

**DAYANANDA SAGAR ACADEMY OF TECHNOLOGY  
AND MANAGEMENT**

**DEPARTMENT OF INFORMATION SCIENCE AND  
ENGINEERING  
Bengaluru, Karnataka-560082**



**"MOBILE APPLICATION DEVELOPMENT LAB MANUAL"  
(18CSMP68)**

**As per VTU Revised Syllabus**



**PREPARED BY:**

**Dr G Manjula  
Associate Professor,  
Department of ISE, DSATM, Bengaluru  
Email: manjula-ise@dsatm.edu.in**

**Table of content**

<b>Sl. No.</b>	<b>Programs</b>	<b>Page No.</b>
<b>Part - A</b>		
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
2	Develop an Android application using controls like Button, TextView, EditText for designing a calculator having basic functionality like Addition, Subtraction, Multiplication, and Division.	5
3	<p>Create a SIGN-Up activity with Username and Password. Validation of password should happen based on the following rules:</p> <ul style="list-style-type: none"> <li>- Password should contain uppercase and lowercase letters.</li> <li>- Password should contain letters and numbers.</li> <li>- Password should contain special characters.</li> <li>- Minimum length of the password (the default value is 8).</li> </ul> <p>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.</p>	9
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.	18
5	<p>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.</p>	22
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".	27
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.	31
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.	34
<b>Part - B</b>		
9	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.	40

10	Develop an application to demonstrate the use of Asynchronous tasks in android. The asynchroneoustask 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".	44
11	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.	48

## Introduction to Android

### What is Android?

Android is an operating system and programming platform developed by Google for smartphones and other mobile devices (such as tablets). It can run on many different devices from many different manufacturers. Android includes a software development kit for writing original code and assembling software modules to create apps for Android users. It also provides a marketplace to distribute apps. All together, Android represents an ecosystem for mobile apps.



### Why develop apps for Android?

Apps are developed for a variety of reasons: addressing business requirements, building new services, creating new businesses, and providing games and other types of content for users. Developers choose to develop for Android in order to reach the majority of mobile device users.

### Most popular platform for mobile apps

As the world's most popular mobile platform, Android powers hundreds of millions of mobile devices in more than 190 countries around the world. It has the largest installed base of any mobile platform and is still growing fast. Every day another million users power up their Android devices for the first time and start looking for apps, games, and other digital content.

### Easy to develop apps

Use the Android software development kit (SDK) to develop apps that take advantage of the Android operating system and UI. The SDK includes a comprehensive set of development tools including a debugger, software libraries of prewritten code, a device emulator, documentation, sample code, and tutorials. Use these tools to create apps that look great and take advantage of the hardware capabilities available on each device.

To develop apps using the SDK, use the Java programming language for developing the app and Extensible Markup Language (XML) files for describing data resources. By writing the code in Java and creating a single app binary, you will have an app that can run on both phone and tablet form factors. You can declare your UI in lightweight sets of XML resources, one set for parts of the UI that are common to all form factors, and other

sets for features specific to phones or tablets. At runtime, Android applies the correct resource sets based on its screen size, density, locale, and so on.

To help you develop your apps efficiently, Google offers a full Java Integrated Development Environment (IDE) called Android Studio, with advanced features for developing, debugging, and packaging Android apps. Using Android Studio, you can develop on any available Android device, or create virtual devices that emulate any hardware configuration.

Android provides a rich development architecture. You don't need to know much about the components of this architecture, but it is useful to know what is available in the system for your app to use. The following diagram shows the major components of the Android stack — the operating system and development architecture.

In the figure above:

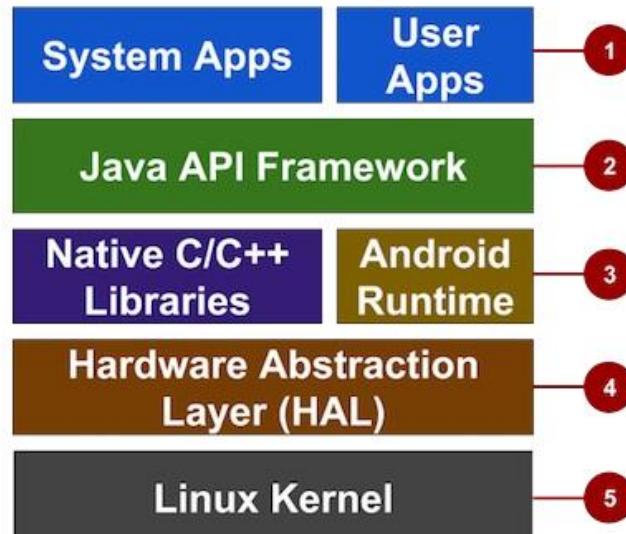
1. Apps: Your apps live at this level, along with core system apps for email, SMS messaging, calendars, Internet

browsing, or contacts.

2. Java API Framework: All features of Android are available to developers through application programming interfaces

(APIs) written in the Java language. You don't need to know the details of all of the APIs to learn how to develop

Android apps, but you can learn more about the following APIs, which are useful for creating apps:



View System used to build an app's UI, including lists, buttons, and menus. Resource Manager used to access to non-code resources such as localized strings, graphics, and layout files. Notification Manager used to display custom alerts in the status bar. Activity Manager that manages the lifecycle of apps. Content Providers that enable apps to access data from other apps.

2. All framework APIs that Android system apps use.

3. Libraries and Android Runtime: Each app runs in its own process and with its own instance of the Android Runtime, which enables multiple virtual machines on low-memory devices. Android also includes a set of core runtime libraries that provide most of the functionality of the Java programming language, including some Java 8 language features that the Java API framework uses. Many core Android system components and services are built from native code that require native libraries written in C and C++. These native libraries are available to apps through the Java API framework.

4. Hardware Abstraction Layer (HAL): This layer provides standard interfaces that expose device hardware capabilities to the higher-level Java API framework. The HAL consists of multiple library modules, each of which implements an interface for a specific type of hardware component, such as the camera or bluetooth module.

5. Linux Kernel: The foundation of the Android platform is the Linux kernel. The above layers rely on the Linux kernel for underlying functionalities such as threading and low-level memory management. Using a Linux kernel enables Android to take advantage of key security features and allows device manufacturers to develop hardware drivers for a well-known kernel.

**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.



XML Code

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

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="88dp"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="38dp"
        android:layout_marginBottom="616dp"
        android:text="Dayanand Sagar Academy of Technology and Management -DSATM"
        android:textSize="18sp"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="98dp"
        android:layout_height="50dp"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="15dp"
        android:layout_marginBottom="623dp"
        app:srcCompat="@drawable/dsat" />

    <View
        android:layout_width="match_parent"
        android:layout_height="1dp"
        android:background="#000000" />
```

```

        android:id="@+id/view"
        android:layout_width="wrap_content"
        android:layout_height="4dp"
        android:layout_alignParentBottom="true"
        android:layout_marginBottom="576dp"
        android:background="#4444" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="110dp"
    android:layout_marginBottom="494dp"
    android:text="Dr G Manjula"
    android:textAlignment="center"
    android:textSize="36sp"
    android:textStyle="bold" />

<TextView
    android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="145dp"
    android:layout_marginBottom="458dp"
    android:text="Associate Professor"
    android:textAlignment="center"
    android:textSize="18sp" />

<TextView
    android:id="@+id/textView3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="115dp"
    android:layout_marginBottom="412dp"
    android:text="Phone No.+91 7259457206"
    android:textAlignment="center"
    android:textSize="18sp" />

<TextView
    android:id="@+id/textView4"
    android:layout_width="386dp"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="9dp"
    android:layout_marginBottom="355dp"
    android:text="N0.12,Sri Venkateshwara Nilaya, Manchenahalli Road,Attibele,Bengaluru-562107"
    android:textAlignment="center"
    android:textSize="18sp" />

<TextView
    android:id="@+id/textView5"
    android:layout_width="251dp"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="88dp"

```

```

        android:layout_marginBottom="307dp"
        android:text="Email: milindmanjula@gmail.com
ise@dsatm.edu.in" />

<TextView
    android:id="@+id/textView6"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="155dp"
    android:layout_marginBottom="268dp"
    android:text="Website:www.dsatm.edu.in" />

</RelativeLayout>
Java Code
package com.example.visitingcard;

import androidx.appcompat.app.AppCompatActivity;

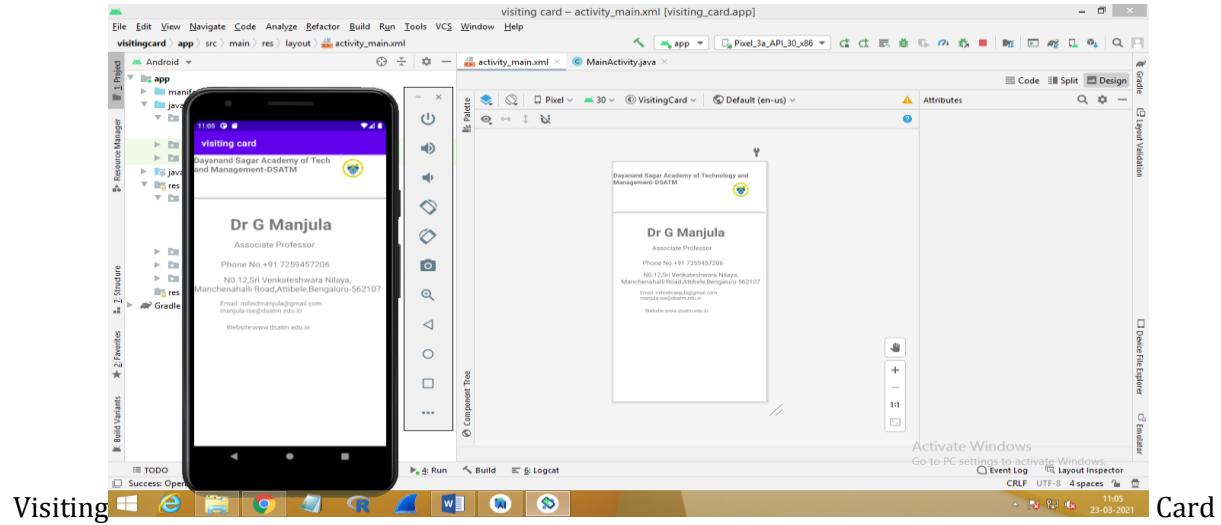
import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

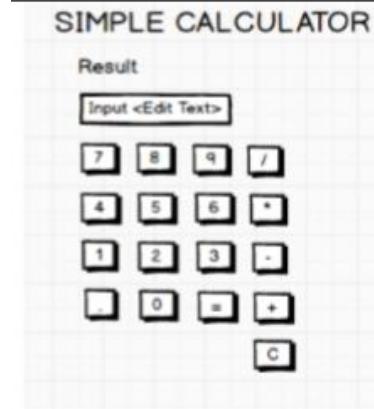
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}

```

Output Pgm1



**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.**



XML Code

```

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

    <TextView
        android:id="@+id/textView"
        android:layout_width="300dp"
        android:layout_height="64dp"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="41dp"
        android:layout_marginBottom="643dp"
        android:text="Simple Calculator"
        android:textAlignment="center"
        android:textSize="36sp"
        android:textStyle="bold" />

    <TextView
        android:id="@+id/tv1"
        android:layout_width="140dp"
        android:layout_height="66dp"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="140dp"
        android:layout_marginBottom="310dp"
        android:text="0"
        android:textAlignment="center"
        android:textSize="18sp" />

    <Button
        android:id="@+id/button"
        android:layout_width="142dp"
        android:layout_height="73dp"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:onClick="add"
        android:layout_marginEnd="231dp"
        android:layout_marginBottom="231dp" />

```

```

        android:layout_marginBottom="145dp"
        android:text="ADD" />

<Button
    android:id="@+id/button2"
    android:layout_width="130dp"
    android:layout_height="77dp"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:onClick="sub"
    android:layout_marginEnd="41dp"
    android:layout_marginBottom="141dp"
    android:text="SUB" />

<Button
    android:id="@+id/button3"
    android:layout_width="138dp"
    android:layout_height="77dp"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:onClick="mul"
    android:layout_marginEnd="241dp"
    android:layout_marginBottom="46dp"
    android:text="MULT" />

<Button
    android:id="@+id/button4"
    android:layout_width="124dp"
    android:layout_height="77dp"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:onClick="div"
    android:layout_marginEnd="49dp"
    android:layout_marginBottom="40dp"
    android:text="DIV" />

<EditText
    android:id="@+id/editText1"
    android:layout_width="279dp"
    android:layout_height="75dp"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="74dp"
    android:layout_marginBottom="523dp"
    android:ems="10"
    android:hint="Enter Number1"
    android:inputType="textPersonName"
    android:text="" />

<EditText
    android:id="@+id/editText2"
    android:layout_width="292dp"
    android:layout_height="86dp"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="62dp"
    android:layout_marginBottom="427dp"
    android:ems="10"
    android:hint="Enter Number 2"
    android:inputType="textPersonName"
    android:text="" />
</RelativeLayout>
```

**Java Code**

```

package com.example.simplecalculator;

import androidx.appcompat.app.AppCompatActivity;

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

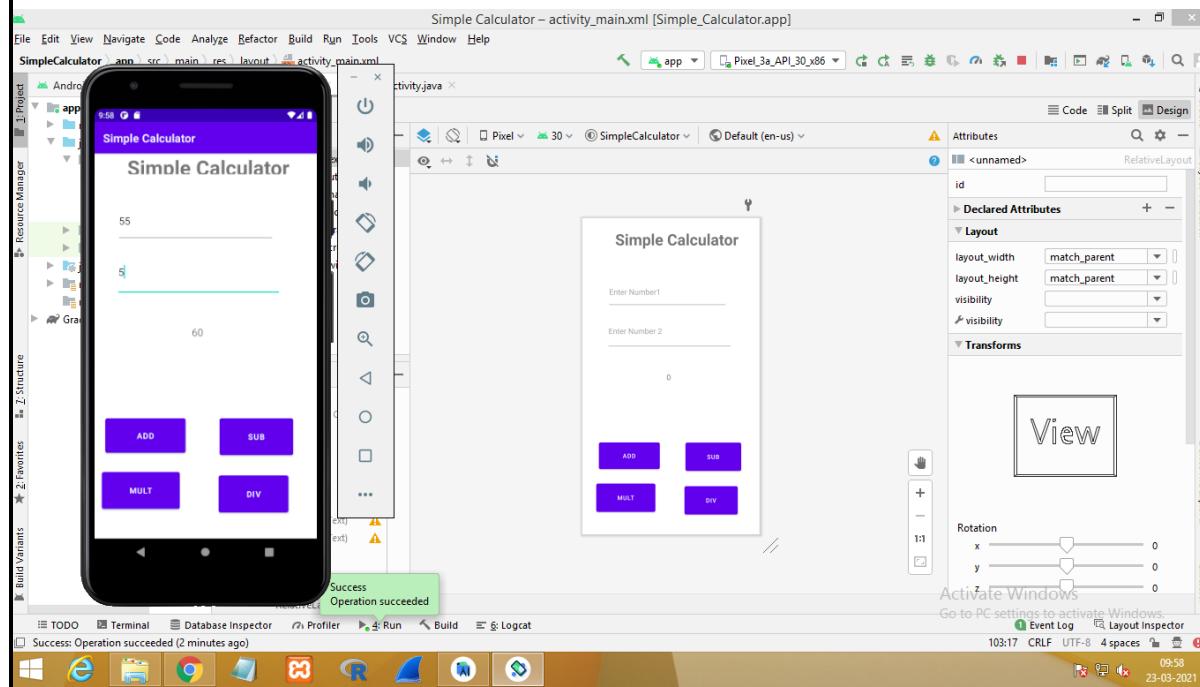
public class MainActivity extends AppCompatActivity {
    EditText e1,e2;
    TextView tv;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        e1 = (EditText) findViewById(R.id.editText1);
        e2 = (EditText) findViewById(R.id.editText2);
        tv = (TextView) findViewById(R.id.tv1);
    }
    public void add(View V){
        int a1=Integer.parseInt(e1.getText().toString());
        int a2=Integer.parseInt(e2.getText().toString());
        int res=a1+a2;
        tv.setText(""+res);
    }
    public void sub(View V){
        int a1=Integer.parseInt(e1.getText().toString());
        int a2=Integer.parseInt(e2.getText().toString());
        int res=a1-a2;
        tv.setText(""+res);
    }
    public void mul(View V){
        int a1=Integer.parseInt(e1.getText().toString());
        int a2=Integer.parseInt(e2.getText().toString());
        int res=a1*a2;
        tv.setText(""+res);
    }
    public void div(View V){
        float a1=Integer.parseInt(e1.getText().toString());
        float a2=Integer.parseInt(e2.getText().toString());
        float res=a1/a2;
        tv.setText(""+res);
    }
}

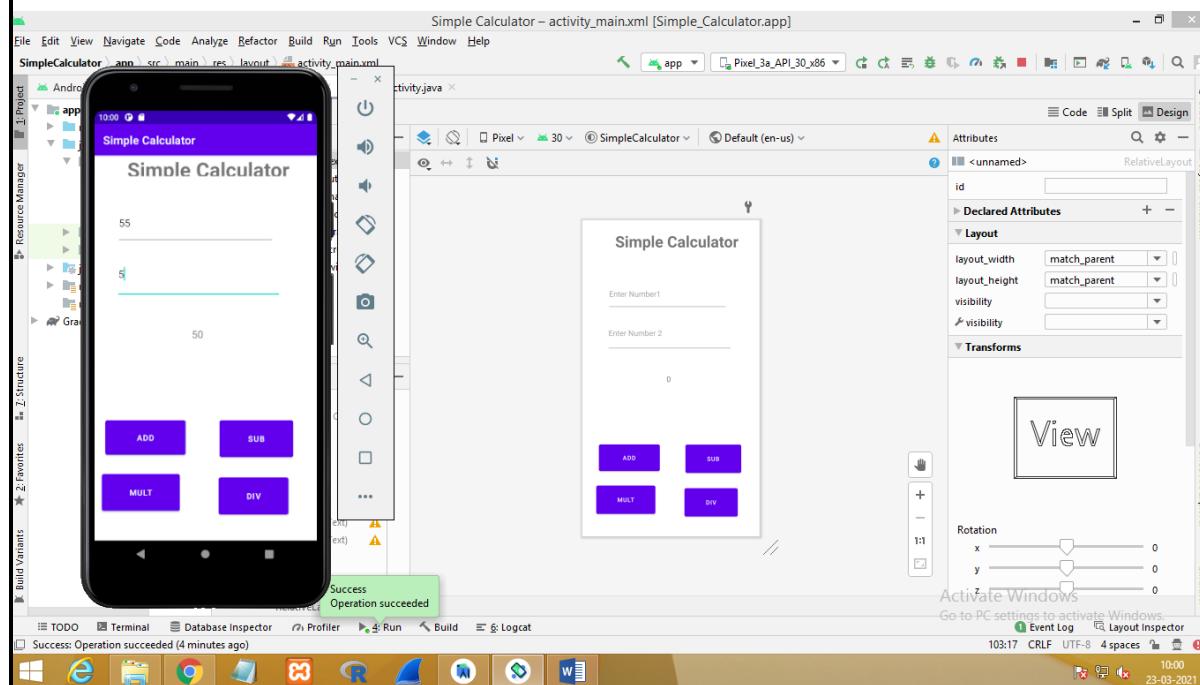
```

Result

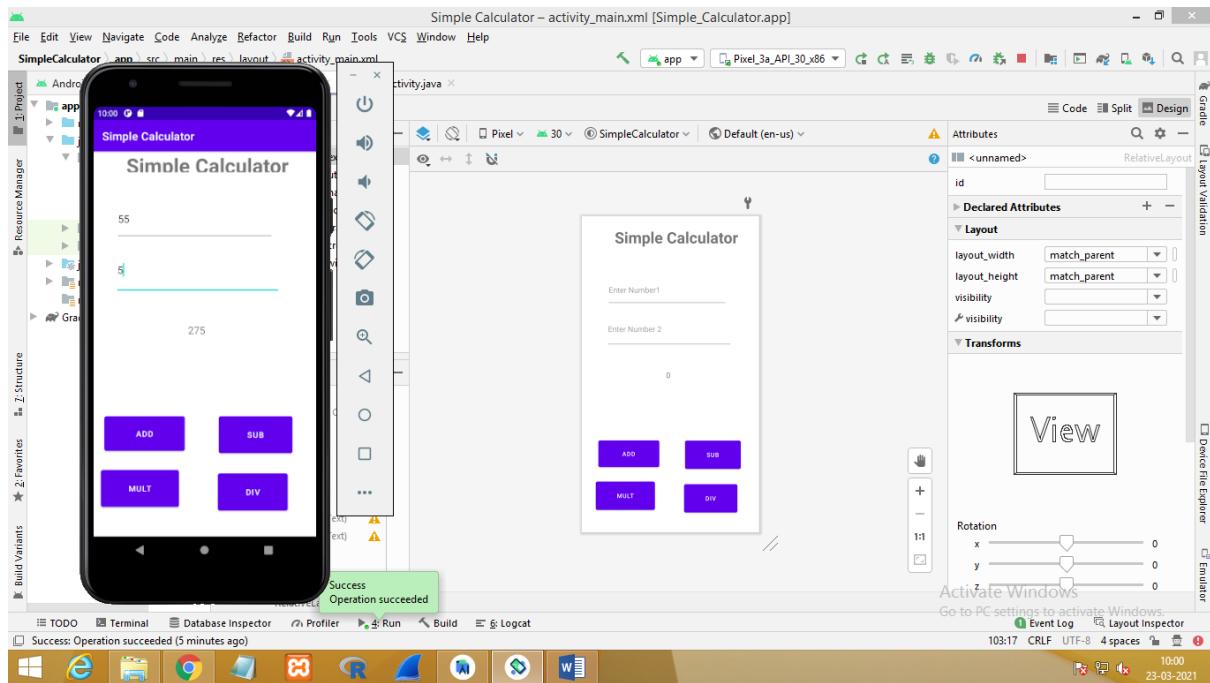
Add



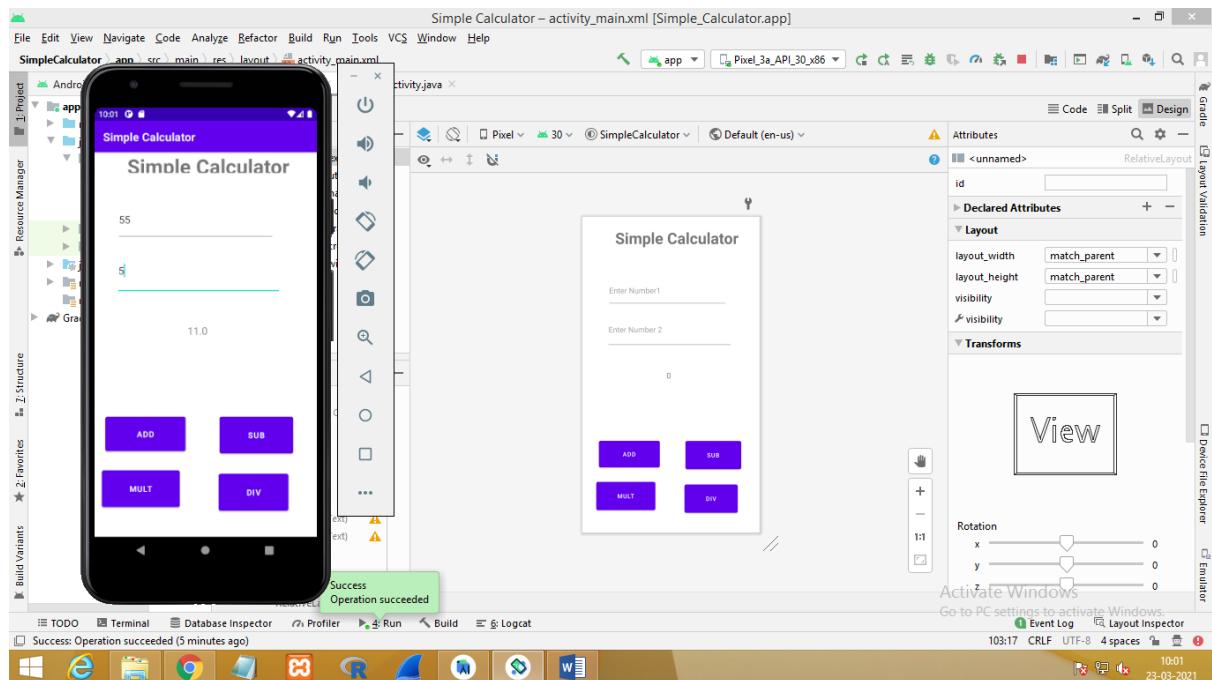
SUB



MUL



DIV



**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.

**activity\_signup.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout android="http://schemas.android.com/apk/res/android"
    app="http://schemas.android.com/apk/res-auto"
    tools="http://schemas.android.com/tools" layout_width="match_parent"
    layout_height="match_parent"
    orientation="vertical"
    context=".SignUpActivity">
    <TextView
        layout_width="match_parent"
        layout_height="wrap_content"
        textSize="48sp"
        textAlign="center"
        text="Sign Up" />
    <EditText
        id="@+id/emailEditText"
        layout_width="match_parent"
        layout_height="wrap_content"
        xmlns:layout_margin="4dp"
        textSize="24sp"
```

```
        hint="Email ID"  
    />  
  
    <EditText  
        id="@+id/passwordEditText"  
        layout_width="match_parent"  
        layout_height="wrap_content"  
        layout_margin="4dp"  
        layout_marginTop="32dp"  
        textSize="24sp"  
        inputType="textPassword"  
        hint="Password"  
    />  
  
    <Button  
        id="@+id/signUpBtn"  
        layout_width="match_parent"  
        layout_height="wrap_content"  
        layout_margin="4dp"  
        text="Sign Up"  
    />  
  
    </LinearLayout>
```

SignUpActivity.java

```
import .AppCompatActivity;  
import .Intent;  
import .Bundle;  
import .View;  
import .Button;  
import  
.EditText;  
import .Toast;  
import .Pattern;  
  
public class SignUpActivity extends AppCompatActivity {  
    EditText emailEditText, passwordEditText;
```

Button

```

signUpBtn;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_signup);
emailEditText = findViewById(R.id.emailEditText);
passwordEditText = findViewById(R.id.passwordEditText); signUpBtn =
findViewById(R.id.signUpBtn); signUpBtn.setOnClickListener(new
View.OnClickListener() {@Override
public void onClick(View v) {
String email = emailEditText.getText().toString();
String password = passwordEditText.getText().toString();if
(!isValidPassword(password)) {
Toast.makeText(SignUpActivity.this, "Password doesn't match rules"
, Toast.LENGTH_SHORT).show();
return;
}
}
Intent intent = new Intent(SignUpActivity.this, LoginActivity.class);
intent.putExtra("email", email);
intent.putExtra("password", password);
startActivity(intent);
}
});
}
Pattern lowerCase = Pattern.compile("^[a-z]*$");
Pattern upperCase = Pattern.compile("^[A-Z]*$");
Pattern number = Pattern.compile("^[0-9]*$");
Pattern specialCharacter = Pattern.compile("^[^a-zA-Z0-9]*$");
private Boolean isValidPassword(String password) {
// Checks if password length is less than 8
}

```

```
if (password.length() < 8) {  
    return false;  
}  
  
// Returns false if password doesn't contain a lower case character  
if (!lowerCase.matcher(password).matches()) {  
    return false;  
}  
  
// Returns false if password doesn't contain an upper case character  
if (!upperCase.matcher(password).matches()) {  
    return false;  
}  
  
// Returns false if password doesn't contain a number  
if (!number.matcher(password).matches()) {  
  
    return false;  
}  
  
// Returns false if password doesn't contain a special character  
if (!specialCharacter.matcher(password).matches()) {  
    return false;  
}  
  
return true;  
}  
}  
  
activity_login.xml  
  
<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout android="http://schemas.android.com/apk/res/android"  
app="http://schemas.android.com/apk/res-auto"  
tools="http://schemas.android.com/tools" layout_width="match_parent"  
android:layout_height="match_parent"  
orientation="vertical"  
context=".SignUpActivity">  
<TextView  
layout_width="match_parent"
```

```
layout_height="wrap_content"
textSize="48sp"
textAlignment="center"
text="Login" />
<EditText
    id="@+id/emailEditText"
    layout_width="match_parent"

    layout_height="wrap_content"
    layout_margin="4dp"
    textSize="24sp"
    hint="Email ID"
/>
<EditText
    id="@+id/passwordEditText"
    layout_width="match_parent"
    layout_height="wrap_content"
    layout_margin="4dp"
    layout_marginTop="32dp"
    textSize="24sp"
    inputType="textPassword"
    hint="Password"
/>
<Button
    id="@+id/loginBtn"
    "
    layout_width="match_parent"
    layout_height="wrap_content"
    layout_margin="4dp"
    text="Login"
/>
</LinearLayout
>LoginActivity.jav
```

```
a

import .AppCompatActivity;
import .Intent;
import .Bundle;

import .View;
import .Button;
import
.EditText;
import .Toast;

public class LoginActivity extends AppCompatActivity {
    EditText emailEditText, passwordEditText;
    Button
    loginBtn;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login); emailEditText
        = findViewById(R.id.emailEditText);
        passwordEditText = findViewById(R.id.passwordEditText); loginBtn =
        findViewById(R.id.loginBtn);
        String registeredEmail = getIntent().getStringExtra("email");
        String registeredPassword = getIntent().getStringExtra("password");
        loginBtn.setOnClickListener(new View.OnClickListener() { @Override
            public void onClick(View v) {
                String email = emailEditText.getText().toString();
                String password = passwordEditText.getText().toString();
                if (registeredEmail.equals(email) && registeredPassword.equals(password))
                {
                    Intent intent = new Intent(LoginActivity.this, LoginSuccessActivity.class);
                    startActivity(intent);
                } else {

```

```

        Toast.makeText(LoginActivity.this, "Invalid Credentials", Toast.LENGTH_SHORT).show();
    }
}
});
}
}

activity_login_success.xml

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout android="http://schemas.android.com/apk/res/android"
    app="http://schemas.android.com/apk/res-auto"
    tools="http://schemas.android.com/tools" layout_width="match_parent"
    layout_height="match_parent"
    orientation="vertical"
    context=".LoginSuccessActivity">
    <TextView
        layout_width="match_parent"
        layout_height="wrap_content"
        text="Login Successful"
        textAlign="center"
        textSize="36sp"/>
</LinearLayout>

```

LoginSuccessActivity.java

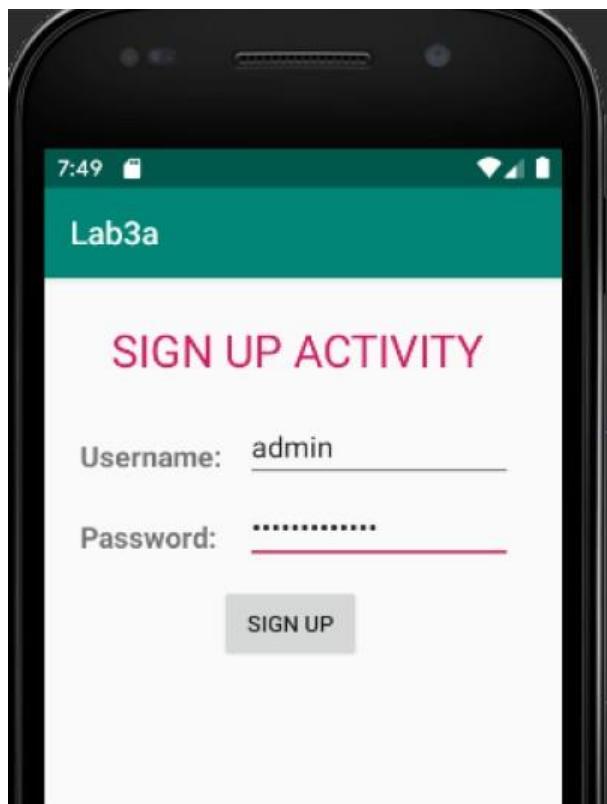
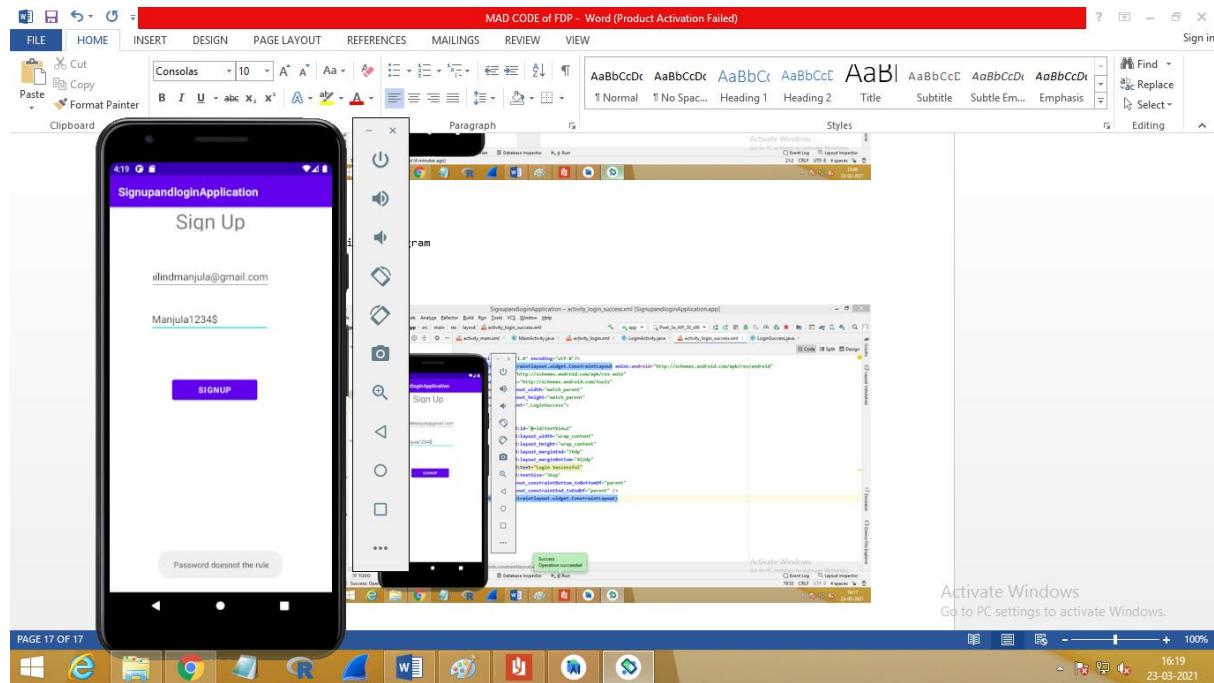
```

import .AppCompatActivity;
import .Bundle;

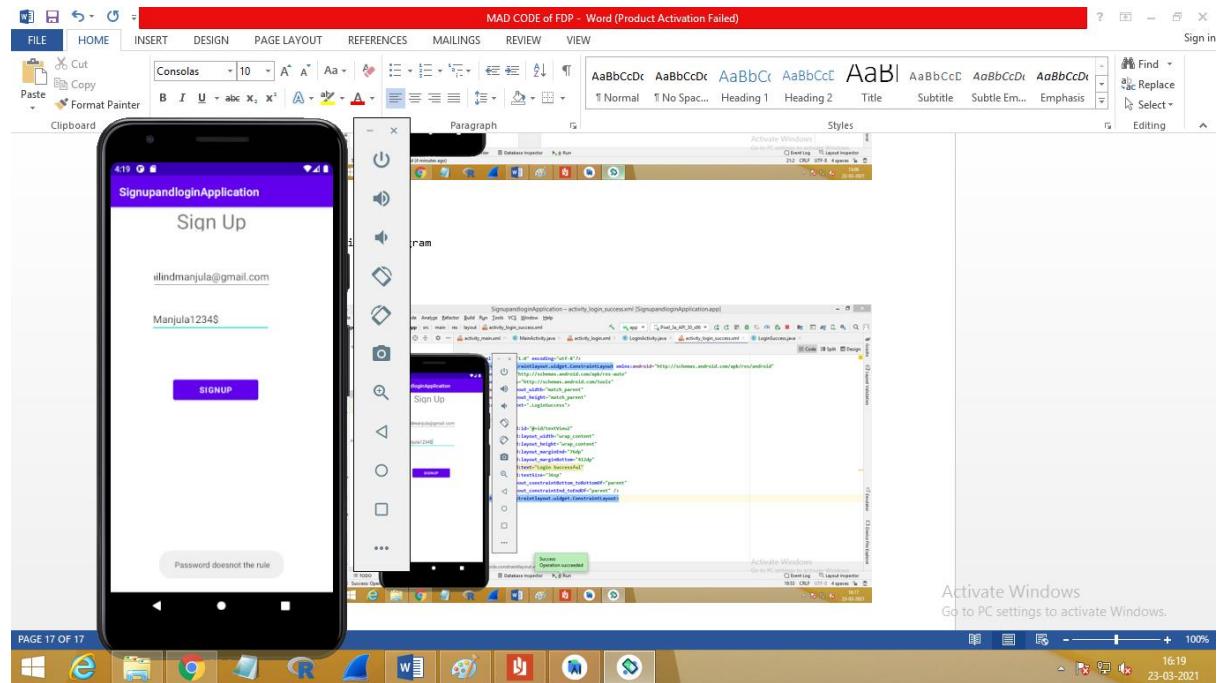
public class LoginSuccessActivity extends AppCompatActivity {

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

```



Example 2: output



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

## **Xml Code**

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

    <TextView
        android:id="@+id/textView"
        android:layout_width="423dp"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="27dp"
        android:layout_marginBottom="679dp"
        android:text="WALL PAPER CHANGE APPLICATION"
        android:textSize="30sp"
        android:textStyle="bold" />

    <Button
        android:id="@+id/button"
        android:layout_width="364dp"
        android:layout_height="94dp"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="62dp"
        android:layout_marginBottom="428dp"
        android:text="CLICK HERE TO CHANGE WALL PAPER" />
</RelativeLayout>
```

```
java Code
package com.example.wallpaperchangeapplication;

import androidx.appcompat.app.AppCompatActivity;

import android.app.WallpaperManager;
import android.graphics.Bitmap;
import android.graphics.drawable.BitmapDrawable;
import android.graphics.drawable.Drawable;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

import java.io.IOException;
import java.util.Timer;
import java.util.TimerTask;

public class MainActivity extends AppCompatActivity
    Button changewallpaper;
    Timer mytimer;
    Drawable drawable;
    WallpaperManager wpm;
```

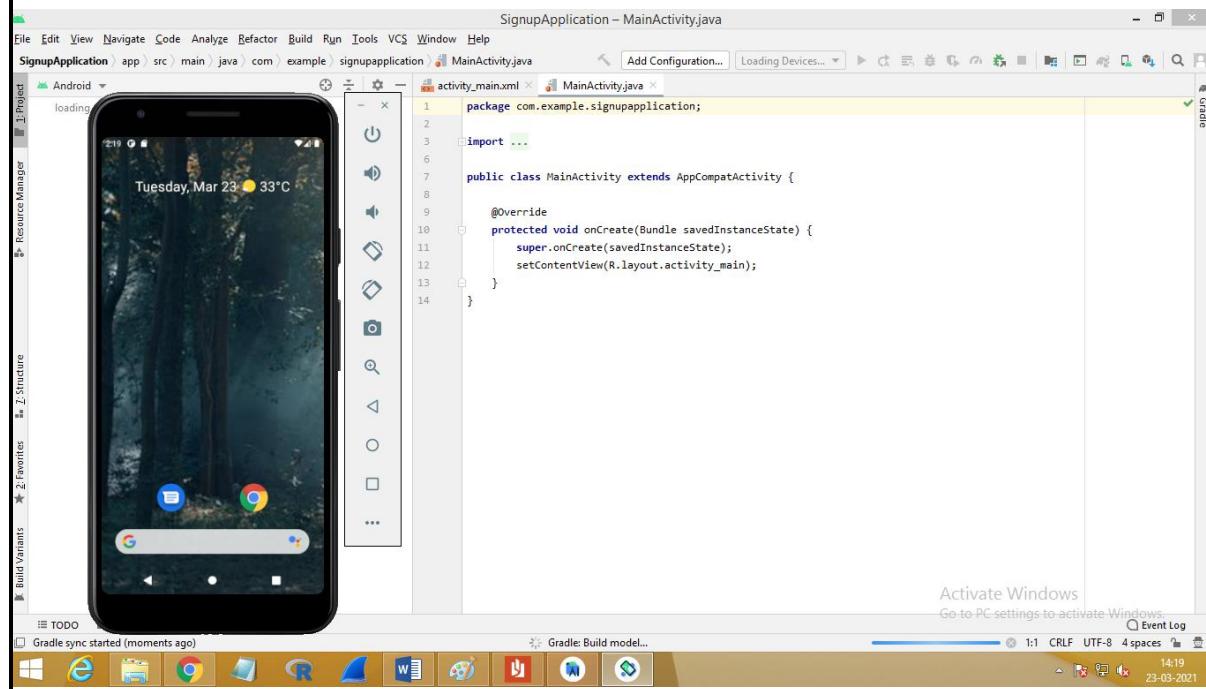
@Ovannido

```

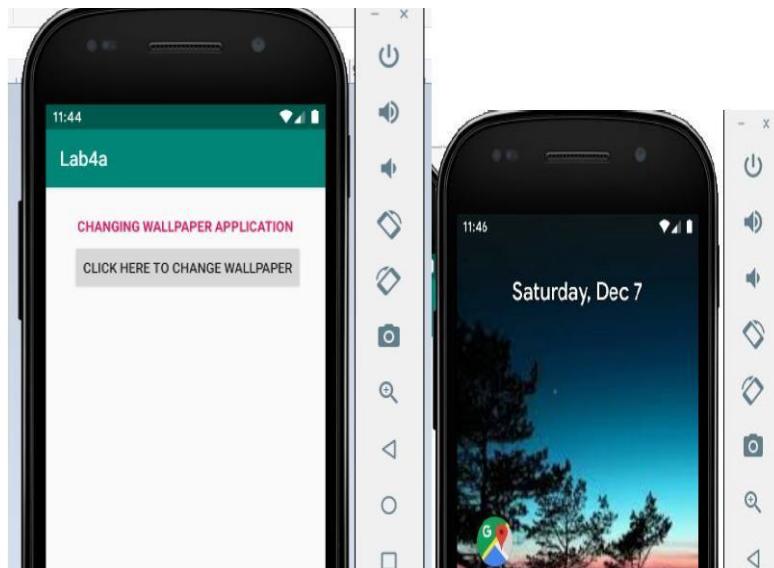
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    mytimer=new Timer();
    wpm=WallpaperManager.getInstance(this);
    changewallpaper=findViewById(R.id.button);
    changewallpaper.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            setWallpaper();
        }
    });
}
private void setWallpaper(){
    mytimer.schedule(new TimerTask() {
        @Override
        public void run() {
            if (prev==1){
                drawable=getResources().getDrawable(R.drawable.image1);
                prev=2;
            }
            else if (prev==2) {
                drawable = getResources().getDrawable(R.drawable.image2);
                prev = 3;
            }
            else if (prev==3){
                drawable = getResources().getDrawable(R.drawable.image3);
                prev=4;
            }
            else{
                drawable = getResources().getDrawable(R.drawable.image4);
                prev=1;
            }
            Bitmap wallpaper=((BitmapDrawable)drawable).getBitmap();
            try{
                wpm.setBitmap(wallpaper);
            }
            catch (IOException e){
                e.printStackTrace();
            }
        }
    },0,3000);
}
}

```

Output Wall paper Change



Example 2: output



**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.**

Xml code

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

    <TextView
        android:layout_width="300dp"
        android:layout_height="51dp"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="66dp"
        android:layout_marginBottom="590dp"
        android:text="COUNTER APPLICATION"
        android:textSize="24sp" />

    <Button
        android:id="@+id/btn_start"
        android:layout_width="141dp"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="233dp"
        android:layout_marginBottom="221dp"
        android:text="START" />

    <Button
        android:id="@+id/btn_stop"
        android:layout_width="146dp"
        android:layout_height="38dp"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="31dp"
        android:layout_marginBottom="226dp"
        android:text="STOP" />

    <TextView
        android:id="@+id/textView"
        android:layout_width="259dp"
        android:layout_height="31dp"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="70dp"
        android:layout_marginBottom="504dp"
        android:text="TextView" />

</RelativeLayout>
```

Java Code

```
package com.example.counterapplicationfinal;
```

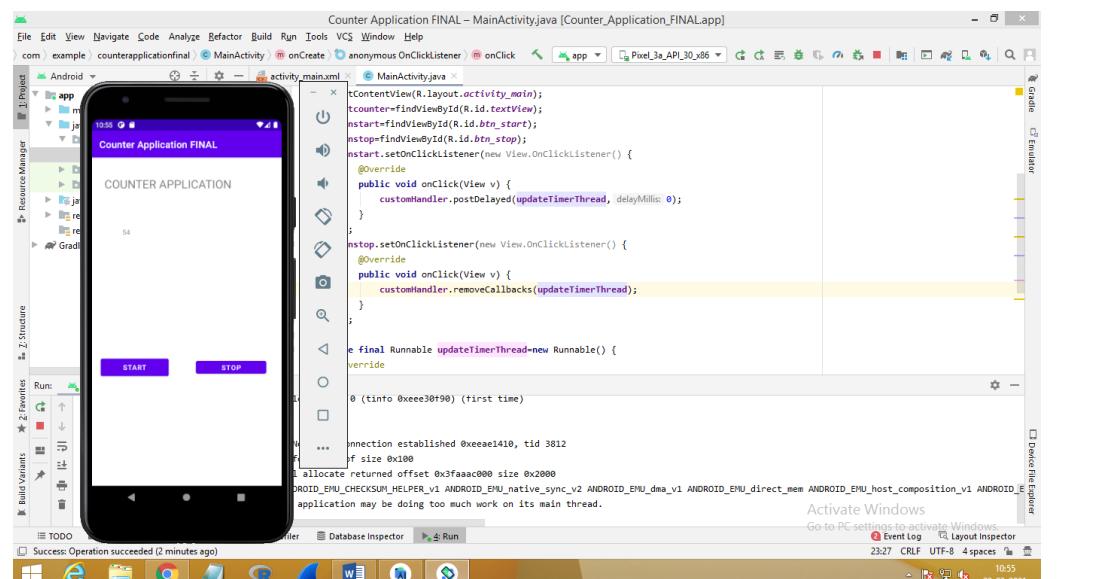
```

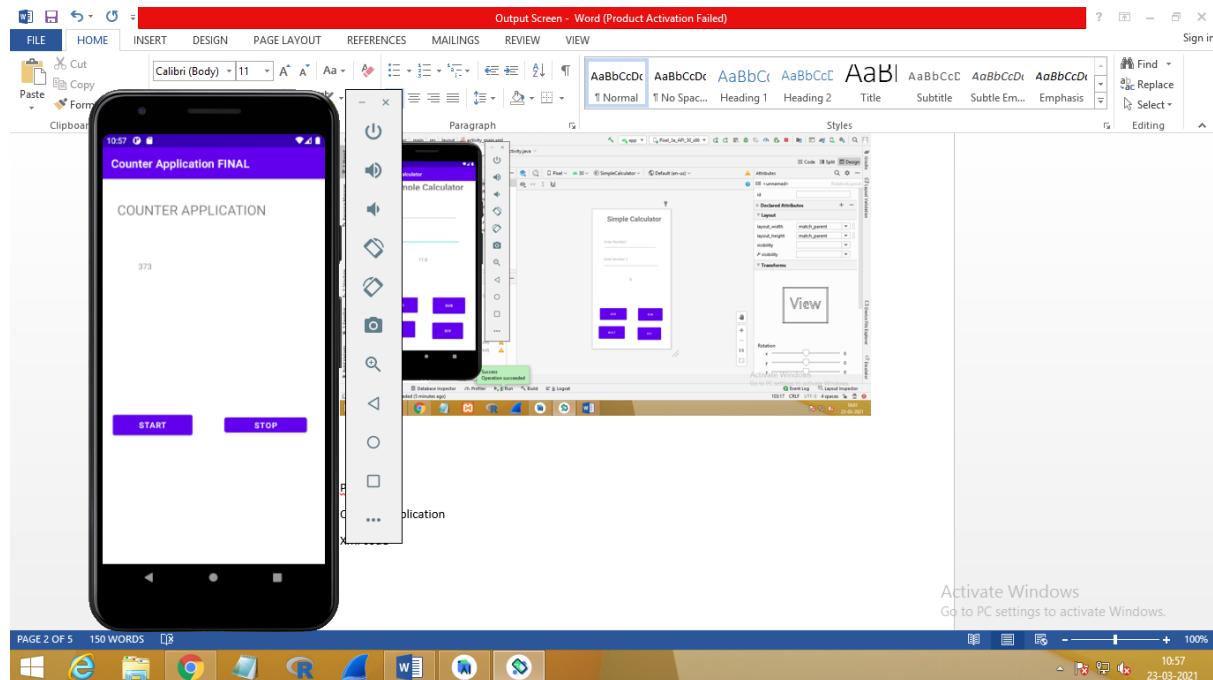
import androidx.appcompat.app.AppCompatActivity;

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

public class MainActivity extends AppCompatActivity {
    Button btnstart,btnstop;
    TextView txtcounter;
    int i=1;
    Handler customHandler=new Handler();
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        txtcounter=findViewById(R.id.textView);
        btnstart=findViewById(R.id.btn_start);
        btnstop=findViewById(R.id.btn_stop);
        btnstart.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                customHandler.postDelayed(updateTimerThread,0);
            }
        });
        btnstop.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                customHandler.removeCallbacks(updateTimerThread);
            }
        });
    }
    private final Runnable updateTimerThread=new Runnable() {
        @Override
        public void run() {
            txtcounter.setText(""+i);
            customHandler.postDelayed(this,1000);
            i++;
        }
    };
}

```





**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”.

PARSING XML AND JSON DATA		
PARSING XML AND JSON DATA	XML DATA	JSON Data
<input type="button" value="Parse XML Data"/>	City_Name: Mysore Latitude: 12.295 Longitude: 76.639 Temperature: 22 Humidity: 90%	City_Name: Mysore Latitude: 12.295 Longitude: 76.639 Temperature: 22 Humidity: 90%
<input type="button" value="Parse JSON Data"/>		

```
Notepad city.xml
<?xml version="1.0"?>
<record>
<place>
  <name>Mysore</name>
  <lat> 12.295</lat>
  <long>76.639</long>
  <temperature>22</temperature>
  <humidity>90%</humidity>
</place>
<place>
```

```

<name>Bangalore</name>
<lat> 12.97165</lat>
<long>77.5946</long>
<temperature>25</temperature>
<humidity>74%</humidity>
</place>
</records>

```

Notepad city.Json

[

```
{
  "name": "HASSAN",
  "lat": "12.295",
  "long": "76.639",
  "temperature": "22",
  "humidity": "92%"
},
```

```
{
  "name": "MANDYA",
  "lat": "12.97165",
  "long": "77.5946",
  "temperature": "25",
  "humidity": "74%"
}
```

]

XML

```

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

    <TextView
        android:layout_width="178dp"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="129dp"
        android:layout_marginBottom="560dp"
        android:text="PARSER"
        android:textSize="36sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/button"
        android:layout_width="118dp"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="285dp"
        android:layout_marginBottom="500dp"
        android:onClick="parserxml"
        android:text="XML" />

```

```

<Button
    android:id="@+id/button2"
    android:layout_width="127dp"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="39dp"
    android:layout_marginBottom="503dp"
    android:onClick="parsejson"
    android:text="JSON" />

<TextView
    android:id="@+id/display"
    android:layout_width="371dp"
    android:layout_height="406dp"

    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_marginTop="-45dp"
    android:layout_marginEnd="19dp"
    android:layout_marginBottom="21dp"
    android:text="" />

</RelativeLayout>

```

## Java Code

```

package com.example.parser;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.view.View;
import android.widget.TextView;
import android.widget.Toast;

import org.json.JSONArray;
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 java.nio.charset.StandardCharsets;

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

public class MainActivity extends AppCompatActivity {
    TextView display;

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

    public void parserxml(View V) {

```

```

try {
    InputStream is = getAssets().open("city.xml");
    DocumentBuilderFactory documentBuilderFactory = DocumentBuilderFactory.newInstance();
    DocumentBuilder documentBuilder = documentBuilderFactory.newDocumentBuilder();
    Document document = documentBuilder.parse(is);
    StringBuilder stringBuilder = new StringBuilder();
    stringBuilder.append("XML DATA");
    stringBuilder.append("\n-----");
    NodeList nodeList = document.getElementsByTagName("place");
    for (int i = 0; i < nodeList.getLength(); i++) {
        Node node = nodeList.item(i);
        if (node.getNodeType() == Node.ELEMENT_NODE) {
            Element element = (Element) node;
            stringBuilder.append("\n Name").append(getValue("name", element));
            stringBuilder.append("\n Latitude").append(getValue("lat", element));
            stringBuilder.append("\n Longitude").append(getValue("long", element));
            stringBuilder.append("\n Temperature").append(getValue("temperature", element));
            stringBuilder.append("\n Humidity").append(getValue("humidity", element));
            stringBuilder.append("\n-----");
        }
    }
    display.setText(stringBuilder.toString());
} catch (Exception e) {
    e.printStackTrace();
    Toast.makeText(MainActivity.this, "Error in reading XML File", Toast.LENGTH_LONG).show();
}

}

public void parsejson(View V) {
    String json;
    StringBuilder stringBuilder = new StringBuilder();
    try {
        InputStream is = getAssets().open("city.json");
        int size = is.available();
        byte[] buffer = new byte[size];
        is.read(buffer);
        json = new String(buffer, StandardCharsets.UTF_8);
        JSONArray jsonArray = new JSONArray(json);
        stringBuilder.append("JSON DATA");
        stringBuilder.append("\n-----");
        for (int i = 0; i < jsonArray.length(); i++) {
            JSONObject jsonObject = jsonArray.getJSONObject(i);
            stringBuilder.append("\n Name:").append(jsonObject.getString("name"));
            stringBuilder.append("\n Latitude:").append(jsonObject.getString("lat"));
            stringBuilder.append("\n Longitude").append(jsonObject.getString("long"));
            stringBuilder.append("\n Temperature").append(jsonObject.getString("temperature"));
            stringBuilder.append("\n Humidity").append(jsonObject.getString("humidity"));
            stringBuilder.append("\n-----");
            display.setText(stringBuilder.toString());
            is.close();
        }
    } catch (Exception e) {
        e.printStackTrace();
        Toast.makeText(MainActivity.this, "Error in reading Json File", Toast.LENGTH_LONG).show();
    }
}

```

```

private String getValue(String tag, Element element) {
    return element.getElementsByTagName(tag).item(0).getChildNodes().item(0).getNodeValue();
}
}

```

City Xml Code

```

<?xml version="1.0"?>
<records>
    <place>
        <name>Mysore</name>
        <lat> 12.295</lat>
        <long>76.639</long>
        <temperature>22</temperature>
        <humidity>90%</humidity>
    </place>
    <place>
        <name>Bangalore</name>
        <lat> 12.97165</lat>
        <long>77.5946</long>
        <temperature>25</temperature>
        <humidity>74%</humidity>
    </place>
</records>

```

City Json

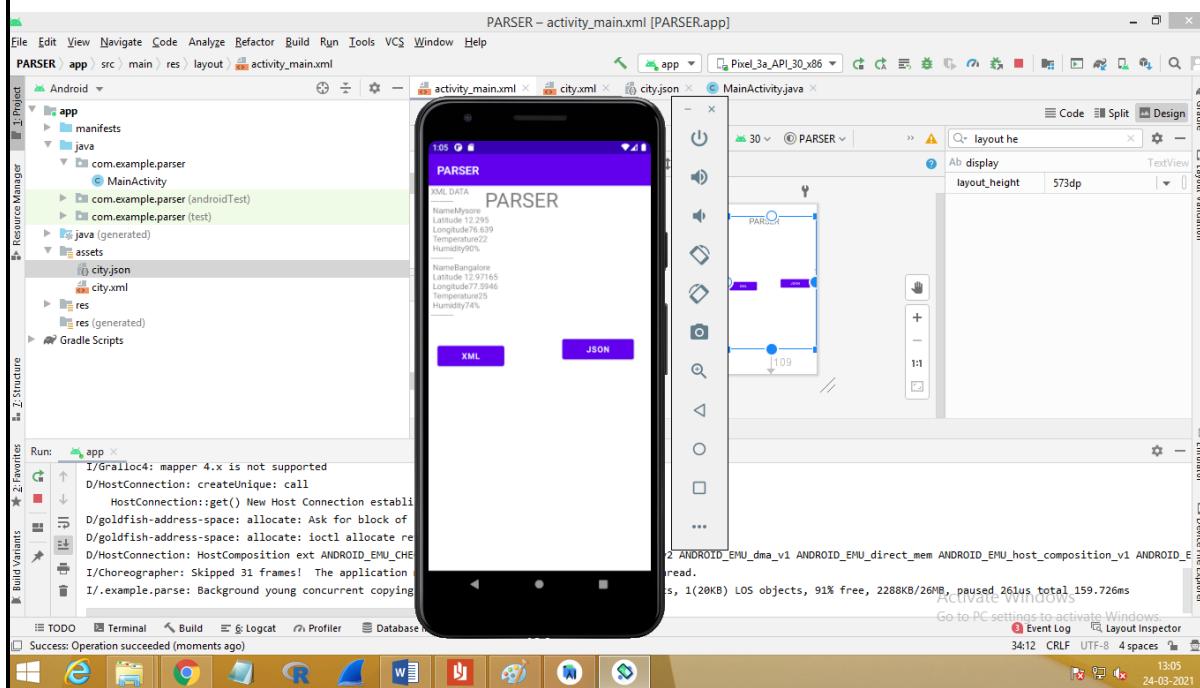
```

[
    {
        "name": "HASSAN",
        "lat": "12.295",
        "long": "76.639",
        "temperature": "22",
        "humidity": "92%"
    },
    {
        "name": "MANDYA",
        "lat": "12.97165",
        "long": "77.5946",
        "temperature": "25",
        "humidity": "74%"
    }
]

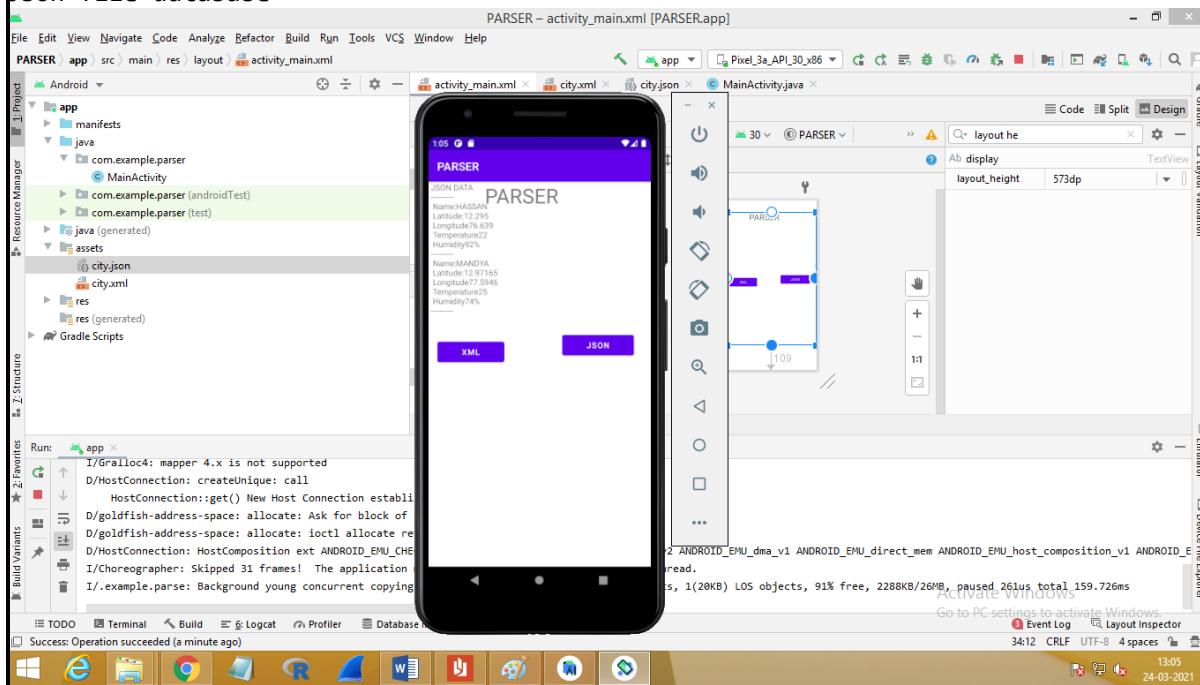
```

Code

Output xml database



### Json file database



**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.**



XML Code:

```

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

    <TextView
        android:layout_width="281dp"
        android:layout_height="75dp"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="53dp"
        android:layout_marginBottom="633dp"
        android:text="TEXT TO SPEECH"
        android:textSize="30sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.037" />

    <EditText
        android:id="@+id/editText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="24dp"
        android:layout_marginBottom="489dp"
        android:ems="10"
        android:inputType="textPersonName"
        android:text="Enter Text here"
        android:textSize="30sp" />

    <Button
        android:id="@+id/button"
        android:layout_width="215dp"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
    
```

```

        android:layout_marginEnd="94dp"
        android:layout_marginBottom="265dp"
        android:onClick="convert"
        android:text="CONVERT" />

    
```

Java Code

```

package com.example.texttospeech;

import androidx.appcompat.app.AppCompatActivity;

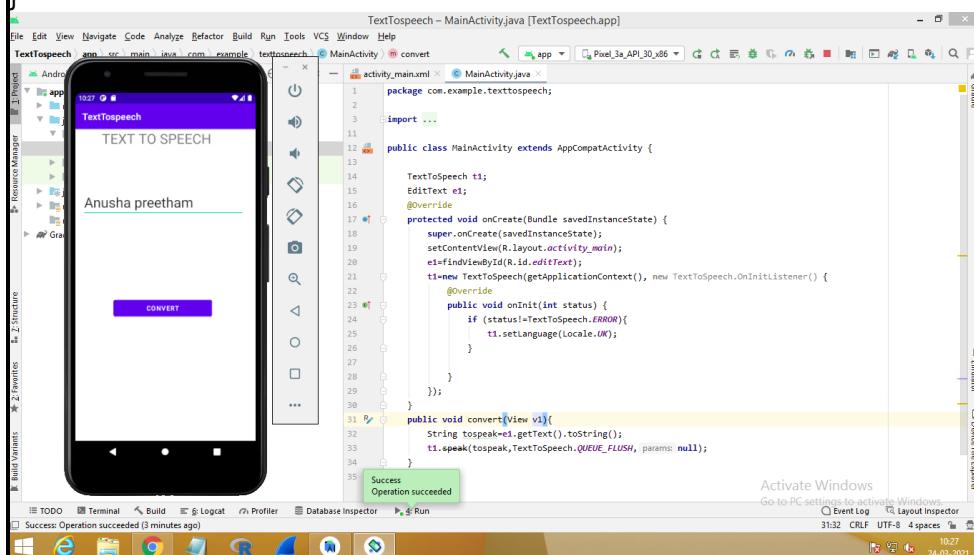
import android.os.Bundle;
import android.speech.tts.TextToSpeech;
import android.view.View;
import android.widget.EditText;

import java.util.Locale;

public class MainActivity extends AppCompatActivity {

    TextToSpeech t1;
    EditText e1;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        e1=findViewById(R.id.editText);
        t1=new TextToSpeech(getApplicationContext(), new TextToSpeech.OnInitListener() {
            @Override
            public void onInit(int status) {
                if (status!=TextToSpeech.ERROR){
                    t1.setLanguage(Locale.UK);
                }
            }
        });
    }
    public void convert(View v1){
        String tospeak=e1.getText().toString();
        t1.speak(tospeak,TextToSpeech.QUEUE_FLUSH,null);
    }
}

```



**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.

Call Application

Xml Code]

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

    <EditText
        android:id="@+id/phoneNumberEditText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="171dp"
        android:layout_marginBottom="647dp"
        android:ems="10"
        android:inputType="phone" />

    <Button
        android:id="@+id/clearBtn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="42dp"
        android:layout_marginBottom="640dp"
        android:text="clear"
        android:textSize="18sp" />

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:onClick="inputNumber"
        android:layout_marginEnd="294dp"
        android:layout_marginBottom="533dp"
        android:backgroundTint="#00BCD4"
        android:text="1" />

    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:onClick="inputNumber"
        android:layout_marginEnd="177dp"
```

```
        android:layout_marginBottom="532dp"
        android:backgroundTint="#00BCD4"
        android:text="2" />

<Button
    android:id="@+id/button3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:onClick="inputNumber"
    android:layout_marginEnd="55dp"
    android:layout_marginBottom="532dp"
    android:backgroundTint="#00BCD4"
    android:text="3" />

<Button
    android:id="@+id/button4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:onClick="inputNumber"
    android:layout_marginEnd="291dp"
    android:layout_marginBottom="458dp"
    android:backgroundTint="#00BCD4"
    android:text="4" />

<Button
    android:id="@+id/button5"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:onClick="inputNumber"
    android:layout_marginEnd="172dp"
    android:layout_marginBottom="457dp"
    android:backgroundTint="#00BCD4"
    android:text="5" />

<Button
    android:id="@+id/button6"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:onClick="inputNumber"
    android:layout_marginEnd="55dp"
    android:layout_marginBottom="455dp"
    android:backgroundTint="#00BCD4"
    android:text="6" />

<Button
    android:id="@+id/button7"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:onClick="inputNumber"
    android:layout_marginEnd="291dp"
    android:layout_marginBottom="371dp"
    android:backgroundTint="#00BCD4"
```

```
    android:text="7" />

<Button
    android:id="@+id/button8"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:onClick="inputNumber"
    android:layout_marginEnd="172dp"
    android:layout_marginBottom="371dp"
    android:backgroundTint="#00BCD4"
    android:text="8" />

<Button
    android:id="@+id/button9"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:onClick="inputNumber"
    android:layout_marginEnd="51dp"
    android:layout_marginBottom="371dp"
    android:backgroundTint="#00BCD4"
    android:text="9" />

<Button
    android:id="@+id/star"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:onClick="inputNumber"
    android:layout_marginEnd="286dp"
    android:layout_marginBottom="277dp"
    android:backgroundTint="#00BCD4"
    android:text "*" />

<Button
    android:id="@+id/zero"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:onClick="inputNumber"
    android:layout_marginEnd="164dp"
    android:layout_marginBottom="279dp"
    android:backgroundTint="#00BCD4"
    android:text="0" />

<Button
    android:id="@+id/hash"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:onClick="inputNumber"
    android:layout_marginEnd="52dp"
    android:layout_marginBottom="276dp"
    android:backgroundTint="#00BCD4"
    android:text="#" />
```

```

<Button
    android:id="@+id/saveBtn"
    android:layout_width="157dp"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"

    android:layout_marginEnd="211dp"
    android:layout_marginBottom="176dp"
    android:backgroundTint="#03A9F4"
    android:text="save" />

<Button
    android:id="@+id/callBtn"
    android:layout_width="157dp"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="48dp"
    android:layout_marginBottom="179dp"
    android:backgroundTint="#8BC34A"
    android:text="call" />
</RelativeLayout>

```

Java Code

```

package com.example.callapplication;

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 {
    EditText phoneNumberEditText;
    Button clearBtn, saveBtn, callBtn;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        phoneNumberEditText=findViewById(R.id.phoneNumberEditText);
        clearBtn=findViewById(R.id.clearBtn);
        callBtn=findViewById(R.id.callBtn);
        saveBtn=findViewById(R.id.saveBtn);
        clearBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                phoneNumberEditText.setText("");
            }
        });
        callBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String phoneNumber=phoneNumberEditText.getText().toString();
                Intent intent=new Intent( Intent.ACTION_DIAL);

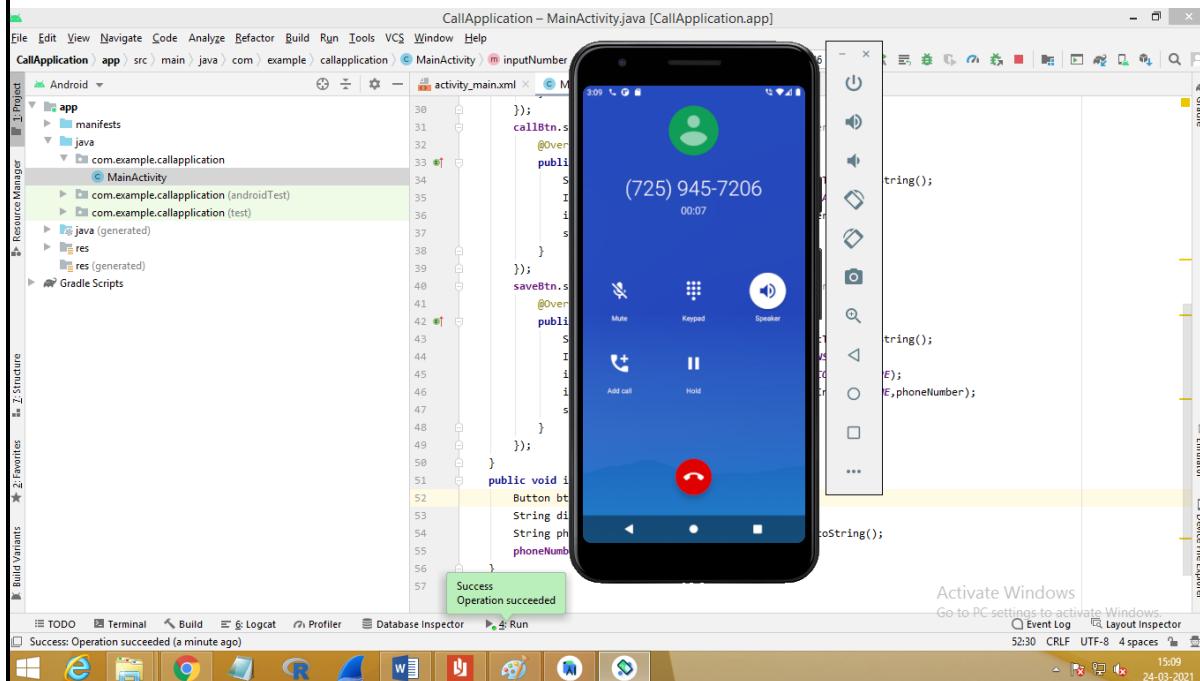
```

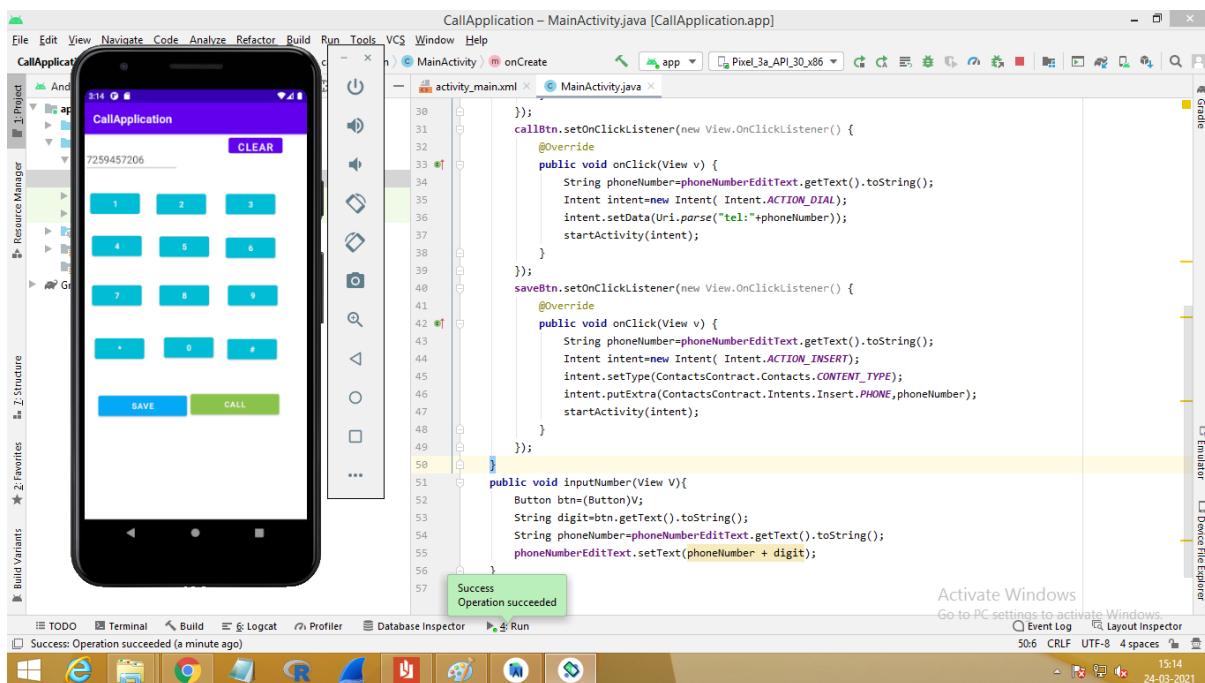
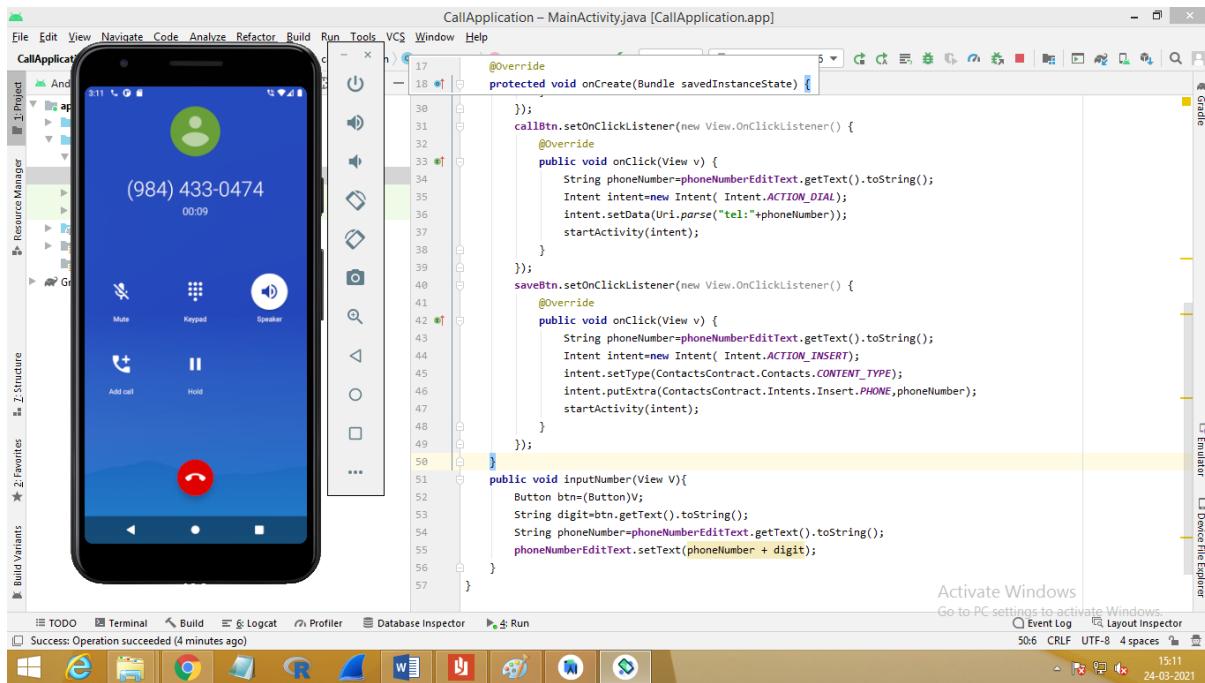
```

        intent.setData(Uri.parse("tel:"+phoneNumber));
        startActivity(intent);
    }
});
saveBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String phoneNumber=phoneNumberEditText.getText().toString();
        Intent intent=new Intent( Intent.ACTION_INSERT);
        intent.setType(ContactsContract.Contacts.CONTENT_TYPE);
        intent.putExtra(ContactsContract.Inserts.PHONE,phoneNumber);
        startActivity(intent);
    }
});
}
public void inputNumber(View V){
    Button btn=(Button)V;
    String digit=btn.getText().toString();
    String phoneNumber=phoneNumberEditText.getText().toString();
    phoneNumberEditText.setText(phoneNumber + digit);
}
}

```

## Output



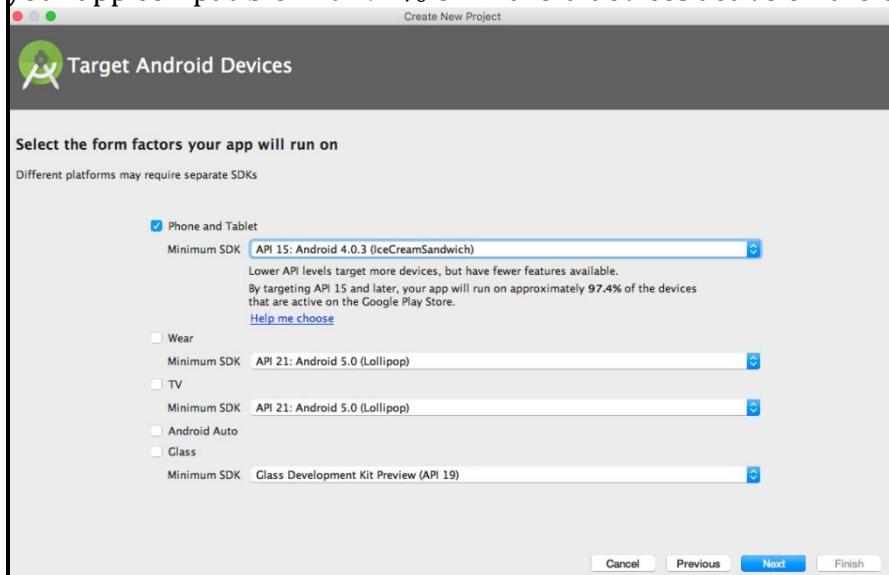


## Starting an Android Studio project

After you have successfully installed the Android Studio IDE, double-click the Android Studio application icon to start it. Choose **Start a new Android Studio project** in the Welcome window, and name the project the same name that you want to use for the app. When choosing a unique Company Domain, keep in mind that apps published to the Google Play must have a unique package name. Since domains are unique, prepending the app's name with your name, or your company's domain name, should provide an adequately unique package name. If you are not planning to publish the app, you can accept the default example domain. Be aware that changing the package name later is extra work.

## **Choosing target devices and the minimum SDK**

When choosing Target Android Devices, Phone and Tablet are selected by default, as shown in the figure below. The choice shown in the figure for the Minimum SDK — **API 15: Android 4.0.3 (IceCreamSandwich)** — makes your app compatible with 97% of Android devices active on the Google Play Store.

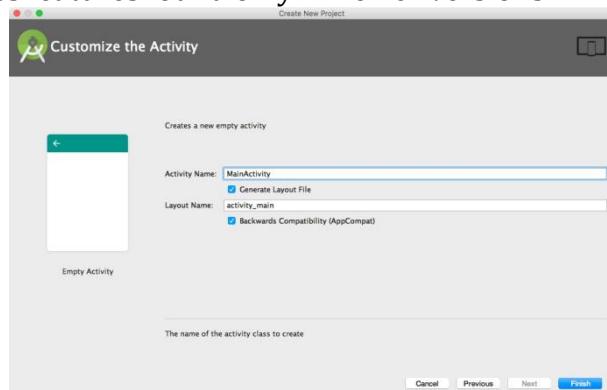


Different devices run different versions of the Android system, such as Android 4.0.3 or Android 4.4. Each successive version often adds new APIs not available in the previous version. To indicate which set of APIs are available, each version specifies an API level. For instance, Android 1.0 is API level 1 and Android 4.0.3 is API level 15. The Minimum SDK declares the minimum Android version for your app. Each successive version of Android provides compatibility for apps that were built using the APIs from previous versions, so your app should *always* be compatible with future versions of Android while using the documented Android APIs.

### Choosing a template

Android Studio pre-populates your project with minimal code for an activity and a screen layout based on a *template*. A variety of templates are available, ranging from a virtually blank template (Add No Activity) to various types of activities. [171.1: Create Your First Android App](#) You can customize the activity after choosing your template. For example, the Empty Activity template provides a single activity accompanied by a single layout resource for the screen. You can choose to accept the commonly used name for the activity (such as **MainActivity**) or change the name on the Customize the Activity screen. Also, if you use the Empty Activity template, be sure to check the following if they are not already checked:

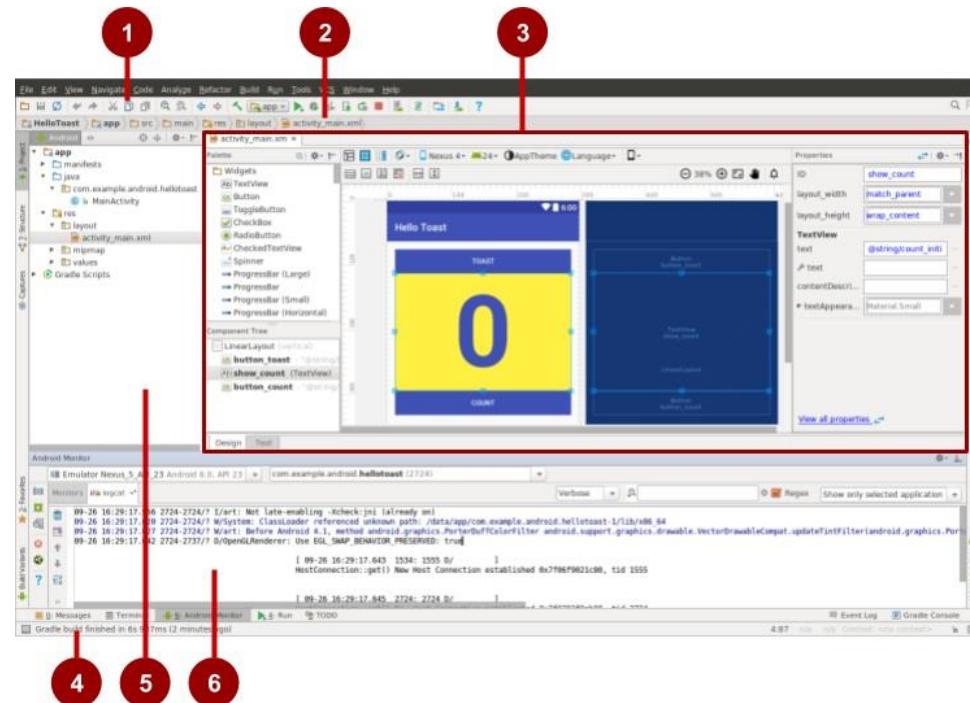
**Generate Layout file:** Leave this checked to create the layout resource connected to this activity, which is usually named **activity\_main.xml**. The layout defines the user interface for the activity. **Backwards Compatibility (AppCompat):** Leave this checked to include the AppCompat library so that the app is compatible with previous versions of Android even if it uses features found only in newer versions.



Android Studio creates a folder for the newly created project in the `AndroidStudioProjects` folder on your computer.

### Android Studio window panes

The Android Studio main window is made up of several logical areas, or *panes*, as shown in the figure below.



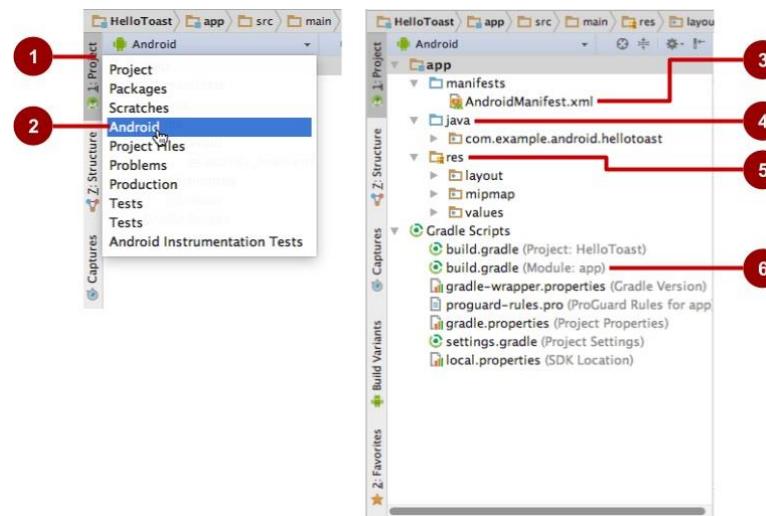
In the above figure:

1. **The Toolbar.** The toolbar carries out a wide range of actions, including running the Android app and launching Android tools.
2. **The Navigation Bar.** The navigation bar allows navigation through the project and open files for editing. It provides a more compact view of the project structure.
3. **The Editor Pane.** This pane shows the contents of a selected file in the project. For example, after selecting a layout (as shown in the figure), this pane shows the layout editor with tools to edit the layout. After selecting a Java code file, this pane shows the code with tools for editing the code.
4. **The Status Bar.** The status bar displays the status of the project and Android Studio itself, as well as any warnings or messages. You can watch the build progress in the status bar.
5. **The Project Pane.** The project pane shows the project files and project hierarchy.
6. **The Monitor Pane.** The monitor pane offers access to the TODO list for managing tasks, the Android Monitor for monitoring app execution (shown in the figure), the logcat for viewing log messages, and the Terminal application for performing Terminal activities.

**Tip:** You can organize the main window to give yourself more screen space by hiding or moving panes. You can also use keyboard shortcuts to access most features. See [Keyboard Shortcuts](#) for a complete list.

### Exploring a project

Each project in Android Studio contains the `AndroidManifest.xml` file, component source-code files, and associated resource files. By default, Android Studio organizes your project files based on the file type, and displays them within the Project: Android view in the left tool pane, as shown below. The view provides quick access to your project's key files. To switch back to this view from another view, click the vertical **Project** tab in the far left column of the Project pane, and choose **Android** from the pop-up menu at the top of the Project pane, as shown in the figure below



In the figure above:

1. The **Project** tab. Click to show the project view.
2. The **Android** selection in the project drop-down menu.
3. The **AndroidManifest.xml** file. Used for specifying information about the app for the Android runtime environment. The template you choose creates this file.
4. The **java** folder. This folder includes activities, tests, and other components in Java source code. Every activity, service, and other component is defined as a Java class, usually in its own file. The name of the first activity (screen) the user sees, which also initializes app-wide resources, is customarily `MainActivity`.
5. The **res** folder. This folder holds resources, such as XML layouts, UI strings, and images. An activity usually is associated with an XML resource file that specifies the layout of its views. This file is usually named after its activity or function.
6. The **build.gradle (Module: App)** file. This file specifies the module's build configuration. The template you choose creates this file, which defines the build configuration, including the `minSdkVersion` attribute that declares the minimum version for the app, and the `targetSdkVersion` attribute that declares the highest (newest) version for which the app has been optimized. This file also includes a list of *dependencies*, which are libraries required by the code — such as the `AppCompat` library for supporting a wide range of Android versions.

**Program 9. (PART-B): 5. Create an application to demonstrate a basic media player that allows the user to Forward Backward, Play and Pause anaudio. Also, make use of the indicator in the seek bar to move the audio forward or backward as required.**

XML-CODE:

```

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

    <SeekBar
        android:id="@+id/seekBar"
        android:layout_width="255dp"
        android:layout_height="28dp"
        android:layout_alignParentEnd="true"
    
```

```

        android:layout_alignParentBottom="true"
        android:layout_marginEnd="66dp" android:layout_marginBottom="311dp"
    />

<ImageButton android:id="@+id/rewind"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="256dp"
    android:layout_marginBottom="219dp"
    app:srcCompat="@android:drawable/ic_media_rew" />

<ImageButton android:id="@+id/playButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="168dp"
    android:layout_marginBottom="223dp"
    app:srcCompat="@android:drawable/ic_lock_power_off" />

<ImageButton android:id="@+id/forward"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="79dp" android:layout_marginBottom="220dp"
    app:srcCompat="@android:drawable/ic_media_ff" />

</RelativeLayout>

```

#### Java CODE:

```

import androidx.appcompat.app.AppCompatActivity;
import android.annotation.SuppressLint;
import android.media.MediaPlayer; import
android.os.Bundle;
import android.os.Handler;
import android.view.View;
import android.widget.ImageButton;
import android.widget.SeekBar; import
android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    private ImageButton playButton,forward,rewind;
    private SeekBar seekbar;

```

```

private MediaPlayer mediaPlayer; private
Handler handler = new Handler();
@SuppressLint("ClickableViewAccessibility")

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main); playButton =
    findViewById(R.id.playButton); forward =
    findViewById(R.id.forward);
    rewind = findViewById(R.id.rewind);

    seekbar = findViewById(R.id.seekBar);
    prepareMediaPlayer();
    seekbar.setMax(100);
    playButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if(mediaPlayer.isPlaying()){
                handler.removeCallbacks(updater);
                mediaPlayer.pause();

            }else {
                mediaPlayer.start();

                updateSeekBar();
            }
        }
    });
}

forward.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if(mediaPlayer.getDuration()>mediaPlayer.getCurrentPosition() +
10000){
            mediaPlayer.seekTo(mediaPlayer.getCurrentPosition() + 10000);
            updateSeekBar();
        }
    }
});

```

**Program 10: (PART-B): 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".**

XML-CODE:

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="123dp"
        android:layout_marginBottom="630dp"
        android:text="Async Task"
        android:textSize="36sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent" app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
    <Button
        android:id="@+id/buttonstart"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="167dp"
        android:layout_marginBottom="441dp"
        android:text="Start" />
    <Button
        android:id="@+id/buttonstop"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"

```

```

        android:layout_alignParentBottom="true"
        android:layout_marginEnd="169dp"
        android:layout_marginBottom="328dp"
        android:text="Stop" />
<TextView android:id="@+id/marqueeText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginStart="50dp"
        android:layout_marginTop="250dp"
        android:layout_marginEnd="117dp"
        android:layout_marginBottom="207dp"
        android:ellipsize="marquee"
        android:marqueeRepeatLimit="marquee_forever"
        android:scrollHorizontally="true"
        android:singleLine="true"
        android:text="Demonstration of Asynchronous Task !!!!"
        android:textSize="20sp"
        android:textStyle="bold"
        android:visibility="invisible" />
</RelativeLayout>

```

JAVA-CODE:

```

import androidx.appcompat.app.AppCompatActivity;import
android.os.AsyncTask;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;
public class MainActivity extends AppCompatActivity {
    TextView marqtxt;
    Button btnstart, btnstop;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        marqtxt = (TextView) findViewById(R.id.marqueeText); btnstart
        = (Button) findViewById(R.id.buttonstart); btnstop = (Button)
        findViewById(R.id.buttonstop); btnstart.setOnClickListener(new
        View.OnClickListener() {
            @Override

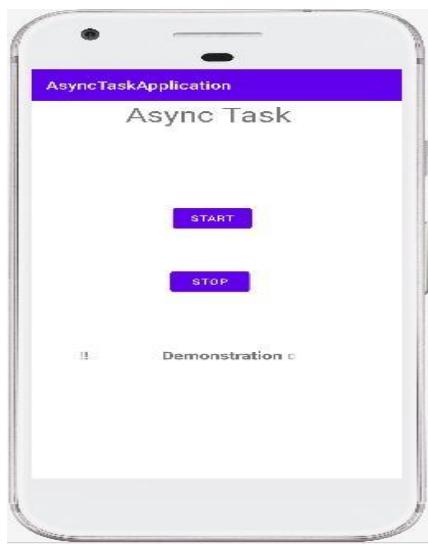
```

```

public void onClick(View v) {
    ExampleAsyncTask task = new ExampleAsyncTask();
    task.execute();
}
});
btnstop.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        marqtxt.setSelected(false);
        marqtxt.setVisibility(View.INVISIBLE);
    }
});
}
private class ExampleAsyncTask extends AsyncTask<String, String, String>{
    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        Toast.makeText(getApplicationContext(),"Async Task
Started!!!!!!",Toast.LENGTH_SHORT).show();
    }
    @Override
    protected String doInBackground(String... strings) {try {
        Thread.sleep(250);
    }
    catch (InterruptedException e){
        e.printStackTrace();
    }
    return null;
}
    @Override
    protected void onPostExecute(String s) {
        super.onPostExecute(s);
        marqtxt.setVisibility(View.VISIBLE);
        marqtxt.setSelected(true);
    }
}
}

```

OUTPUT:



**Program 11: (PART-B): 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.**

XML-CODE:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent" android:layout_height="match_parent"
    tools:context=".MainActivity">
    <TextView android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="152dp"
        android:layout_marginBottom="564dp"
        android:text="ClipBoard"
        android:textSize="36sp" />
    <EditText
        android:id="@+id/editText1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="127dp"
        android:layout_marginBottom="496dp"
        android:ems="10"
        android:hint="Enter the text here"
        android:inputType="textPersonName"
        android:text="" />
    <EditText
        android:id="@+id/editText2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="122dp"
```

```

        android:layout_marginBottom="411dp"
        android:ems="10" android:hint="Copied
        Text"
        android:inputType="textPersonName"
        android:text="" />
<Button
        android:id="@+id/copy"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="279dp"
        android:onClick="copy"
        android:layout_marginBottom="312dp"
        android:text="Copy" />
<Button
        android:id="@+id/paste"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="92dp"
        android:onClick="paste"
        android:layout_marginBottom="313dp"
        android:text="Paste" />
</RelativeLayout>

```

## JAVA-CODE:

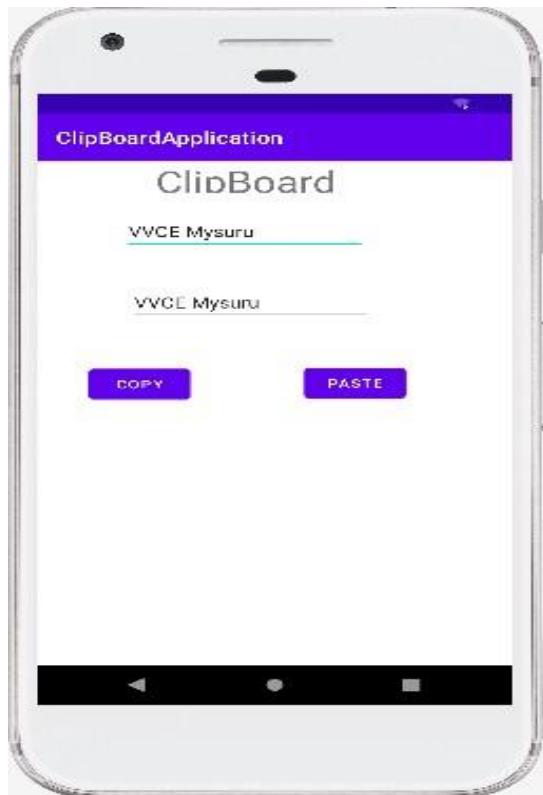
```

import androidx.appcompat.app.AppCompatActivity;import
android.content.ClipData;
import android.content.ClipboardManager;
import android.os.Bundle;
import android.view.View; import
android.widget.EditText;
public class MainActivity extends AppCompatActivity {
    ClipboardManager cbm;
    ClipData cd;
    EditText e1,e2;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

```

```
e1 = (EditText)findViewById(R.id.editText1);e2 =  
    (EditText)findViewById(R.id.editText2);  
    cbm =(ClipboardManager) getSystemService(CLIPBOARD_SERVICE);  
}  
public void copy(View V){  
    String text = e1.getText().toString();  
    cd = ClipData.newPlainText("text",text);  
    cbm.setPrimaryClip(cd);  
}  
public void paste(View V){  
    ClipData cd2 = cbm.getPrimaryClip();  
    ClipData.Item item = cd2.getItemAt(0); String  
    copied = item.getText().toString();  
    e2.setText(copied);  
}  
}  
}
```

OUTPUT:



### Extra Sample programs

## 1. Sample Intent program

### XML 1 Code

```

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

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="91dp"
        android:layout_marginBottom="631dp"
        android:text="FIRST ACTIVITY"
        android:textSize="36sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/button"
        android:onClick="second"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="163dp"
        android:layout_marginBottom="340dp"
        android:text="SECOND" />

</RelativeLayout>

```

### Java Code 1

```

package com.example.intentapplication;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
    public void second(View V){
        Intent i = new Intent(getApplicationContext(),MainActivity2.class);
        startActivity(i);
    }
}

```

XML code

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity2">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="32dp"
        android:layout_marginBottom="535dp"
        android:text="SECOND ACTIVITY"
        android:textSize="36sp" />

    <Button
        android:id="@+id/button2"
        android:onClick="first"
        android:layout_width="140dp"
        android:layout_height="63dp"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="135dp"
        android:layout_marginBottom="319dp"
        android:text="FIRST"
        android:textSize="12sp" />
</RelativeLayout>
```

Java Code 2

```
package com.example.intentapplication;

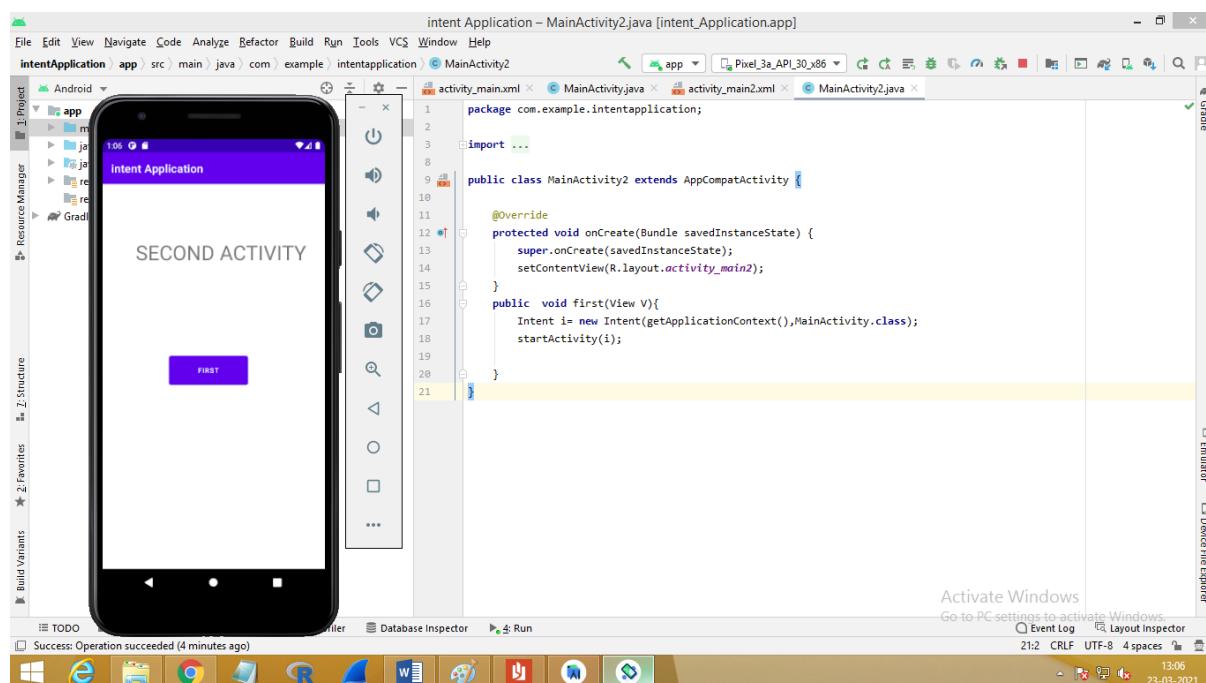
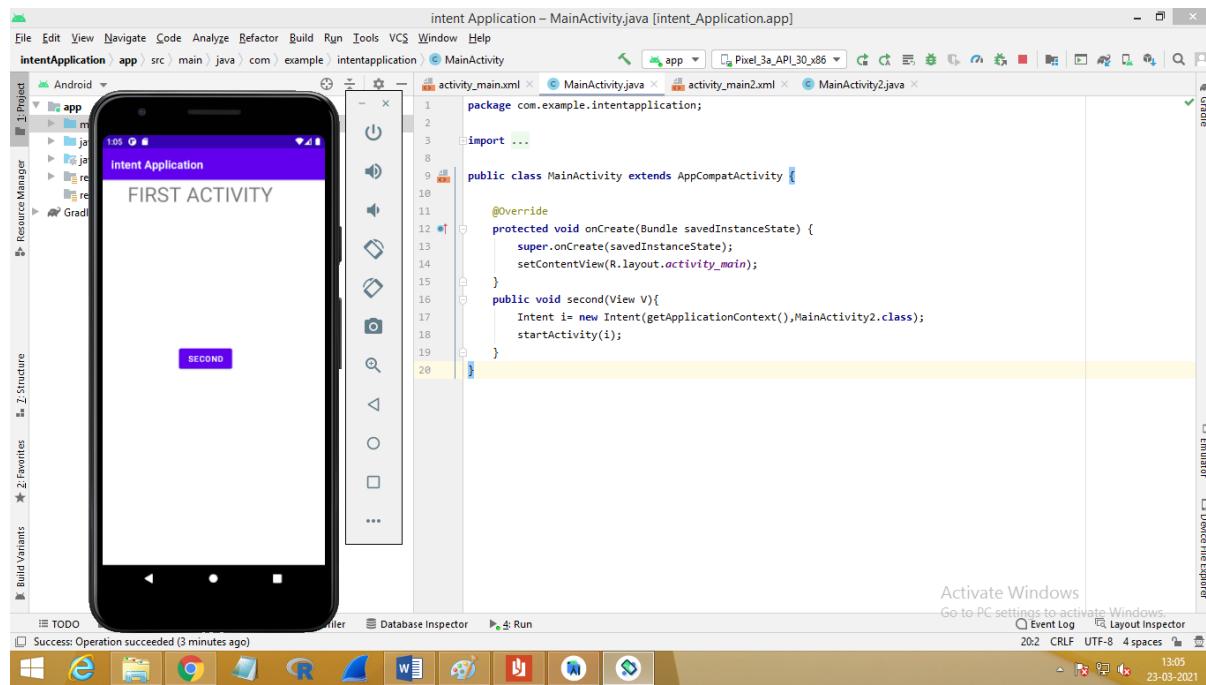
import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;

public class MainActivity2 extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main2);
    }
    public void first(View V){
        Intent i= new Intent(getApplicationContext(),MainActivity.class);
        startActivity(i);
    }
}
```

Output Screen



## **MOBILE APPLICATION DEVELOPMENT**

### **COURSE CODE:18CSMP68**

#### **Sample Viva Questions**

1. What is an Android?
2. Illustrate the Android lifecycle Activity.
3. What Is an Android SDK?
4. What is Android "compatibility"?
5. What are the key components Android Architecture?
6. Describe the Android Framework.
7. What are the data types supported by AIDL?
8. What is Gradle Framework ?
9. Why do we need AVD?
10. How do you add gradle dependencies ?
11. What is the difference between Mobile Application Testing and Mobile Testing?
12. What are the different data storage options available on the Android platform?
13. The list of data storage options on the Android platform.
14. Describe Activities.
15. What are Intents? What are the types of Intents?
16. What is Application class?
17. What is a View?
18. What is a view Group?
19. What are these UI components that we can use in our application?
20. Define Constraint Layout.
21. Why did we need Constraint Layout?
22. What are the different types of Android widgets?
23. Which are the files that demonstrate implementing and using of the custom widget?
24. Enumerate the three key loops while monitoring an activity?
25. What are the major steps involved in creating a bounded service through AIDL?