

Roll No. 42

Exam Seat No. _____

VIVEKANANDEDUCATION SOCIETY'S INSTITUTE OF TECHNOLOGY

HashuAdvani Memorial Complex, Collector's Colony, R. C. Marg,
Chembur, Mumbai – 400074. Contact No. 02261532532



Since 1962

CERTIFICATE

Certified that Mr./Miss Ronak Bipin Pathare

of SYMCA 2nd Shift

has satisfactorily completed a course of the necessary experiments in

Mobile Computing Lab under my supervision

in the Institute of Technology in the academic year 2020 - 2022.

Principal

Head of Department

Faculty In-charge

External Examiner



**V.E.S. Institute of Technology, Collector Colony,
Chembur, Mumbai
Department of M.C.A**

INDEX

Sr. No	Contents	Page Number	Marks	Sign
1	Android program using Various UI Components	2		
2	Android program using different layouts and views	12		
3	Android program based on Intents	23		
4	Android program for notifications and alert box	29		
5	Android program to perform CRUD operation using SQLite DB	36		
6	Android program using shared preference ,Internal and External storage	50		
7	Android program to work with graphics and animation	64		
8	Android program to work with google maps and location	74		
9	Android program to work with images and videos	82		
10	Android program based on Rest API	92		
11	Flutter program using layout ,widget and state management	110		
12	Flutter program to work with SQLite Database	115		
13	Flutter program based on RestAPI	125		
14	Mini project	131		

Practical No – 1

Aim: Android program using Various UI Components.

Theory:

Android RatingBar:

- Android RatingBar can be used to get the rating from the user. The Rating returns a floating-point number. It may be 2.0, 3.5, 4.0 etc.
- Android RatingBar displays the rating in stars. Android RatingBar is the subclass of AbsSeekBar class.
- The getRating() method of android RatingBar class returns the rating number.

Android CheckBox:

- Android CheckBox is a type of two state button either checked or unchecked.
- There can be a lot of usage of checkboxes. For example, it can be used to know the hobby of the user, activate/deactivate the specific action etc.
- Android CheckBox class is the subclass of CompoundButton class.

Methods of CheckBox class:

There are many inherited methods of View, TextView, and Button classes in the CheckBox class.

Some of them are as follows:

`public Boolean isChecked()` : Returns true if it is checked otherwise false.

`public void setChecked(boolean status)` : Changes the state of the CheckBox.

RadioButton:

- RadioButton is a two states button which is either checked or unchecked. If a single radio button is unchecked, we can click it to make checked radio button.
- Once a radio button is checked, it cannot be marked as unchecked by user.

RadioGroup:

- RadioButton is generally used with RadioGroup. RadioGroup contains several radio buttons, marking one radio button as checked makes all other radio buttons as unchecked.

A) Create an activity with all possible UI Components (Edit Text, Text view, Radio Button, Check Box, and Rating Bar). Get the values from the components and display them on the next activity

Code:

Activity_Main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <TextView
        android:id="@+id/formtv"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:text="Feedback Form"
        android:textSize="20sp"
        android:textStyle="bold"
        android:layout_margin="25sp"/>
    <TextView
        android:id="@+id/nametv"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/formtv"
        android:layout_marginLeft="50sp"
        android:layout_marginTop="23dp"
        android:text="Enter Name"
        android:textSize="15sp" />
    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/formtv"
        android:layout_marginLeft="144dp"
        android:layout_marginTop="11dp"
        android:layout_toRightOf="@+id/nametv"
        android:hint="Enter Name"
        android:id="@+id/nameet"/>
    <TextView
        android:id="@+id/gendertv"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/nametv"
        android:layout_marginLeft="50sp"
        android:layout_marginTop="58dp"
        android:text="Select Gender"
        android:textSize="15sp" />
    <RadioGroup
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/rggender"
        android:layout_below="@+id/nameet"
        android:layout_toRightOf="@+id/gendertv"
        android:layout_marginLeft="144dp">
        <RadioButton
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Male"
```

```
        android:id="@+id/rbmale" />
    <RadioButton
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Female"
        android:id="@+id/rbfemale" />
</RadioGroup>
<TextView
    android:id="@+id/subjectstv"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/gendertv"
    android:layout_marginLeft="50sp"
    android:layout_marginTop="90dp"
    android:text="Select Subjects"
    android:textSize="15sp" />
<CheckBox
    android:id="@+id/cbmc"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/rggender"
    android:layout_toRightOf="@+id/subjectstv"
    android:layout_marginLeft="144dp"
    android:text="MC" />
<CheckBox
    android:id="@+id/cbbc"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/cbmc"
    android:layout_toRightOf="@+id/subjectstv"
    android:layout_marginLeft="144dp"
    android:text="BC" />
<CheckBox
    android:id="@+id/cbgc"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/cbbc"
    android:layout_toRightOf="@+id/subjectstv"
    android:layout_marginLeft="144dp"
    android:text="GC" />
<TextView
    android:id="@+id/ratingtv"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/subjectstv"
    android:layout_marginLeft="50sp"
    android:layout_marginTop="90dp"
    android:text="Rating"
    android:textSize="15sp" />
<RatingBar
    android:id="@+id/ratebar"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/cbgc"
    android:layout_marginLeft="63dp"
    android:layout_marginTop="3dp"
    android:layout_toRightOf="@+id/ratingtv"
    android:numStars="5" />
<Button
    android:id="@+id/submit"
    android:layout_width="wrap_content"
```

```

        android:layout_height="wrap_content"
        android:layout_below="@+id/ratingtv"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="50dp"
        android:text="Submit" />
    </RelativeLayout>

```

MainActivity.java

```

package com.example.practical_1;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.*;
public class MainActivity extends AppCompatActivity {
    Button submitBtn;
    EditText etname;
    RadioGroup rg;
    RadioButton rb1,rb2;
    CheckBox cb1,cb2,cb3;
    RatingBar rbar;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        getActionBar().setTitle("Feedback Form");
        EditText etname= findViewById(R.id.nameet);
        rb1 = (RadioButton) findViewById(R.id.rbmale);
        rb2 = (RadioButton) findViewById(R.id.rbfemale);
        cb1 = findViewById(R.id.cbmc);
        cb2 = findViewById(R.id.cbbc);
        cb3 = findViewById(R.id.cbgc);
        RatingBar rbar = findViewById(R.id.ratebar);
        Button submitBtn= (Button)findViewById(R.id.submit);
        submitBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String name = etname.getText().toString();
                String rb=String.valueOf(rbar.getRating());
                Intent intent = new
                Intent(getApplicationContext(),MainActivity2.class);
                intent.putExtra("Name", name);
                if(rb1.isChecked()){
                    String male = rb1.getText().toString();
                    intent.putExtra("Gender", male);
                }
                if(rb2.isChecked()) {
                    String female = rb2.getText().toString();
                    intent.putExtra("Gender", female);
                }
                StringBuilder result=new StringBuilder();
                if(cb1.isChecked()) {
                    result.append("\n Mobile Computing");
                }
                if(cb2.isChecked()) {
                    result.append("\n Blockchain");
                }
                if(cb3.isChecked()) {
                    result.append("\n Green Computing");
                }
            }
        });
    }
}

```

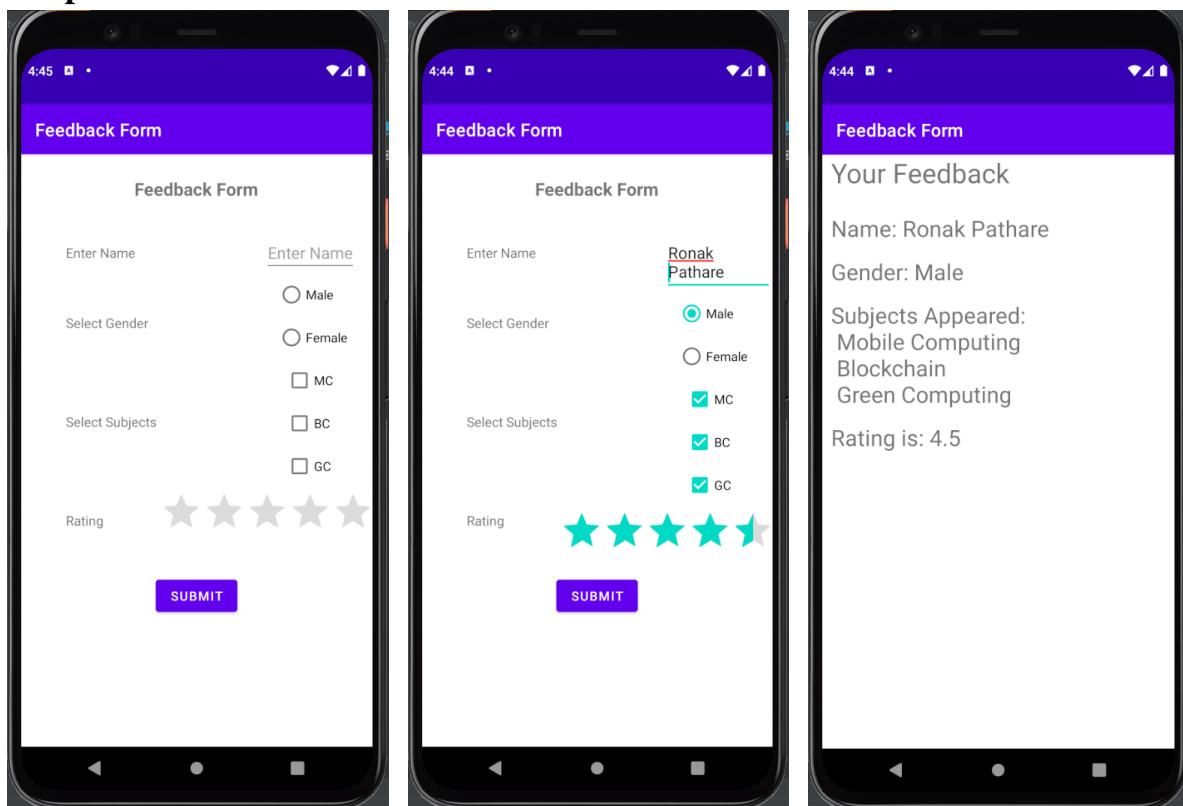
```
        intent.putExtra("Subjects Appeared", (CharSequence)
result);
        intent.putExtra("Rating", rb);
        startActivity(intent);
    }});}}
```

Activity_Main2.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=".MainActivity2">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:layout_margin="10dp">
        <TextView
            android:id="@+id/textView5"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Your Feedback"
            android:textSize="30dp"
            android:layout_marginBottom="25dp"/>
        <TextView
            android:id="@+id/textView6"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="TextView"
            android:textSize="25dp"
            android:layout_marginBottom="15dp"/>
        <TextView
            android:id="@+id/textView7"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="TextView"
            android:textSize="25dp"
            android:layout_marginBottom="15dp"/>
        <TextView
            android:id="@+id/textView8"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="TextView"
            android:textSize="25dp"
            android:layout_marginBottom="15dp"/>
        <TextView
            android:id="@+id/textView9"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="TextView"
            android:textSize="25dp"
            android:layout_marginBottom="15dp"/>
    </LinearLayout>
</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity2.java

```
package com.example.practical_1;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.widget.TextView;
public class MainActivity2 extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main2);
        getSupportActionBar().setTitle("Feedback Form");
        TextView t1,t2,t3,t4;
        t1 = (TextView)findViewById(R.id.textView6);
        t2 = (TextView)findViewById(R.id.textView7);
        t3 = (TextView)findViewById(R.id.textView8);
        t4 = (TextView)findViewById(R.id.textView9);
        Intent intent = getIntent();
        String name = intent.getStringExtra("Name");
        String gender = intent.getStringExtra("Gender");
        String subjects = intent.getStringExtra("Subjects Appeared");
        String rb = intent.getStringExtra("Rating");
        t1.setText("Name: " +name);
        t2.setText("Gender: "+gender);
        t3.setText("Subjects Appeared: "+subjects);
        t4.setText("Rating is: "+rb);
    }
}
```

Output:

B) Create a login form. For a successful login, display the welcome page, and in case of failure display an alert box indicating an error message and attempts made. Disable submit button after 3 wrong attempts and display the alert message indicating the same.

Code:

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"
    android:orientation="vertical"
    android:layout_marginVertical="15dp"
    android:layout_marginHorizontal="15dp"
    tools:context=".MainActivity">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">
        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Login"
            android:textSize="45dp"
            android:layout_marginTop="140dp"
            android:textColor="#282B2A">
        </TextView>
        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Enter credentials to Login"
            android:textSize="18dp"
            android:textColor="#686a69">
        </TextView>
        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Username"
            android:textSize="25dp"
            android:layout_marginTop="15dp"
            android:textColor="#686a69">
        </TextView>
        <EditText
            android:id="@+id/username"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="abcxyz@gmail.com">
        </EditText>
        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Password"
            android:textSize="25dp"
            android:layout_marginTop="15dp"
```

```

        android:textColor="#686a69">
    </TextView>
    <EditText
        android:id="@+id/password"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="textPassword"
        android:hint="At least 8 characters"
        android:layout_marginBottom="15dp">
    </EditText>
    <Button
        android:id="@+id/loginBtn"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Login">
    </Button>
</LinearLayout>
</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java

```

package com.example.practical3_2;

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

public class MainActivity extends AppCompatActivity {
    Button loginBtn;
    EditText username, password;
    int count = 0;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        getSupportActionBar().setTitle("Loginpage");
        username = findViewById(R.id.username);
        password = findViewById(R.id.password);
        loginBtn = findViewById(R.id.loginBtn);
        Button loginBtn= (Button)findViewById(R.id.loginBtn);
        loginBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String userName = username.getText().toString();
                String pass = password.getText().toString();
                if (userName.equals("ronakP@gmail.com") &&
pass.equals("ronak"))
                {
                    Toast.makeText(getApplicationContext(), "Login
Successful",Toast.LENGTH_SHORT).show();
                    Intent intent = new
Intent(getApplicationContext(),MainActivity2.class);
                    intent.putExtra("Username", userName);
                    startActivity(intent);
                }
                else{
                    Toast.makeText(getApplicationContext(), "Login Failed",
                }
            }
        });
    }
}
```

```

        Toast.LENGTH_SHORT).show();
        count++;
        if(count==3) {
            Toast.makeText(getApplicationContext(),"You have
made 3 unsuccessful attempts", Toast.LENGTH_SHORT).show();
            loginBtn.setEnabled(false);
        }
    }
});
```

Activity_Main2.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:layout_margin="15dp"
tools:context=".MainActivity2">

<TextView
    android:id="@+id/landingText"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:text=""
    android:textSize="30dp"/>

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

MainActivity2.java

```

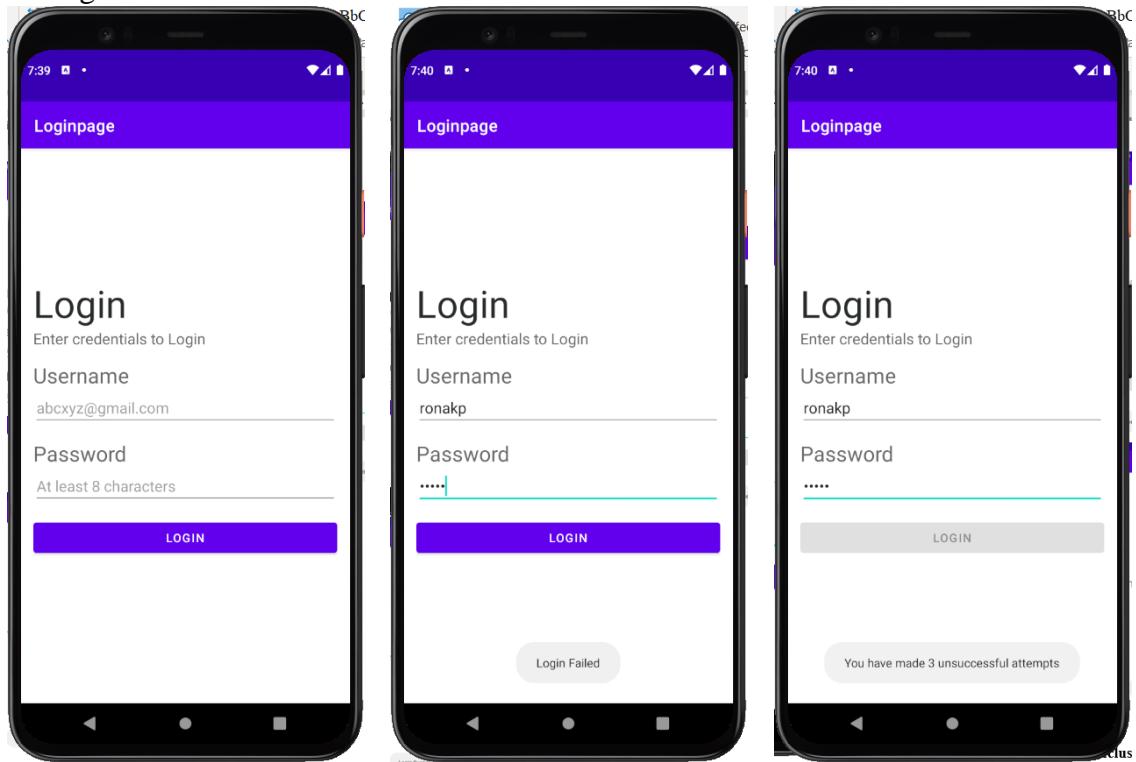
package com.example.practical3_2;

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

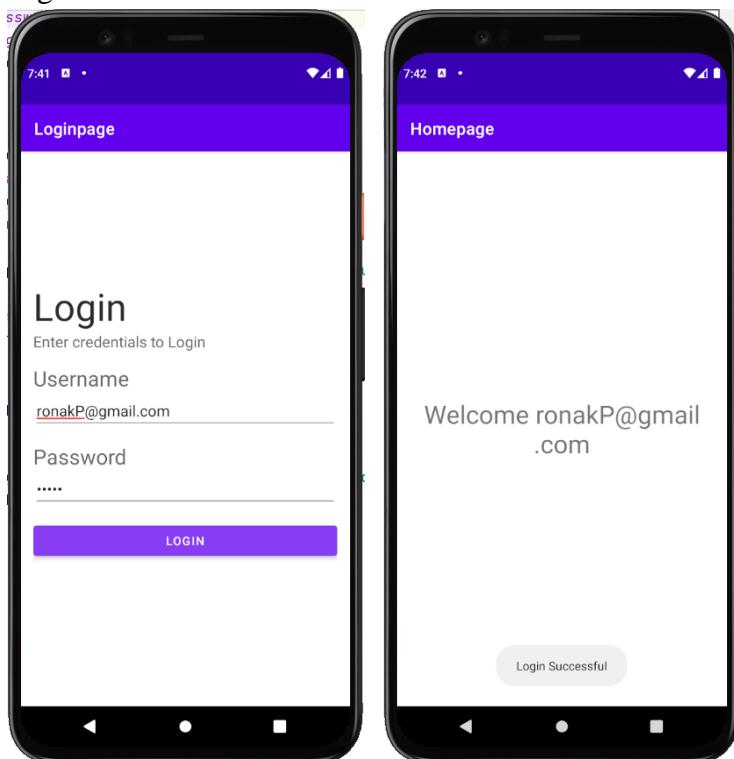
public class MainActivity2 extends AppCompatActivity {
    TextView text;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main2);
        getActionBar().setTitle("Homepage");
        text = (TextView) findViewById(R.id.landingText);
        Intent intent = getIntent();
        String username = intent.getStringExtra("Username");
        text.setText("Welcome " + username);
    }
    @Override
    public void onBackPressed() { }
```

Output:

Wrong credentials:



Right Credentials:

**Conclusion:**

We have developed different android applications using UI components.

Practical No – 2

Aim: Android program using different layouts and views

Theory:

1. Android Layout:

- Layout basically refers to the arrangement of elements on a page these elements are likely to be images, texts or styles. These are a part of Android Jetpack.
- They define the structure of android user interface in the app, like in an activity. All elements in the layout are built with the help of Views and View Groups.
- These layouts can have various widgets like buttons, labels, textboxes, and many others.
- Types of layouts in Android are:
 - 1) **Linear Layout:** We use this layout to place the elements in a linear manner. A Linear manner means one element per line. This layout creates various kinds of forms on Android. In this, arrangement of the elements is in a top to bottom manner.
 - 2) **Relative Layout:** This layout is for specifying the position of the elements in relation to the other elements that are present there. In the relative layout, alignment of the position of the elements to the parent container is possible.
 - 3) **Constraint Layout:** ConstraintLayout is a ViewGroup subclass, used to specify the position of layout constraints for every child View relative to other views present. A ConstraintLayout is similar to a RelativeLayout, but having more power.
 - 4) **Frame Layout:** FrameLayout is a ViewGroup subclass, used to specify the position of View elements it contains on the top of each other to display only a single View inside the FrameLayout.
 - 5) **Table Layout:** TableLayout is a ViewGroup subclass, used to display the child View elements in rows and columns.
 - 6) **Gridview Layout:** GridView is a ViewGroup that is used to display a scrollable list of items in a grid view of rows and columns.
 - 7) **Absolute layout:** An Absolute Layout lets you specify exact locations of its children. Absolute layouts are less flexible and harder to maintain than other types of layouts without absolute positioning

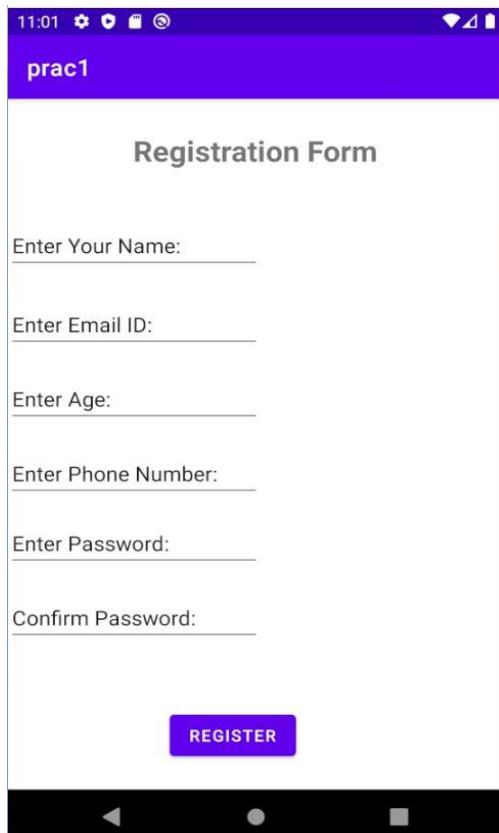
2. Android views

- A View is a simple building block of a user interface. It is a small rectangular box that can be TextView, EditText, or even a button. It occupies the area on the screen in a rectangular area and is responsible for drawing and event handling. View is a superclass of all the graphical user interface components.
- Now you might be thinking what is the use of a View. So, the use of a view is to draw content on the screen of the user's Android device. A view can be easily implemented in an Application using the java code. Its creation is easier in the XML layout file of the project. Like, the project for hello world that we had made initially.
- Types of Android Views
 1. TextView
 2. EditText
 3. Button
 4. Image Button
 5. Date Picker
 6. RadioButton
 7. CheckBox buttons
 8. Image View

A) Design a registration form to show the working of different layout.**Code:****Registration.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/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        tools:ignore="MissingConstraints"
        android:text="Registration Form"
        android:textSize="25sp"
        android:textStyle="bold"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.497"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        android:layout_marginTop="28dp"/>
    <EditText
        android:id="@+id/editText2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="44dp"
        android:ems="10"
        android:inputType="textPersonName"
        android:text="Enter Your Name:"
        app:layout_constraintTop_toBottomOf="@+id/textView1"
        tools:layout_editor_absoluteX="31dp"
        tools:ignore="MissingConstraints" />
    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginLeft="8dp"
        android:layout_marginTop="56dp"
        android:text="Register"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.422"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/editText11" />
    <EditText
        android:id="@+id/editText7"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="24dp"
        android:ems="10"
        android:inputType="textPersonName"
        android:text="Enter Email ID:"
        app:layout_constraintTop_toBottomOf="@+id/editText2"
        tools:layout_editor_absoluteX="31dp"
```

```
        tools:ignore="MissingConstraints" />
<EditText
    android:id="@+id/editText8"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:ems="10"
    android:inputType="textPersonName"
    android:text="Enter Age:"
    app:layout_constraintTop_toBottomOf="@+id/editText7"
    tools:layout_editor_absoluteX="31dp"
    tools:ignore="MissingConstraints" />
<EditText
    android:id="@+id/editText9"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:ems="10"
    android:inputType="textPersonName"
    android:text="Enter Phone Number:"
    app:layout_constraintTop_toBottomOf="@+id/editText8"
    tools:layout_editor_absoluteX="31dp"
    tools:ignore="MissingConstraints" />
<EditText
    android:id="@+id/editText10"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:ems="10"
    android:inputType="textPersonName"
    android:text="Enter Password:"
    app:layout_constraintTop_toBottomOf="@+id/editText9"
    tools:layout_editor_absoluteX="31dp"
    tools:ignore="MissingConstraints" />
<EditText
    android:id="@+id/editText11"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:ems="10"
    android:inputType="textPersonName"
    android:text="Confirm Password:"
    app:layout_constraintTop_toBottomOf="@+id/editText10"
    tools:layout_editor_absoluteX="31dp"
    tools:ignore="MissingConstraints" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Output:

B) Create an application to design simple calculator to perform addition, subtraction, multiplication and division. Show message for divide by zero error using Toast.

Code:**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">
    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        tools:ignore="MissingConstraints">
        <TextView
            android:id="@+id/calctv"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_centerHorizontal="true"
            android:layout_marginTop="20sp"
            android:text="Calculator"
```

```
    android:textSize="25sp"
    android:textStyle="bold">></TextView>
<EditText
    android:id="@+id/num1et"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/calctv"
    android:layout_marginLeft="50sp"
    android:layout_marginTop="60sp"
    android:hint="Enter 1st number"
    tools:ignore="MissingConstraints"
    />
<EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:hint="Enter 2nd number"
    android:layout_marginTop="60sp"
    android:layout_below="@id/calctv"
    android:layout_toRightOf="@+id/num1et"
    tools:ignore="MissingConstraints"
    android:id="@+id/num2et"/>
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Add"
    android:layout_below="@+id/num1et"
    android:layout_marginTop="10sp"
    android:layout_marginLeft="50sp"
    android:id="@+id/btnadd"/>
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Sub"
    android:layout_below="@+id/num2et"
    android:layout_toRightOf="@+id/btnadd"
    android:layout_marginTop="10sp"
    android:layout_marginLeft="120sp"
    android:id="@+id/btnsub"/>
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Mul"
    android:layout_below="@+id/btnadd"
    android:layout_marginTop="10sp"
    android:layout_marginLeft="50sp"
    android:id="@+id/btnmul"/>
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Div"
    android:layout_below="@+id/btnsub"
    android:layout_toRightOf="@+id/btnadd"
    android:layout_marginTop="10sp"
    android:layout_marginLeft="120sp"
    android:id="@+id/btndiv"/>
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
```

```

        android:layout_below="@+id/btnmul"
        android:text="Result"
        android:layout_marginTop="20sp"
        android:textSize="20sp"
        android:layout_centerHorizontal="true"
        android:id="@+id/resulttv"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/resulttv"
        android:text=""
        android:layout_marginTop="20sp"
        android:textSize="20sp"
        android:layout_centerHorizontal="true"
        android:id="@+id/displaytv"/>
</RelativeLayout>
</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java

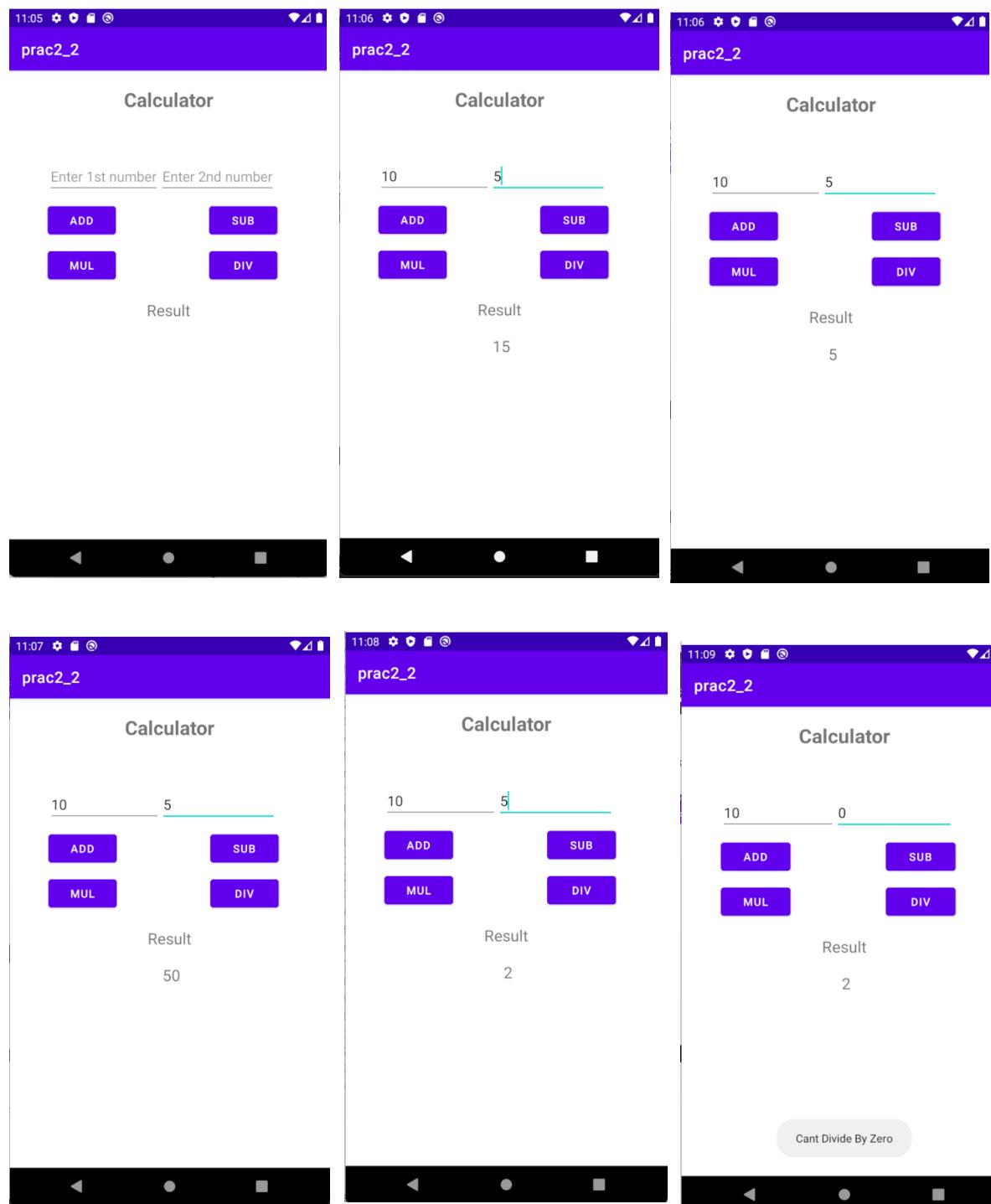
```

package com.example.prac2_2;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
    Button add,sub,mul,div;
    EditText n1,n2;
    TextView disptv;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        add=findViewById(R.id.btnadd);
        sub=findViewById(R.id.btnsub);
        mul=findViewById(R.id.btnmul);
        div=findViewById(R.id.btndiv);
        n1=findViewById(R.id.num1et);
        n2=findViewById(R.id.num2et);
        disptv=findViewById(R.id.displaytv);
        add.setOnClickListener(new View.OnClickListener()
        {
            @Override
            public void onClick(View view)
            {
                int n3=0;
                String
                num1=n1.getText().toString();
                String
                num2=n2.getText().toString();
                n3=Integer.parseInt(num1)+Integer.parseInt(num2);
                disptv.setText(" "+n3);
            }
        });
    }
}
```

```
});  
  
    sub.setOnClickListener(new View.OnClickListener() {  
        @Override  
        public void onClick(View view)  
        { int n4=0;  
        String  
        num1=n1.getText().toString();  
        String  
        num2=n2.getText().toString();  
        n4=Integer.parseInt(num1)-  
                Integer.parseInt(num2  
        ); dispv.setText(" "+n4);  
    }  
});  
  
mul.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        int n4=0;  
        String num1=n1.getText().toString();  
        String num2=n2.getText().toString();  
        n4=Integer.parseInt(num1)*Integer.parseInt(num2);  
        dispv.setText(" "+n4);  
    }  
});  
  
div.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        int n4=0;  
        String num1 = n1.getText().toString();  
        String num2 = n2.getText().toString();  
        if(Integer.parseInt(num2)==0){  
            Toast.makeText(MainActivity.this,"Cant Divide By Zero",  
Toast.LENGTH_LONG).show();  
        }  
        else {  
            n4 = Integer.parseInt(num1) / Integer.parseInt(num2);  
            dispv.setText(" " + n4);  
        }  
    }  
});  
});  
}  
}
```

Output:

C) Create an application for Unit Conversion.**Code:****Activity_main.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity"
    android:paddingTop="40sp">
    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="51dp"
        android:layout_x="14dp"
        android:layout_y="95dp"
        android:text="Temperature Converter"
        android:textSize="30dp" />
    <EditText
        android:id="@+id/editTextNumberDecimal2"
        android:layout_width="198dp"
        android:layout_height="wrap_content"
        android:layout_x="40dp"
        android:layout_y="177dp"
        android:ems="10"
        android:inputType="numberDecimal"
        android:textSize="25dp" />
    <RadioGroup
        android:id="@+id/rg"
        android:layout_width="286dp"
        android:layout_height="116dp"
        android:layout_x="38dp"
        android:layout_y="271dp">
        <RadioButton
            android:id="@+id/ctf"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text=" Celcius to Fahrenheit"
            android:textSize="25dp" />
        <RadioButton
            android:id="@+id/ftc"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Fahrenheit to Celcius "
            android:textSize="25dp" />
    </RadioGroup>
    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_x="37dp"
        android:layout_y="533dp"
        android:text="TextView"
        android:textSize="25dp"
```

```

        android:paddingBottom="20sp" />
<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="66dp"
    android:layout_x="101dp"
    android:layout_y="416dp"
    android:text="Convert"
    android:textSize="25dp"
    android:onClick="onClickMethod" />
</LinearLayout>
```

MainActivity.java

```

package com.example.prac2_2;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.TextView;
public class MainActivity extends
    AppCompatActivity { Button btn;
    RadioGroup rg;
    RadioButton
    rb1,rb2;
    EditText et;
    TextView tv;
    @Override
    protected void onCreate(Bundle
        savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        rg = (RadioGroup) findViewById(R.id.rg);
    }
    public void onClickMethod(View v){
        rb1 = (RadioButton) findViewById(R.id.ctf); rb2 = (RadioButton)
        findViewById(R.id.ftc);
        et = (EditText)
        findViewById(R.id.editTextNumberDecimal2); double val =
        Double.parseDouble(String.valueOf(et.getText())); tv =
        (TextView) findViewById(R.id.textView2);
        if(rb1.isChecked()){
            double res = (val*9)/5+32;
            tv.setText("Farenheit: "+res);
        }
        else if(rb2.isChecked()){
            double res = (val-32)*5/9;
            tv.setText("Celcius: "+res);
        } }}
```

Output:

Temperature Converter

45

- Celcius to Fahrenheit
 Fahrenheit to Celcius

Farenheit: 113.0

CONVERT

Temperature Converter

127

- Celcius to Fahrenheit
 Fahrenheit to Celcius

Celcius: 52.77777777777778

CONVERT**Conclusion:**

We have successfully implemented different types of layouts and Views.

Practical No – 3

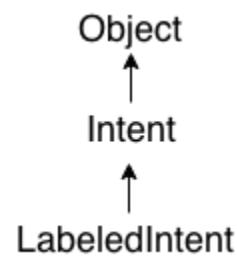
Aim: Android program based on Intents.

Theory:

1. Android Intent

- **Android Intent** is the *message* that is passed between components such as activities, content providers, broadcast receivers, services etc.
- It is generally used with `startActivity()` method to invoke activity, broadcast receivers etc.
- The **dictionary meaning** of intent is *intention or purpose*. So, it can be described as the intention to do action.
- The `LabeledIntent` is the subclass of `android.content.Intent` class.
- Android intents are mainly used to:

- Start the service
- Launch an activity
- Display a web page
- Display a list of contacts
- Broadcast a message
- Dial a phone call etc.



• Types of Android Intents

There are two types of intents in android: implicit and explicit.

1) Implicit Intent

Implicit Intent doesn't specify the component. In such case, intent provides information of available components provided by the system that is to be invoked.

For example, you may write the following code to view the webpage.

```

Intent intent=new Intent(Intent.ACTION_VIEW);
intent.setData(Uri.parse("http://www.javatpoint.com"));
startActivity(intent);
  
```

2) Explicit Intent

Explicit Intent specifies the component. In such case, intent provides the external class to be invoked.

For example

```

Intent i = new Intent(getApplicationContext(), ActivityTwo.class);
startActivity(i);
  
```

A) Create an application of implicit intent that displays a web page.

Code:

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.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="example.javatpoint.com.implicitintent.MainActivity">
    <EditText
        android:id="@+id/editText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="60dp"
        android:ems="10"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.575"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginRight="8dp"
        android:layout_marginLeft="156dp"
        android:layout_marginTop="172dp"
        android:text="Visit"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/editText" />
</android.support.constraint.ConstraintLayout>
```

MainActivity.java

```
package com.example.prac3_1;
import android.content.Intent;
import android.net.Uri;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

public class MainActivity extends AppCompatActivity {

    Button button;
    EditText editText;

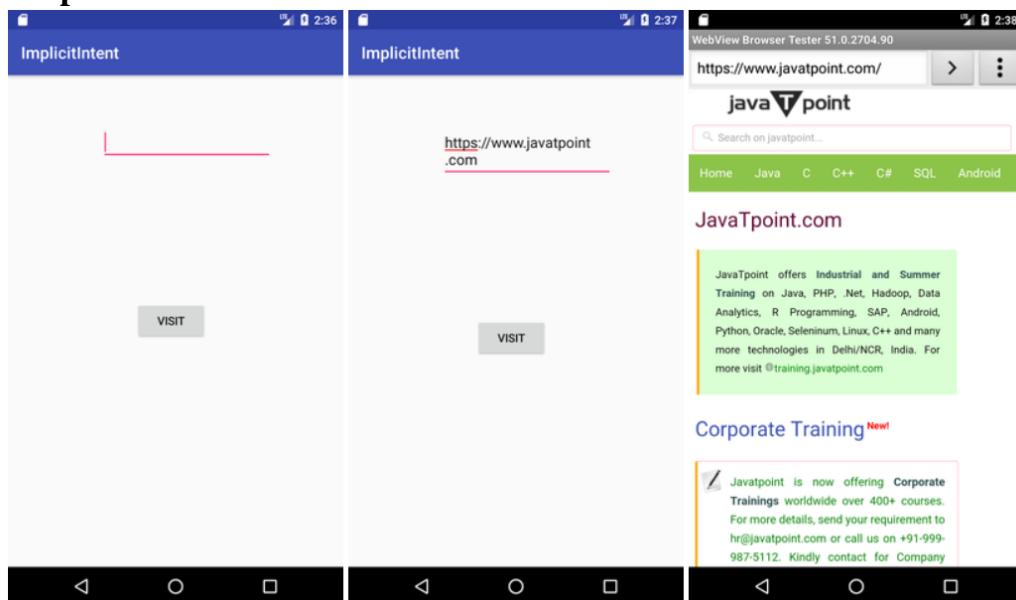
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
```

```

button = findViewById(R.id.button);
editText = findViewById(R.id.editText);

button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        String url=editText.getText().toString();
        Intent intent=new Intent(Intent.ACTION_VIEW, Uri.parse(u
rl));
        startActivity(intent);
    }
});
}
}

```

Output:

B) Create an application of explicit intent that calls one activity from another and vice versa.

Code:**ActivityOne_main.xml**

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="http://schem
as.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="example.javatpoint.com.explicitintent.FirstActivity"
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:text="First Activity"
        app:layout_constraintBottom_toBottomOf="parent"

```

```

        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.454"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.06" />
    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="392dp"
        android:onClick="callSecondActivity"
        android:text="Call second activity"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</android.support.constraint.ConstraintLayout>

```

MainActivityOne.java

```

package example.prac3_2;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;

public class FirstActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_first);
    }
    public void callSecondActivity(View view){
        Intent i = new Intent(getApplicationContext(), SecondActivity.class);
        i.putExtra("Value1", "Android By Ronak ");
        i.putExtra("Value2", "Simple Tutorial");
        // Set the request code to any code you like, you can identify this
        // callback via this code
        startActivityForResult(i);
    }
}

```

ActivityTwo_main.xml

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.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="example.javatpoint.com.explicitintent.SecondActivity">

```

```

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:text="Second Activity"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.454"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.06" />

<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="392dp"
    android:onClick="callFirstActivity"
    android:text="Call first activity"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
</android.support.constraint.ConstraintLayout>

```

MainActivityTwo.java

```

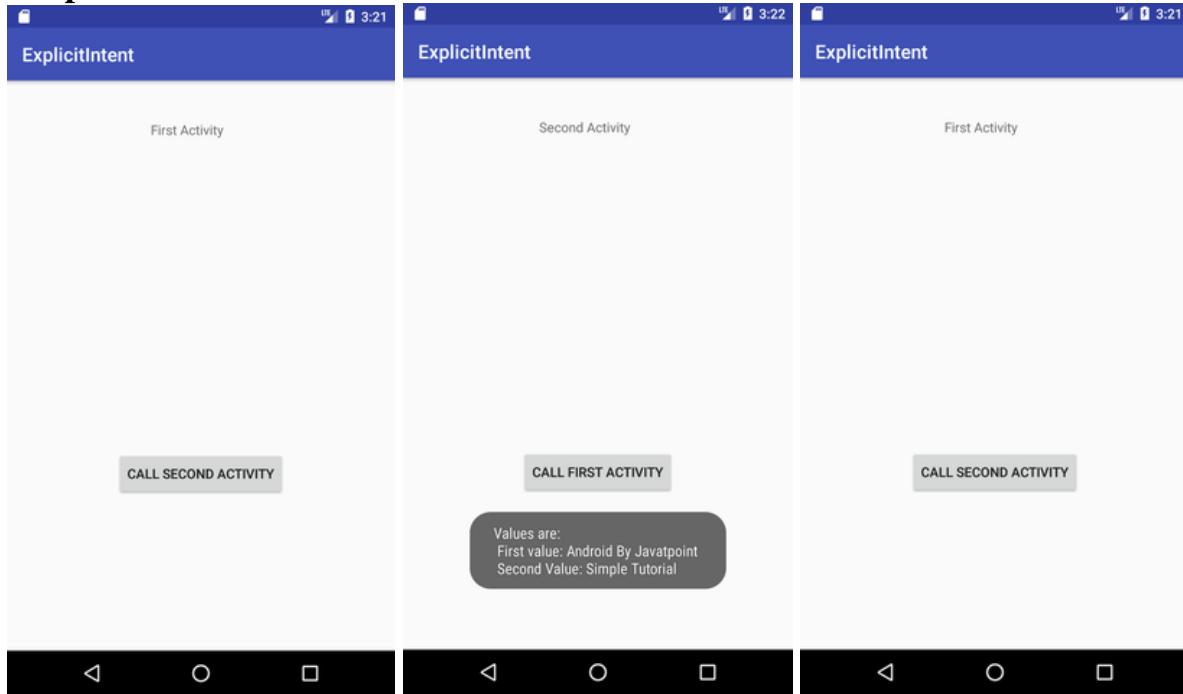
package example.javatpoint.com.explicitintent;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Toast;

public class SecondActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_second);
        Bundle extras = getIntent().getExtras();
        String value1 = extras.getString("Value1");
        String value2 = extras.getString("Value2");
        Toast.makeText(getApplicationContext(), "Values are:\n First value: "+value1+
                "\n Second Value: "+value2, Toast.LENGTH_LONG).show();
    }
    public void callFirstActivity(View view) {
        Intent i = new Intent(getApplicationContext(), FirstActivity.class);
        startActivity(i);
    }
}

```

Output:**Conclusion:**

We have successfully implemented android program based on intents.

Practical No – 4

Aim: Android program for notifications and alert box.

Theory:

Android Notification

- Android Notification provides short, timely information about the action happened in the application, even it is not running. The notification displays the icon, title and some amount of the content text.
- The properties of Android notification are set using **NotificationCompat.Builder** object. Some of the notification properties are mention below:
 - **setSmallIcon()**: It sets the icon of notification.
 - **setContentTitle()**: It is used to set the title of notification.
 - **setContentText()**: It is used to set the text message.
 - **setAutoCancel()**: It sets the cancelable property of notification.
 - **setPriority()**: It sets the priority of notification.

Alert Box:

- Alert Dialog shows the Alert message and gives the answer in the form of yes or no. Alert Dialog displays the message to warn you and then according to your response the next step is processed.
- Android Alert Dialog is built with the use of three **fields: Title, Message area, Action Button.**
- Alert Dialog code has three **methods**:
 - `public AlertDialog.Builder setTitle(CharSequence)`: This method is used to set the title of AlertDialog.
 - `public AlertDialog.Builder setMessage(CharSequence)`: This method is used to set the message for AlertDialog.
 - `public AlertDialog.Builder setIcon(int)`: This method is used to set the icon over AlertDialog.

A) Create an application that sends a notification on click of an button and when clicked on that notification open another page.

Code:

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.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="example.javatpoint.com.androidnotification.MainActivity"
>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="ANDROID NOTIFICATION"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.091"
        android:textAppearance="@style/Base.TextAppearance.AppCompat.Medium"
/>
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/button"
        android:layout_marginBottom="112dp"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:text="Notify"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" />
</android.support.constraint.ConstraintLayout>
```

MainActivity.java

```
package example.javatpoint.com.androidnotification;

import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.Context;
import android.content.Intent;
import android.support.v4.app.NotificationCompat;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

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

```

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

private void addNotification() {
    NotificationCompat.Builder builder =
        new NotificationCompat.Builder(this)
            .setSmallIcon(R.drawable.messageicon) //set icon for notification
            .setContentTitle("Notifications Example") //set title of notification
            .setContentText("This is a notification message")//this is notification message
            .setAutoCancel(true) // makes auto cancel of notification
            .setPriority(NotificationCompat.PRIORITY_DEFAULT);
    //set priority of notification

    Intent notificationIntent = new Intent(this, NotificationView.class);
    notificationIntent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
    //notification message will get at NotificationView
    notificationIntent.putExtra("message", "This is a notification message");

    PendingIntent pendingIntent = PendingIntent.getActivity(this, 0, notificationIntent,
        PendingIntent.FLAG_UPDATE_CURRENT);
    builder.setContentIntent(pendingIntent);

    // Add as notification
    NotificationManager manager = (NotificationManager) getSystemService(Context.NOTIFICATION_SERVICE);
    manager.notify(0, builder.build());
}
}

```

Activity_Notification_View.xml

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.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="example.javatpoint.com.androidnotification.NotificationView">
    <TextView
        android:id="@+id/textView2"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:text="your detail of notification..."
        android:textAppearance="@style/Base.TextAppearance.AppCompat.Medium" />

```

```

<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.096"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView2"
    app:layout_constraintVertical_bias="0.206"
    android:textAppearance="@style/Base.TextAppearance.AppCompat.Med
ium"/>

</android.support.constraint.ConstraintLayout>

```

NotificationView.java

```

package example.javatpoint.com.androidnotification;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.TextView;
import android.widget.Toast;

public class NotificationView extends AppCompatActivity {
    TextView textView;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_notification_view);
        textView = findViewById(R.id.textView);
        //getting the notification message
        String message=getIntent().getStringExtra("message");
        textView.setText(message);
    }
}

```

strings.xml

```

<resources>
    <string name="app_name">AndroidNotification</string>
    <string name="notification_activity">NotificationView</string>
</resources>

```

AndroidManifest.xml

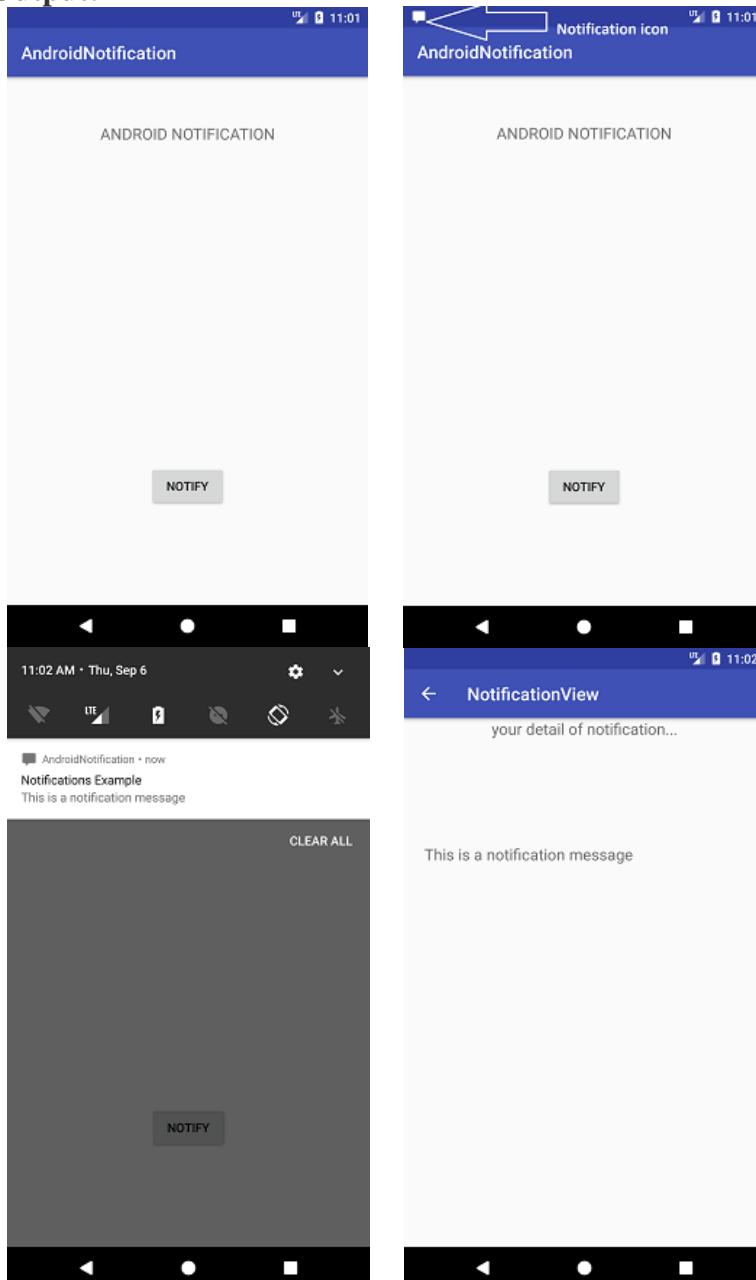
```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="example.javatpoint.com.androidnotification">
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">

```

```
<intent-filter>
    <action android:name="android.intent.action.MAIN" />
    <category android:name="android.intent.category.LAUNCHER"
    " />
</intent-filter>
</activity>
<activity android:name=".NotificationView"
    android:label="@string/notification_activity"
    android:parentActivityName=".MainActivity">
    <meta-data
        android:name="android.support.PARENT_ACTIVITY"
        android:value=".MainActivity"/>
</activity>
</application>

</manifest>
```

Output:

B) Create an application that displays an alert box on click of a button.**Code:****Activity_main.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.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="example.javatpoint.com.alertdialog.MainActivity">
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/button"
        android:text="Close app"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</android.support.constraint.ConstraintLayout>
```

Strings.xml

```
<resources>
    <string name="app_name">AlertDialog</string>
    <string name="dialog_message">Welcome to Alert Dialog</string>
    <string name="dialog_title">Javatpoint Alert Dialog</string>
</resources>
```

MainActivity.java

```
package example.javatpoint.com.alertdialog;
import android.content.DialogInterface;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.app.AlertDialog;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
    Button closeButton;
    AlertDialog.Builder builder;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        closeButton = (Button) findViewById(R.id.button);
        builder = new AlertDialog.Builder(this);
        closeButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

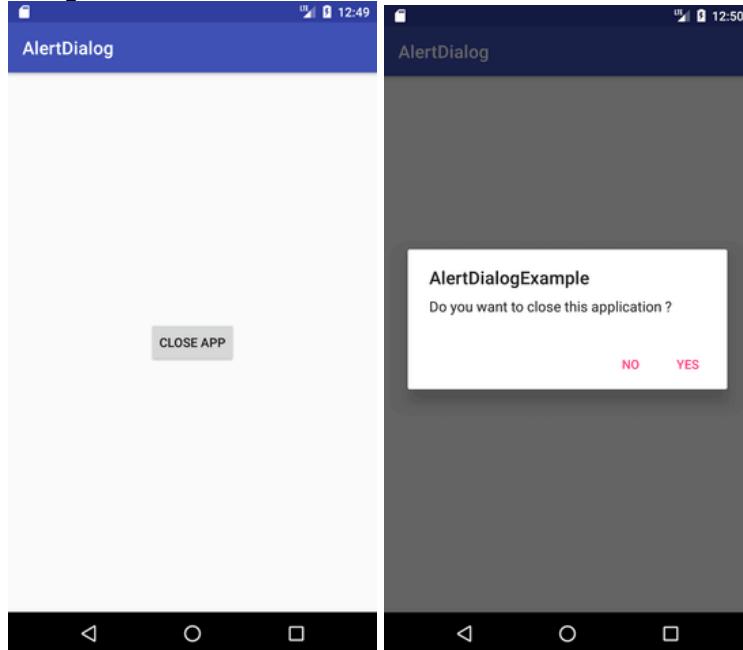
                //Uncomment the below code to Set the message and title
                //from the strings.xml file
                builder.setMessage(R.string.dialog_message).setTitle(R.string.dialog_title);

                //Setting message manually and performing action on butt
                on click
            }
        });
    }
}
```

```

        builder.setMessage("Do you want to close this application ?")
                .setCancelable(false)
                .setPositiveButton("Yes", new DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface dialog, int id) {
                        finish();
                        Toast.makeText(getApplicationContext(), "you choose yes action for alertbox",
                                Toast.LENGTH_SHORT).show();
                    }
                })
                .setNegativeButton("No", new DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface dialog, int id) {
                        // Action for 'NO' Button
                        dialog.cancel();
                        Toast.makeText(getApplicationContext(), "you choose no action for alertbox",
                                Toast.LENGTH_SHORT).show();
                    }
                });
        //Creating dialog box
        AlertDialog alert = builder.create();
        //Setting the title manually
        alert.setTitle("AlertDialogExample");
        alert.show();
    } });
}

```

Output:**Conclusion:**

We have successfully implemented android program based on notifications and alert box.

Practical 5

Aim: Android program to perform CRUD operation using SQLite DB

Theory:

What is SQLite?

SQLite is an SQL Database. I am assuming here that you are familiar with SQL databases. So in SQL database, we store data in tables. The tables are the structure of storing data consisting of rows and columns. We are not going in depth of what is an SQL database and how to work in SQL database. If you are going through this post, then you must know the **Basics of SQL**.

What is CRUD?

As the heading tells you here, we are going to learn the CRUD operation in SQLite Database. **CRUD** is nothing but an abbreviation for the basic operations that we perform in any database. And the operations are

- **Create**
- **Read**
- **Update**
- **Delete**

Android SQLite Database Example App Apk

- Before moving ahead on this tutorial if you want to know what we will be building, you can get the final apk of this tutorial from the link given below.
- **Android SQLite Database Example App Apk Download**

Android SQLite Database Example

- [Creating a new Android Studio Project](#)
- As always we will create a new Android Studio Project. For this example, I have a new project named **SQLiteCRUDExample**.
- Once your project is loaded, we can start working on it.

The Database Structure

- The first thing needed is the database structure. We create database structure according to the system. But here we are not building an application, and it is only an example demonstrating the use of SQLite Database. So for this, I will use the following table structure.

employees		
id	int	PK
name	varchar(200)	
department	varchar(200)	
joiningdate	datetime	
salary	double	

Database Structure

Now we have only a single table, but in real-world scenarios, you will have multiple tables with some complex relationships. Also, remember one thing whenever you create a table create a column named id with int as **PRIMARY KEY** and **AUTOINCREMENT**. (If you are confused don't worry we will see now how do we create tables in database using SQL).

SQL Queries

Now let's see how we can create the above table in our SQLite database.

Creating the Table –

```
CREATE TABLE employees (
    id INTEGER NOT NULL CONSTRAINT employees_pk PRIMARY KEY AUTOINCREMENT,
    name varchar(200) NOT NULL,
    department varchar(200) NOT NULL,
    joiningdate datetime NOT NULL,
    salary double NOT NULL
);
```

Creating a new record –

```
INSERT INTO employees
(name, department, joiningdate, salary)
VALUES
('Belal Khan', 'Technical', '2017-09-30 10:00:00', '40000');
```

Reading All Existing Records –

```
SELECT * FROM employees;
```

Reading Specific Record –

```
SELECT * FROM employees WHERE id = 1;
```

Updating a Record –

```
UPDATE employees
SET
name = 'Belal Haque',
department = 'Research and Development',
salary = '100000'
WHERE id = 1;
```

Deleting a Record –

```
DELETE FROM employees WHERE id = 1;
```

These are just some simple basics operations, but we can perform many tasks in our database. For this, you need to learn SQL in detail.

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"
    android:padding="10dp"
    tools:context=".MainActivity">
    <com.google.android.material.button.MaterialButton
        android:id="@+id/main_create_button"
        style="@style/Widget.MaterialComponents.Button.Icon"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="10dp"
        android:gravity="center"
        android:text="Create"
        android:textStyle="bold"
        android:visibility="visible"
        app:backgroundTint="#009806" />
    <RelativeLayout
        android:id="@+id/rel_lay1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/main_create_button"
        android:visibility="gone">
        <com.google.android.material.textfield.TextInputLayout
            android:id="@+id/main_name"
            style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="10dp"
            app:boxCornerRadiusBottomEnd="10dp"
            app:boxCornerRadiusBottomStart="10dp"
            app:boxCornerRadiusTopEnd="10dp"
            app:boxCornerRadiusTopStart="10dp">
            <com.google.android.material.textfield.TextInputEditText
                android:id="@+id/name"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:hint="Name*"
                android:inputType="text"
                android:textColor="@color/black" />
        </com.google.android.material.textfield.TextInputLayout>
        <com.google.android.material.textfield.TextInputLayout
            android:id="@+id/main_email"
            style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_below="@+id/main_name"
            android:layout_marginTop="10dp"
            app:boxCornerRadiusBottomEnd="10dp"
            app:boxCornerRadiusBottomStart="10dp"
            app:boxCornerRadiusTopEnd="10dp"
            app:boxCornerRadiusTopStart="10dp">
```

```
<com.google.android.material.textfield.TextInputEditText
    android:id="@+id/email"

        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Email ID*"
        android:inputType="textEmailAddress"
        android:textColor="@color/black" />
</com.google.android.material.textfield.TextInputLayout>
<com.google.android.material.textfield.TextInputLayout
    android:id="@+id/main_course"

style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/main_email"
    android:layout_marginTop="10dp"
    app:boxCornerRadiusBottomEnd="10dp"
    app:boxCornerRadiusBottomStart="10dp"
    app:boxCornerRadiusTopEnd="10dp"
    app:boxCornerRadiusTopStart="10dp">
<com.google.android.material.textfield.TextInputEditText
    android:id="@+id/course"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Course*"
    android:inputType="text"
    android:textColor="@color/black" />
</com.google.android.material.textfield.TextInputLayout>
<com.google.android.material.button.MaterialButton
    android:id="@+id/create_button"
    style="@style/Widget.MaterialComponents.Button.Icon"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"

        android:layout_below="@+id/main_course"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="10dp"
        android:gravity="center"
        android:text="Create"
        android:visibility="visible"
        android:textStyle="bold"
        app:backgroundTint="#009806" />
<com.google.android.material.button.MaterialButton
    android:id="@+id/update_button"
    style="@style/Widget.MaterialComponents.Button.Icon"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/main_course"
    android:visibility="gone"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="10dp"
    android:gravity="center"
    android:text="Update"
    android:textStyle="bold"
    app:backgroundTint="#009806" />
</RelativeLayout>
<androidx.recyclerview.widget.RecyclerView
    android:id="@+id/student_display_recycler_view"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
```

```
    android:layout_below="@+id/rel_lay1"
    android:layout_marginTop="10dp" />
</RelativeLayout>
```

student_card_view.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"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="5dp"
    android:background="@color/white">
    <androidx.cardview.widget.CardView
        android:id="@+id/row_card"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true"
        android:layout_margin="5dp"
        android:layout_alignParentEnd="true"
        app:cardBackgroundColor="@color/white"
        app:cardCornerRadius="10dp"
        app:cardElevation="8dp"
        app:contentPadding="5dp">
        <ImageView
            android:layout_width="wrap_content"
            android:id="@+id/more_rv"
            android:layout_margin="5dp"
            android:layout_height="wrap_content"
            android:layout_gravity="right"
            android:src="@drawable/ic_baseline_more_vert_24"
            />

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical">
            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:orientation="horizontal">
                <TextView
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:layout_margin="5dp"
                    android:text="ID:"
                    android:textColor="@color/black"
                    android:textSize="17sp"
                    android:textStyle="bold" />
                <TextView
                    android:id="@+id/id_rv"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:layout_margin="5dp"
                    android:text="id"
                    android:textColor="@color/black"
                    android:textSize="17sp" />
            </LinearLayout>
            <LinearLayout
                android:layout_width="match_parent"
```

```
        android:layout_height="wrap_content"
        android:orientation="horizontal">
    <TextView

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="5dp"
        android:text="Name:"
        android:textColor="@color/black"
        android:textSize="17sp"
        android:textStyle="bold" />
    <TextView
        android:id="@+id/name_rv"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="5dp"
        android:text="name"
        android:textColor="@color/black"
        android:textSize="17sp" />
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="5dp"
        android:text="Email ID:"
        android:textColor="@color/black"
        android:textSize="17sp"
        android:textStyle="bold" />
    <TextView
        android:id="@+id/email_rv"

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="5dp"
        android:text="email"
        android:textColor="@color/black"
        android:textSize="17sp" />
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="5dp"
        android:text="Course:"
        android:textColor="@color/black"
        android:textSize="17sp"
        android:textStyle="bold" />
    <TextView
        android:id="@+id/course_rv"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="5dp"
        android:text="course"
        android:textColor="@color/black"
```

```

        android:textSize="17sp" />
    </LinearLayout>
</LinearLayout>
</androidx.cardview.widget.CardView>
```

create menu folder and create menu_rv

menu_rv.xml

```

<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item
        android:id="@+id/update_menu"
        android:title="Update"
        />
    <item
        android:id="@+id/delete_menu"
        android:title="Delete"
        />
</menu>
```

SQLiteHelper.kt-

```

package com.karthik.crudsqlite
import android.content.ContentValues
import android.content.Context
import android.database.Cursor
import android.database.sqlite.SQLiteDatabase
import android.database.sqlite.SQLiteOpenHelper
class SQLiteHelper(context: Context) :
    SQLiteOpenHelper(context, DATABASE_NAME, null, DATABASE_VERSION) {
    companion object {
        private const val DATABASE_VERSION = 1
        private const val DATABASE_NAME = "student.db"
        private const val TABLE_STUDENT = "table_student"
        private const val ID = "id"
        private const val NAME = "name"
        private const val EMAIL = "email"

        private const val COURSE = "course"
    }
    override fun onCreate(db: SQLiteDatabase?) {
        val createTableStudent =
            ("CREATE TABLE $TABLE_STUDENT ($ID INTEGER PRIMARY KEY, $NAME
TEXT, $EMAIL TEXT, $COURSE TEXT)")
        db.execSQL(createTableStudent)
    }
    override fun onUpgrade(db: SQLiteDatabase?, oldVersion: Int,
newVersion: Int) {
        db.execSQL("DROP TABLE IF EXISTS $TABLE_STUDENT")
        onCreate(db)
    }
    fun insertStudent(studentModel: StudentModel): Long {
        val db = this.writableDatabase
        val contentValues = ContentValues()
        contentValues.put(ID, studentModel.id)
        contentValues.put(NAME, studentModel.name)
        contentValues.put(EMAIL, studentModel.email)
        contentValues.put(COURSE, studentModel.course)
    }
}
```

```

        val success = db.insert(TABLE_STUDENT, null, contentValues)
        db.close()
        return success
    }
    fun readAllStudents(): ArrayList<StudentModel> {
        val studentList: ArrayList<StudentModel> = ArrayList()
        val readQuery = "SELECT * FROM $TABLE_STUDENT"
        val db = this.readableDatabase
        val cursor: Cursor?
        try {
            cursor = db.rawQuery(readQuery, null)
        } catch (e: Exception) {

            e.printStackTrace()
            db.execSQL(readQuery)
            return ArrayList()
        }
        var id: Int
        var name: String
        var email: String
        var course: String
        if (cursor.moveToFirst()) {
            do {
                //           id = cursor.getInt(cursor.getColumnIndex("id"))
                id = cursor.getString(0).toInt()
                name = cursor.getString(cursor.getColumnIndex("name"))
                email = cursor.getString(cursor.getColumnIndex("email"))
                course = cursor.getString(cursor.getColumnIndex("course"))
                val studentModel = StudentModel(id = id, name = name, email
= email, course = course)
                studentList.add(studentModel)
            }while (cursor.moveToNext())
        }
        return studentList
    }
    fun updateStudent(studentModel: StudentModel):Int {
        val db = this.writableDatabase
        val contentValues = ContentValues()
        contentValues.put(ID, studentModel.id)
        contentValues.put(NAME, studentModel.name)
        contentValues.put(EMAIL, studentModel.email)
        contentValues.put(COURSE, studentModel.course)
        val success = db.update(TABLE_STUDENT, contentValues, "id =
${studentModel.id}", null)
        db.close()

        return success
    }
    fun deleteStudentById(id: Int): Int {
        val db = this.writableDatabase
        val contentValues = ContentValues()
        contentValues.put(ID, id)
        val success = db.delete(TABLE_STUDENT, "id = $id", null)
        db.close()
        return success
    }
}

```

MainActivity.kt-

```

package com.example.sqlite_crud

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import androidx.recyclerview.widget.LinearLayoutManager
import com.karthik.crudsqliite.databinding.ActivityMainBinding

class MainActivity : AppCompatActivity() {
    lateinit var binding: ActivityMainBinding
    private lateinit var sqLiteHelper: SQLiteHelper
    private var studentAdapter: StudentAdapter? = null
    private var studentModel: StudentModel? = null

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)
        sqLiteHelper = SQLiteHelper(this)
        binding.mainCreateButton.setOnClickListener {
            binding.relLay1.visibility = View.VISIBLE
            binding.mainCreateButton.visibility = View.GONE
            binding.createButton.visibility = View.VISIBLE
            binding.updateButton.visibility = View.GONE
        }
        binding.createButton.setOnClickListener {
            addStudent()
        }
        binding.updateButton.setOnClickListener {
            updateStudent()
        }
        displayStudent()
        studentAdapter?.setOnItemClickListener {
            binding.name.setText(it.name)
            binding.email.setText(it.email)
            binding.course.setText(it.course)
            studentModel = it
            binding.relLay1.visibility = View.VISIBLE
            binding.createButton.visibility = View.GONE
            binding.updateButton.visibility = View.VISIBLE
        }
        studentAdapter!!.setOnDeleteItem {
            deleteStudent(it.id)
        }
    }

    private fun addStudent() {
        val name = binding.name.text.toString()
        val email = binding.email.text.toString()
        val course = binding.course.text.toString()
        if (name.isEmpty() || email.isEmpty() || course.isEmpty()) {
            Toast.makeText(applicationContext, "Enter Required values!", Toast.LENGTH_SHORT).show()
        } else {
            val studentModel = StudentModel(name = name, email = email,
            course = course)
            val status = sqLiteHelper.insertStudent(studentModel)

            if (status > -1) {

```

```

        Toast.makeText(applicationContext, "Student Added into
DB!", Toast.LENGTH_SHORT)
            .show()
        binding.rellay1.visibility = View.GONE
        binding.mainCreateButton.visibility = View.VISIBLE
        displayStudent()
        clearEdittext()
    } else {
        Toast.makeText(applicationContext, "Values not added!",
Toast.LENGTH_SHORT).show()
    }
}
private fun displayStudent() {
    binding.studentDisplayRecyclerView.layoutManager =
LinearLayoutManager(this)
    studentAdapter = StudentAdapter(applicationContext)
    binding.studentDisplayRecyclerView.adapter = studentAdapter
    val studentList = sqLiteHelper.readAllStudents()
    studentAdapter!!.addItems(studentList)
}
private fun updateStudent() {
    val name = binding.name.text.toString()
    val email = binding.email.text.toString()
    val course = binding.course.text.toString()
    if(name == studentModel?.name && email == studentModel?.email &&
course == studentModel?.course){
        Toast.makeText(applicationContext, "Record not changed...",,
Toast.LENGTH_SHORT).show()
        return
    }
    if(studentModel == null) return

    val student = StudentModel(id= studentModel!!.id, name= name,
email= email, course= course)
    val status = sqLiteHelper.updateStudent(student)
    if(status > -1){
        clearEdittext()
        displayStudent()
        binding.rellay1.visibility = View.GONE
        Toast.makeText(applicationContext, "Update Success!",,
Toast.LENGTH_SHORT).show()
    } else {
        Toast.makeText(applicationContext, "Update Failed!",,
Toast.LENGTH_SHORT).show()
    }
}
private fun deleteStudent(id: Int) {
    if (id == null) return
    val builder = AlertDialog.Builder(this)
    builder.setMessage("Are you sure you want to delete this student?")
    builder.setCancelable(true)
    builder.setPositiveButton("Yes") { dialog, _ ->
        sqLiteHelper.deleteStudentById(id)
        displayStudent()
        dialog.dismiss()
    }
    builder.setNegativeButton("No") { dialog, _ ->
        dialog.dismiss()
    }
    val alert = builder.create()
}

```

```

        alert.show()
    }
    private fun clearEdittext() {
        binding.name.setText("")
        binding.email.setText("")

        binding.course.setText("")
        binding.name.requestFocus()
    }
}

```

StudentModel.kt

```

package com.karthik.crudsqlite
import java.util.*
data class StudentModel (
    var id: Int = getId(),
    var name: String = "",
    var email: String = "",
    var course: String = ""
) {
    companion object{
        fun getId(): Int {
            val random = Random()
            return random.nextInt(100)
        }
    }
}

```

StudentAdapter.kt

```

package com.karthik.crudsqlite
import android.content.Context
import android.content.Intent
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.ImageView

import android.widget.PopupMenu
import android.widget.TextView
import androidx.recyclerview.widget.RecyclerView
class StudentAdapter(val context: Context):
RecyclerView.Adapter<StudentAdapter.StudentViewHolder>() {
    private var studentList: ArrayList<StudentModel> = ArrayList()
    private var onClickDeleteItem : ((StudentModel) -> Unit)? = null
    private var onClickItem : ((StudentModel) -> Unit)? = null
    fun addItems(items: ArrayList<StudentModel>){
        this.studentList = items
        notifyDataSetChanged()
    }
    fun setOnClickItems(callback: (StudentModel)-> Unit) {
        this.onClickItem = callback
    }
    fun setOnClickDeleteItem(callback: (StudentModel)-> Unit){
        this.onClickDeleteItem = callback
    }
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int) =
StudentViewHolder(

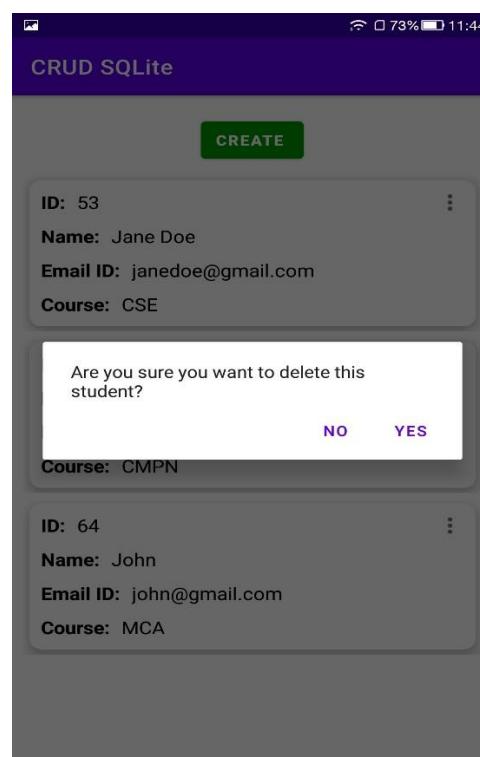
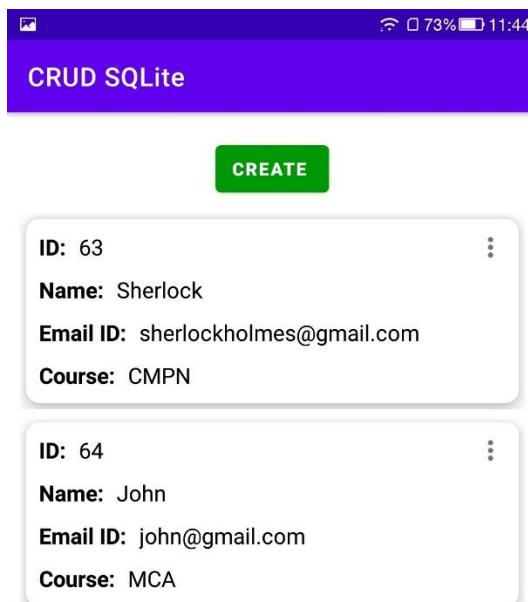
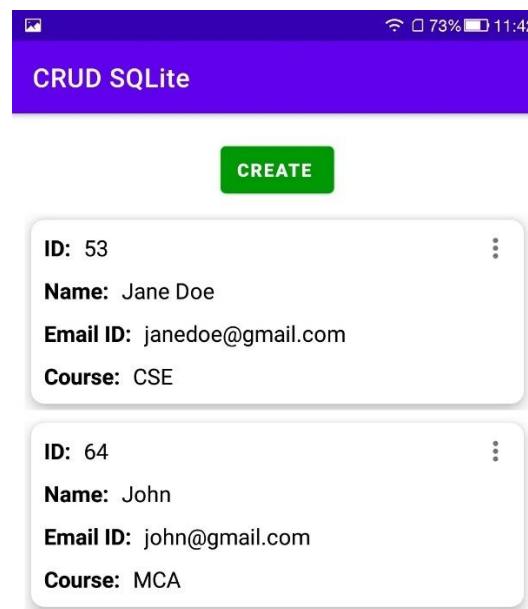
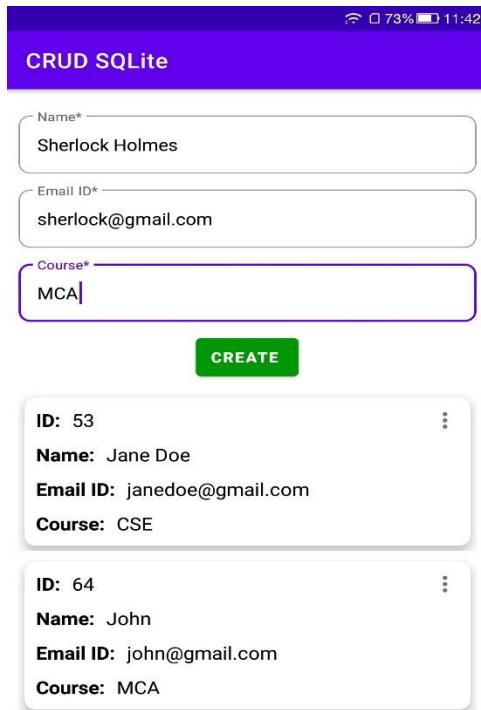
```

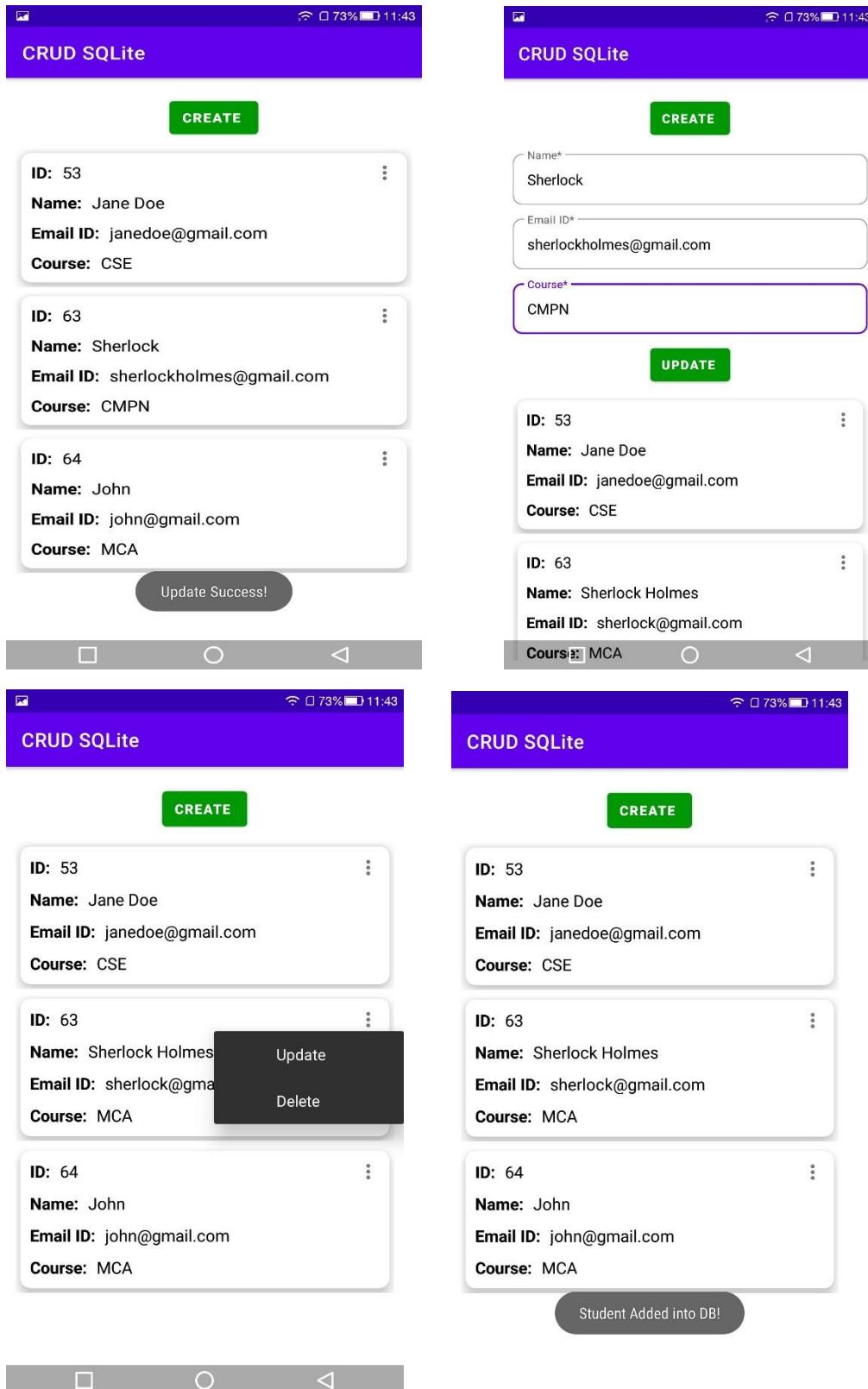
```
LayoutInflater.from(parent.context).inflate(R.layout.student_card_view,
parent, false)
)
override fun onBindViewHolder(holder: StudentViewHolder, position: Int)
{
    val student = studentList[position]
    holder.bindView(student)
    holder.moreBtn.setOnClickListener {
        popUpMenu(it, student)
    }
}
private fun popUpMenu(view: View, student: StudentModel) {
    val popUpMenus = PopupMenu(context, view)
    popUpMenus.inflate(R.menu.menu_rv)

    popUpMenus.setOnMenuItemClickListener {
        when(it.itemId){
            R.id.update_menu->{
                onClickItem?.invoke(student)
                true
            }
            R.id.delete_menu->{
                onClickDeleteItem?.invoke(student)
                true
            }
            else -> true
        }
    }
    popUpMenus.show()
    val popUp = PopupMenu::class.java.getDeclaredField("mPopup")
    popUp.isAccessible = true
    val menu = popUp.get(popUpMenus)
    menu.javaClass.getDeclaredMethod("setForceShowIcon",
Boolean::class.java)
        .invoke(menu, true)
    }
    override fun getItemCount(): Int {
        return studentList.size
    }
    class StudentViewHolder(var view: View): RecyclerView.ViewHolder(view) {
        var id = view.findViewById<TextView>(R.id.id_rv)
        var name = view.findViewById<TextView>(R.id.name_rv)
        var email = view.findViewById<TextView>(R.id.email_rv)
        var course = view.findViewById<TextView>(R.id.course_rv)
        var moreBtn = view.findViewById<ImageView>(R.id.more_rv)
        fun bindView(studentModel: StudentModel){

            id.text = studentModel.id.toString()
            name.text = studentModel.name
            email.text = studentModel.email
            course.text = studentModel.course
        }
    }
}
```

Output:





Conclusion:

Android program to perform CRUD operation using SQLite DB executed successfully.

Practical No – 6

Aim: To implement file I/O and Shared Preferences.

Theory:

Data and file storage overview

Android uses a file system that's similar to disk-based file systems on other platforms. The system provides several options for you to save your app data:

- **App-specific storage:** Store files that are meant for your app's use only, either in dedicated directories within an internal storage volume or different dedicated directories within external storage. Use the directories within internal storage to save sensitive information that other apps shouldn't access.
- **Shared storage:** Store files that your app intends to share with other apps, including media, documents, and other files.
- **Preferences:** Store private, primitive data in key-value pairs.
- **Databases:** Store structured data in a private database using the Room persistence library.

Reading and Writing Text File in Android Internal Storage

Android offers openFileInput and openFileOutput from the Java I/O classes to modify reading and writing streams from and to local files.

`openFileOutput(): This method is used to create and save a file.`

Its syntax is given below:

```
FileOutputStream fOut = openFileOutput("file name", Context.MODE_PRIVATE);
```

The method openFileOutput() returns an instance of FileOutputStream. After that we can call write method to write data on the file.

Its syntax is given below:

```
String str = "test data";
fOut.write(str.getBytes());
fOut.close();
```

`openFileInput(): This method is used to open a file and read it. It returns an instance of FileInputStream.`

Its syntax is given below:

```
FileInputStream fin = openFileInput(file);
```

After that, we call read method to read one character at a time from the file and then print it.

Its syntax is given below:

```
int c;
String temp="";
while( (c = fin.read()) != -1){
    temp = temp + Character.toString((char)c);
}
fin.close();
```

Shared Preference:

Interface for accessing and modifying preference data returned by Context.getSharedPreferences(String, int). For any particular set of preferences, there is a single instance of this class that all clients share. Modifications to the preferences must go through an Editor object to ensure the preference values remain in a consistent state and control when they are committed to storage. Objects that are returned from the various get methods must be treated as immutable by the application.

Two Ways To Save Data Through Shared Preference:

There are two different ways to save data in Android through Shared Preferences – One is using Activity based preferences and other is creating custom preferences.

Activity Preferences:

- For activity preferences developer have to call function getPreferences (int mode) available in Activity class
- Use only when one preference file is needed in Activity
- It doesn't require name as it will be the only preference file for this activity
- Developer doesn't usually prefer using this even if they need only one preference file in Activity. They prefer using custom getSharedPreferences(String name,int mode).

To use activity preferences developer have to call function getPreferences (int mode) available in Activity class. The function getPreferences(int mode) call the other function used to create custom preferences i.e getSharedPreferences(String name,int mode). Just because Activity contains only one preference file so getPreferences(int mode) function simply pass the name of Activity class to create a preference file.

Important Note: Mode are discussed in Custom preferences.

Custom Preferences:

- Developer needs to use getSharedPreferences(String name,int mode) for custom preferences
- Used in cases when more than one preference file required in Activity
- Name of the preference file is passed in first parameter

Custom Preferences can be created by calling function getSharedPreferences(String name,int mode), it can be called from anywhere in the application with reference of Context. Here name is any preferred name for example: User,Book etc. and mode is used to set kind of privacy for file.

There are three types of Mode in Shared Preference:

1. Context.MODE_PRIVATE – default value (Not accessible outside of your application)

MODE_PRIVATE – It is a default mode. MODE_PRIVATE means that when any preference file is created with private mode then it will not be accessible outside of your application. This is the most common mode which is used.

2. Context.MODE_WORLD_READABLE – readable to other apps

MODE_WORLD_READABLE – If developer creates a shared preference file using mode world readable then it can be read by anyone who knows its name, so any other outside application can easily read data of your app. This mode is very rarely used in App.

3. Context.MODE_WORLD_WRITEABLE – read/write to other apps

MODE_WORLD_WRITEABLE – It's similar to mode world readable but with both kind of accesses i.e read and write. This mode is never used in App by Developer.

Q1. Program to create a file in a directory and perform following file operations,

- a) Write into a file
- b) Read from a file
- c) Delete a file

Code:

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"
    android:orientation="vertical"
    android:layout_margin="15dp"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/textview"
        android:text="Enter Content"
        android:textSize="20dp"
        app:layout_editor_absoluteX="0dp"
        app:layout_editor_absoluteY="30dp"/>

    <EditText
        android:id="@+id/etextForFile"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Type Content Here"
        android:textSize="20dp"
        app:layout_constraintTop_toBottomOf="@+id/textview"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        />

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

```

    app:layout_constraintTop_toBottomOf="@+id/etextForFile">

<RelativeLayout
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="2"
    android:gravity="center"
    android:layout_marginHorizontal="2.5dp">

    <Button
        android:id="@+id/Save"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:onClick="BtnWrite"
        android:text="save"/>
</RelativeLayout>

<RelativeLayout
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="2"
    android:gravity="center"
    android:layout_marginHorizontal="2.5dp">

    <Button
        android:id="@+id/Load"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:onClick="BtnRead"
        android:text="load" />
</RelativeLayout>

<RelativeLayout
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="2"
    android:gravity="center"
    android:layout_marginHorizontal="2.5dp">

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

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

```

MainActivity.java

```

package com.example.practical4_1;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.EditText;
import android.view.View;
import android.widget.Toast;
import java.io.BufferedReader;
import java.io.File;

```

```

import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;

public class MainActivity extends AppCompatActivity {
    EditText mEditText;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        mEditText = findViewById(R.id.etextForFile);
    }
    public void BtnWrite(View v) {
        try {
            FileOutputStream fileout=openFileOutput("mytextfield.txt",
            MODE_PRIVATE);
            OutputStreamWriter outputWriter=new
            OutputStreamWriter(fileout);
            outputWriter.write(mEditText.getText().toString());
            outputWriter.close();

            Toast.makeText(getApplicationContext(), "File saved successfully!",
                Toast.LENGTH_SHORT).show();
            Toast.makeText(this, "Saved to " + getFilesDir() + "/" +
            "mytextfield",
                Toast.LENGTH_LONG).show();
        } catch (Exception e) {
            e.printStackTrace();
        }
        mEditText.setText("");
    }

    public void BtnRead(View v) {
        FileInputStream fileIn = null;
        try {
            fileIn=openFileInput("mytextfield.txt");
            InputStreamReader InputRead= new InputStreamReader(fileIn);
            BufferedReader br = new BufferedReader(InputRead);
            StringBuilder sb = new StringBuilder();
            String text;
            while ((text = br.readLine()) != null) {
                sb.append(text).append("\n");
            }
            mEditText.setText(sb.toString());
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        } finally {
            if (fileIn != null) {
                try {
                    fileIn.close();
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        }
    }
}

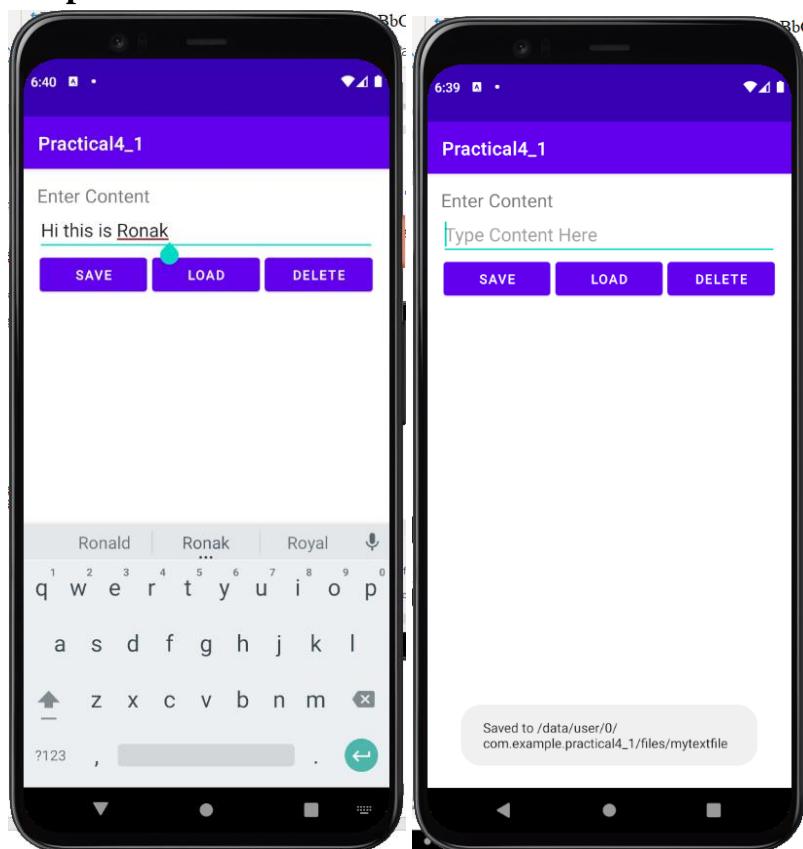
```

```
}

public void deleteBtn(View v) {
    File file;
    try {
        file = new File(this.getFilesDir(), "mytextfield.txt");
        if (file.exists()) {
            file.delete();
            Toast.makeText(this, "File mytextfield deleted
Successfully",
                Toast.LENGTH_LONG).show();
        }
        else {
            Toast.makeText(this, "File mytextfield doesn't exist",
                Toast.LENGTH_LONG).show();
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    mEditText.setText("");
}

}
```

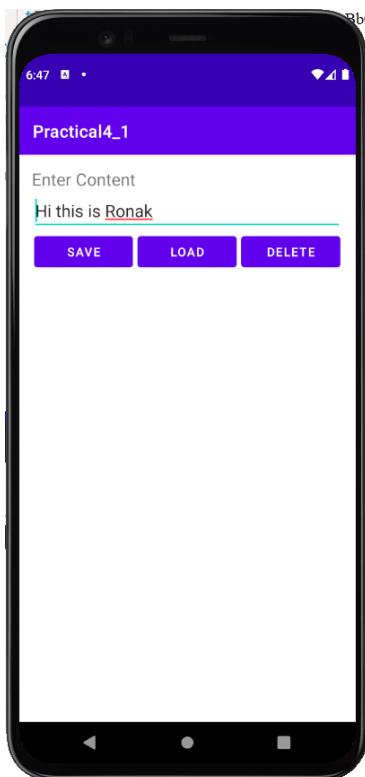
Output:



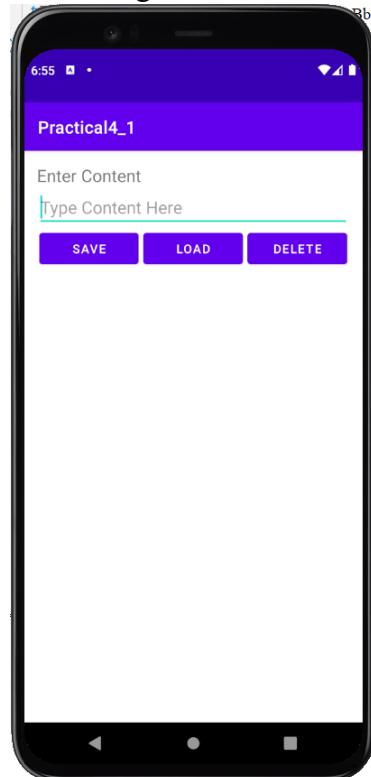
File Saved:

```
✓ com.example.practical4_1 drwx-----  
  > cache drwxrws--x  
  > code_cache drwxrws--x  
  > files drwxrwx--x  
    mytextfile.txt -rw-rw----
```

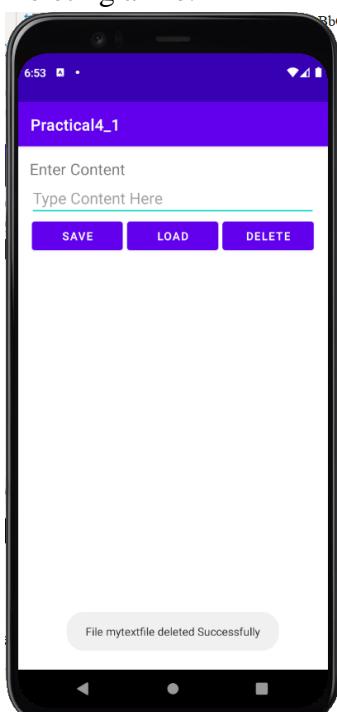
Reading a file:



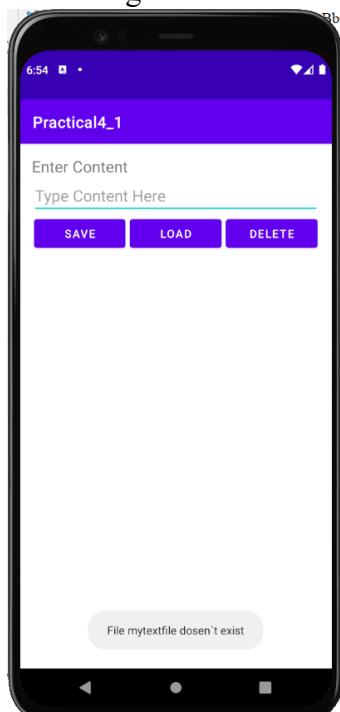
Reading a file when it doesn't exist:



Deleting a file:



Deleting a file when it doesn't exist:



Before Deleting file:

com.example.practical4_1	drwx-----	
> cache	drwxrws--x	
> code_cache	drwxrws--x	
> files	drwxrwx--x	
mytextfile.txt	-rw-rw----	

After Deleting a file:

com.example.practical4_1	drwx-----	
> cache	drwxrws--x	
> code_cache	drwxrws--x	
files	drwxrwx--x	

Q2. Create a new project and create a login Activity. In this create a login UI asking user email and password with an option of remember me checkbox. Also a button displaying Sign In or Register using shared preferences.

Code:

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"
    android:orientation="vertical"
    android:layout_margin="15dp"
    tools:context=".MainActivity">

    <EditText
        android:id="@+id/email"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Email"
        android:textSize="20sp"
        app:layout_editor_absoluteX="0dp"
        app:layout_editor_absoluteY="30dp" />

    <EditText
        android:id="@+id/pass"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Password"
        android:inputType="textPassword"
        android:textSize="20sp"
        app:layout_constraintTop_toBottomOf="@+id/email" />

    <CheckBox
        android:id="@+id/remCheck"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Remember me"
```

```

        android:textSize="18sp"
        app:layout_constraintTop_toBottomOf="@+id/pass"/>
<Button
    android:id="@+id/signInBtn"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Sign in or Register"
    app:layout_constraintTop_toBottomOf="@+id/remCheck"/>
</androidx.constraintlayout.widget.ConstraintLayout>

```

MainActivity.java

```

package com.example.practical4_2;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.EditText;

public class MainActivity extends AppCompatActivity {
    private EditText etmail;
    private EditText etpass;
    private CheckBox cbrememberme;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        etmail = findViewById(R.id.email);
        etpass = findViewById(R.id.pass);
        cbrememberme = findViewById(R.id.remCheck);

        Button loginButton = (Button) findViewById(R.id.signInBtn);
        loginButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                attemptLogin();
            }
        });
        if (!new PrefManager(this).isUserLoggedOut()) {
            startHomeActivity();
        }
    }
    private void attemptLogin() {
        etmail.setError(null);
        etpass.setError(null);
        String email = etmail.getText().toString();
        String password = etpass.getText().toString();
        boolean cancel = false;
        View focusView = null;
        if (TextUtils.isEmpty(password)) {
            etpass.setError("Field Required");
            focusView = etpass;
            cancel = true;
        } else if (!isValidPassword(password)) {
            etpass.setError("Invalid Password");
            focusView = etpass;
        }
    }
}

```

```

        cancel = true;
    }
    if (TextUtils.isEmpty(email)) {
        etmail.setError("Field Required");
        focusView = etmail;
        cancel = true;
    } else if (!isValidEmail(email)) {
        etmail.setError("Invalid Email");
        focusView = etmail;
        cancel = true;
    }

    if (cancel) {
        focusView.requestFocus();
    } else {
        if (cbrememberme.isChecked())
            saveLoginDetails(email, password);
        startHomeActivity();
    }
}

private void startHomeActivity() {
    Intent intent = new Intent(this, ActivityHome.class);
    startActivity(intent);
    finish();
}

private void saveLoginDetails(String email, String password) {
    new PrefManager(this).saveLoginDetails(email, password);
}

private boolean isValidEmail(String email) {
    return email.contains("@");
}

private boolean isPasswordValid(String password) {
    return password.length() > 4;
}
}
}

```

PrefManager.java

```

package com.example.practical4_2;

import android.content.Context;
import android.content.SharedPreferences;

public class PrefManager {
    Context context;
    PrefManager(Context context) {
        this.context = context;
    }

    public void saveLoginDetails(String email, String password) {
        SharedPreferences sharedpreferences =
context.getSharedPreferences("LoginDetails", Context.MODE_PRIVATE);
        SharedPreferences.Editor editor = sharedpreferences.edit();
        editor.putString("Email", email);
        editor.putString("Password", password);
        editor.commit();
    }
}

```

```

    }
    public boolean isUserLoggedOut() {
        SharedPreferences sharedpreferences =
context.getSharedPreferences("LoginDetails", Context.MODE_PRIVATE);
        boolean isEmpty = sharedpreferences.getString("Email",
"").isEmpty();
        boolean isPasswordEmpty = sharedpreferences.getString("Password",
"").isEmpty();
        return isEmpty || isPasswordEmpty;
    }
}

```

activity_home.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=".ActivityHome">
<TextView
    android:id="@+id/text"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_marginTop="250dp"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"/>

<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/logoutBtn"
    android:text="Logout"
    app:layout_constraintTop_toBottomOf="@id/text"/>
</androidx.constraintlayout.widget.ConstraintLayout>

```

Activity Home.java

```

package com.example.practical4_2;

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

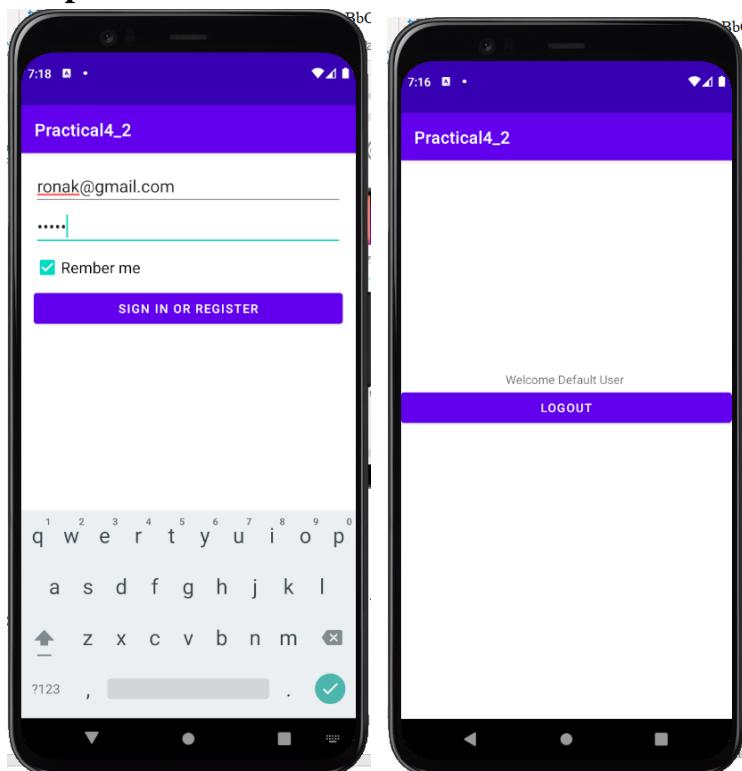
public class ActivityHome extends AppCompatActivity {
    TextView tvtext;
    Button btlogout;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_home);
        tvtext = findViewById(R.id.text);
    }
}

```

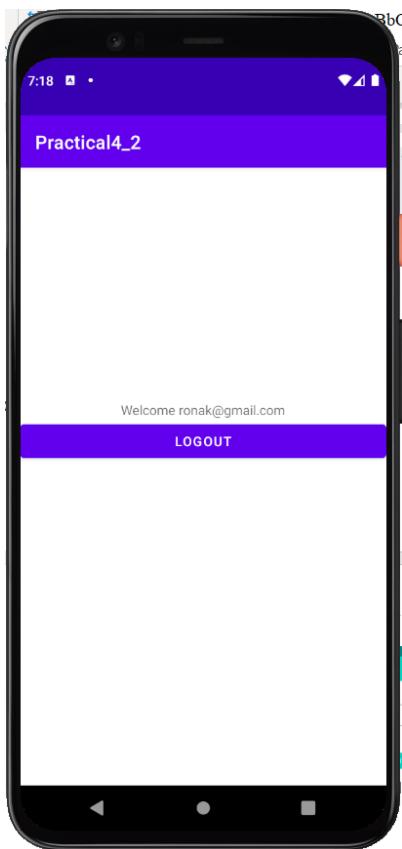
```
        btlogout = findViewById(R.id.logoutBtn);
        Sharedpreferences sharedpreferences =
getSharedpreferences("LoginDetails", Context.MODE_PRIVATE);
        Sharedpreferences.Editor editor = sharedpreferences.edit();
        String email = sharedpreferences.getString("Email", "Default User
");
        tvtext.setText("Welcome " + email);
        btlogout.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                editor.remove("Email");
                editor.remove("Password");
                editor.apply();
                finish();
            }
        });
    }

}
```

Output:

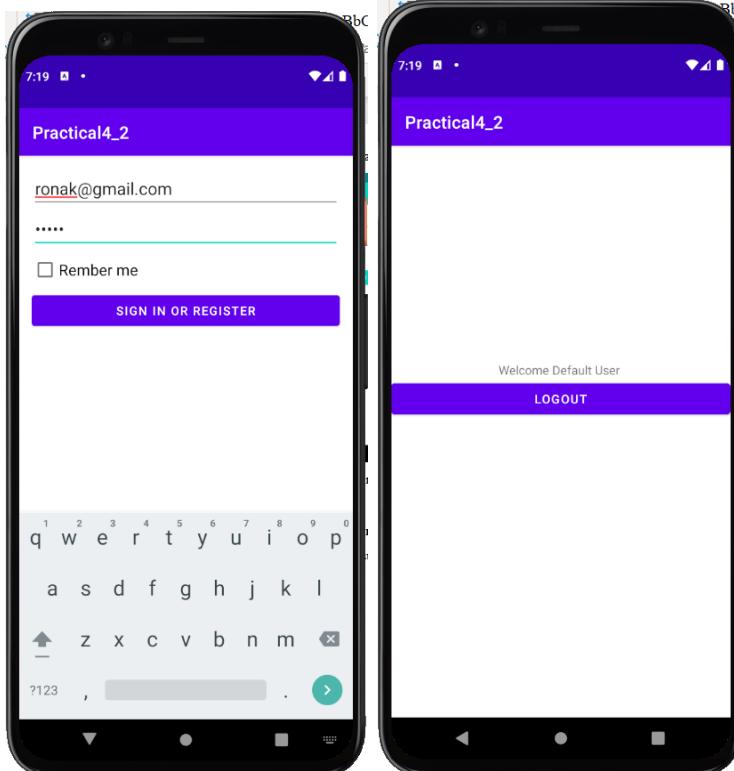


Closing app without logging out and relaunching the app

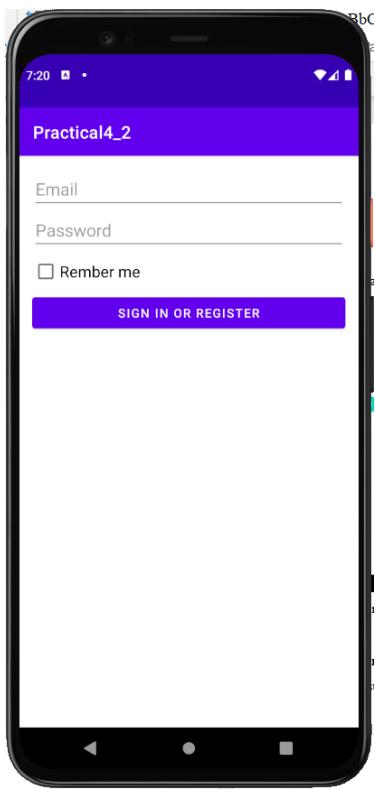


App redirects to home page as it remembers the credentials.

Login with credentials while check box is unticked:



Closing the app without logging out and relaunching the app



Application redirects to login page as it doesn't remember the credentials.

Conclusion:

We successfully implemented file I/O and Shared preference.

Practical No – 7

Aim: To perform the animation on an image and to apply various filters on an image.

Theory:

ImageView:

ImageView class is used to display any kind of image resource in the android application either it can be android.graphics.Bitmap or android.graphics.drawable.Drawable (it is a general abstraction for anything that can be drawn in Android). ImageView class or android.widget.ImageView inherits the android.view.View class which is the subclass of Kotlin.Any class. Application of ImageView is also in applying tints to an image in order to reuse a drawable resource and create overlays on background images. Moreover, ImageView is also used to control the size and movement of an image.

Whenever ImageView is added to an activity, it means there is a requirement for an image resource. Thus it is oblivious to provide an Image file to that ImageView class. It can be done by adding an image file that is present in the Android Studio itself or we can add our own image file. Android Studio owns a wide range of drawable resources which are very common in the android application layout. The following are the steps to add a drawable resource to the ImageView class.

Image Effects:

Android allows you to manipulate images by adding different kinds of effects to the images. You can easily apply image processing techniques to add certain kinds of effects to images. The effects could be brightness, darkness, grayscale conversion e.t.c.

Method & description:

1. **copy(Bitmap.Config config, boolean isMutable):**

This method copy this bitmap's pixels into the new bitmap.

2. **createBitmap(DisplayMetrics display, int width, int height, Bitmap.Config config):**

Returns a mutable bitmap with the specified width and height.

3. **createBitmap(int width, int height, Bitmap.Config config):**

Returns a mutable bitmap with the specified width and height.

4. **createBitmap(Bitmap src):**

Returns an immutable bitmap from the source bitmap.

5. **extractAlpha():**

Returns a new bitmap that captures the alpha values of the original.

6. getConfig():

This method return that config, otherwise return null.

7. getDensity():

Returns the density for this bitmap.

8. getRowBytes():

Return the number of bytes between rows in the bitmap's pixels.

9. setPixel(int x, int y, int color):

Write the specified Color into the bitmap (assuming it is mutable) at the x,y coordinate.

10. setDensity(int density):

This method specifies the density for this bitmap.

A) Perform the following animation on the image:

1. Move.
2. Rotate.
3. Expand.

Code:**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"
    android:layout_margin="15dp"
    android:orientation="vertical"
    tools:context=".MainActivity">
    <LinearLayout
        android:id="@+id/layout1"
        android:layout_marginTop="60dp"
        android:layout_width="match_parent"
        android:orientation="vertical"
        android:layout_height="wrap_content">
        <ImageView
            android:id="@+id/imgView"
            android:layout_width="150dp"
            android:layout_height="150dp"
            android:src="@drawable/car">
        </ImageView>
    </LinearLayout>
    <LinearLayout
        android:layout_marginTop="100dp"
        android:layout_width="match_parent"
        android:orientation="horizontal"
        android:weightSum="3"
        android:layout_height="wrap_content">
```

```

    app:layout_constraintTop_toBottomOf="@+id/layout1">
<Button
    android:id="@+id/expand"
    android:layout_width="wrap_content"
    android:layout_marginHorizontal="5dp"
    android:layout_weight="1"
    android:text="Expand"
    android:layout_height="wrap_content">
</Button>
<Button
    android:id="@+id/rotate"
    android:layout_width="wrap_content"
    android:layout_weight="1"
    android:layout_marginHorizontal="5dp"
    android:text="Rotate"
    android:layout_height="wrap_content">
</Button>
<Button
    android:id="@+id/move"
    android:layout_width="wrap_content"
    android:layout_weight="1"
    android:layout_marginHorizontal="5dp"
    android:text="Move"
    android:layout_height="wrap_content">
</Button>
</LinearLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

```

MainActivity.java

```

package com.example.practical5_1;

import androidx.appcompat.app.AppCompatActivity;
import android.animation.AnimatorSet;
import android.animation.ObjectAnimator;
import android.os.Bundle;
import android.view.View;
import android.view.animation.Animation;
import android.view.animation.AnimationUtils;
import android.widget.Button;

public class MainActivity extends AppCompatActivity {
    Button expandBtn;
    Button moveBtn;
    Button rotateBtn;
    float init = 0f;
    float ext = 90f;
    float xStart = 120f;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        long animationDuration = 1000;

        expandBtn = findViewById(R.id.expand);
        moveBtn = findViewById(R.id.move);
        rotateBtn = findViewById(R.id.rotate);
        View imgView = findViewById(R.id.imageView);
    }
}

```

```

        expandBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Animation aniSlide =
                AnimationUtils.loadAnimation(getApplicationContext(),R.anim.zoom_in);

                imgView.setVisibility(View.VISIBLE);
                imgView.startAnimation(aniSlide);
            }
        });

        rotateBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                ObjectAnimator rotateAnimation=
                ObjectAnimator.ofFloat(imgView,"rotation",init, ext);
                rotateAnimation.setDuration(animationDuration);
                AnimatorSet animatorSet = new AnimatorSet();
                animatorSet.playTogether(rotateAnimation);
                animatorSet.start();
                init+=90;
                ext+=90;
            }
        });

        moveBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                ObjectAnimator
                animatorX=ObjectAnimator.ofFloat(imgView,"x",xStart);
                animatorX.setDuration(animationDuration);
                ObjectAnimator
                animatorY=ObjectAnimator.ofFloat(imgView,"y",200f);
                animatorY.setDuration(animationDuration);
                ObjectAnimator alphaAnimation =
                ObjectAnimator.ofFloat(imgView,View.ALPHA,1.0f,0.0f);
                alphaAnimation.setDuration(animationDuration);
                AnimatorSet animatorSet = new AnimatorSet();
                animatorSet.playTogether(animatorX);
                animatorSet.start();
                xStart+=120f;
            }
        });
    }
}

```

zoom_in.xml

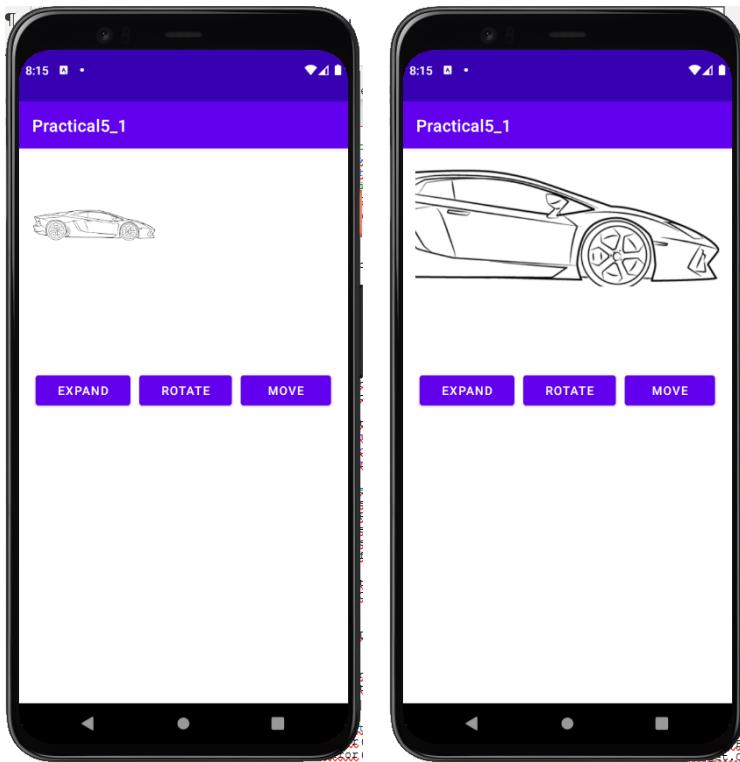
```

<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android">
    <scale
        xmlns:android="http://schemas.android.com/apk/res/android"
        android:duration="1000"
        android:fromXScale="2"
        android:fromYScale="2"
        android:pivotX="50%"
        android:pivotY="50%"
        android:toXScale="4"
        android:toYScale="4" >
    </scale>
</set>

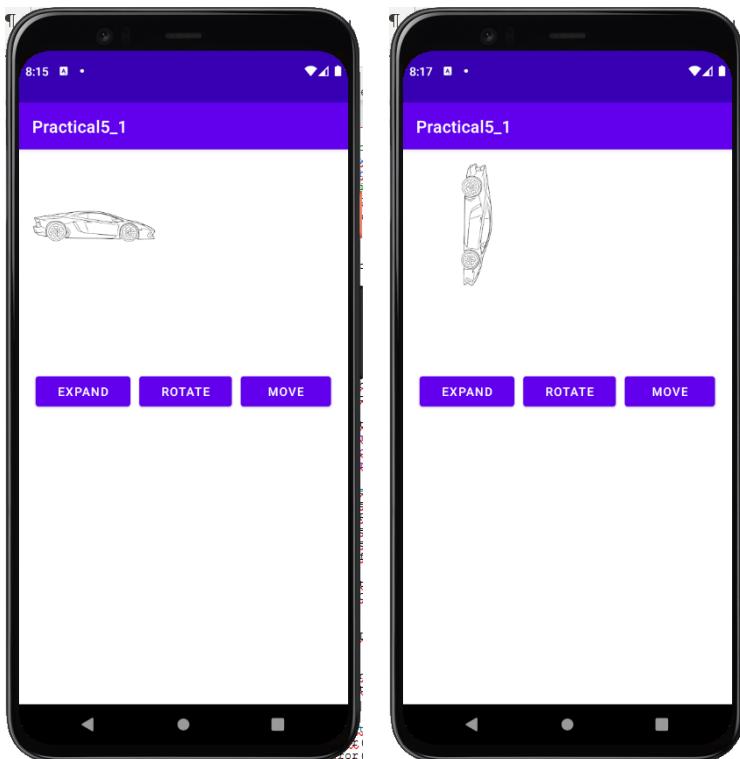
```

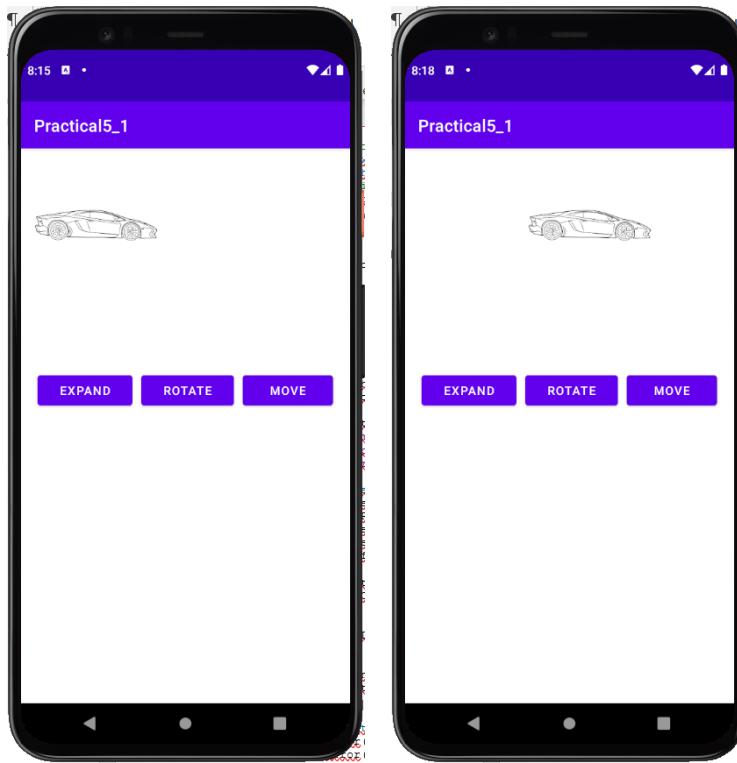
Output:

Expand:



Rotate:



Move:**B) Apply the following effects on the image:**

1. Brightness.
2. Darkness.
3. Grayscale.

Code:**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"
    android:orientation="vertical"
    android:layout_margin="15dp"
    tools:context=".MainActivity">

    <LinearLayout
        android:layout_marginTop="10dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/layout1"
        android:orientation="vertical">
        <ImageView
            android:id="@+id/imgView"
            android:layout_width="match_parent"
            android:layout_gravity="center_horizontal"
```

```

        android:layout_height="match_parent"
        android:src="@drawable/image"
    
```

```

</Imageview>
</LinearLayout>

<LinearLayout
    android:layout_marginTop="0dp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:layout_constraintTop_toBottomOf="@+id/layout1"
    android:orientation="horizontal"
    android:weightSum="3"           >
    <Button
        android:id="@+id/dark"
        android:layout_width="wrap_content"
        android:layout_marginHorizontal="2.5dp"
        android:layout_weight="1"
        android:text="Darkness"
        android:layout_height="wrap_content">
    
```

```

    </Button>
    <Button
        android:id="@+id/bright"
        android:layout_width="wrap_content"
        android:layout_weight="1"
        android:layout_marginHorizontal="2.5dp"
        android:text="Brightness"
        android:layout_height="wrap_content">
    
```

```

    </Button>
    <Button
        android:id="@+id/gray"
        android:layout_width="wrap_content"
        android:layout_weight="1"
        android:layout_marginHorizontal="2.5dp"
        android:text="Grayscale"
        android:layout_height="wrap_content">
    
```

```

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

```

MainActivity.xml

```

package com.example.practical5_2;

import androidx.appcompat.app.AppCompatActivity;
import android.graphics.Bitmap;
import android.graphics.Color;
import android.graphics.drawable.BitmapDrawable;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;

public class MainActivity extends AppCompatActivity {
    Button darkBtn, brightBtn, grayBtn;
    ImageView imgView;
    private Bitmap bmp;
    private Bitmap operation;

    @Override
    protected void onCreate(Bundle savedInstanceState) {

```

```

super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
darkBtn = (Button) findViewById(R.id.dark);
brightBtn = (Button) findViewById(R.id.bright);
grayBtn= (Button) findViewById(R.id.gray);
imgView = (ImageView) findViewById(R.id.imageView);
BitmapDrawable abmp = (BitmapDrawable) imgView.getDrawable();
bmp = abmp.getBitmap();

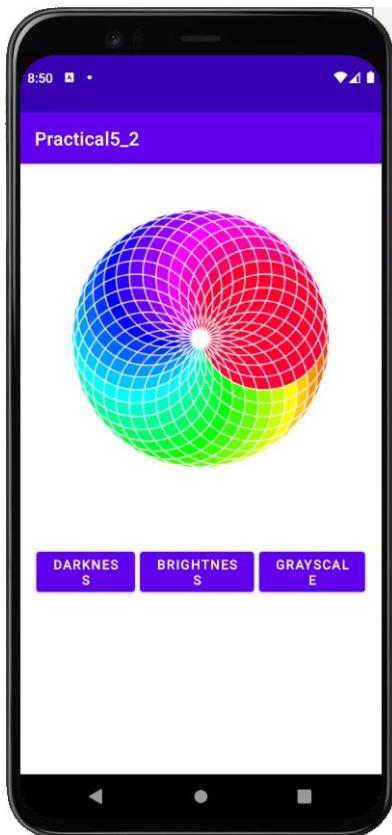
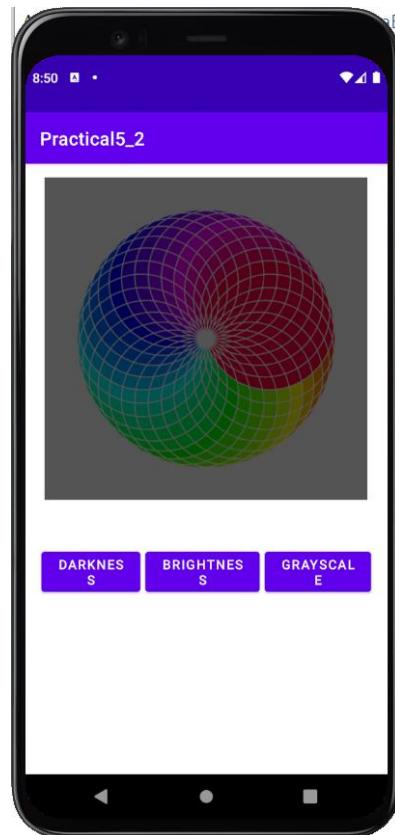
darkBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        operation =
Bitmap.createBitmap(bmp.getWidth(), bmp.getHeight(), bmp.getConfig());
        double x=0.33;
        for (int i = 0; i < bmp.getWidth(); i++) {
            for (int j = 0; j < bmp.getHeight(); j++) {
                int p = bmp.getPixel(i, j);
                int r = Color.red(p);
                int g = Color.green(p);
                int b = Color.blue(p);
                r = (int) (x * r);
                g = (int) (x * g);
                b = (int) (x * b);
                operation.setPixel(i, j, Color.argb(Color.alpha(p),
r, g, b));
            }
        }
        imgView.setImageBitmap(operation);
    }
});

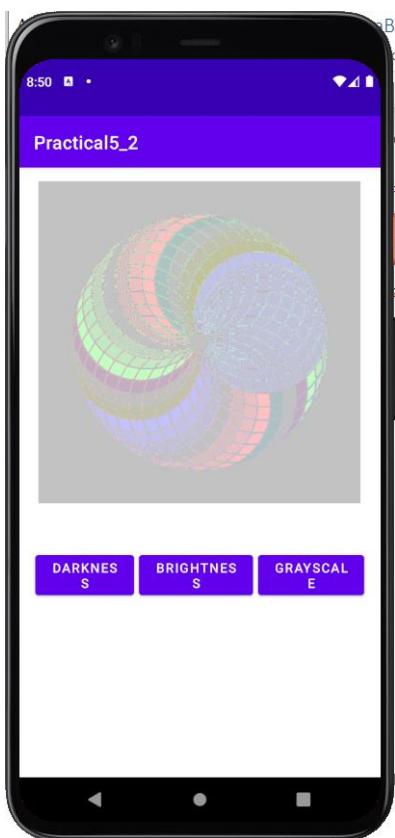
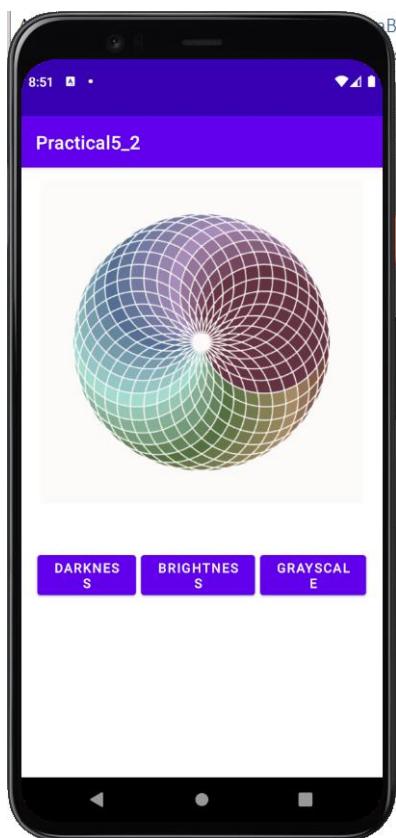
brightBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        operation= Bitmap.createBitmap(bmp.getWidth(),
                bmp.getHeight(), bmp.getConfig());
        for(int i=0; i<bmp.getWidth(); i++){
            for(int j=0; j<bmp.getHeight(); j++) {
                int p = bmp.getPixel(i, j);
                int r = Color.red(p);
                int g = Color.green(p);
                int b = Color.blue(p);
                int alpha = Color.alpha(p);
                r = 100 + r;
                g = 100 + g;
                b = 100 + b;
                alpha = 100 + alpha;
                operation.setPixel(i, j, Color.argb(alpha, r, g,
b));
            }
        }
        imgView.setImageBitmap(operation);
    }
});

grayBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        operation =
Bitmap.createBitmap(bmp.getWidth(), bmp.getHeight(),

```

```
        bmp.getConfig());
        double x=0.33;
        for (int i = 0; i < bmp.getWidth(); i++) {
            for (int j = 0; j < bmp.getHeight(); j++) {
                int p = bmp.getPixel(i, j);
                int r = Color.red(p);
                int g = Color.green(p);
                int b = Color.blue(p);
                r = ((int) (x*r)+(int) (x*g)+(int) (x*b));
                g = ((int) (x*r)+(int) (x*g)+(int) (x*b));
                b = ((int) (x*r)+(int) (x*g)+(int) (x*b));
                operation.setPixel(i, j, Color.argb(Color.alpha(p),
r, g, b));
            }
        }
        imgView.setImageBitmap(operation);
    });
}
}
```

Output:**Regular:****Darkness:**

Brightness:**Greyscale:****Conclusion:**

We have successfully performed the animation on the image and applied various effects to an image.

Practical No – 8

Aim: Android program to work with google maps and location

- 1) "Add marker" method to be used in application students are creating.
- 2) calculate route distance between two locations.

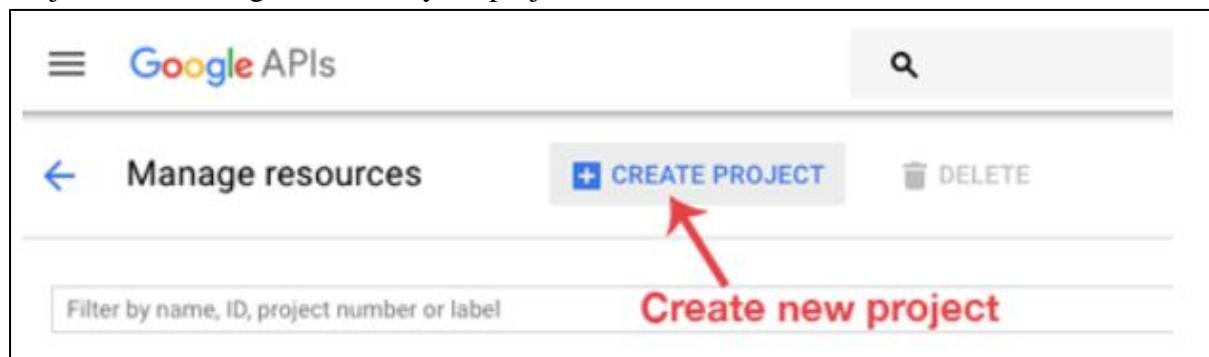
Theory:

Google Maps API:

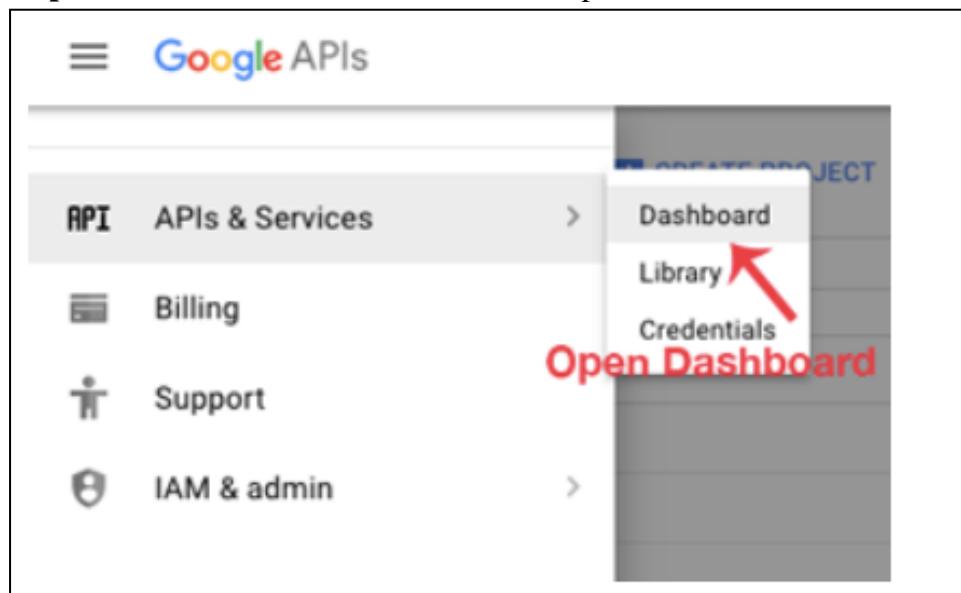
An API key is needed to access the Google Maps servers. This key is free, and you can use it with any of your applications. If you haven't created project, you can follow the below steps to get started:

Step 1: Open Google developer console and sign in with your gmail account: <https://console.developers.google.com/project>

Step 2: Now create new project. You can create new project by clicking on the Create Project button and give name to your project.



Step 3: Now click on APIs & Services and open **Dashboard** from it.

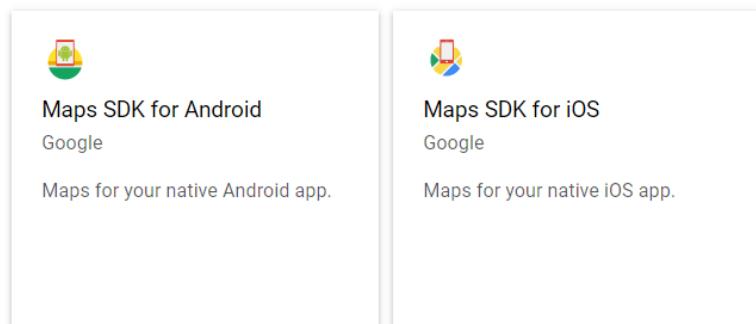


Step 4: In this open **Enable APIS AND SERVICES.**

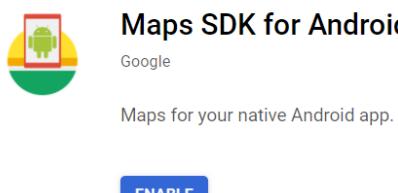
The screenshot shows the Google APIs Dashboard. On the left, there's a sidebar with 'APIs & Services' selected. The main area has a 'Dashboard' tab selected. At the top right, there's a blue button labeled 'ENABLE APIS AND SERVICES' with a red arrow pointing to it. Below the dashboard, there's a section titled 'Enabled APIs and services' with a note: 'Some APIs and services are enabled automatically'. There's also a 'Library' and 'Credentials' section.

Step 5: Now open Map SDK for Android.

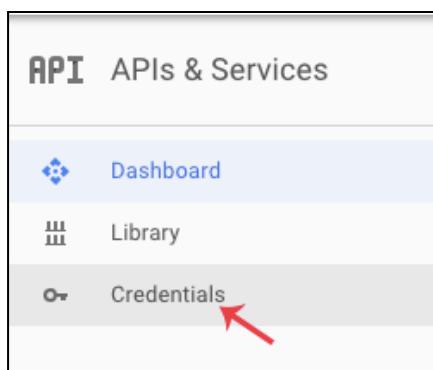
Maps



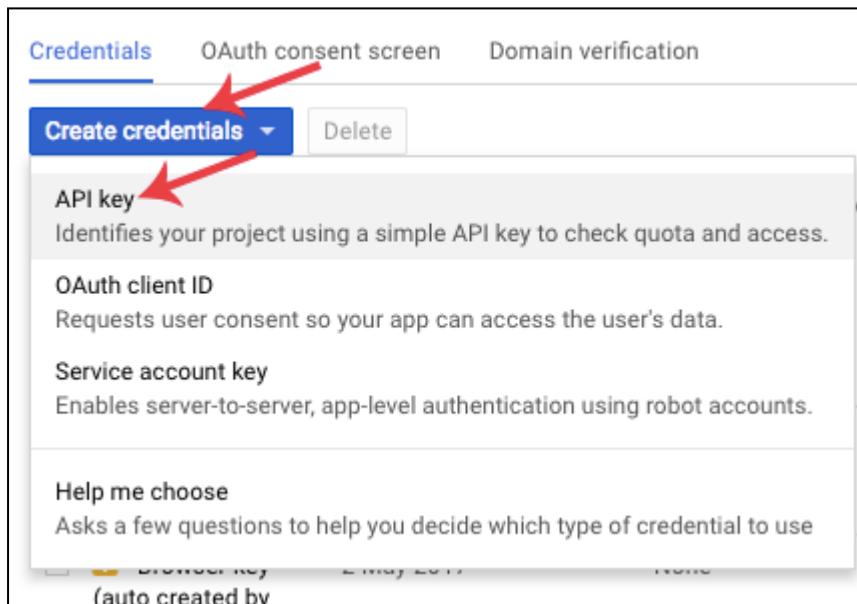
Step 6: Now enable the Google Maps Android API.



Step 7: Now go to **Credentials**



Step 8: Here click on Create credentials and choose API key



Step 9: Now API your API key will be generated. Copy it and save it somewhere as we will need it when implementing Google Map in our Android project.

API key created

Use this key in your application by passing it with the `key=API_KEY` parameter.

Your API key 

 Restrict your key to prevent unauthorized use in production.

[CLOSE](#) [RESTRICT KEY](#)

Code:**Strings:**

```
<resources>
    <string name="app_name">Practical7_1</string>
    <string name="map_key" translatable="false">AIzaSyC2Vg8vFEv9Ft2UJ6s-6SBESWuGBuT5dvw</string>
</resources>
```

Manifest:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.practical7_1">

    <meta-data android:name="com.google.android.geo.API_KEY"
        android:value="@string/map_key"/>

    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission
        android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />

    <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.Practical7_1">
        <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>
    </application>
</manifest>
```

Dependencies:

```
implementation 'com.google.android.libraries.maps:maps:3.1.0-beta'
implementation 'com.google.android.gms:play-services-maps:18.0.0'
implementation 'com.google.android.gms:play-services-location:18.0.0'
implementation 'com.google.maps.android:android-maps-utils:2.3.0'
```

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"
```

```

    android:layout_margin="20dp"
    android:orientation="vertical"
    tools:context=".MainActivity">
<fragment
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/google_map"
    android:name="com.google.android.gms.maps.SupportMapFragment"/>
</androidx.constraintlayout.widget.ConstraintLayout>

```

MainActivity.java

```

package com.example.practical7_1;

import android.Manifest;
import android.annotation.SuppressLint;
import android.content.pm.PackageManager;
import android.location.Location;
import android.os.Bundle;
import android.widget.Toast;
import com.google.android.gms.location.FusedLocationProviderClient;
import com.google.android.gms.location.LocationServices;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;

```



```

public class MainActivity extends AppCompatActivity implements
OnMapReadyCallback{
    Location currentLocation;
    GoogleMap gMap;
    FusedLocationProviderClient fusedLocationProviderClient;
    private static final int REQUEST_CODE = 101;

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

        fusedLocationProviderClient =
LocationServices.getFusedLocationProviderClient(this);
        fetchlastLocation();
    }
    private void fetchlastLocation() {
        if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED &&
ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(this, new String[]

```

```

{Manifest.permission.ACCESS_FINE_LOCATION}, REQUEST_CODE);
    return;
}
Task<Location> task =
fusedLocationProviderClient.getLastLocation();
task.addOnSuccessListener(new OnSuccessListener<Location>() {
    @Override
    public void onSuccess(Location location) {
        if(location != null){
            currentLocation = location;

Toast.makeText(getApplicationContext(),currentLocation.getLatitude()
                +" "+currentLocation.getLongitude(),
Toast.LENGTH_SHORT).show();
        SupportMapFragment supportMapFragment =
(SupportMapFragment)
getSupportFragmentManager().findFragmentById(R.id.google_map);
        supportMapFragment.getMapAsync(MainActivity.this);
    }
})
});
}

@Override
public void onMapReady(@NonNull GoogleMap googleMap) {
    double srcLat = currentLocation.getLatitude();
    double srcLong = currentLocation.getLongitude();
    LatLng latLng = new
LatLng(currentLocation.getLatitude(),currentLocation.getLongitude());
    MarkerOptions markerOptions = new MarkerOptions().position(latLng)
        .title(srcLat + ":" + srcLong);
    googleMap.animateCamera(CameraUpdateFactory.newLatLng(latLng));

googleMap.animateCamera(CameraUpdateFactory.newLatLngZoom(latLng,10));
    googleMap.addMarker(markerOptions);

    gMap = googleMap;
    gMap.setOnMapClickListener(new GoogleMap.OnMapClickListener() {
        int count = -1;
        @Override
        public void onMapClick(@NonNull LatLng latLng) {
            MarkerOptions markerOptions1 = new MarkerOptions();
            markerOptions1.position(latLng);
            markerOptions1.title(latLng.latitude+ ":" +
latLng.longitude);
            count++;
            if(count % 2 != 0){
                gMap.clear();
            }
            gMap.animateCamera(CameraUpdateFactory.newLatLng(latLng));

gMap.animateCamera(CameraUpdateFactory.newLatLngZoom(latLng,10));
            gMap.addMarker(markerOptions1);
            double res =
distance(srcLat,latLng.latitude,srcLong,latLng.longitude);
            Toast.makeText(getApplicationContext(), "Distance = "+ res
, Toast.LENGTH_SHORT).show();
        }
    });
}
}

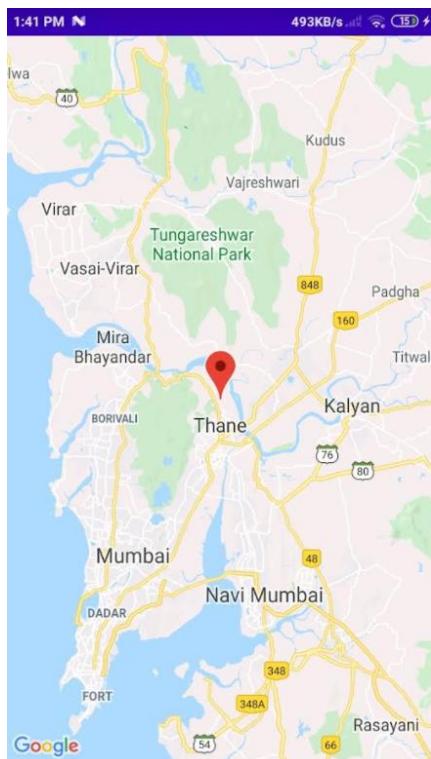
```

```
public static double distance(double lat1, double lat2, double lon1,
double lon2)
{
    lon1 = Math.toRadians(lon1);
    lon2 = Math.toRadians(lon2);
    lat1 = Math.toRadians(lat1);
    lat2 = Math.toRadians(lat2);
    double dlon = lon2 - lon1;
    double dlat = lat2 - lat1;
    double a = Math.pow(Math.sin(dlat / 2), 2)
               + Math.cos(lat1) * Math.cos(lat2)
               * Math.pow(Math.sin(dlon / 2), 2);
    double c = 2 * Math.asin(Math.sqrt(a));
    double r = 6371;
    return(c * r);
}

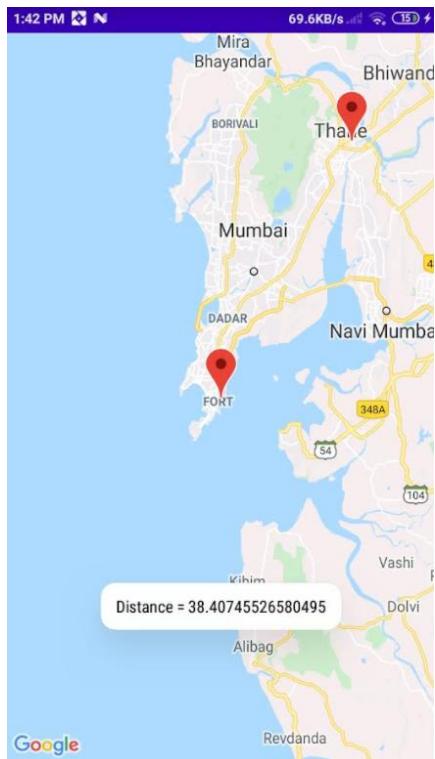
@SuppressLint("MissingSuperCall")
@Override
public void onRequestPermissionsResult(int requestCode, @NonNull
String[] permissions, @NonNull int[] grantResults) {
    switch (requestCode) {
        case REQUEST_CODE:
            if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED);
                fetchlastLocation();
                break;
    }
}
}
```

Output:

By Default, marker is placed at current location.



After marking another location, the distance between two locations is displayed in toast message.



Conclusion:

Successfully implemented marker method and calculated distance between two locations.

Practical No – 9

Aim: To Write a program to record and play audio, play video and use GPS

Theory:

Audio Player:

Android has a built-in microphone through which you can capture audio and store it , or play it in your phone. There are many ways to do that but the most common way is through the MediaRecorder class.

Android provides a MediaRecorder class to record audio or video. In order to use the MediaRecorder class ,you will first create an instance of the MediaRecorder class. Its syntax is given below.

```
MediaRecorder myAudioRecorder = new MediaRecorder();
```

To set the source , output and encoding format and output file. The syntax is given below.

```
myAudioRecorder.set AudioSource(MediaRecorder.AudioSource.MIC);
myAudioRecorder.set OutputFormat(MediaRecorder.OutputFormat.THREE_GPP);
myAudioRecorder.set AudioEncoder(MediaRecorder.OutputFormat.AMR_NB);
myAudioRecorder.set outputFile(outputFile);
```

After specifying the audio source and format and its output file, we can then call the two basic methods: prepare and start to start recording the audio.

```
myAudioRecorder.prepare();
myAudioRecorder.start();
```

Apart from these methods , there are other methods listed in the MediaRecorder class that allow you to have more control over audio and video recording.

Video Player:

By the help of MediaController and VideoView classes, we can play the video files in android. It can load images from various sources taking care of computing its measurement from the video so that it can be used for any layout manager, providing display options such as scaling and tinting.

MediaController class

The android.widget.MediaController is a view that contains media controls like play/pause, previous, next, fast-forward, rewind etc.

VideoView class

The android.widget.VideoView class provides methods to play and control the video player. The commonly used methods of VideoView class are as follows:

To initiate a video view :

```
VideoView simpleVideoView = (VideoView)
findViewById(R.id.simpleVideoView);
simpleVideoView.setVideoURI(Uri.parse("android.resource://" +
getPackageName()+"."+R.raw.videoname));
```

setVideoUri(Uri uri): This method is used to set the absolute path of the video file which is going to be played. This method takes a Uri object as an argument.

GPS

Get the last known location

Once you have created the Location Services client you can get the last known location of a user's device. When your app is connected to these you can use the fused location provider's getLastLocation() method to retrieve the device location.

The precision of the location returned by this call is determined by the permission setting you put in your app manifest, as described in the guide on how to request location permissions.

To request the last known location, call the getLastLocation() method.

getLastLocation() gets a location estimate more quickly and minimizes battery usage that can be attributed to your app. However, the location information might be out of date, if no other clients have actively used location recently.

getCurrentLocation() gets a fresher, more accurate location more consistently. However, this method can cause active location computation to occur on the device

A) Write a program to

- a) Record an audio and play
- b) Play a video in Videoview.
- c) Display the current location of your device (Latitude & Longitude)

Code:**Activity_main.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"
    tools:context=".MainActivity"
    android:orientation="vertical"
    android:padding="16dp">

    <TextView
        android:id="@+id/head"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="2dp"
        android:text="AUDIO PLAYER"
        android:textAlignment="center"
        android:textSize="24sp"
        android:textStyle="bold" />

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

        <Button
            android:id="@+id/rec"
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:layout_margin="5dp"
            android:text="START RECORD"
            app:drawableTopCompat="@drawable/ic_start_recording"
            android:onClick="rec"
            tools:textAlignment="center" />

        <Button
            android:id="@+id/stop"
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:layout_margin="5dp"
            android:text="STOP RECORD"
            android:onClick="stop"
            app:drawableTopCompat="@drawable/ic_stop_recording"
            tools:textAlignment="center" />

    </LinearLayout>

```

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

    <Button
        android:id="@+id/play"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_margin="5dp"
        android:text=" PLAY AUDIO "
        app:drawableTopCompat="@drawable/ic_play_audio"
        android:onClick="play"
        tools:textAlignment="center" />

    <Button
        android:id="@+id/pause"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_margin="5dp"
        android:text=" STOP PLAY "
        android:onClick="pause"
        app:drawableTopCompat="@drawable/ic_pause_audio"
        tools:textAlignment="center" />
</LinearLayout>

<TextView
    android:id="@+id/textView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    android:text="VIDEO PLAYER"
    android:textAlignment="center"
    android:textSize="24sp"
    android:textStyle="bold" />

<FrameLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/textView">
    <VideoView
        android:id="@+id/videoView"
        android:layout_width="match_parent"
        android:layout_height="208dp" />
</FrameLayout>

<Button
    android:id="@+id/btnGetLocation"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_margin="5dp"
    android:text="Show Location"
    android:textSize="20sp" />

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

```
        android:layout_gravity="center"
        android:layout_margin="5dp"
        android:background="#FF03DAC5"
        android:hint="Location"
        android:textColor="@color/black"
        android:textAlignment="center"
        android:textSize="24sp" />

</LinearLayout>
```

MainActivity.java

```
package com.example.prac6_1;

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

import android.Manifest;
import android.content.Context;
import android.content.ContextWrapper;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.location.Location;
import android.location.LocationManager;
import android.media.MediaPlayer;
import android.media.MediaRecorder;
import android.os.Build;
import android.os.Bundle;
import android.os.Environment;
import android.provider.Settings;
import android.view.View;
import android.widget.Button;
import android.widget.MediaController;
import android.widget.TextView;
import android.widget.Toast;
import android.widget.VideoView;

import java.io.File;

public class MainActivity extends AppCompatActivity {
    private static int MICROPHONE_PERMISSION_CODE= 200;
    MediaRecorder mediaRecorder;
    MediaPlayer mediaPlayer;
    private static final int REQUEST_LOCATION = 1;
    Button btnGetLocation;
    TextView showLocation;
    LocationManager locationManager;
    String latitude, longitude;

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

```
VideoView videoView = findViewById(R.id.videoView);

videoView.setVideoPath("android.resource://" + getPackageName() + "/" + R.raw.video);
MediaController mediaController = new MediaController(this);
mediaController.setAnchorView(videoView);
ActivityCompat.requestPermissions(this, new String[] {
    Manifest.permission.ACCESS_FINE_LOCATION},
REQUEST_LOCATION);
showLocation = findViewById(R.id.showLocation);
btnGetLocation = findViewById(R.id.btnGetLocation);

btnGetLocation.setOnClickListener(new View.OnClickListener() {
    @RequiresApi(api = Build.VERSION_CODES.M)
    @Override
    public void onClick(View v) {
        locationManager = (LocationManager)
            getSystemService(Context.LOCATION_SERVICE);
        if
(!locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER)) {
            OnGPS();
        } else {
            getLocation();
        }
    }
});    videoView.setMediaController(mediaController);
}

public void rec(View v){
try{
    mediaRecorder = new MediaRecorder();
    mediaRecorder.setAudioSource(MediaRecorder.AudioSource.MIC);

mediaRecorder.setOutputFormat(MediaRecorder.OutputFormat.THREE_GPP);
    mediaRecorder.setOutputFile(getRecordingFilePath());

mediaRecorder.setAudioEncoder(MediaRecorder.AudioEncoder.AMR_NB);
    mediaRecorder.prepare();
    mediaRecorder.start();
    Toast.makeText(this,"Recording Started",
Toast.LENGTH_SHORT).show();
}
catch (Exception e){
    e.printStackTrace();
}
public void stop(View v){
    mediaRecorder.stop();
    mediaRecorder.release();
    mediaRecorder = null;
    Toast.makeText(this,"Recording Stopped",
Toast.LENGTH_SHORT).show();
}
public void play(View v){
try{
    mediaPlayer = new MediaPlayer();
    mediaPlayer.setDataSource(getRecordingFilePath());
    mediaPlayer.prepare();
    mediaPlayer.start();
    Toast.makeText(this,"Recorded Audio
Playing",Toast.LENGTH_SHORT).show();
} catch (Exception e){
```

```

        e.printStackTrace();
    }
}

public void pause(View v){
    mediaPlayer.stop();
    mediaPlayer.release();
    mediaPlayer = null;
    Toast.makeText(this,"Audio Stopped", Toast.LENGTH_SHORT).show();
}

private boolean micCheck(){

if(this.getPackageManager().hasSystemFeature(PackageManager.FEATURE_MICROPHONE)) {
    return true;
}
else{
    return false;
}
}

private void getmicpremission(){
    if (ContextCompat.checkSelfPermission(MainActivity.this,
        Manifest.permission.RECORD_AUDIO)
==PackageManager.PERMISSION_DENIED) {
        ActivityCompat.requestPermissions(MainActivity.this,new
String[]{

Manifest.permission.RECORD_AUDIO},MICROPHONE_PERMISSION_CODE);
    }
}

private String getRecordingFilePath() {
    ContextWrapper contextWrapper = new
ContextWrapper(getApplicationContext());
    File musicDirectory
=contextWrapper.getExternalFilesDir(Environment.DIRECTORY_MUSIC);
    File file = new File(musicDirectory, "testRecordingFile"+".mp3");
    return file.getPath();
}

private void OnGPS() {
    final AlertDialog.Builder builder = new AlertDialog.Builder(this);
    builder.setMessage("Enable
GPS").setCancelable(false).setPositiveButton("Yes", new
DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which)
{
            startActivity(new
Intent(Settings.ACTION_LOCATION_SOURCE_SETTINGS));
        }
    }).setNegativeButton("No", new
DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            dialog.cancel();
        }
    });
    final AlertDialog alertDialog = builder.create();
    alertDialog.show();
}

private void getLocation() {
    if (ActivityCompat.checkSelfPermission(
        MainActivity.this,Manifest.permission.ACCESS_FINE_LOCATION)

```

```
!=
        PackageManager.PERMISSION_GRANTED &&
ActivityCompat.checkSelfPermission(
            MainActivity.this,
Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
    ActivityCompat.requestPermissions(this, new
        String[]{Manifest.permission.ACCESS_FINE_LOCATION},
REQUEST_LOCATION);
} else {
    Location locationGPS =
locationManager.getLastKnownLocation(LocationManager.GPS_PROVIDER);
    if (locationGPS != null) {
        double lat = locationGPS.getLatitude();
        double longi = locationGPS.getLongitude();
        latitude = String.valueOf(lat);
        longitude = String.valueOf(longi);
        showLocation.setText("Your Location: " + "\n" + "Latitude:
" + latitude + "\n" +
                "Longitude: " + longitude);
    } else {
        Toast.makeText(this, "Unable to find location.",
Toast.LENGTH_SHORT).show();
    }
}
}
```

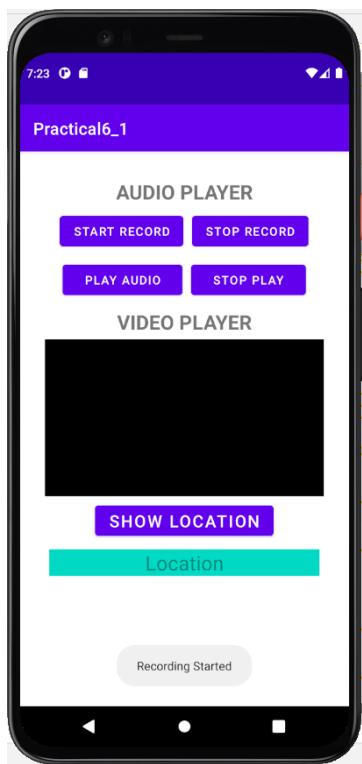
AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.prac6_1">
    <uses-permission android:name="android.permission.RECORD_AUDIO" />
    <uses-permission
        android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
        <uses-permission android:name="android.permission.STORAGE" />
        <uses-permission android:name="android.permission.INTERNET" />
        <uses-permission
            android:name="android.permission.ACCESS_FINE_LOCATION"/>
            <uses-permission
                android:name="android.permission.ACCESS_COARSE_LOCATION" />
                <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.Prac6_1">
                    <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>
                </application>
            </manifest>
```

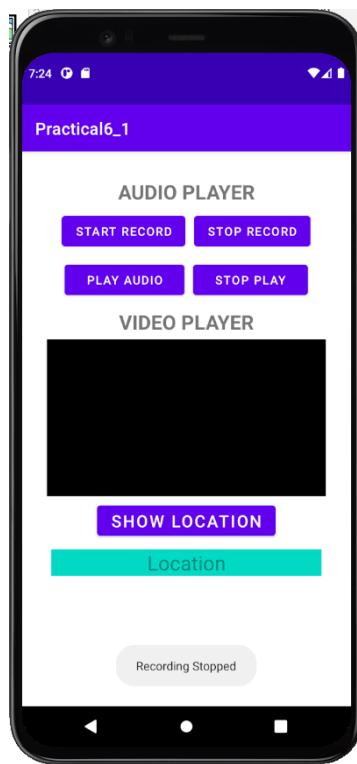
Video file stored in raw folder in res.

Output:

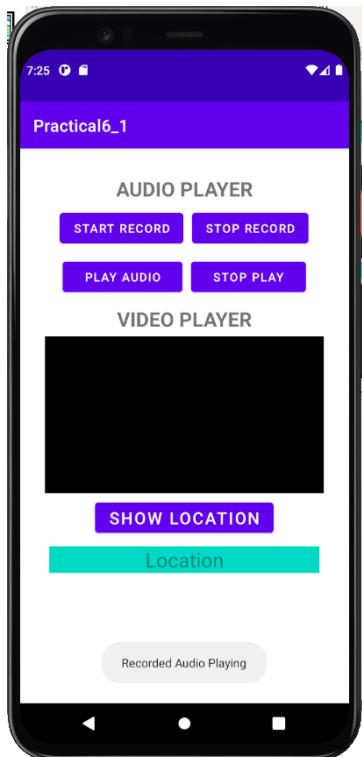
Start Record



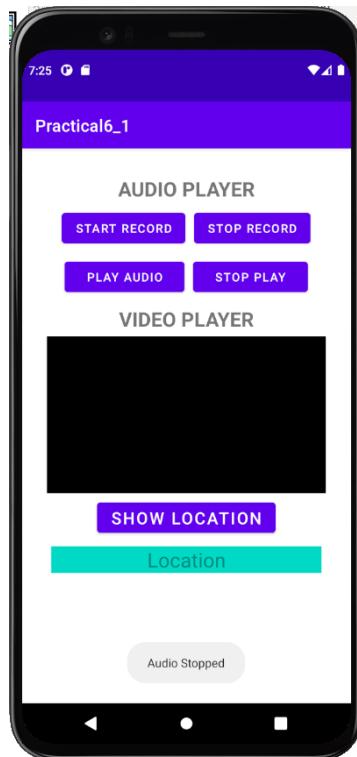
Stop Record

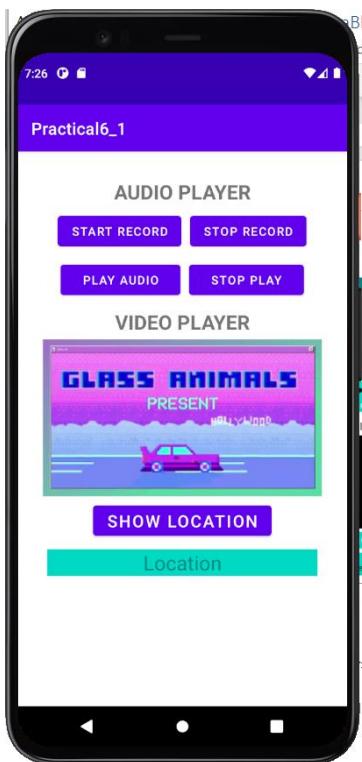
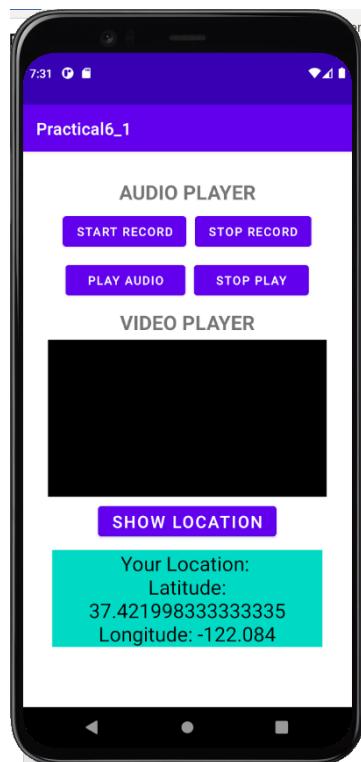


Play Audio



Stop Play



Play Video**Location****Conclusion:**

The Function of audio recording and playing, video streaming and device location in android implemented successfully.

Practical No – 10

Aim: Android program based on Rest API.

Theory:

HttpURLConnection:

A URLConnection with support for HTTP-specific features. See the spec for details.

Uses of this class follow a pattern:

1. Obtain a new HttpURLConnection by calling URL.openConnection() and casting the result to HttpURLConnection.
2. Prepare the request. The primary property of a request is its URI. Request headers may also include metadata such as credentials, preferred content types, and session cookies.
3. Optionally upload a request body. Instances must be configured with setDoOutput(true) if they include a request body. Transmit data by writing to the stream returned by URLConnection.getOutputStream().
4. Read the response. Response headers typically include metadata such as the response body's content type and length, modified dates and session cookies. The response body may be read from the stream returned by URLConnection.getInputStream(). If the response has no body, that method returns an empty stream.
5. Disconnect. Once the response body has been read, the HttpURLConnection should be closed by calling disconnect(). Disconnecting releases the resources held by a connection so they may be closed or reused.

For example, to retrieve the webpage at <http://www.android.com/>:

```
URL url = new URL("http://www.android.com/");
HttpURLConnection urlConnection = (HttpURLConnection) url.openConnection();
try {
    InputStream in = new BufferedInputStream(urlConnection.getInputStream());
    readStream(in);
} finally {
    urlConnection.disconnect();
}
```

JsonObject:

JsonObject class represents an immutable JSON object value (an unordered collection of zero or more name/value pairs). It also provides unmodifiable map view to the JSON object name/value mappings. A JsonObject instance can be created from an input source using JsonReader.readObject().

For example:

```
JsonReader jsonReader = Json.createReader(...);
JsonObject object = jsonReader.readObject();
jsonReader.close();
```

It can also be built from scratch using a JsonObjectBuilder. For example 1: An empty JSON object can be built as follows:

```
JsonObject object = Json.createObjectBuilder().build();
```

JsonObject values can be JsonObject, JSONArray, JsonString, JsonNumber, JsonValue.TRUE, JsonValue.FALSE, JsonValue.NULL. These values can be accessed using various accessor methods.

In the above example 2, "John" can be got using

```
String firstName = object.getString("firstName");
```

This map object provides read-only access to the JSON object data, and attempts to modify the map, whether direct or via its collection views, result in an UnsupportedOperationException.

The map object's iteration ordering is based on the order in which name/value pairs are added to the corresponding builder or the order in which name/value pairs appear in the corresponding stream.

Volley:

Volley is an HTTP library that makes networking for Android apps easier and most importantly, faster. Volley is available on GitHub.

Volley offers the following benefits:

1. Automatic scheduling of network requests.
2. Multiple concurrent network connections.
3. Transparent disk and memory response caching with standard HTTP cache coherence.
4. Support for request prioritization.
5. Cancellation request API. You can cancel a single request, or you can set blocks or scopes of requests to cancel.
6. Ease of customization, for example, for retry and backoff.
7. Strong ordering that makes it easy to correctly populate your UI with data fetched asynchronously from the network.
8. Debugging and tracing tools.

Volley excels at RPC-type operations used to populate a UI, such as fetching a page of search results as structured data. It integrates easily with any protocol and comes out of the box with support for raw strings, images, and JSON. By providing built-in support for the features you need, Volley frees you from writing boilerplate code and allows you to concentrate on the logic that is specific to your app.

Classes in Volley Library:

1. Request Queue: It is the interest one uses for dispatching requests to the network. One can make a request queue on demand if required, but typically it is created early on, at startup time, and keep it around and use it as a Singleton.
2. Request: All the necessary information for making web API call is stored in it. It is the base for creating network requests(GET, POST).

Retrofit:

Retrofit is a type-safe HTTP client for Android and Java – developed by Square (Dagger, Okhttp). Retrofit is a type-safe REST client for Android, Java and Kotlin developed by Square. The library provides a powerful framework for authenticating and interacting with APIs and sending network requests with OkHttp. See this guide to understand how OkHttp works. This library makes downloading JSON or XML data from a web API fairly straightforward. Once the data is downloaded then it is parsed into a Plain Old Java Object (POJO) which must be defined for each "resource" in the response.

```
<dependency>
<groupId>com.squareup.retrofit2</groupId>
<artifactId>retrofit</artifactId>
<version>2.3.0</version>
</dependency>
<dependency>
<groupId>com.squareup.retrofit2</groupId>
<artifactId>converter-gson</artifactId>
<version>2.3.0</version>
</dependency>
```

Retrofit models REST endpoints as Java interfaces, making them very simple to understand and consume. Retrofit works by modeling over a base URL and by making interfaces return the entities from the REST endpoint. For simplicity purposes we're going to take a small part of the JSON by modeling our User class that is going to take the values when we have received them.

Code:

- A. Create a basic application that allows you to download HTML from a given web page using HttpURLConnection. (add internet permission)**

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"
    android:orientation="vertical"
    android:layout_margin="15dp"
    tools:context=".MainActivity">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        android:layout_margin="10dp">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical">
            <EditText
                android:id="@+id/http_url_editor"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:hint="Input a web page url to get." />
            <Button
                android:id="@+id/http_url_request_button"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:text="Download Page"/>
        </LinearLayout>
        <ScrollView
            android:layout_width="match_parent"
            android:layout_height="match_parent">
            <TextView
                android:id="@+id/http_url_response_text_view"
                android:layout_width="match_parent"
                android:layout_height="wrap_content" />
        </ScrollView>
    </LinearLayout>
</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java

```
package com.example.practical8_1;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.os.Handler;
import android.os.Message;
import android.text.TextUtils;
import android.util.Log;
import android.view.View;
```

```
import android.webkit.URLUtil;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;

public class MainActivity extends AppCompatActivity {
    private static final String TAG_HTTP_URL_CONNECTION =
"HTTP_URL_CONNECTION";
    private static final int REQUEST_CODE_SHOW_RESPONSE_TEXT = 1;
    private static final String KEY_RESPONSE_TEXT = "KEY_RESPONSE_TEXT";
    private static final String REQUEST_METHOD_GET = "GET";
    private EditText requestUrlEditor = null;
    private Button requestUrlButton = null;
    private TextView responseTextView = null;
    private Handler uiUpdater = null;

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

        initControls();
        requestUrlButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String reqUrl = requestUrlEditor.getText().toString();
                if(!TextUtils.isEmpty(reqUrl)) {
                    if(URLUtil.isHttpsUrl(reqUrl) ||
URLUtil.isHttpsUrl(reqUrl)) {
                        startSendHttpRequestThread(reqUrl);
                    } else {
                        Toast.makeText(getApplicationContext(), "The
request url is not a valid http or https url.", Toast.LENGTH_LONG).show();
                    }
                } else {
                    Toast.makeText(getApplicationContext(), "The request
url can not be empty.", Toast.LENGTH_LONG).show();
                }
            }
        });
    }

    private void initControls()
    {
        if(requestUrlEditor == null) {
            requestUrlEditor =
(EditText)findViewById(R.id.http_url_editor);
        }
        if(requestUrlButton == null) {
            requestUrlButton =
(Button)findViewById(R.id.http_url_request_button);
        }
    }
}
```

```

        if(responseTextView == null) {
            responseTextView =
(TextView)findViewById(R.id.http_url_response_text_view);
        }
        {
            uiUpdater = new Handler()
            {
                @Override
                public void handleMessage(Message msg) {
                    if(msg.what == REQUEST_CODE_SHOW_RESPONSE_TEXT)
                    {
                        Bundle bundle = msg.getData();
                        if(bundle != null)
                        {
                            String responseText =
bundle.getString(KEY_RESPONSE_TEXT);
                            responseTextView.setText(responseText);
                        }
                    }
                }
            };
        }
    }

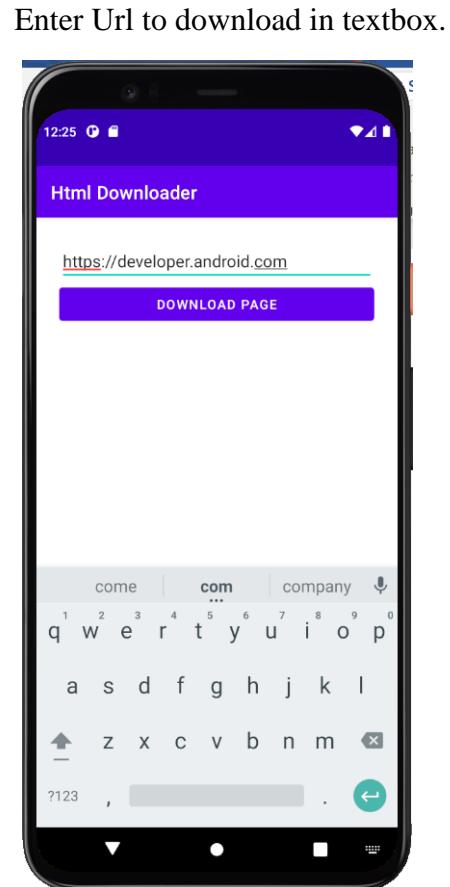
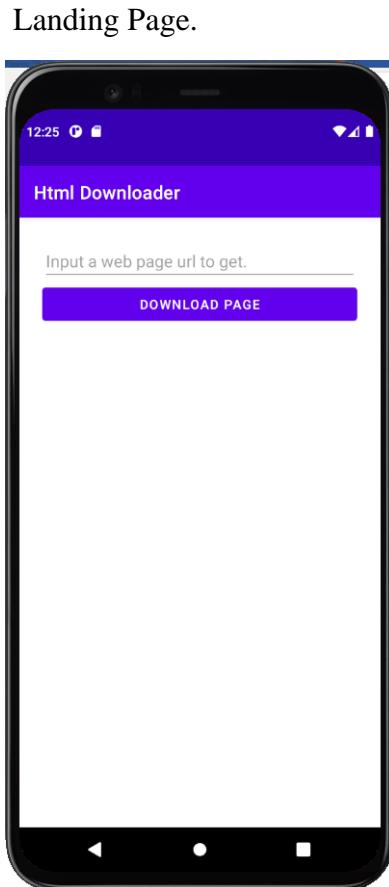
private void startSendHttpRequestThread(final String reqUrl)
{
    Thread sendHttpRequestThread = new Thread()
    {
        @Override
        public void run() {
            HttpURLConnection httpConn = null;
            InputStreamReader isReader = null;
            BufferedReader bufReader = null;
            StringBuffer readTextBuf = new StringBuffer();
            try {
                URL url = new URL(reqUrl);
                httpConn = (HttpURLConnection)url.openConnection();
                httpConn.setRequestMethod(REQUEST_METHOD_GET);
                httpConn.setConnectTimeout(10000);
                httpConn.setReadTimeout(10000);
                InputStream inputStream = httpConn.getInputStream();
                isReader = new InputStreamReader(inputStream);
                bufReader = new BufferedReader(isReader);
                String line = bufReader.readLine();
                while(line != null) {
                    readTextBuf.append(line);
                    line = bufReader.readLine();
                }
                Message message = new Message();
                message.what = REQUEST_CODE_SHOW_RESPONSE_TEXT;
                Bundle bundle = new Bundle();
                bundle.putString(KEY_RESPONSE_TEXT,
readTextBuf.toString());
                message.setData(bundle);
                uiUpdater.sendMessage(message);
            }catch(MalformedURLException ex) {
                Log.e(TAG_HTTP_URL_CONNECTION, ex.getMessage(), ex);
            }catch(IOException ex) {
                Log.e(TAG_HTTP_URL_CONNECTION, ex.getMessage(), ex);
            }finally {
                try {

```

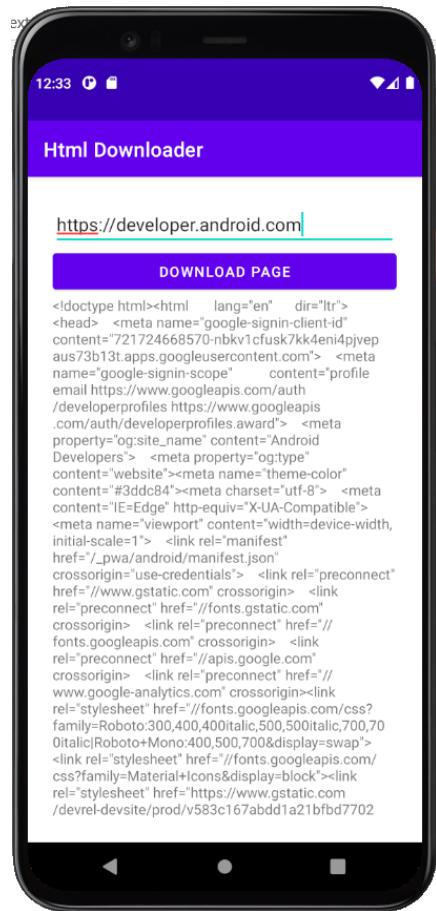
```
        if (bufReader != null) {
            bufReader.close();
            bufReader = null;
        }
        if (isReader != null) {
            isReader.close();
            isReader = null;
        }
        if (httpConn != null) {
            httpConn.disconnect();
            httpConn = null;
        }
    }catch (IOException ex) {
        Log.e(TAG_HTTP_URL_CONNECTION, ex.getMessage(),
ex);
    }
}
}
}
}
}
sendHttpRequestThread.start();
}
}
```

Permission:

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

Output:

Click on “Download Page” button to get response.



B. Create an application to parse the data using JSONObject methods and set it in the Text View's. (employee name and salary stored in JSON format). (add internet permission and change compile and target sdk to 31)

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"
    android:orientation="vertical"
    android:layout_margin="20sp"
    tools:context=".MainActivity">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">
        <TextView
            android:id="@+id/name"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Name"
            android:textSize="20sp" />
        <TextView
            android:id="@+id/salary"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Salary"
            android:textSize="20sp" />
    </LinearLayout>
</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java

```
package com.example.practical8_2;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.TextView;
import org.json.JSONException;
import org.json.JSONObject;

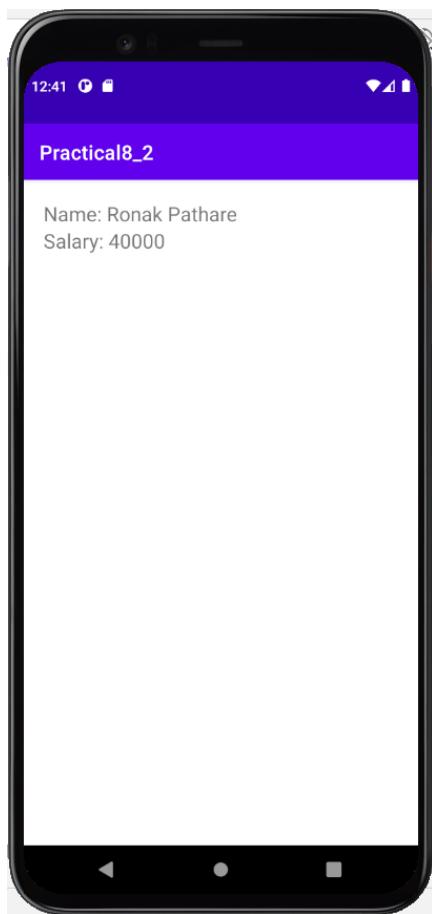
public class MainActivity extends AppCompatActivity {
    String JSON_STRING = "{\"employee\":{\"name\":\"Ronak Pathare\"},\"salary\":40000}";
    String name, salary;
    TextView employeeName, employeeSalary;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        employeeName = (TextView) findViewById(R.id.name);
        employeeSalary = (TextView) findViewById(R.id.salary);
        try {
```

```
JSONObject obj = new JSONObject(JSON_STRING);
JSONObject employee = obj.getJSONObject("employee");

name = employee.getString("name");
salary = employee.getString("salary");

employeeName.setText("Name: "+name);
employeeSalary.setText("Salary: "+salary);
} catch (JSONException e) {
    e.printStackTrace();
}
}
```

Output:

**C. Write a basic application to (use volley library), create a button and on click of the button a HTTP request will be send to server. The response from the server is then displayed using Toast on the screen. (implementation
'com.android.volley:volley:1.2.1' to be added in app gradle, create your own mocky link, add internet persmission)**

Code:

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"
    android:layout_margin="20sp"
    tools:context=".MainActivity">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">
        <Button
            android:id="@+id/buttonRequest"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Fetch Response" />
    </LinearLayout>
</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java

```
package com.example.practical8_3;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;

public class MainActivity extends AppCompatActivity {
    private static final String TAG = MainActivity.class.getName();
    private Button btnRequest;
    private RequestQueue mRequestQueue;
    private StringRequest mStringRequest;
    private String url = "https://run.mocky.io/v3/bd3ba8eb-958c-41dc-993c-4412edc2439c";
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
```

```

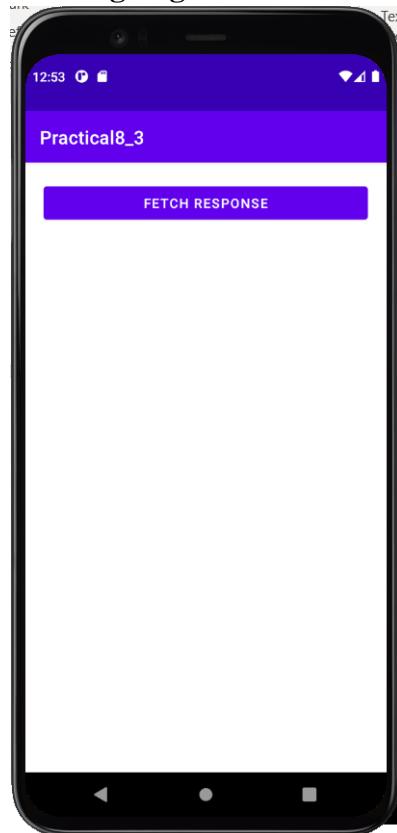
        setContentView(R.layout.activity_main);
        btnRequest = (Button) findViewById(R.id.buttonRequest);

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

    private void sendAndRequestResponse() {

        mRequestQueue = Volley.newRequestQueue(this);
        mStringRequest = new StringRequest(Request.Method.GET, url, new
Response.Listener<String>() {
            @Override
            public void onResponse(String response) {
                Toast.makeText(getApplicationContext(), "Response :" +
response.toString(), Toast.LENGTH_LONG).show();
            }
        }, new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError error) {
                Log.i(TAG, "Error :" + error.toString());
            }
        });
        mRequestQueue.add(mStringRequest);
    }
}

```

Output:**Landing Page.****Click on “Fetch Response” button**

D. Write a program to create sign up page design with 3 Edit Texts(name, email and password) and one sign up Button.On click of button , data in EditText is validated. Implement signup api to save the data in database. After getting response from api display the message on the screen by using a Toast(useRetrofit).

(**android:usesCleartextTraffic="true"** and internet permission in manifest, change url in controller, implementation 'com.squareup.retrofit2:retrofit:2.9.0'
implementation 'com.squareup.retrofit2:converter-gson:2.9.0'
add dependency)

Code:

```
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"
    android:orientation="vertical"
    android:layout_margin="20sp"
    tools:context=".MainActivity">
    <LinearLayout
        android:layout_width="match_parent"
        android:orientation="vertical"
        android:layout_height="match_parent">
        <EditText
            android:id="@+id/t1"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="name"
            android:inputType="textPersonName"/>
        <EditText
            android:id="@+id/t2"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="username"
            android:inputType="textEmailAddress"/>
        <EditText
            android:id="@+id/t3"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="password"
            android:inputType="textPassword"/>
        <Button
            android:id="@+id/SignUpBtn"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Sign up"/>
    </LinearLayout>
</androidx.constraintlayout.widget.ConstraintLayout>
```

Signup.php

```
<?php
$conn = mysqli_connect("localhost","root","","apidb");

$name=trim($_POST['name']);
$email=trim($_POST['email']);
$pass=trim($_POST['password']);

if($conn)
{
$qry = "select * from tb_user where email='$email'";
$result = mysqli_query($conn,$qry);

if(mysqli_num_rows($result)>0)
{
    $response['message']="exist";
    echo json_encode($response);
}
else
{
    $qry = "insert into tb_user(name,email,password)
values('$name','$email','$pass')";
    if(mysqli_query($conn,$qry))
    {
        $response['message']="success";
    }
    else{
        $response['message']="failed";
    }
}

echo json_encode($response);
}
}
else{
    $response['message']="failed";
    echo json_encode($response);
}
?>
```

ResponseModel.java

```
package com.example.practical8_4;

public class ResponseModel {
    String message;

    public ResponseModel(String message) {
        this.message = message;
    }
    public ResponseModel() {}

    public String getMessage() {
        return message;
    }
    public void setMessage(String message) {
        this.message = message;
    }
}
```

```
}
```

Apiset.java

```
package com.example.practical8_4;
import retrofit2.Call;
import retrofit2.http.Field;
import retrofit2.http.FormUrlEncoded;
import retrofit2.http.POST;

public interface apiset {
    @FormUrlEncoded
    @POST("signup.php")
    Call<ResponseModel> verifyuser(
        @Field("name") String name,
        @Field("email") String email,
        @Field("password") String password
    );
}
```

Controller.java

```
package com.example.practical8_4;

import retrofit2.Retrofit;
import retrofit2.converter.gson.GsonConverterFactory;

public class Controller {
    private static final String url="http://192.168.131.1/api/";
    private static Controller clientobject;
    private static Retrofit retrofit;

    Controller() {
        retrofit=new Retrofit.Builder()
            .baseUrl(url)
            .addConverterFactory(GsonConverterFactory.create())
            .build();
    }

    public static synchronized Controller getInstance() {
        if(clientobject==null)
            clientobject=new Controller();
        return clientobject;
    }

    apiset getapi() {
        return retrofit.create(apiset.class);
    }
}
```

MainActivity.java

```
package com.example.practical8_4;

import androidx.appcompat.app.AppCompatActivity;
import android.graphics.Color;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

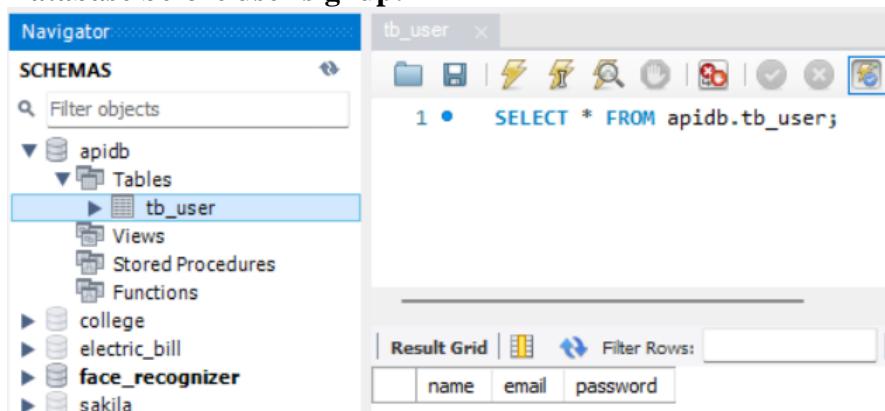
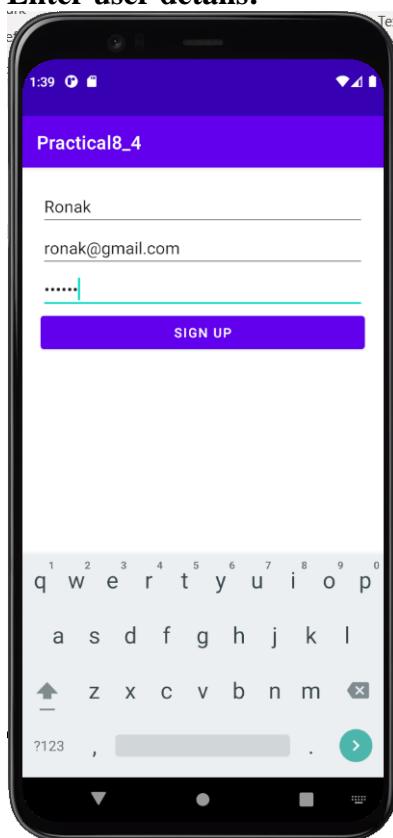
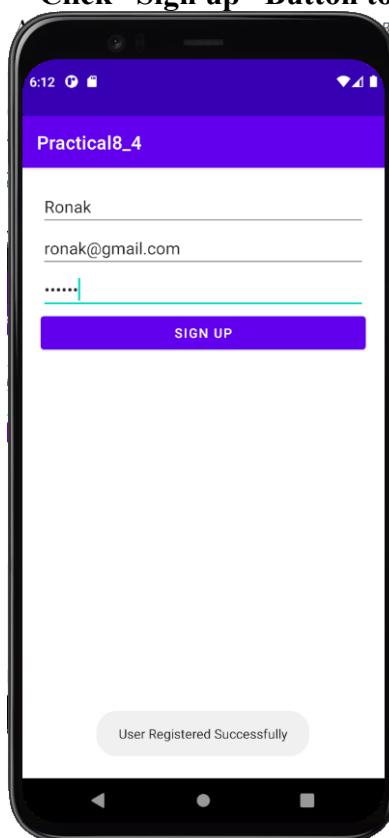
public class MainActivity extends AppCompatActivity {
    EditText t1,t2,t3;
    Button SignUpBtn;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        t1=(EditText)findViewById(R.id.t1);
        t2=(EditText)findViewById(R.id.t2);
        t3=(EditText)findViewById(R.id.t3);
        SignUpBtn=(Button)findViewById(R.id.SignUpBtn);
        SignUpBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                processSignUp();
            }
        });
    }
    void processSignUp(){
        String name = t1.getText().toString();
        String email = t2.getText().toString();
        String password = t3.getText().toString();
        Call<ResponseModel> call = Controller
            .getInstance()
            .getapi()
            .verifyuser(name,email,password);
        call.enqueue(new Callback<ResponseModel>() {
            @Override
            public void onResponse(Call<ResponseModel> call,
Response<ResponseModel> response) {
                ResponseModel obj=response.body();
                String output=obj.getMessage();
                if(output.equals("exist")){
                    t1.setText("");
                    t2.setText("");
                    t3.setText("");
                    Toast.makeText(getApplicationContext(),"User Already Exist", Toast.LENGTH_LONG).show();
                }
                if(output.equals("success")){
                    Toast.makeText(getApplicationContext(),"User Registered Successfully", Toast.LENGTH_LONG).show();
                }
                if(output.equals("failed")){

```

```

        Toast.makeText(getApplicationContext(),"Something went
wrong", Toast.LENGTH_LONG).show();
    }
    if(output.equals("failed")) {
        Toast.makeText(getApplicationContext(),"Connection
failure", Toast.LENGTH_LONG).show();
    }
}
@Override
public void onFailure(Call<ResponseModel> call, Throwable t) {
    Toast.makeText(getApplicationContext(),"Something went
wrong!!!", Toast.LENGTH_LONG).show();
}
});}}

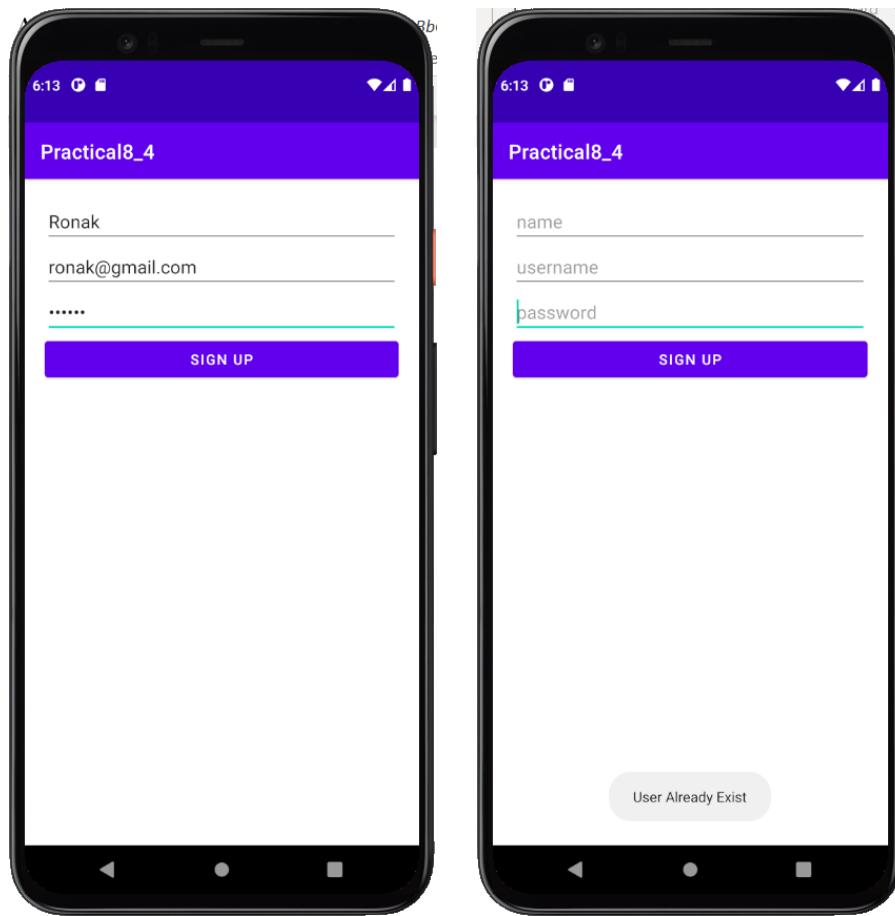
```

Output:**Database before user signup:****Enter user details:****Click “Sign up” Button to Register**

Database after registering user:

The screenshot shows a MySQL query results page. At the top, a green bar displays the message: "Showing rows 0 - 1 (2 total, Query took 0.0006 seconds.)". Below this, the SQL query "SELECT * FROM `tb_user`" is shown. A toolbar below the query includes options: Profiling, Edit inline, Edit, Explain SQL, Create PHP code, and Refresh. Further down are buttons for Show all, Number of rows (set to 25), Filter rows, and a search bar. The tb_user table has three columns: name, email, and password. One row is displayed: Ronak, ronak@gmail.com, and 123456.

name	email	password
Ronak	ronak@gmail.com	123456

Trying to register already existing user:**Conclusion:**

Hence we successfully implemented Rest API.

Practical No – 11

Aim: Flutter program using layout, widget and state management.

Theory:

Flutter:

- Flutter is a free and open-source mobile UI framework created by Google and released in May 2017.
- This means that you can use one programming language and one codebase to create two different apps (for iOS and Android).
- Flutter consists of two important parts:
 1. An SDK (Software Development Kit): A collection of tools that are going to help you develop your applications. This includes tools to compile your code into native machine code (code for iOS and Android).
 2. A Framework (UI Library based on widgets): A collection of reusable UI elements (buttons, text inputs, sliders, and so on) that you can personalize for your own needs.
- To develop with Flutter, you will use a programming language called Dart. Dart focuses on front-end development, and you can use it to create mobile and web applications.

Layouts in Flutter:

- The main concept of the layout mechanism is the widget.
- We know that flutter assume everything as a widget.
- So, the image, icon, text, and even the layout of your app are all widgets.
- Here, some of the things you do not see on your app UI, such as rows, columns, and grids that arrange, constrain, and align the visible widgets are also the widgets.
- Flutter allows us to create a layout by composing multiple widgets to build more complex widgets.
- We can categories the layout widget into two types:
 1. Single Child Widget: The single child layout widget is a type of widget, which can have only one child widget inside the parent layout widget. These widgets can also contain special layout functionality. Flutter provides us many single child widgets to make the app UI attractive. If we use these widgets appropriately, it can save our time and makes the app code more readable.
 2. Multiple Child Widget: The multiple child widgets are a type of widget, which contains more than one child widget, and the layout of these widgets are unique. For example, Row widget laying out of its child widget in a horizontal direction, and Column widget laying out of its child widget in a vertical direction. If we combine the Row and Column widget, then it can build any level of the complex widget.

Flutter State Management:

- The widget can be classified into two categories, one is a Stateless widget, and another is a Stateful widget.
- The Stateless widget does not have any internal state.
- A Stateful widget is dynamic and has a state. It means we can modify it easily throughout its lifecycle without reinitialized it again.
- A state is information that can be read when the widget is built and might change or modified over a lifetime of the app.
- If you want to change your widget, you need to update the state object, which can be done by using the `setState()` function available for Stateful widgets.
- In Flutter, the state management categorizes into two conceptual types, which are given below:
 1. Ephemeral State: It is a type of state which is related to the specific widget, or you can say that it is a state that contains in a single widget. In this kind of state, you do not need to use state management techniques. The common example of this state is Text Field.
 2. App State: It is different from the ephemeral state. It is a type of state that we want to share across various parts of our app and want to keep between user sessions. Thus, this type of state can be used globally.

Code:

- A. Create a widget `ProductBox` that contains the details of the product, such as image, name, price, and description. In the `ProductBox` widget, we use the following child widgets: Container, Row, Column, Expanded, Card, Text, Image, etc. This widget contains the following layout:**
(make changes in `widget_test runApp` line)

main.dart

```
import 'package:flutter/material.dart';
void main() {
  runApp(MyApp());
}
class MyApp extends StatelessWidget {
  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: MyHomePage(title: 'Product layout demo home page'),
    );
  }
}

class MyHomePage extends StatelessWidget {
  MyHomePage({Key? key, required this.title}) : super(key: key);
  final String title;
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text("Product Listing")),
      body: ListView(
        shrinkWrap: true,
        padding: const EdgeInsets.fromLTRB(2.0, 10.0, 2.0, 10.0),
        children: <Widget>[
          ProductBox(
            name: "iPhone",
            description: "Your new superpower.",
            price: 90100,
            image: "iPhone.png"
          ),
          ProductBox(
            name: "Pixel",
            description: "Packed with all the things you want most in a
phone.",
            price: 65000,
            image: "Pixel.jpg"
          ),
          ProductBox(
            name: "Laptop",
            description: "Republic of Gamers",
            price: 270000,
            image: "Laptop.png"
          ),
          ProductBox(
            name: "Tablet",
            
```

```

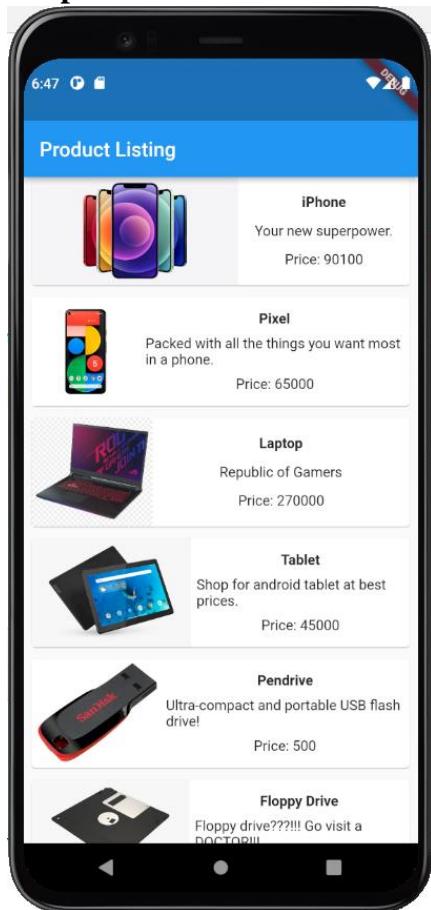
        description: "Shop for android tablet at best prices.",
        price: 45000,
        image: "Tablet.png"
    ),
    ProductBox(
        name: "Pendrive",
        description: "Ultra-compact and portable USB flash drive!",
        price: 500,
        image: "Pendrive.png"
    ),
    ProductBox(
        name: "Floppy Drive",
        description: "Floppy drive??!! Go visit a DOCTOR!!!",
        price: 1000,
        image: "Floppy.png"
    ),
),
],
)
);
}
}
}

class ProductBox extends StatelessWidget {
ProductBox({Key? key, required this.name, required this.description, required this.price, required this.image}) :super(key: key);
final String name;
final String description;
final int price;
final String image;
Widget build(BuildContext context) {
return Container(
padding: EdgeInsets.all(2),
height: 120,
child: Card(
child: Row(
mainAxisAlignment: MainAxisAlignment.spaceEvenly,
children: <Widget>[
Image.asset("assets/" + image),
Expanded(
child: Container(
padding: EdgeInsets.all(5),
child: Column(
mainAxisAlignment:
MainAxisAlignment.spaceEvenly,
children: <Widget>[
Text(
this.name, style: TextStyle(
fontWeight: FontWeight.bold
)
),
Text(this.description), Text(
"Price: " + this.price.toString()
),
]
),
)
),
]
),
);
}
}
}

```

pubsec.yml

```
assets:
  - assets/iPhone.png
  - assets/Laptop.png
  - assets/Pendrive.png
  - assets/Pixel.jpg
  - assets/Tablet.png
  - assets/Floppy.png
```

Output:**Conclusion:**

Hence we successfully flutter program using layout, widget and state.

Practical 12

AIM: Flutter program to work with SQLite Database.

Theory:

Flutter:

1. Flutter is a free and open-source mobile UI framework created by Google and released in May 2017.
2. In a few words, it allows you to create a native mobile application with only one codebase.
3. This means that you can use one programming language and one codebase to create two different apps (for iOS and Android).
- Flutter consists of two important parts:
 - An SDK (Software Development Kit): A collection of tools that are going to help you develop your applications. This includes tools to compile your code into native machine code (code for iOS and Android).
 - A Framework (UI Library based on widgets): A collection of reusable UI elements (buttons, text inputs, sliders, and so on) that you can personalize for your own needs.
 - To develop with Flutter, you will use a programming language called Dart.
 - Dart focuses on front-end development, and you can use it to create mobile and web applications.

SQLite:

1. SQLite database is the de-facto and standard SQL based embedded database engine. It is small and time-tested database engine.
2. Sqflite package provides a lot of functionality to work efficiently with SQLite database.
3. It provides standard methods to manipulate SQLite database engine. The core functionality provided by sqflite package is as follows:
 - a. Create / Open (openDatabase method) a SQLite database.
 - b. Execute SQL statement (execute method) against SQLite database.
 - c. Advanced query methods (query method) to reduce to code required to query and get information from SQLite database.

A) Write a Flutter program to work with SQLite Database.**Code:****main.dart**

```

import 'package:flutter/material.dart';
import 'car.dart';
import 'dbhelper.dart';
void main() => runApp(MyApp());
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'TutorialKart - Flutter',
      theme: ThemeData(
        primarySwatch: Colors.purple,
      ),
      home: MyHomePage(),
    );
  }
}
class MyHomePage extends StatefulWidget {
  @override
  _MyHomePageState createState() => _MyHomePageState();
}
class _MyHomePageState extends State<MyHomePage> {
  final dbHelper = DatabaseHelper.instance;
  List<Car> cars = [];
  List<Car> carsByName = [];
//controllers used in insert operation UI
  TextEditingController nameController = TextEditingController();
  TextEditingController milesController = TextEditingController();
//controllers used in update operation UI
  TextEditingController idUpdateController = TextEditingController();
  TextEditingController nameUpdateController = TextEditingController();
  TextEditingController milesUpdateController = TextEditingController();
//controllers used in delete operation UI
  TextEditingController idDeleteController = TextEditingController();
//controllers used in query operation UI
  TextEditingController queryController = TextEditingController();
  final GlobalKey<ScaffoldState> _scaffoldKey = new
  GlobalKey<ScaffoldState>();
  void _showMessageInScaffold(String message){
    _scaffoldKey.currentState!.showSnackBar(
      SnackBar(
        content: Text(message),
      )
    );
  }
  @override
  Widget build(BuildContext context) {
    return DefaultTabController(
      length: 5,
      child: Scaffold(
        key: _scaffoldKey,
        appBar: AppBar(
          bottom: TabBar(
            tabs: [
              Tab(
                text: "Insert",

```

```
),
Tab(
    text: "View",
),
Tab(
    text: "Query",
),
Tab(
    text: "Update",
),
Tab(
    text: "Delete",
),
],
),
),
title: Text('TutorialKart - Flutter SQLite Tutorial'),
),
body: TabBarView(
    children: [
        Center(
            child: Column(
                children: <Widget>[
                    Container(
                        padding: EdgeInsets.all(20),
                        child: TextField(
                            controller: nameController,
                            decoration: InputDecoration(
                                border: OutlineInputBorder(),
                                labelText: 'Car Name',
                            ),
                        ),
                    ),
                    Container(
                        padding: EdgeInsets.all(20),
                        child: TextField(
                            controller: milesController,
                            decoration: InputDecoration(
                                border: OutlineInputBorder(),
                                labelText: 'Car Miles',
                            ),
                        ),
                    ),
                    RaisedButton(
                        child: Text('Insert Car Details'),
                        onPressed: () {
                            String name = nameController.text;
                            int miles = int.parse(milesController.text);
                            _insert(name, miles);
                        },
                    ),
                ],
            ),
        ),
        Container(
            child: ListView.builder(
                padding: const EdgeInsets.all(8),
                itemCount: cars.length + 1,
                itemBuilder: (BuildContext context, int index) {
                    if (index == cars.length) {
                        return RaisedButton(
                    },
                ],
            ),
        ),
    ],
),
Container(
    child: ListView.builder(
        padding: const EdgeInsets.all(8),
        itemCount: cars.length + 1,
        itemBuilder: (BuildContext context, int index) {
            if (index == cars.length) {
                return RaisedButton(

```

```

        child: Text('Refresh'),
        onPressed: () {
            setState(() {
                _queryAll();
            });
        },
    );
}
return Container(
    height: 40,
    child: Center(
        child: Text(
            '[${cars[index].id}] ${cars[index].name} - '
            '${cars[index].miles} miles',
            style: TextStyle(fontSize: 18),
        ),
    ),
),
),
Center(
    child: Column(
        children: <Widget>[
            Container(
                padding: EdgeInsets.all(20),
                child: TextField(
                    controller: queryController,
                    decoration: InputDecoration(
                        border: OutlineInputBorder(),
                        labelText: 'Car Name',
                    ),
                    onChanged: (text) {
                        if (text.length >= 2) {
                            setState(() {
                                _query(text);
                            });
                        } else {
                            setState(() {
                                carsByName.clear();
                            });
                        }
                    },
                ),
                height: 100,
            ),
            Container(
                height: 300,
                child: ListView.builder(
                    padding: const EdgeInsets.all(8),
                    itemCount: carsByName.length,
                    itemBuilder: (BuildContext context, int index) {
                        return Container(
                            height: 50,
                            margin: EdgeInsets.all(2),
                            child: Center(
                                child: Text(
                                    '[${carsByName[index].id}] '
                                    '${carsByName[index].name} - ${carsByName[index].miles} miles',
                                    style: TextStyle(fontSize: 18),
                                ),
                            );
                    },
                ),
            );
        ],
    ),
);
}

```


car.dart:

```
import 'package:flutter/material.dart';
import 'dbhelper.dart';
class Car {
    int? id;
```

```
String? name;
int? miles;
Car(this.id, this.name, this.miles);
Car.fromMap(Map<String, dynamic> map)
  id = map['id'];
  name = map['name'];
  miles = map['miles'];
}
Map<String, dynamic> toMap() {
  return {
    DatabaseHelper.columnId: id,
    DatabaseHelper.columnName: name,
    DatabaseHelper.columnMiles: miles,
  };
}
```

dbhelper.dart:

```
import 'package:flutter/material.dart';
import 'package:path/path.dart';
import 'package:sqflite/sqflite.dart';
import 'car.dart';
class DatabaseHelper {
    static final _databaseName = "cardb.db";
    static final _databaseVersion = 1;
    static final table = 'cars_table';
    static final columnId = 'id';
    static final columnName = 'name';
    static final columnMiles = 'miles';
// make this a singleton class
    DatabaseHelper._privateConstructor();
    static final DatabaseHelper instance =
DatabaseHelper._privateConstructor();
// only have a single app-wide reference to the database
    static Database? _database;
    Future<Database?> get database async {
        if (_database != null) return _database;
// lazily instantiate the db the first time it is accessed
        _database = await _initDatabase();
        return _database;
    }
// this opens the database (and creates it if it doesn't exist)
    _initDatabase() async {
        String path = join(await getDatabasesPath(), _databaseName);
        return await openDatabase(path,
            version: _databaseVersion,
            onCreate: _onCreate);
    }
// SQL code to create the database table
    Future _onCreate(Database db, int version) async {
        await db.execute('''
CREATE TABLE $table (
$cOLUMNId INTEGER PRIMARY KEY AUTOINCREMENT,
$cOLUMNName TEXT NOT NULL,
$cOLUMNMiles INTEGER NOT NULL
)
''');
    }
// Helper methods
// Inserts a row in the database where each key in the Map is a column name
```

```

// and the value is the column value. The return value is the id of the
// inserted row.
Future<int> insert(Car car) async {
    Database? db = await instance.database;
    return await db!.insert(table, {'name': car.name, 'miles': car.miles});
}
// All of the rows are returned as a list of maps, where each map is
// a key-value list of columns.
Future<List<Map<String, dynamic>>> queryAllRows() async {
    Database? db = await instance.database;
    return await db!.query(table);
}
// Queries rows based on the argument received
Future<List<Map<String, dynamic>>> queryRows(name) async {
    Database? db = await instance.database;
    return await db!.query(table, where: "$columnName LIKE '%$name%'");
}
// All of the methods (insert, query, update, delete) can also be done
// using raw SQL commands. This method uses a raw query to give the row count.
Future<int?> queryRowCount() async {
    Database? db = await instance.database;
    return Sqflite.firstIntValue(await db!.rawQuery('SELECT COUNT(*) FROM
$table'));
}
// We are assuming here that the id column in the map is set. The other
// column values will be used to update the row.
Future<int> update(Car car) async {
    Database? db = await instance.database;
    int id = car.toMap()['id'];
    return await db!.update(table, car.toMap(), where: '$columnId = ?',
whereArgs: [id]);
}
// Deletes the row specified by the id. The number of affected rows is
// returned. This should be 1 as long as the row exists.
Future<int> delete(int id) async {
    Database? db = await instance.database;
    return await db!.delete(table, where: '$columnId = ?', whereArgs:
[id]);
}

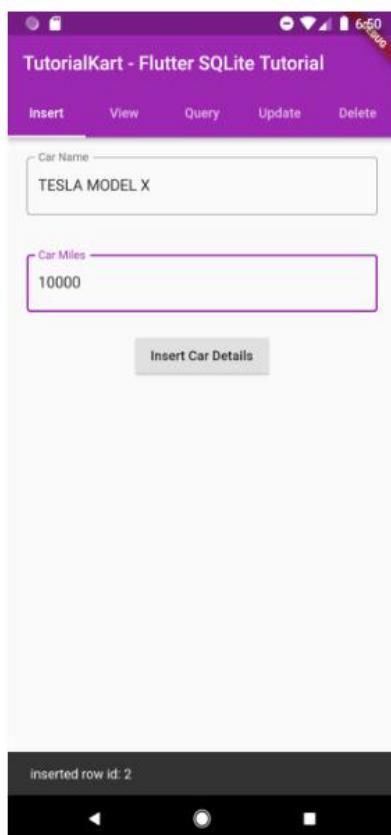
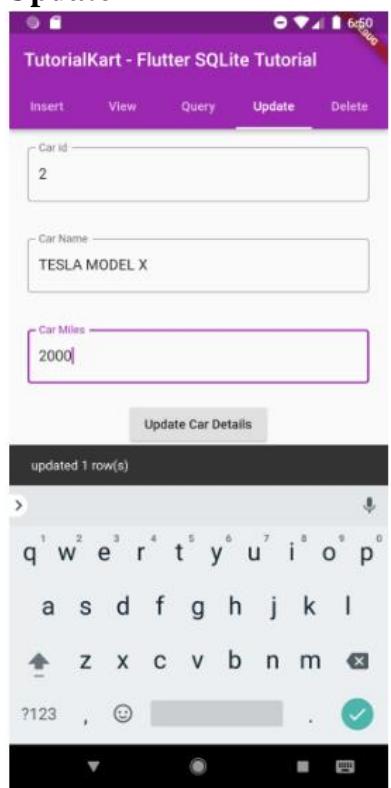
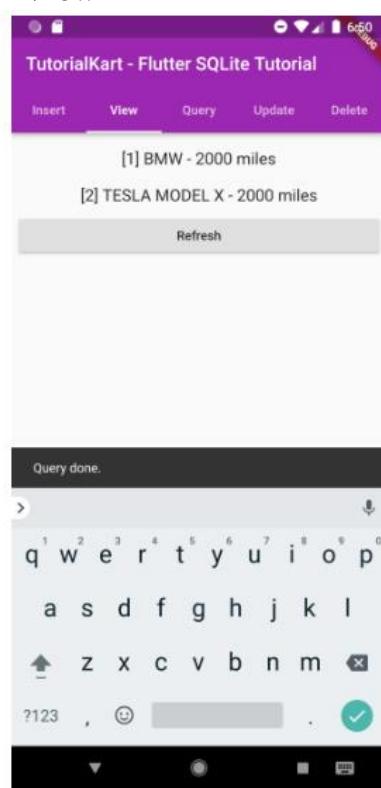
```

pubspec.yaml:

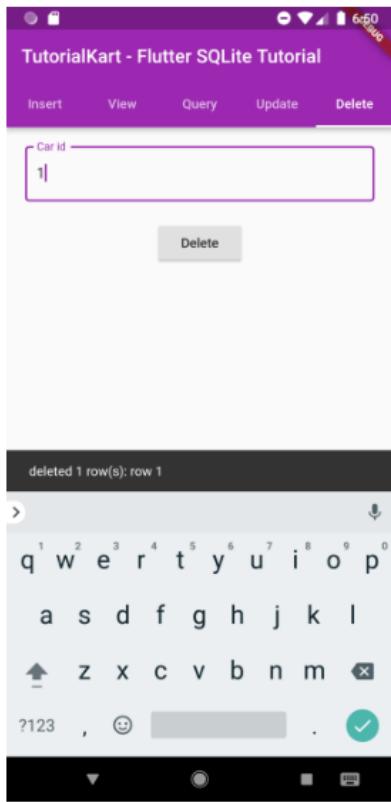
```

dependencies:
  http: ^0.12.0+2
  flutter:
    sdk: flutter
  sqflite: ^2.0.0
  path: ^1.0.0

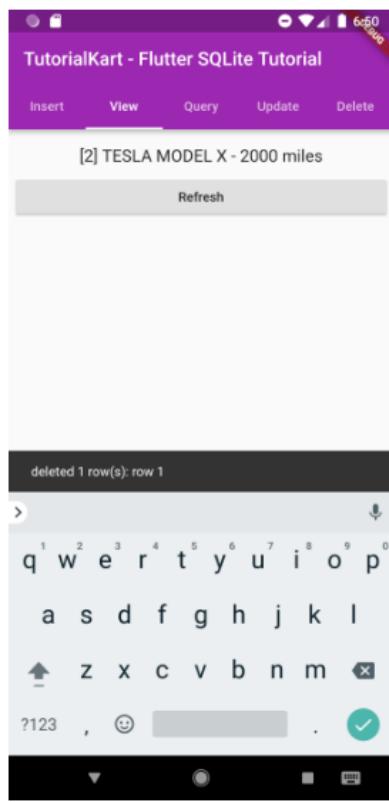
```

Output:**Insert****View****Update****View**

Delete



View



Conclusion:

Hence, we successfully implemented Flutter program to work with SQLite Database.

Practical 13

Aim: Flutter program based on Rest API.

Theory:

Flutter:

1. Flutter is a free and open-source mobile UI framework created by Google and released in May 2017. In a few words, it allows you to create a native mobile application with only one codebase. This means that you can use one programming language and one codebase to create two different apps (for iOS and Android).
2. Flutter consists of two important parts:
 - a. An SDK (Software Development Kit): A collection of tools that are going to help you develop your applications. This includes tools to compile your code into native machine code (code for iOS and Android).
 - b. A Framework (UI Library based on widgets): A collection of reusable UI elements (buttons, text inputs, sliders, and so on) that you can personalize for your own needs.
 - c. To develop with Flutter, you will use a programming language called Dart.
 - d. Dart focuses on front-end development, and you can use it to create mobile and web applications.

Flutter Rest API:

1. Flutter provides http package to use http resources.
2. The http package uses await and async features and provides many high-level methods such as read, get, post, put, head, and delete methods for sending and receiving data from remote locations.
3. These methods simplify the development of REST-based mobile applications.
4. The http package also provides a standard http client class that supports the persistent connection.
5. This class is useful when a lot of requests to be made on a particular server. It should be closed properly using the close() method. Otherwise, it works as an http class.

A) Write a Flutter program to implement REST API:

Code:

main.dart

```
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:http/http.dart' as http;
import 'dart:async';
import 'dart:convert';
import 'Product.dart';

void main() => runApp(MyApp(products: fetchProducts()));

List<Product> parseProducts(String responseBody) {
  final parsed = json.decode(responseBody).cast<Map<String, dynamic>>();
  return parsed.map<Product>((json) => Product.fromMap(json)).toList();
}
```

```
Future<List<Product>> fetchProducts() async {
    final response = await
http.get('http://192.168.131.1/api/products.json');
    if (response.statusCode == 200) {
        return parseProducts(response.body);
    } else {
        throw Exception('Unable to fetch products from the REST API');
    }
}
class MyApp extends StatelessWidget {
    final Future<List<Product>> products;
    MyApp({Key? key, required this.products}) : super(key: key);
// This widget is the root of your application.
@Override
Widget build(BuildContext context) {
    return MaterialApp(
        title: 'Flutter Demo',
        theme: ThemeData(
            primarySwatch: Colors.blue,
        ),
        home: MyHomePage(title: 'Product Navigation demo home page',
products: products),
    );
}
class MyHomePage extends StatelessWidget {
    final String title;
    final Future<List<Product>> products;
    MyHomePage({Key? key, required this.title, required this.products}) :
super(key: key);
// final items = Product.getProducts();
@Override
Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(title: Text("Product Navigation")),
        body: Center(
            child: FutureBuilder<List<Product>>(
                future: products, builder: (context, snapshot) {
                    if (snapshot.hasError) print(snapshot.error);
                    if (snapshot.hasData) {
                        return ProductBoxList(items: snapshot.data);
                    } else {
                        return Center(child: CircularProgressIndicator());
                    }
                },
            ),
        ),
    );
}
class ProductBoxList extends StatelessWidget {
    final List<Product>? items;
    ProductBoxList({Key? key, required this.items});
    @override
Widget build(BuildContext context) {
    return ListView.builder(
        itemCount: items!.length,
        itemBuilder: (context, index) {
            return GestureDetector(
                child: ProductBox(item: items![index]),
                onTap: () {

```

```
        Navigator.push(
            context, MaterialPageRoute(
                builder: (context) => ProductPage(item: items![index]),
            ),
        );
    },
),
);
}
}

class ProductPage extends StatelessWidget {
ProductPage({Key? key, required this.item}) : super(key: key);
final Product item;
@Override
Widget build(BuildContext context) {
return Scaffold(
appBar: AppBar(title: Text(this.item.name),),
body: Center(
child: Container(
padding: EdgeInsets.all(0),
child: Column(
mainAxisAlignment: MainAxisAlignment.start,
crossAxisAlignment: CrossAxisAlignment.start,
children: <Widget>[
Image.asset("assets/" + this.item.image),
Expanded(
child: Container(
padding: EdgeInsets.all(5),
child: Column(
mainAxisAlignment: MainAxisAlignment.spaceEvenly,
children: <Widget>[
Text(this.item.name, style:
TextStyle(fontWeight: FontWeight.bold)),
Text(this.item.description),
Text("Price: " + this.item.price.toString()),
RatingBox(),
],
),
),
),
],
),
),
),
),
);
}
}

class RatingBox extends StatefulWidget {
@Override
_RatingBoxState createState() => _RatingBoxState();
}

class _RatingBoxState extends State<RatingBox> {
int _rating = 0;
void _setRatingAsOne() {
setState(() {
_rating = 1;
});
}
void _setRatingAsTwo() {
setState(() {
```

```
        _rating = 2;
    });
}
void _setRatingAsThree() {
    setState(() {
        _rating = 3;
    });
}
Widget build(BuildContext context) {
    double _size = 20;
    print(_rating);
    return Row(
        mainAxisAlignment: MainAxisAlignment.end,
        crossAxisAlignment: CrossAxisAlignment.end,
        mainAxisSize: MainAxisSize.max,
        children: <Widget>[
            Container(
                padding: EdgeInsets.all(0),
                child: IconButton(
                    icon: (
                        _rating >= 1
                            ? Icon(Icons.star, size: _size,)
                            : Icon(Icons.star_border, size: _size,))
                ),
                color: Colors.red[500], onPressed: _setRatingAsOne, iconSize:
_size,
            ),
            Container(
                padding: EdgeInsets.all(0),
                child: IconButton(
                    icon: (
                        _rating >= 2
                            ? Icon(Icons.star, size: _size,)
                            : Icon(Icons.star_border, size: _size,))
                ),
                color: Colors.red[500],
                onPressed: _setRatingAsTwo,
                iconSize: _size,
            ),
            Container(
                padding: EdgeInsets.all(0),
                child: IconButton(
                    icon: (
                        _rating >= 3 ?
                            Icon(Icons.star, size: _size,)
                            : Icon(Icons.star_border, size: _size,))
                ),
                color: Colors.red[500],
                onPressed: _setRatingAsThree,
                iconSize: _size,
            ),
        ],
    );
}
class ProductBox extends StatelessWidget {
ProductBox({Key? key, required this.item}) : super(key: key);
final Product item;
```

```

Widget build(BuildContext context) {
  return Container(
    padding: EdgeInsets.all(2), height: 140,
    child: Card(
      child: Row(
        mainAxisAlignment: MainAxisAlignment.spaceEvenly,
        children: <Widget>[
          Image.asset("assets/" + this.item.image),
          Expanded(
            child: Container(
              padding: EdgeInsets.all(5),
              child: Column(
                mainAxisAlignment: MainAxisAlignment.spaceEvenly,
                children: <Widget>[
                  Text(this.item.name,
style: TextStyle(fontWeight: FontWeight.bold)),
                  Text(this.item.description),
                  Text("Price: " + this.item.price.toString()),
                  RatingBox(),
                ],
              )
            )
          )
        ],
      );
    );
}

```

product.dart:

```

class Product {
  final String name;
  final String description;
  final int price;
  final String image;
  Product(this.name, this.description, this.price, this.image);
  factory Product.fromMap(Map<String, dynamic> json) {
    return Product(
      json['name'],
      json['description'],
      json['price'],
      json['image'],
    );
  }
}

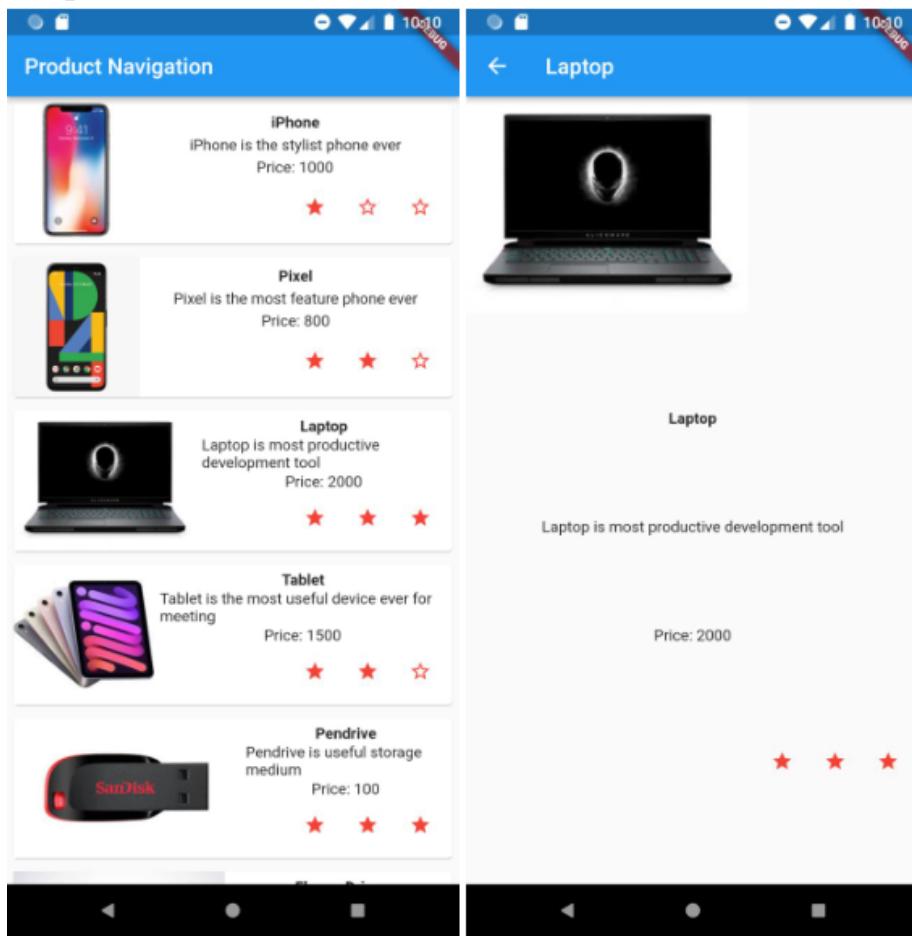
```

pubspec.yaml:

```

assets:
  - assets/iPhone.png
  - assets/Laptop.png
  - assets/Pendrive.png
  - assets/Pixel.jpg
  - assets/Tablet.png
  - assets/Floppy.png

```

Output:**Conclusion:**

Hence, we successfully implemented REST API using Flutter.

Mini Project

Title: Weather Application

Developer:

42, Ronak Pathare

60, Siddesh Waman

Introduction:

Our application '**WEATHER APPLICATION**' will be useful for finding temperature of any city in India. The Application will ask The Application also asks for city name various attention Keywords from user to utilize the services. This Application show temperature of all cities.

Objective:

The basic aim of building this application is to have multiple and easy ways to locate the temperature of city.

The detailed descriptions of the objectives are as follows: -

- Show the current temperature of city
- Show the humidity of the city
- Show the climate of the city good or bad

Advantages:

- Flexible system
- User friendly and dynamic
- Accurate
- Save important Time.

Description:

The present solution provided by WRD Systems involves a tracking unit, a city name-based gateway and a software that runs on a laptop or Desktop PC. The gateway hardware is always connected to the laptop or PC.

Limitations of the System :-

- It will work only when phone is on
- It require all the access of the internet
- It Require the city name

Significance of the System :-

- **1.Socio-economic Significance**

In this study, the system will inspire the other researchers to develop more effective and efficient Advance Mobile tracking system via Android device.

- **2.Technological Significance**

In real life, we want the people to make more used to this one of the latest technologies.

- **3 Safety Features**

If a person is lost, he/she can easily be tracked and brought back to a safe place by using this system.

Requirements

- **Hardware Requirements:**

A typical PC (CPU, Monitor, Keyboard and Mouse)

- CPU: Intel P4(1.6GHz) or higher; 1GB RAM
- Monitor: A 15" CRT or better
- Standard PS/2 Keyboard and Mouse

- **Software Requirements**

- Windows 10
- Netbeans IDE 8.2 & Android Development Tools(ADT) Plugin
- JDK 1.8 & JRE 1.8

- **Deployment Platform**

- **Hardware Requirements**

- An Android compatible device with atleast 1 GHz ARM Cortex Processor and 120MB RAM

- **Software Requirements**

- Android OS(1.6 or higher)
 - Base System software

Code:**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"
    android:layout_margin="15dp"
    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"
            android:padding="10dp"
            android:weightSum="3">
            <EditText
                android:layout_width="0dp"
                android:layout_height="wrap_content"
                android:layout_weight="2.5"
                android:textColor="#000"
                android:textSize="16dp"
                android:background="#fff"
                android:padding="10dp"
                android:gravity="center"
                android:layout_gravity="center"
                android:id="@+id/textField"
                />
            <ImageView
                android:layout_width="0dp"
                android:layout_height="wrap_content"
                android:layout_weight="0.5"
                android:src="@drawable/search"
                android:id="@+id/search"/>
        </LinearLayout>
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical"
            android:padding="10dp">
            <TextView
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:text="Text Temp"
                android:textColor="#fff"
                android:textSize="50dp"
                android:id="@+id/tempText" />
            <TextView
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:text="Description"
                android:textColor="#fff"
                android:textSize="20dp"
```

```

        android:id="@+id/descText" />
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Humidity"
        android:textColor="#fff"
        android:textSize="20dp"
        android:id="@+id/humidityText" />
    </LinearLayout>
</LinearLayout>
</androidx.constraintlayout.widget.ConstraintLayout>

```

MainActivity.java

```

package com.example.mini_project;

import androidx.appcompat.app.AppCompatActivity;
import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {
    ImageView search;
    TextView tempText , descText , humidityText;
    EditText textField;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        search = findViewById(R.id.search);
        tempText = findViewById(R.id.tempText);
        descText = findViewById(R.id.descText);
        humidityText = findViewById(R.id.humidityText);
        textField = findViewById(R.id.textField);
        search.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                getWeatherData(textField.getText().toString().trim());
            }
        });
    }
    private void getWeatherData(String name){
        ApiInterface apiInterface =
        ApiClient.getClient().create(ApiInterface.class);
        Call<Example> call = apiInterface.getWeatherData(name);
        call.enqueue(new Callback<Example>() {
            @Override
            public void onResponse(Call<Example> call, Response<Example>
response) {
                tempText.setText("Temp"+
" "+response.body().getMain().getTemp()+" C");
                descText.setText("Feels Like"+"
"+response.body().getMain().getFeels_like());
                humidityText.setText("Humidity"+""+
response.body().getMain().getHumidity());
            }
        });
    }
}

```

```
    ;
    }
    @Override
    public void onFailure(Call<Example> call, Throwable t) {
        }
    }) ;
}
}
```

ApiClient.java

```
package com.example.mini_project;

import retrofit2.Retrofit;
import retrofit2.converter.gson.GsonConverterFactory;

public class ApiClient {
    private static Retrofit retrofit = null;
    public static Retrofit getClient(){
        if (retrofit == null){
            retrofit = new Retrofit.Builder()
                .baseUrl("https://api.openweathermap.org/data/2.5/")
                .addConverterFactory(GsonConverterFactory.create())
                .build();
        }
        return retrofit;
    }
}
```

ApiInterface.java

```
package com.example.mini_project;
import retrofit2.Call;
import retrofit2.http.GET;
import retrofit2.http.Query;
public interface ApiInterface {
    @GET("weather?appid=92756c24107bc39dd0a7541f66ba55c5&units=metric")
    Call<Example> getWeatherData(@Query("q") String name);
}
```

Example.java

```
package com.example.mini_project;

import com.google.gson.annotations.SerializedName;
public class Example {
    @SerializedName("main")
    private Main main;
    public Main getMain() {
        return main;
    }
    public void setMain(Main main) {
        this.main = main;
    }
}
```

Main.java

```
package com.example.mini_project;
import com.google.gson.annotations.SerializedName;
public class Main {
    @SerializedName("temp")
    String temp;
    @SerializedName("humidity")
    String humidity;
    @SerializedName("feels_like")
    String feels_like;
    public String getTemp() {
        return temp;
    }
    public void setTemp(String temp) {
        this.temp = temp;
    }
    public String getHumidity() {
        return humidity;
    }
    public void setHumidity(String humidity) {
        this.humidity = humidity;
    }
    public String getFeels_like() {
        return feels_like;
    }
    public void setFeels_like(String feels_like) {
        this.feels_like = feels_like;
    }
}
```

strings.xml

```
<resources>
    <string name="app_name">WeatherApplication</string>
</resources>
```

style.xml

```
<resources>
    <!-- Base application theme. -->
    <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
        <!-- Customize your theme here. -->
        <item name="colorPrimary">@color/colorPrimary</item>
        <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
        <item name="colorAccent">@color/colorAccent</item>
    </style>
</resources>
```

colors.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="colorPrimary">#6200EE</color>
    <color name="colorPrimaryDark">#3700B3</color>
    <color name="colorAccent">#03DAC5</color>
</resources>
```

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.mini_project">
    <uses-permission android:name="android.permission.INTERNET"/>
    <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.Mini_project"
        android:usesCleartextTraffic="true">
        <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>
    </application>
</manifest>
```

Results:



Future Enhancements

- Adding better UI to improve user experience.
- High end devices compatible.
- Other OS compatible.
- More Accuracy.
- Easier Interface.

Conclusion:

- The approach of this android app was to have multiple and easy ways to locate the temperature of city.
- The application is free of cost and can be used anytime anywhere.

Web references:

1. <http://developer.android.com/index.html>
2. <https://developers.google.com/android/?csw=1>
3. <http://androidcommunity.com/>
4. <http://android.wordpress.org/>