K.S.R. COLLEGE OF ENGINEERING (Autonomous): TIRUCHENGODE - 637 215

Vision of the Institution

- We envision to achieve status as an excellent educational institution in the global knowledge hub, making self-learners, experts, ethical and responsible engineers, technologists, scientists, managers, administrators and entrepreneurs who will significantly contribute to research and environment friendly sustainable growth of the nation and the world.

Mission of the Institution

- To inculcate in the students self-learning abilities that enable them to become competitive and considerate engineers, technologists, scientists, managers, administrators and entrepreneurs by diligently imparting the best of education, nurturing environmental and social needs.
- To foster and maintain a mutually beneficial partnership with global industries and Institutions through knowledge sharing, collaborative research and innovation.

DEPARTMENTOF COMPUTER SCIENCE AND ENGINEERING

Vision of the Department

- To create ever green professionals for software industry, academicians for knowledge cultivation and researchers for contemporary society modernization.

Mission of the Department

- To produce proficient design, code and system engineers for software development.
- To keep updated contemporary technology and fore coming challenges for welfare of the society.

Programme Educational Objectives (PEOs)

- Figure out, formulate, analyze typical problems and develop effective solutions by imparting the idea and principles of science, mathematics, engineering fundamentals and computing.
- Competent professionally and successful in their chosen career through life-long learning.
- Excel individually or as member of a team in carrying out projects and exhibit social needs and follow professional ethics.

Programme Outcomes (POs) –COMPUTER SCIENCE AND ENIGNEERING

PO1	**Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
PO2	**Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
PO3	**Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
PO4	**Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
PO5	**Modern tool usage:** Create, select, and apply appropriate techniques, resource, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
PO6	**The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
PO7	Environmental and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
PO8	**Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
PO9	**Individual and team work:** Function effectively as an individual, and as a member or leader diverse teams, and in multidisciplinary settings.
PO10	Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO11	**Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO12	**Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadcast context of technological change.
PSO1	**Technical competency:** Develop and Implement computer solutions that accomplish goals to the industry, government or research by exploring new technologies.
PSO2	**Professional awareness:** Grow intellectually and professionally in the chosen field.

Course Objectives:

- To know the components and structure of mobile application development frameworks for Android OS based mobiles.
- To understand how to work with various mobile application development frameworks.
- To learn the basic and important design concepts and issues of development of mobile applications.
- To understand the capabilities and limitations of mobile devices.

Course Outcomes: On Completion of this Course, the student will be able to

CO-1: Design and Implement various mobile applications using emulators.

CO-2: Develop mobile application using graphical primitives.

CO-3: Design android mobile apps using native data handling.

CO-4: Develop mobile apps using background tasks and notifications.

CO-5: Deploy applications to hand-held devices.

Course articulation matrix:

16CS723- Mobile Application Development Laboratory															
CO	Course Outcomes	Programme Outcomes													
		PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PS O1	PSO2
C407.1	Design and Implement various mobile applications using emulators.	3	3	2	1	-	2	-	-	-	3	-	-	3	2
C407.2	Develop mobile application using graphical primitives.	-		3	3	3	2	-	-	-	3	-	-	3	2
C407.3	Design android mobile apps using native data handling.	3	-	3	3	3	2	-		-	3	-	-	3	2
C407.4	Develop mobile apps using background tasks and notifications.	-	-	3	3	-	2	-	2	-	2	-	-	3	2
C407.5	Deploy applications to hand-held devices.	3	-	2	1	-	2	1	-	2	3	-	3	3	2
Vertical Average Value		**3**	**3**	**3**	**3**	**3**	**2**	**1**	**2**	**2**	**3**	**-**	**3**	**3**	**2**

16CS723	MOBILE APPLICATION DEVELOPMENT LABORATORY	L	T	P	C
		0	0	3	2

Prerequisite: *Basic knowledge about programming languages and fundamental concepts of JAVA(16CS002)*

Objectives:

- *To know the components and structure of mobile application development frameworks for Android OS based mobiles.*
- To understand how to work with various mobile application development frameworks.
- *To learn the basic and important design concepts and issues of development of mobile applications.*
- To understand the capabilities and limitations of mobile devices.

List of Experiments:

1. Develop an application that uses GUI components, Font and Colors
2. Develop an application that uses Layout Managers and event listeners.
3. Develop a native calculator application.
4. Write an application that draws basic graphical primitives on the screen.
5. Develop an application that makes use of database.
6. Develop an application that makes use of RSS Feed.
7. Implement an application that implements Multi-threading
8. Develop a native application that uses GPS location information.
9. Implement an application that writes data to the SD card.
10. Implement an application that creates an alert upon receiving a message.
11. Write a mobile application that creates alarm clock

Total: 45 Periods

Course Outcomes: On completion of this course, the student should be able to

- Design and Implement various mobile applications using emulators.
- Develop mobile application using graphical primitives.
- *Design android mobile apps using native data handling.*
- Develop mobile apps using background tasks and notifications.
- *Deploy applications to hand-held devices.*

LIST OF EXPERIMENTS

Ex.No: 1	DEVELOP AN APPLICATION THAT USES GUI COMPONENTS, FONT AND COLORS
Date:	

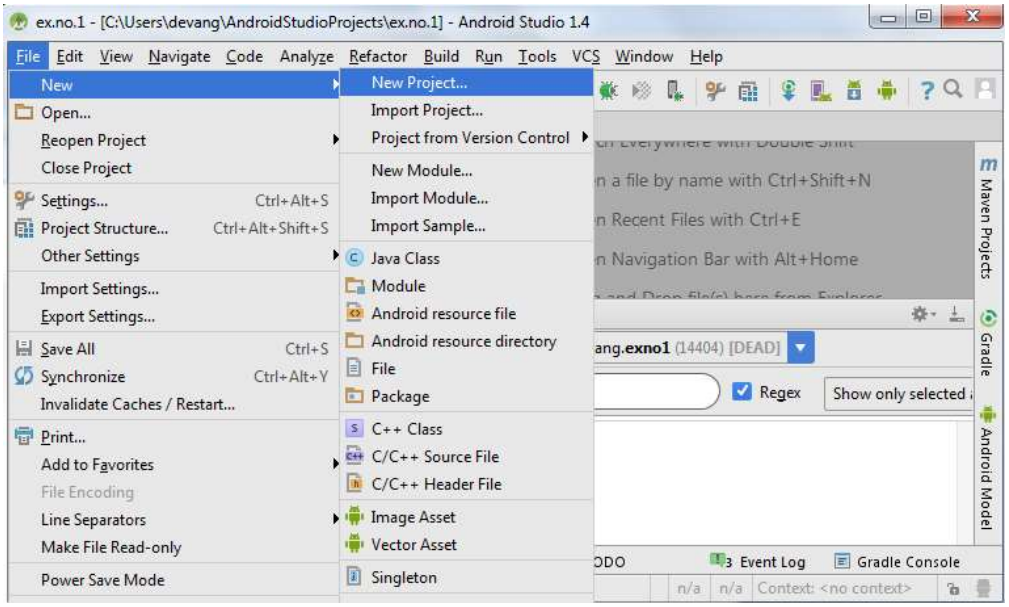
Aim:

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

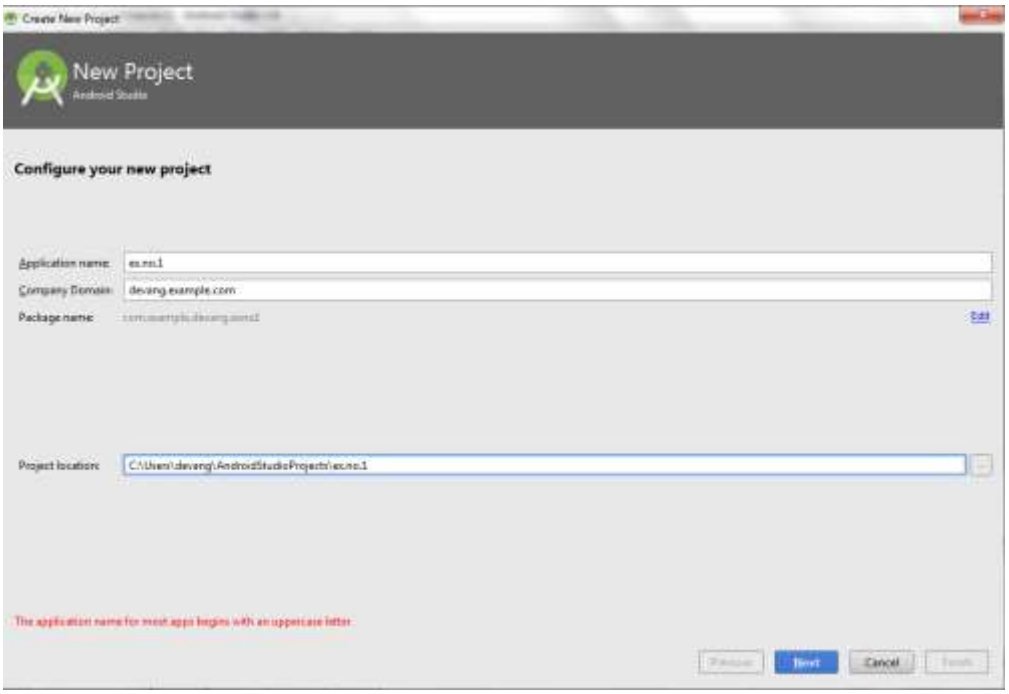
Procedure:

Creating a New project:

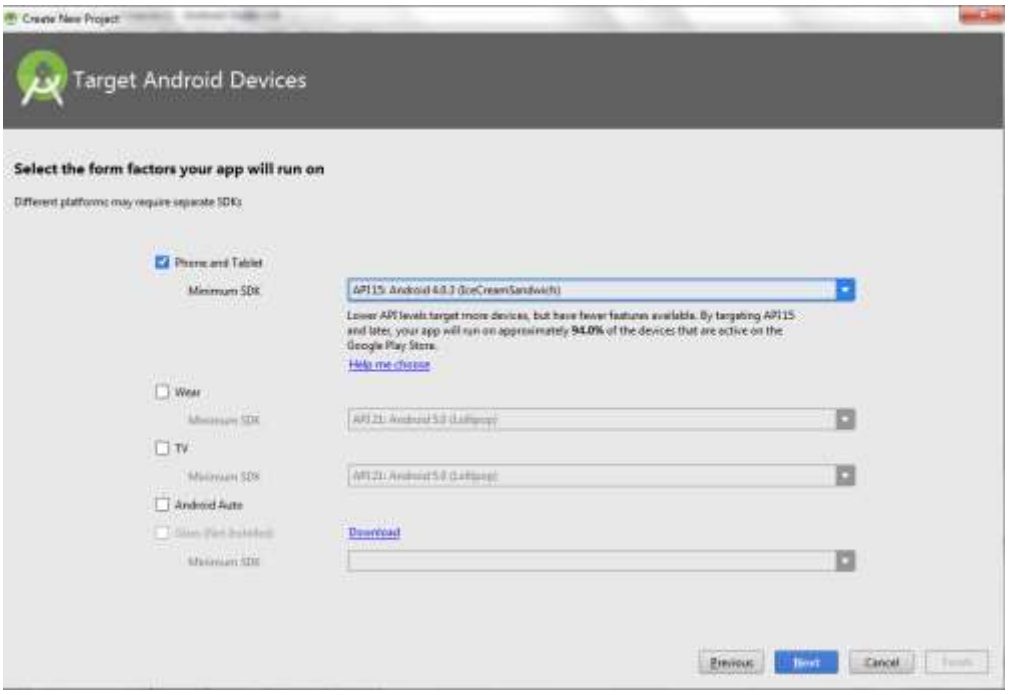
1. Open Android Studio and then click on **File -> New -> New project.**



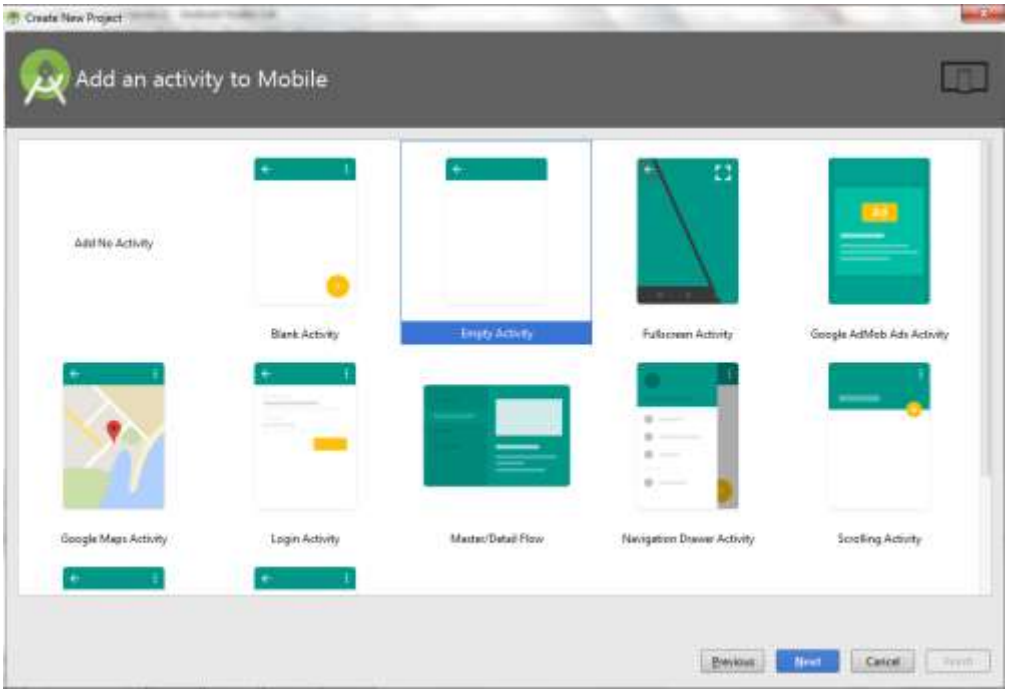
2. Then type the Application name as "**ex.no.1"** and click **Next.**



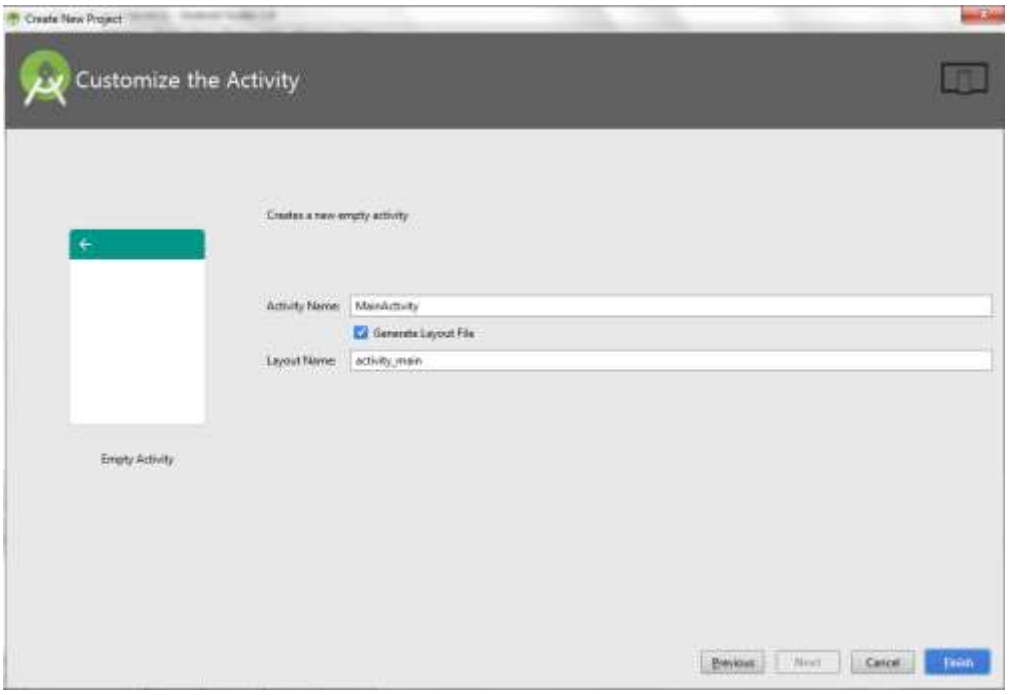
3. Then select the **Minimum SDK** as shown below and click **Next**.



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

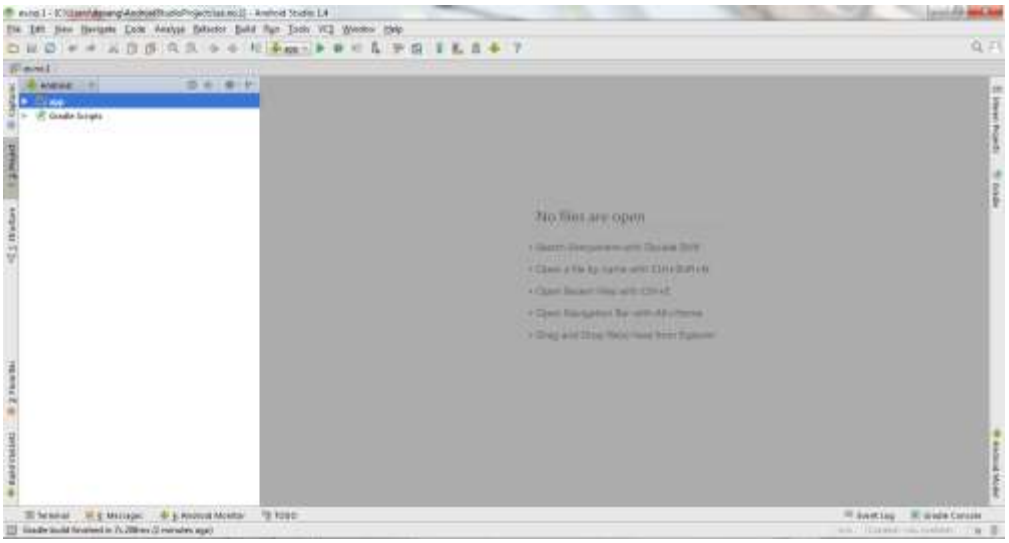


5. Finally click F**inish**.



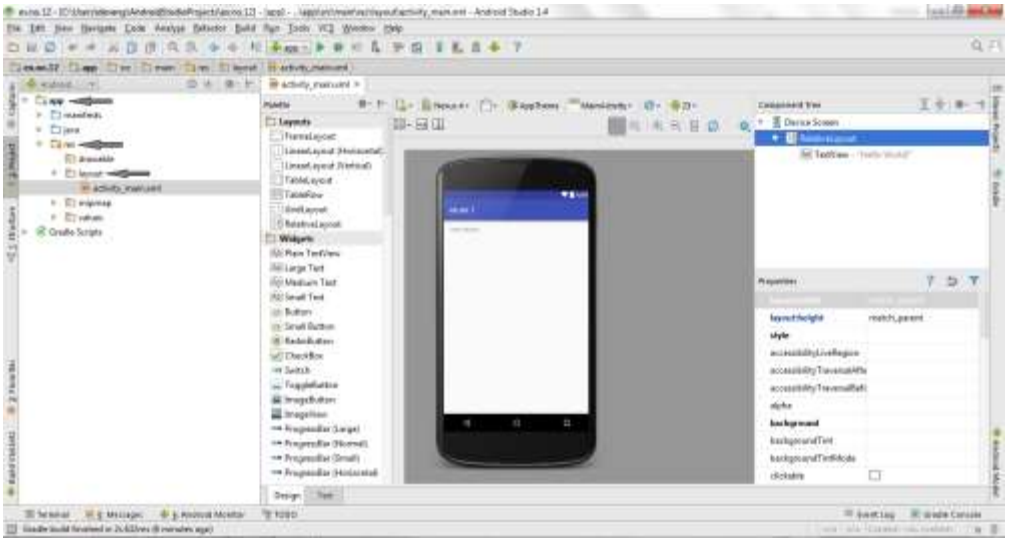
6. It will take some time to build and load the project.

7. After completion it will look as given below.

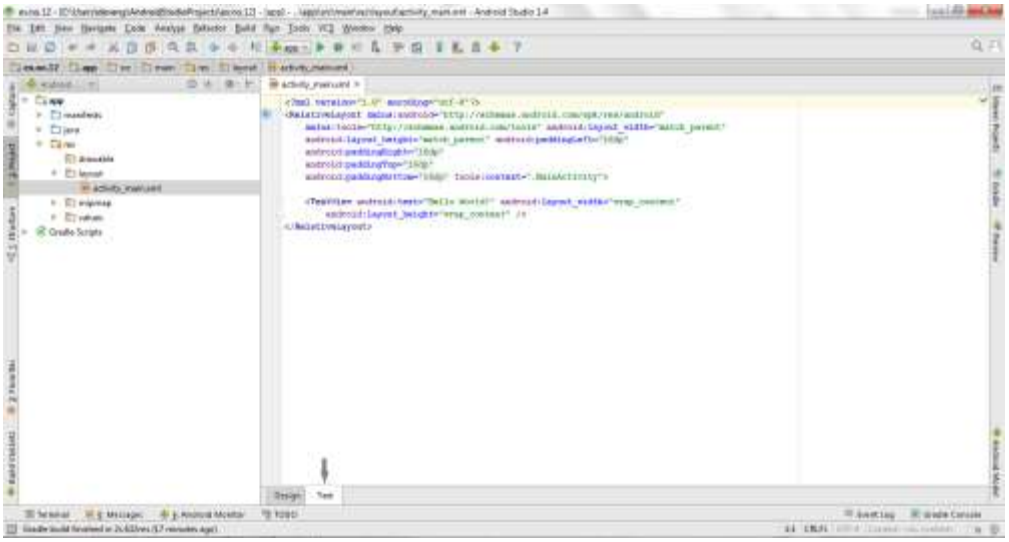


Designing layout for the Android Application:

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



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

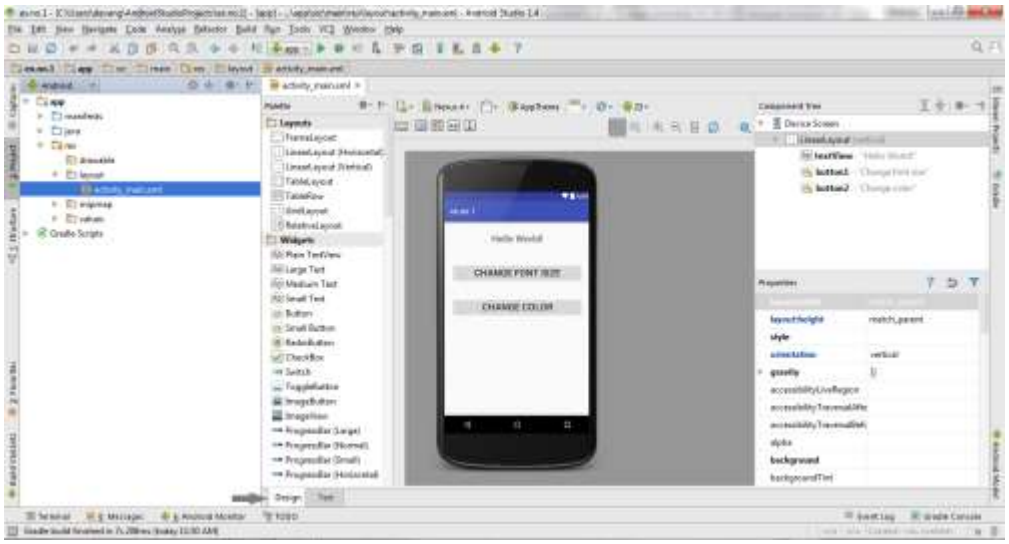


10. Then delete the code which is there and type the code as given below.<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:orientation="vertical"
android:layout_width="match_parent"
android:layout_height="match_parent">

<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>
```

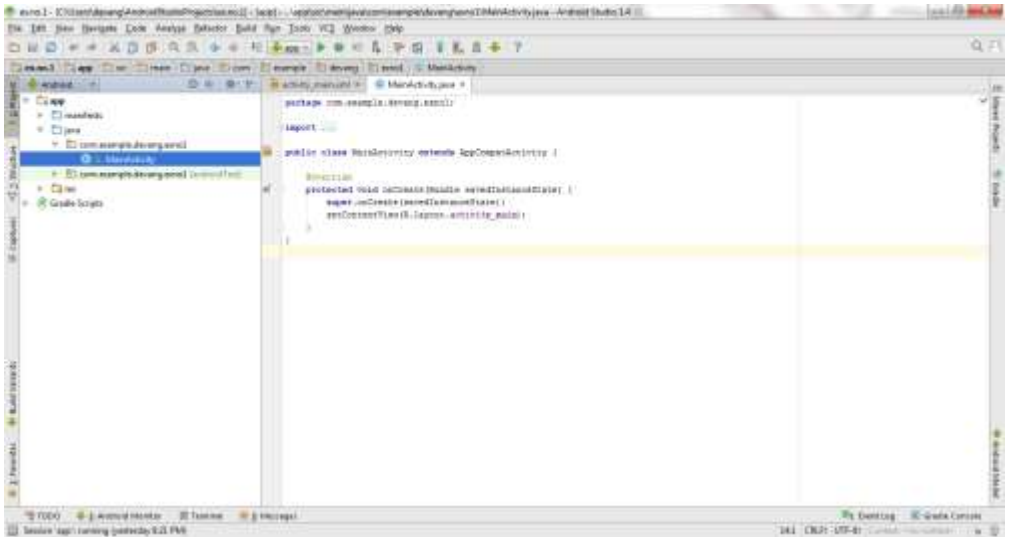
11. Now click on Design and your application will look as given below.



12. So now the designing part is completed.

Java Coding for the Android Application:

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

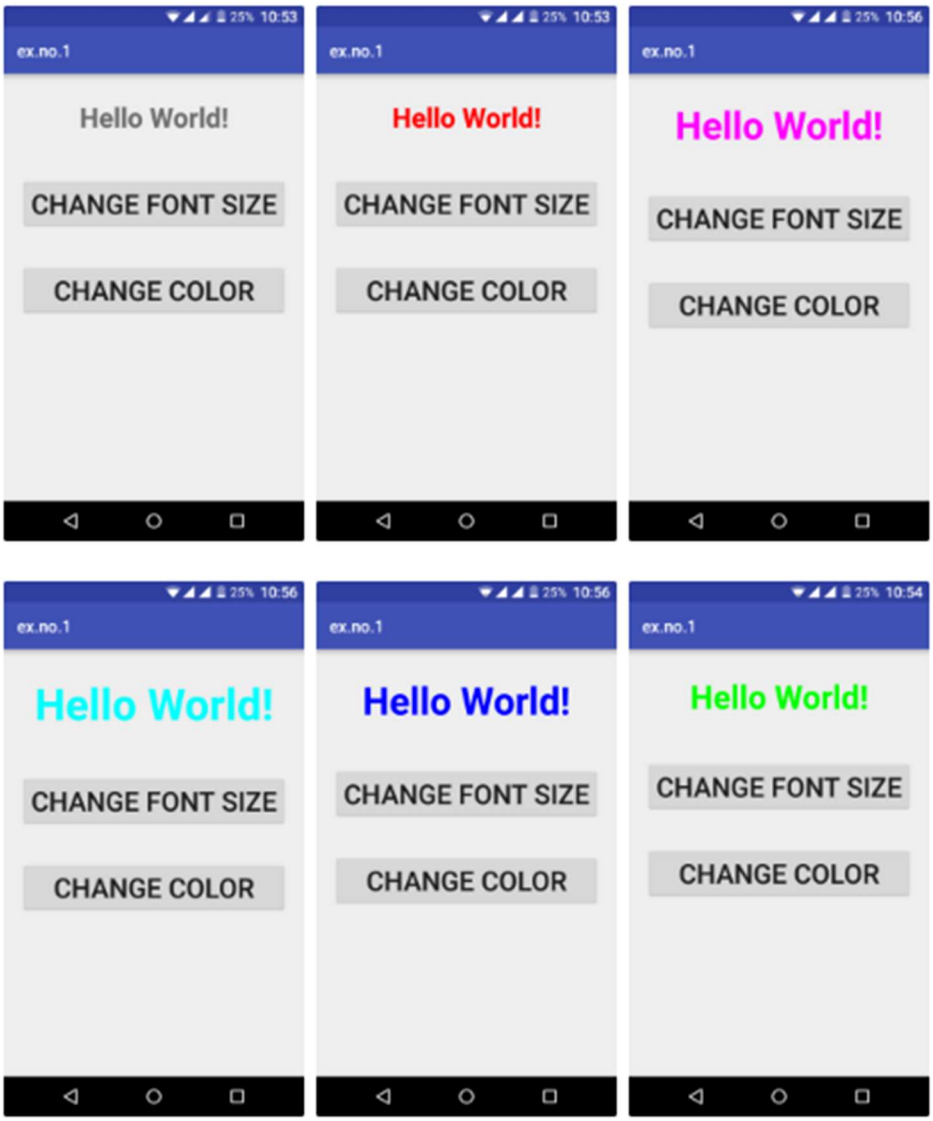


14. Then delete the code which is there and type the code as given below.

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;
}
});
```

Output:



```
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;
}
});
}
}
```

15. So now the Coding part is also completed.

16. Now run the application to see the output.

Observation	25	
Record	10	
Total	35	
Signature		

Result:

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

Ex.No: 2	DEVELOP AN APPLICATION THAT USES LAYOUT MANAGERS AND EVENT LISTENERS.
Date:	

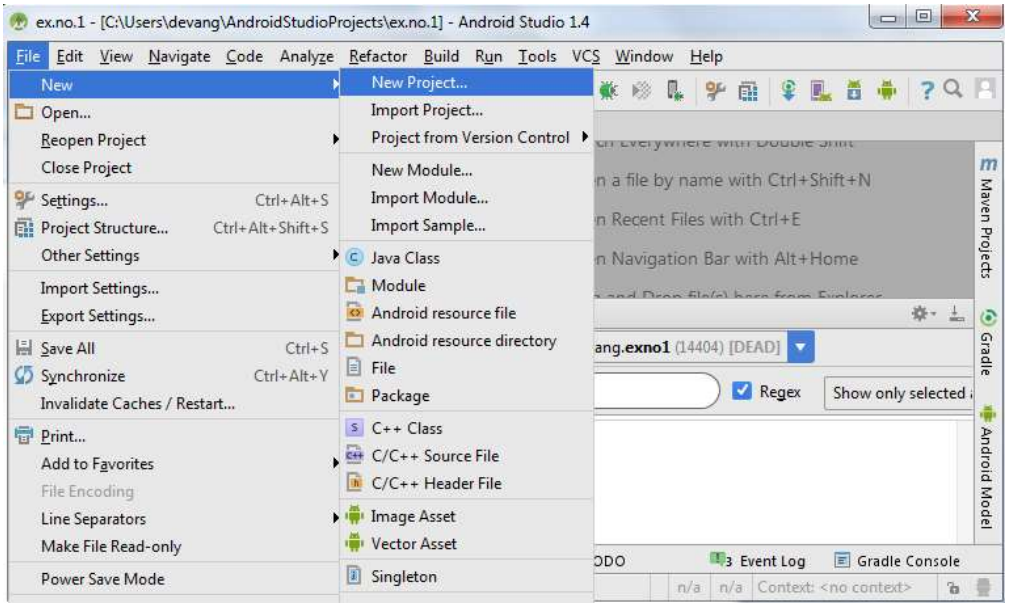
Aim:

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

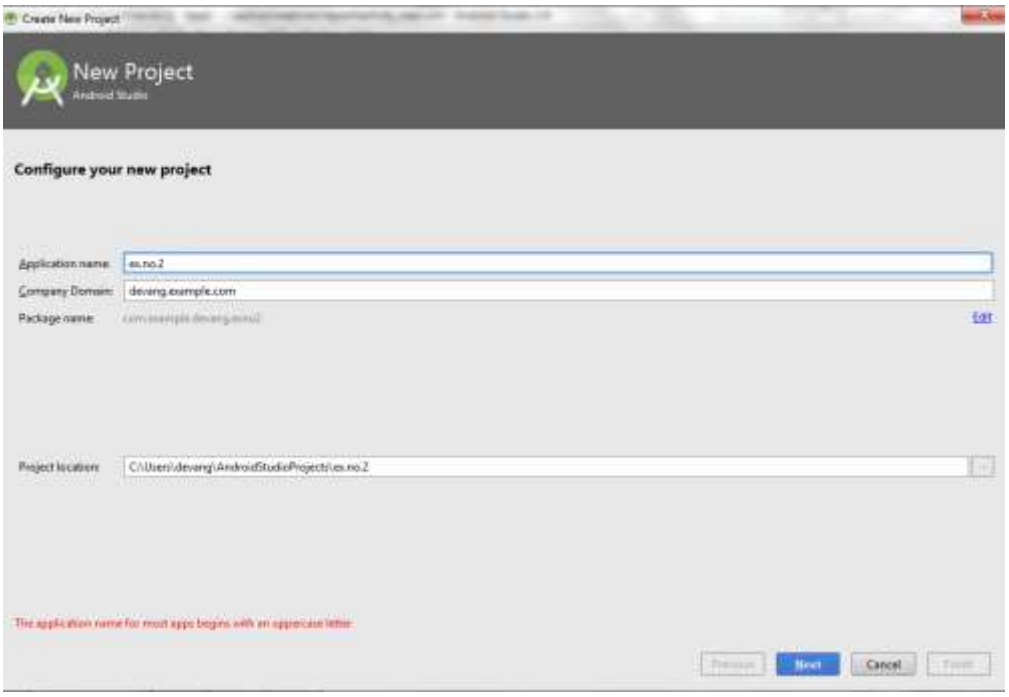
Procedure:

Creating a New project:

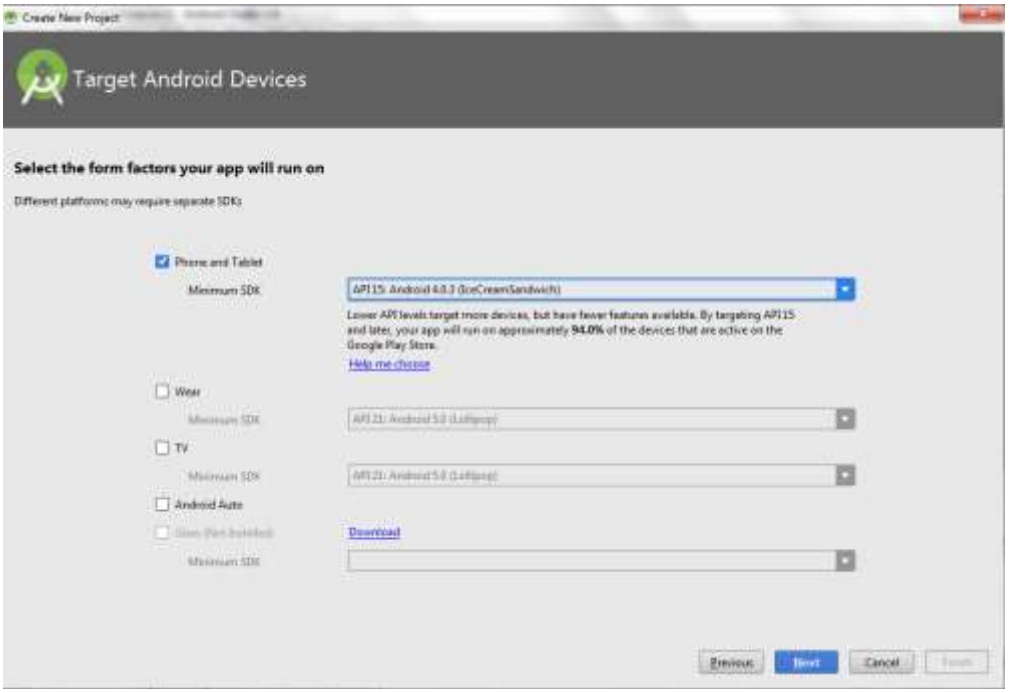
1. Open Android Stdio and then click on **File -> New -> New project.**



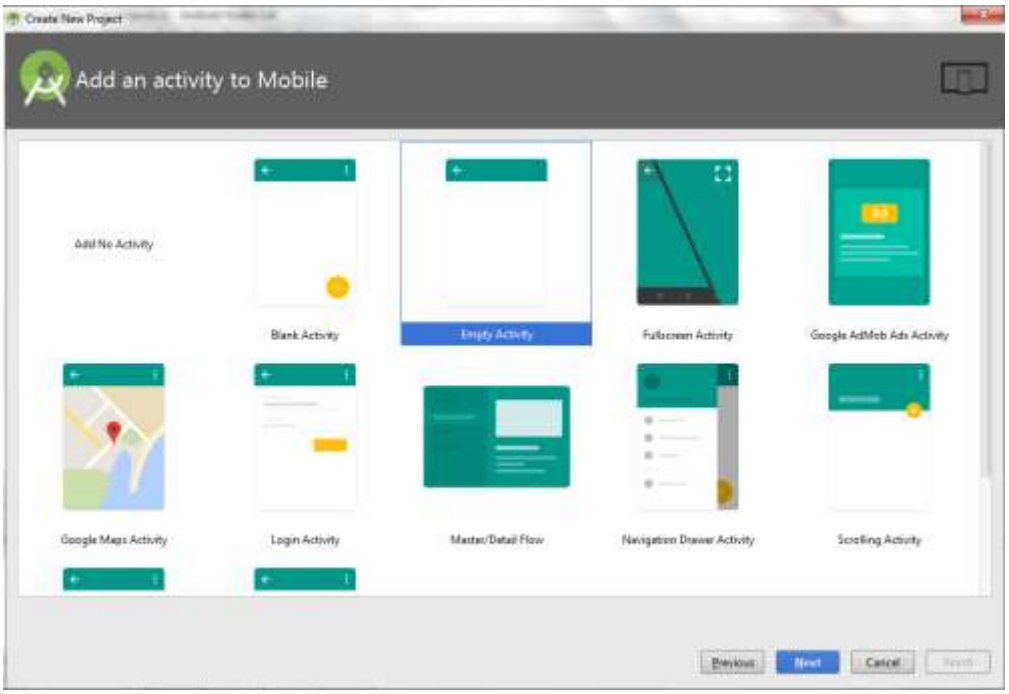
2. Then type the Application name as "**ex.no.2**" and click **Next.**



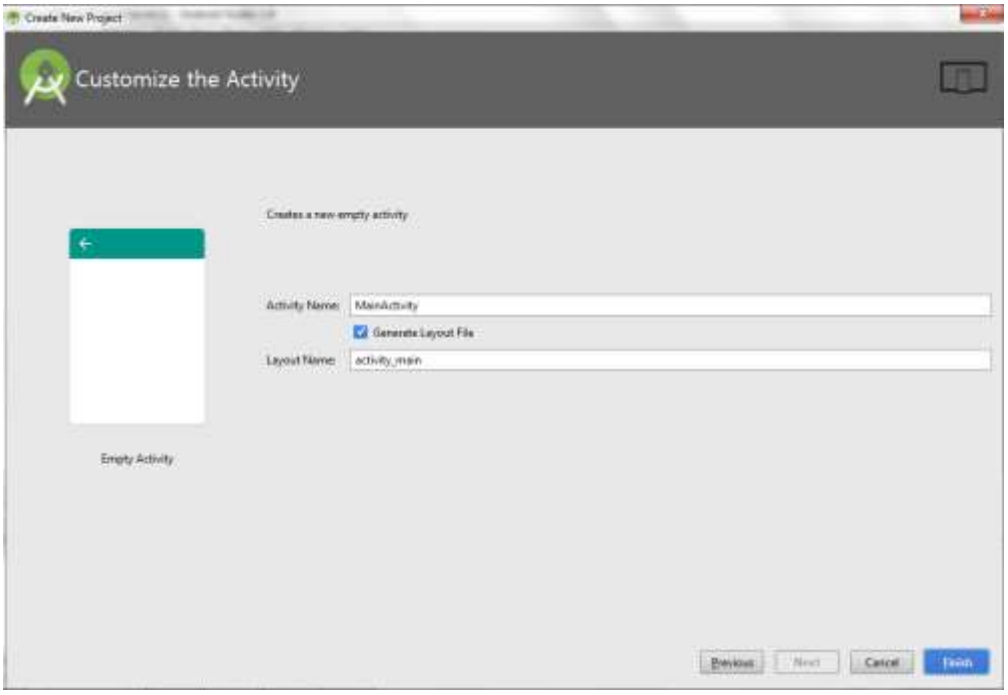
3. Then select the **Minimum SDK** as shown below and click **Next**.



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

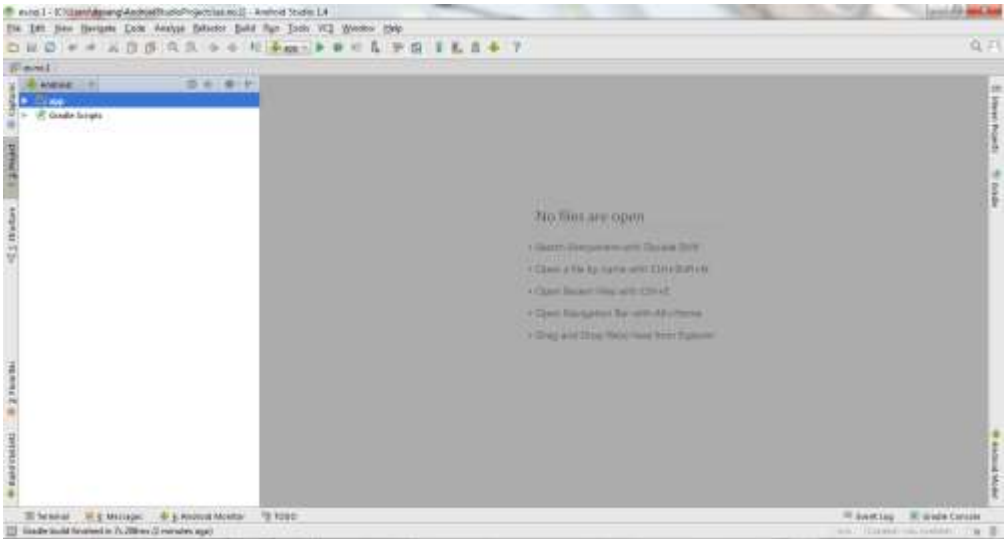


5. Finally click F**inish**.



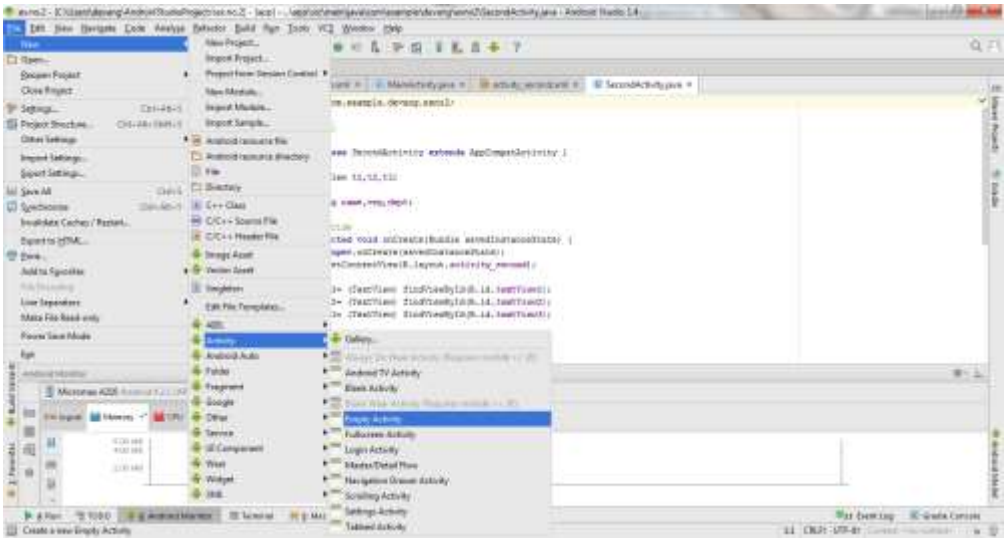
6. It will take some time to build and load the project.

7. After completion it will look as given below.

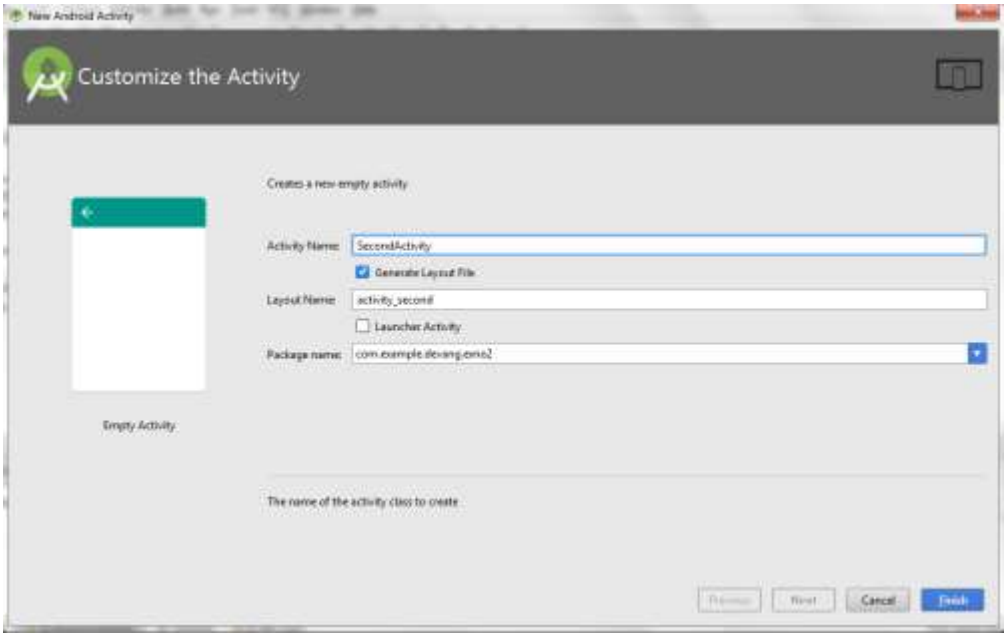


Creating Second Activity for the Android Application:

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



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

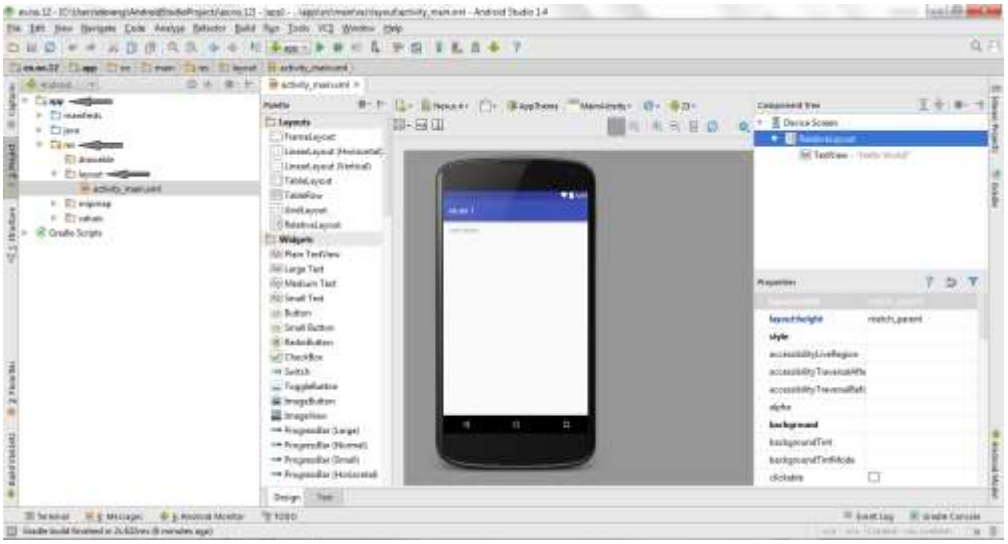


10. Thus Second Activity For the application is created.

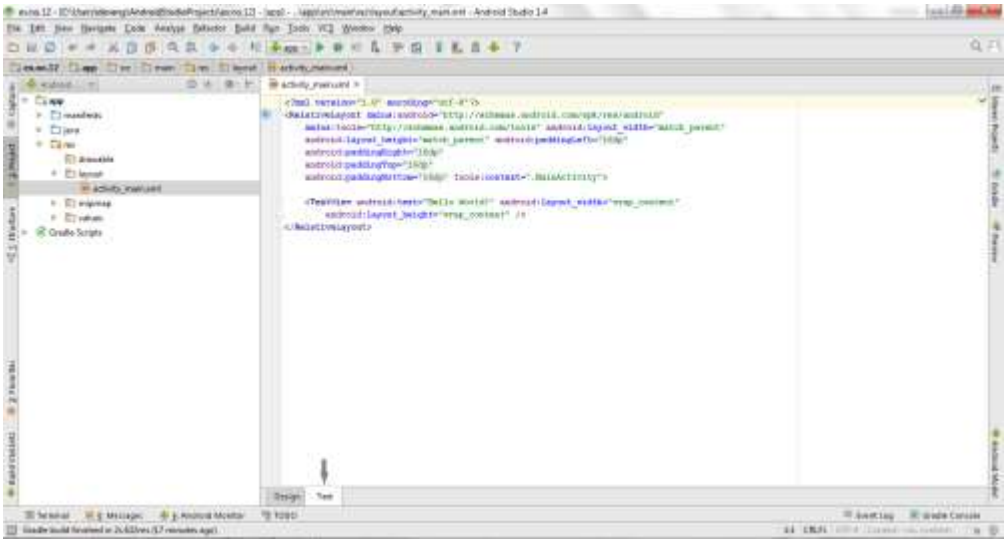
Designing layout for the Android Application:

Designing Layout for Main Activity:

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



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



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

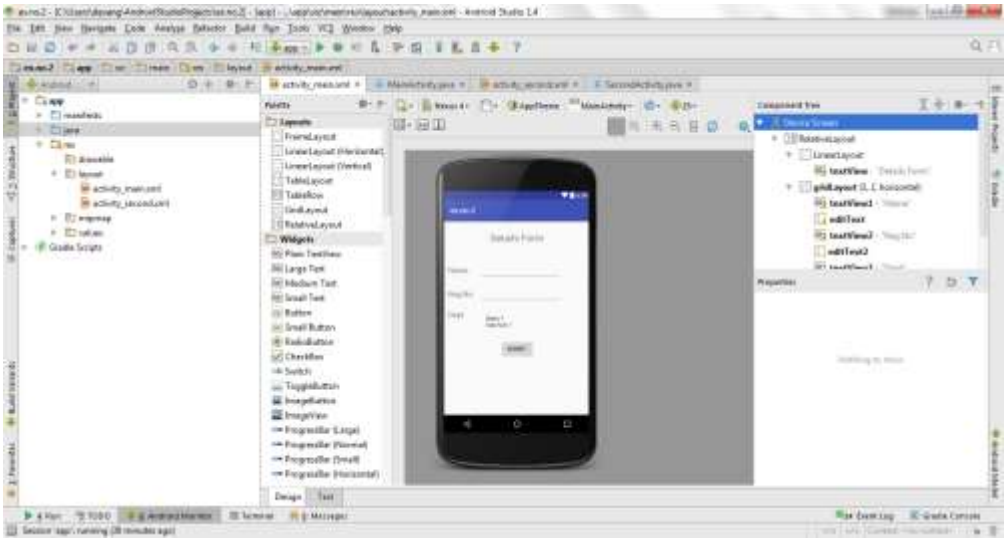
Code for Activity_main.xml:

```
<?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>
```

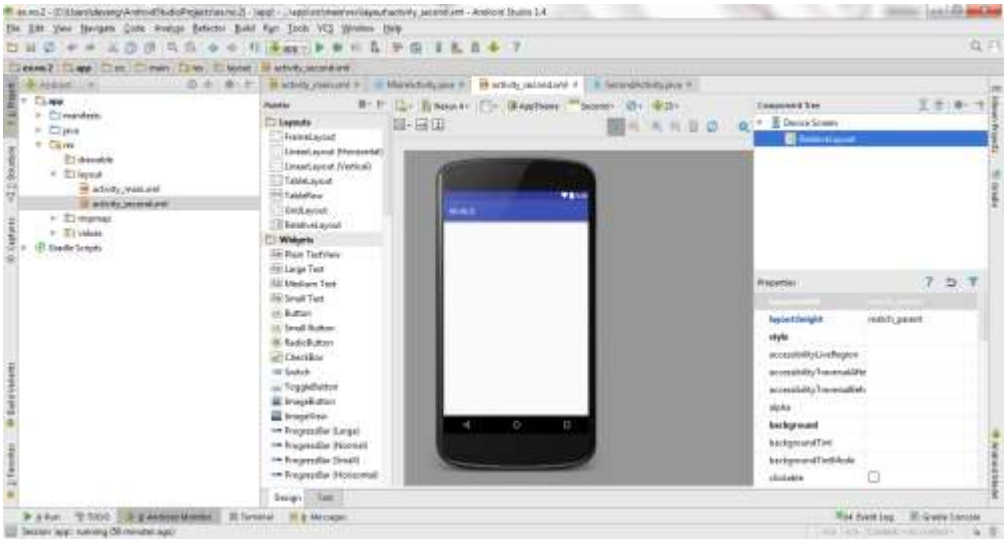
14. Now click on Design and your activity will look as given below.



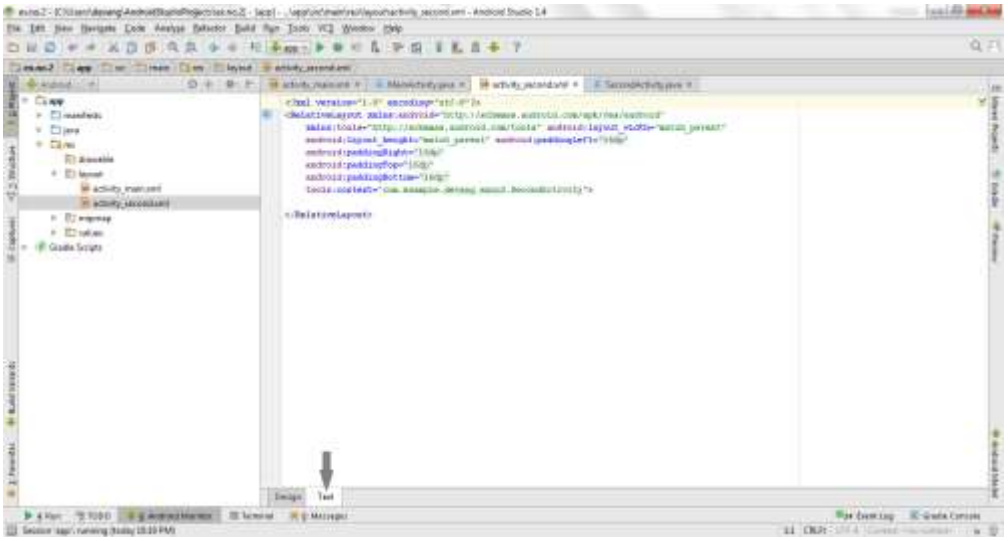
15. So now the designing part of Main Activity is completed.

Designing Layout for Second Activity:

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



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



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

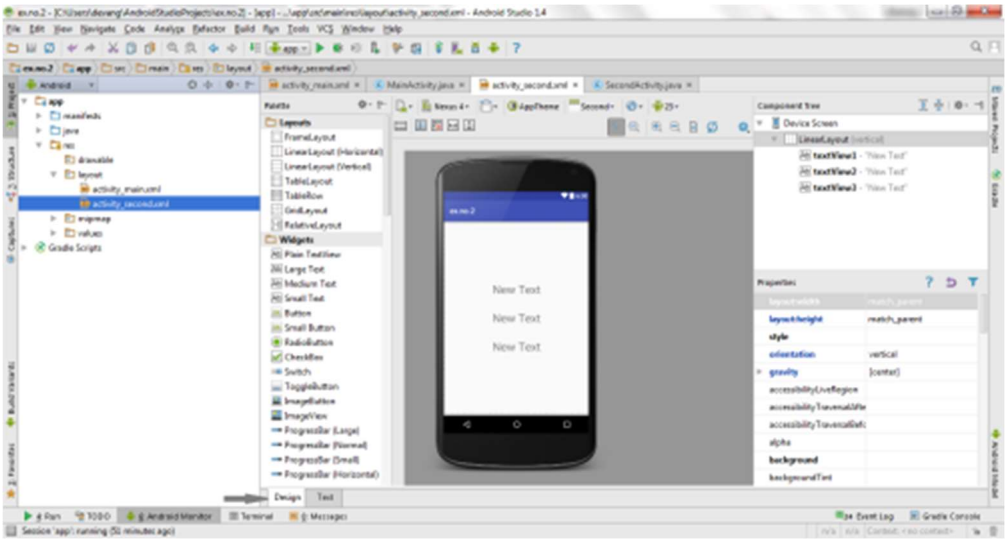
Code for Activity_second.xml:

```
<?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>

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

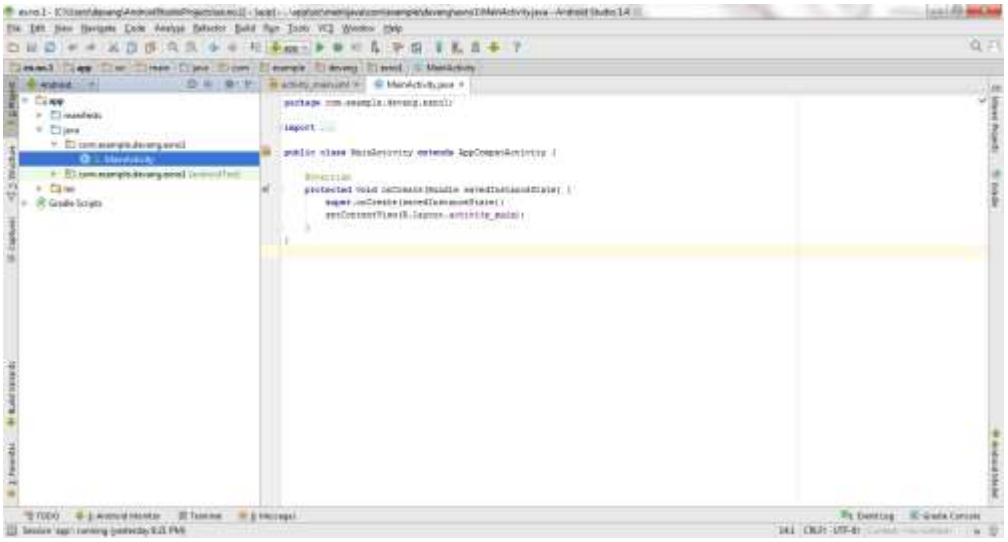


20. So now the designing part of Second Activity is also completed.

Java Coding for the Android Application:

Java Coding for Main Activity:

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



22. 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;
```

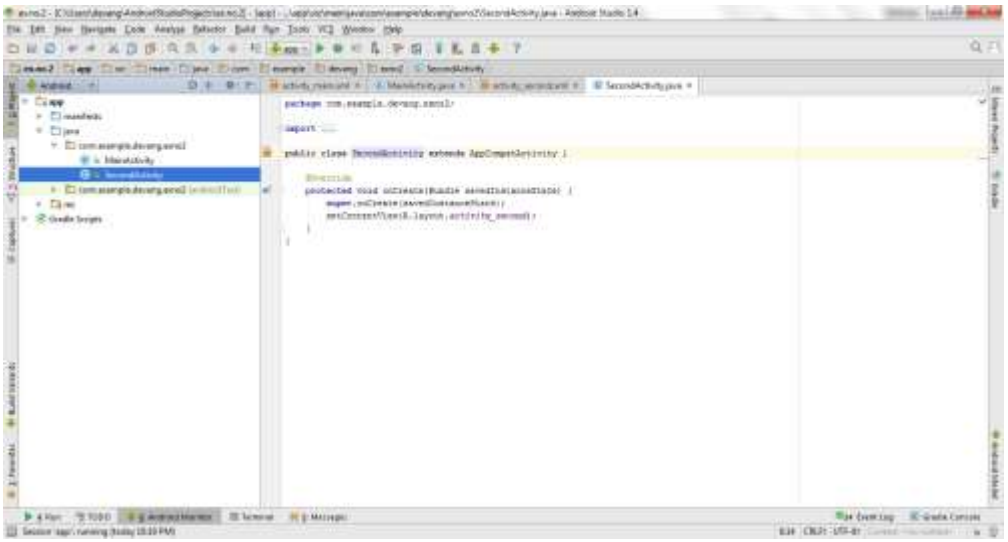
```
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Spinner;

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);

}
});
}
}
```

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

Java Coding for Second Activity:

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

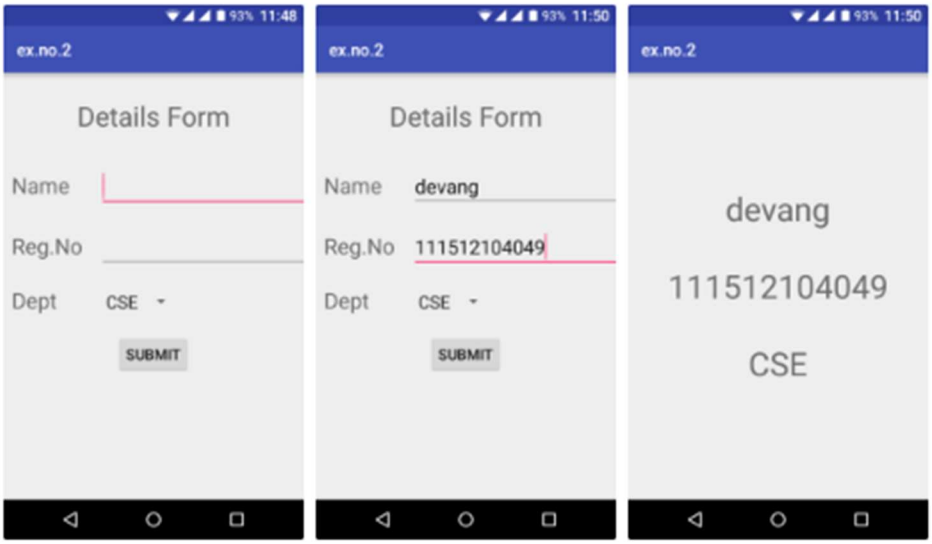


25. Then delete the code which is there and type the code as given below.

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;
```

Output:


93% 11:48
ex.no.2
Details Form
Name
Reg.No
Dept
CSE
SUBMIT
93% 11:50
ex.no.2
Details Form
Name
devang
Reg.No
111512104049
Dept
CSE
SUBMIT
93% 11:50
ex.no.2
devang
111512104049
CSE

```
@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);

}
}
```

26. So now the Coding part of Second Activity is also completed.

27. Now run the application to see the output.

Observation	25	
Record	10	
Total	35	
Signature		

Result:

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

EX.NO: 3 Date:	DEVELOP A NATIVE CALCULATOR APPLICATION.

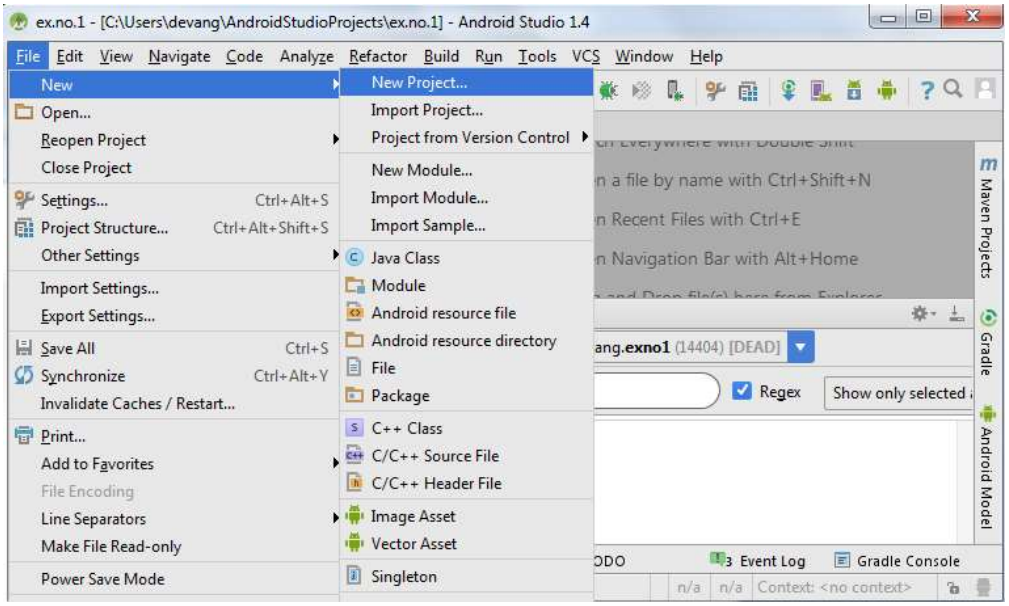
Aim:

To develop a Simple Android Application for Native Calculator.

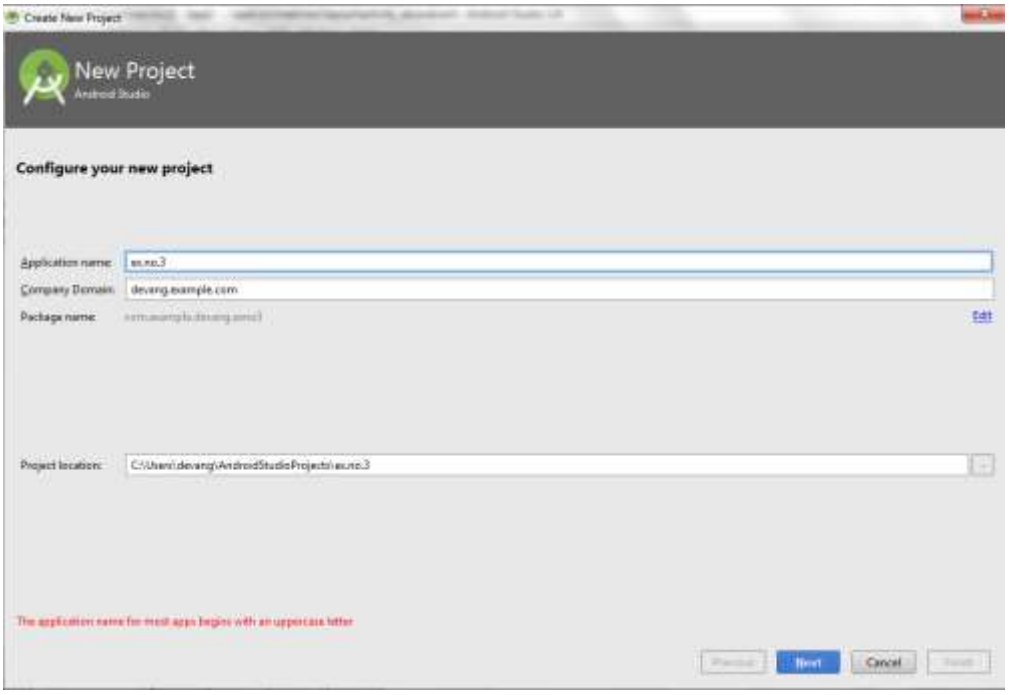
Procedure:

Creating a New project:

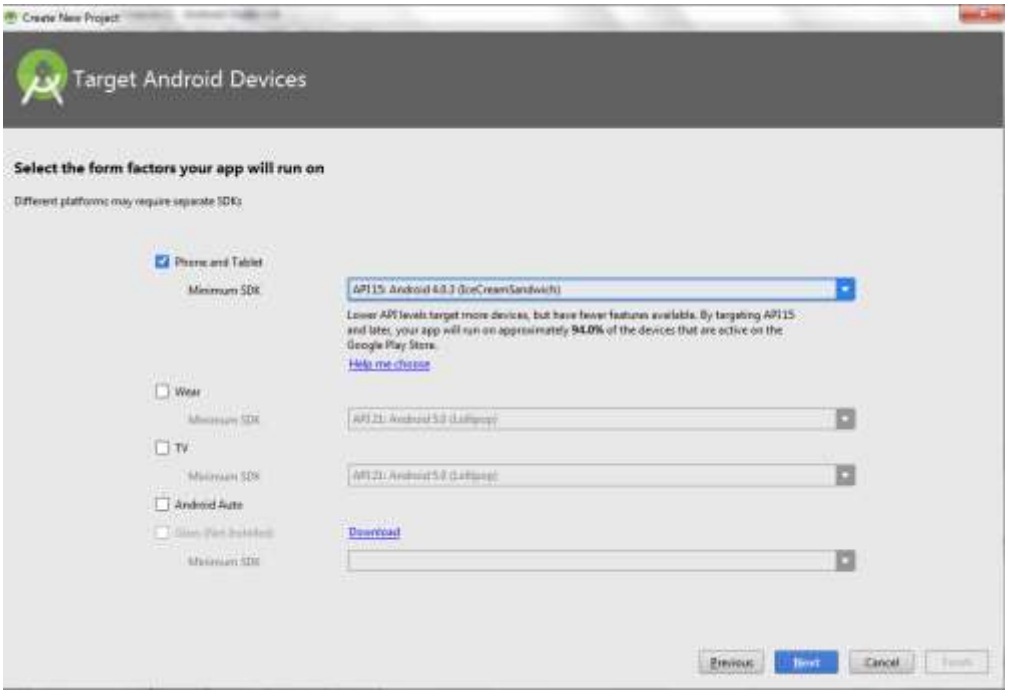
1. Open Android Stdio and then click on **File -> New -> New project.**



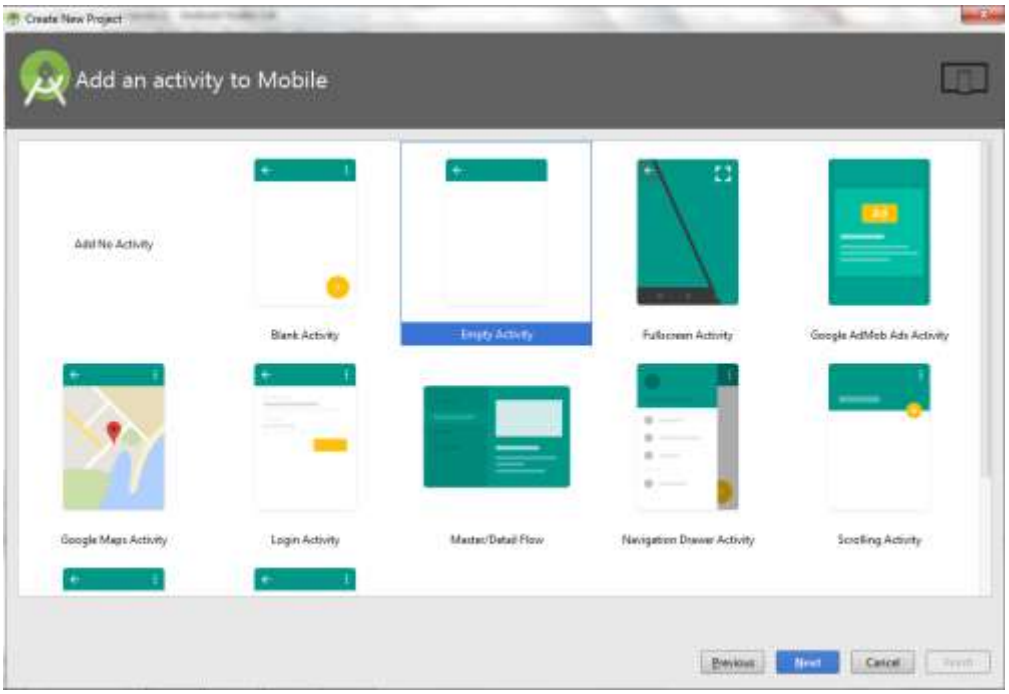
2. Then type the Application name as "**ex.no.3"** and click **Next.**



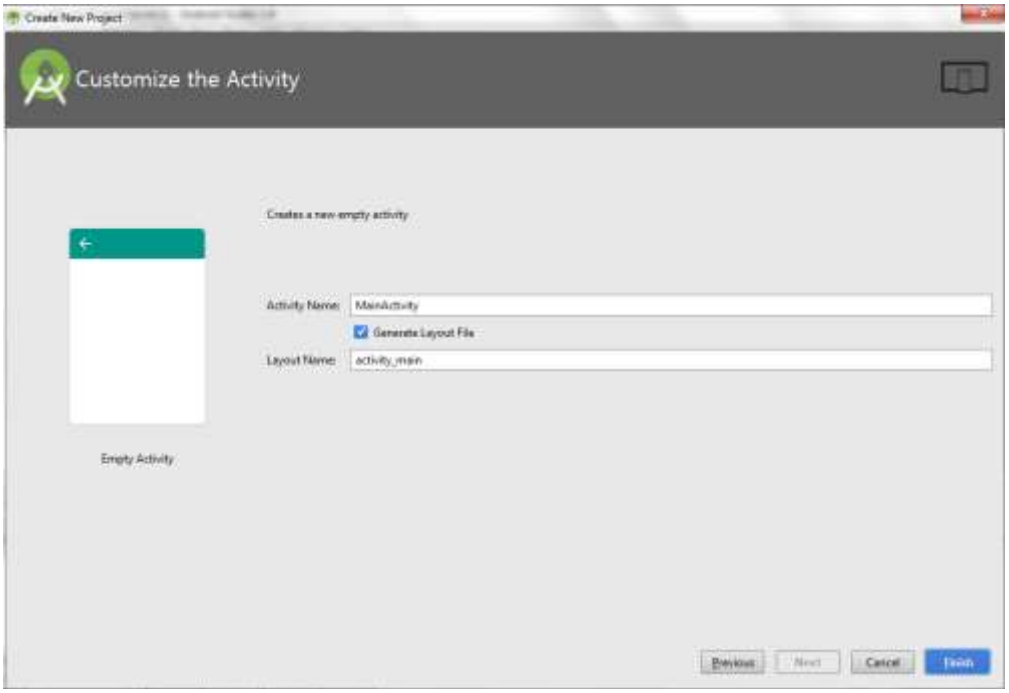
3. Then select the **Minimum SDK** as shown below and click **Next**.



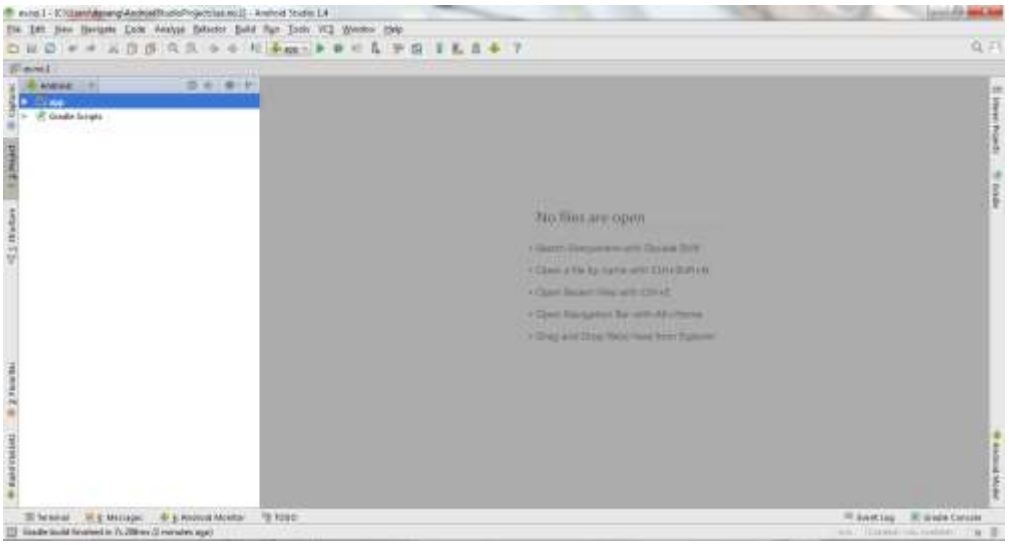
4. Then select the **Empty Activity** and click **Next.**



5. Finally click F**inish**.

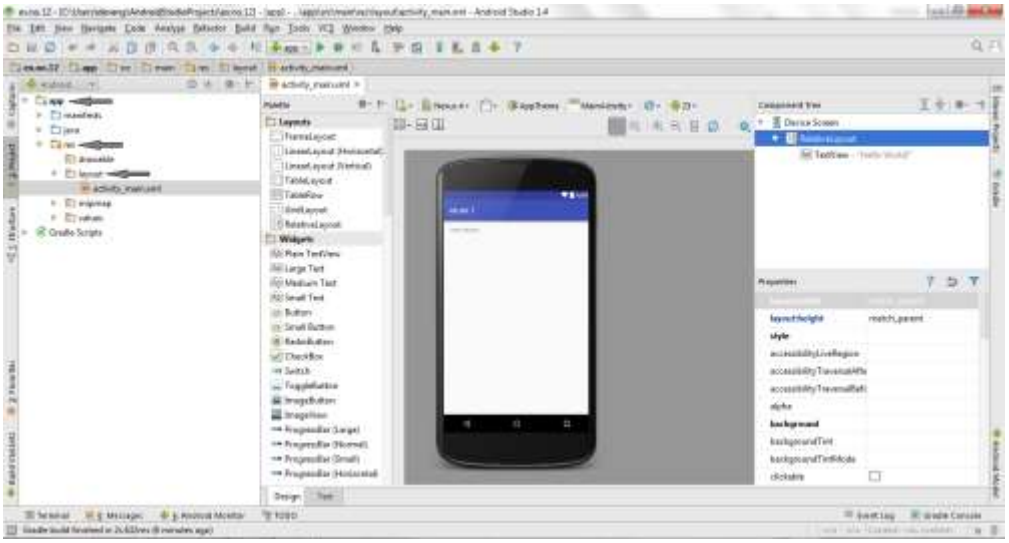


6. It will take some time to build and load the project.
7. After completion it will look as given below.

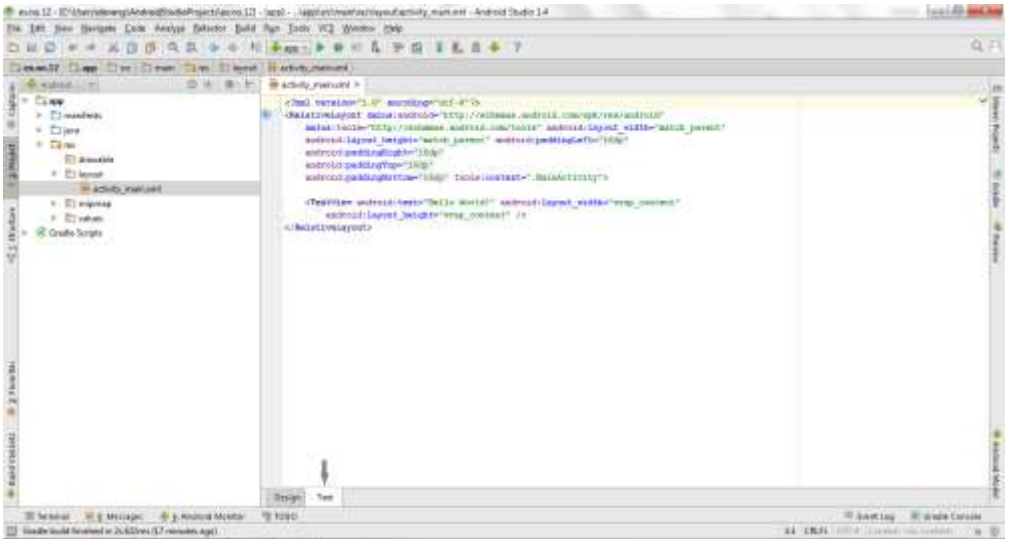


Designing layout for the Android Application:

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



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



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

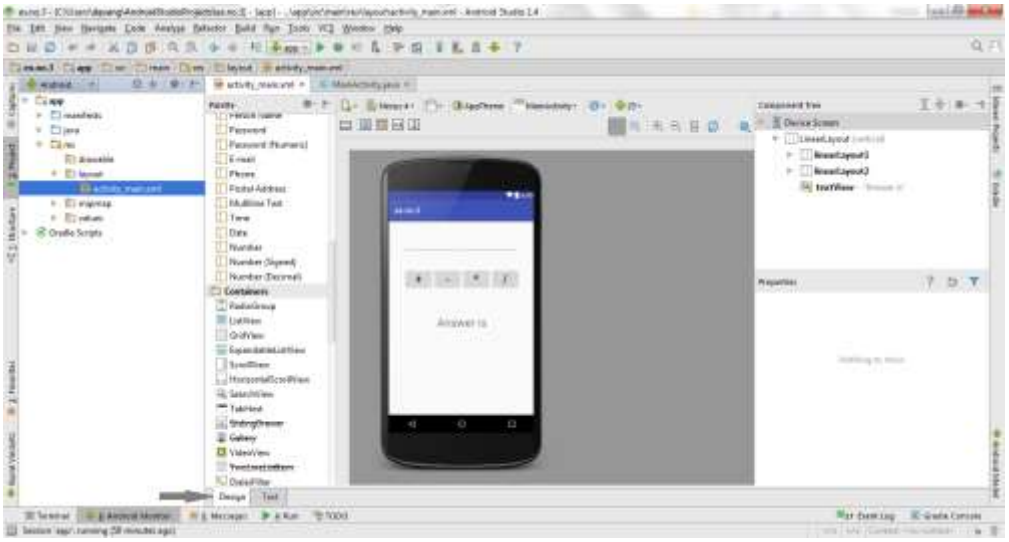
Code for Activity_main.xml:

```
<?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>
```

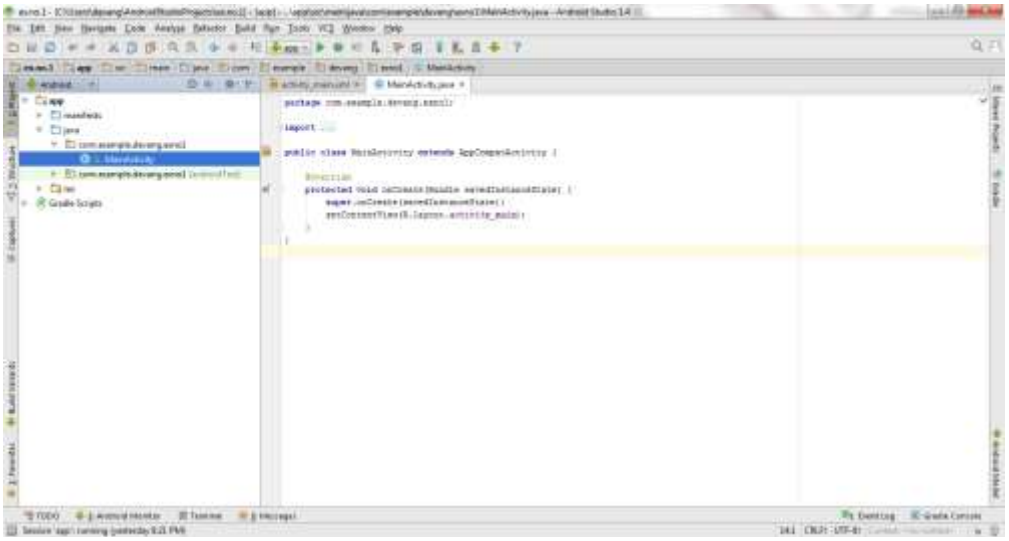
11. Now click on Design and your application will look as given below.



12. So now the designing part is completed.

Java Coding for the Android Application:

13. Click on **app -> java -> com.example.exno3 -> MainActivity.**



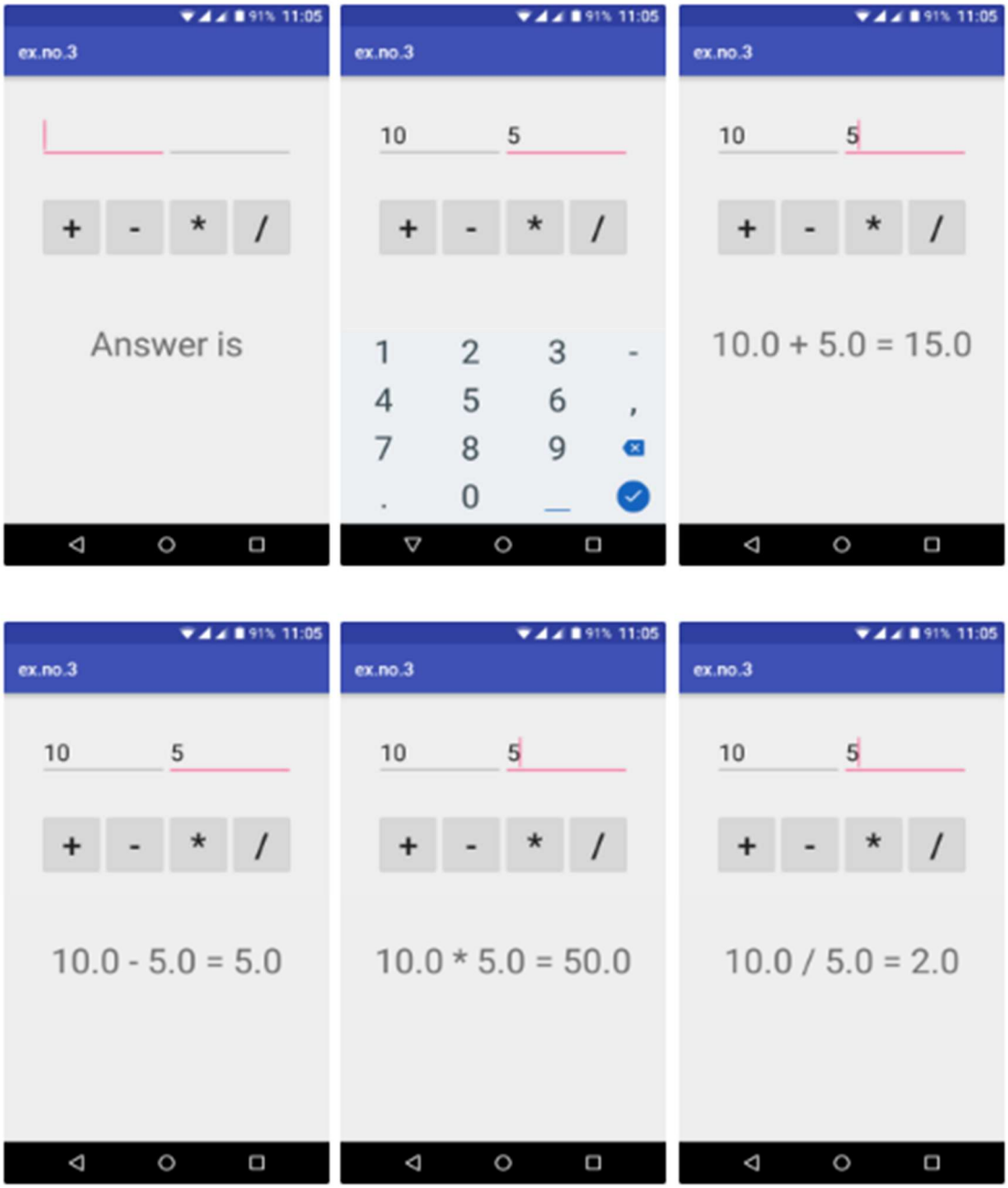
14. Then delete the code which is there and type the code as given below.

Code for MainActivity.java:

```
package com.example.devang.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 = "/";
```

Output:



```
result = num1 / num2;
break;
default:
break;
}
// form the output line
Result.setText(num1 + " " + oper + " " + num2 + " = " + result);
}
}
```

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

Observation	25	
Record	10	
Total	35	
Signature		

Result:

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

Ex.No: 4	WRITE AN APPLICATION THAT DRAWS BASIC GRAPHICAL PRIMITIVES ON THE SCREEN.
Date:	

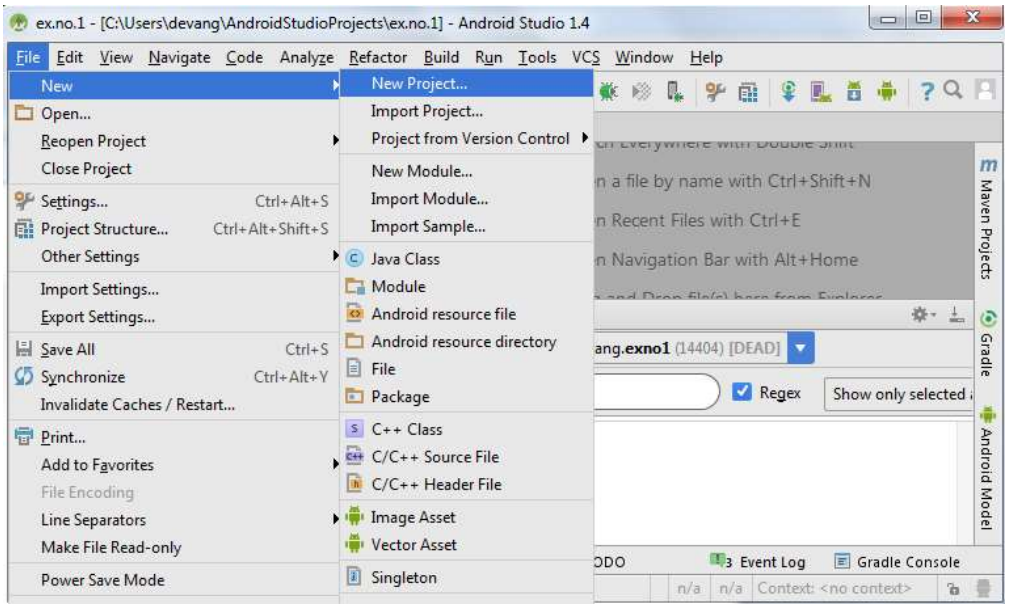
Aim:

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

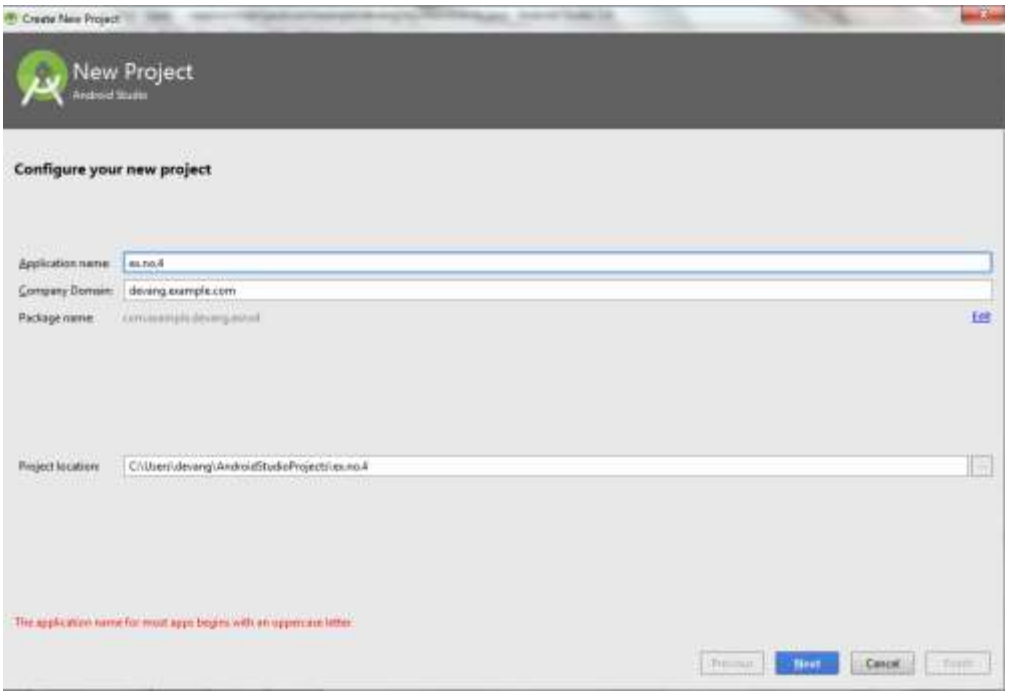
Procedure:

Creating a New project:

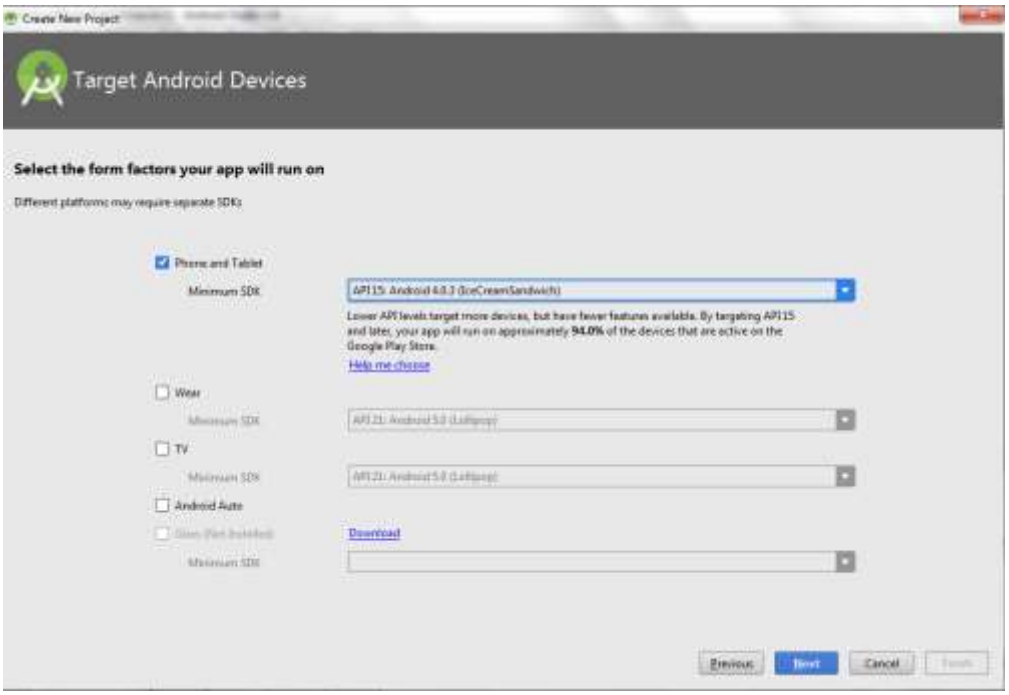
1. Open Android Studio and then click on **File -> New -> New project.**



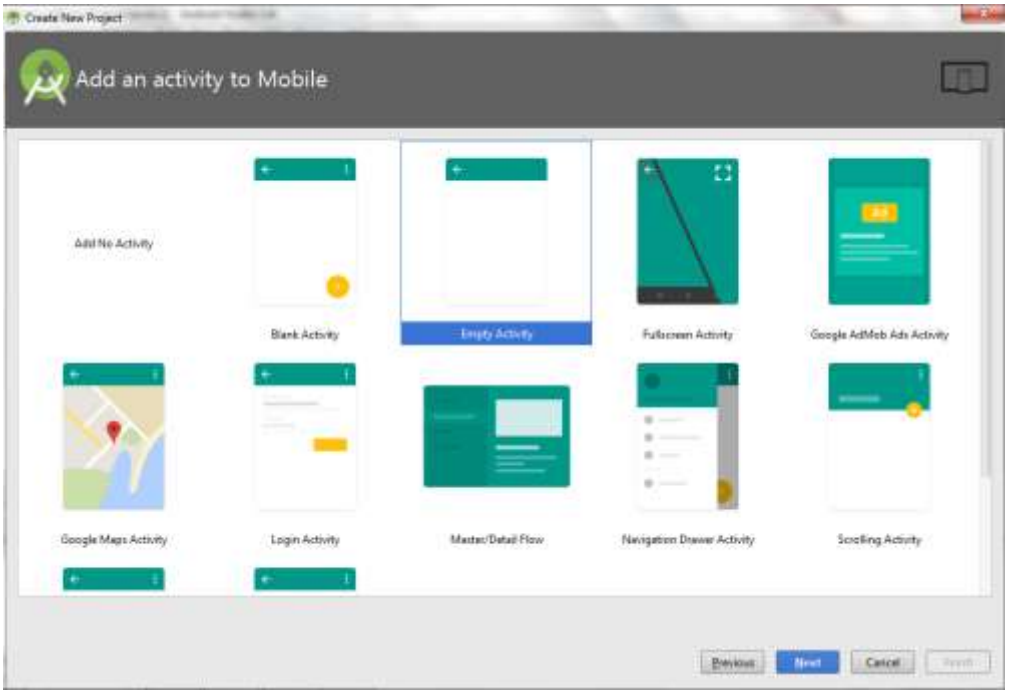
2. Then type the Application name as "**ex.no.4"** and click **Next.**



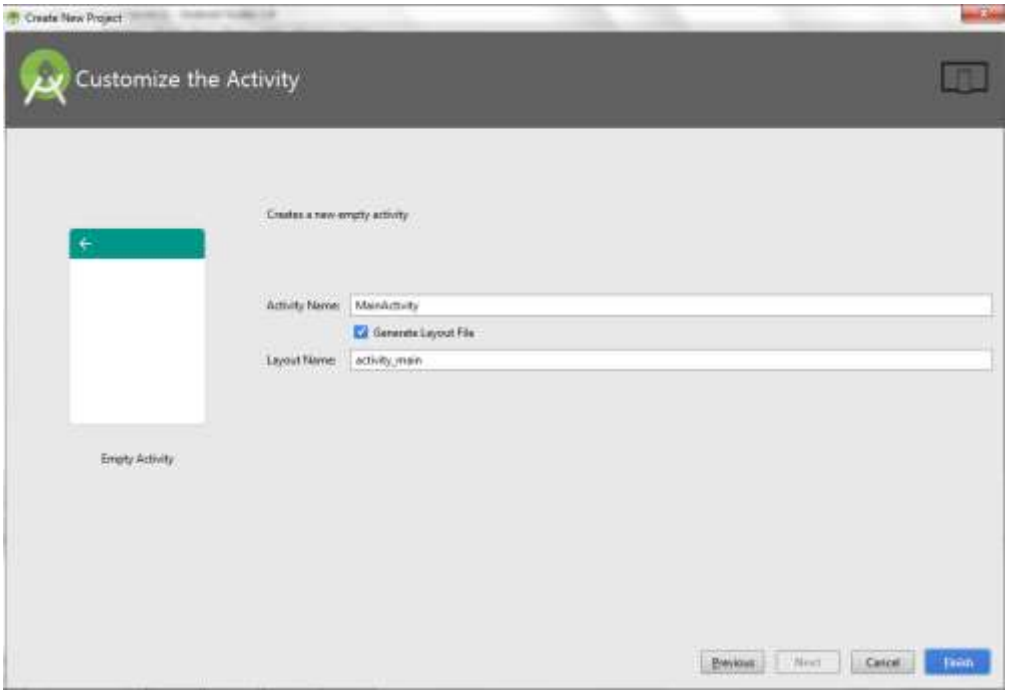
3. Then select the **Minimum SDK** as shown below and click **Next**.



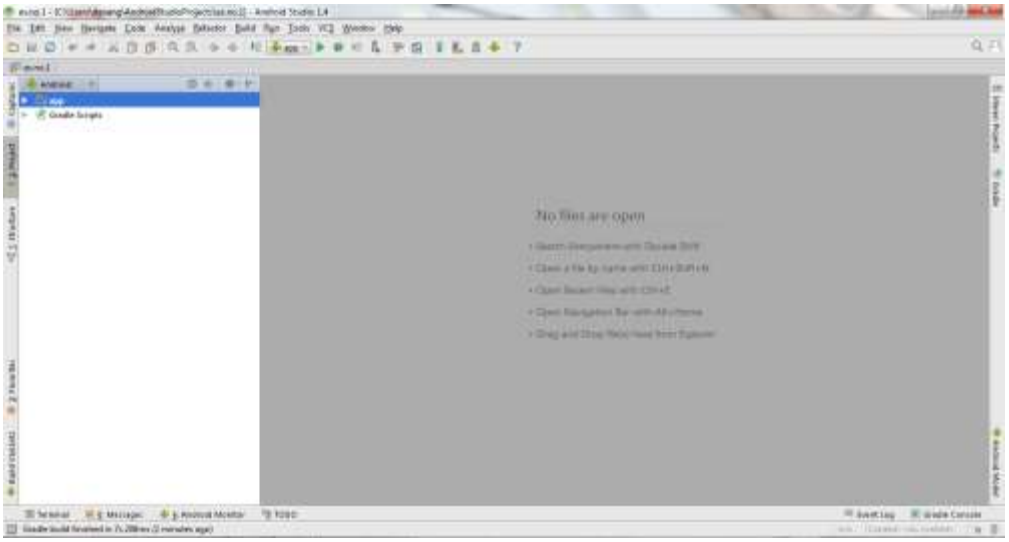
4. Then select the **Empty Activity** and click **Next.**



5. Finally click F**inish**.

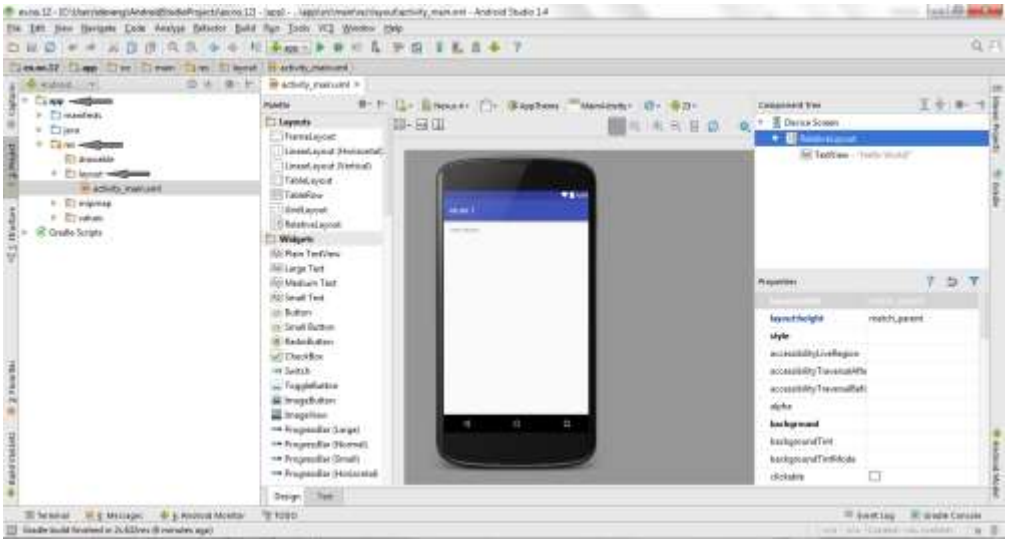


6. It will take some time to build and load the project.
7. After completion it will look as given below.

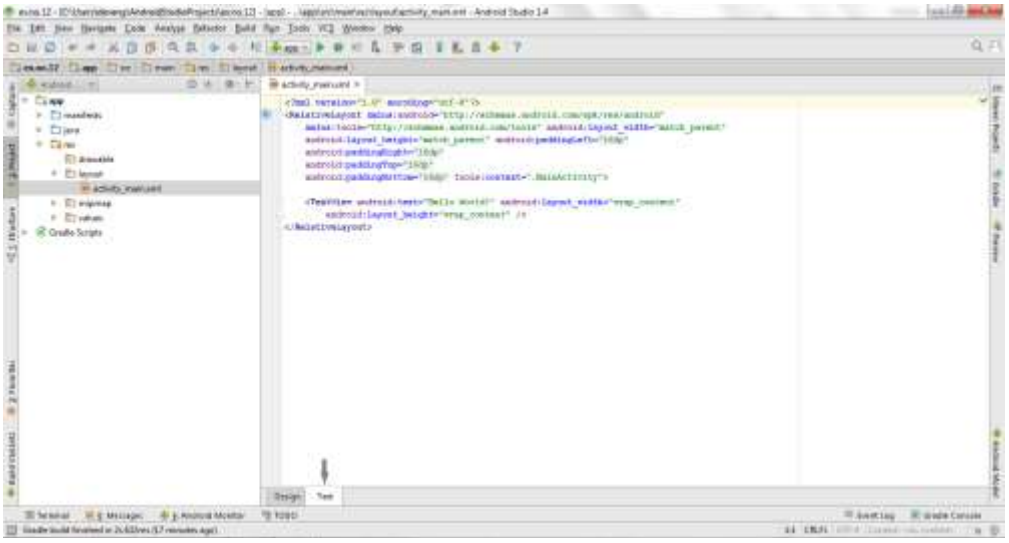


Designing layout for the Android Application:

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



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

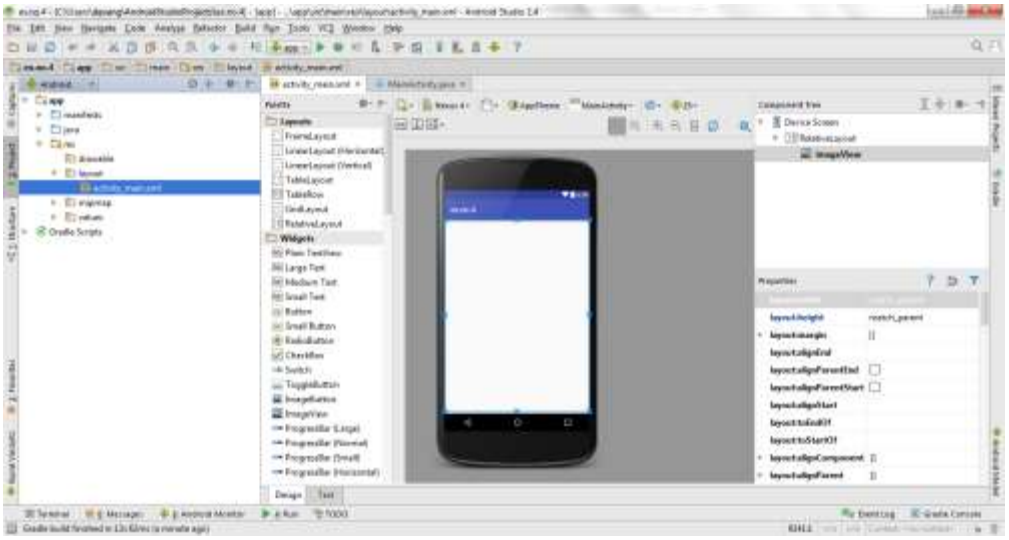


10. Then delete the code which is there and type the code as given below.

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>
```

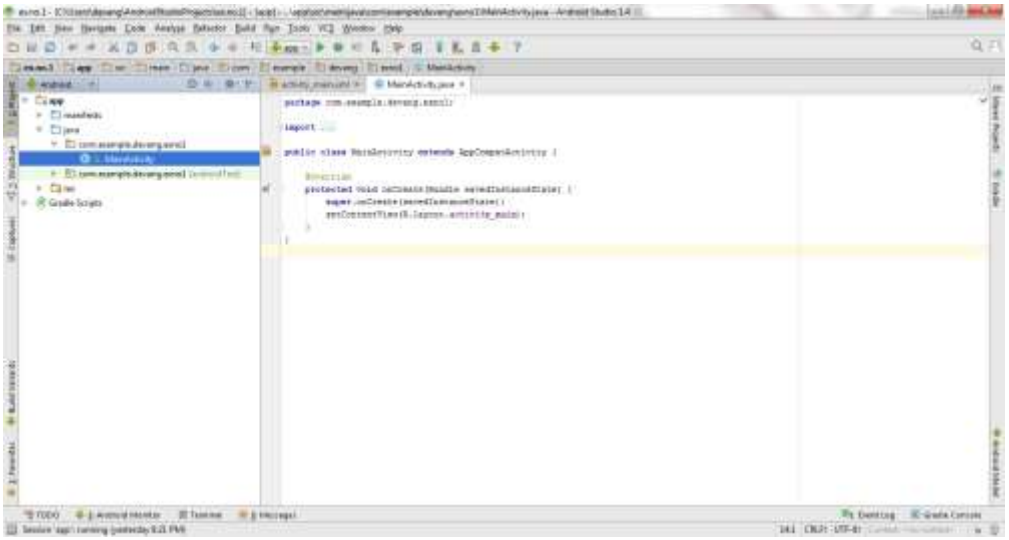
11. Now click on **Design** and your application will look as given below.



12. So now the designing part is completed.

Java Coding for the Android Application:

13. Click on **app -> java -> com.example.exno4 -> MainActivity.**

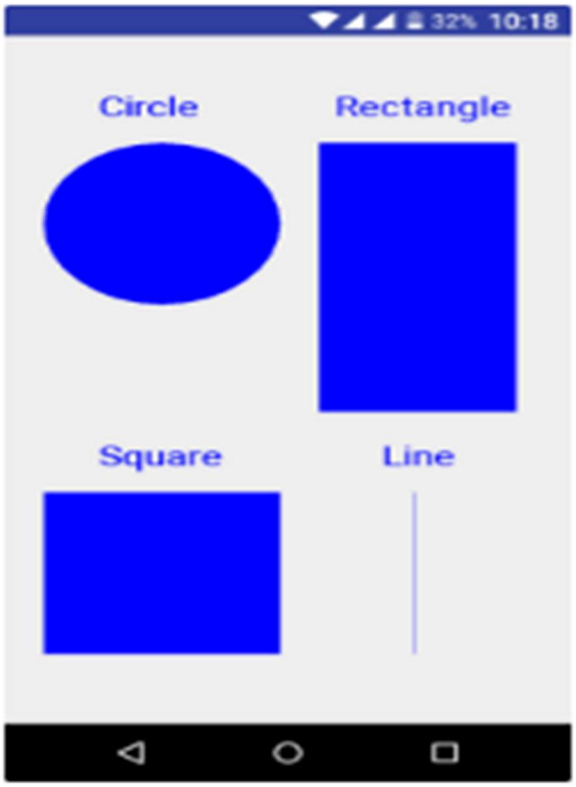


14. Then delete the code which is there and type the code as given below.

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;
```

Output:



```
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);
}
}
```

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

Observation	25	
Record	10	
Total	35	
Signature		

Result:

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

EX.NO: 5	DEVELOP AN APPLICATION THAT MAKES USE OF DATABASE.
Date:	

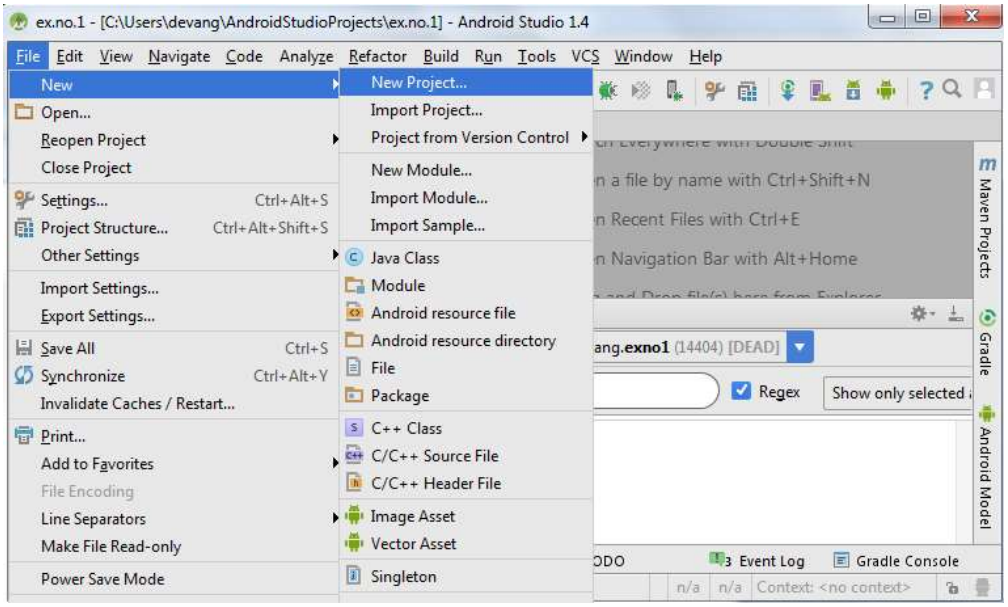
Aim:

To develop an application that makes use of database.

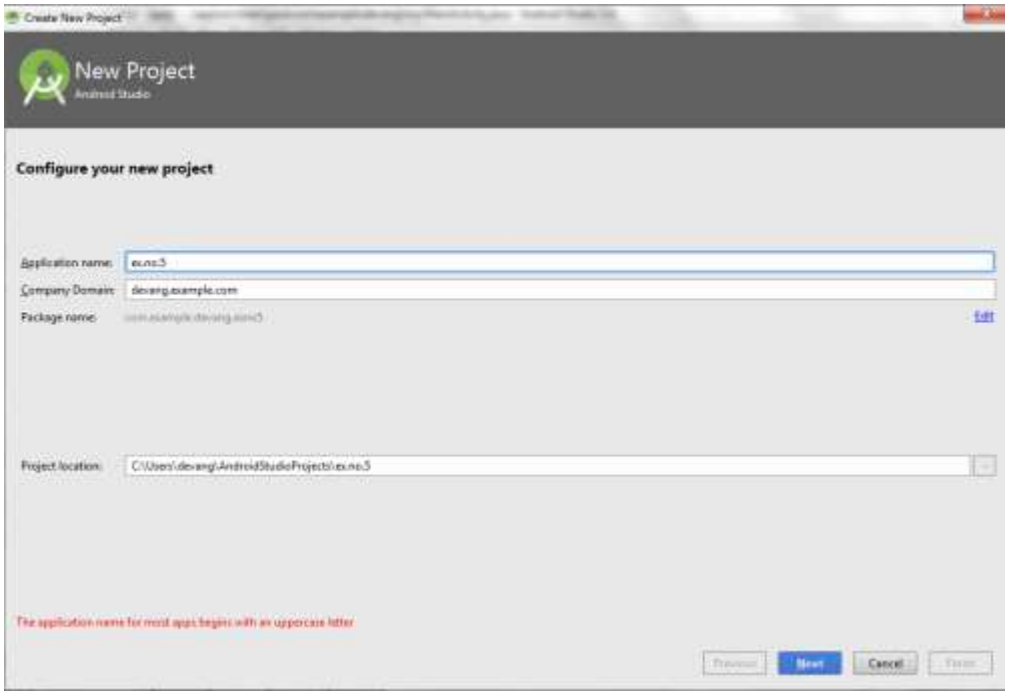
Procedure:

Creating a New project:

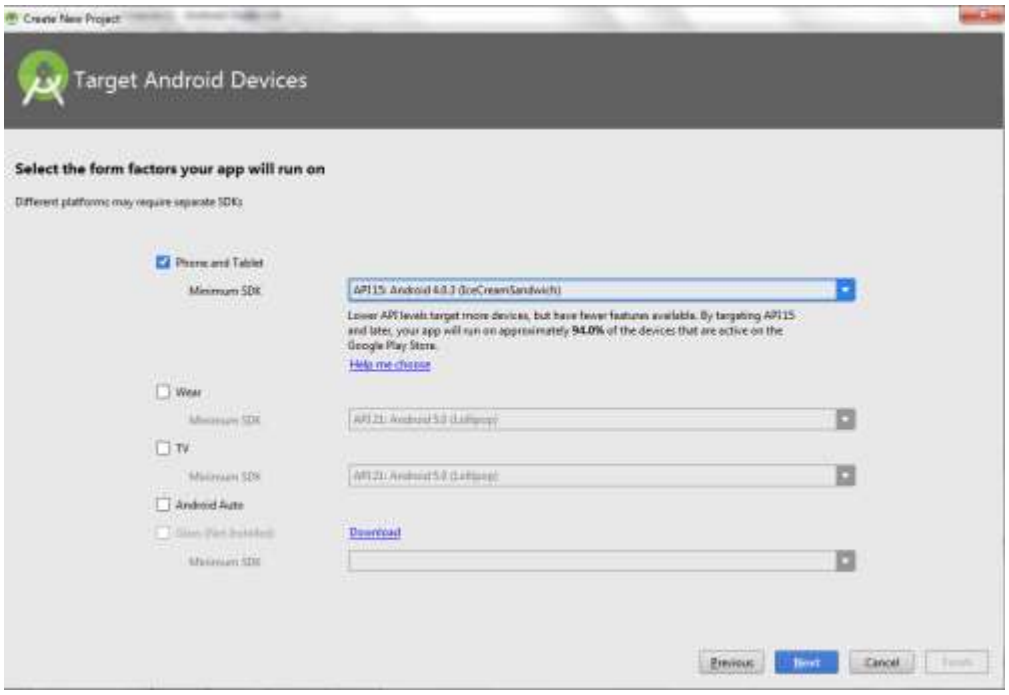
1. Open Android Studio and then click on **File -> New -> New project.**



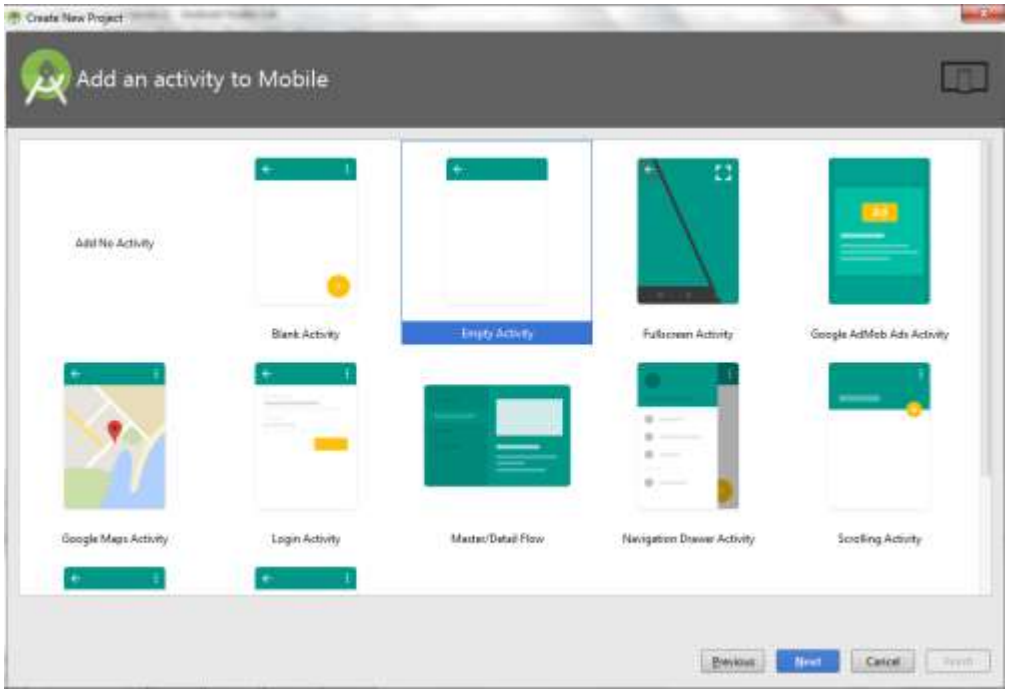
2. Then type the Application name as "**ex.no.5**" and click **Next.**



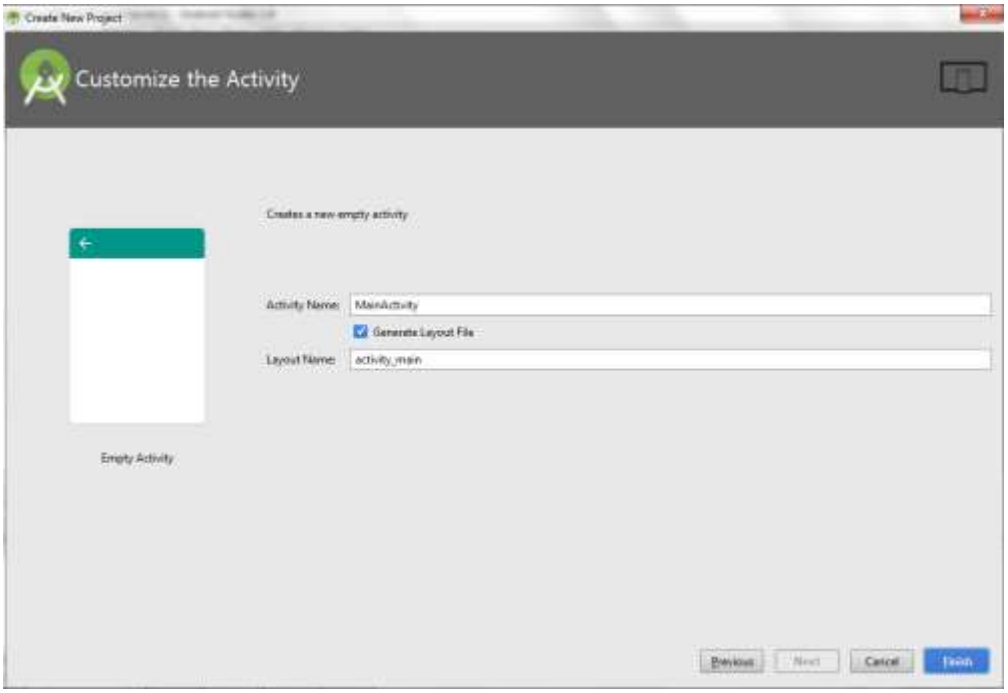
3. Then select the **Minimum SDK** as shown below and click **Next**.



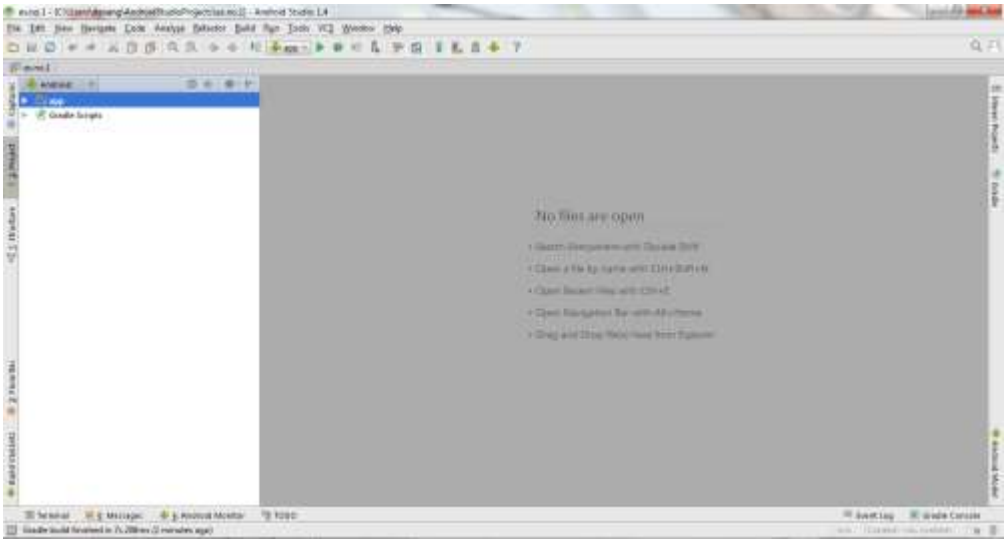
4. Then select the **Empty Activity** and click **Next.**



5. Finally click F**inish**.

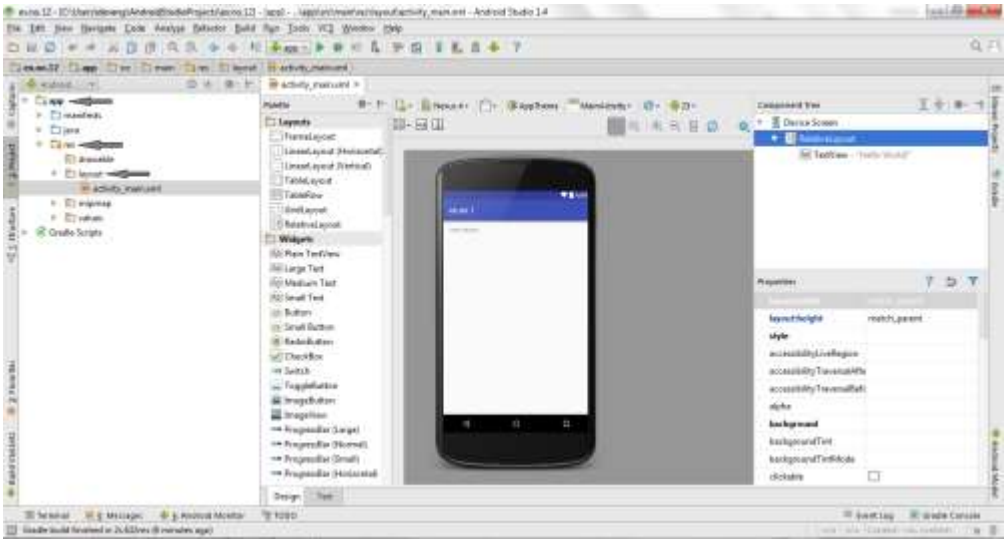


6. It will take some time to build and load the project.
7. After completion it will look as given below.

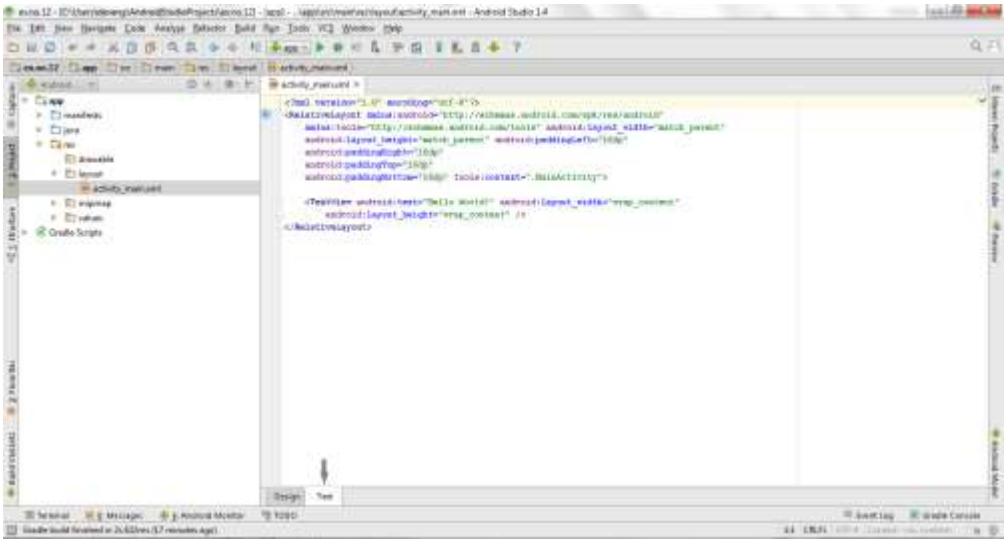


Designing layout for the Android Application:

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



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



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

Code for Activity_main.xml:

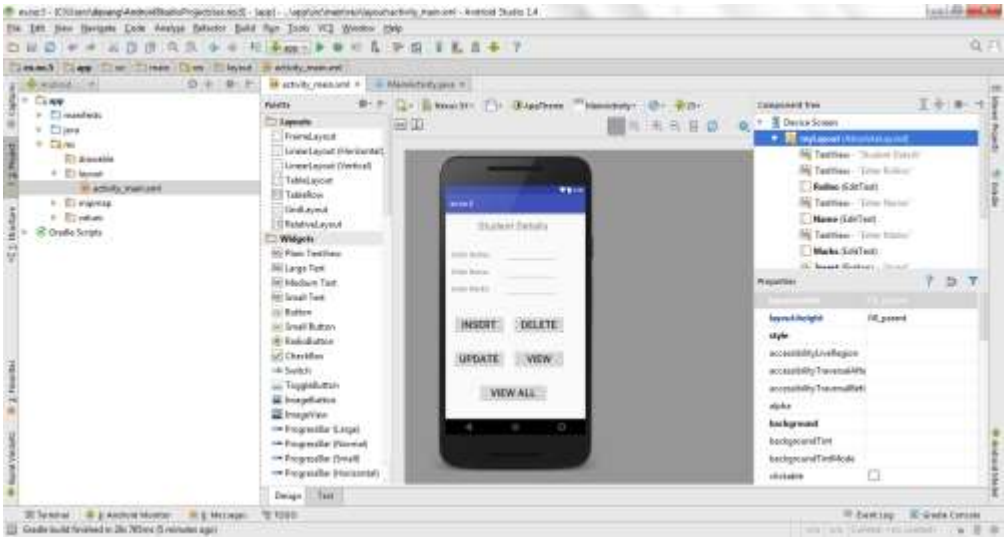
```
<?xml version="1.0" encoding="utf-8"?>
<AbsoluteLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="match_parent"
android:layout_height="match_parent">
<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_x="50dp"
android:layout_y="20dp"
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>
```

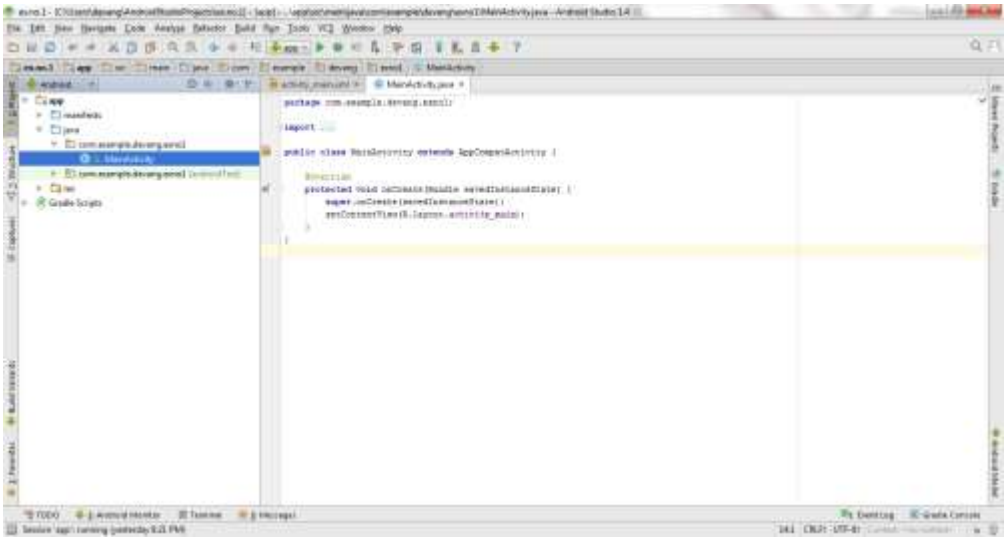
11. Now click on **Design** and your application will look as given below.



12. So now the designing part is completed.

Java Coding for the Android Application:

13. Click on **app -> java -> com.example.exno5 -> MainActivity.**



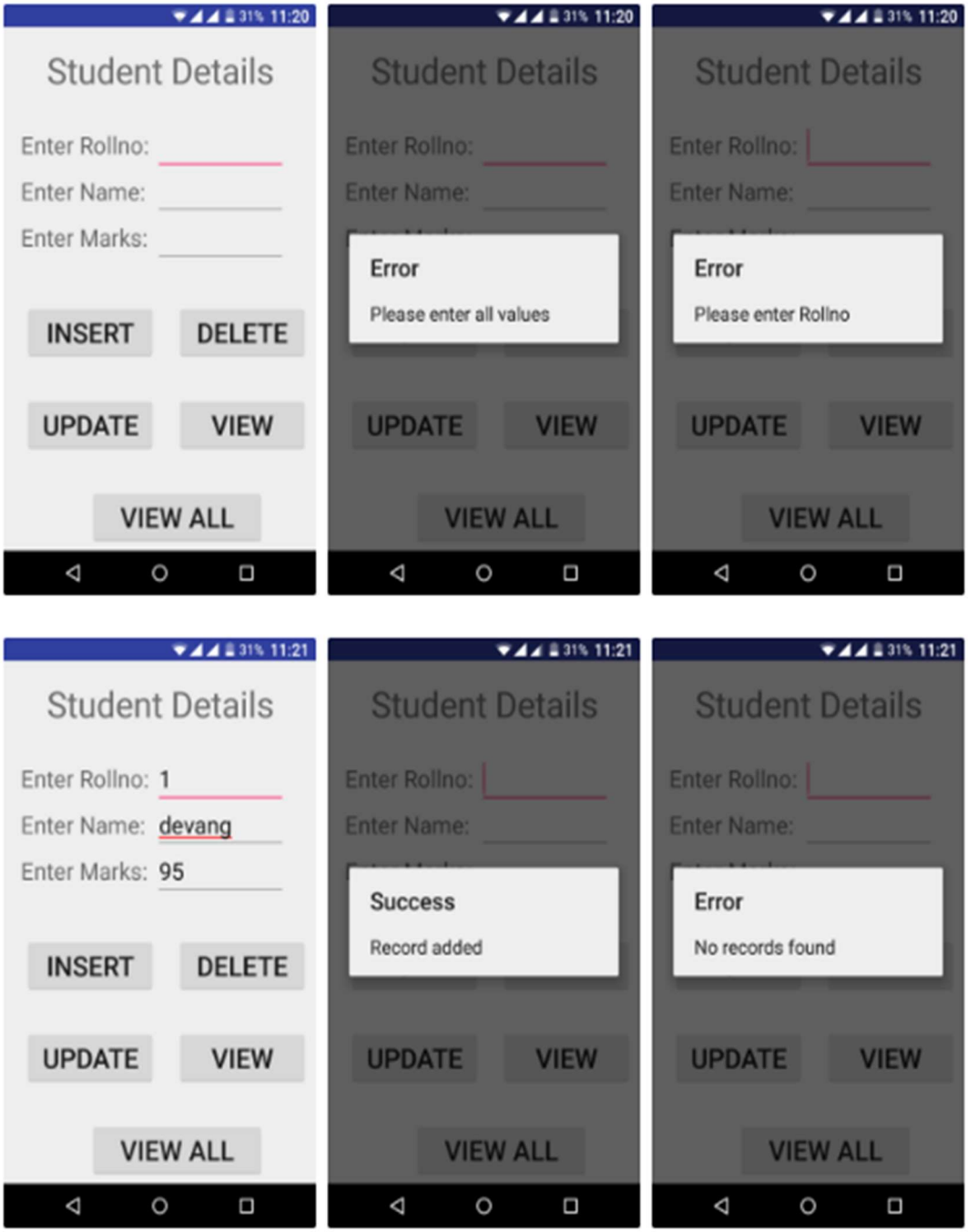
Then delete the code which is there and type the code as given below.

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)
{
```

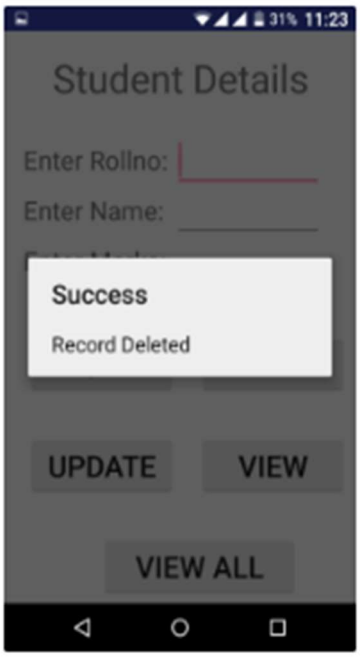
```
if(view==Insert)
{
// Checking for empty fields
if(Rollno.getText().toString().trim().length()==0||
Name.getText().toString().trim().length()==0||
Marks.getText().toString().trim().length()==0)
{
showMessage("Error", "Please enter all values");
return;
}
db.execSQL("INSERT INTO student VALUES('"+Rollno.getText()+"','"+Name.getText()+
"','"+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()) {
```

Output:

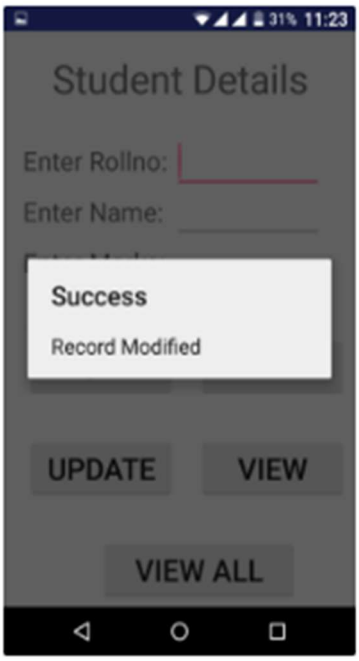

Student Details
Enter Rollno:
Enter Name:
Enter Marks:
INSERT
DELETE
UPDATE
VIEW
VIEW ALL
Error
Please enter all values
Error
Please enter Rollno
1
devang
95
Success
Record added
Error
No records found

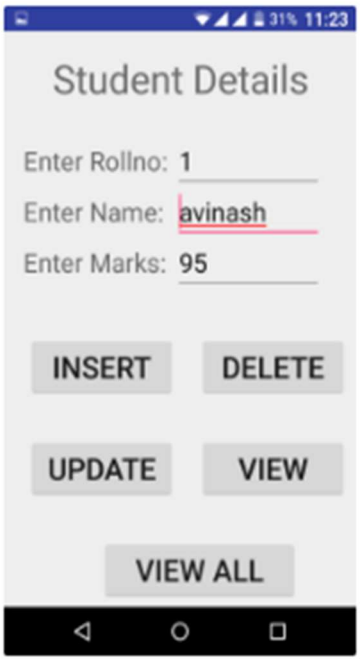
```
db.execSQL("UPDATE student SET name='" + Name.getText() + "',marks='" +
Marks.getText() +
"' WHERE rollno='"+Rollno.getText()+"'");
showMessage("Success", "Record Modified");
}
else {
showMessage("Error", "Invalid Rollno");
}
clearText();
}
// Display a record from the Student table
if(view==View)
{
// 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();
}
}
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());
}
```


31% 11:22
Student Details
Student Details
Rollno: 1
Name: devang
Marks: 95
Rollno: 2
Name: akhil
Marks: 90
Rollno: 3
Name: abdul
Marks: 80
VIEW ALL


31% 11:23
Student Details
Enter Rollno:
Enter Name:
Success
Record Deleted
UPDATE
VIEW
VIEW ALL


31% 11:23
Student Details
Student Details
Rollno: 2
Name: akhil
Marks: 90
Rollno: 3
Name: abdul
Marks: 80
VIEW ALL


31% 11:23
Student Details
Enter Rollno:
Enter Name:
Success
Record Modified
UPDATE
VIEW
VIEW ALL


31% 11:23
Student Details
Enter Rollno: 1
Enter Name: avinash
Enter Marks: 95
INSERT
DELETE
UPDATE
VIEW
VIEW ALL

```
}
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();
}
}
```

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

Observation	25	
Record	10	
Total	35	
Signature		

Result:

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

Ex.No: 6 Date:	DEVELOP AN APPLICATION THAT MAKES USE OF RSS FEED.

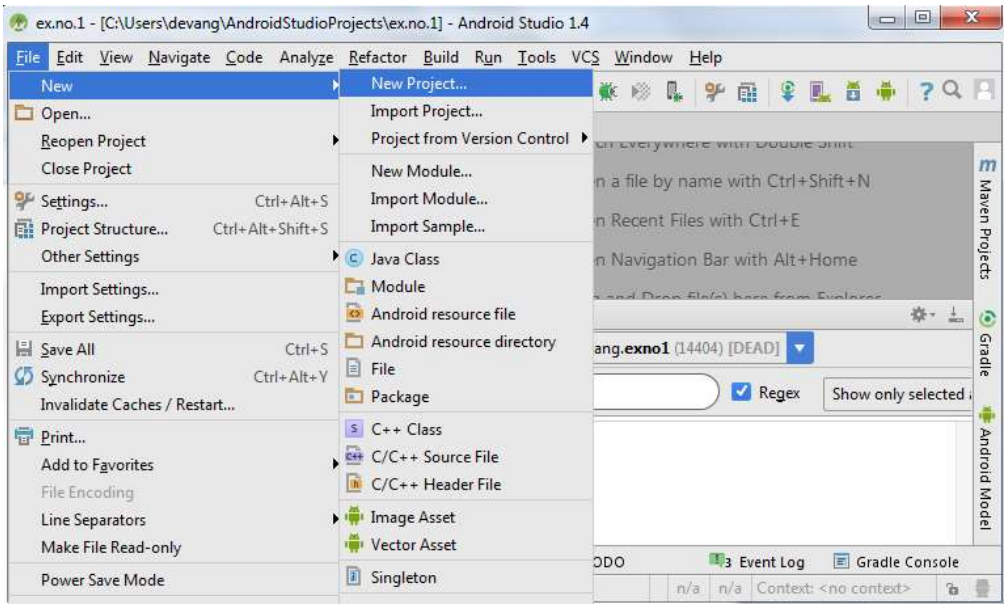
Aim:

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

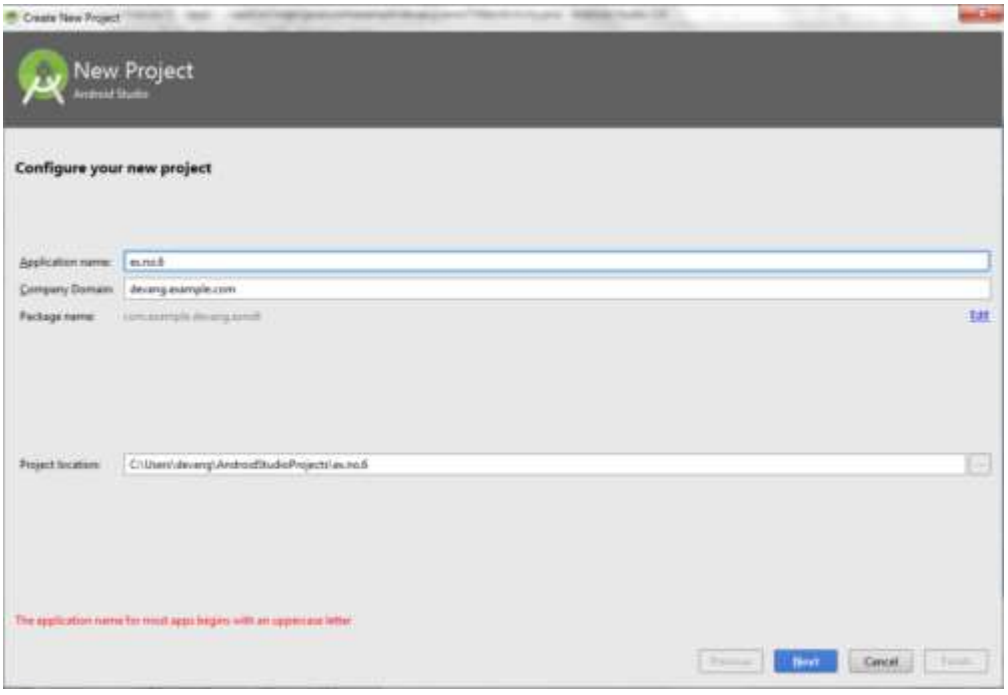
Procedure:

Creating a New project:

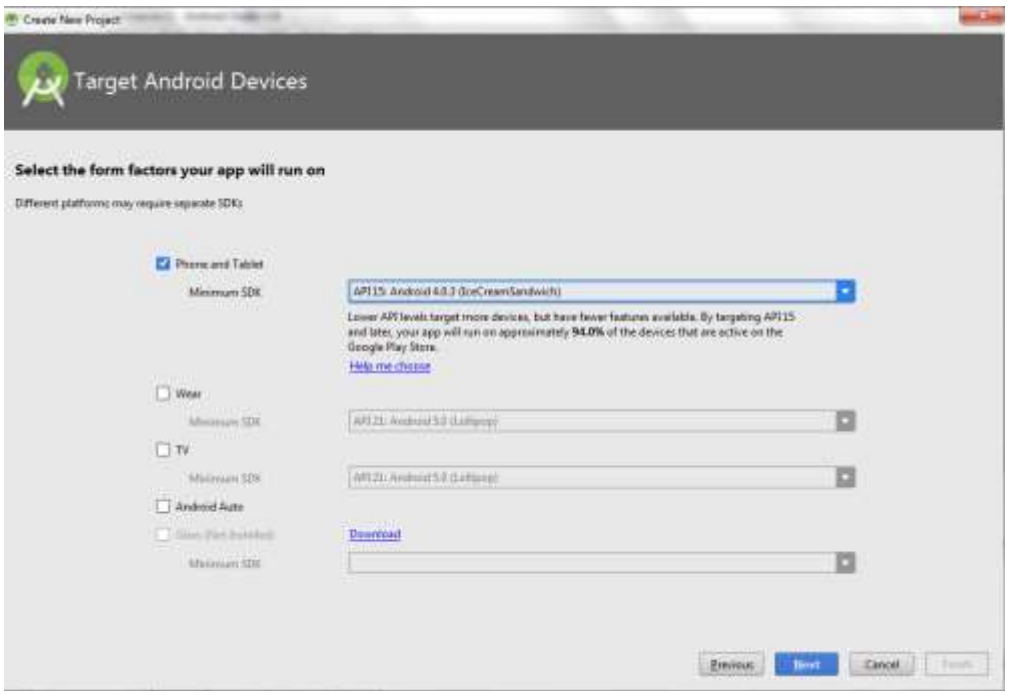
1. Open Android Studio and then click on **File -> New -> New project.**



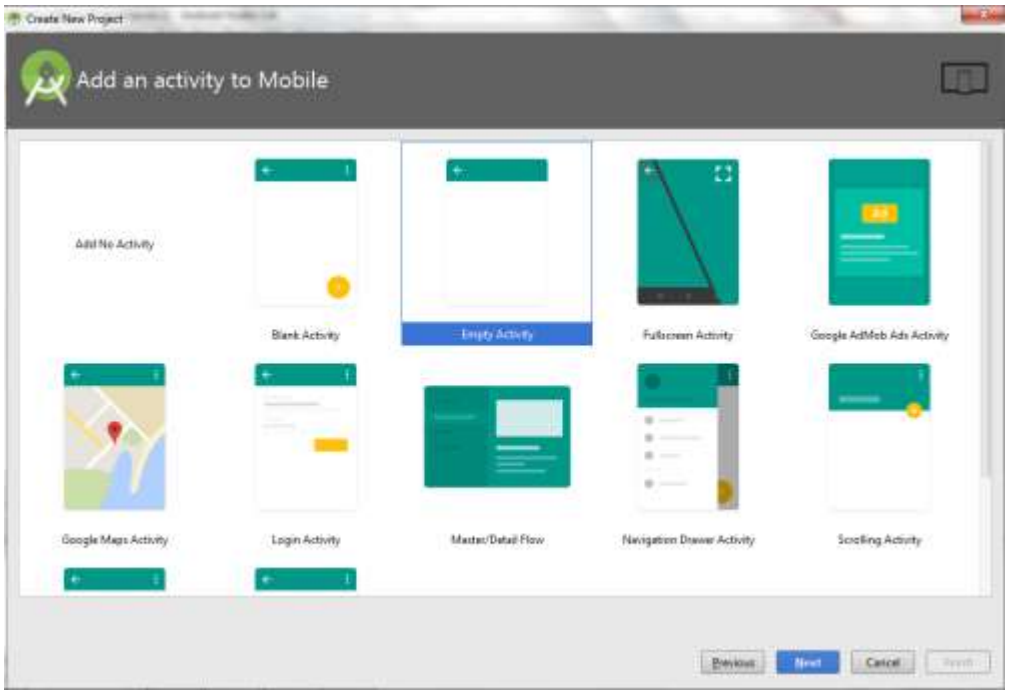
2. Then type the Application name as "**ex.no.6"** and click **Next.**



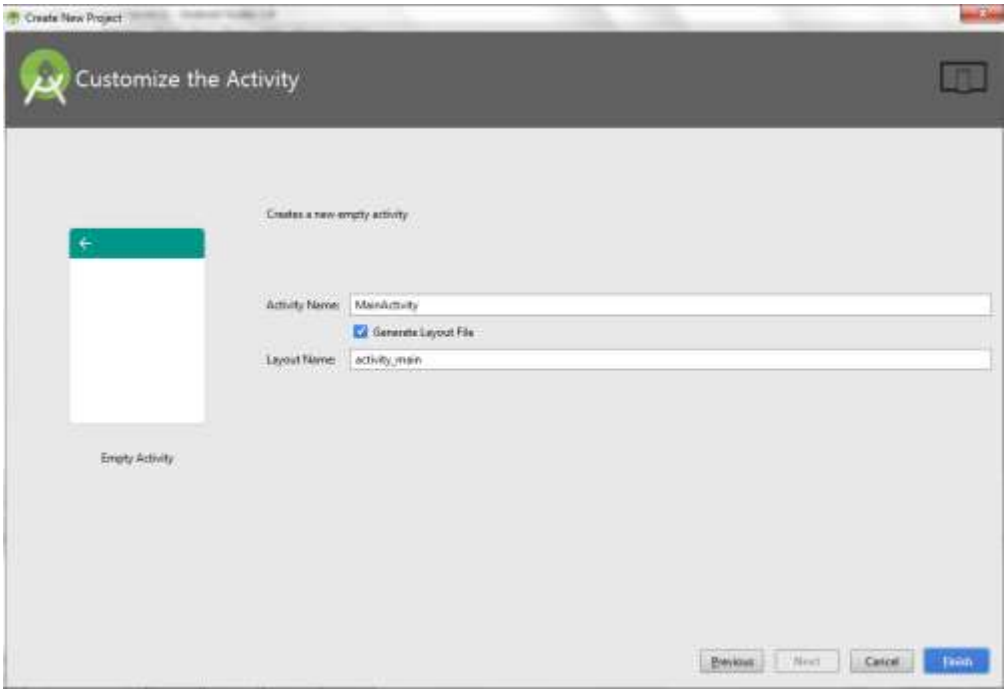
3. Then select the **Minimum SDK** as shown below and click **Next**.



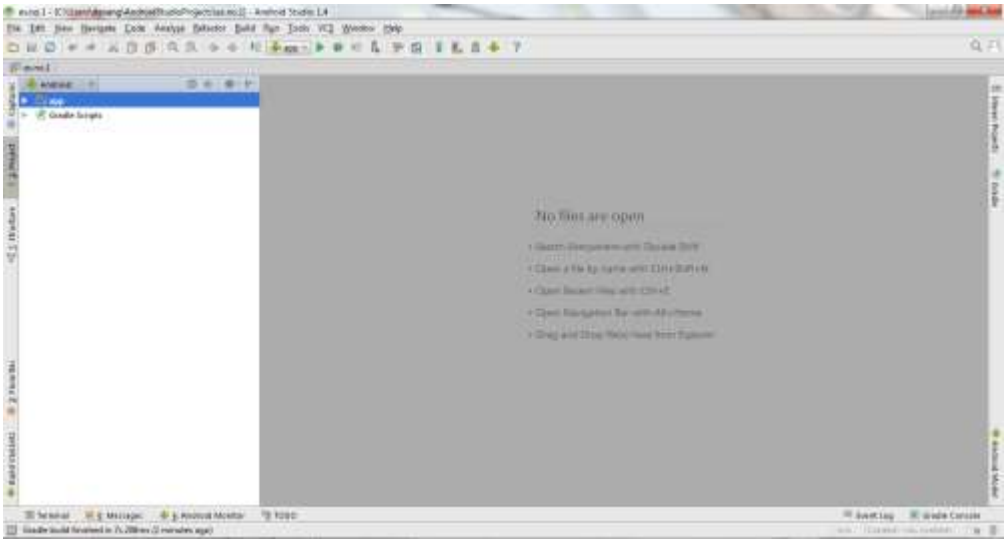
4. Then select the **Empty Activity** and click **Next.**



5. Finally click F**inish**.

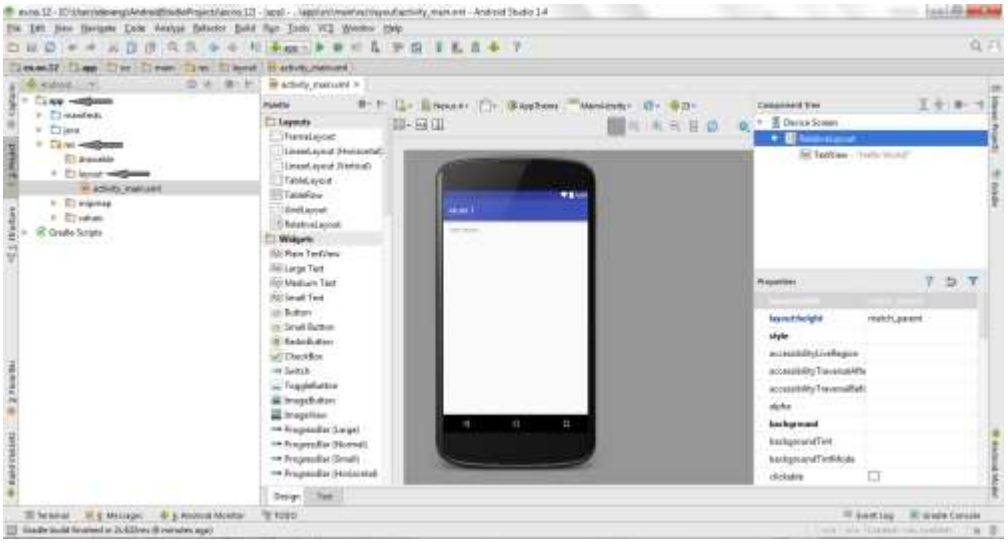


6. It will take some time to build and load the project.
7. After completion it will look as given below.

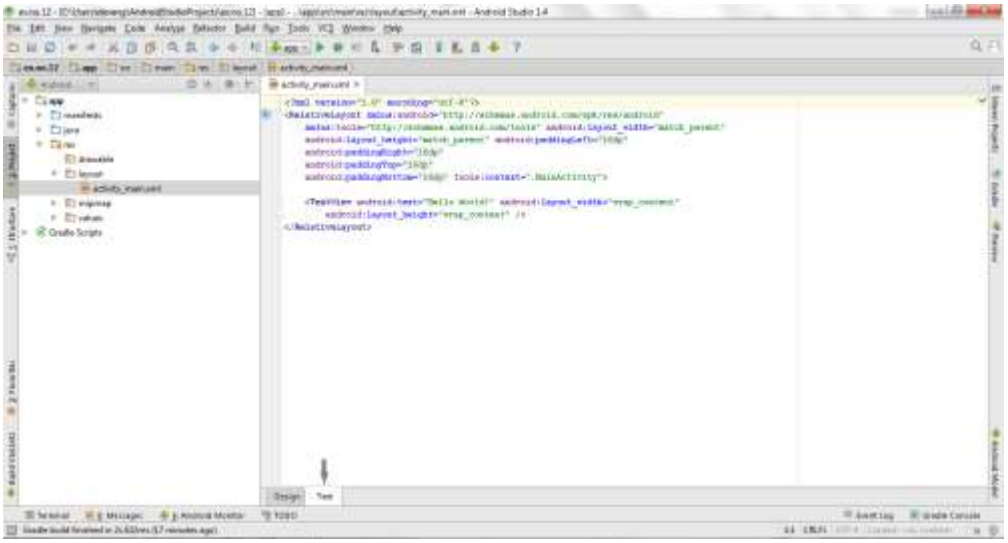


Designing layout for the Android Application:

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



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

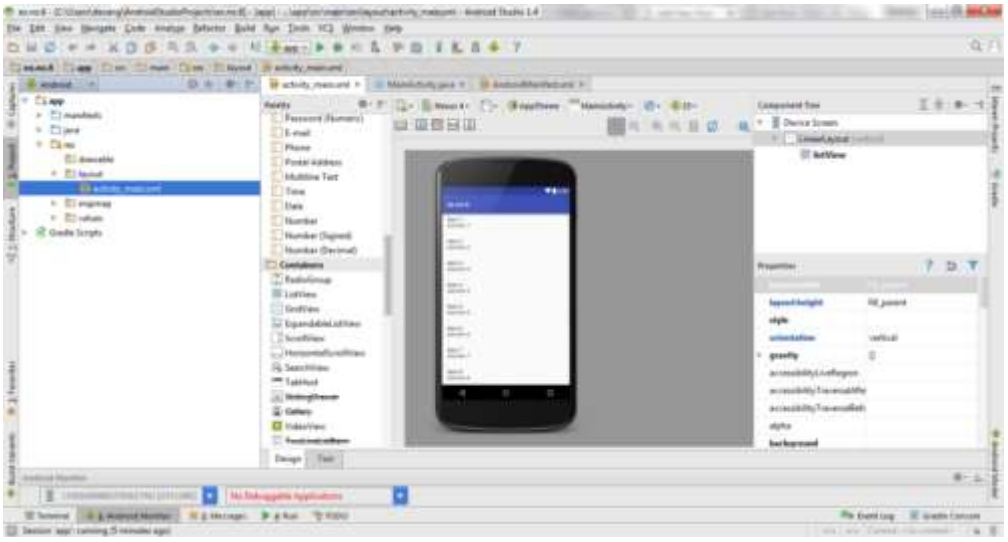


10. Then delete the code which is there and type the code as given below.

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>
```

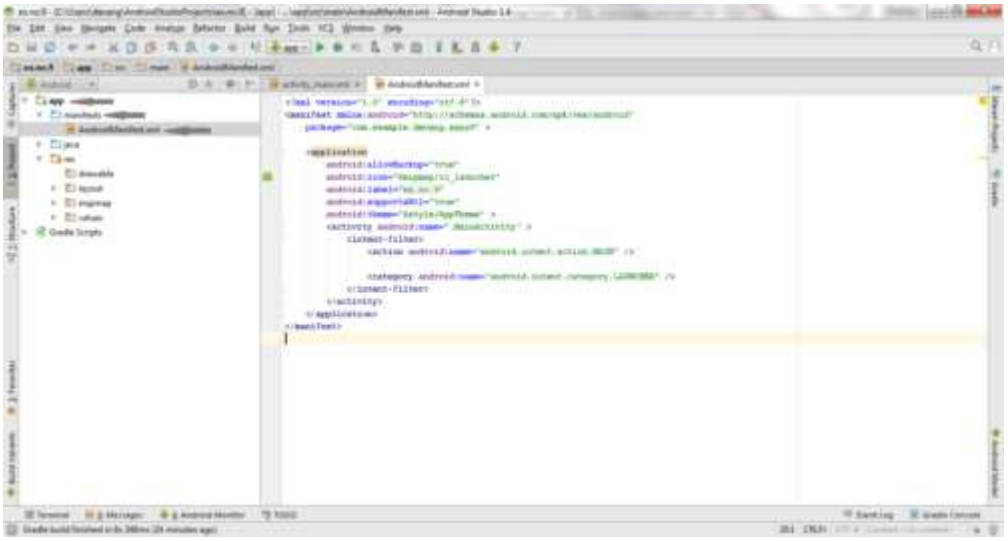
11. Now click on **Design** and your application will look as given below.



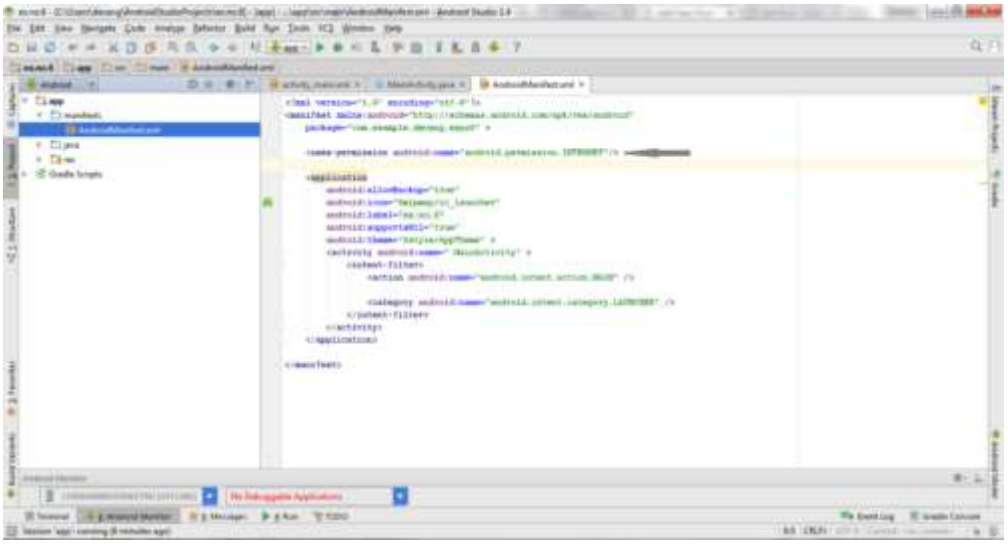
12. So now the designing part is completed.

Adding permissions in Manifest for the Android Application:

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



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



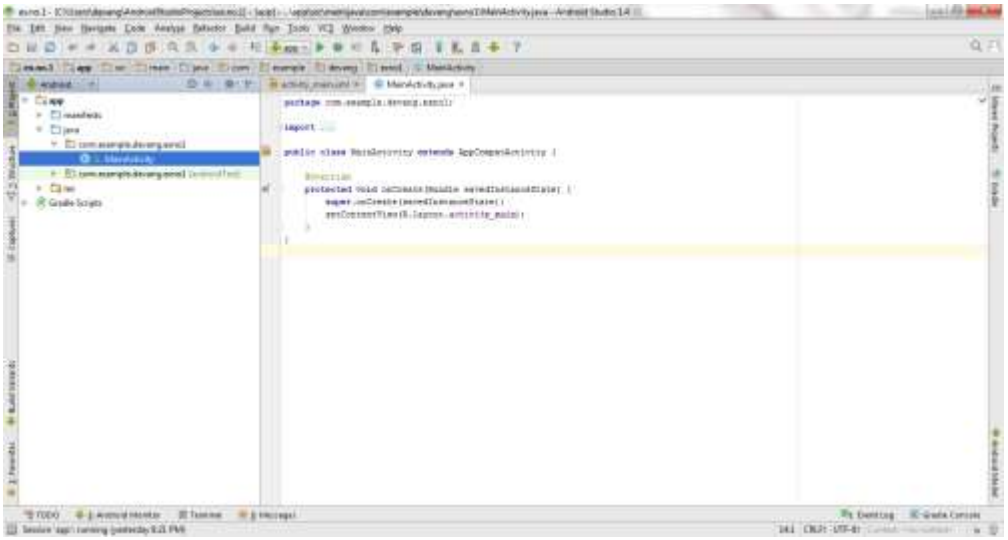
Code for AndroidManifest.xml:

```
<?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>
```

15. So now the Permissions are added in the Manifest.

Java Coding for the Android Application:

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



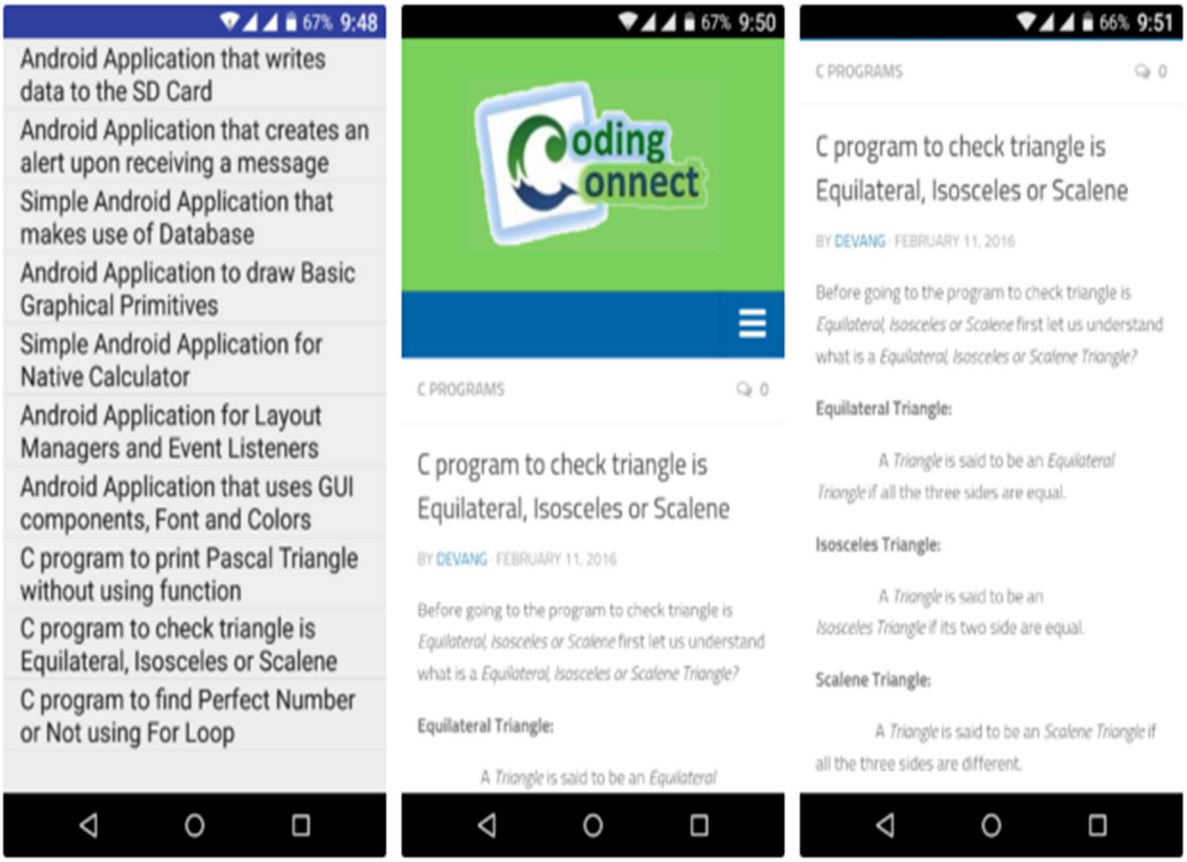
17. Then delete the code which is there and type the code as given below.

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)
{
```

Output:


67% 9:48
Android Application that writes data to the SD Card
Android Application that creates an alert upon receiving a message
Simple Android Application that makes use of Database
Android Application to draw Basic Graphical Primitives
Simple Android Application for Native Calculator
Android Application for Layout Managers and Event Listeners
Android Application that uses GUI components, Font and Colors
C program to print Pascal Triangle without using function
C program to check triangle is Equilateral, Isosceles or Scalene
C program to find Perfect Number or Not using For Loop
67% 9:50
C PROGRAMS
0
C program to check triangle is Equilateral, Isosceles or Scalene
BY DEVANG · FEBRUARY 11, 2016
Before going to the program to check triangle is Equilateral, Isosceles or Scalene first let us understand what is a Equilateral, Isosceles or Scalene Triangle?
Equilateral Triangle:
A Triangle is said to be an Equilateral
66% 9:51
C PROGRAMS
0
C program to check triangle is Equilateral, Isosceles or Scalene
BY DEVANG · FEBRUARY 11, 2016
Before going to the program to check triangle is Equilateral, Isosceles or Scalene first let us understand what is a Equilateral, Isosceles or Scalene Triangle?
Equilateral Triangle:
A Triangle is said to be an Equilateral Triangle if all the three sides are equal.
Isosceles Triangle:
A Triangle is said to be an Isosceles Triangle if its two side are equal.
Scalene Triangle:
A Triangle is said to be an Scalene Triangle if all the three sides are different.

```
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;
}
}
}
```

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

Observation	25	
Record	10	
Total	35	
Signature		

Result:

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

Ex.No: 7	IMPLEMENT AN APPLICATION THAT IMPLEMENTS MULTI THREADING.
Date:	

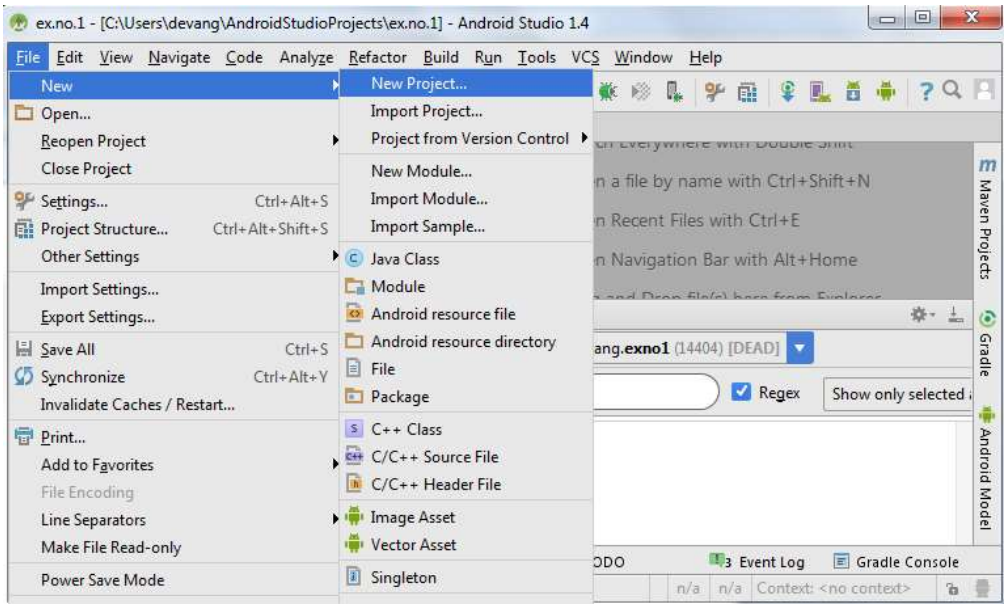
Aim:

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

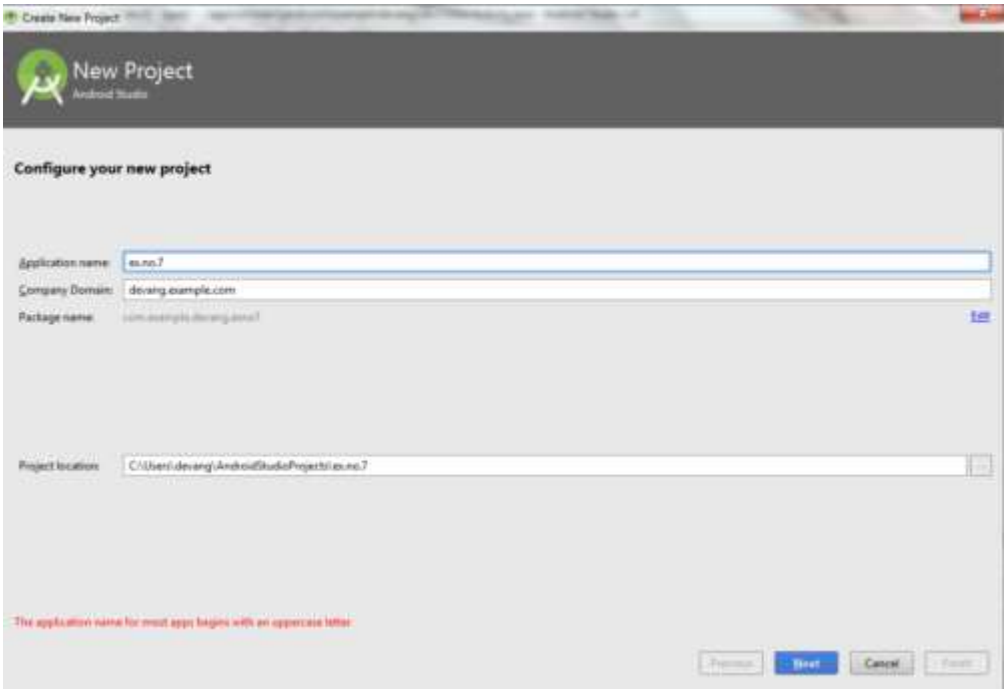
Procedure:

Creating a New project:

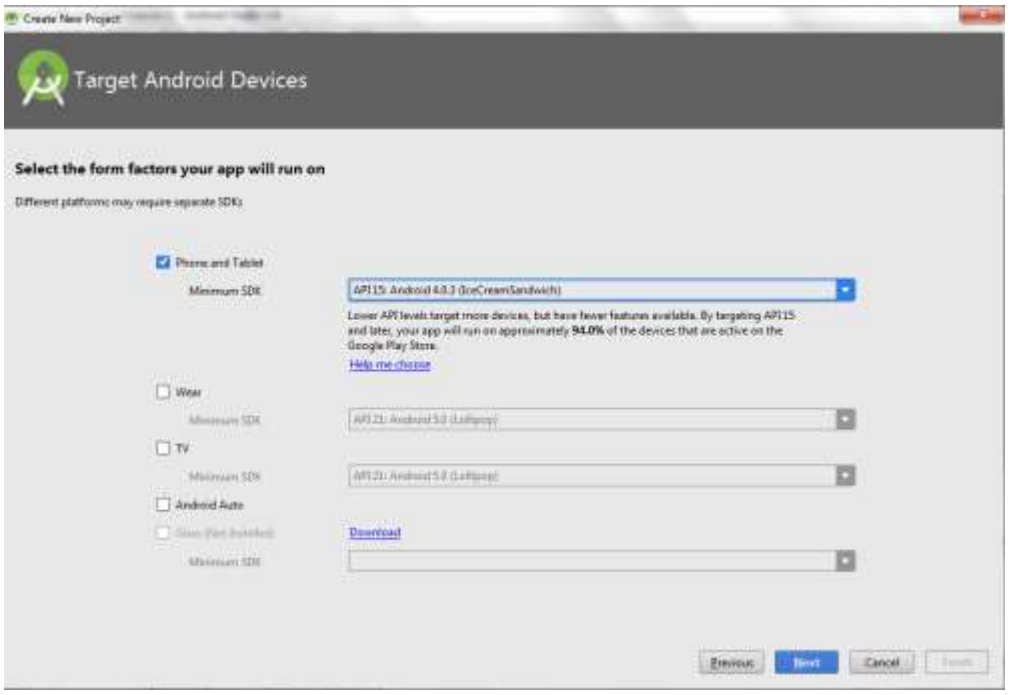
1. Open Android Studio and then click on **File -> New -> New project.**



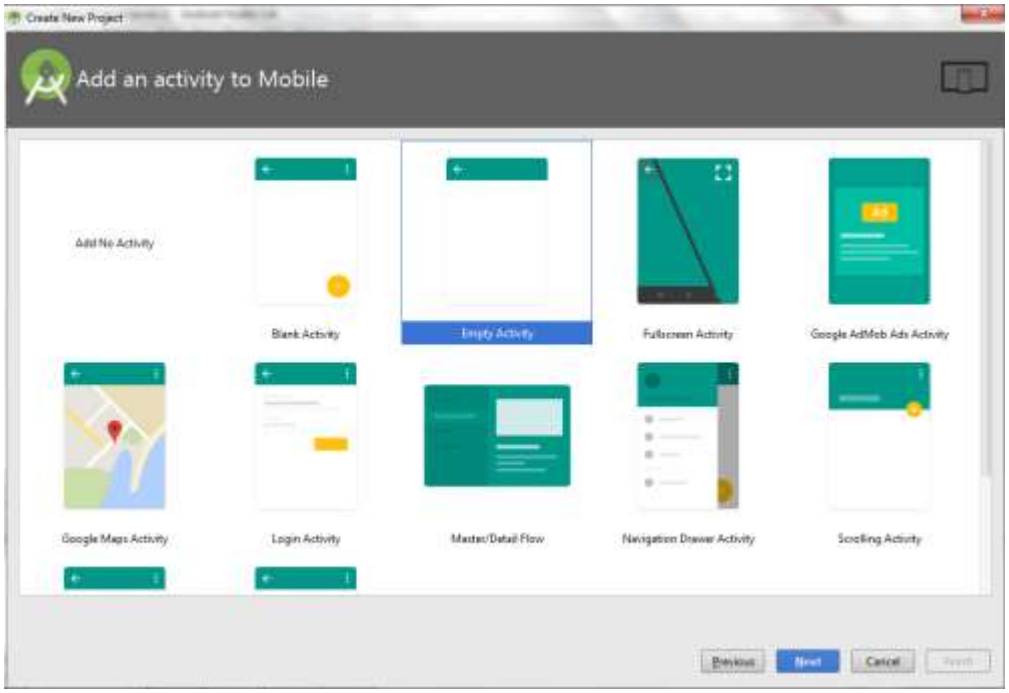
2. Then type the Application name as "**ex.no.7"** and click **Next.**



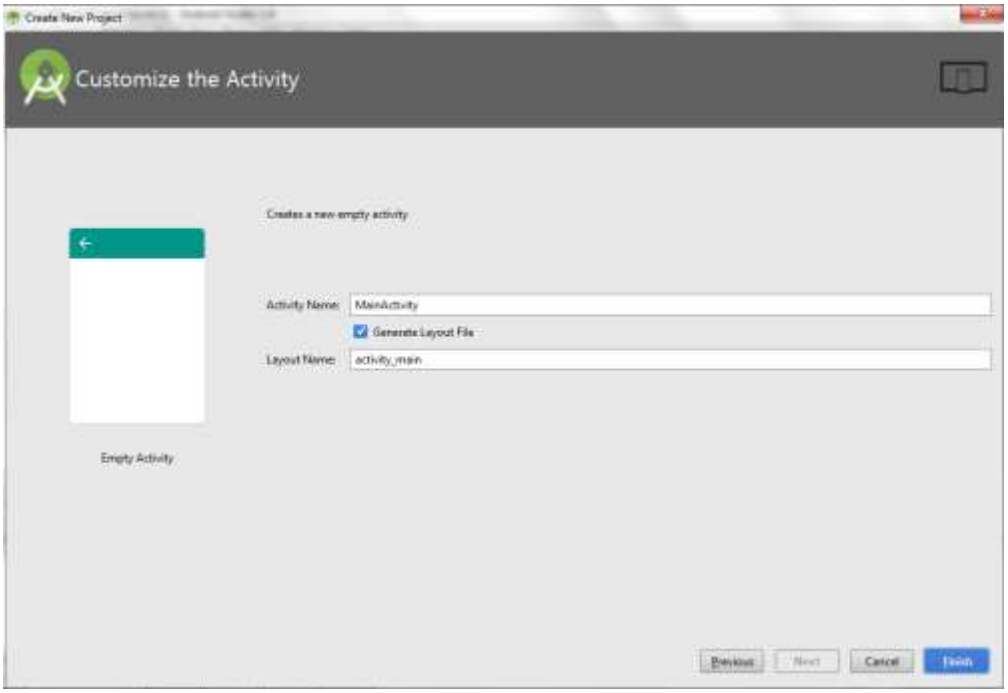
3. Then select the **Minimum SDK** as shown below and click **Next**.



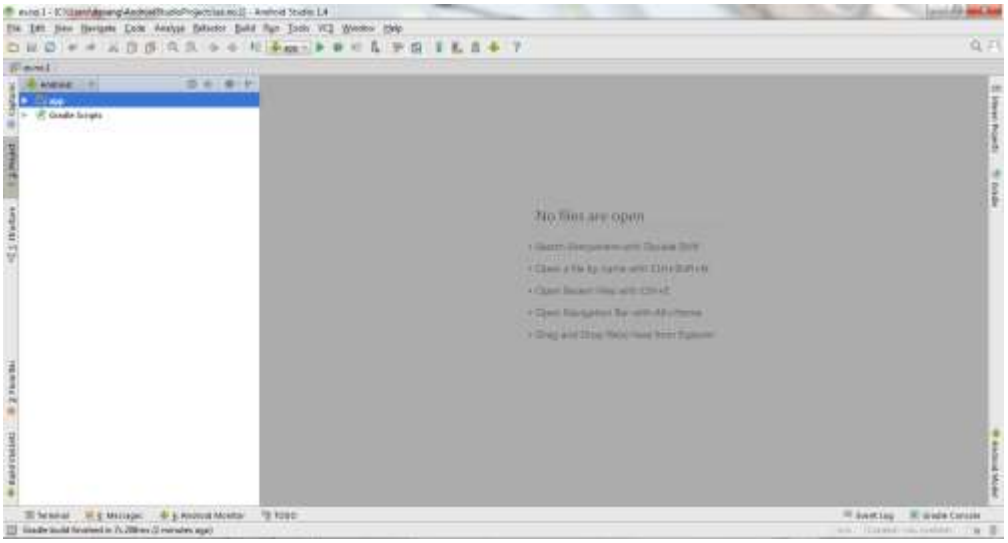
4. Then select the **Empty Activity** and click **Next.**



5. Finally click F**inish**.

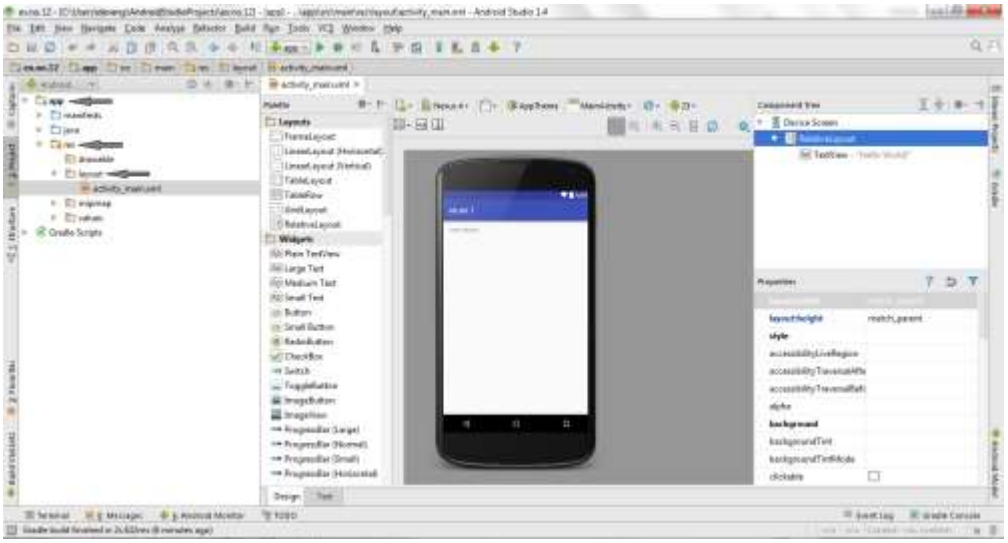


6. It will take some time to build and load the project.
7. After completion it will look as given below.

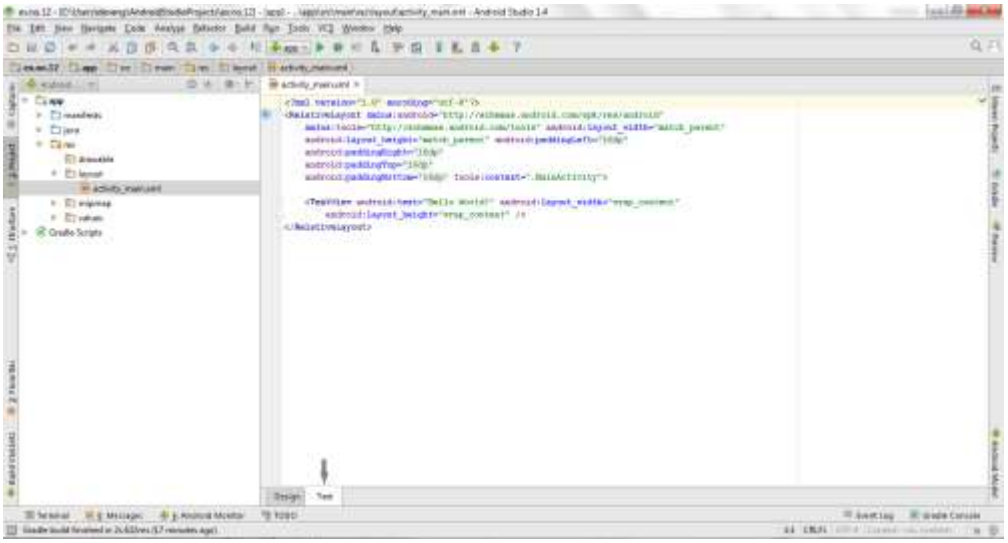


Designing layout for the Android Application:

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



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



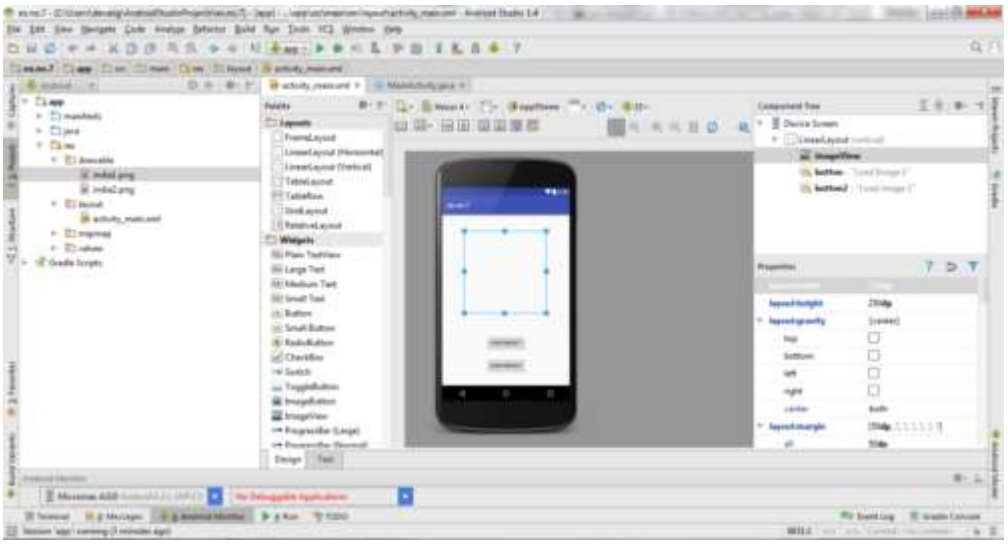
10. Then delete the code which is there and type the code as given below.

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>
```

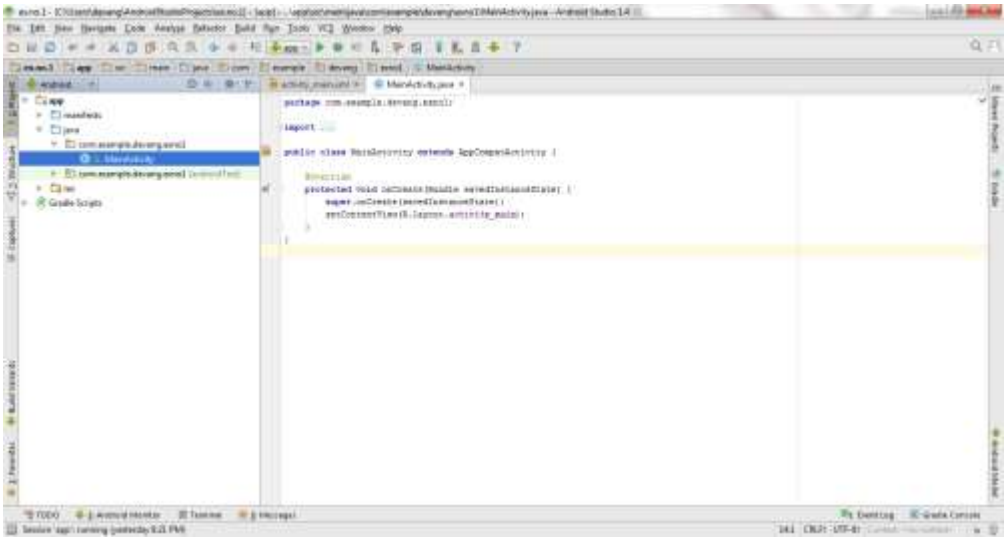
11. Now click on **Design** and your application will look as given below.



12. So now the designing part is completed.

Java Coding for the Android Application:

13. Click on **app -> java -> com.example.exno7 -> MainActivity.**

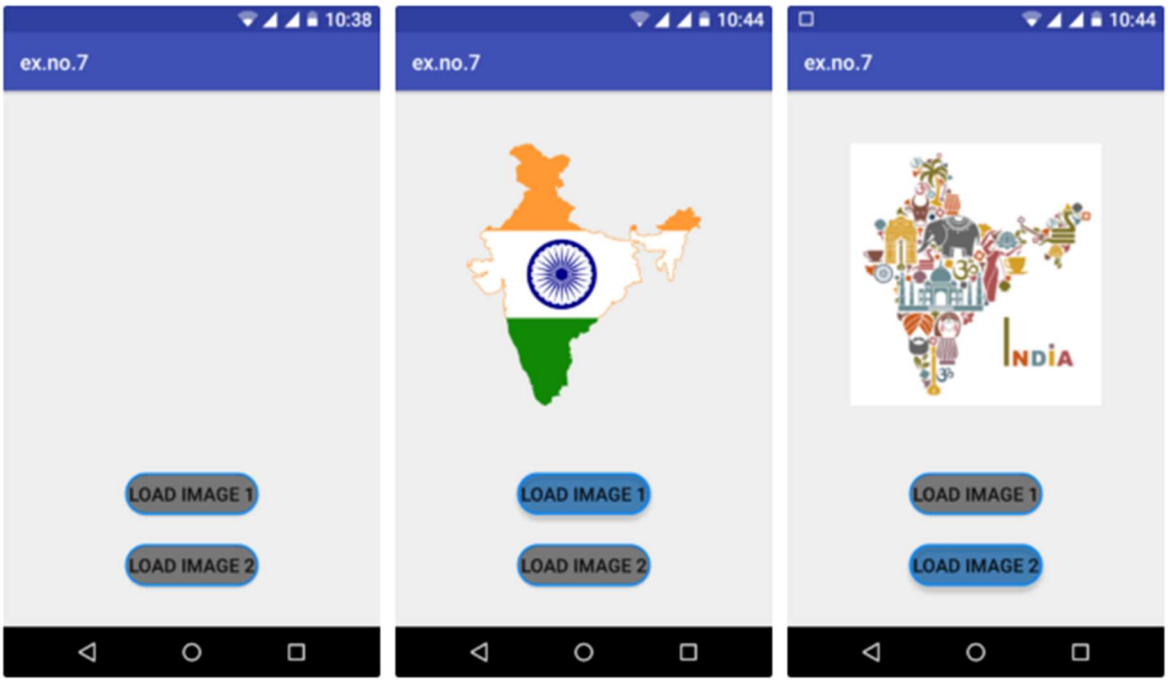


14. Then delete the code which is there and type the code as given below.

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()
{
```

Output:


10:38
ex.no.7
LOAD IMAGE 1
LOAD IMAGE 2
10:44
ex.no.7
LOAD IMAGE 1
LOAD IMAGE 2
10:44
ex.no.7
INDIA
LOAD IMAGE 1
LOAD IMAGE 2

```
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();
}
});
}
}
```

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

Observation	25	
Record	10	
Total	35	
Signature		

Result:

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

Ex.No: 8	DEVELOP A NATIVE APPLICATION THAT USES GPS LOCATION INFORMATION.
Date:	

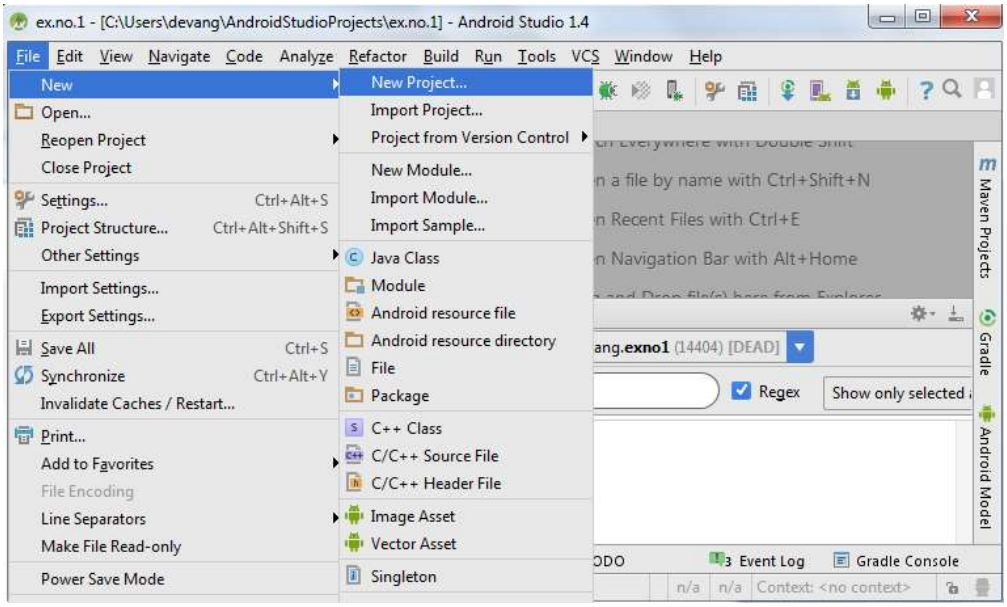
Aim:

To develop a native application that uses gps location information.

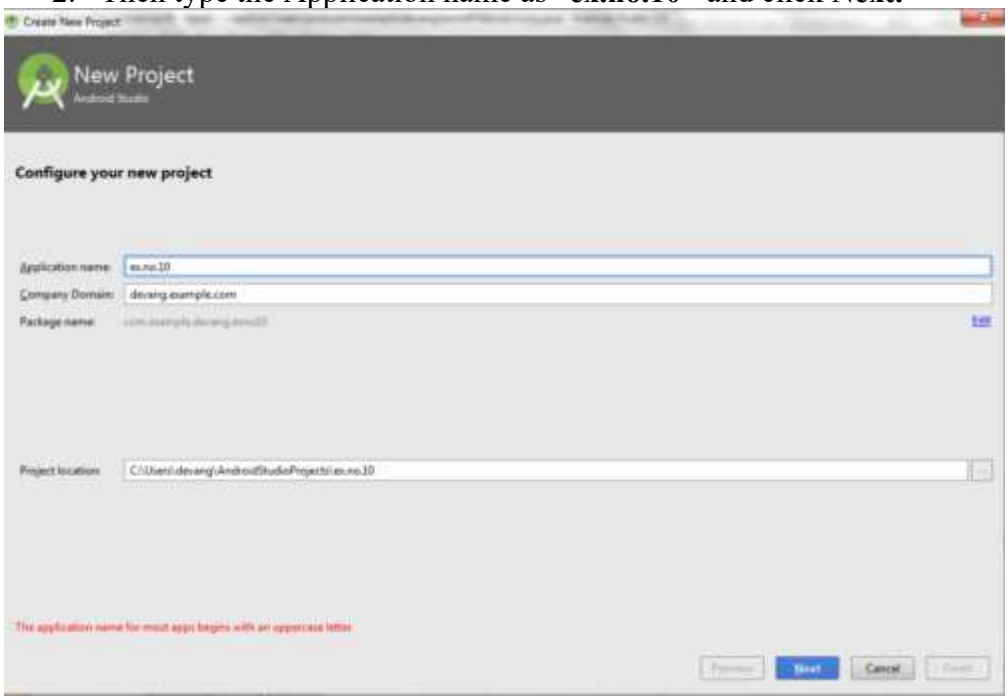
Procedure:

Creating a New project:

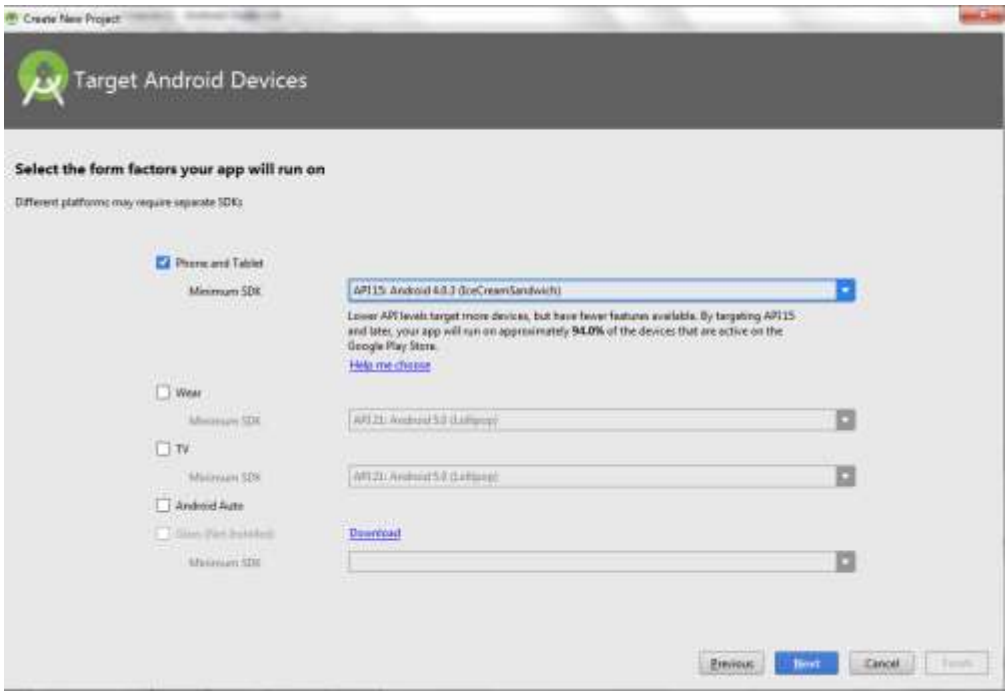
1. Open Android Studio and then click on **File -> New -> New project.**



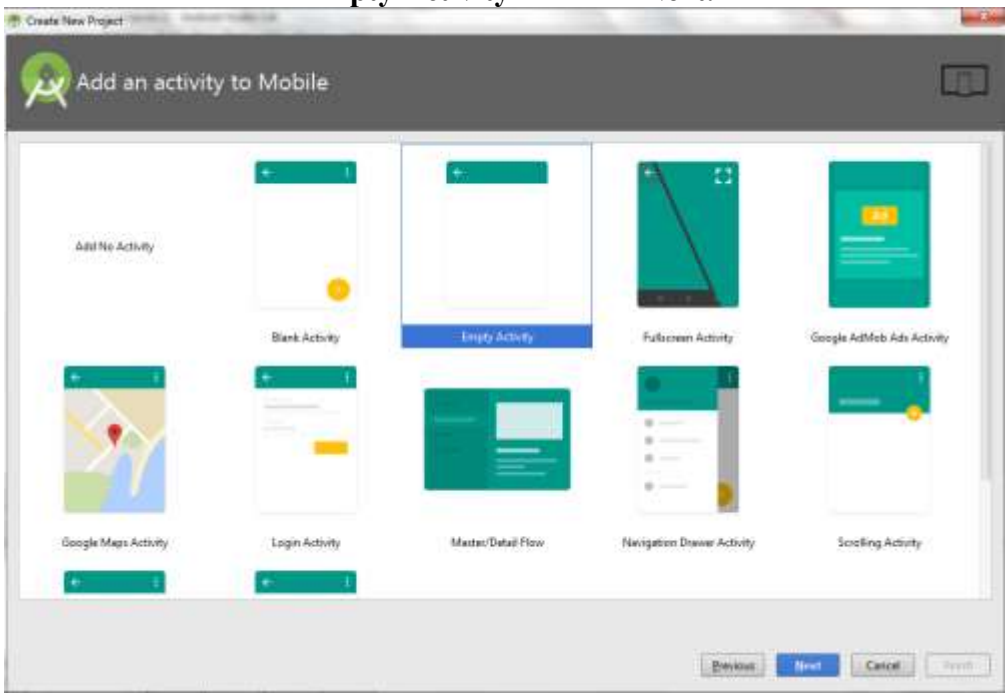
2. Then type the Application name as "**ex.no.10"** and click **Next.**



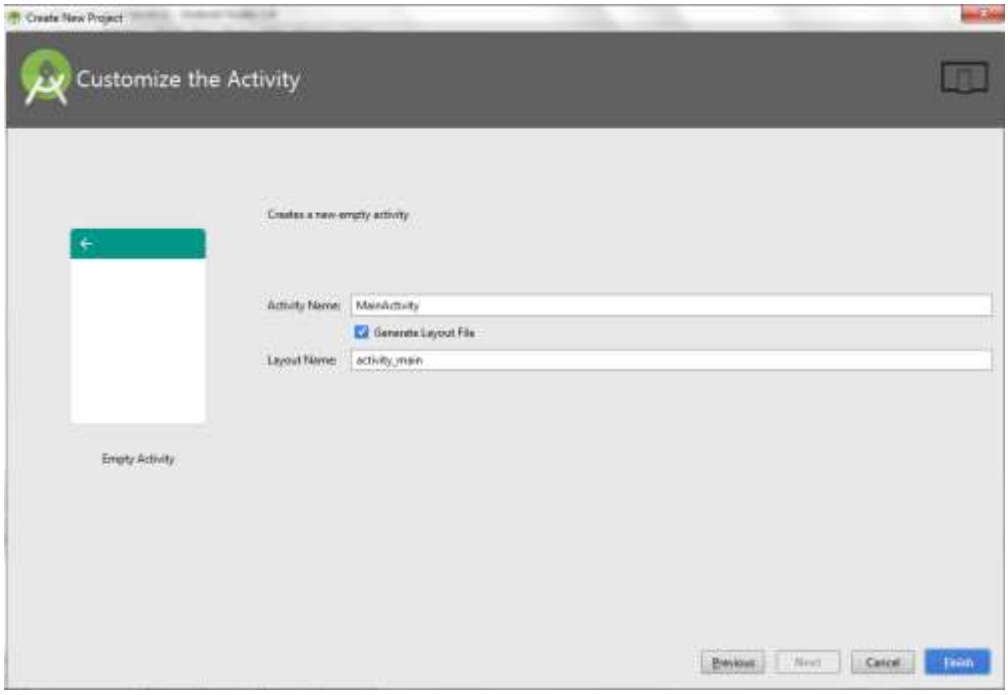
3. Then select the **Minimum SDK** as shown below and click **Next**.



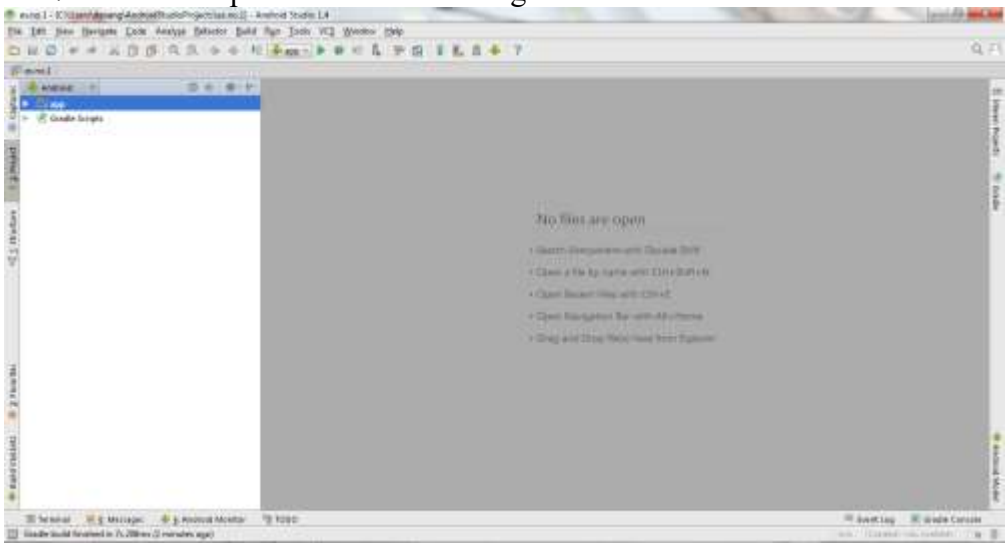
4. Then select the **Empty Activity** and click **Next.**



5. Finally click F**inish**.

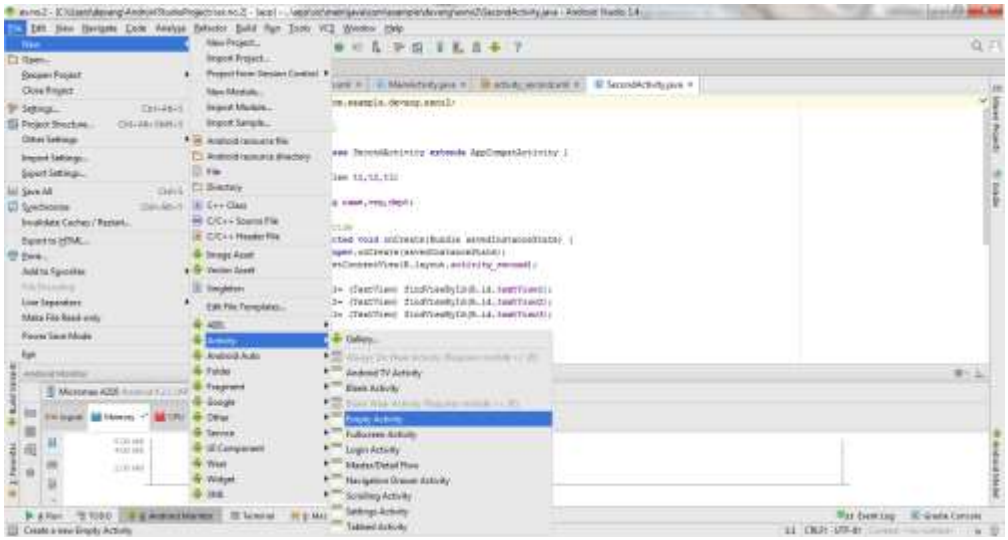


6. It will take some time to build and load the project.
7. After completion it will look as given below.

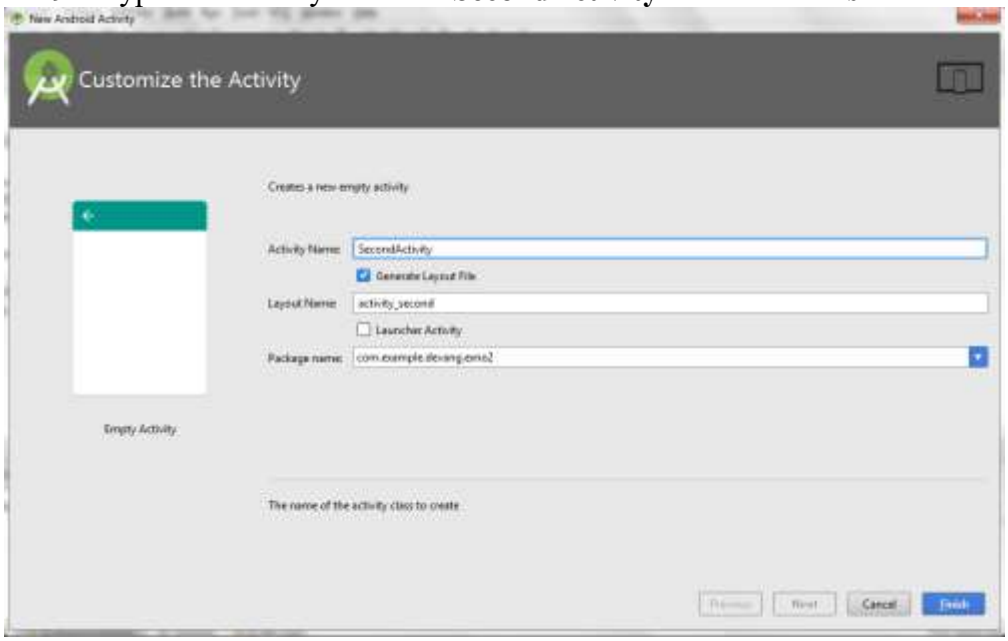


8. Creating Second Activity for the Android Application:

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

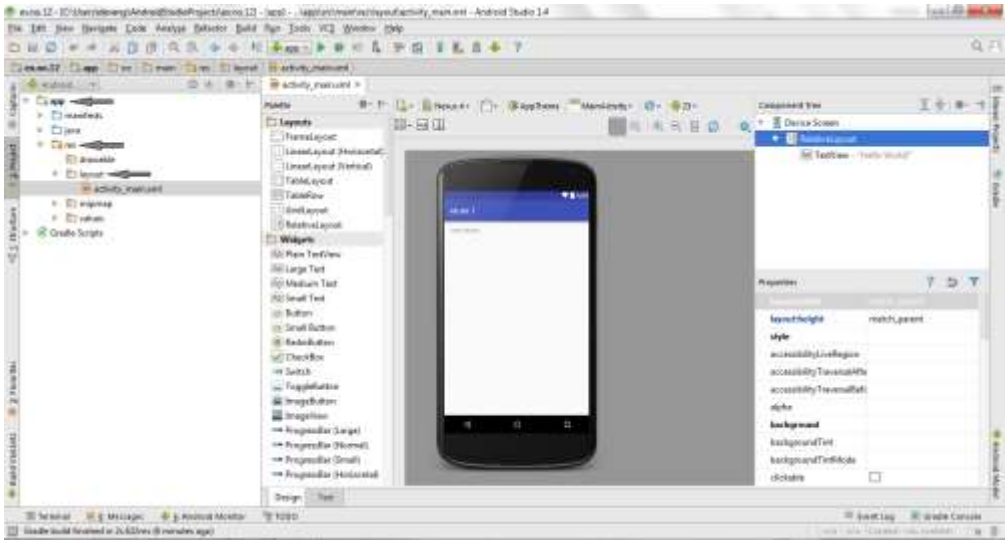


9. Type the Activity Name as **SecondActivity** and click **Finish** button.

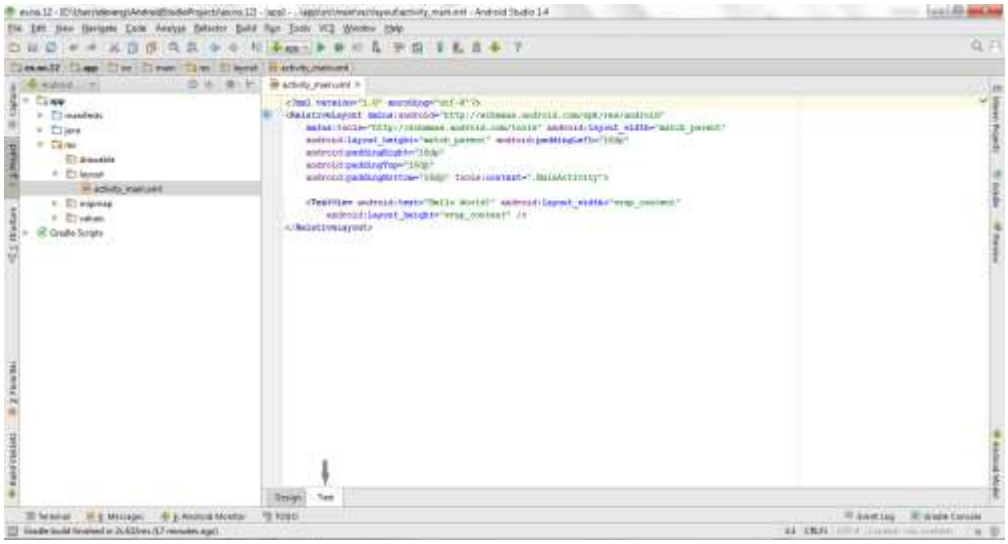


10. Thus Second Activity For the application is created.
11. Designing layout for the Android Application:

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



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



13. 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:id="@+id/relativeLayout1"
android:layout_width="match_parent"
android:layout_height="match_parent" >
<Button
android:id="@+id/show_Location
"
android:layout_width="wrap_cont
```

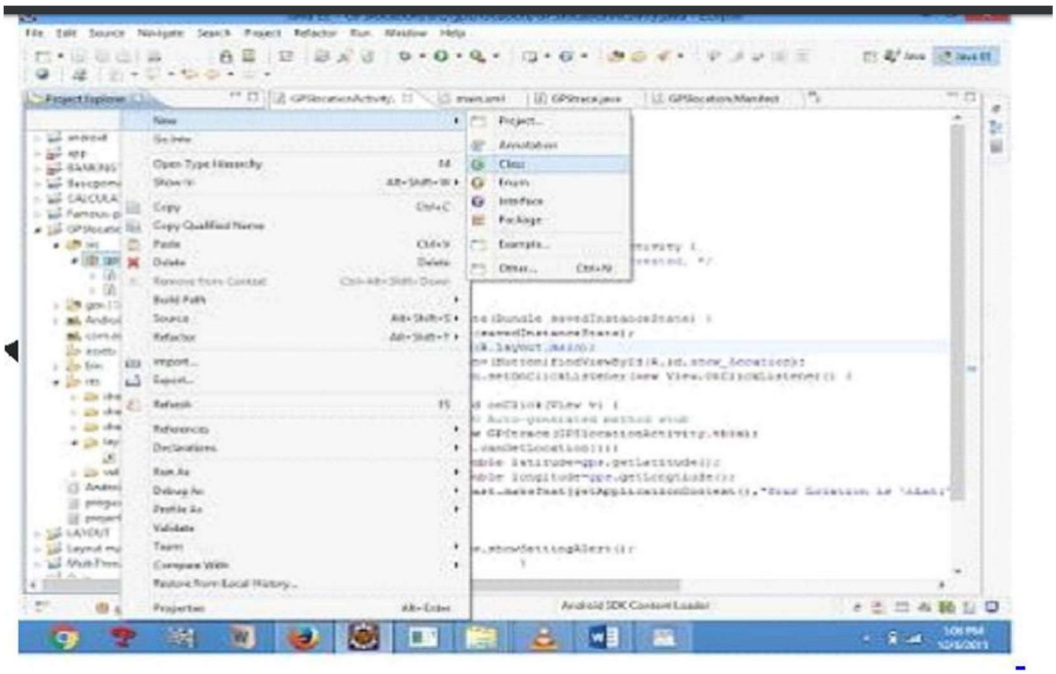
```
ent"
android:layout_height="wrap_cont
ent
android:text="Show_Location"
android:layout_centerVertical="tr
ue"android:layout_centerHorizontal="true"
/>
</RelativeLayout>
```

14. Now select mainactivity.java file and type the following code. In my coding man activity
15. name is GPSlocation Activity. Package gps.location;

```
//import android.R;
importandroid.app.Activity;
importandroid.os.Bundle;
importandroid.view.View;
importandroid.widget.Button;
importandroid.widget.Toast;
public class GPSlocationActivity extends Activity {
/** Called when the activity is first created. */
Button btnShowLocation;
GPStracegps;
@Override
public void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.main);
btnShowLocation=(Button)findViewById(R.id.show_Location);
btnShowLocation.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
// TODO Auto-generated method stub
gps=new GPStrace(GPSlocationActivity.this);
if(gps.canGetLocation()){
double latitude=gps.getLatitude();
double longitude=gps.getLongtiude();
Toast.makeText(getApplicationContext(),"Your Location is
\nLat:"+latitude+"\nLong:"+longitude, Toast.LENGTH_LONG).show();
}
else
{
gps.showSettingAlert();
}
}
```

});
}
}

16. Go to src folder and Right Click on your package folder and choose new class and give theclass names as GPS trace



17. Select the GPStrace.java file and paste the following code.

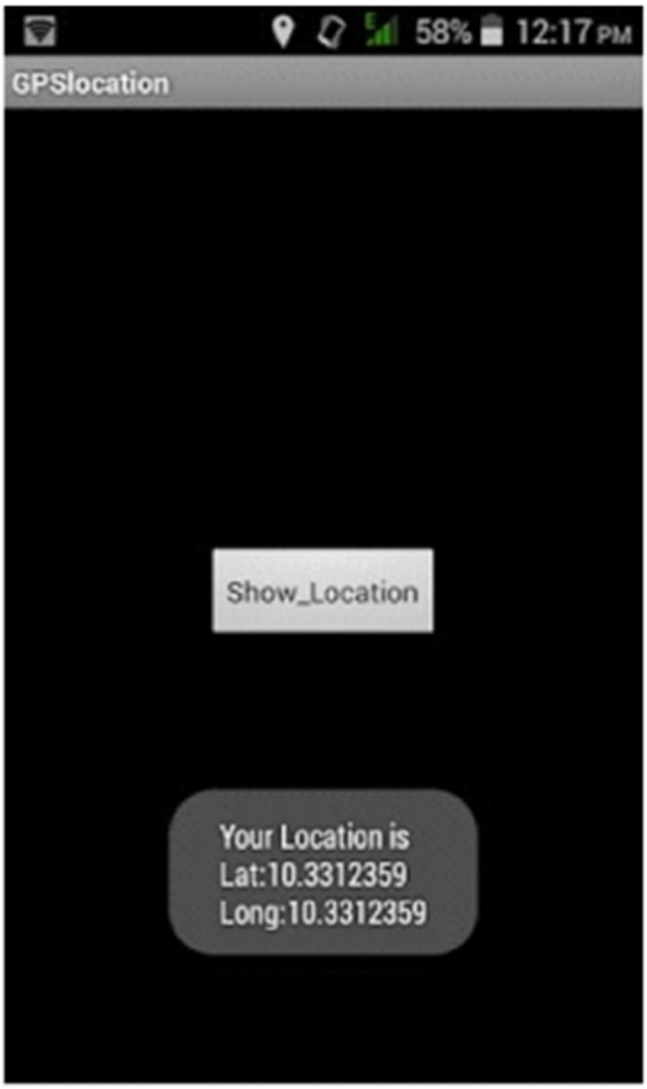
```
Package gps.location;
import android.app.AlertDialog;
import android.app.Service;
import android.content.Context;
import
android.content.DialogInterface;
import android.content.Intent;
import android.location.Location;
import
android.location.LocationListener
; import
android.location.LocationManage
r; import android.os.Bundle;
import android.os.IBinder; import
android.provider.Settings;
public class GPStrace extends Service implements
LocationListener{ private final Context context; Boolean
```

```
isGPSEnabled=false; booleancanGetLocation=false; boolean
isNetworkEnabled=false;
Location
location; double
latitude; double
longtitude;
private static final long MIN_DISTANCE_CHANGE_FOR_UPDATES=10;
private static final long MIN_TIME_BW_UPDATES=1000*60*1;
protected LocationManagerlocationManager;
public GPStrace(Context context){
this.context=conte
xt;
getLocation();
}
public Location getLocation()
{
try{
locationManager=(LocationManager)
context.getSystemService(LOCATION_SERVICE);
isGPSEnabled=locationManager.isProviderEnabled(LocationManager.GPS_PROVID
ER)
;
isNetworkEnabled=locationManager.isProviderEnabled(LocationManager.NETWOR
K_PR
OVI DER);
if(!isGPSEnabled&& !isNetworkEnabled){
}else{
this.canGetLocation=true;
if(isNetworkEnabled){
locationManager.requestLocationUpdates(
LocationManager.NETWORK_PROVIDER,
MIN_TIME_BW_UPDATES,
MIN_DISTANCE_CHANGE_FOR_UPDATES,this);
}
if(locationManager!=null){
location=locationManager.getLastKnownLocation(LocationManager.NETWORK_P
ROVI
DER)
;
if(location !=null){
latitude=location.getLatitude();
longtitude=location.getLongitude();
}
```

```
}
}
if(isGPSEnabled){
if(location==null){
locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER,MIN_
TIM
E_B
W_UPDATES, MIN_DISTANCE_CHANGE_FOR_UPDATES, this);
if(locationManager!=null){
location=locationManager.getLastKnownLocation(LocationManager.GPS_PROV
IDER); if(location!=null){
latitude=location.getLatitude();
longtitude=location.getLongitude();
}
}
}
}
}
catch(Exception e)
{
e.printStackTrace();
}
return location;
}
public void stopUsingGPS(){
if(locationManager!=null){
locationManager.removeUpdates(GPStrace.this);
} } public double
getLatitude(){
if(location!=null){
latitude=location.getLatitude();
}
return latitude;
}
public double getLongtiude(){
if(location!=null){
longtitude=location.getLatitude();
}
return longtitude;
}
public booleancanGetLocation(){
return this.canGetLocation;
```

```
}
public void showSettingAlert(){
AlertDialog.BuilderalertDialog=new AlertDialog.Builder(context);
alertDialog.setTitle("GPS is settings"); alertDialog.setMessage("GPS is not
enabled.Do you want to go to setting menu?");
alertDialog.setPositiveButton("settings", new DialogInterface.OnClickListener() {
@Override
public void onClick(DialogInterfacedialog,int which){
Intent intent=new Intent(Settings.ACTION_LOCATION_SOURCE_SETTINGS);
context.startActivity(intent);
}
});
alertDialog.setNegativeButton("cancel", new DialogInterface.OnClickListener() {
@Override
public void onClick(DialogInterface dialog, int which) {
// TODO Auto-generated method stub
dialog.cancel();
}
});
alertDialog.show();
}
@Override
public void onLocationChanged(Location location) {
// TODO Auto-generated method stub
}
@Override
public void onProviderDisabled(String provider) {
// TODO Auto-generated method stub
}
@Override
public void onProviderEnabled(String provider) {
// TODO Auto-generated method stub
}
@Override
public void onStatusChanged(String provider, int status, Bundle extras) {
// TODO Auto-generated method stub
}
@Override
public IBinderonBind(Intent
intent) { // TODO Auto-generated
method stub return null;
}
}
```

Output:


58% 12:17 PM
GPSlocation
Show_Location
Your Location is
Lat:10.3312359
Long:10.3312359

Go to manifest.xml file and add the code below

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

18. Now go to main.xml and right click .select run as option and select runconfiguration
19. Android output is present in the android emulator as shown in below.

Observation	25	
Record	10	
Total	35	
Signature		

Result:

Thus Android Application that a native is uses gps location information is developed and executed successfully.

Ex.No: 9	IMPLEMENT AN APPLICATION THAT WRITES DATA TO THE SD CARD.
Date:	

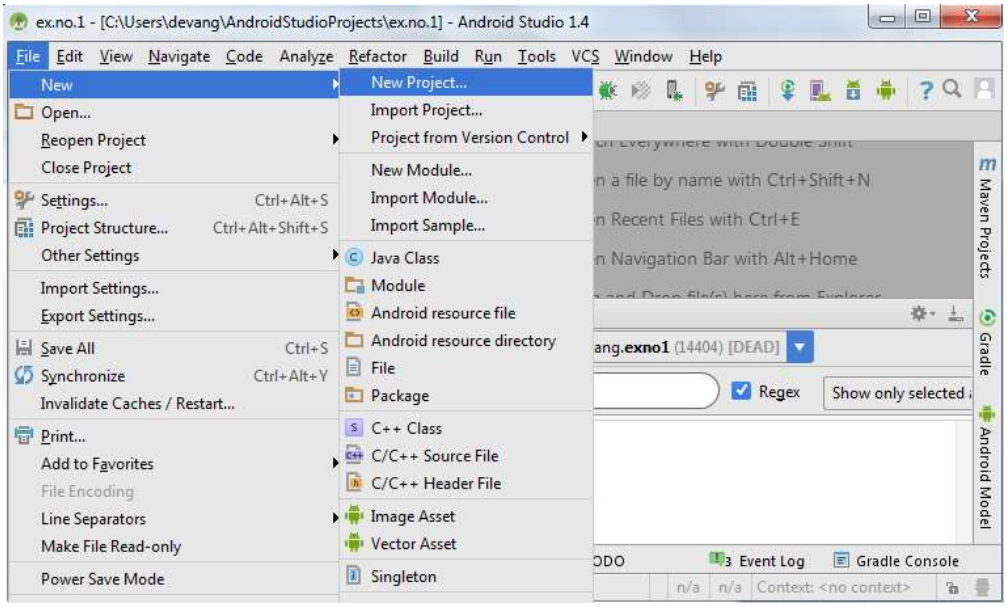
Aim:

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

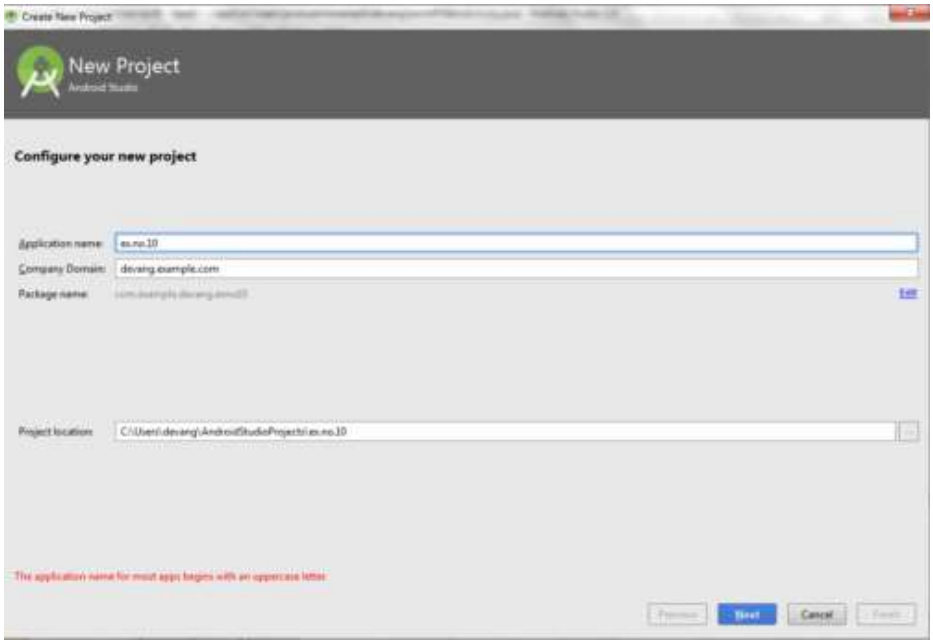
Procedure:

Creating a New project:

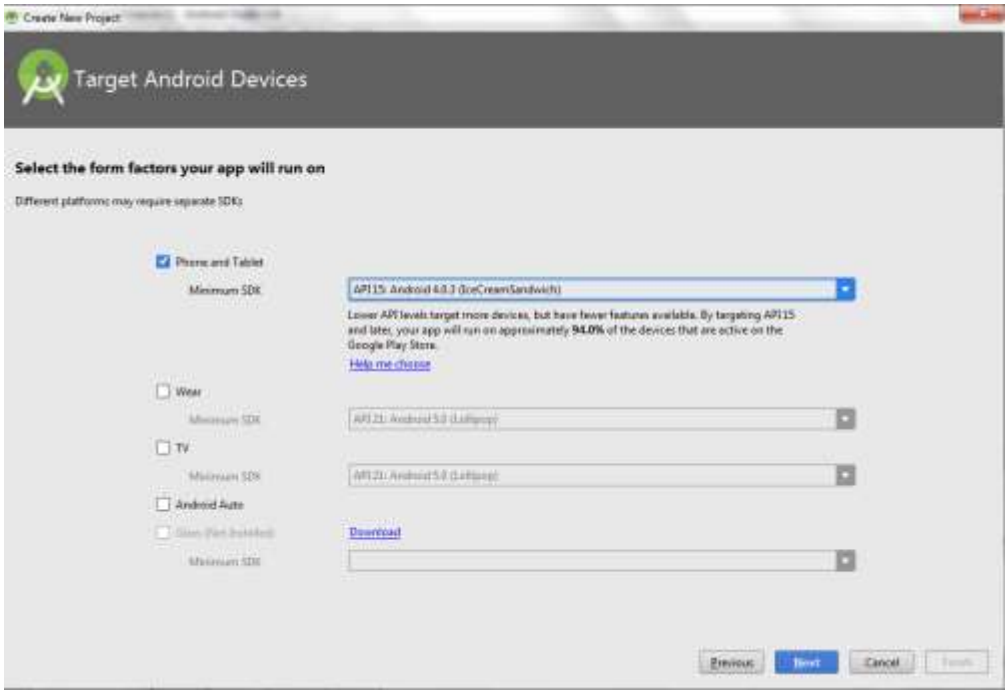
1. Open Android Studio and then click on **File -> New -> New project.**



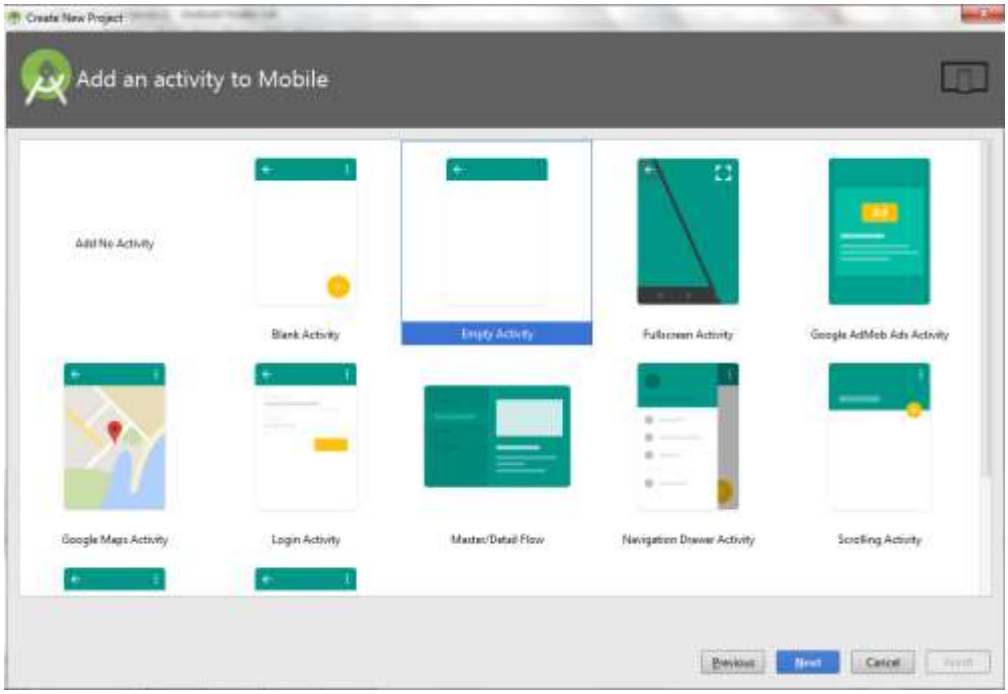
2. Then type the Application name as "**ex.no.10**" and click **Next.**



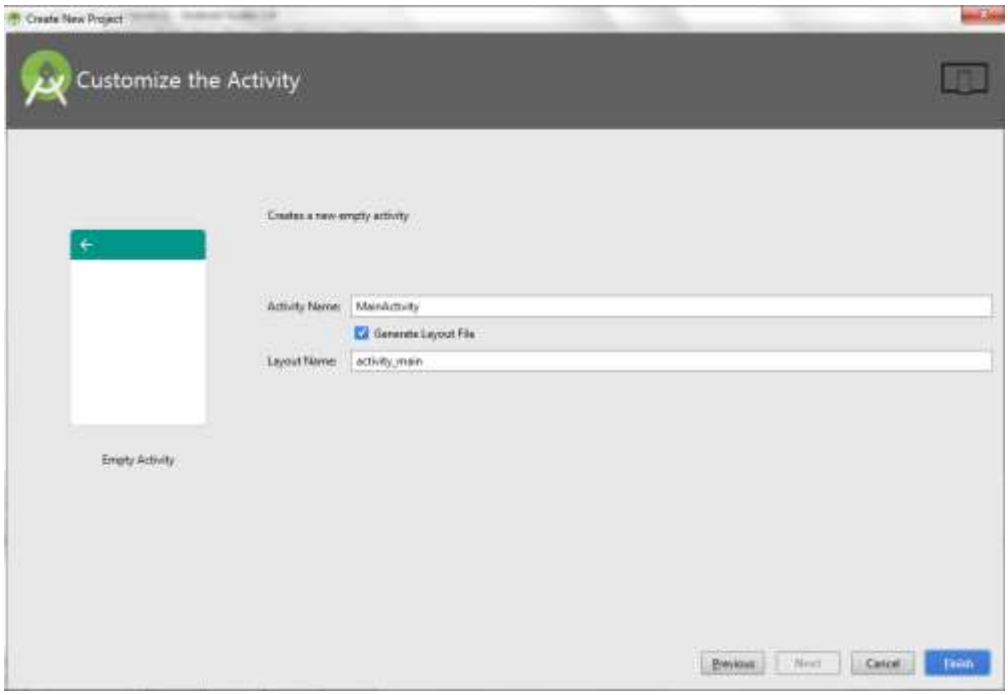
3. Then select the **Minimum SDK** as shown below and click **Next**.



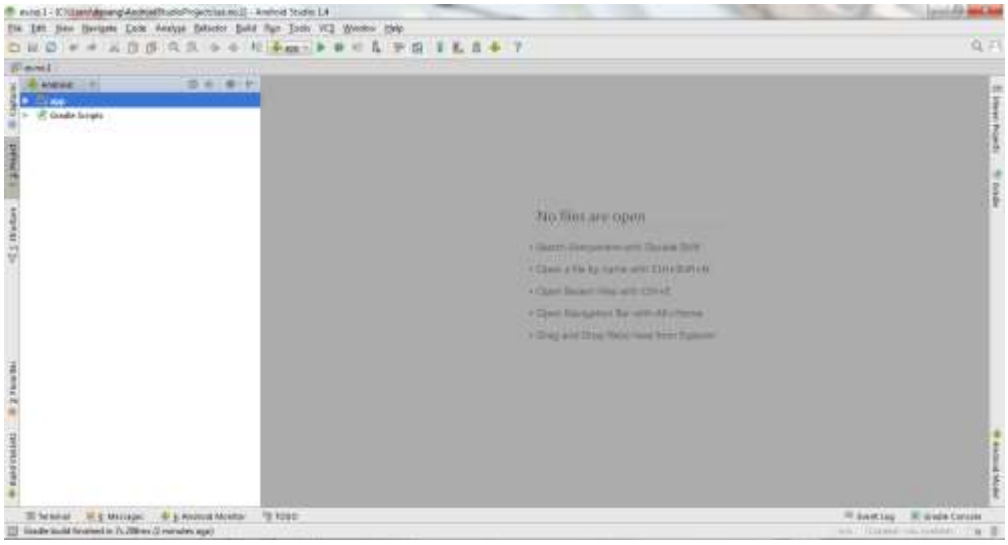
4. Then select the **Empty Activity** and click **Next.**



5. Finally click F**inish**.

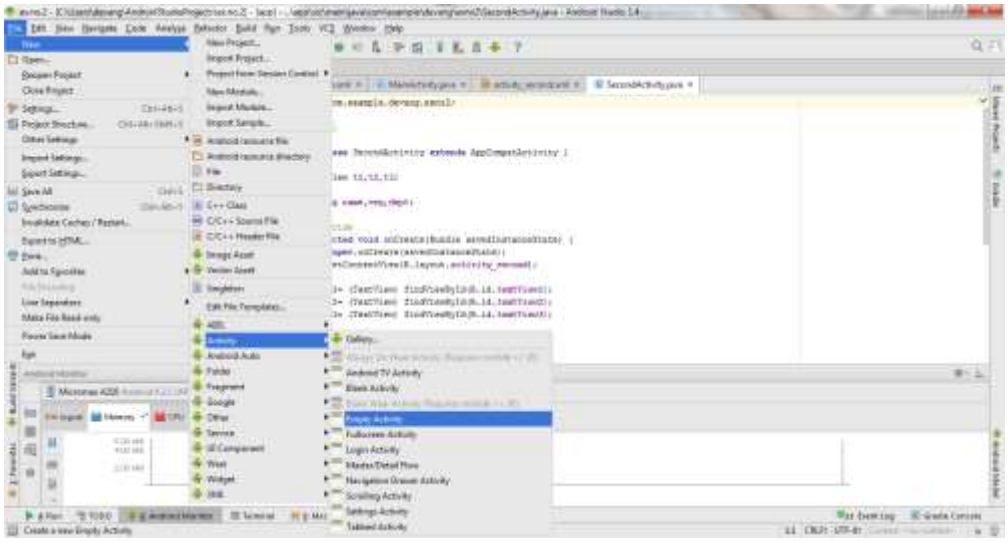


6. It will take some time to build and load the project.
7. After completion it will look as given below.

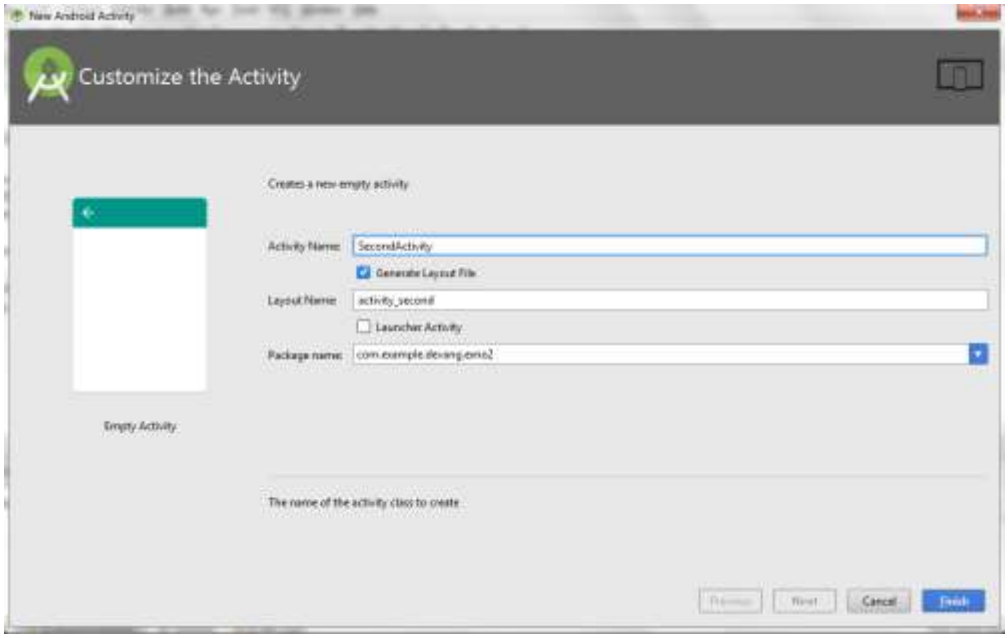


Creating Second Activity for the Android Application:

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



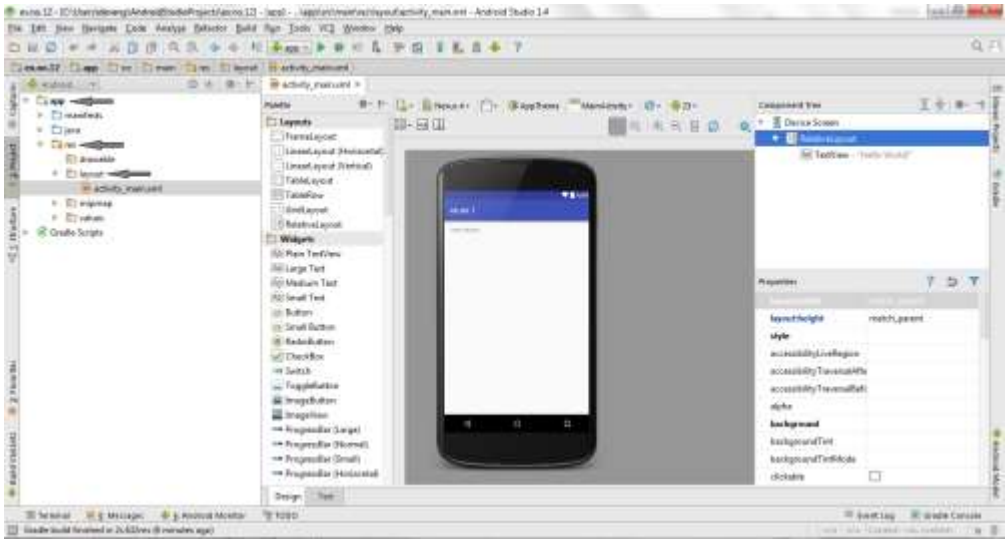
9. Type the Activity Name as **SecondActivity** and click **Finish** button.



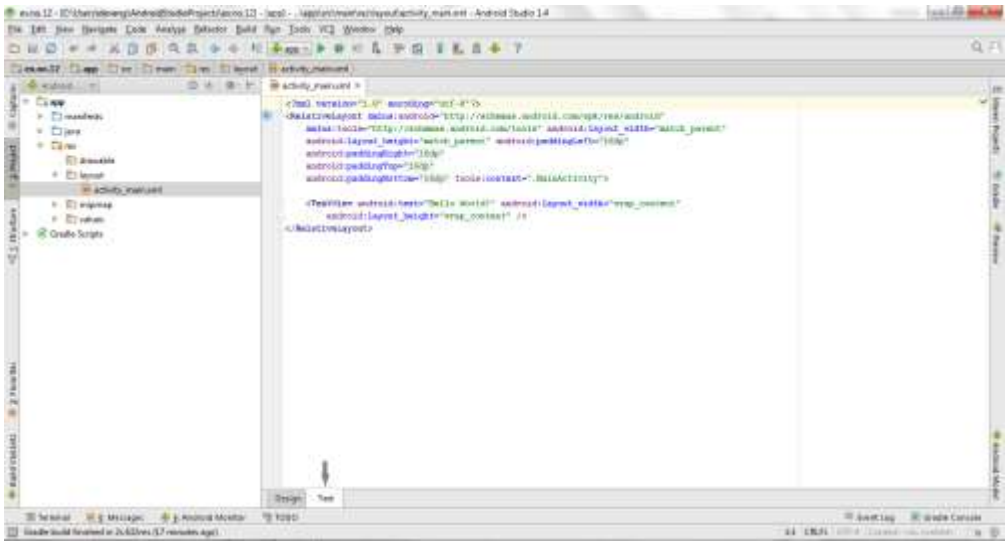
10. Thus Second Activity For the application is created.

Designing layout for the Android Application:

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



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



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

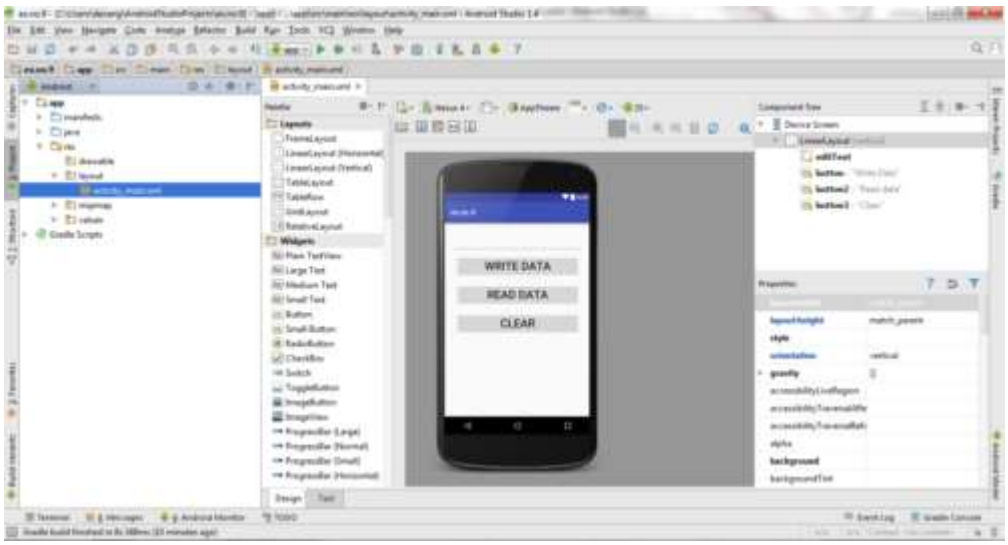
Code for Activity_main.xml:

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

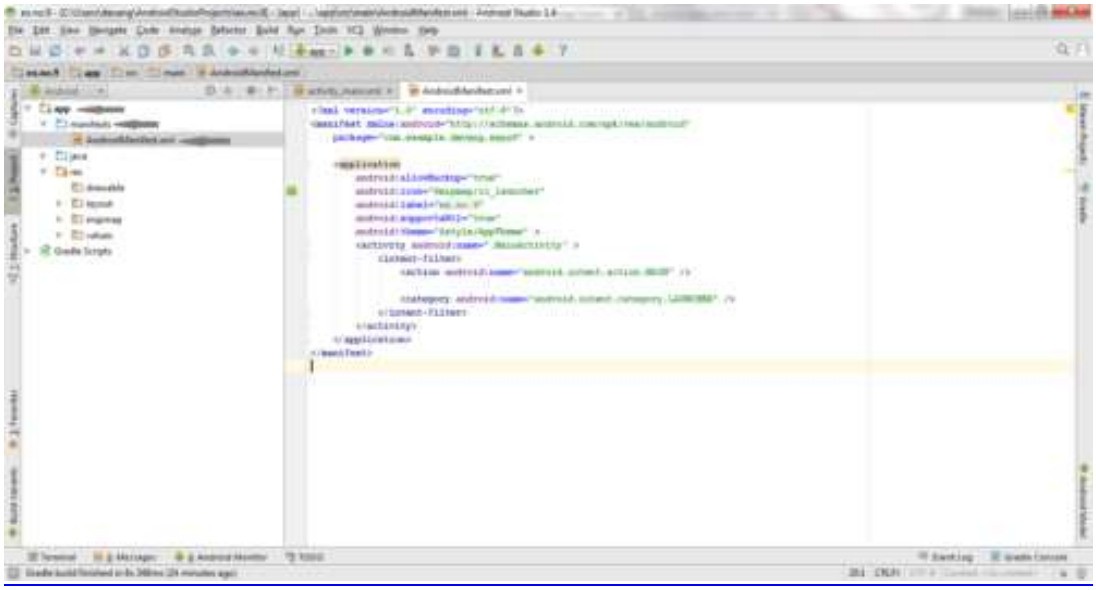
14. Now click on **Design** and your application will look as given below.



15. So now the designing part is completed.

Adding permissions in Manifest for the Android Application:

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



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

Code for AndroidManifest.xml:

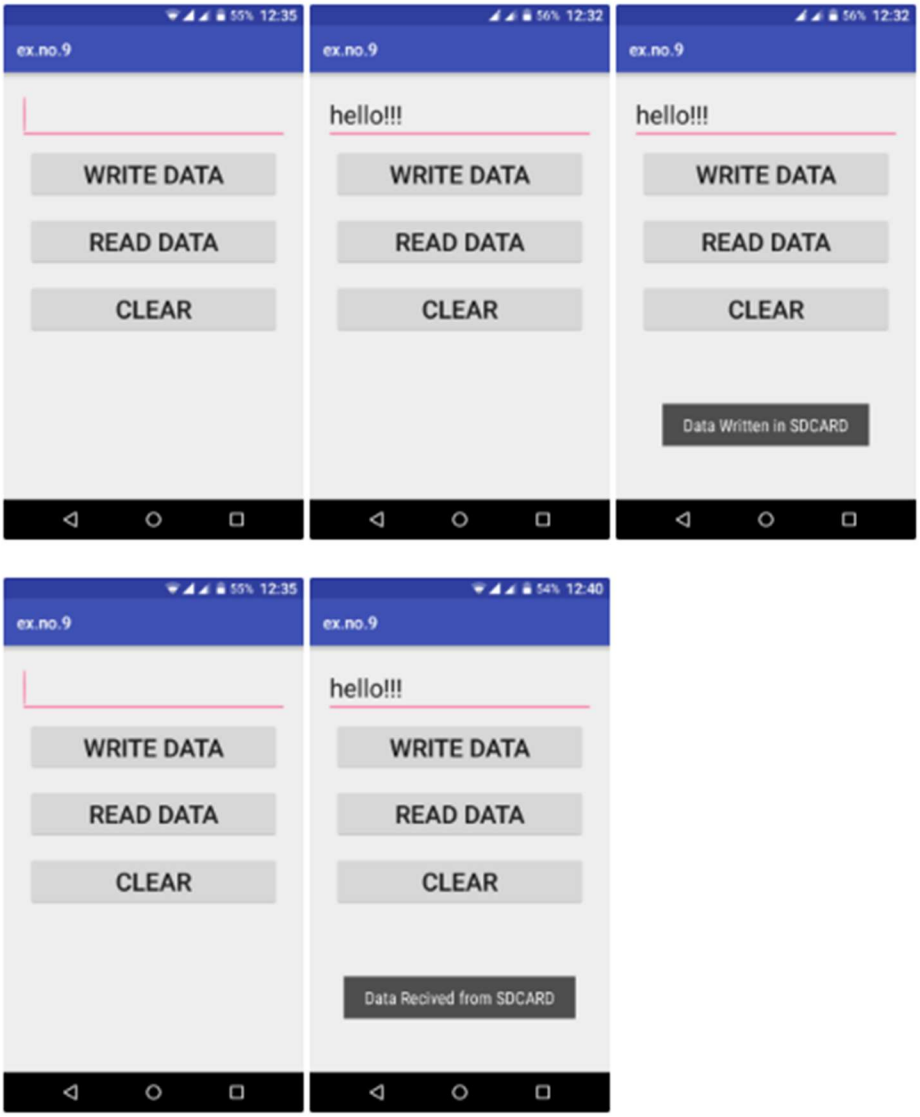
```
package com.example.exno9;

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();
               FileOutputStreamfout=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);
                BufferedReaderbr = 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:


55% 12:35
ex.no.9
WRITE DATA
READ DATA
CLEAR
56% 12:32
ex.no.9
hello!!!
WRITE DATA
READ DATA
CLEAR
56% 12:32
ex.no.9
hello!!!
WRITE DATA
READ DATA
CLEAR
Data Written in SDCARD
55% 12:35
ex.no.9
WRITE DATA
READ DATA
CLEAR
54% 12:40
ex.no.9
hello!!!
WRITE DATA
READ DATA
CLEAR
Data Recived from SDCARD

```
    });
  }
}
```

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

Observation	25	
Record	10	
Total	35	
Signature		

Result:

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

Ex.No: 10	IMPLEMENT AN APPLICATION THAT CREATES AN ALERT UPON RECEIVING A MESSAGE.
Date:	

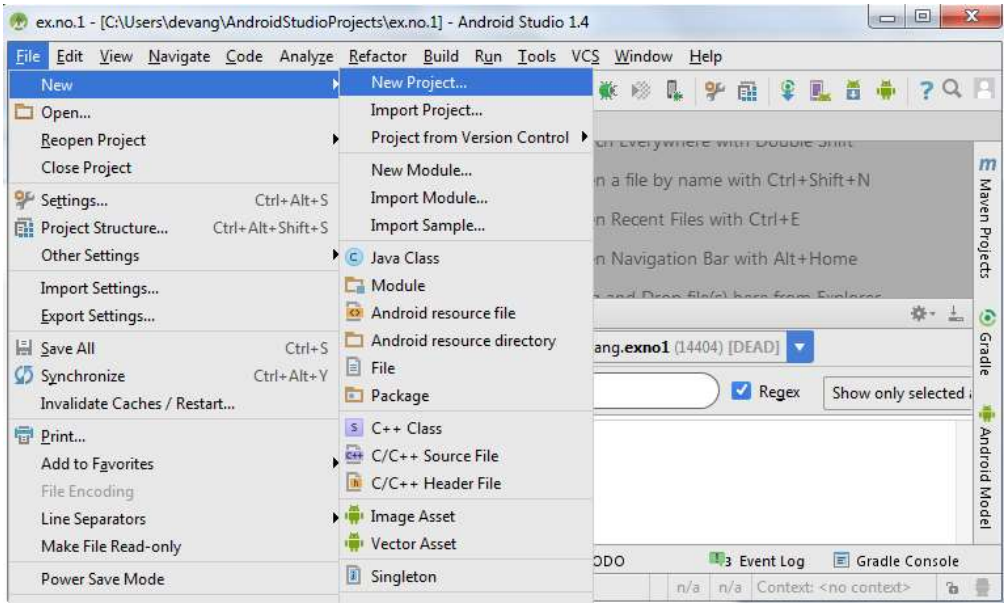
Aim:

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

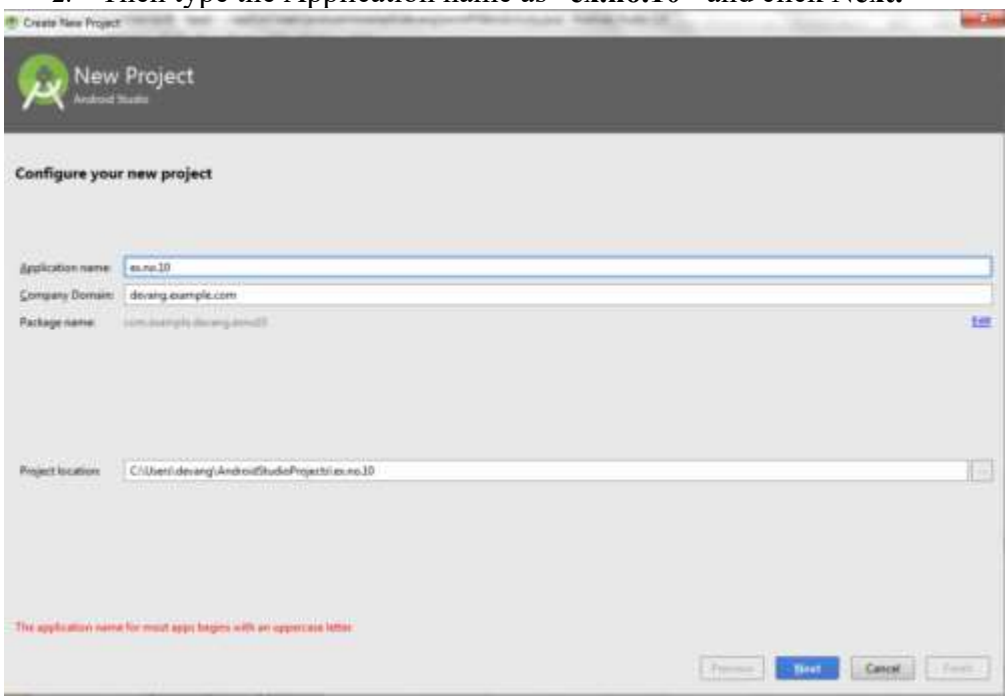
Procedure:

Creating a New project:

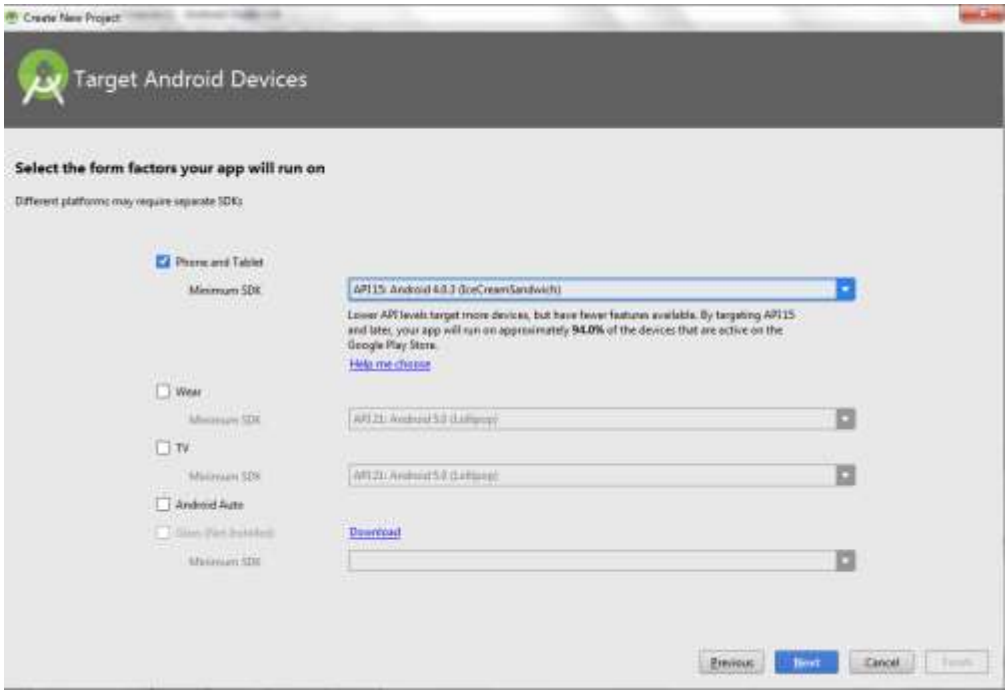
1. Open Android Studio and then click on **File -> New -> New project.**



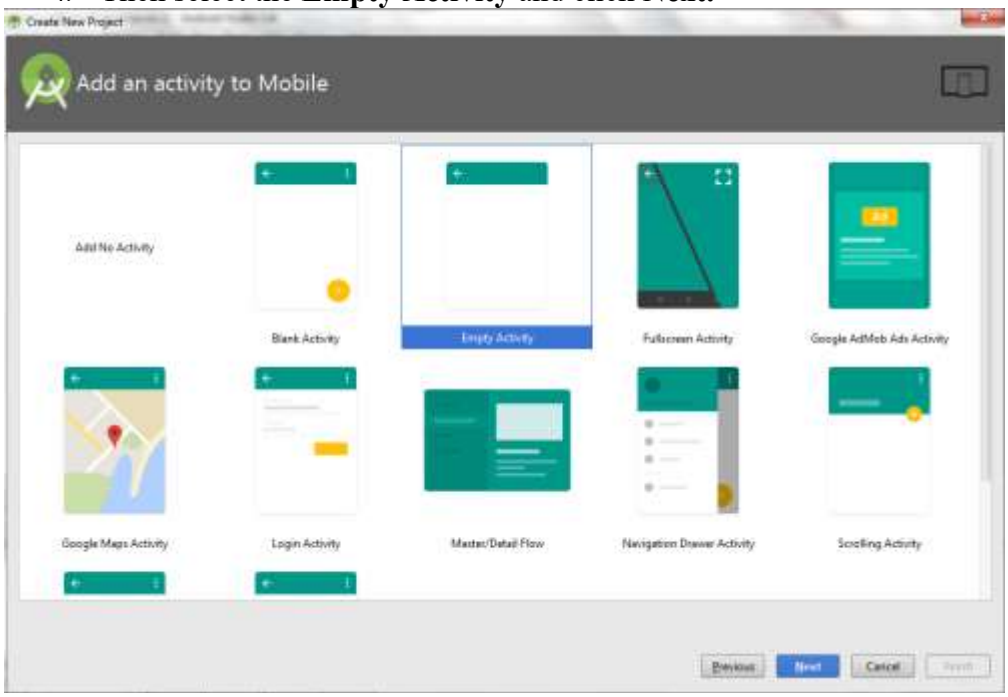
2. Then type the Application name as "**ex.no.10"** and click **Next.**



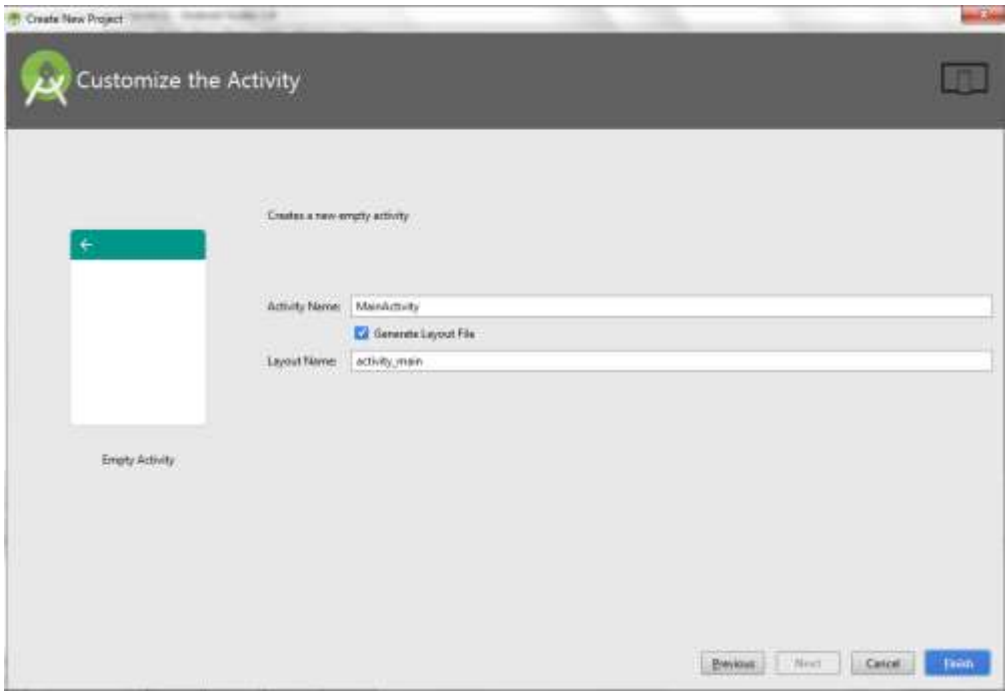
3. Then select the **Minimum SDK** as shown below and click **Next**.



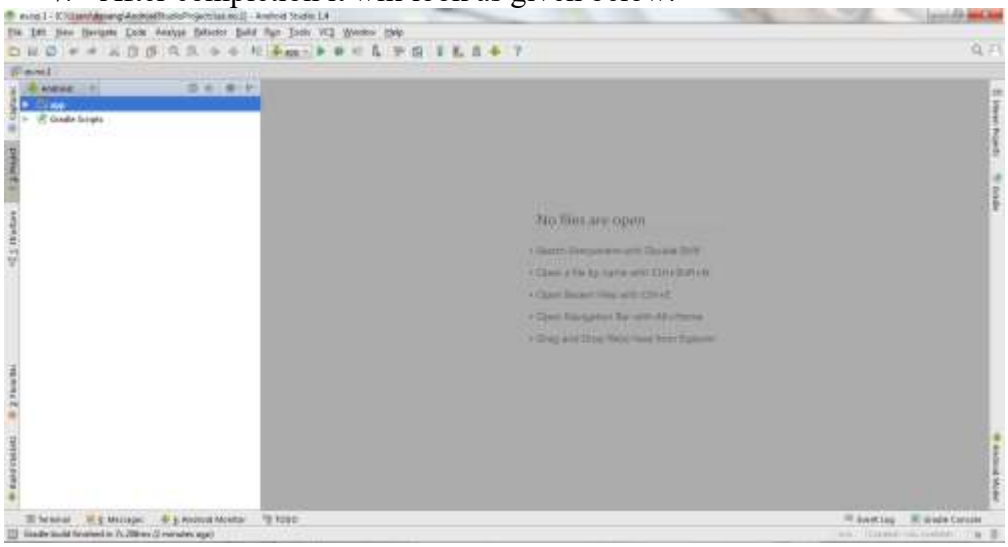
4. Then select the **Empty Activity** and click **Next.**



5. Finally click F**inish**.

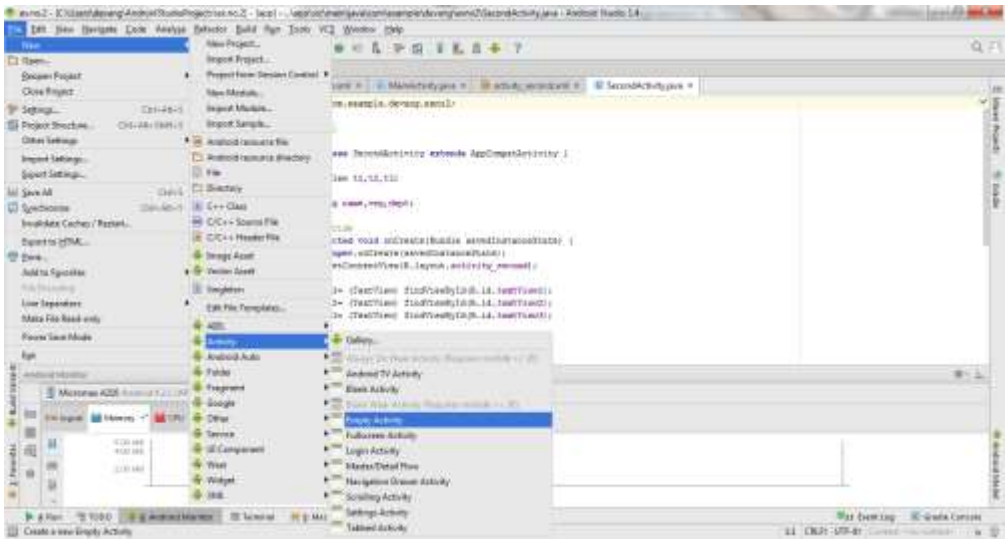


6. It will take some time to build and load the project.
7. After completion it will look as given below.

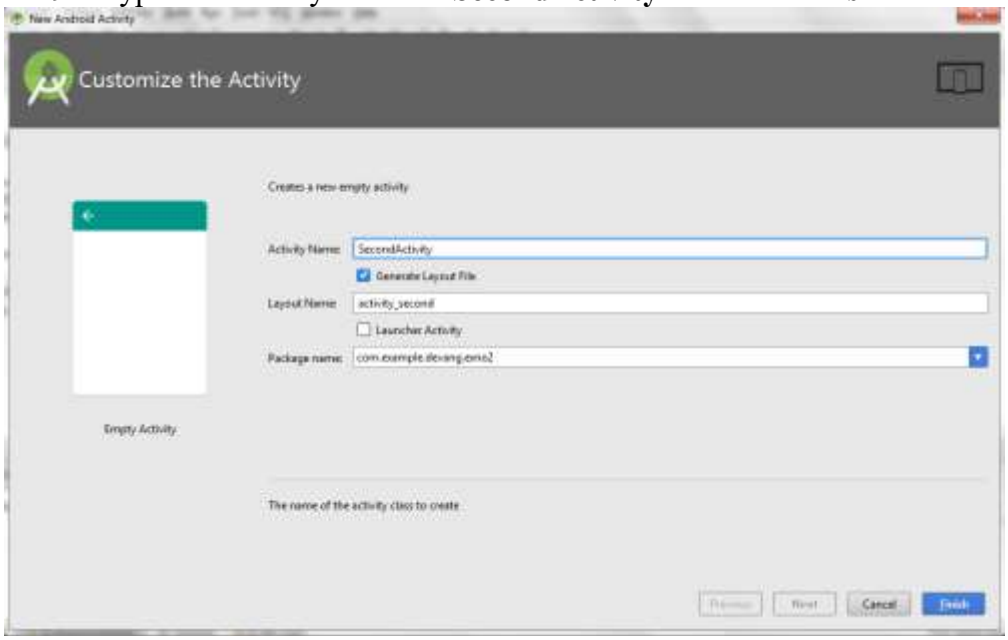


Creating Second Activity for the Android Application:

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



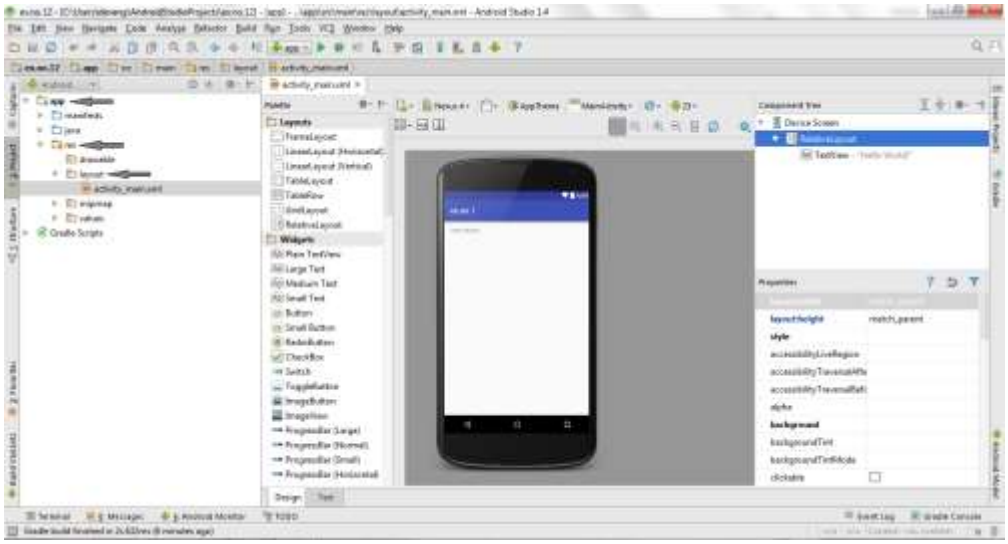
9. Type the Activity Name as **SecondActivity** and click **Finish** button.



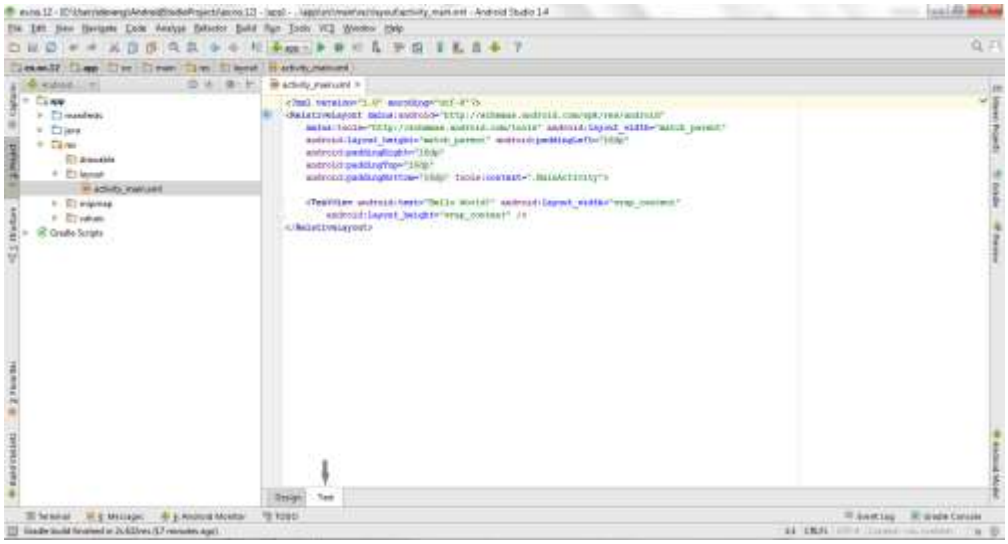
10. Thus Second Activity For the application is created.

Designing layout for the Android Application:

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



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



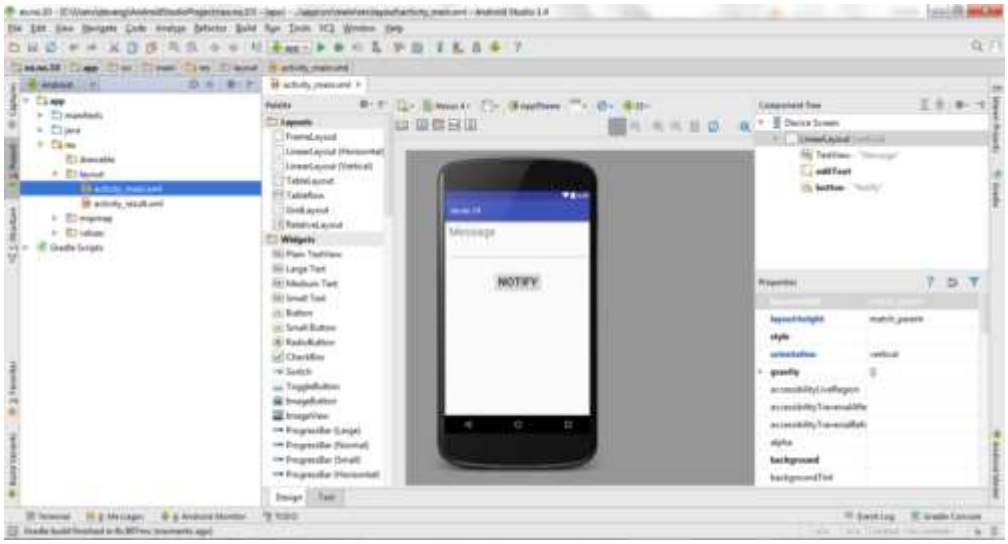
13. Then delete the code which is there and type the code as given below.

Code for Activity_main.xml:

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

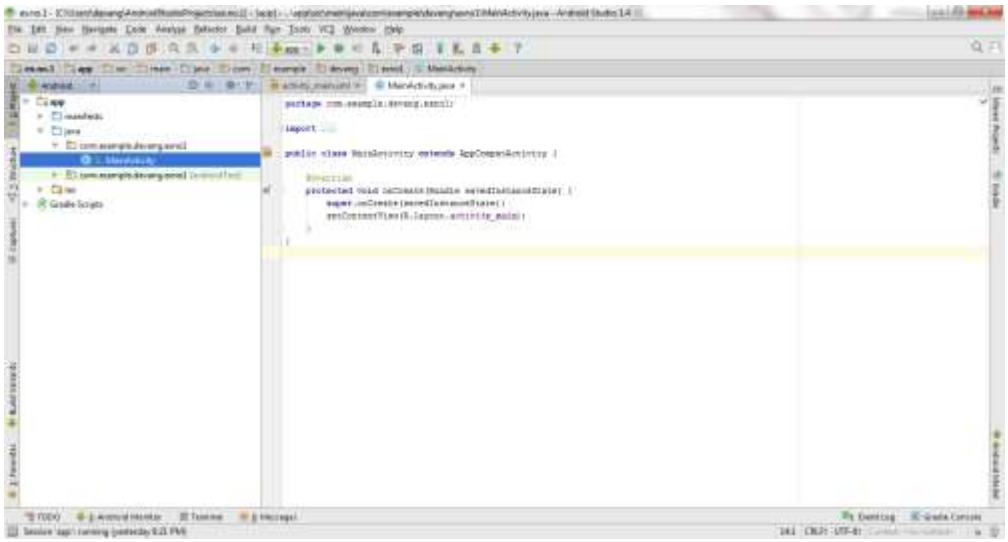
14. Now click on **Design** and your application will look as given below.



15. So now the designing part is completed.

Java Coding for the Android Application:

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

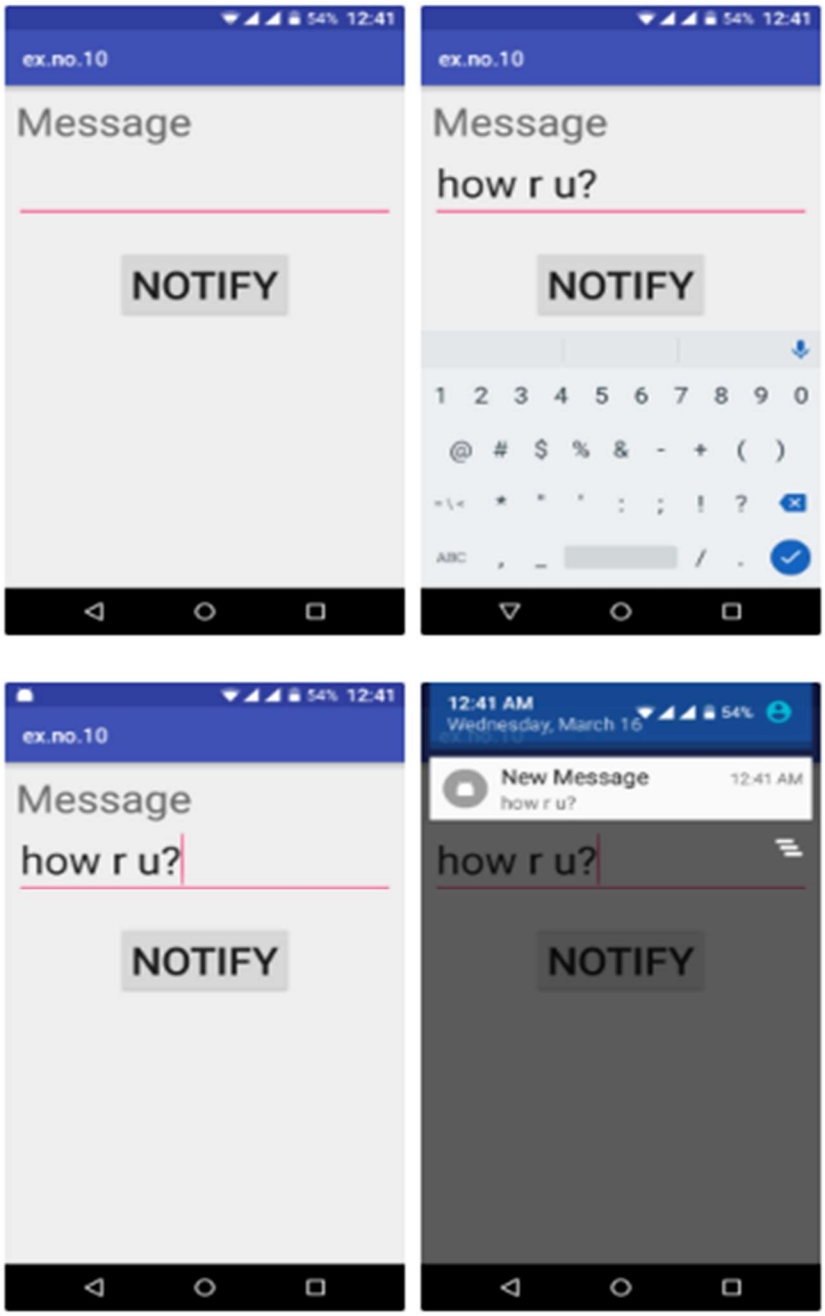


17. Then delete the code which is there and type the code as given below.

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()
```

Output:


ex.no.10
Message
NOTIFY
ex.no.10
Message
how r u?
NOTIFY
ex.no.10
Message
how r u?
NOTIFY
12:41 AM
Wednesday, March 16
New Message
how r u?
how r u?
NOTIFY

```
        {
            @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).setCo
ntentIntent(pending).build();
                NotificationManager manager = (NotificationManager)
getSystemService(NOTIFICATION_SERVICE);
                noti.flags |= Notification.FLAG_AUTO_CANCEL;
                manager.notify(0, noti);
            }
        });
    }
}
```

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

Observation	25	
Record	10	
Total	35	
Signature		

Result:

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

Ex.No: 11	WRITE A MOBILE APPLICATION THAT CREATES ALARM CLOCK.
Date:	

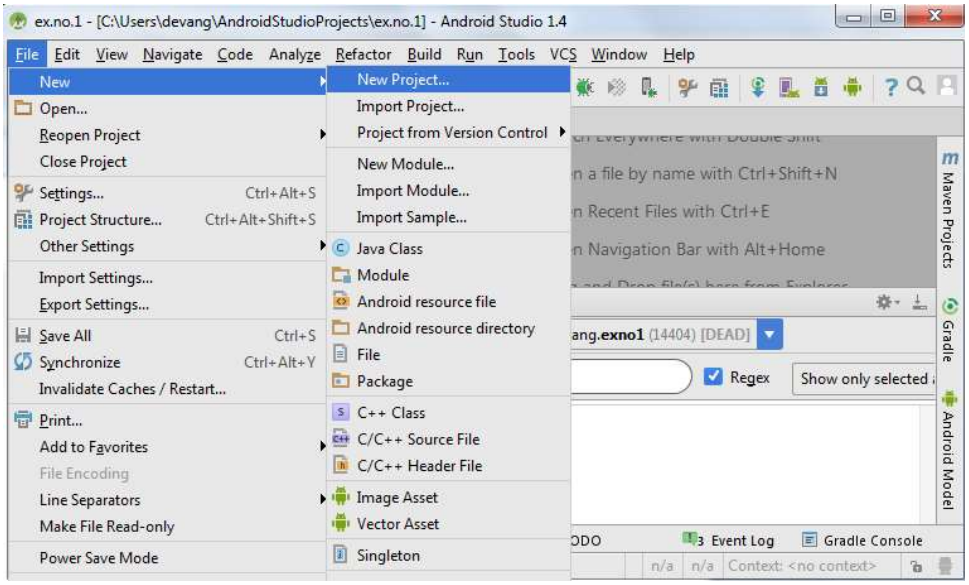
Aim:

To develop a Android Application that creates Alarm Clock.

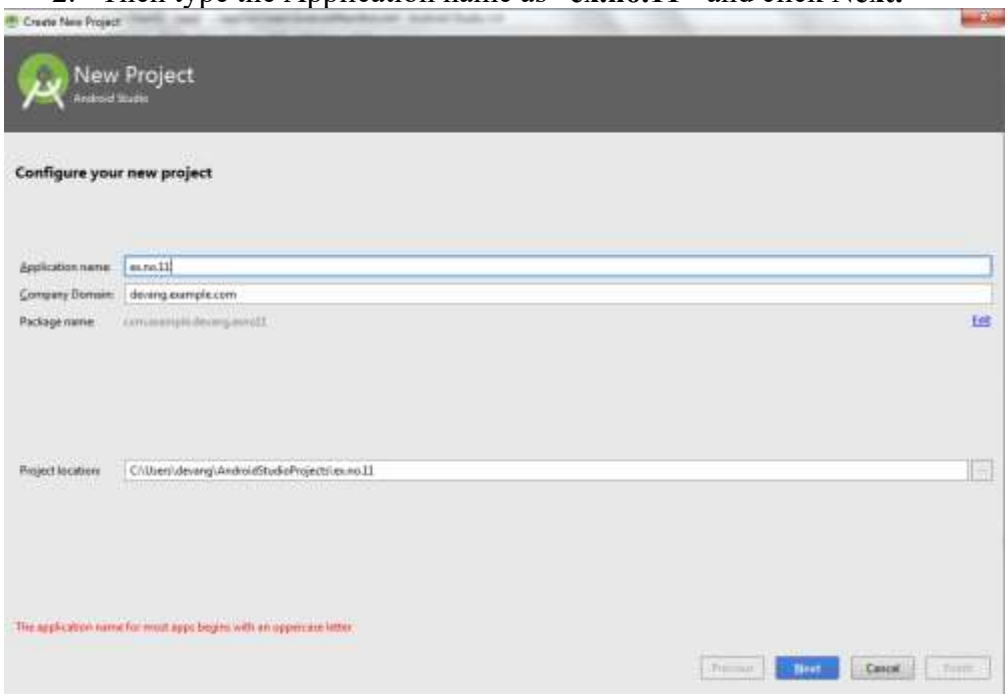
Procedure:

Creating a New project:

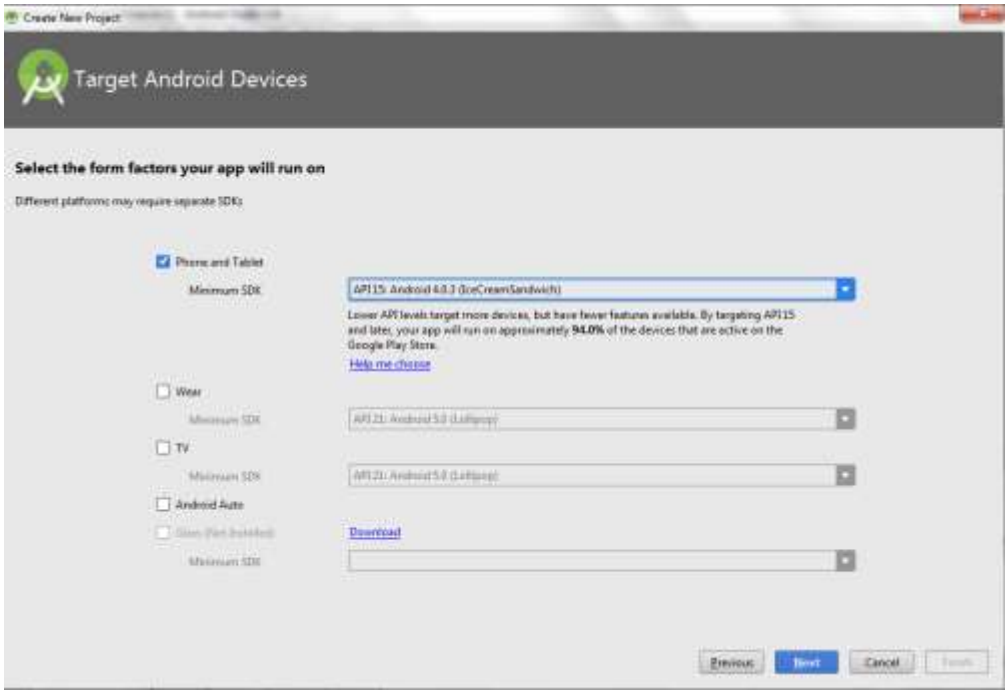
1. Open Android Studio and then click on **File -> New -> New project.**



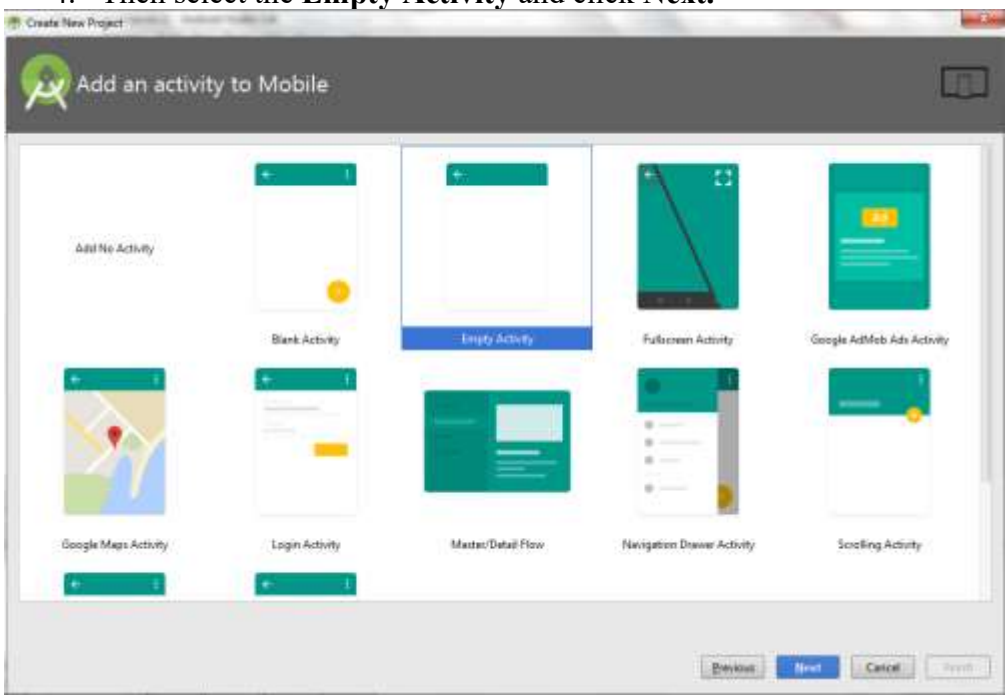
2. Then type the Application name as "**ex.no.11"** and click **Next.**



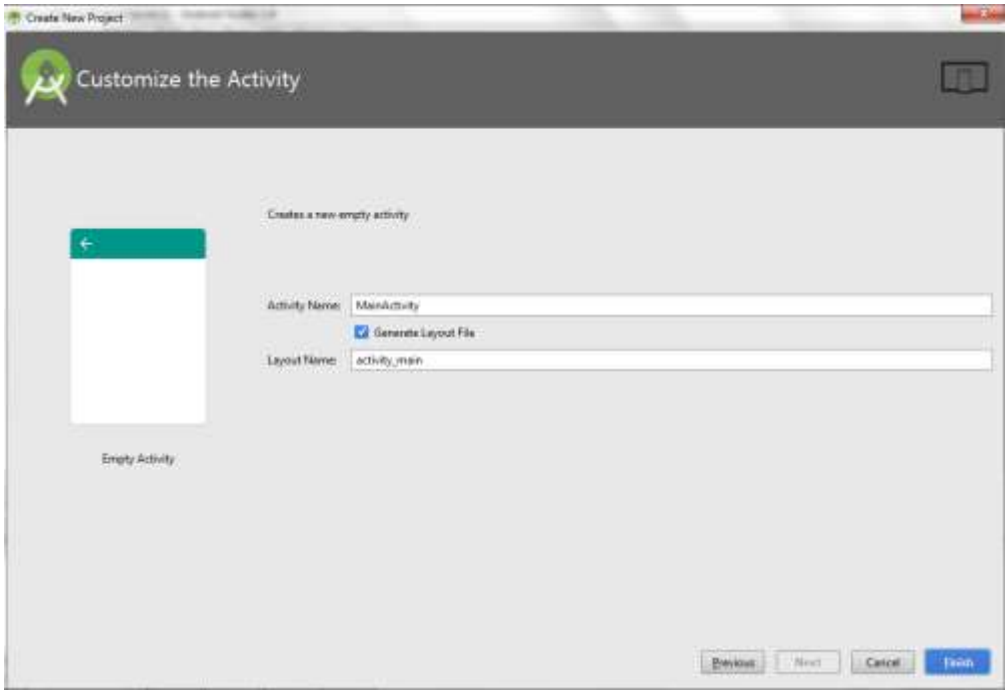
3. Then select the **Minimum SDK** as shown below and click **Next**.



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

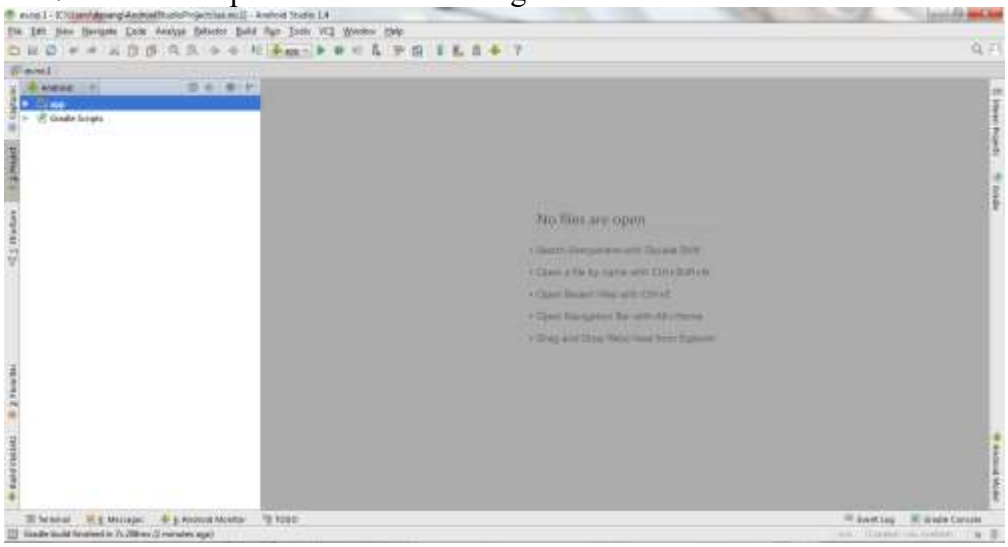


5. Finally click F**inish**.



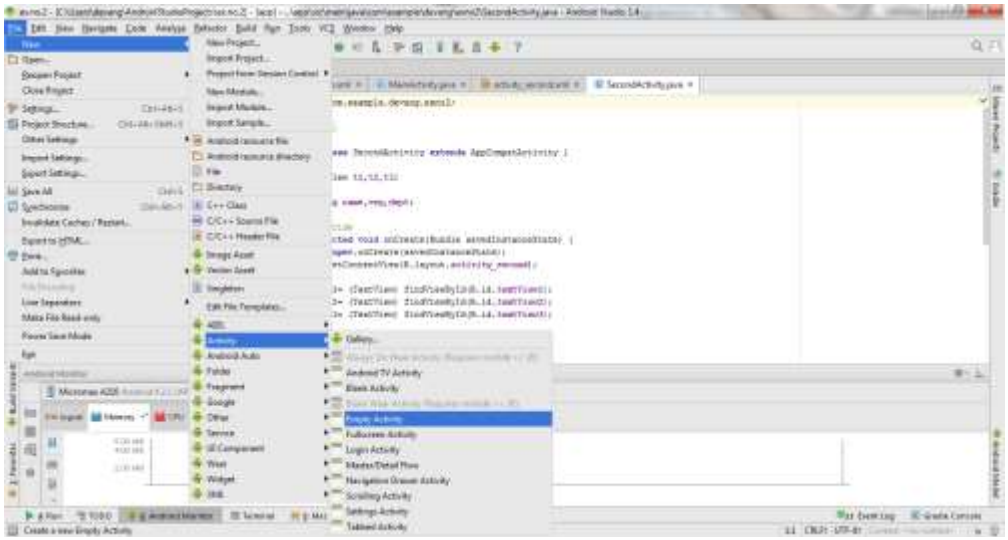
6. It will take some time to build and load the project.

7. After completion it will look as given below.

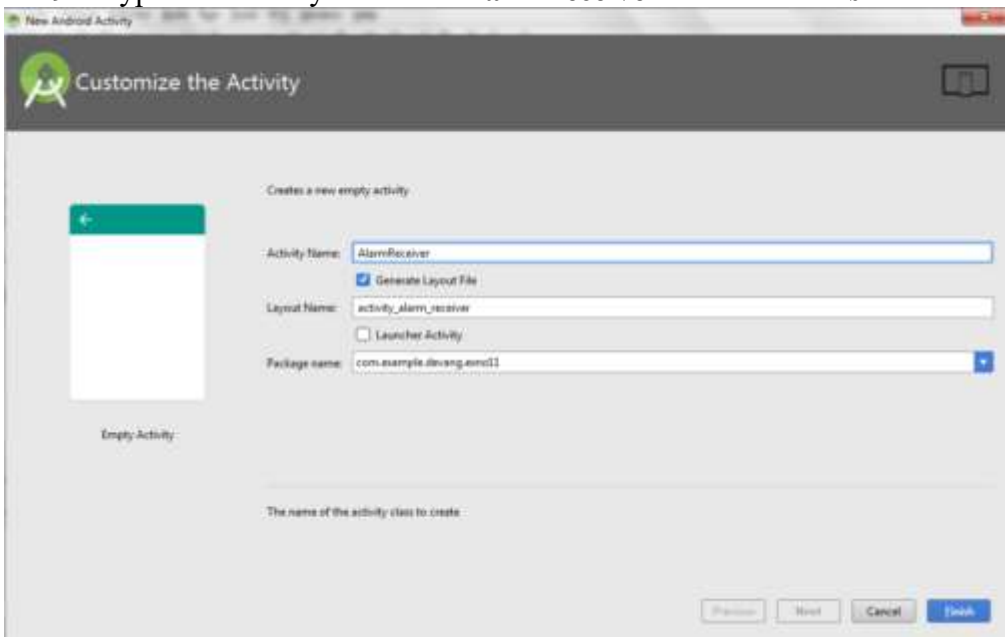


Creating Second Activity for the Android Application:

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



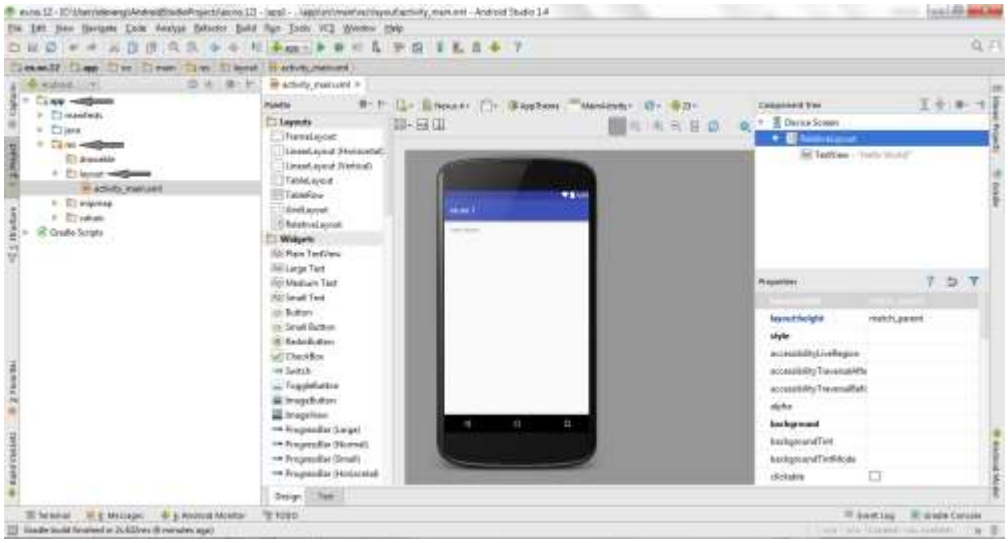
9. Type the Activity Name as **AlarmReceiver** and click **Finish** button.



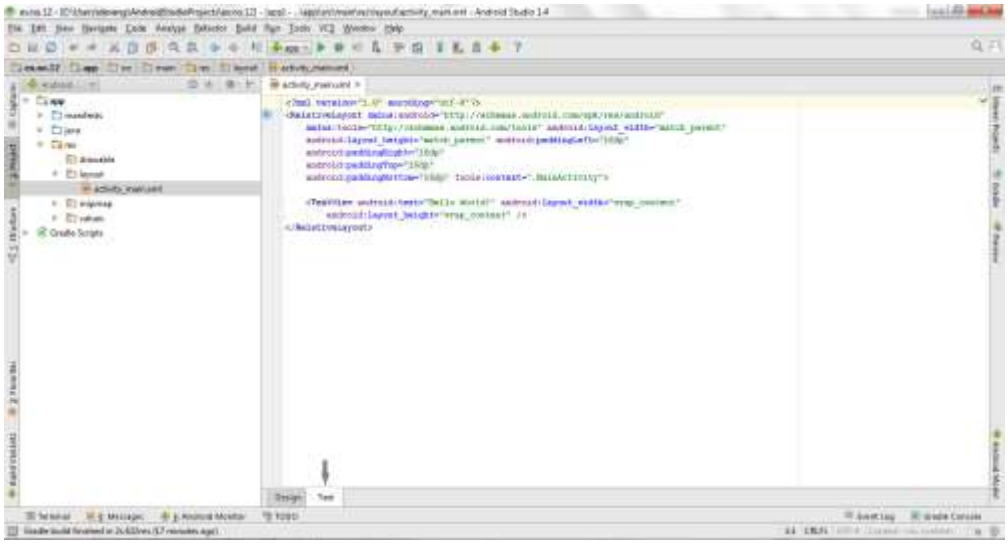
10. Thus Second Activity For the application is created.

Designing layout for the Android Application:

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



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



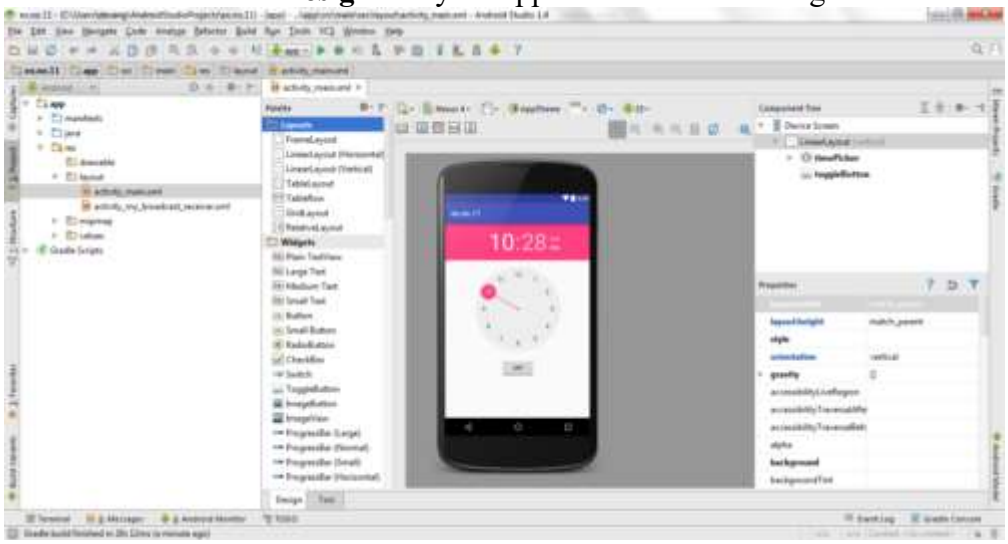
13. Then delete the code which is there and type the code as given below.

Code for Activity_main.xml:

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

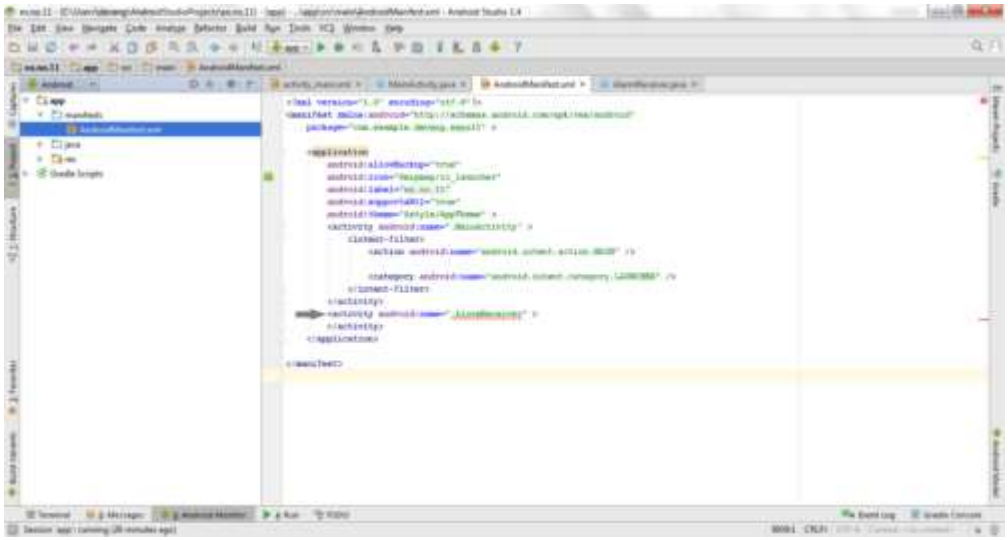
14. Now click on **Design** and your application will look as given below.



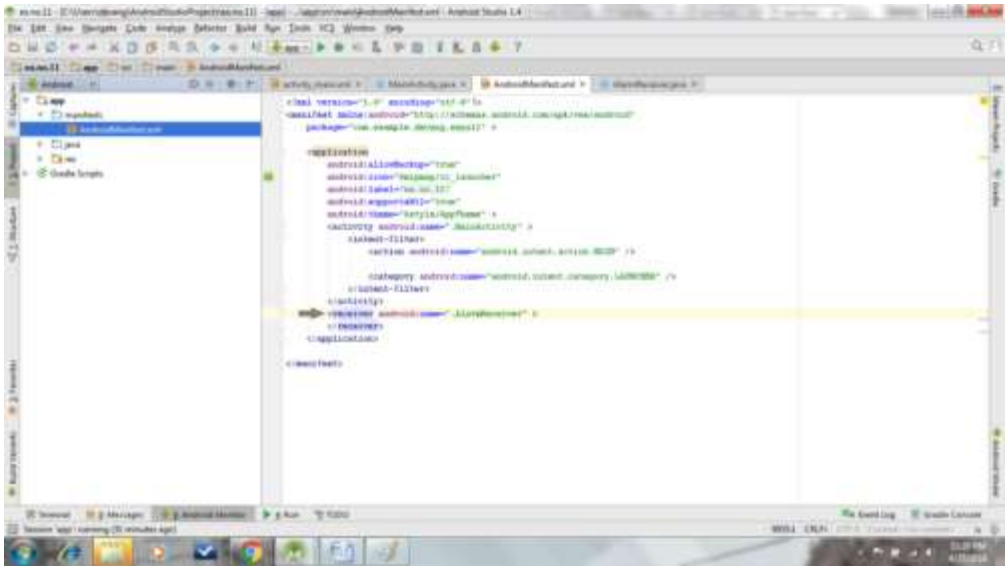
15. So now the designing part is completed.

Changes in Manifest for the Android Application:

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



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



Code for AndroidManifest.xml:

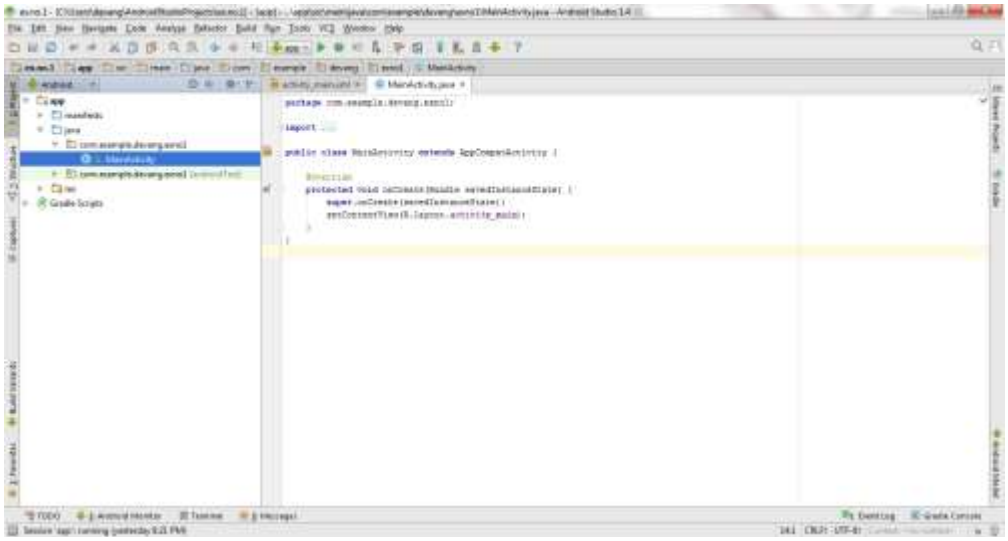
```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
   package="com.example.exno11" >

<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>
```

18. So now the changes are done in the Manifest.

Java Coding for the Android Application:

Java Coding for Main Activity:

19. Click on **app -> java -> com.example.exno11 ->MainActivity.**



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

Code for MainActivity.java:

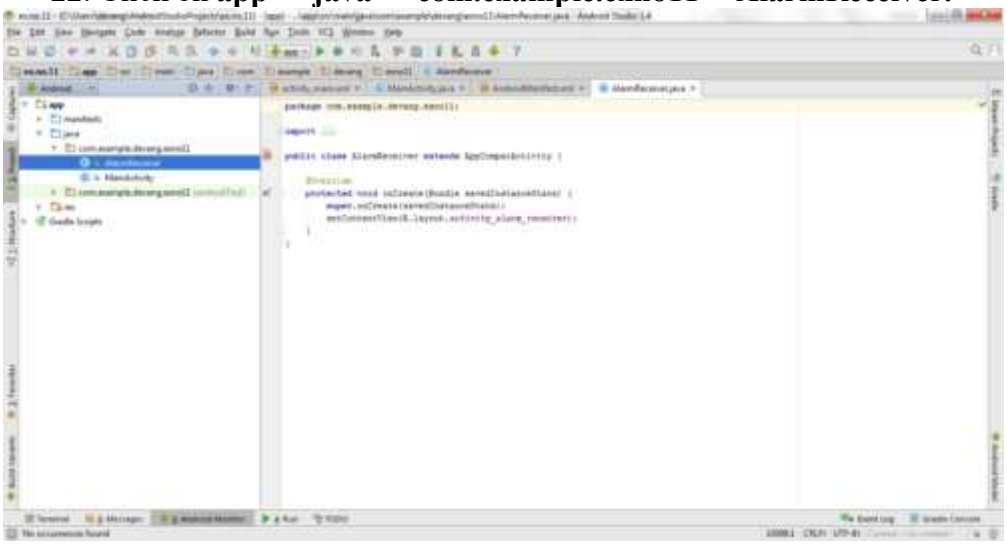
```
package com.example.exno11;
 import android.app.AlarmManager;
import android.app.PendingIntent;
import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.TimePicker;
import android.widget.Toast;
import android.widget.ToggleButton;
 import java.util.Calendar;
 public class MainActivity extends AppCompatActivity
{
   TimePickeralarmTimePicker;
   PendingIntentpendingIntent;
   AlarmManageralarmManager;

   @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();
      }  }}
```

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

Java Coding for Alarm Receiver:

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

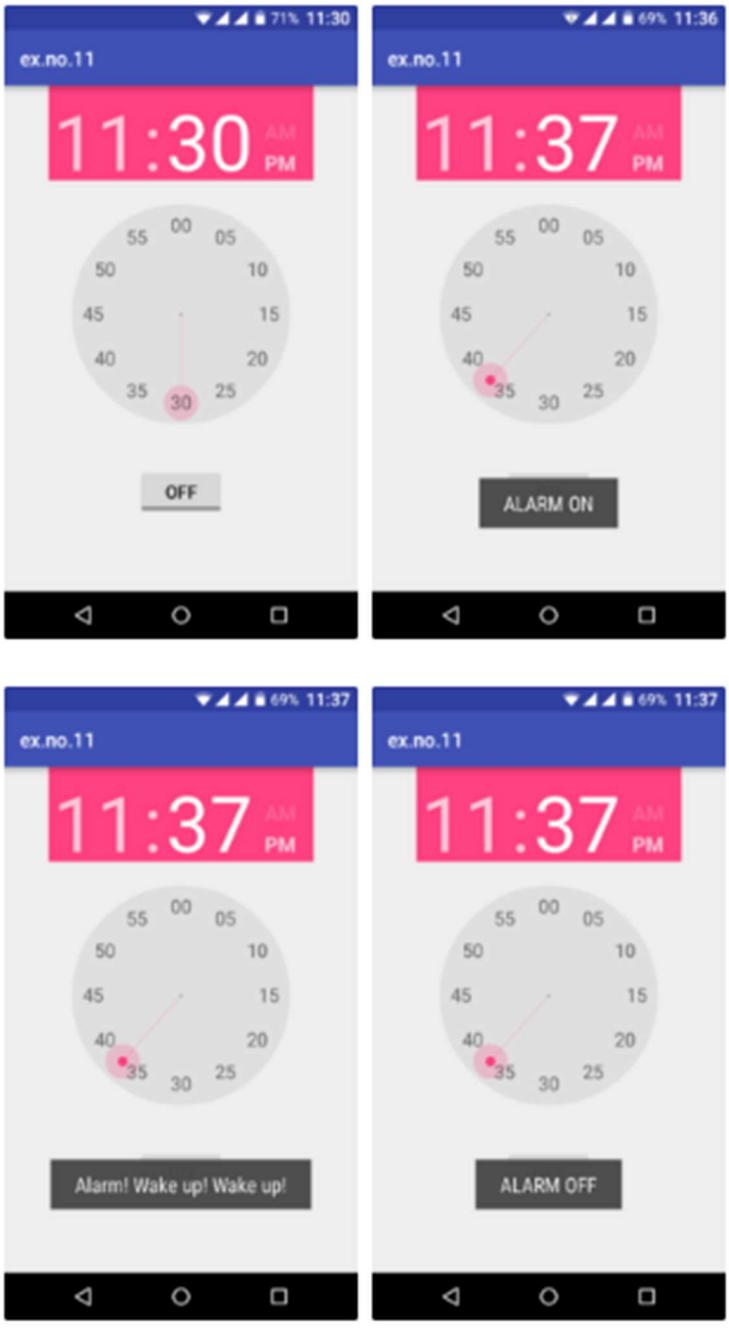


23. Then delete the code which is there and type the code as given below.

Code for AlarmReceiver.java:

```
package com.example.exno11;import android.content.BroadcastReceiver;
```

Output:


71% 11:30
ex.no.11
11:30 AM PM
OFF
69% 11:36
ex.no.11
11:37 AM PM
ALARM ON
69% 11:37
ex.no.11
11:37 AM PM
Alarm! Wake up! Wake up!
69% 11:37
ex.no.11
11:37 AM PM
ALARM OFF

```
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();   }
}
```

24. So now the Coding part of Alarm Receiver is also completed.

25. Now run the application to see the output.

Observation	25	
Record	10	
Total	35	
Signature		

Result:

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