

Name: Madepelli Sadhana

Student ID: N01651380

Date: 09/01/25

Subject: Android Development LAB -1

Git installation

Download the git from <https://git-scm.com/>

The screenshot shows the official Git website (<https://git-scm.com/>). At the top, there's a search bar with the placeholder "Type / to search entire site...". Below the header, there's a brief introduction to Git: "Git is a **free and open source** distributed version control system designed to handle everything from small to very large projects with speed and efficiency." To the right of this text is a diagram illustrating Git's distributed nature with multiple repositories connected by arrows. Below the introduction, there are several sections: "About" (with a gear icon), "Documentation" (with a book icon), "Downloads" (with a download arrow icon), and "Community" (with a speech bubble icon). A sidebar on the left features a thumbnail for the book "Pro Git" by Scott Chacon and Ben Straub. On the right side, there's a prominent section for the "Latest source Release 2.47.1" with a "Release Notes (2024-11-25)" link and a "Download for Mac" button. Below this, there are links for "Mac GUIs", "Windows Build", "Tarballs", and "Source Code". At the bottom of the page, it says "Companies & Projects Using Git".

Once git is installed, you can verify its installation on your PC by using the git --version command.

```
● ○ ● madepelli — ubuntu@ip-172-31-27-176: ~/Docker-Zero-to-Hero/examples/first-docker-fil...
(base) madepellis-Air:~ madepelli$ git --version
git version 2.15.0
(base) madepellis-Air:~ madepelli$
```

Download github desktop from the following link <https://desktop.github.com/download/>

Let's get started!

Add a repository to GitHub Desktop to start collaborating



The screenshot shows the GitHub Desktop application interface. At the top, there is a search bar labeled "Filter your repositories" and a refresh button. Below it, a blue header bar contains the text "Create a Tutorial Repository...". Underneath, there are four main options: "Clone a Repository from the Internet...", "Create a New Repository on your Local Drive...", and "Add an Existing Repository from your Local Drive...". A "ProTip!" box at the bottom left suggests dragging and dropping a repository folder here. The background features a light gray gradient with small, faint icons of a bell, a person, a rocket, and a gear.

Filter your repositories

Create a Tutorial Repository...

Clone a Repository from the Internet...

Create a New Repository on your Local Drive...

Add an Existing Repository from your Local Drive...

ProTip! You can drag & drop an existing repository folder here to add it to Desktop

Then create a private repository named android_lab_1

The screenshot shows a GitHub repository page for 'android_lab_1'. The repository is private and owned by 'madepellisadhana'. It contains 7 commits, 1 branch, and 0 tags. The README file describes it as a repository to implement a lab 1 for android. The repository has 0 stars, 1 watcher, and 0 forks. It uses the MIT license and is written in Kotlin.

Copy the https link then Open the terminal and clone the android_lab_1 git repository by using git clone command

```
(base) madepellis-MacBook-Air:lab 1 madepelli$ git clone https://github.com/madepellisadhana/android_lab_1.git
Cloning into 'android_lab_1'...
Username for 'https://github.com': madepellisadhana@gmail.com
Password for 'https://madepellisadhana@gmail.com@github.com':
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (4/4), done.
(base) madepellis-MacBook-Air:lab 1 madepelli$
```

After successful cloning then add the screenshot to git by using git add . it will add all the files and subfolders in my current directory

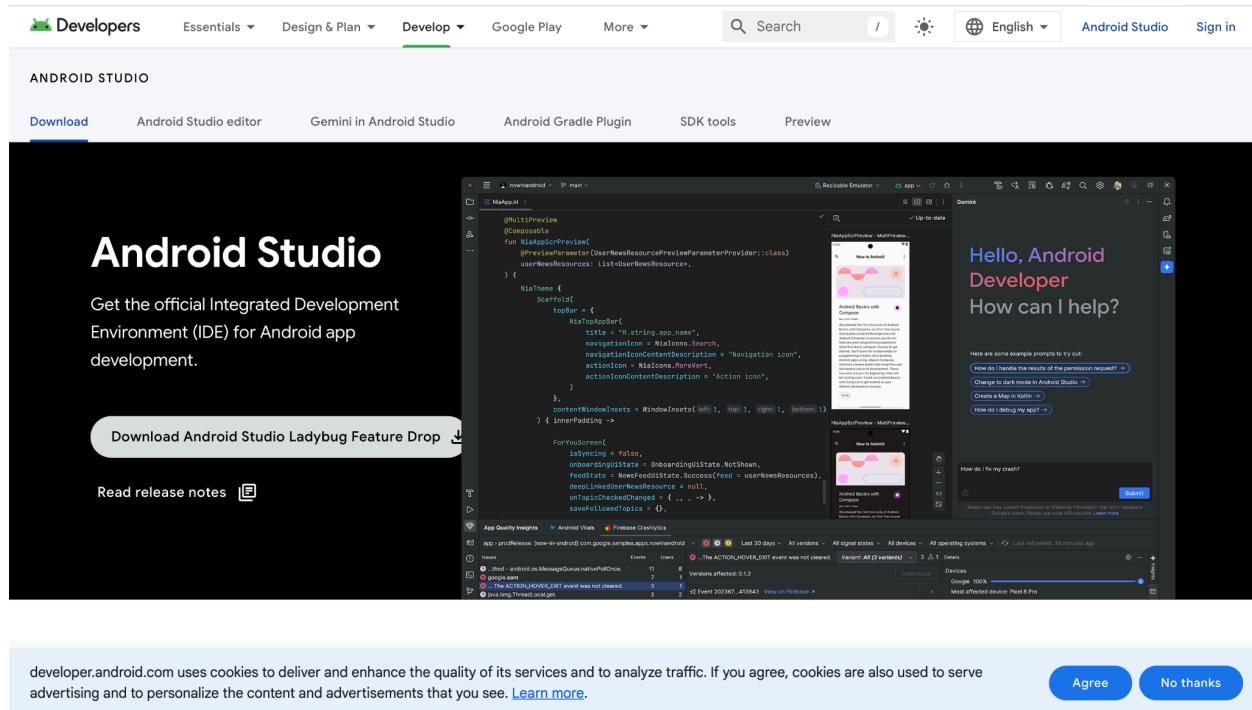
```

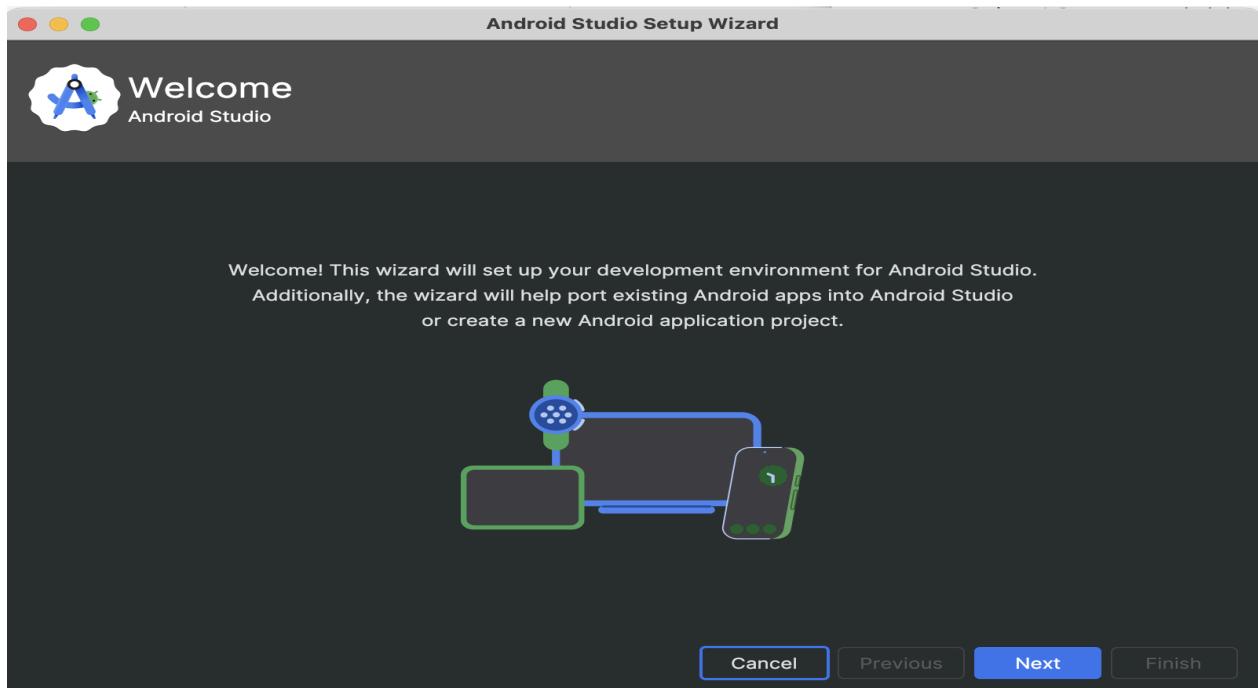
(base) madepellis-MacBook-Air:android_lab_1 madepelli$ git add .
(base) madepellis-MacBook-Air:android_lab_1 madepelli$ git commit -m "success"
[main 0450889] success
 4 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 screen shots/Screenshot 2025-01-09 at 12.10.53 PM.png
 create mode 100644 screen shots/Screenshot 2025-01-09 at 12.16.00 PM.png
 create mode 100644 screen shots/Screenshot 2025-01-09 at 12.32.07 PM.png
 create mode 100644 screen shots/Screenshot 2025-01-09 at 12.36.47 PM.png
(base) madepellis-MacBook-Air:android_lab_1 madepelli$ git push
Counting objects: 7, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (7/7), done.
Writing objects: 100% (7/7), 712.73 KiB | 24.58 MiB/s, done.
Total 7 (delta 0), reused 0 (delta 0)
To https://github.com/madepellisadhana/android_lab_1.git
 931cf92..0450889  main -> main
(base) madepellis-MacBook-Air:android_lab_1 madepelli$ 

```

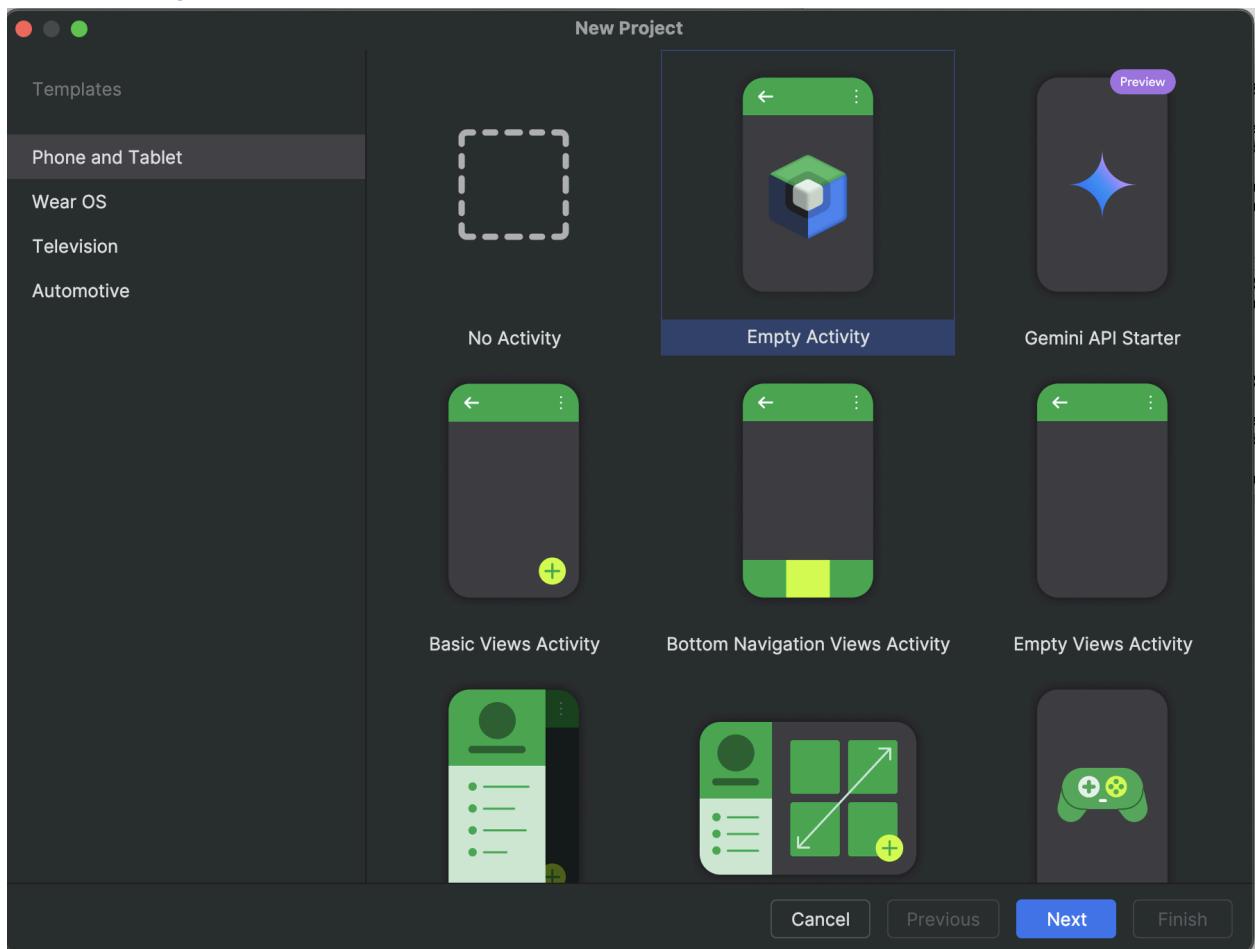
**Git commit -m “success” this command will commit the changes by a message called success
After committing, push your changes to your remote repository**

Task 1: Setting Up the Android Development Environment

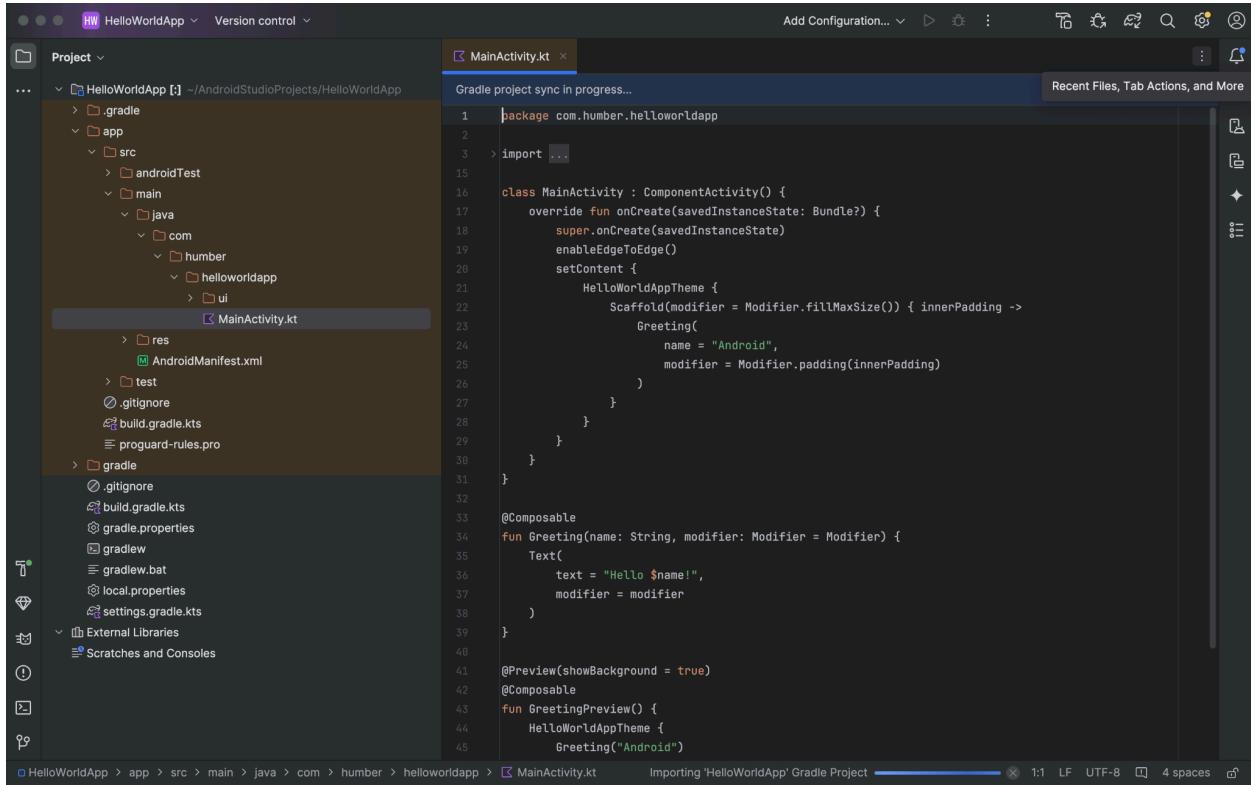




Task 2: Creating the "Hello Android" Application



After installation of Android studio select empty activity then click on next then give a name as HelloWorldApp and package name as com.humber.helloworldapp then click on finish



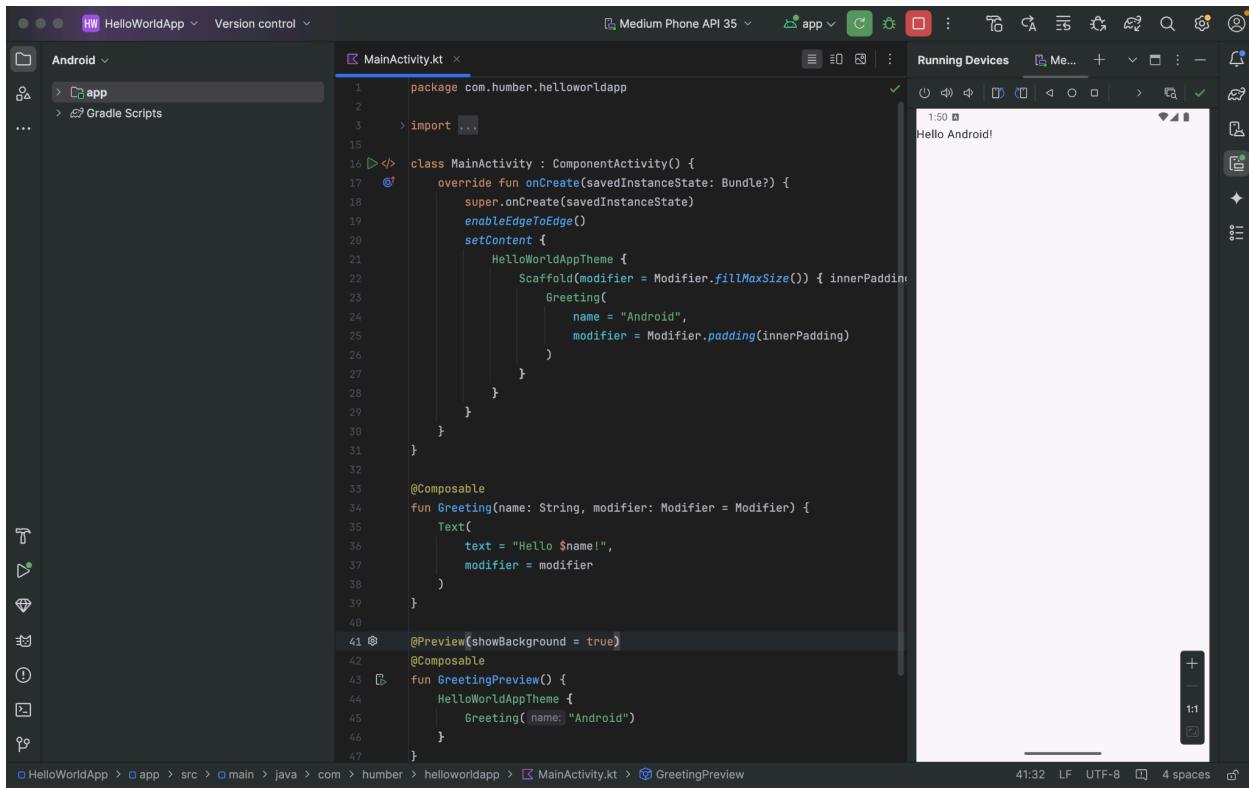
The screenshot shows the Android Studio interface with the following details:

- Project Bar:** HW HelloWorldApp, Version control.
- Project Tree:** HelloWorldApp [] -> AndroidStudioProjects/HelloWorldApp. It includes gradle, app (src, androidTest, main, java, com, humber, helloworldapp, ui), res, test, .gitignore, build.gradle.kts, proguard-rules.pro, gradle, .gitignore, build.gradle.kts, gradle.properties, gradlew, gradlew.bat, local.properties, settings.gradle.kts, External Libraries, and Scratches and Consoles.
- Main Activity File:** MainActivity.kt (selected in the tree). The code is as follows:

```
1 package com.humber.helloworldapp
2
3 import ...
4
5 class MainActivity : ComponentActivity() {
6     override fun onCreate(savedInstanceState: Bundle?) {
7         super.onCreate(savedInstanceState)
8         enableEdgeToEdge()
9         setContent {
10             HelloWorldAppTheme {
11                 Scaffold(modifier = Modifier.fillMaxSize()) { innerPadding ->
12                     Greeting(
13                         name = "Android",
14                         modifier = Modifier.padding(innerPadding)
15                     )
16                 }
17             }
18         }
19     }
20
21     @Composable
22     fun Greeting(name: String, modifier: Modifier = Modifier) {
23         Text(
24             text = "Hello $name!",
25             modifier = modifier
26         )
27     }
28
29     @Preview(showBackground = true)
30     @Composable
31     fun GreetingPreview() {
32         HelloWorldAppTheme {
33             Greeting("Android")
34         }
35     }
36 }
37
38
39
40
41
42
43
44
45 }
```

At the bottom, it says "Importing 'HelloWorldApp' Gradle Project".

To configure the Android virtual device open tools menu in android studio choose the hardware profile such as pixel 6 then click on finish . then run the code the output will be displayed as “Hello Android!”



To push the kotlin code to git I used git add . to add all the files in the present directory . Then git commit -m “files” this will commit the changes including the message as files. After committing the push will update the changes to the remote repository.

```

(base) madepellis-MacBook-Air:AndroidStudioProjects madepellis$ git add .
(base) madepellis-MacBook-Air:AndroidStudioProjects madepellis$ git commit -m "files"
[main 7aeed05] files
 48 files changed, 1156 insertions(+)
 create mode 100644 AndroidStudioProjects/HelloworldApp/.gitignore
 create mode 100644 AndroidStudioProjects/HelloworldApp/.idea/.gitignore
 create mode 100644 AndroidStudioProjects/HelloworldApp/.idea/codeStyles/Project.xml
 create mode 100644 AndroidStudioProjects/HelloworldApp/.idea/codeStyles/codeStyleConfig.xml
 create mode 100644 AndroidStudioProjects/HelloworldApp/.idea/deploymentTargetSelector.xml
 create mode 100644 AndroidStudioProjects/HelloworldApp/.idea/graddle.xml
 create mode 100644 AndroidStudioProjects/HelloworldApp/.idea/kotlin.xml
 create mode 100644 AndroidStudioProjects/HelloworldApp/.idea/migrations.xml
 create mode 100644 AndroidStudioProjects/HelloworldApp/.idea/misc.xml
 create mode 100644 AndroidStudioProjects/HelloworldApp/.idea/resources/configurations.xml
 create mode 100644 AndroidStudioProjects/HelloworldApp/app/.gitignore
 create mode 100644 AndroidStudioProjects/HelloworldApp/app/build.gradle.kts
 create mode 100644 AndroidStudioProjects/HelloworldApp/app/proguard-rules.pro
 create mode 100644 AndroidStudioProjects/HelloworldApp/app/src/androidTest/java/com/humber/helloworldapp/ExampleInstrumentedTest.kt
 create mode 100644 AndroidStudioProjects/HelloworldApp/app/src/main/AndroidManifest.xml
 create mode 100644 AndroidStudioProjects/HelloworldApp/app/src/main/java/com/humber/helloworldapp/MainActivity.kt
 create mode 100644 AndroidStudioProjects/HelloworldApp/app/src/main/java/com/humber/helloworldapp/color/Color.kt
 create mode 100644 AndroidStudioProjects/HelloworldApp/app/src/main/java/com/humber/helloworldapp/ui/theme/Theme.kt
 create mode 100644 AndroidStudioProjects/HelloworldApp/app/src/main/java/com/humber/helloworldapp/ui/theme/type.kt
 create mode 100644 AndroidStudioProjects/HelloworldApp/app/src/main/res/drawable/ic_launcher_background.xml
 create mode 100644 AndroidStudioProjects/HelloworldApp/app/src/main/res/drawable/ic_launcher_foreground.xml
 create mode 100644 AndroidStudioProjects/HelloworldApp/app/src/main/res/mipmap-anydpi-v26/ic_launcher.xml
 create mode 100644 AndroidStudioProjects/HelloworldApp/app/src/main/res/mipmap-anydpi-v26/ic_launcher_round.xml
 create mode 100644 AndroidStudioProjects/HelloworldApp/app/src/main/res/mipmap-anydpi-v26/ic_launcher_webp
 create mode 100644 AndroidStudioProjects/HelloworldApp/app/src/main/res/mipmap-anydpi-v26/ic_launcher_webp
 create mode 100644 AndroidStudioProjects/HelloworldApp/app/src/main/res/mipmap-mdpi/ic_launcher.webp
 create mode 100644 AndroidStudioProjects/HelloworldApp/app/src/main/res/mipmap-mdpi/ic_launcher_round.webp
 create mode 100644 AndroidStudioProjects/HelloworldApp/app/src/main/res/mipmap-xhdpi/ic_launcher.webp
 create mode 100644 AndroidStudioProjects/HelloworldApp/app/src/main/res/mipmap-xhdpi/ic_launcher_round.webp
 create mode 100644 AndroidStudioProjects/HelloworldApp/app/src/main/res/mipmap-xxhdpi/ic_launcher_webp
 create mode 100644 AndroidStudioProjects/HelloworldApp/app/src/main/res/values/colors.xml
 create mode 100644 AndroidStudioProjects/HelloworldApp/app/src/main/res/values/themes.xml
 create mode 100644 AndroidStudioProjects/HelloworldApp/app/src/main/res/xml/backup_rules.xml
 create mode 100644 AndroidStudioProjects/HelloworldApp/app/src/main/res/xml/data_extraction_rules.xml
 create mode 100644 AndroidStudioProjects/HelloworldApp/app/src/test/java/com/humber/helloworldapp/ExampleUnitTest.kt
 create mode 100644 AndroidStudioProjects/HelloworldApp/app/src/main/resources/gradle.properties
 create mode 100644 AndroidStudioProjects/HelloworldApp/app/src/main/resources/gradle-wrapper.properties
 create mode 100644 AndroidStudioProjects/HelloworldApp/app/src/main/resources/libs.versions.toml
 create mode 100644 AndroidStudioProjects/HelloworldApp/app/src/main/resources/wrapper.gradle-wrapper.jar
 create mode 100644 AndroidStudioProjects/HelloworldApp/app/src/main/resources/wrapper.gradle-wrapper.properties
 create mode 100755 AndroidStudioProjects/HelloworldApp/gradlew
 create mode 100644 AndroidStudioProjects/HelloworldApp/gradlew.bat
 create mode 100644 AndroidStudioProjects/HelloworldApp/settings.gradle.kts
(base) madepellis-MacBook-Air:AndroidStudioProjects madepellis$ git push
To https://github.com/madepellis/hana/android_lab_1.git
 3cab9f8..7aeed05 main->main
(base) madepellis-MacBook-Air:AndroidStudioProjects madepellis$ 

```

The GitHub repository link:

https://github.com/madepellisadhana/android_lab_1.git

Observations:

Task 1:

I successfully obtained and set up Android Studio from the official site. The installation process was simple since I adhered to the setup wizard. The essential elements such as the Android SDK, SDK Platform-Tools, and Emulator were automatically chosen for installation, simplifying the process. I successfully set up an Android Virtual Device (AVD) using a Pixel 6 hardware profile and the most recent stable API level. Once the setup was finished, the emulator became ready for operation. This allowed me to grasp how Android Studio operates with emulators for testing purposes.

Task 2:

Initiating a new project in Android Studio was simple. I picked the "Empty Activity" template and opted for Kotlin as the programming language. Adjusting the minimum SDK to the most recent stable API level ensured support for the latest Android functionalities. Running app was successful. Then "Hello Android!" is displayed

Task 3:

I created a private GitHub repository named HelloWorldApp . Then I committed all the screenshots and pushed them into the git repository. By this class I learnt how to push the code to a remote repository. And I invited the instructor as a collaborator to the repository so that the project is accessible for review.

The screenshot shows the GitHub repository settings page for 'madepellisadhana / android_lab_1'. The 'Settings' tab is selected. On the left, there's a sidebar with sections like General, Access (Collaborators selected), Code and automation, Security, and Integrations. The main area is titled 'Who has access' and shows that it's a 'Private repository'. It lists 'DIRECT ACCESS' with 1 user (vitalii.bohudskyi@humber.ca) and a 'Manage' button. Below this is a 'Manage access' section with a search bar and a pending invite for vitalii.bohudskyi@humber.ca. At the bottom, there are navigation links for 'Previous' and 'Next'.