

SRI SIVASUBRAMANIYA NADAR COLLEGE OF ENGINEERING

(AN AUTONOMOUS INSTITUTION,
AFFILIATED TO ANNA UNIVERSITY)

Rajiv Gandhi Salai (OMR), Kalavakkam - 603 110.

LABORATORY RECORD

NAME : SHIVCHARAN T
Reg. No. : 205001100
Dept. : CSE Sem. : VII Sec. : B

ssn

**SRI SIVASUBRAMANIYA NADAR
COLLEGE OF ENGINEERING, CHENNAI**
(AN AUTONOMOUS INSTITUTION, AFFILIATED TO ANNA UNIVERSITY)

BONAFIDE CERTIFICATE

Certified that this is the bonafide record of the practical work done in the

UCS1711 - Mobile Application Development Laboratory by

Name SHIVCHARAN T

Register Number 205001100

Semester VII

Branch CSE, 2020-2024

Sri Sivasubramaniya Nadar College of Engineering, Kalavakkam.

During the Academic year 2023-24

A. Balaji
Faculty 22/11/23


Head of the Department

Submitted for the.....Practical Examination held at SSNCE
on.....

Internal Examiner

External Examiner

INDEX

Name : Shivcharan T Reg. No. 205001100

Sem : VII Sec : B

Ex. No.	Date of Expt.	Title of the Experiment	Page No.	Signature of the Faculty	Remarks
1	22/08/2023	Application using GUI components, font, color, layout manager & event listeners	1	7	
2	29/08/2023	Simulation of a keyboard	14	7	
3	05/09/2023	Application development using basic graphical primitives	26		
4	12/09/2023	Android Application development with database	36		
5	19/09/2023	Android App using Multi-Threading	56	78	
6	26/09/2023	Android App for Location Tracking	66		
7	3/10/2023	Android App to read / write file from (to SD - Card)	78		
8	10/10/2023	Android App to send SMS & Notification	95	7	
9	17/10/2023	Menu driven application	109	7	
10	24/10/2023	Alarm Clock - Android App	115	78	
11	31/10/2023	Hybrid Mobile App	122	7	
12	07/11/2023	Mini Project	139		

Exercise 1 - Designing Health Insurance Form using GUI Components

T Shivcharan
205001100
CSE-B

Aim

Generate a Health Insurance registration form to register the patient details under each group.

Patient Details - Title, Patient Name, Patient Phone Number (Mobile or Landline use Checkbox), Address, Age, DOB, Gender, Marital Status

Employer Details - Patient Employer, Employment Status (full time, parttime, unemployed, retired, student, other Checkbox)

Emergency contact Details - Name, Relationship, Address, Phone Number

Use Submit (Button) to submit the details and display the contents. Use the Reset button to clear the form. Display using Table layout.

Code

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView 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">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginLeft="16dp"
        android:layout_marginRight="16dp"
        android:layout_marginBottom="8dp"
        android:orientation="vertical"
        tools:context=".MainActivity">
        <TextView
            android:id="@+id/titleTextView"
            android:layout_width="match_parent"
            android:layout_height="128dp"
            android:gravity="center"
            android:text="Health Insurance Registration"
            android:textSize="32sp"
            android:textStyle="bold"
            android:typeface="serif" >
        <TextView
            android:id="@+id/title"
            android:layout_width="match_parent"
            android:layout_height="32dp"
            android:gravity="left"
            android:text="Patient Details"
            android:textSize="22sp"
            android:textStyle="bold"
            android:typeface="serif" >
        <TextView
            android:id="@+id/name"
```

```
    android:layout_width="match_parent"
    android:layout_height="26dp"
    android:layout_marginTop="16dp"
    android:gravity="left"
    android:text="Name"
    android:textSize="18sp"
    android:textStyle="bold"
    android:typeface="normal" >
<EditText
    android:id="@+id/editName"
    android:layout_width="match_parent"
    android:layout_height="54dp"
    android:hint="Name:"
    android:inputType="text"
    android:textColor="#0000FF"
    android:typeface="sans" >
<TextView
    android:id="@+id/number"
    android:layout_width="match_parent"
    android:layout_height="26dp"
    android:layout_marginTop="16dp"
    android:gravity="left"
    android:text="Number"
    android:textSize="18sp"
    android:textStyle="bold"
    android:typeface="normal" >
<RadioGroup
    android:id="@+id/radio_group_id"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="4dp">
    <RadioButton
        android:id="@+id/phone_radio_button_id"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Phone" >
    <RadioButton
        android:id="@+id/landline_radio_button_id"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Landline" >
</RadioGroup>
<EditText
    android:id="@+id/editNumber"
    android:layout_width="match_parent"
    android:layout_height="54dp"
    android:hint="Number:"
    android:inputType="phone" >
<TextView
    android:id="@+id/address"
    android:layout_width="match_parent"
```

```
    android:layout_height="26dp"
    android:layout_marginTop="16dp"
    android:gravity="left"
    android:text="Address"
    android:textSize="18sp"
    android:textStyle="bold"
    android:typeface="normal" >
<EditText
    android:id="@+id/editAddress"
    android:layout_width="match_parent"
    android:layout_height="54dp"
    android:hint="Address:"
    android:inputType="text"
    android:textColor="#0000FF"
    android:typeface="sans" >
<TextView
    android:id="@+id/age"
    android:layout_width="match_parent"
    android:layout_height="26dp"
    android:layout_marginTop="16dp"
    android:gravity="left"
    android:text="Age"
    android:textSize="18sp"
    android:textStyle="bold"
    android:typeface="normal" >
<EditText
    android:id="@+id/editAge"
    android:layout_width="match_parent"
    android:layout_height="54dp"
    android:hint="Age:"
    android:inputType="number"
    android:textColor="#0000FF"
    android:typeface="sans" >
<TextView
    android:id="@+id/dob"
    android:layout_width="match_parent"
    android:layout_height="26dp"
    android:layout_marginTop="16dp"
    android:gravity="left"
    android:text="DOB"
    android:textSize="18sp"
    android:textStyle="bold"
    android:typeface="normal" >
<DatePicker
    android:id="@+id/dobPicker"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center" >
```

```
<TextView
    android:id="@+id/gender"
    android:layout_width="match_parent"
    android:layout_height="26dp"
    android:layout_marginTop="16dp"
    android:gravity="left"
    android:text="Gender"
    android:textSize="18sp"
    android:textStyle="bold"
    android:typeface="normal" >
<RadioGroup
    android:id="@+id/gender_group"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="4dp">
    <RadioButton
        android:id="@+id/male"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Male" >
    <RadioButton
        android:id="@+id/female"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Female" >
</RadioGroup>
<TextView
    android:id="@+id/marital_status"
    android:layout_width="match_parent"
    android:layout_height="26dp"
    android:layout_marginTop="16dp"
    android:gravity="left"
    android:text="Marital Status"
    android:textSize="18sp"
    android:textStyle="bold"
    android:typeface="normal" >
<Spinner
    android:id="@+id/marital_status_spinner"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:entries="@array/marital_status_options" >
<TextView
    android:id="@+id/employer"
    android:layout_width="match_parent"
    android:layout_height="32dp"
    android:layout_marginTop="48dp"
    android:gravity="left"
    android:text="Employer Details"
    android:textSize="22sp"
```

```
        android:textStyle="bold"
        android:typeface="serif" >
<TextView
    android:id="@+id/employer_name"
    android:layout_width="match_parent"
    android:layout_height="26dp"
    android:layout_marginTop="16dp"
    android:gravity="left"
    android:text="Employer Name"
    android:textSize="18sp"
    android:textStyle="bold"
    android:typeface="normal" >
<EditText
    android:id="@+id/editEmployerName"
    android:layout_width="match_parent"
    android:layout_height="54dp"
    android:hint="Employer Name:"
    android:inputType="text"
    android:textColor="#0000FF"
    android:typeface="sans" >
<TextView
    android:id="@+id/employment_status"
    android:layout_width="match_parent"
    android:layout_height="26dp"
    android:layout_marginTop="16dp"
    android:gravity="left"
    android:text="Employment Status"
    android:textSize="18sp"
    android:textStyle="bold"
    android:typeface="normal" >
<Spinner
    android:id="@+id/employment_status_spinner"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:entries="@array/employment_status_options" >
<TextView
    android:id="@+id/emergency"
    android:layout_width="match_parent"
    android:layout_height="32dp"
    android:layout_marginTop="48dp"
    android:gravity="left"
    android:text="Emergency Details"
    android:textSize="22sp"
    android:textStyle="bold"
    android:typeface="serif" >
<TextView
    android:id="@+id/emergency_name"
    android:layout_width="match_parent"
    android:layout_height="26dp"
    android:layout_marginTop="16dp"
    android:gravity="left"
```

```
    android:text="Emergency Contact Name"
    android:textSize="18sp"
    android:textStyle="bold"
    android:typeface="normal" >
<EditText
    android:id="@+id/edit_emergency_name"
    android:layout_width="match_parent"
    android:layout_height="54dp"
    android:hint="Employer Contact Name:"
    android:inputType="text"
    android:textColor="#0000FF"
    android:typeface="sans" >
<TextView
    android:id="@+id/emergency_relationship"
    android:layout_width="match_parent"
    android:layout_height="26dp"
    android:layout_marginTop="16dp"
    android:gravity="left"
    android:text="Relationship"
    android:textSize="18sp"
    android:textStyle="bold"
    android:typeface="normal" >
<EditText
    android:id="@+id/edit_emergency_relationship"
    android:layout_width="match_parent"
    android:layout_height="54dp"
    android:hint="Relationship:"
    android:inputType="text"
    android:textColor="#0000FF"
    android:typeface="sans" >
<TextView
    android:id="@+id/emergency_address"
    android:layout_width="match_parent"
    android:layout_height="26dp"
    android:layout_marginTop="16dp"
    android:gravity="left"
    android:text="Address"
    android:textSize="18sp"
    android:textStyle="bold"
    android:typeface="normal" >
<EditText
    android:id="@+id/edit_emergency_address"
    android:layout_width="match_parent"
    android:layout_height="54dp"
    android:hint="Address:"
    android:inputType="text"
    android:textColor="#0000FF"
    android:typeface="sans" >
<TextView
```

```
        android:id="@+id/emergency_phone"
        android:layout_width="match_parent"
        android:layout_height="26dp"
        android:layout_marginTop="16dp"
        android:gravity="left"
        android:text="Phone"
        android:textSize="18sp"
        android:textStyle="bold"
        android:typeface="normal" >
    <EditText
        android:id="@+id/edit_emergency_phone"
        android:layout_width="match_parent"
        android:layout_height="54dp"
        android:hint="Phone Number:"
        android:inputType="phone"
        android:textColor="#0000FF"
        android:typeface="sans" >
    <Button
        android:id="@+id/submit"
        android:layout_width="match_parent"
        android:layout_height="64dp"
        android:layout_marginTop="32dp"
        android:backgroundTint="#4CAF50"
        android:gravity="center"
        android:hint="Register"
        android:onClick="openTableActivity" > <Button

        android:id="@+id/reset"
        android:layout_width="match_parent"
        android:layout_height="64dp"
        android:layout_marginTop="16dp"
        android:backgroundTint="#F44336"
        android:gravity="center"
        android:hint="Reset"
        android:onClick="resetFormFields" >
    /LinearLayout>
/ScrollView>
```

MainActivity.java

```
package com.example.a1;
import android.content.Intent;
import android.support.v7.app.AppCompatActivity; import
android.os.Bundle;
import android.view.View;
import android.widget.DatePicker;
import android.widget.EditText;
import android.widget.RadioButton;
import android.widget.Spinner;
import java.util.Calendar;
public class MainActivity extends AppCompatActivity {
```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
}
public void openTableActivity(View view) {
    Intent intent = new Intent(this, TableActivity.class);
    intent.putExtra("name", ((EditText)
findViewById(R.id.editName)).getText().toString()); if (((RadioButton)
findViewById(R.id.phone_radio_button_id)).isChecked())
        intent.putExtra("number_type", "Phone");
    else if (((RadioButton)
findViewById(R.id.phone_radio_button_id)).isChecked())
        intent.putExtra("number_type", "Landline");
    intent.putExtra("number", ((EditText)
findViewById(R.id.editNumber)).getText().toString());
    intent.putExtra("address", ((EditText)
findViewById(R.id.editAddress)).getText().toString());
    intent.putExtra("age", ((EditText)
findViewById(R.id.editAge)).getText().toString());
    DatePicker datePicker = findViewById(R.id.dobPicker);
    int day = datePicker.getDayOfMonth();
    int month = datePicker.getMonth();
    int year = datePicker.getYear();
    Calendar calendar = Calendar.getInstance();
    calendar.set(year, month, day);
    intent.putExtra("dob", calendar.getTime().toString());
    if (((RadioButton) findViewById(R.id.male)).isChecked())
intent.putExtra("gender", "Male");
    else if (((RadioButton) findViewById(R.id.female)).isChecked())
        intent.putExtra("gender", "Female");
    intent.putExtra("marital_status", ((Spinner)
findViewById(R.id.marital_status_spinner)).getSelectedItem().toString());
    intent.putExtra("employer_name", ((EditText)
findViewById(R.id.editEmployerName)).getText().toString());
    intent.putExtra("employment_status", ((Spinner)
findViewById(R.id.employment_status_spinner)).getSelectedItem().toString());
    intent.putExtra("emergency_name", ((EditText)
findViewById(R.id.edit_emergency_name)).getText().toString());
    intent.putExtra("emergency_relationship", ((EditText)
findViewById(R.id.edit_emergency_relationship)).getText().toString());
    intent.putExtra("emergency_address", ((EditText)
findViewById(R.id.edit_emergency_address)).getText().toString());
    intent.putExtra("emergency_phone", ((EditText)
findViewById(R.id.edit_emergency_phone)).getText().toString());
    startActivity(intent);
}
public void resetFormFields(View view) {

```

```

        ((EditText) findViewById(R.id.editName)).setText("");
        ((RadioButton)
        findViewById(R.id.phone_radio_button_id)).setChecked(false); ((RadioButton)
        findViewById(R.id.landline_radio_button_id)).setChecked(false); ((EditText)
        findViewById(R.id.editNumber)).setText("");
        ((EditText) findViewById(R.id.editAddress)).setText("");
        ((EditText) findViewById(R.id.editAge)).setText("");
        ((DatePicker) findViewById(R.id.dobPicker)).updateDate(1970, 0, 1); ((RadioButton)
        findViewById(R.id.male)).setChecked(false);
        ((RadioButton) findViewById(R.id.female)).setChecked(false); ((Spinner)
        findViewById(R.id.marital_status_spinner)).setSelection(0);
        ((EditText) findViewById(R.id.editEmployerName)).setText(""); ((Spinner)
        findViewById(R.id.employment_status_spinner)).setSelection(0);
        ((EditText) findViewById(R.id.edit_emergency_name)).setText(""); ((EditText)
        findViewById(R.id.edit_emergency_relationship)).setText(""); ((EditText)
        findViewById(R.id.edit_emergency_address)).setText(""); ((EditText)
        findViewById(R.id.edit_emergency_phone)).setText(""); }

}


```

activity_table.xml

```

<?xml version="1.0" encoding="utf-8"?>
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:stretchColumns="*">
    <TableRow>
        <TextView
            android:id="@+id/textView_title1"
            android:layout_width="100dp"
            android:layout_height="wrap_content"
            android:text="Patient details"
            android:textAlignment="center"
            android:textSize="48px"
            android:textStyle="bold" >
    /TableRow>
    <TableRow>
        <TextView
            android:id="@+id/textView_name"
            android:layout_width="100dp"
            android:layout_height="wrap_content"
            android:text="Name" >
        <TextView
            android:id="@+id/textView_name_value"
            android:layout_width="match_parent"
            android:layout_height="wrap_content" >/TableRow>
    <TableRow>
        <TextView
            android:id="@+id/textView_number"

```

```
    android:layout_width="100dp"
    android:layout_height="wrap_content"
    android:text="Number" >
<TextView
    android:id="@+id/textView_number_value"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >/TableRow>
<TableRow>
<TextView
    android:id="@+id/textView_address"
    android:layout_width="100dp"
    android:layout_height="wrap_content"
    android:text="Address" >
<TextView
    android:id="@+id/textView_address_value"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >/TableRow>
<TableRow>
<TextView
    android:id="@+id/textView_age"
    android:layout_width="100dp"
    android:layout_height="wrap_content"
    android:text="Age" >
<TextView
    android:id="@+id/textView_age_value"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >/TableRow>
<TableRow>
<TextView
    android:id="@+id/textView_dob"
    android:layout_width="100dp"
    android:layout_height="wrap_content"
    android:text="DOB" >
<TextView
    android:id="@+id/textView_dob_value"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >/TableRow>
<TableRow>
<TextView
    android:id="@+id/textView_gender"
    android:layout_width="100dp"
    android:layout_height="wrap_content"
    android:text="Gender" >
<TextView
    android:id="@+id/textView_gender_value"
    android:layout_width="match_parent"
```

```
        android:layout_height="wrap_content" >
    /TableRow>
    <TableRow>
        <TextView
            android:id="@+id/textView_marital_status"
            android:layout_width="100dp"
            android:layout_height="wrap_content"
            android:text="Marital Status" >
        <TextView
            android:id="@+id/textView_marital_status_value"
            android:layout_width="match_parent"
            android:layout_height="wrap_content" >
    /TableRow>
    <TableRow>
        <TextView
            android:id="@+id/textView_title2"
            android:layout_width="100dp"
            android:layout_height="wrap_content"
            android:layout_marginTop="32px"
            android:text="Employer details"
            android:textAlignment="center"
            android:textSize="48px"
            android:textStyle="bold" >
    /TableRow>
    <TableRow>
        <TextView
            android:id="@+id/textView_employer_name"
            android:layout_width="100dp"
            android:layout_height="wrap_content"
            android:text="Employer Name" >
        <TextView
            android:id="@+id/textView_employer_name_value"
            android:layout_width="match_parent"
            android:layout_height="wrap_content" >
    /TableRow>
    <TableRow>
        <TextView
            android:id="@+id/textView_employment_status"
            android:layout_width="100dp"
            android:layout_height="wrap_content"
            android:text="Employement Status" >
        <TextView
            android:id="@+id/textView_employment_status_value"
            android:layout_width="match_parent"
            android:layout_height="wrap_content" >
    /TableRow>
    <TableRow>
        <TextView
            android:id="@+id/textView_title3"
            android:layout_width="100dp"
```

```
        android:layout_height="wrap_content"
        android:layout_marginTop="32px"
        android:text="Emergency details"
        android:textAlignment="center"
        android:textSize="48px"
        android:textStyle="bold" >
    
```

```
<TableRow>
```

```
<TableRow>
    <TextView
        android:id="@+id/textView_emergency_name"
        android:layout_width="100dp"
        android:layout_height="wrap_content"
        android:text="Emergency name" >

```

```
    <TextView
        android:id="@+id/textView_emergency_name_value"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" >
```

```
</TableRow>
```

```
<TableRow>
    <TextView
        android:id="@+id/textView_emergency_relationship"
        android:layout_width="100dp"
        android:layout_height="wrap_content"
        android:text="Emergency relationship" >

```

```
    <TextView
        android:id="@+id/textView_emergency_relationship_value"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" >
```

```
</TableRow>
```

```
<TableRow>
    <TextView
        android:id="@+id/textView_emergency_address"
        android:layout_width="100dp"
        android:layout_height="wrap_content"
        android:text="Emergency address" >

```

```
    <TextView
        android:id="@+id/textView_emergency_address_value"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" >
```

```
</TableRow>
```

```
<TableRow>
    <TextView
        android:id="@+id/textView_emergency_phone"
        android:layout_width="100dp"
        android:layout_height="wrap_content"
        android:text="Emergency phone" >

```

```
    <TextView
        android:id="@+id/textView_emergency_phone_value"
        android:layout_width="match_parent"
```

```

        android:layout_height="wrap_content" >
    /TableRow>
/>TableLayout>
TableActivity.java
package com.example.a1;
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.TextView;
public class TableActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_table);
        Intent intent = getIntent();
        Bundle extras = intent.getExtras();
        if (extras == null) return;
        ((TextView)
        findViewById(R.id.textView_name_value)).setText(extras.getString("name")); ((TextView)
        findViewById(R.id.textView_number_value)).setText(extras.getString("number")); ((TextView)
        findViewById(R.id.textView_address_value)).setText(extras.getString("address"));
        ((TextView)
        findViewById(R.id.textView_age_value)).setText(extras.getString("age")); ((TextView)
        findViewById(R.id.textView_dob_value)).setText(extras.getString("dob")); ((TextView)
        findViewById(R.id.textView_gender_value)).setText(extras.getString("gender")); ((TextView)
        findViewById(R.id.textView_marital_status_value)).setText(extras.getString("marital_status"));
        ((TextView)
        findViewById(R.id.textView_employer_name_value)).setText(extras.getString("employer_name"));
        ((TextView)
        findViewById(R.id.textView_employment_status_value)).setText(extras.getString("employment_status"));
        ((TextView)
        findViewById(R.id.textView_emergency_name_value)).setText(extras.getString("emergency_name"));
        ((TextView)
        findViewById(R.id.textView_emergency_relationship_value)).setText(extras.getString("emergency_relationship"));
        ((TextView)
        findViewById(R.id.textView_emergency_address_value)).setText(extras.getString("emergency_address"));
        ((TextView)
        findViewById(R.id.textView_emergency_phone_value)).setText(extras.getString("emergency_phone"));
    }
}
strings.xml
<resources>
    <string name="app_name">A1 /string>
    <string-array name="marital_status_options">
        <item>Never Married /item>
        <item>Married /item>
        <item>Widowed /item>

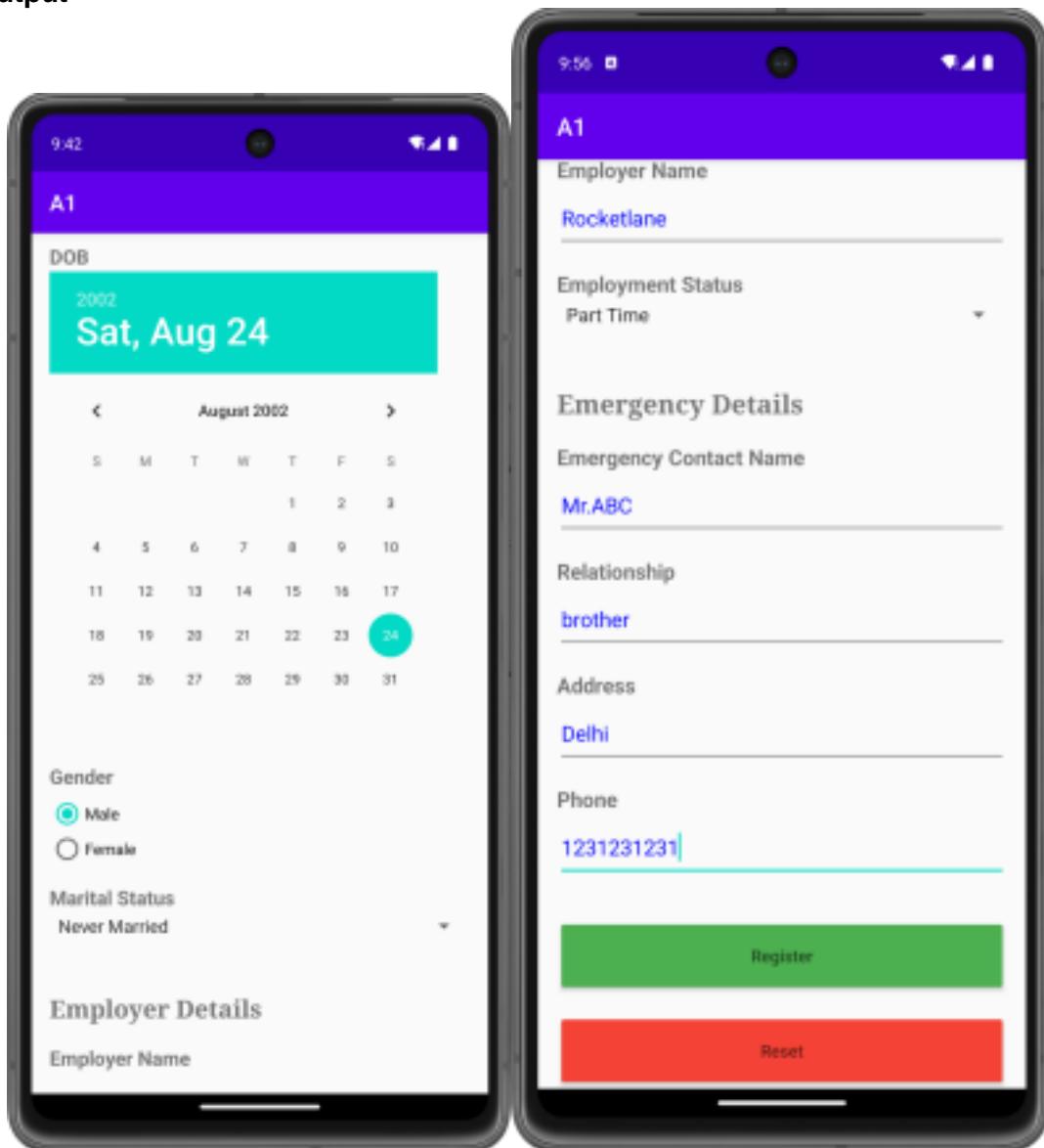
```

```

<item>Divorced /item>
<item>Separated /item>
/string-array>
<string-array name="employment_status_options">
    <item>Full Time /item>
    <item>Part Time /item>
    <item>Unemployed /item>
    <item>Retired /item>
    <item>Student /item>
    <item>Other /item>
/string-array>
<string name="title_activity_table">All Details /string> /resources>

```

Output



Learning outcomes

Basic GUI components and layouts that are available
Passing data between activities

Exercise 2 - Keyboard application

T Shivcharan
205001100
CSE-A

Aim

To implement a keyboard application using Android Studio

Code

activity_main.xml

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

    <TextView
        android:id="@+id/text_bar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="6dp"
        android:layout_marginTop="24dp"
        android:layout_marginRight="6dp"
        android:hint="Enter text..."
        android:text=""
        android:textSize="24sp" />

    <GridLayout
        android:id="@+id/keyboard"
        android:layout_width="match_parent"
        android:layout_height="200dp"
        android:layout_alignParentBottom="true"
        android:background="#202020"
        android:padding="1dp" />
</RelativeLayout>
```

MainActivity.java

```
package com.example.qwerty;

import androidx.appcompat.app.AppCompatActivity;
import androidx.core.content.res.ResourcesCompat;
import android.os.Bundle;
import android.util.DisplayMetrics;
import android.view.Gravity;
import android.widget.Button;
import android.widget.GridLayout;
import android.widget.TextView;
import android.graphics.Typeface;
```

```

public class MainActivity extends AppCompatActivity {
    private final String[][] letterKeys = {
        {"Q", "W", "E", "R", "T", "Y", "U", "I", "O", "P"},
        {"A", "S", "D", "F", "G", "H", "J", "K", "L"},
        {"↑", "Z", "X", "C", "V", "B", "N", "M", "←"},
        {"?123", " ", ",", " ", ".," , "↔" }
    };
    private final String[][] numberKeys = {
        {"1", "2", "3", "4", "5", "6", "7", "8", "9", "0"},
        {"@", "#", "₹", "_", "&", "-", "+", "(", ")","/", "*",
        {"\"", "'", ".", ";", "!", "?", "<", ">","=", "←" },
        {"ABC", " ", " ", " ", ".," , "↔" }
    };
    private TextView textBar;

    private GridLayout keyboard;
    private DisplayMetrics displayMetrics;
    private Boolean isCaps = false;
    private Typeface customFont;

    private void init() {
        textBar = findViewById(R.id.text_bar);
        textBar.setText("|");
        keyboard = findViewById(R.id.keyboard);

        displayMetrics = new DisplayMetrics();
        getWindowManager().getDefaultDisplay().getMetrics(displayMetrics);

        customFont = ResourcesCompat.getFont(this, R.font.oswald);
    }
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        init();
        initKeyboard(letterKeys);
    }

    private void initKeyboard(String[][] keys) {
        keyboard.removeAllViews();

        int rowIndex = 0;
        for (String[] row : keys) {
            int collIndex = 0;
            int width = displayMetrics.widthPixels / (row.length + 4);

            for (String letter : row) {

```

```

        addButtonToKeyboard(letter, width, rowIndex, colIndex);
        colIndex++;
    }
    rowIndex++;
}
}

private boolean isChar(String key) {
    return key.length() == 1 && Character.isLetter(key.charAt(0));
}

private void addButtonToKeyboard(String key, int width, int row, int col) {
    Button button = new Button(this);
    if (isChar(key)) button.setText(key.toLowerCase());
    else {
        button.setText(key);
        button.setTypeface(customFont);
        button.setWidth(0);
        button.setPadding(0, 0, 0, 0);
    }

    GridLayout.LayoutParams params = new GridLayout.LayoutParams();
    params.width = width;
    params.height = GridLayout.LayoutParams.WRAP_CONTENT;
    params.rowSpec = GridLayout.spec(row);
    params.columnSpec = GridLayout.spec(col);
    params.setGravity(Gravity.FILL);

    button.setLayoutParams(params);
    button.setClickable(true);
    setHandler(button, key);

    keyboard.addView(button);
}

private void setHandler(Button button, String key) {
    if (key.length() == 1 && Character.isLetter(key.charAt(0))) {
        button.setOnClickListener(view -> {
            if (isCaps) textBar.append(key.toUpperCase());
            else textBar.append(key.toLowerCase());
        });
        return;
    }

    switch (key) {
        case "↑":
            button.setOnClickListener(view -> isCaps = !isCaps);
            break;
    }
}

```

```
        case "←":
            button.setOnClickListener(view -> {
                CharSequence currentText = textBar.getText();
                if(currentText.length()!=0) {
                    CharSequence newText = currentText.subSequence(0, currentText.length() - 1);
                    textBar.setText(newText);
                }
            });
            break;
        case "?123":
            button.setOnClickListener(view -> initKeyboard(numberKeys));
            break;
        case "ABC":
            button.setOnClickListener(view -> initKeyboard(letterKeys));
            break;
            case "↔":
                button.setOnClickListener(view -> textBar.append("\n"));
                break;
        default:
            button.setOnClickListener(view -> {
                CharSequence currentText = textBar.getText();
                CharSequence newText = currentText.subSequence(0, currentText.length() - 1);
                textBar.setText(newText);
                textBar.append(String.valueOf(key.charAt(0)));
                textBar.append("|");
            });
            break;
    }
}

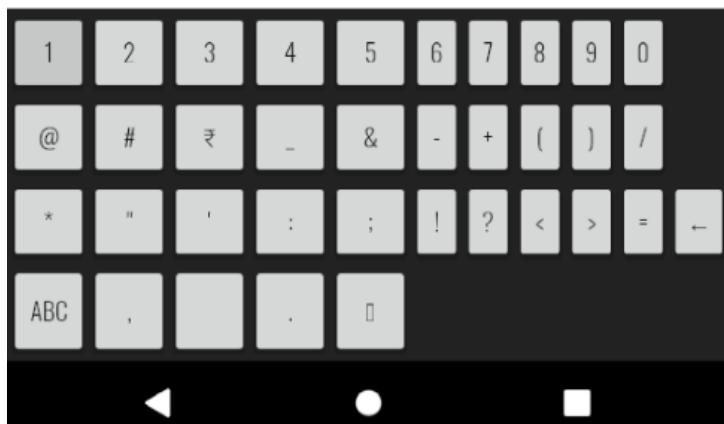
}
```

Output

Android Emulator - Pixel_2_API_30:5554



ERROR_404|



Learning outcomes

Thus a keyboard was implemented using Android Studio

Exercise 3 - Application Development using basic graphicalPrimitives

T Shivcharan
205001100
CSE-B

Aim

Design a CAR using Shape drawables with the help of relevant shapes such as Line, Circle, Rectangle and Arc.

- Move the car forward by pressing forward button so that car moves from a predefined

starting point to the predefined endpoint.

- b. On pressing backward button, rotate the car to 180 degrees from the current point to the starting point.
- c. Implement a Tap-to-zoom animation on any image.
- d. Implement the Card flipping animation.

Code

Drawables:

car.xml

```
<layer-list xmlns:android="http://schemas.android.com/apk/res/android">

    <item android:drawable="@drawable/car_body" />

    <item
        android:drawable="@drawable/car_roof"
        android:gravity="top"
    />

    <item
        android:drawable="@drawable/car_window"
        android:top="10dp"
        android:left="20dp"
        android:right="50dp"
        android:bottom="20dp"
    />

    <item
        android:drawable="@drawable/car_window"
        android:top="10dp"
        android:left="75dp"
        android:right="20dp"
        android:bottom="20dp"
    />

    <item
        android:drawable="@drawable/car_wheel"
        android:gravity="left|bottom"
        android:left="5dp"
        android:bottom="0dp"
    />

    <item
        android:drawable="@drawable/car_wheel"
        android:gravity="right|bottom"
        android:right="5dp"
        android:bottom="0dp"
    />

</layer-list>
```

car_body.xml

```
<shape xmlns:android="http://schemas.android.com/apk/res/android">
    <solid android:color="#123" />
```

```
<corners android:radius="8dp" />
<size android:width="100dp" android:height="40dp"/>
</shape>
```

car_roof.xml

```
<shape xmlns:android="http://schemas.android.com/apk/res/android">
    <solid android:color="#000000" />
    <corners android:radius="2dp" />
    <size android:width="20dp" android:height="15dp" />
</shape>
```

car_wheel.xml

```
<shape android:shape="oval"
    xmlns:android="http://schemas.android.com/apk/res/android">
    <solid android:color="#000000" />
    <size android:width="20dp" android:height="20dp" />
    <corners android:radius="3dp" />
</shape>
```

car_window.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
    <solid android:color="#333" />
    <corners android:radius="4dp" />
    <size android:width="20dp" android:height="20dp" />
</shape>
```

lane_marking.xml

```
<shape xmlns:android="http://schemas.android.com/apk/res/android">
    <solid android:color="#FFFFFF" />
    <size android:width="10dp" android:height="2dp" />
</shape>
```

road.xml

```
<layer-list xmlns:android="http://schemas.android.com/apk/res/android">
    <!-- Road Background (Gray) -->
    <item>
        <shape android:shape="rectangle">
            <solid android:color="#333" />
        </shape>
    </item>
    <item android:drawable="@drawable/lane_marking"
        android:top="10dp"
        android:right="80dp"
        android:left="10dp"
        android:bottom="10dp"/>
```

```

<item android:drawable="@drawable/lane_marking"
      android:top="10dp"
      android:right="45dp"
      android:left="45dp"
      android:bottom="10dp"/>
<item android:drawable="@drawable/lane_marking"
      android:top="10dp"
      android:right="10dp"
      android:left="80dp"
      android:bottom="10dp"/>
</layer-list>

```

Layout :

activity_main.xml

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    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"
    tools:context=".MainActivity">

    <ImageView
        android:id="@+id/carImageView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/car"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.302"
        tools:layout_editor_absoluteX="0dp" />

    <ImageView
        android:id="@+id/roadImageView"
        android:layout_width="410dp"
        android:layout_height="111dp"
        android:layout_marginTop="256dp"
        android:layout_marginEnd="1dp"
        android:layout_marginBottom="364dp"
        android:src="@drawable/road"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button

```

```

        android:id="@+id/forwardButton"
        android:layout_width="117dp"
        android:layout_height="45dp"
        android:layout_marginTop="60dp"
        android:text="Forward"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/roadImageView" />
<Button
        android:id="@+id/sunFlipId"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="108dp"
        android:text="Flip Image"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.496"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/backwardButton"
        app:layout_constraintVertical_bias="0.384" />
<Button
        android:id="@+id/backwardButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:text="Backward"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/forwardButton" />
<ImageView
        android:id="@+id/sunId"
        android:layout_width="79dp"

        android:layout_height="55dp"
        android:layout_marginTop="84dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="?attr/actionModeCloseDrawable" />
    <!-- Add other UI elements here if needed -->
</androidx.constraintlayout.widget.ConstraintLayout>

```

MainActivity.java

```

package com.example.abex3;
import androidx.appcompat.app.AppCompatActivity;
import android.animation.ObjectAnimator;
import android.os.Bundle;
import android.os.Handler;

```

```
import android.view.View;
import android.view.animation.AccelerateDecelerateInterpolator;
import android.widget.Button;
import android.widget.ImageView;
public class MainActivity extends AppCompatActivity {
    private ImageView carImageView;
    private ImageView sunImageView;
    private Button forwardButton;
    private Button backwardButton;
    private Button sunFlipButton;
    private int carXPosition = 0;
    private final int endpoint = 800;
    private final int startpoint = 0;
    boolean isClicked=false;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        carImageView = findViewById(R.id.carImageView);
        sunImageView = findViewById(R.id.sunId);
        forwardButton = findViewById(R.id.forwardButton);
        backwardButton = findViewById(R.id.backwardButton);
        sunFlipButton = findViewById(R.id.sunFlipId);
        forwardButton.setOnClickListener(new View.OnClickListener() {
            boolean isOperationInProgress = false;
            @Override
            public void onClick(View v) {
                if (!isOperationInProgress) {
                    isOperationInProgress = true;
                    backwardButton.setEnabled(false);
                    forwardButton.setEnabled(false);
                    final Handler handler = new Handler();
                    final Runnable carMovement = new Runnable() {
                        @Override
                        public void run() {
                            carXPosition += 10;
                            if (carXPosition <= endpoint) {
                                carImageView.setX(carXPosition);
                                handler.postDelayed(this, 100); // 100
milliseconds delay
                            } else {
                                isOperationInProgress = false;
                                backwardButton.setEnabled(true);
                                forwardButton.setEnabled(true);
                            }
                        }
                    };
                    handler.post(carMovement);
                }
            }
        });
    }
}
```

```

        }
    }
});  

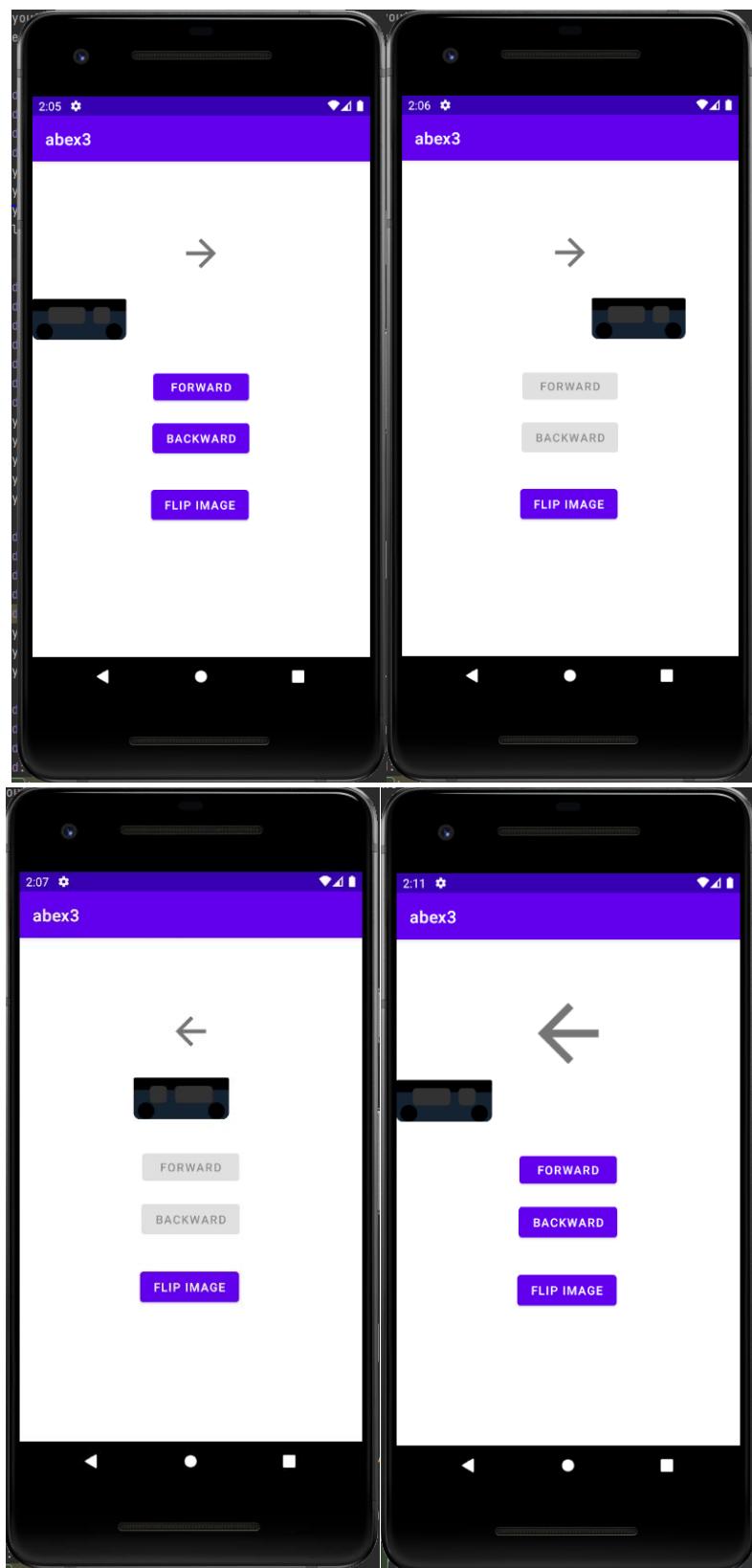
backwardButton.setOnClickListener(new View.OnClickListener() {
    boolean isOperationInProgress = false;
    private boolean isCarFlipped = false;
    @Override
    public void onClick(View v) {
        if (!isOperationInProgress) {
            isOperationInProgress = true;
            backwardButton.setEnabled(false);
            forwardButton.setEnabled(false);
            flipCar();
            final Handler handler = new Handler();
            final Runnable carMovement = new Runnable() {
                @Override
                public void run() {
                    carXPosition -= 10;
                    if (carXPosition >= startpoint) {
                        carImageView.setX(carXPosition);
                        handler.postDelayed(this, 100); // 100
milliseconds delay
                    } else {
                        isOperationInProgress = false;
                        backwardButton.setEnabled(true);
                        forwardButton.setEnabled(true);
                        flipCar();
                    }
                }
            };
            handler.post(carMovement);
        }
    }
};  

private void flipCar() {
    ObjectAnimator flipAnimator;
    if (isCarFlipped) {
        flipAnimator = ObjectAnimator.ofFloat(carImageView,
"rotationY", 180f, 0f);
        isCarFlipped = false;
    } else {
        flipAnimator = ObjectAnimator.ofFloat(carImageView,
"rotationY", 0f, 180f);
        isCarFlipped = true;
    }
    flipAnimator.setDuration(500);
    flipAnimator.setInterpolator(new
AccelerateDecelerateInterpolator());
    flipAnimator.start();
}

```

```
        }
    });
    sunImageView.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if(!isClicked) {
                sunImageView.setScaleX(2);
                sunImageView.setScaleY(2);
                isClicked=true;
            }
            else{
                sunImageView.setScaleX(1);
                sunImageView.setScaleY(1);
                isClicked=false;
            }
        }
    });
    sunFlipButton.setOnClickListener(new View.OnClickListener() {
        private boolean isArrowFlipped = false;
        @Override
        public void onClick(View v) {
            ObjectAnimator flipAnimator;
            if (isArrowFlipped) {
                flipAnimator = ObjectAnimator.ofFloat(sunImageView,
"rotationY", 180f, 0f);
                isArrowFlipped = false;
            } else {
                flipAnimator = ObjectAnimator.ofFloat(sunImageView,
"rotationY", 0f, 180f);
                isArrowFlipped = true;
            }
            flipAnimator.setDuration(500);
            flipAnimator.setInterpolator(new
AccelerateDecelerateInterpolator()));
            flipAnimator.start();
        }
    });
}
```

Output



Learning outcomes

- Proficient in creating user interfaces using shape drawables and relevant shapes.
- Implemented event listeners for user interaction and learned to navigate between activities.

- Successfully added animations, improving user engagement with tap-to-zoom and card flipping features.

Ex. No. 4 Android Application Development using Database

Aim: Develop a Product information application in Android that enables to perform CRUD operations on data stored in SQLite Database.

Layout Used: Linear Layout, Table Layout

Intents: Insert, Retrieve, Retrieve All, Update, Delete and Main

Code:

MainActivity.java:

```
package com.example.ex4;
import androidx.appcompat.app.AppCompatActivity;import
android.content.Intent;
import android.os.Bundle; import
android.view.View; import
android.widget.Button;
public class MainActivity extends AppCompatActivity {
```

 @Override

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

        final Database[] db = new Database[1]; Button create
            = findViewById(R.id.create);
```

```
create.setOnClickListener(new View.OnClickListener() {@Override
    public void onClick(View v) {
        db[0] = new Database(MainActivity.this);
    }
});

Button insert = findViewById(R.id.insert);
insert.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(MainActivity.this, Insert.class);startActivity(intent);
    }
});

Button delete = findViewById(R.id.delete); delete.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(MainActivity.this,Delete.class);startActivity(intent);
    }
});

Button retrieve = findViewById(R.id.read); retrieve.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(MainActivity.this,Retrieve.class);startActivity(intent);
    }
});
```

```

        Button retrieveAll = findViewById(R.id.read_all); retrieveAll.setOnClickListener(new
        View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(MainActivity.this,RetrieveAll.class);startActivity(intent);
            }
        });

        Button update = findViewById(R.id.update); update.setOnClickListener(new
        View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(MainActivity.this,Update.class);startActivity(intent);
            }
        });
    }

}

```

Database.java:

```

package com.example.ex4;

import android.content.ContentValues;
import android.content.Context; import
android.database.Cursor;
import android.database.sqlite.SQLiteDatabase; import
android.database.sqlite.SQLiteOpenHelper;

public class Database extends SQLiteOpenHelper {private static

final String DB_NAME = "product";

```

```
private static final int DB_VERSION = 3;

private static final String TABLE_NAME = "prod";private static

final String ID_COL = "id";

private static final String NAME_COL = "name"; private static

final String BRAND_COL = "brand";private static final String

DESC_COL = "desc"; private static final String PRICE_COL ="

"price";

public Database(Context context) {
    super(context, DB_NAME, null, DB_VERSION);
}

@Override
public void onCreate(SQLiteDatabase db) {

    String query = "CREATE TABLE " + TABLE_NAME + " (" +
        + ID_COL + " INTEGER PRIMARY KEY, "
        + NAME_COL + " TEXT, "
        + BRAND_COL + " TEXT, "
        + DESC_COL + " TEXT, "
        + PRICE_COL + " TEXT)";

    // at last we are calling a exec sql
    // method to execute above sql querydb.execSQL(query);
}

// this method is use to add new course to our sqlite database.
```

```
public void addProduct(String id, String name, String brand, String desc, String price) {  
  
    // on below line we are creating a variable for  
    // our sqlite database and calling writable method  
    // as we are writing data in our database. SQLiteDatabase db =  
    this.getWritableDatabase();  
  
    // on below line we are creating a  
    // variable for content values.  
    ContentValues values = new ContentValues();  
  
    // on below line we are passing all values  
    // along with its key and value pair.  
    values.put(ID_COL, id); values.put(NAME_COL,  
    name); values.put(BRAND_COL, brand);  
    values.put(DESC_COL, desc);  
    values.put(PRICE_COL, price);  
  
    // after adding all values we are passing  
    // content values to our table. db.insert(TABLE_NAME, null,  
    values);  
  
    // at last we are closing our  
    // database after adding database.db.close();  
}  
  
public void deleteProduct(String id){ SQLiteDatabase db =  
    this.getWritableDatabase();  
    db.delete(TABLE_NAME, ID_COL+"=?", new String[]{id});  
}  
  
public void updateProduct(String id, String price){
```

```
SQLiteDatabase db = this.getWritableDatabase(); ContentValues  
values = new ContentValues(); values.put(PRICE_COL, price);  
db.update(TABLE_NAME, values, ID_COL + "=?", new  
String[]{String.valueOf(id)});  
}  
  
public Cursor retrieveAll(){  
    SQLiteDatabase db = this.getReadableDatabase();  
    return db.query(TABLE_NAME, null, null, null, null, null, null);  
}  
  
public Cursor retrieve(String id){  
    SQLiteDatabase db = this.getReadableDatabase();String[] projection =  
{  
    NAME_COL,  
    BRAND_COL,  
    DESC_COL,  
    PRICE_COL  
};  
  
// Define the condition for retrieval (e.g., where id = ?)String selection =  
ID_COL + " = ?";  
String[] selectionArgs = { id };  
  
// Execute the query Cursor  
cursor = db.query(  
    TABLE_NAME,      // Table name projection,  
    null,           // Columns to return selection,  
    selection,       // Selection (WHERE clause)  
    selectionArgs,   // Selection arguments null,  
    null,           // Group bynull,    // Having  
    null            // Order by  
);
```

```
    // The cursor now contains the retrieved row(s) return cursor;
}

@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    // this method is called to check if the table exists already. db.execSQL("DROP TABLE IF
    EXISTS " + TABLE_NAME);
    onCreate(db);
}
}
```

Insert.java:

```
package com.example.ex4;

import android.content.Intent; import
android.os.Bundle; import
android.util.Log; import
android.view.View; import
android.widget.Button; import
android.widget.RadioButton;
import android.widget.RadioGroup; import
android.widget.TextView; import

androidx.appcompat.app.AppCompatActivity; public

class Insert extends AppCompatActivity {

    @Override
```

```
protected void onCreate(Bundle savedInstanceState){  
    super.onCreate(savedInstanceState); setContentView(R.layout.insert);
```

```
Button backi = findViewById(R.id.backi); backi.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(Insert.this,MainActivity.class);startActivity(intent);
    }
});

Button submiti = findViewById(R.id.submiti); submiti.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Database db = new Database(Insert.this);

        TextView textView = findViewById(R.id.idi);String id =
        textView.getText().toString(); textView =
        findViewById(R.id.namei); String name =
        textView.getText().toString();

        RadioGroup radioGroup = findViewById(R.id.radioGroup);int
        selectedRadioButtonId =
        radioGroup.getCheckedRadioButtonId(); RadioButton
        selectedRadioButton =
        findViewById(selectedRadioButtonId);
        String brand = selectedRadioButton.getText().toString();Log.d("Debug",brand);

        textView = findViewById(R.id.desci);String desc
        = textView.getText().toString();
        Log.d("Debug",desc);

        textView = findViewById(R.id.pricei);String price
        =
```

```
textView.getText().toString();
```

```
        db.addProduct(id,name,brand,desc,price);
    }
});
}
}
```

Retrieve:

```
package com.example.ex4;

import android.content.Intent; import
android.database.Cursor;import
android.os.Bundle; import
android.view.View; import
android.widget.Button; import
android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;

public class Retrieve extends AppCompatActivity {

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

        Button backr = findViewById(R.id.backr);
        backr.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(Retrieve.this,MainActivity.class);startActivity(intent);
            }
        });
    }
}
```

```
Button retrieve = findViewById(R.id.retrieve); retrieve.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View v) {
        TextView textView = findViewById(R.id.idr);String id =
        textView.getText().toString();

        Database db = new Database(Retrieve.this);Cursor cursor =
        db.retrieve(id);
        String name","",brand","",desc","",price"";

        if (cursor.moveToFirst()) {

            do {
                // Retrieve values from 'column1' and 'column2' as stringsint columnIndex =
                cursor.getColumnIndex("name");
                if (columnIndex != -1) {
                    name = cursor.getString(columnIndex);
                }

                columnIndex =
                cursor.getColumnIndex("brand");
                if(columnIndex!=-1){
                    brand=cursor.getString(columnIndex);
                }

                columnIndex =
                cursor.getColumnIndex("desc");
                if(columnIndex!=-1){
                    desc=cursor.getString(columnIndex);
                }

                columnIndex =
                cursor.getColumnIndex("price");
                if(columnIndex!=-1){
                    price=cursor.getString(columnIndex);
                }
            }
        }
    }
});
```



```
        } while (cursor.moveToNext());
    }

    textView = findViewById(R.id.namer);
    textView.setText(name);

    textView = findViewById(R.id.brandr);
    textView.setText(brand);

    textView = findViewById(R.id.descr);
    textView.setText(desc);

    textView = findViewById(R.id.pricer);
    textView.setText(price);

    });
}
}
```

Retrieve All:

```
package com.example.ex4;

import android.content.Context; import
android.content.Intent; import
android.database.Cursor;import
android.graphics.Color; import
android.os.Bundle; import
android.util.Log;
import android.view.View; import
android.view.ViewGroup;import
android.widget.Button;
import android.widget.TableLayout;import
android.widget.TableRow;
```

```
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity; import
androidx.constraintlayout.widget.ConstraintLayout;import
androidx.constraintlayout.widget.ConstraintSet;

import org.w3c.dom.Text;
public class RetrieveAll extends AppCompatActivity {private TextView
newTextView(String text){
    TextView tv = new TextView(RetrieveAll.this);

    tv.setText(text);
    tv.setPadding(4,4,4,4);
    tv.setTextSize(24);
    TableRow.LayoutParams layoutParams = new
TableRow.LayoutParams(
        TableRow.LayoutParams.WRAP_CONTENT, // Adjust width as
needed                                         TableRow.LayoutParams.WRAP_CONTENT
                                         // Adjust height as
needed
    );
}

    layoutParams.rightMargin = 20;
    tv.setLayoutParams(layoutParams);

    return tv;
}

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

    Database db = new Database(RetrieveAll.this);
```

```
Cursor cursor = db.retrieveAll();

ConstraintLayout parent = findViewById(R.id.parent);

// Create layout params for the TableLayout
//     ConstraintLayout.LayoutParams tableLayoutParams = new
ConstraintLayout.LayoutParams(
//         ConstraintLayout.LayoutParams.MATCH_PARENT,
//         ConstraintLayout.LayoutParams.MATCH_PARENT
//     );
//     tableLayoutParams.startToStart =
ConstraintLayout.LayoutParams.PARENT_ID
;
//     tableLayoutParams.endToEnd =
ConstraintLayout.LayoutParams.PARENT_ID
;
//     tableLayoutParams.topToTop =
ConstraintLayout.LayoutParams.PARENT_ID
;
//     tableLayoutParams.bottomToBottom =
ConstraintLayout.LayoutParams.PARENT_ID
;

TableLayout tl = new TableLayout(RetrieveAll.this);

TableRow trh = new TableRow(RetrieveAll.this);TextView tv0
= new TextView("Id"); trh.addView(tv0);

tv0 = new TextView("Name");
trh.addView(tv0);

tv0 = new TextView("Brand");trh.addView(tv0);

tv0 = new TextView("Description");
```

```
trh.addView(tv0);  
tv0 = new TextView("Price");
```

```
trh.addView(tv0);

tl.addView(trh);

String id="",name:"",brand="",desc="",price="";
if(cursor.moveToFirst()) {
    do {
        // Retrieve values from 'column1' and 'column2' as strings
        int columnIndex = cursor.getColumnIndex("id");
        if (columnIndex != -1) {
            id = cursor.getString(columnIndex);
        }

        columnIndex =
        cursor.getColumnIndex("name");
        if(columnIndex!=-1){
            name=cursor.getString(columnIndex);
        }

        columnIndex =
        cursor.getColumnIndex("brand");
        if(columnIndex!=-1){
            brand=cursor.getString(columnIndex);
        }

        columnIndex =
        cursor.getColumnIndex("desc");
        if(columnIndex!=-1){
            desc=cursor.getString(columnIndex);
        }

        columnIndex =
        cursor.getColumnIndex("price");
        if(columnIndex!=-1){
            price=cursor.getString(columnIndex);
        }
    }
}
```

```
Log.d("Debug",id+" "+name+" "+brand+" "+desc+" "+price);
```

```
    TableRow tr = new TableRow(RetrieveAll.this);

    TextView tv1 = new TextView(id); TextView tv2
    = new TextView(name);TextView tv3 =
    new TextView(brand);TextView tv4 =
    new TextView(desc);TextView tv5 =
    new TextView(price);

    tr.addView(tv1)
    ;
    tr.addView(tv2)
    ;
    tr.addView(tv3)
    ;
    tr.addView(tv4)
    ;
    tr.addView(tv5)
    ;

    tl.addView(tr);

} while (cursor.moveToNext());
}

parent.addView(tl);

// app:layout_constraintEnd_toEndOf="parent"
// app:layout_constraintStart_toStartOf="parent"
// app:layout_constraintTop_toTopOf="parent"

Button retrieveAll = findViewById(R.id.backra); retrieveAll.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(RetrieveAll.this,MainActivity.class);
    }
})
```

```
        startActivity(intent);  
    }  
});
```

```
    }  
}
```

Update:

```
package com.example.ex4;  
  
import android.content.Intent;import  
android.os.Bundle; import  
android.view.View; import  
android.widget.Button;import  
android.widget.TextView;  
import androidx.appcompat.app.AppCompatActivity;  
  
public class Update extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState){  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.update);  
  
        Button backu = findViewById(R.id.backu);  
        backu.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View v) {  
                Intent intent = new Intent(Update.this,MainActivity.class);startActivity(intent);  
            }  
        });  
  
        Button update = findViewById(R.id.update); update.setOnClickListener(new  
        View.OnClickListener() {
```

@Override

```
public void onClick(View v) {
    Database db = new Database(Update.this);TextView
    textView = findViewById(R.id.idu); String id =
    textView.getText().toString(); textView =
    findViewById(R.id.priceu); String price =
    textView.getText().toString();

    db.updateProduct(id,price);
}
});
}
}
```

Delete.java:

```
package com.example.ex4;

import android.content.Intent;import
android.os.Bundle; import
android.view.View; import
android.widget.Button;import
android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;

public class Delete extends AppCompatActivity {

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

        Button backd = findViewById(R.id.backd);
        backd.setOnClickListener(new View.OnClickListener() {
```

```
@Override  
public void onClick(View v) {
```

```
        Intent intent = new  
        Intent(Delete.this,MainActivity.class);  
        startActivity(intent);  
    }  
});
```

```
Button delete = findViewById(R.id.delete); delete.setOnClickListener(new  
View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        Database db = new Database(Delete.this);  
  
        TextView textView = findViewById(R.id.idd);String id =  
        textView.getText().toString();  
  
        db.deleteProduct(id);  
    }  
});  
}  
}
```

Activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout  
    xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:tools="http://schemas.android.com/tools"  
    xmlns:app="http://schemas.android.com/apk/res-auto"  
    android:layout_width="match_parent" android:layout_height="match_parent"  
    tools:context=".MainActivity">  
  
<RelativeLayout android:layout_width="match_parent"  
    android:layout_height="match_parent">
```

```
<!-- Center the LinearLayout vertically -->
<LinearLayout android:id="@+id/verticalLayout"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerInParent="true"
    android:orientation="vertical">

    <!-- Add your UI elements within this LinearLayout -->

    <Button
        android:id="@+id/create"
        android:layout_width="177dp"
        android:layout_height="62dp" android:text="Create" />

    <Button
        android:id="@+id/insert"
        android:layout_width="177dp"
        android:layout_height="62dp" android:text="Insert" />

    <Button
        android:id="@+id/read"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Retrieve" />

    <Button
        android:id="@+id/read_all"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Retrieve All" />

    <Button
```

```
        android:id="@+id/update"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Update" />

    <Button
        android:id="@+id/delete"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Delete" />

    <!-- Add more views here as needed -->
    </LinearLayout>
</RelativeLayout>

</androidx.constraintlayout.widget.ConstraintLayout>
```

Insert.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    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">

    <LinearLayout android:layout_width="409dp"
        android:layout_height="665dp"
        android:layout_marginTop="50dp"
        android:orientation="vertical"
        app:layout_constraintBottom_toBottomOf="parent"
```

```
app:layout_constraintStart_toStartOf="parent" app:layout_constraintTop_toTopOf="parent">>

<LinearLayout android:layout_width="match_parent"
    android:layout_height="87dp"
    android:orientation="horizontal">

    <TextView android:id="@+id/textView3"
        android:layout_width="120dp"
        android:layout_height="55dp"
        android:layout_weight="1"
        android:text="Product Id"
        android:textSize="24sp" />

    <EditText
        android:id="@+id/idi"
        android:layout_width="wrap_content"
        android:layout_height="58dp"
        android:layout_weight="1" android:ems="10"
        android:inputType="textPersonName"
        android:textSize="24sp" />
</LinearLayout>

<LinearLayout android:layout_width="match_parent"
    android:layout_height="84dp"
    android:orientation="horizontal">

    <TextView android:id="@+id/textView5"
        android:layout_width="131dp"
        android:layout_height="58dp"
```

```
    android:layout_weight="1"
    android:text="Product Name"
    android:textSize="24sp" />

<EditText android:id="@+id/namei"
    android:layout_width="wrap_content"
    android:layout_height="59dp"
    android:layout_weight="1" android:ems="10"
    android:inputType="textPersonName"
    android:textSize="24sp" />
</LinearLayout>

<LinearLayout android:layout_width="match_parent"
    android:layout_height="94dp"
    android:orientation="horizontal">

    <TextView android:id="@+id/textView6"
        android:layout_width="198dp"
        android:layout_height="71dp"
        android:layout_weight="1"
        android:text="Brand"
        android:textSize="24sp" />

    <RadioGroup android:id="@+id/radioGroup"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="1" >

        <RadioButton android:id="@+id/radio1"

```

```
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="24sp" android:text="A" />

    <RadioButton android:id="@+id/radio2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="24sp" android:text="B" />
</RadioGroup>

</LinearLayout>

<LinearLayout android:layout_width="match_parent"
    android:layout_height="93dp"
    android:orientation="horizontal">

    <TextView android:id="@+id/textView7"
        android:layout_width="146dp"
        android:layout_height="73dp"
        android:layout_weight="1"
        android:text="Description"
        android:textSize="24sp" />

    <EditText android:id="@+id/desci"
        android:layout_width="wrap_content"
        android:layout_height="75dp"
        android:layout_weight="1" android:ems="10"
        android:inputType="textPersonName"
```

```
        android:textSize="24sp" />
    </LinearLayout>

<LinearLayout android:layout_width="match_parent"
    android:layout_height="96dp"
    android:orientation="horizontal">

    <TextView android:id="@+id/textView8"
        android:layout_width="114dp"
        android:layout_height="79dp"
        android:layout_weight="1"
        android:text="Price"
        android:textSize="24sp" />

    <EditText android:id="@+id/pricei"
        android:layout_width="wrap_content"
        android:layout_height="82dp"
        android:layout_weight="1" android:ems="10"
        android:inputType="textPersonName"
        android:textSize="24sp" />
</LinearLayout>

<Button
    android:id="@+id/submiti"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Submit" />

<Button
    android:id="@+id/backi"
    android:layout_width="match_parent"
```

```
        android:layout_height="wrap_content" android:text="Back"
    />

</LinearLayout>
</androidx.constraintlayout.widget.ConstraintLayout>
```

Retrieve.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    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">

    <LinearLayout android:id="@+id/linearLayout"
        android:layout_width="414dp"
        android:layout_height="203dp"
        android:orientation="vertical"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="1.0"
        app:layout_constraintStart_toStartOf="parent"
        tools:ignore="MissingConstraints"
        tools:layout_editor_absoluteY="-2dp">

        <LinearLayout android:layout_width="match_parent"
            android:layout_height="103dp"
            android:orientation="horizontal">

            <TextView android:id="@+id/textView"
                android:layout_width="144dp"
```

```
        android:layout_height="64dp"
        android:layout_weight="1" android:text="Product Id"
        android:textSize="24sp" />

<EditText
    android:id="@+id/idr"
    android:layout_width="wrap_content"
    android:layout_height="83dp"
    android:layout_weight="1" android:ems="10"
    android:inputType="textPersonName"
    android:textSize="24sp" />
</LinearLayout>

<Button
    android:id="@+id/retrieve"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Retrieve" />

<Button
    android:id="@+id/backr"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" android:text="Back"
    />
</LinearLayout>

<LinearLayout android:layout_marginTop="50dp"
    android:layout_width="411dp"
    android:layout_height="459dp"
    android:orientation="vertical"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="parent"
```

```
app:layout_constraintTop_toBottomOf="@+id/linearLayout">

<LinearLayout android:layout_width="match_parent"
    android:layout_height="84dp"
    android:orientation="horizontal">

    <TextView android:id="@+id/textView5"
        android:layout_width="131dp"
        android:layout_height="58dp"
        android:layout_weight="1"
        android:text="Product Name"
        android:textSize="24sp" />

    <TextView android:id="@+id/namer"
        android:layout_width="146dp"
        android:layout_height="73dp"
        android:layout_weight="1"
        android:text=""
        android:textSize="24sp" />
</LinearLayout>

<LinearLayout android:layout_width="match_parent"
    android:layout_height="94dp"
    android:orientation="horizontal">

    <TextView android:id="@+id/textView6"
        android:layout_width="198dp"
        android:layout_height="71dp"
        android:layout_weight="1"
        android:text="Brand"
```

```
        android:textSize="24sp" />

    <TextView android:id="@+id/brandr"
        android:layout_width="146dp"
        android:layout_height="73dp"
        android:layout_weight="1"
        android:text=""
        android:textSize="24sp" />

</LinearLayout>

<LinearLayout android:layout_width="match_parent"
    android:layout_height="93dp"
    android:orientation="horizontal">

    <TextView android:id="@+id/textView0"
        android:layout_width="146dp"
        android:layout_height="73dp"
        android:layout_weight="1"
        android:text="Description"
        android:textSize="24sp" />

    <TextView android:id="@+id/descr"
        android:layout_width="146dp"
        android:layout_height="73dp"
        android:layout_weight="1"
        android:text=""
        android:textSize="24sp" />
</LinearLayout>

<LinearLayout>
```

```

        android:layout_width="match_parent"
        android:layout_height="96dp"
        android:orientation="horizontal">

        <TextView android:id="@+id/textView8"
            android:layout_width="114dp"
            android:layout_height="79dp"
            android:layout_weight="1"
            android:text="Price"
            android:textSize="24sp" />

        <TextView android:id="@+id/pricer"
            android:layout_width="146dp"
            android:layout_height="73dp"
            android:layout_weight="1"
            android:text=""
            android:textSize="24sp" />
    </LinearLayout>
</androidx.constraintlayout.widget.ConstraintLayout>
```

Retrieve All:

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    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:id="@+id/parent">
```

```
<Button  
    android:id="@+id/backra" android:layout_width="match_parent"  
    android:layout_height="wrap_content" android:text="Back"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintStart_toStartOf="parent" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>
```

Update.xml:

```
<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout  
    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">  
  
<LinearLayout android:layout_width="412dp"  
    android:layout_height="352dp"  
    android:orientation="vertical"  
    app:layout_constraintStart_toStartOf="parent" tools:layout_editor_absoluteY="186dp"  
    tools:ignore="MissingConstraints">  
  
<LinearLayout android:layout_width="match_parent"  
    android:layout_height="107dp"
```

```
        android:orientation="horizontal">>

    <TextView android:id="@+id/textView2"
        android:layout_width="105dp"
        android:layout_height="80dp"
        android:layout_weight="1"
        android:textSize="24sp"
        android:text="Product Id" />

    <EditText android:id="@+id/edu"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_weight="1" android:ems="10"
        android:inputType="textPersonName"
        android:textSize="24sp" />
</LinearLayout>

<LinearLayout android:layout_width="match_parent"
    android:layout_height="97dp"
    android:orientation="horizontal">

    <TextView android:id="@+id/textView4"
        android:layout_width="108dp"
        android:layout_height="79dp"
        android:layout_weight="1"
        android:textSize="24sp"
        android:text="Price" />

    <EditText android:id="@+id/priceu"
```

```
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_weight="1" android:ems="10"
        android:textSize="24sp"
        android:inputType="textPersonName" />
    </LinearLayout>

    <Button
        android:id="@+id/update"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Update" />

    <Button
        android:id="@+id/backu"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" android:text="Back"
        />
</LinearLayout>
</androidx.constraintlayout.widget.ConstraintLayout>
```

Delete.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    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">

    <LinearLayout android:layout_width="414dp"
        android:layout_height="475dp"
```

```
        android:orientation="vertical" app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" tools:layout_editor_absoluteY="101dp"
        tools:ignore="MissingConstraints">

    <LinearLayout android:layout_width="match_parent"
        android:layout_height="152dp"
        android:orientation="horizontal">

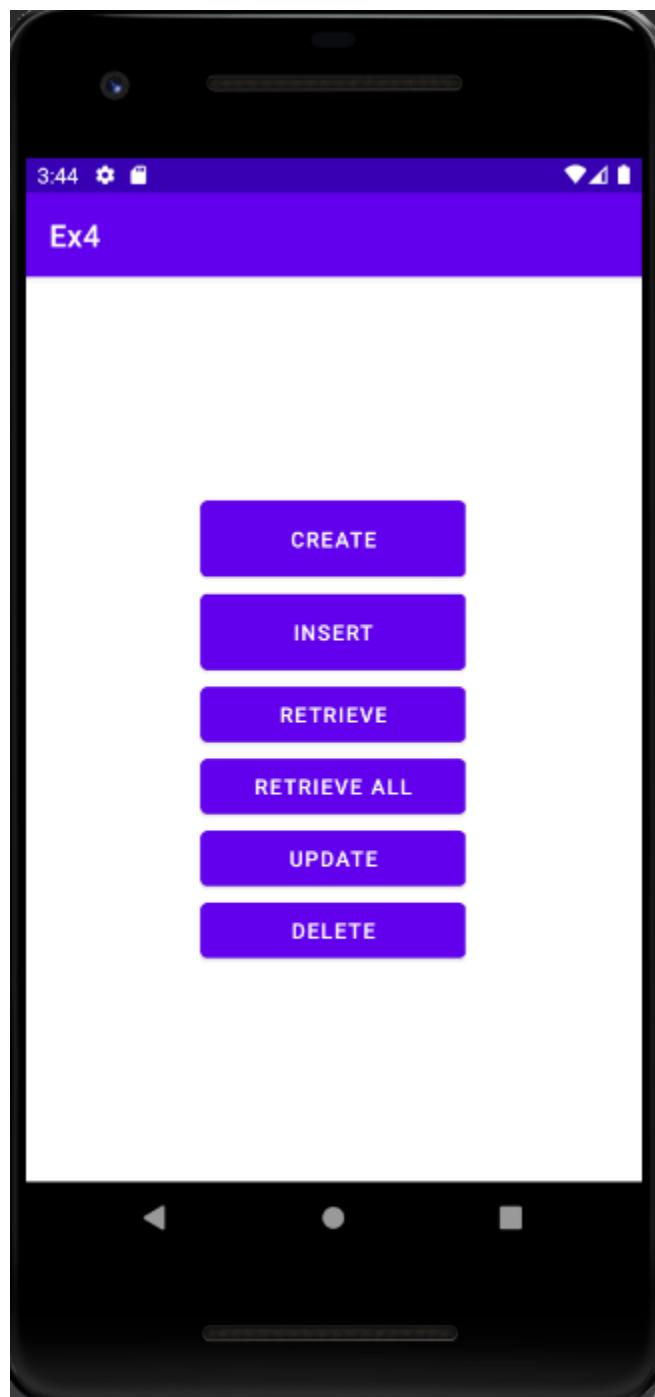
        <TextView android:id="@+id/textView"
            android:layout_width="144dp"
            android:layout_height="89dp"
            android:layout_weight="1"
            android:textSize="24sp"
            android:text="Product Id" />

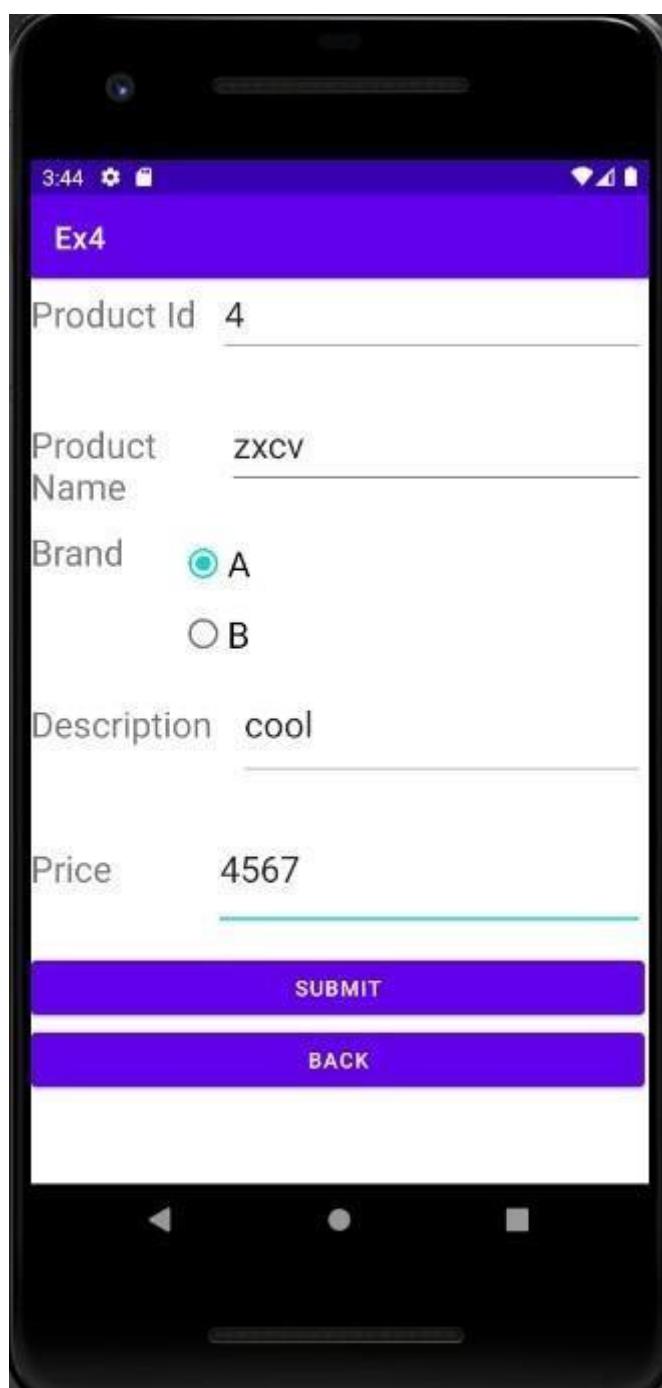
        <EditText android:id="@+id/idd"
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:layout_weight="1" android:ems="10"
            android:inputType="textPersonName"
            android:textSize="24sp" />
    </LinearLayout>

    <Button
        android:id="@+id/delete"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Delete" />
```

```
<Button  
    android:id="@+id/backd"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content" android:text="Back"  
    />  
</LinearLayout>  
</androidx.constraintlayout.widget.ConstraintLayout>
```

Output:

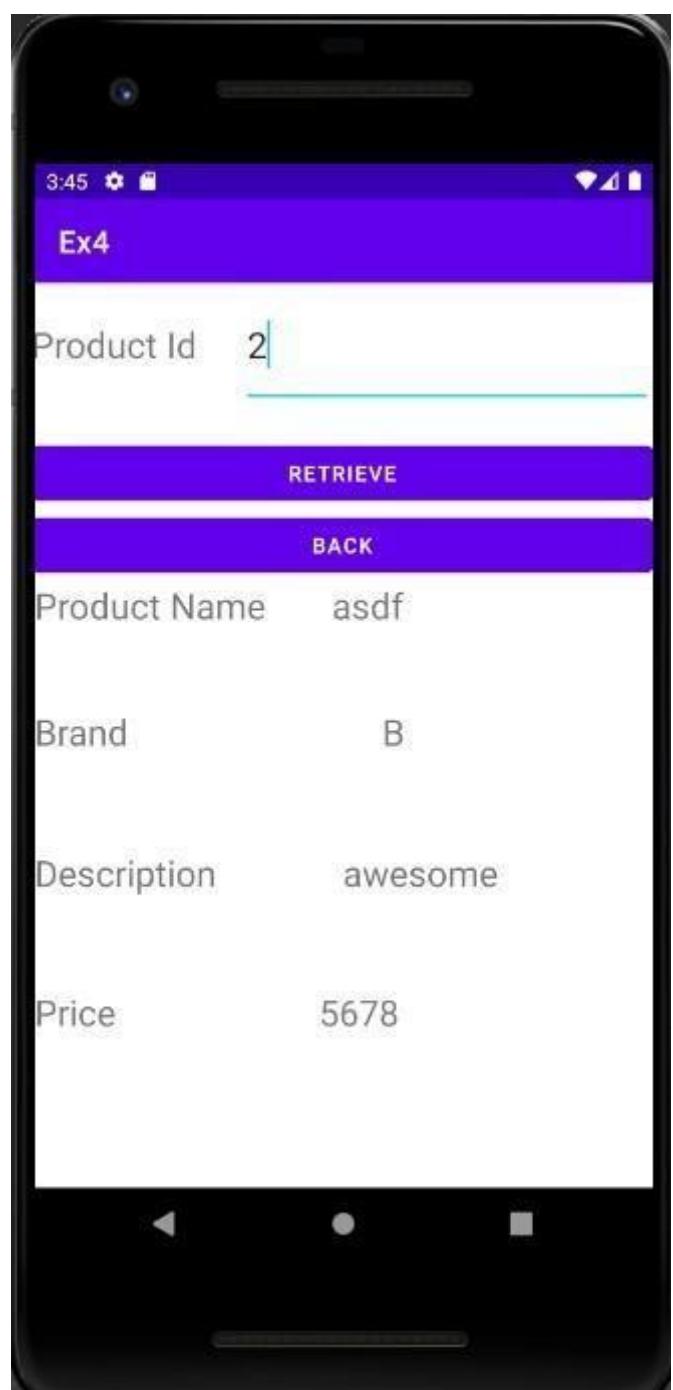


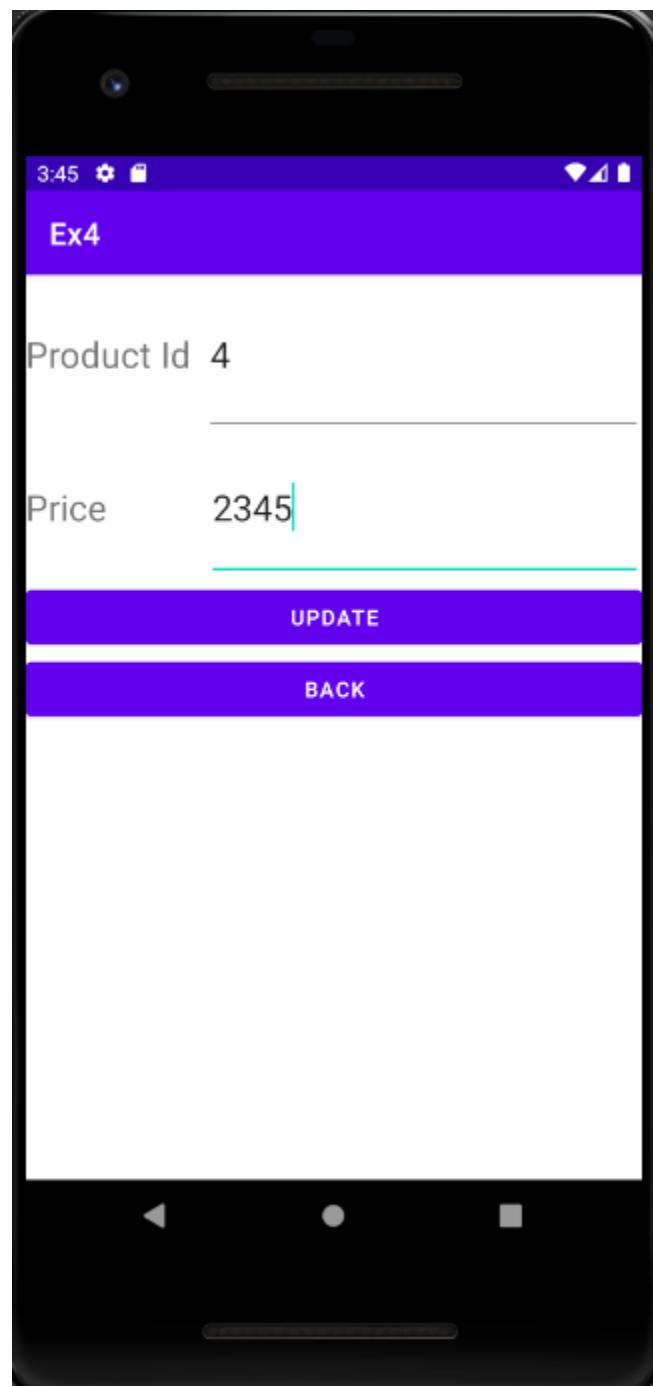


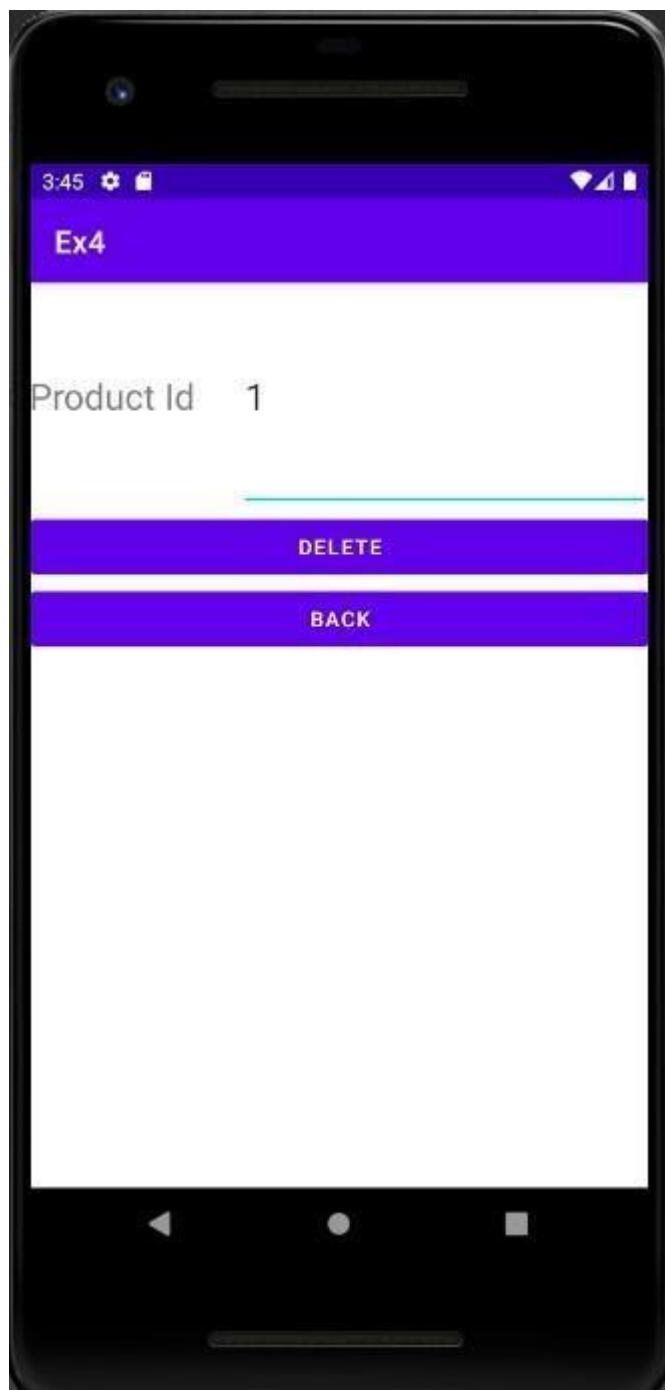
The screenshot shows a mobile application interface. At the top, there is a purple header bar with the title "Ex4". Below the header is a white table with black borders. The table has five columns: "Id", "Name", "Brand", "Description", and "Price". There are four rows of data in the table. The first row contains the values "1", "qwer", "A", "nice", and "1234". The second row contains "2", "asdf", "B", "awesome", and "5678". The third row contains "4", "zxcv", "A", "cool", and "4567". At the bottom of the screen, there is a purple footer bar with a white "BACK" button in the center. The phone's navigation bar is visible at the very bottom.

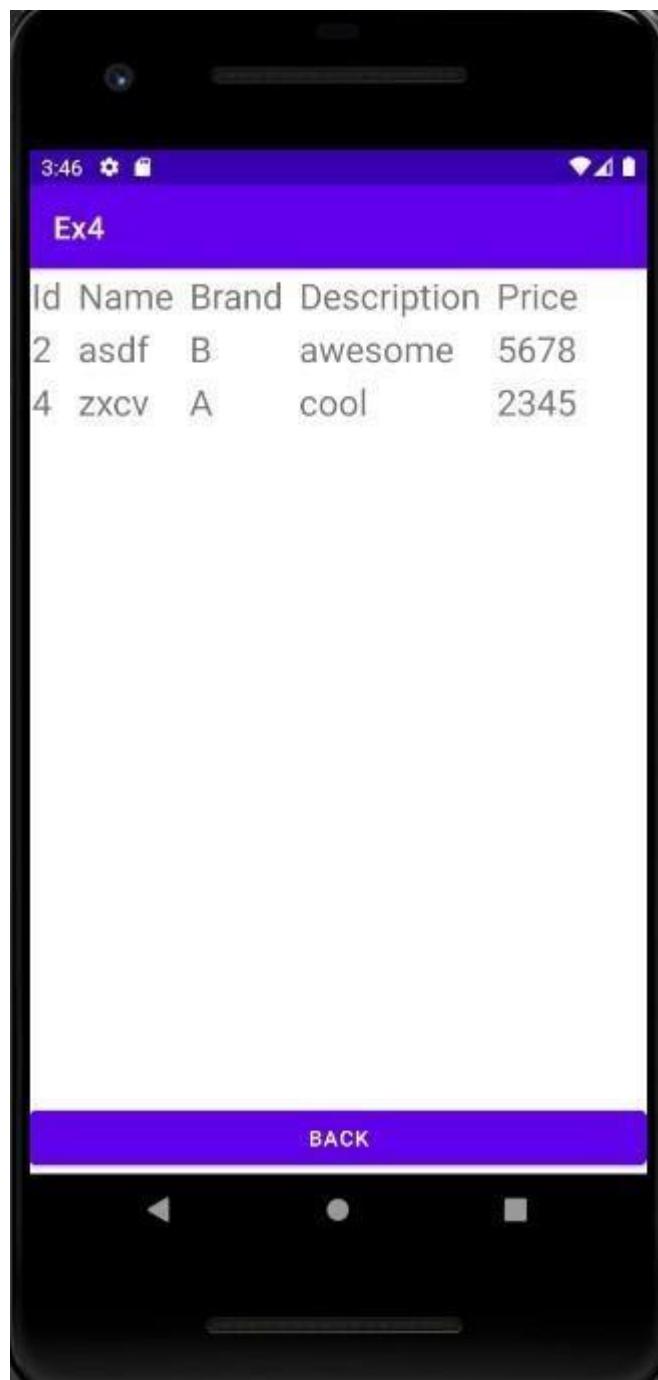
Id	Name	Brand	Description	Price
1	qwer	A	nice	1234
2	asdf	B	awesome	5678
4	zxcv	A	cool	4567

BACK









Best Practices:

- Used apt names for xml and java files.
- Set padding and margins for dynamically added elements

Learning Outcomes:

- Learnt to integrate sqlite3 in android studio
- Learnt to do CRUD operations

Ex. No. 5 Android Application using Multithreading

Aim:

Develop an android application to perform multithreading. Define 3 threads to run concurrently when “start” button is clicked.

The first thread should change the color of the text indefinitely
The second thread should implement a moving banner

The third thread should display a counter starting from 0 to 1000 When the “Stop” button

is pressed all the threads should be stopped

Layouts Used: None. Three textViews.

Code:

MainActivity.java:

```
package com.example.ex5;

import androidx.appcompat.app.AppCompatActivity;
import android.graphics.Color;import
android.os.Bundle;
```

```
import android.util.Log; import  
android.view.View;import  
android.widget.Button;  
import android.widget.TextView;  
  
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        TextView t1 = findViewById(R.id.t1);Thread1  
        th1 = new Thread1(t1);  
  
        TextView t2 = findViewById(R.id.t2);Thread2  
        th2 = new Thread2(t2);  
  
        TextView t3 = findViewById(R.id.t3);Thread3  
        th3 = new Thread3(t3);
```

```
final boolean[] init = {false};

Button start = findViewById(R.id.start); start.setOnClickListener(new
View.OnClickListener() {

    @Override

    public void onClick(View v) {if(!init[0]){

        th1.start();

        th2.start();

        th3.start();

        init[0] = true;

    }

    else{

        Log.d("debug","hello");

        th1.pause(false); th2.pause(false);

        th3.pause(false);

    }

}

});
```

```
Button stop = findViewById(R.id.stop); stop.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View v) {
        th1.pause(true); th2.pause(true);
        th3.pause(true);
    }
});
```

}

}

Thread1.java:

```
package com.example.ex5;

import android.graphics.Color;import
android.util.Log;
import android.widget.TextView;
```

```
public class Thread1 extends Thread{TextView t;  
    int red = 120; int  
    green = 120;int blue  
    = 120;  
    boolean paused = false; Object  
    lock = new Object();  
    Thread1(TextView t){  
        this.t=t;  
    }  
  
    public void pause(boolean paused){synchronized  
        (lock){  
            if(paused) this.paused  
            = true;else{  
                this.paused = false;  
                lock.notifyAll();  
            }  
        }  
    }
```

```
Log.d("Debug",""+paused);

}

public void run(){

while(true) {try

{

int color = Color.rgb(red, green, blue);t.setTextColor(color);

red = (red + 20) % 255; green =

(green + 10) % 255;blue = (blue + 5)

% 255;

Thread.sleep(500);

synchronized (lock){

while(paused){

try{

lock.wait();

}catch(InterruptedException e){
```

```
        }

    }

}

} catch (InterruptedException e) {

    e.printStackTrace();

}

}

}
```

Thread2.java:

```
package com.example.ex5;

import android.util.Log;

import android.view.animation.TranslateAnimation;import
android.widget.TextView;

public class Thread2 extends Thread{TextView t;
```

```
int dir = 1;

int translationDistance = 300;boolean
paused=false; Object lock = new
Object(); Thread2(TextView t){

    this.t=t;

}

public void pause(boolean paused){synchronized
(lock){

    if(paused)

        this.paused =


    true;

    else{

        this.paused = false;

        lock.notifyAll();

    }

}

Log.d("Debug",""+paused);

}

public void run(){
```

```
while (!paused) {try {

    TranslateAnimation animation;if (dir ==

    1) {

        animation = new TranslateAnimation(-translationDistance,translationDistance, 0,
0);

    } else {

        animation = new TranslateAnimation(translationDistance,
-translationDistance, 0, 0);

    }

    animation.setDuration(3000); // Keep the total duration the same

    animation.setFillAfter(true);

    t.startAnimation(animation);

}

Thread.sleep(3000)

; dir = 1 - dir;

synchronized (lock){

    while(paused){

        try{

            lock.wait();

        }

    }

}
```

```
        }catch(InterruptedException e){  
            }  
        }  
    } catch (InterruptedException e) {  
        e.printStackTrace();  
    }  
}
```

Thread3.java:

```
package com.example.ex5;  
  
import android.util.Log;  
import android.widget.TextView;
```

```
public class Thread3 extends Thread{TextView t;  
    int ctr=0;  
    boolean paused = false; Object  
    lock = new Object();  
  
    Thread3(TextView t){  
        this.t=t;  
    }  
  
    public void pause(boolean paused){synchronized  
        (lock){  
            if(paused)  
                this.paused =  
                    true;  
            else{  
                this.paused = false;  
                lock.notifyAll();  
            }  
        }  
        Log.d("Debug",""+paused);  
    }  
}
```

```
public void run(){
```

```
while (ctr < 3000 && !paused) {try {
    Thread.sleep(1000)
    ; ctr += 1;

    // Update the TextView on the UI threadt.post(new
    Runnable() {
        @Override public void
        run() {
            t.setText(Integer.toString(ctr));
        }
    });
}

synchronized (lock){
    while(paused){
        try{
            lock.wait();
        }catch(InterruptedException e){
            }
    }
}
```

```
    }

} catch (InterruptedException e) {paused =
    true; e.printStackTrace();

}

}

}

}
```

Activity_main.xml:

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

<androidx.constraintlayout.widget.ConstraintLayout
    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" tools:context=".MainActivity"
    tools:layout_editor_absoluteX="-1dp"
    tools:layout_editor_absoluteY="-83dp">

    <TextView
```

```
    android:id="@+id/t1" android:layout_width="wrap_content"
    android:layout_height="wrap_content" android:text="Hello
    World!" android:textSize="24sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.383" />
```

```
<TextView android:id="@+id/t2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" android:text="Hello
    World!" android:textSize="24sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintHorizontal_bias="0.501"
    app:layout_constraintLeft_toLeftOf="parent"
```

```
    app:layout_constraintRight_toRightOf="parent"  
    app:layout_constraintTop_toTopOf="parent" app:layout_constraintVertical_bias="0.266"  
/>
```

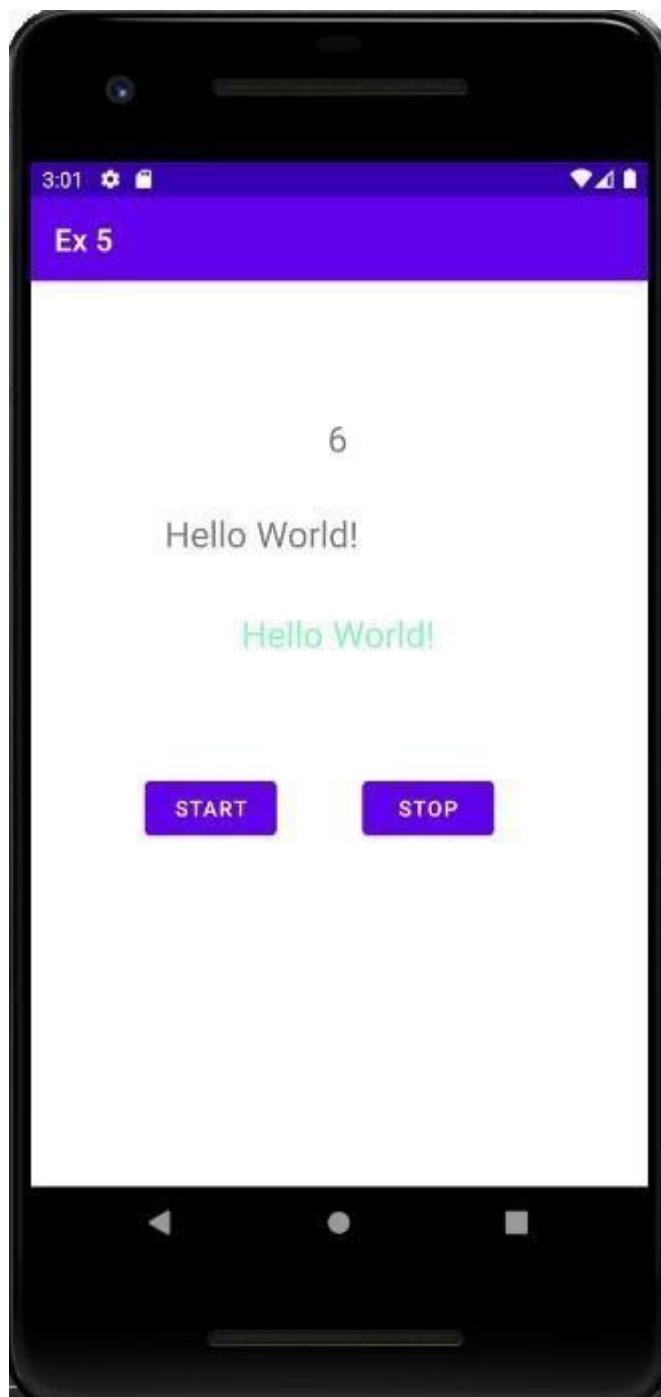
```
<TextView android:id="@+id/t3"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="0" android:textSize="24sp"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintHorizontal_bias="0.498"  
    app:layout_constraintLeft_toLeftOf="parent"  
    app:layout_constraintRight_toRightOf="parent"  
    app:layout_constraintTop_toTopOf="parent" app:layout_constraintVertical_bias="0.155"  
/>
```

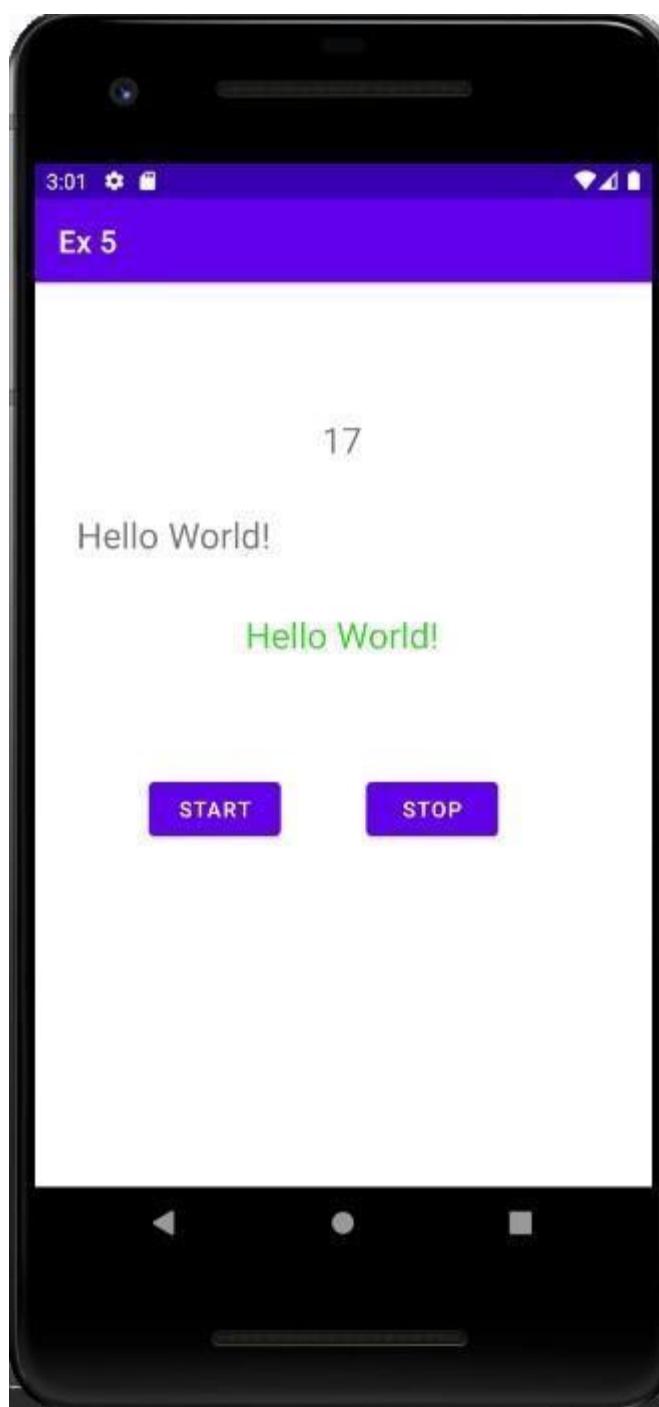
```
<Button  
    android:id="@+id/stop" android:layout_width="wrap_content"  
    android:layout_height="wrap_content"
```

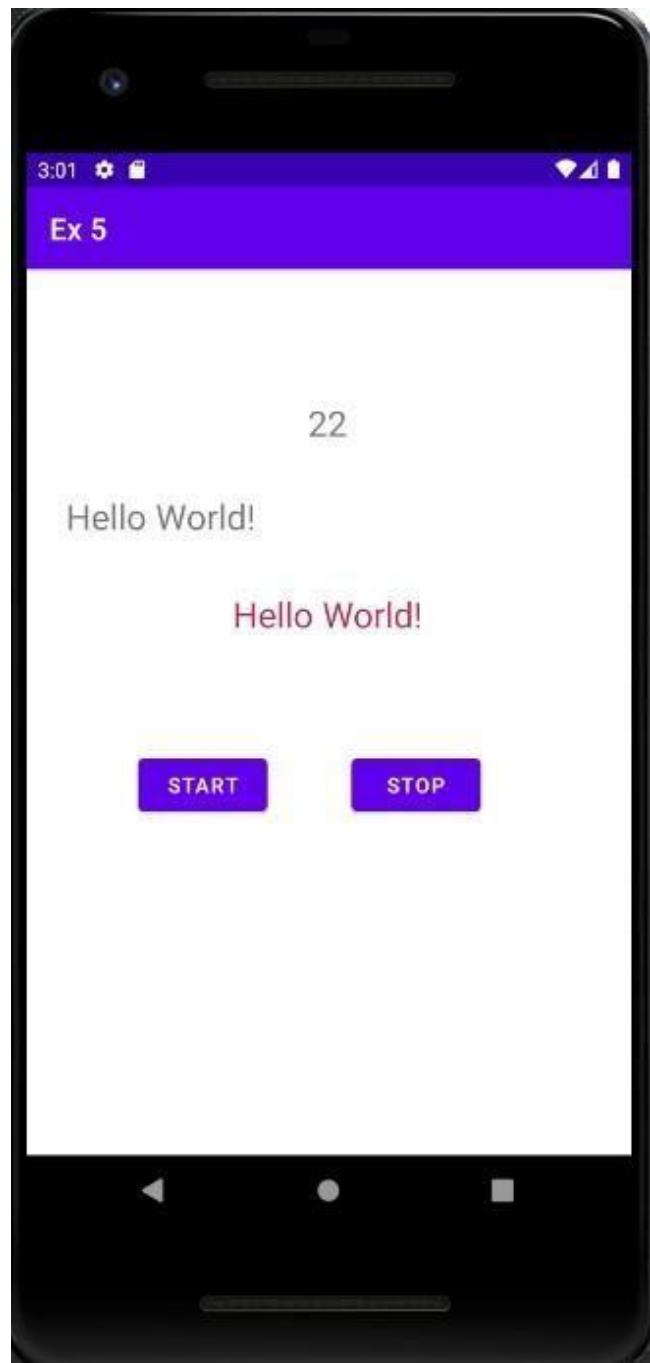
```
    android:layout_marginTop="76dp" android:text="Stop"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.684"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toBottomOf="@+id/t1" />
```

```
<Button  
    android:id="@+id/start" android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_marginTop="76dp" android:text="Start"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.236"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toBottomOf="@+id/t1" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>
```

Output:









Best Practices:

- Names for ids of buttons were set meaningfully
- Implemented pause and resume in a single function

- Handled exceptions

Learning Outcomes:

- Learnt to implement multithreading
- Learnt to start, stop and resume threads

Ex. No. 6 Android Application for Location Tracking

Aim: 1. Develop an Android Application that uses Geographical Positioning System (GPS) to display the user's current location in terms of Latitude and Longitude. 2. Develop a mobile app to display the Geo location of a given place.

Layouts Use: None. TextViews.

Code:

MainActivity.java:

```
package com.example.ex6;

import android.Manifest;
import android.annotation.SuppressLint;import
android.content.Context;
import android.content.Intent;
import android.content.pm.PackageManager;import
android.location.Location;
import android.location.LocationManager;import
android.os.Bundle;
import android.os.Looper; import
android.provider.Settings;import
android.util.Log;
import android.view.View; import
android.widget.Button; import
android.widget.EditText;import
android.widget.TextView;import
android.widget.Toast;
```

```
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;import
androidx.core.app.ActivityCompat;

import com.google.android.gms.location.FusedLocationProviderClient;import
com.google.android.gms.location.LocationCallback;
import com.google.android.gms.location.LocationRequest; import
com.google.android.gms.location.LocationResult; import
com.google.android.gms.location.LocationServices;import
com.google.android.gms.tasks.OnCompleteListener;import
com.google.android.gms.tasks.Task;

import android.location.Address; import
android.location.Geocoder;import
android.widget.Toast; import
java.io.IOException;
import java.util.List;

public class MainActivity extends AppCompatActivity {

    // initializing
    // FusedLocationProviderClient
    // object
    FusedLocationProviderClient mFusedLocationClient;

    // Initializing other items
    // from layout file
    TextView latitudeTextView, longitTextView;int
    PERMISSION_ID = 44;

    @Override
    protected void onCreate(Bundle savedInstanceState) {super.onCreate(savedInstanceState);
```

```
setContentView(R.layout.activity_main);

latitudeTextView = findViewById(R.id.latTextView);
longitTextView = findViewById(R.id.lonTextView);

mFusedLocationClient = LocationServices.getFusedLocationProviderClient(this);

// method to get the location
getLastLocation();

Button display = findViewById(R.id.dis); display.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String loc = ((EditText)
findViewById(R.id.loc)).getText().toString();Log.d("debug",loc);
        getLatLngForPlace(loc);
    }
});
```

```
@SuppressLint("MissingPermission")private
void getLastLocation() {
    // check if permissions are givenif
    (checkPermissions()) {

        // check if location is enabledif
        (isLocationEnabled()) {

            // getting last
            // location from
            // FusedLocationClient
```

```
// object

mFusedLocationClient.getLastLocation().addOnCompleteListener(new
OnCompleteListener<Location>() {
    @Override
    public void onComplete(@NonNull Task<Location> task) {Location location
        = task.getResult();
        if (location == null) {
            requestNewLocationData();
        } else {
            latitudeTextView.setText(location.getLatitude() + "");
            longitTextView.setText(location.getLongitude() + "");
        }
    }
});

} else {
    Toast.makeText(this, "Please turn on" + " your location...",
Toast.LENGTH_LONG).show();
    Intent intent = new
Intent(Settings.ACTION_LOCATION_SOURCE_SETTINGS);
    startActivity(intent);
}
} else {
    // if permissions aren't available,
    // request for permissions
    requestPermissions();
}
}

@SuppressLint("MissingPermission") private void
requestNewLocationData() {

// Initializing LocationRequest
// object with appropriate methods
LocationRequest mLocationRequest = new LocationRequest();
```

```
mLocationRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);
    mLocationRequest.setInterval(5); mLocationRequest.setFastestInterval(0);
    mLocationRequest.setNumUpdates(1);

    // setting LocationRequest
    // on FusedLocationClient
    mFusedLocationClient =
LocationServices.getFusedLocationProviderClient(this);
    mFusedLocationClient.requestLocationUpdates(mLocationRequest,
mLocationCallback, Looper.myLooper());
}

private LocationCallback mLocationCallback = new LocationCallback() {

    @Override
    public void onLocationResult(LocationResult locationResult) { Location mLastLocation
        = locationResult.getLastLocation(); latitudeTextView.setText("Latitude: " +
        mLastLocation.getLatitude()
+ "");
        longitudeTextView.setText("Longitude: " +
        mLastLocation.getLongitude() + "");
    }
};

// method to check for permissions private
boolean checkPermissions() {
    return ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) ==
PackageManager.PERMISSION_GRANTED &&
ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) ==
PackageManager.PERMISSION_GRANTED;
```

```
// If we want background location
// on Android 10.0 and higher,
// use:
// ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_BACKGROUND_LOCATION) ==
PackageManager.PERMISSION_GRANTED
}

// method to request for permissionsprivate
void requestPermissions() {
    ActivityCompat.requestPermissions(this, new String[]{
        Manifest.permission.ACCESS_COARSE_LOCATION,
        Manifest.permission.ACCESS_FINE_LOCATION},
    PERMISSION_ID);
}

// method to check
// if location is enabled
private boolean isLocationEnabled() {
    LocationManager locationManager = (LocationManager)
getSystemService(Context.LOCATION_SERVICE);
    return locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER) ||
locationManager.isProviderEnabled(LocationManager.NETWORK_PROVIDER);
}

// If everything is alright then@Override
public void
onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull
int[] grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions, grantResults);
```

```
if (requestCode == PERMISSION_ID) {
    if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
        getLastLocation();
    }
}
}

@Override
public void onResume() {
    super.onResume();
    if (checkPermissions()) {
        getLastLocation();
    }
}

private void getLatLngForPlace(String placeName) {Geocoder
    geocoder = new Geocoder(this);

    try {
        List<Address> addresses = geocoder.getFromLocationName(placeName, 1);
        if (addresses != null && !addresses.isEmpty()) {Address address =
            addresses.get(0);
            double latitude = address.getLatitude(); double
            longitude = address.getLongitude();
            latitudeTextView.setText("'" + latitude);
            longitTextView.setText("'" + longitude);
        } else {
            // Handle the case where the place name couldn't be geocoded
            Toast.makeText(this, "Place not found",
Toast.LENGTH_SHORT).show();
        }
    } catch (IOException e) {
```

```
        e.printStackTrace();
    }
}

}
```

Activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    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" tools:context=".MainActivity">

    <TextView android:id="@+id/textView4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="48dp"
        android:fontFamily="sans-serif-black"
        android:text="Enter location:" android:textSize="24sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/lonTextView"
        tools:ignore="MissingConstraints" />

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
        android:layout_marginTop="124dp"
        android:fontFamily="sans-serif-black"
        android:text="Latitude:" android:textSize="24sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.4"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        tools:ignore="MissingConstraints" />

<TextView android:id="@+id/latTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="36dp"
        android:text="" android:textSize="24sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.406"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView"
        tools:ignore="MissingConstraints" />

<TextView android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="24dp"
        android:fontFamily="sans-serif-black"
        android:text="Longitude:" android:textSize="24sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.427"
        app:layout_constraintStart_toStartOf="parent"
```

```
    app:layout_constraintTop_toBottomOf="@+id/latTextView"
    tools:ignore="MissingConstraints" />

<TextView android:id="@+id/lonTextView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="48dp"
    android:text="" android:textSize="24sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.44"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView2"
    tools:ignore="MissingConstraints" />
```

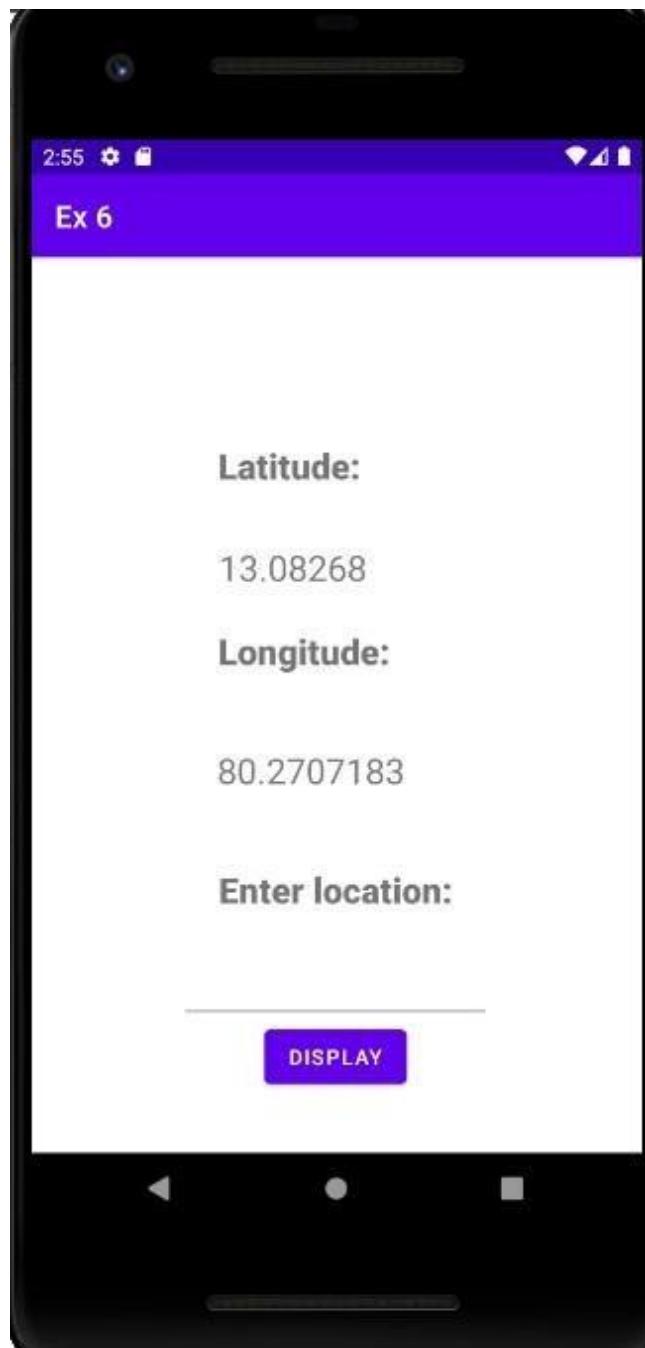
```
<Button
    android:id="@+id/dis" android:layout_width="wrap_content"
    android:layout_height="wrap_content" android:text="Display"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView4"
    app:layout_constraintVertical_bias="0.639" />
```

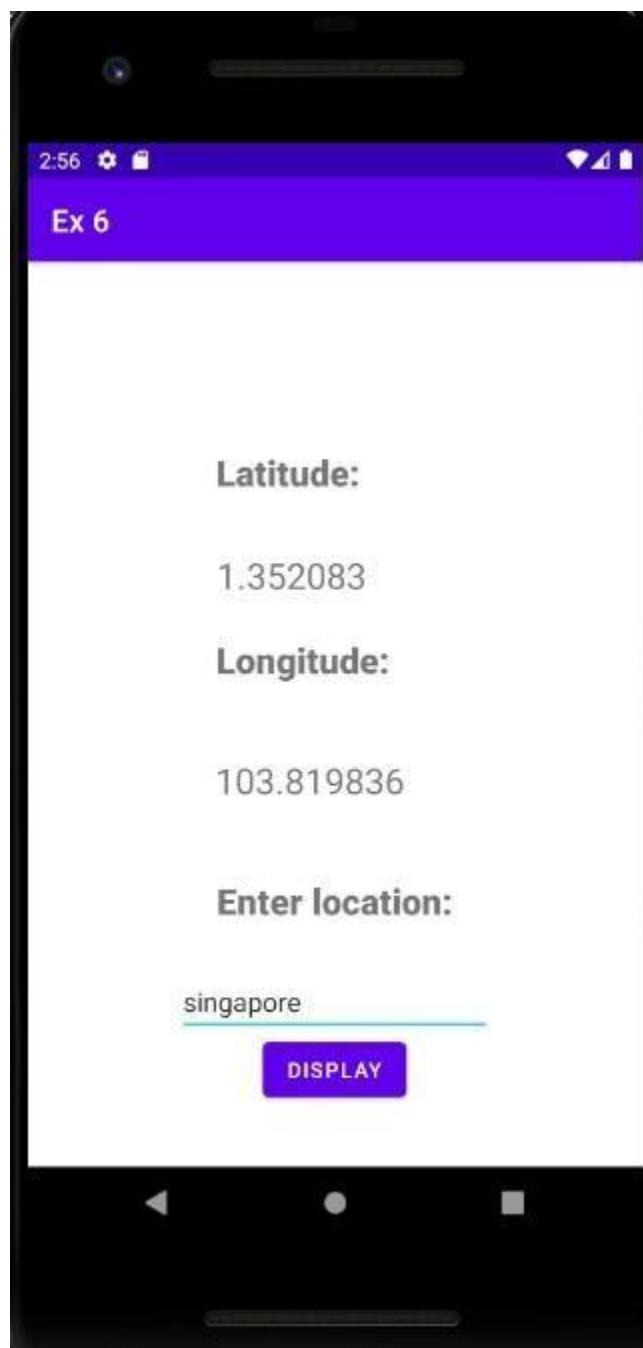
```
<EditText
    android:id="@+id/loc"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" android:ems="10"
    android:inputType="textPersonName"
    app:layout_constraintBottom_toBottomOf="parent"
```

```
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.497"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView4"
    app:layout_constraintVertical_bias="0.25" />
```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

Output:





Best Practices:

- Used meaningful ids
- Aligned the textviews

Learning Outcomes:

- | Learnt to retrieve coordinates of current locations
- | Learnt to retrieve coordinates of a different location using geocoder

Ex 7: Android Application to write/read a file to/from the SD Card

Aim:

Develop an Android Application to write/read some contents to/from the SD Card.

1. In a TextView write the contents of the file.
2. Use another TextView to read the file name from user.
3. On clicking 'Write' Button,

Create a file mentioned in 2nd TextView.

Write the contents (using 1st TextView) in the file.

Store the file in the SD card.

4. On clicking 'Read' Button,

Move to a new activity.

Read the file name(TextView)

Read the contents of the file from SD card and display in a new TextView.

Layouts Used: Main Activity and Read Intents. Edit and Text Views.

Code:

MainActivity.java:

```
package com.example.ex7;
```

```
import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.os.Bundle;
```

```
import android.os.Environment;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;

public class MainActivity extends AppCompatActivity {

    private static final int REQUEST_WRITE_EXTERNAL_STORAGE = 1;
    @Override

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

```
// Request the WRITE_EXTERNAL_STORAGE permission if not granted

    if (ContextCompat.checkSelfPermission(this,
Manifest.permission.WRITE_EXTERNAL_STORAGE)

        != PackageManager.PERMISSION_GRANTED) {

            ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.WRITE_EXTERNAL_STORAGE},
REQUEST_WRITE_EXTERNAL_STORAGE);

    } else {

        // Permission already granted, perform file operations

        Button button = findViewById(R.id.write);

        button.setOnClickListener(new View.OnClickListener() {

            @Override

            public void onClick(View v) {

                EditText et1 = findViewById(R.id.et1);

                String file = et1.getText().toString();




                EditText et2 = findViewById(R.id.et2);

                String content = et2.getText().toString();

                createAndWriteFileToSDCard(file,content);

            }

        });

    }

    Button read = findViewById(R.id.read1);

    read.setOnClickListener(new View.OnClickListener() {
```

```
    @Override  
  
    public void onClick(View v) {  
  
        Intent intent = new Intent(MainActivity.this, Read.class);  
  
        startActivity(intent);  
  
    }  
  
});  
  
}  
  
  
// Handle permission request results  
  
@Override  
  
public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {  
  
    super.onRequestPermissionsResult(requestCode, permissions, grantResults);  
  
    if (requestCode == REQUEST_WRITE_EXTERNAL_STORAGE) {  
  
        if (grantResults.length > 0 && grantResults[0] ==  
PackageManager.PERMISSION_GRANTED) {  
  
            Toast.makeText(this, "Permission granted. Can write to SD card.",  
Toast.LENGTH_SHORT).show();  
  
            Button button = findViewById(R.id.write);  
            button.setOnClickListener(new View.OnClickListener() {  
  
                @Override  
  
                public void onClick(View v) {  
  
                    EditText et1 = findViewById(R.id.et1);  
  
                    String file = et1.getText().toString();  
                }  
            });  
        }  
    }  
}
```

```
        EditText et2 = findViewById(R.id.et2);
        String content = et2.getText().toString();
        createAndWriteFileToSDCard(file,content);
    }
});
```

```
} else {
```

```
    Toast.makeText(this, "Permission denied. Cannot write to SD card.",  
Toast.LENGTH_SHORT).show();
```

```
}
```

```
}
```

```
}
```

```
private void createAndWriteFileToSDCard(String fileName,String fileContent) { //
```

```
    Check if external storage is available
```

```
    if (isExternalStorageWritable()) {
```

```
        File sdCard = Environment.getExternalStorageDirectory();
```

```
        File directory = new File(sdCard.getAbsolutePath() + "/ex7"); // Change to  
your desired directory  
        directory.mkdirs();
```

```
        File file = new File(directory, fileName+".txt"); // Change the file name as  
needed
```

```
    try {
```

```
        FileOutputStream fos = new FileOutputStream(file);
```

```

        fos.write(fileContent.getBytes());

        fos.close();

        Toast.makeText(this, "File created and written to SD card",
Toast.LENGTH_SHORT).show();

    } catch (IOException e) {

        Log.e("FileWriteError", "Error writing to file on SD card: " +
e.getMessage());

    }

} else {

    Toast.makeText(this, "SD card is not available for writing.",
Toast.LENGTH_SHORT).show();

}

}

```

```

private boolean isExternalStorageWritable() {

    String state = Environment.getExternalStorageState();

    return Environment.MEDIA_MOUNTED.equals(state);

}

```

Read.java:

```
package com.example.ex7;
```

```
import android.Manifest;
```

```
import android.content.pm.PackageManager;
import android.os.Bundle;
import android.os.Environment;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat; import
androidx.core.content.ContextCompat;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.io.IOException;

public class Read extends AppCompatActivity {
```

```
private static final int REQUEST_READ_EXTERNAL_STORAGE = 2;

private TextView fileContentsTextView;

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

    fileContentsTextView = findViewById(R.id.content);

    // Request the READ_EXTERNAL_STORAGE permission if not granted
    if (ContextCompat.checkSelfPermission(this,
        Manifest.permission.READ_EXTERNAL_STORAGE)
        != PackageManager.PERMISSION_GRANTED) {

        ActivityCompat.requestPermissions(this, new
            String[]{Manifest.permission.READ_EXTERNAL_STORAGE},
            REQUEST_READ_EXTERNAL_STORAGE);

    } else {
        // Permission already granted, perform file reading
        Button readButton = findViewById(R.id.read2);

        readButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                EditText fileNameEditText = findViewById(R.id.name);
            }
        });
    }
}
```

```
        String fileName = fileNameEditText.getText().toString();

        readFileFromSDCard(fileName);

    }

});
```

}

}

// Handle permission request results

```
@Override

public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {

    super.onRequestPermissionsResult(requestCode, permissions,
grantResults);

    if (requestCode == REQUEST_READ_EXTERNAL_STORAGE) {

        if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {

            Toast.makeText(this, "Permission granted. Can read from SD card.", Toast.LENGTH_SHORT).show();

            Button readButton = findViewById(R.id.read2);

            readButton.setOnClickListener(new View.OnClickListener() {
                @Override

                public void onClick(View v) {

                    EditText fileNameEditText = findViewById(R.id.name);

                    String fileName = fileNameEditText.getText().toString();

                    readFileFromSDCard(fileName);

                }
            });
        }
    }
}
```

```
    });

} else {

    Toast.makeText(this, "Permission denied. Cannot read from SD card.",  
Toast.LENGTH_SHORT).show();

}

}

}
```

```
private void readFileFromSDCard(String fileName) {

if (isExternalStorageReadable()) {

    File sdCard = Environment.getExternalStorageDirectory();

    File directory = new File(sdCard.getAbsolutePath() + "/ex7"); // Change to  
your directory

    File file = new File(directory, fileName + ".txt");



if (file.exists()) {

    try {

        BufferedReader br = new BufferedReader(new FileReader(file));
        StringBuilder text = new StringBuilder();

        String line;





while ((line = br.readLine()) != null) {

        text.append(line);

        text.append('\n');

    }

}
```

```

        br.close();

        fileContentsTextView.setText(text.toString());

    } catch (IOException e) {

        Log.e("FileReadError", "Error reading file on SD card: " +
e.getMessage());

        fileContentsTextView.setText("Error reading file.");

    }

} else {

    fileContentsTextView.setText("File not found.");

}

} else {

    fileContentsTextView.setText("SD card is not available for reading.");

}

}

private boolean isExternalStorageReadable() {
    String state = Environment.getExternalStorageState();

    return Environment.MEDIA_MOUNTED.equals(state) ||
Environment.MEDIA_MOUNTED_READ_ONLY.equals(state);

}

}

```

Activity_main.xml:

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

```
<androidx.constraintlayout.widget.ConstraintLayout
    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"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/tv2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Enter text"
        android:textSize="24sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.387" />

    <EditText
        android:id="@+id/et1"
```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="184dp"
    android:ems="10"
    android:inputType="textPersonName"
    android:text=""
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.497"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
/>

```

```
<EditText
    android:id="@+id/et2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:ems="10"
    android:inputType="textPersonName"
    android:text=""
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.497"
```

```
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent" />
```

```
<TextView  
        android:id="@+id/tv1"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:layout_marginTop="128dp"  
        android:text="Enter file name"  
        android:textSize="24sp"  
        app:layout_constraintEnd_toEndOf="parent"  
        app:layout_constraintStart_toStartOf="parent"  
        app:layout_constraintTop_toTopOf="parent"  
    />
```

```
<Button  
        android:id="@+id/write"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:text="Write"  
        app:layout_constraintBottom_toBottomOf="parent"  
        app:layout_constraintEnd_toEndOf="parent"  
        app:layout_constraintHorizontal_bias="0.498"
```

```
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.65" />

<Button
    android:id="@+id/read1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="156dp"
    android:text="Read"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Read.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    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">

        <TextView
            android:id="@+id/textView3"
            android:layout_width="207dp"
            android:layout_height="47dp"
            android:text="File Content:"
            android:textSize="24sp"
            app:layout_constraintBottom_toBottomOf="parent"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="parent"
            app:layout_constraintVertical_bias="0.602" />

        <TextView
            android:id="@+id/textView"
            android:layout_width="125dp"
            android:layout_height="50dp"
            android:text="File Name"
            android:textSize="24sp"
            app:layout_constraintBottom_toBottomOf="parent"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintHorizontal_bias="0.461"
```

```
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:layout_constraintVertical_bias="0.19" />
```

```
<EditText  
        android:id="@+id/name"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:ems="10"  
        android:inputType="textPersonName"  
        android:text=""  
        android:textSize="24sp"  
        app:layout_constraintBottom_toBottomOf="parent"  
        app:layout_constraintEnd_toEndOf="parent"  
        app:layout_constraintHorizontal_bias="0.496"  
        app:layout_constraintStart_toStartOf="parent"  
        app:layout_constraintTop_toTopOf="parent"  
        app:layout_constraintVertical_bias="0.29" />  
  
<EditText  
        android:id="@+id/content"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:ems="10"
```

```
    android:inputType="textPersonName"  
    android:text=""  
    android:textSize="24sp"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.496"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:layout_constraintVertical_bias="0.724" />
```

```
<Button  
    android:id="@+id/read2"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Read"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.498"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:layout_constraintVertical_bias="0.407" />
```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

AndroidManifest.xml:

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

<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.ex7">

    <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/Theme.Ex7">

        <activity
            android:name=".MainActivity"
            android:exported="true">

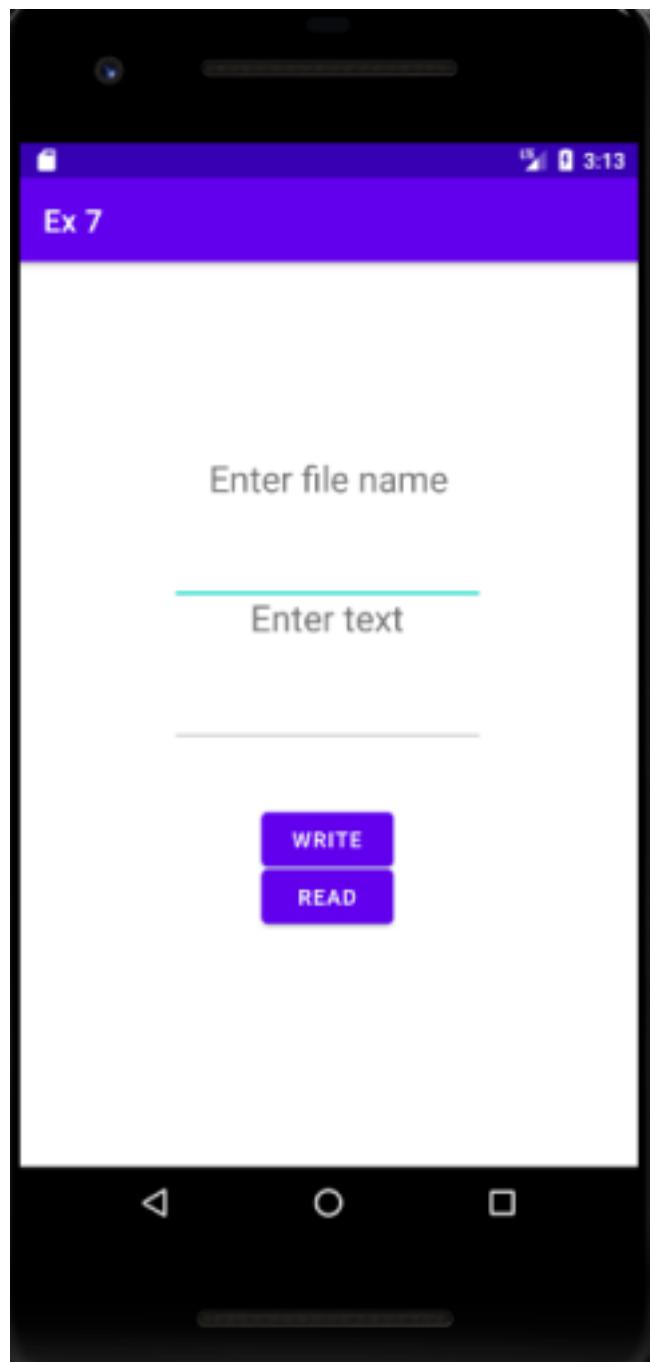
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>

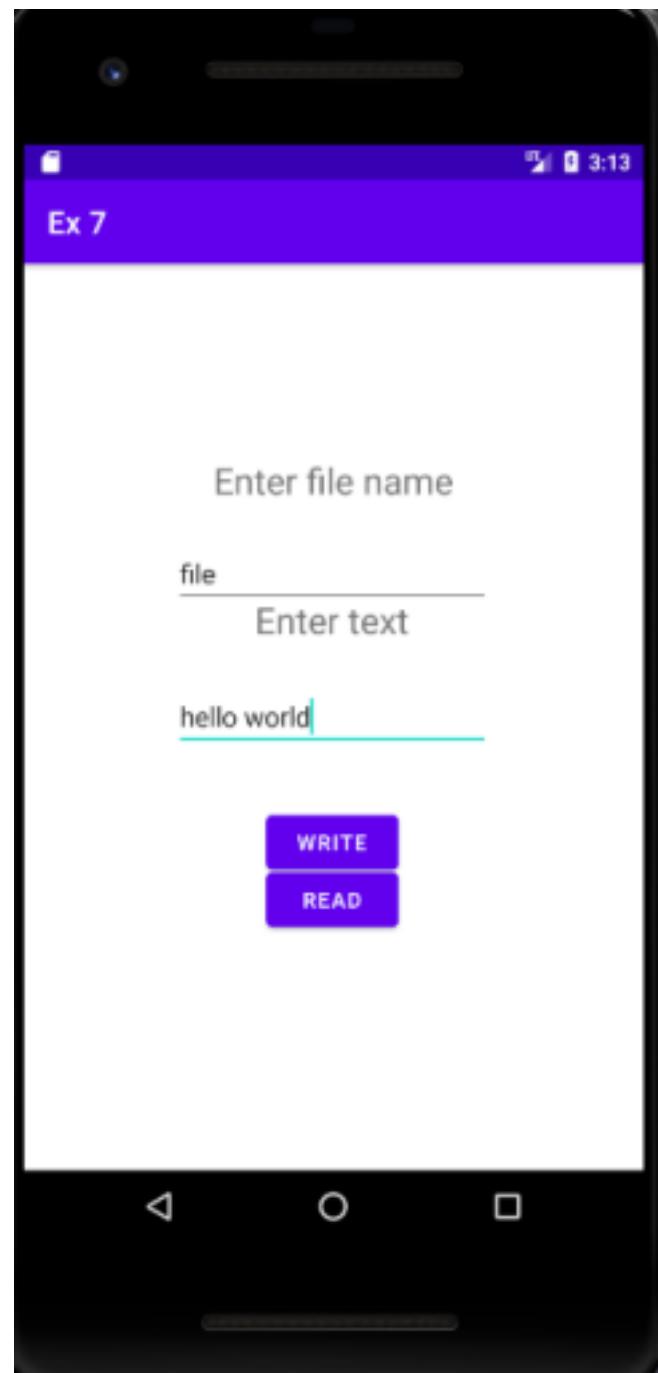
        </activity>
        <activity android:name=".Read"></activity>
    </application>

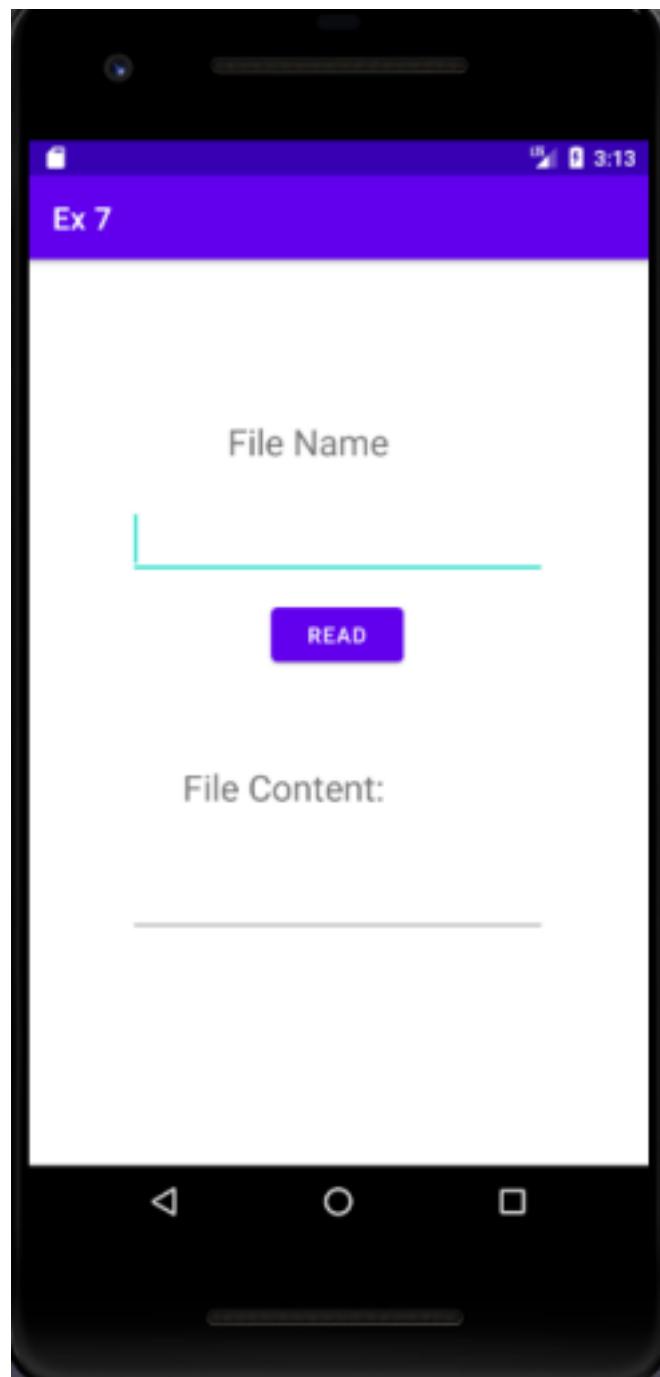
    <uses-permission
        android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

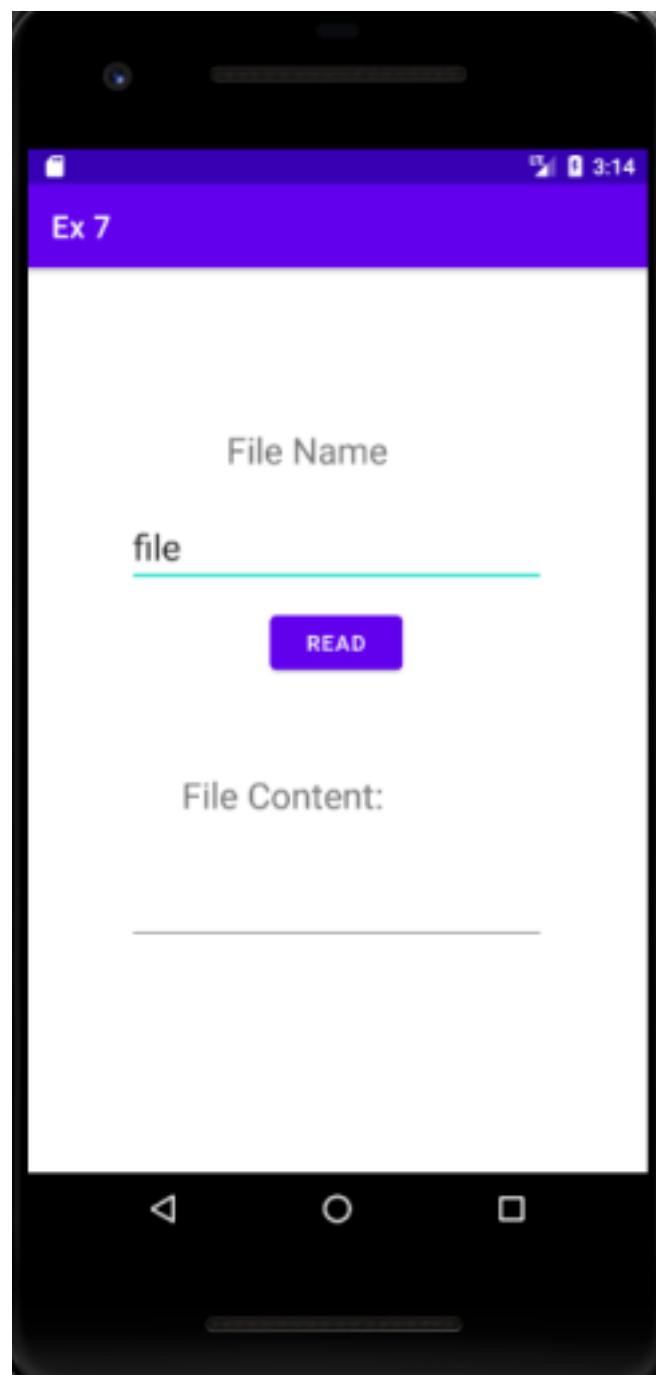
```
<uses-permission  
    android:name="android.permission.READ_EXTERNAL_STORAGE" />  
  
<uses-permission  
    android:name="android.permission.MANAGE_EXTERNAL_STORAGE" />  
  
</manifest>
```

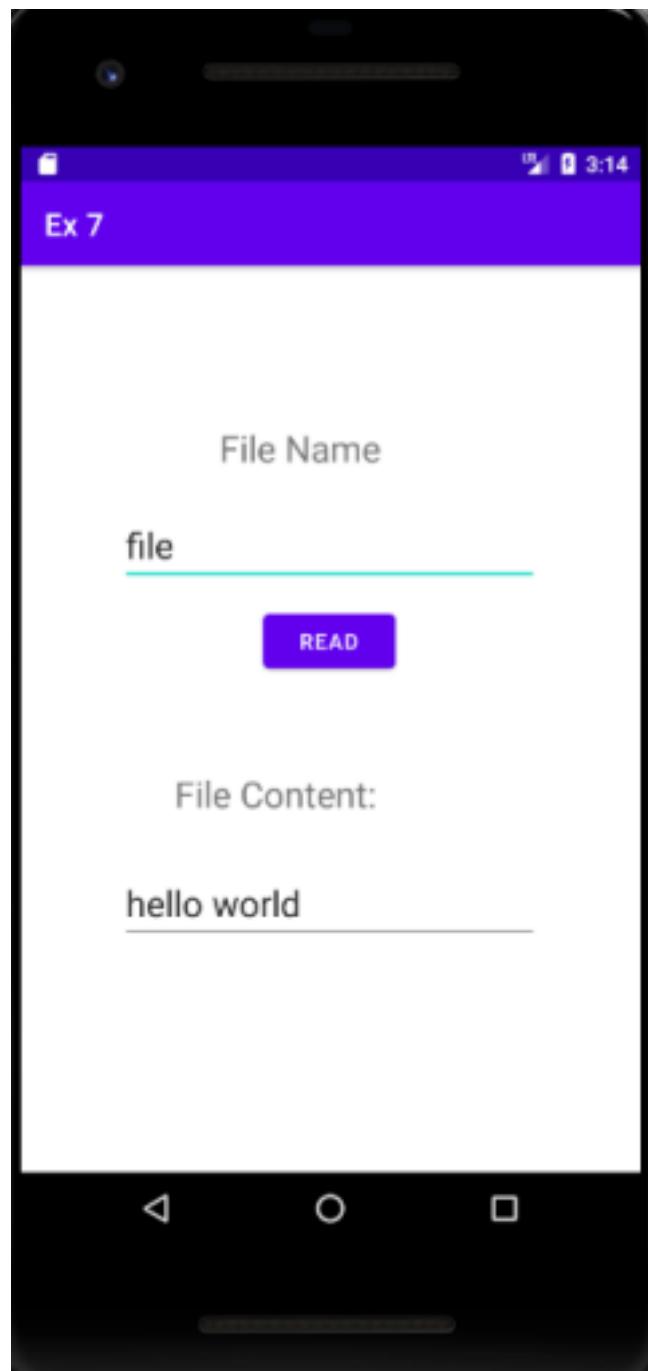
Output:

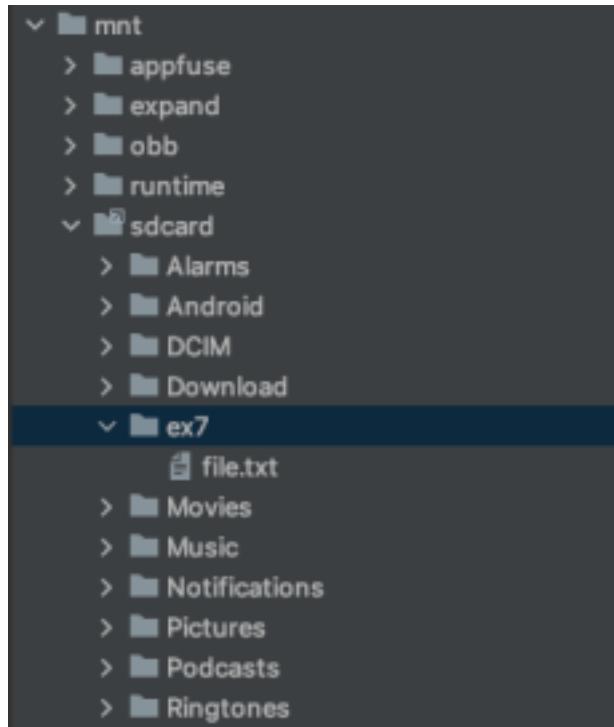












Best Practices:

- Used appropriate ids for buttons, views and intents
- Aligned views.

Learning Outcomes:

- Learnt to create a file
- Learnt to write to a file and store the file in SD card
- Learnt to read from a file that is in SD card

Assignment 8 SMS Sending and Notification

Objective:

To develop an android app that sends SMS and creates an alert upon receiving the SMS with text in the notification.

Source Code:

MainActivity.java:

```
package com.example.ex8;
import android.Manifest;
import android.content.pm.PackageManager;
import android.os.Bundle;
import android.telephony.SmsManager;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

public class MainActivity extends AppCompatActivity {
    private static final int MY_PERMISSIONS_REQUEST_SEND_SMS = 0;
    Button sendBtn;
    EditText txtphoneNo;
    EditText txtMessage;
    String phoneNo;
    String message;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        sendBtn = (Button) findViewById(R.id.sendbtn);
        txtphoneNo = (EditText) findViewById(R.id.etPhone);
        txtMessage = (EditText) findViewById(R.id.content);
        sendBtn.setOnClickListener(new View.OnClickListener() {
```

```
public void onClick(View view) {
    sendSMSMessage();
}
});

}

protected void sendSMSMessage() {
    phoneNo = txtphoneNo.getText().toString();
    message = txtMessage.getText().toString();
    if (ContextCompat.checkSelfPermission(this, Manifest.permission.SEND_SMS)
        != PackageManager.PERMISSION_GRANTED) {
        if (ActivityCompat.shouldShowRequestPermissionRationale(
            this, Manifest.permission.SEND_SMS)) {
        } else {
            ActivityCompat.requestPermissions(this,
                new String[] {Manifest.permission.SEND_SMS},
                MY_PERMISSIONS_REQUEST_SEND_SMS);
        }
    }
}

@Override
public void onRequestPermissionsResult(int requestCode, String permissions[], int[] grantResults)
switch (requestCode) {
    case MY_PERMISSIONS_REQUEST_SEND_SMS: {
        if (grantResults.length > 0
            && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
            SmsManager smsManager = SmsManager.getDefault();
            smsManager.sendTextMessage(phoneNo, null, message, null, null);
            Toast.makeText(getApplicationContext(), "SMS sent.", Toast.LENGTH_LONG).show();
        }
        else {
            Toast.makeText(getApplicationContext(), "SMS failed, please try again.", Toast.LENGTH_LONG).show();
            return;
        }
    }
}
```

```
}
```

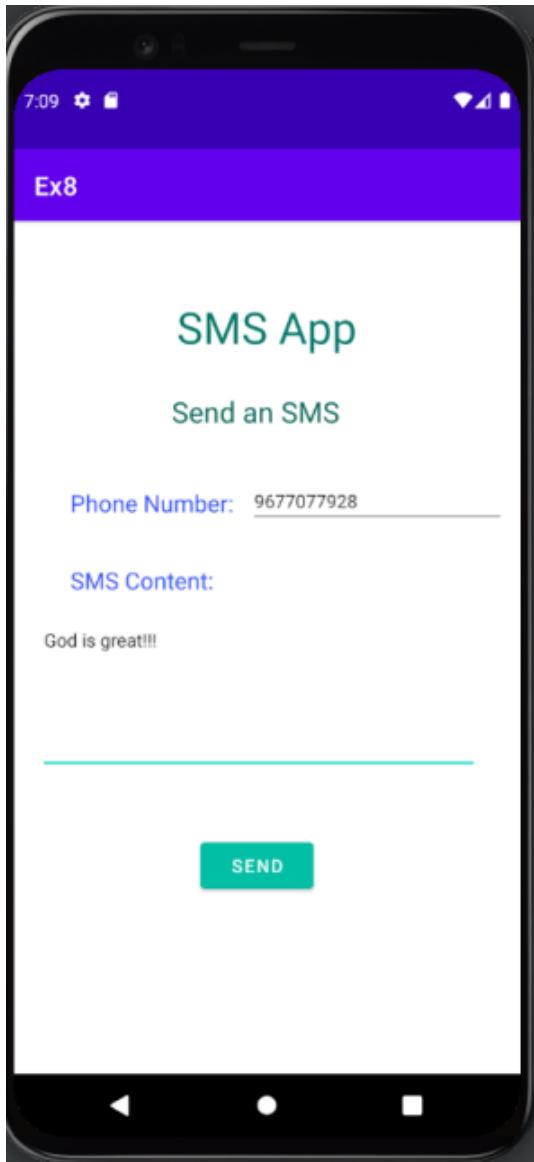
activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    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"
    tools:context=".MainActivity">
    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="SMS App"
        android:textAppearance="@style/TextAppearance.AppCompat.Display1" android:textColor="#067A6A"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.096" />
    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="28dp"
        android:layout_marginEnd="140dp"
        android:text="Send an SMS"
        android:textAppearance="@style/TextAppearance.AppCompat.Large"
        android:textColor="#0C675A"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView2" />
    <TextView
        android:id="@+id/phno"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```

```
        android:layout_marginStart="44dp"
        android:layout_marginTop="44dp"
        android:text="Phone Number:"
        android:textAppearance="@style/TextAppearance.AppCompat.Medium"
        android:textColor="#304FFE"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView" />
    <TextView
        android:id="@+id/smstv"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="44dp"
        android:layout_marginTop="104dp"
        android:text="SMS Content:"
        android:textAppearance="@style/TextAppearance.AppCompat.Medium" android:textColor="#304FFE"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView" />
    <EditText
        android:id="@+id/etPhone"
        android:layout_width="199dp"
        android:layout_height="39dp"
        android:layout_marginStart="12dp"
        android:layout_marginTop="36dp"
        android:ems="10"
        android:inputType="phone"
        android:textAppearance="@style/TextAppearance.AppCompat.Body1"
        app:layout_constraintStart_toEndOf="@+id/phno"
        app:layout_constraintTop_toBottomOf="@+id/textView" />
    <EditText
        android:id="@+id/content"
        android:layout_width="341dp"
        android:layout_height="123dp"
        android:layout_marginTop="68dp"
        android:layout_marginEnd="32dp"
        android:ems="10"
        android:gravity="start|top"
        android:hint="Type here"
```

```
        android:inputType="textMultiLine"
        android:textAppearance="@style/TextAppearance.AppCompat.Body1"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/etPhone" />
    <Button
        android:id="@+id/sendbtn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="48dp"
        android:layout_marginEnd="160dp"
        android:backgroundTint="#00BFA5"
        android:text="SEND"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/content" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Output:



Result:

The required program was built and executed successfully.

Assignment 9

Alarm Clock - Android Application

Develop an Alarm Clock Android Application.

1. Use permission for WAKE_LOCK.
2. Have a TimePicker component followed by a ToggleButton to select time and Alarm On / Off.
3. Use the AlarmManager to set the alarm and send notification on the alarm trigger.
4. Perform 3 different notifications
 - a. Show a message to user in the activity UI
 - b. Play the alarm ringtone
 - c. Send an Android notification message

Code:

Activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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"
    tools:context=".MainActivity">
    <TimePicker
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/time"
        />
    <ToggleButton
        android:id="@+id/set"
        android:layout_width="150dp"
        android:layout_height="50dp"
        android:layout_below="@+id/time"
        android:layout_marginLeft="130dp"
        />
</RelativeLayout>
```

Display.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
```

```
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
android:layout_width="match_parent"
android:layout_height="match_parent">
<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Close App to turn off alarm!"
    android:padding="50dp"
    android:textSize="20dp"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Mainactivity.java:

```
package com.example.alarm;

import androidx.appcompat.app.AppCompatActivity;
import android.app.AlarmManager;
import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.Intent;
import android.os.Build;
import android.os.Bundle;
import android.text.format.Time;
import android.view.View;
import android.widget.Button;
import android.widget.TimePicker;
import android.widget.Toast;
import android.widget.ToggleButton;
import com.example.alarm.databinding.ActivityMainBinding; import java.util.Calendar;
public class MainActivity extends AppCompatActivity { private ActivityMainBinding binding;
@Override
protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
binding = ActivityMainBinding.inflate(getLayoutInflater()); setContentView(binding.getRoot());
```

```
createNotificationChannel();
ToggleButton b1 = findViewById(R.id.set);
TimePicker t1 = findViewById(R.id.time);
t1.setIs24HourView(true);
b1.setOnCheckedChangeListener((buttonView, isChecked) -> {
if (isChecked) {
setAlarm();
} else {
cancelAlarm();
}
});
}

private void createNotificationChannel() {
if(Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
CharSequence name = "alarm";
String desc = "Channel for alarm";
int importance = NotificationManager.IMPORTANCE_HIGH;
NotificationChannel channel = new NotificationChannel("alarm", name, importance);
channel.setDescription(desc);
NotificationManager notifs =
getSystemService(NotificationManager.class);
notifs.createNotificationChannel(channel);
}
}

public void setAlarm() {
TimePicker timePicker = findViewById(R.id.time);
AlarmManager alarm = (AlarmManager) getSystemService(ALARM_SERVICE);
Intent intent = new Intent(this, AlarmReceiver.class);
PendingIntent pendingIntent =
PendingIntent.getBroadcast(this.getApplicationContext(), 234, intent, PendingIntent.FLAG_IMMUTABLE);
int hour = timePicker.getHour();
int minute = timePicker.getMinute();
Calendar calendar = Calendar.getInstance();
calendar.set(Calendar.HOUR_OF_DAY, hour);
calendar.set(Calendar.MINUTE, minute);
calendar.set(Calendar.SECOND, 0);
long triggerTime = calendar.getTimeInMillis();
int timeInSec = 1;
```

```
alarm.set(AlarmManager.RTC_WAKEUP, triggerTime, pendingIntent); Toast.makeText(this, "Alarm set for "
+ hour + ":" + minute, Toast.LENGTH_SHORT).show();
}

public void cancelAlarm() {
AlarmManager alarm = (AlarmManager) getSystemService(ALARM_SERVICE);
Intent intent = new Intent(this, AlarmReceiver.class);
PendingIntent pendingIntent =
PendingIntent.getBroadcast(this.getApplicationContext(), 234, intent, PendingIntent.FLAG_IMMUTABLE);
if(alarm != null) {
alarm.cancel(pendingIntent);
}
Toast.makeText(this, "Alarm unset!", Toast.LENGTH_SHORT).show(); }
```

Alarmreceiver.java:

```
package com.example.alarm;
import static androidx.core.content.ContextCompat.startActivity;
import android.app.AlarmManager;
import android.app.Notification;
import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.media.Ringtone;
import android.media.RingtoneManager;
import android.net.Uri;
import android.os.Build;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.app.NotificationCompat;
import androidx.core.app.NotificationManagerCompat;
public class AlarmReceiver extends BroadcastReceiver {
static Uri alarmrt1 =
```

```
RingtoneManager.getDefaultUri(RingtoneManager.TYPE_ALARM);
@Override
public void onReceive(Context context, Intent intent) {
    Toast.makeText(context, "INSIDE WOHOOO", Toast.LENGTH_LONG).show();
    Intent i = new Intent(context, In.class);
    intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK |
    Intent.FLAG_ACTIVITY_CLEAR_TASK);
    PendingIntent p = PendingIntent.getActivity(context, 0, i, PendingIntent.FLAG_IMMUTABLE);
    NotificationCompat.Builder builder = new
    NotificationCompat.Builder(context, "alarm")
    .setSmallIcon(R.drawable.ic_launcher_foreground)
    .setContentTitle("Your Alarm is going off!!!")
    .setContentInfo("You set this alarm!")
    .setAutoCancel(true)
    .setDefaults(NotificationCompat.DEFAULT_ALL)
    .setPriority(NotificationCompat.PRIORITY_HIGH)
    .setContentIntent(p);
    NotificationManagerCompat notifications =
    NotificationManagerCompat.from(context);
    if (ActivityCompat.checkSelfPermission(context,
    android.Manifest.permission.POST_NOTIFICATIONS) != PackageManager.PERMISSION_GRANTED) {
        return;
    }
    else {
        notifications.notify(123, builder.build());
    }
    Ringtone ringtone =
    RingtoneManager.getRingtone(context.getApplicationContext(), alarmrt1); //Toast.makeText(context,
    ringtone.toString(),
    Toast.LENGTH_SHORT).show();
    intent.putExtra("RINGTONE_URI", alarmrt1);
    ringtone.play();
}
public static Uri getInstant() {
    return alarmrt1;
}
}
```

In.java:

```
package com.example.alarm;
import android.content.Intent;
import android.media.Ringtone;
import android.media.RingtoneManager;
import android.net.Uri;
import android.os.Bundle;
import android.widget.Toast;
//ce3c069
import androidx.appcompat.app.AppCompatActivity;
public class In extends AppCompatActivity {
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.display);
Intent intent = getIntent();
Uri ringtoneUri = intent.getParcelableExtra("RINGTONE_URI"); Ringtone ringtone =
RingtoneManager.getRingtone(In.this, ringtoneUri);
//Toast.makeText(In.this, ringtone.toString(),
Toast.LENGTH_SHORT).show();
if (ringtone.isPlaying()) {
//Toast.makeText(In.this, "yes", Toast.LENGTH_SHORT).show(); ringtone.stop();
}
else {
//Toast.makeText(In.this, "no", Toast.LENGTH_SHORT).show(); }
}
}
```

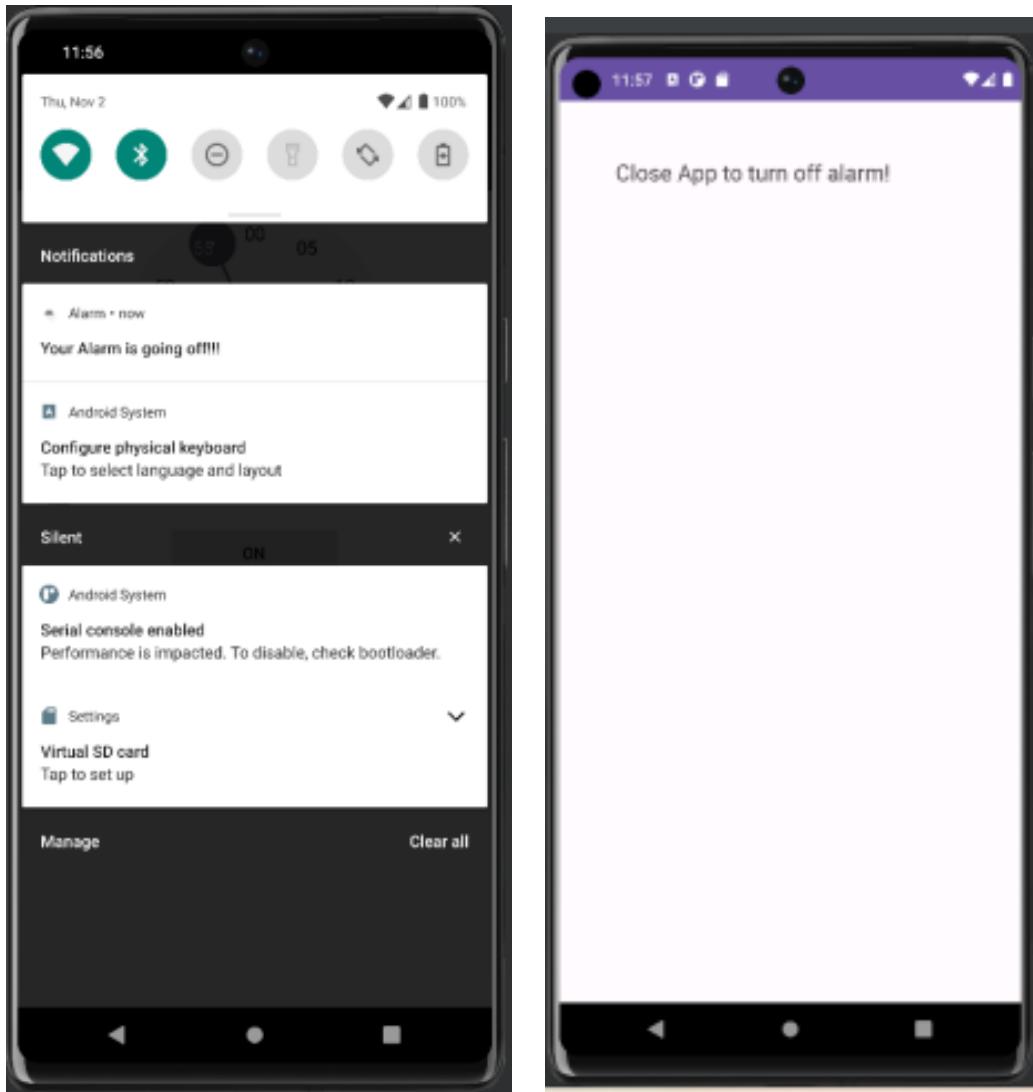
Androidmanifest.java:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools">
<uses-permission android:name="android.permission.POST_NOTIFICATIONS" /> <uses-permission
android:name="android.permission.WAKE_LOCK" /> <uses-permission
android:name="android.permission.VIBRATE" />
<application
```

```
    android:allowBackup="true"
    android:dataExtractionRules="@xml/data_extraction_rules"
    android:fullBackupContent="@xml/backup_rules"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/Theme.Alarm"
    tools:targetApi="31">
<activity
    android:name=".MainActivity"
    android:exported="true">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    
</activity>
<activity android:name=".In"
    android:exported="false"
/>
<receiver android:name=".AlarmReceiver"
/>
</application>
</manifest>
```

Output:





Best Practices

1. Keep the index form clean and uncluttered.
2. Use default ringtone as it is known by everyone.
3. Ensure the form works on various mobile devices.

Learning Outcomes

1. Learnt how to use UI Elements for Android App development.
2. Learnt how to use Phone shared storage.

Assignment 10

Menu Driven App

Objective:

Develop a Menu driven App that displays the option menu that contains names of different countries and when each of the country is clicked, the description about the country should be displayed.

Code:**Mainactivity.java:**

```
package com.example.menucountry;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

    TextView country, desc;

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

        country = findViewById(R.id.country);
        desc = findViewById(R.id.desc);
    }

    public boolean onCreateOptionsMenu(Menu menu) {
        MenuInflater inflater = getMenuInflater();
        inflater.inflate(R.menu.country, menu);
        return true;
    }

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

```
switch (id){  
    case R.id.ind:  
        // Toast.makeText(getApplicationContext(),"Item 1 Selected",Toast.LENGTH_LONG).show();  
        country.setText("India");  
        desc.setText(R.string.india);  
        return true;  
    case R.id.bah:  
        // Toast.makeText(getApplicationContext(),"Item 2 Selected",Toast.LENGTH_LONG).show();  
        country.setText("Bahrain");  
        desc.setText(R.string.bahrain);  
        return true;  
  
    case R.id.pak:  
        // Toast.makeText(getApplicationContext(),"Item 3 Selected",Toast.LENGTH_LONG).show();  
        country.setText("Pak");  
        desc.setText(R.string.pakistan);  
        return true;  
    case R.id.is:  
        // Toast.makeText(getApplicationContext(),"Item 3 Selected",Toast.LENGTH_LONG).show();  
        country.setText("Israel");  
        desc.setText(R.string.israel);  
        return true;  
    case R.id.chn:  
        // Toast.makeText(getApplicationContext(),"Item 3 Selected",Toast.LENGTH_LONG).show();  
        country.setText("China");  
        desc.setText(R.string.china);  
        return true;  
    case R.id.can:  
        // Toast.makeText(getApplicationContext(),"Item 3 Selected",Toast.LENGTH_LONG).show();  
        country.setText("Canada");  
        desc.setText(R.string.canada);  
        return true;  
    case R.id.uk:  
        // Toast.makeText(getApplicationContext(),"Item 3 Selected",Toast.LENGTH_LONG).show();  
        country.setText("UK");  
        desc.setText(R.string.uk);  
        return true;  
    default:  
        return super.onOptionsItemSelected(item);
```

```
        }
    }
}
```

Activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    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"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/country"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text=""
        android:textColor="@color/black"
        android:layout_margin="48dp"
        android:padding="24dp"
        android:textSize="24dp"
        android:textStyle="bold"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:padding="24dp"
        android:layout_margin="24dp"
        android:id="@+id/desc"
        android:layout_width="320dp"
        android:layout_height="wrap_content"
        app:layout_constraintTop_toBottomOf="@+id/country"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" />
```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

Country.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:id="@+id/ind"
        android:title="India" />
    <item android:id="@+id/bah"
        android:title="Bahrain" />
    <item android:id="@+id/pak"
        android:title="Pakistan" />
    <item android:id="@+id/is"
        android:title="Israel" />
    <item android:id="@+id/can"
        android:title="Canada" />
    <item android:id="@+id/uk"
        android:title="UK" />
    <item android:id="@+id/chn"
        android:title="China" />
</menu>
```

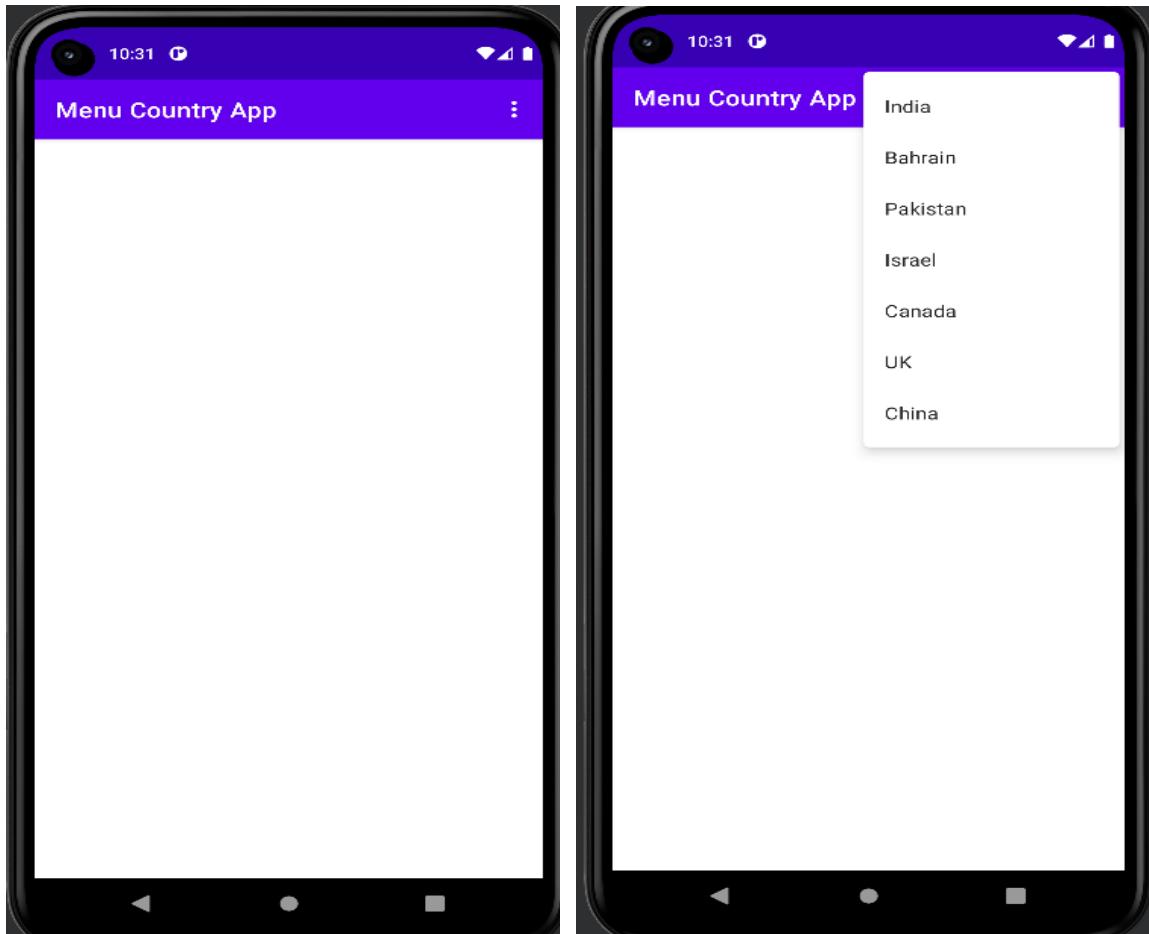
String.xml:

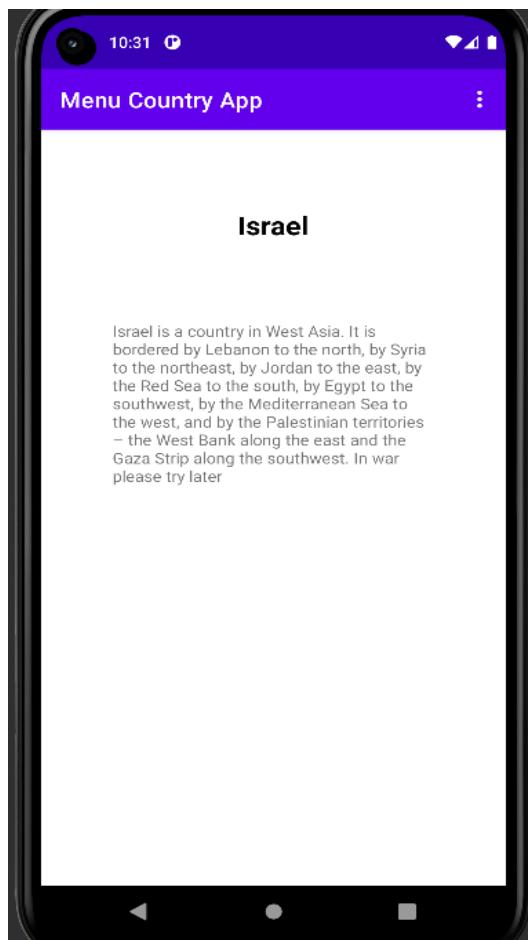
```
<resources>
    <string name="app_name">Menu Country App</string>
    <string name="india">India, officially the Republic of India (ISO: Bhārat Gaṇarājya), is a country in South Asia. It is the seventh-largest country by area; the most populous country as of June 2023; and from the time of its independence in 1947, the world's most populous democracy</string>
    <string name="pakistan">
        Pakistan (Urdu: پاکستان [pa:kista:n]),[d] officially the Islamic Republic of Pakistan (ISO: اسلامی جمہوریہ پاکستان, Islāmi jamhūriyāh pākistān), is a country in South Asia.
    </string>
    <string name="bahrain">
        Bahrain (/ba:'reɪn/bah-RAYN, /'reɪn/; Arabic: البحرين, romanized: al-Bahrayn, locally [æl bah're:n]), officially the Kingdom of Bahrain,[a] is an island country in West Asia.
    </string>
    <string name="israel">
        Israel is a country in West Asia. It is bordered by Lebanon to the north, by Syria to the northeast, by Jordan to the east, by the Red Sea to the south, by Egypt to the southwest, by the Mediterranean Sea to the west, and by the Palestinian territories – the West Bank along the east and the Gaza Strip along the southwest.
    </string>
```

```
In war please try later
</string>
<string name="china">
    China , officially the People's Republic of China (PRC),[k] is a country in East Asia.
</string>
<string name="canada">
    Canada is a country in North America. Its ten provinces and three territories extend from the Atlantic
Ocean to the Pacific Ocean and northward into the Arctic Ocean, making it the world's second-largest country
by total area,
</string>
<string name="uk">
The United Kingdom of Great Britain and Northern Ireland, commonly known as the United Kingdom (UK) or
Britain,[k][14] is an island country in Northwestern Europe, off the north-western coast of the continental
mainland.
</string>

</resources>
```

Output:





Learning Outcome :

Learnt to edit the menu on android studio
Learnt the steps to deploy a new resource file

Assignment 11

App to display a Web Page

Objective:

Develop an android application to display a static web page with contents that is constructed using formatting tags. Also should load the web page if present in the specified URL.

Code:**Mainactivity.java:**

```
package com.example.webpage;
import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.view.View;
import android.webkit.WebView;
import android.webkit.WebViewClient;
import android.widget.Button;
import android.widget.EditText;

public class MainActivity extends AppCompatActivity {

    private WebView webView;
    private EditText url;
    private Button getButton;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        webView = findViewById(R.id.webView);
        url = findViewById(R.id.url);
        getButton = findViewById(R.id.load);
        webView.getSettings().setJavaScriptEnabled(true);
    }
    // Load static HTML content
    String staticHtml = "<html>\n" +
            "<body>\n" +
            "\n" +
            "<h1 style=\"color:red; font-family:sans-serif\">This is a HTML" +
            " Site</h1>\n" +
            "\n" +
```

```

"<p style=\"color:red;\">A red paragraph.</p>\n" +
"\n" +
"<h2>An Unordered HTML List</h2>\n" +
"\n" +
"<ul>\n" +
" <li>MAD lab</li>\n" +
" <li>GML lab</li>\n" +
" <li>SNA lab</li>\n" +
"</ul>\n" +
"\n" +
"</body>\n" +
"</html>";

webView.loadData(staticHtml, "text/html", "UTF-8");
webView.setWebViewClient(new WebViewClient()
{
    @Override
    public boolean shouldOverrideUrlLoading(WebView view, String url)
    {
        //view.loadUrl(url);
        System.out.println("hello");
        return false;
    }
});
getButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        webView.loadUrl("https://" + url.getText().toString());
    }
});
}
}

```

Main_activity.xml:

```

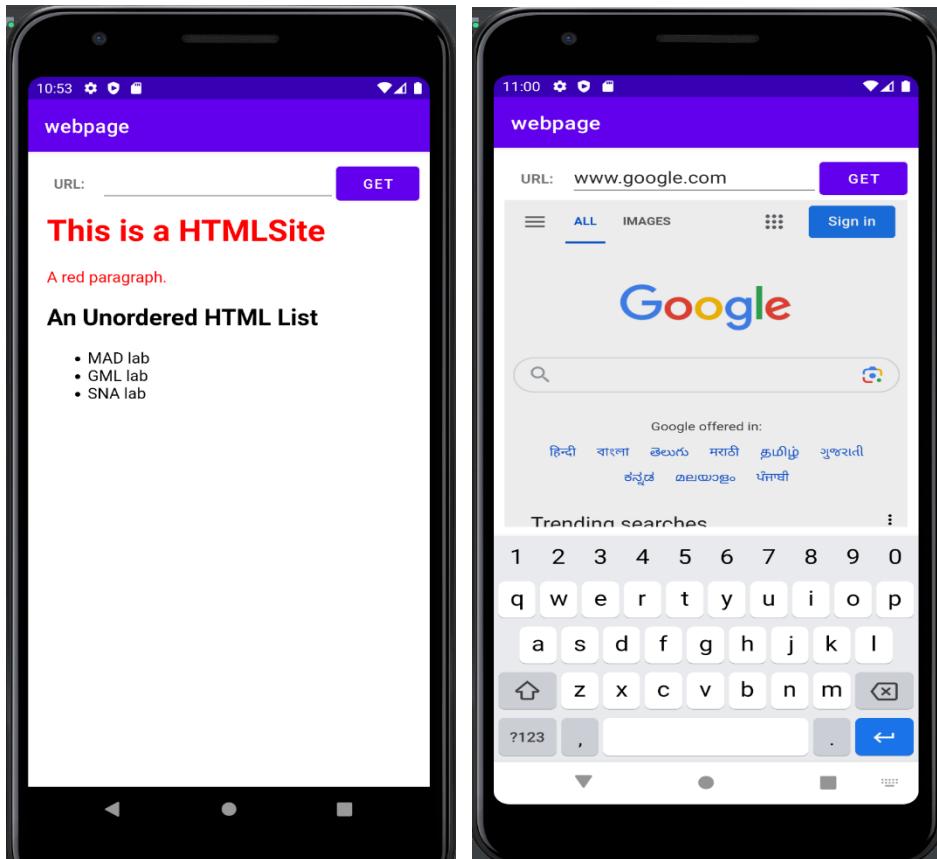
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"

```

```
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="10dp"
    tools:context=".MainActivity">
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">
<TextView
    android:id="@+id/textView"
    android:layout_width="60dp"
    android:layout_height="wrap_content"
    android:paddingBottom="10dp"
    android:paddingLeft="15dp"
    android:textStyle="bold"
    android:text="URL:" />
<EditText
    android:id="@+id/url"
    android:layout_width="237dp"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:ems="10"
    android:inputType="text" />
<Button
    android:id="@+id/load"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:text="GET" />
</LinearLayout>
<WebView
    android:id="@+id/webView"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
</WebView>
```

```
</LinearLayout>  
</LinearLayout>
```

Output:



Learning Outcome :

Learnt to access internet

Learnt how to parse URLs

MOBILE APPLICATION DEVELOPMENT LAB

MINI PROJECT REPORT

205001096

Selvendran MV

205001100

Shivcharan T

205001129

Vishal

CoNOTE

Problem Statement:

Our mini mobile application project focuses on developing a Collaborative Notes Management App with permissions. The main purpose of this app is to enable users to store and manage their notes efficiently and share it. Each user has a login(in-app or Google). Each note comprises a title and a description. The application provides a user-friendly interface for users to easily view all their stored notes. Furthermore, users have the flexibility to modify existing notes and delete them as necessary, offering a comprehensive and straightforward solution for note organization and management. Another feature is that users can share the app with others and give permissions to either read or write in the same note.

Functional Requirements:

Functional requirements include the features that the system provides to the user.

1. Login/Register Page:

- Users should be able to create new accounts or log in securely to the system.
- They can use Google sign-in or in-app sign-in.
- The registration process should include necessary user information and validation checks.
- Passwords should be securely stored using encryption techniques.
- Users should receive appropriate error messages for unsuccessful login attempts.

2. Homepage:

- Upon successful login, users are directed to a personalized homepage.
- The homepage contains the notes with small preview of the content
- Quick access links to essential features, such as creating new notes or accessing shared projects, should be available.

3. Notes Page:

- Users can create, view, and edit their own notes in a clean and intuitive interface.

4. Share Notes:

- Users should be able to share their notes with other users.
- Access permissions (view-only or edit) can be specified when sharing notes.
- Shared notes should appear in the recipient's interface in real-time or with minimal delay.

5. Edit Permissions (Read Only/Write):

- Note owners should have the ability to set permissions for collaborators on a per-user basis.
- Collaborators with "write" permissions can edit the content of shared notes.
- Collaborators with "read-only" permissions can view the notes but not make changes.

Non-Functional Requirements:

These are requirements that are not functional in nature. Specifically, these are the constraints that the system must work within.

- **UI/UX:**

The application must provide a clear user experience to enable seamless use.

- **Performance:**

- The system should provide responsive interfaces with low latency, even under peak usage conditions.

- Notes retrieval and editing should occur swiftly, providing a seamless user experience.
- **Scalability:**
 - The system architecture should be scalable to accommodate a growing number of users, notes, and concurrent activities.

Functionalities:

-Login/Register Module:

This module facilitates user authentication and registration processes. Users can securely log in with their credentials or create a new account by providing necessary information. The module includes password encryption for security and ensures a smooth and safe onboarding experience for users.

-Note Module:

The Note module allows users to create, view, and edit their notes. Users can employ rich text formatting, embed images and media, and organize their notes efficiently. The module also includes features such as search, sort, and categorization to enhance note organization. Revision history is maintained for tracking changes made to notes over time.

-Share Module:

This module enables users to share their notes with specific individuals or groups. It supports both internal and external collaboration by providing a mechanism to invite collaborators. Users can set access permissions (view-only or edit) when sharing notes, and shared notes are updated in real-time or with minimal delay in the recipients' interfaces.

-Addition Module:

The Addition Module allows users to add new content, notes, or elements to the system. This includes creating new notes, adding collaborators to shared notes, and incorporating new features or functionalities to enhance the user experience. It ensures a straightforward process for users to expand and enrich their work within the system.

-Deletion Module:

This module handles the removal of content or elements from the system. Users can delete individual notes, remove collaborators from shared notes, or delete unnecessary data. The Deletion Module should include confirmation mechanisms to prevent accidental data loss and ensure intentional removal.

-Permission Module:

The Permission Module allows note owners to manage access control for collaborators. Users can set permissions on a per-user basis, specifying whether collaborators have

read-only or write/edit access. The module ensures that collaboration remains secure and transparent by notifying users when permission settings are modified.

-Signout Module:

The Signout Module provides users with a secure way to log out of the system. Upon signing out, the user's session is terminated, enhancing security by preventing unauthorized access to the account. It ensures that users can confidently conclude their sessions and protect their data from unauthorized access

System Design:

1. Architecture:

The architecture allows for collaborative and real-time note creation and sharing while ensuring data consistency and security across multiple devices.

1. Client-Side (Mobile App):

- User Interface (UI): The mobile app's UI allows users to interact with the application, create, edit, and view notes.
- Authentication Module: Handles user authentication, login, and registration, often utilizing third-party authentication services like Google or Facebook.

- Note Management Module: Manages the creation, editing, and deletion of notes locally before syncing with the server.

2. Server-Side:

- Authentication Server: Manages user authentication, validating user credentials and generating access tokens for secure communication.

- Application Server: Handles business logic, note creation, modification, and deletion requests. Manages user permissions and collaborations.

3. Database Management:

- Real-time Database: Stores note data and user information in a real-time database, allowing instant updates and synchronization across multiple devices.

4. Third-Party Services:

- Authentication Providers: Integrates with third-party authentication services (e.g., Google, Facebook) for user convenience and security.

5. Security Measures:

- Access Controls: Implements role-based access controls to manage user permissions and restrict access to sensitive data.

2. Database Design:

The Firebase Realtime Database is a cloud-hosted NoSQL database that lets you store and sync data between your users in real time. Storage for Firebase lets you upload and share user generated content, such as images and video, which allows you to build rich media content into your apps. This is used to store images of users and recipes.

3. Collections:

1. User:

- User Name
- Email
- Password

2. Note:

- Note Id
- Note title
- Note content
- Note user permissions

3. Permission:

- Permission Type
- Users associated

Technologies used:

1. Android Studio (Java):

The application is built for the Android platform using Java within the Android Studio environment. Android Studio provides a comprehensive integrated

development environment (IDE) for Android app development, facilitating the creation of user-friendly and efficient Android applications.

2. Firebase (Authentication, Real-time Database, Storage):

Firebase is employed to enhance the app's functionality. It serves as a comprehensive backend solution, handling user authentication securely, providing a real-time database for dynamic data updates, and offering storage for multimedia content. Firebase simplifies the development process by offering reliable and scalable cloud services.

3. RecyclerView:

The app utilizes the RecyclerView component to dynamically display lists of notes. RecyclerView efficiently handles large data sets, optimizing the user interface by recycling views and improving performance. This ensures a smooth and responsive user experience when navigating through a collection of notes.

4. Custom Adapter Classes:

Custom adapter classes are implemented to seamlessly bridge the gap between the app's data source and the user interface. These adapters define how data is presented in predefined layouts/views, enhancing the flexibility and customization of the app's visual representation. Custom adapters contribute to a cohesive and visually appealing user experience.

5. Google Login Services:

The app integrates Google login services to streamline the authentication process. This feature enables users to effortlessly log in and sign up using their Google credentials. Leveraging Google login services enhances user convenience, potentially increasing user adoption and engagement while maintaining a secure authentication mechanism.

Output Screenshots:

9:19 AM | 0.4KB/s

Bluetooth 4G 31%



LOG IN USING GOOGLE

OR

Email

Password

SIGN IN

Don't have an account? Register

9:19 AM | 13.9KB/s

Bluetooth 4G 31%



Name

Email

Password

Verify Password

REGISTER

9:19 AM | 0.2KB/s

Bluetooth 4G 31%



Choose an account

to continue to CoNote



Vishal Sachan

sachanvishal.092@gmail.com



Vishal Sachan

vishal.29082003@gmail.com



vishal sachan

vishalsachan.555@gmail.com



Vishal Rajesh

vishal2010073@ssn.edu.in



Prakhar Sachan

sprakhar639@gmail.com



Prakhar Sachan

sprakhararya773@gmail.com



D+ Add another account

To continue, Google will share your name,
email address and profile picture with CoNote.
Before using this app, review its [privacy policy](#)
and [terms of service](#).

9:19 AM | 5.4KB/s 📈 ☀

Bluetooth 4G 31%

My Notes



About App 📜



Certainly! Here are the key points with emojis:

0. index starts from 0

1. Collaboration 🤝:



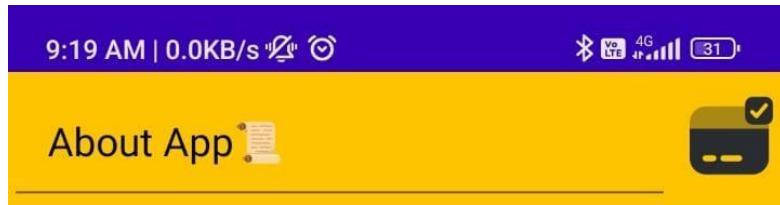
sem7

hello

my name is

dhoni





Certainly! Here are the key points with emojis:

0. index starts from 0

1. Collaboration 🤝:

- Create and share notes for seamless teamwork.

2. Permissions 🔒 :

- Set "View Only" or "Edit" permissions for collaborators.

3. Real-time Editing 🔄:

- Simultaneously collaborate on notes in real-time.

4. Version History 📚:

- Track changes and revert to previous versions.

5. User-Friendly Interface 💻:

- Intuitive design for easy adoption and efficient use.

9:20 AM | 0.0KB/s 📡 ⚡

Bluetooth 4G 31%

My Notes



About App 📜



Certainly! Here are the key points with emojis:

0. index starts from 0

1. Collaboration 🤝:

Vishal	▲	👑
vishal2010073@ssn.edu.in		
Permission- 🎉		
Tester 1	📝	🔴
test.user.1@email.com		
Permission- 🔵		
Tester 2	👁️	⚫
test.user.2@email.com		
Permission- 💯		
Shivcharan😎	👁️	⚫
shivcharan2010538@ssn.edu.in		
Permission- 💯		



Manage Users!

9:20 AM | 0.8KB/s

4G 30%

My Notes



About App 📜



Certainly! Here are the key points with emojis:

0. index starts from 0

1. Collaboration 🤝:

User	Email	Permission	Action
Vishal	vishal2010073@ssn.edu.in	Gold crown	Eye icon
Tester 1	test.user.1@email.com	Red pen	Trash icon
Tester 2	test.user.2@email.com	Blue eye	Red edit icon
Shivcharan	shivcharan2010538@ssn.edu.in	Yellow eye	Trash icon

9:20 AM | 0.0KB/s 📡 ⊗

Bluetooth 4G 30%

My Notes



About App 📜



Certainly! Here are the key points with emojis:

0. index starts from 0

1. Collaboration 🤝:



Email Address

Nick Name

ADD USER

By adding the user, you grant necessary permissions.

9:20 AM | 0.0KB/s



Title

Write here...

Best Practices:

- The code written to develop the application is concise and easily understandable.
- Each intent layout is provided with its own java file.
- The attributes of the database are clearly defined.
- The colour template used is in such a way that it becomes easier to use the application.
- Proper naming convention is used to define variables and functions.

Learning Outcomes:

- We learnt how to use android studio to create applications.
- We learnt how to create a database to store and display the information accordingly.
- We learnt how to use different concepts involved in android development to build the application.
- We learnt how to handle different layouts to provide a seamless experience.
- We learnt how to design the different pages using the underlying xml file.

Code:

MainActivity.java

```
package com.example.project_7;

import androidx.activity.result.ActivityResult;
import androidx.activity.result.ActivityResultCallback;
import androidx.activity.result.ActivityResultLauncher;
import androidx.activity.result.IntentSenderRequest;
import androidx.activity.result.contract.ActivityResultContract;
import androidx.activity.result.contract.ActivityResultContracts;

import androidx.appcompat.app.AppCompatActivity;

import android.annotation.TargetApi;
import android.app.Activity;
import android.content.Intent;
import android.content.IntentSender;
import android.content.SharedPreferences;
import android.net.Uri;
import android.os.Build;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.view.Window;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import com.example.project_7.model.UserModel;
import com.google.android.gms.auth.api.identity.BeginSignInRequest;
import com.google.android.gms.auth.api.identity.BeginSignInResult;
import com.google.android.gms.auth.api.identity.Identity;
import com.google.android.gms.auth.api.identity.SignInClient;
import com.google.android.gms.auth.api.identity.SignInCredential;
import com.google.android.gms.auth.api.signin.GoogleSignIn;
import com.google.android.gms.auth.api.signin.GoogleSignInAccount;
```

```
import com.google.android.gms.auth.api.signin.GoogleSignInClient;
import com.google.android.gms.auth.api.signin.GoogleSignInOptions;
import com.google.android.gms.common.api.ApiException;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import com.google.firebaseio.database.DataSnapshot;
import com.google.firebaseio.database.DatabaseReference;
import com.google.firebaseio.database.FirebaseDatabase;

import org.jetbrains.annotations.NotNull;

import java.io.Externalizable;
import java.io.Serializable;
import java.util.Map;
import java.util.Optional;
import java.util.concurrent.atomic.AtomicBoolean;

public class MainActivity extends AppCompatActivity {
    private static final int REQ_ONE_TAP = 2; // Can be any integer unique to the
                                             // Activity.
    private boolean showOneTapUI = true;
    private SignInClient oneTapClient;
    private BeginSignInRequest signUpRequest;

    GoogleSignInClient mGoogleSignInClient;
    private static int RC_SIGN_IN = 100;

    FirebaseDatabase database = FirebaseDatabase.getInstance();
    DatabaseReference usersRef = database.getReference("Users");

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

        // Configure sign-in to request the user's ID, email address, and basic
        // profile. ID and basic profile are included in DEFAULT_SIGN_IN.
        GoogleSignInOptions gso = new
        GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)
            .requestEmail()
            .build();

        // Build a GoogleSignInClient with the options specified by gso.
        mGoogleSignInClient = GoogleSignIn.getClient(this, gso);

        // Check for existing Google Sign In account, if the user is already signed in
```

```

// the GoogleSignInAccount will be non-null.

Button button = findViewById(R.id.button);
button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        signIn();
    }
});

TextView register = findViewById(R.id.registerTextView);
register.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent = new Intent(MainActivity.this, RegisterUser.class);
        startActivity(intent);
    }
});

```



```

Button signIn = findViewById(R.id.signInButton);
signIn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        EditText emailEditText = findViewById(R.id.emailEditText);
        EditText passwordField = findViewById(R.id.passwordEditText);
        String email = emailEditText.getText().toString();
        String password = passwordField.getText().toString();

        usersRef.get().addOnCompleteListener(new
        OnCompleteListener<DataSnapshot>() {
            @TargetApi(Build.VERSION_CODES.N)
            @Override
            public void onComplete(@NotNull Task<DataSnapshot> task) {
                if (!task.isSuccessful()) {
                    Toast.makeText(MainActivity.this, "Error Getting Data!",
                    Toast.LENGTH_SHORT).show();
                }
                else {
                    Map<String, Object> registeredUsers = (Map<String, Object>)
                    task.getResult().getValue();
                    if(registeredUsers != null) {
                        registeredUsers.forEach((key, userData) -> {
                            Optional.ofNullable((Map<String, Object>) userData)
                                .map(userMap -> {
                                    String emailId = (String) userMap.get("email");
                                    String pass = Optional.ofNullable((String)
                                        userMap.get("password")).orElse(null);

```

```

        if (emailId != null && emailId.equals(email)) {
            if (pass!= null && !pass.isEmpty() &&
pass.equals(password)) {
                SharedPreferences preferences =
getSharedPreferences("user_prefs", MODE_PRIVATE);
                SharedPreferences.Editor editor =
preferences.edit();
                editor.putString("email", email);
                editor.apply();

                Intent intent = new Intent(MainActivity.this,
MyNotes.class);
                startActivity(intent);
                Toast.makeText(MainActivity.this, "Signed
In!", Toast.LENGTH_SHORT).show();
                finish();
                return false;
            } else {
                Toast.makeText(MainActivity.this, "Wrong
Password!", Toast.LENGTH_SHORT).show();
                return false;
            }
        }
        return true; // Indicate that the iteration should
continue
    })
.orElseGet(() -> {
    // Handle the case where userData is not a Map
    Toast.makeText(MainActivity.this, "Invalid User
Data!", Toast.LENGTH_SHORT).show();
    return false; // Indicate that the iteration should
continue
});
}

} else {
    Toast.makeText(MainActivity.this, "Email Not Registered!",
Toast.LENGTH_SHORT).show();
}
}
}
}
});
```

```

private void signIn() {
    Intent signInIntent = mGoogleSignInClient.getSignInIntent();
    startActivityForResult(signInIntent, RC_SIGN_IN);
}

@Override
public void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    // Result returned from launching the Intent from
    GoogleSignInClient.getSignInIntent(...);
    if (requestCode == RC_SIGN_IN) {
        // The Task returned from this call is always completed, no need to attach
        // a listener.
        Task<GoogleSignInAccount> task =
        GoogleSignIn.getSignedInAccountFromIntent(data);
        handleSignInResult(task);
    }
}

private void handleSignInResult(Task<GoogleSignInAccount> completedTask)
{
    try {
        GoogleSignInAccount acct =
        completedTask.getResult(ApiException.class);
        if (acct != null) {
            SharedPreferences preferences = getSharedPreferences("user_prefs",
            MODE_PRIVATE);
            SharedPreferences.Editor editor = preferences.edit();
            editor.putString("email", acct.getEmail());
            editor.apply();
            registerUserFromGoogleSignIn(acct.getGivenName(), acct.getEmail());
//            Toast.makeText(MainActivity.this, "Signed In!",
//            Toast.LENGTH_SHORT).show();
            Intent intent = new Intent(getApplicationContext(), MyNotes.class);
            startActivity(intent);
            finish();
        } else {
            Toast.makeText(MainActivity.this, "Not Signed In!",
            Toast.LENGTH_SHORT).show();
        }
    } catch (ApiException e) {
        // The ApiException status code indicates the detailed failure reason.
        // Please refer to the GoogleSignInStatusCodes class reference for more
        information.
        Log.d("Sign In Error :", e.getMessage());
    }
}

```

```

private void registerUserFromGoogleSignIn(String name, String email) {
    usersRef.get().addOnCompleteListener(new
    OnCompleteListener<DataSnapshot>() {
        @TargetApi(Build.VERSION_CODES.N)
        @Override
        public void onComplete(@NotNull Task<DataSnapshot> task) {
            if (!task.isSuccessful()) {
                Toast.makeText(MainActivity.this, "Error Getting Data!",
                Toast.LENGTH_SHORT).show();
            }
            else {
                Map<String, Object> registeredUsers = (Map<String, Object>)
                task.getResult().getValue();
                AtomicBoolean alreadyRegistered = new AtomicBoolean(false);
                if(registeredUsers != null) {
                    registeredUsers.forEach((key, userData) -> {
                        String emailId = (String) ((Map<String, Object>)
                        userData).get("email");
                        if(emailId.equals(email)) {
                            alreadyRegistered.set(true);
                            //Toast.makeText(MainActivity.this, "Already Registered -
                            "+email, Toast.LENGTH_SHORT).show();
                            return;
                        }
                    });
                }
                if(!alreadyRegistered.get()){
                    registerUser(name, email);
                }
            }
        }
    });
}

private void registerUser(String name, String email) {
    UserModel user = new UserModel(name, email);
    usersRef.push().setValue(user);
}

//oneTapClient = Identity.getSignInClient(this);
//    signUpRequest = BeginSignInRequest.builder()
//    .setGoogleIdTokenRequestOptions(BeginSignInRequest.GoogleIdTokenReq
//    uestOptions.builder())
//    .setSupported(true)

```

```
//    // Your server's client ID, not your Android client ID.
//    .setServerClientId(getString(R.string.client_id))
//    // Show all accounts on the device.
//    .setFilterByAuthorizedAccounts(false)
//    .build()
//    .build();
//
//
//   ActivityResultLauncher<IntentSenderRequest> activityResultLauncher =
//    registerForActivityResult(new
//        ActivityResultContracts.StartIntentSenderForResult(),
//        new ActivityResultCallback<ActivityResult>() {
//    @Override
//    public void onActivityResult(ActivityResult result) {
//        if(result.getResultCode() == Activity.RESULT_OK) {
//            try {
//                SignInCredential credential =
//                    oneTapClient.getSignInCredentialFromIntent(result.getData());
//                String idToken = credential.getIdToken();
//                if (idToken != null) {
//                    // Got an ID token from Google. Use it to authenticate
//                    // with your backend.
//                    //
//                    Log.d("TAG", "Got ID token.");
//                    //
//                    String email = credential.getId();
//                    Toast.makeText(getApplicationContext(), "Email: "+email,
//                        Toast.LENGTH_SHORT).show();
//                }
//            } catch (ApiException e) {
//                // ...
//                //Caller has been temporarily blocked due to too many canceled sign-in
//                prompts.
//                // dial this in ur android device -> *#*#66382723#*#
//                e.printStackTrace();
//            }
//        }
//    });
//    Button button = findViewById(R.id.button);
//    button.setOnClickListener(new View.OnClickListener() {
//    @Override
//    public void onClick(View view) {
//        oneTapClient.beginSignIn(signUpRequest)
//            .addOnSuccessListener(MainActivity.this, new
```

```

    OnSuccessListener<BeginSignInResult>() {
    //@Override
    //public void onSuccess(BeginSignInResult result) {
    //    IntentSenderRequest intentSenderRequest =
    //    new
    IntentSenderRequest.Builder(result.getPendingIntent().getIntentSender()).buil
    d();
    //    activityResultLauncher.launch(intentSenderRequest);
    //    }
    //})
    //    .addOnFailureListener(MainActivity.this, new OnFailureListener() {
    //@Override
    //public void onFailure( Exception e) {
    //    // No Google Accounts found. Just continue presenting the signed-out UI.
    //    Log.d("TAG", e.getLocalizedMessage());
    //    }
    //});
    //    }
    //);
}

```

Activity.xml

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    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:background="@drawable/background"
    tools:context=".MainActivity">

    <androidx.cardview.widget.CardView
        android:id="@+id/signInCard"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="100dp"
        android:layout_marginStart="22dp"
        android:layout_marginEnd="22dp"
        app:cardElevation="8dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent">

```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center_horizontal"
    android:orientation="vertical">

    <androidx.cardview.widget.CardView
        xmlns:android="http://schemas.android.com/apk/res/android"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:cardElevation="4dp">

        <!-- ImageView to hold the image -->
        <ImageView
            android:id="@+id/cardImageview"
            android:layout_width="match_parent"
            android:layout_height="240dp"
            android:scaleType="centerCrop"
            android:src="@drawable/app_name"/>

    </androidx.cardview.widget.CardView>

    <Button
        android:id="@+id/button"
        style="@style/Widget.AppCompat.Button"
        android:layout_width="280dp"
        android:layout_height="54dp"
        android:layout_marginTop="16dp"
        android:elevation="@dimen/cardview_compat_inset_shadow"
        android:text="Log In Using Google"
        app:icon="@drawable/ic_google"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="parent" />

    <TextView
        android:id="@+id/termsTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:text="OR"
        android:textSize="18sp"
        android:textStyle="bold" />

    <androidx.cardview.widget.CardView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
```

```
        android:layout_margin="16dp"
        app:cardElevation="8dp">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:padding="16dp">

        <!-- Username Field -->
        <EditText
            android:id="@+id/emailEditText"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Email"
            android:inputType="text"
            android:minHeight="48dp" />

        <!-- Password Field -->
        <EditText
            android:id="@+id/passwordEditText"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Password"
            android:inputType="textPassword"
            android:minHeight="48dp" />

        <!-- Sign In Button -->
        <Button
            android:id="@+id/signInButton"
            style="@style/Widget.AppCompat.Button"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="16dp"
            android:text="Sign In" />

        <!-- Register TextView -->
        <TextView
            android:id="@+id/registerTextView"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="8dp"
            android:text="Don't have an account? Register"
            android:textColor="?attr/colorAccent"
            android:textStyle="italic" />

    </LinearLayout>
```

```
</androidx.cardview.widget.CardView>
</LinearLayout>
</androidx.cardview.widget.CardView>
</androidx.constraintlayout.widget.ConstraintLayout>
```