PLANTOPIA

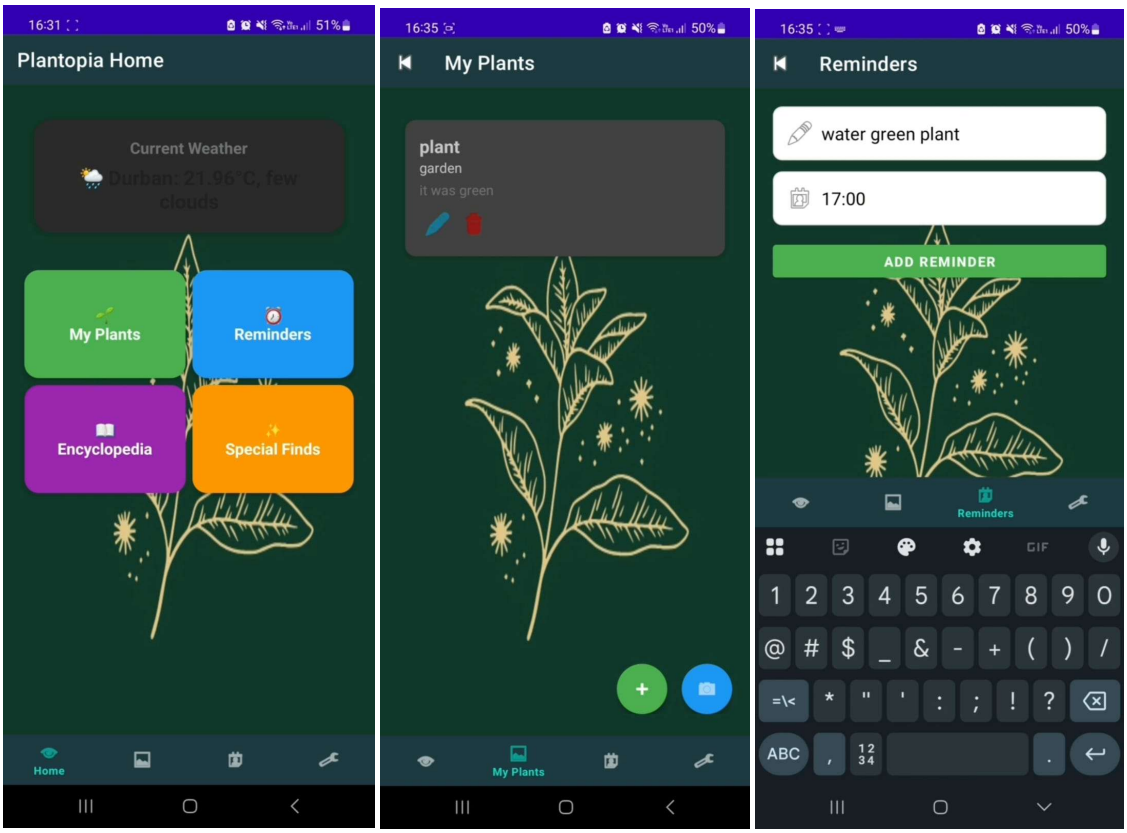
PROGRAMMING 3D – PROG 7314

PART 3: README FILE

TEAM LEAD: Bela Premchund – ST10257468

CONTENTS:

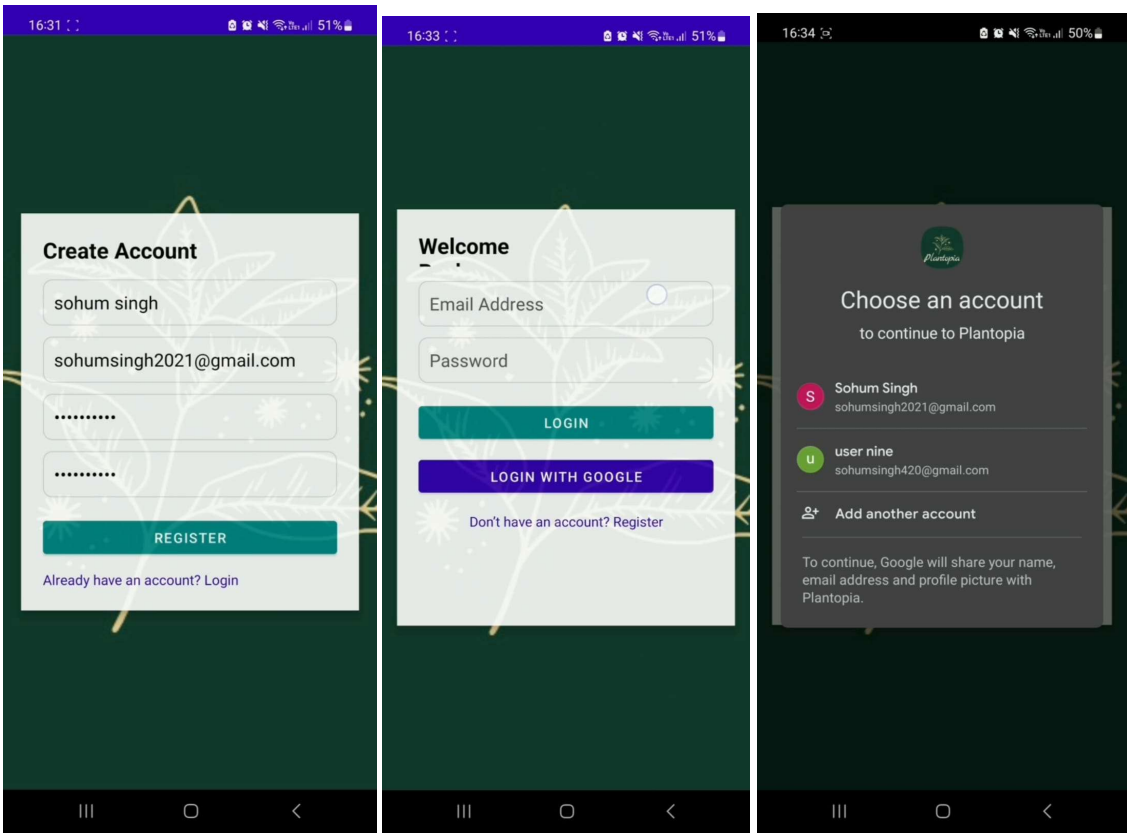
OUR APP:



Plantopia is designed to be users' plant buddy. It will allow the user to scan any plant, and they will be provided with the name, Latin name, a description of the plant, and the place of origin. Users would also be able to save these plants to their personal page called 'My Plants.'

They would then be able to create custom reminders for taking care of their plant. Whether it is to water it or to repot it, users would be able to tailor their reminders to the plant.

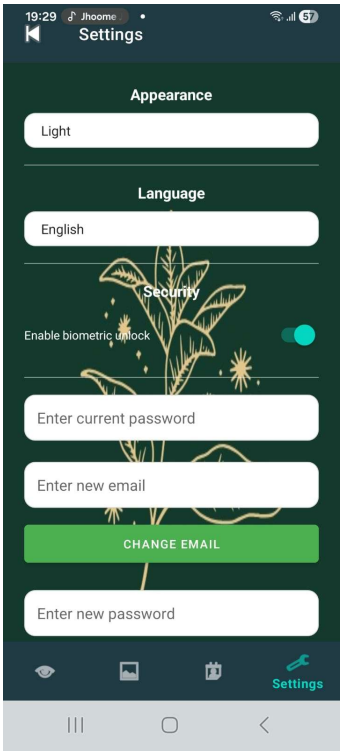
Users can find any information about any plant they are curious about, through our encyclopaedia. This will allow users to expand their knowledge and feed into their passion.



All user information is protected by a password as users would need to register and login with their own email addresses and passwords.

Users will also be able to sign in using their Google accounts (SSO).

A new feature we did add was offline capabilities. Users are able to add plants to their 'Special Finds' page when they are offline. When a user reconnects, the updates will be saved.



Users have the option to change the theme of their app from dark to light. They would also be able to change their account details such as their password.

A new feature we added was the language option. Users can switch between English, Afrikaans and isiZulu.

Users can also enable biometric sign-in where they can unlock their app using their fingerprints.

When a user logs out, they will be redirected to the welcome page.

RELEASE NOTES (FEATURES ADDED):

- On the home page we added a **Weather API**, allowing users to view the weather on that day.
- **My Plants** allow users to take a picture and store the information of any plant they want.
- **Encyclopaedia** allows users to search and view the information of any plant they wish.
- **Special Finds** is where users can store any plants they think are 'special', for example, their favourite rose.
- For **Login and Sign-up** users can either enter their information manually, or they can sign in via Google SSO.
- Users can change the **Language** of the app. They have the options of English (default), Afrikaans, or isiZulu.
- Users have the option to change their **Display settings**. This means they can select either **Light or Dark mode.** This works **offline.**
- Users can change their **Account settings –** they can change their **usernames/emails and passwords.**
- For the **Reminders**, users can set their own custom reminder for a task and when the task is due, they will receive a **notification.** This uses our **RESTful API.**

MEMBERS:

Bela Premchund – ST10257468

Altaf Ally – ST10254661

Ahmed Gangat – ST10247416

Sohum Singh - ST10257909

REQUIREMENTS:

In order to run this code, users need to have Android Studio, between version: Meerkat 2024.3.1 to Otter 2025.2.1

The following is the code for the libs.versions.toml File:

```
[versions]

agp = "8.13.0"

kotlin = "2.0.21"

coreKtx = "1.17.0"

junit = "4.13.2"

junitVersion = "1.3.0"

espressoCore = "3.7.0"

lifecycleRuntimeKtx = "2.9.4"

activityCompose = "1.8.0"

composeBom = "2024.09.00"

appcompat = "1.7.1"

constraintlayout = "2.2.1"

recyclerview = "1.4.0"

coordinatorlayout = "1.3.0"

material = "1.13.0"

roomCommonJvm = "2.8.3"

workRuntimeKtx = "2.11.0"


[libraries]
```

androidx-core-ktx = { group = "androidx.core", name = "core-ktx", version.ref = "coreKtx" }

junit = { group = "junit", name = "junit", version.ref = "junit" }

androidx-junit = { group = "androidx.test.ext", name = "junit", version.ref = "junitVersion" }

androidx-espresso-core = { group = "androidx.test.espresso", name = "espresso-core", version.ref = "espressoCore" }

androidx-lifecycle-runtime-ktx = { group = "androidx.lifecycle", name = "lifecycle-runtime-ktx", version.ref = "lifecycleRuntimeKtx" }

androidx-activity-compose = { group = "androidx.activity", name = "activity-compose", version.ref = "activityCompose" }

androidx-compose-bom = { group = "androidx.compose", name = "compose-bom", version.ref = "composeBom" }

androidx-compose-ui = { group = "androidx.compose.ui", name = "ui" }

androidx-compose-ui-graphics = { group = "androidx.compose.ui", name = "ui-graphics" }

androidx-compose-ui-tooling = { group = "androidx.compose.ui", name = "ui-tooling" }

androidx-compose-ui-tooling-preview = { group = "androidx.compose.ui", name = "ui-tooling-preview" }

androidx-compose-ui-test-manifest = { group = "androidx.compose.ui", name = "ui-test-manifest" }

androidx-compose-ui-test-junit4 = { group = "androidx.compose.ui", name = "ui-test-junit4" }

androidx-compose-material3 = { group = "androidx.compose.material3", name = "material3" }

androidx-appcompat = { group = "androidx.appcompat", name = "appcompat", version.ref = "appcompat" }

androidx-constraintlayout = { group = "androidx.constraintlayout", name = "constraintlayout", version.ref = "constraintlayout" }

androidx-recyclerview = { group = "androidx.recyclerview", name = "recyclerview", version.ref = "recyclerview" }

androidx-coordinatorlayout = { group = "androidx.coordinatorlayout", name = "coordinatorlayout", version.ref = "coordinatorlayout" }

material = { group = "com.google.android.material", name = "material", version.ref = "material" }

androidx-room-common-jvm = { group = "androidx.room", name = "room-common-jvm", version.ref = "roomCommonJvm" }

androidx-work-runtime-ktx = { group = "androidx.work", name = "work-runtime-ktx", version.ref = "workRuntimeKtx" }

[plugins]

android-application = { id = "com.android.application", version.ref = "agp" }

kotlin-android = { id = "org.jetbrains.kotlin.android", version.ref = "kotlin" }

kotlin-compose = { id = "org.jetbrains.kotlin.plugin.compose", version.ref = "kotlin" }

GITHUB LINK:

Bela Premchund (main link with collaborators): https://github.com/VCWVL/prog7314-poe-ST10257468.git

Altaf Ally: https://github.com/VCWVL/prog7314-poe-Altaf-Ally.git

Ahmed Gangat: https://github.com/VCWVL/prog7314-poe -AhmedGangat.git

Sohum Singh: https://github.com/VCWVL/prog7314-poe -Sohum-Singh.git

VIDEO LINK:

https://drive.google.com/drive/folders/1eXmf0MT_DTSh4hcO5x0ABtdS1Abe8x7x

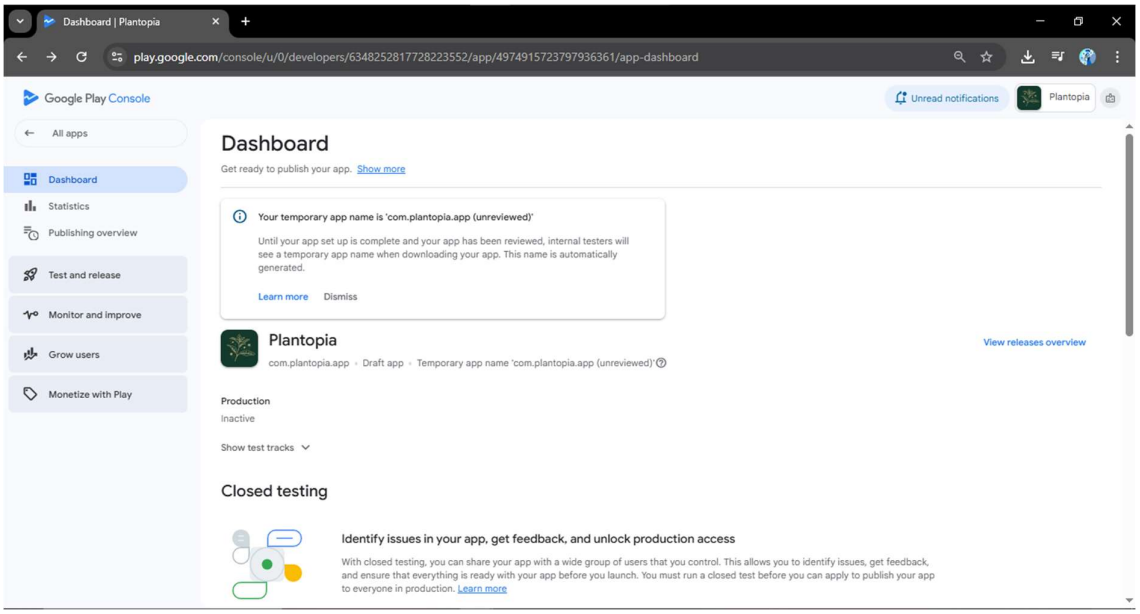
DEVELOPER INFORMATION:

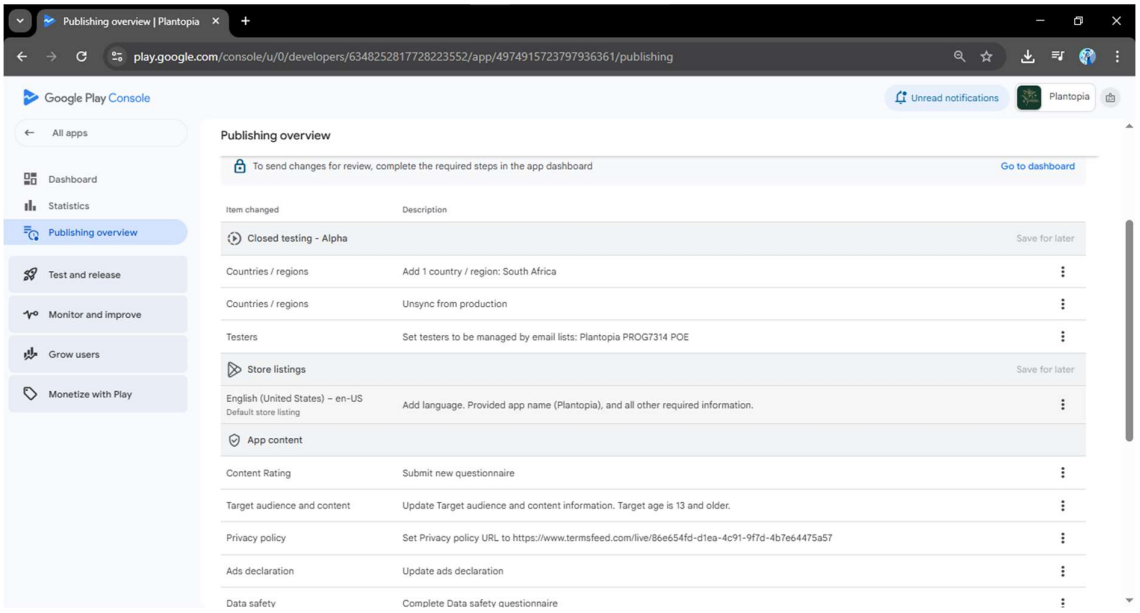
Developer: Altaf Ally

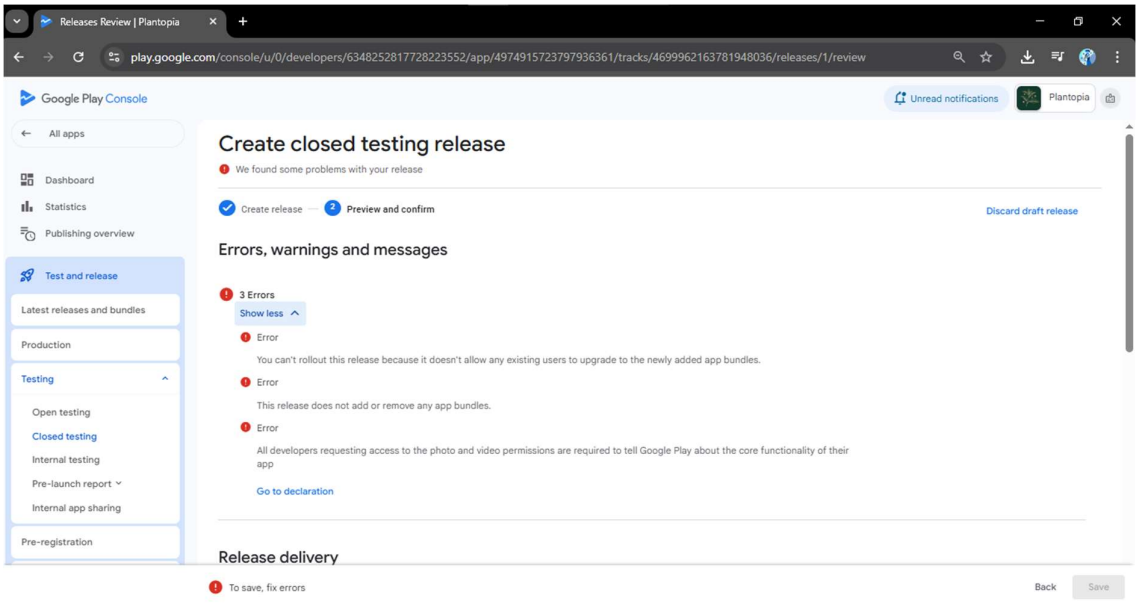
Student Number: ST10254661

Contact: ST10254661@vcconnect.edu.za

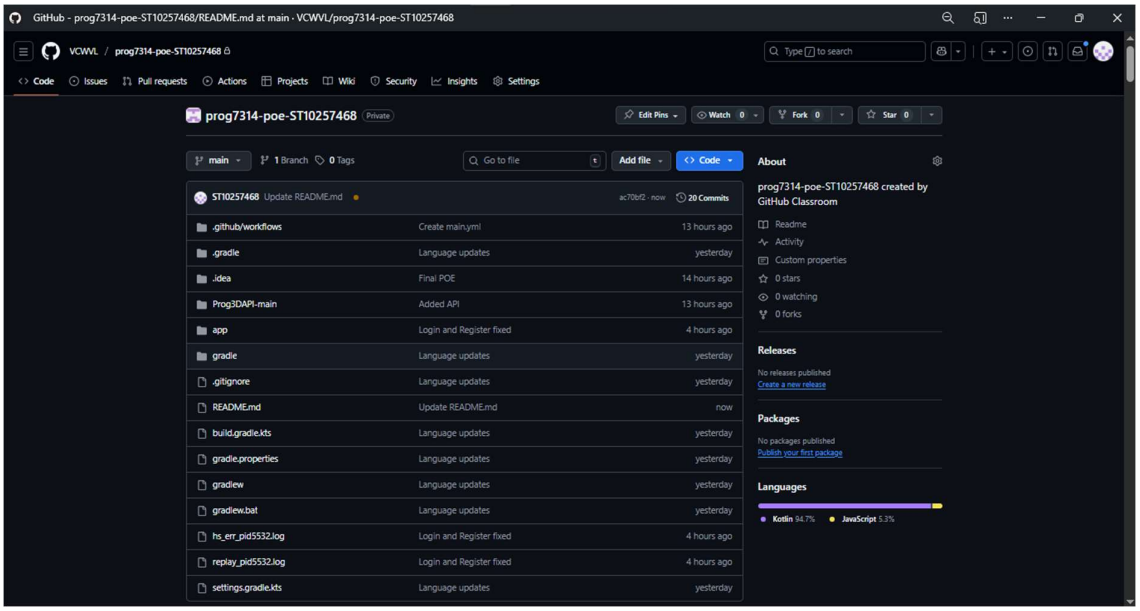
PLAYSTORE PUBLISHING:





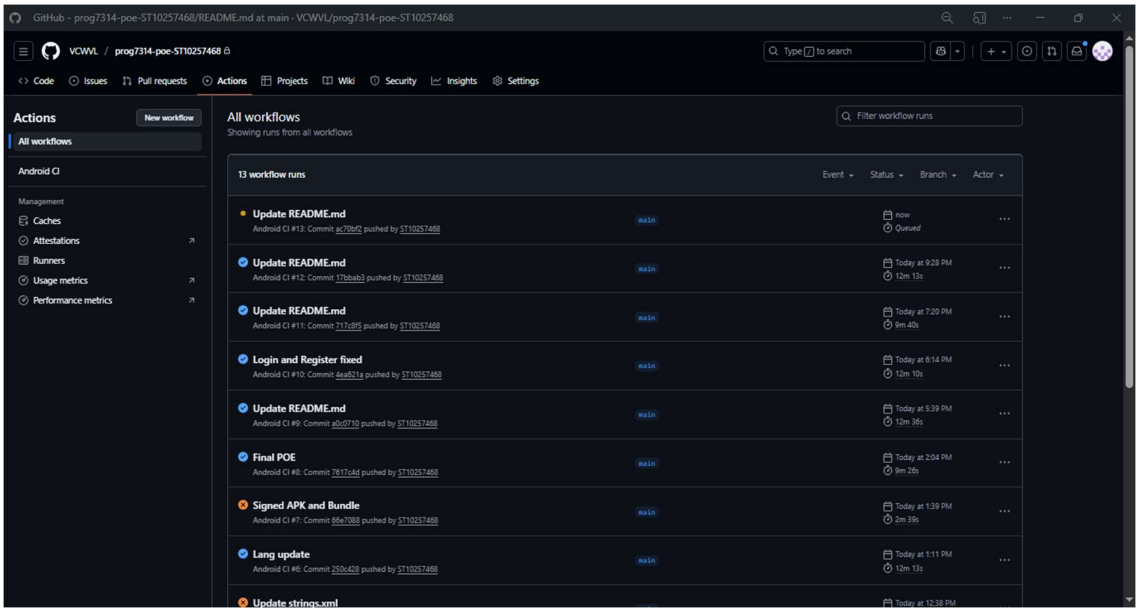


UTILISATION OF GITHUB AND GITHUB ACTIONS:



The image above shows that we posted the code to the app and the API to our GitHub repository. There are over 15 commits, showing what and why we changed the code.

With the use of GitHub Actions, we were able to create a new workflow that allowed us to test our app. This test runs each time a new commit is made. This ensures that the app will run on any platform.



AI-DECLARATION:

ChatGPT, 2025. *Change frame transparency.* [Online]
Available at: https://chatgpt.com/c/68e43a5a-e444-8333-8255-ed693bf4583e
[Accessed 6 October 2025].

ChatGPT, 2025. Fix Gradle error. [Online]

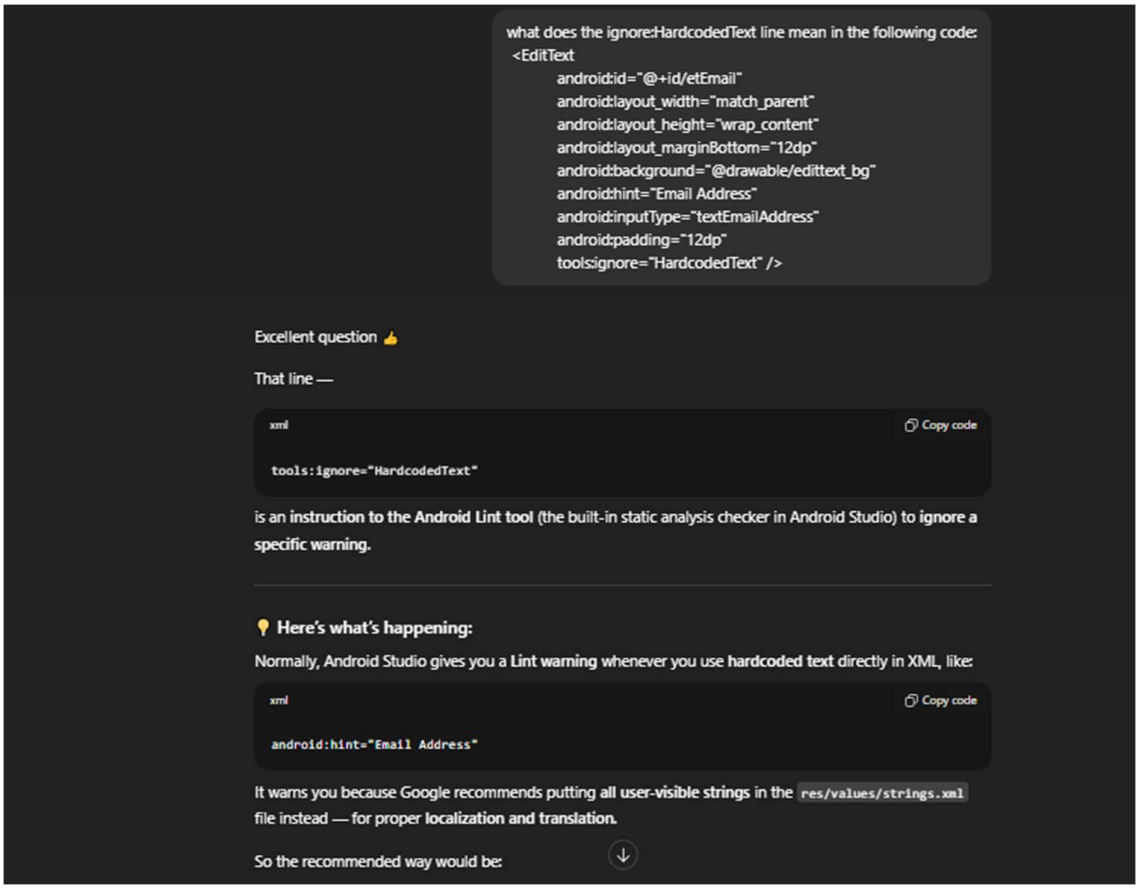
Available at: https://chatgpt.com/c/691b77b4-0eec-8326-8f38-4774a7b85188

[Accessed 17 October 2025]

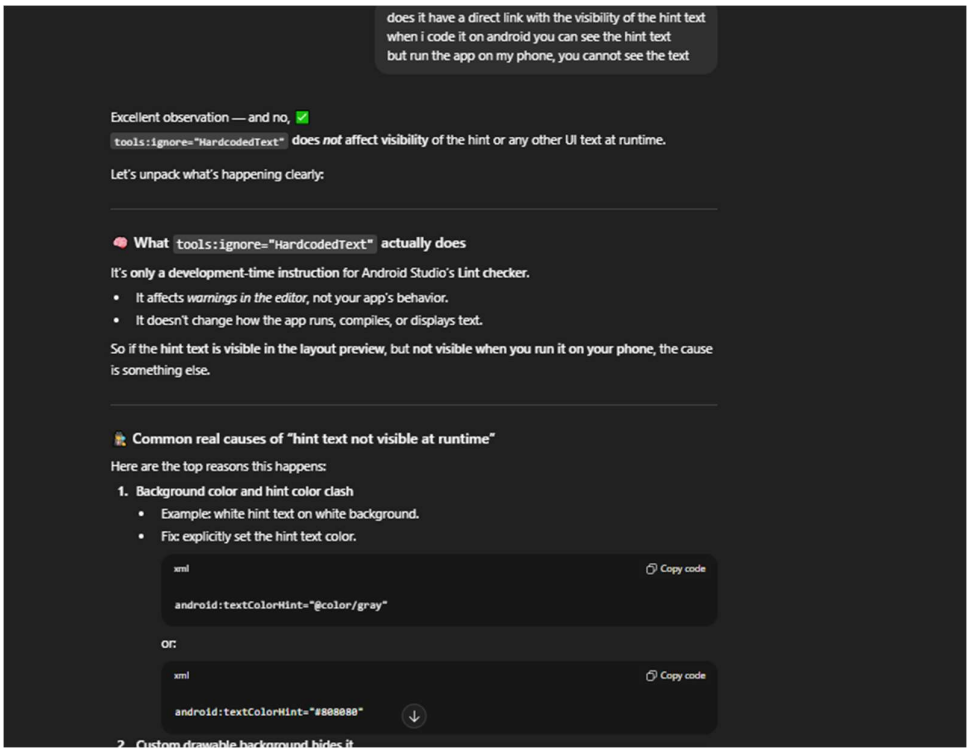
We did use ChatGPT in this assignment to assist with the UI. When we coded the UI in Android Studio, you could see the ‘hint’ text on the register and login pages. When we ran the app from our phones, the ‘hint prompts’ were not visible.

The solution was to hardcore the colour of the text to ensure that is visible on any phone. We used ChatGPT to help us get the colour code for the hint text.

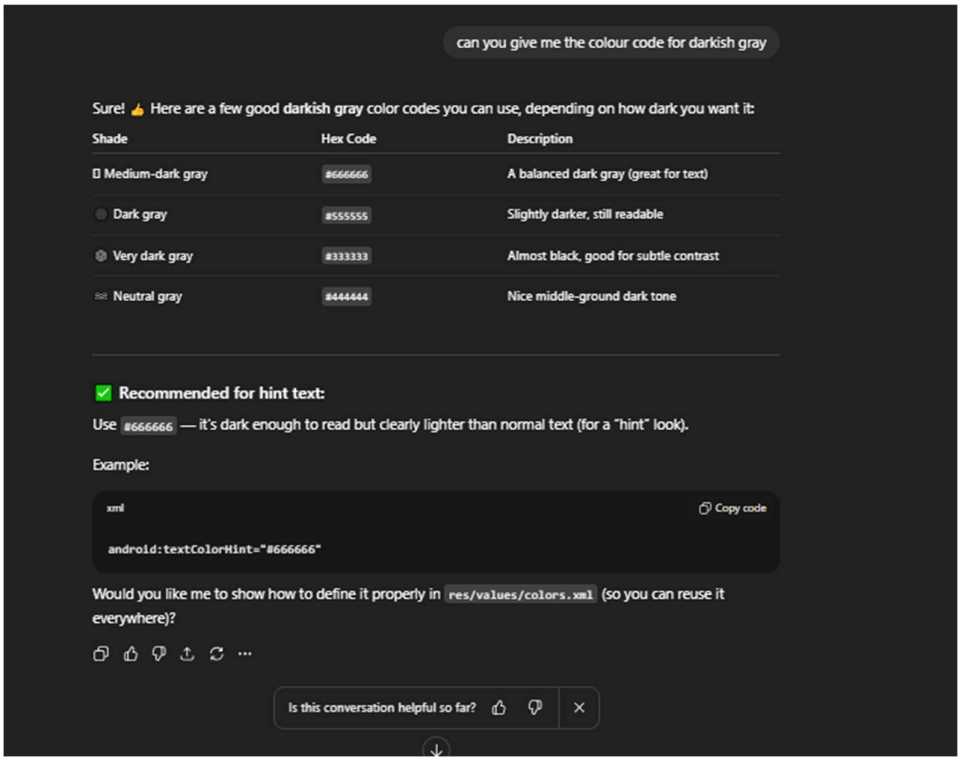
This can be seen in the images below:



One of the content fixes included this line on code: tools:ignore=”HardcodedText.” I asked ChatGPT for more details about this.

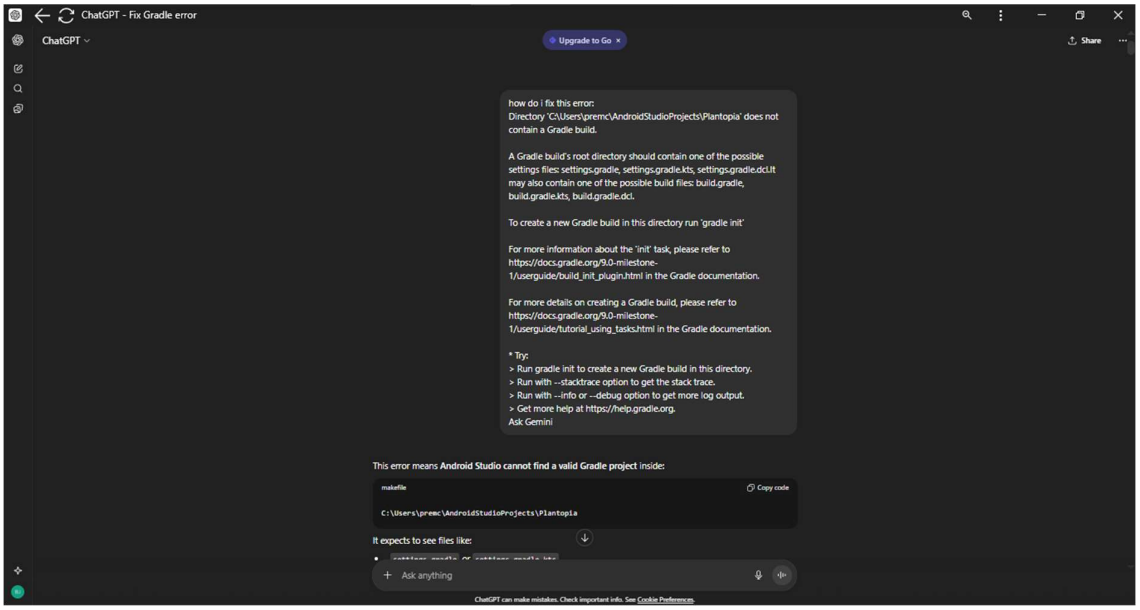


Here we asked ChatGPT for a plausible reason for our error.



Here we asked ChatGPT for the colour code so we could hardcode the colour into the pages.

We also used it to fix the errors in the gradle and other files in order for the signed APK to be created.



REFERENCES:

ChatGPT, 2025. *Change frame transparency*. [Online]
Available at: https://chatgpt.com/c/68e43a5a-e444-8333-8255-ed693bf4583e
[Accessed 6 October 2025].

ChatGPT, 2025. *Fix Gradle error*. [Online]
Available at: https://chatgpt.com/c/691b77b4-0eec-8326-8f38-4774a7b85188
[Accessed 17 November 2025].

Cocca, G., 2023. *The REST API Handbook – How to Build, Test, Consume, and Document REST APIs*. [Online]
Available at: https://www.freecodecamp.org/news/build-consume-and-document-a-rest-api/
[Accessed 4 October 2025].

Fireship, 2021. *RESTful APIs in 100 Seconds // Build an API from Scratch with Node.js Express*. [Online]
Available at: https://www.youtube.com/watch?v=-MTSQjw5DrM
[Accessed 2 October 2025].

Lackner, P., 2021. *How to Build a Simple REST API With Ktor + Android App*. [Online]
Available at: https://www.youtube.com/watch?v=c6I3Dw0xDlQ
[Accessed 4 October 2025].

OpenWeatherMap, 2024. *Weather API Documentation*. [Online]
Available at: https://openweathermap.org/api
[Accessed 3 October 2025].

Plant.id, 2024. *Plant Identification API Documentation*. [Online]
Available at: https://web.plant.id/documentation
[Accessed 3 October 2025].

Trefle API, 2024. *Trefle Plant Data API*. [Online]
Available at: https://trefle.io/
[Accessed 3 October 2025].