RGM COLLEGE OF ENGINEERING AND TECHNOLOGY

(AUTONOMOUS ESTD.1995)

Accredited by NAAC of UGC, New Delhi with 'A' Grade

Nandyal-518501, Kurnool (Dist), AP.

LABORATORY CERTIFICATE

This is certify that Mr. /Miss_____

Regd.No_____of____year_____has successfully completed the experiments in _____lab of the _____branch prescribed by the RGMCET (Autonomous).Nandyal.For the academic year_____.

Signature of the Staff Members

Date: Signature of the HOD

Signature of the Internal Examiner Signature of the External Examiner

RGM COLLEGE OF ENGINEERING AND TECHNOLOGY

(AUTONOMOUS ESTD.1995)

Accredited by NAAC of UGC, New Delhi with 'A' Grade

Nandyal-518501, Kurnool (Dist), AP.

ANDROID PROGRAMMING LAB RECORD

RGM COLLEGE OF ENGINEERING AND TECHNOLOGY

(AUTONOMOUS ESTD.1995)

Accredited by NAAC of UGC, New Delhi with 'A' Grade

Nandyal-518501, Kurnool (Dist), AP.

INDEX

12		Write an android program to develop Media player application			
13		a) Write an android program to develop Video view application b) Write an android program to develop Audio Recording application			
14		a) Write an android program to develop Video Recording application. b) Write an android program to develop Camera and Gallery application.			
15		a) Create an android application to get latitude and longitude value by using Location Service. b) Create an android application to display X, Y Sensor values by using Sensor Service.			
16		a) Create an android application to get the notifications on Notification Bar by Using Notification Service. b) Create an android application to display available Wi-Fi devices and Paired Wi-Fi devices by using Wi Fi Service			
17		a) Create an android application to get the Bluetooth devices and list of devices using Bluetooth and Vibrator Service. b) Create an android application to get the System Announcements by using Broadcast Receiver.			
18		Create an android application to share the data between multiple applications by using Content Provider			
19		Create an android application to display different Dialog Boxes.			
20		Create an android application to display current location on Google maps by using Google-Maps Service			

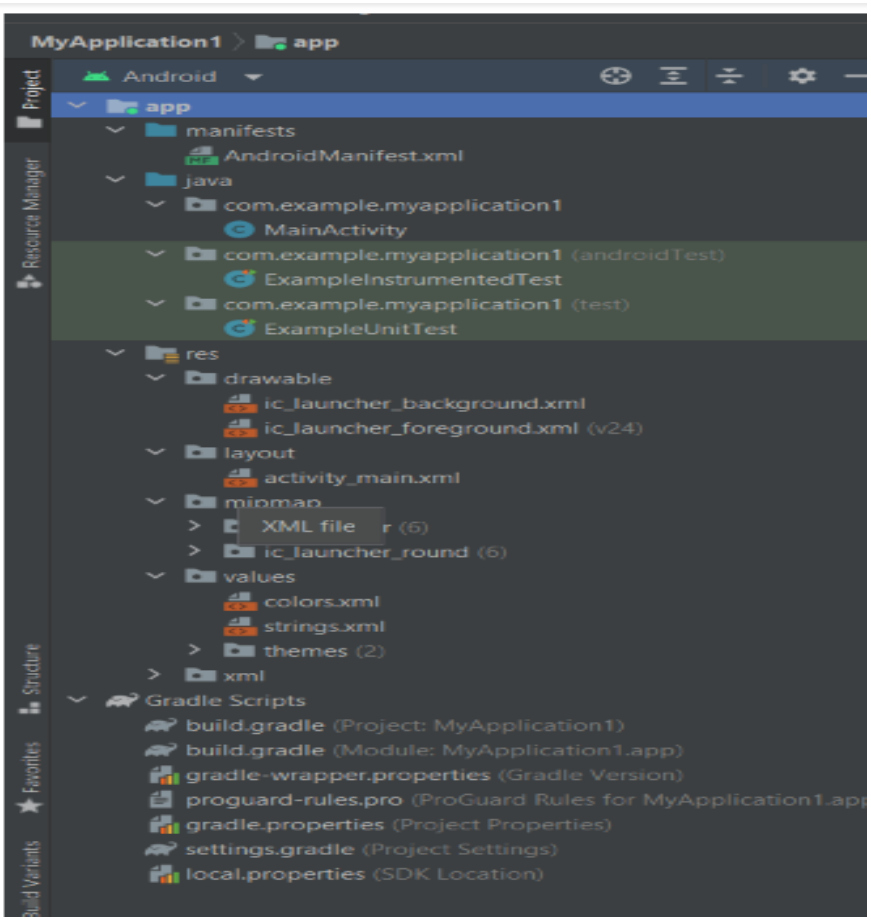
1 System Requirements to install Android Studio

- The following are the system requirements for Android Studio on Windows.
  - 64-bit Microsoft® Windows® 8/10/11
  - x86_64 CPU architecture; 2nd generation Intel Core or newer, or AMD CPU with support for a Windows Hypervisor
  - 8 GB RAM or more
  - 8 GB of available disk space minimum (IDE + Android SDK + Android Emulator)
  - 1280 x 800 minimum screen resolution

Project Structure in Android Application:

Android Project Folder Structure

- JetBrains community developed Android Studio, the official IDE (Integrated Development Environment) for Android app development and is freely distributed by Google.
- After the complete Android Architecture setup, we can construct an Android application in the studio. For each sample application, we should establish a new project and learn the folder structure.
- After setting up an Android development environment in Android Studio, we can create an example application, and our project folder structure will look like this.



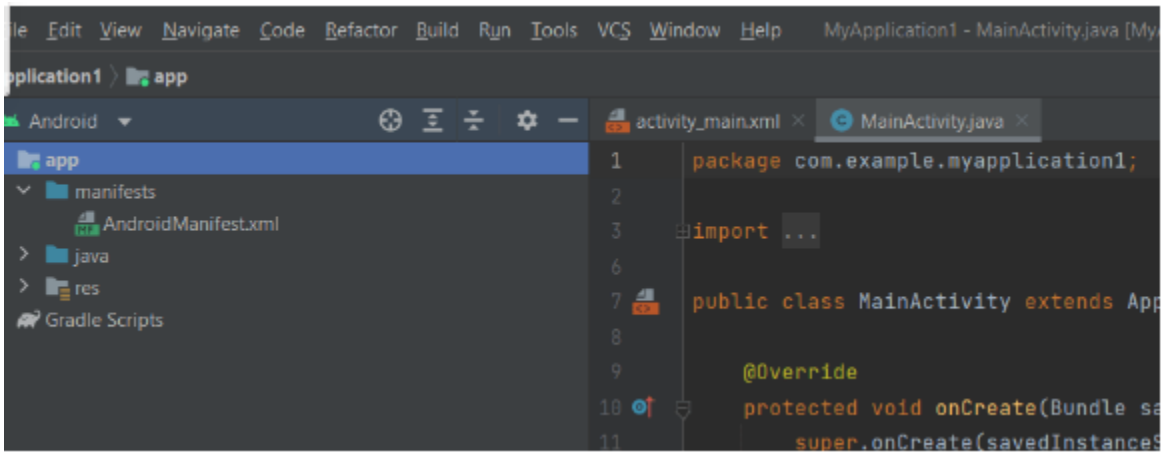
- The structure of the Android project on disk can differ from the above picture. Select Project from the Project selection instead of Android to see the Project's actual file structure.
- Many app modules, source code files, and resource files are included in the Android project. We'll go through all of the Android app's directories and files.

- Manifests Folder
- Java Folder
- res (Resources) Folder
- Drawable Folder
- Layout Folder
- Mipmap Folder
- Values Folder
- Gradle Scripts

Manifests Folder

- This folder includes the AndroidManifest.xml, which we will use to create the Android application. This file holds information about our programs, such as the Android **versions, metadata**, the **states packages** for the java code, and other app components. **It functions as a bridge between the Android operating system and our application.**

Let's see the manifests folder structure of the Android application shown below.



AndroidManifest.xml

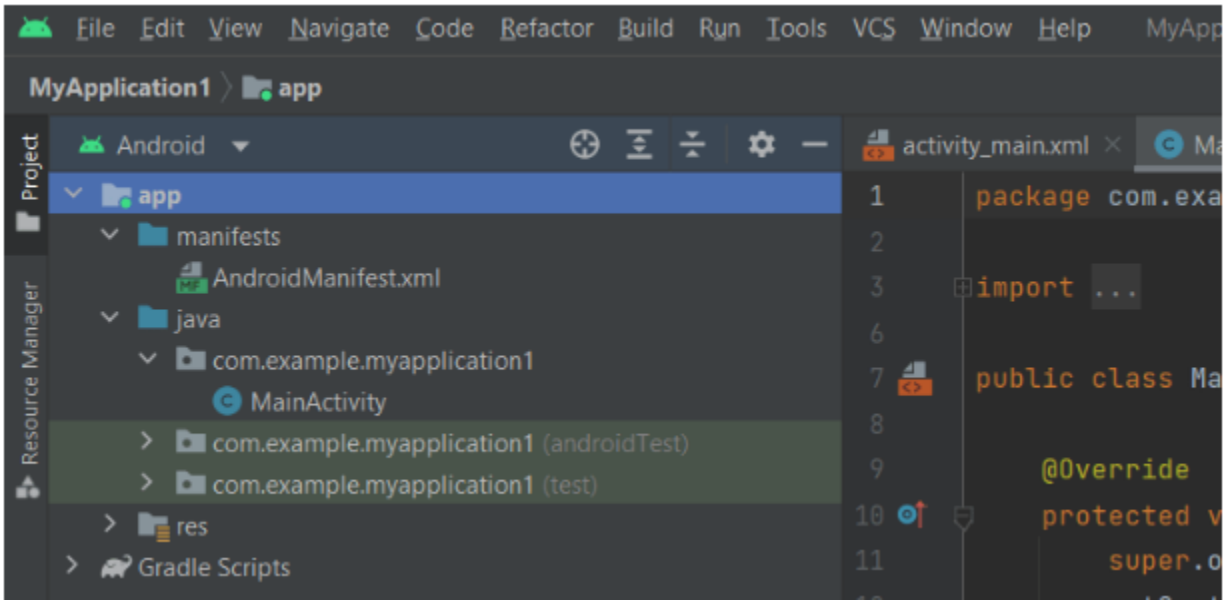
XML code:

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

   < 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 >
```

Java folder

- The Java folder holds all of the java and Kotlin source code (.java) files created during app development, as well as additional Test files. When we begin a fresh Java project, the class file MainActivity.java is automatically produced under the package name "com.example.myapplication1" as seen here.



Java Code

```
package com.example.myapplication1;
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
public class MainActivity extends AppCompatActivity {
  @Override
  protected void onCreate ( Bundle savedInstanceState )
  {
    super.onCreate ( savedInstanceState ) ;
    setContentView ( R.layout.activity_main) ;
  }
}
```

Kotlin Code

```
package com.example.myapplication1
 import androidx.appcompat.app.AppCompatActivity import android.os.Bundle
 class MainActivity : AppCompatActivity ( ) {
 override fun onCreate ( savedInstanceState : Bundle ? )
{
super.onCreate ( savedInstanceState )
        setContentView ( R.layout.activity_main )
 }
}
```

Folder res/mipmap

- This section includes launcher.xml files that specify the icons on the home screen. It has several icon densities based on the device size, such as hdpi, mdpi, and xhdpi.

Folder res/values

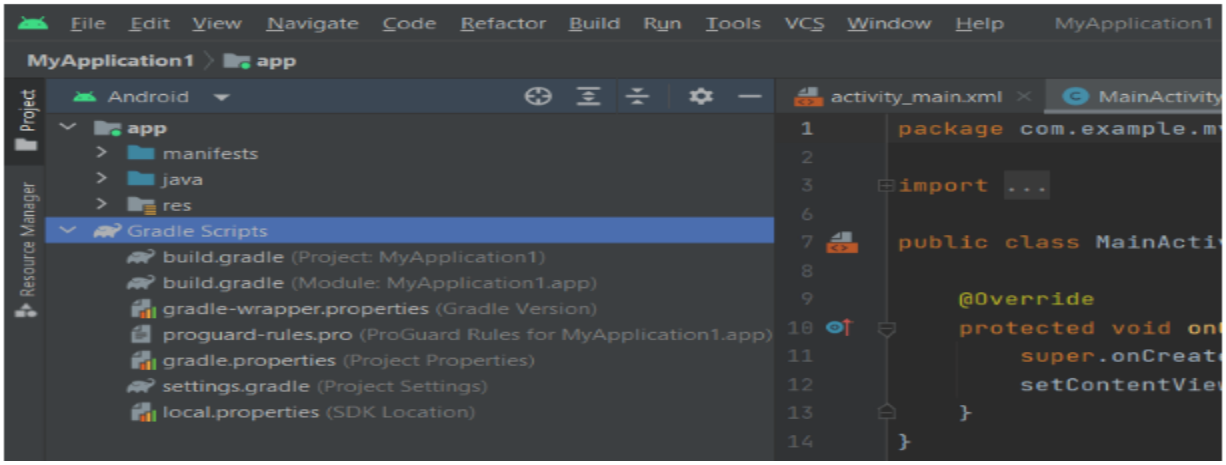
- The Values folder contains various XML files such as strings, dimensions, colors, and style definitions. The strings.xml file, which includes the resources, is one of the most crucial.

XML Code

```
< resources >
   < string name = "app_name"> NameOfTheApplication </string>
   < string name = "checked" > Checked </ string >
   < string name = "unchecked" > Unchecked </ string >
</ resources >
```

Gradle Scripts directory

- Gradle is an automated build system that includes several files that create a build configuration. Buildscripts are used in the build.gradle (Project), and plugins and implementations are used in the build.gradle (Module) to make settings that may be used by all of our application modules.



EXPERIMENTS

Experminet 2A: Create android application to display RGMCET Text Message

Step By Step Process:

Activity_main.xml

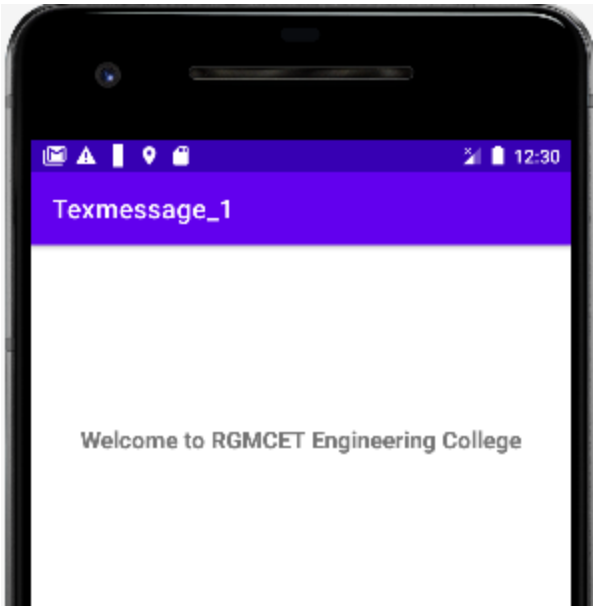
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="16dp"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/tv1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="120dp"
        android:textSize="18sp"
        android:textStyle="bold"/>

</RelativeLayout>
```

MainActivity.java

==========

```
package com.example.texmessage_1;

import androidx.appcompat.app.AppCompatActivity;

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

public class MainActivity extends AppCompatActivity {
    TextView txt1;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        txt1=(TextView)findViewById(R.id.tv1);
        txt1.setText("Welcome to RGMCET Engineering College");

    }
}
```

Output:



2B: Create an android application to display RGMCET Message by using Button

Step By Step Process:

Activity_main.xml:

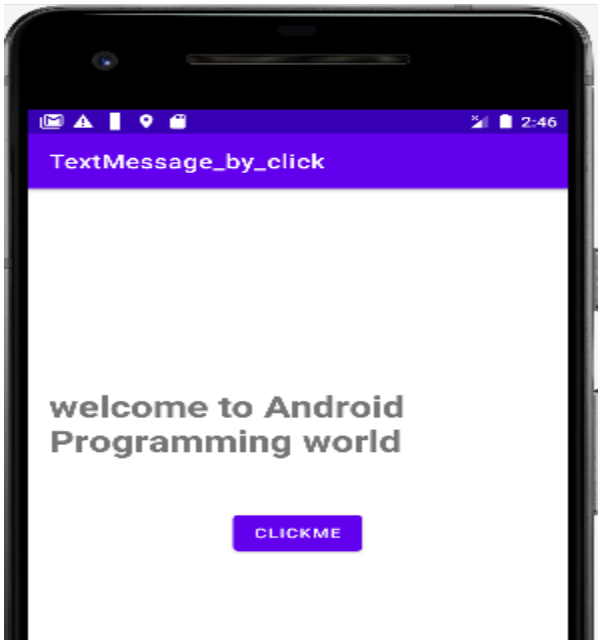
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"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

    <TextView
        android:id="@+id/tv1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:paddingTop="200dp"
        android:paddingBottom="50dp"
        android:paddingRight="16dp"
        android:paddingLeft="16dp"
        android:textStyle="bold"
        android:textSize="30sp"/>
    <Button
        android:id="@+id/btn1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/tv1"
        android:text="ClickMe"
        android:layout_centerHorizontal="true"/>



</RelativeLayout>
```

MainActivity.java

==========

```
package com.example.textmessage_by_click;

import androidx.appcompat.app.AppCompatActivity;

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

public class MainActivity extends AppCompatActivity {
    TextView textView;
    Button button;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        textView=(TextView) findViewById(R.id.tv1);
        button=(Button) findViewById(R.id.btn1);
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                textView.setText("welcome to Android Programming world");
            }
        });
    }
}
```

Output:



Experminet 3A: Create an android application to call different activities by using Implicit and Explicit Intents.

Step By Step Process:

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"
    android:padding="20dp"
    tools:context=".MainActivity">
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter Your Url Here"
        android:id="@+id/etv1"
        android:textStyle="bold"
        android:textColor="#881414"/>
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/btn1"
        android:text="Click Me"/>

</LinearLayout>
```

MainActivity.java:

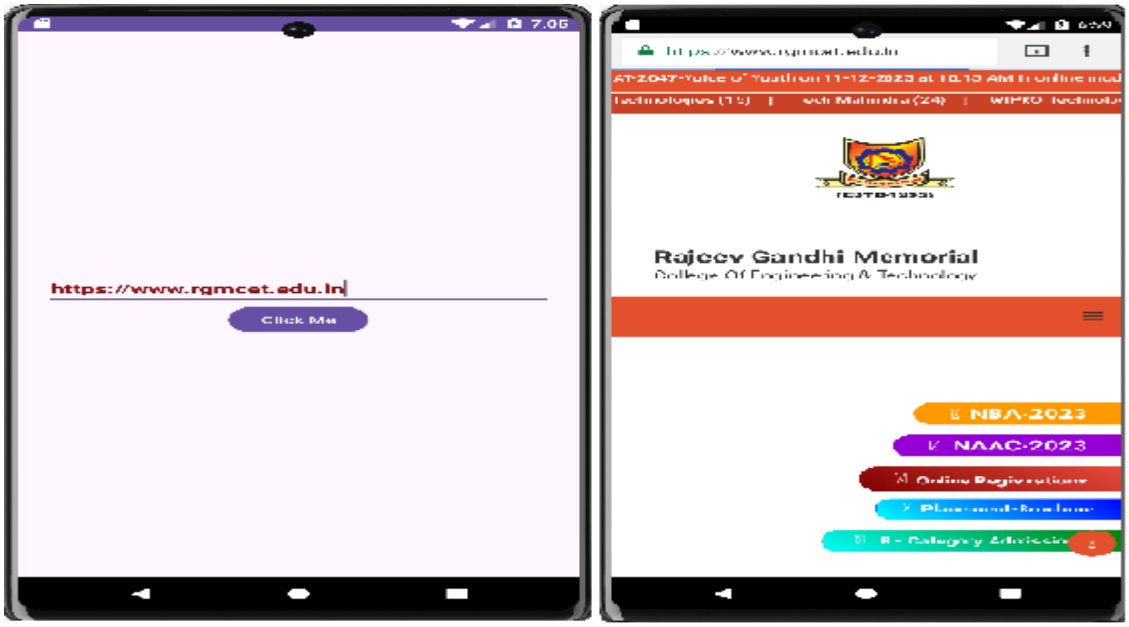
```
package com.example.relativelayoutexample;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

public class MainActivity extends AppCompatActivity {
    EditText etv1;
    Button btn1;
```

```
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        etv1=(EditText) findViewById(R.id.etv1);
        btn1=(Button) findViewById(R.id.btn1);
        btn1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String url=etv1.getText().toString();
                Intent intent=new Intent(Intent.ACTION_VIEW, Uri.parse(url));
                startActivity(intent);
            }
        });
    }
}
```

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">
    <uses-permission android:name="android.permission.INTERNET"/>
    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.RelativeLayoutExample"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

</manifest>
```

Output:


https://www.rgmcet.edu.in
Click Me
Rajeev Gandhi Memorial
NBA-2023
NAAC-2023

Experminet 3B: Create an android application using explicit Intent

Step By Step Process:

Android_main_activityone.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=".MainActivityOne">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/tv1"
        android:text="First Activity"
        android:textSize="30dp"
        android:textColor="#59ADD3"
        android:textStyle="bold"/>
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/btn1"
        android:text="Second Activity"/>

</LinearLayout>
```

Android_main_activitytwo.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=".MainActivityTwo">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/tv1"
        android:text="Second Activity"
        android:textSize="30dp"
        android:textColor="#59ADD3"
        android:textStyle="bold"/>
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/btn2"
        android:text="First Activity"/>

</LinearLayout>
```

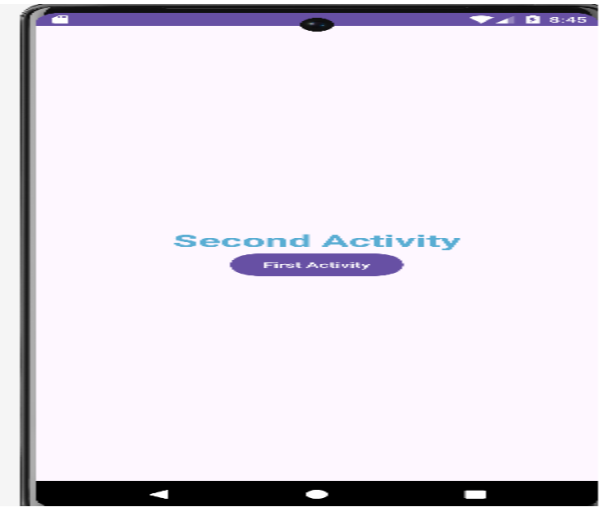
MainActivityone.java

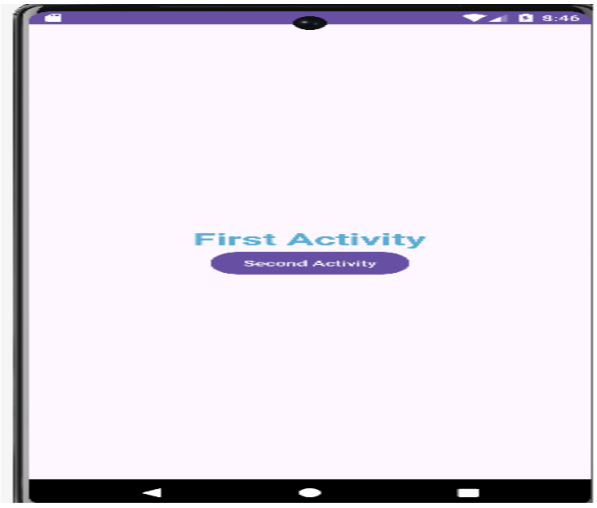
```
package com.example.explicitintent;

import androidx.appcompat.app.AppCompatActivity;

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

public class MainActivityOne extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main_one);
        Button button=(Button) findViewById(R.id.btn1);
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent=new Intent(getApplicationContext(),
MainActivityTwo.class);
                startActivity(intent);
            }
        });
    }
}
```

MaintActivitytwo.java

```
package com.example.explicitintent;

import androidx.appcompat.app.AppCompatActivity;

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

public class MainActivityTwo extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main_two);
        Button button=(Button) findViewById(R.id.btn2);
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent=new Intent(getApplicationContext(),
MainActivityOne.class);
                startActivity(intent);
            }
        });
    }
}
```

Ouput:


8:45
Second Activity
First Activity


8:46
First Activity
Second Activity

Experminet 4A: Create an android application to select item from given list by using AutoCompleteTextView (ACTV)

Activity_main.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"
    android:paddingBottom="16dp"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    android:id="@+id/activity_main"
    tools:context=".MainActivity">

    <AutoCompleteTextView
        android:id="@+id/actv1"
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="83dp"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:hint="Color" />
    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/image"
        android:padding="5dp"
        android:layout_alignTop="@id/actv1"
        android:layout_alignBottom="@id/actv1"
        android:layout_alignRight="@id/actv1"
        android:src="@drawable/arrow"/>

    <AutoCompleteTextView
        android:id="@+id/actv2"
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="150dp"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:hint="Item" />
    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/image1"
        android:padding="5dp"
```

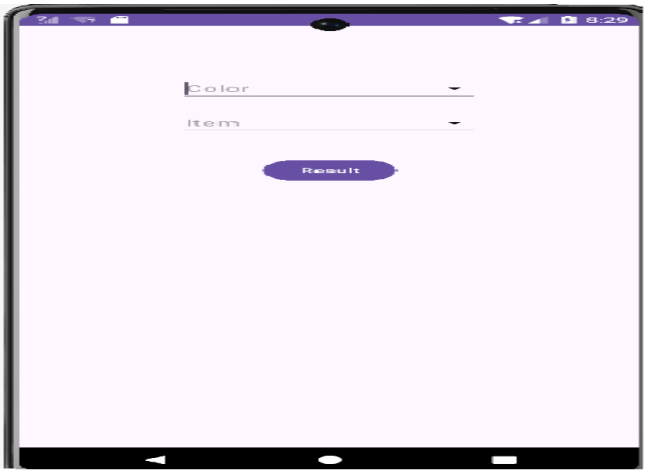
```
        android:layout_alignTop="@id/actv2"
        android:layout_alignBottom="@id/actv2"
        android:layout_alignRight="@id/actv2"
        android:src="@drawable/arrow"/>
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/button"
        android:text="Result"
        android:layout_below="@id/actv2"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="46dp"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="80dp"
        android:layout_below="@id/button"
        android:layout_centerHorizontal="true"
        android:id="@+id/tv"/>

</RelativeLayout>
```

MainActivity.java

```
package com.example.actv;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.AutoCompleteTextView;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        final AutoCompleteTextView actv1=(AutoCompleteTextView)
findViewById(R.id.actv1);
        final AutoCompleteTextView actv2=(AutoCompleteTextView)
findViewById(R.id.actv2);
        Button button=(Button) findViewById(R.id.button);
        final TextView tv=(TextView) findViewById(R.id.tv);
        ImageView image=(ImageView) findViewById(R.id.image);
        ImageView image1=(ImageView) findViewById(R.id.image1);
        actv1.setThreshold(2);
```

```
        actv2.setThreshold(2);
        ArrayAdapter<String>adapter=new ArrayAdapter<String>(this,
android.R.layout.simple_dropdown_item_1line,colors);
        actv1.setAdapter(adapter);

        ArrayAdapter<String>adapter1=new ArrayAdapter<String>(this,
android.R.layout.simple_dropdown_item_1line,item);
        actv2.setAdapter(adapter1);

        image.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                actv1.showDropDown();
            }
        });
        image1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                actv2.showDropDown();
            }
        });
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String s=actv1.getText().toString();
                String s1=actv2.getText().toString();
                tv.setText(s +' '+s1);
            }
        });
    }
    private static final String[] colors=new String[]{"red","blue"};
    private static final String[] item=new String[]{"car","bike"};

}
```

Ouput:



Experiment 4B:- Create an android application to display dropdown menu items and pick one item by using Spinner Component

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"
    android:padding="20dp"
    tools:context=".MainActivity">
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="My Form"
        android:textStyle="bold"
        android:textColor="#E91E63"
        android:textSize="30dp"
        android:gravity="center"/>

    <EditText
        android:id="@+id/et_name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter Name"/>
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/et_password"
        android:hint="Enter Password"
        android:inputType="textPassword"/>
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/et_phone"
        android:hint="Enter Phone No"
        android:inputType="phone"/>
    <RadioGroup
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <RadioButton
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Male"
            android:id="@+id/rb_male"
            android:textStyle="bold"
            android:textColor="#1637ED"
```

```
            android:textSize="20dp"
            android:checked="false"/>
        <RadioButton
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Female"
            android:id="@+id/rb_female"
            android:textStyle="bold"
            android:textColor="#1637ED"
            android:textSize="20dp"/>
    </RadioGroup>
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Select Your Interested Language"
        android:textStyle="italic"
        android:textColor="#E91E63"
        android:textSize="15dp"/>
    <CheckBox
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/cb_android"
        android:text="Android"
        android:textSize="20dp"
        android:textColor="#1637ED"
        android:textStyle="bold"/>
    <CheckBox
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/cb_Java"
        android:text="Java"
        android:textSize="20dp"
        android:textColor="#1637ED"
        android:textStyle="bold"/>
    <CheckBox
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/cb_python"
        android:text="Python"
        android:textSize="20dp"
        android:textColor="#1637ED"
        android:textStyle="bold"/>
    <Spinner
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/sp_cities"
        android:entries="@array/citites"
        android:padding="20dp"/>
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Click Me"
```

```
        android:onClick="getAllValues"/>
</LinearLayout>
```

Create one resource file for spinner content in the path of: res==>values==>string.xml

```
<resources>
    <string name="app_name">Spinner</string>
    <string-array name="spinnerlist">
        <item>Cities</item>
        <item>Nandyal</item>
        <item>Hyderabad</item>
        <item>Bang</item>
        <item>Kurnool</item>
        <item>Mumbai</item>
    </string-array>
</resources>
```

MainActivity.java

```
package com.example.spinner;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.view.View;
import android.widget.CheckBox;
import android.widget.EditText;
import android.widget.RadioButton;
import android.widget.Spinner;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    EditText et_name,et_password,et_phone;
    RadioButton rb_male,rb_female;
    CheckBox cb_Android,cb_java,cb_python;
    Spinner sp_cities;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        et_name=(EditText) findViewById(R.id.et_name);
        et_password=(EditText) findViewById(R.id.et_password);
        et_phone=(EditText) findViewById(R.id.et_phone);
        rb_male=(RadioButton) findViewById(R.id.rb_male);
        rb_female=(RadioButton) findViewById(R.id.rb_female);
        cb_Android=(CheckBox) findViewById(R.id.cb_android);
        cb_java=(CheckBox) findViewById(R.id.cb_Java);
        cb_python=(CheckBox) findViewById(R.id.cb_python);
        sp_cities=(Spinner) findViewById(R.id.sp_cities);
    }
    public void getAllValues(View view){
```

```
                // Get Values from EditText
                String name=et_name.getText().toString();
                String password= et_password.getText().toString();
                String phoneNo=et_phone.getText().toString();
                // Get Values from radio button
                String gender="";
                if(rb_male.isChecked()){
                    gender=rb_male.getText().toString();
                }
                if(rb_female.isChecked()){
                    gender=rb_female.getText().toString();
                }
                //get values from check box
                String interest="";
                if(cb_Android.isChecked()){
                    interest+=cb_Android.getText().toString()+"\n";
                }
                if(cb_java.isChecked()){
                    interest+=cb_java.getText().toString()+"\n";
                }
                if(cb_python.isChecked()){
                    interest+=cb_python.getText().toString()+"\n";
                }
                //get values from spinner
                String city=sp_cities.getSelectedItem().toString();

                // ----------Print All values in Toast-----
                String
        res=name+"\n"+phoneNo+"\n"+password+"\n"+gender+"\n"+interest+"\n"+city;
                Toast.makeText(this,res,Toast.LENGTH_SHORT).show();

            }
        }
```

Ouput:

Practical 5A: Create an android application to display internal storage data using Array Adapter.

Step 1: Create a new project and name it Simpl ArrayAdapterExample.

Open Android Studio -> Select File -> New -> New Project. Fill the forms and click "Finish" button.

Step 2: Now open app -> res -> layout -> xml (or) activity_main.xml and add following code :

```
<?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"
    tools:context=".AdapterDemo">
    <ListView
        android:id="@+id/listview_one"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:divider="@color/black"
        android:dividerHeight="2dp"/>

</LinearLayout>
```

Step 3: Create a new ui_view_one.xml and add the below code(right click on layout ---->new ---->select layout resource file name it as ui_view_one.xml)

Note: this layout is created to provide a Textview for array list which we created in .java file

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="20dp"
    android:background="#F4DF22">

    <TextView
        android:id="@+id/textview_one"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="ramesh"
        android:textSize="20dp"
        android:textStyle="bold"
        android:textColor="@color/black"/>

</LinearLayout>
```

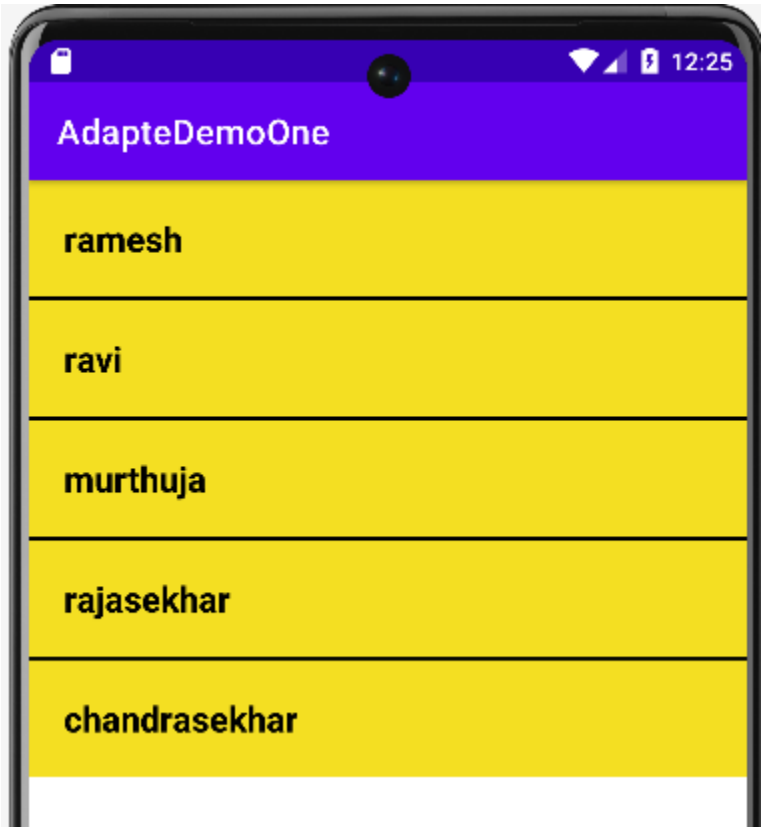
Step 4: Now Open app -> java -> package -> MainActivity.java and add the below code. Here we will use ArrayAdapter to display the items in Listview.

```
package com.example.adaptedemoone;

import androidx.appcompat.app.AppCompatActivity;
```

```
import android.os.Bundle;
import android.widget.ArrayAdapter;
import android.widget.ListView;

public class AdapterDemo extends AppCompatActivity {
    String[] name_arr={"ramesh","ravi","murthuja","rajasekhar","chandrasekhar"};
    ListView listview_one;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_adapter_demo);
        listview_one=(ListView) findViewById(R.id.listview_one);
        ArrayAdapter<String>arrayAdapter=new
ArrayAdapter<String>(this,R.layout.ui_view_one,R.id.textview_one,name_arr);
        listview_one.setAdapter(arrayAdapter);
    }
}
```

Output:

Now run the App in Emulator and you will see the below output:



Practical 5B: Create an android application to display internal storage data in vertical format by using Custom Adapter.

Step1:

- **First Create New Project with name cutomAdapter and under activity_customadapter.xml add below code to create on listview**

Activity_customadapter.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"
    tools:context="com.example.adapterdemofour.adapterfour">

    <ListView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:divider="#000"
        android:dividerHeight="2dp"
        android:id="@+id/listview_four"/>

</LinearLayout>
```

Step2:

- **Now Create one custom view for your custom array list by right click on layout ---->new ---->Layout resource file---->Name it as : ui_view.custome.xml and add below code:**

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal"
    android:padding="10dp">

    <ImageView
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:id="@+id/imageview_four"
        android:src="@drawable/default_icon"/>
      //here the default_icon is taken from drawable folder by using vector
asset(right click on drawable folder---->new---->vector asset--->clock icon(here
will display many icons select any one icon and name it and finish

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Animal"
        android:layout_marginLeft="10dp"
        android:textSize="20dp"
        android:textStyle="bold"
        android:id="@+id/textview_four"/>
```

```
</LinearLayout>
```

Note: this above .xml code is used to view Images along with Name of the image on the screen so here we have taken two views those are: ImageView,TextView

Step3:

- **Now download animals images and paste it into drawable folder by right click on drawable and click on paste then all images will be under this folder**

Step4:

- **Now create one Animal.java class(right click package select class and name it as Animal.java and click on finish then add below code) to return animal_name & animal_image on the view and this class can be used to create custom list in main java that is customAdapter.java**

```
package com.example.adapterdemofour;

public class Animal {
    String animal_name;
    int animal_image;
    public Animal(int animal_image,String animal_name){

        this.animal_image=animal_image;
        this.animal_name=animal_name;
    }
    public int getAnimalImage(){
        return animal_image;
    }
    public String getAnimalName(){
        return animal_name;
    }
}
```

Step5:

- **Now write the code on MainActivity that is customAdapter.java to implement animal_name& animal_image which mentioned in Animal.java**

```
package com.example.adapterdemofour;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.widget.ListView;

import com.example.adaptedemoone.R;

import java.util.ArrayList;

public class adapterfour extends AppCompatActivity {
    ListView listview_four;
    ArrayList animals_list;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_adapterfour);
        listview_four=(ListView) findViewById(R.id.listview_four);
       //This custom arraylist-------------------------------------------
```

```
            animals_list=new ArrayList();
            animals_list.add(new Animal(R.drawable.image1,"bear"));
            animals_list.add(new Animal(R.drawable.image10,"Qiwi"));
            animals_list.add(new Animal(R.drawable.image11,"parrot"));
            animals_list.add(new Animal(R.drawable.image2,"pickok"));
            animals_list.add(new Animal(R.drawable.image3,"Dove"));
            animals_list.add(new Animal(R.drawable.image9,"cat"));
            //--------------------XXXXXXXXXXXX-------------------------------------
           // this belwo code can add after creating myadapter.java
            myadpterfour myadpterfour=new
myadpterfour(this,R.layout.ui_view_four,animals_list);
            listview_four.setAdapter(myadpterfour);
        }
    }
```

Step6:

- **Now create one adapter java class under the package and add below code:**

```
package com.example.adapterdemofour;

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.ImageView;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;

import com.example.adaptedemoone.R;

import java.util.ArrayList;

public class myadpterfour extends ArrayAdapter<Animal>
{
    ArrayList<Animal> animal_list;
    LayoutInflater layoutInflater;
    public myadpterfour(Context context, int resource, ArrayList<Animal>
objects) {
        super(context, resource, objects);
        animal_list=objects;
        layoutInflater=LayoutInflater.from(context);
    }

    @Override
    public int getCount() {
        return animal_list.size();
    }

    @NonNull
    @Override
    public View getView(int position, @Nullable View convertView, @NonNull
ViewGroup parent) {
        View v=layoutInflater.inflate(R.layout.ui_view_four,null);
        ImageView imageView=(ImageView) v.findViewById(R.id.imageview_four);
        TextView textView=(TextView) v.findViewById(R.id.textview_four);
        imageView.setImageResource(animal_list.get(position).getAnimalImage());
        textView.setText(animal_list.get(position).getAnimalName());
        return v;
```

```
        }
    }
```

Output:

Practical 6: Create an android application to display WhatsApp videos in grid view by using Custom Adapter

Step1: create one new Adapter.java class in com.example.whatsappvideos and add the following code

```
package com.example.whatsappvideos;

import android.content.Context;
import android.content.Intent;
import android.os.Environment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import com.bumptech.glide.Glide;

import java.util.ArrayList;

public class Adapter extends RecyclerView.Adapter<Adapter.ViewHolder> {

    Context context;
    ArrayList<ModelClass> fileslist;

    public Adapter(Context context, ArrayList<ModelClass> fileslist) {
        this.context = context;
        this.fileslist = fileslist;
    }

    @NonNull
    @Override
    public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {

        View view=
LayoutInflater.from(context).inflate(R.layout.item_layout,null,false);
        return new ViewHolder(view);

    }

    @Override
    public void onBindViewHolder(@NonNull ViewHolder holder, int position) {

        final ModelClass modelClass=fileslist.get(position);
        if (modelClass.getUri().toString().endsWith(".mp4")){
            holder.play.setVisibility(View.VISIBLE);
        }else {
            holder.play.setVisibility(View.INVISIBLE);
        }
        Glide.with(context).load(modelClass.getUri()).into(holder.mainstatus);
        holder.mainstatus.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if (modelClass.getUri().toString().endsWith(".mp4")){
                    final String path=fileslist.get(position).getPath();
                    String destpath=
```

```
        Environment.getExternalStorageDirectory().getAbsolutePath()+Constant.SAVE_FOLDER_NAME
        ;
                            Intent intent=new Intent(context,Video.class);
                            intent.putExtra("DEST_PATH_VIDEO",destpath);
                            intent.putExtra("FILE_VIDEO",path);
                            intent.putExtra("FILENAME_VIDEO",modelClass.getFilename());
                            intent.putExtra("URI_VIDEO",modelClass.getUri().toString());
                            context.startActivity(intent);

                        }else{
                            final String path=fileslist.get(position).getPath();
                            String destpath=
        Environment.getExternalStorageDirectory().getAbsolutePath()+Constant.SAVE_FOLDER_NAME
        ;
                            Intent intent=new Intent(context,Picture.class);
                            intent.putExtra("DEST_PATH",destpath);
                            intent.putExtra("FILE",path);
                            intent.putExtra("FILENAME",modelClass.getFilename());
                            intent.putExtra("URI",modelClass.getUri().toString());
                            context.startActivity(intent);

                        }
                    }
                });

            }

            @Override
            public int getItemCount() {

                return fileslist.size();
            }

            public class ViewHolder extends RecyclerView.ViewHolder {

                ImageView mainstatus,play;
                public ViewHolder(@NonNull View itemView) {
                    super(itemView);
                    mainstatus=itemView.findViewById(R.id.thumnailofstatus);
                    play=itemView.findViewById(R.id.play);


                }
            }
        }
```

step2: create one more new class constant.java in com.example.whatsappvideos and add the following code

```
package com.example.whatsappvideos;

public class Constant {

    public static final String FOLDER_NAME="/WhatsApp/";
    public static final String SAVE_FOLDER_NAME="/StatusSaver/";

}
```

step3: create one more new class ModelClass.java in com.example.whatsappvideos and add the following code

```
package com.example.whatsappvideos;

import android.net.Uri;

public class ModelClass {
    String path,filename;
    Uri uri;

    public ModelClass(String path, String filename, Uri uri) {
        this.path = path;
        this.filename = filename;
        this.uri = uri;
    }

    public ModelClass() {

    }

    public String getPath() {
        return path;
    }

    public void setPath(String path) {
        this.path = path;
    }

    public String getFilename() {
        return filename;
    }

    public void setFilename(String filename) {
        this.filename = filename;
    }

    public Uri getUri() {
        return uri;
    }

    public void setUri(Uri uri) {
        this.uri = uri;
    }
}
```

step4: Implement the following code in MainActivity.java

```
package com.example.whatsappvideos;

import static android.os.Build.VERSION.SDK_INT;

import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.recyclerview.widget.RecyclerView;
import androidx.recyclerview.widget.StaggeredGridLayoutManager;
import androidx.swiperefreshlayout.widget.SwipeRefreshLayout;

import android.Manifest;
import android.content.pm.PackageManager;
import android.net.Uri;
import android.os.Bundle;
import android.os.Environment;
```

```
import android.os.Handler;
import android.widget.Toast;

import java.io.File;
import java.util.ArrayList;

public class MainActivity extends AppCompatActivity {
    int requestCode=1;
    Adapter adapter;
    File[] files;
    RecyclerView recyclerView;
    SwipeRefreshLayout refreshLayout;
    ArrayList<ModelClass> fileslist=new ArrayList<>();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        recyclerView=findViewById(R.id.recyclerview);
        refreshLayout=findViewById(R.id.swipe);
        checkPermission();
        //setuplayout();
        refreshLayout.setOnRefreshListener(new SwipeRefreshLayout.OnRefreshListener() {
            @Override
            public void onRefresh() {
                refreshLayout.setRefreshing(true);
                setuplayout();{
                    new Handler().postDelayed(new Runnable() {
                        @Override
                        public void run() {
                            refreshLayout.setRefreshing(false);

                        }
                    },1000);
                }
            }
        });

    }

    private void setuplayout() {
        fileslist.clear();
        recyclerView.setHasFixedSize(true);
        StaggeredGridLayoutManager staggeredGridLayoutManager=new
StaggeredGridLayoutManager(3,StaggeredGridLayoutManager.VERTICAL);
        recyclerView.setLayoutManager(staggeredGridLayoutManager);
        adapter=new Adapter(MainActivity.this,getData());
        recyclerView.setAdapter(adapter);
        adapter.notifyDataSetChanged();
    }

    private ArrayList<ModelClass> getData() {
        ModelClass f;
        String targetpath=
Environment.getExternalStorageDirectory().getAbsolutePath()+Constant.FOLDER_NAME+"Media/.Sta
tuses";
        File targetdir=new File(targetpath);
        files=targetdir.listFiles();
        for (int i=0;i<files.length;i++){
            File file=files[i];
            f=new ModelClass();
```

```
            f.setUri(Uri.fromFile(file));
            f.setPath(files[i].getAbsolutePath());
            f.setFilename(file.getName());
            if (!f.getUri().toString().endsWith("NoMedia")){
                fileslist.add(f);
            }
        }
        return fileslist;
    }

    private void checkPermission() {
        if (SDK_INT>23){
            if (checkSelfPermission(Manifest.permission.WRITE_EXTERNAL_STORAGE)==
PackageManager.PERMISSION_GRANTED){
                //MainCode
                setuplayout();

            }else {
                ActivityCompat.requestPermissions(MainActivity.this,new
String[]{Manifest.permission.WRITE_EXTERNAL_STORAGE},requestCode);
            }

        }else{
            Toast.makeText(getApplicationContext(),"Already",Toast.LENGTH_LONG).show();
            setuplayout();
        }
    }
}
```

step5: Create one more new class splash.java and implement the following code

```
            package com.example.whatsappvideos;

            import androidx.appcompat.app.AppCompatActivity;

            import android.content.Intent;
            import android.os.Bundle;
            import android.os.Handler;
            import android.view.WindowManager;

            public class splash extends AppCompatActivity {
                final static int SPLASH_TIMER=2000;

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

            getWindow().setFlags(WindowManager.LayoutParams.FLAGS_CHANGED,WindowManager.Layo
            utParams.FLAG_FULLSCREEN);



                    new Handler().postDelayed(new Runnable() {
                        @Override
                        public void run() {
                            Intent intent=new Intent(splash.this,MainActivity.class);
                            startActivity(intent);
                            finish();

                        }
```

```
                },SPLASH_TIMER);




            }
        }
```

step6:activity_splash.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"
    android:background="@color/white"
    tools:context=".splash">

    <ImageView
        android:layout_width="150dp"
        android:layout_height="150dp"
        android:src="@drawable/splashlogo"
        android:layout_centerInParent="true"/>

      <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:textSize="15sp"
        android:layout_centerHorizontal="true"
        android:layout_marginBottom="55dp"
        android:textStyle="bold"
        android:text="from"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_marginBottom="30dp"
        android:textSize="20sp"
        android:text="Tech Projects"
        android:layout_centerHorizontal="true"
        android:textStyle="bold"
        android:textColor="#77b8f8"/>


</RelativeLayout>
```

Step7:activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.swiperefreshlayout.widget.SwipeRefreshLayout
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:background="#FFE8E8"
    android:id="@+id/swipe"
    tools:context=".MainActivity">

    <androidx.recyclerview.widget.RecyclerView
```

```
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:id="@+id/recyclerview"
            android:fitsSystemWindows="true">

        </androidx.recyclerview.widget.RecyclerView>

    </androidx.swiperefreshlayout.widget.SwipeRefreshLayout>
```

Step8:item_layout.xml

```
    <?xml version="1.0" encoding="utf-8"?>
    <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        xmlns:app="http://schemas.android.com/apk/res-auto"
        android:backgroundTint="#FFE8E8"
        android:background="#FDF9F3"
        android:padding="5dp">

        <RelativeLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_margin="5dp">
            <androidx.cardview.widget.CardView
                android:layout_width="100dp"
                android:layout_height="100dp"
                android:backgroundTint="#7Fa1C3"
                app:cardCornerRadius="100dp">

                <androidx.cardview.widget.CardView
                    android:layout_width="90dp"
                    android:layout_height="90dp"
                    android:layout_gravity="center"
                    app:cardCornerRadius="90dp">

                    <ImageView
                        android:layout_width="90dp"
                        android:layout_height="90dp"
                        android:layout_gravity="center"
                        android:id="@+id/thumnailofstatus"
                        android:scaleType="centerCrop">
                    </ImageView>
                    <ImageView
                        android:layout_width="30dp"
                        android:layout_height="30dp"
                        android:src="@drawable/baseline_play_arrow_24"
                        android:layout_gravity="center"
                        android:id="@+id/play"
                        app:tint="@color/white">
                    </ImageView>

                </androidx.cardview.widget.CardView>

            </androidx.cardview.widget.CardView>
        </RelativeLayout>

    </RelativeLayout>
```

Step9:custom_dialog.xml

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

```
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            xmlns:app="http://schemas.android.com/apk/res-auto">

            <RelativeLayout
                android:layout_width="300dp"
                android:layout_height="220dp"
                android:background="#E1E1E1"
                android:layout_centerInParent="true">
                <androidx.cardview.widget.CardView
                    android:layout_width="90dp"
                    android:layout_height="90dp"
                    android:layout_marginTop="10dp"
                    android:id="@+id/okcard"
                    android:layout_centerHorizontal="true"
                    app:cardCornerRadius="90dp">
                    <ImageView
                        android:layout_width="match_parent"
                        android:layout_height="match_parent"
                        android:id="@+id/oklogo"
                        android:src="@drawable/ok"
                        android:scaleType="centerCrop">

                    </ImageView>

                </androidx.cardview.widget.CardView>
                <TextView
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:textColor="#434343"
                    android:textSize="25sp"
                    android:textStyle="bold"
                    android:text="Saved"
                    android:layout_centerHorizontal="true"
                    android:layout_marginTop="15dp"
                    android:layout_below="@+id/okcard">

                </TextView>
                <Button
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:text="OK"
                    android:layout_centerHorizontal="true"
                    android:id="@+id/okbutton"
                    android:layout_marginBottom="10dp"
                    android:layout_alignParentBottom="true">

                </Button>

            </RelativeLayout>


        </RelativeLayout>
```

Step10: create one more new class Picture.java in com.example.whatsappvideos and add the following code

```
package com.example.whatsappvideos;

import androidx.appcompat.app.AppCompatActivity;
```

```
import android.app.Dialog;
import android.content.Intent;
import android.media.MediaScannerConnection;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.Toast;

import com.bumptech.glide.Glide;

import java.io.File;
import java.io.IOException;

public class Picture extends AppCompatActivity {
    ImageView mparticularimage,download,mychatapp,share;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_picture);
        getSupportActionBar().setTitle("Picture");
        mparticularimage=findViewById(R.id.particularimage);
        share=findViewById(R.id.share);
        download=findViewById(R.id.download);
        mychatapp=findViewById(R.id.mychatapp);
        share.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Toast.makeText(getApplicationContext(),"share is
clicked",Toast.LENGTH_LONG).show();
            }
        });
        Intent intent=getIntent();
        String destpath=intent.getStringExtra("DEST_PATH");
        String file=intent.getStringExtra("FILE");
        String uri=intent.getStringExtra("URI");
        String filename=intent.getStringExtra("FILENAME");
        File destpath2=new File(destpath);
        File file1=new File(file);
        Glide.with(getApplicationContext()).load(uri).into(mparticularimage);
        download.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                try {

org.apache.commons.io.FileUtils.copyFileToDirectory(file1,destpath2);
                }catch (IOException e){
                    e.printStackTrace();
                }
                MediaScannerConnection.scanFile(getApplicationContext(), new
String[]{destpath + filename}, new String[]{"*/*"},
                        new MediaScannerConnection.MediaScannerConnectionClient() {
                            @Override
                            public void onMediaScannerConnected() {

                            }

                            @Override
```

```
                                public void onScanCompleted(String path, Uri uri) {

                                }
                            });
                    }
                });
                Dialog dialog=new Dialog(Picture.this);
                dialog.setContentView(R.layout.custom_dialog);
                dialog.show();
                Button button=dialog.findViewById(R.id.okbutton);
                button.setOnClickListener(new View.OnClickListener() {
                    @Override
                    public void onClick(View v) {

                    }
                });


            }
        }
```

Step11: activity_picture.xml

```
        <?xml version="1.0" encoding="utf-8"?>
        <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
            xmlns:tools="http://schemas.android.com/tools"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:backgroundTint="#FDF9F3"
            tools:context=".Picture">
            <RelativeLayout
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:padding="5dp"
                android:background="@color/white"
                <ImageView
                    android:layout_width="match_parent"
                    android:layout_height="match_parent"
                    android:src="@mipmap/ic_launcher_round"
                    android:layout_centerInParent="true"
                    android:layout_above="@+id/components"
                    android:id="@+id/particularimage">
                </ImageView>
                <LinearLayout
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:id="@+id/components"
                    android:background="#FFE8E8"
                    android:layout_alignParentBottom="true"
                    android:orientation="horizontal">
                    <ImageView
                        android:layout_width="50dp"
                        android:layout_height="50dp"
                        android:layout_margin="5dp"
                        android:src="@drawable/chatapp"
                        android:layout_weight="1"
                        android:id="@+id/mychatapp">
                    </ImageView>
                    <ImageView
                        android:layout_width="50dp"
                        android:layout_height="50dp"
                        android:layout_margin="5dp"
```

```
            android:src="@drawable/download"
            android:layout_weight="1"
            android:id="@+id/download">
        </ImageView>
        <ImageView
            android:layout_width="50dp"
            android:layout_height="50dp"
            android:layout_margin="5dp"
            android:src="@drawable/share"
            android:layout_weight="1"
            android:id="@+id/share">
        </ImageView>

    </LinearLayout>


</RelativeLayout>
```

Step12: create one more new class Video.java in com.example.whatsappvideos and add the following code

```
package com.example.whatsappvideos;

import androidx.appcompat.app.AppCompatActivity;

import android.app.Dialog;
import android.content.Intent;
import android.media.MediaScannerConnection;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.MediaController;
import android.widget.Toast;
import android.widget.VideoView;

import com.bumptech.glide.Glide;

import java.io.File;
import java.io.IOException;

public class Video extends AppCompatActivity {
    ImageView download,mychatapp,share;
    VideoView mparticularvideo;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_video);
        getSupportActionBar().setTitle("Video");
        mparticularvideo=findViewById(R.id.particularvideo);
        share=findViewById(R.id.share);
        download=findViewById(R.id.download);
        mychatapp=findViewById(R.id.mychatapp);
        share.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Toast.makeText(getApplicationContext(),"share is
clicked",Toast.LENGTH_LONG).show();
            }
```

```
        });
        Intent intent=getIntent();
        String destpath=intent.getStringExtra("DEST_PATH_VIDEO");
        String file=intent.getStringExtra("FILE_VIDEO");
        String uri=intent.getStringExtra("URI_VIDEO");
        String filename=intent.getStringExtra("FILENAME_VIDEO");
        File destpath2=new File(destpath);
        File file1=new File(file);
        MediaController mediaController=new MediaController(this);
        mediaController.setAnchorView(mparticularvideo);
        Uri uri1=Uri.parse(uri);
        mparticularvideo.setMediaController(mediaController);
        mparticularvideo.setVideoURI(uri1);
        mparticularvideo.requestFocus();
        mparticularvideo.start();
        //Glide.with(getApplicationContext()).load(uri).into(mparticularimage);
        download.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                try {

org.apache.commons.io.FileUtils.copyFileToDirectory(file1,destpath2);
                }catch (IOException e){
                    e.printStackTrace();
                }
                MediaScannerConnection.scanFile(getApplicationContext(), new
String[]{destpath + filename}, new String[]{"*/*"},
                        new
MediaScannerConnection.MediaScannerConnectionClient() {
                            @Override
                            public void onMediaScannerConnected() {

                            }

                            @Override
                            public void onScanCompleted(String path, Uri uri) {

                            }
                        });
            }
        });
        Dialog dialog=new Dialog(Video.this);
        dialog.setContentView(R.layout.custom_dialog);
        dialog.show();
        Button button=dialog.findViewById(R.id.okbutton);
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

            }
        });

    }
}
```

Step13: activity_video.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"
 android:backgroundTint="#FDF9F3"
 tools:context=".Video">
 <RelativeLayout
     android:layout_width="match_parent"
     android:layout_height="match_parent"
     android:padding="5dp"
     android:background="@color/white"

<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_centerHorizontal="true"
    android:layout_centerVertical="true"
    android:layout_above="@+id/components"
    android:layout_centerInParent="true">
 <VideoView
     android:layout_width="match_parent"
     android:layout_height="match_parent"
     android:layout_centerInParent="true"
     android:layout_centerHorizontal="true"
     android:layout_centerVertical="true"
     android:fitsSystemWindows="true"
     android:layout_gravity="center"
     android:id="@+id/particularvideo"
     android:layout_marginBottom="5dp">
 </VideoView>


</RelativeLayout>
 <LinearLayout
     android:layout_width="match_parent"
     android:layout_height="wrap_content"
     android:id="@+id/components"
     android:background="#FFE8E8"
     android:layout_alignParentBottom="true"
     android:orientation="horizontal">
     <ImageView
         android:layout_width="50dp"
         android:layout_height="50dp"
         android:layout_margin="5dp"
         android:src="@drawable/chatapp"
         android:layout_weight="1"
         android:id="@+id/mychatapp">
     </ImageView>
     <ImageView
         android:layout_width="50dp"
         android:layout_height="50dp"
         android:layout_margin="5dp"
         android:src="@drawable/download"
         android:layout_weight="1"
         android:id="@+id/download">
     </ImageView>
     <ImageView
         android:layout_width="50dp"
         android:layout_height="50dp"
         android:layout_margin="5dp"
         android:src="@drawable/share"
         android:layout_weight="1"
         android:id="@+id/share">
```

```
            </ImageView>

        </LinearLayout>


    </RelativeLayout>
```

Step14: build.gradle (Module: app)

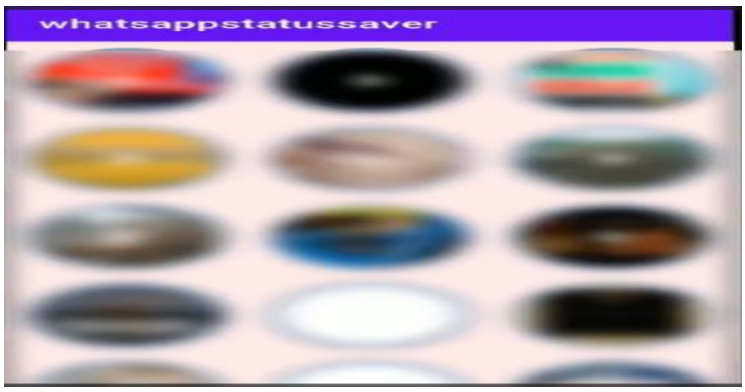
```
plugins {
    id 'com.android.application'
}

android {
    namespace 'com.example.whatsappvideos'
    compileSdk 33

    defaultConfig {
        applicationId "com.example.whatsappvideos"
        minSdk 24
        targetSdk 33
        versionCode 1
        versionName "1.0"

        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }

    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-
optimize.txt'), 'proguard-rules.pro'
        }
    }
    compileOptions {
        sourceCompatibility JavaVersion.VERSION_1_8
        targetCompatibility JavaVersion.VERSION_1_8
    }
}

dependencies {

    implementation 'androidx.appcompat:appcompat:1.6.1'
    implementation 'com.google.android.material:material:1.9.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.1.4'
    testImplementation 'junit:junit:4.13.2'
    androidTestImplementation 'androidx.test.ext:junit:1.1.5'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.5.1'

    implementation 'org.apache.commons:commons-lang3:3.5'
    implementation 'androidx.swiperefreshlayout:swiperefreshlayout:1.2.0-
alpha01'
    implementation 'org.apache:apache:23'
    implementation 'commons-io:commons-io:2.9.0'
    implementation 'com.github.bumptech.glide:glide:4.12.0'
}
```

step15:AndroidManifest.xml

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

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

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:requestLegacyExternalStorage="true"
        android:supportsRtl="true"
        android:theme="@style/Theme.WhatsAppVideos"
        tools:targetApi="31">
        <activity
            android:name=".Picture"
            android:exported="false" />
        <activity
            android:name=".Video"
            android:exported="false" />

        <provider
            android:name="androidx.core.content.FileProvider"
            android:authorities="${applicaitonId}.provider"
            android:exported="false"
            android:grantUriPermissions="true">
            <meta-data
                android:name="android.support.FILE_PROVIDER_PATHS"
                android:resource="@xml/file_provider_paths" />
        </provider>

        <activity
            android:name=".MainActivity"
            android:exported="false" />
        <activity
            android:name=".splash"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

</manifest>
```

Output:



Practical 7: Create an android application to display webpage by using Web view Component

Activity_main.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">

    <WebView
        android:id="@+id/wView"
        android:layout_width="match_parent"
        android:layout_height="match_parent"/>
    <ProgressBar
        android:id="@+id/pgBar"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:visibility="gone"
        android:layout_centerInParent="true"/>

</RelativeLayout>
```

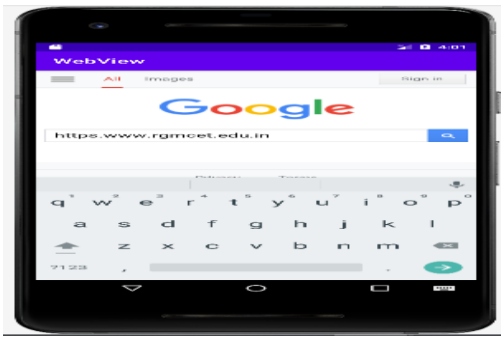
AndroidManifest.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">
    <uses-permission android:name="android.permission.INTERNET"/>
    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/Theme.WebView"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

</manifest>
```

MainActivity.java

```
package com.example.webview;

import androidx.appcompat.app.AppCompatActivity;

import android.graphics.Bitmap;
import android.os.Bundle;
import android.view.View;
import android.webkit.WebView;
import android.webkit.WebViewClient;
import android.widget.ProgressBar;

import java.net.URI;

public class MainActivity extends AppCompatActivity {
    WebView wView;
    ProgressBar pgBar;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        wView=findViewById(R.id.wView);
        pgBar=findViewById(R.id.pgBar);
        wView.loadUrl("https://www.google.com");
        wView.setWebViewClient(new WebViewClient(){
            @Override
                    public  void onPageStarted(WebView view, String url, Bitmap
favicon){
                pgBar.setVisibility(View.VISIBLE);
                super.onPageStarted(view,url,favicon);
            }
            @Override
            public void onPageFinished(WebView view,String URL){
                pgBar.setVisibility(View.GONE);
                super.onPageFinished(view,URL);
            }
        });
    }
    @Override
    public void onBackPressed(){

        if(wView.canGoBack()){
            wView.goBack();

        }else {
            super.onBackPressed();
        }
    }
}
```

Output:



Practical 8: Create an android application to display different webpages in fragments by using Fragments Component

Step1: create new project with name WebViewFragments

Step2: Use the following code and update build.gradle(Module:app)

```
plugins {
    id 'com.android.application'
}

android {
    namespace 'com.example.webviewfragments'
    compileSdk 33

    defaultConfig {
        applicationId "com.example.webviewfragments"
        minSdk 24
        targetSdk 33
        versionCode 1
        versionName "1.0"

        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }

    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-
optimize.txt'), 'proguard-rules.pro'
        }
    }
    compileOptions {
        sourceCompatibility JavaVersion.VERSION_1_8
        targetCompatibility JavaVersion.VERSION_1_8
    }
}

dependencies {

    implementation 'androidx.appcompat:appcompat:1.6.1'
    implementation 'com.google.android.material:material:1.9.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.1.4'
    testImplementation 'junit:junit:4.13.2'
    androidTestImplementation 'androidx.test.ext:junit:1.1.5'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.5.1'
    //the below code need to sync with latest design to do this need need
which version wee need?
    //implementation 'com.android.support:design: 28.0.0'
            ''
}
```

step3: create the following 3 framents by fragment class: to create a new fragment right click on com.example.WebViewFragments----->New---->Select Fragment--->select blank fragment and name it to create 3 fragments like FirstFragment,SecondFragment & ThirdFragment

step4: now create on pageAdapter class (right click on com.example.WebViewFragments----->New--->Java Class---->Name it as PageAdapter-->click on Finish)

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

   <com.google.android.material.tabs.TabLayout
       android:layout_width="match_parent"
       android:layout_height="wrap_content"
       android:id="@+id/sliding_tabs"
       app:tabMode="fixed"/>
    <androidx.viewpager.widget.ViewPager
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:background="@android:color/white"
        android:id="@+id/viwepager"/>

</LinearLayout>
```

fragment_first.xml;

==========

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

   <WebView
       android:layout_width="match_parent"
       android:layout_height="match_parent"
       android:id="@+id/webView1"/>

</FrameLayout>
```

FirstFragment.java

==========

```
package com.example.webviewfragments;

import android.os.Bundle;

import androidx.fragment.app.Fragment;

import android.view.LayoutInflater;
```

```
import android.view.View;
import android.view.ViewGroup;
import android.webkit.WebSettings;
import android.webkit.WebView;

public class FirstFragment extends Fragment {

     @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
       View myView=inflater.inflate(R.layout.fragment_first,container,false);
         WebView myWebView=myView.findViewById(R.id.webView1);
         myWebView.loadUrl("https://www.google.com");
         WebSettings webSettings=myWebView.getSettings();
         webSettings.setJavaScriptEnabled(true);

        return myView;
    }
}
```

fragment_second.xml

==========

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

    <WebView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/webView2"/>

</FrameLayout>
```

SecondFragment.java

==========

```
package com.example.webviewfragments;

import android.os.Bundle;

import androidx.fragment.app.Fragment;

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.webkit.WebResourceRequest;
import android.webkit.WebSettings;
import android.webkit.WebView;
import android.webkit.WebViewClient;


public class SecondFragment extends Fragment {
```

```
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        View view=inflater.inflate(R.layout.fragment_second, container, false);
        WebView webView=view.findViewById(R.id.webView2);
        webView.loadUrl("https://www.rgmcet.edu.in");
        webView.setWebViewClient(new WebViewClient(){
            @Override
            public boolean shouldOverrideUrlLoading(WebView view,
WebResourceRequest request) {
                return false;
            }
        });
        WebSettings webSettings=webView.getSettings();
        webSettings.setJavaScriptEnabled(true);
       return view;
    }
}
```

fragment_third.xml

==========

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

    <!-- TODO: Update blank fragment layout -->
    <TextView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:text="this is third fragment"
        android:textColor="@android:color/white"
        android:textSize="25sp"
        android:gravity="center"
        android:background="@android:color/holo_orange_dark"/>

</FrameLayout>
```

ThirdFragment.java

==========

```
package com.example.webviewfragments;

import android.os.Bundle;

import androidx.fragment.app.Fragment;

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

/**
 * A simple {@link Fragment} subclass.
```

```
 * Use the {@link ThirdFragment#newInstance} factory method to
 * create an instance of this fragment.
 */
public class ThirdFragment extends Fragment {

    // TODO: Rename parameter arguments, choose names that match
    // the fragment initialization parameters, e.g. ARG_ITEM_NUMBER
    private static final String ARG_PARAM1 = "param1";
    private static final String ARG_PARAM2 = "param2";

    // TODO: Rename and change types of parameters
    private String mParam1;
    private String mParam2;

    public ThirdFragment() {
        // Required empty public constructor
    }

    /**
     * Use this factory method to create a new instance of
     * this fragment using the provided parameters.
     *
     * @param param1 Parameter 1.
     * @param param2 Parameter 2.
     * @return A new instance of fragment ThirdFragment.
     */
    // TODO: Rename and change types and number of parameters
    public static ThirdFragment newInstance(String param1, String param2) {
        ThirdFragment fragment = new ThirdFragment();
        Bundle args = new Bundle();
        args.putString(ARG_PARAM1, param1);
        args.putString(ARG_PARAM2, param2);
        fragment.setArguments(args);
        return fragment;
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        if (getArguments() != null) {
            mParam1 = getArguments().getString(ARG_PARAM1);
            mParam2 = getArguments().getString(ARG_PARAM2);
        }
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_third, container, false);
    }
}
```

PageAdapter.java:

==========

```
package com.example.webviewfragments;

import androidx.annotation.NonNull;
```

```
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;
import androidx.fragment.app.FragmentManager;
import androidx.fragment.app.FragmentStatePagerAdapter;

public class PageAdapter extends FragmentStatePagerAdapter {
    final int pageCount=3;
    private String tabTitles[]=new String[]{"Tab 1","Tab 2","Tab 3"};

    public PageAdapter(@NonNull FragmentManager fm) {
        super(fm,BEHAVIOR_RESUME_ONLY_CURRENT_FRAGMENT);
    }

    @NonNull
    @Override
    public Fragment getItem(int position) {
        switch (position){
            case 0:
                return new FirstFragment();
            case 1:
                return new SecondFragment();
            case 2:
                return new ThirdFragment();
            default:
                return null;
        }

    }

    @Override
    public int getCount() {
        return pageCount;
    }

    @Nullable
    @Override
    public CharSequence getPageTitle(int position) {
        return tabTitles[position];
    }
}
```

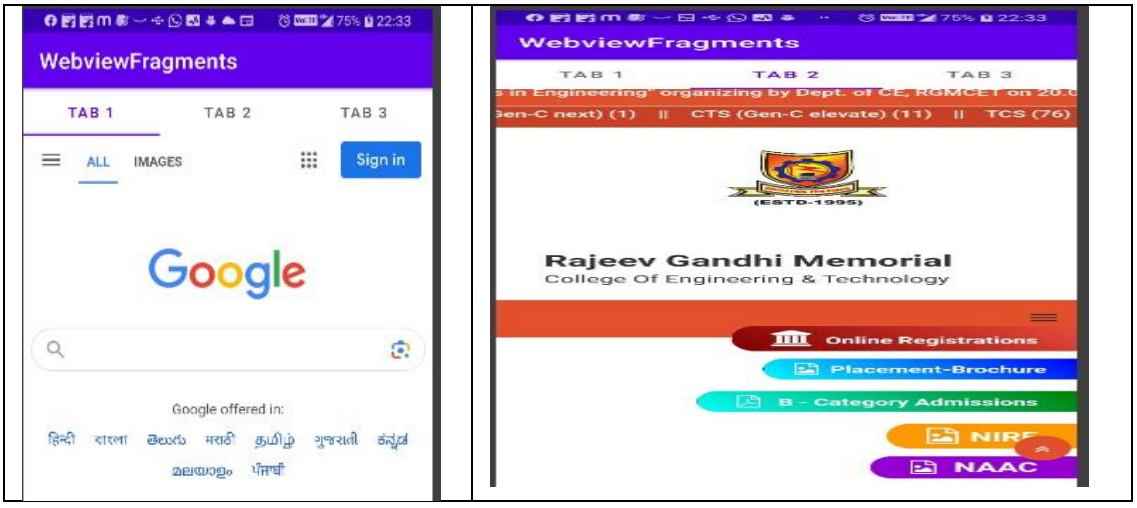
MainActivity.java

==========

```
package com.example.webviewfragments;

import androidx.appcompat.app.AppCompatActivity;
import androidx.viewpager.widget.ViewPager;

import android.os.Bundle;

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

public class MainActivity extends AppCompatActivity {

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

```
            //get the viewpager and set its pageadapter so that it can display items
            ViewPager vp=findViewById(R.id.viwepager);
            PageAdapter pA=new PageAdapter(getSupportFragmentManager());
            vp.setAdapter(pA);
            //give the tabloayout the viewpager
            TabLayout tL=findViewById(R.id.sliding_tabs);
            tL.setupWithViewPager(vp);

        }
    }
```

Output:



Practical 9: Create an android application to store the data by using Shared Preferences

Shared Preferences:

- One of the way to store data in in android
- It saves and retrieves data in the form of key & value pair

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:background="#0091EA"
    android:gravity="center"
    tools:context=".MainActivity">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="BJR_Tech"
        android:textSize="30sp"
        android:textStyle="bold"
        android:textColor="#fff"/>
</LinearLayout>
```

Activity_home.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"
    android:background="#0091EA"
    tools:context=".HomeActivity">
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/btnlogout"
        android:textColor="#FFAB00"
        android:textSize="30sp"
        android:layout_alignParentRight="true"
        android:layout_margin="11dp"
        android:text="Logout"/>
</RelativeLayout>
```

Activity_login.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:background="#FFAB00"
    android:gravity="center"
    tools:context=".LoginActivity">
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/btnlogin"
        android:text="Login"
        android:textSize="30sp"/>
</LinearLayout>
```

MainActivity.java:

```
package com.rgmcet.sharedpreferences;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.os.Handler;
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        new Handler().postDelayed(new Runnable() {
            @Override
            public void run() {

                SharedPreferences
pref=getSharedPreferences("login",MODE_PRIVATE);
                Boolean check=pref.getBoolean("flag",false);
                Intent iNext;
                if(check){ //for true(if user is logged in)
                    iNext=new Intent(MainActivity.this,HomeActivity.class);
                }else{  //for false(either first time or user is logged out)
                    iNext=new Intent(MainActivity.this,LoginActivity.class);
                }
                startActivity(iNext);
            }
        },4000 );
    }
}
```

HomeActivity.java

```
package com.rgmcet.sharedpreferences;
import androidx.appcompat.app.AppCompatActivity;
```

```
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
public class HomeActivity extends AppCompatActivity {
    Button btnLogout;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_home);

        btnLogout=findViewById(R.id.btnlogout);
        btnLogout.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                SharedPreferences
pref=getSharedPreferences("login",MODE_PRIVATE);
                SharedPreferences.Editor editor= pref.edit();
                editor.putBoolean("flag",false);
                editor.apply();
            }
        });
    }
}
```

LoginActivity.java

```
package com.rgmcet.sharedpreferences;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
public class LoginActivity extends AppCompatActivity {
        Button btnLogin;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
        btnLogin=findViewById(R.id.btnlogin);
        btnLogin.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                //code for verification
                SharedPreferences
pref=getSharedPreferences("login",MODE_PRIVATE);
                SharedPreferences.Editor editor= pref.edit();
                editor.putBoolean("flag",true);
                editor.apply();
                Intent iHome=new Intent(LoginActivity.this,HomeActivity.class);
                startActivity(iHome);
            }
        });
}
```

Step1:

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"
    tools:context=".MainActivity">

    <FrameLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/framelayout_two"/>
</LinearLayout>
```

Step2:

- **Create one fragment for login with name Login.java(right click on package---->new---->fragment----->fragment(Blank)----->name it as Login----->click on finish)**

Step3:

- **Create one fragment for login with name Register.java(right click on package---->new---->fragment----->fragment(Blank)----->name it as Register----->click on finish)**

Step4:

Fragment_login.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".Login"
    android:gravity="center"
    android:orientation="vertical"
    android:background="#D5E4F1">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Login Here"
        android:textSize="30sp"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="If not registered...."
        android:textSize="20sp"
        android:layout_marginTop="25dp"/>
    <TextView
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Click Here"
        android:textSize="30sp"
        android:textStyle="bold"
        android:textColor="#FF5722"
        android:id="@+id/register_tv"/>
</LinearLayout>
```

Step5:

Fragment_register.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".Register"
    android:gravity="center"
    android:orientation="vertical"
    android:background="#E5BFEC">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Register Here"
        android:textSize="30sp"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="If already registered...."
        android:textSize="20sp"
        android:layout_marginTop="25dp"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Click Here"
        android:textSize="30sp"
        android:textStyle="bold"
        android:textColor="#FF5722"
        android:id="@+id/login_tv"/>
</LinearLayout>
```

Step6:

MainActivity.java

```
package com.example.fragmentsex3;

import androidx.appcompat.app.AppCompatActivity;
import androidx.fragment.app.FragmentManager;
import androidx.fragment.app.FragmentTransaction;

import android.os.Bundle;
import android.widget.FrameLayout;

public class MainActivity extends AppCompatActivity {
    FrameLayout framelayout_two;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

    framelayout_two=(FrameLayout)findViewById(R.id.framelayout_two);
        FragmentManager fragmentManager=getSupportFragmentManager();
        FragmentTransaction
fragmentTransaction=fragmentManager.beginTransaction();
        fragmentTransaction.add(R.id.framelayout_two,new Login());
        fragmentTransaction.commit();

    }
}
```

Step7:

Login.java

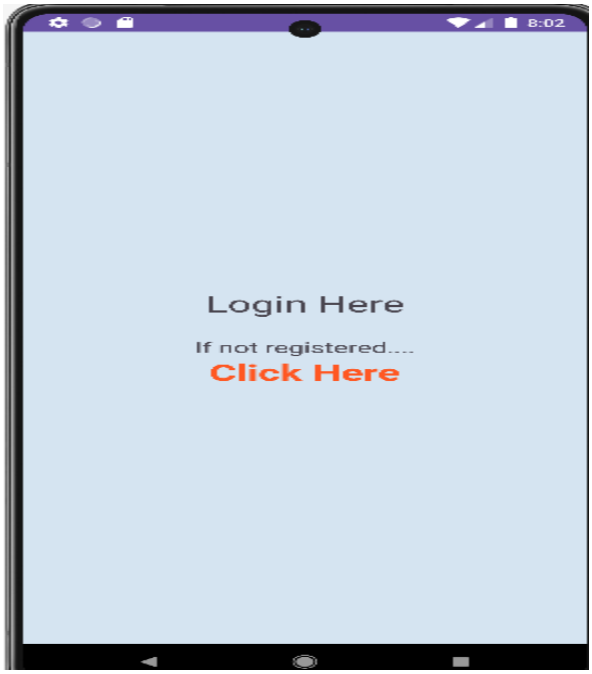
```
package com.example.fragmentsex3;

import android.os.Bundle;

import androidx.fragment.app.Fragment;
import androidx.fragment.app.FragmentManager;
import androidx.fragment.app.FragmentTransaction;

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;


public class Login extends Fragment {
    TextView register_tv;
    View view;
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
container,
                             Bundle savedInstanceState) {

view=inflater.inflate(R.layout.fragment_login,container,false);
        register_tv=(TextView) view.findViewById(R.id.register_tv);
        register_tv.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                FragmentManager
fragmentManager=getParentFragmentManager();
                FragmentTransaction
fragmentTransaction=fragmentManager.beginTransaction();
                fragmentTransaction.replace(R.id.framelayout_two,new
Register()).commit();
            }
        });
        return view;
    }
}
```

Step8:

Register.java

```
package com.example.fragmentsex3;

import android.os.Bundle;

import androidx.fragment.app.Fragment;
import androidx.fragment.app.FragmentManager;
import androidx.fragment.app.FragmentTransaction;

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;


public class Register extends Fragment {
    View view;
    TextView login_tv;
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
container,
                             Bundle savedInstanceState) {
       view=inflater.inflate(R.layout.fragment_register, container,
false);
       login_tv=(TextView) view.findViewById(R.id.login_tv);
       login_tv.setOnClickListener(new View.OnClickListener() {
           @Override
           public void onClick(View view) {
               FragmentManager
fragmentManager=getParentFragmentManager();
               FragmentTransaction
fragmentTransaction=fragmentManager.beginTransaction();
               fragmentTransaction.replace(R.id.framelayout_two,new
Login()).commit();
           }
       });
        return view;
    }
}
```

Ouput:


8:02
Login Here
If not registered....
Click Here

Practical 10: Create an android application and demonstrate concept of SQLite Database storage method

Step1: create a project with name SqlliteExample and add the following code in activity_main.xml

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

   <EditText
       android:layout_width="match_parent"
       android:layout_height="wrap_content"
       android:hint="Enter Name"
              android:id="@+id/reg_name"/>
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter Email"
        android:id="@+id/reg_email"/>
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter password"
        android:inputType="textPassword"
        android:id="@+id/reg_pass"/>
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter Gender"
        android:id="@+id/reg_gender"/>
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="register"
        android:onClick="registerUser"/>

</LinearLayout>
```

Step2: Now create a new java class with name Dbheloper.java and add the following code

```
package com.example.sqliteexample;

import android.content.ContentValues;
import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

import androidx.annotation.Nullable;
```

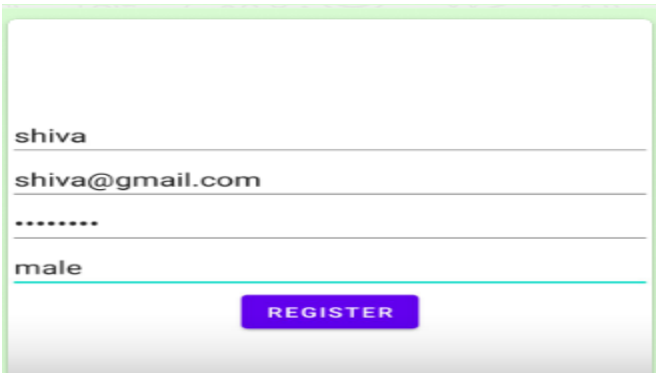
```
public class DbHeloper extends SQLiteOpenHelper {
   private static final String DATABASE_NAME="demo_db";
   private static final int DATABASE_VERSION=1;
    public DbHeloper(@Nullable Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        String CREATE_TABLE_QUERY="CREATE TABLE register(id INTEGER PRIMARY KEY
AUTOINCREMENT,name TEXT,email TEXT,password TEXT,gender TEXT)";
        db.execSQL(CREATE_TABLE_QUERY);

    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        db.execSQL("DROP TABLE IF EXISTS register");
        onCreate(db);

    }
    public boolean registerUserHelper(String name1,String email1,String pass1,String
gender1){

        SQLiteDatabase sqLiteDatabase=this.getWritableDatabase();
        ContentValues contentValues=new ContentValues();
        contentValues.put("name",name1);
        contentValues.put("email",email1);
        contentValues.put("password",pass1 );
        contentValues.put("gender",gender1);
        long l=sqLiteDatabase.insert("register",null,contentValues);
        sqLiteDatabase.close();
        if(l>0){
            return true;
        }else {
            return false;
        }

    }
}
```

step3: Now add the following code in MainActivity.java

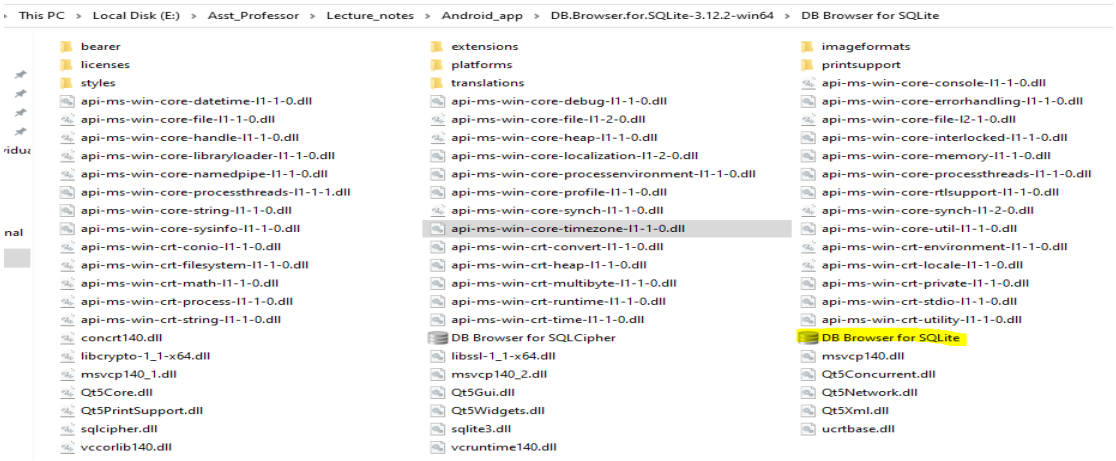
```
    package com.example.sqliteexample;

    import androidx.appcompat.app.AppCompatActivity;

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

    public class MainActivity extends AppCompatActivity {
        EditText reg_name,reg_email,reg_pass,reg_gender;
        DbHeloper dbHeloper;
        @Override
        protected void onCreate(Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
            setContentView(R.layout.activity_main);
            reg_name=(EditText) findViewById(R.id.reg_name);
```

```
            reg_email=(EditText)findViewById(R.id.reg_email);
            reg_pass=(EditText) findViewById(R.id.reg_pass);
            reg_gender=(EditText) findViewById(R.id.reg_gender);
            dbHeloper=new DbHeloper(getApplicationContext());
        }
        public void registerUser(View v){
            String name1=reg_name.getText().toString();
            String email1=reg_email.getText().toString();
            String pass1=reg_pass.getText().toString();
            String gender1=reg_gender.getText().toString();
            boolean b=dbHeloper.registerUserHelper(name1,email1,pass1,gender1);
            if(b==true) {
                Toast.makeText(this, "user registered successful",
    Toast.LENGTH_LONG).show();
                reg_name.setText("");
                reg_email.setText("");
                reg_pass.setText("");
                reg_gender.setText("");
            }else {
                Toast.makeText(this, "error.....!", Toast.LENGTH_LONG).show();
            }
        }
    }
```

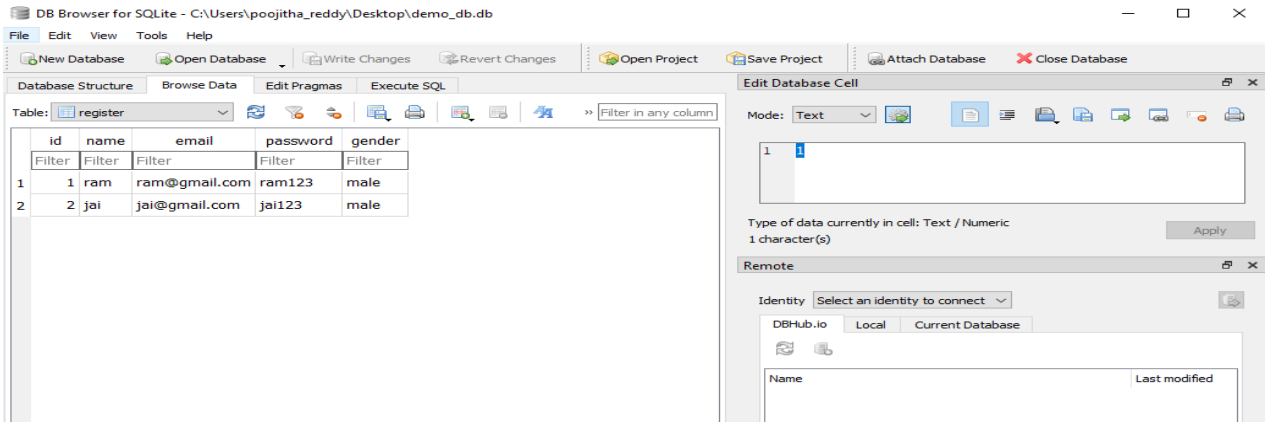
Output:



- After enter the details click on register button then all values are saved in project folder under database named demo_db database, to see the data in a table we have to download sqllite database brower in zip file from google and save in our system and open the following application



- By using highleted application we view the data from register table which is create and inserted values by using above code
- To see database folder we need to go to the following path
- Click on device file explorer which available on right side of android studio--->expand data folder---->again data folder----->now search for the package like com.example.sqlliteexample---->expand this and here we can see databases folder and under this we see database demo_db---->now right click on this and save as in your system with extension like demo_db.db and open this in DB brower for sqllite and see the result



Practical 11: Create an android application to perform different types of operations (Send SMS, Making Call and sending Email) by using Telephony app

Making Call Using Android Programming:

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"
    tools:context=".CallDemo"
    android:gravity="center"
    android:padding="20dp"
    android:orientation="vertical">
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter Phone No"
        android:id="@+id/phoneno"
        android:inputType="phone"
        android:textSize="25dp"/>
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Call"
        android:textSize="25dp"
        android:onClick="callToNumber"/>

</LinearLayout>
```

MainActivity.java:

```
package com.example.permissionsdemo;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;

public class CallDemo extends AppCompatActivity {
        EditText phoneno;
```

```
    private  final int REQUEST_CALL_CODE=1;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_call_demo);
        phoneno=(EditText) findViewById(R.id.phoneno);
    }
    public void callToNumber(View view){

        if(ContextCompat.checkSelfPermission
                (CallDemo.this, Manifest.permission.CALL_PHONE)==
PackageManager.PERMISSION_GRANTED){
            call();
        }else {
            ActivityCompat.requestPermissions(CallDemo.this,new
String[]{Manifest.permission.CALL_PHONE},REQUEST_CALL_CODE);
        }
    }
    Void call(){
        String phno=phoneno.getText().toString();
        Intent intent=new Intent(Intent.ACTION_CALL);
        intent.setData(Uri.parse("tel:"+phno));
        startActivity(intent);
        return null;
    }
    @Override
    public void onRequestPermissionsResult(int requestCode,
@NonNull String[] permissions, @NonNull int[] grantResults) {
        super.onRequestPermissionsResult(requestCode,
permissions, grantResults);
        if(grantResults.length>0 &&
grantResults[0]==PackageManager.PERMISSION_GRANTED &&
REQUEST_CALL_CODE==1){
            call();
        }
    }
}
```

Manifest.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">
    <uses-permission android:name="android.permission.CALL_PHONE" />
    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.PermissionsDemo"
```

```
        tools:targetApi="31">
        <activity
            android:name=".CallDemo"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

</manifest>
```

Ouput:

Sending Email:

==========

```
package com.example.emailapp;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

public class MainActivity extends AppCompatActivity {
    EditText email,subject,message;
    Button send;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        email=(EditText) findViewById(R.id.email);
        subject=(EditText) findViewById(R.id.subject);
        message=(EditText) findViewById(R.id.message);
        send=(Button) findViewById(R.id.button);
        send.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String emailto=(email).getText().toString();
                String[] my_recievers=emailto.split(",");
                String subjectmsg=(subject).getText().toString();
                String msg=(message).getText().toString();

                Intent intent=new Intent(Intent.ACTION_SEND);
                intent.setData(Uri.parse("email"));
                intent.putExtra(Intent.EXTRA_EMAIL,my_recievers);
                intent.putExtra(Intent.EXTRA_SUBJECT,subjectmsg);
                intent.putExtra(Intent.EXTRA_TEXT,msg);
                intent.setType("message/rfc822");
                Intent
chooser=Intent.createChooser(intent,"Choose An email client");
                startActivity(chooser);
            }
        });
    }
    }
```

Following will be the content of res/layout/activity_main.xml file

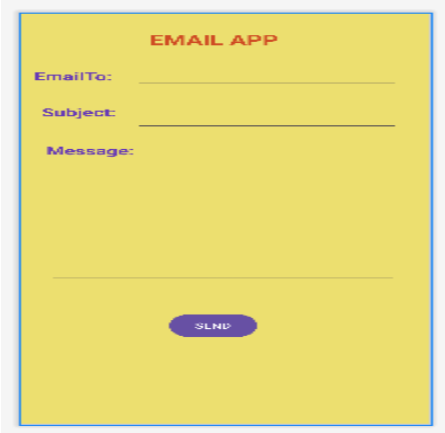
```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#ECDF6F"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="EMAIL APP"
        android:textColor="#CF4C23"
        android:textSize="25dp"
        android:textStyle="bold"
```

```
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.466"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.045" />


    <TextView
        android:layout_width="83dp"
        android:layout_height="36dp"
        android:text="EmailTo:"
        android:textColor="#673AB7"
        android:textSize="20dp"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.046"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.143" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Subject:"
        android:textColor="#673AB7"
        android:textSize="20dp"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.068"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.232" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Message:"
        android:textColor="#673AB7"
        android:textSize="20dp"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.086"
```

```
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.329" />

    <EditText
        android:id="@+id/email"
        android:layout_width="265dp"
        android:layout_height="46dp"
        android:ems="10"
        android:inputType="text"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.794"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.127" />

    <EditText
        android:id="@+id/subject"
        android:layout_width="265dp"
        android:layout_height="46dp"
        android:ems="10"
        android:inputType="text"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.794"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.239" />

    <EditText
        android:id="@+id/message"
        android:layout_width="350dp"
        android:layout_height="202dp"
        android:ems="10"
        android:inputType="text"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.491"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.527" />

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

```
        android:layout_height="wrap_content"
        android:text="SEND"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.467"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.78" />

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

Output:

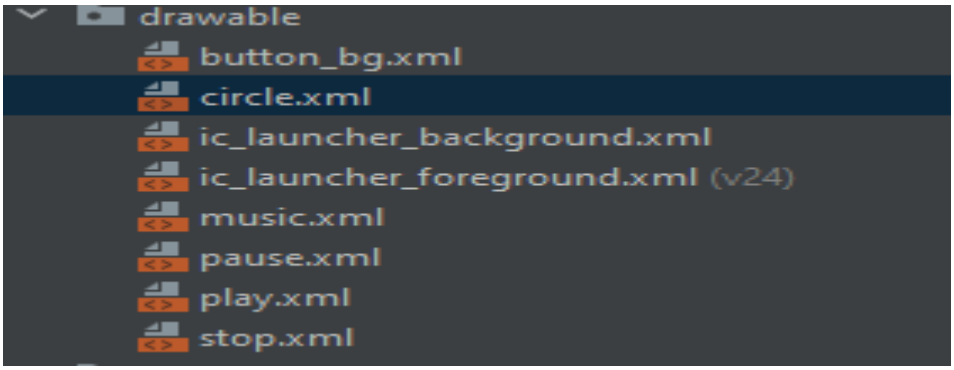


Practical 12: Write an android program to develop Media player application

Step1: first Create 4 images (1 music icon, play icon, pause icon, stop icon) under res--->drwable--->(with names: music.xml,pause.xml,play.xml,stop.xml)

Example: first in project folder right click on drawable-->new---->click on vector asset--->click on clip art icon---->search for music icon and select music note icon--->name it for this as music and size 50dpX50dp--->then click on next--->finish

Note: like this create for play.xml,pause.xml,stop.xml



Step2: Now check in res folder---->values folder---> here check for below colors are added or not in colors.xml file, if not add new tag for colors.

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="black">#FF000000</color>
    <color name="white">#FFFFFFFF</color>
    <color name="purple_200">#FFBB86FC</color>
    <color name="purple_500">#FF6200EE</color>
    <color name="purple_700">#4211B1</color>
    <color name="teal_200">#FF03DAC5</color>
    <color name="teal_700">#FF018786</color>
    <color name="blue">#1737E6</color>
    <color name="lightblue">#B3BBEC</color>
    <color name="blue_dark">#1A2D96</color>

</resources>
```

Step3: Now create one drawable resource file as circle.xml(right click on drwable folder---->click on new---->click on drawable resource file----->name it as circle.xml and add below code

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

```
    <solid
        android:color="@color/blue"/>
    <size
        android:width="200dp"
        android:height="200dp"/>
    <stroke
        android:color="@color/purple_500"
        android:width="6dp"/>
    <corners
        android:radius="100dp"/>
</shape>
```

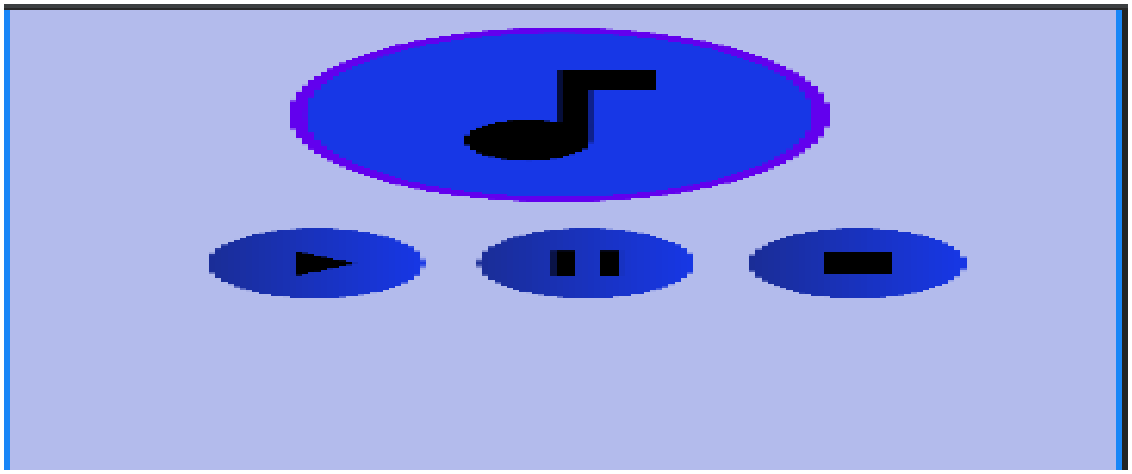
Step4: Now create one drawable resource file as button_bg.xml(right click on drwable folder---->click on new---->click on drawable resource file----->name it as button_bg.xml and add below code

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <size
        android:width="80dp"
        android:height="80dp"/>
    <corners
        android:radius="40dp"/>
    <gradient
        android:startColor="@color/blue_dark"
        android:endColor="@color/blue"/>
</selector>
```

Step5: now under Activity_main.xml design the layout as below 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"
    android:background="@color/lightblue">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:orientation="vertical"
        android:layout_centerInParent="true">
        <ImageView
            android:layout_width="200dp"
            android:layout_height="200dp"
            android:src="@drawable/music"
            android:background="@drawable/circle"
```

```
            android:padding="30dp">
        </ImageView>
        <LinearLayout
            android:layout_marginTop="30dp"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal"
            android:gravity="center">
            <ImageButton
                android:id="@+id/play"
                android:layout_width="80dp"
                android:layout_height="80dp"
                android:layout_marginLeft="20dp"
                android:background="@drawable/circle"
                android:contentDescription="pause"
                android:src="@drawable/play" />
            <ImageButton
                android:id="@+id/pause"
                android:layout_width="80dp"
                android:layout_height="80dp"
                android:layout_marginLeft="20dp"
                android:background="@drawable/circle"
                android:contentDescription="stop"
                android:src="@drawable/pause" />
            <ImageButton
                android:id="@+id/stop"
                android:layout_width="80dp"
                android:layout_height="80dp"
                android:layout_marginLeft="20dp"
                android:background="@drawable/circle"
                android:contentDescription="play"
                android:src="@drawable/stop" />
        </LinearLayout>

    </LinearLayout>

</RelativeLayout>
```



Step6: Create a new directory to save music/mp3 song in res folder with name raw

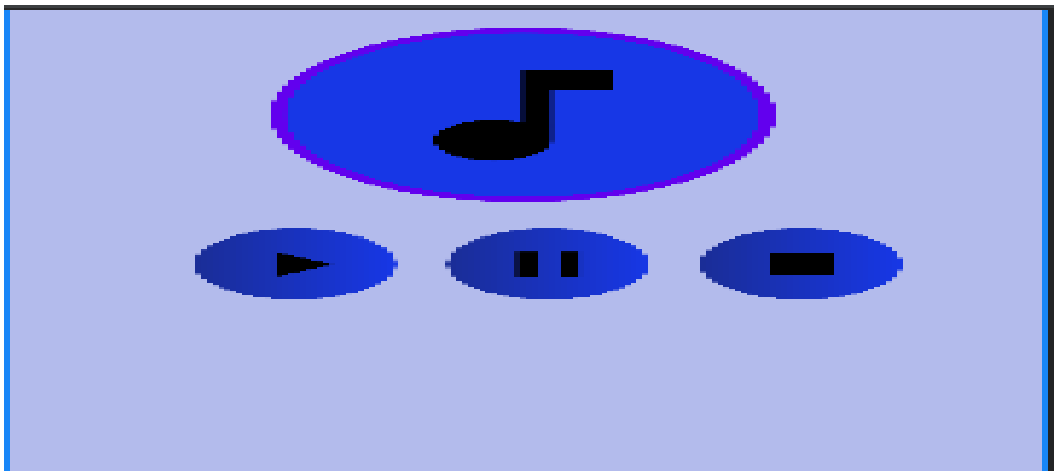
In project folder---->res folder----->right click and click on new----->click on Android resource directory--->directory name: raw ,value:raw and click ok--->now download on mp3 song and paste it into this folder.

Step7: Now write the below code in MainAcitivity.java

```
package com.example.mediaplayerex1;

import androidx.appcompat.app.AppCompatActivity;

import android.media.MediaPlayer;
import android.os.Bundle;
import android.view.View;
import android.widget.ImageButton;

public class MainActivity extends AppCompatActivity {
    ImageButton play,pause,stop;
    MediaPlayer mediaPlayer;
    boolean isPlaying=false;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        play=findViewById(R.id.play);
        pause=findViewById(R.id.pause);
        stop=findViewById(R.id.stop);
        mediaPlayer = MediaPlayer.create(this,R.raw.hanuman);
        play.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if(!isPlaying){
                    mediaPlayer.start();
                    isPlaying=true;
                }
            }
```

```
        });
        pause.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if(isPlaying){
                    mediaPlayer.pause();
                    isPlaying=false;
                }
            }
        });
        stop.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if(isPlaying){
                    mediaPlayer.stop();
                    isPlaying=false;
                }
            }
        });

    }
}
```

Output:



Practical 13 A: Write an android program to develop VideoView Application

Step1:

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"
    tools:context=".MainActivity">

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">
    <VideoView
        android:layout_width="match_parent"
        android:layout_height="250dp"
        android:id="@+id/videoView"/>
    </LinearLayout>

</LinearLayout>
```

Step2:

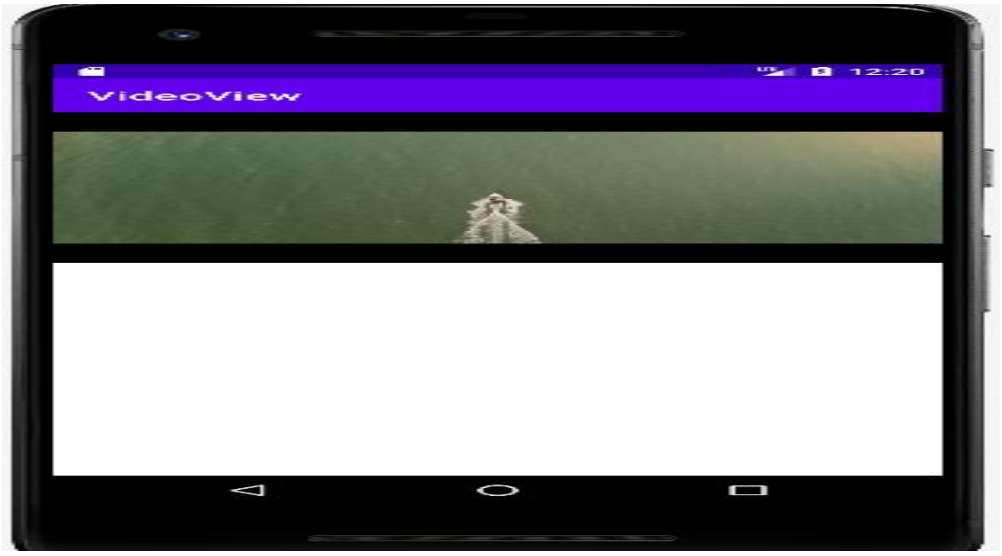
- **Download one video and create one raw resource directory under res folder and paste this video(video name should be small letters)**
  - **Right click on res folder-----click on new---->click on android resource directory---->enter directory name: raw, value=raw and click on ok button----->now copy the video and right click on raw folder and click on paste**

Step3:

```
package com.example.videoview;

import androidx.appcompat.app.AppCompatActivity;

import android.net.Uri;
import android.os.Bundle;
import android.widget.MediaController;
import android.widget.VideoView;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        VideoView videoView=(VideoView) findViewById(R.id.videoView);
        String vPath="android.resource://"+getPackageName()+"/raw/godfather";
        Uri videoUri=Uri.parse(vPath);
        //videoView.setVideoPath(vPath);
        videoView.setVideoURI(videoUri);
        videoView.start();
        MediaController mediaController=new MediaController(this);
```

```
            videoView.setMediaController(mediaController);
            mediaController.setAnchorView(videoView);
        }
    }
```

Output:



Practical 13B: Write an android program to develop Audio Recording Application

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/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="316dp"
        android:text="Record"
        android:onClick="btnRecordPressed"
        app:layout_constraintEnd_toStartOf="@+id/button2"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="316dp"
        android:text="Stop"
        android:onClick="btnStopPressed"
        app:layout_constraintEnd_toStartOf="@+id/button3"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toEndOf="@+id/button"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/button3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="316dp"
        android:text="Play"
        android:onClick="btnPlayPressed"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toEndOf="@+id/button2"
        app:layout_constraintTop_toTopOf="parent" />


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

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">
    <uses-permission android:name="android.permission.RECORD_AUDIO"/>
    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.AudioRecordingEx"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

</manifest>
```

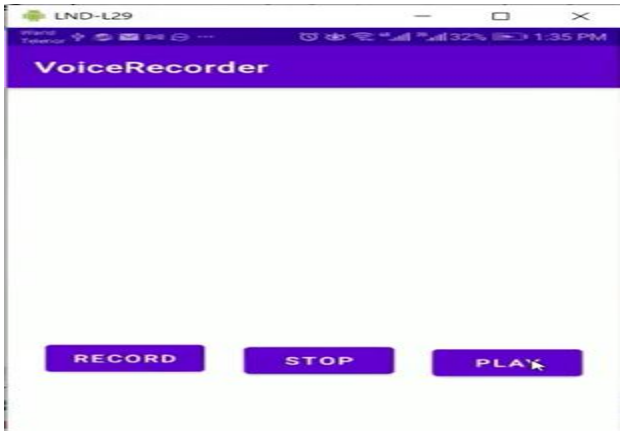
MainActivity.Java

```
package com.example.audiorecordingex;

import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import android.Manifest;
import android.content.ContextWrapper;
import android.content.pm.PackageManager;
import android.media.MediaPlayer;
import android.media.MediaRecorder;
import android.os.Bundle;
import android.os.Environment;
import android.view.View;
import android.widget.Toast;

import java.io.File;

public class MainActivity extends AppCompatActivity {
    MediaRecorder mediaRecorder;
    MediaPlayer mediaPlayer;
```

```
    private  static int MICROPHONE_PERMISSION_CODE=200;

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

    }
    public void btnRecordPressed(View v){

        try {
            mediaRecorder = new MediaRecorder();

mediaRecorder.setAudioSource(MediaRecorder.AudioSource.MIC);

mediaRecorder.setOutputFormat(MediaRecorder.OutputFormat.THREE_GPP);
            mediaRecorder.setOutputFile(getRecordingFilePath());

mediaRecorder.setAudioEncoder(MediaRecorder.AudioEncoder.AMR_NB);
            mediaRecorder.prepare();
            mediaRecorder.start();
            Toast.makeText(this,"recording is
started",Toast.LENGTH_LONG).show();
        }catch(Exception e){
            e.printStackTrace();
        }
    }
    public void btnStopPressed(View v){

        mediaRecorder.stop();
        mediaRecorder.release();
        mediaRecorder=null;
        Toast.makeText(this,"recording is
stopped",Toast.LENGTH_LONG).show();

    }
    public void btnPlayPressed(View v){

        try {
            mediaPlayer = new MediaPlayer();
            mediaPlayer.setDataSource(getRecordingFilePath());
            mediaPlayer.prepare();
            mediaPlayer.start();
            Toast.makeText(this,"recording is
Playing",Toast.LENGTH_LONG).show();
        }catch(Exception e){
            e.printStackTrace();
        }
    }
    private boolean isMicroPhonePresent() {
```

```
            if
        (this.getPackageManager().hasSystemFeature(PackageManager.FEATURE_MICR
        OPHONE)) {
                return true;
            } else {
                return false;
            }
        }
            private void getMicroPhonePermission(){

                if(ContextCompat.checkSelfPermission(this,

        android.Manifest.permission.RECORD_AUDIO)==PackageManager.PERMISSION_D
        ENIED) {
                        ActivityCompat.requestPermissions(
                                this, new
        String[]{Manifest.permission.RECORD_AUDIO},
        MICROPHONE_PERMISSION_CODE);
                }
            }
            public String getRecordingFilePath(){
                ContextWrapper contextWrapper=new
        ContextWrapper(getApplicationContext());
                File
        musicDirectory=contextWrapper.getExternalFilesDir(Environment.DIRECTOR
        Y_MUSIC);
                File file=new
        File(musicDirectory,"TestRecordingFile"+".MP3");
                return file.getPath();
            }


        }
```

OutPut:



Practical 14 A: Write an android program to develop Video Recording Application

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/button"
        android:layout_width="153dp"
        android:layout_height="68dp"
        android:text="Record Video"
        android:onClick="recordVideoButtonPressed"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.435"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.499" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

AndroidManifest.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
    <uses-permission android:name="android.permission.CAMERA"/>
    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/Theme.VideoRecording"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

</manifest>
```

MainActivity.java

```
package com.example.videorecording;

import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.net.Uri;
import android.os.Bundle;
import android.provider.MediaStore;
import android.util.Log;
import android.view.View;

import java.net.URI;

public class MainActivity extends AppCompatActivity {

    private static  int CAMERA_PERMISSION_CODE=100;
    private static int VIDEO_RECORD_CODE=101;
    private Uri videoPath;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        if(isCameraPresentInPhone()){
            Log.i("VIDEO_RECORD_TAG","CAMERA IS DETECTED");
            getCameraPermission();
        }else{
            Log.i("VIDEO_RECORD_TAG","CAMERA IS NOT DETECTED");
                    }

    }
    public void recordVideoButtonPressed(View v){

        recordVideo();
    }
    private boolean isCameraPresentInPhone(){


if(getPackageManager().hasSystemFeature(PackageManager.FEATURE_CAMERA_ANY)){

            return  true;
        }else{
            return false;
        }
    }
    private void getCameraPermission(){

        if(ContextCompat.checkSelfPermission(this, Manifest.permission.CAMERA)
                ==PackageManager.PERMISSION_DENIED){

            ActivityCompat.requestPermissions(this,new String[]
                    {Manifest.permission.CAMERA},CAMERA_PERMISSION_CODE);
        }
```

```
    }
    private void recordVideo(){

        Intent intent=new Intent(MediaStore.ACTION_VIDEO_CAPTURE);
        startActivityForResult(intent,VIDEO_RECORD_CODE);
    }
    @Override
    protected void onActivityResult(int requestCode, int resultCode, @Nullable
Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        if(requestCode==VIDEO_RECORD_CODE){
            if(resultCode==RESULT_OK){
                videoPath=data.getData();
                Log.i("VIDEO_RECORD_TAG","Video is Recorded and available
path"+videoPath);
            } else if (resultCode==RESULT_CANCELED) {

                Log.i("VIDEO_RECORD_TAG"," Recorded video is
cancelled"+videoPath);
            }
            else{
                Log.i("VIDEO_RECORD_TAG"," Recorded video has got Some error");
            }
        }
    }
}
```

Output:

Practical 14 B: Write an android program to develop Camera Application

Step 1: Create a new project in Android Studio and name it CameraCodeExample

Step 2: Open res -> layout -> xml or (activity_main.xml) and add following code Here I have used frame layout to load my fragments.

```
<?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">

    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="150dp"
        android:layout_marginTop="200dp"
        android:onClick="CameraButton"
        android:text="Camera" />
</RelativeLayout>
```

AndroidMainfest.xml:

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

    <uses-permission android:name="android.permission.CAMERA"/>
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/Theme.CameraApp"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

</manifest>
```

MainActivity.java

```
package com.example.cameraapp;

import static android.Manifest.permission.CAMERA;
import static android.Manifest.permission.WRITE_EXTERNAL_STORAGE;

import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;

import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.net.Uri;
import android.os.Bundle;
import android.os.Environment;
import android.os.StrictMode;
import android.provider.MediaStore;
import android.view.View;

import java.io.File;
import java.io.IOException;

public class MainActivity extends AppCompatActivity {
    public static int index=0;
    public final String directory=
Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_PICTURES)+"/myCam
era";
/* here myCamera is a folder that is created under emulator internal
storage(pictures==>myCamera) to save images which is captured from camera app */
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        ActivityCompat.requestPermissions
                (this,new String[]{CAMERA, WRITE_EXTERNAL_STORAGE},
PackageManager.PERMISSION_GRANTED);
//execute the code without having below code if any error then uncomment this code
        /* StrictMode.VmPolicy.Builder builder=new StrictMode.VmPolicy.Builder();
        StrictMode.setVmPolicy(builder.build()); */

    }
    public void CameraButton(View v){
        index++;
        String file=directory+index+".jpg";
        File newFile=new File(file);
        try {
            newFile.createNewFile();
        } catch (IOException e) {
            throw new RuntimeException(e);
        }
        Uri outputFileUri= Uri.fromFile(newFile);
        Intent cameraIntent=new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
        cameraIntent.putExtra(MediaStore.EXTRA_OUTPUT,outputFileUri);
        startActivity(cameraIntent);
    }
}
Output:
```

•

Practical 15A: Create an android application to get latitude and longitude value by using location service:

Step1: create a new project with name locationExample and implement the following code in

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">

    <fragment
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/showMap"
        android:name="com.google.android.gms.maps.SupportMapFragment"/>

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

***

Step2:

Now create one API Key for google maps service by the following steps

- Before implement the following steps should check whether you logged in you system with your mailid only.
- In google search type console developer---->click on google developer console---->now click on create project---->now enter you project name-->click on create---->now click enable APIs and Services-->now click on maps SDK for Android---->now click on enable---->now click on credential which is at leftside panel--->now click on create credentials---->select API key--->now here one key is generated copy this to use in your project where required and click on close---->now click on created APIKey1--->now check the box under Applications restriction as none----> then come out from this website

Step3:

- The module build.gradle file includes the following map dependency which is required by the maps SDK for android.

```
dependencies {
  implementation 'com.google.android.gms:play-services-maps:18.1.0'
  }
```

build.gradle (Module: app):

```
plugins {
    id 'com.android.application'
}

android {
    namespace 'com.example.locationexample'
    compileSdk 33

    defaultConfig {
        applicationId "com.example.locationexample"
        minSdk 24
        targetSdk 33
        versionCode 1
        versionName "1.0"

        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }

    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-
optimize.txt'), 'proguard-rules.pro'
        }
    }
    compileOptions {
        sourceCompatibility JavaVersion.VERSION_1_8
        targetCompatibility JavaVersion.VERSION_1_8
    }
}

dependencies {

    implementation 'androidx.appcompat:appcompat:1.6.1'
    implementation 'com.google.android.material:material:1.9.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.1.4'
    testImplementation 'junit:junit:4.13.2'
    androidTestImplementation 'androidx.test.ext:junit:1.1.5'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.5.1'
    implementation 'com.google.android.gms:play-services-maps:18.1.0'
}
```

Step4: now under res---->values---->string.xml file add the following string with generated API key

```
<resources>
  <string name="app_name">LocationExample</string>
 <string name="my_map_api_key">AIzaSyBSHaytlK-VWWmVn_hGmpxxoTAa1BYqdLQ</string>
</resources>
```

Note: here highlated string is pasted from which we have created API key in the above step

Step5:

- **Now in your AndroidManifest.xml file add the following meta data**

```
<meta-data
    android:name="com.google.android.geo.API_KEY"
    android:value="@string/my_map_api_key" />
```

AndroidManifest.xml

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

    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.INTERNET" />

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/Theme.LocationExample"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

        <meta-data
            android:name="com.google.android.geo.API_KEY"
            android:value="@string/my_map_api_key" />
    </application>

</manifest>
```

Step6: now add the following lines in the android_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">

    <fragment
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/showMap"
        android:name="com.google.android.gms.maps.SupportMapFragment"/>
```

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

Sete7: Now write the code in the following MainActivity.java

```
package com.example.locationexample;

import android.os.Bundle;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;

public class MainActivity extends AppCompatActivity implements OnMapReadyCallback {
    private GoogleMap myMap;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        SupportMapFragment mapFragment= (SupportMapFragment)
getSupportFragmentManager().findFragmentById(R.id.showMap);
        mapFragment.getMapAsync(this);
    }

    @Override
    public void onMapReady(@NonNull GoogleMap googleMap) {
        myMap=googleMap;
        LatLng Nandyal=new LatLng(15.4800 ,78.4800);
        myMap.addMarker(new MarkerOptions().position(Nandyal).title("Nandyal"));
        myMap.moveCamera(CameraUpdateFactory.newLatLng(Nandyal));

    }
}
```

Output:

Practical 15B: Create an android application to display X, Y Sensor values by using Sensor Service.

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:gravity="center"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Sensor Values"
        android:id="@+id/txtValues"
        android:textSize="34sp"
        android:textStyle="bold"/>


</LinearLayout>
```

MainActivity.java

```
package com.example.sensorsexample;

import androidx.appcompat.app.AppCompatActivity;

import android.hardware.Sensor;
import android.hardware.SensorEvent;
import android.hardware.SensorEventListener;
import android.hardware.SensorManager;
import android.os.Bundle;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity implements SensorEventListener {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        SensorManager sensorManager=(SensorManager) getSystemService(SENSOR_SERVICE);
        if(sensorManager!=null){
            Sensor
accleroSensor=sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
            if(accleroSensor!=null){

sensorManager.registerListener(this,accleroSensor,SensorManager.SENSOR_DELAY_NORMAL);
            }

        }else{
            Toast.makeText(this,"sensor service is not
detected",Toast.LENGTH_LONG).show();
        }
    }
```

```
        @Override
        public void onSensorChanged(SensorEvent event) {
            if (event.sensor.getType() == Sensor.TYPE_ACCELEROMETER) {
                ((TextView)findViewById(R.id.txtValues)).
                        setText("X:"+event.values[0]+", Y: "+event.values[1]+", Z:
    "+event.values[2]);
            }
        }

        @Override
        public void onAccuracyChanged(Sensor sensor, int accuracy) {

        }
    }
```

Output:

Practical 16A: Create an android application to get the notifications on Notification Bar by Using Notification Service.

Step1: Take one array disk image and save it into drawable folder with extension of PNG.

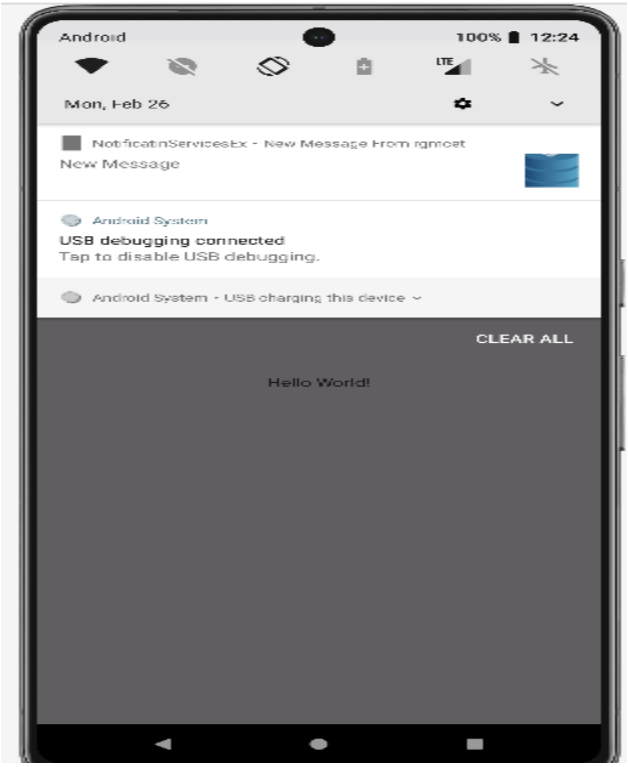
Example:

Step2: write the following code in MainActivity.java

```
package com.example.notificatinservicesex;

import androidx.appcompat.app.AppCompatActivity;
import androidx.core.content.res.ResourcesCompat;

import android.app.Notification;
import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.graphics.Bitmap;
import android.graphics.drawable.BitmapDrawable;
import android.graphics.drawable.Drawable;
import android.os.Bundle;

public class MainActivity extends AppCompatActivity {
    private static final  String CHANNEL_ID="My Channel";
    private static final  int NOTIFICATION_ID=100;

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

        Drawable drawable=
ResourcesCompat.getDrawable(getResources(),R.drawable.db,null);
        BitmapDrawable bitmapDrawable=(BitmapDrawable)drawable;
        Bitmap largeIcon=bitmapDrawable.getBitmap();
        Notification notification;
        NotificationManager nm= (NotificationManager)
getSystemService(NOTIFICATION_SERVICE);
        if (android.os.Build.VERSION.SDK_INT >=
android.os.Build.VERSION_CODES.O) {
           notification=
                    new Notification.Builder(this).

setLargeIcon(largeIcon).setSmallIcon(R.drawable.db)
                            .setContentText("New Message").
```

```
                                    setSubText("New Message From rgmcet")
                                    .setChannelId(CHANNEL_ID)
                                    .build();
                    nm.createNotificationChannel
                            (new NotificationChannel(CHANNEL_ID,"New
        Channel",NotificationManager.IMPORTANCE_HIGH));
                }else {
                    notification=
                            new Notification.Builder(this).

        setLargeIcon(largeIcon).setSmallIcon(R.drawable.db)
                                    .setContentText("New Message").
                                    setSubText("New Message From rgmcet")
                                    .build();
                }
                nm.notify(NOTIFICATION_ID,notification);
            }
        }
```

OutPut:



Practical 16B: Create an android application to display available Wi-Fi devices and Paired Wi-Fi devices by using Wi Fi Service

Step1: create a new project with name wifiExample and design the following code under 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"
    tools:context=".MainActivity"
    android:orientation="vertical">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Available Wifi Networks!"
        android:textAlignment="center"
        android:layout_marginTop="20sp"
        android:textStyle="bold"
        android:textSize="30dp"/>
    <ListView
        android:layout_width="match_parent"
        android:layout_height="fill_parent"
        android:layout_margin="20sp"
        android:id="@+id/myListView"/>

</LinearLayout>
```

Step2: create a java class under the package of com.example.wifiexample with name ListAdapter.java and the following code

```
package com.example.wifiservice;

import android.content.Context;
import android.net.wifi.ScanResult;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.TextView;

import java.util.List;

public class ListAdapter extends BaseAdapter {
    Context context;
    LayoutInflater inflater;
```

```
        List<ScanResult> wifilist;

        public ListAdapter(Context context,List<ScanResult>wifilist) {
            this.context=context;
            this.wifilist=wifilist;
            inflater=(LayoutInflater)
    context.getSystemService(context.LAYOUT_INFLATER_SERVICE);
        }

        @Override
        public int getCount() {
            return wifilist.size();
        }

        @Override
        public Object getItem(int i) {
            return null;
        }

        @Override
        public long getItemId(int i) {
            return 0;
        }

        @Override
        public View getView(int i, View view, ViewGroup viewGroup) {
            Holder holder;
            View view1=view;
            if(view1==null){
                view1=inflater.inflate(R.layout.list_item,null);
                holder=new Holder();
                holder.tvDetails=(TextView)
    view1.findViewById(R.id.txtWifiName);
                view1.setTag(holder);
            }else {
                holder=(Holder) view1.getTag();
            }
            holder.tvDetails.setText(wifilist.get(i).SSID);
            return view1;
        }
        class Holder{
            TextView tvDetails;
        }
    }
```

step3: create one new layout file with name list_item.xml with the following code(res---->right click on layout--->select layout resource file and name it as list_item.xml)

list_item.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/txtWifiName"
        android:textSize="20sp"
        android:textStyle="bold"
        android:textColor="#FF5722"/>

</RelativeLayout>
```

Step4:MainActivity.java

```
package com.example.wifiservice;

import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import android.Manifest;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.IntentFilter;
import android.content.pm.PackageManager;
import android.net.wifi.WifiManager;
import android.os.Bundle;
import android.widget.ListView;

import java.util.List;

public class MainActivity extends AppCompatActivity {
    WifiManager wifiManager;
    WifiReceiver wifiReceiver;
    ListAdapter listAdapter;
    ListView wifiList;
    List mywifiList;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        wifiList = (ListView) findViewById(R.id.myListView);
        wifiManager = (WifiManager)
getSystemService(Context.WIFI_SERVICE);
        wifiReceiver = new WifiReceiver();
        registerReceiver(wifiReceiver, new
IntentFilter(WifiManager.SCAN_RESULTS_AVAILABLE_ACTION));
        if (ContextCompat.checkSelfPermission(getApplicationContext(),
Manifest.permission.ACCESS_FINE_LOCATION) !=
```

```
        PackageManager.PERMISSION_GRANTED) {
                    ActivityCompat.requestPermissions(this, new
        String[]{Manifest.permission.ACCESS_FINE_LOCATION}, 0);
                } else {
                    scanWifiList();
                }
            }

            private void scanWifiList() {
                wifiManager.startScan();
                if (ActivityCompat.checkSelfPermission(this,
        Manifest.permission.ACCESS_FINE_LOCATION) !=
        PackageManager.PERMISSION_GRANTED) {
                    // TODO: Consider calling
                    //    ActivityCompat#requestPermissions
                    // here to request the missing permissions, and then
        overriding
                    //   public void onRequestPermissionsResult(int
        requestCode, String[] permissions,
                    //                                          int[]
        grantResults)
                    // to handle the case where the user grants the
        permission. See the documentation
                    // for ActivityCompat#requestPermissions for more details.
                    return;
                }
                mywifiList = wifiManager.getScanResults();
                setAdapter();
            }

            private void setAdapter() {
                listAdapter=new
        ListAdapter(getApplicationContext(),mywifiList);
                wifiList.setAdapter(listAdapter);
            }

            class WifiReceiver extends BroadcastReceiver {

                @Override
                public void onReceive(Context context, Intent intent) {

                }
            }
        }
```

Step5: AndroidManifest.xml

```
        <?xml version="1.0" encoding="utf-8"?>
        <manifest xmlns:android="http://schemas.android.com/apk/res/android"
            xmlns:tools="http://schemas.android.com/tools">
            <uses-permission
        android:name="android.permission.ACCESS_FINE_LOCATION"/>
            <uses-permission
        android:name="android.permission.ACCESS_COARSE_LOCATION"/>
```

```
    <uses-permission
android:name="android.permission.ACCESS_WIFI_STATE" />
    <uses-permission
android:name="android.permission.CHANGE_WIFI_STATE"/>


    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.WifiService"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

</manifest>
```

Ouput:

Practical 17A: Create an android application to get the Bluetooth devices and list of devices using Bluetooth and Vibrator Service.

Activity_main.xml

Note: create two images for Bluetooth connected & disable using vector asset and save in drawable folder with name: bluetooth_connected & Bluetooth_disabled

```
<?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"
    tools:context=".MainActivity"
    android:orientation="vertical"
    android:padding="20sp">

    <ImageView
        android:layout_width="match_parent"
        android:layout_height="150dp"
        android:src="@drawable/bluetooth_disabled"
        android:id="@+id/img"/>
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="TURN ON"
        android:backgroundTint="#6CCDB2"
        android:textColor="@color/black"
        android:textStyle="bold"
        android:textSize="20sp"
        android:layout_gravity="center"
        android:layout_marginTop="10sp"
        android:id="@+id/btnon"/>
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="TURN OFF"
        android:backgroundTint="#CD746C"
        android:textColor="@color/black"
        android:textStyle="bold"
        android:textSize="20sp"
        android:layout_gravity="center"
        android:layout_marginTop="10sp"
        android:id="@+id/btnoff"/>
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="DISCOVERABLE"
        android:backgroundTint="#B86CCD"
        android:textColor="@color/black"
        android:textStyle="bold"
```

```
        android:textSize="20sp"
        android:layout_gravity="center"
        android:layout_marginTop="10sp"
        android:id="@+id/btndis"/>
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="GET PAIRED LIST"
        android:backgroundTint="#FDD08D"
        android:textColor="@color/black"
        android:textStyle="bold"
        android:textSize="20sp"
        android:layout_gravity="center"
        android:layout_marginTop="10sp"
        android:id="@+id/btnlist"/>
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/txtlist"/>

</LinearLayout>
```

MainActivity.java

```
package com.example.bluetoothex;

import androidx.appcompat.app.AppCompatActivity;

import android.annotation.SuppressLint;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import java.util.Set;

public class MainActivity extends AppCompatActivity {
    Button btnon,btnoff,btndis,btnlis;
    TextView txtlist;
    ImageView img;
    BluetoothAdapter bluetoothAdapter;
    private static final int Request_Enable=0;
    private static final int Request_Discover=1;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        btnon=findViewById(R.id.btnon);
        btnoff=findViewById(R.id.btnoff);
        btndis=findViewById(R.id.btndis);
```

```
        btnlis=findViewById(R.id.btnlist);
        txtlist=findViewById(R.id.txtlist);
        img=findViewById(R.id.img);
        bluetoothAdapter=BluetoothAdapter.getDefaultAdapter();
        if(bluetoothAdapter.isEnabled()){
            img.setImageResource(R.drawable.bluetooth_connected);
        }else {
            img.setImageResource(R.drawable.bluetooth_disabled);
        }
        btnon.setOnClickListener(new View.OnClickListener() {
            @SuppressLint("MissingPermission")
            @Override
            public void onClick(View view) {
                if(!bluetoothAdapter.isEnabled()){
                    Toast.makeText(MainActivity.this,"Turning
on...",Toast.LENGTH_LONG).show();
                    Intent intent=new
Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
                    startActivityForResult(intent,Request_Enable);
                    img.setImageResource(R.drawable.bluetooth_connected);
                }
                else {
                    Toast.makeText(MainActivity.this,"Already
on...",Toast.LENGTH_LONG).show();
                }
            }
        });
        btndis.setOnClickListener(new View.OnClickListener() {
            @SuppressLint("MissingPermission")
            @Override
            public void onClick(View view) {
                if(!bluetoothAdapter.isDiscovering()){
                    Toast.makeText(MainActivity.this,"Make Your Phone
Discoverable...",Toast.LENGTH_LONG).show();
                    Intent intent=new
Intent(BluetoothAdapter.ACTION_REQUEST_DISCOVERABLE);
                    startActivityForResult(intent,Request_Discover);
                }
            }
        });
        btnoff.setOnClickListener(new View.OnClickListener() {
            @SuppressLint("MissingPermission")
            @Override
            public void onClick(View view) {
                if(bluetoothAdapter.isEnabled()){
                    bluetoothAdapter.disable();
                    Toast.makeText(MainActivity.this,"Turned
off...",Toast.LENGTH_LONG).show();
                    img.setImageResource(R.drawable.bluetooth_disabled);
                }
                else {
                    Toast.makeText(MainActivity.this,"Already
off...",Toast.LENGTH_LONG).show();
                }
            }
        });
```

```
            btnlis.setOnClickListener(new View.OnClickListener() {
                @SuppressLint("MissingPermission")
                @Override
                public void onClick(View view) {
                    if(bluetoothAdapter.isEnabled()){
                        txtlist.setText("List of All Devices: ");
                        @SuppressLint("MissingPermission")
                        Set<BluetoothDevice>
deviceset=bluetoothAdapter.getBondedDevices();
                        for(BluetoothDevice device:deviceset){

txtlist.append("\nDevice"+device.getName()+":"+device);
                        }
                    }else {
                        Toast.makeText(MainActivity.this,"Please Turn on Your
Bluetooth...",Toast.LENGTH_LONG).show();
                    }
                }
            });
        }
    }
```

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">
    <uses-permission android:name="android.permission.BLUETOOTH" />
    <uses-permission android:name="android.permission.BLUETOOTH_ADMIN"/>
    <uses-permission android:name="android.permission.BLUETOOTH_CONNECT"
/>

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

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.BluetoothEx"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

```
</manifest>
```

Output:

Example: Vibrator Service

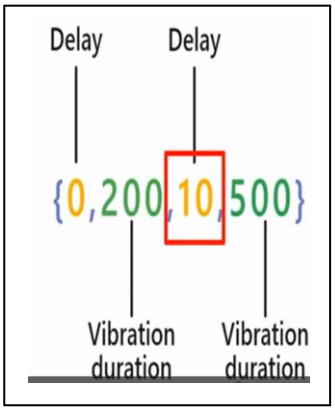
activity_main.xml

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

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="START"
        android:id="@+id/start"/>
    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="STOP"
        android:id="@+id/stop"
        android:layout_marginTop="20sp"/>

</LinearLayout>
```

MainActivity.java:

```
package com.example.vibratorservice;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Build;
import android.os.Bundle;
import android.os.VibrationEffect;
import android.os.Vibrator;
import android.view.View;
import android.widget.Button;

public class MainActivity extends AppCompatActivity {
    Button start,stop;
    Vibrator vibrator;

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

        vibrator=(Vibrator)getSystemService(VIBRATOR_SERVICE);
        start.setOnClickListener(new View.OnClickListener() {
            @Override
```

```
                    public void onClick(View view) {
                        if(!vibrator.hasVibrator()){
                            return;
                        }

                        if(Build.VERSION.SDK_INT >= Build.VERSION_CODES.O){


            vibrator.vibrate(VibrationEffect.createOneShot(1000,VibrationEffect.
            DEFAULT_AMPLITUDE));

                        }else {
                            long[] parttern={0,200,10,500};
                            vibrator.vibrate(parttern,-1);
                        }


                    }
                });
                stop.setOnClickListener(new View.OnClickListener() {
                    @Override
                    public void onClick(View view) {
                        vibrator.cancel();
                    }
                });
            }

        }
```



AndroidManifest.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">
    <uses-permission android:name="android.permission.VIBRATE"/>
    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.VibratorService"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

</manifest>
```

Practical 17 B: Create android application to get the system announcements by using Broadcast Receiver

Step1: First create a new project with name BroadcastSenderApp and write the following code for xml & java

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/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Send Broadcast Message"
        android:onClick="onBroadcastSendBtnClicked"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java

```
package com.example.systemanouncementsexample;

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 onBroadcastSendBtnClicked(View v){

        Intent intent=new Intent();
        intent.setAction("com.rgmcet.myBroadcastMessage");
        intent.setFlags(Intent.FLAG_INCLUDE_STOPPED_PACKAGES);
        sendBroadcast(intent);
    }
}
```

Step2: Now create another new project with name BroadcastReceiverApp

Step3: Now create on create one BroadcastReceiver class(right click on com.example.BroadcastReceiverApp package--->click on new----->click on other-->select Broadcast Receiver and give the name MyBroadcastReceiver and click on finish(note after creating this class one receiver component is added in Android manifest.xml file now modify this by using following code)
AndroidManifest.xml

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

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/Theme.BroadcastReceiverApp"
        tools:targetApi="31">
        <receiver
            android:name=".MyBroadcastReceiver"
            android:enabled="true"
            android:exported="true">
            <intent-filter>
                <action android:name="com.rgmcet.myBroadcastMessage">
                </action>
            </intent-filter>
        </receiver>

        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

</manifest>
```

MyActivity.java

```
package com.example.broadcastreceiverapp;

import androidx.appcompat.app.AppCompatActivity;

import android.content.IntentFilter;
import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

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

```
        IntentFilter intentFilter=new
    IntentFilter("com.rgmcet.myBroadcastMessage");
        MyBroadcastReceiver myBroadcastReceiver=new MyBroadcastReceiver();
        registerReceiver(myBroadcastReceiver,intentFilter);
    }
}
```

MyBroadcastReceiver.java

```
package com.example.broadcastreceiverapp;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.util.Log;
import android.widget.Toast;

public class MyBroadcastReceiver extends BroadcastReceiver {

    @Override
    public void onReceive(Context context, Intent intent) {
        // TODO: This method is called when the BroadcastReceiver is receiving
        // an Intent broadcast.
        Log.i("BroadcastReceiver","My Broadcase Message is recieved");
        Toast.makeText(context," My Broadcase Message is recieved
",Toast.LENGTH_LONG).show();
    }
}
```

Execution Process:

Now execute first execute BroadcastSenderApp project then after execute BroadcastReceiverApp and stop emulator and again opent the BroadcastSenderApp and click button in emulator and observe that "broadcast message is received".

Step4: now do some changes in second project and implement actual system broadcast messages by using following changes:

AndroidManifest.xml:

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

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/Theme.BroadcastReceiverApp"
        tools:targetApi="31">
        <receiver
            android:name=".MyBroadcastReceiver"
            android:enabled="true"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.BATTERY_LOW">
                </action>
            </intent-filter>
        </receiver>
```

```
                    <activity
                        android:name=".MainActivity"
                        android:exported="true">
                        <intent-filter>
                            <action android:name="android.intent.action.MAIN" />

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

            </manifest>
```

MainActivity.java

```
package com.example.broadcastreceiverapp;

import androidx.appcompat.app.AppCompatActivity;

import android.content.IntentFilter;
import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        IntentFilter intentFilter=new
IntentFilter("android.intent.action.BATTERY_LOW");
        MyBroadcastReceiver myBroadcastReceiver=new MyBroadcastReceiver();
        registerReceiver(myBroadcastReceiver,intentFilter);
    }
}
```

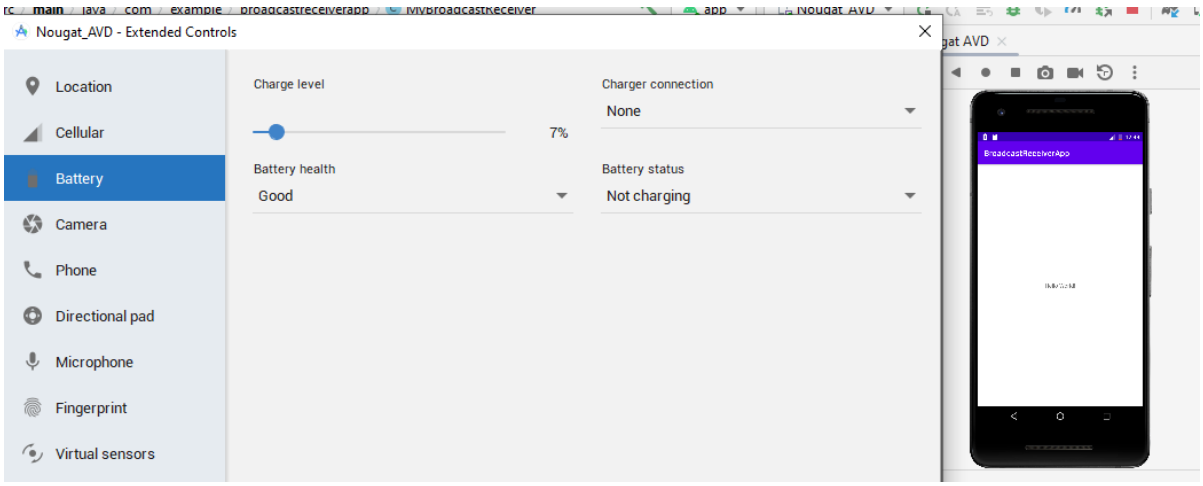
MyBroadcastReceiver.java

```
package com.example.broadcastreceiverapp;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.util.Log;
import android.widget.Toast;

public class MyBroadcastReceiver extends BroadcastReceiver {

    @Override
    public void onReceive(Context context, Intent intent) {
        // TODO: This method is called when the BroadcastReceiver is receiving
        // an Intent broadcast.
        Log.i("MyBroadcastReceiver","Your Battery is Low");
        Toast.makeText(context,"Your Battery is Low",Toast.LENGTH_LONG).show();
    }
}
```

Execution:

- Now no need execute BroadcastSenderApp and because android operating system itself send a message so just execute BroadcastReceiverApp and

observe in the emulator by changing charging level from 100% to low using extended control in the emulator



Practical 18: Create an android application to share the data between multiple applications by using Content Provider

Activity_main.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">

   <ListView
       android:layout_width="match_parent"
       android:layout_height="wrap_content"
       android:id="@+id/list"/>

</RelativeLayout>
```

AndroidManifest.xml

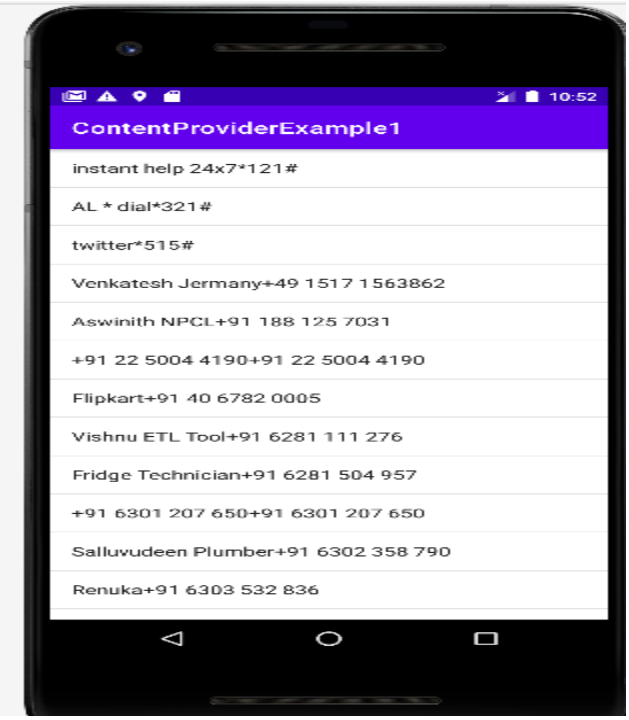
```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">
    <uses-permission android:name="android.permission.READ_CONTACTS"/>
    <uses-permission android:name="android.permission.WRITE_CONTACTS"/>

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/Theme.ContentProviderExample1"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

</manifest>
```

MainActivity.java

```
package com.example.contentproviderexample1;

import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;
```

```
import android.Manifest;
import android.content.ContentResolver;
import android.content.pm.PackageManager;
import android.database.Cursor;
import android.net.Uri;
import android.os.Bundle;
import android.provider.ContactsContract;
import android.widget.ArrayAdapter;
import android.widget.ListView;

import java.util.ArrayList;

public class MainActivity extends AppCompatActivity {
    ListView list;
    ArrayList<String>listdata;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        list=findViewById(R.id.list);
        listdata=new ArrayList<String>();
        fetchContact();
    }

    private void fetchContact() {
        if (ContextCompat.checkSelfPermission
                (this, Manifest.permission.READ_CONTACTS) !=
PackageManager.PERMISSION_GRANTED) ;
        ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.READ_CONTACTS}, 0);
    ContentResolver resolver = getContentResolver();
    Uri uri = ContactsContract.CommonDataKinds.Phone.CONTENT_URI;
    String[] projection = null;
    String selection = null;
    String[] selectionargs = null;
    String order = null;
    Cursor cursor = resolver.query(uri, projection, selection, selectionargs,
order);
    if(cursor.getCount()>0){
        while(cursor.moveToNext()){
            String
name=cursor.getString(cursor.getColumnIndex(ContactsContract.CommonDataKinds.Pho
ne.DISPLAY_NAME));
            String
number=cursor.getString(cursor.getColumnIndex(ContactsContract.CommonDataKinds.P
hone.NUMBER));
            String fullContact=name+number;
            listdata.add(fullContact);
        }
    }
        ArrayAdapter<String>adapter=new ArrayAdapter<String>(this,
android.R.layout.simple_list_item_1,listdata);
        list.setAdapter(adapter);
    }

}
```

Output:


10:52
ContentProviderExample1
instant help 24x7*121#
AL * dial*321#
twitter*515#
Venkatesh Jermany+49 1517 1563862
Aswinith NPCL+91 188 125 7031
+91 22 5004 4190+91 22 5004 4190
Flipkart+91 40 6782 0005
Vishnu ETL Tool+91 6281 111 276
Fridge Technician+91 6281 504 957
+91 6301 207 650+91 6301 207 650
Salluvudeen Plumber+91 6302 358 790
Renuka+91 6303 532 836

Practical 19: Create an android application to display different Dialog Boxes.

Alert Dialogbox:

- They are used to help users answer questions, make selections,confirm actions, and read warning or error messages
- It is a window that partially obscures the activity that launched it.

Example1: AlerDialogBox

Activity_main.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"
    android:paddingBottom="16dp"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Name: "
        android:id="@+id/txtName"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:layout_marginBottom="71dp"
        android:layout_above="@+id/btnAcc"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"/>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="AccNo : "
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:id="@+id/txtAcc"
        android:layout_above="@+id/btnBal"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_marginBottom="86dp"/>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Balance: "
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:id="@+id/txtBal"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_alignParentBottom="true"
```

```
            android:layout_marginBottom="140dp"/>
        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Show"
            android:id="@+id/btnName"
            android:layout_alignBottom="@+id/txtName"
            android:layout_alignParentRight="true"
            android:layout_alignParentEnd="true"/>
        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Show"
            android:id="@+id/btnAcc"
            android:layout_alignBottom="@+id/txtAcc"
            android:layout_alignLeft="@+id/btnName"
            android:layout_alignStart="@id/btnName"/>
        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Show"
            android:id="@+id/btnBal"
            android:layout_alignBottom="@+id/txtBal"
            android:layout_alignParentRight="true"
            android:layout_alignParentEnd="true"/>


    </RelativeLayout>
```

MainActivity.java

```
package com.example.alertdialogueboxexample;

import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;

import android.content.DialogInterface;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {
    TextView txtName,txtAcc,txtBal;
    Button btnName,btnAcc,btnBal;

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

        btnName=findViewById(R.id.btnName);
        btnAcc=findViewById(R.id.btnAcc);
        btnBal=findViewById(R.id.btnBal);

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

```
                txtName.setText("Ramesh");
            }
        });
        btnAcc.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                txtAcc.setText("SBI6709");

            }
        });
        btnBal.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                AlertDialog.Builder builder=new
AlertDialog.Builder(MainActivity.this);
                builder.setMessage("Are You sure").setPositiveButton("ok",
new DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int which) {
                        txtBal.setText("5800");
                    }
                }).setNegativeButton("Cancel",null);
                AlertDialog alert=builder.create();
                alert.show();

            }
        });
    }
    @Override
    public void onBackPressed(){
        AlertDialog.Builder builder=new
AlertDialog.Builder(MainActivity.this);
        builder.setTitle("Really Exit").
                setMessage("Are You sure").setPositiveButton("ok", new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                MainActivity.super.onBackPressed();
            }
        }).setNegativeButton("Cancel",null).setCancelable(false);
        AlertDialog alert=builder.create();
        alert.show();

    }

}
```

Output:

Practical 20: Create an android application to display current location on Google maps by using Google-Maps Service

Step1: Add the following implementations in build.gradle and click on syncnow and test whether properly sysnced or not

Build.gradle(Module:App)

```
plugins {
    id 'com.android.application'
}

android {
    namespace 'com.example.currentlocationexample'
    compileSdk 33

    defaultConfig {
        applicationId "com.example.currentlocationexample"
        minSdk 24
        targetSdk 33
        versionCode 1
        versionName "1.0"

        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }

    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-
optimize.txt'), 'proguard-rules.pro'
        }
    }
    compileOptions {
        sourceCompatibility JavaVersion.VERSION_1_8
        targetCompatibility JavaVersion.VERSION_1_8
    }
}

dependencies {

    implementation 'androidx.appcompat:appcompat:1.6.1'
    implementation 'com.google.android.material:material:1.9.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.1.4'
    testImplementation 'junit:junit:4.13.2'
    androidTestImplementation 'androidx.test.ext:junit:1.1.5'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.5.1'
    implementation 'com.google.android.gms:play-services-maps:18.1.0'
    implementation 'com.google.android.gms:play-services-location:21.0.1'
    implementation 'com.karumi:dexter:6.2.1'
}
```

Now create one API Key for google maps service by the following steps

- Before implement the following steps should check whether you logged in you system with your mailid only.

- In google search type console developer---->click on google developer console---->now click on create project---->now enter you project name-->click on create---->now click enable APIs and Services-->now click on maps SDK for Android---->now click on enable---->now click on credential which is at leftside panel--->now click on create credentials---->select API key--->now here one key is generated copy this to use in your project where required and click on close---->now click on created APIKey1--->now check the box under Applications restriction as none----> then come out from this website

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">
    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/Theme.CurrentLocationExample"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <meta-data
            android:name="com.google.android.geo.API_KEY"
            android:value="AIzaSyCpXvnVCXNDjieNZK2Cr-YA7JBLEhcINgM"/>
    </application>

</manifest>
```

Activity_main.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">

   <fragment
       android:id="@+id/goole_map"
```

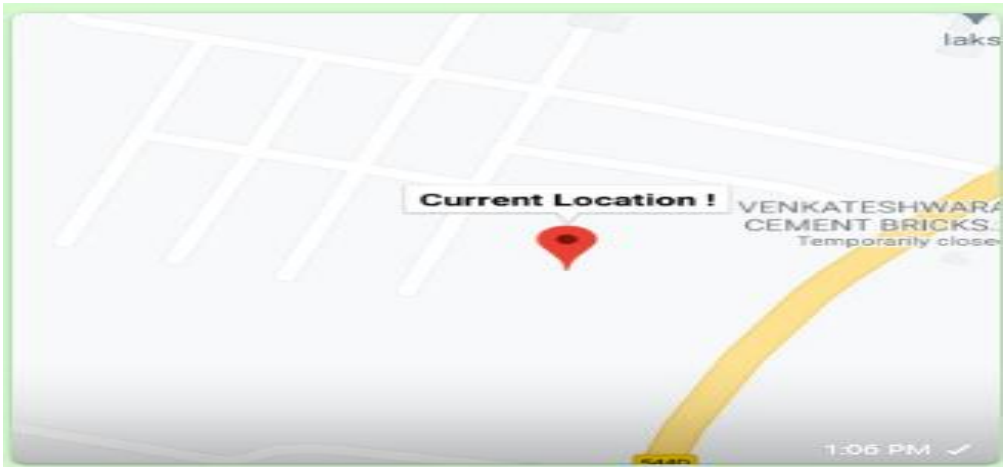
```
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:name="com.google.android.gms.maps.SupportMapFragment"/>

    </RelativeLayout>
```

MainActivity.java

```
package com.example.currentlocationexample;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;

import android.Manifest;
import android.content.pm.PackageManager;
import android.location.Location;
import android.os.Bundle;
import android.view.WindowManager;
import android.widget.Toast;

import com.google.android.gms.location.FusedLocationProviderClient;
import com.google.android.gms.location.LocationServices;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import com.karumi.dexter.Dexter;
import com.karumi.dexter.PermissionToken;
import com.karumi.dexter.listener.PermissionDeniedResponse;
import com.karumi.dexter.listener.PermissionGrantedResponse;
import com.karumi.dexter.listener.PermissionRequest;
import com.karumi.dexter.listener.single.PermissionListener;

public class MainActivity extends AppCompatActivity {
    SupportMapFragment supportMapFragment;
    FusedLocationProviderClient fusedLocationProviderClient;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
WindowManager.LayoutParams.SOFT_INPUT_MASK_ADJUST);
        supportMapFragment = (SupportMapFragment)
getSupportFragmentManager().findFragmentById(R.id.goole_map);
        fusedLocationProviderClient = (FusedLocationProviderClient)
LocationServices.getFusedLocationProviderClient(this);

Dexter.withContext(getApplicationContext()).withPermission(Manifest.permission.A
CCESS_FINE_LOCATION)
                .withListener(new PermissionListener() {
                    @Override
                    public void onPermissionGranted(PermissionGrantedResponse
permissionGrantedResponse) {
                        getCurrentLocation();

                    }
```

```
                    @Override
                    public void onPermissionDenied(PermissionDeniedResponse
permissionDeniedResponse) {

                    }

                    @Override
                    public void
onPermissionRationaleShouldBeShown(PermissionRequest permissionRequest,
PermissionToken permissionToken) {

                        permissionToken.continuePermissionRequest();
                    }
                }).check();

    }

    private void getCurrentLocation() {
        if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED
&& ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
            // TODO: Consider calling
            //    ActivityCompat#requestPermissions
            // here to request the missing permissions, and then overriding
            //   public void onRequestPermissionsResult(int requestCode,
String[] permissions,
            //                                          int[] grantResults)
            // to handle the case where the user grants the permission. See the
documentation
            // for ActivityCompat#requestPermissions for more details.
            return;
        }
        Task<Location> task = fusedLocationProviderClient.getLastLocation();
        task.addOnSuccessListener(new OnSuccessListener<Location>() {
            @Override
            public void onSuccess(Location location) {
                supportMapFragment.getMapAsync(new OnMapReadyCallback() {
                    @Override
                    public void onMapReady(@NonNull GoogleMap googleMap) {
                        if (location!=null){
                            LatLng latLng=new
LatLng(location.getLatitude(),location.getLongitude());
                            MarkerOptions markerOptions=new
MarkerOptions().position(latLng).title("Current Location !");
                            googleMap.addMarker(markerOptions);

googleMap.animateCamera(CameraUpdateFactory.newLatLngZoom(latLng,15));

                        }else{
                            Toast.makeText(MainActivity.this,"Pleae on your
location app permission",Toast.LENGTH_LONG).show();
                        }
                    }
                });
            }
        });
    }
}
```

Output:


Current Location !
VENKATESHWARA
CEMENT BRICKS
Temporarily close
1:06 PM