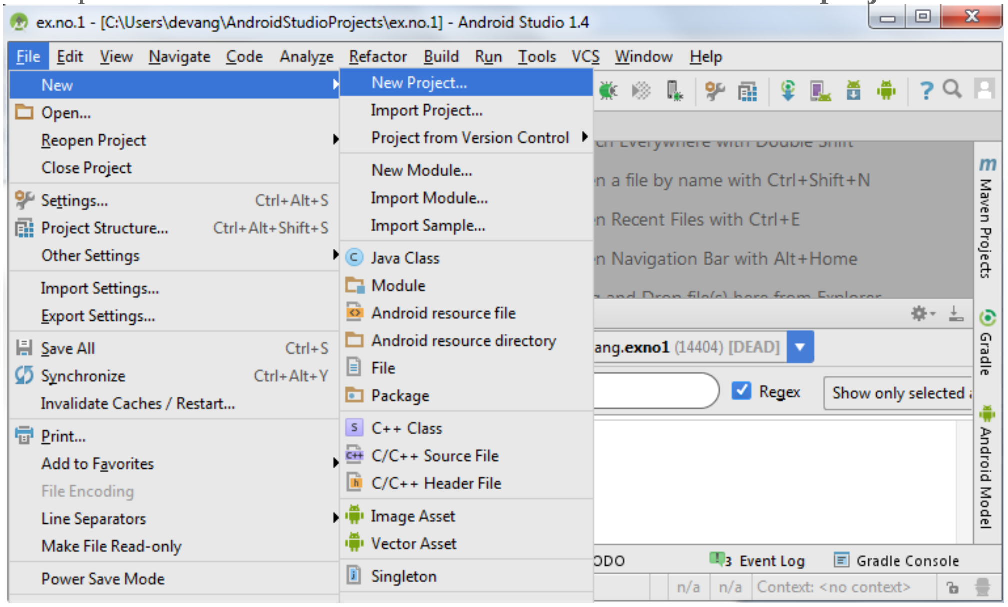
Android Application that uses GUI components, Font and Colors

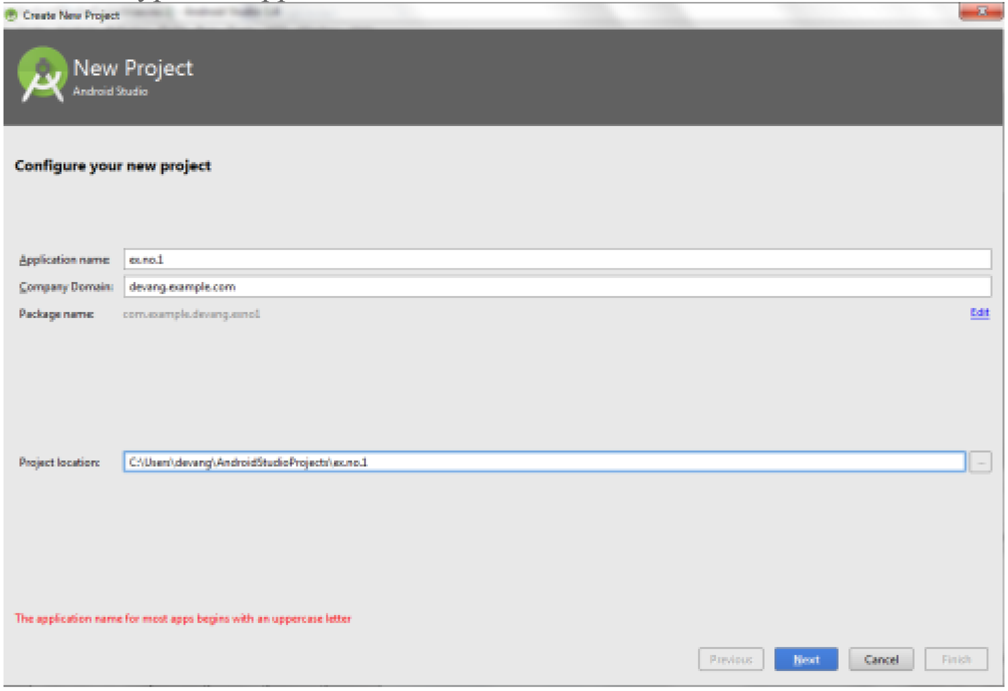
Procedure:

Creating a New project:

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



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



Configure your new project

Application name: ex.no.1

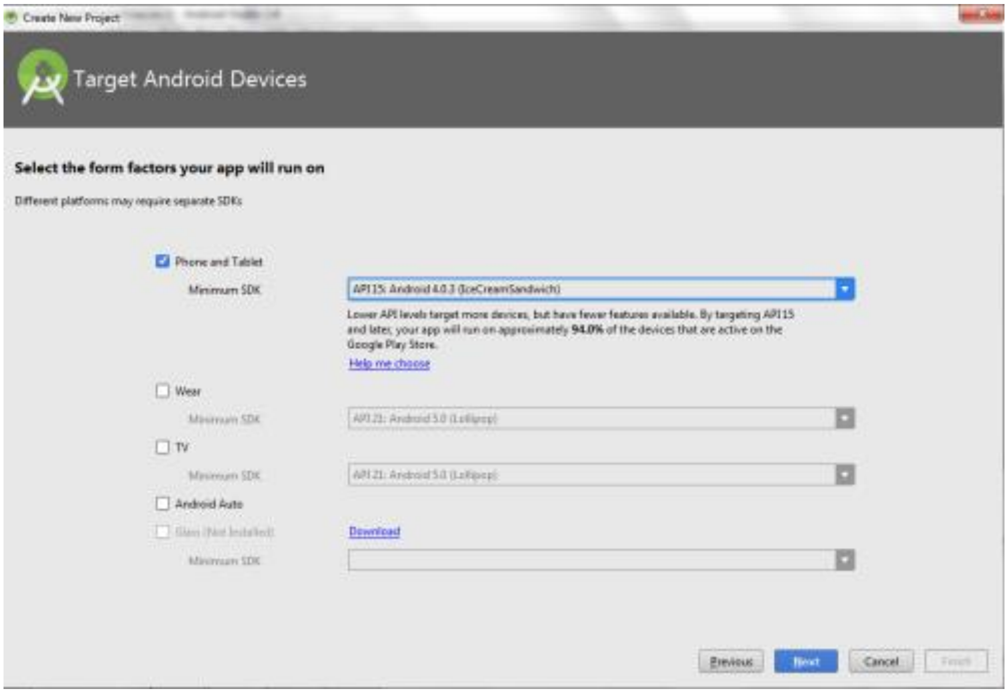
Company Domain: devang.example.com

Package name: com.example.devang.exno1 Edit

Project location: C:\Users\devang\AndroidStudioProjects\ex.no.1

The application name for most apps begins with an uppercase letter

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



Target Android Devices

Select the form factors your app will run on

Different platforms may require separate SDKs

☑ Phone and Tablet

Minimum SDK: API15: Android 4.0.3 (IceCreamSandwich)

Lower API levels target more devices, but have fewer features available. By targeting API15 and later, your app will run on approximately **94.0%** of the devices that are active on the Google Play Store.

☐ Wear

Minimum SDK: API 21: Android 5.0 (Lollipop)

☐ TV

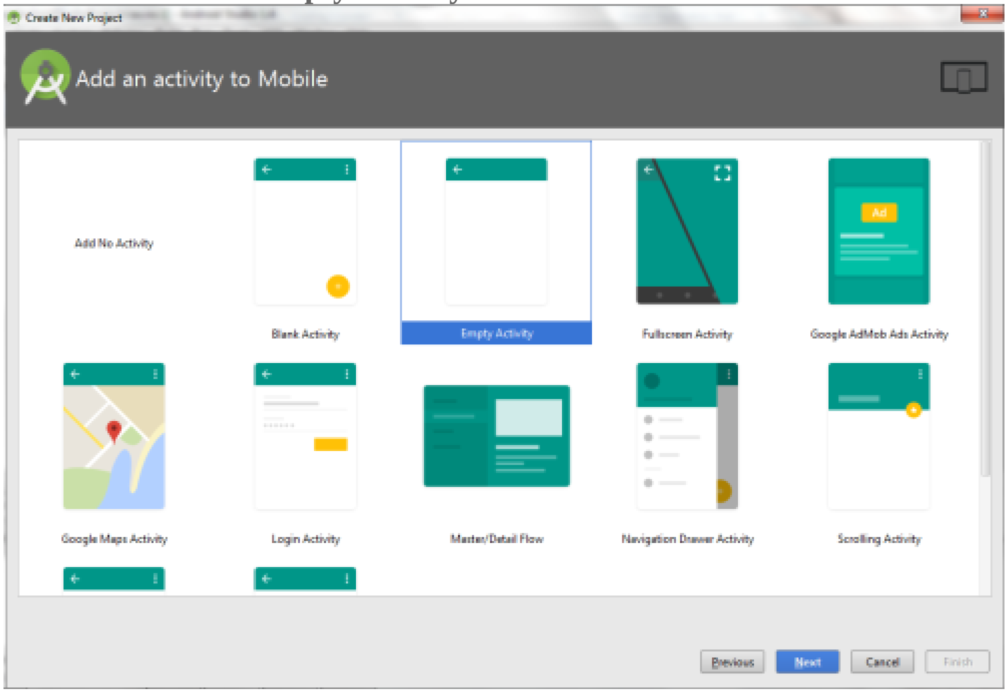
Minimum SDK: API 21: Android 5.0 (Lollipop)

☐ Android Auto

☐ Glass (Not Installed)

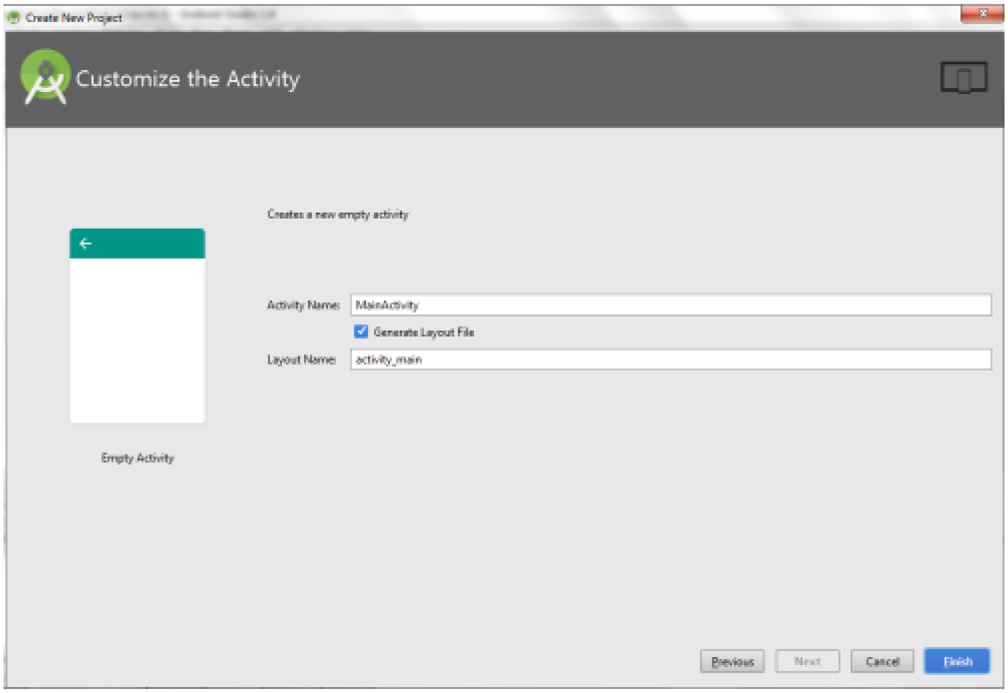
Minimum SDK:

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



Add an activity to Mobile

- Finally click F**inish**.



Create New Project

Customize the Activity

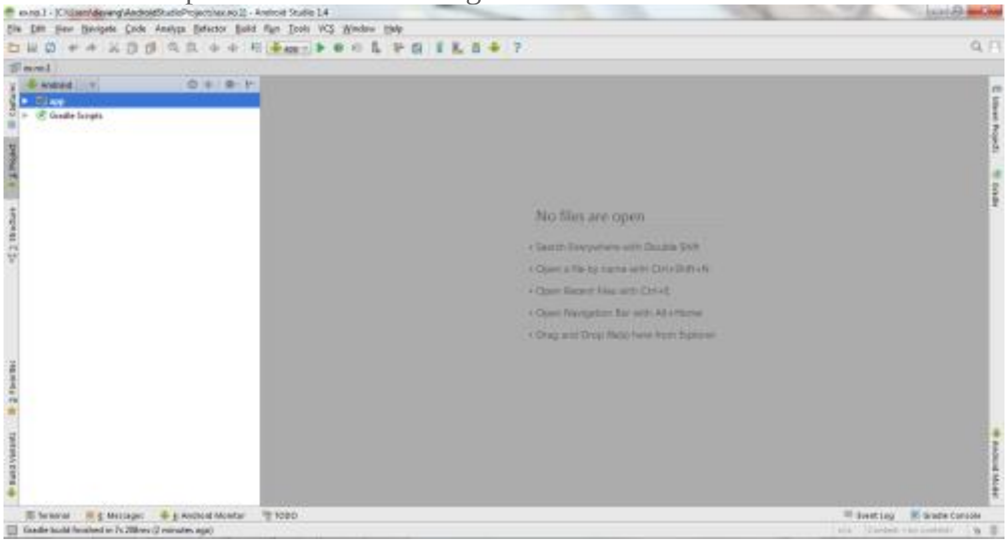
Creates a new empty activity

Activity Name: MainActivity

☑ Generate Layout File

Layout Name: activity_main

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

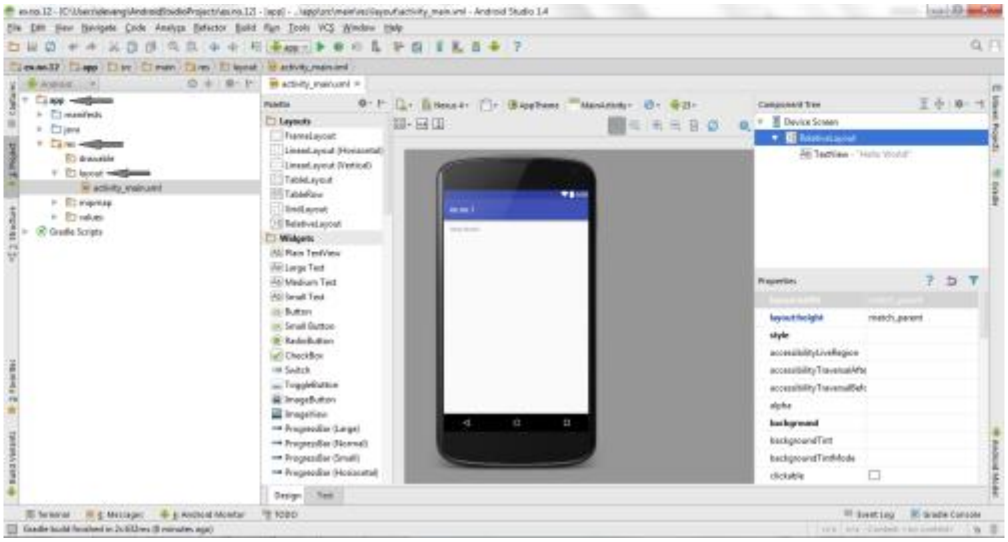


No files are open

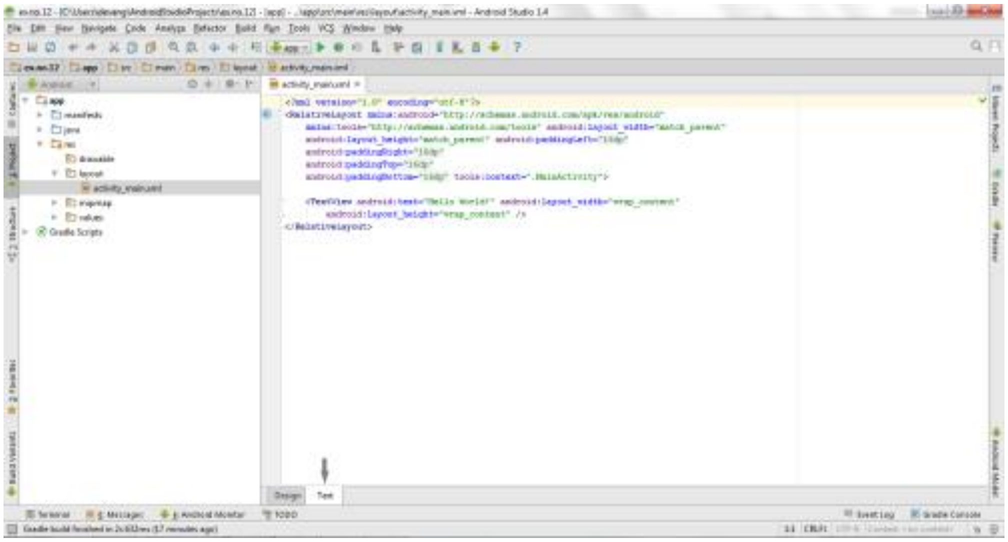
- Search Everywhere with Double Shift
- Open a file by name with Ctrl+Shift+N
- Open Recent Files with Ctrl+E
- Open Navigation Bar with Alt+Home
- Drag and Drop file(s) here from Explorer

Designing layout for the Android Application:

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



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



```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
    android:layout_height="match_parent" android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    android:paddingBottom="16dp" tools:context=".MainActivity">

    <TextView android:text="Hello World!" android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</RelativeLayout>
```

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

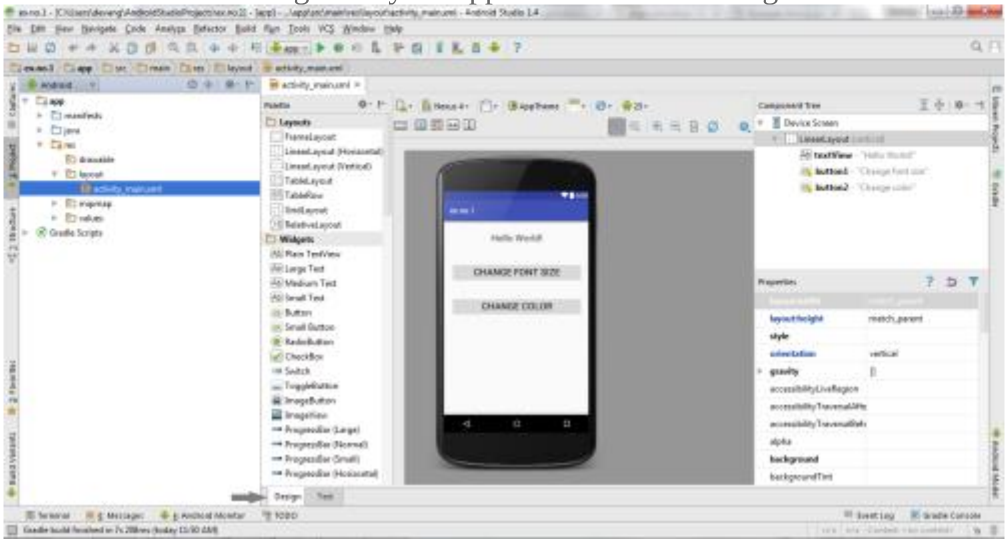
Code for Activity_main.xml:

?

```
<?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>
```

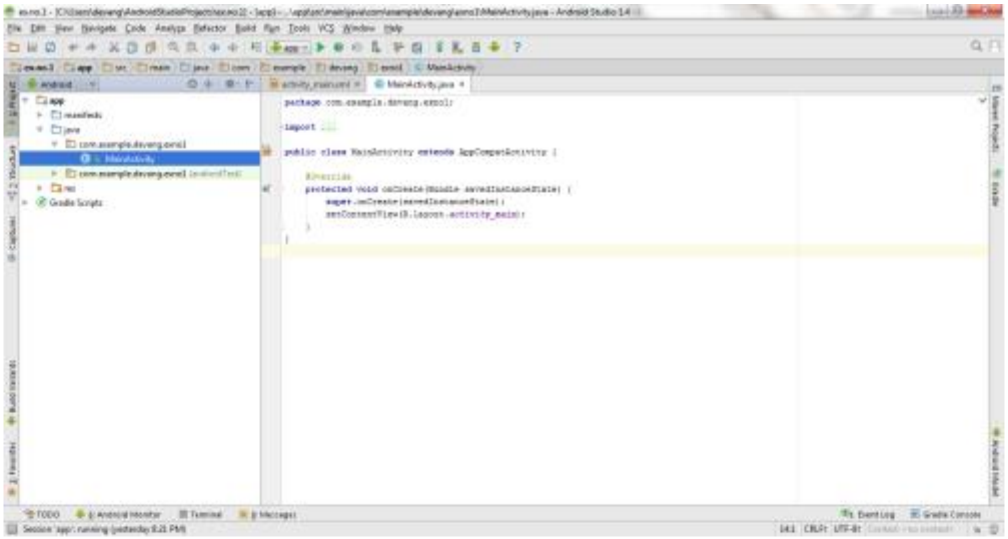
- Now click on Design and your application will look as given below.



- So now the designing part is completed.

Java Coding for the Android Application:

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



```
package com.example.devang.exno1;

import ...

public class MainActivity extends AppCompatActivity {

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

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

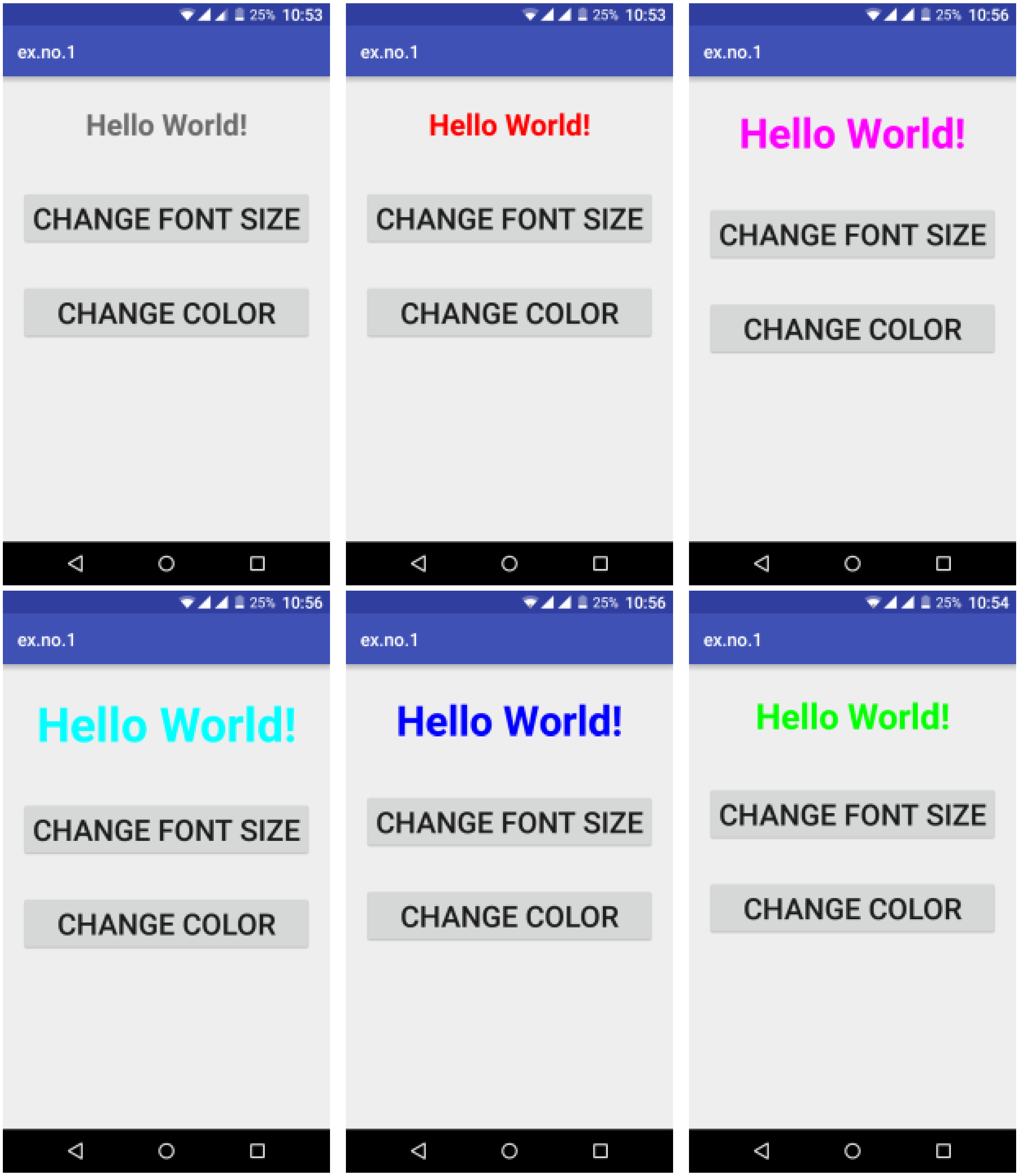
Code for MainActivity.java:

?

```
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.

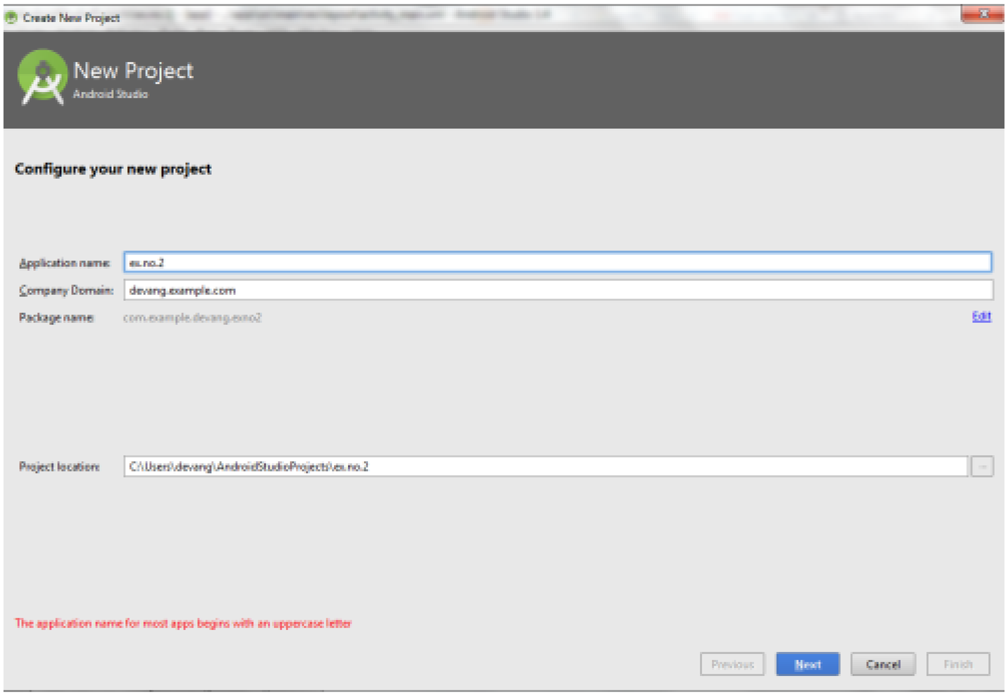
Android Application for Layout Managers and Event Listeners

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.**



Configure your new project

Application name: ex.no.2

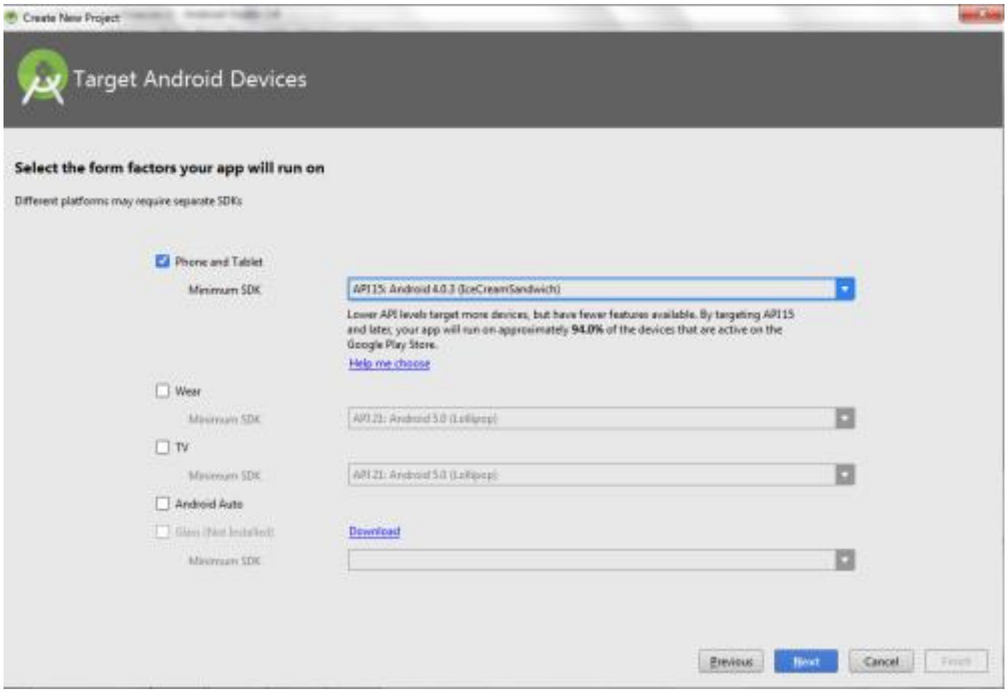
Company Domain: devang.example.com

Package name: com.example.devang.exno2

Project location: C:\Users\devang\AndroidStudioProjects\ex.no.2

The application name for most apps begins with an uppercase letter

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



Create New Project

Target Android Devices

Select the form factors your app will run on

Different platforms may require separate SDKs

☑ Phone and Tablet

Minimum SDK: API 15: Android 4.0.3 (IceCreamSandwich)

Lower API levels target more devices, but have fewer features available. By targeting API 15 and later, your app will run on approximately **94.0%** of the devices that are active on the Google Play Store.

Help me choose

□ Wear

Minimum SDK: API 21: Android 5.0 (Lollipop)

□ TV

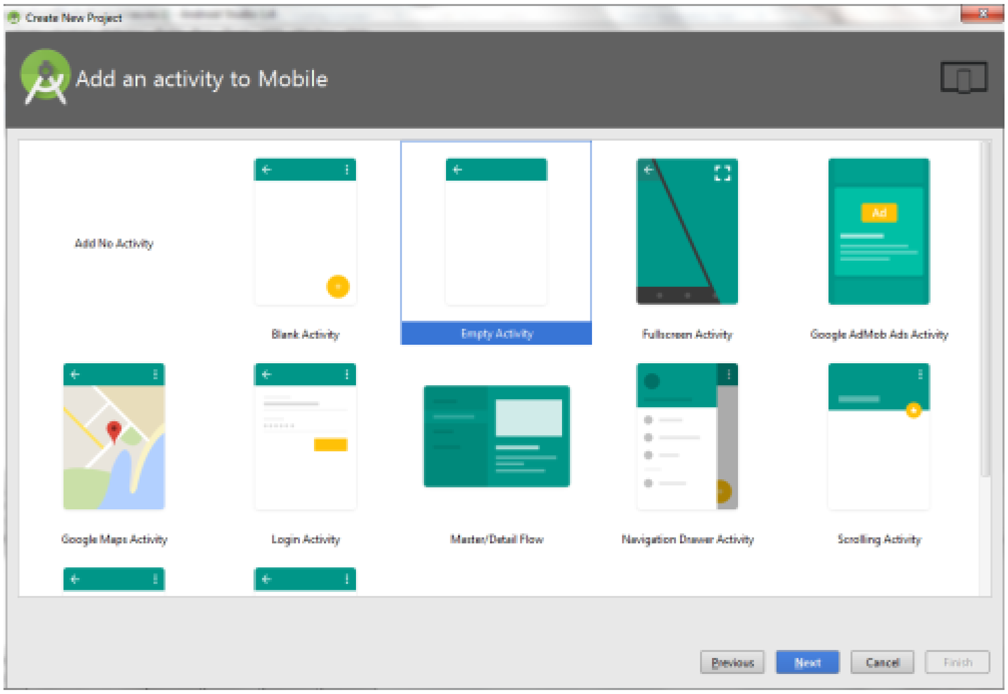
Minimum SDK: API 21: Android 5.0 (Lollipop)

□ Android Auto

□ Glass (Not Installed) Download

Minimum SDK

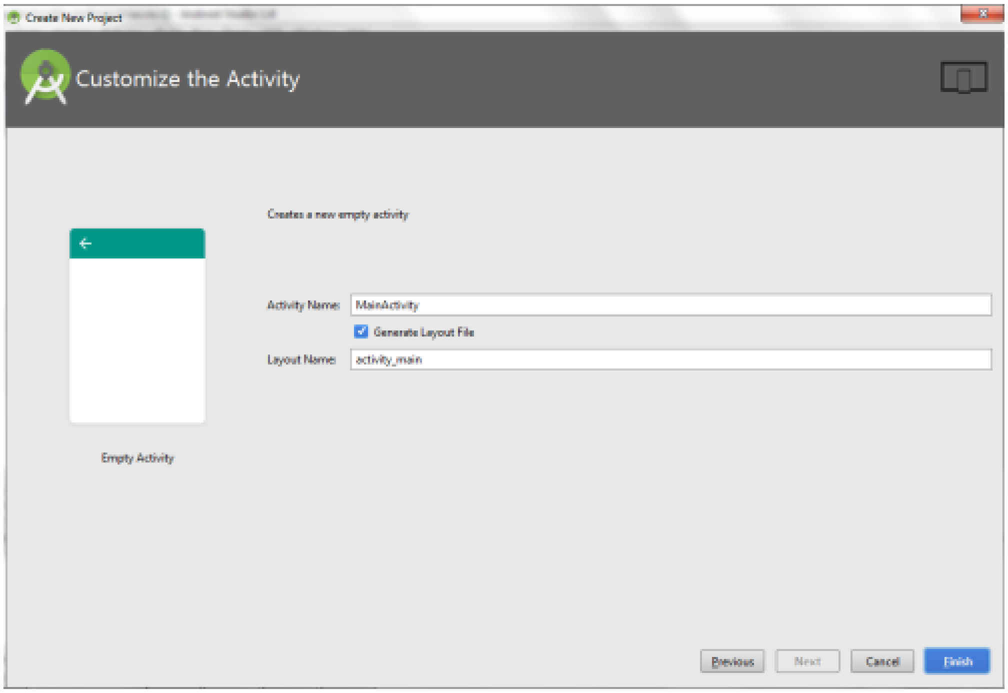
- Then select the **Empty Activity** and click **Next.**



Create New Project

Add an activity to Mobile

- Finally click F**inish**.



Creates a new empty activity

Activity Name: MainActivity

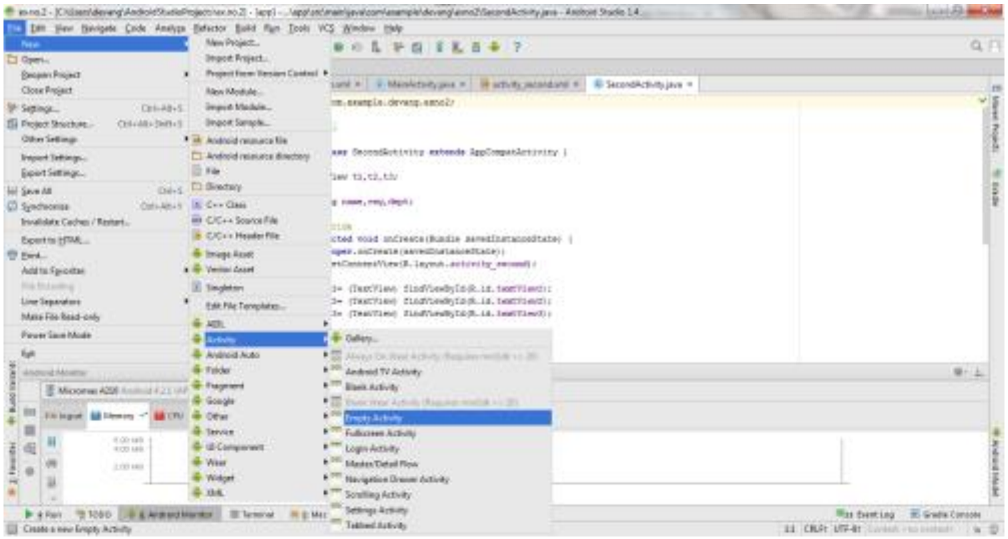
☑ Generate Layout File

Layout Name: activity_main

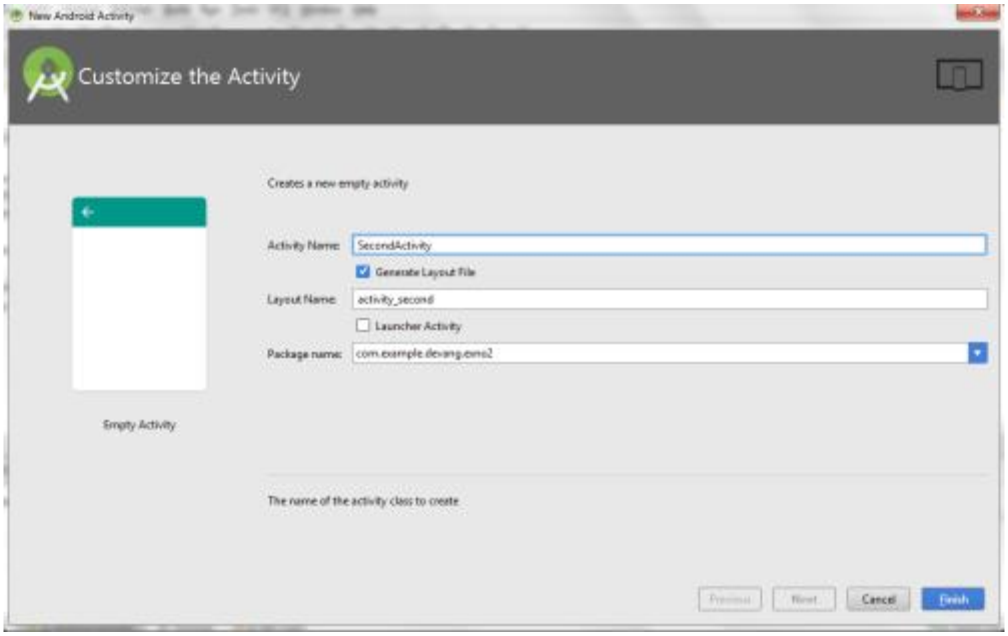
- It will take some time to build and load the project.
- After completion it will look as given below.

Creating Second Activity for the Android Application:

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



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



Creates a new empty activity

Activity Name: SecondActivity

☑ Generate Layout File

Layout Name: activity_second

☐ Launcher Activity

Package name: com.example.devang.exno2

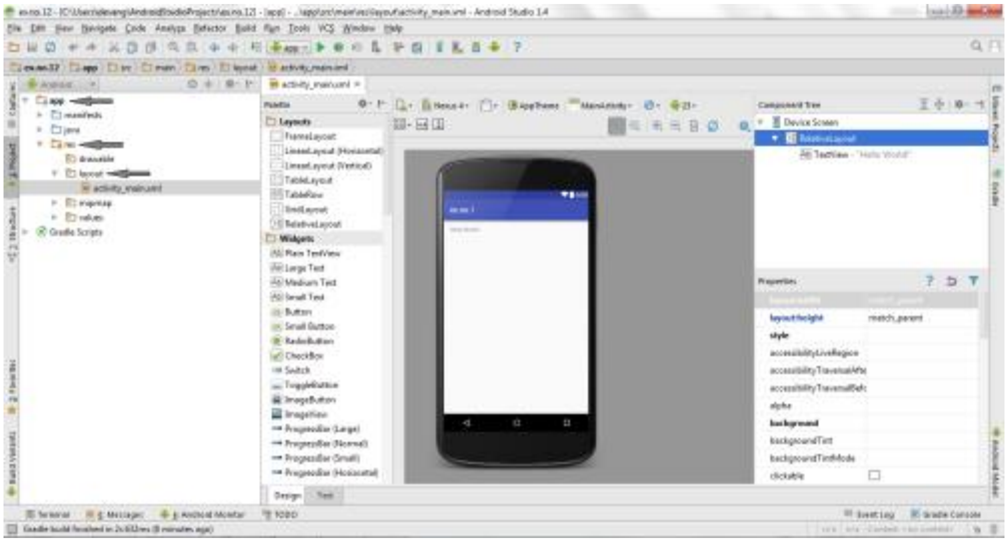
The name of the activity class to create

- Thus Second Activity For the application is created.

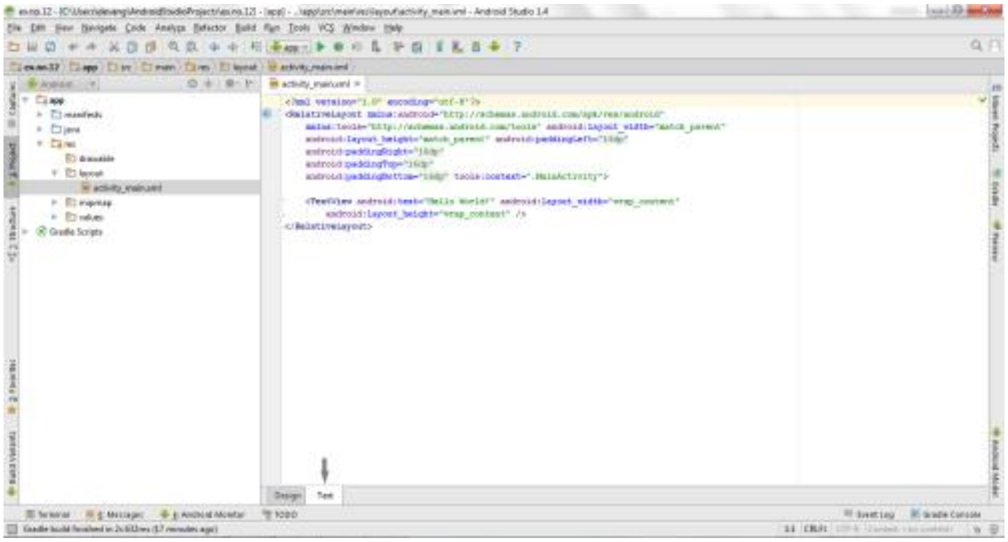
Designing layout for the Android Application:

Designing Layout for Main Activity:

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



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



```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
    android:layout_height="match_parent" android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    android:paddingBottom="16dp" tools:context=".MainActivity">

    <TextView android:text="Hello World!" android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</RelativeLayout>
```

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

Code for Activity_main.xml:

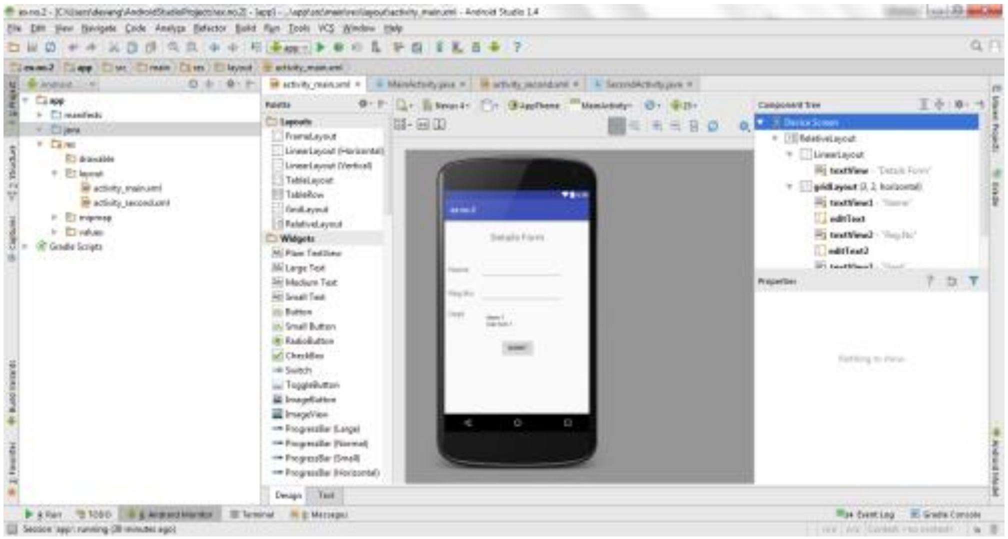
?

```
<?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>
```

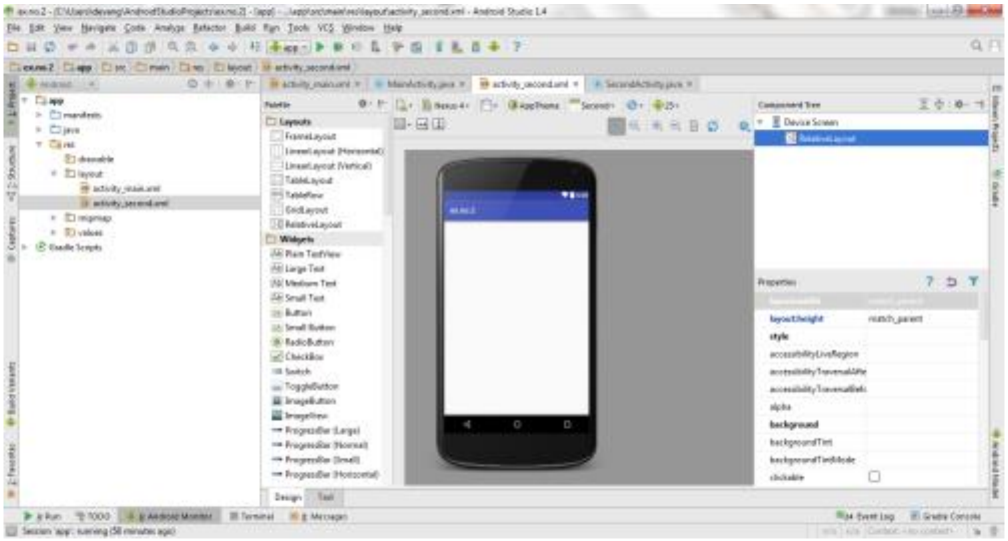
- Now click on Design and your activity will look as given below.



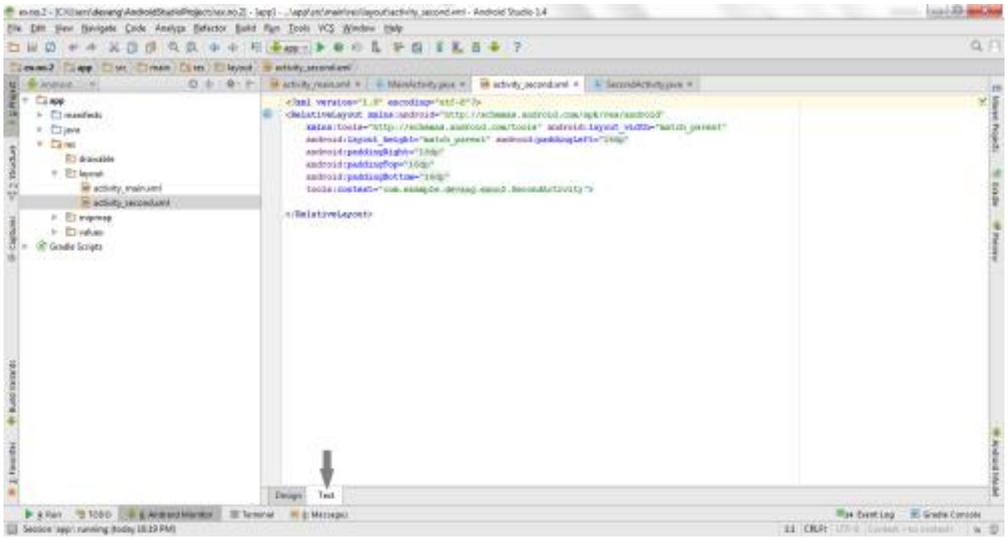
- So now the designing part of Main Activity is completed.

Designing Layout for Second Activity:

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



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



```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
    android:layout_height="match_parent" android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    android:paddingBottom="16dp"
    tools:context="com.example.devang.exno2.SecondActivity">

</RelativeLayout>
```

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

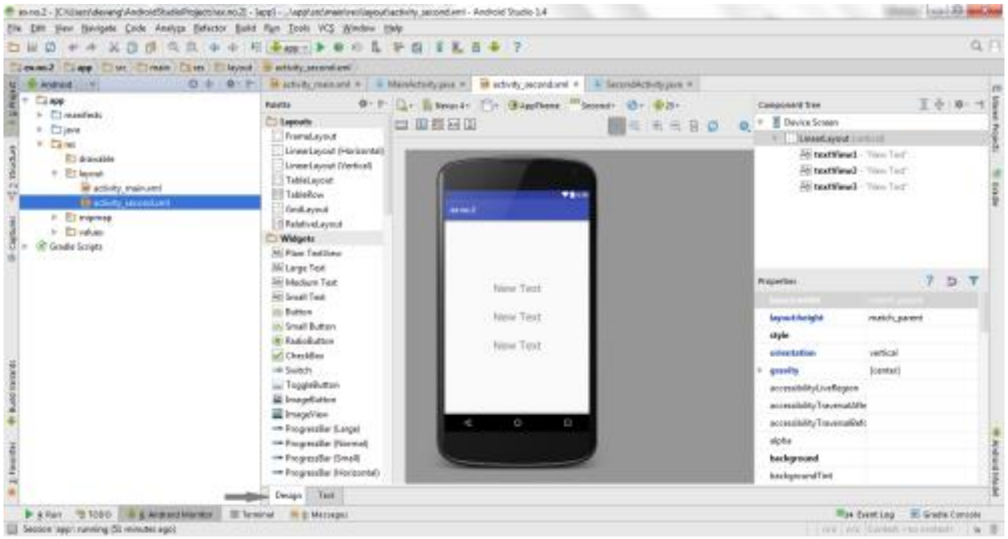
Code for Activity_second.xml:

?

```
<?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.



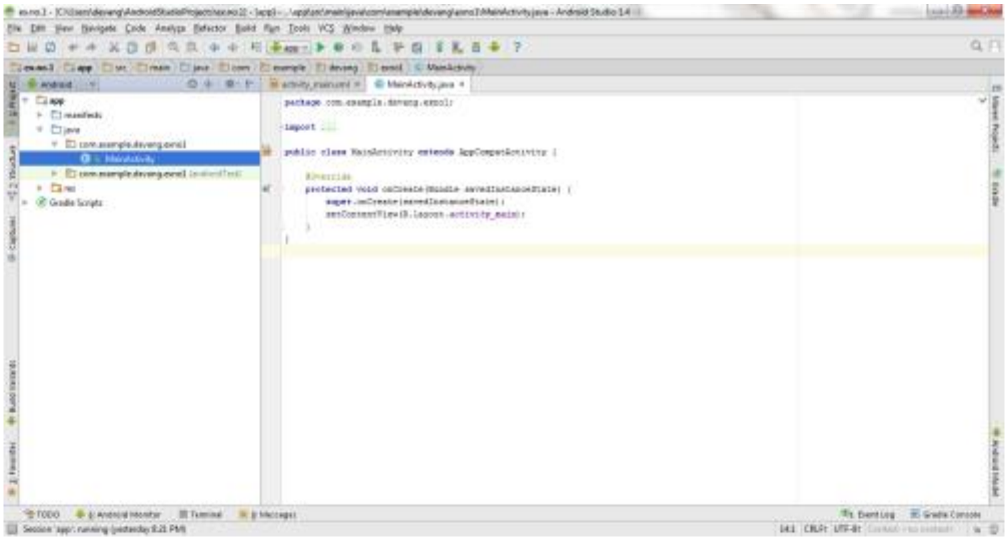
layout:height match_parent
style
orientation vertical
gravity [center]

- 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.**



```
package com.example.devang.exno1;

import ...

public class MainActivity extends AppCompatActivity {

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

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

Code for MainActivity.java:

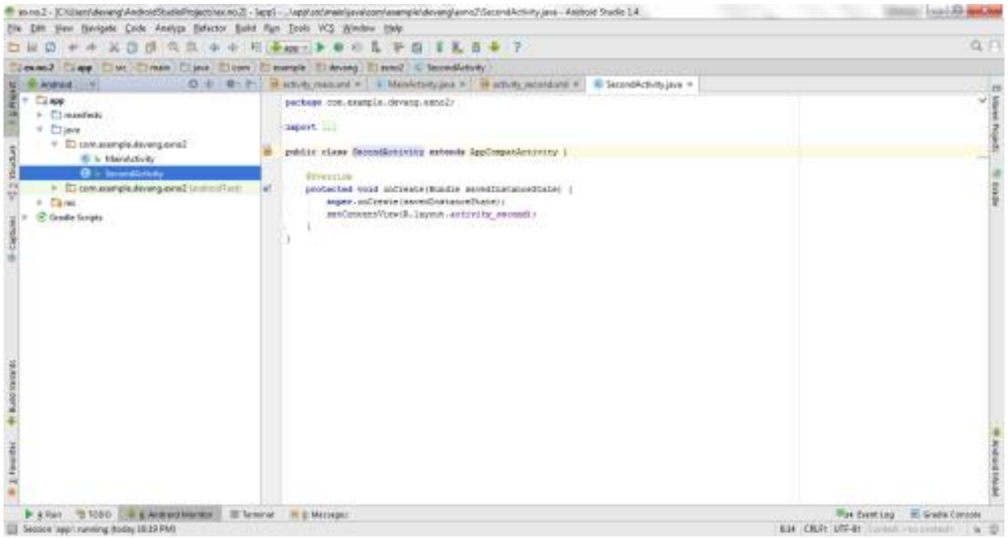
?

```
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_a
        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.**



```
package com.example.devang.exno2;

import ...

public class SecondActivity extends AppCompatActivity {

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

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

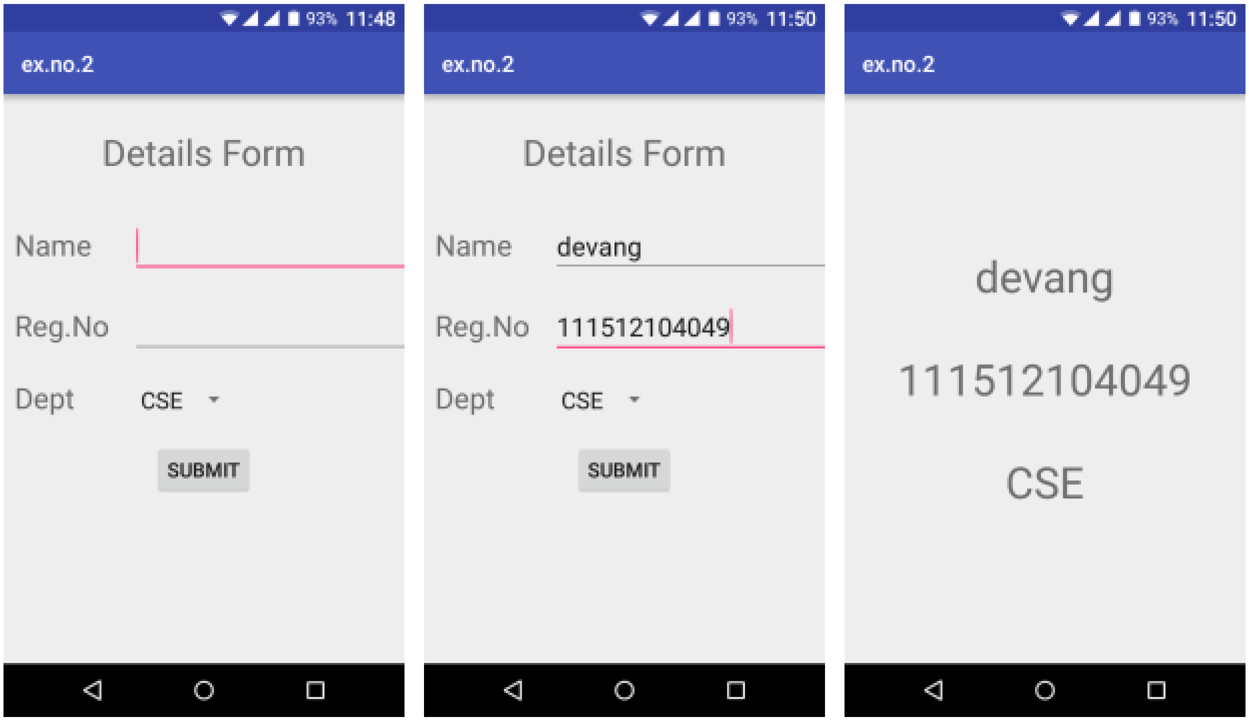
Code for SecondActivity.java:

?

```
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:

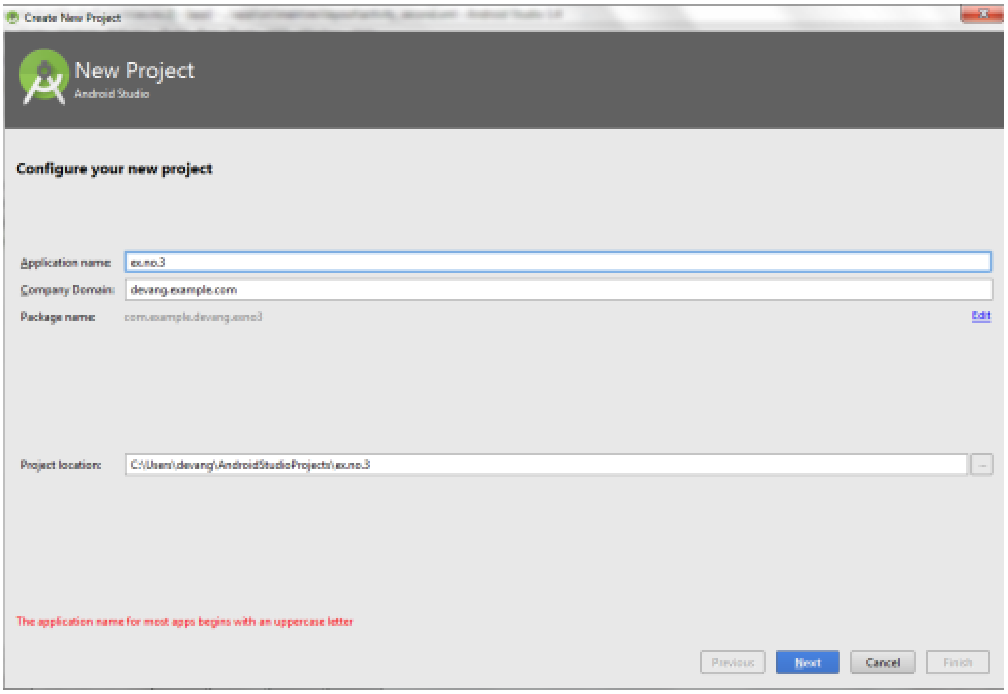
Thus a Simple Android Application that uses Layout Managers and Event Listeners is developed and executed successfully.

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.**



Configure your new project

Application name: ex.no.3

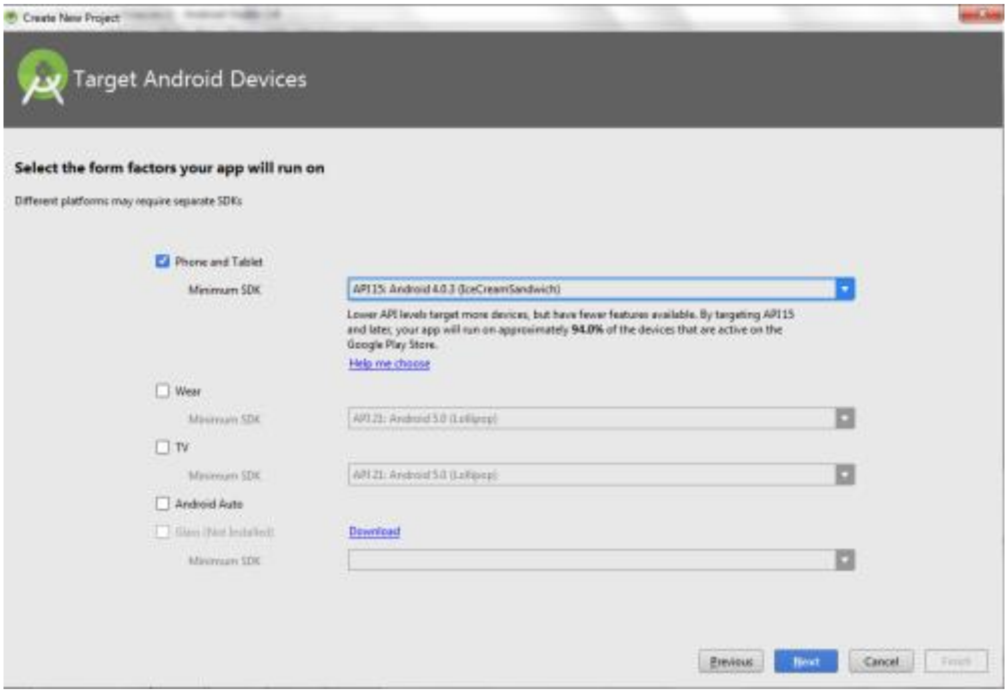
Company Domain: devang.example.com

Package name: com.example.devang.exno3

Project location: C:\Users\devang\AndroidStudioProjects\ex.no.3

The application name for most apps begins with an uppercase letter

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



Create New Project

Target Android Devices

Select the form factors your app will run on

Different platforms may require separate SDKs

☑ Phone and Tablet

Minimum SDK: API 15: Android 4.0.3 (IceCreamSandwich)

Lower API levels target more devices, but have fewer features available. By targeting API 15 and later, your app will run on approximately **94.0%** of the devices that are active on the Google Play Store.

Help me choose

☐ Wear

Minimum SDK: API 21: Android 5.0 (Lollipop)

☐ TV

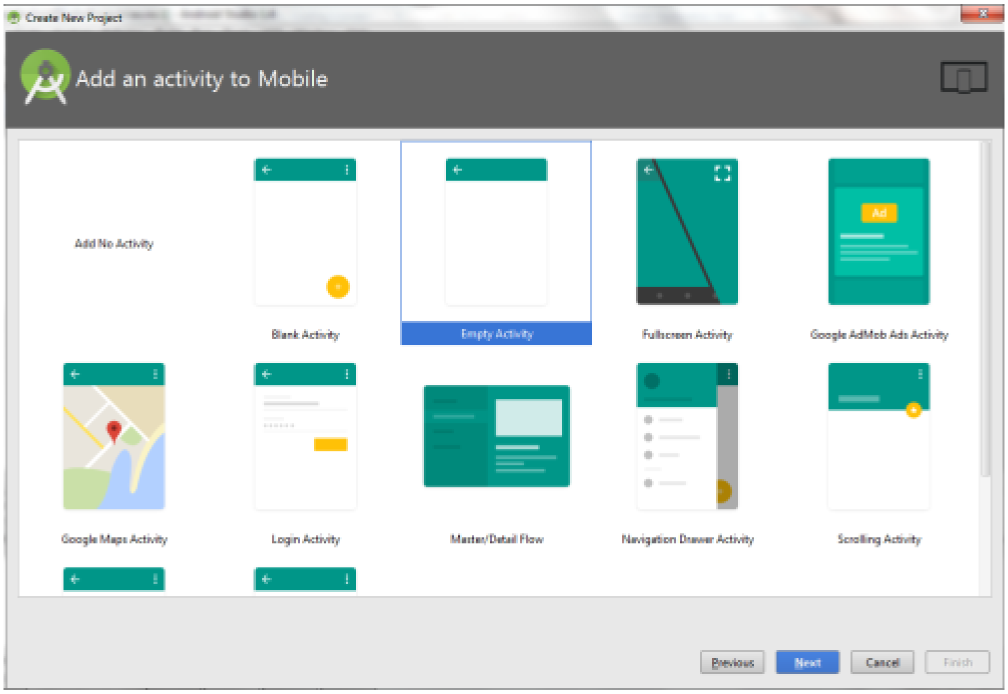
Minimum SDK: API 21: Android 5.0 (Lollipop)

☐ Android Auto

☐ Glass (Not Installed) Download

Minimum SDK

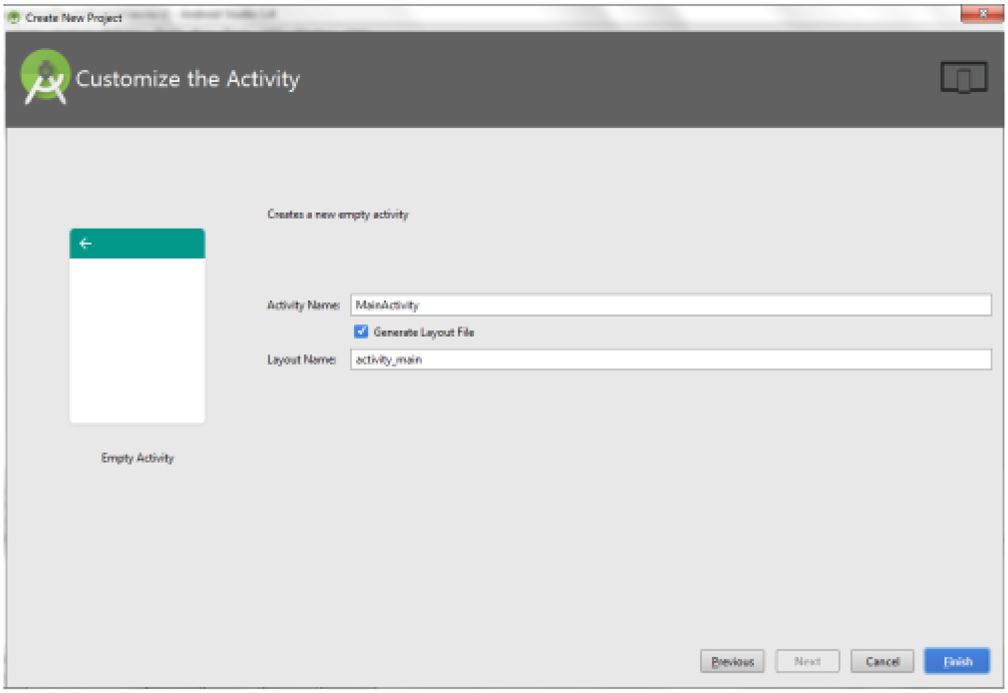
- Then select the **Empty Activity** and click **Next.**



Create New Project

Add an activity to Mobile

- Finally click F**inish**.



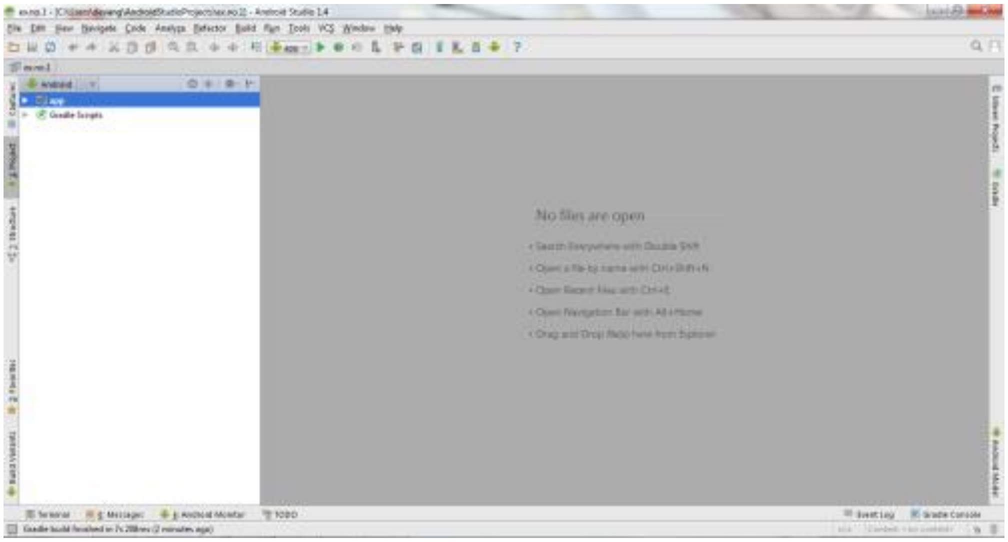
Creates a new empty activity

Activity Name: MainActivity

☑ Generate Layout File

Layout Name: activity_main

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

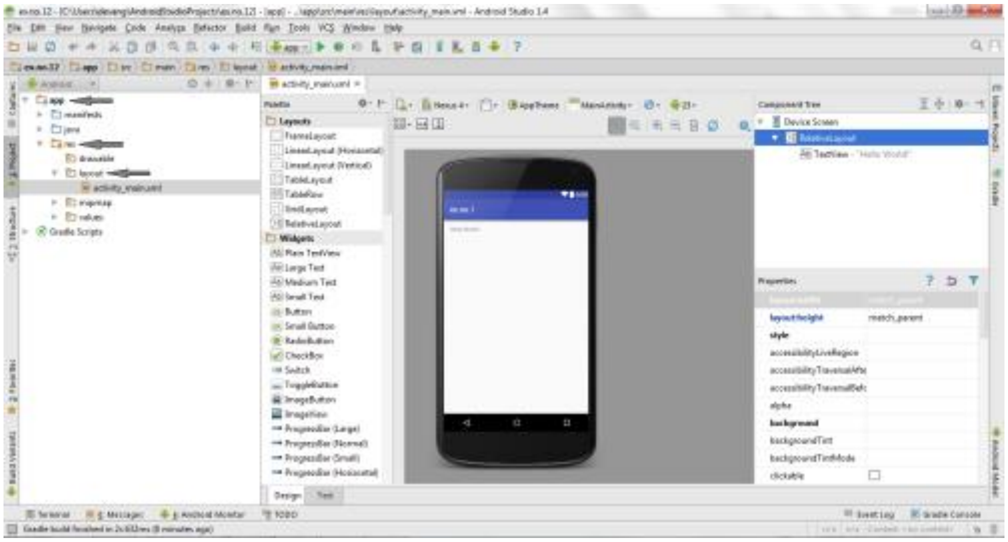


No files are open

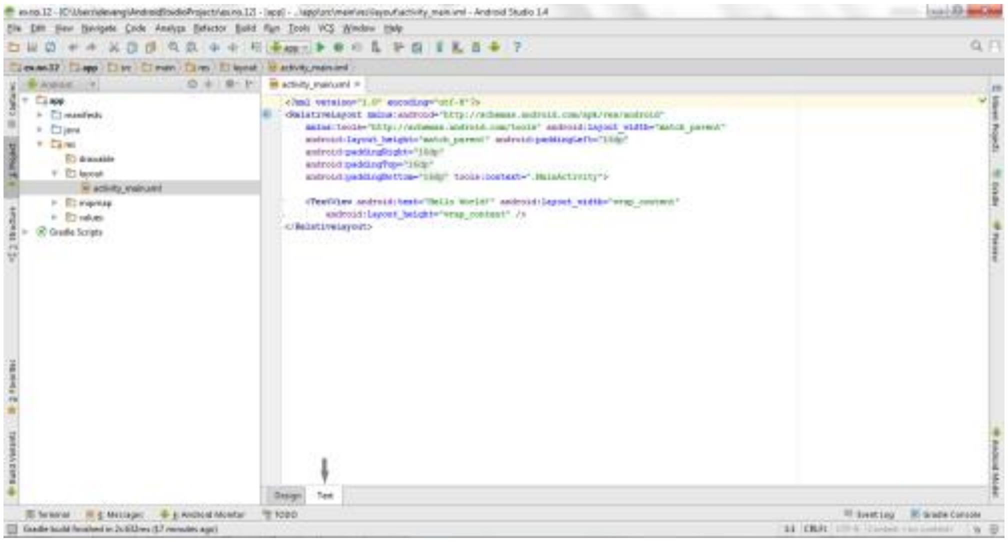
- Search Everywhere with Double Shift
- Open a file by name with Ctrl+Shift+N
- Open Recent Files with Ctrl+E
- Open Navigation Bar with Alt+Home
- Drag and Drop file(s) here from Explorer

Designing layout for the Android Application:

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



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



```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
    android:layout_height="match_parent" android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    android:paddingBottom="16dp" tools:context=".MainActivity">

    <TextView android:text="Hello World!" android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</RelativeLayout>
```

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

Code for Activity_main.xml:

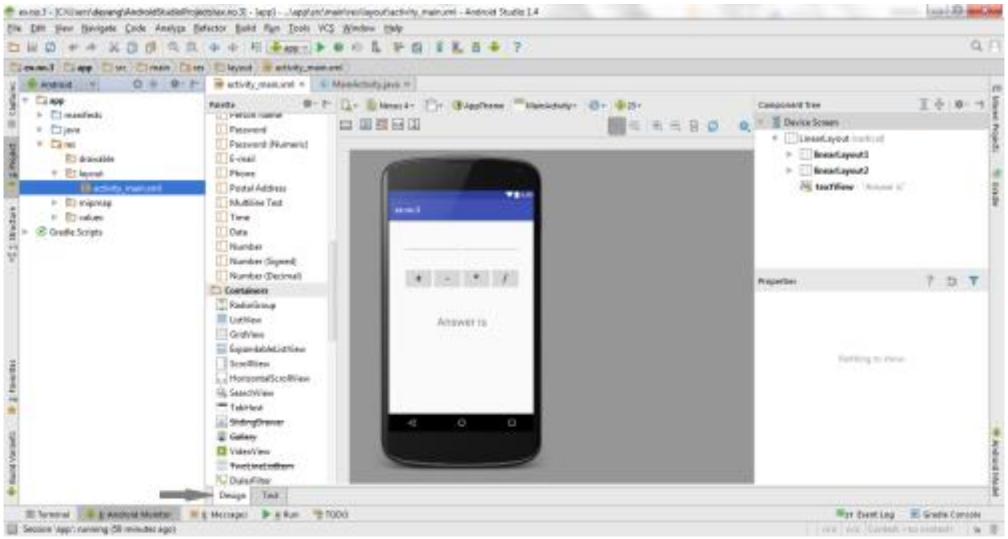
?

```
<?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>
```

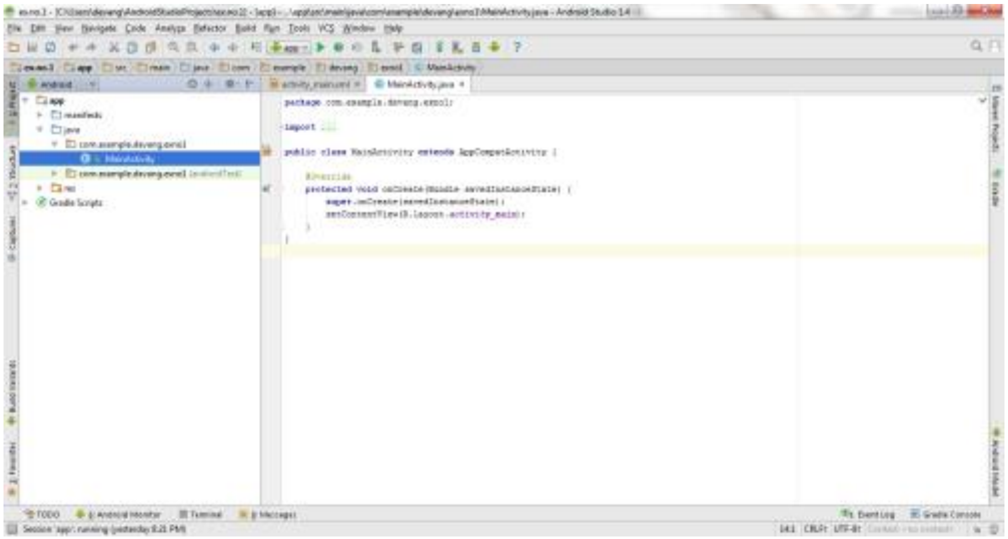
- Now click on Design and your application will look as given below.



- So now the designing part is completed.

Java Coding for the Android Application:

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



```
package com.example.devang.exno1;

import ...

public class MainActivity extends AppCompatActivity {

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

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

Code for MainActivity.java:

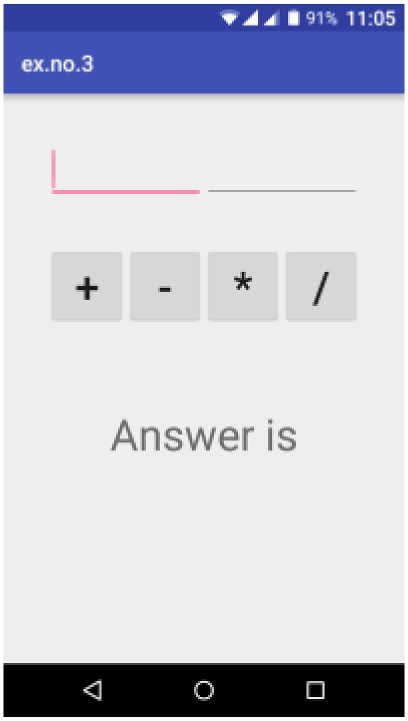
?

```
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);
    }
}
```

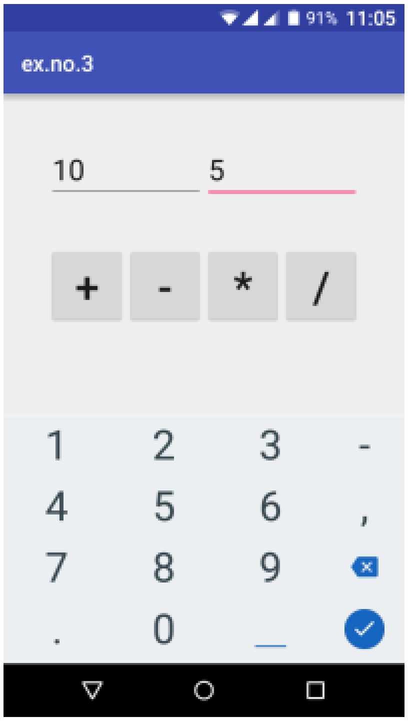
- So now the Coding part is also completed.

- Now run the application to see the output.

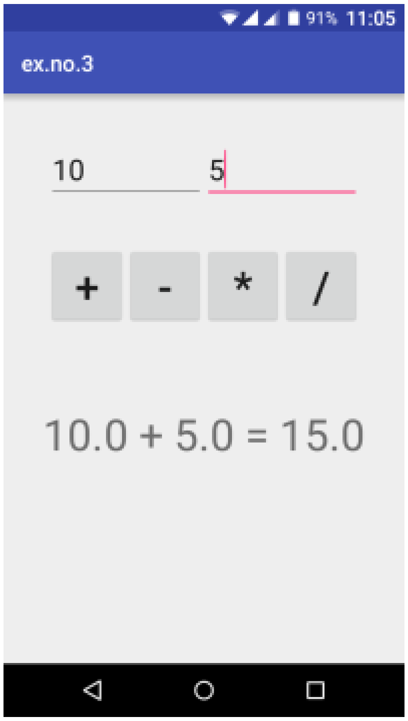
Output:



Answer is

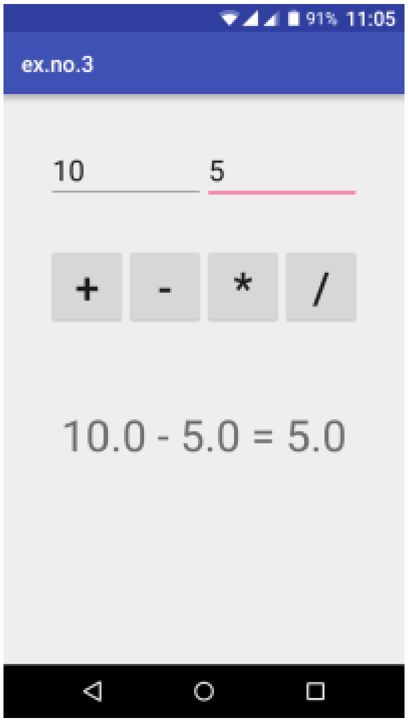


10 5



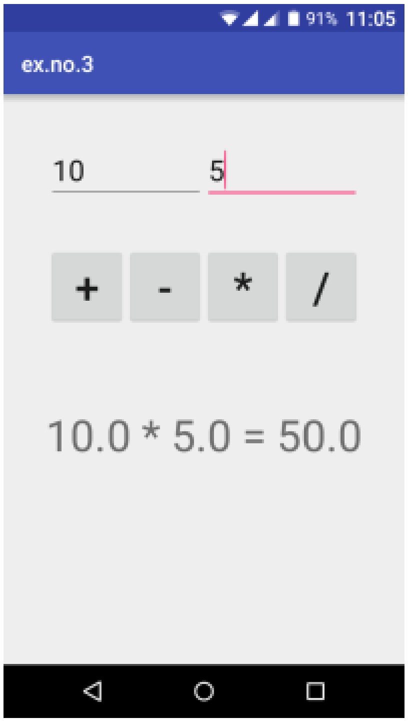
10 5

10.0 + 5.0 = 15.0



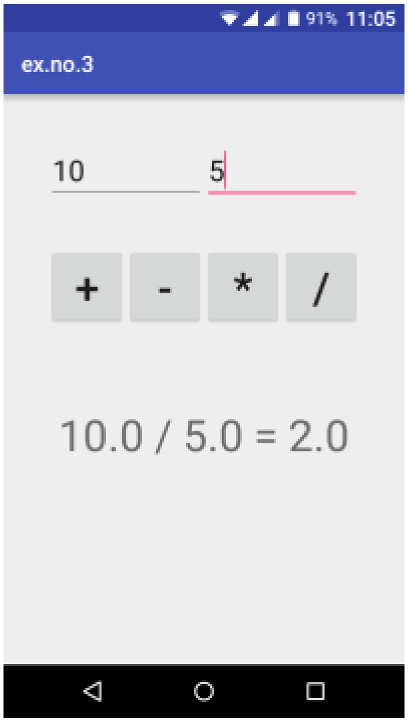
10 5

10.0 - 5.0 = 5.0



10 5

10.0 * 5.0 = 50.0



10 5

10.0 / 5.0 = 2.0

Result:

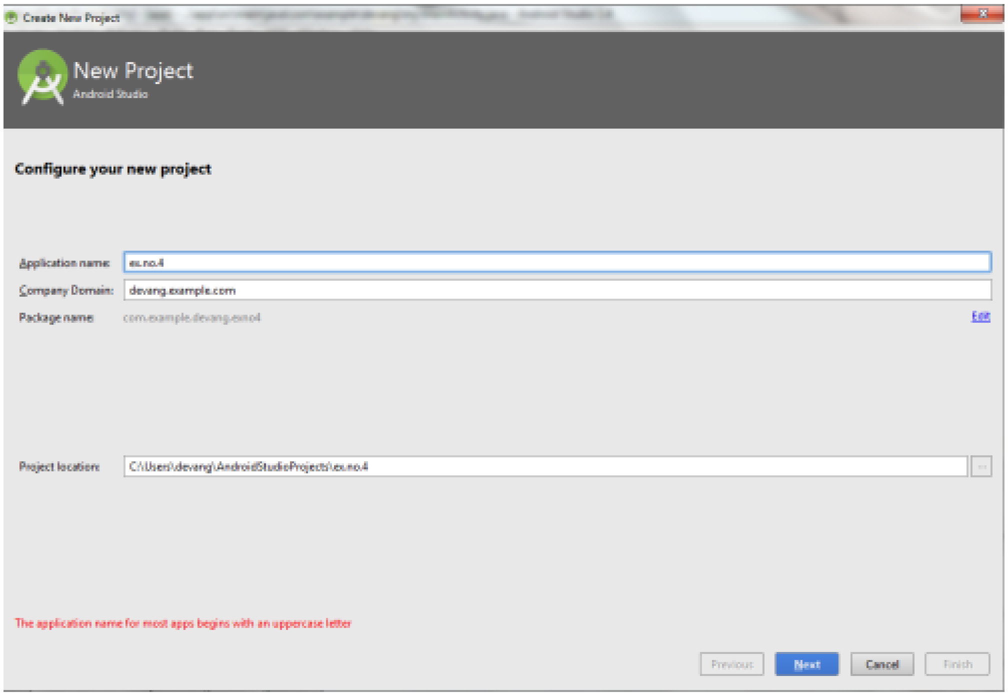
Thus a Simple Android Application for Native Calculator is developed and executed successfully.

Android Application to draw Basic Graphical Primitives

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.**



Configure your new project

Application name: ex.no.4

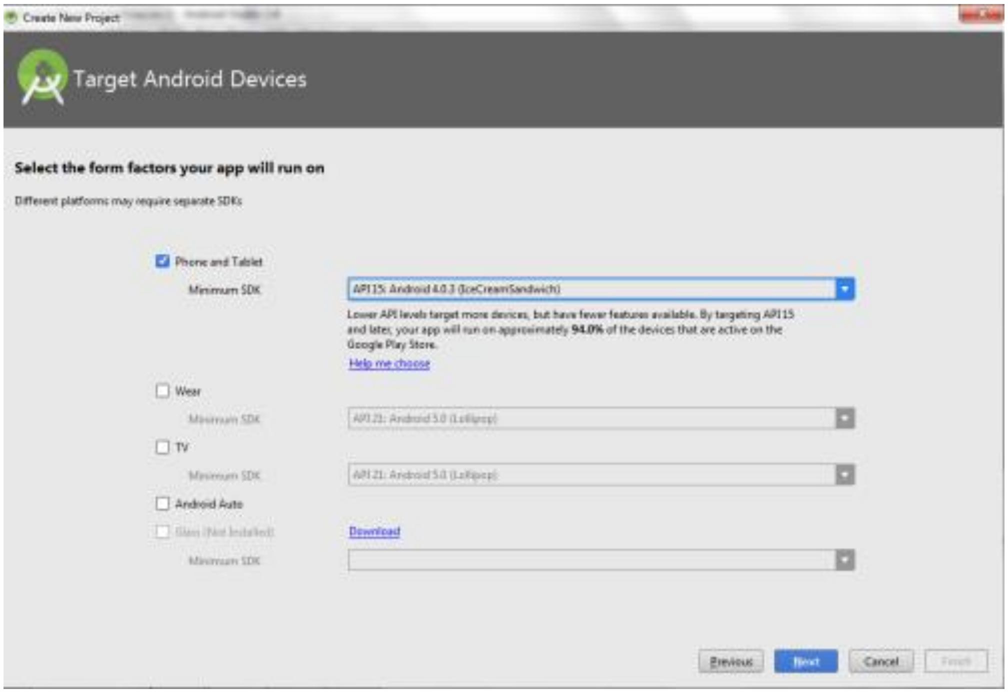
Company Domain: devang.example.com

Package name: com.example.devang.exno4

Project location: C:\Users\devang\AndroidStudioProjects\ex.no.4

The application name for most apps begins with an uppercase letter

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



Target Android Devices

Select the form factors your app will run on

Different platforms may require separate SDKs

☑ Phone and Tablet

Minimum SDK: API 15: Android 4.0.3 (IceCreamSandwich)

Lower API levels target more devices, but have fewer features available. By targeting API 15 and later, your app will run on approximately **94.0%** of the devices that are active on the Google Play Store.

Help me choose

□ Wear

Minimum SDK: API 21: Android 5.0 (Lollipop)

□ TV

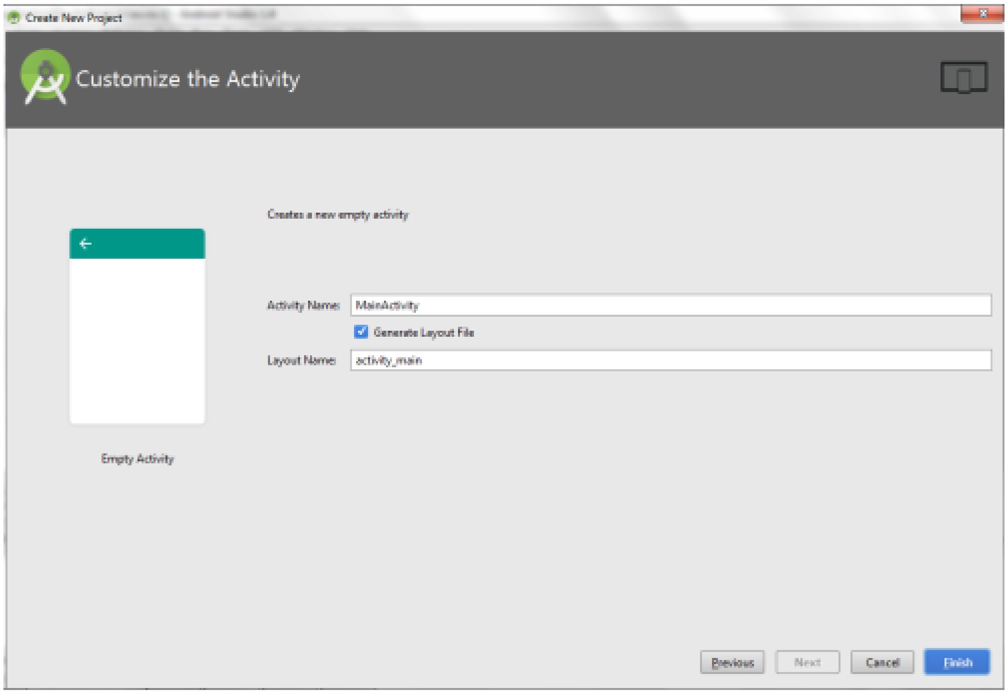
Minimum SDK: API 21: Android 5.0 (Lollipop)

□ Android Auto

□ Glass (Not Installed) Download

Minimum SDK

- Then select the **Empty Activity** and click **Next.**
- Finally click F**inish**.



Customize the Activity

Creates a new empty activity

Activity Name: MainActivity

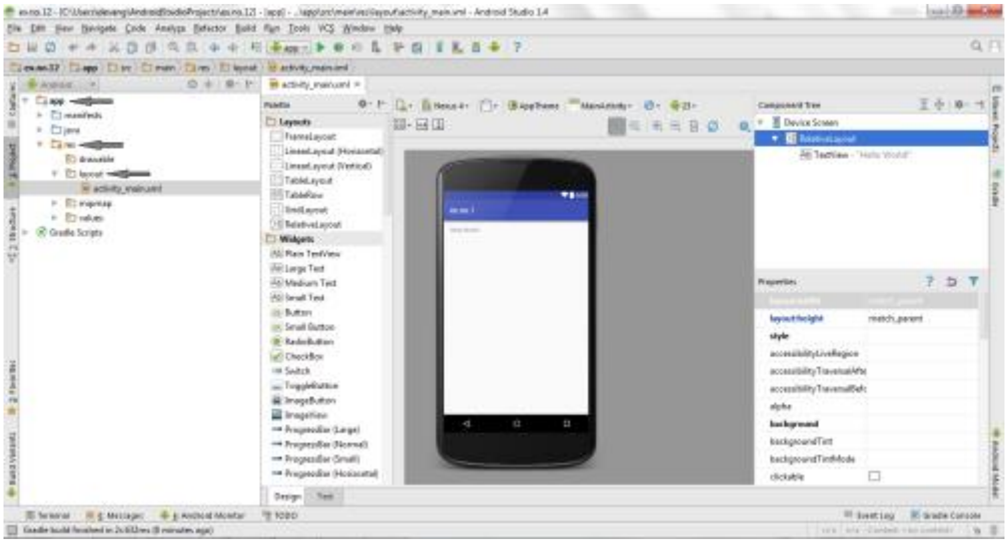
☑ Generate Layout File

Layout Name: activity_main

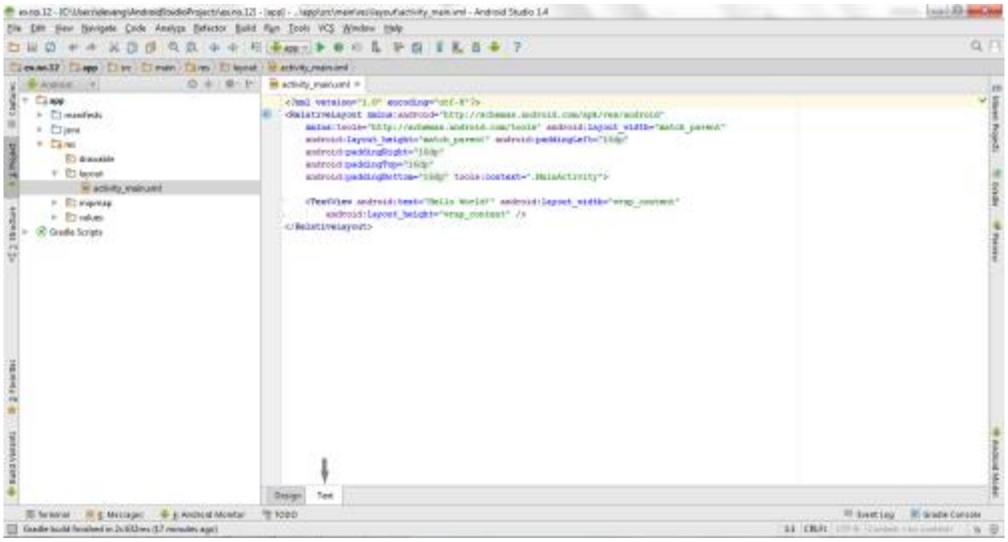
- It will take some time to build and load the project.
- After completion it will look as given below.

Designing layout for the Android Application:

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



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


ex.no.12 - [C:\Users\...\AndroidStudioProjects\ex.no.12] - [app] - ...\app\src\main\res\layout\activity_main.xml - Android Studio 1.4
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
activity_main.xml
Design
Text
Terminal
Messages
Android Monitor
TODO
Event Log
Gradle Console
Gradle build finished in 2s 632ms (17 minutes ago)

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
    android:layout_height="match_parent" android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    android:paddingBottom="16dp" tools:context=".MainActivity">

    <TextView android:text="Hello World!" android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</RelativeLayout>
```

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

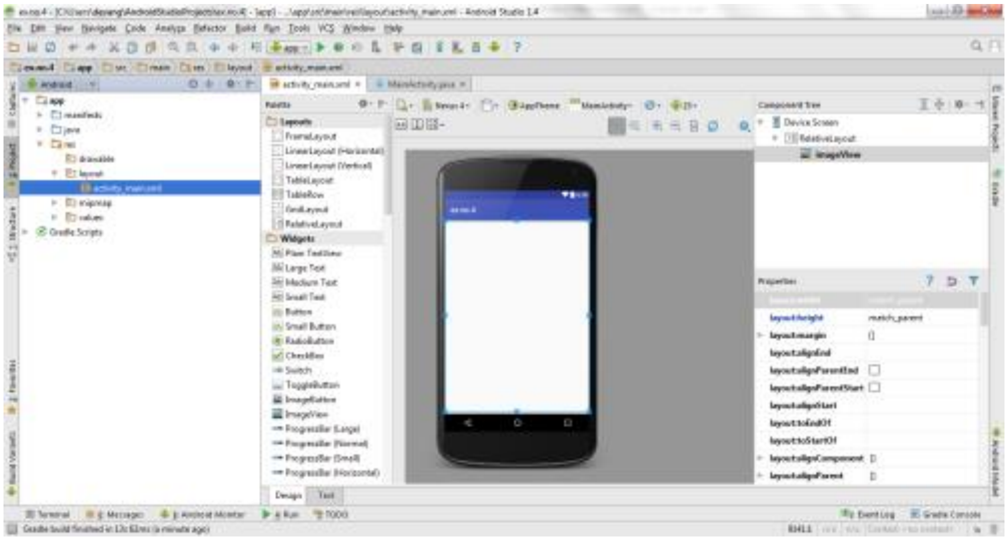
Code for Activity_main.xml:

?

```
<?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>
```

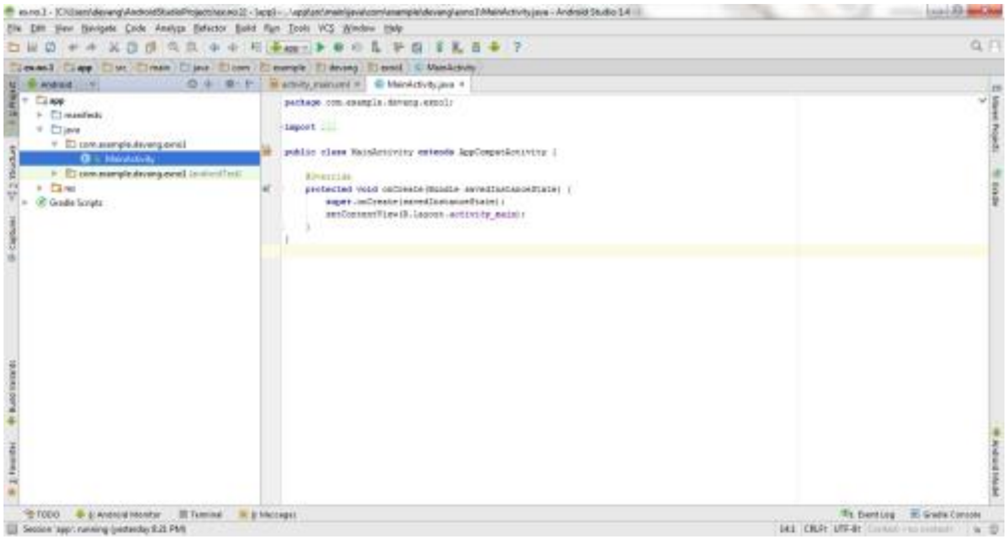
- Now click on **Design** and your application will look as given below.



- So now the designing part is completed.

Java Coding for the Android Application:

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



```
package com.example.devang.exno1;

import ...

public class MainActivity extends AppCompatActivity {

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

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

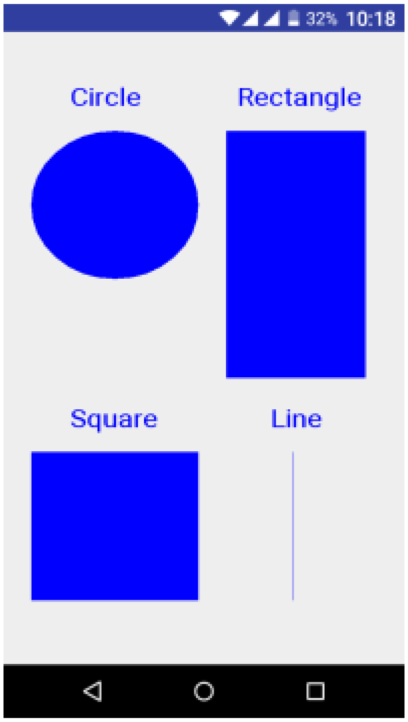
Code for MainActivity.java:

```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.

Simple Android Application that makes use of Database

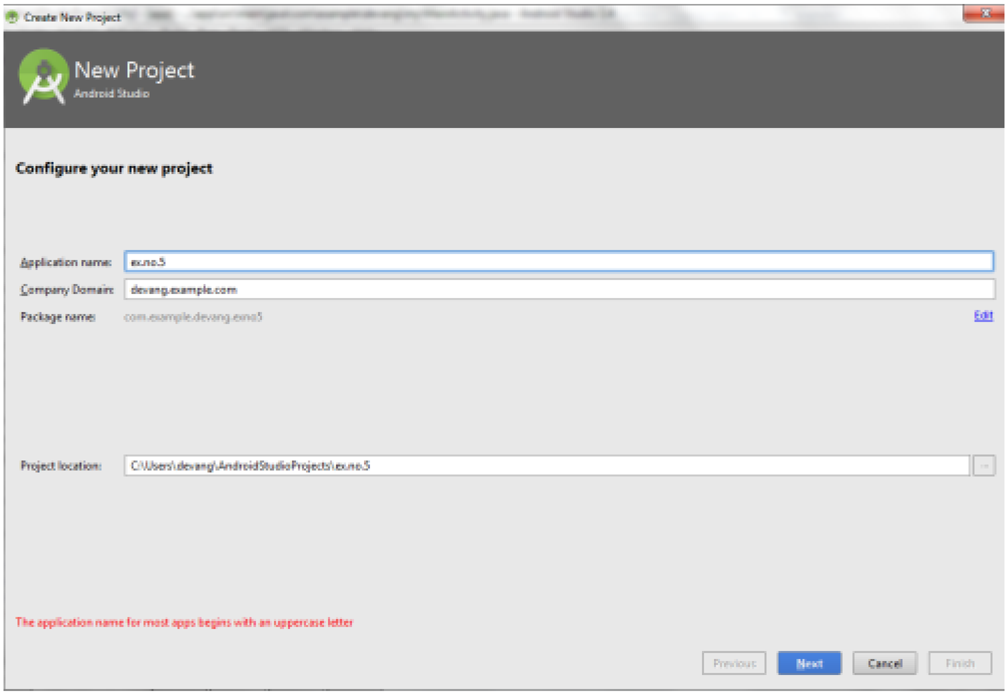
Aim:

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.**



Configure your new project

Application name: ex.no.5

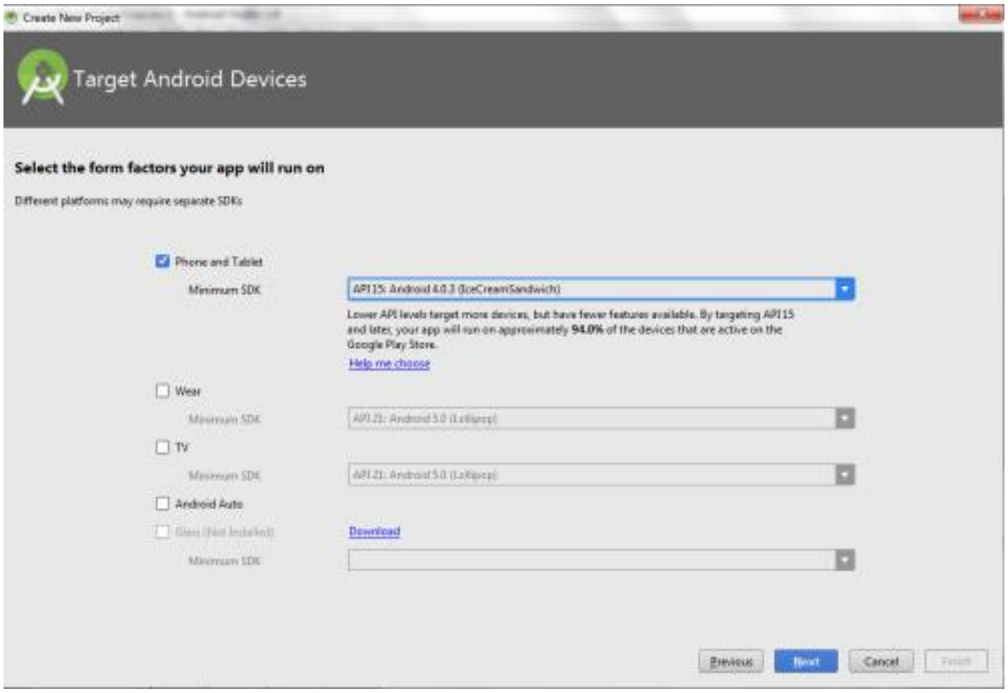
Company Domain: devang.example.com

Package name: com.example.devang.exno5 Edit

Project location: C:\Users\devang\AndroidStudioProjects\ex.no.5

The application name for most apps begins with an uppercase letter

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


Create New Project

Target Android Devices

Select the form factors your app will run on

Different platforms may require separate SDKs

☑ Phone and Tablet

Minimum SDK: API 15: Android 4.0.3 (IceCreamSandwich)

Lower API levels target more devices, but have fewer features available. By targeting API 15 and later, your app will run on approximately **94.0%** of the devices that are active on the Google Play Store.

Help me choose

□ Wear

Minimum SDK: API 21: Android 5.0 (Lollipop)

□ TV

Minimum SDK: API 21: Android 5.0 (Lollipop)

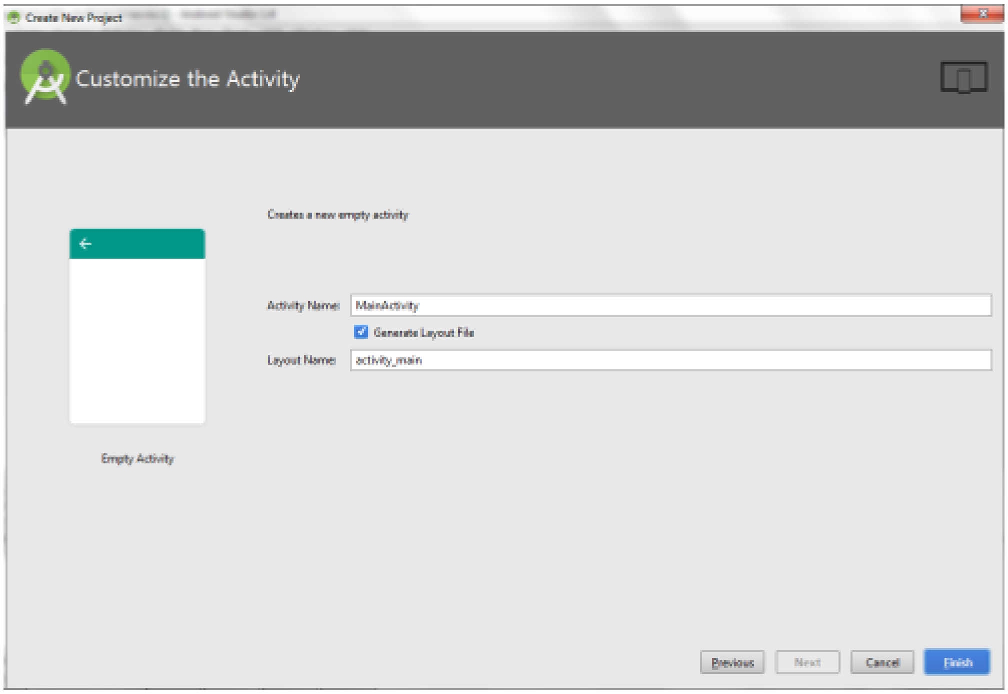
□ Android Auto

□ Glass (Not Installed) Download

Minimum SDK

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

- Finally click F**inish**.


Create New Project

Customize the Activity

Creates a new empty activity

Activity Name: MainActivity

☑ Generate Layout File

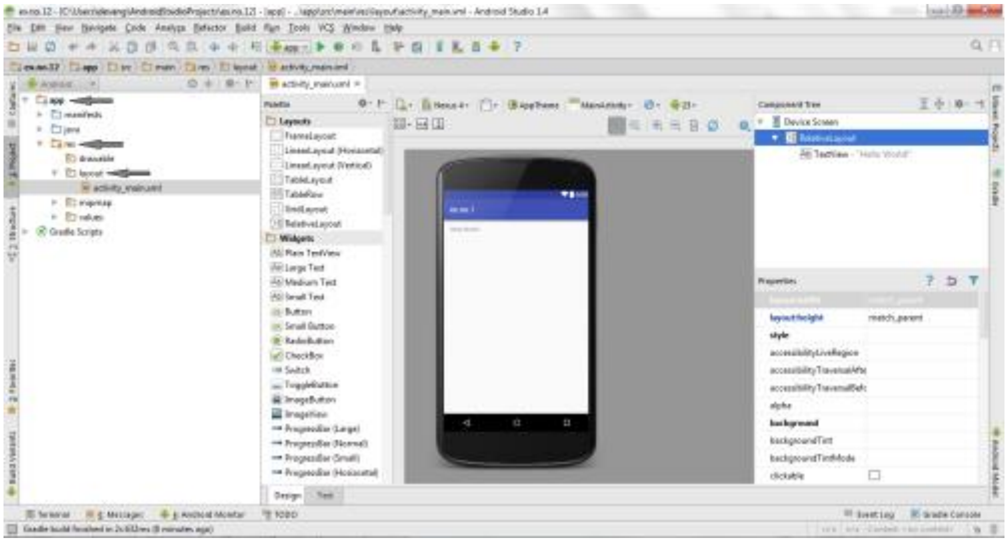
Layout Name: activity_main

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

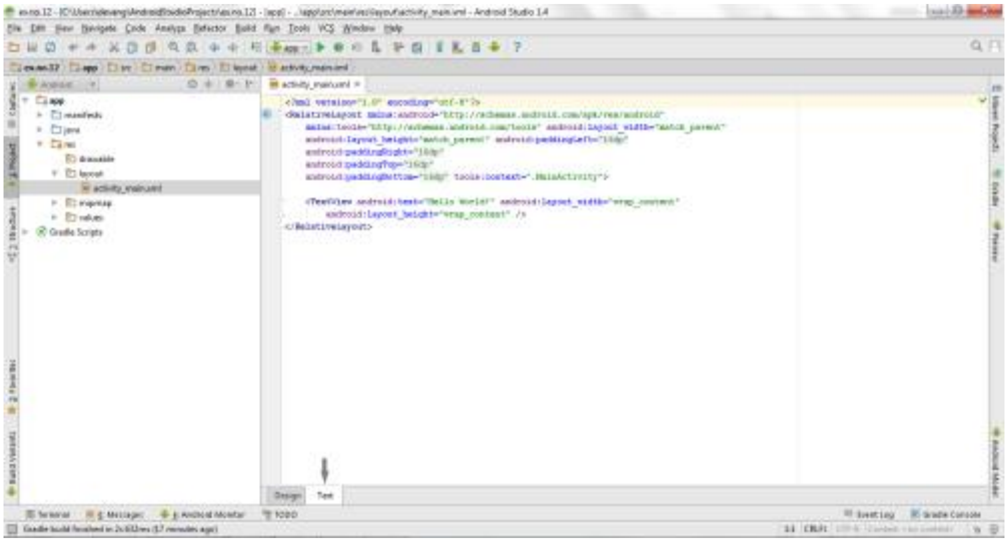
- After completion it will look as given below.

Designing layout for the Android Application:

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



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



```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
    android:layout_height="match_parent" android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    android:paddingBottom="16dp" tools:context=".MainActivity">

    <TextView android:text="Hello World!" android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</RelativeLayout>
```

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

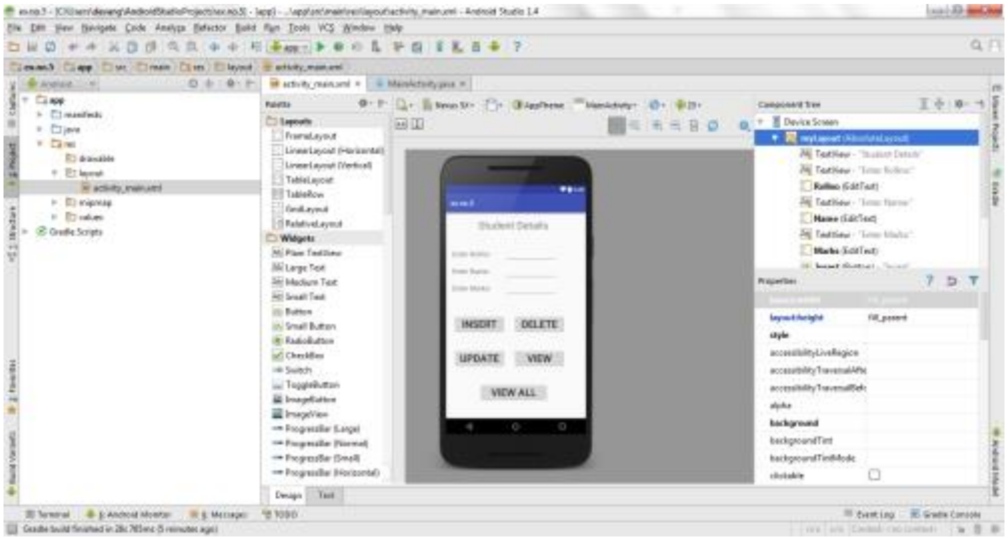
Code for Activity_main.xml:

?

```
<?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>
```

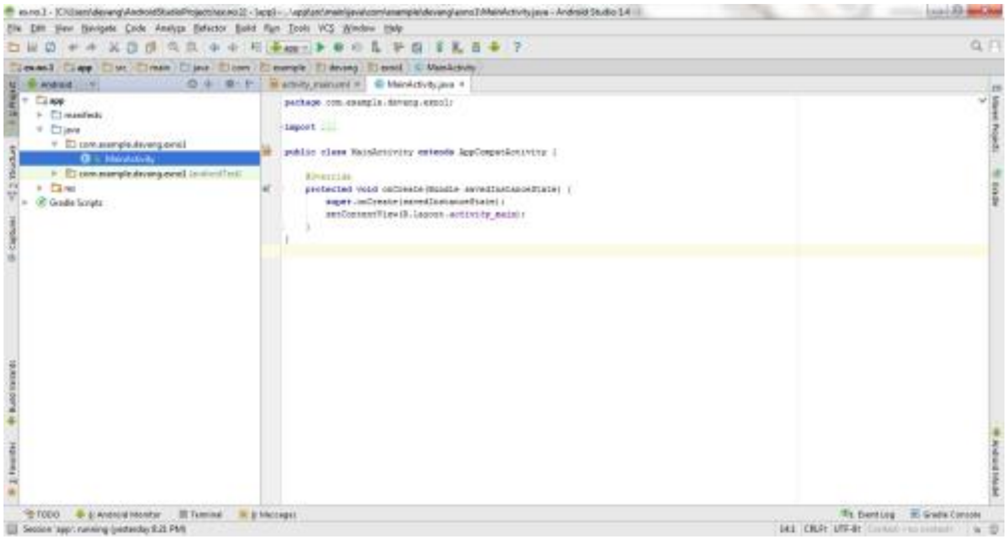
- Now click on **Design** and your application will look as given below.



- So now the designing part is completed.

Java Coding for the Android Application:

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



```
package com.example.devang.exno1;

import ...

public class MainActivity extends AppCompatActivity {

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

- 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();
        }
```

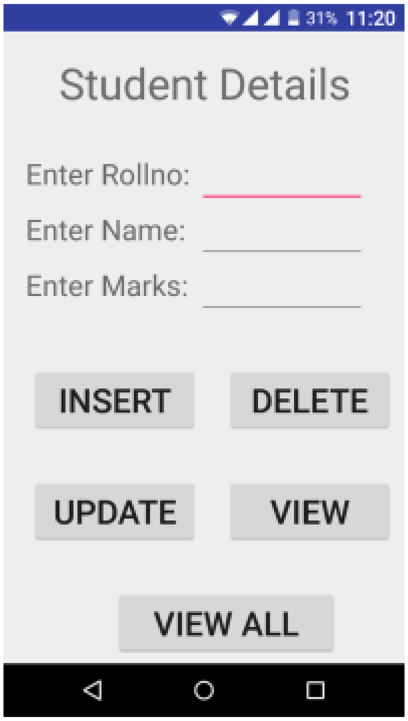
```
// Deleting a record from the Student table
if(view==Delete)
{
    // Checking for empty roll number
    if(Rollno.getText().toString().trim().length()==0)
    {
        showMessage("Error", "Please enter Rollno");
        return;
    }
    Cursor c=db.rawQuery("SELECT * FROM student WHERE rollno='"+Rollno.getText()+"'", 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:

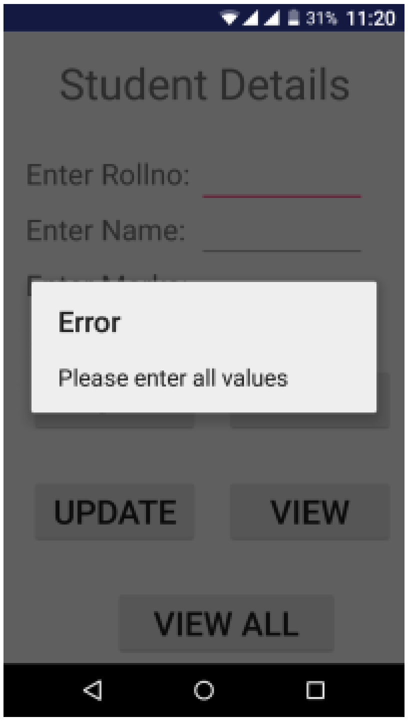


Student Details

Enter Rollno:

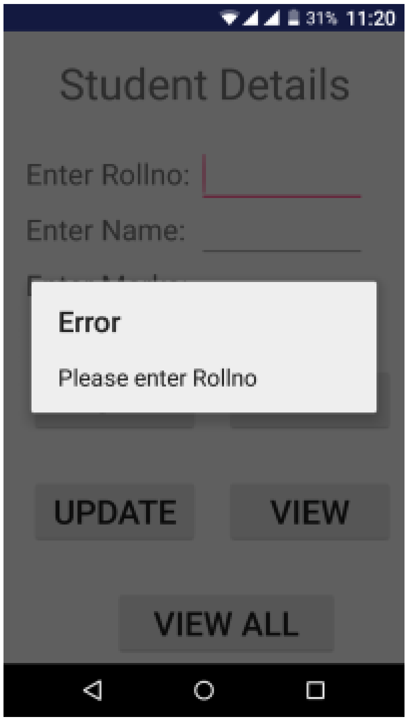
Enter Name:

Enter Marks:



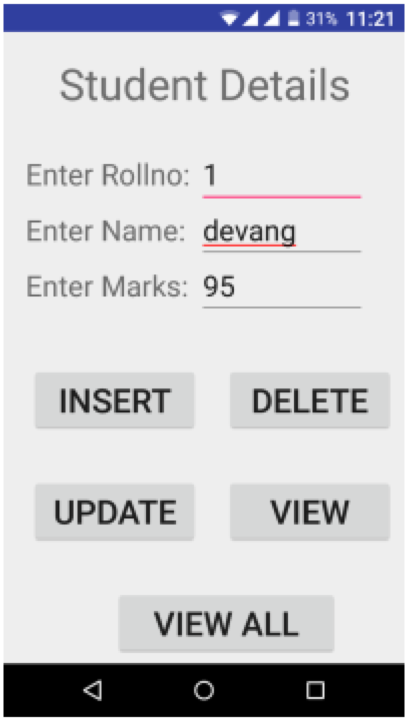
Error

Please enter all values



Error

Please enter Rollno

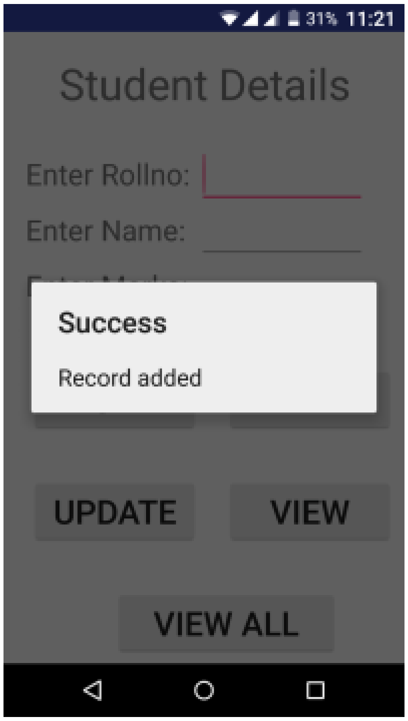


Student Details

Enter Rollno: 1

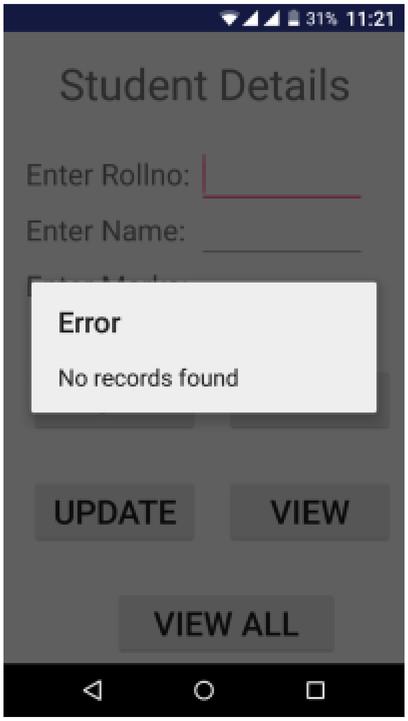
Enter Name: devang

Enter Marks: 95



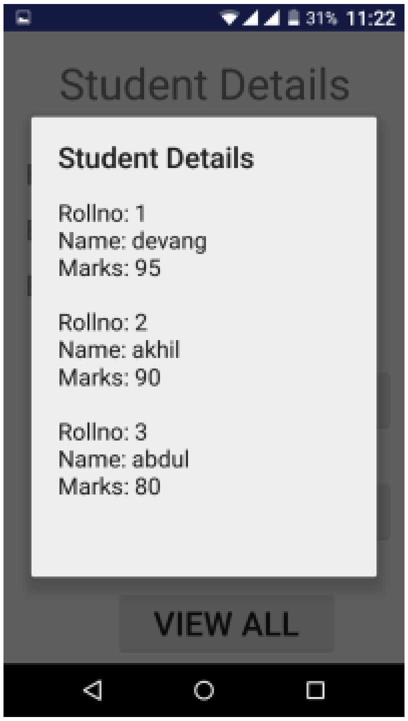
Success

Record added



Error

No records found

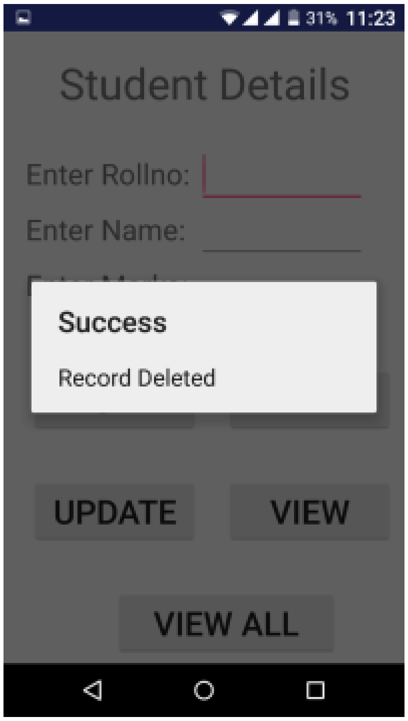


Student Details

Rollno: 1
Name: devang
Marks: 95

Rollno: 2
Name: akhil
Marks: 90

Rollno: 3
Name: abdul
Marks: 80

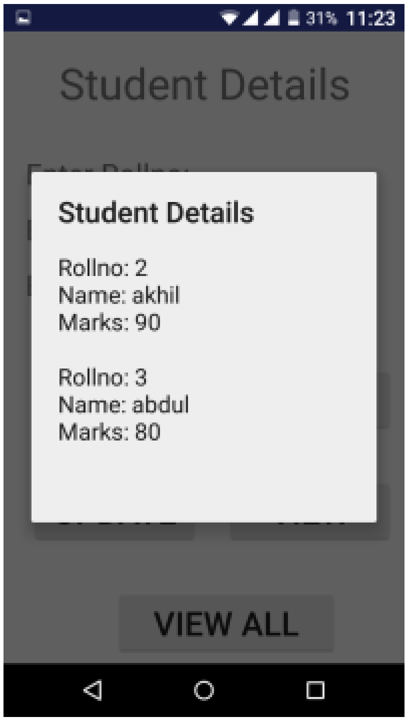


Enter Rollno:

Enter Name:

Success

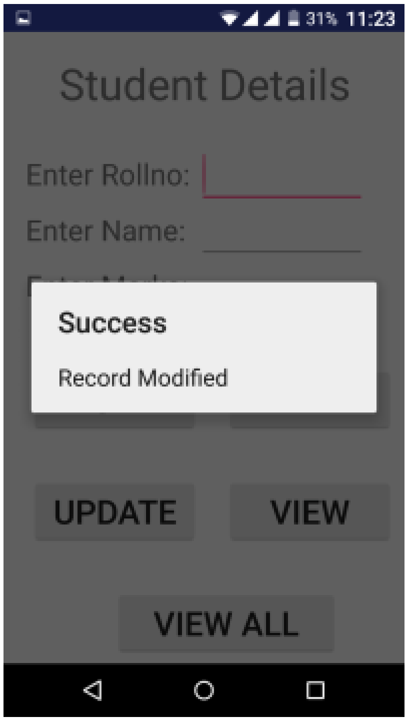
Record Deleted



Student Details

Rollno: 2
Name: akhil
Marks: 90

Rollno: 3
Name: abdul
Marks: 80

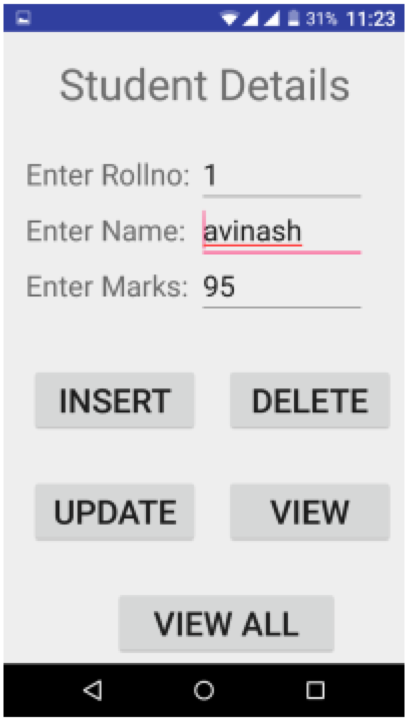


Enter Rollno:

Enter Name:

Success

Record Modified



Enter Rollno: 1

Enter Name: avinash

Enter Marks: 95

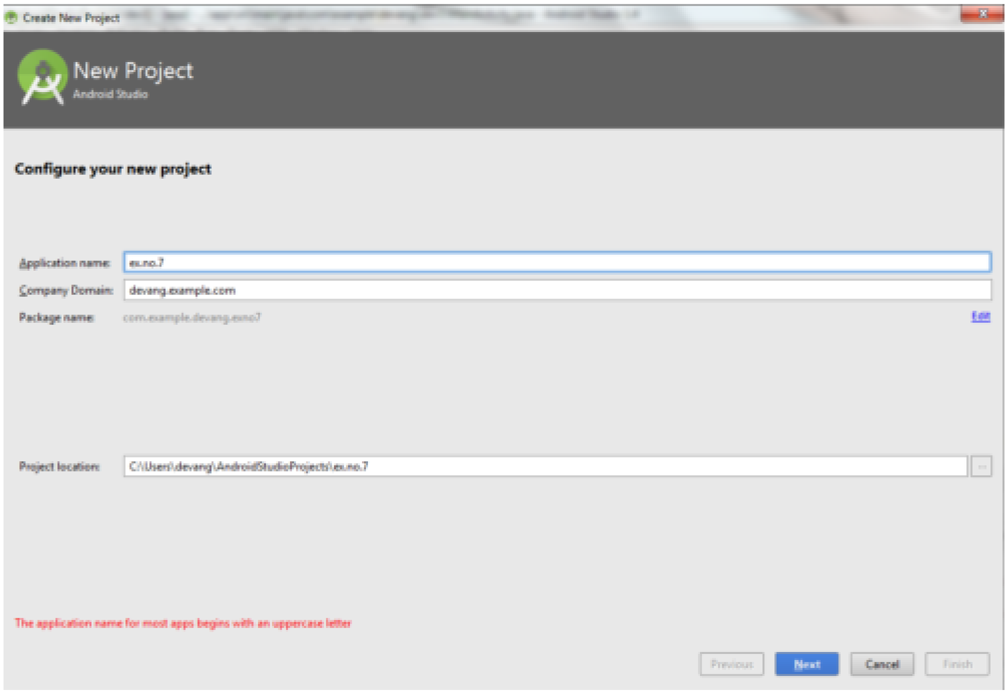
Result:

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

Android Application that implements Multi threading

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.**



Configure your new project

Application name: ex.no.7

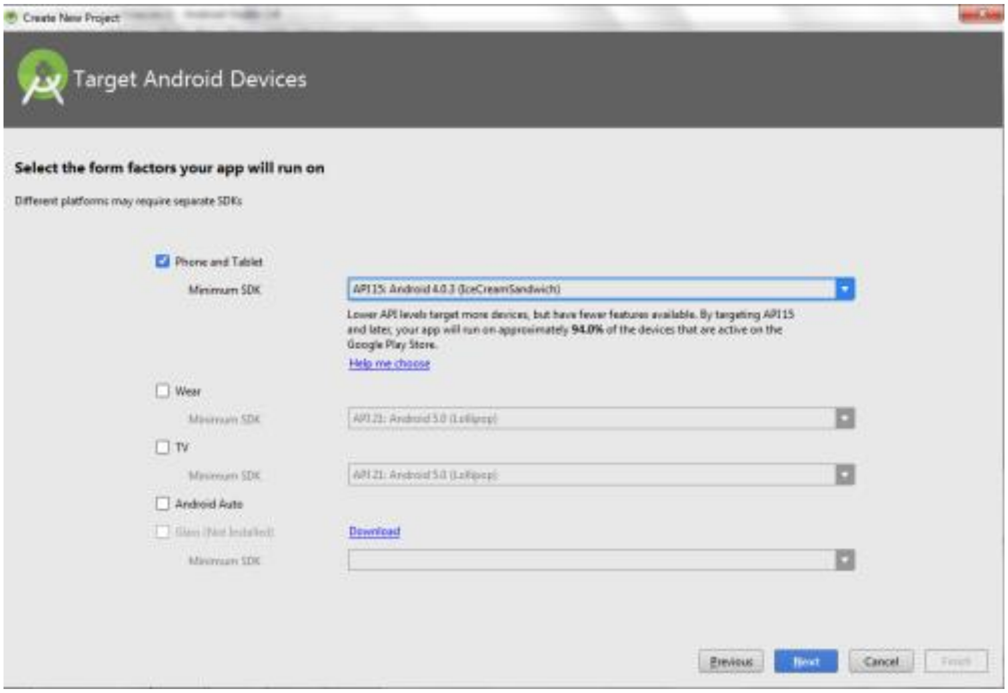
Company Domain: devang.example.com

Package name: com.example.devang.exno7

Project location: C:\Users\devang\AndroidStudioProjects\ex.no.7

The application name for most apps begins with an uppercase letter

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



Create New Project

Target Android Devices

Select the form factors your app will run on

Different platforms may require separate SDKs

☑ Phone and Tablet

Minimum SDK: API 15: Android 4.0.3 (IceCreamSandwich)

Lower API levels target more devices, but have fewer features available. By targeting API 15 and later, your app will run on approximately **94.0%** of the devices that are active on the Google Play Store.

Help me choose

☐ Wear

Minimum SDK: API 21: Android 5.0 (Lollipop)

☐ TV

Minimum SDK: API 21: Android 5.0 (Lollipop)

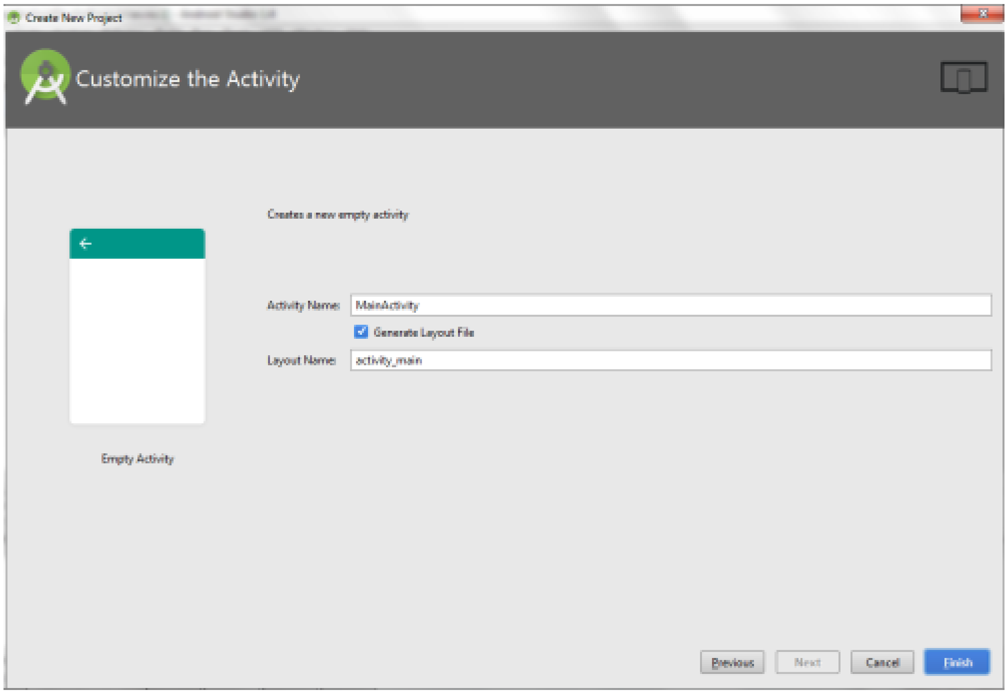
☐ Android Auto

☐ Glass (Not Installed) Download

Minimum SDK:

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

- Finally click F**inish**.



Create New Project

Customize the Activity

Creates a new empty activity

Activity Name: MainActivity

☑ Generate Layout File

Layout Name: activity_main

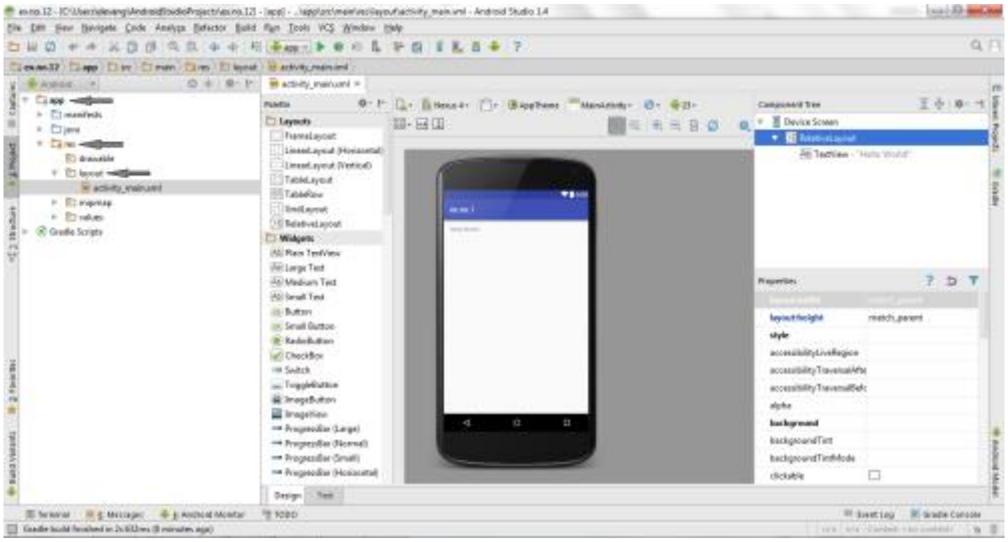
Empty Activity

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

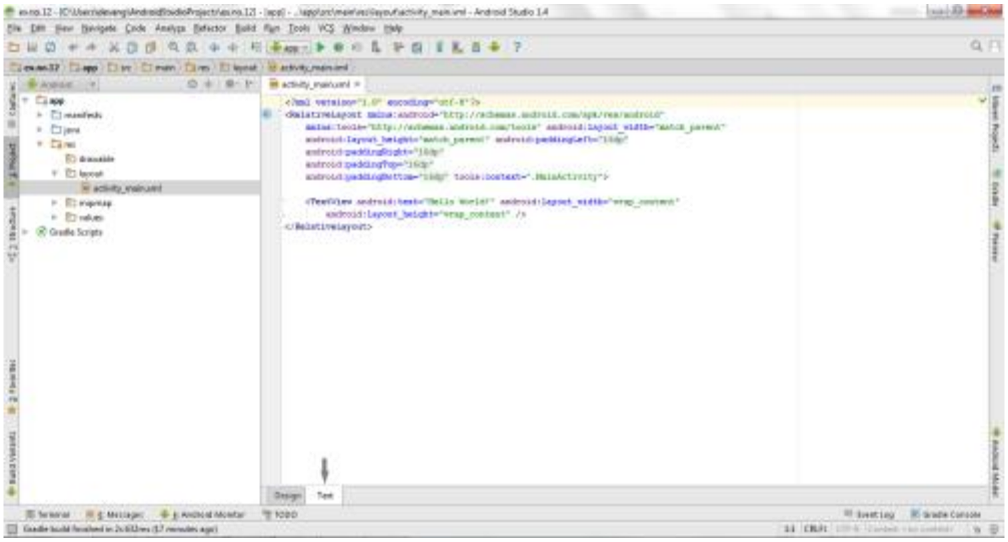
- After completion it will look as given below.

Designing layout for the Android Application:

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



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



```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
    android:layout_height="match_parent" android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    android:paddingBottom="16dp" tools:context=".MainActivity">

    <TextView android:text="Hello World!" android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</RelativeLayout>
```

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

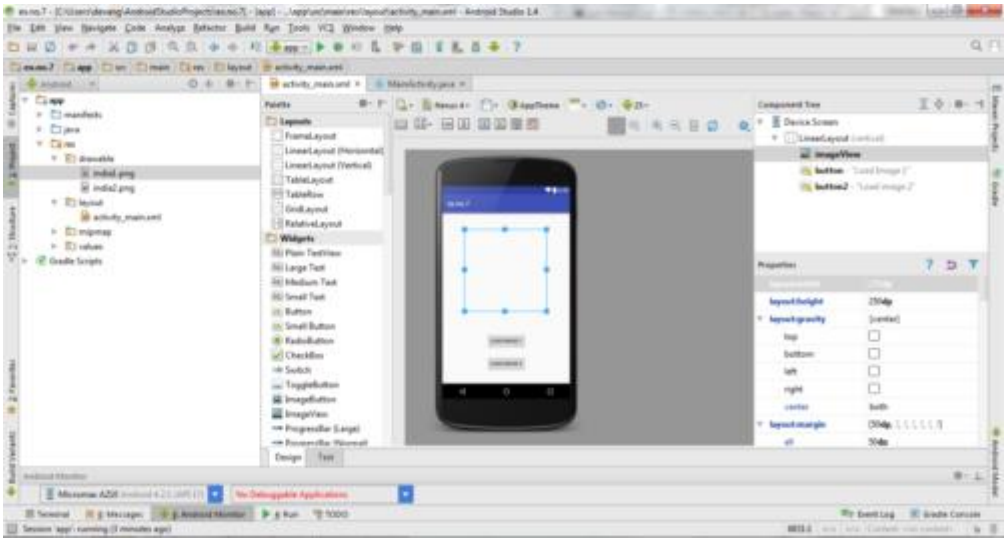
Code for Activity_main.xml:

?

```
<?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>
```

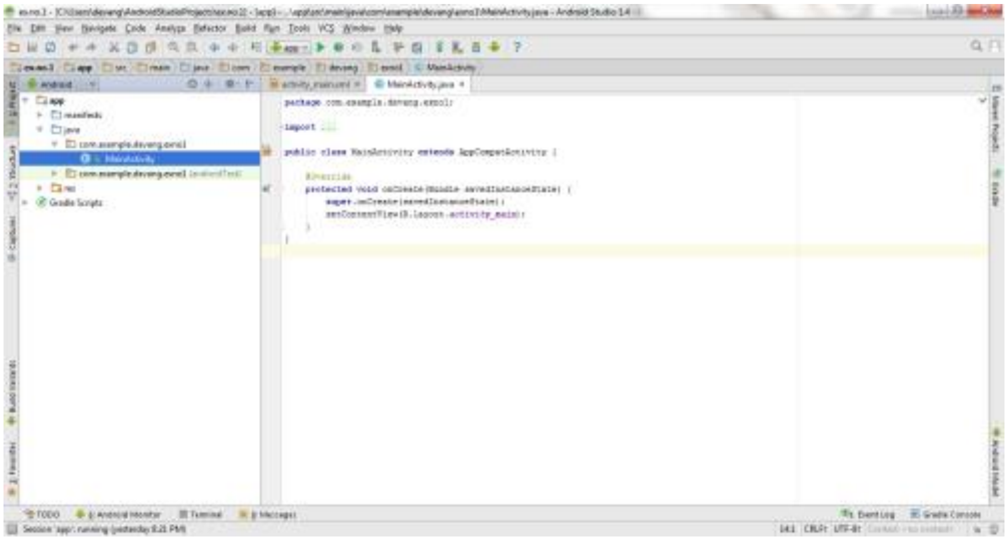
- Now click on **Design** and your application will look as given below.



- So now the designing part is completed.

Java Coding for the Android Application:

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



```
package com.example.devang.exno1;

import ...;

public class MainActivity extends AppCompatActivity {

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

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

Code for MainActivity.java:

?

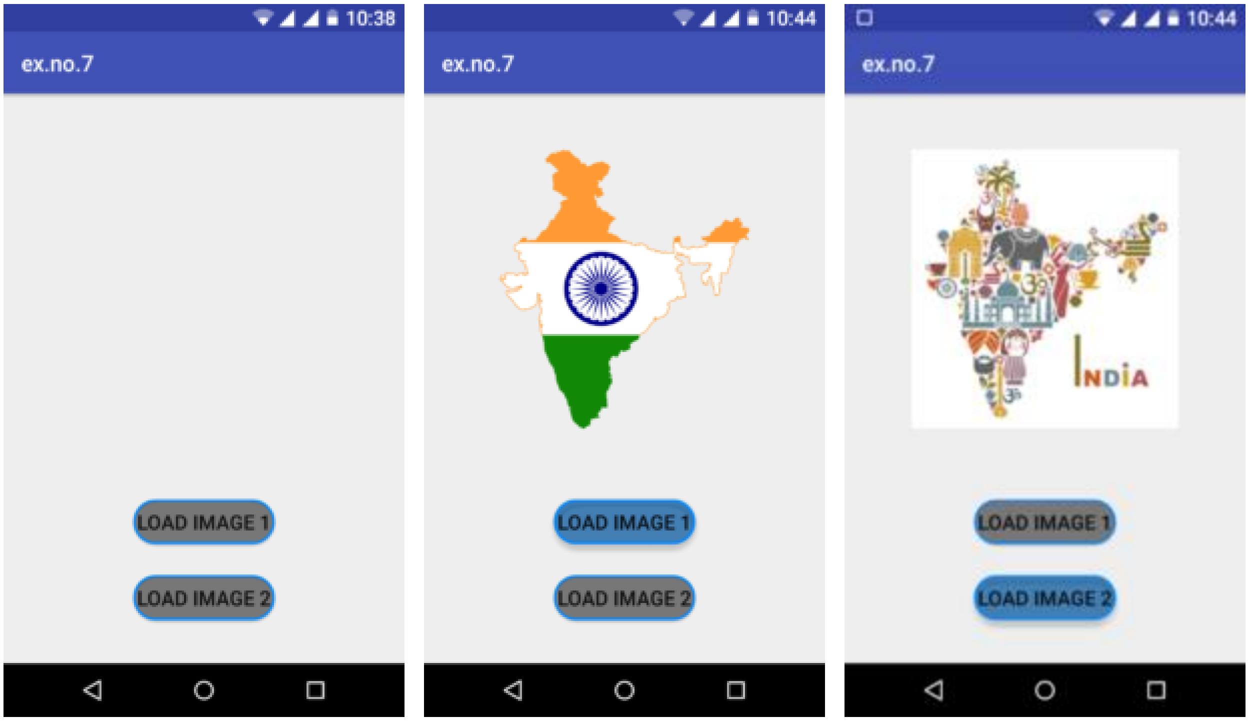
```
package com.example.exno7;

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.

TO DOWNLOAD THE IMAGES : CLICK HERE

Output:



Result:

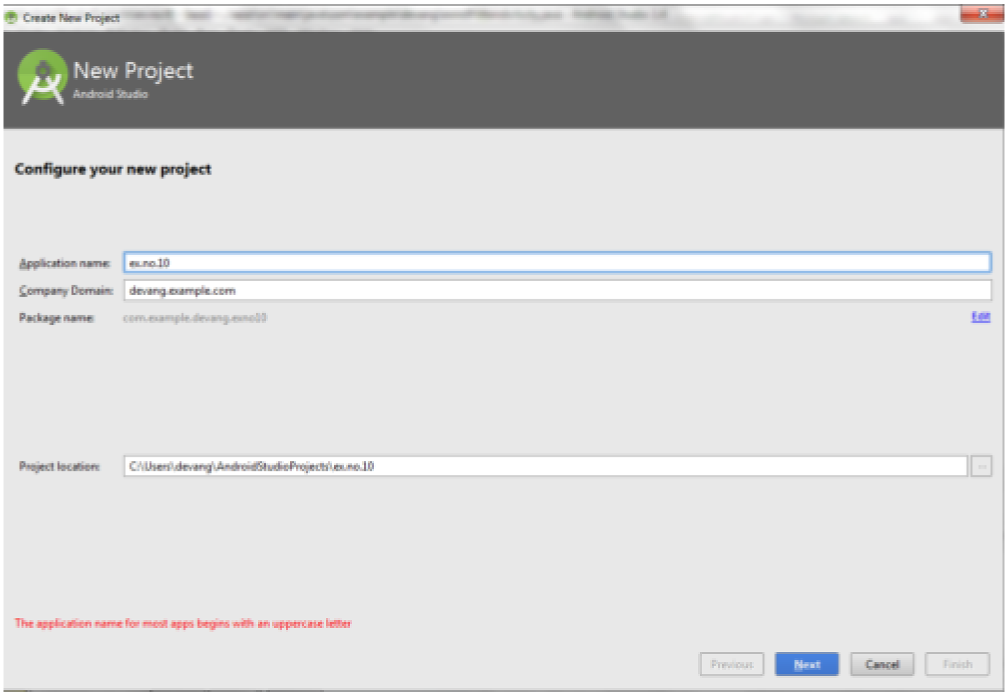
Thus Android Application that implements Multi threading is developed and executed successfully.

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.**



Configure your new project

Application name: ex.no.10

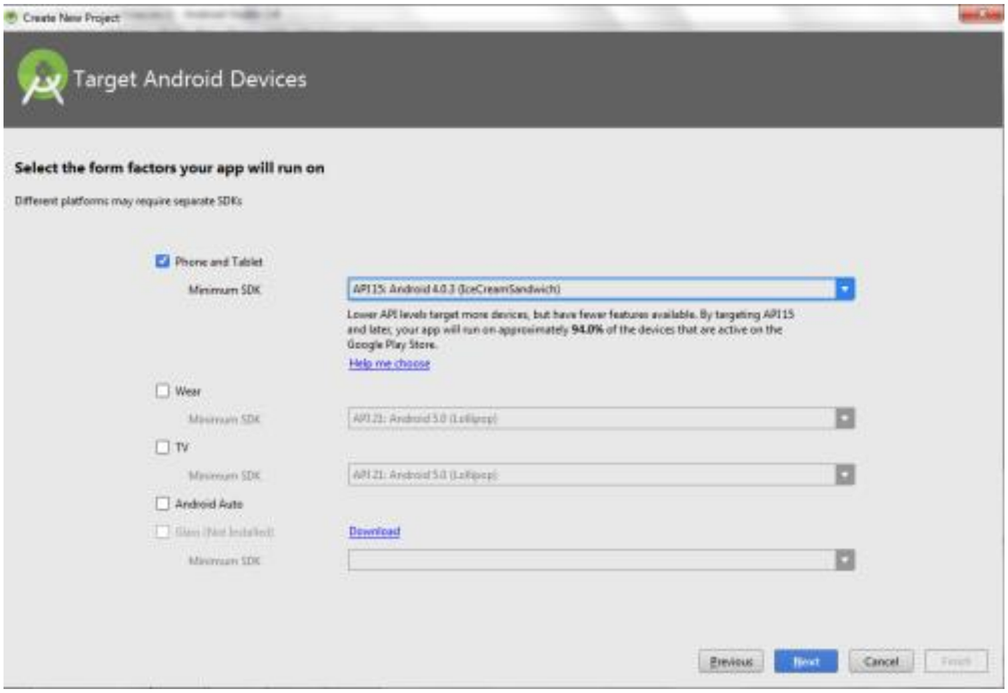
Company Domain: devang.example.com

Package name: com.example.devang.exno10

Project location: C:\Users\devang\AndroidStudioProjects\ex.no.10

The application name for most apps begins with an uppercase letter

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



Create New Project

Target Android Devices

Select the form factors your app will run on

Different platforms may require separate SDKs

☑ Phone and Tablet

Minimum SDK: API 15: Android 4.0.3 (IceCreamSandwich)

Lower API levels target more devices, but have fewer features available. By targeting API 15 and later, your app will run on approximately **94.0%** of the devices that are active on the Google Play Store.

Help me choose

□ Wear

Minimum SDK: API 21: Android 5.0 (Lollipop)

□ TV

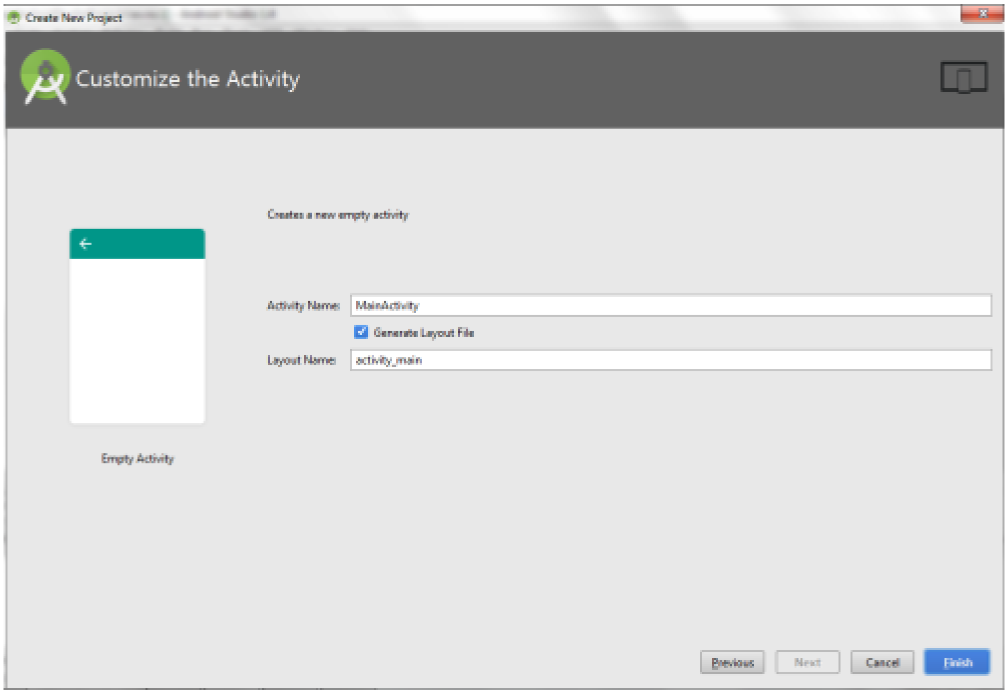
Minimum SDK: API 21: Android 5.0 (Lollipop)

□ Android Auto

□ Glass (Not Installed) Download

Minimum SDK:

- Then select the **Empty Activity** and click **Next.**
- Finally click F**inish**.



Create New Project

Customize the Activity

Creates a new empty activity

Activity Name: MainActivity

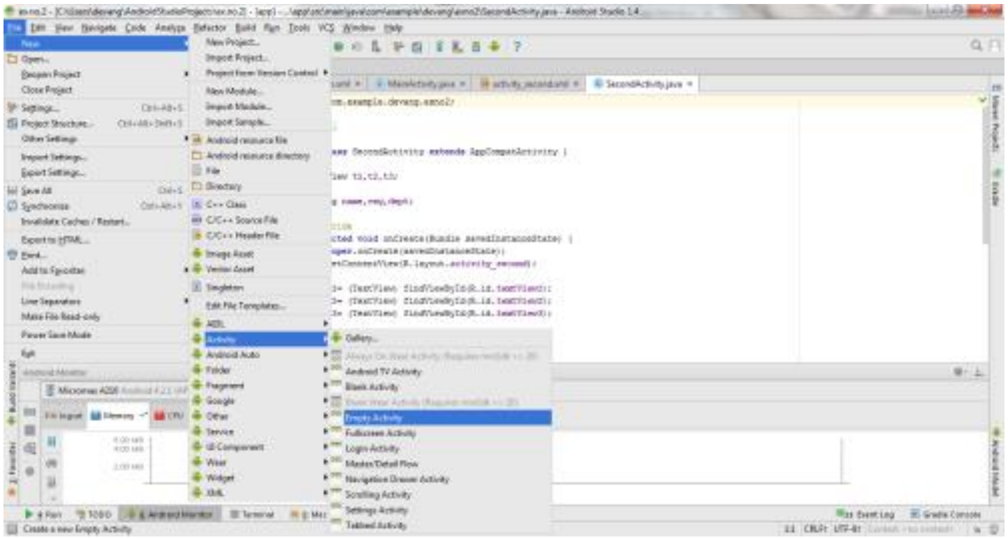
☑ Generate Layout File

Layout Name: activity_main

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

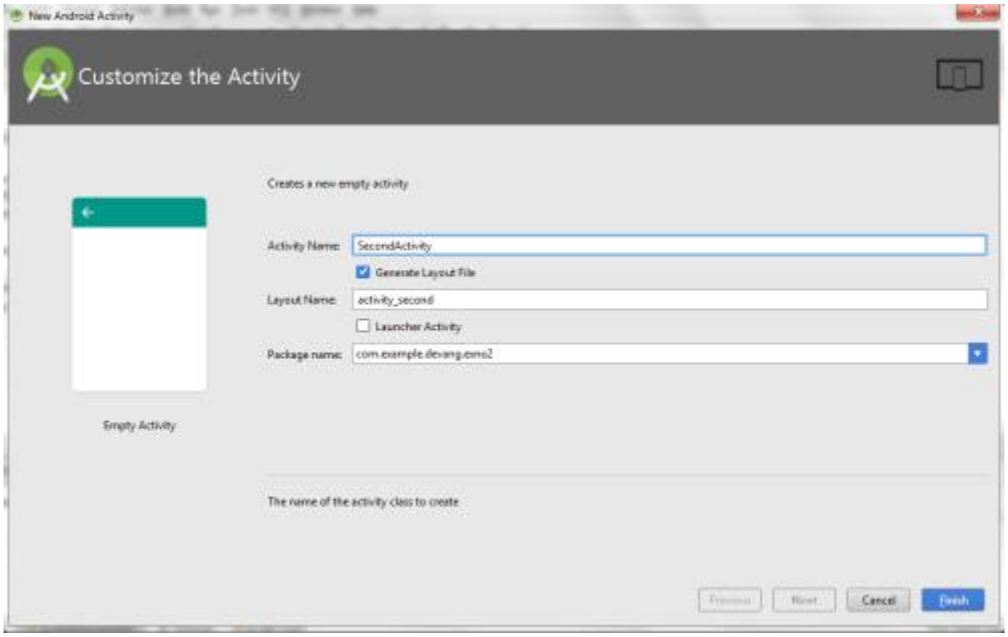
Creating Second Activity for the Android Application:

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



```
m.example.devang.exno2;

ass SecondActivity extends AppCompatActivity {

iew t1,t2,t3;

g name,reg,dept;

ride
cted void onCreate(Bundle savedInstanceState) {
uper.onCreate(savedInstanceState);
etContentView(R.layout.activity_second);

1= (TextView) findViewById(R.id.textView1);
2= (TextView) findViewById(R.id.textView2);
3= (TextView) findViewById(R.id.textView3);
```

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



New Android Activity

Customize the Activity

Creates a new empty activity

Activity Name: SecondActivity

☑ Generate Layout File

Layout Name: activity_second

☐ Launcher Activity

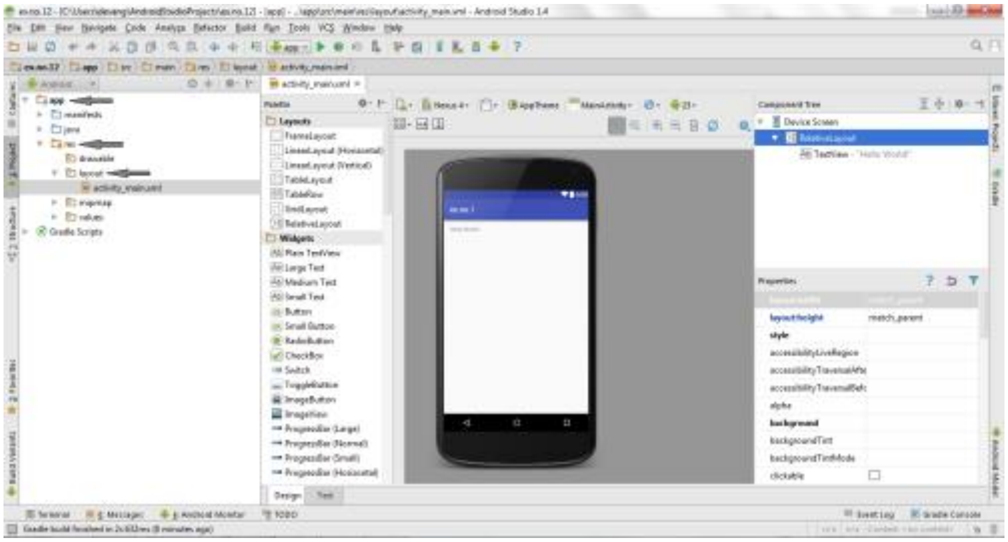
Package name: com.example.devang.exno2

The name of the activity class to create

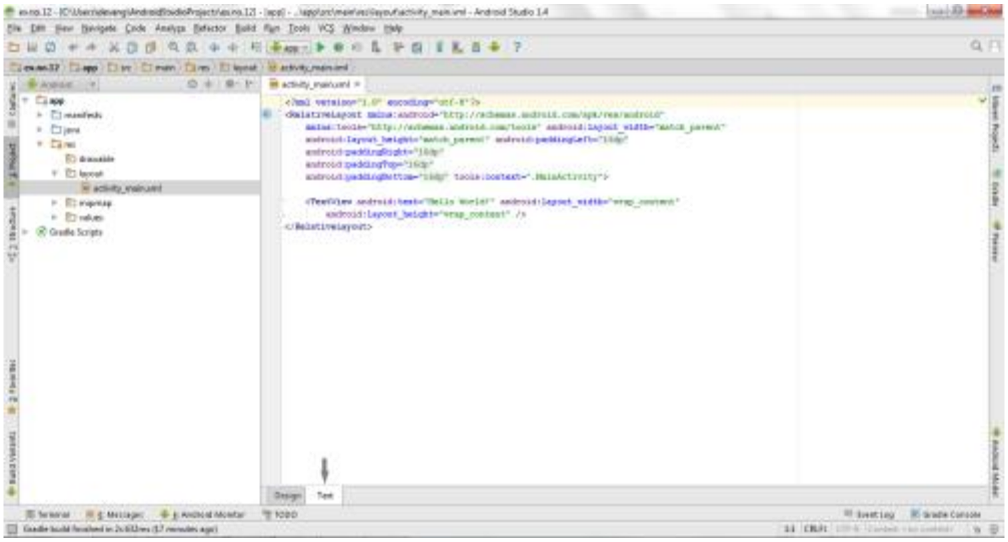
- Thus Second Activity For the application is created.

Designing layout for the Android Application:

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



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



```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
    android:layout_height="match_parent" android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    android:paddingBottom="16dp" tools:context=".MainActivity">

    <TextView android:text="Hello World!" android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</RelativeLayout>
```

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

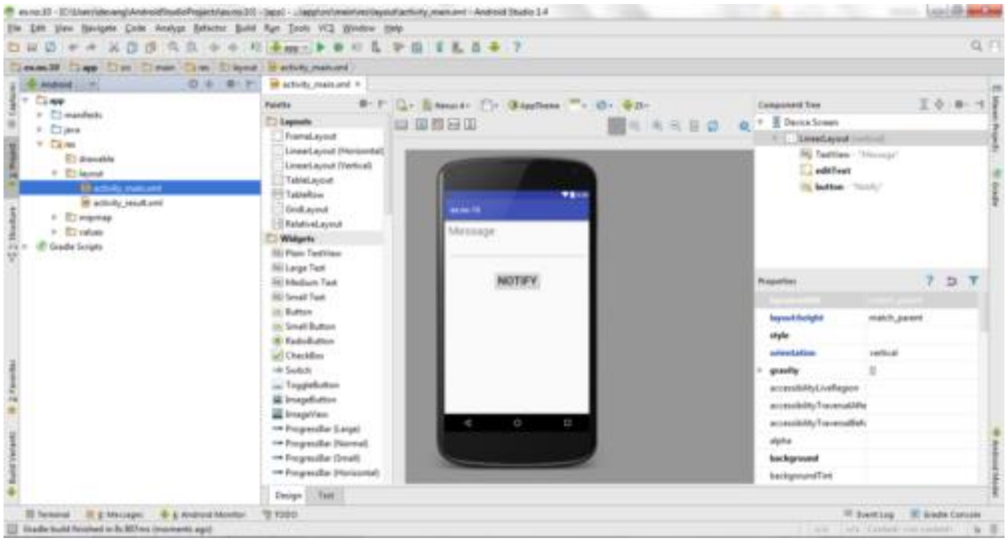
Code for Activity_main.xml:

?

```
<?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>
```

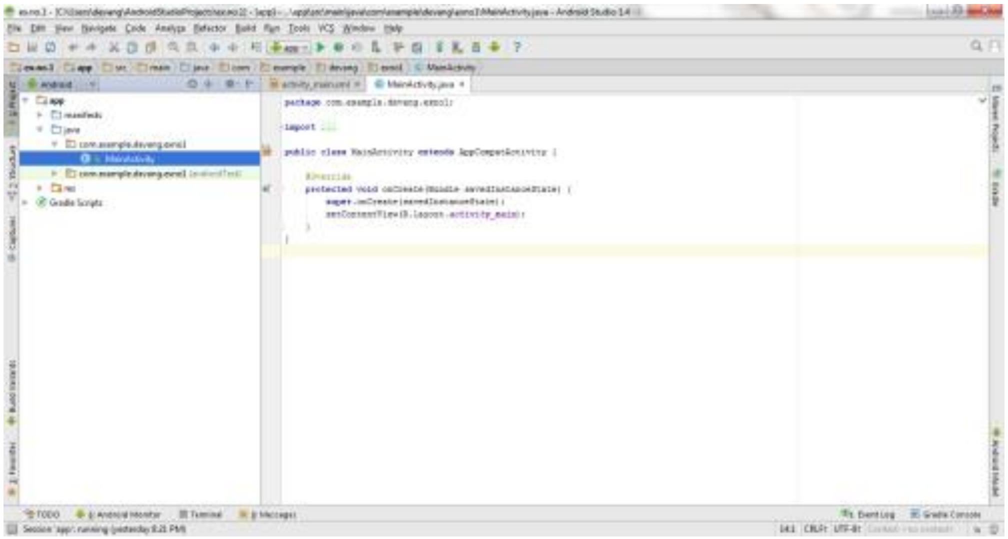
- Now click on **Design** and your application will look as given below.



- So now the designing part is completed.

Java Coding for the Android Application:

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



```
package com.example.devang.exno1;

import ...

public class MainActivity extends AppCompatActivity {

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

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

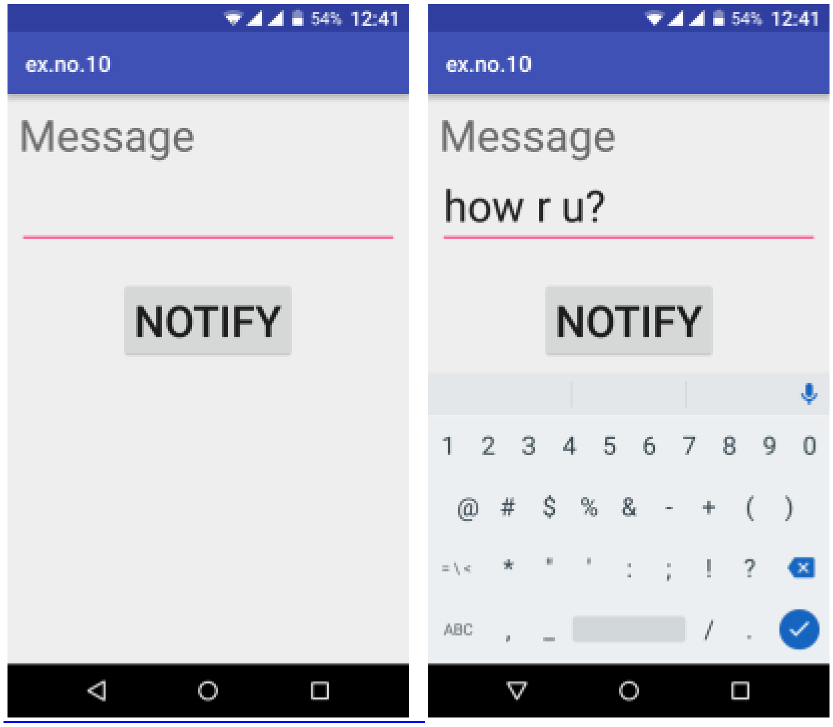
Code for MainActivity.java:

?

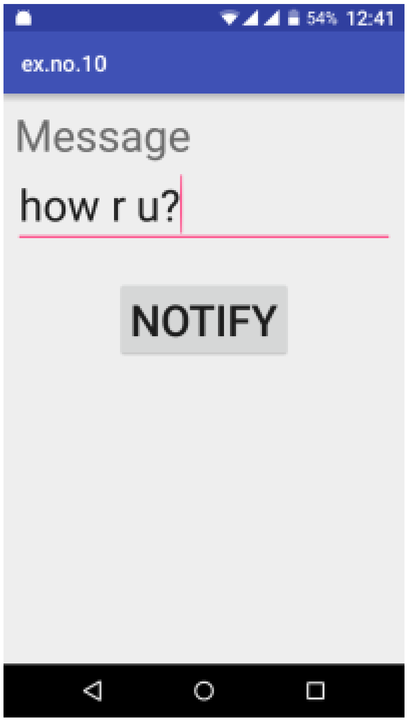
```
package com.example.exno10;

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

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

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

        notify.setOnClickListener(new View.OnClickListener()
        {
            @Override
            public void onClick(View v)
            {
                Intent intent = new Intent(MainActivity.this, SecondActivity.class);
                PendingIntent pending = PendingIntent.getActivity(MainActivity.this, 0, intent, 0);
                Notification noti = new Notification.Builder(MainActivity.this).setContentTitle("New Message")
                .setContentText(e.getText().toString()).setSmallIcon(R.mipmap.ic_launcher).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:

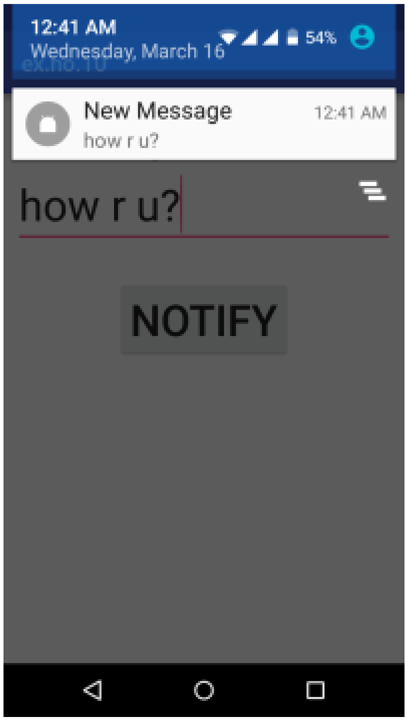


how r u?



Message

how r u?



New Message 12:41 AM

how r u?

how r u?

Result:

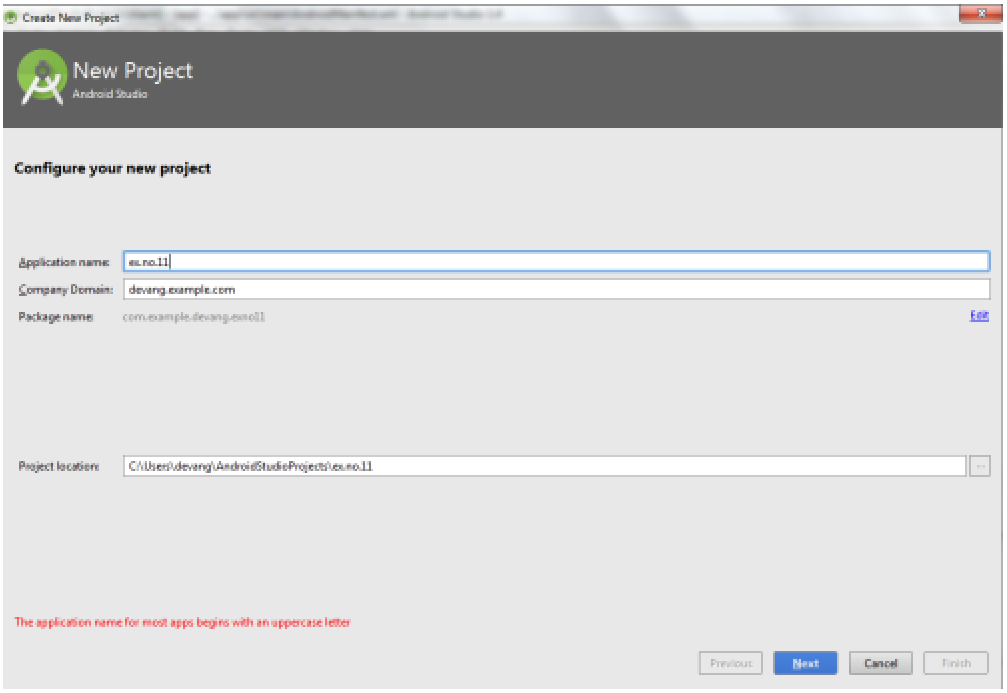
Thus Android Application that creates an alert upon receiving a message is developed and executed successfully.

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.**



Configure your new project

Application name: ex.no.11

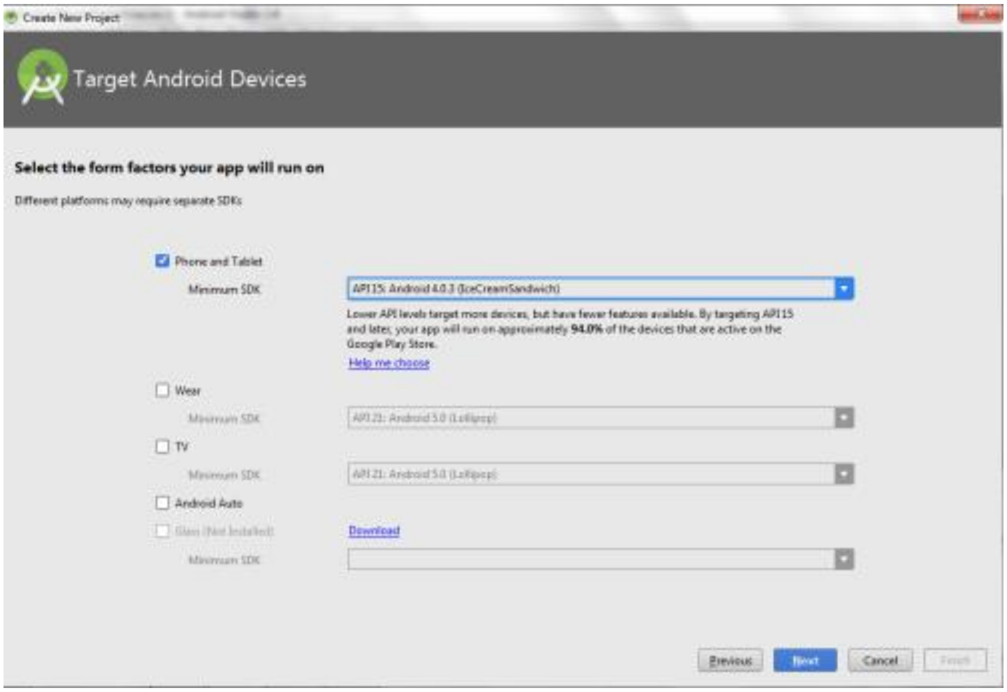
Company Domain: devang.example.com

Package name: com.example.devang.exno11

Project location: C:\Users\devang\AndroidStudioProjects\ex.no.11

The application name for most apps begins with an uppercase letter

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



Target Android Devices

Select the form factors your app will run on

Different platforms may require separate SDKs

☑ Phone and Tablet

Minimum SDK: API 15: Android 4.0.3 (IceCreamSandwich)

Lower API levels target more devices, but have fewer features available. By targeting API 15 and later, your app will run on approximately **94.0%** of the devices that are active on the Google Play Store.

□ Wear

Minimum SDK: API 21: Android 5.0 (Lollipop)

□ TV

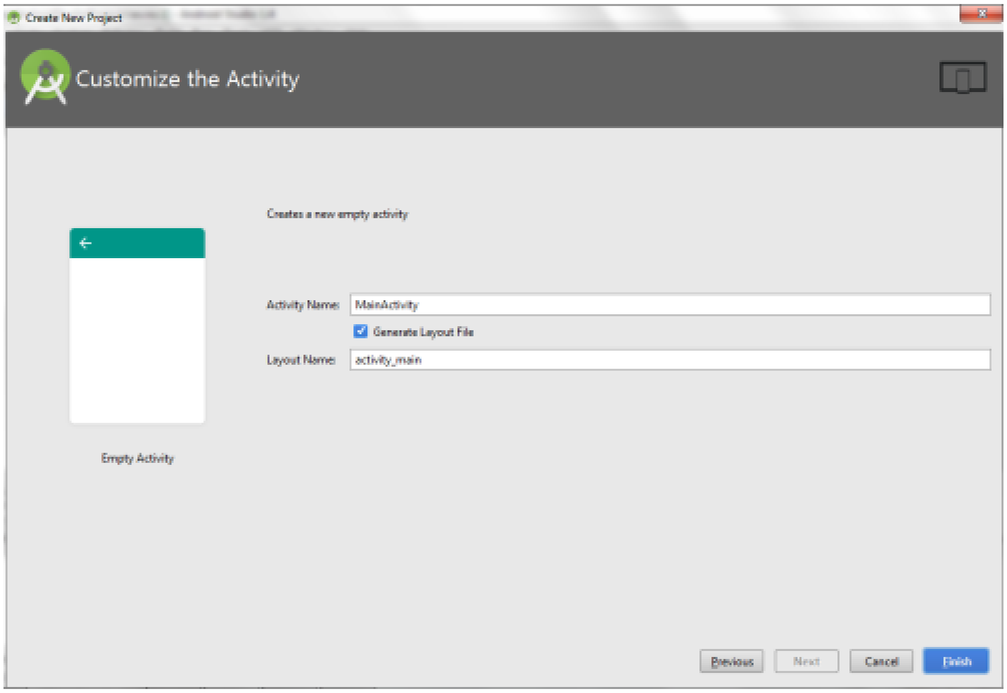
Minimum SDK: API 21: Android 5.0 (Lollipop)

□ Android Auto

□ Glass (Not Installed)

Minimum SDK:

- Then select the **Empty Activity** and click **Next.**
- Finally click F**inish**.



Customize the Activity

Creates a new empty activity

Activity Name: MainActivity

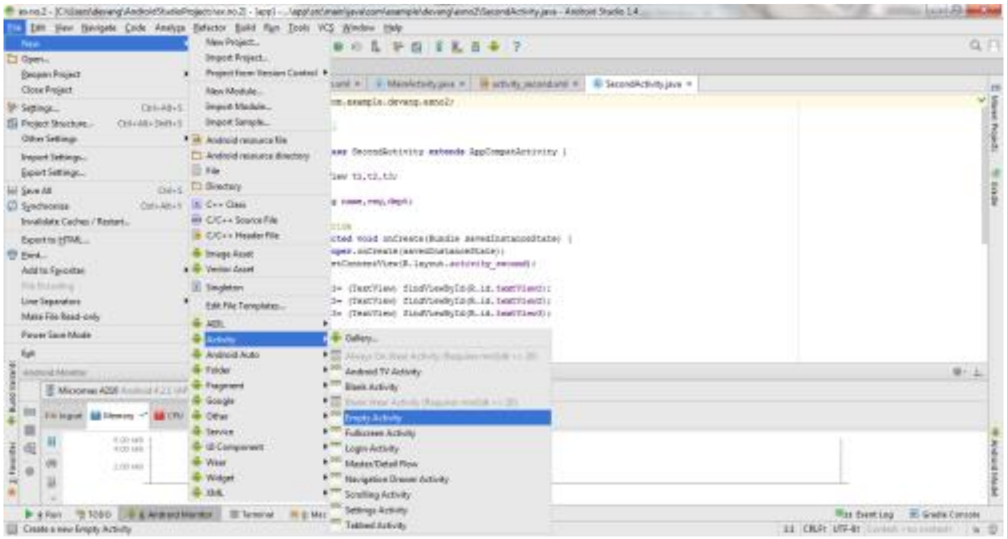
☑ Generate Layout File

Layout Name: activity_main

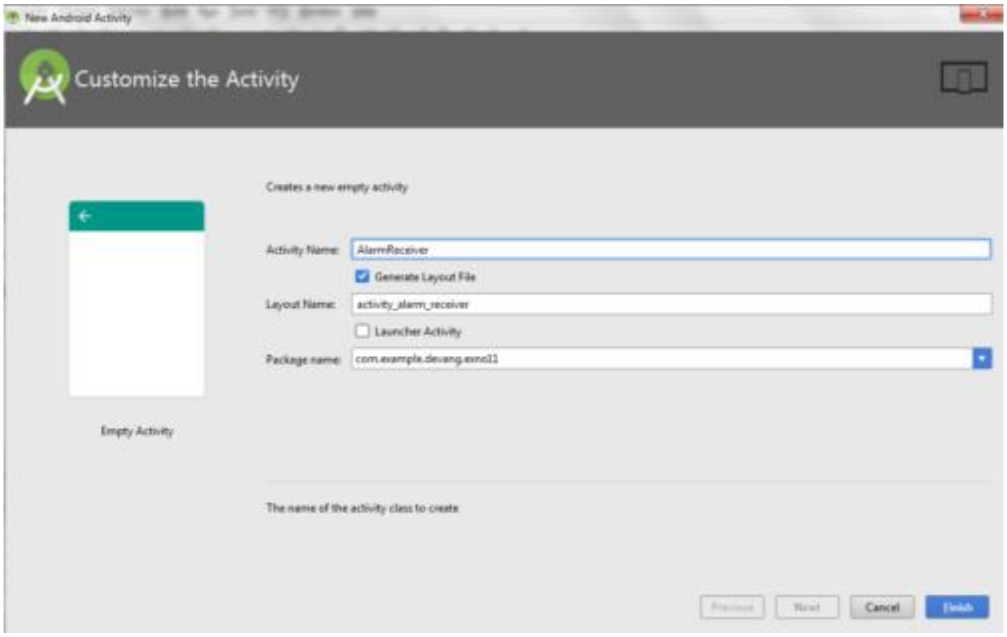
- It will take some time to build and load the project.
- After completion it will look as given below.

Creating Second Activity for the Android Application:

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



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



Creates a new empty activity

Activity Name: AlarmReceiver

☑ Generate Layout File

Layout Name: activity_alarm_receiver

☐ Launcher Activity

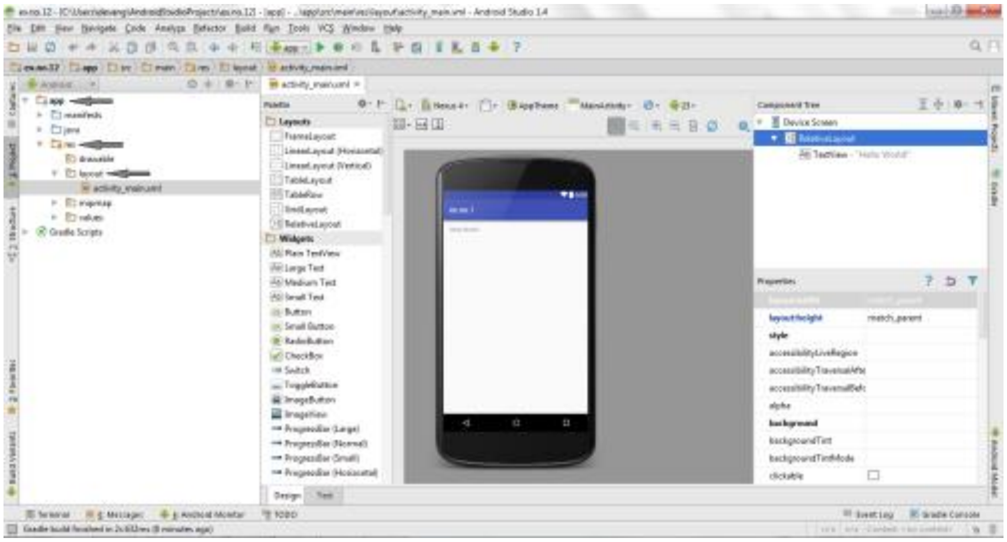
Package name: com.example.devang.exno11

The name of the activity class to create

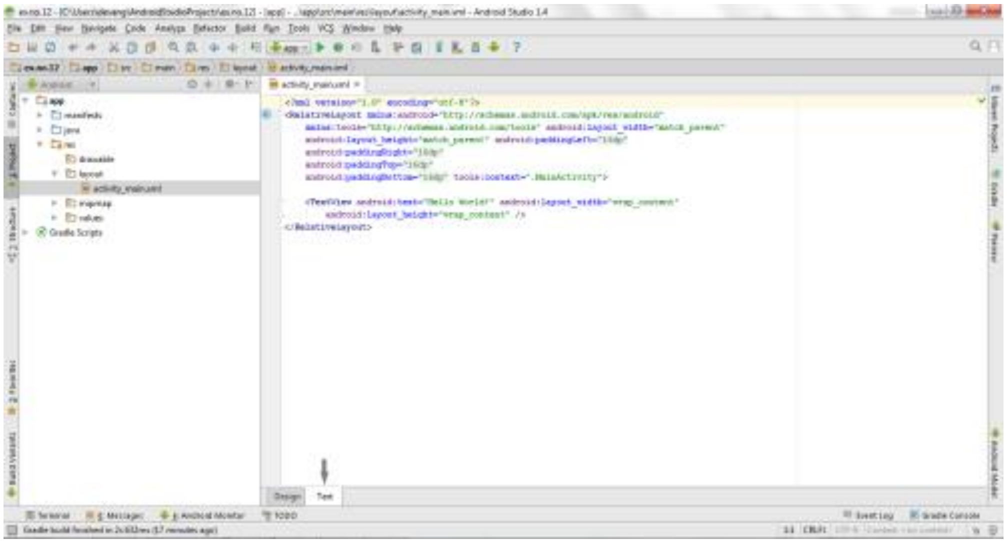
- Thus Second Activity For the application is created.

Designing layout for the Android Application:

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



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



```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
    android:layout_height="match_parent" android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    android:paddingBottom="16dp" tools:context=".MainActivity">

    <TextView android:text="Hello World!" android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</RelativeLayout>
```

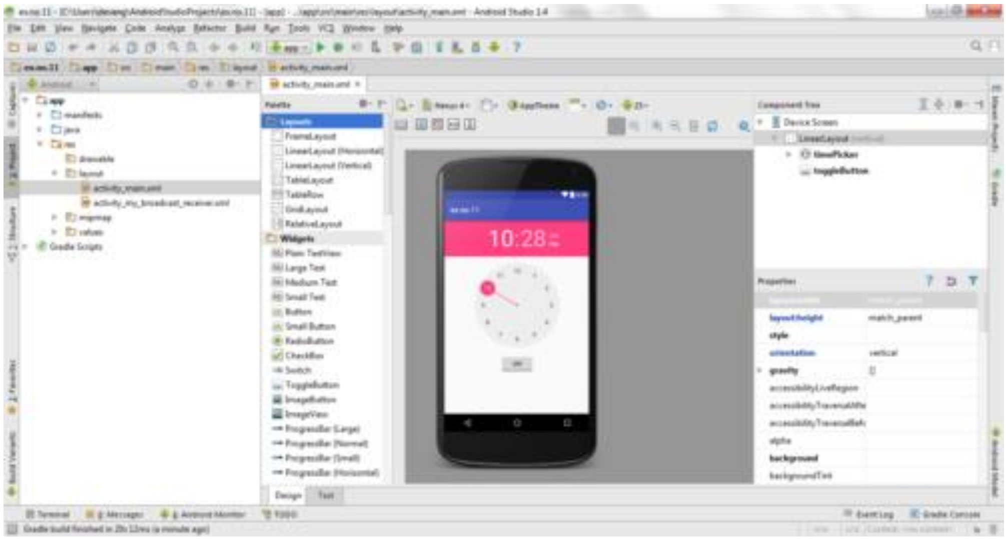
- Then delete the code which is there and type the code as given below.

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>
```

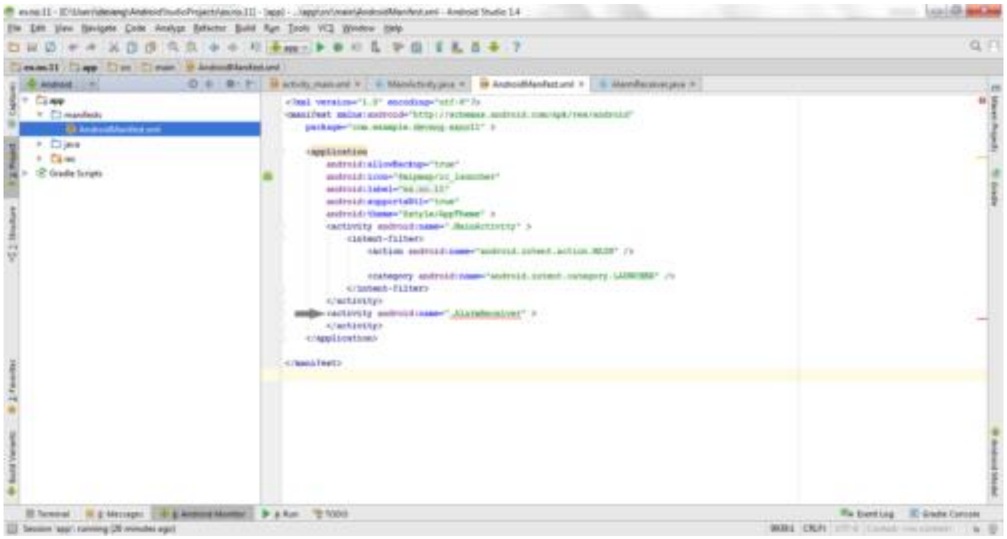
- Now click on **Design** and your application will look as given below.



- So now the designing part is completed.

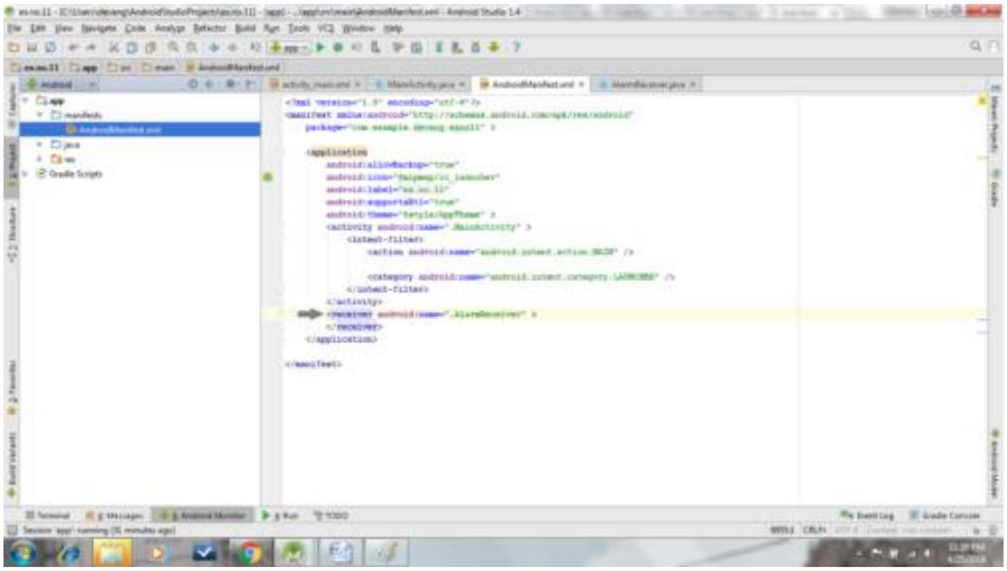
Changes in Manifest for the Android Application:

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



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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="ex.no.11"
        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>
        <activity android:name=".AlarmReceiver" >
        </activity>
    </application>

</manifest>
```

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



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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="ex.no.11"
        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>
```

Code for AndroidManifest.xml:

?

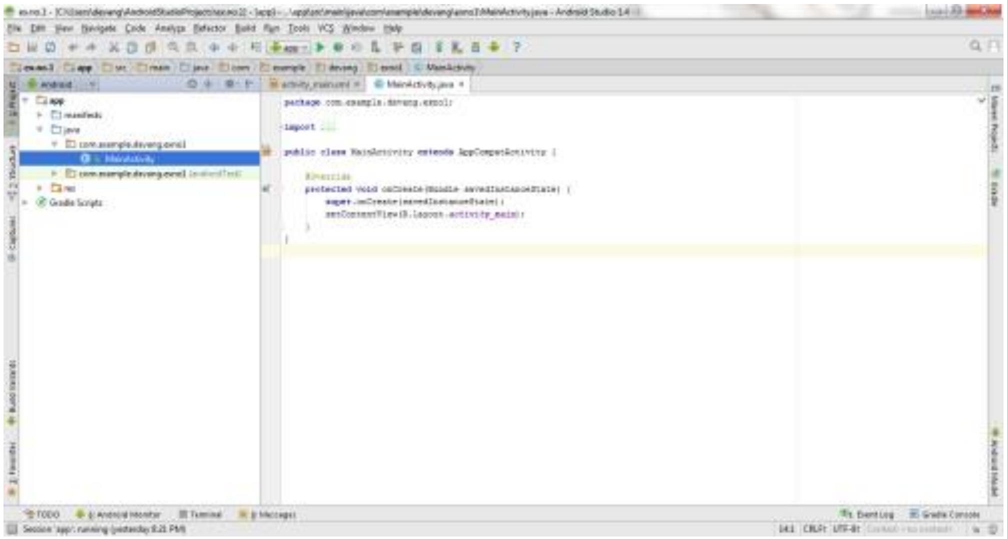
```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.exno11" >

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

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

</manifest>
```

- So now the changes are done in the Manifest.

Java Coding for the Android Application:

Java Coding for Main Activity:

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



```
package com.example.devang.exno1;

import ...

public class MainActivity extends AppCompatActivity {

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

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

Code for MainActivity.java:

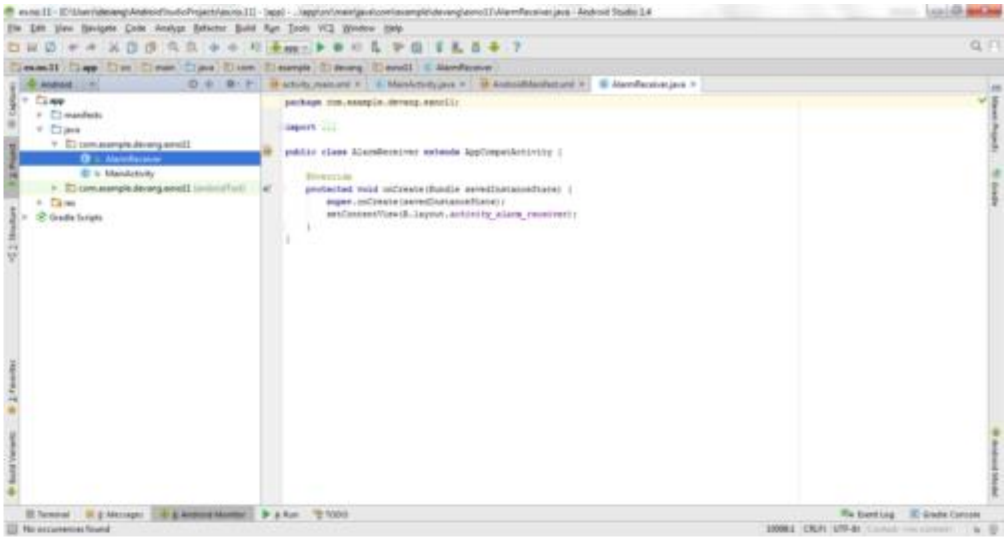
?

```
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.**



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

import ...

public class AlarmReceiver extends AppCompatActivity {

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

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

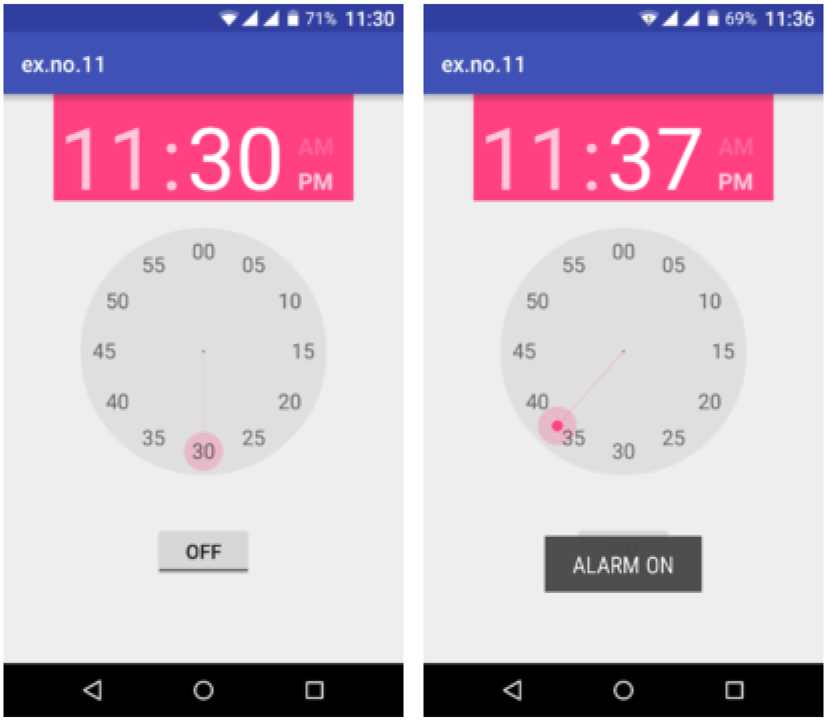
Code for AlarmReceiver.java:

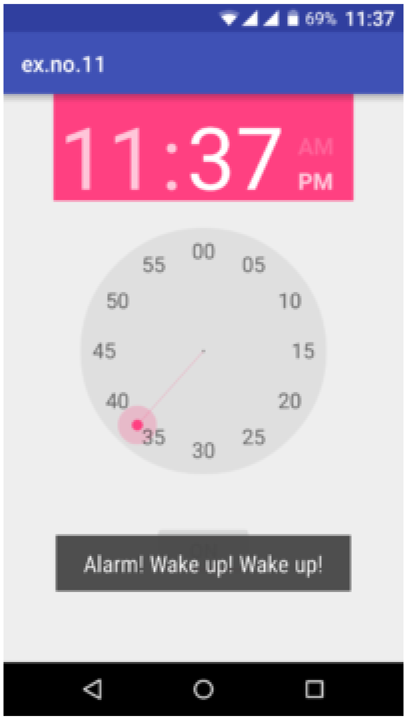
?

```
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();
        }
    }
```

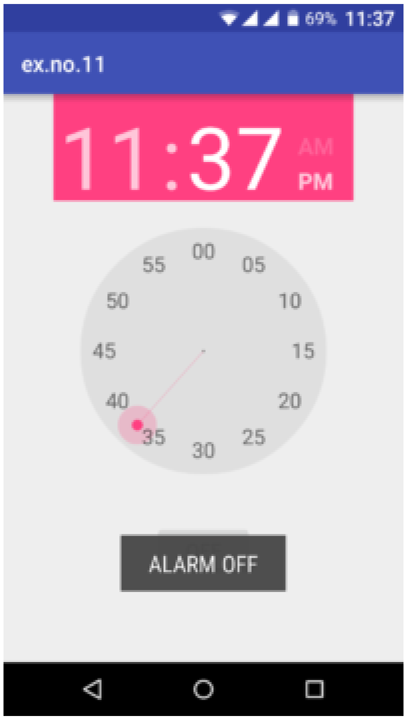
- So now the Coding part of Alarm Receiver is also completed.
- Now run the application to see the output.

Output:





Alarm! Wake up! Wake up!



ALARM OFF

Result:

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