# Exercise 1:

Fragment 1:

Sets fragment background colour to E77272 and specifies the associated class i.e. Fragment1

<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="#E77272"  
 tools:context=".Fragment1">  
  
 <!-- *TODO: Update blank fragment layout* -->  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:text="@string/hello\_blank\_fragment" />  
  
</FrameLayout>

Fragment 2:

Sets fragment background colour to 009688 and specifies the associated class i.e. Fragment2.

<?xml version="1.0" encoding="utf-8"?>  
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="#009688"  
 tools:context=".Fragment2">  
  
 <!-- *TODO: Update blank fragment layout* -->  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:text="@string/hello\_blank\_fragment" />  
  
</FrameLayout>

Activity\_main.xml:

This XML defines a horizontally oriented LinearLayout containing two fragments (Fragment1 and Fragment2), each taking up half the screen width using layout\_weight value 1 for both.

<LinearLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:id="@+id/main"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="horizontal">  
  
 <fragment  
 android:id="@+id/fragment1"  
 android:name="com.example.myfragment.Fragment1"  
 android:layout\_width="0dp"  
 android:layout\_height="match\_parent"  
 android:layout\_weight="1" />  
  
 <fragment  
 android:id="@+id/fragment2"  
 android:name="com.example.myfragment.Fragment2"  
 android:layout\_width="0dp"  
 android:layout\_height="match\_parent"  
 android:layout\_weight="1" />  
</LinearLayout>

Screenshot:

A close-up of a pink and blue striped background

Description automatically generated

# Exercise 2:

Toolbar setup:

Menu creation: The following code inflates the menu from the menu\_main.xml file to display menu options in the toolbar:

@Override  
public boolean onCreateOptionsMenu(Menu menu) {  
 getMenuInflater().inflate(R.menu.*menu\_main*, menu);  
 return true;  
}

Menu Item Selection:

The following code handles user interaction with the menu items and changes the background color:

@Override  
public boolean onOptionsItemSelected(MenuItem item) {  
 ConstraintLayout mainLayout = findViewById(R.id.*mainLayout*);  
  
  
 int itemId = item.getItemId();  
  
  
 if (itemId == R.id.*menu\_red*) {  
 item.setChecked(!item.isChecked());  
 mainLayout.setBackgroundColor(Color.*RED*);  
 return true;  
 } else if (itemId == R.id.*menu\_green*) {  
 item.setChecked(!item.isChecked());  
 mainLayout.setBackgroundColor(Color.*GREEN*);  
 return true;  
 } else if (itemId == R.id.*menu\_yellow*) {  
 item.setChecked(!item.isChecked());  
 mainLayout.setBackgroundColor(Color.*YELLOW*);  
 return true;  
 } else if (itemId == R.id.*menu\_blue*) {  
 item.setChecked(!item.isChecked());  
 mainLayout.setBackgroundColor(Color.*BLUE*);  
 return true;  
 } else if (itemId == R.id.*menu\_pink*) {  
 item.setChecked(!item.isChecked());  
 mainLayout.setBackgroundColor(Color.*MAGENTA*); // Pink  
 return true;  
 } else {  
 return super.onOptionsItemSelected(item);  
 }  
}

Menu\_main.xml

The following code defines a menu with options (Red, Green, Yellow, Blue, Pink), each checkable, allowing users to select a color and change the app's background dynamically.

<menu xmlns:android="http://schemas.android.com/apk/res/android">  
 <item  
 android:id="@+id/menu\_red"  
 android:title="Red"  
 android:checkable="true" />  
 <item  
 android:id="@+id/menu\_green"  
 android:title="Green"  
 android:checkable="true" />  
 <item  
 android:id="@+id/menu\_yellow"  
 android:title="Yellow"  
 android:checkable="true" />  
 <item  
 android:id="@+id/menu\_blue"  
 android:title="Blue"  
 android:checkable="true" />  
 <item  
 android:id="@+id/menu\_pink"  
 android:title="Pink"  
 android:checkable="true" />  
</menu>

Strings.xml

The following code defines the string resources for the application.

<resources>  
 <string name="app\_name">Menu Application</string>  
 <string name="menu\_red">Red</string>  
 <string name="menu\_green">Green</string>  
 <string name="menu\_yellow">Yellow</string>  
 <string name="menu\_blue">Blue</string>  
 <string name="menu\_pink">Pink</string>  
  
</resources>

A screenshot of a phone

Description automatically generatedA white background with black border

Description automatically generatedA screenshot of a green screen

Description automatically generatedA screenshot of a blue screen

Description automatically generated

# Exercise 3:

Activity\_main.xml

The following codes displays a text view that displays a message and a button which triggers a message.

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:paddingTop="100dp"  
 android:paddingLeft="20dp"  
 android:paddingRight="20dp"  
 android:gravity="center">  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Clicking the button will display the SnackBar"  
 android:gravity="center\_horizontal"  
 android:layout\_marginBottom="16dp" />  
  
 <Button  
 android:id="@+id/callbackButton"  
 android:layout\_width="300dp"  
 android:layout\_height="50dp"  
 android:gravity="center"  
 android:layout\_margin="16dp"  
 android:background="#9C27B0"  
 android:backgroundTint="@color/black"  
 android:text="Tap To Display SnackBar Message"  
 android:textColor="@android:color/white" />  
  
 </LinearLayout>  
  
</androidx.constraintlayout.widget.ConstraintLayout>

Mainactivity.java  
  
The following code displays a TextView that shows a message and a Button that triggers a Snackbar message when clicked.

package com.example.snackbarapplication; // Update package name accordingly  
  
import android.os.Bundle;  
import android.view.View;  
import com.google.android.material.snackbar.Snackbar; // Highlighted Snackbar import  
import androidx.appcompat.app.AppCompatActivity;  
  
public class MainActivity extends AppCompatActivity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*); // Set the layout for the activity  
  
 // Highlighted code to show Snackbar on button click  
 findViewById(R.id.*callbackButton*).setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 Snackbar.*make*(view, "This is a Snackbar", Snackbar.*LENGTH\_LONG*).show(); // Display Snackbar  
 }  
 });  
 }  
}

A white screen with black text

Description automatically generatedA white screen with black text

Description automatically generated