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 =
1000010101110101100010111110101110111110010011100110101000101111011100100111
011
100010101111001011100001001110110011010000000

128 Random Bits Generated =
11011110110100011001100111111000100001011001000111101100110111111100110100011010
011011010011101011011101100011101011001101000

RES/SRES = 10000010010100101111101010110101
```

Application using database

AndroidManifest.xml

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

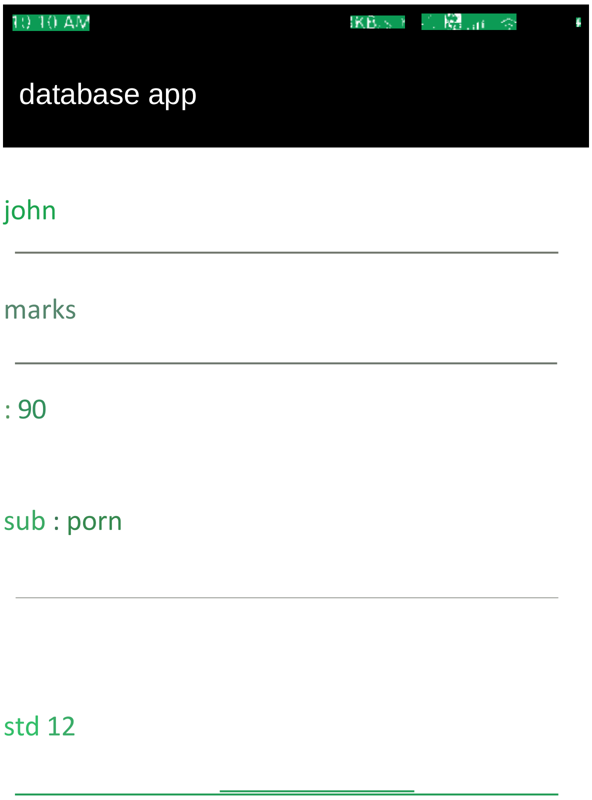
activity_main.xml

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

<!--Edit text to enter course name-->
<EditText
android:id="@+id/idEdtCourseName" android:layout_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:layout_height="wrap_content" android:layout margin="10dp" android:hint="Enter Course
Duration" />

<!--edit text to display course tracks-->
<EditText
android:id="@+id/idEdtCourseTracks" android:layout_width="match_parent" android:layout height="wrap
content" android:layout_margin="10dp" android:hint="Enter Course Tracks" />

<!--edit text for course description-->
<EditText
android:id="@+id/idEdtCourseDescription” android:layout width="match parent"
android:layout_height="wrap_content"

android:layout_margin="10dp" android:hint="Enter Course Description" />

<!--button for adding new course-->
<Button
android:id="@+id/idBtnAddCourse" android:layout_width="match_parent" android:layout height="wrap content"
android:layout margin="10dp" android:text="Add Course" android:textAllCaps="false" />

</LinearLayout>
```

DBHandler

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

android.database.sqlite.SQLiteOpenHelper; public class DBHandler extends SQLiteOpenHelper

(

// creatingaconstant variables forourdatabase. // below variable is for our database name. privatestaticfinalString
DB_NAME="coursedb ,

I/ below int is our database version privatestaticfinalintDB_VERSION =1;

// below variable is for our table name.
```

```
private static final String TABLE_NAME = "mycourses";
// below variable is for our id column. private staticfinal StringID_COL ="id";

I/ belowvariable is forourcoursename column privatestaticfinalStringNAME_COL="name",

// below variable id for our course duration column. private static final String DURATION_COL = "duration";

// 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";

// 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) {
// on below line we are creating
// an sqlite query and we are // setting our column names // 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)";

// at last we are calling a exec sql
// 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) (

// on below line we are creating a variable for
// our sqlite database and calling writable method
// as we are writing data in our database. SQLiteDatabase db = this.getWritabIeDatabase();

// on below line we are creating a
// variable for content values.
ContentValues values = new ContentValues();

// 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);

// after adding allvalues wearepassing // content values to our table. db.insert(TABLE_NAME, null, values);

// at last we are closing our
// database after adding database. db.close();



@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) { //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);

// 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);

// creating a new dbhandler class
// and passing our context to it.
dbHandler = new DBHandler(MainActivity.this);

// below line is to add on click listener for our add course button.
addCourseBtn.setOnClickListener(new View.OnClickListener() (
@Override
public void onClick(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( );

                    });
```



john

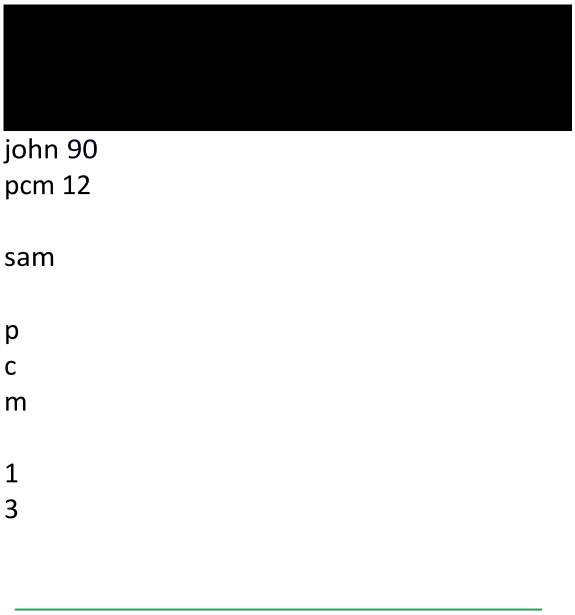
marks

: 90

sub : porn

std 12

Course has been added.


[redacted]

john 90
pcm 12

sam

p
c
m

1
3

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);
  }
}
```

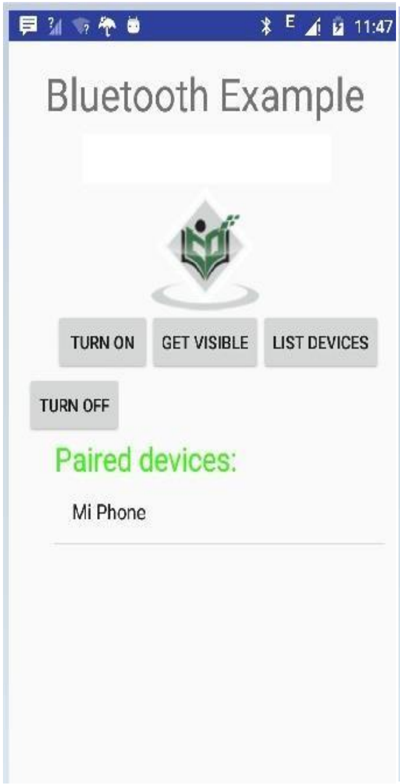
activity_main.xml

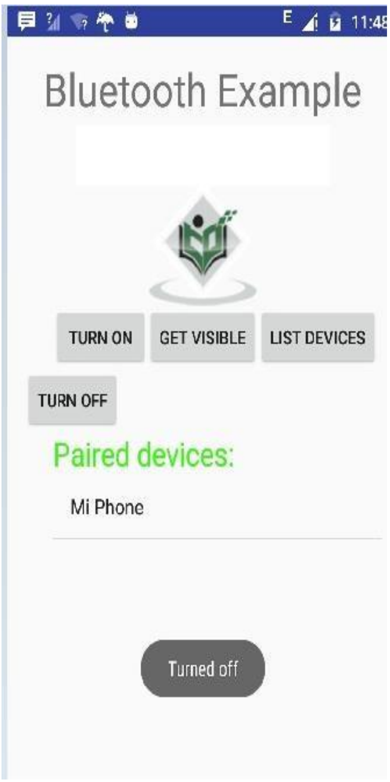
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
```

```
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"
```

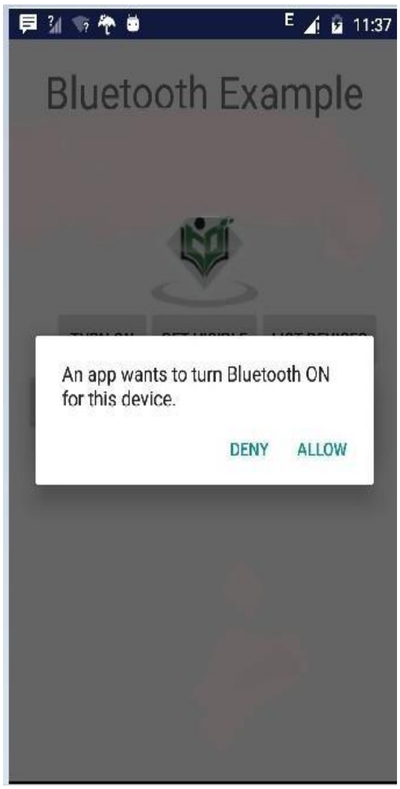
```
    android:layout_toRightOf="@+id/imageView"
    android:layout_toEndOf="@+id/imageView" />
  <Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="turn off"
    android:onClick="off"
    android:id="@+id/button4"
    android:layout_below="@+id/button"
    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 :





Turned off



An app wants to turn Bluetooth ON for this device.

Alert Message

```
package com.example.notificationdemo; import
android.app.Activity; import android.app.NotificationManager;
importandroid.app.PendingIntent;importandroid.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 savedInstancestate) ( super.onCreate(savedInstancestate);
setContentView(R.layout.activity_main);

b1 = (Button)findViewById(R.id.button); b1.setOnClickListener(new View.OnClickListener()
{ @Override
publicvoidonClick(Viewv){ addNotification();

    });


private void addNotification() { NotificationCompat.Builder
builder= new NotificationCompat.Builder(this)
.setSmallIcon(R.drawable.abc)
.setContentTitle("Notifications Example")
.setContentText("This is a test notification");

Intent notificationIntent = new Intent(this, MainActivity.class);
PendingIntent contentIntent = PendingIntent.getActivity(this, 0,
notificationIntent, PendingIntent.FLAG_UPDATE_CURRENT);
builder.setContentIntent(contentIntent);

/I Add as notification

NotificationManager manager =
(NotificationManager)
getsystemservice(Context.NOTIFICATION_SERVICE)
; manager.notify(0, builder.build());
```

notification.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
android:orientation="vertical"
android:layout width="fill parent" android:layout height="fill parent" >

<TextView android:layout_width="fill_parent"
android:layout_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 savedInstancestate)( super.onCreate(savedInstancestate);
setContentView(R.layout.notification);
```

activity_main.xml

```
<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:paddingBottom="@dimen/activity_vertical_margin"
android:paddingLeft="@dimen/activity horizontal margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity vertical margin"
tools:context="MainActivity">


<TextView android:id="@+id/textView1" android:layout_width="wrap_content"
android:layout height="wrap content" android:text="Notification Example"
android:layout_alignParentTop="true" android:layout centerHorizontal="true"
android:textSize="30dp" />

<TextView android:id="@+id/textView2" android:layout_width="wrap_content"
android:layout height="wrap content" android:text="Tutorials point " android:textColor=”#ff87ff09"
android:textSize="30dp"
android:layout below="@+id/textView1" android:layout_centerHorizontal="true"
android:layout_marginTop="48dp" />

<ImageButton
android:layout width="wrap content" android:layout_height="wrap_content"
android:id="@+id/imageButton" android:src="@drawable/abc" android:layout
below="@+id/textView2" android:layout_centerHorizontal="true" android:layout_marginTop="42dp"
/>

<Button
android:layout width="wrap content" android:layout_height="wrap_content"
android:text="Notification" android:id="@+id/button" android:layout
marginTop="62dp" android:layout_below="@+id/imageButton" android:layout
centerHorizontal="true" />

</RelativeLayout>
```

strings.xml

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<resources>
<string name="action_settings">Settings</string>
<string name="app name">tutorialspoint </strings
</resources>
```

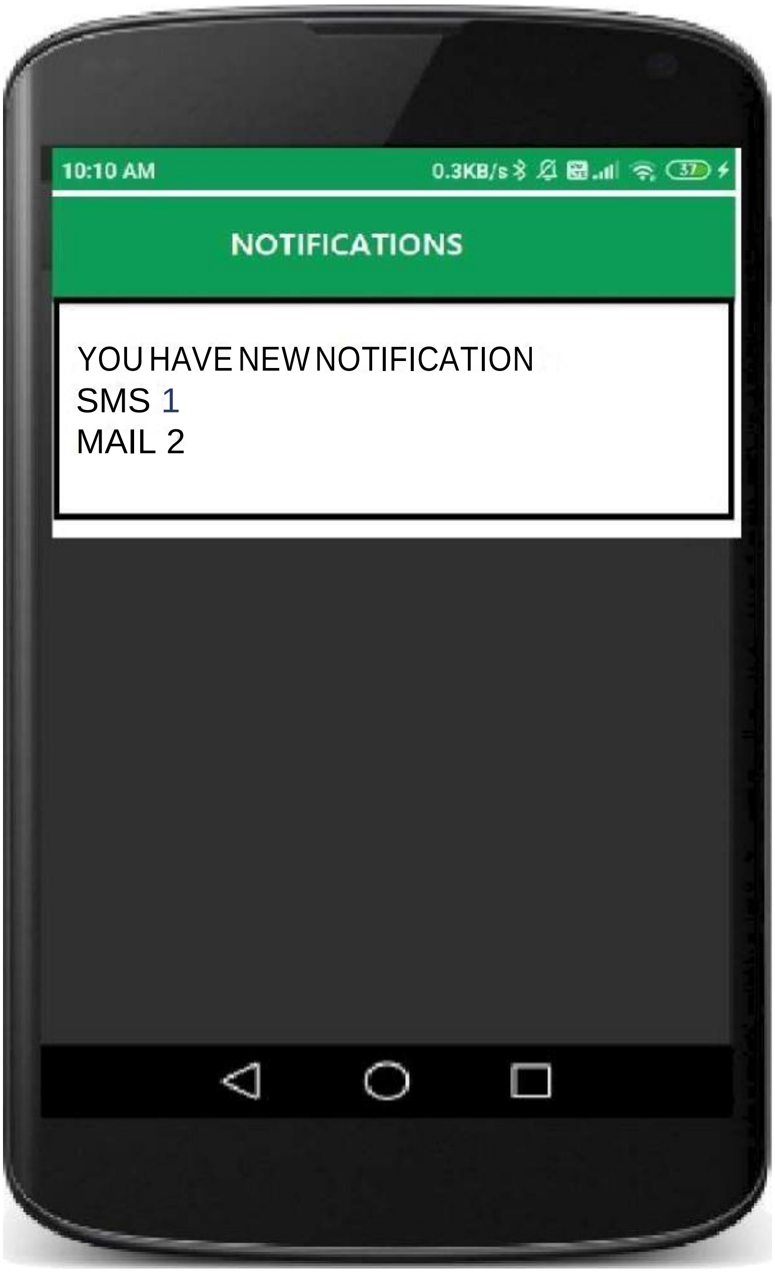
AndroidManifest.xml

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

<application android:allowBackup="true"
android:icon="@drawable/ic_launcher" 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:label="Details of
notification" android:parentActivityName=".MainActivity">
<meta-data android:name="android.support.PARENT ACTIVITY" android:value=".MainActivity"/>
</activity>

</application>
</manifest>
```

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

/* Invoking the default notification service */
NotificationCompat.Builder mBuilder = new NotificationCompat.Builder(this);

mBuilder.setContentTitle("New Message"); mBuilder.setContentText("You’ve received new
message."); mBuilder.setTicker("New Message Alert!");
mBuilder.setSmallIcon(R.drawable.woman);

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

/*AddBigViewSpecific Configuration */
NotificationCompat.InboxStyle inboxstyle = new NotificationCompat.InboxStyle();

String[] events = new String[6];
events[0]=newString("This isfirstline.
                                       ")
; 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 resultIntent = new Intent(this,
NotificationView.class);

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

/* Adds the Intent that starts the Activity to the top of the stack */
stackBuilder.addNextIntent(resultIntent);
PendingIntent resultPendingIntent
=stackBuiIder.getPendingIntent(0,PendingIntent.FLAG UPDATE CURRENT);

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

```
/* notificationID allows you to update the notification later
on. */ mNotificationManager.notify(notificationID,
mBuilder.build());
```



NOTIFICATIONS

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"
android:layout_width="match_parent"
xmlns:tooIs="http://schemas.android.com/tools"
xmlns:app="http://schemas.android.com/apk/res-auto"
tools:context="abu.emicaIculator.MainActivity"
android:layout height="match parent">
<android.support.v4.widget.NestedScrollView android:layout_width="match_parent" android:layout
height="match parent"
app:layout_behavior="@string/appbar_scroIling_view_behavior">
<LinearLayout
android:layout width="fiII parent" android:layout height="match parent"
android:layout_marginTop="?attr/actionBarSize" android:orientation="verticaI"
android:paddingLeft="20dp" android:paddingRight="20dp"
android:paddingTop="10dp">

<android.support.design.widget.TextInputLayout
android:id="@+id/input_Iayout_principal" android:layout width="match parent"
android:layout_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.TextInputLayout android:id="@+id/input layout interest"
android:layout_width="match_parent"

android:layout_height="wrap_content">

<EditText android:id="@+id/interest"
android:layout width="match parent" android:layout_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.TextInputLayout
android:id="@+id/input_Iayout_tenure" android:Iayout_width="match_parent"
android:layout height="wrap content">

<EditText android:id="@+id/years"
android:layout_width="match parent" android:layout_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:layout_height="wrap_content" android:text="Calculate"
android:background="@color/colorPrimary"
android:layout_marginTop="40dp"

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


<android.support.design.widget.TextInputLayout android:id="@+id/input layout
emi" android:layout_width="match_parent" android:layout height="wrap
content" android:layout_marginTop="40dp">

<EditText

android:id="@+id/emi" android:layout_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.TextInputLayout android:id="@+id/input_layout_total_Interest"
android:layout_width="match_parent" android:layout height="wrap content"
android:layout_marginTop="10dp">

<EditText android:id="@+id/interest_total" 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.NestedScrollView>

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

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

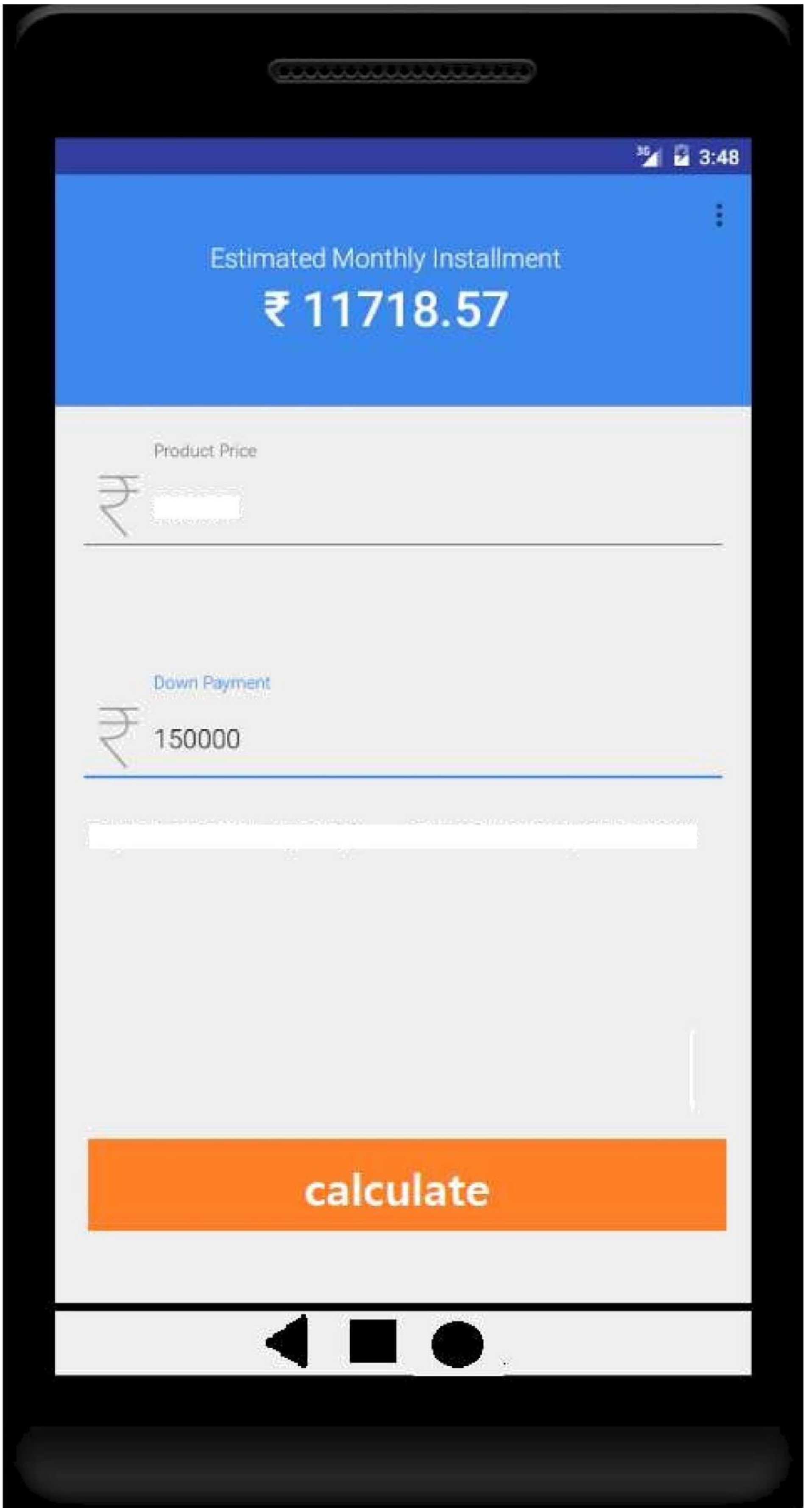
string XML Code

```
<resources>
<string name="app_name">EMI Calculators/strings
<string name="hint_principal">PrincipalAmount 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 total">Total Interest for Loan Z</strings
</resources>
```

Main Activity.java.

```
package abu.emicalculator; import android.os.Bundle;
import android.support.v7.app.AppCompatActivity; import android.text.TextUtils;
importandroid.view.View;importandroid.widget.Button;import
android.widget.EditText; public class MainActivity extends AppCompatActivity {
Button emiCalcBtn;
@Override
protected void onCreate(Bundle savedInstancestate) ( super.onCreate(savedInstancestate);
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_total); final EditText result =
(EditText) findViewByld(R.id.emi); emiCalcBtn = (Button) findViewByld(R.id.btn
calculate2); emiCalcBtn.setOnClickListener(new View.OnClickListener() ( @Override
public void onClick(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.parseFloat(st1); float i = Float.parseFloat(st2); float y = Float.parseFloat(st3); float Principal
= calPric(p); float Rate =
calInt(i); float Months =
calMonth(y);

float Dvdnt = calDvdnt(Rate, Months);
float FD = calFinalDvdnt(Principal, Rate, Dvdnt); float D =
calDivider(Dvdnt); float emi = calEmi(FD, D); float TA = calTa(emi,
Months);
float ti = calTotalInt(TA, Principal); result.setText(String.valueOf(emi)); TI.setText(String.valueOf(ti));

});

publicfloatcalPric(floatp)( return (float)(p);

public float calInt(float i)( return (float)(i / 12 /

100); publicfloatcalMonth(floaty)( return

(float)(y * 12);

publicfloatcalDvdnt(floatRate,floatMonths){ return (float)(Math.pow(1 + Rate, Months));

publicfloatcalFinalDvdnt(floatPrincipal,floatRate,floatDvdnt){ return (float)(Principal * Rate * Dvdnt);

public float calDivider(float Dvdnt) ( return (float)(Dvdnt - 1);

publicfloatcalEmi(float FD,FloatD)( return (float)(FD / D);

publicfloatcalTa(floatemi,FloatMonths){ return (float)(emi * Months);

public float calTotalInt(float TA, float Principal) { return (float)(TA -

Principal);
```



3:48

Estimated Monthly Installment

₹ 11718.57

Product Price

Down Payment

150000

Photo gallery

AndroidManifest.xml

```
<!-- permissions for reading external storage -->
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
android:hardwareAccelerated="false"
android:largeHeap="true"
```

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

<!--recycler view for displaying the list of images-->
<androidx.recyclerview.widget.RecyclerView android:id="@+id/idRVImages"
android:layout_width="match_parent" android:layout_height="match_parent" />
</RelativeLayout> layout Resource file
<?xml version="1.0" encoding="utf-8"?>
<androidx.cardview.widget.CardView
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
android:layout_width="wrap_content” android:layout_height="wrap_content"
android:layout gravity="center" android:layout_margin="3dp" android:elevation="8dp"
app:cardCornerRadius="8dp">

<!--Image view for displaying the
image in our card layout in recycler
view-->
<ImageView
android:id="@+id/idIVIma
ge"

android:layout_width="100dp" android:layout_height="100dp" android:layout
gravity="center" android:scaleType="centerCrop" />

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

Empty Activity > New Java class import android.content.Context; import android.content.Intent; 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.squareup.picasso.Picasso;
import java.io.File; import java.util.ArrayList;

public class RecyclerViewAdapter extends
RecyclerView.Adapter<RecyclerViewAdapter.RecyclerViewHolder> {

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 RecyclerViewAdapter(Context context, ArrayList<String> imageP athArrayList) { this.context =
context;
this.imagePathArrayList = imagePathArrayList;


@NonNull @Override
public RecyclerViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
                    // Inflate Layout in this method which we have created.
                    View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.card_layout, return
                    new
parent, false);     RecyclerViewHolder(view);


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

// on below line we are getting the file from the
// path which we have stored in our list.
File imgFile = new File(imagePathArrayList.get(position));

// on below line we are checking if the file exists or not. if (imgFile.exists()) (

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

Picasso.get().load(imgFile).placeholder(R.drawable.ic_launcher_background).into(holder.imagel V);

// on below line we are adding click listener to our item of recycler
view. holder.itemView.setOnClickListener(new
View.OnClickListener() ( @Override
public void onClick(View v) (

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

        /f on below line we are passing the image path to our new
activ   i.putExtra("imgPath", imagePathArrayList.get(position));
ity.    // at last we are starting our activity. context.startActivity(i);

                        });



        @Override
        public int getItemCount() (
        // this method returns
        // the size of recyclerview
        return imagePathArrayList.size();
```

```
// View Holder Class to handle Recycler View.
public static class RecyclerViewHolder extends RecycIerView.ViewHoIder (

IIcreating variables forourviews. privatefinalImageViewimageIV;

public RecycIerViewHoIder(@NonNuII View itemView) ( super(itemView);
/f initializing our views with their ids.
imageIV = itemView.findViewById(R.id.idlVImage);
```

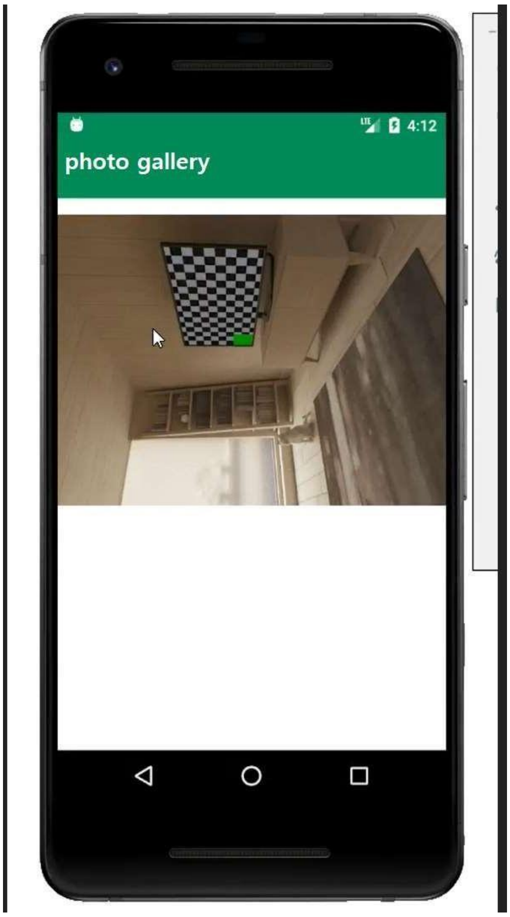
MainActivity.java

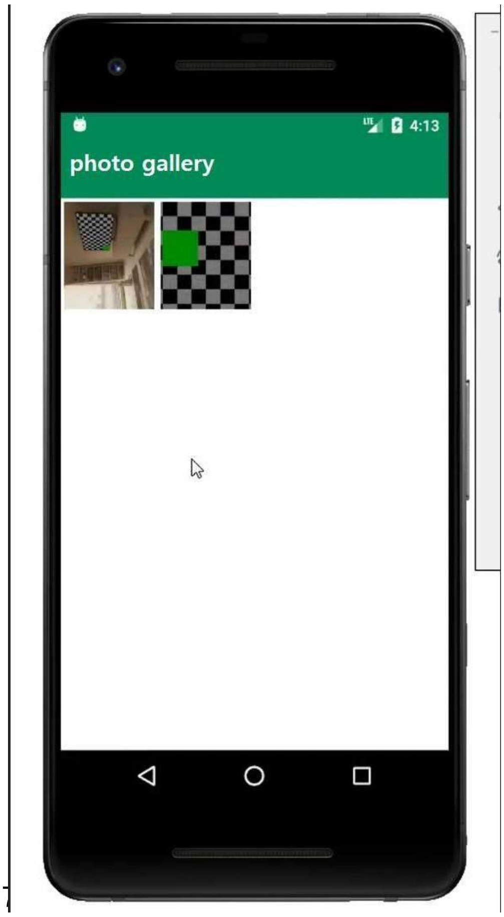
```
import android.content.pm.PackageManager; import android.database.Cursor;
import android.os.Bundle;
import android.provider.MediaStore; import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity; import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;
import androidx.recyclerview.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 savedInstancestate) ( super.onCreate(savedInstancestate);
setContentView(R.layout.activity_main);



II creating a new array list and
II initializing our recycler view. imagePaths = new
ArrayList<>(); imagesRV = findViewByld(R.id.idRVImages);

II we are calling a method to request
IIthepermissions toreadexternalstorage. requestPermissions();

II calling a method to
IIprepareourrecyclerview. 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 a method to get all images from our external 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.getExternalStorageState().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 bystring.
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);

/I onbelowlinewe are getting image file path int
dataColumnIndex =
cursor.getColumnIndex(MediaStore.Images.Media.DATA);

/I after that we are getting the image file path
/I and adding that path in our array list. imagePaths.add(cursor.getString(dataColumnIndex));

imageRVAdapter.notifyDataSetChanged();
// after adding the data to our
f/ array listwe are closing ourcursor. cursor.close();


@Override
public void onRequestPermissionsResult(int requestCode, String permissions[], int[] grantResults) (
IIthismethod iscalledafterpermissions hasbeengranted. switch (requestCode) (
//we arechecking thepermission code. casePERMISSION REQUEST CODE:
/I in this case we are checking if the permissions are
accepted or not.
if (grantResults.length > 0) (
boolean storageAccepted =
grantResults[0] ==
PackageManager.PERMISSION_GRANTED;
if (storageAccepted) (
// if the permissions are accepted we are
displaying a toast message
                                         // and calling a method to get image path.
                                         Toast.makeText(this,"Permissions Granted..",
Toast.LENGTH_SHORT).show
();                                      getImagePath();
                                 } else{
                                         // if permissions are denied we are closing the app

anddisplayingthetoastmessa
ge.
```

```
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/andr
oid"
xmlns:tooIs="http://schemas.android.com/tools"
android:Iayout_width="match_parent" android:Iayout_height="match_parent"
tools:context=".ImageDetailActivity">

<!--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 (

// creating a string variable, image view variable
// and a variable for our scale gesture detector class. String imgPath;

private ImageView imageView;
private ScaleGestureDetector scaleGestureDetector;

// 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);

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

// initializing our image view.
imageView = findViewByld(R.id.idlVlmage);
for our image.
```

```
  // on below line we are initializing our scale gesture detector for zoom in and out
  scaleGestureDetector = new ScaleGestureDetector(this, new ScaleListener());
  // on below line we are getting our image file from its path. File imgFile = new File(imgPath);

  // if the file exists then we are loading that image in our image view. if (imgFile.exists()) {
        Picasso.get().load(imgFile).placeholder(R.drawable.ic_launcher_background).into(imageView);

        @Override
        public boolean onTouchEvent(MotionEvent motionEvent) (
        // inside on touch event method we are calling on
        // touch event method and passing our motion event to it.
        scaleGestureDetector.onTouchEvent(motionEvent); return
        true;

        private class ScaleListener extends ScaleGestureDetector.SimpleOnScaleGestureListener (
        // on below line we are creating a class for our scale // listener and extending it with gesture
        listener. @Override
 public boolean onScale(ScaleGestureDetector scaleGestureDetector) (

 // inside on scale method we are setting scale
 // for our image in our image view.
 mScaleFactor *= scaleGestureDetector.getScaleFactor(); 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.setScaleX(mScaleFactor);
imageView.setScaleY(mScaleFactor); return true;
```





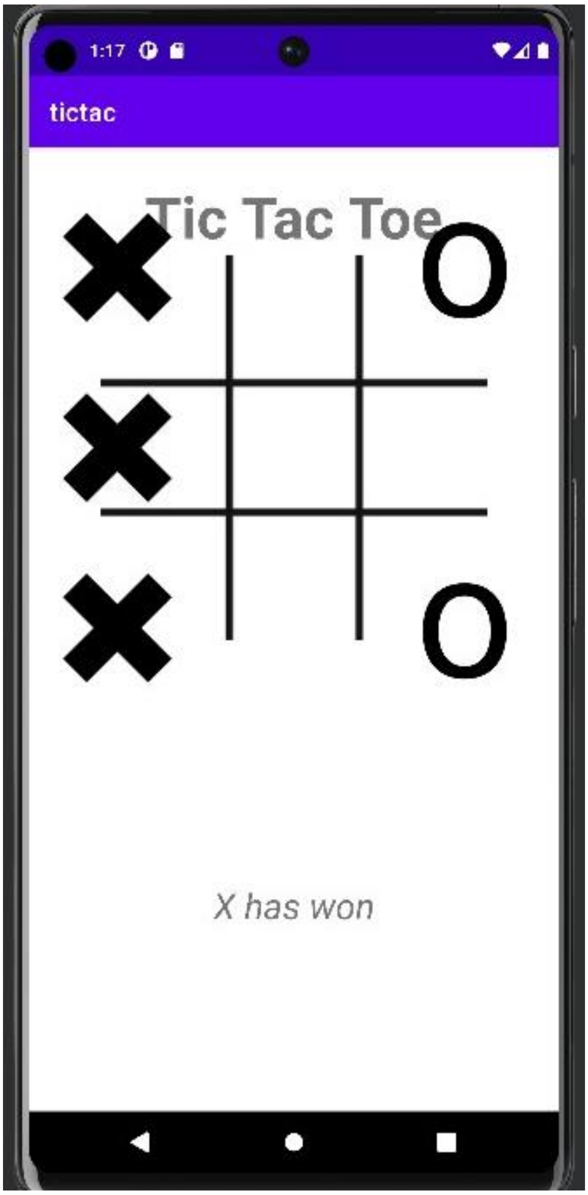
Game Application

```
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>
```



Tic Tac Toe

X has won

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.

2. 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.

3. 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

4. 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.

5. 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.