PROGRAM 7

AIM: Write an application to implement Date Picker.

ALGORITHM:

- Create a New Android Project:
- Design the Layout:
- Implement the Java Code:
- Open MainActivity.java.
- Run the Application:

CODE:

Activity_main.XML code:-

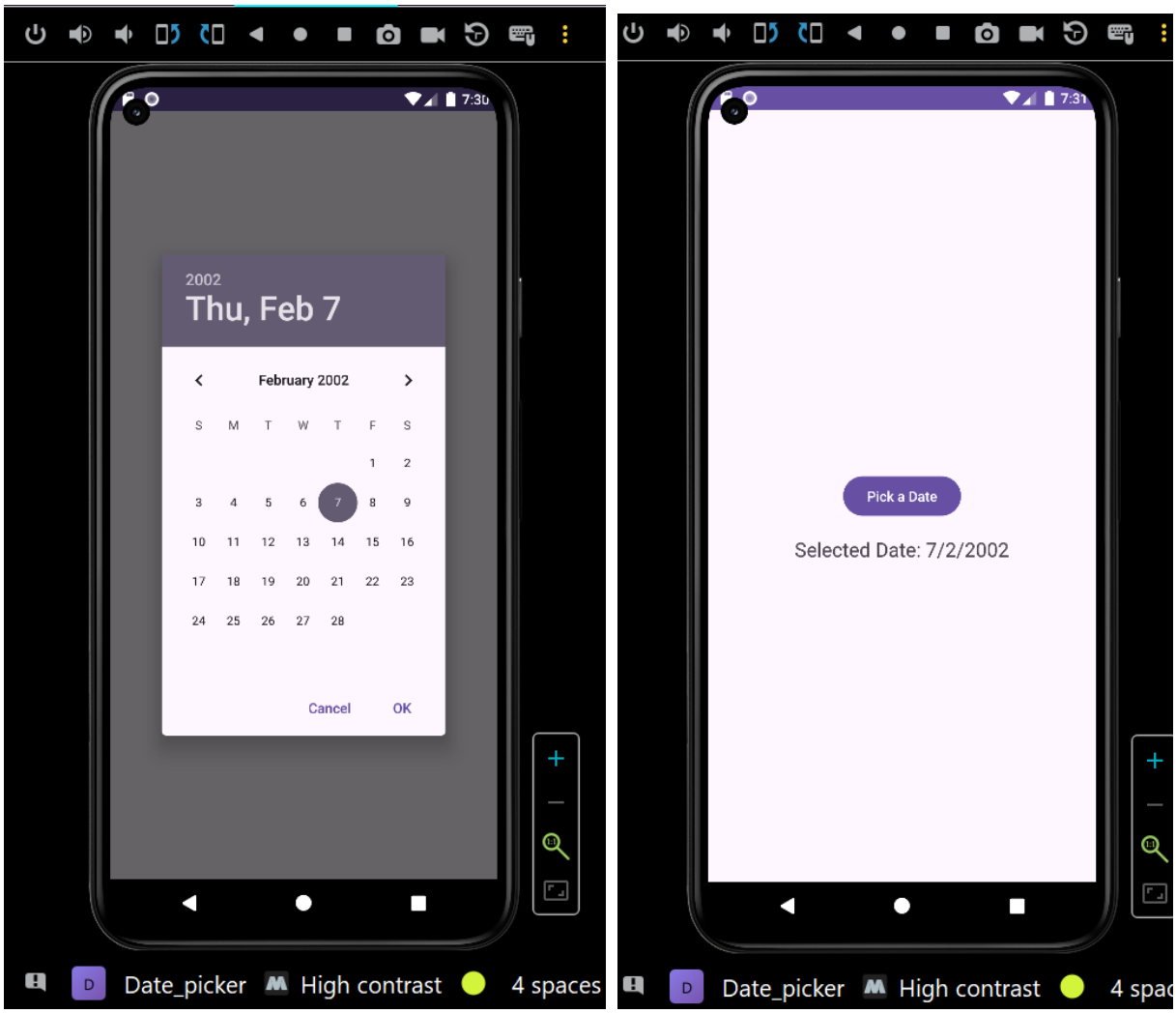
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="match_parent">
   <Button
      android:id="@+id/b1"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Pick a Date"
      android:layout_centerInParent="true"/>
   <TextView
      android:id="@+id/t1"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_below="@id/b1"
      android:layout_marginTop="16dp"
      android:textSize="20dp"
      android:layout_centerHorizontal="true"/>
</RelativeLayout>
```

Main_Activity.Java code:-

```
package com.example.date_picker;
import android.app.DatePickerDialog;
import android.os.Bundle;
```

```
import android.view.View;
import android.widget.Button;
import android.widget.DatePicker;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;
import java.util.Calendar;

public class MainActivity extends AppCompatActivity {

   private Button b1;
   private TextView t1;

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

      b1 = findViewById(R.id.b1);
      t1 = findViewById(R.id.t1);

      b1.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View v) {
            launch();
         }
      });
   }
   private void launch() {
      // Get the current date
      Calendar c = Calendar.getInstance();
      int day = c.get(Calendar.DAY_OF_MONTH);
      int month = c.get(Calendar.MONTH);
      int year = c.get(Calendar.YEAR);
      // Create a date picker dialog
      DatePickerDialog d1 = new DatePickerDialog(this,new
DatePickerDialog.OnDateSetListener() {
               @Override
               public void onDateSet(DatePicker view, int year,int month,int day) {
                  // Update the TextView with the selected date
                  String Date = day + "/" + (month + 1) + "/" + year;
                  t1.setText("Selected Date: " + Date);
               }
            },year,month,day);
      // Show the date picker dialog
      d1.show();
   }
}
```

OUTPUT:



RESULT: The program to implement Date Picker is completed and executed successfully.

PROGRAM 8

AIM: Write an application to implement Student Registration form using Listview.

ALGORITHM:

- **Create a new Android Studio project**: Start by creating a new Android Studio project with an appropriate name.
- **Design the layout**: Design the layout for your Student Registration form using XML in the layout file.
- **Set up ListView**: Add a ListView component to your layout file where you want to display the list of registered students.
- Run the Application:

CODE:

Activity_main.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=".MainActivity">

    <EditText
        android:id="@+id/editTextName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Name"
        android:inputType="textPersonName" />

    <EditText
        android:id="@+id/editTextEmail"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/editTextName"
        android:hint="Email"
        android:inputType="textEmailAddress" />

    <EditText
        android:id="@+id/editTextPhone"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:hint="Phone"
        android:inputType="phone" />
```

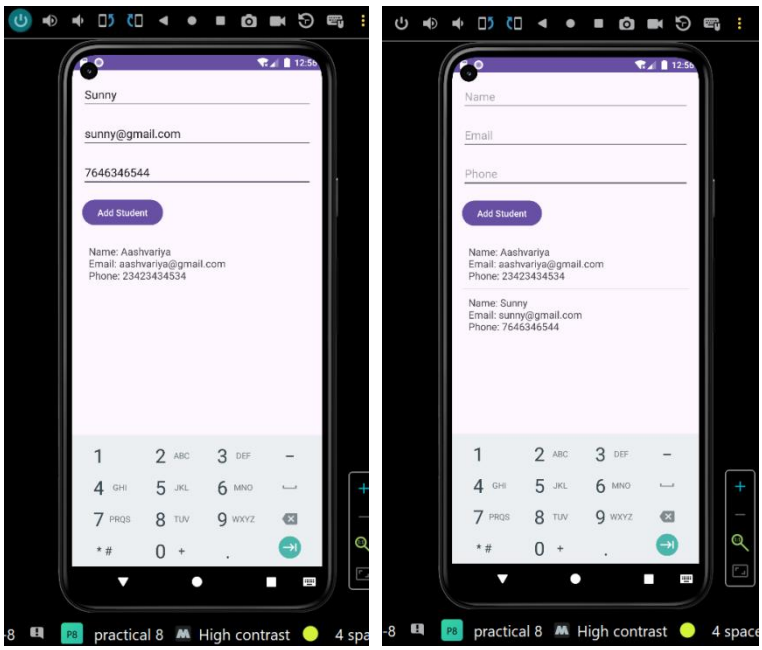
```
    <Button
        android:id="@+id/buttonAdd"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:text="Add Student" />

    <ListView
        android:id="@+id/listViewStudents"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/buttonAdd"
        android:layout_marginTop="16dp" />

</RelativeLayout>
```

Main_Activity.Java code:-

```
import android.os.Bundle;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ListView;
import androidx.appcompat.app.AppCompatActivity;
import java.util.ArrayList;
public class MainActivity extends AppCompatActivity {

    EditText editTextName, editTextEmail, editTextPhone;
    Button buttonAdd;
    ListView listViewStudents;
    ArrayList<String> studentsList;
    ArrayAdapter<String> adapter;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        editTextName = findViewById(R.id.editTextName);
        editTextEmail = findViewById(R.id.editTextEmail);
        editTextPhone = findViewById(R.id.editTextPhone);
        buttonAdd = findViewById(R.id.buttonAdd);
        listViewStudents = findViewById(R.id.listViewStudents);
        studentsList = new ArrayList<>();
        adapter = new ArrayAdapter<>(this, R.layout.list_item_student, R.id.textViewStudent,
```

```
studentsList);
        listViewStudents.setAdapter(adapter);
        buttonAdd.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                addStudent();
            }
        });
    }
    private void addStudent() {
        String name = editTextName.getText().toString().trim();
        String email = editTextEmail.getText().toString().trim();
        String phone = editTextPhone.getText().toString().trim();
        if (!name.isEmpty() && !email.isEmpty() && !phone.isEmpty()) {
            String studentDetails = "Name: " + name + "\nEmail: " + email + "\nPhone: " + phone;
            studentsList.add(studentDetails);
            adapter.notifyDataSetChanged();
            editTextName.getText().clear();
            editTextEmail.getText().clear();
            editTextPhone.getText().clear();
        }
    }
}
```

OUTPUT:



RESULT: The program to implement Student Registration form using Listview has been completed and executed successfully.

PROGRAM 9

AIM: Write an application to implement a Context menu.

ALGORITHM:

- Create a new Android project in Android Studio.
- Design the layout of the main activity (activity_main.xml) with a TextViewor any other view that you want to attach the context menu to.
- Override the onCreateContextMenu() method in the MainActivity to create the context menu.
- Override the onContextItemSelected() method to handle item selection from the context menu.
- Run the application on an Android device or emulator.

CODE:

Activity_main.XML code:-

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="match_parent">
   <Button
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:id="@+id/b1"
      android:text="press me one sec"
      android:layout_centerInParent="true"/>
</RelativeLayout>
```

Main_Activity.Java code:-

```
import android.os.Bundle;
import android.view.ContextMenu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

   private Button b1;
```

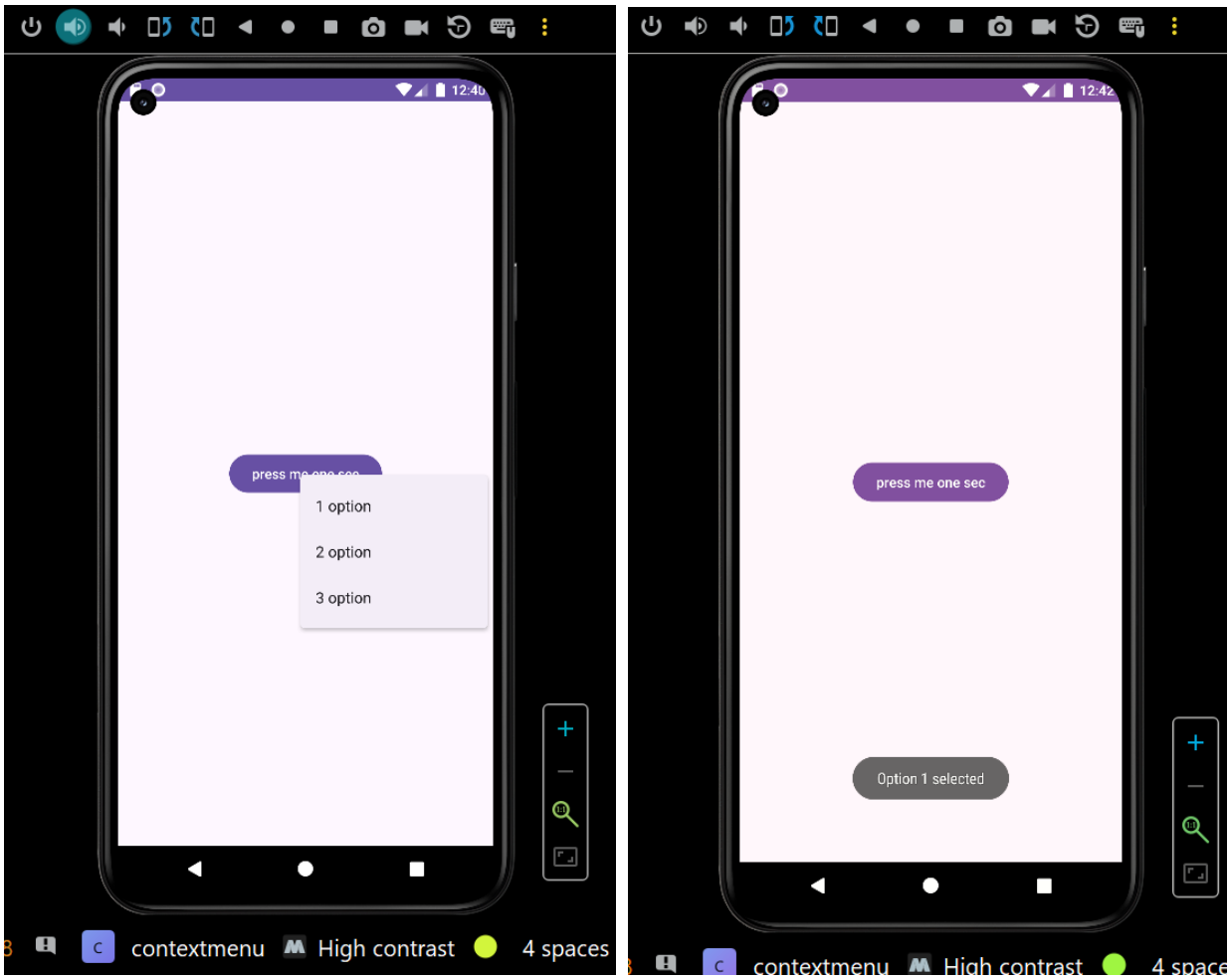
```
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        b1 = findViewById(R.id.b1);



        registerForContextMenu(b1);
    }

    @Override
    public void onCreateContextMenu(ContextMenu menu, View v,
ContextMenu.ContextMenuInfo menuInfo) {
        super.onCreateContextMenu(menu, v, menuInfo);
        getMenuInflater().inflate(R.menu.context_menu, menu);
    }
        @Override
        public boolean onContextItemSelected(MenuItem item) {
            // Handle context menu item selection using if-else statements
            if (item.getItemId() == R.id.i1) {
                Toast.makeText(this, "Option 1 selected", Toast.LENGTH_SHORT).show();
                return true;
            } else if (item.getItemId() == R.id.i2) {
                Toast.makeText(this, "Option 2 selected", Toast.LENGTH_SHORT).show();
                return true;
            } else if (item.getItemId() == R.id.i3) {
                Toast.makeText(this, "Option 3 selected", Toast.LENGTH_SHORT).show();
                return true;
            } else {
                return super.onContextItemSelected(item);
            }
        }

    }
```

Context_menu.XML code:-

```
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item
        android:id="@+id/i1"
        android:title="1 option" />
```

```
    <item
        android:id="@+id/i2"
        android:title="2 option" />
    <item
        android:id="@+id/i3"
        android:title="3 option" />
</menu>
```

OUTPUT:



RESULT: The program to implement a Context menu is completedand executed successfully.

PROGRAM 10

AIM: Write an application to implement an Option Menu.

ALGORITHM:

- Create a new Android project in Android Studio.
- Design the layout of the main activity (activity_main.xml) with a TextViewor any other view to display the selected option from the menu.
- Override the onCreateOptionsMenu() method in the MainActivity to create the option menu.
- Override the onOptionsItemSelected() method to handle item selection from the option menu.
- Run the application on an Android device or emulator.

CODE:

Activity_main.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=".MainActivity">
   </RelativeLayout>
```

Option_menu.XML code:-

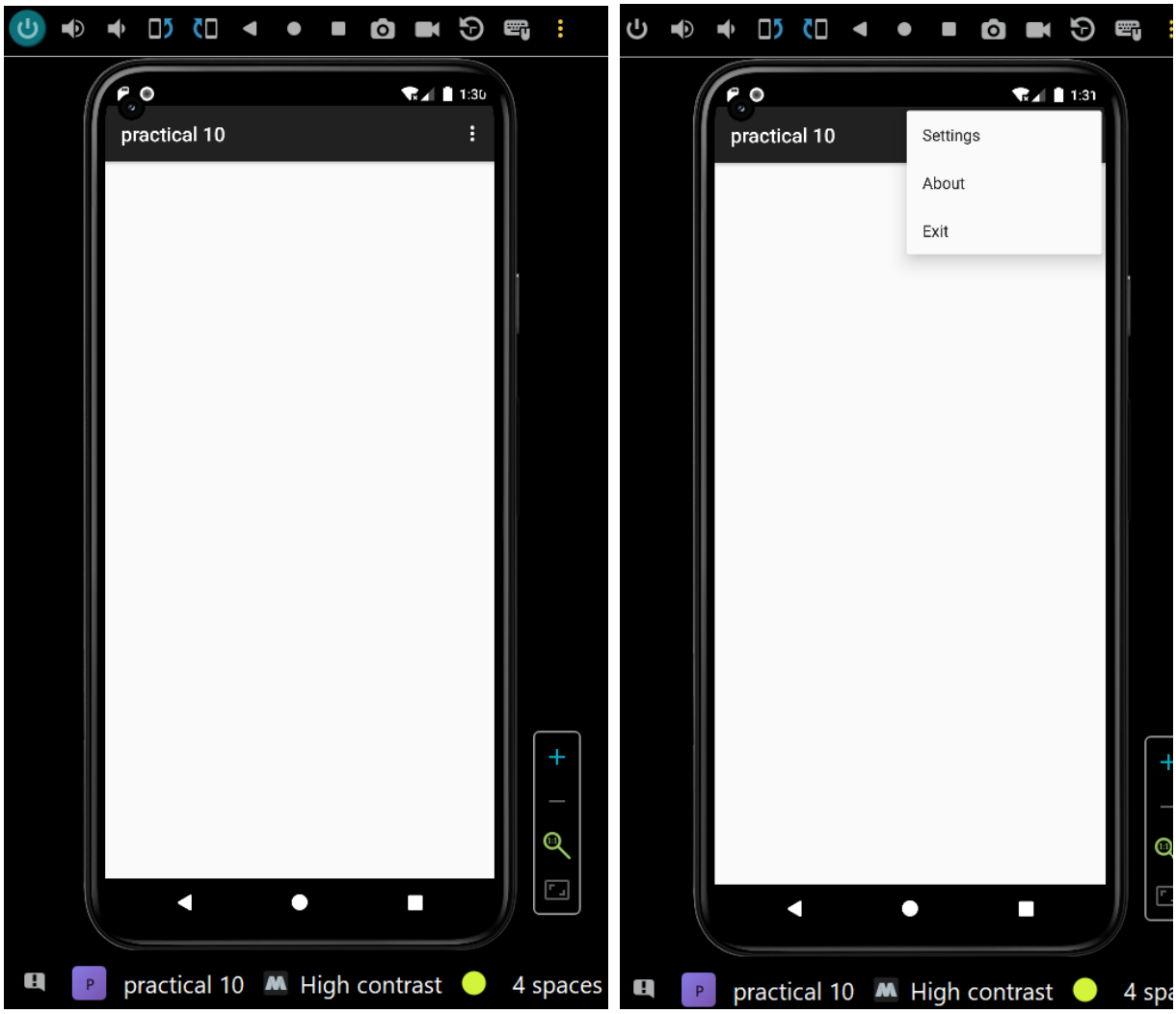
```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item
        android:id="@+id/action_settings"
        android:title="Settings"/>
    <item
        android:id="@+id/action_about"
        android:title="About"/>
    <item
        android:id="@+id/action_exit"
        android:title="Exit"/>
</menu>
```

Main_Activity.Java code:-

```
import android.os.Bundle;
```

```
import android.view.Menu;
import android.view.MenuItem;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

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


    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.option_menu, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        int id = item.getItemId();

        // Handle menu item clicks
        if (id == R.id.action_settings) {
            // Handle Settings click
            return true;
        } else if (id == R.id.action_about) {
            // Handle About click
            return true;
        } else if (id == R.id.action_exit) {
            // Handle Exit click
            finish(); // Close the activity
            return true;
        }

        return super.onOptionsItemSelected(item);
    }
}
```

OUTPUT:



RESULT: The program to implement an Option Menu is completed and executed successfully.

PROGRAM 11

AIM: Write an application program to show how to Set and retrieve shared preferences.

ALGORITHM:

- Create a new Android project in Android Studio.
- Define constants for the preference keys in your project.
- Set shared preferences when needed using the SharedPreferences class.
- Retrieve shared preferences when needed using the SharedPreferences class.You can use the appropriate getter methods to retrieve the values of preferences.
- Run the application on an Android device or emulator.

CODE:

Activity_main.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=".MainActivity"
 tools:ignore="HardcodedText">

 <TextView
        android:id="@+id/textview"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="32dp"
        android:text="Shared Preferences Demo"
        android:textColor="@android:color/black"
        android:textSize="24sp" />

 <EditText
        android:id="@+id/edit1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/textview"
        android:layout_marginStart="16dp"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="16dp"
        android:hint="Enter your Name"
```

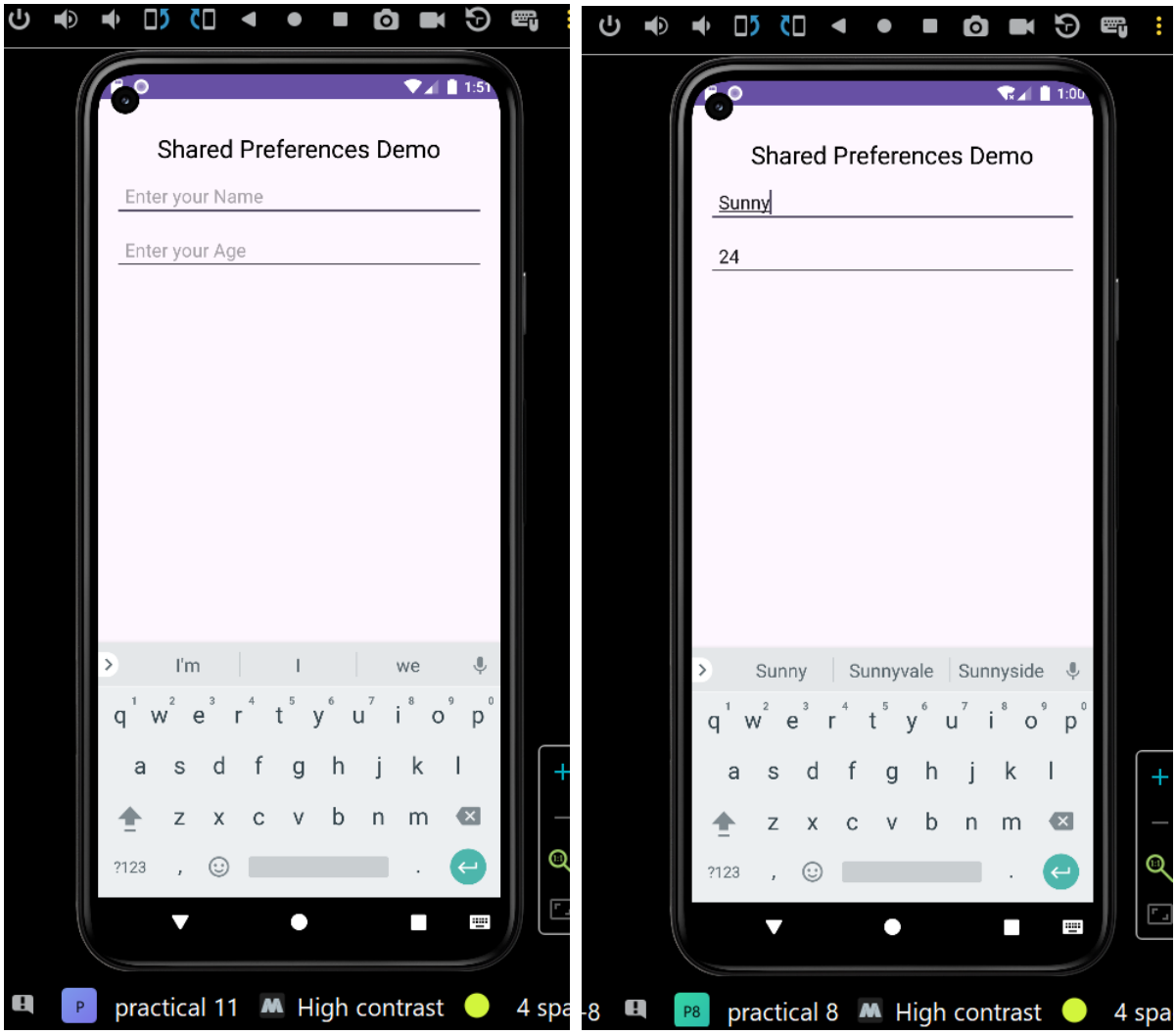
```
        android:padding="10dp" />

<EditText
        android:id="@+id/edit2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/edit1"
        android:layout_marginStart="16dp"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="16dp"
        android:hint="Enter your Age"
        android:inputType="number"
        android:padding="10dp" />
</RelativeLayout>
```

Main_Activity.Java code:-

```
import androidx.appcompat.app.AppCompatActivity;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.widget.EditText;

public class MainActivity extends AppCompatActivity {
   private EditText name, age;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      name = findViewById(R.id.edit1);
      age = findViewById(R.id.edit2);
   }
   @Override
   protected void onResume() {
      super.onResume();
      SharedPreferences sh = getSharedPreferences("MySharedPref", MODE_PRIVATE);
      String s1 = sh.getString("name", "");
      int a = sh.getInt("age", 0);

      name.setText(s1);
      age.setText(String.valueOf(a));
   }
   @Override
   protected void onPause() {
      super.onPause();
```

```
        SharedPreferences sharedPreferences = getSharedPreferences("MySharedPref",
MODE_PRIVATE);
        SharedPreferences.Editor myEdit = sharedPreferences.edit();

        myEdit.putString("name", name.getText().toString());
        myEdit.putInt("age", Integer.parseInt(age.getText().toString()));
        myEdit.apply();
    }
}
```

OUTPUT:



RESULT: The program to show how to Set and retrieve shared preferences.is completed and executed successfully.

PROGRAM 12

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

ALGORITHM:

- Create a new Android project in Android Studio.
- Define the database schema: Decide on the structure of your database,including the tables and columns you'll need.
- Create a subclass of SQLiteOpenHelper: This class will help you manage database creation and version management.
- Implement the logic for your app's activities: Write the Java code to handle user interactions and data management in your app's activities.
- Test your application: Run your application on an Android device or emulator to test its functionality.

CODE:

Activity_main.XML code:-

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

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

    <EditText
        android:id="@+id/Rollno"
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:layout_x="175dp"
        android:layout_y="100dp"
```

```
    android:inputType="number"
    android:textSize="20sp" />

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

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

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

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

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

<Button
    android:id="@+id/Delete"
```

```
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:layout_x="200dp"
        android:layout_y="300dp"
        android:text="Delete"
        android:textSize="30dp" />

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

    <Button
        android:id="@+id/View"
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:layout_x="200dp"
        android:layout_y="400dp"
        android:text="View"
        android:textSize="30dp" />

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

</AbsoluteLayout>
```

Main_Activity.Java code:-

```
import android.annotation.SuppressLint;
import android.app.Activity;
import android.app.AlertDialog.Builder;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
```

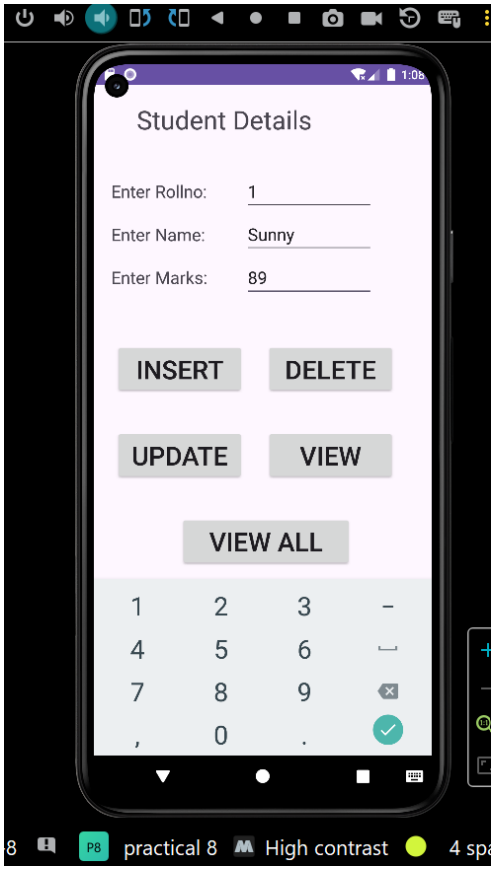
```
import android.widget.Button;
import android.widget.EditText;
public class MainActivity extends Activity implements OnClickListener
{
   EditText Rollno,Name,Marks;
   Button Insert,Delete,Update,View,ViewAll;
   SQLiteDatabase db;
   @Override
   public void onCreate(Bundle savedInstanceState){
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);

      Rollno=(EditText)findViewById(R.id.Rollno);
      Name=(EditText)findViewById(R.id.Name);
      Marks=(EditText)findViewById(R.id.Marks);
      Insert=(Button)findViewById(R.id.Insert);
      Delete=(Button)findViewById(R.id.Delete);
      Update=(Button)findViewById(R.id.Update);
      View=(Button)findViewById(R.id.View);
      ViewAll=(Button)findViewById(R.id.ViewAll);

      Insert.setOnClickListener(this);
      Delete.setOnClickListener(this);
      Update.setOnClickListener(this);
      View.setOnClickListener(this);
      ViewAll.setOnClickListener(this);
      // Creating database and table
      db=openOrCreateDatabase("StudentDB", Context.MODE_PRIVATE, null);
      db.execSQL("CREATE TABLE IF NOT EXISTS student(rollno VARCHAR,name
VARCHAR,marks VARCHAR);");
   }
   public void onClick(View view)
   {
     if(view==Insert)
      {
         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;
         }
```

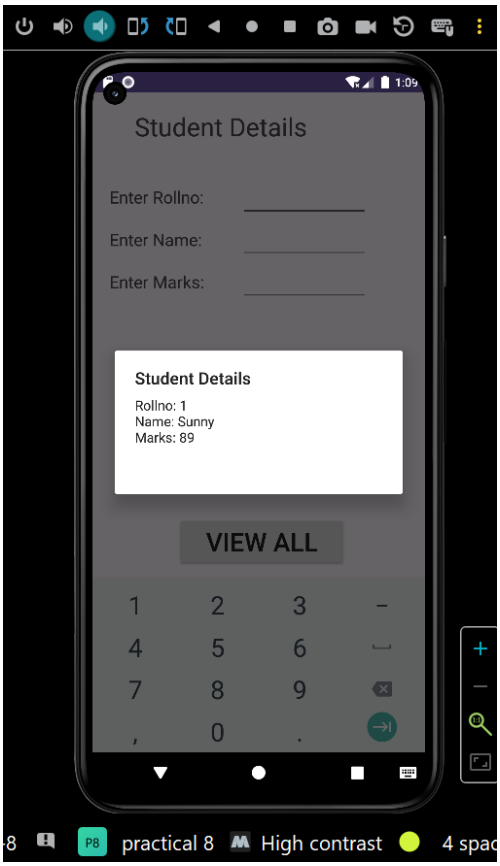
```
        db.execSQL("INSERT INTO student
VALUES('"+Rollno.getText()+"','"+Name.getText()+"','"+Marks.getText()+"');");
        showMessage("Success", "Record added");
        clearText();
      }
      if(view==Delete)
      {
       if(Rollno.getText().toString().trim().length()==0)
        {
          showMessage("Error", "Please enter Rollno");
          return;
        }
        @SuppressLint("Recycle") 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();
      }
       if(view==Update)
      {
        if(Rollno.getText().toString().trim().length()==0)
        {
          showMessage("Error", "Please enter Rollno");
          return;
        }
        @SuppressLint("Recycle") 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();
        }
         if(view==View)
        {
            if(Rollno.getText().toString().trim().length()==0)
            {
                showMessage("Error", "Please enter Rollno");
                return;
            }
            @SuppressLint("Recycle") Cursor c=db.rawQuery("SELECT * FROM student
WHERE rollno='"+Rollno.getText()+"'", null);
            if(c.moveToFirst())
            {
                Name.setText(c.getString(1));
                Marks.setText(c.getString(2));
            }
            else
            {
                showMessage("Error", "Invalid Rollno");
                clearText();
            }
        }
         if(view==ViewAll)
        {
            Cursor c=db.rawQuery("SELECT * FROM student", null);
            if(c.getCount()==0)
            {
                showMessage("Error", "No records found");
                return;
            }
            StringBuilder buffer=new StringBuilder();
            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();
    }
}
```

OUTPUT:





RESULT: The program To develop a Simple Android Application that makes use of Database is completed and executed successfully.

PROGRAM 13

AIM: Implement an application that writes data to the SD card.

ALGORITHM:

- Check for runtime permissions: Starting from Android 6.0 (API level 23),you need to request permissions at runtime. Check if the permission is granted, and if not, request it from the user.
- Determine the file path: Decide where you want to write the data on the SDcard. You can get the external storage directory using Environment.getExternalStorageDirectory().
- Write data to a file: Use Java I/O operations to write data to a file on the SDcard. You can use classes like FileOutputStream or BufferedWriter to writedata.
- Handle exceptions: Make sure to handle exceptions that may occur duringfile writing, such as IOException.
- Test your application: Run the application on an Android device or emulator.

CODE:

Activity_main.XML code:-

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

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

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

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

   <Button
      android:id="@+id/button3"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:layout_margin="10dp"
      android:text="Clear"
      android:textSize="30dp" />
</LinearLayout>
```

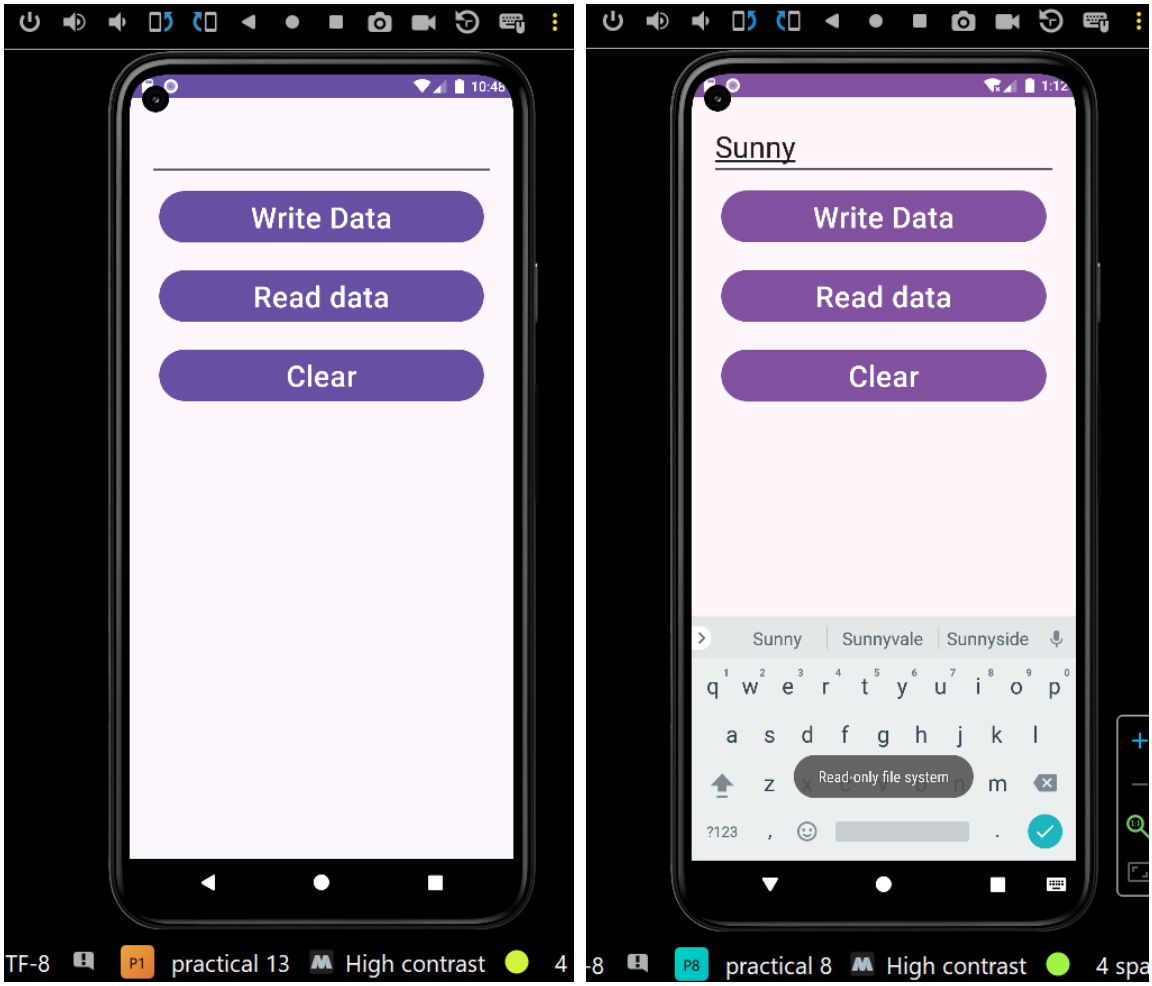
Main_Activity.Java code:-

```
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.InputStreamReader;
public class MainActivity extends AppCompatActivity
{
   EditText e1;
   Button write,read,clear;
   @Override
   protected void onCreate(Bundle savedInstanceState){
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      e1= (EditText) findViewById(R.id.editText);
      write= (Button) findViewById(R.id.button);
      read= (Button) findViewById(R.id.button2);
      clear= (Button) findViewById(R.id.button3);
      write.setOnClickListener(new View.OnClickListener(){
         @Override
```

```
    public void onClick(View v)
    {
       String message=e1.getText().toString();
       try{
          File f=new File("D:\\test\\nikki.txt");
          f.createNewFile();
          FileOutputStream fout=new FileOutputStream(f);
          fout.write(message.getBytes());
          fout.close();
          Toast.makeText(getBaseContext(),"Data Written in
SDCARD",Toast.LENGTH_LONG).show();
       }
       catch (Exception e)
       {
  Toast.makeText(getBaseContext(),e.getMessage(),Toast. LENGTH_LONG).show();
       }
    }
    });
    read.setOnClickListener(new View.OnClickListener()
    {
       @Override
       public void onClick(View v)
       {
          String message;
          String buf = "";
          try
          {
             File f = new File("D:\\test\\nikki.txt");
             FileInputStream fin = new FileInputStream(f);
             BufferedReader br = new BufferedReader(new
                   InputStreamReader(fin));
             while ((message = br.readLine()) != null)
             {
                buf += message;
             }
             e1.setText(buf);
             br.close();
             fin.close();
             Toast.makeText(getBaseContext(),"Data Recived from
SDCARD",Toast.LENGTH_LONG).show();
          }
          catch (Exception e)
```

```
            {
                Toast.makeText(getBaseContext(), e.getMessage(),
                        Toast.LENGTH_LONG).show();
            }
        } });
    clear.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v){
            e1.setText(""); }
    });
}}
```

OUTPUT:



RESULT: The program to Implement an application that writes data to the SD card is completed and executed successfully.

PROGRAM 14

AIM: Implement an application that creates an alert upon receiving a message.

ALGORITHM:

- Add permissions to the AndroidManifest.xml file: Declare the necessary permissions to receive SMS messages.
- Create a BroadcastReceiver: Create a class that extends BroadcastReceiver tdisten for incoming SMS messages.
- Register the BroadcastReceiver in the AndroidManifest.xml file:
- Handle runtime permissions: Starting from Android 6.0 (API level 23), youneed to request permissions at runtime.
- Test your application: Run the application on an Android device or emulator.

CODE:

Activity_main.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=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="180dp"
        android:gravity="center_horizontal"
        android:text="Press The Back Button of Your Phone."
        android:textSize="30dp"
        android:textStyle="bold" />
</RelativeLayout>
```

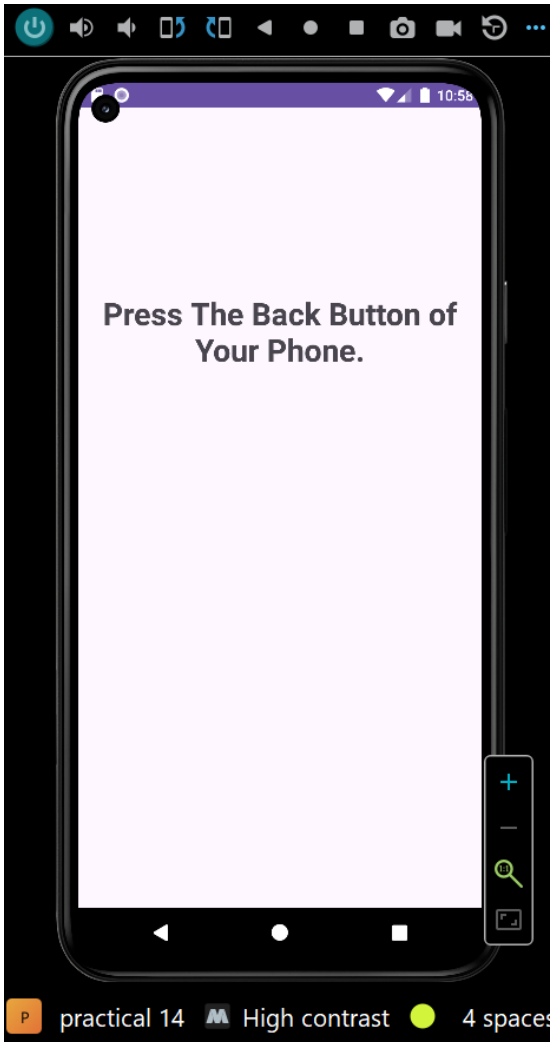
Main_Activity.Java code:-

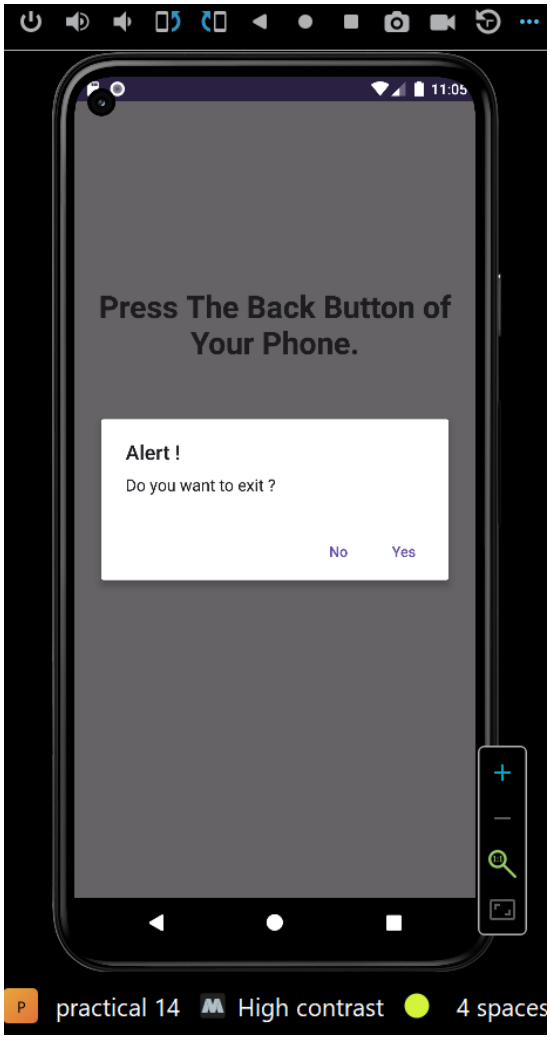
```
import android.content.DialogInterface;
import android.os.Bundle;
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
```

```
public class MainActivity extends AppCompatActivity {

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

    // Declare the onBackPressed method when the back button is pressed this method will call
    @Override
    public void onBackPressed() {
        // Create the object of AlertDialog Builder class
        AlertDialog.Builder builder = new AlertDialog.Builder(MainActivity.this);

        // Set the message show for the Alert time
        builder.setMessage("Do you want to exit ?");

        // Set Alert Title
        builder.setTitle("Alert !");

        // Set Cancelable false for when the user clicks on the outside the Dialog Box then it will remain show
        builder.setCancelable(false);

        // Set the positive button with yes name Lambda OnClickListener method is use of DialogInterface interface.
        builder.setPositiveButton("Yes", (DialogInterface.OnClickListener) (dialog, which) -> {
            // When the user click yes button then app will close
            finish();
        });

        // Set the Negative button with No name Lambda OnClickListener method is use of DialogInterface interface.
        builder.setNegativeButton("No", (DialogInterface.OnClickListener) (dialog, which) -> {
            // If user click no then dialog box is canceled.
            dialog.cancel();
        });

        // Create the Alert dialog
```

```
        AlertDialog alertDialog = builder.create();
        // Show the Alert Dialog box
        alertDialog.show();
    }
}
```

OUTPUT:





RESULT: The program to Implement an application that creates an alert upon receiving a message is completed and executed successfully.

PROGRAM 15

AIM: Develop a standard calculator application to perform basic calculations like addition, subtraction, multiplication and division.

ALGORITHM:

- Create a new Android project: Open Android Studio and create a new projectwith an appropriate name and package.
- Design the layout of the calculator: Design the user interface (UI) of the calculator using XML layout files. You'll need buttons for digits (0-9), operators (+, -, *, /), a clear button (C), and an equals button (=). You canarrange these buttons in a grid layout to resemble a traditional calculator.
- Write the Java code to handle user interactions and perform calculations.
- Test the calculator: Run the application on an Android device or emulator .

CODE:

Activity_main.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=".MainActivity">

    <EditText
        android:id="@+id/editText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="16dp"
        android:inputType="numberDecimal"
        android:textSize="24sp" />
    <EditText
        android:id="@+id/editText1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/editText"
        android:inputType="numberDecimal"
        android:textSize="24sp" />

    <Button
        android:id="@+id/buttonAdd"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```

```
        android:layout_below="@+id/editText1"
        android:text="Add" />

    <Button
        android:id="@+id/buttonSubtract"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/buttonAdd"
        android:layout_marginStart="16dp"
        android:text="Subtract" />

    <Button
        android:id="@+id/buttonMultiply"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/buttonAdd"
        android:layout_marginTop="16dp"
        android:text="Multiply" />

    <Button
        android:id="@+id/buttonDivide"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/buttonMultiply"
        android:layout_marginStart="16dp"
        android:text="Divide" />

    <TextView
        android:id="@+id/textResult"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/buttonDivide"
        android:layout_marginTop="16dp"
        android:text="Result:"
        android:textSize="24sp" />
</RelativeLayout>
```

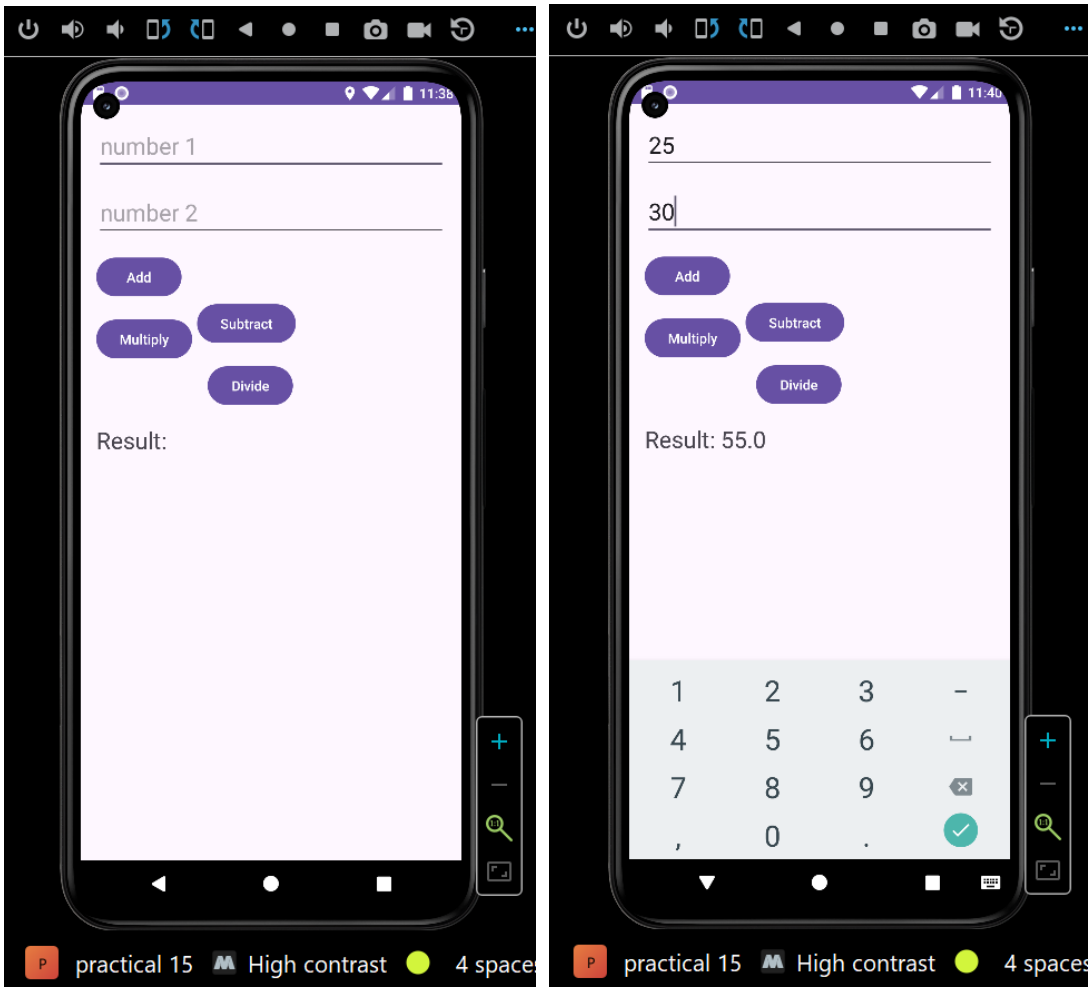
Main_Activity.Java code:-

```
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
public class MainActivity extends AppCompatActivity {

    private EditText editText;
```

```
private EditText editText1;
private Button buttonAdd, buttonSubtract, buttonMultiply, buttonDivide;
private TextView textResult;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    editText = findViewById(R.id.editText);
    editText1 = findViewById(R.id.editText1);
    buttonAdd = findViewById(R.id.buttonAdd);
    buttonSubtract = findViewById(R.id.buttonSubtract);
    buttonMultiply = findViewById(R.id.buttonMultiply);
    buttonDivide = findViewById(R.id.buttonDivide);
    textResult = findViewById(R.id.textResult);
    buttonAdd.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            calculate('+');
        }
    });
    buttonSubtract.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            calculate('-');
        }
    });
    buttonMultiply.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            calculate('*');
        }
    });
    buttonDivide.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            calculate('/');
        } });
}
private void calculate(char operator) {
    double result = 0;
    double operand1 = Double.parseDouble(editText.getText().toString());
    double operand2 = Double.parseDouble(editText1.getText().toString());
    switch (operator) {
        case '+':
            result = operand1 + operand2;
            break;
        case '-':
```

```
            result = operand1 - operand2;
            break;
        case '*':
            result = operand1 * operand2;
            break;
        case '/':
            if (operand2 != 0)
                result = operand1 / operand2;
            else
                textResult.setText("Cannot divide by zero!");
            break;
    }

    textResult.setText("Result: " + result);
  }
}
```

OUTPUT:



RESULT: The program to Develop a standard calculator application to perform basic calculations like addition, subtraction, multiplication and division. is completed and executed successfully.

PROGRAM 16

AIM: Simulate paintbrush application in android.

ALGORITHM:

- Set Up Android Studio: Install Android Studio and create a new project.
- Design Layout: Design the user interface (UI) for the painting application.
- Create a custom view or use the Android Canvas API to draw on the screen. This custom view will be where users paint. You'll need to handle touch events to detect user input for drawing.
- Draw with Paint: Create a Paint object that defines the properties of the brush(color, stroke width, etc.).
- Test Your App: Test your painting app thoroughly to ensure it works correctly for different drawing scenarios and screen sizes.

CODE:

Activity_main.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=".MainActivity">

   <LinearLayout
      android:id="@+id/linear"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:orientation="vertical">

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

         <ImageButton
            android:id="@+id/btn_undo"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:src="@drawable/ic_undo"
            android:text="Undo" />
```

```
            <ImageButton
                android:id="@+id/btn_save"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_weight="1"
                android:src="@drawable/ic_floppy_disk"
                android:text="Save" />

            <ImageButton
                android:id="@+id/btn_color"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_weight="1"
                android:src="@drawable/ic_colorpicker"
                android:text="Color" />

            <ImageButton
                android:id="@+id/btn_stroke"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_weight="1"
                android:src="@drawable/ic_paint_brush"
                android:text="Stroke" />
        </LinearLayout>

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical">
            <com.google.android.material.slider.RangeSlider
                android:id="@+id/rangebar"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:visibility="gone" />

        </LinearLayout>

    </LinearLayout>
    <com.raghav.paint.DrawView
        android:id="@+id/draw_view"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_below="@id/linear"
        android:layout_centerInParent="true"
        tools:ignore="MissingClass" />

</RelativeLayout>
```

Main_Activity.Java code:-

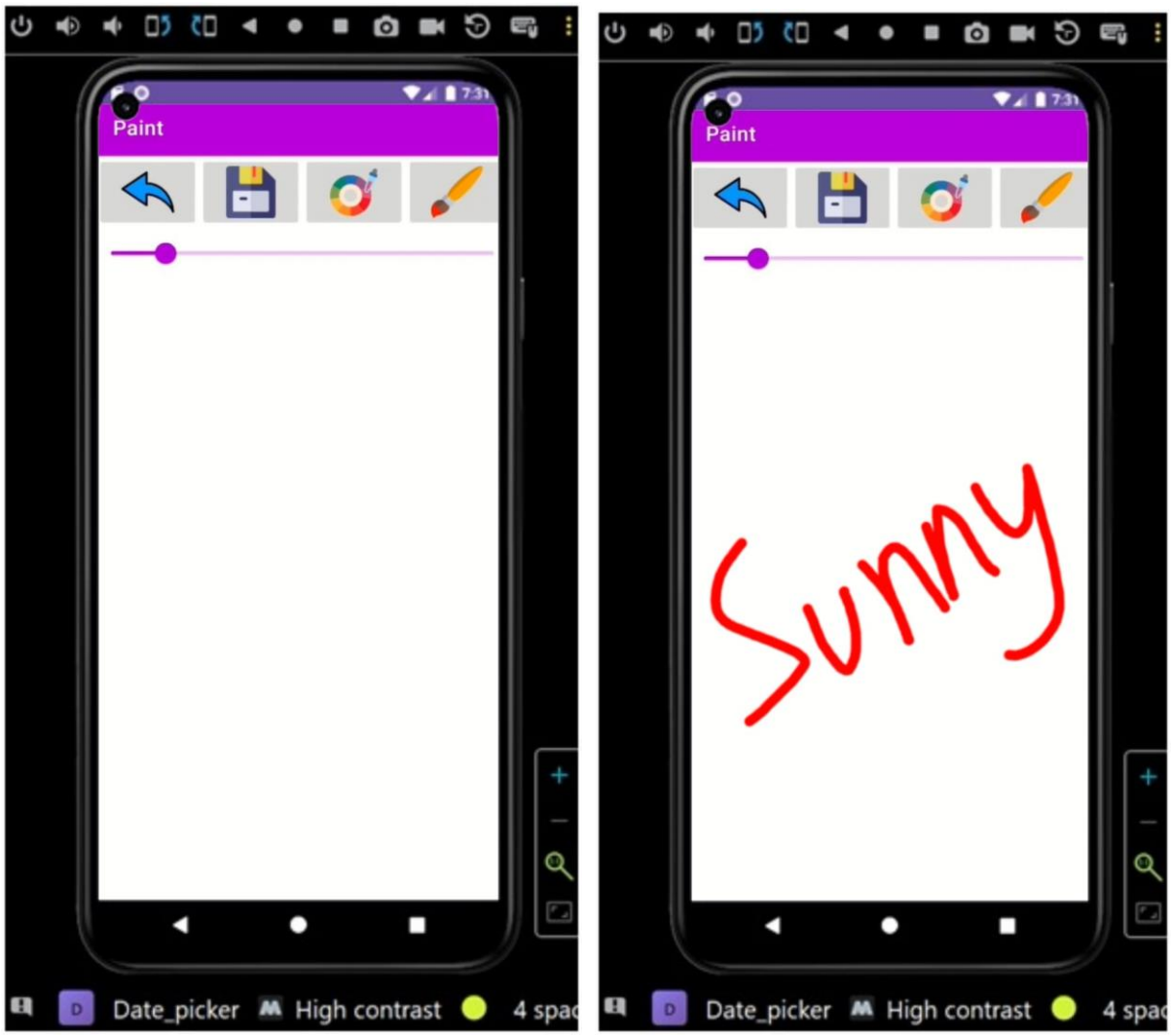
```
import android.content.ContentValues;
import android.graphics.Bitmap;
import android.graphics.Color;
import android.net.Uri;
import android.os.Bundle;
import android.os.Environment;
import android.provider.MediaStore;
import android.view.View;
import android.view.ViewTreeObserver;
import android.widget.ImageButton;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import com.google.android.material.slider.RangeSlider;
import java.io.OutputStream;
import petrov.kristiyan.colorpicker.ColorPicker;

public class MainActivity extends AppCompatActivity {

    private DrawView paint;
    private ImageButton save, color, stroke, undo;
    private RangeSlider rangeSlider;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        paint = (DrawView) findViewById(R.id.draw_view);
        rangeSlider = (RangeSlider) findViewById(R.id.rangebar);
        undo = (ImageButton) findViewById(R.id.btn_undo);
        save = (ImageButton) findViewById(R.id.btn_save);
        color = (ImageButton) findViewById(R.id.btn_color);
        stroke = (ImageButton) findViewById(R.id.btn_stroke);


        undo.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                paint.undo();
            }
        });

        save.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                Bitmap bmp = paint.save();
```

```
            OutputStream imageOutStream = null;

            ContentValues cv = new ContentValues();

            // name of the file
            cv.put(MediaStore.Images.Media.DISPLAY_NAME, "drawing.png");

            // type of the file
            cv.put(MediaStore.Images.Media.MIME_TYPE, "image/png");

            // location of the file to be saved
            cv.put(MediaStore.Images.Media.RELATIVE_PATH,
Environment.DIRECTORY_PICTURES);

            // get the Uri of the file which is to be created in the storage
            Uri uri =
getContentResolver().insert(MediaStore.Images.Media.EXTERNAL_CONTENT_URI, cv);
            try {
               // open the output stream with the above uri
               imageOutStream = getContentResolver().openOutputStream(uri);

               // this method writes the files in storage
               bmp.compress(Bitmap.CompressFormat.PNG, 100, imageOutStream);

               // close the output stream after use
               imageOutStream.close();
            } catch (Exception e) {
               e.printStackTrace();
            }
         }
      });
      // the color button will allow the user
      // to select the color of his brush
      color.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            final ColorPicker colorPicker = new ColorPicker(MainActivity.this);
            colorPicker.setOnFastChooseColorListener(new
ColorPicker.OnFastChooseColorListener() {
                  @Override
                  public void setOnFastChooseColorListener(int position, int color) {
                     // get the integer value of color
                     // selected from the dialog box and
                     // set it as the stroke color
                     paint.setColor(color);
                  }
                  @Override
                  public void onCancel() {
```

```
                        colorPicker.dismissDialog();
                    }
                })
                // set the number of color columns
                // you want to show in dialog.
                .setColumns(5)
                // set a default color selected
                // in the dialog
                .setDefaultColorButton(Color.parseColor("#000000"))
                .show();
        }
    });
    // the button will toggle the visibility of the RangeBar/RangeSlider
    stroke.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            if (rangeSlider.getVisibility() == View.VISIBLE)
                rangeSlider.setVisibility(View.GONE);
            else
                rangeSlider.setVisibility(View.VISIBLE);
        }
    });

    // set the range of the RangeSlider
    rangeSlider.setValueFrom(0.0f);
    rangeSlider.setValueTo(100.0f);

    // adding a OnChangeListener which will
    // change the stroke width
    // as soon as the user slides the slider
    rangeSlider.addOnChangeListener(new RangeSlider.OnChangeListener() {
        @Override
        public void onValueChange(@NonNull RangeSlider slider, float value, boolean
fromUser) {
            paint.setStrokeWidth((int) value);
        }
    });

    // pass the height and width of the custom view
    // to the init method of the DrawView object
    ViewTreeObserver vto = paint.getViewTreeObserver();
    vto.addOnGlobalLayoutListener(new ViewTreeObserver.OnGlobalLayoutListener()
{
        @Override
        public void onGlobalLayout() {
            paint.getViewTreeObserver().removeOnGlobalLayoutListener(this);
            int width = paint.getMeasuredWidth();
            int height = paint.getMeasuredHeight();
```

```
            paint.init(height, width);
        }
    });
  }
}
```

OUTPUT:



RESULT: The program to Simulate paintbrush application in android is completed and executed successfully.

PROGRAM 17

AIM: To develop an application of drawing an object in android.

ALGORITHM:

- Set Up Android Studio: Install Android Studio and create a new project.
- Design Layout: Design the user interface (UI) for drawing objects.
- Handle Touch Events: Implement touch event listeners to detect when theuser touches the screen, moves their finger, or lifts it.
- Draw Objects: Write code to draw various shapes and objects such as lines, circles, rectangles, etc., based on user input.
- Implement Undo/Redo Functionality (Optional): If desired, you can add functionality to undo or redo drawing actions.
- Add Color and Stroke Options (Optional): Provide options for users to choose different colors and stroke sizes for drawing objects.
- Run the application on an Android device or emulator.

CODE:

Activity_main.XML code:-

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

   <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="@string/hello_world" />

</RelativeLayout>
```

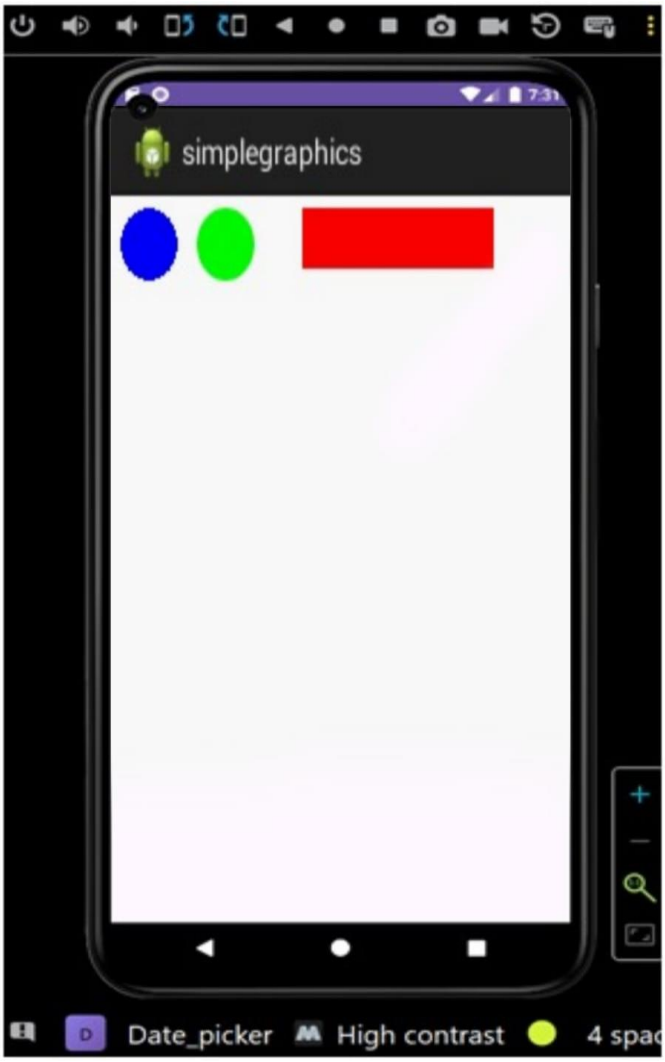
Main_Activity.Java code:-

```
import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.content.Context;
import android.graphics.Canvas;
```

```
import android.graphics.Color;
import android.graphics.Paint;
import android.view.View;
public class MainActivity extends Activity {
     DemoView demoview;
   @Override
   public void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      demoview = new DemoView(this);
      setContentView(demoview);
   }
    private class DemoView extends View{
      public DemoView(Context context){
         super(context);
      }
        @Override
        protected void onDraw(Canvas canvas) {
         super.onDraw(canvas);
           Paint paint = new Paint();
         paint.setStyle(Paint.Style.FILL);

         paint.setColor(Color.WHITE);
         canvas.drawPaint(paint);

         paint.setAntiAlias(false);
         paint.setColor(Color.BLUE);
         canvas.drawCircle(20, 20, 15, paint);

         paint.setAntiAlias(true);
         paint.setColor(Color.GREEN);
         canvas.drawCircle(60, 20, 15, paint);

         paint.setAntiAlias(false);
         paint.setColor(Color.RED);
         canvas.drawRect(100, 5, 200, 30, paint);



         canvas.rotate(-45);
         paint.setStyle(Paint.Style.FILL);
         canvas.drawText("Graphics Rotation", 40, 180, paint);
           canvas.restore();
      }
```

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

OUTPUT:



RESULT: The program to develop an application of drawing an object in android is completed and executed successfully.

PROGRAM 18

AIM: Implement a webview application in android.

ALGORITHM:

- Set Up Android Studio: Install Android Studio and create a new project.
- Configure WebView in Java Code: In your activity file (typically MainActivity.java), find the WebView by its ID and configure it.
- Load a Web Page: Use the WebView's loadUrl() method to load a web pageinto the WebView. You can load a URL from the internet or from local assets.
- Add Permissions (Optional): If your app requires internet access.
- Run the application on an Android device or emulator.

CODE:

Activity_main.XML code:-

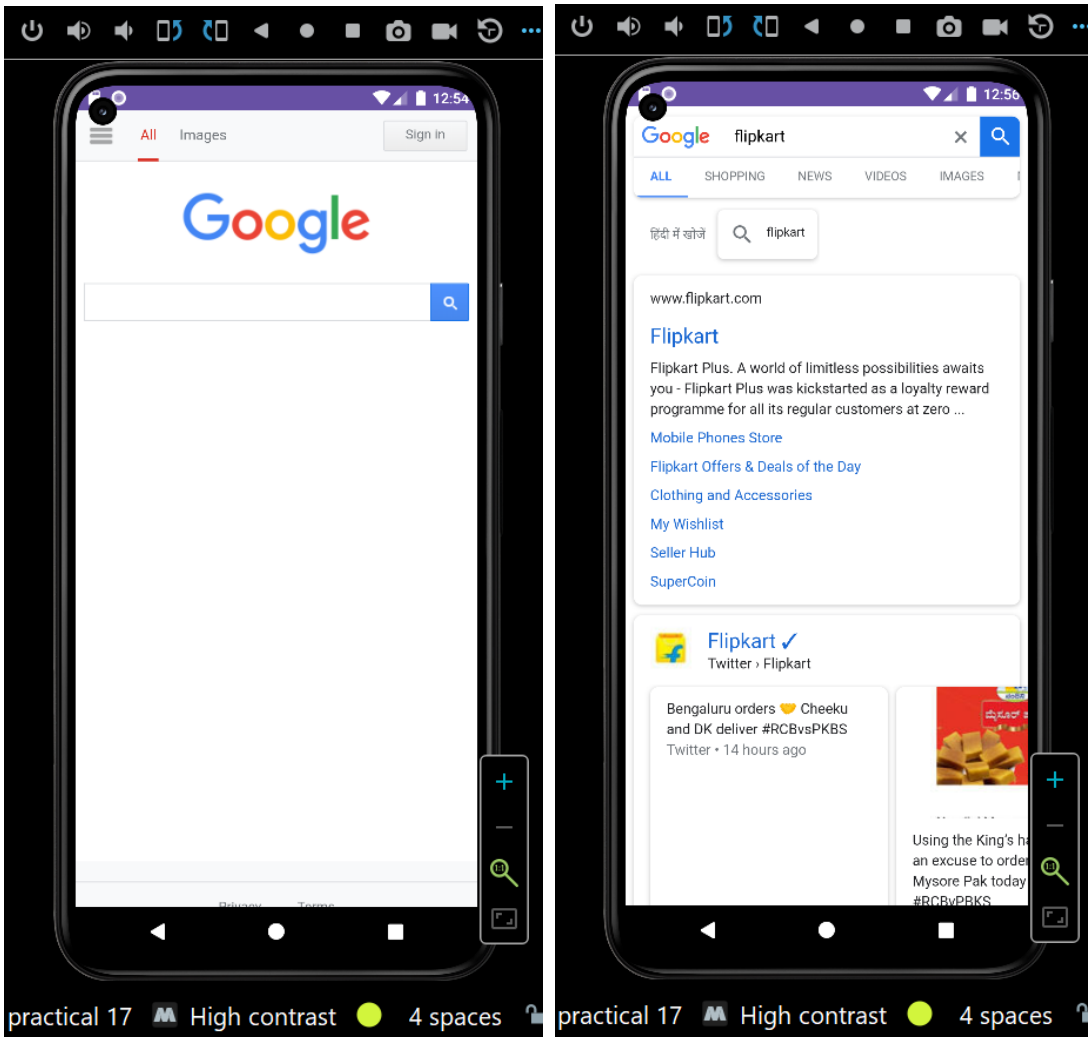
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android = "http://schemas.android.com/apk/res/android"
  android:id = "@+id/parent"
  xmlns:tools = "http://schemas.android.com/tools"
  android:layout_width = "match_parent"
  android:layout_height = "match_parent"
  tools:context = ".MainActivity">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

Main_Activity.Java code:-

```
import android.os.Bundle;
import android.webkit.WebSettings;
import android.webkit.WebView;
import android.webkit.WebViewClient;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {
   private WebView webView;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_main);
        WebView webView = (WebView)findViewById(R.id.webView);
        webView.setWebViewClient(new WebViewClient());
        webView.loadUrl("https://www.google.com");
        WebSettings webSettings = webView.getSettings();
        webSettings.setJavaScriptEnabled(true);
    }
    public void onBackPressed(){
        if (webView.canGoBack()){
            webView.goBack();
        } else {
            super.onBackPressed();
        }
    }
}
```

OUTPUT:



RESULT: The program to Implement a webview application in android is completed and executed successfully.