**Introduction to Mobile Application and app development.**

**St10447706.**

**Dr Rodney Mushininga.**

# Table of Contents:

Assignment 2 break down............page 1-2

Purpose Of App.............................page 3

Welcome Page..............................page 4

Main Screen..................................page 5

Screenshots................................. page 6-20

Read Me File............................... page 21

References................................... page 22

**THE Tamagotchi App.**

**The main purpose of the app is to have an output of a virtual pet that does 3 things.**

**The Virtual pet will:**

**Play**

**Eat**

**Clean.**

**This will mainly depend on the user clicking a button that will result in the action of the virtual app.**

**The Assignment expects the student to code the following:**

**An app that will rely on an ‘ OnClickListener’.**

**The user is expected to go on Android Studio or the virtual machine on Azure lab services. I chose the Android Studio app because of the speed of the app, in comparison to the virtual machine.**

**The student is required to display two screens on their app.**

**The first screen being the welcome page and the second screen being the main page where all the action will take place.**

**The student is expected to have a virtual pet that will perform functions of eating, to fulfil its hunger, playing to fulflil their happiness and cleaning itself.**

**How the assignment was orchestrated.**

**Well, through the usage of the Android Studio app, i went on it and opened the empty views activity.**

**Thereafter, i named my file and was done with the first screen. I named it main activity,**

**The second part of the work i did, was that i right clicked on the ‘ File’.**

**With the multiple options i saw i clicked on the ‘ Empty Activity’.**

**By clicking that, i was certain that my second screen was completed.**

**I named my second screen bubbles.**

**I later went to Micrososft , then Google to get my pictures.**

**I right clicked the pictures, saved them and copied them individually in my Drawables.**

**I named them again once copying and pasting them in their repectful places.**

**Each picture was allocated a screen depending on their functions.**

Welcome Page.

With my welcome page, i did the following:

I used my first screen to do code my first page.

I named my virtual pet Cocoa.

I then dragged the Textview, start Button, and image view and dropped them.

I constrained them and changed the font to a condensed monospace.

Moreover, i changed the textsize and Background to how i wanted it to be .

I then went to the main activity of the welcome page to start coding the welcome page.

I declared the variables for the Textview, button and image and re-named them so that it could be refactored and identified by the main activity.

For that i used the findViewById .

Lastly, for my welcome page, i put the indent to that the welcome page could move to the second screen after clicking the start button.

Main Screen

Still using Kotlin Language i started with my main page/ second screen.

I imported all variables including:

Button.

Imageview.

ProgressBar.

TextView.

Right after ,i made a Private Function, so that i could use my Progress Bar.

On my Activity Main, i dragged and dropped 3 of each;

1. Button
2. Progress bar
3. Textview

I changed fonts and background and moved on to my Main Activity.

I declared all functions using findViewById and onClickListener.

I decalared all my pictures, the clean, eating and playing using Drawables .

And lastly, i intent so that both screens could work together.

I used YouTube so that i could know how exactly to use the progress bar in the app on the following platform:

<https://abhiandroid.com/ui/progressbar#gsc.tab=0> accessed on 28 April 2024.

I then used a handler . The use of a handler is to:

(1) to schedule messages and runnables to be executed at some point in the future; and (2) to enqueue an action to be performed on a different thread than your own.

I proceeded to go to Google to gain knowledge on how to use and declare my hangler on the following website:

<https://developer.android.com/reference/android/os/Handler> accessed on 28 April 2024.

Lastly i assigned the progress bars to a value of certain numbers to make sure that when the either of the buttons are clicked the progress bar move up to a certain point until they stop .

I did all of that for all my buttons and later increased the size of the bars so that it would be more visable .

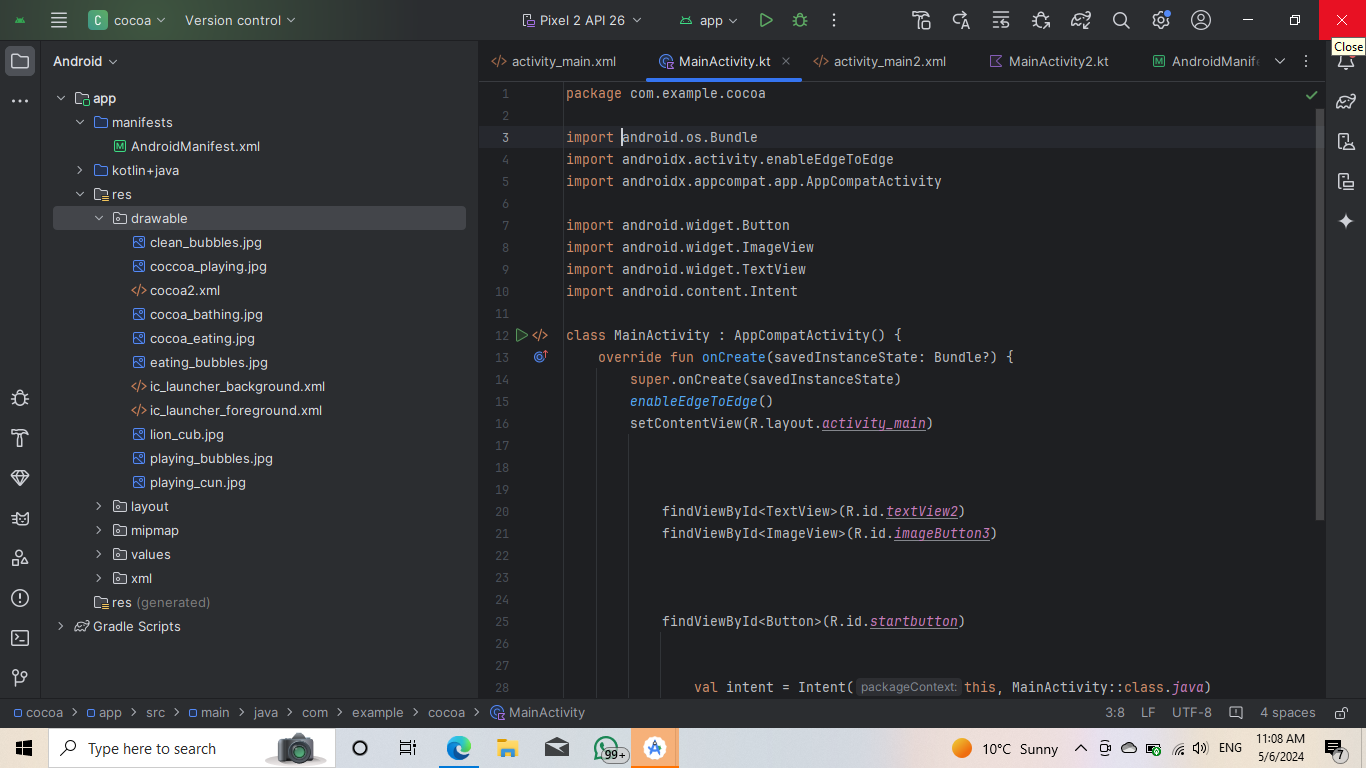
Lastly, with help from the module manual i went onto the multiplication table task that we had as an ice to learn properly and have more guidance on how to indent and successfully link my welcome screen to my main screen and both my screens work hand in hand.

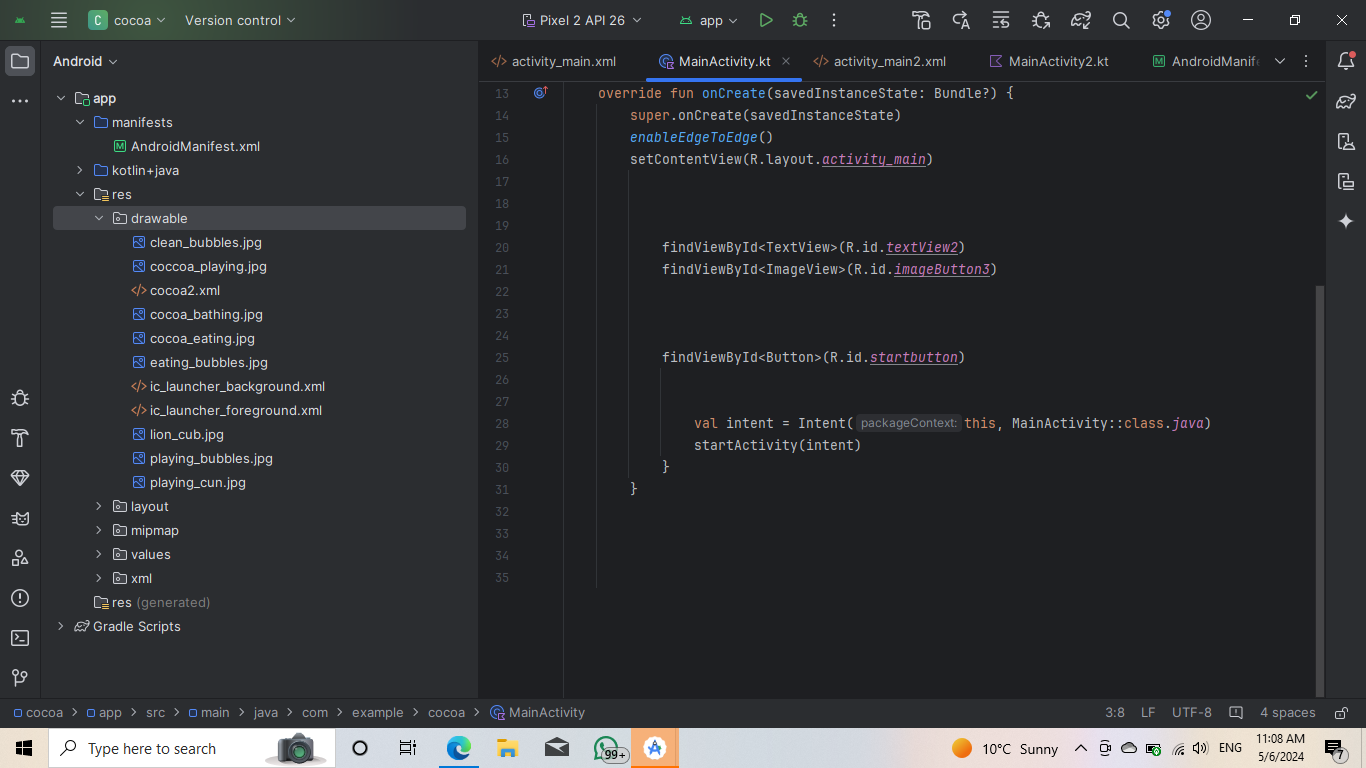
This was the end of the code for me and my app began to run succesfully.

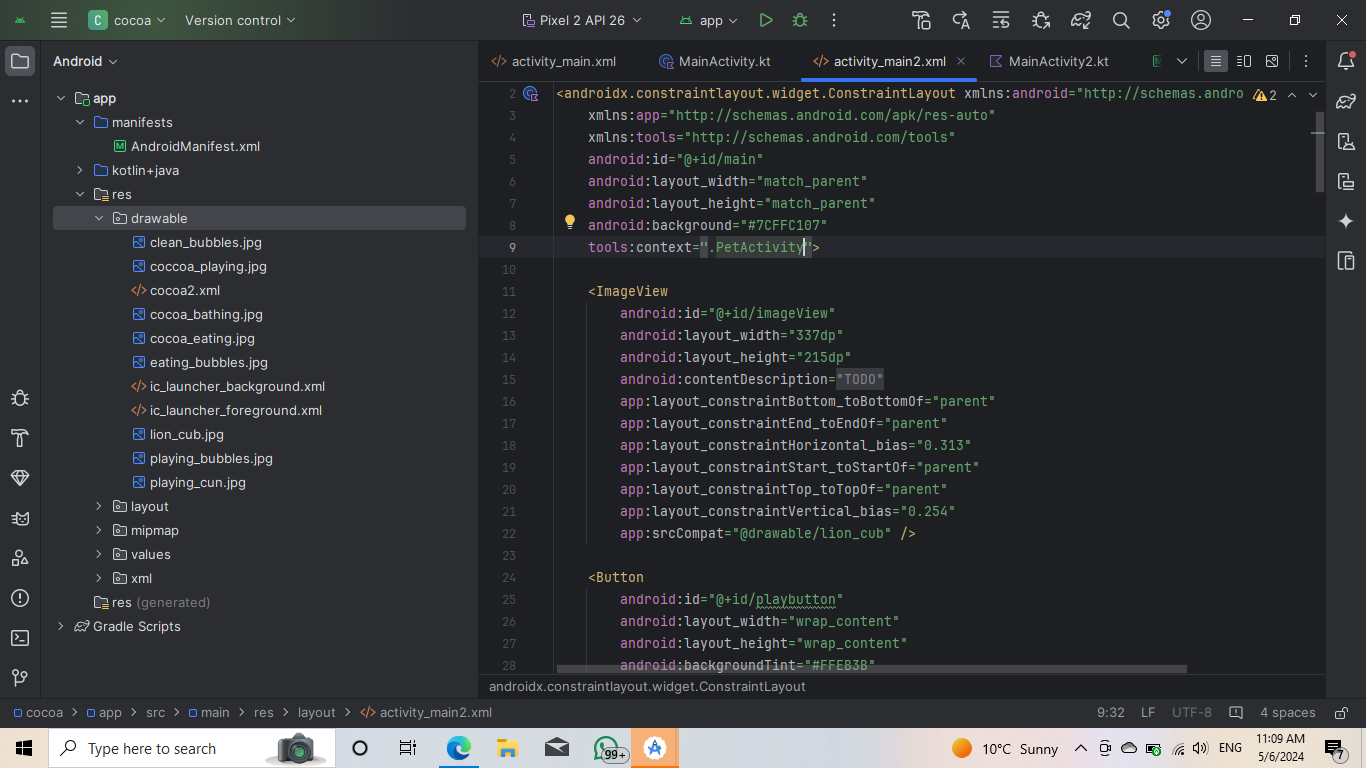
But for safety reasons i decided to debug my app and to also make sure that the emulator i used was the correct one, it being the pixel 6 and i used the oreo version .

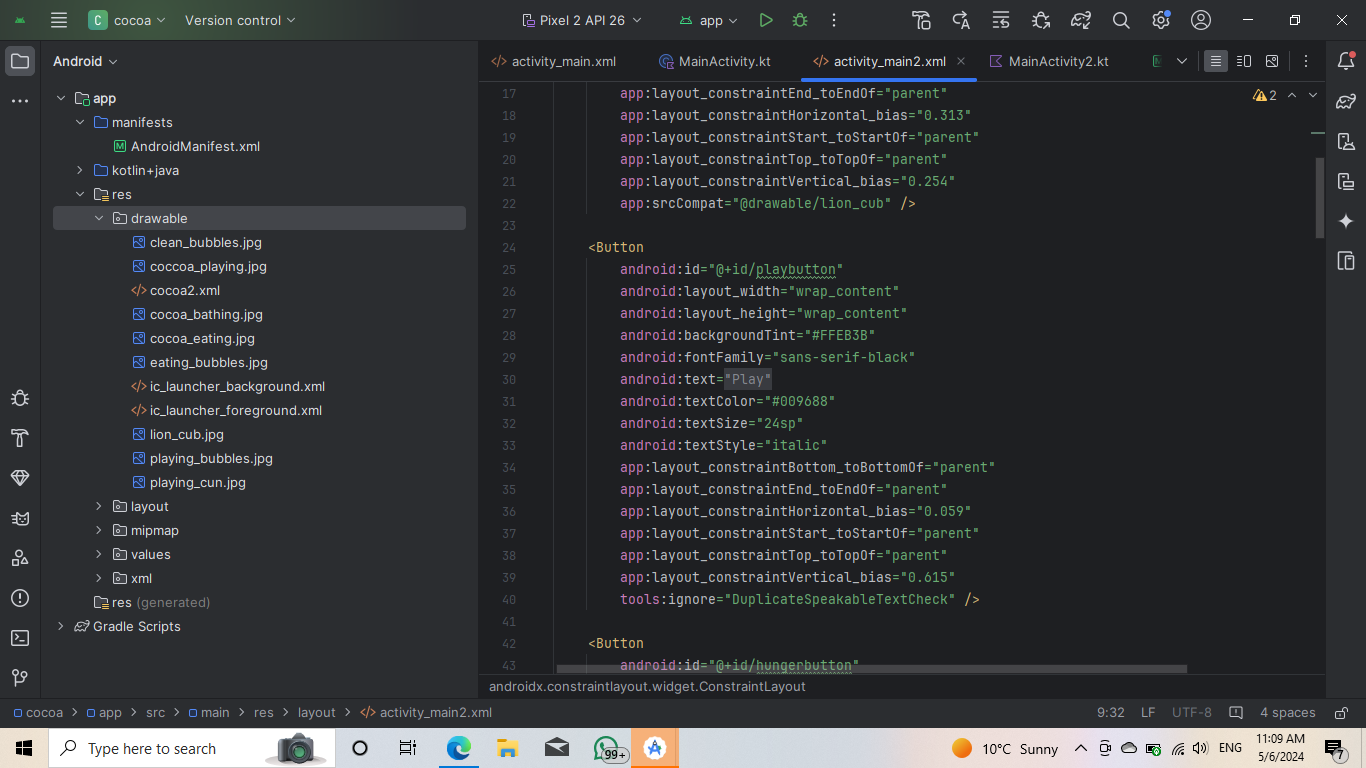
Now i was done.

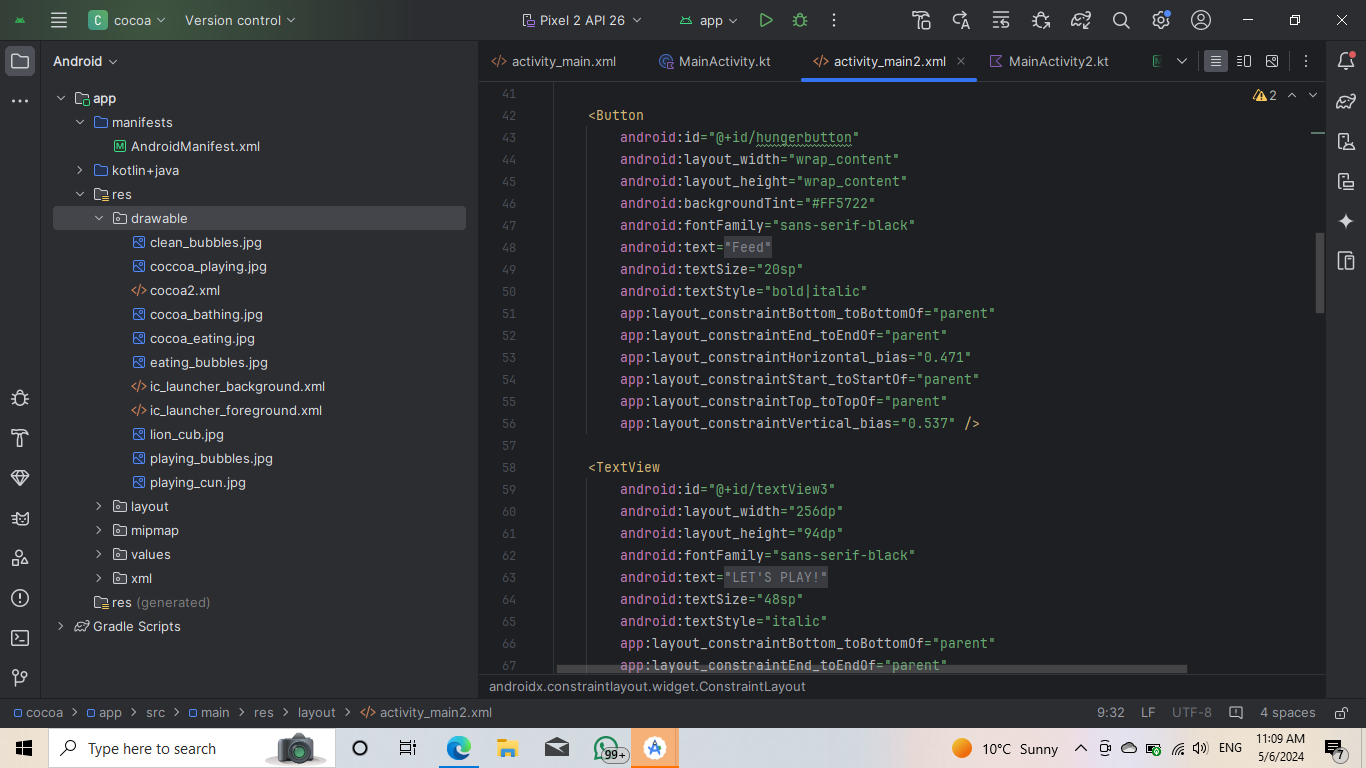
The images are as follows:

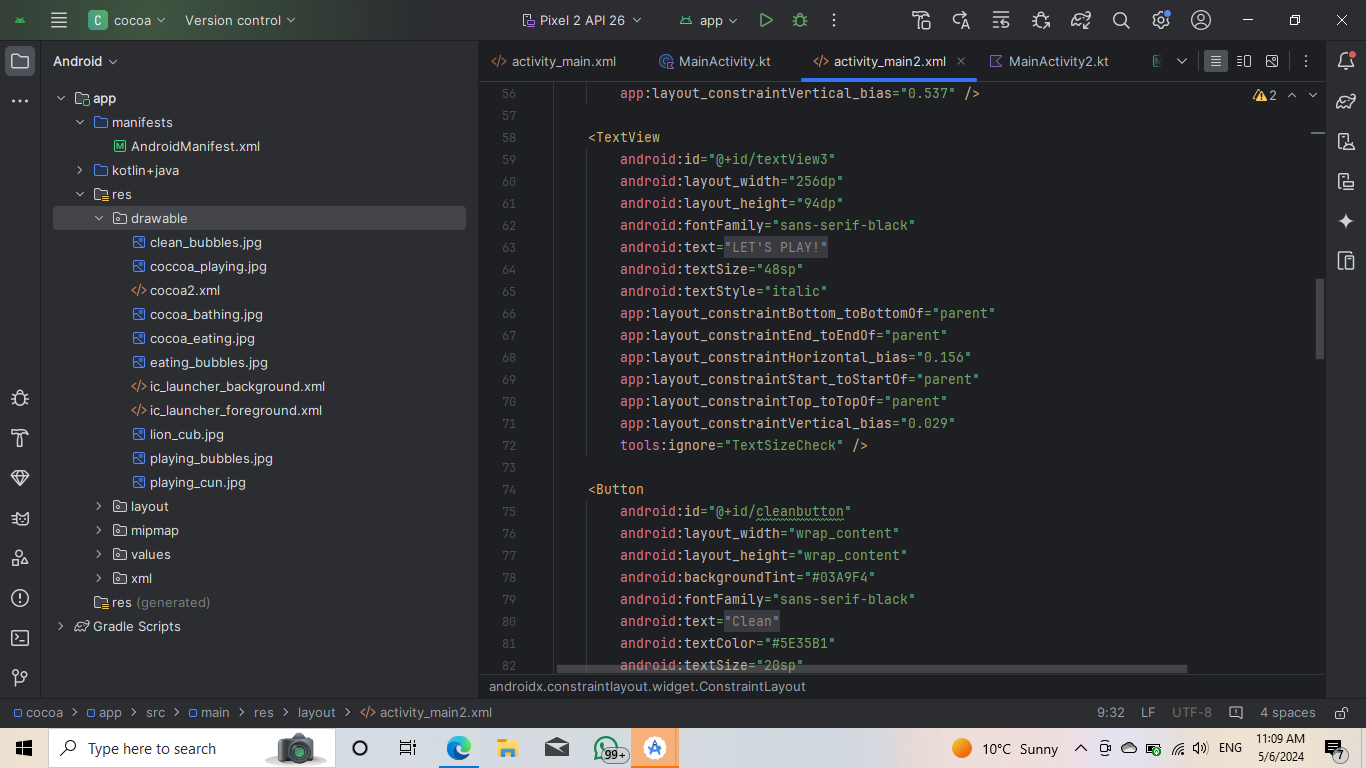


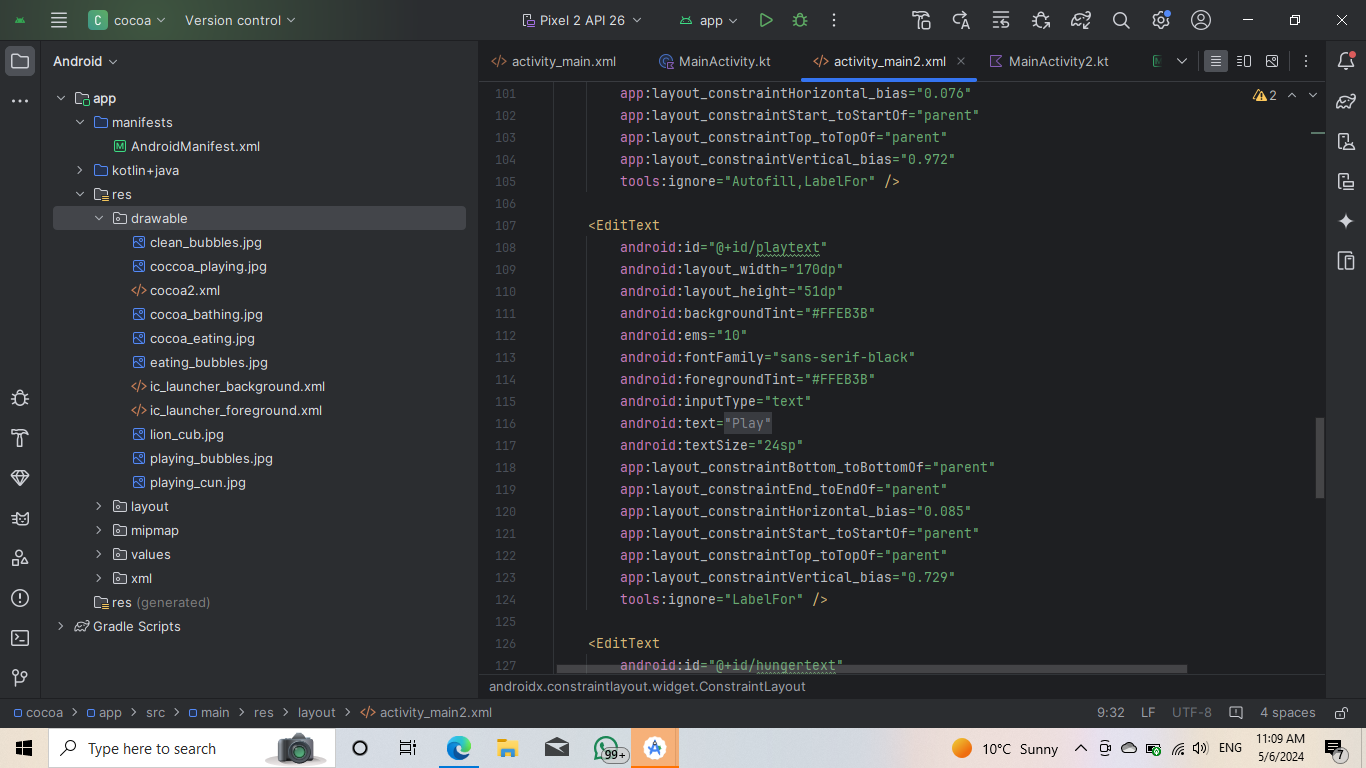


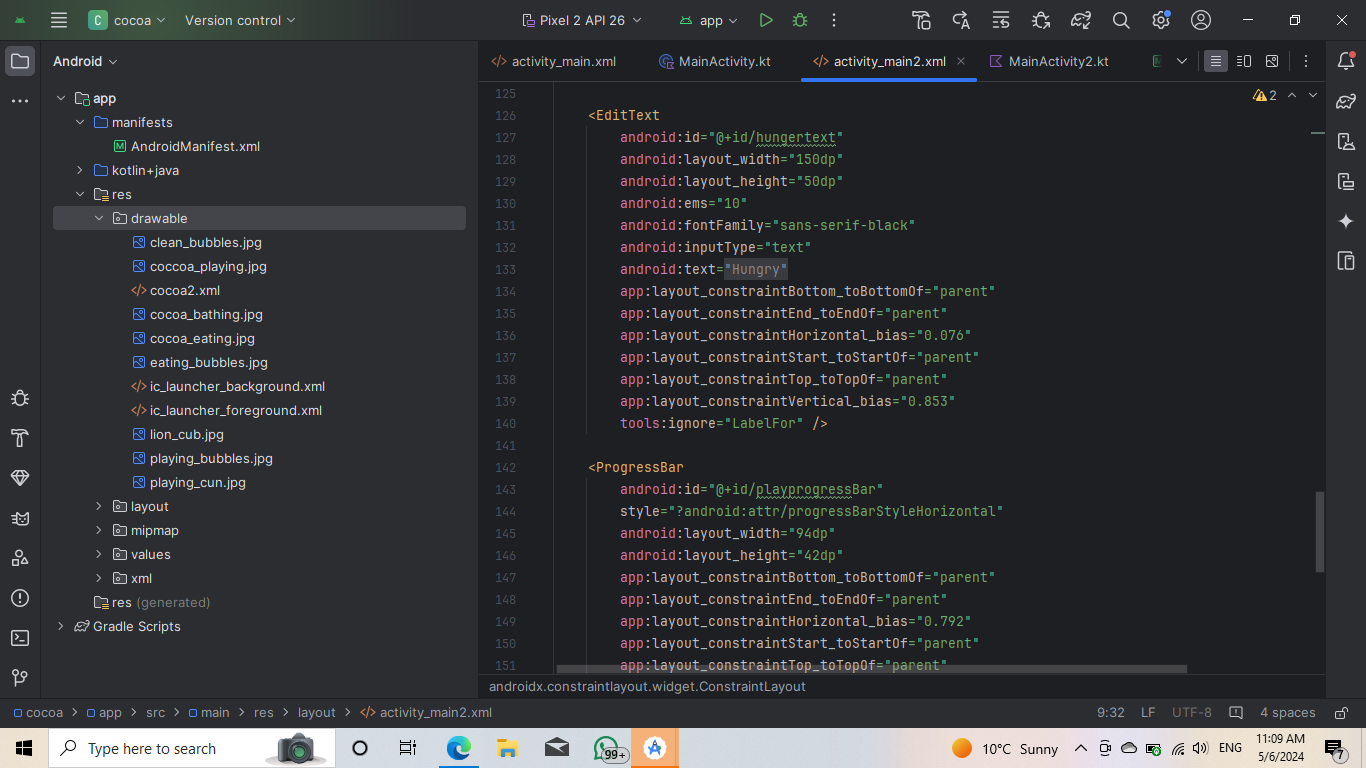


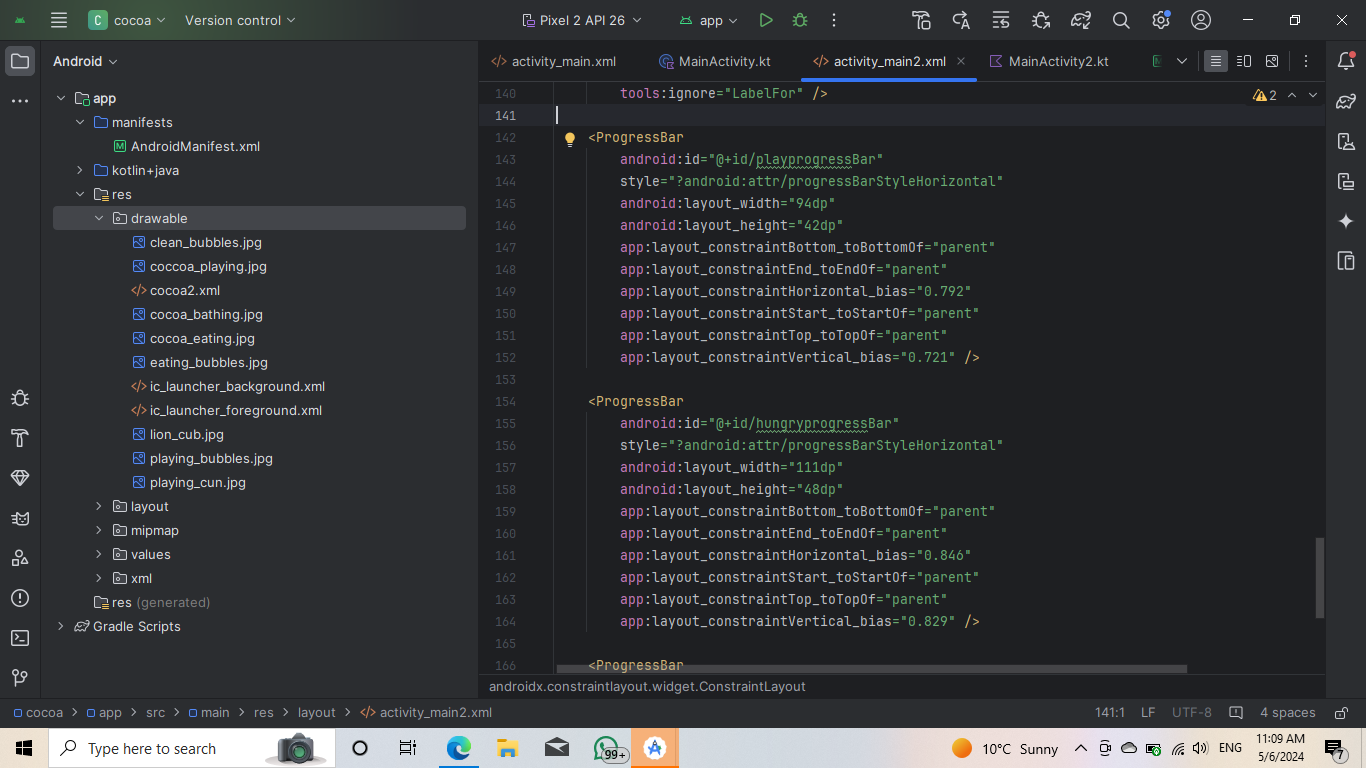


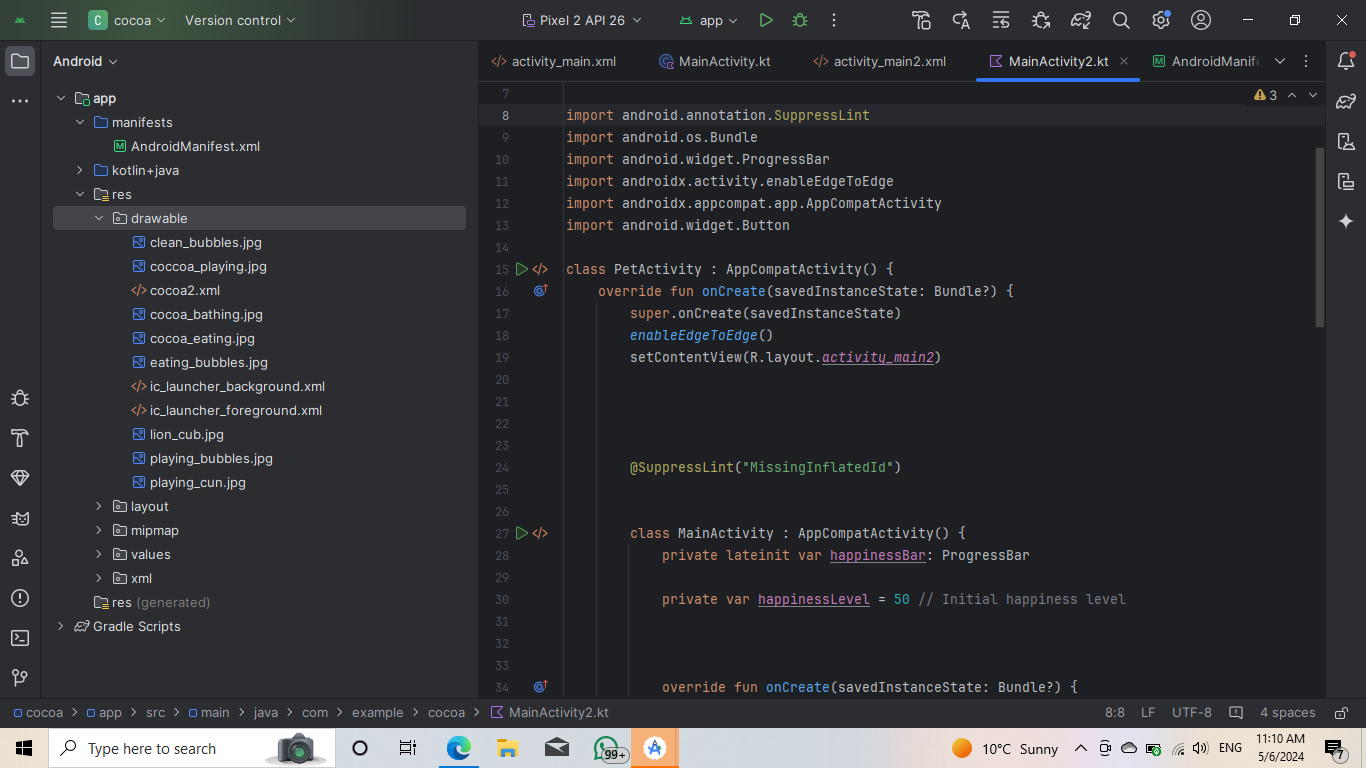


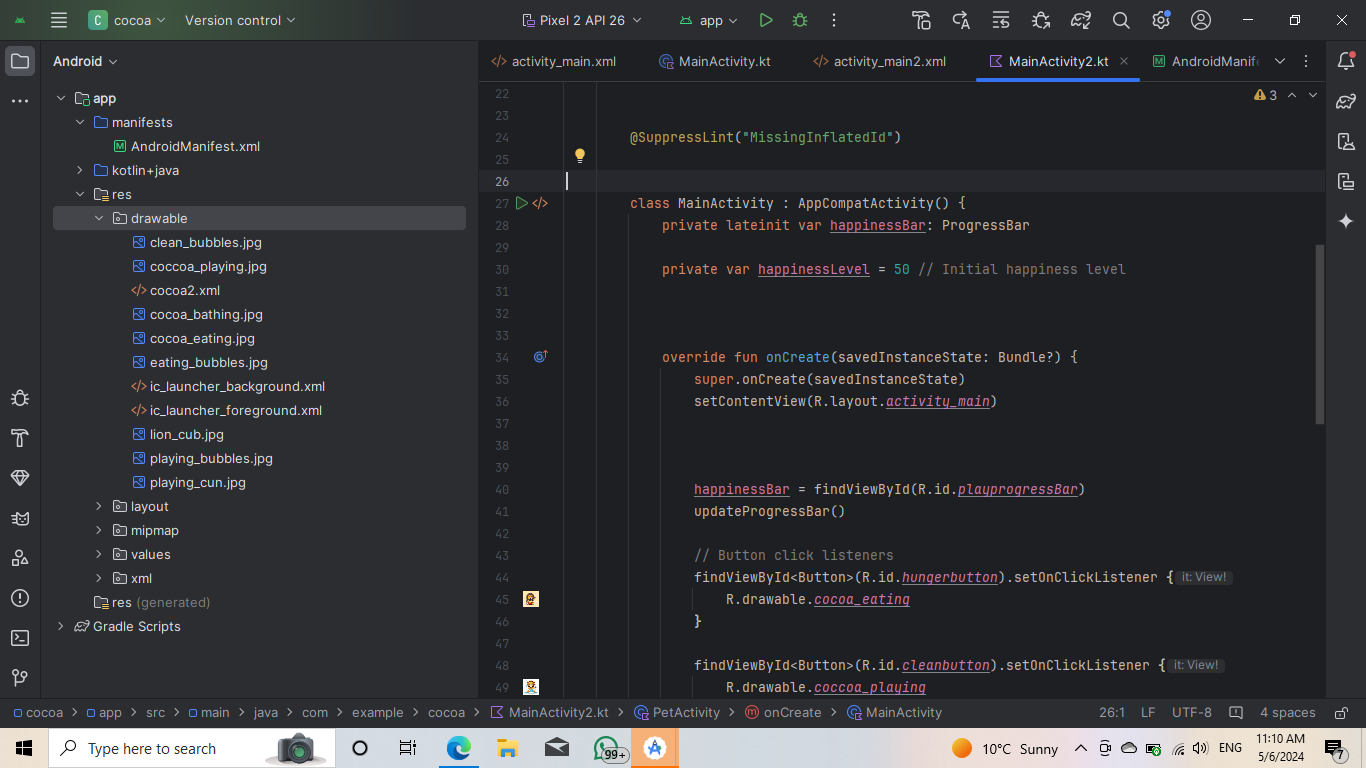


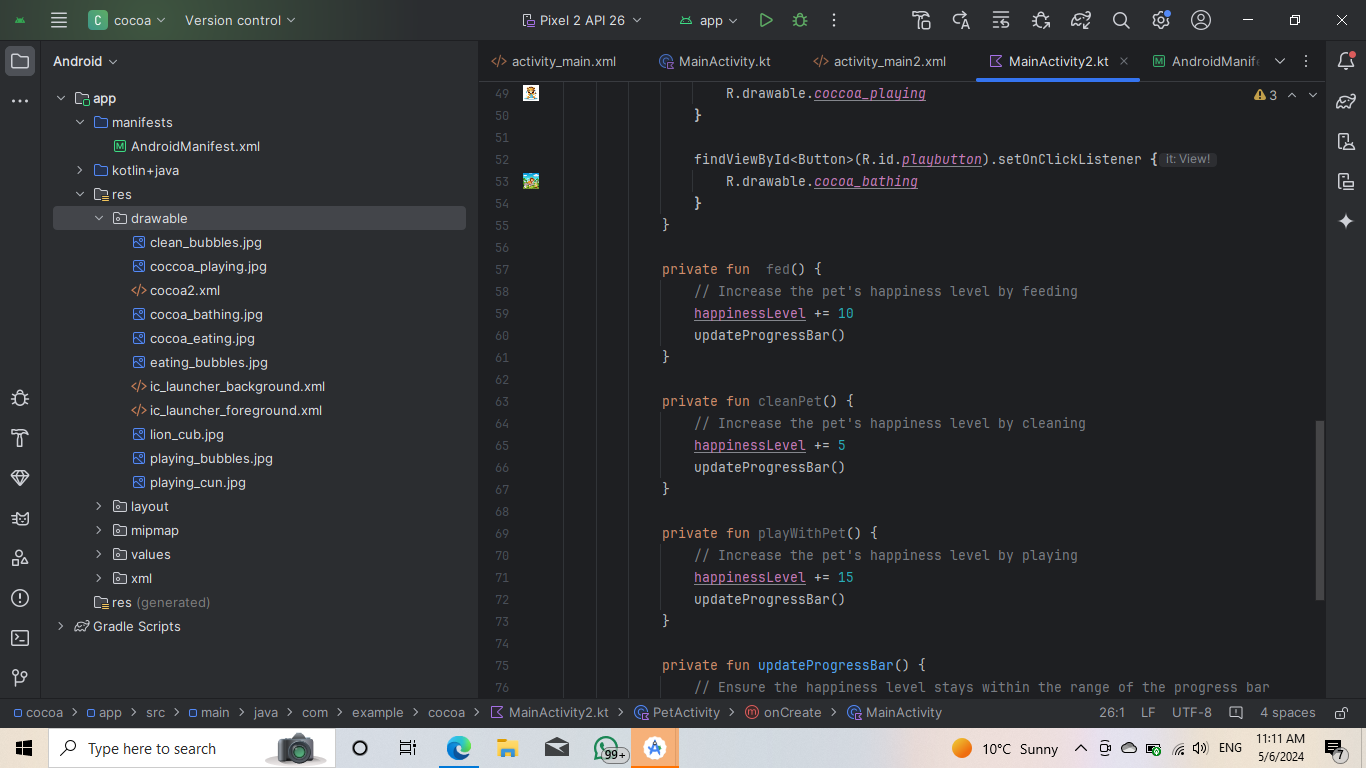


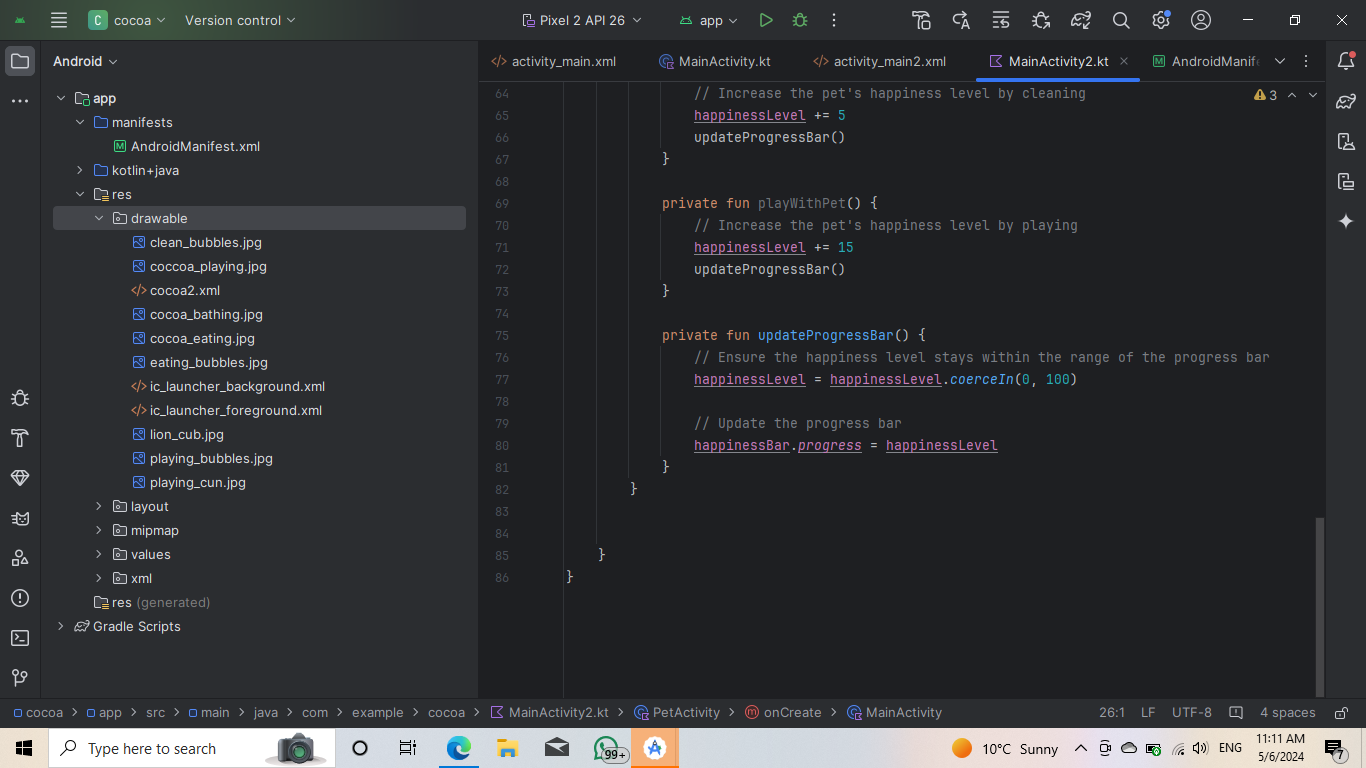


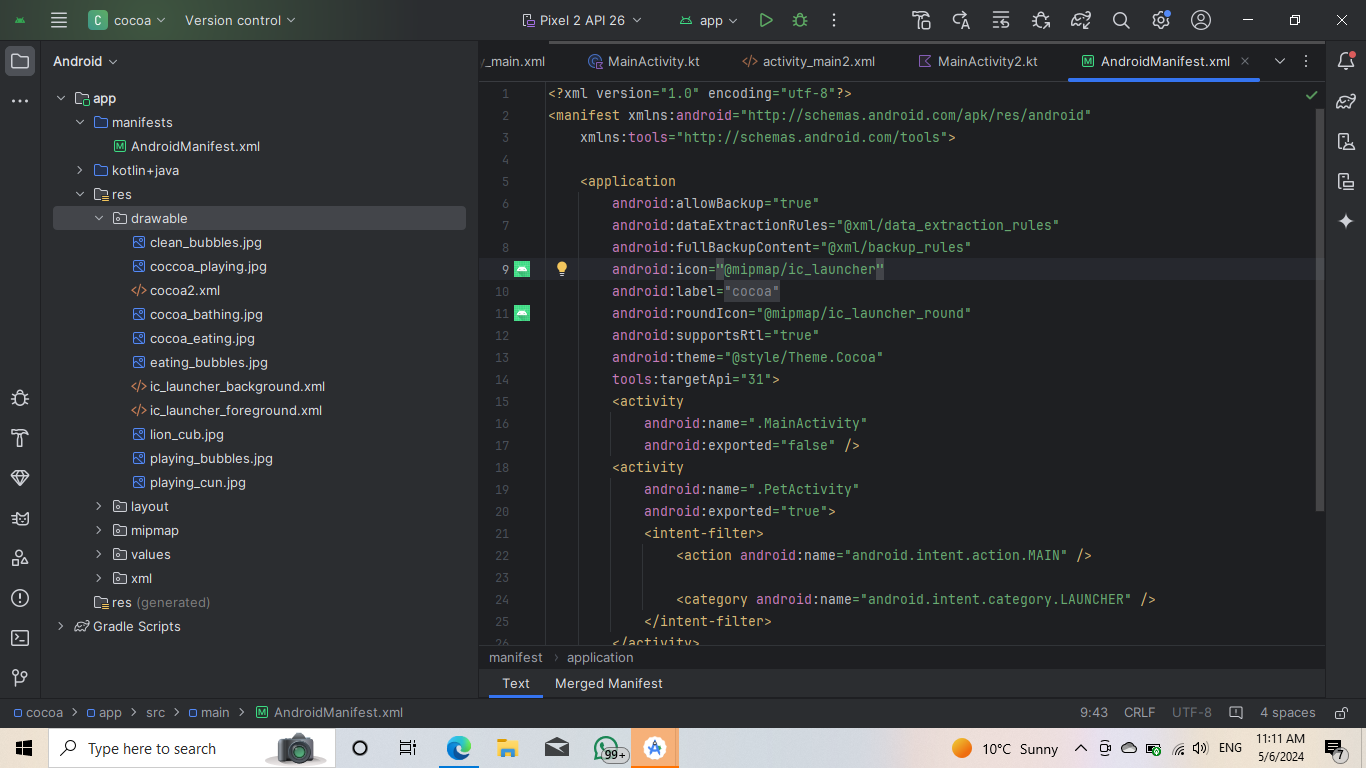


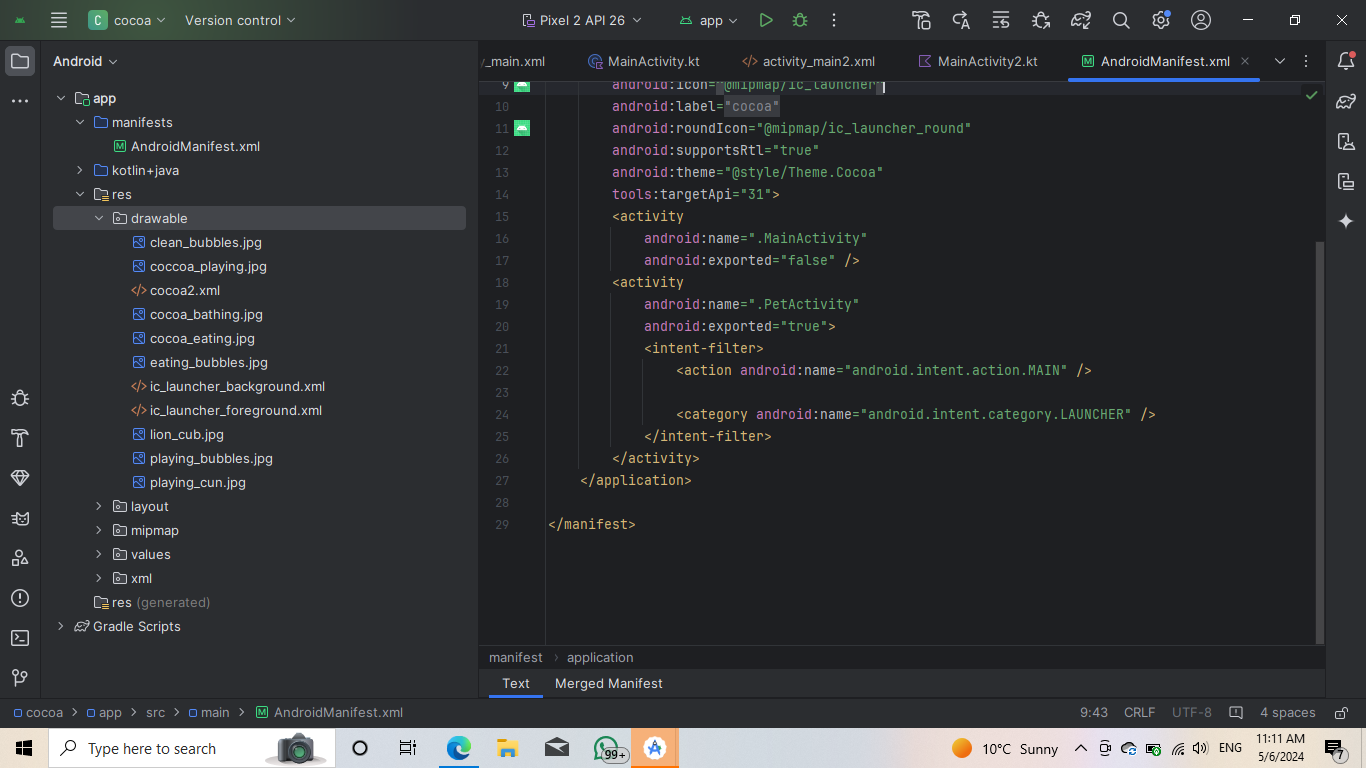












The Read Me File is as follows:

How to run your emulator:

1. Start BlueStacks

2. Click on the gear icon to open the settings

3. Look for Advanced.

4. You find enable ADB (Android Debug Bridge)

5. You will see a port number , take note of it. It is normally 5555 or 5554

6. Doulbe click on the emulator script on your desktop ( it has two little gears on it)

7. Enter your port number to connect the emulator the ADB.

8. You can now open Android Studio / Visual Studio code and start coding and run your app.

Enjoy mobile dev 😊

## References:

<https://developer.android.com/reference/android/os/Handler> accessed on April 28 2024.

[53 Animated Animals ideas | cartoon animals, animated animals, cartoon (pinterest.com)](https://za.pinterest.com/egtegt/animated-animals/) accessed on May 6 2024

<https://developer.android.com/reference/android/os/Handler> accessed on April 28 2024.

[How to screen record in Windows 10 or 11 | ZDNET](https://www.zdnet.com/article/how-to-screen-record-in-windows-10-or-11/) accessed on May 6 2024