Experiment No.01

Problem Definition: Develop an application that uses GUI components, fonts and colors.

Objectives of the Experiment

1. To understand and implement App developing using Android studio.
2. To implement GUI components ,font and colors.

Theory

Theory:

- The user interface(UI) for an android app is built as a hierarchy of layouts and widgets. The layouts are ViewGroup objects, containers that control how their child views are positioned on the screen. Widgets are view objects, UI components such as buttons and text boxes.
- Android provides an XML vocabulary for ViewGroup and View classes, so most of our UI is defined in XML files. However, rather than teach you to write XML, this shows how to create a layout using Android studio's layout editor. This writes the XML for you as you drag and drop views to build your layout. This helps us to develop an application that uses GUI components.

Source Code

MainActivity.java

```
package com.example.termwork1cdiv;
import androidx.appcompat.app.AppCompatActivity;
import android.graphics.Color;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

   float f=30;
   int ch = 1;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);

TextView t = (TextView) findViewById(R.id.textView);
      Button b1 = (Button) findViewById(R.id.button3);
      Button b2 = (Button) findViewById(R.id.button2);

      b1.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View v) {
t.setTextSize(f);
            f = f + 5;
```

```
            if (f == 50) {
               f = 30;
            }

         }
      });
      b2.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View v) {

            switch (ch){
               case 1:
t.setTextColor(Color.BLUE);
                  break;

               case 2:
t.setTextColor(Color.GREEN);
                  break;

               case 3:
t.setTextColor(Color.BLACK);
                  break;

               case 4:
t.setTextColor(Color.MAGENTA);
                  break;

               case 5:
t.setTextColor(Color.CYAN);
                  break;
            }

ch++;
            if(ch==6){
ch=1;
            }

         }
      });

   }
}
```

XML Code(main.xml)

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayoutxmlns: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=".MainActivity"
android:orientation="vertical"
android:padding="50dp"
```

```
>

<TextView
android:id="@+id/text"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:gravity="center"
android:text="Hello and Welcome!"


android:textSize="30sp"
android:textStyle="bold"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintTop_toTopOf="parent" />


<TextView
android:id="@+id/textView"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="TRY ME"
android:padding="20dp"
android:gravity="center"
tools:layout_editor_absoluteX="172dp"
tools:layout_editor_absoluteY="165dp"
android:textSize="30sp"
android:textStyle="bold"
      />


<Button
android:id="@+id/button2"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="Color Change"
android:padding="15dp"
tools:layout_editor_absoluteX="74dp"
tools:layout_editor_absoluteY="408dp" />

<Button
android:id="@+id/button3"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="Size Change"
android:padding="15dp"
tools:layout_editor_absoluteX="230dp"
tools:layout_editor_absoluteY="408dp" />

</LinearLayout>
```

Colors.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
<color name="purple_200">#FFBB86FC</color>
<color name="purple_500">#E91E63</color>
<color name="purple_700">#F44336</color>
<color name="teal_200">#FFC107</color>
<color name="teal_700">#FF018786</color>
<color name="black">#FF000000</color>
<color name="white">#FFFFFFFF</color>
</resources>
```

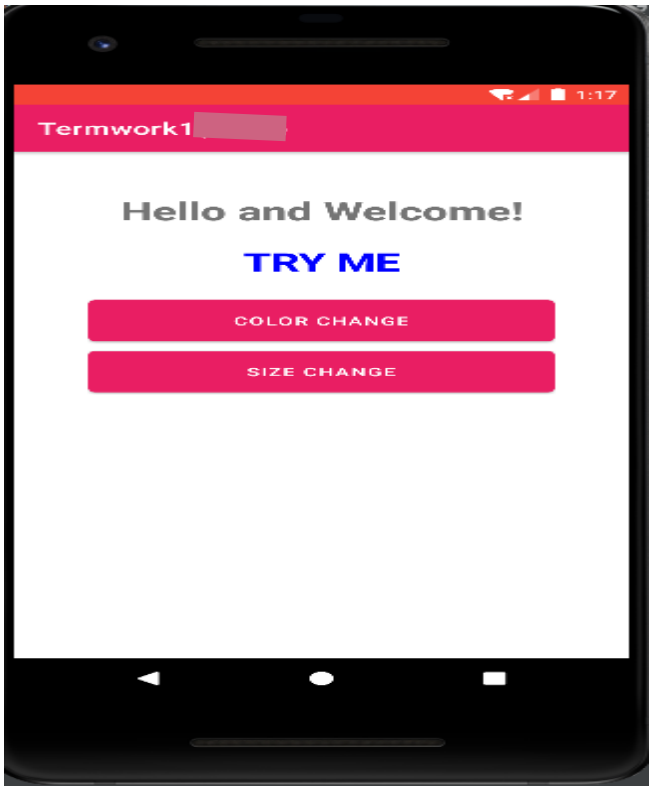
References:

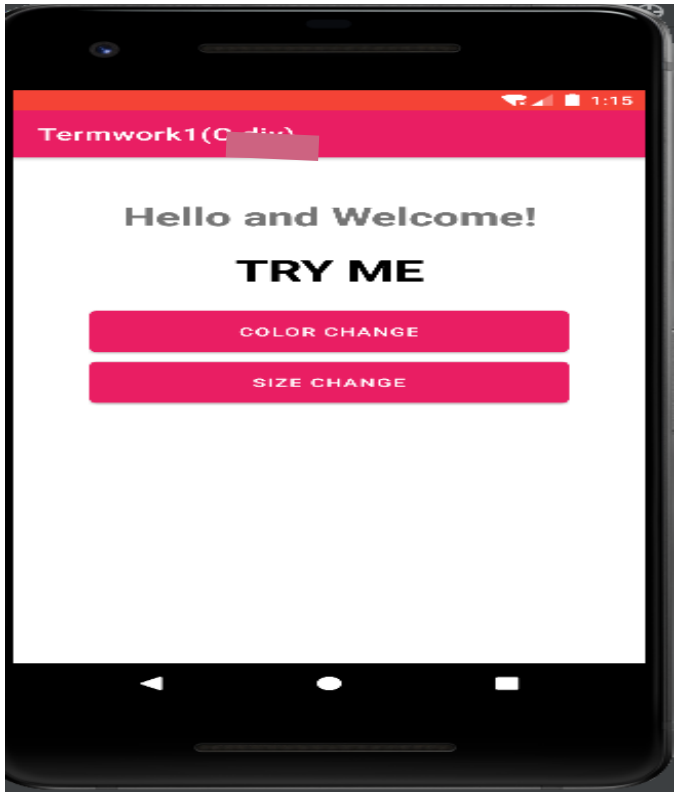
Android Studio 3.5 Development Essentials, Java Edition, 2019 Neil Smyth/ Payload Media, Inc.

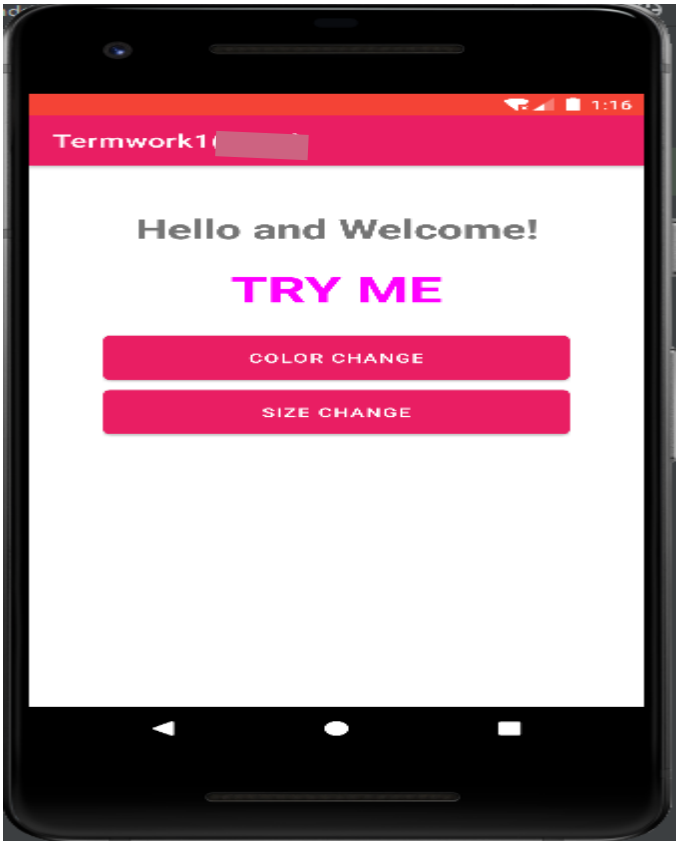
Conclusion:

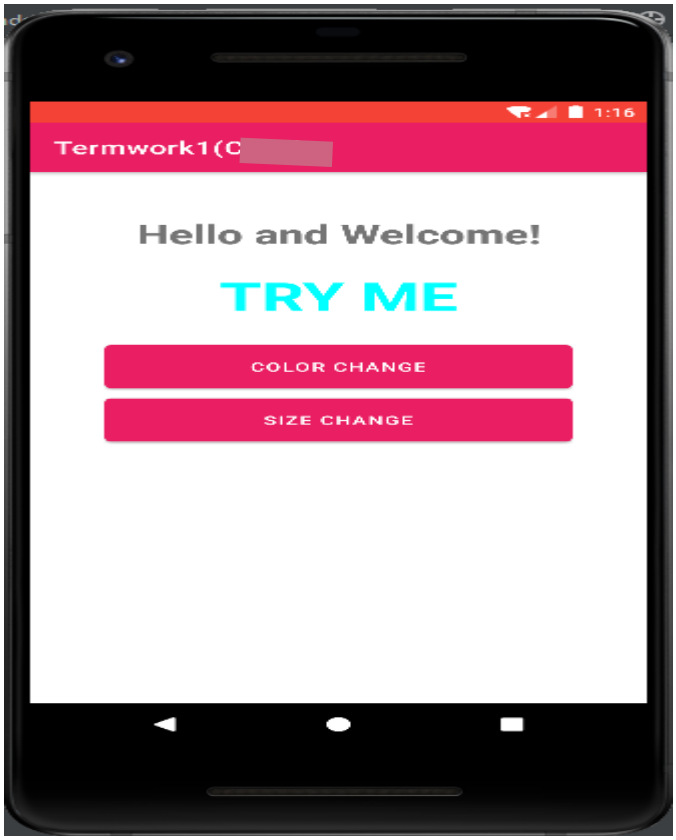
In this termwork, we learnt how to use the GUI components like Font and colours to build an android application.

OUTPUT


1:17
Termwork1
Hello and Welcome!
TRY ME
COLOR CHANGE
SIZE CHANGE


1:15
Hello and Welcome!
TRY ME
COLOR CHANGE
SIZE CHANGE


1:16
Hello and Welcome!
TRY ME
COLOR CHANGE
SIZE CHANGE


1:16
Hello and Welcome!
TRY ME
COLOR CHANGE
SIZE CHANGE

Experiment No.02

Problem Definition: Develop an application that uses layout managers and event listeners.

Objectives of the Experiment

1. To understand and implement different layout manager..
2. To understand the working of event listener.

Theory

Layout: A layout defines the structure for a user interface in your app, such as in an activity. All elements in the layout are built using a hierarchy of View and ViewGroup objects. A View usually draws something the user can see and interact with. Whereas a ViewGroup is an invisible container that defines the layout structure for View and other ViewGroup objects.

Android Layout Types

- Linear Layout : Linear Layout is a view group that aligns all children in a single direction, vertically or horizontally.
- Relative Layout : Relative Layout is a view group that displays child views in relative positions.
- Table Layout : Table Layout is a view that groups views into rows and columns.
- Absolute Layout : Absolute Layout enables you to specify the exact location of its children.
- Frame Layout : The Frame Layout is a placeholder on screen that you can use to display a single view.
- List View: List View is a view group that displays a list of scrollable items.
- Grid View: Grid View is a ViewGroup that displays items in a two-dimensional, scrollable grid.

Some Layout Attributes

Attributes	Description
android:id	This is the ID which uniquely identifies the view.
android:layout_width	This is the width of the layout.
android:layout_height	This is the height of the layout
android:layout_marginBottom	This is the extra space on the bottom side of the layout.
android:layout_marginTop	This is the extra space on the top side of the layout.
android:layout_weight	This specifies how much of the extra space in the layout should be allocated to the View.

Event Listeners : An event listener is an interface in the View class that contains a single callback method. These methods will be called by the Android framework when the View to which the listener has been registered is triggered by user interaction with the item in the UI.

Event Listeners & Event Handlers

Event Handler	Event Listener & Description
onClick()	**OnClickListener()** This is called when the user either clicks or touches or focuses upon any widget like button, text, image etc. You will use onClick() event handler to handle such event.

onLongClick()	**OnLongClickListener()** This is called when the user either clicks or touches or focuses upon any widget like button, text, image etc. for one or more seconds. You will use onLongClick() event handler to handle such event.
onFocusChange()	**OnFocusChangeListener()** This is called when the widget looses its focus ie. user goes away from the view item. You will use onFocusChange() event handler to handle such event.
onKey()	**OnFocusChangeListener()** This is called when the user is focused on the item and presses or releases a hardware key on the device. You will use onKey() event handler to handle such event.
onTouch()	**OnTouchListener()** This is called when the user presses the key, releases the key, or any movement gesture on the screen. You will use onTouch() event handler to handle such event.
onMenuItemClick()	**OnMenuItemClickListener()** This is called when the user selects a menu item. You will use onMenuItemClick() event handler to handle such event.
onCreateContextMenu()	**onCreateContextMenuItemListener()** This is called when the context menu is being built(as the result of a sustained "long click)

Source Code
Activity_main.xml

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

MainActivity.java

```
package com.example.demo1;
import android.content.Intent;
import androidx.appcompat.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);

         }
      });
   }
}
```

Second_activity.xml

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

SecondActivity.java

```
package com.example.demo1;
import android.content.Intent;
import androidx.appcompat.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);

    }
}
```

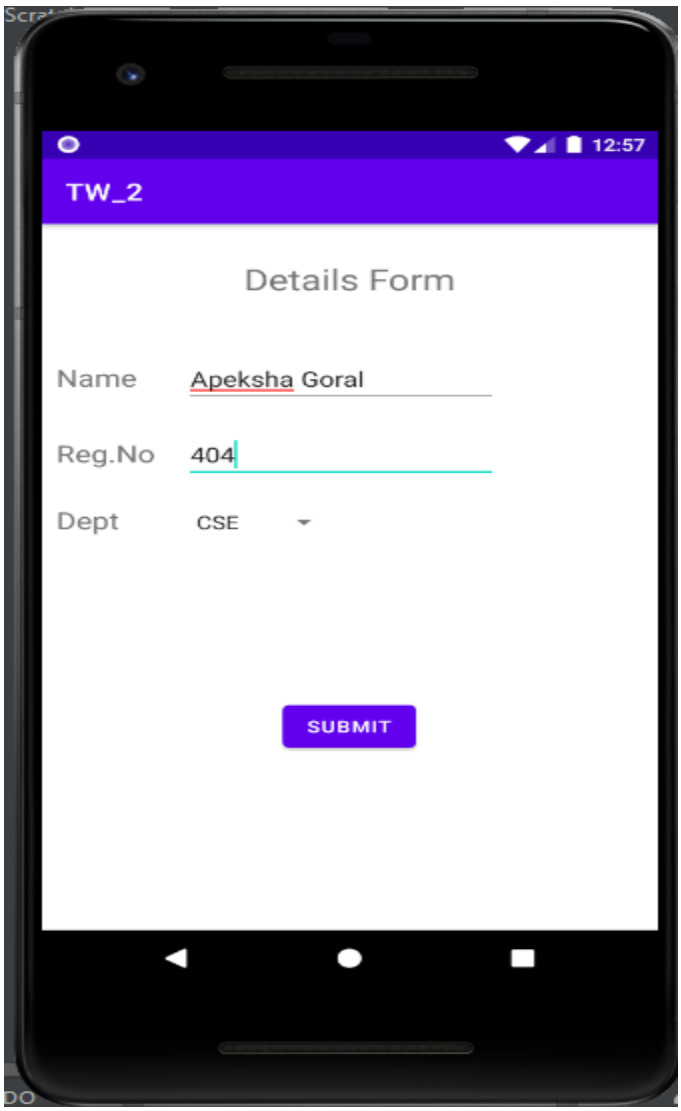
REFERENCES

www.tutorialspoint.com/android/android_event_handling.htm.

CONCLUSION

In this experiment, we learnt how to design an application that uses layout managers and event listeners.

OUTPUT



Experiment No.03

PROBLEM DEFINITION: Develop a native calculator application.

OBJECTIVES OF THE EXPERIMENT

1. To learn how to develop a native calculator application .
2. To learn about android and its activity.

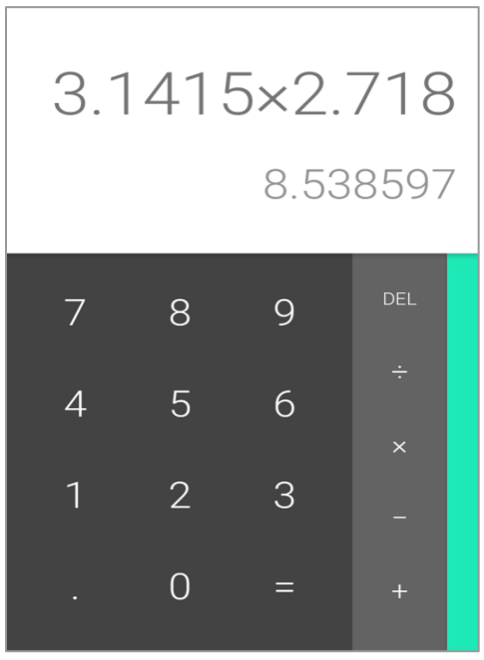
THEORY

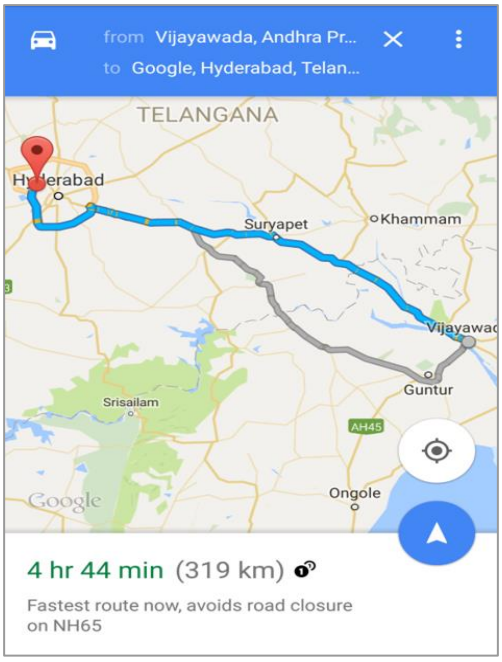
Activity: An activity represents a single screen with a user interface just like window or frame of Java.Android activity is the subclass of ContextThemeWrapper class.

The Activity class defines the following call backs i.e. events. You don't need to implement all the callbacks methods. However, it's important that you understand each one and implement those that ensure your app behaves the way users expect.

Method	Description
onCreate	called when activity is first created.
onStart	called when activity is becoming visible to the user.
onResume	called when activity will start interacting with the user.
onPause	called when activity is not visible to the user.
onStop	called when activity is no longer visible to the user.
onRestart	called after your activity is stopped, prior to start.
onDestroy	called before the activity is destroyed

Examples of activities





SOURCE CODE

Activity_main.xml

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

<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:layout_x="8dp"
android:layout_y="5dp"
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:layout_x="3dp"
android:layout_y="379dp"
android:inputType="numberDecimal"
android:textSize="20sp" />

</LinearLayout>

<LinearLayout
android:id="@+id/linearLayout2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_margin="40dp"
android:orientation="horizontal"
android:rotationX="-4">

<Button
android:id="@+id/Add"
android:layout_width="100dp"
android:layout_height="36dp"
android:layout_weight="1"
```

```
android:layout_x="15dp"
android:layout_y="28dp"
android:text="Add"
android:textSize="14sp" />


<Button
android:id="@+id/Sub"
android:layout_width="100dp"
android:layout_height="36dp"
android:layout_weight="1"
android:layout_x="-2dp"
android:layout_y="66dp"
android:text="Sub"
android:textSize="14sp" />

<Button
android:id="@+id/Mul"
android:layout_width="100dp"
android:layout_height="36dp"
android:layout_weight="1"
android:layout_x="-6dp"
android:layout_y="103dp"
android:text="Mul"
android:textSize="14sp" />

<Button
android:id="@+id/Div"
android:layout_width="100dp"
android:layout_height="36dp"
android:layout_weight="1"
android:layout_x="19dp"
android:layout_y="152dp"
android:text="Div"
android:textSize="14sp" />
</LinearLayout>

<TextView
android:id="@+id/textView"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="Answer is"
android:textAlignment="center"
android:textSize="24sp" />
</LinearLayout>
```

MainActivity.java

```
package com.example.calculator;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.os.Bundle;
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);
    }
}
```

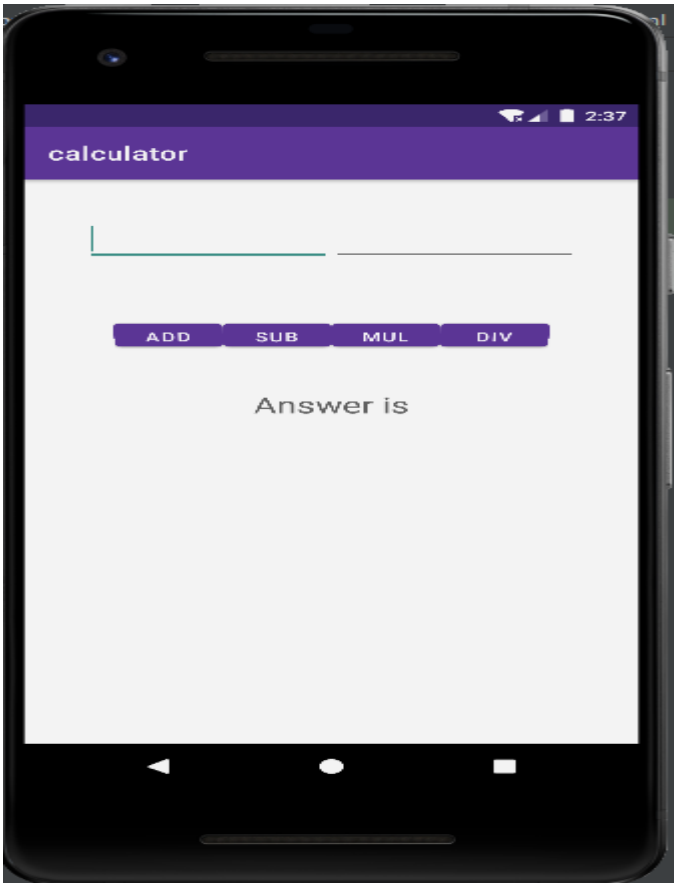
REFERENCE

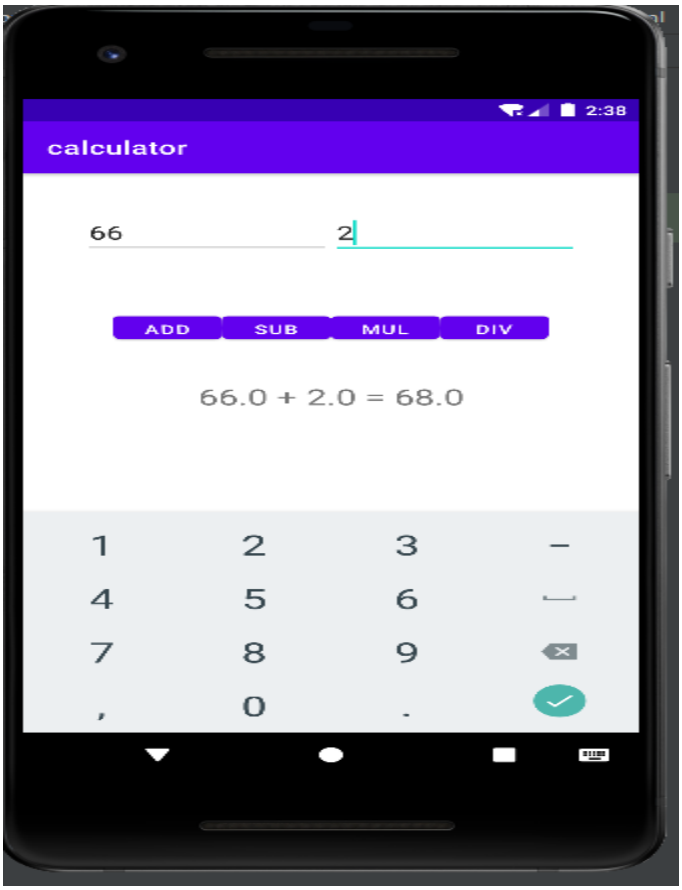
Android Studio 3.5 Development Essentials, Java Edition, 2019 Neil Smyth/ Payload Media, Inc.

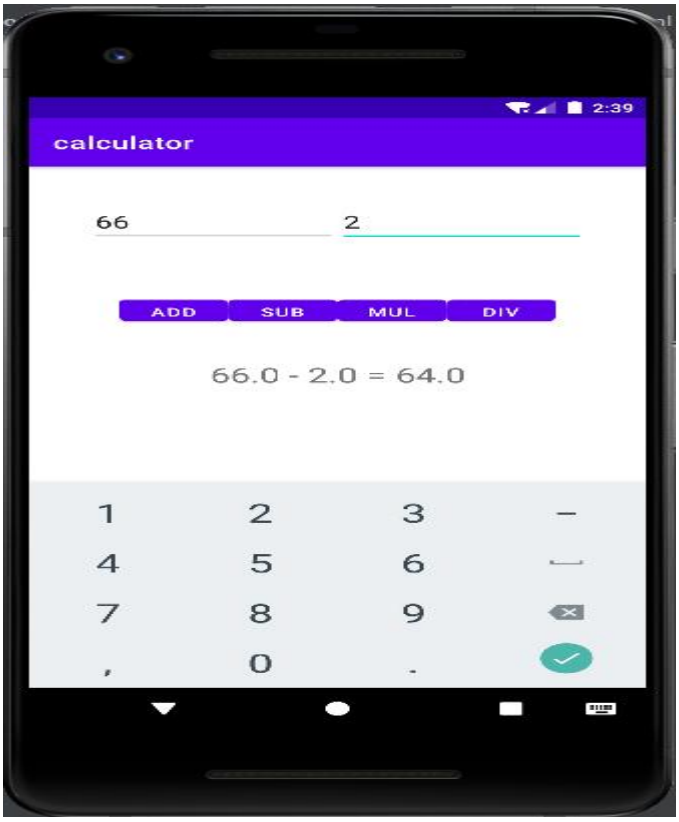
CONCLUSION

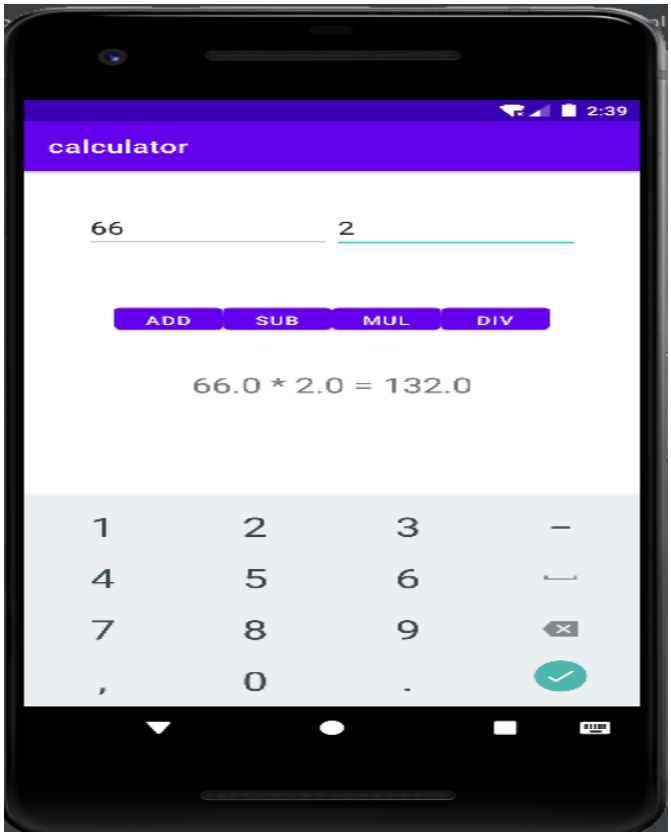
In this experiment,we have learnt to build a native calculator application.

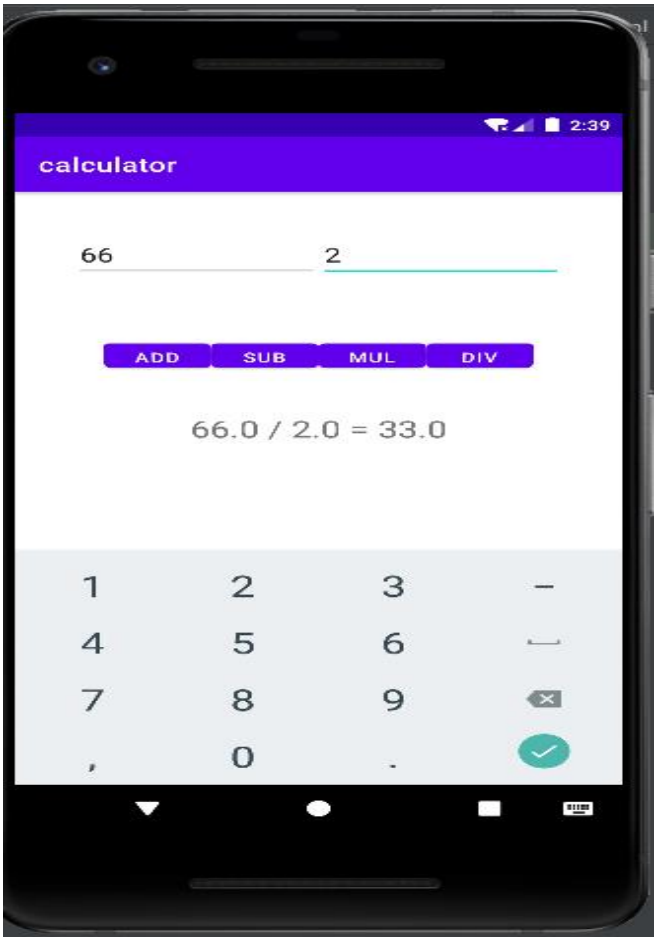
OUTPUT


2:37
calculator
ADD
SUB
MUL
DIV
Answer is


2:38
calculator
66
2
ADD
SUB
MUL
DIV
66.0 + 2.0 = 68.0


2:39
calculator
66
2
ADD
SUB
MUL
DIV
66.0 - 2.0 = 64.0


2:39
calculator
66
2
ADD
SUB
MUL
DIV
66.0 * 2.0 = 132.0


2:39
calculator
66
2
ADD
SUB
MUL
DIV
66.0 / 2.0 = 33.0
1
2
3
4
5
6
7
8
9
,
0
.

Experiment No.04

PROBLEM DEFINITION: Develop an application that make use of database.

OBJECTIVES OF THE EXPERIMENT

1. To learn how to develop an android application that make use of database.
2. To introduce SQLite & Ionic framework..

THEORY

- Mostly in Andrid Programming storing data into database is not common practicebut a very strong tool used by android to communicate with a database is SQLite.
- It is an open source database for structured data in relational database.

SOURCE CODE

MainActivity.java

```
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
{
EditTextRollno,Name,Marks;
  Button Insert,Delete,Update,View,ViewAll;
SQLiteDatabasedb;
  /** 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(rollnoVARCHAR,nameVARCHAR,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();
   }
}
```

activity_main.xml

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

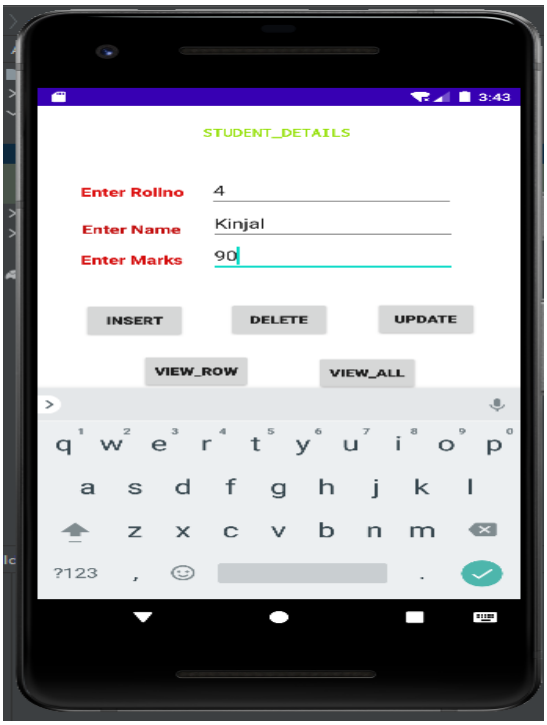
REFERENCE

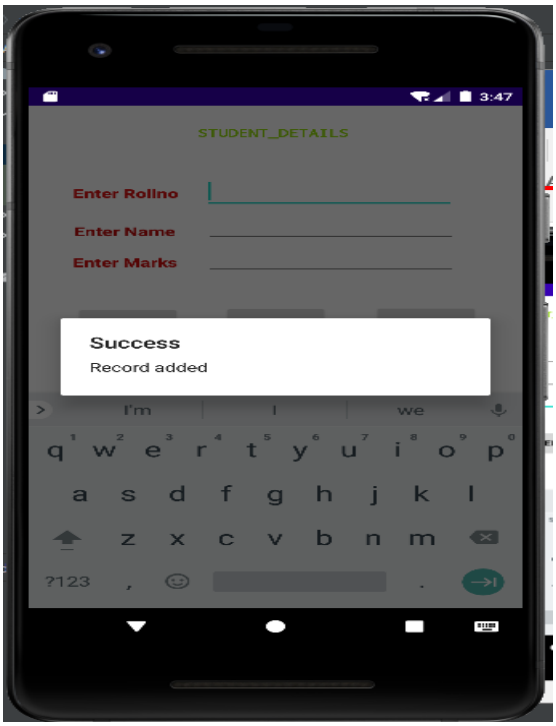
Android Studio 3.5 Development Essentials, Java Edition, 2019 Neil Smyth/ Payload Media, Inc.

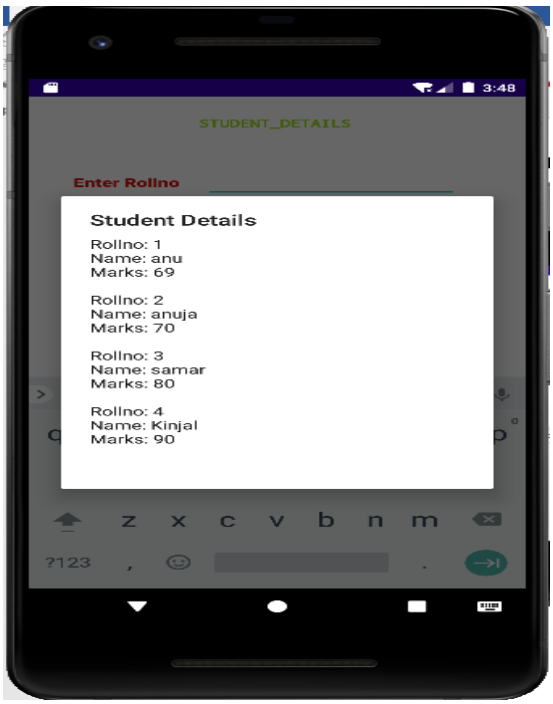
CONCLUSION

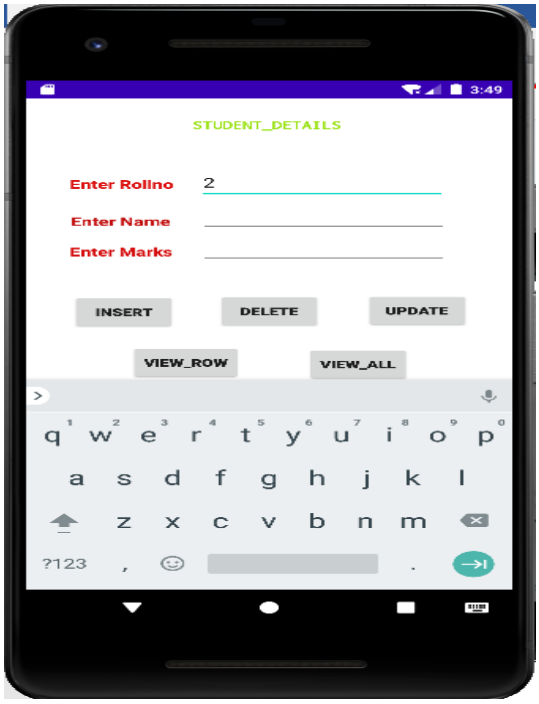
In this experiment, we learnt how to design an application that makes use of database.

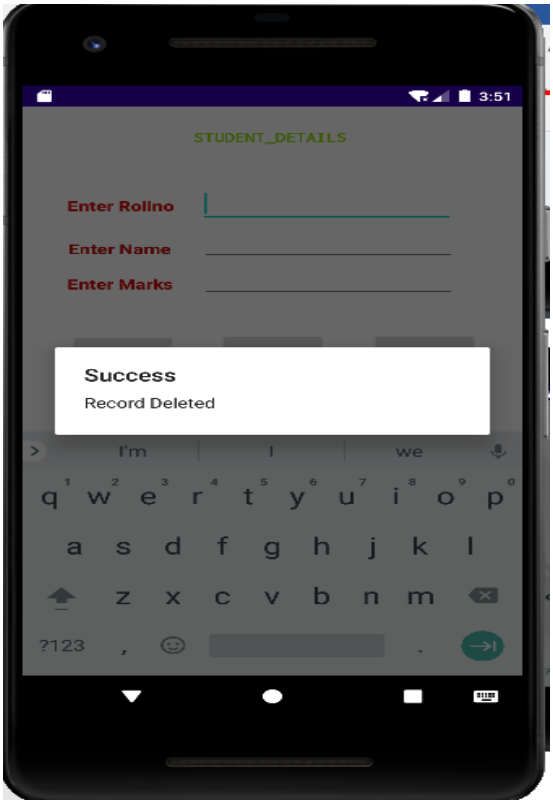
OUTPUT

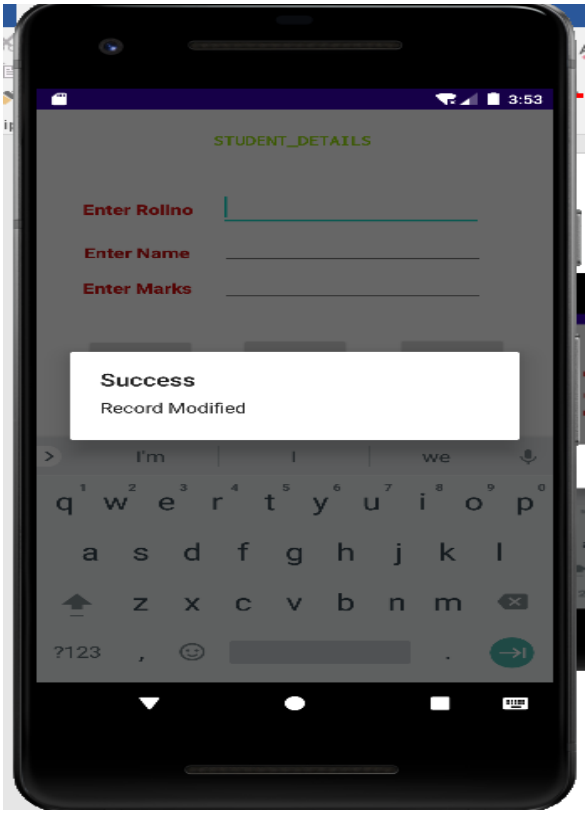





3:48
STUDENT_DETAILS
Enter Rollno
Student Details
Rollno: 1
Name: anu
Marks: 69
Rollno: 2
Name: anuja
Marks: 70
Rollno: 3
Name: samar
Marks: 80
Rollno: 4
Name: Kinjal
Marks: 90


3:49
STUDENT_DETAILS
Enter Rollno
2
Enter Name
Enter Marks
INSERT
DELETE
UPDATE
VIEW_ROW
VIEW_ALL


3:51
STUDENT_DETAILS
Enter Rollno
Enter Name
Enter Marks
Success
Record Deleted


3:53
STUDENT_DETAILS
Enter Rollno
Enter Name
Enter Marks
Success
Record Modified

Experiment No.05

PROBLEM DEFINITION: Develop an application that make use of notification.

OBJECTIVES OF THE EXPERIMENT

1. To understand how to develop an application that make use of notification.
2. To develop activity life cycle, views, layouts and events.

THEORY

Notification: A notification is a message you can display to the user outside of your application's normal UI. When you tell the system to issue a notification, it first appears as an icon in the notification area. To see the details of the notification, the user opens the notification drawer. Both the notification area and the notification drawer are system-controlled areas that the user can view at any time.

The NotificationCompat.Builder Class

Sl.No.	Constants & Description
1	**Notification build()** Combine all of the options that have been set and return a new Notification object.
2	**NotificationCompat.Builder setAutoCancel (boolean autoCancel)** Setting this flag will make it so the notification is automatically canceled when the user clicks it in the panel.
3	**NotificationCompat.Builder setContent (RemoteViews views)** Supply a custom RemoteViews to use instead of the standard one.
4	**NotificationCompat.Builder setContentInfo (CharSequence info)** Set the large text at the right-hand side of the notification.
5	**NotificationCompat.Builder setContentIntent (PendingIntent intent)** Supply a PendingIntent to send when the notification is clicked.
6	**NotificationCompat.Builder setContentText (CharSequence text)** Set the text (second row) of the notification, in a standard notification.
7	**NotificationCompat.Builder setContentTitle (CharSequence title)** Set the text (first row) of the notification, in a standard notification.
8	**NotificationCompat.Builder setDefaults (int defaults)** Set the default notification options that will be used.
9	**NotificationCompat.Builder setLargeIcon (Bitmap icon)** Set the large icon that is shown in the ticker and notification.
10	**NotificationCompat.Builder setNumber (int number)** Set the large number at the right-hand side of the notification.
11	**NotificationCompat.Builder setOngoing (boolean ongoing)** Set whether this is an ongoing notification.
12	**NotificationCompat.Builder setSmallIcon (int icon)** Set the small icon to use in the notification layouts.

SOURCE CODE

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayoutxmlns: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=".MainActivity">

<Button
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:id="@+id/notify_btn"
android:shadowColor="@color/teal_200"
android:text="Click to get notification"/>

</RelativeLayout>
```

MainActivity.java

```
package com.example.notificationdemo1;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.NotificationCompat;
import androidx.core.app.NotificationManagerCompat;
import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.Intent;
import android.os.Build;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import java.nio.channels.Channel;

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

notifyBtn = findViewById(R.id.notify_btn);

notifyBtn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
NotificationCompat.Builder builder = new NotificationCompat.Builder(MainActivity.this, "My Notification");
builder.setContentTitle("My Notification");
builder.setContentText("This is the notification you received...");
builder.setAutoCancel(true);
builder.setSmallIcon(R.drawable.ic_launcher_background);
builder.setPriority(NotificationCompat.PRIORITY_DEFAULT);
```

```
        Intent intent = new Intent(MainActivity.this, NotificationActivity.class);
intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
intent.putExtra("message","This is the notification you received...");

PendingIntentpendingIntent =
PendingIntent.getActivity(MainActivity.this,0,intent,PendingIntent.FLAG_UPDATE_CURRENT);
builder.setContentIntent(pendingIntent);


NotificationManagerCompatmanagerCompat = NotificationManagerCompat.from(MainActivity.this);
managerCompat.notify(1, builder.build());
        }
    });
  }
}
```

XML CODE

```
<?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=".NotificationActivity">

<TextView
android:id="@+id/text_view"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:textSize="26sp"/>
</RelativeLayout>
```

JAVA CODE

```
package com.example.notificationdemo1;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.TextView;

public class NotificationActivity extends AppCompatActivity {

  @Override
  protected void onCreate(Bundle savedInstanceState){
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_notification);
TextViewtextView = findViewById(R.id.text_view);
String message = getIntent().getStringExtra("message");
textView.setText(message);

  }
}
```

Reference

https://www.tutorialspoint.com/android/android_notifications.htm.

Conclusion

In this experiment, we learnt how to make use of a notification to build an android application.

OUTPUT