CS8662 MOBILE APPLICATION DEVELOPMENT LABORATORY

LIST OF EXPERIMENTS:

1. Develop an application that uses GUI components, Font, Layout Managers and event listeners.
2. Develop an application that makes use of databases
3. Develop a native application that uses GPS location information
4. Implement an application that creates an alert upon receiving a message
5. Develop an application that makes use of RSS Feed.
6. Create an application using Sensor Manager
7. Develop a Mobile application for simple and day to day needs (Mini Project)
8. Create an android application that converts the user input text to voice.

EX. NO. : 1	Develop an application that uses GUI components, Font and Colors
DATE :	

AIM:

To develop an android application that uses GUI Components, Font and colors.

ALGORITHM:

1. Create a New Android Project:
   - Click New   in the toolbar.
   - In the window that appears, open the Android folder, select Android Application Project, and click next.
   - Provide the application name and the project name and then finally give the desired package name.
   - Choose a launcher icon for your application and then select Blank Activity and then click Next
   - Provide the desired Activity name for your project and then click Finish.
2. Create a New AVD (Android Virtual Device):
   - click Android Virtual Device Manager from the toolbar.
   - In the Android Virtual Device Manager panel, click New.
   - Fill in the details for the AVD. Give it a name, a platform target, an SD card size, and a skin (HVGA is default).
   - Click Create AVD and Select the new AVD from the Android Virtual Device Manager and click Start.
3. Design the graphical layout with a text view and two command buttons.
4. Run the application.
5. On pressing the change color button, color of the text gets changed.
6. On pressing the change font size button, the size of the font gets altered.
7. Close the Android project.

PROGRAM CODE

MainActivity.java

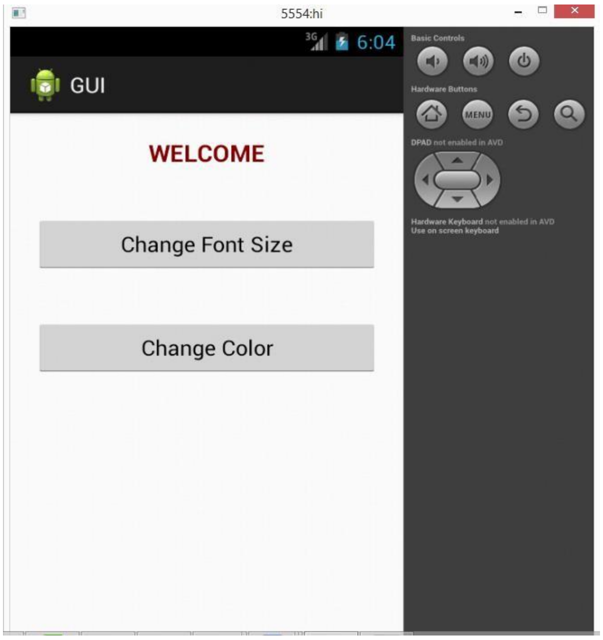
```
packagecom.example.gui;
importandroid.os.Bundle;
importandroid.app.Activity;
importandroid.graphics.Typeface;
importandroid.graphics.Color;
importandroid.view.View;
importandroid.widget.Button;
importandroid.widget.TextView;

publicclassMainActivityextends Activity {
float font = 24;
inti = 1;
```

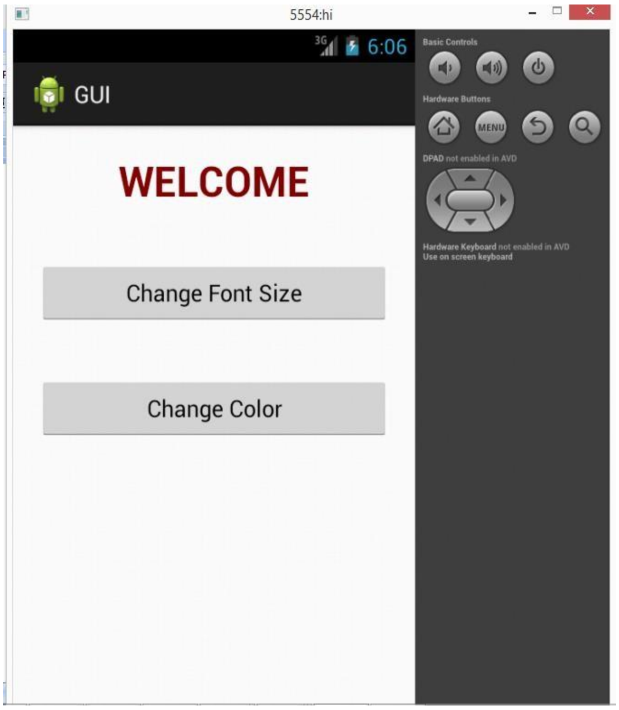
```
        protectedvoidonCreate(Bundle savedInstanceState) {
                super.onCreate(savedInstanceState);
                setContentView(R.layout.activity_main);
                finalTextView t1 = (TextView)findViewById(R.id.textView1);
                Button b1 = (Button)findViewById(R.id.button1);
                b1.setOnClickListener(newView.OnClickListener() {

                                publicvoidonClick(View view) {
                                t1.setTextSize(font);
                                font = font+4;
                                if(font==40)
                                        font = 20;
                }
                });
                Button b2 = (Button)findViewById(R.id.button2);
                b2.setOnClickListener(newView.OnClickListener() {

                        publicvoidonClick(View view) {
                                switch(i)
                                {
                                case 1:
                                t1.setTextColor(Color.parseColor("#0000FF"));
                                break;
                                case 2:
                                        t1.setTextColor(Color.parseColor("#00FF00"));
                                        break;
                        case 3:
                                        t1.setTextColor(Color.parseColor("#FF0000"));
                                        break;
                        case 4:
                                        t1.setTextColor(Color.parseColor("#800000"));
                                        break;

                        }
                        i++;
                        if(i==5)
        i = 1;
                     }
                });
   }
}
```

activity_main.xml

```
<LinearLayoutxmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="fill_parent"
android:layout_height="fill_parent"
android:orientation="vertical"
>

<TextView
android:id="@+id/textView1"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="WELCOME"
android:layout_margin="20sp"
android:gravity="center"
android:textSize="20sp"
android:textStyle="bold"
        />

<Button
android:id="@+id/button1"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_margin="20sp"
android:gravity="center"
android:text="Change Font Size" />

<Button
android:id="@+id/button2"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:gravity="center"
android:layout_margin="20sp"
android:text="Change Color" />

</LinearLayout>
```

OUTPUT:





RESULT:

Thus, the program for android application GUI Components, Font and colors was executed successfully.

EX. NO:2 **DATE :**	**Develop an application that makes use of database**

AIM:

To develop an android application that makes use of database.

ALGORITHM:

1. Create a New Android Project:
   - Click New in the toolbar.
   - In the window that appears, open the Android folder, select Android Application Project, and click next.
   - Provide the application name and the project name and then finally give the desired package name.
   - Choose a launcher icon for your application and then select Blank Activity and then click Next
   - Provide the desired Activity name for your project and then click Finish.
2. Create a New AVD (Android Virtual Device):
   - click Android Virtual Device Manager from the toolbar.
   - In the Android Virtual Device Manager panel, click New.
   - Fill in the details for the AVD. Give it a name, a platform target, an SD card size, and a skin (HVGA is default).
   - Click Create AVD and Select the new AVD from the Android Virtual Device Manager and click Start.
3. Design the graphical layout.
4. Run the application.
5. Perform the database operation.(Insert/delete/view/update)
6. Close the Android project.

PROGRAM CODE

MainActivity.java

```
package com.example.admin.myapplication;

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 MainActivityextends 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();
    }
}
```

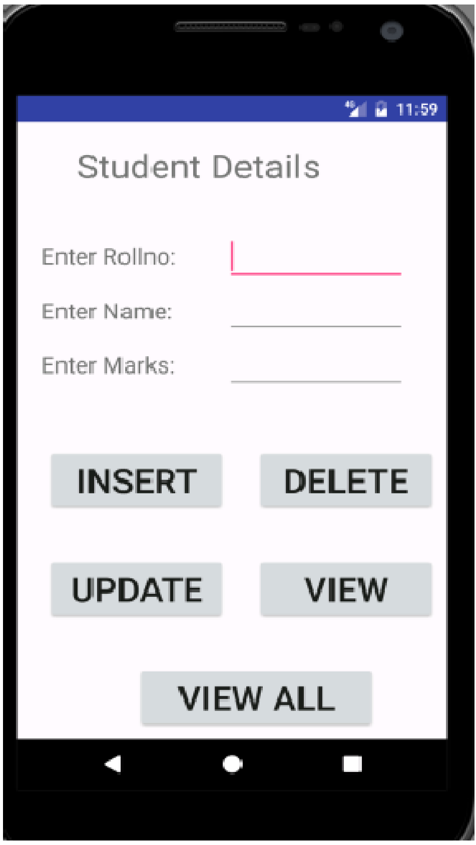
activity_main.xml

```
<?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>
```

OUTPUT:

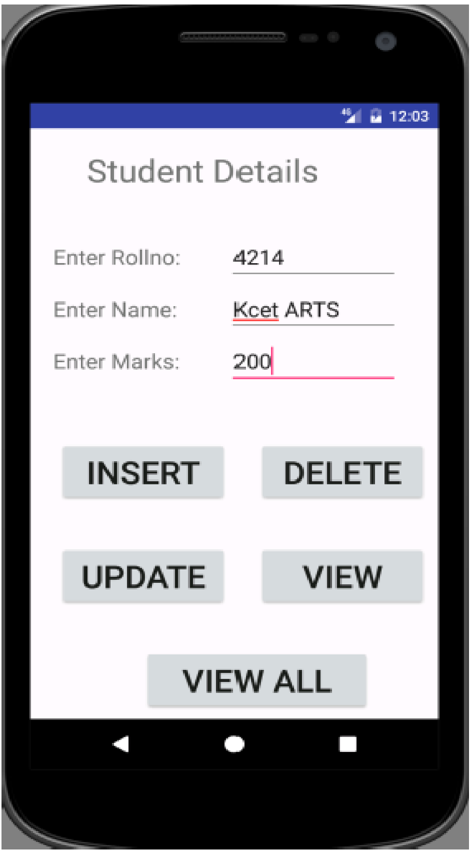


Student Details

Enter Rollno:

Enter Name:

Enter Marks:

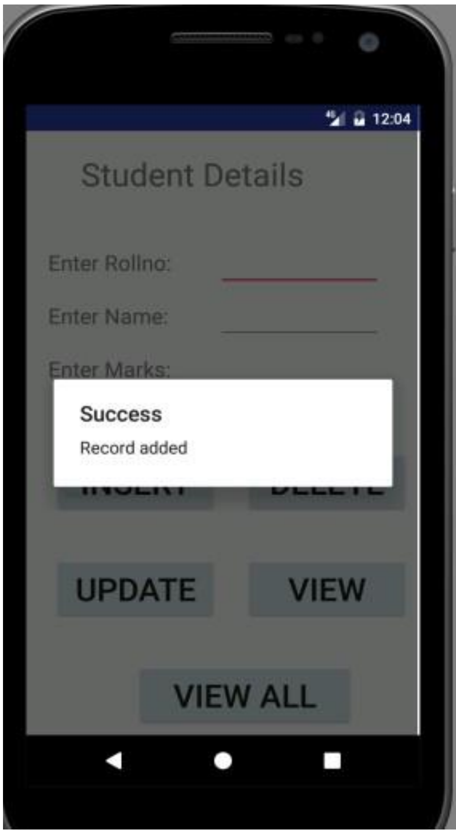


Student Details

Enter Rollno: 4214

Enter Name: Kcet ARTS

Enter Marks: 200



Student Details

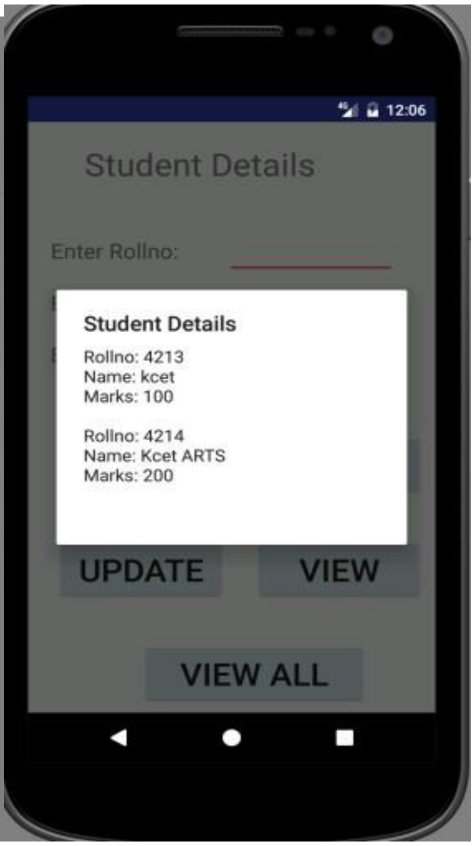
Enter Rollno:

Enter Name:

Enter Marks:

Success

Record added



Student Details

Enter Rollno:

Student Details

Rollno: 4213
Name: kcet
Marks: 100

Rollno: 4214
Name: Kcet ARTS
Marks: 200

RESULT:

Thus, the program for android application that makes use of database was executed successfully

EX. NO. :3	Develop a native application that uses GPS location information
DATE :	

AIM:

To develop an android application that uses GPS location information.

ALGORITHM:

1. Create a New Android Project:
   - Click New   in the toolbar.
   - In the window that appears, open the Android folder, select Android Application Project, and click next.
   - Provide the application name and the project name and then finally give the desired package name.
   - Choose a launcher icon for your application and then select Blank Activity and then click Next
   - Provide the desired Activity name for your project and then click Finish.
2. Create a New AVD (Android Virtual Device):
   - click Android Virtual Device Manager from the toolbar.
   - In the Android Virtual Device Manager panel, click New.
   - Fill in the details for the AVD. Give it a name, a platform target, an SD card size, and a skin (HVGA is default).
   - Click Create AVD and Select the new AVD from the Android Virtual Device Manager and click Start.
3. Design the graphical layout.
4. Run the application.
5. The requested data is retrieved from the database named myFriendsDb.
6. Close the Android project.

PROGRAM CODE

UseGps.java

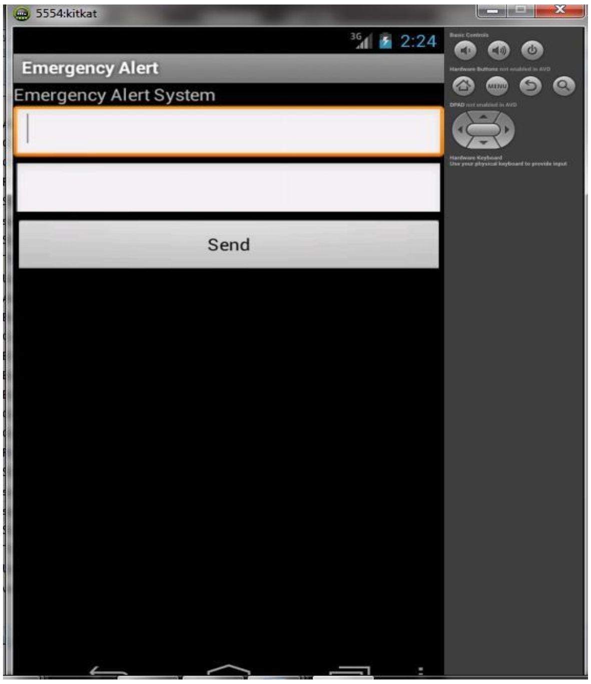
```
package com.emergency;
import android.app.Activity;
import android.content.Context;
import android.location.Location;
import  android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
```

```
public class UseGps extends Activity
{
        Button buttonSend;
        EditTexttextSMS;
        EditTexttextlon;
public void onCreate(Bundle savedInstanceState)
{
super.onCreate(savedInstanceState);
setContentView(R.layout.main);
buttonSend = (Button) findViewById(R.id.buttonSend);
textSMS = (EditText) findViewById(R.id.editTextSMS);
textlon = (EditText) findViewById(R.id.textlon);
LocationManagermlocManager                                                    =
(LocationManager)getSystemService(Context.LOCATION_SERVICE);
LocationListenermlocListener = new MyLocationListener();
mlocManager.requestLocationUpdates( LocationManager.GPS_PROVIDER, 0, 0, mlocListener);
}
public class MyLocationListener implements LocationListener
{
public void onLocationChanged(Location loc)
{
loc.getLatitude();
loc.getLongitude();
Double lat=loc.getLatitude();
Double lon=loc.getLongitude();
textSMS.setText(lat.toString());
textlon.setText(lon.toString());
}
public void onProviderDisabled(String provider)
{
Toast.makeText( getApplicationContext(),"Gps Disabled",Toast.LENGTH_SHORT ).show();
}
public void onProviderEnabled(String provider)
{
Toast.makeText( getApplicationContext(), "Gps Enabled", Toast.LENGTH_SHORT).show();
}
}
}
```

main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayoutxmlns:android="http://schemas.android.com/apk/res/android"
android:orientation="vertical"
android:layout_width="fill_parent"
android:layout_height="fill_parent"
>
<TextView
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:text="Emergency Alert System"
    />

<EditText
android:id="@+id/editTextSMS"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:gravity="top" />

<EditText
android:id="@+id/textlon"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:gravity="top" />

<Button
android:id="@+id/buttonSend"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:text="Send" />

</LinearLayout>
```

OUTPUT:



RESULT:

Thus, the program for android application that makes use of GPS information was executed successful

EX. NO. :4	**Implement an application that creates an alert upon receiving a message**
DATE :	

AIM:

To develop an android application that creates an alert upon receiving a message.

ALGORITHM:

1. Create a New Android Project:
   - Click New in the toolbar.
   - In the window that appears, open the Android folder, select Android Application Project, and click next.
   - Provide the application name and the project name and then finally give the desired package name.
   - Choose a launcher icon for your application and then select Blank Activity and then click Next
   - Provide the desired Activity name for your project and then click Finish.
2. Create a New AVD (Android Virtual Device):
   - click Android Virtual Device Manager from the toolbar.
   - In the Android Virtual Device Manager panel, click New.
   - Fill in the details for the AVD. Give it a name, a platform target, an SD card size, and a skin (HVGA is default).
   - Click Create AVD and Select the new AVD from the Android Virtual Device Manager and click Start.
3. Design the layout by adding a text box and a command button.
4. Run the application.
5. If the entered E-mail doesn’t match the given E-mail id, then an alert will be displayed.
6. If the entered E-mail id matches with the provided mail-id then login is successful.
7. Close the Android project.

PROGRAM CODE:

MainActivity.java

```
package com.pa.Alert;
import android.os.Bundle;
import android.app.Activity;
import android.content.Intent;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
```

```
public class MainActivity extends Activity {
        private Button BTN;
        private EditText email;

        protected void onCreate(Bundle savedInstanceState) {
                super.onCreate(savedInstanceState);
                setContentView(R.layout.activity_main);
                BTN = (Button) findViewById(R.id.btn);
                email = (EditText) findViewById(R.id.emailInput);
                BTN.setOnClickListener(new OnClickListener() {
                     public void onClick(View v) {
                          String val = email.getText().toString();
                            if (val == null || val.length() <= 0) {
                                Toast.makeText(getApplicationContext(),
                                        "Please Enter the email", Toast.LENGTH_LONG).show();
                            } else if (val.equals("enpboss@gmail.com")) {
                                        Intent intent = new Intent(getApplicationContext(),
                                                SecondActivity.class);
                                        startActivity(intent);
                                        Toast.makeText(getApplicationContext(),
                                                "Login Success", Toast.LENGTH_LONG).show();
                             } else {
                                Toast.makeText(getApplicationContext(),
                                "Please Enter valid email", Toast.LENGTH_LONG)
                                                        .show();
                                }
                        }
                });
        }
}
```

SecondActivity.java

```
package com.pa.Alert;

import android.app.Activity;
import android.os.Bundle;

public class SecondActivity extends Activity {
        @Override
        protected void onCreate(Bundle savedInstanceState) {
                // TODO Auto-generated method stub
                super.onCreate(savedInstanceState);
                setContentView(R.layout.second_activity);
        }
}
```

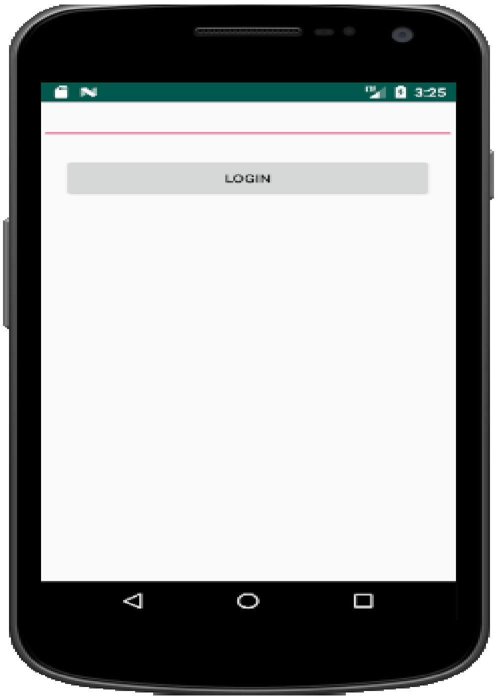
Main_activity.xml

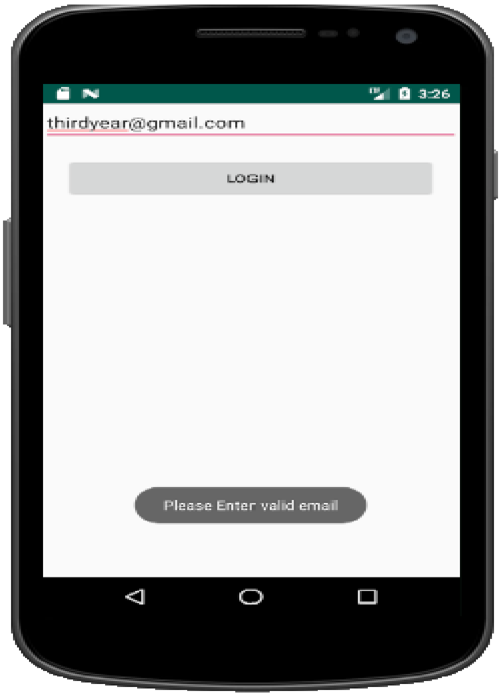
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayoutxmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical" >

<EditText
android:id="@+id/emailInput"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:ems="10" />

<Button
android:id="@+id/btn"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_margin="20sp"
android:gravity="center"
android:text="Login" />

</LinearLayout>
```

AndroidMainfest.Xml

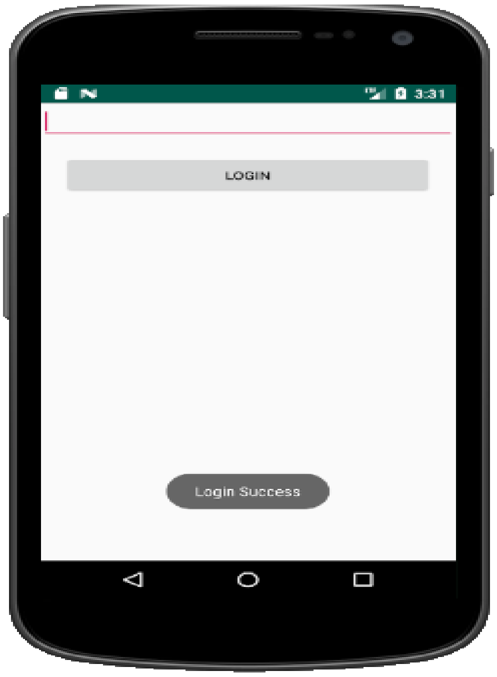
```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
   package="com.example.admin.myapplication">

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

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

OUTPUT:




thirdyear@gmail.com

Please Enter valid email



Login Success

RESULT:

Thus, the program for android application that creates an alert upon receiving a message was executed successfully.

EX. NO. :5	**Develop an application that makes use of RSS Feed**
DATE :	

AIM:

To develop an android application that makes use of RSS (Rich Site Summary) Feed.

ALGORITHM:

10. Create a New Android Project:
    - Click New in the toolbar.
    - In the window that appears, open the Android folder, select Android Application Project, and click next.
    - Provide the application name and the project name and then finally give the desired package name.
    - Choose a launcher icon for your application and then select Blank Activity and then click Next
    - Provide the desired Activity name for your project and then click Finish.
11. Create a New AVD (Android Virtual Device):
    - click Android Virtual Device Manager from the toolbar.
    - In the Android Virtual Device Manager panel, click New.
    - Fill in the details for the AVD. Give it a name, a platform target, an SD card size, and a skin (HVGA is default).
    - Click Create AVD and Select the new AVD from the Android Virtual Device Manager and click Start.
12. Design the graphical layout using buttons, text and ImageView.
13. Run the application.
14. Fetch the details from the URL.
15. Display the output by clicking the result button.
16. Close the Android project.

PROGRAM CODE:

MainActivity.java

```
package com.example.admin.myapplication;

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

public class MainActivityextends ListActivity
{
    List headlines;
    List links;

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

class MyAsyncTaskextends AsyncTask<Object,Void,ArrayAdapter>
    {
        @Override
protected ArrayAdapterdoInBackground(Object[] params)
        {
headlines = new ArrayList();
links = new ArrayList();
try
{
                URL url = new URL("https://codingconnect.net/feed");

XmlPullParserFactory factory = XmlPullParserFactory.newInstance();
factory.setNamespaceAware(false);
XmlPullParserxpp = factory.newPullParser();

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

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

```
if (insideItem)
links.add(xpp.nextText()); //extract the link of article
}
                    }
else if(eventType==XmlPullParser.END_TAG&&xpp.getName().equalsIgnoreCase("item"))
                    {
insideItem=false;
                    }
eventType = xpp.next(); //move to next element
}

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

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

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

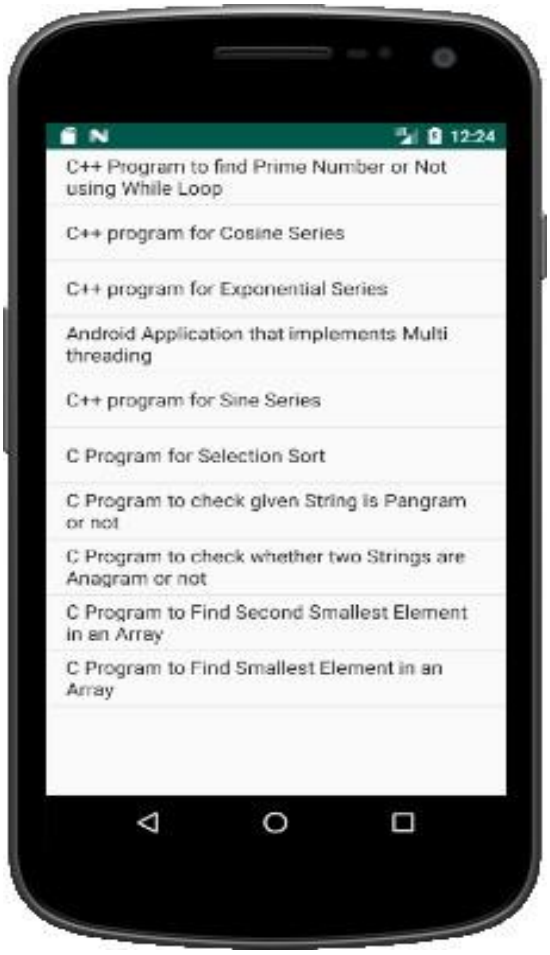
AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.exno10" >



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

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

</manifest>
```

activity_main.xml

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

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

</LinearLayout>
```

OUTPUT:



C++ Program to find Prime Number or Not using While Loop

C++ program for Cosine Series

C++ program for Exponential Series

Android Application that implements Multi threading

C++ program for Sine Series

C Program for Selection Sort

C Program to check given String is Pangram or not

C Program to check whether two Strings are Anagram or not

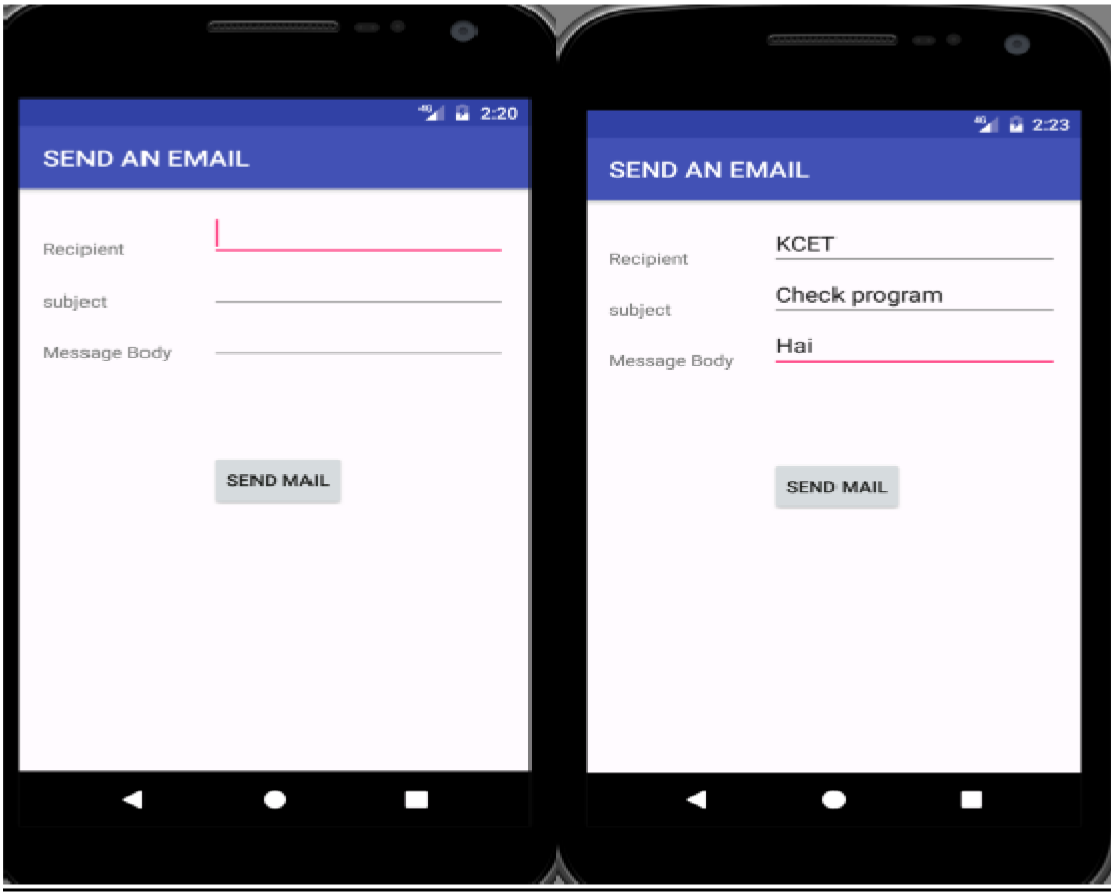
C Program to Find Second Smallest Element in an Array

C Program to Find Smallest Element in an Array

RESULT:

Thus, the program for android application that makes use of RSS Feed was executed successfully.

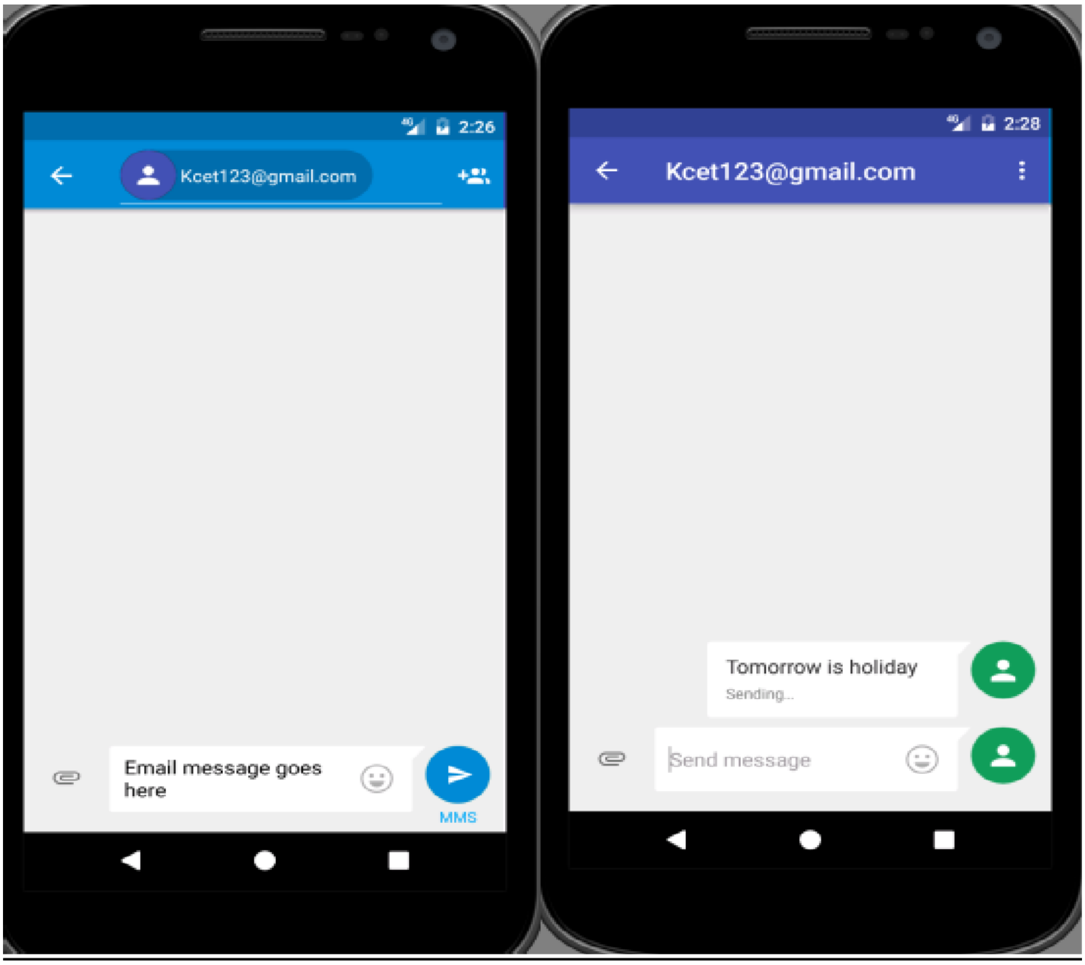
OUTPUT:



Recipient KCET

subject Check program

Message Body Hai



Kcet123@gmail.com

Email message goes here

Kcet123@gmail.com

Tomorrow is holiday

Sending...

RESULT:

Thus, the program for android application to send an email was executed successfully.

EX. NO. :6	**Create an application using Sensor Manager**
DATE :	

AIM:

To develop an Android application that makes use of the SensorManager to capture accelerometer sensor data and display it on the screen.

ALGORITHM:

1. Create a New Android Project:

- Click `New` in the toolbar.
- In the window that appears, open the `Android` folder, select `Android Application Project`, and click Next.
- Provide the application name, project name, and desired package name.
- Choose a launcher icon for your application, select `Blank Activity`, and click `Next`.
- Provide the desired activity name for your project and click `Finish`.

2. Create a New AVD (Android Virtual Device):

- Click `Android Virtual Device Manager` from the toolbar.
- In the Android Virtual Device Manager panel, click `New`.
- Fill in the details for the AVD. Give it a name, a platform target, an SD card size, and a skin (HVGA is default).
- Click `Create AVD` and select the new AVD from the Android Virtual Device Manager and click `Start`.

3. Design the Graphical Layout:

- Add `TextView` elements to display the x, y, and z values of the accelerometer.
- Optionally add buttons or other UI elements if required.

4. Initialize SensorManager:

- In the `MainActivity.java`, initialize the `SensorManager` and `Sensor` for the accelerometer in the `onCreate()` method.

5. Register Sensor Listener:

- Create a `SensorEventListener` to capture sensor data and update the values in the `TextView` elements.
- Use `onSensorChanged()` to capture accelerometer data.

6. Unregister Listener on Pause:

- Override the `onPause()` method to unregister the listener to save battery when the app is paused.

7. Run the Application:

- Run the application in the emulator or on a physical device to view the accelerometer data in real time.

8. Display the Output:

- The accelerometer data (x, y, z) will be displayed in the `TextView` elements.

9. Close the Android Project:

- After testing, you can close the project from Android Studio.

PROGRAM CODE:

MainActivity.java:

```
package com.example.sensormanagerapp;

import android.content.Context;
import android.hardware.Sensor;
import android.hardware.SensorEvent;
import android.hardware.SensorEventListener;
import android.hardware.SensorManager;
import android.os.Bundle;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity implements SensorEventListener {

    private SensorManager sensorManager;
    private Sensor accelerometer;
    private TextView xValue, yValue, zValue;

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

        // Initialize TextViews
        xValue = findViewById(R.id.xValue);
        yValue = findViewById(R.id.yValue);
        zValue = findViewById(R.id.zValue);

        // Initialize SensorManager and Accelerometer
        sensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);
        if (sensorManager != null) {
            accelerometer = sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
        }

        // Register the listener
        sensorManager.registerListener(this, accelerometer,
SensorManager.SENSOR_DELAY_NORMAL);
    }

    @Override
    public void onSensorChanged(SensorEvent event) {
        // Get accelerometer values
        float x = event.values[0];
        float y = event.values[1];
        float z = event.values[2];

        // Update UI with the accelerometer data
        xValue.setText("X: " + x);
        yValue.setText("Y: " + y);
        zValue.setText("Z: " + z);
    }

    @Override
    public void onAccuracyChanged(Sensor sensor, int accuracy) {
        // Can be used to handle accuracy changes, if needed
    }

    @Override
    protected void onPause() {
        super.onPause();
        // Unregister listener to save battery
        sensorManager.unregisterListener(this);
    }

    @Override
    protected void onResume() {
        super.onResume();
        // Re-register listener when app resumes
        sensorManager.registerListener(this, accelerometer,
SensorManager.SENSOR_DELAY_NORMAL);
    }
}
```

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"
    android:padding="16dp">

    <TextView
        android:id="@+id/xValue"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="X: 0.0"
        android:textSize="24sp" />

    <TextView
        android:id="@+id/yValue"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Y: 0.0"
        android:textSize="24sp" />

    <TextView
```

```
        android:id="@+id/zValue"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Z: 0.0"
        android:textSize="24sp" />
</LinearLayout>
```

AndroidManifest.xml:

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

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

OUTPUT:

RESULT:

The Android application successfully captures and displays accelerometer sensor data in real-time using Android’s SensorManager.

EX. NO. :7 A	**Mini Project 1**
DATE :	**Write a mobile application that creates alarm clock**

AIM:

To develop an android application that creates alarm clock.

ALGORITHM:

1. Create a New Android Project:
    - Click New in the toolbar.
    - In the window that appears, open the Android folder, select Android Application Project, and click next.
    - Provide the application name and the project name and then finally give the desired package name.
    - Choose a launcher icon for your application and then select Blank Activity and then click Next
    - Provide the desired Activity name for your project and then click Finish.
2. Create a New AVD (Android Virtual Device):
    - click Android Virtual Device Manager from the toolbar.
    - In the Android Virtual Device Manager panel, click New.
    - Fill in the details for the AVD. Give it a name, a platform target, an SD card size, and a skin (HVGA is default).
    - Click Create AVD and Select the new AVD from the Android Virtual Device Manager and click Start.
3. Design the graphical layout.
4. Run the application.
5. When the application starts alarm sound will be invoked.
6. Stop alarm button is clicked to stop the alarm.
7. Close the Android project.

PROGRAM CODE:

MainActivity.java

```
package com.example.admin.myapplication;

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 MainActivityextends AppCompatActivity
{
TimePickeralarmTimePicker;
PendingIntentpendingIntent;
AlarmManageralarmManager;

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

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

AlarmReceiverActivity.java

```
package com.example.admin.myapplication;

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 AlarmReceiverextends 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();
   }
}
```

activity_main.xml

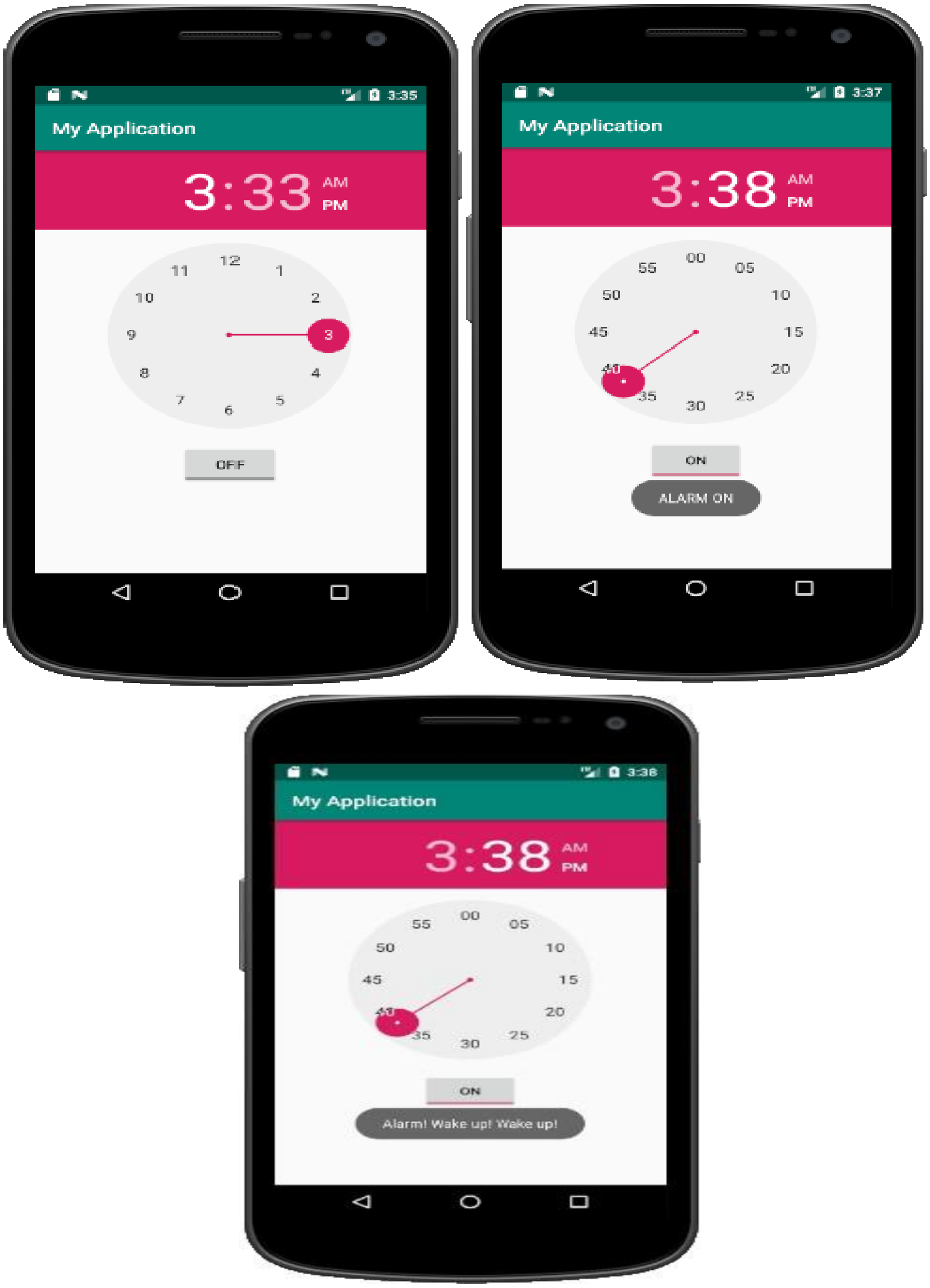
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayoutxmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical">

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

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

</LinearLayout>
```

AndroidMainfest.Xml

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
   package="com.example.admin.myapplication">

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

```
<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>
```

OUTPUT:



ALARM ON

Alarm! Wake up! Wake up!

RESULT:

Thus, the program for android application for an alarm was executed successfully.

EX. NO. :7 B	**Mini Project2**
DATE :	**Develop a native calculator application**

AIM:

To develop a calculator android application.

ALGORITHM:

1. Create a New Android Project:
    - Click New in the toolbar.
    - In the window that appears, open the Android folder, select Android Application Project, and click next.
    - Provide the application name and the project name and then finally give the desired package name.
    - Choose a launcher icon for your application and then select Blank Activity and then click Next
    - Provide the desired Activity name for your project and then click Finish.
2. Create a New AVD (Android Virtual Device):
    - click Android Virtual Device Manager from the toolbar.
    - In the Android Virtual Device Manager panel, click New.
    - Fill in the details for the AVD. Give it a name, a platform target, an SD card size, and a skin (HVGA is default).
    - Click Create AVD and Select the new AVD from the Android Virtual Device Manager and click Start.
3. Run the application.
4. Provide any two input numbers.
5. Choose any arithmetic operations of your choice and the output gets displayed on the display screen of the calculator application.
6. Close the Android project.

PROGRAM CODE:

MainActivity.java

```
package com.example.calculator_two;
import android.app.Activity;
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 {
        private Button nine, eig, sev, six, fiv, four, thr, two, one, zero, dot,
                        plus, mins, div, mul, eq, cl;
        private EditText et;
private String s = "0";
private int result = 0;
private char lO = ' ';

protected void onCreate(Bundle savedInstanceState) {
        // TODO Auto-generated method stub
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        nine = (Button) findViewById(R.id.b9);
        eig = (Button) findViewById(R.id.b8);
        sev = (Button) findViewById(R.id.b7);
        six = (Button) findViewById(R.id.b6);
        fiv = (Button) findViewById(R.id.b5);
        four = (Button) findViewById(R.id.b4);
        thr = (Button) findViewById(R.id.b3);
        two = (Button) findViewById(R.id.b2);
        one = (Button) findViewById(R.id.b1);
        zero = (Button) findViewById(R.id.b0);
        dot = (Button) findViewById(R.id.bd);
        plus = (Button) findViewById(R.id.bpl);
        mins = (Button) findViewById(R.id.bmin);
        div = (Button) findViewById(R.id.bdiv);
        mul = (Button) findViewById(R.id.bmul);
        eq = (Button) findViewById(R.id.beq);
        cl = (Button) findViewById(R.id.bcl);
        et = (EditText) findViewById(R.id.tv);
        zero.setOnClickListener(this);
        nine.setOnClickListener(this);
        eig.setOnClickListener(this);
        sev.setOnClickListener(this);
        six.setOnClickListener(this);
        fiv.setOnClickListener(this);
        four.setOnClickListener(this);
        thr.setOnClickListener(this);
        two.setOnClickListener(this);
        one.setOnClickListener(this);
        dot.setOnClickListener(this);
        plus.setOnClickListener(this);
        mins.setOnClickListener(this);
        div.setOnClickListener(this);
        mul.setOnClickListener(this);
        eq.setOnClickListener(this);
        cl.setOnClickListener(this);
        et.setOnClickListener(this);
}

public void onClick(View v) {
```

```
switch (v.getId()) {
case R.id.b0:
case R.id.b1:
case R.id.b2:
case R.id.b3:
case R.id.b4:
case R.id.b5:
case R.id.b6:
case R.id.b7:
case R.id.b8:
case R.id.b9:

        String inDigit = ((Button) v).getText().toString();
        if (s.equals("0")) {
                s = inDigit;
        } else {
                s += inDigit;
        }
        et.setText(s);
        if (lO == '=') {
                result = 0;
                lO = ' ';
        }
        break;
case R.id.bpl:
        compute();
        lO = '+';
        break;
case R.id.bmin:
        compute();
        lO = '-';
        break;
case R.id.bdiv:
        compute();
        lO = '/';
        break;
case R.id.bmul:
        compute();
        lO = '*';
        break;
case R.id.beq:
        compute();
        lO = '=';
        break;
case R.id.bcl:
        result = 0;
        s = "0";
```

```
                lO = ' ';
                et.setText("0");
                break;
            }
        }

        private void compute() {
            int inNum = Integer.parseInt(s);
            s = "0";
            if (lO == ' ') {
                result = inNum;
            } else if (lO == '+') {
                result += inNum;
            } else if (lO == '-') {
                result -= inNum;
            } else if (lO == '*') {
                result *= inNum;
            } else if (lO == '/') {
                result /= inNum;
            } else if (lO == '=') {
                // Keep the result for the next operation
            }
            et.setText(String.valueOf(result));
        }
}
```

activity_main.xml

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

<EditText
android:id="@+id/tv"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:hint="enter no. here"
android:textSize="30dp" />

<LinearLayout
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginTop="100dp"
android:orientation="horizontal"
android:weightSum="4">
```

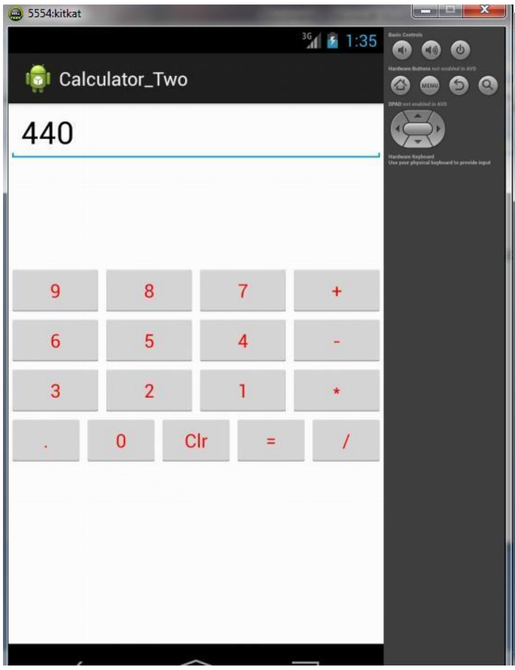
```
<Button
android:id="@+id/b9"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_weight="1"
android:text="9"
android:textColor="#ff0000" />

<Button
android:id="@+id/b8"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_weight="1"
android:text="8"
android:textColor="#ff0000" />

<Button
android:id="@+id/b7"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_weight="1"
android:text="7"
android:textColor="#ff0000" />

<Button
android:id="@+id/bpl"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_weight="1"
android:text="+"
android:textColor="#ff0000" />
</LinearLayout>

<LinearLayout
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:orientation="horizontal"
android:weightSum="4">

<Button
android:id="@+id/b6"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_weight="1"
android:text="6"
android:textColor="#ff0000" />
```

```
<Button
android:id="@+id/b5"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_weight="1"
android:text="5"
android:textColor="#ff0000" />

<Button
android:id="@+id/b4"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_weight="1"
android:text="4"
android:textColor="#ff0000" />

<Button
android:id="@+id/bmin"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_weight="1"
android:text="-"
android:textColor="#ff0000" /></LinearLayout>

<LinearLayout
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:orientation="horizontal"
android:weightSum="4">

<Button
android:id="@+id/b3"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_weight="1"
android:text="3"
android:textColor="#ff0000" />

<Button
android:id="@+id/b2"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_weight="1"
android:text="2"
android:textColor="#ff0000" />
```

```
<Button
android:id="@+id/b1"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_weight="1"
android:text="1"
android:textColor="#ff0000" />

<Button
android:id="@+id/bmul"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_weight="1"
android:text="*"
android:textColor="#ff0000" />
</LinearLayout>

<LinearLayout
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:orientation="horizontal"
android:weightSum="5">

<Button
android:id="@+id/bd"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_weight="1"
android:text="."
android:textColor="#ff0000" />

<Button
android:id="@+id/b0"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_weight="1"
android:text="0"
android:textColor="#ff0000" />

<Button
android:id="@+id/bcl"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_weight="1"
```

```
android:text="Clr"
android:textColor="#ff0000" />


<Button
android:id="@+id/beq"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_weight="1"
android:text="="
android:textColor="#ff0000" />

<Button
android:id="@+id/bdiv"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_weight="1"
android:text="/"
android:textColor="#ff0000" />
</LinearLayout>

</LinearLayout>
```

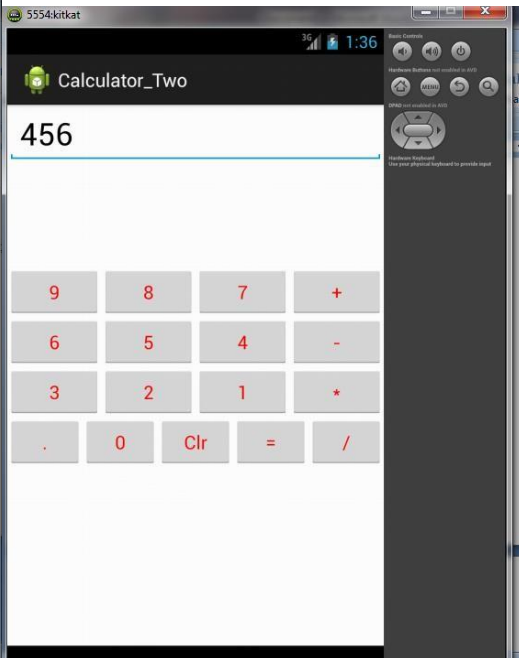
OUTPUT:



5554:kitkat

Calculator_Two

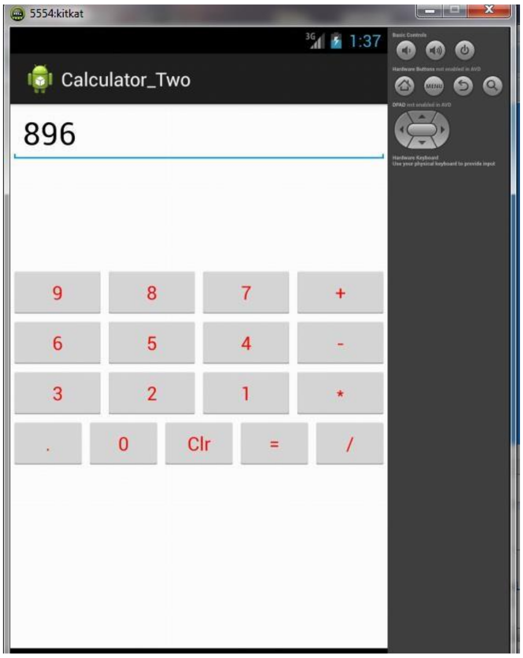
440



5554:kitkat

Calculator_Two

456



5554:kitkat

Calculator_Two

896

RESULT:

Thus, the program for android based calculator application was executed successfully.

EX. NO. :8	**Create An Android Application That Converts the User Input Text to Voice.**
DATE :	

AIM:

To develop an Android application that converts user input text into voice using Android's `TextToSpeech` (TTS) engine.

ALGORITHM:

1. Create a New Android Project:
   - Click `New` in the toolbar.
   - In the window that appears, open the `Android` folder, select `Android Application Project`, and click `Next`.
   - Provide the application name, project name, and desired package name.
   - Choose a launcher icon for your application, select `Blank Activity`, and click `Next`.
   - Provide the desired activity name for your project and click `Finish`.

- Create a New AVD (Android Virtual Device):
- Click `Android Virtual Device Manager` from the toolbar.
- In the Android Virtual Device Manager panel, click `New`.
- Fill in the details for the AVD. Give it a name, a platform target, an SD card size, and a skin (HVGA is default).
- Click `Create AVD` and select the new AVD from the Android Virtual Device Manager and click `Start`.

3. Design the Graphical Layout:
   - Add an `EditText` to take the user’s text input.
   - Add a `Button` that, when clicked, will convert the entered text to speech.

4. Initialize TextToSpeech Engine:
   - In the `MainActivity.java`, initialize the `TextToSpeech` engine in the `onCreate()` method.

5. Set Up Button Action:
   - Set up the button to fetch the text entered in the `EditText` and pass it to the `TextToSpeech` engine to be converted to voice when the button is clicked.

6. Handle Lifecycle Events:
   - Implement proper lifecycle handling by shutting down the `TextToSpeech` engine in the `onDestroy()` method to release system resources.

7. Run the Application:
   - Run the application in the emulator or on a physical device to test the text-to-speech functionality.

8. Display the Output:
   - The user will input text, click the button, and the application will convert the text to speech.

9. Close the Android Project:

- After testing, you can close the project from Android Studio.

PROGRAM CODE:

MainActivity.java:

```
package com.example.texttospeechapp;

import android.os.Bundle;
import android.speech.tts.TextToSpeech;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;

import java.util.Locale;

public class MainActivity extends AppCompatActivity {

    private TextToSpeech textToSpeech;
    private EditText editText;
    private Button speakButton;

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

        // Initialize UI elements
        editText = findViewById(R.id.editText);
        speakButton = findViewById(R.id.speakButton);

        // Initialize TextToSpeech engine
        textToSpeech = new TextToSpeech(getApplicationContext(), new TextToSpeech.OnInitListener() {
            @Override
            public void onInit(int status) {
                if (status == TextToSpeech.SUCCESS) {
                    // Set language for TextToSpeech engine
                    int result = textToSpeech.setLanguage(Locale.US);
                    if (result == TextToSpeech.LANG_MISSING_DATA || result ==
TextToSpeech.LANG_NOT_SUPPORTED) {
                        Toast.makeText(MainActivity.this, "Language not supported",
Toast.LENGTH_SHORT).show();
                    }
                } else {
                    Toast.makeText(MainActivity.this, "TextToSpeech Initialization Failed!",
Toast.LENGTH_SHORT).show();
                }
            }
        });

        // Set button click listener
        speakButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // Get input text and convert to speech
                String text = editText.getText().toString();
                if (!text.isEmpty()) {
                    textToSpeech.speak(text, TextToSpeech.QUEUE_FLUSH, null, null);
                } else {
                    Toast.makeText(MainActivity.this, "Please enter text", Toast.LENGTH_SHORT).show();
                }
            }
        });
    }

    @Override
    protected void onDestroy() {
        if (textToSpeech != null) {
            textToSpeech.stop();
            textToSpeech.shutdown();
        }
        super.onDestroy();
    }
}
```

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"
    android:padding="16dp">

    <EditText
        android:id="@+id/editText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter text to speak"
        android:textSize="18sp" />

    <Button
        android:id="@+id/speakButton"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Speak"
        android:textSize="18sp"
        android:layout_marginTop="16dp"/>
</LinearLayout>
```

AndroidManifest.xml:

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

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

OUTPUT:

RESULT:

The Android application successfully converts user input text to speech using Android's `TextToSpeech` API. The application captures user input, processes it through the `TextToSpeech` engine, and outputs the speech through the device.