

Roll No.42

Exam Seat No. _____

VIVEKANAND EDUCATION SOCIETY'S INSTITUTE OF TECHNOLOGY

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



Since 1962

CERTIFICATE

Certified that Mr. **CHRISTY PHILIP [ROLL NO:42]** of **SYMCA-2B** has satisfactorily completed a course of the necessary experiments in **MCAL34 – Mobile Computing Lab** under the supervision of **Mrs. Vaishali Gatty** in the Institute of Technology in the academic year **2022- 2023**.

Principal

Head of Department

Lab In-charge

Subject Teacher

**V.E.S. Institute of Technology, Collector Colony,
Chembur, Mumbai**



Department of M.C.A

MOBILE COMPUTING INDEX

Sr. No	Contents	Date Of Performance	Date Of Submission	Marks	Sign
1	Android program using Various UI Components	15/09/2022	29/02/2022	10	
2	Android program using different layouts and views	29/09/2022	06/10/2022	10	
3	Android program based on Intents	30/09/2022	06/10/2022	10	
4	Android program for notifications and alert box	06/10/2022	13/10/2022	10	
5	Android program to perform CRUD operation using SQLite DB	13/10/2022	20/10/2022	10	
6	Android program using shared preference ,Internal and External storage	14/10/2022	21/10/2022	10	
7	Android program to work with graphics and animation	21/10/2022	27/10/2022	10	
8	Android program to work with google maps and location	27/10/2022	03/11/2022	10	
9	Android program to work with images and videos	03/11/2022	15/11/2022	10	
10	Android program based on Rest API	15/11/2022	17/11/2022	10	
11	Flutter program using layout ,widget and state management	17/11/2022	18/11/2022	10	
12	Flutter program to work with SQLite Database	18/11/2022	22/11/2022	10	
13	Flutter program based on RestAPI	18/11/2022	22/11/2022	10	

PRACTICAL 1

Study of different android applications using UI components.

A)

AIM: - Android program using various UI Components.(EditText, RatingBar, CheckBox, RadioButton, Button etc). Get the values from the components and display them on the next activity.

THEORY: -

Android apps can be written using Kotlin, Java, and C++ languages. The Android SDK tools compile your code along with any data and resource files into an APK or an Android App Bundle.

An Android package, which is an archive file with an .apk suffix, contains the contents of an Android app that are required at runtime and it is the file that Android-powered devices use to install the app.

Each Android app lives in its own security sandbox, protected by the following Android security features:

The Android operating system is a multi-user Linux system in which each app is a different user. By default, the system assigns each app a unique Linux user ID (the ID is used only by the system and is unknown to the app). The system sets permissions for all the files in an app so that only the user ID assigned to that app can access them.

Each process has its own virtual machine (VM), so an app's code runs in isolation from other apps.

By default, every app runs in its own Linux process. The Android system starts the process when any of the app's components need to be executed, and then shuts down the process when it's no longer needed or when the system must recover memory for other apps.

App components

App components are the essential building blocks of an Android app. Each component is an entry point through which the system or a user can enter your app. Some components depend on others.

There are four different types of app components:

Activities

Services

Broadcast receivers

Content providers

Activities

An activity is the entry point for interacting with the user. It represents a single screen with a user interface. For example, an email app might have one activity that shows a list of new emails, another activity to compose an email, and another activity for reading emails. Although the activities work together to form a cohesive user experience in the email app, each one is independent of the others. As such, a different app can start any one of these activities if the email app allows it. For example, a camera app can start the activity in the email app that composes new mail to allow the user to share a picture.

Services

A service is a general-purpose entry point for keeping an app running in the background for all kinds of reasons. It is a component that runs in the background to perform long-running operations or to perform work for remote processes. A service does not provide a user interface. For example, a service might play music in the background while the user is in a different app, or it might fetch data over the network without blocking user interaction with an activity.

Started services tell the system to keep them running until their work is completed.

Bound services run because some other app (or the system) has said that it wants to make use of the service. This is basically the service providing an API to another process. The system thus knows there is a dependency between these processes, so if process A is bound to a service in process B, it knows that it needs to keep process B (and its service) running for A. Further, if process A is something the user cares about, then it also knows to treat process B as something the user also cares about.

Broadcast receivers

A broadcast receiver is a component that enables the system to deliver events to the app outside of a regular user flow, allowing the app to respond to system-wide broadcast announcements.

Because broadcast receivers are another well-defined entry into the app, the system can deliver broadcasts even to apps that aren't currently running. So, for example, an app can schedule an alarm to post a notification to tell the user about an upcoming event... and by delivering that alarm to a BroadcastReceiver of the app, there is no need for the app to remain running until the alarm goes off.

Content providers

A content provider manages a shared set of app data that you can store in the file system, in a SQLite database, on the web, or on any other persistent storage location that your app can access. Through the content provider, other apps can query or modify the data if the content provider allows it. For example, the Android system provides a content provider that manages the user's contact information.

An intent is created with an Intent object, which defines a message to activate either a specific component (explicit intent) or a specific type of component (implicit intent).

There are separate methods for activating each type of component:

- You can start an activity or give it something new to do by passing an Intent to `startActivity()` or `startActivityForResult()` (when you want the activity to return a result).
- With Android 5.0 (API level 21) and later, you can use the `JobScheduler` class to schedule actions. For earlier Android versions, you can start a service (or give new instructions to an ongoing service) by passing an Intent to `startService()`. You can bind to the service by passing an Intent to `bindService()`.
- You can initiate a broadcast by passing an Intent to methods such as `sendBroadcast()`, `sendOrderedBroadcast()`, or `sendStickyBroadcast()`.
- You can perform a query to a content provider by calling `query()` on a `ContentResolver`

UI COMPONENTS

1. TextView

TextView is a UI Component that displays the text to the user on their Display Screen.

There are various attributes to describe the TextView some of them are named below:

Android: id – it is a unique id for the control.

Android: width – It displays the exact width of the TextView.

Android: height – It displays the exact height of the TextView.

Android:textColor – It set the color of the text.

Android: gravity – It is to align the TextView.

2. EditText

EditText is a user interface control that allows the users to enter some text.

3. Button

This is a UI that is used to perform some action as soon as the user clicks on it.

4. ImageButton

It is the same as a Button but it's used to display an image on the button to perform an Action. In this, we need to give the source of the image so that the system can load it.

5. ToggleButton

The toggle button displays the ON/OFF states of a button with a light indicator.

6. RadioButton

Radio button in Android is the one that has only two possible states, that are either checked or unchecked. Initially, it is in the unchecked state, once it's checked it can't be unchecked.

7. RadioGroup

It's a group of Radio buttons that are alike. In this, only one of all the buttons can be chosen.

8. CheckBox

A CheckBox is the UI control that has two states that are either checked or unchecked. If we have a group of CheckBox, we can select as many as we want, unlike RadioGroup.

9. ProgressBar

In Android, we have a progress bar that shows the progress of some action that is happening like pasting a file to some location. A progress bar can be in two modes:

1) Determinate Mode:

In this, the progress is shown with the percent of action completed. Also, the time to be taken is already determined.

2) Indeterminate Mode:

In this, there is no idea of when the task would be completed, therefore, it functions continuously.

10. RatingBar

A rating bar in Android is an extended version of a seekbar. It is used to give the rating by touching it. In the rating bar, a user can rate at a scale of 5 with a difference of 0.5.

Its rating is in Stars. The user needs to tap/click the stars.

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:background="#D7ECEF"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Feedback Form"
        android:textSize="20sp"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.094" />

    <TextView
        android:id="@+id/textView3"
```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="NAME: -"
    android:textSize="16sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.11"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.206" />
```

```
<EditText
    android:id="@+id/editTextTextPersonName"
    android:layout_width="257dp"
    android:layout_height="48dp"
    android:ems="10"
    android:inputType="textPersonName"
    android:minHeight="48dp"
    android:text="Please Enter Your Full Name"
    app:layout_constraintBaseline_toBaselineOf="@+id/textView3"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.529"
    app:layout_constraintStart_toEndOf="@+id/textView3" />
```

```
<TextView
    android:id="@+id/textView4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Gender: -"
    android:textSize="16sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.113"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView3"
    app:layout_constraintVertical_bias="0.153" />
```

```
<RadioGroup
    android:id="@+id/radioGroup"
    android:layout_width="271dp"
```

```
        android:layout_height="50dp"
        android:layout_marginTop="52dp"
        android:orientation="horizontal"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.586"
        app:layout_constraintStart_toEndOf="@+id/textView4"
        app:layout_constraintTop_toBottomOf="@+id/editTextTextPersonName">

    <RadioButton
        android:id="@+id radioButton4"
        android:layout_width="129dp"
        android:layout_height="wrap_content"
        android:text="Male" />

    <RadioButton
        android:id="@+id radioButton5"
        android:layout_width="139dp"
        android:layout_height="match_parent"
        android:text="Female" />
</RadioGroup>

<TextView
    android:id="@+id textView5"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Subject: -"
    android:textSize="16sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.113"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<CheckBox
    android:id="@+id checkBox"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Green Computing"
    android:textSize="12sp"
    app:layout_constraintBottom_toBottomOf="parent"
```

```
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.044"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.61" />
```

```
<CheckBox
    android:id="@+id/checkBox2"
    android:layout_width="130dp"
    android:layout_height="47dp"
    android:text="Cloud Computing"
    android:textSize="12sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toEndOf="@+id/checkBox"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.61" />
```

```
<CheckBox
    android:id="@+id/checkBox3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Mobile Computing"
    android:textSize="12sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toEndOf="@+id/checkBox2"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.61" />
```

```
<TextView
    android:id="@+id/textView6"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Rating: -"
    android:textSize="16sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toStartOf="@+id/ratingBar"
    app:layout_constraintHorizontal_bias="0.535"
    app:layout_constraintStart_toStartOf="parent"
```

```
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.738" />

<RatingBar
    android:id="@+id/ratingBar"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.639"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.756"
    tools:ignore="SpeakableTextPresentCheck" />

<Button
    android:id="@+id/button3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="#B0BDFF22"
    android:text="SUBMIT"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.527"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.891" />

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

MainActivity.java

```
package com.example.prac1a;

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);
    getSupportActionBar().setTitle("Feedback Form");
    EditText etname= findViewById(R.id.editTextTextPersonName);
    rb1 = (RadioButton) findViewById(R.id.radioButton4);
    rb2 = (RadioButton) findViewById(R.id.radioButton5);
    cb1 = findViewById(R.id.checkBox);
    cb2 = findViewById(R.id.checkBox2);
    cb3 = findViewById(R.id.checkBox3);
    RatingBar rbar = findViewById(R.id.ratingBar);
    Button submitBtn= (Button)findViewById(R.id.button3);
    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.coordinatorlayout.widget.CoordinatorLayout
    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">

    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.coordinatorlayout.widget.CoordinatorLayout>

```

MainActivity2.java

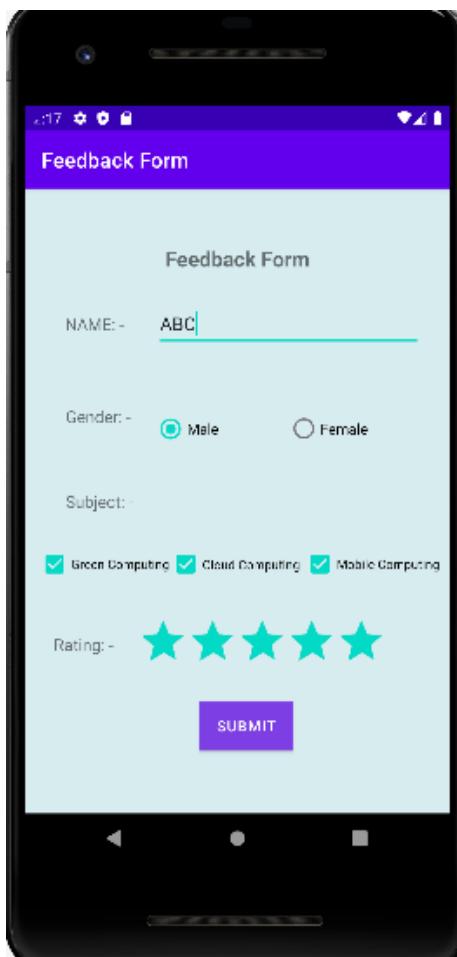
```

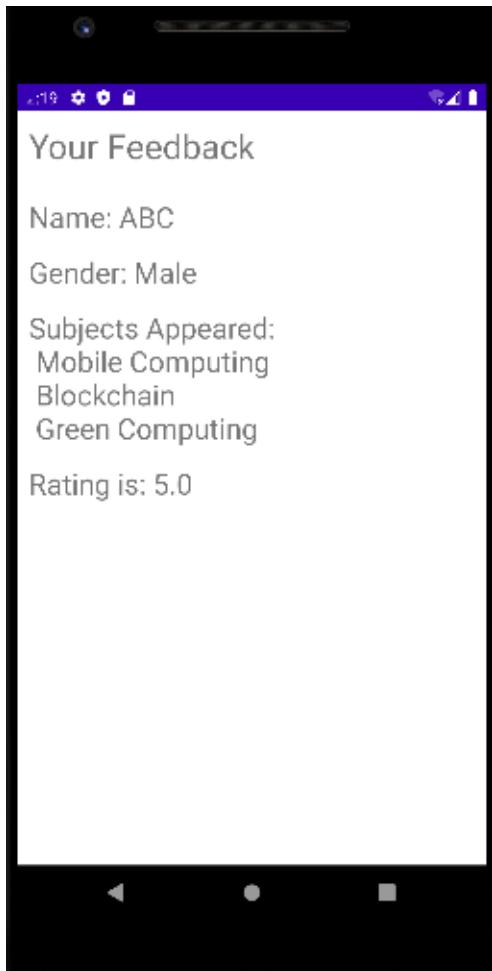
package com.example.prac1a;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.widget.TextView;
public class MainActivity2 extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        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: -





CONCLUSION: -

From this practical I have learned to implement Android program using various UI Components.(EditText, RatingBar, CheckBox, RadioButton, Button etc). Get the values from the components and display them on the next activity.

B)

AIM: - 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 the 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"
    tools:context=".MainActivity">
```

```
<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="LOGIN"
    android:textSize="34sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.15"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.24" />
```

```
<TextView
    android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Enter credentials to Login"
    android:textSize="16sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.206"
    app:layout_constraintStart_toStartOf="parent"
```

```
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.349" />

<EditText
    android:id="@+id/ET1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:ems="10"
    android:inputType="textPersonName"
    android:text=""
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.199"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.529" />
```

```
<TextView
    android:id="@+id/textView3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="LOGIN ID:"
    android:textSize="20sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.133"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.438" />
```

```
<TextView
    android:id="@+id/textView4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="PASSWORD:"
    android:textSize="20sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
```

```
    app:layout_constraintHorizontal_bias="0.133"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.622" />

<EditText
    android:id="@+id/ET2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:ems="10"
    android:inputType="textPersonName"
    android:text=""
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.233"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.725" />

<Button
    android:id="@+id/Btn1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="SUBMIT"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.912" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java

```
package com.example.prac1b;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
```

```

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

public class MainActivity extends AppCompatActivity {
    TextView t1,t2;
    Button b1;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        t1=findViewById(R.id.ET1);
        t2=findViewById(R.id.ET2);
        b1=findViewById(R.id.Btn1);
        b1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String email=t1.getText().toString();
                String pass=t1.getText().toString();
                Intent intent=new Intent(MainActivity.this,MainActivity2.class);
                intent.putExtra("Email",email);
                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">

    <TextView
        android:id="@+id/T1"

```

```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="TextView"
        android:textSize="20sp"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

<TextView
    android:id="@+id/textView5"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="WELCOME"
    android:textSize="20sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.447" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity2.java

```

package com.example.prac1b;

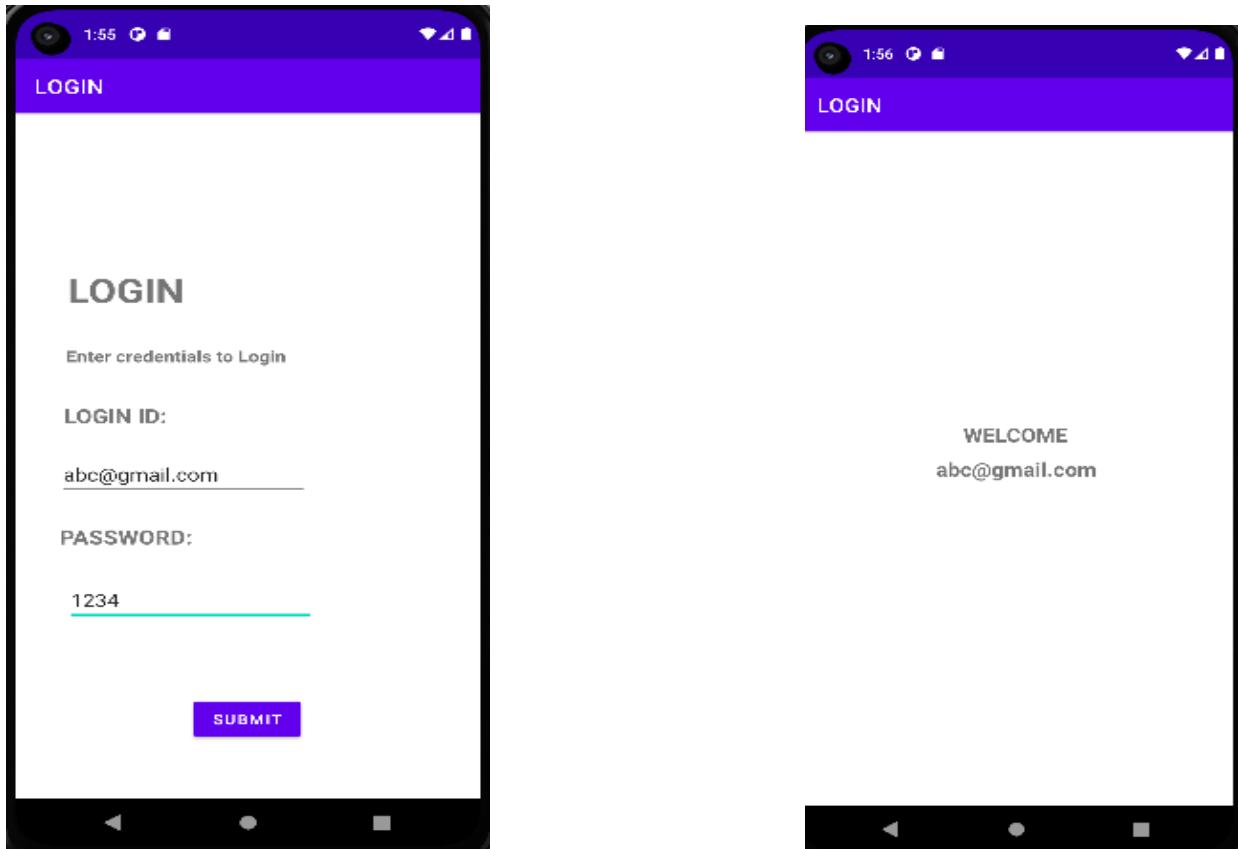
import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.provider.ContactsContract;
import android.widget.TextView;

public class MainActivity2 extends AppCompatActivity {
    TextView t1;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
```

```
setContentView(R.layout.activity_main2);
String email= getIntent().getStringExtra("Email");
t1=findViewById(R.id.TI);
t1.setText(email);
}
}
```

OUTPUT: -



CONCLUSION -

From this practical I have learned to implement Login Form using UI Components.

PRACTICAL 2

A)

AIM: - Design a Registration form to show the working of different layouts.

THEORY: -

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.

This can have two orientations:

- a. Vertical Orientation – It is shown above where the widgets such as TextViews, and all in a vertical manner.
- b. Horizontal Orientation – It is shown above where the widgets such as TextViews, and all in a horizontal 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. To define it in such a way, we write the following:

`android:layout_alignParentTop= “true”`

`android:layout_alignParentLeft= “true”`

If we write the above code, the element will get aligned on the top left of the parent container.

If we want to align it with some other element in the same container, it can be defined as follows:

`android:layout_alignLeft= “@+id/element_name”`

`android:layout_below= “@+id/element_name”`

3. Constraint Layout

A ConstraintLayout is a ViewGroup which allows you to position and size widgets in a flexible way. Note: ConstraintLayout is available as a support library that you can use on Android

systems starting with API level 9 (Gingerbread). As such, we are planning on enriching its API and capabilities over time.

4. Table Layout

A layout that arranges its children into rows and columns. A TableLayout consists of a number of TableRow objects, each defining a row (actually, you can have other children). TableLayout containers do not display border lines for their rows, columns, or cells.

5. Frame Layout

FrameLayout is designed to block out an area on the screen to display a single item. Generally, FrameLayout should be used to hold a single child view, because it can be difficult to organize child views in a way that's scalable to different screen sizes without the children overlapping each other. You can, however, add multiple children to a FrameLayout and control their position within the FrameLayout by assigning gravity to each child, using the android:layout_gravity attribute.

6. List View

An adapter actually bridges between UI components and the data source that fill data into UI Component. Adapter holds the data and send the data to adapter view, the view can takes the data from adapter view and shows the data on different views like as spinner, list view, grid view etc.

7. Grid View

any web pages are based on a grid-view, which means that the page is divided into columns: Using a grid-view is very helpful when designing web pages. It makes it easier to place elements on the page.

8. Absolute Layout

An AbsoluteLayout is used to position and size children using explicit values. The position is specified by the upper-left corner of the child relative to the upper-left corner of the AbsoluteLayout , in device-independent units. AbsoluteLayout also implements a proportional positioning and sizing feature.

9. Web View

Android System WebView is a system component for the Android operating system (OS) that enables Android apps to display web content directly inside an application.

There are two ways to open web content on an Android device: You can use a traditional web browser or an Android application that includes WebView in the layout.

A developer who wants to add browser functionality to an application can include the WebView library and create an instance of a WebView class. This approach essentially embeds a browser within the app to do things like render webpages and execute script.

10. Scroll View

In Android, a ScrollView is a view group that is used to make vertically scrollable views. A scroll view contains a single direct child only. In order to place multiple views in the scroll view, one needs to make a view group(like LinearLayout) as a direct child and then we can define many views inside it.

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">
```

```
<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="48dp"
    android:text="Registration Form"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
```

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

```
<TextView
    android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="64dp"
    android:text="Name"
    app:layout_constraintBottom_toTopOf="@+id/textView3"
    app:layout_constraintEnd_toStartOf="@+id/editTextTextPersonName"
    app:layout_constraintHorizontal_bias="0.451"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView"
    app:layout_constraintVertical_bias="1.0" />
```

```
<TextView
    android:id="@+id/textView3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="176dp"
    android:text="Email-Id"
    app:layout_constraintEnd_toStartOf="@+id/editTextTextPersonName2"
    app:layout_constraintHorizontal_bias="0.423"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```
<TextView
    android:id="@+id/textView4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="56dp"
    android:text="Age"
    app:layout_constraintEnd_toStartOf="@+id/editTextTextPersonName3"
    app:layout_constraintHorizontal_bias="0.418"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView3" />
```

```
<TextView
    android:id="@+id/textView5"
```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="52dp"
    android:text="Address"
    app:layout_constraintEnd_toStartOf="@+id/editTextTextPersonName4"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView4" />
```

```
<TextView
    android:id="@+id/textView6"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="52dp"
    android:text="Phone Number"
    app:layout_constraintEnd_toStartOf="@+id/editTextTextPersonName5"
    app:layout_constraintHorizontal_bias="0.507"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView5" />
```

```
<TextView
    android:id="@+id/textView8"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="28dp"
    android:layout_marginTop="44dp"
    android:text="Enter Password"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView6" />
```

```
<TextView
    android:id="@+id/textView9"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginTop="52dp"
    android:text="Confirm Password"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView8" />
```

```
<EditText
```

```
    android:id="@+id/editTextTextPersonName"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="76dp"
    android:ems="10"
    android:inputType="textPersonName"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.731"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```
<EditText
    android:id="@+id/editTextTextPersonName2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="28dp"
    android:layout_marginEnd="48dp"
    android:ems="10"
    android:inputType="textPersonName"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/editTextTextPersonName" />
```

```
<EditText
    android:id="@+id/editTextTextPersonName3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="28dp"
    android:ems="10"
    android:inputType="textPersonName"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.766"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/editTextTextPersonName2" />
```

```
<EditText
    android:id="@+id/editTextTextPersonName4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="28dp"
    android:layout_marginEnd="48dp"
```

```
        android:ems="10"
        android:inputType="textPersonName"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/editTextTextPersonName3" />

<EditText
        android:id="@+id/editTextTextPersonName5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="28dp"
        android:layout_marginEnd="48dp"
        android:ems="10"
        android:inputType="textPersonName"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/editTextTextPersonName4" />

<Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" />

<EditText
        android:id="@+id/editTextTextPassword"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="28dp"
        android:ems="10"
        android:inputType="textPassword"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.297"
        app:layout_constraintStart_toEndOf="@+id/textView8"
        app:layout_constraintTop_toBottomOf="@+id/editTextTextPersonName5" />

<EditText
        android:id="@+id/editTextTextPassword2"
        android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
    android:layout_marginTop="28dp"
    android:ems="10"
    android:inputType="textPassword"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.217"
    app:layout_constraintStart_toEndOf="@+id/textView9"
    app:layout_constraintTop_toBottomOf="@+id/editTextTextPassword" />

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

MainActivity.java

```
package com.example.practical5;

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;

public class MainActivity extends AppCompatActivity {
    Button switchToSecondActivity;
    EditText name,email,age,address,number;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        switchToSecondActivity = findViewById(R.id.button);
        name=findViewById(R.id.editTextTextPersonName);
        email=findViewById(R.id.editTextTextPersonName2);
        age=findViewById(R.id.editTextTextPersonName3);
        address=findViewById(R.id.editTextTextPersonName4);
        number=findViewById(R.id.editTextTextPersonName5);
        switchToSecondActivity.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                switchActivities();
            }
        });
    }
}
```

```

        }
    });
}

private void switchActivities() {
    Intent switchActivityIntent = new Intent(this, MainActivity2.class);
    switchActivityIntent.putExtra("name", name.getText().toString());
    switchActivityIntent.putExtra("email", email.getText().toString());
    switchActivityIntent.putExtra("age", age.getText().toString());
    switchActivityIntent.putExtra("address", address.getText().toString());
    switchActivityIntent.putExtra("number", number.getText().toString());
    startActivity(switchActivityIntent);
}
}

```

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="match_parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintBottom_toBottomOf="parent"
        android:orientation="vertical"
        android:gravity="center"
    >

```

```

    <TextView
        android:layout_width="wrap_content"

```

```
        android:layout_height="wrap_content"
        android:id="@+id/name"
        android:hint="name"
        android:layout_marginTop="30dp"
    />
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/email"
    android:hint="name"
    android:layout_marginTop="30dp"
/><TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/age"
    android:hint="name"
    android:layout_marginTop="30dp"
/><TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/address"
    android:hint="name"
    android:layout_marginTop="30dp"
/>
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/number"
    android:hint="name"
    android:layout_marginTop="30dp"
/>
</LinearLayout>
</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity2.java

```
package com.example.practical5;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.widget.TextView;

public class MainActivity2 extends AppCompatActivity {

    TextView name,email,age,address,number;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main2);
        name=findViewById(R.id.name);
        email=findViewById(R.id.email);
        age=findViewById(R.id.age);
        address=findViewById(R.id.address);
        number=findViewById(R.id.number);

        name.setText(getIntent().getStringExtra("name"));
        email.setText(getIntent().getStringExtra("email"));
        age.setText(getIntent().getStringExtra("age"));
        address.setText(getIntent().getStringExtra("address"));
        number.setText(getIntent().getStringExtra("number"));
    }
}
```

OUTPUT: -

Practical 5	
Registration Form	
Name	Admin
EmailId	admin@gmail.com
Age	21
Address	chembur
Phone Number	9876543210
Enter Password	***
Confirm Password	***
BUTTON	

CONCLUSION: -

From this practical I have learned the implementation of Registration form to show the working of different layouts.

B)

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

CODE: -

Activitymain.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/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="120dp"
        android:hint="Result"
        android:textSize="50sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.54"
        app:layout_constraintStart_toStartOf="parent" />

    <EditText
        android:id="@+id/editTextTextPersonName"
        android:layout_width="178dp"
        android:layout_height="48dp"
        android:ems="10"
        android:inputType="textPersonName"
        android:hint="1st Number"
        app:layout_constraintBottom_toTopOf="@+id/addbutton"
        app:layout_constraintEnd_toStartOf="@+id/editTextTextPersonName2"
        app:layout_constraintHorizontal_bias="0.25"
        app:layout_constraintStart_toStartOf="parent"
```

```
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.514" />

<EditText
    android:id="@+id/editTextTextPersonName2"
    android:layout_width="170dp"
    android:layout_height="48dp"
    android:ems="10"
    android:inputType="textPersonName"
    android:hint="2nd Number"
    app:layout_constraintBottom_toTopOf="@+id/subbutton"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<Button
    android:id="@+id/subbutton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginEnd="52dp"
    android:layout_marginBottom="4dp"
    android:onClick="onClicksubBtn"
    android:text="Sub"
    app:layout_constraintBottom_toTopOf="@+id/divbutton"
    app:layout_constraintEnd_toStartOf="@+id/resetbutton"
    app:layout_constraintHorizontal_bias="1.0"
    app:layout_constraintStart_toEndOf="@+id/addbutton" />
<Button
    android:id="@+id/resetbutton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginEnd="16dp"
    android:layout_marginBottom="8dp"
    android:onClick="onClickresetBtn"
    android:text="Reset"
    app:layout_constraintBottom_toTopOf="@+id/modbutton"
    app:layout_constraintEnd_toEndOf="parent" />
<Button
    android:id="@+id/modbutton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
```

```
    android:layout_marginEnd="16dp"
    android:layout_marginBottom="124dp"
    android:onClick="onClickmodBtn"
    android:text="Mod"
    app:layout_constraintBottom_toTopOf="@+id/textView"
    app:layout_constraintEnd_toEndOf="parent" />
<Button
    android:id="@+id/addbutton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="32dp"
    android:layout_marginBottom="8dp"
    android:onClick="onClickaddBtn"
    android:text="Add"
    app:layout_constraintBottom_toTopOf="@+id/mulbutton"
    app:layout_constraintStart_toStartOf="parent" />
<Button
    android:id="@+id/divbutton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginEnd="52dp"
    android:layout_marginBottom="124dp"
    android:onClick="onClickdivBtn"
    android:text="Div"
    app:layout_constraintBottom_toTopOf="@+id/textView"
    app:layout_constraintEnd_toStartOf="@+id/modbutton"
    app:layout_constraintHorizontal_bias="1.0"
    app:layout_constraintStart_toEndOf="@+id/mulbutton" />
<Button
    android:id="@+id/mulbutton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="32dp"
    android:layout_marginBottom="124dp"
    android:onClick="onClickmulBtn"
    android:text="Mul"
    app:layout_constraintBottom_toTopOf="@+id/textView"
    app:layout_constraintStart_toStartOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Mainactivity.java

```
package com.example.calculator;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
public class MainActivity extends AppCompatActivity {
EditText n1, n2 ;
TextView res;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    n1 = findViewById(R.id.editTextTextPersonName);
    n2 = findViewById(R.id.editTextTextPersonName2);
    res = findViewById(R.id.textView);
}
public void onClickaddBtn(View v)
{
    Toast.makeText(this, "Clicked on Button", Toast.LENGTH_LONG).show();
    int num1 = Integer.parseInt(n1.getText().toString());
    int num2 = Integer.parseInt(n2.getText().toString());
    res.setText(Integer.toString(num1+num2));
}
public void onClicksubBtn(View v)
{
    Toast.makeText(this, "Clicked on Button", Toast.LENGTH_LONG).show();
    int num1 = Integer.parseInt(n1.getText().toString());
    int num2 = Integer.parseInt(n2.getText().toString());
    res.setText(Integer.toString(num1-num2));
}
public void onClickmulBtn(View v)
{
    Toast.makeText(this, "Clicked on Button", Toast.LENGTH_LONG).show();
    int num1 = Integer.parseInt(n1.getText().toString());
    int num2 = Integer.parseInt(n2.getText().toString());
    res.setText(Integer.toString(num1*num2));
}
```

```

}

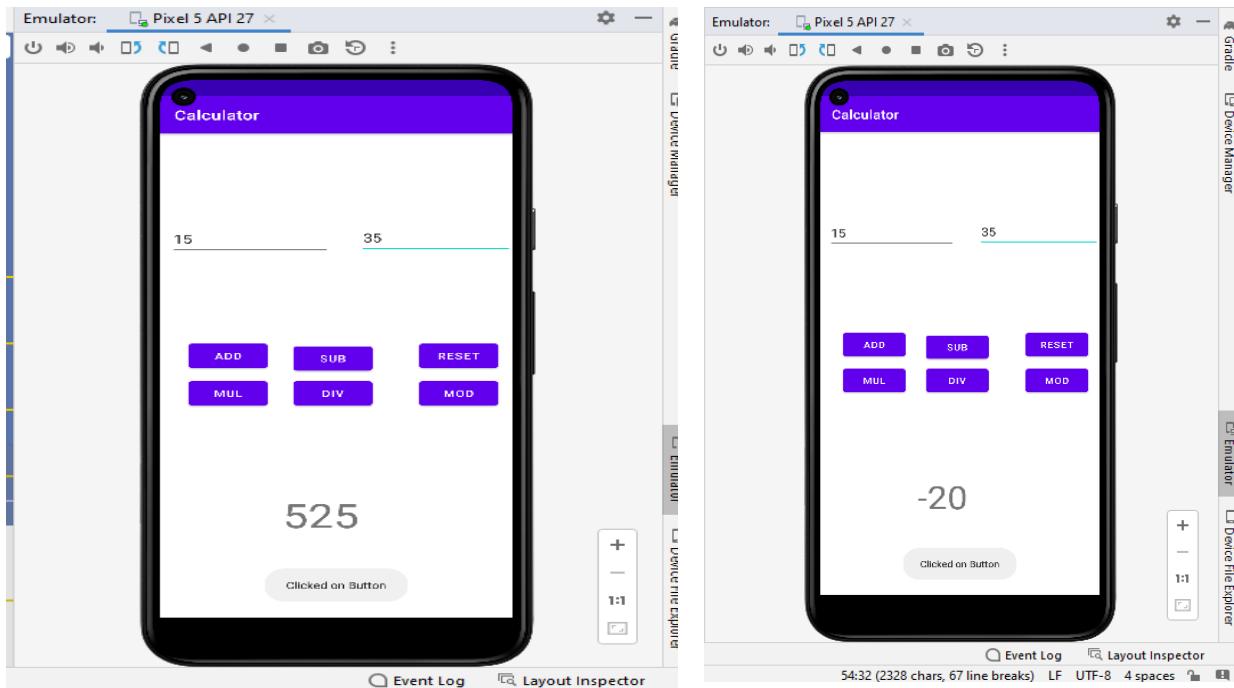
public void onClickdivBtn(View v)
{
    Toast.makeText(this, "Clicked on Button", Toast.LENGTH_LONG).show();
    int num1 = Integer.parseInt(n1.getText().toString());
    int num2 = Integer.parseInt(n2.getText().toString());
    res.setText(Integer.toString(num1/num2));
}

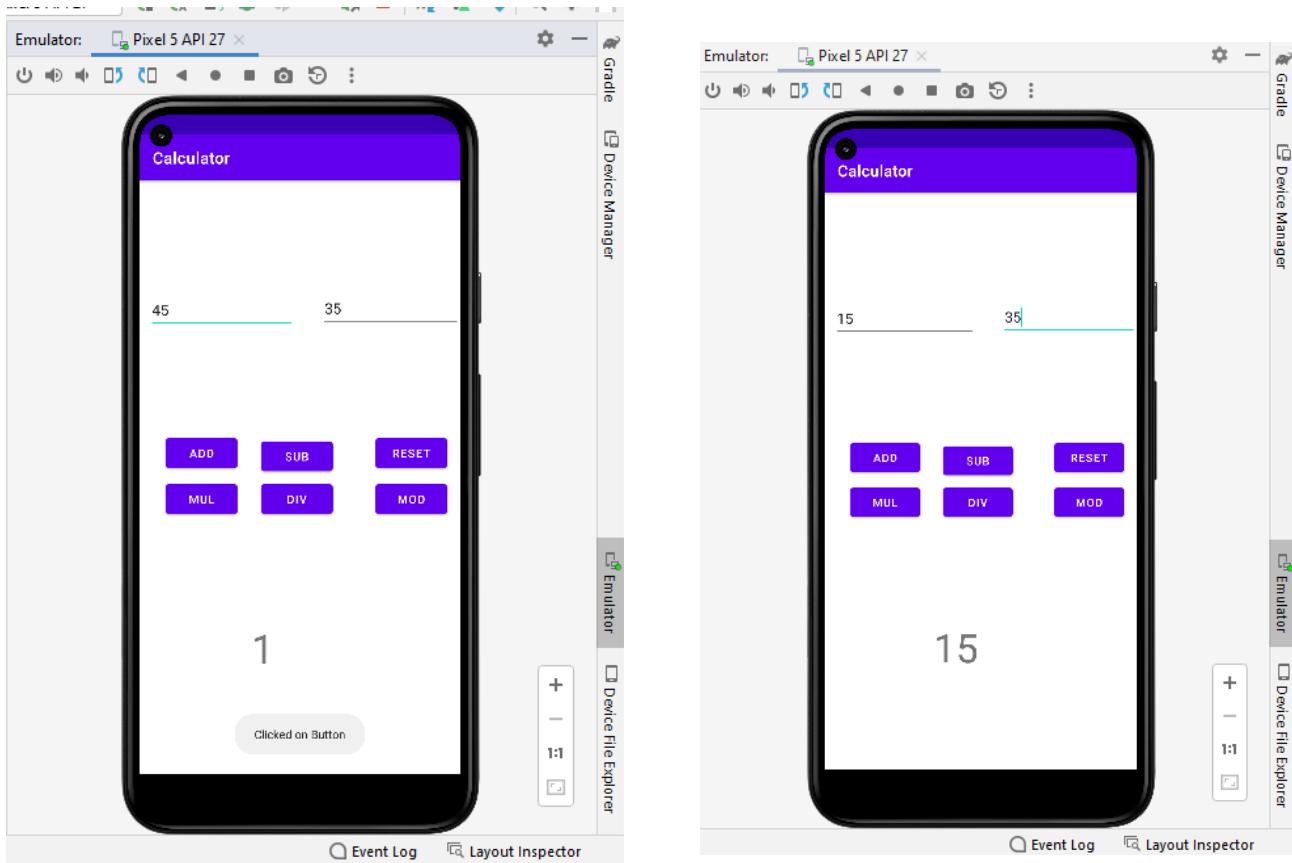
public void onClickresetBtn(View v)
{
    Toast.makeText(this, "Clicked on Button", Toast.LENGTH_LONG).show();
    n1.setText("");
    n2.setText("");
}

public void onClickmodBtn(View v)
{
    Toast.makeText(this, "Clicked on Button", Toast.LENGTH_LONG).show();
    int num1 = Integer.parseInt(n1.getText().toString());
    int num2 = Integer.parseInt(n2.getText().toString());
    res.setText(Integer.toString(num1%num2));
}
}

```

OUTPUT: -





CONCLUSION: -

From this practical I have learned to implement an application to design simple calculator to perform addition, subtraction, multiplication and division. Show message for divide by zero error using Toast.

C)

AIM: - Create an application for Unit Conversion.

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:id="@+id/ll"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/textView"
        android:layout_width="354dp"
        android:layout_height="55dp"
        android:text="TEMPERATURE CONVERTER"
        android:textSize="24sp"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.072" />

    <EditText
        android:id="@+id/Etn1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="Enter Temperature"
        android:inputType="number"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.497"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
```

```
    app:layout_constraintVertical_bias="0.297" />

<RadioGroup
    android:id="@+id/RG1"
    android:layout_width="282dp"
    android:layout_height="90dp"
    android:orientation="vertical"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent">

    <RadioButton
        android:id="@+id/RB1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Celsius To Fahrenheit"
        android:textSize="20sp" />

    <RadioButton
        android:id="@+id/RB2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Fahrenheit To Celsius"
        android:textSize="20sp" />
</RadioGroup>

<Button
    android:id="@+id/Btn1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Convert"
    android:onClick="add"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.881"
```

```
/>

<TextView
    android:id="@+id/Txt1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text=""
    android:textSize="24sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.473"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.707" />

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

MainActivity.java

```
package com.example.prac1b;

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

import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.RadioButton;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
    public void add(View v)
    {
```

```
ConstraintLayout ll=(ConstraintLayout)findViewById(R.id.ll);
TextView result=(TextView)findViewById(R.id.Txt1);
EditText et1=(EditText)findViewById(R.id.Etn1);

//get value from edit text box and convert into double
double a=Double.parseDouble(String.valueOf(et1.getText()));
RadioButton cb=(RadioButton)findViewById(R.id.RB1);
RadioButton fb=(RadioButton)findViewById(R.id.RB2);

//check which radio button is checked
if(cb.isChecked())
{
    //change background colour

    //display conversion
    result.setText(f2c(a)+" degree C");
    //cb.setChecked(false);
    fb.setChecked(true);
}
else
{
    result.setText(c2f(a)+" degree F");
    //fb.setChecked(false);
    cb.setChecked(true);
}

//Celcius to Fahrenheit method
private double c2f(double c)
{
    return (c*9)/5+32;
}

//Fahrenheit to Celcius method
private double f2c(double f)
{
    return (f-32)*5/9;
}
}
```

OUTPUT: -

T-CONVO

TEMPERATURE CONVERTER

Enter Temperature

- Celsius To Fahrenheit
- Fahrenheit To Celsius

CONVERT

T-CONVO

TEMPERATURE CONVERTER

50

- Celsius To Fahrenheit
- Fahrenheit To Celsius

10.0 degree C

CONVERT

T-CONVO

TEMPERATURE CONVERTER

30

- Celsius To Fahrenheit
- Fahrenheit To Celsius

86.0 degree F

CONVERT

T-CONVO

TEMPERATURE CONVERTER

50

- Celsius To Fahrenheit
- Fahrenheit To Celsius

CONVERT

T-CONVO

TEMPERATURE CONVERTER

30

- Celsius To Fahrenheit
- Fahrenheit To Celsius

CONVERT

CONCLUSION: -

From this practical I have learned the implementation of application for Unit Conversion.

PRACTICAL 3

AIM: - Android program based on Intents

THEORY: -

GEO-LOCATION: -

One of the unique features of mobile applications is location awareness. Mobile users take their devices with them everywhere, and adding location awareness to your app offers users a more contextual experience. The location APIs available in Google Play services facilitate adding location awareness to your app with automated location tracking, wrong-side-of-the-street detection, geofencing, and activity recognition.

1) float distanceTo(Location dest)

Returns the approximate distance in meters between this location and the given location.

2) float getAccuracy()

Get the estimated accuracy of this location, in meters.

3) double getAltitude()

Get the altitude if available, in meters above sea level.

4) float getBearing()

Get the bearing, in degrees.

5) double getLatitude()

Get the latitude, in degrees.

6) double getLongitude()

Get the longitude, in degrees.

1) abstract void onConnected(Bundle connectionHint)

This callback method is called when location service is connected to the location client successfully. You will use connect() method to connect to the location client.

2) abstract void onDisconnected()

This callback method is called when the client is disconnected. You will use disconnect() method to disconnect from the location client.

3) abstract void onConnectionFailed(ConnectionString result)

This callback method is called when there was an error connecting the client to the service.

INTENTS: -

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.

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.

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

MESSAGE: -

This article is about sending a text SMS over the phone using the SMSManager class in an Android application. For this, a basic knowledge of the fundamentals of android app development, creating a new project, running an android app, Views, and handling of click event buttons is required.

SMSManager class manages operations like sending a text message, data message, and multimedia messages (MMS). For sending a text message method sendTextMessage() is used likewise for multimedia message sendMultimediaMessage() and for data message sendDataMessage() method is used. The details of each function are:

sendTextMessage()

```
sendTextMessage(String destinationAddress,  
String scAddress, String text, PendingIntent sentIntent,  
PendingIntent deliveryIntent, long messageId)
```

sendDataMessage()

```
sendDataMessage(String destinationAddress,  
String scAddress, short destinationPort, byte[] data,  
PendingIntent sentIntent, PendingIntent deliveryIntent)
```

sendMultimediaMessage()

```
sendMultimediaMessage(Context context,  
Uri contentUri, String locationUrl,  
Bundle configOverrides, PendingIntent sentIntent)
```

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">

<EditText
    android:id="@+id/ed1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:ems="10"
    android:hint="Enter URL of the Website"
    android:inputType="textPersonName"
    android:minHeight="48dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.532"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.205" />

<Button
    android:id="@+id/wb"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="WEBSITE"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.361" />

<Button
    android:id="@+id/loc"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="LOCATION"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.67" />
```

```
<EditText
    android:id="@+id/ed2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:ems="10"
    android:hint="Enter the Location"
    android:inputType="textPersonName"
    android:minHeight="48dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<EditText
    android:id="@+id/ed3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:ems="10"
    android:hint="Enter the Message"
    android:inputType="textPersonName"
    android:minHeight="48dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.502"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.793" />

<Button
    android:id="@+id/msg"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="MESSAGE"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.935" />
```

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

MainActivity.java

```
package com.example.prac3;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Button wb,loc,msg;
        EditText ed1,ed2,ed3;
        wb=findViewById(R.id.wb);
        loc=findViewById(R.id.loc);
        msg=findViewById(R.id.msg);
        ed1=findViewById(R.id.ed1);
        ed2=findViewById(R.id.ed2);
        ed3=findViewById(R.id.ed3);
        wb.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String url = ed1.getText().toString();
                Intent urlIntent = new Intent(Intent.ACTION_VIEW, Uri.parse(url));
                startActivity(urlIntent);
            }
        });
        loc.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String map = ed2.getText().toString();
                Uri gmmIntentUri = Uri.parse("geo:0,0?q="+map);
```

```
Intent mapIntent = new Intent(Intent.ACTION_VIEW, gmmIntentUri);
mapIntent.setPackage("com.google.android.apps.maps");
startActivity(mapIntent);

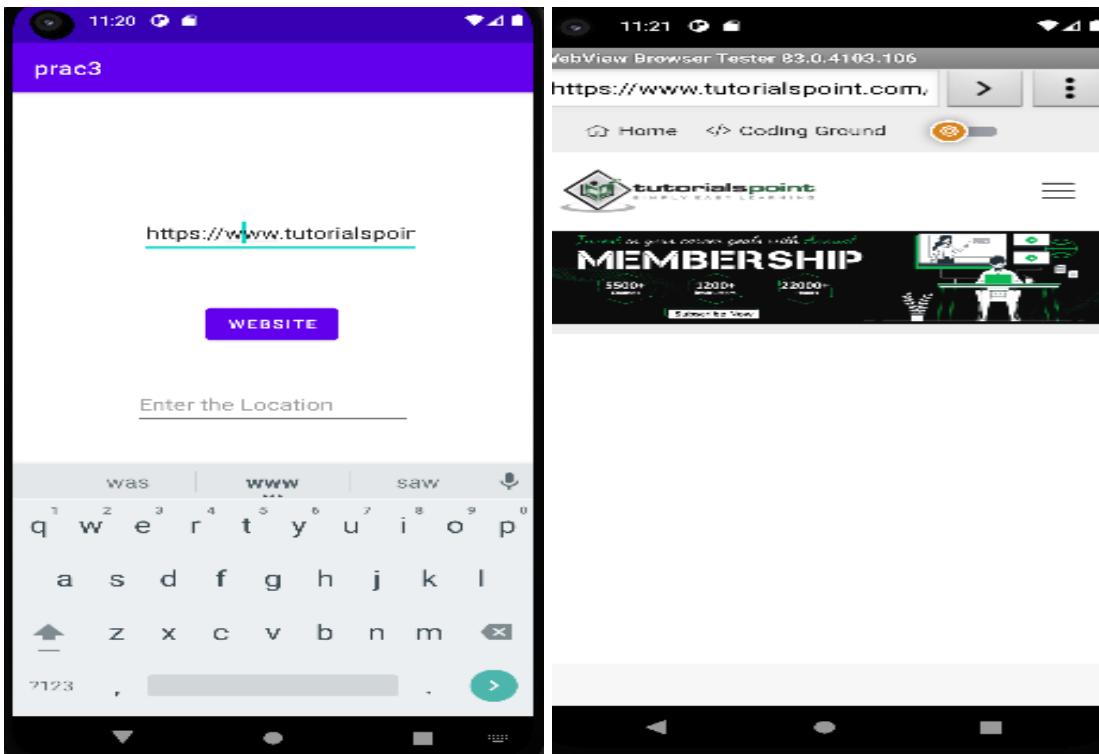
}

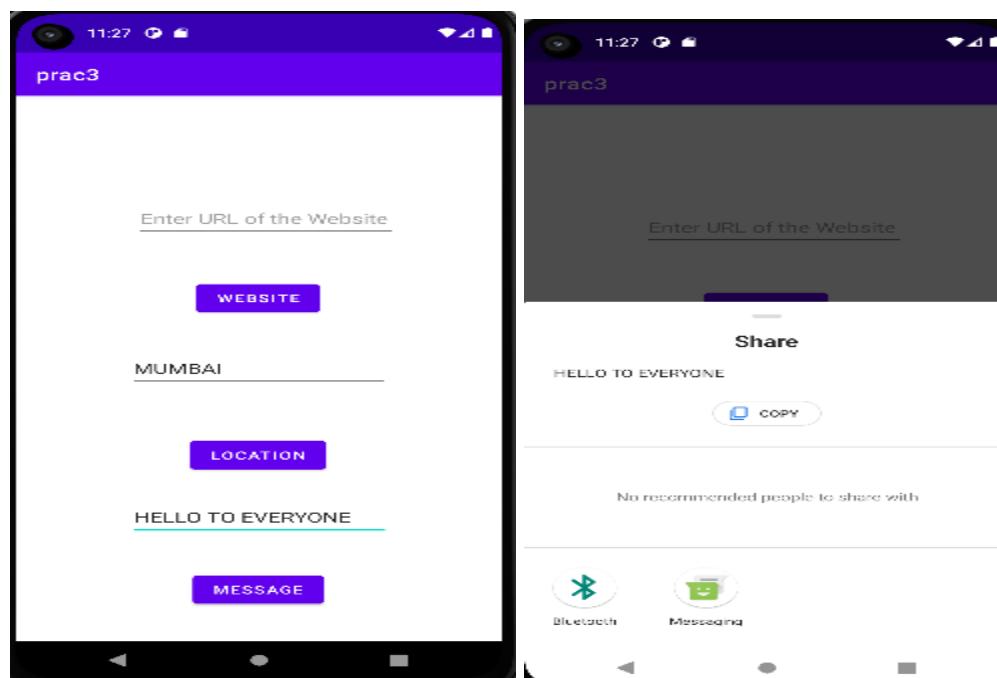
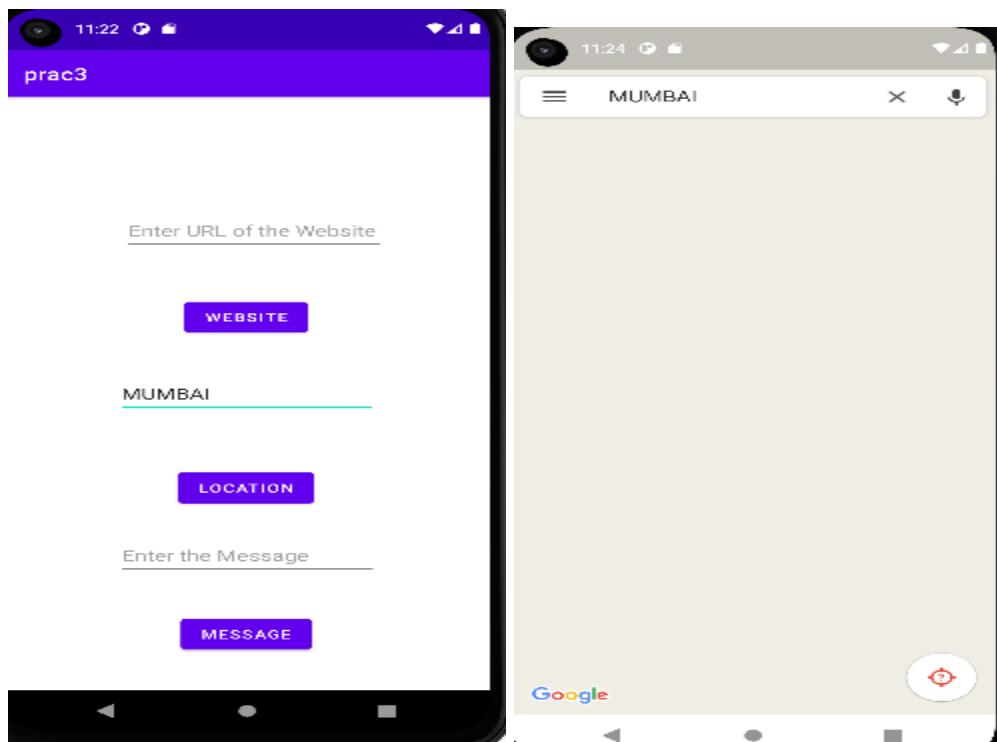
});

msg.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        String textMsg = ed3.getText().toString();
        Intent intent2 = new Intent();
        intent2.setAction(Intent.ACTION_SEND);
        intent2.setType("text/plain");
        intent2.putExtra(Intent.EXTRA_TEXT, textMsg );
        startActivity(Intent.createChooser(intent2, "Share via"));

    }
});
```

OUTPUT: -





CONCLUSION: -

From this practical I have learned to implement Android program based on Intents.

PRACTICAL 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.

Set Android Notification Properties

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.

Create and Send Notifications

Step 1 - Create Notification Builder

As a first step is to create a notification builder using `NotificationCompat.Builder.build()`.

```
NotificationCompat.Builder mBuilder = new NotificationCompat.Builder(this);
```

Step 2 - Setting Notification Properties

Once you have `Builder` object, you can set its Notification properties using `Builder` object as per your requirement.

```
mBuilder.setSmallIcon(R.drawable.notification_icon);
```

```
mBuilder.setContentTitle("Notification Alert, Click Me!");
```

```
mBuilder.setContentText("Hi, This is Android Notification Detail!");
```

Step 3 - Attach Actions

This is an optional part and required if you want to attach an action with the notification. An action allows users to go directly from the notification to an Activity in your application, where they can look at one or more events or do further work.

The action is defined by a PendingIntent containing an Intent that starts an Activity in your application.

For example, if you want to start Activity when the user clicks the notification text in the notification drawer, you add the PendingIntent by calling setContentIntent().

A PendingIntent object helps you to perform an action on your applications behalf, often at a later time, without caring of whether or not your application is running.

Step 4 - Issue the notification

Finally, you pass the Notification object to the system by calling NotificationManager.notify() to send your notification. Make sure you call NotificationCompat.Builder.build() method on builder object before notifying it.

```
NotificationManager mNotificationManager = (NotificationManager)  
        getSystemService(Context.NOTIFICATION_SERVICE);  
  
        // notificationID allows you to update the notification later on.  
mNotificationManager.notify(notificationID, mBuilder.build());
```

ANDROID 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, and Action Button.

Alert Dialog code has three methods:

setTitle() method for displaying the Alert Dialog box Title

setMessage() method for displaying the message

setIcon() method is used to set the icon on the Alert dialog box.

Step By Step Implementation

Step 1: Create a New Project in Android Studio

To create a new project in Android Studio please refer to How to Create/Start a New Project in Android Studio. The code for that has been given in both Java and Kotlin Programming Language for Android.

Step 2: Working with the XML Files

Next, go to the activity_main.xml file, which represents the UI of the project. Below is the code for the activity_main.xml file. Comments are added inside the code to understand the code in more detail.

Step 3: Working with the MainActivity File

Go to the MainActivity File and refer to the following code. Below is the code for the MainActivity File. Comments are added inside the code to understand the code in more detail.

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">

    <Button
        android:id="@+id/not"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="NOTIFY"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.275" />

    <Button
        android:id="@+id/alt"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="468dp"
        android:text="ALERT"
```

```
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

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

MainActivity.java

```
package com.example.prac4new;

import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.NotificationCompat;
import androidx.core.app.NotificationManagerCompat;

import android.app.Notification;
import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Build;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
    Button sendNotice,alert;
    AlertDialog.Builder builder;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        sendNotice = (Button) findViewById(R.id.not);
        alert = (Button) findViewById(R.id.alt);
        if(Build.VERSION.SDK_INT >= Build.VERSION_CODES.O){
            NotificationChannel channel= new NotificationChannel("My Notification","My
Notification",NotificationManager.IMPORTANCE_HIGH);
```

```

NotificationManager manager =getSystemService(NotificationManager.class);
manager.createNotificationChannel(channel);
}
sendNotice.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        String message="Hello Programming Digest";

        NotificationCompat.Builder builder = new
NotificationCompat.Builder(MainActivity.this,"My Notification");
        builder.setContentTitle("NotificationTitle");
        builder.setContentText(message);
        builder.setSmallIcon(R.drawable.ic_baseline_notifications_24);
        builder.setAutoCancel(true);
        Intent intent = new Intent(MainActivity.this, Notifyactivity.class);
        intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        intent.putExtra("message",message);
        PendingIntent
pendingIntent=PendingIntent.getActivity(MainActivity.this,0,intent,PendingIntent.FLAG_UPDA
TE_CURRENT);
        builder.setContentIntent(pendingIntent);
builder.setPriority(Notification.PRIORITY_MAX);

        NotificationManagerCompat
managerCompat=NotificationManagerCompat.from(MainActivity.this);
        managerCompat.notify(1,builder.build());

    }
});

builder = new AlertDialog.Builder(this);
alert.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

        //Setting message manually and performing action on button 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("EXIT");
alert.show();
}
});
```

activity notifyactivity.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=".Notifyactivity">

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

```
    android:layout_height="wrap_content"
    android:textSize="24sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Notifyactivity.java

```
package com.example.prac4new;

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

public class Notifyactivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_notifyactivity);
        TextView textView=(TextView) findViewById(R.id.txt);
        String message=getIntent().getStringExtra("message");
        textView.setText(message);

    }
}
```

OUTPUT: -

NOTIFY-ALERT

NOTIFY-ALERT

NotificationTitle • Now
Hello Programming Digest

NOTIFY

NOTIFY

ALERT

ALERT

NOTIFY-ALERT

Hello Programming Digest

NOTIFY-ALERT

NOTIFY

EXIT

Do you want to close this
application ?

NO YES

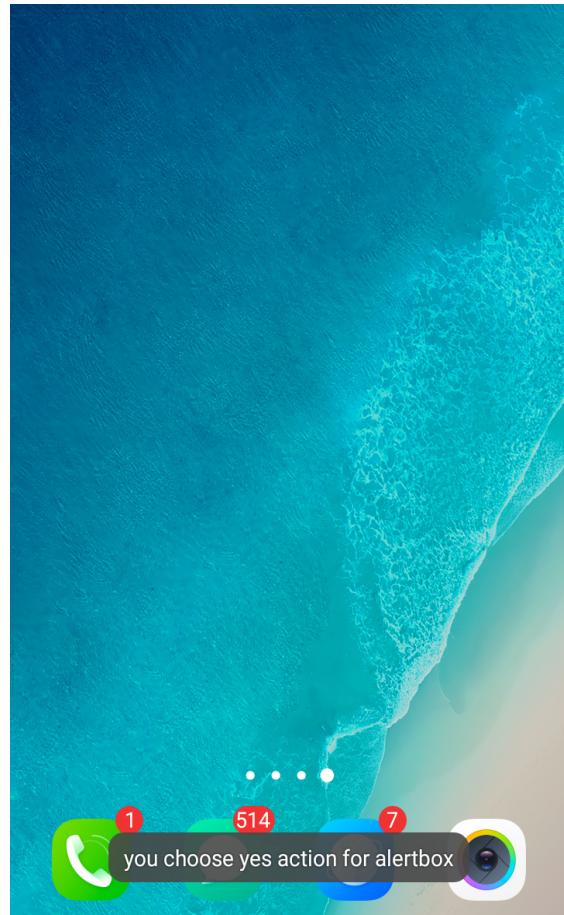
ALERT

NOTIFY-ALERT

NOTIFY

ALERT

you choose no action for alertbox



CONCLUSION: -

From this practical I have learned the implementation of Android program for 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

STEPS: -

1) activity_main.xml contains the code for the UI

2) The code for below files are written in code section

- MainActivity.java
- MyDBHandler.java
- Student.java

3) The database SQLite is in the Android since newest version.

4) Check for newest version.

5) In device file explorer section we can click and goto the project and expand data

6) The database can be viewed in the data section there.

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">

    <EditText
        android:id="@+id/et1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="Enter your ROLL NO"
        android:inputType="textPersonName"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.338"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.072" />

    <TextView
        android:id="@+id/tv1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="ROLL NO: -"
        android:textSize="16sp"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.205"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.022" />

    <TextView
```

```
    android:id="@+id/tv2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="NAME: -"
    android:textSize="16sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.194"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.159" />
```

```
<EditText
    android:id="@+id/et2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:ems="10"
    android:inputType="textPersonName"
    android:hint="Enter your Name"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.338"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.214" />
```

```
<TextView
    android:id="@+id/tv3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="EMAIL-ID: -"
    android:textSize="16sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.214"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.29" />
```

```
<EditText
    android:id="@+id/et3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:ems="10"
    android:inputType="textPersonName"
    android:hint="Enter your Email"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.338"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.346" />
```

```
<TextView
    android:id="@+id/tv4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="COURSE: -"
    android:textSize="16sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.202"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.424" />
```

```
<EditText
    android:id="@+id/et4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:ems="10"
    android:inputType="textPersonName"
    android:hint="Enter your Course"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.348"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
```

```
app:layout_constraintVertical_bias="0.483" />

<TextView
    android:id="@+id/tv5"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="CONTACT NO: -"
    android:textSize="16sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.235"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.555" />

<EditText
    android:id="@+id/et5"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:ems="10"
    android:hint="Enter your Contact No"
    android:inputType="phone"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.348"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.619" />

<TextView
    android:id="@+id/tv6"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="ADDRESS: -"
    android:textSize="16sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.208"
```

```
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.681" />
```

```
<EditText
    android:id="@+id/et6"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:ems="10"
    android:inputType="textPersonName"
    android:hint="Enter your Address"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.338"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.752" />
```

```
<Button
    android:id="@+id/load"
    android:layout_width="96dp"
    android:layout_height="49dp"
    android:text="LOAD"
    android:onClick="loadStudents"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.852" />
```

```
<Button
    android:id="@+id/upd"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="UPDATE"
    android:onClick="updateStudent"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.996"
```

```
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.852" />
```

```
<Button
    android:id="@+id/ins"
    android:layout_width="101dp"
    android:layout_height="47dp"
    android:text="INSERT"
    android:onClick="addStudent"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.345"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.849" />
```

```
<Button
    android:id="@+id/del"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="DELETE"
    android:onClick="deleteStudent"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.672"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.852" />
```

```
<TextView
    android:id="@+id/tv8"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="TextView"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="parent"
```

```
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.924" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java

```
package com.example.prac5;

import android.os.Bundle;
import android.text.method.ScrollingMovementMethod;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    //TextView resultText;
    EditText rno;
    EditText name;
    EditText email;
    EditText course;
    EditText contact;
    EditText address;
    TextView output1;
    MyDBHandler dbHandler;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        rno = (EditText) findViewById(R.id.et1);
        name = (EditText) findViewById(R.id.et2);
        email = (EditText) findViewById(R.id.et3);
        course = (EditText) findViewById(R.id.et4);
        contact = (EditText) findViewById(R.id.et5);
        address = (EditText) findViewById(R.id.et6);
        output1 = findViewById(R.id.tv8);
        //output1.setMovementMethod(new ScrollingMovementMethod());
        dbHandler = new MyDBHandler(this);
    }
}
```

```

public void loadStudents(View view) {
    output1.setText(dbHandler.loadHandler());
    rno.setText("");
    name.setText("");
    email.setText("");
    course.setText("");
    contact.setText("");
    address.setText("");
}

public void addStudent (View view) {
    if(!rno.getText().toString().isEmpty() && !name.getText().toString().isEmpty() &&
    !email.getText().toString().isEmpty() && !course.getText().toString().isEmpty() &&
    !contact.getText().toString().isEmpty() && !address.getText().toString().isEmpty()) {
        int id = Integer.parseInt(rno.getText().toString());
        String nm = name.getText().toString();
        String em = email.getText().toString();
        String cr = course.getText().toString();
        int cn = Integer.parseInt(contact.getText().toString());
        String ad = address.getText().toString();
        Student student = new Student(id, nm, em, cr, cn, ad);

        long insertId=dbHandler.addHandler(student);
        if(insertId== -1){
            output1.setText("Record already exists");
        }
        else{
            rno.setText("");
            name.setText("");
            email.setText("");
            course.setText("");
            contact.setText("");
            address.setText("");
            output1.setText("Record added");
        }
    }
    else{
        output1.setText("Please fill correct id and name");
    }
}

```

```

public void updateStudent(View view) {
    if(!rno.getText().toString().isEmpty() && !name.getText().toString().isEmpty() &&
!email.getText().toString().isEmpty() && !course.getText().toString().isEmpty() &&
!contact.getText().toString().isEmpty() && !address.getText().toString().isEmpty()) {
        boolean result = dbHandler.updateHandler(Integer.parseInt(rno.getText().toString()),
name.getText().toString(), email.getText().toString(), course.getText().toString(),
Integer.parseInt(contact.getText().toString()), address.getText().toString());
        if (result) {
            rno.setText("");
            name.setText("");
            email.setText("");
            course.setText("");
            contact.setText("");
            address.setText("");
            output1.setText("Record Updated");
        } else {
            output1.setText("No Record Found");
        }
    } else{
        output1.setText("Please fill correct id and name");
    }
}

```

```

public void deleteStudent(View view) {
    if(!rno.getText().toString().isEmpty()) {
        boolean result = dbHandler.deleteHandler(Integer.parseInt(
rno.getText().toString()));
        if (result) {
            rno.setText("");
            name.setText("");
            email.setText("");
            course.setText("");
            contact.setText("");
            address.setText("");
            output1.setText("Record Deleted");
        } else {
            output1.setText("No Record Found");
        }
    } else{

```

```

        output1.setText("Please fill correct id");
    }
}

@Override
protected void onDestroy() {
    super.onDestroy();
    dbHandler.close();
}

}

```

MyDBHandler.java

```

package com.example.prac5;

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

public class MyDBHandler extends SQLiteOpenHelper {

    private static final int DATABASE_VERSION = 3;
    private static final String DATABASE_NAME = "studentDB.db";
    private static final String TABLE_STUDENTS = "newstudent";
    private static final String COLUMN_RNO = "rno";
    private static final String COLUMN_NAME = "name";
    private static final String COLUMN_EMAIL = "email";
    private static final String COLUMN.Course = "course";
    private static final String COLUMN_CONTACT = "contact";
    private static final String COLUMN_ADDRESS = "address";
}

```

MyDBHandler(Context context)

```

{
    super(context, DATABASE_NAME, null, DATABASE_VERSION);
}

```

@Override

```

public void onCreate(SQLiteDatabase db) {
    String CREATE_STUDENT_TABLE = "CREATE TABLE " +
        TABLE_STUDENTS + "(" + COLUMN_RNO + " INTEGER PRIMARY KEY," +
        COLUMN_NAME +
        + " TEXT, " + COLUMN_EMAIL + " TEXT, " + COLUMN_COURSE + " TEXT, " +
        COLUMN_CONTACT + " INTEGER, " + COLUMN_ADDRESS + " TEXT" + ")";
    db.execSQL(CREATE_STUDENT_TABLE);
}

@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion,
        int newVersion) {
    db.execSQL("DROP TABLE IF EXISTS " + TABLE_STUDENTS);
    onCreate(db);
}

String loadHandler() {
    String result = "";
    String query = "Select*FROM " + TABLE_STUDENTS;
    SQLiteDatabase db = this.getWritableDatabase();
    Cursor cursor = db.rawQuery(query, null);
    while (cursor.moveToNext()) {
        int result_0 = cursor.getInt(0);
        String result_1 = cursor.getString(1);
        String result_2 = cursor.getString(2);
        String result_3 = cursor.getString(3);
        int result_4 = cursor.getInt(4);
        String result_5 = cursor.getString(5);
        result += String.valueOf(result_0) + " " + result_1 + " " + result_2 + " " + result_3 + " " +
        String.valueOf(result_4)+" " + result_5;
        System.getProperty("line.separator");
    }
    cursor.close();
    db.close();
    if(result.equals(""))
        result="No Record Found";
    return result;
}

long addHandler(Student student) {

```

```

long id;
ContentValues values = new ContentValues();
values.put(COLUMN_RNO, student.getID());
values.put(COLUMN_NAME, student.getName());
values.put(COLUMN_EMAIL, student.getEmail());
values.put(COLUMN_COURSE, student.getCourse());
values.put(COLUMN_CONTACT, student.getContact());
values.put(COLUMN_ADDRESS, student.getAddress());
SQLiteDatabase db = this.getWritableDatabase();
id = db.insert(TABLE_STUDENTS, null, values);
db.close();
return id;
}

boolean updateHandler(int rno, String name, String email, String course, int contact, String address) {
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues args = new ContentValues();
    args.put(COLUMN_RNO, rno);
    args.put(COLUMN_NAME, name);
    args.put(COLUMN_EMAIL, email);
    args.put(COLUMN_COURSE, course);
    args.put(COLUMN_CONTACT, contact);
    args.put(COLUMN_ADDRESS, address);
    return db.update(TABLE_STUDENTS, args, COLUMN_RNO + "=" + rno, null) > 0;
}

boolean deleteHandler(int ID) {
    boolean result = false;
    String query = "Select*FROM " + TABLE_STUDENTS + " WHERE " + COLUMN_RNO +
    " = " + String.valueOf(ID) + "";
    SQLiteDatabase db = this.getWritableDatabase();
    Cursor cursor = db.rawQuery(query, null);
    Student student = new Student();
    if (cursor.moveToFirst()) {
        student.setID(Integer.parseInt(cursor.getString(0)));
        db.delete(TABLE_STUDENTS, COLUMN_RNO + "=?", 
        new String[] {
            String.valueOf(student.getID())
        });
    }
}

```

```
        cursor.close();
        result = true;
    }
    db.close();
    return result;
}

}
```

Student.java

```
package com.example.prac5;
```

```
public class Student {
    private int rno;
    private String name;
    private String email;
    private String course;
    private int contact;
    private String address;
```

```
Student() {
}
```

```
Student(int rno, String name, String email, String course, int contact, String address) {
    this.rno = rno;
    this.name = name;
    this.email=email;
    this.course=course;
    this.contact=contact;
    this.address=address;
}
```

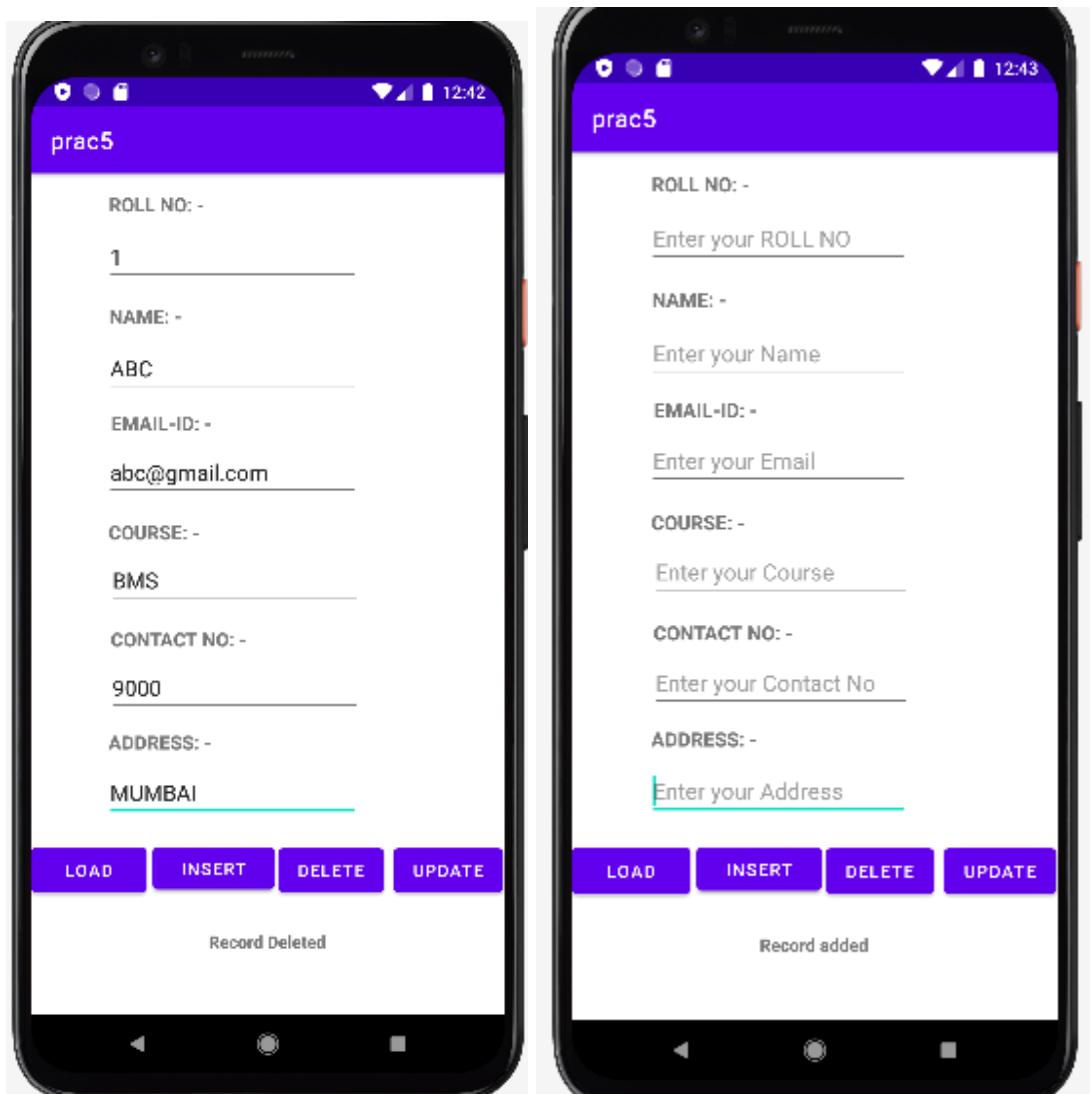
```
void setID(int id) {
    this.rno = id;
}
```

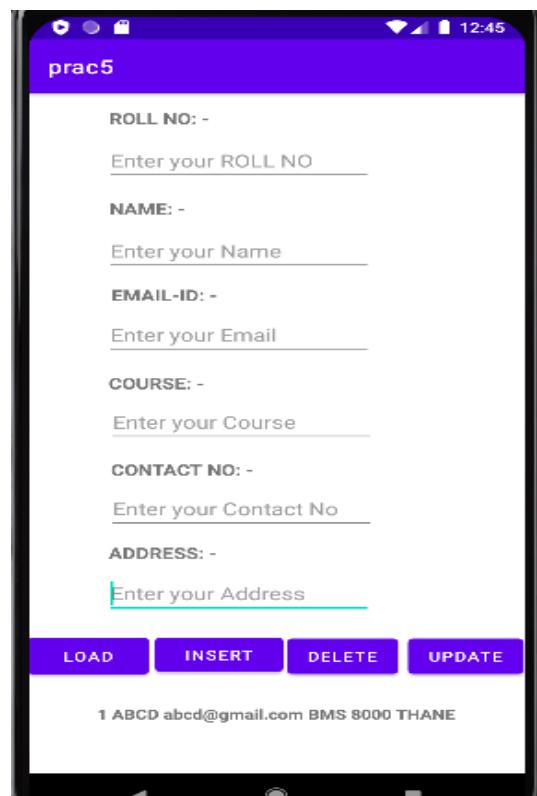
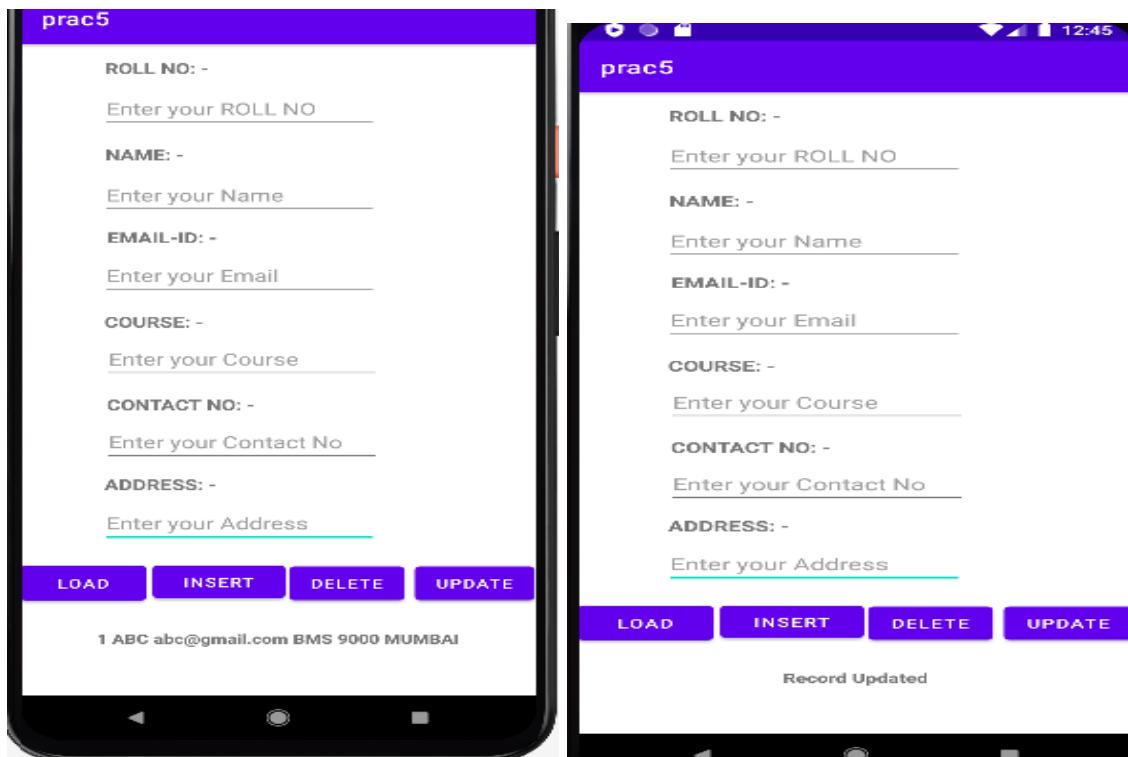
```
int getID() {
    return this.rno;
}
```

```
void setName(String name) {
    this.name = name;
}
```

```
String getName() {
    return this.name;
}
void setEmail(String email) {
    this.email = email;
}
String getEmail() {
    return this.email;
}
void setCourse(String course) {
    this.course = course;
}
String getCourse() {
    return this.course;
}
void setContact(int contact) {
    this.contact = contact;
}
int getContact() {
    return this.contact;
}
void setAddress(String address) {
    this.address = address;
}
String getAddress() {
    return this.address;
}
```

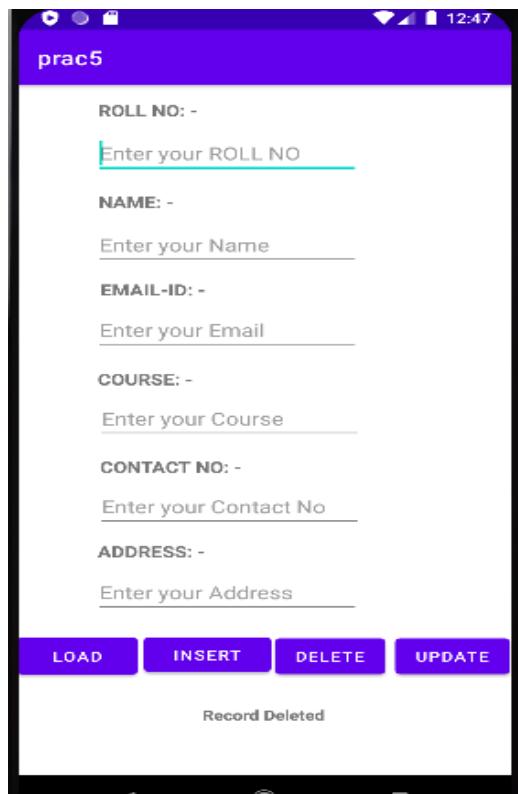
OUTPUT: -





Device File Explorer->Data Output

Tables Database Metadata					
Table: newstudent		Page:	Jump	<<	<
rno	name	email	course	contact	address
1	ABCD	abcd@gm...	BMS	8000	THANE



Device File Explorer->Data Output

Tables Database Metadata					
Table: newstudent		Page:	Jump	<<	<
rno	name	email	course	contact	address

CONCLUSION: -

From this practical I have learned to implement Android program to perform CRUD operation using SQLite DB

PRACTICAL 6

AIM: - Android program using shared preference ,Internal and External storage

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.

A) Program to create a file in a directory and perform following file operations, Write into a file, Read from a file, 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"
    tools:context=".MainActivity">
```

```
<EditText
    android:id="@+id/et2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:ems="10"
    android:hint="Enter File Name"
    android:inputType="textPersonName"
    android:minHeight="48dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.303"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.317" />
```

```
<TextView
    android:id="@+id/txt2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="FILE NAME: -"
    android:textSize="20sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.209"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.248" />
```

```
<TextView
    android:id="@+id/txt1"
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
    android:text="DIRECTORY NAME: -"
    android:textSize="20sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.27"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.053" />
```

```
<EditText
    android:id="@+id/et1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:ems="10"
    android:hint="Enter Directory Name"
    android:inputType="textPersonName"
    android:minHeight="48dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.303"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.113" />
```

```
<TextView
    android:id="@+id/txt3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="CONTENT: -"
    android:textSize="20sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.202"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.468" />
```

```
<EditText
    android:id="@+id/et3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:ems="10"
    android:hint="Enter File Content"
    android:inputType="textPersonName"
    android:minHeight="48dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.303"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.542" />
```

```
<Button
    android:id="@+id	btn1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="READ"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.102"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.664" />
```

```
<Button
    android:id="@+id	btn2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="WRITE"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.664" />
```

```
<Button
```

```
        android:id="@+id	btn3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="DELETE"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.9"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.664" />

<TextView
    android:id="@+id	txt4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="16sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.816" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java

```
package com.example.prac6;

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;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileOutputStream;
```

```
import java.io.FileReader;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        EditText edit = findViewById(R.id.et1);
        EditText edit2 = findViewById(R.id.et2);
        EditText edit3 = findViewById(R.id.et3);
        Button button = findViewById(R.id.btn1);
        Button button2 = findViewById(R.id.btn2);
        Button button3 = findViewById(R.id.btn3);
        TextView output = findViewById(R.id.txt4);
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                try {
                    StringBuilder result = new StringBuilder();
                    String line;
                    String folder = getApplication().getFilesDir().getAbsolutePath() + File.separator +
edit.getText().toString();
                    File subFolder = new File(folder);
                    BufferedReader bufferedReader = new BufferedReader(new FileReader(new
File(folder, edit2.getText().toString())));
                    while ((line = bufferedReader.readLine()) != null) {
                        result.append(line);
                    }
                    output.setText(result.toString());
                } catch (Exception e) {
                    Toast.makeText(getApplicationContext(), e.getMessage(),
Toast.LENGTH_LONG).show();
                }
            }
        });
        button2.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
```

```

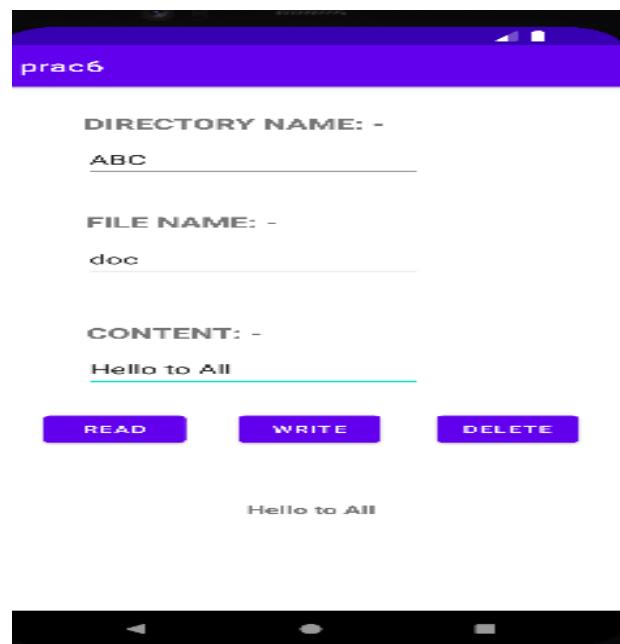
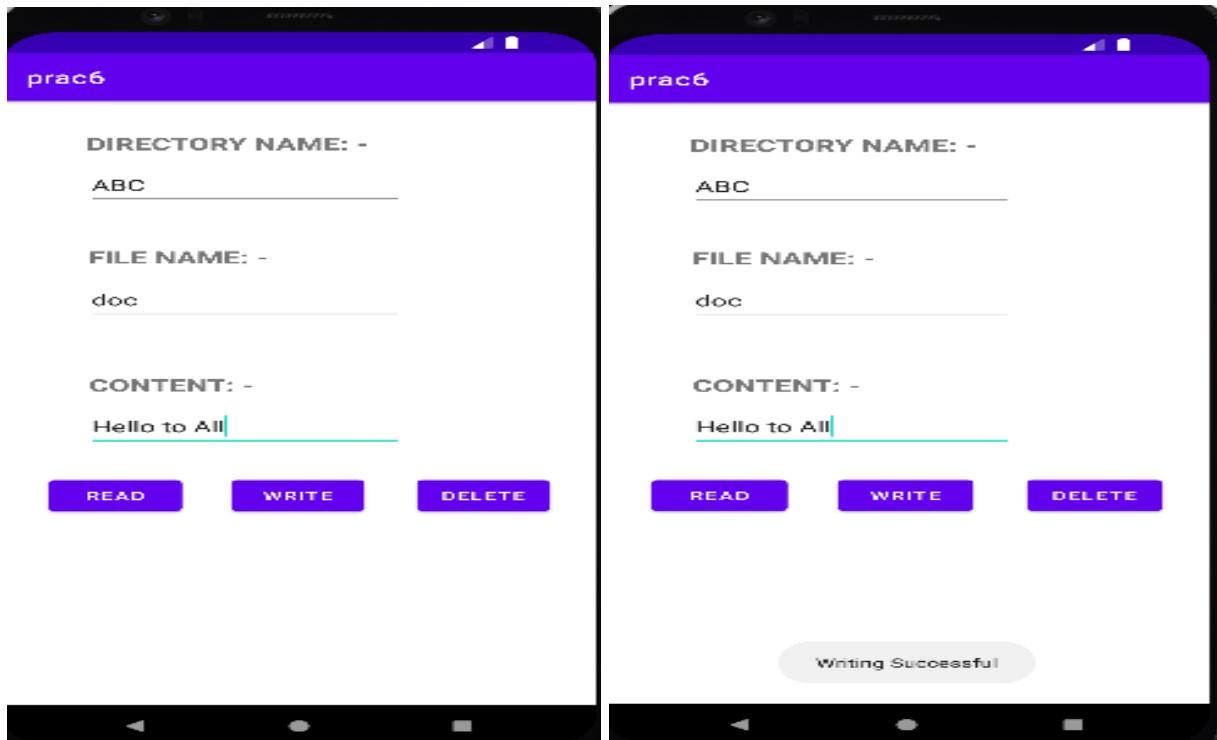
        try {
            String folder = getApplication().getFilesDir().getAbsolutePath() + File.separator +
edit.getText().toString();
            File subFolder = new File(folder);
            if (!subFolder.exists()) {
                subFolder.mkdirs();
            }
            FileOutputStream outputStream = new FileOutputStream(new File(subFolder,
edit2.getText().toString()));
            outputStream.write(edit3.getText().toString().getBytes());
            outputStream.close();
            Toast.makeText(getApplicationContext(), "Writing Successful",
Toast.LENGTH_LONG).show();
        } catch (Exception e) {
            Toast.makeText(getApplicationContext(), e.getMessage(),
Toast.LENGTH_LONG).show();
        }
    });

button3.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        try {
            String folder = getApplication().getFilesDir().getAbsolutePath() + File.separator +
edit.getText().toString();
            File subFolder = new File(folder);
            File file = new File(folder, edit2.getText().toString());
            if (file.exists()) {
                file.delete();
            }
            Toast.makeText(getApplicationContext(), "Deletion Successful",
Toast.LENGTH_LONG).show();
        } catch (Exception e) {
            Toast.makeText(getApplicationContext(), e.getMessage(),
Toast.LENGTH_LONG).show();
        }
    }
});

```

```
});  
}  
}  
}
```

OUTPUT: -



DEVICE FILE EXPLORER OUTPUT

com.example.prac6	drwxrwx--x	2022-09-23 14:22	4 KB
cache	drwxrws--x	2022-10-20 15:00	4 KB
code_cache	drwxrws--x	2022-10-20 15:00	4 KB
files	drwxrwx--x	2022-10-20 15:22	4 KB
ABC	drwx-----	2022-10-20 15:19	4 KB
doc	-rw-----	2022-10-20 15:21	12 B



DIRECTORY NAME: -

ABC

FILE NAME: -

doc

CONTENT: -

Enter File Content

READ

WRITE

DELETE

Deletion Successful



DEVICE FILE EXPLORER OUTPUT

com.example.prac6	drwxrwx--x	2022-09-23 14:22	4 KB
cache	drwxrws--x	2022-10-20 15:00	4 KB
code_cache	drwxrws--x	2022-10-20 15:00	4 KB
files	drwxrwx--x	2022-10-20 15:22	4 KB
ABC	drwx-----	2022-10-20 15:23	4 KB

B) 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"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/txt1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Enter Email: -"
        android:textSize="20sp"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.349"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.207" />

    <EditText
        android:id="@+id/et1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="Enter your email"
        android:inputType="textPersonName"
        android:minHeight="48dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.497"
```

```
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.268" />

<TextView
    android:id="@+id/txt2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Enter Password: -"
    android:textSize="20sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.4"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.375" />

<EditText
    android:id="@+id/et2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:ems="10"
    android:hint="Enter your password"
    android:inputType="textPersonName"
    android:minHeight="48dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.497"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.438" />

<CheckBox
    android:id="@+id/cb1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Remember Me"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
```

```
    app:layout_constraintHorizontal_bias="0.351"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.527" />

<Button
    android:id="@+id	btn1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="SIGN IN"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.74" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java

```
package com.example.prac6b;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.EditText;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

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

```

SharedPreferences sharedpreferences =
    this.getSharedPreferences("LoginDetails", Context.MODE_PRIVATE);
String e=sharedpreferences.getString("Email","");
if(e.length()>0)startHomeActivity();
EditText email =findViewById(R.id.et1);
EditText pass = findViewById(R.id.et2);
CheckBox cbbx = findViewById(R.id.cb1);
Button button = findViewById(R.id.btn1);

button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (cbbx.isChecked())
            saveLoginDetails(email.getText().toString(), pass.getText().toString());
        startHomeActivity();
    }
});

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

private void saveLoginDetails(String email, String password) {
    Toast.makeText(this,"pref",Toast.LENGTH_LONG);
    new Prefmanager(this).saveLoginDetails(email, password);
}

```

Prefmanager.java

```

package com.example.prac6b;

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();
}

```

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=".HomeActivity">

    <TextView
        android:id="@+id/txt3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="WELCOME TO APP"
        android:textSize="24sp"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.499" />
</androidx.constraintlayout.widget.ConstraintLayout>

```

HomeActivity.java

```
package com.example.prac6b;
```

```

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

public class HomeActivity extends AppCompatActivity {

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

```

OUTPUT: -



Enter Email: -

Enter your email

Enter Password: -

Enter your password

Remember Me

Enter Email: -

ABC

Enter Password: -

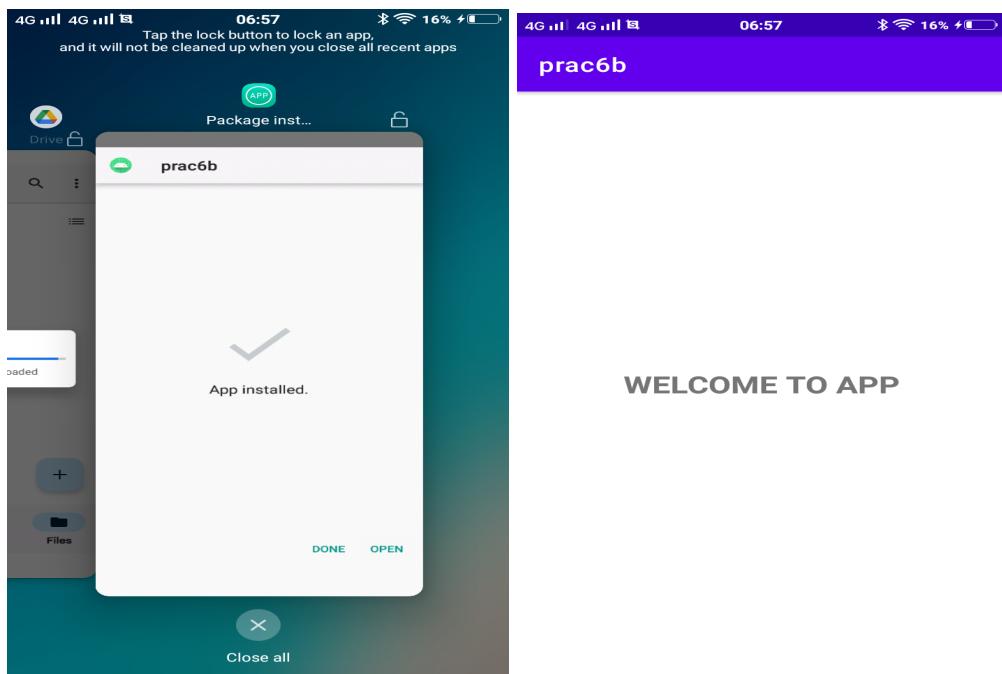
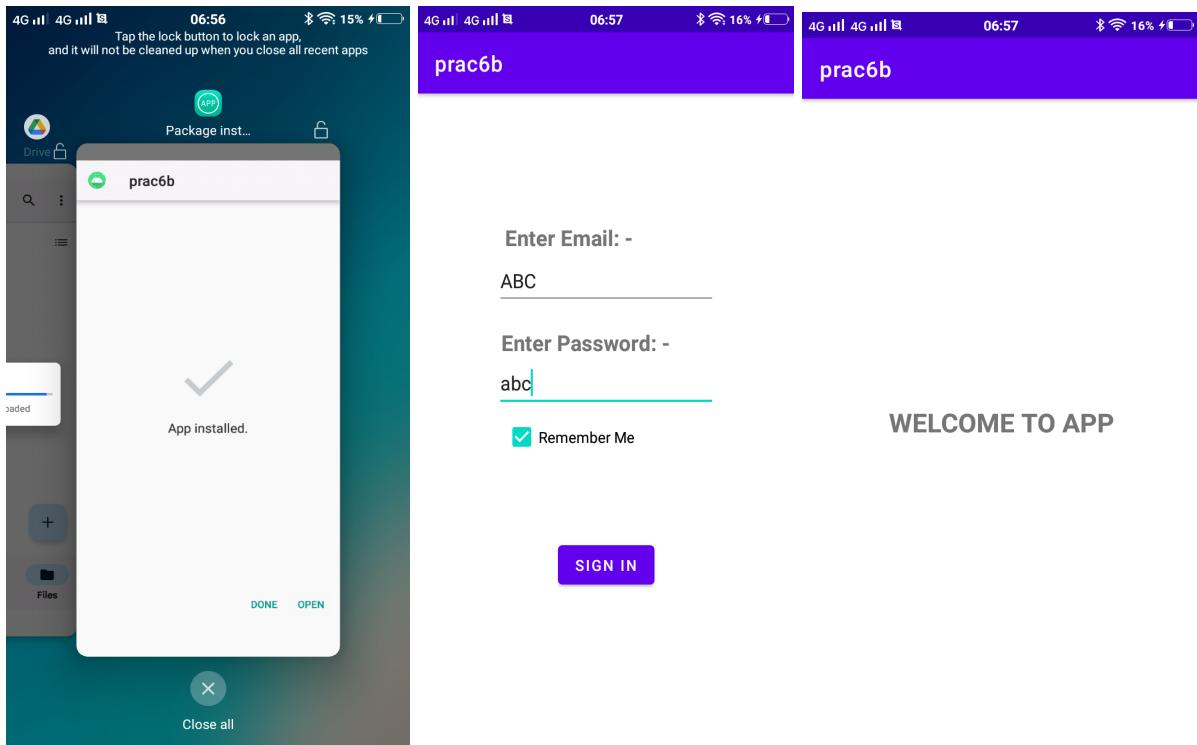
abc

Remember Me

WELCOME TO APP

SIGN IN

SIGN IN



CONCLUSION: -

From this practical I have learned to implement Android program using shared preference ,Internal and External storage.

PRACTICAL 7

AIM: - Android program to work with graphics and animation

THEORY: -

1) Tween Animation

Tween Animation takes some parameters such as start value , end value, size , time duration , rotation angle e.t.c and perform the required animation on that object. It can be applied to any type of object. So in order to use this , android has provided us a class called Animation.

In order to perform animation in android , we are going to call a static function loadAnimation() of the class AnimationUtils. We are going to receive the result in an instance of Animation Object. Its syntax is as follows –

```
Animation animation = AnimationUtils.loadAnimation(getApplicationContext(),  
    R.anim.myanimation);
```

Note the second parameter. It is the name of the our animation xml file. You have to create a new folder called anim under res directory and make an xml file under anim folder.

This animation class has many useful functions which are listed below –

Sr.No	Method & Description
1	<code>start()</code> This method starts the animation.
2	<code>setDuration(long duration)</code> This method sets the duration of an animation.

3	getDuration() This method gets the duration which is set by above method
4	end() This method ends the animation.
5	cancel() This method cancels the animation.

In order to apply this animation to an object , we will just call the startAnimation() method of the object. Its syntax is –

```
ImageView image1 = (ImageView)findViewById(R.id.imageView1);
image.startAnimation(animation);
```

2) Image Effects

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

Android provides Bitmap class to handle images. This can be found under android.graphics.bitmap. There are many ways through which you can instantiate bitmap. We are creating a bitmap of image from the imageView.

```
private Bitmap bmp;
private ImageView img;
img = (ImageView)findViewById(R.id.imageView1);
BitmapDrawable abmp = (BitmapDrawable)img.getDrawable();
```

Now we will create bitmap by calling getBitmap() function of BitmapDrawable class. Its syntax is given below –

```
bmp = abmp.getBitmap();
```

An image is nothing but a two dimensional matrix. Same way you will handle a bitmap. An image consist of pixels. So you will get pixels from this bitmap and apply processing to it. Its syntax is as follows –

```
for(int i=0; i<bmp.getWidth(); i++){
    for(int j=0; j<bmp.getHeight(); j++){
        int p = bmp.getPixel(i, j);
    }
}
```

The `getWidth()` and `getHeight()` functions returns the height and width of the matrix. The `getPixel()` method returns the pixel at the specified index. Once you got the pixel, you can easily manipulate it according to your needs.

Apart from these methods, there are other methods that helps us manipulate images more better.

Sr.N o	Method & description
1	<code>copy(Bitmap.Config config, boolean isMutable)</code> This method copy this bitmap's pixels into the new bitmap
2	<code>createBitmap(DisplayMetrics display, int width, int height, Bitmap.Config config)</code> Returns a mutable bitmap with the specified width and height
3	<code>createBitmap(int width, int height, Bitmap.Config config)</code> Returns a mutable bitmap with the specified width and height
4	<code>createBitmap(Bitmap src)</code> Returns an immutable bitmap from the source bitmap

5	<code>extractAlpha()</code> Returns a new bitmap that captures the alpha values of the original
6	<code>getConfig()</code> This method return that config, otherwise return null
7	<code>getDensity()</code> Returns the density for this bitmap
8	<code>getRowBytes()</code> Return the number of bytes between rows in the bitmap's pixels
9	<code>setPixel(int x, int y, int color)</code> Write the specified Color into the bitmap (assuming it is mutable) at the x,y coordinate
10	<code>setDensity(int density)</code> This method specifies the density for this bitmap

A) To perform the animation on an image and to apply various filters on an image.

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"
    tools:context=".MainActivity">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="153dp"
        android:layout_height="231dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.264"
        app:srcCompat="@drawable/doraemon" />

    <Button
        android:id="@+id	btn1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="MOVE"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.049"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.713" />
```

```

<Button
    android:id="@+id	btn2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="ROTATE"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.713" />

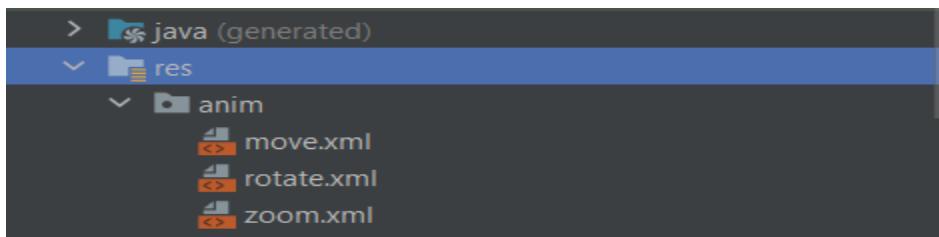
<Button
    android:id="@+id	btn3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="EXPAND"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.949"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.713" />
</androidx.constraintlayout.widget.ConstraintLayout>

```

Now we have to make xml files

STEPS: -

- 1) Right click on res→New→Android Resource Directory
- 2) After choosing new resource directory
- 3) Dialog box appears set it to anim from the list of values given
- 4) Now right click on anim folder→New→Anim resource file
- 5) Dialog box appears enter the file name
- 6) Make below 3 files by repeating step 4 and 5



move.xml

```
<?xml version="1.0" encoding="utf-8"?>
<set
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:interpolator="@android:anim/linear_interpolator"
    android:fillAfter="true">

    <translate
        android:fromXDelta="0%p"
        android:toXDelta="75%p"
        android:duration="700" />
</set>
```

rotate.xml

```
<?xml version="1.0" encoding="utf-8"?>
<set
    xmlns:android="http://schemas.android.com/apk/res/android">
    <rotate
        android:duration="6000"
        android:fromDegrees="0"
        android:pivotX="50%"
        android:pivotY="50%"
        android:toDegrees="360" />

    <rotate
        android:duration="6000"
        android:fromDegrees="360"
        android:pivotX="50%"
        android:pivotY="50%"
        android:startOffset="5000"
        android:toDegrees="0" />
</set>
```

zoom.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:fromXScale="0.5"  
    android:toXScale="3.0"  
    android:fromYScale="0.5"  
    android:toYScale="3.0"  
    android:duration="5000"  
    android:pivotX="50%"  
    android:pivotY="50%" >  
</scale>  
  
<scale xmlns:android="http://schemas.android.com/apk/res/android"  
    android:startOffset="5000"  
    android:fromXScale="3.0"  
    android:toXScale="0.5"  
    android:fromYScale="3.0"  
    android:toYScale="0.5"  
    android:duration="5000"  
    android:pivotX="50%"  
    android:pivotY="50%" >  
</scale>  
  
</set>
```

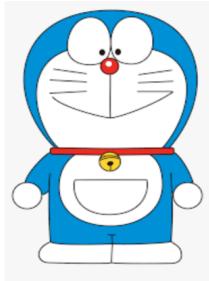
MainActivity.java

```
package com.example.prac7;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.os.Bundle;  
import android.view.View;  
import android.view.animation.Animation;  
import android.view.animation.AnimationUtils;  
import android.widget.Button;  
import android.widget.ImageView;  
  
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {
```

```
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
ImageView im = findViewById(R.id.imageView);
Button mvbtn = findViewById(R.id.btn1);
Button robtn = findViewById(R.id.btn2);
Button zobtn = findViewById(R.id.btn3);

mvbtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // To add move animation
        Animation animation = AnimationUtils.loadAnimation(getApplicationContext(),
R.anim.move);
        im.startAnimation(animation);
    }
});
robtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // To add rotate animation
        Animation animation = AnimationUtils.loadAnimation(getApplicationContext(),
R.anim.rotate);
        im.startAnimation(animation);
    }
});
zobtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // To add zoom animation
        Animation animation = AnimationUtils.loadAnimation(getApplicationContext(),
R.anim.zoom);
        im.startAnimation(animation);
    }
});}}
```

OUTPUT: -



MOVE

ROTATE

EXPAND

MOVE

ROTATE

EXPAND



MOVE

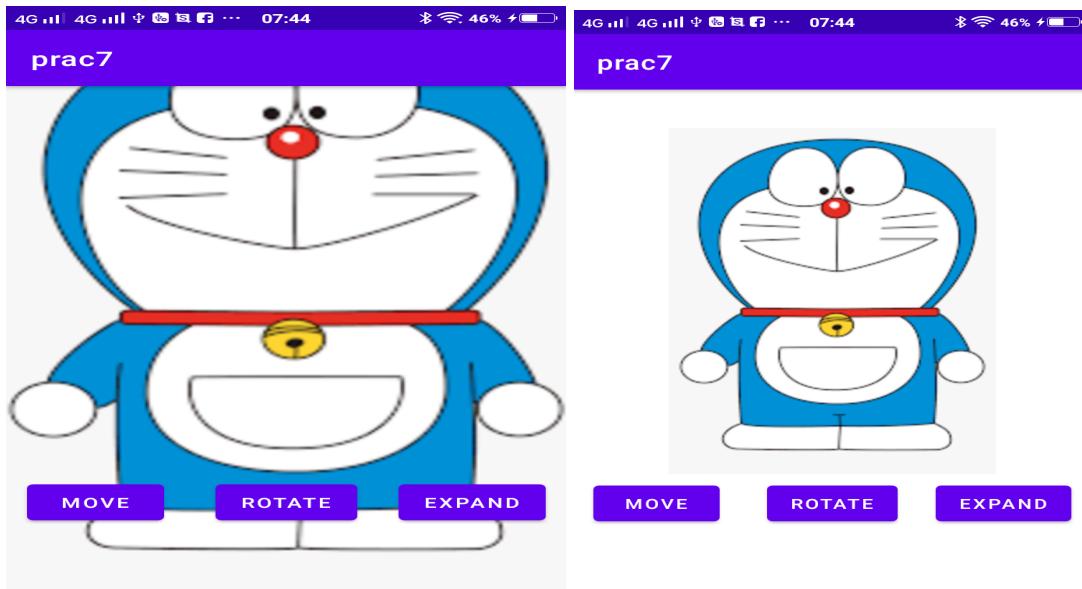
ROTATE

EXPAND

MOVE

ROTATE

EXPAND



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"
    tools:context=".MainActivity">
```

```
<ImageView
    android:id="@+id/iv1"
    android:layout_width="320dp"
    android:layout_height="356dp"
    app:layout_constraintBottom_toBottomOf="parent"
```

```
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.496"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.246"
    app:srcCompat="@drawable/scenaricview" />
```

```
<Button
    android:id="@+id	btn1"
    android:layout_width="138dp"
    android:layout_height="48dp"
    android:text="BRIGHTNESS"
    android:onClick="bright"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.542"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.707" />
```

```
<Button
    android:id="@+id	btn2"
    android:layout_width="138dp"
    android:layout_height="48dp"
    android:text="DARKNESS"
    android:onClick="dark"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.542"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.806" />
```

```
<Button
    android:id="@+id	btn3"
    android:layout_width="139dp"
    android:layout_height="48dp"
    android:text="GRAYSCALE"
    android:onClick="gray"
    app:layout_constraintBottom_toBottomOf="parent"
```

```
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.544"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.905" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java

```
package com.example.prac7b;

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 {
    private Bitmap bmp;
    private Bitmap operation;
    Button bright,dark,gray;
    ImageView im;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        bright = findViewById(R.id.btn1);
        dark = findViewById(R.id.btn2);
        gray = findViewById(R.id.btn3);
        im = findViewById(R.id.iv1);
        BitmapDrawable abmp = (BitmapDrawable) im.getDrawable();
        bmp = abmp.getBitmap();
    }
    public void bright(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));
    }
}
im.setImageBitmap(operation);
}

public void dark(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 = r - 50;
            g = g - 50;
            b = b - 50;
            alpha = alpha -50;
            operation.setPixel(i, j, Color.argb(Color.alpha(p), r, g, b));
        }
    }
    im.setImageBitmap(operation);
}

public void gray(View view) {
    operation = Bitmap.createBitmap(bmp.getWidth(),bmp.getHeight(), bmp.getConfig());
    double red = 0.33;
}

```

```

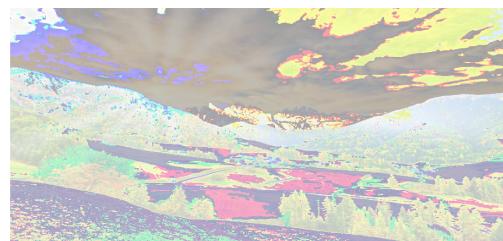
double green = 0.59;
double blue = 0.11;

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) red * r;
        g = (int) green * g;
        b = (int) blue * b;
        operation.setPixel(i, j, Color.argb(Color.alpha(p), r, g, b));
    }
}
im.setImageBitmap(operation);
}
}

```

OUTPUT: -



BRIGHTNESS

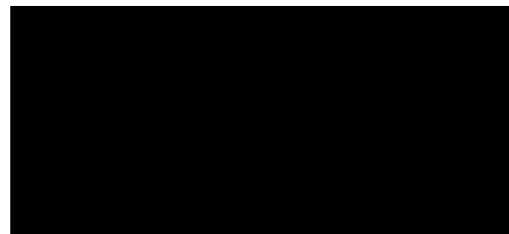
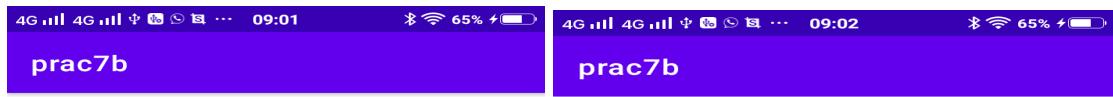
DARKNESS

GRAYSCALE

BRIGHTNESS

DARKNESS

GRAYSCALE



BRIGHTNESS

DARKNESS

GRAYSCALE

BRIGHTNESS

DARKNESS

GRAYSCALE

CONCLUSION: -

From this practical I have learned to implement Android program to work with graphics and animation.

PRACTICAL 8

AIM: - Android program to work with google maps and location

THEORY: -

Location

Android allows us to integrate google maps in our application. You can show any location on the map , or can show different routes on the map e.t.c. You can also customize the map according to your choices.

Sr.No	Method & description
1	<code>addCircle(CircleOptions options)</code> This method add a circle to the map
2	<code>addPolygon(PolygonOptions options)</code> This method add a polygon to the map
3	<code>addTileOverlay(TileOverlayOptions options)</code> This method add tile overlay to the map
4	<code>animateCamera(CameraUpdate update)</code> This method Moves the map according to the update with an animation
5	<code>clear()</code> This method removes everything from the map.

	getMyLocation()
6	This method returns the currently displayed user location.
	moveCamera(CameraUpdate update)
7	This method repositions the camera according to the instructions defined in the update
	setTrafficEnabled(boolean enabled)
8	This method Toggles the traffic layer on or off.
	snapshot(GoogleMap.SnapshotReadyCallback callback)
9	This method Takes a snapshot of the map
	stopAnimation()
10	This method stops the camera animation if there is one in progress

Calculate Distance

Developers, today, experiment and implement various kinds of features, including the feature to calculate the distance between two points or GPS coordinates in the Android mobile app. This can be achieved through Google Maps API. And it has made GPS Coordinate Distance Calculator easier to find how far two GPS coordinates converter is on a map.

If you are planning to develop a travel app, location tracking, taxi booking, or any on-demand delivery app, such a feature is very useful. Being a leading custom taxi booking solution developer, we have developed 50+ taxi booking apps and 40+ on-demand delivery apps. In most of these native applications, we have used this feature for real-time tracking a lot of times.

In this Android app article, we'll develop a demo app that will display the route between two locations.

SphericalUtil.computeDistanceBetween(loc1, loc2);

CODE: -

Building API

Steps: -

1) Goto below site

<https://developers.google.com/maps/documentation/android-sdk/get-api-key>

2) Click on Goto Credentials

Console Cloud SDK

1. Go to the [Google Maps Platform > Credentials](#) page.

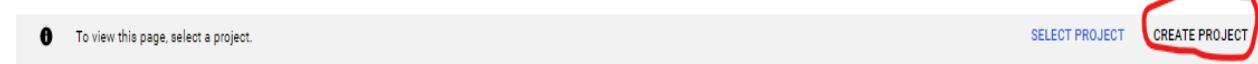
[Go to the Credentials page](#)

2. On the [Credentials](#) page, click [Create credentials > API key](#).
The [API key created](#) dialog displays your newly created API key.

3. Click [Close](#).
The new API key is listed on the [Credentials](#) page under [API keys](#).
(Remember to [restrict the API key](#) before using it in production.)



3) Click create project



4) Enter Details

New Project

You have 10 projects remaining in your quota. Request an increase or delete projects. [Learn more](#)

[MANAGE QUOTAS](#)

Project name * [?](#)

Project ID: abcd-366906. It cannot be changed later. [EDIT](#)

Organization * [?](#)

Select an organization to attach it to a project. This selection can't be changed later.

Location * [BROWSE](#)

Parent organization or folder

[CREATE](#) [CANCEL](#)

5) Click Create Credentials

Google Cloud ABCD Search for resources, docs, products, and more

Google Maps Platform Credentials All Google Maps Platform APIs [+ CREATE CREDENTIALS](#)

Overview APIs Metrics Quotas Credentials Support

To view all credentials visit [Credentials in APIs & Services](#)

⚠ Remember to configure the OAuth consent screen with information about your application

API Keys

Name	Creation date
No API keys to display	

6) Choose API Key

The screenshot shows the Google Cloud Platform interface for the Google Maps Platform. The left sidebar has a 'Google Maps Platform' icon and links for Overview, APIs, Metrics, Quotas, Credentials (which is selected and highlighted in blue), and Support. The main content area is titled 'Credentials' and shows 'All Google Maps Platform APIs'. A 'CREATE CREDENTIALS' button is at the top right. Below it, there are three options: 'API key' (selected), 'OAuth client ID', and 'Service account'. Each option has a brief description. A note says 'Remember to configure the OAuth consent screen' with a warning icon. On the right, a 'Help me choose' section asks questions to decide which type of credential to use.

7) Below dialog box appear with API Key

API key created

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

Your API key

⚠ This key is unrestricted. To prevent unauthorized use, we recommend restricting where and for which APIs it can be used. [Edit API key](#) to add restrictions. [Learn more](#)

[CLOSE](#)

8) You have to enable API by going to dashboard

APIs Section→Choose Maps SDK for Android→Choose Enable

Start your Free Trial with \$300 in credit. Don't worry—you won't be charged if you run out of credits. [Learn more](#)

☰ Google Cloud ABCD Search for resources, docs, projects

Google Maps Platform APIs

Select an API to view details. Figures are for the last 30 days.

API ↑	Requests
No rows to display	

Additional APIs

Select an API to view details in Marketplace

API
Maps SDK for Android
Directions API
Distance Matrix API
Maps Elevation API
Maps Embed API
Geocoding API

☰ Google Cloud ABCD ←

Maps SDK for Android

Google

Maps for your native Android app.

ENABLE

Click to enable this API

OVERVIEW DOCUMENTATION SUPPORT

build.gradle(:app)

```

dependencies {

    implementation 'androidx.appcompat:appcompat:1.5.1'
    implementation 'com.google.android.material:material:1.7.0'
    implementation 'com.google.android.gms:play-services-maps:18.1.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.1.4'
    implementation 'com.google.maps.android:android-maps-utils:2.3.0'
    testImplementation 'junit:junit:4.13.2'
    androidTestImplementation 'androidx.test.ext:junit:1.1.3'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.4.0'
}

}

```

Below line to be added in build.gradle which is above highlighted

```
implementation 'com.google.maps.android:android-maps-utils:2.3.0'
```

AndroidManifest.xml

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.example.testpractical8">
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.INTERNET" />
    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.Testpractical8"
        tools:targetApi="31">
        <meta-data
            android:name="com.google.android.geo.API_KEY"
            android:value="AIzaSyCmZqZXt2U-jRrSd1uSwXP8nI8EHSzo3RM" />
        <activity
            android:name=".MapsActivity"
            android:exported="true"

```

```

    android:label="@string/title_activity_maps">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

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

</manifest>

```

Following line should be added in AndroidManifest.xml refer above code

```

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

```

Between quotes add API KEY

In android:value=""

activity_maps.xml

```

<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=".MapsActivity">

    <fragment
        android:id="@+id/map"
        android:name="com.google.android.gms.maps.SupportMapFragment"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.0"

```

```
tools:context="net.simplifiedcoding.mymapapp.MapsActivity" />
```

```
<EditText  
    android:id="@+id/et1"  
    android:layout_width="160dp"  
    android:layout_height="44dp"  
    android:ems="10"  
    android:hint="Longitude"  
    android:inputType="textPersonName"  
    android:minHeight="48dp"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.0"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:layout_constraintVertical_bias="0.045" />
```

```
<EditText  
    android:id="@+id/et2"  
    android:layout_width="160dp"  
    android:layout_height="52dp"  
    android:ems="10"  
    android:hint="Latitude"  
    android:inputType="textPersonName"  
    android:minHeight="48dp"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.0"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:layout_constraintVertical_bias="0.153" />
```

```
<Button  
    android:id="@+id	btn1"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="LOCATOR"  
  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"
```

```
    app:layout_constraintHorizontal_bias="0.051"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.26" />

<EditText
    android:id="@+id/etn3"
    android:layout_width="192dp"
    android:layout_height="42dp"
    android:ems="10"
    android:inputType="textPersonName"
    android:hint="2nd Longitude"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.904"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.044" />

<EditText
    android:id="@+id/etn4"
    android:layout_width="188dp"
    android:layout_height="50dp"
    android:ems="10"
    android:inputType="textPersonName"
    android:hint="2nd Latitude"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.905"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.155" />

<Button
    android:id="@+id	btn2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="DISTANCE"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="@+id/map"
```

```
    app:layout_constraintHorizontal_bias="0.768"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.26" />

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

MapsActivity.java

```
package com.example.testpractical8;

import androidx.fragment.app.FragmentActivity;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

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.example.testpractical8.databinding.ActivityMapsBinding;
import com.google.maps.android.SphericalUtil;

public class MapsActivity extends FragmentActivity implements OnMapReadyCallback {

    private GoogleMap mMap;
    private ActivityMapsBinding binding;
    LatLng loc1;
    LatLng loc2;
    Double distance;
    EditText edit1,edit2,edit3,edit4;
    double lon,lat,lon1,lat1;
    String a,b,c,d;
```

```

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

    setContentView(R.layout.activity_maps);
    Button button=findViewById(R.id.btn1);
    Button button2=findViewById(R.id.btn2);
    edit1=findViewById(R.id.et1);
    edit2=findViewById(R.id.et2);
    edit3=findViewById(R.id.etn3);
    edit4=findViewById(R.id.etn4);
    // Obtain the SupportMapFragment and get notified when the map is ready to be used.
    SupportMapFragment mapFragment = (SupportMapFragment)
getSupportFragmentManager()
        .findFragmentById(R.id.map);
    mapFragment.getMapAsync(this);
    button.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View v) {
            a= edit1.getText().toString();
            b= edit2.getText().toString();

            lon=Double.parseDouble(a);
            lat=Double.parseDouble(b);

            LatLng sydney = new LatLng(lon,lat);
            mMap.addMarker(new MarkerOptions().position(sydney).title("Marker in Sydney"));
            mMap.moveCamera(CameraUpdateFactory.newLatLng(sydney));
            Toast.makeText(MapsActivity.this, lon+" "+lat, Toast.LENGTH_LONG).show();
        }
    });
    button2.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            c=edit3.getText().toString();
            d=edit4.getText().toString();
            lon1=Double.parseDouble(c);
            lat1=Double.parseDouble(d);
            LatLng sydney = new LatLng(lon1,lat1);
        }
    });
}

```

```

        loc1 = new LatLng(lon, lat);
        loc2 = new LatLng(lon1, lat1);
        mMap.addMarker(new MarkerOptions().position(sydney).title("Marker in Sydney"));
        mMap.moveCamera(CameraUpdateFactory.newLatLng(sydney));
        distance = SphericalUtil.computeDistanceBetween(loc1, loc2);
        Toast.makeText(MapsActivity.this, "Distance is \n " + String.format("%.2f", distance / 1000) + "km", Toast.LENGTH_SHORT).show();
    }

    });

}

@Override
public void onMapReady(GoogleMap googleMap) {
    mMap = googleMap;

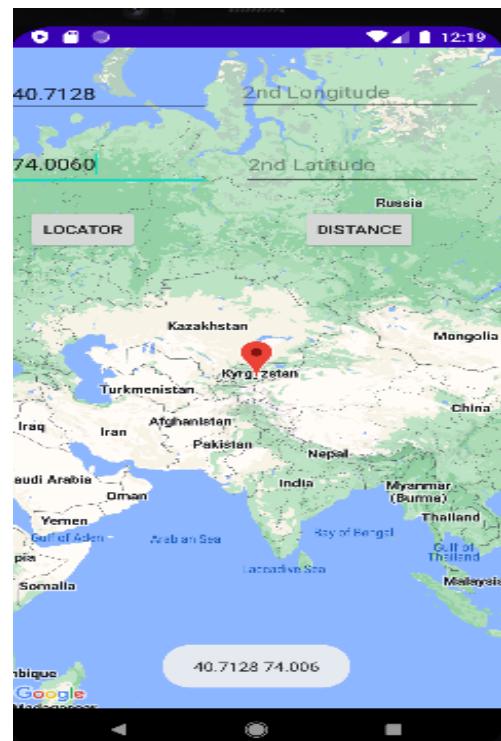
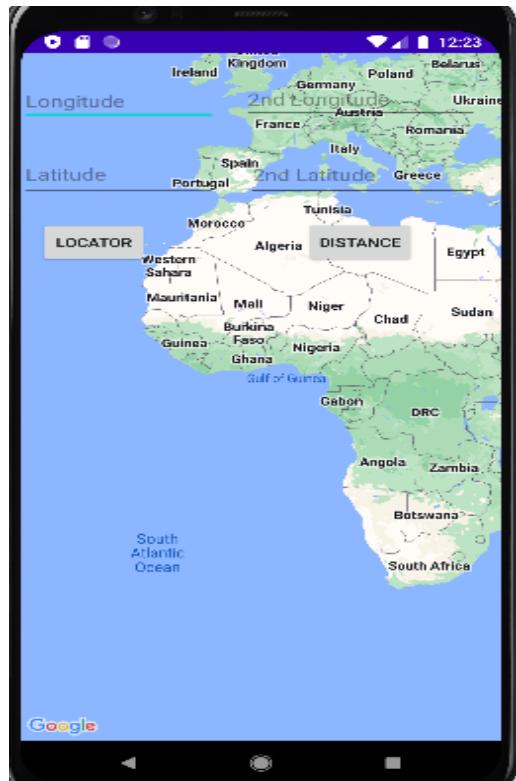
    // Add a marker in Sydney and move the camera
    //LatLang sydney = new LatLang(lon,lat);
    //mMap.addMarker(new MarkerOptions().position(sydney).title("Marker in Sydney"));
    //mMap.moveCamera(CameraUpdateFactory.newLatLang(sydney));

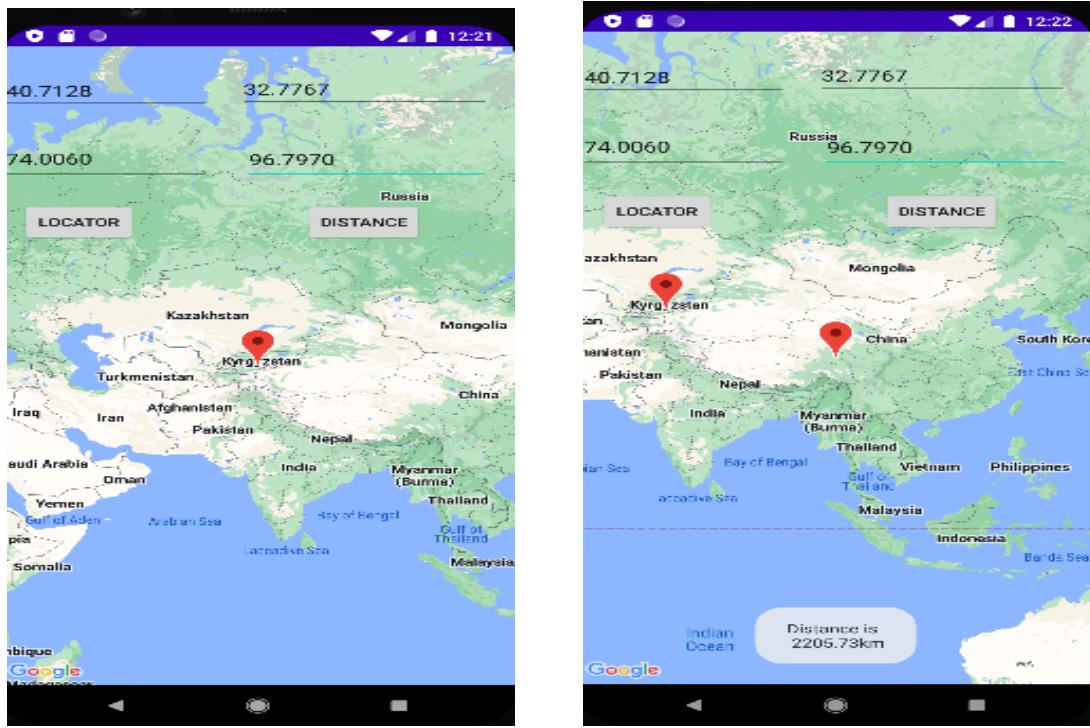
}

}

```

OUTPUT: -





CONCLUSION: -

From this practical I have learned to implement Android program to work with google maps and location

PRACTICAL 9

AIM: - Android program to work with images and videos

THEORY: -

Audio Player:

Android has a built-in microphone through which you can capture audio and store it , or play it on 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.setOutputFormat(MediaRecorder.OutputFormat.THREE_GPP);  
myAudioRecorder.setAudioEncoder(MediaRecorder.OutputFormat.AMR_NB);  
myAudioRecorder.setOutputFile(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.

Important Methods of MediaRecorder Class

Method	Description
set AudioSource()	This method will specify the source of the audio to be recorded.
set AudioEncoder()	This method is used to specify the audio encoder.
set OutputFormat()	This method is used to specify the output format of our audio.

<code>setOutputFile()</code>	This method is used to specify the path of recorded audio files that are to be stored.
<code>stop()</code>	This method is used to stop the recording process.
<code>start()</code>	This method is used to start the recording process.
<code>release()</code>	This method is used to release the resource that is associated with the Media recorder class.

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:

Methods

public void setMediaController(MediaController controller)

sets the media controller to the video view.

public void setVideoURI (Uri uri)

sets the URI of the video file.

public void start()

starts the video view.

public void stopPlayback()

stops the playback.

public void pause()

pauses the playback.

public void suspend()

suspends the playback.

public void resume()

resumes the playback.

public void seekTo(int millis)

seeks to specified time in milliseconds

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

Note: -

- 1) Permission need to be set in Androidmanifest.xml
- 2) You need to make a raw folder and insert the music and video file in it
- 3) Location functionality check properly by installing in your device
- 4) Right click on res->Android Resource Directory→Choose raw→Create it→You can copy the video and images from any folder in your PC and paste it here
- 5) Name of the video file should be in small letters
- 6) Name of the file should only be given not the entire path.

CODE: -

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.example.prac9">
```

```

<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:dataExtractionRules="@xml/data_extraction_rules"
    android:fullBackupContent="@xml/backup_rules"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/Theme.Prac9"
    tools:targetApi="31">
    <activity
        android:name=".MainActivity"
        android:exported="true">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
</application>
</manifest>

```

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"
    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"
    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 "
        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"
        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.prac9;
import static android.Manifest.permission.RECORD_AUDIO;
import static android.Manifest.permission.WRITE_EXTERNAL_STORAGE;
import android.Manifest;
import android.app.AlertDialog;
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.net.Uri;
import android.os.Build;
import android.os.Bundle;
import android.os.Environment;
import android.provider.Settings;
import android.util.Log;
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 androidx.annotation.NonNull;
import androidx.annotation.RequiresApi;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import java.io.File;
import java.io.IOException;
import java.util.Random;

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.parmanu);
        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();
            }
        }
    }
}

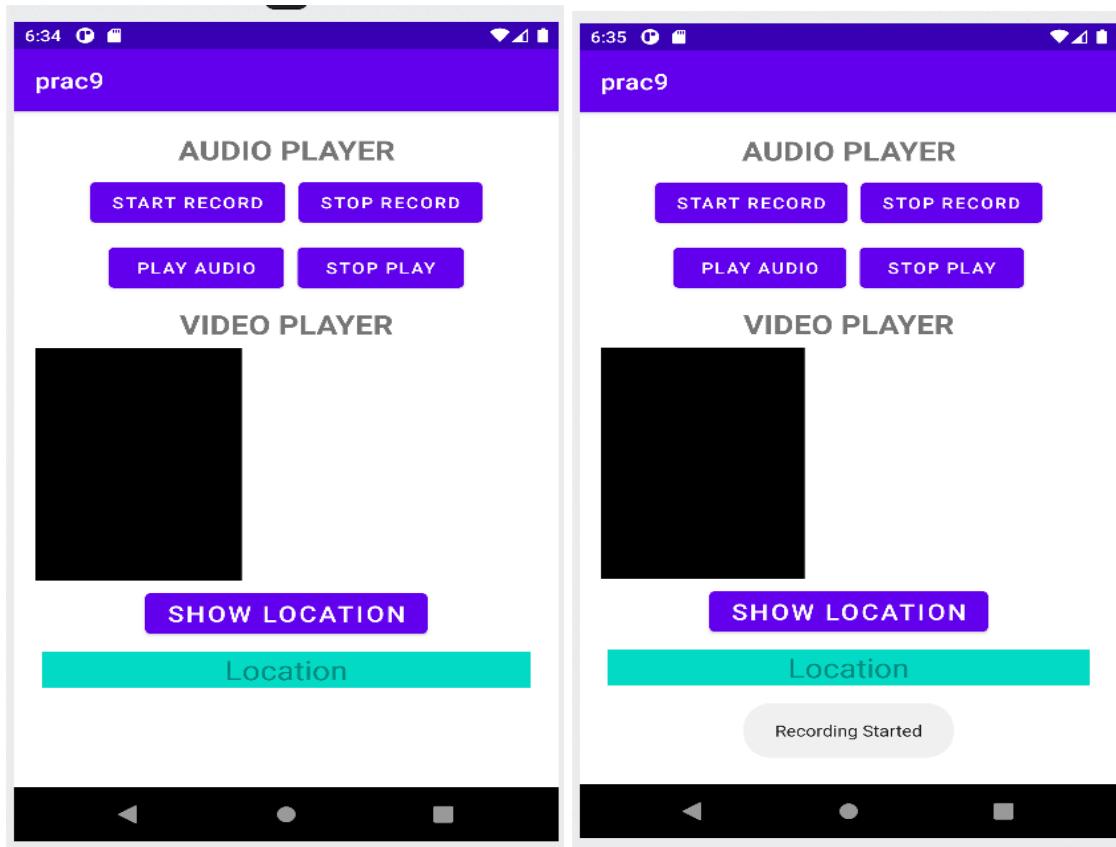
```

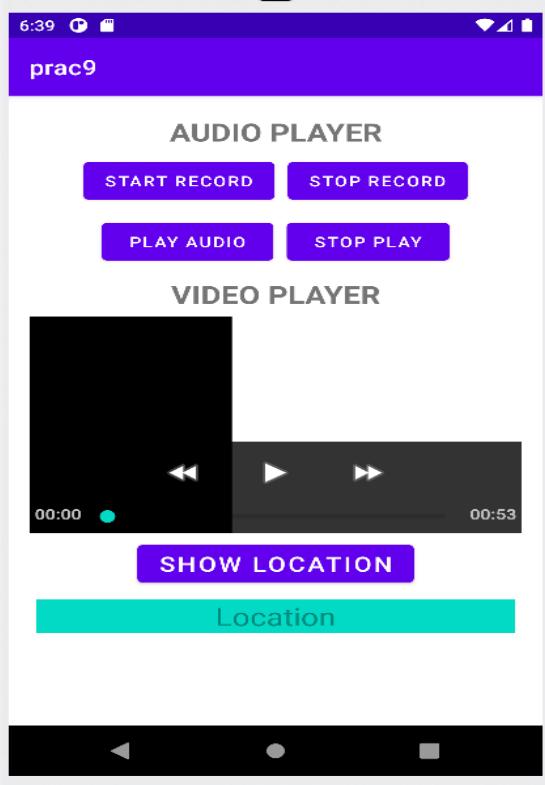
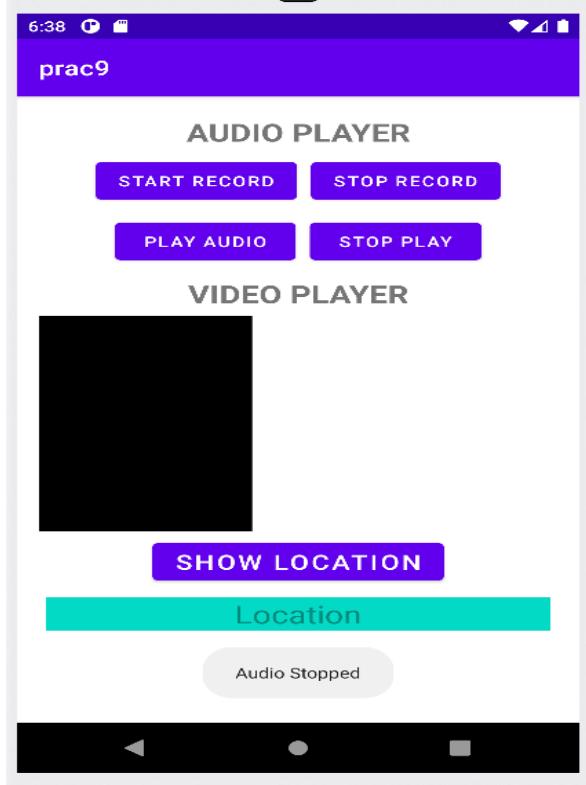
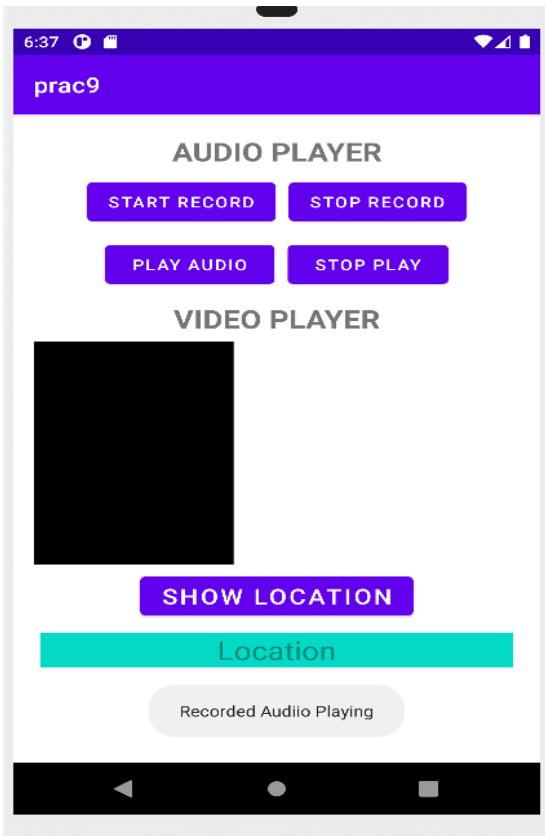
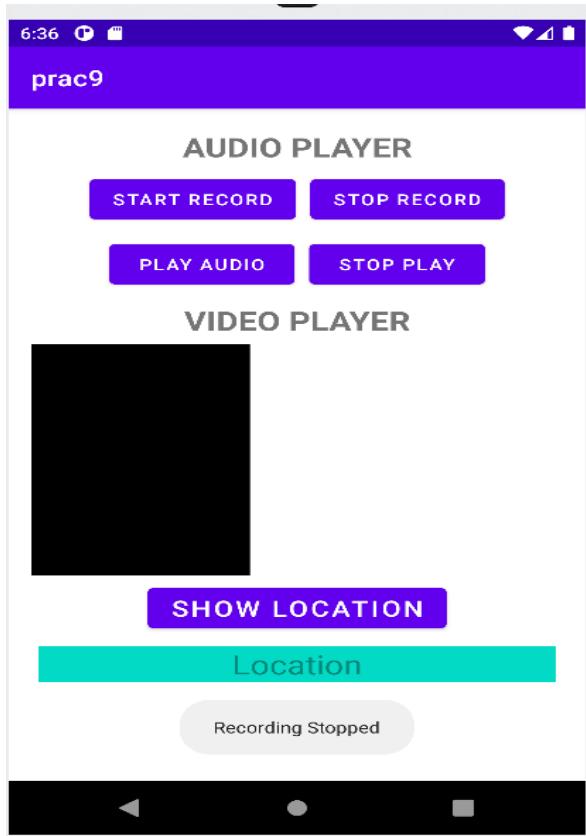
```

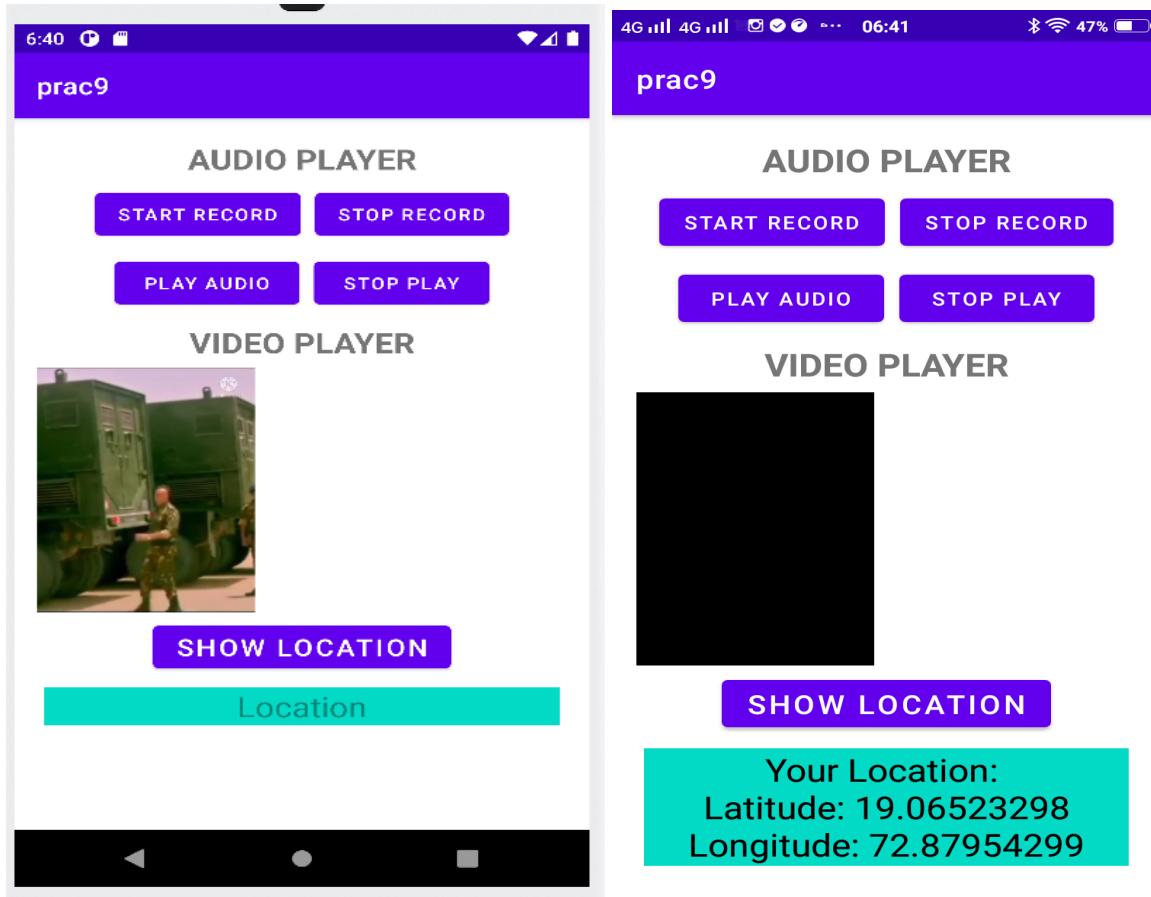
latitude = String.valueOf(lat);
longitude = String.valueOf(longi);
    showLocation.setText("Your Location: " + "\n" + "Latitude: " + latitude + "\n" +
"Longitude: " + longitude);
} else {
    Toast.makeText(this, "Unable to find location.",
    Toast.LENGTH_SHORT).show();
}
}
}
}
}

```

OUTPUT: -







CONCLUSION: -

From this practical I have learned to implement Android program to work with images and videos.

PRACTICAL 10

AIM: - Android program based on Rest API

- A) Create a basic application that allows you to download HTML from a given web page using HttpURLConnection.
- 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).
- C) Write a basic application to (use volley library), create a button and on click of the button a HTTP request will be sent to the server. The response from the server is then displayed using Toast on the screen.
- D) Write a program to create signup 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 the database. After getting a response from the api, display the message on the screen by using a Toast. (useRetrofit)

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.

NOTE: -

DataActivity.java

1) <https://designer.mocky.io/design>

For the volley program i.e. DataActivity.java you have to visit the above site

Enter the data as shown in below image

Then click on Generate my HTTP response

For the data you can visit below link and copy and modify as per your need

<https://run.mocky.io/v3/e563fe74-f4ab-47b9-ba6c-6fc147cd6579>

In the code which is shown in the below part between quotes you need to paste your URL

private String url =""

Status

200 - OK

The HTTP Code of the HTTP response you'll receive.

Response Content Type	REQUIRED	Charset	REQUIRED
application/json	x ▾	UTF-8	▾

The Content-Type header that will be sent with the response.

HTTP Headers

OPTIONAL

```
{
  "X-Foo-Bar": "Hello World"
}
```

Customize the HTTP headers sent in the response. Define the headers as a JSON object.

HTTP Response Body

OPTIONAL

```
{
  "users": [
    {
      "id": "42",
      "name": "CHRISTY PHILIP",
      "email": "christyphilip1234@gmail.com",
      "gender": "male",
      "contact": {
        "mobile": "+91 8454879108".
      }
    }
  ]
}
```

2) add dependency in build.gradle(app)

implementation 'com.android.volley:volley:1.2.1'

```
dependencies {

    implementation 'androidx.appcompat:appcompat:1.5.1'
    implementation 'com.google.android.material:material:1.7.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.1.4'
    testImplementation 'junit:junit:4.13.2'
    androidTestImplementation 'androidx.test.ext:junit:1.1.4'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.5.0'
    implementation 'com.android.volley:volley:1.2.1'
}
```

3) In AndroidManifest.xml

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

FormActivity.java

1) add dependency in build.gradle(app)

implementation 'com.squareup.retrofit2:retrofit:2.9.0'

implementation 'com.squareup.retrofit2:converter-gson:2.5.0'

```

dependencies {

    implementation 'androidx.appcompat:appcompat:1.5.1'
    implementation 'com.google.android.material:material:1.7.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.1.4'
    testImplementation 'junit:junit:4.13.2'
    androidTestImplementation 'androidx.test.ext:junit:1.1.4'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.5.0'
    implementation 'com.android.volley:volley:1.2.1'
    implementation 'com.squareup.retrofit2:retrofit:2.9.0'
    implementation 'com.squareup.retrofit2:converter-gson:2.5.0'
}

```

2) In AndroidManifest.xml

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

3) Inside htdocs create a folder and keep PHP file in it

4) Controller.java: - Below path should be noted(May change as you make)

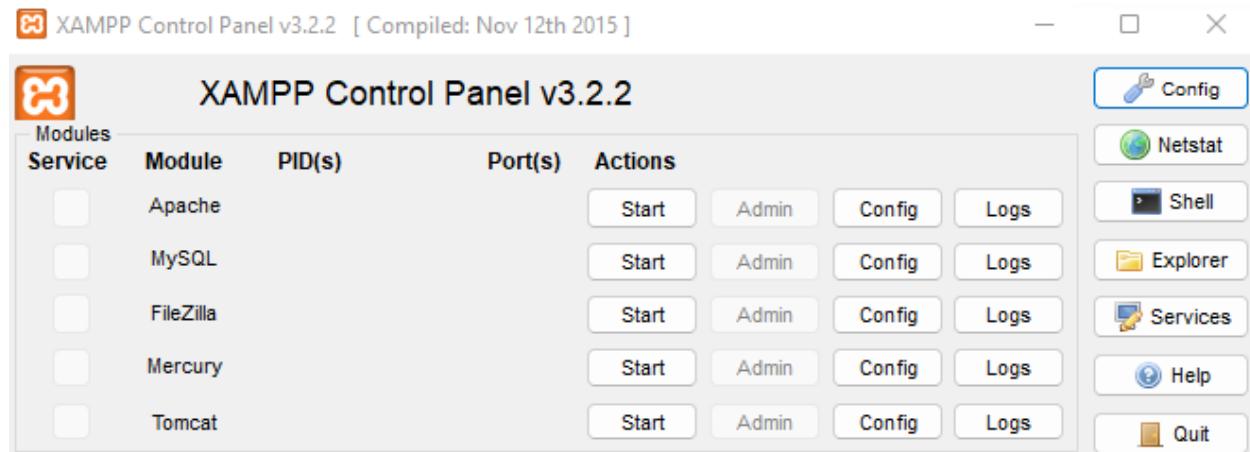
private static final String url="http://10.0.2.2:8080/PHP/Signup.php";

5) Apiset.java: -Below part path of php file(May change as you make)

@POST("/PHP/Signup.php")

Xampp

1) Goto Xampp start the Apache and MySQL



2) In the Browser enter the localhost:portnumber, you will be directed to below image click on circled part

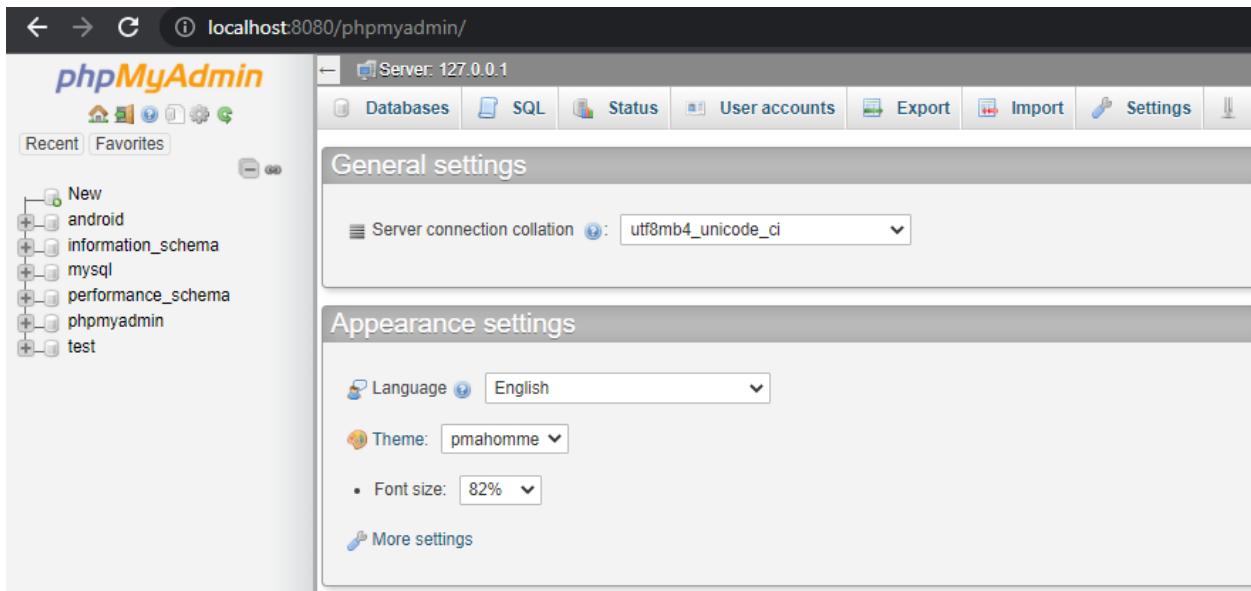


Welcome to XAMPP for Windows 7.0.24

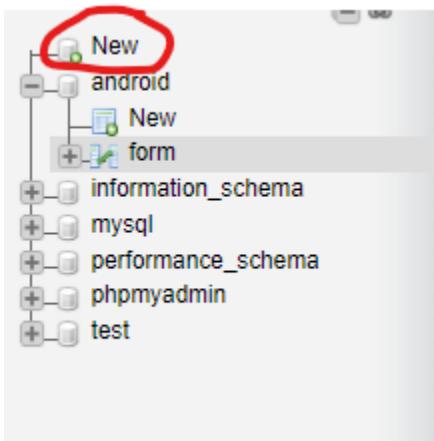
You have successfully installed XAMPP on this system! Now you can start using Apache, MariaDB, PHP and other components. You can find more info in the FAQs section or check the HOW-TO Guides for getting started with PHP applications.

XAMPP is meant only for development purposes. It has certain configuration settings that make it easy to develop locally but that are insecure if you want to have your installation accessible to others. If you want have your XAMPP accessible from the internet, make sure you understand the implications and you checked the FAQs to learn how to protect your site. Alternatively you can use WAMP, MAMP or LAMP which are similar packages which are more suitable for production.

3) Below window appears



4) Click on New



5) Enter database name and click Create

Databases

[Create database](#)

A screenshot of the 'Databases' section in MySQL Workbench. It shows a 'Create database' button with a 'Create' icon, a text input field containing 'android', and a dropdown menu for 'Collation'. A 'Create' button is also visible.

6) Enter Name and enter number of the columns and click Go

[Create table](#)

Name: Number of columns:

A screenshot of the 'Create table' dialog in MySQL Workbench. It has fields for 'Name' and 'Number of columns' (set to 4), and a 'Go' button.

7) Enter details as shown below

The screenshot shows the phpMyAdmin interface at localhost:8080/phpmyadmin/server_databases.php?server=1. The left sidebar lists databases: New, android, information_schema, mysql, performance_schema, phpmyadmin, and test. The main area is titled "Table: form" under "Database: android". The table structure is defined as follows:

Name	Type	Length/Values	Default	Collation
name	VARCHAR	50	None	
email	VARCHAR	50	None	
password	VARCHAR	50	None	

Below the table structure, there are fields for "Table comments:", "Collation:", and "Storage:", both currently empty. A "PARTITION definition:" field is also present.

8) Click on create below image window is shown

The screenshot shows the "Structure" tab in phpMyAdmin. The table "form" has the following structure:

#	Name	Type	Collation	Attributes	Null	Default	Comment
1	name	varchar(50)	latin1_swedish_ci		No	None	
2	email	varchar(50)	latin1_swedish_ci		No	None	
3	password	varchar(50)	latin1_swedish_ci		No	None	

At the bottom, there are buttons for "Check all", "With selected:", "Browse", "Change", "Drop", "Print", "Propose table structure", "Track table", "Move columns", "Indexes", and "Add". An "Add" button is set to 1 column(s) after "password".

The screenshot shows the phpMyAdmin interface. At the top, there are tabs: Browse, Structure, SQL, Search, Insert, and Export. Below the tabs, a message box displays a green checkmark and the text: "MySQL returned an empty result set (i.e. zero rows). (Query took 0.0009 seconds)". Underneath this message, a SQL query is shown: "SELECT * FROM `form`". At the bottom of the interface, there is a row of buttons labeled: name, email, password.

For writing code of php

- Open Notepad and write the code
- Save it with .php extension inside a folder created by you in the htdocs folder
- Now goto Browser

Enter below link

http://localhost:your_port_number/your_filename.php



The above ip should be used in your code in Controller.java 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"
    tools:context=".MainActivity">
```

```
<Button
    android:id="@+id	btn1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
```

```
    android:text="HTML"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.228" />
```

```
<Button
    android:id="@+id	btn2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/json"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.387" />
```

```
<Button
    android:id="@+id	btn3"
    android:layout_width="119dp"
    android:layout_height="55dp"
    android:text="@string/data_fetch"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.576" />
```

```
<Button
    android:id="@+id	btn4"
    android:layout_width="100dp"
    android:layout_height="60dp"
    android:text="@string/form"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.517"
    app:layout_constraintStart_toStartOf="parent"
```

```
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.742" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

activity_html.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=".HtmlActivity">

    <EditText
        android:id="@+id/et1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="textPersonName"
        android:hint="Enter website link"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.497"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.058" />

    <Button
        android:id="@+id	btn5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="VIEW HTML"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.219" />
```

```
<TextView
    android:id="@+id/txt1"
    android:layout_width="375dp"
    android:layout_height="436dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.526"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.79" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

activity_json.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=".JsonActivity">
    <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>

activity_data.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=".DataActivity">

    <Button
        android:id="@+id/buttonRequest"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Fetch Response" />

</LinearLayout>

activity_form.html
<?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="Enter Name"
        android:inputType="textPersonName"
        android:minHeight="48dp" />

<EditText
    android:id="@+id/t2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Enter Username"
    android:inputType="textEmailAddress"
    android:minHeight="48dp" />

<EditText
    android:id="@+id/t3"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Enter Password"
    android:inputType="textPassword"
    android:minHeight="48dp" />

<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","","","android");
$name=trim($_POST['name']);
$email=trim($_POST['email']);
$pass=trim($_POST['password']);
if($conn)
{
$qry = "select * from form where email='".$email."'";
$result = mysqli_query($conn,$qry);
if(mysqli_num_rows($result)>0)
{
$response['message']="exist";
```

```

echo json_encode($response);
}
else
{
$qry = "insert into form(name,email,password)values('$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);

}
?>

```

MainActivity.java

```

package com.example.prac10;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class MainActivity extends AppCompatActivity {
    Button html,json,data, form;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

```

html = findViewById(R.id.btn1);
json=findViewById(R.id.btn2);
data=findViewById(R.id.btn3);
form=findViewById(R.id.btn4);
html.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(MainActivity.this, HtmlActivity.class);
        startActivity(intent);
    }
});
json.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(MainActivity.this,JsonActivity.class);
        startActivity(intent);
    }
});
data.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(MainActivity.this, DataActivity.class);
        startActivity(intent);
    }
});
form.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(MainActivity.this, FormActivity.class);
        startActivity(intent);
    }
});
}
}

```

HtmlActivity.java

```

package com.example.prac10;

import androidx.appcompat.app.AppCompatActivity;

```

```
import android.os.Bundle;
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 HtmlActivity 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_html);
        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.isHttpUrl(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.et1);
    }
    if(requestUrlButton == null) {
        requestUrlButton =
            (Button)findViewById(R.id.btn5);
    }
    if(responseTextView == null) {
        responseTextView =
            (TextView)findViewById(R.id.txt1);
    }
    {
        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();
}
}

```

JsonActivity.java

```

package com.example.prac10;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.widget.TextView;

import org.json.JSONException;
import org.json.JSONObject;

public class JsonActivity extends AppCompatActivity {

```

```

String JSON_STRING = "{\"employee\":{\"name\":\"Christy Philip\",\"salary\":40000}}";
String name, salary;
TextView employeeName, employeeSalary;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_json);
    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();
    }
}
}

```

DataActivity.java

```

package com.example.prac10;

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 DataActivity 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/e563fe74-f4ab-47b9-ba6c-6fc147cd6579";

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

FormActivity.java

```
package com.example.prac10;

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 FormActivity extends AppCompatActivity {
    EditText t1,t2,t3;
    Button SignUpBtn;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_form);
        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(),""+t+call, Toast.LENGTH_LONG).show();
    }
});}}}

```

ResponseModel.java

```

package com.example.prac10;

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;
}
}
```

Controller.java

```
package com.example.prac10;
import retrofit2.Retrofit;
import retrofit2.converter.gson.GsonConverterFactory;
public class Controller {
    private static final String url="http://10.0.2.2:8080/PHP/Signup.php/";
    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);
    }
}
```

Apiset.java

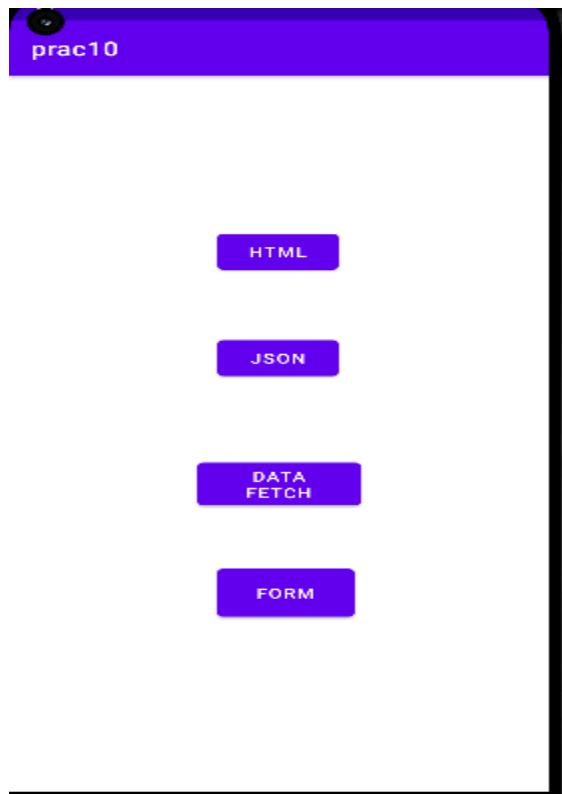
```
package com.example.prac10;

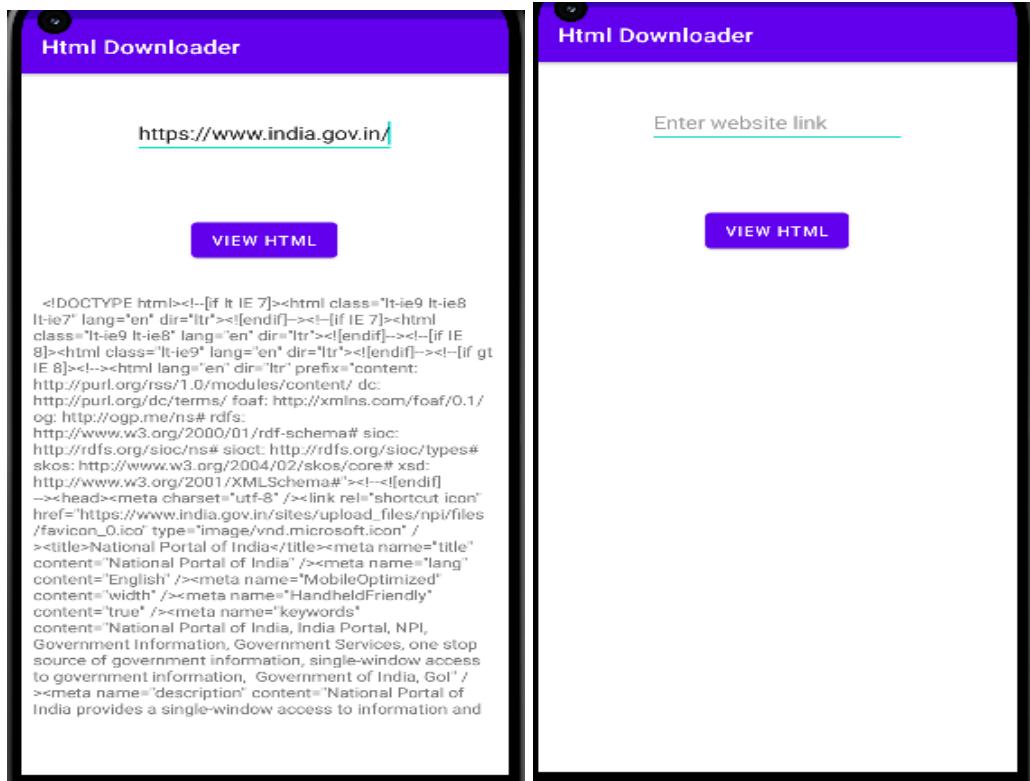
import retrofit2.Call;
import retrofit2.http.Field;
import retrofit2.http.FormUrlEncoded;
import retrofit2.http.POST;
```

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

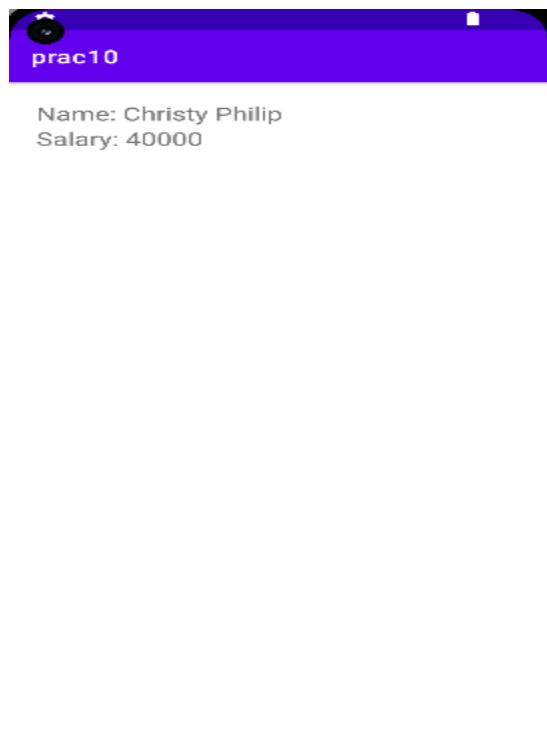
OUTPUT: -

Html

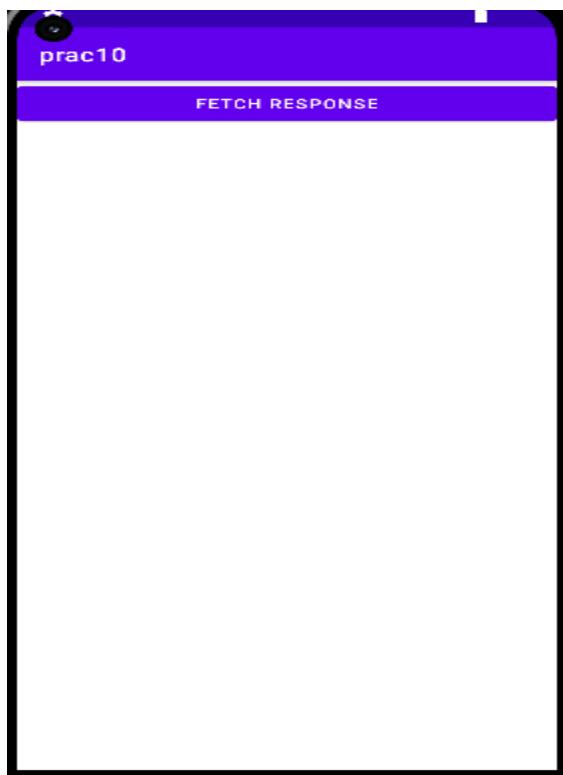




Json

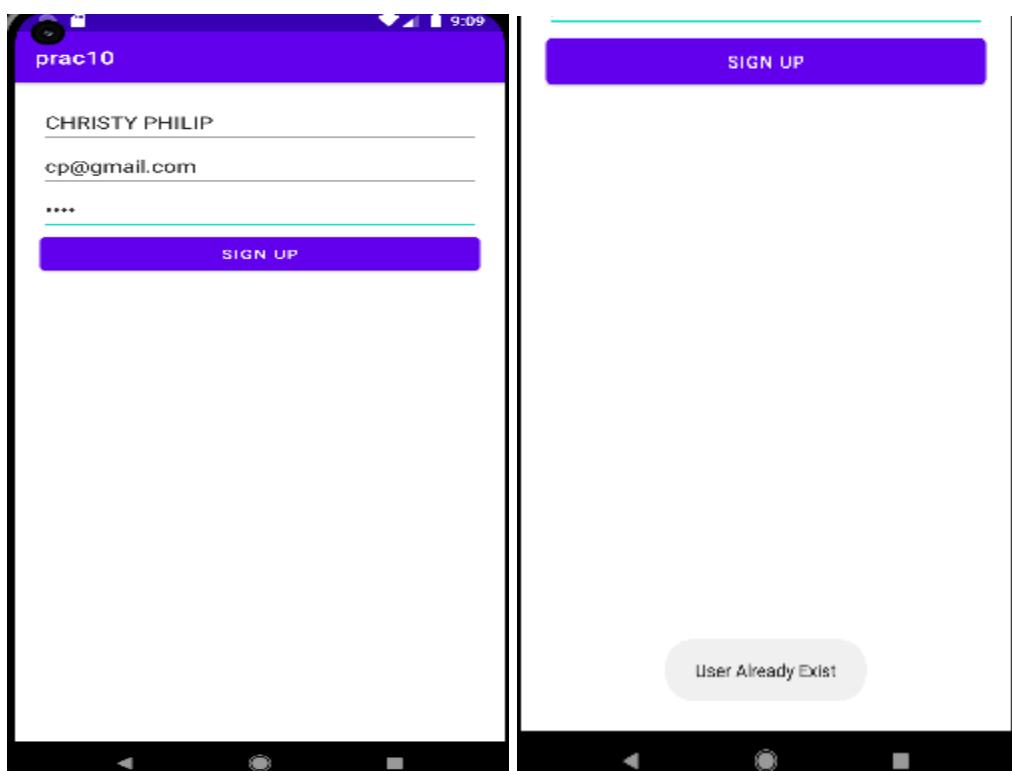
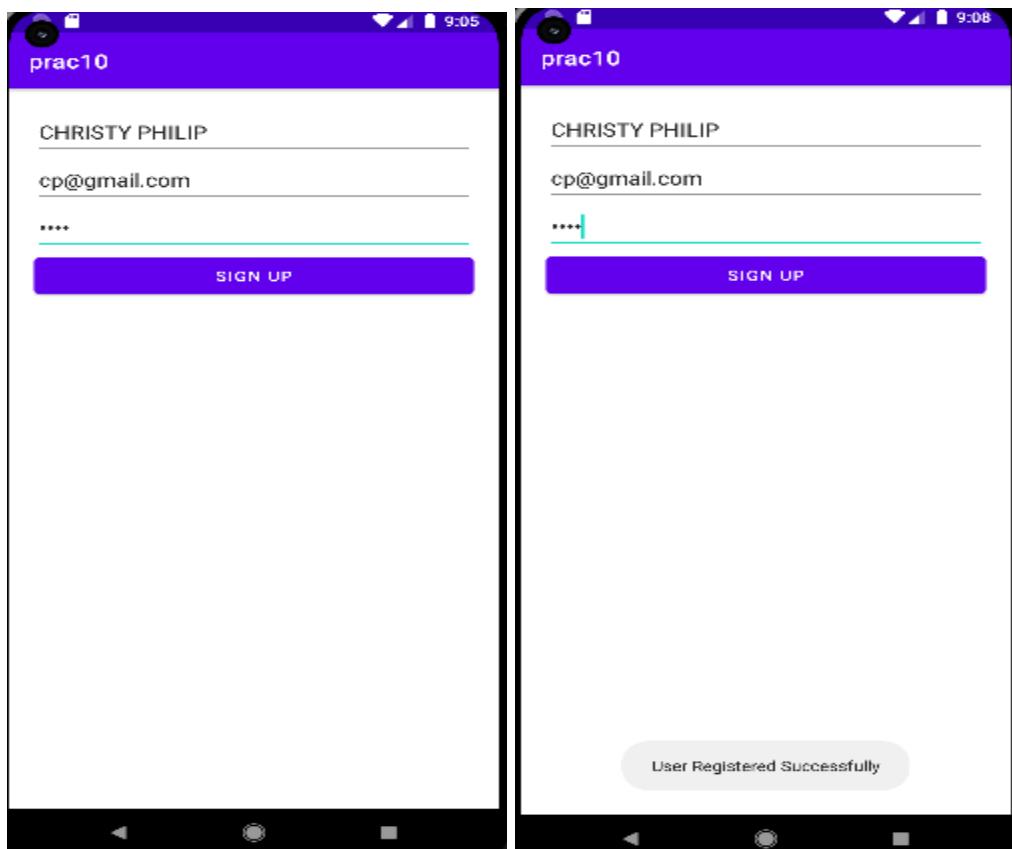


Volley



```
practise Response {  
    "users": [  
        {  
            "id": "42", "name": "CHRISTY PHILIP",  
            "email": "christyphilip1234@gmail.com",  
            "gender": "male",  
            "contact": {  
                "mobile": "+91 8454879108",  
                "office": "00 000000"  
            }  
        },  
        {  
            "id": "62",  
            "name": "Chinmay Vyapari",  
            "email": "chinmayvyapari@gmail.com",  
            "gender": "male",  
            "contact": {  
                "mobile": "+91 8108856863",  
                "office": "00 000000"  
            }  
        },  
        {  
            "id": "63",  
            "name": "Devendra Yadav",  
            "email": "devendra@gmail.com",  
            "gender": "male",  
            "contact": {  
                "mobile": "+91 9685423651",  
                "office": "00 356975"  
            }  
        },  
        {  
            "id": "52",  
            "name": "Yash Sawant",  
            "email": "yash@gmail.com",  
            "gender": "male",  
            "contact": {  
                "mobile": "+91 8454879108",  
                "office": "00 000000"  
            }  
        }  
    ]  
}
```

Retrofit



Result on localhost:8080

⚠ Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are disabled.

Showing rows 0 - 0 (1 total, Query took 0.0011 seconds.)

```
SELECT * FROM `form` WHERE 1
```

Show all | Number of rows: 25 Filter rows: Search this table

+ Options

name	email	password
CHRISTY PHILIP	cp@gmail.com	abcd

CONCLUSION: -

From this practical I have learned to implement Android program based on Rest API

PRACTICAL 11

AIM: - Flutter program using layout, widget and state management. 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.

THEORY: -

A) LAYOUTS

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. For example, we can see the below image that shows three icons with a label under each one.

Types of Layout Widgets

We can categories the layout widget into two types:

- Single Child Widget
- Multiple Child Widget

Single Child Widgets

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. The list of different types of single child widgets are:

Multiple Child Widgets

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.

The state of an app can very simply be defined as anything that exists in the memory of the app while the app is running. This includes all the widgets that maintain the UI of the app including the buttons, text fonts, icons, animations, etc. So now as we know what these states are, let's dive directly into our main topic i.e what are these stateful and stateless widgets and how do they differ from one another.

B) STATE MANAGEMENT & WIDGETS

State: The State is the information that can be read synchronously when the widget is built and might change during the lifetime of the widget.

In other words, the state of the widget is the data of the objects that its properties (parameters) are sustaining at the time of its creation (when the widget is painted on the screen). The state can also change when it is used for example when a CheckBox widget is clicked a check appears on the box.

Stateless Widget: The widgets whose state can not be altered once they are built are called stateless widgets. These widgets are immutable once they are built i.e any amount of change in the variables, icons, buttons, or retrieving data can not change the state of the app. Below is the basic structure of a stateless widget. Stateless widget overrides the build() method and returns a widget. For example, we use Text or the Icon is our flutter application where the state of the widget does not change in the runtime. It is used when the UI depends on the information within the object itself. Other examples can be Text, RaisedButton, IconButton.

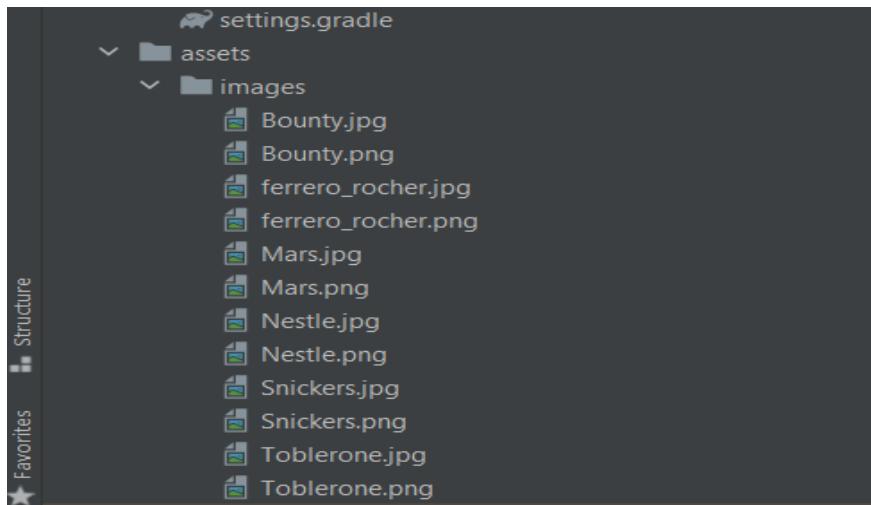
Stateful Widgets: The widgets whose state can be altered once they are built are called stateful Widgets. These states are mutable and can be changed multiple times in their lifetime. This simply means the state of an app can change multiple times with different sets of variables, inputs, data. Below is the basic structure of a stateful widget. Stateful widget overrides the createState() and returns a State. It is used when the UI can change dynamically. Some examples can be CheckBox, RadioButton, Form, TextField.

NOTE: -

The below image contains file structure for keeping images

The assets directory should be created inside your project and inside that images and then keep images

Project→assets→images→keep all images



CODE: -

pubspec.yaml

```
assets:
  - assets/images/Bounty.png
  - assets/images/ferrero_rocher.png
  - assets/images/Mars.png
  - assets/images/Nestle.png
  - assets/images/Snickers.png
  - assets/images/Toblerone.png
```

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,
```

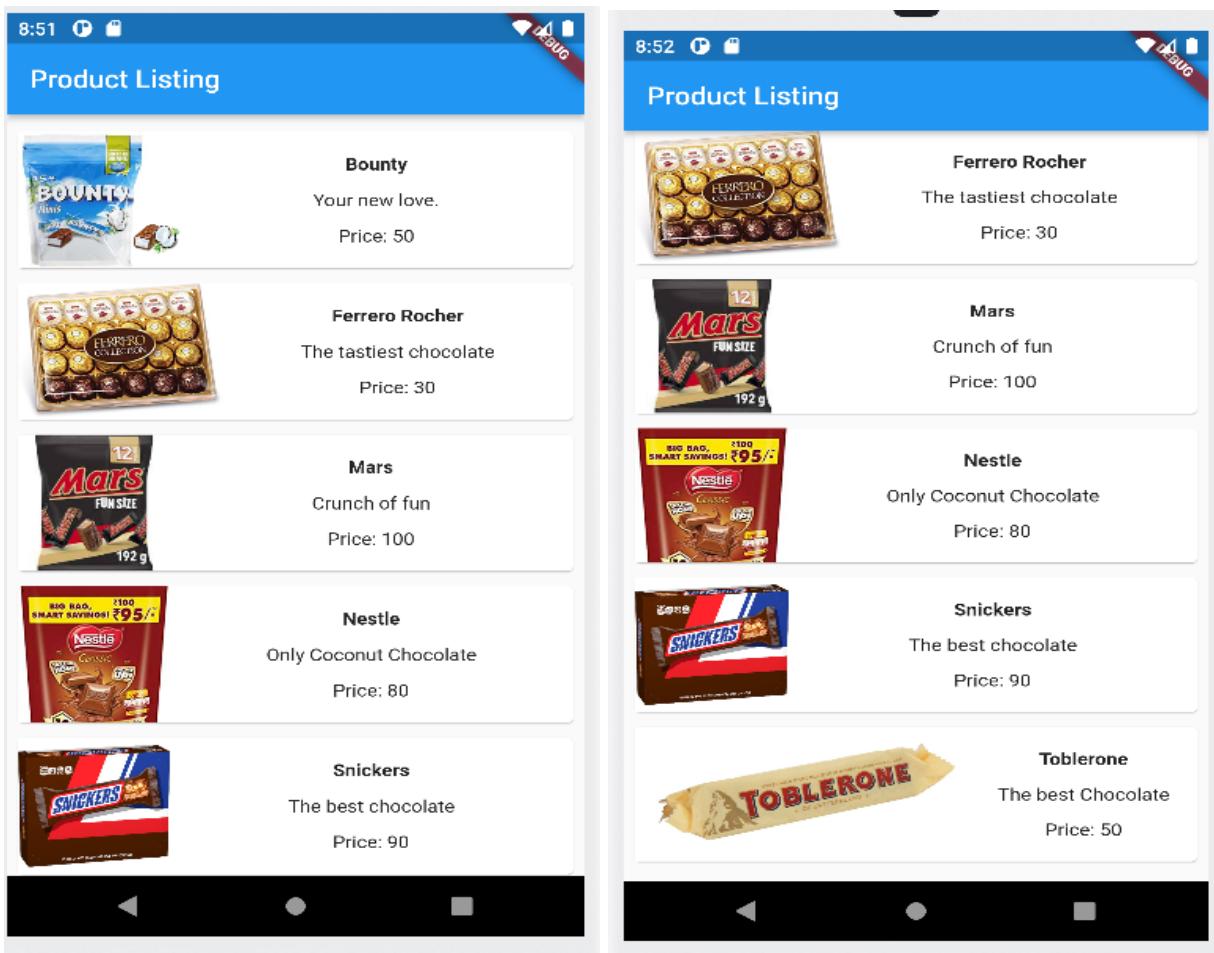
```
        ),  
        home: MyHomePage(title: 'Product layout demo home page'),  
    );  
}  
class MyHomePage extends StatelessWidget {  
    MyHomePage({Key? key, required this.title}) : super(key: key);  
    final String title;  
    @override  
    Widget build(BuildContext context){  
        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: "Bounty",  
                        description: "Your new love.",  
                        price: 50,  
                        image: "images/Bounty.png"  
                    ),  
                    ProductBox(  
                        name: "Ferrero Rocher",  
                        description: "The tastiest chocolate",  
                        price: 30,  
                        image: "images/ferrero_rocher.png"  
                    ),  
                    ProductBox(  
                        name: "Mars",  
                        description: "Crunch of fun",  
                        price: 100,  
                        image: "images/Mars.png"  
                    ),  
                    ProductBox(  
                        name: "Nestle",  
                        description: "Only Coconut Chocolate",  
                        price: 80,  
                        image: "images/Nestle.png"  
                    ),  
                    ProductBox(  
                ]  
            )  
        );  
    }  
}
```

```
        name: "Snickers",
        description: "The best chocolate",
        price: 90,
        image: "images/Snickers.png"
    ),
    ProductBox(
        name: "Toblerone",
        description: "The best Chocolate",
        price: 50,
        image: "images/Toblerone.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()),
    ],
)
)
]
)
);
}}
```

OUTPUT:-



CONCLUSION:-

From this practical I have learned to implement Flutter program using layout, widget and state management.

PRACTICAL 12

AIM: - Flutter program to work with SQLite Database. Create a student table with the following fields: Name, Roll No, Class and Contact No. Perform Insert, display, update and delete operations using flutter .

THEORY: -

What Is SQLite

SQLite is an open source relational database, it is used to create a database, perform different operations like add, delete, and remove data. SQLite does not require a server or backend code, all the data is saved to a text file in the device.

SQLite in Flutter

If you are seeking a way to use mobile device storage as a database in your Flutter application, then you can obtain it by just integrating the sqflite package.

Why SQLite and Sqflite?

SQLite is an open source relational database, it is used to create a database, perform different operation like add, delete, and remove data and one of the most popular ways to store data locally. SQLite does not require a server or backend code. All the data is saved to a text file in the device. For this article, we will be using the package Sqflite to connect with SQLite. Sqflite is packages for connecting to SQLite databases in Flutter.

sqflite

Sqflite is a Flutter plugin for SQLite, a self-contained, high-reliability, embedded, SQL database engine. SQLite plugin for Flutter. Supports iOS, Android and MacOS. While it's not a first-party plugin, it's recommended in the official Flutter cookbook.

- Support transactions and batches.
- Automatic version management during open.
- Helpers for insert/query/update/delete queries.
- DB operation executed in a background thread on iOS and Android.

path_provider

A Flutter plugin for finding commonly used locations on the filesystem. Supports Android, iOS, Linux, macOS and Windows. Not all methods are supported on all platforms.

path vs path_provider

path is a package to manipulate paths: join them, convert to absolute, add prefixes, get path info and so on.

path_provider is dedicated to more specific OS operations, e.g. downloads, temporary files, application documents are stored in different places based on the OS (obviously, file structure in Android is different than in iOS, Windows, Linux, etc.).

CODE: -

pubspec.yaml: -

```
name: prac12
description: A new Flutter project.
publish_to: 'none' # Remove this line if you wish to publish to pub.dev
version: 1.0.0+1
```

```
environment:
```

```
  sdk: '>=2.18.4 <3.0.0'
```

```
dependencies:
```

```
  flutter:    # Required for every Flutter project
  sdk: flutter # Required for every Flutter project
  cupertino_icons: ^1.0.2 # Only required if you use Cupertino (iOS style) icons
  sqflite: ^2.0.0+3
  path: ^1.0.0
  http: ^0.12.0+2
  flutter_lints: ^2.0.0
```

```
dev_dependencies:
```

```
  flutter_test:
    sdk: flutter # Required for a Flutter project that includes tests
```

```
flutter:
```

```
  uses-material-design: true
```

student.dart: -

```
import 'package:flutter/material.dart';
import 'dbhelper.dart';
class Student {
```

```

int? rno;
String? name;
String? section;
String? contactno;
Student(this.rno, this.name, this.section, this.contactno);
Student.fromMap(Map<String, dynamic> map) {
  rno = map['rno'];
  name = map['name'];
  section = map['section'];
  contactno = map['contactno'];
}
Map<String, dynamic> toMap() {
  return {
    DatabaseHelper.columnRno: rno,
    DatabaseHelper.columnName: name,
    DatabaseHelper.columnSection: section,
    DatabaseHelper.columnContactno: contactno,
  };
}
}
}

```

dbhelper.dart

```

import 'package:flutter/material.dart';
import 'package:path/path.dart';
import 'package:sqflite/sqflite.dart';
import 'student.dart';
class DatabaseHelper {
  static final _databaseName = "studentdb.db";
  static final _databaseVersion = 1;
  static final table = 'student_table';
  static final columnRno = 'rno';
  static final columnName = 'name';
  static final columnSection = 'section';
  static final columnContactno = 'contactno';
  // 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 (
            $columnRno INTEGER PRIMARY KEY,
            $columnName TEXT NOT NULL,
            $columnSection TEXT NOT NULL,
            $columnContactno TEXT NOT NULL
        )
    );
}
// Helper methods
// Inserts a row in the database where each key in the Map is a column name
// and the value is the column value. The return value is the id of the
// inserted row.
Future<int> insert(Student student) async {
    Database? db = await instance.database;
    return await db!.insert(table, {'name': student.name, 'section': student.section, 'contactno': student.contactno});
}
// All of the rows are returned as a list of maps, where each map is
// a key-value list of columns.
Future<List<Map<String, dynamic>>> queryAllRows() async {
    Database? db = await instance.database;
    return await db!.query(table);
}
// Queries rows based on the argument received
Future<List<Map<String, dynamic>>> queryRows(name) async {
    Database? db = await instance.database;
    return await db!.query(table, where: "$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(Student student) async {
    Database? db = await instance.database;
    int rno = student.toMap()['rno'];
    return await db!.update(table, student.toMap(), where: '$columnRno = ?', whereArgs: [rno]);
}

// 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 rno) async {
    Database? db = await instance.database;
    return await db!.delete(table, where: '$columnRno = ?', whereArgs: [rno]);
}
}

```

main.dart

```

import 'package:flutter/material.dart';
import 'student.dart';
import 'dbhelper.dart';
void main() => runApp(MyApp());
class MyApp extends StatelessWidget {
    @override
    Widget build(BuildContext context) {
        return MaterialApp(
            title: 'Flutter Sqlita',
            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<Student> students = [];
    List<Student> studentByName = [];
    //controllers used in insert operation UI
    TextEditingController nameController = TextEditingController();
    TextEditingController sectionController = TextEditingController();
    TextEditingController contactnoController = TextEditingController();
    //controllers used in update operation UI
    TextEditingController rollnoUpdateController = TextEditingController();
    TextEditingController nameUpdateController = TextEditingController();
    TextEditingController sectionUpdateController = TextEditingController();
    TextEditingController contactnoUpdateController = TextEditingController();
    //controllers used in delete operation UI
    TextEditingController rollnoDeleteController = TextEditingController();
    //controllers used in query operation UI
    TextEditingController queryController = TextEditingController();
    final GlobalKey<ScaffoldState> _scaffoldKey = new GlobalKey<ScaffoldState>();
    void _showMessageInScaffold(String message){
        ScaffoldMessenger.of(context).showSnackBar( Snackbar(
            content: Text(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('STUDENT DATABASE'),
),
body: TabBarView(
    children: [
        Center(
            child: Column(
                children: <Widget>[
                    Container(
                        padding: EdgeInsets.all(20),
                        child: TextField(
                            controller: nameController,
                            decoration: InputDecoration(
                                border: OutlineInputBorder(),
                                labelText: 'Student Name',
                            ),
                        ),
                    ),
                ],
            ),
        ),
    ],
),
Container(
    padding: EdgeInsets.all(20),
    child: TextField(
        controller: sectionController,
```

```
decoration: InputDecoration(  
    border: OutlineInputBorder(),  
    labelText: 'Student Section',  
)  
(  
),  
(  
),  
(  
),  
Container(  
  
padding: EdgeInsets.all(20),  
child: TextField(  
    controller: contactnoController,  
    decoration: InputDecoration(  
        border: OutlineInputBorder(),  
        labelText: 'Student Contact',  
)  
,  
,  
,  
,  
ElevatedButton(  
  
child: Text('Insert Student Details'),  
onPressed: () {  
    String name = nameController.text;  
    String section = sectionController.text;  
    String contactno = contactnoController.text;  
    _insert(name, section, contactno);  
},  
,  
],  
,  
,  
Container(  
child: ListView.builder(  
padding: const EdgeInsets.all(8),  
itemCount: students.length + 1,  
itemBuilder: (BuildContext context, int index) {  
    if (index == students.length) {  
        return ElevatedButton(  
    }  
})
```

```
        child: Text('Refresh'),
        onPressed: () {
            setState(() {
                _queryAll();
            });
        },
    );
}

return Container(
    height: 60,
    child: Center(
        child: Text(
            'ROLLNO:${students[index].rno}'
            'NAME:${students[index].name}'
            'SECTION:${students[index].section}'
            'CONTACTNO:${students[index].contactno}',
            style: TextStyle(fontSize: 18),
        ),
        ),
    );
}

Center(
    child: Column(
        children: <Widget>[
            Container(
                padding: EdgeInsets.all(20),
                child: TextField(
                    controller: queryController,
                    decoration: InputDecoration(
                        border: OutlineInputBorder(),
                        labelText: 'NAME',
                    ),
                    onChanged: (text) {
                        if (text.length >= 2) {
                            setState(() {

```

```
_query(text);
});
} else {
setState();
studentByName.clear();
});
}
},
),
height: 100,
),
Container(
height: 300,
child: ListView.builder(
padding: const EdgeInsets.all(8),
itemCount: studentByName.length,
itemBuilder: (BuildContext context, int index) {
return Container(
height: 50,
margin: EdgeInsets.all(2),
child: Center(
child: Text(
'ROLLNO:${studentByName[index].rno} ,NAME:${studentByName[index].name}
,SECTION:${studentByName[index].section}
,CONTACTNO:${studentByName[index].contactno}',
style: TextStyle(fontSize: 18),
),
),
);
},
),
],
),
),
Center()
```

```
child: Column(  
    children: <Widget>[  
        Container(  
            padding: EdgeInsets.all(20),  
            child: TextField(  
                controller: rollnoUpdateController,  
                decoration: InputDecoration(  
                    border: OutlineInputBorder(),
```

labelText: 'Student Rollno',

),
),
),
Container(

```
padding: EdgeInsets.all(20),  
child: TextField(  
controller: nameUpdateController,  
decoration: InputDecoration(  
border: OutlineInputBorder(),  
labelText: 'Student Name',
```

),
),
),
Container(

```
padding: EdgeInsets.all(20),  
child: TextField(  
controller: sectionUpdateController,  
decoration: InputDecoration(  
border: OutlineInputBorder(),  
labelText: 'Student Section',
```

),
),
),

Container(

```
padding: EdgeInsets.all(20),
child:TextField(
controller: contactnoUpdateController,
decoration: InputDecoration(
border: OutlineInputBorder(),
labelText: 'Student Contact'

),
),
),
),
ElevatedButton(
child: Text('Insert Student Details'),
onPressed: () {
int rno = int.parse(rollnoUpdateController.text);
String name = nameUpdateController.text;
String section = sectionUpdateController.text;
String contactno = contactnoUpdateController.text;
_update(rno, name, section, contactno);
},
),
],
),
),
),
),
Center(
child: Column(
children: <Widget>[
Container(
padding: EdgeInsets.all(20),
child: TextField(
controller: rollnoDeleteController,
decoration: InputDecoration(
border: OutlineInputBorder(),
labelText: 'Student Rollno',

),
),
),
),
),
ElevatedButton(
```

```

        child: Text('Delete'),
        onPressed: () {
            int rno = int.parse(rollnoDeleteController.text);
            _delete(rno);
        },
    ),
],
),
),
),
],
),
),
),
);
}
void _insert(name, section, contact) async {
// row to insert
    Map<String, dynamic> row = {
        DatabaseHelper.columnName: name,
        DatabaseHelper.columnSection: section,
        DatabaseHelper.columnContactno: contact,
    };
    Student student = Student.fromMap(row);
    final rno = await dbHelper.insert(student);
    _showMessageInScaffold('inserted row id: $rno');
}
void _queryAll() async {
    final allRows = await dbHelper.queryAllRows();
    students.clear();
    allRows.forEach((row) => students.add(Student.fromMap(row)));
    _showMessageInScaffold('Query done.');
    setState(() {});
}
void _query(name) async {
    final allRows = await dbHelper.queryRows(name);
    studentByName.clear();
    allRows.forEach((row) => studentByName.add(Student.fromMap(row)));
}
void _update(rno, name, section, contact) async {
// row to update
    Student student = Student(rno, name, section, contact);

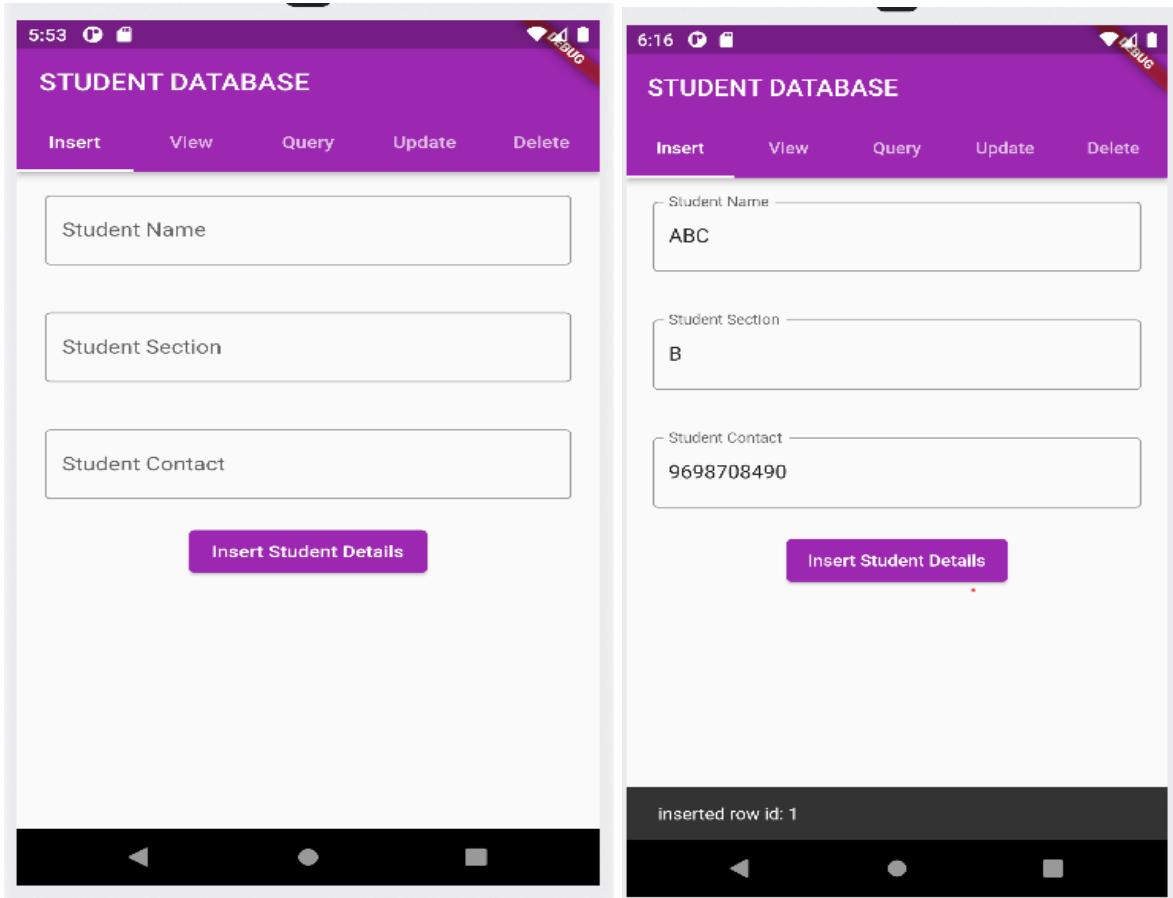
```

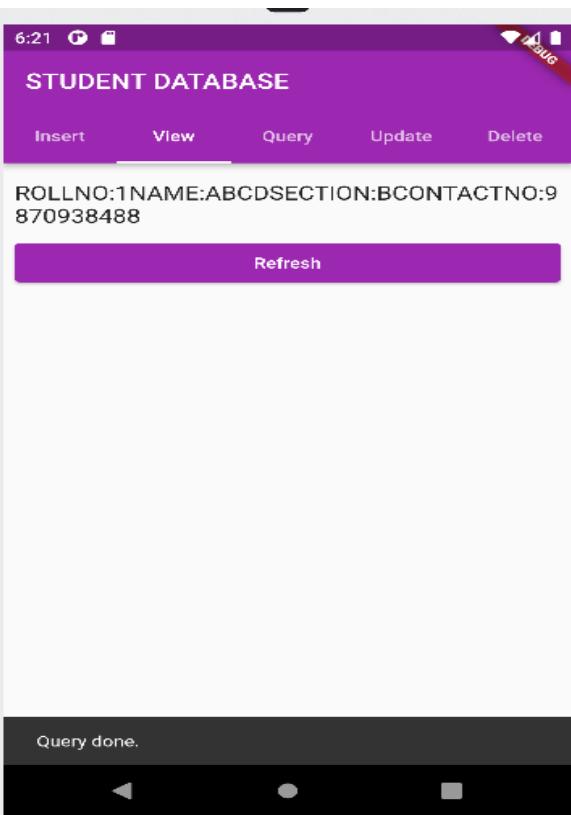
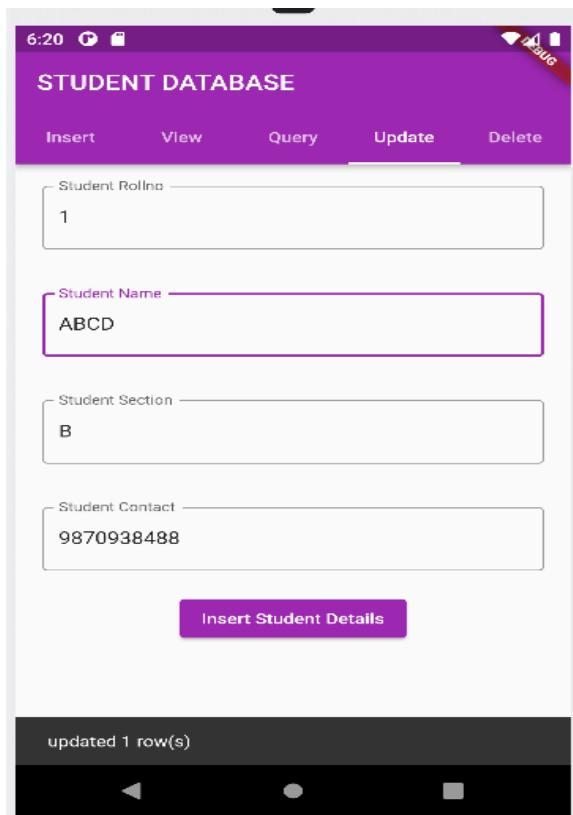
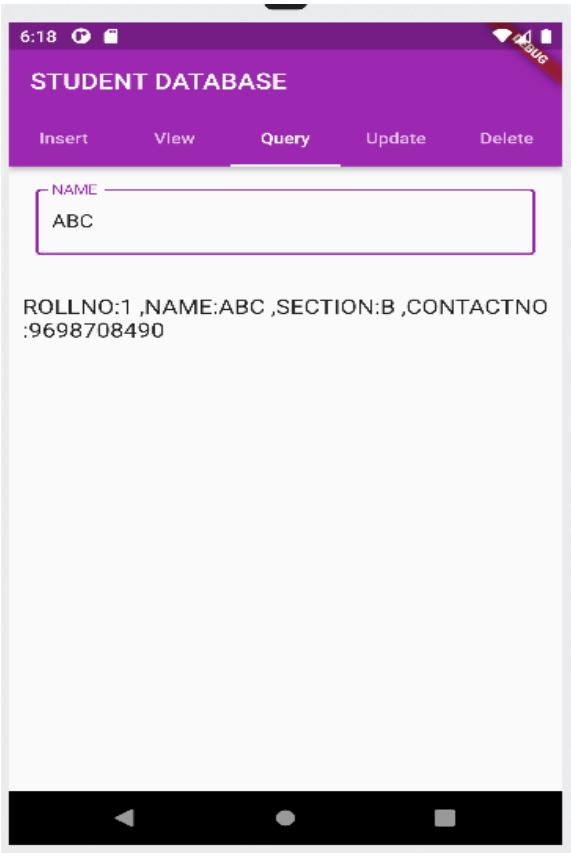
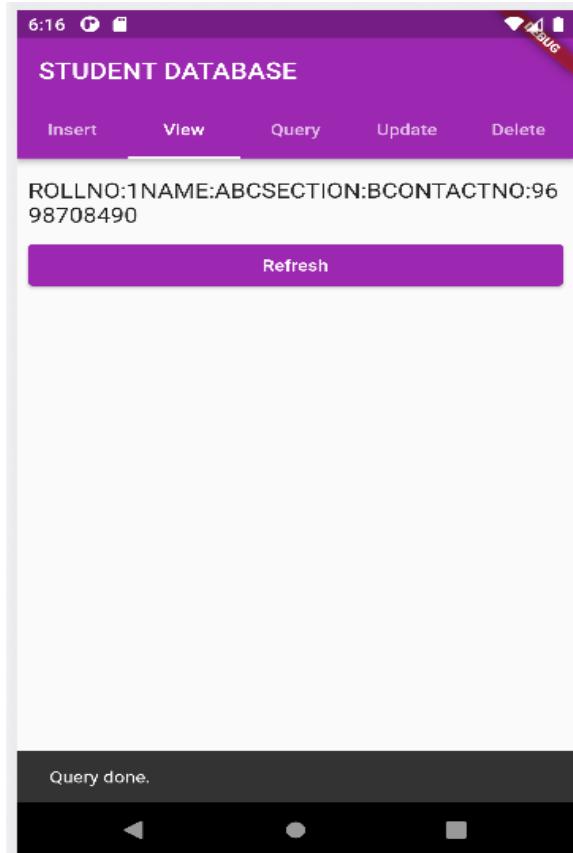
```

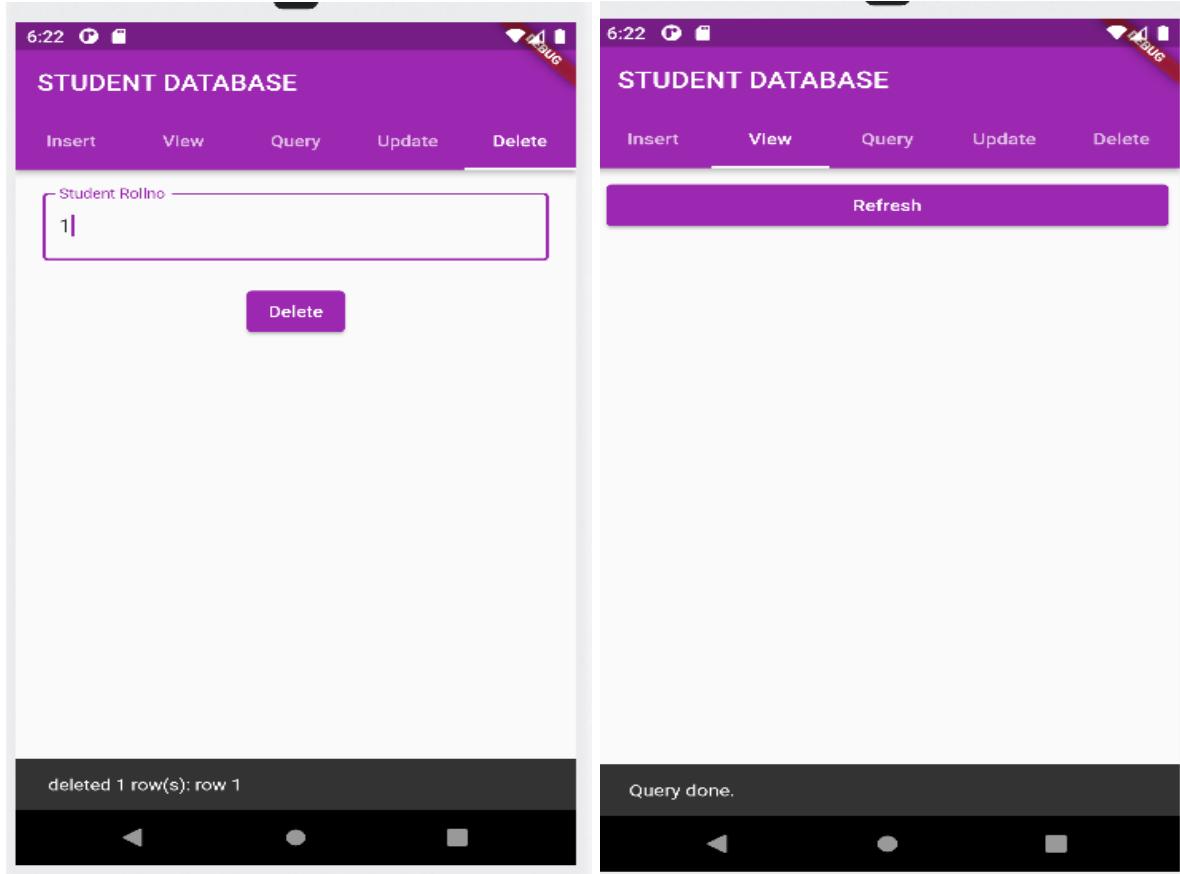
final rowsAffected = await dbHelper.update(student);
_showMessageInScaffold('updated $rowsAffected row(s)');
}
void _delete(rno) async {
// Assuming that the number of rows is the id for the last row.
final rowsDeleted = await dbHelper.delete(rno);
_showMessageInScaffold('deleted $rowsDeleted row(s): row $rno');
}
}

```

OUTPUT: -







CONCLUSION: -

From this practical I have learned to implement Flutter program to work with SQLite DatabaseCreate and perform Insert, display, update and delete operations using.

PRACTICAL 13

AIM: - Flutter program based on Rest API.

THEORY: -

Along with building a UI in Flutter, we can also integrate it with the backend. Most applications use API to display the user data. We will use the HTTP package, which provides advanced methods to perform operations. REST API uses simple http calls to communicate with JSON data because:

- It uses await & async features.
- It provides various methods.
- It provides class and http to perform web requests.

Install the http dependency and add it in pubspec.yaml file in order to use API in the application.

CODE: -

pubspec.yaml

name: prac13

description: A new Flutter project.

publish_to: 'none' # Remove this line if you wish to publish to pub.dev

version: 1.0.0+1

environment:

 sdk: '>=2.18.4 <3.0.0'

dependencies:

 flutter:

 sdk: flutter

 http: ^0.13.4

 cupertino_icons: ^1.0.2

```
dev_dependencies:
```

```
  flutter_test:
```

```
    sdk: flutter
```

```
flutter_lints: ^2.0.0
```

```
flutter:
```

```
  uses-material-design: true
```

main.dart

```
import 'package:flutter/material.dart';
import 'package:prac13/gender.dart';
import 'dart:async';
import 'dart:convert';
import 'package:http/http.dart' as http;
import 'gender.dart';
void main() {
  runApp(const MyApp());
}
class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Flutter Gender Prediction',
      theme: ThemeData(
        primarySwatch: Colors.purple,
      ),
      home: const MyHomePage(title: 'Gender Prediction'),
    );
}
class MyHomePage extends StatefulWidget {
  const MyHomePage({Key? key, required this.title}) : super(key: key);
  final String title;
  @override
  State<MyHomePage> createState() => _MyHomePageState();
```

```
}

class _MyHomePageState extends State<MyHomePage> {
    late Future<Gender> gender;
    TextEditingController nameController = TextEditingController();
    String genderop = "";
    String nameop = "";
    String probop = "";
    Future<Gender> getPred(String name) async {
        final response = await http.get(Uri.parse(
            "https://api.genderize.io/?name=" + name.toLowerCase().trim()));
        if (response.statusCode == 200) {
            print(jsonDecode(response.body));
            setState(() {
                genderop = "Gender: " +
                    jsonDecode(response.body)["gender"].toString().toUpperCase();
                nameop = "Name: " +
                    jsonDecode(response.body)["name"].toString().toUpperCase();
                probop = "Probability: " +
                    jsonDecode(response.body)["probability"].toString();
            });
            return Gender.fromJson(jsonDecode(response.body));
        } else {
            print("Failed to load data");
            throw Exception('Failed to load data');
        }
    }
    @override
    void initState() {
        super.initState();
    }
    String? get _errorText {
        final text = nameController.text;
        if (text.isEmpty) {
            return 'Can\'t be empty';
        }
        return null;
    }
    @override
    Widget build(BuildContext context) {
        return Scaffold(
```

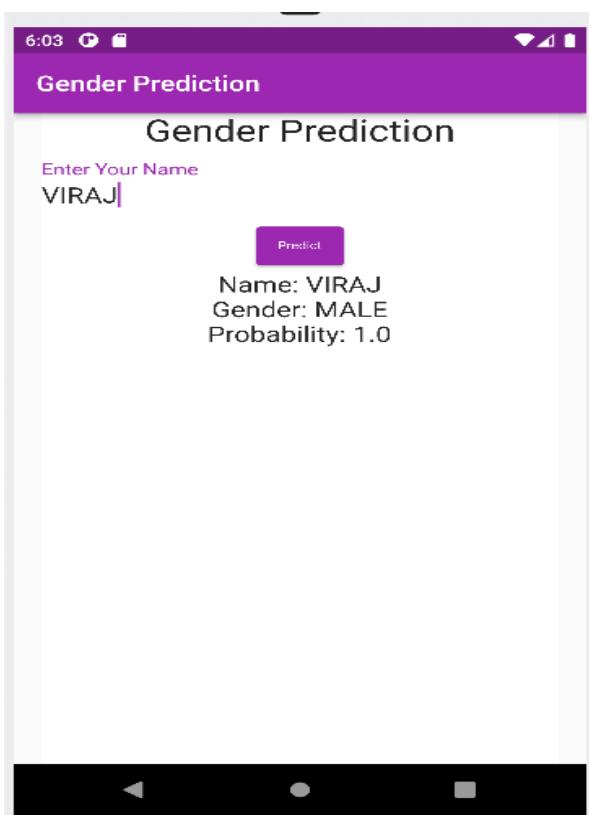
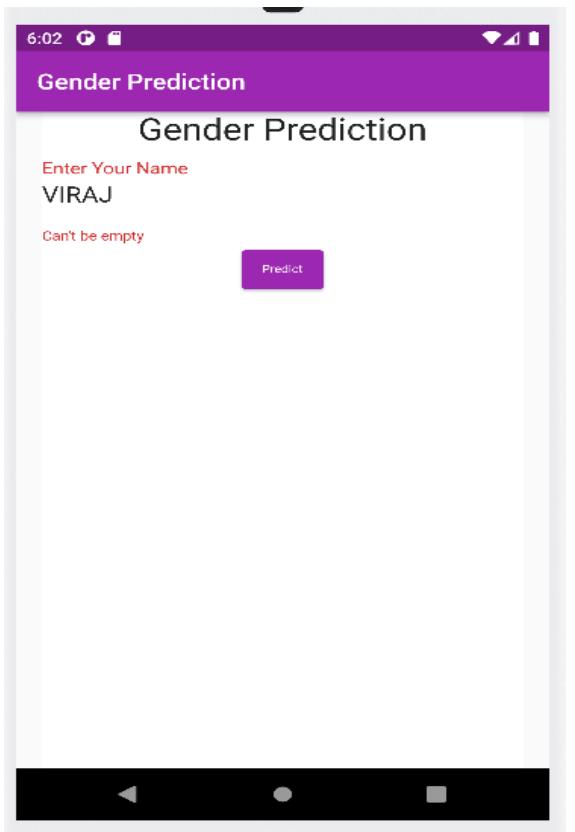
```
appBar: AppBar(  
    title: Text(widget.title),  
,  
body: Center(  
// Center is a layout widget. It takes a single child and positions it  
// in the middle of the parent.  
    child: Container(  
        alignment: Alignment.center,  
        color: Colors.white,  
        margin: const EdgeInsets.only(left: 20.0, right: 20.0),  
        child: Column(  
            children: <Widget>[  
                Text(  
                    'Gender Prediction',  
                    style: TextStyle(fontSize: 28.0),  
                ),  
                TextField(  
                    controller: nameController,  
                    style: TextStyle(fontSize: 20.0),  
                    decoration: InputDecoration(  
                        border: InputBorder.none,  
                        labelText: 'Enter Your Name',  
                        hintText: 'Enter Your Name',  
                        errorText: _errorText,  
                    ),  
                    ),  
                    ),  
                    ElevatedButton(  
                        child: Text(  
                            'Predict'),  
                        style: ElevatedButton.styleFrom(  
                            primary: Colors.purple,  
                            textStyle: const TextStyle(  
                                color: Colors.white,  
                                fontSize: 10,  
                                fontStyle: FontStyle.normal),  
                        ),  
                        onPressed: () {  
                            if (nameController.text != null) {  
                                gender = getPred(nameController.text);  
                            }  
                        }  
                    )  
                ]  
            )  
        )  
    )  
);
```

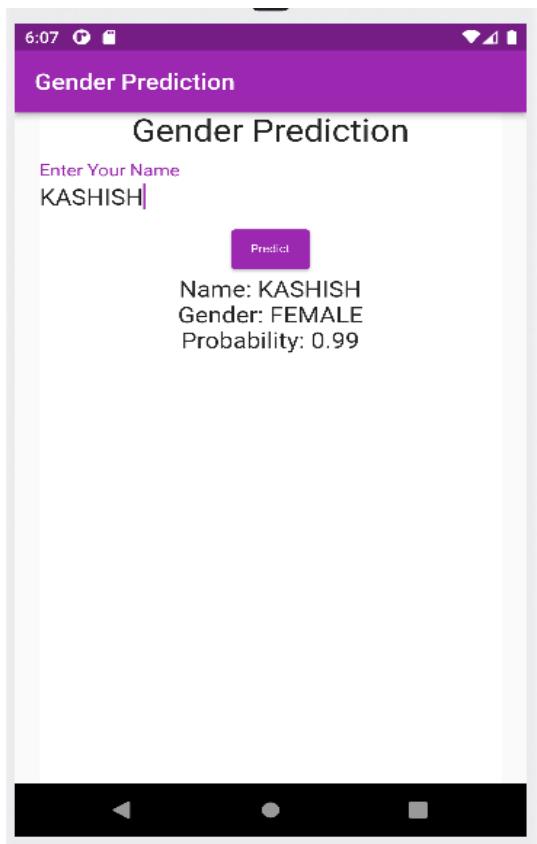
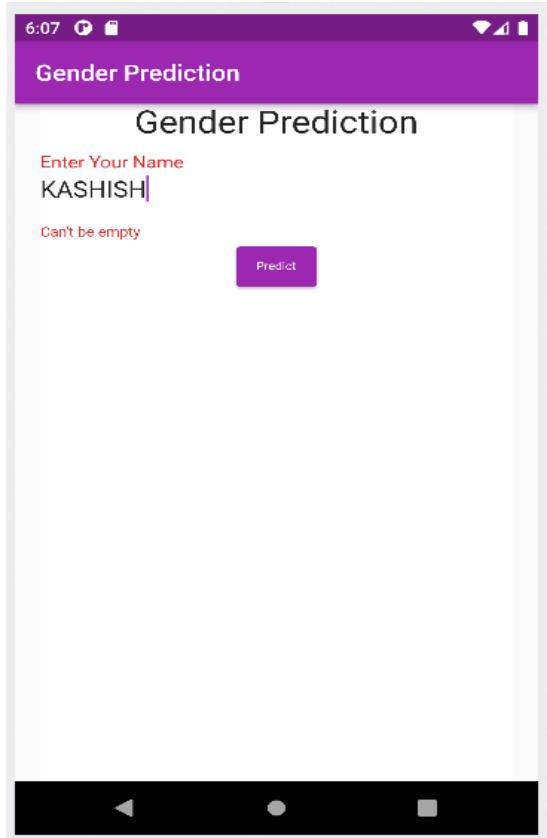
```
        },
        new Text(
            "$nameop",
            style: TextStyle(fontSize: 20.0),
        ),
        new Text(
            "$genderop",
            style: TextStyle(fontSize: 20.0),
        ),
        new Text(
            "$probop",
            style: TextStyle(fontSize: 20.0),
        ),
    ],
),
),
),
),
),
);
}
}
```

gender.dart

```
class Gender {
    final String? gender;
    final String? name;
    final double? probability;
    Gender(
        this.gender,
        this.name,
        this.probability,
    );
    factory Gender.fromMap(Map<String, dynamic> json) {
        return Gender(json['gender'], json['name'], json['probability']);
    }
    factory Gender.fromJson(Map<String, dynamic> json) {
        return Gender(json['gender'], json['name'], json['probability']);
    }
}
```

OUTPUT: -





CONCLUSION: -

From this practical I have learned to implement Flutter program based on Rest API.