

BMS Institute of Technology & Management,  
Yelahanka, Bangalore – 560064

## Mobile Application Development (18CSMP68) Lab Manual



**Dr. Surekha K. & Dr. Geeta Patil**

## Course Details

**Course Name** : Mobile Application Development  
**Development Course Code** : 18CSMP68  
**Course prerequisite** : Core Java

## Course Objectives

**Upon completion of this course, students are expected to:**

1. Build an application using Android development environment
2. Experiment with the method of storing, sharing and retrieving the data in Android Applications
3. Examine responsive user interface across wide range of devices
4. Create a mobile Application by using various components like activity, views, services, content providers and receivers

## **SYLLABUS**

### **MOBILE APPLICATION DEVELOPMENT**

**Subject Code: 18CSMP68****IA Marks: 40****No. of Practical Hrs. /Week: 0:0:2****Exam Marks: 60****Total No. of Practical Hrs: 2 Hours/Week****No. of Credits: 02****Descriptions (if any):**

1. The installation procedure of the Android Studio/Java software must be demonstrated and carried out in groups.
2. Students should use the latest version of Android Studio/Java/ Kotlin to execute these programs. Diagrams given are for representational purposes only, students are expected to improvise on them.
3. Part B programs should be developed as an application and are to be demonstrated as a mini project in a group by adding extra features or the students can also develop their application and demonstrate it as a mini-project. (Projects/programs are not limited to the list given in Part B).

## **PART A**

**Program 1**

Create an application to design a Visiting Card. The Visiting card should have a company logo at the top right corner. The company name should be displayed in Capital letters, aligned to the center. Information like the name of the employee, job title, phone number, address, email, fax and the website address is to be displayed. Insert a horizontal line between the job title and the phone number.

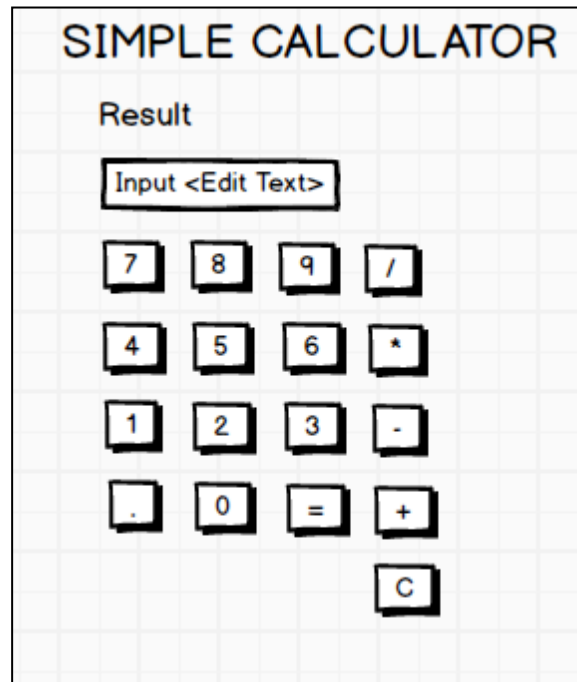
COMPANY NAME 

---

Name  
Job Title  
Phone Number  
Address  
Email, website, fax details

**Program 2**

Develop an Android application using controls like Button, TextView, EditText for designing a Calculator having basic functionality like Addition, Subtraction, Multiplication, and Division.



### Program 3

Create a SIGN Up activity with Username and Password. Validation of password should happen based on the following rules:

- Password should contain uppercase and lowercase letters.
- Password should contain letters and numbers.
- Password should contain special characters.
- Minimum length of the password (the default value is 8).

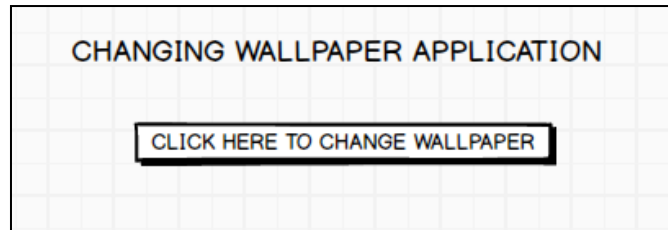
On successful **SIGN UP** proceed to the next Login activity. Here the user should **SIGN IN** using the Username and Password created during signup activity. If the Username and Password are matched then navigate to the next activity which displays a message saying “Successful Login” or else display a toast message saying “Login Failed”. The user is given only two attempts and after that display a toast message saying “Failed Login Attempts” and disable the SIGN IN button. Use Bundle to transfer information from one activity to another.

The mockup for the SIGNUP ACTIVITY shows a title bar at the top. Below it, there are two input fields: one for 'Username:' and one for 'Password:'. At the bottom of the form is a button labeled 'SIGN UP'.

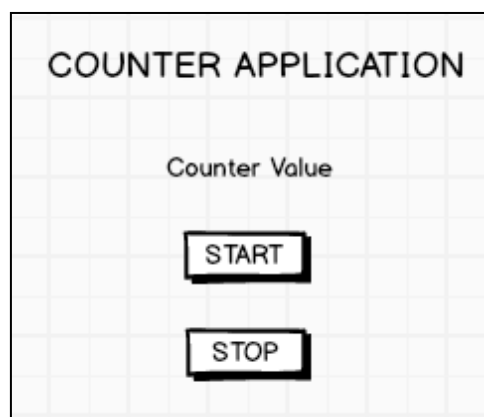
The mockup for the LOGIN ACTIVITY shows a title bar at the top. Below it, there are two input fields: one for 'Username:' and one for 'Password:'. At the bottom of the form is a button labeled 'SIGN IN'.

**Program 4**

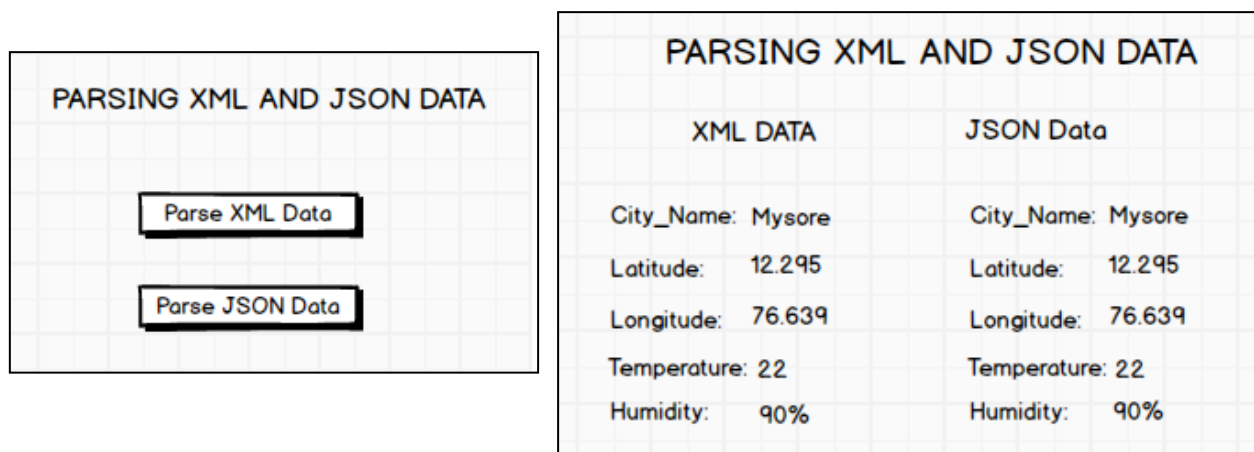
Develop an application to set an image as wallpaper. On click of a button, the wallpaper image should start to change randomly every 30 seconds.

**Program 5**

Write a program to create an activity with two buttons START and STOP. On Pressing of the START button, the activity must start the counter by displaying the numbers from One and the counter must keep on counting until the STOP button is pressed. Display the counter value in a TextView control.

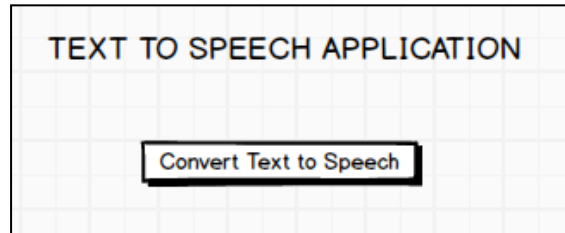
**Program 6**

Create two files of XML and JSON type with values for City\_Name, Latitude, Longitude, Temperature, and Humidity. Develop an application to create an activity with two buttons to parse the XML and JSON files which when clicked should display the data in their respective layouts side by side.



**Program 7**

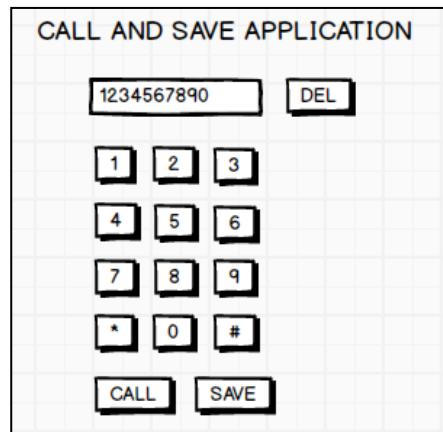
Develop a simple application with one Edit Text so that the user can write some text in it. Create a button called “Convert Text to Speech” that converts the user input text into voice.



The screenshot shows a mobile application interface titled "TEXT TO SPEECH APPLICATION". It features a single text input field with a light blue background and a black border. Below the input field is a button labeled "Convert Text to Speech".

**Program 8**

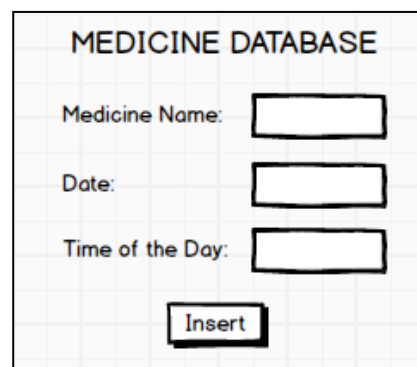
Create an activity like a phone dialer with CALL and SAVE buttons. On pressing the CALL button, it must call the phone number and on pressing the SAVE button it must save the number to the phone contacts.



The screenshot shows a mobile application interface titled "CALL AND SAVE APPLICATION". It features a text input field containing the number "1234567890" and a "DEL" button to its right. Below the input field is a numeric keypad with buttons for digits 1 through 9, 0, \*, and #. At the bottom of the interface are two buttons labeled "CALL" and "SAVE".

**PART B****Program 1**

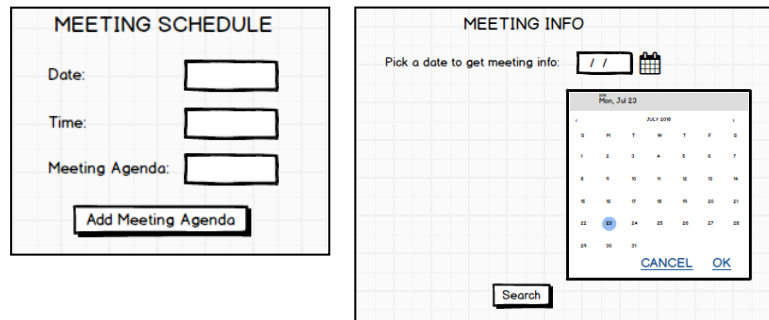
Write a program to enter Medicine Name, Date and Time of the Day as input from the user and store it in the SQLite database. Input for Time of the Day should be either Morning or Afternoon or Evening or Night. Trigger an alarm based on the Date and Time of the Day and display the Medicine Name.



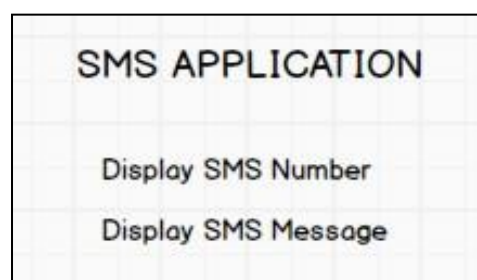
The screenshot shows a mobile application interface titled "MEDICINE DATABASE". It features three text input fields labeled "Medicine Name:", "Date:", and "Time of the Day:". Below these fields is a button labeled "Insert".

**Program 2**

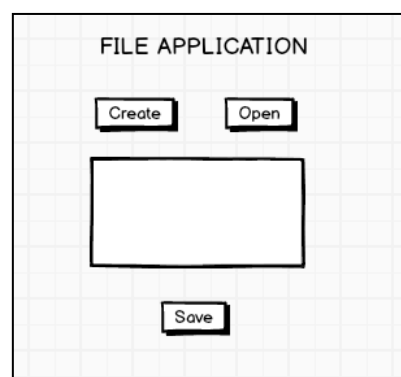
Develop a content provider application with an activity called “Meeting Schedule” which takes Date, Time and Meeting Agenda as input from the user and store this information into the SQLite database. Create another application with an activity called “Meeting Info” having DatePicker control, which on the selection of a date should display the Meeting Agenda information for that particular date, else it should display a toast message saying “No Meeting on this Date”.

**Program 3**

Create an application to receive an incoming SMS which is notified to the user. On clicking this SMS notification, the message content and the number should be displayed on the screen. Use appropriate emulator control to send the SMS message to your application.

**Program 4**

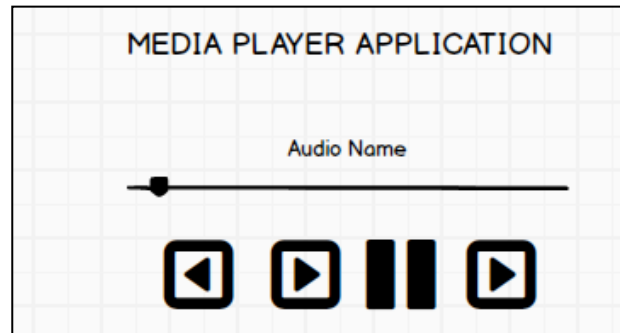
Write a program to create an activity having a Text box, and also Save, Open and Create buttons. The user has to write some text in the Text box. On pressing the Create button the text should be saved as a text file in Mksdcard. On subsequent changes to the text, the Save button should be pressed to store the latest content to the same file. On pressing the Open button, it should display the contents from the previously stored files in the Text box. If the user tries to save the contents in the Textbox to a file without creating it, then a toast message has to be displayed saying “FirstCreate a File”.



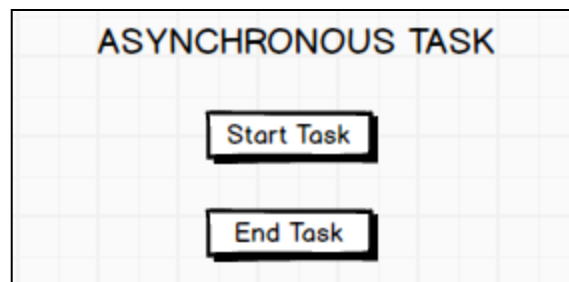


**Program 5**

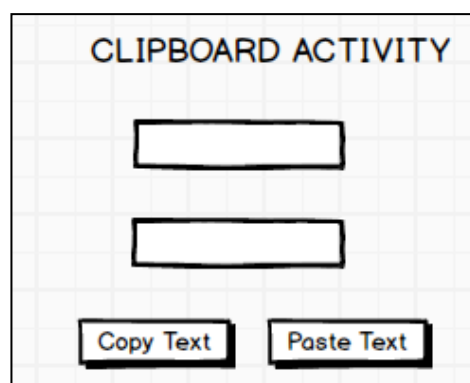
Create an application to demonstrate a basic media player that allows the user to Forward, Backward, Play and Pause an audio. Also, make use of the indicator in the seek bar to move the audio forward or backward as required.

**Program 6**

Develop an application to demonstrate the use of Asynchronous tasks in android. The asynchronous task should implement the functionality of a simple moving banner. On pressing the **Start Task** button, the banner message should scroll from right to left. On pressing the **Stop Task** button, the banner message should stop. Let the banner message be "Demonstration of Asynchronous Task".

**Program 7**

Develop an application that makes use of the clipboard framework for copying and pasting of the text. The activity consists of two EditText controls and two Buttons to trigger the copy and paste functionality.



**Program 8**

Create an AIDL service that calculates Car Loan EMI. The formula to calculate EMI is

$$E = P * (r(1+r)^n)/((1+r)^n - 1)$$

where

E = The EMI payable on the car loan amount

P = The Car loan Principal Amount

r = The interest rate value computed on a monthly basis

n = The loan tenure in the form of months

The down payment amount has to be deducted from the principal amount paid towards buying the Car. Develop an application that makes use of this AIDL service to calculate the EMI. This application should have four EditText to read the Principal Amount, Down Payment, Interest Rate, Loan Term (in months) and a button named as “Calculate Monthly EMI”. On click of this button, the result should be shown in a TextView. Also, calculate the EMI by varying the Loan Term and Interest Rate values.

The screenshot shows a mobile application titled "CAR EMI CALCULATOR". It has a light gray background with a grid pattern. There are four input fields, each preceded by a label: "Principal Amount:", "Down Payment:", "Interest Rate:", and "Loan Term (in months):". Below these fields is a button labeled "Calculate Monthly EMI". To the right of the input fields, the text "EMI: Result" is displayed, indicating where the calculated EMI value will be shown.

## Course Outcomes

After successful completion of the Course, the participants will be able to

<b>18CSMP68.1</b>	Build an application using Android development environment
<b>18CSMP68.2</b>	Experiment with the method of storing, sharing and retrieving the data in Android Applications
<b>18CSMP68.3</b>	Examine responsive user interface across wide range of devices
<b>18CSMP68.4</b>	Create a mobile Application by using various components like activity, views, services, content providers and receivers

## CO-PO Mapping

CO No.	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
<b>18CSMP68.1</b>	3											
<b>18CSMP68.2</b>		3										
<b>18CSMP68.3</b>		3										
<b>18CSMP68.4</b>			3									

## Procedure to Conduct Practical Examination

### Experiment distribution

- For laboratories having only one part: Students are allowed to pick one experiment from the lot with equal opportunity.
- For laboratories having PART A and PART B: Students are allowed to pick one experiment from PART A and one experiment from PART B, with equal opportunity.

**Change of experiment is allowed only once and marks allotted for procedure to be made zero of the changed part only.**

### Marks Distribution (Courseed to change in accordance with university regulations)

- For laboratories having only one part –  
Procedure + Execution + Viva-Voce: 15+70+15= 100 Marks
- For laboratories having PART A and PART B
  - Part A – Procedure + Execution + Viva = 6 + 28 + 6 = 40 Marks
  - Part B – Procedure + Execution + Viva = 9 + 42 + 9 = 60 Marks

# 1. Android Studio Tutorials

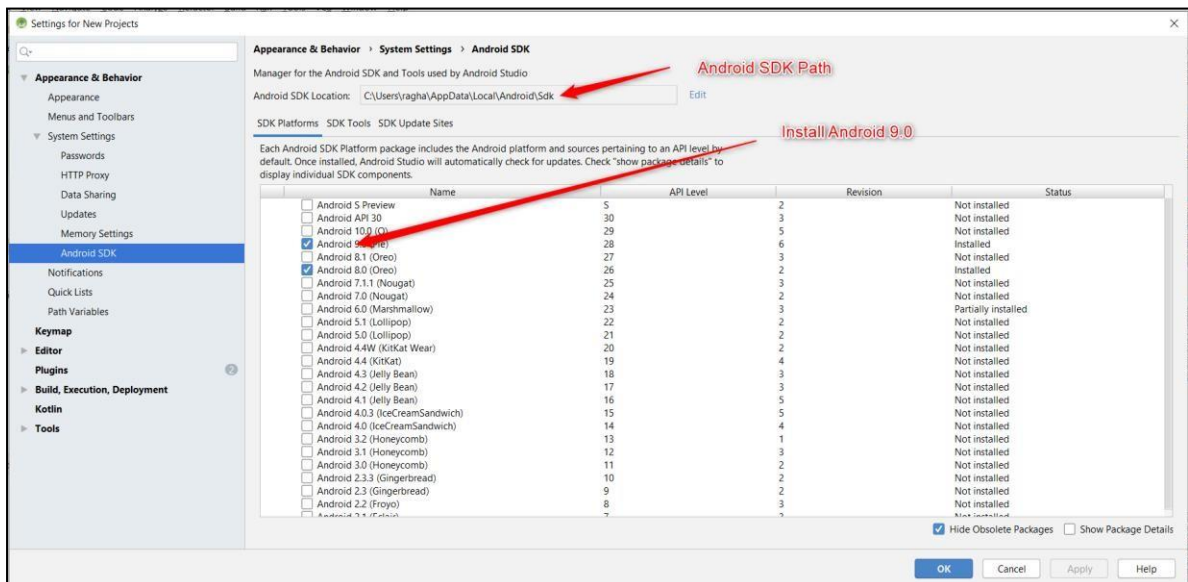
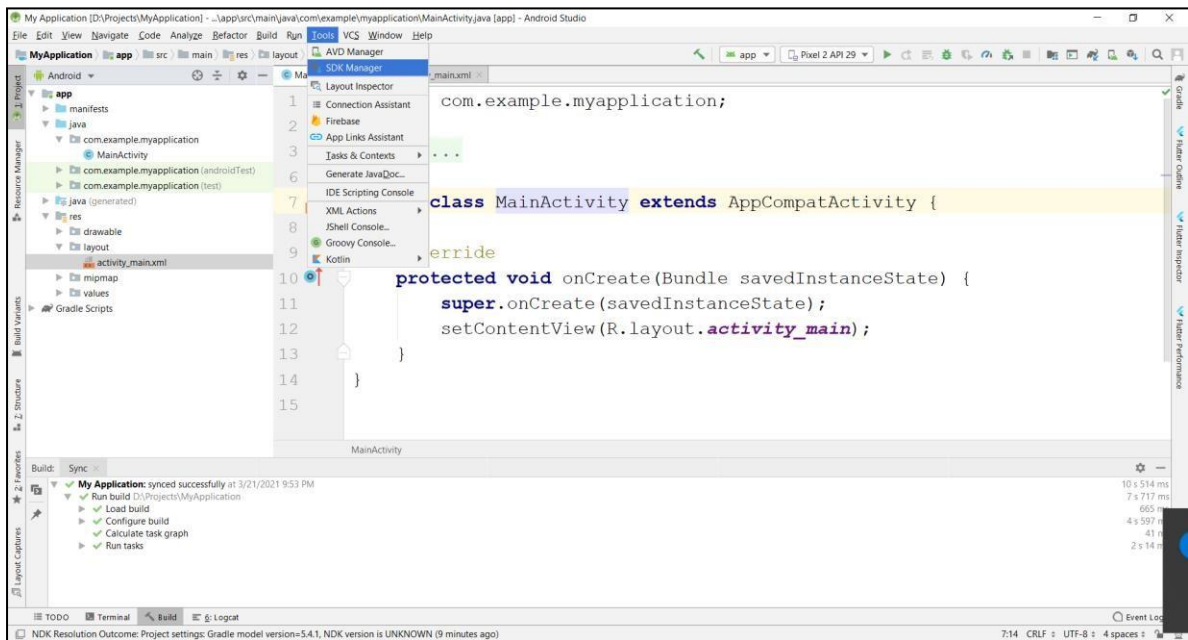
## 1.1 Install Android Studio and Packages:

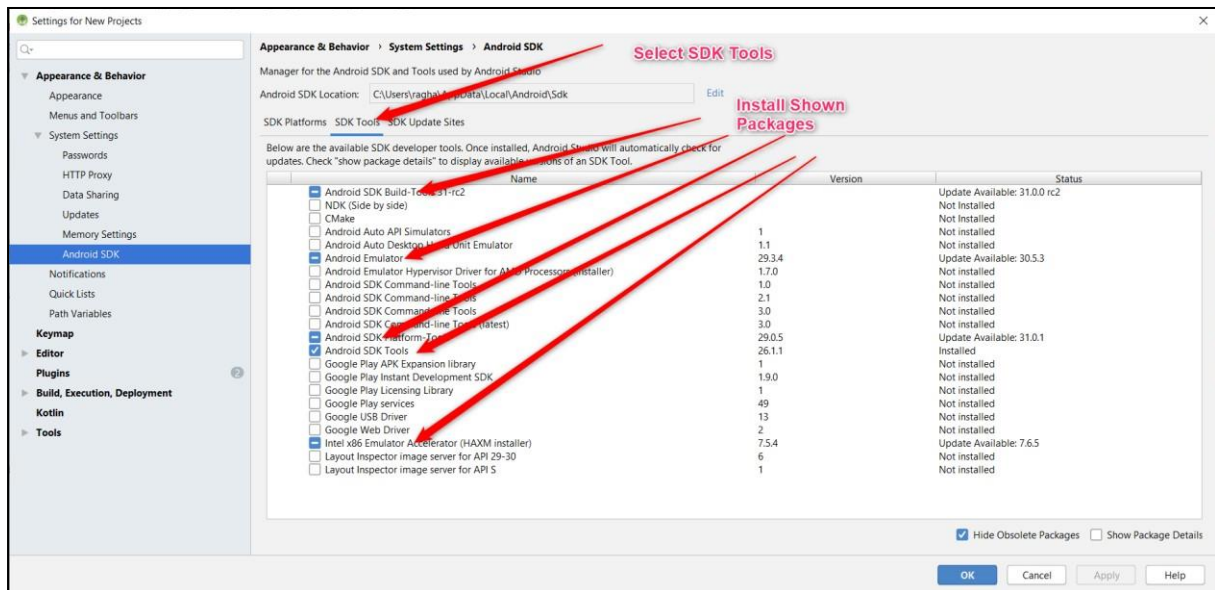
Download Android Studio from the below link :

<https://developer.android.com/studio>

## 1.2 Configure Android SDK packages:

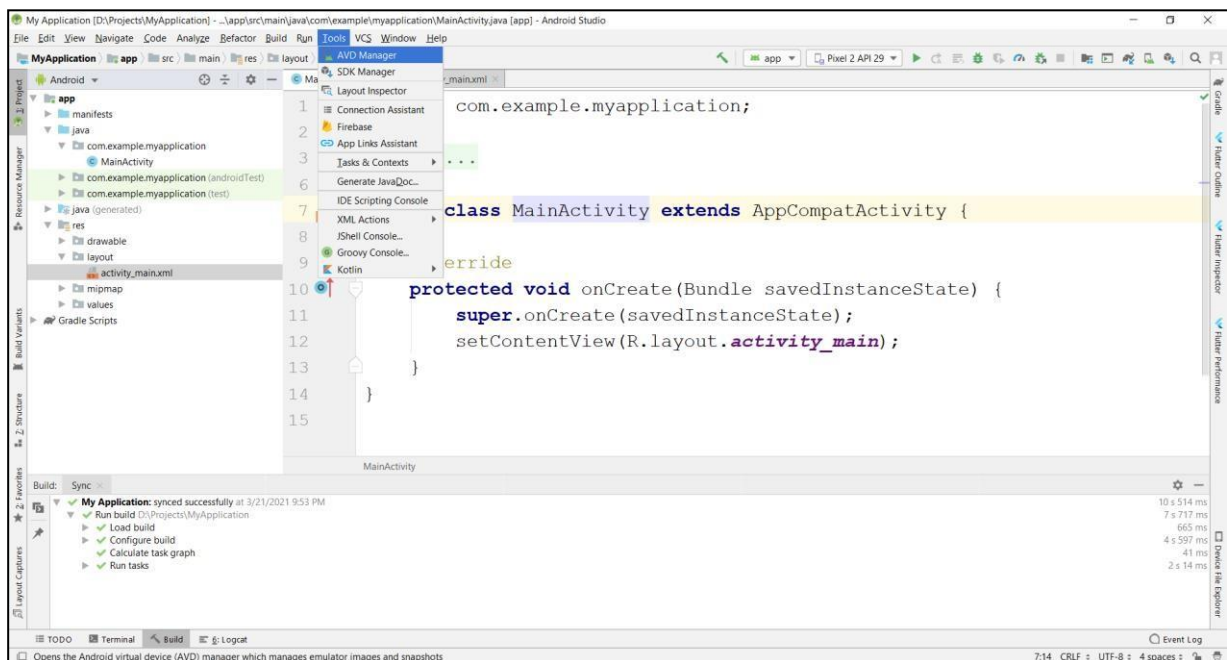
Go to Tools → SDK Manager

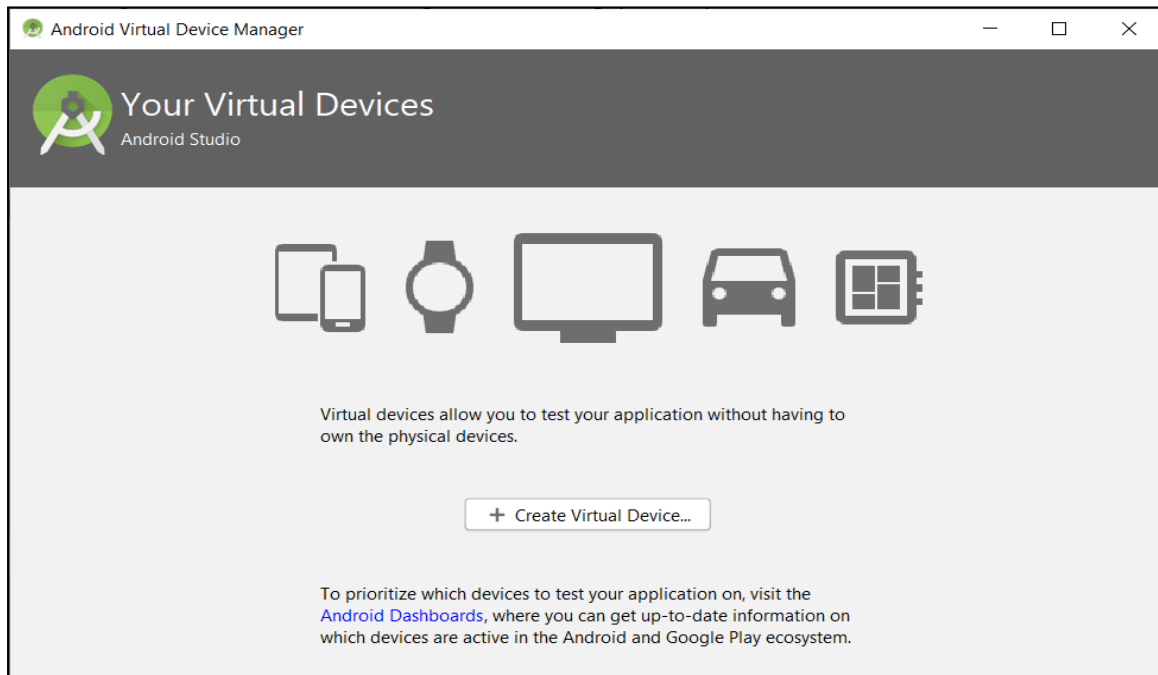




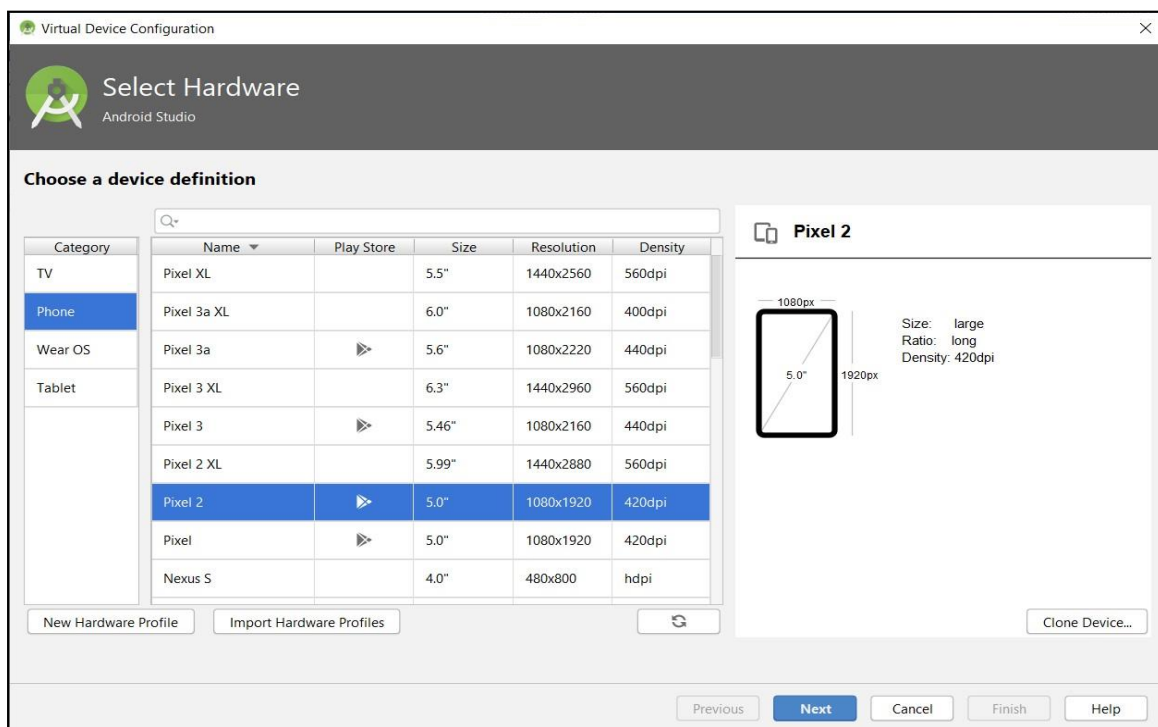
### 1.3 Creating Emulator

Go to Tools → Select AVD Manager

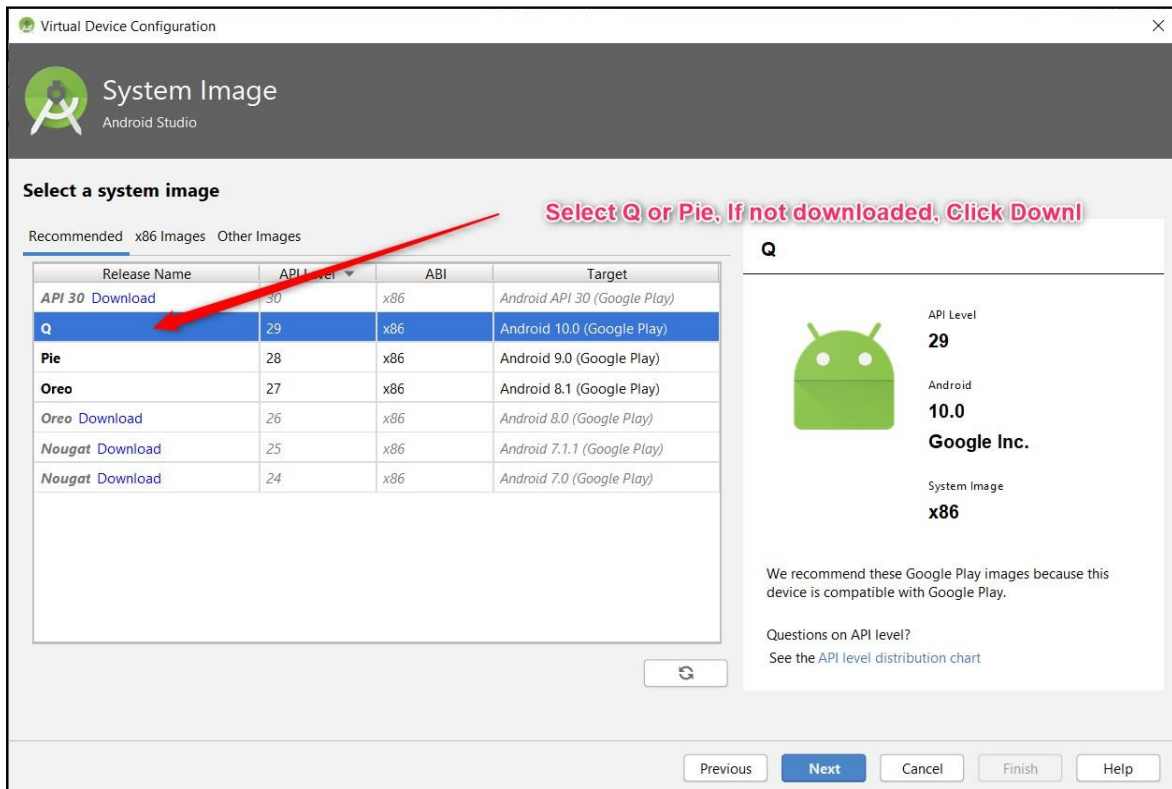




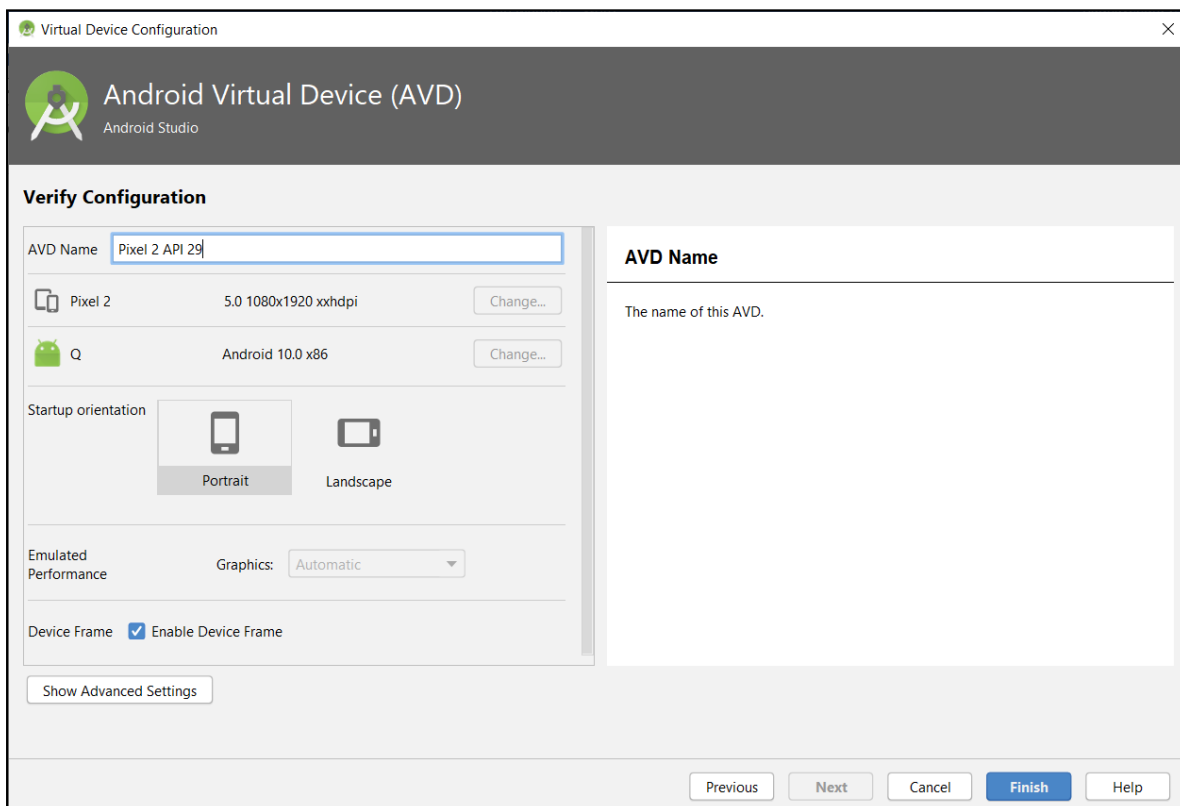
Select Create Virtual Device → Select Phone → Pixel 2 → Press Next



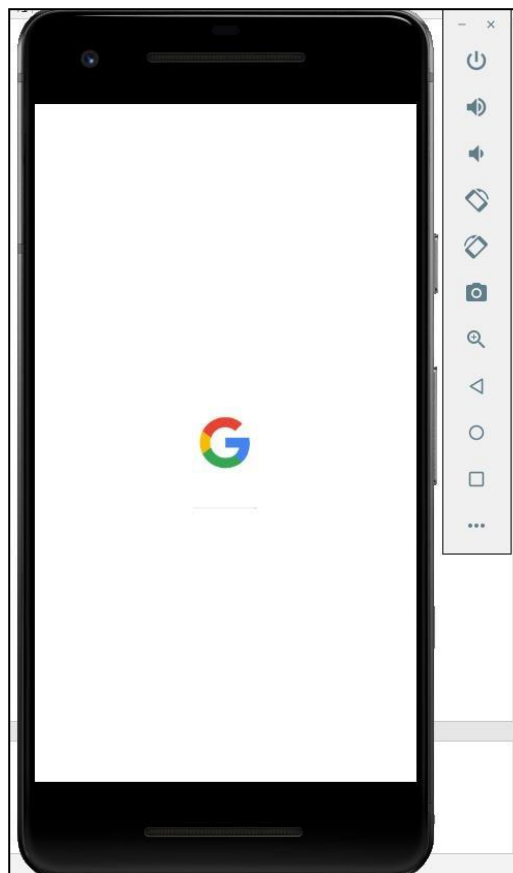
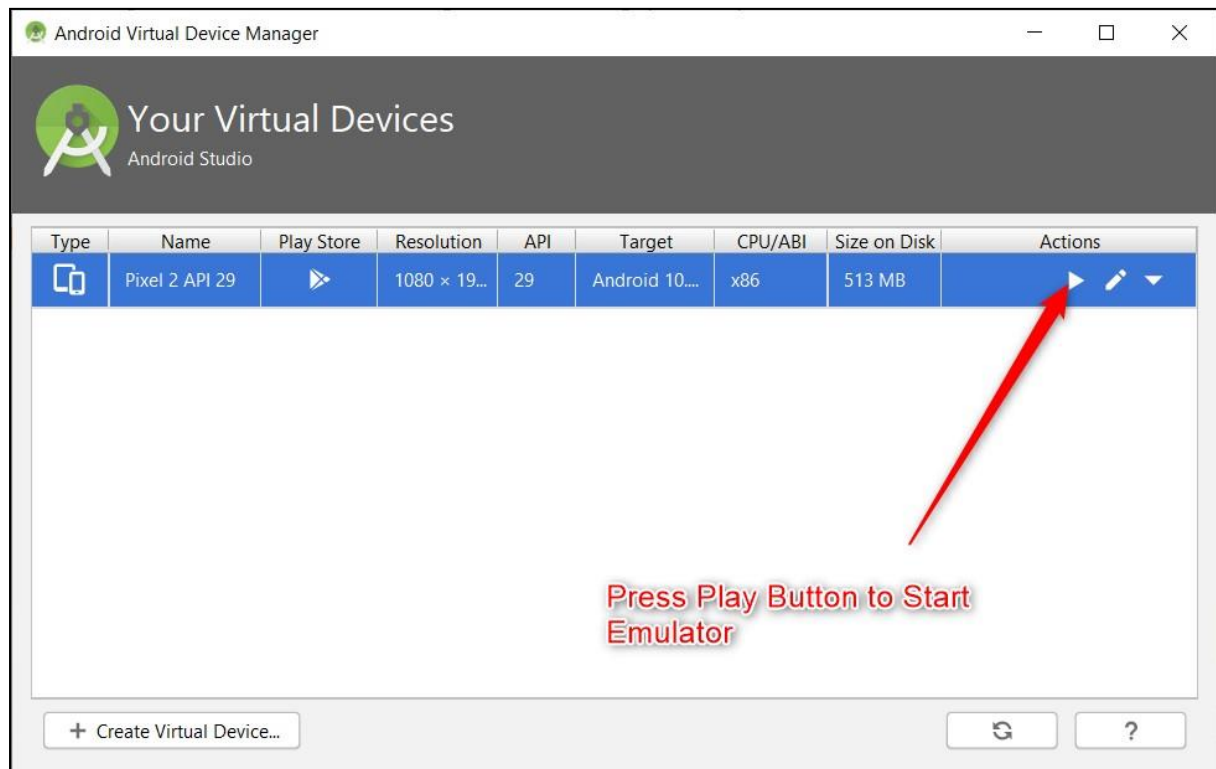
Select Android Q, if not already downloaded press download, After download completes Select Q and Press Next Button.



Enter AVD Name and Press Finish.



Press Play Button to Start Emulator

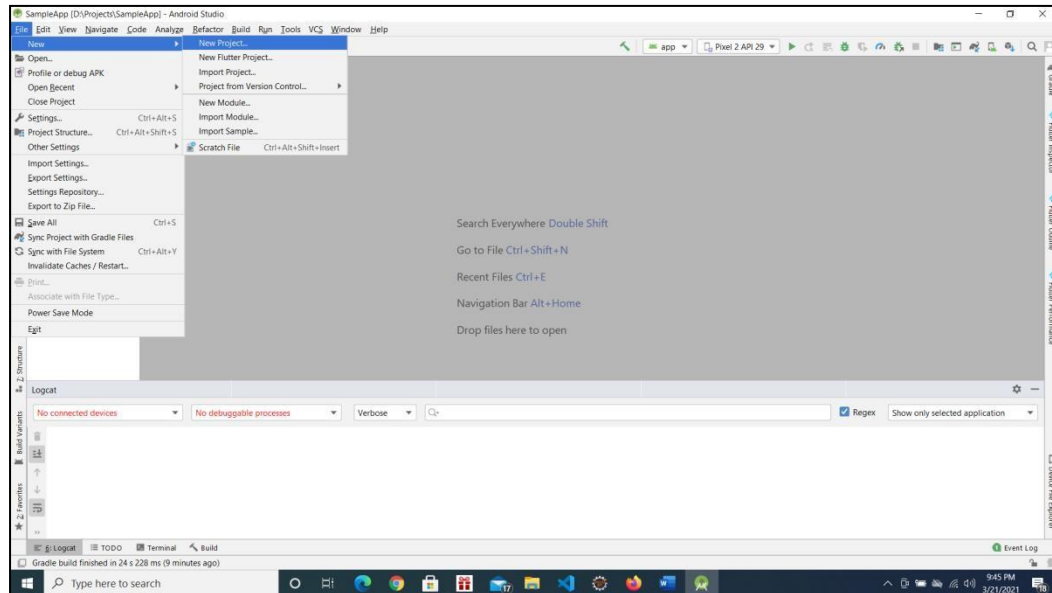




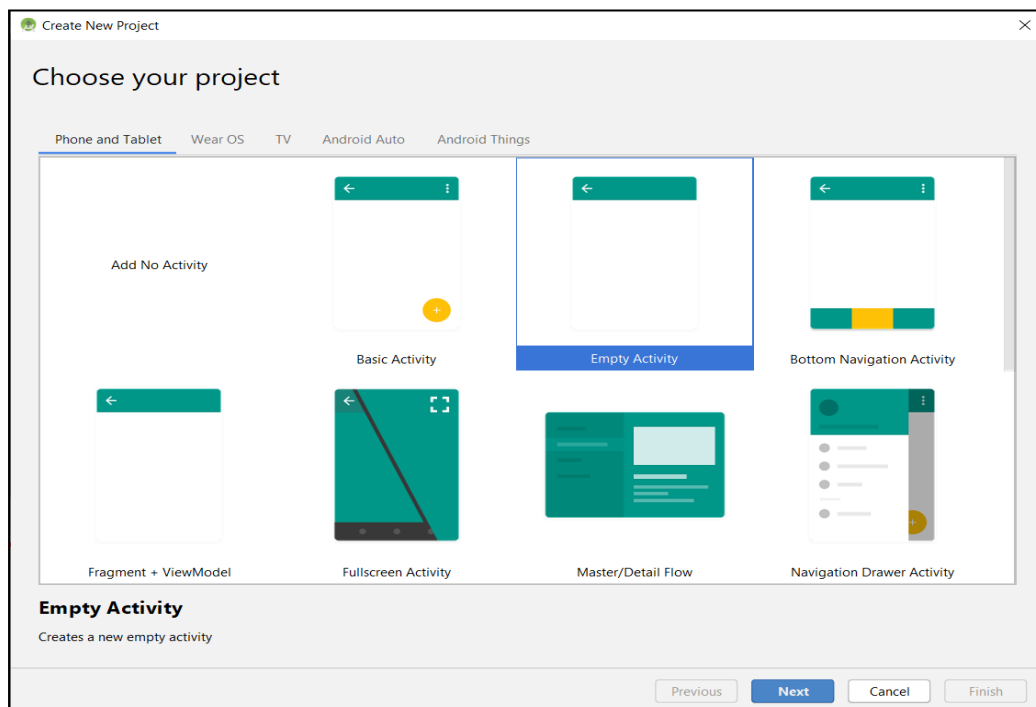
## 1.4 Creating a New Project in Android

While creating a New Project for First Time, make sure Android Studio is connected to internet. It downloads the required packages from internet.

Go to File → New → New Project



Choose Phone and Tablet → Empty Activity → Press Next



In Configure your Project Screen, Enter below details and Press Finish Button.

Enter Name of the Application → This will be application name this will be visible with Home Screen Icon.

Package Name → Enter package name atleast two identifier (Eg: com.example). Best Practice is 3 or more identifier (Eg: com.example.firstapp).

Save Location → Location where to save the Project

Language → Choose Java

Minimum API Level → Android 5.0

Select Checkbox Use androidx.artifacts folder as below screenshot.

Create New Project

Configure your project

Name  
My Application

Package name  
com.example.myapplication

Save location  
D:\Projects\MyApplication

Language  
Java

Minimum API level  
API 21: Android 5.0 (Lollipop)

Empty Activity

Creates a new empty activity

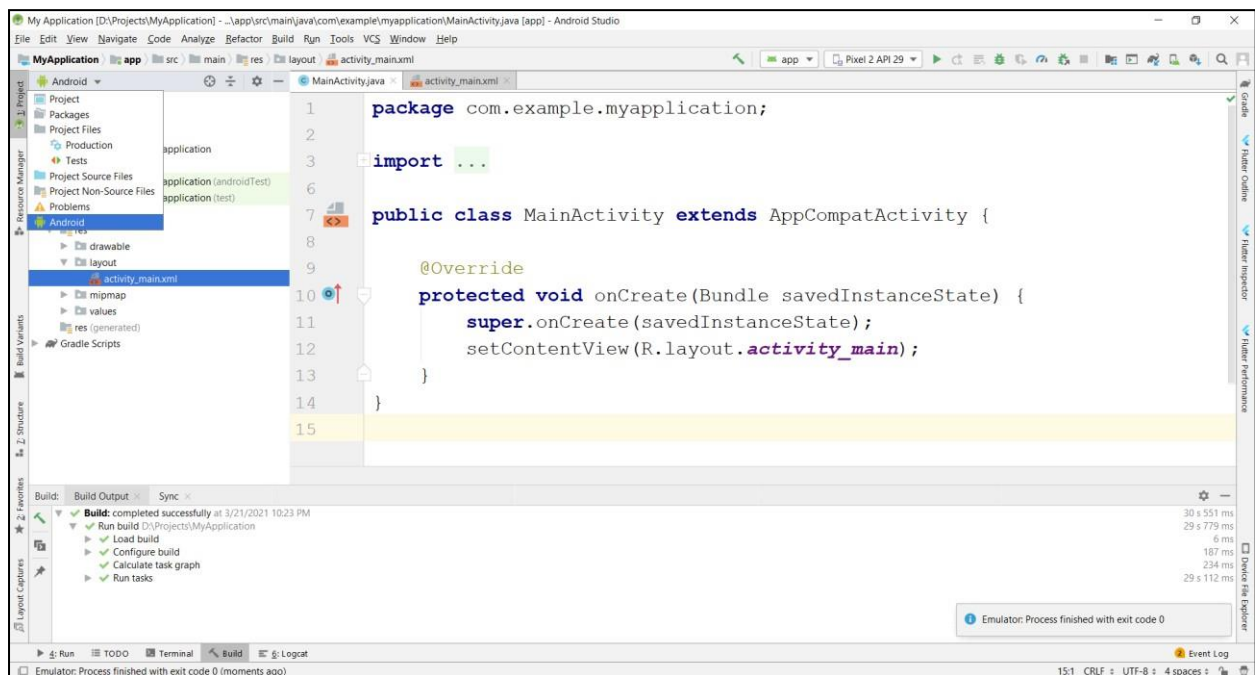
Information: Your app will run on approximately 94.1% of devices.  
[Help me choose](#)

☐ This project will support instant apps

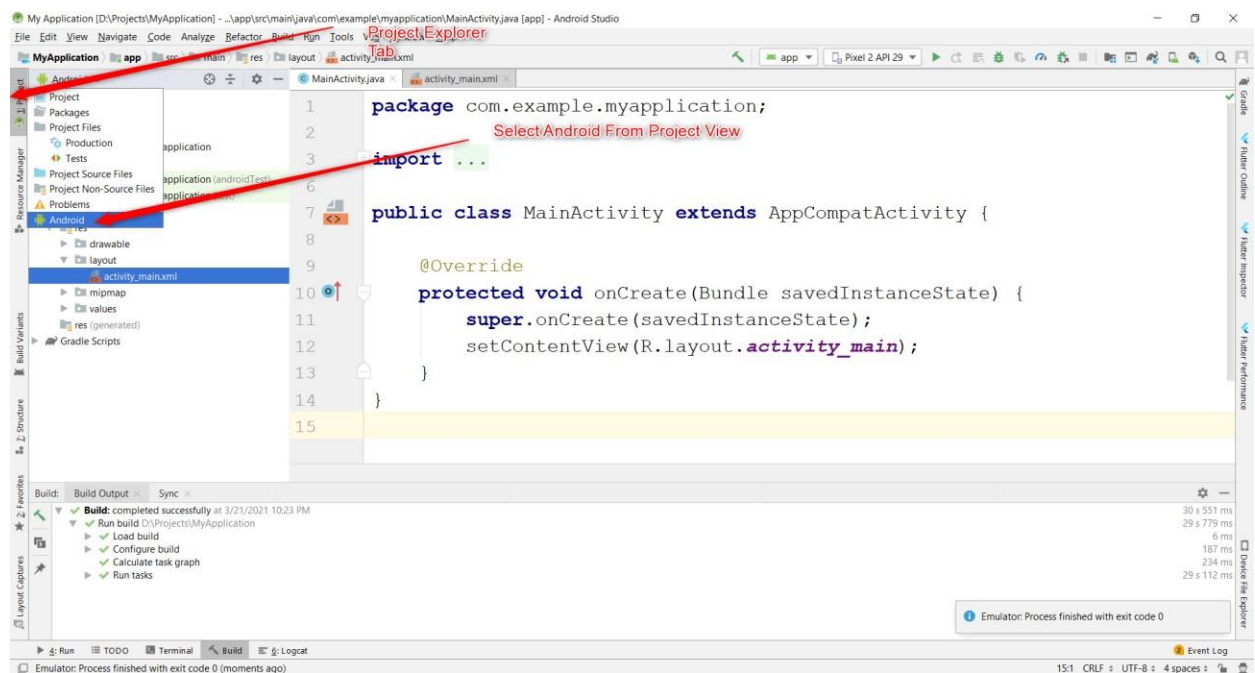
☒ Use androidx.\* artifacts

Previous Next Cancel Finish

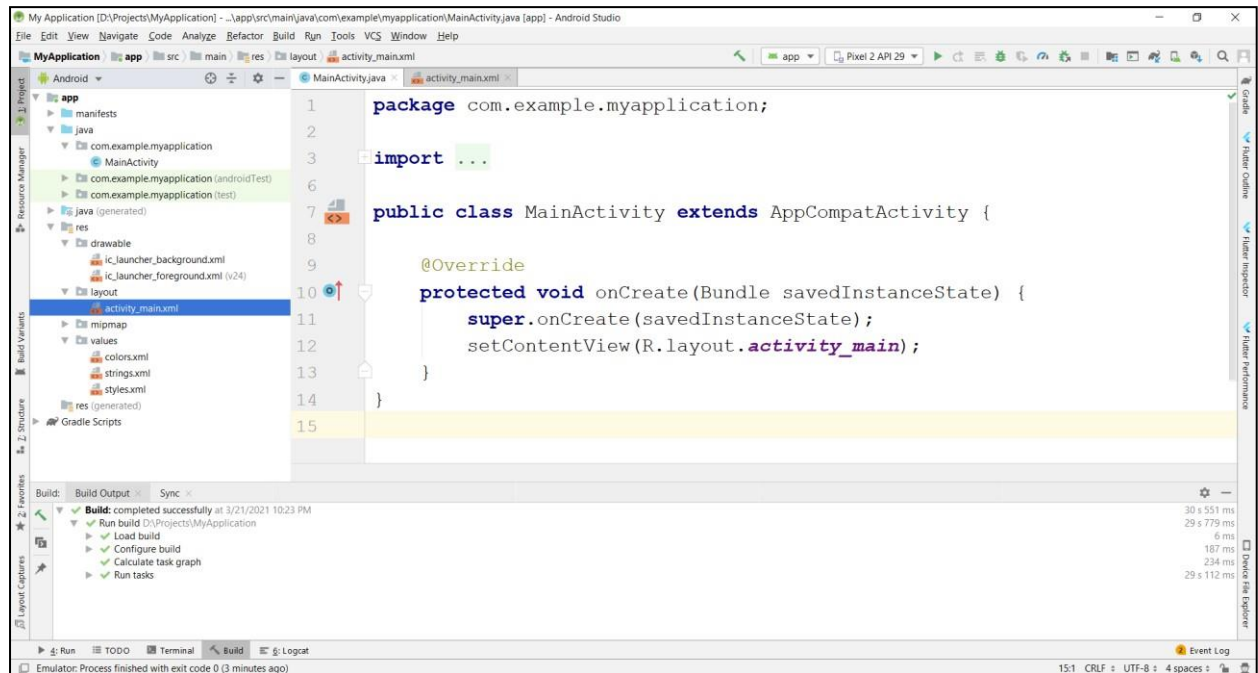
## 1.5 Android Project Structure:



Select Project Explorer and Select Android from Project View

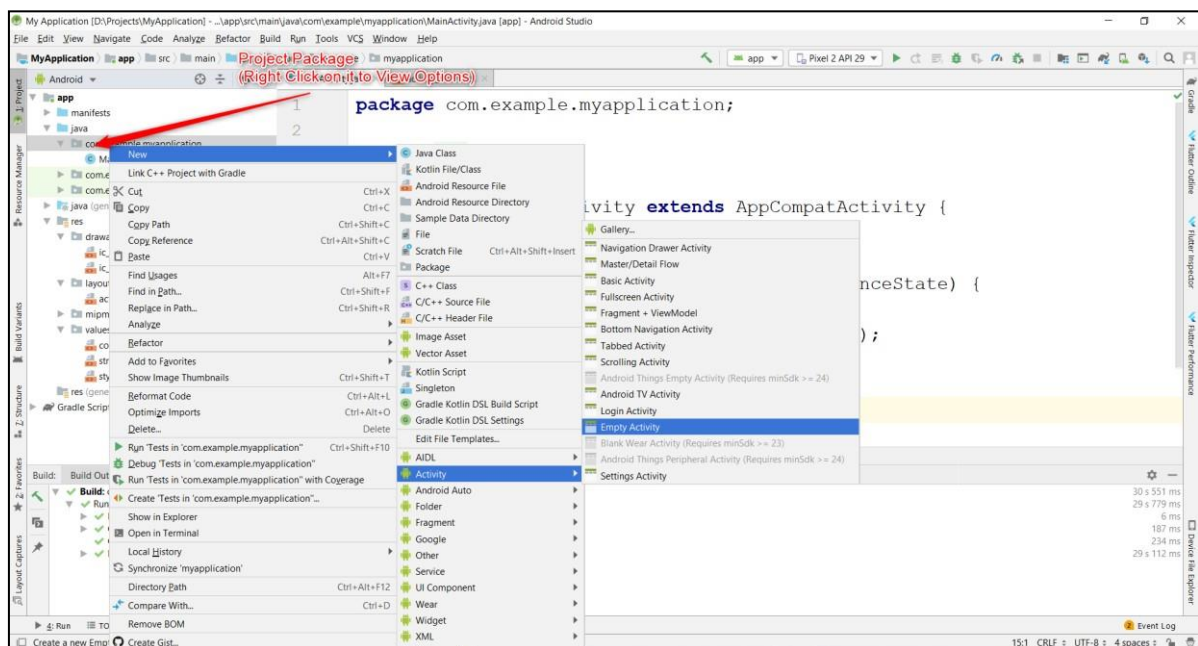


## Basic View:

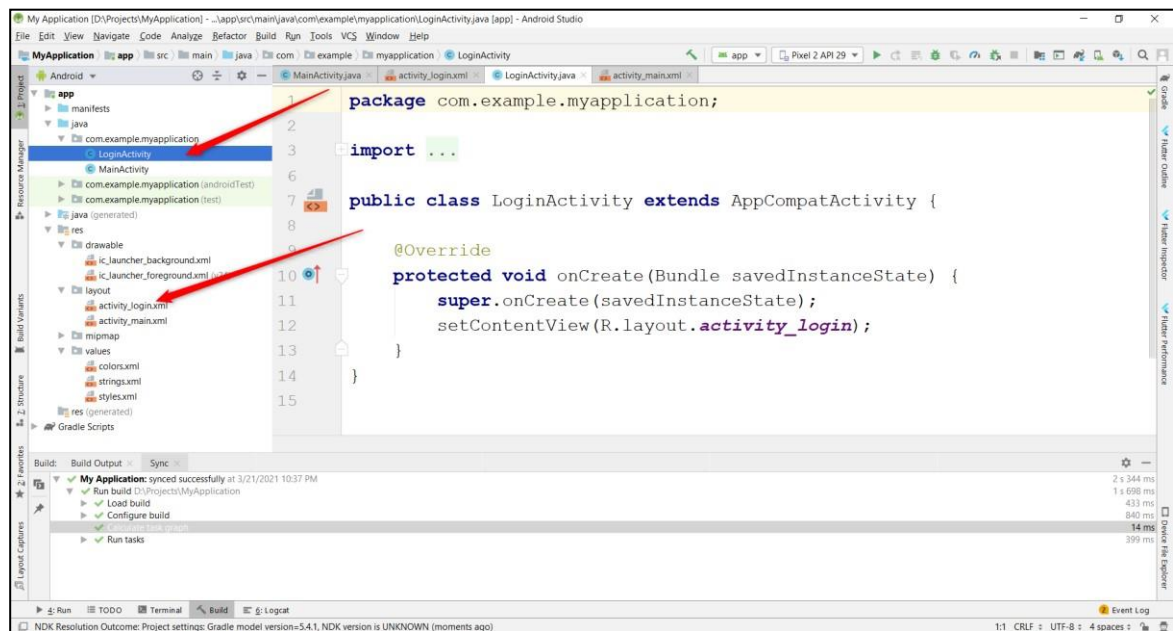
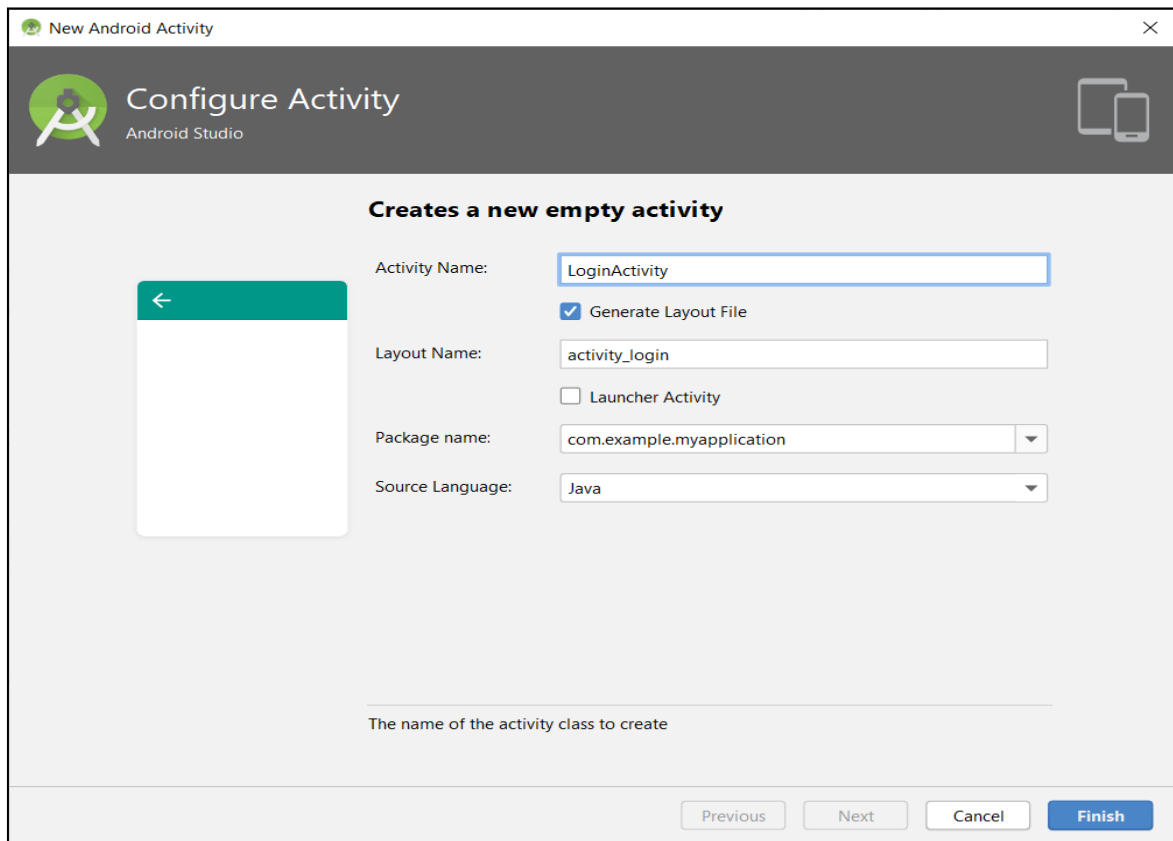


## 2. Creating an Activity in Android

Right Click on Package → New → Activity → Empty Activity

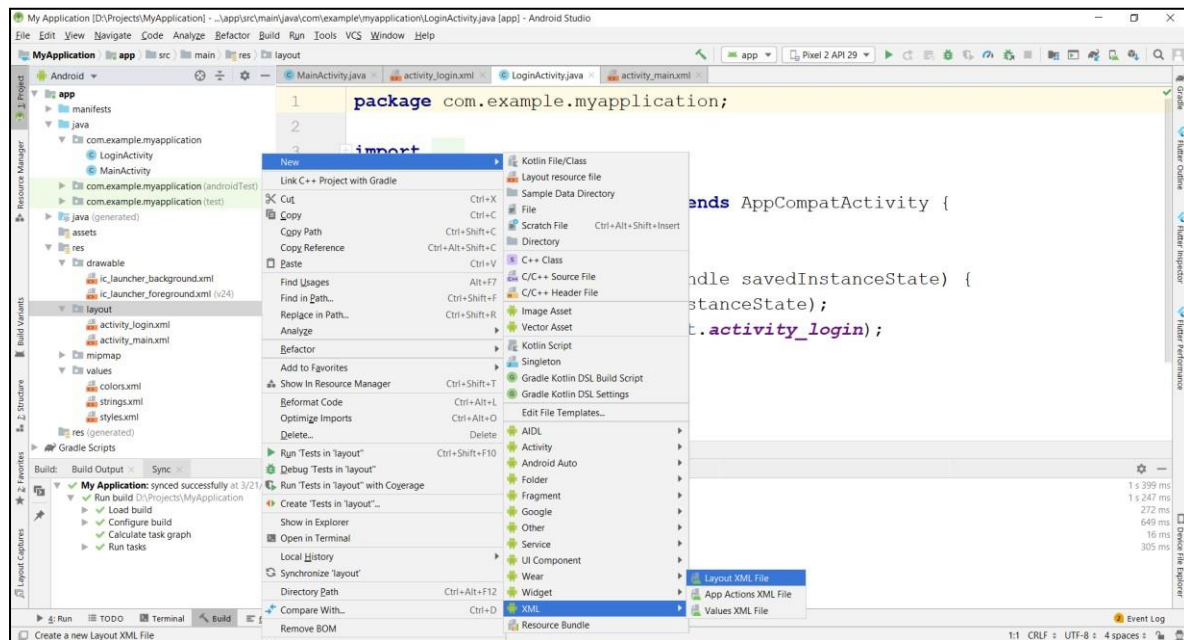


Enter Activity Name and Press Finish

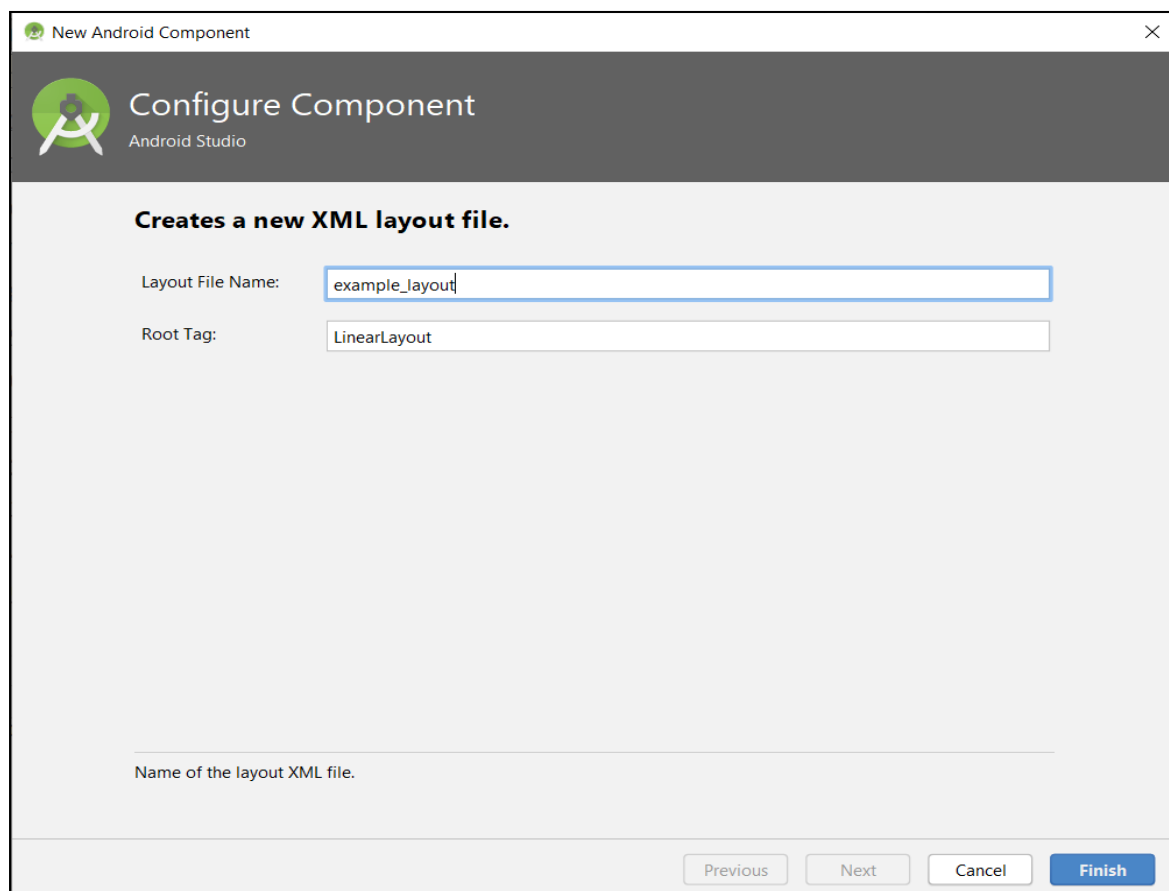


## 1.6 Creating a Layout in Android

Right Click on Layout Folder → New → XML → Layout XML File

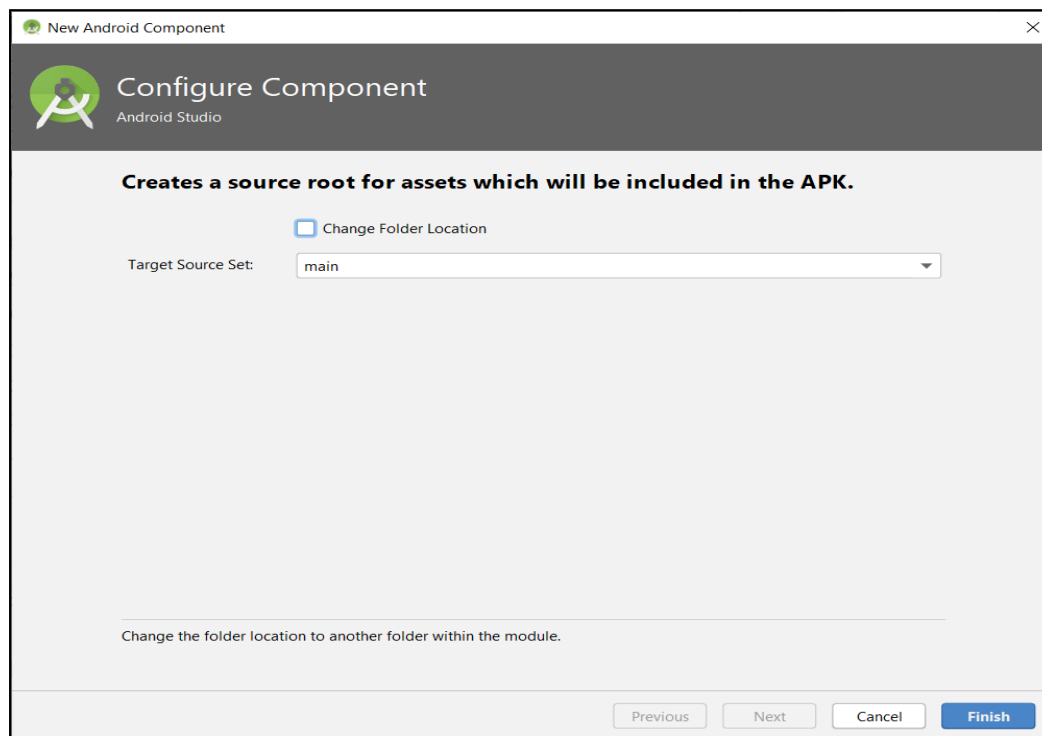
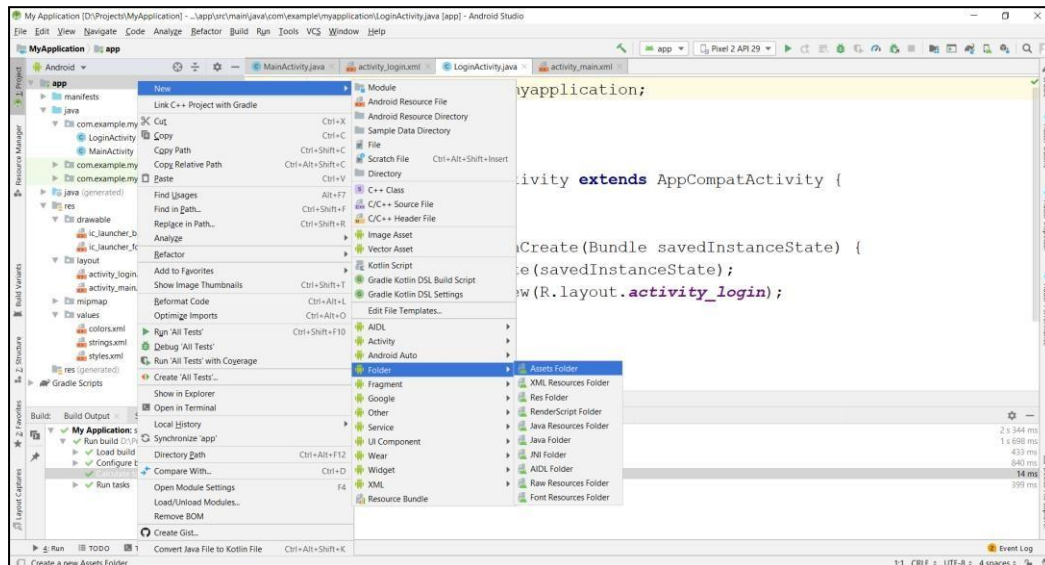


Enter xml file name and press Finish

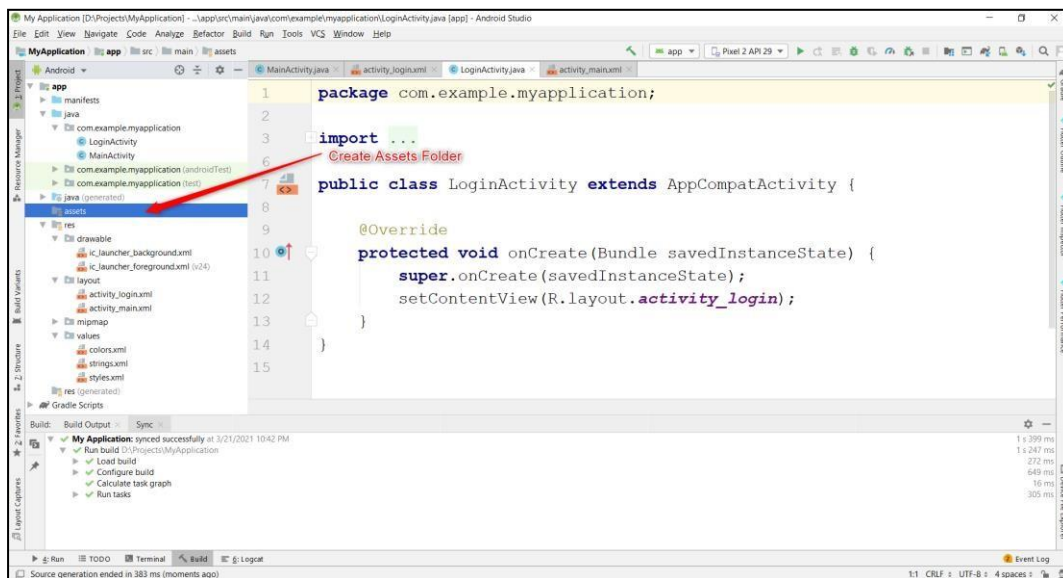


## 1.7 Creating Assets Folder in Android

Right Click on app folder → New → Folder → Assets Folder → Press Finish Button

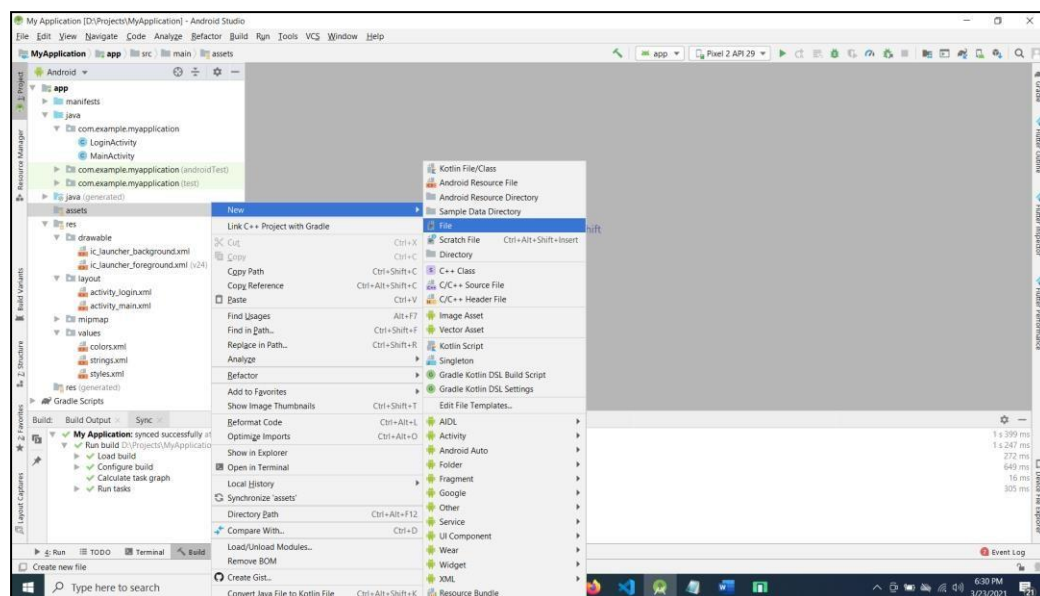




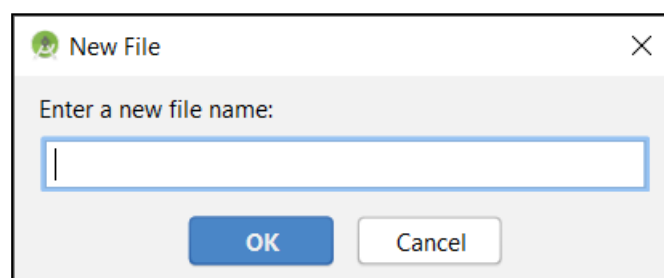


## 1.8 Creating File in assets Folder:

Right Click on assets folder → New → File



Enter filename with extension (Eg: abc.xml)

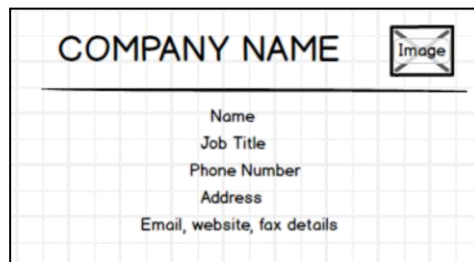




## Programs PART A


### Program 1

Create an application to design a Visiting Card. The Visiting card should have a company logo at the top right corner. The company name should be displayed in Capital letters, aligned to the center. Information like the name of the employee, job title, phone number, address, email, fax and the website address is to be displayed. Insert a horizontal line between the job title and the phone number.



1. Create a New Android Project with Empty Activity.
2. Open activity\_main.xml file from res → layout folder, check/add Linear Layout as the root view.
3. Create layout using nested Relative Layout and TextView.
4. Use View background property to draw the line
5. Add Image to drawable folder and reference the image in the layout using @drawable/<image\_name>
6. Use android:layout\_gravity/android:gravity properties to center the components.

## Design

BMS INSTITUTE OF TECHNOLOGY AND MANAGEMENT	
<b>Dr. Geeta Patil</b>	
Associate Professor	
+91-97641-123424	
Bangalore	
Email:geetapatil@bmsit.in	
Website:https://bmsit.ac.in	

## activity\_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:paddingLeft="20dp"
    android:paddingTop="30dp"
    android:paddingRight="20dp"
    tools:context=".MainActivity">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="70dp">

        <TextView
            android:id="@+id/textView"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginStart="-45dp"
            android:layout_marginBottom="20dp"
            android:gravity="center"
            android:text="BMS Institute of Technology And Management"
            android:textAllCaps="true"
            android:textColor="#E61717"
            android:textSize="17sp" />

        <ImageView
            android:id="@+id/imageView4"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginStart="-61dp"
            android:layout_marginBottom="20dp"
            android:layout_toRightOf="@id/textView"
            app:srcCompat="@drawable/logo" />

    </RelativeLayout>

    <View
        android:layout_width="match_parent"
        android:layout_height="2dp"
        android:background="#000000"
    />

    <TextView
        android:layout_width="match_parent"
```

```
android:layout_height="wrap_content"
android:text="Dr. Geeta Patil"
android:textStyle="bold"
android:textSize="20dp"
android:layout_marginBottom="10dp"
android:layout_marginTop="20dp"
android:textColor="#000000"
android:gravity="center" />
```

<TextView

```
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="Associate Professor"
android:textSize="18dp"
android:layout_marginBottom="10dp"
android:layout_marginTop="10dp"
android:textColor="#000000"
android:gravity="center" />
```

<View

```
android:layout_width="match_parent"
android:layout_height="2dp"
android:background="#000000" />
```

<TextView

```
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="+91-97641-123424"
android:textSize="18dp"
android:layout_marginBottom="10dp"
android:layout_marginTop="10dp"
android:textColor="#000000" android:gravity="center" />
```

<TextView

```
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="Bangalore"
android:textSize="18dp"
android:layout_marginBottom="10dp"
android:layout_marginTop="10dp"
android:textColor="#000000"
android:gravity="center" />
```

<TextView

```
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="Email:geetapatil@bmsit.in"
android:textSize="18dp"
android:layout_marginBottom="10dp"
android:layout_marginTop="10dp"
    android:textColor="#000000"
    android:gravity="center" />
```

```
<TextView
    android:id="@+id/textView2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dp"
    android:layout_marginBottom="10dp"
    android:gravity="center"
    android:text="Website:https://bmsit.ac.in"
    android:textColor="#000000"
    android:textSize="18dp" />
</LinearLayout>
```

## MainActivity.java

```
package com.example.ex1_visitingcard;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

## Sample Output

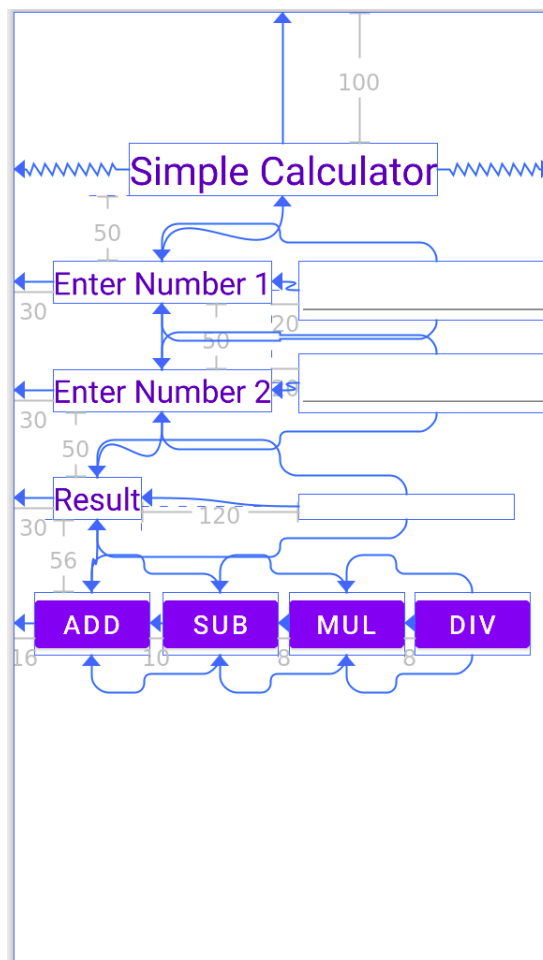


## Program 2

Develop an Android application using controls like Button, TextView, EditText for designing a Calculator having basic functionality like Addition, Subtraction, Multiplication, and Division.

1. Create a New Android Project with Empty Activity.
2. Open activity\_main.xml file from res→ layout folder, check/add Constraint Layout as the root view.
3. Create Layout using Drag and Drop framework.
4. Open MainActivity.java file, Override onCreate() method and bring activity\_main.xml file on screen using setContentView() and bring the view references using findViewById() method.
5. Add Listeners to Button Click Event:
6. Create a class which implements OnClickListener interface.
7. Override onClick() method of OnClickListener Interface.
8. Register the button for click event by calling setOnClickListener() method of View class and pass the object of the class that implemented OnClickListener Interface.
9. Create a logic to Add/Subtract/Multiply/Divide to perform arithmetic operation.

### Design



## activity\_main.xml

```
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/txtVTitle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Simple Calculator"
        android:textSize="30sp"
        android:textColor="@color/purple_700"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        android:layout_marginTop="100dp"/>

    <TextView
        android:id="@+id/txtVNum1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="30dp"
        android:layout_marginTop="50dp"
        android:text="Enter Number 1"
        android:textColor="@color/purple_700"
        android:textSize="24sp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/txtVTitle"/>

    <TextView
        android:id="@+id/txtVNum2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="30dp"
        android:layout_marginTop="50dp"
        android:text="Enter Number 2"
        android:textColor="@color/purple_700"
        android:textSize="24sp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/txtVNum1"/>

    <TextView
        android:id="@+id/txtVResult"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="30dp"
```

```
android:layout_marginTop="50dp"
android:text="Result"
android:textColor="@color/purple_700"
android:textSize="24sp"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/txtVNum2"/>
```

<EditText

```
android:id="@+id/txtNum1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginStart="20dp"
android:ems="10"
android:inputType="textPersonName"
android:text=" "
app:layout_constraintBottom_toBottomOf="@+id/txtVNum1"
app:layout_constraintStart_toEndOf="@+id/txtVNum1"
app:layout_constraintTop_toTopOf="@+id/txtVNum1"
app:layout_constraintVertical_bias="0.0" />
```

<EditText

```
android:id="@+id/txtNum2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginStart="20dp"
android:ems="10"
android:inputType="textPersonName"
android:text=" "
app:layout_constraintBottom_toBottomOf="@+id/txtVNum2"
app:layout_constraintStart_toEndOf="@+id/txtVNum2"
app:layout_constraintTop_toTopOf="@+id/txtVNum2"
app:layout_constraintVertical_bias="1.0" />
```

<TextView

```
android:id="@+id/txtVResult1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginStart="120dp"
android:ems="10"
android:inputType="textPersonName"
android:text=" "
app:layout_constraintBottom_toBottomOf="@+id/txtVResult"
app:layout_constraintStart_toEndOf="@+id/txtVResult"
app:layout_constraintTop_toTopOf="@+id/txtVResult"
app:layout_constraintVertical_bias="1.0" />
```

<Button

```
android:id="@+id/butAdd"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginStart="16dp"
android:layout_marginTop="56dp"
```



```
android:text="ADD"
android:textSize="20sp"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/txtVResult"/>
```

```
<Button
    android:id="@+id/butSub"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="10dp"
    android:text="SUB"
    android:textSize="20sp"
    app:layout_constraintBottom_toBottomOf="@+id/butAdd"
    app:layout_constraintStart_toEndOf="@+id/butAdd"
    app:layout_constraintTop_toTopOf="@+id/butAdd"
    app:layout_constraintVertical_bias="0.0"/>
```

```
<Button
    android:id="@+id/butMul"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:text="MUL"
    android:textSize="20sp"
    app:layout_constraintBottom_toBottomOf="@+id/butSub"
    app:layout_constraintStart_toEndOf="@+id/butSub"
    app:layout_constraintTop_toTopOf="@+id/butSub"
    app:layout_constraintVertical_bias="0.0"/>
```

```
<Button
    android:id="@+id/butDiv"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:text="DIV"
    android:textSize="20sp"
    app:layout_constraintBottom_toBottomOf="@+id/butMul"
    app:layout_constraintStart_toEndOf="@+id/butMul"
    app:layout_constraintTop_toTopOf="@+id/butMul"
    app:layout_constraintVertical_bias="0.0"/>
```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

## MainActivity.java

```
package com.example.tempcal;
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 implements View.OnClickListener {
    Button but_Add, but_Sub, but_Mul, but_Div;
    EditText txt_Num1, txt_Num2;
    TextView txt_Res;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

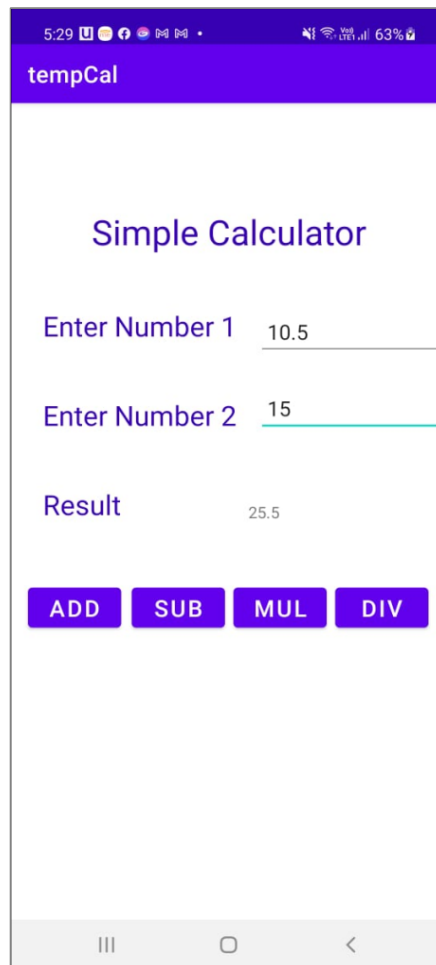
        but_Add = (Button) findViewById(R.id.butAdd);
        but_Sub = (Button) findViewById(R.id.butSub);
        but_Mul = (Button) findViewById(R.id.butMul);
        but_Div = (Button) findViewById(R.id.butDiv);

        txt_Num1 = (EditText) findViewById(R.id.txtNum1);
        txt_Num2 = (EditText) findViewById(R.id.txtNum2);
        txt_Res = (TextView) findViewById(R.id.txtVResult1);

        but_Add.setOnClickListener(this);
        but_Sub.setOnClickListener(this);
        but_Mul.setOnClickListener(this);
        but_Div.setOnClickListener(this);
    }

    @Override
    public void onClick(View v) {
        double Op1=Double.parseDouble(txt_Num1.getText().toString());
        double Op2=Double.parseDouble(txt_Num2.getText().toString());
        double Res;
        if (v.equals(but_Add))
            Res=Op1+Op2;
        else if(v.equals(but_Sub))
            Res=Op1-Op2;
        else if(v.equals(but_Mul))
            Res=Op1*Op2;
        else if(v.equals(but_Div))
            Res=Op1/Op2;
        else
            Res=0;
        txt_Res.setText(String.valueOf(Res));
    }
}
```

## Sample Output



### Program 3

Create a SIGN Up activity with Username and Password. Validation of password should happen based on the following rules:

- Password should contain uppercase and lowercase letters.
- Password should contain letters and numbers.
- Password should contain special characters.
- Minimum length of the password (the default value is 8).

On successful **SIGN UP** proceed to the next Login activity. Here the user should **SIGN IN** using the Username and Password created during signup activity. If the Username and Password are matched then navigate to the next activity which displays a message saying “Successful Login” or else display a toast message saying “Login Failed”. The user is given only two attempts and after that display a toast message saying “Failed Login Attempts” and disable the SIGN IN button. Use Bundle to transfer information from one activity to another.

The image shows two wireframe layouts for an Android application. The first wireframe, titled 'SIGNUP ACTIVITY', features a light gray grid background. It has two text labels, 'Username:' and 'Password:', each followed by a rectangular input field. Below these fields is a button labeled 'SIGN UP'. The second wireframe, titled 'LOGIN ACTIVITY', also has a light gray grid background. It features two text labels, 'Username:' and 'Password:', each followed by a rectangular input field. Below these fields is a button labeled 'SIGN IN'.

1. Create a New Android Project with Empty Activity.
2. Open activity\_main.xml file from res→ layout folder, check/add Constraint Layout as the root view.
3. Create Signup Layout using Drag and Drop framework design the layout.
4. Create One more Empty Activity LoginActivity using Android Studio Create Activity Flow (Refer Android Studio Tutorial)
5. Open activity\_login.xml file from res→layout folder, check/add Constraint Layout as the root view.
6. Create Login Layout using Drag and Drop framework.
7. Add Listeners to Button Click Event:
  - Create a class which implements OnClickListener interface.
  - Override onClick() method of OnClickListener Interface.
  - Register the button for click event by calling setOnClickListener() method of View class and pass the object of the class that implemented OnClickListener Interface.
8. Use Regular Expression `"^(?=.*[A-Z])(?=.*[a-z])(?=.*\\d)(?=.*[@$!]) [A-Za-z\\d@$!]{8,}$"` to validate the password.

## Design

## activity\_main.xml

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

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="100dp"
        android:text="SIGN UP"
        android:textColor="@color/purple_700"
        android:textSize="22dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        />
</androidx.constraintlayout.widget.ConstraintLayout>
```

```
app:layout_constraintTop_toTopOf="parent" />
```

```
<TextView
    android:id="@+id/textView3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="30dp"
    android:layout_marginTop="50dp"
    android:text="USERNAME"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView2" />
```

```
<TextView
    android:id="@+id/textView4"
    android:layout_width="82dp"
    android:layout_height="34dp"
    android:layout_marginTop="50dp"
    android:text="PASSWORD"
    app:layout_constraintStart_toStartOf="@+id/textView3"
    app:layout_constraintTop_toBottomOf="@+id/textView3" />
```

```
<EditText
    android:id="@+id/txt_username"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="40dp"
    android:layout_marginEnd="10dp"
    android:ems="10"
    android:inputType="textPersonName"
    app:layout_constraintBottom_toBottomOf="@+id/textView3"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toEndOf="@+id/textView3"
    app:layout_constraintTop_toTopOf="@+id/textView3" />
```

```
<EditText
    android:id="@+id/txt_password"
    android:layout_width="0dp"
    android:layout_height="40dp"
    android:layout_marginTop="26dp"
    android:ems="10"
    android:inputType="textPassword"
    app:layout_constraintEnd_toEndOf="@+id/txt_username"
    app:layout_constraintStart_toStartOf="@+id/txt_username"
    app:layout_constraintTop_toBottomOf="@+id/txt_username" />
```

```
<Button
    android:id="@+id/btn_signup"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="30dp"
    android:text="Sign Up"
    app:layout_constraintEnd_toEndOf="parent"
```

```
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/txt_password" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

### activity\_login.xml file

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

    <TextView
        android:id="@+id/textView7"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="50dp"
        android:text="Login"
        android:textSize="22dp"
        android:textColor="@color/purple_700"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/textView9"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="30dp"
        android:layout_marginTop="50dp"
        android:text="USERNAME"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView7" />

    <EditText
        android:id="@+id/txt_login_username"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="20dp"
        android:layout_marginEnd="20dp"
        android:ems="10"
        android:inputType="textPersonName"
        app:layout_constraintBottom_toBottomOf="@+id/textView9"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toEndOf="@+id/textView9"
        app:layout_constraintTop_toTopOf="@+id/textView9" />

    <TextView
```

```

    android:id="@+id/textView10"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="30dp"
    android:layout_marginTop="50dp"
    android:text="PASSWORD"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView9" />

```

<EditText

```

    android:id="@+id/txt_login_password"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:ems="10"
    android:inputType="textPassword"
    app:layout_constraintEnd_toEndOf="@+id/txt_login_username"
    app:layout_constraintStart_toStartOf="@+id/txt_login_username"
    app:layout_constraintTop_toTopOf="@+id/textView10" />

```

<Button

```

    android:id="@+id/btn_login_signin"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="50dp"
    android:text="Login"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/txt_login_password" />

```

</androidx.constraintlayout.widget.ConstraintLayout>

## MainActivity.java

```

package com.allen.prog3;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

```

```

public class MainActivity extends AppCompatActivity implements View.OnClickListener {
    EditText txtUsername;
    EditText txtPassword;
    Button btnSignup;
    String regularExpression = "^(?=.*[A-Z])(?=.*[a-z])(?=.*\\d)(?=.*[@$!%*^&])[A-Za-z\\d@$!%*^&]{8,}$";

```

@Override



```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    txtUsername = (EditText) findViewById(R.id.txt_username);
    txtPassword = (EditText) findViewById(R.id.txt_password);
    btnSignup = (Button) findViewById(R.id.btn_signup);
    btnSignup.setOnClickListener(this);
}

public void onClick(View v) {
    String username = txtUsername.getText().toString();
    String password = txtPassword.getText().toString();
    if (validatePassword(password)) {
        Bundle bundle = new Bundle();
        bundle.putString("user", username);
        bundle.putString("Lab@2018", password);
        Intent it = new Intent(this, LoginActivity.class);
        it.putExtra("data", bundle);
        startActivity(it);
    } else {
        Toast.makeText(getApplicationContext(), "Invalid Password", Toast.LENGTH_LONG).show();
    }
}

public boolean validatePassword(String password) {
    Pattern pattern = Pattern.compile(regularExpression);
    Matcher matcher = pattern.matcher(password);
    return matcher.matches();
}
}
```

## LoginActivity.java file

```
package com.allen.prog3;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class LoginActivity extends AppCompatActivity implements View.OnClickListener {
    EditText txtLoginUsername;
    EditText txtLoginPassword;
    Button btnLogin;
    String user, pass;
    int count = 0;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
```

```
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_login);
txtLoginUsername = (EditText) findViewById(R.id.txt_login_username);
txtLoginPassword = (EditText) findViewById(R.id.txt_login_password);
btnLogin = (Button) findViewById(R.id.btn_login_signin);
btnLogin.setOnClickListener(this);
Bundle bundle = getIntent().getBundleExtra("data");
user = bundle.getString("user");
pass = bundle.getString("Lab@2018");
}

public void onClick(View v) {
    String user1 = txtLoginUsername.getText().toString();
    String pass1 = txtLoginPassword.getText().toString();
    if (user.equals(user1) && pass.equals(pass1)) {
        Toast.makeText(this, "Login Successful", Toast.LENGTH_LONG).show();
    } else {
        count++;
        if (count == 3) {
            btnLogin.setEnabled(false);
            Toast.makeText(this, "Failed Login Attempts", Toast.LENGTH_LONG).show();
        } else {
            Toast.makeText(this, "Login Failed " + count, Toast.LENGTH_LONG).show();
        }
    }
}
}
```

## Sample Output

6:25 66%

Prog3

SIGN UP

USERNAME geeta patil

PASSWORD ...

SIGN UP

Invalid Password

III O <

This screenshot shows a mobile application interface for a sign-up process. The status bar at the top displays the time as 6:25 and battery level at 66%. The app's title bar is purple and labeled 'Prog3'. Below the title bar, the text 'SIGN UP' is centered. There are two input fields: 'USERNAME' with the text 'geeta patil' and 'PASSWORD' with masked characters '...'. A purple 'SIGN UP' button is positioned below the password field. At the bottom of the screen, there is a grey button labeled 'Invalid Password'. The Android navigation bar at the very bottom shows the back, home, and recent apps icons.

6:23 66%

Prog3

SIGN UP

USERNAME geeta patil

PASSWORD .....

SIGN UP

III O <

This screenshot shows the same mobile application interface as the previous one, but with a slightly different time (6:23) and battery level (66%). The layout and elements are identical, including the 'SIGN UP' title, input fields for 'USERNAME' (geeta patil) and 'PASSWORD' (masked with dots), the purple 'SIGN UP' button, and the 'Invalid Password' button at the bottom. The Android navigation bar is also present at the bottom.

6:24 66%

Prog3

Login

Username geeta patil

PASSWORD ...

LOGIN

Login Failed 1

6:33 66%

Prog3

Login

Username geeta patil

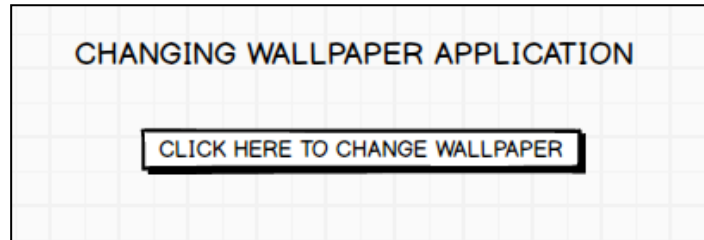
PASSWORD .....

LOGIN

Login Successful

### Program 4

Develop an application to set an image as wallpaper. On click of a button, the wallpaper image should start to change randomly every 30 seconds.



1. Create a New Android Project with Empty Activity.
2. Open activity\_main.xml file from res → layout folder, check/add LinearLayout as the root view.
3. Create the layout
4. Add 3 or More images to drawable folder (res → drawable)
5. Declare uses permission android.permission.SET\_WALLPAPER in the AndroidManifest.xml file
6. Schedule Timer task to change the wallpaper on every 30 seconds interval.
7. Initialize and use WallpaperManager.setBitmap() method to change the wallpaper.

#### activity\_main.xml

```
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="CHANGING WALLPAPER APPLICATION"
        android:textColor="@color/design_default_color_primary_dark"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.583"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.067" />
```

```
<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="72dp"
    android:layout_marginTop="53dp"
    android:layout_marginEnd="35dp"
    android:layout_marginBottom="590dp"
    android:text="CLICK HERE TO CHANGE WALLPAPER"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.714"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="1.0" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

## MainActivity.java

```
package com.example.changebackground;
import androidx.appcompat.app.AppCompatActivity;
import android.app.WallpaperManager;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.drawable.AnimationDrawable;
import android.graphics.drawable.BitmapDrawable;
import android.graphics.drawable.Drawable;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;
import java.io.IOException;
import java.util.Timer;
import java.util.TimerTask;

public class MainActivity extends AppCompatActivity {

    Button changewallpaper;
    Timer mytimer;
    Drawable drawable;
    WallpaperManager wpm;
    int prev=1;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

```
mytimer = new Timer();
wpm = WallpaperManager.getInstance(this);

changewallpaper = findViewById(R.id.button);

changewallpaper.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        setWallpaper();
    }
});

private void setWallpaper()
{

    mytimer.schedule(new TimerTask() {
        @Override
        public void run() {
            if(prev==1) {
                drawable = getResources().getDrawable(R.drawable.one);
                prev = 2;
            }
            else if(prev==2) {
                drawable = getResources().getDrawable(R.drawable.two);
                prev=3;
            }
            else if(prev==3) {
                drawable = getResources().getDrawable(R.drawable.three);
                prev=4;
            }
            else if(prev==4) {
                drawable = getResources().getDrawable(R.drawable.four);
                prev=5;
            }
            else if(prev==5) {
                drawable = getResources().getDrawable(R.drawable.five);
                prev=1;
            }
            }

        Bitmap wallpaper = ((BitmapDrawable)drawable).getBitmap();

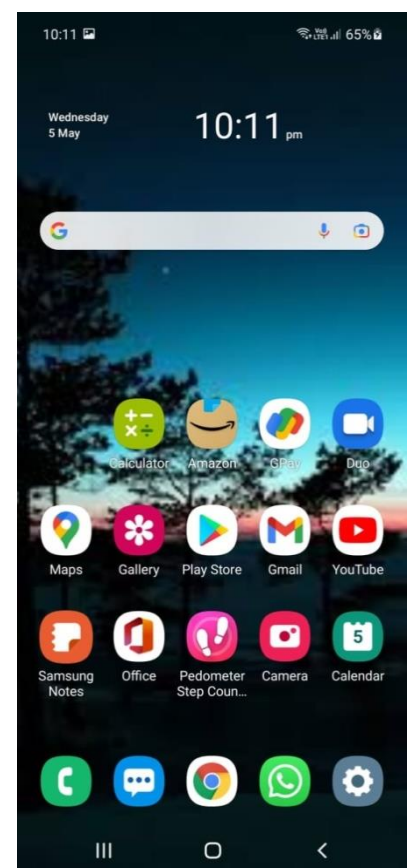
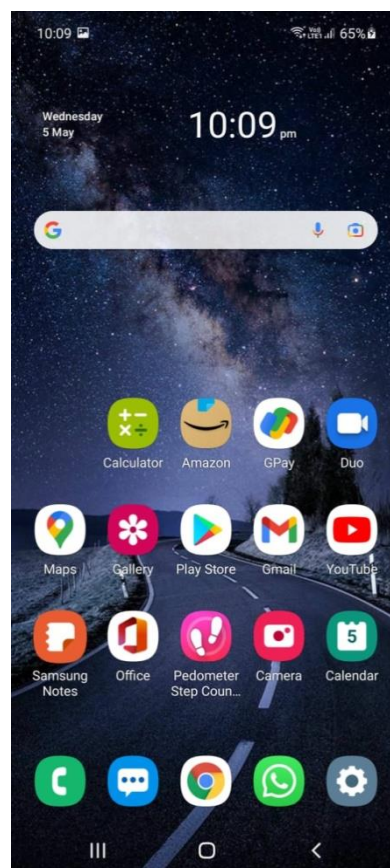
        try {
            wpm.setBitmap(wallpaper);
        }
        catch (IOException e) {
            e.printStackTrace();
        }
    },0,30000);
}
```

## AndriodManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.changebackground"
    <uses-permission android:name="android.permission.SET_WALLPAPER"/>
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/Theme.ChangeBackground">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

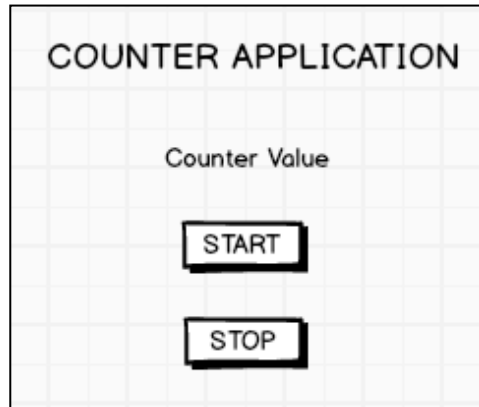
## Sample Output





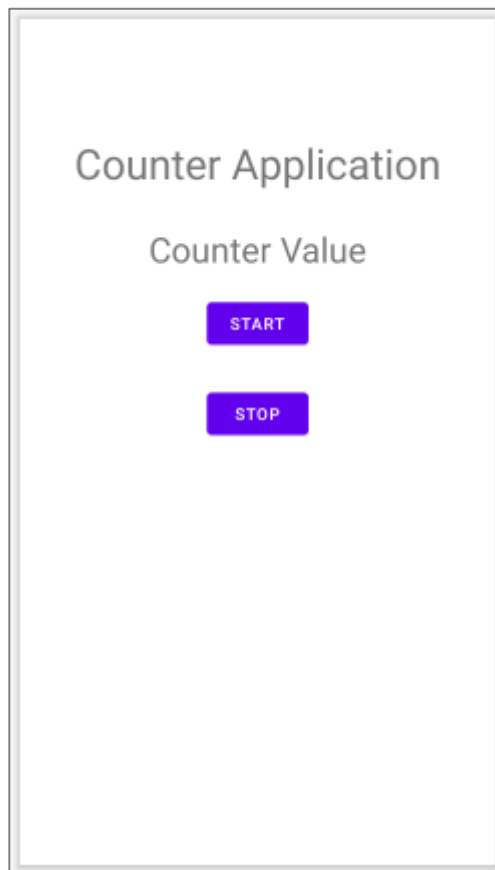
### Program 5

Write a program to create an activity with two buttons START and STOP. On Pressing of the START button, the activity must start the counter by displaying the numbers from One and the counter must keep on counting until the STOP button is pressed. Display the counter value in a TextViewcontrol.



1. Create a New Android Project with Empty Activity.
2. Open activity\_main.xml file from res→ layout folder, check/add ConstraintLayout as the root view.
3. Create the layout design using Drag and Drop framework.
4. Add Listeners to Button Click Event:
  - Create a class which implements OnClickListener interface.
  - Override onClick() method of OnClickListener Interface.
  - Register the button for click event by calling setOnClickListener() method of View class and pass the object of the class that implemented OnClickListener Interface.
5. Create a Thread to start the counter logic.
6. Steps to Create a Thread
  - Create a class that extends Thread Class.
  - Override run method of Thread Class.
  - Use start() method of thread class to start the thread.
7. Create Handler class to receive message from child thread, Handler executes in Main Thread.
8. Steps to Create Handler
  - Create Object of type Handler.
  - OverridhandleMessage() of handler class.
9. Pass the counter value to be displayed to the handler.
10. Update the UI to display the counter value received from thread.

## Design



## activity\_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/lbl_counter"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="100dp"
        android:text="Counter Application"
        android:textSize="36sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/lbl_text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="30dp"
        android:text="Counter Value"
        android:textSize="30sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView" />

    <Button
        android:id="@+id/btn_start"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:text="Start"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/lbl_text" />

    <Button
        android:id="@+id/btn_stop"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="30dp"
        android:text="Stop"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/btn_start" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

```
app:layout_constraintTop_toBottomOf="@+id/btn_start" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

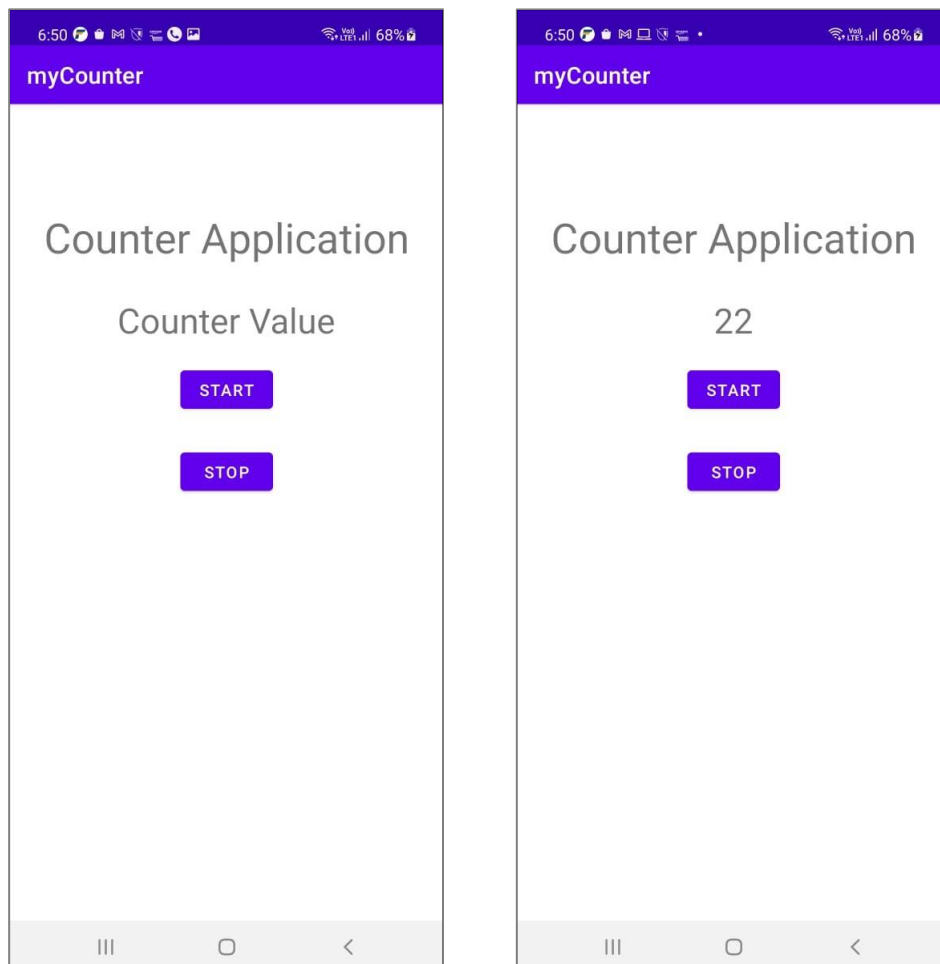
## MainActivity.java

```
package com.example.mycounter;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.os.Handler;
import android.os.Message;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import org.w3c.dom.Text;
public class MainActivity extends AppCompatActivity implements View.OnClickListener{
    TextView lblCounter;
    Button btnStart,btnStop;
    int counter=0;
    boolean running=false;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        lblCounter=(TextView)findViewById(R.id.lbl_text);
        btnStart=(Button)findViewById(R.id.btn_start);
        btnStop=(Button)findViewById(R.id.btn_stop);
        btnStop.setOnClickListener(this);
        btnStart.setOnClickListener(this);
    }
    public void onClick(View v)
    {
        if(v.equals(btnStart))
        {
            counter=0;
            running=true;
            new MyCounter().start();
        }
        else if(v.equals(btnStop))
        {
            running=false;
        }
    }
    Handler handler=new Handler()
    {
        public void handleMessage(Message m)
        {
            lblCounter.setText(String.valueOf(m.what));
        }
    };
    class MyCounter extends Thread
    {

```

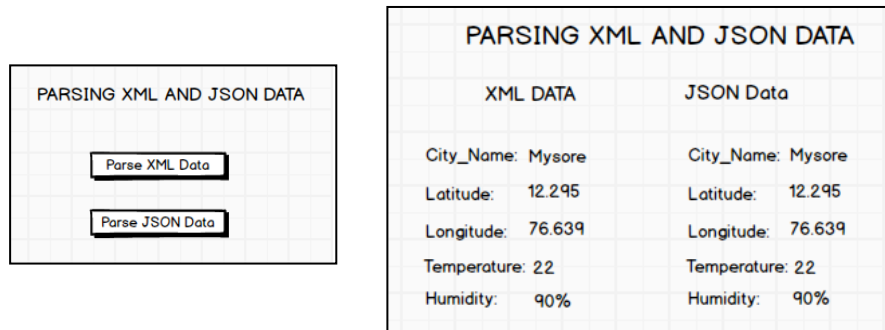
```
public void run()
{
    while(running)
    {
        counter++;
        handler.sendMessage(counter);
        try {
            Thread.sleep(1000);
        }
        catch(Exception e) { }
    }
}
```

### Sample Output



## Program 6

Create two files of XML and JSON type with values for City\_Name, Latitude, Longitude, Temperature, and Humidity. Develop an application to create an activity with two buttons to parse the XML and JSON files which when clicked should display the data in their respective layouts side by side.

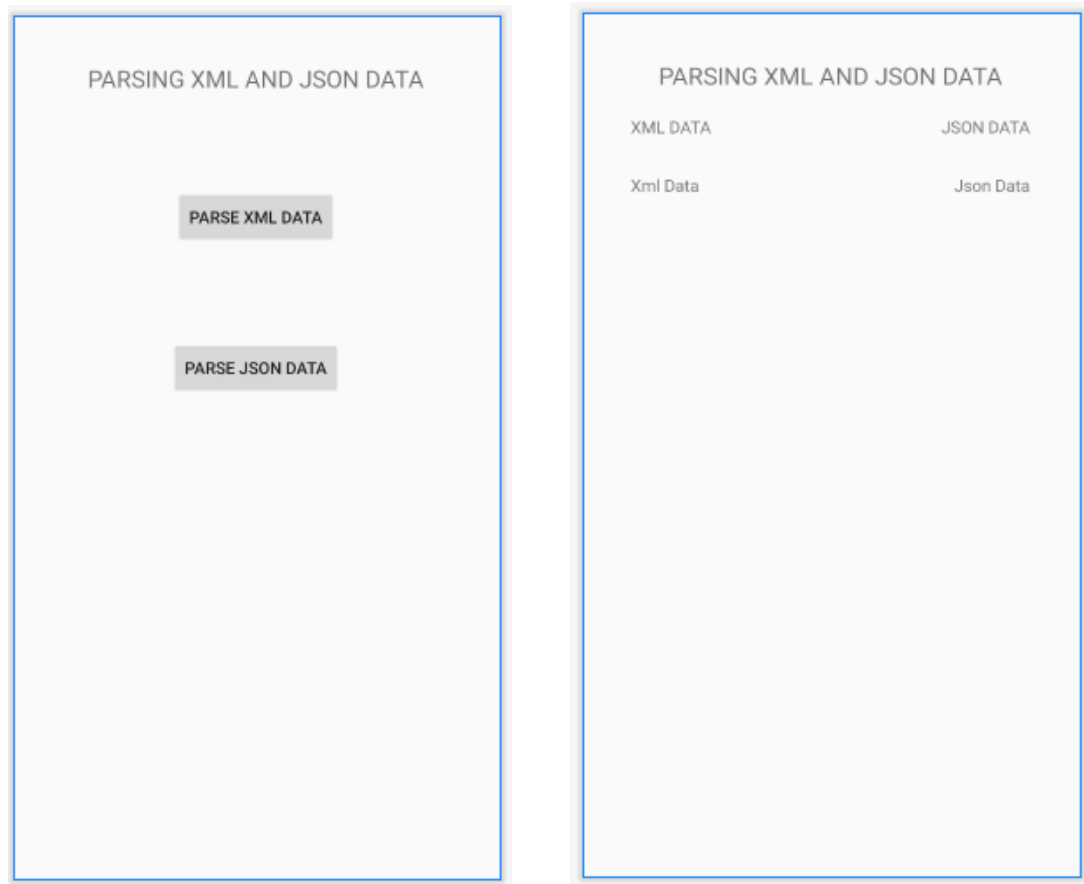


1. Create a New Android Project with Empty Activity.
2. Open activity\_main.xml file from res→ layout folder, check/add ConstraintLayout as the root view.
3. Create the layout design using Drag and Drop framework.
4. Add Listeners to Button Click Event:
  - Create a class which implements OnClickListener interface.
  - Override onClick() method of OnClickListener Interface.
  - Register the button for click event by calling setOnClickListener() method of View class and pass the object of the class that implemented OnClickListener Interface.
5. Create assets folder (Refer Section Android Studio Tutorial)
6. Create **input.xml** file inside assets folder and paste the below Xml Data
 

```
<?xml version="1.0"?>
<records>
<employee>
<city_name>Mysore</city_name>
<Latitude>12.295</Latitude>
<Longitude>76.639</Longitude>
<Temperature>22</Temperature>
<Humidity>90%</Humidity>
</employee>
</records>
```
7. Create **input.json** file inside assets folder and paste the below Json Data
 

```
{
  "employee": {
    "city_name": "Mysore",
    "Latitude": "12.295",
    "Longitude": "76.639",
    "Temperature": 22,
    "Humidity": "90%"
  }
}
```
8. Read the XML and Json Data in the files and display on screen

## Design



### activity\_main.xml

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

    <Button
        android:id="@+id/btn_parsexml"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="80dp"
        android:text="Parse XML Data"
        app:layout_constraintEnd_toEndOf="parent" app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView4" />
```

```
<Button
android:id="@+id/btn_parsejson"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="80dp"
android:text="Parse Json Data"
app:layout_constraintEnd_toEndOf="parent" app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/btn_parsexml" />

<TextView
android:id="@+id/textView4"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="40dp"
android:text="PARSING XML AND JSON DATA"
android:textSize="20dp"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

## activity\_view.xml

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

<TextView
android:id="@+id/lbl_xml_data"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="30dp"
android:text="Xml Data"
app:layout_constraintStart_toStartOf="@+id/textView2"
app:layout_constraintTop_toBottomOf="@+id/textView2" />

<TextView
android:id="@+id/textView"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="40dp"
android:text="PARSING XML AND JSON DATA"
android:textSize="20dp"
```



```
app:layout_constraintEnd_toEndOf="parent"
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="40dp"
    android:layout_marginTop="20dp"
    android:text="XML DATA"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView" />

<TextView
    android:id="@+id/textView3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:layout_marginEnd="40dp"
    android:text="JSON DATA"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView" />

<TextView
    android:id="@+id/lbl_json_data"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="30dp"
    android:text="Json Data"
    app:layout_constraintEnd_toEndOf="@+id/textView3"
    app:layout_constraintTop_toBottomOf="@+id/textView3" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

## MainActivity.java

```
package com.example.parta_program6;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class MainActivity extends AppCompatActivity implements View.OnClickListener {

    Button btnParseXml, btnParseJson;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState); setContentView(R.layout.activity_main);
        btnParseXml=(Button)findViewById(R.id.btn_parsexml);
        btnParseJson=(Button)findViewById(R.id.btn_parsejson);
```

```
btnParseJson.setOnClickListener(this); btnParseXml.setOnClickListener(this);
}

@Override
public void onClick(View v) {

    if(v.equals(btnParseJson))
    {
        Intent it=new Intent(this,ViewActivity.class);
        it.putExtra("mode",1);
        startActivity(it);

    }
    else if(v.equals(btnParseXml))
    {
        Intent it=new Intent(this,ViewActivity.class);
        it.putExtra("mode",2);
        startActivity(it);
    }
}
}
```

## ViewActivity.java

```
package com.example.parta_program6;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.widget.TextView;

import org.json.JSONObject;
import
org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.Node;
import
org.w3c.dom.NodeList;

import java.io.InputStream;

import javax.xml.parsers.DocumentBuilder; import
javax.xml.parsers.DocumentBuilderFactory;

public class ViewActivity extends AppCompatActivity {

    TextView lblXmlData, lblJsonData;

    int
    mode=0;
    @Override
    e
    protected void onCreate(Bundle savedInstanceState) {
```

```
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_view);
lblXmlData=(TextView)findViewById(R.id.lbl_xml_data);
lblJsonData=(TextView)findViewById(R.id.lbl_json_data);
mode=getIntent().getIntExtra("mode",0);

if(mode==
1)
parseJson()
;

else
parseXmlDocument();

}

public String parseXmlDocument()
{
try {

InputStream is = getAssets().open("input.xml");

DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();
Document doc = dBuilder.parse(is);

Element
element=doc.getDocumentElement();
element.normalize();

NodeList nList = doc.getElementsByTagName("employee");for (int i=0;

i<nList.getLength(); i++) {

Node node = nList.item(i);
if (node.getNodeType() ==
Node.ELEMENT_NODE) {Element element2 =
(Element) node;
lblXmlData.setText("City Name : " + getValue("city_name", element2)+"\n");
lblXmlData.append("Latitude : " + getValue("Latitude", element2)+"\n");
lblXmlData.append("Longitude : " + getValue("Longitude", element2)+"\n");
lblXmlData.append("Temperature : " + getValue("Temperature", element2)+"\n");
lblXmlData.append("Humidity : " + getValue("Humidity", element2)+"\n");

}
}
}
catch (Exception e) {e.printStackTrace();}return
null;
}

private static String getValue(String tag, Element element) { NodeList nList =
element.getElementsByTagName(tag).item(0).getChildNodes();Node node =
```

```

nodeList.item(0);
return node.getNodeValue();
}

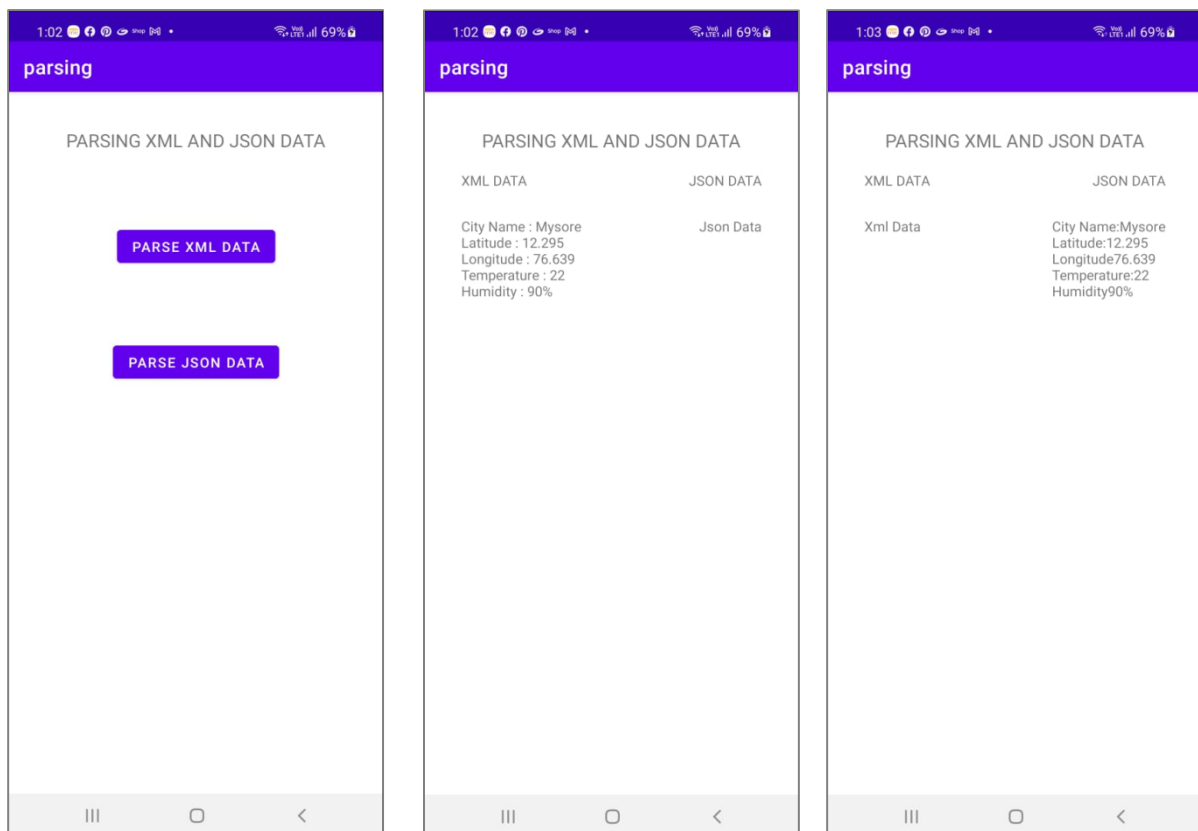
public void parseJson()
{
try { InputStream inputStream=getAssets().open("input.json");
byte[] data=new byte[inputStream.available()];
inputStream.read(data);

String readData=new String(data);
JSONObject jsonObject=new JSONObject(readData);
JSONObject jsonObject1=jsonObject.getJSONObject("employee");
lblJsonData.setText("City Name:"+jsonObject1.getString("city_name")+"\n");
lblJsonData.append("Latitude:"+jsonObject1.getString("Latitude")+"\n");
lblJsonData.append("Longitude:"+jsonObject1.getString("Longitude")+"\n");
lblJsonData.append("Temperature:"+jsonObject1.getInt("Temperature")+"\n");
lblJsonData.append("Humidity"+jsonObject1.getString("Humidity")+"\n");

}
catch (Exception e) {e.printStackTrace();}
}
}

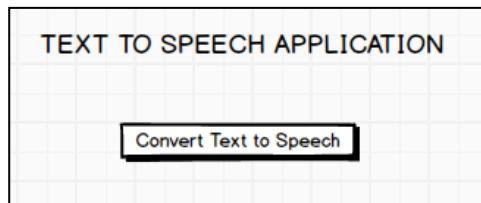
```

## Sample Output



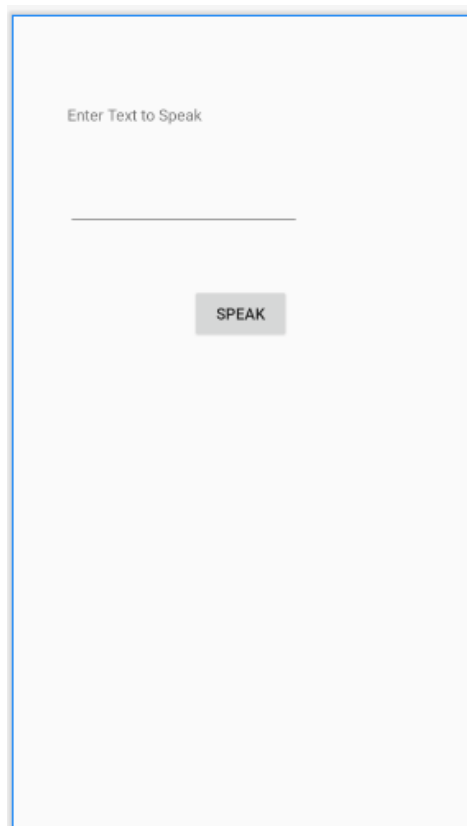
## Program 7

Develop a simple application with one EditText so that the user can write some text in it. Create a button called “Convert Text to Speech” that converts the user input text into voice.



1. Create a New Android Project with Empty Activity.
2. Open activity\_main.xml file from res→ layout folder, check/add ConstraintLayout as the root view.
3. Create the layout design using Drag and Drop framework.
4. Add Listeners to Button Click Event:
  - Create a class which implements OnClickListener interface.
  - Override onClick() method of OnClickListener Interface.
  - Register the button for click event by calling setOnClickListener() method of View class and pass the object of the class that implemented OnClickListener Interface.
5. Initialize TextToSpeech Engine and the Language to Speak using setLanguage() method
6. Use Speak() method to speak the text passed to it.

## Design



**activity\_main.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/txt_texttospeak"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="50dp"
        android:layout_marginTop="80dp"
        android:text="Enter Text to Speak"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <EditText android:id="@+id/editText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="48dp"
        android:ems="10"
        android:inputType="textPersonName"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="@+id/textView"
        app:layout_constraintTop_toBottomOf="@+id/textView" />

    <Button
        android:id="@+id/btn_speak"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="52dp" android:text="Speak"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/editText" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

## MainActivity.java

```
package com.example.parta.parta_program7;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.speech.tts.TextToSpeech;
import android.view.View;
import android.widget.Button;
```

```
import android.widget.EditText;
import android.widget.Toast;
import java.util.Locale;

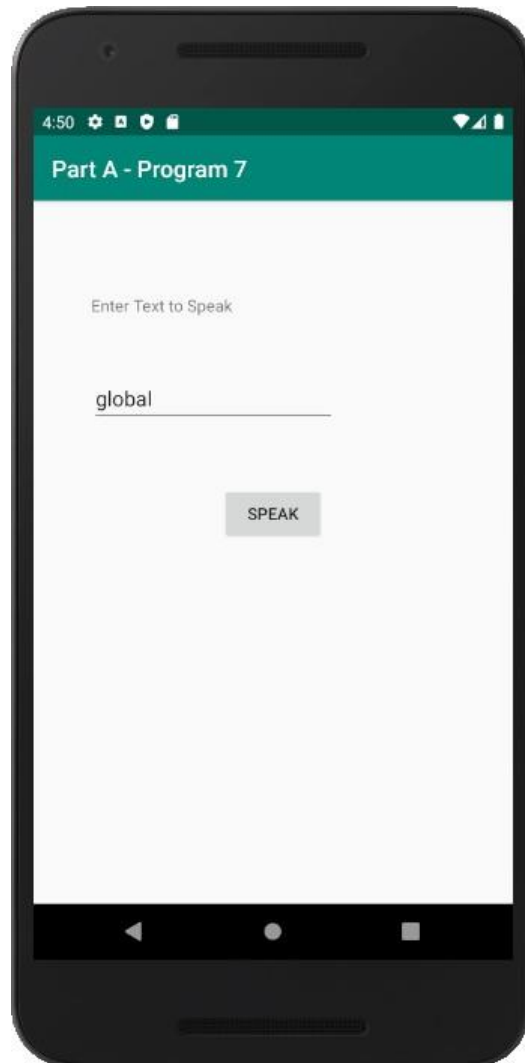
public class MainActivity extends AppCompatActivity implements View.OnClickListener {

    EditText txtSpeak;
    Button btnSpeak;
    TextToSpeech textToSpeech;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        txtSpeak=(EditText)findViewById(R.id.editText);
        btnSpeak=(Button)findViewById(R.id.btn_speak);
        btnSpeak.setOnClickListener(this);
        textToSpeech=new TextToSpeech(getApplicationContext(),
        new TextToSpeech.OnInitListener() {
            @Override
            public void onInit(int status) {
                if(status!=TextToSpeech.ERROR)
                {
                    Toast.makeText(getApplicationContext(),"Success", Toast.LENGTH_LONG).show();
                }
            }
        });
        textToSpeech.setLanguage(Locale.UK);
    }

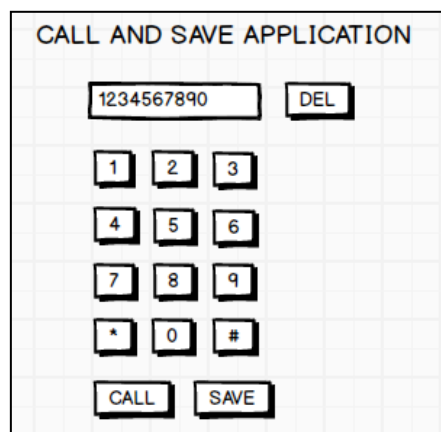
    public void onClick(View v)
    {
        String text=txtSpeak.getText().toString();
        textToSpeech.speak(text,TextToSpeech.QUEUE_FLUSH,null
        );
    }
}
```

## Sample Output



## Program 8

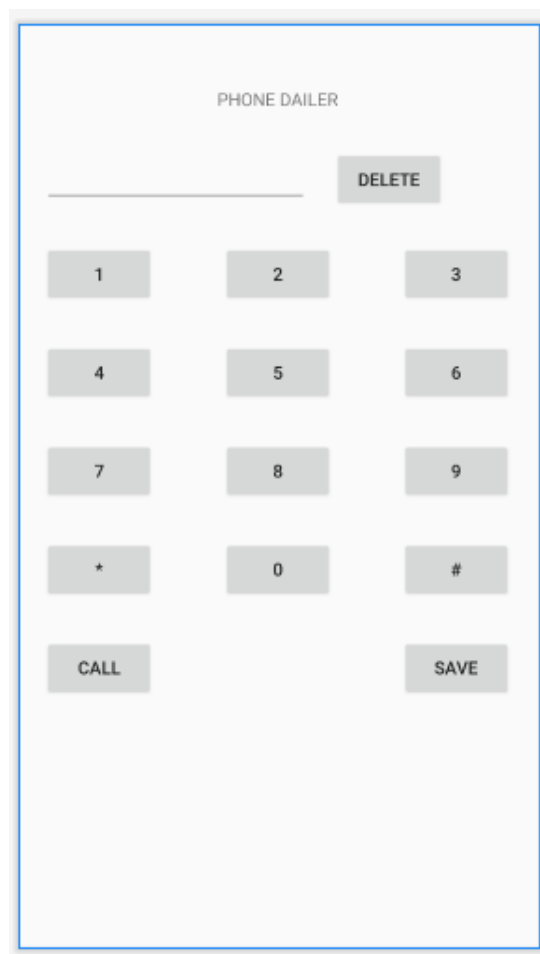
Create an activity like a phone dialer with CALL and SAVE buttons. On pressing the CALL button, it must call the phone number and on pressing the SAVE button it must save the number to the phone contacts.





1. Create a New Android Project with Empty Activity.
2. Open activity\_main.xml file from res → layout folder, check/add ConstraintLayout as the root view.
3. Create the layout design using Drag and Drop framework.
4. Add Listeners to Button Click Event:
  - Create a class which implements OnClickListener interface.
  - Override onClick() method of OnClickListener Interface.
  - Register the button for click event by calling setOnClickListener() method of View class and pass the object of the class that implemented OnClickListener Interface.
5. Declare uses permission android.permission.CALL\_PHONE in the manifest file.
6. Use ACTION\_CALL intent name and pass the “tel:<phone-number>” as URI in intent data and start the call activity.
7. Use intent name and pass the “Telephone Number” and “unknown” as name as intent data call Contacts Save Activity.

## Design



## activity\_main.xml

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

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="50dp"
        android:text="PHONE DAILER"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <EditText
        android:id="@+id/txt_phonenumber"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="20dp"
        android:layout_marginTop="30dp"
        android:ems="10"
        android:inputType="textPersonName"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView" />

    <Button android:id="@+id/btn_delete"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="20dp"
        android:layout_marginTop="30dp"
        android:text="Delete"
        app:layout_constraintStart_toEndOf="@+id/txt_phonenumber"
        app:layout_constraintTop_toBottomOf="@+id/textView" />

    <Button
        android:id="@+id/btn_one"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="20dp"
        android:layout_marginTop="30dp"
        android:text="1"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/txt_phonenumber" />
```

```
<Button
android:id="@+id/btn_two"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="30dp"
android:text="2"
app:layout_constraintEnd_toStartOf="@+id/btn_three"
app:layout_constraintStart_toEndOf="@+id/btn_one"
app:layout_constraintTop_toBottomOf="@+id/txt_phonenumber" />
```

```
<Button
android:id="@+id/btn_three"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="30dp"
android:layout_marginEnd="20dp"
android:text="3"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintTop_toBottomOf="@+id/txt_phonenumber" />
```

```
<Button
android:id="@+id/btn_four"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginStart="20dp"
android:layout_marginTop="30dp"
android:text="4"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/btn_one" />
```

```
<Button
android:id="@+id/btn_five"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="30dp"
android:text="5"
app:layout_constraintEnd_toStartOf="@+id/btn_six"
app:layout_constraintStart_toEndOf="@+id/btn_four"
app:layout_constraintTop_toBottomOf="@+id/btn_two" />
```

```
<Button
android:id="@+id/btn_six"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="30dp"
android:layout_marginEnd="20dp"
android:text="6"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintTop_toBottomOf="@+id/btn_three" />
```

```
<Button
```

```
android:id="@+id/btn_seven"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginStart="20dp"
android:layout_marginTop="30dp"
android:text="7"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/btn_four" />
```

```
<Button
android:id="@+id/btn_eight"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="30dp"
android:text="8"
app:layout_constraintEnd_toStartOf="@+id/btn_nine"
app:layout_constraintStart_toEndOf="@+id/btn_seven"
app:layout_constraintTop_toBottomOf="@+id/btn_five" />
```

```
<Button
android:id="@+id/btn_nine"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="30dp"
android:layout_marginEnd="20dp"
android:text="9"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintTop_toBottomOf="@+id/btn_six" />
```

```
<Button
android:id="@+id/btn_zero"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="30dp"
android:text="0"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/btn_eight" />
```

```
<Button
android:id="@+id/btn_call"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginStart="20dp"
android:layout_marginTop="30dp" android:text="Call"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/btn_zero" />
```

```
<Button
android:id="@+id/btn_save"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
```

```

android:layout_marginTop="30dp"
android:layout_marginEnd="20dp"
android:text="Save"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintTop_toBottomOf="@+id/btn_zero" />

<Button
    android:id="@+id/btn_start"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="20dp"
    android:layout_marginTop="30dp"
    android:text="*"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/btn_seven" />

<Button
    android:id="@+id/btn_hash"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="30dp"
    android:layout_marginEnd="20dp"
    android:text="#"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/btn_nine" />
</androidx.constraintlayout.widget.ConstraintLayout>

```

## MainActivity.java

```

package com.example.part_a_program_8;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.provider.ContactsContract;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

public class MainActivity extends AppCompatActivity implements View.OnClickListener{

    Button btnOne,btnTwo,btnThree,btnFour,btnFive;
    Button btnSix,btnSeven,btnEight,btnNine,btnZero;
    Button btnDel,btnStar,btnHash,btnCall,btnSave;

    EditTexttxtPhonenumber;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}

```

```
btnOne=(Button)findViewById(R.id.btn_one);
btnOne.setOnClickListener(this);

btnTwo=(Button)findViewById(R.id.btn_two);
btnTwo.setOnClickListener(this);

btnThree=(Button)findViewById(R.id.btn_three);
btnThree.setOnClickListener(this);

btnFour=(Button)findViewById(R.id.btn_four);
btnFour.setOnClickListener(this);

btnFive=(Button)findViewById(R.id.btn_five);
btnFive.setOnClickListener(this);

btnSix=(Button)findViewById(R.id.btn_six);
btnSix.setOnClickListener(this);

btnSeven=(Button)findViewById(R.id.btn_seven);
btnSeven.setOnClickListener(this);

btnEight=(Button)findViewById(R.id.btn_eight);
btnEight.setOnClickListener(this);

btnNine=(Button)findViewById(R.id.btn_nine);
btnNine.setOnClickListener(this);

btnZero=(Button)findViewById(R.id.btn_zero);
btnZero.setOnClickListener(this);

btnStar=(Button)findViewById(R.id.btn_start);
btnStar.setOnClickListener(this);

btnHash=(Button)findViewById(R.id.btn_hash);
btnHash.setOnClickListener(this);

btnCall=(Button)findViewById(R.id.btn_call);
btnCall.setOnClickListener(this);

btnSave=(Button)findViewById(R.id.btn_save);
btnSave.setOnClickListener(this);

btnDel=(Button)findViewById(R.id.btn_delete);
btnDel.setOnClickListener(this);

txtPhonenumber=(EditText)findViewById(R.id.txt_phonenumber);
txtPhonenumber.setText("");
}

public void onClick(View v)
{
    if(v.equals(btnOne))
```

```
txtPhonenumber.append("1");

else if(v.equals(btnTwo)) txtPhonenumber.append("2");

else if(v.equals(btnThree))txtPhonenumber.append("3");

else if(v.equals(btnFour))txtPhonenumber.append("4");

else if(v.equals(btnFive))txtPhonenumber.append("5");

else if(v.equals(btnSix)) txtPhonenumber.append("6");

else if(v.equals(btnSeven))txtPhonenumber.append("7");

else if(v.equals(btnEight))txtPhonenumber.append("8");

else if(v.equals(btnNine))txtPhonenumber.append("9");

else if(v.equals(btnZero))txtPhonenumber.append("0");

else if(v.equals(btnStar)) txtPhonenumber.append("*");

else if(v.equals(btnHash))txtPhonenumber.append("#");

else if(v.equals(btnSave))
{
Intent contactIntent= new Intent
                    (ContactsContract.Intents.Insert.ACTION);
contactIntent.setType
                    (ContactsContract.RawContacts.CONTENT_TYPE);

contactIntent
.putExtra(ContactsContract.Intents.Insert.NAME,"Unknown");
contactIntent.putExtra(ContactsContract.Intents.Insert.PHONE,
txtPhonenumber.getText().toString());

startActivity(contactIntent);

}

else if(v.equals(btnDel))
{
String data=txtPhonenumber.getText().toString();
if(data.length(>0)
    txtPhonenumber.setText(data.substring(0,data.length()-1));
else
    txtPhonenumber.setText("");
}

btnCall.setOnClickListener(new View.OnClickListener()
{
@Override
public void onClick(View v) {
```

```
String data = txtPhonenumber.getText().toString();Intent
intent=new Intent(Intent.ACTION_DIAL);
intent.setData(Uri.parse("tel:"+ data));
startActivity(intent);
    }
    }
);
}
}
```

## AndriodManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="com.example.part_a_program_8">

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

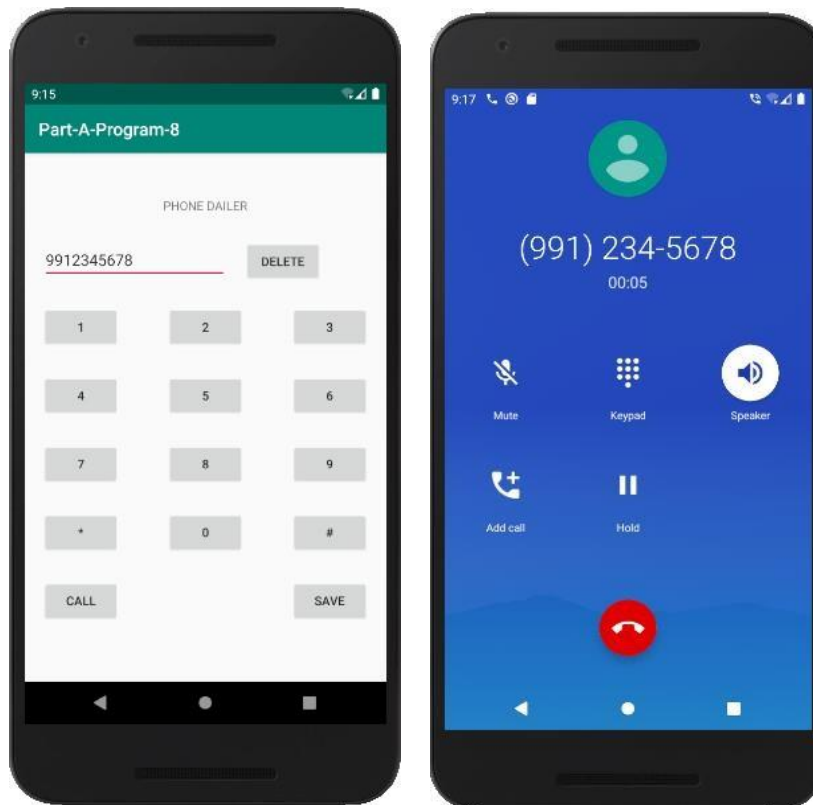
<application
android:allowBackup="true"
android:icon="@mipmap/ic_launcher"
android:label="@string/app_name"
android:roundIcon="@mipmap/ic_launcher_round"
android:supportRtl="true"
android:theme="@style/AppTheme">
<activity android:name=".MainActivity">
<intent-filter>
<action android:name="android.intent.action.MAIN" />

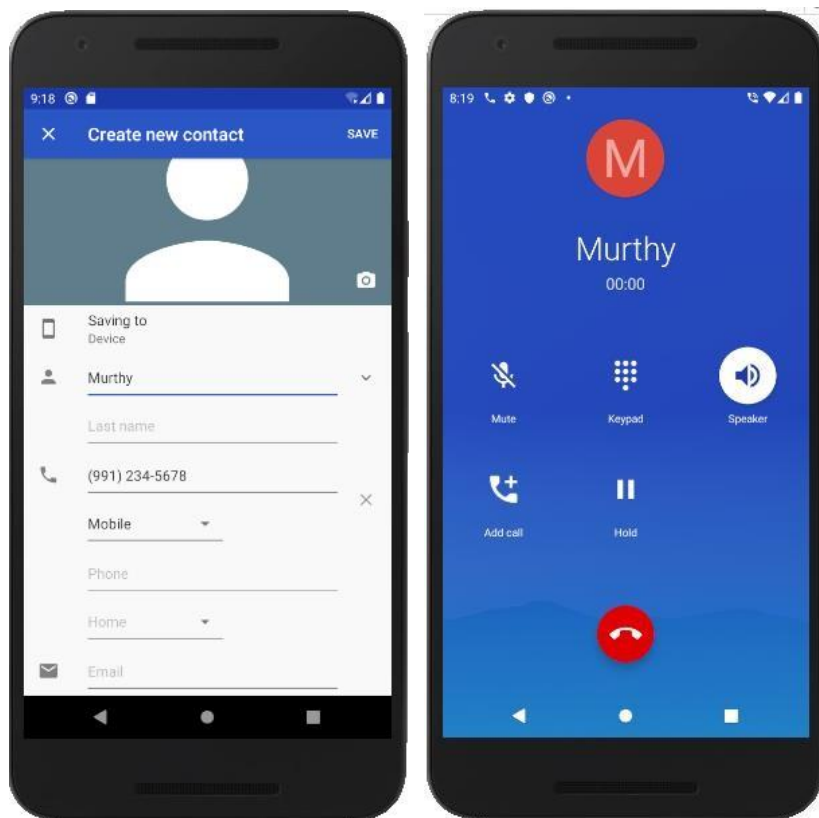
<category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
</activity>
</application>

</manifest>
```



## Sample Output





## Reference Books

1. Google Developer Training, "Android Developer Fundamentals Course – Concept Reference", Google Developer Training Team, 2017.  
<https://www.gitbook.com/book/google-developer-training/android-developer-fundamentals-course-concepts/details>  
(Download pdf file from the above link)
2. Erik Hellman, "Android Programming – Pushing the Limits", 1<sup>st</sup> Edition, Wiley India Pvt Ltd, 2014. ISBN-13: 978-8126547197
3. Dawn Griffiths and David Griffiths, "Head First Android Development", 1<sup>st</sup> Edition, O'Reilly SPD Publishers, 2015. ISBN-13: 978-9352131341
4. Bill Phillips, Chris Stewart and Kristin Marsicano, "Android Programming: The Big Nerd Ranch Guide", 3<sup>rd</sup> Edition, Big Nerd Ranch Guides, 2017. ISBN-13: 978-0134706054
5. Data from Global Academy of Technology