COMPUTER LABORATORY MANUAL

Mobile Application Development
(CSC491L)
Version 2.0

Student Name:	
Class / Section:	
Roll Number:	
CGPA:	
Email ID:	

DEPARTMENT OF SOFTWARE ENGINEERING (DSE)
FOUNDATION UNIVERSITY RAWALPINDI CAMPUS (FURC)

www.fui.edu.pk
http://www.fui.edu.pk/FURC/

PREFACE

This Lab Manual has been specially designed with the idea of improving and enhancing the problem-solving skills of students through consistent practice and creativity. The Lab Manual covers all the topics taught in the BS Software Engineering core course Mobile Application Development. Each lab activity has been divided into three parts; Learning Objectives, Practice Questions and Exercises. Learning Objectives indicate the specific areas of JAVA Language programming, in which a student will become proficient after successfully completing the lab activity. These questions have been carefully devised with the idea of keeping the students interested as well as motivated to complete the designated tasks. There are about 1 to 3 exercise questions included in each lab activity. These questions are intended to refine a student's advanced level programming skills. After the completion of a lab activity, students will take the POST-Lab tasks home, complete these tasks and submit the source code files to course instructor.

PREPARED BY

Lab manual is prepared by Mr. Muhammad Asif under the supervision of Head of Department in the year 2019. Revised in January 2020 by adding more experiments and done some corrections.

GENERAL INSTRUCTIONS

a. Students are required to maintain the lab manual with them till the end of the semester.
b. All readings, answers to questions and illustrations must be solved on the place provided. If more space is required, then additional sheets may be attached.
c. It is the responsibility of the student to have the manual graded before deadlines as given by the instructor
d. Loss of manual will result in re submission of the complete manual.
e. Students are required to go through the experiment before coming to the lab session. Lab session details will be given in training schedule.
f. Students must bring the manual in each lab.
g. Keep the manual neat clean and presentable.
h. Plagiarism is strictly forbidden. No credit will be given if a lab session is plagiarized and no re submission will be entertained.
i. Marks will be deducted for late submission.
j. You can use separate sheets for Lab exercises and attach them with the lab manuals.

VERSION HISTORY

Date	Update By	Details
January 2019	Muhammad Asif	Version 1.0. Initial draft prepared and experiments outlined.
January 2020	Muhammad Asif	Version 2.0. Addition of Introduction to IDE, Latest Application Development exercises and Tasks with cross development platforms e.g. Flutter

MARKS

LAB #	Date Conducted	Lab Title	Max. Marks	Marks Obtained	Instructor Sign
1		ANDROID APPLICATION THAT USES GUI COMPONENTS, FONT AND COLORS	10		
2		ANDROID APPLICATION FOR LAYOUT MANAGERS AND EVENT LISTENERS	10		
3		SIMPLE ANDROID APPLICATION FOR NATIVE CALCULATOR	10		
4		ANDROID APPLICATION TO DRAW BASIC GRAPHICAL PRIMITIVES	10		
5		SIMPLE ANDROID APPLICATION THAT MAKES USE OF DATABASE	10		
6		ANDROID APPLICATION THAT MAKES USE OF RSS FEED	10		
7		ANDROID APPLICATION THAT IMPLEMENTS MULTI THREADING	10		
8		ANDROID APPLICATION THAT WRITES DATA TO THE SD CARD	10		
9		ANDROID APPLICATION THAT CREATES AN ALERT UPON RECEIVING A MESSAGE	10		
10		ANDROID APPLICATION THAT CREATES ALARM CLOCK	10		
11		ANDROID ANALOG CLOCK AND DIGITAL CLOCK EXAMPLE	10		
12		ANDROID MEDIA PLAYER EXAMPLE	10		
13		ANDROID GOOGLE MAP	10		
14		ANDROID TEXTTOSPEECH TUTORIAL	10		
15		ANDROID SENSOR TUTORIAL	10		
16		CROSS-PLATFORM TOOLS FOR APP DEVELOPMENT (ANDROID, IOS AND WINDOWS PHONE)	10		
Grand Total					

LIST OF EXPERIMENTS

EXERCISE 1 – ANDROID APPLICATION THAT USES GUI COMPONENTS, FONT AND COLORS 5

EXERCISE 2 – ANDROID APPLICATION FOR LAYOUT MANAGERS AND EVENT LISTENERS.....18

EXERCISE 3 – SIMPLE ANDROID APPLICATION FOR NATIVE CALCULATOR 34

EXERCISE 4 – ANDROID APPLICATION TO DRAW BASIC GRAPHICAL PRIMITIVES 47

EXERCISE 5 – SIMPLE ANDROID APPLICATION THAT MAKES USE OF DATABASE.....57

EXERCISE 6 – ANDROID APPLICATION THAT MAKES USE OF RSS FEED.....73

EXERCISE 7 – ANDROID APPLICATION THAT IMPLEMENTS MULTI THREADING 87

EXERCISE 8 – ANDROID APPLICATION THAT WRITES DATA TO THE SD CARD .97

EXERCISE 9 – ANDROID APPLICATION THAT CREATES AN ALERT UPON RECEIVING A MESSAGE.....109

EXERCISE 10 – ANDROID APPLICATION THAT CREATES ALARM CLOCK.....122

EXERCISE 11 – ANDROID ANALOG CLOCK AND DIGITAL CLOCK EXAMPLE....136

EXERCISE 12 – ANDROID MEDIA PLAYER EXAMPLE.....142

EXERCISE 13 – ANDROID GOOGLE MAP 153

EXERCISE 14 – ANDROID TEXTTOSPEECH TUTORIAL.....163

EXERCISE 15 – ANDROID SENSOR TUTORIAL 172

EXERCISE 16 – CROSS-PLATFORM TOOLS FOR APP DEVELOPMENT (ANDROID, IOS AND WINDOWS PHONE) 179

EXERCISE 1 – ANDROID APPLICATION THAT USES GUI COMPONENTS, FONT AND COLORS

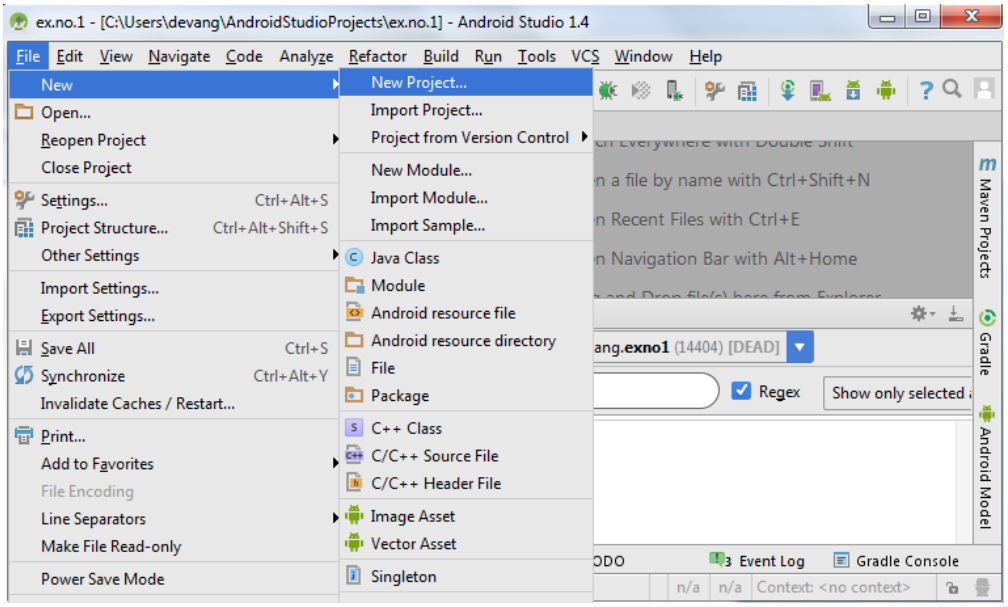
Aim:

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

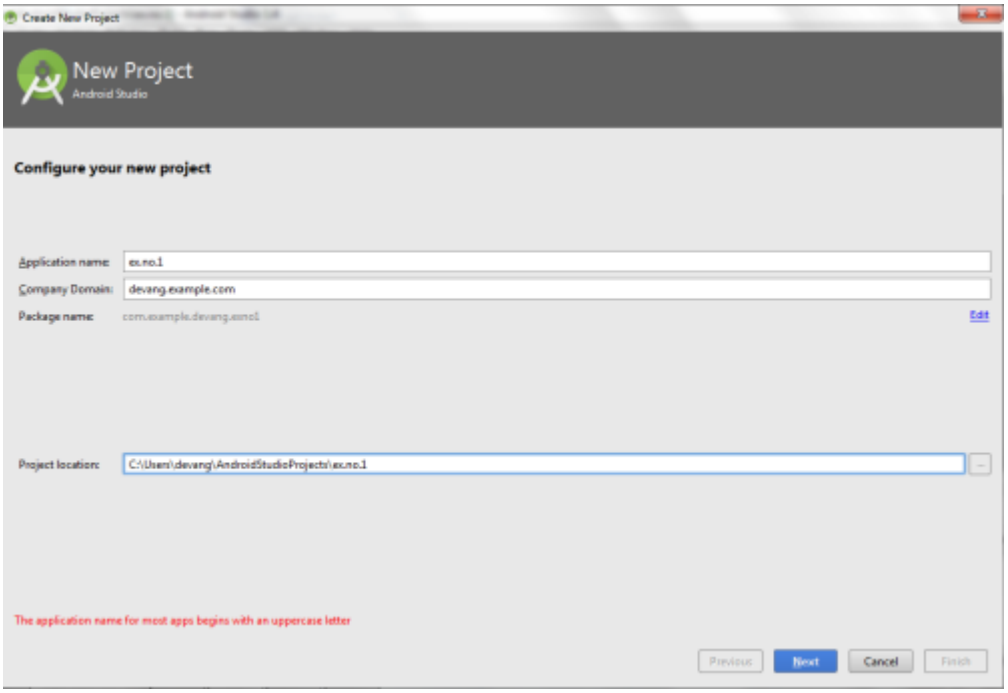
Procedure:

Creating a New project:

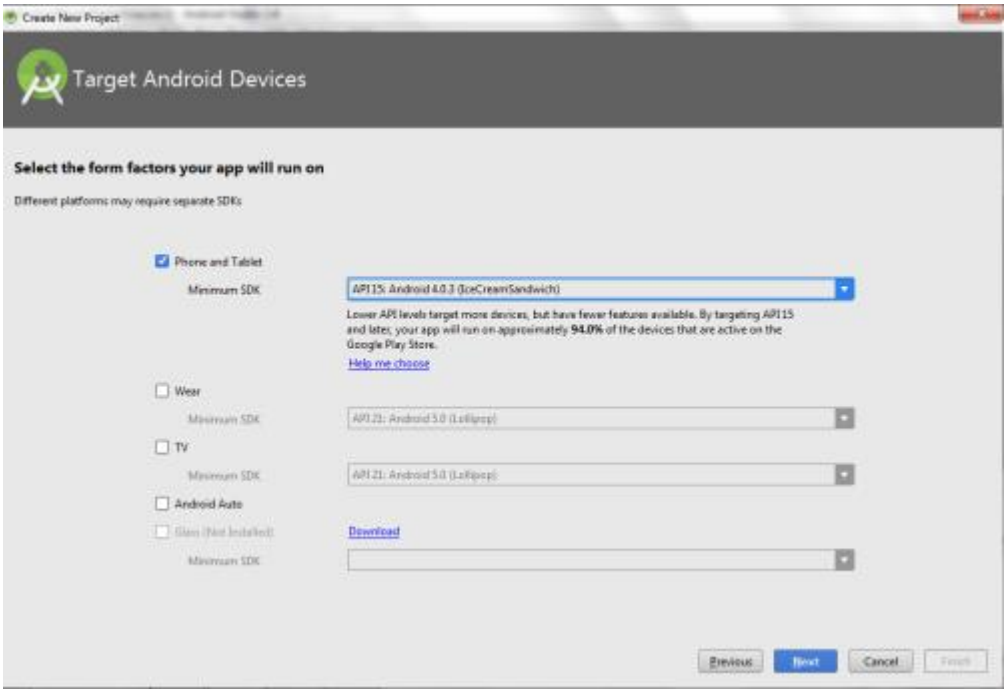
- Open Android Studio and then click on **File -> New -> New project.**



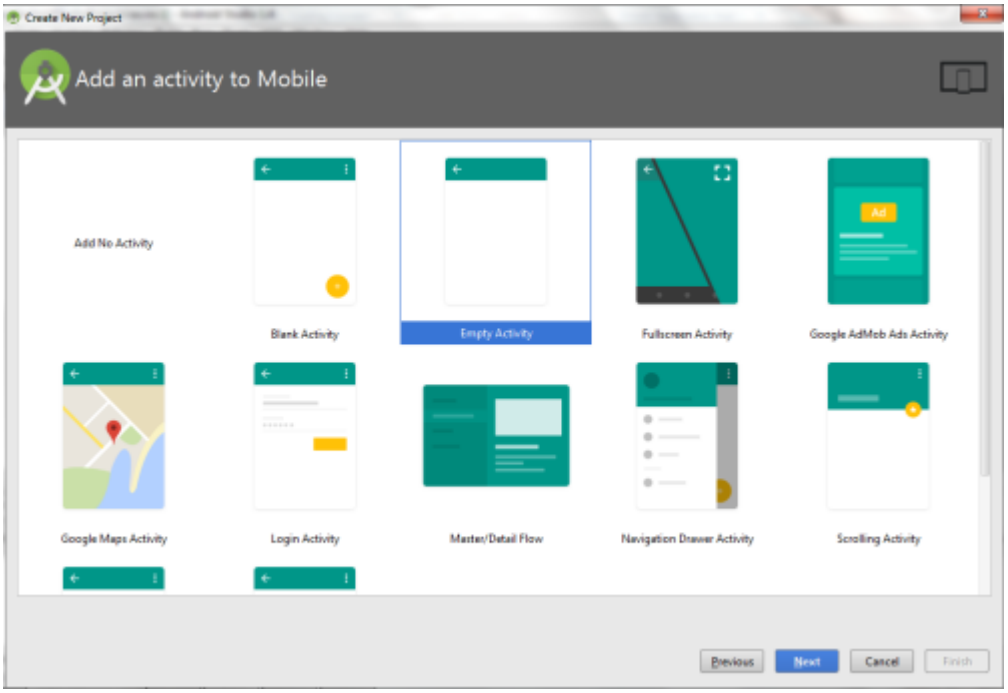
- Then type the Application name as "**ex.no.1"** and click **Next.**



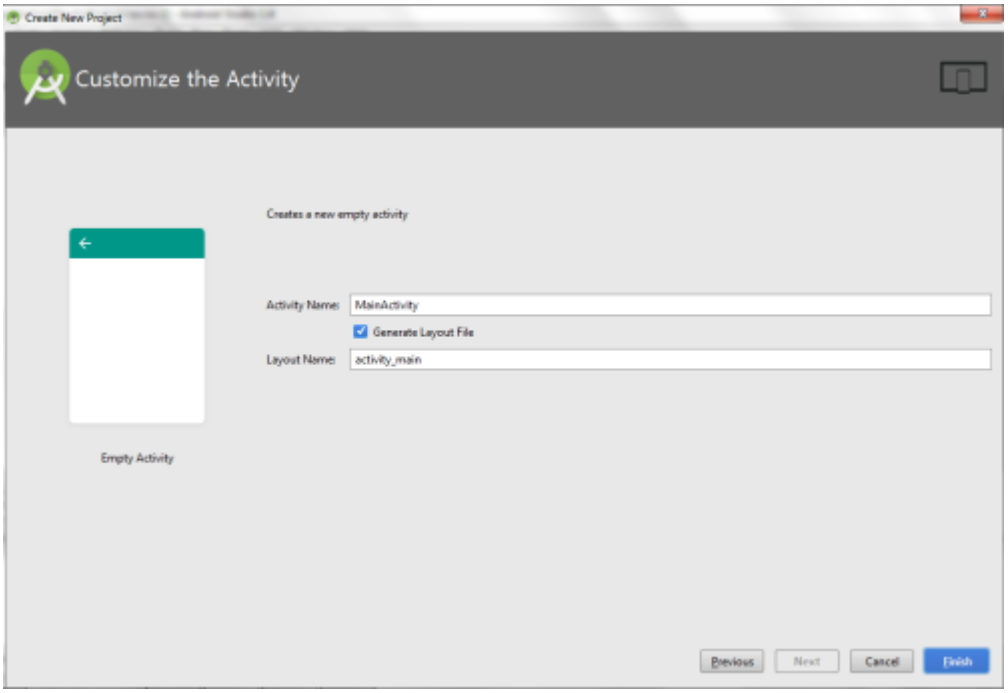
- Then select the **Minimum SDK** as shown below and click **Next**.



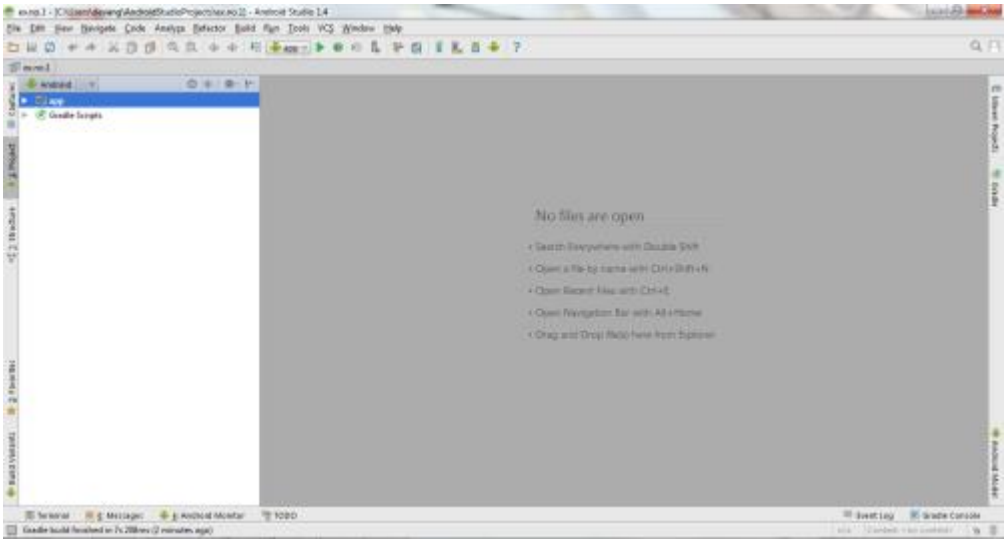
- Then select the **Empty Activity** and click **Next.**



- Finally click F**inish**.

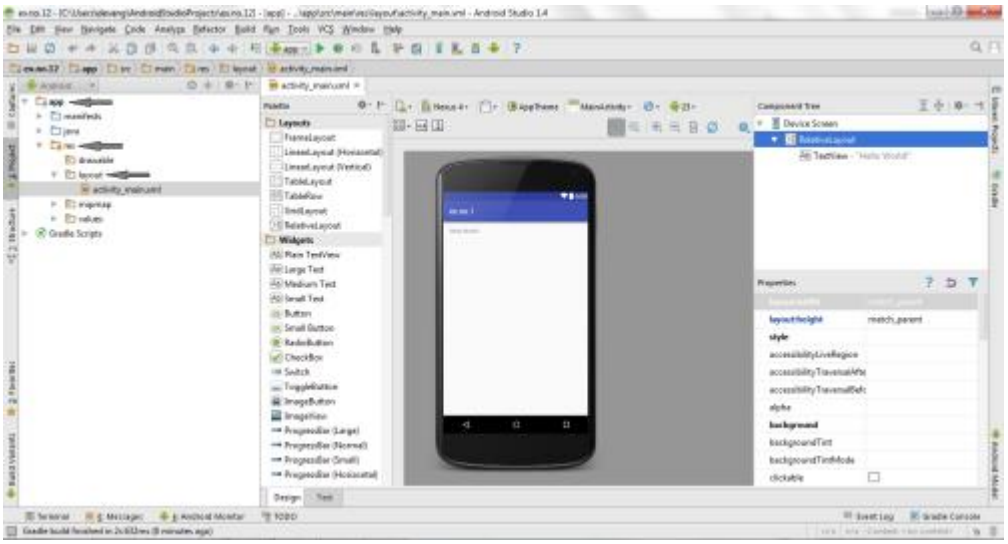


- It will take some time to build and load the project.
- After completion it will look as given below.

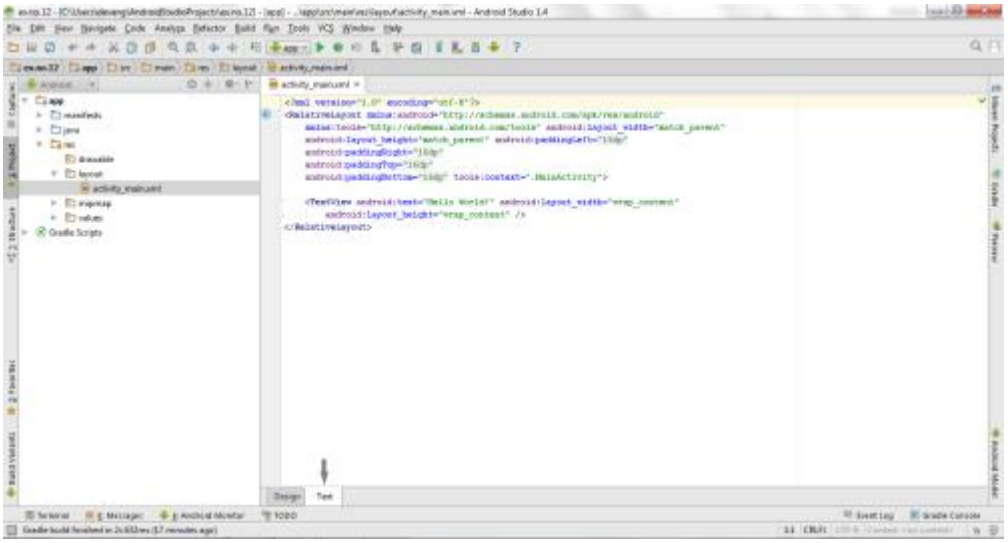


Designing layout for the Android Application:

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



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

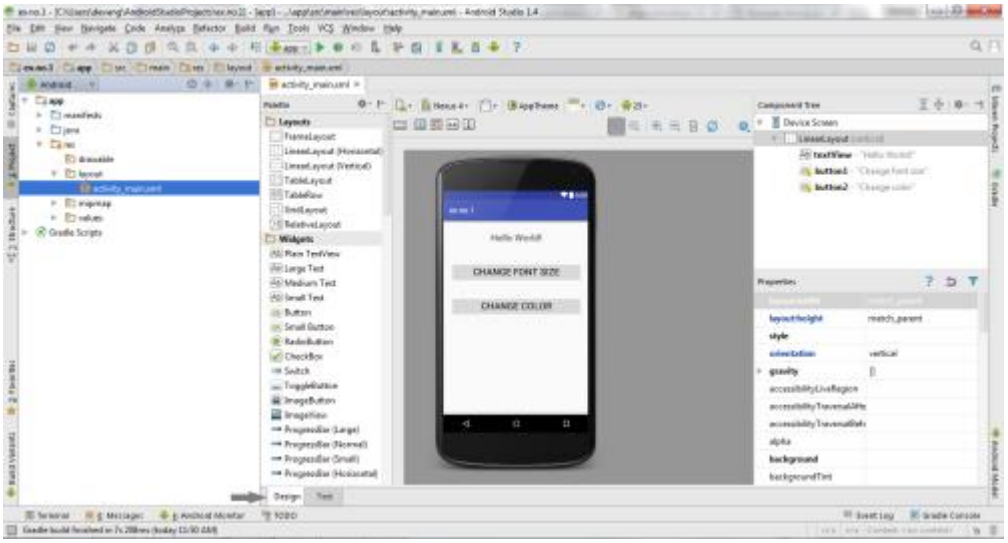


- Then delete the code which is there and type the code as given below.

Code for Activity_main.xml:

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

    <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>
```

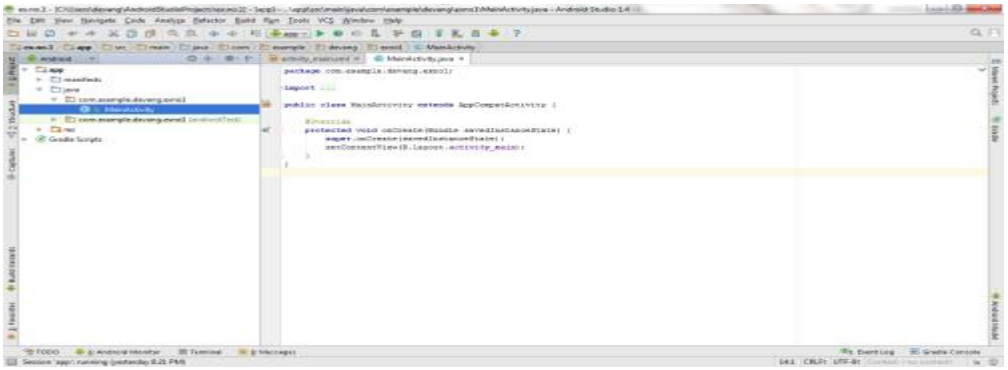
- Now click on Design and your application will look as given below.



- So now the designing part is completed.

Java Coding for the Android Application:

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



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

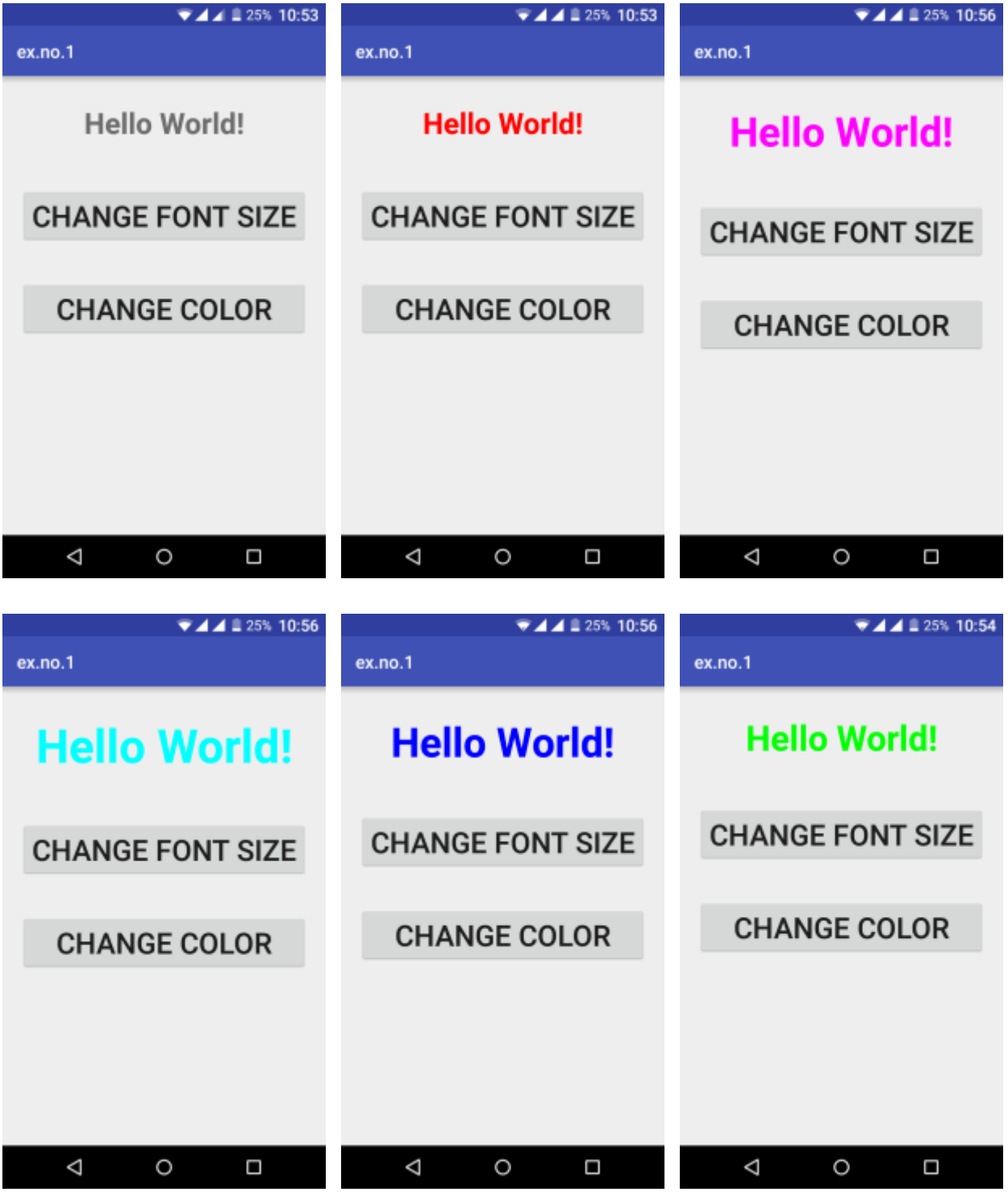
Code for MainActivity.java:

```
package com.example.exno1;

import android.graphics.Color;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity
{
    int ch=1;
    float font=30;
    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_main);
        final TextView t= (TextView) findViewById(R.id.textView);
        Button b1= (Button) findViewById(R.id.button1);
        b1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                t.setTextSize(font);
                font = font + 5;
                if (font == 50)
                    font = 30;
            }
        });
        Button b2= (Button) findViewById(R.id.button2);
        b2.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                switch (ch) {
                    case 1:
                        t.setTextColor(Color.RED);
                        break;
                    case 2:
                        t.setTextColor(Color.GREEN);
                        break;
                    case 3:
                        t.setTextColor(Color.BLUE);
                        break;
                    case 4:
                        t.setTextColor(Color.CYAN);
                        break;
                    case 5:
                        t.setTextColor(Color.YELLOW);
                        break;
                    case 6:
                        t.setTextColor(Color.MAGENTA);
                        break;
                }
                ch++;
                if (ch == 7)
                    ch = 1;
            }
        });
    }
}
```

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

Output:



Result:

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

EXERCISE 1.1:

Execute the above exercise of font size and color in Android studio and show the exact GUI output? [2]

TASK 1.1

In the above example, Change the font styles of the text “Hello World!” by adding another button?

EXERCISE 1.2:

Develop an android application to display current date and time on the Button widget. [6]

EXERCISE 1.3: [2]

Develop an android application to display the following widgets:

1. Android Button
2. Android Toast
3. Custom Toast
4. ToggleButton
5. CheckBox
6. AlertDialog
7. Spinner
8. AutoCompleteTextView
9. RatingBar
10. DatePicker
11. TimePicker
12. ProgressBar
13. Check Box
14. Radio Button

RESOURCES:

https://www.codingconnect.net/mobile-application-development-lab/
https://www.javatpoint.com/android-tutorial
https://www.tutorialspoint.com/android
https://developer.android.com/guide
Https://developer.Android.com/training/basics/firstapp/creating-project
Https://www.raywenderlich.com/120177/beginning-Android-development-tutorial-installingAndroid-studio
https://www.youtube.com/playlist?list=PLS1QulWo1RIbb1cYyzZpLFCKvdYV_yJ-E
Https://www.udemy.com/java-tutorial/
Https://www.w3schools.com/xml/
Http://www.Androidguys.com/

EXERCISE 2 – ANDROID APPLICATION FOR LAYOUT MANAGERS AND EVENT LISTENERS

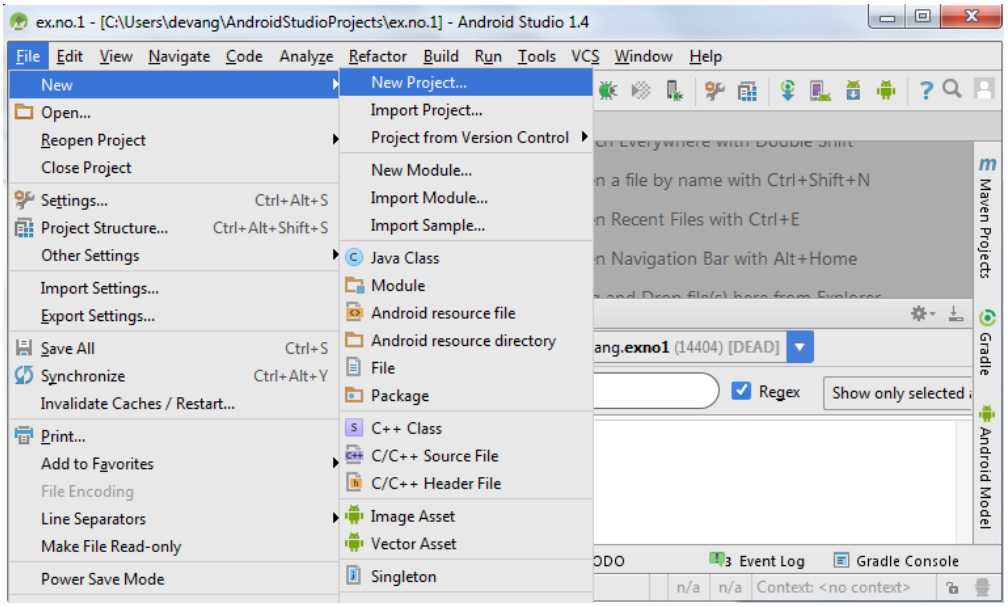
Aim:

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

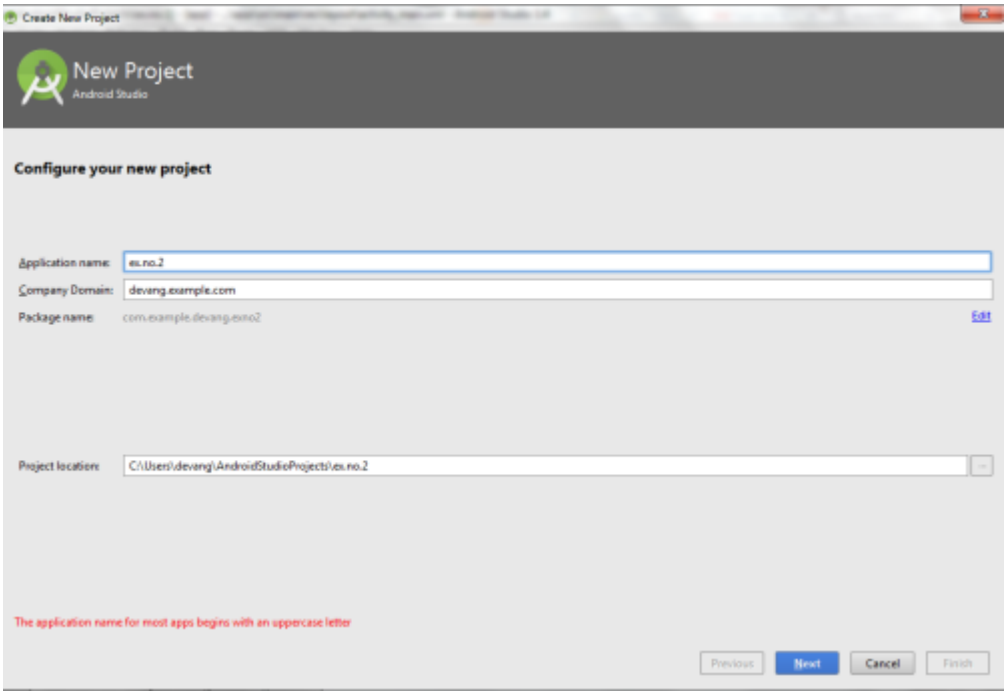
Procedure:

Creating a New project:

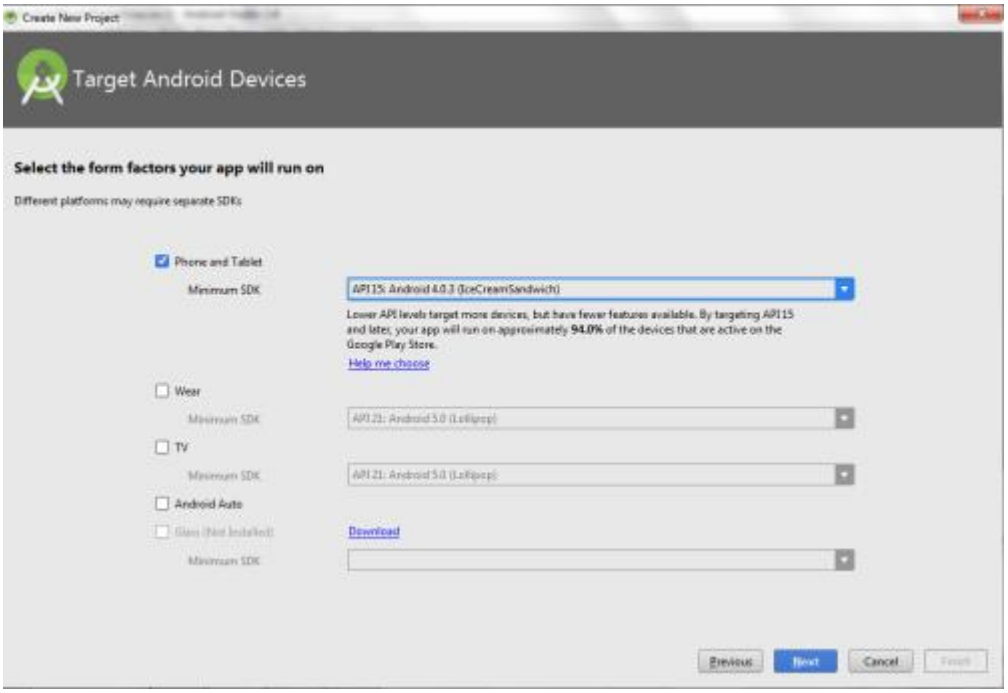
- Open Android Stdio and then click on **File -> New -> New project.**



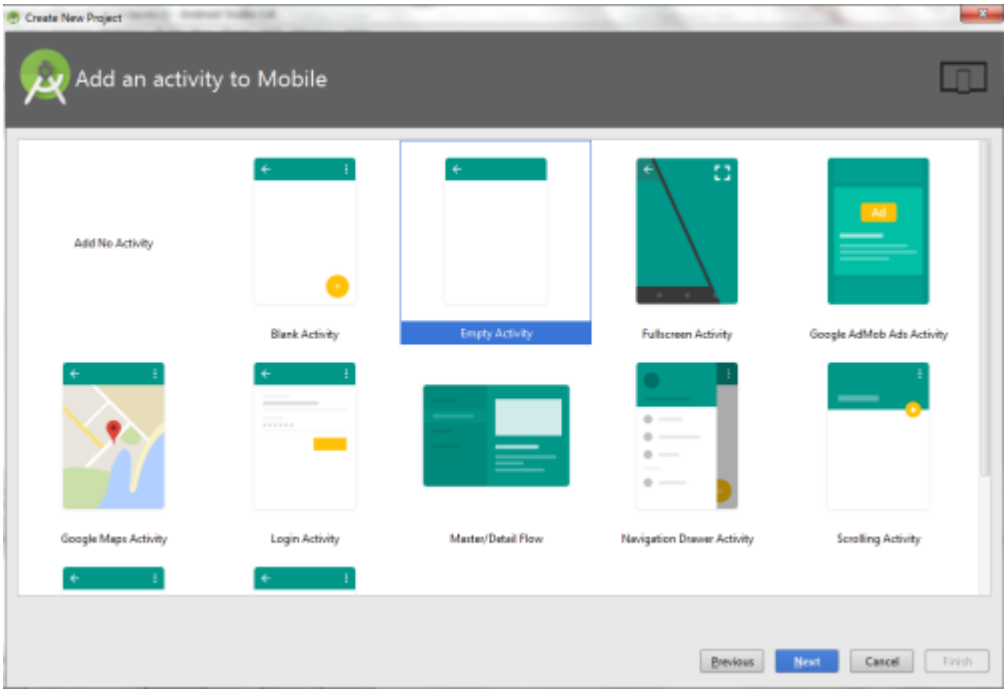
- Then type the Application name as "**ex.no.2"** and click **Next.**



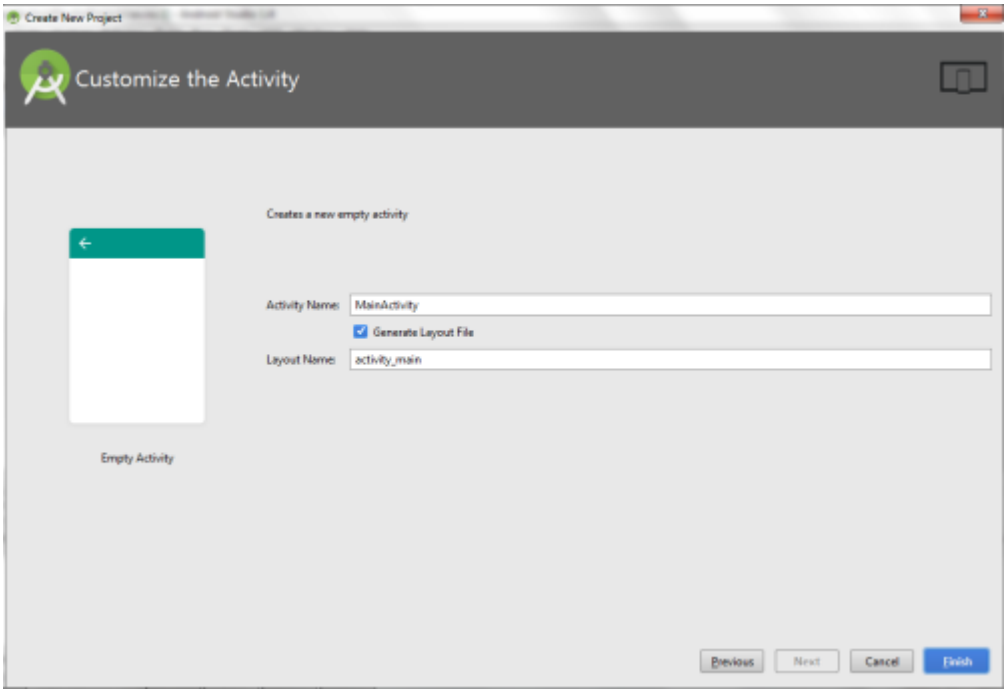
- Then select the **Minimum SDK** as shown below and click **Next**.



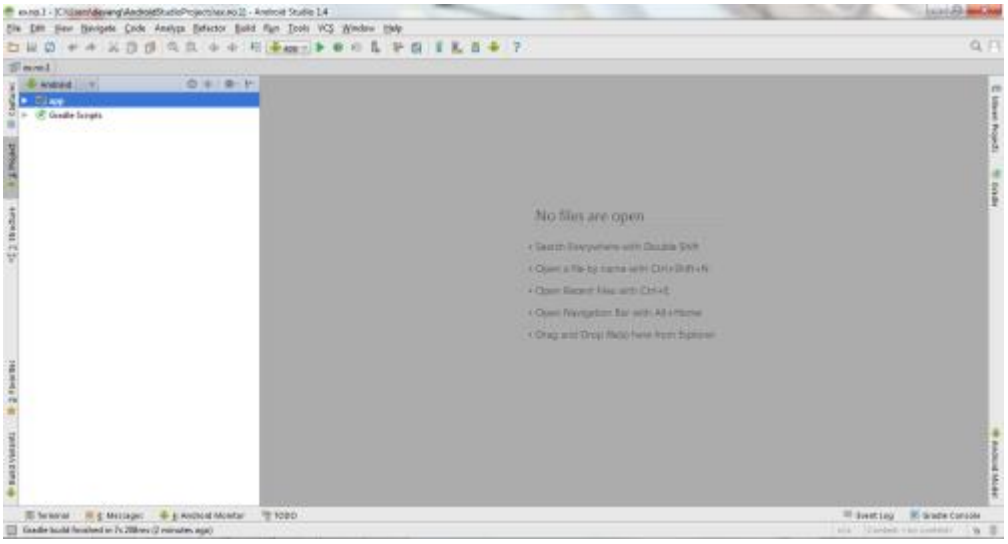
- Then select the **Empty Activity** and click **Next.**



- Finally click F**inish**.

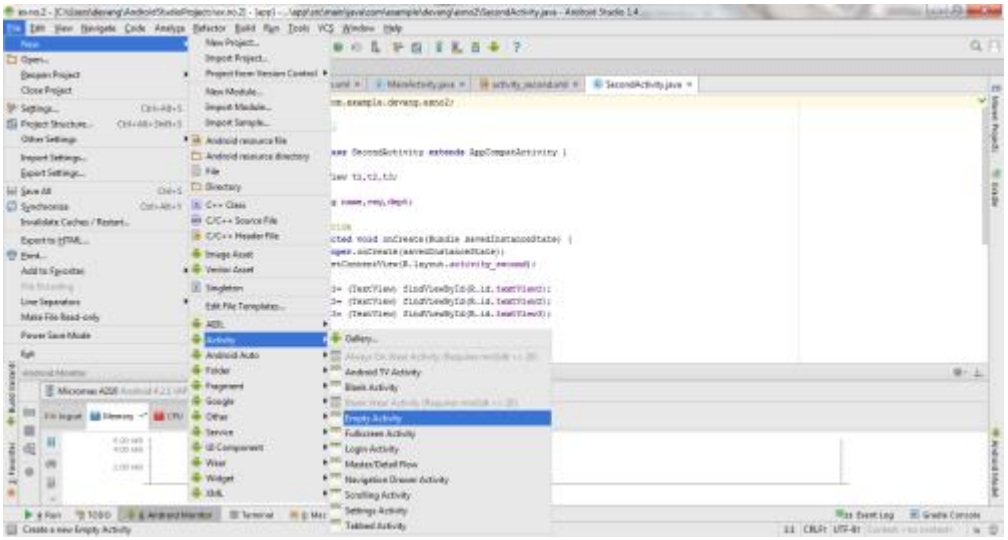


- It will take some time to build and load the project.
- After completion it will look as given below.

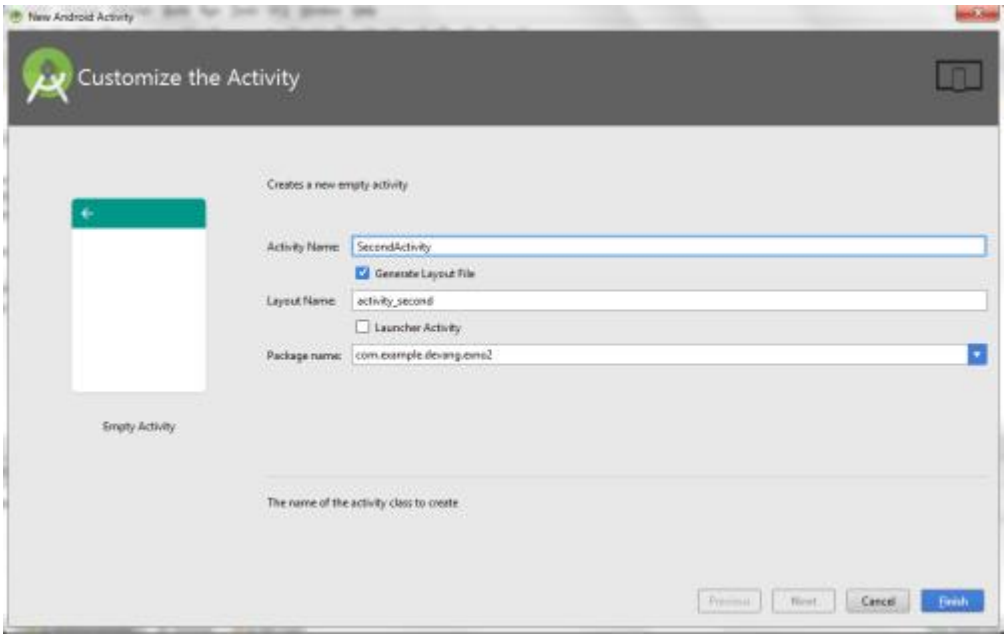


Creating Second Activity for the Android Application:

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



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

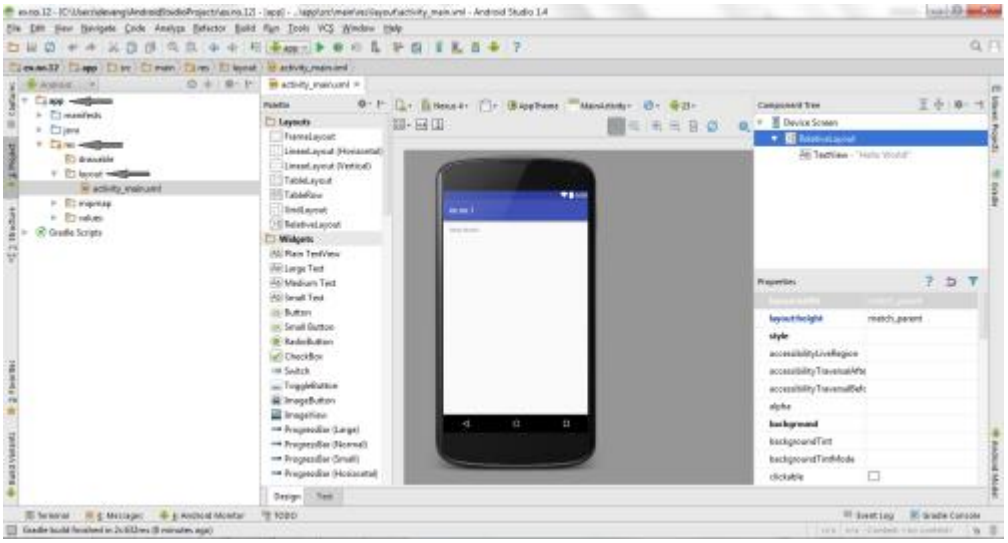


- Thus Second Activity For the application is created.

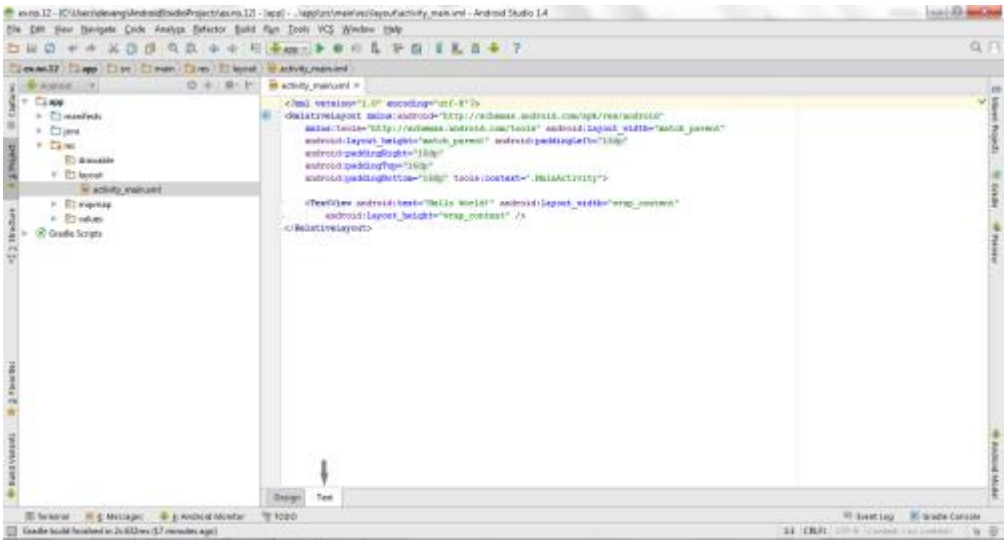
Designing layout for the Android Application:

Designing Layout for Main Activity:

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



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



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

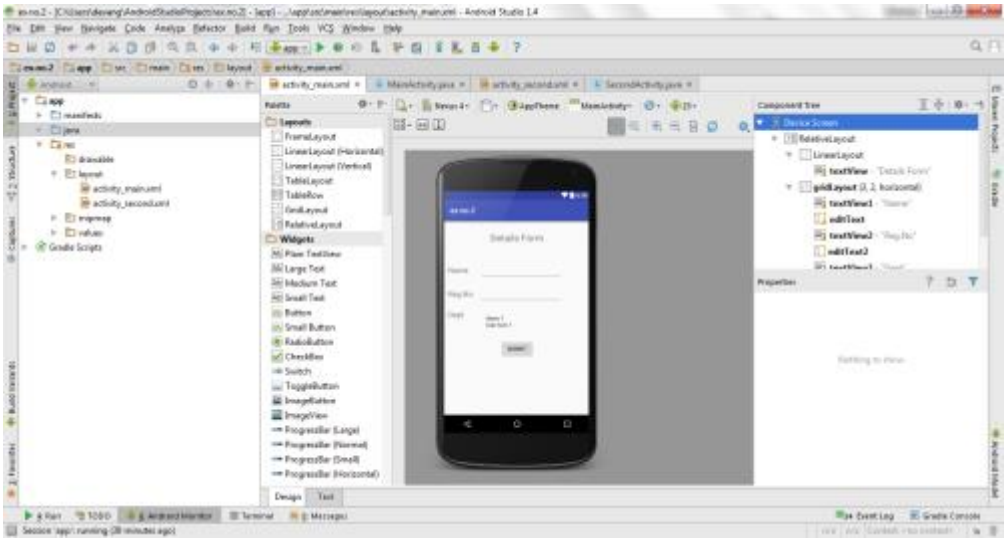
Code for Activity_main.xml:

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

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="100dp">
        <TextView
            android:id="@+id/textView"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_margin="30dp"
            android:text="Details Form"
```

```
        android:textSize="25sp"
        android:gravity="center"/>
</LinearLayout>

<GridLayout
    android:id="@+id/gridLayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginTop="100dp"
    android:layout_marginBottom="200dp"
    android:columnCount="2"
    android:rowCount="3">
    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:layout_row="0"
        android:layout_column="0"
        android:text="Name"
        android:textSize="20sp"
        android:gravity="center"/>

    <EditText
        android:id="@+id/editText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:layout_row="0"
        android:layout_column="1"
        android:ems="10"/>

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:layout_row="1"
        android:layout_column="0"
        android:text="Reg.No"
        android:textSize="20sp"
        android:gravity="center"/>

    <EditText
        android:id="@+id/editText2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:layout_row="1"
        android:layout_column="1"
        android:inputType="number"
        android:ems="10"/>

    <TextView
        android:id="@+id/textView3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:layout_row="2"
        android:layout_column="0"
```

```
            android:text="Dept"
            android:textSize="20sp"
            android:gravity="center"/>

        <Spinner
            android:id="@+id/spinner"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_margin="10dp"
            android:layout_row="2"
            android:layout_column="1"
            android:spinnerMode="dropdown"/>

    </GridLayout>

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_centerInParent="true"
        android:layout_marginBottom="150dp"
        android:text="Submit"/>

</RelativeLayout>
```

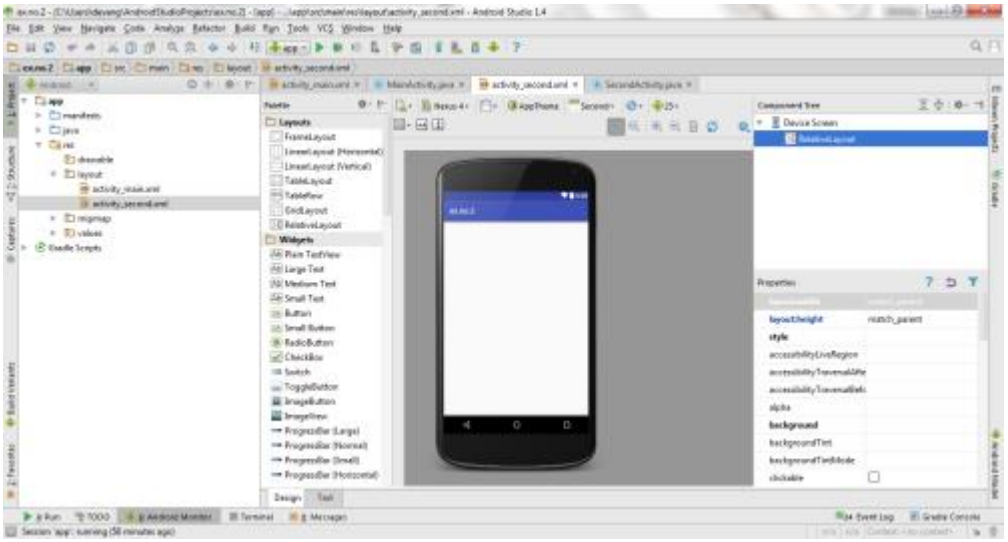
- Now click on Design and your activity will look as given below.



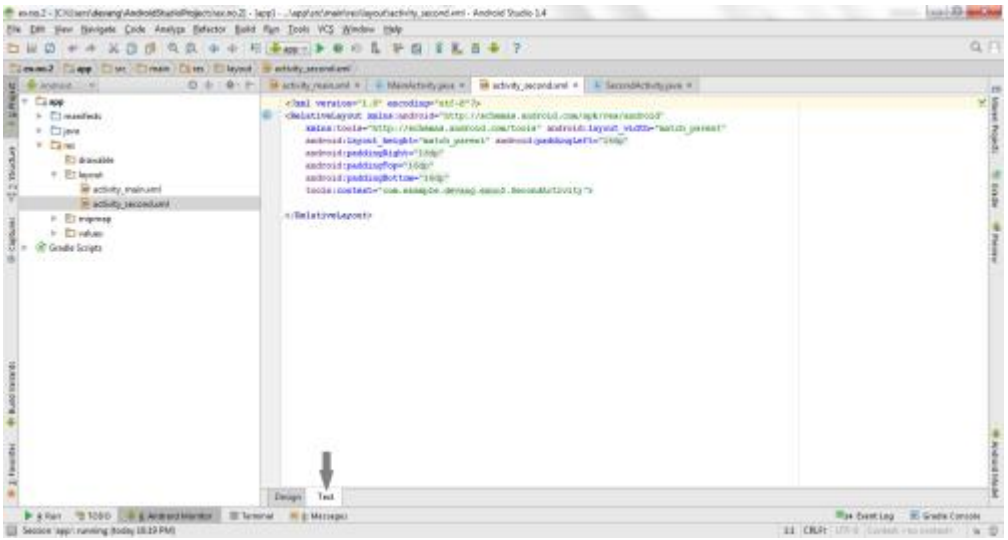
- So now the designing part of Main Activity is completed.

Designing Layout for Second Activity:

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



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



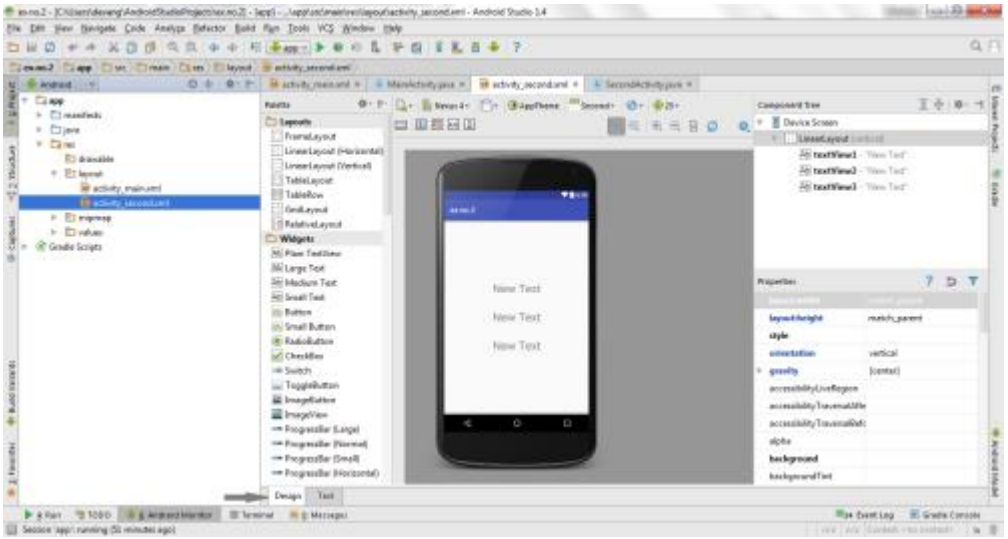
- Then delete the code which is there and type the code as given below.

Code for Activity_second.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.devang.exno2.SecondActivity"
    android:orientation="vertical"
    android:gravity="center">

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="20dp"
        android:text="New Text"
        android:textSize="30sp"/>
```

```
    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="20dp"
        android:text="New Text"
        android:textSize="30sp"/>

    <TextView
        android:id="@+id/textView3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="20dp"
        android:text="New Text"
        android:textSize="30sp"/>

</LinearLayout>
```

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

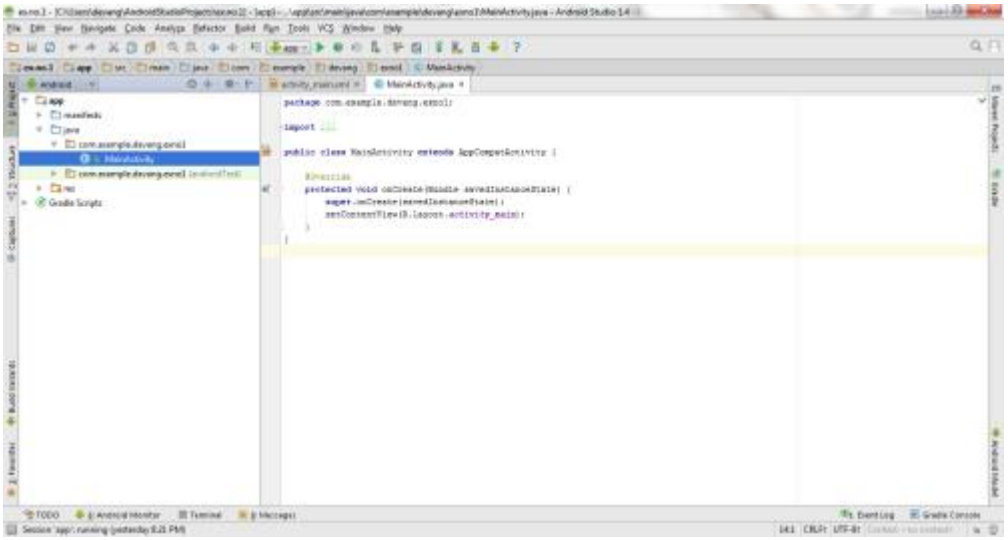


- So now the designing part of Second Activity is also completed.

Java Coding for the Android Application:

Java Coidng for Main Activity:

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



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

Code for MainActivity.java:

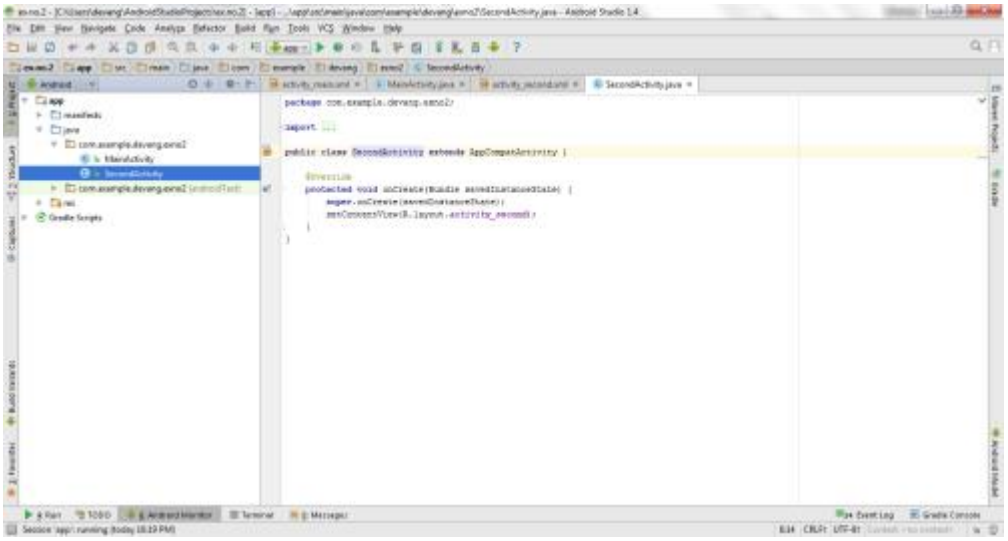
```
package com.example.exno2;

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

public class MainActivity extends AppCompatActivity {

    //Defining the Views
    EditText e1,e2;
    Button bt;
    Spinner s;

    //Data for populating in Spinner
    String [] dept_array={"CSE","ECE","IT","Mech","Civil"};

    String name,reg,dept;

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

        //Referring the Views
        e1= (EditText) findViewById(R.id.editText);
        e2= (EditText) findViewById(R.id.editText2);

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

        s= (Spinner) findViewById(R.id.spinner);
```

```
        //Creating Adapter for Spinner for adapting the data from array to
Spinner
        ArrayAdapter adapter= new
ArrayAdapter(MainActivity.this,android.R.layout.simple_spinner_item,dept_array
);
        s.setAdapter(adapter);

        //Creating Listener for Button
        bt.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                //Getting the Values from Views(Edittext & Spinner)
                name=e1.getText().toString();
                reg=e2.getText().toString();
                dept=s.getSelectedItem().toString();

                //Intent For Navigating to Second Activity
                Intent i = new Intent(MainActivity.this,SecondActivity.class);

                //For Passing the Values to Second Activity
                i.putExtra("name_key", name);
                i.putExtra("reg_key",reg);
                i.putExtra("dept_key", dept);

                startActivity(i);

            }
        });
    }
}
```

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

Java Coding for Second Activity:

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



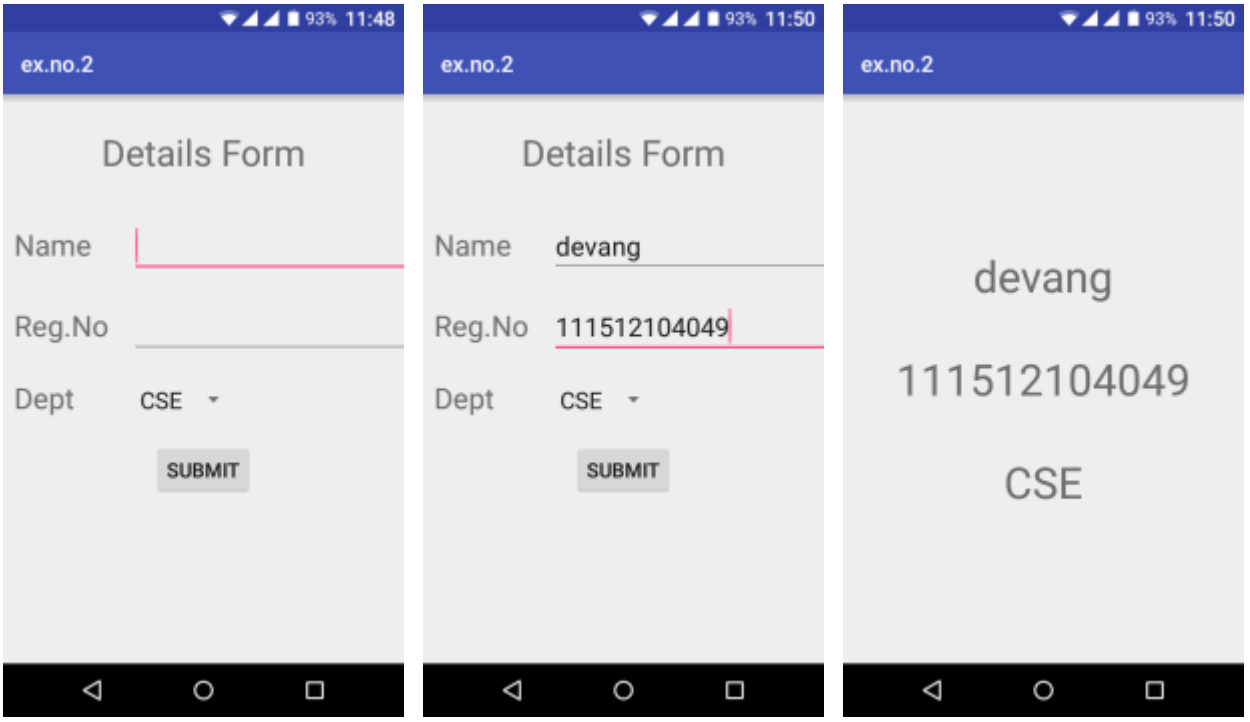
- Then delete the code which is there and type the code as given below.

Code for SecondActivity.java:

```
package com.example.exno2;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.TextView;

public class SecondActivity extends AppCompatActivity {

    TextView t1,t2,t3;

    String name,reg,dept;

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

        t1= (TextView) findViewById(R.id.textView1);
        t2= (TextView) findViewById(R.id.textView2);
        t3= (TextView) findViewById(R.id.textView3);

        //Getting the Intent
        Intent i = getIntent();

        //Getting the Values from First Activity using the Intent received
        name=i.getStringExtra("name_key");
        reg=i.getStringExtra("reg_key");
        dept=i.getStringExtra("dept_key");

        //Setting the Values to Intent
        t1.setText(name);
        t2.setText(reg);
        t3.setText(dept);

    }
}
```

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

Output:



Result:

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

EXERCISE 2.1:

Execute the above exercise of Layout Manager and Event Listener in Android studio and show the exact GUI output? [2]

TASK 2.1

Develop an android application to display Toast on clicking a button. This should be a customize button which includes setGravity and setMargin?

EXERCISE 2.2:

Develop an android application By using Implicit and Explicit Intent. By using Implicit intent, A web page should be open in a web browser. Also call 2nd activity from the first activity by using explicit intent. [8]

RESOURCES:

https://www.codingconnect.net/mobile-application-development-lab/
https://www.javatpoint.com/android-tutorial
https://www.tutorialspoint.com/android
https://developer.android.com/guide
Https://developer.Android.com/training/basics/firstapp/creating-project
Https://www.raywenderlich.com/120177/beginning-Android-development-tutorial-installingAndroid-studio
https://www.youtube.com/playlist?list=PLS1QulWo1RIbb1cYyzZpLFCKvdYV_yJ-E
Https://www.udemy.com/java-tutorial/
Https://www.w3schools.com/xml/
Http://www.Androidguys.com/

EXERCISE 3 – SIMPLE ANDROID APPLICATION FOR NATIVE CALCULATOR

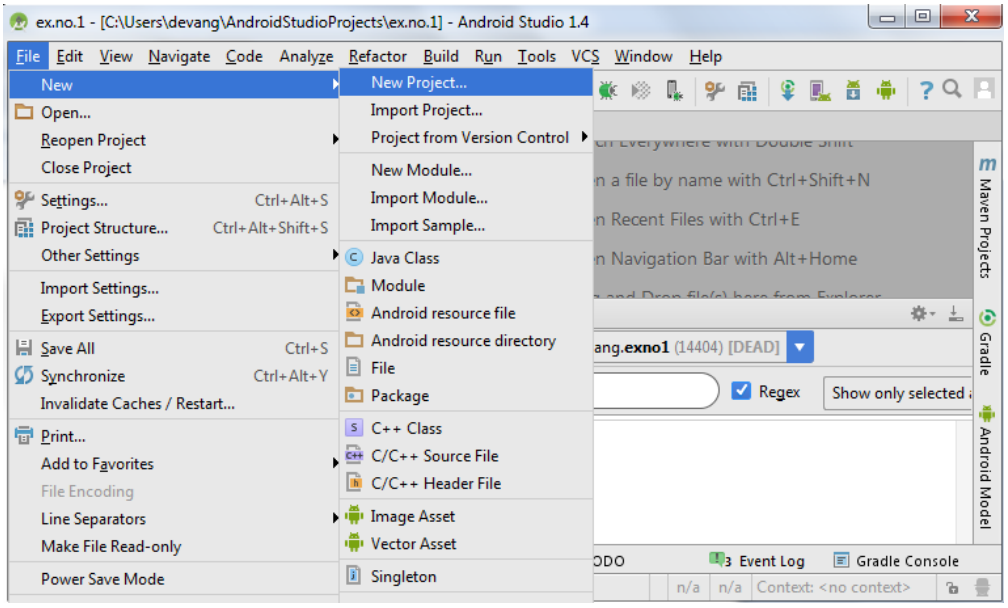
Aim:

To develop a Simple Android Application for Native Calculator.

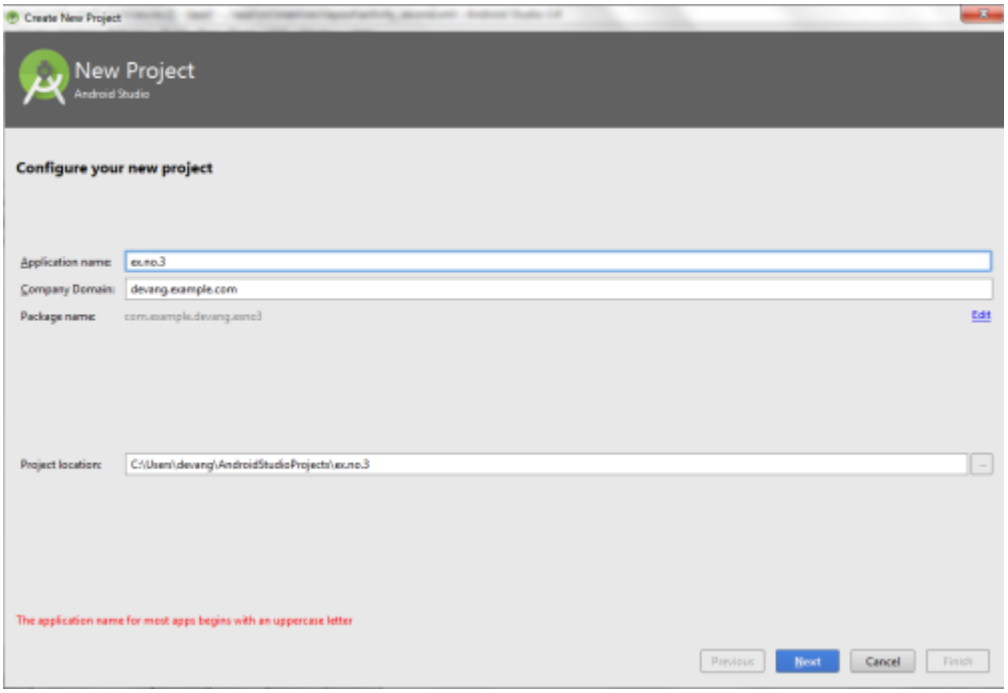
Procedure:

Creating a New project:

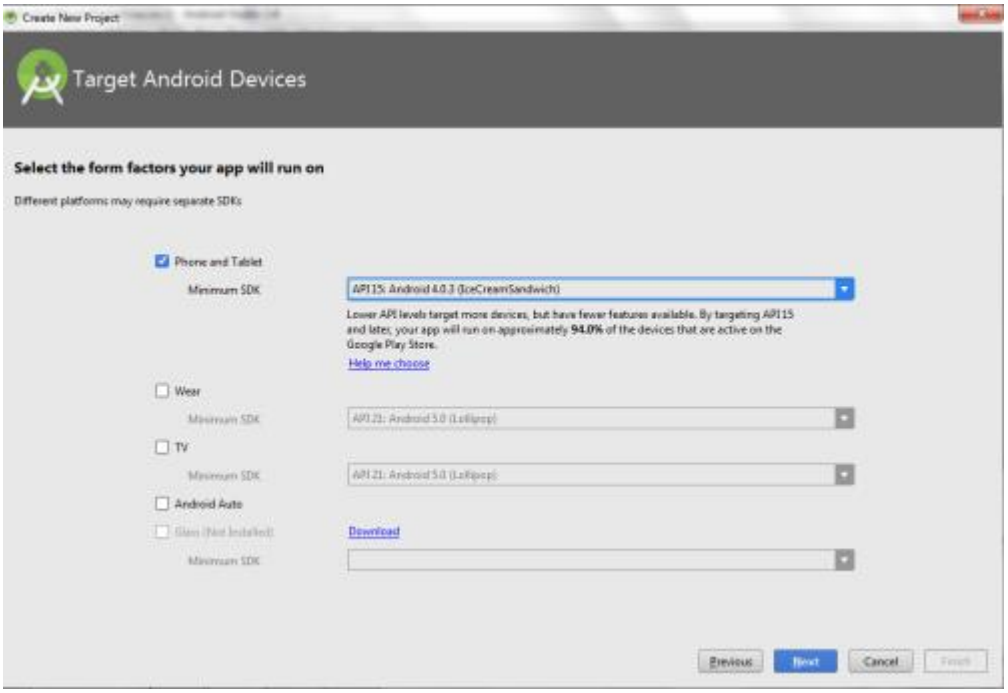
- Open Android Stdio and then click on **File -> New -> New project.**



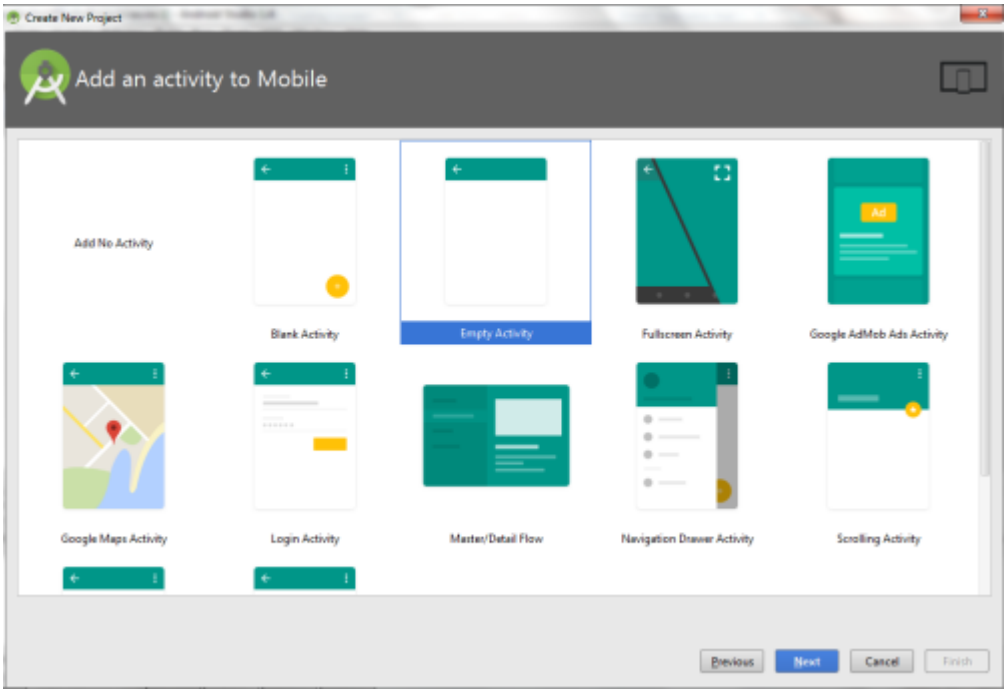
- Then type the Application name as "**ex.no.3"** and click **Next.**



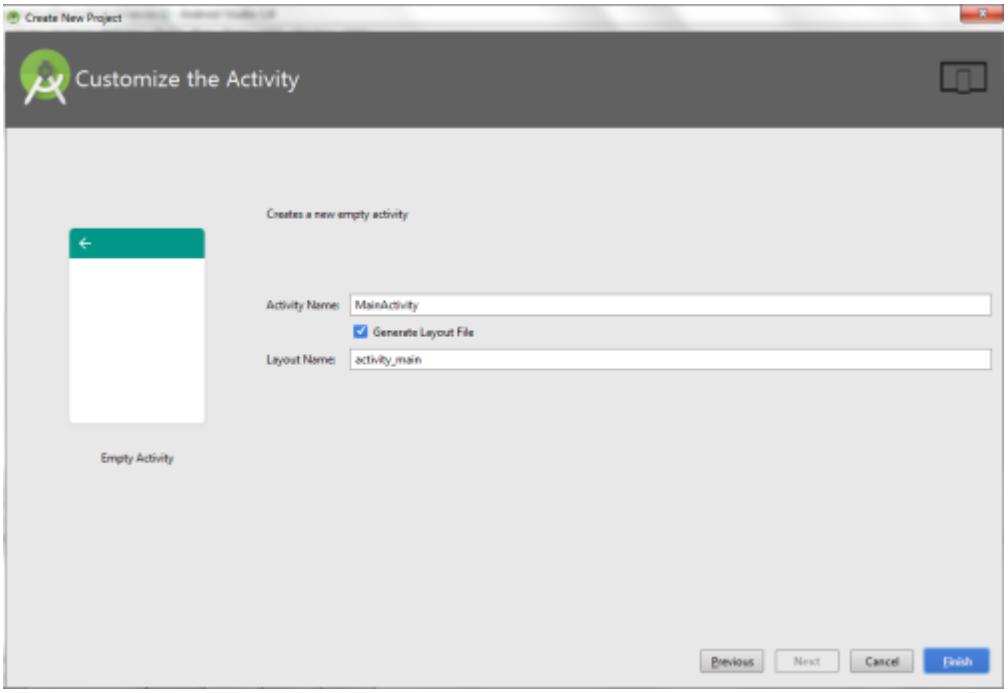
- Then select the **Minimum SDK** as shown below and click **Next**.



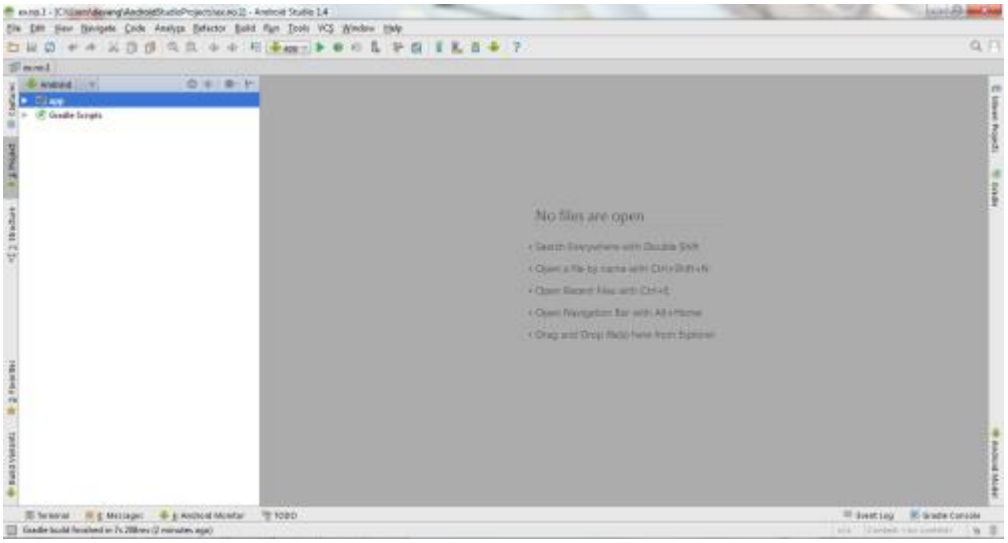
- Then select the **Empty Activity** and click **Next.**



- Finally click F**inish**.

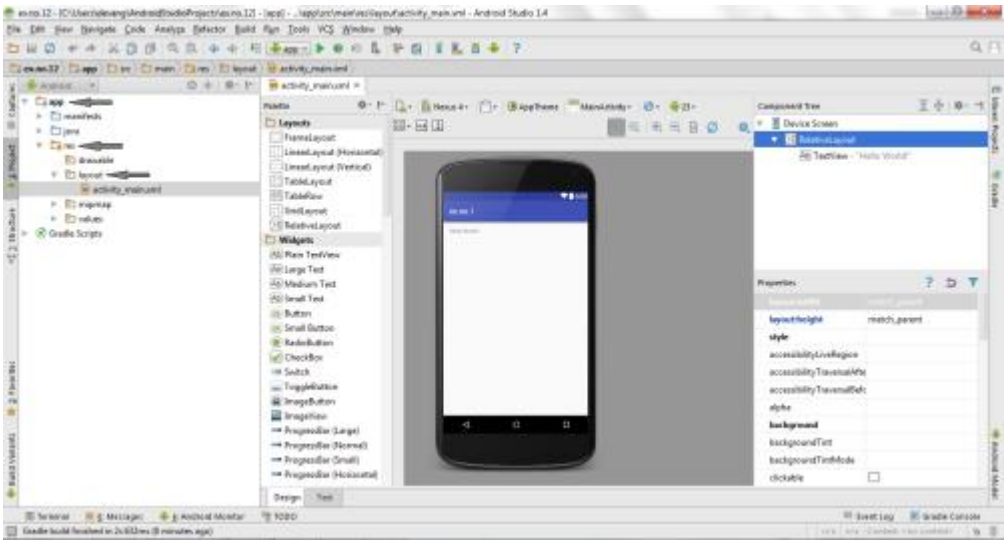


- It will take some time to build and load the project.
- After completion it will look as given below.

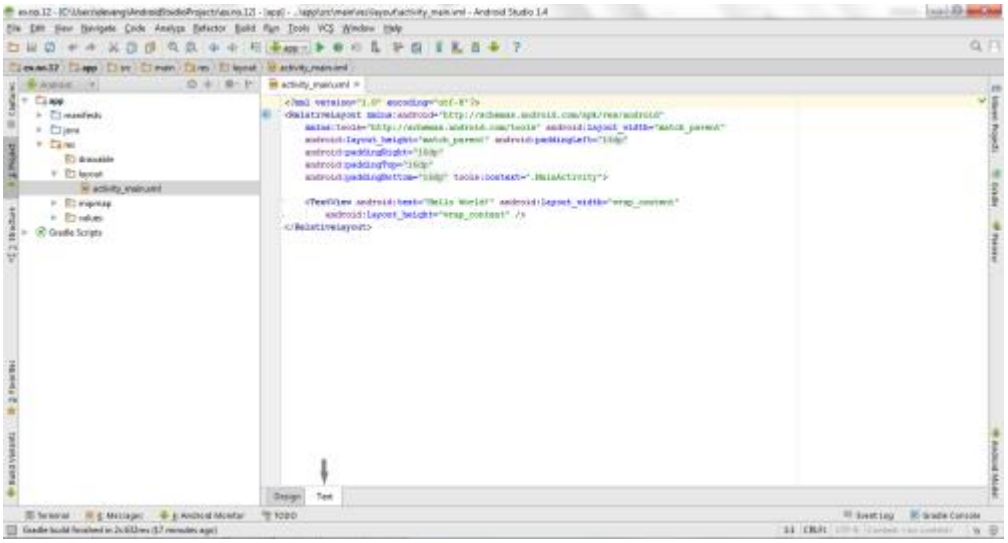


Designing layout for the Android Application:

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



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



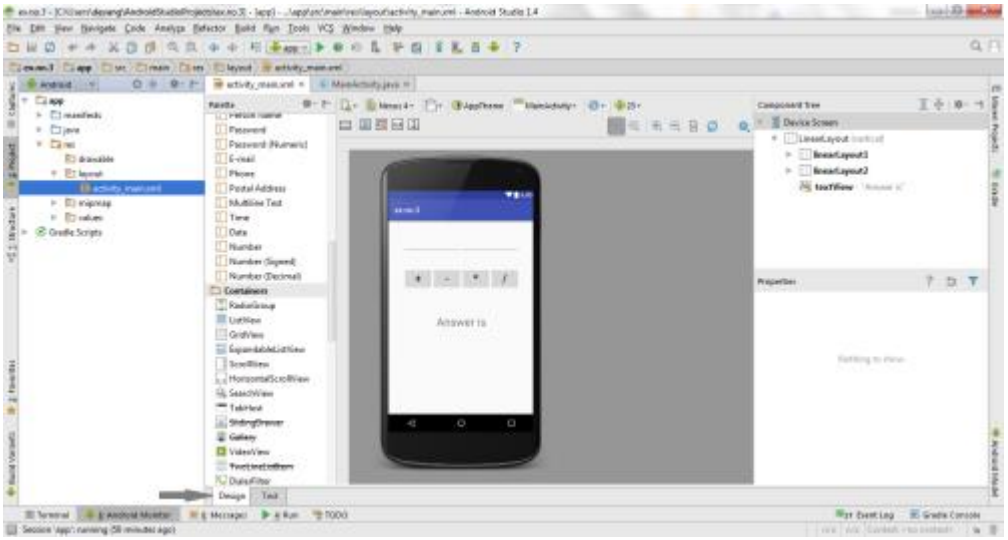
- Then delete the code which is there and type the code as given below.

Code for Activity_main.xml:

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

    <LinearLayout
        android:id="@+id/linearLayout1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="20dp">

        <EditText
            android:id="@+id/editText1"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:inputType="numberDecimal"
            android:textSize="20sp" />

        <EditText
            android:id="@+id/editText2"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:inputType="numberDecimal"
            android:textSize="20sp" />

    </LinearLayout>

    <LinearLayout
        android:id="@+id/linearLayout2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
```

```
        android:layout_margin="20dp">

        <Button
            android:id="@+id/Add"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="+"
            android:textSize="30sp"/>

        <Button
            android:id="@+id/Sub"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="-"
            android:textSize="30sp"/>

        <Button
            android:id="@+id/Mul"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="*"
            android:textSize="30sp"/>

        <Button
            android:id="@+id/Div"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="/"
            android:textSize="30sp"/>

    </LinearLayout>

    <TextView
        android:id="@+id/textView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="50dp"
        android:text="Answer is"
        android:textSize="30sp"
        android:gravity="center"/>

</LinearLayout>
```

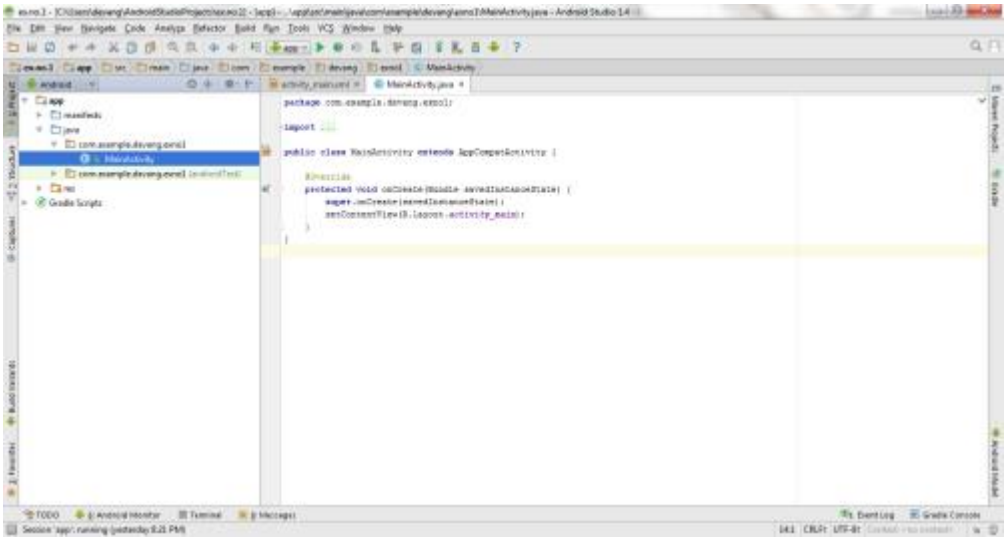
- Now click on Design and your application will look as given below.



- So now the designing part is completed.

Java Coding for the Android Application:

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



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

Code for MainActivity.java:

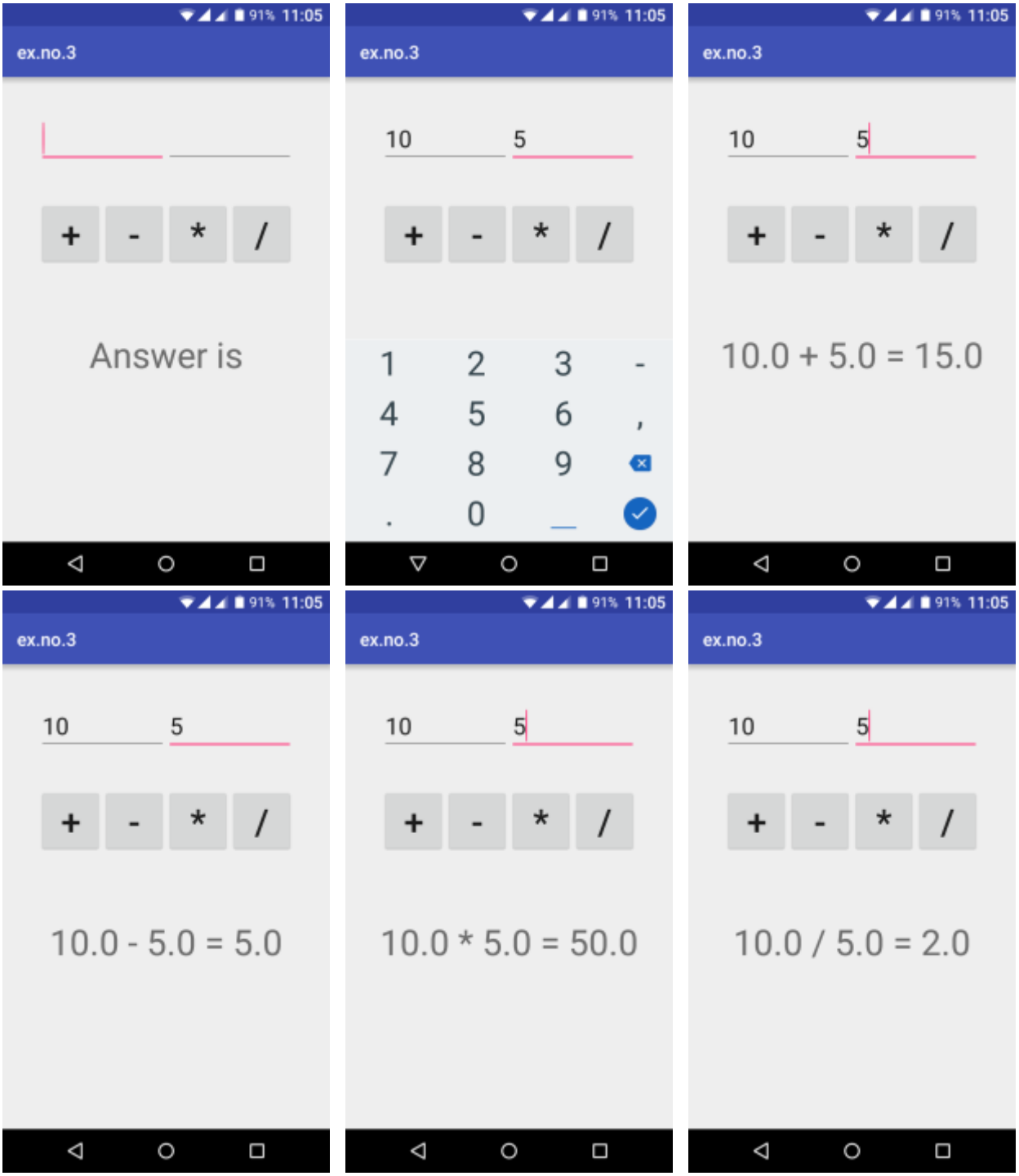
```
package com.example.devang.exno3;

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

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

Output:



Result:

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

EXERCISE 3.1:

Execute the above exercise of Native Calculator in Android studio and show the exact GUI output? [2]

EXERCISE 3.2:

Develop an android application for Scientific Calculator. Mathematical functionality should be present e.g. Sin, Tan, Cos, Sqrt, log etc. [8]

RESOURCES:

https://www.codingconnect.net/mobile-application-development-lab/
https://www.javatpoint.com/android-tutorial
https://www.tutorialspoint.com/android
https://developer.android.com/guide
Https://developer.Android.com/training/basics/firstapp/creating-project
Https://www.raywenderlich.com/120177/beginning-Android-development-tutorial-installingAndroid-studio
https://www.youtube.com/playlist?list=PLS1QulWo1RIbb1cYyzZpLFCKvdYV_yJ-E
Https://www.udemy.com/java-tutorial/
Https://www.w3schools.com/xml/
Http://www.Androidguys.com/

EXERCISE 4 – ANDROID APPLICATION TO DRAW BASIC GRAPHICAL PRIMITIVES

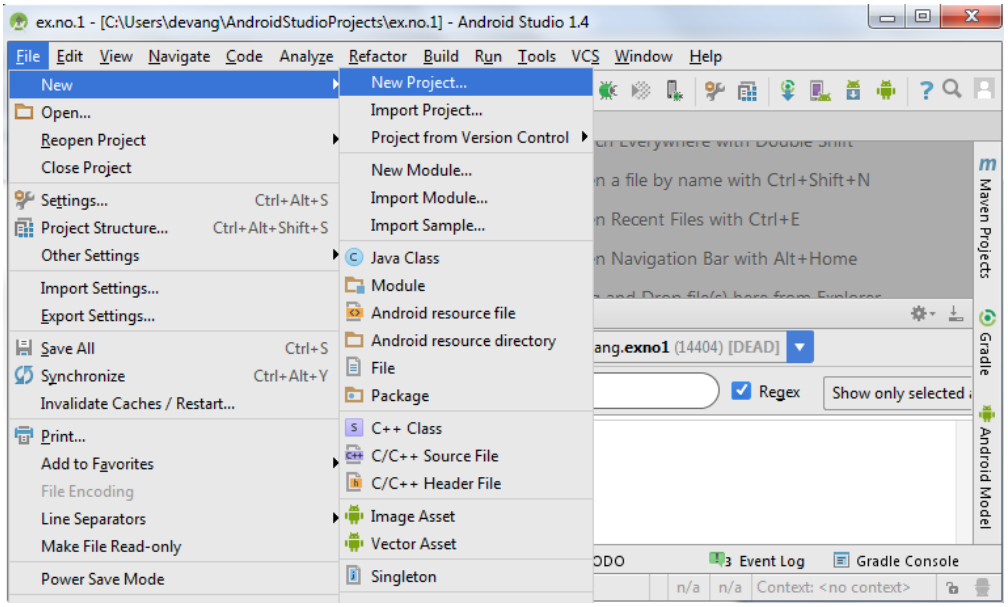
Aim:

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

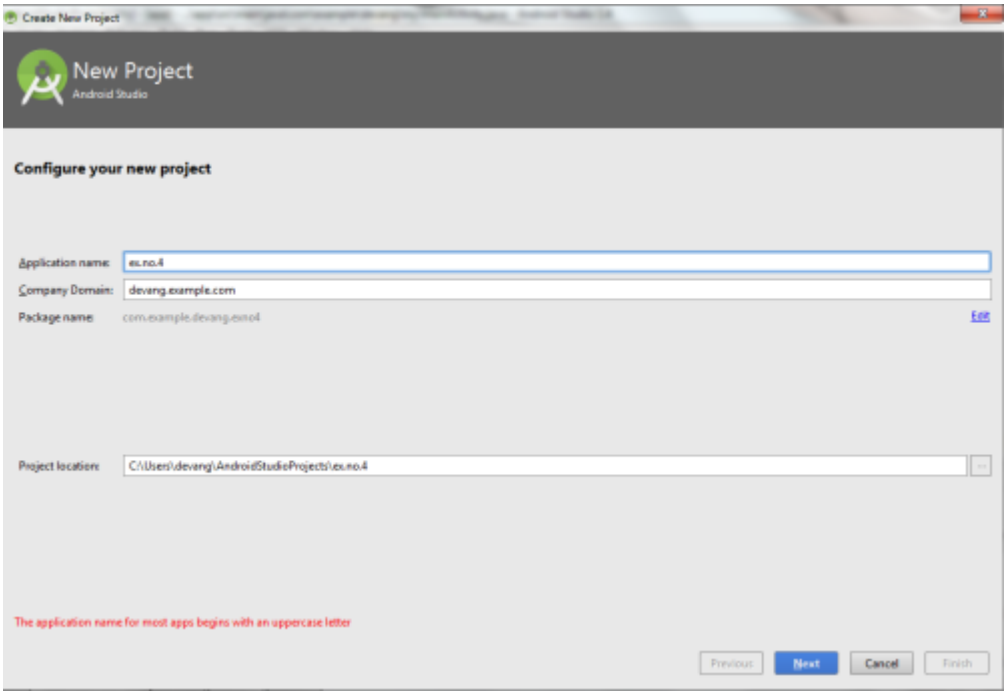
Procedure:

Creating a New project:

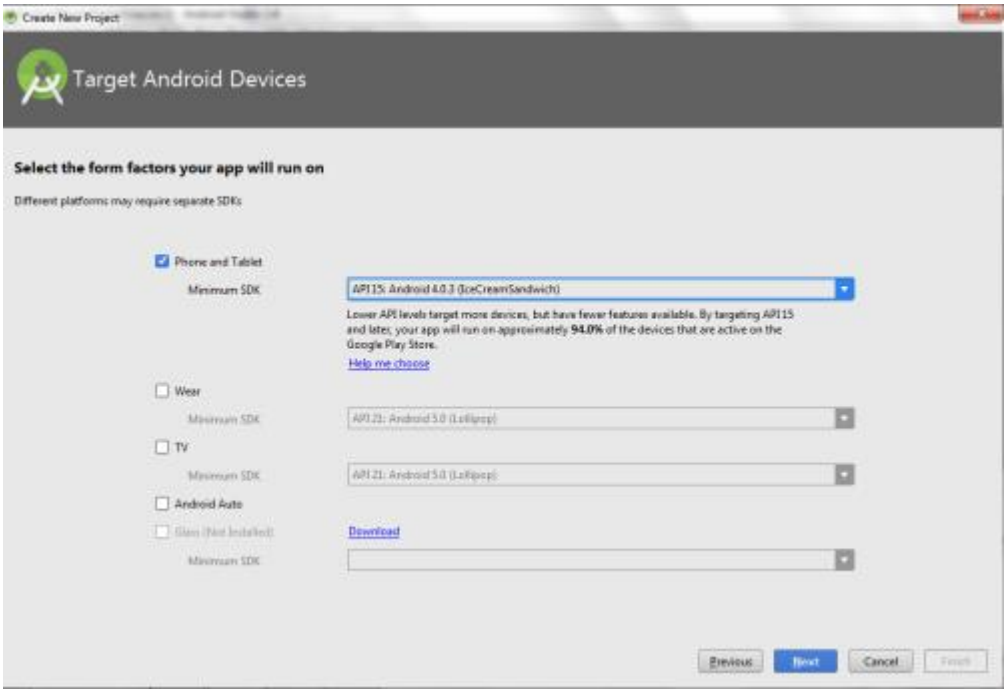
- Open Android Studio and then click on **File -> New -> New project.**



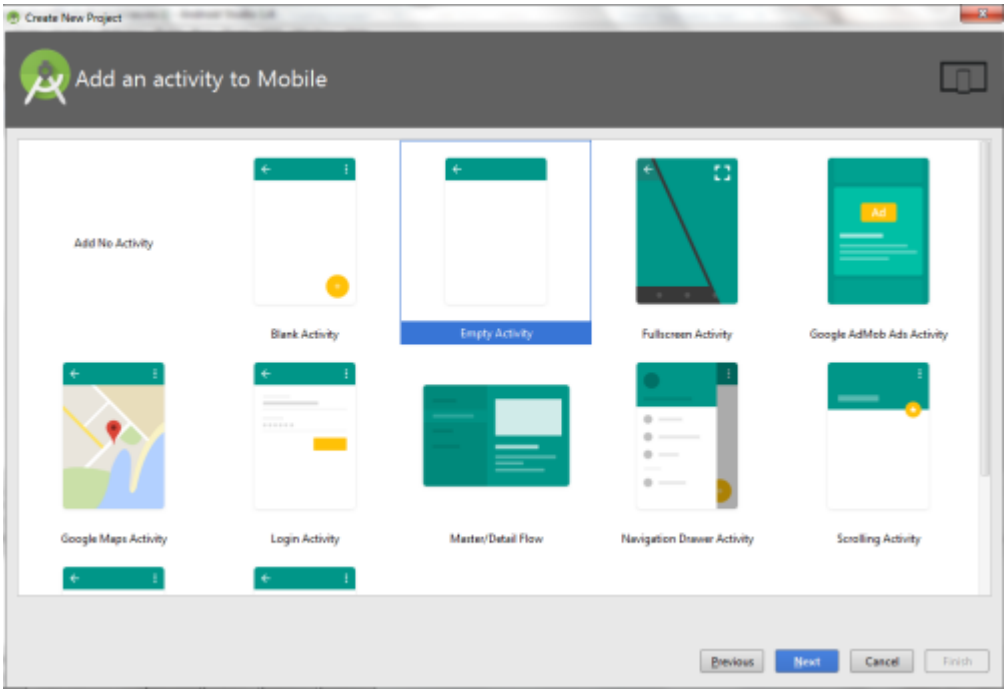
- Then type the Application name as "**ex.no.4"** and click **Next.**



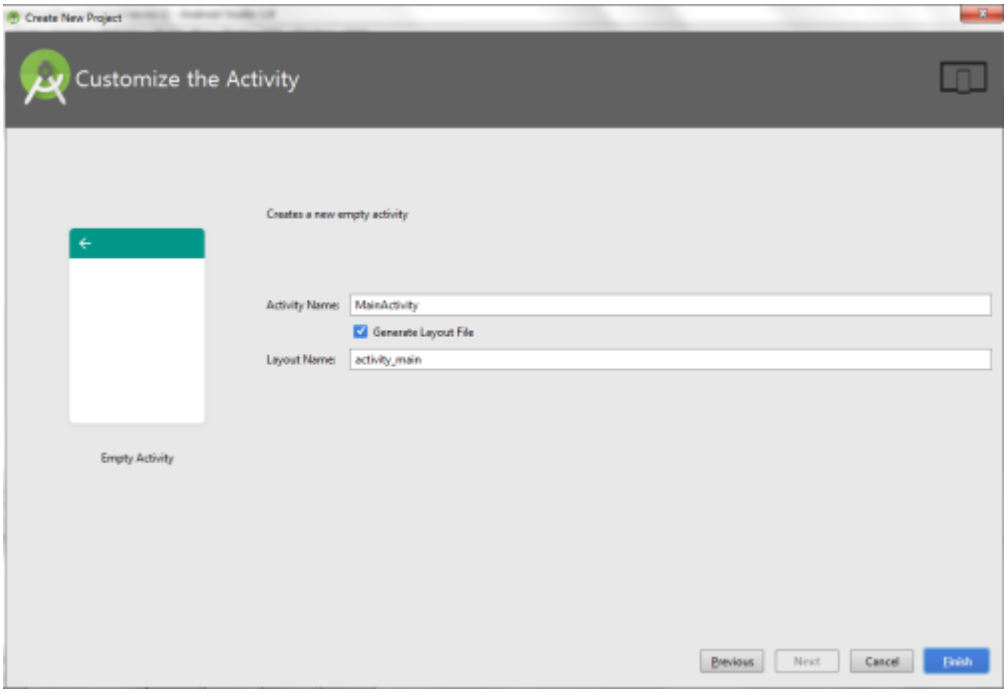
- Then select the **Minimum SDK** as shown below and click **Next**.



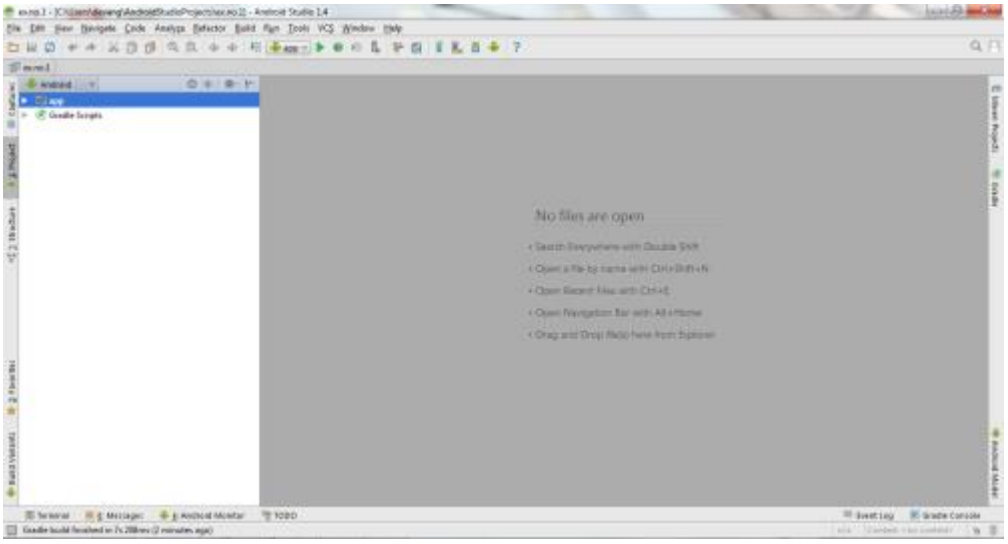
- Then select the **Empty Activity** and click **Next.**



- Finally click F**inish**.

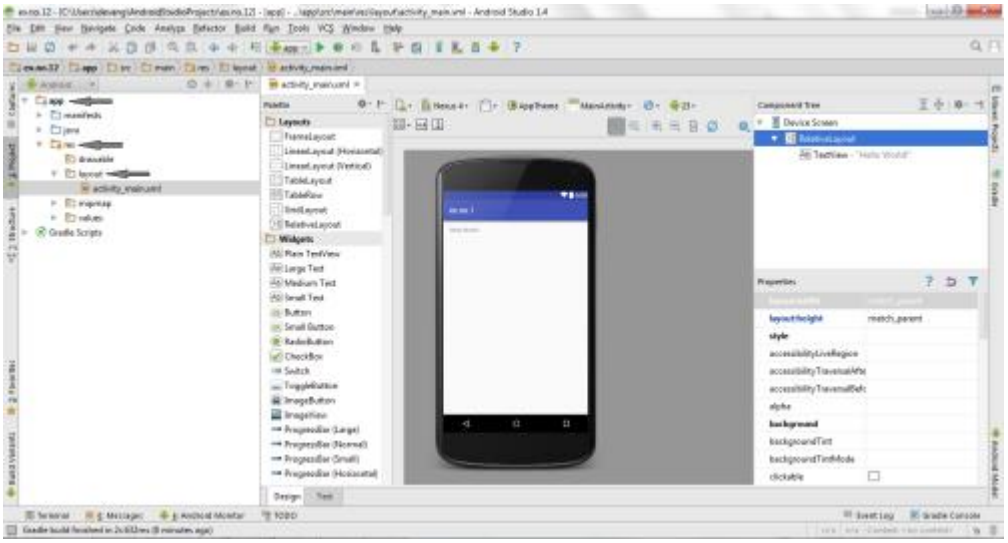


- It will take some time to build and load the project.
- After completion it will look as given below.

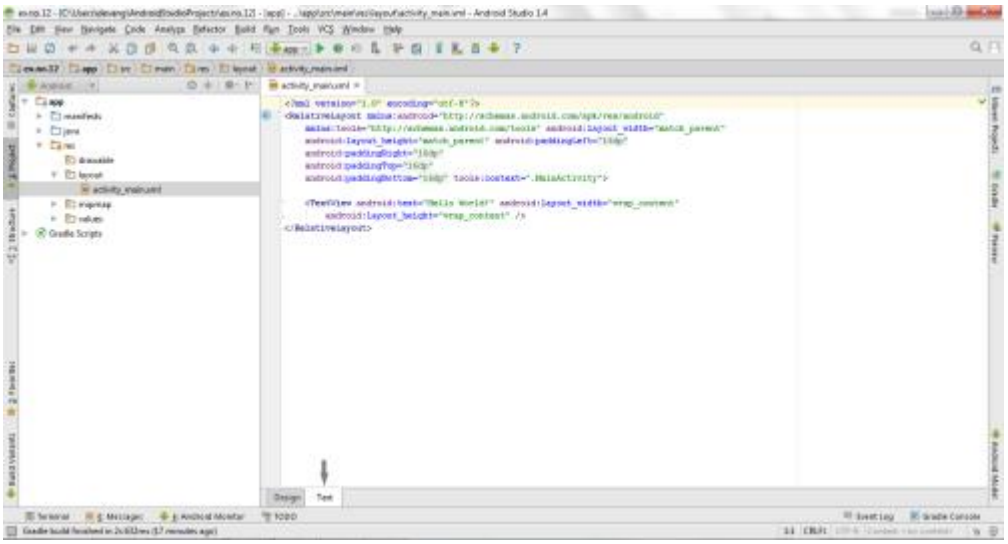


Designing layout for the Android Application:

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



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

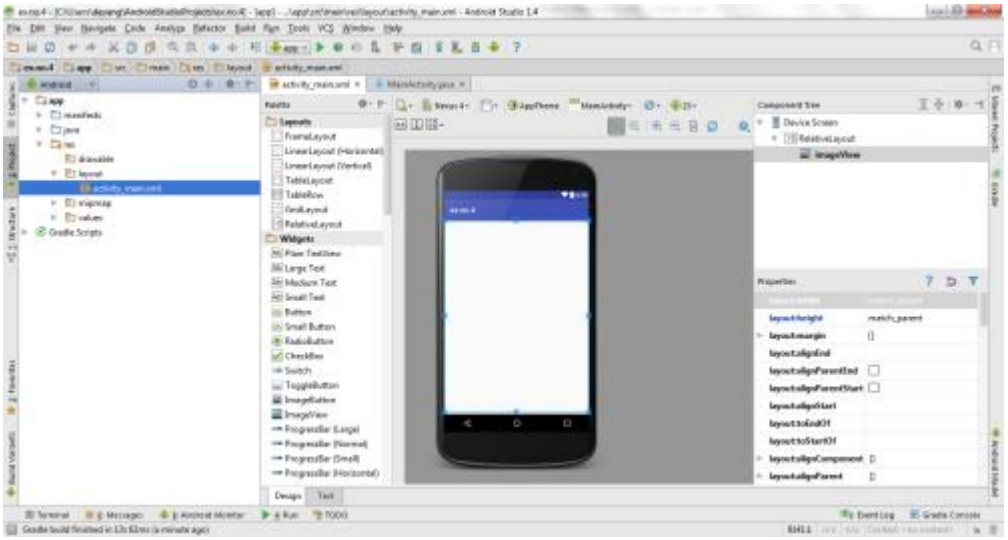


- Then delete the code which is there and type the code as given below.

Code for Activity_main.xml:

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

    <ImageView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/imageView" />
</RelativeLayout>
```

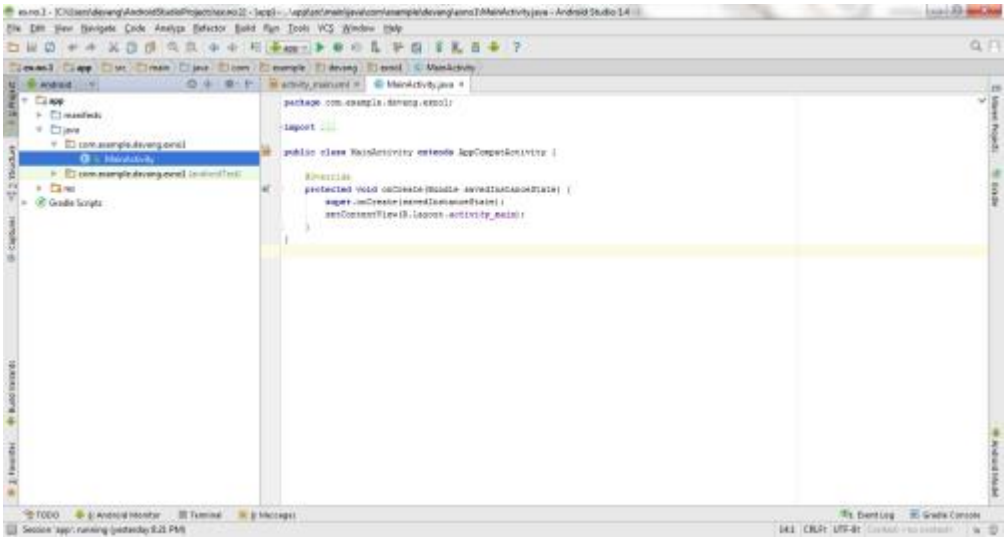
- Now click on **Design** and your application will look as given below.



- So now the designing part is completed.

Java Coding for the Android Application:

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



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

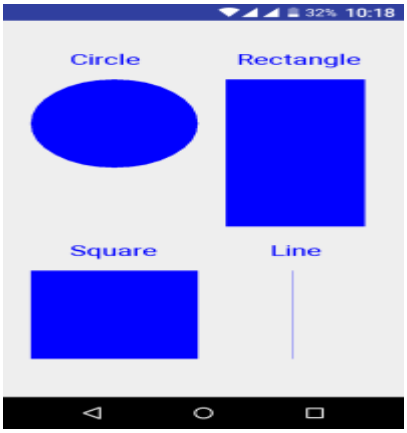
Code for MainActivity.java:

```
package com.example.exno4;

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

```
public class MainActivity extends Activity
{
    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        //Creating a Bitmap
        Bitmap bg = Bitmap.createBitmap(720, 1280, Bitmap.Config.ARGB_8888);

        //Setting the Bitmap as background for the ImageView
        ImageView i = (ImageView) findViewById(R.id.imageView);
        i.setBackgroundDrawable(new BitmapDrawable(bg));

        //Creating the Canvas Object
        Canvas canvas = new Canvas(bg);

        //Creating the Paint Object and set its color & TextSize
        Paint paint = new Paint();
        paint.setColor(Color.BLUE);
        paint.setTextSize(50);

        //To draw a Rectangle
        canvas.drawText("Rectangle", 420, 150, paint);
        canvas.drawRect(400, 200, 650, 700, paint);

        //To draw a Circle
        canvas.drawText("Circle", 120, 150, paint);
        canvas.drawCircle(200, 350, 150, paint);

        //To draw a Square
        canvas.drawText("Square", 120, 800, paint);
        canvas.drawRect(50, 850, 350, 1150, paint);

        //To draw a Line
        canvas.drawText("Line", 480, 800, paint);
        canvas.drawLine(520, 850, 520, 1150, paint);
    }
}
```

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

Output:



Result:

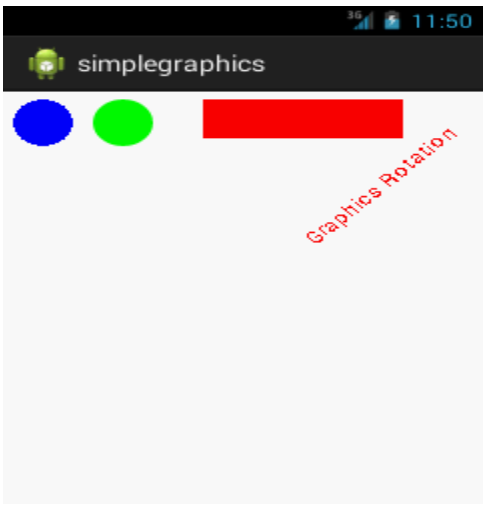
Thus a Simple Android Application that draws basic Graphical Primitives on the screen is developed and executed successfully.

EXERCISE 4.1:

Execute the above exercise of graphics in Android studio and show the exact GUI output? [2]

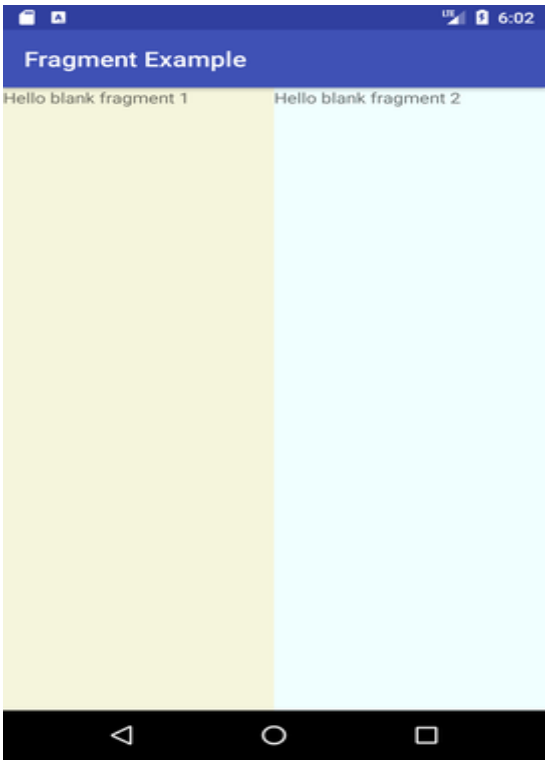
TASK 4.1

Develop an android application to display simple graphics i.e. Oval, Rectangle with different colors.



EXERCISE 4.2:

Develop an android application to display two fragments. [8]



RESOURCES:

https://www.codingconnect.net/mobile-application-development-lab/
https://www.javatpoint.com/android-tutorial
https://www.tutorialspoint.com/android
https://developer.android.com/guide
Https://developer.Android.com/training/basics/firstapp/creating-project
Https://www.raywenderlich.com/120177/beginning-Android-development-tutorial-installingAndroid-studio
https://www.youtube.com/playlist?list=PLS1QulWo1RIbb1cYyzZpLFCKvdYV_yJ-E
Https://www.udemy.com/java-tutorial/
Https://www.w3schools.com/xml/
Http://www.Androidguys.com/

EXERCISE 5 – SIMPLE ANDROID APPLICATION THAT MAKES USE OF DATABASE

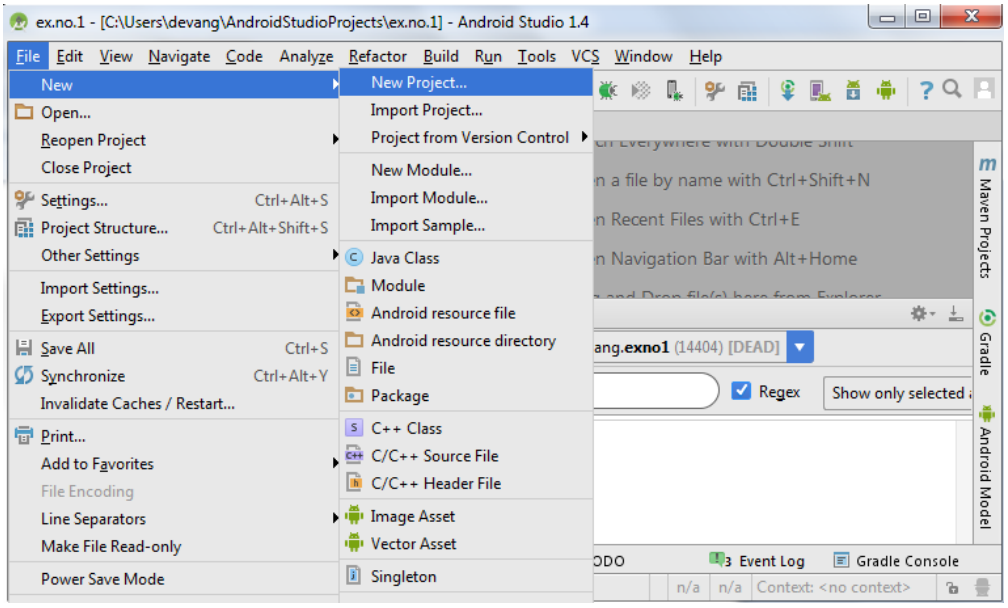
Aim:

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

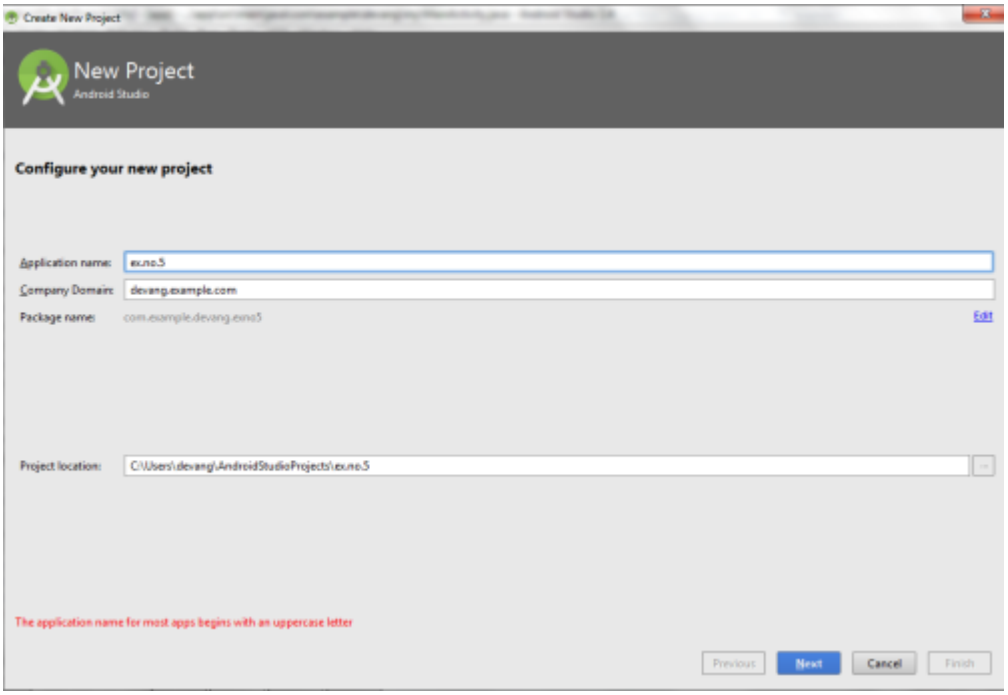
Procedure:

Creating a New project:

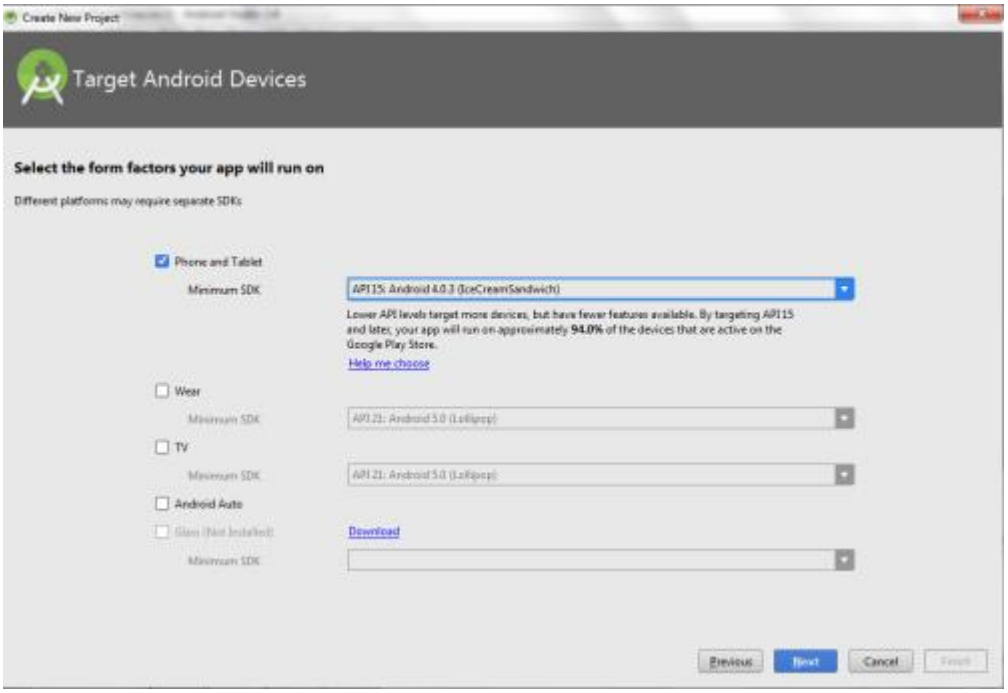
- Open Android Studio and then click on **File -> New -> New project.**



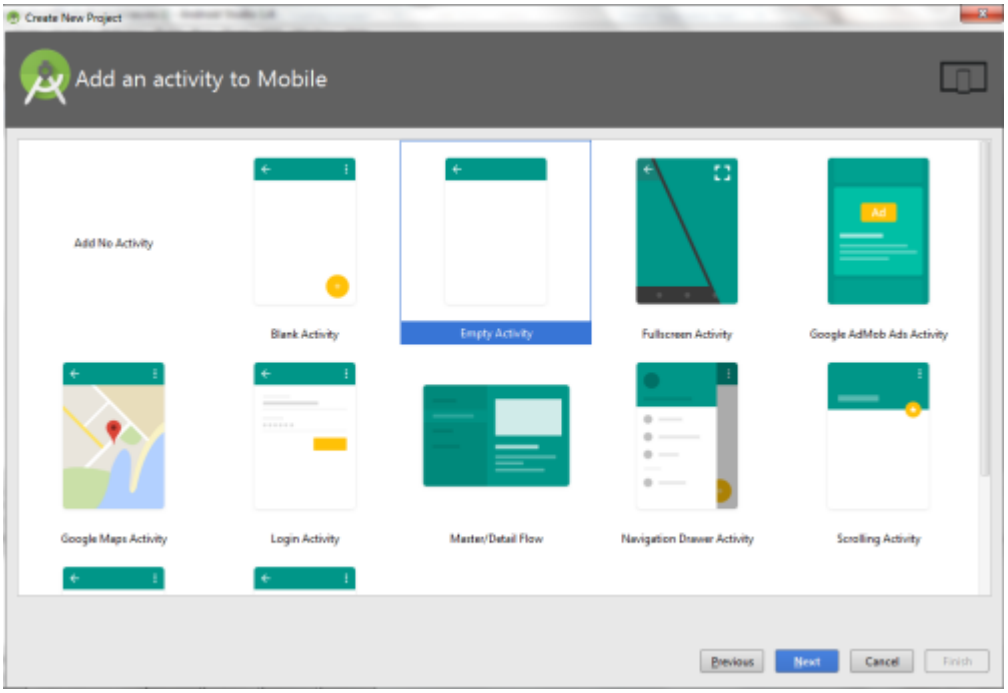
- Then type the Application name as "**ex.no.5"** and click **Next.**



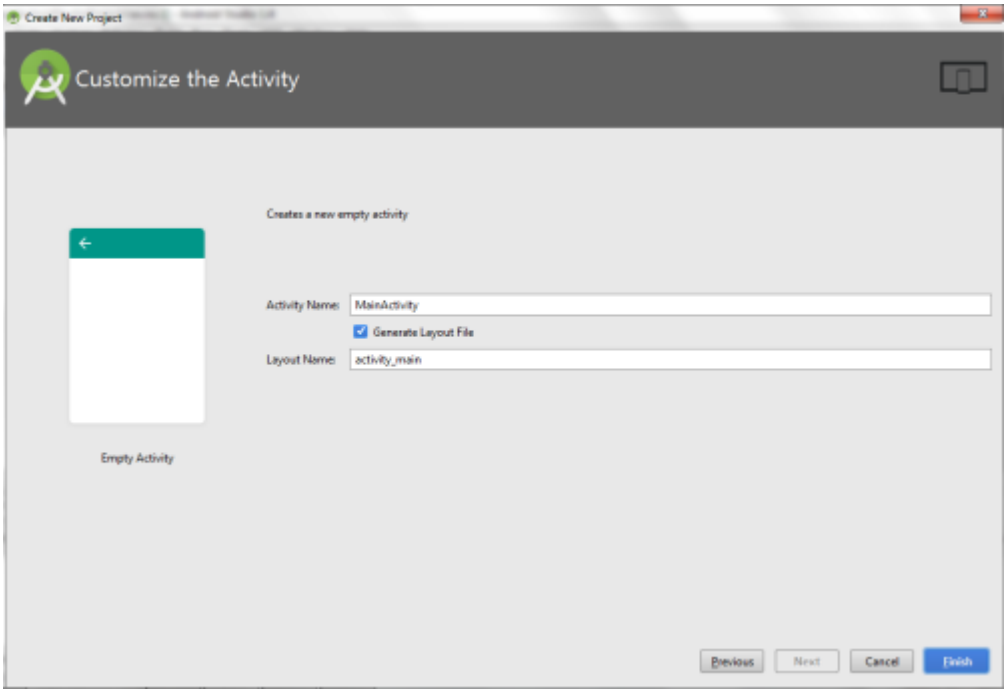
- Then select the **Minimum SDK** as shown below and click **Next**.



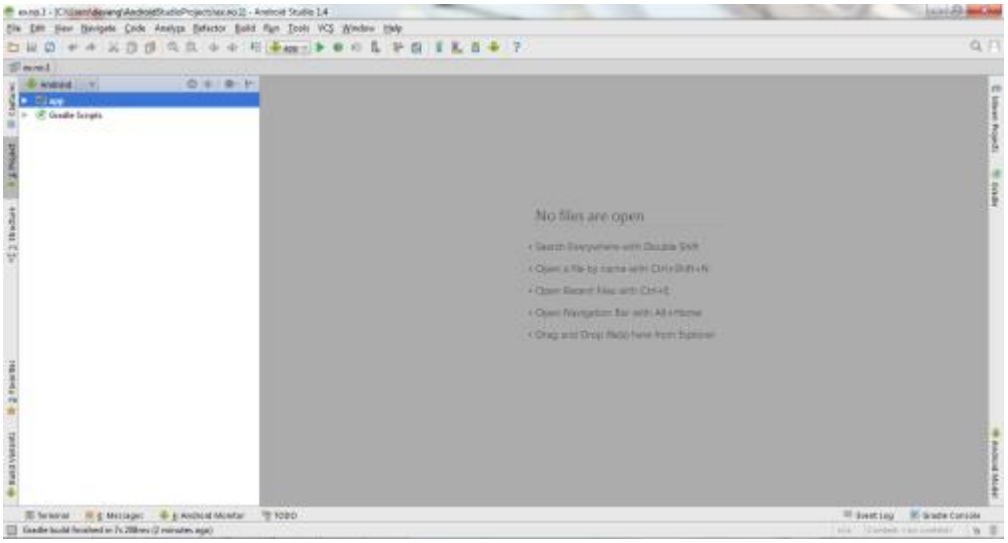
- Then select the **Empty Activity** and click **Next.**



- Finally click F**inish**.

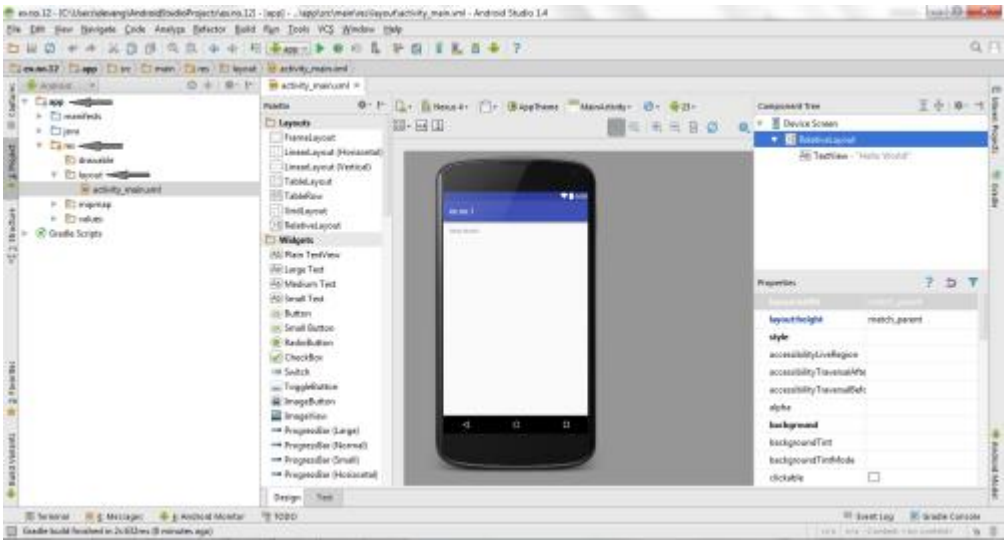


- It will take some time to build and load the project.
- After completion it will look as given below.

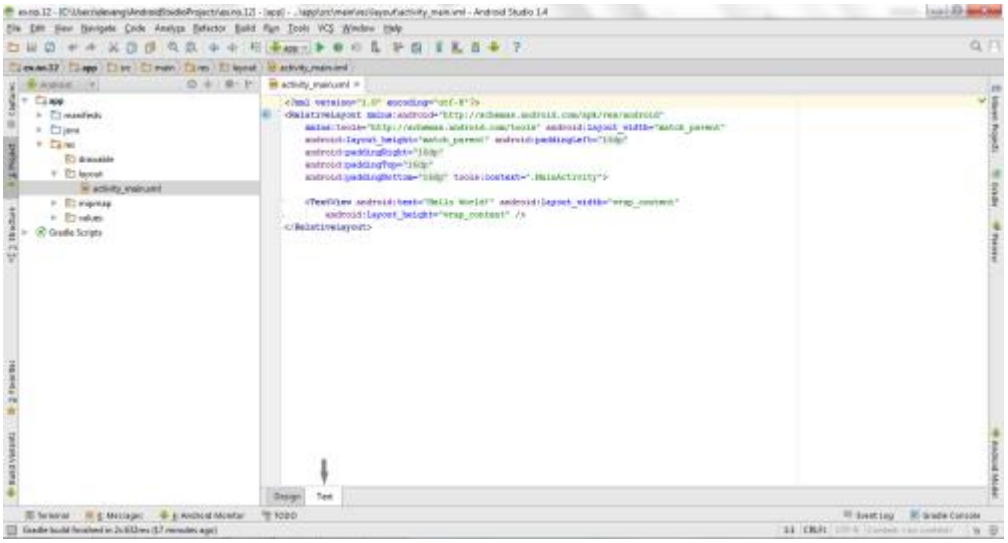


Designing layout for the Android Application:

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



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



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

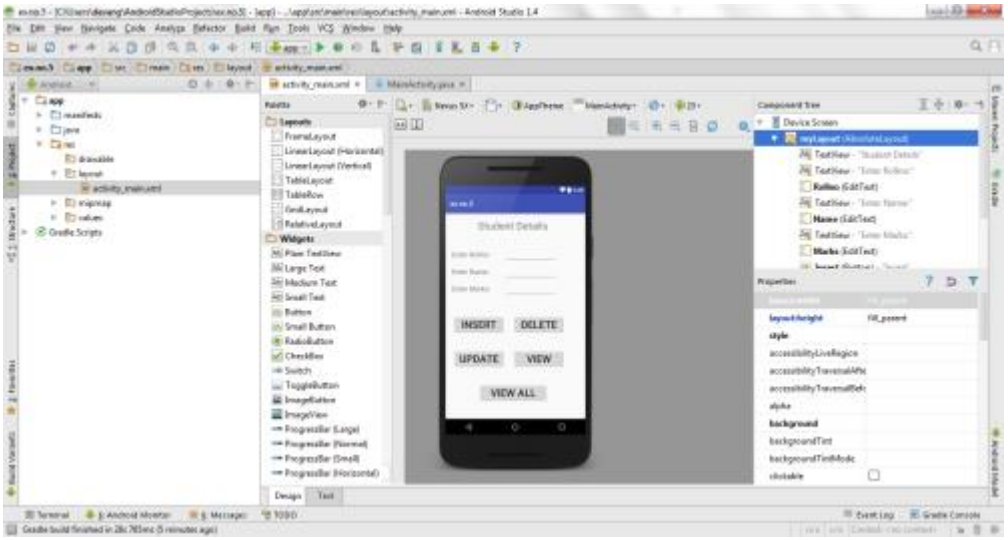
Code for Activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<AbsoluteLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_x="50dp"
        android:layout_y="20dp"
        android:text="Student Details"
        android:textSize="30sp" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_x="20dp"
        android:layout_y="110dp"
        android:text="Enter Rollno:"
        android:textSize="20sp" />

    <EditText
        android:id="@+id/Rollno"
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:layout_x="175dp"
        android:layout_y="100dp"
        android:inputType="number"
        android:textSize="20sp" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_x="20dp"
        android:layout_y="160dp"
        android:text="Enter Name:"
        android:textSize="20sp" />
```

```
<EditText
    android:id="@+id/Name"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_x="175dp"
    android:layout_y="150dp"
    android:inputType="text"
    android:textSize="20sp" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_x="20dp"
    android:layout_y="210dp"
    android:text="Enter Marks:"
    android:textSize="20sp" />

<EditText
    android:id="@+id/Marks"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_x="175dp"
    android:layout_y="200dp"
    android:inputType="number"
    android:textSize="20sp" />

<Button
    android:id="@+id/Insert"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_x="25dp"
    android:layout_y="300dp"
    android:text="Insert"
    android:textSize="30dp" />

<Button
    android:id="@+id/Delete"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_x="200dp"
    android:layout_y="300dp"
    android:text="Delete"
    android:textSize="30dp" />

<Button
    android:id="@+id/Update"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_x="25dp"
    android:layout_y="400dp"
    android:text="Update"
    android:textSize="30dp" />

<Button
    android:id="@+id/View"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_x="200dp"
```

```
        android:layout_y="400dp"
        android:text="View"
        android:textSize="30dp" />

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

</AbsoluteLayout>
```

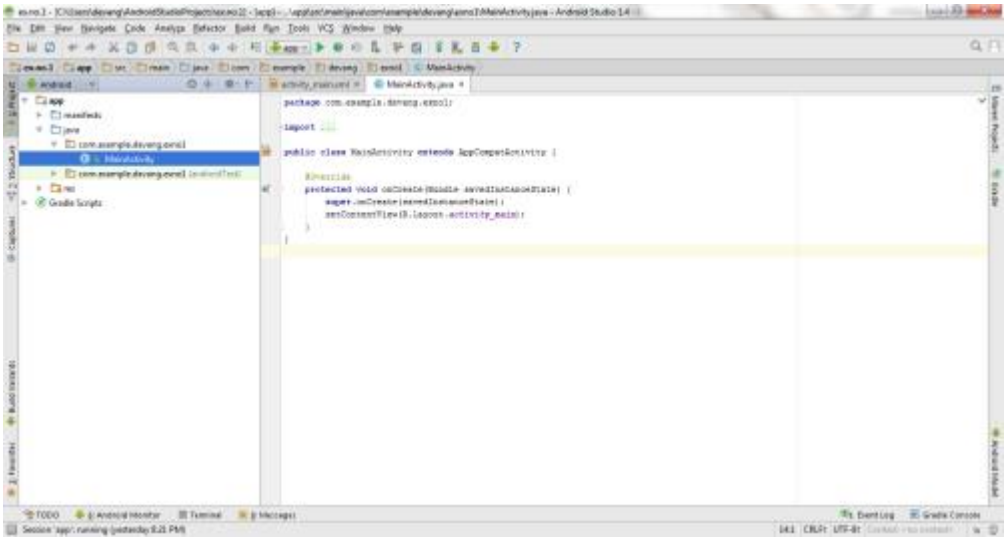
- Now click on **Design** and your application will look as given below.



- So now the designing part is completed.

Java Coding for the Android Application:

- 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);
```

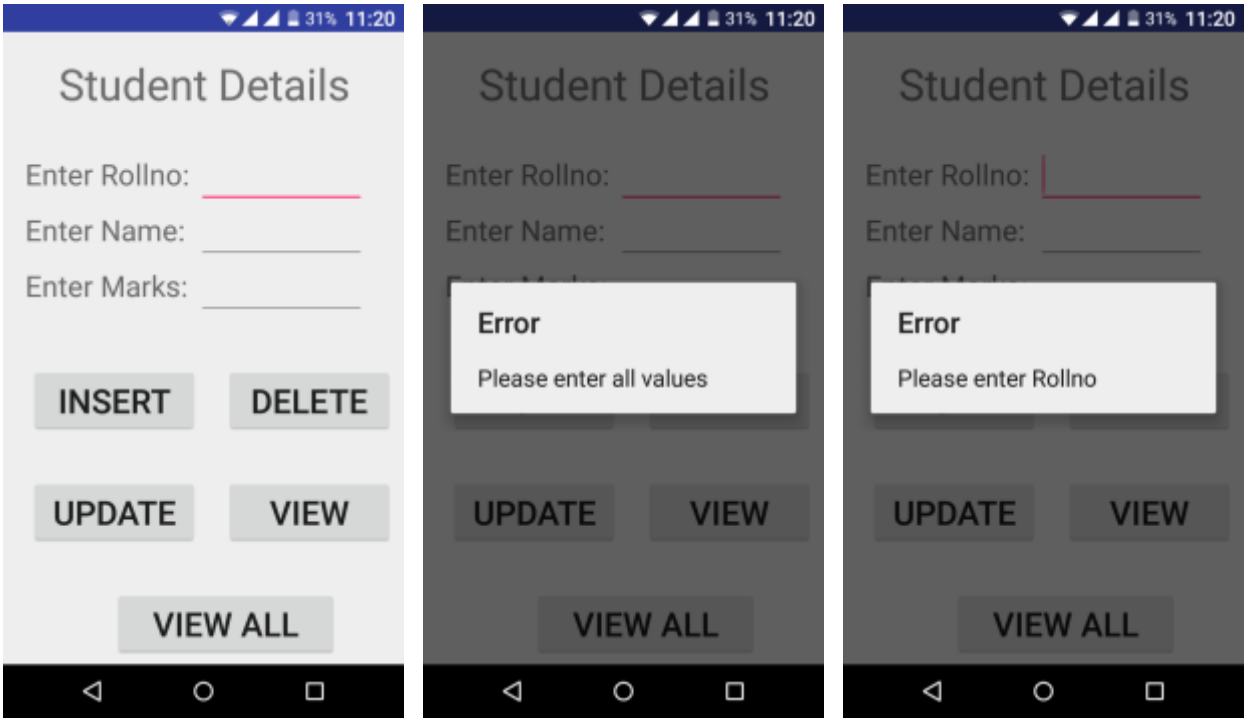
```
        Update.setOnClickListener(this);
        View.setOnClickListener(this);
        ViewAll.setOnClickListener(this);

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

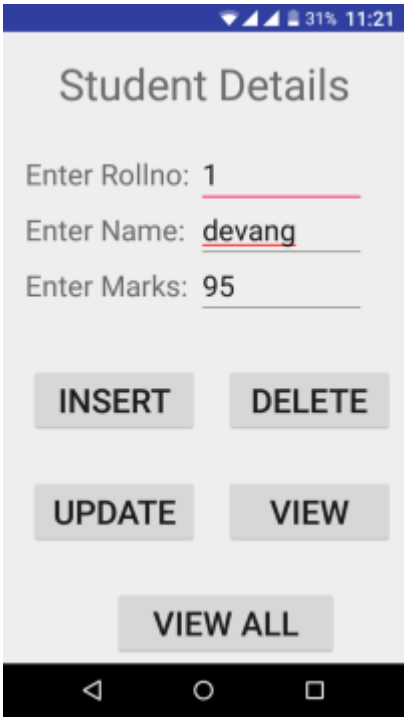
```
            Cursor c=db.rawQuery("SELECT * FROM student WHERE
rollno='"+Rollno.getText()+"'", null);
            if(c.moveToFirst()) {
                db.execSQL("UPDATE student SET name='" + Name.getText() +
"',marks='" + Marks.getText() +
                        "' WHERE rollno='"+Rollno.getText()+"'");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

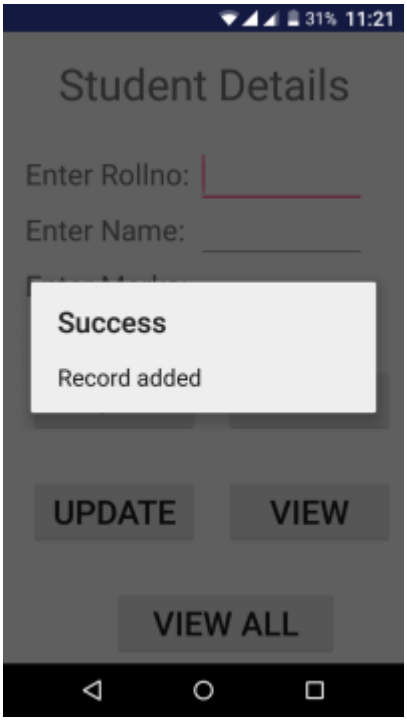
```
    }
    public void clearText()
    {
        Rollno.setText("");
        Name.setText("");
        Marks.setText("");
        Rollno.requestFocus();
    }
}
```

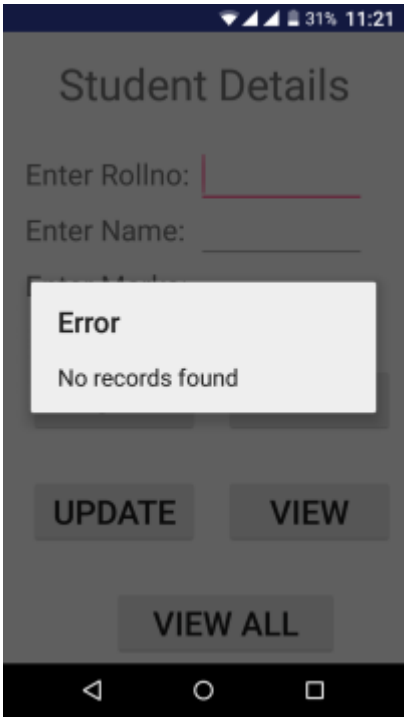
- So now the Coding part is also completed.
- Now run the application to see the output.

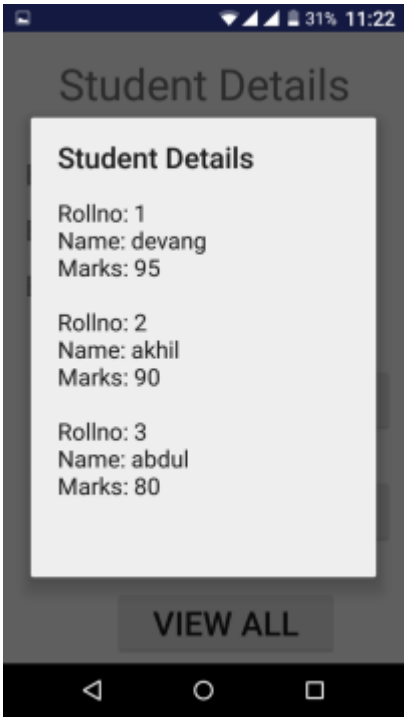
Output:

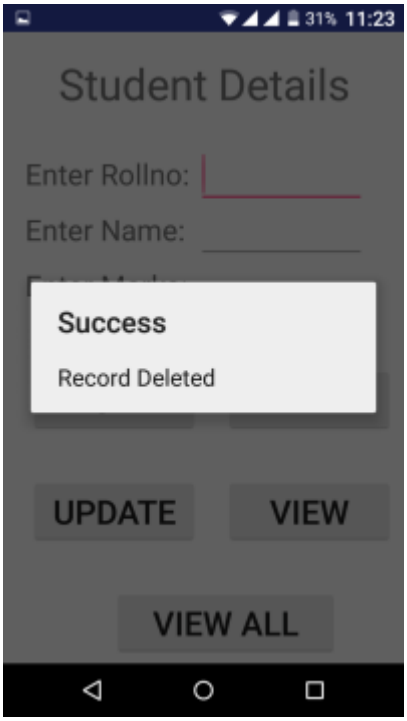



31% 11:21
Student Details
Enter Rollno: 1
Enter Name: devang
Enter Marks: 95
INSERT
DELETE
UPDATE
VIEW
VIEW ALL

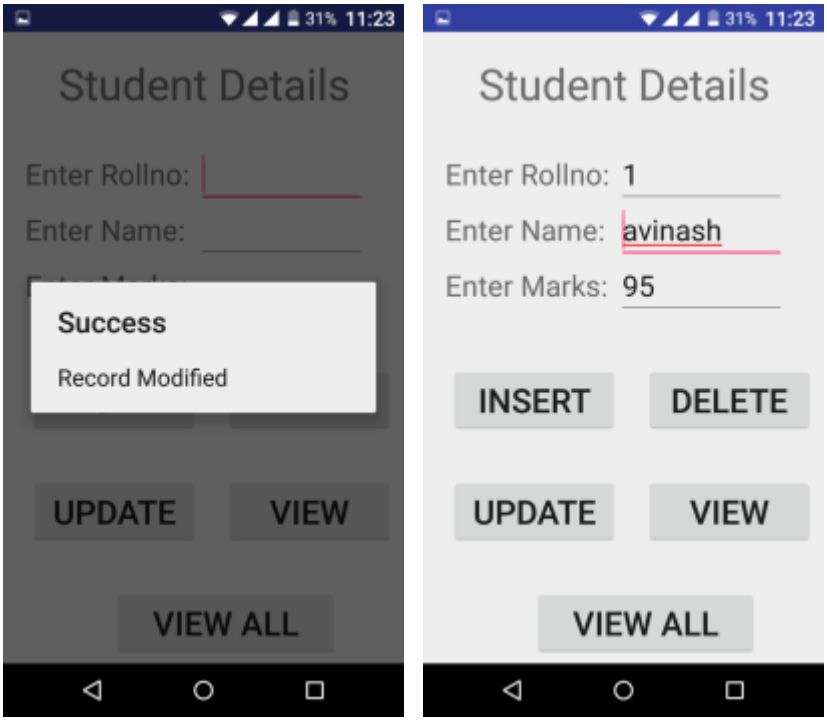

31% 11:21
Student Details
Enter Rollno:
Enter Name:
Success
Record added
UPDATE
VIEW
VIEW ALL


31% 11:21
Student Details
Enter Rollno:
Enter Name:
Error
No records found
UPDATE
VIEW
VIEW ALL


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



Result:

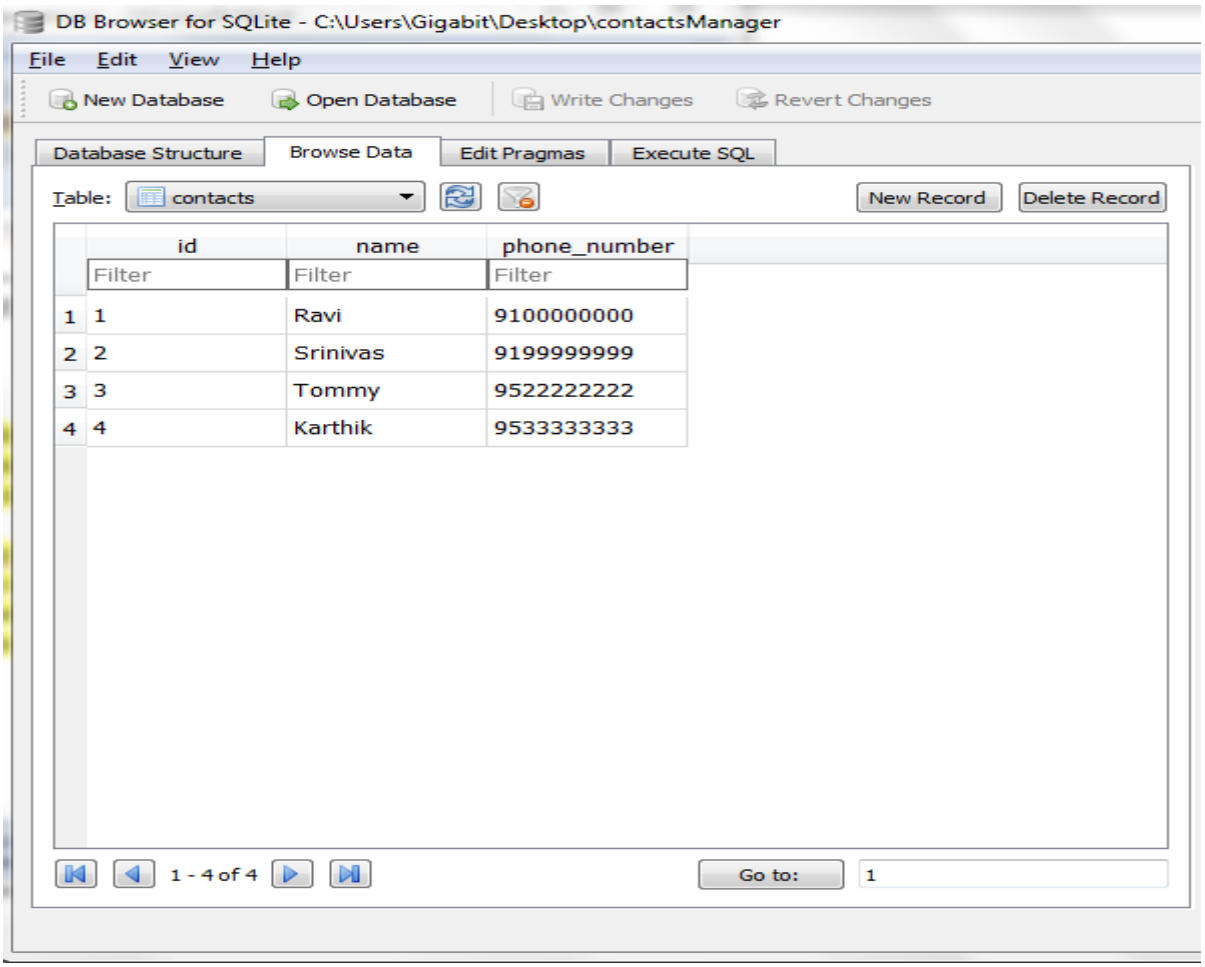
Thus a Simple Android Application that makes use of Database is developed and executed successfully.

EXERCISE 5.1:

Execute the above exercise of database in Android studio and show the exact GUI output? [10]

TASK 5.1

Develop an android application to display all the data of students in a LISTVIEW widget by using Android Sqlite tutorial.



RESOURCES:

https://www.codingconnect.net/mobile-application-development-lab/
https://www.javatpoint.com/android-tutorial
https://www.tutorialspoint.com/android
https://developer.android.com/guide
Https://developer.Android.com/training/basics/firstapp/creating-project
Https://www.raywenderlich.com/120177/beginning-Android-development-tutorial-installingAndroid-studio
https://www.youtube.com/playlist?list=PLS1QulWo1RIbb1cYyzZpLFCKvdYV_yJ-E
Https://www.udemy.com/java-tutorial/
Https://www.w3schools.com/xml/
Http://www.Androidguys.com/

EXERCISE 6 – ANDROID APPLICATION THAT MAKES USE OF RSS FEED

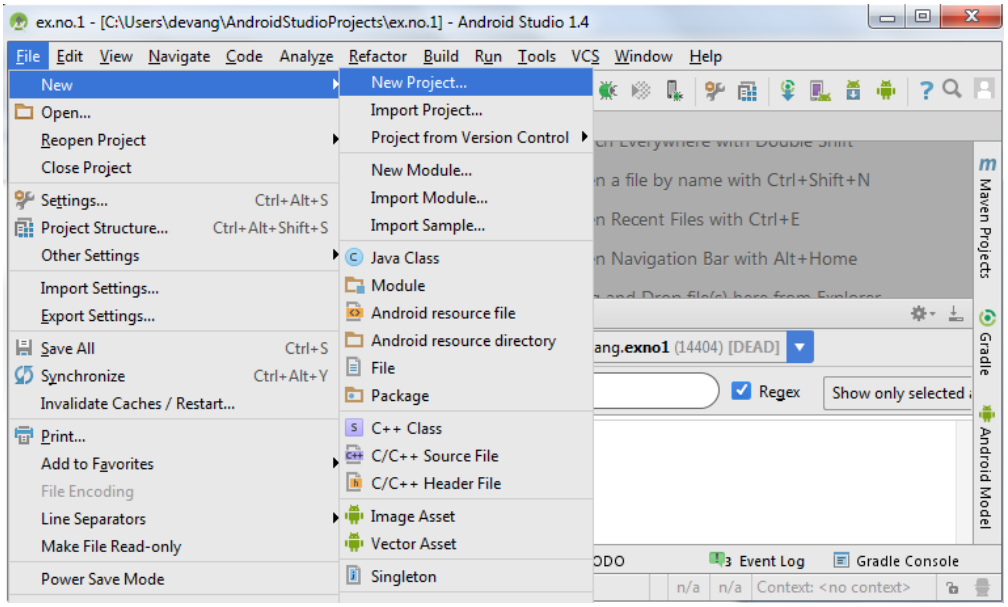
Aim:

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

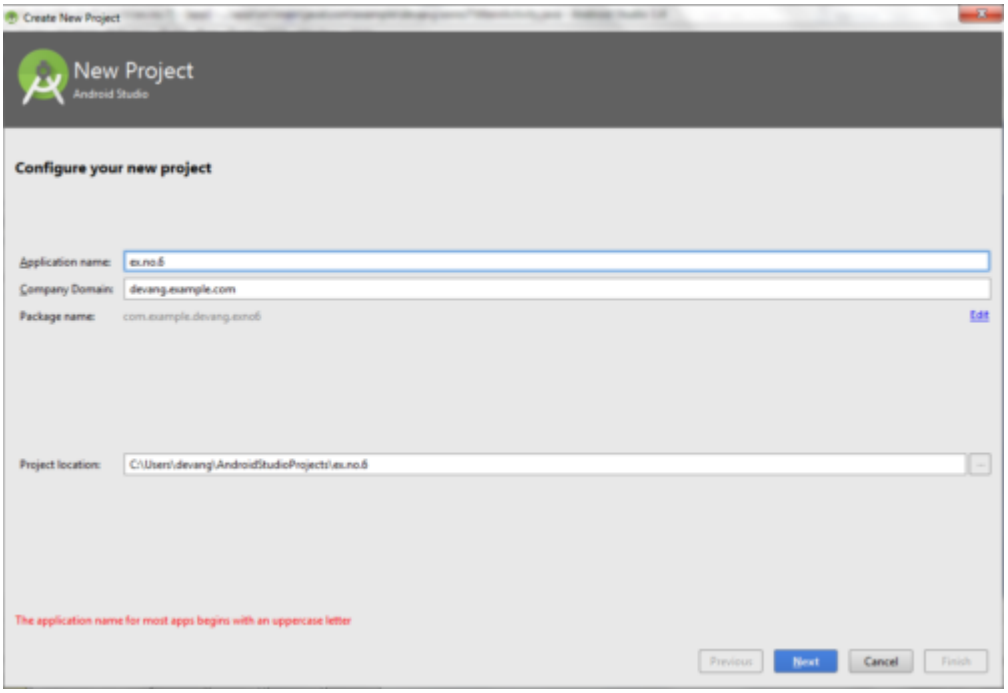
Procedure:

Creating a New project:

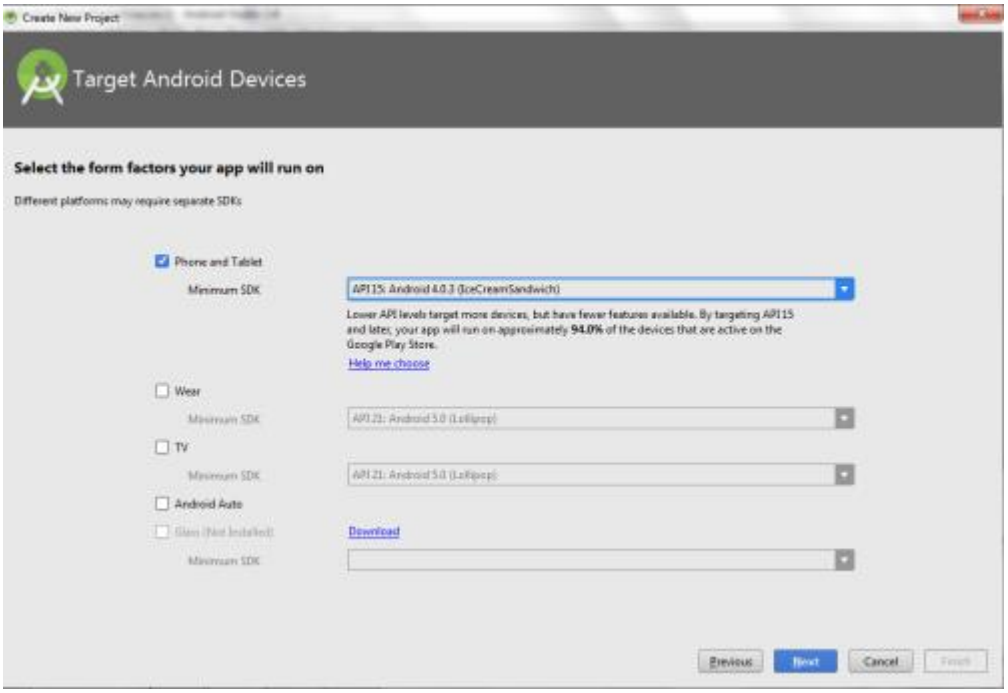
- Open Android Studio and then click on **File -> New -> New project.**



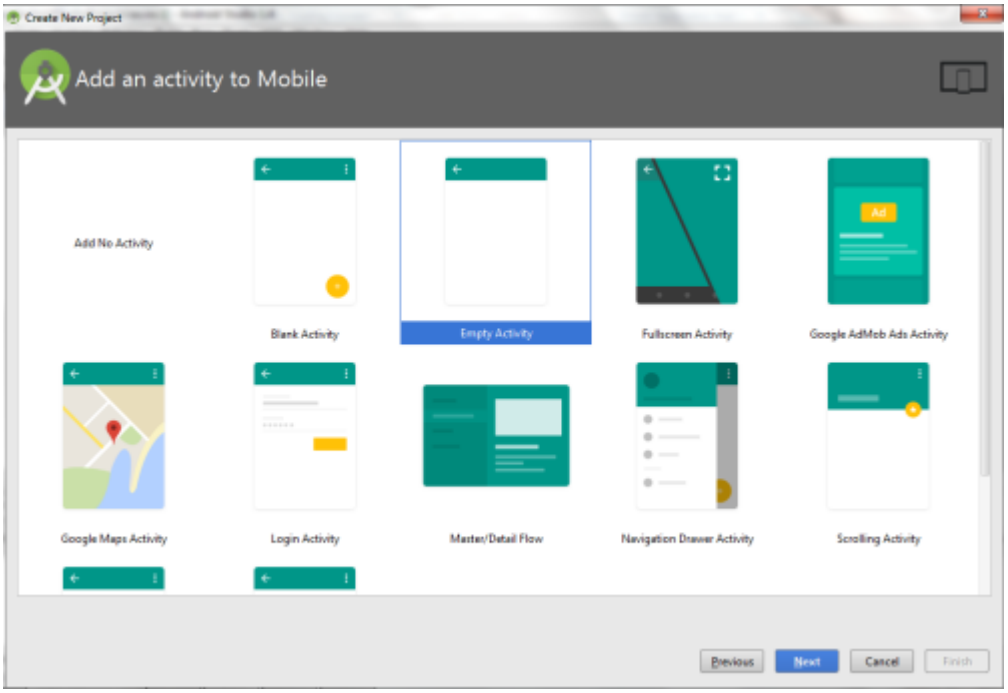
- Then type the Application name as "**ex.no.6"** and click **Next.**



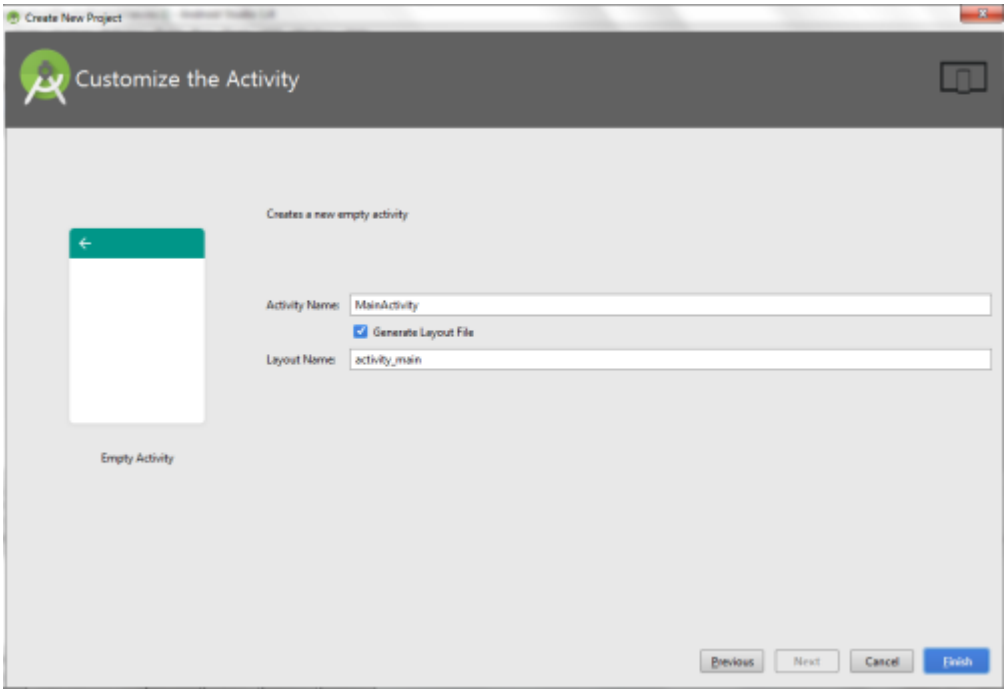
- Then select the **Minimum SDK** as shown below and click **Next**.



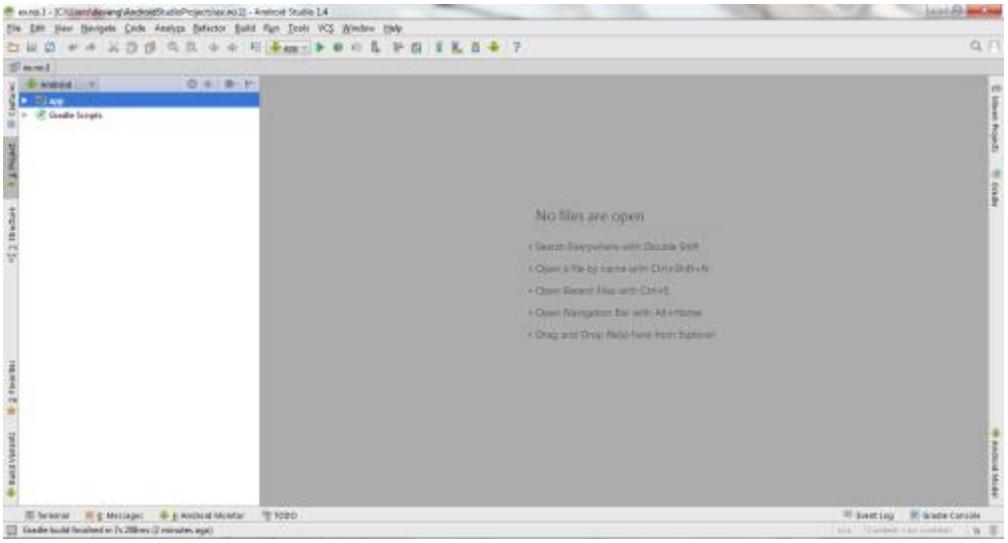
- Then select the **Empty Activity** and click **Next.**



- Finally click F**inish**.

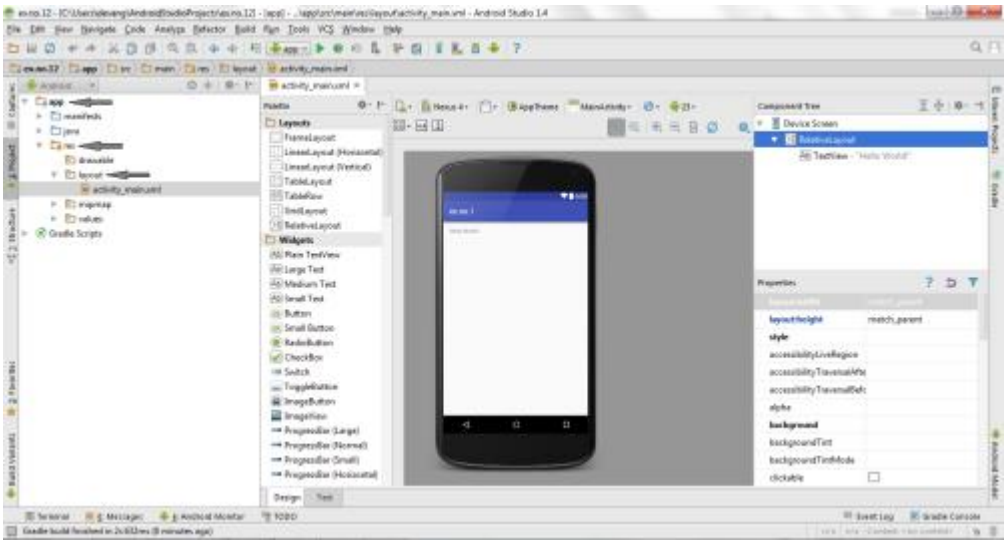


- It will take some time to build and load the project.
- After completion it will look as given below.

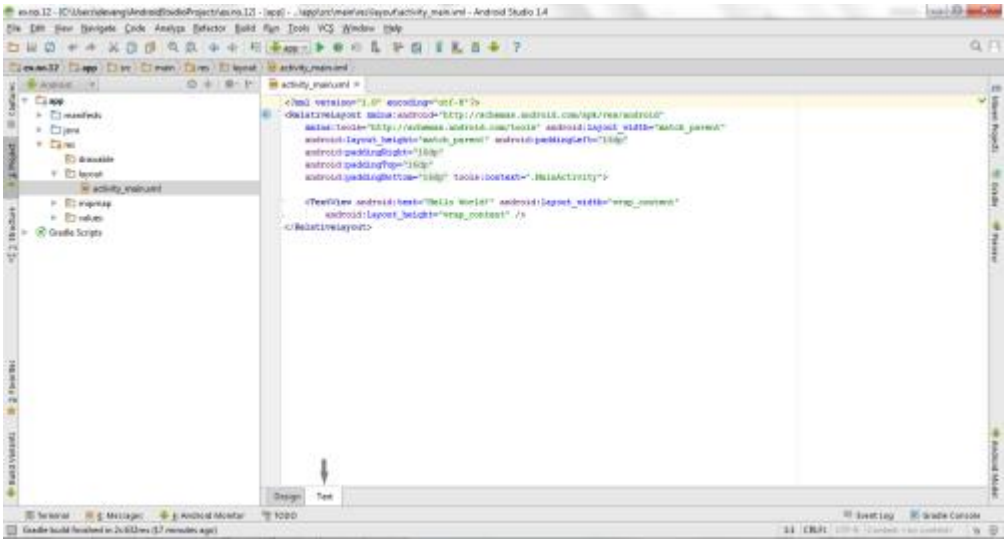


Designing layout for the Android Application:

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



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

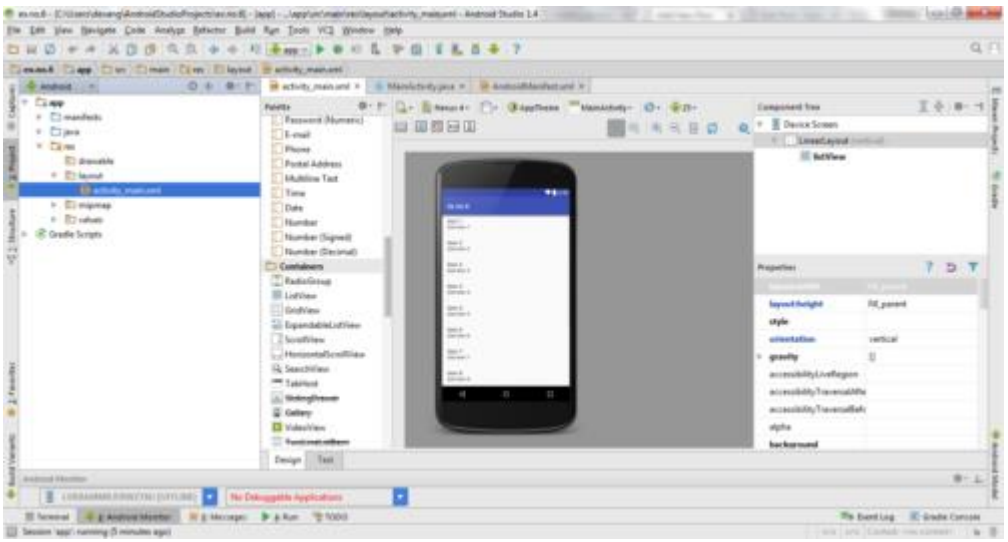


- Then delete the code which is there and type the code as given below.

Code for Activity_main.xml:

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

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

</LinearLayout>
```

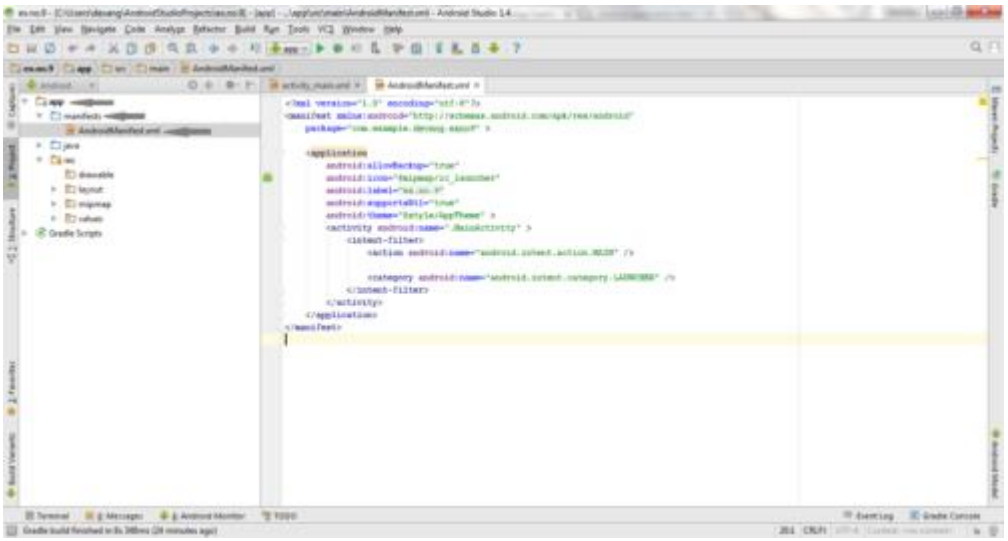
- Now click on **Design** and your application will look as given below.



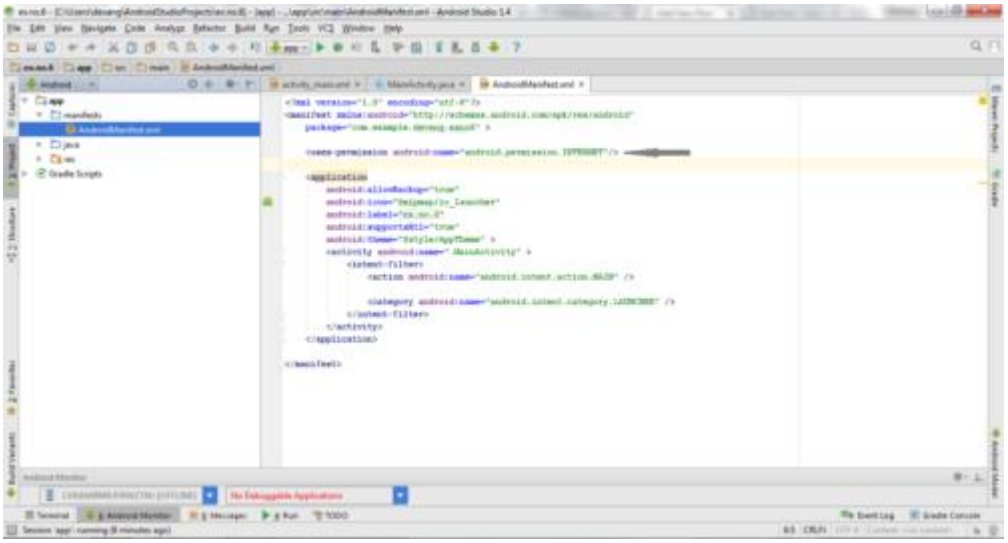
- So now the designing part is completed.

Adding permissions in Manifest for the Android Application:

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



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



Code for AndroidManifest.xml:

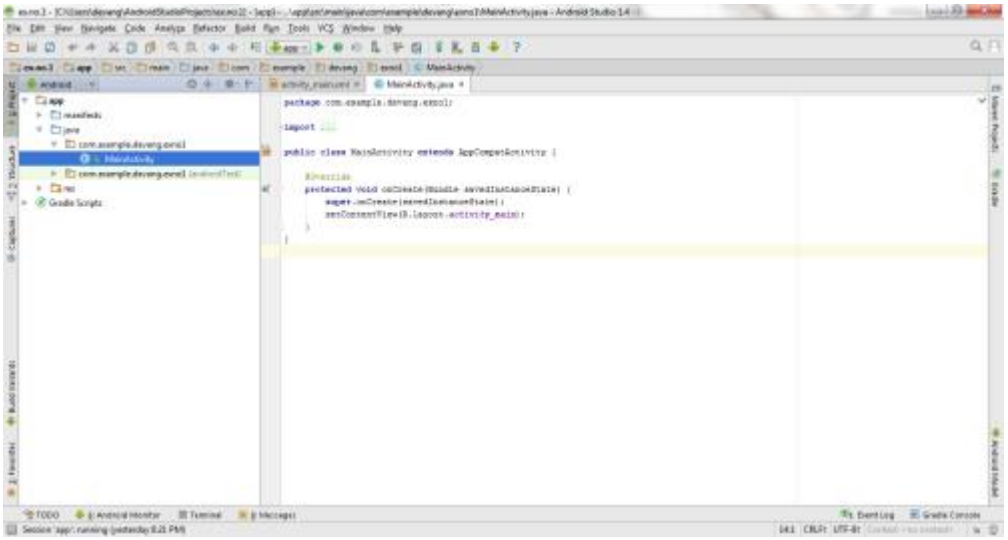
```
?
1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3      package="com.example.exno6" >
4
5      <uses-permission android:name="android.permission.INTERNET"/>
6
7      <application
8          android:allowBackup="true"
9          android:icon="@mipmap/ic_launcher"
10         android:label="@string/app_name"
11         android:supportsRtl="true"
12         android:theme="@style/AppTheme" >
           <activity android:name=".MainActivity" >
               <intent-filter>
```

```
13                  <action android:name="android.intent.action.MAIN" />
14
15                  <category android:name="android.intent.category.LAUNCHER" />
16              </intent-filter>
17          </activity>
18      </application>
19
20  </manifest>
21
22
```

- So now the Permissions are added in the Manifest.

Java Coding for the Android Application:

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



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

Code for MainActivity.java:

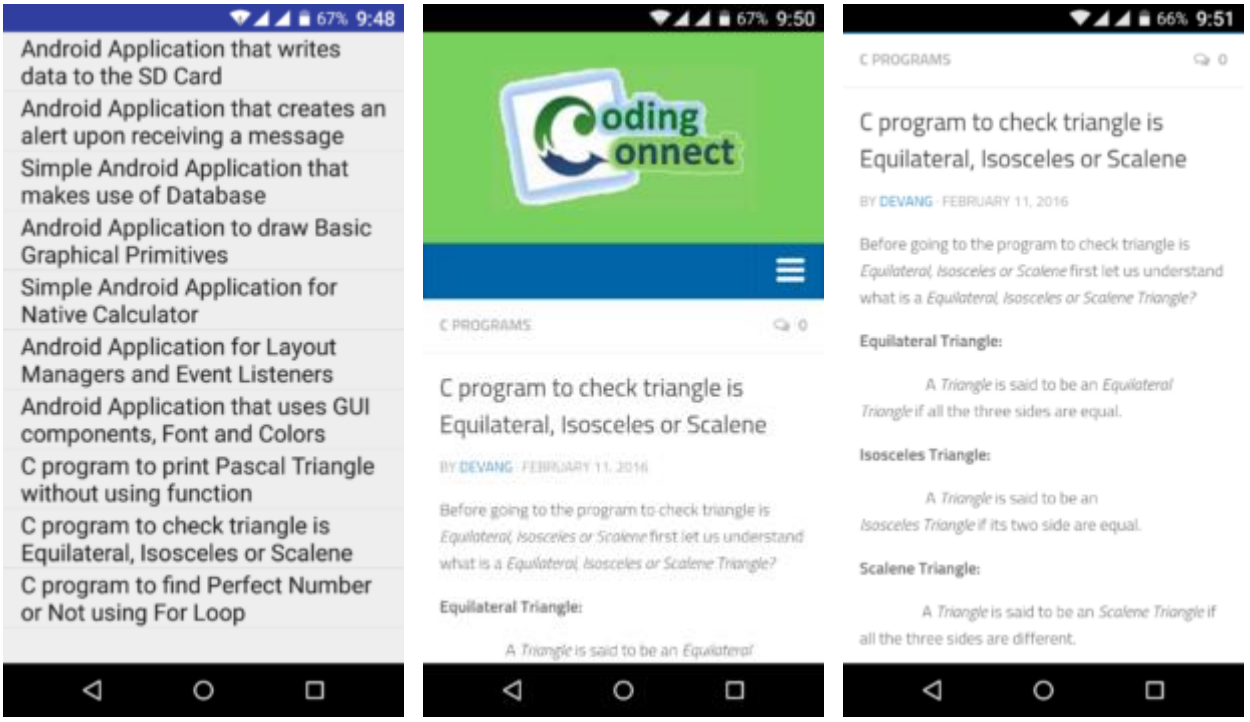
```
package com.example.exno6;

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

```
import java.util.List;

public class MainActivity extends ListActivity
{
    List headlines;
    List links;

    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        new MyAsyncTask().execute();
    }

    class MyAsyncTask extends AsyncTask<Object,Void,ArrayAdapter>
    {
        @Override
        protected ArrayAdapter doInBackground(Object[] params)
        {
            headlines = new ArrayList();
            links = new ArrayList();
            try
            {
                URL url = new URL("https://codingconnect.net/feed");
                XmlPullParserFactory factory =
XmlPullParserFactory.newInstance();
                factory.setNamespaceAware(false);
                XmlPullParser xpp = factory.newPullParser();

                // We will get the XML from an input stream
                xpp.setInput(getInputStream(url), "UTF_8");
                boolean insideItem = false;

                // Returns the type of current event: START_TAG, END_TAG,
etc..
                int eventType = xpp.getEventType();
                while (eventType != XmlPullParser.END_DOCUMENT)
                {
                    if (eventType == XmlPullParser.START_TAG)
                    {
                        if (xpp.getName().equalsIgnoreCase("item"))
                        {
                            insideItem = true;
                        }
                        else if (xpp.getName().equalsIgnoreCase("title"))
                        {
                            if (insideItem)
                                headlines.add(xpp.nextText()); //extract the
headline
                        }
                        else if (xpp.getName().equalsIgnoreCase("link"))
                        {
                            if (insideItem)
                                links.add(xpp.nextText()); //extract the link
of article
                        }
                    }
                    else if(eventType==XmlPullParser.END_TAG &&
xpp.getName().equalsIgnoreCase("item"))
                    {
```

```
                        insideItem=false;
                    }
                    eventType = xpp.next(); //move to next element
                }

            }
            catch (MalformedURLException e)
            {
                e.printStackTrace();
            }
            catch (XmlPullParserException e)
            {
                e.printStackTrace();
            }
            catch (IOException e)
            {
                e.printStackTrace();
            }
            return null;
        }
        protected void onPostExecute(ArrayAdapter adapter)
        {
            adapter = new ArrayAdapter(MainActivity.this,
android.R.layout.simple_list_item_1, headlines);
            setListAdapter(adapter);
        }
    }

    @Override
    protected void onListItemClick(ListView l, View v, int position, long id)
    {
        Uri uri = Uri.parse((links.get(position)).toString());
        Intent intent = new Intent(Intent.ACTION_VIEW, uri);
        startActivity(intent);
    }

    public InputStream getInputStream(URL url)
    {
        try
        {
            return url.openConnection().getInputStream();
        }
        catch (IOException e)
        {
            return null;
        }
    }
}
```

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

Output:



Result:

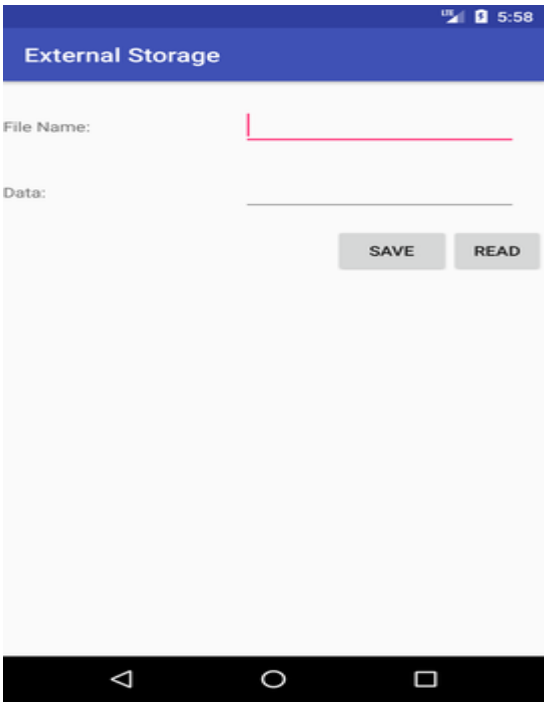
Thus Android Application that makes use of RSS Feed is developed and executed successfully.

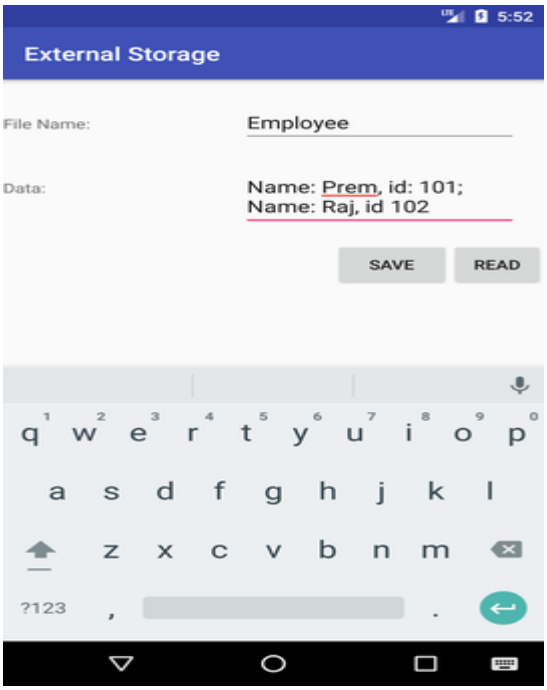
EXERCISE 6.1:

Execute the above exercise of RSS feed in Android studio and show the exact GUI output? [02]

EXERCISE 6.1:

Develop an android application to read and write data on external storage. [08]





RESOURCES:

https://www.codingconnect.net/mobile-application-development-lab/
https://www.javatpoint.com/android-tutorial
https://www.tutorialspoint.com/android
https://developer.android.com/guide
Https://developer.Android.com/training/basics/firstapp/creating-project
Https://www.raywenderlich.com/120177/beginning-Android-development-tutorial-installingAndroid-studio
https://www.youtube.com/playlist?list=PLS1QulWo1RIbb1cYyzZpLFCKvdYV_yJ-E
Https://www.udemy.com/java-tutorial/
Https://www.w3schools.com/xml/
Http://www.Androidguys.com/

EXERCISE 7 – ANDROID APPLICATION THAT IMPLEMENTS MULTI THREADING

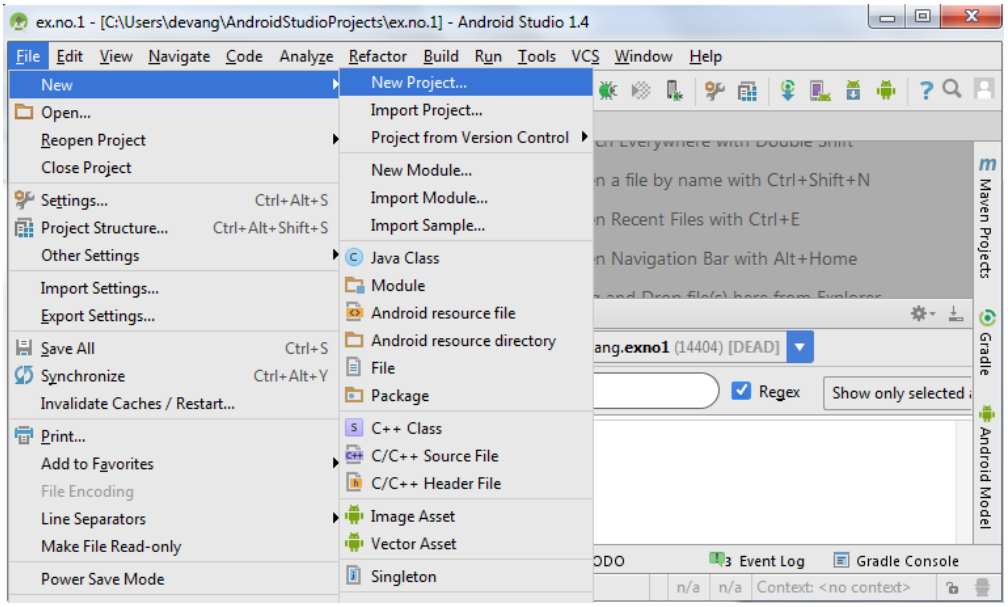
Aim:

To develop a Android Application that implements Multi threading.

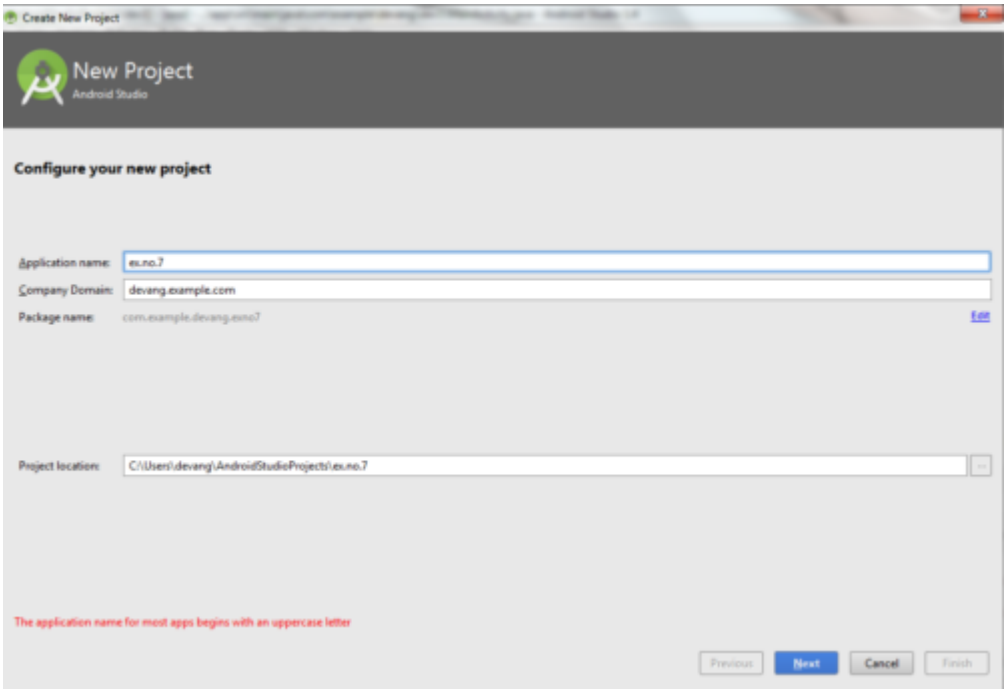
Procedure:

Creating a New project:

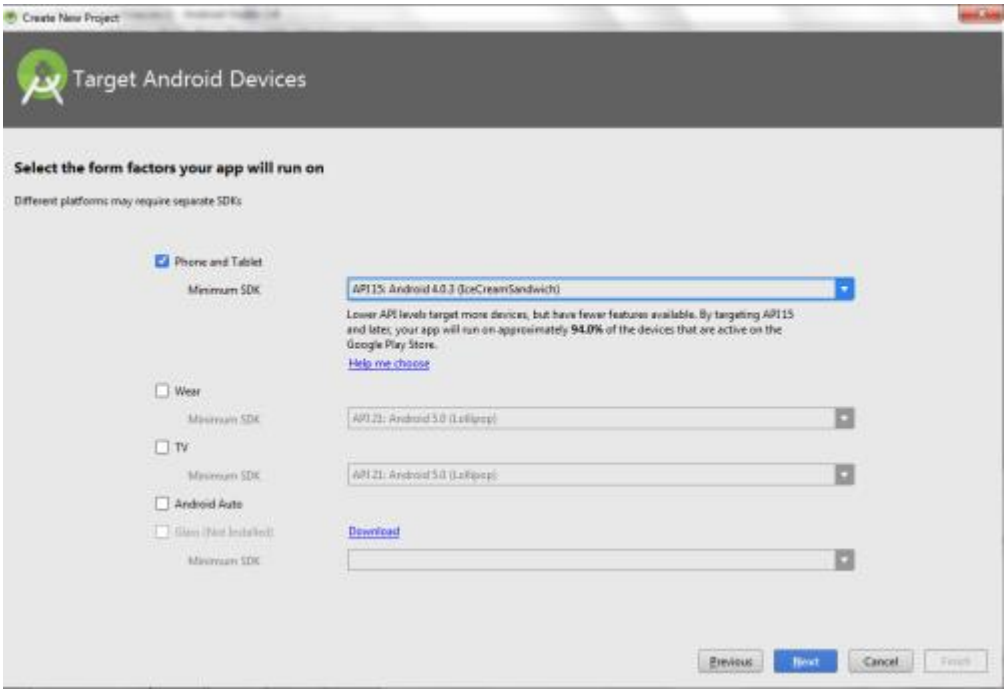
- Open Android Studio and then click on **File -> New -> New project.**



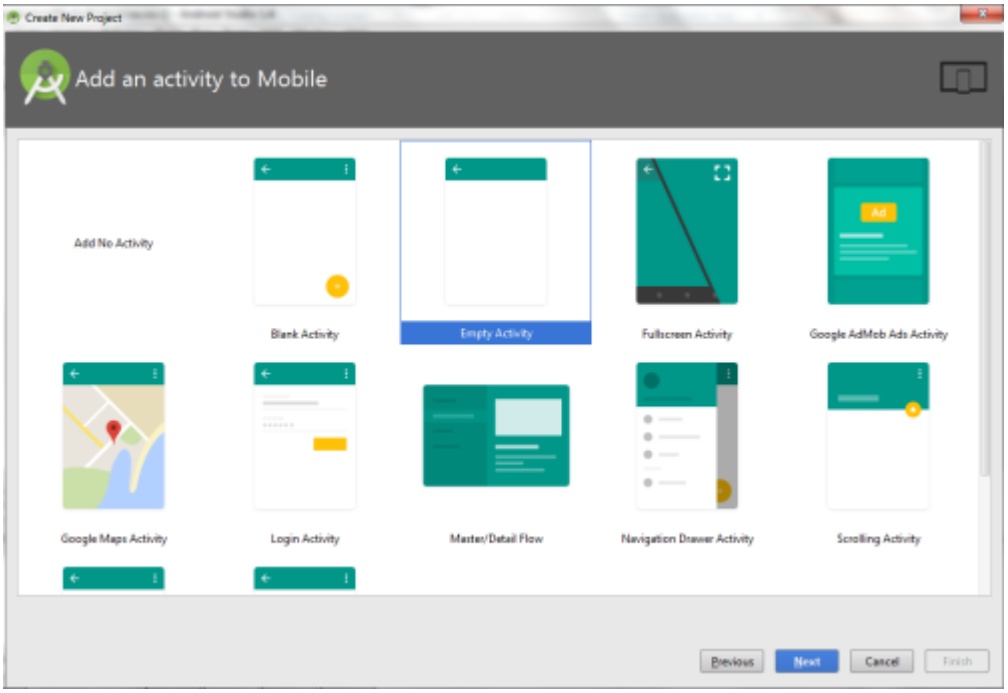
- Then type the Application name as "**ex.no.7"** and click **Next.**



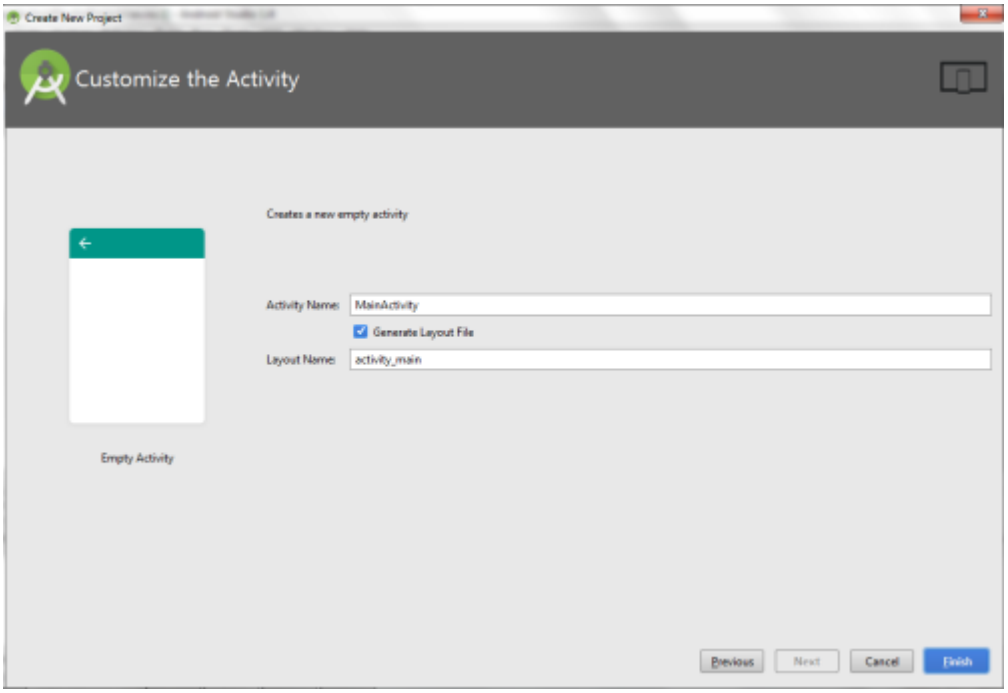
- Then select the **Minimum SDK** as shown below and click **Next**.



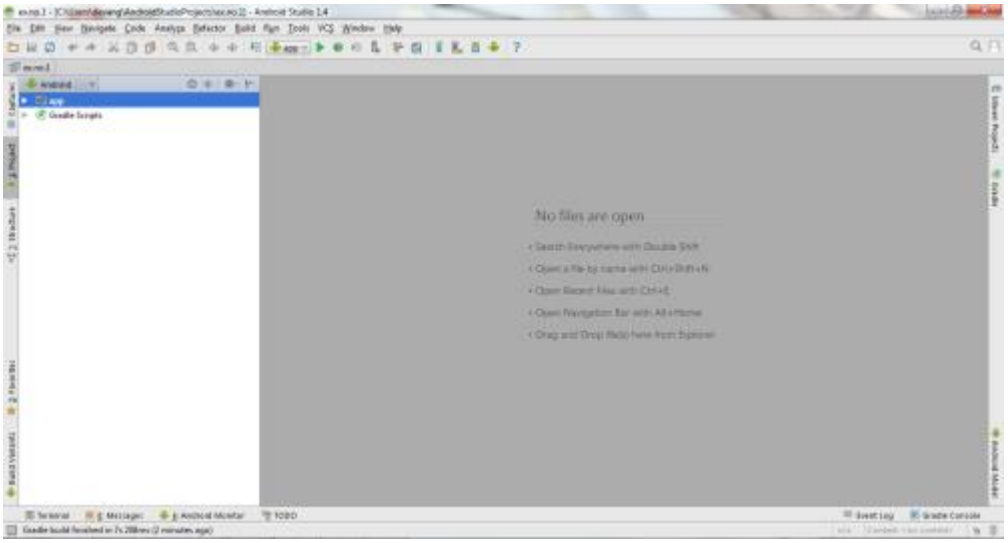
- Then select the **Empty Activity** and click **Next.**



- Finally click F**inish**.

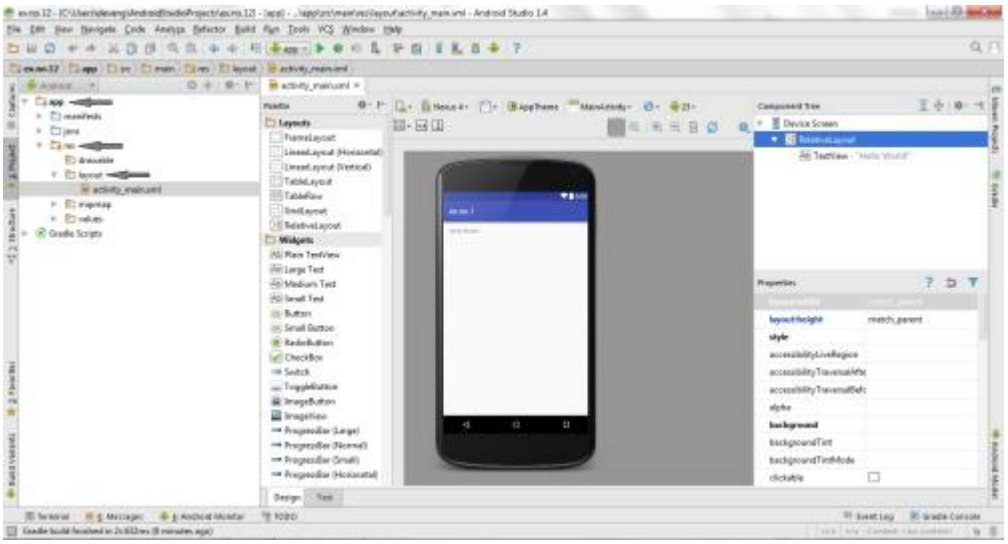


- It will take some time to build and load the project.
- After completion it will look as given below.

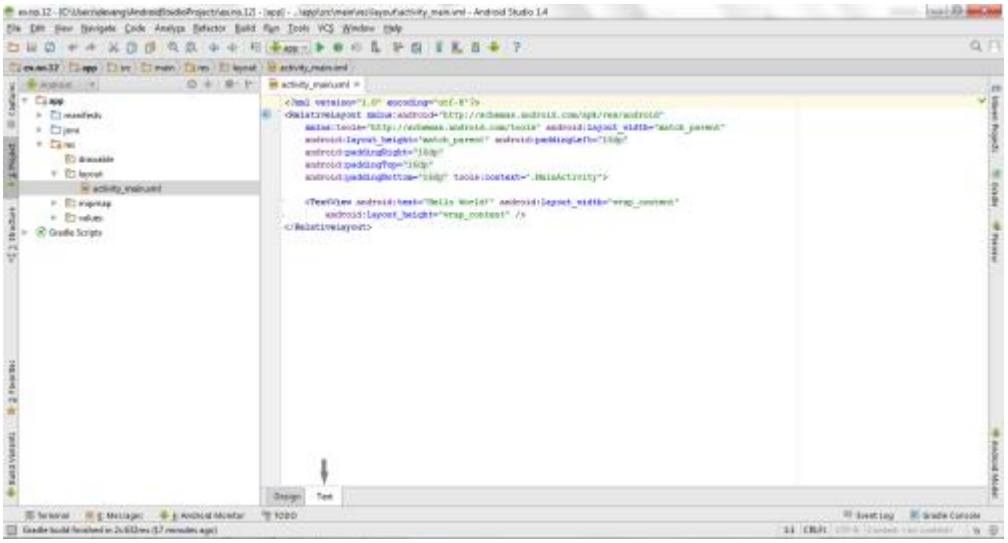


Designing layout for the Android Application:

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



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

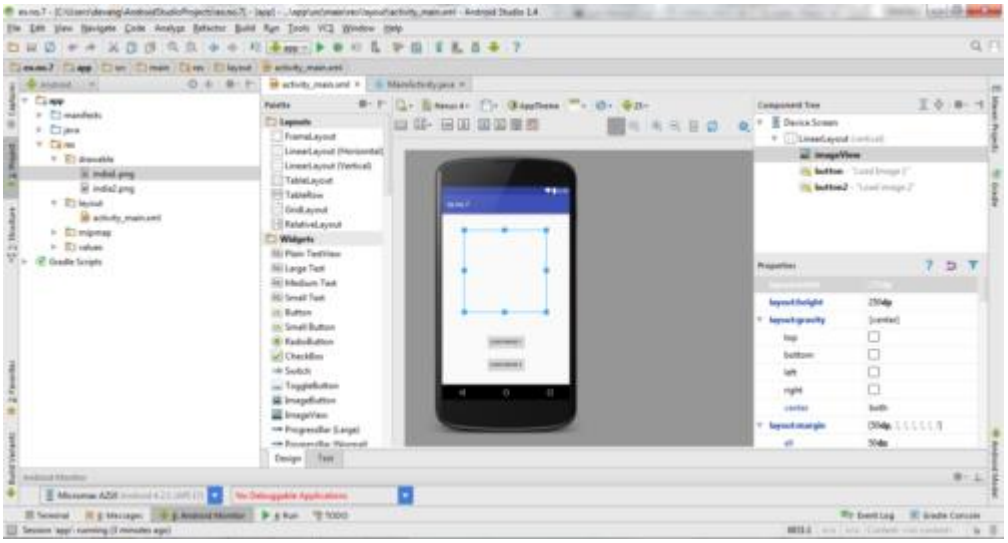


- Then delete the code which is there and type the code as given below.

Code for Activity_main.xml:

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

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="250dp"
        android:layout_height="250dp"
        android:layout_margin="50dp"
        android:layout_gravity="center" />

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:layout_gravity="center"
        android:text="Load Image 1" />

    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:layout_gravity="center"
        android:text="Load image 2" />

</LinearLayout>
```

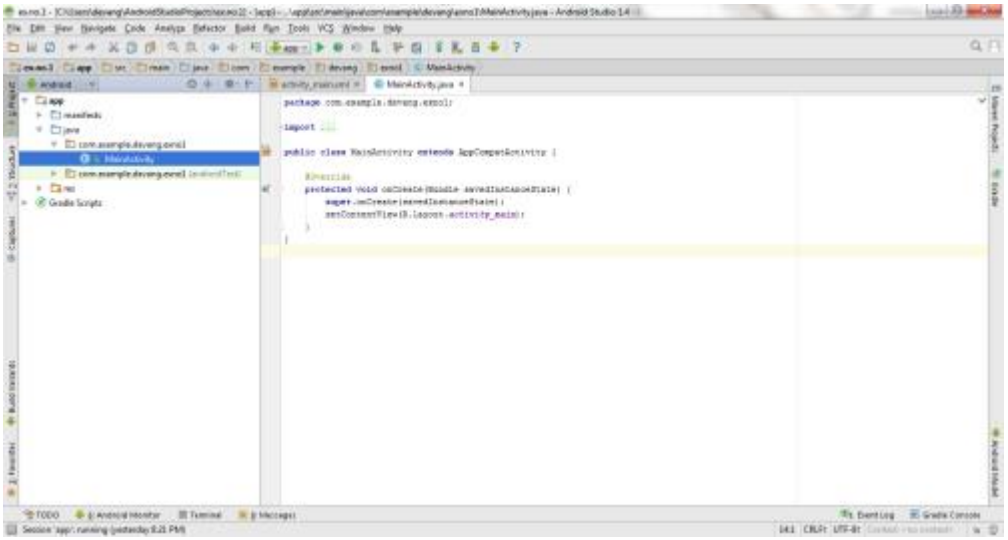
- Now click on **Design** and your application will look as given below.



- So now the designing part is completed.

Java Coding for the Android Application:

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



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

Code for MainActivity.java:

```
package com.example.exno7;

import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
public class MainActivity extends AppCompatActivity
{
    ImageView img;
    Button bt1,bt2;
```

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

        bt1 = (Button)findViewById(R.id.button);
        bt2= (Button) findViewById(R.id.button2);
        img = (ImageView)findViewById(R.id.imageView);

        bt1.setOnClickListener(new View.OnClickListener()
        {
            @Override
            public void onClick(View v)
            {
                new Thread(new Runnable()
                {
                    @Override
                    public void run()
                    {
                        img.post(new Runnable()
                        {
                            @Override
                            public void run()
                            {
                                img.setImageResource(R.drawable.india1);
                            }
                        });
                    }
                }).start();
            }
        });

        bt2.setOnClickListener(new View.OnClickListener()
        {
            @Override
            public void onClick(View v)
            {
                new Thread(new Runnable()
                {
                    @Override
                    public void run()
                    {
                        img.post(new Runnable()
                        {
                            @Override
                            public void run()
                            {
                                img.setImageResource(R.drawable.india2);
                            }
                        });
                    }
                }).start();
            }
        });
    }
}
```

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

Note: Before Running the Application, Copy the Images given below and Paste it in *"app -> res -> drawable"* by pressing "right click mouse button on *drawable*" and selecting the "*Paste*" option.

Output:

Result:

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

EXERCISE 6.1:

Execute the above exercise of multi threading in Android studio and show the exact GUI output? [04]

EXERCISE 6.2:

Develop an android application to load 10 images at a time in the activity. [06]

RESOURCES: https://www.codingconnect.net/mobile-application-development-lab/

https://www.javatpoint.com/android-tutorial
https://www.tutorialspoint.com/android
https://developer.android.com/guide
Https://developer.Android.com/training/basics/firstapp/creating-project
Https://www.raywenderlich.com/120177/beginning-Android-development-tutorial-installingAndroid-studio
https://www.youtube.com/playlist?list=PLS1QulWo1RIbb1cYyzZpLFCKvdYV_yJ-E
Https://www.udemy.com/java-tutorial/
Https://www.w3schools.com/xml/
Http://www.Androidguys.com/

EXERCISE 8 – ANDROID APPLICATION THAT WRITES DATA TO THE SD CARD

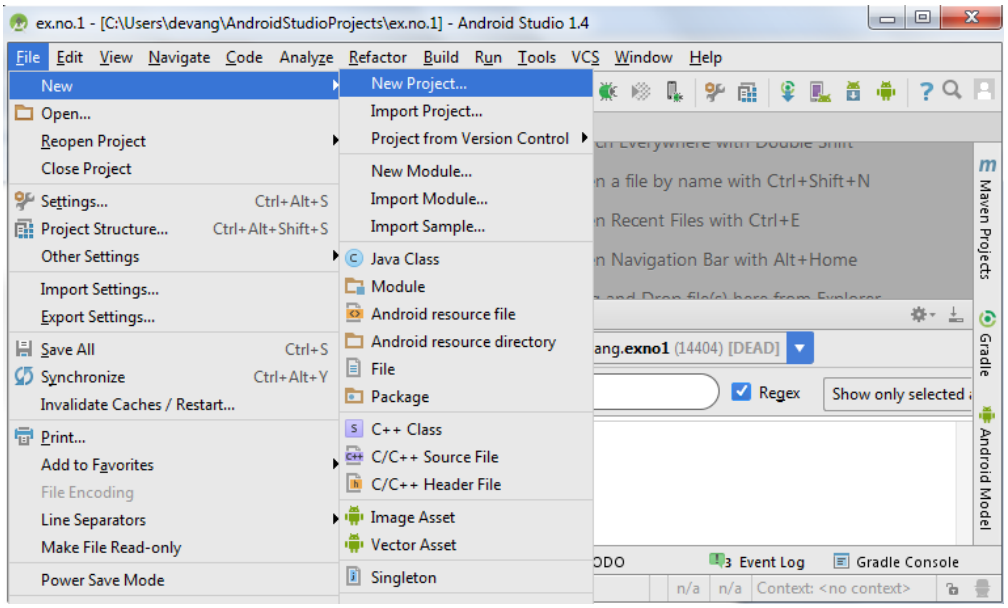
Aim:

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

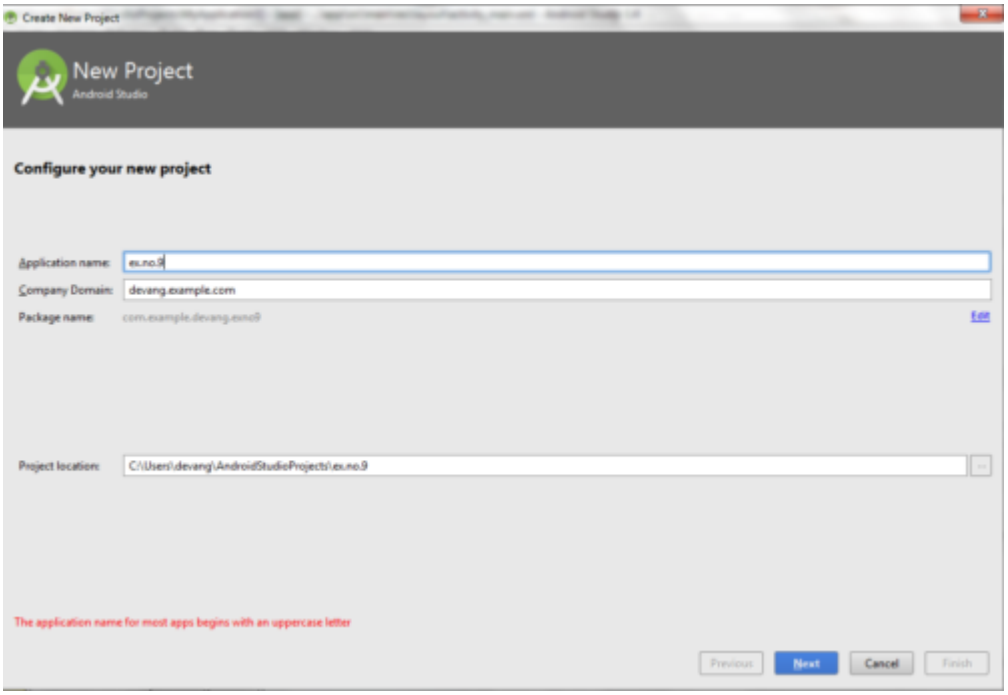
Procedure:

Creating a New project:

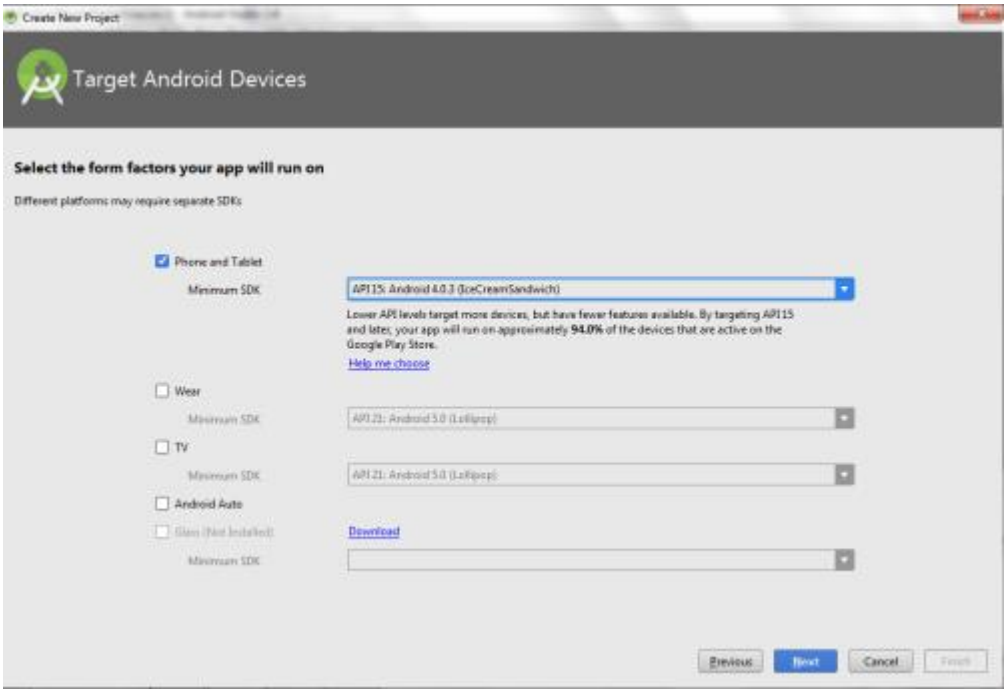
- Open Android Studio and then click on **File -> New -> New project.**



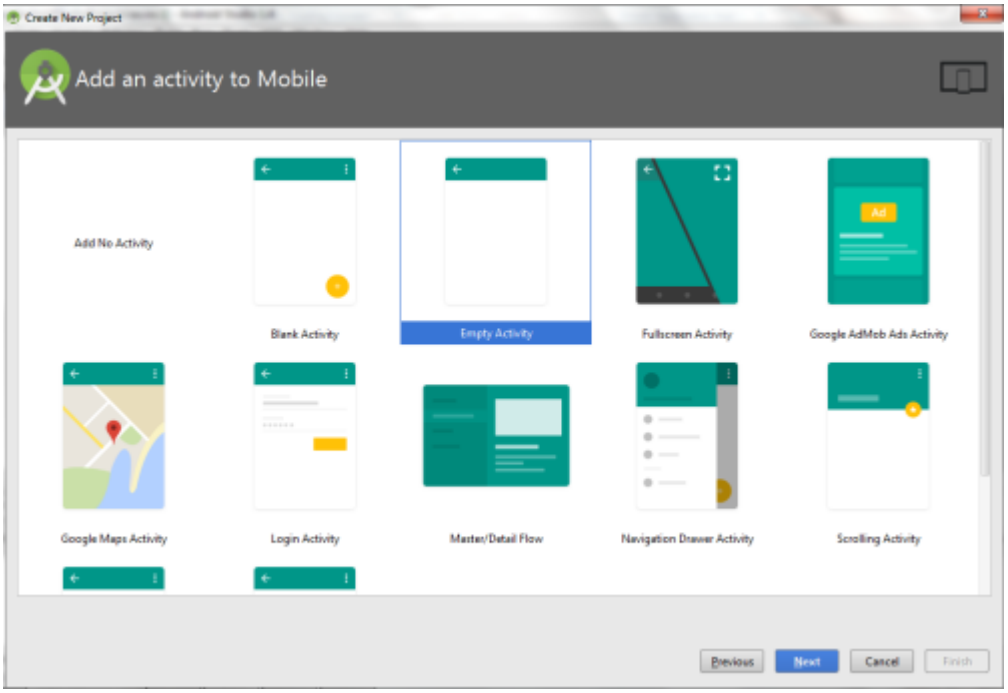
- Then type the Application name as "**ex.no.9"** and click **Next.**



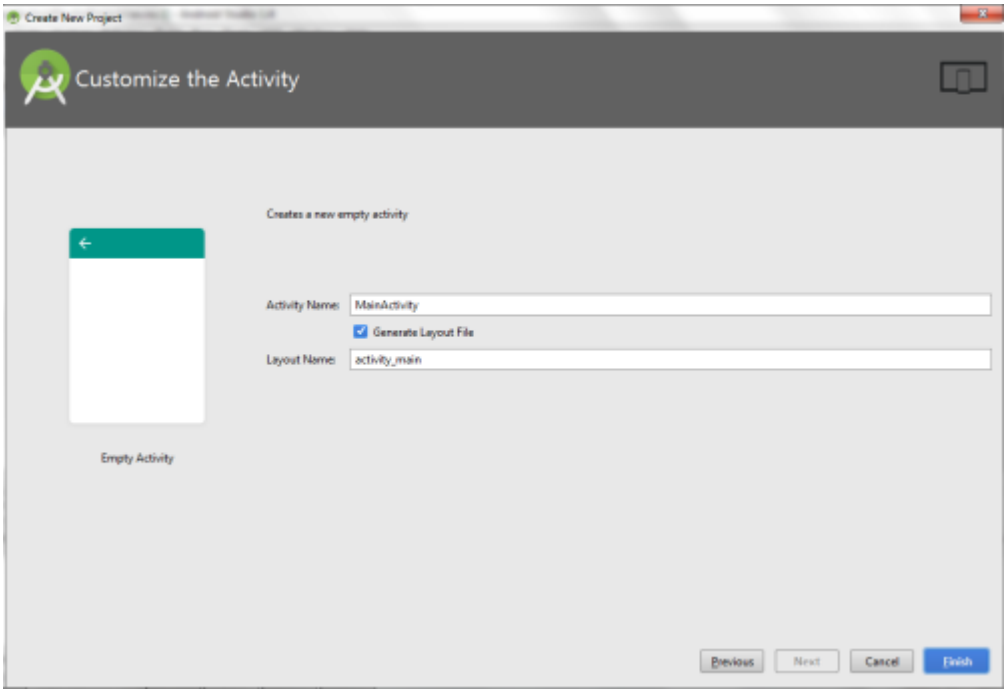
- Then select the **Minimum SDK** as shown below and click **Next**.



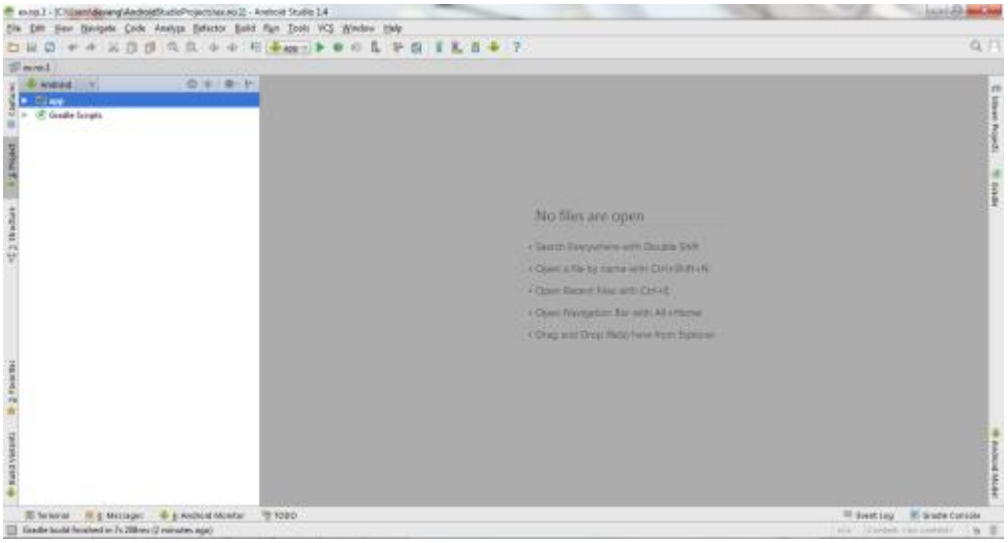
- Then select the **Empty Activity** and click **Next.**



- Finally click F**inish**.

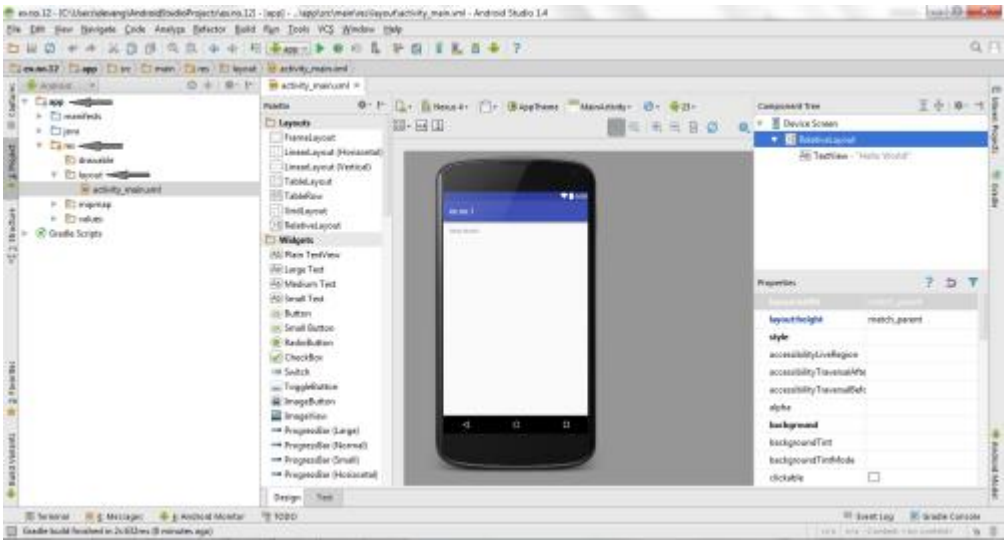


- It will take some time to build and load the project.
- After completion it will look as given below.

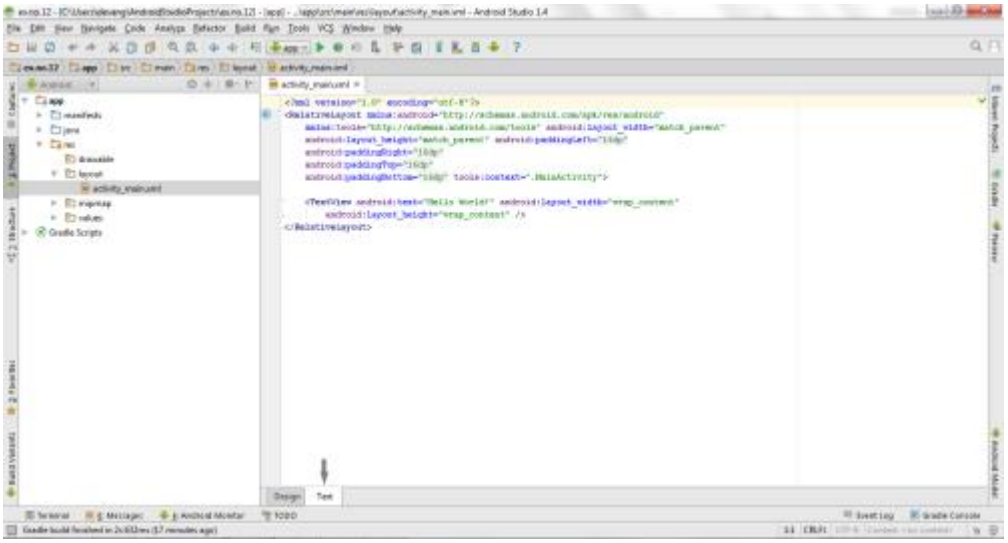


Designing layout for the Android Application:

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



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



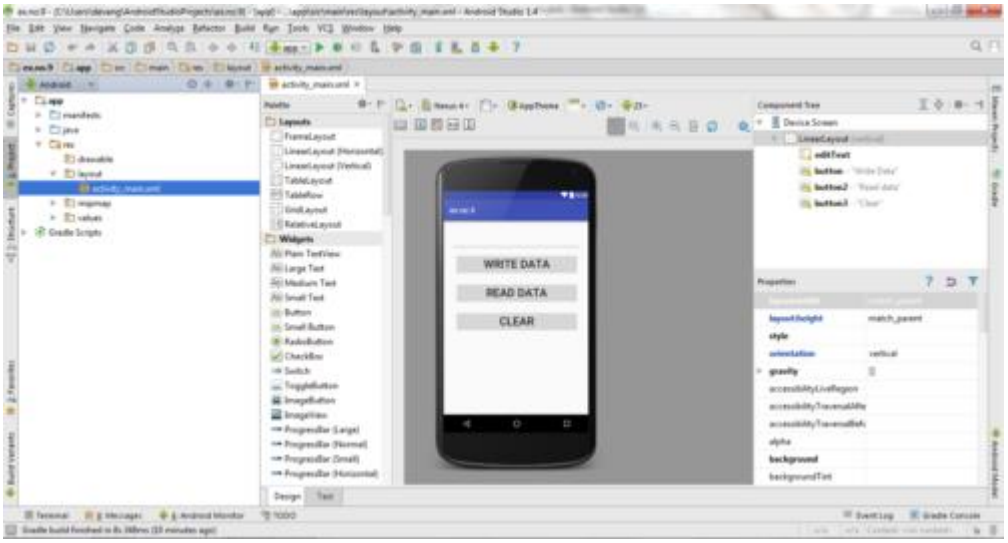
- Then delete the code which is there and type the code as given below.

Code for Activity_main.xml:

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

    <EditText
        android:id="@+id/editText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:singleLine="true"
        android:textSize="30dp" />

    <Button
        android:id="@+id/button"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:text="Write Data"
        android:textSize="30dp" />

    <Button
        android:id="@+id/button2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:text="Read data"
        android:textSize="30dp" />

    <Button
        android:id="@+id/button3"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
```

```
        android:text="Clear"
        android:textSize="30dp" />

</LinearLayout>
```

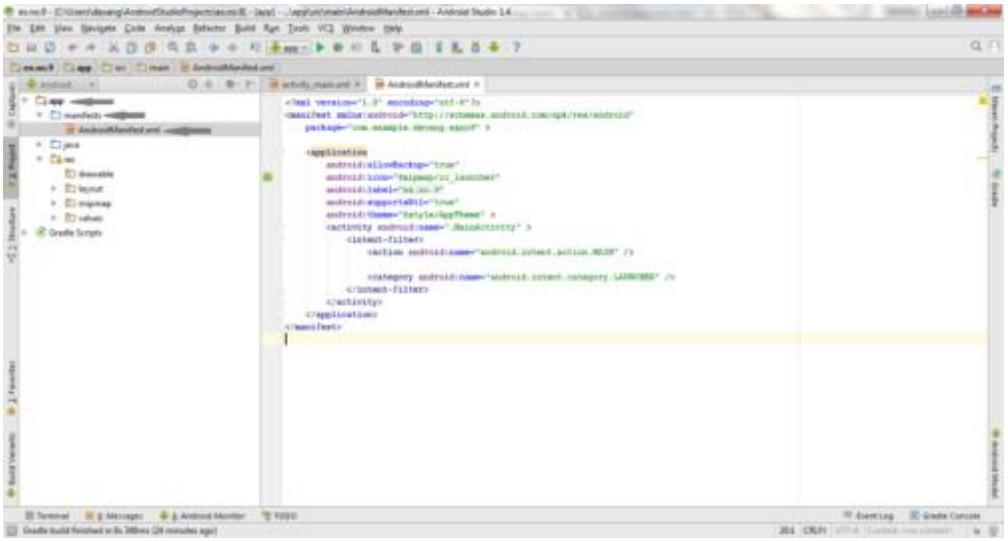
- Now click on **Design** and your application will look as given below.



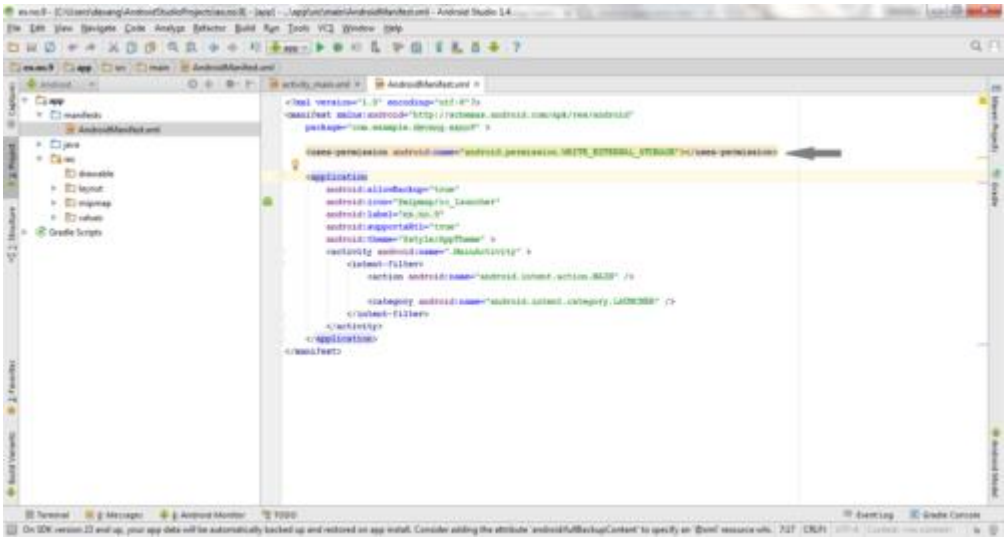
- So now the designing part is completed.

Adding permissions in Manifest for the Android Application:

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



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



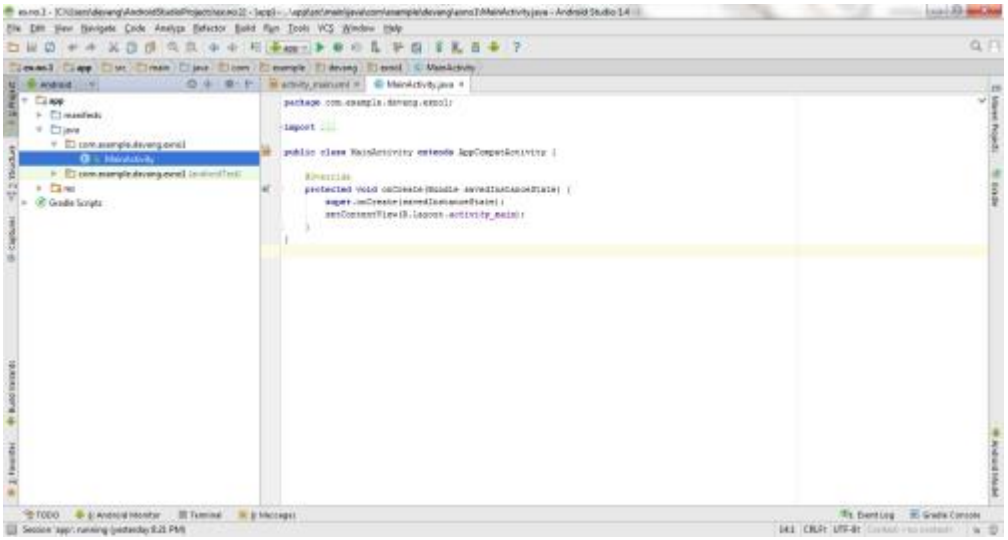
Code for AndroidManifest.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.exno9" >

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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/AppTheme" >
        <activity android:name=".MainActivity" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

- So now the Permissions are added in the Manifest.

Java Coding for the Android Application:

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



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

Code for MainActivity.java:

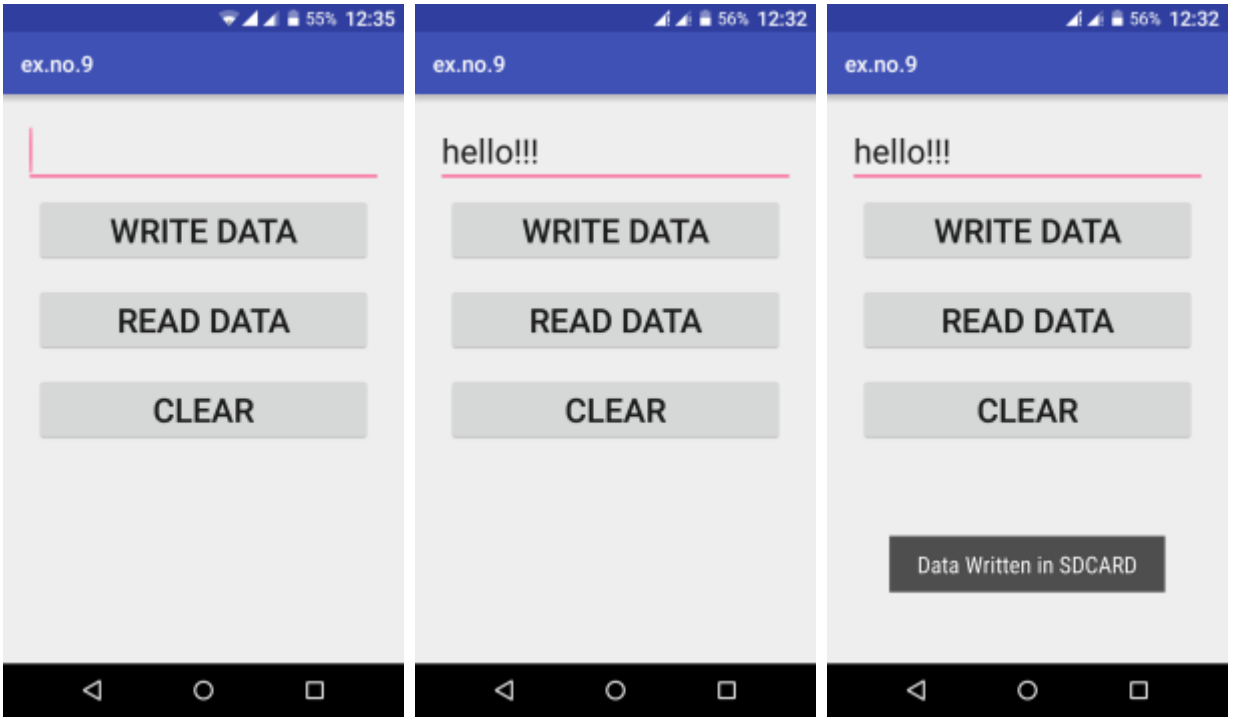
```
package com.example.exno9;

import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.InputStreamReader;

public class MainActivity extends AppCompatActivity
{
    EditText e1;
    Button write,read,clear;
    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        e1= (EditText) findViewById(R.id.editText);
        write= (Button) findViewById(R.id.button);
        read= (Button) findViewById(R.id.button2);
        clear= (Button) findViewById(R.id.button3);

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

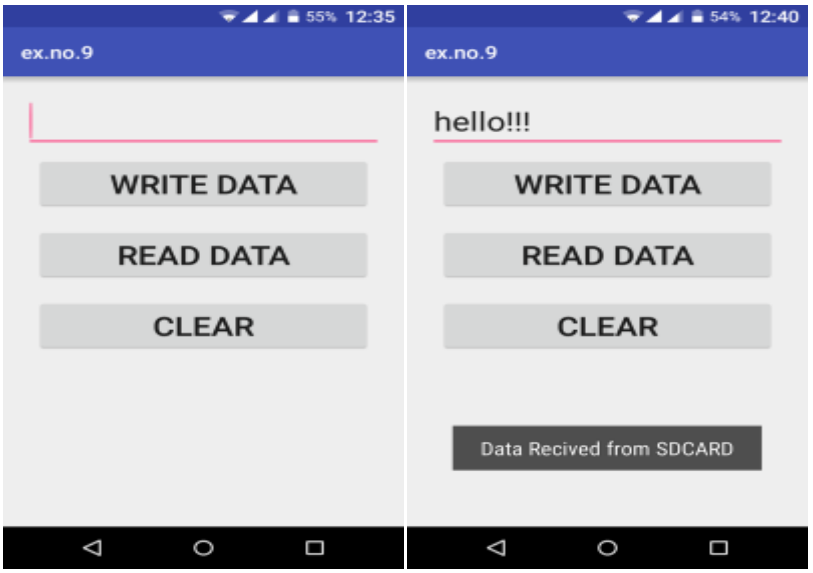
```
                String message=e1.getText().toString();
                try
                {
                    File f=new File("/sdcard/myfile.txt");
                    f.createNewFile();
                    FileOutputStream fout=new FileOutputStream(f);
                    fout.write(message.getBytes());
                    fout.close();
                    Toast.makeText(getBaseContext(),"Data Written in
SDCARD",Toast.LENGTH_LONG).show();
                }
                catch (Exception e)
                {
                    Toast.makeText(getBaseContext(),e.getMessage(),Toast.LENGT
H_LONG).show();
                }
            }
        });

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

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

```
}
```

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

Output:



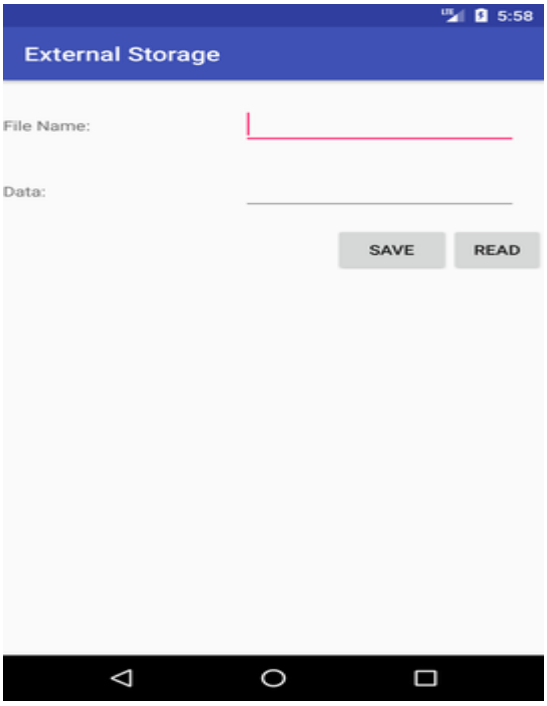


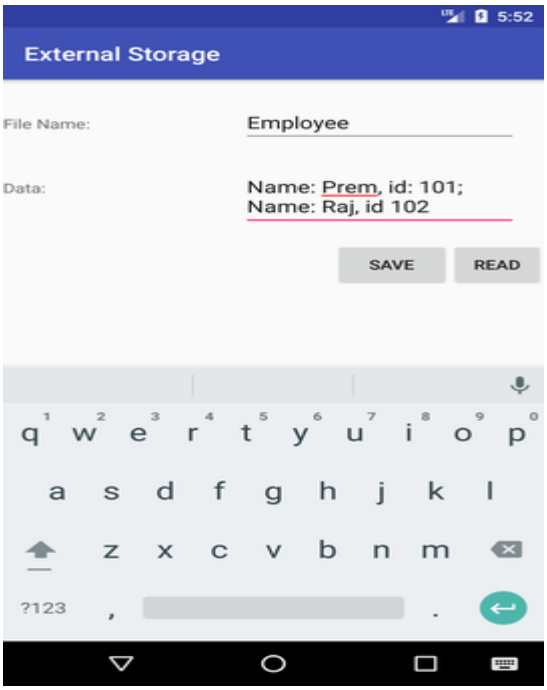
Result:

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

EXERCISE 8.1:

Develop an android application to read and write data on Internal storage. [10]





RESOURCES:

https://www.codingconnect.net/mobile-application-development-lab/
https://www.javatpoint.com/android-tutorial
https://www.tutorialspoint.com/android
https://developer.android.com/guide
Https://developer.Android.com/training/basics/firstapp/creating-project
Https://www.raywenderlich.com/120177/beginning-Android-development-tutorial-installingAndroid-studio
https://www.youtube.com/playlist?list=PLS1QulWo1RIbb1cYyzZpLFCKvdYV_yJ-E
Https://www.udemy.com/java-tutorial/
Https://www.w3schools.com/xml/
Http://www.Androidguys.com/

EXERCISE 9 – ANDROID APPLICATION THAT CREATES AN ALERT UPON RECEIVING A MESSAGE

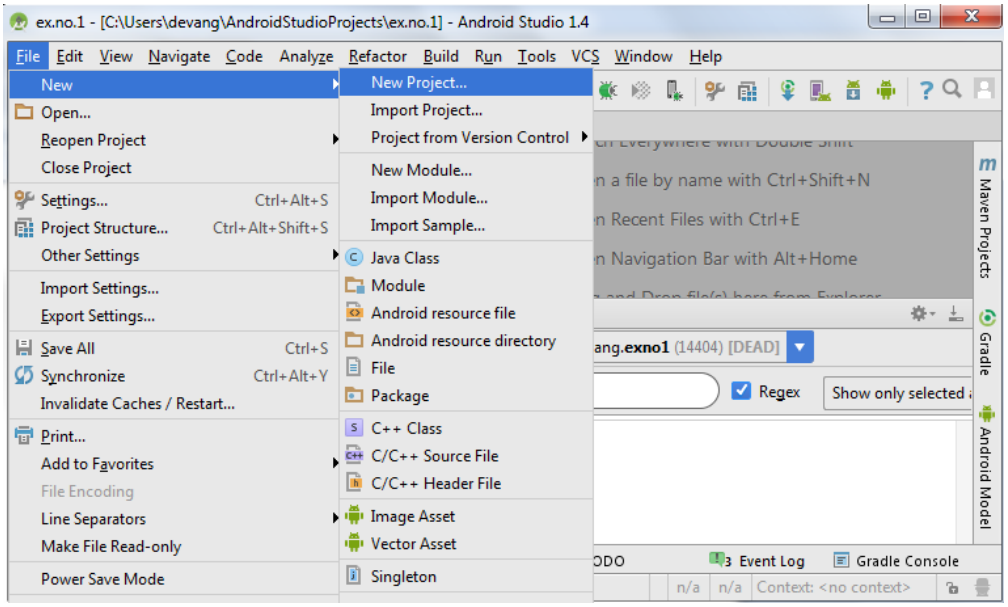
Aim:

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

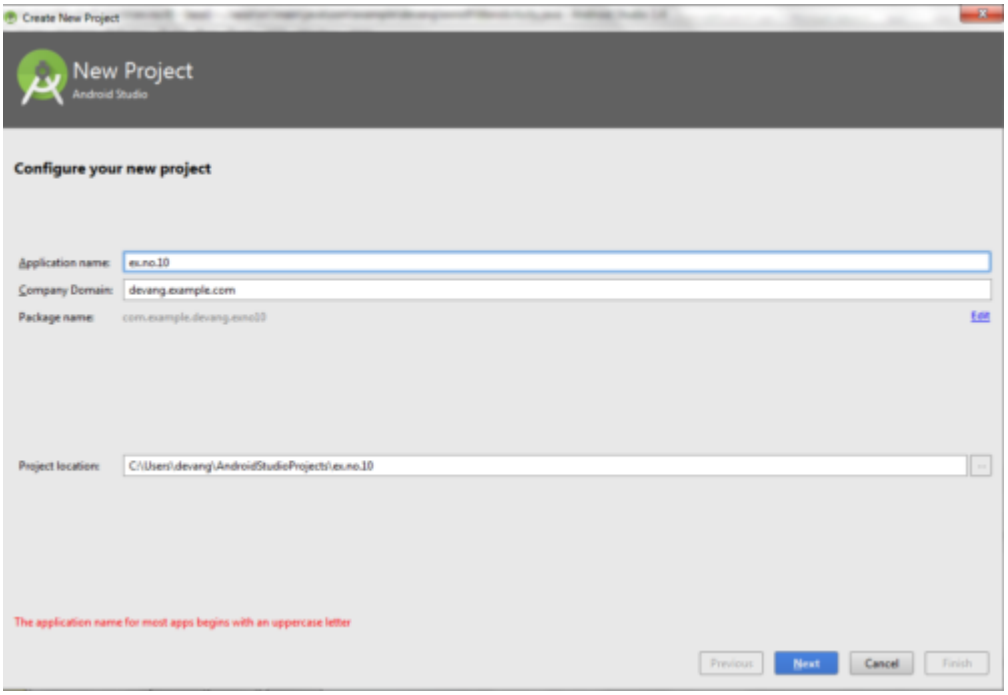
Procedure:

Creating a New project:

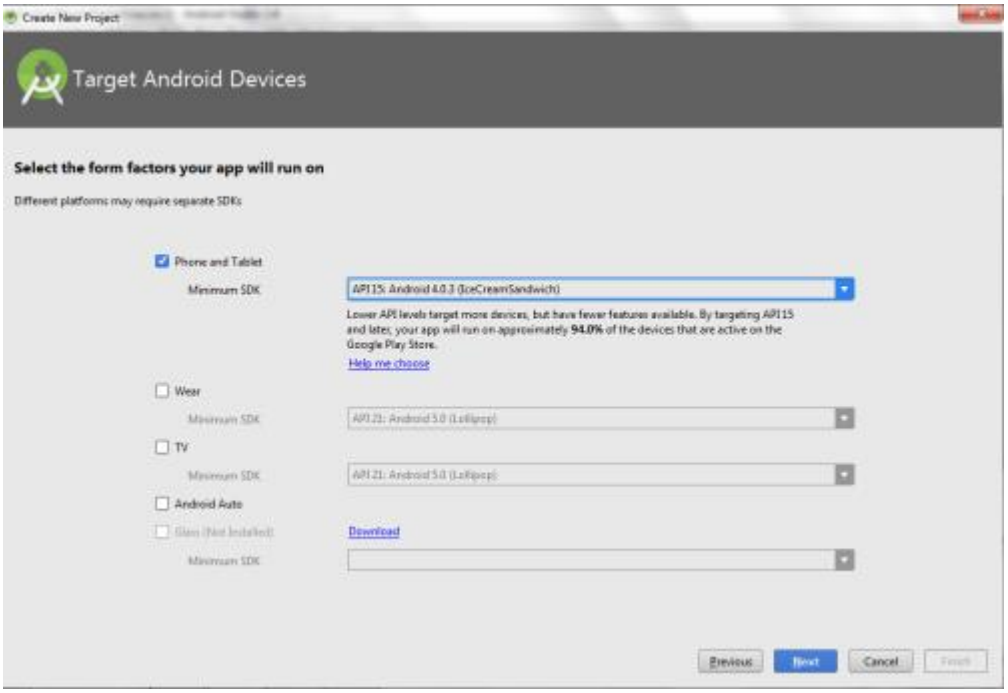
- Open Android Studio and then click on **File -> New -> New project.**



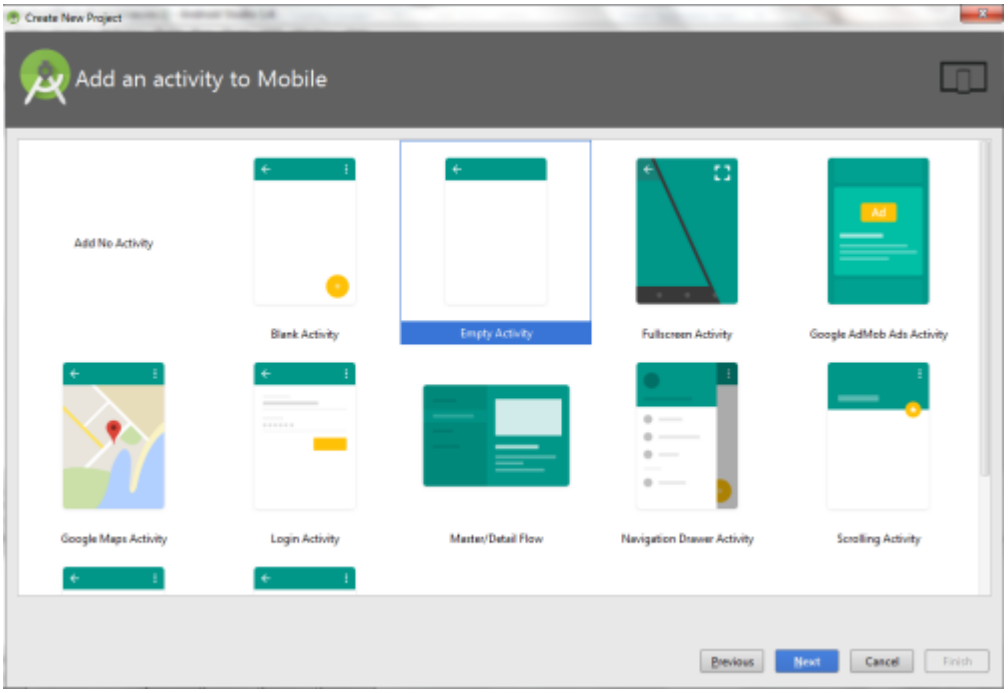
- Then type the Application name as "**ex.no.10"** and click **Next.**



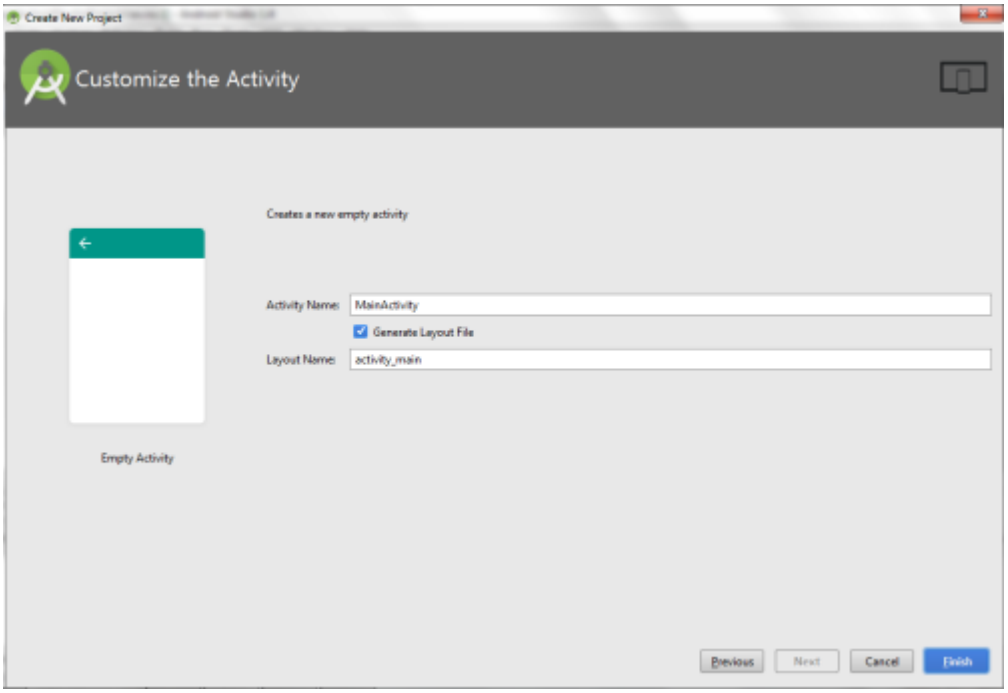
- Then select the **Minimum SDK** as shown below and click **Next**.



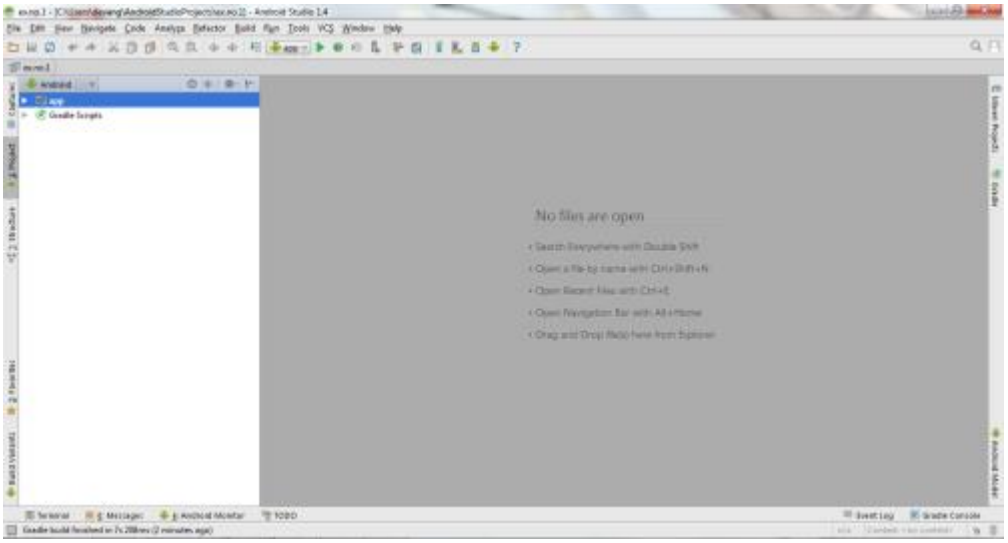
- Then select the **Empty Activity** and click **Next.**



- Finally click F**inish**.

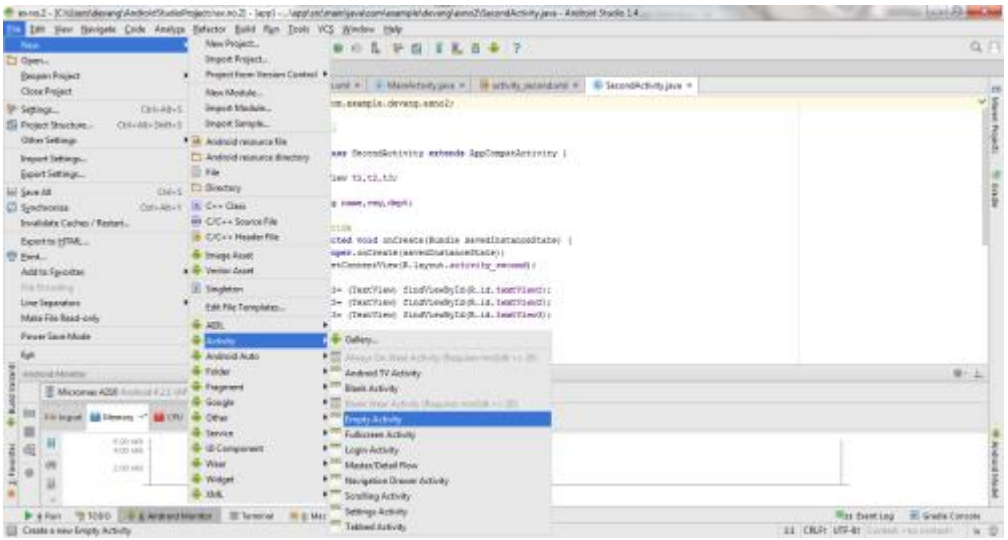


- It will take some time to build and load the project.
- After completion it will look as given below.

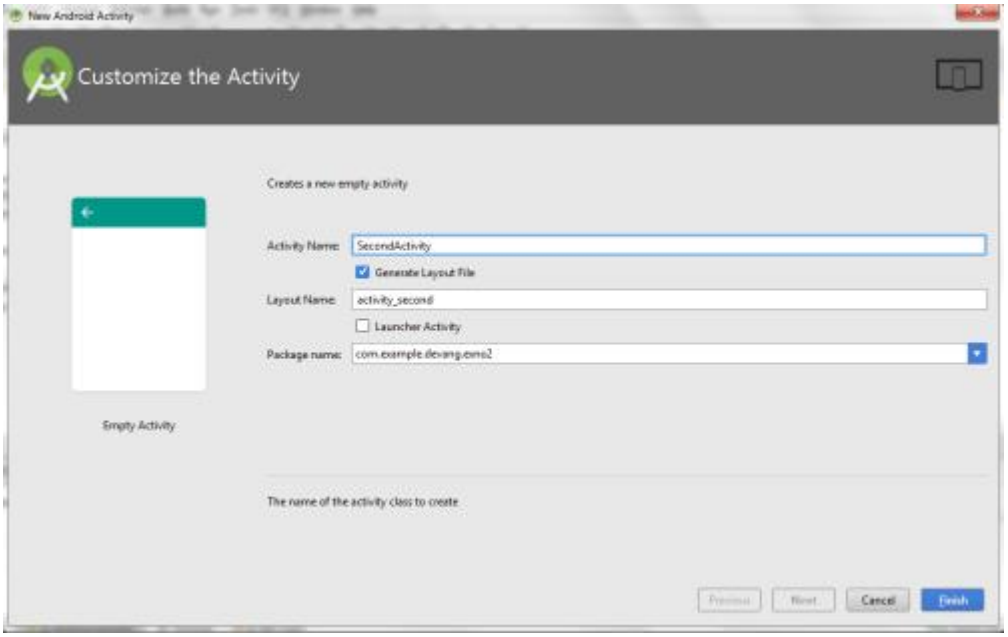


Creating Second Activity for the Android Application:

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



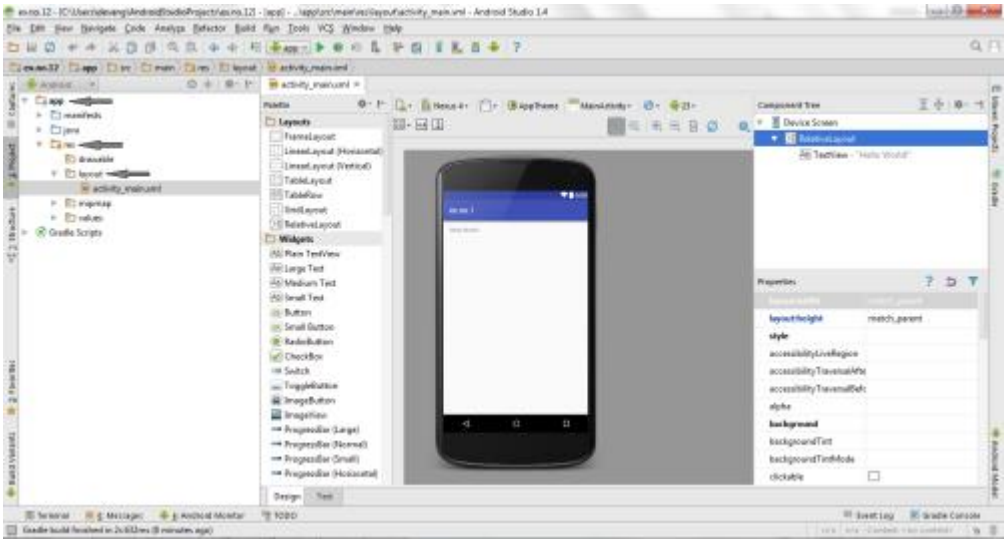
- Type the Activity Name as **SecondActivity** and click **Finish** button.



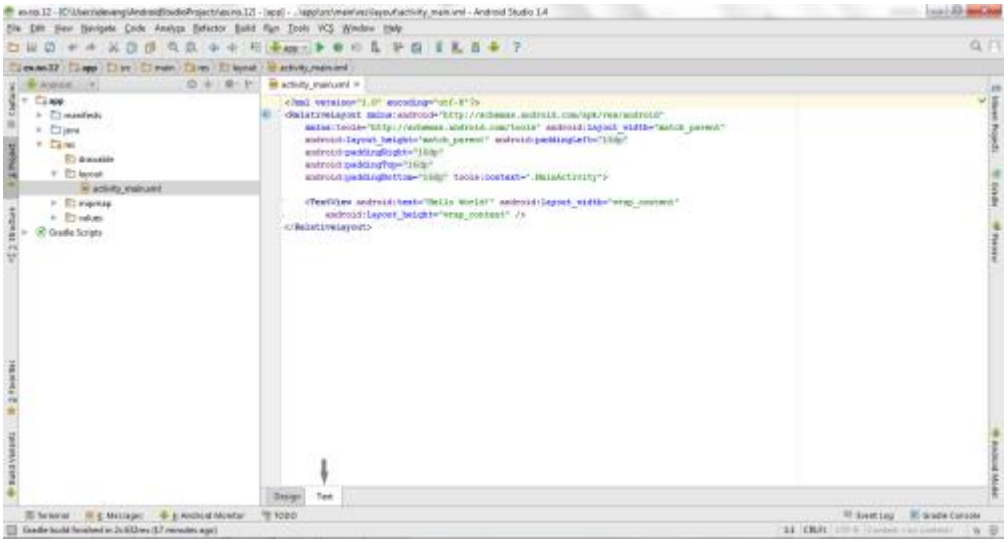
- Thus Second Activity For the application is created.

Designing layout for the Android Application:

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



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

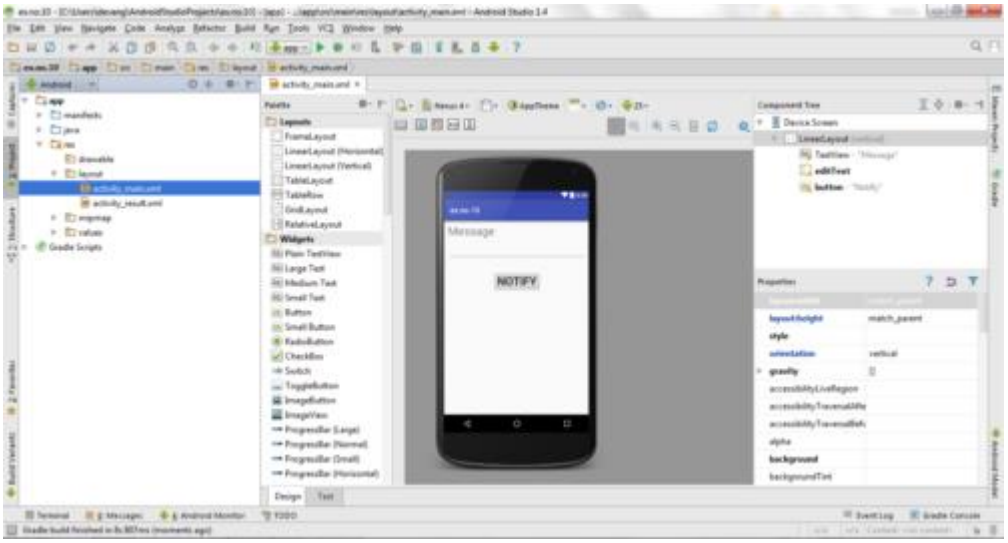


- Then delete the code which is there and type the code as given below.

Code for Activity_main.xml:

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

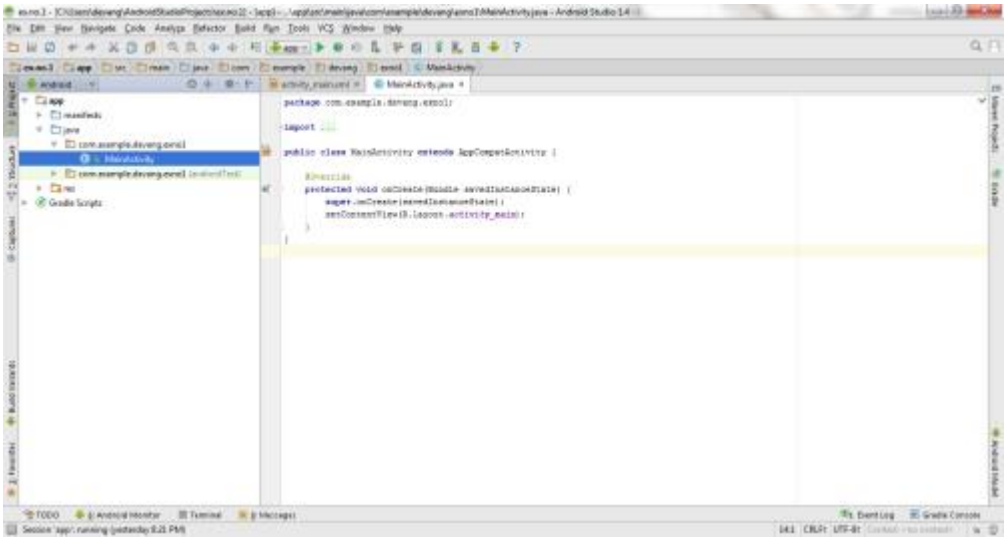
- Now click on **Design** and your application will look as given below.



- So now the designing part is completed.

Java Coding for the Android Application:

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



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

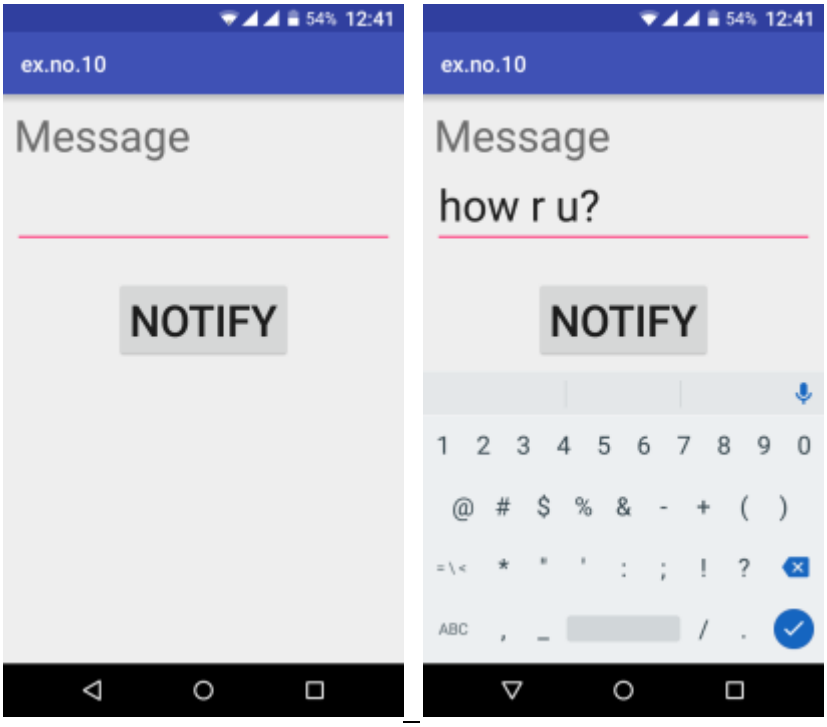
Code for MainActivity.java:

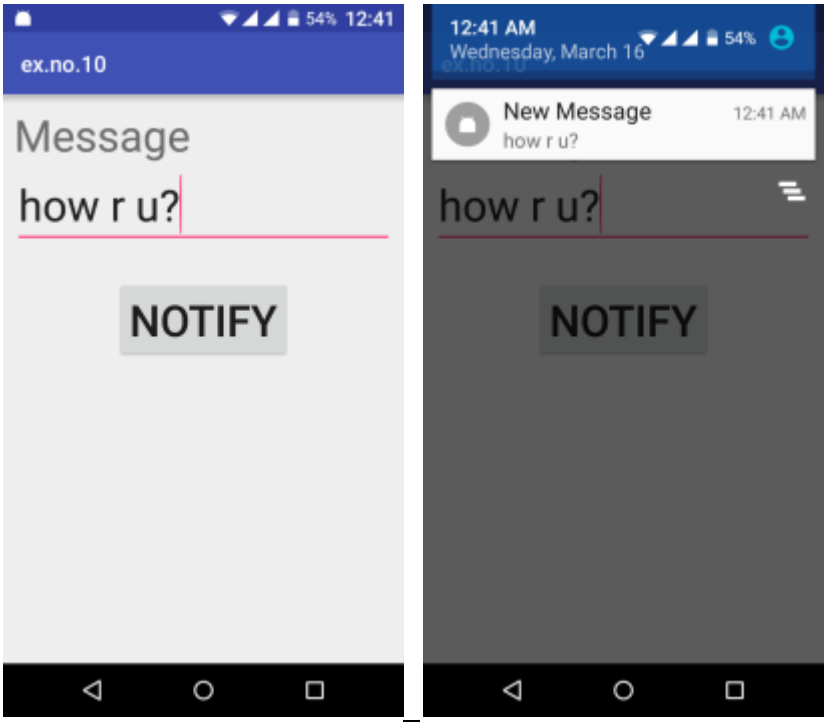
```
package com.example.exno10;

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

```
import android.widget.EditText;

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

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

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

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

Output:





Result:

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

EXERCISE 9.1:

Develop an android application to send an SMS to multiple recipients. [05]

TASK 9.1

Develop an android application to send an email to multiple recipients.

EXERCISE 9.2:

Develop an android application to generate a phone call. [05]

RESOURCES:

https://www.codingconnect.net/mobile-application-development-lab/
https://www.javatpoint.com/android-tutorial
https://www.tutorialspoint.com/android
https://developer.android.com/guide
Https://developer.Android.com/training/basics/firstapp/creating-project
Https://www.raywenderlich.com/120177/beginning-Android-development-tutorial-installingAndroid-studio
https://www.youtube.com/playlist?list=PLS1QulWo1RIbb1cYyzZpLFCKvdYV_yJ-E
Https://www.udemy.com/java-tutorial/
Https://www.w3schools.com/xml/
Http://www.Androidguys.com/

EXERCISE 10 – ANDROID APPLICATION THAT CREATES ALARM CLOCK

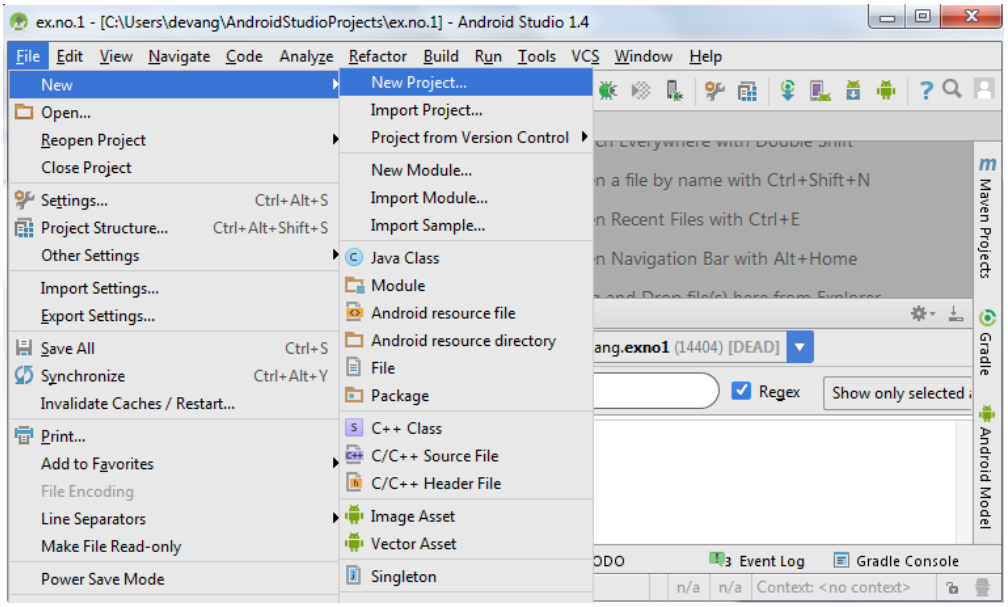
Aim:

To develop a Android Application that creates Alarm Clock.

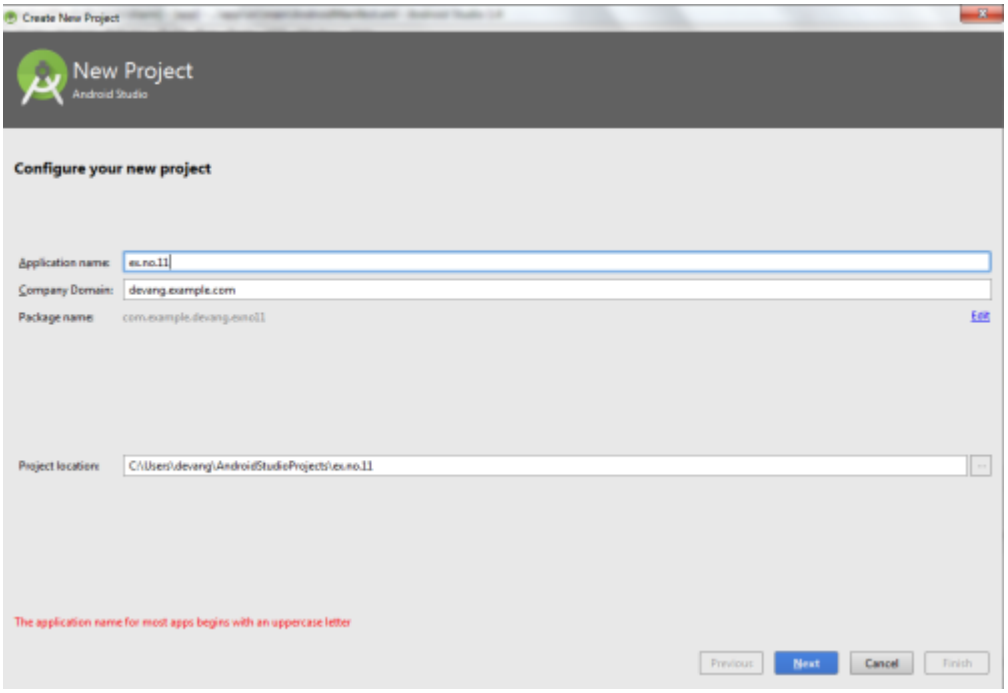
Procedure:

Creating a New project:

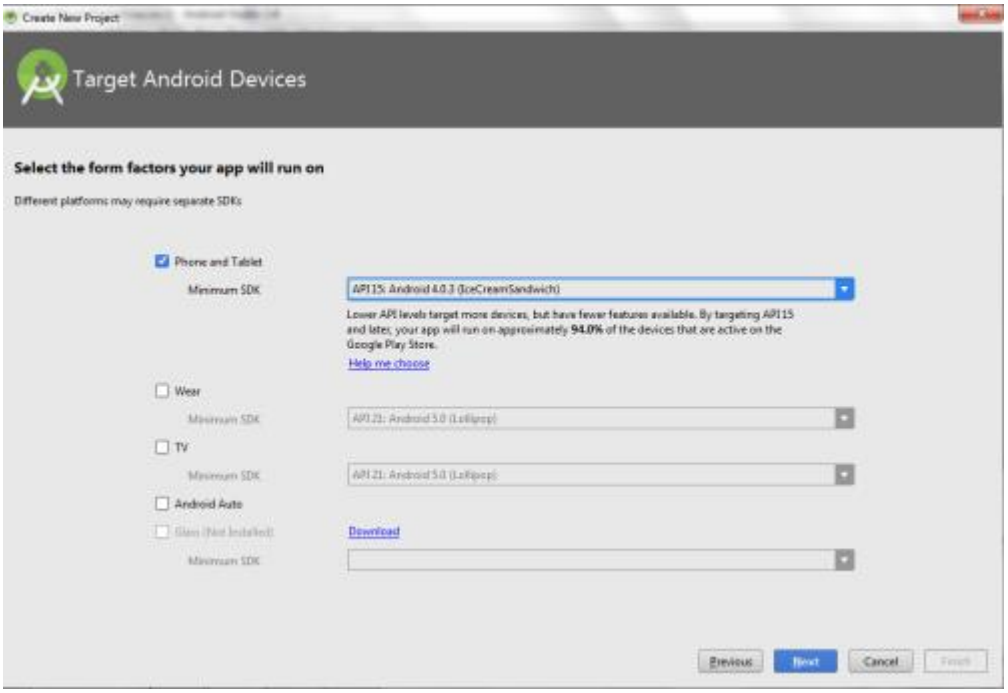
- Open Android Studio and then click on **File -> New -> New project.**



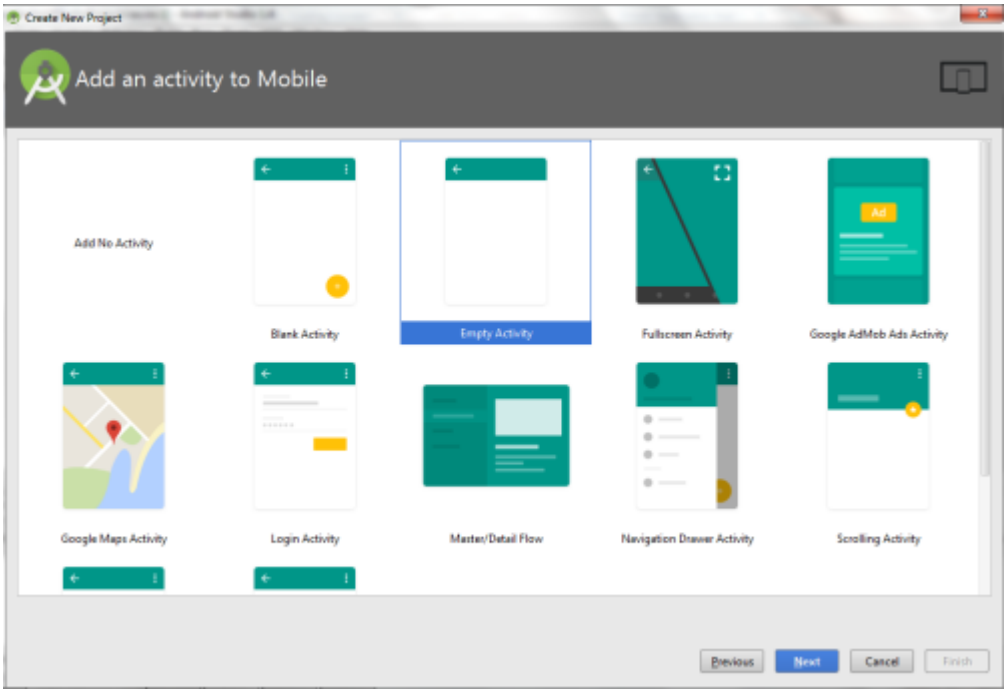
- Then type the Application name as "**ex.no.11"** and click **Next.**



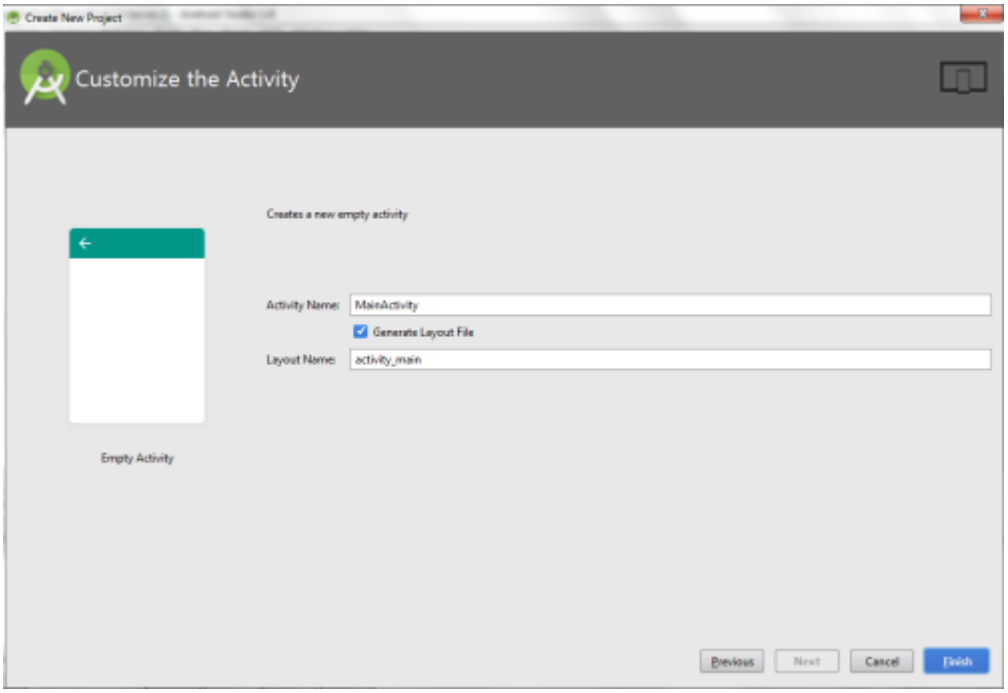
- Then select the **Minimum SDK** as shown below and click **Next**.



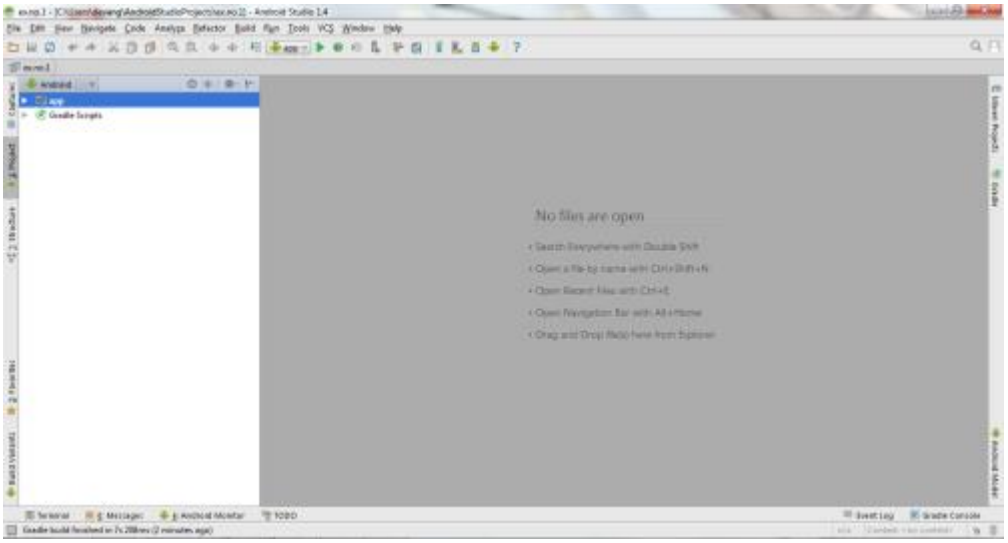
- Then select the **Empty Activity** and click **Next.**



- Finally click F**inish**.

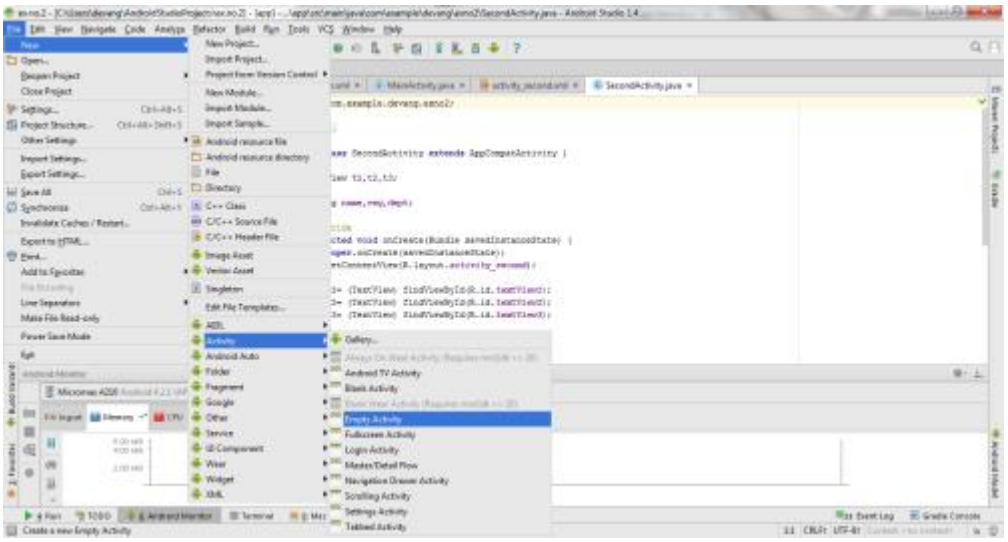


- It will take some time to build and load the project.
- After completion it will look as given below.

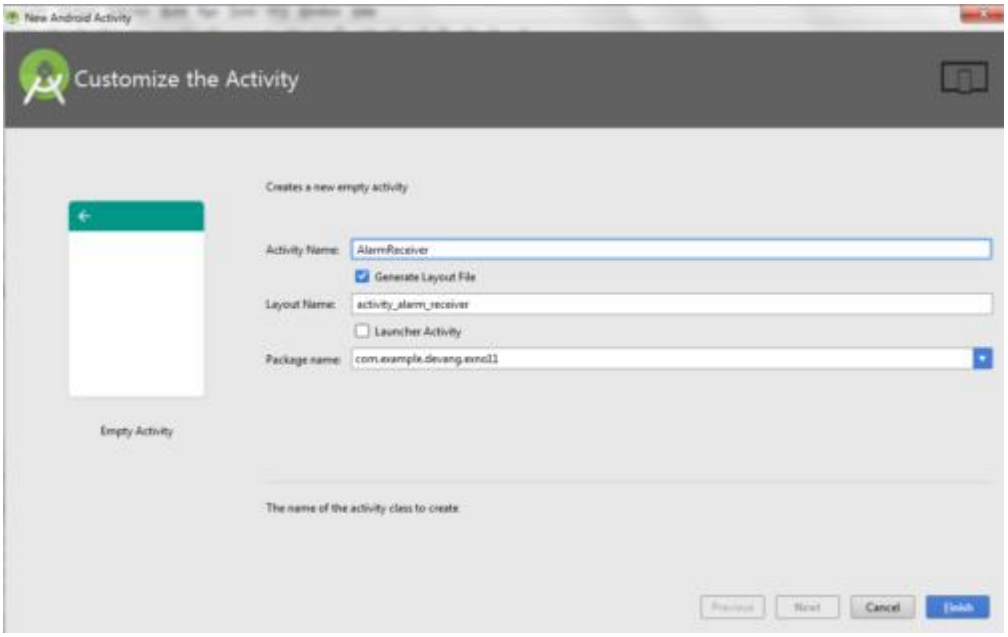


Creating Second Activity for the Android Application:

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



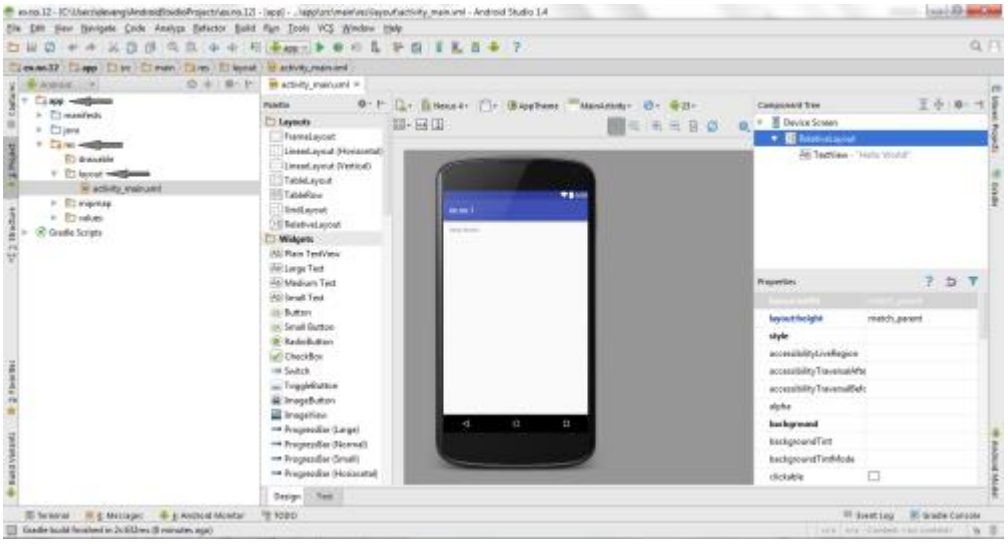
- Type the Activity Name as **AlarmReceiver** and click **Finish** button.



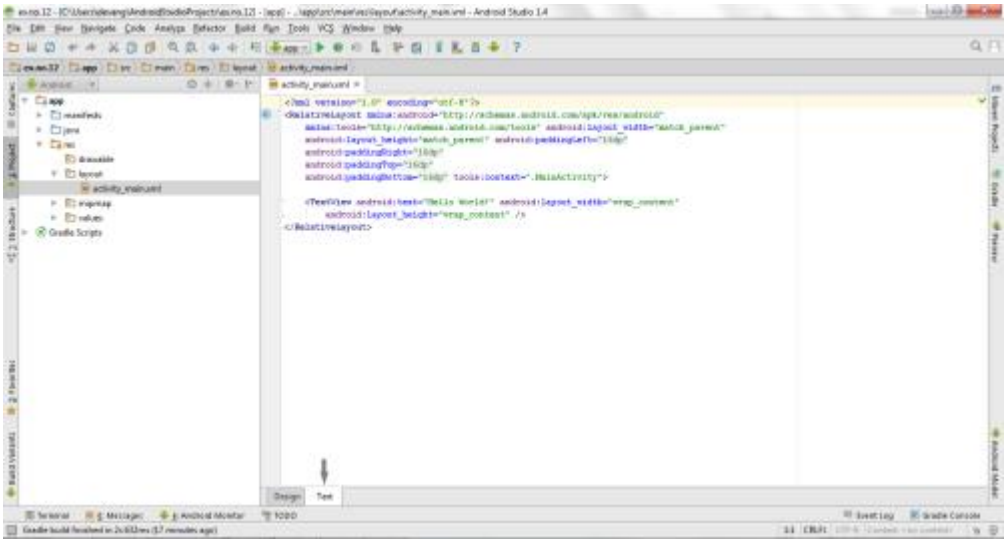
- Thus Second Activity For the application is created.

Designing layout for the Android Application:

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



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

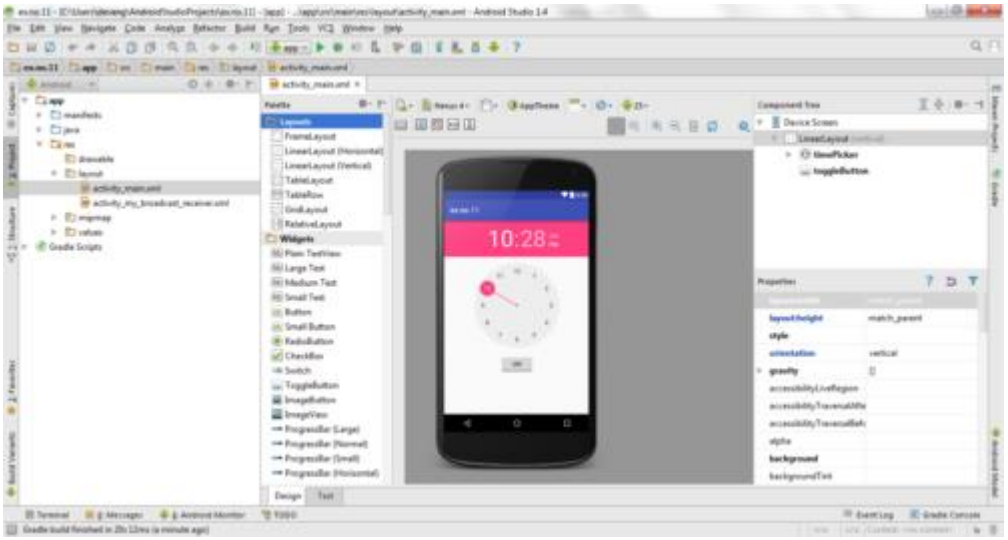


- Then delete the code which is there and type the code as given below.

Code for Activity_main.xml:

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

    <TimePicker
        android:id="@+id/timePicker"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center" />

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

</LinearLayout>
```

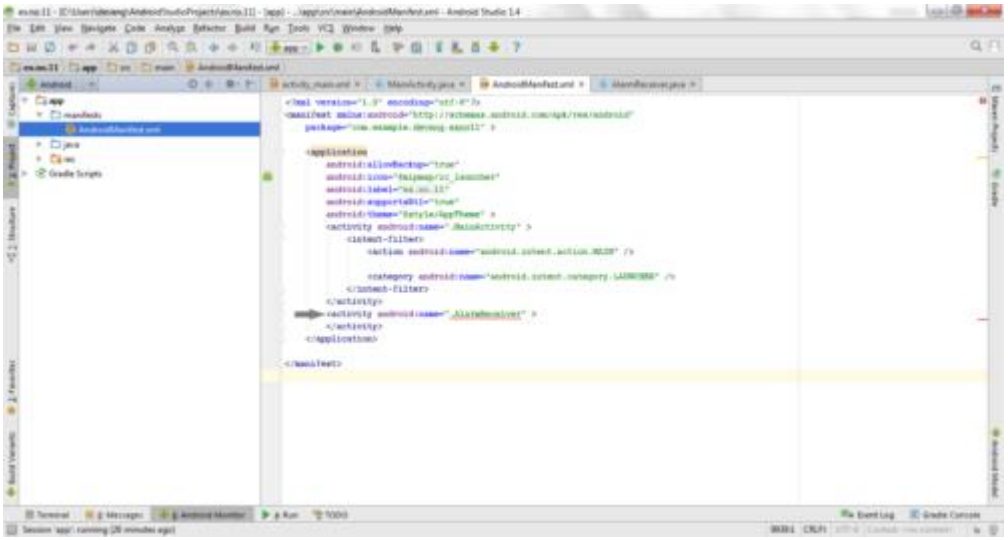
- Now click on **Design** and your application will look as given below.



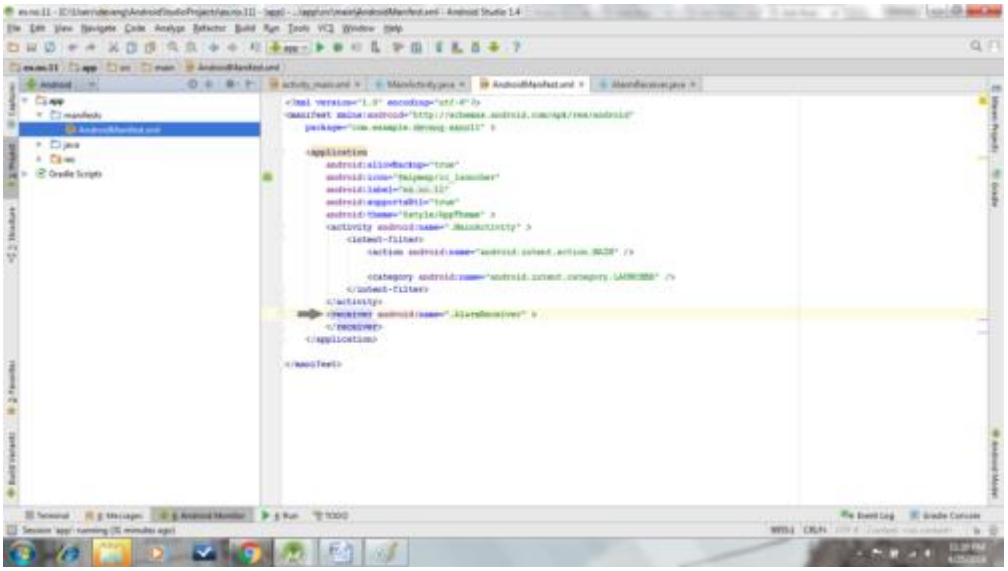
- So now the designing part is completed.

Changes in Manifest for the Android Application:

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



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



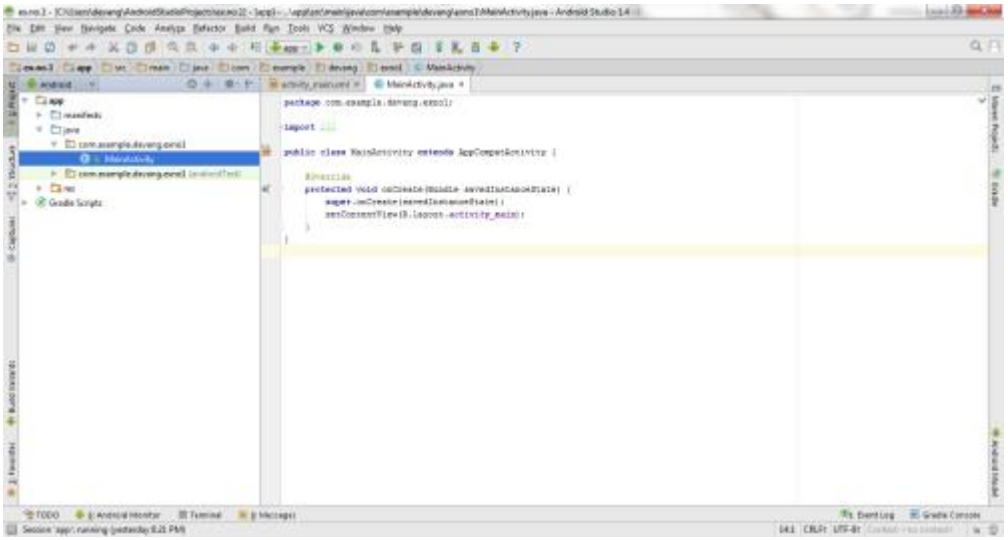
Code for AndroidManifest.xml:

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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/AppTheme" >
        <activity android:name=".MainActivity" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

</manifest>
```

Java Coding for the Android Application:

Java Coding for Main Activity:

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



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

Code for MainActivity.java:

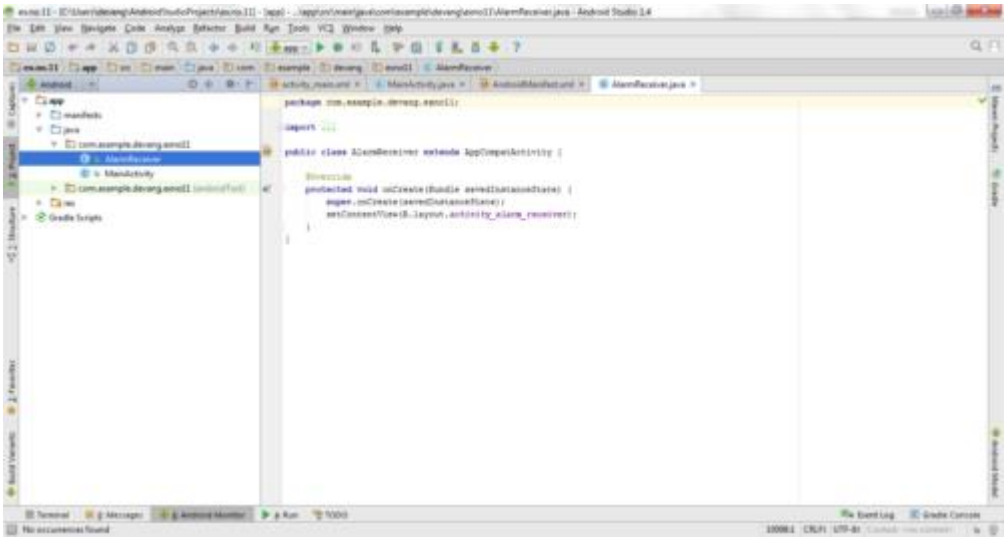
```
package com.example.exno11;

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

import java.util.Calendar;

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

    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        alarmTimePicker = (TimePicker) findViewById(R.id.timePicker);
        alarmManager = (AlarmManager) getSystemService(ALARM_SERVICE);
    }
    public void OnToggleClicked(View view)
    {
        long time;
        if (((ToggleButton) view).isChecked())
        {
            Toast.makeText(MainActivity.this, "ALARM ON",
Toast.LENGTH_SHORT).show();
            Calendar calendar = Calendar.getInstance();
```

```
            calendar.set(Calendar.HOUR_OF_DAY,
alarmTimePicker.getCurrentHour());
            calendar.set(Calendar.MINUTE, alarmTimePicker.getCurrentMinute());
            Intent intent = new Intent(this, AlarmReceiver.class);
            pendingIntent = PendingIntent.getBroadcast(this, 0, intent, 0);

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

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

Java Coding for Alarm Receiver:

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



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

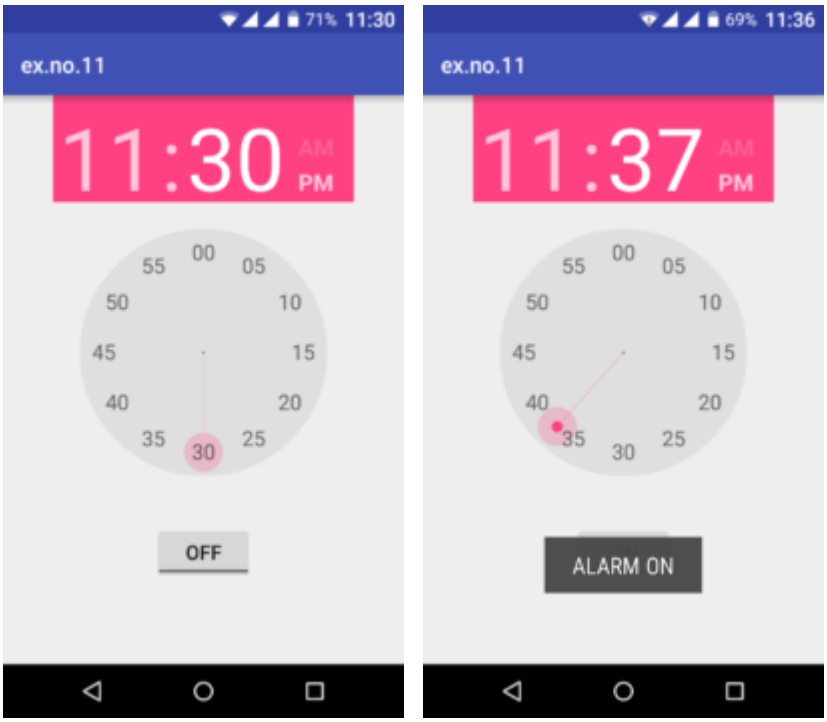
Code for AlarmReceiver.java:

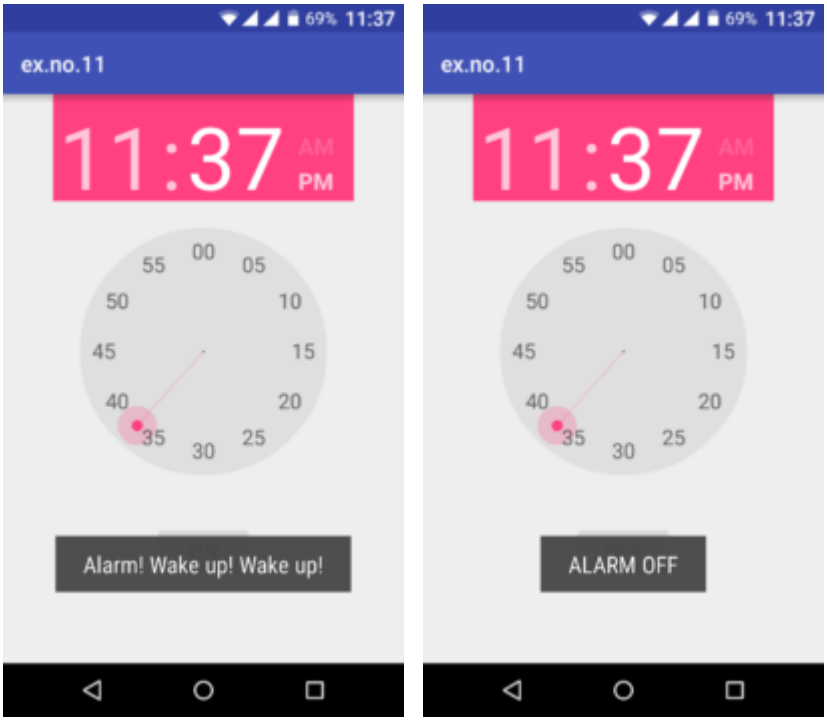
```
package com.example.exno11;
```

```
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.media.Ringtone;
import android.media.RingtoneManager;
import android.net.Uri;
import android.widget.Toast;

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

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

Output:





Result:

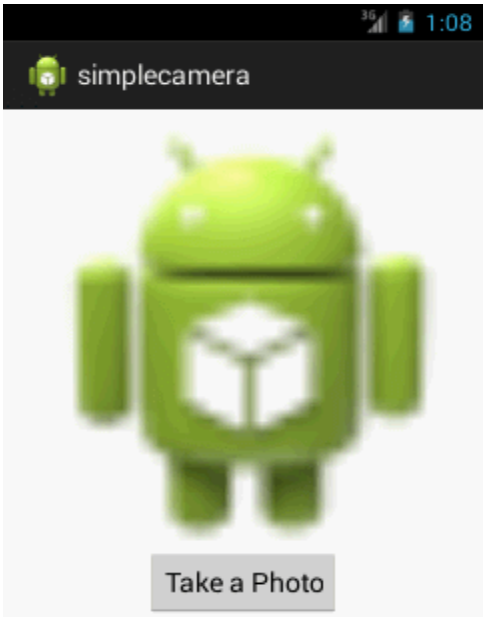
Thus Android Application that creates Alarm Clock is developed and executed successfully.

EXERCISE 10.1: [05]

Develop an android application to Alarm a clock and a button to off the alarm.

EXERCISE 10.2: [05]

Develop an android application to capture a photo from your mobile.



RESOURCES:

https://www.codingconnect.net/mobile-application-development-lab/
https://www.javatpoint.com/android-tutorial
https://www.tutorialspoint.com/android
https://developer.android.com/guide
Https://developer.Android.com/training/basics/firstapp/creating-project
Https://www.raywenderlich.com/120177/beginning-Android-development-tutorial-installingAndroid-studio
https://www.youtube.com/playlist?list=PLS1QulWo1RIbb1cYyzZpLFCKvdYV_yJ-E
Https://www.udemy.com/java-tutorial/
Https://www.w3schools.com/xml/
Http://www.Androidguys.com/

EXERCISE 11 – ANDROID ANALOG CLOCK AND DIGITAL CLOCK EXAMPLE

The **android.widget.AnalogClock** and **android.widget.DigitalClock** classes provides the functionality to display analog and digital clocks.

Android analog and digital clocks are used to show time in android application.

Android AnalogClock is the subclass of View class.

Android DigitalClock is the subclass of TextView class. Since Android API level 17, it is *deprecated*. You are recommended to use **TextClock** Instead.

The AnalogClock was deprecated in API level 23. This widget is no longer supported. Instead if you want to use AnalogClock in your application you need to hard code. It does not appear in API level 27 to drag from palette.

In android, you need to drag analog and digital clocks from the pallet to display analog and digital clocks.

activity_main.xml

Now, drag the analog and digital clocks, now the xml file will look like this.

File: activity_main.xml

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

<AnalogClock
android:id="@+id/analogClock1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentTop="true"
android:layout_centerHorizontal="true"
android:layout_marginLeft="136dp"
android:layout_marginTop="296dp"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />

<DigitalClock
android:id="@+id/digitalClock1"
android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"
android:layout_below="@+id/analogClock1"
android:layout_centerHorizontal="true"
android:layout_marginLeft="176dp"
android:layout_marginTop="84dp"
android:text="DigitalClock"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />
</android.support.constraint.ConstraintLayout>
```

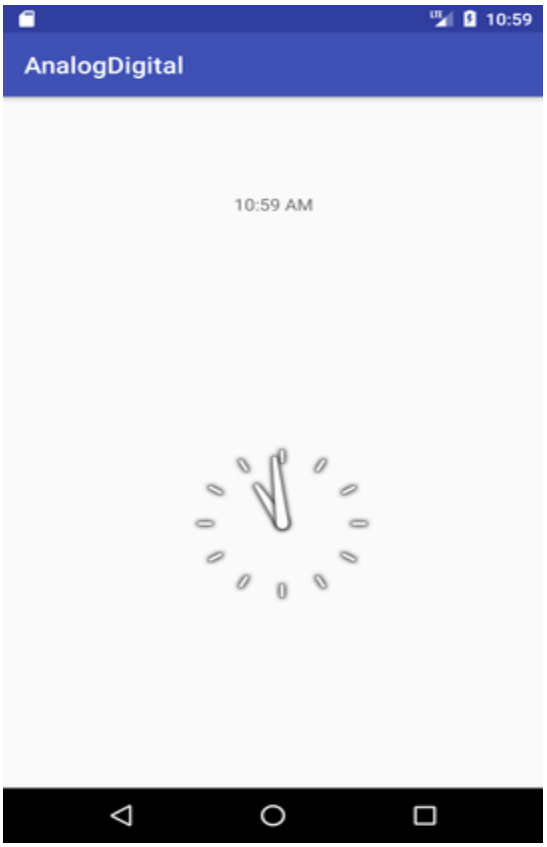
Activity class

We have not write any code here.

File: MainActivity.java

```
package example.javatpoint.com.analogdigital;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);

    }

}
```

Output:



TASK 11.1

Develop an android application to have an analog clock as the above example.

EXERCISE 1:

Develop an android application to display a digital clock on an activity. [10]

RESOURCES:

https://www.codingconnect.net/mobile-application-development-lab/
https://www.javatpoint.com/android-tutorial
https://www.tutorialspoint.com/android
https://developer.android.com/guide
Https://developer.Android.com/training/basics/firstapp/creating-project
Https://www.raywenderlich.com/120177/beginning-Android-development-tutorial-installingAndroid-studio
https://www.youtube.com/playlist?list=PLS1QulWo1RIbb1cYyzZpLFCKvdYV_yJ-E
Https://www.udemy.com/java-tutorial/
Https://www.w3schools.com/xml/
Http://www.Androidguys.com/

EXERCISE 12 – ANDROID MEDIA PLAYER EXAMPLE

We can play and control the audio files in android by the help of **MediaPlayer class**.

Here, we are going to see a simple example to play the audio file. In the next page, we will see the example to control the audio playback like start, stop, pause etc.

MediaPlayer class

The **android.media.MediaPlayer** class is used to control the audio or video files.

Activity class

Let's write the code of to play the audio file. Here, we are going to play maine.mp3 file located inside the sdcard/Music directory.

File: MainActivity.java

```
package com.example.audiomediaplayer1;


import android.media.MediaPlayer;
import android.net.Uri;
import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.widget.MediaController;
import android.widget.VideoView;


public class MainActivity extends Activity {


  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_main);


        MediaPlayer mp=new MediaPlayer();
        try{
            mp.setDataSource("/sdcard/Music/maine.mp3");//Write your location here
            mp.prepare();
            mp.start();


        }catch(Exception e){e.printStackTrace();}


    }


    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.activity_main, menu);
        return true;
    }


}
```

Android MediaPlayer Example of controlling the audio

Let's see a simple example to start, stop and pause the audio play.

activity_main.xml

Drag three buttons from pallete to start, stop and pause the audio play. Now the xml file will look like this:

File: MainActivity.java

```
<RelativeLayout xmlns:androclass="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".MainActivity" >

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_marginTop="30dp"
        android:text="Audio Controller" />

    <Button
        android:id="@+id/button1"
        style="?android:attr/buttonStyleSmall"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/textView1"
```

```
        android:layout_below="@+id/textView1"
        android:layout_marginTop="48dp"
        android:text="start" />

    <Button
        android:id="@+id/button2"
        style="?android:attr/buttonStyleSmall"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignTop="@+id/button1"
        android:layout_toRightOf="@+id/button1"
        android:text="pause" />

    <Button
        android:id="@+id/button3"
        style="?android:attr/buttonStyleSmall"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignTop="@+id/button2"
        android:layout_toRightOf="@+id/button2"
        android:text="stop" />

</RelativeLayout>
```

Activity class

Let's write the code to start, pause and stop the audio player.

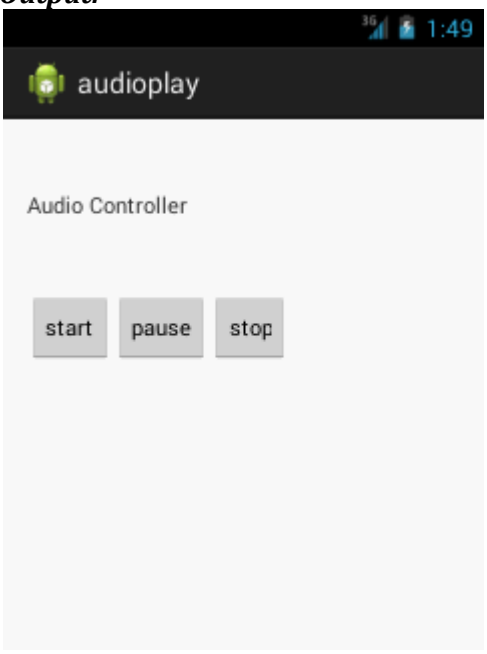
File: MainActivity.java

```
package com.example.audioplay;

import android.media.MediaPlayer;
import android.os.Bundle;
import android.os.Environment;
import android.app.Activity;
import android.view.Menu;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;

public class MainActivity extends Activity {
    Button start,pause,stop;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        start=(Button)findViewById(R.id.button1);
        pause=(Button)findViewById(R.id.button2);
        stop=(Button)findViewById(R.id.button3);
        //creating media player
        final MediaPlayer mp=new MediaPlayer();
        try{
```

```
            //you can change the path, here path is external directory(e.g. sdcard) /Music/maine
.mp3

        mp.setDataSource(Environment.getExternalStorageDirectory().getPath()+"/Music/main
e.mp3");



        mp.prepare();

        }catch(Exception e){e.printStackTrace();}



        start.setOnClickListener(new OnClickListener() {

            @Override

            public void onClick(View v) {

                mp.start();

            }

        });

        pause.setOnClickListener(new OnClickListener() {

            @Override

            public void onClick(View v) {

                mp.pause();

            }

        });

        stop.setOnClickListener(new OnClickListener() {

            @Override

            public void onClick(View v) {

                mp.stop();

            }

        });
```

```
    }
}
```

Output:

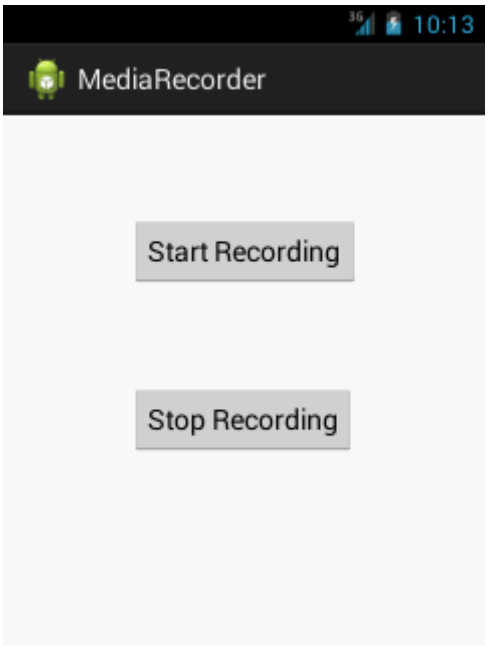


TASK 12.1

Develop an android application to Play, Stop and Pause audio and video files.

EXERCISE 12.1:

Develop an android application to record from media recorder from your mobile. [05]



EXERCISE 12.2:

Develop an android application to enable, disable and make discoverable Bluetooth programmatically. [05]

RESOURCES:

https://www.codingconnect.net/mobile-application-development-lab/
https://www.javatpoint.com/android-tutorial
https://www.tutorialspoint.com/android
https://developer.android.com/guide
Https://developer.Android.com/training/basics/firstapp/creating-project
Https://www.raywenderlich.com/120177/beginning-Android-development-tutorial-installingAndroid-studio
https://www.youtube.com/playlist?list=PLS1QulWo1RIbb1cYyzZpLFCKvdYV_yJ-E
Https://www.udemy.com/java-tutorial/
Https://www.w3schools.com/xml/
Http://www.Androidguys.com/

EXERCISE 13 – ANDROID GOOGLE MAP

Android provides facility to integrate Google map in our application. Google map displays your current location, navigate location direction, search location etc. We can also customize Google map according to our requirement.

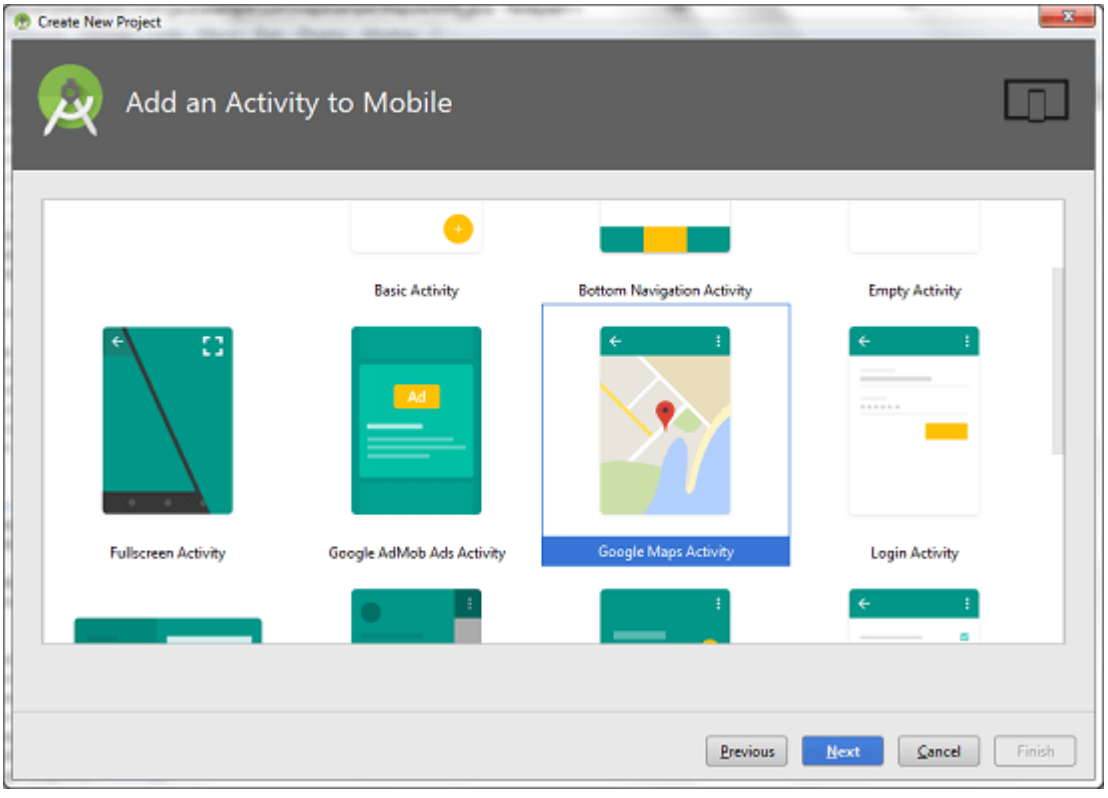
Types of Google Maps

There are four different types of Google maps, as well as an optional to no map at all. Each of them gives different view on map. These maps are as follow:

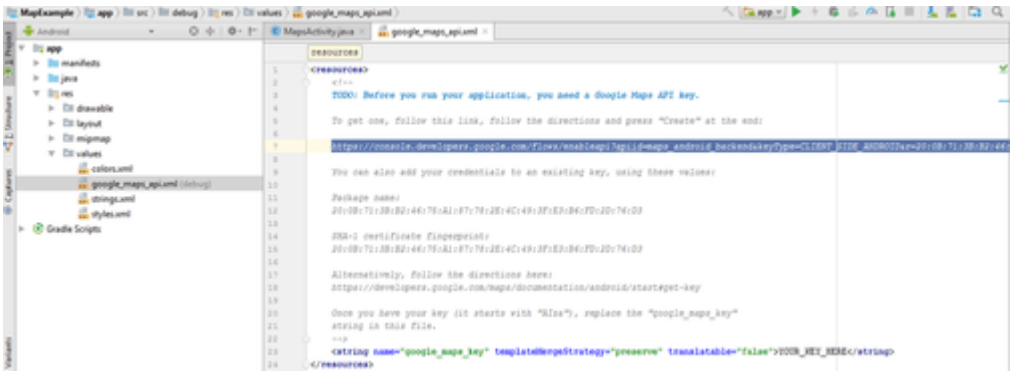
1. **Normal:** This type of map displays typical road map, natural features like river and some features build by humans.
2. **Hybrid:** This type of map displays satellite photograph data with typical road maps. It also displays road and feature labels.
3. **Satellite:** Satellite type displays satellite photograph data, but doesn't display road and feature labels.
4. **Terrain:** This type displays photographic data. This includes colors, contour lines and labels and perspective shading.
5. **None:** This type displays an empty grid with no tiles loaded.

Example of Google Map

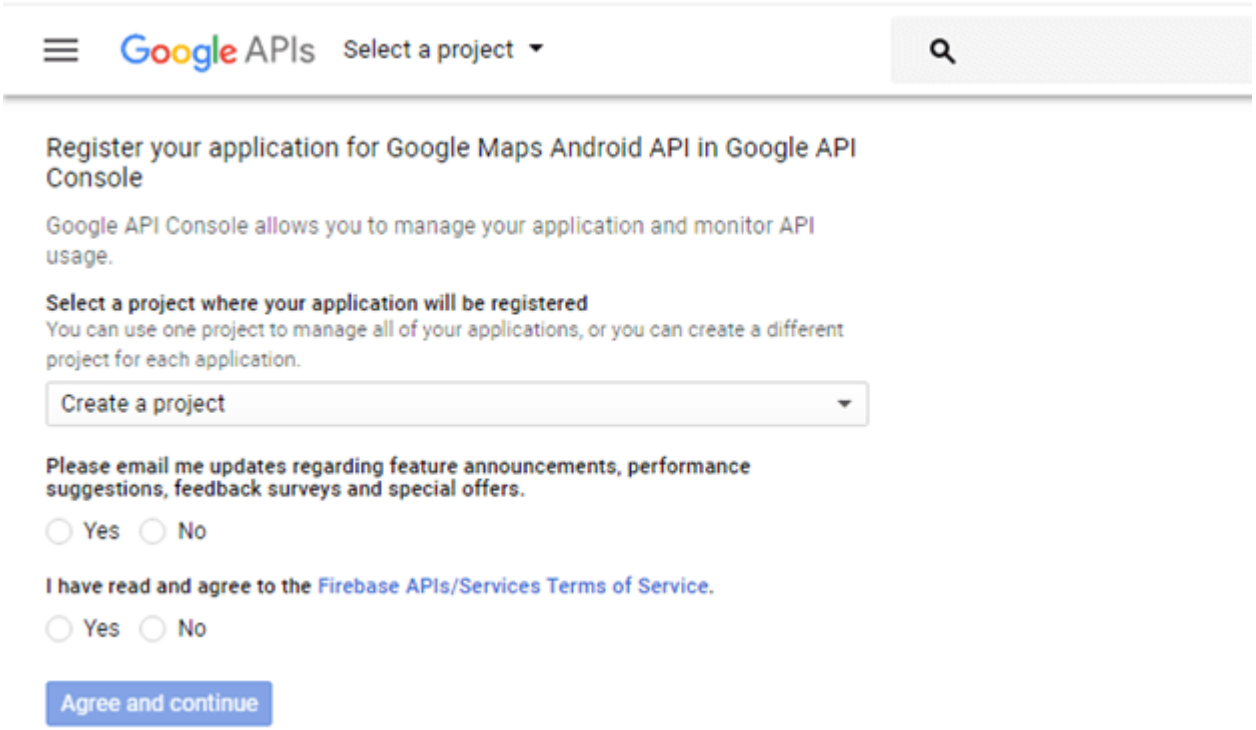
Let's create an example of Google map integrating within our app. For doing this we select Google Maps Activity.



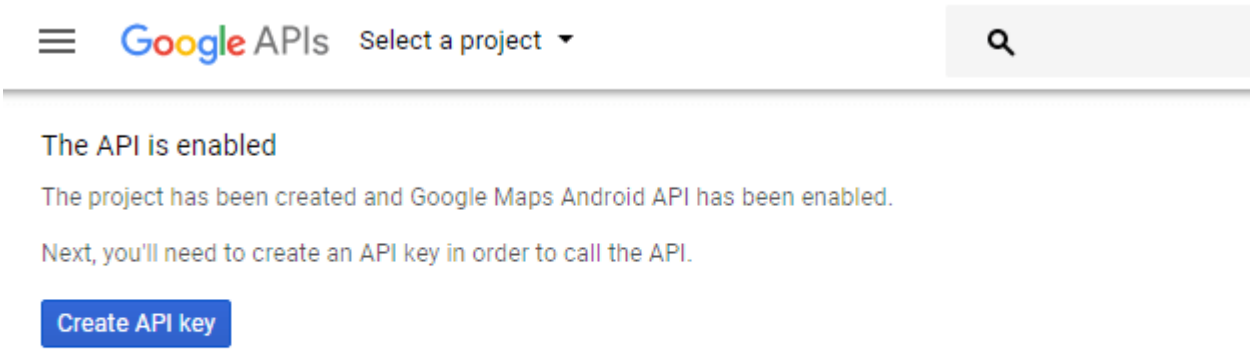
Copy the URL from google_map_api.xml file to generate Google map key.



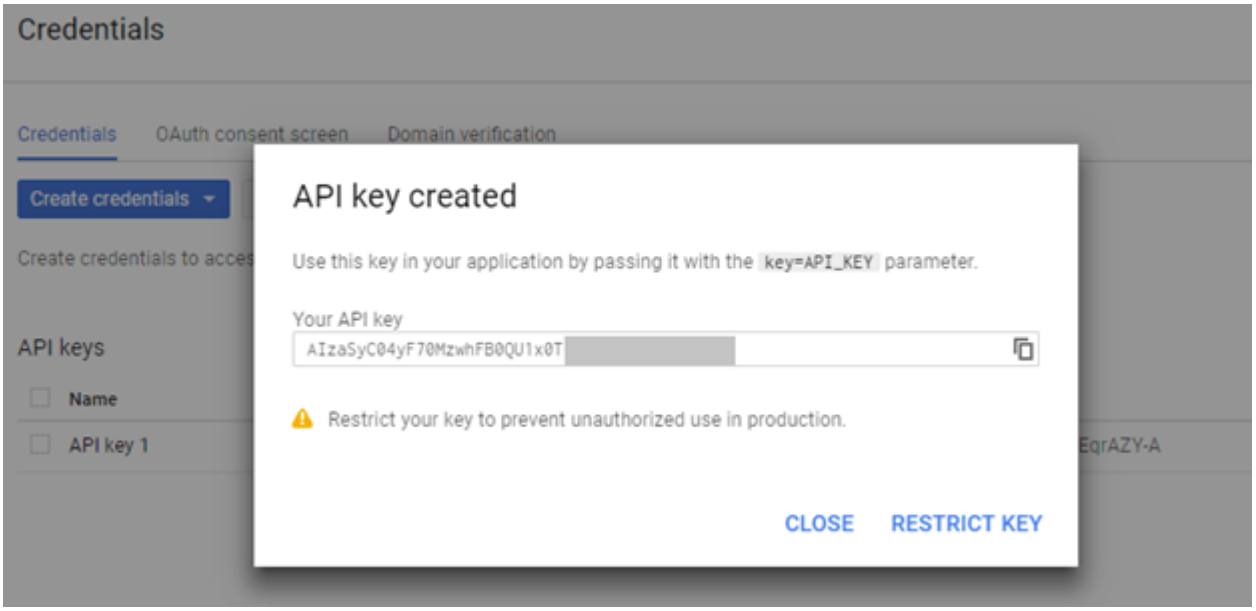
Paste the copied URL at the browser. It will open the following page.



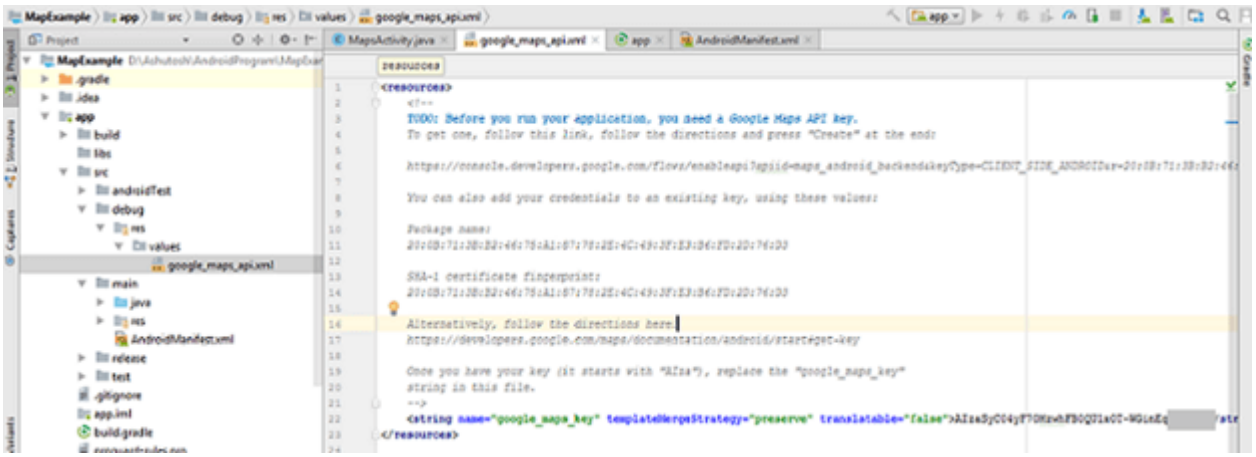
Click on Create API key to generate API key.



After clicking on Create API key, it will generate our API key displaying the following screen.



Copy this generated API key in our *google_map_api.xml* file



activity_maps.xml

```
<fragment xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:map="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:id="@+id/map"
android:name="com.google.android.gms.maps.SupportMapFragment"
android:layout_width="match_parent"
android:layout_height="match_parent"
   tools:context="example.com.mapexample.MapsActivity" />
```

MapsActivity.java

To get the GoogleMap object in our MapsActivity.java class we need to implement the OnMapReadyCallback interface and override the onMapReady() callback method.

```
package example.com.mapexample;



import android.support.v4.app.FragmentActivity;

import android.os.Bundle;

import com.google.android.gms.maps.CameraUpdateFactory;

import com.google.android.gms.maps.GoogleMap;

import com.google.android.gms.maps.OnMapReadyCallback;

import com.google.android.gms.maps.SupportMapFragment;

import com.google.android.gms.maps.model.LatLng;

import com.google.android.gms.maps.model.MarkerOptions;


public class MapsActivity extends FragmentActivity implements OnMapReadyCallback{


   private GoogleMap mMap;


   @Override

   protected void onCreate(Bundle savedInstanceState) {

      super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_maps);

        // Obtain the SupportMapFragment and get notified when the map is ready to be used.

        SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentMan
    ager()

                .findFragmentById(R.id.map);

        mapFragment.getMapAsync(this);



    }



    @Override

    public void onMapReady(GoogleMap googleMap) {

        mMap = googleMap;



        // Add a marker in Sydney and move the camera

        LatLng sydney = new LatLng(-34, 151);

        mMap.addMarker(new MarkerOptions().position(sydney).title("Marker in Sydney"));

        mMap.moveCamera(CameraUpdateFactory.newLatLng(sydney));



    }

}
```

Required Permission

Add the following user-permission in AndroidManifest.xml file.

```
1. <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
2. <uses-
   permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
3. <uses-permission android:name="android.permission.INTERNET" />
```

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="example.com.mapexample">
    <!--
        The ACCESS_COARSE/FINE_LOCATION permissions are not required to use
        Google Maps Android API v2, but you must specify either coarse or fine
        location permissions for the 'MyLocation' functionality.
    -->
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-
permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.INTERNET" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">

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

```
        <activity
            android:name=".MapsActivity"
            android:label="@string/title_activity_maps">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />


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


</manifest>
```

build.gradel

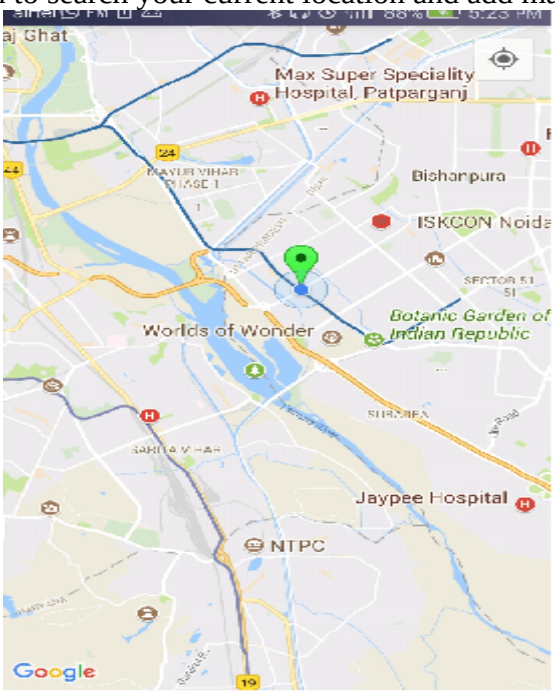
Add the following dependencies in *build.gradel* file.

```
1.  dependencies {
2.      implementation fileTree(dir: 'libs', include: ['*.jar'])
3.      implementation 'com.android.support:appcompat-v7:26.1.0'
4.      implementation 'com.google.android.gms:play-services-maps:11.8.0'
5.      testImplementation 'junit:junit:4.12'
6.      androidTestImplementation 'com.android.support.test:runner:1.0.1'
7.      androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.1'
8.  }
```

Output

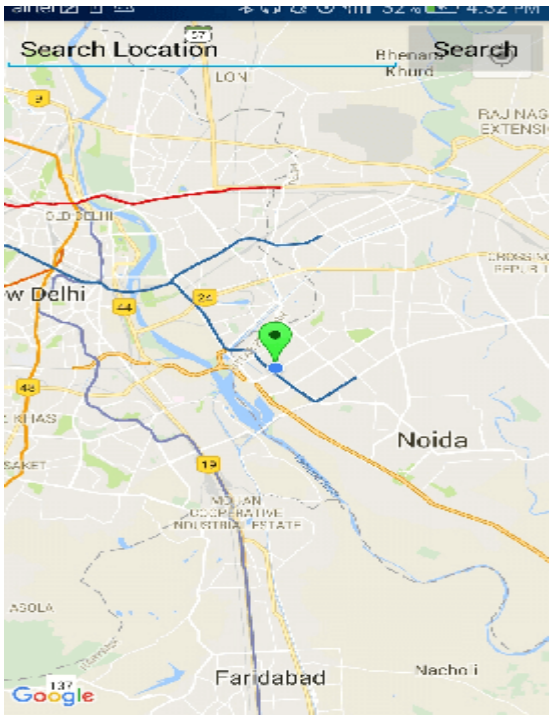
EXERCISE 13.1:

Develop an android application to search your current location and add marker to it. [05]



EXERCISE 13.2:

Develop an android application to search your current location using Geo coder. [05]



RESOURCES:

https://www.codingconnect.net/mobile-application-development-lab/
https://www.javatpoint.com/android-tutorial
https://www.tutorialspoint.com/android
https://developer.android.com/guide
Https://developer.Android.com/training/basics/firstapp/creating-project
Https://www.raywenderlich.com/120177/beginning-Android-development-tutorial-installingAndroid-studio
https://www.youtube.com/playlist?list=PLS1QulWo1RIbb1cYyzZpLFCKvdYV_yJ-E
Https://www.udemy.com/java-tutorial/
Https://www.w3schools.com/xml/
Http://www.Androidguys.com/

EXERCISE 14 – ANDROID TEXTTOSPEECH TUTORIAL

In android, you can convert your text into speech by the help of **TextToSpeech** class. After completion of the conversion, you can playback or create the sound file.

Constructor of TextToSpeech class

- **TextToSpeech(Context context, TextToSpeech.OnInitListener)**

TextToSpeech.OnInitListener Interface

You need to implement TextToSpeech.OnInitListener interface, for performing event handling on TextToSpeech engine.

Android TextToSpeech Example

Let's write the code to convert text into voice.

activity_main.xml

Drag one textview, one edittext and one button for the layout. Now the activity_main.xml file will look like this:

File: activity_main.xml

```
<RelativeLayout xmlns:androclass="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" >

    <EditText
        android:id="@+id/editText1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
```

```
    android:layout_alignParentTop="true"
    android:layout_marginLeft="77dp"
    android:layout_marginTop="42dp"
    android:ems="10" >

    <requestFocus />
</EditText>

<Button
    android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/editText1"
    android:layout_below="@+id/editText1"
    android:layout_marginLeft="59dp"
    android:layout_marginTop="39dp"
    android:text="Speak" />

<TextView
    android:id="@+id/textView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignBaseline="@+id/editText1"
    android:layout_alignBottom="@+id/editText1"
    android:layout_alignParentLeft="true"
```

```
        android:text="Enter Text:" />


    </RelativeLayout>
```

Activity class

Let's see the code to speak the given text.

File: MainActivity.java

```
    package com.example.texttospeech;


    import android.os.Bundle;

    import android.app.Activity;

    import android.view.Menu;

    import java.util.Locale;


    import android.app.Activity;

    import android.os.Bundle;

    import android.speech.tts.TextToSpeech;

    import android.util.Log;

    import android.view.View;

    import android.widget.Button;

    import android.widget.EditText;

    public class MainActivity extends Activity implements

    TextToSpeech.OnInitListener {

    /** Called when the activity is first created. */


    private TextToSpeech tts;
```

```
private Button buttonSpeak;
private EditText editText;

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

tts = new TextToSpeech(this, this);
buttonSpeak = (Button) findViewById(R.id.button1);
editText = (EditText) findViewById(R.id.editText1);

buttonSpeak.setOnClickListener(new View.OnClickListener() {
   @Override
   public void onClick(View arg0) {
      speakOut();
   }

});
}

@Override
public void onDestroy() {
// Don't forget to shutdown tts!
if (tts != null) {
```

```
    tts.stop();
    tts.shutdown();
}
super.onDestroy();
}

@Override
public void onInit(int status) {

if (status == TextToSpeech.SUCCESS) {

    int result = tts.setLanguage(Locale.US);

    if (result == TextToSpeech.LANG_MISSING_DATA
            || result == TextToSpeech.LANG_NOT_SUPPORTED) {
        Log.e("TTS", "This Language is not supported");
    } else {
        buttonSpeak.setEnabled(true);
        speakOut();
    }

} else {
    Log.e("TTS", "Initilization Failed!");
}
```

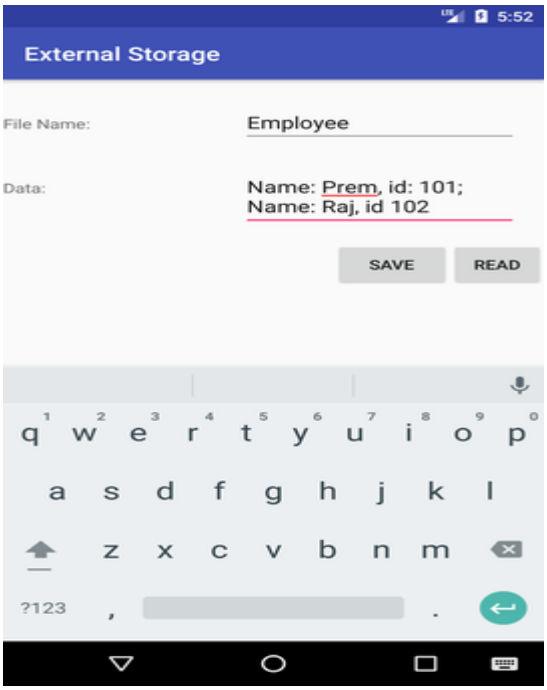
```
}

private void speakOut() {
String text = editText.getText().toString();
tts.speak(text, TextToSpeech.QUEUE_FLUSH, null);
}

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.activity_main, menu);
        return true;
    }

}
```

TASK 14.1

Make an app using the above text to speech exercise and display output (as a screen shot) and code also.

EXERCISE 14.1:

Develop an android application to read and save data to some external source such as SD card. [10]



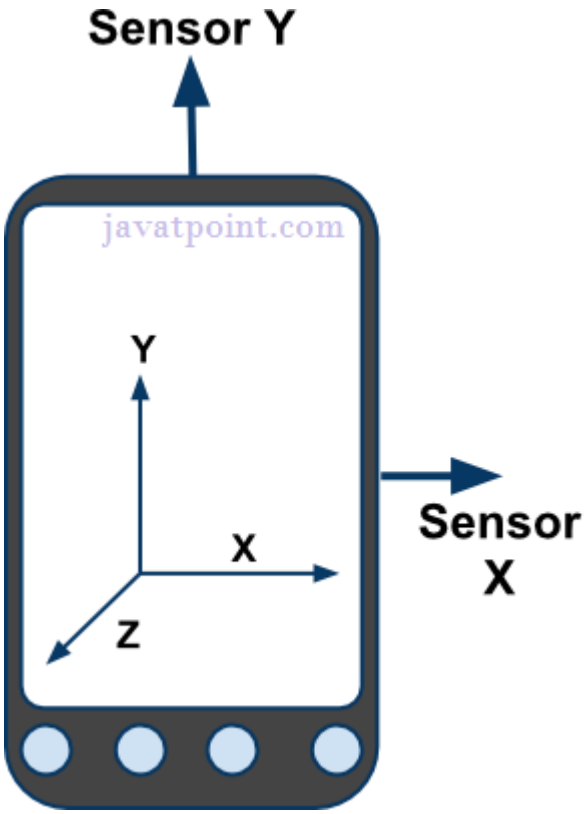
RESOURCES:

https://www.codingconnect.net/mobile-application-development-lab/
https://www.javatpoint.com/android-tutorial
https://www.tutorialspoint.com/android
https://developer.android.com/guide
Https://developer.Android.com/training/basics/firstapp/creating-project
Https://www.raywenderlich.com/120177/beginning-Android-development-tutorial-installingAndroid-studio
https://www.youtube.com/playlist?list=PLS1QulWo1RIbb1cYyzZpLFCKvdYV_yJ-E
Https://www.udemy.com/java-tutorial/
Https://www.w3schools.com/xml/
Http://www.Androidguys.com/

EXERCISE 15 – ANDROID SENSOR TUTORIAL

Sensors can be used to monitor the three-dimensional device movement or change in the environment of the device.

Android provides sensor API to work with different types of sensors.



Types of Sensors

Android supports three types of sensors:

1) Motion Sensors

These are used to measure acceleration forces and rotational forces along with three axes.

2) Position Sensors

These are used to measure the physical position of device.

3) Environmental Sensors

These are used to measure the environmental changes such as temperature, humidity etc.

Android Sensor API

Android sensor api provides many classes and interface. The important classes and interfaces of sensor api are as follows:

1) SensorManager class

The **android.hardware.SensorManager** class provides methods :

- to get sensor instance,
- to access and list sensors,
- to register and unregister sensor listeners etc.

You can get the instance of SensorManager by calling the method getSystemService() and passing the SENSOR_SERVICE constant in it.

```
1. SensorManager sm = (SensorManager)getSystemService(SENSOR_SERVICE);
```

2) Sensor class

The **android.hardware.Sensor** class provides methods to get information of the sensor such as sensor name, sensor type, sensor resolution, sensor type etc.

3) SensorEvent class

Its instance is created by the system. It provides information about the sensor.

4) SensorEventListener interface

It provides two call back methods to get information when sensor values (x,y and z) change or sensor accuracy changes.

Public and abstract methods	Description
void onAccuracyChanged(Sensor sensor, int accuracy)	it is called when sensor accuracy is changed.
void onSensorChanged(SensorEvent event)	it is called when sensor values are changed.

Android simple sensor app example

Let's see the two sensor examples.

1. A sensor example that prints x, y and z axis values. Here, we are going to see that.
2. A sensor example that changes the background color when device is shuffled. Click for changing background color of activity sensor example

activity_main.xml

There is only one textview in this file.

File: activity_main.xml

```
<RelativeLayout xmlns:androclass="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" >

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        android:layout_marginLeft="92dp"
        android:layout_marginTop="114dp"
        android:text="TextView" />

</RelativeLayout>
```

Activity class

Let's write the code that prints values of x axis, y axis and z axis.

File: MainActivity.java

```
package com.example.sensorsimple;
import android.app.Activity;
import android.os.Bundle;
import android.widget.TextView;
import android.widget.Toast;
import android.hardware.SensorManager;
import android.hardware.SensorEventListener;
import android.hardware.SensorEvent;
```

```
import android.hardware.Sensor;
import java.util.List;
public class MainActivity extends Activity {
    SensorManager sm = null;
    TextView textView1 = null;
    List list;

    SensorEventListener sel = new SensorEventListener(){
        public void onAccuracyChanged(Sensor sensor, int accuracy) {}
        public void onSensorChanged(SensorEvent event) {
            float[] values = event.values;
            textView1.setText("x: "+values[0]+"\ny: "+values[1]+"\nz: "+values[2]);
        }
    };

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

        /* Get a SensorManager instance */
        sm = (SensorManager)getSystemService(SENSOR_SERVICE);

        textView1 = (TextView)findViewById(R.id.textView1);

        list = sm.getSensorList(Sensor.TYPE_ACCELEROMETER);
```

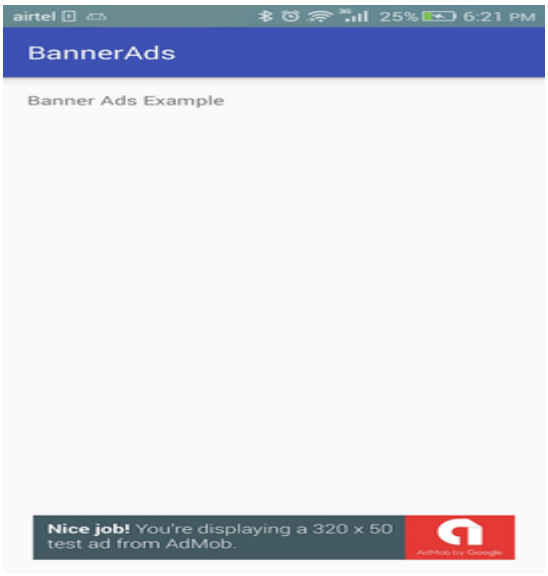
```
        if(list.size()>0){

          sm.registerListener(sel, (Sensor) list.get(0), SensorManager.SENSOR_DELAY_NOR
    MAL);

        }else{

          Toast.makeText(getBaseContext(), "Error: No Accelerometer.", Toast.LENGTH_LO
    NG).show();

        }

      }


      @Override

      protected void onStop() {

        if(list.size()>0){

          sm.unregisterListener(sel);

        }

        super.onStop();

      }

}
```

TASK 15.1

Execute the Mobile sensor code and show the output of the program.

EXERCISE 15.1:

Develop an android application to display Google Ad Mob banner ads. [10]



RESOURCES:

https://www.codingconnect.net/mobile-application-development-lab/
https://www.javatpoint.com/android-tutorial
https://www.tutorialspoint.com/android
https://developer.android.com/guide
Https://developer.Android.com/training/basics/firstapp/creating-project
Https://www.raywenderlich.com/120177/beginning-Android-development-tutorial-installingAndroid-studio
https://www.youtube.com/playlist?list=PLS1QulWo1RIbb1cYyzZpLFCKvdYV_yJ-E
Https://www.udemy.com/java-tutorial/
Https://www.w3schools.com/xml/
Http://www.Androidguys.com/

EXERCISE 16 – CROSS-PLATFORM TOOLS FOR APP DEVELOPMENT (ANDROID, IOS AND WINDOWS PHONE)

Today startups and SMEs find cross-platform as an excellent form of technology to develop an app on multiple **platforms like Android, iOS, and Windows** simultaneously.

This means by building a single app you can target both Android and iOS, thus, maximizing your reach to the target audience.

In fact, the **cross-platform application development market** surpassed the figure of **$7.9** in **2019**.

Ideally, cross-platform technology delivers native-like apps because of the advent of advanced tools and technologies that allow developers to build apps that may appear similar to native apps.

Also, in such a scenario when the number of apps in the Google Play Store was most recently placed at around **$2.6 million apps** in March 2019. Businesses wouldn't want to risk missing their presence on Google play store or any other platform.

Budgeting always an issue for businesses if they go for native apps, this is where cross-platform technology has emerged as the premium choice for businesses that aim to build their app multiple platforms.

1. Adobe PhoneGap

PhoneGap is owned by Adobe and is one of the **best cross-platform development tools to use in 2019**. It's based on the open source framework Apache Cordova that gives you access to complete set of PhoneGap toolset which helps streamline the app development process and include the options:

Debugging tools allow you to inspect HTML, CSS, and debug codes in JavaScript. Here is the list of tools:

For iOS App Development

Safari Web Inspector Tool

Steps to Use:

1. Take your iOS device and connect to your computer.
2. Now, install and launch Safari on your system.
3. Make your PhoneGap application launched on iOS Device.
4. Open the menu of Safari Develop, and look for your iOS Device in the list.
5. Select "PhoneGap Webview" listed under your iOS device.

For Android App Development

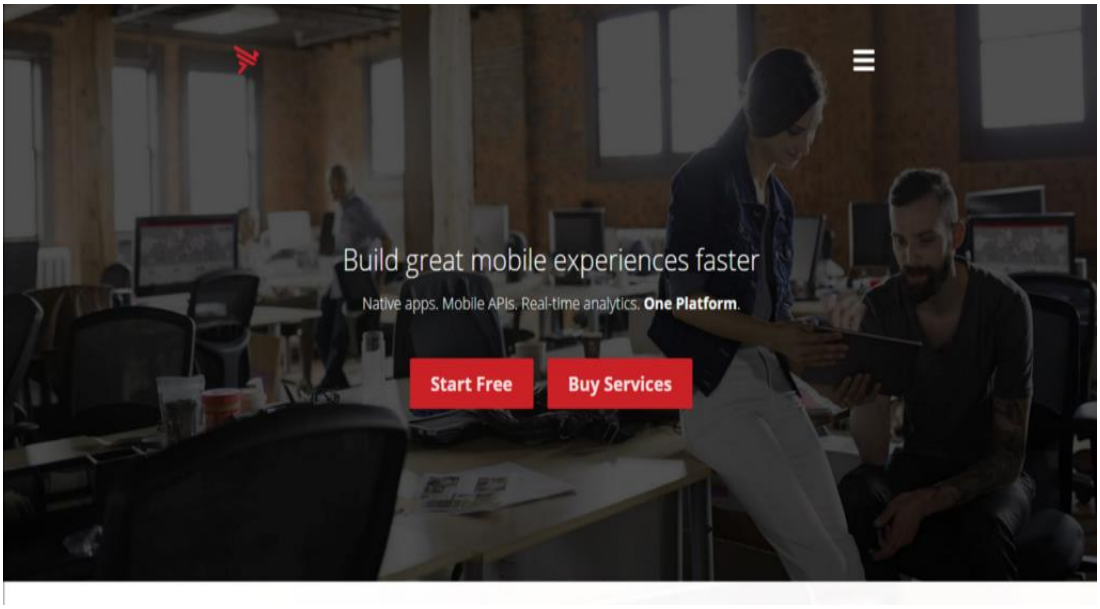
Chrome Developer Tool

Steps to Use:

1. Make sure your Android test device supports all the developer options.
2. Now launch your Google Chrome web browser.
3. Look for chrome://inspect in Chrome.
4. Select PhoneGap Application on your device.
5. Developer tools will launch.

One of the reasons why I am suggesting PhoneGap is because anyone can learn how to use their tools, even if you don't have experience of using them. PhoneGap takes care of the development process by compiling all your work in the Cloud, so you don't need to maintain native SDKs.

2. Appcelerator

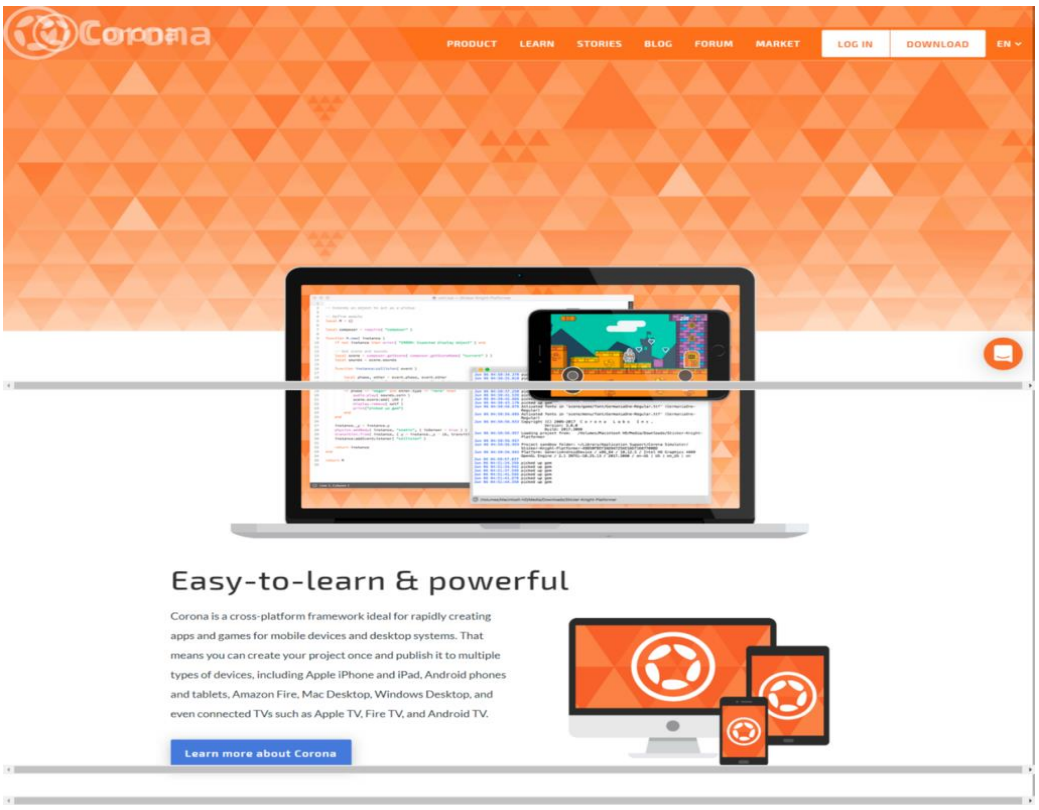


Appcelerator is a **cross-platform mobile app development platform** that helps get your app ready in a faster way by simplifying the whole process. By using a single JavaScript code you can build native-like apps and mobile apps with cloud-like performance. Another top benefit of Appcelerator is their quality as it can be used for building apps for any device or operating system.

The tool also makes it easy for you to use and test your apps using the automated mobile tests that allow you to measure your app usage and results of your app project. You can detect bugs, crashes, and also make some adjustments to improve the overall performance of your app.

With Appcelerator, you will be provided with access to Hyperloop that is one of the best cross-platform APIs for the multi-platform application development.

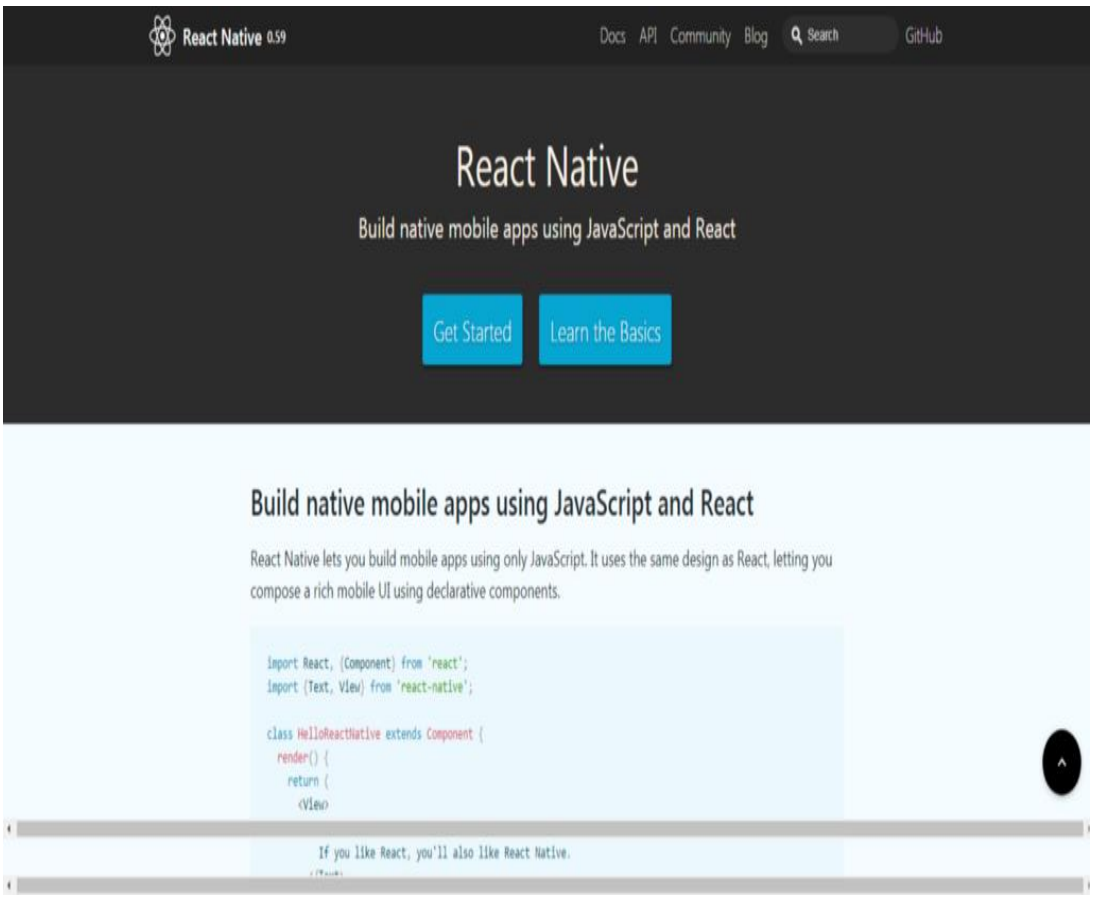
3. Corona



Corona is a cross-platform ideal for creating games and **apps for mobile devices, desktop, and tv devices** using just one code base. This tool speeds up your coding process and you can easily update your code, save the changes, and instantly see the results on real devices. With Corona, your apps are optimized for performance because of the lightweight scripting power of Lua that enhances your app performance at every level.

Corona is free to use cross-platform app development tool that primarily used in 2d games as it's great to use for high-quality graphics and high-speed development of games.

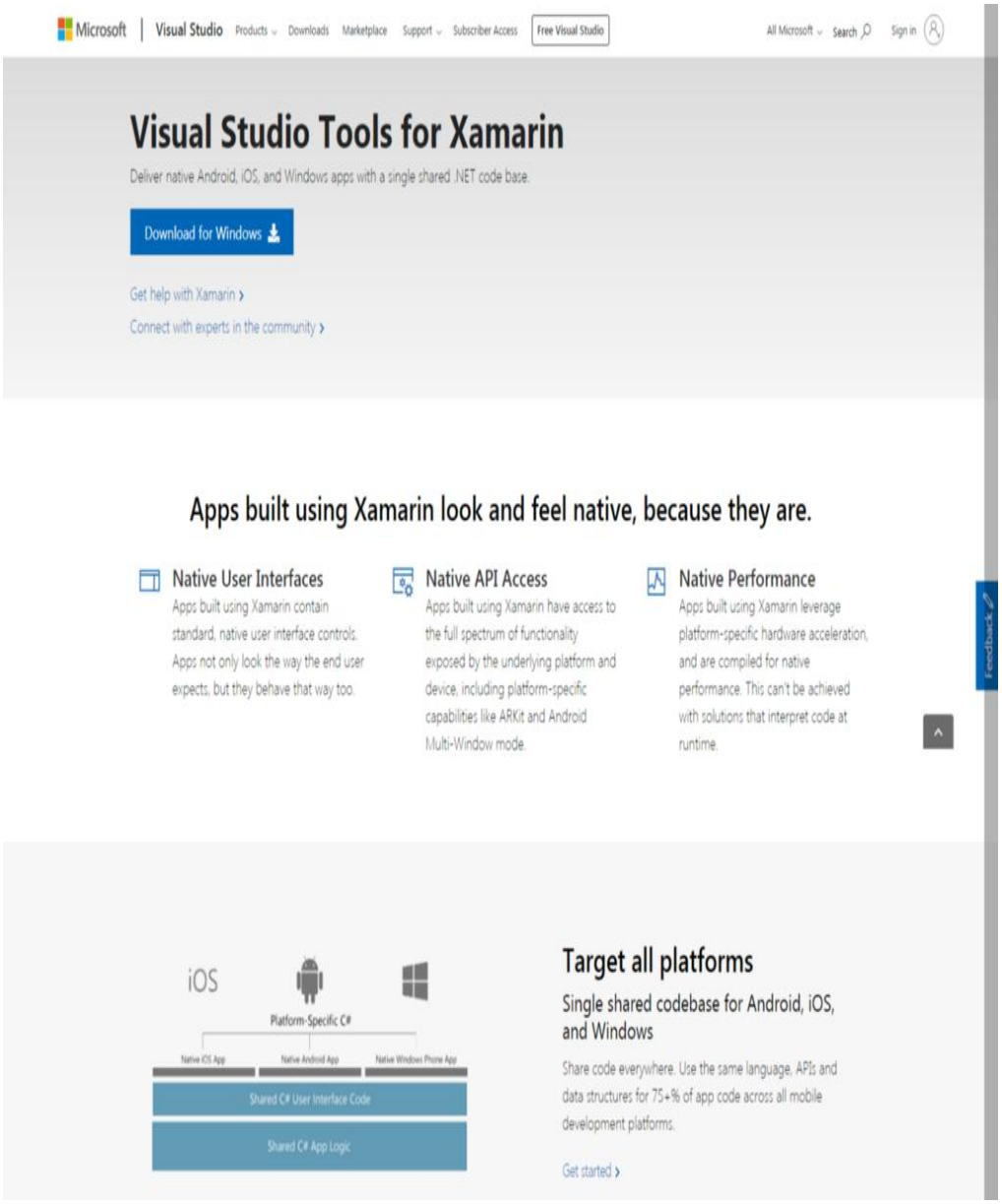
4. React Native



React Native allow you to create native applications and uses JavaScript as a programming language to build apps. The strong side of React Native is that you can write modules in languages such as C, Swift, and Java. The best part is you can work on image editing and video processing that aren't possible with the other API frameworks.

React Native is unquestionably the best platform to use for cross-platform app development because it interprets your source code and convert it to the native elements in less time. Both Facebook and Instagram have used React Native to build their native apps that are the most used applications of the world. So, you can trust on React Native.

5. Xamarin



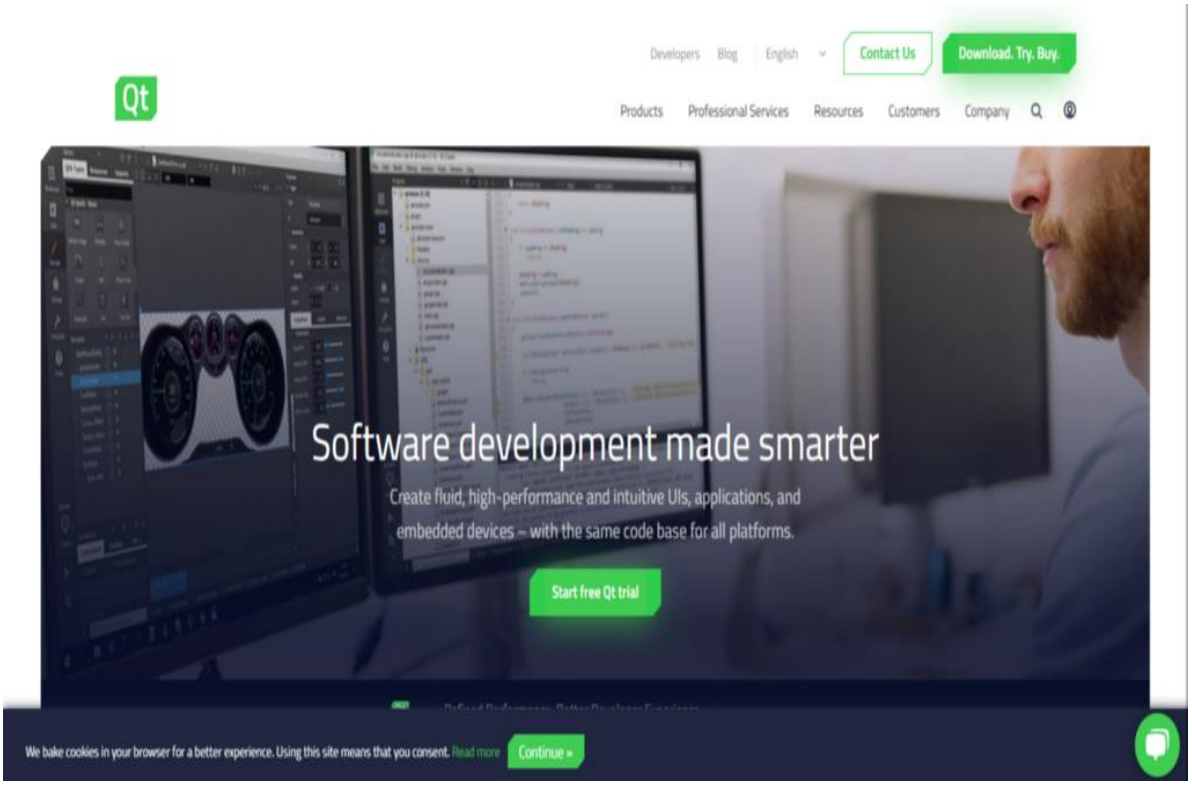
Microsoft Visual Studio **Xamarin's** allows you to build apps for different platforms such as Windows, iOS, and Android using a single .net code.

The best part of **Xamarin cross-platform tool** is that all the apps built on it look and feel like native apps that is because it uses the native interfaces that work the same way a user wants to use them.

With Xamarin, you can give your app a platform-specific hardware boosts to achieve the performance similar to native apps. Also, most of your coding approx. 75% will be the same, regardless of the platform you're building your mobile application for. Xamarin works on a single code by identifying it and accelerates the process for cross-platform mobile app development.

Xamarin works on both Mac and PC systems and offers you tools such as debugging, UI design tool, and code editing.

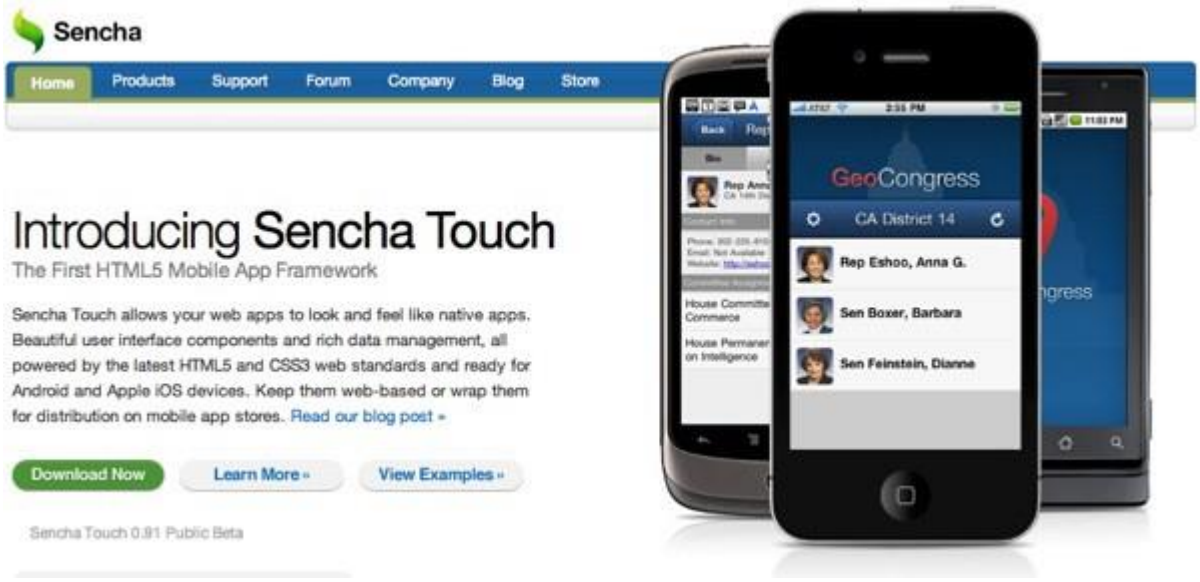
6. Qt



Qt is the **best cross-platform tool for mobile app development**. Why I'm counting this tool in the best cross-platform tools is because of its quality features that allow creating fluid, UIs, applications, and embedded devices with the same code for Android, iOS, and Windows.

If your app is not performing well and you want to rework on it, you can easily make changes to your app using Qt that will automatically make all the changes applied to your app. This software tool also allows you to see how your app is performing on different platforms. Moreover, it's easy to use and don't have a complex interface like some other cross-platform tools I've seen.

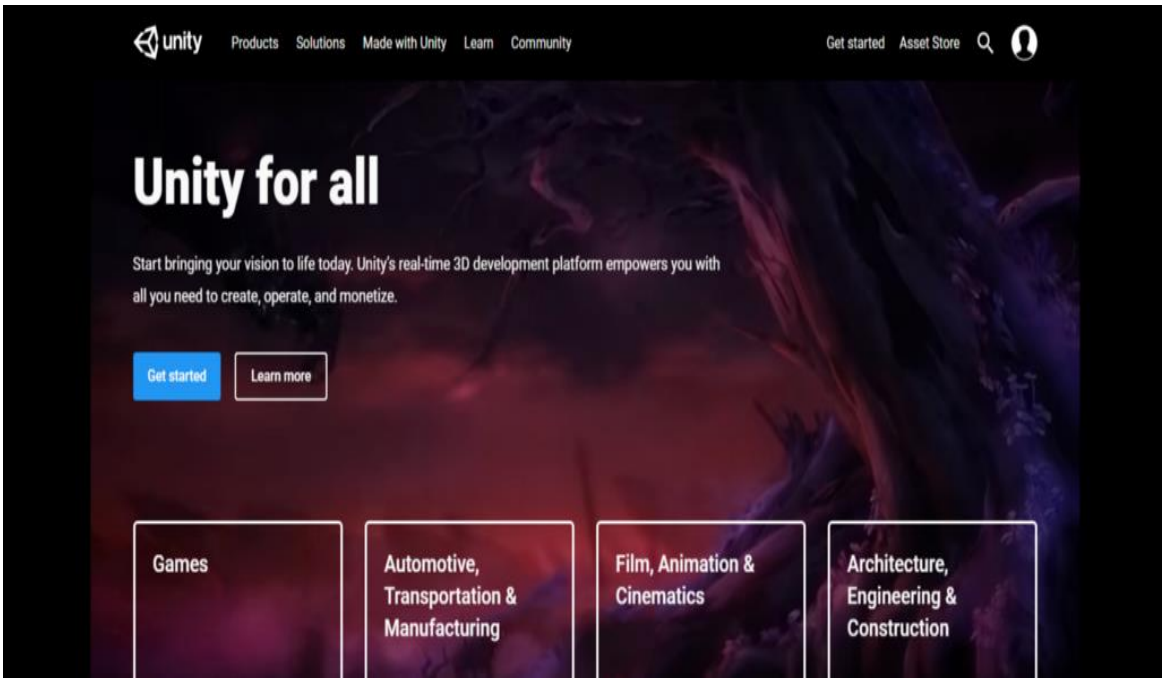
7. Sencha



With **Sencha** you will get all the modern Java and JavaScript frameworks that help you build your web apps easily for any device. It provides you 115+ fully supported and test UI components that you can easily integrate into your apps. It is one of the most comprehensive tools to **perform end-to-end testing of apps on all the platforms**.

In addition, Sencha provides you with the "Themer" to create reusable themes by customizing themes built on iOS, Ext JS, ExtAngular, and ExtReact. Sencha offers a data visualization table that makes it easier for you to track your app information. This also makes it possible for you to organize your app content and how your content is displayed on the browser, device, and screen size.

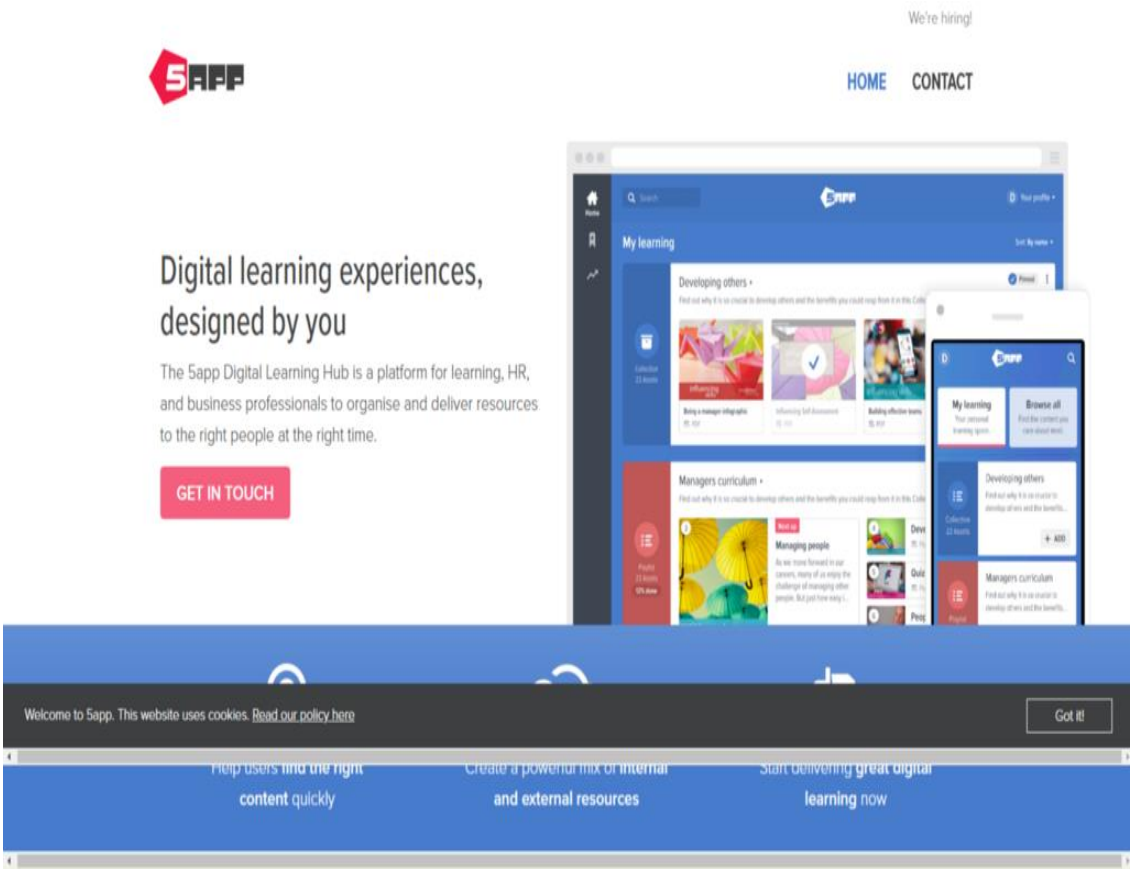
8. Unity3D



This **cross-platform app development tool** is so popular because of its graphics quality that is absolutely incredible. It's so easy to use this tool and you can use it for more than just a mobile app. With Unity3D tool you can export your app or games to 17 platforms that include—iOS, Android, Windows, Xbox, PlayStation, Linux, Web, and Wii.

Unity3d can also be used to track user analytics and share your app on social networks. You can also connect with the network of Unity3D developers called Unity Connect to find help and get your questions answered if you're having tech issues with coding or something else.

9. 5App



5App is a unique tool designed specifically for businesses into learning, HR consulting, and firms that want to organize and deliver resources to their employees or to the right people at the right time.

5Apps uses HTML5 and JavaScript for coding of apps and emphasis on the security of app data. The tool allows you to quickly create relevant content to support your employees' learning and performance.

The finished app is compatible with both Android and iOS devices, so you can choose accordingly as per your company's needs.

10. FLUTTER

Flutter, built by Google, takes cross-platform development to a new level. It was first released in May 2017, and its first stable 1.0 release was made in December 2018.

Flutter is on the bleeding edge of cross-platform development, and it deserves special attention. It's so close to the "code once, deploy twice" phenomenon, by completely sidestepping the native platforms. And it's quickly gaining speed in the developer community.

The way Flutter works is by compiling Dart source code to native code, which runs on the Dart virtual machine. You can compare this to Xamarin, which runs native code directly on the smartphone hardware, and to React Native, which runs interpreted JavaScript code in a native app.

A particularly interesting feature of Flutter is called *Hot Reload*. With Hot Reload, changes to the source code of an app can be directly injected at runtime. It's like reloading a web page, without needing to recompile the entire app. Hot Reload also retains the state of the app, so you can code and interact with the app at the same time.

Similarly to other tools, Flutter provides shared code for platform-specific features such as Android's and Apple's UI design, as well as the option to build platform-specific plugins. Platform-specific APIs and SDKs can be used natively.

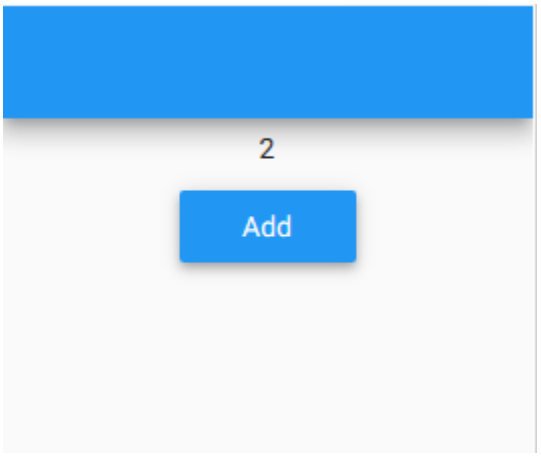
EXERCISE 16.1:

Develop a Digital Counter app in Flutter. Run the below program and show the output. [05]

DART:

```
import 'package:flutter/material.dart';

class Counter extends StatefulWidget {
  _CounterState createState() => _CounterState();
}

class _CounterState extends State<Counter> {
  double val;

  void initState() {
    super.initState();
    val = 0;
  }

  void change() {
    setState(() {
      val += 1;
    });
  }

  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(),
      body: Center(
        child: Column(
```

```
        children: <Widget>[
          Padding(
              padding: const EdgeInsets.all(8.0),
              child: Center(child: Text('$val'))),
          MaterialButton(
            color: Theme.of(context).primaryColor,
            child: Text(
              'Add',
              style: TextStyle(color: Colors.white),
            ),
            onPressed: () => change(),
          ),
        ],
      ),
    ),
  );
 }
}

class MyApp extends StatelessWidget {
 @override
 Widget build(BuildContext context) {
  return MaterialApp(
    debugShowCheckedModeBanner: false,
    home: Center(
      child: Container(
        child: Counter(),
      ),
    ),
  );
 }
}

Future<void> main() async {
 runApp(MyApp()); }
```



EXERCISE 16.2:

Develop a spinning image app in Flutter. Run the below program and show the output. [05]

DART:

```
import 'package:flutter/material.dart';

void main() async {
  runApp(
    MaterialApp(
      debugShowCheckedModeBanner: false,
      home: Scaffold(
        body: MyApp(),
      ),
    ),
  );
}

class MyApp extends StatefulWidget {
  @override
  _MyAppState createState() => _MyAppState();
}

class _MyAppState extends State<MyApp>
    with SingleTickerProviderStateMixin {
  AnimationController controller;
  Animation<double> animation;

  @override
  void initState() {
    super.initState();

    controller = AnimationController(
      duration: Duration(seconds: 1),
      vsync: this,
    );

    animation = CurvedAnimation(
      parent: controller,
      curve: Curves.easeInOutCubic,
    ).drive(Tween(begin: 0, end: 2));
  }

  @override
  void dispose() {
    controller.dispose();
    super.dispose();
  }

  @override
  Widget build(BuildContext context) {
```

```
return GestureDetector(
  onTap: () {
    controller
      ..reset()
      ..forward();
  },
  child: RotationTransition(
    turns: animation,
    child: Stack(
      children: [
        Positioned.fill(
          child: FlutterLogo(),
        ),
        Center(
          child: Text(
            'Click me!',
            style: TextStyle(
              fontSize: 60.0,
              fontWeight: FontWeight.bold,       ),             ),           ),         ],       ),      ),    ); }}
```

TASK 16.1

Develop a Fibonacci series app in Flutter.

RESOURCES:

https://flutter.dev/
https://hackernoon.com/9-popular-cross-platform-tools-for-app-development-in-2019-53765004761b
https://hackernoon.com/getting-started-with-cross-platform-app-development-in-2019-dd2bf7f6161b
https://ionicframework.com/
https://www.merixstudio.com/blog/cross-platform-mobile-development/