# Practical No- 1

###### Experiment 1: Install Android Studio and Run Hello World App.

Android Studio provides a complete integrated development environment (IDE) including an advanced code editor and a set of app templates. In addition, it contains tools for development, debugging, testing, and performance that make it faster and easier to develop apps. You can test your apps with a large range of preconfigured emulators or on your own mobile device, build production apps, and publish on the Google Play store.

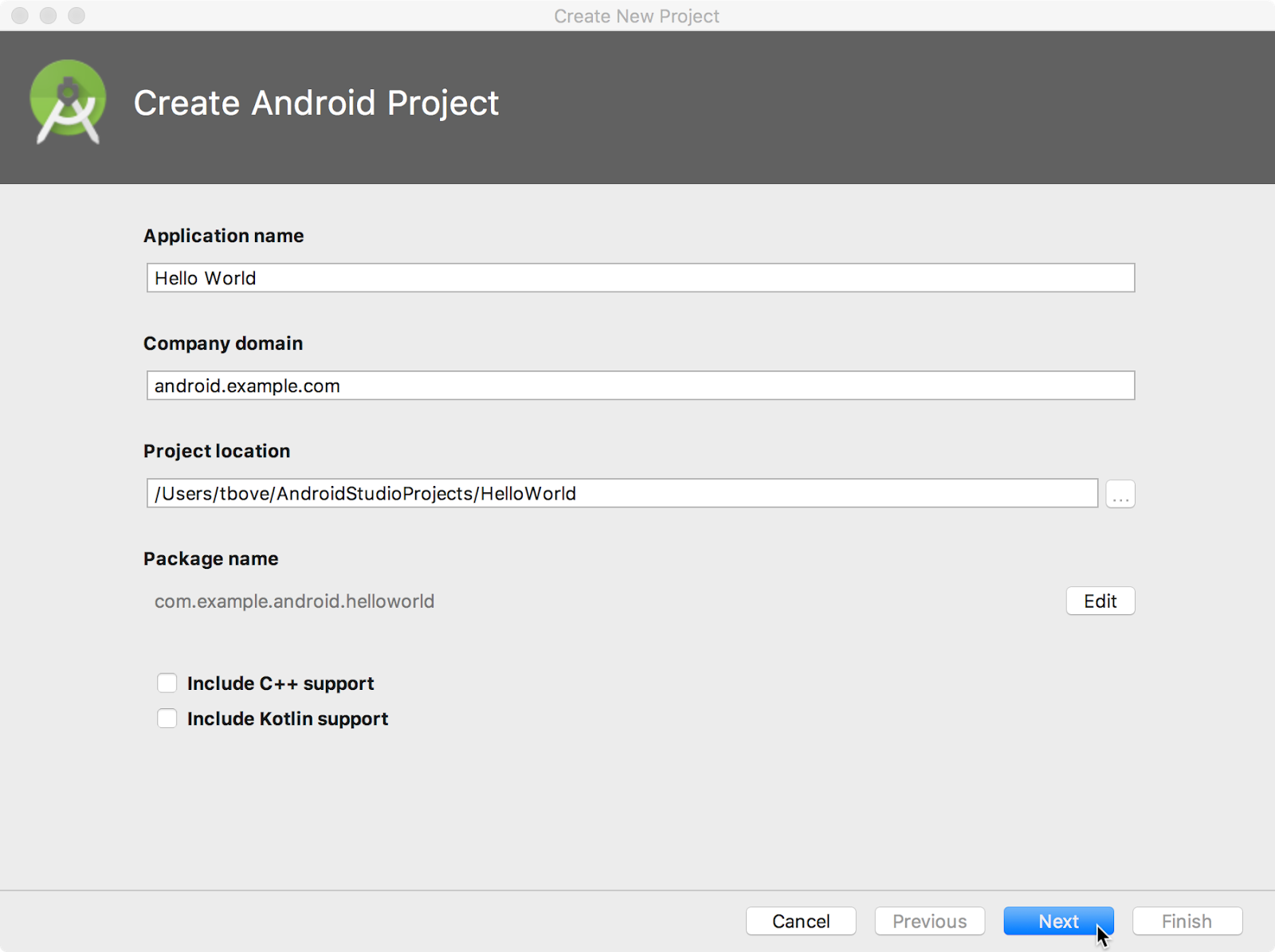
1. Navigate to the [Android developers site](https://developer.android.com/sdk/index.html) and follow the instructions to download and [install](https://developer.android.com/studio/install.html) [Android Studio](https://developer.android.com/studio/install.html).
2. Accept the default configurations for all steps, and ensure that all components are selected for installation.
3. After finishing the install, the Setup Wizard will download and install some additional components including the Android SDK. Be patient, this might take some time depending on your Internet speed, and some of the steps may seem redundant.
4. When the download completes, Android Studio will start, and you are ready to create your first project.

Task 2: Create the Hello World app

In this task, you will create an app that displays "Hello World" to verify that Android studio is correctly installed, and to learn the basics of developing with Android Studio.

* 1. Create the app project

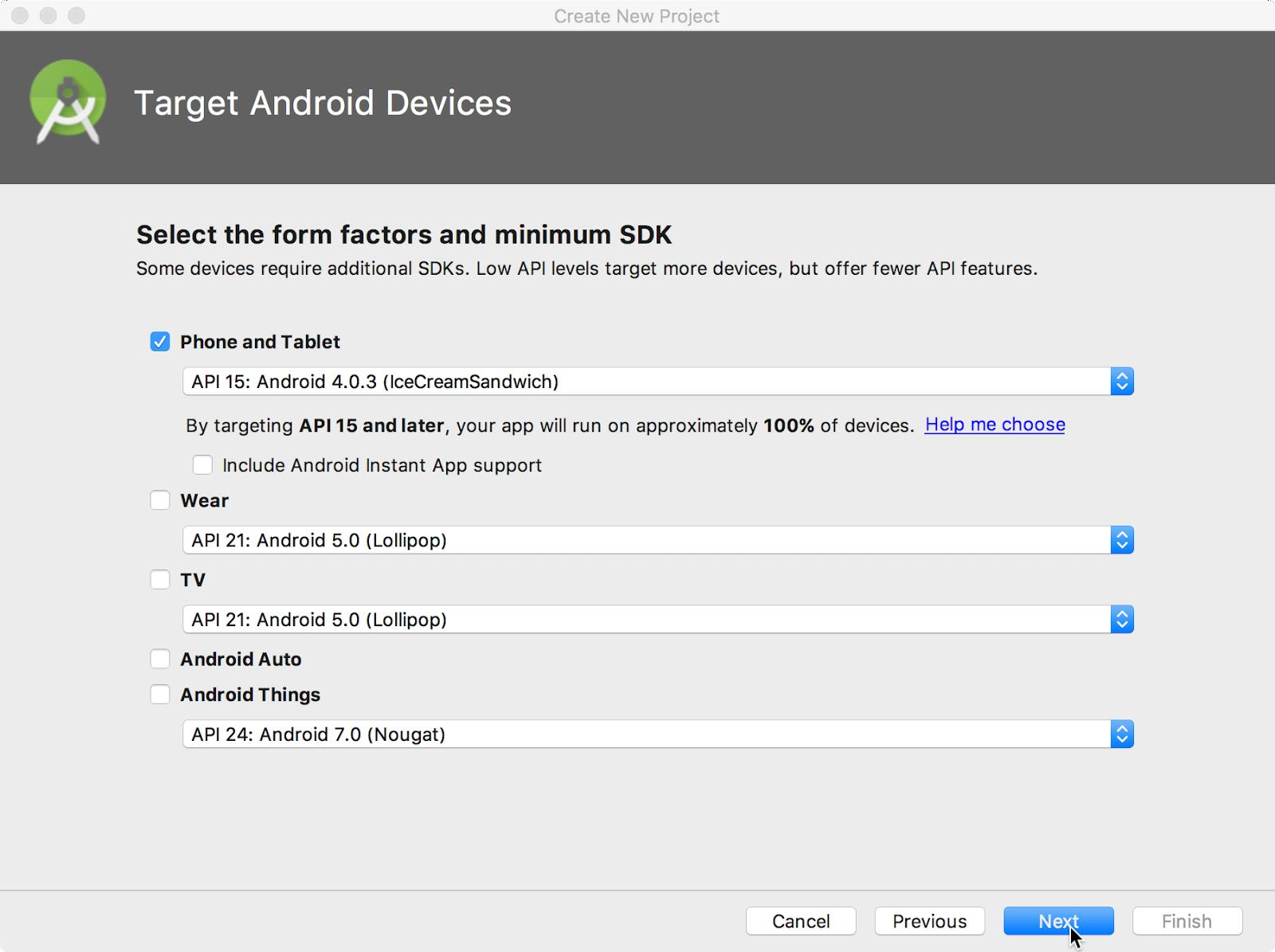
1. Open Android Studio if it is not already opened.
2. In the main **Welcome to Android Studio** window, click **Start a new Android Studio project**. In the **Create Android Project** window, enter **Hello World** for the **Application name**.



1. Verify that the default **Project location** is where you want to store your Hello World app and other Android Studio projects, or change it to your preferred directory.
2. Accept the default **android.example.com** for **Company Domain**, or create a unique company domain.

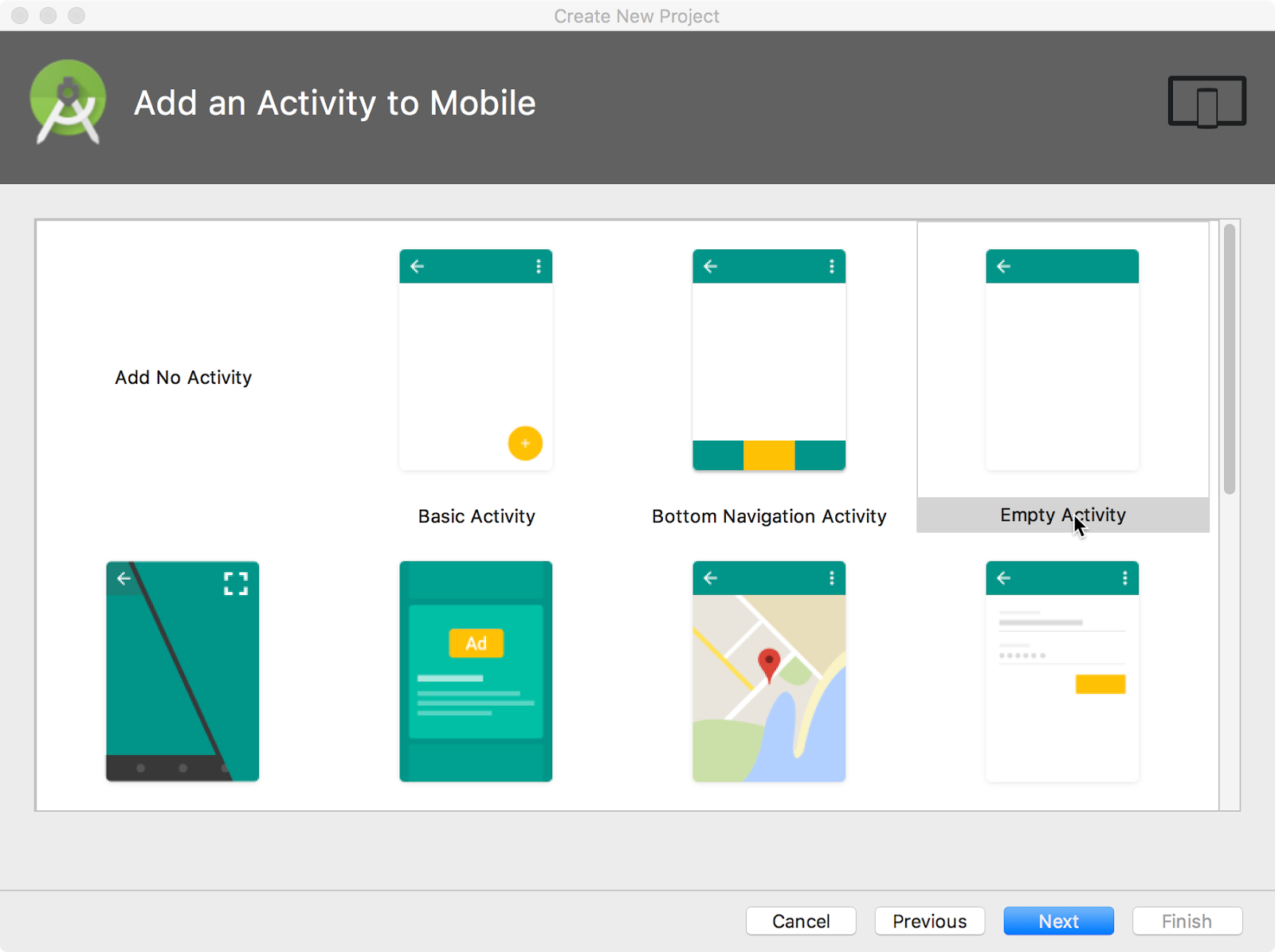
If you are not planning to publish your app, you can accept the default. Be aware that changing the package name of your app later is extra work.

1. Leave unchecked the options to **Include C++ support** and **Include Kotlin support**, and click **Next**.
2. On the **Target Android Devices** screen, **Phone and Tablet** should be selected. Ensure that **API 15: Android 4.0.3 IceCreamSandwich** is set as the Minimum SDK; if it is not, use the popup menu to set it.

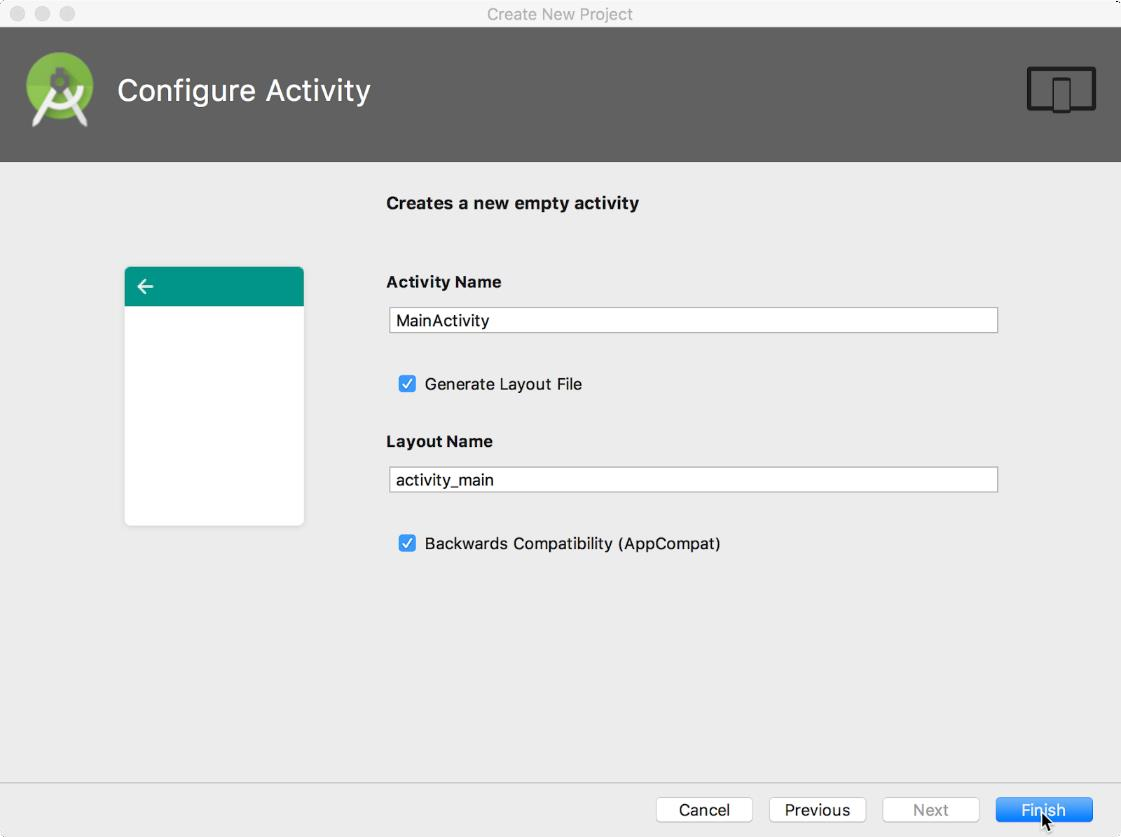


These are the settings used by the examples in the lessons for this course. As of this writing, these settings make your Hello World app compatible with 97% of Android devices active on the Google Play Store.

1. Leave unchecked the **Include Instant App support** and all other options. Then click **Next**. If your project requires additional components for your chosen target SDK, Android Studio will install them automatically.
2. The **Add an Activity** window appears. An [Activity](https://developer.android.com/reference/android/app/Activity.html) is a single, focused thing that the user can do. It is a crucial component of any Android app. An Activity typically has a layout associated with it that defines how UI elements appear on a screen. Android Studio provides Activity templates to help you get started. For the Hello World project, choose **Empty Activity** as shown below, and click **Next**.



1. The **Configure Activity** screen appears (which differs depending on which template you chose in the previous step). By default, the empty Activity provided by the template is named MainActivity. You can change this if you want, but this lesson uses MainActivity.



1. Make sure that the **Generate Layout file** option is checked. The layout name by default is activity\_main. You can change this if you want, but this lesson uses activity\_main.
2. Make sure that the **Backwards Compatibility (App Compat)** option is checked. This ensures that your app will be backwards-compatible with previous versions of Android.
3. Click **Finish**.

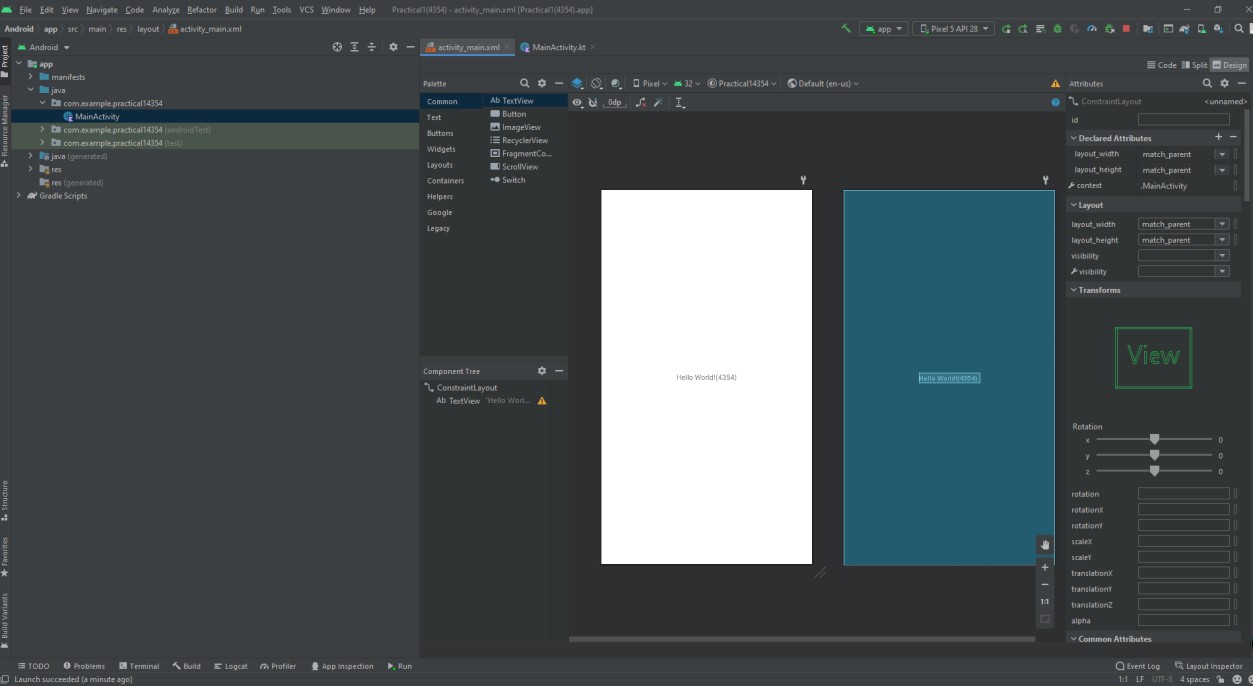
Android Studio creates a folder for your projects, and builds the project with [Gradle](https://gradle.org/) (this may take a few moments).

**Tip**: See the [Configure your build](https://developer.android.com/studio/build/index.html) developer page for detailed information.

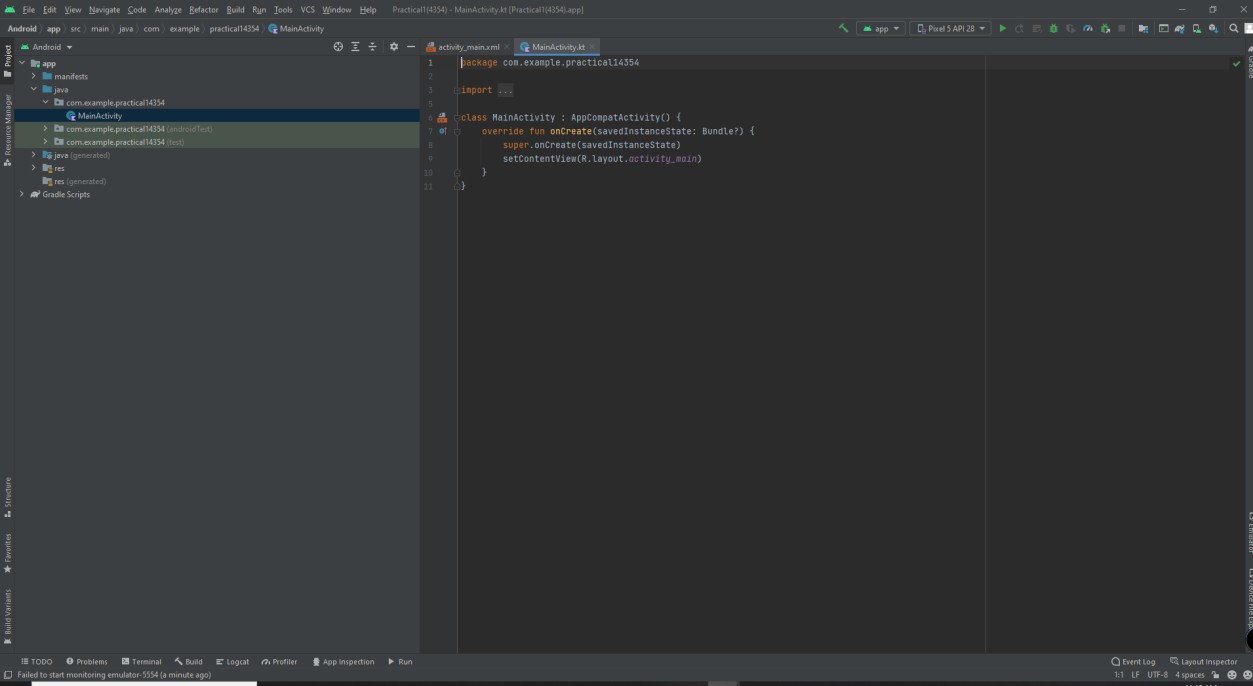
You may also see a "Tip of the day" message with keyboard shortcuts and other useful tips. Click **Close** to close the message.

The Android Studio editor appears. Follow these steps:

1. Click the **activity\_main.xml** tab to see the layout editor.
2. Click the layout editor **Design** tab, if not already selected, to show a graphical rendition of the layout as shown below.



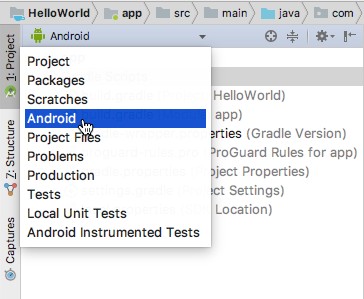
1. Click the **MainActivity.java** tab to see the code editor as shown below.



* 1. Explore the Project > Android pane

In this practical, you will explore how the project is organized in Android Studio.

1. If not already selected, click the **Project** tab in the vertical tab column on the left side of the Android Studio window. The Project pane appears.
2. To view the project in the standard Android project hierarchy, choose **Android** from the popup menu at the top of the Project pane, as shown below.

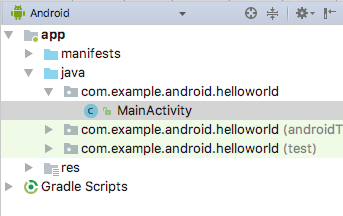


**Note:** This chapter and other chapters refer to the Project pane, when set to **Android**, as the **Project > Android** pane.

* 1. Explore the app and res folders

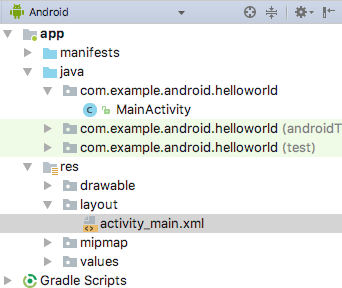
All code and resources for the app are located within the app and res folders.

1. Expand the **app** folder, the **java** folder, and the **com.example.android.helloworld** folder to see the **MainActivity** java file. Double-clicking the file opens it in the code editor.



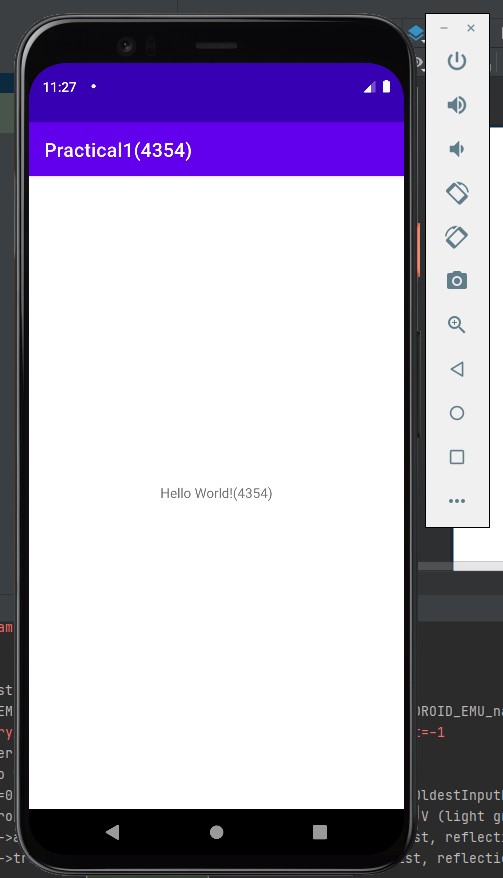
The **java** folder includes Java class files in three subfolders, as shown in the figure above. The **com.example.hello.helloworld** (or the domain name you have specified) folder contains all the files for an app package. The other two folders are used for testing and described in another lesson. For the Hello World app, there is only one package and it contains MainActivity.java. The name of the first Activity (screen) the user sees, which also initializes app-wide resources, is customarily called **MainActivity** (the file extension is omitted in the **Project > Android** pane).

1. Expand the **res** folder and the **layout** folder, and double-click the **activity\_main.xml** file to open it in the layout editor.



The **res** folder holds resources, such as layouts, strings, and images. An Activity is usually associated with a layout of UI views defined as an XML file. This file is usually named after its Activity.

**Output –**



# Practical No – 2

### **Exp #2:** Create an android app with Interactive User Interface using Relative Layouts. (e.g. simple calculator)

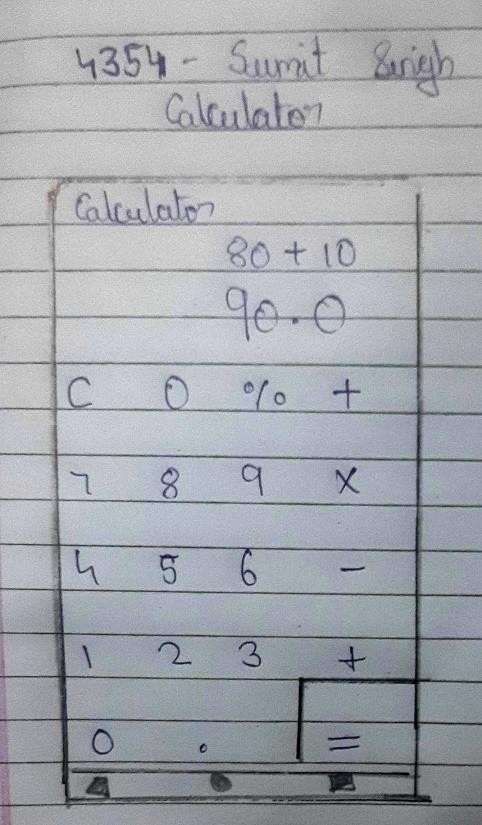
**Thoery:** RelativeLayout is a view group that displays child views in relative positions. The position of each view can be specified as relative to sibling elements (such as to the left-of or below another view) or in positions relative to the parent RelativeLayout area (such as aligned to the bottom, left or center).

The user interface (UI) for an Android app is built as a hierarchy of layouts and widgets. The layouts are ViewGroup objects, containers that control how their child views are positioned on the screen. Widgets are View objects, UI components such as buttons and text boxes.

Layout used:

* 1. LinearLayout
  2. RelativeLayout

**Diagrammatic representation:**



**Classes and Methods Used:**

"EditText e1=(EditText )findViewById(R.id.num1);"

Here num1 is id for textbox and we are just giving a variable name ‘e1’ to text box with id ‘num1’. Similarly, we have to use the same statement for the second textbox with the variable name ‘e2’. For the third text box, we have used

"TextView t1=(TextView) findViewById(R.id.result);"

Here we have used TextView because we only have to display text avoiding it being user-changeable. Now we have to input numbers in form of string using the getText() function. The input statement will be

"String s11=e1.getText().toString();"

Here s11 stores the number entered in textbox 1.We have to do the same with another Textbox(e2). Now store the number in int form and apply addition. store the added value in another variable. To display stored in sum we have to use setText() as follows:

result.setText(final\_sum.toString())

final\_sum stores the sum and it’s necessary to convert it to string(.toString()). Below is the code for the MainActivity.java file. Comments are added inside the code to understand the code in more detail.

**activity\_main.xml**

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

<LinearLayout xmlns:android=["http://schemas.android.com/](http://schemas.android.com/apk/res/android)a[pk/res/android"](http://schemas.android.com/apk/res/android) xmlns:t[ools="ht](http://schemas.android.com/tools)tp:/[/sc](http://schemas.android.com/tools)he[mas.android.com/tools"](http://schemas.android.com/tools) android:layout\_width="match\_parent" android:layout\_height="match\_parent"

android:orientation="vertical" tools:context=".MainActivity">

<include

layout="@layout/input\_layout" android:layout\_width="match\_parent" android:layout\_height="wrap\_content"/>

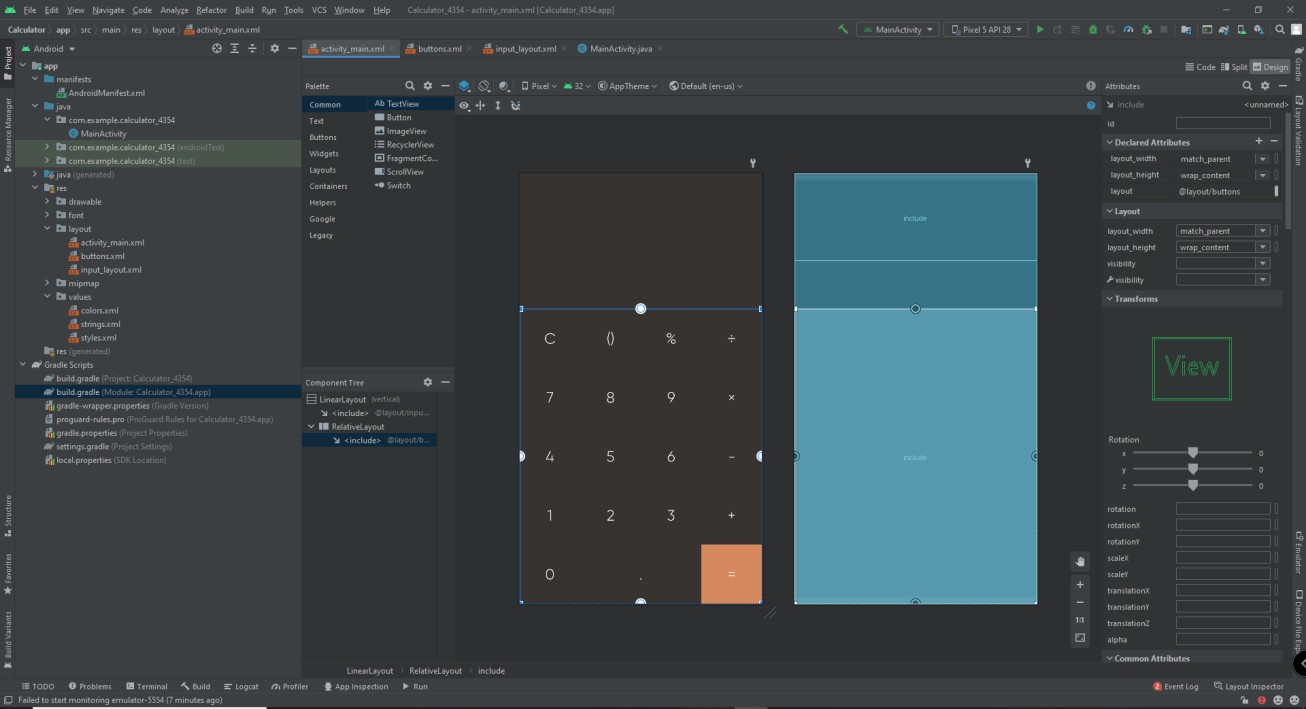
<RelativeLayout android:layout\_width="match\_parent" android:gravity="bottom" android:background="@color/black" android:layout\_height="match\_parent">

<include layout="@layout/buttons"

android:layout\_width="match\_parent" android:layout\_height="wrap\_content" />

</RelativeLayout>

</LinearLayout>



## buttons.xml

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

<RelativeLayout xmlns:android=["http://schemas.android.com/](http://schemas.android.com/apk/res/android)a[pk/res/android"](http://schemas.android.com/apk/res/android) android:layout\_width="match\_parent" android:layout\_height="wrap\_content">

<GridLayout android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:rowCount="5" android:columnCount="6">

<Button

android:id="@+id/btnClear" android:layout\_width="wrap\_content" android:layout\_height="100dp"

android:text="C" android:fontFamily="@font/g\_light" android:textSize="25sp" android:layout\_columnWeight="1" android:textColor="@color/white" android:background="@color/black" android:layout\_row="0" android:layout\_column="0"/>

<Button

android:id="@+id/btnBracket" android:layout\_width="wrap\_content" android:layout\_height="100dp" android:text="()" android:fontFamily="@font/g\_light" android:textSize="25sp" android:layout\_row="0" android:layout\_column="1" android:layout\_columnWeight="1" android:background="@color/black" android:textColor="@color/white"/>

<Button

android:id="@+id/btnPercent" android:layout\_width="wrap\_content" android:layout\_height="100dp" android:text="%" android:textSize="25sp" android:fontFamily="@font/g\_light" android:layout\_row="0"

android:layout\_column="2" android:layout\_columnWeight="1" android:background="@color/black" android:textColor="@color/white"/>

<Button

android:id="@+id/btnDivision" android:layout\_width="wrap\_content" android:layout\_height="100dp" android:text="÷" android:textSize="25sp" android:fontFamily="@font/g\_light" android:layout\_row="0" android:layout\_column="3" android:layout\_columnWeight="1" android:background="@color/black" android:textColor="@color/white"/>

<Button

android:id="@+id/btn7" android:layout\_width="wrap\_content" android:layout\_height="100dp" android:text="7" android:fontFamily="@font/g\_light" android:textSize="25sp" android:layout\_columnWeight="1" android:textColor="@color/white" android:background="@color/black" android:layout\_row="1" android:layout\_column="0"/>

<Button

android:id="@+id/btn8" android:layout\_width="wrap\_content" android:layout\_height="100dp" android:text="8" android:fontFamily="@font/g\_light" android:textSize="25sp" android:layout\_row="1" android:layout\_column="1" android:layout\_columnWeight="1" android:background="@color/black" android:textColor="@color/white"/>

<Button

android:id="@+id/btn9" android:layout\_width="wrap\_content" android:layout\_height="100dp" android:text="9" android:textSize="25sp" android:layout\_row="1" android:fontFamily="@font/g\_light" android:layout\_column="2" android:layout\_columnWeight="1" android:background="@color/black" android:textColor="@color/white"/>

<Button

android:id="@+id/btnMultiply" android:layout\_width="wrap\_content" android:layout\_height="100dp" android:text="×" android:textSize="25sp"

android:fontFamily="@font/g\_light" android:layout\_row="1" android:layout\_column="3" android:layout\_columnWeight="1" android:background="@color/black" android:textColor="@color/white"/>

<Button

android:id="@+id/btn4" android:layout\_width="wrap\_content" android:layout\_height="100dp" android:text="4" android:textSize="25sp" android:fontFamily="@font/g\_light" android:layout\_columnWeight="1" android:textColor="@color/white" android:background="@color/black" android:layout\_row="2" android:layout\_column="0"/>

<Button

android:id="@+id/btn5" android:layout\_width="wrap\_content" android:layout\_height="100dp" android:text="5" android:textSize="25sp" android:layout\_row="2" android:fontFamily="@font/g\_light" android:layout\_column="1" android:layout\_columnWeight="1"

android:background="@color/black" android:textColor="@color/white"/>

<Button

android:id="@+id/btn6" android:layout\_width="wrap\_content" android:layout\_height="100dp" android:text="6" android:textSize="25sp" android:fontFamily="@font/g\_light" android:layout\_row="2" android:layout\_column="2" android:layout\_columnWeight="1" android:background="@color/black" android:textColor="@color/white"/>

<Button

android:id="@+id/btnMinus" android:layout\_width="wrap\_content" android:layout\_height="100dp" android:text="-" android:textSize="25sp" android:layout\_row="2" android:fontFamily="@font/g\_light" android:layout\_column="3" android:layout\_columnWeight="1" android:background="@color/black" android:textColor="@color/white"/>

<Button

android:id="@+id/btn1"

android:layout\_width="wrap\_content" android:layout\_height="100dp" android:text="1" android:textSize="25sp" android:layout\_columnWeight="1" android:fontFamily="@font/g\_light" android:textColor="@color/white" android:background="@color/black" android:layout\_row="3" android:layout\_column="0"/>

<Button

android:id="@+id/btn2" android:layout\_width="wrap\_content" android:layout\_height="100dp" android:text="2" android:textSize="25sp" android:layout\_row="3" android:fontFamily="@font/g\_light" android:layout\_column="1" android:layout\_columnWeight="1" android:background="@color/black" android:textColor="@color/white"/>

<Button

android:id="@+id/btn3" android:layout\_width="wrap\_content" android:layout\_height="100dp" android:text="3" android:textSize="25sp" android:fontFamily="@font/g\_light" android:layout\_row="3"

android:layout\_column="2" android:layout\_columnWeight="1" android:background="@color/black" android:textColor="@color/white"/>

<Button

android:id="@+id/btnPlus" android:layout\_width="wrap\_content" android:layout\_height="100dp" android:text="+" android:fontFamily="@font/g\_light" android:textSize="25sp" android:layout\_row="3" android:layout\_column="3" android:layout\_columnWeight="1" android:background="@color/black" android:textColor="@color/white"/>

<Button

android:id="@+id/btn0" android:layout\_width="wrap\_content" android:layout\_height="100dp" android:text="0" android:fontFamily="@font/g\_light" android:textSize="25sp" android:layout\_columnWeight="1" android:textColor="@color/white" android:background="@color/black" android:layout\_row="4" android:layout\_column="0"/>

<Button

android:id="@+id/btnDot" android:layout\_width="wrap\_content" android:layout\_height="100dp" android:text="." android:fontFamily="@font/g\_light" android:layout\_columnSpan="2" android:textSize="25sp" android:layout\_row="4" android:layout\_column="1" android:layout\_columnWeight="1" android:background="@color/black" android:textColor="@color/white"/>

<Button

android:id="@+id/btnEqual" android:layout\_width="wrap\_content" android:layout\_height="100dp" android:text="=" android:fontFamily="@font/g\_light" android:textSize="25sp" android:layout\_row="4" android:layout\_column="3" android:layout\_columnWeight="1" android:background="@color/orange" android:textColor="@color/white"/>

</GridLayout>

</RelativeLayout>

## input\_layout.xml:

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

<LinearLayout xmlns:android=["http://schemas.android.com/](http://schemas.android.com/apk/res/android)a[pk/res/android"](http://schemas.android.com/apk/res/android) android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:background="@color/black"

android:orientation="vertical">

<TextView android:id="@+id/tvInput"

android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:textSize="30sp" android:text=""

android:gravity="end" android:fontFamily="@font/g\_light" android:background="@color/black" android:padding="10dp" android:textColor="@color/white"

/>

<TextView android:id="@+id/tvOutput"

android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:textSize="60sp" android:text=""

android:gravity="end" android:background="@color/black"

android:padding="10dp" android:fontFamily="@font/g\_light" android:textColor="@color/white"

/>

</LinearLayout>

## MainActivity.java:

package com.example.calculator\_4354;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle; import android.view.View; import android.widget.Button;

import android.widget.TextView;

import org.mozilla.javascript.Context; import org.mozilla.javascript.Scriptable;

public class MainActivity extends AppCompatActivity {

Button btn0, btn1, btn2, btn3, btn4, btn5, btn6, btn7, btn8, btn9, btnPercent, btnPlus, btnMinus, btnMultiply, btnDivision, btnEqual, btnClear, btnDot, btnBracket;

TextView tvInput, tvOutput; String process;

boolean checkBracket = false;

@Override

protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

btn0 = findViewById(R.id.btn0); btn1 = findViewById(R.id.btn1); btn2 = findViewById(R.id.btn2); btn3 = findViewById(R.id.btn3); btn4 = findViewById(R.id.btn4); btn5 = findViewById(R.id.btn5); btn6 = findViewById(R.id.btn6); btn7 = findViewById(R.id.btn7); btn8 = findViewById(R.id.btn8); btn9 = findViewById(R.id.btn9);

btnPlus = findViewById(R.id.btnPlus); btnMinus = findViewById(R.id.btnMinus); btnDivision = findViewById(R.id.btnDivision); btnMultiply = findViewById(R.id.btnMultiply);

btnEqual = findViewById(R.id.btnEqual);

btnClear = findViewById(R.id.btnClear); btnDot = findViewById(R.id.btnDot); btnPercent = findViewById(R.id.btnPercent); btnBracket = findViewById(R.id.btnBracket);

tvInput = findViewById(R.id.tvInput); tvOutput = findViewById(R.id.tvOutput);

btnClear.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) { tvInput.setText(""); tvOutput.setText("");

}

});

btn0.setOnClickListener(new View.OnClickListener() { @Override

public void onClick(View v) {

process = tvInput.getText().toString(); tvInput.setText(process + "0");

}

});

btn1.setOnClickListener(new View.OnClickListener() { @Override

public void onClick(View v) {

process = tvInput.getText().toString(); tvInput.setText(process + "1");

}

});

btn2.setOnClickListener(new View.OnClickListener() { @Override

public void onClick(View v) {

process = tvInput.getText().toString(); tvInput.setText(process + "2");

}

});

btn3.setOnClickListener(new View.OnClickListener() { @Override

public void onClick(View v) {

process = tvInput.getText().toString(); tvInput.setText(process + "3");

}

});

btn4.setOnClickListener(new View.OnClickListener() { @Override

public void onClick(View v) {

process = tvInput.getText().toString(); tvInput.setText(process + "4");

}

});

btn5.setOnClickListener(new View.OnClickListener() { @Override

public void onClick(View v) {

process = tvInput.getText().toString(); tvInput.setText(process + "5");

}

});

btn6.setOnClickListener(new View.OnClickListener() { @Override

public void onClick(View v) {

process = tvInput.getText().toString(); tvInput.setText(process + "6");

}

});

btn6.setOnClickListener(new View.OnClickListener() { @Override

public void onClick(View v) {

process = tvInput.getText().toString(); tvInput.setText(process + "6");

}

});

btn7.setOnClickListener(new View.OnClickListener() { @Override

public void onClick(View v) {

process = tvInput.getText().toString(); tvInput.setText(process + "7");

}

});

btn8.setOnClickListener(new View.OnClickListener() { @Override

public void onClick(View v) {

process = tvInput.getText().toString(); tvInput.setText(process + "8");

}

});

btn9.setOnClickListener(new View.OnClickListener() { @Override

public void onClick(View v) {

}

});

process = tvInput.getText().toString(); tvInput.setText(process + "9");

btnPlus.setOnClickListener(new View.OnClickListener() { @Override

public void onClick(View v) {

process = tvInput.getText().toString(); tvInput.setText(process + "+");

}

});

btnMinus.setOnClickListener(new View.OnClickListener() { @Override

public void onClick(View v) {

process = tvInput.getText().toString(); tvInput.setText(process + "-");

}

});

btnMultiply.setOnClickListener(new View.OnClickListener() { @Override

public void onClick(View v) {

process = tvInput.getText().toString(); tvInput.setText(process + "×");

}

});

btnDivision.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

process = tvInput.getText().toString(); tvInput.setText(process + "÷");

}

});

btnDot.setOnClickListener(new View.OnClickListener() { @Override

public void onClick(View v) {

process = tvInput.getText().toString(); tvInput.setText(process + ".");

}

});

btnPercent.setOnClickListener(new View.OnClickListener() { @Override

public void onClick(View v) {

process = tvInput.getText().toString(); tvInput.setText(process + "%");

}

});

btnBracket.setOnClickListener(new View.OnClickListener() { @Override

public void onClick(View v) {

if (checkBracket) {

process = tvInput.getText().toString(); tvInput.setText(process + ")");

checkBracket = false;

} else {

process = tvInput.getText().toString(); tvInput.setText(process + "("); checkBracket = true;

}

}

});

btnEqual.setOnClickListener(new View.OnClickListener() { @Override

public void onClick(View v) {

process = tvInput.getText().toString();

process = process.replaceAll("×", "\*"); process = process.replaceAll("%", "/100"); process = process.replaceAll("÷", "/");

Context rhino = Context.enter();

rhino.setOptimizationLevel(-1);

String finalResult = "";

try {

Scriptable scriptable = rhino.initStandardObjects();

finalResult = rhino.evaluateString(scriptable, process, "javascript", 1, null).toString();

} catch (Exception e) { finalResult = "0";

}

tvOutput.setText(finalResult);

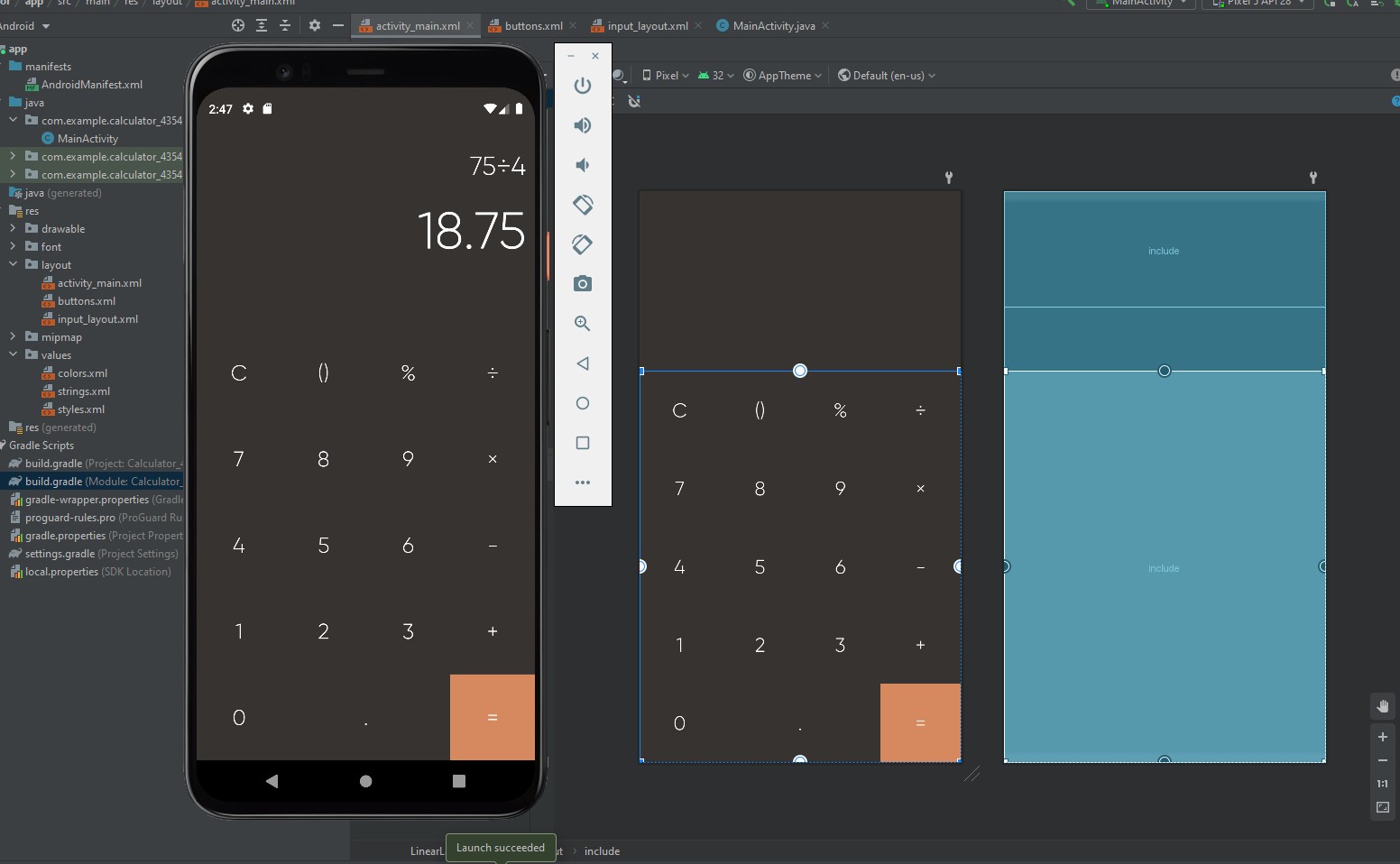
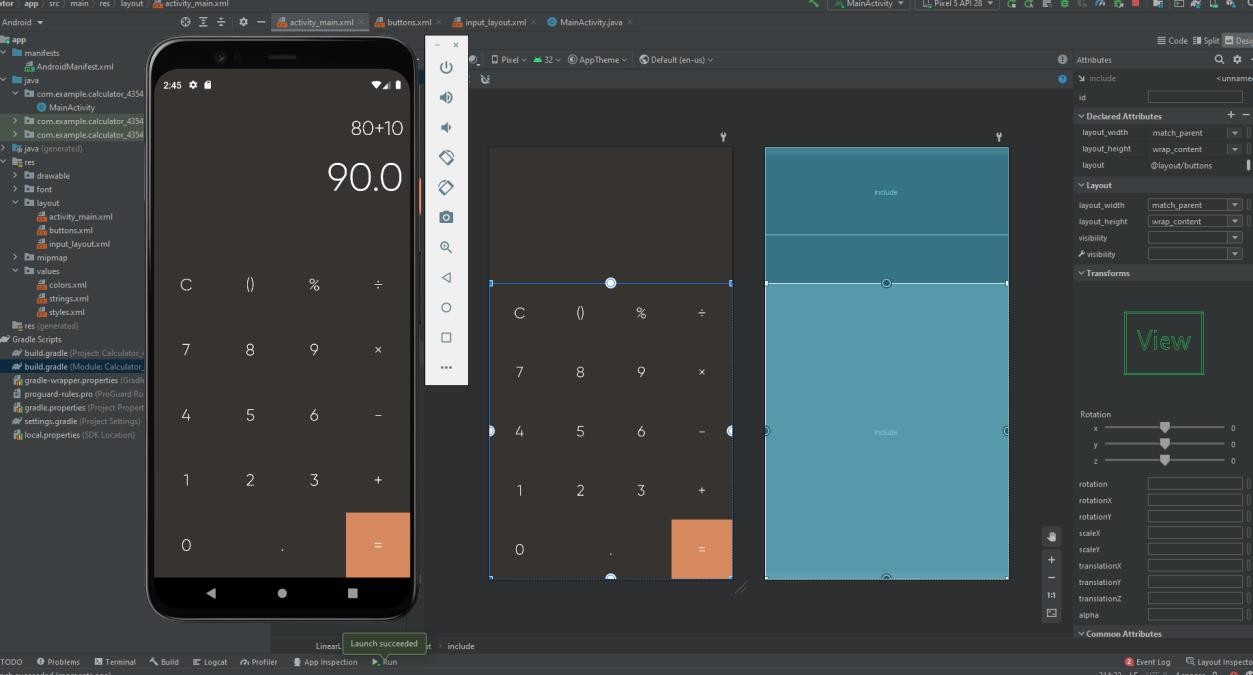
}

});

}

}

## Output:



**Conclusion:** Hence we have created an android app with Interactive User Interface using Relative Layout that is Calculator.

# Practical No- 3

**Experiment 3:** Create an android app that demonstrates working with TextView Elements

**Thoery:** A TextView displays text to the user and optionally allows them to edit it. A TextView is a complete text editor, however the basic class is configured to not allow editing.

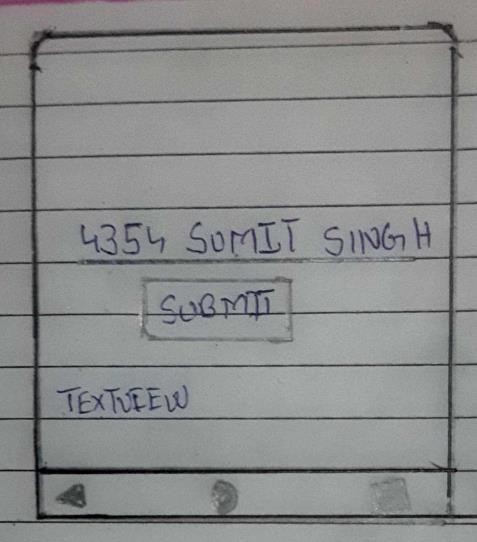
The following code sample shows a typical use, with an XML layout and code to modify the contents of the text view:

<LinearLayout xmlns:android="<http://schemas.android.com/apk/res/android>" android:layout\_width="match\_parent" android:layout\_height="match\_parent">

<TextView android:id="@+id/text\_view\_id" android:layout\_height="wrap\_content" android:layout\_width="wrap\_content" android:text="@string/hello" />

</LinearLayout>

**Diagrammatic representation:**



**Classes and Methods Used:**

public class TextView

extends View implements ViewTreeObserver.OnPreDrawListener

java.lang.Object

↳ android.view.View

↳ android.widget.TextView Nested classes

enum TextView.BufferType

Type of the text buffer that defines the characteristics of the text such as static, styleable, or editable.

interface TextView.OnEditorActionListener

Interface definition for a callback to be invoked when an action is performed on the editor.

class TextView.SavedState

User interface state that is stored by TextView for implementing View#onSaveInstanceState.

**activity\_main.xml:**

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

<RelativeLayout xmlns:android=["http://schemas.android.com/](http://schemas.android.com/apk/res/android)a[pk/res/android"](http://schemas.android.com/apk/res/android) xmlns:app=["http://sc](http://schemas.android.com/apk/res-auto)h[emas.android.com/apk/res-auto"](http://schemas.android.com/apk/res-auto) xmlns:t[ools="ht](http://schemas.android.com/tools)tp:/[/sc](http://schemas.android.com/tools)he[mas.android.com/tools"](http://schemas.android.com/tools) android:layout\_width="match\_parent" android:layout\_height="match\_parent"

android:gravity="center" android:padding="20dp" tools:context=".MainActivity">

<EditText android:id="@+id/editTextName" android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:ems="10" android:inputType="textPersonName" android:paddingStart="20dp" android:paddingTop="20dp" android:paddingEnd="20dp" android:paddingBottom="20dp" />

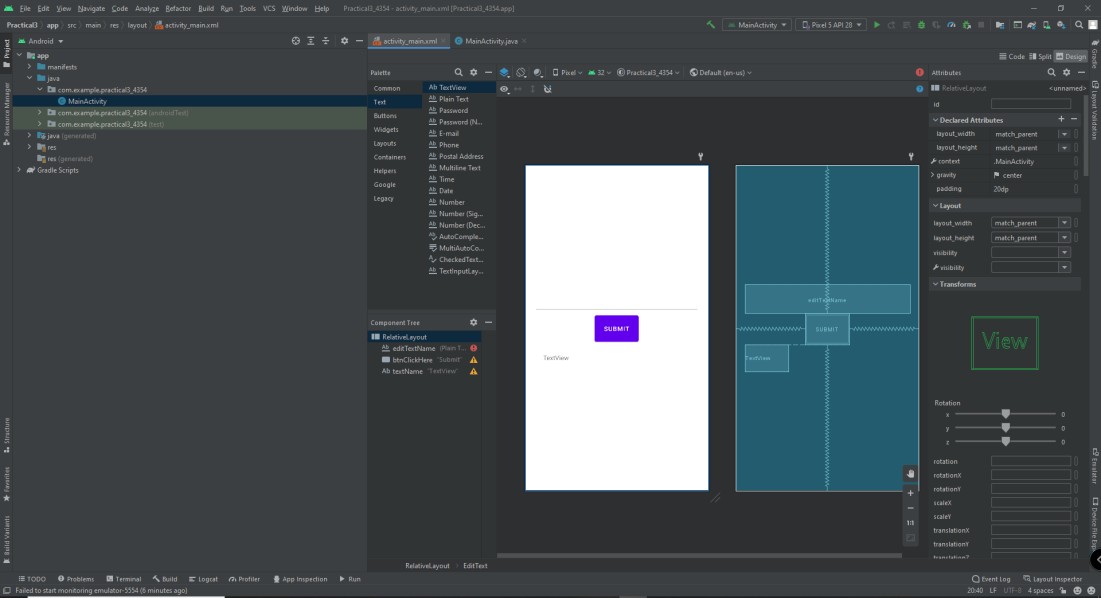
<Button

android:id="@+id/btnClickHere" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_below="@id/editTextName" android:layout\_centerInParent="true" android:paddingStart="20dp" android:paddingTop="20dp" android:paddingEnd="20dp" android:paddingBottom="20dp" android:text="Submit" />

<TextView android:id="@+id/textName" android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content" android:layout\_below="@id/btnClickHere" android:paddingStart="20dp" android:paddingTop="20dp" android:paddingEnd="20dp" android:paddingBottom="20dp" android:text="TextView" />

</RelativeLayout>



#### MainActivity.java:

package com.example.practical3\_4354;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle; import android.view.View; import android.widget.Button; import android.widget.EditText;

import android.widget.TextView;

public class MainActivity extends AppCompatActivity { EditText editTextName;

Button btnClickHere; TextView textName;

@Override

protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity\_main);

editTextName = (EditText) findViewById(R.id.editTextName); btnClickHere = (Button) findViewById(R.id.btnClickHere); textName = (TextView) findViewById(R.id.textName);

btnClickHere.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

String name = editTextName.getText().toString(); textName.setText("Hi " + name);

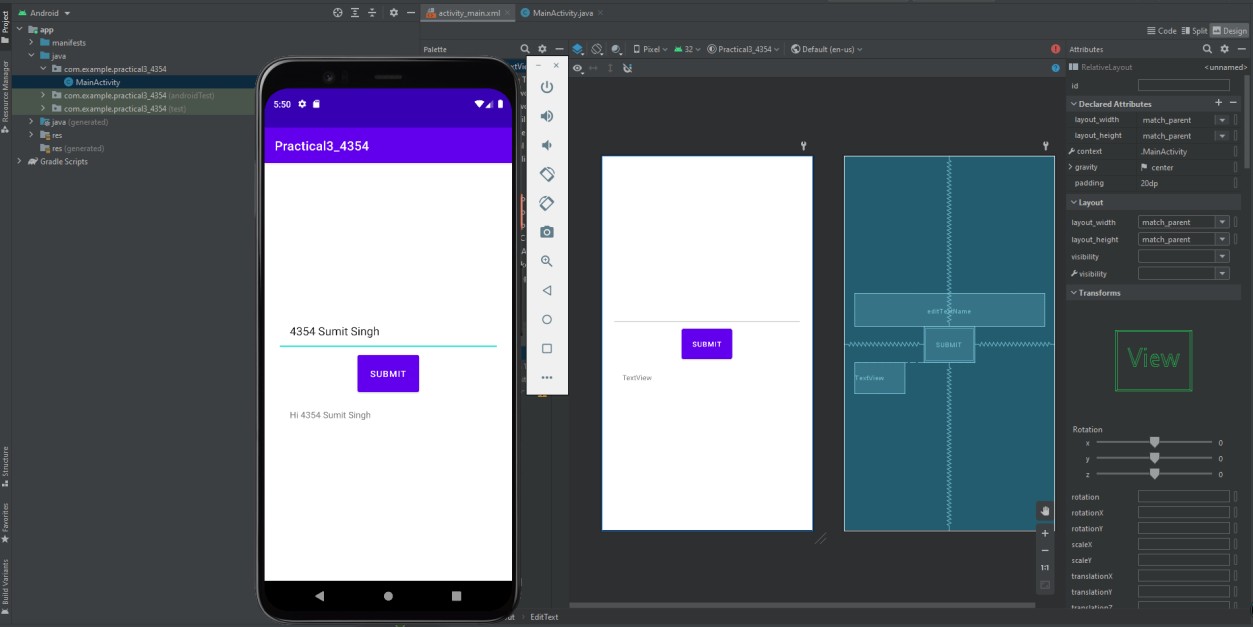
}

});

}

}

**Output:**



**Conclusion:** Hence we Created an android app that demonstrates working with TextView Elements

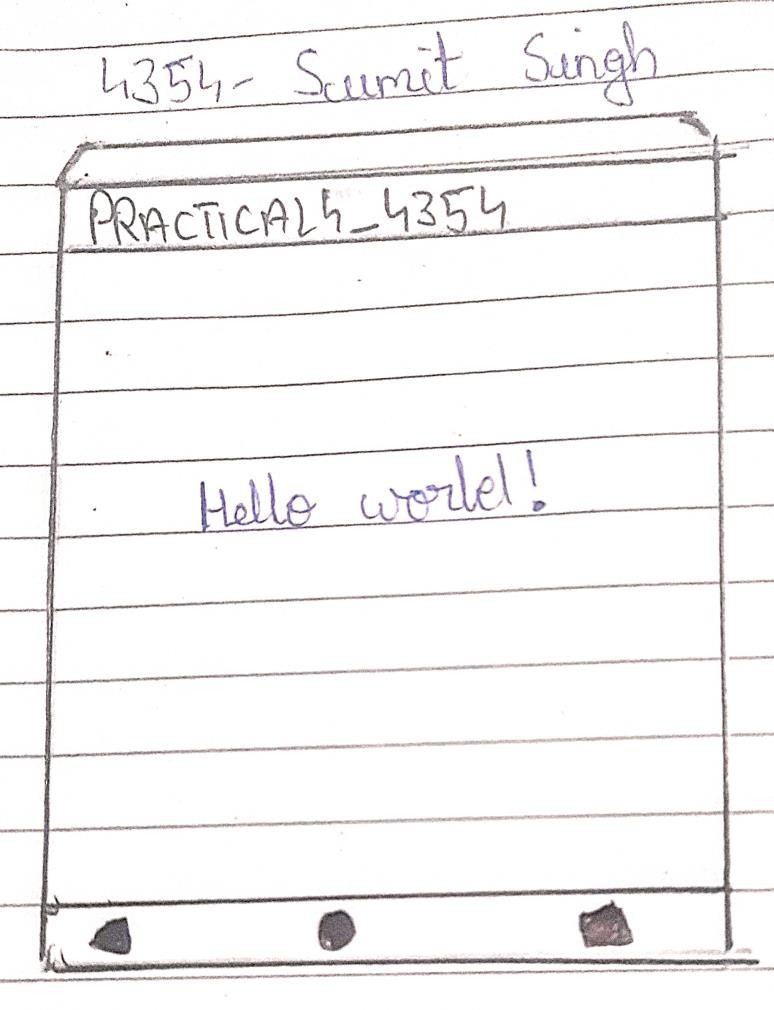
# Practical 4

##### **Experiment 4:** Create an android app that demonstrates Activity Lifecycle and Instance State.

**Theory of Activity Life Cycle:** In android, Activity represents a single screen with a user interface (UI) of an application and it will acts an entry point for users to interact with an app.

Generally, in android activity class uses different callback methods like onCreate(), onStart(), onPause(), onRestart(), onResume(), onStop() and onDestroy() to go through a different stages of activity life cycle.

**Diagrammatic representation:**



**Classes and Methods Used:**

To navigate transitions between stages of the activity lifecycle, the Activity class provides a core set of six callbacks: onCreate(), onStart(), onResume(), onPause(), onStop(), and onDestroy(). The system invokes each of these callbacks as an activity enters a new state.

**onCreate()**

You must implement this callback, which fires when the system first creates the activity. On activity creation, the activity enters the Created state. In the onCreate() method, you perform basic application startup logic that should happen only once for the entire life of the activity.

**onStart()**

When the activity enters the Started state, the system invokes this callback. The onStart() call makes the activity visible to the user, as the app prepares for the activity to enter the foreground and become interactive

**onResume()**

When the activity enters the Resumed state, it comes to the foreground, and then the system invokes the onResume() callback. This is the state in which the app interacts with the user.

**onPause()**

The system calls this method as the first indication that the user is leaving your activity (though it does not always mean the activity is being destroyed); it indicates that the activity is no longer in the foreground (though it may still be visible if the user is in multi-window mode).

**onStop()**

When your activity is no longer visible to the user, it has entered the Stopped state, and the system invokes the onStop() callback.

**activity\_main.xml**

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

<androidx.constraintlayout.widget.ConstraintLayout xmlns:andr[oid="htt](http://schemas.android.com/apk/res/android)p:/[/sc](http://schemas.android.com/apk/res/android)he[mas.android.com/apk/res/android"](http://schemas.android.com/apk/res/android)

xmlns:app=["http://sc](http://schemas.android.com/apk/res-auto)h[emas.android.com/apk/res-auto"](http://schemas.android.com/apk/res-auto)

xmlns:t[ools="ht](http://schemas.android.com/tools)tp:/[/sc](http://schemas.android.com/tools)he[mas.android.com/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\_constraintLeft\_toLeftOf="parent" app:layout\_constraintRight\_toRightOf="parent" app:layout\_constraintTop\_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

**MainActivity.java:**

package com.example.practical4\_4354;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle; import android.util.Log;

public class MainActivity extends AppCompatActivity { @Override

protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity\_main); Log.d("Activity Lifecycle", "onCreate invoked");

}

@Override

protected void onStart() { super.onStart();

Log.d("Activity Lifecycle", "onStart invoked");

}

@Override

protected void onResume() { super.onResume();

Log.d("Activity Lifecycle", "onResume invoked");

}

@Override

protected void onPause() { super.onPause();

Log.d("Activity Lifecycle", "onPause invoked");

}

@Override

protected void onStop() { super.onStop();

Log.d("Activity Lifecycle", "onStop invoked");

}

@Override

protected void onRestart() { super.onRestart();

Log.d("Activity Lifecycle", "onRestart invoked");

}

@Override

protected void onDestroy() { super.onDestroy();

Log.d("Activity Lifecycle", "onDestroy invoked");

}

}

## Output:

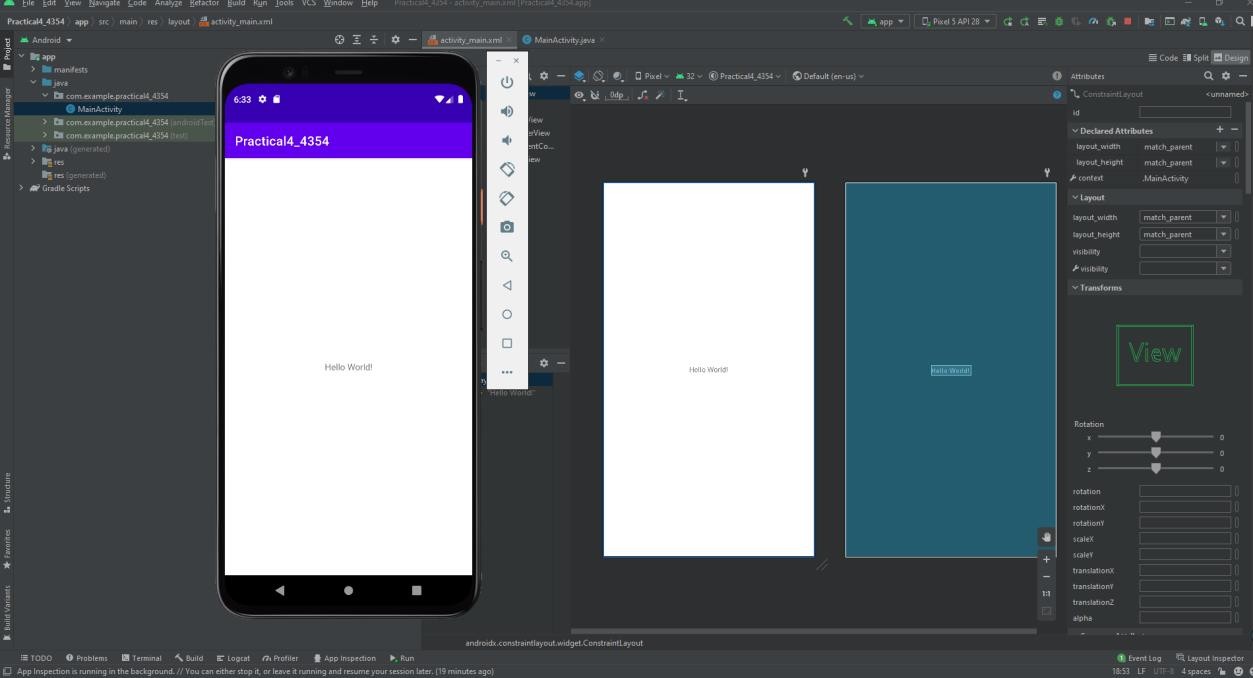
W/practical4\_435: Accessing hidden method Landroid/view/ViewGroup;-

>makeOptionalFitsSystemWindows()V (light greylist, reflection) D/Activity Lifecycle: onCreate invoked

D/Activity Lifecycle: onStart invoked D/Activity Lifecycle: onResume invoked D/Activity Lifecycle: onPause invoked

D/EGL\_emulation: eglMakeCurrent: 0xea218d80: ver 3 1 (tinfo 0xf0444ee0) D/Activity Lifecycle: onStop invoked

D/Activity Lifecycle: onDestroy invoked



**Instance State**

**Source Code:**

**activity\_main.xml:**

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

<LinearLayout xmlns:android=["http://schemas.android.com/](http://schemas.android.com/apk/res/android)a[pk/res/android"](http://schemas.android.com/apk/res/android) xmlns:app=["http://sc](http://schemas.android.com/apk/res-auto)h[emas.android.com/apk/res-auto"](http://schemas.android.com/apk/res-auto) xmlns:t[ools="ht](http://schemas.android.com/tools)tp:/[/sc](http://schemas.android.com/tools)he[mas.android.com/tools"](http://schemas.android.com/tools) android:layout\_width="match\_parent"

android:layout\_height="match\_parent" tools:context=".MainActivity">

<EditText android:id="@+id/editTextName" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_marginTop="30dp" android:textSize="40sp" />

<EditText android:id="@+id/editMobileNo" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_marginTop="30dp" android:textSize="40sp" />

</LinearLayout>

## MainActivity.java

package com.example.instancestate;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle; import android.view.View; import android.widget.EditText;

import android.widget.TextView; import org.w3c.dom.Text;

public class MainActivity extends AppCompatActivity { EditText edName, edMobile;

private String sname, smobile;

@Override

protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity\_main);

edName = (EditText) findViewById(R.id.editTextName); edMobile = (EditText) findViewById(R.id.editMobileNo);

}

@Override

public void onSaveInstanceState(Bundle savedInstanceState) { super.onSaveInstanceState(savedInstanceState);

sname = edName.getText().toString(); smobile = edMobile.getText().toString();

savedInstanceState.putString("MyName", sname); //key-value pair savedInstanceState.putString("MyMobile", smobile); //key-value pair

}

@Override

protected void onRestoreInstanceState(Bundle savedInstanceState) { super.onRestoreInstanceState(savedInstanceState);

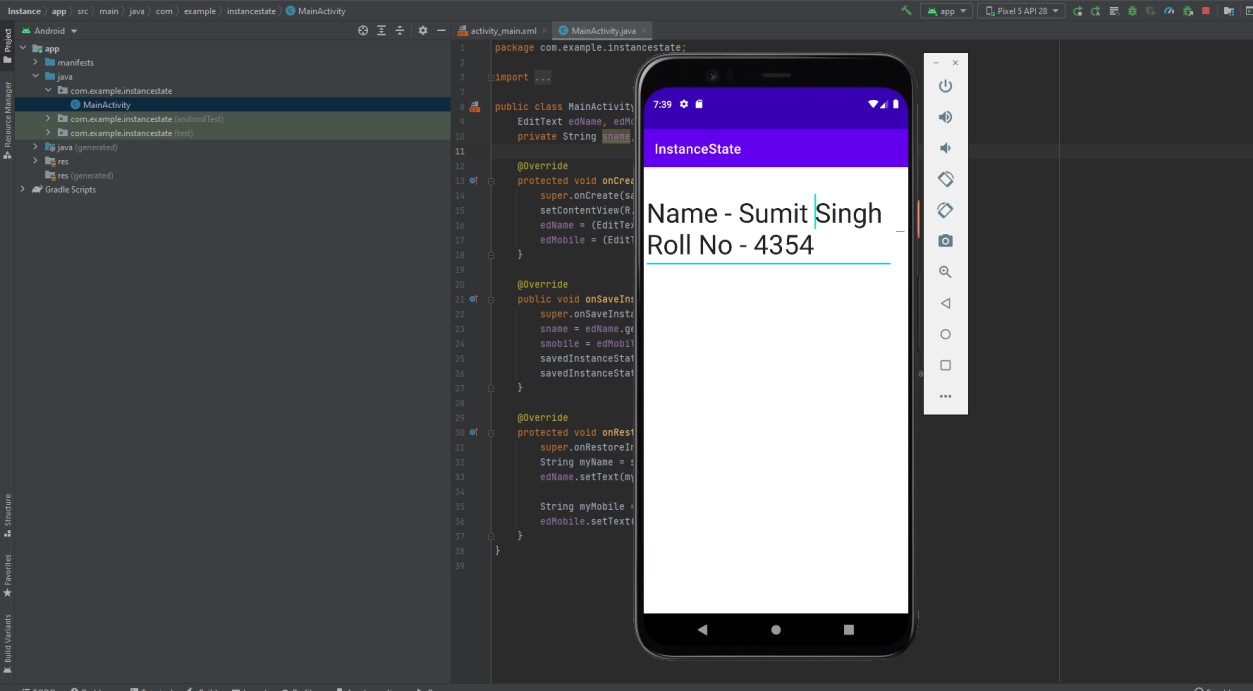
String myName = savedInstanceState.getString("MyName"); edName.setText(myName);

String myMobile = savedInstanceState.getString("MyMobile"); edMobile.setText(myMobile);

}

}

## Output:



**Conclusion:** Hence we have Created an android app that demonstrates Activity Lifecycle and Instance State.

# Practical No - 5

### **Exp 5:** Create an android app that demonstrates the use of Keyboards, Input Controls, Alerts, and Pickers. (eg.

DroidCafe)

**Thoery:** A new app starting with the Basic Activity template that imitates a dessert-ordering app. The user can tap an image to perform an action—in this case display a Toast message—as shown in the figure below. The user can also tap a shopping-cart button to proceed to the next Activity.

The user interface (UI) that appears on a screen of an Android-powered device consists of a hierarchy of objects called views. Every element of the screen is a View.

The View class represents the basic building block for all UI components. View is the base class for classes that provide interactive UI components, such as Button elements. A Button is a UI element the user can tap or click to perform an action.

You can turn any View, such as an ImageView, into a UI element that can be tapped or clicked. You must store the image for the ImageView in the drawables folder of your project.

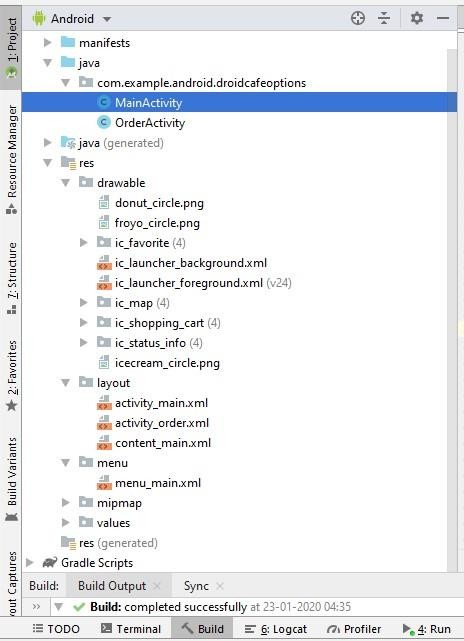
**Classes and Methods Used:**

Extract the string resource for the android:hintattribute value to enter\_phone\_hint. The following attributes should be set for the new EditText (add the layout\_marginLeft attribute for compatibility with older versions of Android):

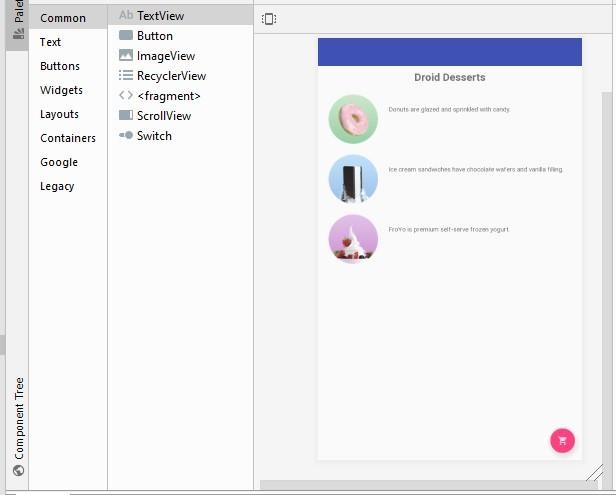
**textEmailAddress:** Tapping the field brings up the email keyboard with the @ symbol located near the space key.

**textPassword:** The characters the user enters turn into dots to conceal the entered password.

**DroidCafeOptions App**



**activity\_main.xml**



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

**<android.support.design.widget.CoordinatorLayout xmlns:android**[**="http:**](http://schemas.android.com/apk/res/android)**//**[**schemas.android.com/apk/res/android**](http://schemas.android.com/apk/res/android)**" xmlns:app**[**="http:**](http://schemas.android.com/apk/res-auto)**//**[**schemas.android.com/apk/res-auto**](http://schemas.android.com/apk/res-auto)**" xmlns:tools**[**="http:**](http://schemas.android.com/tools)**//**[**schemas.android.com/tools**](http://schemas.android.com/tools)**" android:layout\_width="match\_parent" android:layout\_height="match\_parent"**

**tools:context="com.example.android.droidcafeoptions.MainActivity">**

**<android.support.design.widget.AppBarLayout android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:theme="@style/AppTheme.AppBarOverlay">**

**<android.support.v7.widget.Toolbar android:id="@+id/toolbar" android:layout\_width="match\_parent" android:layout\_height="?attr/actionBarSize" android:background="?attr/colorPrimary" app:popupTheme="@style/AppTheme.PopupOverlay" />**

**</android.support.design.widget.AppBarLayout>**

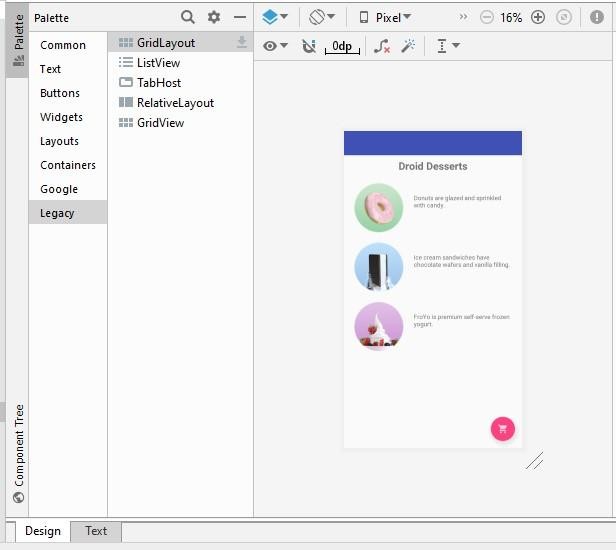
**<include layout="@layout/content\_main" />**

**<android.support.design.widget.FloatingActionButton android:id="@+id/fab" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_gravity="bottom|end" android:layout\_margin="16dp"**

**app:srcCompat="@drawable/ic\_shopping\_cart" />**

**</android.support.design.widget.CoordinatorLayout>**

###### content\_main.xml



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

**<android.support.constraint.ConstraintLayout xmlns:android**[**="http:**](http://schemas.android.com/apk/res/android)**//**[**schemas.android.com/apk/res/android**](http://schemas.android.com/apk/res/android)**"**

**xmlns:app**[**="http:**](http://schemas.android.com/apk/res-auto)**//**[**schemas.android.com/apk/res-auto**](http://schemas.android.com/apk/res-auto)**" xmlns:tools**[**="http:**](http://schemas.android.com/tools)**//**[**schemas.android.com/tools**](http://schemas.android.com/tools)**" android:layout\_width="match\_parent" android:layout\_height="match\_parent"**

**app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior"**

**tools:context="com.example.android.droidcafeoptions.MainActivity" tools:showIn="@layout/activity\_main">**

**<TextView**

**android:id="@+id/textintro" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_marginTop="8dp" android:text="@string/intro\_text" android:textSize="24sp" android:textStyle="bold"**

**app:layout\_constraintLeft\_toLeftOf="parent" app:layout\_constraintRight\_toRightOf="parent" app:layout\_constraintTop\_toTopOf="parent" />**

**<ImageView**

**android:id="@+id/donut" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_marginLeft="24dp" android:layout\_marginStart="24dp" android:layout\_marginTop="24dp" android:contentDescription="@string/donuts" android:onClick="showDonutOrder" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toBottomOf="@+id/textintro" app:srcCompat="@drawable/donut\_circle" />**

**<ImageView**

**android:id="@+id/ice\_cream" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_marginLeft="24dp" android:layout\_marginStart="24dp" android:layout\_marginTop="24dp"**

**android:contentDescription="@string/ice\_cream\_sandwiches" android:onClick="showIceCreamOrder" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toBottomOf="@+id/donut" app:srcCompat="@drawable/icecream\_circle" />**

**<ImageView**

**android:id="@+id/froyo" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_marginLeft="24dp" android:layout\_marginStart="24dp" android:layout\_marginTop="24dp" android:contentDescription="@string/froyo" android:onClick="showFroyoOrder" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toBottomOf="@+id/ice\_cream" app:srcCompat="@drawable/froyo\_circle" />**

**<TextView**

**android:id="@+id/donut\_description" android:layout\_width="0dp" android:layout\_height="wrap\_content" android:layout\_marginEnd="24dp" android:layout\_marginStart="24dp" android:layout\_marginTop="24dp" android:text="@string/donuts" app:layout\_constraintEnd\_toEndOf="parent"**

**app:layout\_constraintStart\_toEndOf="@+id/donut" app:layout\_constraintTop\_toTopOf="@+id/donut" />**

**<TextView**

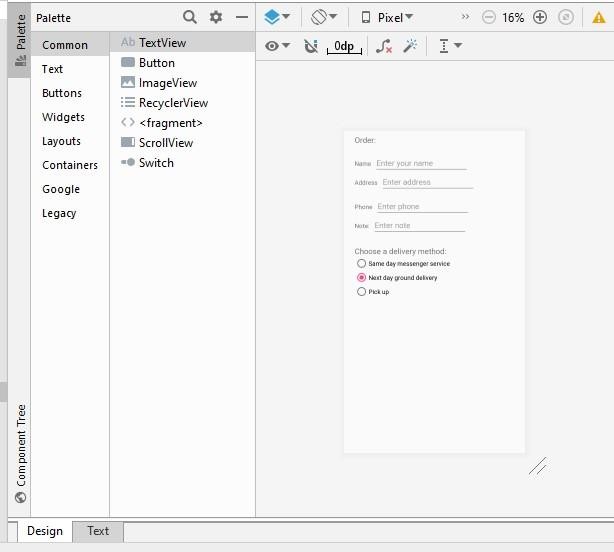
**android:id="@+id/ice\_cream\_description" android:layout\_width="0dp" android:layout\_height="wrap\_content" android:layout\_marginEnd="24dp" android:layout\_marginStart="24dp" android:layout\_marginTop="24dp" android:text="@string/ice\_cream\_sandwiches" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintStart\_toEndOf="@+id/ice\_cream" app:layout\_constraintTop\_toTopOf="@+id/ice\_cream" />**

**<TextView**

**android:id="@+id/froyo\_description" android:layout\_width="0dp" android:layout\_height="wrap\_content" android:layout\_marginEnd="24dp" android:layout\_marginStart="24dp" android:layout\_marginTop="24dp" android:text="@string/froyo" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintStart\_toEndOf="@+id/froyo" app:layout\_constraintTop\_toTopOf="@+id/froyo" />**

**</android.support.constraint.ConstraintLayout>**

###### activity\_order.xml



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

**<android.support.constraint.ConstraintLayout xmlns:android**[**="http:**](http://schemas.android.com/apk/res/android)**//**[**schemas.android.com/apk/res/android**](http://schemas.android.com/apk/res/android)**"**

**xmlns:app**[**="http:**](http://schemas.android.com/apk/res-auto)**//**[**schemas.android.com/apk/res-auto**](http://schemas.android.com/apk/res-auto)**" xmlns:tools**[**="http:**](http://schemas.android.com/tools)**//**[**schemas.android.com/tools**](http://schemas.android.com/tools)**" android:layout\_width="match\_parent" android:layout\_height="match\_parent"**

**tools:context="com.example.android.droidcafeoptions.OrderActivity">**

**<TextView**

**android:id="@+id/order\_textview" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_marginLeft="24dp" android:layout\_marginStart="24dp" android:layout\_marginTop="8dp" android:text="@string/order\_label\_text" android:textSize="18sp"**

**app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toTopOf="parent" />**

**<TextView**

**android:id="@+id/name\_label" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_marginLeft="24dp" android:layout\_marginStart="24dp" android:layout\_marginTop="32dp"**

**android:text="@string/name\_label\_text" app:layout\_constraintStart\_toStartOf="parent"**

**app:layout\_constraintTop\_toBottomOf="@+id/order\_textview" />**

**<EditText**

**android:id="@+id/name\_text" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_marginLeft="8dp" android:layout\_marginStart="8dp" android:ems="10"**

**android:hint="@string/enter\_name\_hint" android:inputType="textPersonName" app:layout\_constraintBaseline\_toBaselineOf="@+id/name\_label"**

**app:layout\_constraintStart\_toEndOf="@+id/name\_label" />**

**<TextView**

**android:id="@+id/address\_label" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_marginLeft="24dp" android:layout\_marginStart="24dp" android:layout\_marginTop="24dp" android:text="@string/address\_label\_text" app:layout\_constraintStart\_toStartOf="parent"**

**app:layout\_constraintTop\_toBottomOf="@+id/name\_label" />**

**<EditText**

**android:id="@+id/address\_text" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_marginLeft="8dp" android:layout\_marginStart="8dp" android:ems="10"**

**android:hint="@string/enter\_address\_hint" android:inputType="textMultiLine"**

**app:layout\_constraintBaseline\_toBaselineOf="@+id/address\_label" app:layout\_constraintStart\_toEndOf="@+id/address\_label" />**

**<TextView**

**android:id="@+id/phone\_label" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_marginLeft="24dp" android:layout\_marginStart="24dp" android:layout\_marginTop="24dp" android:text="@string/phone\_label\_string" app:layout\_constraintStart\_toStartOf="parent"**

**app:layout\_constraintTop\_toBottomOf="@+id/address\_text" />**

**<EditText**

**android:id="@+id/phone\_text" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_marginLeft="8dp" android:layout\_marginStart="8dp" android:ems="10"**

**android:hint="@string/enter\_phone\_hint" android:inputType="phone"**

**app:layout\_constraintBaseline\_toBaselineOf="@+id/phone\_label" app:layout\_constraintStart\_toEndOf="@+id/phone\_label" />**

**<TextView**

**android:id="@+id/note\_label" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_marginLeft="24dp"**

**android:layout\_marginStart="24dp" android:layout\_marginTop="24dp" android:text="@string/note\_label\_text" app:layout\_constraintStart\_toStartOf="parent"**

**app:layout\_constraintTop\_toBottomOf="@+id/phone\_label" />**

**<EditText**

**android:id="@+id/note\_text" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_marginLeft="8dp" android:layout\_marginStart="8dp" android:ems="10"**

**android:hint="@string/enter\_note\_hint" android:inputType="textCapSentences|textMultiLine" app:layout\_constraintBaseline\_toBaselineOf="@+id/note\_label" app:layout\_constraintStart\_toEndOf="@+id/note\_label" />**

**<TextView**

**android:id="@+id/delivery\_label" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_marginLeft="24dp" android:layout\_marginStart="24dp" android:layout\_marginTop="24dp" android:text="@string/choose\_delivery\_method" android:textSize="18sp"**

**app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toBottomOf="@+id/note\_text" />**

**<RadioGroup**

**android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_marginLeft="24dp" android:layout\_marginStart="24dp" android:orientation="vertical"**

**app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toBottomOf="@id/delivery\_label">**

**<RadioButton**

**android:id="@+id/sameday" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:onClick="onRadioButtonClicked"**

**android:text="@string/same\_day\_messenger\_service" />**

**<RadioButton**

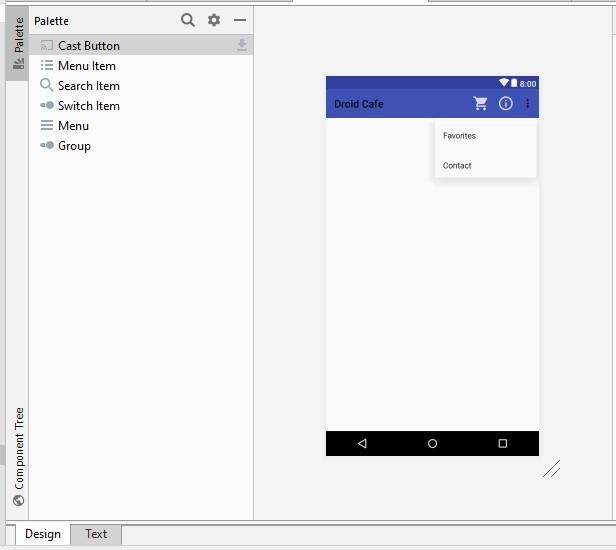
**android:id="@+id/nextday" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:onClick="onRadioButtonClicked" android:text="@string/next\_day\_ground\_delivery" android:checked="true"/>**

**<RadioButton**

**android:id="@+id/pickup" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:onClick="onRadioButtonClicked" android:text="@string/pick\_up" />**

**</RadioGroup> </android.support.constraint.ConstraintLayout>**

###### menu\_main.xml



**<menu xmlns:android**[**="http:**](http://schemas.android.com/apk/res/android)**//**[**schemas.android.com/apk/res/android**](http://schemas.android.com/apk/res/android)**" xmlns:app**[**="http:**](http://schemas.android.com/apk/res-auto)**//**[**schemas.android.com/apk/res-auto**](http://schemas.android.com/apk/res-auto)**" xmlns:tools**[**="http:**](http://schemas.android.com/tools)**//**[**schemas.android.com/tools**](http://schemas.android.com/tools)**"**

**tools:context="com.example.android.droidcafeoptions.MainActivity">**

**<item**

**android:id="@+id/action\_contact" android:orderInCategory="100" android:title="@string/action\_contact" app:showAsAction="never" />**

**<item**

**android:id="@+id/action\_order" android:icon="@drawable/ic\_shopping\_cart" android:orderInCategory="10" android:title="@string/action\_order" app:showAsAction="always" />**

**<item**

**android:id="@+id/action\_status" android:icon="@drawable/ic\_status\_info" android:orderInCategory="20" android:title="@string/action\_status" app:showAsAction="always" />**

**<item**

**android:id="@+id/action\_favorites" android:icon="@drawable/ic\_favorite" android:orderInCategory="30" android:title="@string/action\_favorites" app:showAsAction="ifRoom" />**

**</menu>**

###### OrderActivity.java

**package com.example.android.droidcafeoptions;**

**import android.content.Intent; import android.os.Bundle;**

**import android.support.v7.app.AppCompatActivity; import android.view.View; import android.widget.RadioButton; import android.widget.TextView; import android.widget.Toast;**

**/\*\***

* **This activity handles radio buttons for choosing a delivery method for an**
* **order, and EditText input controls.**

**\*/ public class OrderActivity extends AppCompatActivity {**

**/\*\***

* **Sets the content view to activity\_order, and gets the intent and its \* data.**

**\***

* **@param savedInstanceState Saved instance state bundle.**

**\*/ @Override protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity\_order);**

**// Get the intent and its data.**

**Intent intent = getIntent(); String message = intent.getStringExtra(MainActivity.EXTRA\_MESSAGE);**

**TextView textView = findViewById(R.id.order\_textview); textView.setText(message);**

**}**

**/\*\***

* **Checks which radio button was clicked and displays a toast message to**
* **show the choice.**

**\***

* **@param view The radio button view.**

**\*/**

**public void onRadioButtonClicked(View view) { // Is the button now checked? boolean checked = ((RadioButton) view).isChecked(); // Check which radio button was clicked. switch (view.getId()) {**

**case R.id.sameday: if (checked)**

**// Same day service**

**displayToast(getString(R.string.same\_day\_messenger\_service)); break; case R.id.nextday: if (checked)**

**// Next day delivery**

**displayToast(getString(R.string.next\_day\_ground\_delivery)); break; case R.id.pickup: if (checked) // Pick up displayToast(getString(R.string.pick\_up)); break; default:**

**// Do nothing.**

**break;**

**}**

**}**

**/\*\***

* **Displays the actual message in a toast message.**

**\***

* **@param message Message to display.**

**\*/**

**public void displayToast(String message) { Toast.makeText(getApplicationContext(), message,**

**Toast.LENGTH\_SHORT).show();**

**}**

**}**

###### MainActivity.java

**package com.example.android.droidcafeoptions;**

**import android.content.Intent; import android.os.Bundle;**

**import android.support.design.widget.FloatingActionButton; import android.support.v7.app.AppCompatActivity; import android.support.v7.widget.Toolbar; import android.view.Menu; import android.view.MenuItem; import android.view.View; import android.widget.Toast;**

**/\*\***

* **This app demonstrates images used as buttons and a floating action button to**
* **use an intent to launch a second activity. The app lets a user tap an image**
* **to make a choice. The app displays a Toast message showing the** user’s choice,
* **and sends the choice as data with an intent to launch the second activity.**

**\***

* **This version includes options in the options menu, in which some of the**
* **options appear as icons in the app bar.**

**\*/ public class MainActivity extends AppCompatActivity {**

**// Tag for the intent extra.**

**public static final String EXTRA\_MESSAGE = "com.example.android.droidcafeoptions.extra.MESSAGE";**

**// The order message, displayed in the Toast and sent to the new Activity. private String mOrderMessage;**

**/\*\***

* **Creates the content view, the toolbar, and the floating action button.**
* **This method is provided in the Basic Activity template.**

**\***

* **@param savedInstanceState Saved instance state bundle.**

**\*/ @Override protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity\_main);**

**Toolbar toolbar = findViewById(R.id.toolbar); setSupportActionBar(toolbar);**

**FloatingActionButton fab = findViewById(R.id.fab); fab.setOnClickListener(new View.OnClickListener() {**

**@Override**

**public void onClick(View view) {**

**Intent intent = new Intent(MainActivity.this, OrderActivity.class); intent.putExtra(EXTRA\_MESSAGE, mOrderMessage); startActivity(intent);**

**}**

**});**

**}**

**/\*\***

* **Inflates the menu, and adds items to the action bar if it is present.**

**\***

* **@param menu Menu to inflate.**
* **@return Returns true if the menu inflated.**

**\*/ @Override public boolean onCreateOptionsMenu(Menu menu) {**

**// Inflate the menu; this adds items to the action bar if it is present.**

**getMenuInflater().inflate(R.menu.menu\_main, menu); return true;**

**}**

**/\*\***

* **Handles app bar item clicks.**

**\***

* **@param item Item clicked.**
* **@return True if one of the defined items was clicked.**

**\*/ @Override public boolean onOptionsItemSelected(MenuItem item) { switch (item.getItemId()) { case R.id.action\_order:**

**Intent intent = new Intent(MainActivity.this, OrderActivity.class); intent.putExtra(EXTRA\_MESSAGE, mOrderMessage);**

**startActivity(intent); return true; case R.id.action\_status:**

**displayToast(getString(R.string.action\_status\_message)); return true; case R.id.action\_favorites:**

**displayToast(getString(R.string.action\_favorites\_message)); return true; case R.id.action\_contact:**

**displayToast(getString(R.string.action\_contact\_message)); return true; default: // Do nothing**

**}**

**return super.onOptionsItemSelected(item);**

**}**

**/\*\***

* **Displays a Toast with the message. \***
* **@param message Message to display.**

**\*/**

**public void displayToast(String message) { Toast.makeText(getApplicationContext(), message,**

**Toast.LENGTH\_SHORT).show();**

**}**

**/\*\***

* **Shows a message that the donut image was clicked.**

**\*/**

**public void showDonutOrder(View view) {**

**mOrderMessage = getString(R.string.donut\_order\_message); displayToast(mOrderMessage);**

**}**

**/\*\***

* **Shows a message that the ice cream sandwich image was clicked. \*/**

**public void showIceCreamOrder(View view) {**

**mOrderMessage = getString(R.string.ice\_cream\_order\_message); displayToast(mOrderMessage);**

**}**

**/\*\***

* **Shows a message that the froyo image was clicked.**

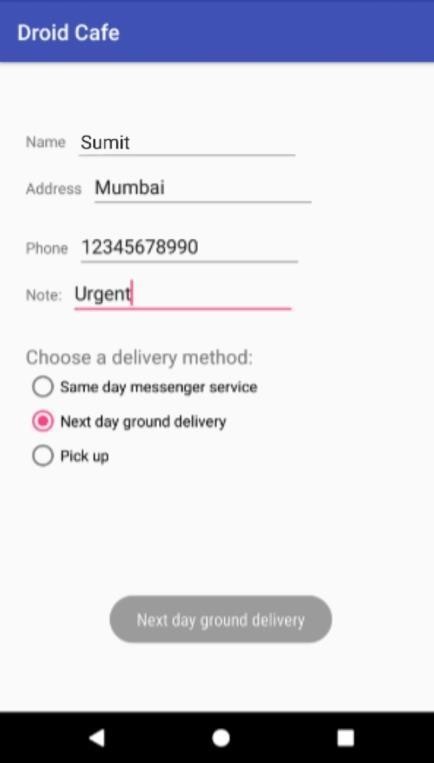
**\*/**

**public void showFroyoOrder(View view) {**

**mOrderMessage = getString(R.string.froyo\_order\_message); displayToast(mOrderMessage);**

**}**

**}**

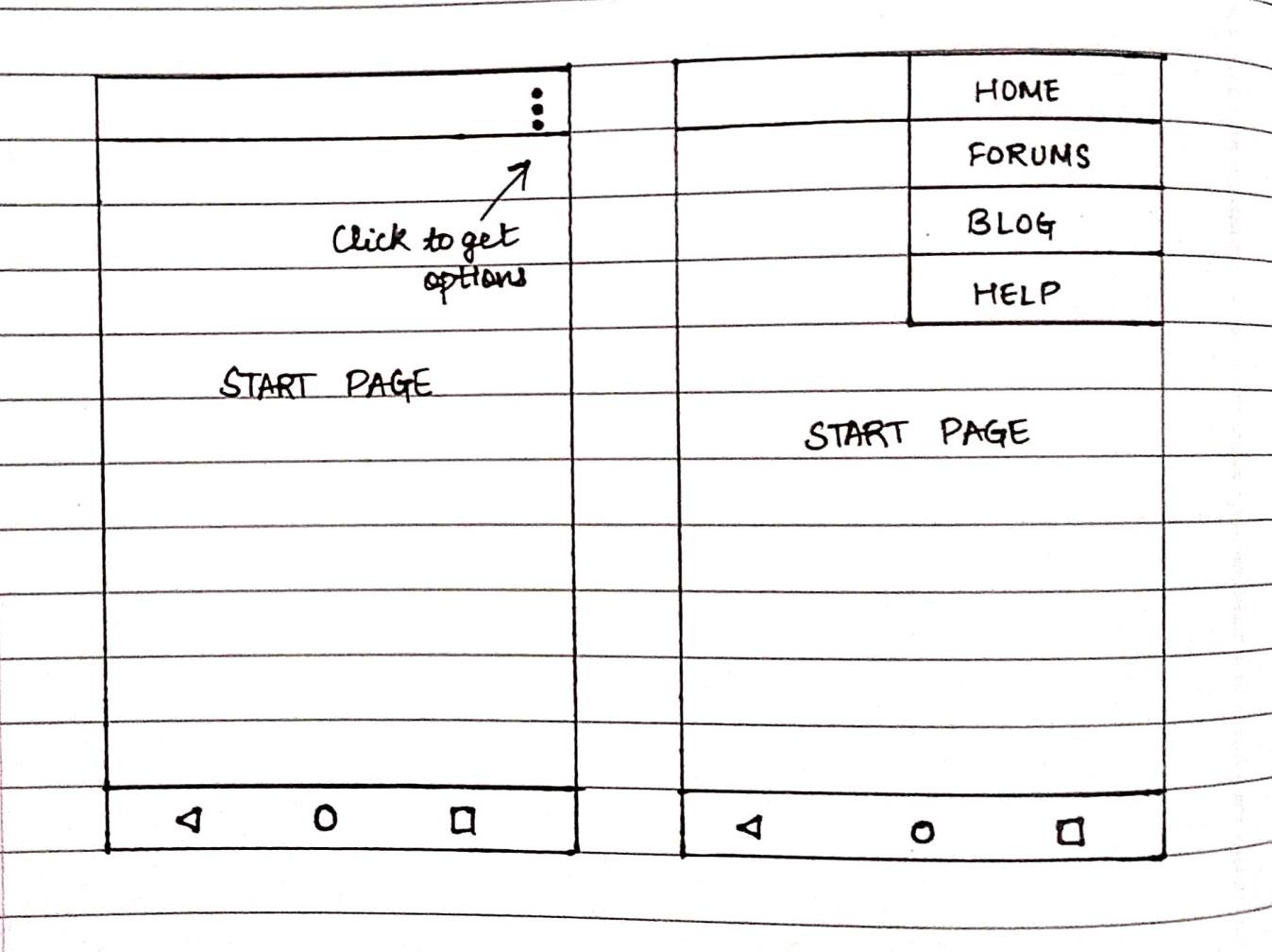


# Practical No – 6

##### **Aim:** Create an android app that demonstrates the use of an Options Menu.

**Theory:** In android, Options Menu is a primary collection of menu items for an [activity](https://www.tutlane.com/tutorial/android/android-activity-lifecycle) and it is useful to implement actions that have a global impact on the app, such as Settings, Search, etc.

**Diagrammatic representation:**



**Classes and Methods Used:**

<menu>

Defines a [Menu,](https://developer.android.com/reference/android/view/Menu) which is a container for menu items. A <menu> element must be the root node for the file and can hold one or more <item> and <group> elements.

<item>

Creates a [MenuItem,](https://developer.android.com/reference/android/view/MenuItem) which represents a single item in a menu. This element may contain a nested <menu> element in order to create a submenu.

<group>

An optional, invisible container for <item> elements. It allows you to categorize menu items so they share properties such as active state and visibility. For more information, see the section about [Creating Menu Groups](https://developer.android.com/guide/topics/ui/menus#groups).

The <item> element supports several attributes you can use to define an item's appearance and behavior. The items in the above menu include the following attributes:

android:id

A resource ID that's unique to the item, which allows the application to recognize the item when the user selects it.

android:icon

A reference to a drawable to use as the item's icon.

android:title

A reference to a string to use as the item's title.

android:showAsAction

Specifies when and how this item should appear as an action item in the app bar.

These are the most important attributes you should use, but there are many more available. For information about all the supported attributes, see the [Menu](https://developer.android.com/guide/topics/resources/menu-resource) [Resource](https://developer.android.com/guide/topics/resources/menu-resource) document.

You can add a submenu to an item in any menu by adding a <menu> element as the child of an <item>. Submenus are useful when your application has a lot of functions that can be organized into topics, like items in a PC application's menu bar (File, Edit, View, etc.).

#### Source Code: activity\_main.xml:

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

<androidx.constraintlayout.widget.ConstraintLayout xmlns:andr[oid="htt](http://schemas.android.com/apk/res/android)p:/[/sc](http://schemas.android.com/apk/res/android)he[mas.android.com/apk/res/android"](http://schemas.android.com/apk/res/android)

xmlns:app=["http://sc](http://schemas.android.com/apk/res-auto)h[emas.android.com/apk/res-auto"](http://schemas.android.com/apk/res-auto) xmlns:t[ools="ht](http://schemas.android.com/tools)tp:/[/sc](http://schemas.android.com/tools)he[mas.android.com/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\_constraintLeft\_toLeftOf="parent" app:layout\_constraintRight\_toRightOf="parent" app:layout\_constraintTop\_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

**menu\_main.xml:**

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

<menu xmlns:android="[http://schemas.android.com/apk/res/android"](http://schemas.android.com/apk/res/android) xmlns:app=["http://sc](http://schemas.android.com/apk/res-auto)h[emas.android.com/apk/res-auto"](http://schemas.android.com/apk/res-auto)>

<item

android:id="@+id/item1" android:icon="@drawable/ic\_baseline\_folder\_shared\_24" android:title="Shared"

app:showAsAction="ifRoom" />

<item

android:id="@+id/item2" android:title="About" app:showAsAction="never" />

<item

android:id="@+id/item3" android:title="Settings" app:showAsAction="never">

<menu>

<item

android:id="@+id/subitem1" android:title="Sub Item 1" />

<item

android:id="@+id/subitem2" android:title="Sub Item 2" />

</menu>

</item>

</menu>

#### MainActivity.java

package com.example.optionmenu;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle; import android.view.Menu;

import android.view.MenuInflater; import android.view.MenuItem; import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity\_main);

}

@Override

public boolean onCreateOptionsMenu(Menu menu) { getMenuInflater().inflate(R.menu.menu,menu); return true;

}

@Override

public boolean onOptionsItemSelected(MenuItem item) { switch (item.getItemId()) {

case R.id.item1:

Toast.makeText(this, "Item 1 selected", Toast.LENGTH\_SHORT).show(); return true;

case R.id.item2:

Toast.makeText(this, "About is selected", Toast.LENGTH\_SHORT).show(); return true;

case R.id.item3:

Toast.makeText(this, "Settings is selected", Toast.LENGTH\_SHORT).show(); return true;

case R.id.subitem1:

Toast.makeText(this, "Sub Item 1 selected", Toast.LENGTH\_SHORT).show(); return true;

case R.id.subitem2:

Toast.makeText(this, "Sub Item 2 selected", Toast.LENGTH\_SHORT).show(); return true;

default:

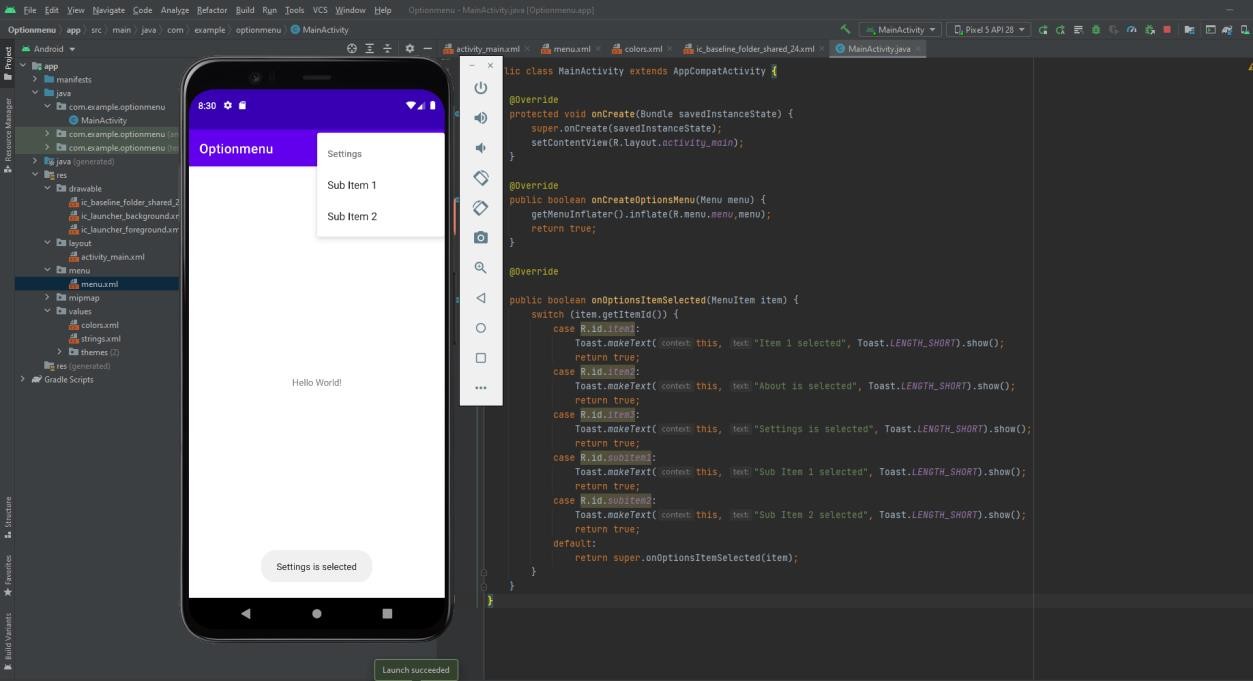
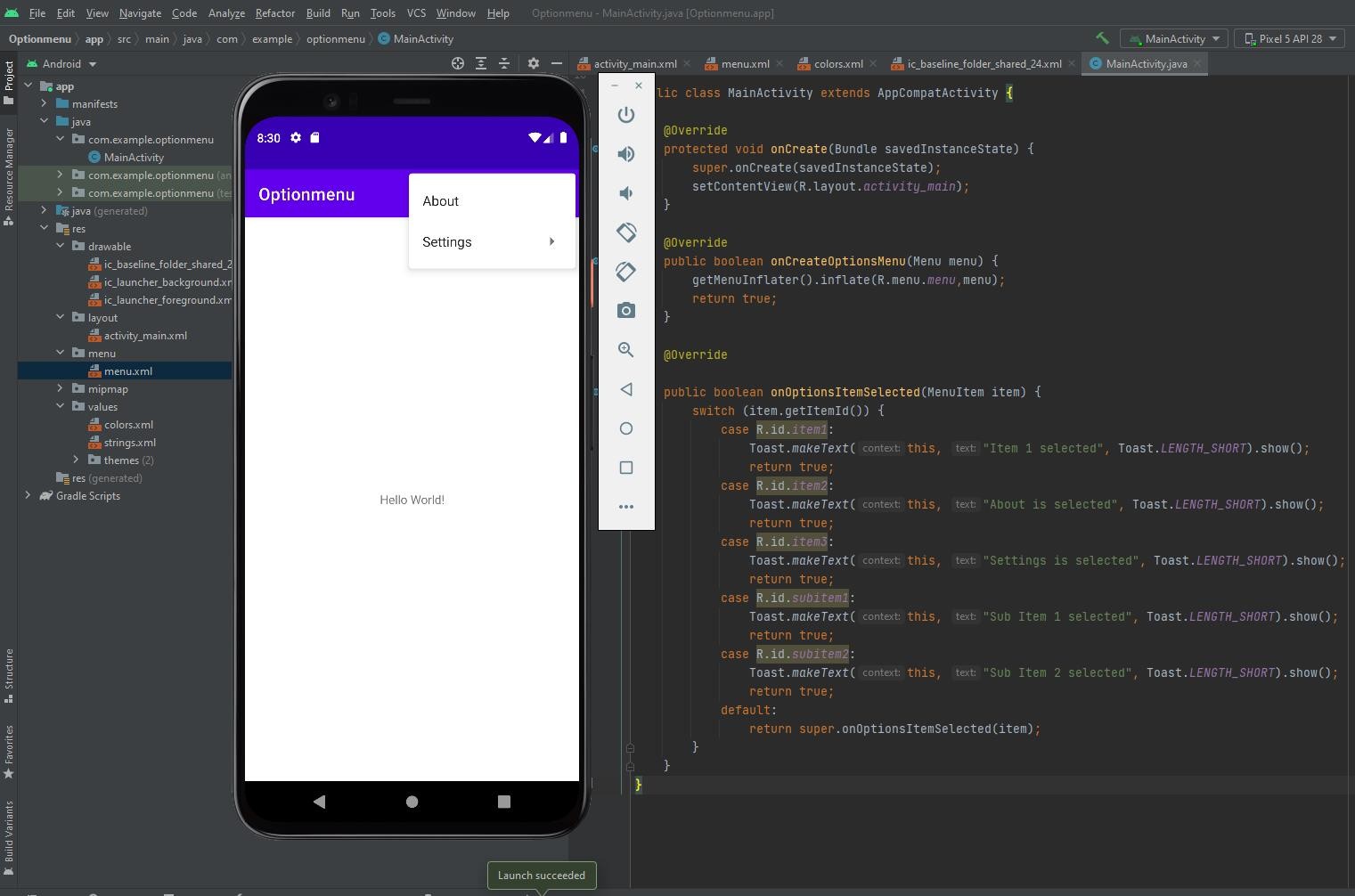
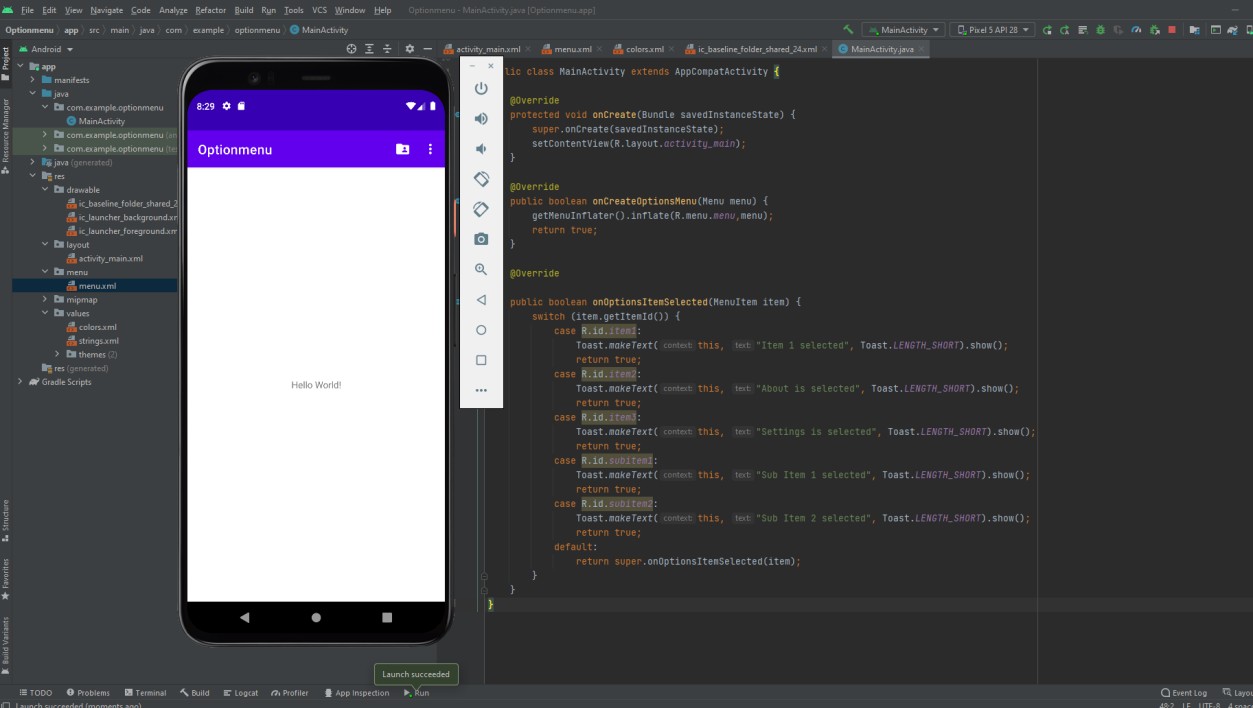
return super.onOptionsItemSelected(item);

}

}

}

**Output:**



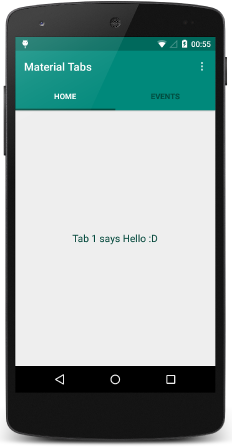
**Conclusion:** Thus we have created an android app that demonstrates the use of an Options Menu.

# Practical No – 7

##### **Aim:** Create an android app that demonstrate Screen Navigation Using the App Bar and Tabs

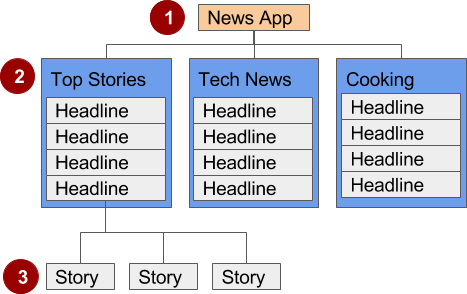
**Theory:** App for tab navigation that shows two tabs to navigate to sibling screens. When the user taps a tab, the screen shows a content screen, depending on which tab the user tapped. The user can also swipe left and right to visit the content screens. The ViewPager class automatically handles user swipes to screens or View elements.

Following is the pictorial representation of android app to demonstrate screen navigation using the app bar and tabs



**Classes and Methods Used:** The onBackPressed() method of the Activity class is called whenever the activity detects the user's press of the Back key.

The default implementation



1. **Parent screen**. A parent screen (such as a news app's home screen) enables navigation down to *child* screens.
   * The main activity of an app is usually the parent screen.
   * Implement a parent screen as an [Activity](https://developer.android.com/reference/android/app/Activity.html) with *descendant* navigation to one or more child screens.
2. **First-level child screen siblings**. Siblings are screens in the same position in the hierarchy that share the same parent screen (like brothers and sisters).
   * In the first level of siblings, the child screens may

be *collection* screens that collect the headlines of stories, as shown above.

* + Implement each child screen as an [Activity](https://developer.android.com/reference/android/app/Activity.html) or [Fragment](https://developer.android.com/reference/android/app/Fragment.html).
  + Implement *lateral* navigation to navigate from one sibling to another on the same level.
  + If there is a second level of screens, the first level child screen is the *parent* to the second level child screen siblings.

Implement *descendant* navigation to the second-level child screens.

1. **Second-level child screen siblings**. In news apps and others that offer multiple levels of information, the second level of child screen siblings might offer content, such as stories.
   * Implement a second-level child screen sibling as another [Activity](https://developer.android.com/reference/android/app/Activity.html) or [Fragment](https://developer.android.com/reference/android/app/Fragment.html).
   * Stories at this level may include embedded story elements such as videos, maps, and comments, which might be implemented as fragments.

simply finishes the current activity, but you can override this to do something else:

**Source Code:**

**activity\_main.xml:**

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

<LinearLayout xmlns:android=["http://schemas.android.com/](http://schemas.android.com/apk/res/android)a[pk/res/android"](http://schemas.android.com/apk/res/android) xmlns:app=["http://sc](http://schemas.android.com/apk/res-auto)h[emas.android.com/apk/res-auto"](http://schemas.android.com/apk/res-auto) xmlns:t[ools="ht](http://schemas.android.com/tools)tp:/[/sc](http://schemas.android.com/tools)he[mas.android.com/tools"](http://schemas.android.com/tools) android:layout\_width="match\_parent" android:layout\_height="match\_parent"

android:orientation="vertical" tools:context=".MainActivity">

<com.google.android.material.tabs.TabLayout android:layout\_width="match\_parent" android:layout\_height="50dp" android:id="@+id/tabLayout" app:tabMode="fixed"

app:tabGravity="fill" app:tabIndicatorGravity="stretch" app:tabSelectedTextColor="@color/white" app:tabIndicatorAnimationMode="elastic" app:tabIndicator="@drawable/tab\_indicator"/>

<androidx.viewpager2.widget.ViewPager2 android:layout\_width="match\_parent" android:layout\_height="match\_parent" android:id="@+id/viewPager2"/>

</LinearLayout>

#### fragment\_sign\_in.xml

<androidx.constraintlayout.widget.ConstraintLayout xmlns:andr[oid="htt](http://schemas.android.com/apk/res/android)p:/[/sc](http://schemas.android.com/apk/res/android)he[mas.android.com/apk/res/android"](http://schemas.android.com/apk/res/android)

xmlns:app=["http://sc](http://schemas.android.com/apk/res-auto)h[emas.android.com/apk/res-auto"](http://schemas.android.com/apk/res-auto) xmlns:t[ools="ht](http://schemas.android.com/tools)tp:/[/sc](http://schemas.android.com/tools)he[mas.android.com/tools"](http://schemas.android.com/tools) android:layout\_width="match\_parent" android:layout\_height="match\_parent" android:background="#E2E5DE" tools:context=".SignInFragment">

<androidx.cardview.widget.CardView android:layout\_width="match\_parent" android:layout\_height="0dp" android:layout\_marginVertical="10dp" app:cardCornerRadius="30dp" app:layout\_constraintBottom\_toTopOf="@+id/signInBtn" app:layout\_constraintTop\_toTopOf="parent">

<androidx.constraintlayout.widget.ConstraintLayout android:layout\_width="match\_parent" android:layout\_height="match\_parent">

<TextView android:id="@+id/textView" android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content" android:layout\_marginStart="24dp" android:layout\_marginTop="24dp" android:text="@string/sign\_in\_account" android:textColor="@color/red"

android:textSize="20sp" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toTopOf="parent" />

<TextView android:id="@+id/textview2" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_marginStart="26dp" android:layout\_marginTop="16dp" android:text="Email ID" android:textColor="@color/black" android:textSize="12sp"

app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toBottomOf="@+id/textView3" />

<EditText android:id="@+id/editEmailSignIN" android:layout\_width="0dp" android:layout\_height="50dp" android:layout\_marginStart="24dp" android:layout\_marginEnd="24dp" android:backgroundTint="@color/dark\_red"

android:drawableRight="@drawable/ic\_baseline\_person" android:ems="10" android:inputType="textEmailAddress" android:textColor="#8c8c8c" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintHorizontal\_bias="0.0" app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/textview2" />

<TextView android:id="@+id/textView3" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_marginStart="24dp" android:text="Account" android:textColor="@color/dark\_red" android:textSize="20sp"

app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toBottomOf="@+id/textView" />

<TextView android:id="@+id/textView4" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_marginStart="26dp" android:layout\_marginTop="8dp" android:text="Password" android:textColor="@color/black" android:textSize="12sp"

app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toBottomOf="@+id/editEmailSignIN" />

<EditText android:id="@+id/editPassSignIn" android:layout\_width="0dp" android:layout\_height="50dp" android:layout\_marginStart="24dp"

android:layout\_marginEnd="24dp" android:backgroundTint="@color/dark\_red" android:drawableRight="@drawable/ic\_baseline\_lock\_open" android:ems="10"

android:inputType="textPassword" android:textColor="#8c8c8c" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintHorizontal\_bias="0.0" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toBottomOf="@+id/textView4" />

<TextView android:id="@+id/textView5" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_marginTop="8dp" android:layout\_marginEnd="24dp" android:text="Forgot Password ?" android:textColor="@color/dark\_red" android:textSize="14sp"

app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintTop\_toBottomOf="@+id/editPassSignIn" />

<ImageView android:id="@+id/fbLoginBtn" android:layout\_width="30dp" android:layout\_height="30dp" android:layout\_marginStart="140dp" android:layout\_marginTop="8dp" android:src="@drawable/facebook"

app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toBottomOf="@+id/textView6" />

<ImageView android:id="@+id/googleLoginBtn" android:layout\_width="40dp" android:layout\_height="40dp" android:layout\_marginStart="48dp" android:layout\_marginTop="5dp" android:src="@drawable/googlelogo"

app:layout\_constraintStart\_toEndOf="@+id/fbLoginBtn" app:layout\_constraintTop\_toBottomOf="@+id/textView6" />

<TextView android:id="@+id/textView6" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_marginStart="160dp" android:layout\_marginTop="8dp" android:text="Login with" android:textColor="@color/black"

app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toBottomOf="@+id/textView5" />

<TextView android:id="@+id/textView7" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_marginStart="80dp" android:layout\_marginTop="8dp"

android:text="Don't have an account?" android:textColor="@color/black" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toBottomOf="@+id/fbLoginBtn" />

<TextView android:id="@+id/signUpText" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_marginStart="8dp" android:text="@string/register\_text" android:textColor="@color/red"

app:layout\_constraintStart\_toEndOf="@+id/textView7" app:layout\_constraintTop\_toTopOf="@+id/textView7" />

<ImageView android:id="@+id/imageView5" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_marginTop="16dp" android:src="@drawable/signup"

app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintHorizontal\_bias="0.5" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toBottomOf="@+id/textView7" />

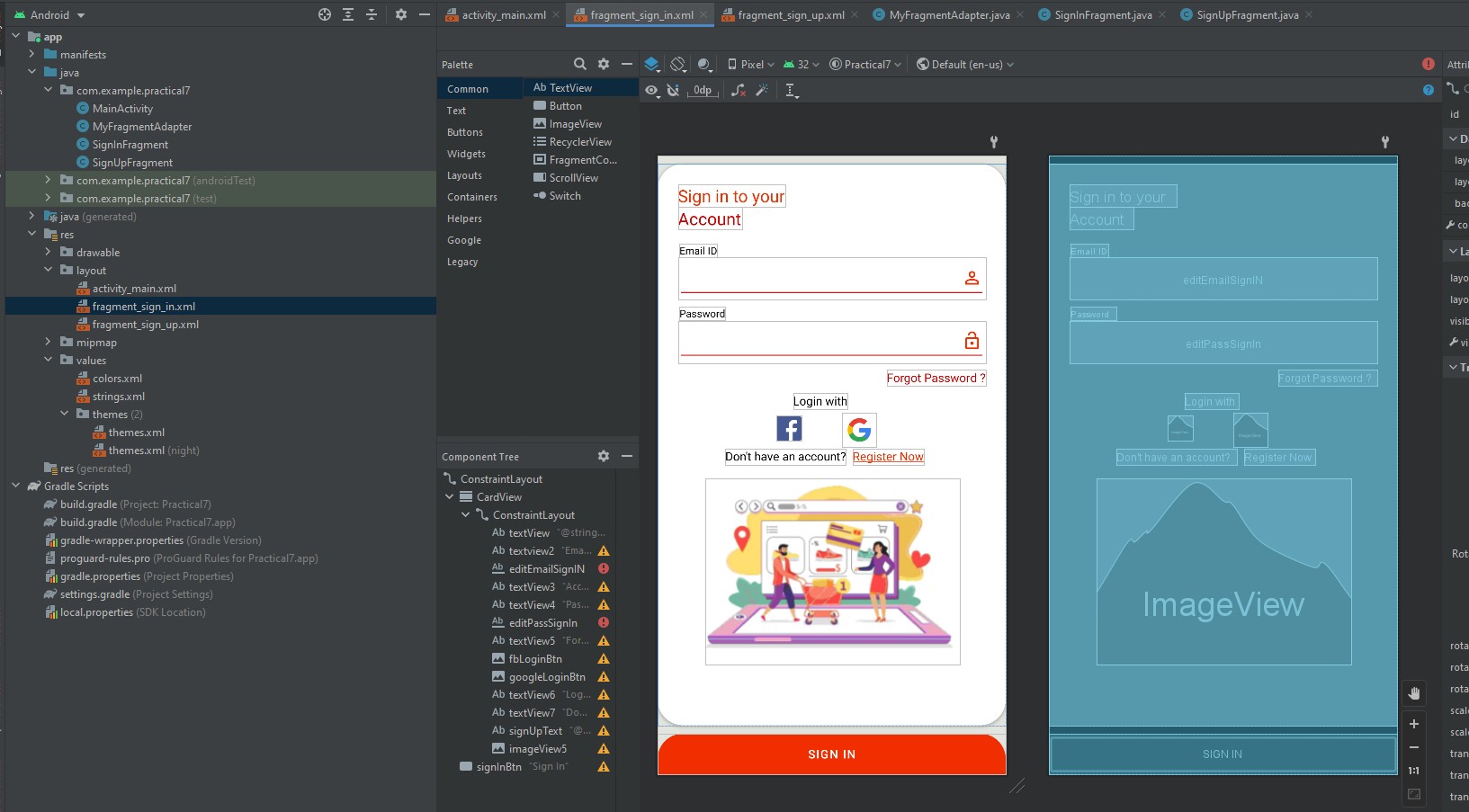
</androidx.constraintlayout.widget.ConstraintLayout>

</androidx.cardview.widget.CardView>

<Button

android:id="@+id/signInBtn" android:layout\_width="0dp" android:layout\_height="wrap\_content" android:text="Sign In" android:background="@drawable/btn\_background" app:layout\_constraintBottom\_toBottomOf="parent" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintHorizontal\_bias="0.5" app:layout\_constraintStart\_toStartOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>



#### fragment\_sign\_up.xml

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

<androidx.constraintlayout.widget.ConstraintLayout xmlns:andr[oid="htt](http://schemas.android.com/apk/res/android)p:/[/sc](http://schemas.android.com/apk/res/android)he[mas.android.com/apk/res/android"](http://schemas.android.com/apk/res/android)

xmlns:app=["http://sc](http://schemas.android.com/apk/res-auto)h[emas.android.com/apk/res-auto"](http://schemas.android.com/apk/res-auto) xmlns:t[ools="ht](http://schemas.android.com/tools)tp:/[/sc](http://schemas.android.com/tools)he[mas.android.com/tools"](http://schemas.android.com/tools)

android:layout\_width="match\_parent" android:layout\_height="match\_parent" android:background="#E2E5DE" tools:context=".SignInFragment">

<androidx.cardview.widget.CardView android:layout\_width="match\_parent" android:layout\_height="0dp" android:layout\_marginVertical="10dp" app:cardCornerRadius="30dp" app:layout\_constraintBottom\_toTopOf="@+id/signUpBtn" app:layout\_constraintTop\_toTopOf="parent">

<androidx.constraintlayout.widget.ConstraintLayout android:layout\_width="match\_parent" android:layout\_height="match\_parent">

<TextView android:id="@+id/textView" android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content" android:layout\_marginStart="24dp" android:layout\_marginTop="24dp" android:text="Create an" android:textColor="@color/red" android:textSize="20sp" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toTopOf="parent" />

<TextView android:id="@+id/textview2" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_marginStart="26dp" android:layout\_marginTop="16dp" android:text="Name" android:textColor="@color/black" android:textSize="12sp"

app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toBottomOf="@+id/textView3" />

<EditText android:id="@+id/editNameSignUp" android:layout\_width="0dp" android:layout\_height="50dp" android:layout\_marginStart="24dp" android:layout\_marginEnd="24dp" android:backgroundTint="@color/dark\_red"

android:drawableRight="@drawable/ic\_baseline\_person" android:ems="10" android:inputType="textEmailAddress" android:textColor="#8c8c8c" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintHorizontal\_bias="0.0" app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/textview2" />

<TextView android:id="@+id/textView3"

android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_marginStart="24dp" android:text="Account" android:textColor="@color/dark\_red" android:textSize="20sp" app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/textView" />

<TextView android:id="@+id/textView4" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_marginStart="26dp" android:layout\_marginTop="8dp" android:text="Email ID" android:textColor="@color/black" android:textSize="12sp"

app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toBottomOf="@+id/editNameSignUp" />

<EditText android:id="@+id/editEmailSignUp" android:layout\_width="0dp" android:layout\_height="50dp" android:layout\_marginStart="24dp" android:layout\_marginEnd="24dp" android:backgroundTint="@color/dark\_red"

android:drawableRight="@drawable/ic\_baseline\_email\_24" android:ems="10"

android:inputType="textEmailAddress" android:textColor="#8c8c8c" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintHorizontal\_bias="0.0" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toBottomOf="@+id/textView4" />

<TextView android:id="@+id/textView7" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_marginStart="68dp" android:layout\_marginTop="12dp"

android:text="Already have an account ?" android:textColor="@color/black" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toBottomOf="@+id/checkBox" />

<TextView android:id="@+id/signInText" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_marginStart="8dp" android:text="Sign in !" android:textColor="@color/dark\_red"

app:layout\_constraintStart\_toEndOf="@+id/textView7" app:layout\_constraintTop\_toTopOf="@+id/textView7" />

<EditText android:id="@+id/editNumberSignUp"

android:layout\_width="0dp" android:layout\_height="50dp" android:layout\_marginStart="24dp" android:layout\_marginEnd="24dp" android:backgroundTint="@color/dark\_red" android:drawableRight="@drawable/ic\_baseline\_call\_24" android:ems="10"

android:inputType="number" android:textColor="#8c8c8c" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintHorizontal\_bias="0.5" app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/textView9" />

<TextView android:id="@+id/textView9" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_marginStart="24dp" android:layout\_marginTop="8dp" android:text="Contact no"

android:textColor="@color/black" android:textSize="12sp" app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/editEmailSignUp" />

<EditText

android:id="@+id/editPassSignUp"

android:layout\_width="0dp" android:layout\_height="50dp" android:layout\_marginStart="24dp" android:layout\_marginEnd="24dp" android:backgroundTint="@color/dark\_red"

android:drawableRight="@drawable/ic\_baseline\_lock\_open" android:ems="10"

android:inputType="textPassword" android:textColor="#8c8c8c" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintHorizontal\_bias="0.5" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toBottomOf="@+id/textvieww" />

<TextView

android:id="@+id/textvieww" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_marginStart="24dp" android:layout\_marginTop="8dp" android:text="Password" android:textColor="@color/black" android:textSize="12sp" app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/editNumberSignUp" />

<CheckBox android:id="@+id/checkBox" android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content" android:layout\_marginStart="24dp" android:layout\_marginTop="16dp" android:text="Accept terms and conditions" android:textColor="@color/dark\_red" app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/editPassSignUp" />

<ImageView android:id="@+id/imageView9" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:src="@drawable/signin"

app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintHorizontal\_bias="0.5" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toBottomOf="@+id/textView7" />

<ProgressBar android:id="@+id/progressBar" style="?android:attr/progressBarStyle" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:visibility="invisible"

app:layout\_constraintBottom\_toBottomOf="parent" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintHorizontal\_bias="0.5" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toTopOf="parent" />

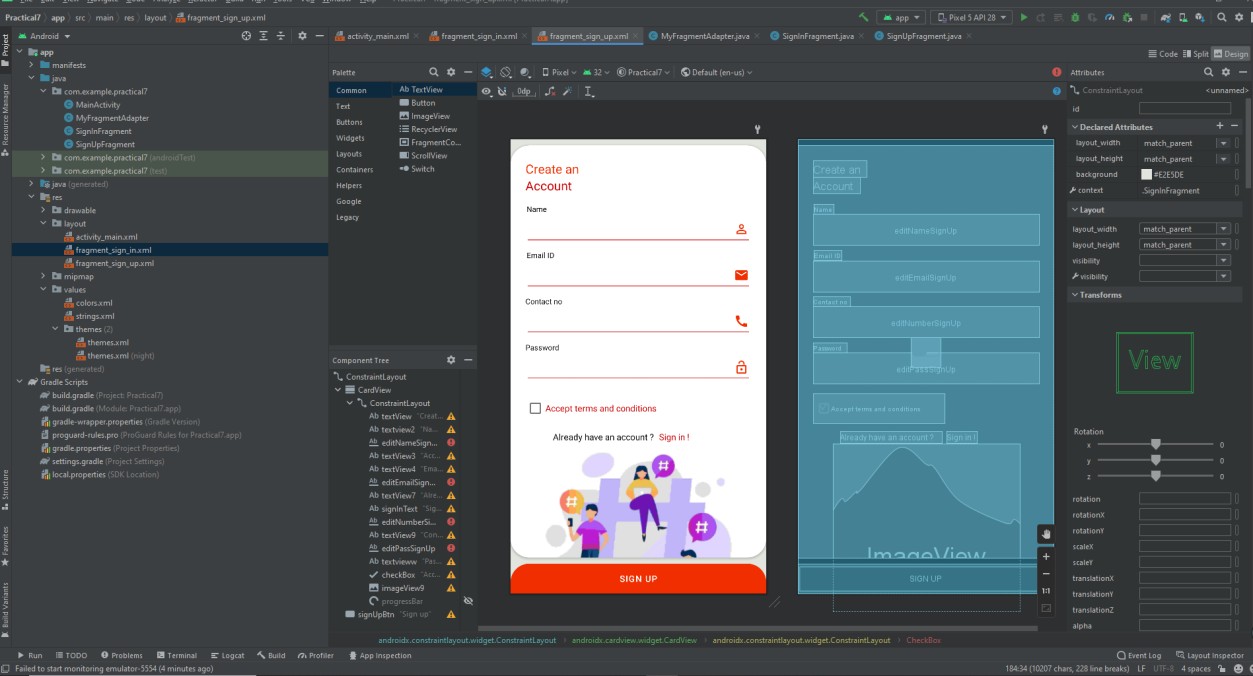
</androidx.constraintlayout.widget.ConstraintLayout>

</androidx.cardview.widget.CardView>

<Button

android:id="@+id/signUpBtn" android:layout\_width="0dp" android:layout\_height="wrap\_content" android:background="@drawable/btn\_background" android:text="Sign up" app:layout\_constraintBottom\_toBottomOf="parent" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintHorizontal\_bias="0.5" app:layout\_constraintStart\_toStartOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>



**SignUpFragment.java**

package com.example.practical7;

import android.os.Bundle;

import androidx.fragment.app.Fragment;

import android.view.LayoutInflater; import android.view.View;

import android.view.ViewGroup;

/\*\*

* A simple {@link Fragment} subclass.
* Use the {@link SignUpFragment#newInstance} factory method to
* create an instance of this fragment.

\*/

public class SignUpFragment extends Fragment {

// TODO: Rename parameter arguments, choose names that match

// the fragment initialization parameters, e.g. ARG\_ITEM\_NUMBER private static final String ARG\_PARAM1 = "param1";

private static final String ARG\_PARAM2 = "param2";

// TODO: Rename and change types of parameters private String mParam1;

private String mParam2;

public SignUpFragment() {

// Required empty public constructor

}

/\*\*

* + Use this factory method to create a new instance of
  + this fragment using the provided parameters.

\*

* + @param param1 Parameter 1.
  + @param param2 Parameter 2.
  + @return A new instance of fragment SignUpFragment.

\*/

// TODO: Rename and change types and number of parameters

public static SignUpFragment newInstance(String param1, String param2) { SignUpFragment fragment = new SignUpFragment();

Bundle args = new Bundle(); args.putString(ARG\_PARAM1, param1); args.putString(ARG\_PARAM2, param2); fragment.setArguments(args);

return fragment;

}

@Override

public void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState);

if (getArguments() != null) {

mParam1 = getArguments().getString(ARG\_PARAM1); mParam2 = getArguments().getString(ARG\_PARAM2);

}

}

@Override

public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {

// Inflate the layout for this fragment

return inflater.inflate(R.layout.fragment\_sign\_up, container, false);

}

}

**SignInFragment.java**

package com.example.practical7;

import android.os.Bundle;

import androidx.fragment.app.Fragment;

import android.view.LayoutInflater; import android.view.View;

import android.view.ViewGroup;

/\*\*

* A simple {@link Fragment} subclass.
* Use the {@link SignInFragment#newInstance} factory method to
* create an instance of this fragment.

\*/

public class SignInFragment extends Fragment {

// TODO: Rename parameter arguments, choose names that match

// the fragment initialization parameters, e.g. ARG\_ITEM\_NUMBER private static final String ARG\_PARAM1 = "param1";

private static final String ARG\_PARAM2 = "param2";

// TODO: Rename and change types of parameters private String mParam1;

private String mParam2;

public SignInFragment() {

// Required empty public constructor

}

/\*\*

* + Use this factory method to create a new instance of
  + this fragment using the provided parameters.

\*

* + @param param1 Parameter 1.
  + @param param2 Parameter 2.
  + @return A new instance of fragment SignInFragment.

\*/

// TODO: Rename and change types and number of parameters

public static SignInFragment newInstance(String param1, String param2) { SignInFragment fragment = new SignInFragment();

Bundle args = new Bundle(); args.putString(ARG\_PARAM1, param1); args.putString(ARG\_PARAM2, param2); fragment.setArguments(args);

return fragment;

}

@Override

public void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState);

if (getArguments() != null) {

mParam1 = getArguments().getString(ARG\_PARAM1); mParam2 = getArguments().getString(ARG\_PARAM2);

}

}

@Override

public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {

// Inflate the layout for this fragment

return inflater.inflate(R.layout.fragment\_sign\_in, container, false);

}

}

**MyFragmnetAdapter.java**

package com.example.practical7;

import androidx.annotation.NonNull; import androidx.fragment.app.Fragment;

import androidx.fragment.app.FragmentManager; import androidx.lifecycle.Lifecycle;

import androidx.viewpager2.adapter.FragmentStateAdapter;

public class MyFragmentAdapter extends FragmentStateAdapter {

public MyFragmentAdapter(@NonNull FragmentManager fragmentManager, @NonNull Lifecycle lifecycle) {

super(fragmentManager, lifecycle);

}

@NonNull @Override

public Fragment createFragment(int position) { if (position == 1){

return new SignUpFragment();

}

return new SignInFragment();

}

@Override

public int getItemCount() { return 2;

}

}

#### MainActivity.java

package com.example.practical7;

import androidx.appcompat.app.AppCompatActivity; import androidx.fragment.app.FragmentManager; import androidx.viewpager2.widget.ViewPager2;

import android.os.Bundle;

import com.google.android.material.tabs.TabLayout;

public class MainActivity extends AppCompatActivity {

private TabLayout tabLayout; private ViewPager2 viewPager2; private MyFragmentAdapter adapter;

@Override

protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity\_main);

tabLayout = findViewById(R.id.tabLayout); viewPager2 = findViewById(R.id.viewPager2);

tabLayout.addTab(tabLayout.newTab().setText("Sign In")); tabLayout.addTab(tabLayout.newTab().setText("Sign Up"));

FragmentManager fragmentManager = getSupportFragmentManager(); adapter = new MyFragmentAdapter(fragmentManager , getLifecycle()); viewPager2.setAdapter(adapter);

tabLayout.addOnTabSelectedListener(new TabLayout.OnTabSelectedListener() { @Override

public void onTabSelected(TabLayout.Tab tab) { viewPager2.setCurrentItem(tab.getPosition());

}

@Override

public void onTabUnselected(TabLayout.Tab tab) {

}

@Override

public void onTabReselected(TabLayout.Tab tab) {

}

});

viewPager2.registerOnPageChangeCallback(new ViewPager2.OnPageChangeCallback() {

@Override

public void onPageSelected(int position) { tabLayout.selectTab(tabLayout.getTabAt(position));

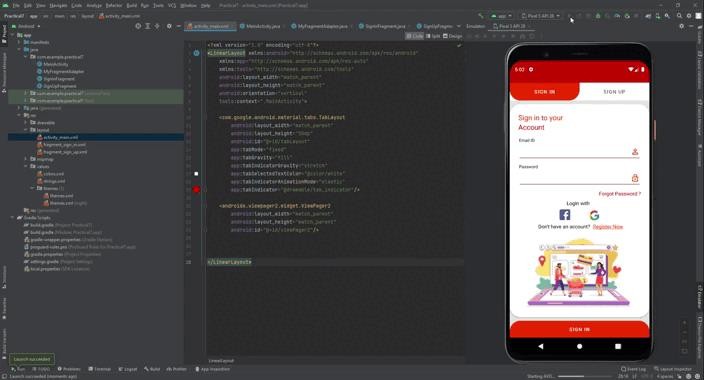
}

});

}

}

#### Output:



**Conclusion:** Thus we have created an android app that demonstrates screen navigation using the app bar and tabs

# Practical No – 8

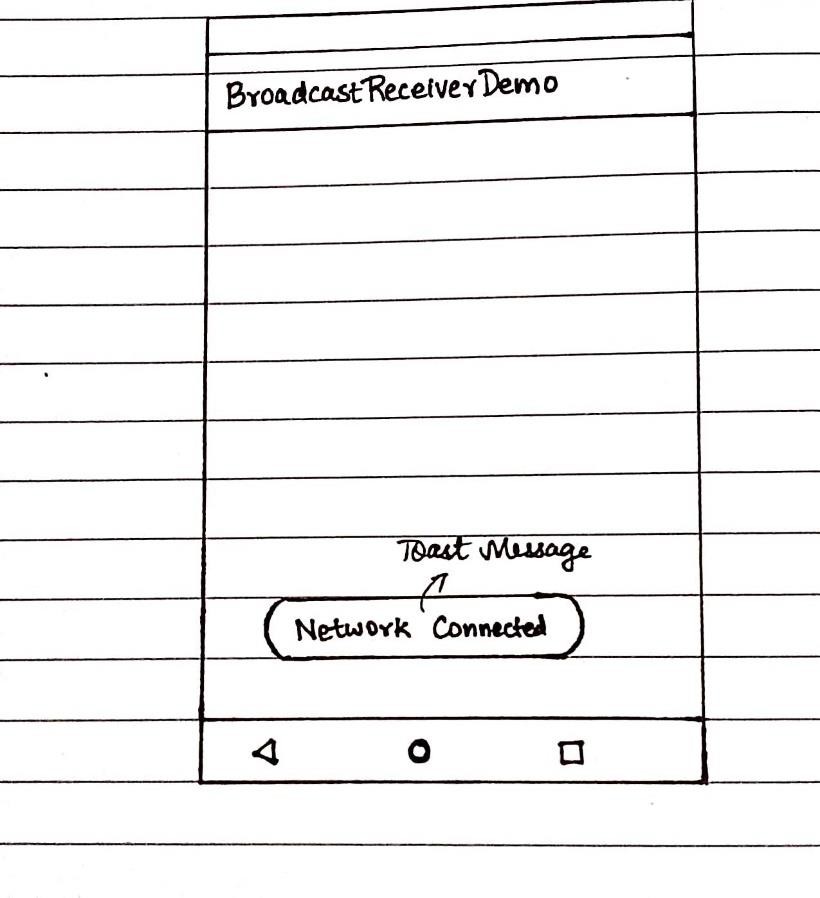
##### **Aim:** Create an android app to Connect to the Internet and use BroadcastReceiver

**Theory:** Broadcast receiver is a medium through which communication is done with the registered app and the outside OS system. We needs to register and unregister this functionality according to our requirement. At first we have to register each broadcast receiver to get appropriate notification during desired situations. Broadcast receiver receives

notification for a small span of time so we can’t do any kind of long running process over here and we need to unregister the same when app goes to background specifically when app is going to pause mode.

It is very easy method to check all the time internet connectivity through Broadcast receiver. It will notify at any point of time when network connection is discontinued or network status changed.

**Diagrammatic representation:**



**Classes and Methods Used:**

The **sendOrderedBroadcast(Intent, String)** method sends broadcasts to one receiver at a time. As each receiver executes in turn, it can propagate a result to the next receiver, or it can completely abort the broadcast so that it won't be passed to other receivers. The order receivers run in can be controlled with the android:priority attribute of the matching intent-filter; receivers with the same priority will be run in an arbitrary order.

The **sendBroadcast(Intent)** method sends broadcasts to all receivers in an undefined order. This is called a Normal Broadcast. This is more efficient, but means that receivers cannot read results from other receivers, propagate data received from the broadcast, or abort the broadcast.

The **LocalBroadcastManager.sendBroadcast** method sends broadcasts to receivers that are in the same app as the sender. If you don't need to send broadcasts across apps, use local broadcasts. The implementation is much more efficient (no interprocess communication needed) and you don't need to worry about any security issues related to other apps being able to receive or send your broadcasts.

When you call **sendBroadcast(Intent, String) or sendOrderedBroadcast(Intent, String, BroadcastReceiver, Handler, int, String, Bundle)**, you can specify a permission parameter. Only receivers who have requested that permission with the tag in their manifest (and subsequently been granted the permission if it is dangerous) can receive the broadcast.

#### activity\_main.xml:

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

<androidx.constraintlayout.widget.ConstraintLayout xmlns:andr[oid="htt](http://schemas.android.com/apk/res/android)p:/[/sc](http://schemas.android.com/apk/res/android)he[mas.android.com/apk/res/android"](http://schemas.android.com/apk/res/android)

xmlns:app=["http://sc](http://schemas.android.com/apk/res-auto)h[emas.android.com/apk/res-auto"](http://schemas.android.com/apk/res-auto) xmlns:t[ools="ht](http://schemas.android.com/tools)tp:/[/sc](http://schemas.android.com/tools)he[mas.android.com/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="Broadcast Receiver by checking internet connection" app:layout\_constraintBottom\_toBottomOf="parent" app:layout\_constraintLeft\_toLeftOf="parent" app:layout\_constraintRight\_toRightOf="parent" app:layout\_constraintTop\_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

#### MainActivity.java:

package com.example.internet;

import androidx.appcompat.app.AppCompatActivity;

import android.content.BroadcastReceiver; import android.content.IntentFilter;

import android.net.ConnectivityManager; import android.os.Bundle;

public class MainActivity extends AppCompatActivity { BroadcastReceiver broadcastReceiver = null; @Override

protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity\_main);

broadcastReceiver = new InternetReceiver(); Internetstatus();

}

public void Internetstatus(){ registerReceiver(broadcastReceiver,new

IntentFilter(ConnectivityManager.CONNECTIVITY\_ACTION));

}

@Override

protected void onPause() { super.onPause(); unregisterReceiver(broadcastReceiver);

}

}

#### CheckInternet.java:

package com.example.internet;

import android.content.Context;

import android.net.ConnectivityManager; import android.net.NetworkInfo;

public class CheckInternet {

public static String getNetworkInfo(Context context){ String status = null;

ConnectivityManager connectivityManager = (ConnectivityManager) context.getSystemService(Context.CONNECTIVITY\_SERVICE);

NetworkInfo networkInfo = connectivityManager.getActiveNetworkInfo(); if (networkInfo!=null){

status = "connected"; return status;

}

else {

status = "disconnected"; return status;

}

}

}

#### InternetReceiver.java:

package com.example.internet;

import android.content.BroadcastReceiver; import android.content.Context;

import android.content.Intent; import android.widget.Toast;

public class InternetReceiver extends BroadcastReceiver { @Override

public void onReceive(Context context, Intent intent) { String status = CheckInternet.getNetworkInfo(context); if (status.equals("connected")){

Toast.makeText(context, "Internet is Connected",Toast.LENGTH\_LONG).show();

}

else if (status.equals("disconnected")){

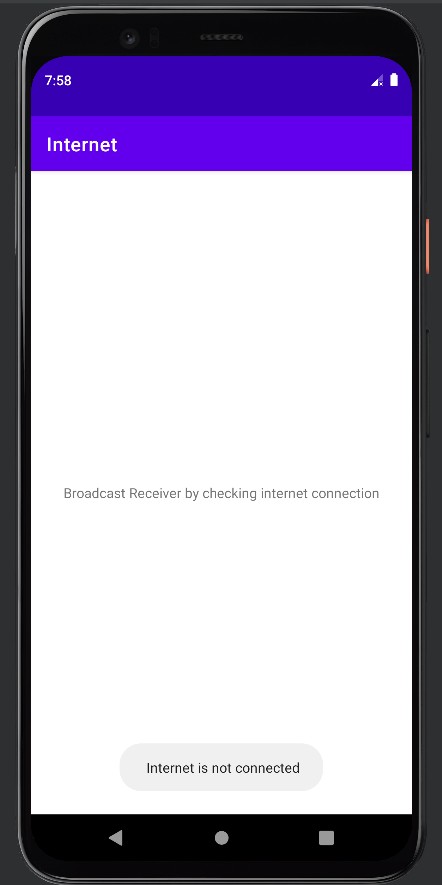
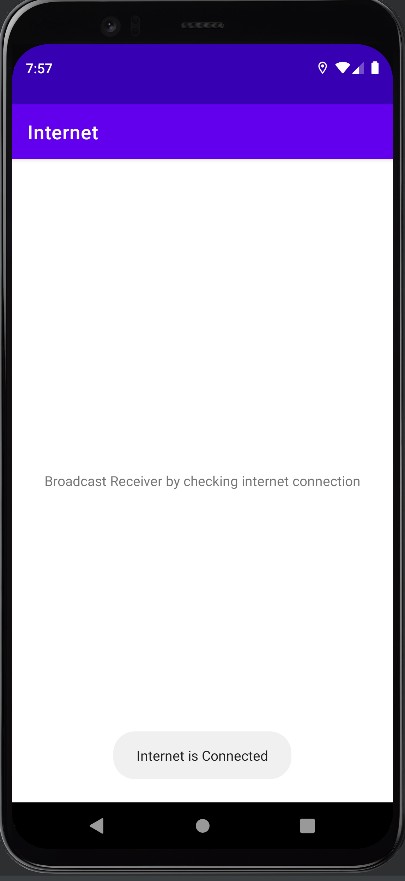
Toast.makeText(context, " Internet is not connected",Toast.LENGTH\_LONG).show();

}

}

}

#### Output:



**Conclusion:** Thus we have created an android app to Connect to the Internet and use BroadcastReceiver

# Practical No – 9

##### **Aim:** Create an android app to show Notifications and Alarm manager

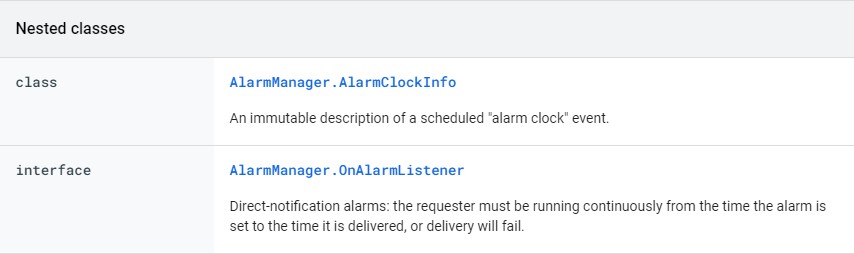
**Theory:** By the help of Android AlarmManager in android, you can schedule your application to run at a specific time in the future. It works whether your phone is running or not.

The Android AlarmManager holds a CPU wake lock that provides guarantee not to sleep the phone until broadcast is handled.

Following is the pictorial image of an android app to show Notifications and Alarm manager



**Classes and Methods Used:**



|  |  |
| --- | --- |
| **Public Methods** |  |
| boolean | [canScheduleExactAlarms](https://developer.android.com/reference/android/app/AlarmManager#canScheduleExactAlarms())()  Called to check if the caller can schedule exact alarms. |
| void | [cancel](https://developer.android.com/reference/android/app/AlarmManager#cancel(android.app.PendingIntent))([PendingIntent](https://developer.android.com/reference/android/app/PendingIntent) operation)  Remove any alarms with a matching [Intent](https://developer.android.com/reference/android/content/Intent). |
| [AlarmManager.A](https://developer.android.com/reference/android/app/AlarmManager.AlarmClockInfo) [larmClockInfo](https://developer.android.com/reference/android/app/AlarmManager.AlarmClockInfo) | [getNextAlarmClock](https://developer.android.com/reference/android/app/AlarmManager#getNextAlarmClock())()  Gets information about the next alarm clock currently scheduled. |
| void | [set](https://developer.android.com/reference/android/app/AlarmManager#set(int%2C%20long%2C%20android.app.PendingIntent))(int type, long triggerAtMillis, [PendingIntent](https://developer.android.com/reference/android/app/PendingIntent) operation) Schedule an alarm. |
| void | [cancel](https://developer.android.com/reference/android/app/AlarmManager#cancel(android.app.AlarmManager.OnAlarmListener))([AlarmManager.OnAlarmListener](https://developer.android.com/reference/android/app/AlarmManager.OnAlarmListener) listener)  Remove any alarm scheduled to be delivered to the given [OnAlarmListener](https://developer.android.com/reference/android/app/AlarmManager.OnAlarmListener). |
| void | [setTimeZone](https://developer.android.com/reference/android/app/AlarmManager#setTimeZone(java.lang.String))([String](https://developer.android.com/reference/java/lang/String) timeZone)  Sets the system's persistent default time zone. |
| void | [setTimeZone](https://developer.android.com/reference/android/app/AlarmManager#setTimeZone(java.lang.String))([String](https://developer.android.com/reference/java/lang/String) timeZone)  Sets the system's persistent default time zone. |
| void | [setExact](https://developer.android.com/reference/android/app/AlarmManager#setExact(int%2C%20long%2C%20android.app.PendingIntent))(int type, long triggerAtMillis, [PendingIntent](https://developer.android.com/reference/android/app/PendingIntent) operation) Schedule an alarm to be delivered precisely at the stated time. |

**canScheduleExactAlarms:** Called to check if the caller can schedule exact alarms. Your app schedules exact alarms when it calls any of the setExact... or setAlarmClock API methods.

**public void cancel (PendingIntent operation):** Remove any alarms with a matching [Intent](https://developer.android.com/reference/android/content/Intent). Any alarm, of any type, whose Intent matches this one (as defined by [Intent#filterEquals](https://developer.android.com/reference/android/content/Intent#filterEquals(android.content.Intent))), will be canceled.

**public AlarmManager.AlarmClockInfo getNextAlarmClock ():** Gets information about the next alarm clock currently scheduled. The alarm clocks considered are those scheduled by any application using the setAlarmClock(AlarmManager.AlarmClockInfo, PendingIntent) method.

#### activity\_main.xml:

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

<LinearLayout xmlns:android=["http://schemas.android.com/](http://schemas.android.com/apk/res/android)a[pk/res/android"](http://schemas.android.com/apk/res/android) xmlns:app=["http://sc](http://schemas.android.com/apk/res-auto)h[emas.android.com/apk/res-auto"](http://schemas.android.com/apk/res-auto) xmlns:t[ools="ht](http://schemas.android.com/tools)tp:/[/sc](http://schemas.android.com/tools)he[mas.android.com/tools"](http://schemas.android.com/tools) android:layout\_width="match\_parent" android:layout\_height="match\_parent"

android:orientation="vertical" tools:context=".MainActivity">

<TextView android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:id="@+id/selectedTime" android:layout\_marginTop="200dp" android:text="08 : 45 PM" android:textAlignment="center" android:textColor="@color/black" android:textSize="46dp"/>

<Button

android:id="@+id/selectTimeBtn" android:layout\_width="match\_parent" android:layout\_height="70dp" android:layout\_marginTop="36dp" android:layout\_marginHorizontal="26dp" android:text="Select Time" android:textSize="26dp"/>

<Button

android:id="@+id/setAlarmBtn" android:layout\_width="match\_parent" android:layout\_height="70dp" android:layout\_marginTop="8dp" android:layout\_marginHorizontal="26dp" android:text="Set Alarm" android:textSize="26dp"/>

<Button

android:id="@+id/cancelAlarmBtn" android:layout\_width="match\_parent" android:layout\_height="70dp" android:layout\_marginTop="8dp" android:layout\_marginHorizontal="26dp" android:text="Cancel Alarm" android:textSize="26dp"/>

</LinearLayout>

#### activity\_destination.xml:

<androidx.constraintlayout.widget.ConstraintLayout xmlns:andr[oid="htt](http://schemas.android.com/apk/res/android)p:/[/sc](http://schemas.android.com/apk/res/android)he[mas.android.com/apk/res/android"](http://schemas.android.com/apk/res/android)

android:layout\_width="match\_parent" android:layout\_height="match\_parent">

</androidx.constraintlayout.widget.ConstraintLayout>

#### DestinationActivity.java:

package com.example.alarmmanager;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

public class DestinationActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity\_destination);

}

}

#### AlarmReceiver.java:

package com.example.alarmmanager;

import android.app.PendingIntent;

import android.content.BroadcastReceiver; import android.content.Context;

import android.content.Intent;

import androidx.core.app.NotificationCompat;

import androidx.core.app.NotificationManagerCompat;

public class AlarmReceiver extends BroadcastReceiver { @Override

public void onReceive(Context context, Intent intent) {

Intent i = new Intent(context,DestinationActivity.class); intent.setFlags(Intent.FLAG\_ACTIVITY\_NEW\_TASK |

Intent.FLAG\_ACTIVITY\_CLEAR\_TASK);

PendingIntent pendingIntent = PendingIntent.getActivity(context,0,i,0);

NotificationCompat.Builder builder = new NotificationCompat.Builder(context,"sumitsingh")

.setSmallIcon(R.drawable.ic\_launcher\_background)

.setContentTitle("Alarm Manager")

.setContentText("Your alarm has started")

.setStyle(new NotificationCompat.BigTextStyle()

.bigText("Your Alarm has started"))

.setAutoCancel(true)

.setDefaults(NotificationCompat.DEFAULT\_ALL)

.setPriority(NotificationCompat.PRIORITY\_HIGH)

.setContentIntent(pendingIntent);

NotificationManagerCompat notificationManagerCompat = NotificationManagerCompat.from(context);

notificationManagerCompat.notify(123,builder.build());

}

}

#### MainActivity.java:

package com.example.alarmmanager;

import androidx.appcompat.app.AppCompatActivity;

import android.app.AlarmManager; import android.app.NotificationChannel; import android.app.NotificationManager; import android.app.PendingIntent; import android.content.Context;

import android.content.Intent; import android.os.Build; import android.os.Bundle; import android.view.View; import android.widget.Toast;

import com.example.alarmmanager.databinding.ActivityMainBinding; import com.google.android.material.timepicker.MaterialTimePicker; import com.google.android.material.timepicker.TimeFormat;

import java.util.Calendar;

public class MainActivity extends AppCompatActivity {

private ActivityMainBinding binding; private MaterialTimePicker picker; private Calendar calendar;

private AlarmManager alarmManager; private PendingIntent pendingIntent;

@Override

protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState);

binding = ActivityMainBinding.inflate(getLayoutInflater()); setContentView(binding.getRoot()); createNotificationChannel();

binding.selectTimeBtn.setOnClickListener(new View.OnClickListener() { @Override

public void onClick(View v) {

showTimePicker();

}

});

binding.setAlarmBtn.setOnClickListener(new View.OnClickListener() { @Override

public void onClick(View v) {

setAlarm();

}

});

binding.cancelAlarmBtn.setOnClickListener(new View.OnClickListener() { @Override

public void onClick(View v) {

cancelAlarm();

}

});

}

private void cancelAlarm() {

Intent intent = new Intent(this,AlarmReceiver.class);

pendingIntent = PendingIntent.getBroadcast(this,0,intent,0);

if (alarmManager == null){

alarmManager = (AlarmManager) getSystemService(Context.ALARM\_SERVICE);

}

alarmManager.cancel(pendingIntent);

Toast.makeText(this, "Alarm Cancelled", Toast.LENGTH\_SHORT).show();

}

private void setAlarm() {

alarmManager = (AlarmManager) getSystemService(Context.ALARM\_SERVICE);

Intent intent = new Intent(this,AlarmReceiver.class);

pendingIntent = PendingIntent.getBroadcast(this,0,intent,0);

alarmManager.setRepeating(AlarmManager.RTC\_WAKEUP,calendar.getTimeInMillis(), AlarmManager.INTERVAL\_DAY,pendingIntent);

Toast.makeText(this, "Alarm set Successfully", Toast.LENGTH\_SHORT).show();

}

private void showTimePicker() {

picker = new MaterialTimePicker.Builder()

.setTimeFormat(TimeFormat.CLOCK\_12H)

.setHour(12)

.setMinute(0)

.setTitleText("Select Alarm Time")

.build();

picker.show(getSupportFragmentManager(),"sumitsingh");

picker.addOnPositiveButtonClickListener(new View.OnClickListener() { @Override

public void onClick(View v) {

if (picker.getHour() > 12){

binding.selectedTime.setText(

String.format("%02d",(picker.getHour()-12))+" : "+String.format("%02d",picker.getMinute())+" PM"

);

}else {

binding.selectedTime.setText(picker.getHour()+" : " + picker.getMinute() + "

AM");

}

calendar = Calendar.getInstance(); calendar.set(Calendar.HOUR\_OF\_DAY,picker.getHour()); calendar.set(Calendar.MINUTE,picker.getMinute()); calendar.set(Calendar.SECOND,0); calendar.set(Calendar.MILLISECOND,0);

}

});

}

private void createNotificationChannel() {

if (Build.VERSION.SDK\_INT >= Build.VERSION\_CODES.O){

CharSequence name = "alarmReminderChannel"; String description = "Channel For Alarm Manager";

int importance = NotificationManager.IMPORTANCE\_HIGH;

NotificationChannel channel = new NotificationChannel("sumitsingh",name,importance);

channel.setDescription(description);

NotificationManager notificationManager = getSystemService(NotificationManager.class);

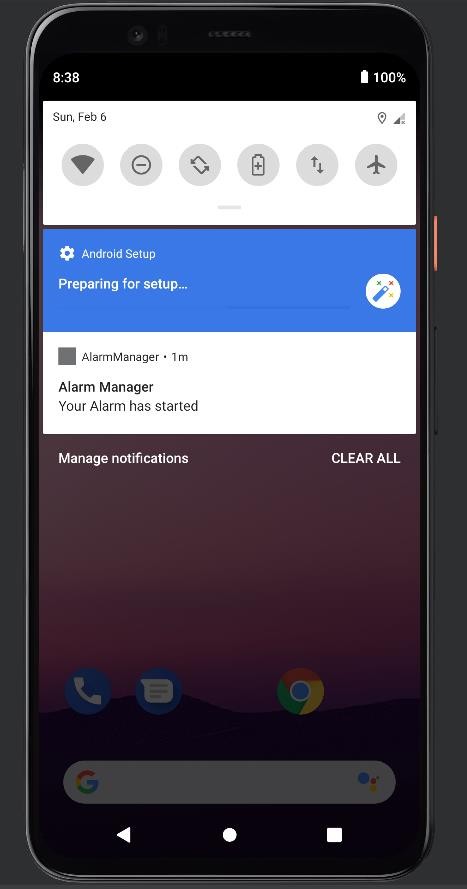
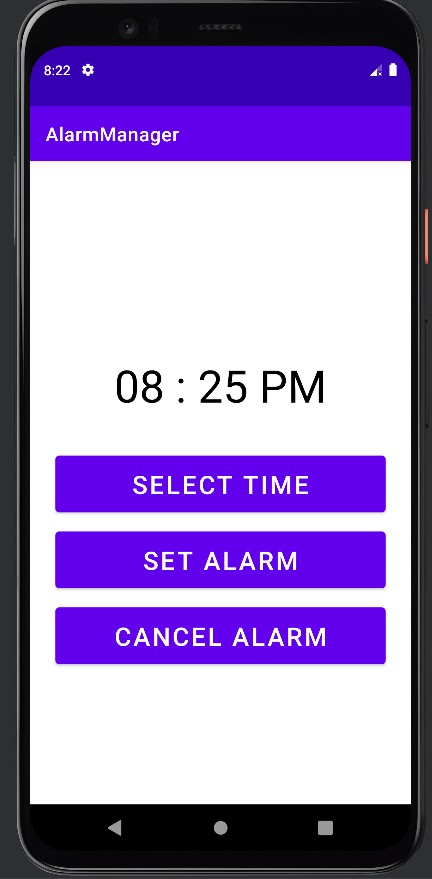
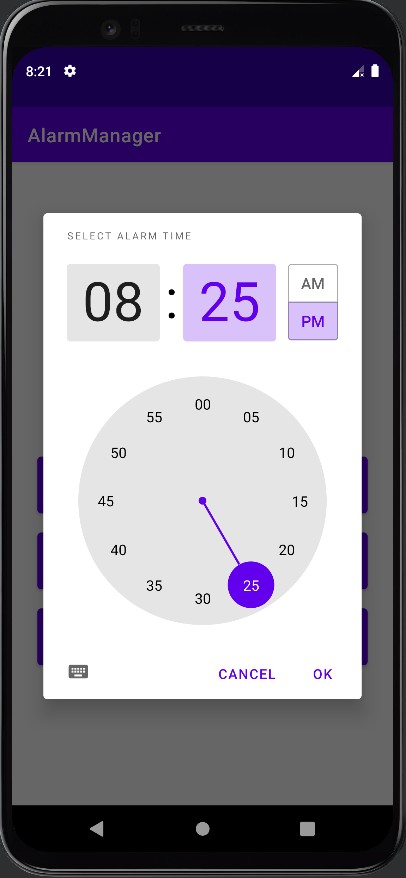
notificationManager.createNotificationChannel(channel);

}

}

}

#### Output:



##### **Conclusion:** Thus we have created an android app to show Notifications and Alarm manager