# A3 Algorithm

import random k=random.getrandbits(128) m=random.getrandbits(128) kb=bin(k)[6:]

mb=bin(m)[4:] kbl=kb[0:64] kbr=kb[64:] mbl=mb[0:64] mbr=mb[64:] a1=int(kbI,2) int(mbr,2) a2=int(kbr,2) int(mbI,2) a3=a1^a2

a4=bin(a3)[2:].zfiII(64) a5=a4[0:32] a6=a4[32:]

a7=int(a5,2) int(a6,2) print("128 Bit Key = ",kb)

print("128 Random Bits Generated = ",mb) print("RES/SRES =",bin(a7)[2:].zfiII(Ien(a5)))

output

128 Bit Key = 1000010101110101100010111110101110111110010011100110101000101111011100100111011

100010101111001011100001001110110011010000000

128 Random Bits Generated = 11011110110100011001100111111000100001011001000111101100110111111100110100011010

011011010011101011011101100011101011001101000

RES/SRES = 10000010010100101111101010110101

# Application using database

AndroidManifest.xml

<uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE" />

activity\_main.xml

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

<LinearLayout

[xmlns:android="http://schemas.android.com/apk/res/android"](http://schemas.android.com/apk/res/android) [xmlns:tooIs="http://schemas.android.com/tools"](http://schemas.android.com/tools) android:layout width="match\_parent"

android:layout height="match parent" android:orientation="verticaI" tools:context=".MainActivity">

<!--Edit text to enter course name-->

<EditText

android:id="@+id/idEdtCourseName" android:Iayout\_width="match\_parent" android:layout height="wrap content" android:layout margin="10dp" android:hint="Enter course Name" />

<!--edit text to enter course duration-->

<EditText

android:id="@+id/idEdtCourseDuration" android:layout width="match parent" android:Iayout\_height="wrap\_content" android:layout margin="10dp" android:hint="Enter Course Duration" />

<!--edit text to display course tracks-->

<EditText

android:id="@+id/idEdtCourseTracks" android:Iayout\_width="match\_parent" android:layout height="wrap content" android:Iayout\_margin="10dp" android:hint="Enter Course Tracks" />

<!--edit text for course description-->

<EditText

android:id="@+id/idEdtCourseDescription” android:layout width="match parent"

android:Iayout\_height="wrap\_content"

android:Iayout\_margin="10dp" android:hint="Enter Course Description" />

<!--button for adding new course-->

<Button

android:id="@+id/idBtnAddCourse" android:Iayout\_width="match\_parent" android:layout height="wrap content" android:layout margin="10dp" android:text="Add Course" android:textAIICaps="faIse" />

</LinearLayout>

DBHandler

import android.content.ContentValues; import android.content.Context;

import android.database.sqIite.SQLiteDatabase; import android.database.sqIite.SQLiteOpenHelper; public class DBHandler extends SQLiteOpenHelper (

*//* creatingaconstant variables forourdatabase. *II* below variable is for our database name. privatestaticfinalString DB\_NAME="coursedb ,

*I/* below int is our database version privatestaticfinalintDB\_VERSION =1;

*II* below variable is for our table name.

private static final String TABLE\_NAME = "mycourses";

*II* below variable is for our id column. private staticfinal StringID\_COL ="id";

*I/* belowvariable is forourcoursename column privatestaticfinalStringNAME\_COL="name",

*II* below variable id for our course duration column. private static final String DURATION\_COL = "duration";

*II* below variable for our course description column. private static final String DESCRIPTION COL = "description";

*//* below variable is for our course tracks column. private static final String TRACKS\_COL = "tracks";

*II* creating a constructor for our database handler. public DBHandIer(Context context) { super(context, DB\_NAME, null, DB\_VERSION);

*I/* below method is for creating a database by running a sqlite query @Override public void onCreate(SQLiteDatabase db) {

*II* on below line we are creating

*II* an sqlite query and we are *II* setting our column names *II* along with their data types. String query = "CREATE TABLE " + TABLE\_NAME + " ("

+ ID\_COL + " INTEGER PRIMARY KEY AUTOINCREMENT, "

+ NAME\_COL + "TEXT,"

+ DURATION\_COL + " TEXT,"

+ DESCRIPTION\_COL + "TEXT,"

+ TRACKS\_COL + " TEXT)";

*II* at last we are calling a exec sql

*II* method to execute above sql query db.execSQL(query);

*/l* this method is use to add new course to our sqlite database.

public void addNewCourse(String courseName, String courseDuration, String courseDescription, String courseTracks) (

*II* on below line we are creating a variable for

*II* our sqlite database and calling writable method

*II* as we are writing data in our database. SQLiteDatabase db = this.getWritabIeDatabase();

*II* on below line we are creating a

*II* variable for content values.

ContentValues values = new ContentValues();

*II* on below line we are passing all values

// along with its key and value pair. values.put(NAME COL, courseName); values.put(DURATION\_COL, courseDuration);

values.put(DESCRIPTION\_COL, courseDescription); values.put(TRACKS\_COL, courseTracks);

*II* after adding allvalues wearepassing *II* content values to our table. db.insert(TABLE\_NAME, null, values);

*II* at last we are closing our

*II* database after adding database. db.close();

@Override

public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) { *II* this method is called to check if the table exists already. db.execSQL("DROP TABLE IF EXISTS " + TABLE\_NAME);

onCreate(db);

MainActivity.java file import android.os.Bundle; import android.view.View;

import android.widget.Button; import android.widget.EditText; import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity; public class MainActivity extends AppCompatActivity {

*//* creating variables for our edittext, button and dbhandler

private EditText courseNameEdt, courseTracksEdt, courseDurationEdt, courseDescriptionEdt; private Button addCourseBtn; private DBHandler dbHandler;

@Override

protected void onCreate(Bundle savedlnstancestate) ( super.onCreate(savedlnstancestate); setContentView(R.layout.activity main);

*II* initializing all our variables.

courseNameEdt = findViewByld(R.id.idEdtCourseName);

courseTracksEdt = findViewByld(R.id.idEdtCourseTracks); courseDurationEdt = findViewByld(R.id.idEdtCourseDuration); courseDescriptionEdt = findViewByld(R.id.idEdtCourseDescription); addCourseBtn = findViewByld(R.id.idBtnAddCourse);

*II* creating a new dbhandler class

*II* and passing our context to it.

dbHandler = new DBHandler(MainActivity.this);

*II* below line is to add on click listener for our add course button. addCourseBtn.setOnCIickListener(new View.OnCIickListener() (

@Override

public void onCIick(View v) {

*/l* below line is to get data from all edit text fields.

String courseName = courseNameEdt.getText().tostring(); String courseTracks = courseTracksEdt.getText().tostring(); String courseDuration = courseDurationEdt.getText().tostring(); String courseDescription = courseDescriptionEdt.getText().tostring();

*/l* validating if the text fields are empty or not.

if (courseName.isEmpty() && courseTracks.isEmpty() && courseDuration.isEmpty() && courseDescription.isEmpty()) (

Toast.makeText(MainActivity.this, "Please enter all the data..", Toast.LENGTH\_SHORT).show(); return;

*/l* on below line we are calling a method to add new

*/l* course to sqlite data and pass all our values to it. dbHandler.addNewCourse(courseName, courseDuration, courseDescription, courseTracks);

*/l* after adding the data we are displaying a toast message. Toast.makeText(MainActivity.this, "Course has been added.",

Toast.LENGTH\_SHORT).show();

courseNameEdt.setText( ); courseDurationEdt.setText( ); courseTracksEdt.setText( ); courseDescriptionEdt.setText( );





database app

john marks : 90

sub : porn

std 12

Course has been added.



john 90

pcm 12

sam

pcm 13

Course has been added.

**Bluetooth**

MainActivity.java

package com.example.sairamkrishna.myapplication;

import android.app.Activity;

import android.bluetooth.BluetoothAdapter;

import android.bluetooth.BluetoothDevice;

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import android.widget.ArrayAdapter;

import android.widget.Button;

import android.widget.ListView;

import android.widget.Toast;

import java.util.ArrayList;

import java.util.Set;

public class MainActivity extends Activity {

Button b1,b2,b3,b4;

private BluetoothAdapter BA;

private Set<BluetoothDevice>pairedDevices;

ListView lv;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

b1 = (Button) findViewById(R.id.button);

b2=(Button)findViewById(R.id.button2);

b3=(Button)findViewById(R.id.button3);

b4=(Button)findViewById(R.id.button4);

BA = BluetoothAdapter.getDefaultAdapter();

lv = (ListView)findViewById(R.id.listView);

}

public void on(View v){

if (!BA.isEnabled()) {

Intent turnOn = new Intent(BluetoothAdapter.ACTION\_REQUEST\_ENABLE);

startActivityForResult(turnOn, 0);

Toast.makeText(getApplicationContext(), "Turned on",Toast.LENGTH\_LONG).show();

} else {

Toast.makeText(getApplicationContext(), "Already on", Toast.LENGTH\_LONG).show();

}

}

public void off(View v){

BA.disable();

Toast.makeText(getApplicationContext(), "Turned off" ,Toast.LENGTH\_LONG).show();

}

public void visible(View v){

Intent getVisible = new Intent(BluetoothAdapter.ACTION\_REQUEST\_DISCOVERABLE);

startActivityForResult(getVisible, 0);

}

public void list(View v){

pairedDevices = BA.getBondedDevices();

ArrayList list = new ArrayList();

for(BluetoothDevice bt : pairedDevices) list.add(bt.getName());

Toast.makeText(getApplicationContext(), "Showing Paired Devices",Toast.LENGTH\_SHORT).show();

final ArrayAdapter adapter = new ArrayAdapter(this,android.R.layout.simple\_list\_item\_1, list);

lv.setAdapter(adapter);

}

}

activity\_main.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:paddingLeft="@dimen/activity\_horizontal\_margin"

android:paddingRight="@dimen/activity\_horizontal\_margin"

android:paddingTop="@dimen/activity\_vertical\_margin"

android:paddingBottom="@dimen/activity\_vertical\_margin"

tools:context=".MainActivity"

android:transitionGroup="true">

<TextView android:text="Bluetooth Example"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:id="@+id/textview"

android:textSize="35dp"

android:layout\_alignParentTop="true"

android:layout\_centerHorizontal="true" />

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Tutorials point"

android:id="@+id/textView"

android:layout\_below="@+id/textview"

android:layout\_centerHorizontal="true"

android:textColor="#ff7aff24"

android:textSize="35dp" />

<ImageView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:id="@+id/imageView"

android:src="@drawable/abc"

android:layout\_below="@+id/textView"

android:layout\_centerHorizontal="true"

android:theme="@style/Base.TextAppearance.AppCompat" />

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Turn On"

android:id="@+id/button"

android:layout\_below="@+id/imageView"

android:layout\_toStartOf="@+id/imageView"

android:layout\_toLeftOf="@+id/imageView"

android:clickable="true"

android:onClick="on" />

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Get visible"

android:onClick="visible"

android:id="@+id/button2"

android:layout\_alignBottom="@+id/button"

android:layout\_centerHorizontal="true" />

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="List devices"

android:onClick="list"

android:id="@+id/button3"

android:layout\_below="@+id/imageView"

android:layout\_toRightOf="@+id/imageView"

android:layout\_toEndOf="@+id/imageView" />

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="turn off"

android:onClick="off"

android:id="@+id/button4"

android:layout\_below="@+id/button"

android:layout\_alignParentLeft="true"

android:layout\_alignParentStart="true" />

<ListView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:id="@+id/listView"

android:layout\_alignParentBottom="true"

android:layout\_alignLeft="@+id/button"

android:layout\_alignStart="@+id/button"

android:layout\_below="@+id/textView2" />

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Paired devices:"

android:id="@+id/textView2"

android:textColor="#ff34ff06"

android:textSize="25dp"

android:layout\_below="@+id/button4"

android:layout\_alignLeft="@+id/listView"

android:layout\_alignStart="@+id/listView" />

</RelativeLayout>

Output :

MainActivity.java

package com.example.sairamkrishna.myapplication;

import android.app.Activity;

import android.bluetooth.BluetoothAdapter;

import android.bluetooth.BluetoothDevice;

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import android.widget.ArrayAdapter;

import android.widget.Button;

import android.widget.ListView;

import android.widget.Toast;

import java.util.ArrayList;

import java.util.Set;

public class MainActivity extends Activity {

Button b1,b2,b3,b4;

private BluetoothAdapter BA;

private Set<BluetoothDevice>pairedDevices;

ListView lv;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

b1 = (Button) findViewById(R.id.button);

b2=(Button)findViewById(R.id.button2);

b3=(Button)findViewById(R.id.button3);

b4=(Button)findViewById(R.id.button4);

BA = BluetoothAdapter.getDefaultAdapter();

lv = (ListView)findViewById(R.id.listView);

}

public void on(View v){

if (!BA.isEnabled()) {

Intent turnOn = new Intent(BluetoothAdapter.ACTION\_REQUEST\_ENABLE);

startActivityForResult(turnOn, 0);

Toast.makeText(getApplicationContext(), "Turned on",Toast.LENGTH\_LONG).show();

} else {

Toast.makeText(getApplicationContext(), "Already on", Toast.LENGTH\_LONG).show();

}

}

public void off(View v){

BA.disable();

Toast.makeText(getApplicationContext(), "Turned off" ,Toast.LENGTH\_LONG).show();

}

public void visible(View v){

Intent getVisible = new Intent(BluetoothAdapter.ACTION\_REQUEST\_DISCOVERABLE);

startActivityForResult(getVisible, 0);

}

public void list(View v){

pairedDevices = BA.getBondedDevices();

ArrayList list = new ArrayList();

for(BluetoothDevice bt : pairedDevices) list.add(bt.getName());

Toast.makeText(getApplicationContext(), "Showing Paired Devices",Toast.LENGTH\_SHORT).show();

final ArrayAdapter adapter = new ArrayAdapter(this,android.R.layout.simple\_list\_item\_1, list);

lv.setAdapter(adapter);

}

}

**activity\_main.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:paddingLeft="@dimen/activity\_horizontal\_margin"

android:paddingRight="@dimen/activity\_horizontal\_margin"

android:paddingTop="@dimen/activity\_vertical\_margin"

android:paddingBottom="@dimen/activity\_vertical\_margin"

tools:context=".MainActivity"

android:transitionGroup="true">

<TextView android:text="Bluetooth Example"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:id="@+id/textview"

android:textSize="35dp"

android:layout\_alignParentTop="true"

android:layout\_centerHorizontal="true" />

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Tutorials point"

android:id="@+id/textView"

android:layout\_below="@+id/textview"

android:layout\_centerHorizontal="true"

android:textColor="#ff7aff24"

android:textSize="35dp" />

<ImageView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:id="@+id/imageView"

android:src="@drawable/abc"

android:layout\_below="@+id/textView"

android:layout\_centerHorizontal="true"

android:theme="@style/Base.TextAppearance.AppCompat" />

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Turn On"

android:id="@+id/button"

android:layout\_below="@+id/imageView"

android:layout\_toStartOf="@+id/imageView"

android:layout\_toLeftOf="@+id/imageView"

android:clickable="true"

android:onClick="on" />

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Get visible"

android:onClick="visible"

android:id="@+id/button2"

android:layout\_alignBottom="@+id/button"

android:layout\_centerHorizontal="true" />

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="List devices"

android:onClick="list"

android:id="@+id/button3"

android:layout\_below="@+id/imageView"

android:layout\_toRightOf="@+id/imageView"

android:layout\_toEndOf="@+id/imageView" />

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="turn off"

android:onClick="off"

android:id="@+id/button4"

android:layout\_below="@+id/button"

android:layout\_alignParentLeft="true"

android:layout\_alignParentStart="true" />

<ListView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:id="@+id/listView"

android:layout\_alignParentBottom="true"

android:layout\_alignLeft="@+id/button"

android:layout\_alignStart="@+id/button"

android:layout\_below="@+id/textView2" />

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Paired devices:"

android:id="@+id/textView2"

android:textColor="#ff34ff06"

android:textSize="25dp"

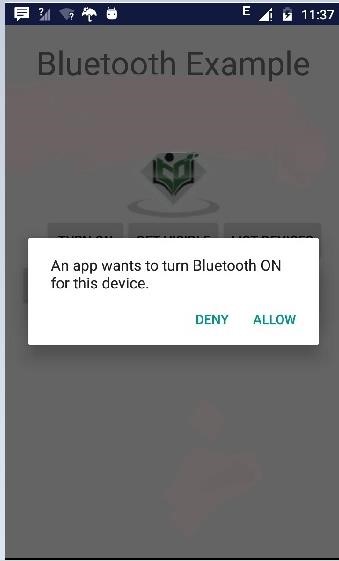
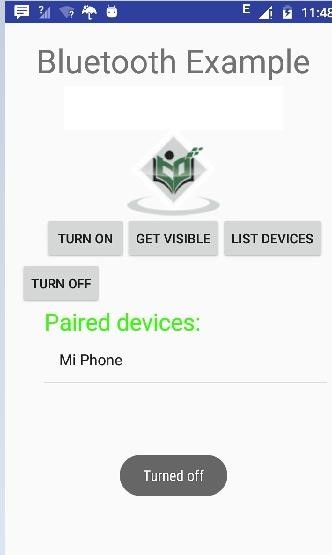
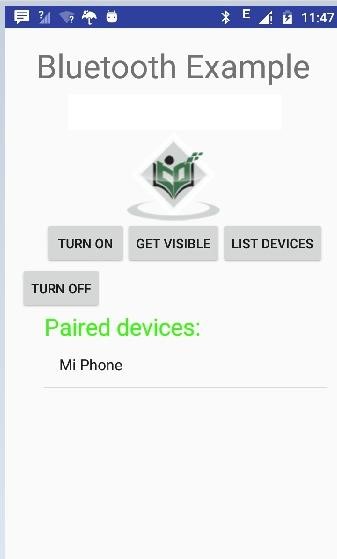
android:layout\_below="@+id/button4"

android:layout\_alignLeft="@+id/listView"

android:layout\_alignStart="@+id/listView" />

</RelativeLayout>

**Output :**



# Alert Message

package com.example.notificationdemo; import android.app.Activity; import android.app.NotificationManager;

import android.app.Pendinglntent; import android.content.Context; import android.content.Intent; import android.support.v4.app.NotificationCompat; import android.os.Bundle;

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

publicclassMainActivityextendsActivity( Button b1; @Override

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

b1 = (Button)findViewById(R.id.button); b1.setOnCIickListener(new View.OnCIickListener() { @Override

publicvoidonCIick(Viewv){ addNotification();



private void addNotification() { NotificationCompat.Builder builder = new NotificationCompat.Builder(this)

.setSmaIIIcon(R.drawabIe.abc)

.setContentTitIe("Notifications Example")

.setContentText("This is a test notification");

Intent notificationlntent = new Intent(this, MainActivity.class);

Pendinglntent contentlntent = Pendinglntent.getActivity(this, 0, notificationlntent, PendingIntent.FLAG\_UPDATE\_CURRENT); builder.setContentIntent(contentlntent);

*/l* Add as notification

NotificationManager manager = (NotificationManager) getsystemservice(Context.NOTIFICATION\_SERVICE); manager.notify(0, builder.build());

notification.xml

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

<LinearLayout [xmlns:android="http://schemas.android.com/apk/res/android"](http://schemas.android.com/apk/res/android) android:orientation="verticaI"

android:layout width="fiII parent" android:layout height="fiII parent" >

<TextView android:Iayout\_width="fiII\_parent" android:Iayout\_height="400dp" android:text="Hi, Your Detailed notification viewgoes here. " />

</LinearLayout>

package com.example.notificationdemo; import android.os.Bundle; import android.app.Activity;

public class NotificationView extends Activity{ @Override

public void onCreate(Bundle savedlnstancestate)( super.onCreate(savedlnstancestate);

setContentView(R.layout.notification);

activity\_main.xml

<ReIativeLayout [xmlns:android="http://schemas.android.com/apk/res/android"](http://schemas.android.com/apk/res/android) [xmlns:tooIs="http://schemas.android.com/tools"](http://schemas.android.com/tools)

android:layout width="match parent" android:layout height="match parent" android:paddingBottom="@dimen/activity\_verticaI\_margin" android:paddingLeft="@dimen/activity horizontal margin" android:paddingRight="@dimen/activity\_horizontaI\_margin" android:paddingTop="@dimen/activity vertical margin" tools:context="MainActivity">

<TextView android:id="@+id/textView1" android:Iayout\_width="wrap\_content" android:layout height="wrap content" android:text="Notification Example" android:Iayout\_aIignParentTop="true" android:layout centerHorizontaI="true" android:textSize="30dp" />

<TextView android:id="@+id/textView2" android:Iayout\_width="wrap\_content"

android:layout height="wrap content" android:text="TutoriaIs point " android:textCoIor=”#ff87ff09"

android:textSize="30dp"

android:layout below="@+id/textView1" android:Iayout\_centerHorizontaI="true" android:Iayout\_marginTop="48dp" />

<ImageButton

android:layout width="wrap content" android:Iayout\_height="wrap\_content" android:id="@+id/imageButton" android:src="@drawabIe/abc" android:layout below="@+id/textView2" android:Iayout\_centerHorizontaI="true" android:Iayout\_marginTop="42dp"

/>

<Button

android:layout width="wrap content" android:Iayout\_height="wrap\_content" android:text="Notification" android:id="@+id/button" android:layout marginTop="62dp" android:Iayout\_beIow="@+id/imageButton" android:layout centerHorizontaI="true" />

</ReIativeLayout>

strings.xml

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

<resources>

<string name="action\_settings">Settings</string>

<string name="app name">tutoriaIspoint </strings

</resources>

AndroidManifest.xml

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

<manifest [xmlns:android="http://schemas.android.com/apk/res/android"](http://schemas.android.com/apk/res/android)

package=”com.example.notificationdemo” >

<appIication android:aIIowBackup="true" android:icon="@drawabIe/ic\_Iauncher" android:label="@string/app name" android:theme="@style/AppTheme" >

<activity android:name="com.example.notificationdemo.MainActivity" android:label="@string/app\_name" >

<intent-filters

<action android:name="android.intent.action.MAIN" />

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

</intent-filters

</activity>

<activity android:name=".NotificationView" android:IabeI="DetaiIs of notification" android:parentActivityName=".MainActivity">

<meta-data android:name="android.support.PARENT ACTIVITY" android:value=".MainActivity"/>

</activity>

</appIication>

</manifest>

protected void displayNotification() { Log.i("Start", "notification");

/\* Invoking the default notification service \*/

NotificationCompat.Builder mBuilder = new NotificationCompat.Builder(this);

mBuiIder.setContentTitIe("New Message"); mBuiIder.setContentText("You’ve received new message."); mBuiIder.setTicker("New Message Alert!"); mBuilder.setSmaIIIcon(R.drawabIe.woman);

/\*Increasenotification numbereverytimeanewnotificationarrives \*/ mBuilder.setNumber(++numMessages);

/\*AddBigViewSpecific Configuration \*/

NotificationCompat.InboxStyIe inboxstyle = new NotificationCompat.InboxStyIe();

String[] events = new String[6];

events[0] = new String("This is first line. "); events[1] = new String("This is second line. "); events[2] = new String("This is third line. "); events[3] = new String("This is 4thline. "); events[4] =newString("This isSthline. ");

events[5] = new String("This is 6thline. ");

*f/* Sets a title for the Inbox style big view inboxStyIe.setBigContentTitIe("Big Title Details:");

*//* Moves events into the big view for (int i=0; i< events.length; i++)( inboxstyle.addLine(events[i]);

mBuilder.setStyIe(inboxstyle);

/\* Creates an explicit intent for an Activity in your app \*/ Intent resultlntent = new Intent(this, NotificationView.class);

TaskStackBuilder stackBuilder = TaskStackBuilder.create(this); stackBuilder.addParentStack(NotificationView.class);

/\* Adds the Intent that starts the Activity tothetopofthe stack \*/ stackBuilder.addNextIntent(resultlntent);

Pendinglntent resultPendinglntent

=stackBuiIder.getPendingIntent(0,PendingIntent.FLAG UPDATE CURRENT);

mBuilder.setContentIntent(resultPendinglntent); mNotificationManager = (NotificationManager) getsystemservice(Context.NOTIFICATION SERVICE);

/\* notificationlD allows you to update the notification later on. \*/ mNotificationManager.notify(notificationlD, mBuiIder.build());



YOU HAVE NEW NOTIFICATION SMS 1

MAIL 2

# Simple Emi Calculator

Activity main.xml

<android.support.design.widget.CoordinatorLayout [xmlns:android="http://schemas.android.com/apk/res/android"](http://schemas.android.com/apk/res/android) android:Iayout\_width="match\_parent" [xmlns:tooIs="http://schemas.android.com/tools"](http://schemas.android.com/tools) [xmlns:app="http://schemas.android.com/apk/res-auto"](http://schemas.android.com/apk/res-auto) tools:context="abu.emicaIcuIator.MainActivity"

android:layout height="match parent">

<android.support.v4.widget.NestedScrollView android:Iayout\_width="match\_parent" android:layout height="match parent"

app:Iayout\_behavior="@string/appbar\_scroIIing\_view\_behavior">

<LinearLayout

android:layout width="fiII parent" android:layout height="match parent" android:Iayout\_marginTop="?attr/actionBarSize" android:orientation="verticaI" android:paddingLeft="20dp" android:paddingRight="20dp" android:paddingTop="10dp">

<android.support.design.widget.TextlnputLayout android:id="@+id/input\_Iayout\_principaI" android:layout width="match parent" android:Iayout\_height="wrap\_content">

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

android:layout width="match parent" android:layout height="wrap content" android:singIeLine="true" " android:inputType="number" android:digits="0123456789.” android:hint="@string/hint\_principaI" />

</android.support.design.widget.TextInputLayout>

<android.support.design.widget.TextlnputLayout android:id="@+id/input layout interest" android:Iayout\_width="match\_parent"

android:Iayout\_height="wrap\_content">

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

android:layout width="match parent" android:Iayout\_height="wrap\_content" android:singIeLine="true" android:inputType="number" android:digits="0123456789.” android:hint="@string/hint\_interest" />

</android.support.design.widget.TextInputLayout>

<android.support.design.widget.TextlnputLayout android:id="@+id/input\_Iayout\_tenure" android:Iayout\_width="match\_parent" android:layout height="wrap content">

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

android:Iayout\_width="match parent" android:Iayout\_height="wrap\_content" android:inputType="number" android:digits="0123456789." android:hint="@string/hint\_years" />

</android.support.design.widget.TextInputLayout>

<Button android:id="@+id/btn\_caIcuIate2" android:layout width="fiII parent" android:Iayout\_height="wrap\_content" android:text="CaIcuIate" android:background="@color/colorPrimary" android:Iayout\_marginTop="40dp"

android:textCoIor="@android:color/white"/>

<android.support.design.widget.TextlnputLayout android:id="@+id/input layout emi" android:Iayout\_width="match\_parent" android:layout height="wrap content" android:Iayout\_marginTop="40dp">

<EditText

android:id="@+id/emi" android:Iayout\_width="match\_parent" android:layout height="wrap content" android:maxEms="0" android:inputType="number" android:hint="@string/hint\_emi" />

</android.support.design.widget.TextInputLayout>

<android.support.design.widget.TextlnputLayout android:id="@+id/input\_Iayout\_totaI\_Interest" android:Iayout\_width="match\_parent" android:layout height="wrap content" android:Iayout\_marginTop="10dp">

<EditText android:id="@+id/interest\_totaI" android:layout width="match parent" android:layout height="wrap content"

android:inputType="number" android:hint="@string/hint interest total" />

</android.support.design.widget.TextInputLayout>

</LinearLayout>

</android.support.v4.widget.NestedScroIIView>

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

compile ’com.android.support:appcompat-v7:23.4.0’ compile ’com.android.support:design:23.1.0’

string XML Code

<resources>

<string name="app\_name">EMI Calculators/strings

<string name="hint\_principaI">PrincipaIAmount T</string>

<string name="hint interest">Interest rate per Year %</strings

<string name="hint\_years">How Many Years</string>

<string name="hint emi">EMI Z</string>

<string name="hint interest totaI">TotaI Interest for Loan Z</strings

</resources>

Main Activity.java.

package abu.emicaIcuIator; import android.os.Bundle;

import android.support.v7.app.AppCompatActivity; import android.text.TextUtils;

import android.view.View; import android.widget.Button; import android.widget.EditText; public class MainActivity extends AppCompatActivity { Button emiCalcBtn;

@Override

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

final EditText P = (EditText)findViewByld(R.id.principal); finalEditTextI=(EditText) findViewByld(R.id.interest); final EditText Y=(EditText) findViewByld(R.id.years);

final EditText TI = (EditText) findViewByld(R.id.interest\_totaI); final EditText result = (EditText) findViewByld(R.id.emi); emiCalcBtn = (Button) findViewByld(R.id.btn calculate2); emiCalcBtn.setOnCIickListener(new View.OnCIickListener() ( @Override

public void onCIick(View v) ( String st1 = P.getText().tostring(); String st2 = I.getText().tostring(); String st3

= Y.getText().tostring(); if (TextUtils.isEmpty(st1)) { P.setError("Enter Prncipal Amount"); P.requestFocus(); return;

if (TextUtils.isEmpty(st2)) { I.setError("Enter Interest Rate"); I.requestFocus(); return;

if (TextUtils.isEmpty(st3)) { Y.setError("Enter Years"); Y.requestFocus(); return;

float p = Float.parseFIoat(st1); float i = Float.parseFIoat(st2); float y = Float.parseFIoat(st3); float Principal

= caIPric(p); float Rate = caIInt(i); float Months = caIMonth(y);

float Dvdnt = caIDvdnt(Rate, Months);

float FD = caIFinaIDvdnt(Principal, Rate, Dvdnt); float D = caIDivider(Dvdnt); float emi = caIEmi(FD, D); float TA = caITa(emi, Months);

float ti = caITotaIInt(TA, Principal); result.setText(String.vaIueOf(emi)); TI.setText(String.vaIueOf(ti));

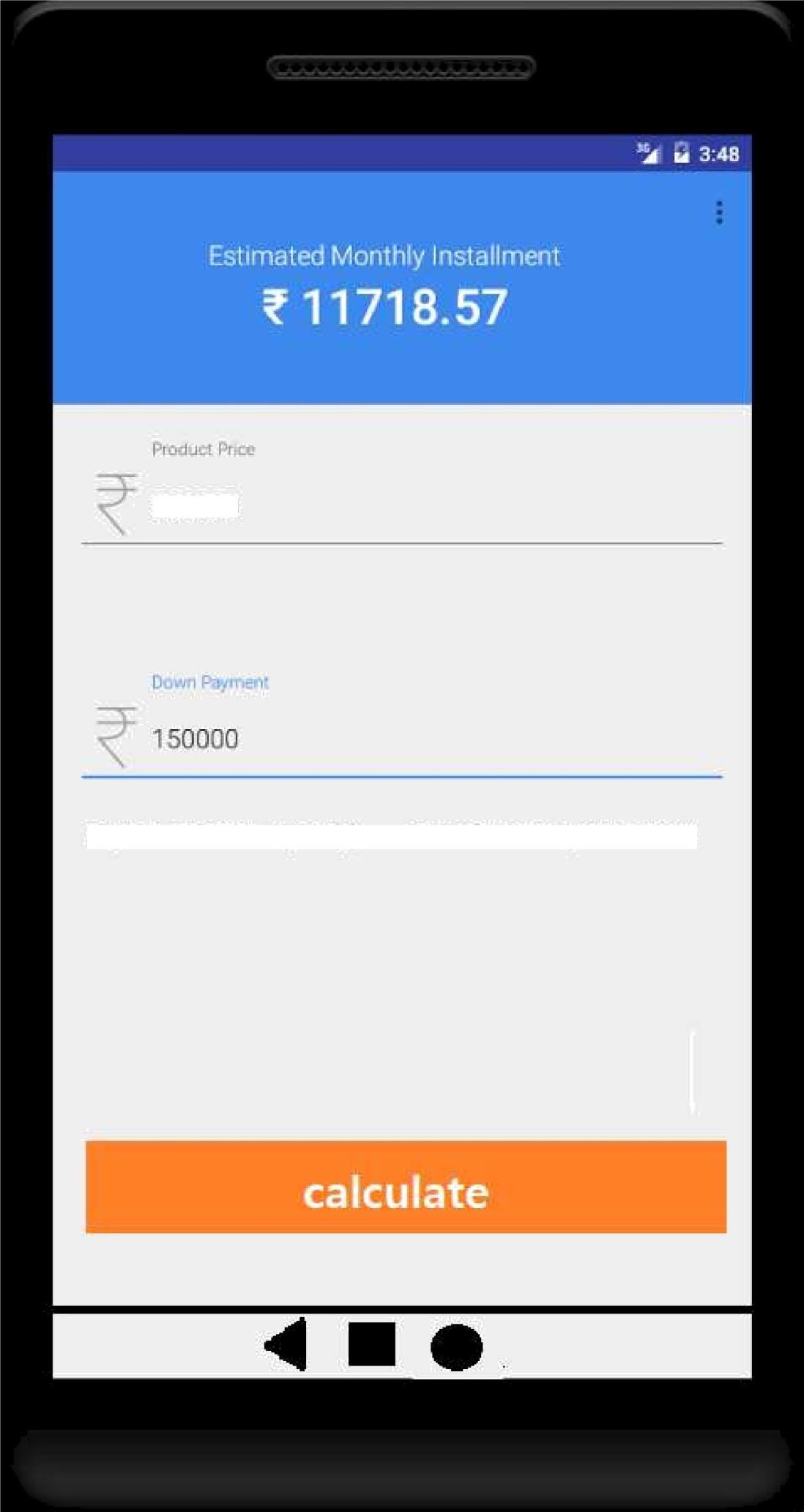


publicfloatcaIPric(floatp)( return (float)(p);

public float caIInt(float i)( return (float)(i / 12 / 100); publicfloatcaIMonth(floaty)( return (float)(y \* 12);

publicfloatcaIDvdnt(floatRate,floatMonths){ return (float)(Math.pow(1 + Rate, Months)); publicfloatcaIFinaIDvdnt(floatPrincipal,floatRate,floatDvdnt){ return (float)(Principal \* Rate \* Dvdnt); public float caIDivider(float Dvdnt) ( return (float)(Dvdnt - 1);

publicfloatcaIEmi(float FD,FloatD)( return (float)(FD / D); publicfloatcaITa(floatemi,FloatMonths){ return (float)(emi \* Months); public float caITotaIInt(float TA, float Principal) { return (float)(TA - Principal);



# Photo gallery

AndroidManifest.xml

<!-- permissions for reading external storage -->

<uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE" /> android:hardwareAcceIerated="faIse"

android:IargeHeap="true"

activity main.xml

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

<ReIativeLayout [xmlns:android="http://schemas.android.com/apk/res/android"](http://schemas.android.com/apk/res/android)

[xmlns:tooIs="http://schemas.android.com/tools"](http://schemas.android.com/tools) android:layout width="match parent" android:Iayout\_height="match\_parent" android:Iayout\_gravity="center” android:gravity="center" android:orientation="verticaI" tools:context=".MainActivity">

<!--recycIer view for displaying the list of images-->

<androidx.recycIerview.widget.RecyclerView android:id="@+id/idRVlmages" android:Iayout\_width="match\_parent" android:Iayout\_height="match\_parent" />

</ReIativeLayout> layout Resource file

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

<androidx.cardview.widget.CardView [xmlns:android="http://schemas.android.com/apk/res/android"](http://schemas.android.com/apk/res/android) [xmlns:app="http://schemas.android.com/apk/res-auto"](http://schemas.android.com/apk/res-auto) android:Iayout\_width="wrap\_content” android:Iayout\_height="wrap\_content"

android:layout gravity="center" android:Iayout\_margin="3dp" android:eIevation="8dp" app:cardCornerRadius="8dp">

<!--Image view for displaying the image in our card layout in recycler view-->

<ImageView android:id="@+id/idlVlmage"

android:Iayout\_width="100dp" android:Iayout\_height="100dp" android:layout gravity="center" android:scaIeType="centerCrop" />

</androidx.cardview.widget.CardView>

Empty Activity > New Java class import android.content.Context; import android.content.Intent; import android.view.Layoutlnflater; import android.view.View;

import android.view.ViewGroup; import android.widget.ImageView;

import androidx.annotation.NonNull;

import androidx.recycIerview.widget.RecyclerView; import com.squareup.picasso.Picasso; import java.io.File; import java.utiI.ArrayList;

public class RecyclerViewAdapter extends RecyclerView.Adapter<RecycIerViewAdapter.RecycIerViewHoIder> {

*I/* creating avariable forourcontext and array list. private final Context context; private final ArrayList<String> imagePathArrayList;

*II* on below line we have created a constructor.

public RecycIerViewAdapter(Context context, ArrayList<String> imagePathArrayList) { this.context = context;

this.imagePathArrayList = imagePathArrayList;

@NonNuII @Override

public RecyclerViewHolder onCreateViewHoIder(@NonNuII ViewGroup parent, int viewType) {

*II* Inflate Layout in this method which we have created.

View view = Layoutlnflater.from(parent.getContext()).inflate(R.layout.card\_Iayout, return new

parent, false); RecycIerViewHoIder(view);

@Override

public void onBindViewHoIder(@NonNuII RecyclerViewHolder holder, int position) (

*II* on below line we are getting the file from the

*II* path which we have stored in our list.

File imgFile = new File(imagePathArrayList.get(position));

*II* on below line we are checking if the file exists or not. if (imgFiIe.exists()) (

*//* if the file exists then we are displaying that file in our image view using picasso library.

Picasso.get().load(imgFiIe).placeholder(R.drawabIe.ic\_Iauncher\_background).into(holder.imagel V);

// on below line we are adding click listener to our item of recycler view. holder.itemView.setOnCIickListener(new View.OnCIickListener() ( @Override

public void onCIick(View v) (

*//* inside on click listener we are creating a new intent Intent i = new Intent(context, ImageDetaiIActivity.class);

activity.

*/f* on below line we are passing the image path to our new i.putExtra("imgPath", imagePathArrayList.get(position));

*//* at last we are starting our activity. context.startActivity(i);



@Override

public int getltemCount() (

*II* this method returns

*II* the size of recyclerview

return imagePathArrayList.size();

*//* View Holder Class to handle Recycler View.

public static class RecyclerViewHolder extends RecycIerView.ViewHoIder (

*II*creating variables forourviews. privatefinalImageViewimagelV;

public RecycIerViewHoIder(@NonNuII View itemView) ( super(itemView);

*/f* initializing our views with their ids.

imagelV = itemView.findViewById(R.id.idlVlmage);

MainActivity.java

import android.content.pm.PackageManager; import android.database.Cursor; import android.os.Bundle;

import android.provider.MediaStore; import android.widget.Toast;

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

import androidx.recycIerview.widget.GridLayoutManager; import androidx.recycIerview.widget.RecyclerView;

import java.utiI.ArrayList;

import static android.Manifest.permission.READ EXTERNAL STORAGE; public class MainActivity extends AppCompatActivity {

*//* on below line we are creating variables for

*//* our array list, recycler view and adapter class.

private staticfinalintPERMISSION REQUEST CODE =200; private ArrayList<String> imagePaths; private RecyclerView imagesRV;

private RecyclerViewAdapter imageRVAdapter;

@Override

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

*II* creating a new array list and

*II* initializing our recycler view. imagePaths = new ArrayList<>(); imagesRV = findViewByld(R.id.idRVlmages);

*II* we are calling a method to request

*II*thepermissions toreadexternalstorage. requestPermissions();

*II* calling a method to

*II*prepareourrecyclerview. prepareRecyclerView();

private boolean checkPermission() (

*II* in thismethod we are checking if thepermissions are grantedornot and returning the result.

int result = ContextCompat.checkSeIfPermission(getApplicationContext(), READ\_EXTERNAL\_STORAGE); return result == PackageManager.PERMISSION GRANTED;

private void requestPermissions() ( if (checkPermission()) (

// if the permissions are already granted we are calling

*/f* amethod to get allimages from ourexternal storage. Toast.makeText(this, "Permissions granted..", Toast.LENGTH\_SHORT).show();

getlmagePath();

} else (

*l/* if the permissions are not granted we are

//callingamethodtorequestpermissions. requestPermission();

private void requestPermission() (

//onbelowlinewearerequesting thereadexternalstoragepermissions. ActivityCompat.requestPermissions(this, new

String[](READ EXTERNAL STORAGE), PERMISSION REQUEST CODE);

private void prepareRecyclerView() {

*II* in this method we are preparing our recycler view.

*II* on below line we are initializing our adapter class.

imageRVAdapter = new RecyclerViewAdapter(MainActivity.this, imagePaths);

*II* on below line we are creating a new grid layout manager. GridLayoutManager manager = new GridLayoutManager(MainActivity.this, 4);

*II* on below line we are setting layout

*II* manager and adapter to our recycler view. imagesRV.setLayoutManager(manager); imagesRV.setAdapter(imageRVAdapter);

private void getlmagePath() (

*II* in this method we are adding all our image paths

*II* in our arraylist which we have created.

// on below line we are checking if the device is having an sd card or not. boolean isSDPresent = android.os.Environment.getExternaIStorageState().equals(android.os.Environment.MEDIA\_MO UNTED);

if (isSDPresent) {

*/f* if the sd card is present we are creating a new list in

// which we are getting our images data with their ids. final String[] columns = (MediaStore.Images.Media.DATA,

MediaStore.Images.Media. ID};

// on below line we are creating anew

// string to order our images by string.

final String orderBy = MediaStore.Images.Media.\_ID;

*/f* this method will stores all the images

// from the gallery in Cursor Cursor cursor =

getContentResolver().query(MediaStore.Images.Media.EXTERNAL\_CONTENT\_URI, columns, null, null, orderBy);

// below line is to get total number of images int count = cursor.getCount();

// on below line we are running a loop to add

*f/* theimage file path in ourarray list. for (int i = 0; i < count; i++) (

//onbelowlinewe are movingourcursor position cursor.moveToPosition(i);

*/l* onbelowlinewe are getting image file path int dataColumnlndex = cursor.getCoIumnIndex(MediaStore.Images.Media.DATA);

*/l* after that we are getting the image file path

*/l* and adding that path in our array list. imagePaths.add(cursor.getString(dataColumnlndex));

imageRVAdapter.notifyDataSetChanged();

// after adding the data to our

*f/* array listwe are closing ourcursor. cursor.close();

@Override

public void onRequestPermissionsResuIt(int requestCode, String permissions[], int[] grantResults) (

*II*thismethod iscalledafterpermissions hasbeengranted. switch (requestCode) (

//we arechecking thepermission code. casePERMISSION REQUEST CODE:

*/l* in this case we are checking if the permissions are accepted or not.

if (grantResuIts.length > 0) (

boolean storageAccepted = grantResuIts[0] == PackageManager.PERMISSION\_GRANTED;

if (storageAccepted) (

*//* if the permissions are accepted we are displaying a toast message

Toast.LENGTH\_SHORT).show();

anddisplayingthetoastmessage.

} else{

*//* and calling a method to get image path. Toast.makeText(this,"Permissions Granted..",

getlmagePath();

*//* if permissions are denied we are closing the app

Toast.makeText(this, "Permissions denied, Permissions are required to use the app..", Toast.LENGTH\_SHORT).show();

break; activity\_image\_detaiI.xml

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

<ReIativeLayout [xmlns:android="http://schemas.android.com/apk/res/android"](http://schemas.android.com/apk/res/android)

[xmlns:tooIs="http://schemas.android.com/tools"](http://schemas.android.com/tools) android:Iayout\_width="match\_parent" android:Iayout\_height="match\_parent" tools:context=".ImageDetaiIActivity">

<!--image view to display our image-->

<ImageView

android:id="@+id/idlVlmage" android:Iayout\_width="match\_parent" android:layout height="300dp" android:Iayout\_centerInParent="true" />

</ReIativeLayout>

ImageDetaiIActivity.java import android.os.Bundle; import android.view.MotionEvent;

import android.view.ScaleGestureDetector; import android.widget.ImageView;

import androidx.appcompat.app.AppCompatActivity; import com.squareup.picasso.Picasso; import java.io.File;

public class ImageDetailActivity extends AppCompatActivity (

*II* creating a string variable, image view variable

*//* and a variable for our scale gesture detector class. String imgPath;

private ImageView imageView;

private ScaleGestureDetector scaleGestureDetector;

*II* on below line we are defining our scale factor. private float mScaleFactor = 1.0f; @Override

protected void onCreate(Bundle savedlnstancestate) ( super.onCreate(savedlnstancestate); setContentView(R.layout.activity\_image\_detaiI);

*II* on below line getting data which we have passed from our adapter class. imgPath = getIntent().getStringExtra("imgPath");

*II* initializing our image view.

imageView = findViewByld(R.id.idlVlmage);

for our image.

*II* on below line we are initializing our scale gesture detector for zoom in and out scaleGestureDetector = new ScaIeGestureDetector(this, new ScaleListener());

*II* on below line we are getting our image file from its path. File imgFile = new File(imgPath);

*II* if the file exists then we are loading that image in our image view. if (imgFiIe.exists()) {

Picasso.get().load(imgFiIe).placeholder(R.drawabIe.ic\_Iauncher\_background).into(imageView);

@Override

public boolean onTouchEvent(MotionEvent motionEvent) (

*II* inside on touch event method we are calling on

*II* touch event method and passing our motion event to it. scaleGestureDetector.onTouchEvent(motionEvent); return true;

private class ScaleListener extends ScaIeGestureDetector.SimpIeOnScaIeGestureListener (

*II* on below line we are creating a class for our scale *II* listener and extending it with gesture listener. @Override

public boolean onScaIe(ScaIeGestureDetector scaleGestureDetector) (

*//* inside on scale method we are setting scale

*//* for our image in our image view.

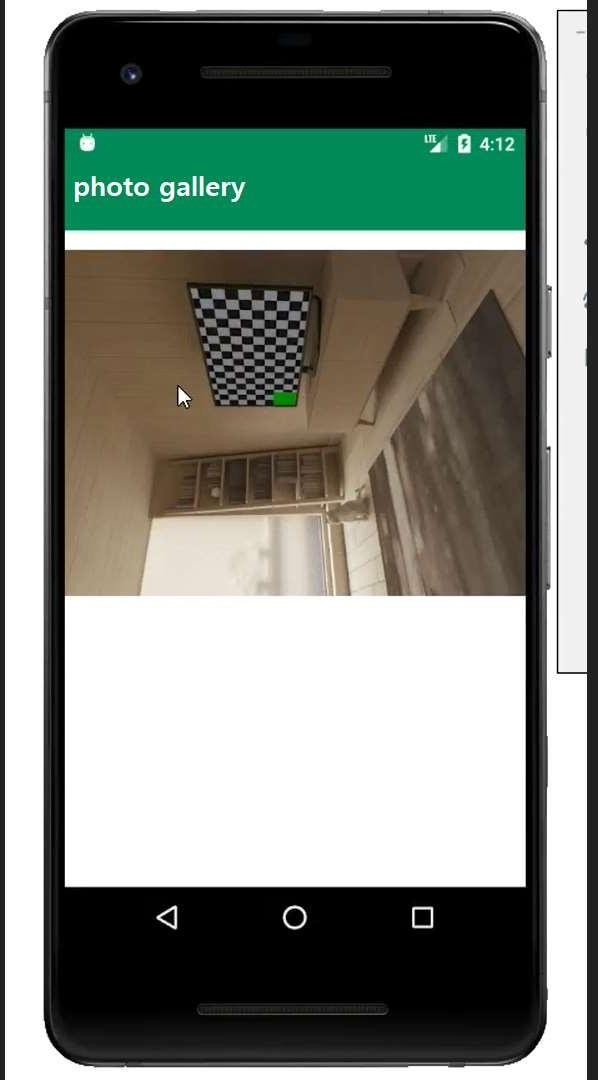
mScaleFactor \*= scaIeGestureDetector.getScaIeFactor(); mScaleFactor = Math.max(0.1f, Math.min(mScaleFactor, 10.0f));

// on below line we are setting

*f/* scale x and scale y to our image view. imageView.setScaIeX(mScaleFactor);

imageView.setScaIeY(mScaleFactor); return true;

Output



**Game Application**

package com.example.tictac;

import android.os.Bundle;

import android.view.View;

import android.widget.ImageView;

import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

boolean gameActive = true;

// Player representation

// 0 - X

// 1 - O

int activePlayer = 0;

int[] gameState = {2, 2, 2, 2, 2, 2, 2, 2, 2};

// State meanings:

// 0 - X

// 1 - O

// 2 - Null

// put all win positions in a 2D array

int[][] winPositions = {{0, 1, 2}, {3, 4, 5}, {6, 7, 8},

{0, 3, 6}, {1, 4, 7}, {2, 5, 8},

{0, 4, 8}, {2, 4, 6}};

public static int counter = 0;

// this function will be called every time a

// players tap in an empty box of the grid

public void playerTap(View view) {

ImageView img = (ImageView) view;

int tappedImage = Integer.parseInt(img.getTag().toString());

// game reset function will be called

// if someone wins or the boxes are full

if (!gameActive) {

gameReset(view);

}

// if the tapped image is empty

if (gameState[tappedImage] == 2) {

// increase the counter

// after every tap

counter++;

// check if its the last box

if (counter == 9) {

// reset the game

gameActive = false;

}

// mark this position

gameState[tappedImage] = activePlayer;

// this will give a motion

// effect to the image

img.setTranslationY(-1000f);

// change the active player

// from 0 to 1 or 1 to 0

if (activePlayer == 0) {

// set the image of x

img.setImageResource(R.drawable.x);

activePlayer = 1;

TextView status = findViewById(R.id.status);

// change the status

status.setText("O's Turn - Tap to play");

} else {

// set the image of o

img.setImageResource(R.drawable.o);

activePlayer = 0;

TextView status = findViewById(R.id.status);

// change the status

status.setText("X's Turn - Tap to play");

}

img.animate().translationYBy(1000f).setDuration(300);

}

int flag = 0;

// Check if any player has won

for (int[] winPosition : winPositions) {

if (gameState[winPosition[0]] == gameState[winPosition[1]] &&

gameState[winPosition[1]] == gameState[winPosition[2]] &&

gameState[winPosition[0]] != 2) {

flag = 1;

// Somebody has won! - Find out who!

String winnerStr;

// game reset function be called

gameActive = false;

if (gameState[winPosition[0]] == 0) {

winnerStr = "X has won";

} else {

winnerStr = "O has won";

}

// Update the status bar for winner announcement

TextView status = findViewById(R.id.status);

status.setText(winnerStr);

}

}

// set the status if the match draw

if (counter == 9 && flag == 0) {

TextView status = findViewById(R.id.status);

status.setText("Match Draw");

}

}

// reset the game

public void gameReset(View view) {

gameActive = true;

activePlayer = 0;

for (int i = 0; i < gameState.length; i++) {

gameState[i] = 2;

}

// remove all the images from the boxes inside the grid

((ImageView) findViewById(R.id.imageView0)).setImageResource(0);

((ImageView) findViewById(R.id.imageView1)).setImageResource(0);

((ImageView) findViewById(R.id.imageView2)).setImageResource(0);

((ImageView) findViewById(R.id.imageView3)).setImageResource(0);

((ImageView) findViewById(R.id.imageView4)).setImageResource(0);

((ImageView) findViewById(R.id.imageView5)).setImageResource(0);

((ImageView) findViewById(R.id.imageView6)).setImageResource(0);

((ImageView) findViewById(R.id.imageView7)).setImageResource(0);

((ImageView) findViewById(R.id.imageView8)).setImageResource(0);

TextView status = findViewById(R.id.status);

status.setText("X's Turn - Tap to play");

}

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

}

}

<?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="@color/green"

tools:context=".MainActivity">

<!--title text-->

<TextView

android:id="@+id/textView"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="23dp"

android:text="GFG Tic Tac Toe"

android:textSize="45sp"

android:textStyle="bold"

app:fontFamily="cursive"

app:layout\_constraintLeft\_toLeftOf="parent"

app:layout\_constraintRight\_toRightOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

<!--image of the grid-->

<ImageView

android:id="@+id/imageView"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:contentDescription="Start"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

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

app:srcCompat="@drawable/grid" />

<LinearLayout

android:id="@+id/linearLayout"

android:layout\_width="0dp"

android:layout\_height="420dp"

android:orientation="vertical"

app:layout\_constraintBottom\_toBottomOf="@+id/imageView"

app:layout\_constraintEnd\_toEndOf="@+id/imageView"

app:layout\_constraintStart\_toStartOf="@+id/imageView"

app:layout\_constraintTop\_toTopOf="@+id/imageView">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_weight="1"

android:orientation="horizontal">

<!--images of the grid boxes-->

<ImageView

android:id="@+id/imageView0"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_weight="1"

android:onClick="playerTap"

android:padding="20sp"

android:tag="0" />

<ImageView

android:id="@+id/imageView1"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_weight="1"

android:onClick="playerTap"

android:padding="20sp"

android:tag="1" />

<ImageView

android:id="@+id/imageView2"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_weight="1"

android:onClick="playerTap"

android:padding="20sp"

android:tag="2" />

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_weight="1"

android:orientation="horizontal">

<ImageView

android:id="@+id/imageView3"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_weight="1"

android:onClick="playerTap"

android:padding="20sp"

android:tag="3" />

<ImageView

android:id="@+id/imageView4"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_weight="1"

android:onClick="playerTap"

android:padding="20sp"

android:tag="4" />

<ImageView

android:id="@+id/imageView5"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_weight="1"

android:onClick="playerTap"

android:padding="20sp"

android:tag="5" />

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_weight="1"

android:orientation="horizontal">

<ImageView

android:id="@+id/imageView6"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_weight="1"

android:onClick="playerTap"

android:padding="20sp"

android:tag="6" />

<ImageView

android:id="@+id/imageView7"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_weight="1"

android:onClick="playerTap"

android:padding="20sp"

android:tag="7" />

<ImageView

android:id="@+id/imageView8"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_weight="1"

android:onClick="playerTap"

android:padding="20sp"

android:tag="8" />

</LinearLayout>

</LinearLayout>

<!--game status text display-->

<TextView

android:id="@+id/status"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="15sp"

android:text="Status"

android:textSize="28sp"

android:textStyle="italic"

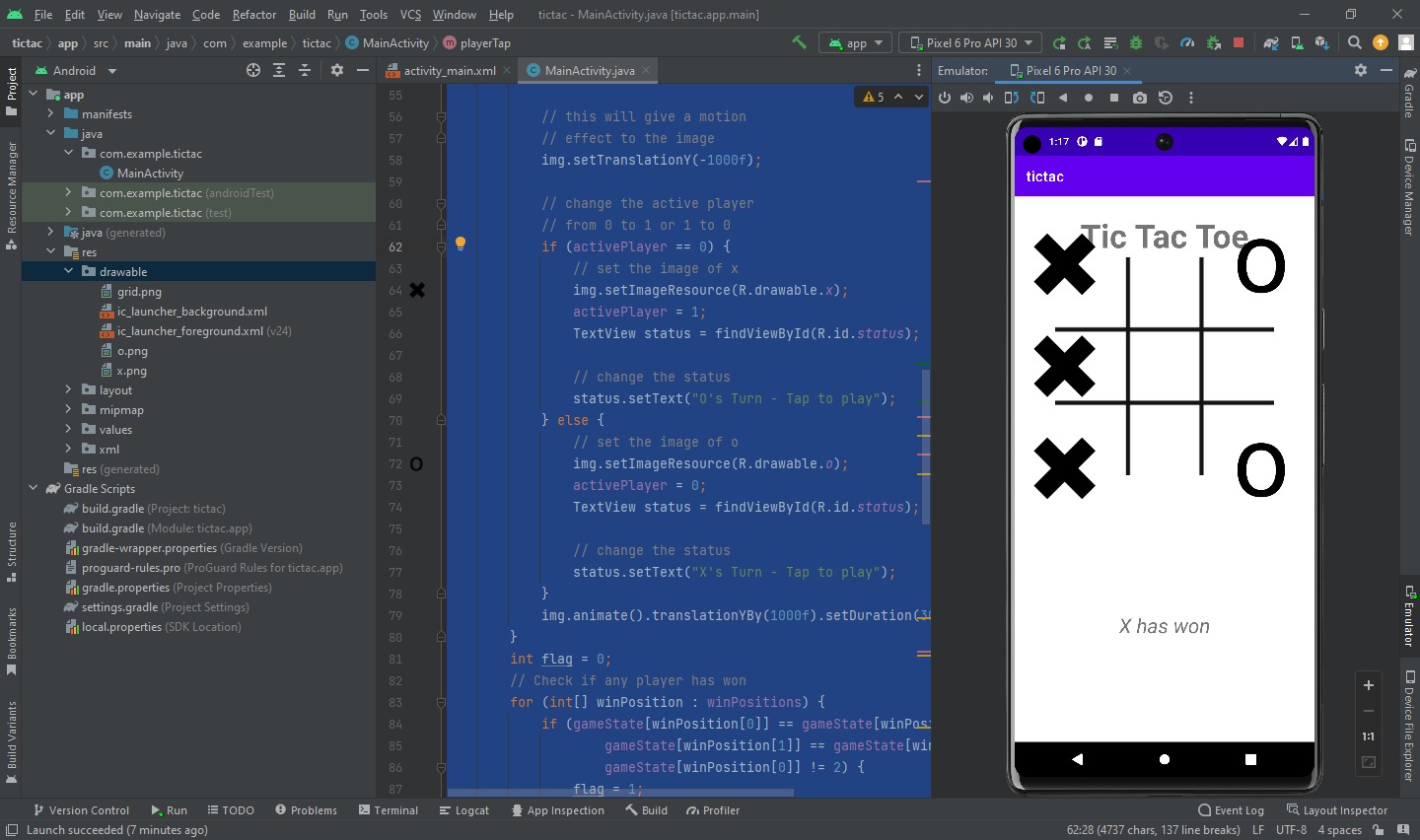
app:layout\_constraintBottom\_toBottomOf="parent"

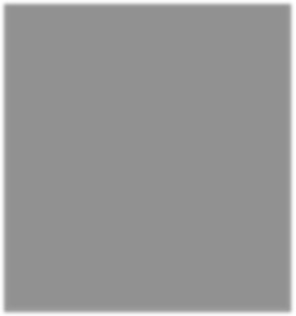
app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

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

</androidx.constraintlayout.widget.ConstraintLayout>





1. **Introduction**:

A netstumbler is a type of software that can be used to detect wireless networks (Wi-Fi) in the vicinity of the computer or device on which it is running. It can be used to identify the names (SSIDs) of wireless networks, the type of encryption used, and the signal strength of the networks. NetStumbler is a popular tool for this purpose, but it is no longer being developed and is not compatible with newer versions of Windows. Alternative tools include inSSIDer, Kismet, andAircrack- ng.

1. **Working:**

NetStumbler works by using the wireless network card in a computer or device to actively scan for wireless networks in the vicinity. When the software is running, it sends out probe requests to identify wireless networks that are within range.

When a wireless network is detected, NetStumbler will gather information about the network, such as the name (SSID), the type of encryption used, the signal

strength, and the MAC address of the access point. This information is then displayed in real-time on the user's computer screen.

NetStumbler also has a built-in GPS receiver, which allows the software to record the location of the wireless networks it detects. This information can be used to create a map of wireless network coverage, which can be helpful for network administrators and for war driving.

Netstumbler uses the 802.11 wireless protocol to detect wireless networks, it sends out probe requests to identify wireless networks that are within range, it doesn't support newer wireless technologies like 802.11ac, 802.11ax.

It's important to note that NetStumbler is only able to detect wireless networks that are configured to broadcast their SSID, or "network name". If a network is configured to not broadcast its SSID, NetStumbler will not be able to detect it.

1. **Uses**:
   * **Wireless network discovery:** NetStumbler can be used to identify wireless networks in the vicinity, including the names of the networks (SSIDs) and the type of encryption used.
   * **Site Surveys:** Netstumbler can be used to conduct wireless site surveys, which are used to identify the best location to set up a wireless accesspoint.
   * **Network troubleshooting:** NetStumbler can be used to diagnose problems with wireless networks, such as poor signal strength or interference from other networks.
   * **Security assessment:** NetStumbler can be used to identify any unauthorized wireless networks that may be operating in a given area, or to identify any potential vulnerabilities in a wireless network.
   * **War driving:** Netstumbler can be used to find and map wireless networks while on the move, this activity is called War driving, it can be used for legal or illegal purposes, it's important to have the authorization of the networks owner before accessing it
2. **Advantages:**
   * **Ease of use:** NetStumbler is a user-friendly tool that is easy to set up and use, even for those with little technical experience.
   * **Real-time data:** NetStumbler provides real-time data on wireless networks in the vicinity, making it easy to identify new or changed networks.
   * **Compatibility:** NetStumbler is compatible with a wide range ofwireless network cards and operating systems.
   * **Free:** NetStumbler is a free and open-source tool.
3. **Disadvantages:**
   * **Incompatibility:** NetStumbler is not compatible with newer versions of Windows, which limits its usefulness for some users.
   * **Limited features:** NetStumbler has a limited set of features compared to more advanced wireless network analysis tools.
   * **Security issues:** Some newer wireless networks use security protocols that NetStumbler is not able to detect, which can lead to false information.
   * **Legal issues:** War driving or accessing to unauthorized networks can lead to legal issues, its important to have the authorization of thenetworkowner before accessing it.
   * **Limited to 802.11b networks:** NetStumbler is limited to only detecting 802.11b wireless networks, and it doesn't support newer wireless technologies.