PRACTICAL 13

Aim: Create an Android application which reads contact name from phone, and display contact name to List View. On long press of any of the list item a context menu shall be display having an option called "Send SMS". On click of "Send SMS" option, an inbuilt messaging application should be open with an appropriate data. (I.e., pass the phone number of selected contact to messaging application) (Note: Make the use of content provider).

Code:
JAVA FILE: (MainActivity.java)

```java
package com.example.practical13;
import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.database.Cursor;
import android.net.Uri;
import android.os.Bundle;
import android.provider.ContactsContract;
import android.view.ContextMenu;
import android.view.MenuItem;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;
import java.util.ArrayList;
public class MainActivity extends AppCompatActivity {
        private static final int REQUEST_CONTACTS_PERMISSION = 1;
        private ArrayList<String> contactNames = new ArrayList<>();
        private ArrayAdapter<String> adapter;
        @Override
        protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        ListView listView = findViewById(R.id.listView);
        adapter = new ArrayAdapter<>(this, android.R.layout.simple_list_item_1,
contactNames);
        listView.setAdapter(adapter);
```

```java
        // Check for Contacts permission
        if (ContextCompat.checkSelfPermission(this,
Manifest.permission.READ_CONTACTS)
                != PackageManager.PERMISSION_GRANTED) {
        ActivityCompat.requestPermissions(this,
                new String[]{Manifest.permission.READ_CONTACTS},
                REQUEST_CONTACTS_PERMISSION);
        } else {
        fetchContacts();
        }
        // Register ListView for context menu
        registerForContextMenu(listView);
        }
        private void fetchContacts() {
        Cursor cursor = getContentResolver().query(
                ContactsContract.CommonDataKinds.Phone.CONTENT_URI,
                null,
                null,
                null,
                null);
        while (cursor.moveToNext()) {
        String name =
cursor.getString(cursor.getColumnIndex(ContactsContract.CommonDataKinds.Phone.DISPL
AY_NAME));
        contactNames.add(name);
        }
        cursor.close();
        adapter.notifyDataSetChanged();
        }
        @Override
        public void onRequestPermissionsResult(int requestCode, @NonNull String[]
permissions, @NonNull int[] grantResults) {
        super.onRequestPermissionsResult(requestCode, permissions, grantResults);
        if (requestCode == REQUEST_CONTACTS_PERMISSION) {
        if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
                fetchContacts();
        }
        }
        }
        @Override
        public void onCreateContextMenu(ContextMenu menu, View v,
ContextMenu.ContextMenuInfo menuInfo) {
        super.onCreateContextMenu(menu, v, menuInfo);
```

```java
        getMenuInflater().inflate(R.menu.context_menu, menu);
        }
        @Override
        public boolean onContextItemSelected(MenuItem item) {
        AdapterView.AdapterContextMenuInfo info =
(AdapterView.AdapterContextMenuInfo) item.getMenuInfo();
        String selectedContactName = contactNames.get(info.position);
        if (item.getItemId() == R.id.send_sms_option) {
        // Get the phone number associated with the selected contact
        String phoneNumber = getPhoneNumberForContact(selectedContactName);
        if (phoneNumber != null) {
                // Create an intent to open the messaging app
                Intent intent = new Intent(Intent.ACTION_SENDTO);
                intent.setData(Uri.parse("smsto:" + phoneNumber));
                startActivity(intent);
        }
        return true;
        }
        return super.onContextItemSelected(item);
        }
        private String getPhoneNumberForContact(String contactName) {
        Cursor cursor = getContentResolver().query(
                ContactsContract.CommonDataKinds.Phone.CONTENT_URI,
                null,
                ContactsContract.CommonDataKinds.Phone.DISPLAY_NAME + " = ?",
                new String[]{contactName},
                null);
        if (cursor != null && cursor.moveToFirst()) {
        String phoneNumber =
cursor.getString(cursor.getColumnIndex(ContactsContract.CommonDataKinds.Phone.NUMBER));
        cursor.close();
        return phoneNumber;
        }
        return null;
        }
}
```

XML FILE: (activity_main.xml)
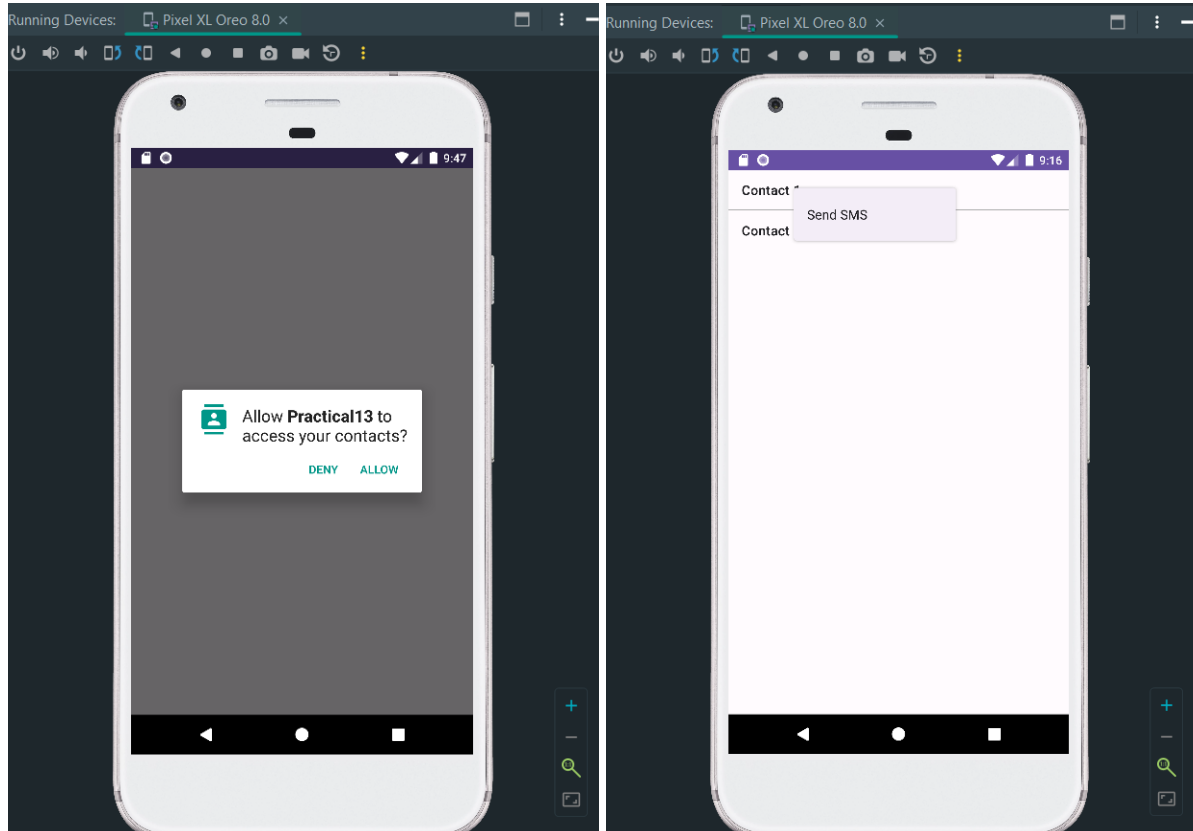
```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:app="http://schemas.android.com/apk/res-auto"
        xmlns:tools="http://schemas.android.com/tools"
        android:layout_width="match_parent"
```

```
        android:layout_height="match_parent"
        android:orientation="vertical"
        tools:context=".MainActivity">
        <ListView
        android:id="@+id/listView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:divider="@android:color/darker_gray"
        android:dividerHeight="1dp" />
</LinearLayout>
```

XML FILE(res/menu): (context_menu.xml)
```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
        <item
        android:id="@+id/send_sms_option"
        android:title="Send SMS" />
</menu>
```

Output:

Enrollment No: 202203103510097