

PLANTOPIA

PROGRAMMING 3D – PROG 7314

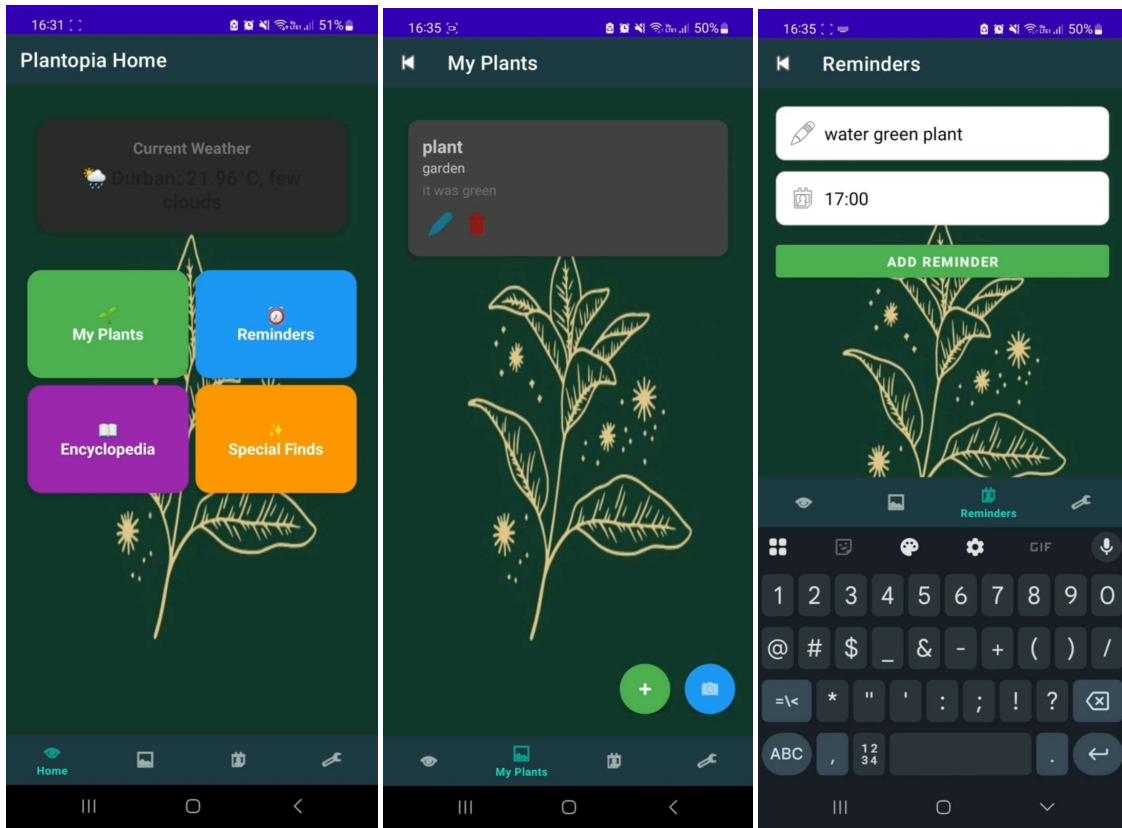
PART 3: README FILE

TEAM LEAD: Bela Premchund – ST10257468

CONTENTS:

- Our app (Screenshots)
- Release notes
- Members
- Requirements (libs.versions.toml File)
- GitHub Link
- Video link
- Developer Information
- Design Considerations
- PlayStore Publication
- Utilisation of GitHub and GitHub Actions
- AI Declaration
- References

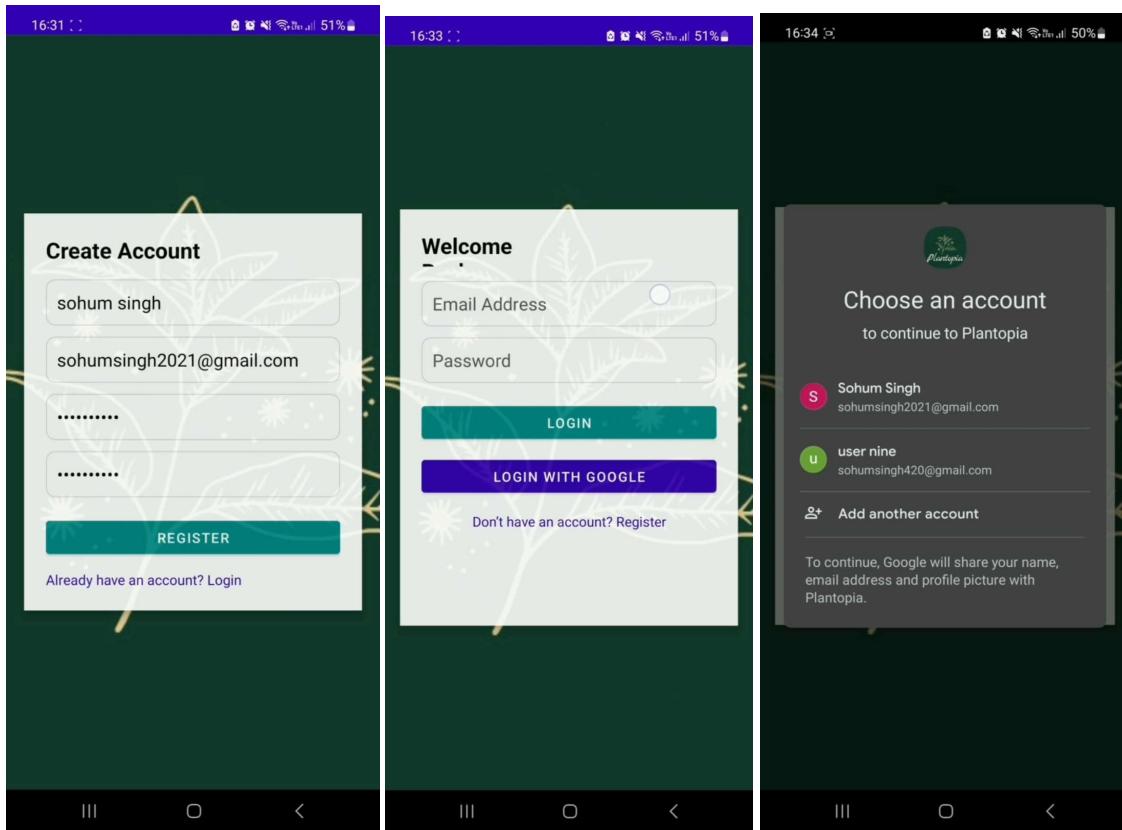
OUR APP:



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.

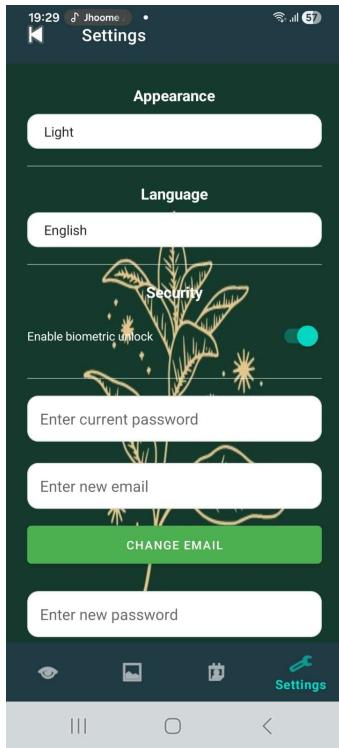


All user information is protected by a password as users would need to register and login with their own email addresses and passwords.

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_DTSsh4hcO5x0ABtdS1Abe8x7x

DEVELOPER INFORMATION:

Developer: Altaf Ally

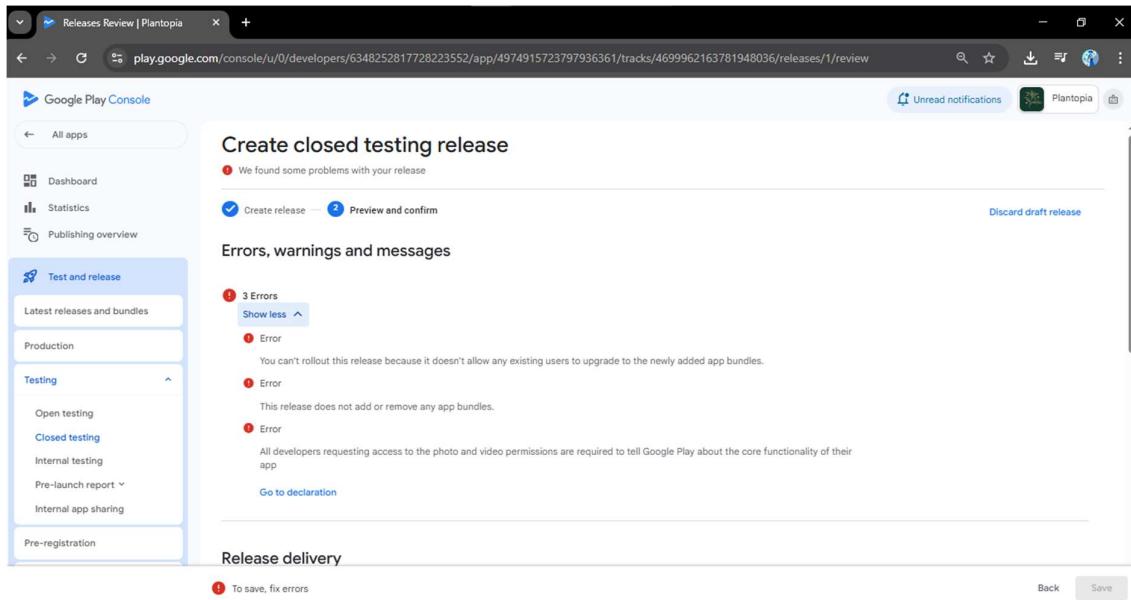
Student Number: ST10254661

Contact: ST10254661@vcconnect.edu.za

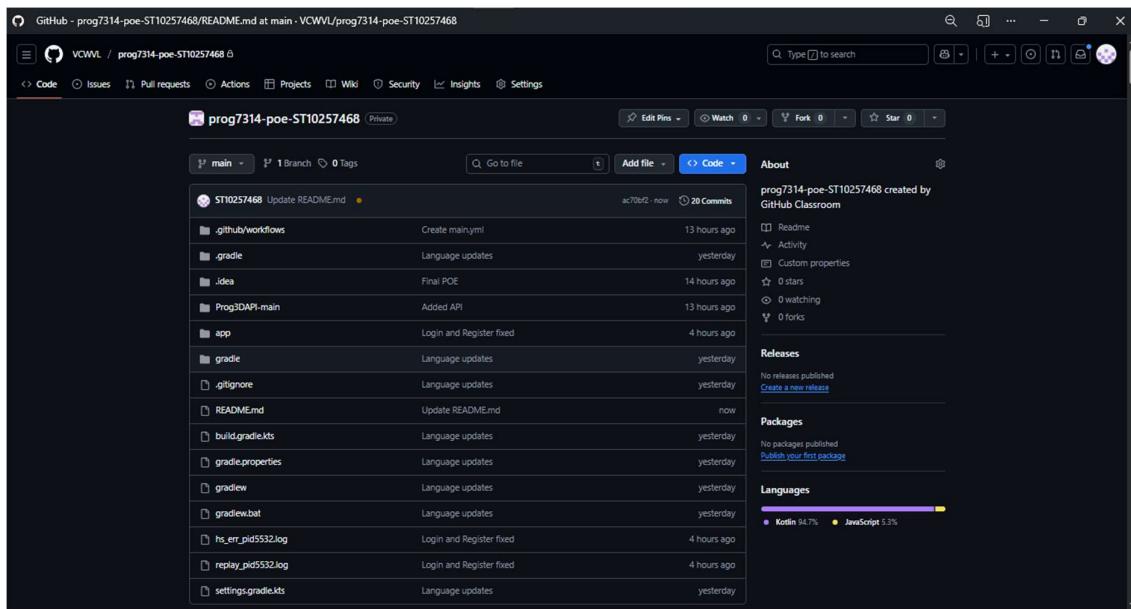
PLAYSTORE PUBLISHING:

The screenshot shows the Google Play Console Dashboard for the app 'Plantopia'. The left sidebar has a 'Dashboard' tab selected. A prominent message box says: 'Your temporary app name is 'com.plantopia.app (unreviewed)''. Below it, under 'Production', it says 'Inactive'. On the right, there's a 'View releases overview' link.

The screenshot shows the Google Play Console Publishing overview page for the same app. The left sidebar has a 'Publishing overview' tab selected. It displays a 'Closed testing - Alpha' section with details: 'Countries / regions' (South Africa), 'Testers' (Plantopia PROG7314 POE), and a note to 'Add 1 country / region'. There are also sections for 'Store listings' (English language listing) and 'App content' (Content Rating, Target audience, Privacy policy, Ads declaration, Data safety).

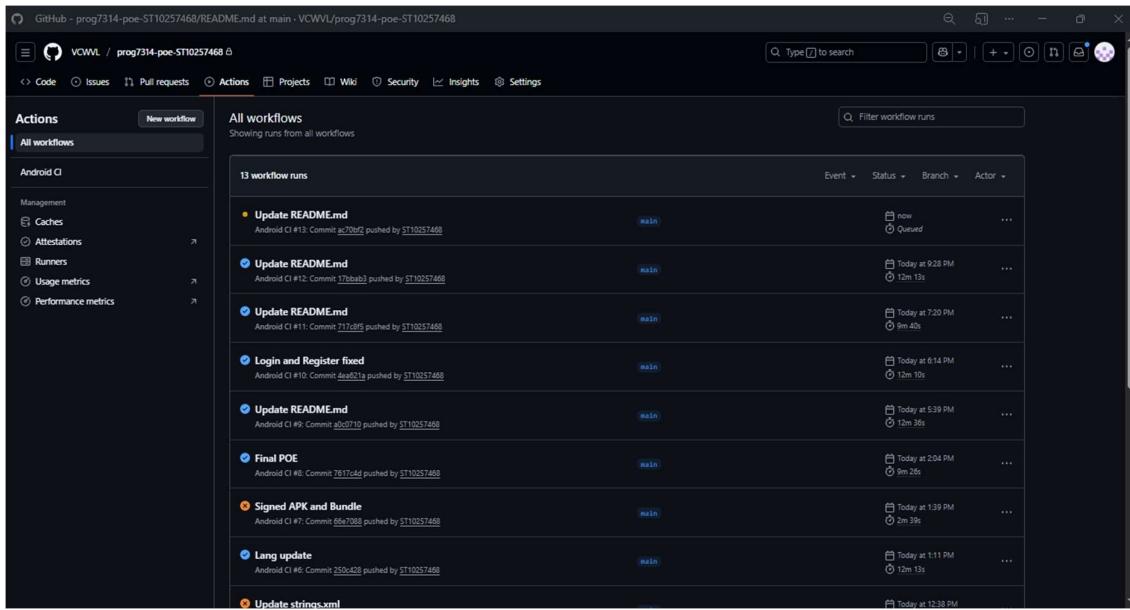


UTILISATION OF GITHUB AND GITHUB ACTIONS:



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:

what does the ignoreHardcodedText line mean in the following code:

```
<EditText  
    android:id="@+id/etEmail"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_marginBottom="12dp"  
    android:background="@drawable/edittext_bg"  
    android:hint="Email Address"  
    android:inputType="textEmailAddress"  
    android:padding="12dp"  
    tools:ignore="HardcodedText" />
```

Excellent question 

That line —

```
xml  
  
tools:ignore="HardcodedText"
```

 Copy code

is an instruction to the **Android Lint tool** (the built-in static analysis checker in Android Studio) to ignore a specific warning.

💡 Here's what's happening:

Normally, Android Studio gives you a Lint warning whenever you use hardcoded text directly in XML, like:

```
xml  
  
android:hint="Email Address"
```

 Copy code

It warns you because Google recommends putting all user-visible strings in the `res/values/strings.xml` file instead — for proper localization and translation.

So the recommended way would be:



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

does it have a direct link with the visibility of the hint text
when i code it on android you can see the hint text
but run the app on my phone, you cannot see the text

Excellent observation — and no, 

`tools:ignore="HardcodedText"` does not affect visibility of the hint or any other UI text at runtime.

Let's unpack what's happening clearly:

👉 What `tools:ignore="HardcodedText"` actually does

It's only a development-time instruction for Android Studio's Lint checker.

- It affects *warnings in the editor*, not your app's behavior.
- It doesn't change how the app runs, compiles, or displays text.

So if the hint text is visible in the layout preview, but not visible when you run it on your phone, the cause is something else.

💡 Common real causes of "hint text not visible at runtime"

Here are the top reasons this happens:

1. Background color and hint color clash

- Example: white hint text on white background.
- Fix explicitly set the hint text color.

```
xml  
  
android:textColorHint="@color/gray"
```

 Copy code

OR:

```
xml  
  
android:textColorHint="#800000"
```

 Copy code

2. Custom drawable background hides it

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

The screenshot shows a dark-themed ChatGPT interface. In the message input field at the top, the user asks: "can you give me the colour code for darkish gray". Below this, ChatGPT responds with a list of color shades and their hex codes:

Shade	Hex Code	Description
Medium-dark gray	#666666	A balanced dark gray (great for text)
Dark gray	#555555	Slightly darker, still readable
Very dark gray	#333333	Almost black, good for subtle contrast
Neutral gray	#444444	Nice middle-ground dark tone

Below the table, there's a note about a checked checkbox: "Recommended for hint text". It suggests using the hex code #666666 for its "dark enough to read but clearly lighter than normal text (for a 'hint' look)". An example XML code snippet is shown:

```
xml
    android:textColorHint="#666666"
```

ChatGPT offers to show how to define this properly in `res/values/colors.xml`. At the bottom, there are upvote, downvote, and share buttons, along with a "Is this conversation helpful so far?" rating bar.

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.

The screenshot shows a dark-themed ChatGPT interface. The user asks: "how do I fix this error: Directory 'C:\Users\premc\AndroidStudioProjects\Plantopia' does not contain a Gradle build." ChatGPT provides a detailed explanation of what a Gradle build needs and how to fix it:

Directory 'C:\Users\premc\AndroidStudioProjects\Plantopia' does not contain a Gradle build.
A Gradle build's root directory should contain one of the possible settings files: settings.gradle, settings.gradle.kts, settings.gradle.dclt
may also contain one of the possible build files: build.gradle, build.gradle.kts, build.gradle.dclt.
To create a new Gradle build in this directory run 'gradle init'
For more information about the 'init' task, please refer to
https://docs.gradle.org/9.0-milestone-1/userguide/build_init_plugin.html in the Gradle documentation.
For more details on creating a Gradle build, please refer to
https://docs.gradle.org/9.0-milestone-1/userguide/tutorial_using_tasks.html in the Gradle documentation.

* Try:
> Run gradle init to create a new Gradle build in this directory.
> Run with --stacktrace option to get the stack trace.
> Run with --info or --debug option to get more log output.
> Get more help at <https://help.gradle.org>.

Ask Gemini!

At the bottom, ChatGPT notes: "This error means Android Studio cannot find a valid Gradle project inside." It shows a file tree for "Plantopia" containing "build.gradle" and "gradle-wrapper.properties". The message input field shows the command: "makefile".

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].