SRI CHANDRASEKHARENDRA SARASWATHI VISWA MAHAVIDYALAYA

(UNIVERSITY ESTABLISHED UNDER SECTION 3 OF UGC ACT 1956)
ENATHUR, KANCHIPURAM – 631 561

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

MOBILE APPLICATION DEVELPOMENT LAB

Name : _____

Reg. No : _____

Class : **IV- B.E. (CSE)**

Subject Code : **CS7P8**

SRI CHANDRASEKHARENDRA SARASWATHI VISWA MAHAVIDYALAYA

(UNIVERSITY ESTABLISHED UNDER SECTION 3 OF UGC ACT 1956)
ENATHUR, KANCHIPURAM – 631 561

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this is the bonafide record of work done by **Mr/Ms.**_____ , with **Reg. No** _____ of **IV Year B.E. (Computer Science and Engineering)** in the **Mobile Application Development lab (CS7P8)** during the academic year 2018.

Station:

Date:

Staff-in-charge **Head of the Department**

Submitted for the Practical examination held on _____ **.**

Internal Examiner **External Examiner**

INDEX

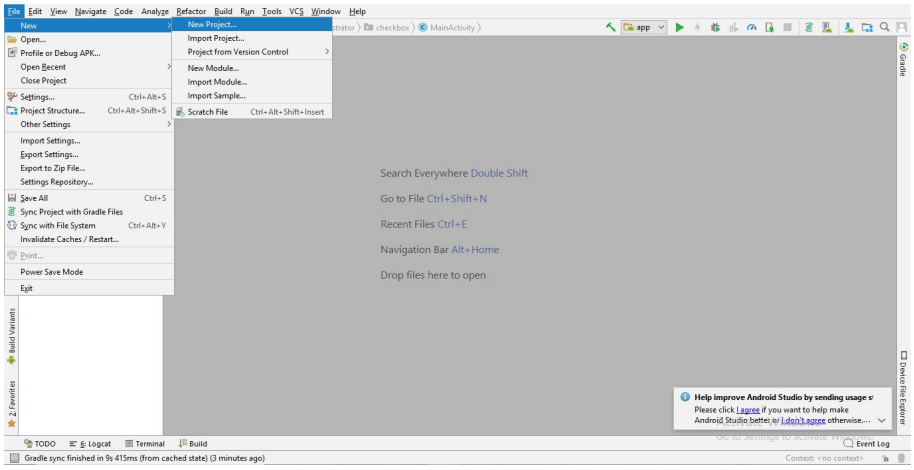
Exp.No:1	DEVELOP AN APPLICATION THAT USES GUI COMPONENTS, FONT AND COLORS
Date:06-07-18	

Aim:

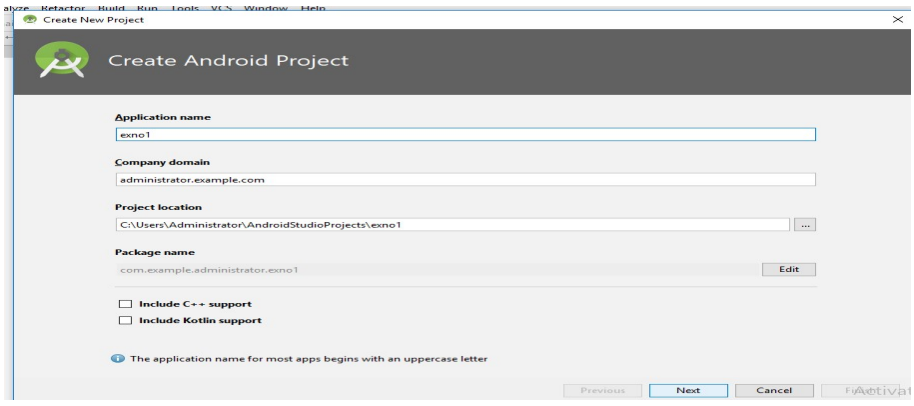
To develop a Simple Android Application that uses GUI components, Font and Colors.

Algorithm:

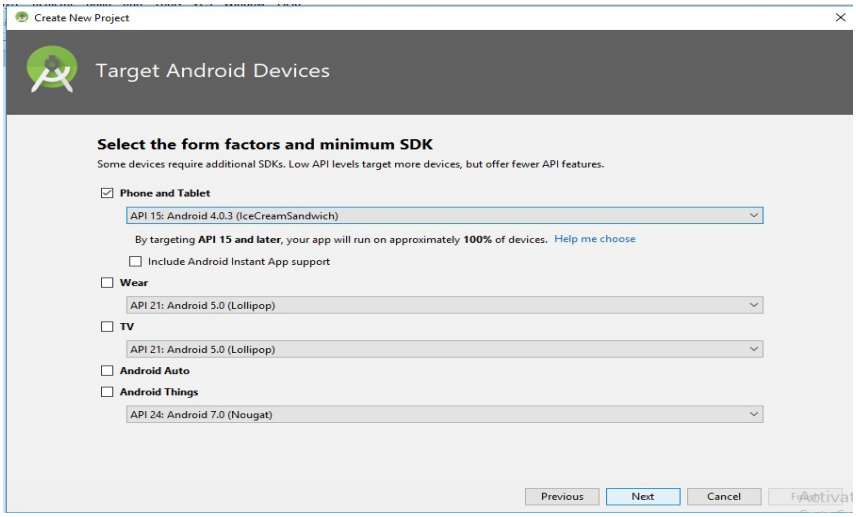
- Start the program.
- Creating a New project:
    a. Open Android Studio and then click on **File -> New -> New project.**



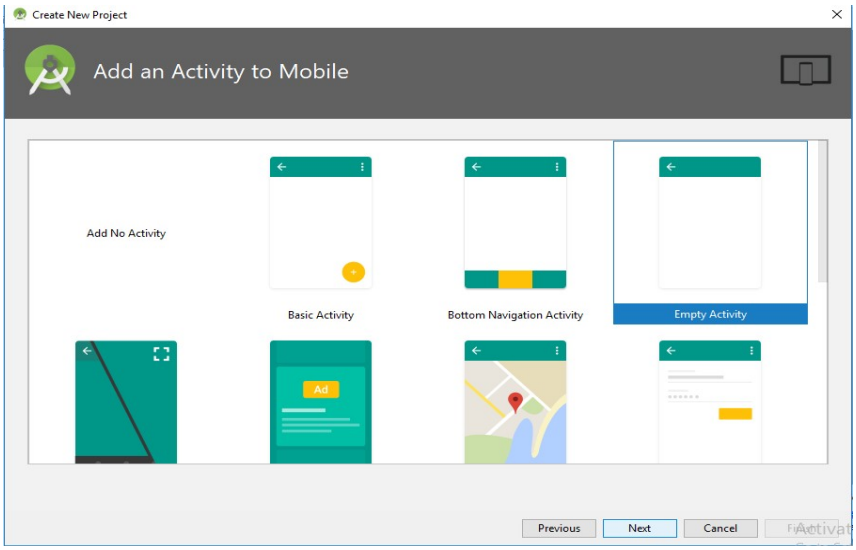
    b. Then type the Application name and click **Next.**



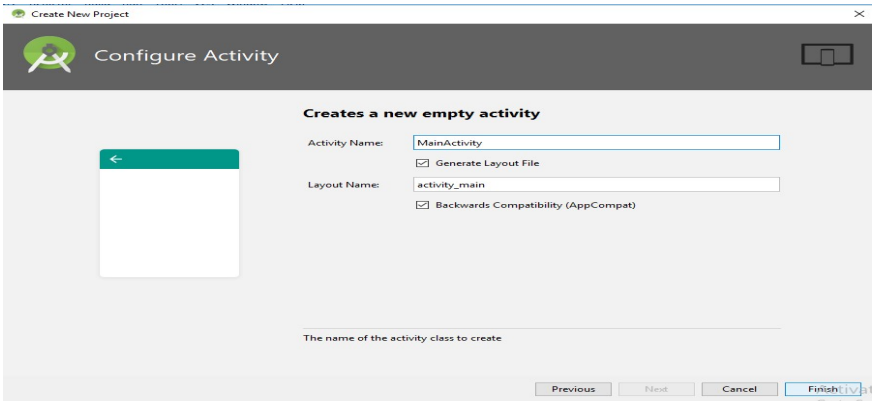
    c. Then select the **Minimum SDK** as shown below and click **Next**.



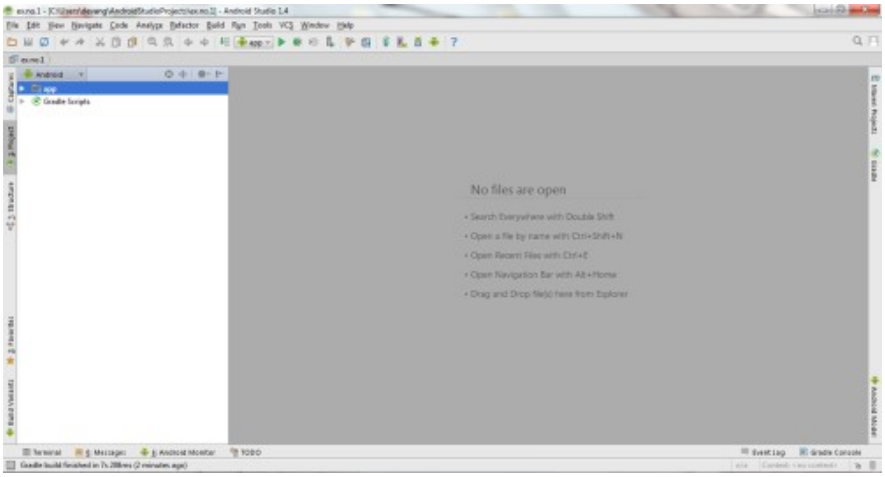
d. Then select the **Empty Activity** and click **Next.**



e. Finally click F**inish**.

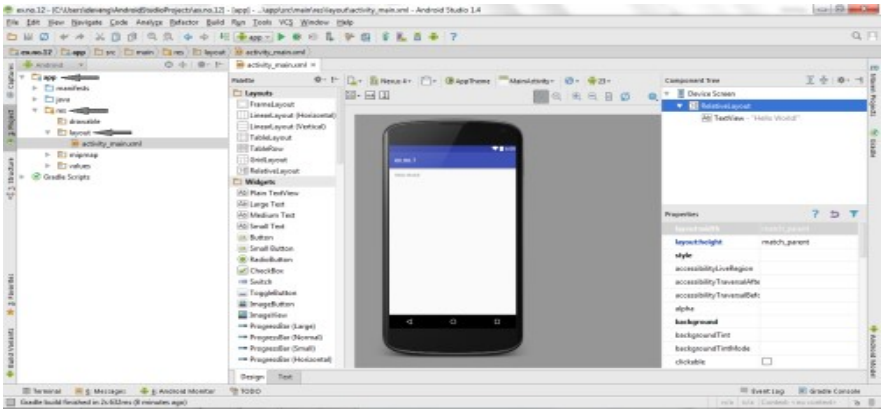


f. It will take some time to build and load the project.
g. After completion it will look as given below.

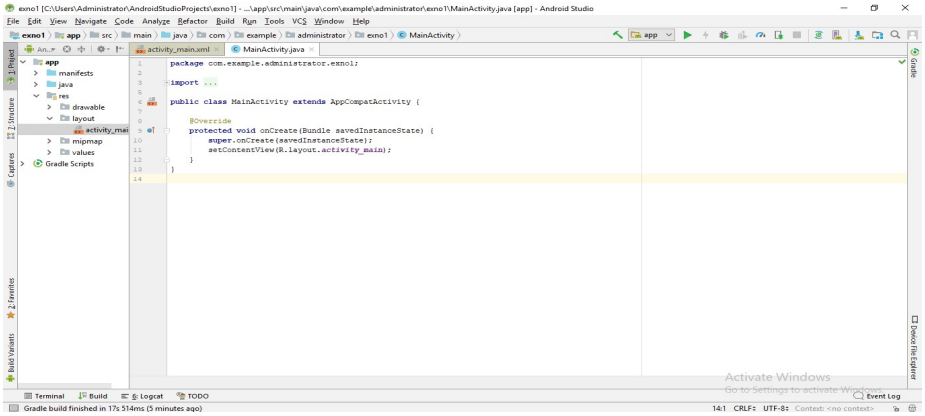


- Designing layout for the Android Application.

  Click on **app -> res -> layout -> activity_main.xml**



- Now click on **Text** as shown below.



- Then delete the code which is there and type the code as given below

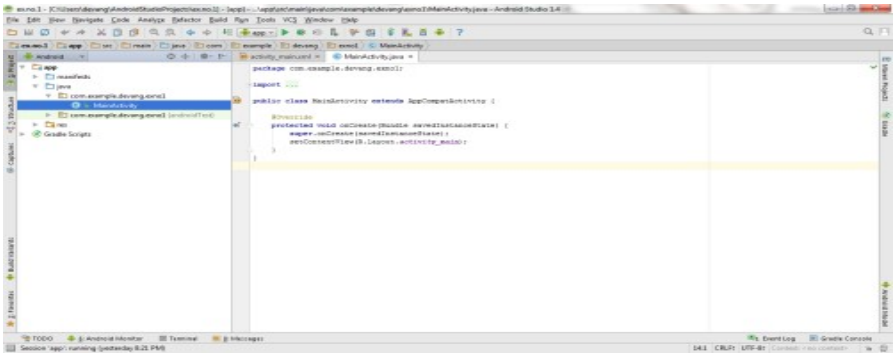
Code for Activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:orientation="vertical"
   android:layout_width="match_parent"
   android:layout_height="match_parent">
   <TextView
      android:id="@+id/textView"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:layout_margin="30dp"
      android:gravity="center"
      android:text="Hello World!"
      android:textSize="25sp"
      android:textStyle="bold" />
   <Button
      android:id="@+id/button1"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:layout_margin="20dp"
      android:gravity="center"
      android:text="Change font size"
      android:textSize="25sp" />
   <Button
      android:id="@+id/button2"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:layout_margin="20dp"
      android:gravity="center"
      android:text="Change color"
      android:textSize="25sp" />
</LinearLayout>
```

- So now the designing part is completed

Java Coding for the Android Application:

Click on **app -> java -> com.example.exno1 -> MainActivity.**



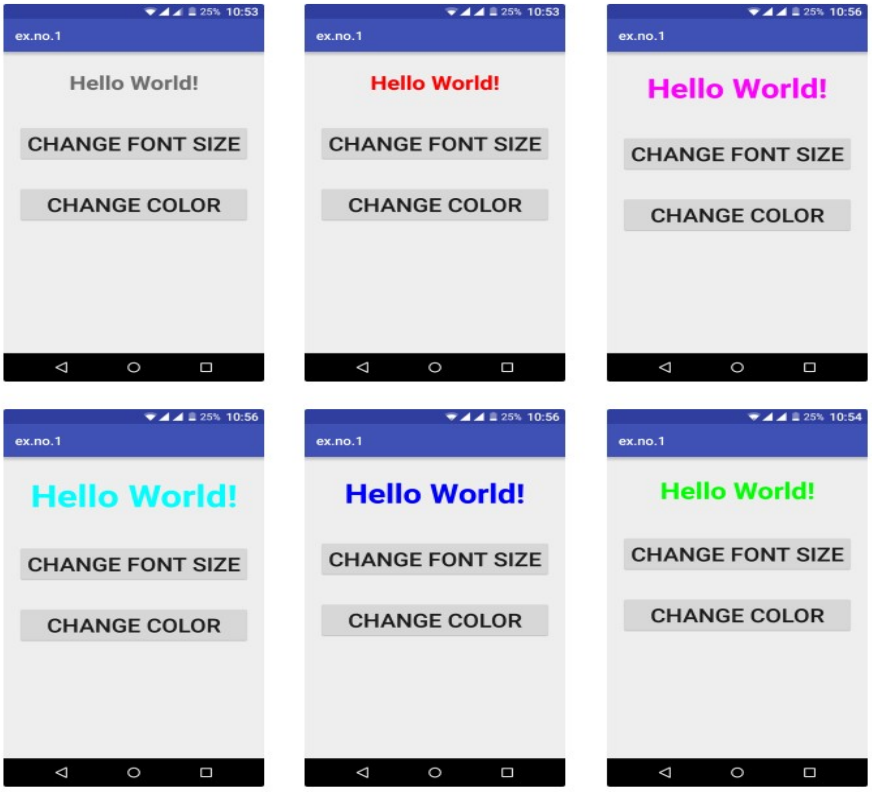
Code for MainActivity.java:

```
package com.example.exno1;
import android.graphics.Color;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
public class MainActivity extends AppCompatActivity
{
   int ch=1;
   float font=30;
   @Override
   protected void onCreate(Bundle savedInstanceState)
   {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      final TextView t= (TextView) findViewById(R.id.textView);
      Button b1= (Button) findViewById(R.id.button1);
      b1.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View v) {
            t.setTextSize(font);
            font = font + 5;
            if (font == 50)
               font = 30;
         }  });
      Button b2= (Button) findViewById(R.id.button2);
      b2.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View v) {
            switch (ch) {
               case 1:
                  t.setTextColor(Color.RED);
                  break;
               case 2:
                  t.setTextColor(Color.GREEN);
                  break;
               case 3:
                  t.setTextColor(Color.BLUE);
                  break;
               case 4:
                  t.setTextColor(Color.CYAN);
```

```
            break;
        case 5:
            t.setTextColor(Color.YELLOW);
            break;
        case 6:
            t.setTextColor(Color.MAGENTA);
            break;
    }
    ch++;
    if (ch == 7)
        ch = 1;
} });   } }
```

- So now the Coding part is also completed.
- Now run the application to see the output.
- Stop the program.

Output:



Result:

Thus a Simple Android Application that uses GUI components, Font and Colors is developed and executed successfully.

EXP.NO:2	DEVELOP AN APPLICATION THAT USES LAYOUT MANAGERS AND EVENT LISTNERS.
DATE: 13-07-18	

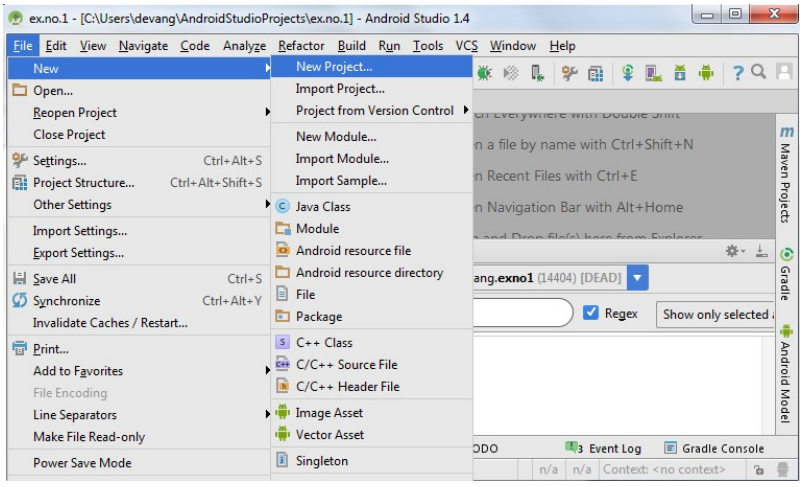
Aim:

To develop a Simple Android Application that uses Layout Managers and Event Listeners.

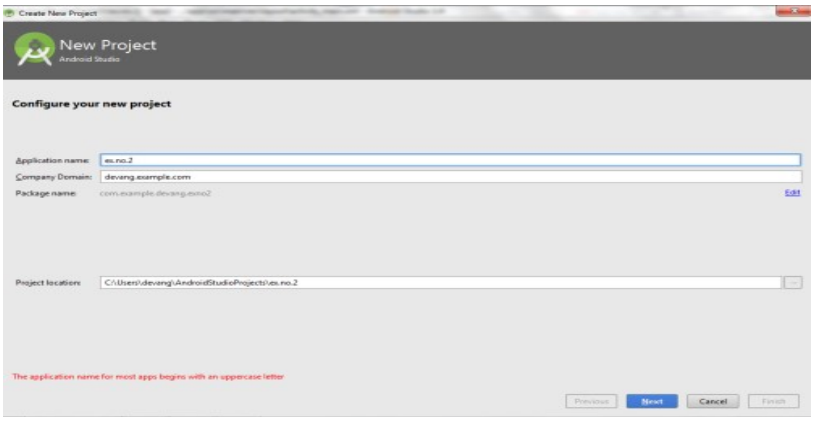
Algorithm:

1. Start the program.
2. Creating a New project.

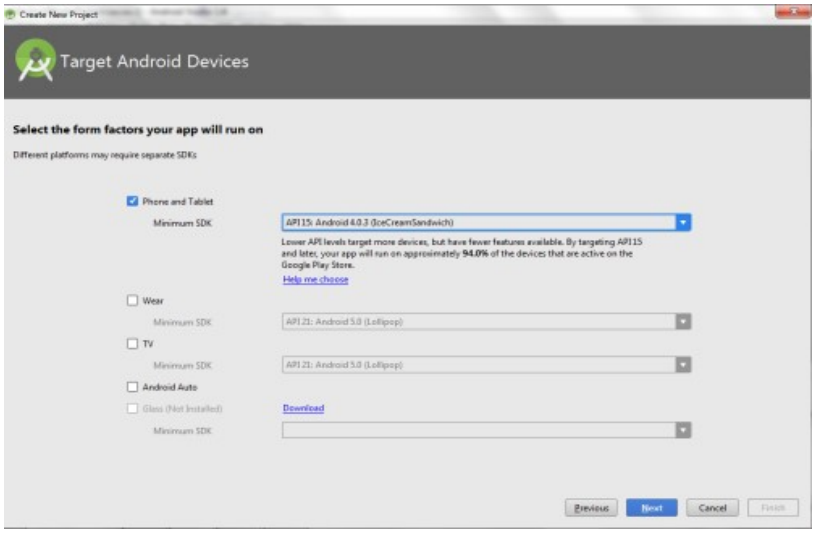
- Open Android Stdio and then click on **File -> New -> New project.**



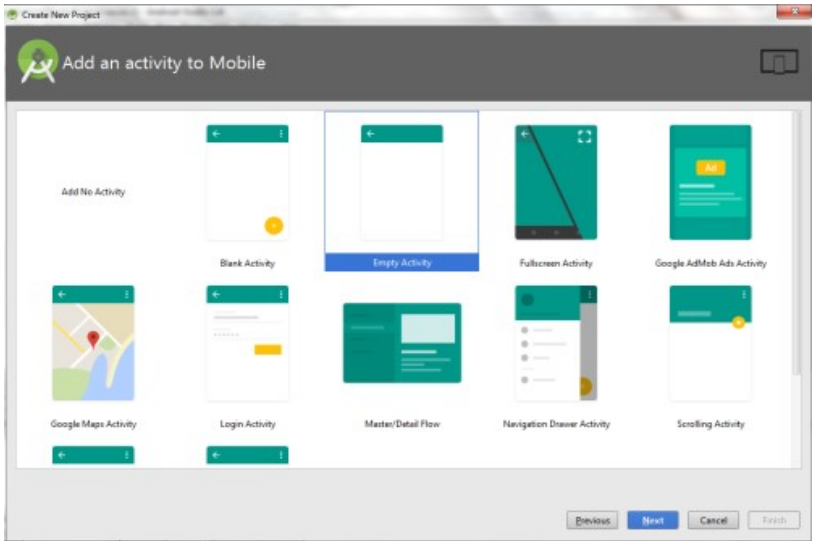
- Then type the Application name as "**ex.no.2"** and click **Next.**



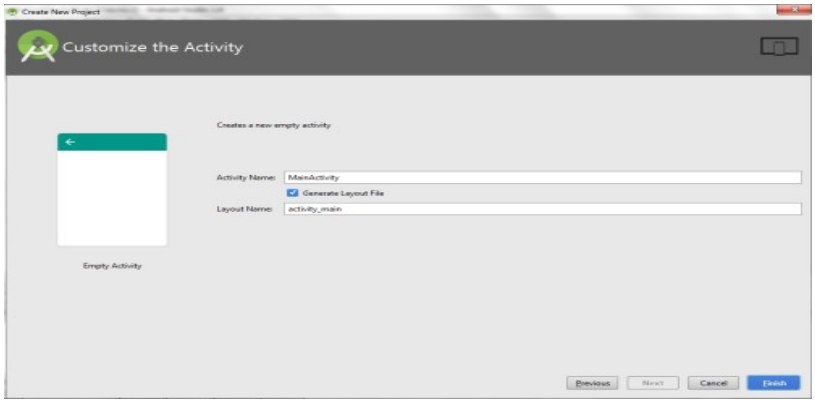
- Then select the **Minimum SDK** as shown below and click **Next**.



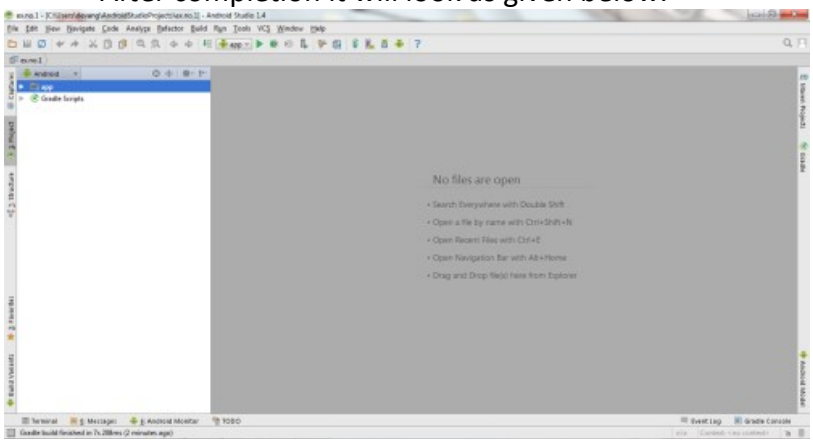
- Then select the **Empty Activity** and click **Next.**



- Finally click F**inish**.

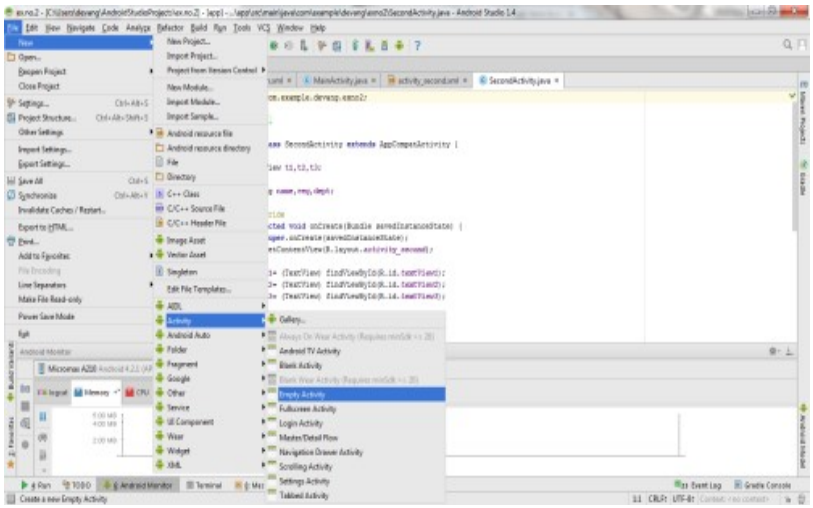


- It will take some time to build and load the project.
- After completion it will look as given below.

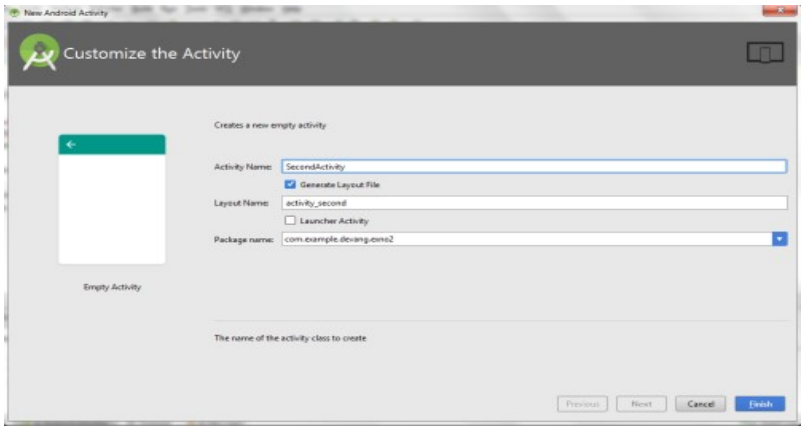


Creating Second Activity for the Android Application:

- Click on **File -> New -> Activity -> Empty Activity.**



- Type the Activity Name as **SecondActivity** and click **Finish** button.

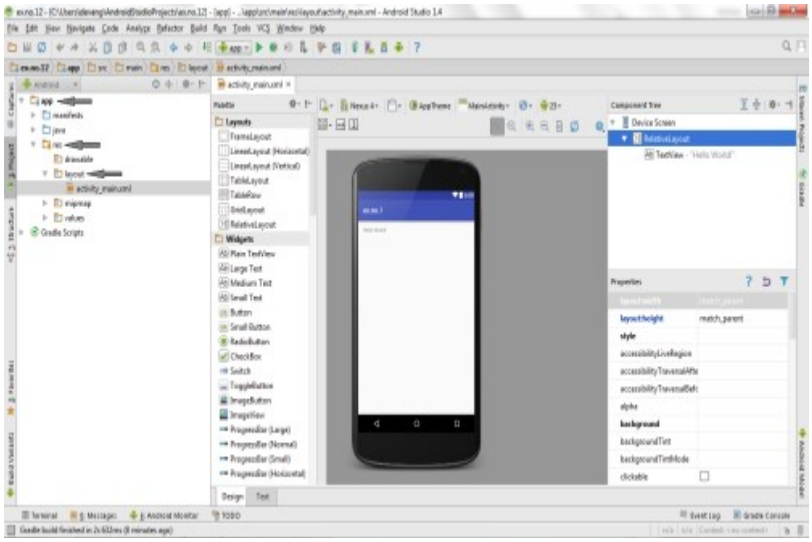


- Thus Second Activity For the application is created.

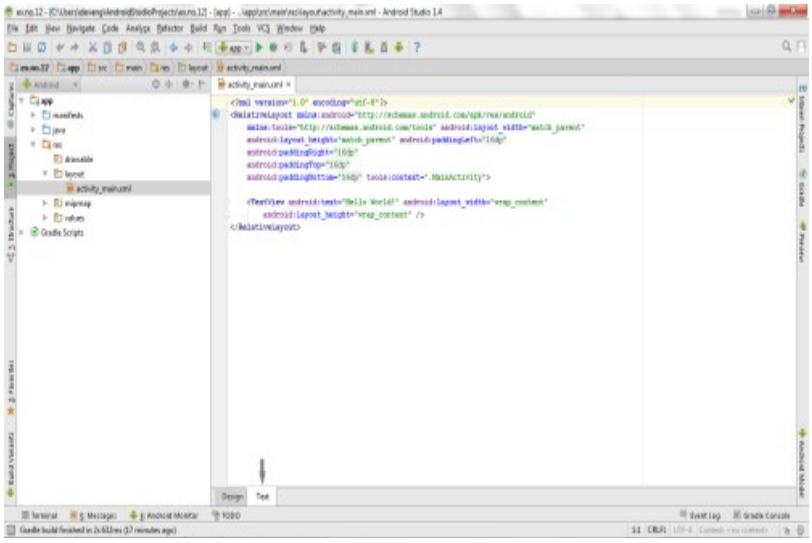
Designing layout for the Android Application:

Designing Layout for Main Activity:

- Click on **app -> res -> layout -> activity_main.xml.**



- Now click on **Text** as shown below.



- Then delete the code which is there and type the code as given below.

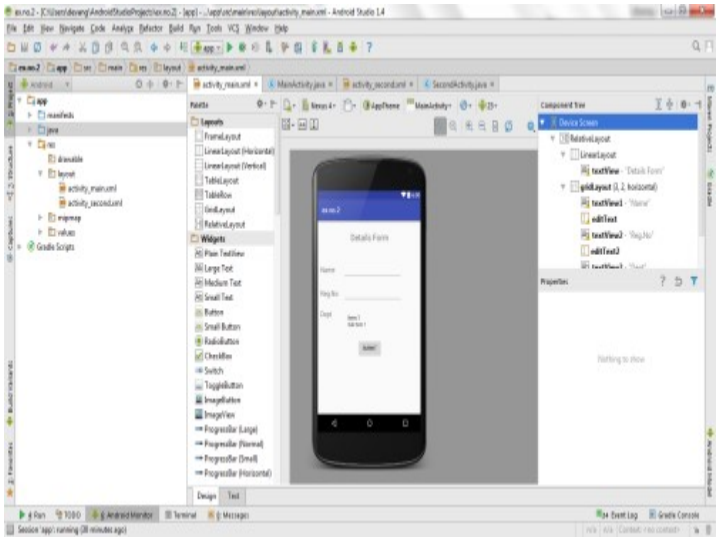
Code for Activity_main.xml:

```
<?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"
   tools:context=".MainActivity">
    <LinearLayout
      android:layout_width="match_parent"
      android:layout_height="100dp">
      <TextView
         android:id="@+id/textView"
         android:layout_width="match_parent"
         android:layout_height="wrap_content"
         android:layout_margin="30dp"
         android:text="Details Form"
         android:textSize="25sp"
         android:gravity="center"/>
   </LinearLayout>
    <GridLayout
      android:id="@+id/gridLayout"
      android:layout_width="match_parent"
      android:layout_height="match_parent"
      android:layout_marginTop="100dp"
      android:layout_marginBottom="200dp"
      android:columnCount="2"
      android:rowCount="3">
      <TextView
         android:id="@+id/textView1"
         android:layout_width="wrap_content"
         android:layout_height="wrap_content"
         android:layout_margin="10dp"
         android:layout_row="0"
         android:layout_column="0"
         android:text="Name"
         android:textSize="20sp"
         android:gravity="center"/>
       <EditText
         android:id="@+id/editText"
         android:layout_width="wrap_content"
         android:layout_height="wrap_content"
         android:layout_margin="10dp"
         android:layout_row="0"
```

```
        android:layout_column="1"
        android:ems="10"/>
    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:layout_row="1"
        android:layout_column="0"
        android:text="Reg.No"
        android:textSize="20sp"
        android:gravity="center"/>
    <EditText
        android:id="@+id/editText2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:layout_row="1"
        android:layout_column="1"
        android:inputType="number"
        android:ems="10"/>
    <TextView
        android:id="@+id/textView3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:layout_row="2"
        android:layout_column="0"
        android:text="Dept"
        android:textSize="20sp"
        android:gravity="center"/>
    <Spinner
        android:id="@+id/spinner"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:layout_row="2"
        android:layout_column="1"
        android:spinnerMode="dropdown"/>
</GridLayout>
<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
```

```
    android:layout_alignParentBottom="true"
    android:layout_centerInParent="true"
    android:layout_marginBottom="150dp"
    android:text="Submit"/>
</RelativeLayout>
```

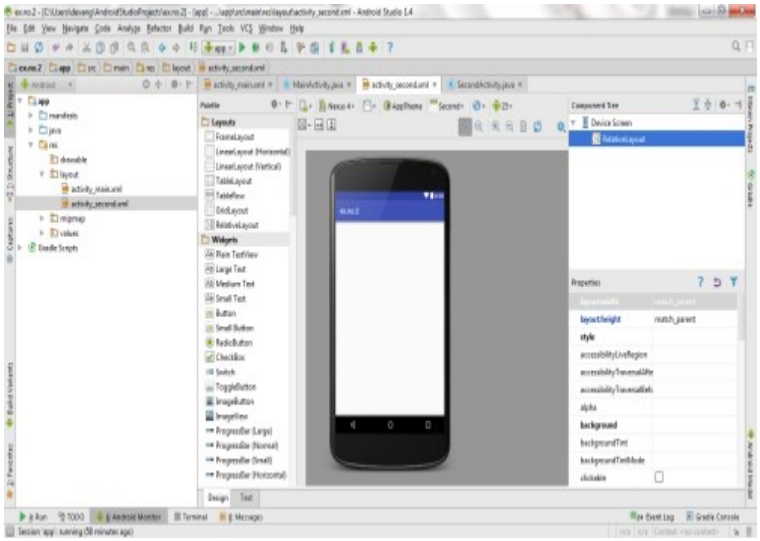
- Now click on Design and your activity will look as given below.



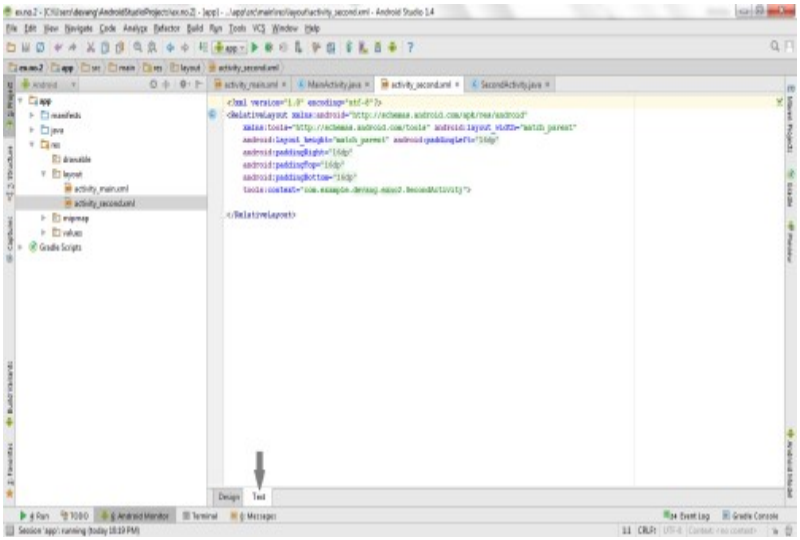
- So now the designing part of Main Activity is completed.

Designing Layout for Second Activity:

- Click on **app -> res -> layout -> activity_second.xml.**



- Now click on **Text** as shown below.



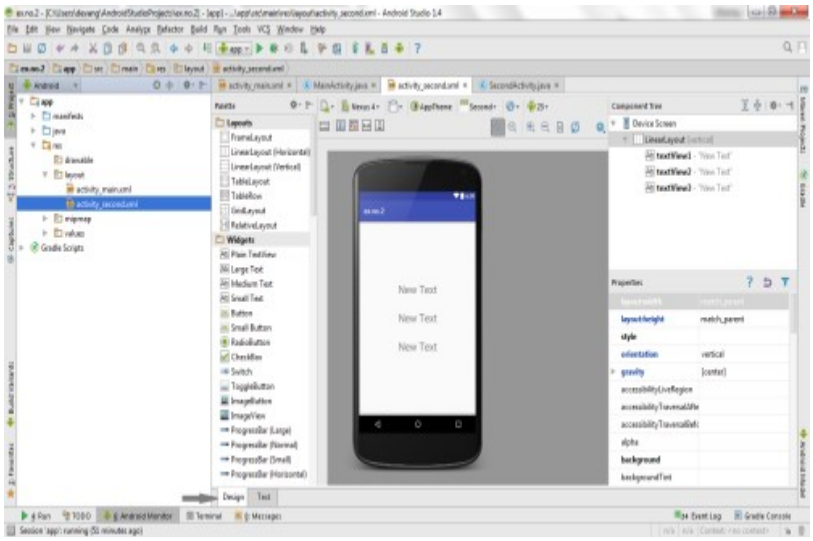
- Then delete the code which is there and type the code as given below.

Code for Activity_second.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   tools:context="com.example.devang.exno2.SecondActivity"
   android:orientation="vertical"
   android:gravity="center">
    <TextView
      android:id="@+id/textView1"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_margin="20dp"
      android:text="New Text"
      android:textSize="30sp"/>
    <TextView
      android:id="@+id/textView2"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_margin="20dp"
      android:text="New Text"
      android:textSize="30sp"/>
    <TextView
      android:id="@+id/textView3"
      android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
    android:layout_margin="20dp"
    android:text="New Text"
    android:textSize="30sp"/>
</LinearLayout>
```

- Now click on Design and your activity will look as given below.

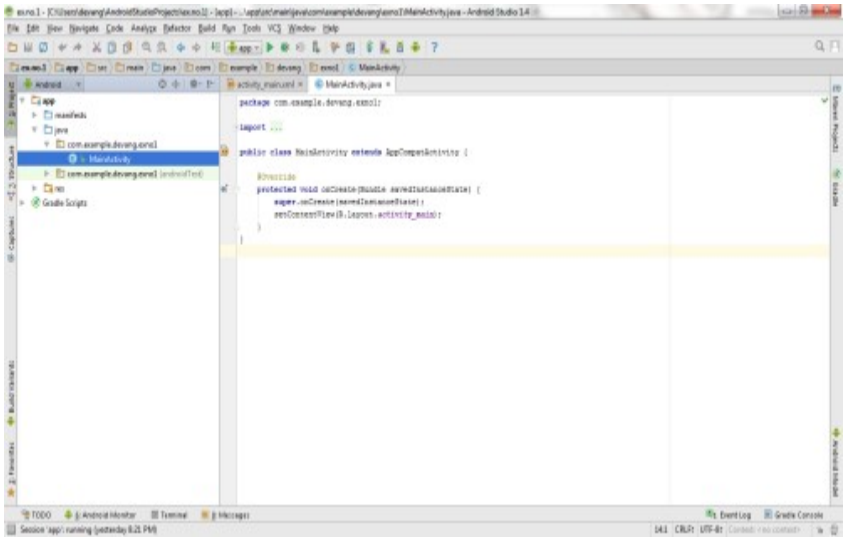


- So now the designing part of Second Activity is also completed.

Java Coding for the Android Application:

Java Coidng for Main Activity:

- Click on **app -> java -> com.example.exno2 -> MainActivity.**



- Then delete the code which is there and type the code as given below.

Code for MainActivity.java:

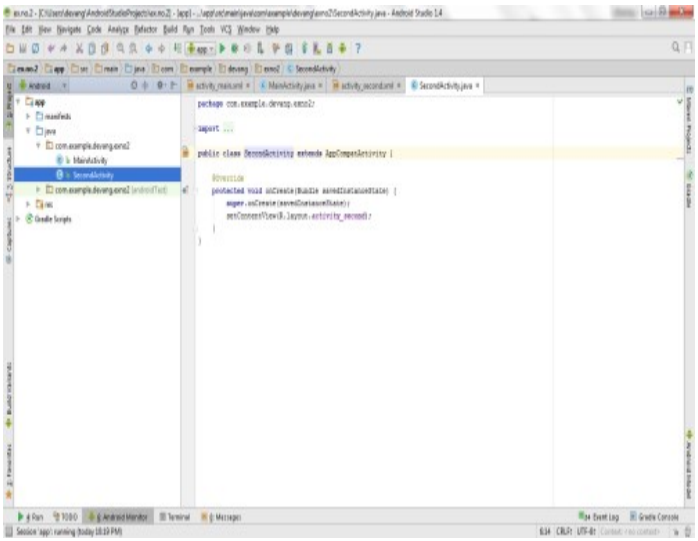
```
package com.example.exno2;
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Spinner;
public class MainActivity extends AppCompatActivity {
   //Defining the Views
   EditText e1,e2;
   Button bt;
   Spinner s;
    //Data for populating in Spinner
   String [] dept_array={"CSE","ECE","IT","Mech","Civil"};
    String name,reg,dept;
    @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
       //Referring the Views
      e1= (EditText) findViewById(R.id.editText);
      e2= (EditText) findViewById(R.id.editText2);
       bt= (Button) findViewById(R.id.button);
       s= (Spinner) findViewById(R.id.spinner);
       //Creating Adapter for Spinner for adapting the data from array to Spinner
      ArrayAdapter adapter= new
ArrayAdapter(MainActivity.this,android.R.layout.simple_spinner_item,dept_array);
      s.setAdapter(adapter);
       //Creating Listener for Button
      bt.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View v) {
             //Getting the Values from Views(Edittext & Spinner)
            name=e1.getText().toString();
            reg=e2.getText().toString();
            dept=s.getSelectedItem().toString();
             //Intent For Navigating to Second Activity
            Intent i = new Intent(MainActivity.this,SecondActivity.class);
             //For Passing the Values to Second Activity
            i.putExtra("name_key", name);
```

```
    i.putExtra("reg_key",reg);
    i.putExtra("dept_key", dept);
     startActivity(i);
}    });    } }
```

- So now the Coding part of Main Activity is completed.

Java Coding for Second Activity:

- Click on **app -> java -> com.example.exno2 -> SecondActivity.**



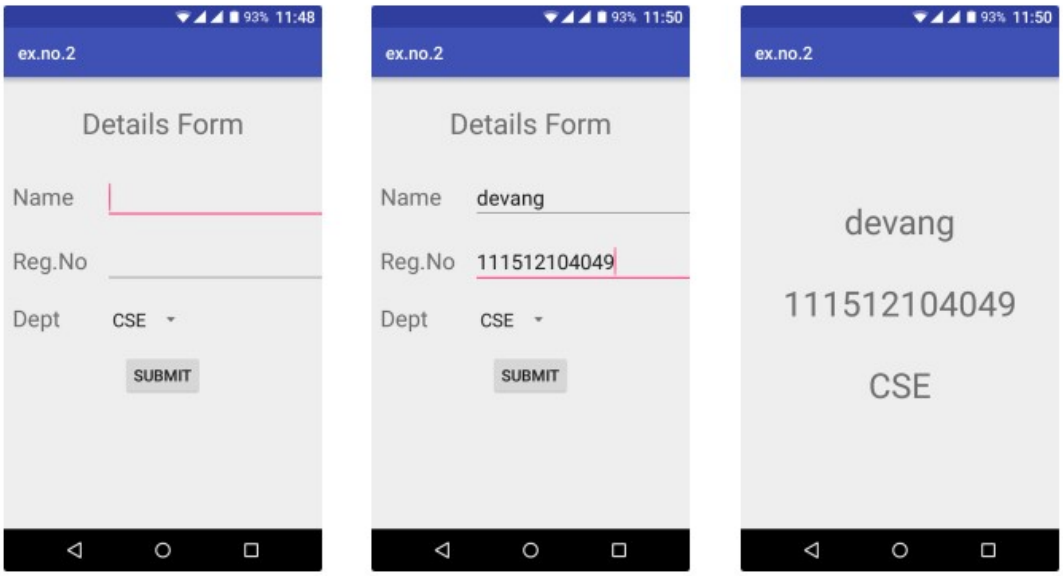
- Then delete the code which is there and type the code as given below.

Code for SecondActivity.java:

```
package com.example.exno2;
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.TextView;
public class SecondActivity extends AppCompatActivity {
    TextView t1,t2,t3;
    String name,reg,dept;
    @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_second);
       t1= (TextView) findViewById(R.id.textView1);
      t2= (TextView) findViewById(R.id.textView2);
```

```
        t3= (TextView) findViewById(R.id.textView3);
         //Getting the Intent
        Intent i = getIntent();
         //Getting the Values from First Activity using the Intent received
        name=i.getStringExtra("name_key");
        reg=i.getStringExtra("reg_key");
        dept=i.getStringExtra("dept_key");
         //Setting the Values to Intent
        t1.setText(name);
        t2.setText(reg);
        t3.setText(dept);
    }
}
```

Output:



Result:

Thus a Simple Android Application that uses Layout Managers and Event Listeners is developed and executed successfully.

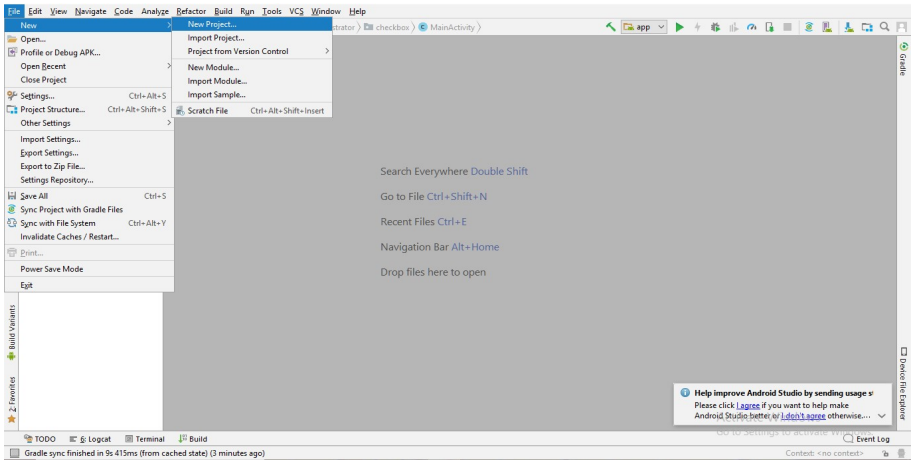
EXP.NO:3	DEVELOP AN NATIVE CALCULATOR APPLICATION
DATE: 20-07-18	

Aim:

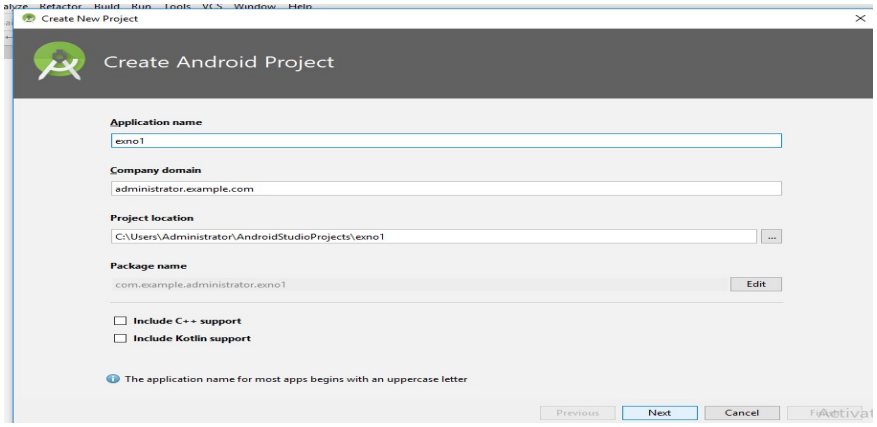
To develop a Simple Android Application for Native Calculator.

Algorithm:

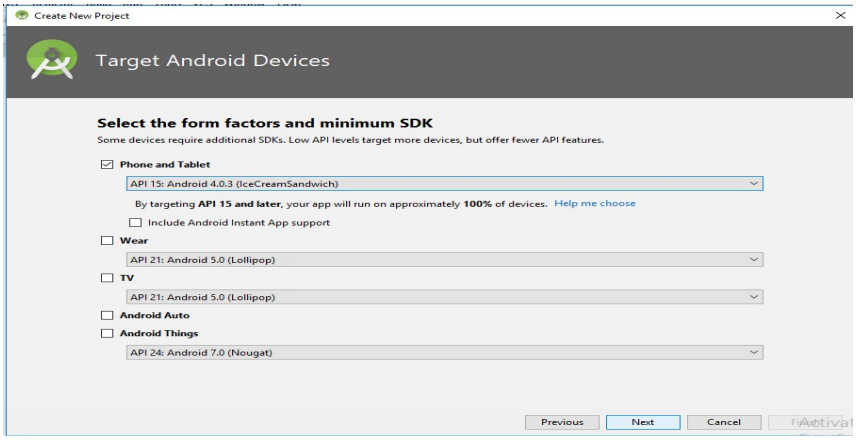
- Start the program.
- Creating a New project:
  - Open Android Studio and then click on **File -> New -> New project.**



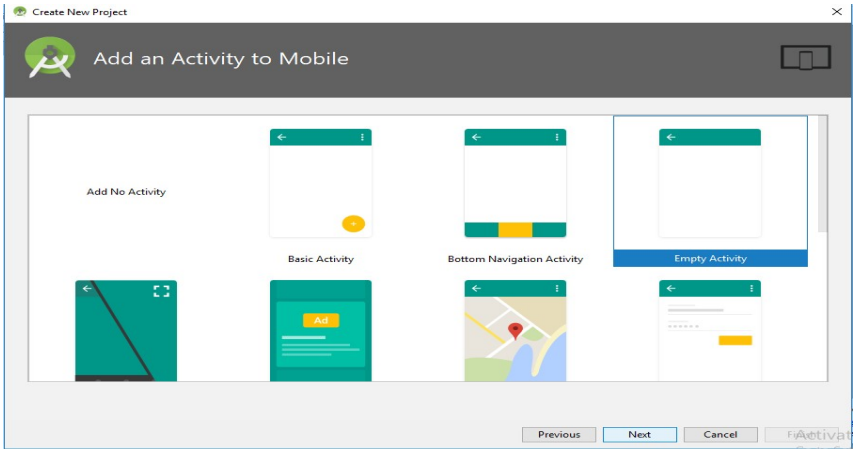
  - Then type the Application name and click **Next.**



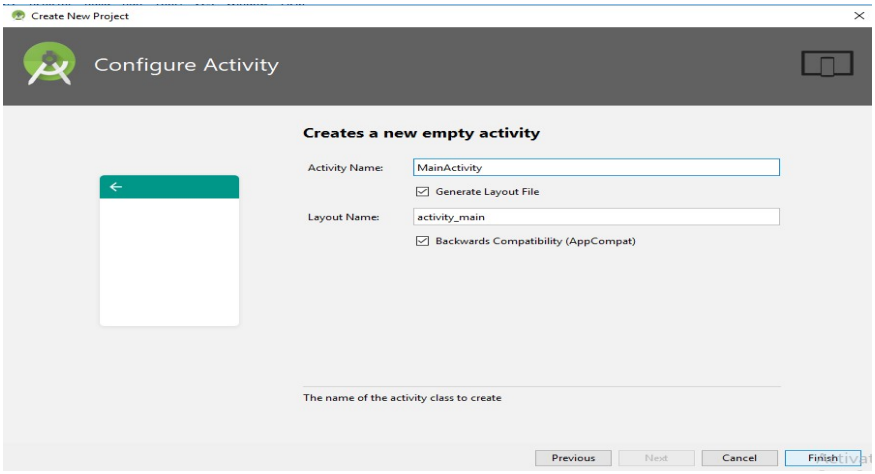
  - Then select the **Minimum SDK** as shown below and click **Next**.



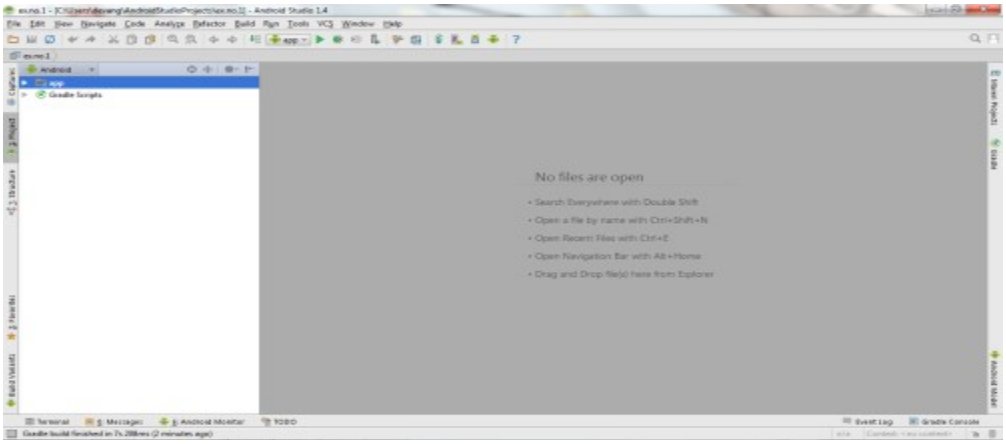
- Then select the **Empty Activity** and click **Next.**



- Finally click F**inish**.

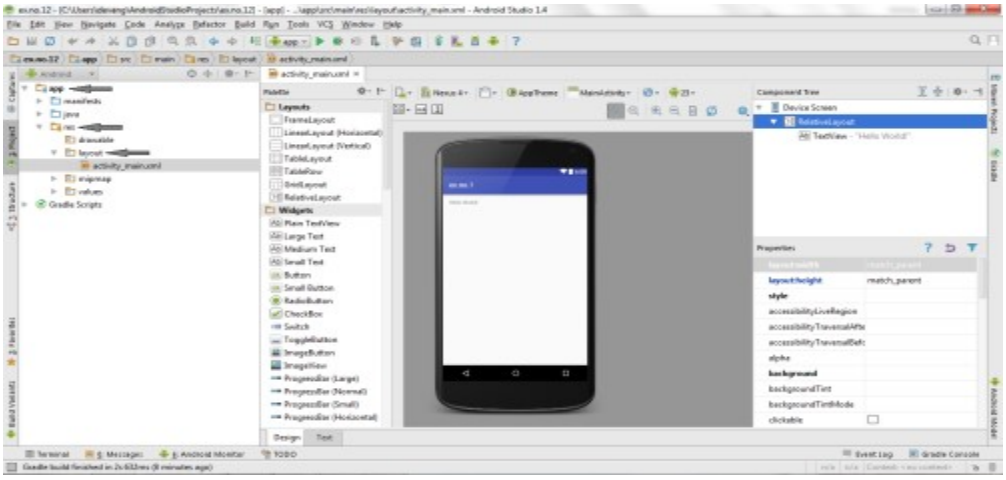


- It will take some time to build and load the project.
- After completion it will look as given below.

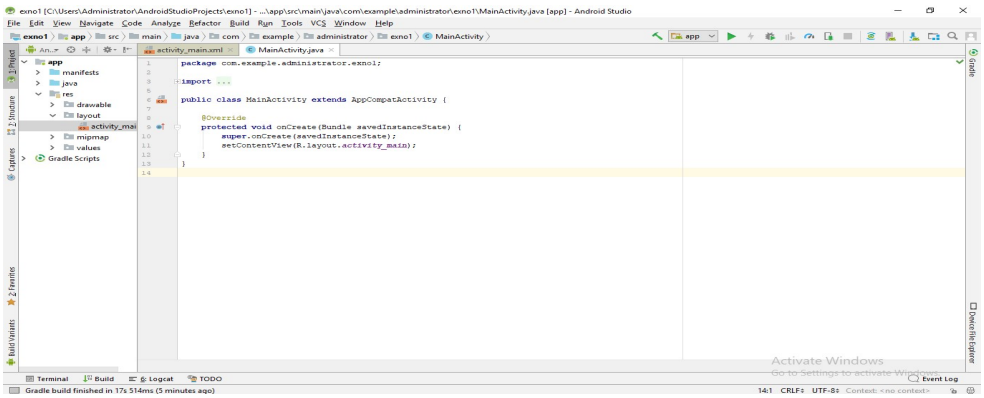


- Designing layout for the Android Application.

  Click on **app -> res -> layout -> activity_main.xml**



  - Now click on **Text** as shown below.



  - Then delete the code which is there and type the code as given below

Code for Activity_main.xml:

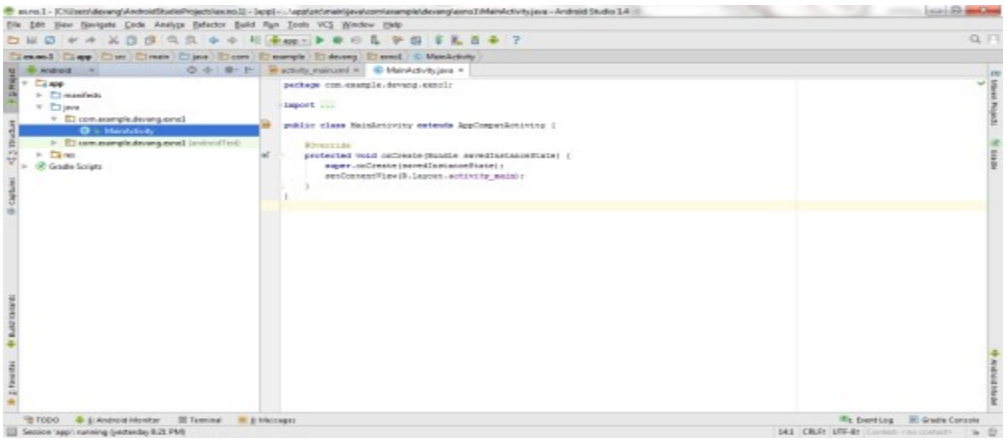
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
   xmlns:android="http://schemas.android.com/apk/res/android"
   android:orientation="vertical"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:layout_margin="20dp">
    <LinearLayout
      android:id="@+id/linearLayout1"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:layout_margin="20dp">
       <EditText
         android:id="@+id/editText1"
         android:layout_width="match_parent"
         android:layout_height="wrap_content"
         android:layout_weight="1"
         android:inputType="numberDecimal"
         android:textSize="20sp" />
       <EditText
         android:id="@+id/editText2"
         android:layout_width="match_parent"
         android:layout_height="wrap_content"
         android:layout_weight="1"
         android:inputType="numberDecimal"
         android:textSize="20sp" />
    </LinearLayout>
    <LinearLayout
      android:id="@+id/linearLayout2"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:layout_margin="20dp">
       <Button
         android:id="@+id/Add"
         android:layout_width="match_parent"
         android:layout_height="wrap_content"
         android:layout_weight="1"
         android:text="+"
         android:textSize="30sp"/>
       <Button
         android:id="@+id/Sub"
         android:layout_width="match_parent"
```

```
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="-"
        android:textSize="30sp"/>
    <Button
        android:id="@+id/Mul"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="*"
        android:textSize="30sp"/>
    <Button
        android:id="@+id/Div"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="/"
        android:textSize="30sp"/>
  </LinearLayout>
  <TextView
    android:id="@+id/textView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="50dp"
    android:text="Answer is"
    android:textSize="30sp"
    android:gravity="center"/>
</LinearLayout>
```

- So now the designing part is completed

Java Coding for the Android Application:

Click on **app -> java -> com.example.exno1 -> MainActivity.**



- Then delete the code which is there and type the code as given below.

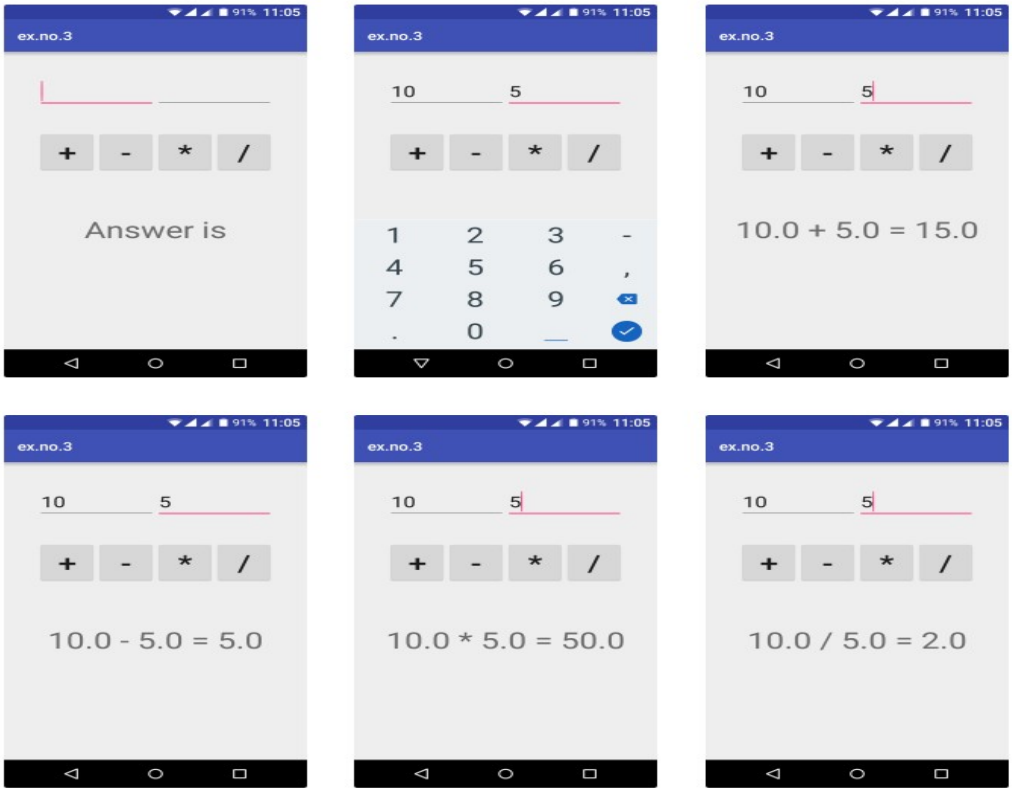
Code for MainActivity.java:

```
package com. exno3;
 import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.text.TextUtils;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
public class MainActivity extends AppCompatActivity implements OnClickListener
{
   //Defining the Views
   EditText Num1;
   EditText Num2;
   Button Add;
   Button Sub;
   Button Mul;
   Button Div;
   TextView Result;
    @Override
   public void onCreate(Bundle savedInstanceState)
   {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
       //Referring the Views
      Num1 = (EditText) findViewById(R.id.editText1);
      Num2 = (EditText) findViewById(R.id.editText2);
      Add = (Button) findViewById(R.id.Add);
      Sub = (Button) findViewById(R.id.Sub);
      Mul = (Button) findViewById(R.id.Mul);
      Div = (Button) findViewById(R.id.Div);
      Result = (TextView) findViewById(R.id.textView);
       // set a listener
      Add.setOnClickListener(this);
      Sub.setOnClickListener(this);
      Mul.setOnClickListener(this);
      Div.setOnClickListener(this);
   }
    @Override
   public void onClick (View v)
```

```
  {
     float num1 = 0;
     float num2 = 0;
     float result = 0;
     String oper = "";
      // check if the fields are empty
     if (TextUtils.isEmpty(Num1.getText().toString()) ||
TextUtils.isEmpty(Num2.getText().toString()))
          return;
      // read EditText and fill variables with numbers
     num1 = Float.parseFloat(Num1.getText().toString());
     num2 = Float.parseFloat(Num2.getText().toString());
      // defines the button that has been clicked and performs the corresponding operation
     // write operation into oper, we will use it later for output
     switch (v.getId())
     {
        case R.id.Add:
           oper = "+";
           result = num1 + num2;
           break;
        case R.id.Sub:
           oper = "-";
           result = num1 - num2;
           break;
        case R.id.Mul:
           oper = "*";
           result = num1 * num2;
           break;
        case R.id.Div:
           oper = "/";
           result = num1 / num2;
           break;
        default:
           break;
     }
     // form the output line
     Result.setText(num1 + " " + oper + " " + num2 + " = " + result);
  }
}
```

- So now the Coding part is also completed.
- Now run the application to see the output.

Output:



Result:

Thus a Simple Android Application for Native Calculator is developed and executed successfully.

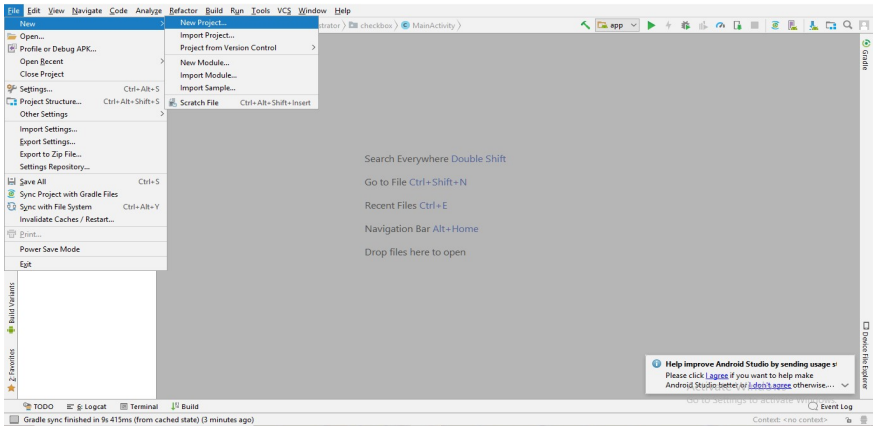
EXP.NO:4	WRITE AN APPLICATION THAT DRAWS BASIC GRAPHICAL PRIMITIVES ON THE SCREEN
DATE: 27-07-18	

Aim :

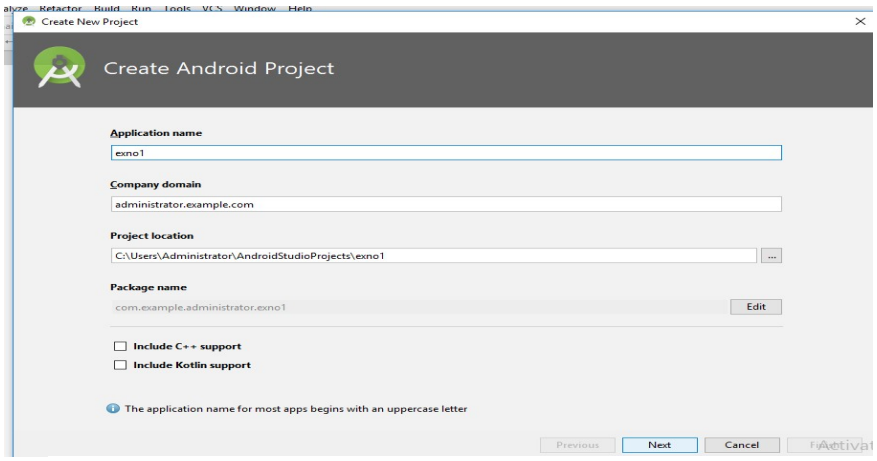
To develop a Simple Android Application that draws basic Graphical Primitives on the screen.

Algorithm:

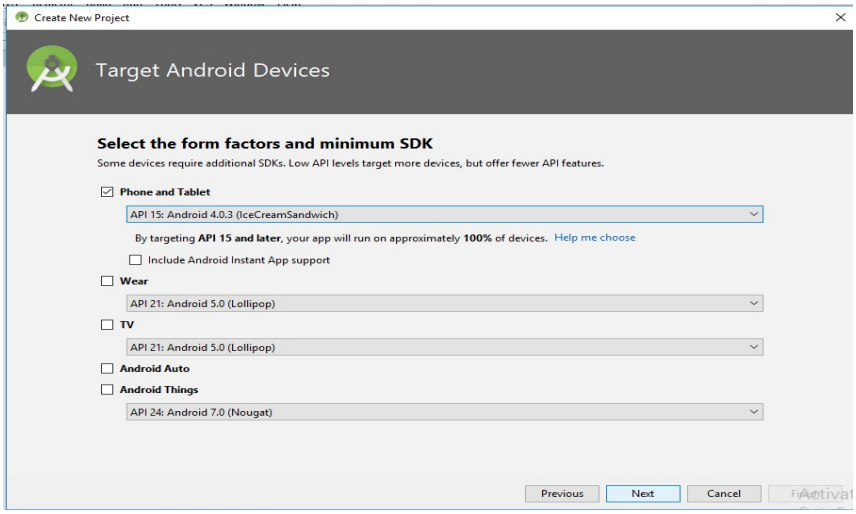
- Start the program.
- Creating a New project:
    - Open Android Studio and then click on **File -> New -> New project.**



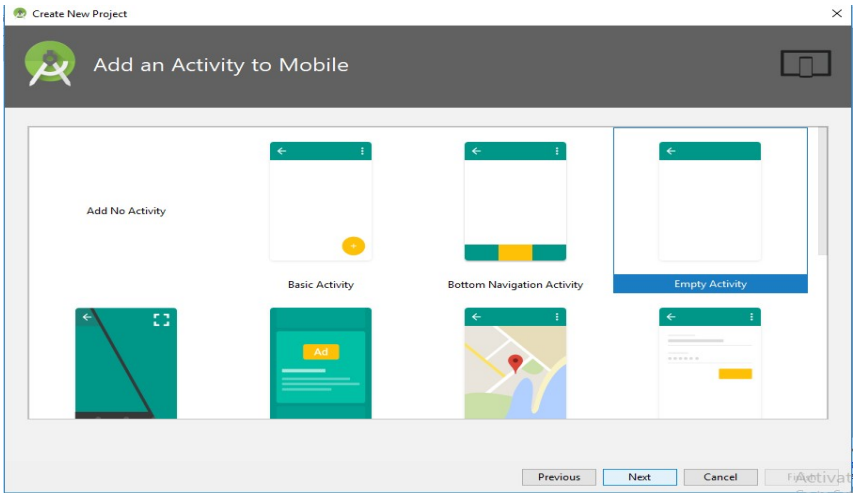
- Then type the Application name and click **Next.**



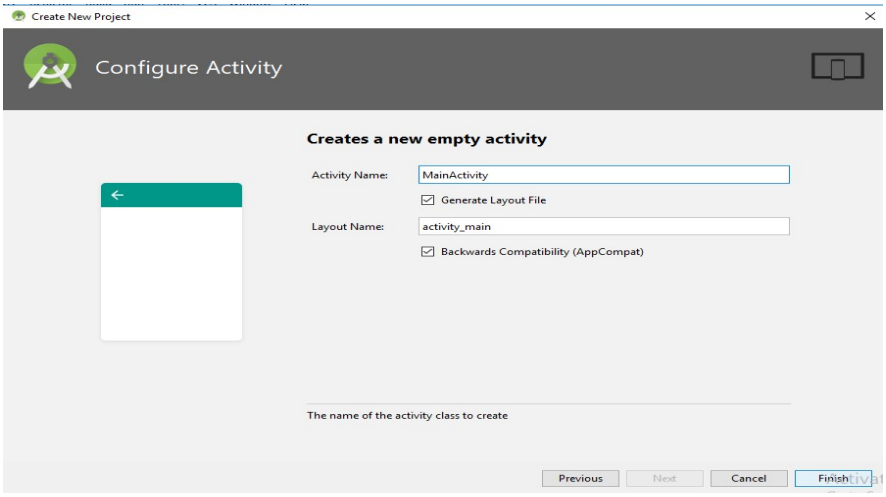
- Then select the **Minimum SDK** as shown below and click **Next**.



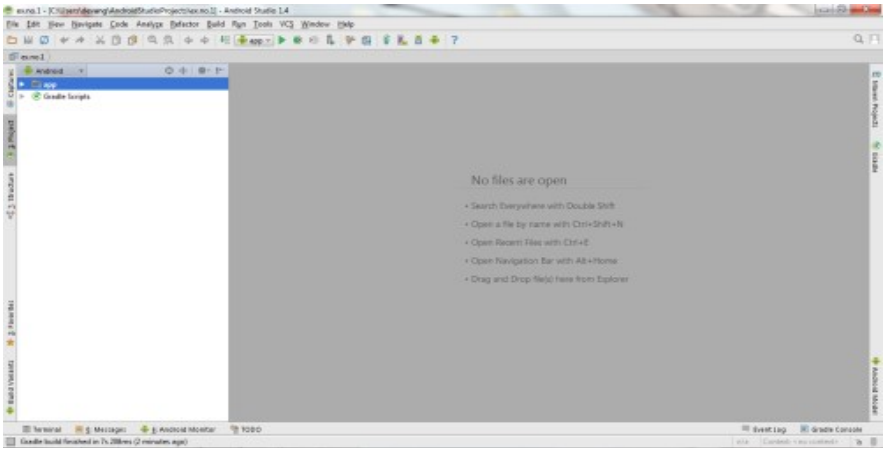
- Then select the **Empty Activity** and click **Next.**



- Finally click F**inish**.

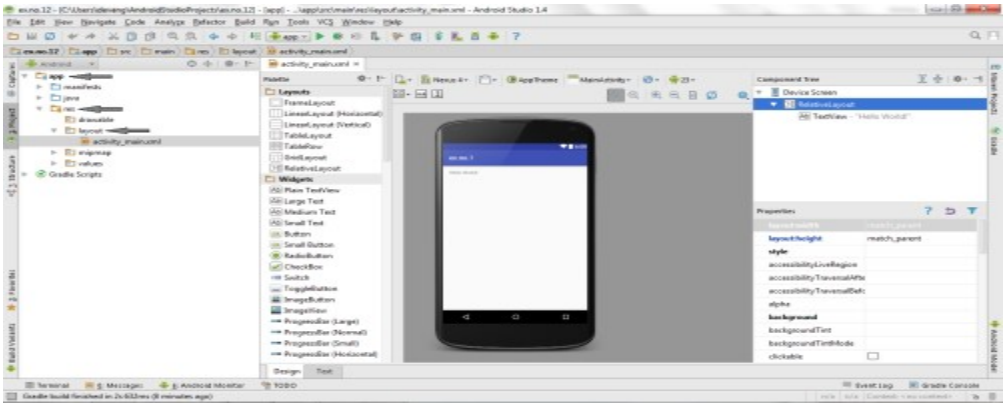


- It will take some time to build and load the project.
- After completion it will look as given below.

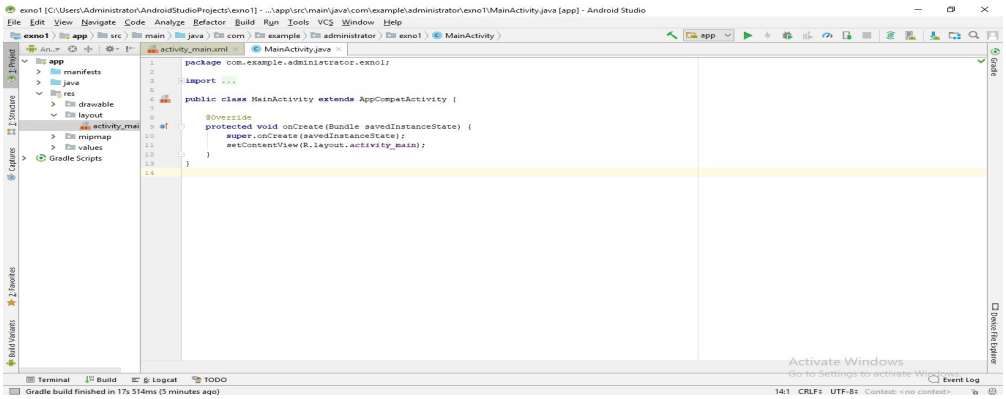


- Designing layout for the Android Application.

  Click on **app -> res -> layout -> activity_main.xml**



- Now click on **Text** as shown below.



- Then delete the code which is there and type the code as given below

Code for Activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="match_parent">
    <ImageView
      android:layout_width="match_parent"
      android:layout_height="match_parent"
      android:id="@+id/imageView" />
</RelativeLayout>
```

- So now the designing part is completed

Java Coding for the Android Application:

Click on **app -> java -> com.example.exno1 -> MainActivity.**

- Then delete the code which is there and type the code as given below.

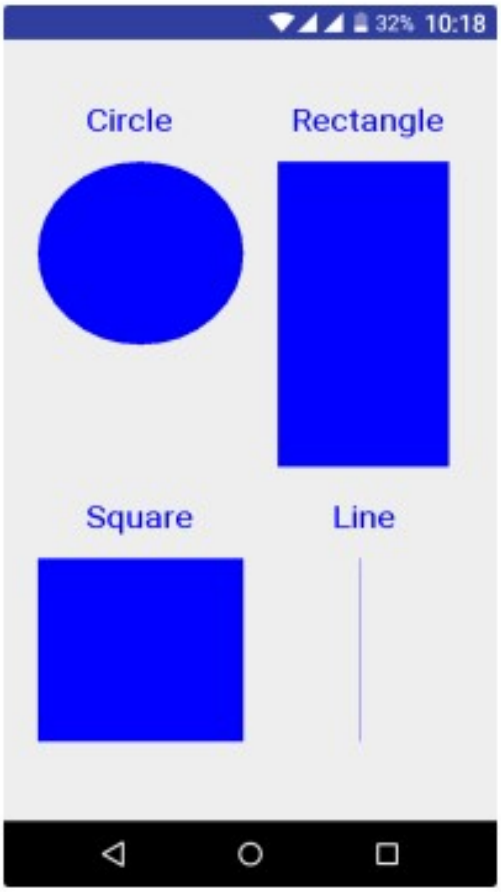
Code for MainActivity.java:

```
package com.example.exno4;

import android.app.Activity;
import android.graphics.Bitmap;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.graphics.drawable.BitmapDrawable;
import android.os.Bundle;
import android.widget.ImageView;
```

```
public class MainActivity extends Activity
{
   @Override
   public void onCreate(Bundle savedInstanceState)
   {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      //Creating a Bitmap
      Bitmap bg = Bitmap.createBitmap(720, 1280, Bitmap.Config.ARGB_8888);
      //Setting the Bitmap as background for the ImageView
      ImageView i = (ImageView) findViewById(R.id.imageView);
      i.setBackgroundDrawable(new BitmapDrawable(bg));
      //Creating the Canvas Object
      Canvas canvas = new Canvas(bg);
      //Creating the Paint Object and set its color & TextSize
      Paint paint = new Paint();
      paint.setColor(Color.BLUE);
      paint.setTextSize(50);
      //To draw a Rectangle
      canvas.drawText("Rectangle", 420, 150, paint);
      canvas.drawRect(400, 200, 650, 700, paint);
      //To draw a Circle
      canvas.drawText("Circle", 120, 150, paint);
      canvas.drawCircle(200, 350, 150, paint);
      //To draw a Square
      canvas.drawText("Square", 120, 800, paint);
      canvas.drawRect(50, 850, 350, 1150, paint);
      //To draw a Line
      canvas.drawText("Line", 480, 800, paint);
      canvas.drawLine(520, 850, 520, 1150, paint);
   }
}
```

- So now the Coding part is also completed.
- Now run the application to see the output.
- Stop the program.

Output:



Result:

Thus a Simple Android Application that draws basic Graphical Primitives on the screen is developed and executed successfully.

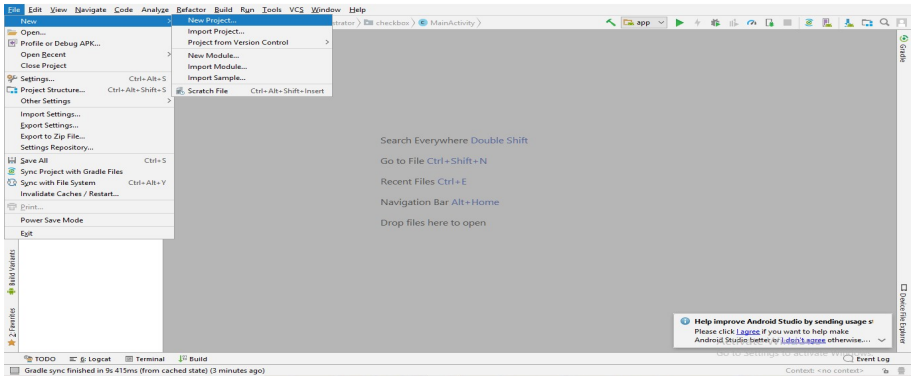
EXP.NO:5	DEVELOP AN APPLICATION THAT MAKES USE OF DATABASE.
DATE: 10-08-18	

Aim :

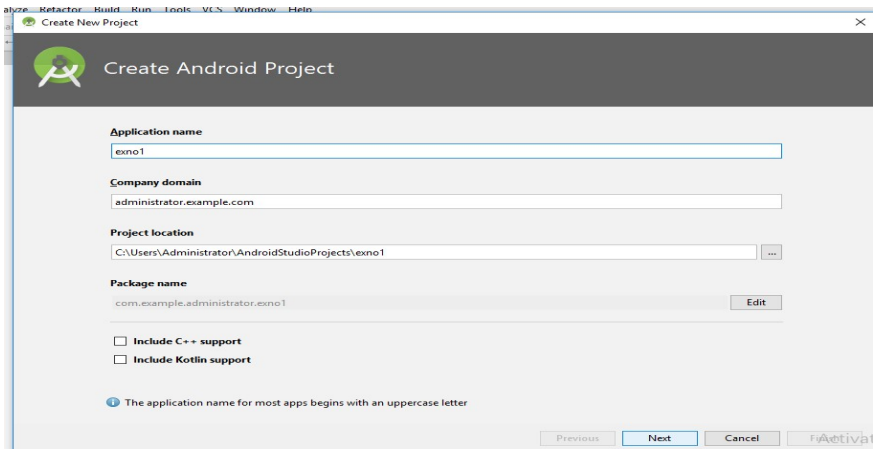
To develop a Simple Android Application that makes use of Database.

Algorithm :

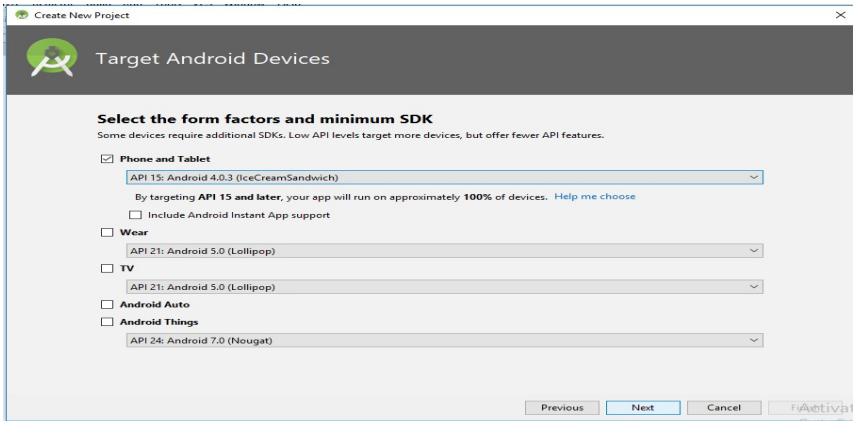
- Start the program.
- Creating a New project:
    - Open Android Studio and then click on **File -> New -> New project.**



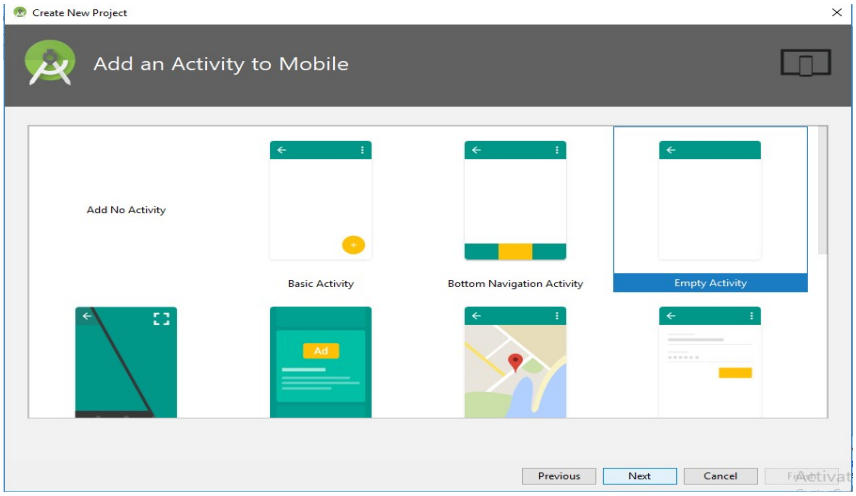
- Then type the Application name and click **Next.**



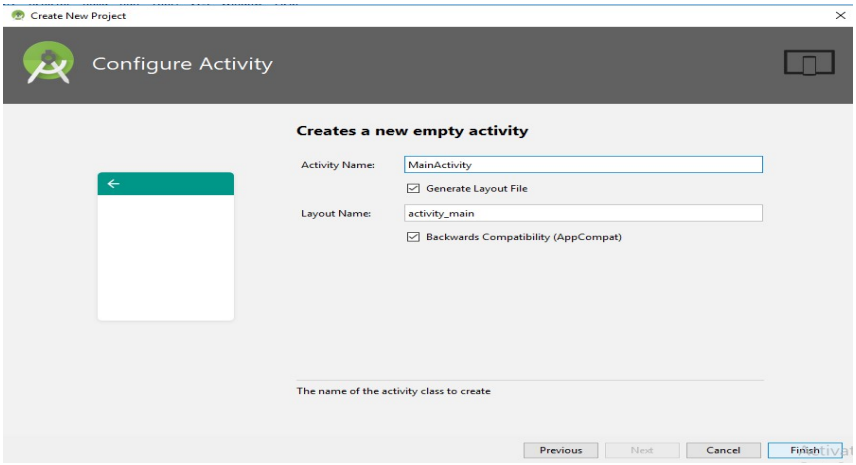
- Then select the **Minimum SDK** as shown below and click **Next**.



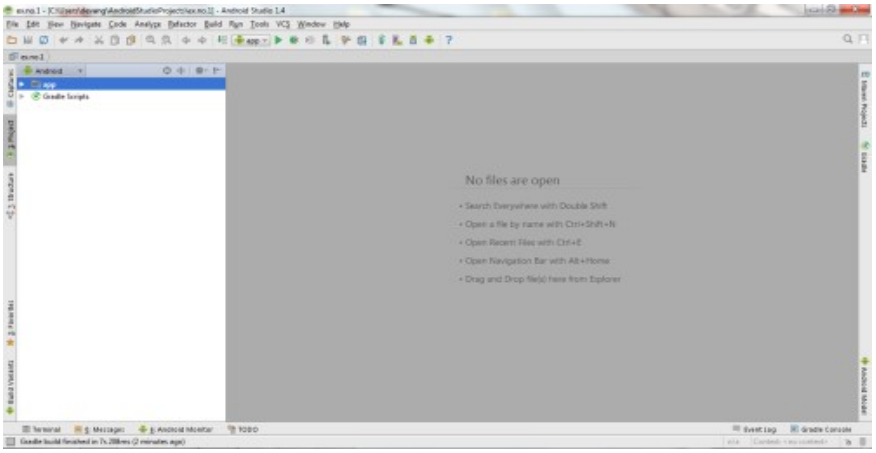
- Then select the **Empty Activity** and click **Next.**



- Finally click F**inish**.

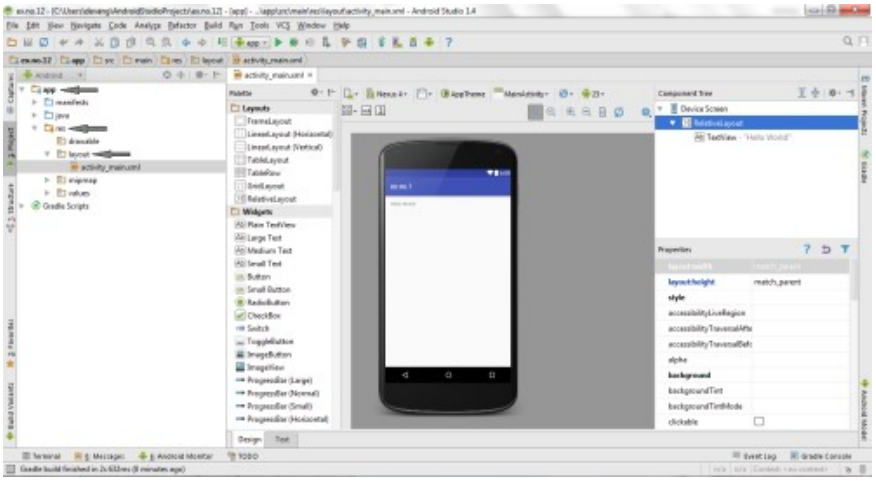


- It will take some time to build and load the project.
- After completion it will look as given below.

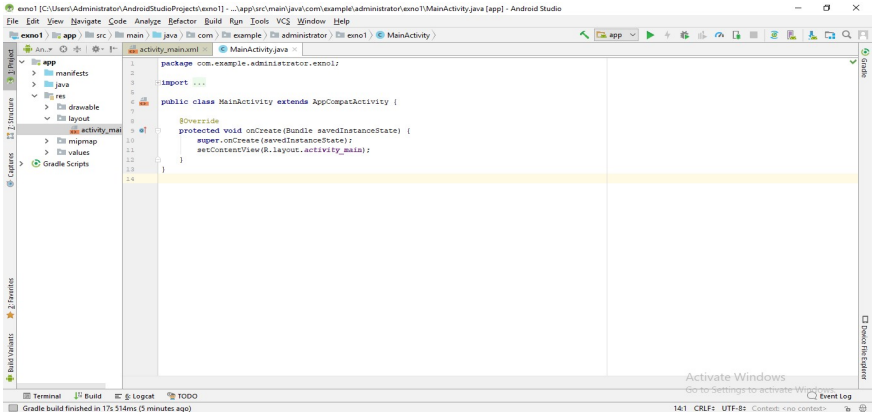


- Designing layout for the Android Application.

  Click on **app -> res -> layout -> activity_main.xml**



- Now click on **Text** as shown below.



- Then delete the code which is there and type the code as given below

Code for Activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<AbsoluteLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="match_parent">
   <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_x="50dp"
      android:layout_y="20dp"
      android:text="Student Details"
      android:textSize="30sp" />
   <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_x="20dp"
      android:layout_y="110dp"
      android:text="Enter Rollno:"
      android:textSize="20sp" />
   <EditText
      android:id="@+id/Rollno"
      android:layout_width="150dp"
      android:layout_height="wrap_content"
      android:layout_x="175dp"
      android:layout_y="100dp"
      android:inputType="number"
      android:textSize="20sp" />
   <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_x="20dp"
      android:layout_y="160dp"
      android:text="Enter Name:"
      android:textSize="20sp" />
   <EditText
      android:id="@+id/Name"
      android:layout_width="150dp"
      android:layout_height="wrap_content"
      android:layout_x="175dp"
      android:layout_y="150dp"
      android:inputType="text"
      android:textSize="20sp" />
   <TextView
```

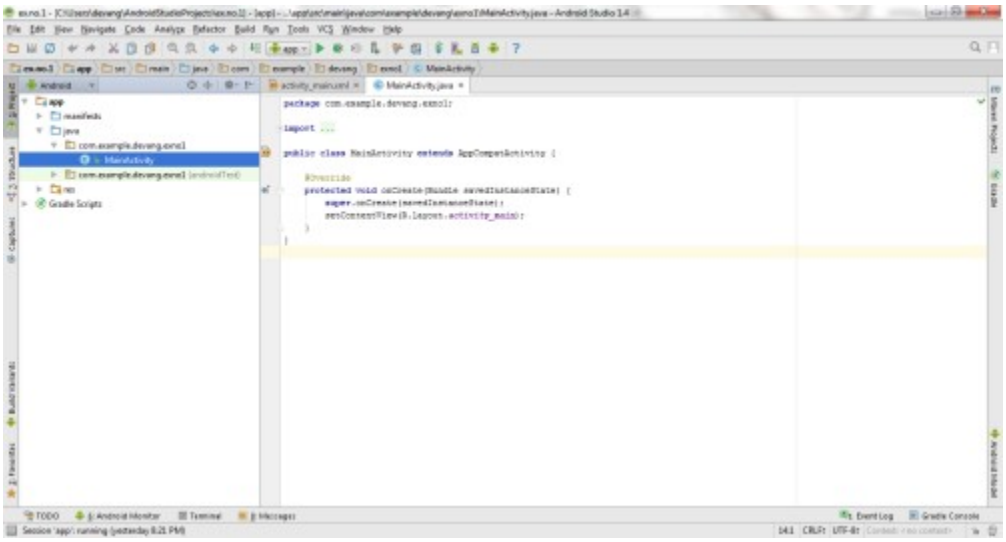
```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_x="20dp"
    android:layout_y="210dp"
    android:text="Enter Marks:"
    android:textSize="20sp" />
<EditText
    android:id="@+id/Marks"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_x="175dp"
    android:layout_y="200dp"
    android:inputType="number"
    android:textSize="20sp" />
<Button
    android:id="@+id/Insert"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_x="25dp"
    android:layout_y="300dp"
    android:text="Insert"
    android:textSize="30dp" />
<Button
    android:id="@+id/Delete"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_x="200dp"
    android:layout_y="300dp"
    android:text="Delete"
    android:textSize="30dp" />
<Button
    android:id="@+id/Update"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_x="25dp"
    android:layout_y="400dp"
    android:text="Update"
    android:textSize="30dp" />
<Button
    android:id="@+id/View"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_x="200dp"
    android:layout_y="400dp"
```

```
    android:text="View"
    android:textSize="30dp" />
  <Button
    android:id="@+id/ViewAll"
    android:layout_width="200dp"
    android:layout_height="wrap_content"
    android:layout_x="100dp"
    android:layout_y="500dp"
    android:text="View All"
    android:textSize="30dp" />
</AbsoluteLayout>
```

- So now the designing part is completed

Java Coding for the Android Application:

Click on **app -> java -> com.example.exno1 -> MainActivity.**



- Then delete the code which is there and type the code as given below.

Code for MainActivity.java:

```
package com.example.exno5;

import android.app.Activity;
import android.app.AlertDialog.Builder;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
```

```
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;

public class MainActivity extends Activity implements OnClickListener
{
   EditText Rollno,Name,Marks;
   Button Insert,Delete,Update,View,ViewAll;
   SQLiteDatabase db;
   /** Called when the activity is first created. */
   @Override
   public void onCreate(Bundle savedInstanceState)
   {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
       Rollno=(EditText)findViewById(R.id.Rollno);
      Name=(EditText)findViewById(R.id.Name);
      Marks=(EditText)findViewById(R.id.Marks);
      Insert=(Button)findViewById(R.id.Insert);
      Delete=(Button)findViewById(R.id.Delete);
      Update=(Button)findViewById(R.id.Update);
      View=(Button)findViewById(R.id.View);
      ViewAll=(Button)findViewById(R.id.ViewAll);
       Insert.setOnClickListener(this);
      Delete.setOnClickListener(this);
      Update.setOnClickListener(this);
      View.setOnClickListener(this);
      ViewAll.setOnClickListener(this);
       // Creating database and table
      db=openOrCreateDatabase("StudentDB", Context.MODE_PRIVATE, null);
      db.execSQL("CREATE TABLE IF NOT EXISTS student(rollno VARCHAR,name VARCHAR,marks
VARCHAR);");
   }
   public void onClick(View view)
   {
      // Inserting a record to the Student table
      if(view==Insert)
      {
         // Checking for empty fields
         if(Rollno.getText().toString().trim().length()==0||
               Name.getText().toString().trim().length()==0||
               Marks.getText().toString().trim().length()==0)
```

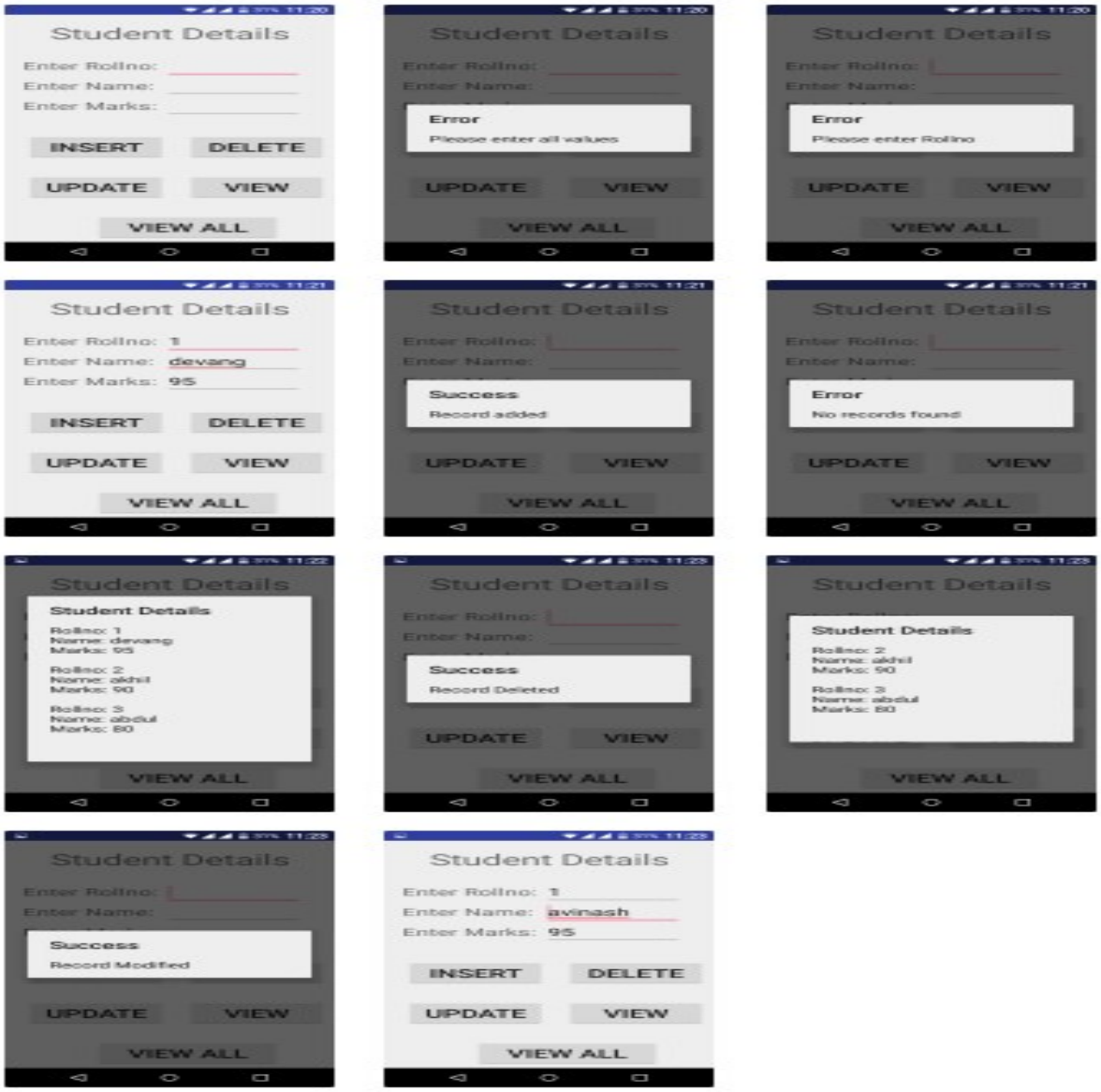
```
        {
           showMessage("Error", "Please enter all values");
           return;
        }
        db.execSQL("INSERT INTO student VALUES('"+Rollno.getText()+"','"+Name.getText()+
              "','"+Marks.getText()+"');");
        showMessage("Success", "Record added");
        clearText();
     }
     // Deleting a record from the Student table
     if(view==Delete)
     {
        // Checking for empty roll number
        if(Rollno.getText().toString().trim().length()==0)
        {
           showMessage("Error", "Please enter Rollno");
           return;
        }
        Cursor c=db.rawQuery("SELECT * FROM student WHERE rollno='"+Rollno.getText()+"'",
null);
        if(c.moveToFirst())
        {
           db.execSQL("DELETE FROM student WHERE rollno='"+Rollno.getText()+"'");
           showMessage("Success", "Record Deleted");
        }
        else
        {
           showMessage("Error", "Invalid Rollno");
        }
        clearText();
     }
     // Updating a record in the Student table
     if(view==Update)
     {
        // Checking for empty roll number
        if(Rollno.getText().toString().trim().length()==0)
        {
           showMessage("Error", "Please enter Rollno");
           return;
        }
        Cursor c=db.rawQuery("SELECT * FROM student WHERE rollno='"+Rollno.getText()+"'",
null);
        if(c.moveToFirst()) {
```

```
            db.execSQL("UPDATE student SET name='" + Name.getText() + "',marks='" +
Marks.getText() +
                  "' WHERE rollno='"+Rollno.getText()+"'");
            showMessage("Success", "Record Modified");
         }
         else {
            showMessage("Error", "Invalid Rollno");
         }
         clearText();
      }
      // Display a record from the Student table
      if(view==View)
      {
         // Checking for empty roll number
         if(Rollno.getText().toString().trim().length()==0)
         {
            showMessage("Error", "Please enter Rollno");
            return;
         }
         Cursor c=db.rawQuery("SELECT * FROM student WHERE rollno='"+Rollno.getText()+"'",
null);
         if(c.moveToFirst())
         {
            Name.setText(c.getString(1));
            Marks.setText(c.getString(2));
         }
         else
         {
            showMessage("Error", "Invalid Rollno");
            clearText();
         }
      }
      // Displaying all the records
      if(view==ViewAll)
      {
         Cursor c=db.rawQuery("SELECT * FROM student", null);
         if(c.getCount()==0)
         {
            showMessage("Error", "No records found");
            return;
         }
         StringBuffer buffer=new StringBuffer();
         while(c.moveToNext())
         {
```

```
            buffer.append("Rollno: "+c.getString(0)+"\n");
            buffer.append("Name: "+c.getString(1)+"\n");
            buffer.append("Marks: "+c.getString(2)+"\n\n");
          }
          showMessage("Student Details", buffer.toString());
        }
    }
    public void showMessage(String title,String message)
    {
       Builder builder=new Builder(this);
       builder.setCancelable(true);
       builder.setTitle(title);
       builder.setMessage(message);
       builder.show();
    }
    public void clearText()
    {
       Rollno.setText("");
       Name.setText("");
       Marks.setText("");
       Rollno.requestFocus();
    }
}
```

- So now the Coding part is also completed.
- Now run the application to see the output.
- Stop the program.

Output:



Result :

Thus a Simple Android Application that makes use of Database is developed and executed successfully.

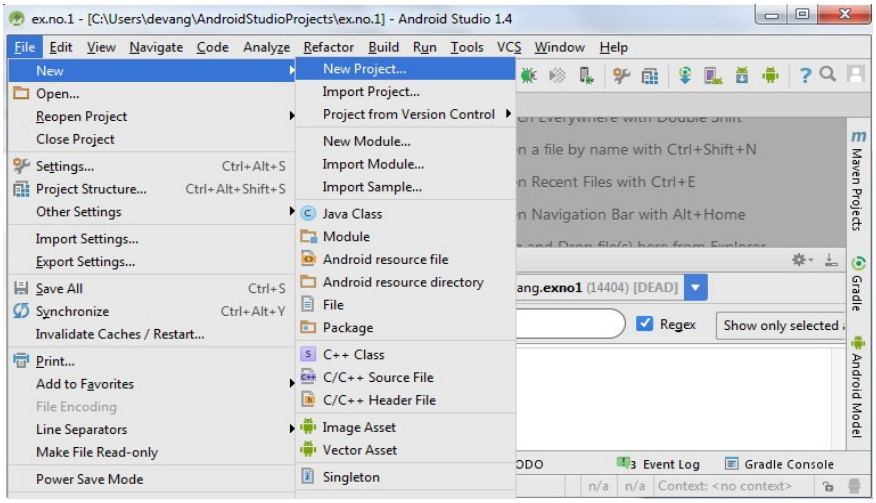
EXP.NO:6	DEVELOP AN APPLICATION THAT MAKES USE OF RSS FEED
DATE: 24-08-18	

Aim :

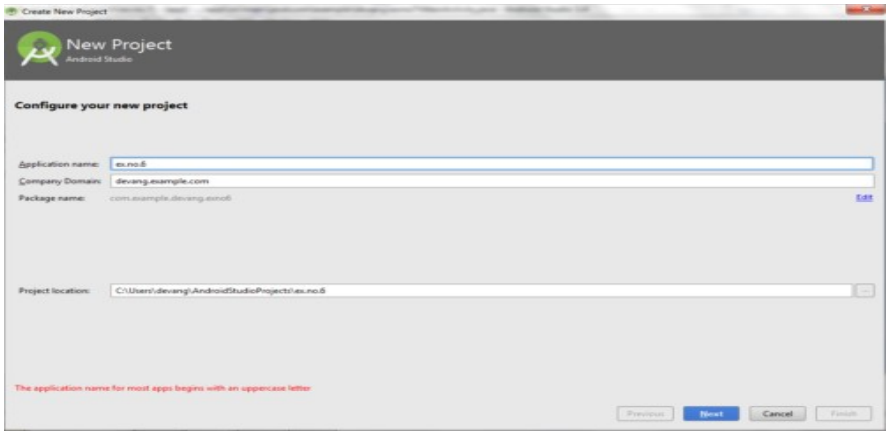
To develop a Android Application that makes use of RSS Feed.

Algorithm :

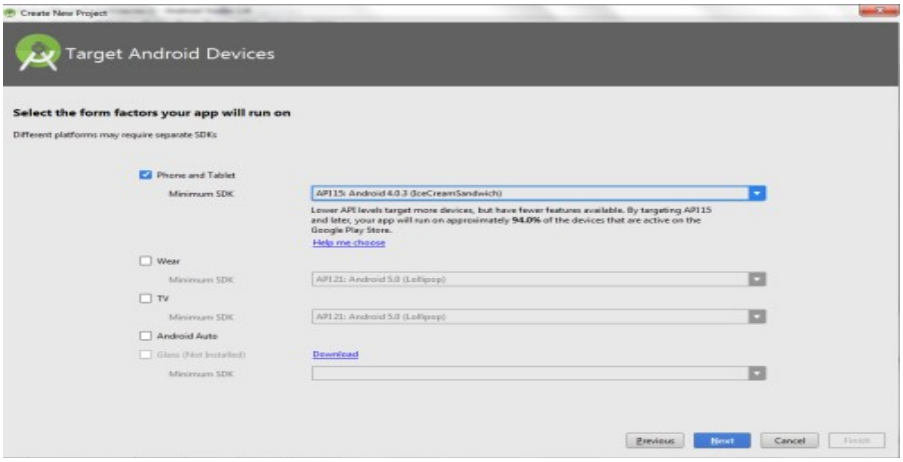
1. Start the program.
2. Creating a New project:
    - Open Android Studio and then click on **File -> New -> New project.**



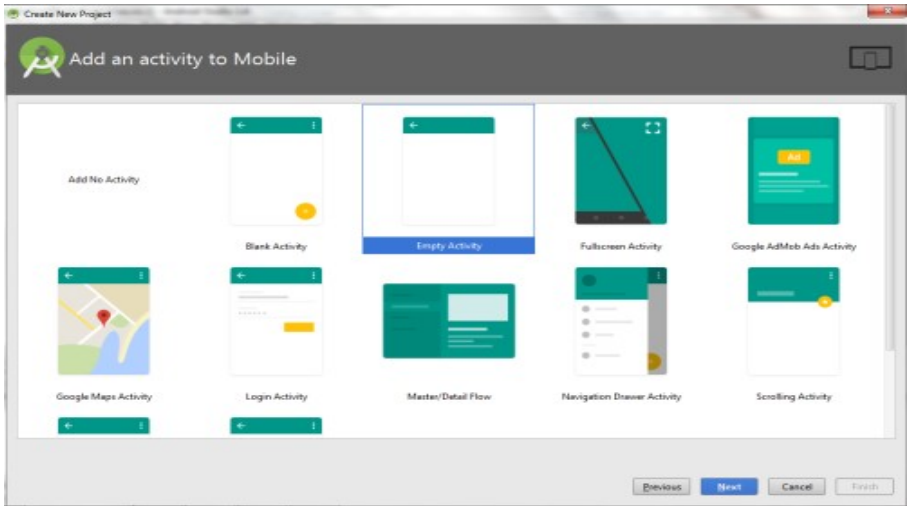
- Then type the Application name as "**ex.no.6"** and click **Next.**



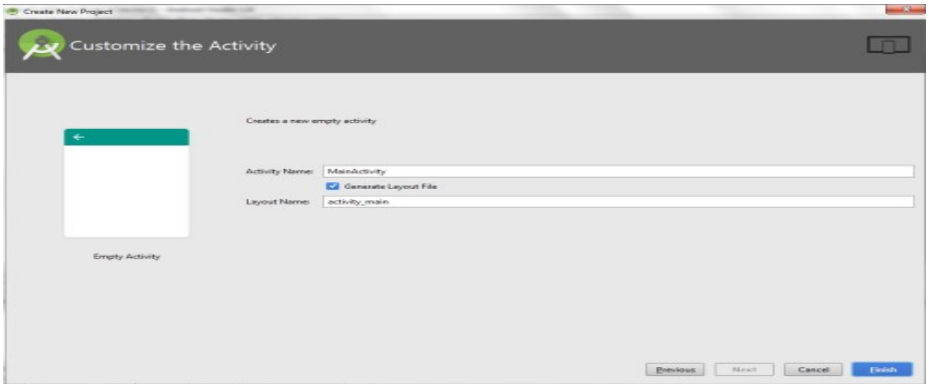
- Then select the **Minimum SDK** as shown below and click **Next**.



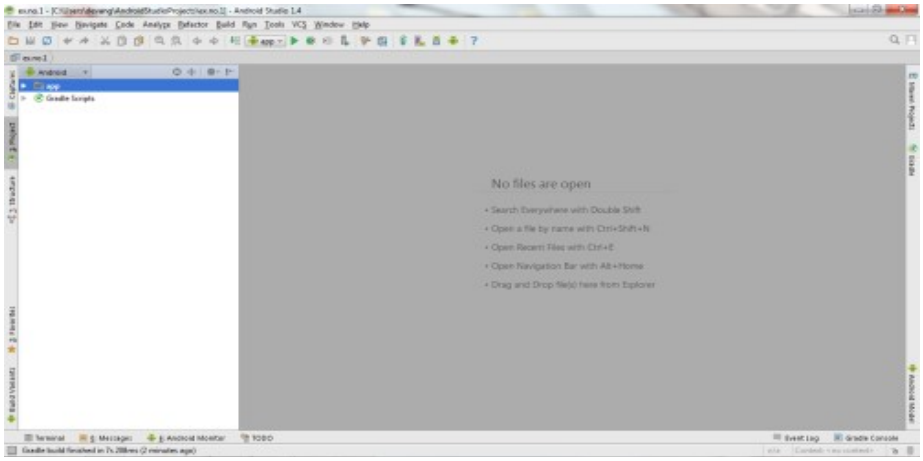
- Then select the **Empty Activity** and click **Next.**



- Finally click F**inish**.

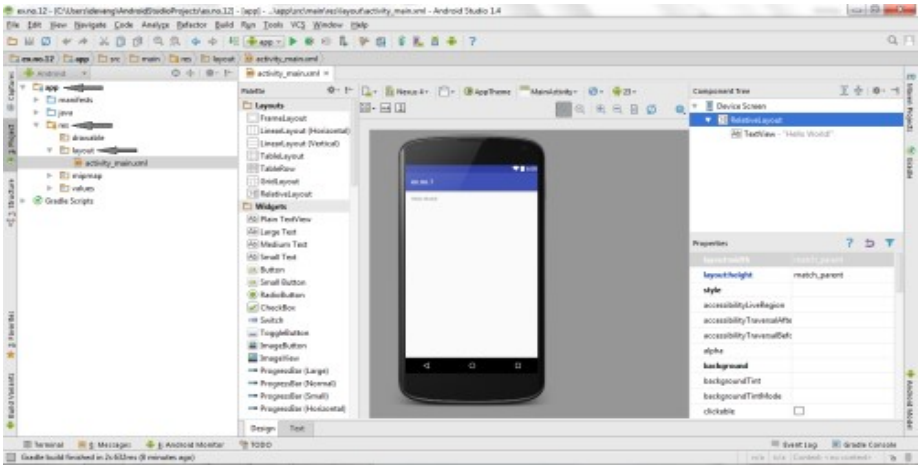


- It will take some time to build and load the project.
- After completion it will look as given below.

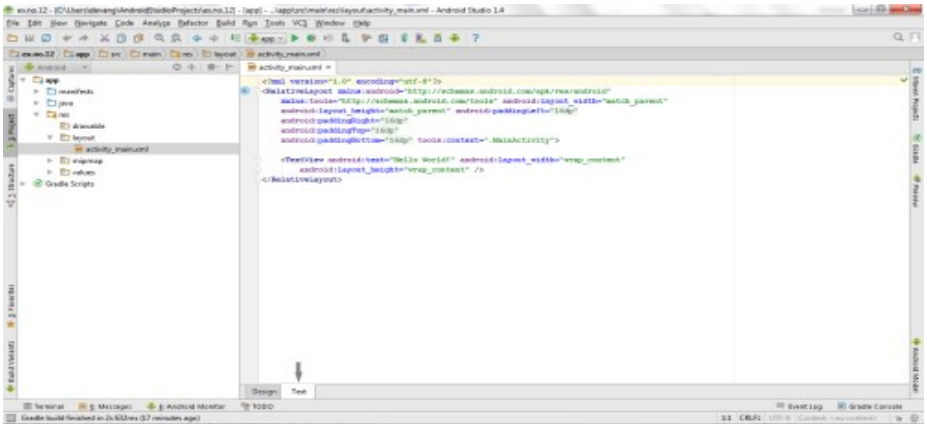


Designing layout for the Android Application:

- Click on **app -> res -> layout -> activity_main.xml**



- Now click on **Text** as shown below.

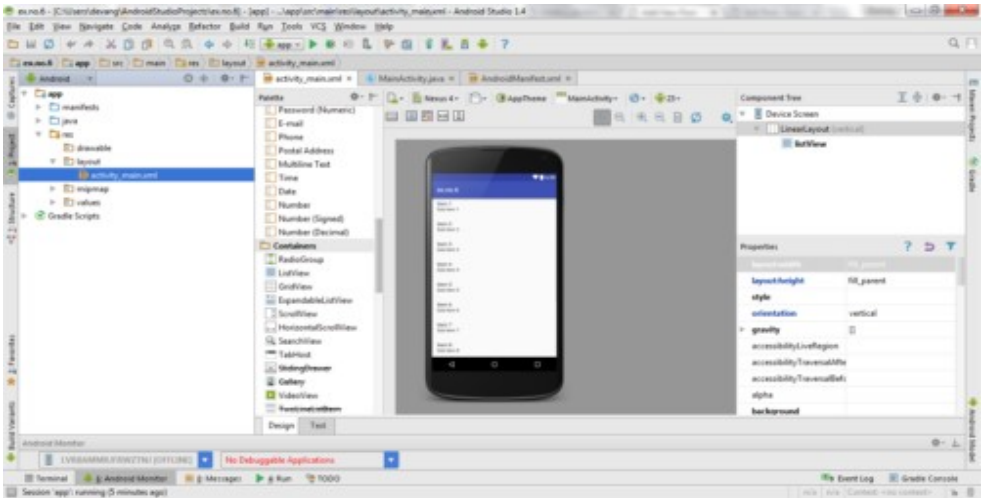


- Then delete the code which is there and type the code as given below.

Code for Activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="fill_parent"
   android:layout_height="fill_parent"
   android:orientation="vertical" >
    <ListView
      android:id="@+id/listView"
      android:layout_width="match_parent"
      android:layout_height="wrap_content" />
</LinearLayout>
```

- Now click on **Design** and your application will look as given below.

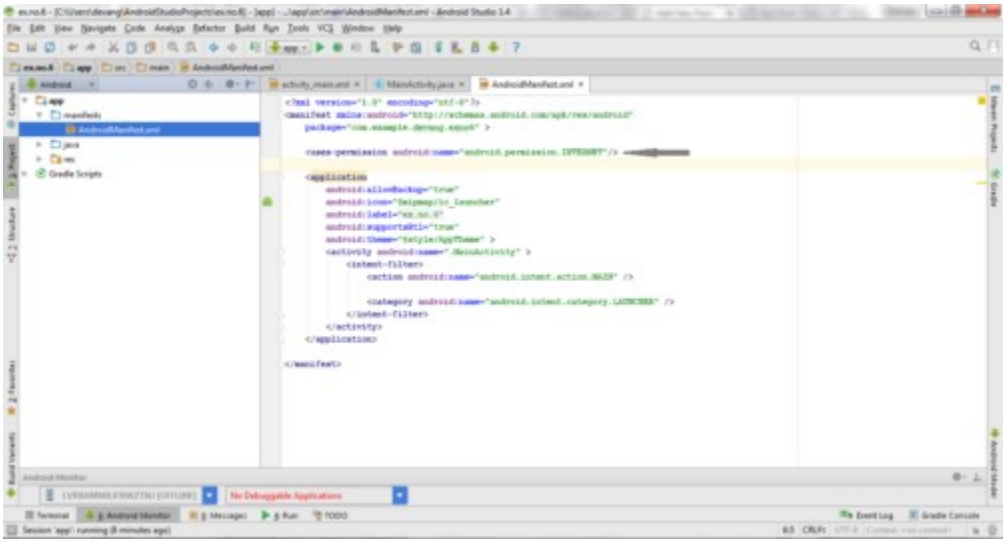


- So now the designing part is completed.

Adding permissions in Manifest for the Android Application:

- Click on **app -> manifests -> AndroidManifest.xml**

- Now include the **INTERNET** permissions in the AndroidManifest.xml file as shown below



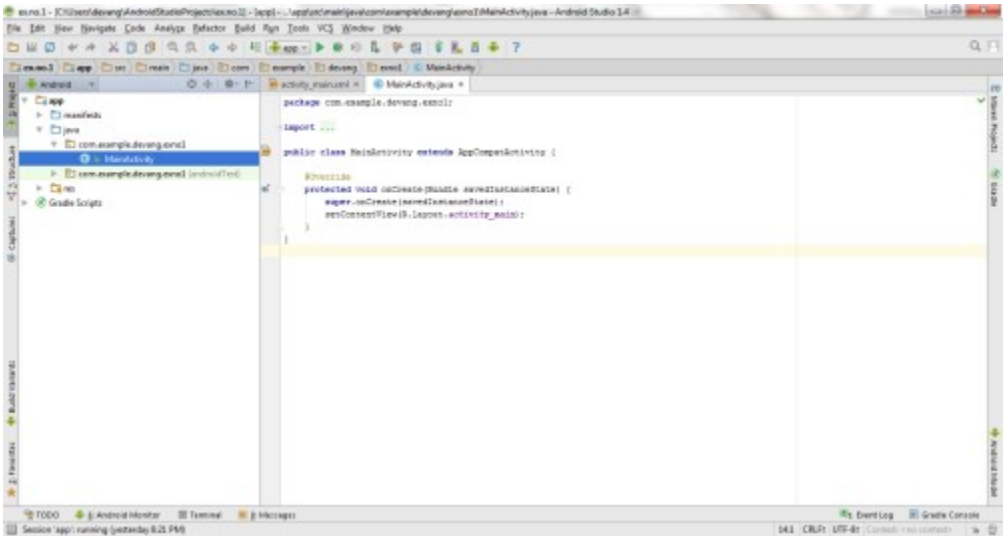
Code for AndroidManifest.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
   package="com.example.exno6" >
    <uses-permission android:name="android.permission.INTERNET"/>
    <application
      android:allowBackup="true"
      android:icon="@mipmap/ic_launcher"
      android:label="@string/app_name"
      android:supportsRtl="true"
      android:theme="@style/AppTheme" >
      <activity android:name=".MainActivity" >
        <intent-filter>
          <action android:name="android.intent.action.MAIN" />
           <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
      </activity>
   </application>
</manifest>
```

- So now the Permissions are added in the Manifest.

Java Coding for the Android Application:

- Click on **app -> java -> com.example.exno6 -> MainActivity.**



- Then delete the code which is there and type the code as given below.

Code for MainActivity.java:

```
package com.example.exno6;

import android.app.ListActivity;
import android.content.Intent;
import android.net.Uri;
import android.os.AsyncTask;
import android.os.Bundle;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import org.xmlpull.v1.XmlPullParser;
import org.xmlpull.v1.XmlPullParserException;
import org.xmlpull.v1.XmlPullParserFactory;
import java.io.IOException;
import java.io.InputStream;
import java.net.MalformedURLException;
import java.net.URL;
import java.util.ArrayList;
import java.util.List;

public class MainActivity extends ListActivity
```

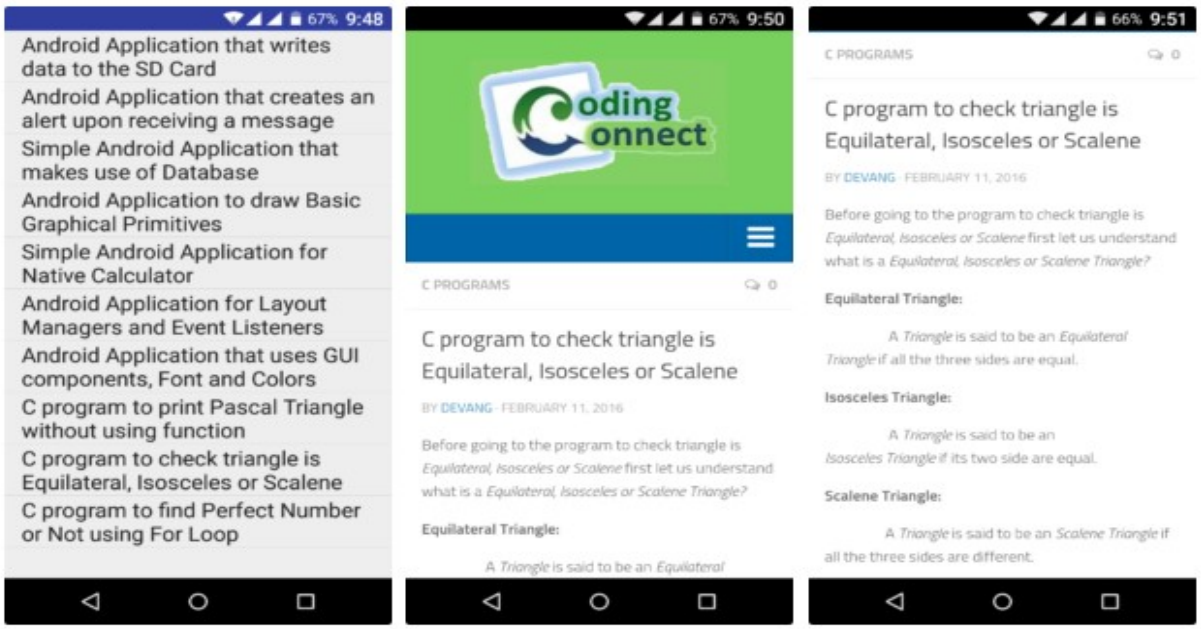
```
{
   List headlines;
   List links;
   @Override
   protected void onCreate(Bundle savedInstanceState)
   {
      super.onCreate(savedInstanceState);
      new MyAsyncTask().execute();
   }
   class MyAsyncTask extends AsyncTask<Object,Void,ArrayAdapter>
   {
      @Override
      protected ArrayAdapter doInBackground(Object[] params)
      {
         headlines = new ArrayList();
         links = new ArrayList();
         try
         {
            URL url = new URL("https://codingconnect.net/feed");
            XmlPullParserFactory factory = XmlPullParserFactory.newInstance();
            factory.setNamespaceAware(false);
            XmlPullParser xpp = factory.newPullParser();
            // We will get the XML from an input stream
            xpp.setInput(getInputStream(url), "UTF_8");
            boolean insideItem = false;
            // Returns the type of current event: START_TAG, END_TAG, etc..
            int eventType = xpp.getEventType();
            while (eventType != XmlPullParser.END_DOCUMENT)
            {
               if (eventType == XmlPullParser.START_TAG)
               {
                  if (xpp.getName().equalsIgnoreCase("item"))
                  {
                     insideItem = true;
                  }
                  else if (xpp.getName().equalsIgnoreCase("title"))
                  {
                     if (insideItem)
                        headlines.add(xpp.nextText()); //extract the headline
                  }
                  else if (xpp.getName().equalsIgnoreCase("link"))
                  {
                     if (insideItem)
                        links.add(xpp.nextText()); //extract the link of article
```

```
                }
            }
            else if(eventType==XmlPullParser.END_TAG &&
xpp.getName().equalsIgnoreCase("item"))
            {
               insideItem=false;
            }
            eventType = xpp.next(); //move to next element
         }

      }
      catch (MalformedURLException e)
      {
         e.printStackTrace();
      }
      catch (XmlPullParserException e)
      {
         e.printStackTrace();
      }
      catch (IOException e)
      {
         e.printStackTrace();
      }
      return null;
   }
   protected void onPostExecute(ArrayAdapter adapter)
   {
      adapter = new ArrayAdapter(MainActivity.this, android.R.layout.simple_list_item_1,
headlines);
      setListAdapter(adapter);
   }
 }
 @Override
 protected void onListItemClick(ListView l, View v, int position, long id)
 {
   Uri uri = Uri.parse((links.get(position)).toString());
   Intent intent = new Intent(Intent.ACTION_VIEW, uri);
   startActivity(intent);
 }
 public InputStream getInputStream(URL url)
 {
   try
   {
      return url.openConnection().getInputStream();
```

```
    }
    catch (IOException e)
    {
       return null;
    }
  }
}
```

- So now the Coding part is also completed.
- Now run the application to see the output.

Output:



Result:

Thus Android Application that makes use of RSS Feed is developed and executed successfully.

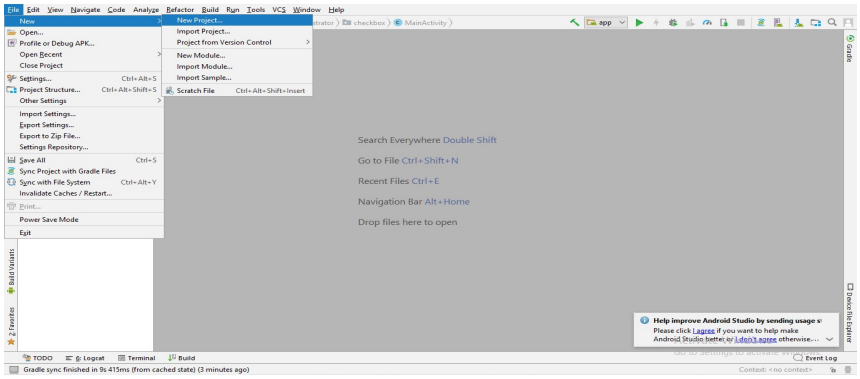
EXP.NO:7	
DATE:31-08-18	**IMPLEMENT AN APPLICATION THAT IMPLEMENTS MULTI THREADING.**

Aim :

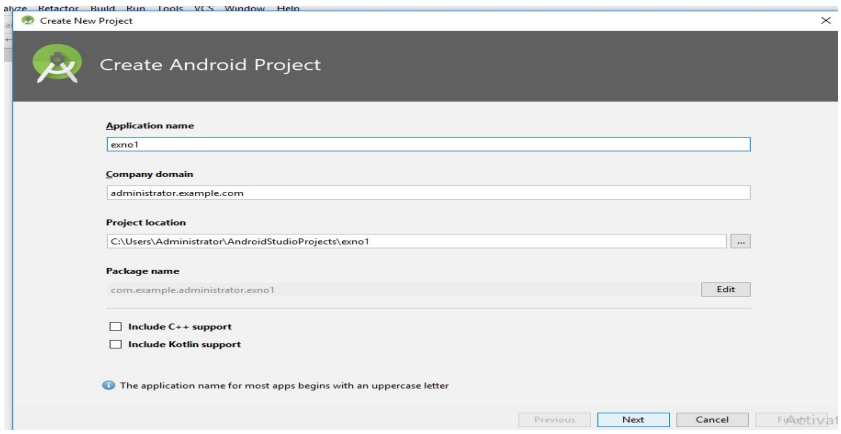
To develop a Android Application that implements Multi threading.

Algorithm:

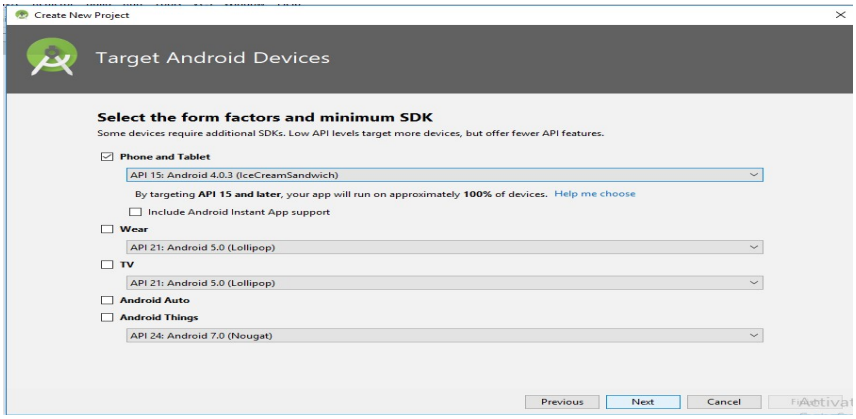
- Start the program.
- Creating a New project:
    - Open Android Studio and then click on **File -> New -> New project.**



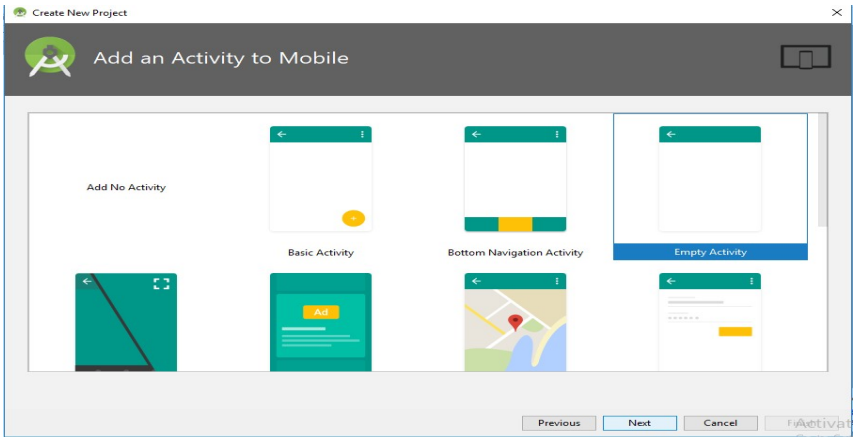
- Then type the Application name and click **Next.**



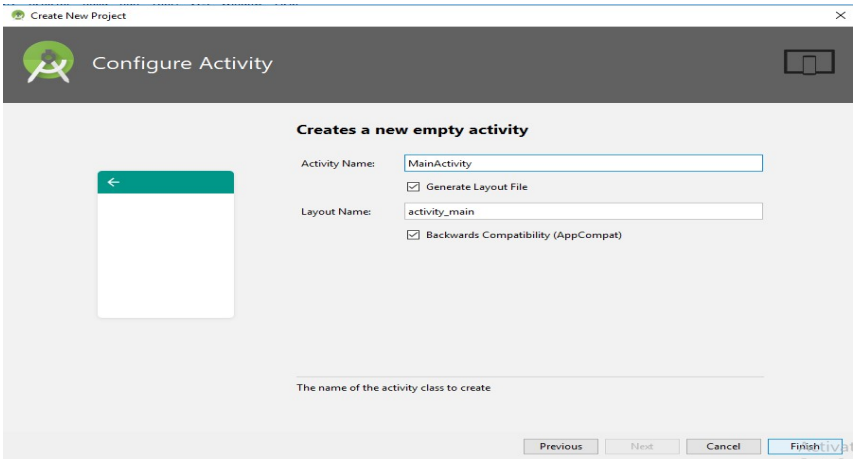
- Then select the **Minimum SDK** as shown below and click **Next**.



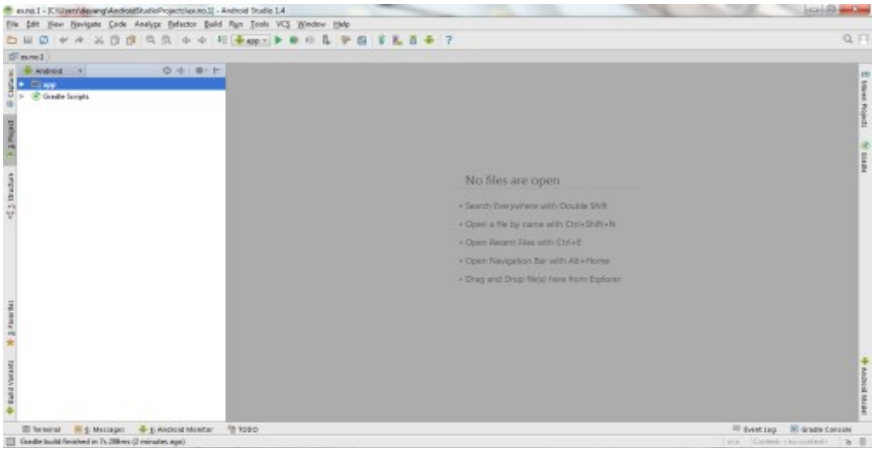
- Then select the **Empty Activity** and click **Next.**



- Finally click F**inish**.

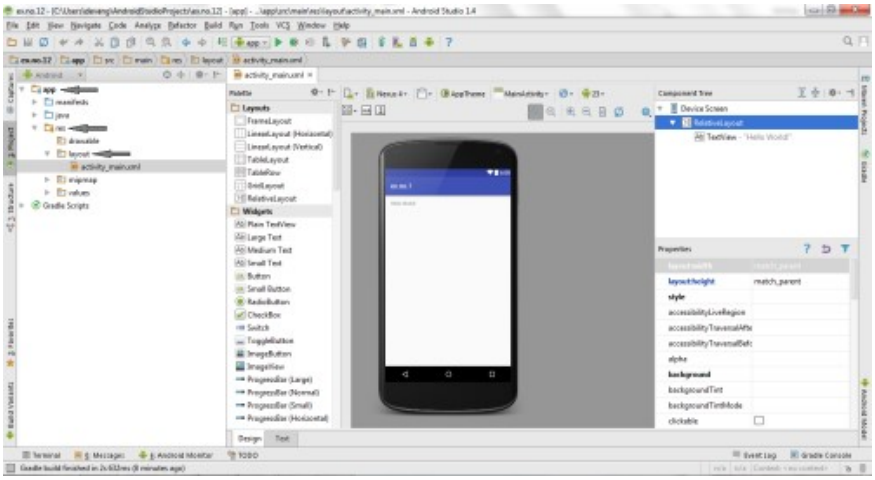


- It will take some time to build and load the project.
- After completion it will look as given below.

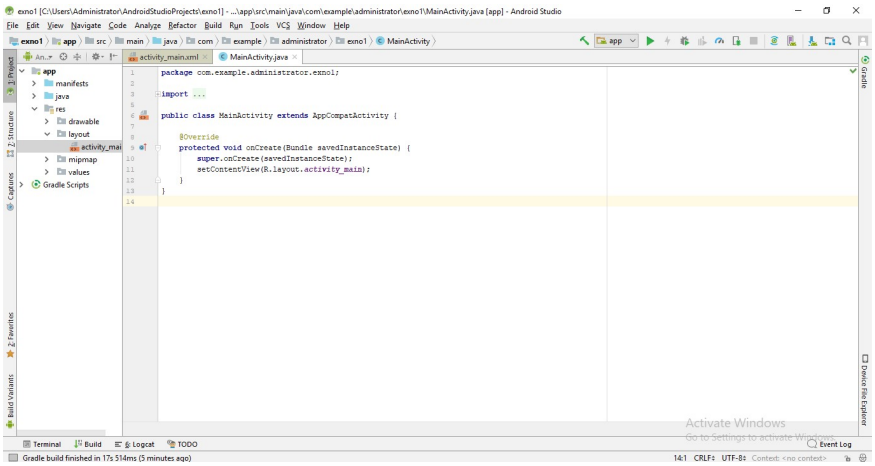


Designing layout for the Android Application.

Click on **app -> res -> layout -> activity_main.xml**



- Now click on **Text** as shown below.



- Then delete the code which is there and type the code as given below

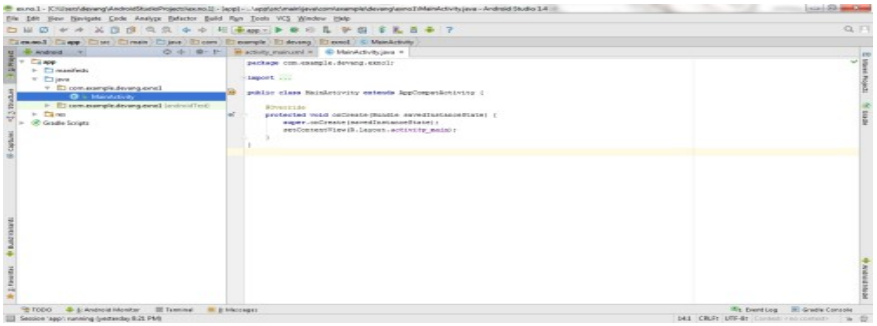
Code for Activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:orientation="vertical" >
   <ImageView
      android:id="@+id/imageView"
      android:layout_width="250dp"
      android:layout_height="250dp"
      android:layout_margin="50dp"
      android:layout_gravity="center" />
   <Button
      android:id="@+id/button"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_margin="10dp"
      android:layout_gravity="center"
      android:text="Load Image 1" />
   <Button
      android:id="@+id/button2"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_margin="10dp"
      android:layout_gravity="center"
      android:text="Load image 2" />
</LinearLayout>
```

- So now the designing part is completed

Java Coding for the Android Application:

Click on **app -> java -> com.example.exno1 -> MainActivity.**



- Then delete the code which is there and type the code as given below.

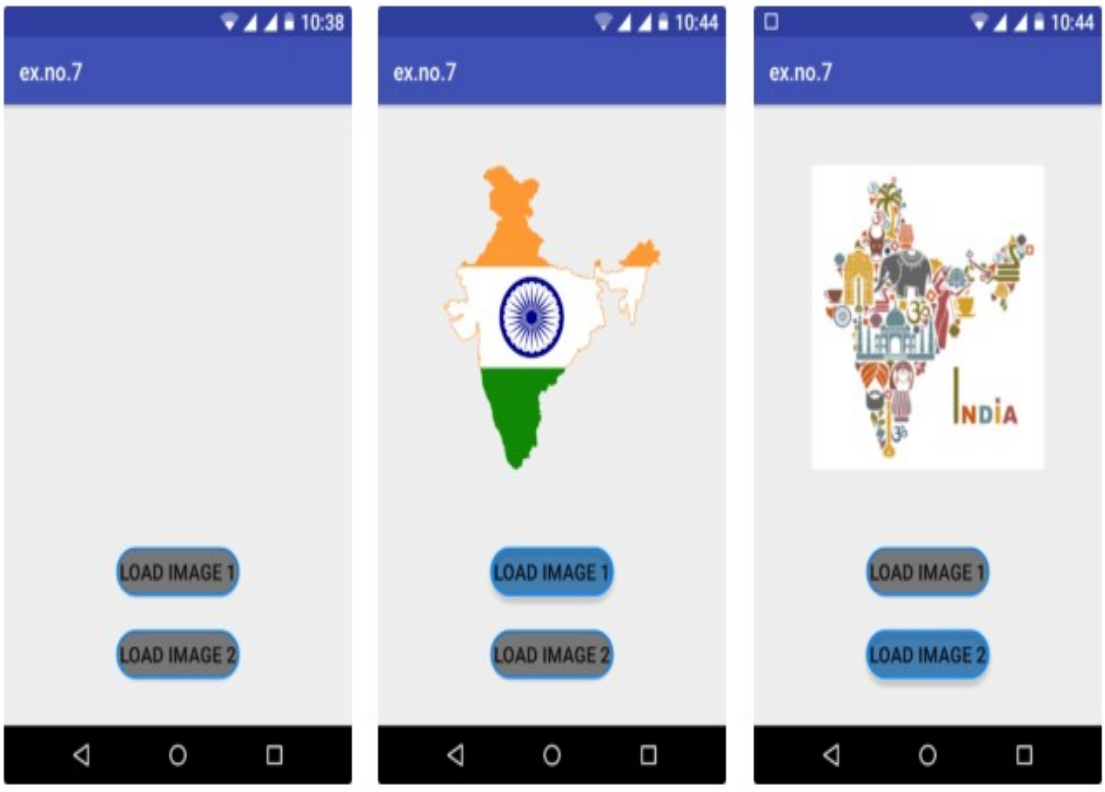
Code for MainActivity.java:

```
package com.example.exno7;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
public class MainActivity extends AppCompatActivity
{
   ImageView img;
   Button bt1,bt2;
   @Override
   protected void onCreate(Bundle savedInstanceState)
   {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      bt1 = (Button)findViewById(R.id.button);
      bt2= (Button) findViewById(R.id.button2);
      img = (ImageView)findViewById(R.id.imageView);
      bt1.setOnClickListener(new View.OnClickListener()
      {
         @Override
         public void onClick(View v)
         {
            new Thread(new Runnable()
            {
               @Override
               public void run()
               {
                  img.post(new Runnable()
                  {
                     @Override
                     public void run()
                     {
                        img.setImageResource(R.drawable.india1);
                  } }); }   }).start();   } });
      bt2.setOnClickListener(new View.OnClickListener()
      {
         @Override
         public void onClick(View v)
         {
            new Thread(new Runnable()    {
               @Override
```

```
        public void run()
        {
           img.post(new Runnable()
           {
              @Override
              public void run()
              {
                 img.setImageResource(R.drawable.india2);
              }   });   }
     }).start();
  } });  } }
```

- So now the Coding part is also completed.
- Now run the application to see the output.
- Stop the program.

Output:



Result:

Thus Android Application that implements Multi threading is developed and executed successfully.

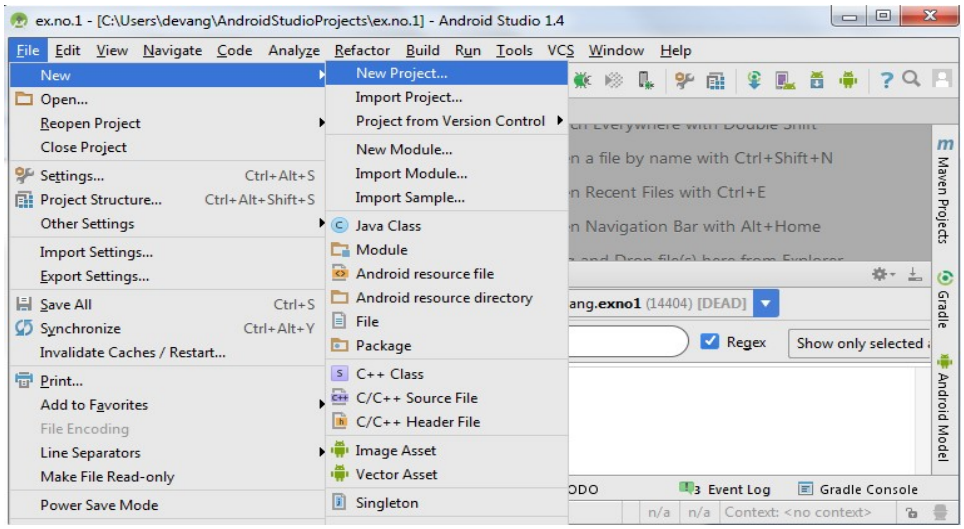
EXP.NO:8	IMPLEMENT AN APPLICATION THAT USES DATA TO THE SDCARD
DATE: 21-09-18	

Aim :

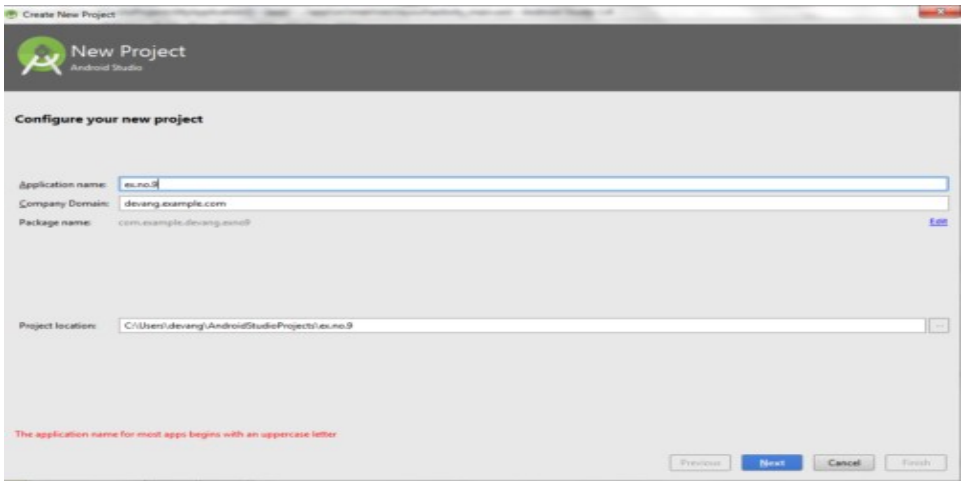
To develop a Android Application that writes data to the SD Card.

Algorithm :

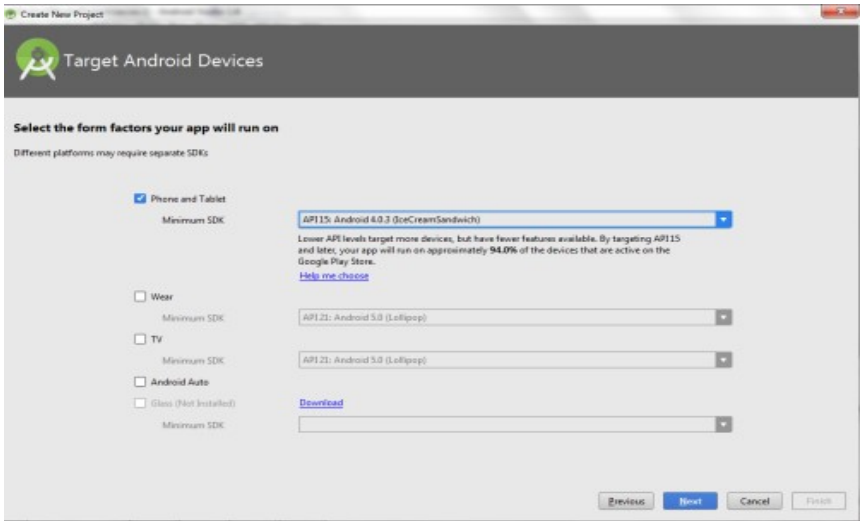
1. Start the program.
2. Creating a New project:
    - Open Android Studio and then click on **File -> New -> New project.**



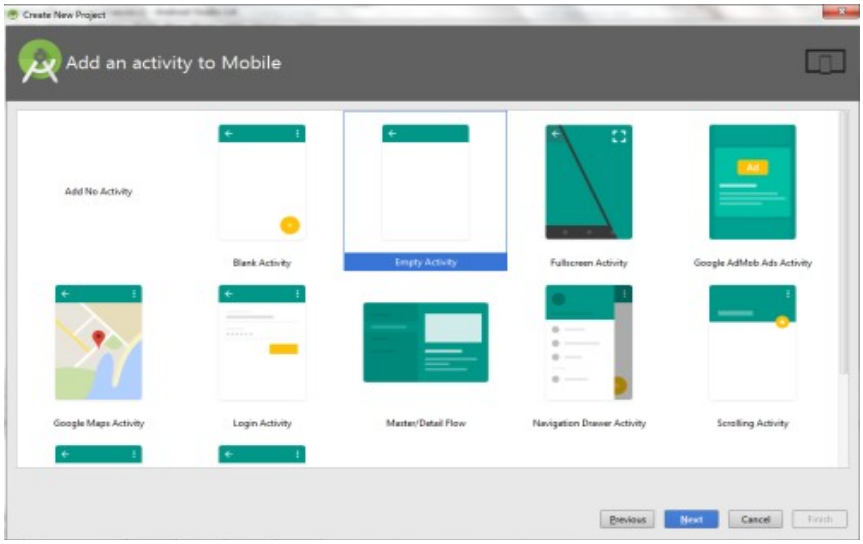
- Then type the Application name as "**ex.no.9"** and click **Next.**



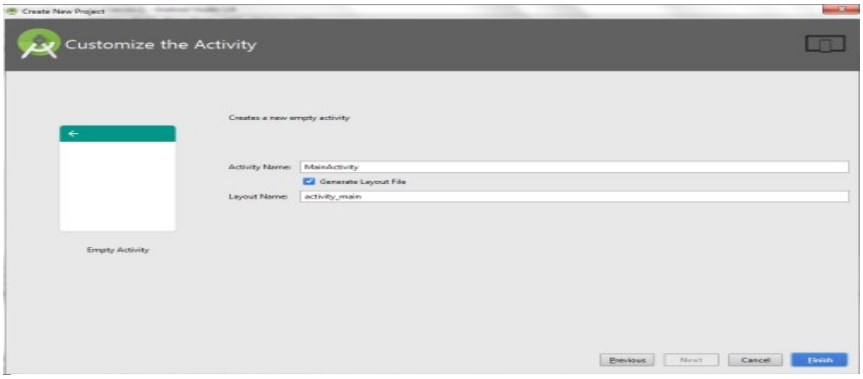
- Then select the **Minimum SDK** as shown below and click **Next**.



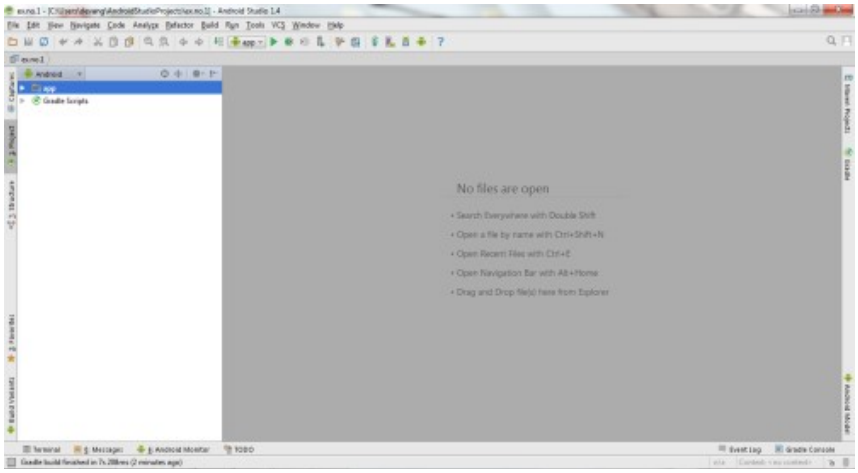
- Then select the **Empty Activity** and click **Next.**



- Finally click F**inish**.

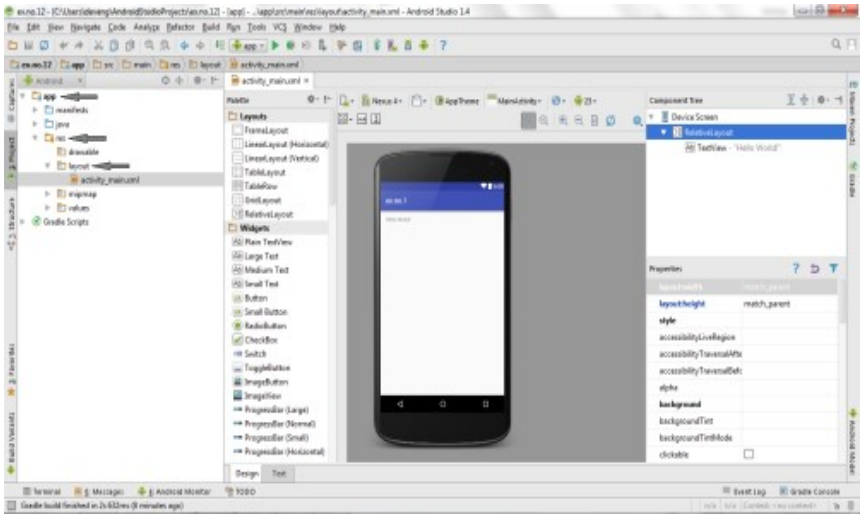


- It will take some time to build and load the project.
- After completion it will look as given below.

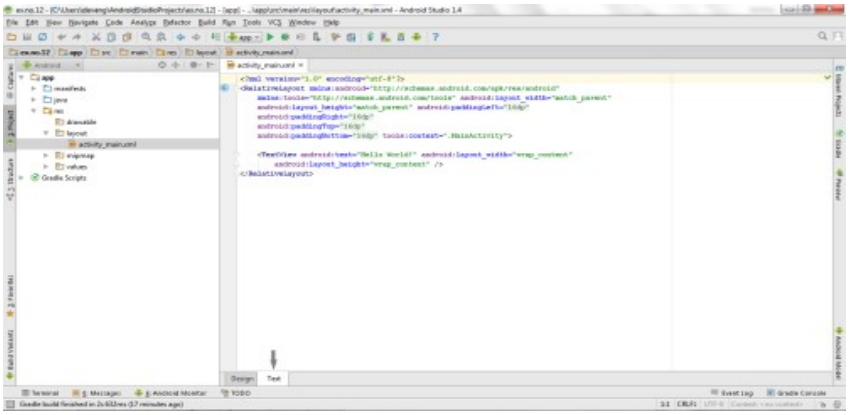


Designing layout for the Android Application:

- Click on **app -> res -> layout -> activity_main.xml.**



- Now click on **Text** as shown below.

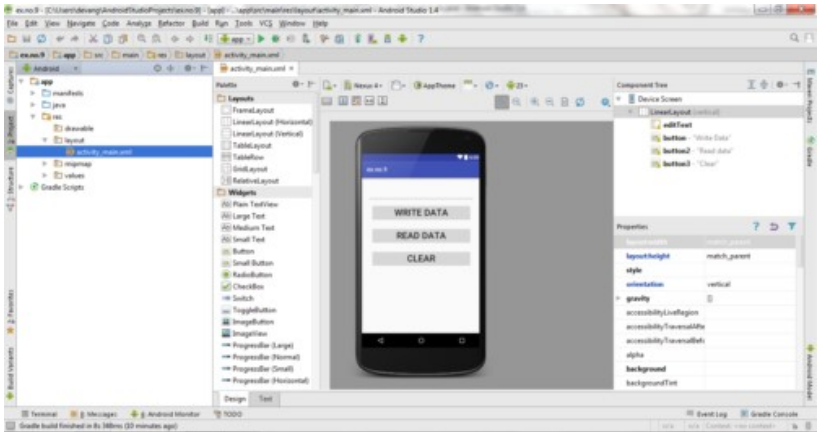


- Then delete the code which is there and type the code as given below.

Code for Activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:layout_margin="20dp"
   android:orientation="vertical">
    <EditText
      android:id="@+id/editText"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:singleLine="true"
      android:textSize="30dp" />
    <Button
      android:id="@+id/button"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:layout_margin="10dp"
      android:text="Write Data"
      android:textSize="30dp" />
    <Button
      android:id="@+id/button2"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:layout_margin="10dp"
      android:text="Read data"
      android:textSize="30dp" />
    <Button
      android:id="@+id/button3"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:layout_margin="10dp"
      android:text="Clear"
      android:textSize="30dp" />
</LinearLayout>
```

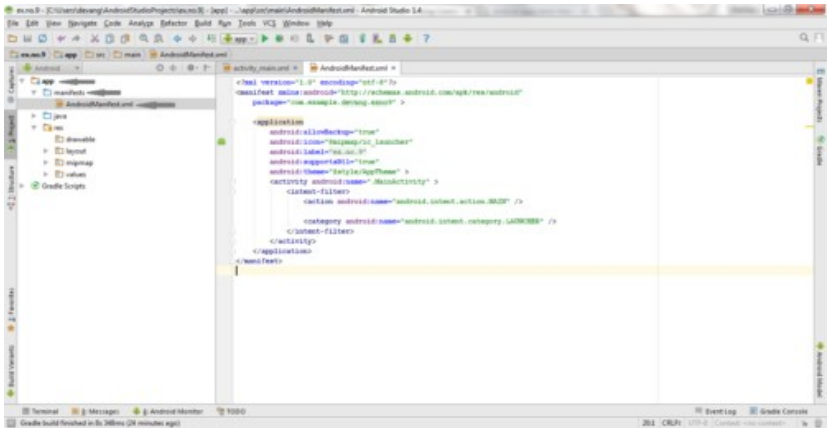
- Now click on **Design** and your application will look as given below.



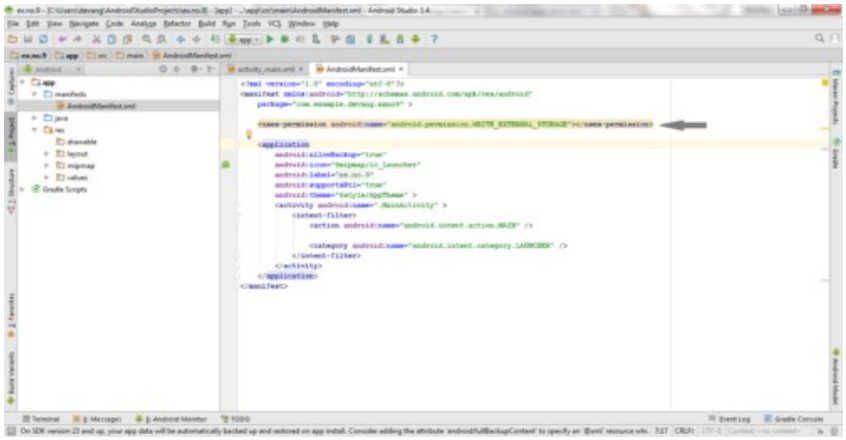
- So now the designing part is completed.

Adding permissions in Manifest for the Android Application:

- Click on **app -> manifests -> AndroidManifest.xml**



- Now include the **WRITE_EXTERNAL_STORAGE** permissions in the AndroidManifest.xml file as shown below



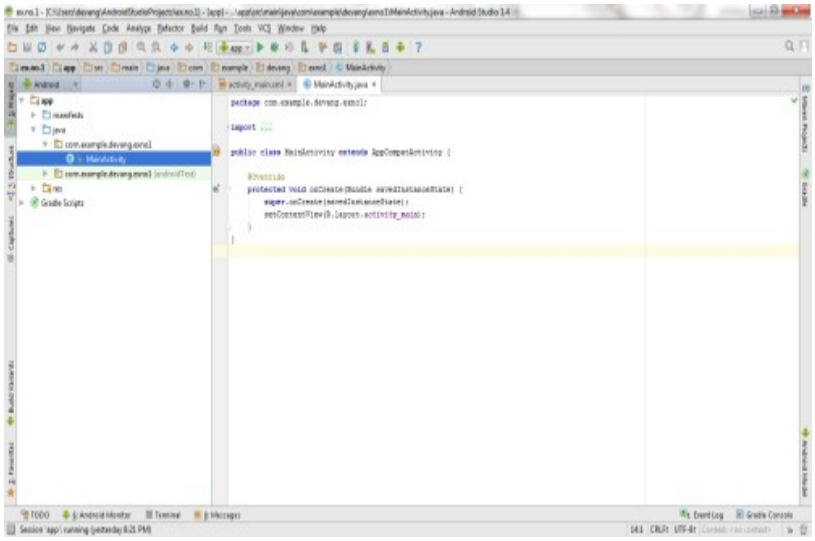
Code for AndroidManifest.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
   package="com.example.exno9" >
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"></uses-
permission>
    <application
      android:allowBackup="true"
      android:icon="@mipmap/ic_launcher"
      android:label="@string/app_name"
      android:supportsRtl="true"
      android:theme="@style/AppTheme" >
      <activity android:name=".MainActivity" >
        <intent-filter>
          <action android:name="android.intent.action.MAIN" />
           <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
      </activity>
   </application>
</manifest>
```

- So now the Permissions are added in the Manifest.

Java Coding for the Android Application:

- Click on **app -> java -> com.example.exno9 -> MainActivity.**



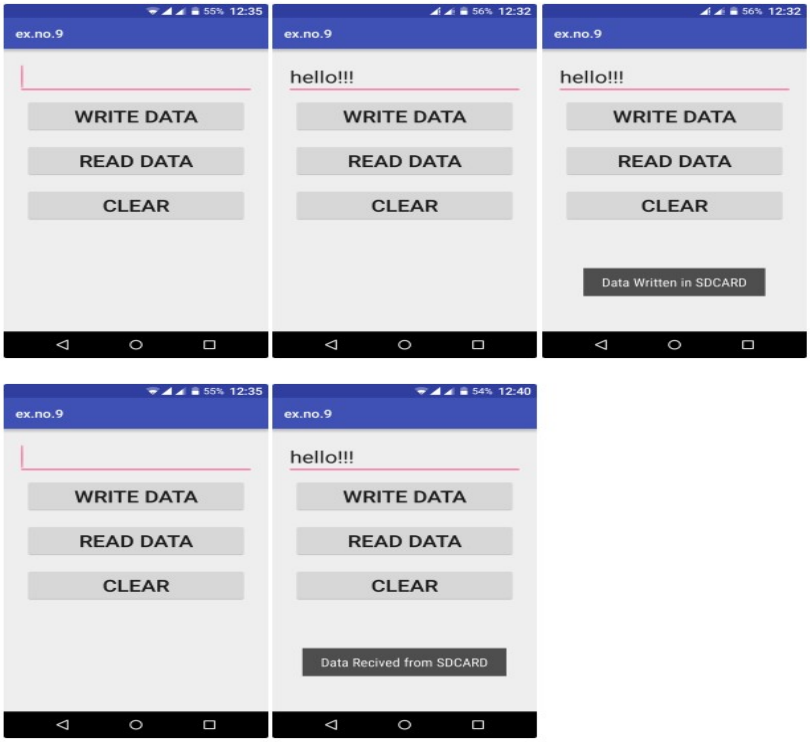
- Then delete the code which is there and type the code as given below.

Code for MainActivity.java:

```
package com.example.exno9;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.InputStreamReader;
public class MainActivity extends AppCompatActivity
{
   EditText e1;
   Button write,read,clear;
   @Override
   protected void onCreate(Bundle savedInstanceState)
   {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
       e1= (EditText) findViewById(R.id.editText);
      write= (Button) findViewById(R.id.button);
      read= (Button) findViewById(R.id.button2);
      clear= (Button) findViewById(R.id.button3);
       write.setOnClickListener(new View.OnClickListener()
      {
         @Override
         public void onClick(View v)
         {
            String message=e1.getText().toString();
            try
            {
               File f=new File("/sdcard/myfile.txt");
               f.createNewFile();
               FileOutputStream fout=new FileOutputStream(f);
               fout.write(message.getBytes());
               fout.close();
               Toast.makeText(getBaseContext(),"Data Written in
SDCARD",Toast.LENGTH_LONG).show();
            }
            catch (Exception e)
```

```
            {
               Toast.makeText(getBaseContext(),e.getMessage(),Toast.LENGTH_LONG).show();
            }
         }
      });
       read.setOnClickListener(new View.OnClickListener()
      {
         @Override
         public void onClick(View v)
         {
            String message;
            String buf = "";
            try
            {
               File f = new File("/sdcard/myfile.txt");
               FileInputStream fin = new FileInputStream(f);
               BufferedReader br = new BufferedReader(new InputStreamReader(fin));
               while ((message = br.readLine()) != null)
               {
                  buf += message;
               }
               e1.setText(buf);
               br.close();
               fin.close();
               Toast.makeText(getBaseContext(),"Data Recived from
SDCARD",Toast.LENGTH_LONG).show();
            }
            catch (Exception e)
            {
               Toast.makeText(getBaseContext(), e.getMessage(), Toast.LENGTH_LONG).show();
            }
         }
      });

      clear.setOnClickListener(new View.OnClickListener()
      {
         @Override
         public void onClick(View v)
         {
            e1.setText("");
         }
      });
   }
}
```

Output:



Result:

Thus Android Application that writes data to the SD Card is developed and executed successfully.

EXP.NO:9	
DATE: 28-09-18	**IMPLEMENT AN APPLICATION THAT CREATES AN ALERT UPON RECEIVING A MESSAGE.**

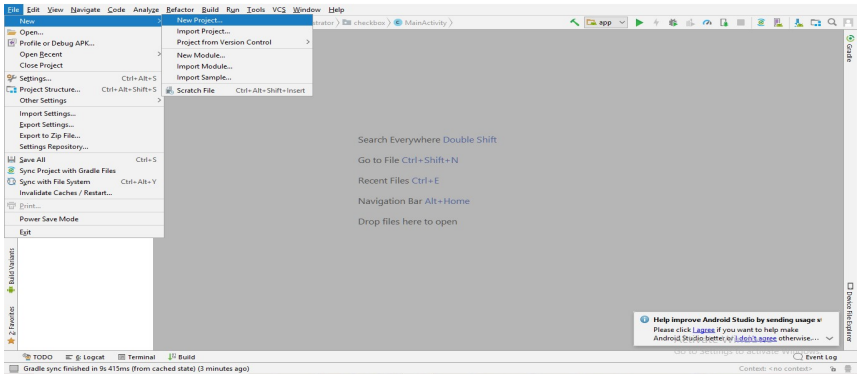
Aim :

To develop a Android Application that creates an alert upon receiving a message.

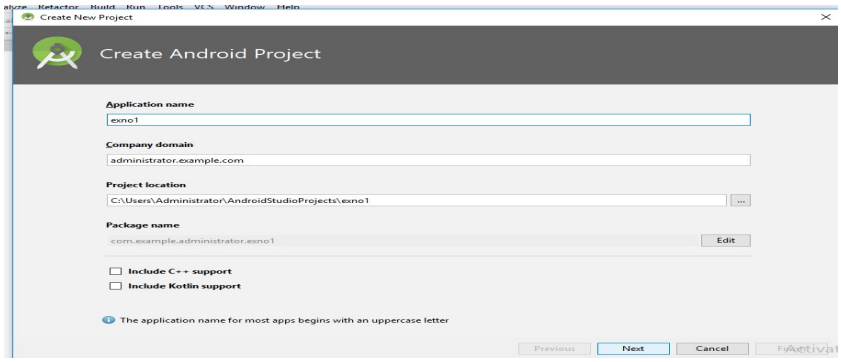
Algorithm :

1. Start the program.
2. Creating a New project.

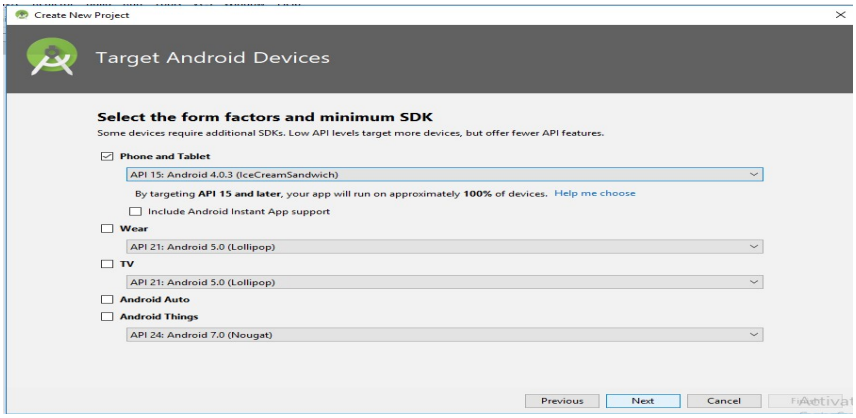
- Open Android Studio and then click on **File -> New -> New project.**



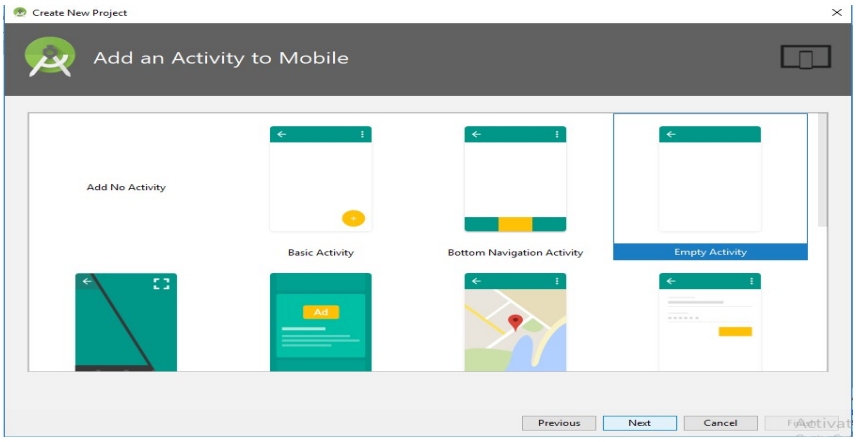
- Then type the Application name and click **Next.**



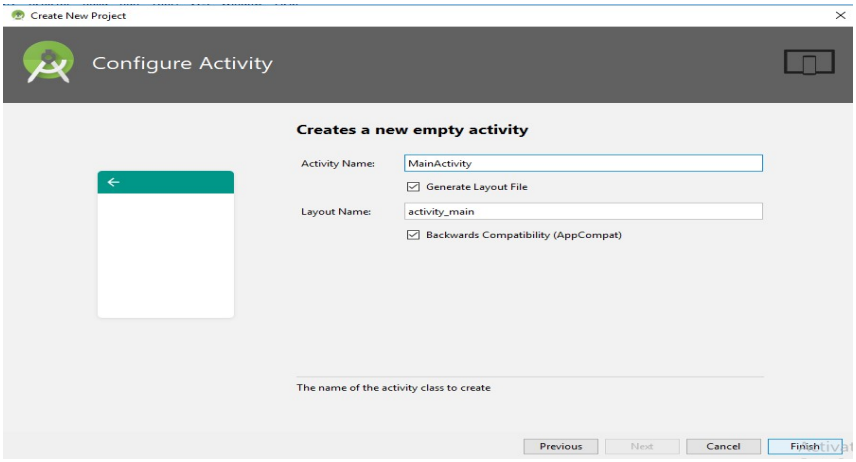
- Then select the **Minimum SDK** as shown below and click **Next**.



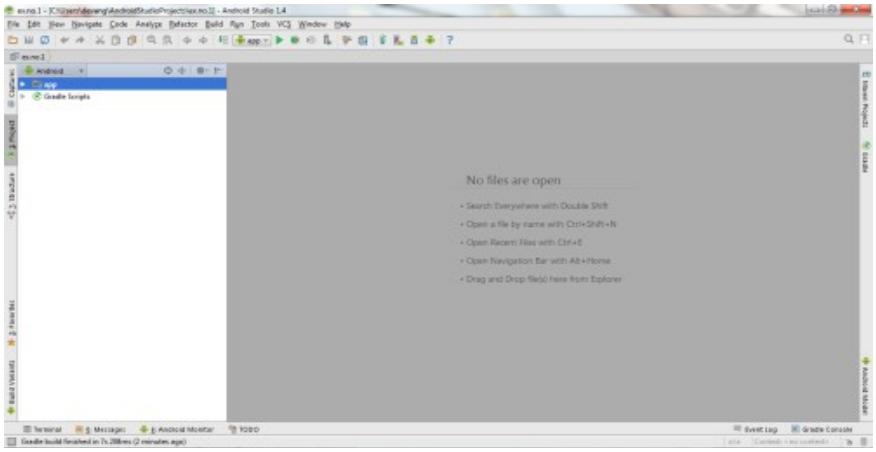
- Then select the **Empty Activity** and click **Next.**



- Finally click F**inish**.

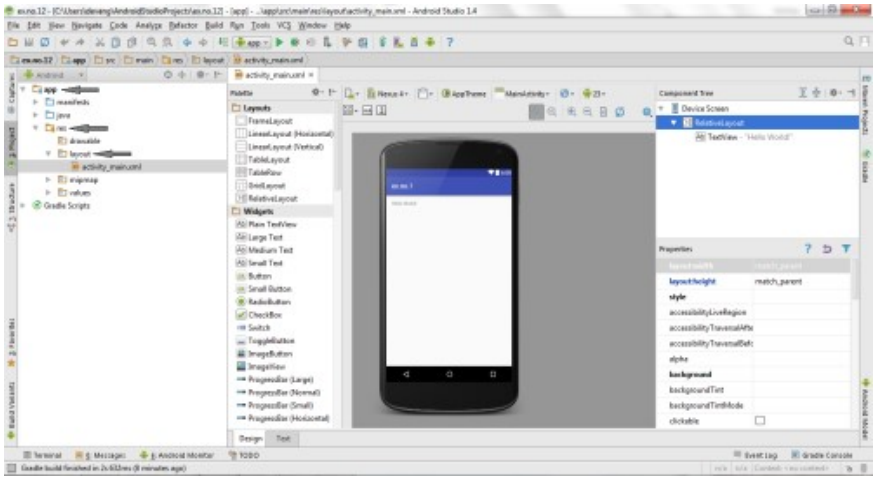


- It will take some time to build and load the project.
- After completion it will look as given below.

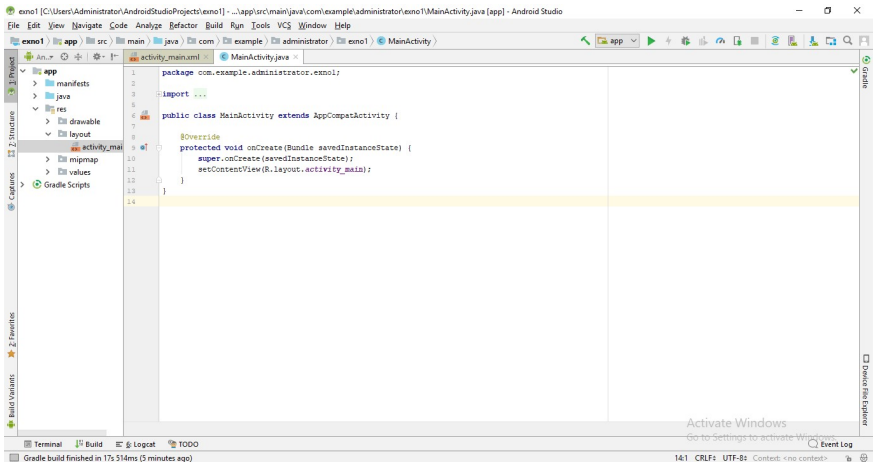


Designing layout for the Android Application.

Click on **app -> res -> layout -> activity_main.xml**



- Now click on **Text** as shown below.



- Then delete the code which is there and type the code as given below

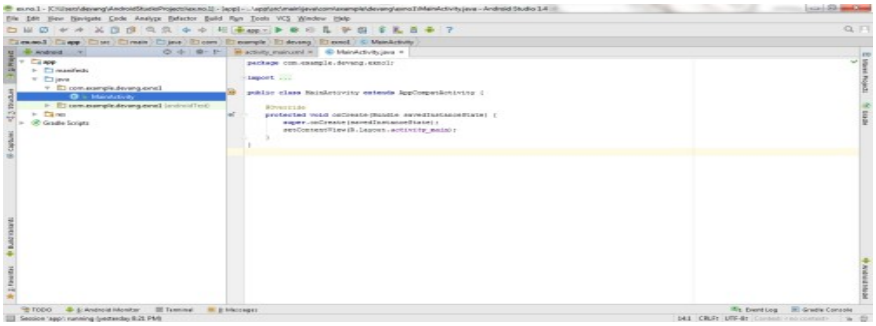
Code for Activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:layout_margin="10dp"
   android:orientation="vertical">
    <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Message"
      android:textSize="30sp" />
    <EditText
      android:id="@+id/editText"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:singleLine="true"
      android:textSize="30sp" />
    <Button
      android:id="@+id/button"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_margin="30dp"
      android:layout_gravity="center"
      android:text="Notify"
      android:textSize="30sp"/>
 </LinearLayout>
```

- So now the designing part is completed

Java Coding for the Android Application:

- Click on **app -> java -> com.example.exno1 -> MainActivity.**



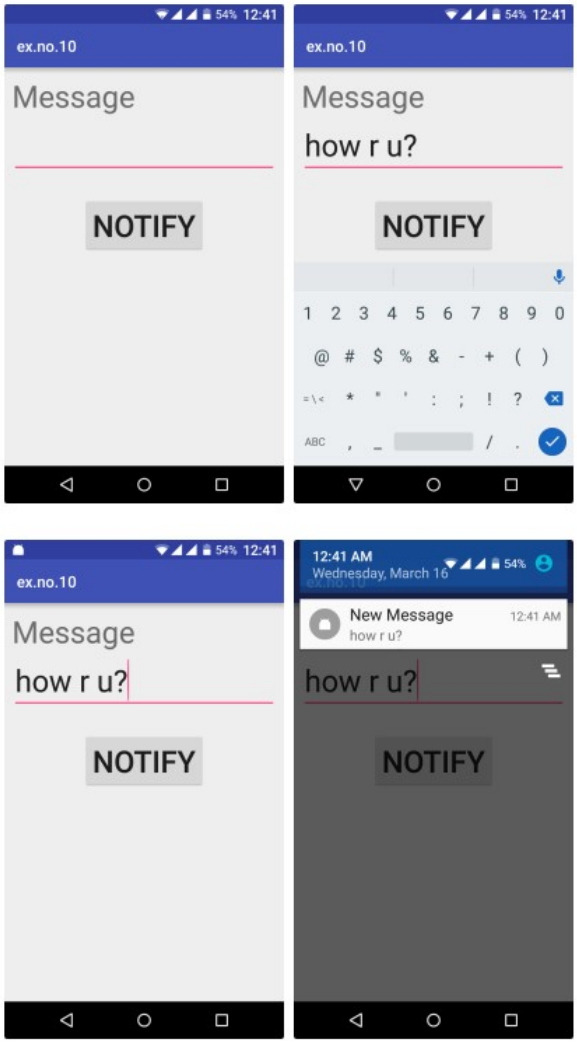
- Then delete the code which is there and type the code as given below.

Code for MainActivity.java:

```
package com.example.exno10;

import android.app.Notification;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

public class MainActivity extends AppCompatActivity
{
   Button notify;
   EditText e;
   @Override
   protected void onCreate(Bundle savedInstanceState)
   {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);

      notify= (Button) findViewById(R.id.button);
      e= (EditText) findViewById(R.id.editText);

      notify.setOnClickListener(new View.OnClickListener()
      {
         @Override
         public void onClick(View v)
         {
            Intent intent = new Intent(MainActivity.this, SecondActivity.class);
            PendingIntent pending = PendingIntent.getActivity(MainActivity.this, 0, intent, 0);
            Notification noti = new Notification.Builder(MainActivity.this).setContentTitle("New
Message").setContentText(e.getText().toString()).setSmallIcon(R.mipmap.ic_launcher).setConte
ntIntent(pending).build();
            NotificationManager manager = (NotificationManager)
getSystemService(NOTIFICATION_SERVICE);
            noti.flags |= Notification.FLAG_AUTO_CANCEL;
            manager.notify(0, noti);
         }
      });
   }}
```

- So now the Coding part is also completed.
- Now run the application to see the output.
- Stop the program.

Output:



Result:

Thus Android Application that creates an alert upon receiving a message is developed and executed successfully.

EXP.NO: 10	WRITE AN MOBILE APPLICATION THAT CREATES AN ALARM CLOCK.
DATE: 12-10-18	

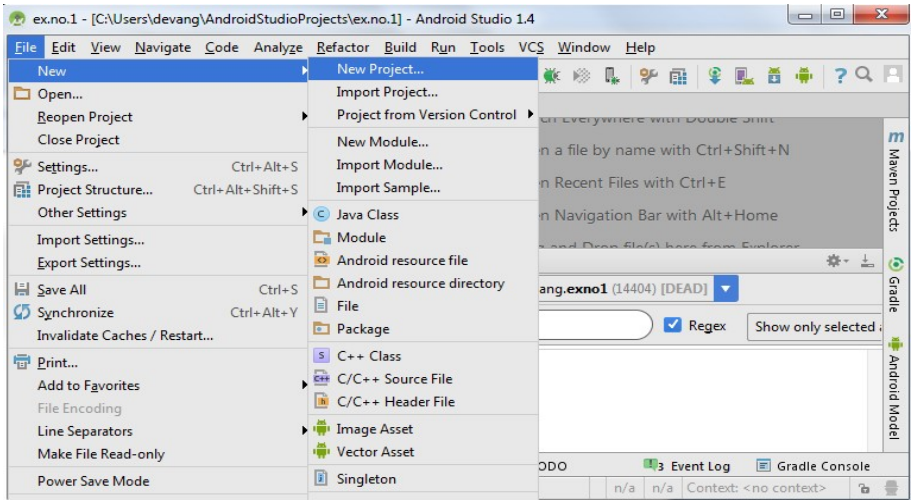
Aim:

To develop a Android Application that creates Alarm Clock.

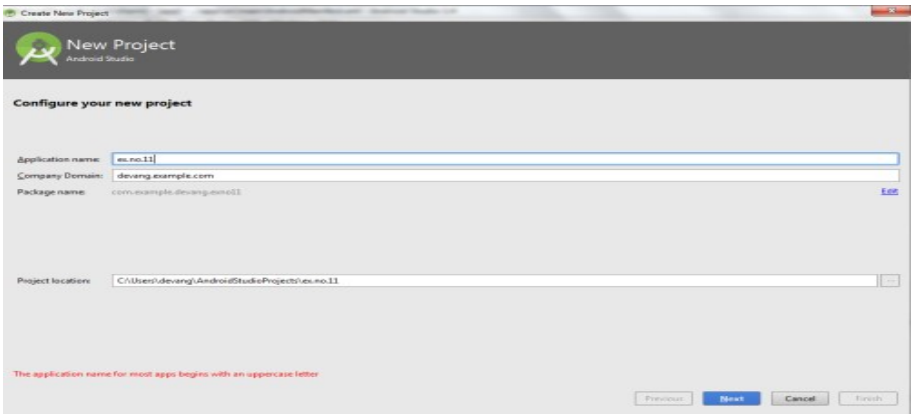
Algorithm:

1. Start the program.
2. Creating a New project.

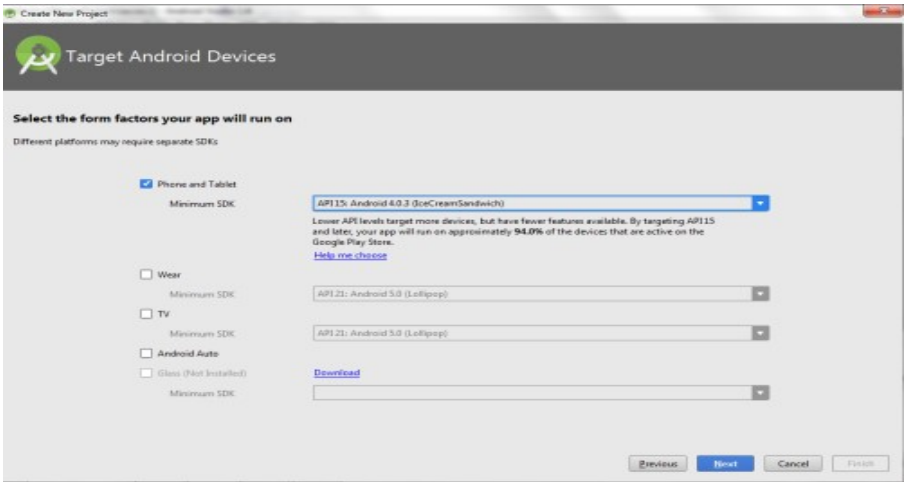
    - Open Android Studio and then click on **File -> New -> New project.**



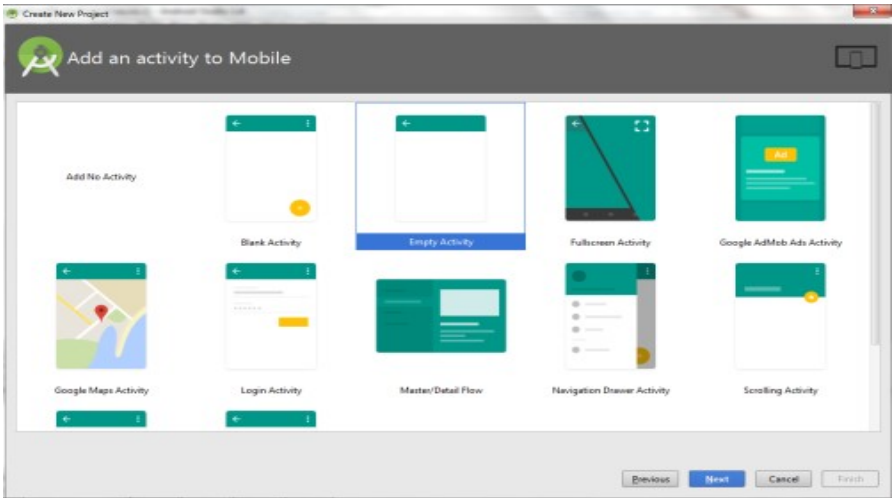
    - Then type the Application name as "**ex.no.10"** and click **Next.**



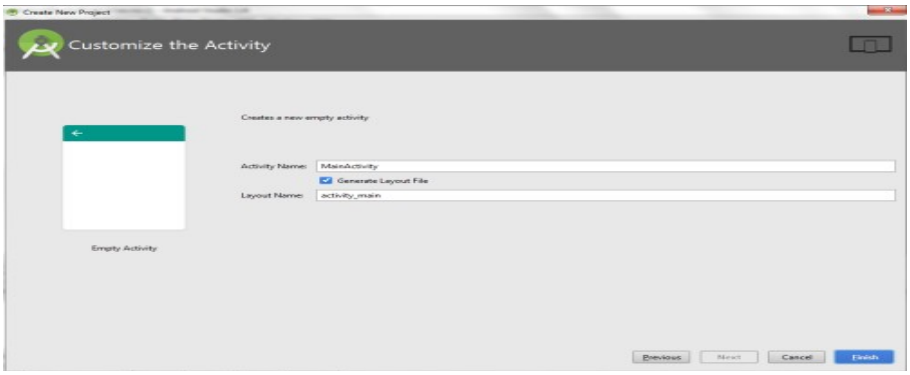
- Then select the **Minimum SDK** as shown below and click **Next**.



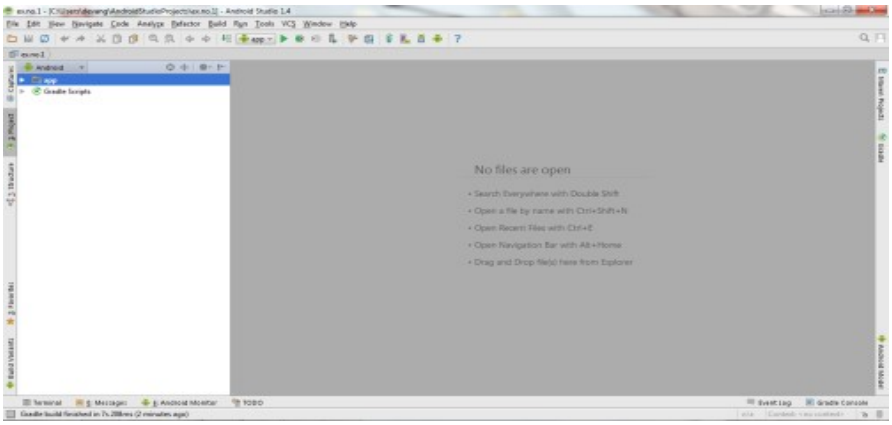
- Then select the **Empty Activity** and click **Next.**



- Finally click F**inish**.

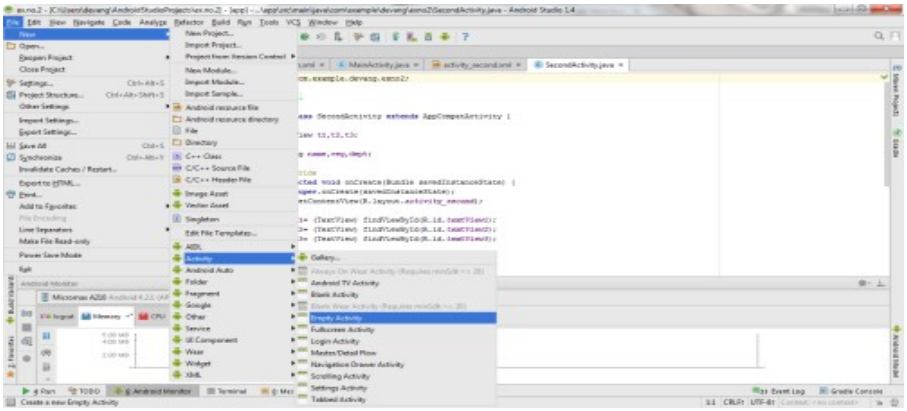


- It will take some time to build and load the project.
- After completion it will look as given below.

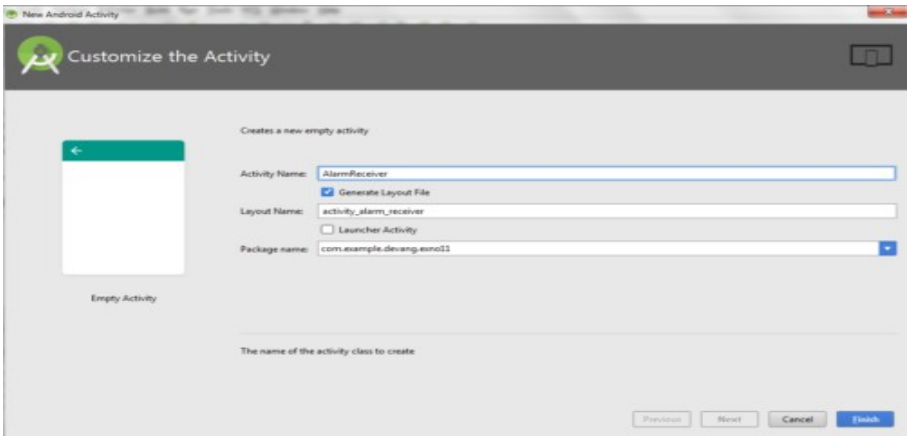


Creating Second Activity for the Android Application:

- Click on **File -> New -> Activity -> Empty Activity.**



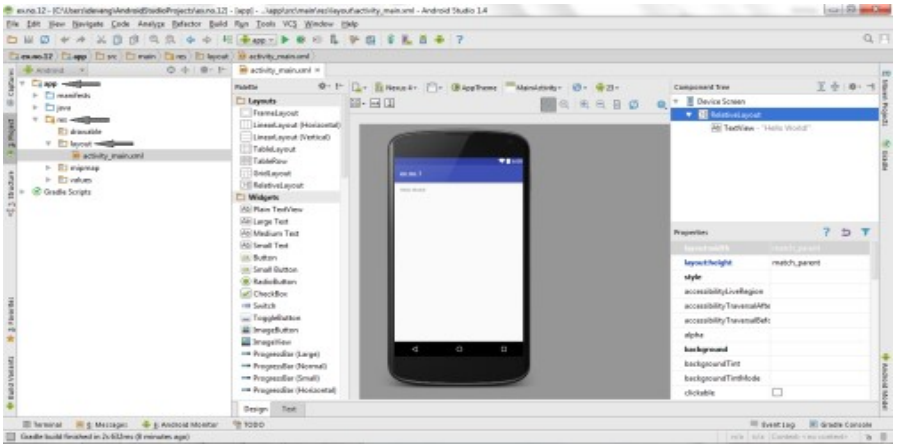
- Type the Activity Name as **AlarmReceiver** and click **Finish** button.



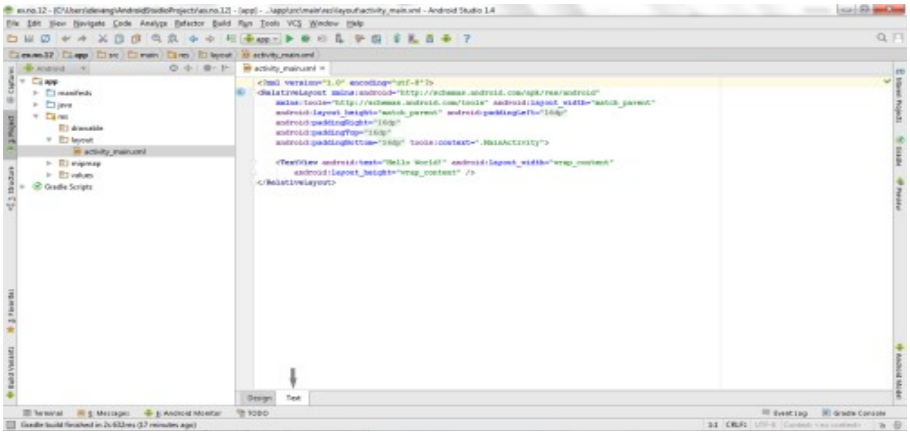
- Thus Second Activity For the application is created.

Designing layout for the Android Application:

- Click on **app -> res -> layout -> activity_main.xml.**



- Now click on **Text** as shown below.



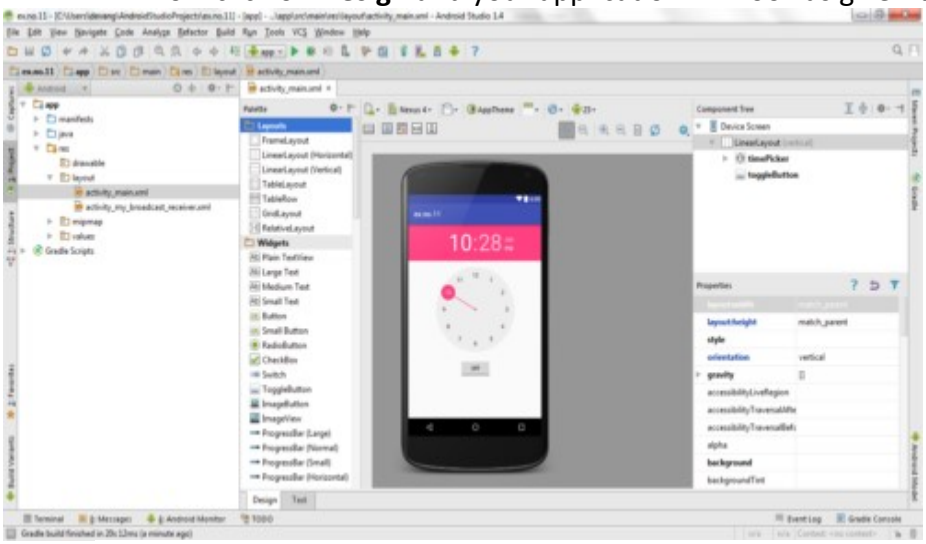
- Then delete the code which is there and type the code as given below.

Code for Activity_main.xml:

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

   <TimePicker
      android:id="@+id/timePicker"
      android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
    android:layout_gravity="center" />
  <ToggleButton
    android:id="@+id/toggleButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_margin="20dp"
    android:checked="false"
    android:onClick="OnToggleClicked" />
</LinearLayout>
```

- Now click on **Design** and your application will look as given below.

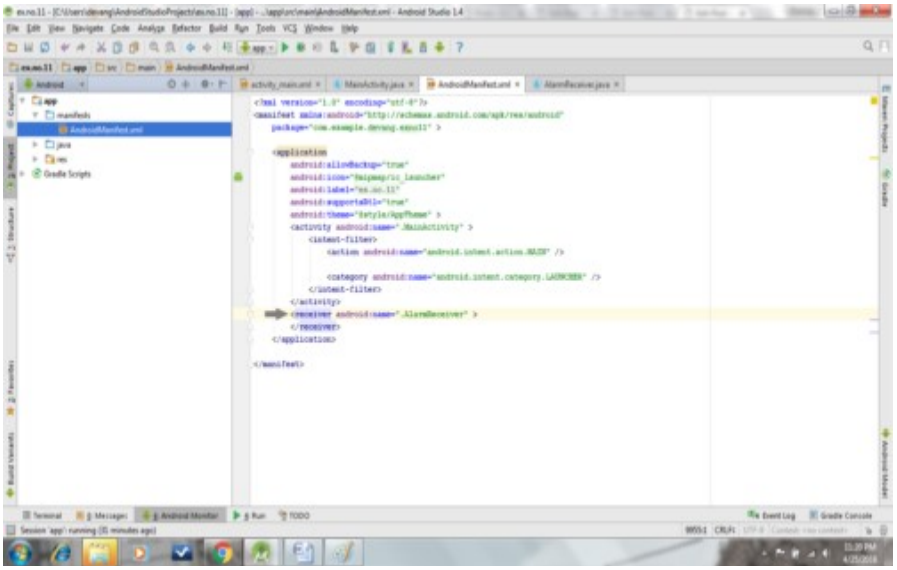


- So now the designing part is completed.

Changes in Manifest for the Android Application:

- Click on **app -> manifests -> AndroidManifest.xml**

- Now change the **activity tag** to **receiver tag** in the AndroidManifest.xml file as shown below



Code for AndroidManifest.xml:

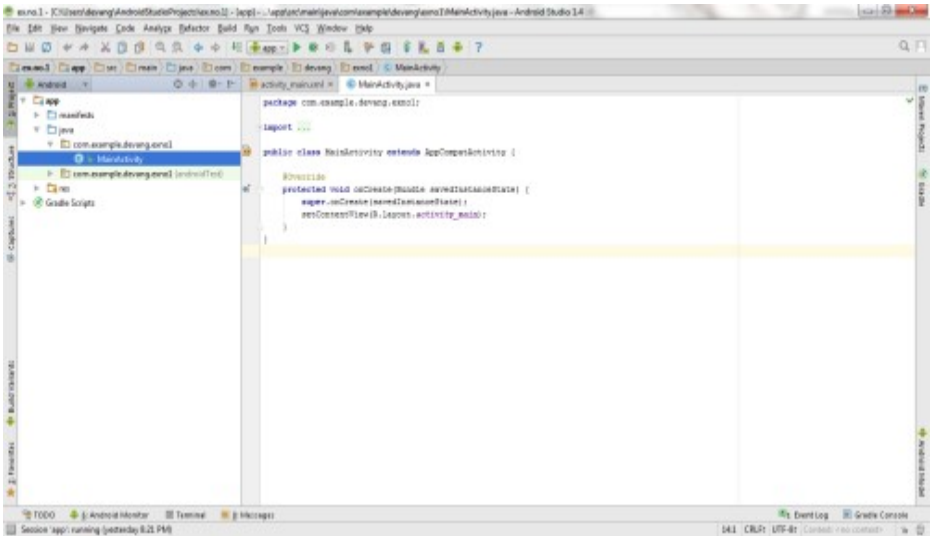
```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
   package="com.example.exno11" >
    <application
      android:allowBackup="true"
      android:icon="@mipmap/ic_launcher"
      android:label="@string/app_name"
      android:supportsRtl="true"
      android:theme="@style/AppTheme" >
      <activity android:name=".MainActivity" >
        <intent-filter>
          <action android:name="android.intent.action.MAIN" />
           <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
      </activity>
      <receiver android:name=".AlarmReceiver" >
      </receiver>
   </application>
</manifest>
```

- So now the changes are done in the Manifest.

Java Coding for the Android Application:

Java Coding for Main Activity:

- Click on **app -> java -> com.example.exno10 -> MainActivity.**



- Then delete the code which is there and type the code as given below.

Code for MainActivity.java:

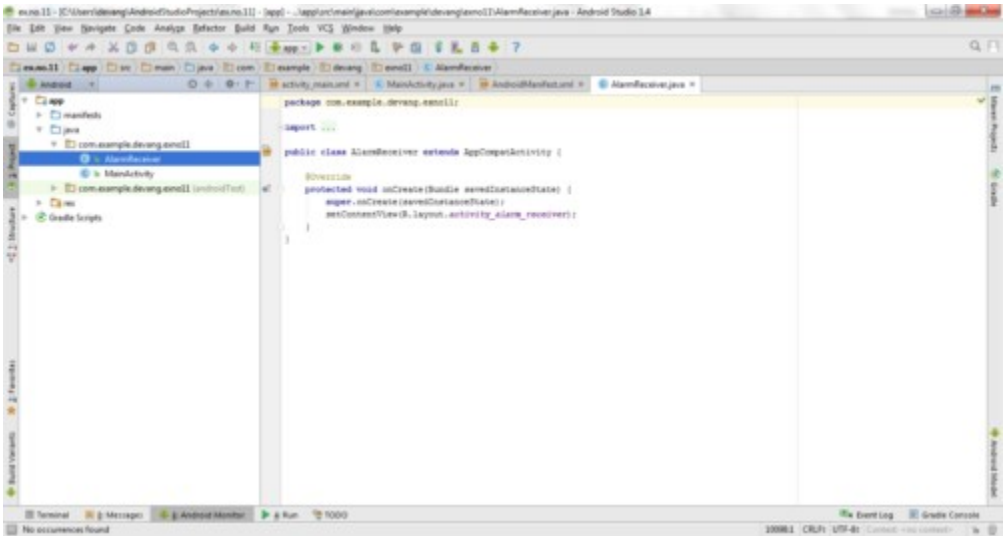
```
package com.example.exno11;

import android.app.AlarmManager;
import android.app.PendingIntent;
import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.TimePicker;
import android.widget.Toast;
import android.widget.ToggleButton;

import java.util.Calendar;

public class MainActivity extends AppCompatActivity
{
   TimePicker alarmTimePicker;
   PendingIntent pendingIntent;
   AlarmManager alarmManager;
```

```
    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        alarmTimePicker = (TimePicker) findViewById(R.id.timePicker);
        alarmManager = (AlarmManager) getSystemService(ALARM_SERVICE);
    }
    public void OnToggleClicked(View view)
    {
        long time;
        if (((ToggleButton) view).isChecked())
        {
            Toast.makeText(MainActivity.this, "ALARM ON", Toast.LENGTH_SHORT).show();
            Calendar calendar = Calendar.getInstance();
            calendar.set(Calendar.HOUR_OF_DAY, alarmTimePicker.getCurrentHour());
            calendar.set(Calendar.MINUTE, alarmTimePicker.getCurrentMinute());
            Intent intent = new Intent(this, AlarmReceiver.class);
            pendingIntent = PendingIntent.getBroadcast(this, 0, intent, 0);

            time=(calendar.getTimeInMillis()-(calendar.getTimeInMillis()%60000));
            if(System.currentTimeMillis()>time)
            {
                if (calendar.AM_PM == 0)
                    time = time + (1000*60*60*12);
                else
                    time = time + (1000*60*60*24);
            }
            alarmManager.setRepeating(AlarmManager.RTC_WAKEUP, time, 10000, pendingIntent);
        }
        else
        {
            alarmManager.cancel(pendingIntent);
            Toast.makeText(MainActivity.this, "ALARM OFF", Toast.LENGTH_SHORT).show();
        }
    }
}
```

- So now the Coding part of Main Activity is completed.

Java Coding for Alarm Receiver:

- Click on **app -> java -> com.example.exno11 -> AlarmReceiver.**

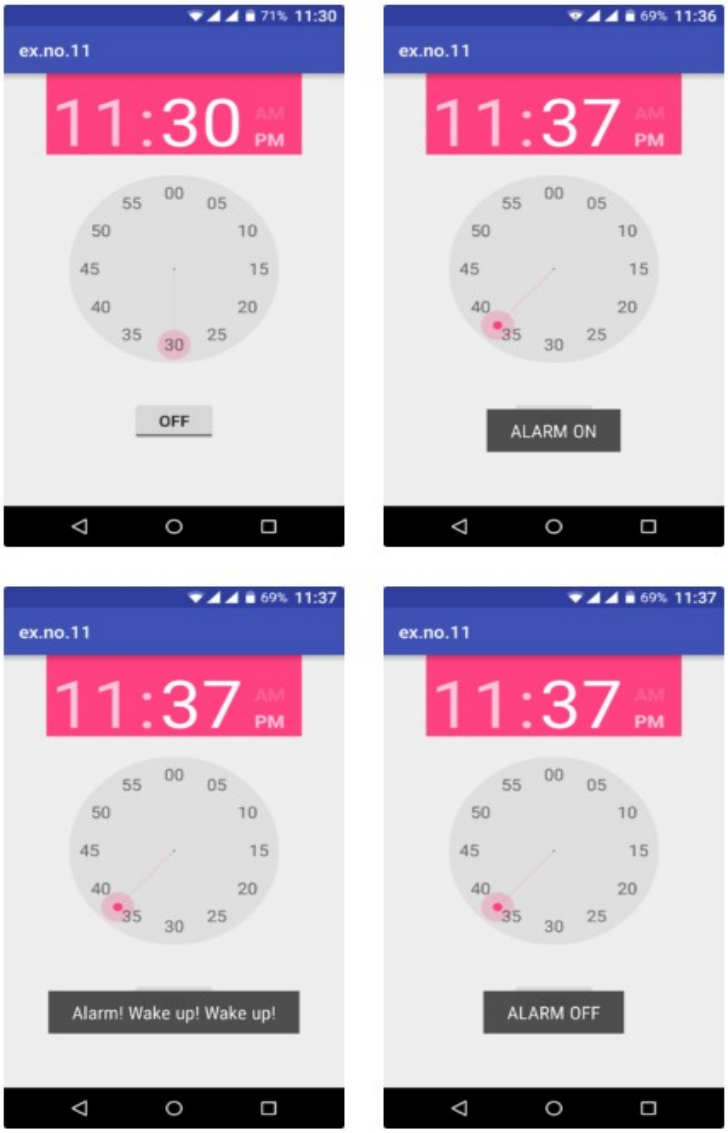


- Then delete the code which is there and type the code as given below.

Code for AlarmReceiver.java:

```
package com.example.exno11;
 import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.media.Ringtone;
import android.media.RingtoneManager;
import android.net.Uri;
import android.widget.Toast;
 public class AlarmReceiver extends BroadcastReceiver
{
   @Override
   public void onReceive(Context context, Intent intent)
   {
      Toast.makeText(context, "Alarm! Wake up! Wake up!", Toast.LENGTH_LONG).show();
      Uri alarmUri = RingtoneManager.getDefaultUri(RingtoneManager.TYPE_ALARM);
      if (alarmUri == null)
      {
         alarmUri = RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION);
      }
      Ringtone ringtone = RingtoneManager.getRingtone(context, alarmUri);
      ringtone.play();
   }
}
```

Output:



Result:

Thus Android Application that creates Alarm Clock is developed and executed successfully.