PRACTICAL-1

Introduction to Android, Introduction to Android Studio IDE, Application Fundamentals: Creating a project, Android components, Activities, Services, Content providers, Broadcast Receivers, Interface Overview, Creating Android Virtual device, USB debugging mode, Android Application Overview. Simple "Hello World" program.

Introduction to Android

- Android is an open-source, Linux-based mobile operating system developed by Google. It is designed primarily for touchscreen mobile devices such as smartphones and tablets but also extends to other devices like smart TVs, wearables, and cars. Android offers a rich user interface (UI) and provides access to a broad range of hardware features and APIs, making it the most widely used mobile OS globally.
- Android applications are primarily written in **Java** or **Kotlin** and run on the Android Runtime (ART) which uses virtual machines optimized for mobile devices.

Introduction to Android Studio IDE

Android Studio is the official Integrated Development Environment (IDE) for Android app development. Built on top of IntelliJ IDEA, Android Studio provides a comprehensive suite of tools to build Android applications, including:

- **Code editor**: Supports features like code completion, refactoring, and linting.
- **Visual layout editor**: Lets developers design UI with drag-and-drop functionality.
- **Android Emulator**: Simulate Android devices on your computer for testing purposes.
- **SDK Manager**: Allows you to install or update the Android SDK and libraries.
- **Gradle build system**: Manages dependencies, build configurations, and app packaging.

Application Fundamentals:

1. Creating a Project

When you create a new Android project in Android Studio, the IDE sets up a structure that contains all the necessary files and folders. Here are the key components:

1. **Project Configuration Files**:
    - build.gradle: Contains project-level and app-level configuration, dependencies, and build settings.
    - AndroidManifest.xml: Declares important metadata about the app, such as permissions, components (activities, services), and features.
2. **App Code and Resources**:
    - **MainActivity.java (or MainActivity.kt)**: The main entry point of your application.

    - **res/ folder**: Contains resources such as layouts (XML files), drawables, and values (colors, strings, dimensions).
3. **Gradle**: Android uses Gradle as its build automation system. This simplifies the build process and integrates dependency management and tasks such as compiling and packaging the app.

2. Android Components

Android applications are built using four main components:

1. **Activities**: An activity represents a single screen in your application. Each activity is responsible for managing the UI and handling user interactions. For example, the screen that shows a list of contacts or the home screen of the app.

2. **Services**: A service is a component that runs in the background to perform long-running operations (such as downloading files or playing music) without a direct user interface.

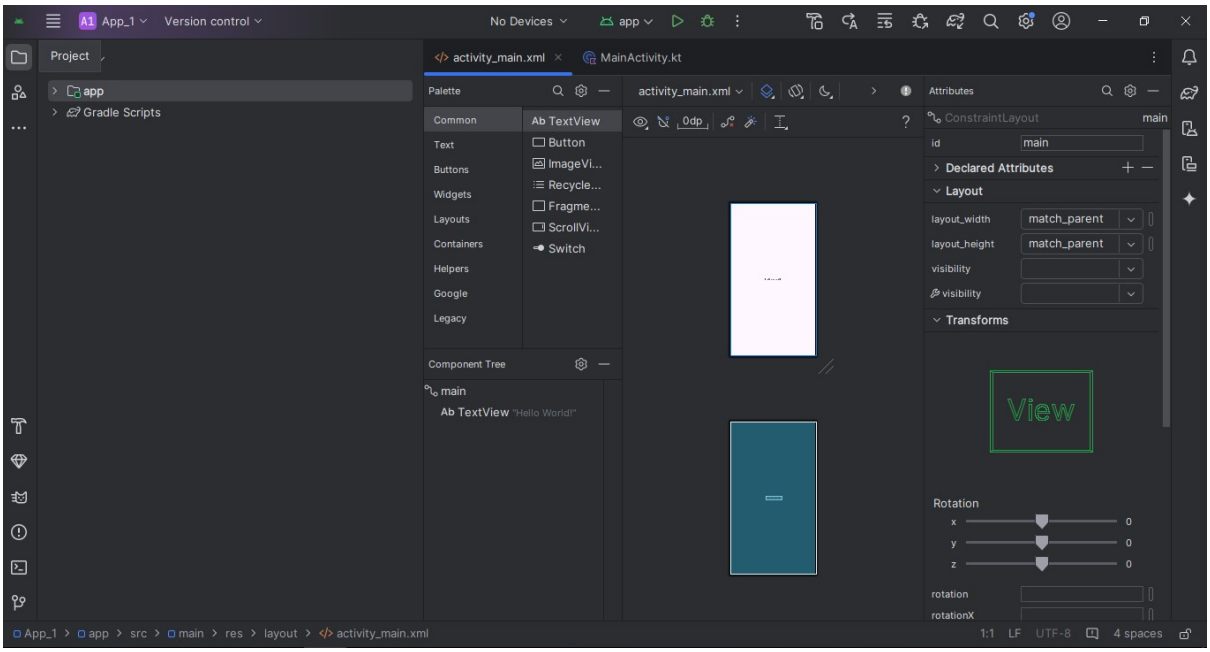
3. **Content Providers**: Content providers allow your app to share data with other apps and manage access to structured data like contacts, media, or files.

4. **Broadcast Receivers**: Broadcast receivers listen for and respond to broadcast messages from other apps or the system. These messages might be system events like battery low, Wi-Fi status change, or incoming SMS.

3. Interface Overview

- Android provides various UI components like **TextViews**, **Buttons**, **ImageViews**, **EditTexts**, and **RecyclerViews** to build user interfaces. These components are typically defined in XML files in the `res/layout` directory.

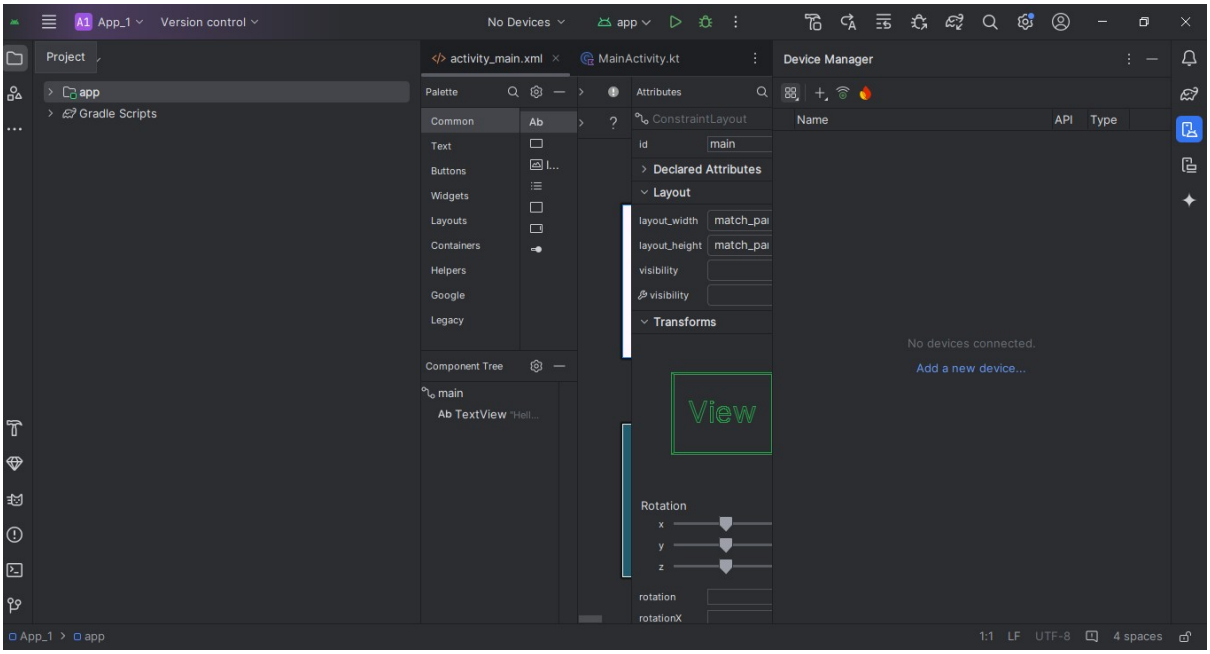
- The **XML Layouts** are used to define the structure of the UI, while the **Java or Kotlin** code in activities handles the interactions and updates to the UI elements.



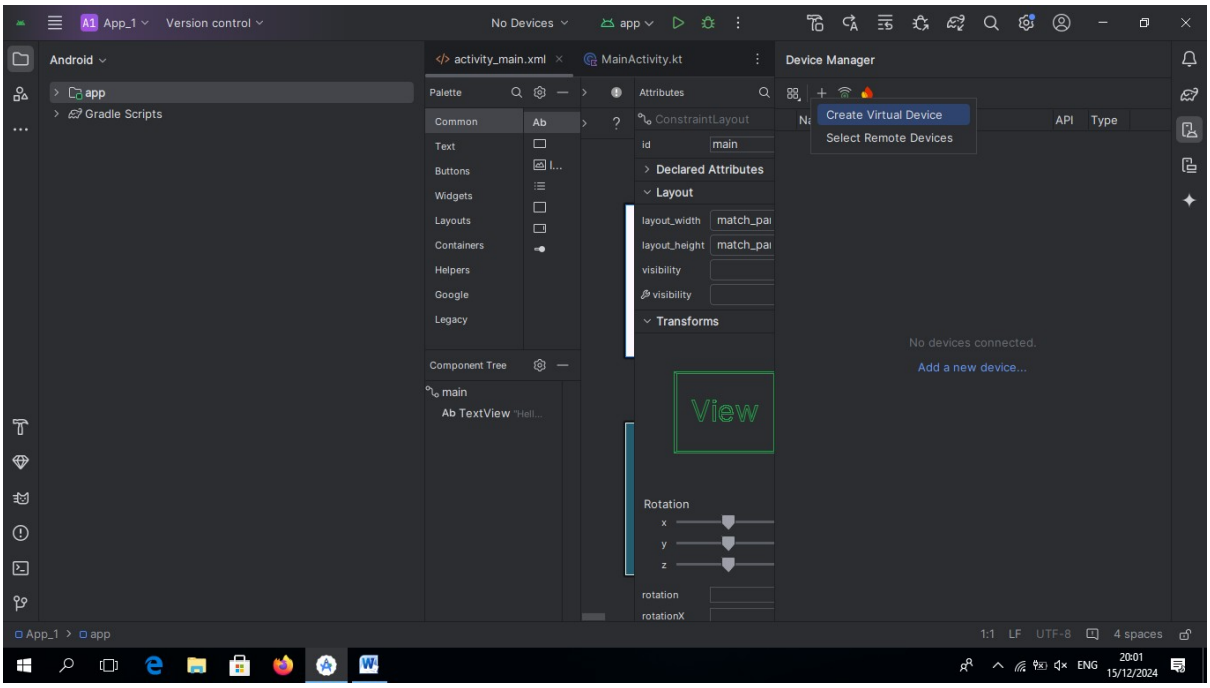
4. Creating Android Virtual Device (AVD)

Android Virtual Devices (AVDs) are emulated devices that allow you to test your application on different screen sizes, Android versions, and hardware configurations. To create an AVD:

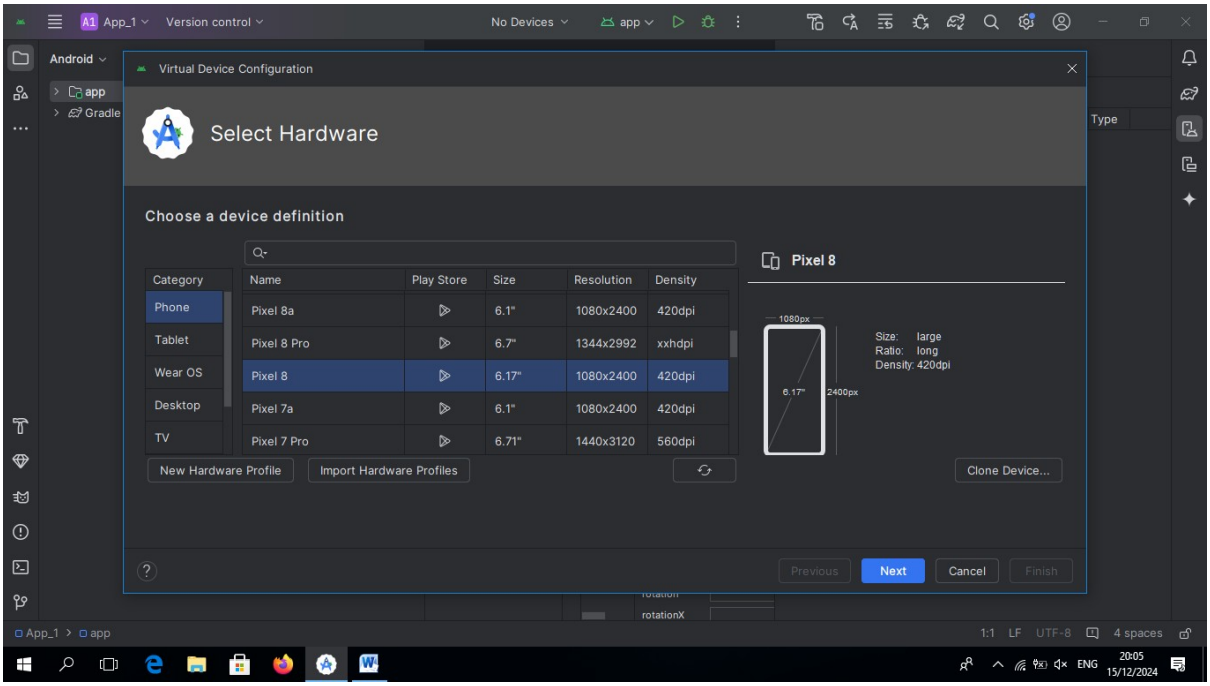
1. Open Android Studio.
2. Go to **Tools > AVD Manager**.



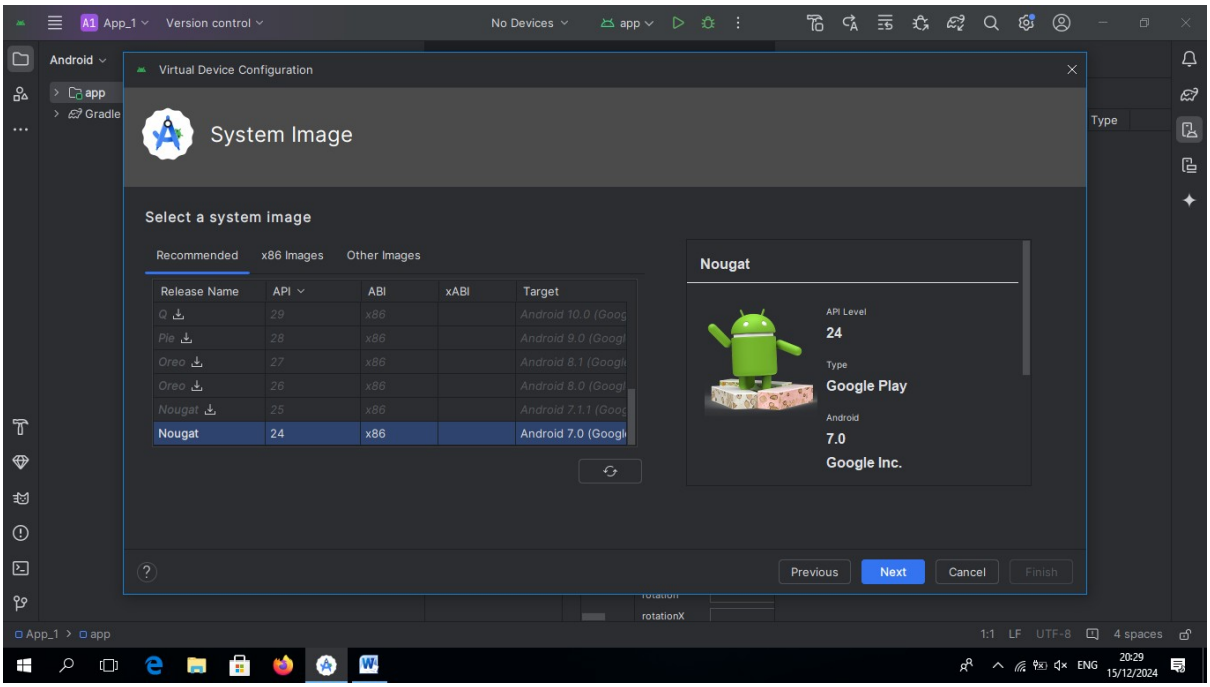
3. Click **Create Virtual Device.**



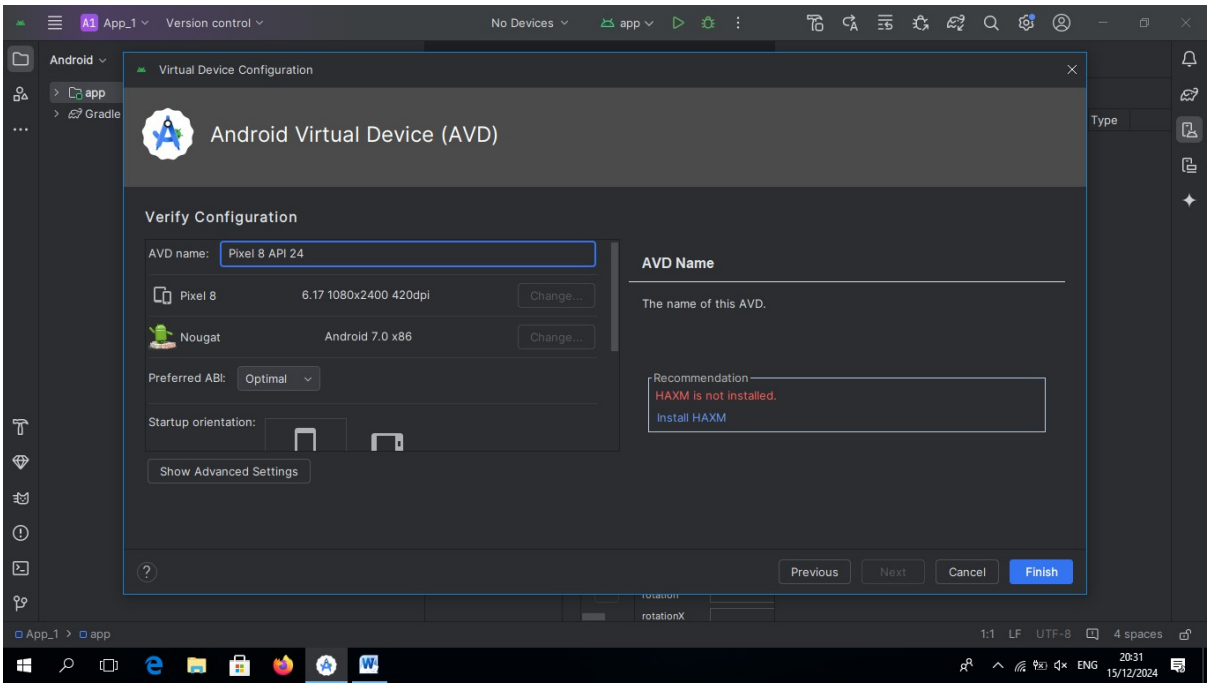
4. Choose a device (e.g., Pixel 8).

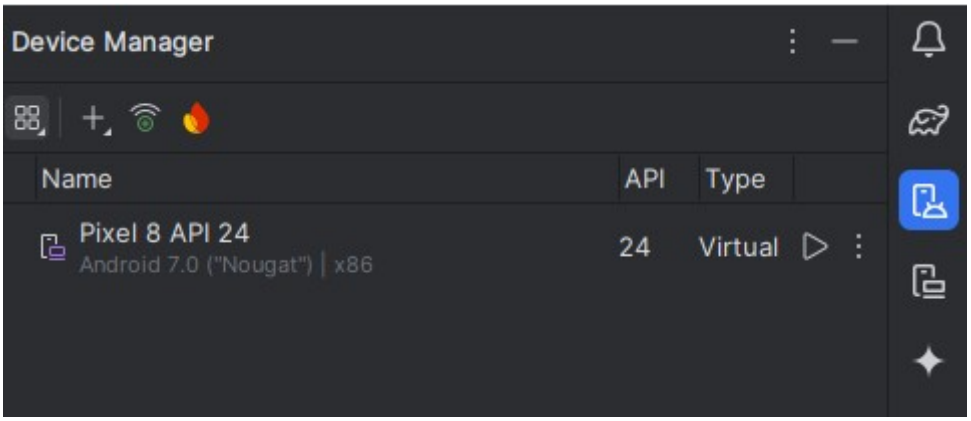


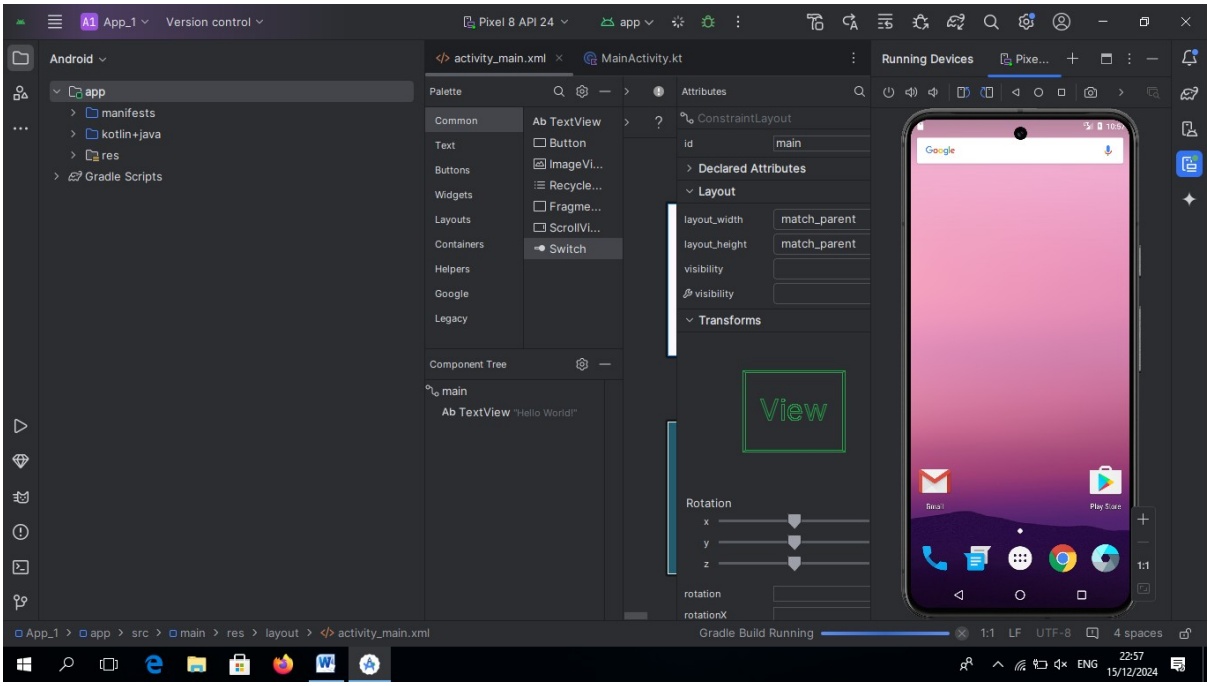
5. Select a system image (an Android version to emulate).



6. Customize the device configuration and start the emulator.







5. USB Debugging Mode

USB Debugging allows Android Studio to communicate with an Android device over USB for testing and debugging purposes. To enable USB debugging:

1. Open **Settings** on your Android device.
2. Go to **About phone** and tap **Build number** 7 times to enable Developer Options.
3. In Developer Options, toggle on **USB debugging**.

Once USB debugging is enabled, connect your device via USB, and Android Studio can deploy and debug apps directly on the device.

6. Android Application Overview

An Android app is essentially a collection of resources and components that interact to provide the user experience. Key concepts include:

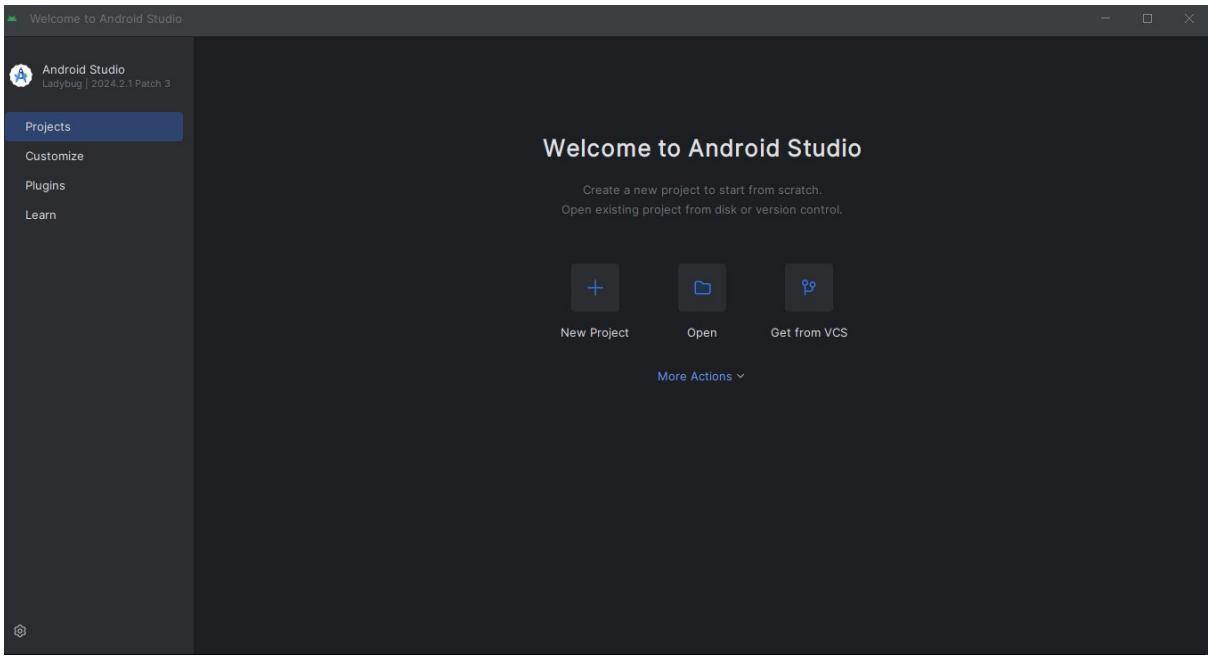
- **Activities**: Define user interfaces.

- **Services**: Handle background tasks.
- **Broadcast Receivers**: Respond to system-wide broadcasts.
- **Content Providers**: Share and access data.

7. Simple "Hello World" Program

(If you installed Android Studio first time, you will get below screen.)

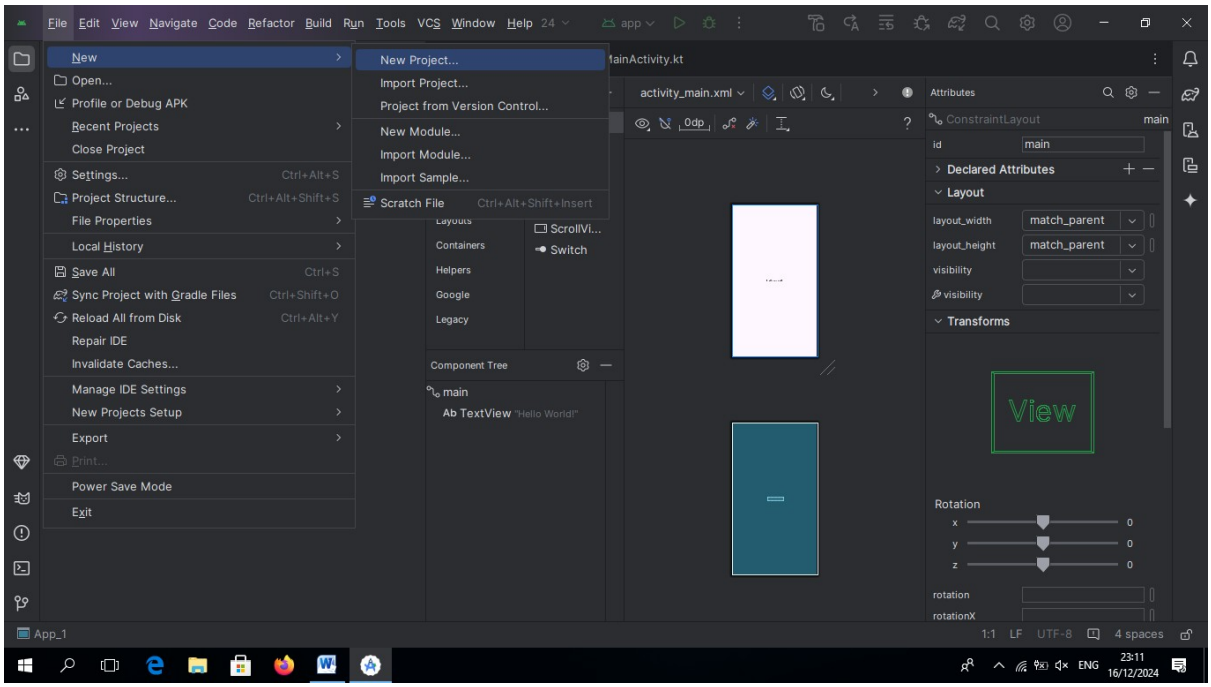
Step 1: Click on New Project



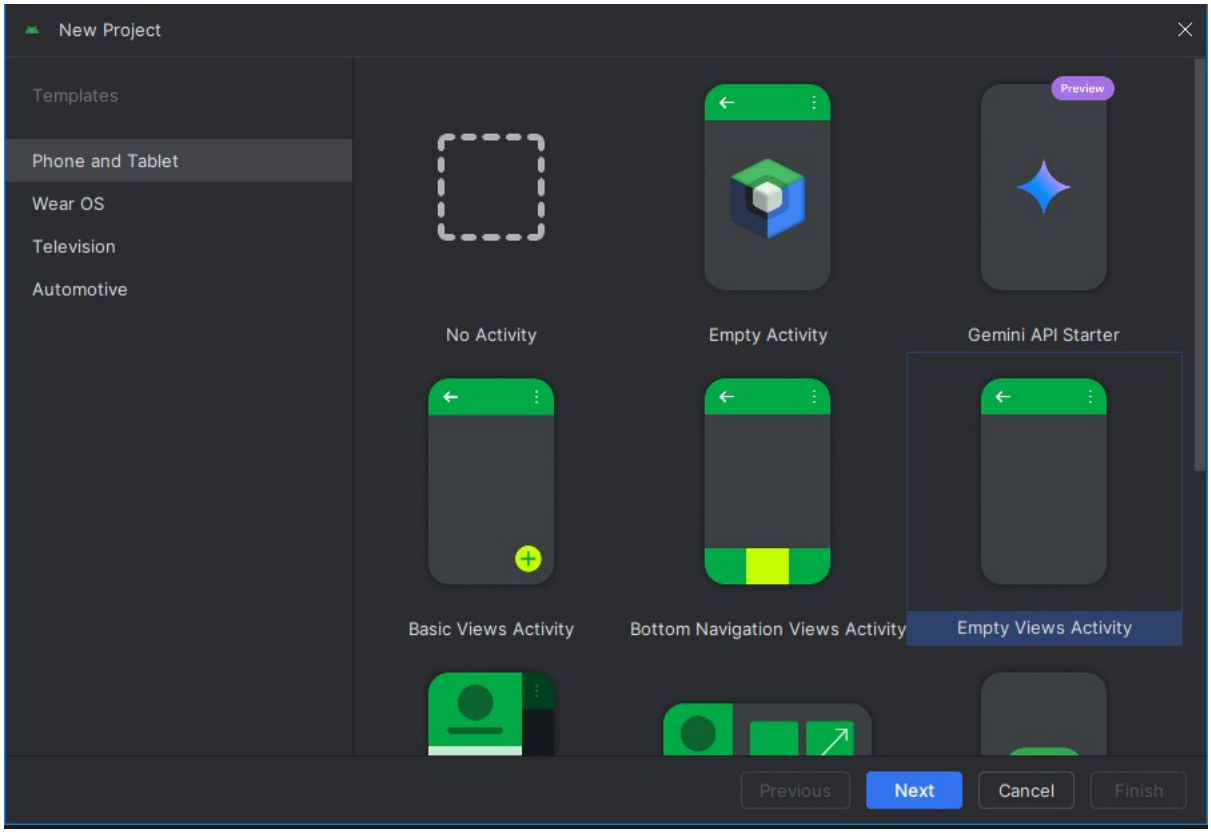
OR

(If you already installed Android studio, your software screen lookalikes below screen.)

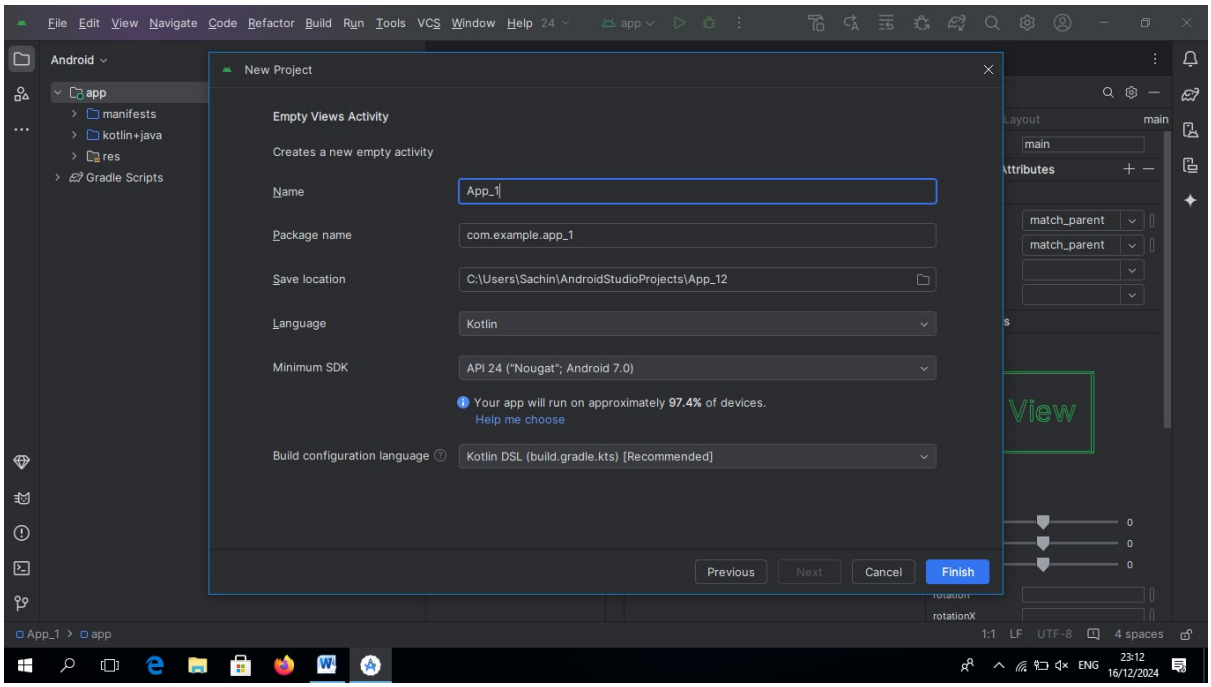
Go to File Menu -> Select New Option -> Select New Project

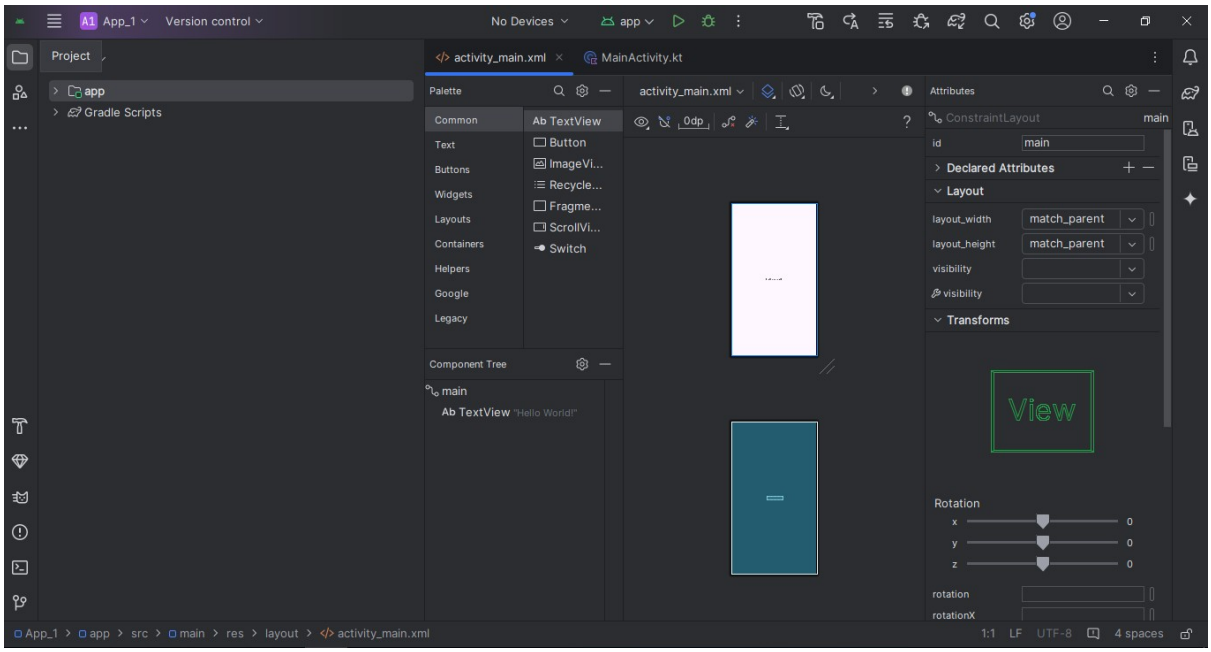


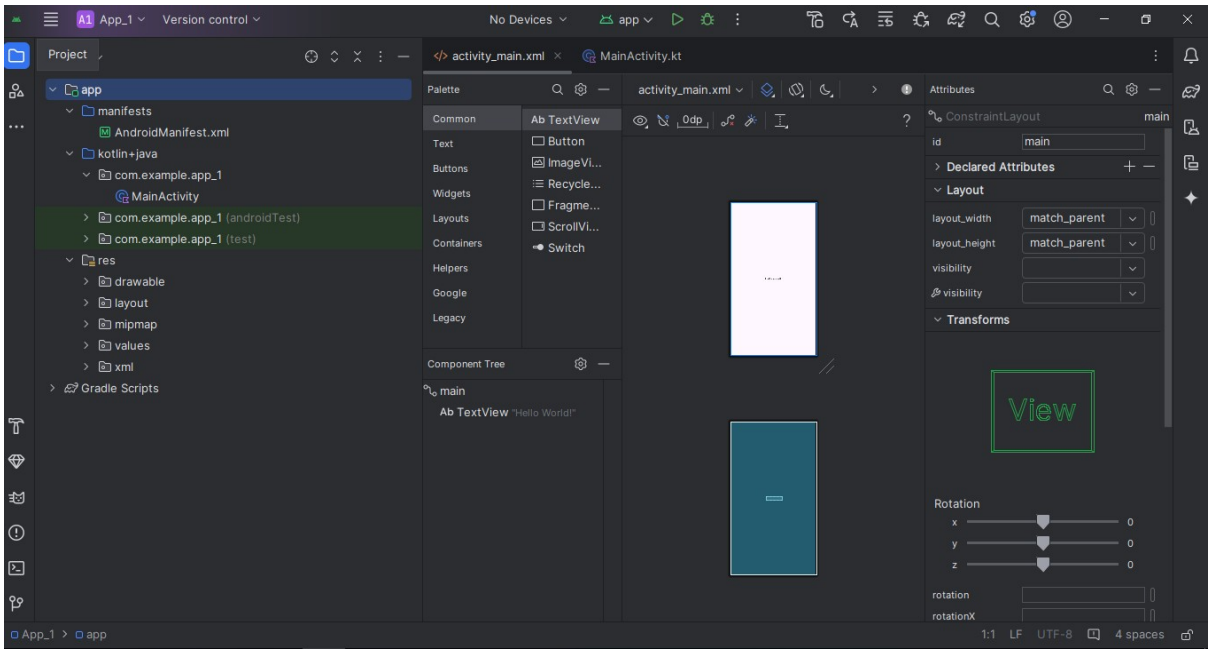
Step 2: Select Phone and Tablet -> then Select Empty Views Activity (or Empty Activity).



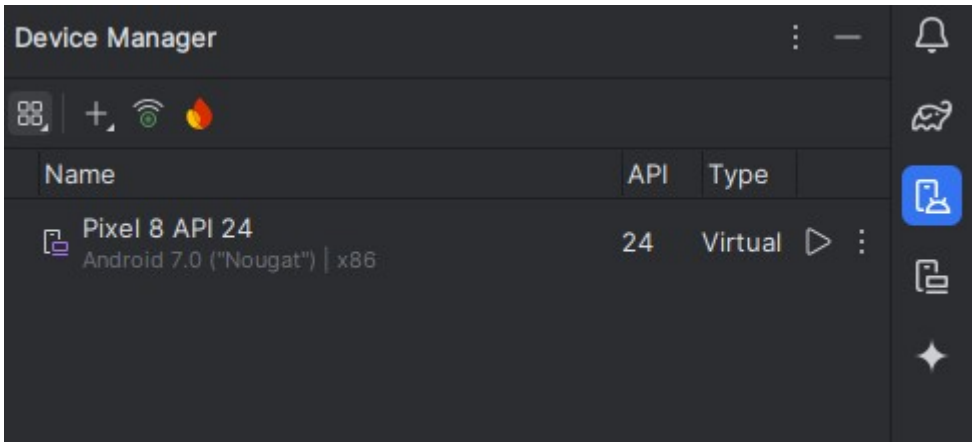
Step 3: Give Name to the project and also select language (Java / Kotlin)

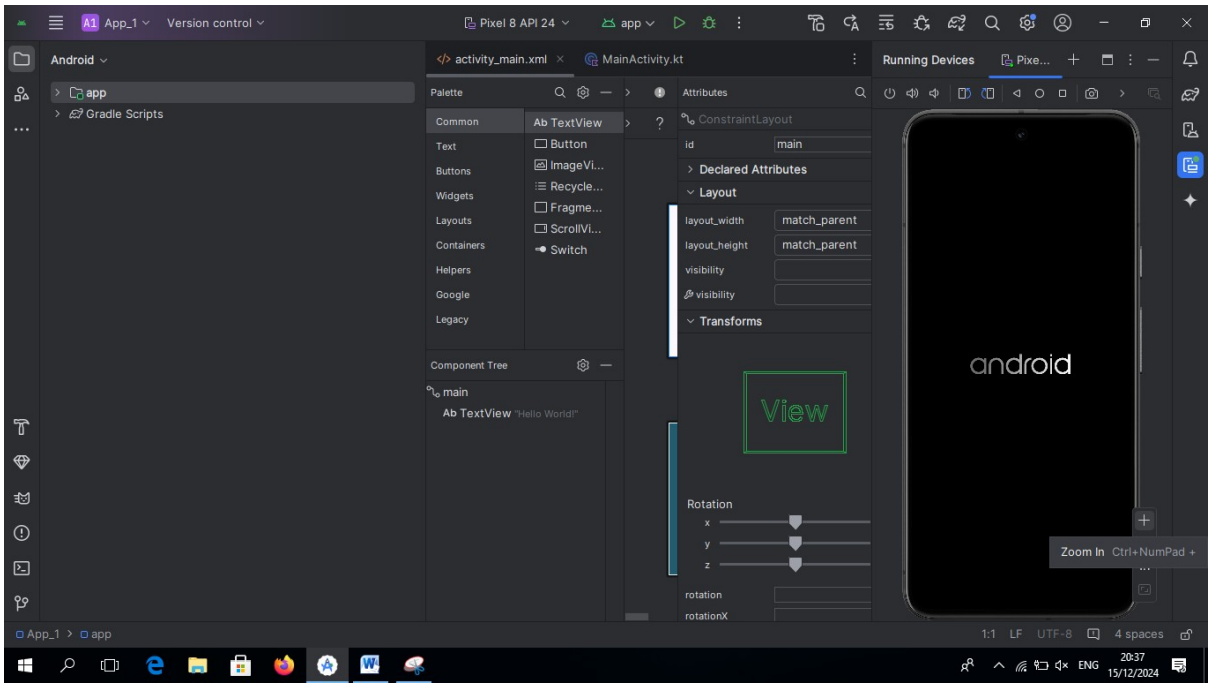

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
New Project
Empty Views Activity
Creates a new empty activity
Name
App_1
Package name
com.example.app_1
Save location
C:\Users\Sachin\AndroidStudioProjects\App_12
Language
Kotlin
Minimum SDK
API 24 ("Nougat"; Android 7.0)
Your app will run on approximately 97.4% of devices.
Help me choose
Build configuration language
Kotlin DSL (build.gradle.kts) [Recommended]
Previous
Next
Cancel
Finish


App_1
Version control
No Devices
Project
activity_main.xml
MainActivity.kt
Palette
Component Tree
Attributes
ConstraintLayout
Declared Attributes
Layout
Transforms
Rotation

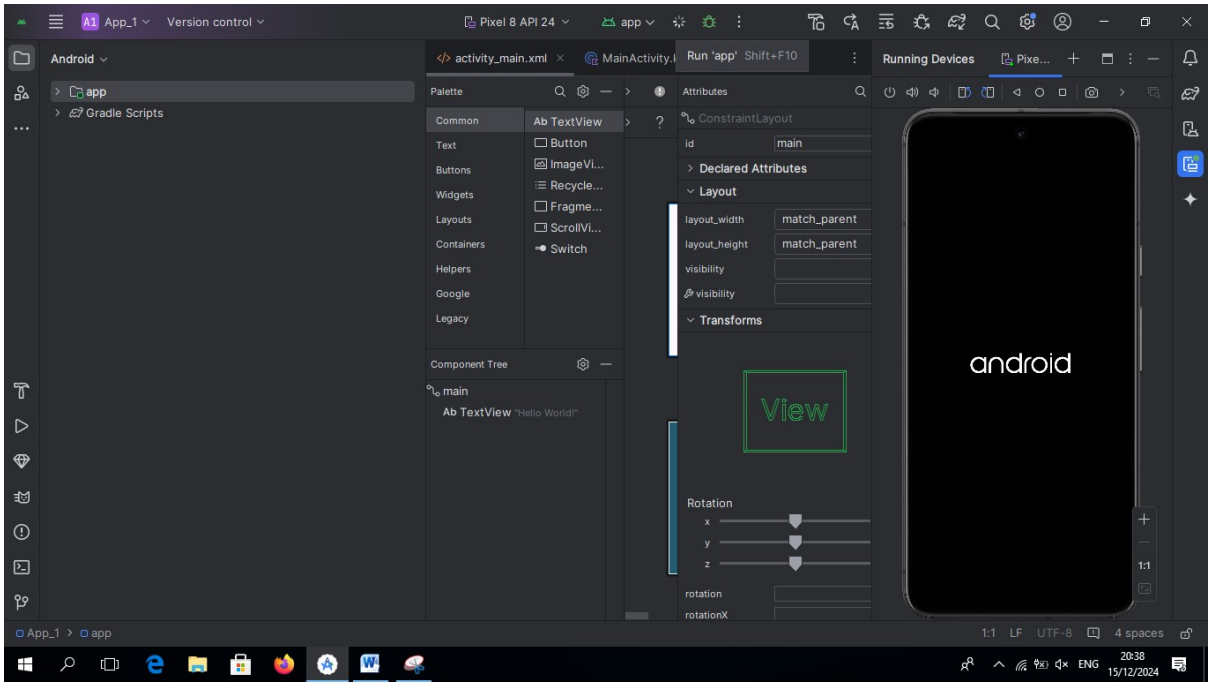


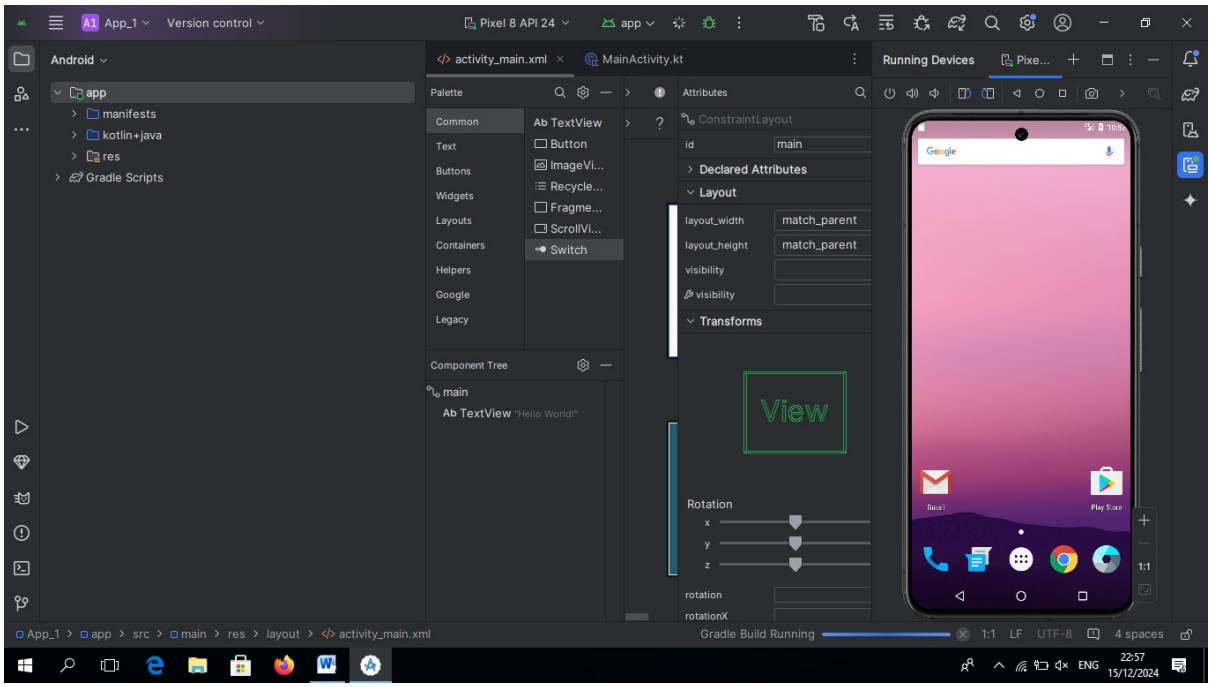
Step 4: Go to Device Manager and to start AVD click on “PLAY” button.





Step 5: Run your app.





Step 6: After Successful Gradle Build you will get the output.

A "Hello World" program in Android is typically the first app you create to get familiar with Android development.

Running the Application

To run the app:

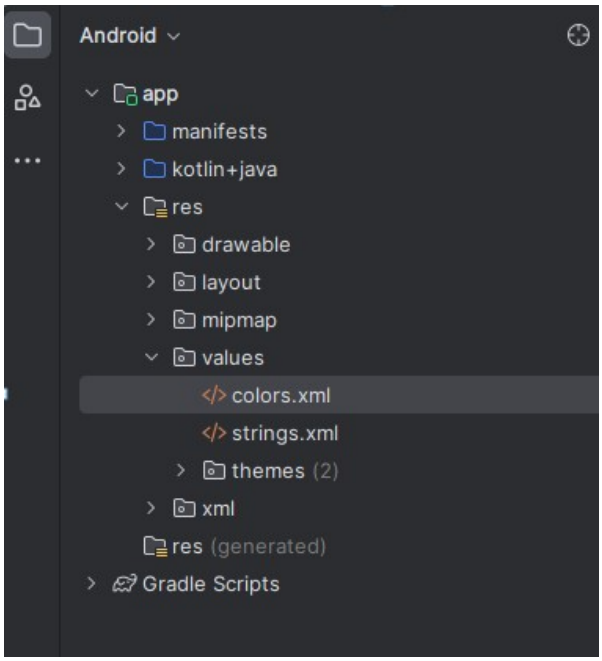
1. Connect an Android device or start an AVD (emulator).
2. In Android Studio, click the **Run** button (green triangle).
3. Select the device or emulator to deploy the app to.
4. The app should now run, displaying a screen with the text "Hello, World!"

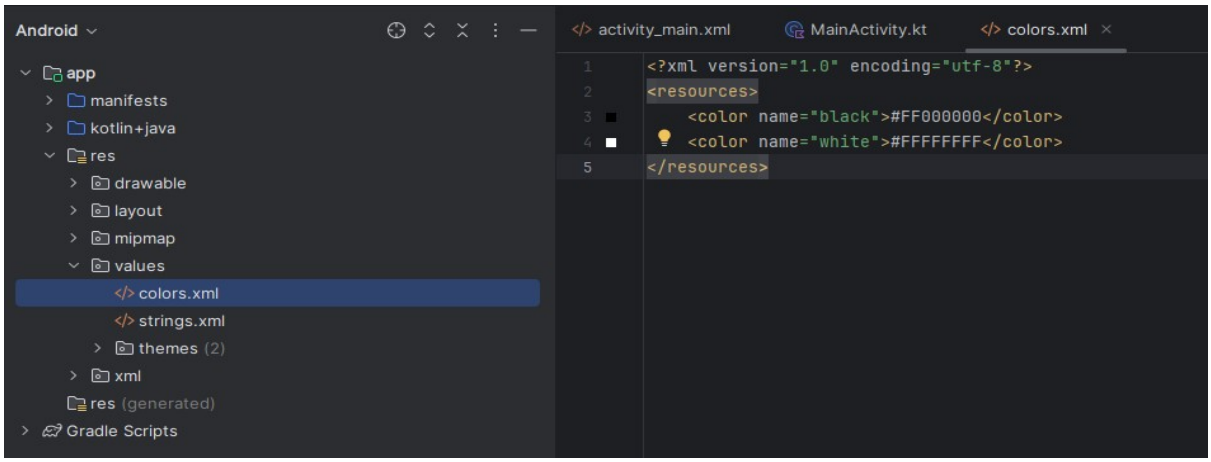
PRACTICAL-2

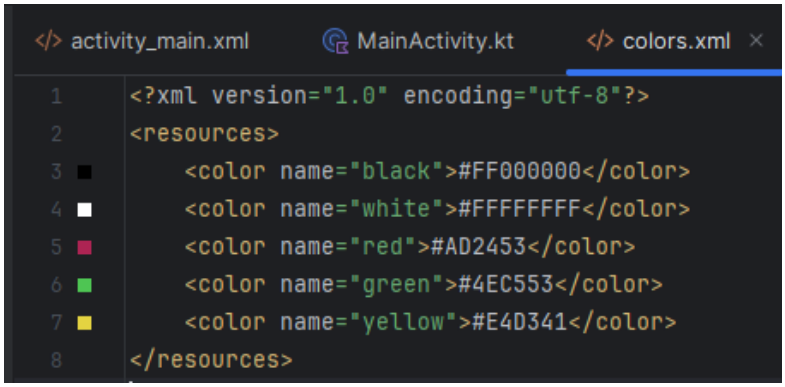
Programming Resources

Android Resources: (Color, Them, String, Drawable, Dimension, Image)

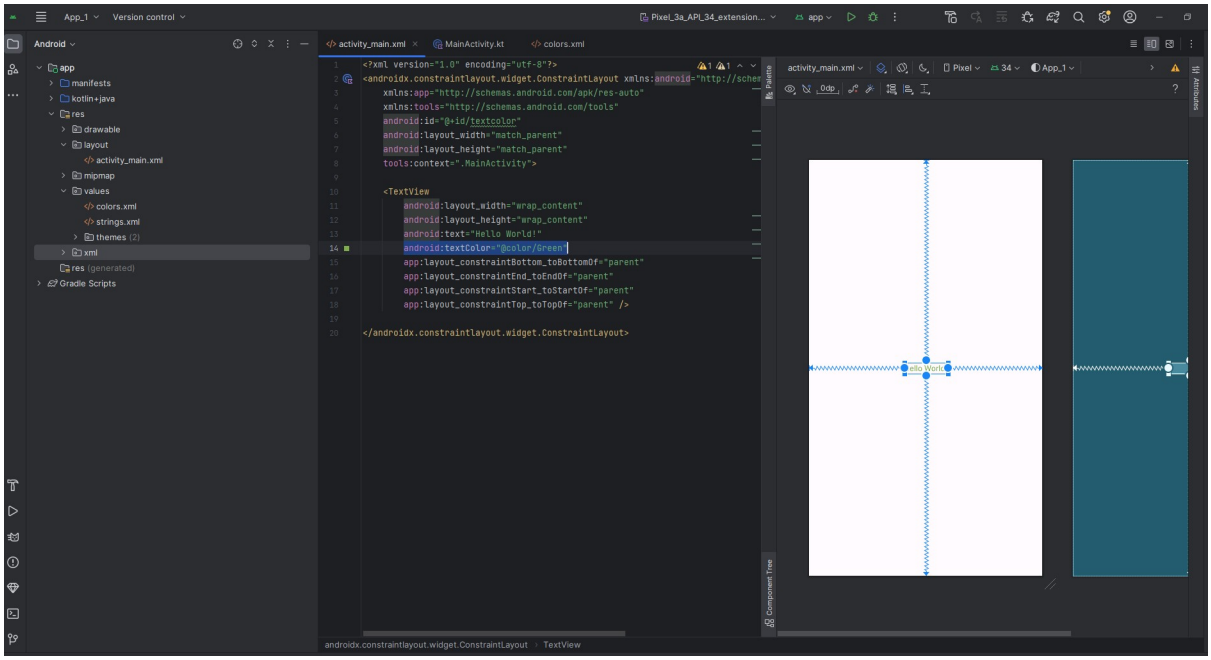
1. Color:

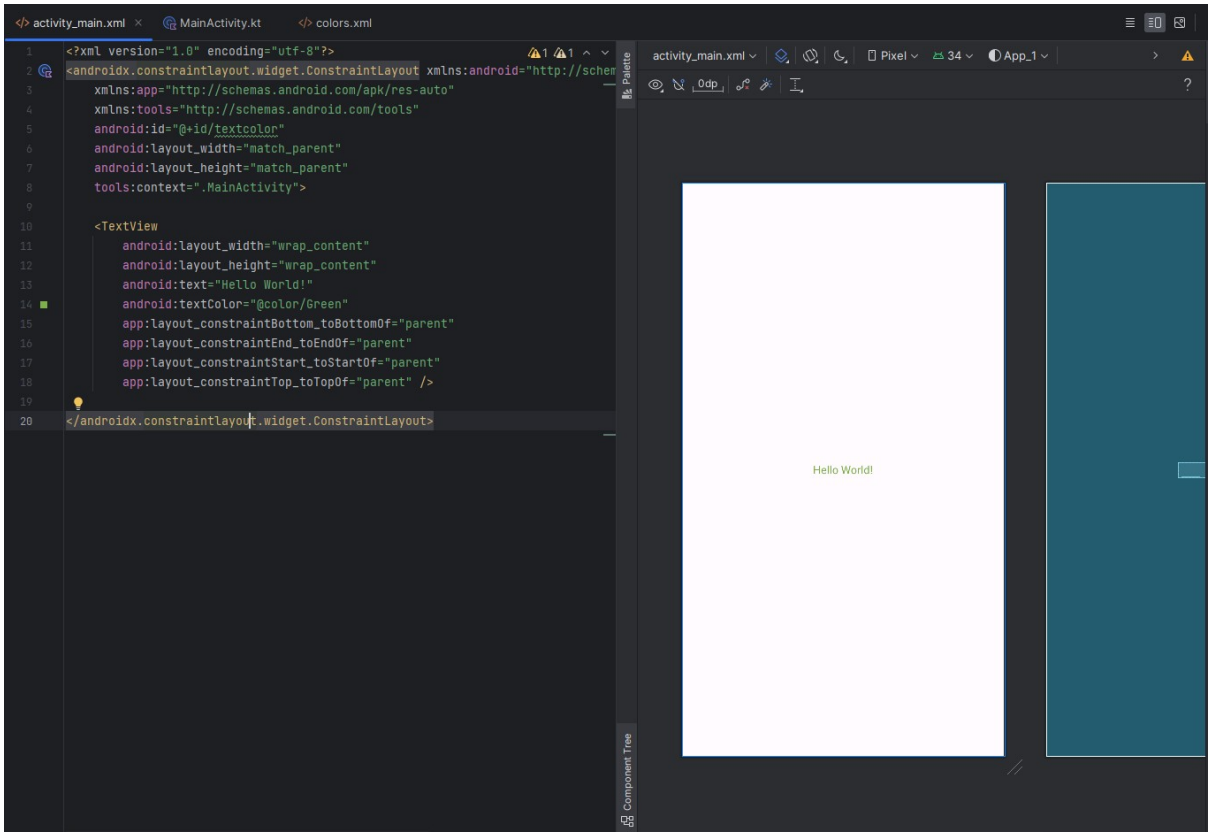




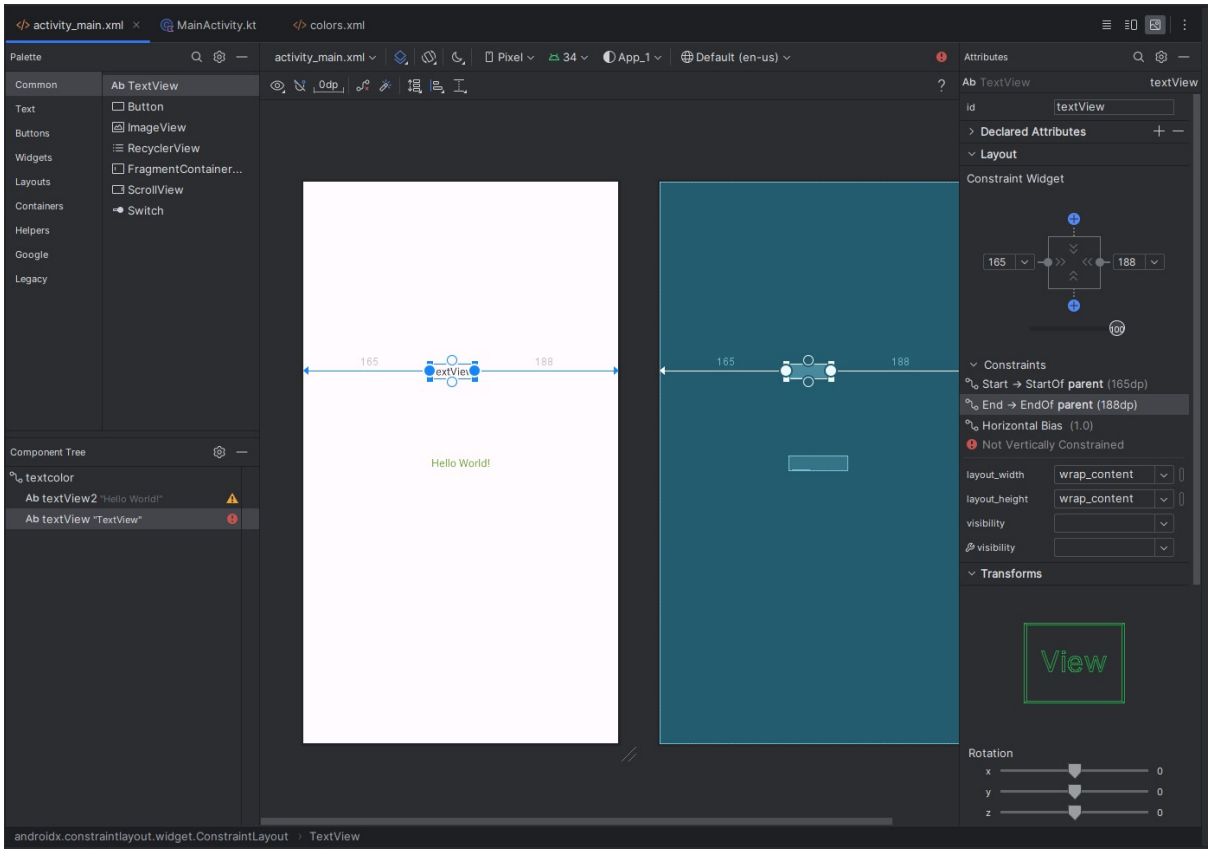


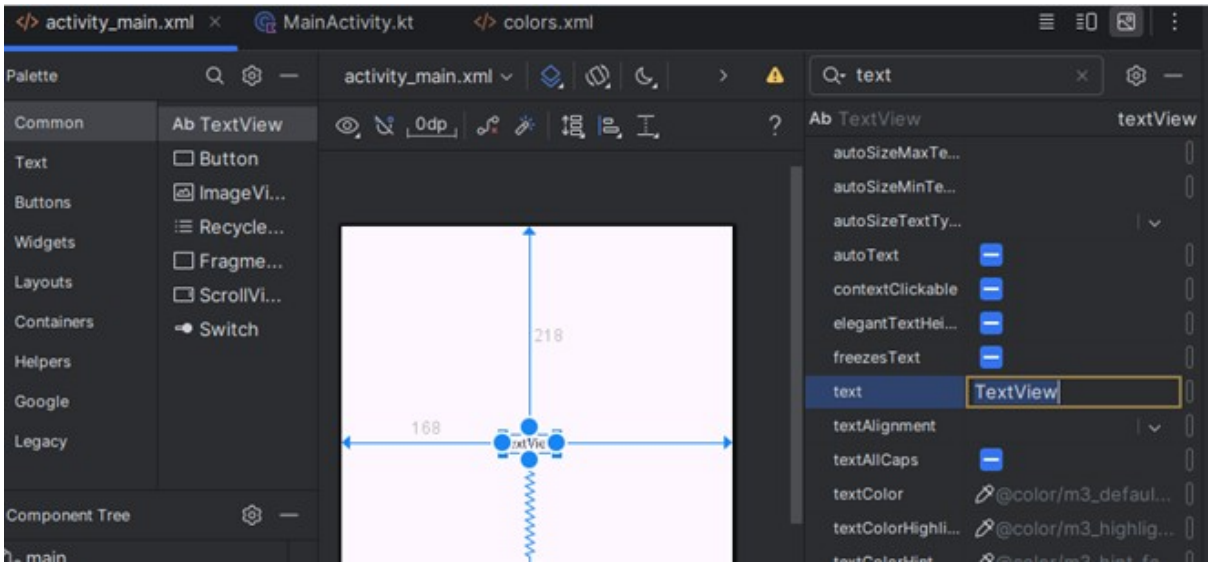
Layout Folder-> activity_main.xml

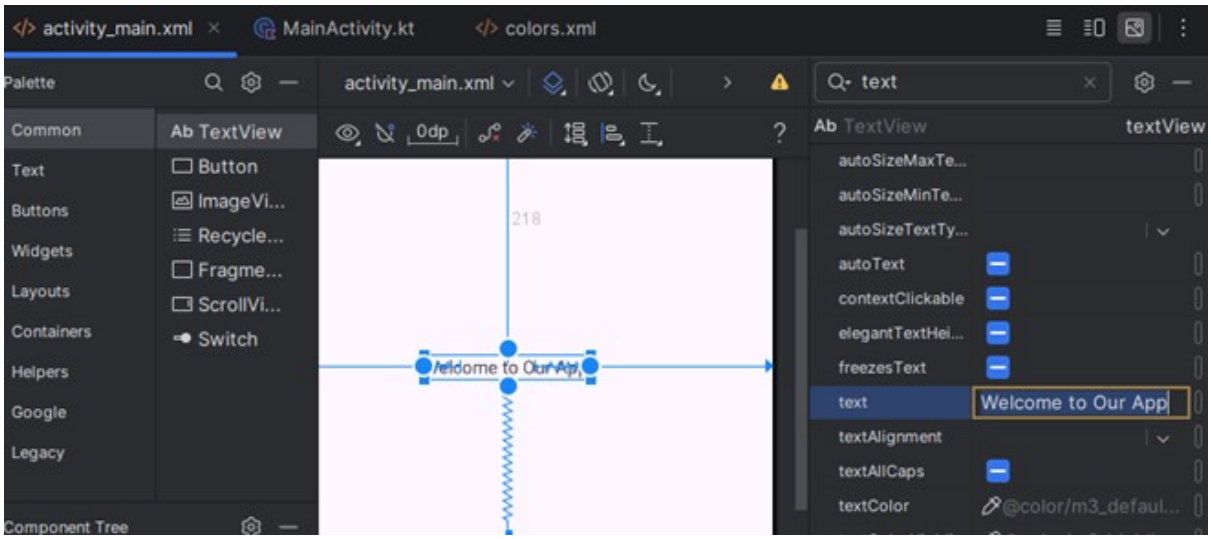


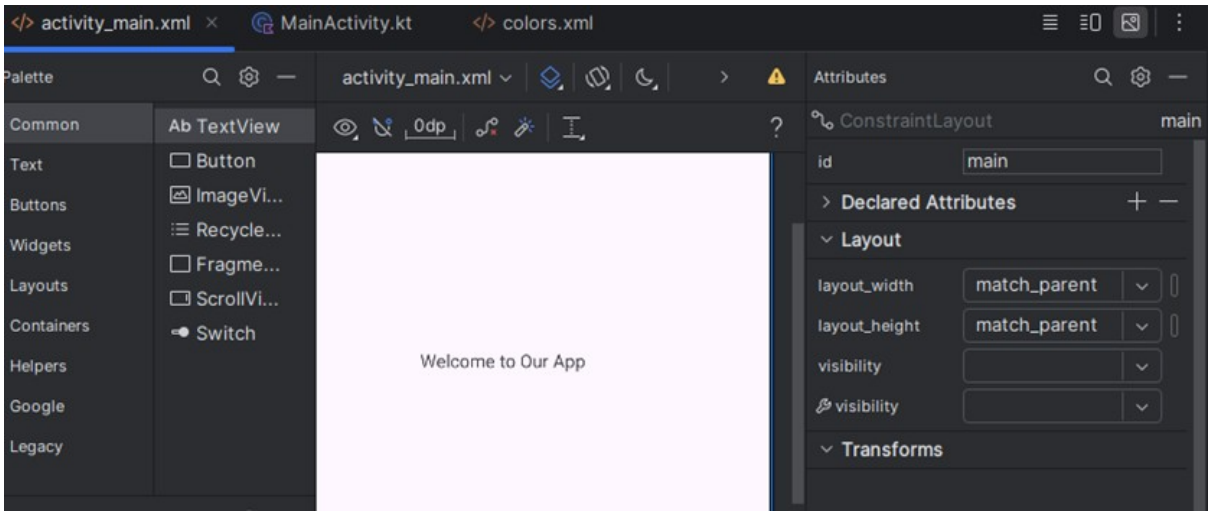


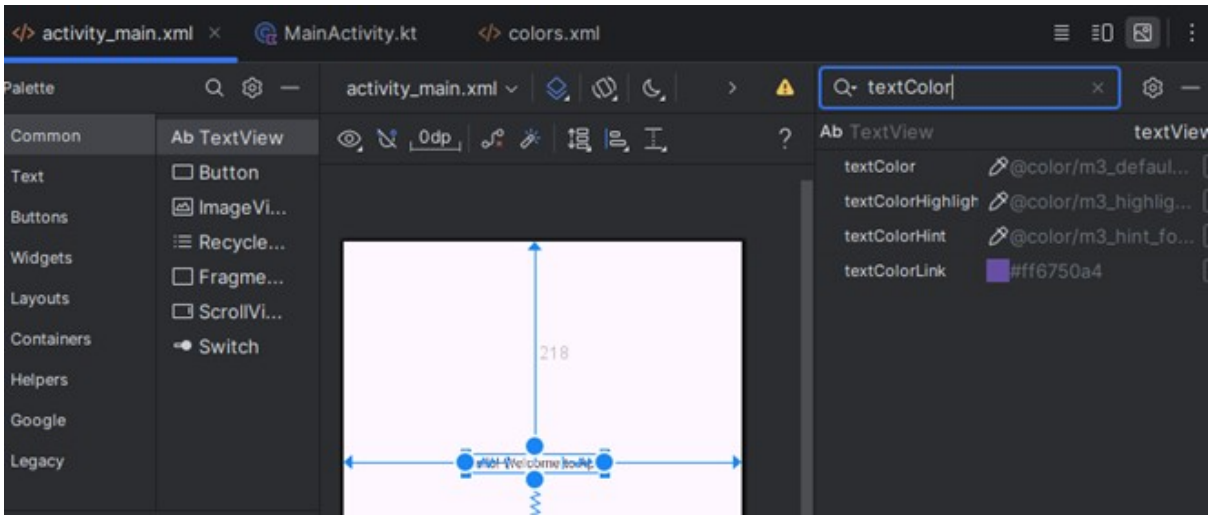
Design Section -> Drag and Drop one more TextView

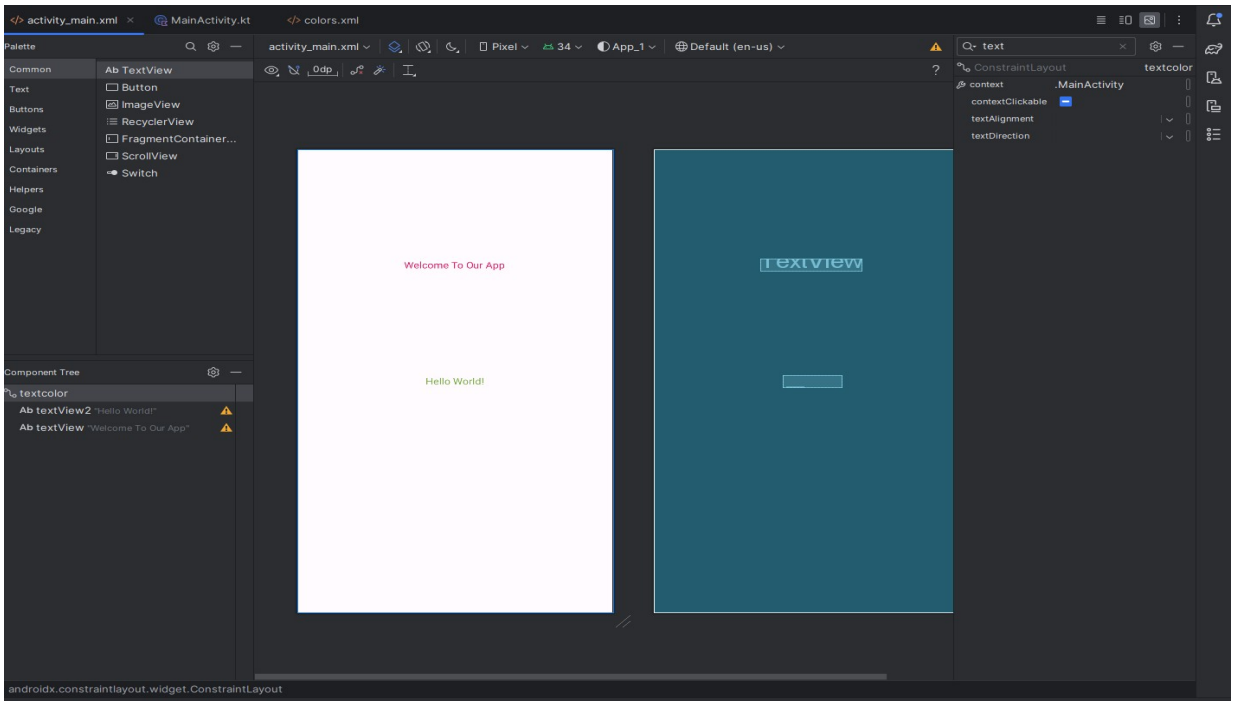





activity_main.xml
MainActivity.kt
colors.xml
Palette
Common
Text
Buttons
Widgets
Layouts
Containers
Helpers
Google
Legacy
Ab TextView
Button
ImageVi...
Recycle...
Fragme...
ScrollVi...
Switch
Component Tree
activity_main.xml
0dp
218
text
Ab TextView
textView
autoSizeMaxTe...
autoSizeMinTe...
autoSizeTextTy...
autoText
contextClickable
elegantTextHei...
freezesText
text
Welcome to Our App
textAlignment
textAllCaps
textColor
@color/m3_defaul...

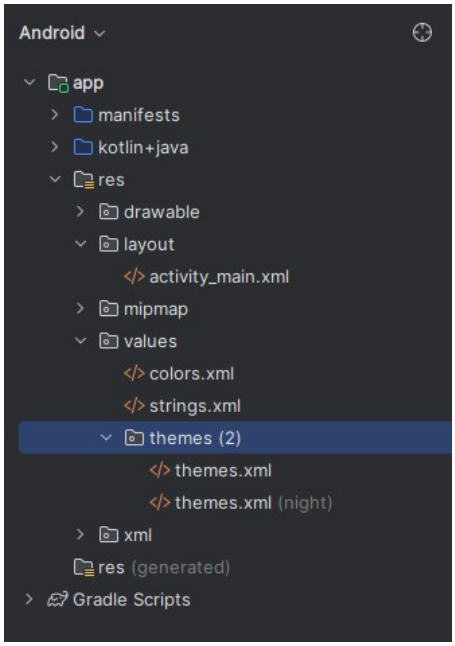

activity_main.xml
MainActivity.kt
colors.xml
Palette
Common
Text
Buttons
Widgets
Layouts
Containers
Helpers
Google
Legacy
Ab TextView
Button
ImageVi...
Recycle...
Fragme...
ScrollVi...
Switch
activity_main.xml
0dp
Welcome to Our App
Attributes
ConstraintLayout
main
id
main
Declared Attributes
Layout
layout_width
match_parent
layout_height
match_parent
visibility
visibility
Transforms

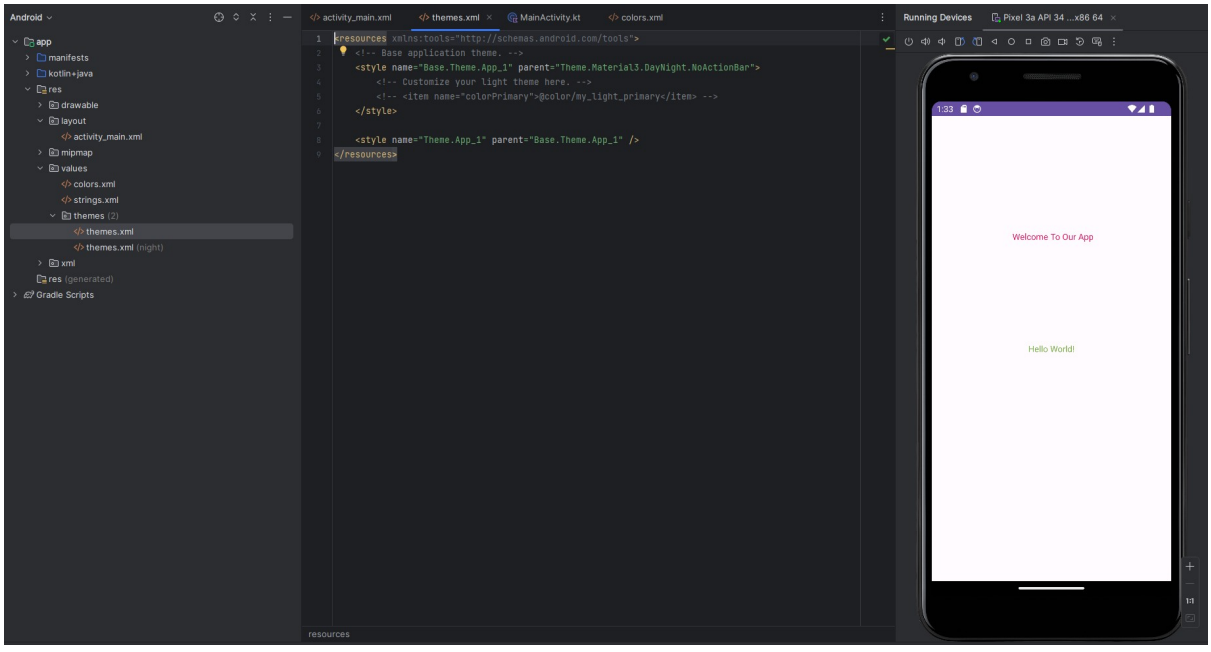

activity_main.xml
MainActivity.kt
colors.xml
Palette
Common
Text
Buttons
Widgets
Layouts
Containers
Helpers
Google
Legacy
Ab TextView
Button
ImageVi...
Recycle...
Fragme...
ScrollVi...
Switch
activity_main.xml
0dp
218
textColor
Ab TextView
textView
textColor
@color/m3_defaul...
textColorHighligh
@color/m3_highlig...
textColorHint
@color/m3_hint_fo...
textColorLink
#ff6750a4


activity_main.xml
MainActivity.kt
colors.xml
Palette
Common
Text
Buttons
Widgets
Layouts
Containers
Helpers
Google
Legacy
TextView
Button
ImageView
RecyclerView
FragmentContainer...
ScrollView
Switch
Component Tree
textcolor
textView2 "Hello World!"
textView "Welcome To Our App"
Welcome To Our App
Hello World!
TextView
ConstraintLayout
textcolor
context
.MainActivity
contextClickable
textAlignment
textDirection
androidx.constraintlayout.widget.ConstraintLayout


Running Devices
Pixel 3a API 34 ...x86 64
1:31
Welcome To Our App
Hello World!

2. Them:


Android
app
manifests
kotlin+java
res
drawable
layout
activity_main.xml
mipmap
values
colors.xml
strings.xml
themes (2)
themes.xml
themes.xml (night)
xml
res (generated)
Gradle Scripts

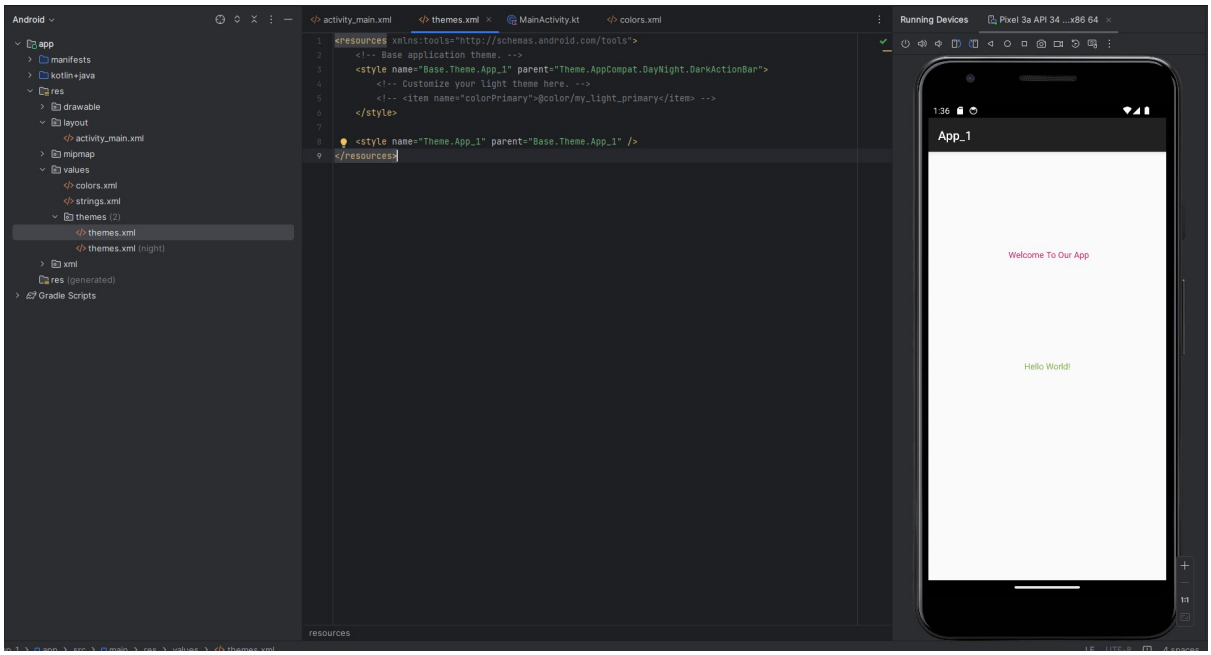

activity_main.xml
themes.xml
MainActivity.kt
colors.xml
<resources xmlns:tools="http://schemas.android.com/tools">
<!-- Base application theme. -->
<style name="Base.Theme.App_1" parent="Theme.Material3.DayNight.NoActionBar">
<!-- Customize your light theme here. -->
<!-- <item name="colorPrimary">@color/my_light_primary</item> -->
</style>
<style name="Theme.App_1" parent="Base.Theme.App_1" />
</resources>
resources
Running Devices
Pixel 3a API 34 ...x86 64
Welcome To Our App
Hello World!

activity_main.xml | themes.xml | MainActivity.kt | colors.xml

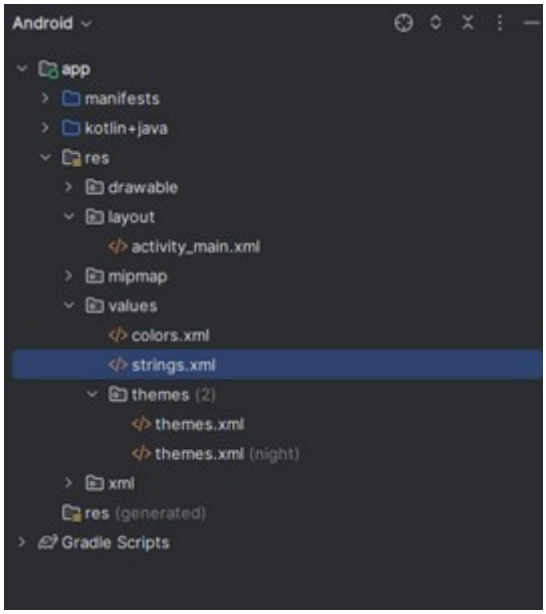
```xml
<resources xmlns:tools="http://schemas.android.com/tools">
    <!-- Base application theme. -->
    <style name="Base.Theme.App_1" parent="Theme.Material3.DayNight.NoActionBar">
        <!-- Customize your light theme here. -->
        <!-- <item name="colorPrimary">@color/my_light_primary</item> -->
    </style>

    <style name="Theme.App_1" parent="Base.Theme.App_1" />
</resources>
```

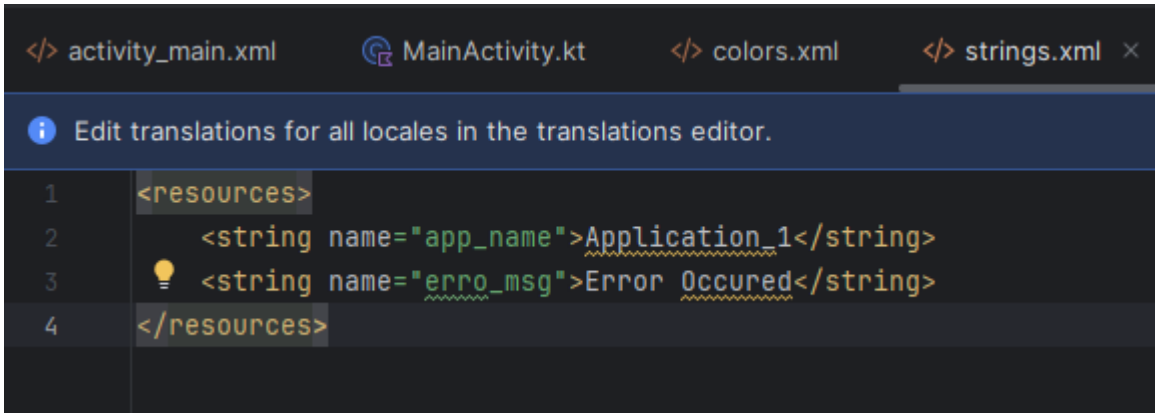
activity_main.xml | themes.xml | MainActivity.kt | colors.xml

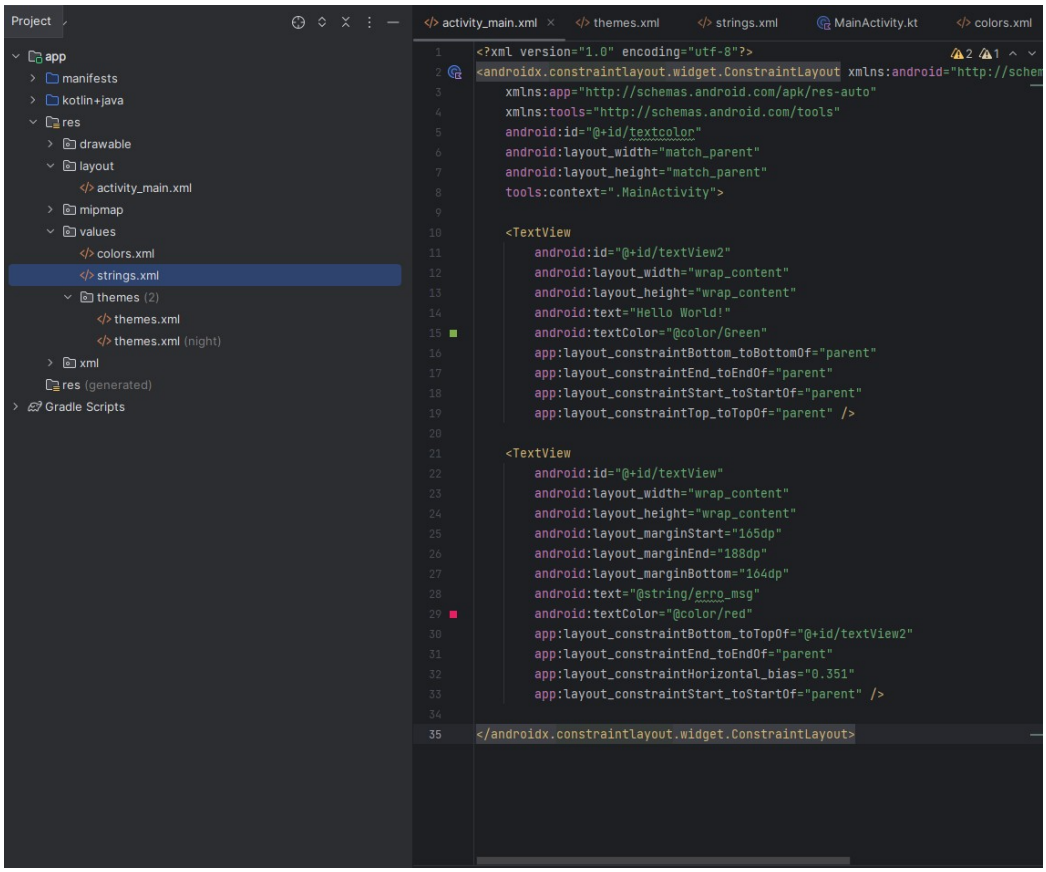
```xml
<resources xmlns:tools="http://schemas.android.com/tools">
    <!-- Base application theme. -->
    <style name="Base.Theme.App_1" parent="Theme.AppCompat.DayNight.DarkActionBar">
        <!-- Customize your light theme here. -->
        <!-- <item name="colorPrimary">@color/my_light_primary</item> -->
    </style>

    <style name="Theme.App_1" parent="Base.Theme.App_1" />
</resources>
```



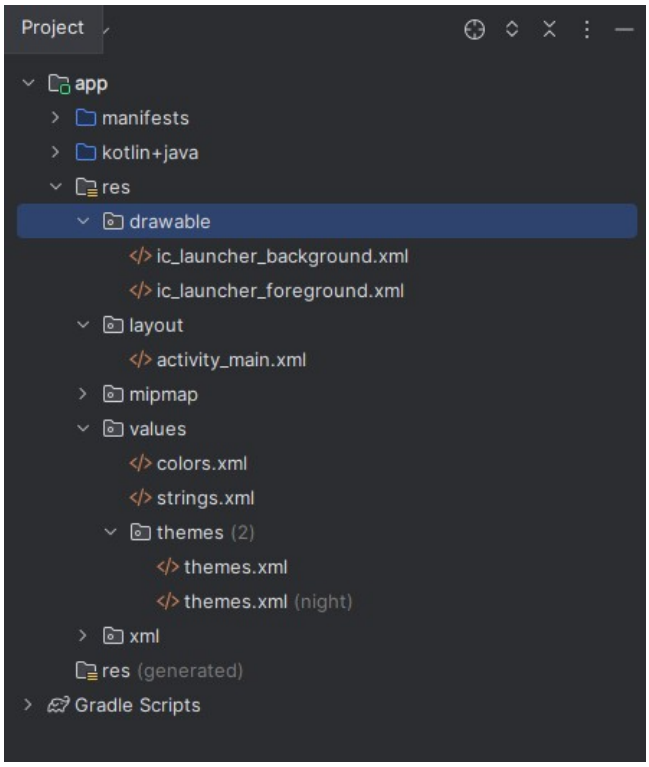
3. String:

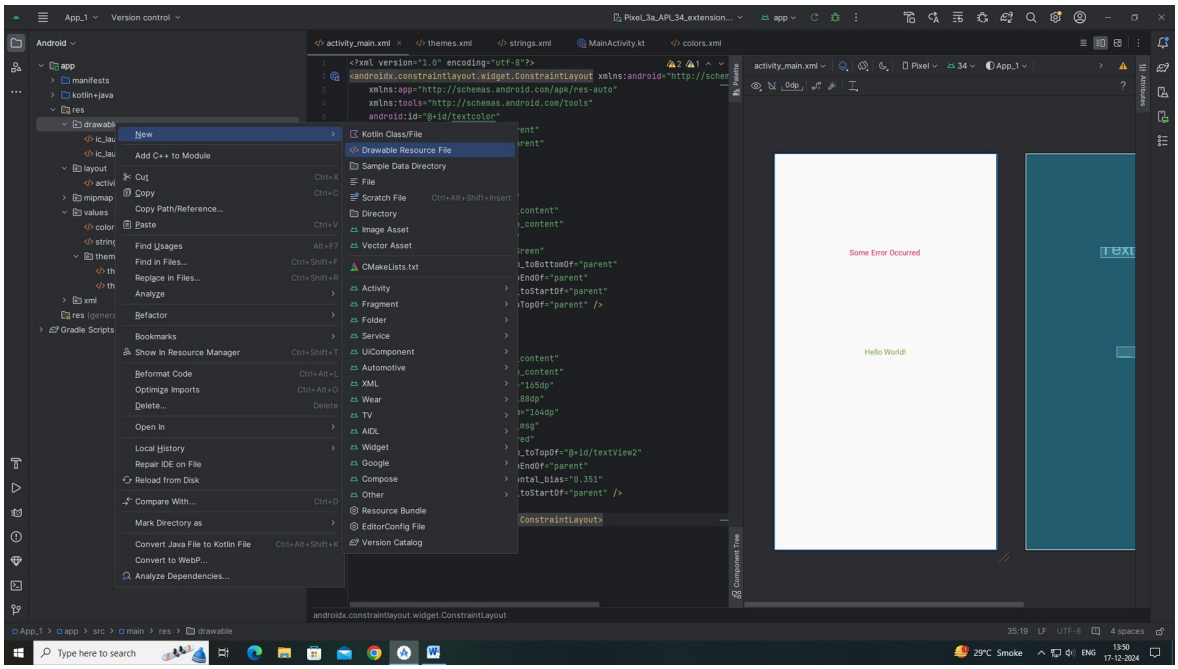

Android
app
manifests
kotlin+java
res
drawable
layout
activity_main.xml
mipmap
values
colors.xml
strings.xml
themes (2)
themes.xml
themes.xml (night)
xml
res (generated)
Gradle Scripts

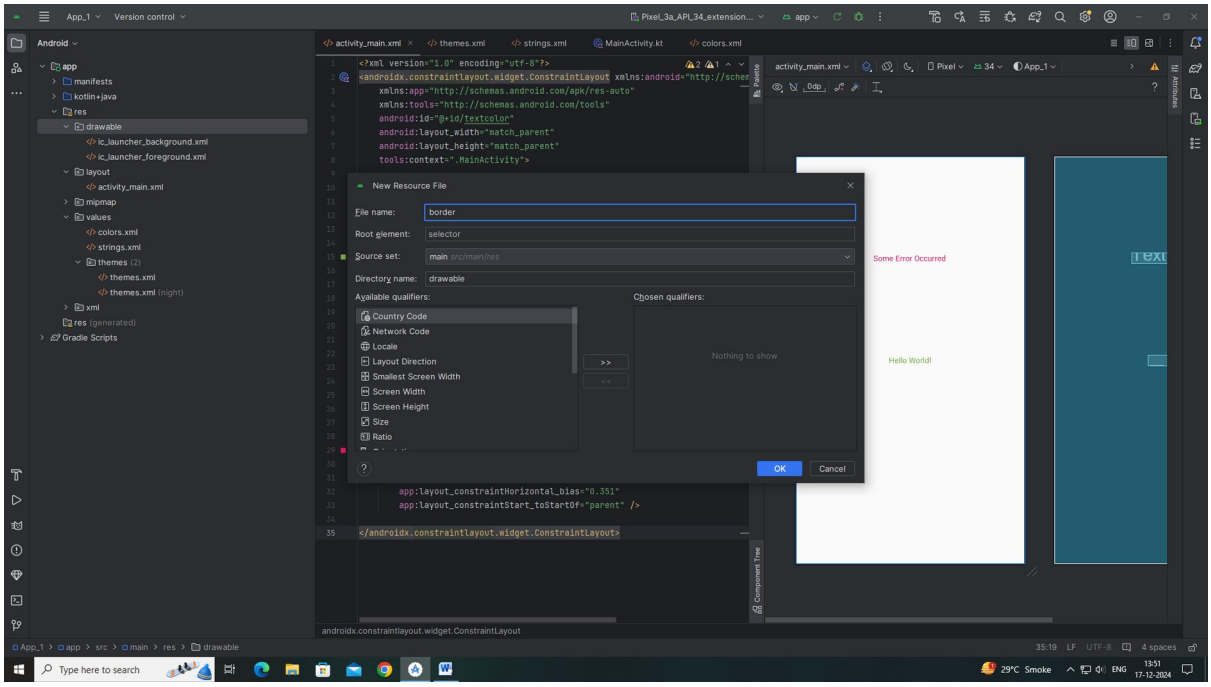

activity_main.xml
MainActivity.kt
colors.xml
strings.xml
Edit translations for all locales in the translations editor.
<resources>
<string name="app_name">Application_1</string>
<string name="erro_msg">Error Occured</string>
</resources>

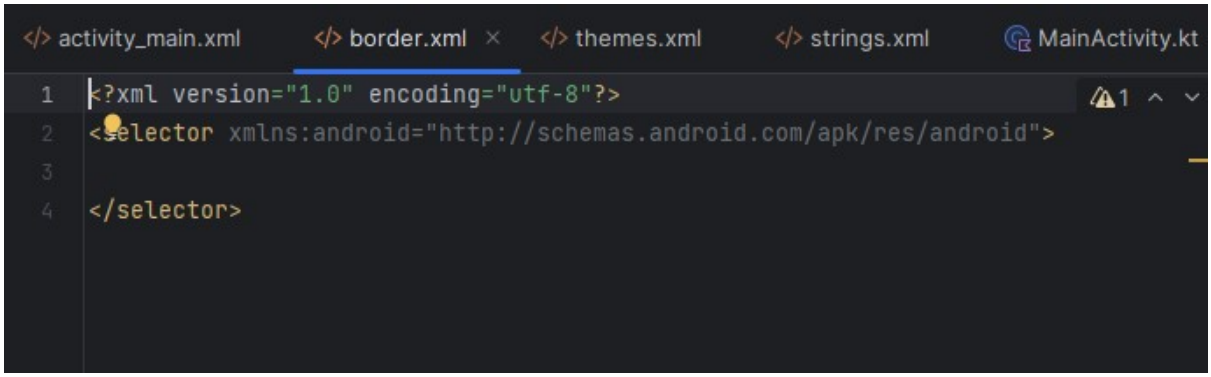

Project
app
manifests
kotlin+java
res
drawable
layout
activity_main.xml
mipmap
values
colors.xml
strings.xml
themes (2)
themes.xml
themes.xml (night)
xml
res (generated)
Gradle Scripts
activity_main.xml
themes.xml
strings.xml
MainActivity.kt
colors.xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schem
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:id="@+id/textcolor"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity">
<TextView
android:id="@+id/textView2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Hello World!"
android:textColor="@color/Green"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />
<TextView
android:id="@+id/textView"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginStart="165dp"
android:layout_marginEnd="188dp"
android:layout_marginBottom="164dp"
android:text="@string/erro_msg"
android:textColor="@color/red"
app:layout_constraintBottom_toTopOf="@+id/textView2"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.351"
app:layout_constraintStart_toStartOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>

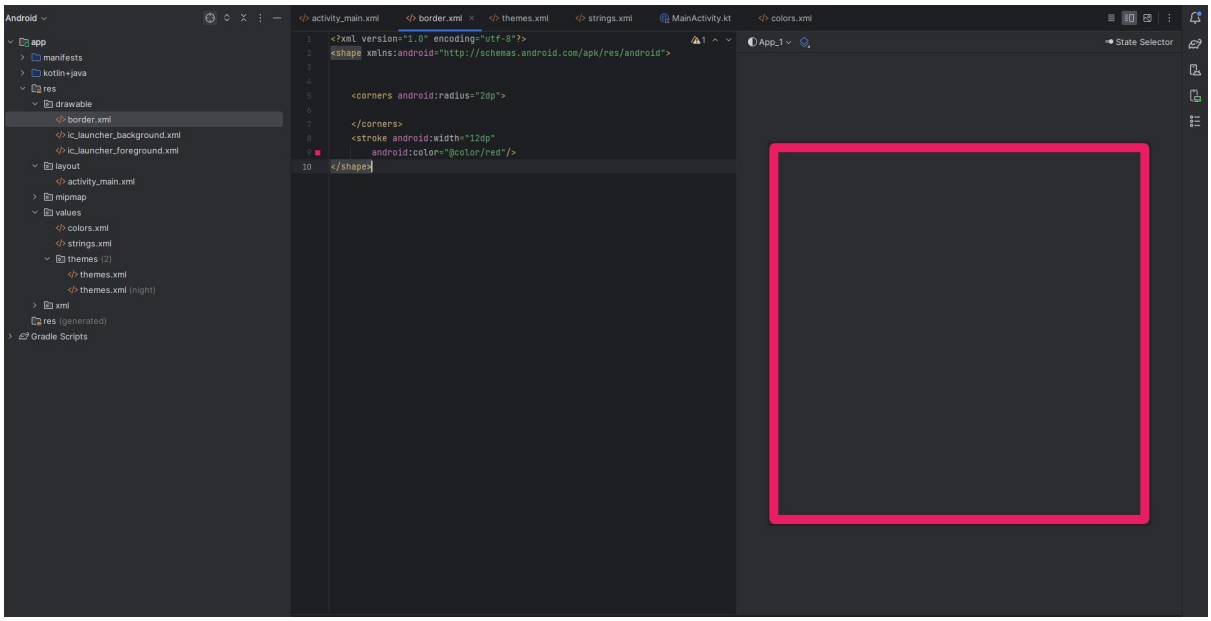
4. Drawable

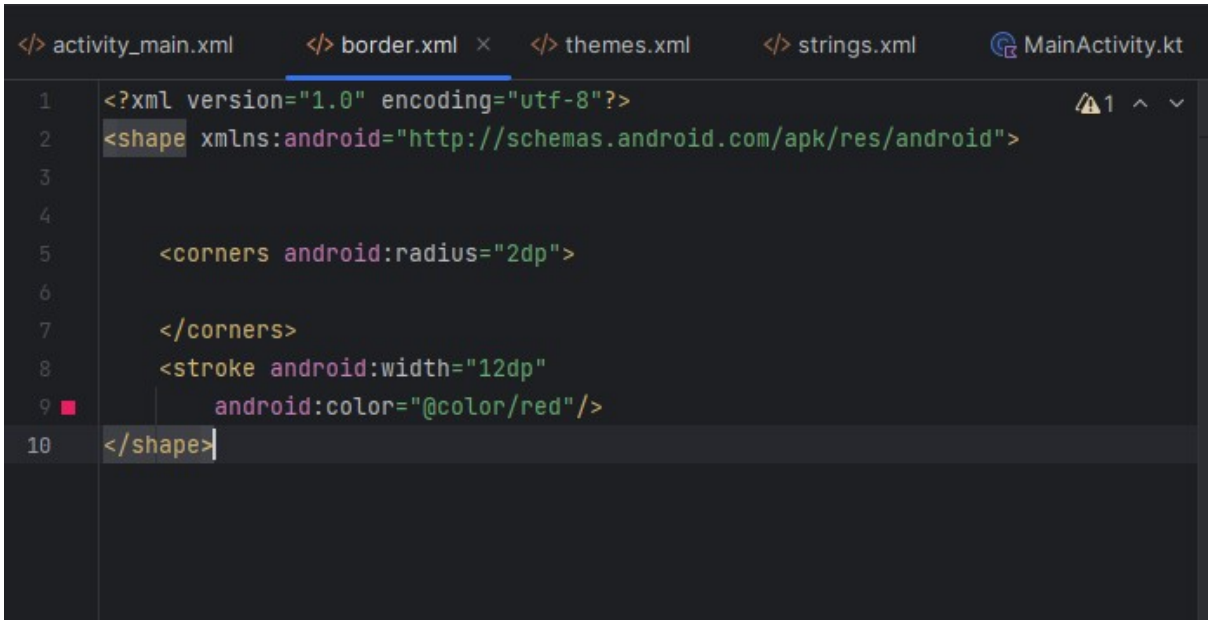

Project
app
manifests
kotlin+java
res
drawable
ic_launcher_background.xml
ic_launcher_foreground.xml
layout
activity_main.xml
mipmap
values
colors.xml
strings.xml
themes (2)
themes.xml
themes.xml (night)
xml
res (generated)
Gradle Scripts

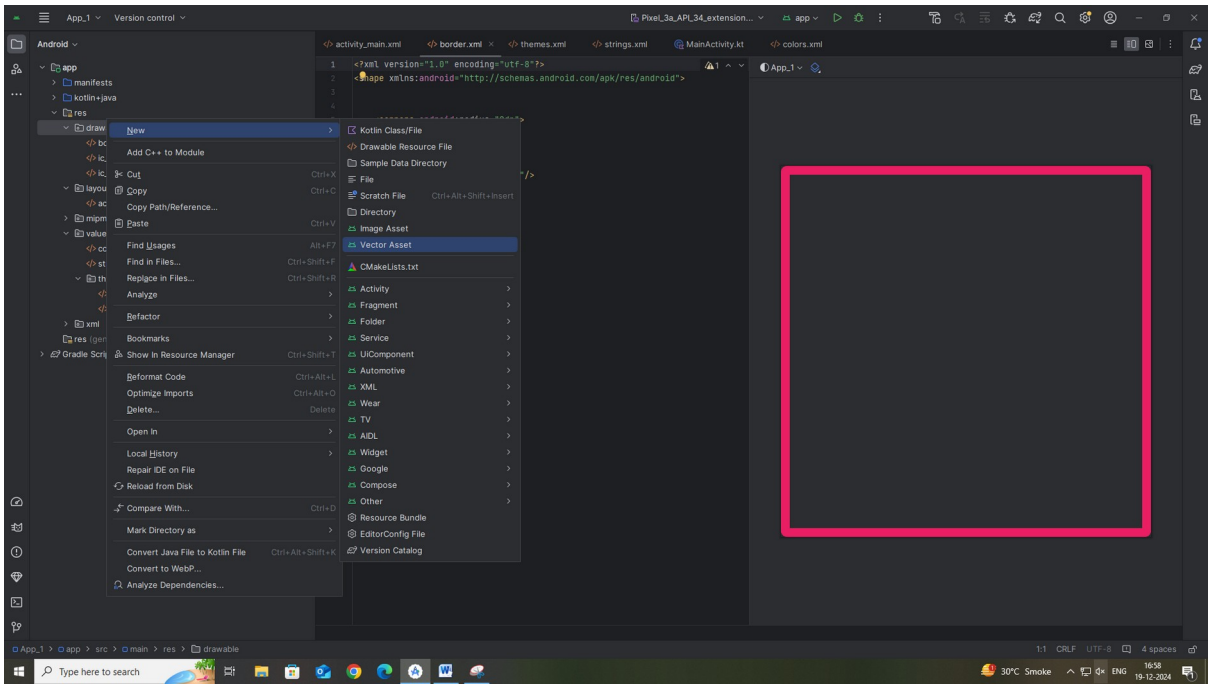

New
Kotlin Class/File
Drawable Resource File
Sample Data Directory
File
Scratch File
Directory
Image Asset
Vector Asset
CMakeLists.txt
Activity
Fragment
Folder
Service
UiComponent
Automotive
XML
Wear
TV
AIDL
Widget
Google
Compose
Other
Resource Bundle
EditorConfig File
Version Catalog
Some Error Occurred
Hello World!

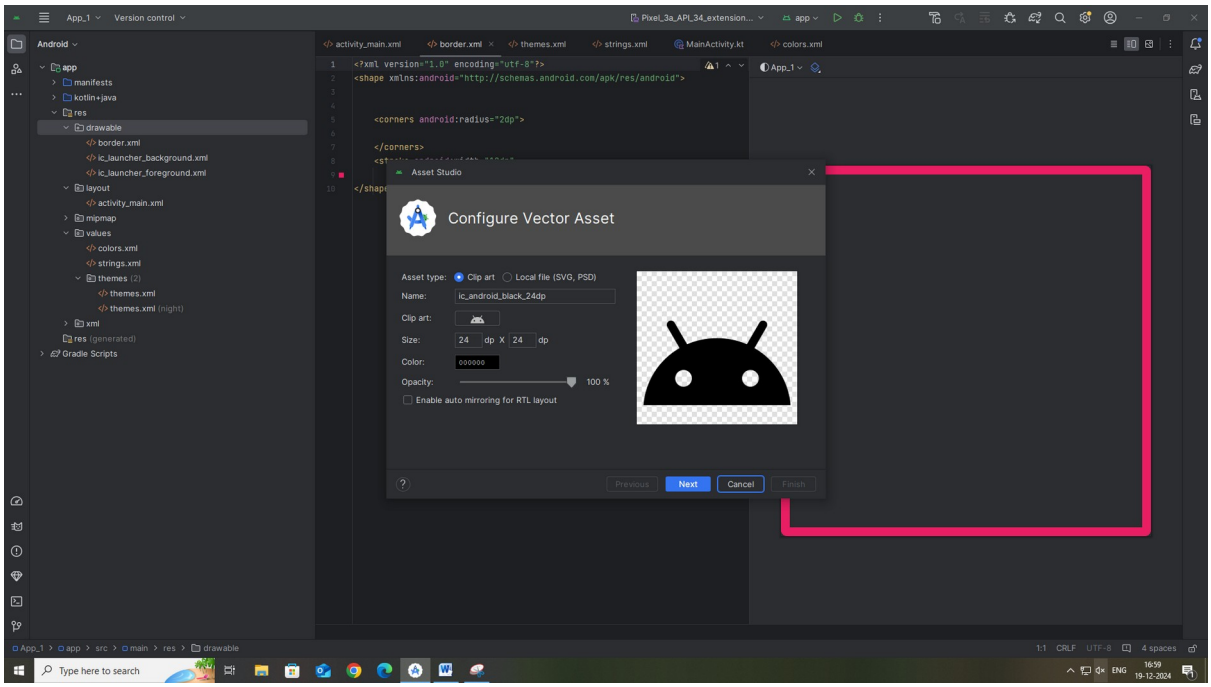

New Resource File
File name: border
Root element: selector
Source set: main src/main/res
Directory name: drawable
Available qualifiers:
Chosen qualifiers:
Country Code
Network Code
Locale
Layout Direction
Smallest Screen Width
Screen Width
Screen Height
Size
Ratio
Nothing to show
OK
Cancel
Some Error Occurred
Hello World!

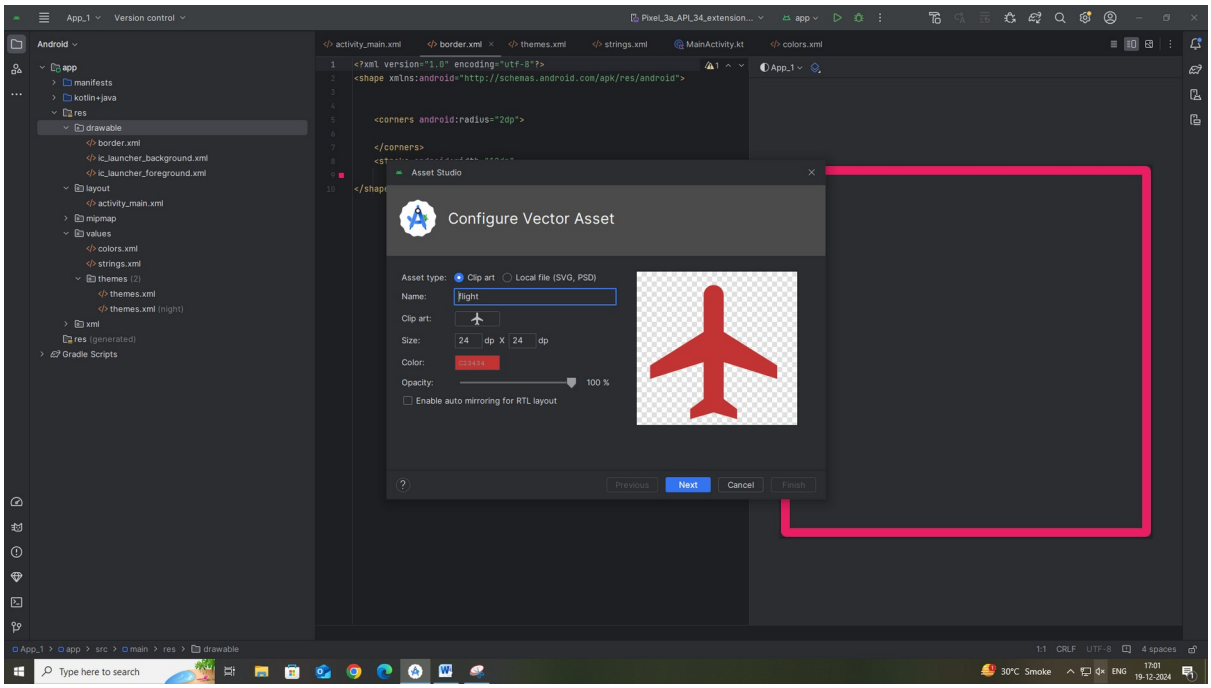

activity_main.xml
border.xml
themes.xml
strings.xml
MainActivity.kt
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
</selector>

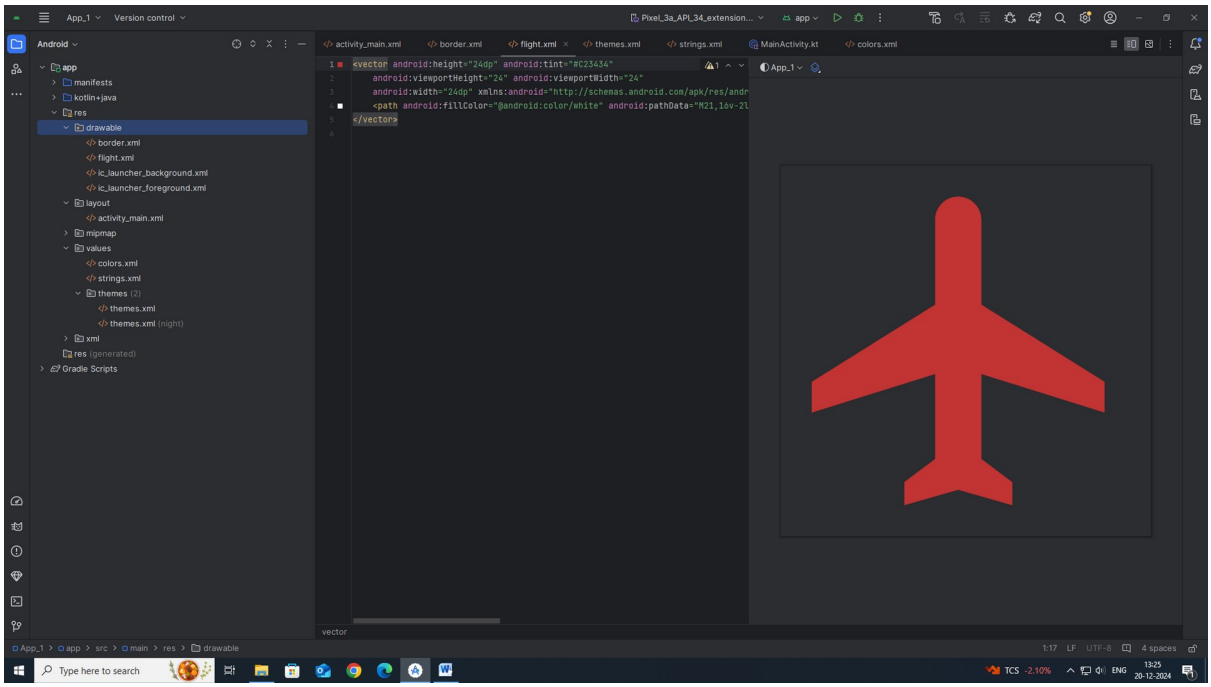

Android
app
manifests
kotlin+java
res
drawable
border.xml
ic_launcher_background.xml
ic_launcher_foreground.xml
layout
activity_main.xml
mipmap
values
colors.xml
strings.xml
themes (2)
themes.xml
themes.xml (night)
xml
res (generated)
Gradle Scripts
activity_main.xml
border.xml
themes.xml
strings.xml
MainActivity.kt
colors.xml
App_1
State Selector
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
<corners android:radius="2dp">
</corners>
<stroke android:width="12dp"
android:color="@color/red"/>
</shape>


activity_main.xml
border.xml
themes.xml
strings.xml
MainActivity.kt
1
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
<corners android:radius="2dp">
</corners>
<stroke android:width="12dp"
android:color="@color/red"/>
</shape>

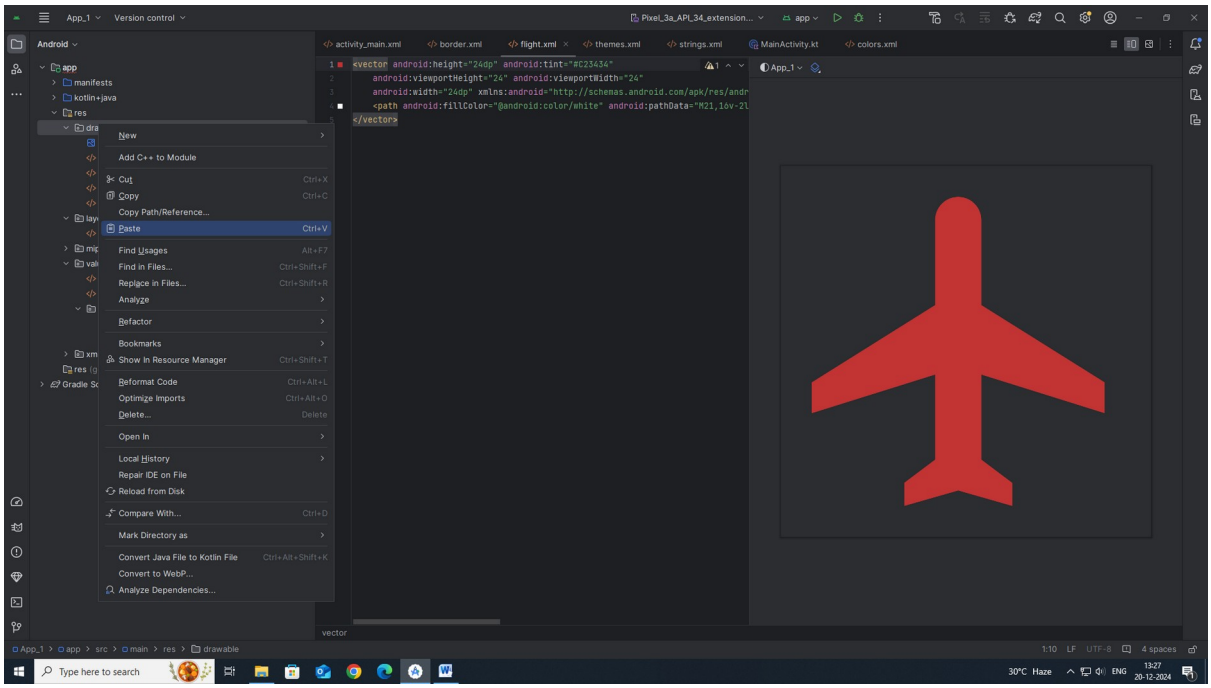

New
Add C++ to Module
Cut
Copy
Copy Path/Reference...
Paste
Find Usages
Find in Files...
Replace in Files...
Analyze
Refactor
Bookmarks
Show In Resource Manager
Reformat Code
Optimize Imports
Delete...
Open In
Local History
Repair IDE on File
Reload from Disk
Compare With...
Mark Directory as
Convert Java File to Kotlin File
Convert to WebP...
Analyze Dependencies...
Kotlin Class/File
Drawable Resource File
Sample Data Directory
File
Scratch File
Directory
Image Asset
Vector Asset
CMakeLists.txt
Activity
Fragment
Folder
Service
UiComponent
Automotive
XML
Wear
TV
AIDL
Widget
Google
Compose
Other
Resource Bundle
EditorConfig File
Version Catalog

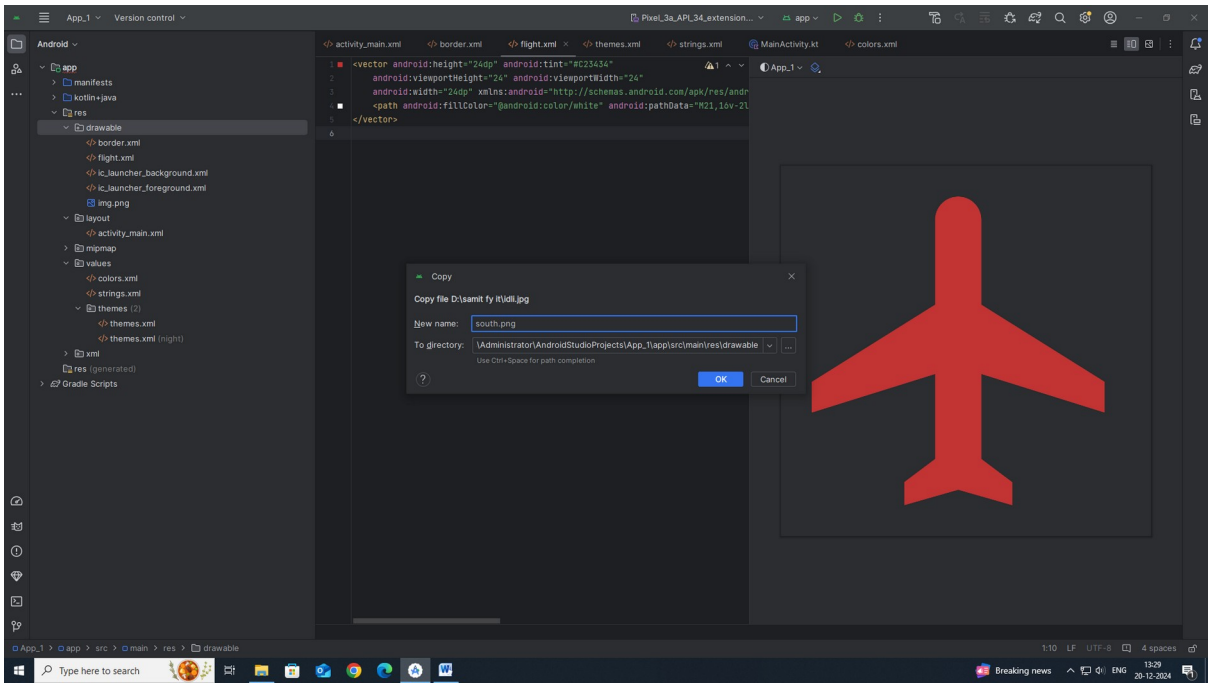

Asset Studio
Configure Vector Asset
Asset type: Clip art Local file (SVG, PSD)
Name: ic_android_black_24dp
Clip art:
Size: 24 dp X 24 dp
Color: 000000
Opacity: 100 %
Enable auto mirroring for RTL layout
Previous
Next
Cancel
Finish

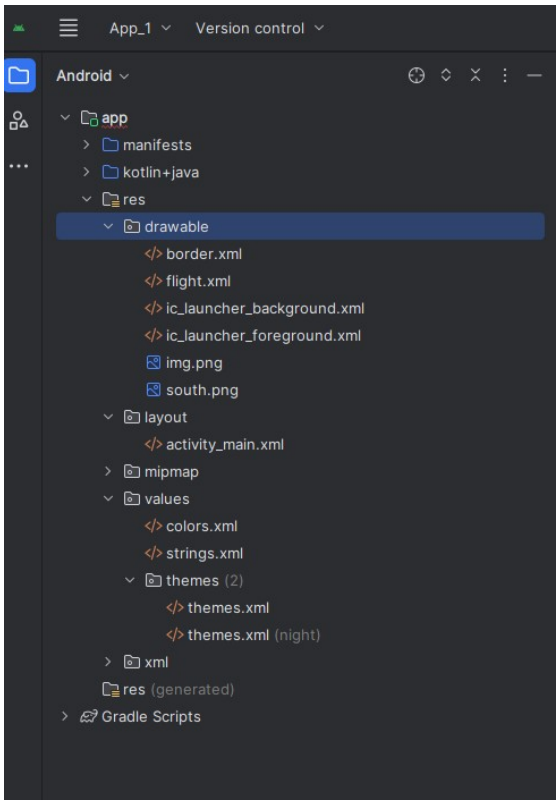




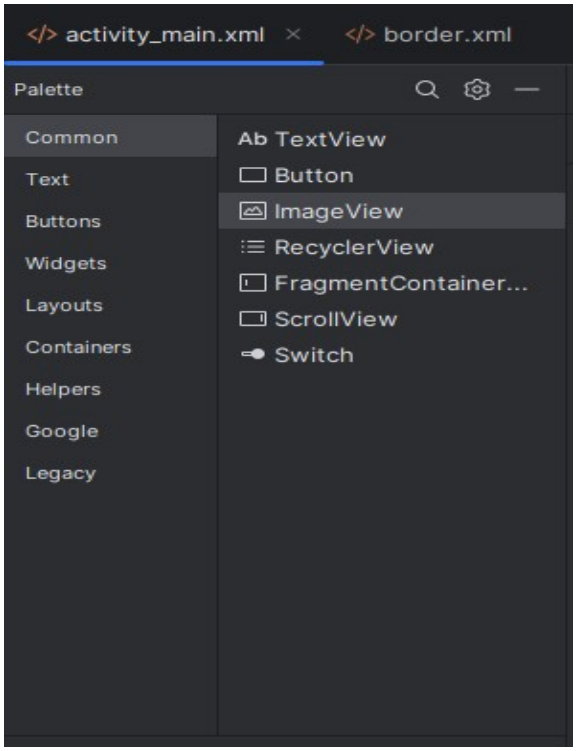
Suppose you want to add image to Drawable folder then copy that image -> then right click on Drawable folder and click on Paste.

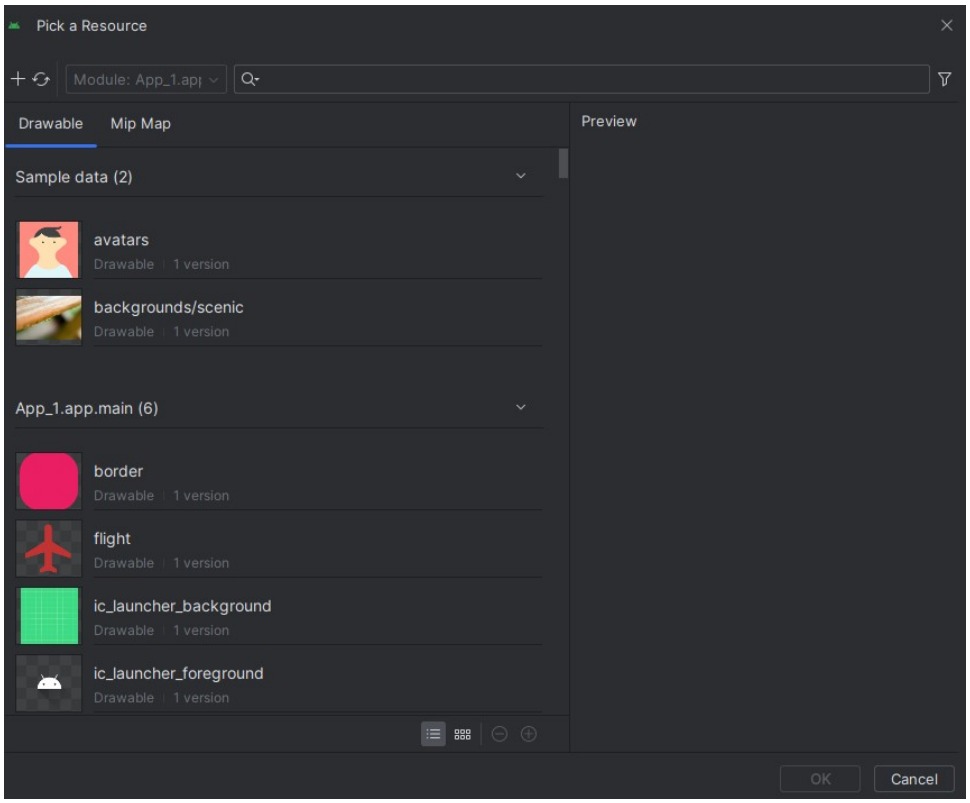

App_1
Version control
Android
app
manifests
kotlin+java
res
drawable
New
Add C++ to Module
Cut
Copy
Copy Path/Reference...
Paste
Find Usages
Find in Files...
Replace in Files...
Analyze
Refactor
Bookmarks
Show in Resource Manager
Reformat Code
Optimize Imports
Delete...
Open In
Local History
Repair IDE on File
Reload from Disk
Compare With...
Mark Directory as
Convert Java File to Kotlin File
Convert to WebP...
Analyze Dependencies...
activity_main.xml
border.xml
flight.xml
themes.xml
strings.xml
MainActivity.kt
colors.xml
<vector android:height="24dp" android:tint="#C23434"
android:viewportHeight="24" android:viewportWidth="24"
android:width="24dp" xmlns:android="http://schemas.android.com/apk/res/andr
<path android:fillColor="@android:color/white" android:pathData="M21,16v-2l
</vector>
vector
App_1 > app > src > main > res > drawable
1:10 LF UTF-8 4 spaces
Type here to search
30°C Haze
ENG
13:27
20-12-2024

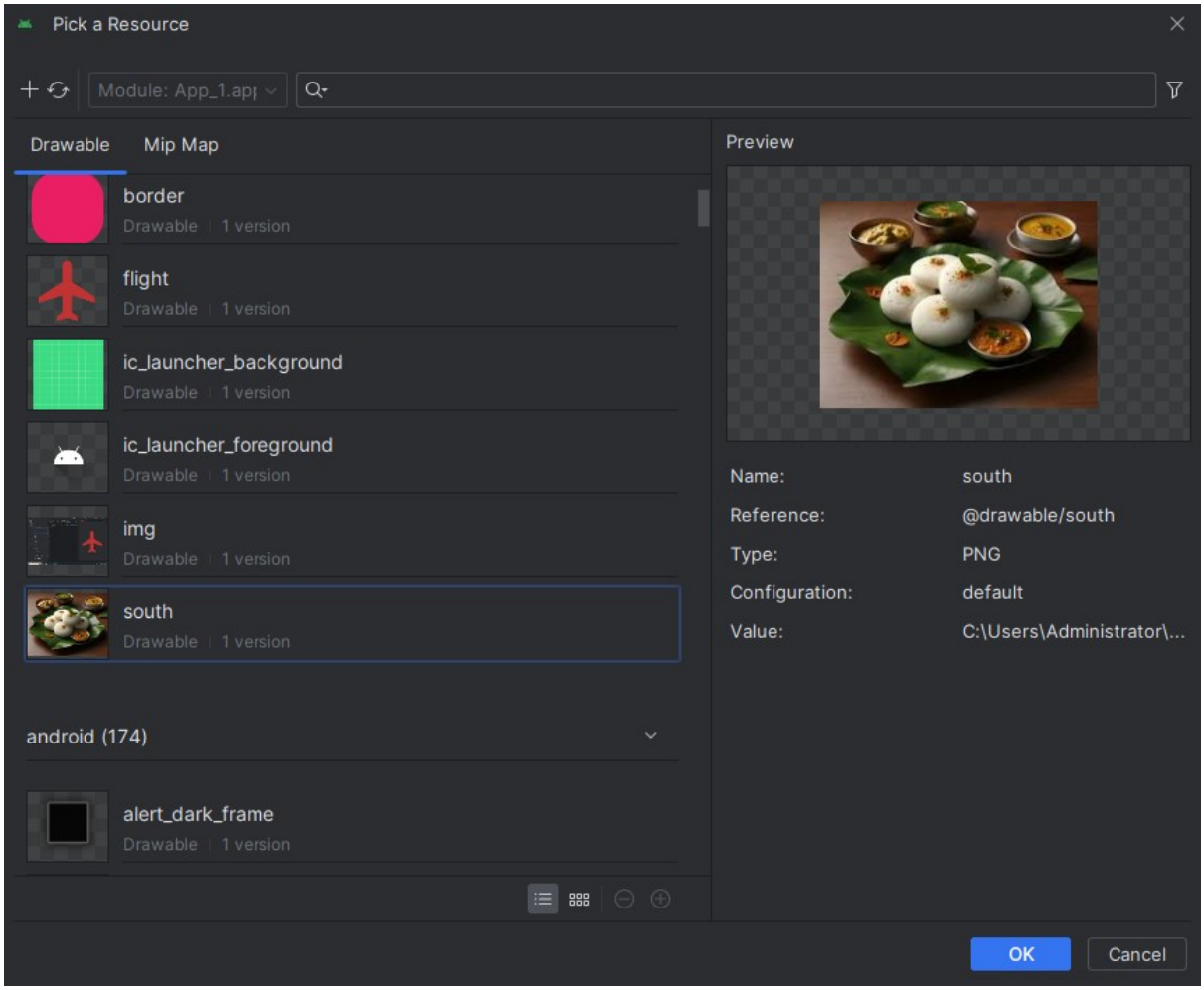

App_1
Version control
Android
app
manifests
kotlin+java
res
drawable
border.xml
flight.xml
ic_launcher_background.xml
ic_launcher_foreground.xml
img.png
layout
activity_main.xml
mipmap
values
colors.xml
strings.xml
themes (2)
themes.xml
themes.xml (night)
xml
res (generated)
Gradle Scripts
activity_main.xml
border.xml
flight.xml
themes.xml
strings.xml
MainActivity.kt
colors.xml
<vector android:height="24dp" android:tint="#C23434"
android:viewportHeight="24" android:viewportWidth="24"
android:width="24dp" xmlns:android="http://schemas.android.com/apk/res/andr
<path android:fillColor="@android:color/white" android:pathData="M21,16v-2l
</vector>
Copy
Copy file D:\samit fy it\Idli.jpg
New name:
south.png
To directory:
\Administrator\AndroidStudioProjects\App_1\app\src\main\res\drawable
Use Ctrl+Space for path completion
OK
Cancel
App_1 > app > src > main > res > drawable
1:10 LF UTF-8 4 spaces
Type here to search
Breaking news
ENG
13:29
20-12-2024

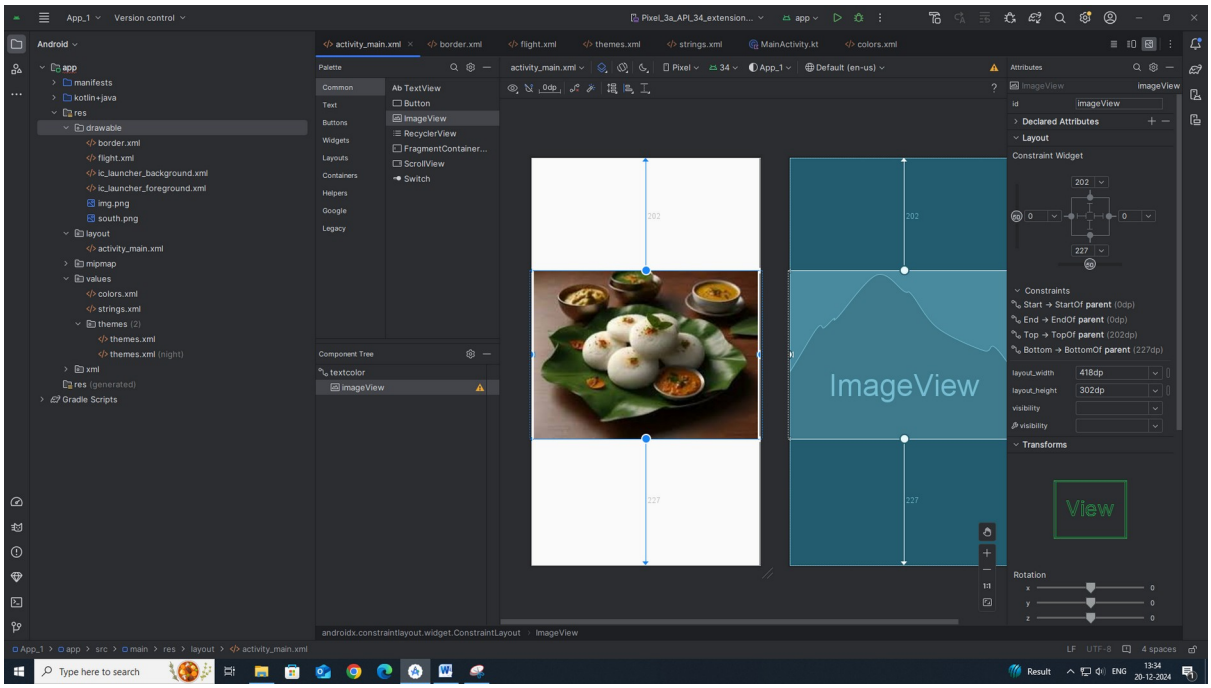


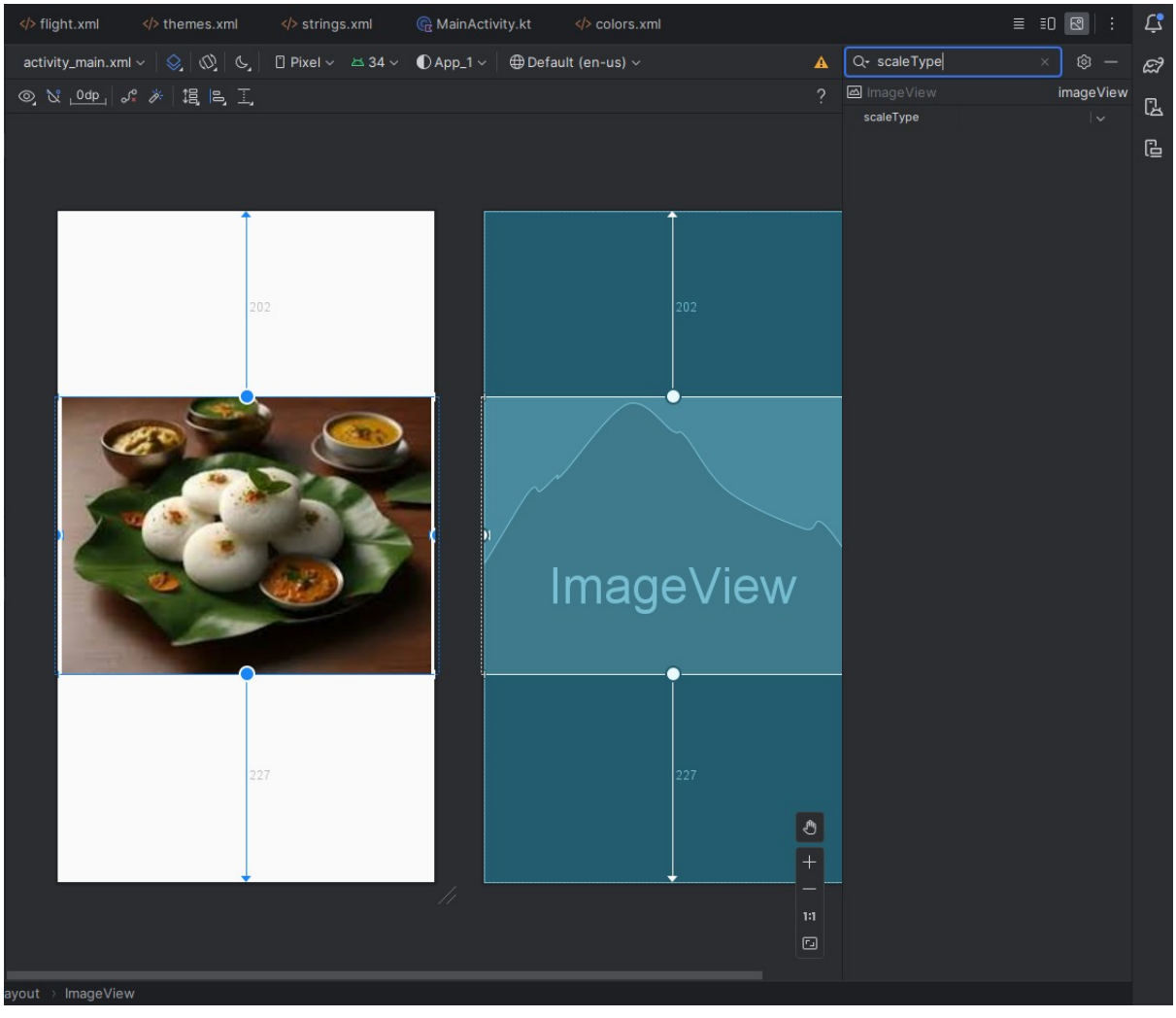
5. Dimension, Image

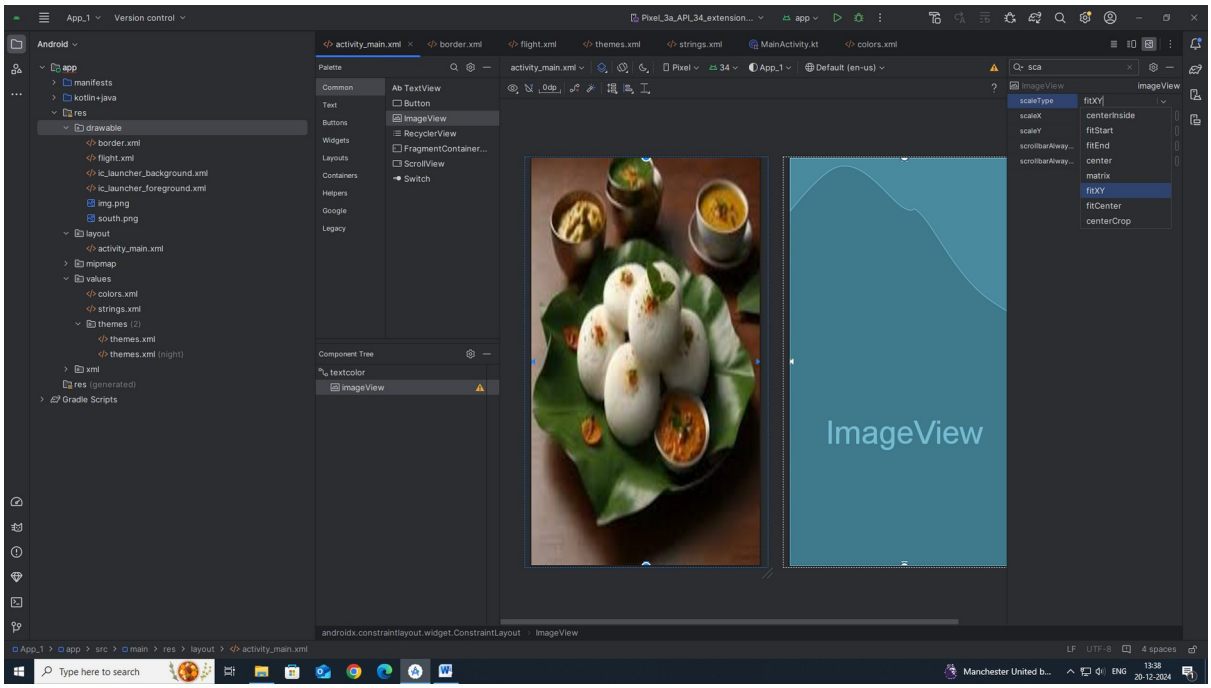


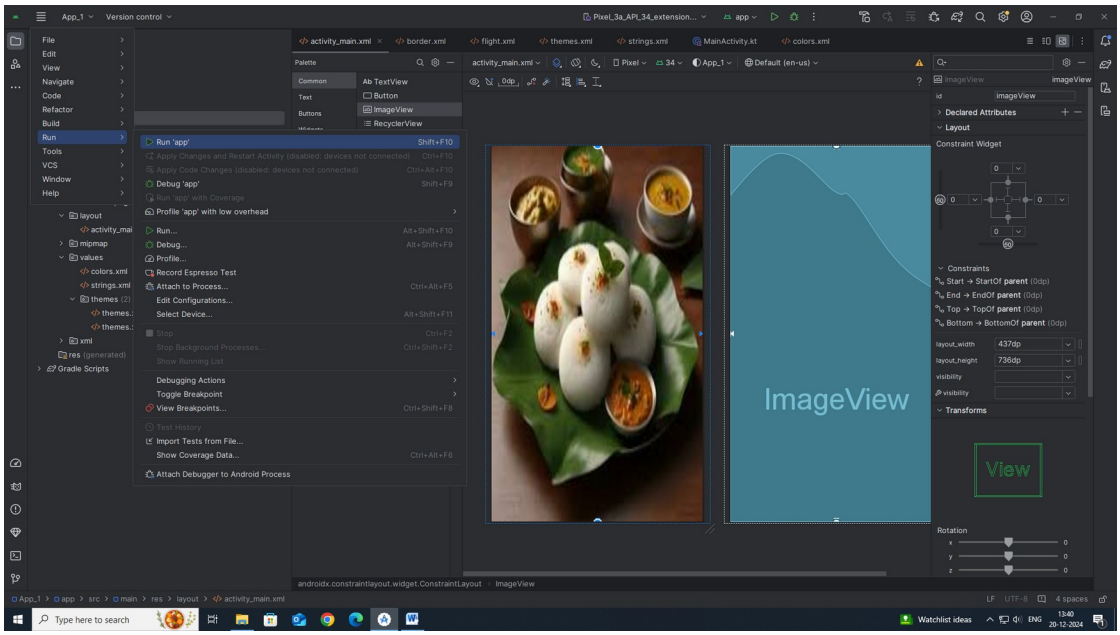

Pick a Resource
Module: App_1.api
Drawable
Mip Map
Preview
Sample data (2)
avatars
Drawable | 1 version
backgrounds/scenic
Drawable | 1 version
App_1.app.main (6)
border
Drawable | 1 version
flight
Drawable | 1 version
ic_launcher_background
Drawable | 1 version
ic_launcher_foreground
Drawable | 1 version
OK
Cancel

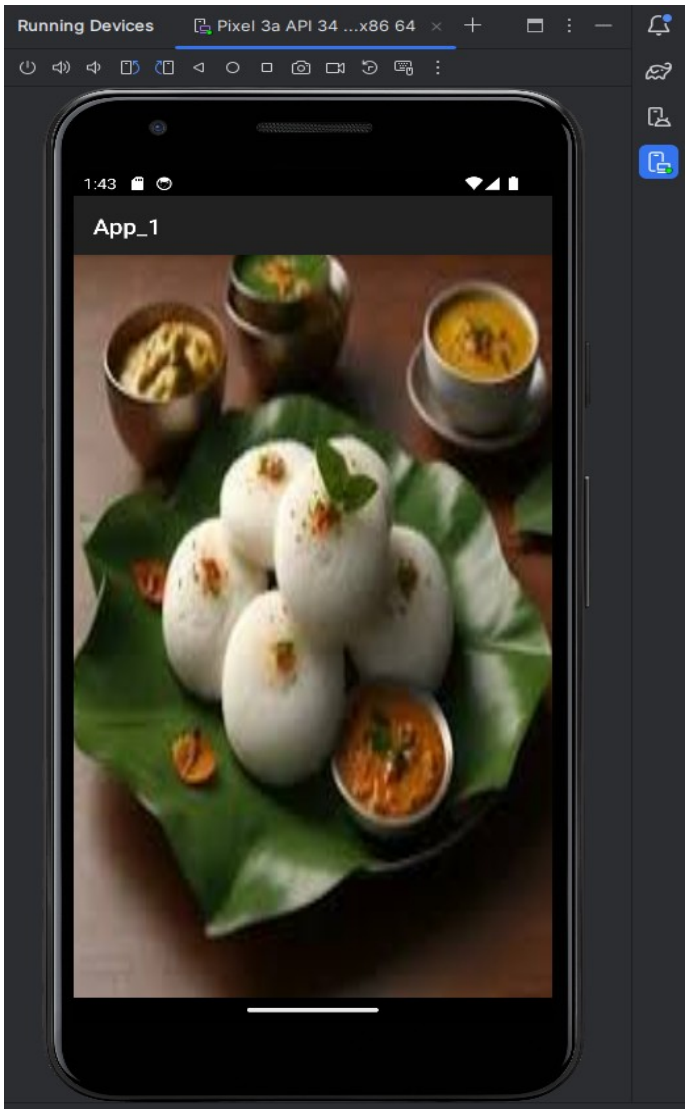

Pick a Resource
Module: App_1.api
Drawable
Mip Map
Preview
border
Drawable | 1 version
flight
Drawable | 1 version
ic_launcher_background
Drawable | 1 version
ic_launcher_foreground
Drawable | 1 version
img
Drawable | 1 version
south
Drawable | 1 version
android (174)
alert_dark_frame
Drawable | 1 version
Name: south
Reference: @drawable/south
Type: PNG
Configuration: default
Value: C:\Users\Administrator\...
OK
Cancel


ImageView
202
227
View


scaleType
ImageView
202
227
ImageView


ImageView


ImageView
View

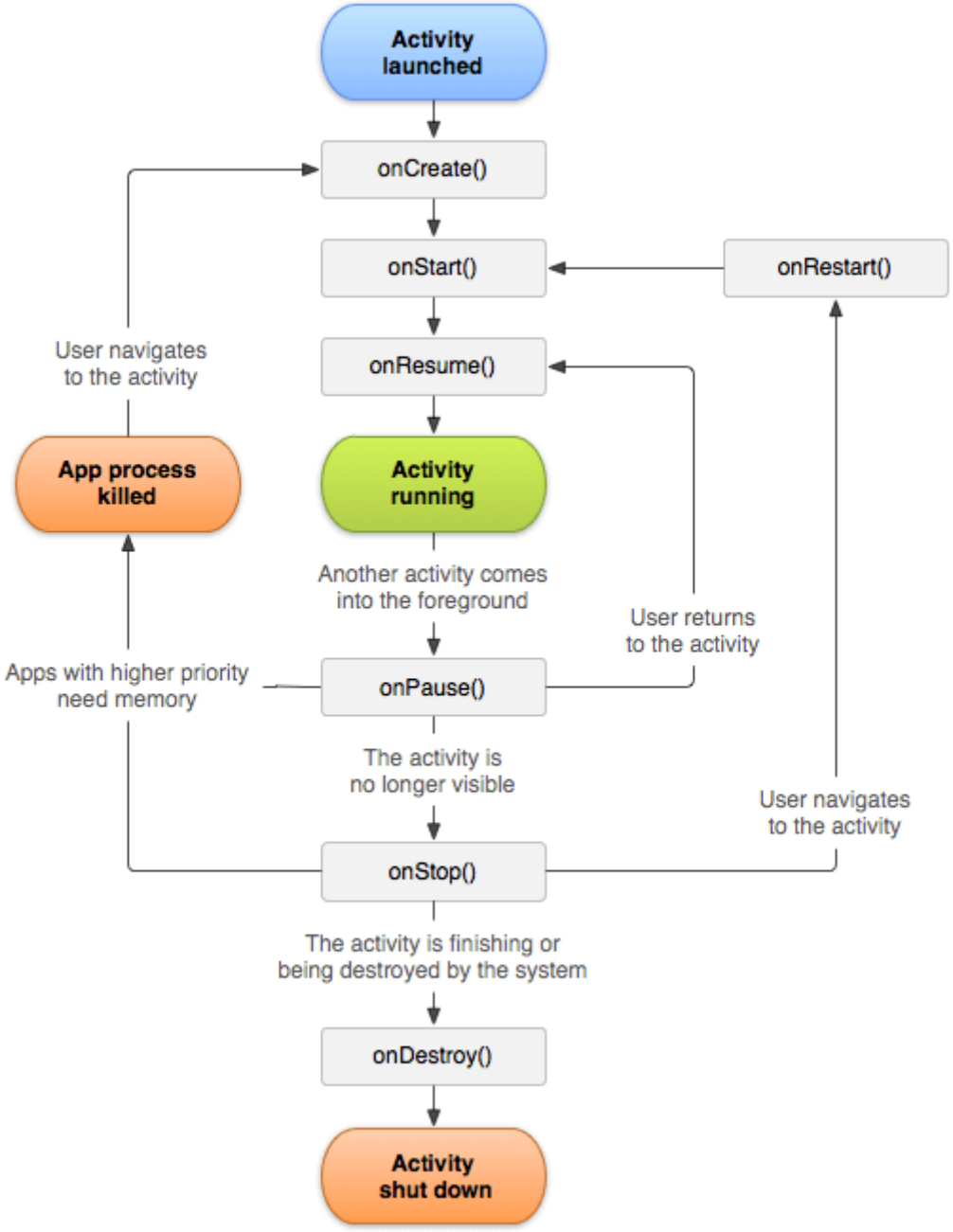

Running Devices
Pixel 3a API 34 ...x86 64
1:43
App_1

PRACTICAL-3

Programming Activities and Fragments

Activity Life Cycle, Activity methods, Multiple Activities, Life Cycle of fragments and multiple fragments.

Activity Lifecycle:



- **onCreate():** Called by the OS when the activity is first created. This is where you initialize any UI elements or data objects. You also have the savedInstanceState of the activity that contains its previously saved state, and you can use it to recreate that state.
- **onStart():** Just before presenting the user with an activity, this method is called. It's always followed by onResume(). In here, you generally should start UI animations, audio based content or anything else that requires the activity's contents to be on screen.
- **onResume():** As an activity enters the foreground, this method is called. Here you have a good place to restart animations, update UI elements, restart camera previews, resume audio/video playback or initialize any components that you release during onPause().
- **onPause():** This method is called before sliding into the background. Here you should stop any visuals or audio associated with the activity such as UI animations, music playback or the camera. This method is followed by onResume() if the activity returns to the foreground or by onStop() if it becomes hidden.
- **onStop():** This method is called right after onPause(), when the activity is no longer visible to the user, and it's a good place to save data that you want to commit to the disk. It's followed by either onRestart(), if this activity is coming back to the foreground, or onDestroy() if it's being released from memory.
- **onRestart():** Called after stopping an activity, but just before starting it again. It's always followed by onStart().
- **onDestroy():** This is the final callback you'll receive from the OS before the activity is destroyed. You can trigger an activity's desctruction by calling finish(), or it can be triggered by the system when the system needs to recoup memory. If your activity includes any background threads or other long-running resources, destruction could lead to a memory leak if they're not released, so you need to remember to stop these processes here as well.

Go to activity_main.xml file→ change layout to RelativeLayout

Comment TextView

Then go to Design tab-→Drag & Drop Button

(text – AboutUs,

layout_centerHorizontal—True,

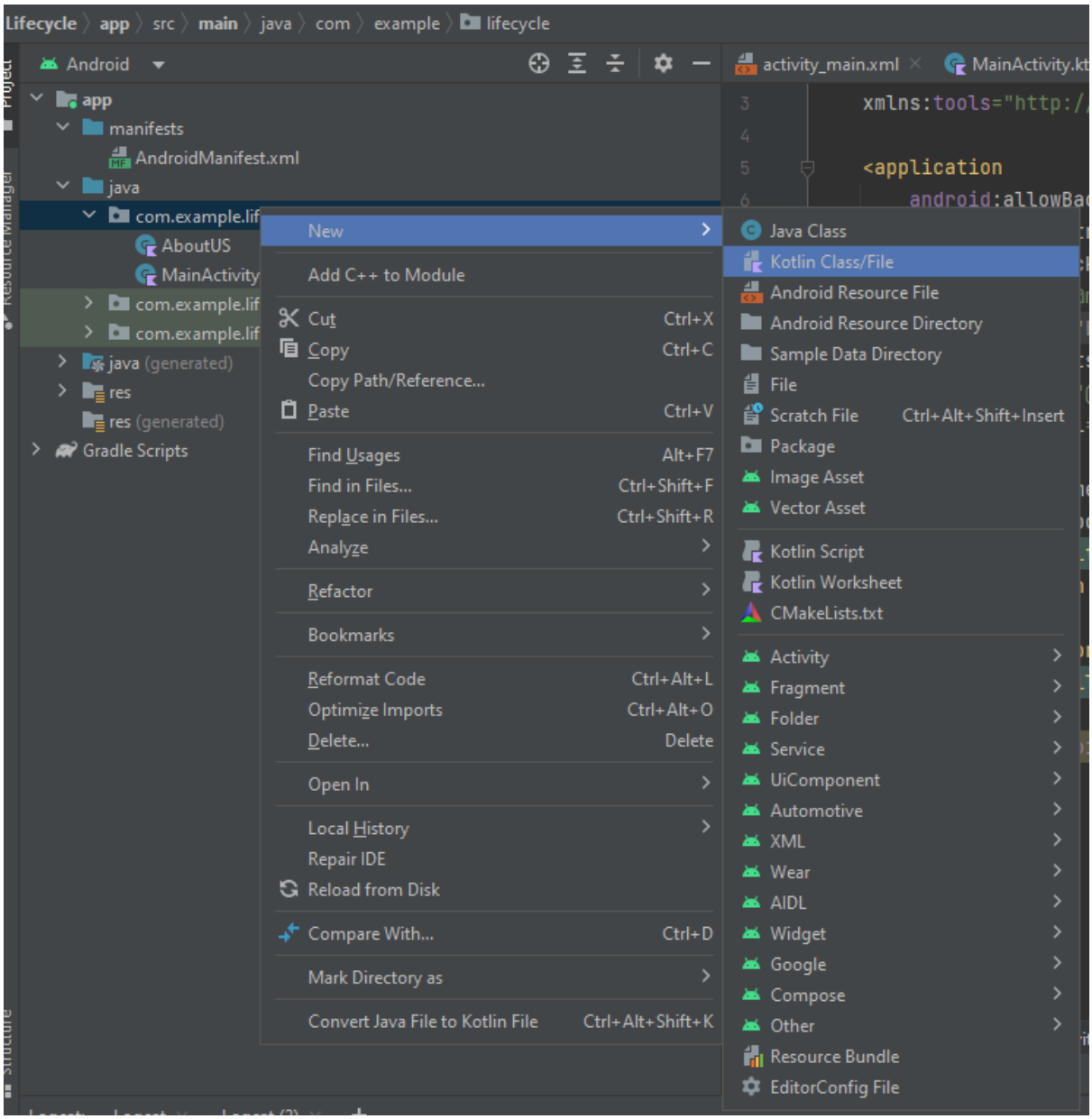
layout_Margin→ select TOP and set value=80dp)

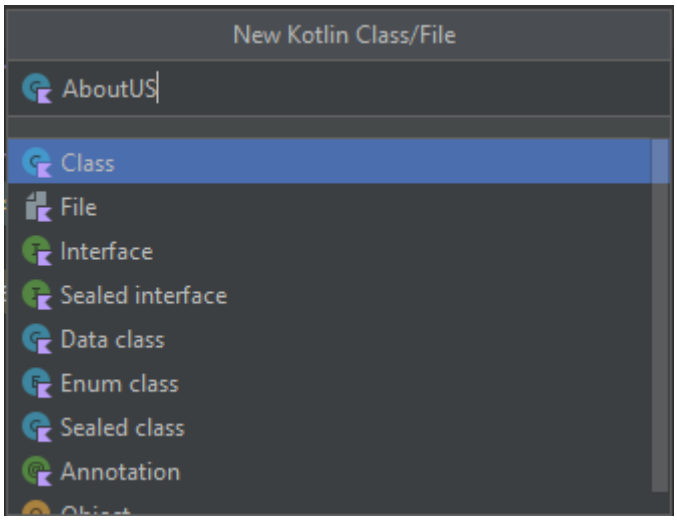
activity_main.xml | MainActivity.kt | AboutUS.kt | AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginTop="80dp"
    tools:context=".MainActivity">

    <!--<TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />-->

    <Button
        android:id="@+id/about_us"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="80dp"
        android:text="AboutUs" />
</RelativeLayout>
```

Change button id and id=”about_us”

Create AboutUs.kt file





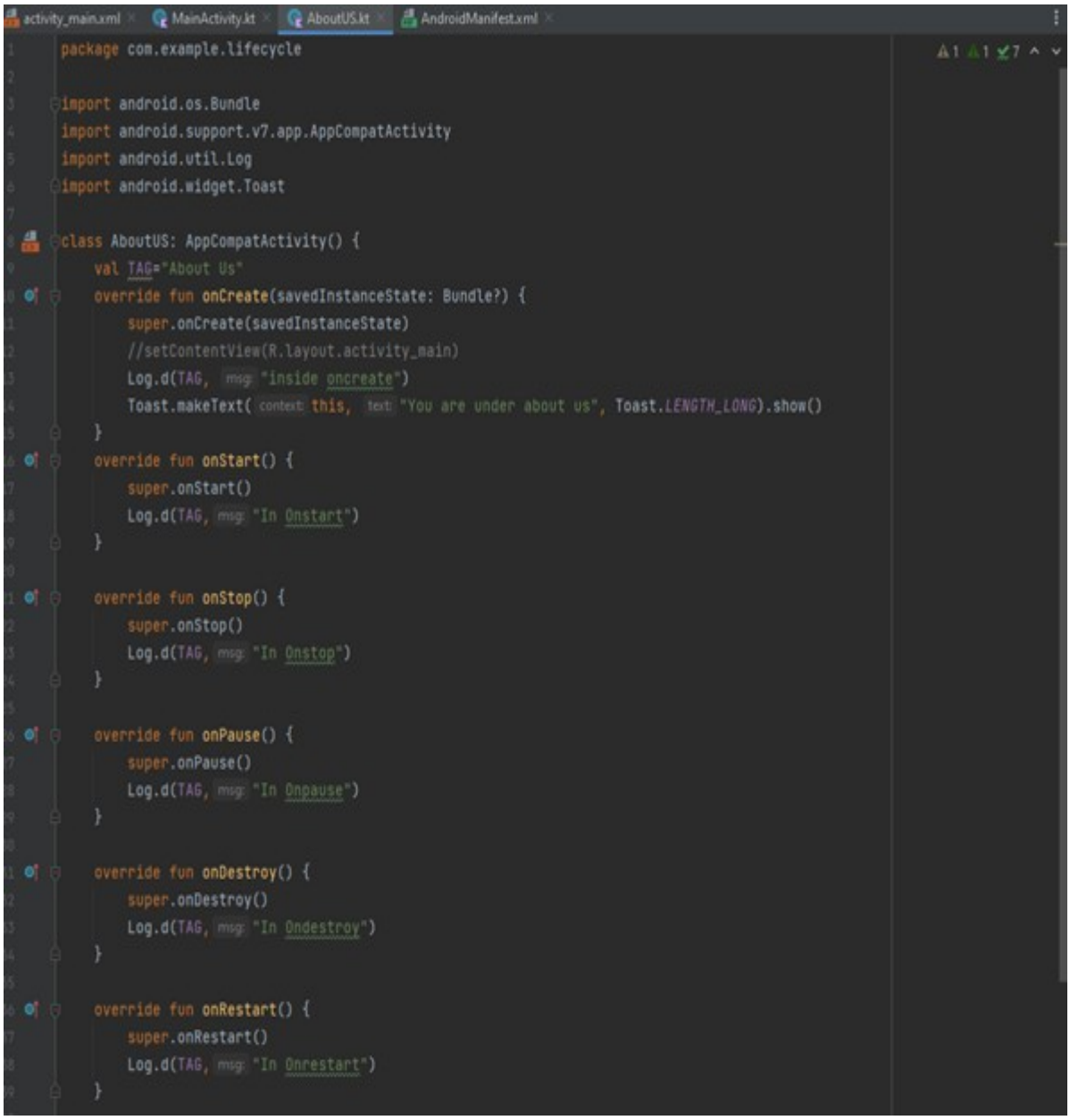
Go to MainActivity.kt

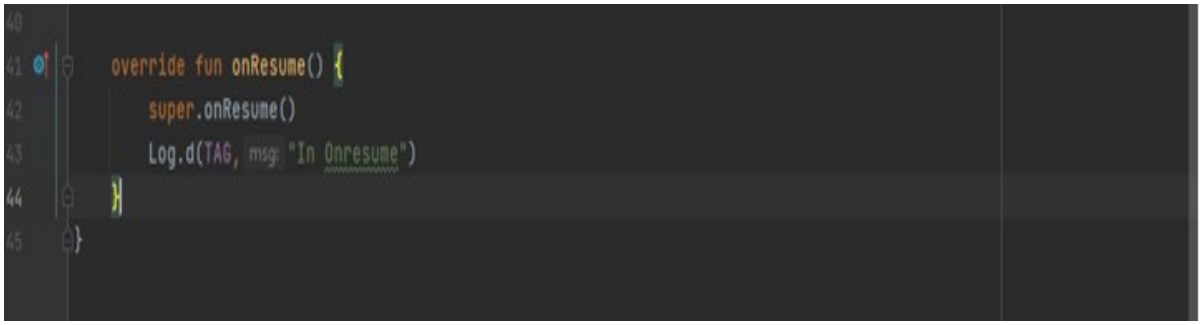
activity_main.xml | MainActivity.kt | AboutUS.kt | AndroidManifest.xml

```
package com.example.lifecycle

import ...

class MainActivity : AppCompatActivity() {
    val TAG="Main Activity"
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        Log.d(TAG, msg: "In Oncreate")
        val about_us =findViewById<Button>(R.id.about_us)
        about_us.setOnClickListener { it: View!
            val i =Intent( packageContext: this,AboutUS::class.java)
            startActivity(i)
        }
    }

    override fun onStart() {
        super.onStart()
        Log.d(TAG, msg: "In Onstart")
    }

    override fun onStop() {
        super.onStop()
        Log.d(TAG, msg: "In Onstop")
    }

    override fun onPause() {
        super.onPause()
        Log.d(TAG, msg: "In Onpause")
    }

    override fun onDestroy() {
        super.onDestroy()
        Log.d(TAG, msg: "In Ondestroy")
    }
```

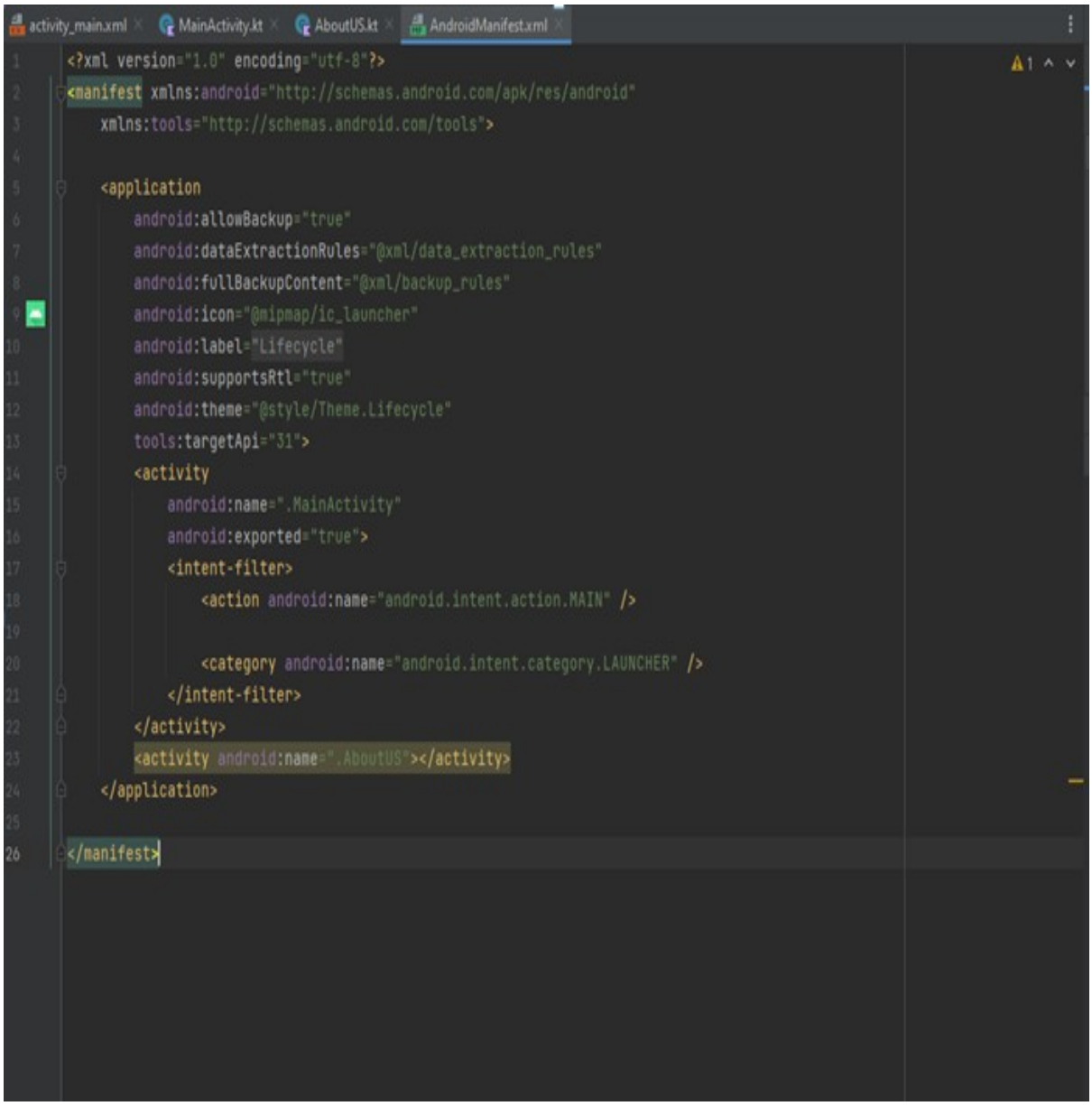
```
    override fun onRestart() {
        super.onRestart()
        Log.d(TAG, msg: "In Onrestart")
    }

    override fun onResume() {
        super.onResume()
        Log.d(TAG, msg: "In Onresume")
    }
}
```

Now go to AboutUs.kt file and add below code.





Go to AndroidMainfest.xml file.



Now Run your Application.

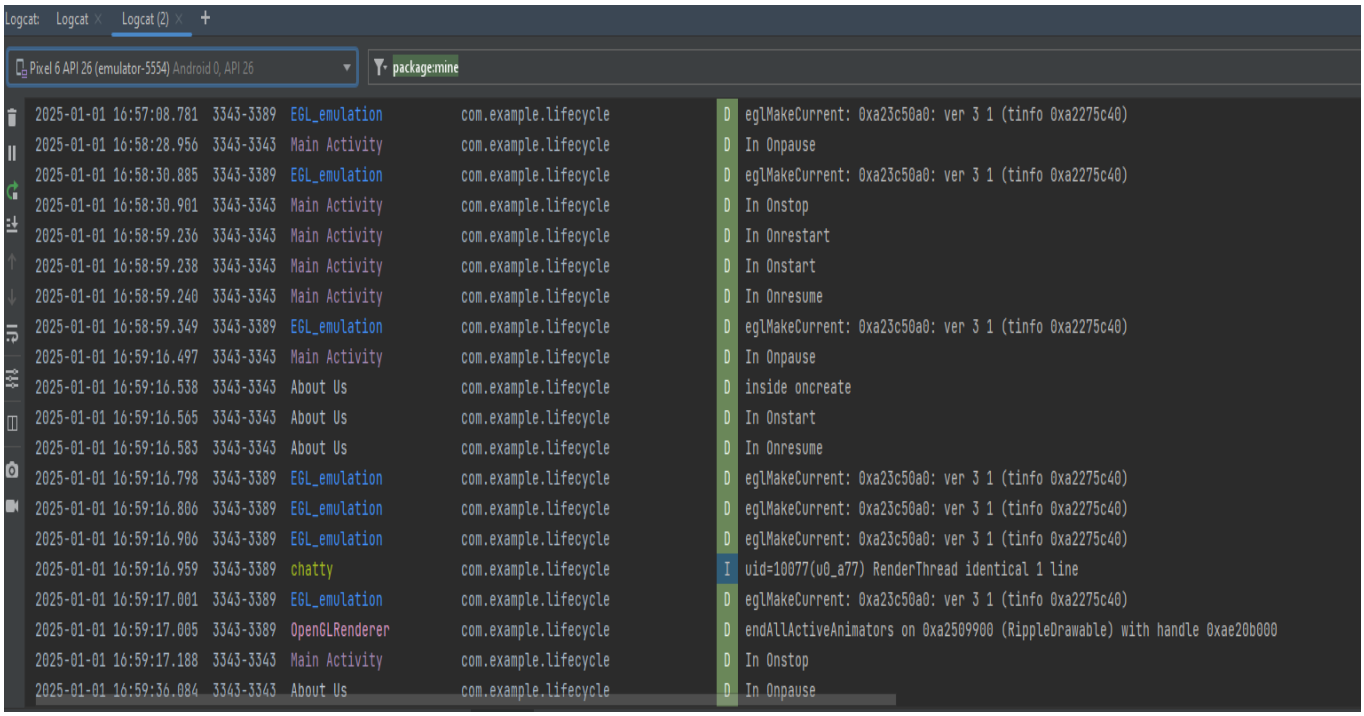
Logcat Logcat Logcat (2)

Pixel 6 API 26 (emulator-5554) Android 0, API 26

package:mine

```
2025-01-01 16:57:08.514  3343-3343  zygote              com.example.lifecycle  I  at void com.android.internal.os.ZygoteInit.main(java.lang.String[]) (ZygoteInit.java:767)
2025-01-01 16:57:08.514  3343-3343  zygote              com.example.lifecycle  I
2025-01-01 16:57:08.540  3343-3343  Main Activity       com.example.lifecycle  D  In Oncreate
2025-01-01 16:57:08.545  3343-3343  Main Activity       com.example.lifecycle  D  In Onstart
2025-01-01 16:57:08.548  3343-3343  Main Activity       com.example.lifecycle  D  In Onresume
2025-01-01 16:57:08.574  3343-3389  OpenGLRenderer      com.example.lifecycle  D  HWUI GL Pipeline
2025-01-01 16:57:08.593  3343-3389  <no-tag>            com.example.lifecycle  D  HostConnection::get() New Host Connection established 0xa4b64380, tid 3389
2025-01-01 16:57:08.607  3343-3389  OpenGLRenderer      com.example.lifecycle  I  Initialized EGL, version 1.4
2025-01-01 16:57:08.607  3343-3389  OpenGLRenderer      com.example.lifecycle  D  Swap behavior 1
2025-01-01 16:57:08.610  3343-3389  OpenGLRenderer      com.example.lifecycle  W  Failed to choose config with EGL_SWAP_BEHAVIOR_PRESERVED, retrying without...
2025-01-01 16:57:08.610  3343-3389  OpenGLRenderer      com.example.lifecycle  D  Swap behavior 0
2025-01-01 16:57:08.622  3343-3389  EGL_emulation       com.example.lifecycle  D  eglCreateContext: 0xa23c50a0: maj 3 min 1 rcv 4
2025-01-01 16:57:08.630  3343-3389  EGL_emulation       com.example.lifecycle  D  eglMakeCurrent: 0xa23c50a0: ver 3 1 (tinfo 0xa2275c40)
2025-01-01 16:57:08.631  3343-3389  eglCodecCommon      com.example.lifecycle  E  glUtilsParamSize: unknow param 0x000082da
2025-01-01 16:57:08.631  3343-3389  eglCodecCommon      com.example.lifecycle  E  glUtilsParamSize: unknow param 0x000082da
2025-01-01 16:57:08.781  3343-3389  EGL_emulation       com.example.lifecycle  D  eglMakeCurrent: 0xa23c50a0: ver 3 1 (tinfo 0xa2275c40)
2025-01-01 16:58:28.956  3343-3343  Main Activity       com.example.lifecycle  D  In Onpause
2025-01-01 16:58:30.885  3343-3389  EGL_emulation       com.example.lifecycle  D  eglMakeCurrent: 0xa23c50a0: ver 3 1 (tinfo 0xa2275c40)
2025-01-01 16:58:30.901  3343-3343  Main Activity       com.example.lifecycle  D  In Onstop
2025-01-01 16:58:59.236  3343-3343  Main Activity       com.example.lifecycle  D  In Onrestart
```

Now click on “ABOUTUS” Button.



PRACTICAL-4

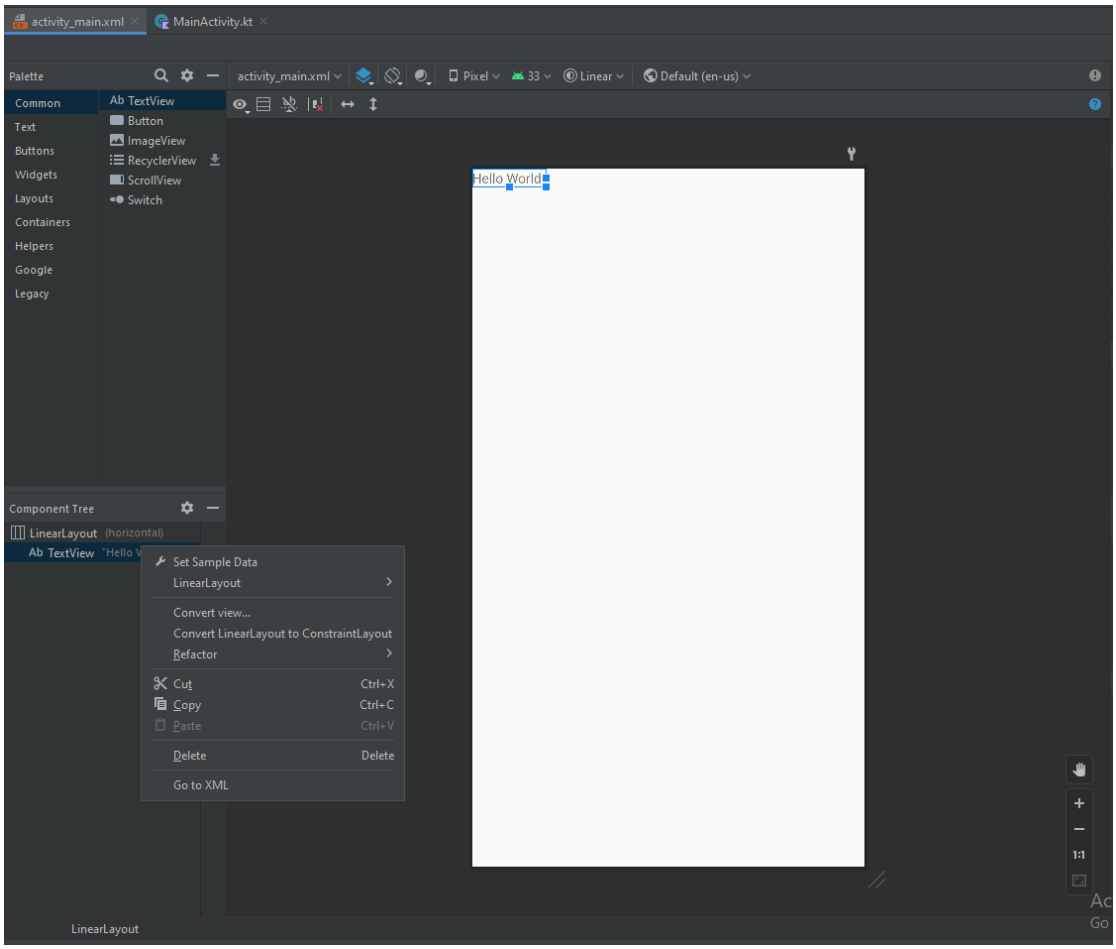
Programs related to different Layouts

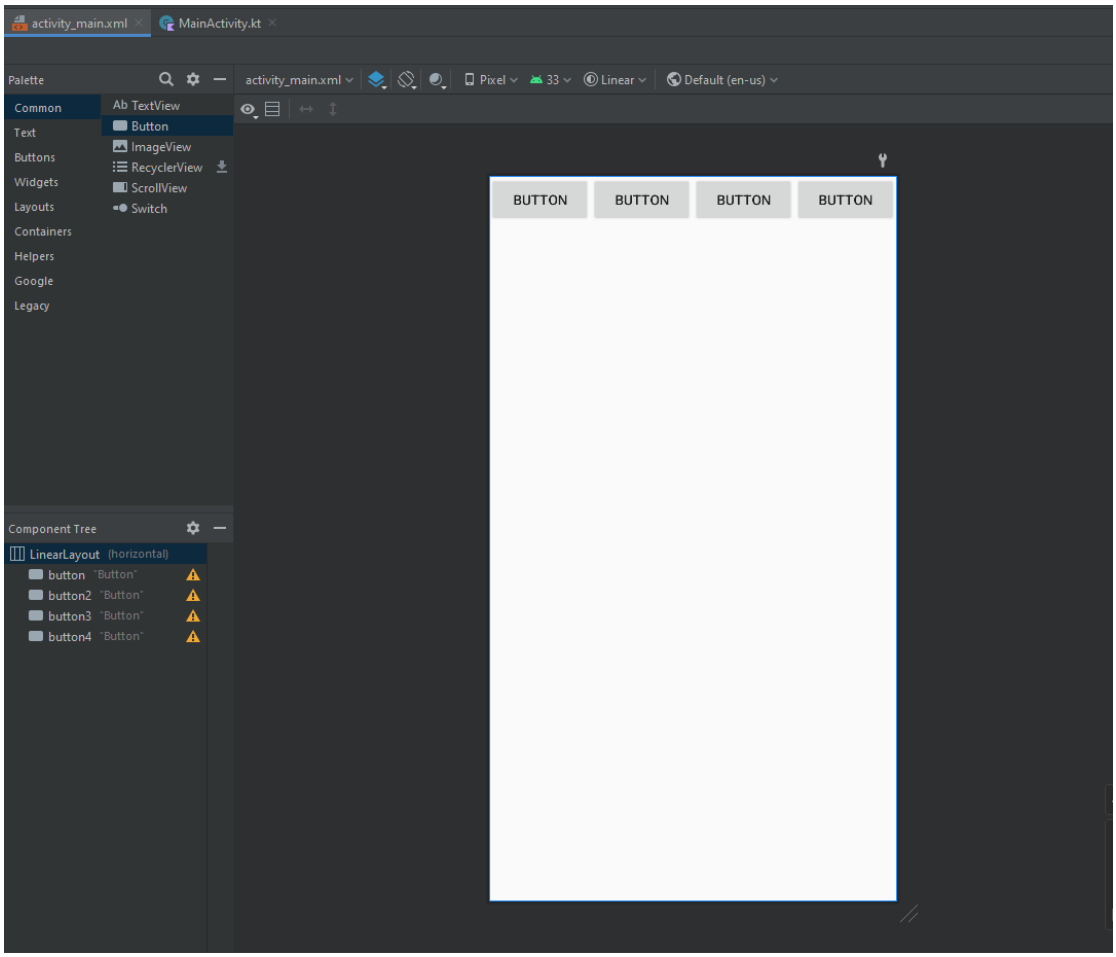
Coordinate, Linear, Relative, Table, Absolute, Frame, List View, Grid View.

1. Linear Layout

activity_main.xml | MainActivity.kt

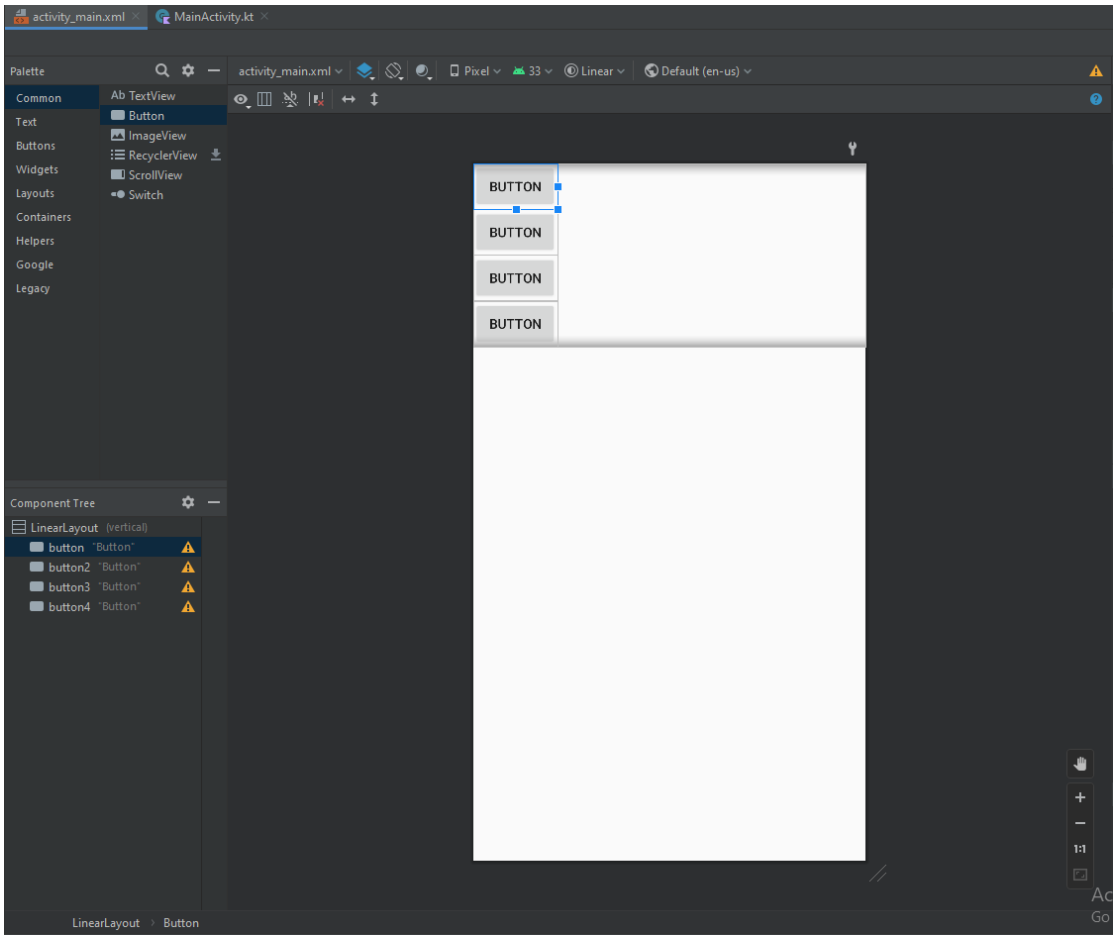
```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</LinearLayout>
```


activity_main.xml
MainActivity.kt
Palette
Common
Text
Buttons
Widgets
Layouts
Containers
Helpers
Google
Legacy
Ab TextView
Button
ImageView
RecyclerView
ScrollView
Switch
Pixel
33
Linear
Default (en-us)
Hello World
Component Tree
LinearLayout (horizontal)
Ab TextView
Set Sample Data
LinearLayout
Convert view...
Convert LinearLayout to ConstraintLayout
Refactor
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Delete Delete
Go to XML
1:1
LinearLayout


activity_main.xml
MainActivity.kt
Palette
Common
Text
Buttons
Widgets
Layouts
Containers
Helpers
Google
Legacy
Ab TextView
Button
ImageView
RecyclerView
ScrollView
Switch
activity_main.xml
Pixel
33
Linear
Default (en-us)
BUTTON
BUTTON
BUTTON
BUTTON
Component Tree
LinearLayout (horizontal)
button "Button"
button2 "Button"
button3 "Button"
button4 "Button"

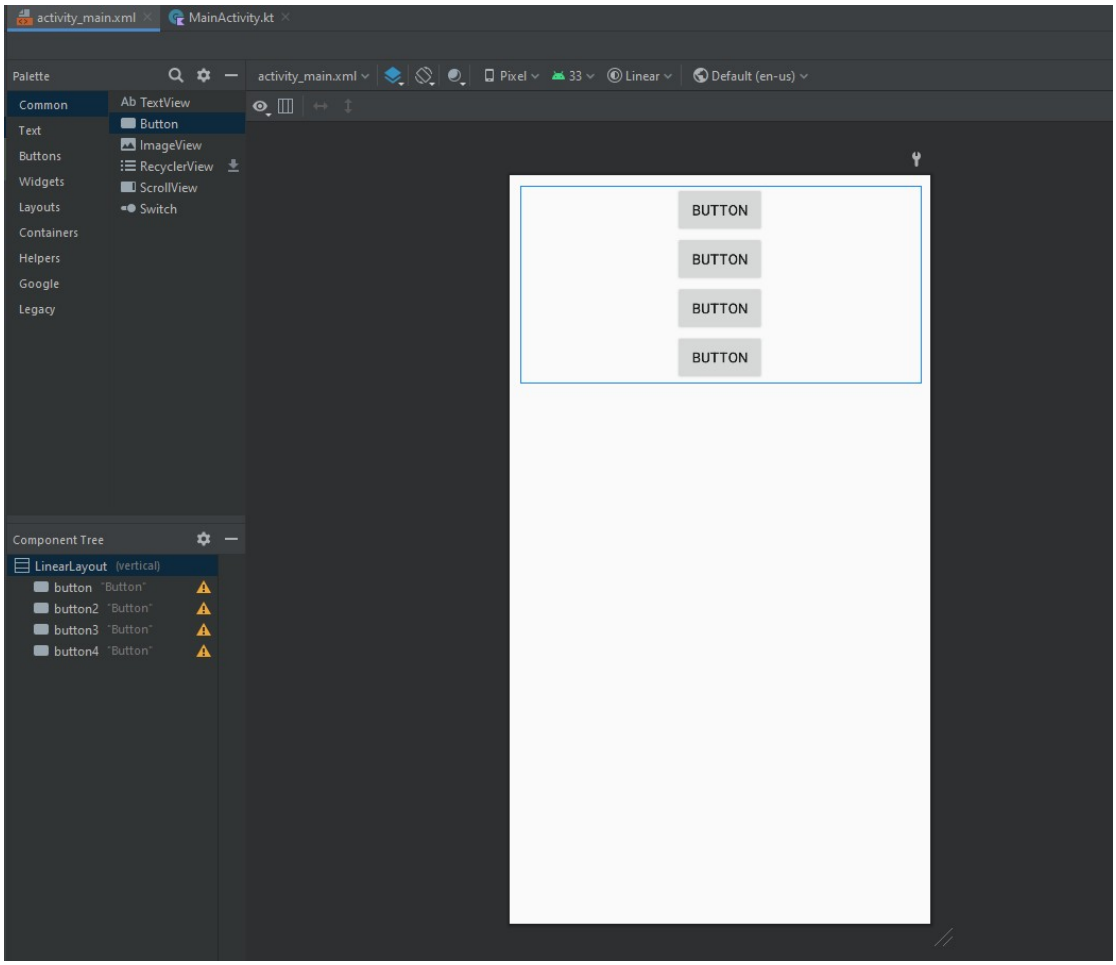
activity_main.xml | MainActivity.kt

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="Button" />

    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="Button" />

    <Button
        android:id="@+id/button3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="Button" />

    <Button
        android:id="@+id/button4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="Button" />
</LinearLayout>
```


activity_main.xml
MainActivity.kt
Palette
Common
Text
Buttons
Widgets
Layouts
Containers
Helpers
Google
Legacy
Ab TextView
Button
ImageView
RecyclerView
ScrollView
Switch
activity_main.xml
Pixel
33
Linear
Default (en-us)
BUTTON
BUTTON
BUTTON
BUTTON
Component Tree
LinearLayout (vertical)
button "Button"
button2 "Button"
button3 "Button"
button4 "Button"
1:1
LinearLayout
Button

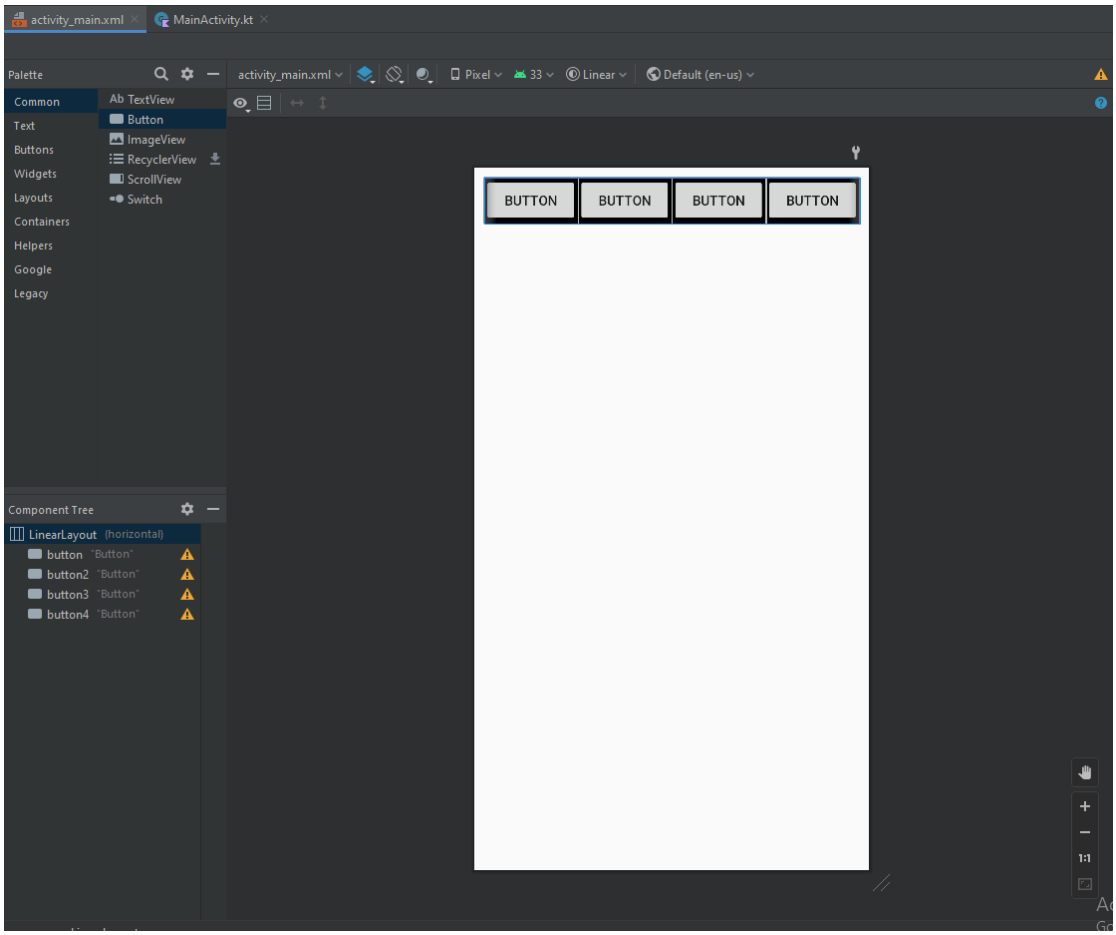
activity_main.xml | MainActivity.kt

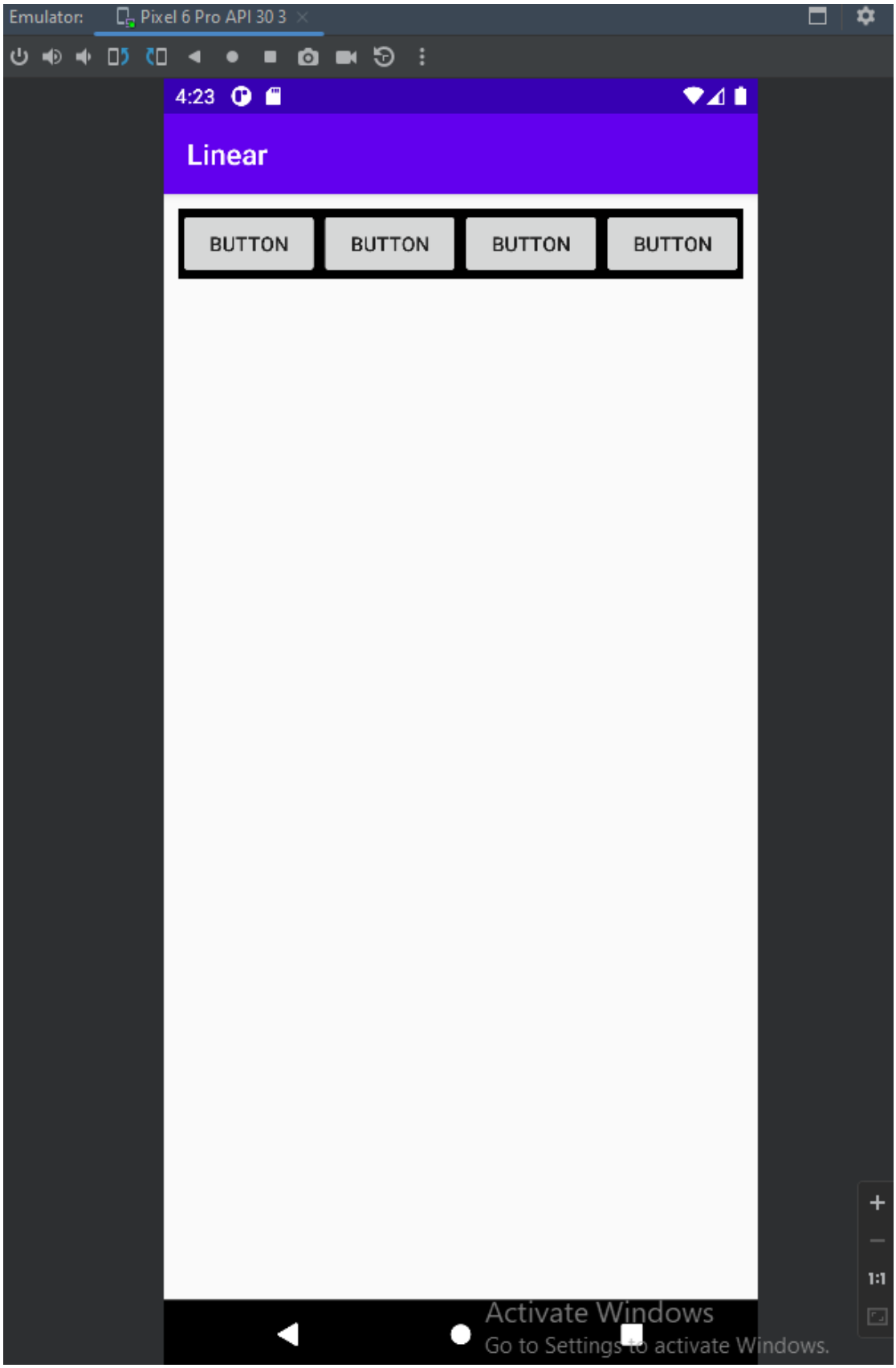
```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="10dp"
    android:gravity="center_horizontal"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="Button" />

    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="Button" />

    <Button
        android:id="@+id/button3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="Button" />

    <Button
        android:id="@+id/button4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="Button" />
```


activity_main.xml
MainActivity.kt
Palette
Common
Text
Buttons
Widgets
Layouts
Containers
Helpers
Google
Legacy
Ab TextView
Button
ImageView
RecyclerView
ScrollView
Switch
activity_main.xml
Pixel
33
Linear
Default (en-us)
BUTTON
BUTTON
BUTTON
BUTTON
Component Tree
LinearLayout (vertical)
button "Button"
button2 "Button"
button3 "Button"
button4 "Button"

activity_main.xml | MainActivity.kt

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="10dp"
    android:background="@color/black"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="Button" />

    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="Button" />

    <Button
        android:id="@+id/button3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="Button" />

    <Button
        android:id="@+id/button4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="Button" />
</LinearLayout>
```


activity_main.xml
MainActivity.kt
Palette
Common
Text
Buttons
Widgets
Layouts
Containers
Helpers
Google
Legacy
Ab TextView
Button
ImageView
RecyclerView
ScrollView
Switch
activity_main.xml
Pixel
33
Linear
Default (en-us)
BUTTON
BUTTON
BUTTON
BUTTON
Component Tree
LinearLayout (horizontal)
button "Button"
button2 "Button"
button3 "Button"
button4 "Button"
1:1


Emulator:
Pixel 6 Pro API 30 3
4:23
Linear
BUTTON
BUTTON
BUTTON
BUTTON
1:1
Activate Windows
Go to Settings to activate Windows.

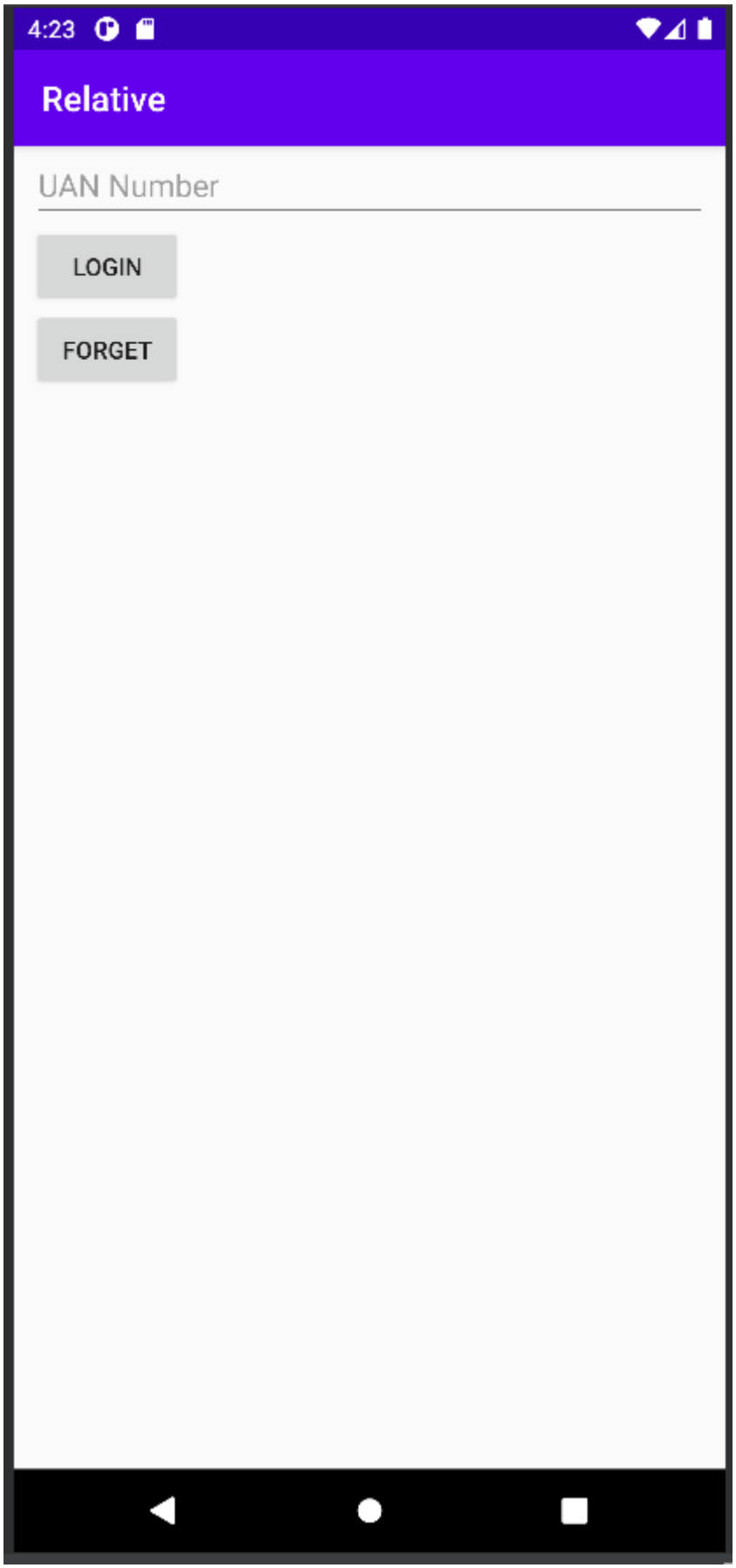
2. Relative Layout

activity_main.xml | MainActivity.kt

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

</RelativeLayout>
```

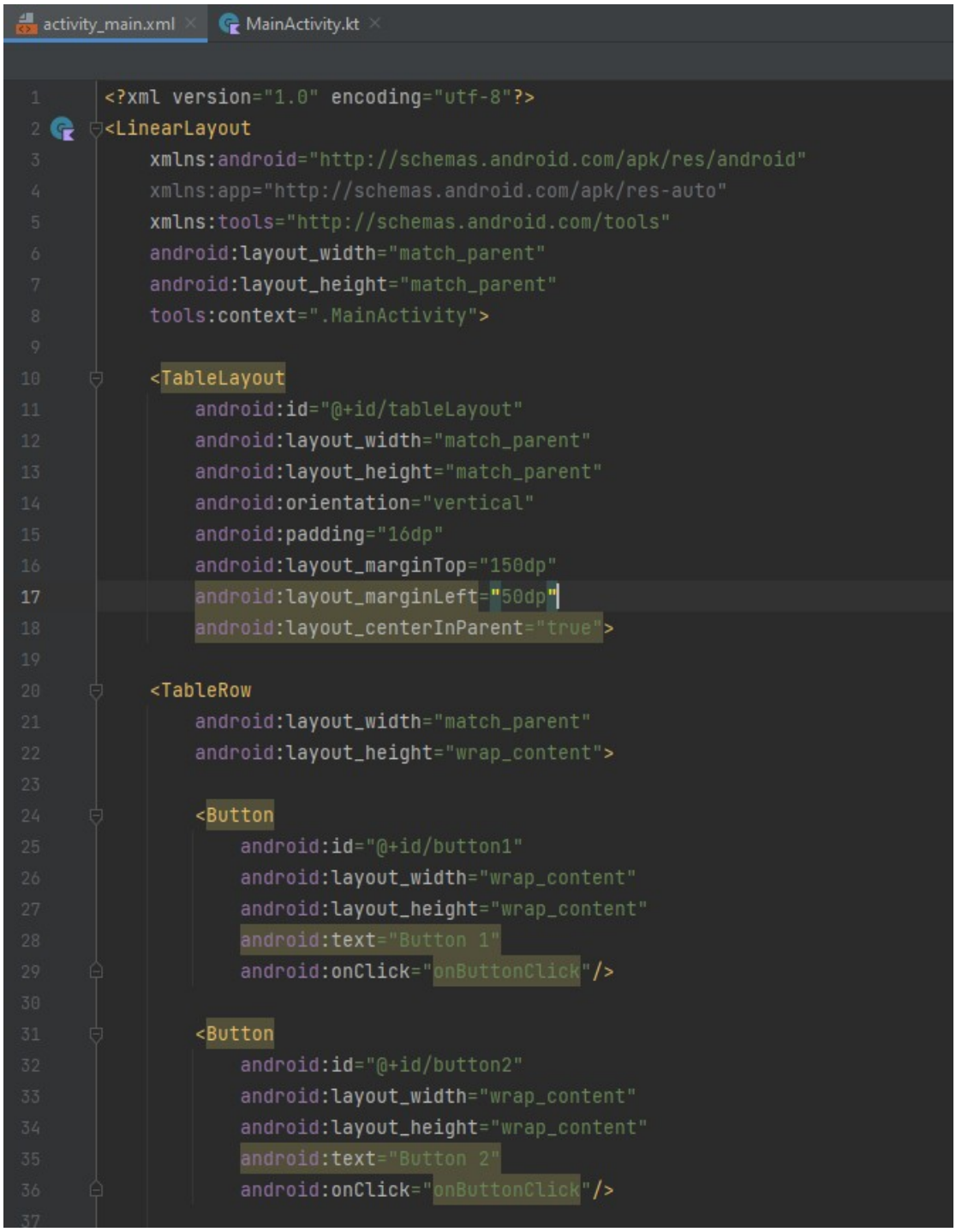
activity_main.xml | MainActivity.kt

```xml
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:paddingLeft="10dp"
    android:paddingRight="10dp"
    tools:context=".MainActivity">
    <EditText
        android:id="@+id/name"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:hint="UAN Number" />
    <LinearLayout
        android:orientation="vertical"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:layout_alignParentStart="true"
        android:layout_alignParentLeft="true"
        android:layout_below="@+id/name">
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Login"
        android:id="@+id/button" />
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Forget"
        android:id="@+id/button2" />
    </LinearLayout>

</RelativeLayout>
```


4:23
Relative
UAN Number
LOGIN
FORGET

3. Table Layout

activity_main.xml



activity_main.xml | MainActivity.kt

```
        <Button
            android:id="@+id/button3"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Button 3"
            android:onClick="onButtonClick"/>
    </TableRow>
        <!-- Row 2 -->
        <TableRow
            android:layout_width="match_parent"
            android:layout_height="wrap_content">

            <Button
                android:id="@+id/button4"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Button 4"
                android:onClick="onButtonClick"/>

            <Button
                android:id="@+id/button5"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Button 5"
                android:onClick="onButtonClick"/>

            <Button
                android:id="@+id/button6"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Button 6"
                android:onClick="onButtonClick"/>
        </TableRow>

```

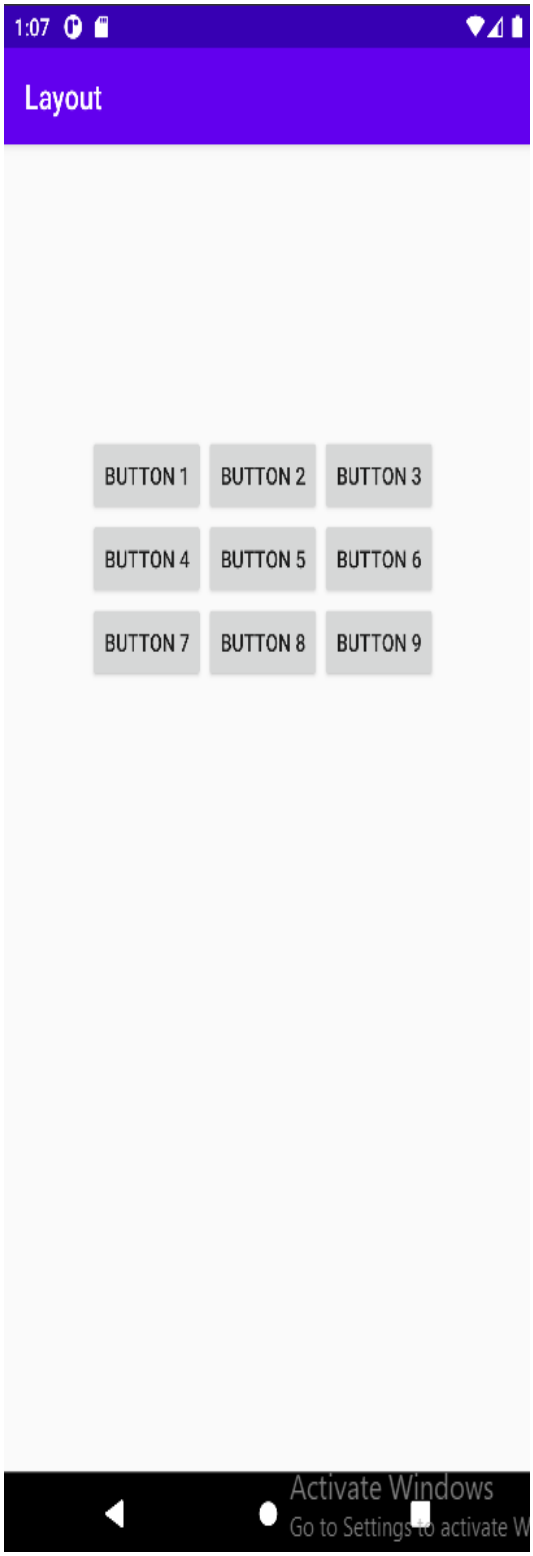
```
71
72          <!-- Row 3 -->
73          <TableRow
74              android:layout_width="match_parent"
75              android:layout_height="wrap_content">
76
77              <Button
78                  android:id="@+id/button7"
79                  android:layout_width="wrap_content"
80                  android:layout_height="wrap_content"
81                  android:text="Button 7"
82                  android:onClick="onButtonClick"/>
83
84              <Button
85                  android:id="@+id/button8"
86                  android:layout_width="wrap_content"
87                  android:layout_height="wrap_content"
88                  android:text="Button 8"
89                  android:onClick="onButtonClick"/>
90
91              <Button
92                  android:id="@+id/button9"
93                  android:layout_width="wrap_content"
94                  android:layout_height="wrap_content"
95                  android:text="Button 9"
96                  android:onClick="onButtonClick"/>
97          </TableRow>
98
99      </TableLayout>
100
101
102 </LinearLayout>
```

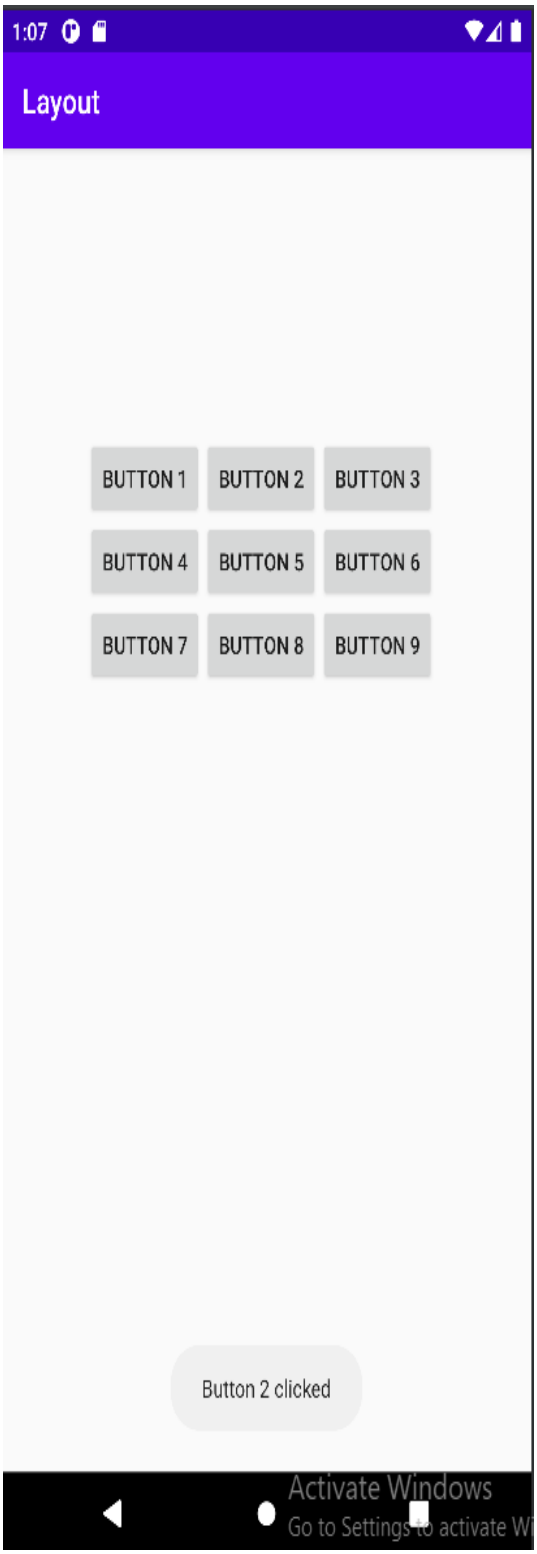
MainActivity.kt

activity_main.xml | MainActivity.kt

```kotlin
package com.example.layout

import android.support.v7.app.AppCompatActivity
import android.os.Bundle
import android.widget.Button
import android.widget.Toast

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        val button1: Button = findViewById(R.id.button1)
        val button2: Button = findViewById(R.id.button2)
        val button3: Button = findViewById(R.id.button3)
        val button4: Button = findViewById(R.id.button4)
        val button5: Button = findViewById(R.id.button5)
        val button6: Button = findViewById(R.id.button6)
        val button7: Button = findViewById(R.id.button7)
        val button8: Button = findViewById(R.id.button8)
        val button9: Button = findViewById(R.id.button9)

        button1.setOnClickListener { onButtonClick( buttonNumber: 1) }
        button2.setOnClickListener { onButtonClick( buttonNumber: 2) }
        button3.setOnClickListener { onButtonClick( buttonNumber: 3) }
        button4.setOnClickListener { onButtonClick( buttonNumber: 4) }
        button5.setOnClickListener { onButtonClick( buttonNumber: 5) }
        button6.setOnClickListener { onButtonClick( buttonNumber: 6) }
        button7.setOnClickListener { onButtonClick( buttonNumber: 7) }
        button8.setOnClickListener { onButtonClick( buttonNumber: 8) }
        button9.setOnClickListener { onButtonClick( buttonNumber: 9) }
    }

    private fun onButtonClick(buttonNumber: Int) {
        Toast.makeText( context: this,  text: "Button $buttonNumber clicked", Toast.LENGTH_SHORT).show()

}
}
```

OUTPUT:-


1:07
Layout
BUTTON 1
BUTTON 2
BUTTON 3
BUTTON 4
BUTTON 5
BUTTON 6
BUTTON 7
BUTTON 8
BUTTON 9
Activate Windows
Go to Settings to activate W


1:07
Layout
BUTTON 1
BUTTON 2
BUTTON 3
BUTTON 4
BUTTON 5
BUTTON 6
BUTTON 7
BUTTON 8
BUTTON 9
Button 2 clicked
Activate Windows
Go to Settings to activate Wi

4. Frame Layout

activity_main.xml | MainActivity.kt

```xml
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <ImageView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:src="@drawable/img"
        android:scaleType="centerCrop"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="50sp"
        android:gravity="center"
        android:text="Frame Layout"
        android:layout_marginTop="300dp"
        android:layout_marginLeft="50dp"
        android:textColor="@color/white"/>

</FrameLayout>
```

activity_main.xml | MainActivity.kt

```kotlin
package com.example.frame

import ...

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
    }
}
```

OUTPUT:-


1:29
Frame
Frame Layout
Activate Windows
Go to Settings to activate Wi

5. Grid View Layout

activity_main.xml | MainActivity.kt

```
<?xml version="1.0" encoding="utf-8"?>
<GridLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:rowCount="3"
    android:columnCount="3"
    android:padding="20dp"
    android:layout_marginTop="150dp"
    android:layout_marginLeft="20dp"
    tools:context=".MainActivity">

    <Button
        android:layout_height="100dp"
        android:layout_width="110dp"
        android:text="1"/>
    <Button
        android:layout_height="100dp"
        android:layout_width="110dp"
        android:text="2"/>
    <Button
        android:layout_height="100dp"
        android:layout_width="110dp"
        android:text="3"/>
    <Button
        android:layout_height="100dp"
        android:layout_width="110dp"
        android:text="4"/>
    <Button
        android:layout_height="100dp"
        android:layout_width="110dp"
        android:text="5"/>
    <Button
        android:layout_height="100dp"
        android:layout_width="110dp"
        android:text="6"/>
    <Button
```

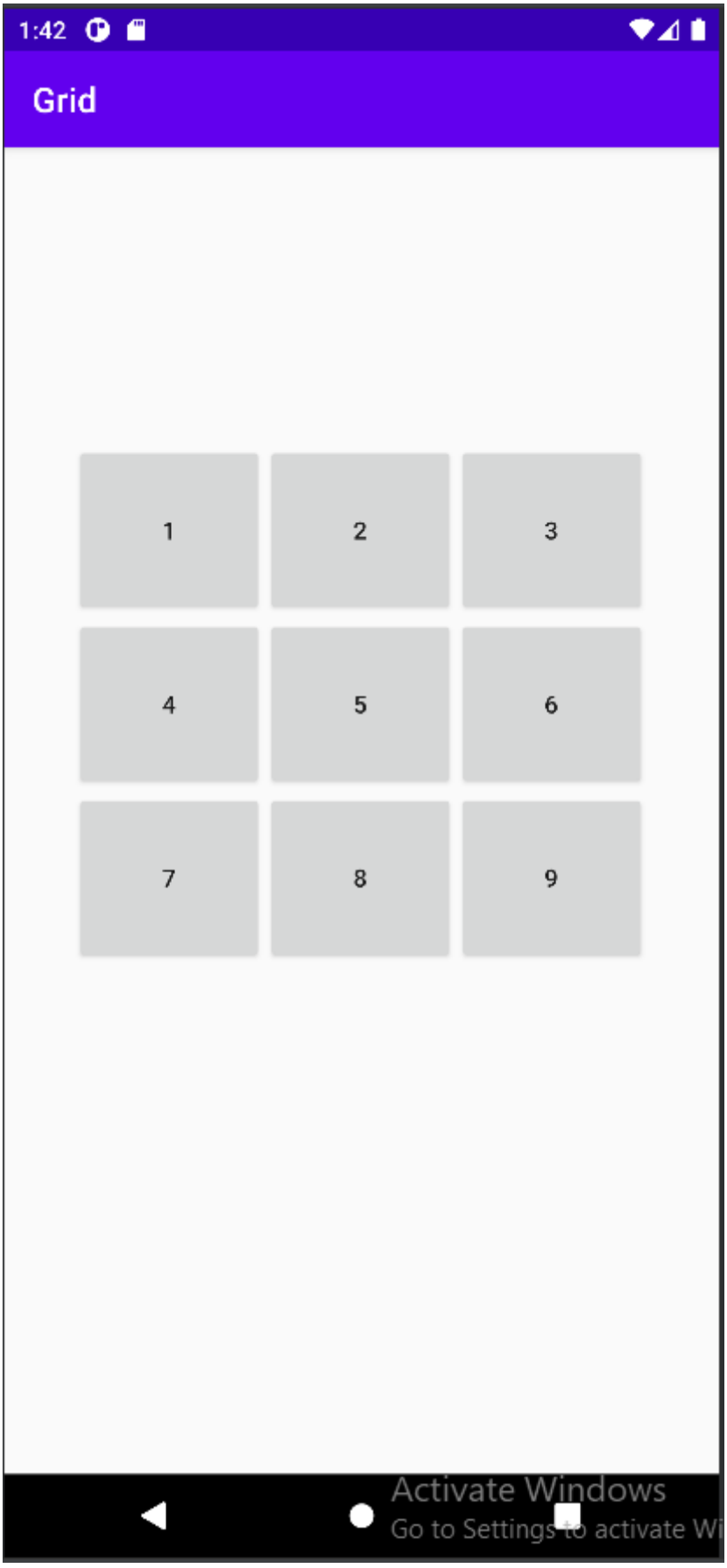
```
    <Button
        android:layout_height="100dp"
        android:layout_width="110dp"
        android:text="7"/>
    <Button
        android:layout_height="100dp"
        android:layout_width="110dp"
        android:text="8"/>
    <Button
        android:layout_height="100dp"
        android:layout_width="110dp"
        android:text="9"/>

</GridLayout>
```

MainActivity.kt

activity_main.xml | MainActivity.kt

```
package com.example.grid

import ...

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
    }
}
```

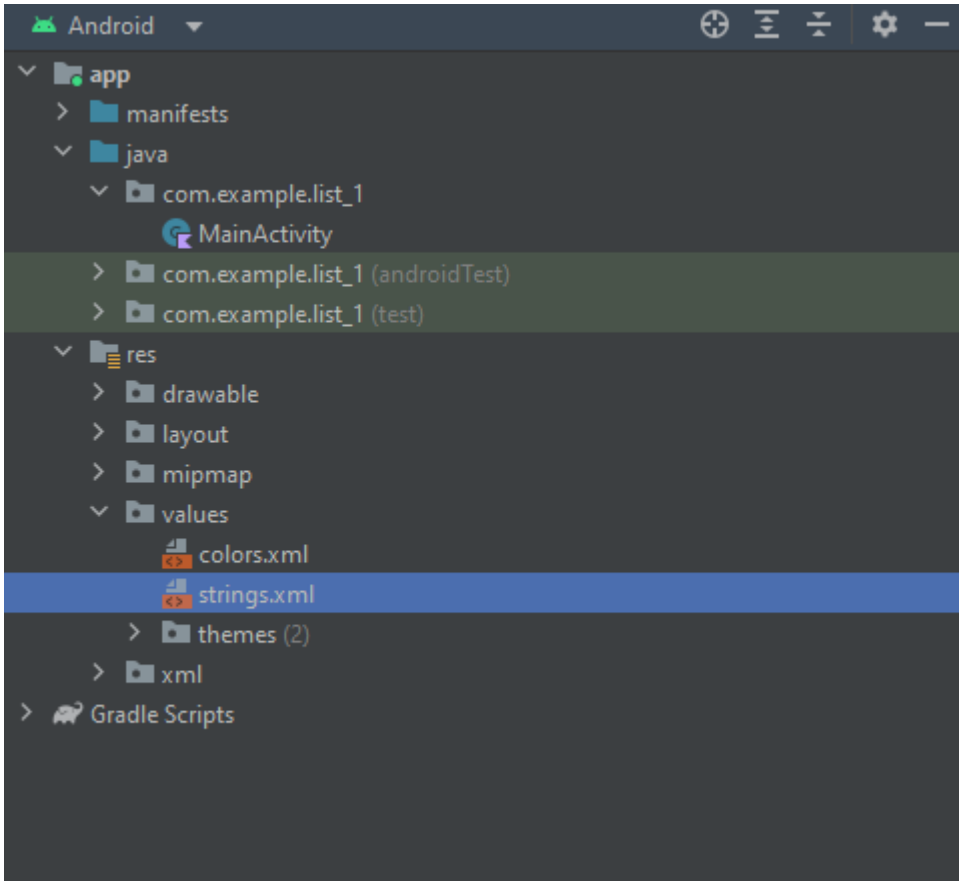
OUTPUT:-

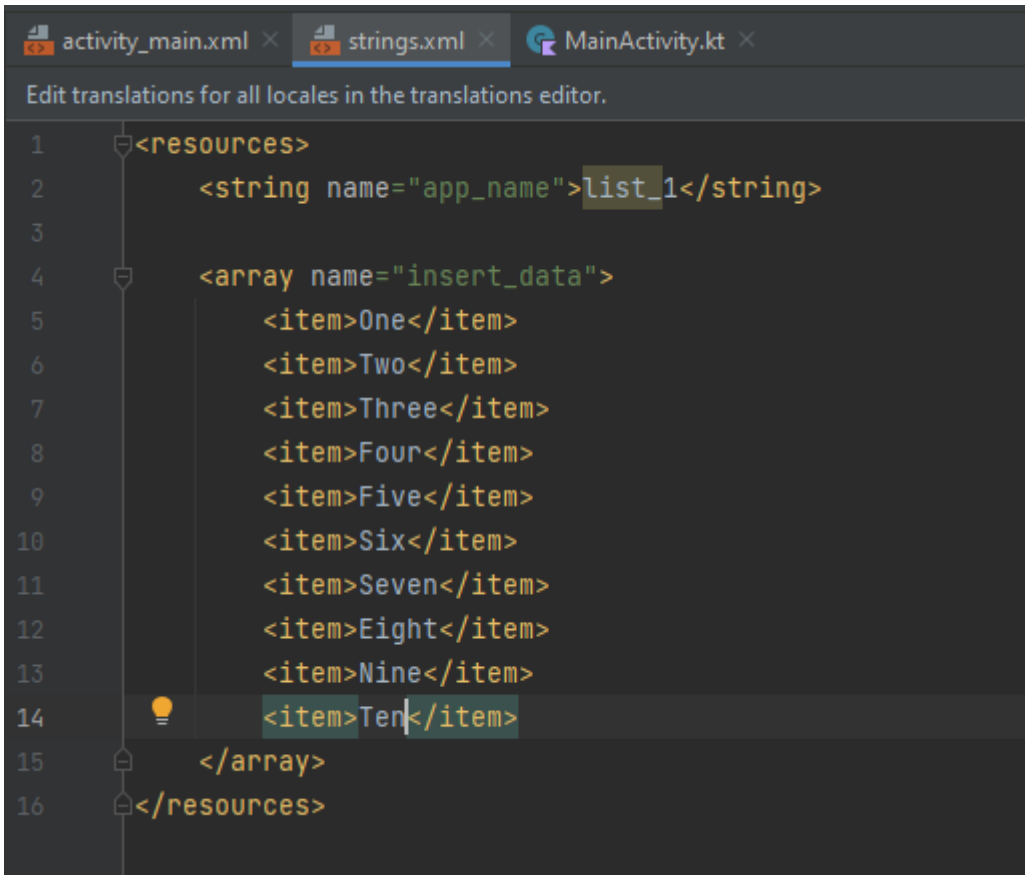

1:42
Grid
1
2
3
4
5
6
7
8
9
Activate Windows
Go to Settings to activate Wi

6. List View Layout

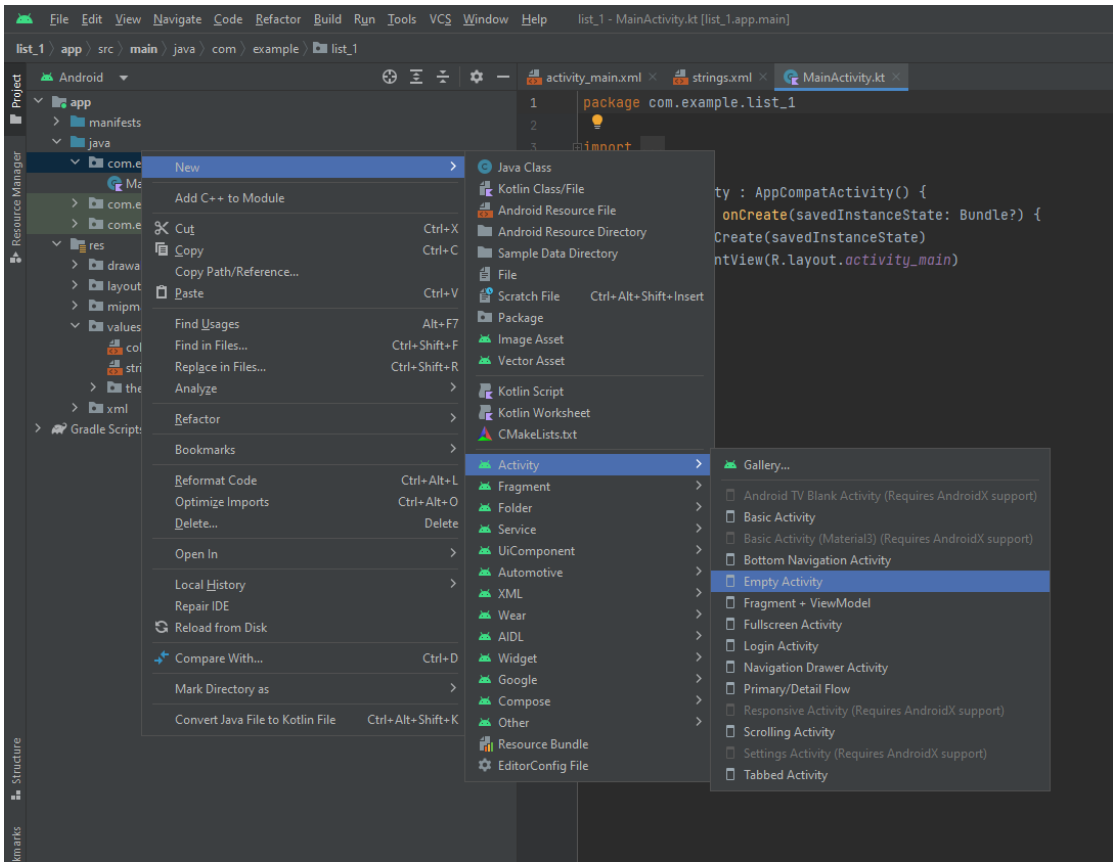
activity_main.xml | strings.xml | MainActivity.kt | activity_list_view.xml | list_view.kt

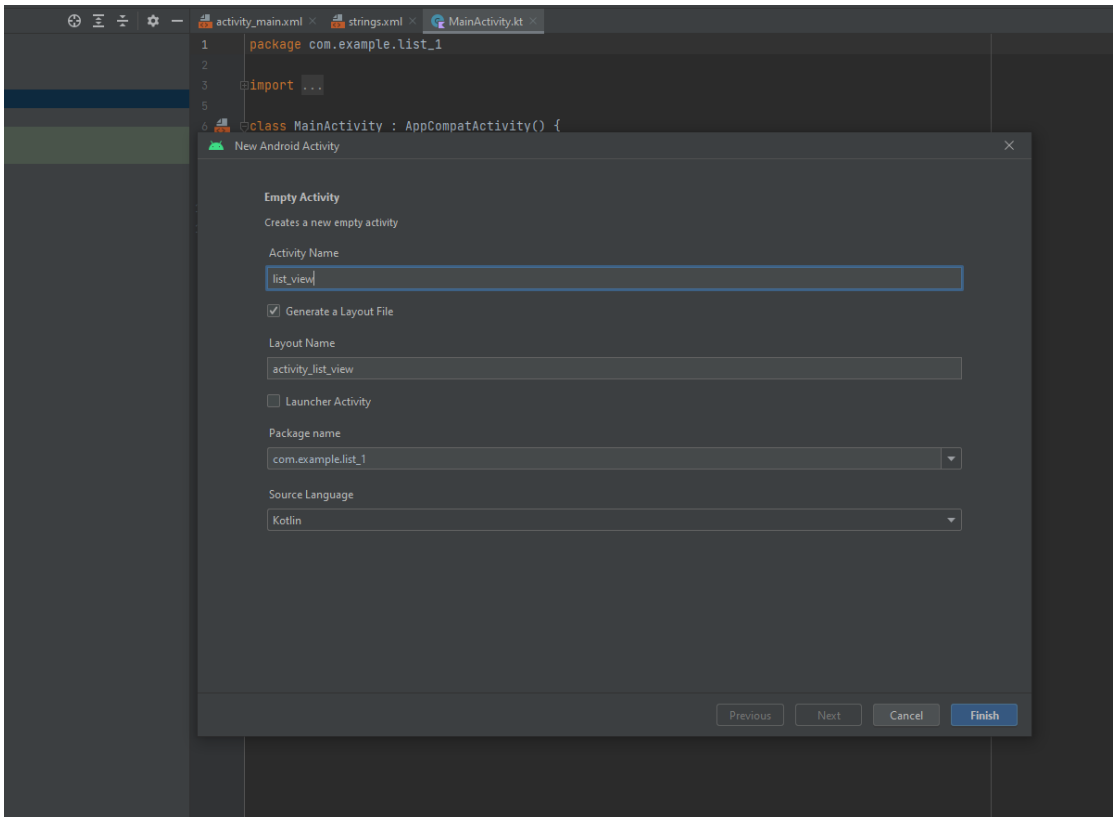
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Click here to view the List"
        android:id="@+id/btn"
        android:layout_marginTop="200dp"
        android:layout_marginLeft="90dp"/>


</LinearLayout>
```

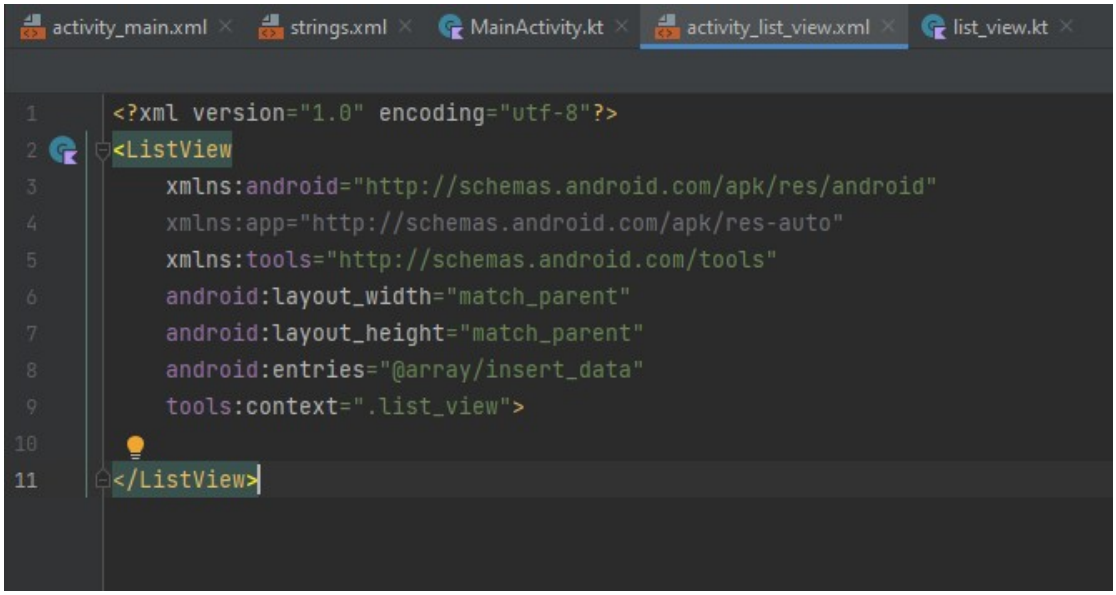




Click on java folder→New→ Activity→Empty Activity




activity_main.xml
strings.xml
MainActivity.kt
package com.example.list_1
import ...
class MainActivity : AppCompatActivity() {
New Android Activity
Empty Activity
Creates a new empty activity
Activity Name
list_view
Generate a Layout File
Layout Name
activity_list_view
Launcher Activity
Package name
com.example.list_1
Source Language
Kotlin
Previous
Next
Cancel
Finish


activity_main.xml
strings.xml
MainActivity.kt
activity_list_view.xml
list_view.kt
<?xml version="1.0" encoding="utf-8"?>
<ListView
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:entries="@array/insert_data"
tools:context=".list_view">
</ListView>

activity_main.xml | strings.xml | MainActivity.kt | activity_list_view.xml | list_view.kt

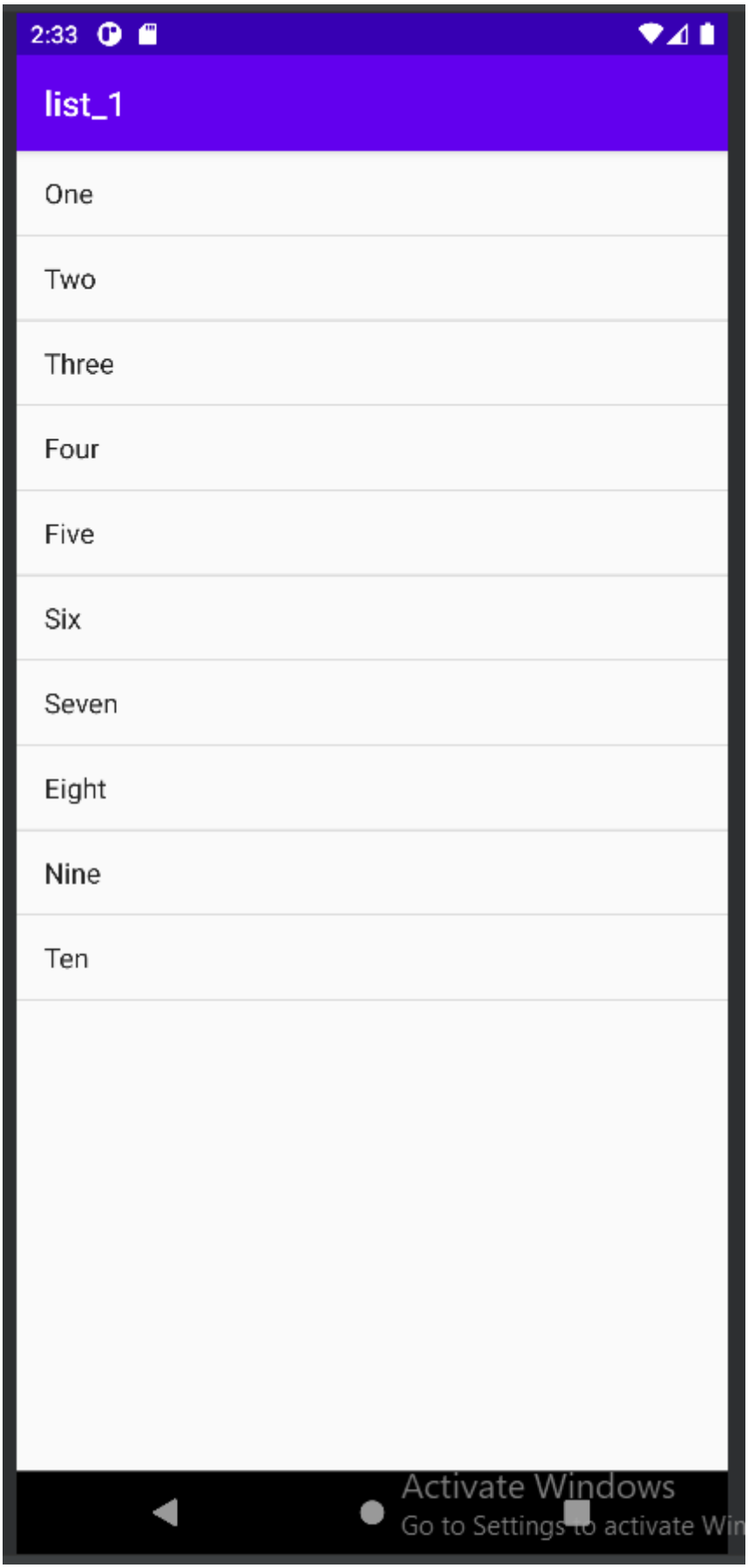
```kotlin
package com.example.list_1

import ...

class list_view : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_list_view)
    }
}
```

activity_main.xml | strings.xml | MainActivity.kt | activity_list_view.xml | list_view.kt

```kotlin
package com.example.list_1

import android.support.v7.app.AppCompatActivity
import android.os.Bundle
import android.content.Intent
import android.widget.Button

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        val btn: Button = findViewById(R.id.btn)

        btn.setOnClickListener { it: View!
            val intent =Intent( packageContext: this, list_view::class.java)
            startActivity(intent)
        }


    }
}
```

OUTPUT:-


2:32
list_1
CLICK HERE TO VIEW THE LIST
Activate Windows
Go to Settings to activate W


2:33
list_1
One
Two
Three
Four
Five
Six
Seven
Eight
Nine
Ten
Activate Windows
Go to Settings to activate Win

7. Absolute Layout

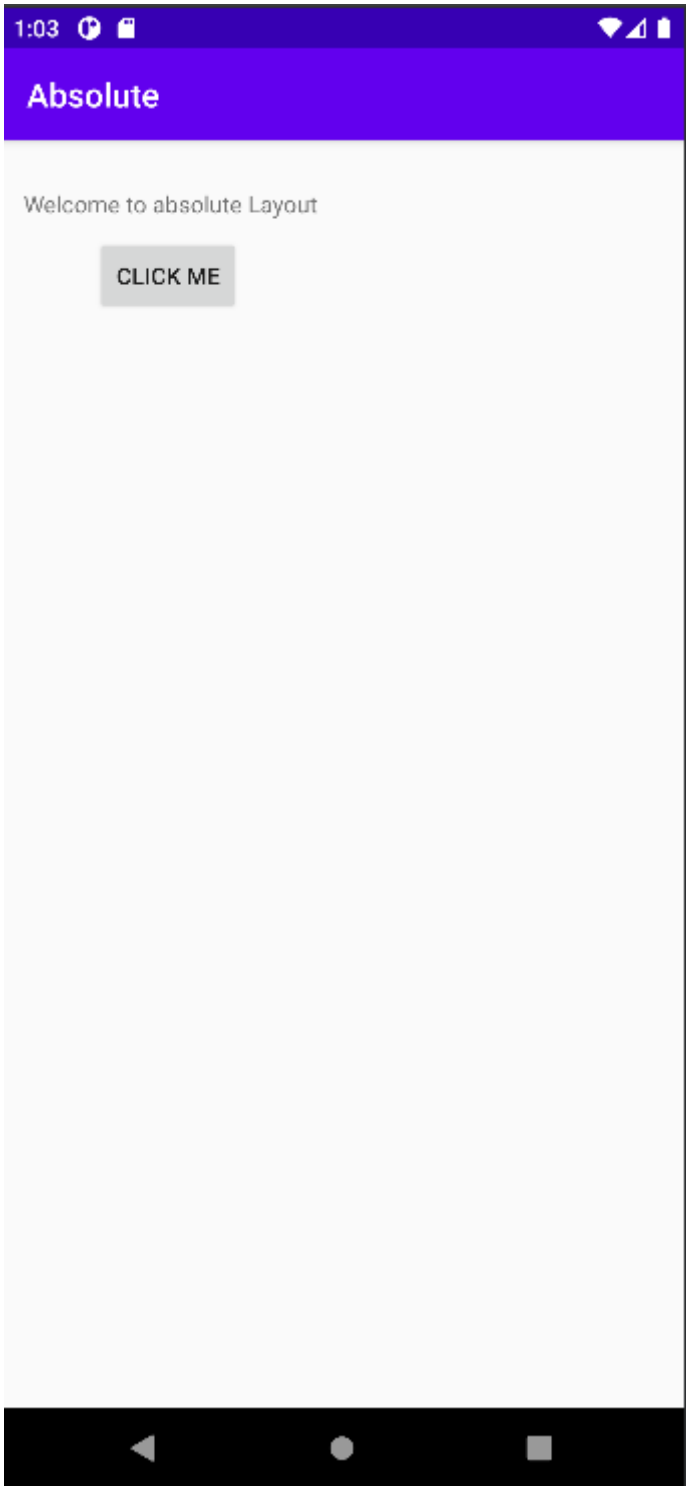
XML Layout file (activity_main.xml):

```
<?xml version="1.0" encoding="utf-8"?>

<AbsoluteLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity">

<TextView
   android:id="@+id/textView1"
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:layout_x="50px"
   android:layout_y="100px"
   android:text="Welcome to absolute Layout"/>
<Button
   android:id="@+id/button1"
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:layout_x="200px"
   android:layout_y="200px"
   android:text="click me"/>

</AbsoluteLayout>
```

Kotlin Activity (MainActivity.kt):

```
package com.example.myapp

import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity() {

   override fun onCreate(savedInstanceState: Bundle?) {
      super.onCreate(savedInstanceState)
      setContentView(R.layout.activity_main)
   }
}
```

OUTPUT:



PRACTICAL-5

Programming UI elements

AppBar, Fragments, UI Components.

activity_main.xml

Copy image and paste in drawable folder.

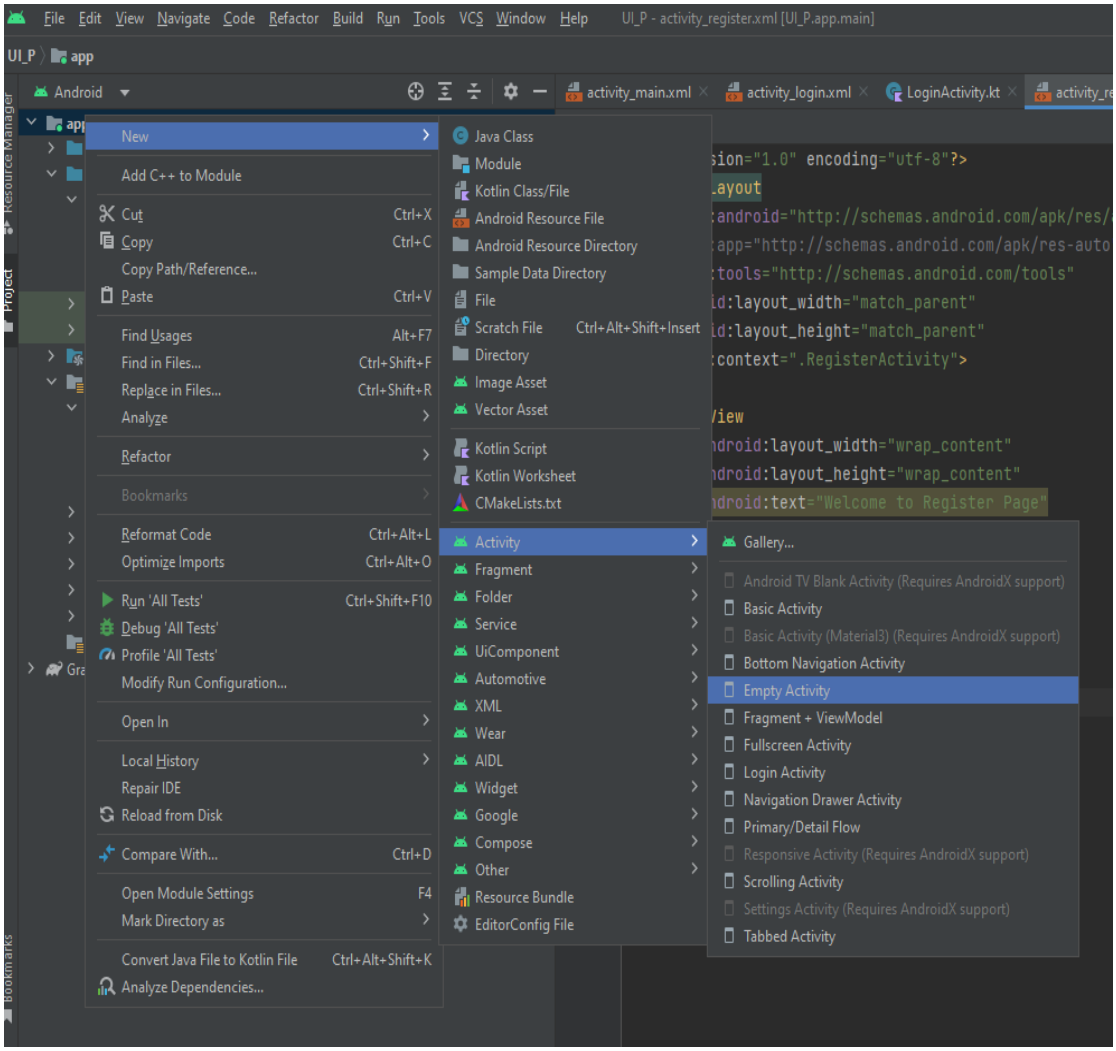
activity_main.xml | activity_login.xml | LoginActivity.kt | activity_register.xml | RegisterActivity.kt | MainActivity.kt

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="@drawable/img"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/name"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:fontFamily="@font/arapey"
        android:text="Welcome To Our App"
        android:textColor="@color/white"
        android:layout_marginTop="160dp"
        android:textSize="30dp" />


    <Button
        android:id="@+id/login"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/name"
        android:layout_margin="20dp"
        android:fontFamily="@font/arapey"
        android:text="Login"
        android:textAllCaps="true"
        android:textSize="20dp"
        android:textAlignment="center"/>
```

```
    <Button
        android:id="@+id/register"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/login"
        android:layout_margin="20dp"
        android:fontFamily="@font/arapey"
        android:text="Register"
        android:textAllCaps="true"
        android:textSize="20dp"
        android:textAlignment="center"/>


</RelativeLayout>
```

Go to java(com.example.ui_pro) → New → Activity → Empty Activity → give name LoginActivity → click finish.

Repeat same process for RegisterActivity.



Come to MainActivity.kt file

activity_main.xml | activity_login.xml | LoginActivity.kt | activity_register.xml | RegisterActivity.kt | MainActivity.kt

```kotlin
package com.example.ui_p

import android.support.v7.app.AppCompatActivity
import android.os.Bundle
import android.widget.Button
import android.content.Intent


class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        val loginbtn: Button = findViewById(R.id.login)
        val registerbtn: Button = findViewById(R.id.register)

        loginbtn.setOnClickListener { it: View!
            val intent= Intent ( packageContext: this, LoginActivity::class.java)
            startActivity(intent)
        }
        registerbtn.setOnClickListener { it: View!
            val intent = Intent( packageContext: this, RegisterActivity::class.java)
            startActivity(intent)
        }
    }
}


```

Now come to activity_login.xml

activity_main.xml | activity_login.xml | LoginActivity.kt | activity_register.xml

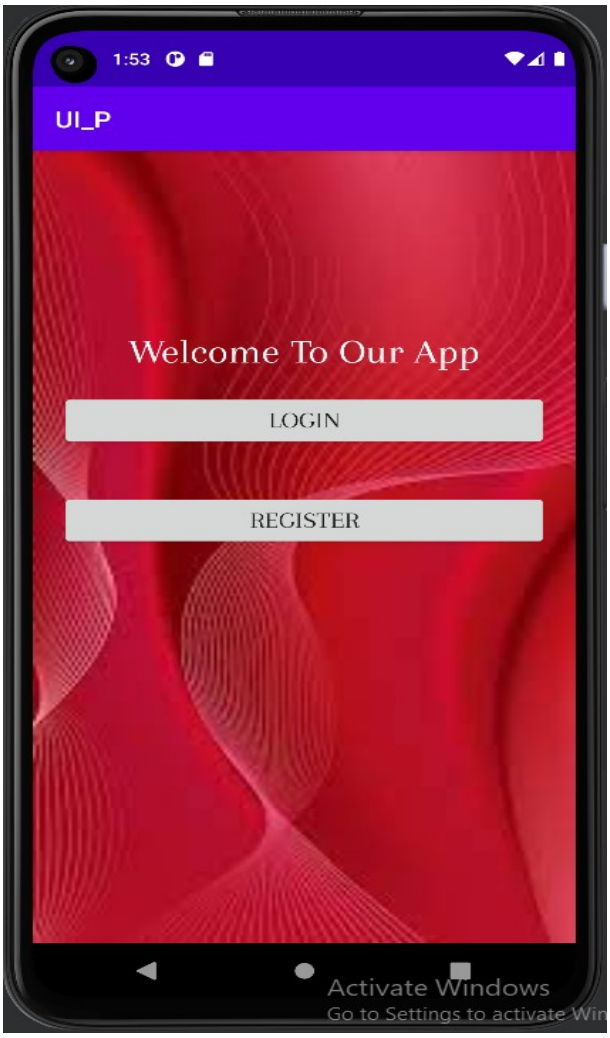
```xml
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".LoginActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Welcome to Login Page"
        android:textSize="30sp"
        android:background="@color/black"
        android:layout_centerInParent="true"
        android:textColor="@color/white" />


</RelativeLayout>
```

Come to activity_register.xml

activity_main.xml | activity_login.xml | LoginActivity.kt | activity_register.xml | RegisterActivity.kt

```xml
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".RegisterActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Welcome to Register Page"
        android:textSize="30sp"
        android:background="@color/black"
        android:layout_centerInParent="true"
        android:textColor="@color/white" />


</RelativeLayout>
```

Now run your application.


1:53
UI_P
Welcome To Our App
LOGIN
REGISTER
Activate Windows
Go to Settings to activate Win


1:53
UI_P
Welcome to Login Page
Activate Windows
Go to Settings to activate Wind

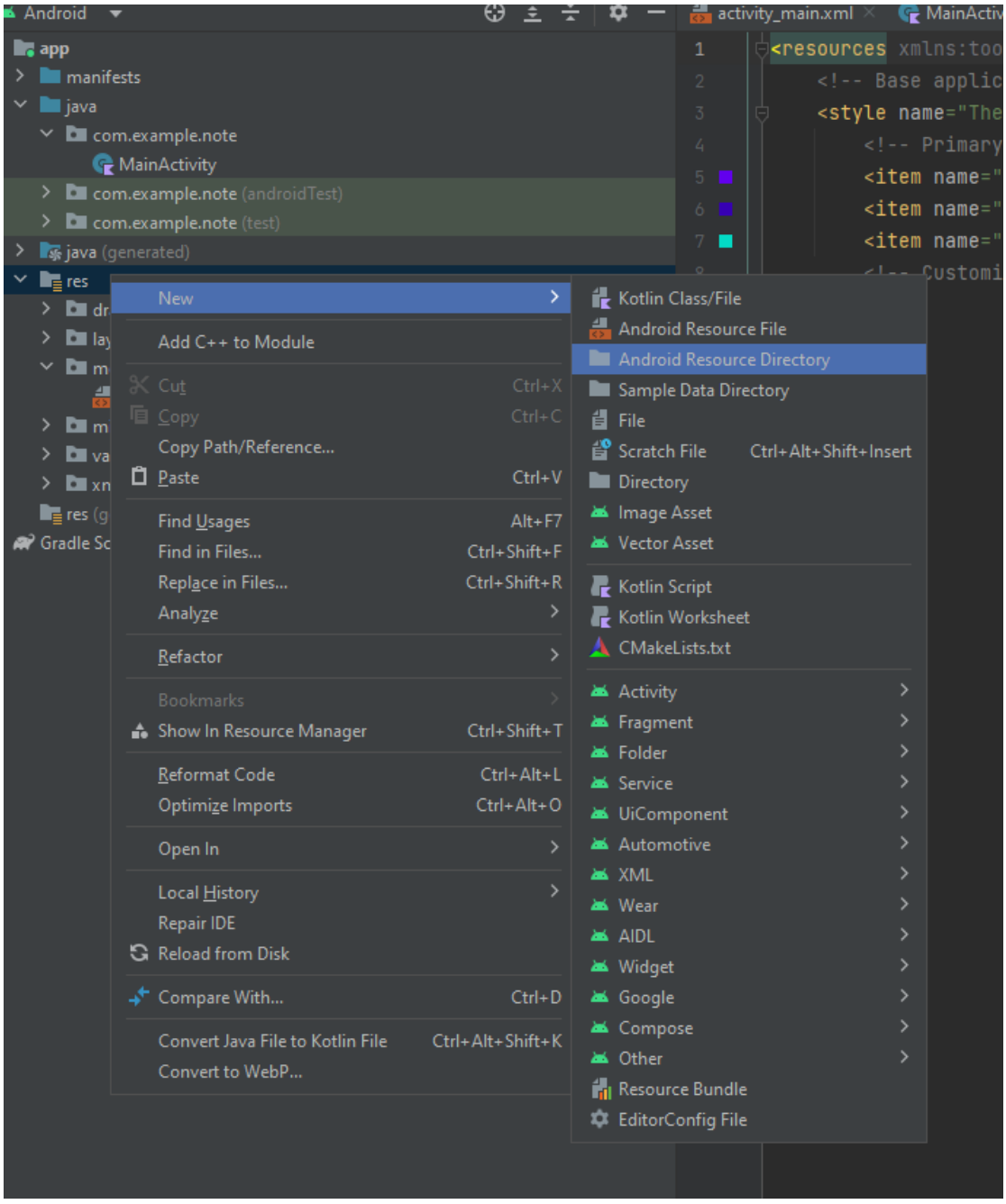

1:54
UI_P
Welcome to Register Page
Activate Windows
Go to Settings to activate Wi

PRACTICAL-6

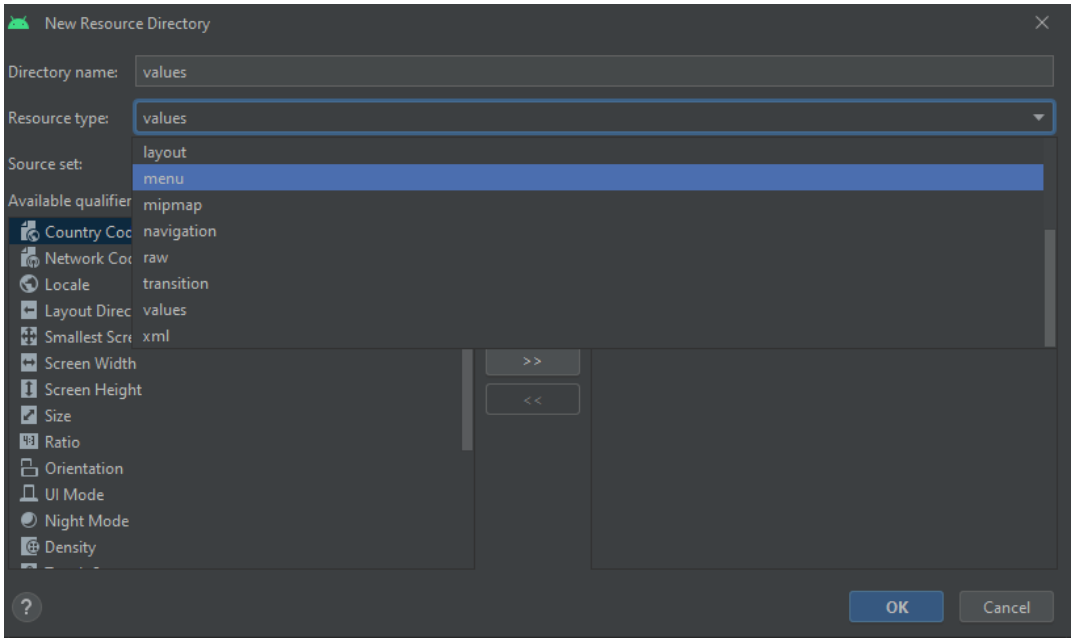
Programming Menus, Dialog, Dialog fragments

1. Menu

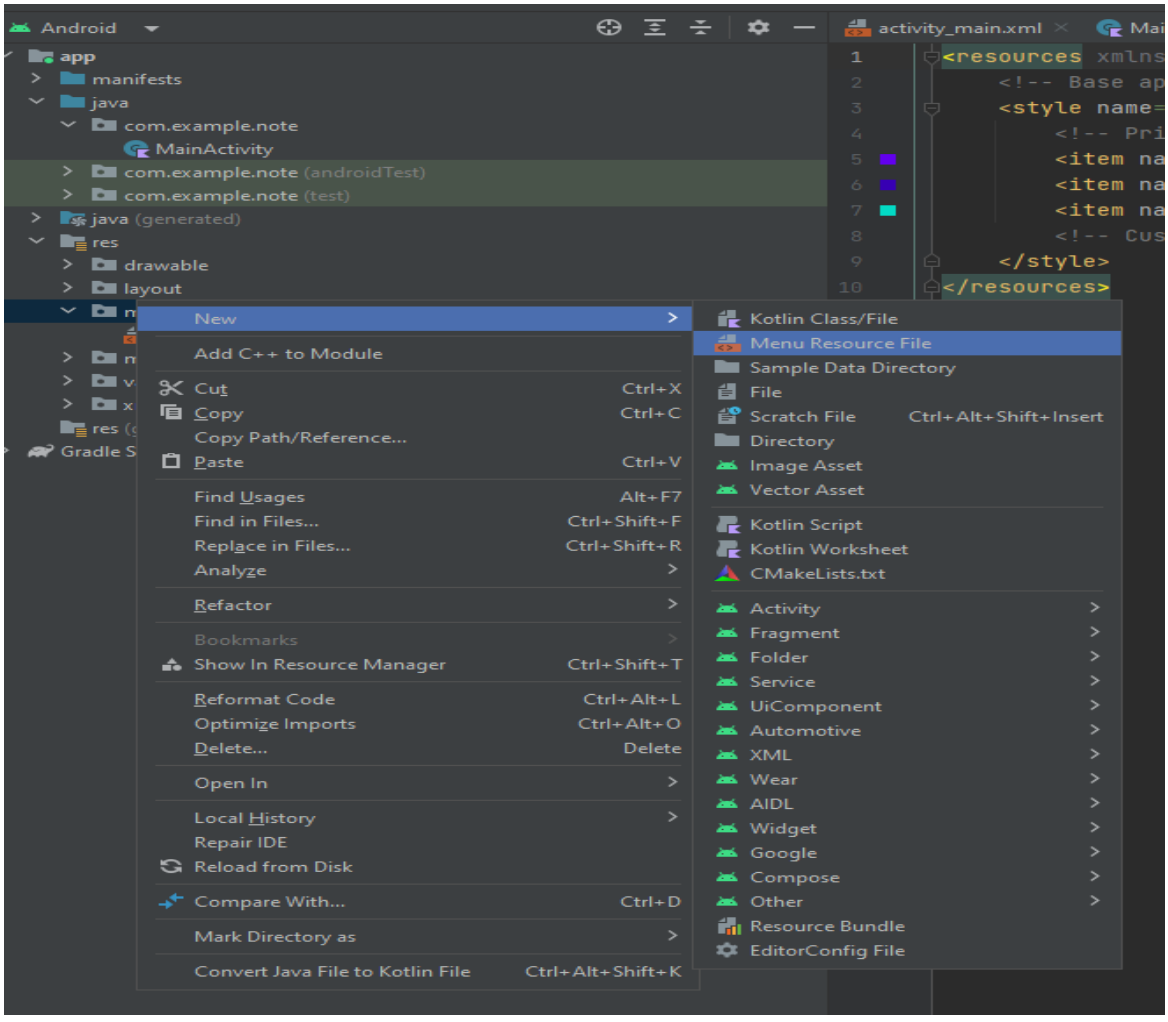
Expand res folder --> Right click on res folder --> New --> then select Android Resource Directory

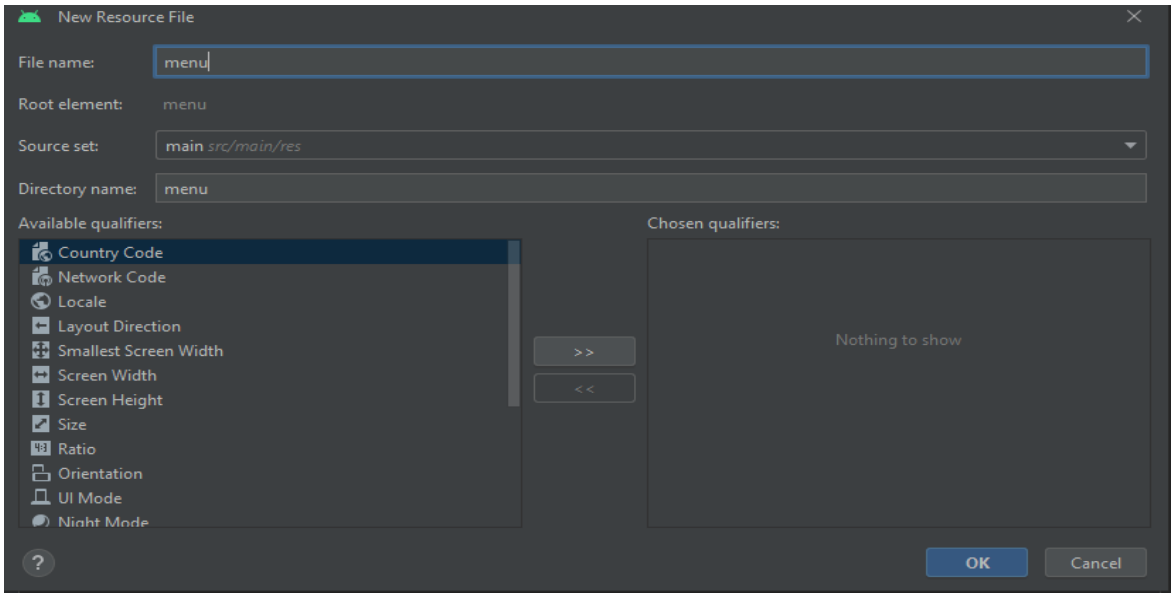


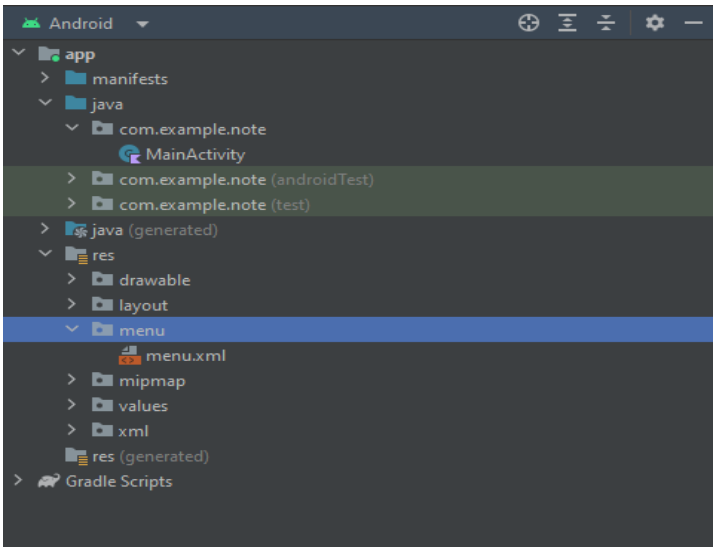
Then Select menu → click ok → menu folder is created in res folder.



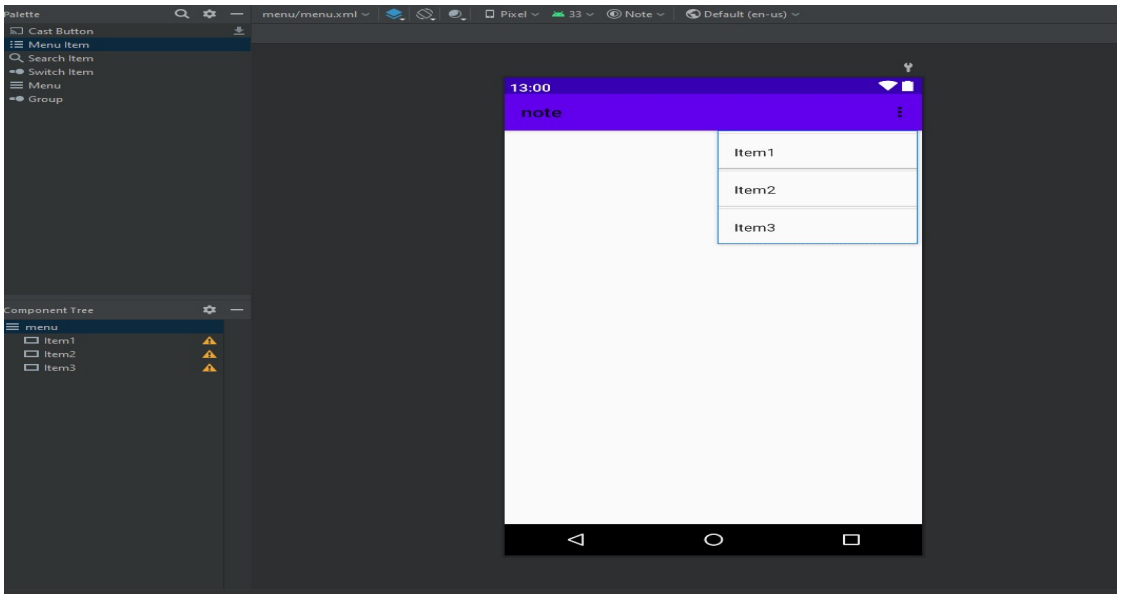
Now right click on menu folder→ select New→ select Menu Resource File.







Then drag & drop 3 Menu Item in menu.xml.



activity_main.xml | MainActivity.kt | values\themes.xml | night\themes.xml | menu.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:android="http://schemas.android.com/apk/res/android">

    <item android:title="Item1"
        android:id="@+id/Item1"/>
    <item android:title="Item2"
        android:id="@+id/Item2"/>
    <item android:title="Item3"
        android:id="@+id/Item3"/>
</menu>
```

Go to MainActivity.kt file and add below code.

activity_main.xml | MainActivity.kt | values\themes.xml | night\themes.xml | menu.xml

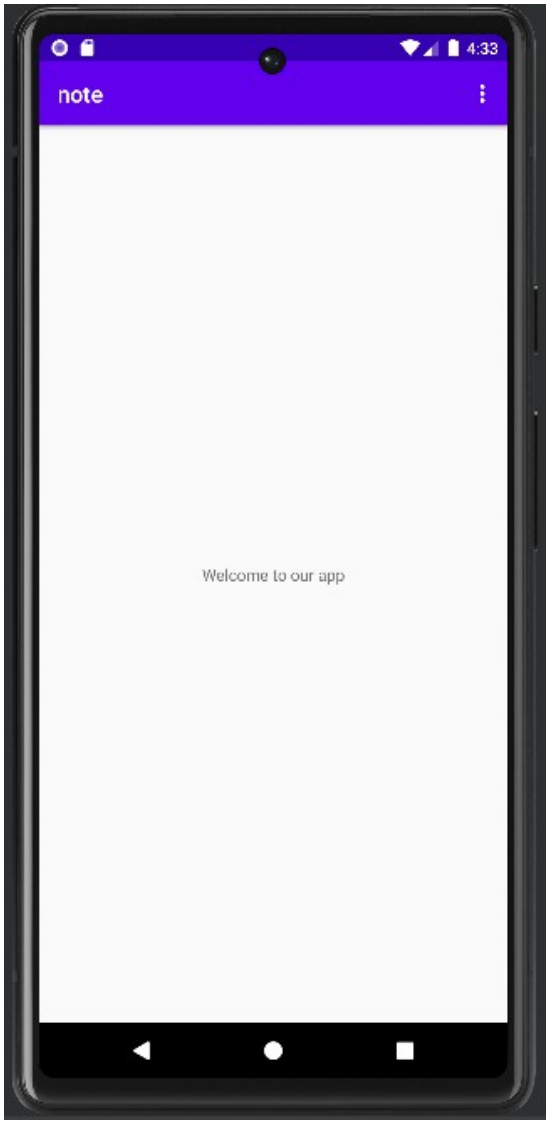
```kotlin
package com.example.note

import android.support.v7.app.AppCompatActivity
import android.os.Bundle
import android.view.Menu
import android.view.MenuItem
import android.widget.Toast

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

    }
    override fun onCreateOptionsMenu(menu: Menu?): Boolean {
        menuInflater.inflate(R.menu.menu, menu)
        return true
    }

    override fun onOptionsItemSelected(item: MenuItem): Boolean {
        val id = item.itemId
        when (id) {
            R.id.Item1 -> {
                Toast.makeText( context: this,  text: "Item 1 Selected", Toast.LENGTH_SHORT).show()
            }
            R.id.Item2 -> {
                Toast.makeText( context: this,  text: "Item 2 Selected", Toast.LENGTH_SHORT).show()
            }
            R.id.Item3 -> {
                Toast.makeText( context: this,  text: "Item 3 Selected", Toast.LENGTH_SHORT).show()
            }
            else -> return super.onOptionsItemSelected(item)
        }
        return true
    }
}
```

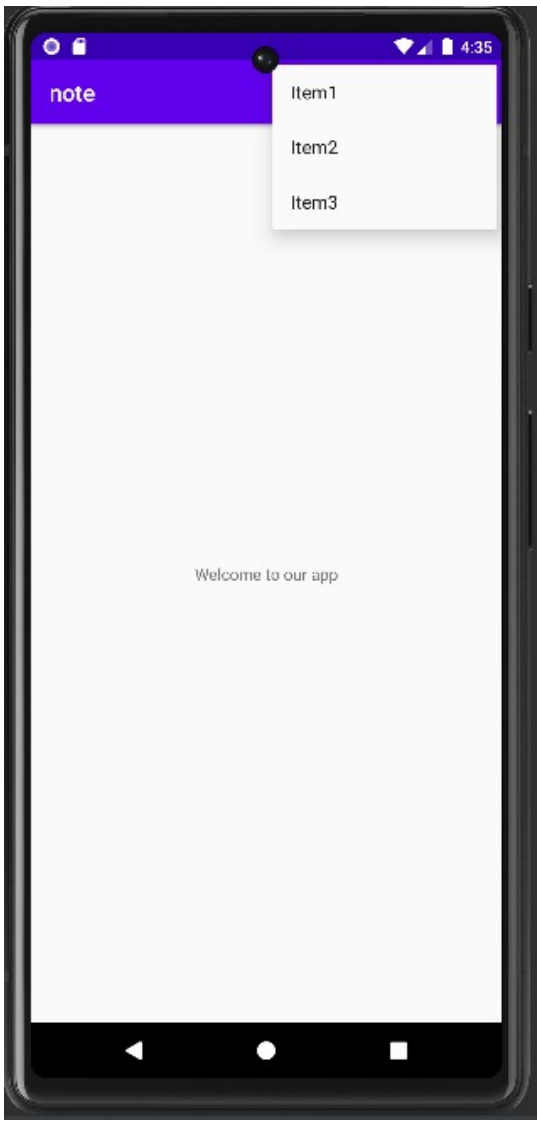
Now go to values folder→ then expand themes folder → check both the theme files.

activity_main.xml | MainActivity.kt | values\themes.xml | night\themes.xml | menu.xml

```
<resources xmlns:tools="http://schemas.android.com/tools">
    <!-- Base application theme. -->
    <style name="Theme.Note" parent="Theme.AppCompat.Light.DarkActionBar">
        <!-- Primary brand color. -->
        <item name="colorPrimary">@color/purple_500</item>
        <item name="colorPrimaryDark">@color/purple_700</item>
        <item name="colorAccent">@color/teal_200</item>
        <!-- Customize your theme here. -->
    </style>
</resources>
```

OUTPUT:






4:37
note
Welcome to our app
Item 3 Selected

2. Dialog:

Change Layout to Linear Layout.

Remove TextView and drag & drop plain text and button.

activity_main.xml | MainActivity.kt

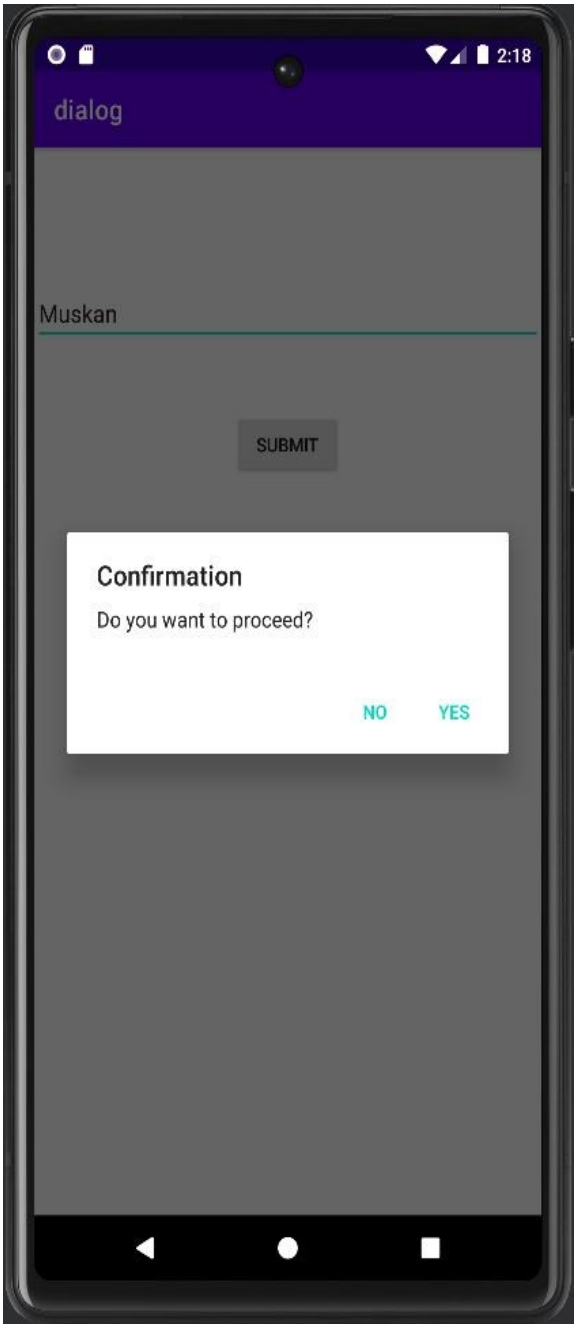
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">


    <EditText
        android:id="@+id/editTextTextPersonName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="textPersonName"
        android:text=""
        android:hint="Enter your name"
        android:layout_marginTop="100dp"/>

    <Button
        android:id="@+id/alert"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Submit"
        android:layout_gravity="center"
        android:layout_marginTop="50dp"/>
</LinearLayout>
```

activity_main.xml | MainActivity.kt

```kotlin
package com.example.dialog

import android.content.DialogInterface
import android.support.v7.app.AppCompatActivity
import android.os.Bundle
import android.widget.Button
import android.support.v7.app.AlertDialog
import android.widget.Toast

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        val submit= findViewById<Button>(R.id.alert)
        submit.setOnClickListener { it: View!
            val simplealert=AlertDialog.Builder( context: this).create()
            simplealert.setTitle("Confirmation")
            simplealert.setMessage("Do you want to proceed?")
            simplealert.setButton(AlertDialog.BUTTON_POSITIVE, text: "Yes")
            {
                _:DialogInterface?, which:Int ->
                Toast.makeText( context: this, text: "You said yes",Toast.LENGTH_LONG).show()
            }
            simplealert.setButton(AlertDialog.BUTTON_NEGATIVE, text: "No")
            {
                _:DialogInterface?, which:Int ->
                Toast.makeText( context: this, text: "You said no",Toast.LENGTH_LONG).show()
            }
            simplealert.show()
        }

    }
}
```

OUTPUT:


2:20
dialog
Enter your name
SUBMIT


2:18
dialog
Muskan
SUBMIT
Confirmation
Do you want to proceed?
NO
YES


2:18
dialog
Muskan
SUBMIT
You said yes


2:19
dialog
Muskan
SUBMIT
You said no

3. Dialog fragments:

Expand res folder→ select layout Folder→right click on layout folder→ select new→select Android Resource File → give name to the file "dialog_fragments".

activity_main.xml | MainActivity.kt | dialog_fragments.xml | Myfragment.kt

```xml
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout

    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <RelativeLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="50dp">

        <RadioGroup
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical"
            android:id="@+id/myradiogroup"
            android:padding="10dp">

            <RadioButton
                android:text="Excellent"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:checked="false"
                android:id="@+id/radioButton1"/>

            <RadioButton
                android:text="Very Good"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:checked="false"
                android:id="@+id/radioButton2"/>
```

activity_main.xml | MainActivity.kt | dialog_fragments.xml | Myfragment.kt

```
            <RadioButton
                android:text="Good"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:checked="false"
                android:id="@+id/radioButton3"/>

            <RadioButton
                android:text="Average"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:checked="false"
                android:id="@+id/radioButton4"/>

            <RadioButton
                android:text="Bad"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:checked="false"
                android:id="@+id/radioButton5"/>

            <Button
                android:text="Submit"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:id="@+id/submitButton"/>

            <Button
                android:text="Cancel"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:id="@+id/cancelButton"/>

        </RadioGroup>
    </RelativeLayout>
</RelativeLayout>
```

Now go to java folder & expand it → right click on com.example.frag→select new→ select Kotlin file class →and give name as "Myfragment" and select kind as "class" → click Ok.

activity_main.xml | MainActivity.kt | dialog_fragments.xml | Myfragment.kt

```kotlin
package com.example.frag

import android.os.Bundle
import android.support.v4.app.DialogFragment
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.Button
import android.widget.RadioButton
import android.widget.RadioGroup
import android.util.Log

class Myfragment: DialogFragment() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
    }

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        val rootView: View = inflater.inflate(R.layout.dialog_fragments, container,  attachToRoot: false)

        val cancelButton = rootView.findViewById<Button>(R.id.cancelButton)
        val submitButton = rootView.findViewById<Button>(R.id.submitButton)
        val surveyRadioGroup = rootView.findViewById<RadioGroup>(R.id.myradiogroup)

        cancelButton.setOnClickListener { it: View!
            dismiss()
        }
```

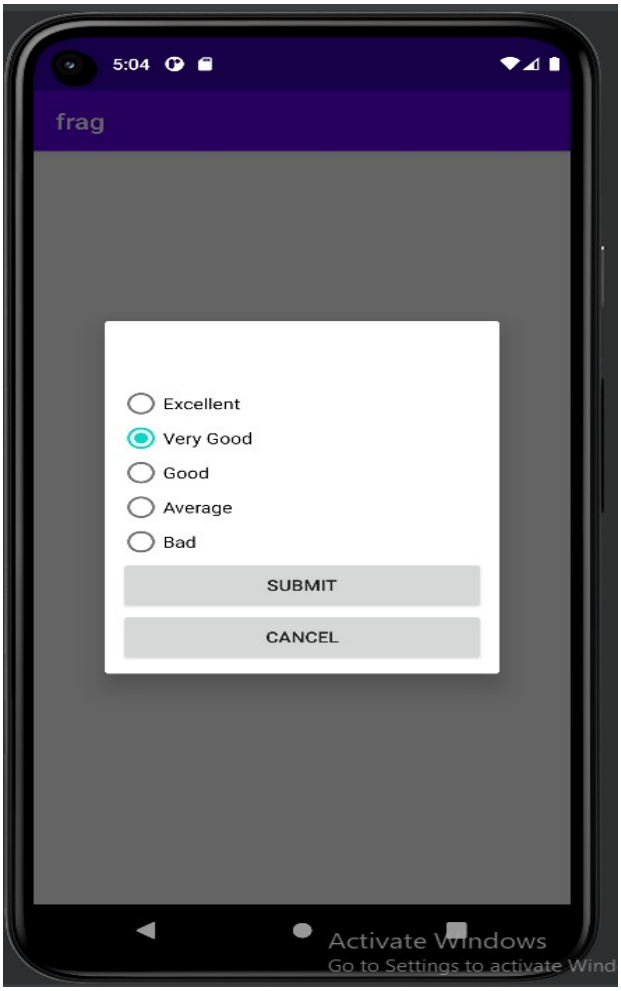
```kotlin

        submitButton.setOnClickListener { it: View!
            val selectedId = surveyRadioGroup.checkedRadioButtonId
            if (selectedId != -1) {
                val selectedRadioButton = rootView.findViewById<RadioButton>(selectedId)
                Log.d( tag: "test", selectedRadioButton.text.toString())
            } else {
                Log.d( tag: "test",  msg: "No option selected")
            }
            dismiss()
        }

        return rootView
    }
}
```

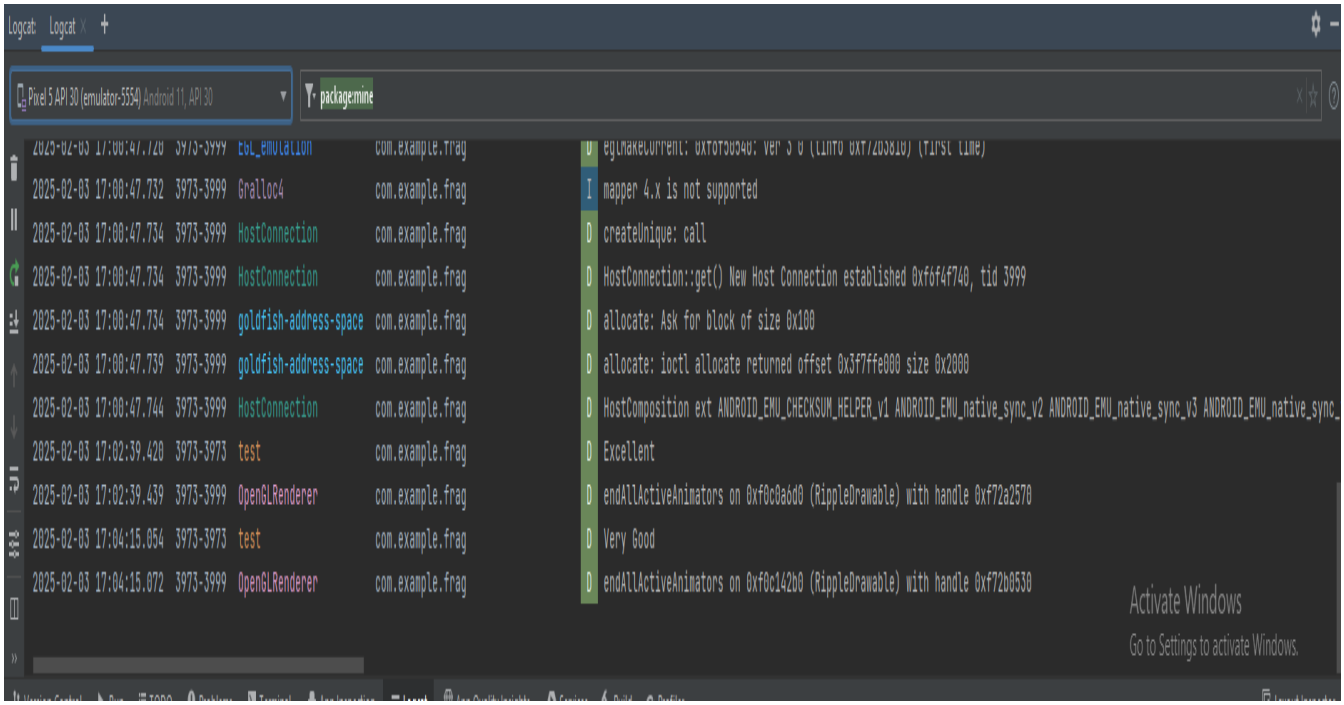
activity_main.xml | MainActivity.kt | dialog_fragments.xml | Myfragment.kt

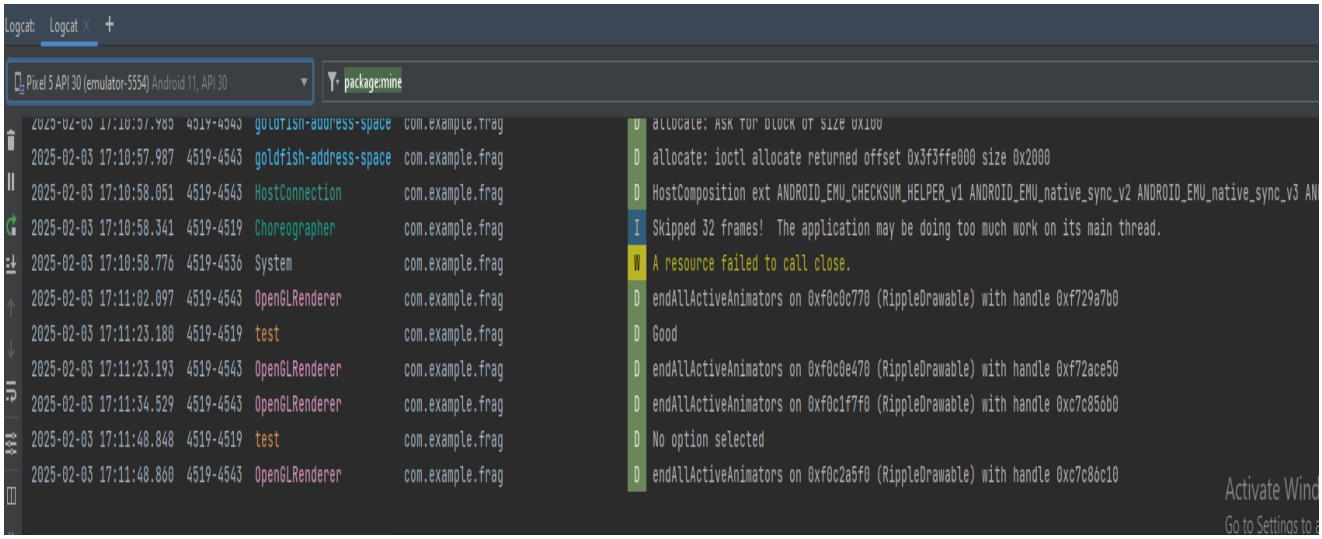
```kotlin
package com.example.frag

import android.support.v7.app.AppCompatActivity
import android.os.Bundle
import android.support.v4.app.FragmentManager

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        val fm: FragmentManager = supportFragmentManager
        val myFragment = Myfragment()
        myFragment.show(fm, tag: "simple fragment")
    }

}
```

OUTPUT:


5:02
frag
Excellent
Very Good
Good
Average
Bad
SUBMIT
CANCEL
Activate Windows
Go to Settings to activate Wind


5:04
frag
Excellent
Very Good
Good
Average
Bad
SUBMIT
CANCEL
Activate Windows
Go to Settings to activate Wind


Logcat
Logcat
Pixel 5 API 30 (emulator-5554) Android 11, API 30
package:mine
2025-02-03 17:00:47.732 3973-3999 Gralloc4 com.example.frag I mapper 4.x is not supported
2025-02-03 17:00:47.734 3973-3999 HostConnection com.example.frag D createUnique: call
2025-02-03 17:00:47.734 3973-3999 HostConnection com.example.frag D HostConnection::get() New Host Connection established 0xf6f4f740, tid 3999
2025-02-03 17:00:47.734 3973-3999 goldfish-address-space com.example.frag D allocate: Ask for block of size 0x100
2025-02-03 17:00:47.739 3973-3999 goldfish-address-space com.example.frag D allocate: ioctl allocate returned offset 0x3f7ffe000 size 0x2000
2025-02-03 17:00:47.744 3973-3999 HostConnection com.example.frag D HostComposition ext ANDROID_EMU_CHECKSUM_HELPER_v1 ANDROID_EMU_native_sync_v2 ANDROID_EMU_native_sync_v3 ANDROID_EMU_native_sync_
2025-02-03 17:02:39.420 3973-3973 test com.example.frag D Excellent
2025-02-03 17:02:39.439 3973-3999 OpenGLRenderer com.example.frag D endAllActiveAnimators on 0xf0c0a6d0 (RippleDrawable) with handle 0xf72a2570
2025-02-03 17:04:15.054 3973-3973 test com.example.frag D Very Good
2025-02-03 17:04:15.072 3973-3999 OpenGLRenderer com.example.frag D endAllActiveAnimators on 0xf0c142b0 (RippleDrawable) with handle 0xf72b0530
Activate Windows
Go to Settings to activate Windows.

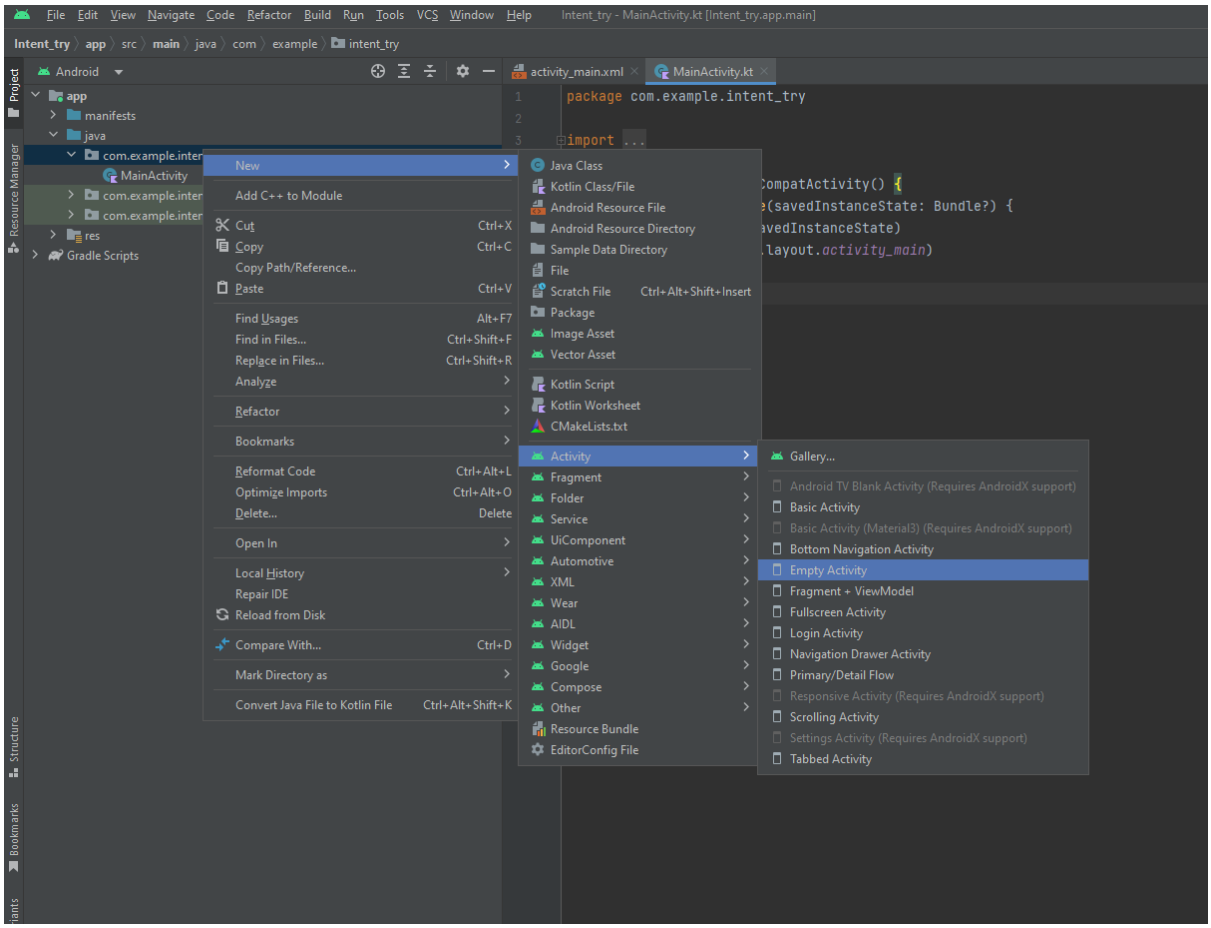

Logcat
Logcat
Pixel 5 API 30 (emulator-5554) Android 11, API 30
package:mine
2025-02-03 17:10:57.987 4519-4543 goldfish-address-space com.example.frag D allocate: ioctl allocate returned offset 0x3f3ffe000 size 0x2000
2025-02-03 17:10:58.051 4519-4543 HostConnection com.example.frag D HostComposition ext ANDROID_EMU_CHECKSUM_HELPER_v1 ANDROID_EMU_native_sync_v2 ANDROID_EMU_native_sync_v3
2025-02-03 17:10:58.341 4519-4519 Choreographer com.example.frag I Skipped 32 frames! The application may be doing too much work on its main thread.
2025-02-03 17:10:58.776 4519-4536 System com.example.frag W A resource failed to call close.
2025-02-03 17:11:02.097 4519-4543 OpenGLRenderer com.example.frag D endAllActiveAnimators on 0xf0c0c770 (RippleDrawable) with handle 0xf729a7b0
2025-02-03 17:11:23.180 4519-4519 test com.example.frag D Good
2025-02-03 17:11:23.193 4519-4543 OpenGLRenderer com.example.frag D endAllActiveAnimators on 0xf0c0e470 (RippleDrawable) with handle 0xf72ace50
2025-02-03 17:11:34.529 4519-4543 OpenGLRenderer com.example.frag D endAllActiveAnimators on 0xf0c1f7f0 (RippleDrawable) with handle 0xc7c856b0
2025-02-03 17:11:48.848 4519-4519 test com.example.frag D No option selected
2025-02-03 17:11:48.860 4519-4543 OpenGLRenderer com.example.frag D endAllActiveAnimators on 0xf0c2a5f0 (RippleDrawable) with handle 0xc7c86c10

PRACTICAL-7

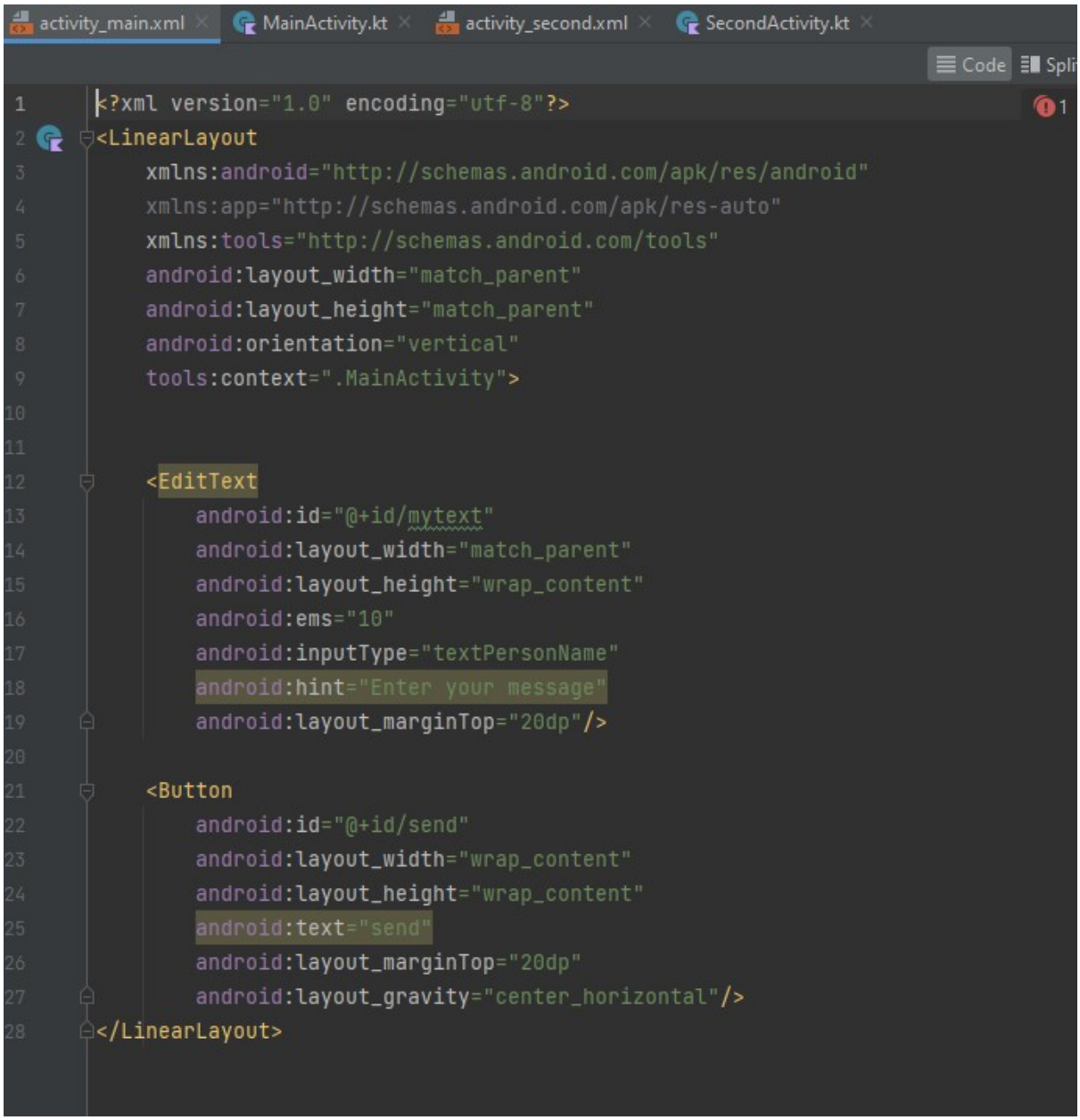
Programs on Intents, Events, Listeners and Adapters

The Android Intent Class, Using Events and Event Listeners

Step 1: Create another activity (i.e. SecondActivity)



Step 2: Come to activity_main.xml file and add below code.



Step 3: Now come to activity_second.xml file and add below code.

activity_main.xml | MainActivity.kt | activity_second.xml | SecondActivity.kt

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".SecondActivity">

    <TextView
        android:id="@+id/display"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="TextView"
        android:layout_marginTop="40dp"/>
</LinearLayout>
```

Step 4: Now come to MainActivity.kt file and add below code.

activity_main.xml | MainActivity.kt | activity_second.xml | SecondActivity.kt

```kotlin
package com.example.intent_try

import android.support.v7.app.AppCompatActivity
import android.os.Bundle
import android.widget.EditText
import android.view.View
import android.content.Intent
import android.widget.Button

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        val message=findViewById<EditText>(R.id.mytext)
        val btn_send = findViewById<Button>(R.id.send)

        // Correctly set the onClickListener for the button
        btn_send.setOnClickListener(object: View.OnClickListener {
            override fun onClick(v: View?) {
                val intent = Intent(applicationContext, SecondActivity::class.java)
                intent.putExtra( name: "msg", message.text.toString())
                startActivity(intent)
            }
        })
    }
}
```

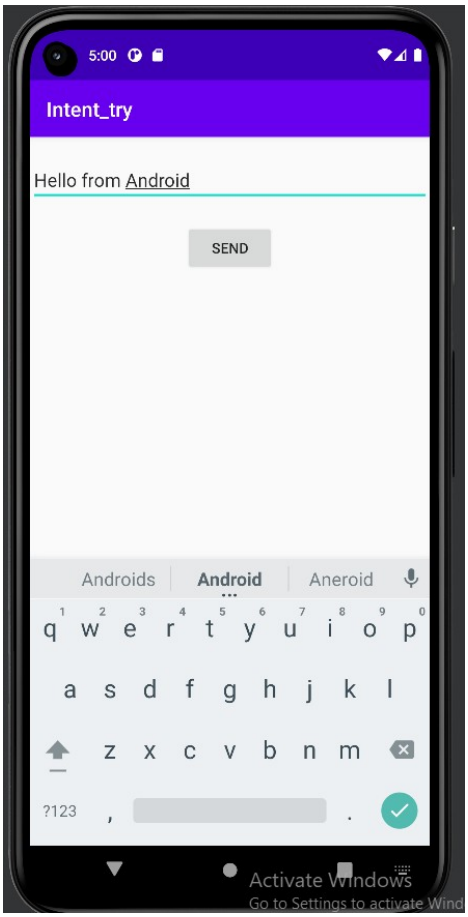
Step 5: Come to SecondActivity.kt file and add below code.

activity_main.xml | MainActivity.kt | activity_second.xml | SecondActivity.kt

```
package com.example.intent_try

import android.support.v7.app.AppCompatActivity
import android.os.Bundle
import android.widget.TextView

class SecondActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_second)
        val text_display=findViewById<TextView>(R.id.display)
        text_display.setText(intent.extras?.getString( key: "msg"))
    }
}
```

Step 6: Now run your project.

OUTPUT:


5:00
Intent_try
Hello from Android
SEND
Androids
Android
Aneroid
Activate Windows
Go to Settings to activate Windo


5:01
Intent_try
Hello from Android
Activate Windows
Go to Settings to activate Windo

PRACTICAL-8

Programs on Services, notification and broadcast receivers

1. Services

Step 1: Come to activity_main.xml file and add Textview and 2 buttons.

activity_main.xml | MainActivity.kt | AndroidManifest.xml | NewService.kt

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Services Example"
        android:textAlignment="center"
        android:textSize="40sp"
        android:textColor="@color/black"
        android:textStyle="bold" />
    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/btStart"
        android:text="Start Service"
        android:padding="10dp"
        android:textSize="30sp"
        android:layout_margin="20dp">
    </Button>
    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/btStop"
        android:text="Stop Service"
        android:padding="10dp"
        android:textSize="30sp"
        android:layout_margin="20dp"
        android:layout_marginBottom="50dp">
    </Button>
</LinearLayout>
```

Step 2: Now come to MainActivity.kt and add below code.

activity_main.xml | MainActivity.kt | AndroidManifest.xml | NewService.kt

```
package com.example.service_ex

import android.content.Intent
import android.support.v7.app.AppCompatActivity
import android.os.Bundle
import android.widget.Button

class MainActivity : AppCompatActivity() {
    private lateinit var btStart:Button
    private lateinit var btStop:Button

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        btStart=findViewById(R.id.btStart)
        btStop=findViewById(R.id.btStop)


        btStart.setOnClickListener { it: View!
            startService(Intent( packageContext: this,NewService::class.java))
        }

        btStop.setOnClickListener { it: View!
            stopService(Intent( packageContext: this,NewService::class.java))
        }


    }
}
```

Step 3: Now right click on com.example.service_ex -> select New -> Kotline class file -> give name as “NewService” and select class -> then click enter.

Step 4: Come to Newservice.kt file and add below code.

activity_main.xml | MainActivity.kt | AndroidManifest.xml | NewService.kt

```
package com.example.service_ex

import android.app.Service
import android.content.Intent
import android.media.MediaPlayer
import android.os.IBinder
import android.provider.Settings

class NewService: Service() {
    private lateinit var player:MediaPlayer

    override fun onStartCommand(intent: Intent?, flags: Int, startId: Int): Int {
        player=MediaPlayer.create( context: this, Settings.System.DEFAULT_RINGTONE_URI)
        player.isLooping=true
        player.start()
        return START_STICKY
    }

    override fun onDestroy(){
        super.onDestroy()
        player.stop()
    }
    override fun onBind(p0: Intent?): IBinder? {
        return null
    }

}
```

Step 5: Now come to AndroidManifest.xml file and add <service android:name=.NewService" /> **line to it.**

activity_main.xml | MainActivity.kt | AndroidManifest.xml | NewService.kt

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="Service_ex"
        android:supportsRtl="true"
        android:theme="@style/Theme.Service_ex"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <service android:name=".NewService"/>
    </application>

</manifest>
```

OUTPUT:

2. Notification

Step 1: In activity_main.xml file, add button and add below code.

activity_main.xml | MainActivity.kt

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Show Notification"
        android:layout_marginTop="250dp"
        android:layout_centerHorizontal="true"
        android:id="@+id/btn_show"/>

</RelativeLayout>

```

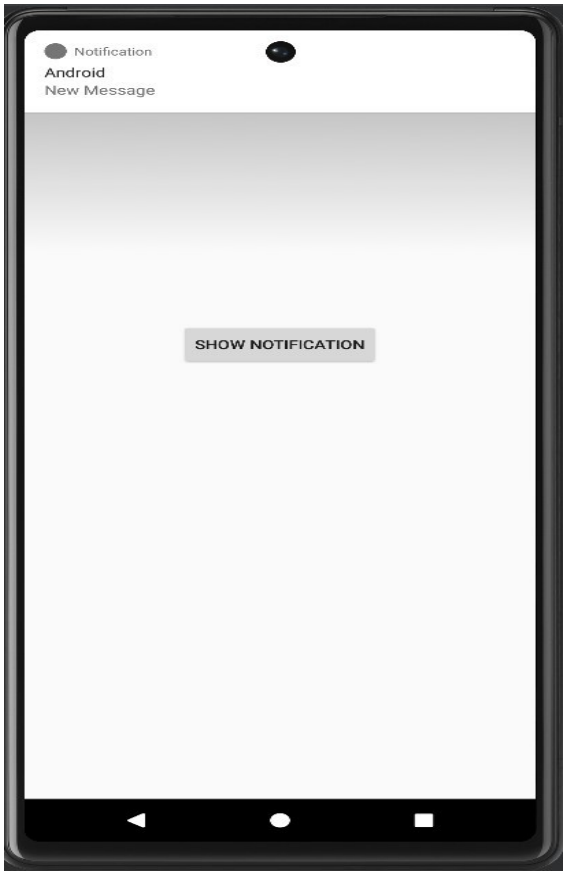
Step 2 : Come to MainActivity.kt file and add below code.

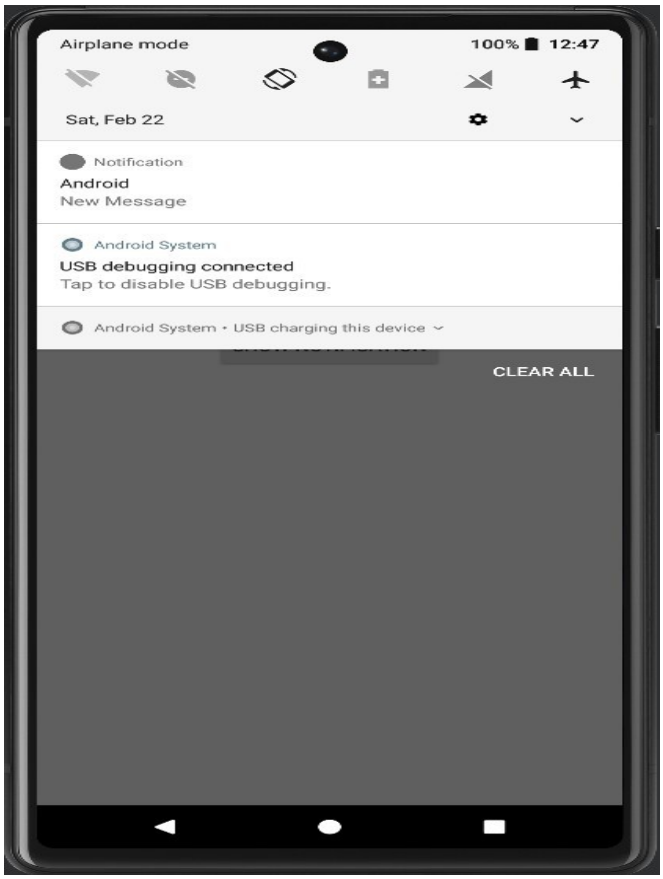
activity_main.xml | MainActivity.kt

```kotlin
package com.example.notification

import android.annotation.SuppressLint
import android.app.Notification
import android.app.NotificationChannel
import android.app.NotificationManager
import android.app.PendingIntent
import android.content.Context
import android.content.Intent
import android.graphics.Color
import android.os.Build
import android.support.v7.app.AppCompatActivity
import android.os.Bundle
import android.widget.Button

class MainActivity : AppCompatActivity() {
    lateinit var notificationManager: NotificationManager
    lateinit var notificationChannel: NotificationChannel
    lateinit var builder: Notification.Builder
    val channelId = "com.example.notification"
    val description = "My Notification"
    @SuppressLint("NotificationPermission")
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        val show = findViewById<Button>(R.id.btn_show)
        notificationManager = getSystemService(Context.NOTIFICATION_SERVICE) as NotificationManager
        show.setOnClickListener { it: View!

            val intent = Intent(applicationContext, MainActivity::class.java)
            val pendingIntent =
                PendingIntent.getActivity( context: this,  requestCode: 0, intent, PendingIntent.FLAG_UPDATE_CURRENT)


            if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
                notificationChannel =
                    NotificationChannel(channelId, description, NotificationManager.IMPORTANCE_HIGH)
                notificationChannel.enableLights( lights: true)
                notificationChannel.lightColor = Color.RED
                notificationChannel.enableVibration( vibration: true)
                notificationManager.createNotificationChannel(notificationChannel)

                builder = Notification.Builder( context: this, channelId)
                    .setContentTitle("Android")
                    .setContentText("New Message")
                    .setSmallIcon(R.mipmap.ic_launcher)
                    .setContentIntent(pendingIntent)
            } else {
                builder = Notification.Builder( context: this)
                    .setContentTitle("Android")
                    .setContentText("New Message")
                    .setSmallIcon(R.mipmap.ic_launcher)
                    .setContentIntent(pendingIntent)
            }
            if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
                notificationManager.notify( id: 0,builder.build())
            }
        }
    }
}
```

OUTPUT:


12:46
Notification
SHOW NOTIFICATION

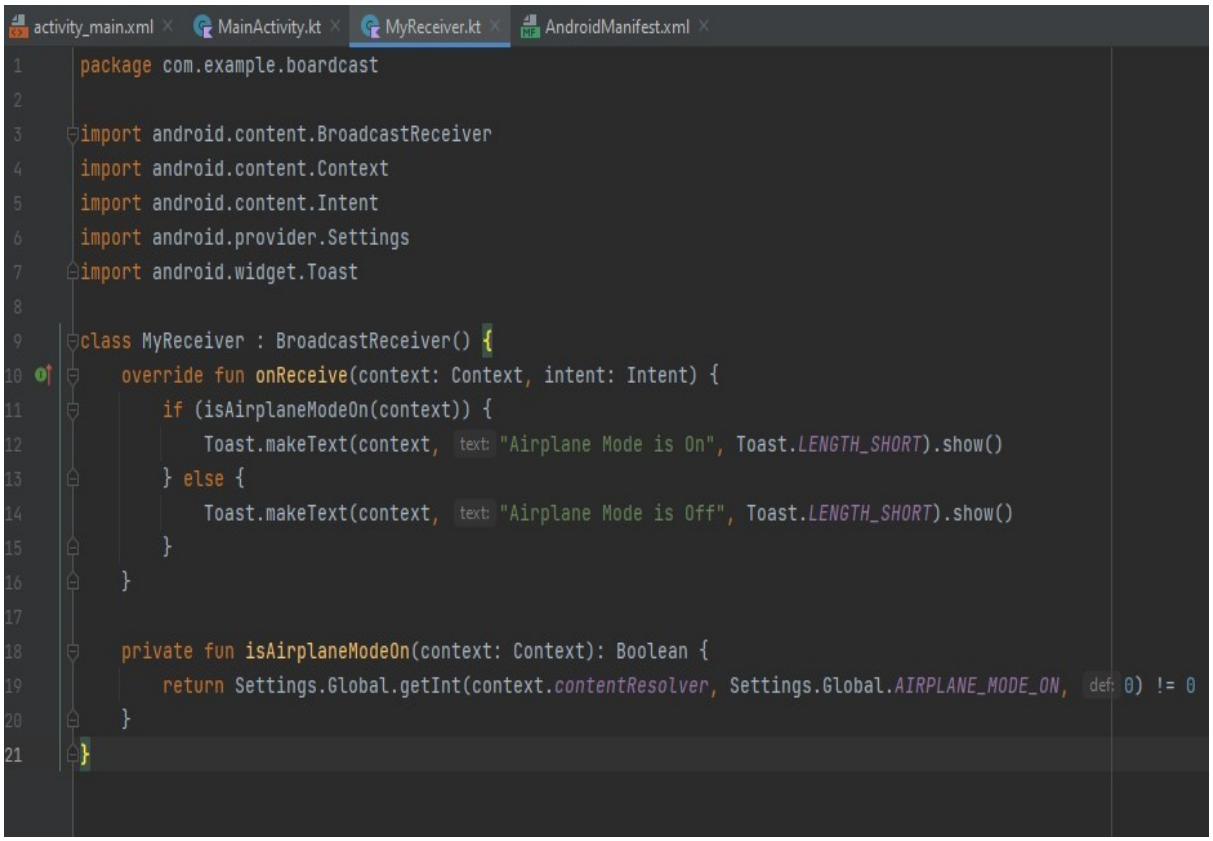

Notification
Android
New Message
SHOW NOTIFICATION



3. Broadcast Receivers

Step 1: Select “com.eample.broadcast”-> New-> select Kotlin Class File-> give name “MyReceiver”-> then select class-> press enter.

Step 2: Now come to MyReceiver.kt file and add below code.



Step 3: Come to MainActivity.kt file and add below code.

activity_main.xml | MainActivity.kt | MyReceiver.kt | AndroidManifest.xml

```kotlin
package com.example.boardcast

import android.content.BroadcastReceiver
import android.content.Context
import android.content.Intent
import android.content.IntentFilter
import android.os.Bundle
import android.widget.Toast
import android.support.v7.app.AppCompatActivity

class MainActivity : AppCompatActivity() {

    private val receiver = MyReceiver()

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
    }

    override fun onStart() {
        super.onStart()
        val filter = IntentFilter(Intent.ACTION_AIRPLANE_MODE_CHANGED)
        registerReceiver(receiver, filter)
    }

    override fun onStop() {
        super.onStop()
        unregisterReceiver(receiver)
    }
}
```

OUTPUT:


1:59
boardcast
Hello World!
Airplane Mode is On


1:58
boardcast
Hello World!
Airplane Mode is Off

PRACTICAL-9

a. Database Programming with SQLite

b. Programming Network Communications and Services (JSON)

a. Database Programming with SQLite

STEP 1: Add below code in activity_main.xml file.

activity_main.xml | UserModel.kt | DBContract.kt | UserDBHelper.kt | MainActivity.kt | build.gradle (:app)

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    tools:context=".MainActivity"
    android:gravity="center">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="SQLite Tutorial - User Management"
        android:textSize="20dp"
        android:padding="10dp" />
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">
        <EditText
            android:id="@+id/edittext_userid"
            android:hint="User ID"
            android:gravity="center"
            android:layout_width="match_parent"
            android:layout_height="wrap_content" />
        <EditText
            android:id="@+id/edittext_name"
            android:hint="User Name"
            android:gravity="center"
            android:layout_width="match_parent"
            android:layout_height="wrap_content" />
        <EditText
            android:id="@+id/edittext_age"
            android:hint="User Age"
            android:gravity="center"
            android:layout_width="match_parent"
            android:layout_height="wrap_content" />
```

```
    </LinearLayout>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
        <Button
            android:id="@+id/button_add_user"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:onClick="addUser"
            android:text="Add" />
        <Button
            android:id="@+id/button_delete_user"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:onClick="deleteUser"
            android:text="Delete" />
        <Button
            android:id="@+id/button_show_all"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:onClick="showAllUsers"
            android:text="Show All" />
    </LinearLayout>
    <TextView
        android:id="@+id/textview_result"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />
    <LinearLayout
        android:id="@+id/ll_entries"
        android:padding="15dp"
        android:orientation="vertical"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"></LinearLayout>
</LinearLayout>
```

STEP 2: Create new Kotlin class file and named that file as “UserModel.kt”. Then write below code.

activity_main.xml | UserModel.kt | DBContract.kt | UserDBHelper.kt | MainActivity.kt | build.gradle (:app)

```
package com.example.data

class UserModel(val userid: String, val name: String, val age: String)
```

STEP 3: Create new Kotlin class file and named that file as “DBContract.kt”. Then write below code.

activity_main.xml | UserModel.kt | DBContract.kt | UserDBHelper.kt | MainActivity.kt

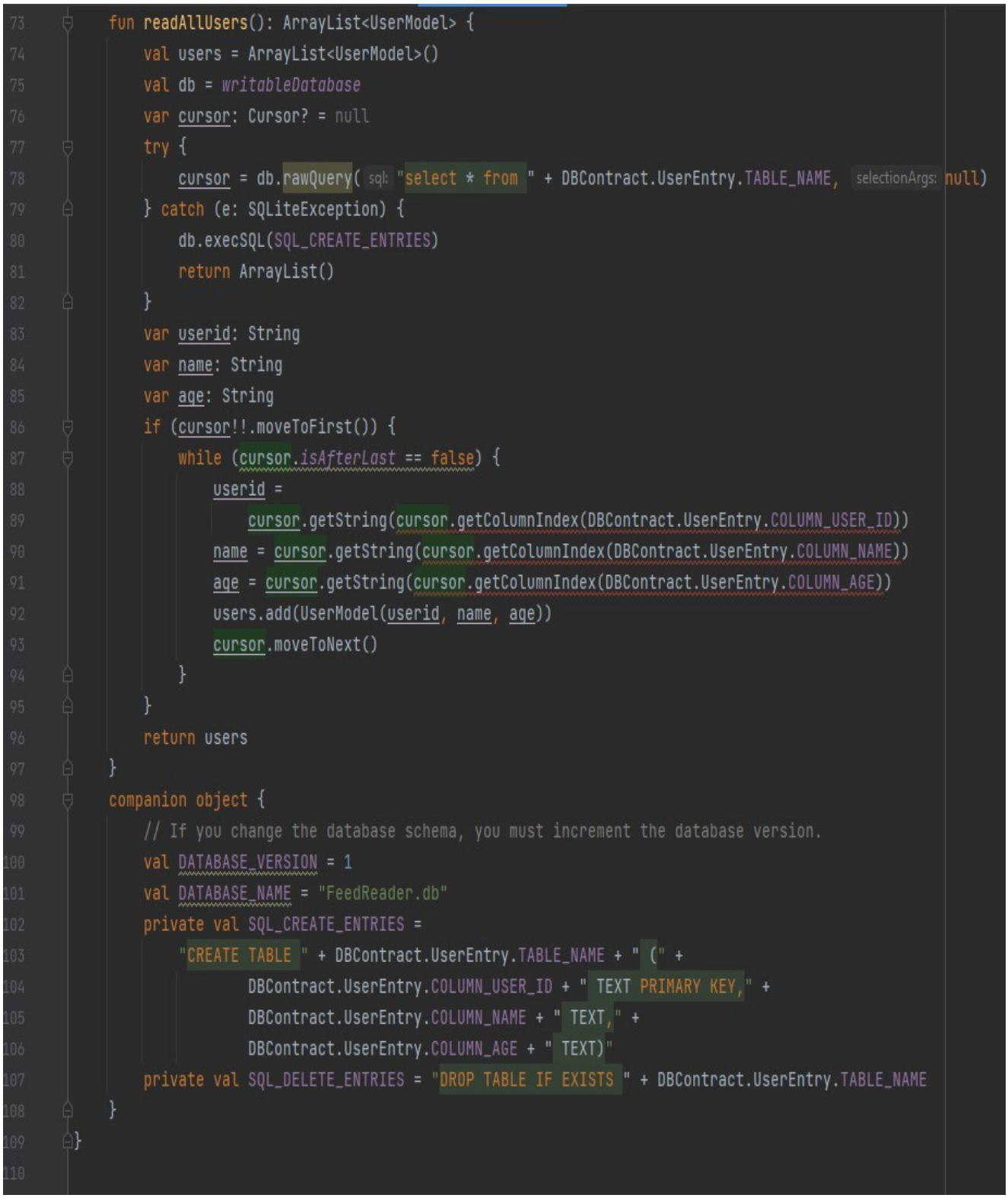
```kotlin
package com.example.data

import android.provider.BaseColumns

object DBContract {
    class UserEntry : BaseColumns {
        companion object {
            val TABLE_NAME = "users"
            val COLUMN_USER_ID = "userid"
            val COLUMN_NAME = "name"
            val COLUMN_AGE = "age"
        }
    }
}
```

STEP 4: Create new Kotlin class file and named that file as “UserDBHelper.kt”. Then write below code.

activity_main.xml | UserModel.kt | DBContract.kt | UserDBHelper.kt | MainActivity.kt | build.gradle (:app)

```kotlin
package com.example.data

import android.content.ContentValues
import android.content.Context
import android.database.Cursor
import android.database.sqlite.SQLiteConstraintException
import android.database.sqlite.SQLiteDatabase
import android.database.sqlite.SQLiteException
import android.database.sqlite.SQLiteOpenHelper
class UsersDBHelper(context: Context) : SQLiteOpenHelper(context, DATABASE_NAME, factory: null,
    DATABASE_VERSION) {
    override fun onCreate(db: SQLiteDatabase) {
        db.execSQL(SQL_CREATE_ENTRIES)
    }
    override fun onUpgrade(db: SQLiteDatabase, oldVersion: Int, newVersion: Int) {
// This database is only a cache for online data, so its upgrade policy is
// to simply to discard the data and start over
        db.execSQL(SQL_DELETE_ENTRIES)
        onCreate(db)
    }
    override fun onDowngrade(db: SQLiteDatabase, oldVersion: Int, newVersion: Int) {
        onUpgrade(db, oldVersion, newVersion)
    }
    @Throws(SQLiteConstraintException::class)
    fun insertUser(user: UserModel): Boolean {
// Gets the data repository in write mode
        val db = writableDatabase
// Create a new map of values, where column names are the keys
        val values = ContentValues()
        values.put(DBContract.UserEntry.COLUMN_USER_ID, user.userid)
        values.put(DBContract.UserEntry.COLUMN_NAME, user.name)
        values.put(DBContract.UserEntry.COLUMN_AGE, user.age)
// Insert the new row, returning the primary key value of the new row
        val newRowId = db.insert(DBContract.UserEntry.TABLE_NAME, nullColumnHack: null, values)
        return true
    }
```

```
    @Throws(SQLiteConstraintException::class)
    fun deleteUser(userid: String): Boolean {
// Gets the data repository in write mode
        val db = writableDatabase
// Define 'where' part of query.
        val selection = DBContract.UserEntry.COLUMN_USER_ID + " LIKE ?"
// Specify arguments in placeholder order.
        val selectionArgs = arrayOf(userid)
// Issue SQL statement.
        db.delete(DBContract.UserEntry.TABLE_NAME, selection, selectionArgs)
        return true
    }
    fun readUser(userid: String): ArrayList<UserModel> {
        val users = ArrayList<UserModel>()
        val db = writableDatabase
        var cursor: Cursor? = null
        try {
            cursor = db.rawQuery( sql: "select * from " + DBContract.UserEntry.TABLE_NAME + " WHERE " +
                    DBContract.UserEntry.COLUMN_USER_ID + "='" + userid + "'", selectionArgs: null)
        } catch (e: SQLiteException) {
// if table not yet present, create it
            db.execSQL(SQL_CREATE_ENTRIES)
            return ArrayList()
        }
        var name: String
        var age: String
        if (cursor!!.moveToFirst()) {
            while (cursor.isAfterLast == false) {
                name = cursor.getString(cursor.getColumnIndex(DBContract.UserEntry.COLUMN_NAME))
                age = cursor.getString(cursor.getColumnIndex(DBContract.UserEntry.COLUMN_AGE))
                users.add(UserModel(userid, name, age))
                cursor.moveToNext()
            }
        }
        return users
    }
```



```
    fun readAllUsers(): ArrayList<UserModel> {
        val users = ArrayList<UserModel>()
        val db = writableDatabase
        var cursor: Cursor? = null
        try {
            cursor = db.rawQuery( sql: "select * from " + DBContract.UserEntry.TABLE_NAME,  selectionArgs: null)
        } catch (e: SQLiteException) {
            db.execSQL(SQL_CREATE_ENTRIES)
            return ArrayList()
        }
        var userid: String
        var name: String
        var age: String
        if (cursor!!.moveToFirst()) {
            while (cursor.isAfterLast == false) {
                userid =
                    cursor.getString(cursor.getColumnIndex(DBContract.UserEntry.COLUMN_USER_ID))
                name = cursor.getString(cursor.getColumnIndex(DBContract.UserEntry.COLUMN_NAME))
                age = cursor.getString(cursor.getColumnIndex(DBContract.UserEntry.COLUMN_AGE))
                users.add(UserModel(userid, name, age))
                cursor.moveToNext()
            }
        }
        return users
    }
    companion object {
        // If you change the database schema, you must increment the database version.
        val DATABASE_VERSION = 1
        val DATABASE_NAME = "FeedReader.db"
        private val SQL_CREATE_ENTRIES =
            "CREATE TABLE " + DBContract.UserEntry.TABLE_NAME + " (" +
                    DBContract.UserEntry.COLUMN_USER_ID + " TEXT PRIMARY KEY," +
                    DBContract.UserEntry.COLUMN_NAME + " TEXT," +
                    DBContract.UserEntry.COLUMN_AGE + " TEXT)"
        private val SQL_DELETE_ENTRIES = "DROP TABLE IF EXISTS " + DBContract.UserEntry.TABLE_NAME
    }
}
```

STEP 5: After this come to “MainActivity.kt” file and below code.

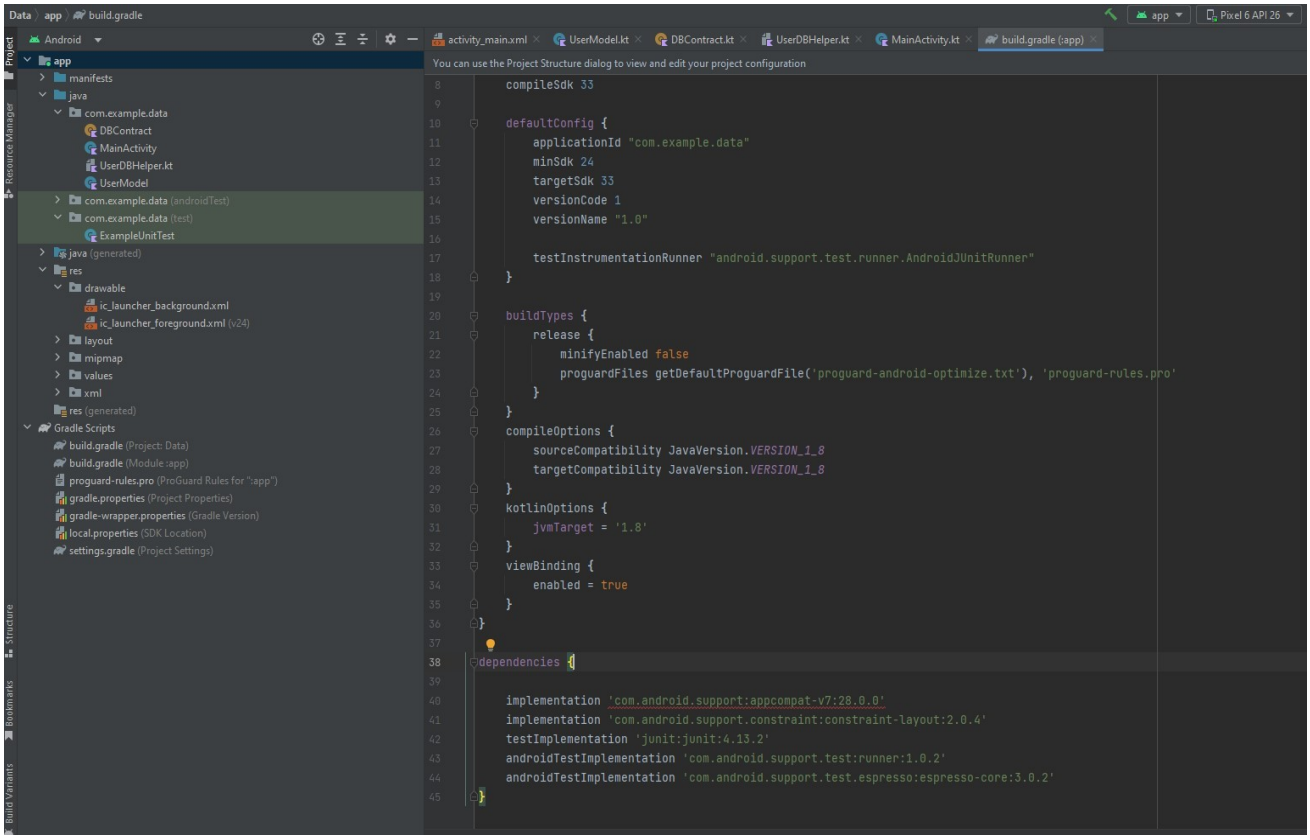
activity_main.xml | UserModel.kt | DBContract.kt | UserDBHelper.kt | MainActivity.kt | build.gradle (:app)

```kotlin
package com.example.data

import android.os.Bundle
import android.support.v7.app.AppCompatActivity
import android.view.View
import android.widget.TextView
import com.example.data.UserModel
import com.example.data.UsersDBHelper
import com.example.data.databinding.ActivityMainBinding


class MainActivity : AppCompatActivity() {
    lateinit var usersDBHelper: UsersDBHelper
    private lateinit var binding: ActivityMainBinding

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityMainBinding.inflate(layoutInflater) // Inflate the view binding
        setContentView(binding.root) // Set the root view

        usersDBHelper = UsersDBHelper( context: this)
    }

    fun addUser(v: View) {
        val userid = binding.edittextUserid.text.toString() // Access views using binding
        val name = binding.edittextName.text.toString()
        val age = binding.edittextAge.text.toString()

        val result = usersDBHelper.insertUser(UserModel(userid = userid, name = name, age = age))

        // Clear all EditTexts
        binding.edittextAge.text.clear()
        binding.edittextName.text.clear()
        binding.edittextUserid.text.clear()

        binding.textviewResult.text = "Added user: $result"
        binding.llEntries.removeAllViews()
    }
```

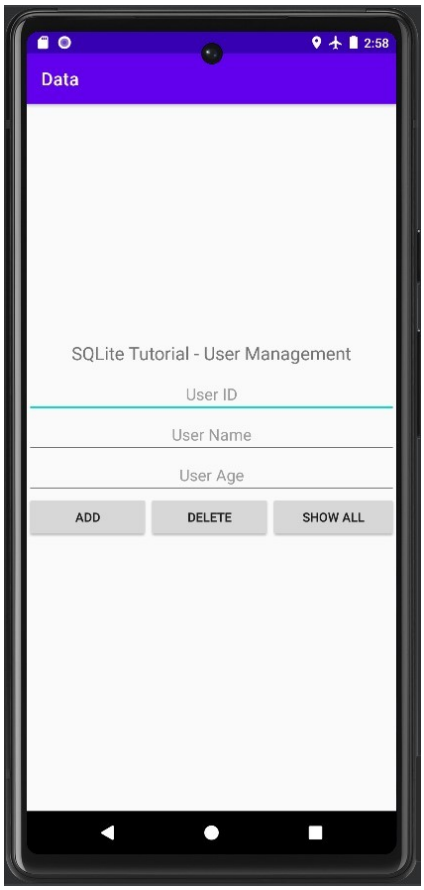
```
39
40      fun deleteUser(v: View) {
41          val userid = binding.edittextUserid.text.toString()
42          val result = usersDBHelper.deleteUser(userid)
43          binding.textviewResult.text = "Deleted user: $result"
44          binding.llEntries.removeAllViews()
45      }
46
47      fun showAllUsers(v: View) {
48          val users = usersDBHelper.readAllUsers()
49          binding.llEntries.removeAllViews()
50          users.forEach { it: UserModel
51              val tvUser = TextView( context: this)
52              tvUser.textSize = 30F
53              tvUser.text = "${it.name} - ${it.age}"
54              binding.llEntries.addView(tvUser)
55          }
56          binding.textviewResult.text = "Fetched ${users.size} users"
57      }
58  }
59
```

STEP 6: Now come to "build.gradle (Module:app)" file and below lines.

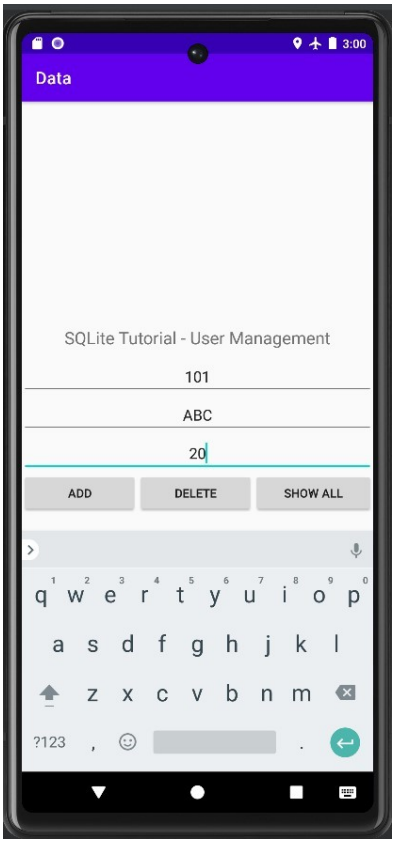
viewBinding{
enabled=true
}

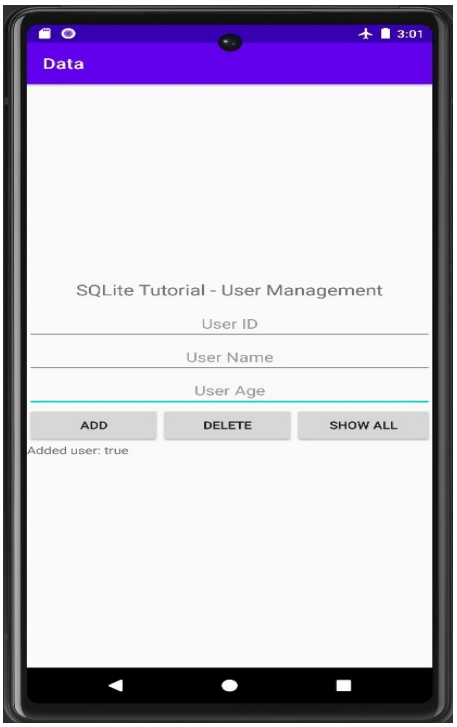


STEP 7: Run Project.



Insert data and click on “ADD” button.





Using “SHOW ALL” button, you will get all the records inserted earlier.

Using “DELETE” button you can easily delete any record.

PRACTICAL-10

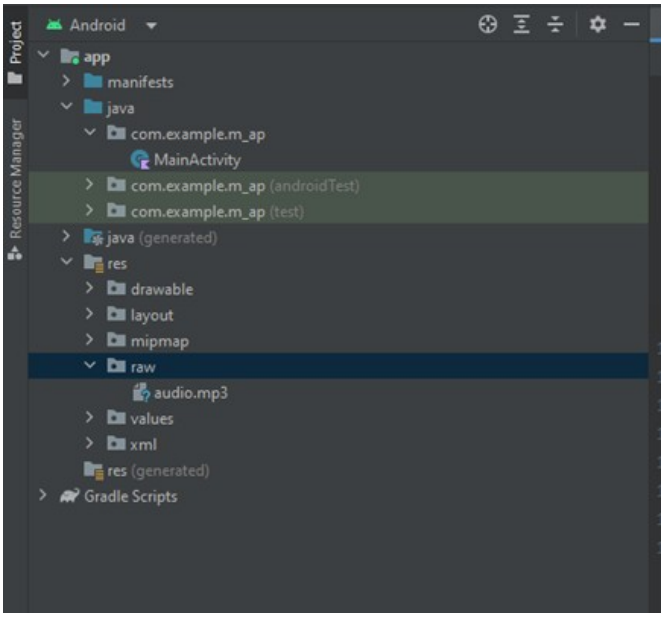
a. Programming Media API and Telephone API

b. Programming Security and Permissions

a. Programming Media API and Telephone API

MEDIA API

Step 1: Under res folder create new directory named as “raw”-> then download any audio file and Paste that file inside the raw directory.



Step 2: Now come to your activity_main.xml file and add below code.

activity_main.xml | MainActivity.kt

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/playButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Play Audio" />

</LinearLayout>
```

Step 3: After this come to your MainActivity.kt file and add below code.

activity_main.xml | MainActivity.kt

```kotlin
package com.example.m_ap

import android.support.v7.app.AppCompatActivity
import android.os.Bundle
import android.widget.Button
import android.media.MediaPlayer

class MainActivity : AppCompatActivity() {
    private var mediaPlayer: MediaPlayer? = null
    private lateinit var playButton: Button
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        playButton = findViewById(R.id.playButton)

        // Initialize MediaPlayer with the audio file located in res/raw
        mediaPlayer = MediaPlayer.create( context: this, R.raw.audio) // audio.mp3 in res/raw folder

        // Set an OnClickListener to the button to play audio
        playButton.setOnClickListener { it: View!
            mediaPlayer?.start() // Start playing the audio
        }
    }

    override fun onDestroy() {
        super.onDestroy()
        mediaPlayer?.release() // Release the media player when done
    }
}
```

Step 4: Run your project.

OUTPUT:

TELEPHONE API

Step 1: Come to activity_main.xml and add below code.

activity_main.xml | MainActivity.kt | AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/phonecall"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Place Call"
        android:layout_marginTop="200dp"/>

</LinearLayout>
```

Step 2: Add required permissions to AndroidMainfest.xml file.

activity_main.xml | MainActivity.kt | AndroidManifest.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">
    <uses-permission android:name="android.permission.CALL_PHONE" />
    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="Telephone_api"
        android:supportsRtl="true"
        android:theme="@style/Theme.Telephone_api"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

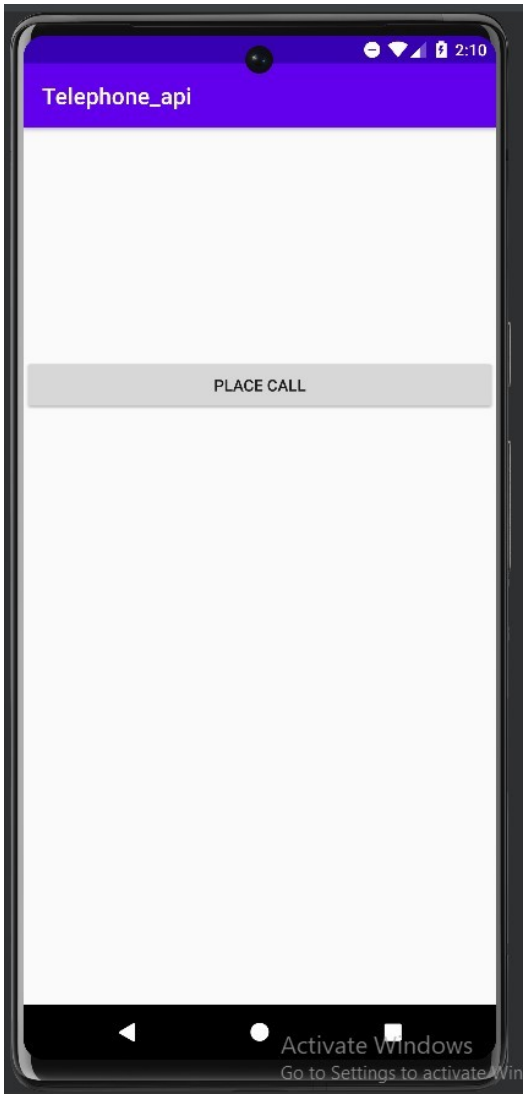
Step 3: Now come to MainActivity.kt file and add below code.

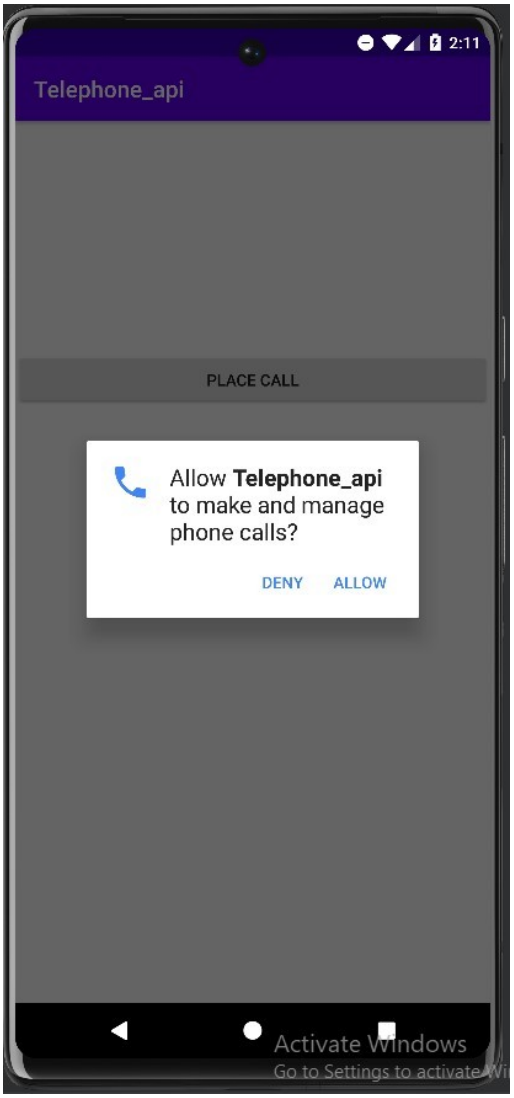
activity_main.xml | MainActivity.kt | AndroidManifest.xml

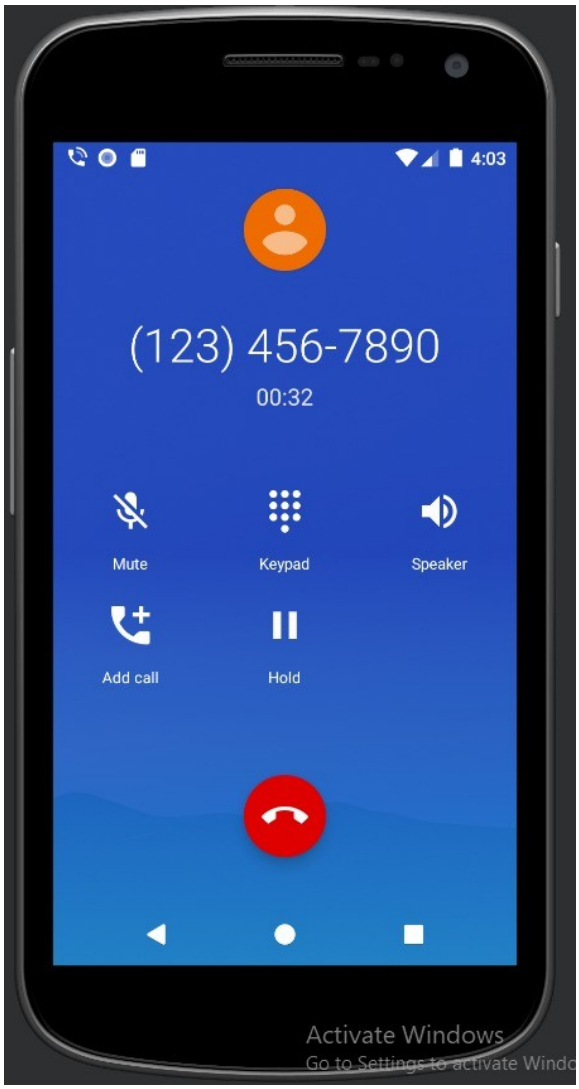
```kotlin
package com.example.telephone_api

import android.content.Intent
import android.content.pm.PackageManager
import android.net.Uri
import android.support.v7.app.AppCompatActivity
import android.os.Bundle
import android.support.v4.app.ActivityCompat
import android.widget.Button

class MainActivity : AppCompatActivity() {

    private val phoneNumber: String = "1234567890"
    private val REQUEST_PHONE_CALL = 1
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        val callButton: Button = findViewById(R.id.phonecall)

        // Set up click listener for the button
        callButton.setOnClickListener { it: View!
            // Check for permission to make a call
            if (ActivityCompat.checkSelfPermission( context: this, android.Manifest.permission.CALL_PHONE) != PackageManager.PERMISSION_GRANTED) {
                // If permission not granted, request permission
                ActivityCompat.requestPermissions( activity: this, arrayOf(android.Manifest.permission.CALL_PHONE), REQUEST_PHONE_CALL)
            } else {
                // If permission granted, make the call
                makeCall()
            }
        }
    }

```

```
    // Function to initiate the phone call
    private fun makeCall() {
        val intent = Intent(Intent.ACTION_CALL, Uri.parse( uriString: "tel:$phoneNumber"))
        startActivity(intent)
    }

    // Handle permission request result
    override fun onRequestPermissionsResult(requestCode: Int, permissions: Array<out String>, grantResults: IntArray) {
        super.onRequestPermissionsResult(requestCode, permissions, grantResults)

        // If permission is granted, make the call
        if (requestCode == REQUEST_PHONE_CALL && grantResults.isNotEmpty() && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
            makeCall()
        }
    }
}
```

OUTPUT:






4:03
(123) 456-7890
00:32
Mute
Keypad
Speaker
Add call
Hold
Activate Windows
Go to Settings to activate Windo

b. Programming Security and Permissions

Step 1: Add required permissions to AndroidMainfest.xml file.

activity_main.xml | MainActivity.kt | AndroidManifest.xml

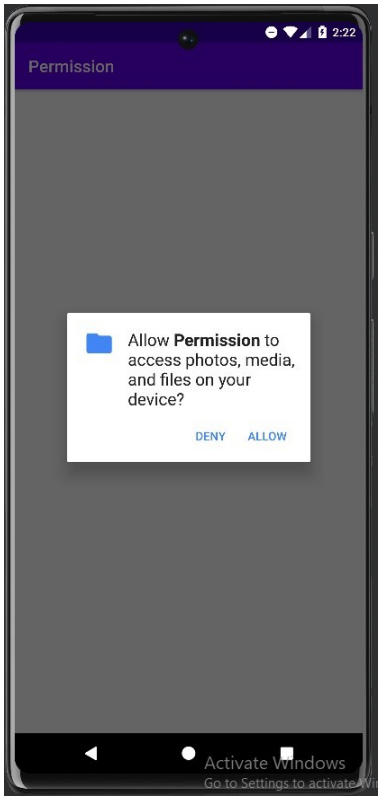
```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="Permission"
        android:supportsRtl="true"
        android:theme="@style/Theme.Permission"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

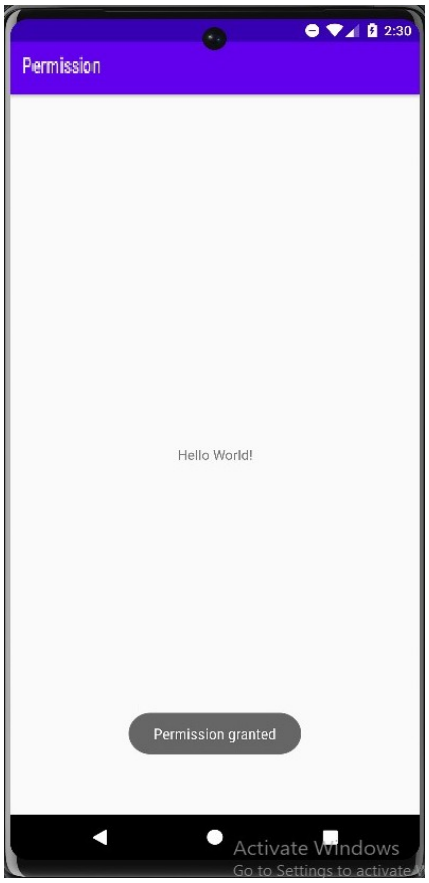
Step 2: Now come to MainActivity.kt file and add below code.

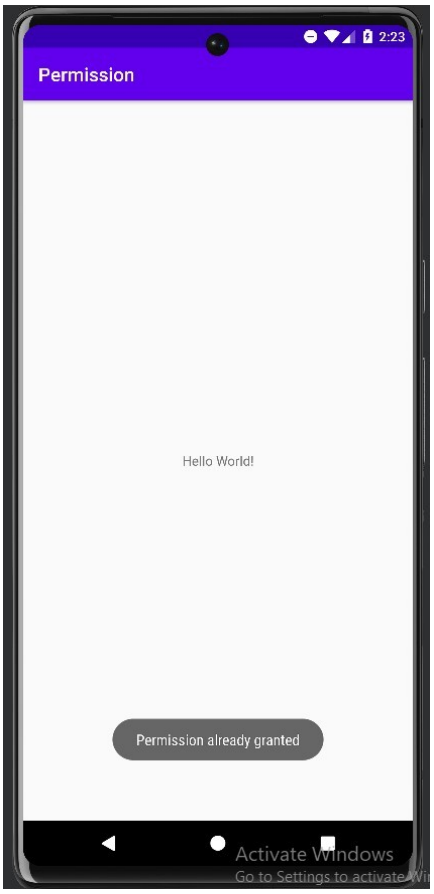
activity_main.xml | MainActivity.kt | AndroidManifest.xml

```kotlin
package com.example.permission

import android.support.v7.app.AppCompatActivity
import android.os.Bundle
import android.support.v4.content.ContextCompat
import android.Manifest
import android.content.pm.PackageManager
import android.os.Build
import android.support.v4.app.ActivityCompat
import android.widget.Toast


class MainActivity : AppCompatActivity() {

    private val PERMISSION_REQUEST_CODE = 1001

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        // Check if permission is already granted
        if (ContextCompat.checkSelfPermission( context: this, Manifest.permission.READ_EXTERNAL_STORAGE)
            != PackageManager.PERMISSION_GRANTED) {

            // Request permission if not granted
            if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
                ActivityCompat.requestPermissions(
                    activity: this,
                    arrayOf(Manifest.permission.READ_EXTERNAL_STORAGE),
                    PERMISSION_REQUEST_CODE
                )
            }
        } else {
            // Permission is already granted, proceed with your action
            Toast.makeText( context: this,  text: "Permission already granted", Toast.LENGTH_SHORT).show()
        }
    }

    // Handle permission result
    override fun onRequestPermissionsResult(
        requestCode: Int, permissions: Array<out String>, grantResults: IntArray
    ) {
        super.onRequestPermissionsResult(requestCode, permissions, grantResults)

        if (requestCode == PERMISSION_REQUEST_CODE) {
            if (grantResults.isNotEmpty() && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
                // Permission granted
                Toast.makeText( context: this,  text: "Permission granted", Toast.LENGTH_SHORT).show()
            } else {
                // Permission denied
                Toast.makeText( context: this,  text: "Permission denied", Toast.LENGTH_SHORT).show()
            }
        }
    }
}
```

OUTPUT:


2:22
Permission
Allow Permission to access photos, media, and files on your device?
DENY
ALLOW
Activate Windows
Go to Settings to activate Wi


2:30
Permission
Hello World!
Permission granted
Activate Windows
Go to Settings to activate


2:23
Permission
Hello World!
Permission already granted
Activate Windows
Go to Settings to activate Wi