



**JSS MAHAVIDYAPEETHA**  
**JSS ACADEMY OF TECHNICAL EDUCATION**  
JSSATE Campus, Uttarahalli – Kengeri Main Road, Bengaluru – 560060  
**Department of Information Science & Engineering**



# **LABORATORY MANUAL**

**Mobile Application Development 18CSMP68**

## **Editorial Committee**

**Dr. Sowmya K.N, Department of ISE**

**Mrs. Punitha M, Department of ISE**

## **Approved by**

**H.O.D, Department of ISE**

## **Document Log**

Name of the document	Mobile Application Development Laboratory
Syllabus Scheme	2020-2021
Current version number and date	V1 / 19.04.2021
Subject code	18CSMP68
Editorial Committee	Dr. Sowmya K.N, Department of ISE Mrs. Punitha M, Department of ISE
Approved by	HOD, Dept. of ISE
Lab faculty	Dr. Sowmya K.N, Department of ISE Mrs. Punitha M, Department of ISE Mrs. Nagashree S, Department of ISE Mr. Anil B.C , Department of ISE

## **ABOUT JSSATEB**

JSS Academy of Technical Education (JSSATE) was established in the year 1997 at Bengaluru and is under the umbrella of JSS Mahavidyapeetha, Mysuru. JSSATE is the result of the vision of our President, His Holiness Jagadguru Sri Shivarathri Deshikendra Mahaswamiji to proactively participate in establishing a world class Institution for Technical Education. The Campus is located on a sprawling 21.17 acres land surrounded by lush green plantation on the South-Western edge of Bengaluru City. The institution is affiliated to Visvesvaraya Technological University (VTU), Belagavi, India

## **About ISE Department**

The Department of Information Science and Engineering is established in the year 1999. The current intake is 120. Department is strengthened by well-qualified faculty members main aim is to emphasize the overall growth and development of students with cutting-edge technology by inculcating competence, commitment & teamwork. The Department comprises excellent infrastructure with well-equipped Laboratories, Classrooms, and Departmental Library.

The Student forum SAMYOG is solely for the benefit of the student community established with the aim of by the students, of the students and from the students. The students actively take part in co-curricular activities like projects, paper & poster presentations, coding, debugging, etc. In the extra-curricular events like NSS, sports, college fest, and the preparation of technical magazines & newsletters. They are also participating in self-initiated outreach activities like YODHA , Book borrow.com, bookbeka.com, medical camps like blood donation camps, eye testing camps, dental camps, and Ayurvedic camps, etc. Industry interaction is established through MOUs. A good number of industrial visits, Invited lecturers, workshops, seminars, and webinars are also organized regularly in emerging technologies from experts of industries and institutions.

Most of our students are well placed in fortune 500 companies. Our students have recorded very good performance in competitive examinations such as GATE, GRE, GMAT, TOEFL, CAT, PGCET, etc. Many of our Alumni joined MS programs in the leading and reputed Universities in the world.

## **Vision of the Institute**

- To be among the finest Institutions providing Engineering and Management Education empowered with research, innovation and entrepreneurship

## **Mission of the Institute**

- Strive towards Excellence in teaching–learning process and nurture personality development.
- Encourage Research, Innovation & Entrepreneurship.
- Train to uphold highest ethical standards in all activities.

## **Vision of the Department**

- To emerge as a centre of academic excellence, by producing competent professionals to meet the global challenges in the field of Information Science and Technology

## **Mission of the Department**

- Prepare the competent professionals to meet the advancements in the industry and academia by imparting quality technical education.
- Enrich the technical ability of students to face the world with confidence, commitment and team-work.
- Inculcate and practice strong techno-ethical values

## **Program Outcomes (POs)**

### **Engineering Graduates will be able to:**

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

## Course Details

**Course Name:** **Mobile Application Development**

**Course Code:** **18CSMP68**

**Course prerequisite:** Core Java

## Course Objectives

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

1. Learn and acquire the art of Android Programming.
2. Configure Android studio to run the applications.
3. Understand and implement Android's User interface functions.
4. Create, modify and query on SQLite database.
5. Inspect different methods of sharing data using services.

## Course Outcomes

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

1. Build, test and debug Android application by setting up Android development environment.
2. Construct adaptive, responsive user interfaces that work across a wide range of devices.
3. Analyze long running tasks and background work in Android applications.
4. Experiment with various methods in storing, sharing and retrieving data in Android applications
5. Inspect the role of permissions and security for Android applications.

## **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: 3 Hours/Week**

**Exam Hours: 03**

**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**

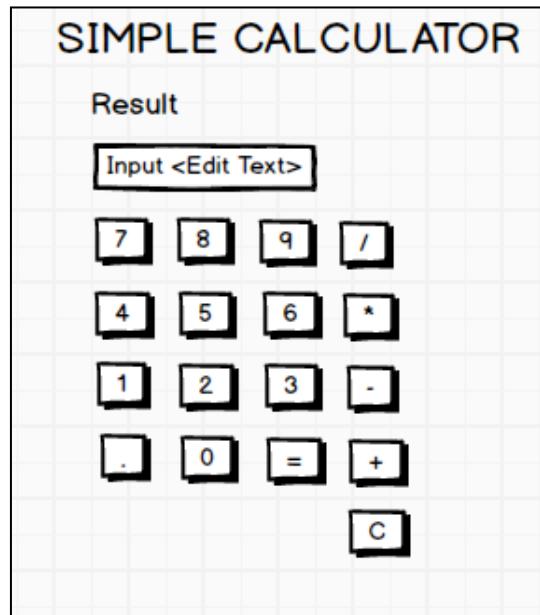
### **Program 1**

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.



### 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.

**SIGNUP ACTIVITY**

Username:

Password:

**SIGN UP**

**LOGIN ACTIVITY**

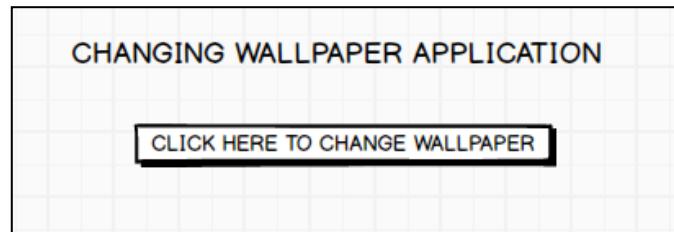
Username:

Password:

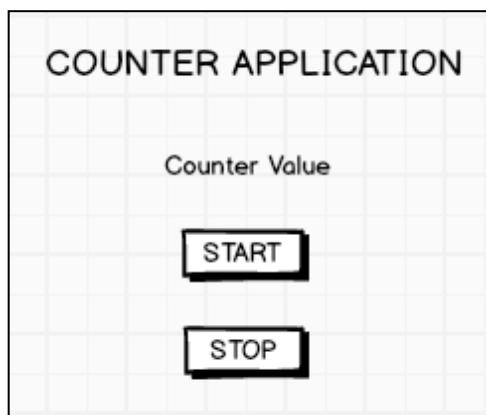
**SIGN IN**

**Program 4**

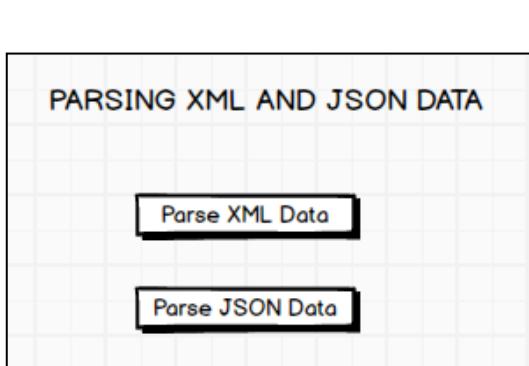
Develop an application to set an image as wallpaper. On click of a button, the wallpaper image should start to change randomly every 30 seconds.

**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**

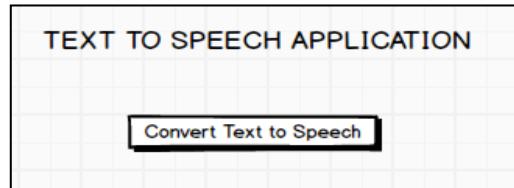
Create two files of XML and JSON type with values for City\_Name, Latitude, Longitude, Temperature, and Humidity. Develop an application to create an activity with two buttons to parse the XML and JSON files which when clicked should display the data in their respective layouts side by side.



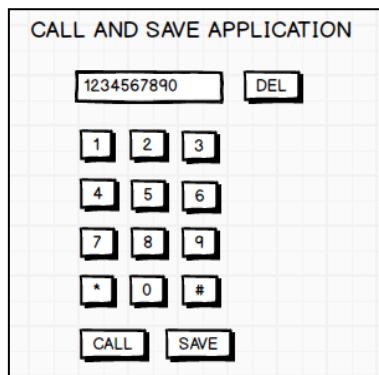
PARSING XML AND JSON DATA	
XML DATA	JSON Data
City_Name: Mysore	City_Name: Mysore
Latitude: 12.295	Latitude: 12.295
Longitude: 76.639	Longitude: 76.639
Temperature: 22	Temperature: 22
Humidity: 90%	Humidity: 90%

**Program 7**

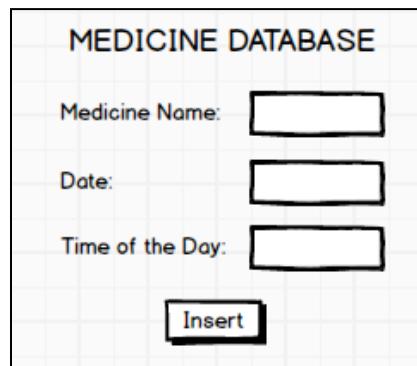
Develop a simple application with one Edit Text so that the user can write some text in it. Create a button called “Convert Text to Speech” that converts the user input text into voice.

**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.

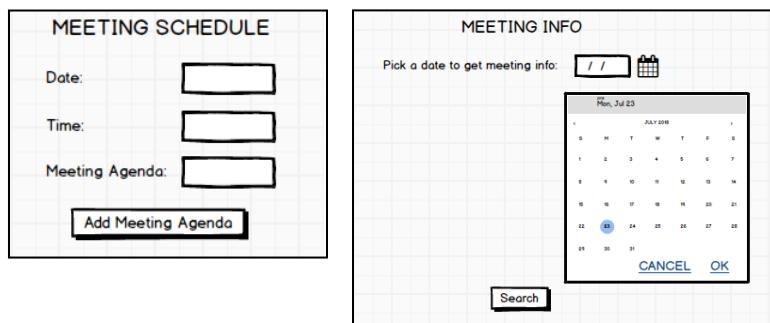
**PART B****Program 1**

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.

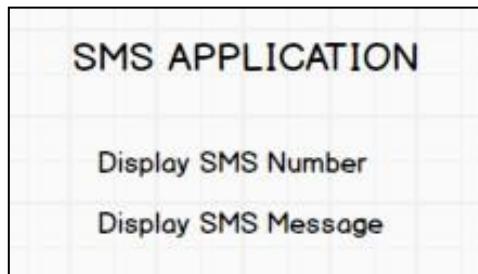


**Program 2**

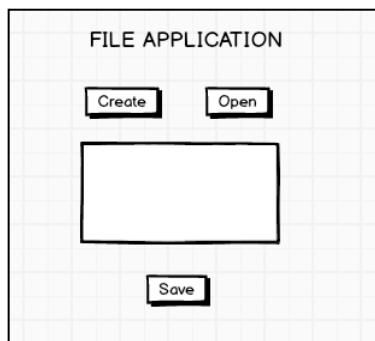
Develop a content provider application with an activity called “Meeting Schedule” which takes Date, Time and Meeting Agenda as input from the user and store this information into the SQLite database. Create another application with an activity called “Meeting Info” having DatePicker control, which on 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.

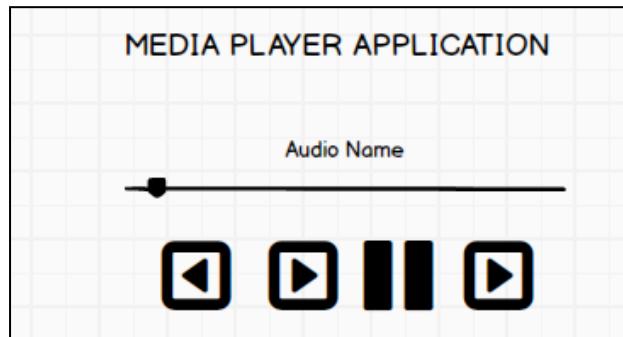
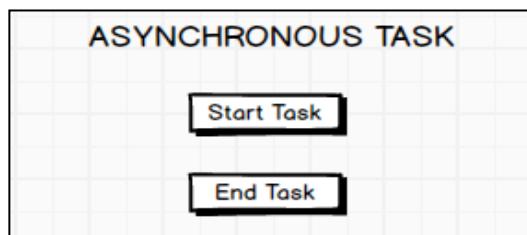
**Program 4**

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**

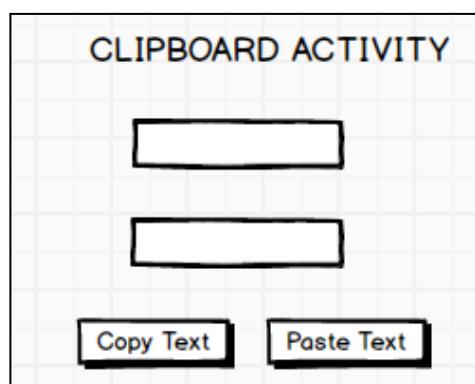
Create an application to demonstrate a basic media player that allows the user to Forward, Backward, Play and Pause an audio. Also, make use of the indicator in the seek bar to move the audio forward or backward as required.

**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

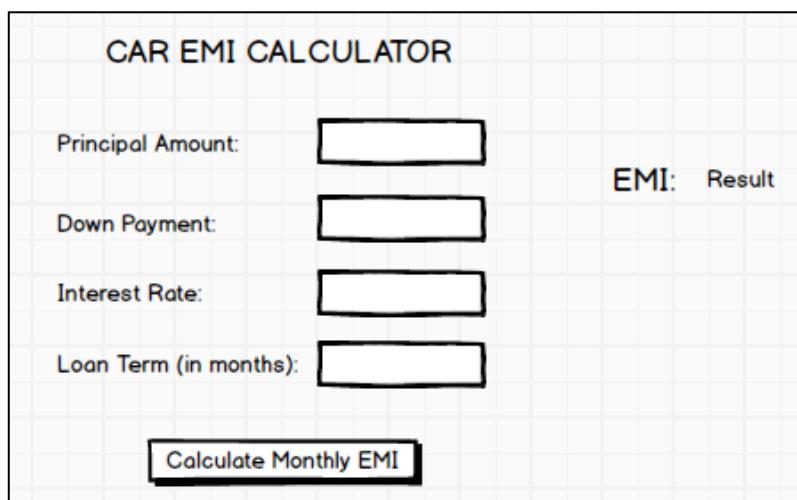
E = The EMI payable on the car loan amount

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.



A wireframe mockup of a mobile application interface titled "CAR EMI CALCULATOR". The interface includes four input fields labeled "Principal Amount", "Down Payment", "Interest Rate", and "Loan Term (in months)". To the right of these fields is a label "EMI: Result". Below the input fields is a button labeled "Calculate Monthly EMI".

### Course Outcomes

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

<b>C323.1</b>	Build, test and debug Android application by setting up Android development environment.
<b>C323.2</b>	Construct adaptive, responsive user interfaces that work across a wide range of devices.
<b>C323.3</b>	Analyze long running tasks and background work in Android applications.
<b>C323.4</b>	Experiment with various methods in storing, sharing and retrieving data in Android applications
<b>C323.5</b>	Inspect the role of permissions and security for Android applications.

### CO-PO Mapping

CO No.	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
<b>C323.1</b>	3	2	2	-	2	1	-	-	1	1	-	1
<b>C323.2</b>	3	2	2	-	2	1	-	-	1	1	-	1
<b>C323.3</b>	3	2	2	-	2	1	-	-	1	1	-	1
<b>C323.4</b>	3	2	2	-	2	1	-	2	1	1	-	1
<b>C323.5</b>	3	2	2	-	2	1	-	2	1	1	-	1

**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 (Course 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
  - i) Part A – Procedure + Execution + Viva =  $6 + 28 + 6 = 40$  Marks
  - ii) Part B – Procedure + Execution + Viva =  $9 + 42 + 9 = 60$  Marks

# 1. Android Studio Tutorials

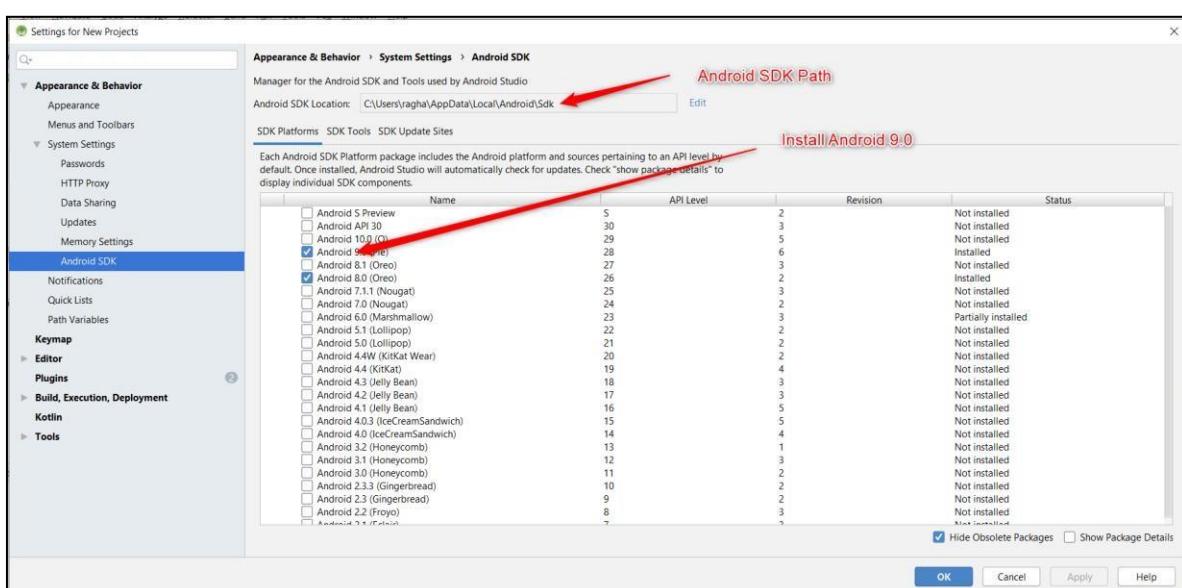
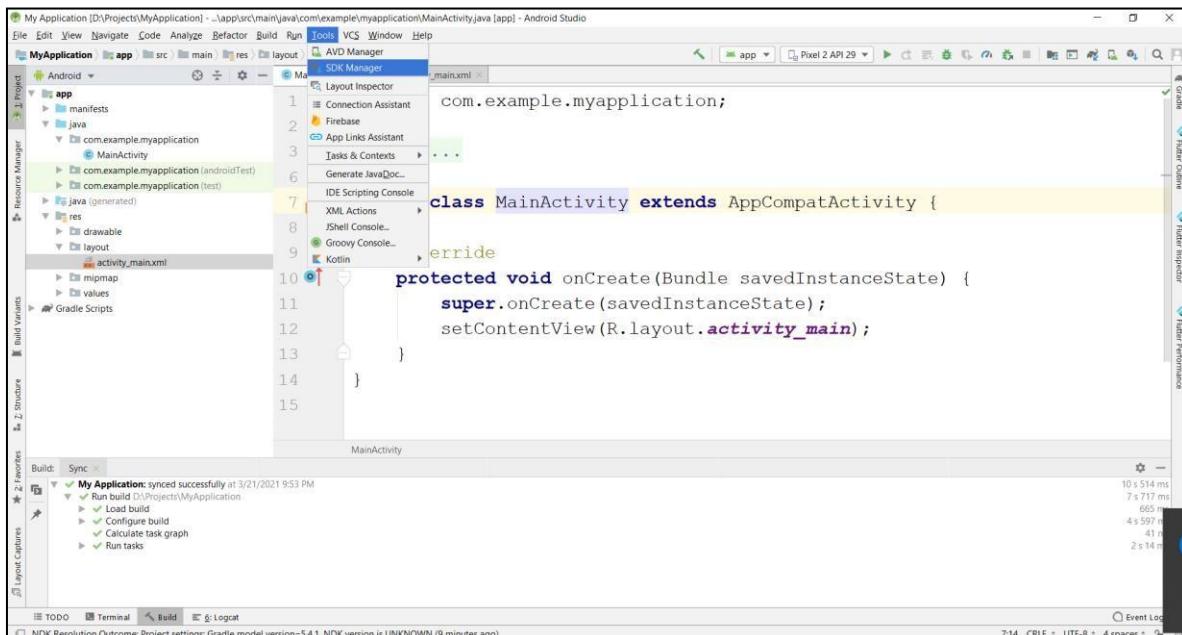
## 1.1 Install Android Studio and Packages:

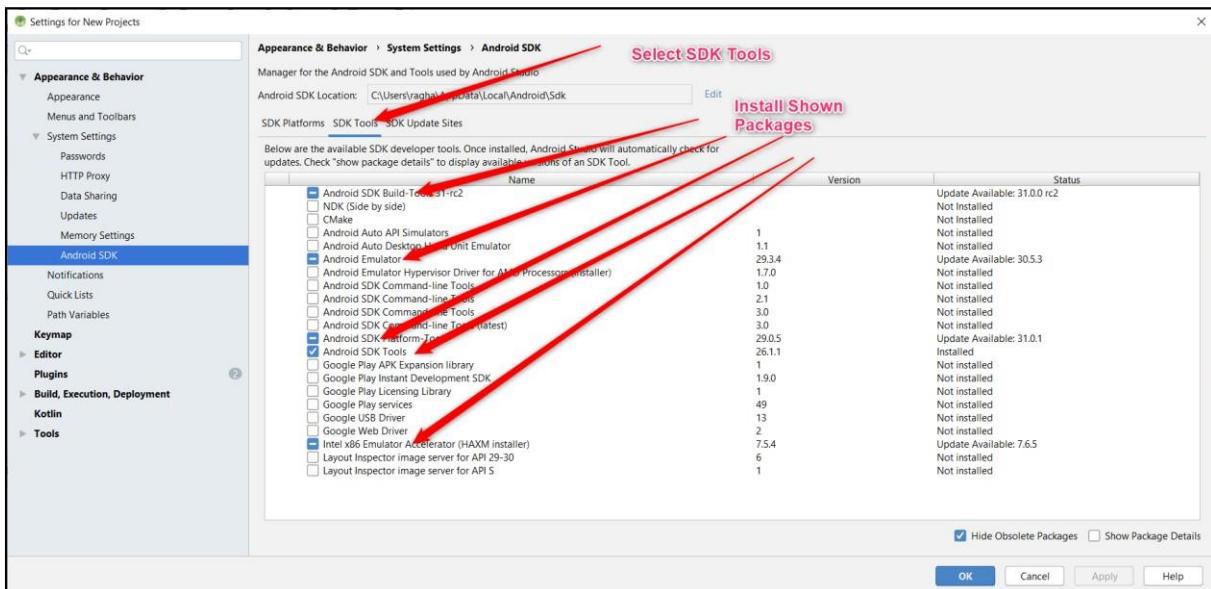
Download Android Version 4.0.2 from the below link

<https://redirector.gvt1.com/edgedl/android/studio/install/4.0.2.0/android-studio-ide-193.6821437-windows.exe>

## 1.2 Configure Android SDK packages:

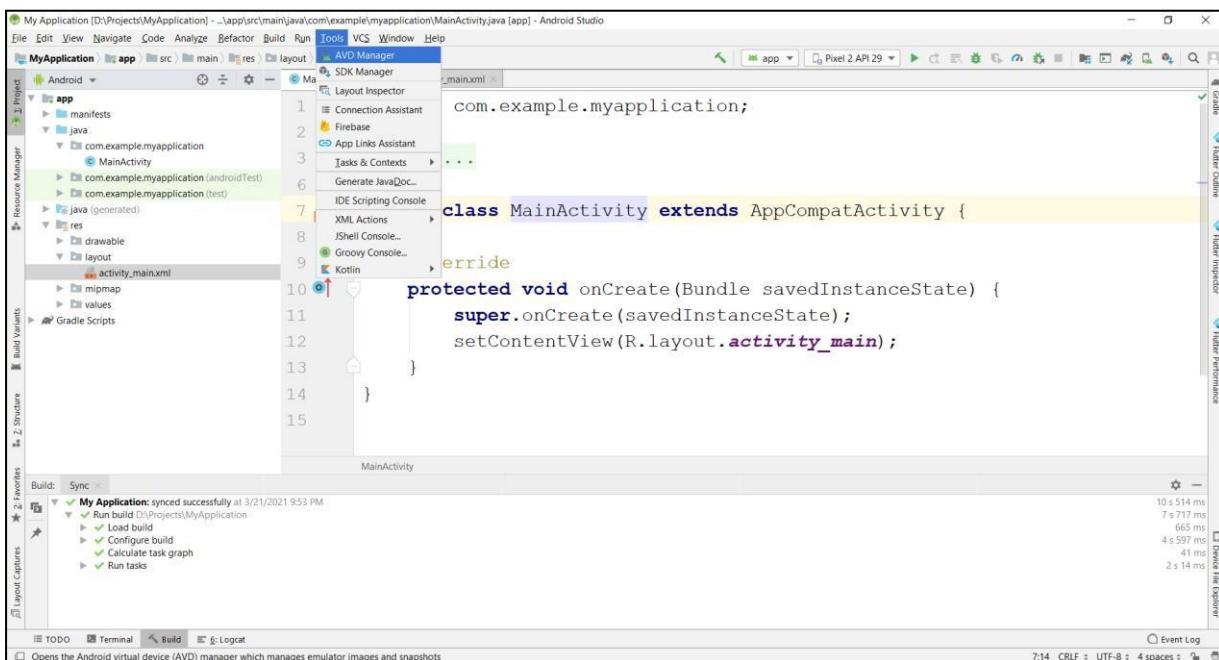
Go to Tools  SDK Manager

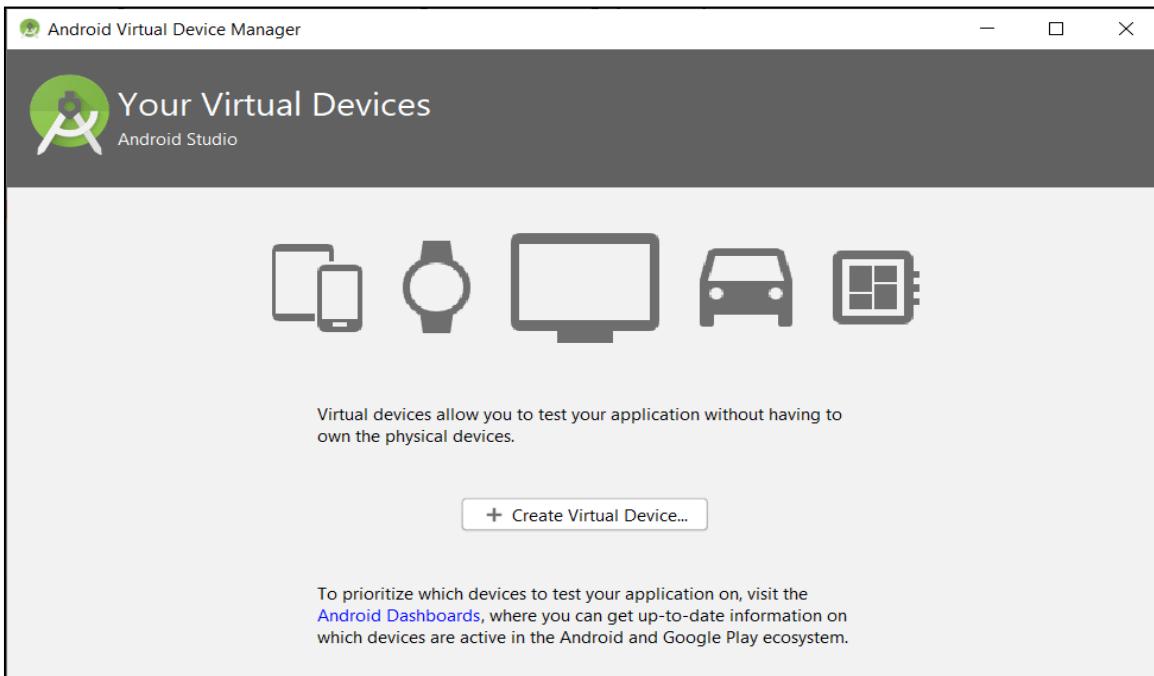




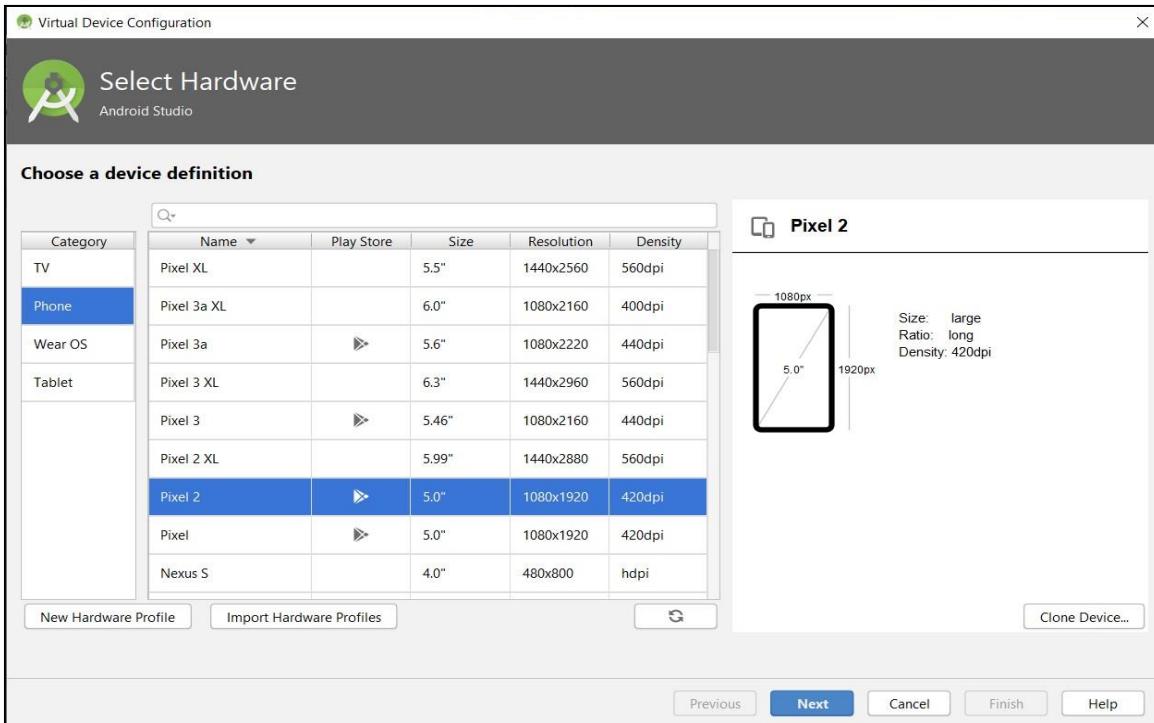
### 1.3 Creating Emulator

Goto Tools Select AVD Manager

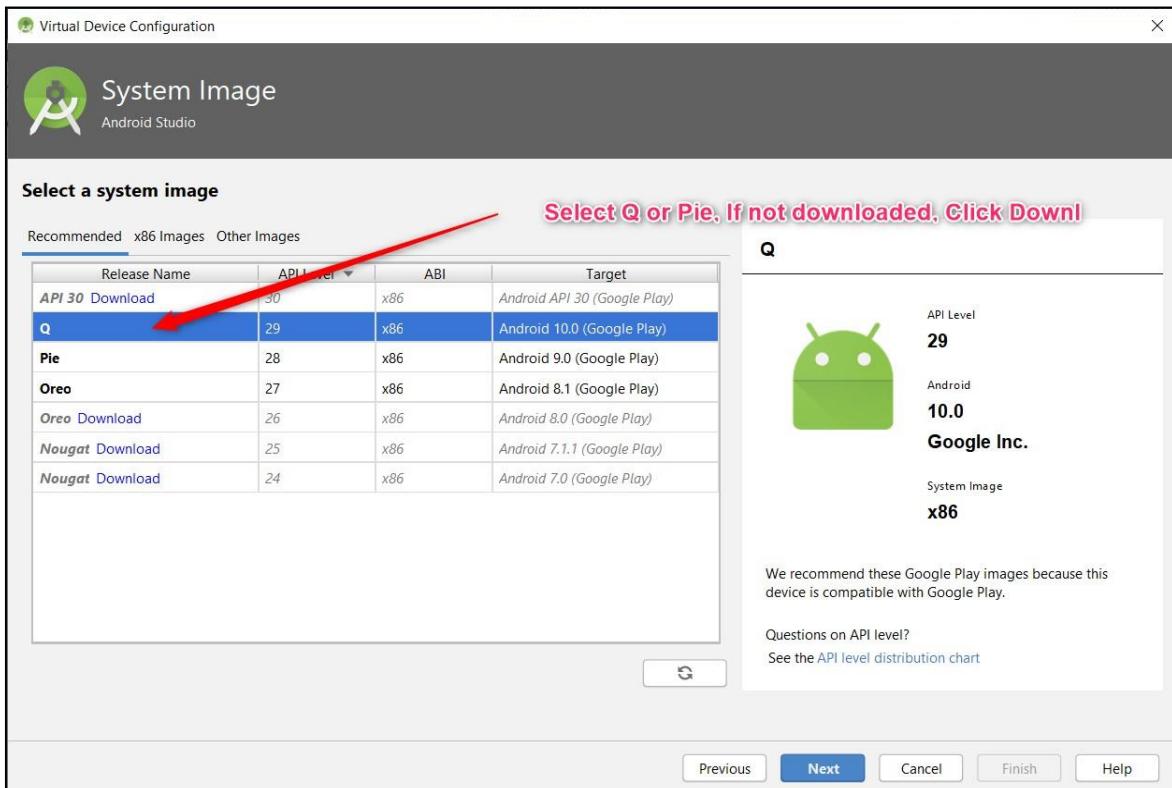




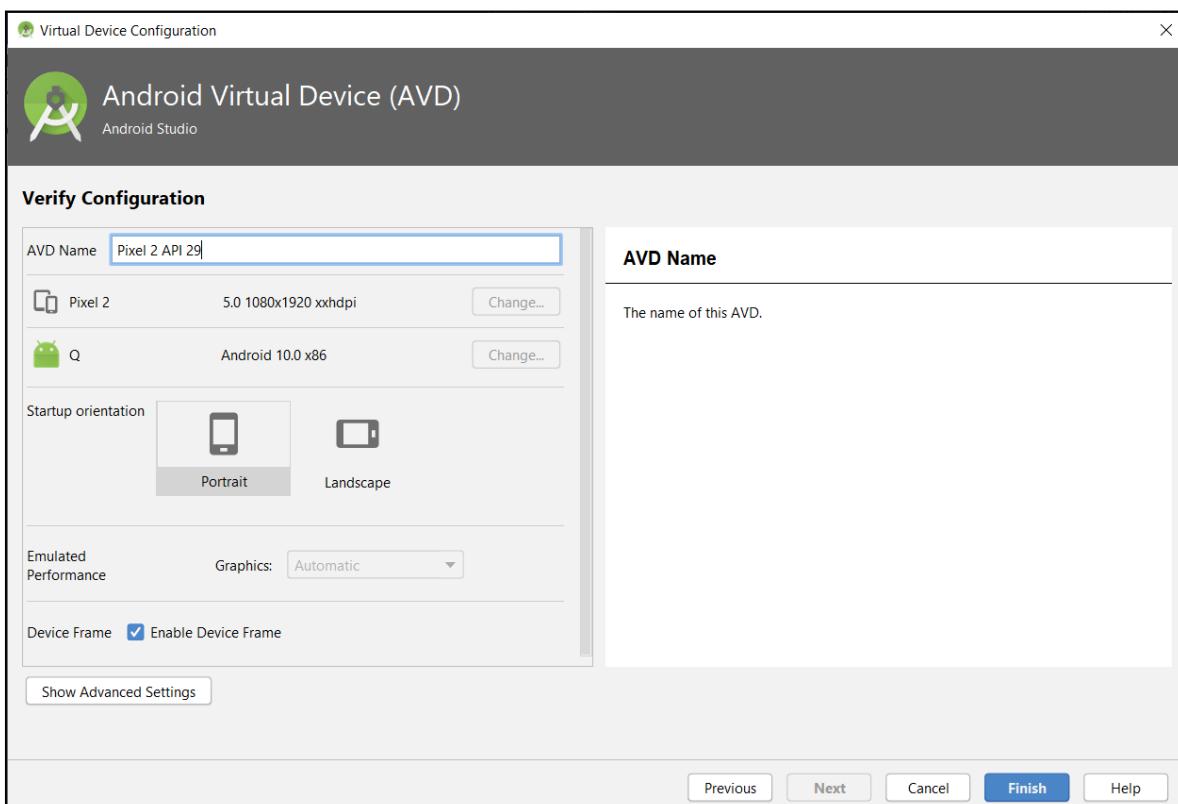
Select Create Virtual Device Select Phone Pixel 2 Press Next



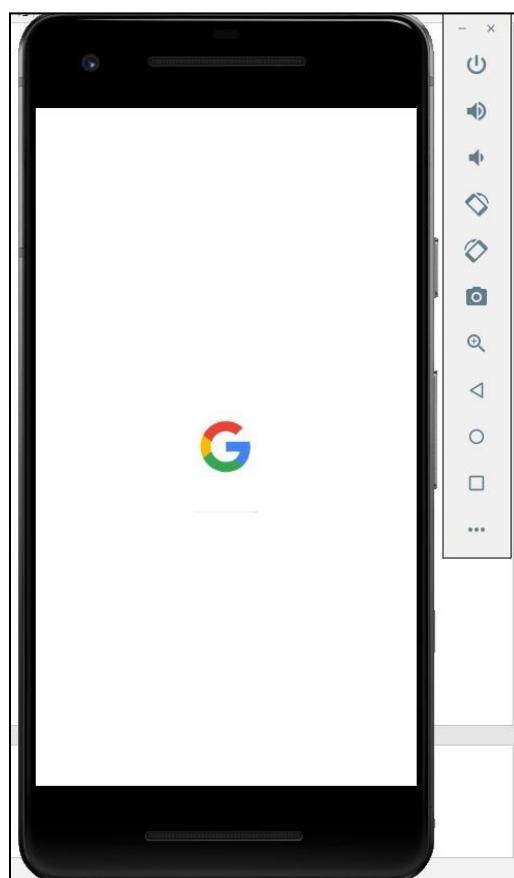
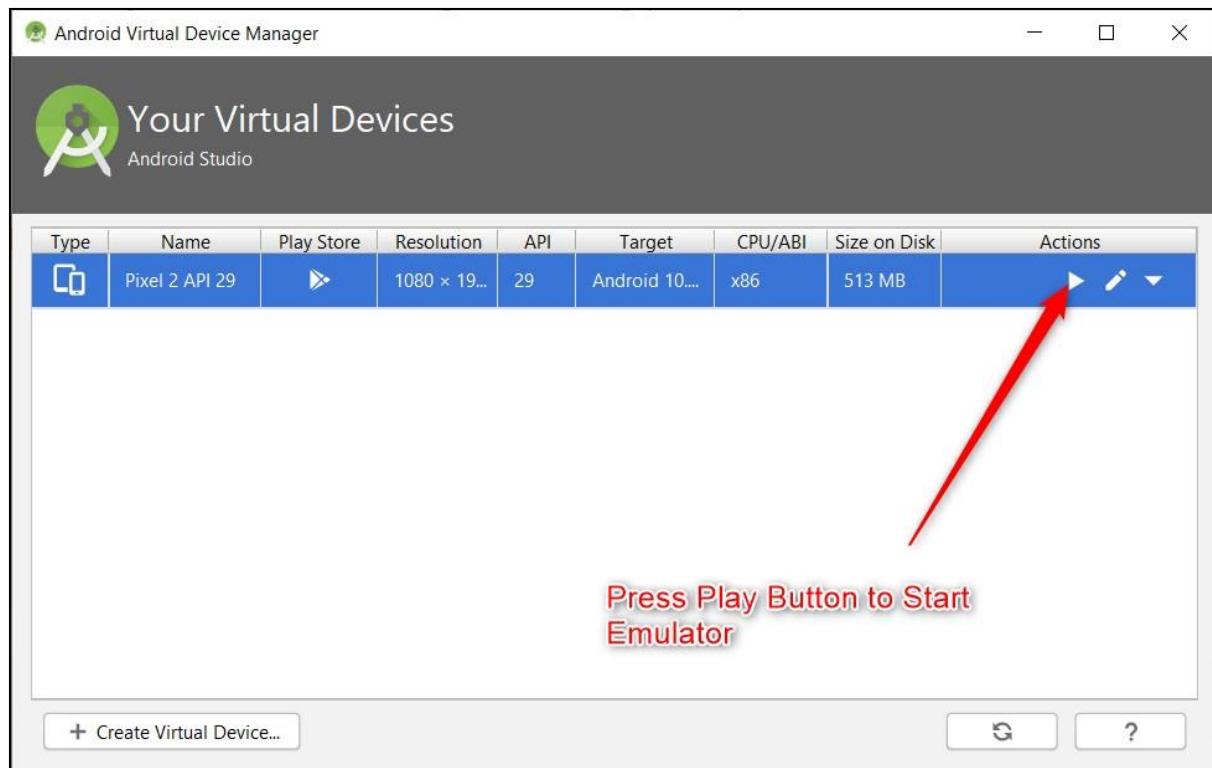
Select Android Q, if not already downloaded press download, After download completes Select Q and Press Next Button.



Enter AVD Name and Press Finish.



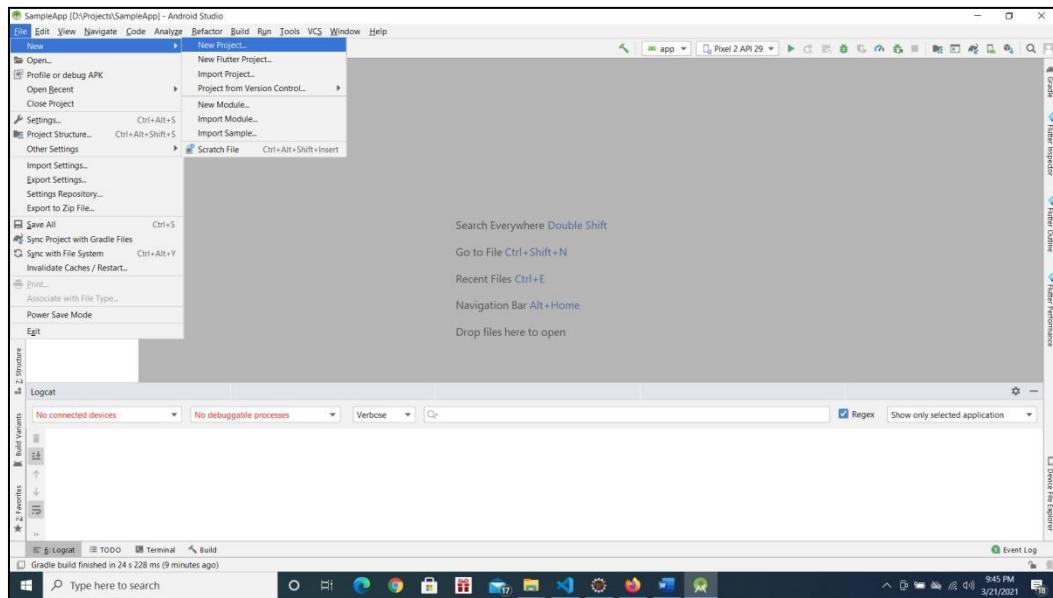
Press Play Button to Start Emulator



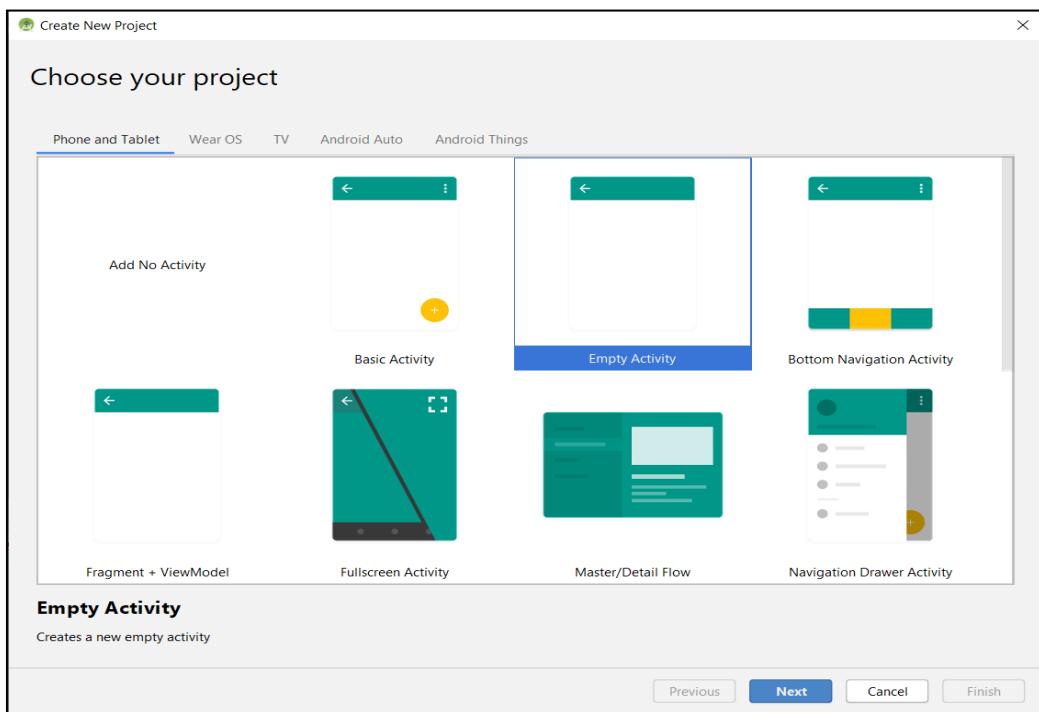
## 1.4 Creating a New Project in Android

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  Screen Icon.

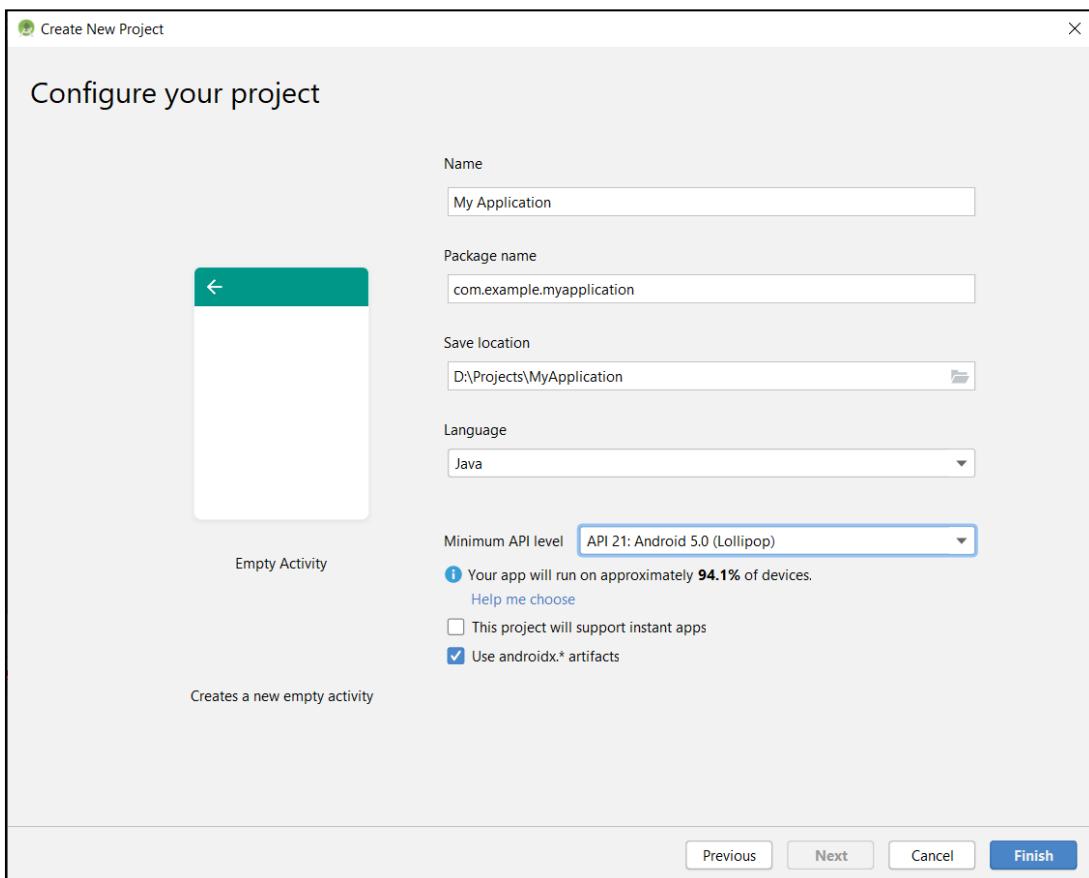
PackageName  Enter package name atleast two identifier (Eg: com.example). Best Practice is 3 or more identifier (Eg: com.example.firstapp).

Save Location  Location where to save the 

Language  Choose Java

Minimum API Level  Android 5.0

Select Checkbox Use androidx.artifacts folder as below screenshot.



## 1.5 Android Project Structure:

The screenshot shows the Android Studio interface. The Project View on the left has 'Android' selected, with 'activity\_main.xml' highlighted. The code editor shows the MainActivity.java file with the following code:

```

package com.example.myapplication;

import ...

public class MainActivity extends AppCompatActivity {

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

```

The bottom status bar indicates 'Emulator: Process finished with exit code 0'.

Select Project Explorer and Select Android from Project View

The screenshot shows the same Android Studio interface as above, but with two red arrows: one pointing to the 'Project Explorer' tab at the top, and another pointing to the 'Android' item in the Project View on the left. The code editor shows the same MainActivity.java file with the same code. The bottom status bar indicates 'Emulator: Process finished with exit code 0'.

## Basic View:

```

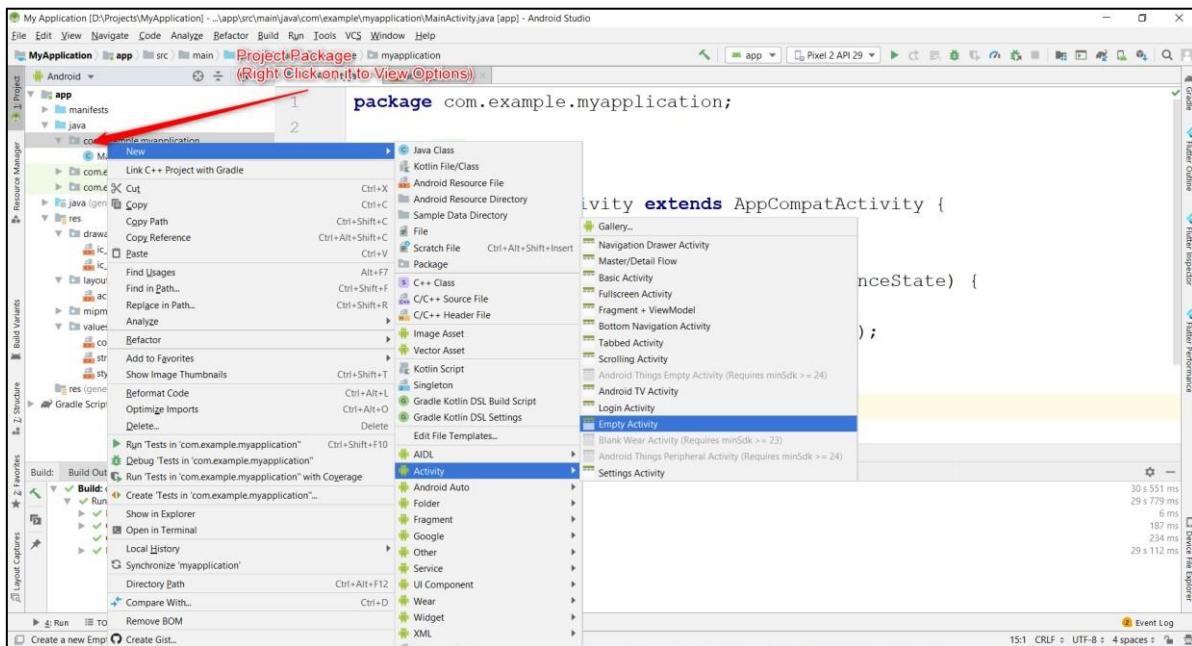
My Application [D:\Projects\MyApplication] - ...app\src\main\java\com\example\myapplication\MainActivity.java [app] - Android Studio
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
MyApplication app src main res layout activity_main.xml
Android Project Resource Manager Build Variants Z-Structure Favorites Layout Captures
1 package com.example.myapplication;
2
3 import ...
4
5 public class MainActivity extends AppCompatActivity {
6
7     @Override
8     protected void onCreate(Bundle savedInstanceState) {
9         super.onCreate(savedInstanceState);
10        setContentView(R.layout.activity_main);
11    }
12
13 }
14
15
Build: Build Output Sync
Build completed successfully at 3/21/2021 10:23 PM
Run build D:\Projects\MyApplication
Load build
Configure build
Calculate task graph
Run tasks
30 s 551 ms
29 s 779 ms
6 ms
187 ms
234 ms
29 s 112 ms
Device file explorer
Event Log
15:1 CRLF : UTF-8 : 4 spaces :
Emulator: Process finished with exit code 0 (3 minutes ago)

```

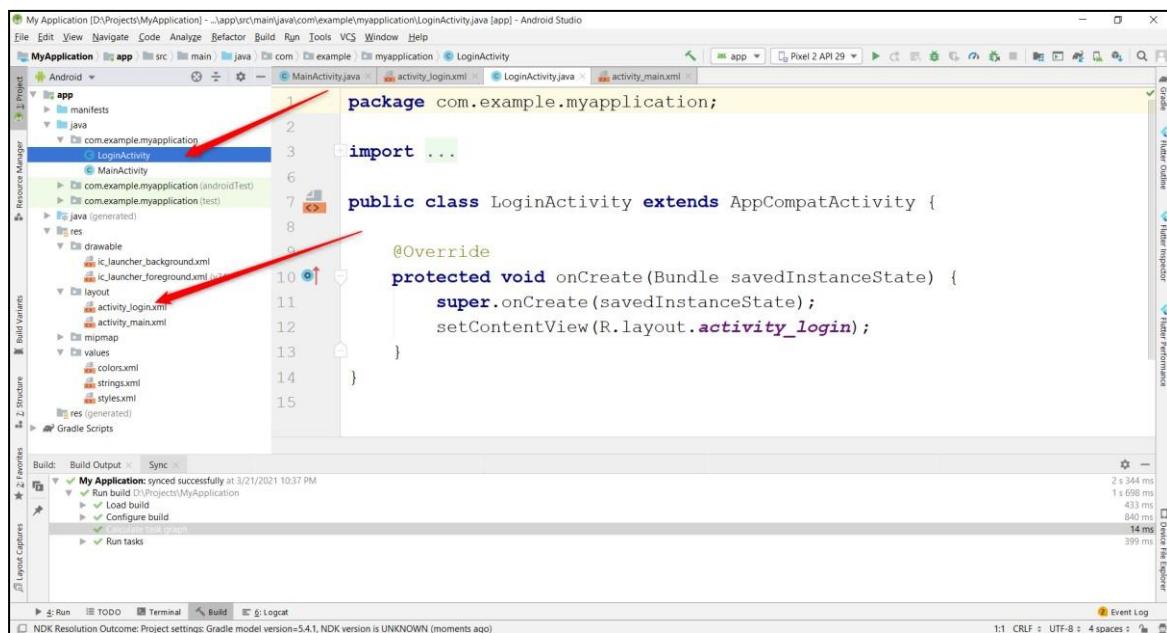
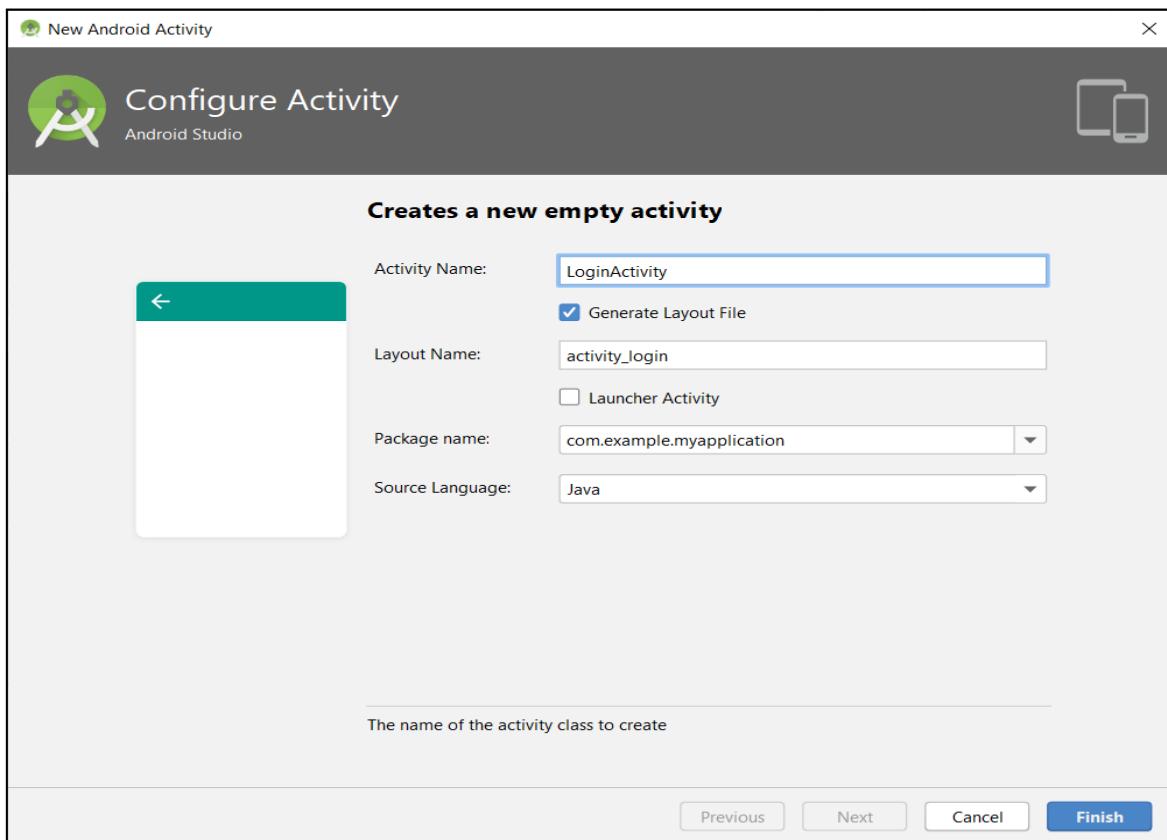
## 2. Importing an Existing Project in Android Studio

## 3. Creating an Activity in Android

Right Click on Package ➔ New ➔ Activity ➔ Empty Activity

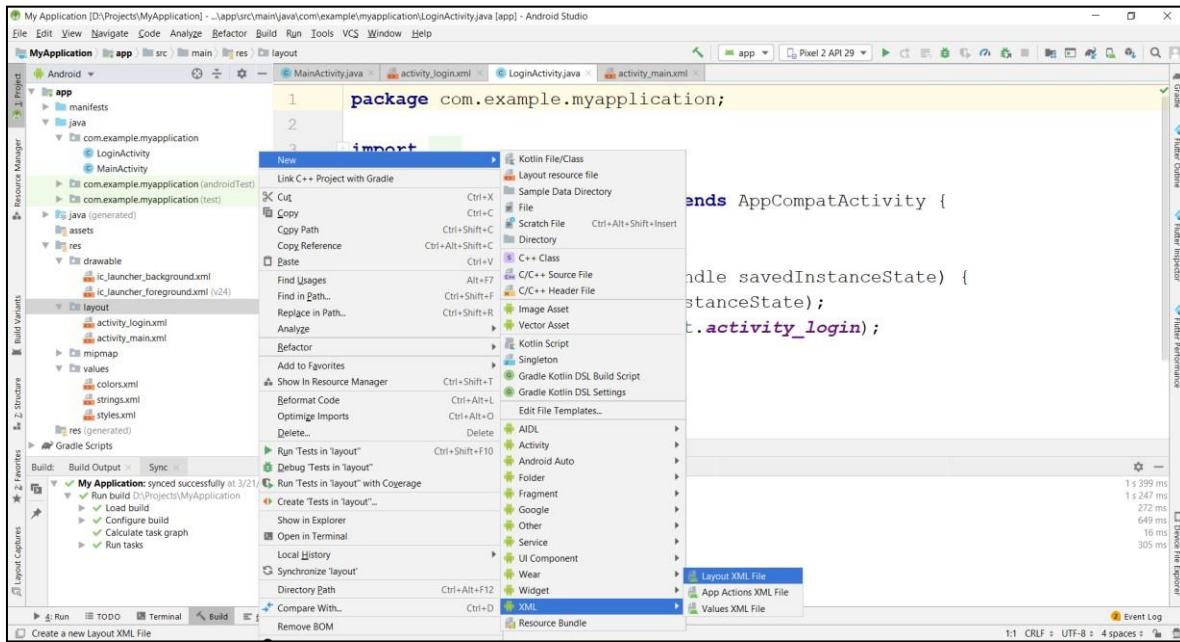


Enter Activity Name and Press Finish

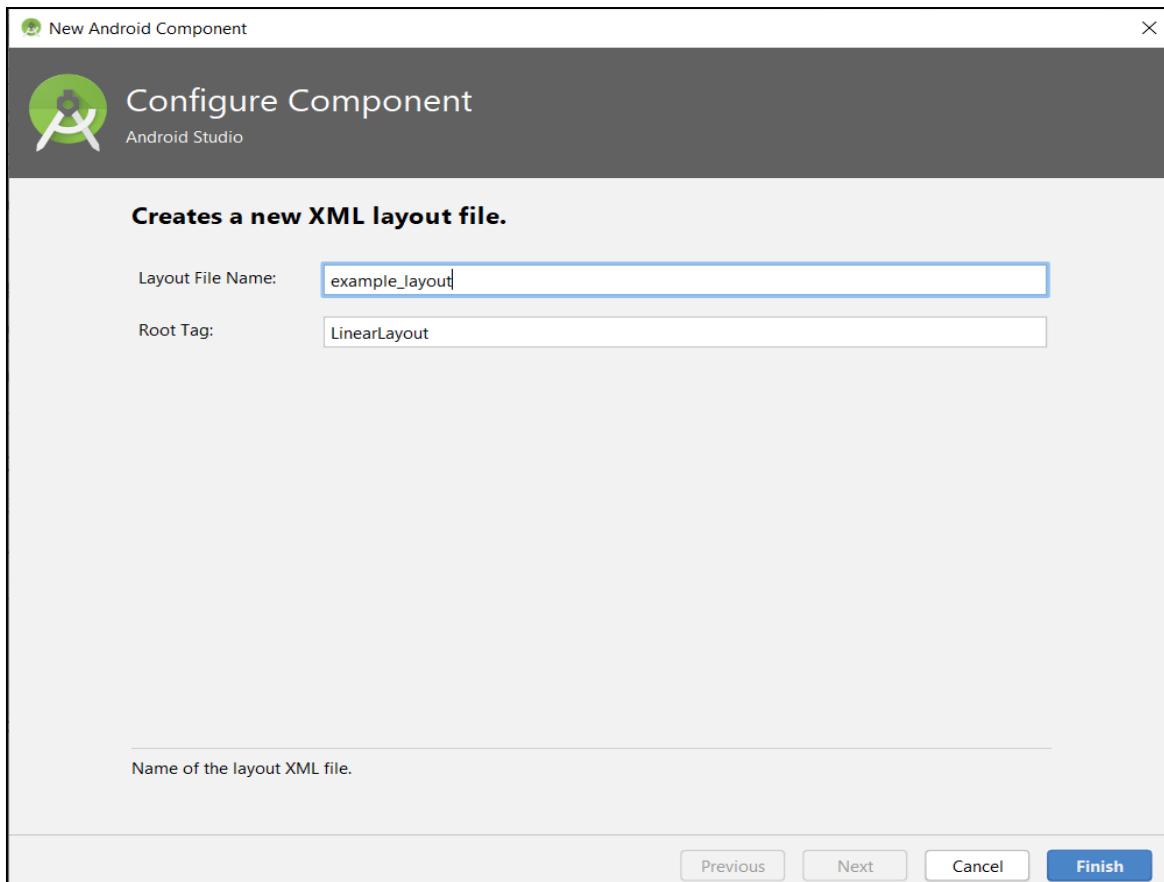


## 1.6 Creating a Layout in Android

Right Click on Layout Folder New XML Layout XML File

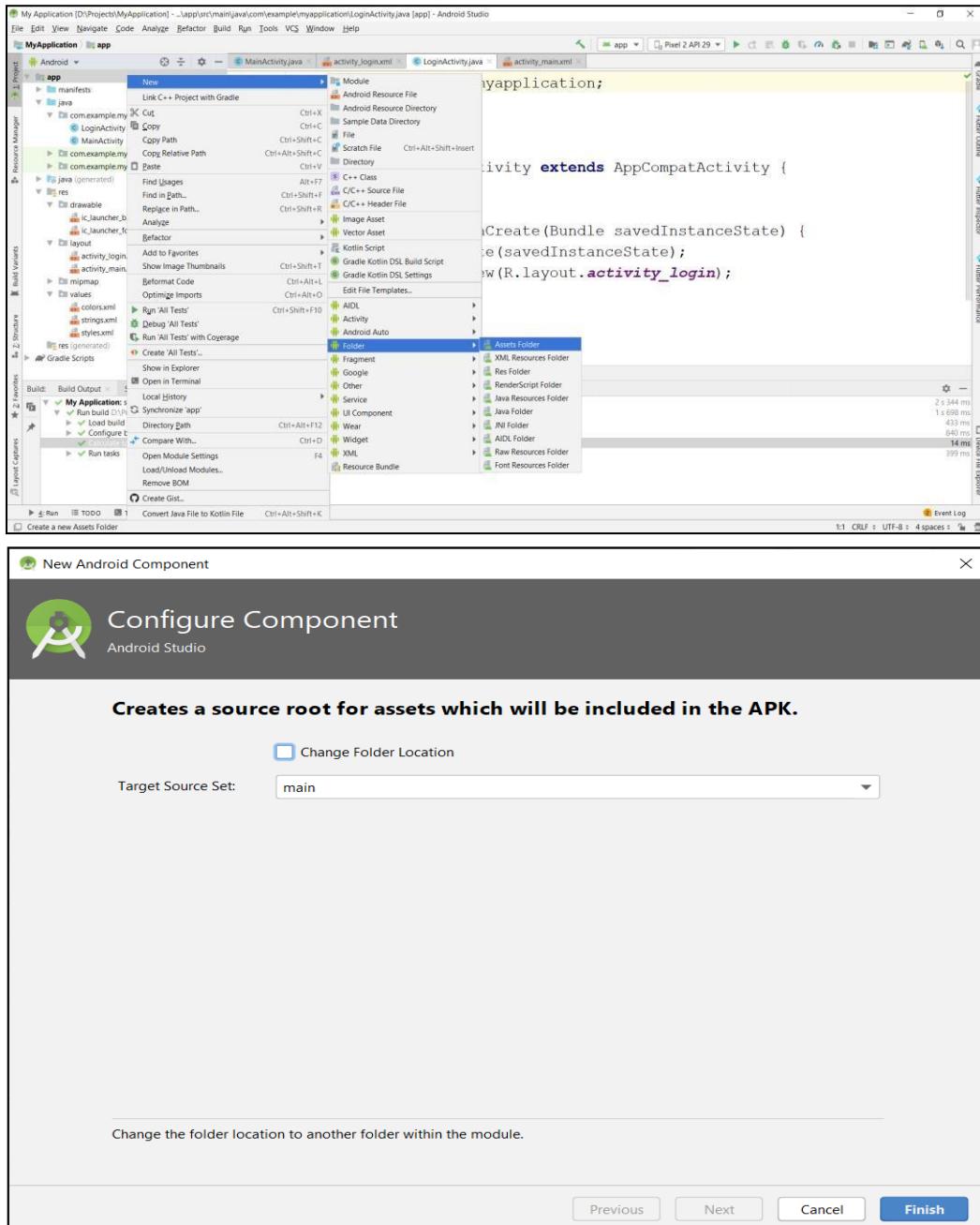


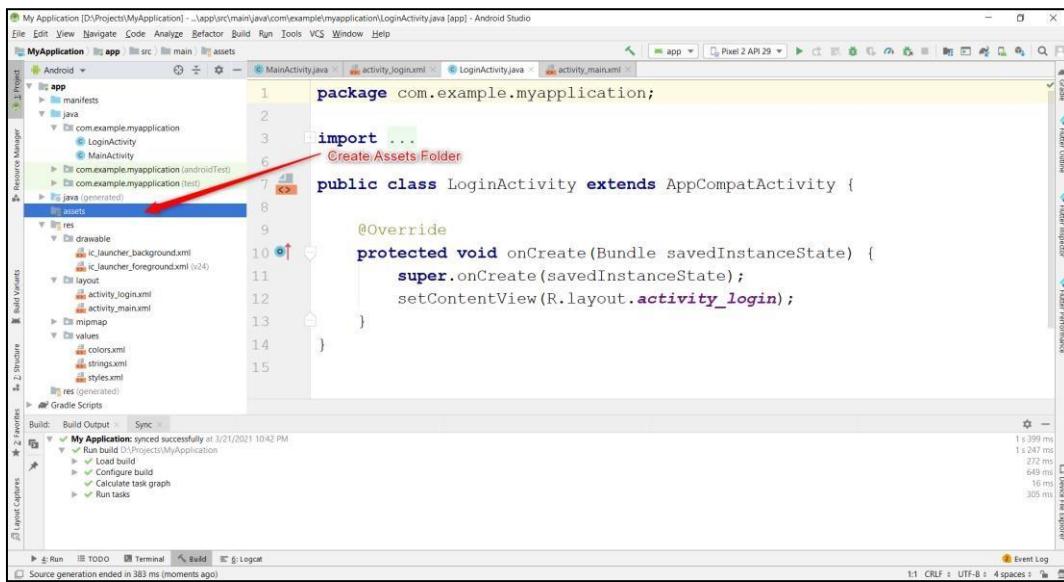
Enter xml file name and press Finish



## 1.7 Creating Assets Folder in Android

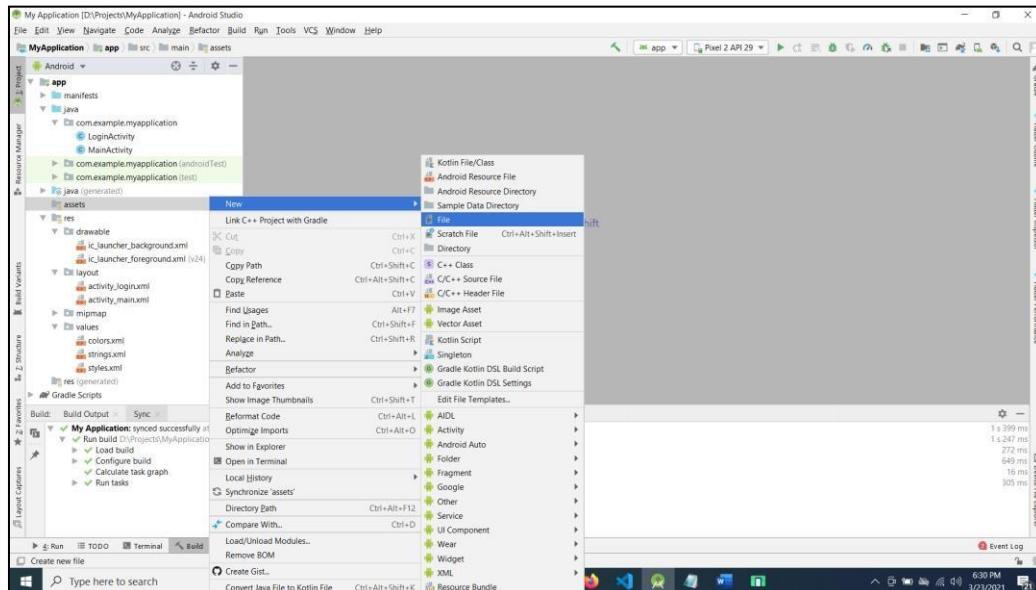
Right Click on app folder New Folder Assets Folder Press Finish Button



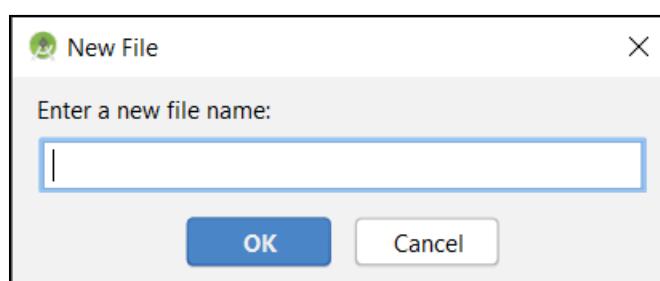


## 1.8 Creating File in assets Folder:

Right Click on assets folder New File



Enter filename with extension (Eg: abc.xml)



## Programs

### PART A

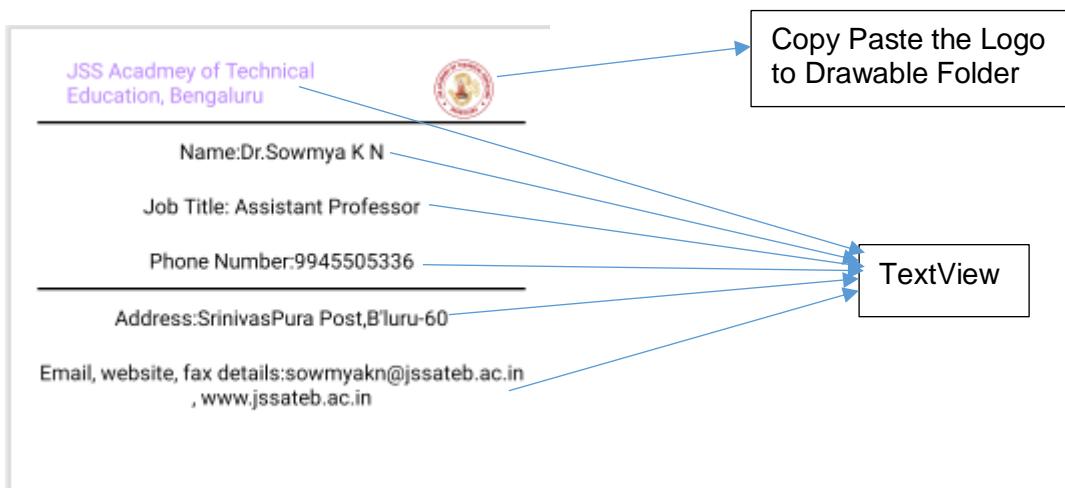
#### Program 1

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 main 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



**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"
    tools:context=".MainActivity"
    android:paddingLeft="20dp"
    android:paddingTop="20dp"
    android:paddingRight="20dp">

    <RelativeLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="COMPANY NAME"
            android:textSize="16dp"
            android:textColor="#000000"
            android:id="@+id/lbl_company_name"
            android:layout_marginRight="10dp" />

        <ImageView
            android:layout_width="50dp"
            android:layout_height="50dp"
            android:src="@drawable/email_icon"
            android:layout_toRightOf="@+id/lbl_company_name"
            android:layout_marginLeft="10dp"/>

    </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="Name:"  
    android:textSize="16dp"  
    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="Job Title:"  
    android:textSize="16dp"  
    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="Phone Number:"  
    android:textSize="16dp"  
    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="Address:"  
    android:textSize="16dp"  
    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, Website, Fax"  
    android:textSize="16dp"  
    android:layout_marginBottom="10dp"  
    android:layout_marginTop="10dp"  
    android:textColor="#000000"  
    android:gravity="center"  
/>
```

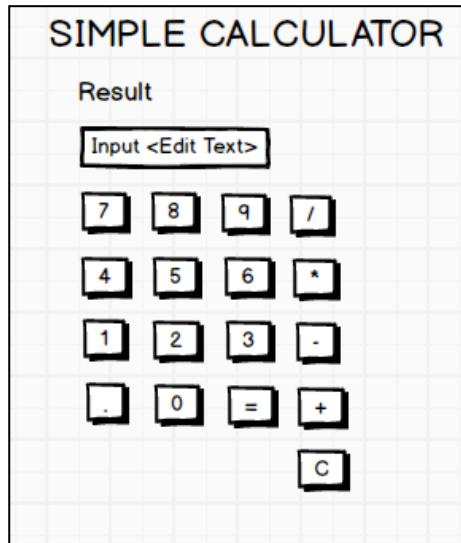
```
</LinearLayout>
```

## Sample Output



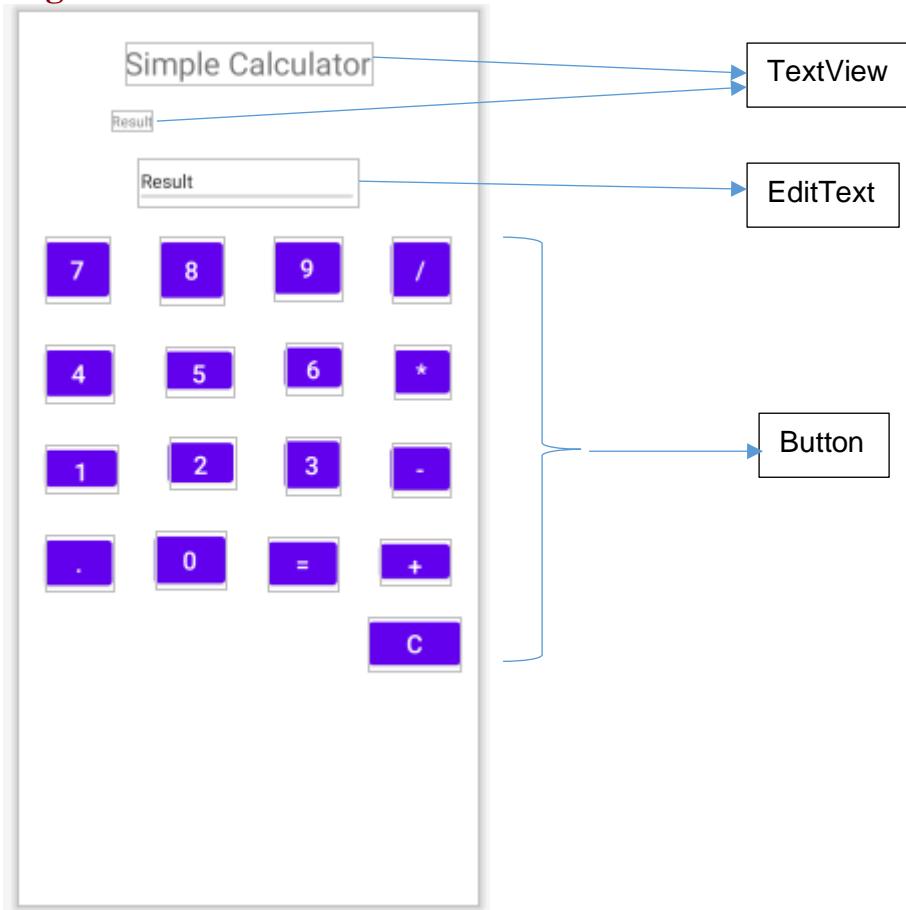
## Program 2

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 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 on 2 operands (Eg: 10+20), If more than 2 operands or wrong input, display invalid input messages.

## 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_clear"
        android:layout_width="90dp"
        android:layout_height="49dp"
        android:layout_marginTop="30dp"
        android:text="C"
        app:layout_constraintStart_toStartOf="@+id	btn_add"
        app:layout_constraintTop_toBottomOf="@+id	btn_add" />

    <Button
        android:id="@+id	btn_sub"
        android:layout_width="90dp"
        android:layout_height="0dp"
        android:layout_marginStart="20dp"
        android:layout_marginTop="30dp"
        android:text="-"
        app:layout_constraintStart_toEndOf="@+id	button_three"
        app:layout_constraintTop_toBottomOf="@+id	btn_mul" />

    <Button
        android:id="@+id	btn_add"
        android:layout_width="90dp"
        android:layout_height="0dp"
        android:layout_marginStart="20dp"
        android:layout_marginTop="30dp"
        android:text="+"
        app:layout_constraintStart_toEndOf="@+id/button_equal"
        app:layout_constraintTop_toBottomOf="@+id	btn_sub" />

    <Button
        android:id="@+id	btn_mul"
        android:layout_width="90dp"
```

```
    android:layout_height="0dp"
    android:layout_marginStart="20dp"
    android:layout_marginTop="30dp"
    android:text="*"
    app:layout_constraintStart_toEndOf="@+id/button_six"
    app:layout_constraintTop_toBottomOf="@+id(btn_div)" />
```

```
<Button
    android:id="@+id/button_equal"
    android:layout_width="55dp"
    android:layout_height="41dp"
    android:layout_marginStart="20dp"
    android:layout_marginTop="30dp"
    android:text ="="
    app:layout_constraintStart_toEndOf="@+id/button_zero"
    app:layout_constraintTop_toBottomOf="@+id/button_three" />
```

```
<Button
    android:id="@+id/button_zero"
    android:layout_width="55dp"
    android:layout_height="41dp"
    android:layout_marginStart="20dp"
    android:layout_marginTop="30dp"
    android:text="0"
    app:layout_constraintStart_toEndOf="@+id/button_dot"
    app:layout_constraintTop_toBottomOf="@+id(btn_two)" />
```

```
<Button
    android:id="@+id/button_dot"
    android:layout_width="55dp"
    android:layout_height="41dp"
    android:layout_marginStart="20dp"
    android:layout_marginTop="30dp"
    android:text ="."
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id	btn_one" />
```

```
<Button
    android:id="@+id/button_three"
    android:layout_width="55dp"
    android:layout_height="41dp"
    android:layout_marginStart="20dp"
    android:layout_marginTop="30dp"
    android:text="3"
```

```
app:layout_constraintStart_toEndOf="@+id(btn_two)"  
app:layout_constraintTop_toBottomOf="@+id(button_six) />
```

```
<Button  
    android:id="@+id(btn_two)"  
    android:layout_width="55dp"  
    android:layout_height="41dp"  
    android:layout_marginStart="20dp"  
    android:layout_marginTop="30dp"  
    android:text="2"  
    app:layout_constraintStart_toEndOf="@+id(btn_one)"  
    app:layout_constraintTop_toBottomOf="@+id(button_five) />
```

```
<Button  
    android:id="@+id(btn_one)"  
    android:layout_width="55dp"  
    android:layout_height="41dp"  
    android:layout_marginStart="20dp"  
    android:layout_marginTop="30dp"  
    android:text="1"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toBottomOf="@+id(button_four) />
```

```
<Button  
    android:id="@+id(button_six)"  
    android:layout_width="55dp"  
    android:layout_height="41dp"  
    android:layout_marginStart="20dp"  
    android:layout_marginTop="30dp"  
    android:text="6"  
    app:layout_constraintStart_toEndOf="@+id(button_five)"  
    app:layout_constraintTop_toBottomOf="@+id(button_nine) />
```

```
<Button  
    android:id="@+id(btn_seven)"  
    android:layout_width="62dp"  
    android:layout_height="53dp"  
    android:layout_marginStart="20dp"  
    android:layout_marginTop="20dp"  
    android:text="7"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toBottomOf="@+id(txt_result) />
```

```
<Button
```

```
    android:id="@+id	btn_eight"
    android:layout_width="56dp"
    android:layout_height="51dp"
    android:layout_marginStart="20dp"
    android:layout_marginTop="20dp"
    android:text="8"
    app:layout_constraintStart_toEndOf="@+id	btn_seven"
    app:layout_constraintTop_toBottomOf="@+id	txt_result" />
```

```
<Button
    android:id="@+id/button_nine"
    android:layout_width="64dp"
    android:layout_height="53dp"
    android:layout_marginStart="20dp"
    android:layout_marginTop="20dp"
    android:text="8"
    app:layout_constraintStart_toEndOf="@+id	btn_eight"
    app:layout_constraintTop_toBottomOf="@+id	txt_result" />
```

```
<Button
    android:id="@+id/button_four"
    android:layout_width="57dp"
    android:layout_height="41dp"
    android:layout_marginStart="20dp"
    android:layout_marginTop="30dp"
    android:text="4"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id	btn_seven" />
```

```
<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="30dp"
    android:text="Calculator"
    android:textSize="26dp"
    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="20dp"
android:layout_marginTop="20dp"
android:text="Result"
android:textSize="18dp"
android:textStyle="bold"
app:layout_constraintEnd_toStartOf="@+id/textView"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/textView" />

```

```

<EditText
    android:id="@+id/txt_result"
    android:layout_width="310dp"
    android:layout_height="46dp"
    android:layout_marginTop="20dp"
    android:ems="10"
    android:inputType="textPersonName"
    android:text="Name"
    app:layout_constraintStart_toStartOf="@+id/textView2"
    app:layout_constraintTop_toBottomOf="@+id/textView2" />

```

```

<Button
    android:id="@+id	btn_div"
    android:layout_width="87dp"
    android:layout_height="49dp"
    android:layout_marginStart="20dp"
    android:layout_marginTop="20dp"
    android:text="/"
    app:layout_constraintStart_toEndOf="@+id/button_nine"
    app:layout_constraintTop_toBottomOf="@+id/txt_result" />

```

```

<Button
    android:id="@+id/button_five"
    android:layout_width="55dp"
    android:layout_height="41dp"
    android:layout_marginStart="20dp"
    android:layout_marginTop="30dp"
    android:text="5"
    app:layout_constraintStart_toEndOf="@+id/button_four"
    app:layout_constraintTop_toBottomOf="@+id/btn_eight" />
</androidx.constraintlayout.widget.ConstraintLayout>

```

## MainActivity.java

```
package com.example.partaprogram2;
```

```
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;
```

```
import java.util.regex.Pattern;
```

```
public class MainActivity extends AppCompatActivity implements
View.OnClickListener {
```

```
    Button btnOne, btnTwo, btnThree, btnFour, btnFive, btnSix;
    Button btnSeven, btnEight, btnNine, btnZero;
```

```
    Button add,sub,mul,div;
    Button btnClear,btnEqual,btnDot;
```

```
    EditText txtResult;
```

```
@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.button_nine);
    btnThree.setOnClickListener(this);
```

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

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

```
    btnSix=(Button)findViewById(R.id.button_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.button_nine);
btnNine.setOnClickListener(this);

add=(Button)findViewById(R.id.btn_add);
add.setOnClickListener(this);

sub=(Button)findViewById(R.id.btn_sub);
sub.setOnClickListener(this);

mul=(Button)findViewById(R.id.btn_mul);
mul.setOnClickListener(this);

div=(Button)findViewById(R.id.btn_div);
div.setOnClickListener(this);

btnClear=(Button)findViewById(R.id.btn_clear);
btnClear.setOnClickListener(this);

btnEqual=(Button)findViewById(R.id.button_equal);
btnEqual.setOnClickListener(this);

btnDot=(Button)findViewById(R.id.button_dot);
btnDot.setOnClickListener(this);

txtResult=(EditText)findViewById(R.id.txt_result);

txtResult.setText("");


}

public void onClick(View v)
{
    if(v.equals(btnOne))
    {
        txtResult.append("1");
    }
}
```

```
        }
        if(v.equals(btnTwo))
        {
            txtResult.append("2");
        }
        if(v.equals(btnThree))
        {
            txtResult.append("3");
        }
        if(v.equals(btnFour))
        {
            txtResult.append("4");
        }
        if(v.equals(btnFive))
        {
            txtResult.append("5");
        }
        if(v.equals(btnSix))
        {
            txtResult.append("6");
        }
        if(v.equals(btnSeven))
        {
            txtResult.append("7");
        }
        if(v.equals(btnEight))
        {
            txtResult.append("8");
        }
        if(v.equals(btnNine))
        {
            txtResult.append("9");
        }
        if(v.equals(btnZero))
        {
            txtResult.append("0");
        }
        if(v.equals(btnClear))
        {
            txtResult.setText("");
        }
        if(v.equals(btnEqual))
        {
            try {
```

```
String data = txtResult.getText().toString();
if (data.contains("/")) {
    String[] operands = data.split("/");
    if(operands.length==2) {
        double operand1 = Double.parseDouble(operands[0]);
        double operand2 = Double.parseDouble(operands[1]);
        double result = operand1 / operand2;
        txtResult.setText(String.valueOf(result));
    }
    else
    {
        Toast.makeText(getApplicationContext(),
            "Invalid Input",
            Toast.LENGTH_LONG).show();
    }
}

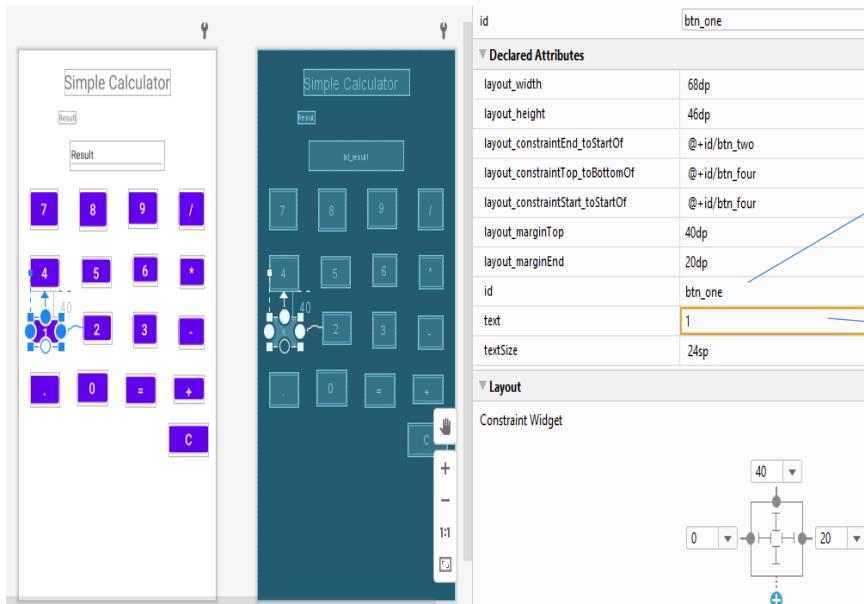
} else if (data.contains("*")) {
    String[] operands = data.split(Pattern.quote("*"));
    if(operands.length==2) {
        double operand1 = Double.parseDouble(operands[0]);
        double operand2 = Double.parseDouble(operands[1]);
        double result = operand1 * operand2;
        txtResult.setText(String.valueOf(result));
    }
    else
    {
        Toast.makeText(getApplicationContext(),
            "Invalid Input",
            Toast.LENGTH_LONG).show();
    }
}

} else if (data.contains("+")) {
    String[] operands = data.split(Pattern.quote("+"));
    if(operands.length==2) {
        double operand1 = Double.parseDouble(operands[0]);
        double operand2 = Double.parseDouble(operands[1]);
        double result = operand1 + operand2;
        txtResult.setText(String.valueOf(result));
    }
    else
    {
        Toast.makeText(getApplicationContext(),
            "Invalid Input",
            Toast.LENGTH_LONG).show();
    }
}
```

```
        }
```

```
    } else if (data.contains("-")) {
        String[] operands = data.split("-");
        if(operands.length==2) {
            double operand1 = Double.parseDouble(operands[0]);
            double operand2 = Double.parseDouble(operands[1]);
            double result = operand1 - operand2;
            txtResult.setText(String.valueOf(result));
        }
        else
        {
            Toast.makeText(getApplicationContext(),
                    "Invalid Input",
                    Toast.LENGTH_LONG).show();
        }
    }
    catch(Exception e)
    {
        Toast.makeText(getApplicationContext(),
                    "Invalid Input",
                    Toast.LENGTH_LONG).show();
    }
}
if(v.equals(add))
{
    txtResult.append("+");
}
if(v.equals(sub))
{
    txtResult.append("-");
}
if(v.equals(mul))
{
    txtResult.append("*");
}
if(v.equals(div))
{
    txtResult.append("/");
}
```

}



Type id value  
here i.e.,btn One

Type the value which needs to displayed in app i.e., 1

This needs to be done for each Button and Plain Text

```
package com.example.calculator3;
```

```
[ ] import ...
```

Implements  
View.OnClickList  
ener

```
public class MainActivity extends AppCompatActivity {
```

Button  
btnOne;

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

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

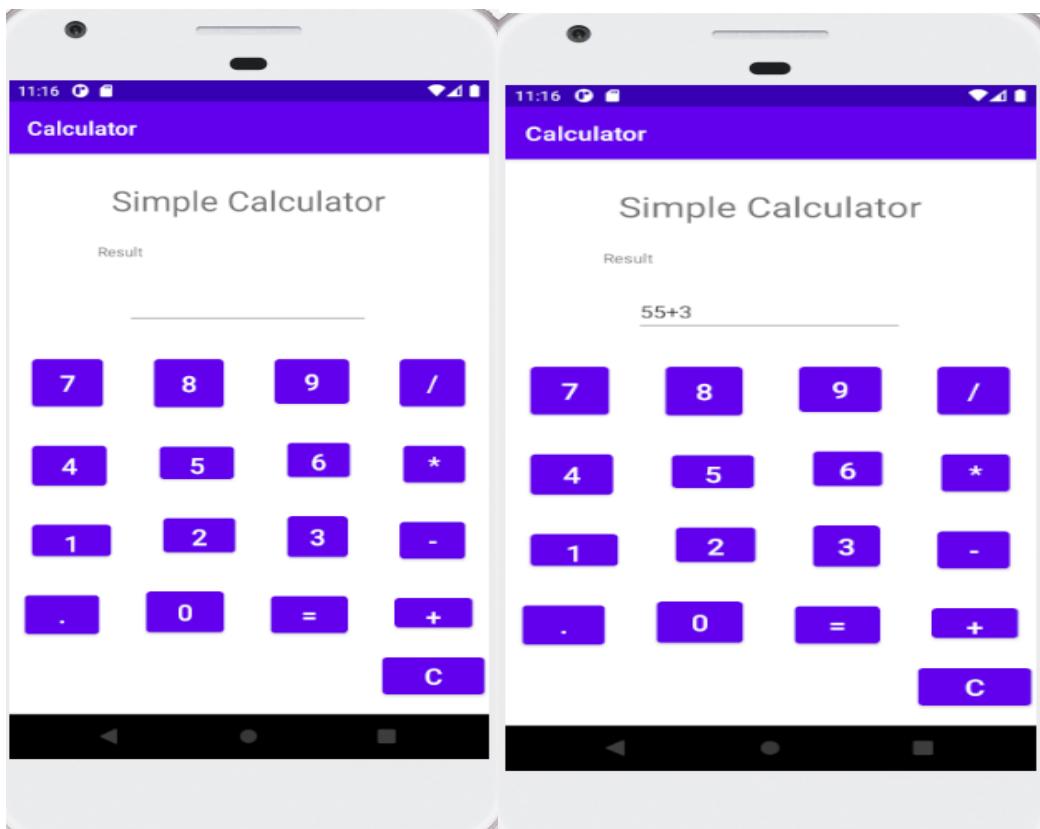
```

if(v.equals(btnEqual))
{
    try {
        String data = txtResult.getText().toString();
        if (data.contains("/")) {
            String[] operands = data.split( regex: "/");
            if(operands.length==2) {
                double operand1 = Double.parseDouble(operands[0]);
                double operand2 = Double.parseDouble(operands[1]);
                double result = operand1 / operand2;
                txtResult.setText(String.valueOf(result));
            }
            else
            {
                Toast.makeText(getApplicationContext(),
                    text: "Invalid Input",
                    Toast.LENGTH_LONG).show();
            }
        }
    }
}

```

- 1) Android Toast can be used to display information for the short period of time.
- 2) A toast contains message to be displayed quickly and disappears after sometime.
- 3) The android.widget.Toast class is the subclass of java.lang.Object class.

## Sample Output

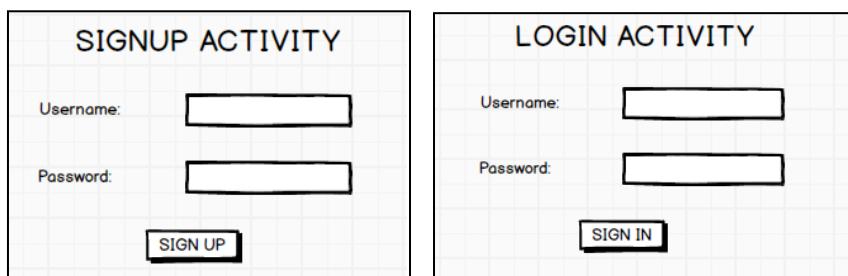


### Program 3

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

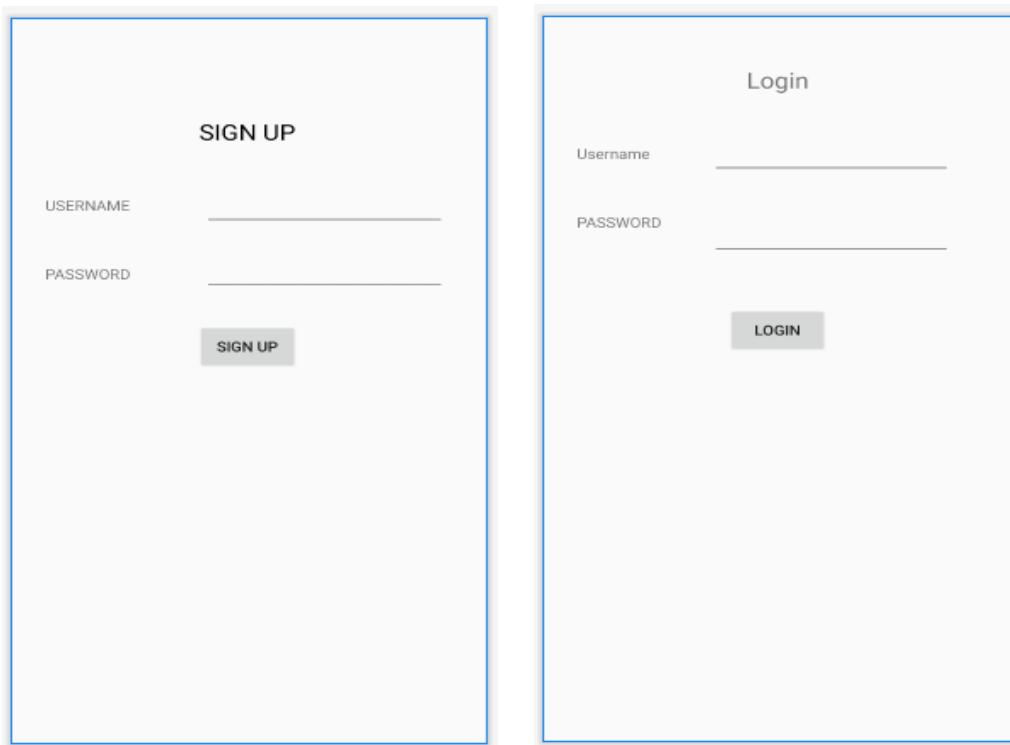
- Password should contain uppercase and lowercase letters.
- Password should contain letters and numbers.
- Password should contain special characters.
- Minimum length of the password (the default value is 8).

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.



1. Create a New Android Project with Empty Activity.
2. Open activity\_main.xml file from res layout folder, check/add Constraint Layout as 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 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="@android:color/background_dark"
    android:textSize="22dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    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="68dp"
    android:layout_height="0dp"
    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"
    android:text="Name"
    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**

```

<?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"
    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" />

```

```

    android:text="Name"
    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.example.parta.program3;

import androidx.appcompat.app.AppCompatActivity;

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

```

```
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]*)([0-9]*)([@$!])[A-Za-z\\d@$!]{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("pass", password);

            Intent it = new Intent(this, LoginActivity.class);
            it.putExtra("data", bundle);
        }
    }

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

```

        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**

```

package com.example.parta.program3;

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);

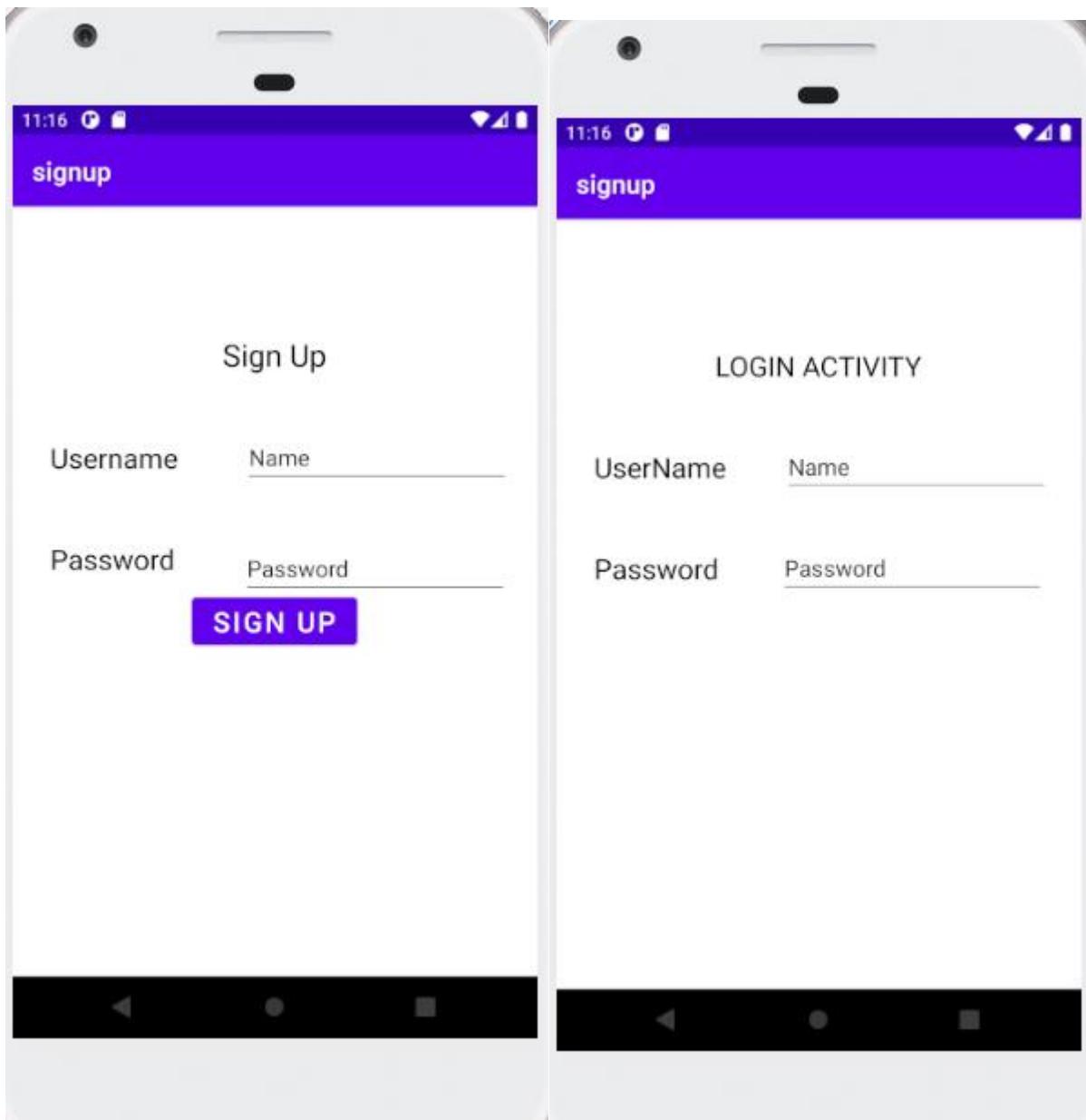
        Bundle bundle=getIntent().getBundleExtra("data");
        user=bundle.getString("user");
        pass=bundle.getString("pass");

    }

    public void onClick(View v)
    {
        String user1=txtLoginUsername.getText().toString();
        String pass1=txtLoginPassword.getText().toString();

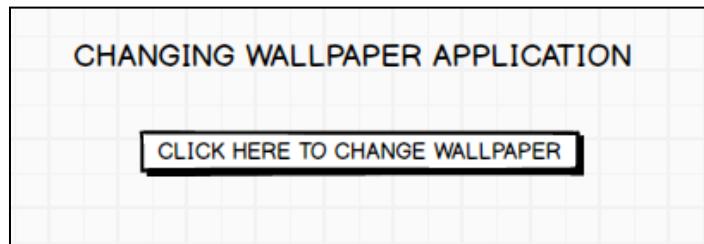
        if(user.equals(user1)&&pass.equals(pass1))
        {
            Toast.makeText(this,"Login Successful"
                ,Toast.LENGTH_LONG).show();
        }
        else
        {
            count++;
            if(count==2)
            {
                btnLogin.setEnabled(false);
                Toast.makeText(this,
                    "Failed Login Attempts "+count
                    ,Toast.LENGTH_LONG).show();
            }
            else
            {
                Toast.makeText(this,"Login Failed "+count
                    ,Toast.LENGTH_LONG).show();
            }
        }
    }
}
```

## Sample Output

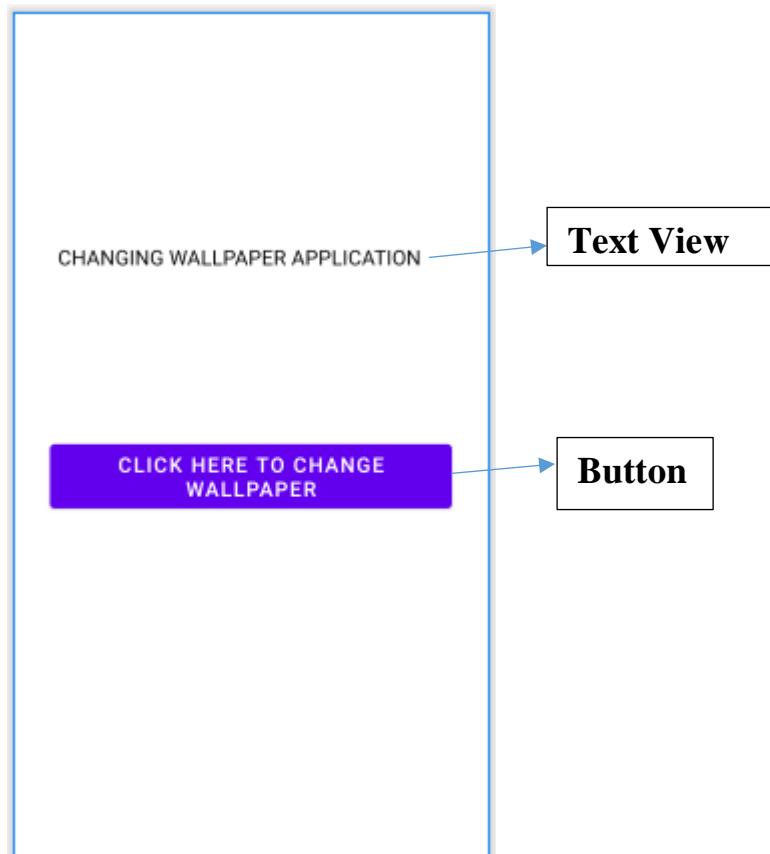


## Program 4

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 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**

```
<?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:gravity="center"
    tools:context=".MainActivity">

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Click here to Change Wallpaper"
        android:id="@+id	btn_start_change_wallpaper"/>

</LinearLayout>
```

## MainActivity.java

```
package com.example.program4;

import androidx.appcompat.app.AppCompatActivity;

import android.app.WallpaperManager;
import android.graphics.BitmapFactory;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

import java.util.Timer;
import java.util.TimerTask;

public class MainActivity extends AppCompatActivity implements View.OnClickListener {

    Button btnChangeWallpaper;

    boolean running;

    int[] imagesArray=new int[]{R.drawable.image1,
        R.drawable.image2,
        R.drawable.image3,R.drawable.image4};

    int i=0;

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

        btnChangeWallpaper=(Button)
            findViewById(R.id.btn_start_change_wallpaper);
        btnChangeWallpaper.setOnClickListener(this);

    }

    public void onClick(View v)
    {
        if(!running)
        {
            new Timer().schedule
                (new MyTimer(),0,30000);
            running=true;
        }
    }

    class MyTimer extends TimerTask
```

```

{
    public void run()
    {

        try {
            WallpaperManager wallpaperManager =
                WallpaperManager.getInstance(getApplicationContext());
            if(i==4)
            {
                i=1;
            }
            if(i==2)
            {
                i=3;
            }
            if(i==3)
            {
                i=2;
            }
            if(i==1)
            {
                i=3;
            }

            wallpaperManager.setBitmap
                (BitmapFactory.decodeResource(getResources()
                    ,imagesArray[i]));
            i++;
        }
        catch(Exception e)
        {

        }
    }
}
}

```

**AndriodManifest.xml**

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.program4">

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

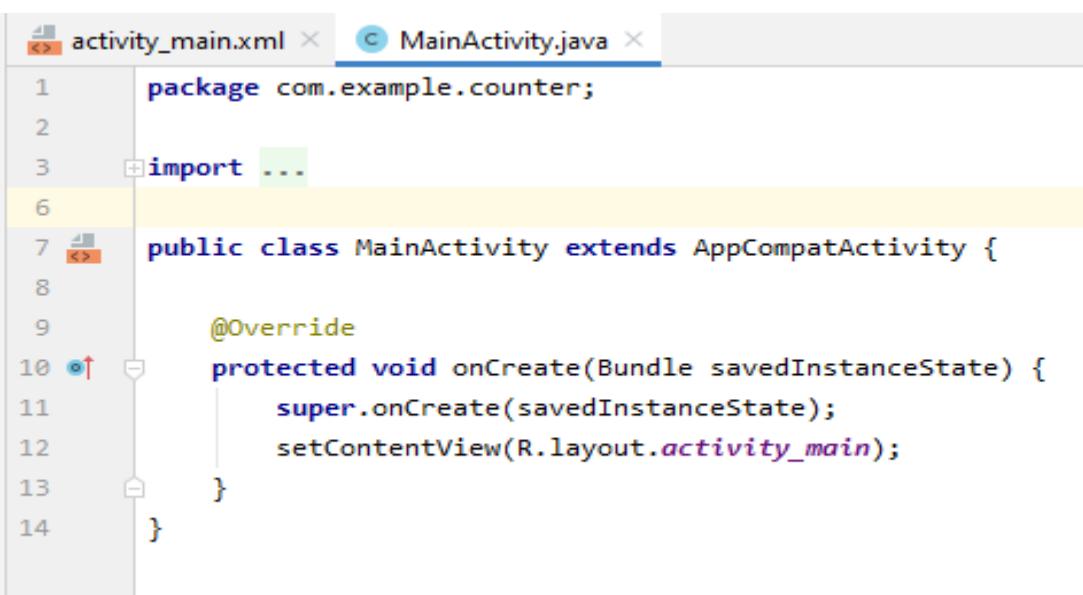
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"

```

```
    android:supportsRtl="true"
    android:theme="@style/AppTheme">
<activity android:name=".MainActivity">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

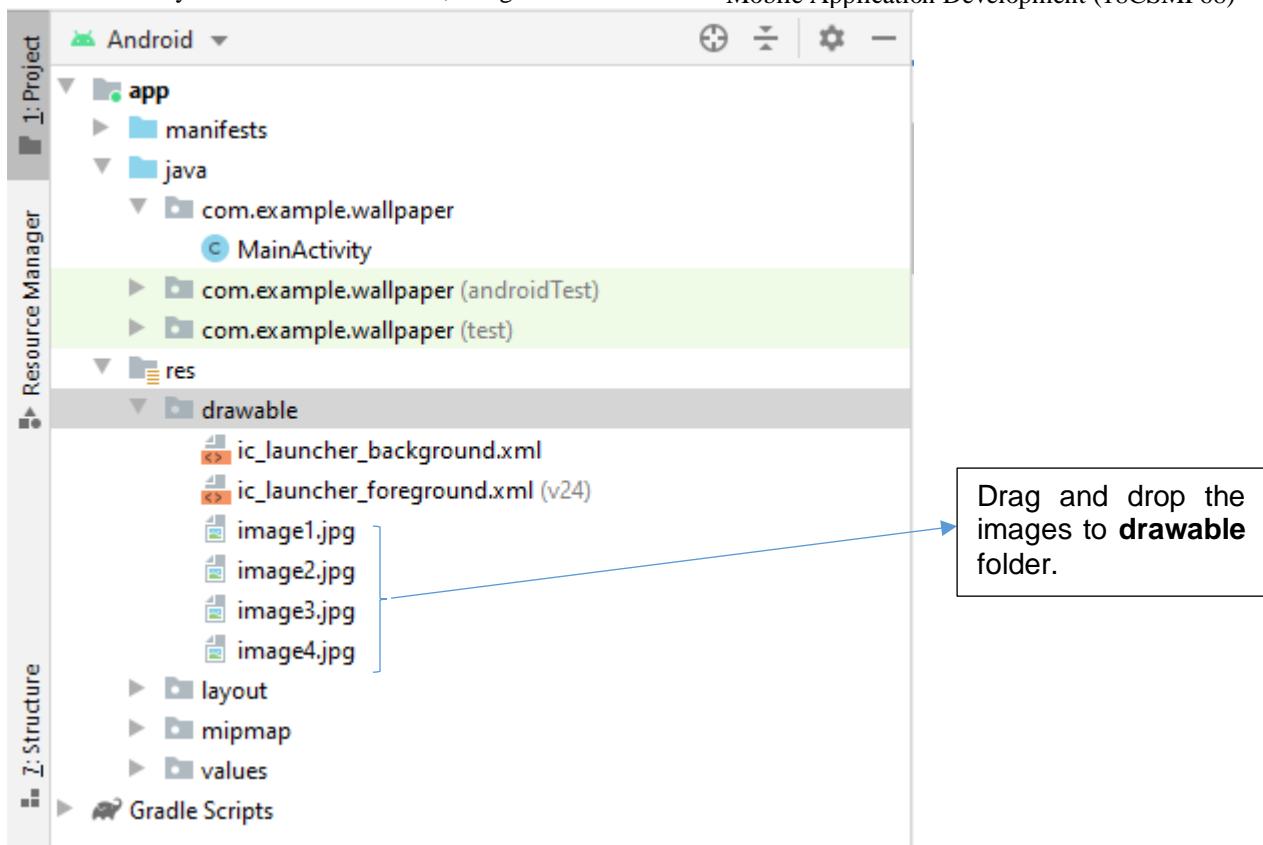
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
</application>

</manifest>
```



The screenshot shows an IDE interface with two tabs: 'activity\_main.xml' and 'MainActivity.java'. The 'MainActivity.java' tab is active, displaying the following Java code:

```
1 package com.example.counter;
2
3 import ...
4
5
6 public class MainActivity extends AppCompatActivity {
7
8     @Override
9     protected void onCreate(Bundle savedInstanceState) {
10         super.onCreate(savedInstanceState);
11         setContentView(R.layout.activity_main);
12     }
13 }
14 }
```



Drag and drop the images to **drawable** folder.

```
activity_main.xml MainActivity.java AndroidManifest.xml
1 package com.example.wallpaper;
2
3 import ...
4
5
6 public class MainActivity extends AppCompatActivity implements View.OnClickListener {
7
8     Button btnChangeWallpaper;
9
10    boolean running;
11
12    int[] imagesArray=new int[]{R.drawable.image1,
13                                R.drawable.image2,
14                                R.drawable.image3,R.drawable.image4};
15
16    int i=0;
17
18    @Override
19    protected void onCreate(Bundle savedInstanceState) {
20        super.onCreate(savedInstanceState);
21        setContentView(R.layout.activity_main);
22        btnChangeWallpaper=(Button)
23                    findViewById(R.id.btn_start_change_wallpaper);
24        btnChangeWallpaper.setOnClickListener(this);
25    }
26
27    @Override
28    public void onClick(View v) {
29        if(!running)
30        {
31            new Timer().schedule
32                (new MyTimer(), delay: 0, period: 30000);
33            running=true;
34        }
35    }
36}
```

Schedules the specified task for repeated fixed-delay execution, beginning after the specified delay.

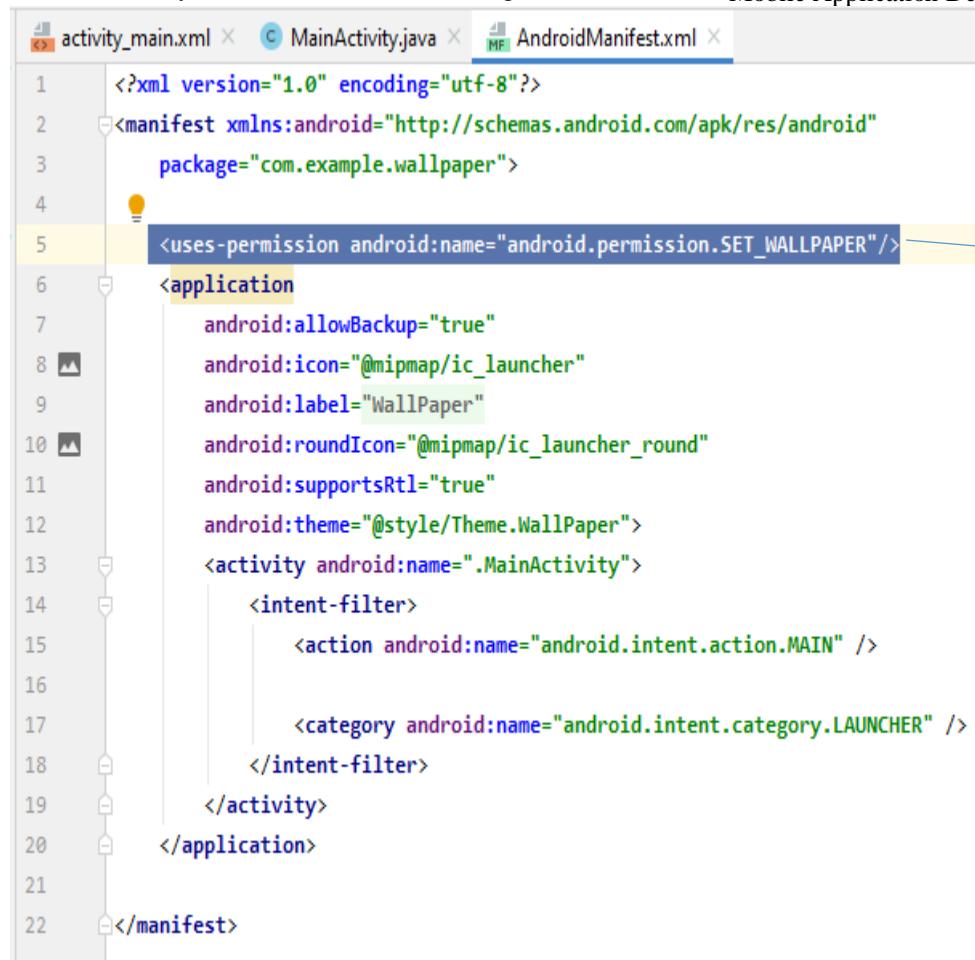
```
class MyTimer extends TimerTask
{
    public void run()
    {
        try {
            WallpaperManager wallpaperManager =
                WallpaperManager.getInstance(getApplicationContext());
            if(i==4)
            {
                i=1;
            }
            if(i==2)
            {
                i=3;
            }
            if(i==3)
            {
                i=2;
            }
            if(i==1)
            {
                i=3;
            }
            wallpaperManager.setBitmap
                (BitmapFactory.decodeResource(getResources()
                    ,imagesArray[i]));
            i++;
        }
    }
}
```

A task that can be scheduled for one-time or repeated execution by a Timer.

Provides access to the system wallpaper. With WallpaperManager, you can get the current wallpaper, get the desired dimensions for the wallpaper, set the wallpaper.

Change the current system wallpaper to a bitmap. The given bitmap is converted to a PNG and stored as the wallpaper.

Creates Bitmap objects from various sources, including files, streams, and byte-arrays.



```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.wallpaper">
    <uses-permission android:name="android.permission.SET_WALLPAPER"/>
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="WallPaper"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.WallPaper">
        <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>
```

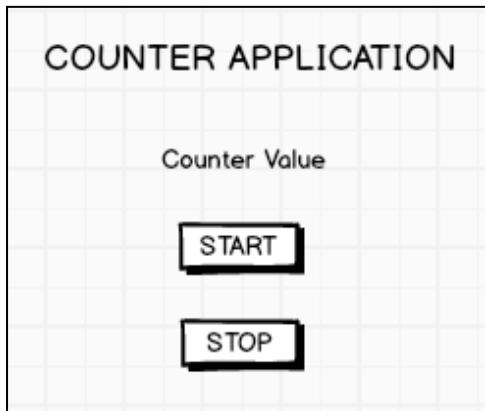
Here user is given permission for setting wallpaper.

## Sample Output

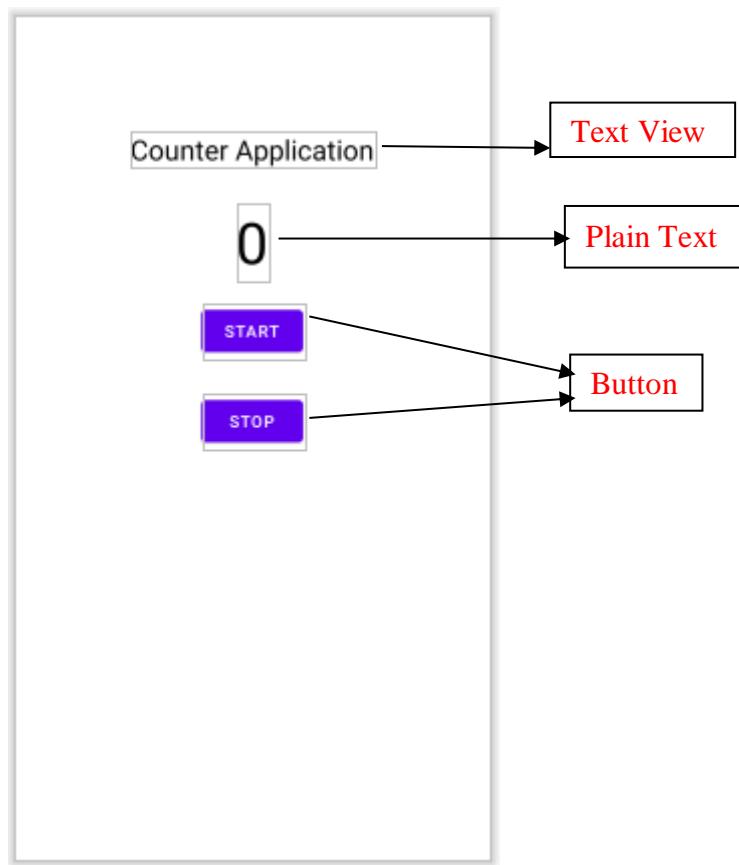


## Program 5

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="18dp"
        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="0"
    android:textColor="@color/colorAccent"
    android:textSize="50dp"
    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>
```

## MainActivity.java

```
package com.example.program5;

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.sendEmptyMessage(counter);  
            try {  
                Thread.sleep(1000);  
            }  
            catch(Exception e)  
            {  
  
            }  
        }  
    }  
}
```

```
activity_main.xml x MainActivity.java x
1 package com.example.counter;
2
3 import ...
4
5
6 public class MainActivity extends AppCompatActivity {
7     @Override
8     protected void onCreate(Bundle savedInstanceState) {
9         super.onCreate(savedInstanceState);
10        setContentView(R.layout.activity_main);
11    }
12
13 }
14 }
```

```
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 = handleMessage(m) > {
    lblCounter.setText(String.valueOf(m.what));
};

```

```

class MyCounter extends Thread {
    public void run() {
        while (running) {
            counter++;
            handler.sendEmptyMessage(counter);
            try {
                Thread.sleep( millis: 1000 );
            } catch (Exception e) {
            }
        }
    }
}

```

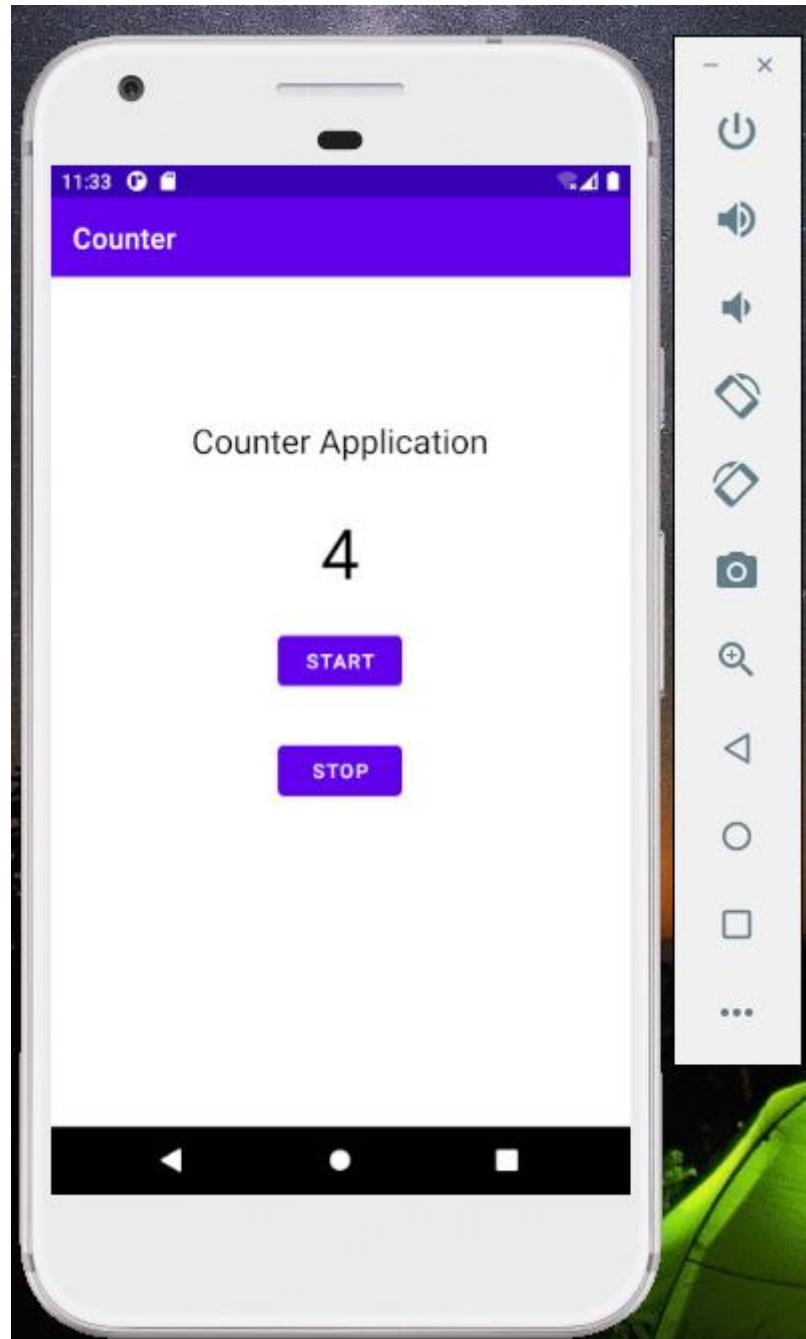
A Handler allows you to send and process Message and Runnable objects associated with a thread's MessageQueue. Each Handler instance is associated with a single thread and that thread's message queue.

A thread is a thread of execution in a program. The Java Virtual Machine allows an application to have multiple threads of execution running concurrently.

Returns true if the message was successfully placed in to the message queue. Returns false on failure, usually because the looper processing the message queue is exiting.

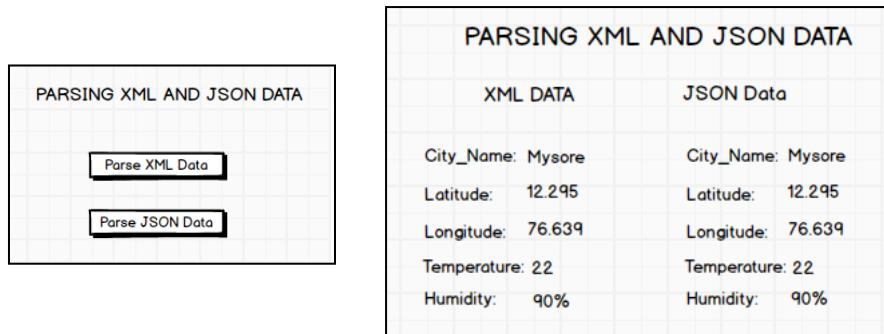
Causes the currently executing thread to sleep for the specified number of milliseconds.

## 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  
Right Click on app => New => Folder => Assets  
New Configuration Window open, Choose the the source as main and click Finish.
6. Create **input.xml** file inside assets folder and paste the below Xml Data
 

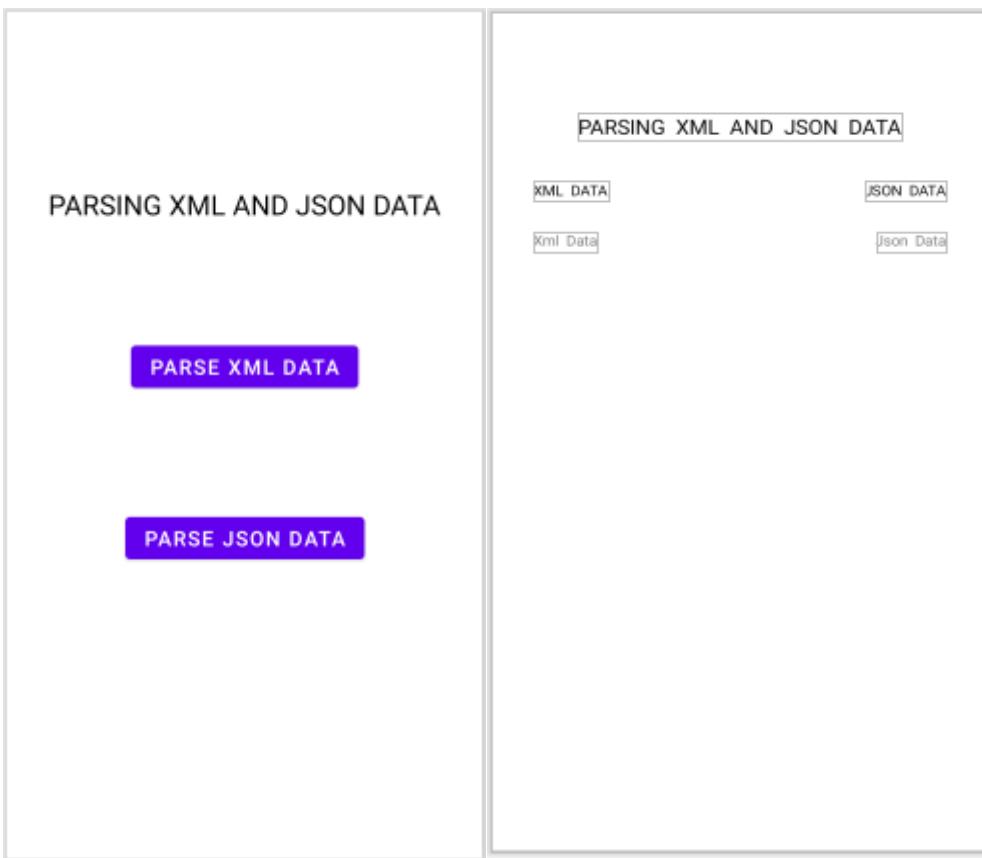
```

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

```

{
  "City": { "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="150dp"
        android:text="Parse XML"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" />
```

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

```
<Button
    android:id="@+id	btn_parsejson"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="80dp"
    android:text="Parse Json"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id	btn_parsexml" />
</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=".ViewActivity2">
```

```
<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="100dp"
    android:text="PARSING XML AND JSON DATA"
    android:textColor="@color/black"
    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="40dp"
    android:text="XML DATA"
    android:textColor="@color/black"
    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="40dp"
    android:layout_marginEnd="40dp"
    android:text="JSON DATA"
    android:textColor="@color/black"
    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.parsingxmltojsondata;

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,ViewActivity2.class);
        it.putExtra("mode",1);
        startActivity(it);
    }
    else if(v.equals(btnParseXml))
    {
        Intent it=new Intent(this,ViewActivity2.class);
        it.putExtra("mode",2);
        startActivity(it);
    }
}
```

**ViewActivity.java**  
**package com.example.parsingxmltojsondata;**

**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 ViewActivity2 extends AppCompatActivity {

    TextView lblXmlData, lblJsonData;

    int mode=0;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_view2);
        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 nodeList =
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");

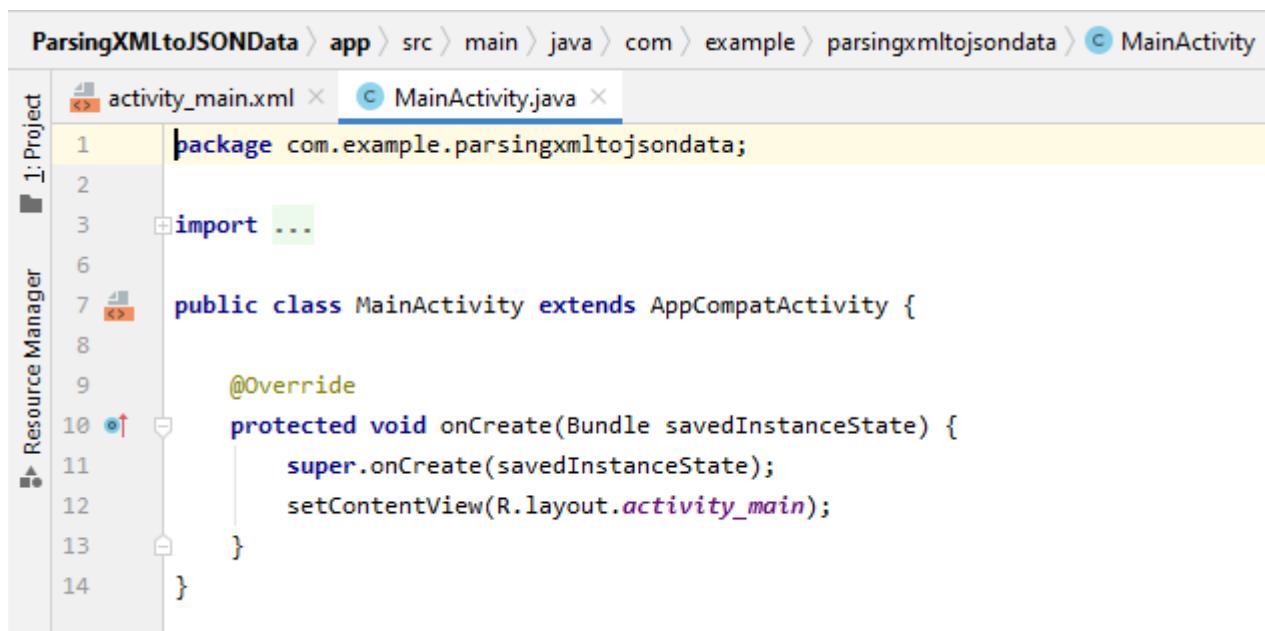
```

```
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();
}
}
```

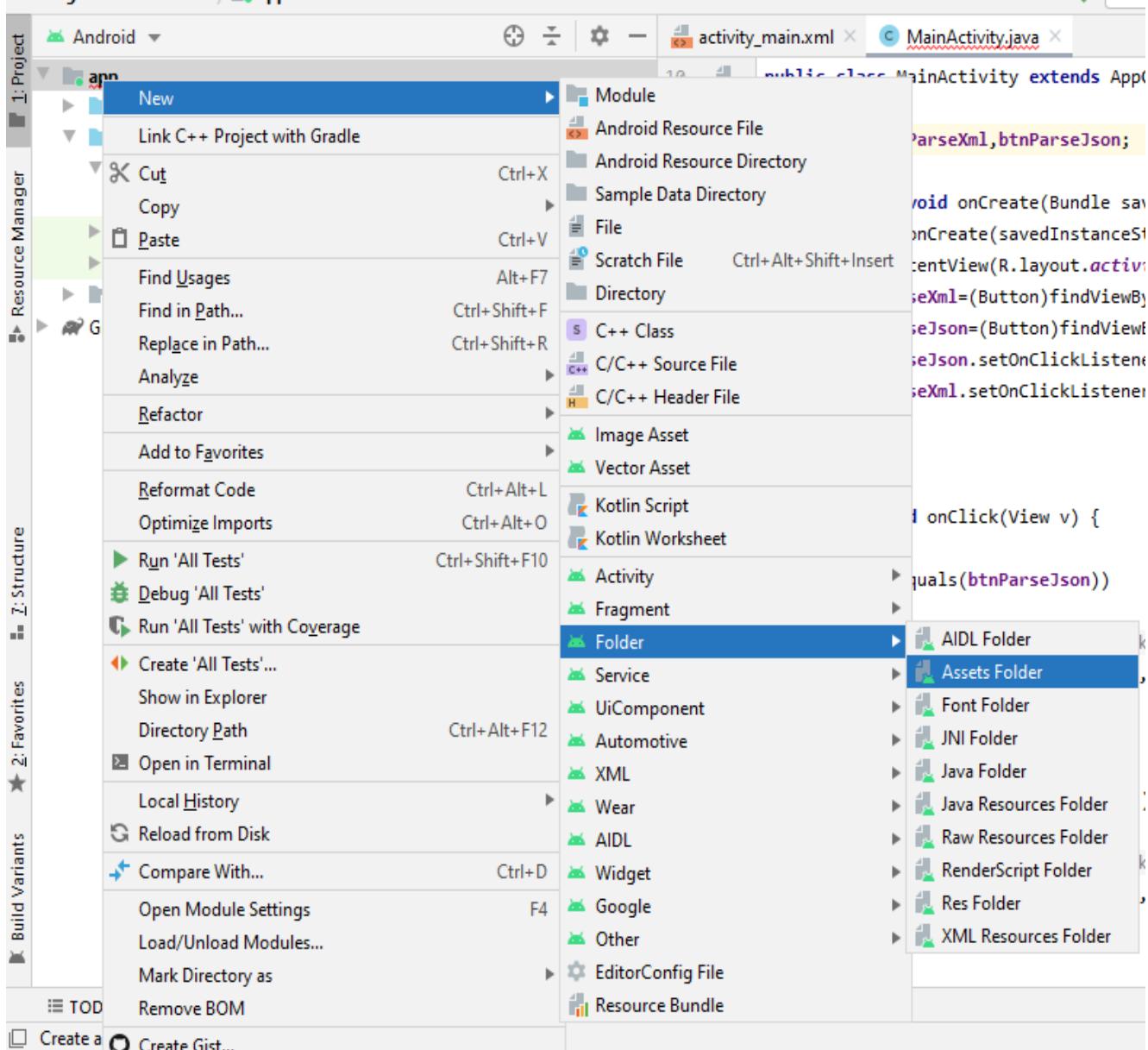


The screenshot shows the Android Studio interface with the project navigation bar at the top. Below it is the code editor for the `MainActivity.java` file. The code defines a `MainActivity` class that extends `AppCompatActivity`. It overrides the `onCreate` method to set the content view to `R.layout.activity_main`.

```
ParsingXMLtoJSONData > app > src > main > java > com > example > parsingxmltojsondata > MainActivity.java

1 package com.example.parsingxmltojsondata;
2
3 import ...
4
5
6
7 public class MainActivity extends AppCompatActivity {
8
9     @Override
10    protected void onCreate(Bundle savedInstanceState) {
11        super.onCreate(savedInstanceState);
12        setContentView(R.layout.activity_main);
13    }
14}
```

```
activity_main.xml MainActivity.java
10 public class MainActivity extends AppCompatActivity implements View.OnClickListener {
11
12     Button btnParseXml,btnParseJson;
13
14     @Override
15     protected void onCreate(Bundle savedInstanceState) {
16         super.onCreate(savedInstanceState);
17         setContentView(R.layout.activity_main);
18         btnParseXml=(Button)findViewById(R.id.btn_parsexml);
19         btnParseJson=(Button)findViewById(R.id.btn_parsejson);
20         btnParseJson.setOnClickListener(this);
21         btnParseXml.setOnClickListener(this);
22     }
23
24     @Override
25     public void onClick(View v) {
26
27         if(v.equals(btnParseJson))
28         {
29             Intent it=new Intent( packageContext: this,ViewActivity.class);
30             it.putExtra( name: "mode", value: 1);
31             startActivity(it);
32         }
33         else if(v.equals(btnParseXml))
34         {
35             Intent it=new Intent( packageContext: this,ViewActivity.class);
36             it.putExtra( name: "mode", value: 2);
37             startActivity(it);
38         }
39     }
40 }
```



The screenshot shows the 'Configure Component' dialog in Android Studio. At the top, there is a logo of a phone with 'AI' inside and the text 'Configure Component'. On the right side, there are icons of two phones. Below the title, there is a checkbox labeled 'Change Folder Location' and a dropdown menu labeled 'Target Source Set' set to 'main'. A large dashed square placeholder is centered on the screen. Below it, the text 'Assets Folder' is displayed, followed by the description 'Creates a source root for assets which will be included in the APK'. At the bottom right, there are buttons for 'Previous', 'Next', 'Cancel', and 'Finish'.

Assets Folder

Creates a source root for assets which will be included in the APK

Finish

Previous Next Cancel

Project

1: Project

Android

activity\_main.xml MainActivity.java input.xml

app manifests java com.example.parsingxmltojsondata MainActivity com.example.parsingxmltojsondata (androidTest) com.example.parsingxmltojsondata (test) assets input.json input.xml res Gradle Scripts

```
<?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>
```

The screenshot shows the Android Studio interface with the code editor open to `MainActivity.java`. The code implements a `ViewActivity2` class that extends `AppCompatActivity`. It initializes two `TextView` components: `lblXmlData` and `lblJsonData`. The `onCreate` method sets the content view to `activity_view2`, finds the views by ID, and checks the intent extra `mode`. If `mode` is 1, it calls `parseJson()`; otherwise, it calls `parseXmlDocument()`.

```
1 package com.example.parsingxmltojsondata;
2
3 import ...
4
5 public class ViewActivity2 extends AppCompatActivity {
6
7     TextView lblXmlData, lblJsonData;
8
9     int mode=0;
10    @Override
11    protected void onCreate(Bundle savedInstanceState) {
12        super.onCreate(savedInstanceState);
13        setContentView(R.layout.activity_view2);
14        lblXmlData=(TextView)findViewById(R.id.lbl_xml_data);
15        lblJsonData=(TextView)findViewById(R.id.lbl_json_data);
16        mode=getIntent().getIntExtra("mode", 0);
17        if(mode==1)
18        {
19            parseJson();
20        }
21        else
22        {
23            parseXmlDocument();
24        }
25    }
26}
```

The screenshot shows two instances of the Android Studio IDE. The top instance displays the `ViewActivity2.java` file, which contains Java code for parsing an XML file named `input.xml`. The code uses `DocumentBuilder` to parse the XML and extract values for elements like `city_name`, `Latitude`, `Longitude`, `Temperature`, and `Humidity`, which are then appended to a string. The bottom instance shows the same project structure, with the `ViewActivity2.java` file open, but the code has been modified to parse a JSON file instead. It uses `JSONObject` to get the JSON object for the `employee` tag and then extracts the same five values. Both files are part of a project named `ParsingXMLtoJSONData`.

```

activity_main.xml MainActivity.java input.json input.xml activity_view2.xml ViewActivity2.java

1: Project
2: Resource Manager
3: Favorites
4: I: Structure
5: 2: Favorites
6: its
7: 1: Project
8: Resource Manager
9: Favorites
10: I: Structure
11: 2: Favorites
12: wild Variants

activity_main.xml MainActivity.java input.json input.xml activity_view2.xml ViewActivity2.java

1: Project
2: Resource Manager
3: Favorites
4: I: Structure
5: 2: Favorites
6: wild Variants

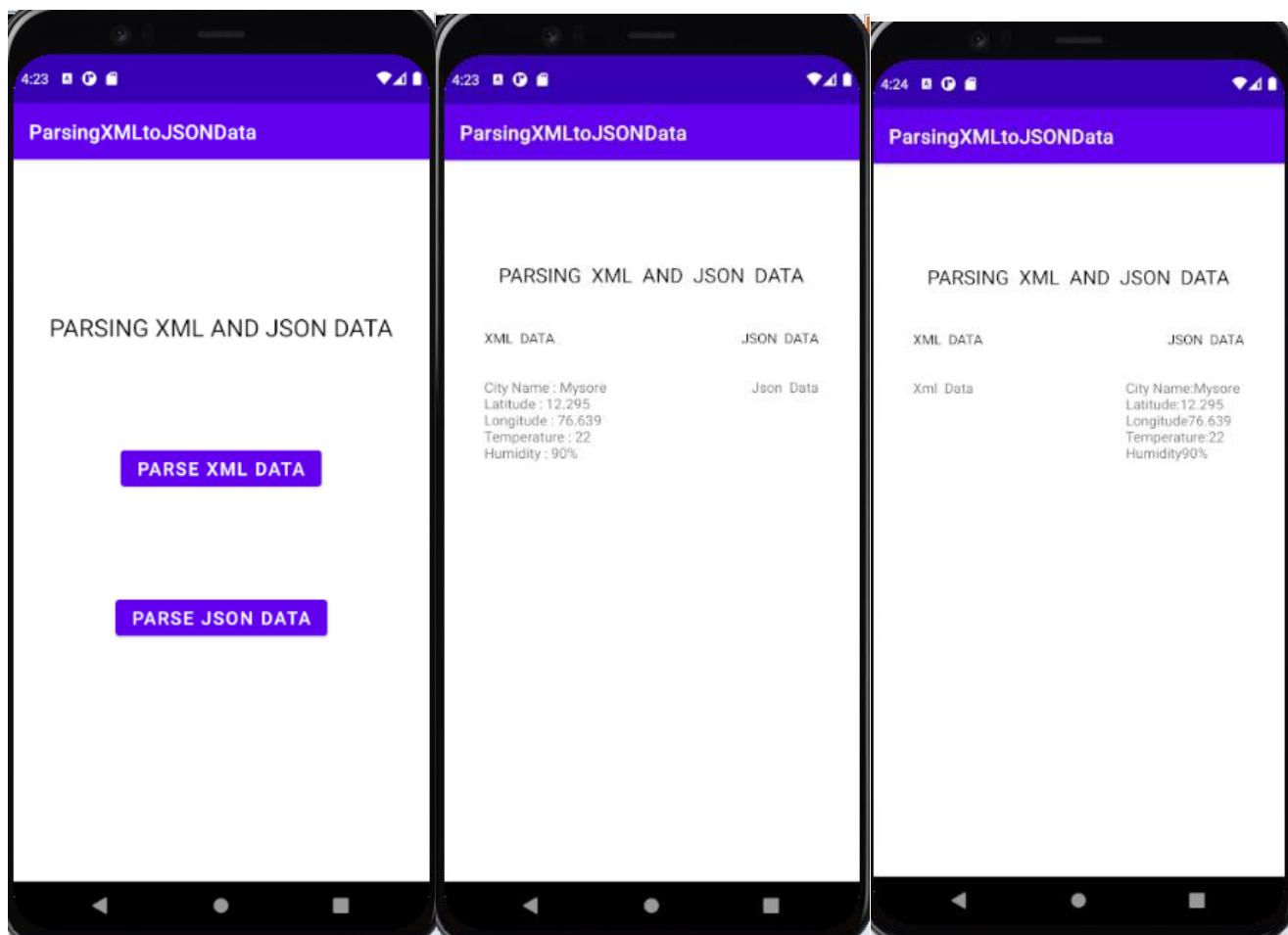
```

```

39     }
40     public String parseXmlDocument()
41     {
42         try {
43             InputStream is = getAssets().open( fileName: "input.xml");
44             DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
45             DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();
46             Document doc = dBuilder.parse(is);
47             Element element=doc.getDocumentElement();
48             element.normalize();
49             NodeList nList = doc.getElementsByTagName("employee");
50             for (int i=0; i<nList.getLength(); i++) {
51                 Node node = nList.item(i);
52                 if (node.getNodeType() == Node.ELEMENT_NODE) {
53                     Element element2 = (Element) node;
54                     lblXmlData.setText("City Name : " + getValue( tag: "city_name", element2)+"\n");
55                     lblXmlData.append("Latitude : " + getValue( tag: "Latitude", element2)+"\n");
56                     lblXmlData.append("Longitude : " + getValue( tag: "Longitude", element2)+"\n");
57                     lblXmlData.append("Temperature : " + getValue( tag: "Temperature", element2)+"\n");
58                     lblXmlData.append("Humidity : " + getValue( tag: "Humidity", element2)+"\n");
59                 }
60             }
61         } catch (Exception e) {e.printStackTrace();}
62         return null;
63     }
64
65     private static String getValue(String tag, Element element) {
66         NodeList nodeList = element.getElementsByTagName(tag).item( index: 0).getChildNodes();
67         Node node = nodeList.item( index: 0);
68         return node.getNodeValue();
69     }
70
71     public void parseJson()
72     {
73         try {
74             InputStream inputStream=getAssets().open( fileName: "input.json");
75             byte[] data=new byte[inputStream.available()];
76             inputStream.read(data);
77
78             String readData=new String(data);
79             JSONObject jsonObject=new JSONObject(readData);
80             JSONObject jsonObject1=jsonObject.getJSONObject("employee");
81             lblJsonData.setText("City Name:"+jsonObject1.getString( name: "city_name")+"\n");
82             lblJsonData.append("Latitude:"+jsonObject1.getString( name: "Latitude")+"\n");
83             lblJsonData.append("Longitude"+jsonObject1.getString( name: "Longitude")+"\n");
84             lblJsonData.append("Temperature:"+jsonObject1.getInt( name: "Temperature")+"\n");
85             lblJsonData.append("Humidity"+jsonObject1.getString( name: "Humidity")+"\n");
86
87         } catch (Exception e) {
88             e.printStackTrace();
89         }
90     }

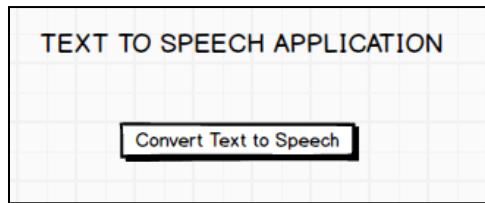
```

## Sample Output



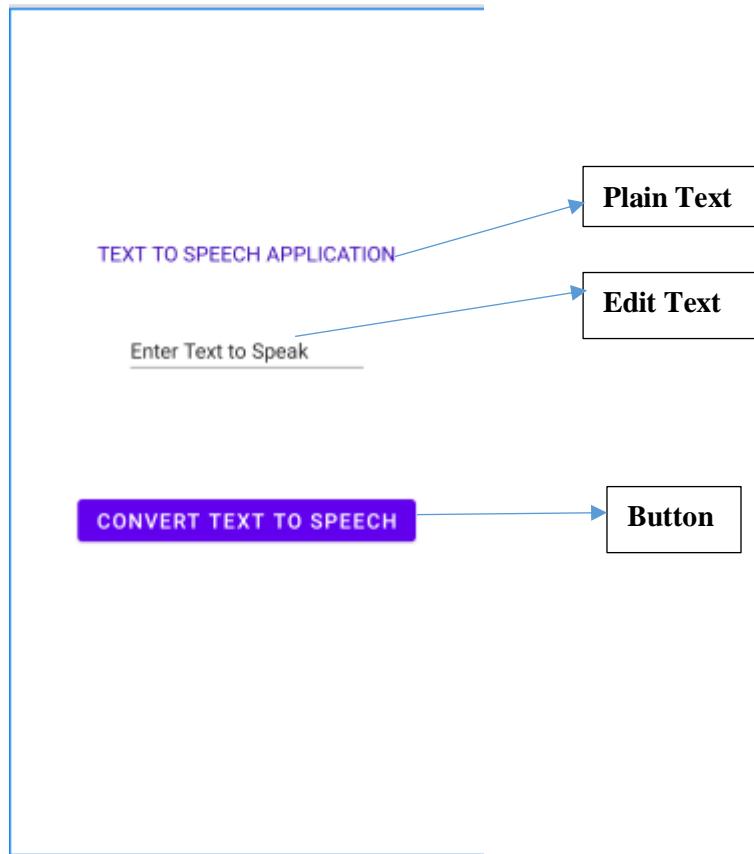
## Program 7

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 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: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="200dp"
        android:text="TEXT TO SPEECH APPLICATION"
        android:textColor="@color/purple_700"
        android:textSize="18sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id	btn_speak"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="100dp"
        android:text="CONVERT TEXT TO SPEECH"
        android:textSize="18sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/editText" />

    <EditText
        android:id="@+id/editText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="50dp"
        android:ems="10"
        android:inputType="textPersonName"
        app:layout_constraintEnd_toEndOf="@+id/textView"
        app:layout_constraintStart_toStartOf="@+id/textView" />
```

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

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

**MainActivity.java**

```
package com.example.texttospeech;

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;

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();
                }
            }
        });
}

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

}

The screenshot shows the Android Studio interface with two tabs open: `activity_main.xml` and `MainActivity.java`. The `MainActivity.java` tab is active, displaying Java code for a Text-to-Speech application.

```
activity_main.xml x MainActivity.java x

1 package com.example.texttospeech;
2
3 import ...
4
5 public class MainActivity extends AppCompatActivity implements View.OnClickListener {
6
7     @Override
8     protected void onCreate(Bundle savedInstanceState) {
9         super.onCreate(savedInstanceState);
10        setContentView(R.layout.activity_main);
11    }
12
13    @Override
14    public void onClick(View v) {
15    }
16
17 }

18 public class MainActivity extends AppCompatActivity implements View.OnClickListener {
19
20     EditText txtSpeak;
21     Button btnSpeak;
22     TextToSpeech textToSpeech;
23
24     @Override
25     protected void onCreate(Bundle savedInstanceState) {
26         super.onCreate(savedInstanceState);
27         setContentView(R.layout.activity_main);
28         txtSpeak=(EditText)findViewById(R.id.editText);
29         btnSpeak=(Button)findViewById(R.id.btn_speak);
30         btnSpeak.setOnClickListener(this);
31         textToSpeech=new TextToSpeech(getApplicationContext(),
32             new TextToSpeech.OnInitListener() {
33                 @Override
34                 public void onInit(int status) {
35                     if(status!=TextToSpeech.ERROR)
36                     {
37                         Toast.makeText(getApplicationContext(),
38                             text: "Success",
39                             Toast.LENGTH_LONG).show();
40                     }
41                 }
42             });
43
44     }
45
46     @Override
47     public void onClick(View v) {
48
49         String text=txtSpeak.getText().toString();
50         textToSpeech.speak(text,
51             TextToSpeech.QUEUE_FLUSH, params: null);
52     }
53
54 }
```

A TextToSpeech instance is created in the `onCreate()` method. The `textToSpeech` object is initialized with the application context and an `OnInitListener`. The `onInit()` method checks if initialization was successful. If so, it displays a success toast message.

`getText():` Returns the text currently displayed.

`speak():` Speaks the text. It includes a queuing strategy and a method to add requests. This method is asynchronous; it adds the request to the queue and then returns.

`QUEUE_FLUSH :` Adds entries in the play list before played and text to be added and replaced by the new text.

A `TextToSpeech` instance can only be used to synthesize text once it has completed its initialization.

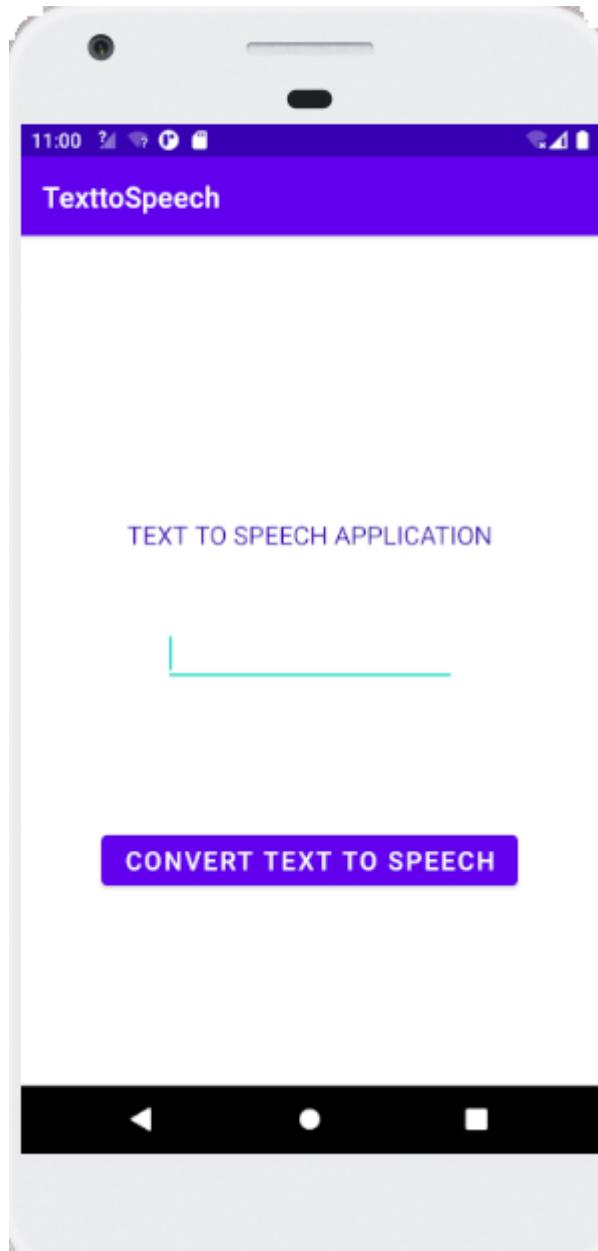
**OnInit()**: Interface definition of a callback to be invoked indicating the completion of the TextToSpeech engine initialization.

`getText()`: Return the text that `TextView` is displaying.

`speak()`: Speaks the string using the specified queuing strategy and speech parameters. This method is asynchronous, i.e. the method just adds the request to the queue of TTS requests and then returns.

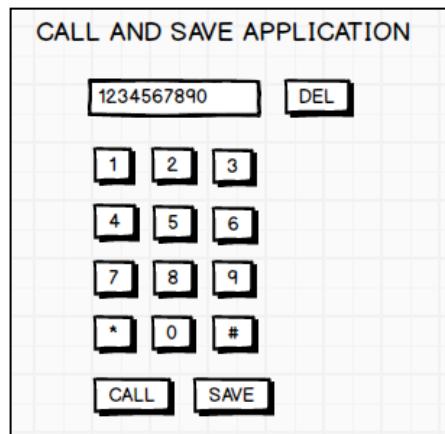
**QUEUE\_FLUSH** : Queue mode where all entries in the playback queue (media to be played and text to be synthesized) are dropped and replaced by the new entry.

## Sample Output



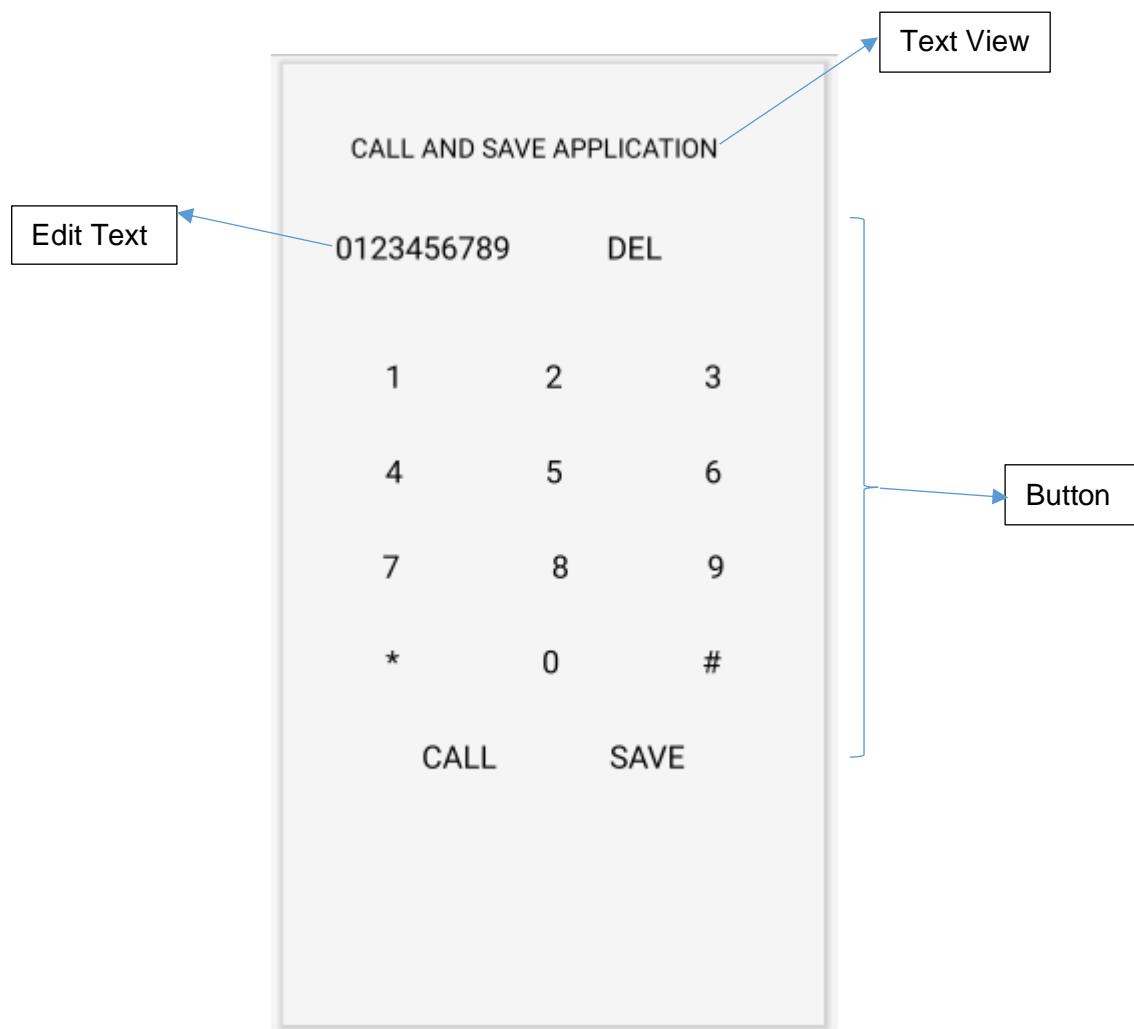
## Program 8

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 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/textView2"
    android:layout_width="309dp"
    android:layout_height="33dp"
    android:layout_marginTop="50dp"
    android:text="CALL AND SAVE APPLICATION"
    android:textColor="@color/black"
    android:textSize="20sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

**<EditText**

```
    android:id="@+id/txt_phonenumber"
    android:layout_width="137dp"
    android:layout_height="58dp"
    android:layout_marginTop="40dp"
    android:layout_marginEnd="30dp"
    android:ems="10"
    android:inputType="textPersonName"
    android:text="0123456789"
    android:textSize="24sp"
    app:layout_constraintEnd_toStartOf="@+id	btn_delete"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView2" />
```

**<Button**

```
    android:id="@+id	btn_delete"
    android:layout_width="105dp"
    android:layout_height="69dp"
    android:layout_marginTop="40dp"
    android:layout_marginEnd="60dp"
```

```
    android:text="DEL"  
    android:textSize="24sp"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintTop_toBottomOf="@+id/textView2" />
```

```
<Button  
    android:id="@+id	btn_four"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_marginStart="50dp"  
    android:layout_marginTop="40dp"  
    android:layout_marginEnd="40dp"  
    android:text="4"  
    android:textSize="24sp"  
    app:layout_constraintEnd_toStartOf="@+id	btn_five"  
    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_marginStart="40dp"  
    android:layout_marginTop="40dp"  
    android:layout_marginEnd="40dp"  
    android:text="5"  
    android:textSize="24sp"  
    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_marginStart="40dp"  
    android:layout_marginTop="40dp"  
    android:layout_marginEnd="50dp"  
    android:text="6"  
    android:textSize="24sp"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintStart_toEndOf="@+id	btn_five"  
    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="50dp"
    android:layout_marginTop="40dp"
    android:layout_marginEnd="50dp"
    android:text="7"
    android:textSize="24sp"
    app:layout_constraintEnd_toStartOf="@+id	btn_eight"
    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_marginStart="40dp"
    android:layout_marginTop="40dp"
    android:layout_marginEnd="40dp"
    android:text="8"
    android:textSize="24sp"
    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_marginStart="40dp"
    android:layout_marginTop="40dp"
    android:layout_marginEnd="50dp"
    android:text="9"
    android:textSize="24sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toEndOf="@+id	btn_eight"
    app:layout_constraintTop_toBottomOf="@+id	btn_six" />
```

**<Button**

```
    android:id="@+id	btn_star"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="50dp"
```

```
    android:layout_marginTop="40dp"
    android:layout_marginEnd="40dp"
    android:text="*"
    android:textSize="24sp"
    app:layout_constraintEnd_toStartOf="@+id(btn_zero)"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id(btn_seven)"/>
```

**<Button**

```
    android:id="@+id(btn_zero)"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="40dp"
    android:layout_marginTop="40dp"
    android:layout_marginEnd="40dp"
    android:text="0"
    android:textSize="24sp"
    app:layout_constraintEnd_toStartOf="@+id(btn_hash)"
    app:layout_constraintStart_toEndOf="@+id(btn_star"
    app:layout_constraintTop_toBottomOf="@+id(btn_eight)"/>
```

**<Button**

```
    android:id="@+id(btn_hash)"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="40dp"
    android:layout_marginTop="40dp"
    android:layout_marginEnd="50dp"
    android:text="#"
    android:textSize="24sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toEndOf="@+id(btn_zero"
    app:layout_constraintTop_toBottomOf="@+id(btn_nine)"/>
```

**<Button**

```
    android:id="@+id(btn_call"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="100dp"
    android:layout_marginTop="40dp"
    android:layout_marginEnd="40dp"
    android:text="CALL"
    android:textSize="24sp"
    app:layout_constraintEnd_toStartOf="@+id(btn_save" />
```

```
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id(btn_star)"/>
```

```
<Button
    android:id="@+id btnSave"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="40dp"
    android:layout_marginTop="40dp"
    android:layout_marginEnd="100dp"
    android:text="SAVE"
    android:textSize="24sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toEndOf="@+id btnCall"
    app:layout_constraintTop_toBottomOf="@+id btnHash" />

<Button
    android:id="@+id btnOne"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="50dp"
    android:layout_marginTop="40dp"
    android:layout_marginEnd="40dp"
    android:text="1"
    android:textSize="24sp"
    app:layout_constraintEnd_toStartOf="@+id btnTwo"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id txtPhoneNumber" />

<Button
    android:id="@+id btnTwo"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="40dp"
    android:layout_marginTop="40dp"
    android:layout_marginEnd="40dp"
    android:text="2"
    android:textSize="24sp"
    app:layout_constraintEnd_toStartOf="@+id btnThree"
    app:layout_constraintStart_toEndOf="@+id btnOne"
    app:layout_constraintTop_toBottomOf="@+id txtPhoneNumber" />
```

```
<Button
    android:id="@+id btnThree"
```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="40dp"
    android:layout_marginTop="40dp"
    android:layout_marginEnd="50dp"
    android:text="3"
    android:textSize="24sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toEndOf="@+id	btn_two"
    app:layout_constraintTop_toBottomOf="@+id	txt_phonenumber" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

### MainActivity.java

```
package com.example.callapp;

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;

    EditText txtPhonenumber;
    @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_star);
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("");
}
```

```
@Override  
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("*");  
    }  
}
```

```

    {
        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(btnCall))
    {
        String data=txtPhonenumber.getText().toString();
        Intent it=new Intent(Intent.ACTION_CALL);
        it.setData(Uri.parse("tel:"+data));
        startActivity(it);

    }
    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("");
        }
    }
}

```

```
        }  
    }  
}  
}
```

### AndriodManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
    package="com.example.callapp">  
  
    <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:supportsRtl="true"  
        android:theme="@style/Theme.CallApp">  
        <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>
```



The screenshot shows an IDE interface with two tabs: 'activity\_main.xml' and 'MainActivity.java'. The 'MainActivity.java' tab is active, displaying the following Java code:

```
1 package com.example.counter;  
2  
3 import ...  
4  
5 public class MainActivity extends AppCompatActivity {  
6  
7     @Override  
8     protected void onCreate(Bundle savedInstanceState) {  
9         super.onCreate(savedInstanceState);  
10        setContentView(R.layout.activity_main);  
11    }  
12  
13 }  
14 }
```

The screenshot shows the Android Studio interface with two tabs open: `activity_main.xml` and `MainActivity.java`. The `MainActivity.java` tab is active, displaying Java code for a mobile application.

```

1 package com.example.callapp;
2
3 import ...
4
5 public class MainActivity extends AppCompatActivity implements View.OnClickListener{
6
7     Button btnOne,btnTwo,btnThree,btnFour,btnFive;
8     Button btnSix,btnSeven,btnEight,btnNine,btnZero;
9     Button btnDel,btnStar,btnHash,btnCall,btnSave;
10
11     EditText txtPhonenumber;
12
13     @Override
14     protected void onCreate(Bundle savedInstanceState) {
15         super.onCreate(savedInstanceState);
16         setContentView(R.layout.activity_main);
17         btnOne=(Button)findViewById(R.id.btn_one);
18         btnOne.setOnClickListener(this);
19
20         btnDel=(Button)findViewById(R.id.btn_delete);
21         btnDel.setOnClickListener(this);
22
23         txtPhonenumber=(EditText)findViewById(R.id.txt_phonenumber);
24
25         txtPhonenumber.setText("");
26     }
27
28     @Override
29     public void onClick(View v) {
30
31         if(v.equals(btnOne))
32         {
33             txtPhonenumber.append("1");
34         }
35         else if(v.equals(btnTwo))
36         {
37             txtPhonenumber.append("2");
38         }
39     }
40
41 }

```

`setText("")`: We are setting the txt box to blank initially.  
`txtPhonenumber.append("1")`: When pressed 1 it appends to the existing numbers.

```

126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150

    else if(v.equals(btnSave))
    {

        Intent contactIntent = new Intent
            (ContactsContract.Insert.ACTION);
        contactIntent.setType
            (ContactsContract.RawContacts.CONTENT_TYPE);

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

        startActivity(contactIntent);

    }
    else if(v.equals(btnCall))
    {
        String data=txtPhonenumber.getText().toString();
        Intent it=new Intent(Intent.ACTION_CALL);
        it.setData(Uri.parse("tel:"+data));
        startActivity(it);
    }

    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("");
        }
    }
}

```

**ContactsContract:** Is an extensible database of contact-related information.

**Insert.ACTION:** Convenience class that contains string constants used to create contact Intents. The action code to use when adding a contact.

**setType:** This is used to create intents that only specify a type and not data, for example to indicate the type of data to return.

**RawContacts:** Constants for the raw contacts table, which contains one row of contact information for each person in each synced account.

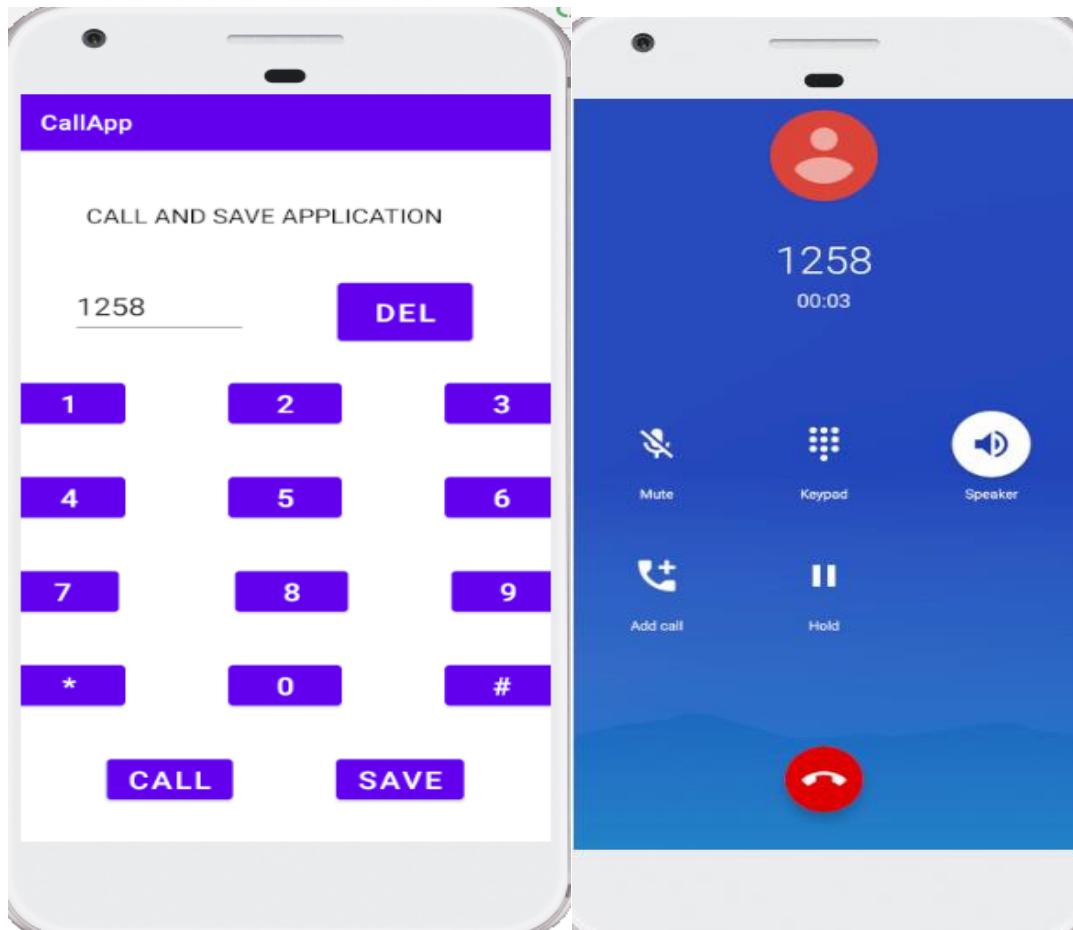
**ACTION\_CALL:** Activity Action: Perform a call to someone specified by the data. Input: If nothing, an empty dialer is started; else getData is URI of a phone number to be dialed or a tel: URI of an explicit phone number.

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

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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="CallApp"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.CallApp">
        <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