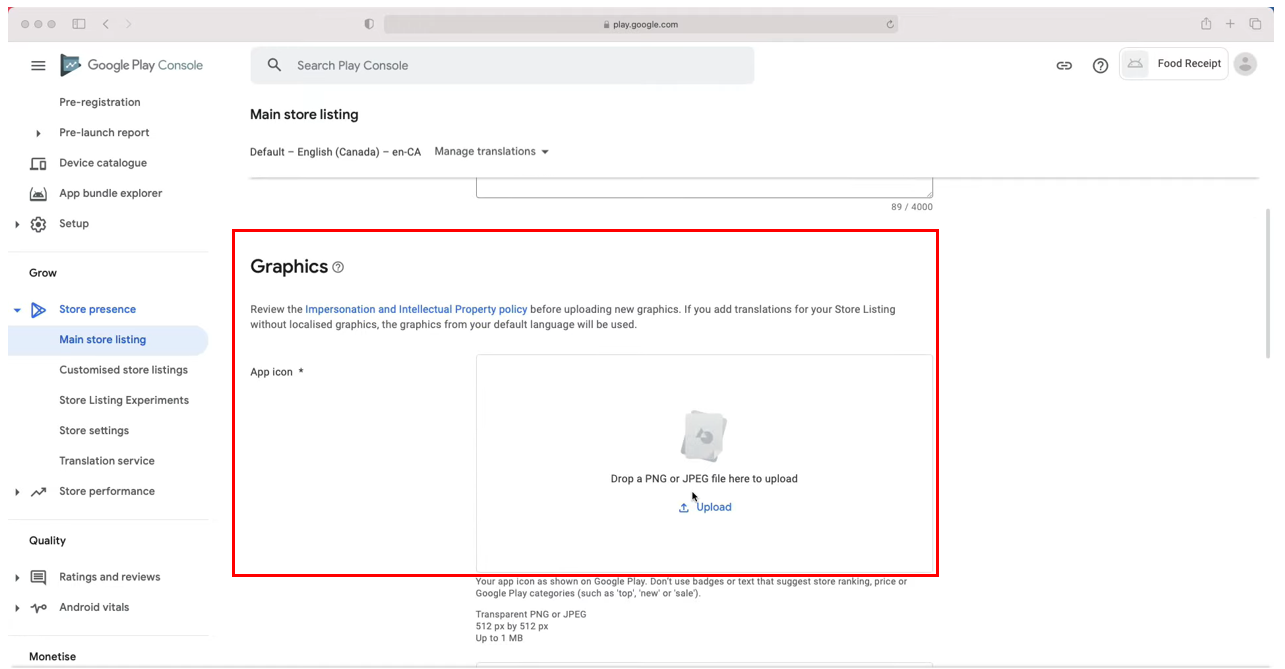
* Go to the **Grow** section, Click on **Store presence** and Click on **Store Settings** in the drop down
* Here provide the details about **App category App or game** and **Category**
* Provide the **Store listing contact details and** save
  + **Email**
  + **Phone number**
  + **Website**



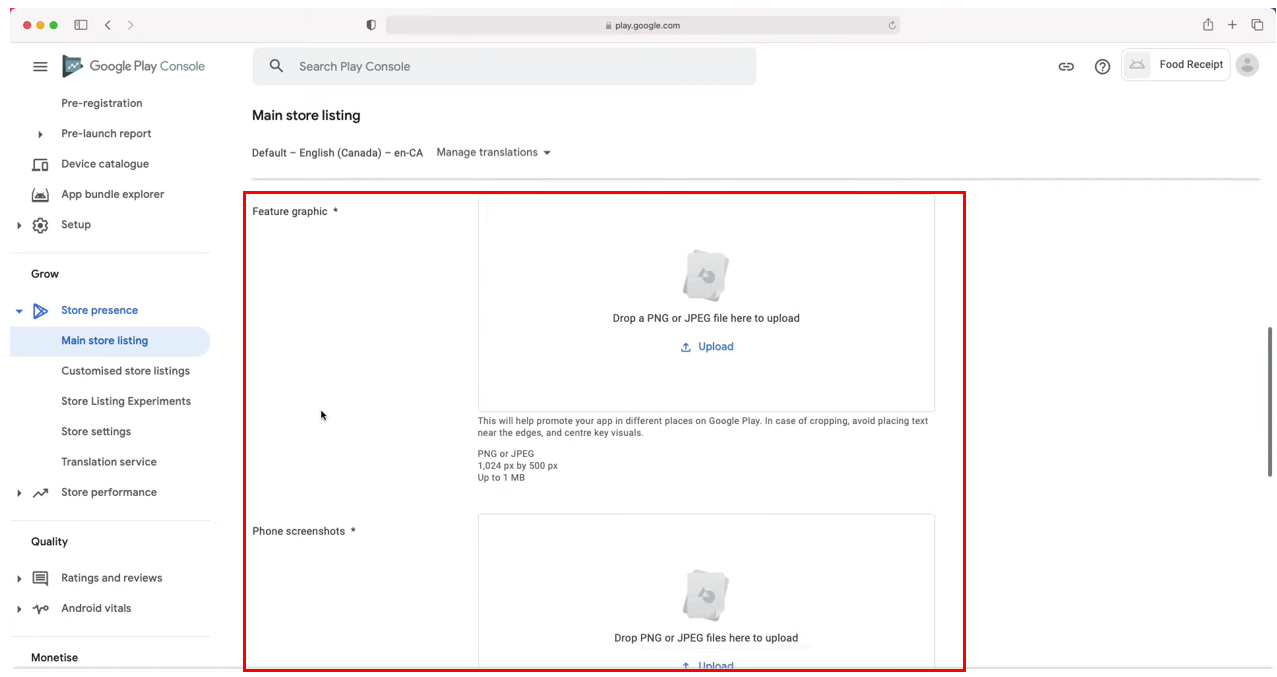
* Next click on **Main store listing** option
* Here we have to provide details about
  + **App name**
  + **Short description**
  + **Full description**
* Scroll down below to fill some more details



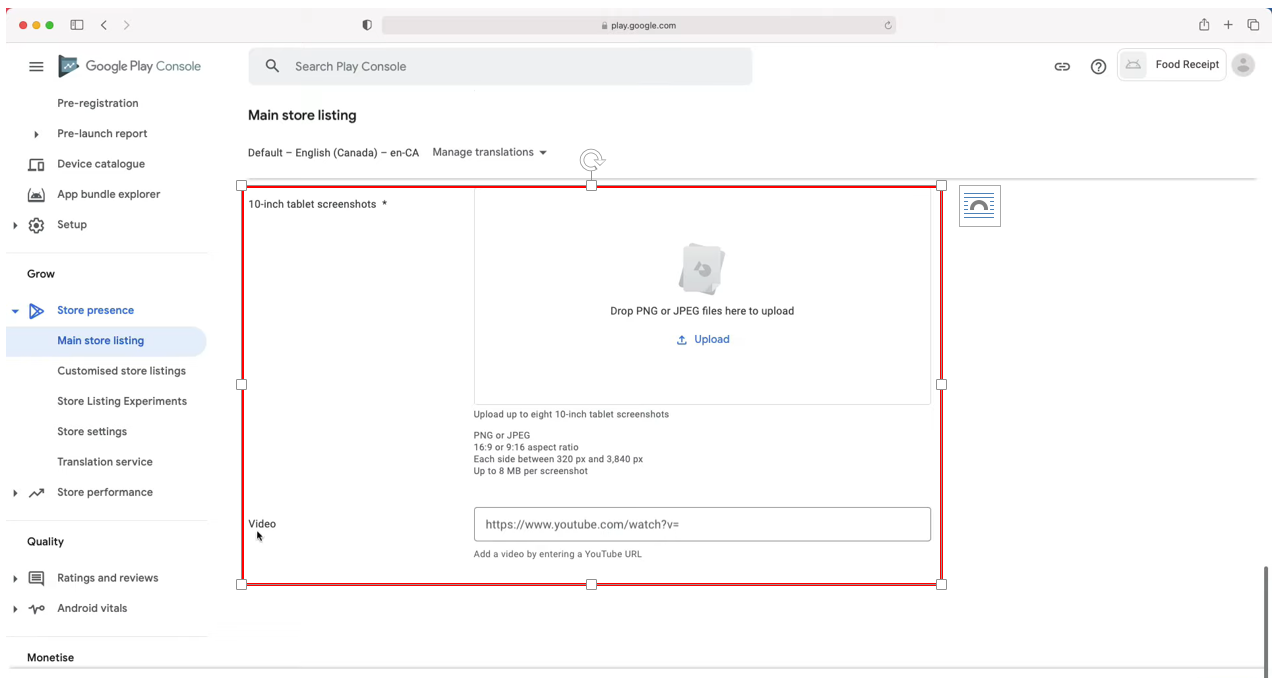
* Once scroll down get an option to fill the details about **Graphics**
* Add the **App icon** PNG or JPEG with below mentioned dimensions



* Upload the **Feature graphics** and **Phone Screenshot,** PNG or JPEG with below mentioned dimensions

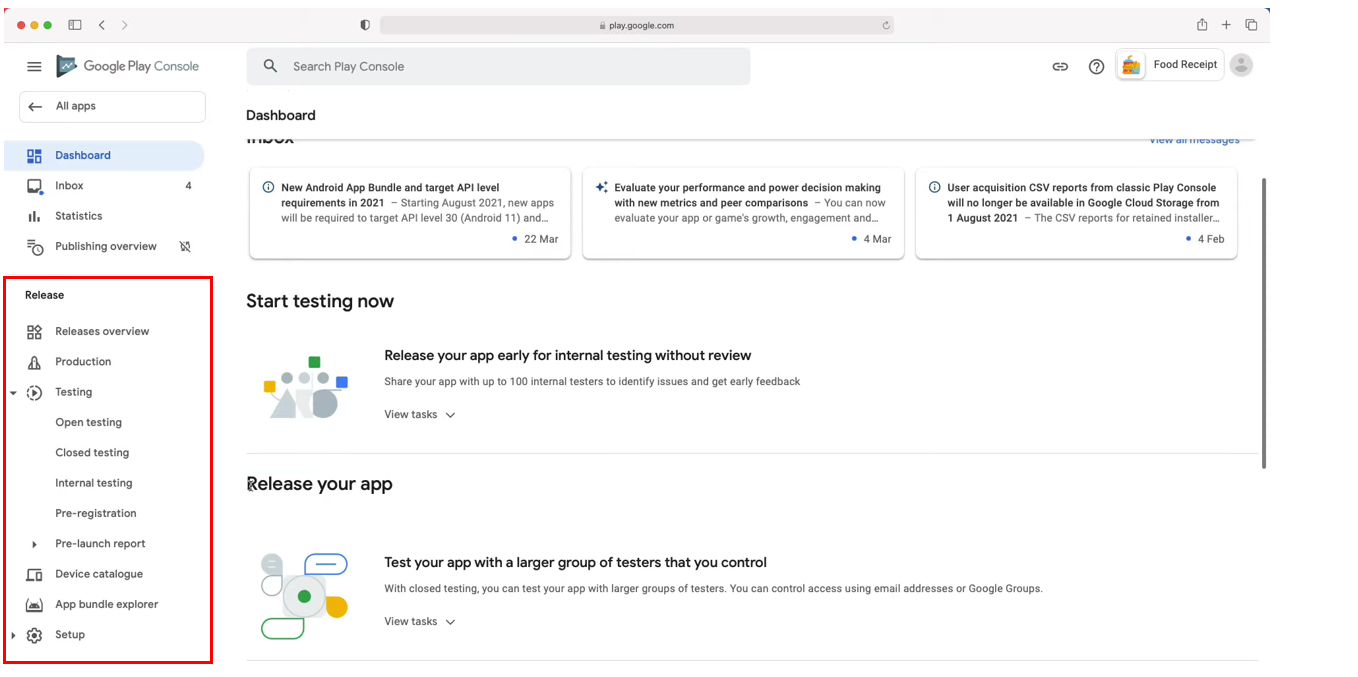


* Upload the **Tablet screenshot** and **Video** link like below**,** PNG or JPEG with below mentioned dimensions
* After filling the details click on **save** button and go to the app dashboard page



**Step 5: Create App Release:**

* Next step is to upload the AAB (Android App Bundle) to Google Play Console.
* Go to **Release** section, here its show various release option like below
  + Production
  + Open Testing
  + Closed Testing
  + Internal Testing
  + Pre-registration



**Internal testing:**

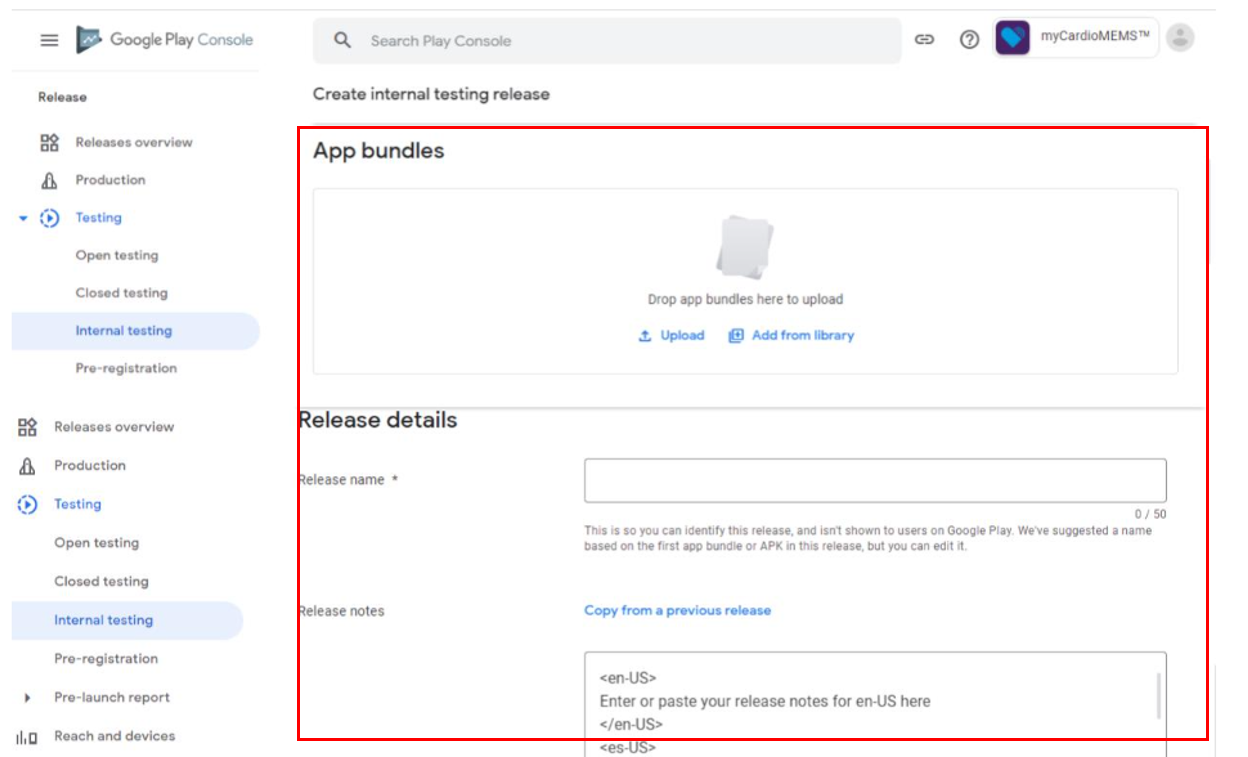
Quickly distribute your builds to a small set of trusted testers without the need to wait for app reviews. Integrate with build qualification processes before promoting to more users.

**Best practices**

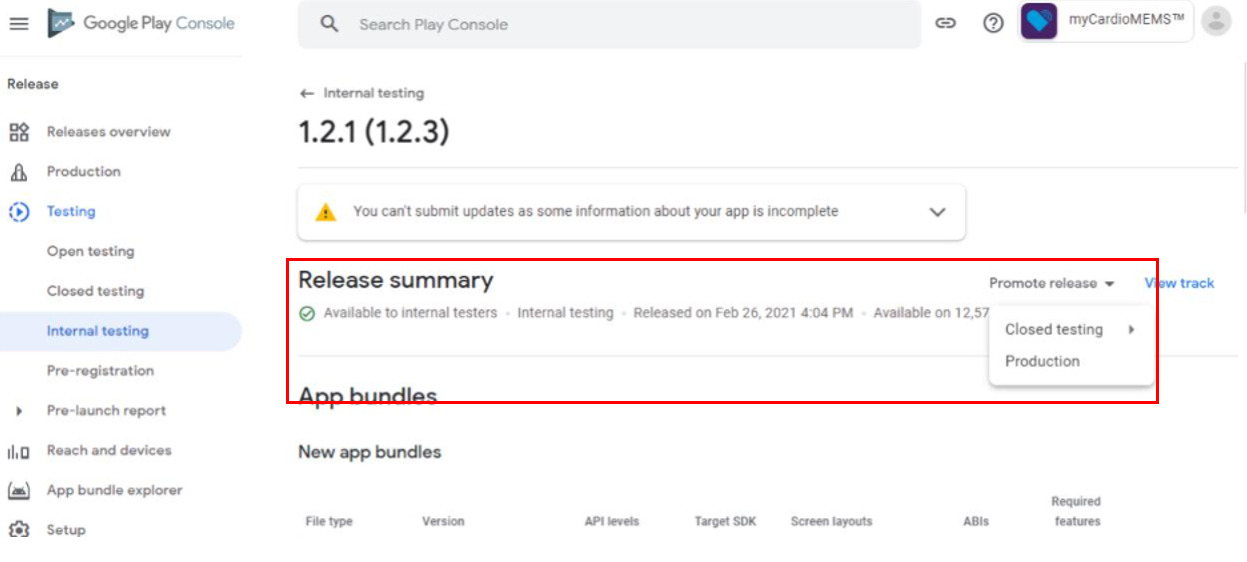
Conduct an initial quality assurance check by quickly distributing your app among up to 100 invited testers

Recommend running an internal test before releasing app to the closed or open tracks. If needed, can run internal tests concurrently with closed and open tests for different versions of your app.

* Upload the AAB (Android App Bundle) and provide the **Release details** 
  + **Release name**
  + **Release notes**



* Once the Internal testing complete then **Promote release** to next testing track by selecting the dropdown.



**Closed testing:**

Get early feedback on new features from trusted users, without impacting your public ratings and reviews

**Best practices**

To get more targeted feedback, run a closed test with specific users. Can use individual email addresses or Google Groups, or even invite selected pre-registered users to join these tests.

Provide a channel for testers to send you feedback. For closed tests, offer testers the ability to provide feedback by email, through a website, or in a message forum.

Opt in to the pre-launch report to discover any issues found in your closed testing track

Gather private feedback during open or closed tests and reply to testers directly in Play Console.

Feedback during open or closed testing is only visible and will not affect your public rating.

Once the Closed testing complete, promote to next testing track

**Open testing:**

Gather quantitative and qualitative feedback on app or game from a large number of testers

**Best practices**

Make test versions of app visible to users on Google Play to gather early feedback and monitor in-game metrics before launch

Discover any issues in open testing track by opting into the pre-launch report

Recruit more users into open test by linking users straight to details page on Google Play from any of marketing channels

Gather private feedback during open or closed tests and reply to testers directly in Play Console. Feedback during open or closed testing is only visible and will not affect public rating.

**Production:**

Make your app or game available to users on Google Play

**Best practices**

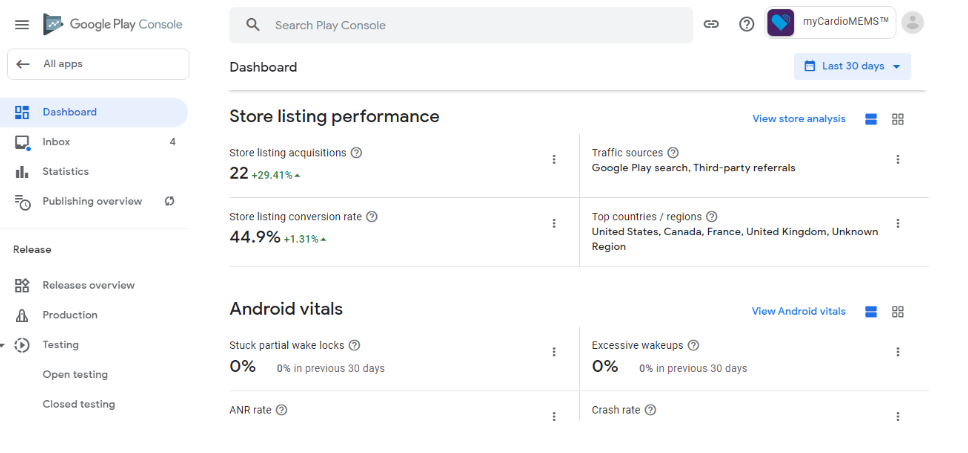
Use pre-launch reports prior to publishing your update to identify and fix issues before they impact your users

After testing app, create a release in the production track to publish it to all users. Be familiar with the three types of publishing status to understand what happens after you submit your app.

Once your app submission is approved, decide when to publish app with managed publishing, instead of it going live immediately after approval

After published app, monitor your Android vitals to quickly identify issues and bad behaviors that affect users on specific devices and Android versions

* Once the Google review complete, we can see the **App Store Listing Performance** like below



**Step 6: Requirement for Azure DevOps Pipeline:**

Required below details to complete the release automaton with the Azure DevOps

* **Service account (**Service account created form Google Play Console)
* **JSON Auth File**

**Note:**

To make the release automation with Azure DevOps, one manual release needs to create in the first time