INDEX

2	**Database Connectivity:** 1. Create an Android application to read and write content in internal storage. 2. Create an Android application to read and write content in external storage. 3. Write an android program for shared preference to store value in name-value pairs. 4. Create a login form with a remember me checkbox. Save the username and password if the checkbox is checked using shared preference and show the welcome page when the login button is clicked. 5. Create an Android application to insert, update, select, and delete records from the student table using SQLite Database. 6. Write a program to create a user registration form, after registration data will be inserted in the SQLite database, and design an activity that displays that information. 7. Android Program to perform CRUD operation using real time database Firebase.			**CO2**
3	**Animation, Multimedia and Location Based Services:** 1. Write an Android application to play, pause, and stop an audio file. 2. Write an Android application to play a video with Media controller. 3. Create an android application that applies different animations on an image. 4. Create an Android application to implement frame animation. 5. Create an Android application to display the current location of your device (display longitude and latitude values). 6. Create an Android application that displays the current location of your device from longitude and latitude values (Reverse Geocoding). 7. Create an Android application that accepts longitude and latitude from the user and marks that location on google map.			**CO3**

4	**REST API integration:** 1. Create an Android application to demonstrate JSON data parsing using OkHttp (you can use https://api.github.com/users JSON data). 2. Create an Android application to demonstrate JSON data parsing using Volley (you can use https://api.github.com/users JSON data).			**CO4**
	3. Create an Android application to demonstrate JSON data parsing using Retrofit (you can use https://api.github.com/users JSON data).			
5	**Introduction to Dart and Flutter:** 1. Write a Flutter program to demonstrate Text widget and its properties. 2. Write a Flutter program to display dog names (demonstrate stateless widget and column widgets). 3. Write a Flutter program that allows the user to enter a city in a text field and displays city name (demonstrate stateful widget). 4. Write a Flutter program to change the background color (demonstrate stateful widget). 5. Write a Flutter Program to display fruit list using ListView. 6. Write a Flutter program to demonstrate navigation (user should be navigated from first screen to second screen). 7. Write a Flutter program to design a Login form using TextField, Check Box, Buttons, Drop down, Switch etc.			**CO5**
6	**Data Handling in Flutter:** 1. Write a Flutter program based on RestAPI to fetch data. 2. Write a flutter program to demonstrate JSON serialization and Deserialization. 3. Write a flutter program to perform CRUD operations using sqflite.			**CO6**

Experiment No 1 - Introduction to Android Application Components and UI Controls

1. Write a program to demonstrate activity life cycle.

Mainactivity.java

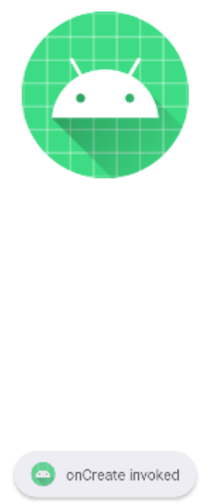
```
package com.example.activitylifecycle;
import android.os.Bundle;
import android.util.Log;
import androidx.appcompat.app.AppCompatActivity;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

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

      Log.d(TAG, "onCreate called");
      Toast.makeText(this, "onCreate", Toast.LENGTH_SHORT).show();
   }

   @Override
   protected void onStart() {
      super.onStart();
      Log.d(TAG, "onStart called");
      Toast.makeText(this, "onStart", Toast.LENGTH_SHORT).show();
   }

   @Override
   protected void onResume() {
      super.onResume();
      Log.d(TAG, "onResume called");
      Toast.makeText(this, "onResume", Toast.LENGTH_SHORT).show();
   }
```

```
    @Override
    protected void onPause() {
        super.onPause();
        Log.d(TAG, "onPause called");
        Toast.makeText(this, "onPause", Toast.LENGTH_SHORT).show();
    }

    @Override
    protected void onStop() {
        super.onStop();
        Log.d(TAG, "onStop called");
        Toast.makeText(this, "onStop", Toast.LENGTH_SHORT).show();
    }

    @Override
    protected void onRestart() {
        super.onRestart();
        Log.d(TAG, "onRestart called");
        Toast.makeText(this, "onRestart", Toast.LENGTH_SHORT).show();
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();
        Log.d(TAG, "onDestroy called");
        Toast.makeText(this, "onDestroy", Toast.LENGTH_SHORT).show();
    }
}
```

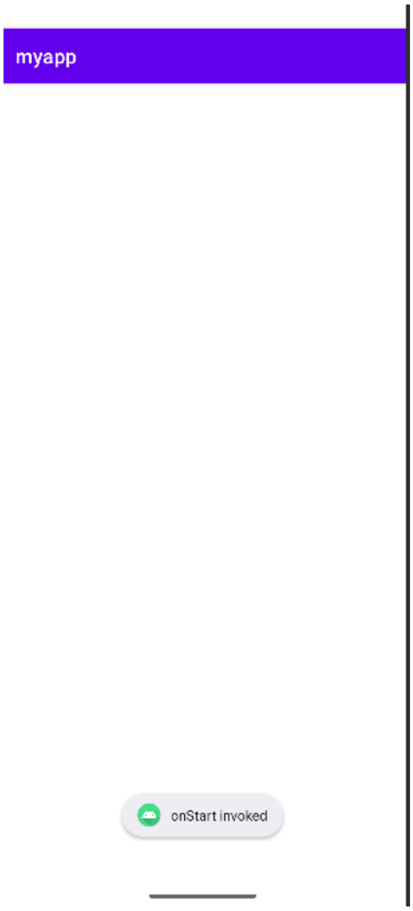
activity_main.xml

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

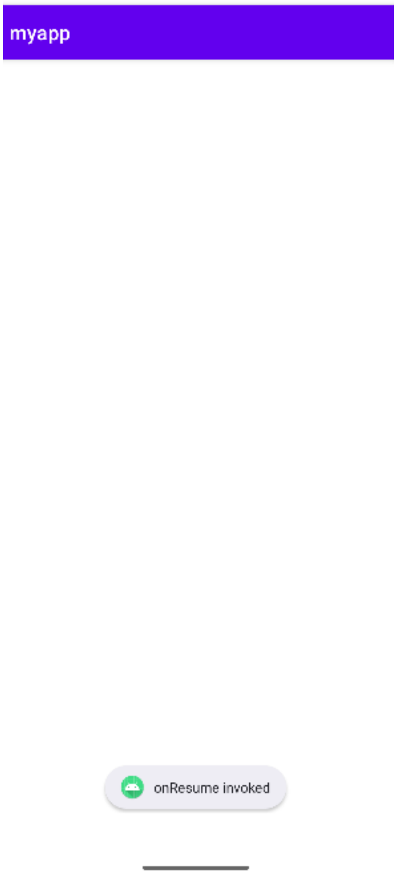
Output :



onCreate invoked

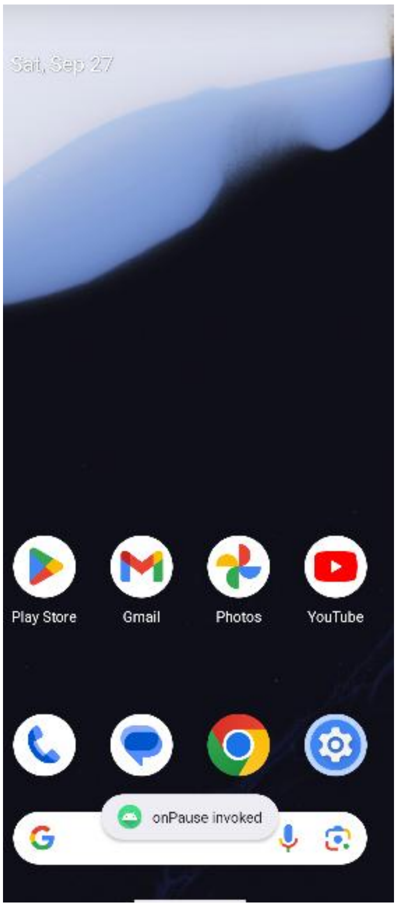


onStart invoked

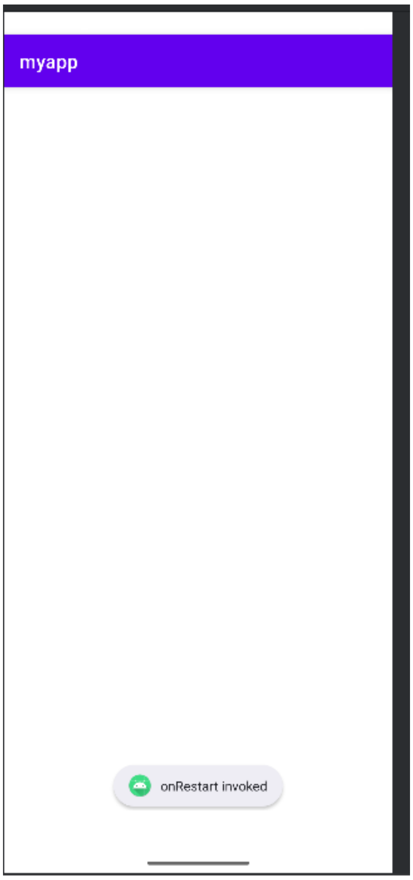


myapp

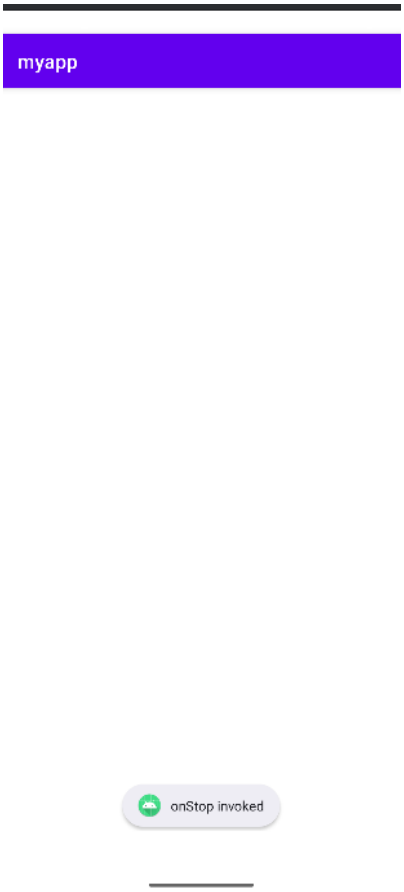
onResume invoked



onPause invoked



onRestart invoked



onStop invoked

2. Create an Employee Registration form using Linear Layout and Relative Layout.

Linear Layout :

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="match_parent">

   <LinearLayout
      android:orientation="vertical"
      android:padding="16dp"
      android:layout_marginTop="80dp"
      android:layout_width="match_parent"
      android:layout_height="wrap_content">

      <TextView
         android:layout_width="wrap_content"
         android:layout_height="wrap_content"
         android:text="Employee Registration"
         android:textSize="22sp"
         android:textStyle="bold"
         android:layout_gravity="center"/>

      <EditText
         android:id="@+id/etName"
         android:layout_width="match_parent"
         android:layout_height="wrap_content"
         android:hint="Full Name"
         android:inputType="textPersonName"
         android:layout_marginTop="16dp"/>

      <EditText
         android:id="@+id/etEmail"
         android:layout_width="match_parent"
         android:layout_height="wrap_content"
         android:hint="Email"
         android:inputType="textEmailAddress"
```

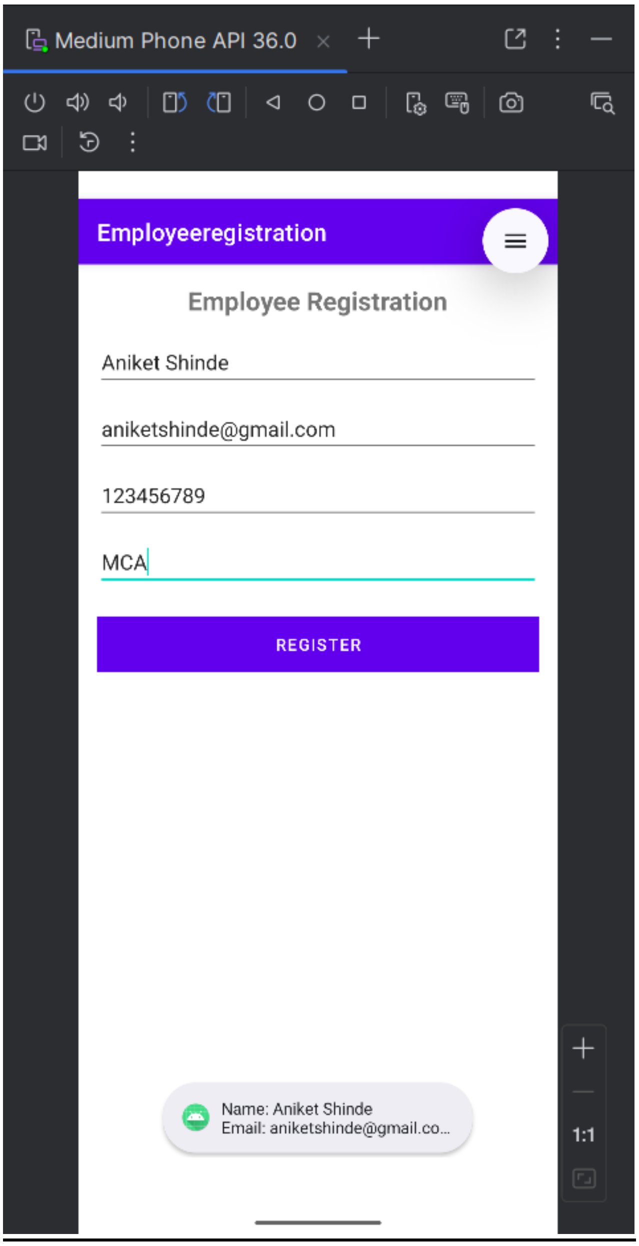
```
            android:layout_marginTop="12dp"/>

        <EditText
            android:id="@+id/etPhone"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Phone"
            android:inputType="phone"
            android:layout_marginTop="12dp"/>

        <EditText
            android:id="@+id/etDept"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Department"
            android:inputType="text"
            android:layout_marginTop="12dp"/>

        <Button
            android:id="@+id/btnRegister"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Register"
            android:layout_marginTop="24dp"
            android:background="@color/purple_500"
            android:textColor="@android:color/white"/>
    </LinearLayout>
</ScrollView>
```

Mainactivity.java

```
package com.example.employeeregistration;

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

public class MainActivity extends AppCompatActivity {
  private EditText etName, etEmail, etPhone, etDept;
  private Button btnRegister;

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

    // Initialize EditText fields
    etName = findViewById(R.id.etName);
    etEmail = findViewById(R.id.etEmail);
    etPhone = findViewById(R.id.etPhone);
    etDept = findViewById(R.id.etDept);

    // Initialize Button
    btnRegister = findViewById(R.id.btnRegister);

    // Set click listener for Register button
    btnRegister.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View v) {

        String name = etName.getText().toString().trim();
        String email = etEmail.getText().toString().trim();
        String phone = etPhone.getText().toString().trim();
        String dept = etDept.getText().toString().trim();

        if (name.isEmpty() || email.isEmpty() || phone.isEmpty() || dept.isEmpty())
{
          Toast.makeText(MainActivity.this, "Please fill all fields",
Toast.LENGTH_SHORT).show();
        } else {
          // Show entered data in a Toast
          String message = "Name: " + name + "\n" +
              "Email: " + email + "\n" +
              "Phone: " + phone + "\n" +
              "Dept: " + dept;
          Toast.makeText(MainActivity.this, message,
Toast.LENGTH_LONG).show();
        }
```

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

Output :



Employeeregistration

Employee Registration

Aniket Shinde

aniketshinde@gmail.com

123456789

MCA

REGISTER

Name: Aniket Shinde
Email: aniketshinde@gmail.co...

Relative Layout

Activity_main.xml :

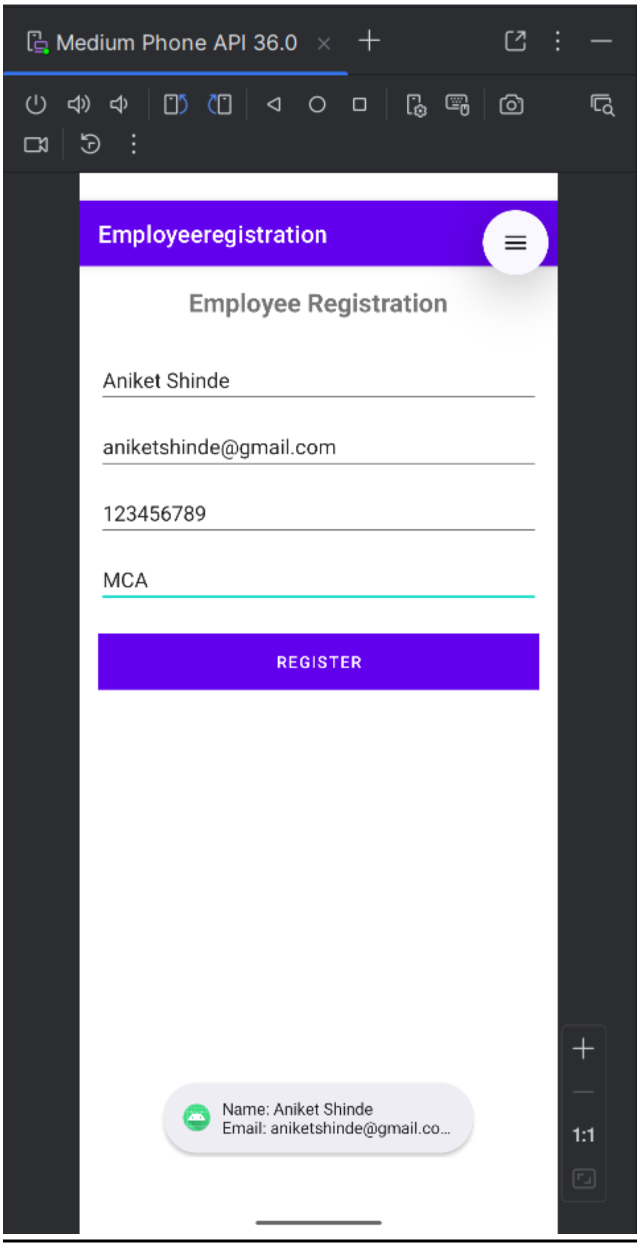
```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="match_parent">

   <RelativeLayout
      android:padding="16dp"
      android:layout_marginTop="80dp"
      android:layout_width="match_parent"
      android:layout_height="wrap_content">

      <TextView
         android:id="@+id/tvTitle"
         android:layout_width="wrap_content"
         android:layout_height="wrap_content"
         android:text="Employee Registration"
         android:textSize="22sp"
         android:textStyle="bold"
         android:layout_centerHorizontal="true"
         android:layout_marginBottom="20dp"/>

      <EditText
         android:id="@+id/etName"
         android:layout_width="match_parent"
         android:layout_height="wrap_content"
         android:hint="Full Name"
         android:inputType="textPersonName"
         android:layout_below="@id/tvTitle"
         android:layout_marginTop="10dp"/>

      <EditText
         android:id="@+id/etEmail"
         android:layout_width="match_parent"
         android:layout_height="wrap_content"
         android:hint="Email"
         android:inputType="textEmailAddress"
```

```
            android:layout_below="@id/etName"
            android:layout_marginTop="12dp"/>

        <EditText
            android:id="@+id/etPhone"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Phone"
            android:inputType="phone"
            android:layout_below="@id/etEmail"
            android:layout_marginTop="12dp"/>

        <EditText
            android:id="@+id/etDept"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Department"
            android:inputType="text"
            android:layout_below="@id/etPhone"
            android:layout_marginTop="12dp"/>

        <Button
            android:id="@+id/btnRegister"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Register"
            android:layout_below="@id/etDept"
            android:layout_marginTop="24dp"
            android:background="@color/purple_500"
            android:textColor="@android:color/white"/>
    </RelativeLayout>
</ScrollView>
```

Mainactivity.java :

```
package com.example.employeeregistration;

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

public class MainActivity extends AppCompatActivity {

    private EditText etName, etEmail, etPhone, etDept;
    private Button btnRegister;

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

        // Initialize EditText fields
        etName = findViewById(R.id.etName);
        etEmail = findViewById(R.id.etEmail);
        etPhone = findViewById(R.id.etPhone);
        etDept = findViewById(R.id.etDept);

        // Initialize Button
        btnRegister = findViewById(R.id.btnRegister);

        // Set click listener for Register button
        btnRegister.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                String name = etName.getText().toString().trim();
                String email = etEmail.getText().toString().trim();
                String phone = etPhone.getText().toString().trim();
                String dept = etDept.getText().toString().trim();

                if (name.isEmpty() || email.isEmpty() || phone.isEmpty() || dept.isEmpty())
{
                    Toast.makeText(MainActivity.this, "Please fill all fields",
Toast.LENGTH_SHORT).show();
                } else {
                    // Show entered data in a Toast
                    String message = "Name: " + name + "\n" +
                    "Email: " + email + "\n" +
                    "Phone: " + phone + "\n" +
                    "Dept: " + dept;
              Toast.makeText(MainActivity.this, message,
Toast.LENGTH_LONG).show();
           }
        }
     });
  }
}
```

Output:



Employeeregistration

Employee Registration

Aniket Shinde

aniketshinde@gmail.com

123456789

MCA

REGISTER

Name: Aniket Shinde
Email: aniketshinde@gmail.co...

3. Design a screen that displays the frame image and write a quote on that.

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:orientation="vertical"
   android:gravity="center"
   android:padding="16dp"
   android:background="#ffffff">

   <FrameLayout
      android:layout_width="300dp"
      android:layout_height="400dp"
      android:background="@drawable/bordered_bg"
      android:layout_marginBottom="30dp">

      <ImageView
         android:id="@+id/ivFrameImage"
         android:layout_width="match_parent"
         android:layout_height="match_parent"
         android:scaleType="centerCrop"
         android:src="@drawable/sample_image"
         android:contentDescription="Frame Image"/>

      <TextView
         android:id="@+id/tvQuote"
         android:layout_width="match_parent"
         android:layout_height="wrap_content"
         android:layout_gravity="bottom"
         android:text="Success is not final, Failure is not fatal"
         android:textSize="16sp"
         android:textStyle="bold"
         android:textColor="#ffffff"
         android:gravity="center"
         android:padding="16dp"
```

```
        android:background="#80000000"/>

    </FrameLayout>

</LinearLayout>
```

Bordered_bg.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
    <solid android:color="#f5f5f5" />
    <stroke
        android:width="8dp"
        android:color="#8B4513" />
    <corners android:radius="4dp" />
</shape>
```

Android_manifest.xml

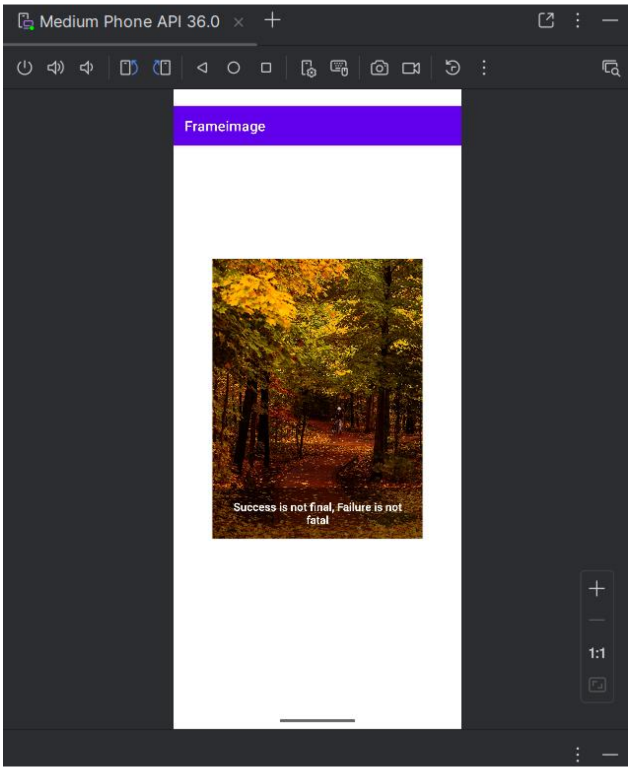
```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

    <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.Frameimage">
        <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>
```

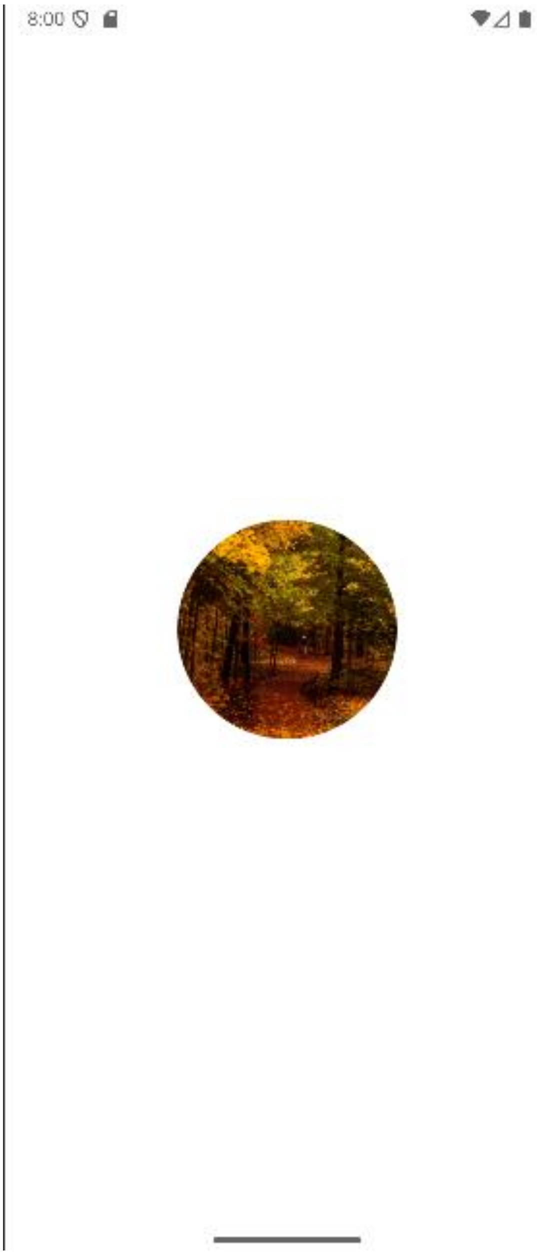
```
</manifest>
```

MainActivity.java

```
package com.example.frameimage;

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


public class MainActivity extends AppCompatActivity {

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

Output:




8:00

4. Create an android application that displays an image using frame layout and when the user clicks on that image another image should be displayed on the screen.

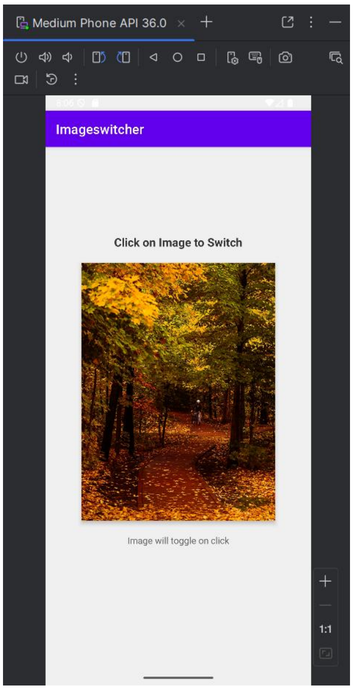
Mainactivity.java

```
package com.example.imageswitcher;

import android.os.Bundle;
import android.widget.ImageView;
import android.widget.FrameLayout;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

   private ImageView ivImage;
   private FrameLayout frameLayout;
   private boolean isFirstImage = true;

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

      ivImage = findViewById(R.id.ivImage);
      frameLayout = findViewById(R.id.frameLayout);

      frameLayout.setOnClickListener(v -> switchImage());
   }

   private void switchImage() {
      if (isFirstImage) {
         ivImage.setImageResource(R.drawable.image2);
         isFirstImage = false;
      } else {
         ivImage.setImageResource(R.drawable.image1);
         isFirstImage = true;
      }
   }
}
```

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center"
    android:padding="16dp"
    android:background="#f0f0f0">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Click on Image to Switch"
        android:textSize="18sp"
        android:textStyle="bold"
        android:layout_marginBottom="20dp"
        android:textColor="#333333"/>

    <FrameLayout
        android:id="@+id/frameLayout"
        android:layout_width="300dp"
        android:layout_height="400dp"
        android:background="#ffffff"
        android:elevation="4dp">

        <ImageView
            android:id="@+id/ivImage"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:scaleType="centerCrop"
            android:src="@drawable/image1"
            android:contentDescription="Switchable Image"/>

    </FrameLayout>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Image will toggle on click"
        android:textSize="14sp"
        android:layout_marginTop="20dp"
        android:textColor="#666666"/>

</LinearLayout>
```

Output :

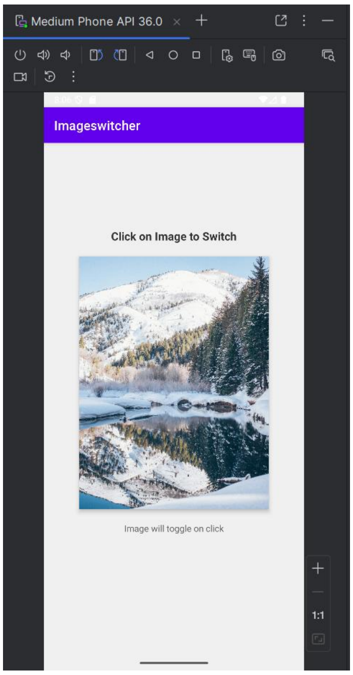


Medium Phone API 36.0

Imageswitcher

Click on Image to Switch

Image will toggle on click



Medium Phone API 36.0

Imageswitcher

Click on Image to Switch

Image will toggle on click

5. Create an android application to add two numbers and display result in Toast Message and AlertDialog.

mainActivity.java

```
package com.example.addnumbers;
import android.os.Bundle;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    private EditText etNum1, etNum2;
    private Button btnToast, btnAlert;

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

        etNum1 = findViewById(R.id.etNum1);
        etNum2 = findViewById(R.id.etNum2);
        btnToast = findViewById(R.id.btnToast);
        btnAlert = findViewById(R.id.btnAlert);

        btnToast.setOnClickListener(v -> showResultInToast());
        btnAlert.setOnClickListener(v -> showResultInAlert());
    }

    private void showResultInToast() {
        String num1Str = etNum1.getText().toString();
        String num2Str = etNum2.getText().toString();

        if (num1Str.isEmpty() || num2Str.isEmpty()) {
            Toast.makeText(this, "Please enter both numbers",
Toast.LENGTH_SHORT).show();
            return;
        }

        int num1 = Integer.parseInt(num1Str);
        int num2 = Integer.parseInt(num2Str);
        int sum = num1 + num2;

        Toast.makeText(this, "Sum: " + sum, Toast.LENGTH_LONG).show();
    }

    private void showResultInAlert() {
        String num1Str = etNum1.getText().toString();
        String num2Str = etNum2.getText().toString();

        if (num1Str.isEmpty() || num2Str.isEmpty()) {
            Toast.makeText(this, "Please enter both numbers",
Toast.LENGTH_SHORT).show();
            return;
        }

        int num1 = Integer.parseInt(num1Str);
        int num2 = Integer.parseInt(num2Str);
        int sum = num1 + num2;

        AlertDialog.Builder builder = new AlertDialog.Builder(this);
        builder.setTitle("Addition Result");
        builder.setMessage(num1 + " + " + num2 + " = " + sum);
        builder.setPositiveButton("OK", (dialog, which) -> dialog.dismiss());
        builder.create().show();
    }
}
```

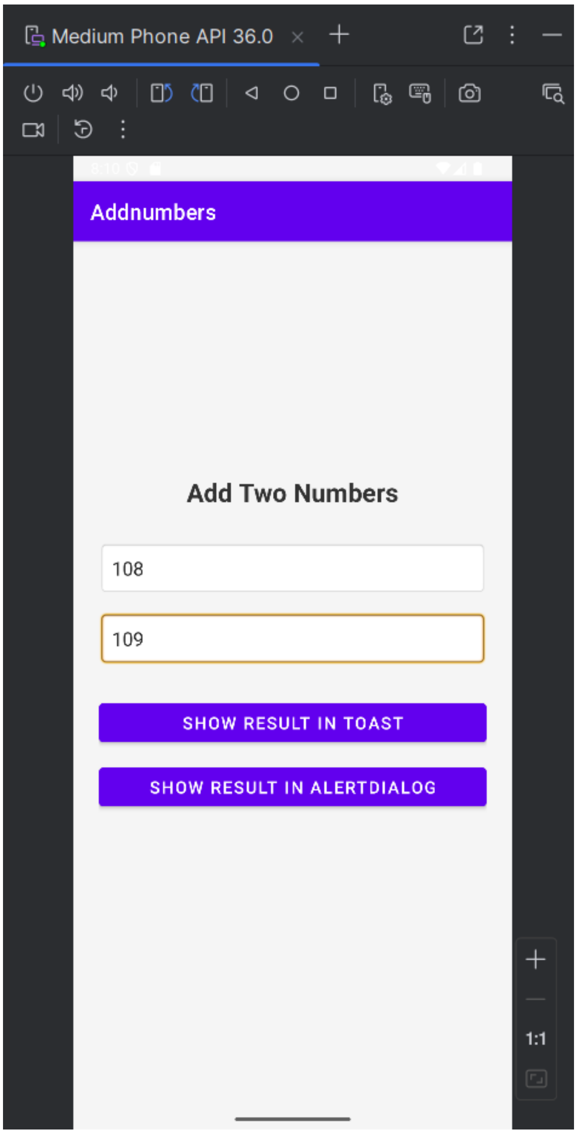
activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center"
    android:padding="24dp"
    android:background="#f5f5f5">
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Add Two Numbers"
    android:textSize="24sp"
    android:textStyle="bold"
    android:textColor="#333333"
    android:layout_marginBottom="30dp"/>

<EditText
    android:id="@+id/etNum1"
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:hint="Enter First Number"
    android:inputType="number"
    android:background="@android:drawable/editbox_background"
    android:paddingLeft="12dp"
    android:paddingRight="12dp"
    android:layout_marginBottom="16dp"/>

<EditText
    android:id="@+id/etNum2"
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:hint="Enter Second Number"
    android:inputType="number"
    android:background="@android:drawable/editbox_background"
    android:paddingLeft="12dp"
    android:paddingRight="12dp"
    android:layout_marginBottom="30dp"/>

<Button
    android:id="@+id/btnToast"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Show Result in Toast"
    android:textSize="16sp"
    android:layout_marginBottom="12dp"/>

<Button
```

```
    android:id="@+id/btnAlert"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Show Result in AlertDialog"
    android:textSize="16sp"/>

</LinearLayout>
```

Output:

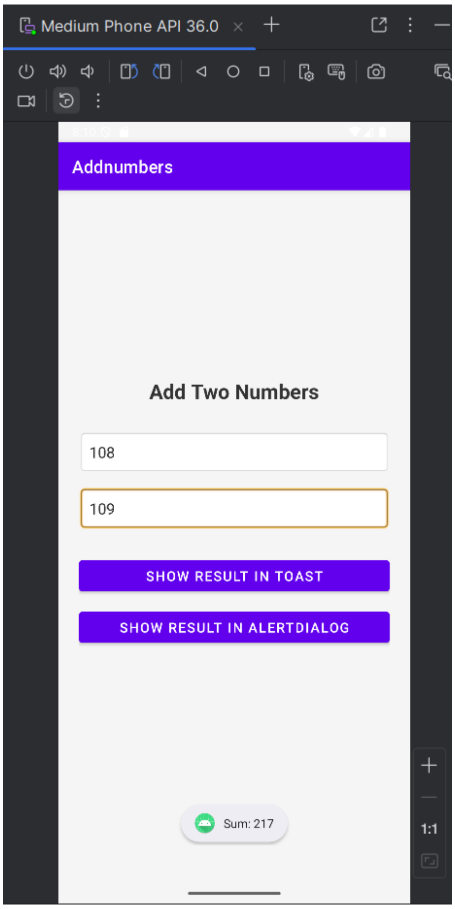


Addnumbers

Add Two Numbers

108

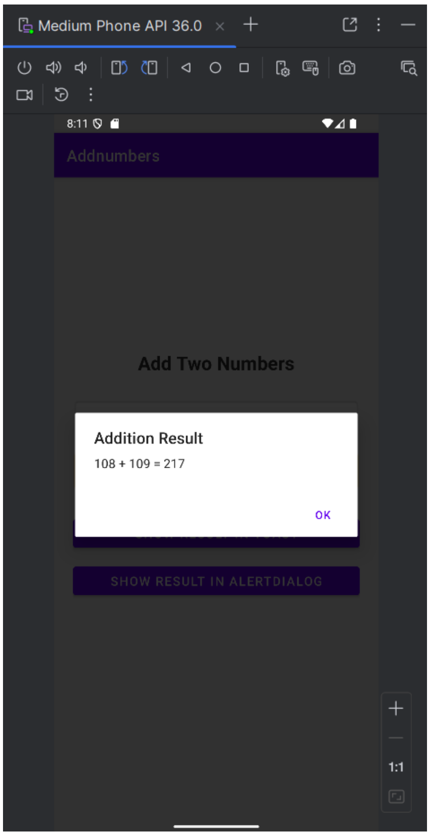
109



108

109

Sum: 217



Addition Result

108 + 109 = 217

OK

6. Create an Android application for the student registration form using the relative layout. Display the entered details on the second page using explicit intent.

Mainactivity.java

```
package com.example.studentregistration;

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

public class MainActivity extends AppCompatActivity {

  private EditText etStudentName, etStudentId, etEmail, etPhone, etCourse;
  private Button btnSubmit;

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

    etStudentName = findViewById(R.id.etStudentName);
    etStudentId = findViewById(R.id.etStudentId);
    etEmail = findViewById(R.id.etEmail);
    etPhone = findViewById(R.id.etPhone);
    etCourse = findViewById(R.id.etCourse);
    btnSubmit = findViewById(R.id.btnSubmit);

    btnSubmit.setOnClickListener(v -> submitForm());
  }

  private void submitForm() {
    String name = etStudentName.getText().toString();
    String id = etStudentId.getText().toString();
    String email = etEmail.getText().toString();
    String phone = etPhone.getText().toString();
    String course = etCourse.getText().toString();
```

```
        if (name.isEmpty() || id.isEmpty() || email.isEmpty() || phone.isEmpty() ||
course.isEmpty()) {
            Toast.makeText(this, "Please fill all fields", Toast.LENGTH_SHORT).show();
            return;
        }

        Intent intent = new Intent(MainActivity.this, DetailActivity.class);
        intent.putExtra("name", name);
        intent.putExtra("id", id);
        intent.putExtra("email", email);
        intent.putExtra("phone", phone);
        intent.putExtra("course", course);
        startActivity(intent);
    }
}
```

detailactivity.java

```
package com.example.studentregistration;

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

public class DetailActivity extends AppCompatActivity {

    private TextView tvName, tvId, tvEmail, tvPhone, tvCourse;
    private Button btnBack;

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

        tvName = findViewById(R.id.tvName);
        tvId = findViewById(R.id.tvId);
        tvEmail = findViewById(R.id.tvEmail);
        tvPhone = findViewById(R.id.tvPhone);
        tvCourse = findViewById(R.id.tvCourse);
        btnBack = findViewById(R.id.btnBack);

        Bundle bundle = getIntent().getExtras();
        if (bundle != null) {
            tvName.setText("Name: " + bundle.getString("name"));
            tvId.setText("ID: " + bundle.getString("id"));
            tvEmail.setText("Email: " + bundle.getString("email"));
            tvPhone.setText("Phone: " + bundle.getString("phone"));
            tvCourse.setText("Course: " + bundle.getString("course"));
        }

        btnBack.setOnClickListener(v -> finish());
    }
}
```

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp"
    android:background="#f5f5f5">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Student Registration Form"
        android:textSize="22sp"
        android:textStyle="bold"
        android:textColor="#333333"
        android:gravity="center"
        android:layout_marginBottom="20dp"/>

    <EditText
        android:id="@+id/etStudentName"
        android:layout_width="match_parent"
        android:layout_height="45dp"
        android:hint="Enter Student Name"
        android:inputType="text"
        android:background="@android:drawable/editbox_background"
        android:paddingLeft="12dp"
        android:paddingRight="12dp"
        android:layout_marginBottom="12dp"/>

    <EditText
        android:id="@+id/etStudentId"
        android:layout_width="match_parent"
        android:layout_height="45dp"
        android:hint="Enter Student ID"
        android:inputType="number"
        android:background="@android:drawable/editbox_background"
        android:paddingLeft="12dp"
        android:paddingRight="12dp"
        android:layout_marginBottom="12dp"/>

    <EditText
        android:id="@+id/etEmail"
        android:layout_width="match_parent"
        android:layout_height="45dp"
        android:hint="Enter Email"
        android:inputType="textEmailAddress"
        android:background="@android:drawable/editbox_background"
        android:paddingLeft="12dp"
        android:paddingRight="12dp"
        android:layout_marginBottom="12dp"/>

    <EditText
        android:id="@+id/etPhone"
        android:layout_width="match_parent"
        android:layout_height="45dp"
        android:hint="Enter Phone Number"
        android:inputType="phone"
        android:background="@android:drawable/editbox_background"
        android:paddingLeft="12dp"
        android:paddingRight="12dp"
        android:layout_marginBottom="12dp"/>
    <EditText
        android:id="@+id/etCourse"
        android:layout_width="match_parent"
        android:layout_height="45dp"
        android:hint="Enter Course"
        android:inputType="text"
        android:background="@android:drawable/editbox_background"
        android:paddingLeft="12dp"
        android:paddingRight="12dp"
        android:layout_marginBottom="20dp"/>

    <Button
        android:id="@+id/btnSubmit"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Submit"
        android:textSize="16sp"/>

</LinearLayout>
```

Activity_detail.xml

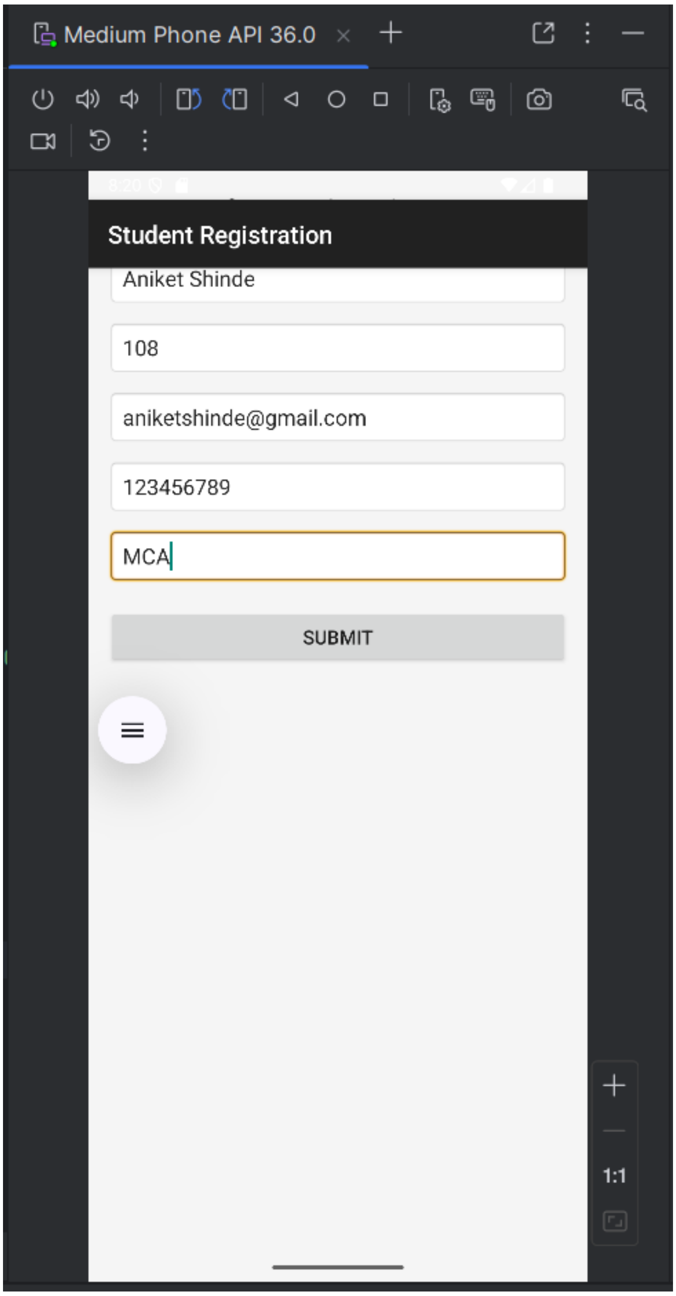
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp"
    android:background="#f5f5f5">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Student Details"
        android:textSize="22sp"
        android:textStyle="bold"
        android:textColor="#333333"
        android:gravity="center"
        android:layout_marginBottom="30dp"/>
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:background="#ffffff"
    android:padding="16dp"
    android:layout_marginBottom="20dp">

    <TextView
        android:id="@+id/tvName"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Name: "
        android:textSize="16sp"
        android:layout_marginBottom="8dp"/>

    <TextView
        android:id="@+id/tvId"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="ID: "
        android:textSize="16sp"
        android:layout_marginBottom="8dp"/>

    <TextView
        android:id="@+id/tvEmail"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Email: "
        android:textSize="16sp"
        android:layout_marginBottom="8dp"/>

    <TextView
        android:id="@+id/tvPhone"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Phone: "
        android:textSize="16sp"
        android:layout_marginBottom="8dp"/>
        <TextView
            android:id="@+id/tvCourse"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Course: "
            android:textSize="16sp"/>

    </LinearLayout>

    <Button
        android:id="@+id/btnBack"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Go Back"
        android:textSize="16sp"/>

</LinearLayout>
```

Androidmanifest.xml

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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="Student Registration"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.AppCompat.Light.DarkActionBar">
        <activity android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".DetailActivity"
        android:exported="false"/>
    </application>

</manifest>
```

Output :



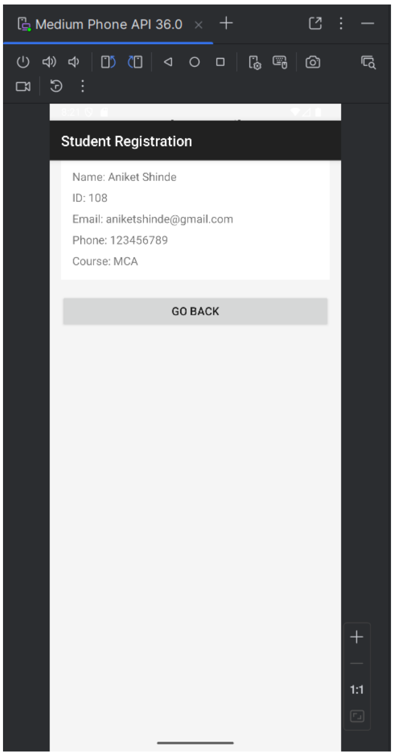
Aniket Shinde

108

aniketshinde@gmail.com

123456789

MCA



Student Registration

Name: Aniket Shinde

ID: 108

Email: aniketshinde@gmail.com

Phone: 123456789

Course: MCA

7. Create an application to implement implicit intent with functionality to open camera, Gallery, Contact, Dial, Browser.

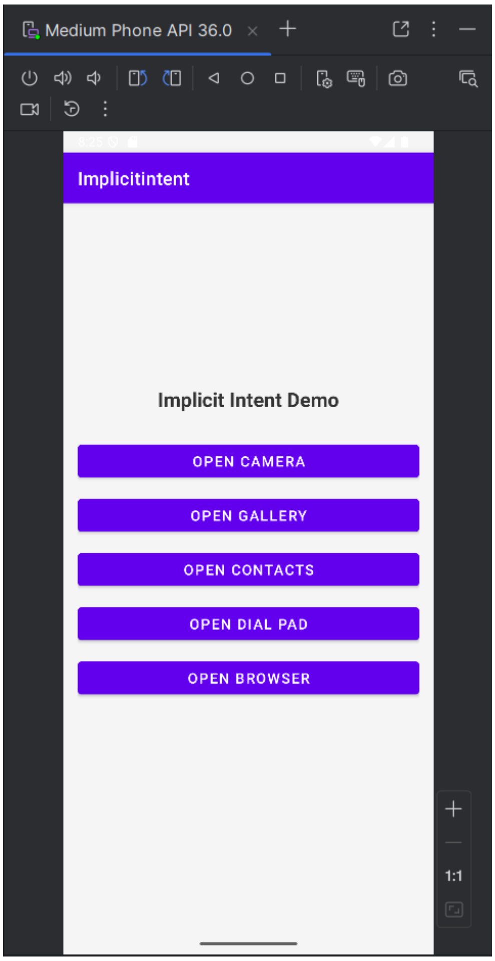
Mainactivity.java

```
package com.example.implicitintent;

import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.provider.ContactsContract;
import android.provider.MediaStore;
import android.widget.Button;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    private Button btnCamera, btnGallery, btnContact, btnDial, btnBrowser;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        btnCamera = findViewById(R.id.btnCamera);
        btnGallery = findViewById(R.id.btnGallery);
        btnContact = findViewById(R.id.btnContact);
        btnDial = findViewById(R.id.btnDial);
        btnBrowser = findViewById(R.id.btnBrowser);

        btnCamera.setOnClickListener(v -> openCamera());
        btnGallery.setOnClickListener(v -> openGallery());
        btnContact.setOnClickListener(v -> openContact());
        btnDial.setOnClickListener(v -> openDial());
        btnBrowser.setOnClickListener(v -> openBrowser());
    }

    private void openCamera() {
        Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
        startActivity(intent);
    }

    private void openGallery() {
        Intent intent = new Intent(Intent.ACTION_PICK);
        intent.setType("image/*");
        startActivity(intent);
    }

    private void openContact() {
        Intent intent = new Intent(Intent.ACTION_PICK);
        intent.setData(ContactsContract.Contacts.CONTENT_URI);
        startActivity(intent);
    }

    private void openDial() {
        Intent intent = new Intent(Intent.ACTION_DIAL);
        intent.setData(Uri.parse("tel:9876543210"));
        startActivity(intent);
    }

    private void openBrowser() {
        Intent intent = new Intent(Intent.ACTION_VIEW);
        intent.setData(Uri.parse("https://www.google.com"));
        startActivity(intent);
    }
}
```

activity_main.xml

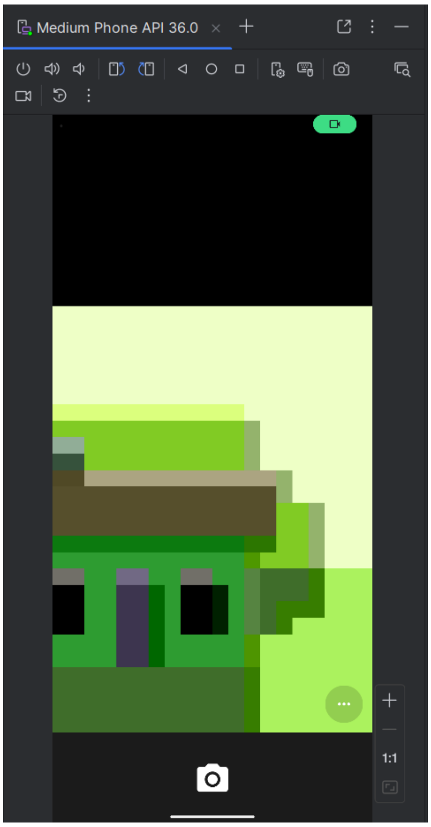
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center"
    android:padding="16dp"
    android:background="#f5f5f5">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Implicit Intent Demo"
        android:textSize="22sp"
        android:textStyle="bold"
        android:textColor="#333333"
        android:layout_marginBottom="30dp"/>

    <Button
        android:id="@+id/btnCamera"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Open Camera"
        android:textSize="16sp"
        android:layout_marginBottom="12dp"/>

    <Button
        android:id="@+id/btnGallery"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Open Gallery"
        android:textSize="16sp"
        android:layout_marginBottom="12dp"/>

    <Button
        android:id="@+id/btnContact"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Open Contacts"
        android:textSize="16sp"
        android:layout_marginBottom="12dp"/>

    <Button
        android:id="@+id/btnDial"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Open Dial Pad"
        android:textSize="16sp"
        android:layout_marginBottom="12dp"/>

    <Button
        android:id="@+id/btnBrowser"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Open Browser"
        android:textSize="16sp"/>

</LinearLayout>
```

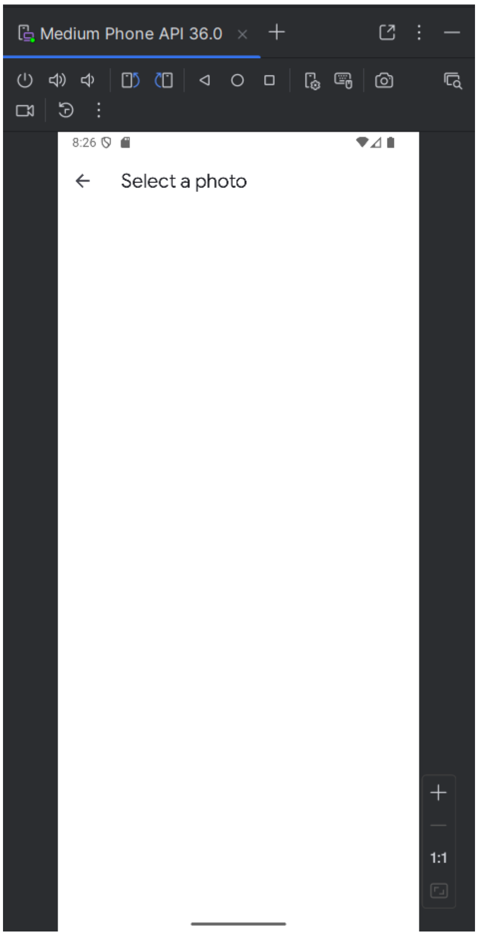
Output:



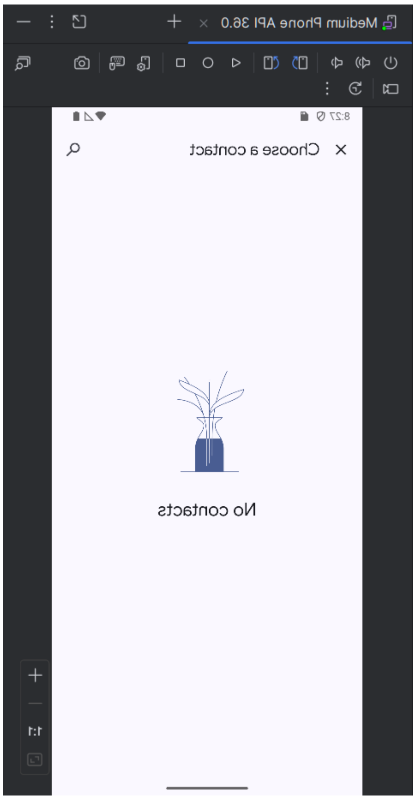
Implicitintent

Implicit Intent Demo

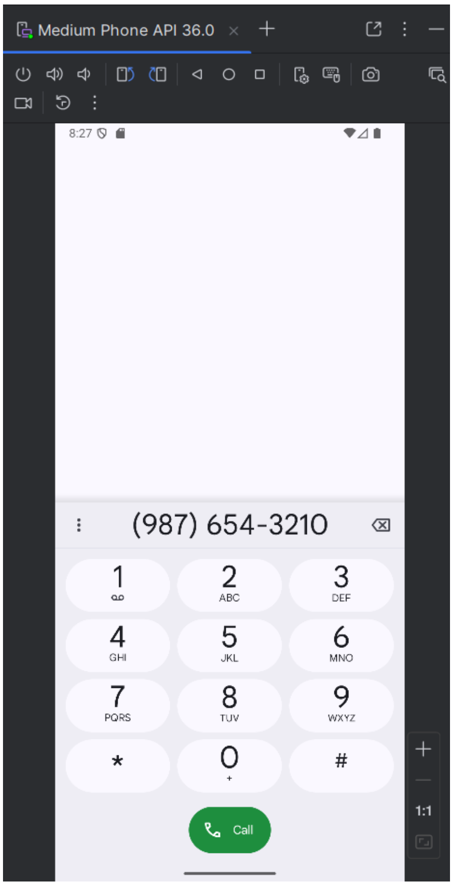




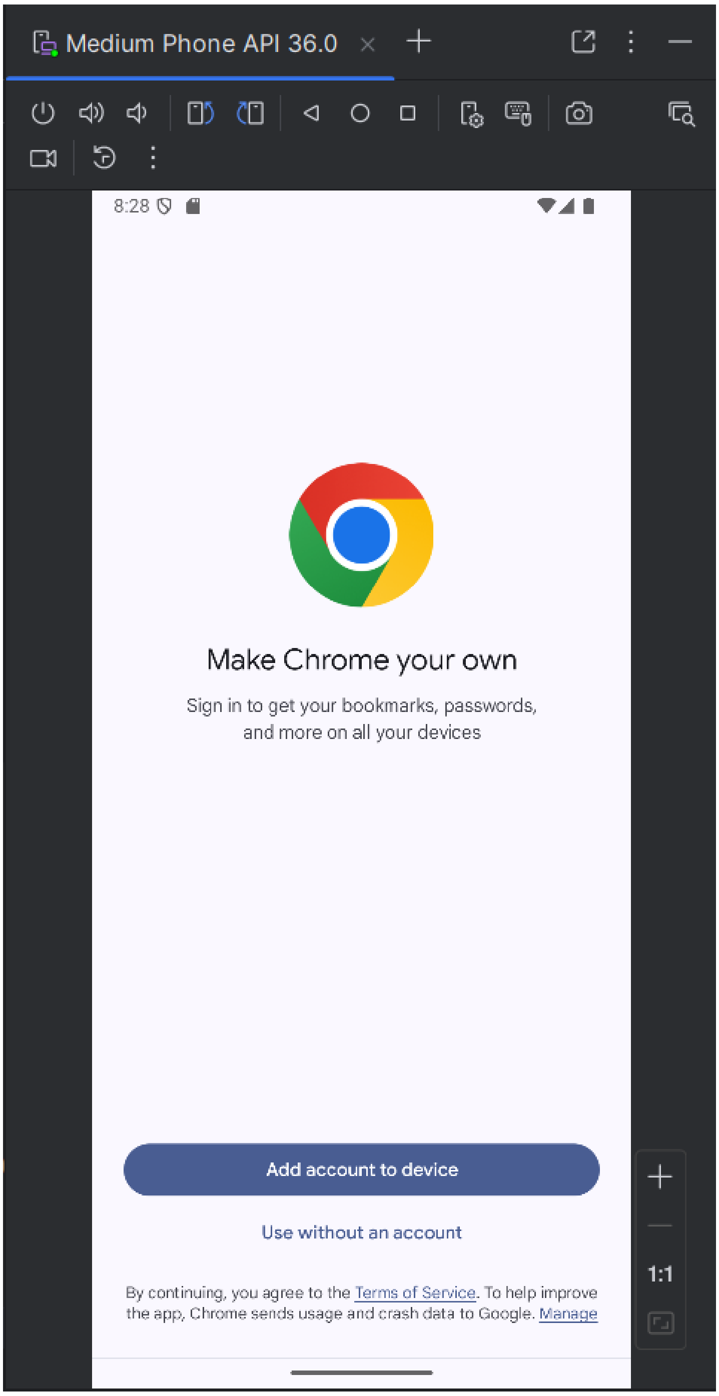
Select a photo



Medium Phone API 36.0
8:27
Choose a contact
No contacts
1:1



Medium Phone API 36.0
8:27
(987) 654-3210
1
2 ABC
3 DEF
4 GHI
5 JKL
6 MNO
7 PQRS
8 TUV
9 WXYZ
*
0 +
#
Call
1:1



Make Chrome your own

Sign in to get your bookmarks, passwords, and more on all your devices

Add account to device

Use without an account

By continuing, you agree to the Terms of Service. To help improve the app, Chrome sends usage and crash data to Google. Manage

AndroidManifest.xml

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

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

  <application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="Implicit Intent"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/Theme.AppCompat.Light.DarkActionBar">
    <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>
```

8. Create an application which has two buttons. When the user clicks on the first button the first fragment will be displayed and when the user clicks on the second button the second fragment will be displayed.

Mainactivity.java

```
package com.example.fragmentswitching;

import android.os.Bundle;
import android.widget.Button;
import androidx.appcompat.app.AppCompatActivity;
import androidx.fragment.app.FragmentManager;
import androidx.fragment.app.FragmentTransaction;

public class MainActivity extends AppCompatActivity {

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

    Button btnFragment1 = findViewById(R.id.btnFragment1);
    Button btnFragment2 = findViewById(R.id.btnFragment2);

    btnFragment1.setOnClickListener(v -> loadFragment(new Fragment1()));
    btnFragment2.setOnClickListener(v -> loadFragment(new Fragment2()));

    if (savedInstanceState == null) {
      loadFragment(new Fragment1());
    }
  }

  private void loadFragment(androidx.fragment.app.Fragment fragment) {
    FragmentManager fragmentManager = getSupportFragmentManager();
    FragmentTransaction fragmentTransaction =
fragmentManager.beginTransaction();
    fragmentTransaction.replace(R.id.frameLayout, fragment);
    fragmentTransaction.addToBackStack(null);
    fragmentTransaction.commit();
    }
}
```

fragment1.java

```
package com.example.fragmentswitching;

import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;

public class Fragment1 extends Fragment {

    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, @Nullable
ViewGroup container, @Nullable Bundle savedInstanceState) {
        return inflater.inflate(R.layout.fragment_1, container, false);
    }
}
```

fragment2.java

```
package com.example.fragmentswitching;

import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;

public class Fragment2 extends Fragment {

    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, @Nullable
ViewGroup container, @Nullable Bundle savedInstanceState) {
        return inflater.inflate(R.layout.fragment_2, container, false);
    }
}
```

activity_main.xml

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

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

        <Button
            android:id="@+id/btnFragment1"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="Fragment 1" />

        <Button
            android:id="@+id/btnFragment2"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="Fragment 2" />
    </LinearLayout>

    <FrameLayout
        android:id="@+id/frameLayout"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

</LinearLayout>
```

Fragment_1.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:orientation="vertical"
    android:background="#FFEB3B">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="This is Fragment 1"
        android:textSize="24sp" />

</LinearLayout>
```

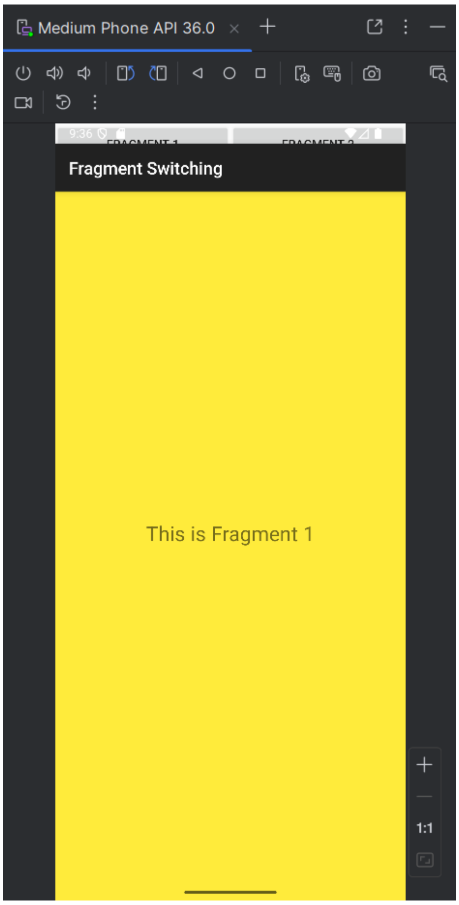
Fragment_2.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:orientation="vertical"
    android:background="#8BC34A">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="This is Fragment 2"
        android:textSize="24sp" />

</LinearLayout>
```

androidManifest.xml

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

   <application
      android:allowBackup="true"
      android:icon="@mipmap/ic_launcher"
      android:label="Fragment Switching"
      android:roundIcon="@mipmap/ic_launcher_round"
      android:supportsRtl="true"
      android:theme="@style/Theme.AppCompat.Light.DarkActionBar">
      <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>
```

Output :



Fragment Switching

This is Fragment 1

9. Write an Android application with five check boxes to list the 5 subjects of your class and radio buttons to display gender. Display the selected subject name when you click any one of the checkboxes and gender in the alert dialog box.

MainActivity.java

```
package com.example.checkboxradiobutton;

import android.os.Bundle;
import android.widget.CheckBox;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.Button;
import android.widget.Toast;
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
```

```
public class MainActivity extends AppCompatActivity {

    private CheckBox cbJava, cbPython, cbAndroid, cbWeb, cbDatabases;
    private RadioGroup rgGender;
    private RadioButton rbMale, rbFemale;
    private Button btnSubmit;

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

        cbJava = findViewById(R.id.cbJava);
        cbPython = findViewById(R.id.cbPython);
        cbAndroid = findViewById(R.id.cbAndroid);
        cbWeb = findViewById(R.id.cbWeb);
        cbDatabases = findViewById(R.id.cbDatabases);
        rgGender = findViewById(R.id.rgGender);
        rbMale = findViewById(R.id.rbMale);
        rbFemale = findViewById(R.id.rbFemale);
        btnSubmit = findViewById(R.id.btnSubmit);

        cbJava.setOnCheckedChangeListener((buttonView, isChecked) -> {
            if (isChecked) {
                Toast.makeText(MainActivity.this, "Java Selected",
Toast.LENGTH_SHORT).show();
            }
        });

        cbPython.setOnCheckedChangeListener((buttonView, isChecked) -> {
            if (isChecked) {
                Toast.makeText(MainActivity.this, "Python Selected",
Toast.LENGTH_SHORT).show();
            }
        });

        cbAndroid.setOnCheckedChangeListener((buttonView, isChecked) -> {
            if (isChecked) {
                Toast.makeText(MainActivity.this, "Android Selected",
Toast.LENGTH_SHORT).show();
            }
        });

        cbWeb.setOnCheckedChangeListener((buttonView, isChecked) -> {
            if (isChecked) {
                Toast.makeText(MainActivity.this, "Web Development Selected",
Toast.LENGTH_SHORT).show();
            }
        });

        cbDatabases.setOnCheckedChangeListener((buttonView, isChecked) -> {
            if (isChecked) {
                Toast.makeText(MainActivity.this, "Databases Selected",
Toast.LENGTH_SHORT).show();
            }
        });

        btnSubmit.setOnClickListener(v -> displaySelections());
    }

    private void displaySelections() {
        StringBuilder subjects = new StringBuilder();
        subjects.append("Selected Subjects:\n");

        if (cbJava.isChecked()) subjects.append("• Java\n");
        if (cbPython.isChecked()) subjects.append("• Python\n");
        if (cbAndroid.isChecked()) subjects.append("• Android\n");
        if (cbWeb.isChecked()) subjects.append("• Web Development\n");
        if (cbDatabases.isChecked()) subjects.append("• Databases\n");

        String gender = "";
        int selectedId = rgGender.getCheckedRadioButtonId();
        if (selectedId == rbMale.getId()) {
            gender = "Male";
        } else if (selectedId == rbFemale.getId()) {
            gender = "Female";
        } else {
            gender = "Not Selected";
        }

        AlertDialog.Builder builder = new AlertDialog.Builder(this);
        builder.setTitle("Your Selections");
        builder.setMessage(subjects.toString() + "\nGender: " + gender);
        builder.setPositiveButton("OK", (dialog, which) -> dialog.dismiss());
        builder.create().show();
    }
}
```

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#f5f5f5">

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

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Select Your Subjects"
            android:textSize="20sp"
            android:textStyle="bold"
            android:textColor="#333333"
            android:layout_marginBottom="16dp"/>

        <CheckBox
            android:id="@+id/cbJava"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Java"
            android:textSize="16sp"
            android:layout_marginBottom="8dp"/>
```

```
<CheckBox
    android:id="@+id/cbPython"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Python"
    android:textSize="16sp"
    android:layout_marginBottom="8dp"/>

<CheckBox
    android:id="@+id/cbAndroid"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Android"
    android:textSize="16sp"
    android:layout_marginBottom="8dp"/>

<CheckBox
    android:id="@+id/cbWeb"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Web Development"
    android:textSize="16sp"
    android:layout_marginBottom="8dp"/>

<CheckBox
    android:id="@+id/cbDatabases"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Databases"
    android:textSize="16sp"
    android:layout_marginBottom="24dp"/>

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Select Your Gender"
    android:textSize="20sp"
    android:textStyle="bold"
    android:textColor="#333333"
```

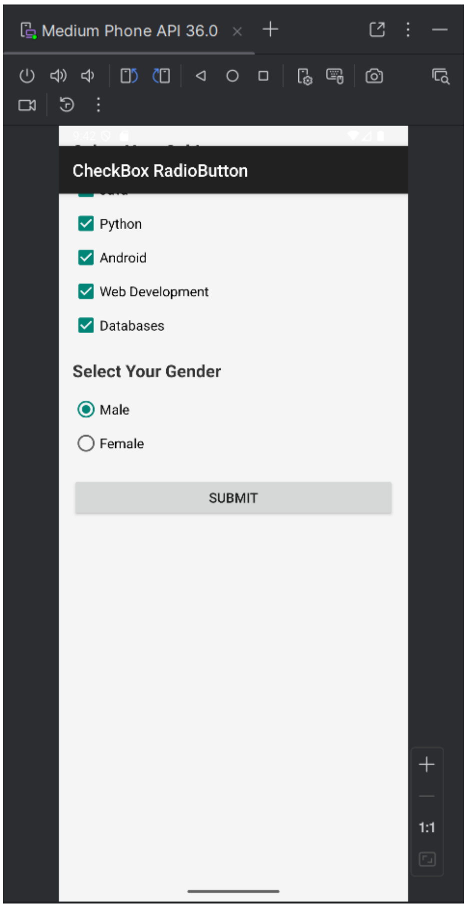
```
        android:layout_marginBottom="16dp"/>

    <RadioGroup
        android:id="@+id/rgGender"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="24dp">

        <RadioButton
            android:id="@+id/rbMale"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Male"
            android:textSize="16sp"
            android:layout_marginBottom="8dp"/>

        <RadioButton
            android:id="@+id/rbFemale"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Female"
            android:textSize="16sp"/>

    </RadioGroup>

    <Button
        android:id="@+id/btnSubmit"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Submit"
        android:textSize="16sp"/>

  </LinearLayout>

</ScrollView>
```

AndroidManifest.xml

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

```
    package="com.example.checkboxradiobutton">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="CheckBox RadioButton"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.AppCompat.Light.DarkActionBar">
        <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>
```

Output



CheckBox RadioButton

☑ Python

☑ Android

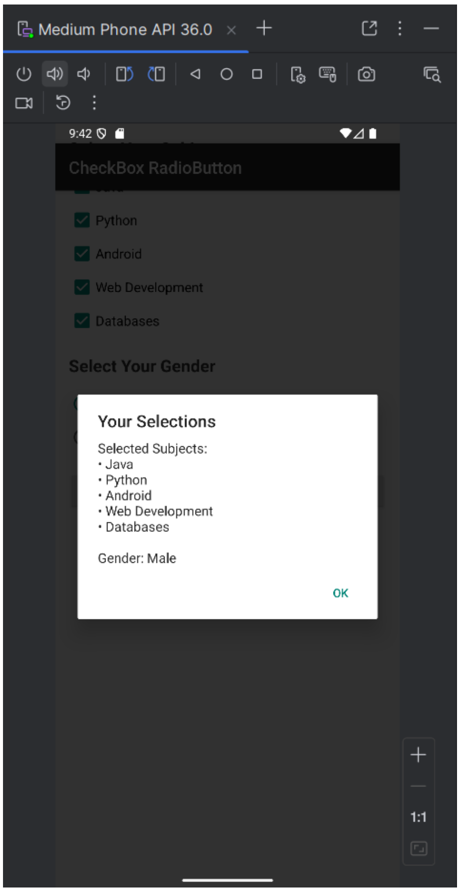
☑ Web Development

☑ Databases

Select Your Gender

● Male

○ Female



Your Selections

Selected Subjects:
• Java
• Python
• Android
• Web Development
• Databases

Gender: Male

10. Create a basic calculator to perform arithmetic operations with divide-by-zero validation. (using Alert box).

Mainactivity.java

```
package com.example.calculator;
import android.os.Bundle;
import android.widget.EditText;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    private EditText etNum1, etNum2;
    private TextView tvResult;
    private Button btnAdd, btnSubtract, btnMultiply, btnDivide, btnClear;

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

        etNum1 = findViewById(R.id.etNum1);
        etNum2 = findViewById(R.id.etNum2);
        tvResult = findViewById(R.id.tvResult);
        btnAdd = findViewById(R.id.btnAdd);
        btnSubtract = findViewById(R.id.btnSubtract);
        btnMultiply = findViewById(R.id.btnMultiply);
        btnDivide = findViewById(R.id.btnDivide);
        btnClear = findViewById(R.id.btnClear);

        btnAdd.setOnClickListener(v -> performOperation('+'));
        btnSubtract.setOnClickListener(v -> performOperation('-'));
        btnMultiply.setOnClickListener(v -> performOperation('*'));
        btnDivide.setOnClickListener(v -> performOperation('/'));
        btnClear.setOnClickListener(v -> clearFields());
    }
```

```
  private void performOperation(char operator) {
    String num1Str = etNum1.getText().toString();
    String num2Str = etNum2.getText().toString();

    if (num1Str.isEmpty() || num2Str.isEmpty()) {
      Toast.makeText(this, "Please enter both numbers",
Toast.LENGTH_SHORT).show();
      return;
    }

    double num1 = Double.parseDouble(num1Str);
    double num2 = Double.parseDouble(num2Str);
    double result = 0;
    boolean isValid = true;

    switch (operator) {
      case '+':
        result = num1 + num2;
        break;
      case '-':
        result = num1 - num2;
        break;
      case '*':
        result = num1 * num2;
        break;
      case '/':
        if (num2 == 0) {
          showDivideByZeroAlert();
          isValid = false;
        } else {
          result = num1 / num2;
        }
        break;
    }

    if (isValid) {
      tvResult.setText("Result: " + result);
    }
  }
```

```
    private void showDivideByZeroAlert() {
        AlertDialog.Builder builder = new AlertDialog.Builder(this);
        builder.setTitle("Error");
        builder.setMessage("Cannot divide by zero!");
        builder.setIcon(android.R.drawable.ic_dialog_alert);
        builder.setPositiveButton("OK", (dialog, which) -> dialog.dismiss());
        builder.create().show();
    }

    private void clearFields() {
        etNum1.setText("");
        etNum2.setText("");
        tvResult.setText("Result: ");
        etNum1.requestFocus();
    }
}
```

activity_main.xml

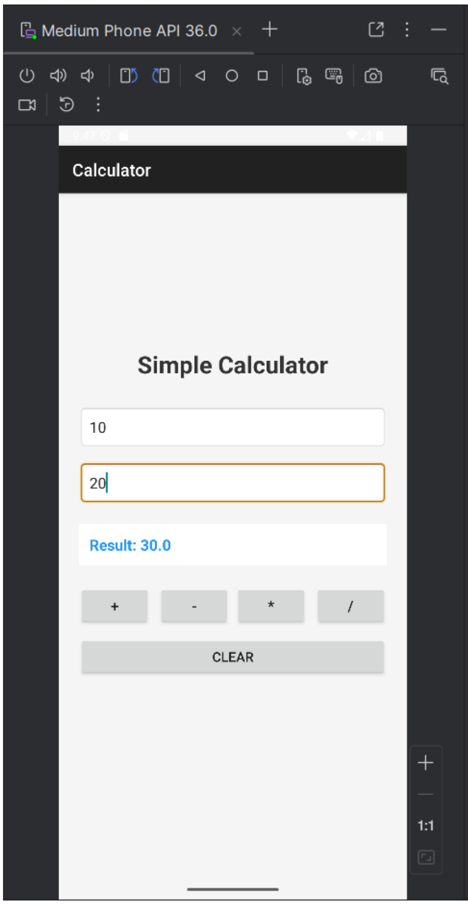
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center"
    android:padding="24dp"
    android:background="#f5f5f5">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Simple Calculator"
        android:textSize="28sp"
        android:textStyle="bold"
        android:textColor="#333333"
        android:layout_marginBottom="30dp"/>

    <EditText
        android:id="@+id/etNum1"
        android:layout_width="match_parent"
        android:layout_height="50dp"
        android:hint="Enter First Number"
        android:inputType="numberDecimal"
        android:background="@android:drawable/editbox_background"
        android:paddingLeft="12dp"
        android:paddingRight="12dp"
        android:layout_marginBottom="16dp"/>

    <EditText
        android:id="@+id/etNum2"
        android:layout_width="match_parent"
        android:layout_height="50dp"
        android:hint="Enter Second Number"
        android:inputType="numberDecimal"
        android:background="@android:drawable/editbox_background"
        android:paddingLeft="12dp"
        android:paddingRight="12dp"
        android:layout_marginBottom="24dp"/>

    <TextView
        android:id="@+id/tvResult"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Result: "
        android:textSize="18sp"
        android:textStyle="bold"
        android:textColor="#2196F3"
        android:padding="12dp"
        android:layout_marginBottom="24dp"
        android:background="#ffffff"/>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:layout_marginBottom="12dp">

        <Button
            android:id="@+id/btnAdd"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="+"
            android:textSize="18sp"
            android:layout_marginRight="6dp"/>

        <Button
            android:id="@+id/btnSubtract"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="-"
            android:textSize="18sp"
            android:layout_marginLeft="3dp"
            android:layout_marginRight="3dp"/>

        <Button
            android:id="@+id/btnMultiply"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="*"
            android:textSize="18sp"
            android:layout_marginLeft="3dp"
            android:layout_marginRight="3dp"/>

        <Button
            android:id="@+id/btnDivide"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="/"
            android:textSize="18sp"
            android:layout_marginLeft="6dp"/>

    </LinearLayout>

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

```
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Clear"
        android:textSize="16sp"/>

</LinearLayout>
```

AndroidManifest.xml

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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="Calculator"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.AppCompat.Light.DarkActionBar">
        <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>
```

Output :

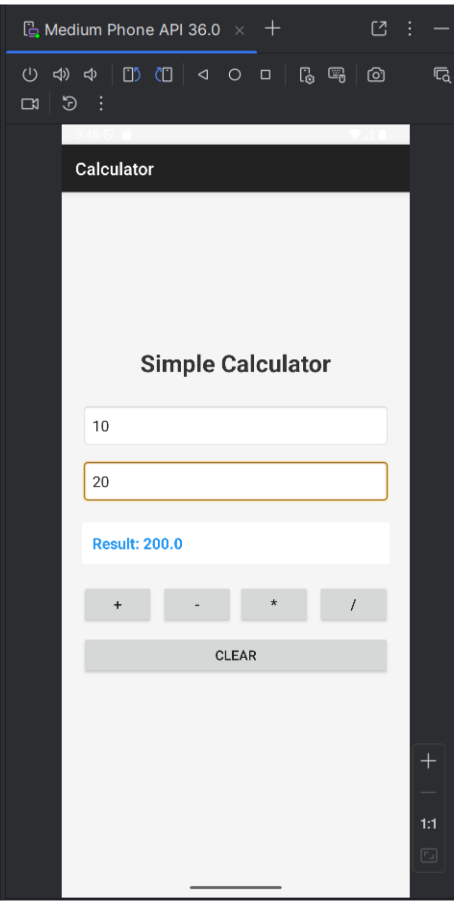


Simple Calculator

10

20

Result: 30.0

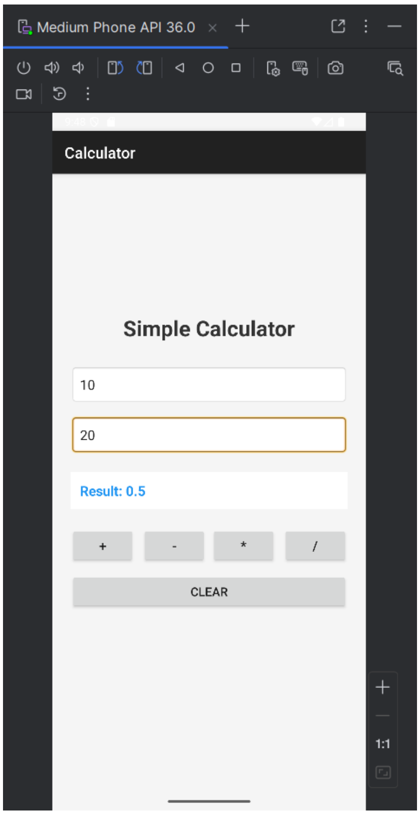


Simple Calculator

10

20

Result: 200.0



Simple Calculator

10

20

Result: 0.5

11. Create an Android application to demonstrate List View using an array adapter.

Mainactivity.java

```
package com.example.listviewadapter;

import android.os.Bundle;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    private ListView lvItems;
    private String[] items = {
            "Android",
            "iPhone",
            "Windows Mobile",
            "Blackberry",
            "WebOS",
            "Ubuntu",
            "Windows7",
            "Max OS X",
            "Linux",
            "iOS 4"
    };

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

        lvItems = findViewById(R.id.lvItems);

        ArrayAdapter<String> adapter = new ArrayAdapter<>(
                this,
                android.R.layout.simple_list_item_1,
                items
        );
        lvItems.setAdapter(adapter);

        lvItems.setOnItemClickListener((parent, view, position, id) -> {
            Toast.makeText(MainActivity.this,
                    "Selected: " + items[position],
                    Toast.LENGTH_SHORT).show();
        });

        lvItems.setOnItemLongClickListener((parent, view, position, id) -> {
            Toast.makeText(MainActivity.this,
                    "Long clicked: " + items[position],
                    Toast.LENGTH_SHORT).show();
            return true;
        });
    }
}
```

<u>activity_main.xml</u>

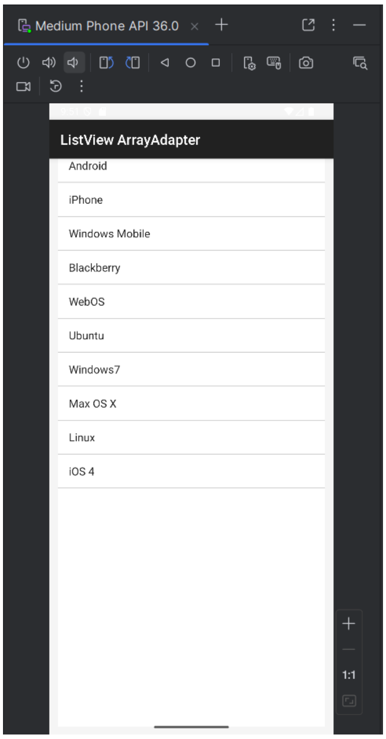
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="12dp"
    android:background="#f5f5f5">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Mobile Devices"
        android:textSize="22sp"
        android:textStyle="bold"
        android:textColor="#333333"
        android:gravity="center"
        android:layout_marginBottom="16dp"
        android:paddingTop="8dp"/>

    <ListView
```

```
        android:id="@+id/lvItems"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="#ffffff"
        android:divider="#cccccc"
        android:dividerHeight="1dp"/>

</LinearLayout>
```

AndroidManifest.xml

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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="ListView ArrayAdapter"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.AppCompat.Light.DarkActionBar">
        <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>
```

Output :



ListView ArrayAdapter

Android

iPhone

Windows Mobile

Blackberry

WebOS

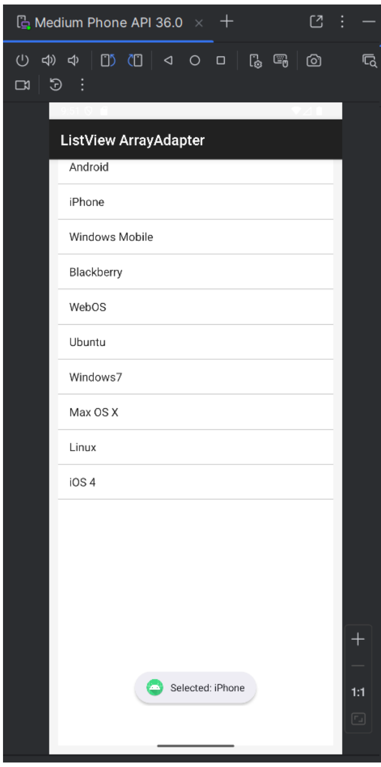
Ubuntu

Windows7

Max OS X

Linux

iOS 4



ListView ArrayAdapter

Android

iPhone

Windows Mobile

Blackberry

WebOS

Ubuntu

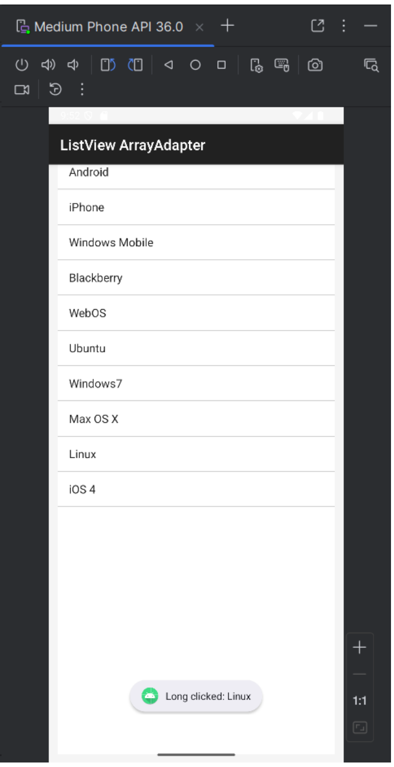
Windows7

Max OS X

Linux

iOS 4

Selected: iPhone



ListView ArrayAdapter

Android

iPhone

Windows Mobile

Blackberry

WebOS

Ubuntu

Windows7

Max OS X

Linux

iOS 4

Long clicked: Linux

12. Create a mobile application for a currency converter. Use a spinner for selecting the currency.

Mainactivity.java

```
package com.example.currencyconverter;

import android.os.Bundle;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    private EditText etAmount;
    private Spinner spnFromCurrency, spnToCurrency;
    private Button btnConvert, btnReset;
    private TextView tvResult;

    // Base currency: INR
    private double[] exchangeRates = {1.0, 83.5, 90.2, 0.56, 104.2};
    private String[] currencies = {"INR", "USD", "EUR", "JPY", "GBP"};

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

        etAmount = findViewById(R.id.etAmount);
        spnFromCurrency = findViewById(R.id.spnFromCurrency);
        spnToCurrency = findViewById(R.id.spnToCurrency);
        btnConvert = findViewById(R.id.btnConvert);
        btnReset = findViewById(R.id.btnReset);
        tvResult = findViewById(R.id.tvResult);

        ArrayAdapter<String> adapter = new ArrayAdapter<>(
            this,
            android.R.layout.simple_spinner_item,
            currencies
        );

adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_ite
m);

        spnFromCurrency.setAdapter(adapter);
        spnToCurrency.setAdapter(adapter);

        spnFromCurrency.setSelection(0);
        spnToCurrency.setSelection(1);

        btnConvert.setOnClickListener(v -> convertCurrency());
        btnReset.setOnClickListener(v -> resetFields());
    }

    private void convertCurrency() {
        String amountStr = etAmount.getText().toString().trim();

        if (amountStr.isEmpty()) {
            Toast.makeText(this, "Please enter amount", Toast.LENGTH_SHORT).show();
            return;
        }

        double amount = Double.parseDouble(amountStr);
        int fromIndex = spnFromCurrency.getSelectedItemPosition();
        int toIndex = spnToCurrency.getSelectedItemPosition();

        double amountInInr = amount * exchangeRates[fromIndex];
        double convertedAmount = amountInInr / exchangeRates[toIndex];

        String fromCurrency = currencies[fromIndex];
        String toCurrency = currencies[toIndex];

        tvResult.setText(String.format("%.2f %s = %.2f %s",
                amount, fromCurrency, convertedAmount, toCurrency));
    }

    private void resetFields() {
        etAmount.setText("");
        spnFromCurrency.setSelection(0);
        spnToCurrency.setSelection(1);
        tvResult.setText("Result: ");
        etAmount.requestFocus();
    }
}
```

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center"
    android:padding="24dp"
    android:background="#f5f5f5">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Currency Converter"
        android:textSize="28sp"
        android:textStyle="bold"
        android:textColor="#333333"
        android:layout_marginBottom="30dp"/>

    <EditText
        android:id="@+id/etAmount"
        android:layout_width="match_parent"
        android:layout_height="50dp"
        android:hint="Enter Amount"
        android:inputType="numberDecimal"
        android:background="@android:drawable/editbox_background"
        android:paddingLeft="12dp"
        android:paddingRight="12dp"
        android:layout_marginBottom="20dp"/>

    <TextView
```

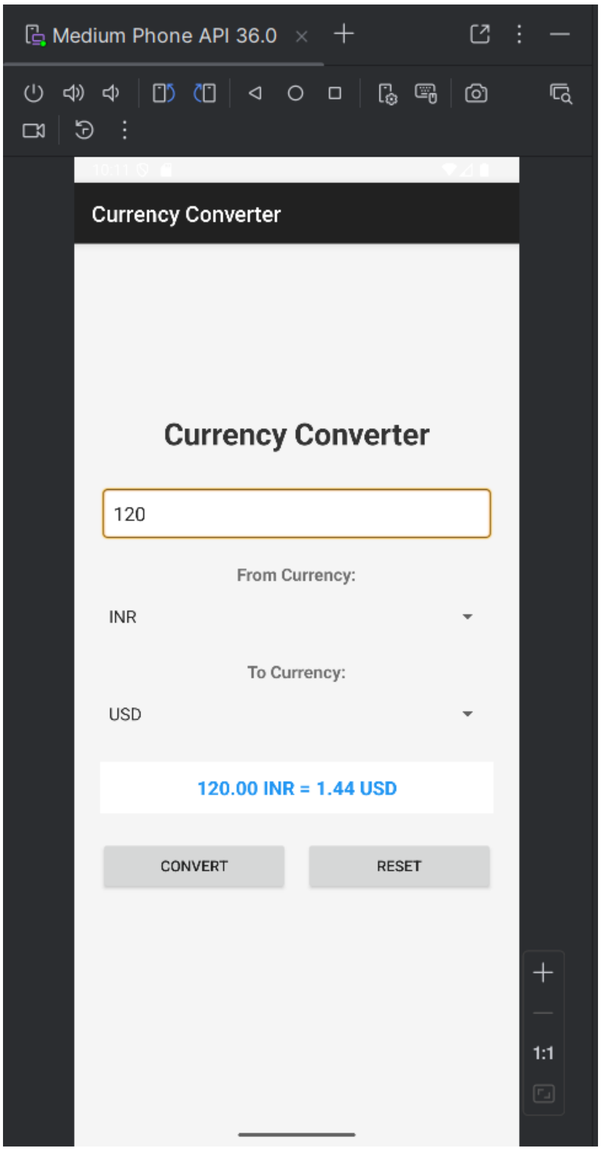
```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="From Currency:"
        android:textSize="16sp"
        android:textStyle="bold"
        android:layout_marginBottom="8dp"/>

    <Spinner
        android:id="@+id/spnFromCurrency"
        android:layout_width="match_parent"
        android:layout_height="40dp"
        android:layout_marginBottom="20dp"/>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="To Currency:"
        android:textSize="16sp"
        android:textStyle="bold"
        android:layout_marginBottom="8dp"/>

    <Spinner
        android:id="@+id/spnToCurrency"
        android:layout_width="match_parent"
        android:layout_height="40dp"
        android:layout_marginBottom="24dp"/>

    <TextView
        android:id="@+id/tvResult"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Result: "
        android:textSize="18sp"
        android:textStyle="bold"
        android:textColor="#2196F3"
        android:padding="12dp"
        android:layout_marginBottom="24dp"
        android:background="#ffffff"
        android:gravity="center"/>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

        <Button
            android:id="@+id/btnConvert"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="Convert"
            android:layout_marginRight="8dp"/>

        <Button
            android:id="@+id/btnReset"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="Reset"
            android:layout_marginLeft="8dp"/>

    </LinearLayout>

</LinearLayout>
```

AndroidManifest.xml

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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="Currency Converter"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.AppCompat.Light.DarkActionBar">
        <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>
```

Output :



Currency Converter

120

From Currency:

INR

To Currency:

USD

120.00 INR = 1.44 USD

13. Write an application to increase font size using seekbar.

Mainactivity.java

```
package com.example.seekbarfontsize;
import android.os.Bundle;
import android.widget.SeekBar;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    private SeekBar seekBar;
    private TextView tvText, tvFontSize;

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

        seekBar = findViewById(R.id.seekBar);
        tvText = findViewById(R.id.tvText);
        tvFontSize = findViewById(R.id.tvFontSize);

        seekBar.setMax(50);
        seekBar.setProgress(20);

        tvText.setTextSize(20);
        tvFontSize.setText("Font Size: 20sp");

        seekBar.setOnSeekBarChangeListener(new
SeekBar.OnSeekBarChangeListener() {
            @Override
            public void onProgressChanged(SeekBar seekBar, int progress, boolean
fromUser) {
                int fontSize = 10 + progress;
                tvText.setTextSize(fontSize);
                tvFontSize.setText("Font Size: " + fontSize + "sp");
            }

            @Override
            public void onStartTrackingTouch(SeekBar seekBar) {
            }

            @Override
            public void onStopTrackingTouch(SeekBar seekBar) {
            }
        });
    }
}
```

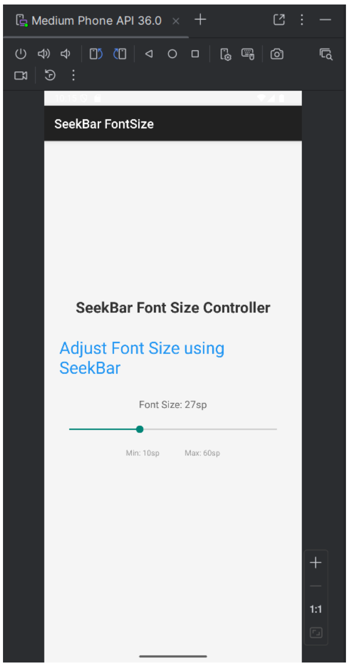
activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center"
    android:padding="24dp"
    android:background="#f5f5f5">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="SeekBar Font Size Controller"
        android:textSize="24sp"
        android:textStyle="bold"
        android:textColor="#333333"
        android:layout_marginBottom="30dp"/>

    <TextView
        android:id="@+id/tvText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Adjust Font Size using SeekBar"
        android:textSize="20sp"
        android:textColor="#2196F3"
        android:layout_marginBottom="30dp"/>

    <TextView
        android:id="@+id/tvFontSize"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Font Size: 20sp"
        android:textSize="16sp"
        android:textColor="#666666"
        android:layout_marginBottom="20dp"/>

    <SeekBar
        android:id="@+id/seekBar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="20dp"/>

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

        <TextView
          android:layout_width="wrap_content"
          android:layout_height="wrap_content"
          android:text="Min: 10sp"
          android:textSize="12sp"
          android:textColor="#999999"
          android:layout_marginRight="20dp"/>

        <TextView
          android:layout_width="wrap_content"
          android:layout_height="wrap_content"
          android:text="Max: 60sp"
          android:textSize="12sp"
          android:textColor="#999999"
          android:layout_marginLeft="20dp"/>

    </LinearLayout>

</LinearLayout>
```

androidManifest.xml

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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="SeekBar FontSize"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.AppCompat.Light.DarkActionBar">
        <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>
```

Output :



SeekBar FontSize

SeekBar Font Size Controller

Adjust Font Size using SeekBar

Font Size: 27sp

Min: 10sp Max: 60sp

14. Create an Android application to demonstrate progressbar.

Mainactivity.java

```
package com.example.progressbar;

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

public class MainActivity extends AppCompatActivity {

   private ProgressBar pbHorizontal, pbCircular;
   private Button btnStart, btnReset;
   private TextView tvProgress;
   private int progress = 0;

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

      pbHorizontal = findViewById(R.id.pbHorizontal);
      pbCircular = findViewById(R.id.pbCircular);
      btnStart = findViewById(R.id.btnStart);
      btnReset = findViewById(R.id.btnReset);
      tvProgress = findViewById(R.id.tvProgress);

      pbHorizontal.setMax(100);
      pbCircular.setMax(100);

      btnStart.setOnClickListener(v -> startProgress());
      btnReset.setOnClickListener(v -> resetProgress());
   }

   private void startProgress() {
      btnStart.setEnabled(false);
      new Thread(() -> {
            for (int i = 0; i <= 100; i++) {
                progress = i;
                pbHorizontal.setProgress(i);
                pbCircular.setProgress(i);
                tvProgress.post(() -> tvProgress.setText("Progress: " + progress + "%"));
                try {
                    Thread.sleep(100);
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }
            btnStart.post(() -> btnStart.setEnabled(true));
        }).start();
    }

    private void resetProgress() {
        progress = 0;
        pbHorizontal.setProgress(0);
        pbCircular.setProgress(0);
        tvProgress.setText("Progress: 0%");
        btnStart.setEnabled(true);
    }
}
```

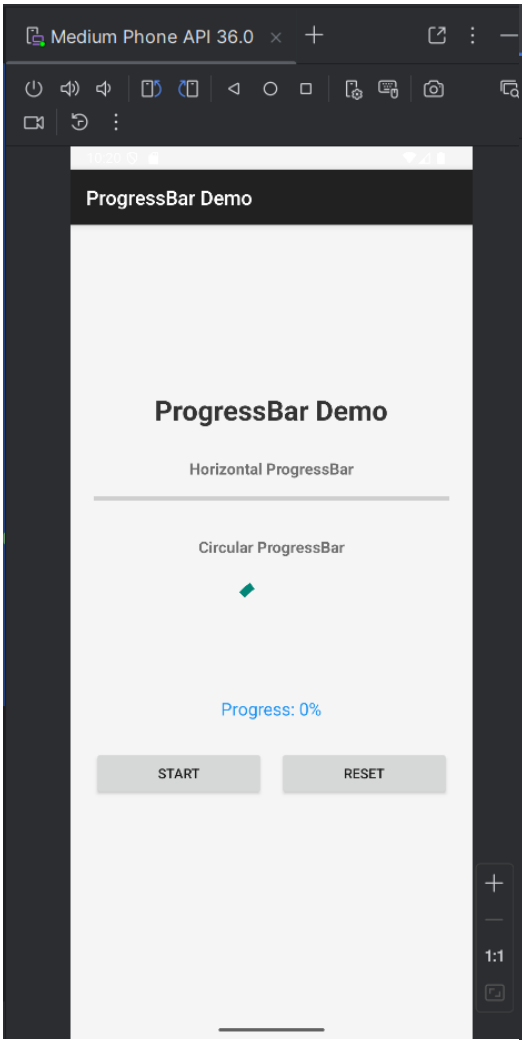
activity_main.xml

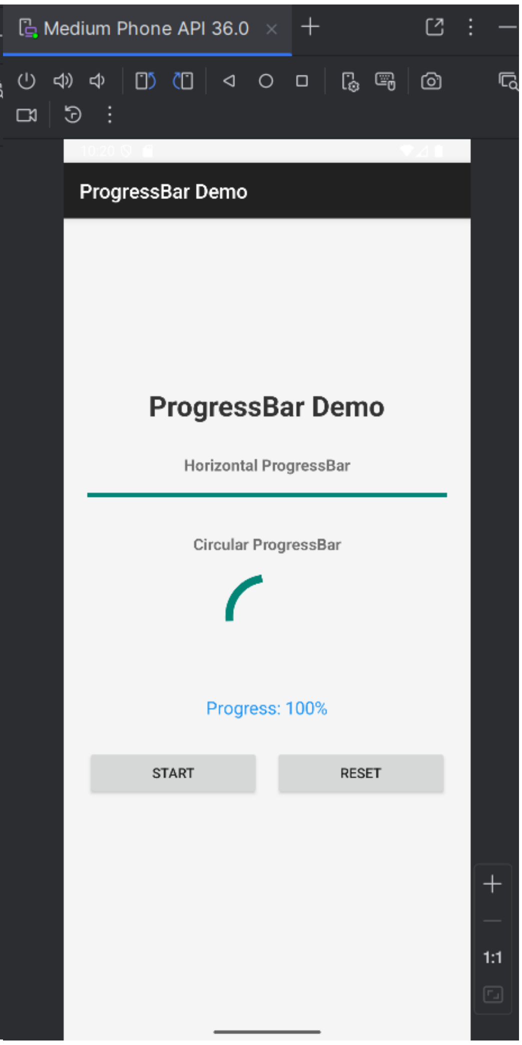
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center"
    android:padding="24dp"
    android:background="#f5f5f5">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="ProgressBar Demo"
        android:textSize="28sp"
        android:textStyle="bold"
        android:textColor="#333333"
        android:layout_marginBottom="30dp"/>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Horizontal ProgressBar"
        android:textSize="16sp"
        android:textStyle="bold"
        android:layout_marginBottom="12dp"/>

    <ProgressBar
        android:id="@+id/pbHorizontal"
        style="?android:attr/progressBarStyleHorizontal"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="30dp"/>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Circular ProgressBar"
        android:textSize="16sp"
        android:textStyle="bold"
        android:layout_marginBottom="12dp"/>

    <ProgressBar
        android:id="@+id/pbCircular"
        style="?android:attr/progressBarStyleLarge"
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:layout_gravity="center"
        android:layout_marginBottom="30dp"/>

    <TextView
        android:id="@+id/tvProgress"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Progress: 0%"
        android:textSize="18sp"
        android:textColor="#2196F3"
        android:layout_marginBottom="30dp"/>

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

        <Button
            android:id="@+id/btnStart"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="Start"
            android:layout_marginRight="8dp"/>

        <Button
            android:id="@+id/btnReset"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="Reset"
            android:layout_marginLeft="8dp"/>

    </LinearLayout>

</LinearLayout>
```

Androidmanifest.xml

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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="ProgressBar Demo"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.AppCompat.Light.DarkActionBar">
        <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>
```

Otuput :





Experiment No – 2 Database Connectivity

1. Create an Android application to read and write content in internal storage.

Mainactivity.java

```
package com.example.internalstorage;

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

import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;

public class MainActivity extends AppCompatActivity {

    private EditText etInput, etOutput;
    private Button btnSave, btnLoad;
    private static final String FILE_NAME = "samplefile.txt";

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

        etInput = findViewById(R.id.etInput);
        etOutput = findViewById(R.id.etOutput);
        btnSave = findViewById(R.id.btnSave);
        btnLoad = findViewById(R.id.btnLoad);

        btnSave.setOnClickListener(v -> saveToFile());
        btnLoad.setOnClickListener(v -> loadFromFile());
```

```
  }

  private void saveToFile() {
     String data = etInput.getText().toString();
     if (data.isEmpty()) {
        Toast.makeText(this, "Please enter data to save",
Toast.LENGTH_SHORT).show();
        return;
     }
     try (FileOutputStream fos = openFileOutput(FILE_NAME, MODE_PRIVATE)) {
        fos.write(data.getBytes());
        Toast.makeText(this, "Data saved to internal storage",
Toast.LENGTH_SHORT).show();
        etInput.setText("");
     } catch (IOException e) {
        e.printStackTrace();
        Toast.makeText(this, "Error saving data", Toast.LENGTH_SHORT).show();
     }
  }

  private void loadFromFile() {
     try (FileInputStream fis = openFileInput(FILE_NAME)) {
        byte[] buffer = new byte[fis.available()];
        fis.read(buffer);
        String data = new String(buffer);
        etOutput.setText(data);
        Toast.makeText(this, "Data loaded from internal storage",
Toast.LENGTH_SHORT).show();
     } catch (IOException e) {
        e.printStackTrace();
        Toast.makeText(this, "Error loading data", Toast.LENGTH_SHORT).show();
     }
  }
}
```

activity_main.xml

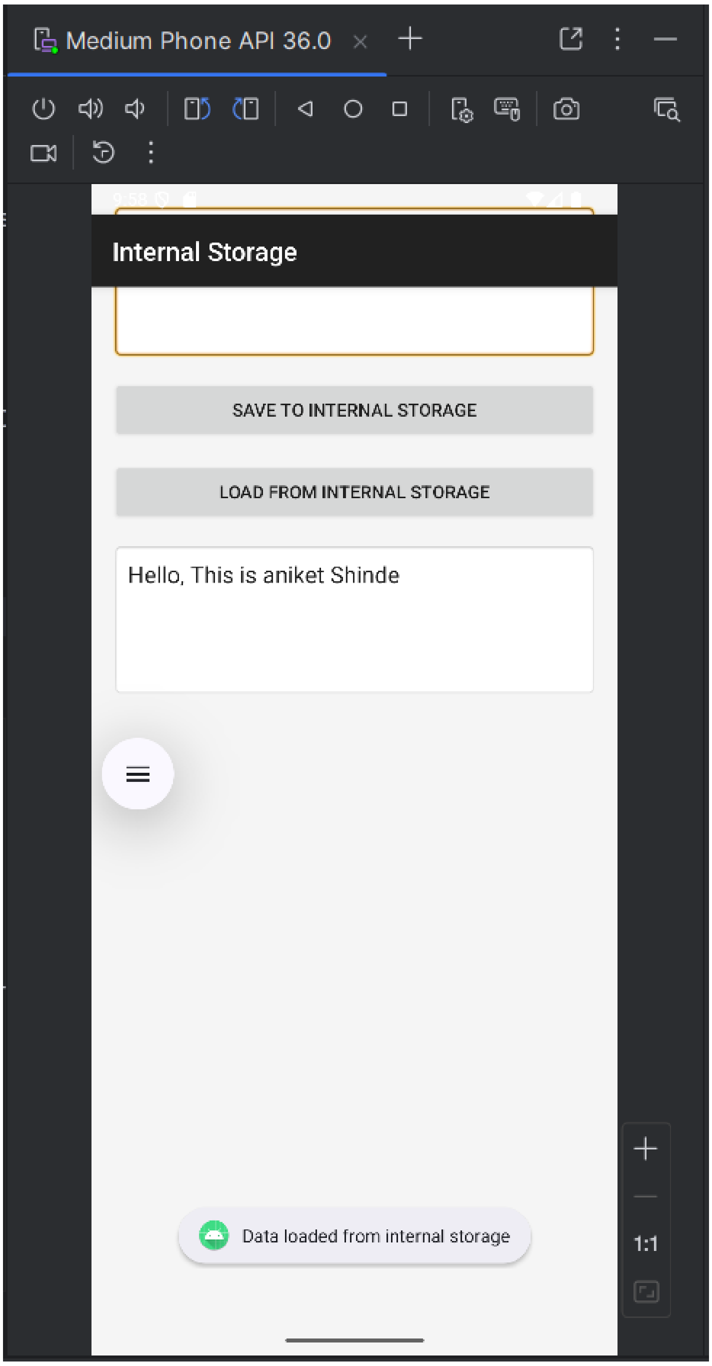
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
  android:padding="16dp"
```

```
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#f5f5f5">

    <EditText
        android:id="@+id/etInput"
        android:layout_width="match_parent"
        android:layout_height="120dp"
        android:hint="Enter text to save"
        android:background="@android:drawable/editbox_background"
        android:padding="12dp"
        android:gravity="top"
        android:inputType="textMultiLine"
        android:layout_marginBottom="16dp"/>

    <Button
        android:id="@+id/btnSave"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Save to Internal Storage"
        android:layout_marginBottom="16dp"/>

    <Button
        android:id="@+id/btnLoad"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Load from Internal Storage"
        android:layout_marginBottom="16dp"/>

    <EditText
        android:id="@+id/etOutput"
        android:layout_width="match_parent"
        android:layout_height="120dp"
        android:hint="Loaded text appears here"
        android:background="@android:drawable/editbox_background"
        android:padding="12dp"
        android:gravity="top"
        android:inputType="textMultiLine"
        android:editable="false"
    android:focusable="false"/>

</LinearLayout>
```

AndroidManifest.xml

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

  <application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="Internal Storage"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/Theme.AppCompat.Light.DarkActionBar">
    <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>
```

Output :



Hello, This is aniket Shinde

Data loaded from internal storage

2. Create an Android application to read and write content in external storage.

Mainactivity.java

```
package com.example.externalstorage;

import android.Manifest;
import android.content.pm.PackageManager;
import android.os.Bundle;
import android.os.Environment;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

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

public class MainActivity extends AppCompatActivity {

    private static final int PERMISSION_REQUEST_CODE = 100;

    private EditText etInput, etOutput;
    private Button btnSave, btnLoad;

    private File externalFile;

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

        etInput = findViewById(R.id.etInput);
        etOutput = findViewById(R.id.etOutput);
        btnSave = findViewById(R.id.btnSave);
        btnLoad = findViewById(R.id.btnLoad);

        btnSave.setOnClickListener(v -> {
            if (checkPermission()) {
                saveToFile();
            } else {
                requestPermission();
            }
        });

        btnLoad.setOnClickListener(v -> {
            if (checkPermission()) {
                loadFromFile();
            } else {
                requestPermission();
            }
        });

        externalFile = new
File(getExternalFilesDir(Environment.DIRECTORY_DOCUMENTS), "samplefile.txt");
    }

    private boolean checkPermission() {
        return ContextCompat.checkSelfPermission(this,
                Manifest.permission.READ_EXTERNAL_STORAGE) ==
PackageManager.PERMISSION_GRANTED &&
                ContextCompat.checkSelfPermission(this,
                        Manifest.permission.WRITE_EXTERNAL_STORAGE) ==
PackageManager.PERMISSION_GRANTED;
    }

    private void requestPermission() {
        ActivityCompat.requestPermissions(this,
                new String[]{
                        Manifest.permission.READ_EXTERNAL_STORAGE,
                        Manifest.permission.WRITE_EXTERNAL_STORAGE
                },
                PERMISSION_REQUEST_CODE);
```

```
  }

  @Override
  public void onRequestPermissionsResult(int requestCode,
                                   @NonNull String[] permissions,
                                   @NonNull int[] grantResults) {
    if (requestCode == PERMISSION_REQUEST_CODE) {
      if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED
            && grantResults[1] == PackageManager.PERMISSION_GRANTED) {
        Toast.makeText(this, "Permission Granted. Try again.",
Toast.LENGTH_SHORT).show();
      } else {
        Toast.makeText(this, "Permission Denied.",
Toast.LENGTH_SHORT).show();
      }
    }
    super.onRequestPermissionsResult(requestCode, permissions, grantResults);
  }

  private void saveToFile() {
    String data = etInput.getText().toString();
    if (data.isEmpty()) {
      Toast.makeText(this, "Please enter data to save",
Toast.LENGTH_SHORT).show();
      return;
    }
    try (FileOutputStream fos = new FileOutputStream(externalFile)) {
      fos.write(data.getBytes());
      Toast.makeText(this, "Data saved to external storage",
Toast.LENGTH_SHORT).show();
      etInput.setText("");
    } catch (IOException e) {
      e.printStackTrace();
      Toast.makeText(this, "Error saving data", Toast.LENGTH_SHORT).show();
    }
  }

  private void loadFromFile() {
    if (!externalFile.exists()) {
      Toast.makeText(this, "No file found to load", Toast.LENGTH_SHORT).show();
      return;
    }
    try (FileInputStream fis = new FileInputStream(externalFile)) {
      byte[] buffer = new byte[(int) externalFile.length()];
      fis.read(buffer);
      String data = new String(buffer);
      etOutput.setText(data);
      Toast.makeText(this, "Data loaded from external storage",
Toast.LENGTH_SHORT).show();
    } catch (IOException e) {
      e.printStackTrace();
      Toast.makeText(this, "Error loading data", Toast.LENGTH_SHORT).show();
    }
  }
}
```

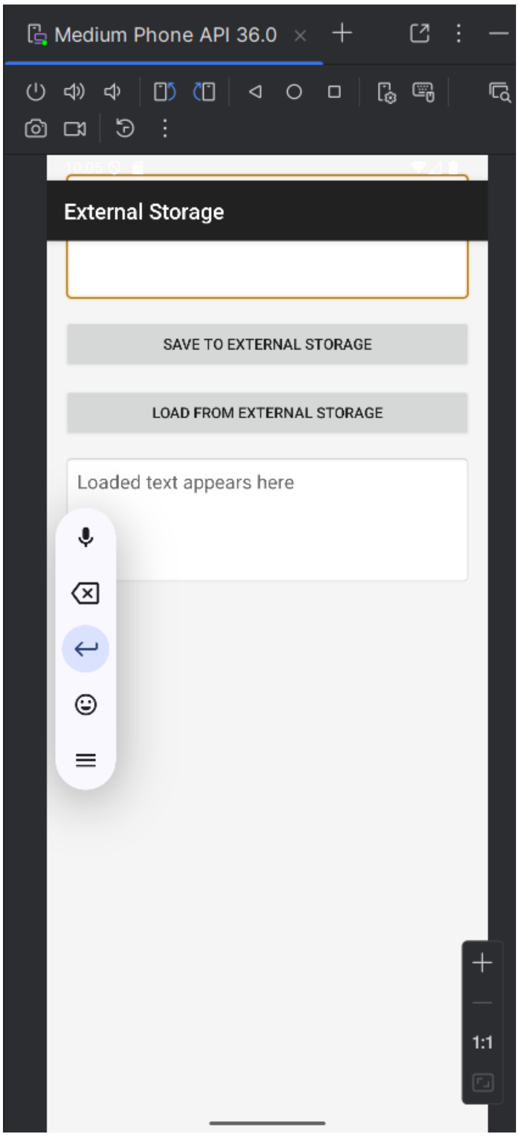
activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:padding="16dp"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#f5f5f5">

    <EditText
        android:id="@+id/etInput"
        android:layout_width="match_parent"
        android:layout_height="120dp"
        android:hint="Enter text to save"
        android:background="@android:drawable/editbox_background"
        android:padding="12dp"
        android:gravity="top"
        android:inputType="textMultiLine"
        android:layout_marginBottom="16dp"/>

    <Button
        android:id="@+id/btnSave"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Save to External Storage"
        android:layout_marginBottom="16dp"/>

    <Button
        android:id="@+id/btnLoad"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Load from External Storage"
        android:layout_marginBottom="16dp"/>

    <EditText
        android:id="@+id/etOutput"
        android:layout_width="match_parent"
        android:layout_height="120dp"
        android:hint="Loaded text appears here"
        android:background="@android:drawable/editbox_background"
        android:padding="12dp"
        android:gravity="top"
        android:inputType="textMultiLine"
        android:editable="false"
        android:focusable="false"/>

</LinearLayout>
```

AndroidManifest.xml

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

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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="External Storage"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.AppCompat.Light.DarkActionBar">
        <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>
```

Output :



External Storage

SAVE TO EXTERNAL STORAGE

LOAD FROM EXTERNAL STORAGE

Loaded text appears here

3. Write an android program for shared preference to store value in name-value pairs.

Mainactivity.java

```
package com.example.sharedpreferencesdemo;

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

public class MainActivity extends AppCompatActivity {

    private EditText etName, etEmail;
    private Button btnSave, btnLoad, btnClear;

    private SharedPreferences sharedPreferences;
    private static final String PREF_NAME = "MyPrefs";

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

        etName = findViewById(R.id.etName);
        etEmail = findViewById(R.id.etEmail);
        btnSave = findViewById(R.id.btnSave);
        btnLoad = findViewById(R.id.btnLoad);
        btnClear = findViewById(R.id.btnClear);

        sharedPreferences = getSharedPreferences(PREF_NAME, MODE_PRIVATE);

        btnSave.setOnClickListener(v -> saveData());
        btnLoad.setOnClickListener(v -> loadData());
        btnClear.setOnClickListener(v -> clearData());
    }
```

```
  private void saveData() {
    String name = etName.getText().toString();
    String email = etEmail.getText().toString();

    if (name.isEmpty() || email.isEmpty()) {
      Toast.makeText(this, "Please fill both fields", Toast.LENGTH_SHORT).show();
      return;
    }

    SharedPreferences.Editor editor = sharedPreferences.edit();
    editor.putString("name", name);
    editor.putString("email", email);
    editor.apply();

    Toast.makeText(this, "Data saved to SharedPreferences",
Toast.LENGTH_SHORT).show();
  }

  private void loadData() {
    String name = sharedPreferences.getString("name", "");
    String email = sharedPreferences.getString("email", "");

    if (name.isEmpty() || email.isEmpty()) {
      Toast.makeText(this, "No data found. Save some data first",
Toast.LENGTH_SHORT).show();
      return;
    }

    etName.setText(name);
    etEmail.setText(email);
    Toast.makeText(this, "Data loaded from SharedPreferences",
Toast.LENGTH_SHORT).show();
  }

  private void clearData() {
    etName.setText("");
    etEmail.setText("");
    SharedPreferences.Editor editor = sharedPreferences.edit();
    editor.clear();
    editor.apply();
```

```
        Toast.makeText(this, "Data cleared", Toast.LENGTH_SHORT).show();
    }
}
```

activity_main.xml

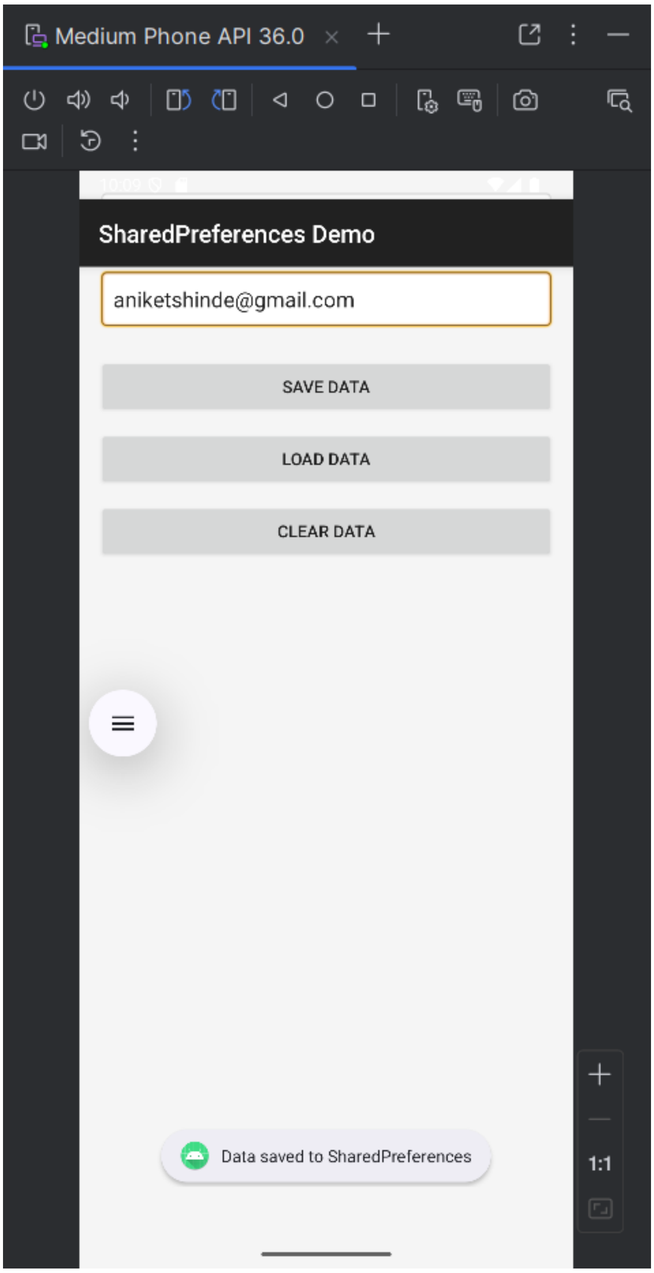
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:padding="16dp"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#f5f5f5">

    <EditText
        android:id="@+id/etName"
        android:layout_width="match_parent"
        android:layout_height="50dp"
        android:hint="Enter Name"
        android:background="@android:drawable/editbox_background"
        android:paddingLeft="12dp"
        android:paddingRight="12dp"
        android:layout_marginBottom="16dp"/>

    <EditText
        android:id="@+id/etEmail"
        android:layout_width="match_parent"
        android:layout_height="50dp"
        android:hint="Enter Email"
        android:background="@android:drawable/editbox_background"
        android:paddingLeft="12dp"
        android:paddingRight="12dp"
        android:inputType="textEmailAddress"
        android:layout_marginBottom="24dp"/>

    <Button
        android:id="@+id/btnSave"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Save Data"
        android:layout_marginBottom="12dp"/>
```

```
  <Button
    android:id="@+id/btnLoad"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Load Data"
    android:layout_marginBottom="12dp"/>

  <Button
    android:id="@+id/btnClear"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Clear Data"/>
</LinearLayout>
```

AndroidManifest.xml

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

  <application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="SharedPreferences Demo"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/Theme.AppCompat.Light.DarkActionBar">
    <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>
```

Output:



SharedPreferences Demo

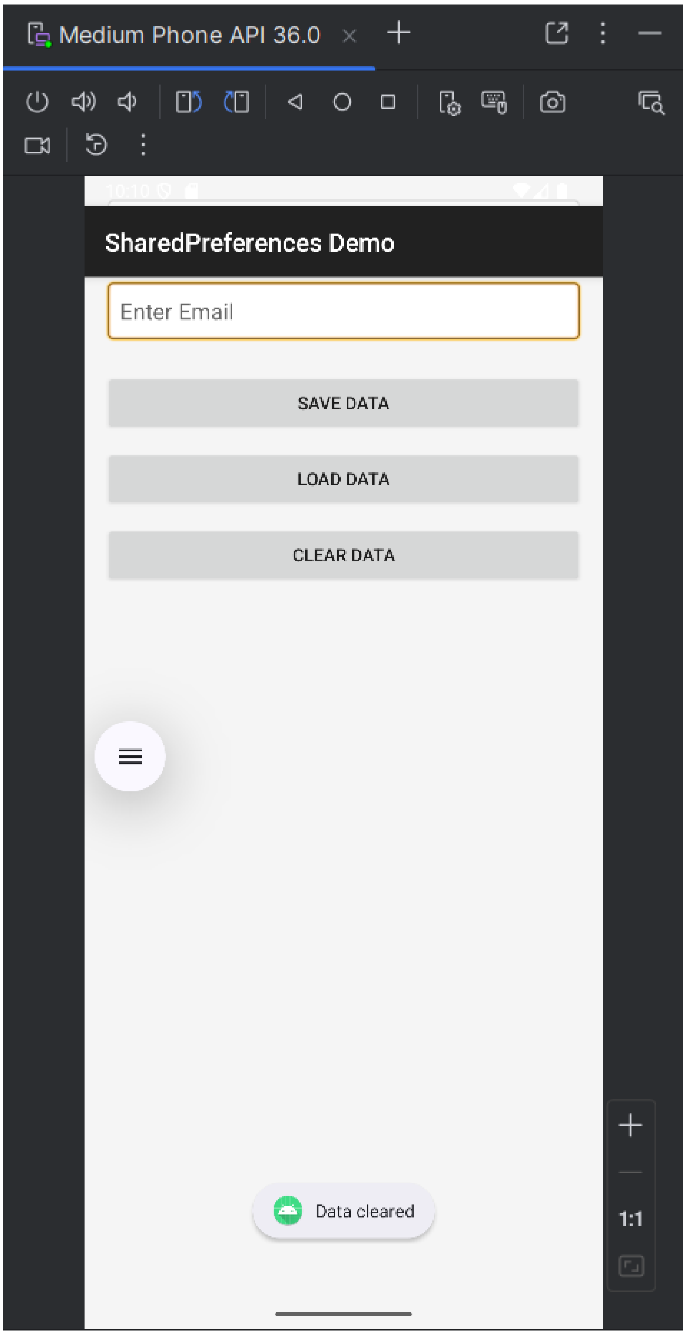
aniketshinde@gmail.com

SAVE DATA

LOAD DATA

CLEAR DATA

Data saved to SharedPreferences


Medium Phone API 36.0
SharedPreferences Demo
Enter Email
SAVE DATA
LOAD DATA
CLEAR DATA
Data cleared
1:1

4. Create a login form with a remember me checkbox. Save the username and password if the checkbox is checked using shared preference and show the welcome page when the login button is clicked.

Mainacitivty.java

```
package com.example.remembermelogin;

import android.content.Context;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.EditText;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    private EditText etUsername, etPassword;
    private CheckBox cbRememberMe;
    private Button btnLogin;

    private SharedPreferences sharedPreferences;
    private static final String PREF_NAME = "LoginPrefs";

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

        etUsername = findViewById(R.id.etUsername);
        etPassword = findViewById(R.id.etPassword);
        cbRememberMe = findViewById(R.id.cbRememberMe);
        btnLogin = findViewById(R.id.btnLogin);

        sharedPreferences = getSharedPreferences(PREF_NAME,
Context.MODE_PRIVATE);

        loadSavedCredentials();
        btnLogin.setOnClickListener(v -> attemptLogin());
    }

    private void loadSavedCredentials() {
        boolean isRemembered = sharedPreferences.getBoolean("rememberMe",
false);
        if (isRemembered) {
            String savedUsername = sharedPreferences.getString("username", "");
            String savedPassword = sharedPreferences.getString("password", "");
            etUsername.setText(savedUsername);
            etPassword.setText(savedPassword);
            cbRememberMe.setChecked(true);
        }
    }

    private void attemptLogin() {
        String username = etUsername.getText().toString();
        String password = etPassword.getText().toString();

        if (username.isEmpty() || password.isEmpty()) {
            Toast.makeText(this, "Please enter username and password",
Toast.LENGTH_SHORT).show();
            return;
        }

        // Assume login success for demonstration
        Toast.makeText(this, "Login Successful", Toast.LENGTH_SHORT).show();

        if (cbRememberMe.isChecked()) {
            SharedPreferences.Editor editor = sharedPreferences.edit();
            editor.putString("username", username);
            editor.putString("password", password);
            editor.putBoolean("rememberMe", true);
            editor.apply();
        } else {
            SharedPreferences.Editor editor = sharedPreferences.edit();
            editor.clear();
            editor.apply();
        }
    }
}
```

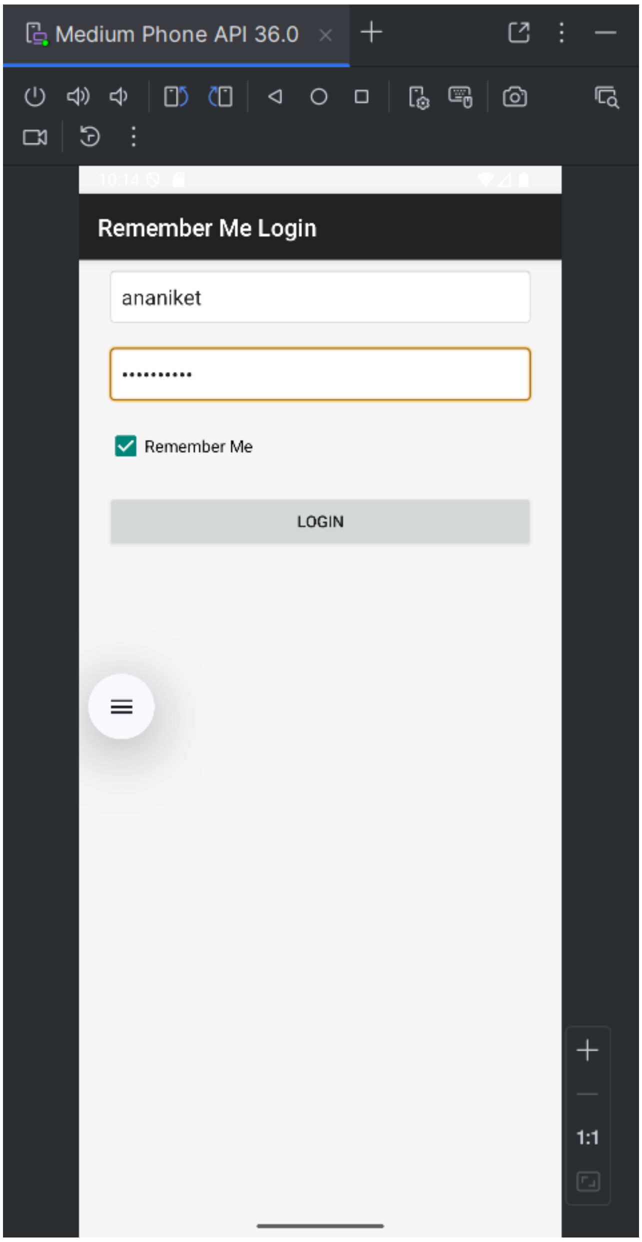
Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:padding="24dp"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#f5f5f5">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Login Form"
        android:textSize="24sp"
        android:textStyle="bold"
        android:textColor="#333333"
        android:gravity="center"
        android:layout_marginBottom="30dp"/>

    <EditText
        android:id="@+id/etUsername"
        android:layout_width="match_parent"
        android:layout_height="50dp"
        android:hint="Username"
        android:background="@android:drawable/editbox_background"
        android:paddingLeft="12dp"
        android:paddingRight="12dp"
        android:layout_marginBottom="16dp"/>

    <EditText
        android:id="@+id/etPassword"
        android:layout_width="match_parent"
        android:layout_height="50dp"
        android:hint="Password"
        android:inputType="textPassword"
        android:background="@android:drawable/editbox_background"
        android:paddingLeft="12dp"
        android:paddingRight="12dp"
        android:layout_marginBottom="20dp"/>

    <CheckBox
        android:id="@+id/cbRememberMe"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Remember Me"
        android:layout_marginBottom="24dp"/>

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

</LinearLayout>
```

Androidmanifest.xml

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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="Remember Me Login"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.AppCompat.Light.DarkActionBar">
        <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>
```

Output:



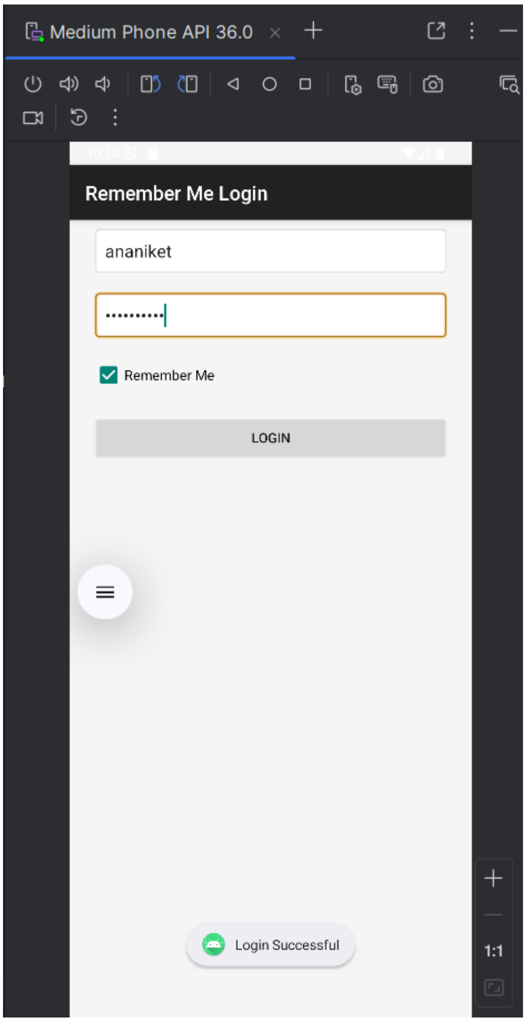
Remember Me Login

ananiket

••••••••••

☑ Remember Me

LOGIN



Remember Me Login

ananiket

••••••••••

☑ Remember Me

LOGIN

Login Successful

5. Create an Android application to insert, update, select, and delete records from the student table using SQLite Database.

MainActivity.java

```
package com.example.sqlitecrud;
import android.database.Cursor;
import android.os.Bundle;
import android.widget.*;
import androidx.appcompat.app.AppCompatActivity;
```

```
public class MainActivity extends AppCompatActivity {

    private EditText etId, etName, etEmail;
    private Button btnInsert, btnUpdate, btnDelete, btnViewAll;
    private TextView tvRecords;
    private DBHelper dbHelper;

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

        etId = findViewById(R.id.etId);
        etName = findViewById(R.id.etName);
        etEmail = findViewById(R.id.etEmail);
        btnInsert = findViewById(R.id.btnInsert);
        btnUpdate = findViewById(R.id.btnUpdate);
        btnDelete = findViewById(R.id.btnDelete);
        btnViewAll = findViewById(R.id.btnViewAll);
        tvRecords = findViewById(R.id.tvRecords);

        dbHelper = new DBHelper(this);

        btnInsert.setOnClickListener(v -> insertData());
        btnUpdate.setOnClickListener(v -> updateData());
        btnDelete.setOnClickListener(v -> deleteData());
        btnViewAll.setOnClickListener(v -> viewAllData());
    }

    private void insertData() {
        String name = etName.getText().toString();
        String email = etEmail.getText().toString();
        if (name.isEmpty() || email.isEmpty()) {
            Toast.makeText(this, "Please enter name and email",
Toast.LENGTH_SHORT).show();
            return;
        }
        boolean inserted = dbHelper.insertStudent(name, email);
        Toast.makeText(this, inserted ? "Data Inserted" : "Insert Failed",
Toast.LENGTH_SHORT).show();
   }

   private void updateData() {
      String id = etId.getText().toString();
      String name = etName.getText().toString();
      String email = etEmail.getText().toString();
      if (id.isEmpty() || name.isEmpty() || email.isEmpty()) {
         Toast.makeText(this, "Please enter id, name and email",
Toast.LENGTH_SHORT).show();
         return;
      }
      boolean updated = dbHelper.updateStudent(id, name, email);
      Toast.makeText(this, updated ? "Data Updated" : "Update Failed",
Toast.LENGTH_SHORT).show();
   }

   private void deleteData() {
      String id = etId.getText().toString();
      if (id.isEmpty()) {
         Toast.makeText(this, "Please enter id", Toast.LENGTH_SHORT).show();
         return;
      }
      boolean deleted = dbHelper.deleteStudent(id);
      Toast.makeText(this, deleted ? "Data Deleted" : "Delete Failed",
Toast.LENGTH_SHORT).show();
   }

   private void viewAllData() {
      Cursor cursor = dbHelper.getAllStudents();
      if (cursor.getCount() == 0) {
         tvRecords.setText("No records found");
         return;
      }
      StringBuilder sb = new StringBuilder();
      while (cursor.moveToNext()) {
         sb.append("ID: ").append(cursor.getInt(0)).append("\n");
         sb.append("Name: ").append(cursor.getString(1)).append("\n");
         sb.append("Email: ").append(cursor.getString(2)).append("\n\n");
```

```
        }
        tvRecords.setText(sb.toString());
    }
}
```

DBHelper.java

```
package com.example.sqlitecrud;

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

public class DBHelper extends SQLiteOpenHelper {

    public static final String DB_NAME = "StudentDB.db";
    public static final int DB_VERSION = 1;

    public static final String TABLE_NAME = "students";
    public static final String COL_ID = "id";
    public static final String COL_NAME = "name";
    public static final String COL_EMAIL = "email";

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

    @Override
    public void onCreate(SQLiteDatabase db) {
        String createTable = "CREATE TABLE " + TABLE_NAME +
                " (" + COL_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, " +
                COL_NAME + " TEXT, " +
                COL_EMAIL + " TEXT)";
        db.execSQL(createTable);
    }

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

    public boolean insertStudent(String name, String email) {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues cv = new ContentValues();
        cv.put(COL_NAME, name);
        cv.put(COL_EMAIL, email);
        long result = db.insert(TABLE_NAME, null, cv);
        return result != -1;
    }

    public Cursor getAllStudents() {
        SQLiteDatabase db = this.getReadableDatabase();
        return db.rawQuery("SELECT * FROM " + TABLE_NAME, null);
    }

    public boolean updateStudent(String id, String name, String email) {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues cv = new ContentValues();
        cv.put(COL_NAME, name);
        cv.put(COL_EMAIL, email);
        int result = db.update(TABLE_NAME, cv, COL_ID + "=?", new String[]{id});
        return result > 0;
    }

    public boolean deleteStudent(String id) {
        SQLiteDatabase db = this.getWritableDatabase();
        int result = db.delete(TABLE_NAME, COL_ID + "=?", new String[]{id});
        return result > 0;
    }
}
```

Activity_main.xml

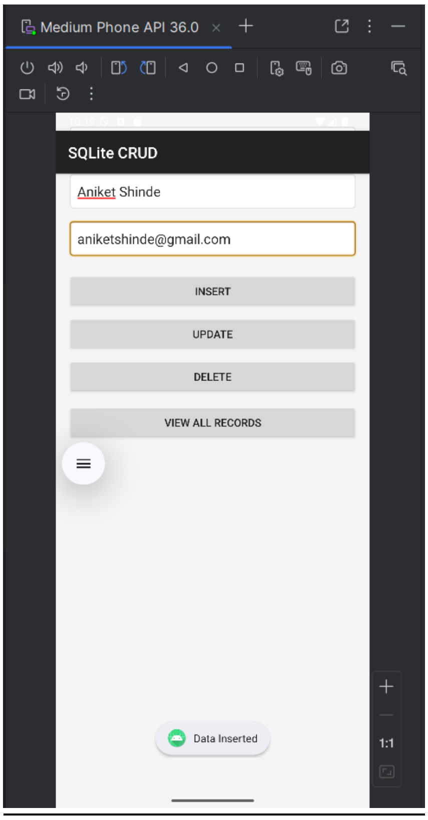
```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
```

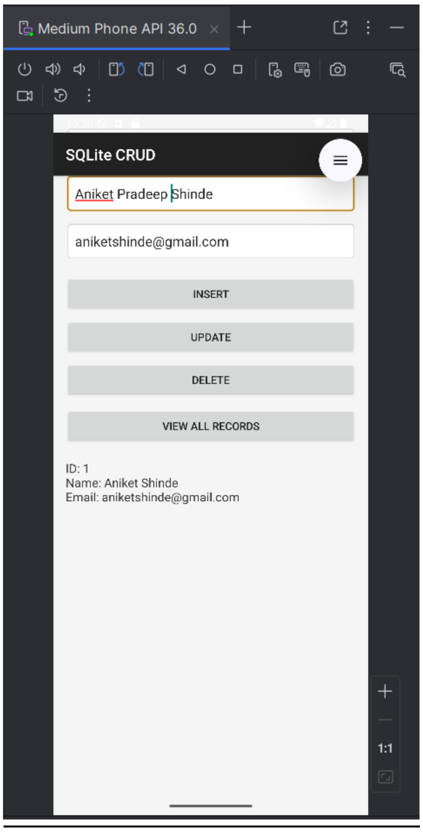
```
android:layout_height="match_parent"
android:padding="16dp"
android:background="#f5f5f5">

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

    <EditText
        android:id="@+id/etId"
        android:layout_width="match_parent"
        android:layout_height="50dp"
        android:hint="Enter ID for Update/Delete"
        android:inputType="number"
        android:background="@android:drawable/editbox_background"
        android:paddingLeft="12dp"
        android:paddingRight="12dp"
        android:layout_marginBottom="12dp"/>

    <EditText
        android:id="@+id/etName"
        android:layout_width="match_parent"
        android:layout_height="50dp"
        android:hint="Enter Name"
        android:background="@android:drawable/editbox_background"
        android:paddingLeft="12dp"
        android:paddingRight="12dp"
        android:layout_marginBottom="12dp"/>

    <EditText
        android:id="@+id/etEmail"
        android:layout_width="match_parent"
        android:layout_height="50dp"
        android:hint="Enter Email"
        android:inputType="textEmailAddress"
        android:background="@android:drawable/editbox_background"
        android:paddingLeft="12dp"
        android:paddingRight="12dp"
        android:layout_marginBottom="20dp"/>
    <Button
        android:id="@+id/btnInsert"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Insert"
        android:layout_marginBottom="8dp"/>

    <Button
        android:id="@+id/btnUpdate"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Update"
        android:layout_marginBottom="8dp"/>

    <Button
        android:id="@+id/btnDelete"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Delete"
        android:layout_marginBottom="12dp"/>

    <Button
        android:id="@+id/btnViewAll"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="View All Records"
        android:layout_marginBottom="20dp"/>

    <TextView
        android:id="@+id/tvRecords"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="16sp"
        android:textColor="#333333"/>

</LinearLayout>

</ScrollView>
```

Androidmanifest.xml

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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="SQLite CRUD"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.AppCompat.Light.DarkActionBar">
        <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>
```

Output:





ID: 1
Name: Aniket Shinde
Email: aniketshinde@gmail.com

6. Write a program to create a user registration form, after registration data will be inserted in the SQLite database, and design an activity that displays that information.

MainActivity.java

```
package com.example.userregistrationsqlite;

import android.os.Bundle;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

  private EditText etName, etEmail, etPassword;
  private Button btnRegister;
  private DBHelper dbHelper;

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

    etName = findViewById(R.id.etName);
    etEmail = findViewById(R.id.etEmail);
    etPassword = findViewById(R.id.etPassword);
    btnRegister = findViewById(R.id.btnRegister);

    dbHelper = new DBHelper(this);

    btnRegister.setOnClickListener(v -> registerUser());
  }

  private void registerUser() {
    String name = etName.getText().toString();
    String email = etEmail.getText().toString();
    String password = etPassword.getText().toString();

    if (name.isEmpty() || email.isEmpty() || password.isEmpty()) {
      Toast.makeText(this, "Fill all fields", Toast.LENGTH_SHORT).show();
```

```
            return;
        }

        boolean inserted = dbHelper.registerUser(name, email, password);
        if (inserted) {
            Toast.makeText(this, "User Registered Successfully",
Toast.LENGTH_SHORT).show();
            etName.setText("");
            etEmail.setText("");
            etPassword.setText("");
        } else {
            Toast.makeText(this, "Registration Failed", Toast.LENGTH_SHORT).show();
        }
    }
}
```

DBHelper.java

```
package com.example.userregistrationsqlite;

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

public class DBHelper extends SQLiteOpenHelper {

    public static final String DB_NAME = "RegistrationDB.db";
    public static final int DB_VERSION = 1;

    public static final String TABLE_NAME = "users";
    public static final String COL_ID = "id";
    public static final String COL_NAME = "name";
    public static final String COL_EMAIL = "email";
    public static final String COL_PASSWORD = "password";

    public DBHelper(Context context) {
        super(context, DB_NAME, null, DB_VERSION);
    }
    @Override
    public void onCreate(SQLiteDatabase db) {
        String createTable = "CREATE TABLE " + TABLE_NAME +
                " (" + COL_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, " +
                COL_NAME + " TEXT, " +
                COL_EMAIL + " TEXT, " +
                COL_PASSWORD + " TEXT)";
        db.execSQL(createTable);
    }

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

    public boolean registerUser(String name, String email, String password) {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues cv = new ContentValues();
        cv.put(COL_NAME, name);
        cv.put(COL_EMAIL, email);
        cv.put(COL_PASSWORD, password);
        long result = db.insert(TABLE_NAME, null, cv);
        return result != -1;
    }
}
```

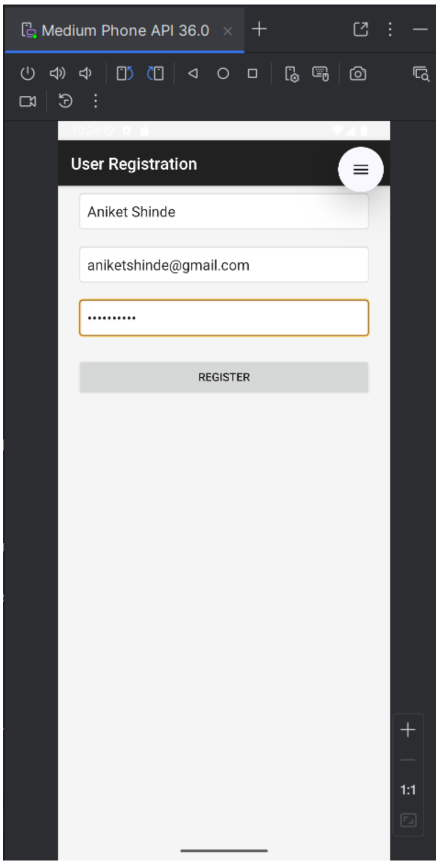
activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:padding="24dp"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#f5f5f5">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="User Registration"
        android:textSize="24sp"
        android:textStyle="bold"
        android:textColor="#333333"
        android:gravity="center"
        android:layout_marginBottom="30dp"/>

    <EditText
        android:id="@+id/etName"
        android:layout_width="match_parent"
        android:layout_height="50dp"
        android:hint="Name"
        android:background="@android:drawable/editbox_background"
        android:paddingLeft="12dp"
        android:paddingRight="12dp"
        android:layout_marginBottom="16dp"/>

    <EditText
        android:id="@+id/etEmail"
        android:layout_width="match_parent"
        android:layout_height="50dp"
        android:hint="Email"
        android:background="@android:drawable/editbox_background"
        android:paddingLeft="12dp"
        android:paddingRight="12dp"
        android:inputType="textEmailAddress"
        android:layout_marginBottom="16dp"/>

    <EditText
        android:id="@+id/etPassword"
        android:layout_width="match_parent"
        android:layout_height="50dp"
        android:hint="Password"
        android:inputType="textPassword"
        android:background="@android:drawable/editbox_background"
        android:paddingLeft="12dp"
        android:paddingRight="12dp"
        android:layout_marginBottom="24dp"/>
    <Button
        android:id="@+id/btnRegister"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Register"/>

</LinearLayout>
```

AndroidManifest.xml

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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="User Registration"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.AppCompat.Light.DarkActionBar">
        <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>
```

Output :

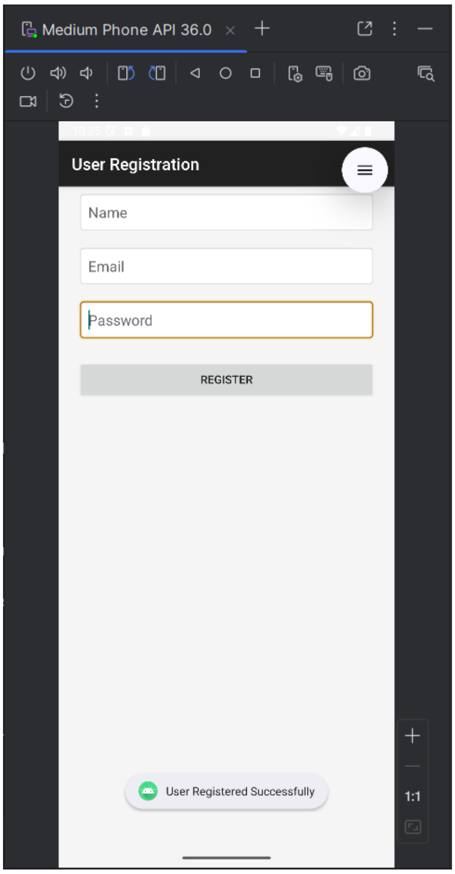


User Registration

Aniket Shinde

aniketshinde@gmail.com

••••••••••



User Registration

Name

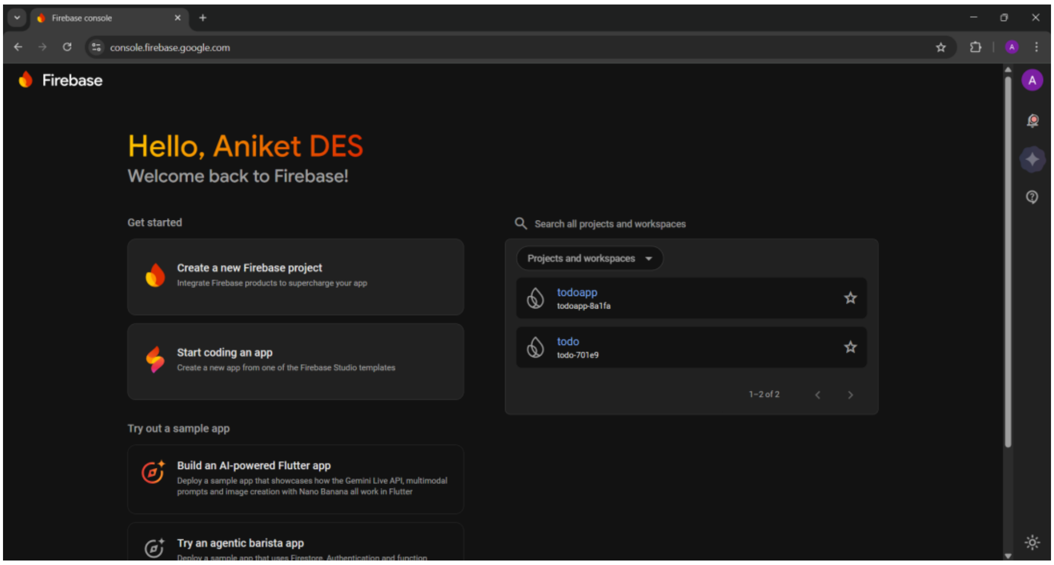
Email

Password

User Registered Successfully

7. Android Program to perform CRUD operation using real time database Firebase.

Step 1 : Go to: https://console.firebase.google.com



Hello, Aniket DES

Welcome back to Firebase!

Create a new Firebase project
Integrate Firebase products to supercharge your app

Start coding an app
Create a new app from one of the Firebase Studio templates

todoapp
todoapp-8a1fa

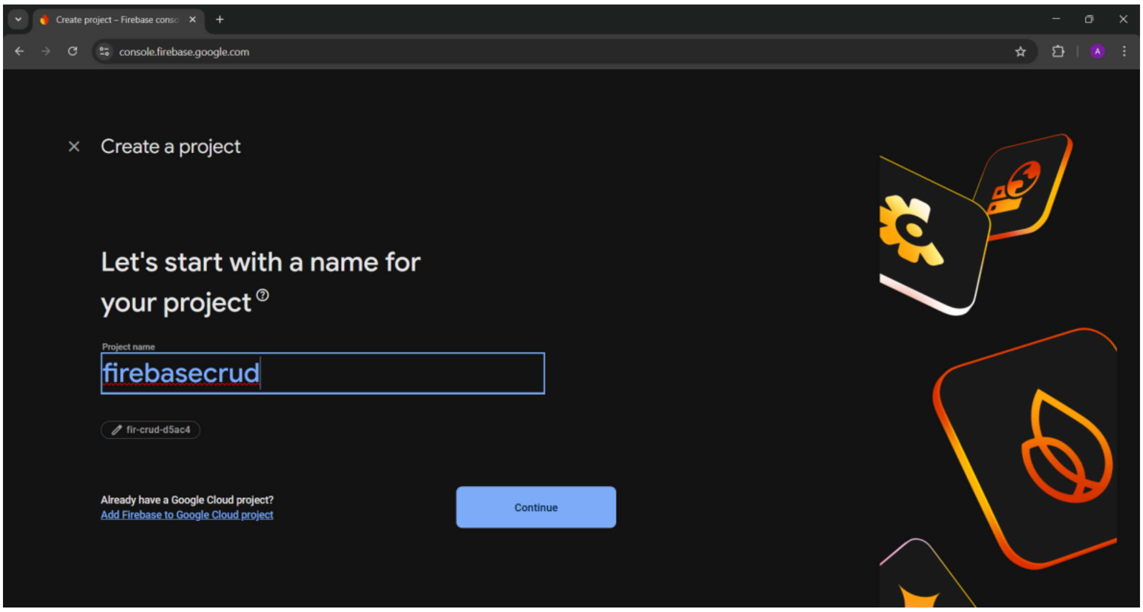
todo
todo-701e9

Build an AI-powered Flutter app
Deploy a sample app that showcases how the Gemini Live API, multimodal prompts and image creation with Nano Banana all work in Flutter

Try an agentic barista app
Deploy a sample app that uses Firestore, Authentication and function

Step 2 :Click **"Add Project"**.

Step 3 : Enter a **Project Name**, click **Continue**.



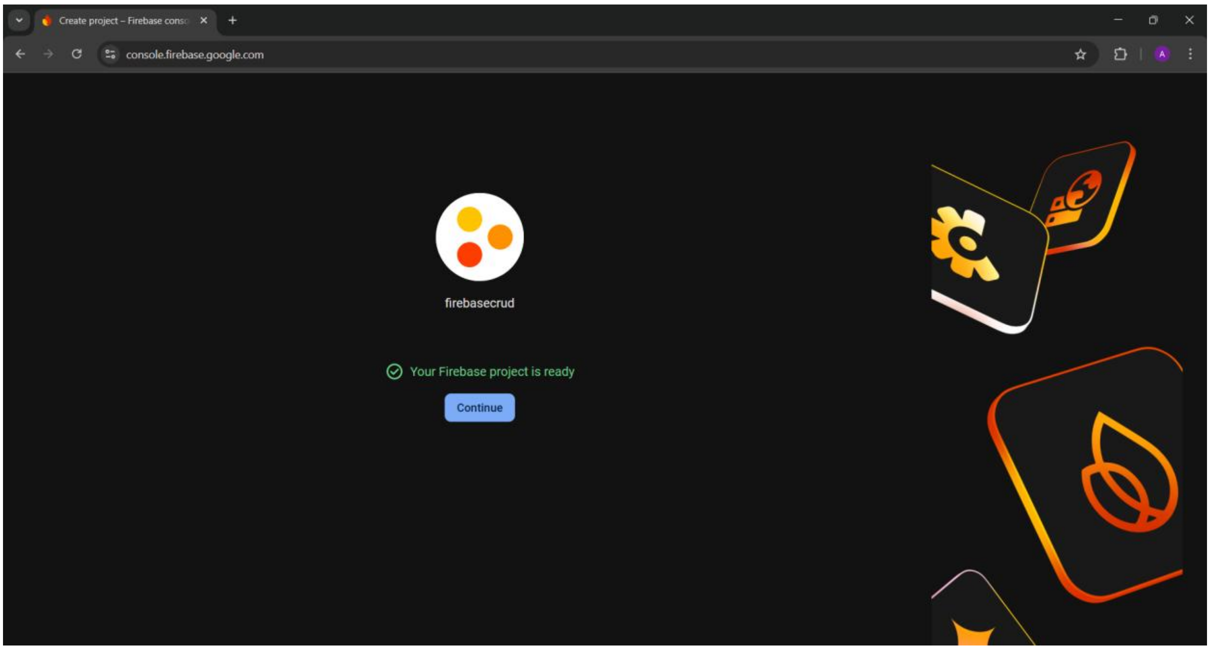
Let's start with a name for your project

Project name: firebasecrud

fir-crud-d5ac4

Already have a Google Cloud project?
Add Firebase to Google Cloud project

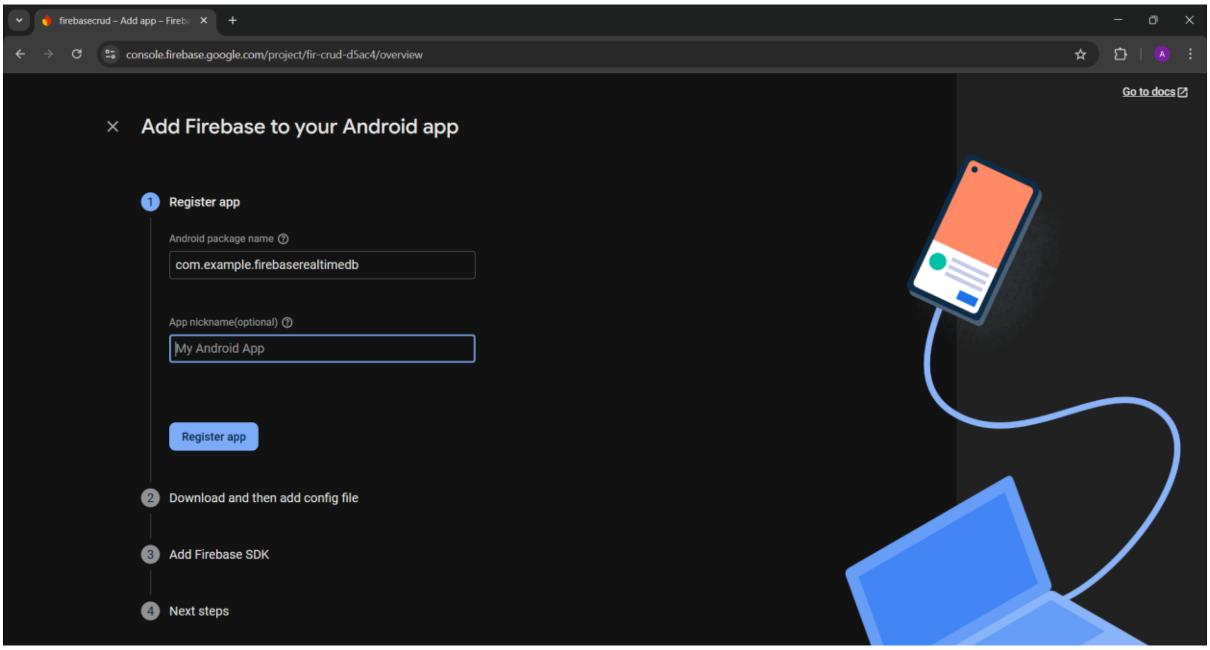
Step 4 : Disable or enable **Google Analytics** (your choice), click **Create Project**.



Your Firebase project is ready

Step 5 : After creating the project, click **"Add app"** → **Android** icon.

- **Android package name** (e.g. com.example.firebasecrud)
- **App nickname** (optional)
- **SHA-1** (required for Firebase Auth, optional for Realtime DB)
-



firebasecrud – Add app – Fireb
console.firebase.google.com/project/fir-crud-d5ac4/overview
Go to docs
Register app

Add Firebase to your Android app

1 Register app

Android package name: com.example.firebaserealtimedb

App nickname(optional): My Android App

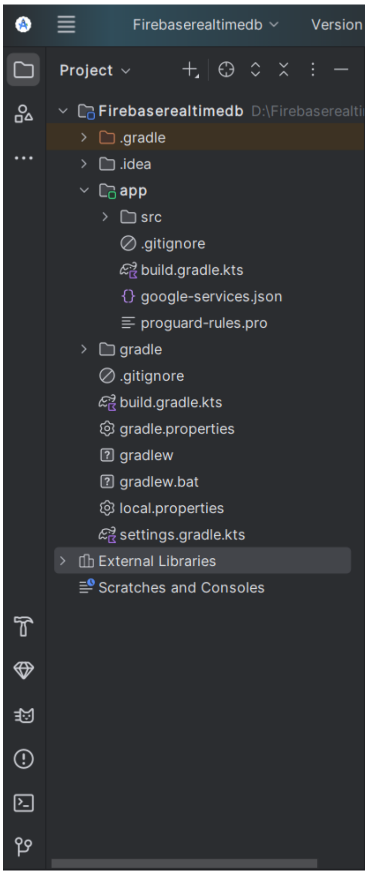
2 Download and then add config file

3 Add Firebase SDK

4 Next steps

Step 6 : Download '**google-services.json**'

1. After registration, click **"Download google-services.json"**.
2. Place this file inside your project:
3. app/ --> paste here



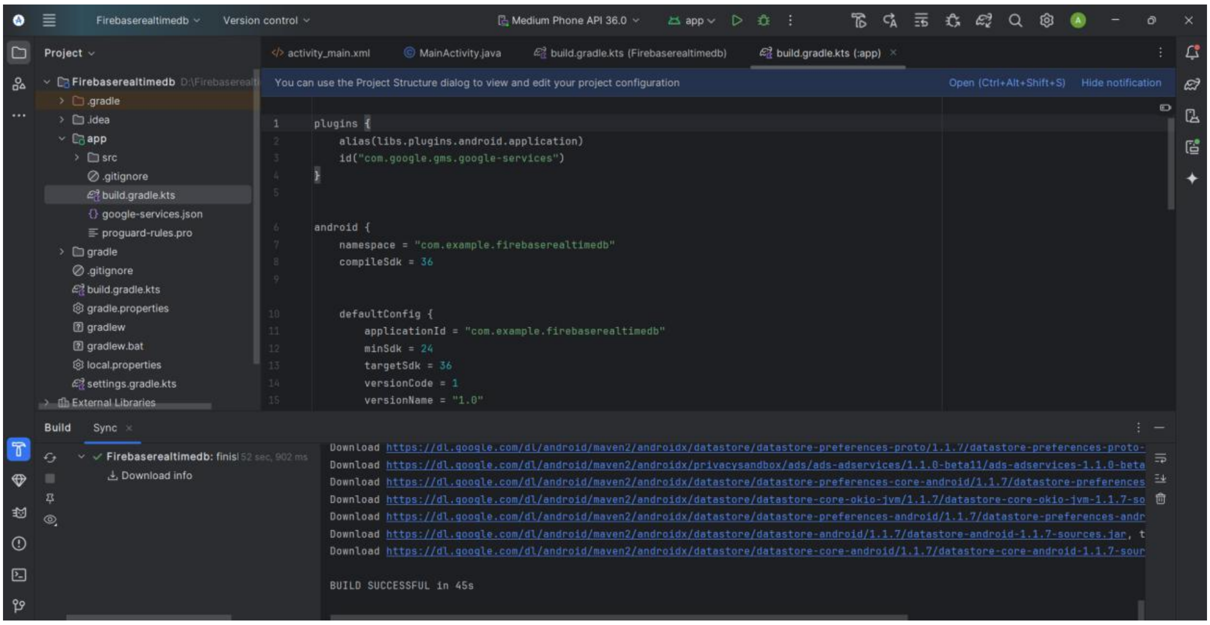
Step 7 : Add Firebase SDK to Your App

In Project-level build.gradle:

```
buildscript {
   dependencies {
      id("com.google.gms.google-services") version "4.4.3" apply false

   }
}
```

In App-level(Add this) build.gradle:

```
plugins {
            alias(libs.plugins.android.application)

id("com.google.gms.google-services") // Add this
}

dependencies {
   // Firebase Realtime Database and Auth
        implementation("com.google.firebase:firebase-database:20.3.0")
        implementation("com.google.firebase:firebase-auth:22.3.0")
        implementation(platform("com.google.firebase:firebase-bom:34.2.0"))
        implementation("com.google.firebase:firebase-analytics")

}
```



You can use the Project Structure dialog to view and edit your project configuration

```
plugins {
    alias(libs.plugins.android.application)
    id("com.google.gms.google-services")
}


android {
    namespace = "com.example.firebaserealtimedb"
    compileSdk = 36


    defaultConfig {
        applicationId = "com.example.firebaserealtimedb"
        minSdk = 24
        targetSdk = 36
        versionCode = 1
        versionName = "1.0"
```

```
Download https://dl.google.com/dl/android/maven2/androidx/datastore/datastore-preferences-proto/1.1.7/datastore-preferences-proto-
Download https://dl.google.com/dl/android/maven2/androidx/privacysandbox/ads/ads-adservices/1.1.0-beta11/ads-adservices-1.1.0-beta
Download https://dl.google.com/dl/android/maven2/androidx/datastore/datastore-preferences-core-android/1.1.7/datastore-preferences
Download https://dl.google.com/dl/android/maven2/androidx/datastore/datastore-core-okio-jvm/1.1.7/datastore-core-okio-jvm-1.1.7-so
Download https://dl.google.com/dl/android/maven2/androidx/datastore/datastore-preferences-android/1.1.7/datastore-preferences-andr
Download https://dl.google.com/dl/android/maven2/androidx/datastore/datastore-android/1.1.7/datastore-android-1.1.7-sources.jar, t
Download https://dl.google.com/dl/android/maven2/androidx/datastore/datastore-core-android/1.1.7/datastore-core-android-1.1.7-sour

BUILD SUCCESSFUL in 45s
```

Step 8 : Sync Project with Gradle

Click **"Sync Now"** when prompted in Android Studio.

Step 9 : Enable Realtime Database in Firebase Console

1. Go to **Firebase Console** → Your Project
2. Open **Build > Realtime Database**
3. Click **"Create Database"**
4. Select your region and start in **test mode** (for development)

```
{
  "rules": {
    ".read": true,
    ".write": true
  }
```

5. Click Publish

Mainactivity.java

```
package com.example.firebaserealtimedb;

import android.os.Bundle;
import android.widget.*;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import com.google.firebase.analytics.FirebaseAnalytics;
import com.google.firebase.database.*;

import java.util.*;

public class MainActivity extends AppCompatActivity {

  EditText etName, etEmail;
  Button btnAdd, btnUpdate, btnDelete, btnViewAll;
  ListView listView;

  FirebaseAnalytics firebaseAnalytics;
  DatabaseReference dbRef;
  List<User> userList;
  String selectedUserId = "";

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

    firebaseAnalytics = FirebaseAnalytics.getInstance(this);

    // Initialize views
    etName = findViewById(R.id.etName);
    etEmail = findViewById(R.id.etEmail);
    btnAdd = findViewById(R.id.btnAdd);
    btnUpdate = findViewById(R.id.btnUpdate);
    btnDelete = findViewById(R.id.btnDelete);
    btnViewAll = findViewById(R.id.btnViewAll);
```

```
        listView = findViewById(R.id.listView);

        dbRef = FirebaseDatabase.getInstance().getReference("users");
        userList = new ArrayList<>();

        // Button actions
        btnAdd.setOnClickListener(view -> {
            addUser();
            logEvent("add_user");
        });

        btnUpdate.setOnClickListener(view -> {
            updateUser();
            logEvent("update_user");
        });

        btnDelete.setOnClickListener(view -> {
            deleteUser();
            logEvent("delete_user");
        });

        btnViewAll.setOnClickListener(view -> {
            retrieveUsers(); // Fetch when View All is clicked
        });

        listView.setOnItemClickListener((adapterView, view, i, l) -> {
            User user = userList.get(i);
            etName.setText(user.getName());
            etEmail.setText(user.getEmail());
            selectedUserId = user.getId();
        });
    }

    private void logEvent(String action) {
        Bundle bundle = new Bundle();
        bundle.putString(FirebaseAnalytics.Param.METHOD, action);
        firebaseAnalytics.logEvent(FirebaseAnalytics.Event.SELECT_CONTENT,
bundle);
    }
    private void addUser() {
        String name = etName.getText().toString().trim();
        String email = etEmail.getText().toString().trim();

        if (name.isEmpty() || email.isEmpty()) {
            Toast.makeText(this, "Enter name and email",
Toast.LENGTH_SHORT).show();
            return;
        }

        String id = dbRef.push().getKey();
        User user = new User(id, name, email);
        dbRef.child(id).setValue(user);
        Toast.makeText(this, "User Added", Toast.LENGTH_SHORT).show();

        etName.setText("");
        etEmail.setText("");
    }

    private void updateUser() {
        if (!selectedUserId.isEmpty()) {
            String name = etName.getText().toString().trim();
            String email = etEmail.getText().toString().trim();

            if (name.isEmpty() || email.isEmpty()) {
                Toast.makeText(this, "Enter valid name and email",
Toast.LENGTH_SHORT).show();
                return;
            }

            User user = new User(selectedUserId, name, email);
            dbRef.child(selectedUserId).setValue(user);
            Toast.makeText(this, "User Updated", Toast.LENGTH_SHORT).show();
            retrieveUsers(); // Refresh after update
        } else {
            Toast.makeText(this, "Select a user to update",
Toast.LENGTH_SHORT).show();
        }
    }
```

```
   private void deleteUser() {
      if (!selectedUserId.isEmpty()) {
         dbRef.child(selectedUserId).removeValue();
         Toast.makeText(this, "User Deleted", Toast.LENGTH_SHORT).show();

         etName.setText("");
         etEmail.setText("");
         selectedUserId = "";

         retrieveUsers(); // Refresh after delete
      } else {
         Toast.makeText(this, "Select a user to delete",
Toast.LENGTH_SHORT).show();
      }
   }

   private void retrieveUsers() {
      dbRef.addListenerForSingleValueEvent(new ValueEventListener() {
         @Override
         public void onDataChange(@NonNull DataSnapshot snapshot) {
            userList.clear();
            for (DataSnapshot postSnapshot : snapshot.getChildren()) {
               User user = postSnapshot.getValue(User.class);
               userList.add(user);
            }

            List<String> displayList = new ArrayList<>();
            for (User user : userList) {
               displayList.add("Name: " + user.getName() + "\nEmail: " +
user.getEmail());
            }

            ArrayAdapter<String> adapter = new ArrayAdapter<>(MainActivity.this,
                  android.R.layout.simple_list_item_1, displayList);
            listView.setAdapter(adapter);

            Toast.makeText(MainActivity.this, "Data displayed on device",
Toast.LENGTH_SHORT).show();
         }
```

```
            @Override
            public void onCancelled(@NonNull DatabaseError error) {
                Toast.makeText(MainActivity.this, "Error: " + error.getMessage(),
Toast.LENGTH_SHORT).show();
            }
        });
    }
}
```

User.java

```
package com.example.firebaserealtimedb;

public class User {
    private String id;
    private String name;
    private String email;

    // Empty constructor required for Firebase
    public User() {}

    public User(String id, String name, String email) {
        this.id = id;
        this.name = name;
        this.email = email;
    }

    public String getId() { return id; }
    public String getName() { return name; }
    public String getEmail() { return email; }

    public void setId(String id) { this.id = id; }
    public void setName(String name) { this.name = name; }
    public void setEmail(String email) { this.email = email; }
}
```

Activity_main.xml

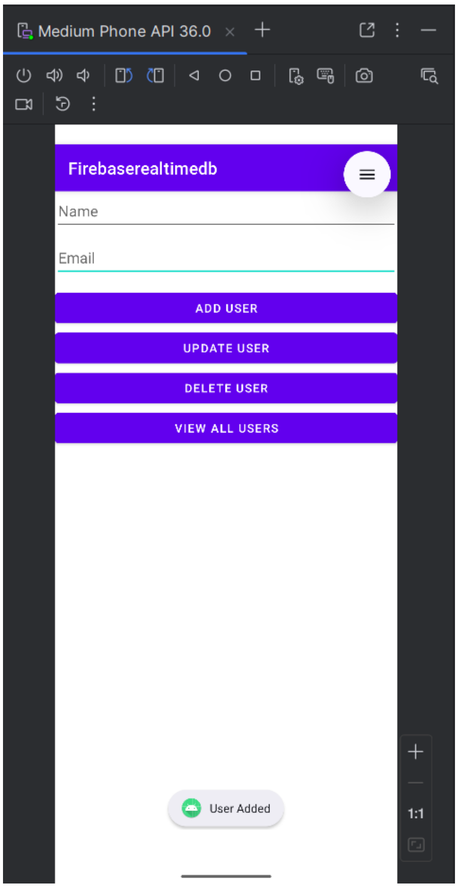
```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:app="http://schemas.android.com/apk/res-auto"
  xmlns:tools="http://schemas.android.com/tools"
  android:id="@+id/main"
  android:layout_width="match_parent"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

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

    <EditText
      android:id="@+id/etName"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:layout_marginBottom="8dp"
      android:hint="Name"
      android:inputType="textPersonName"
      android:minHeight="48dp" />

    <EditText
      android:id="@+id/etEmail"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:layout_marginBottom="12dp"
      android:hint="Email"
      android:inputType="textEmailAddress"
      android:minHeight="48dp" />

    <Button
      android:id="@+id/btnAdd"
      android:text="Add User"
```

```
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

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

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

    <Button
        android:id="@+id/btnViewAll"
        android:text="View All Users"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="8dp" />

    <ListView
        android:id="@+id/listView"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1" />
  </LinearLayout>

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

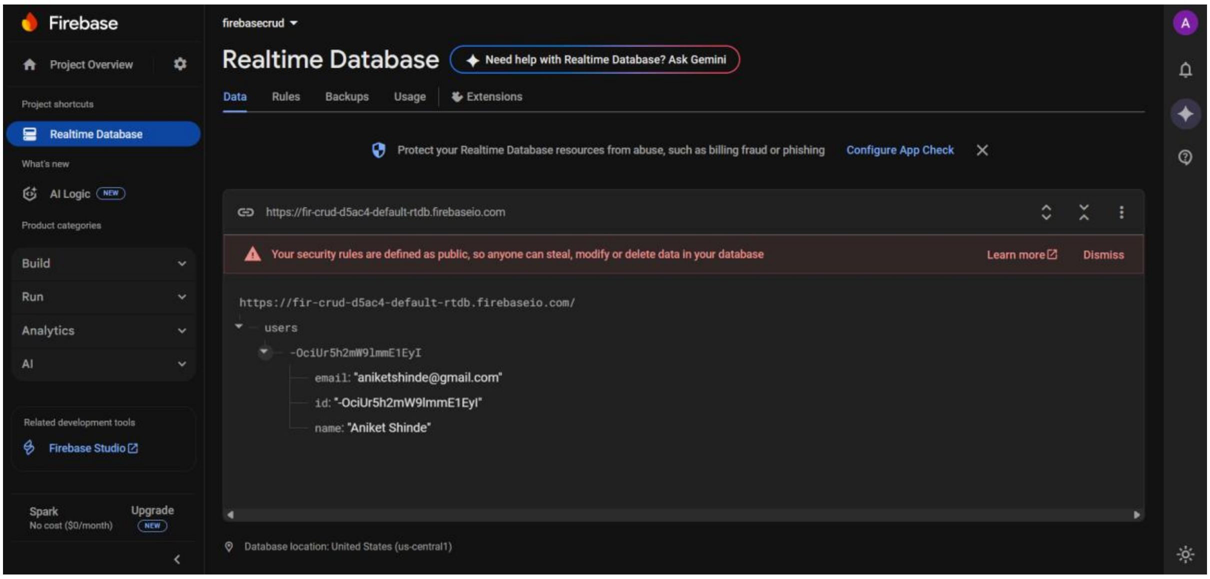
AndroidManifest.xml

```
<?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.firebaserealtimedb">

  <!-- Required Permissions -->
  <uses-permission android:name="android.permission.INTERNET" />
  <uses-permission
android:name="android.permission.ACCESS_NETWORK_STATE" />

  <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.Firebaserealtimedb"
    tools:targetApi="31">
    <meta-data
      android:name="com.google.firebase.database.DatabaseUrl"
      android:value="https://fir-crud-d5ac4-default-rtdb.firebaseio.com/" />
    <meta-data
      android:name="com.google.firebase.messaging.default_notification_icon"
      android:resource="@mipmap/ic_launcher" />
    <meta-data
      android:name="com.google.firebase.messaging.default_notification_color"
      android:resource="@color/purple_500" />

    <!-- Main Activity -->
    <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>
```

Output : 1. When the user created (Added)



User Added



Protect your Realtime Database resources from abuse, such as billing fraud or phishing

https://fir-crud-d5ac4-default-rtdb.firebaseio.com

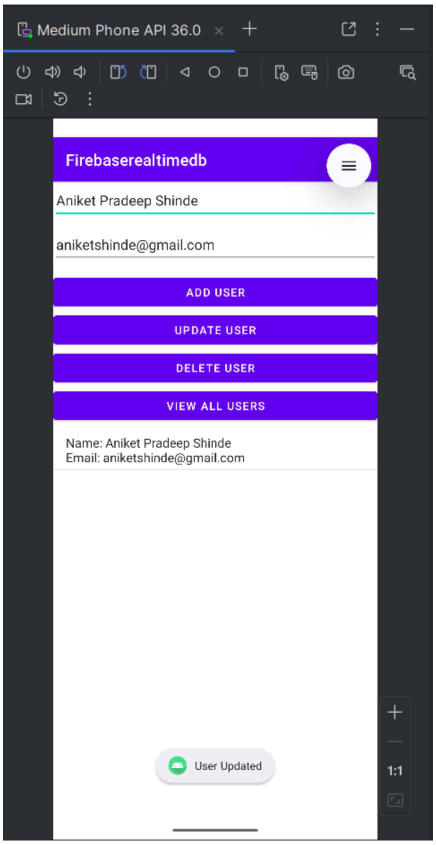
Your security rules are defined as public, so anyone can steal, modify or delete data in your database

```
https://fir-crud-d5ac4-default-rtdb.firebaseio.com/
users
  -OciUr5h2mW9lmmE1EyI
    email: "aniketshinde@gmail.com"
    id: "-OciUr5h2mW9lmmE1EyI"
    name: "Aniket Shinde"
```

Database location: United States (us-central1)

Figure 1 Firebase Database Output on user addition

Emulator output on the User updatation:

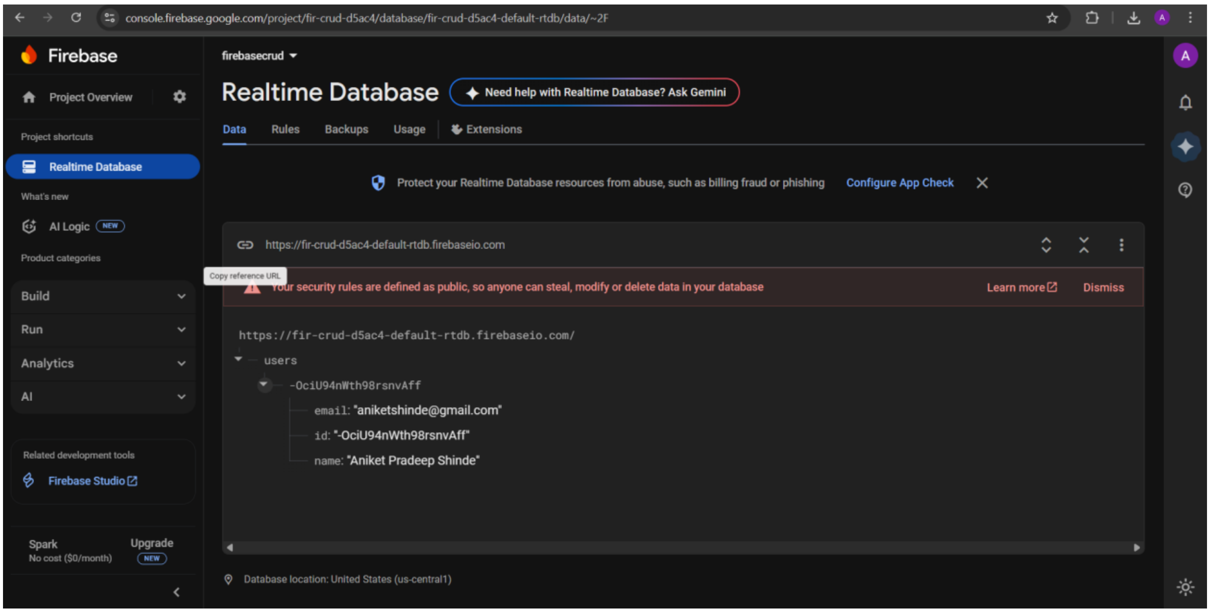


Aniket Pradeep Shinde

aniketshinde@gmail.com

Name: Aniket Pradeep Shinde
Email: aniketshinde@gmail.com

User Updated



Protect your Realtime Database resources from abuse, such as billing fraud or phishing

https://fir-crud-d5ac4-default-rtdb.firebaseio.com

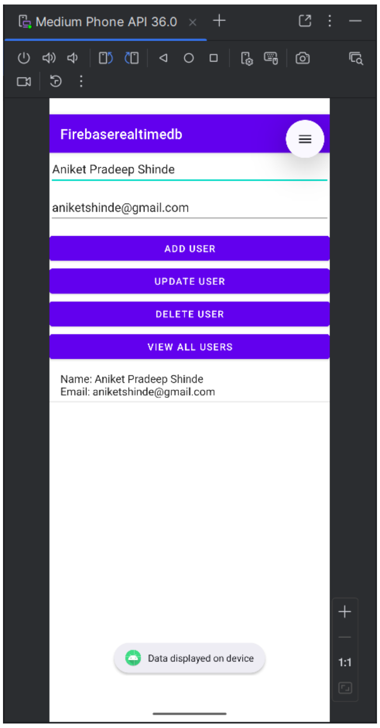
Your security rules are defined as public, so anyone can steal, modify or delete data in your database

```
https://fir-crud-d5ac4-default-rtdb.firebaseio.com/
  users
    -OciU94nWth98rsnvAff
      email: "aniketshinde@gmail.com"
      id: "-OciU94nWth98rsnvAff"
      name: "Aniket Pradeep Shinde"
```

Database location: United States (us-central1)

Figure 2 - Firebase Database Output on user update

Emulator Output on the View all users :



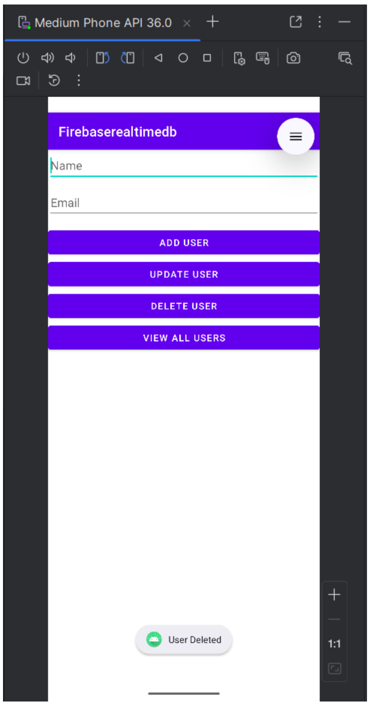
Aniket Pradeep Shinde

aniketshinde@gmail.com

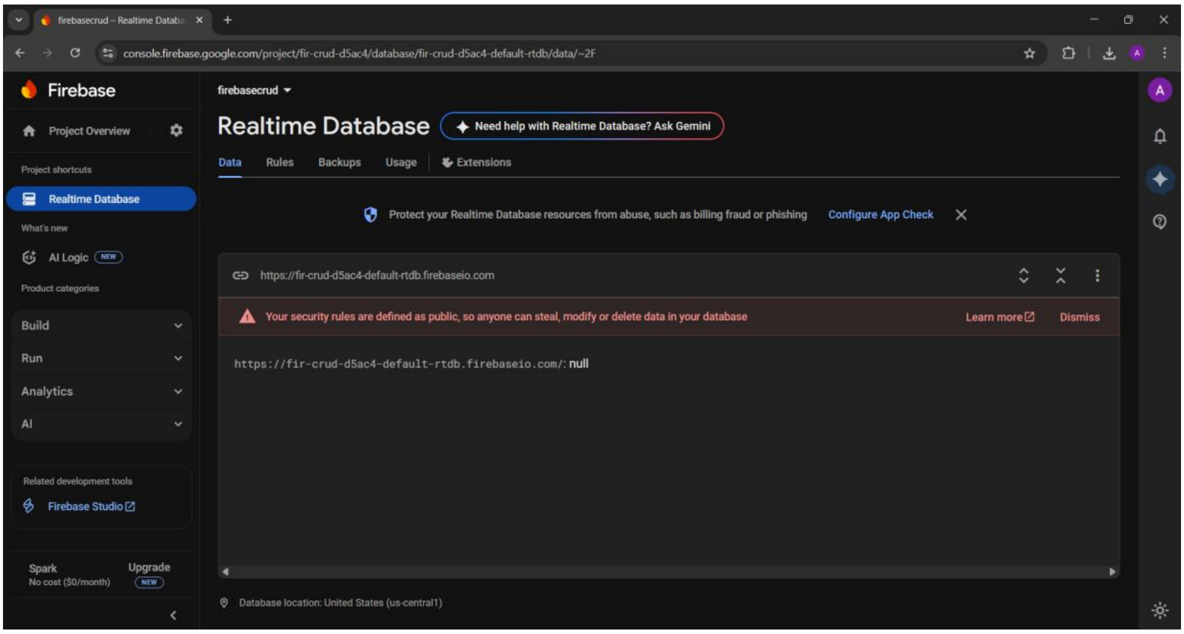
Name: Aniket Pradeep Shinde
Email: aniketshinde@gmail.com

Data displayed on device

Emulator Output on the deleting a user:



User Deleted



Realtime Database

Protect your Realtime Database resources from abuse, such as billing fraud or phishing

https://fir-crud-d5ac4-default-rtdb.firebaseio.com

Your security rules are defined as public, so anyone can steal, modify or delete data in your database

```
https://fir-crud-d5ac4-default-rtdb.firebaseio.com/: null
```

Database location: United States (us-central1)

Figure 3 - Firebase database output on the user deletion

Experiment No – 3

Animation, Multimedia and Location Based Services.

1. Write an Android application to play, pause, and stop an audio file.

Mainactivity.java

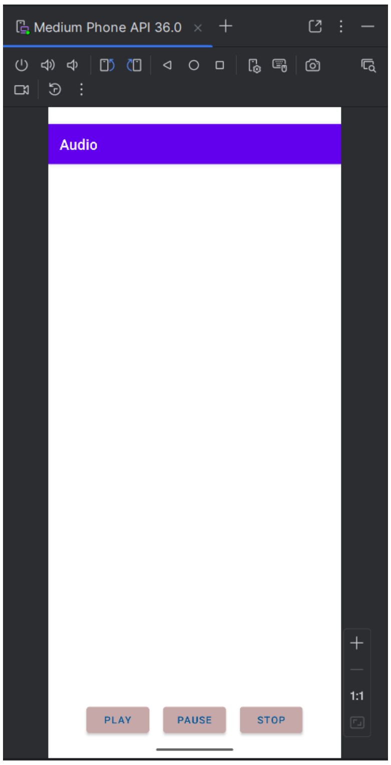
```
package com.example.audio;

import androidx.appcompat.app.AppCompatActivity;
import android.media.MediaPlayer;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class MainActivity extends AppCompatActivity {
   Button play, pause, stop;
   MediaPlayer mp;

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

      play = findViewById(R.id.btnPlay);
      pause = findViewById(R.id.btnPause);
      stop = findViewById(R.id.btnStop);

      play.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View v) {
            if (mp == null) {
               mp = MediaPlayer.create(getApplicationContext(), R.raw.song);
               mp.start();
            } else if (!mp.isPlaying()) {
               mp.start();
            }
         }
      });
```

```
        pause.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if (mp != null && mp.isPlaying()) {
                    mp.pause();
                }
            }
        });

        stop.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if (mp != null) {
                    mp.stop();
                    mp.release();
                    mp = null;
                }
            }
        });
    }
}
```

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="bottom|center_horizontal"
    android:orientation="horizontal"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/btnPlay"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:backgroundTint="#C6A8A8"
        android:text="Play"
        android:textColor="#01579B" />
    <Button
        android:id="@+id/btnPause"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:backgroundTint="#C6A8A8"
        android:text="Pause"
        android:textColor="#01579B" />

    <Button
        android:id="@+id/btnStop"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:backgroundTint="#C6A8A8"
        android:text="Stop"
        android:textColor="#01579B" />
</LinearLayout>
```

Output:



2. Write an Android app to play video with media controller.

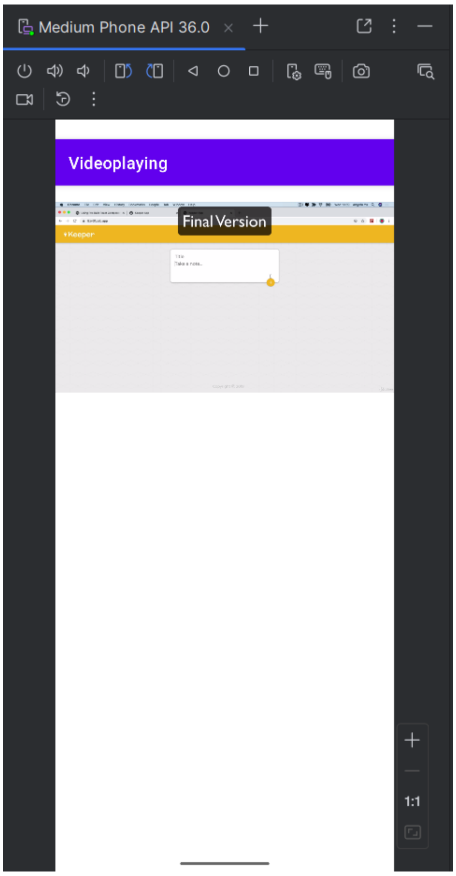
Mainactivity.java

```
package com.example.videoplaying;

import androidx.appcompat.app.AppCompatActivity;

import android.net.Uri;

import android.os.Bundle;

import android.widget.MediaController;

import android.widget.VideoView;

public class MainActivity extends AppCompatActivity {

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);


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


        // Store video resource path in a string variable

        String videoPath = "android.resource://" + getPackageName() + "/" +
R.raw.video;

        Uri uri = Uri.parse(videoPath);


        videoView.setVideoURI(uri);


        // Create and set media controller

        MediaController mediaController = new MediaController(this);

        videoView.setMediaController(mediaController);

        mediaController.setAnchorView(videoView);
        videoView.start();
    }
}
```

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout 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:paddingTop="100dp"
    tools:context=".MainActivity">

    <VideoView
        android:id="@+id/videoview"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />
</FrameLayout>
```

Output :



3. Create an Android app applying different animations on an image

mainactivity.java

```
package com.example.animation;
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 {
    ImageView animImage;
    Button btnFadeIn, btnFadeOut, btnZoomIn, btnRotate, btnSlideRight;

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

        animImage = findViewById(R.id.animImage);
        btnFadeIn = findViewById(R.id.btnFadeIn);
        btnFadeOut = findViewById(R.id.btnFadeOut);
        btnZoomIn = findViewById(R.id.btnZoomIn);
        btnRotate = findViewById(R.id.btnRotate);
        btnSlideRight = findViewById(R.id.btnSlideRight);

        btnFadeIn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Animation fadeIn = AnimationUtils.loadAnimation(MainActivity.this,
R.anim.fadein);
                animImage.startAnimation(fadeIn);
            }
        });

        btnFadeOut.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Animation fadeOut = AnimationUtils.loadAnimation(MainActivity.this,
R.anim.fadeout);
                animImage.startAnimation(fadeOut);
            }
        });

        btnZoomIn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Animation zoomIn = AnimationUtils.loadAnimation(MainActivity.this,
R.anim.zoomin);
                animImage.startAnimation(zoomIn);
            }
        });

        btnRotate.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Animation rotate = AnimationUtils.loadAnimation(MainActivity.this,
R.anim.rotateclockwise);
                animImage.startAnimation(rotate);
            }
        });

        btnSlideRight.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Animation slideRight = AnimationUtils.loadAnimation(MainActivity.this,
R.anim.slideright);
                animImage.startAnimation(slideRight);
            }
        });
    }
}
```

<u>activity_main.xml</u>

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
```

```
android:layout_height="match_parent">

<ImageView
   android:id="@+id/animImage"
   android:layout_width="200dp"
   android:layout_height="200dp"
   android:src="@mipmap/ic_launcher"
   android:layout_centerHorizontal="true"
   android:layout_marginTop="50dp" />

<Button
   android:id="@+id/btnFadeIn"
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:text="Fade In"
   android:layout_below="@id/animImage"
   android:layout_marginTop="32dp"
   android:layout_alignParentStart="true"
   android:layout_marginStart="32dp" />

<Button
   android:id="@+id/btnFadeOut"
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:text="Fade Out"
   android:layout_below="@id/animImage"
   android:layout_centerHorizontal="true"
   android:layout_marginTop="32dp" />

<Button
   android:id="@+id/btnZoomIn"
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:text="Zoom In"
   android:layout_below="@id/animImage"
   android:layout_alignParentEnd="true"
   android:layout_marginTop="32dp"
   android:layout_marginEnd="32dp" />

<Button
```

```
        android:id="@+id/btnRotate"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Rotate"
        android:layout_below="@id/btnFadeOut"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="16dp" />

    <Button
        android:id="@+id/btnSlideRight"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Slide Right"
        android:layout_below="@id/btnRotate"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="16dp" />
</RelativeLayout>
```

Rotateclockwise.xml

```
<?xml version="1.0" encoding="utf-8"?>
<rotate xmlns:android="http://schemas.android.com/apk/res/android"
    android:fromDegrees="0"
    android:toDegrees="360"
    android:pivotX="50%"
    android:pivotY="50%"
    android:duration="3000"
    android:repeatCount="infinite" />
```

Zoomin.xml

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:fillAfter="true">
    <scale
        android:duration="1000"
        android:fromXScale="1.0"
        android:toXScale="3.0"
        android:fromYScale="1.0"
```

```
      android:toYScale="3.0"
      android:pivotX="50%"
      android:pivotY="50%" />
</set>
```

Fadeout.xml

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
   android:fillAfter="true">
   <alpha
      android:duration="1000"
      android:fromAlpha="1.0"
      android:toAlpha="0.0"
      android:interpolator="@android:anim/accelerate_interpolator"/>
</set>
```

Fadein.xml

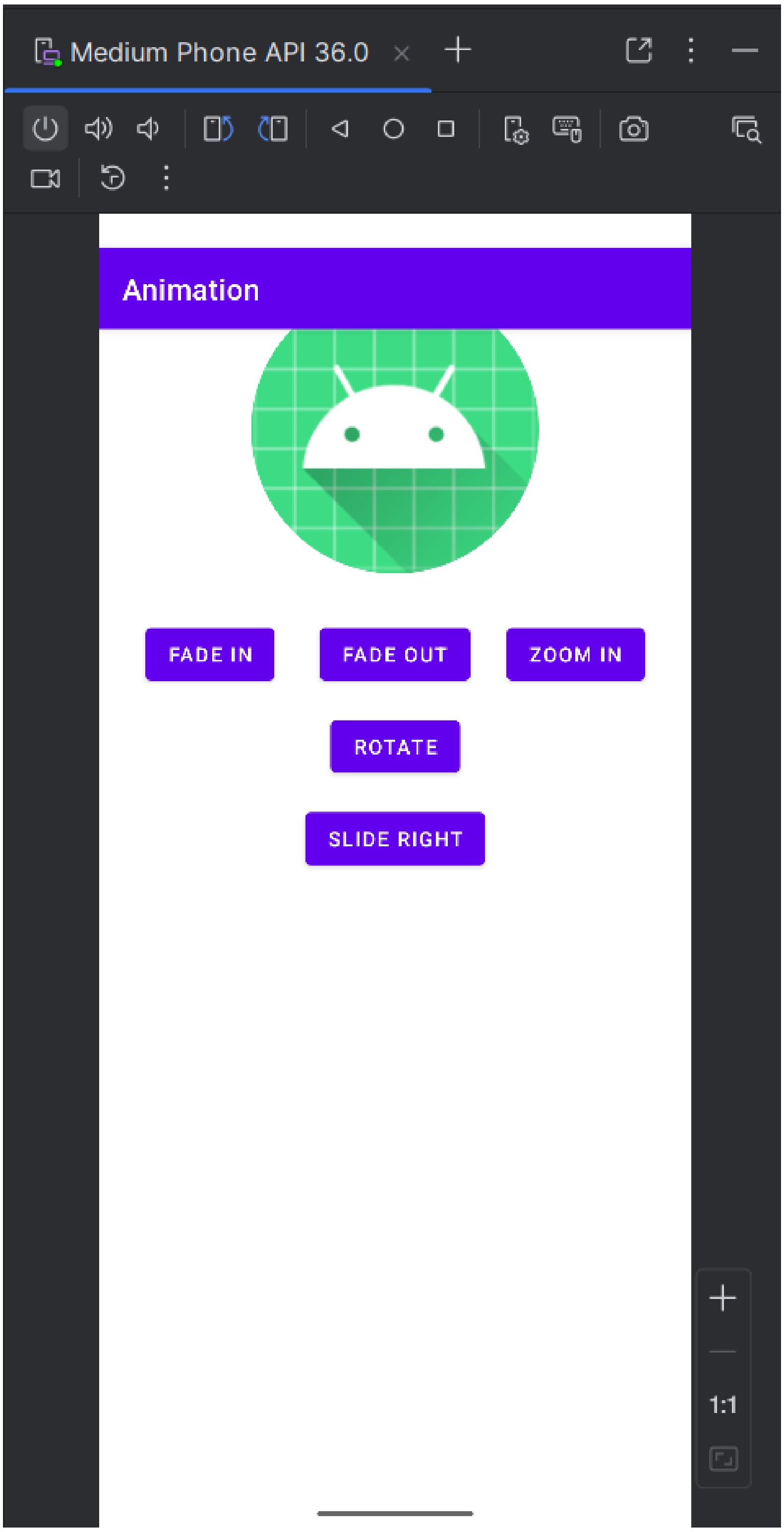
```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android">
   <alpha
      android:duration="1000"
      android:fromAlpha="0.0"
      android:toAlpha="1.0" />
</set>
```

Bounce.xml

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

   <scale
      android:duration="500"
      android:fromXScale="1.0"
      android:fromYScale="0.0"
      android:toXScale="1.0"
```

```
    android:toYScale="1.0" />

</set>
```

Output



4: Create an Android app implementing frame-by-frame animation using AnimationDrawable.

Mainactivity.java

```
package com.example.frameanimation;
import androidx.appcompat.app.AppCompatActivity;
import android.graphics.drawable.AnimationDrawable;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;

public class MainActivity extends AppCompatActivity {

   private ImageView img;
   private Button btnStartStop;
   private AnimationDrawable animation;

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

      img = findViewById(R.id.img);
      btnStartStop = findViewById(R.id.btnstartstop);

      // Get the background animation drawable
      animation = (AnimationDrawable) img.getBackground();

      btnStartStop.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View v) {
            if (animation.isRunning()) {
               animation.stop();
               btnStartStop.setText("Start");
            } else {
               animation.start();
               btnStartStop.setText("Stop");
            }
         }
      });
   }
}
```

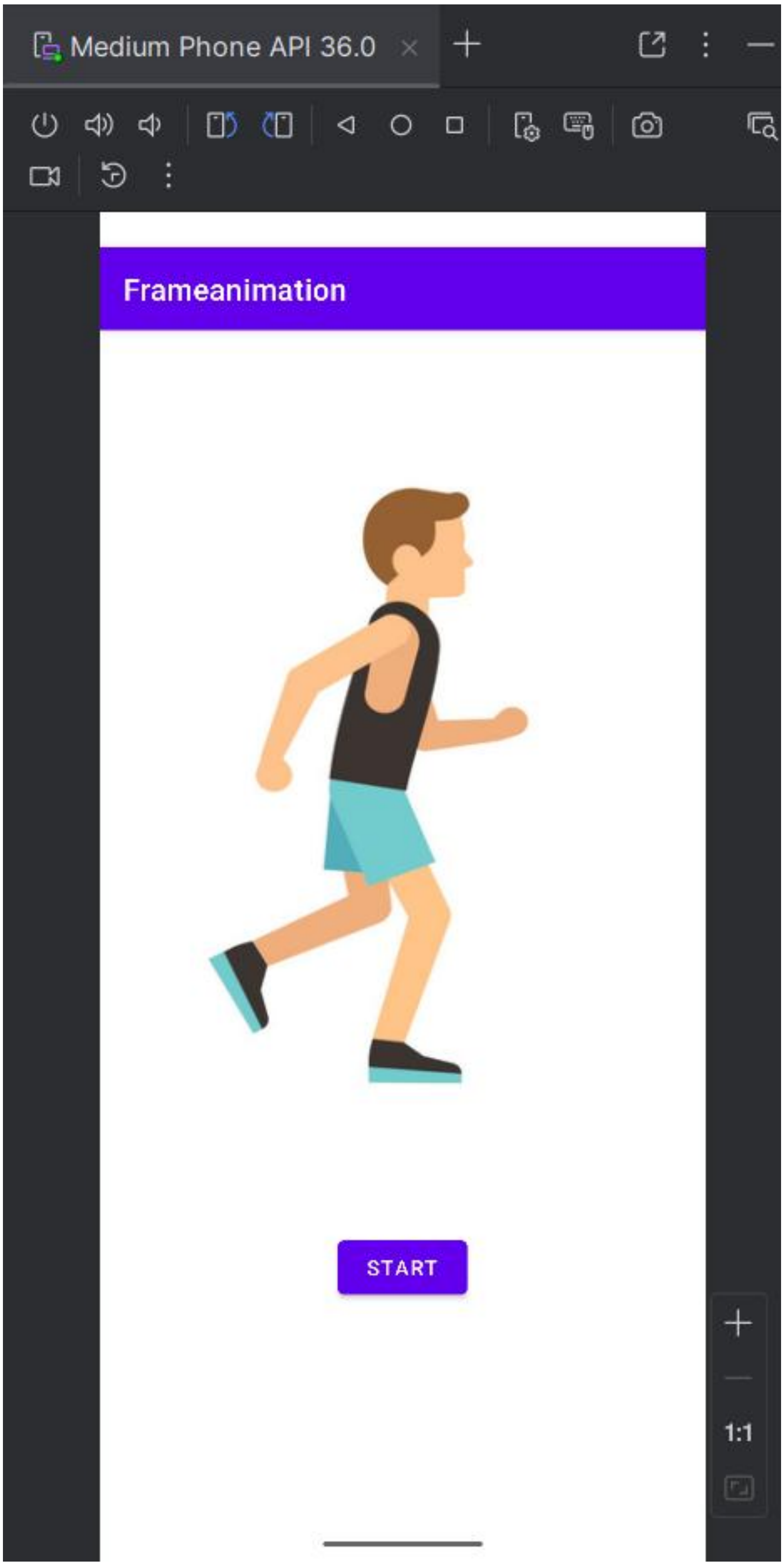
Activity_main.xml

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

   <ImageView
      android:id="@+id/img"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:background="@drawable/running" />

   <Button
      android:id="@+id/btnstartstop"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Start" />
</LinearLayout>
```

Running.xml

```
<?xml version="1.0" encoding="utf-8"?>
<animation-list xmlns:android="http://schemas.android.com/apk/res/android"
   android:oneshot="false">
   <item android:drawable="@drawable/frame1" android:duration="100" />
   <item android:drawable="@drawable/frame2" android:duration="100" />
   <item android:drawable="@drawable/frame3" android:duration="100" />
   <item android:drawable="@drawable/frame4" android:duration="100" />
</animation-list>
```

Output:



5. Creating an Android app to display the current location with longitude and latitude values.

Mainactivity.java

```
package com.example.locationapp;

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

import android.Manifest;
import android.annotation.SuppressLint;
import android.content.pm.PackageManager;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity implements LocationListener
{

    private static final int LOCATION_PERMISSION_REQUEST_CODE = 100;
    private LocationManager locationManager;
    private TextView textView;
    private Button button;

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

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

    private void checkLocationPermissionAndFetchLocation() {
        if (ContextCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(this, new String[] {
Manifest.permission.ACCESS_FINE_LOCATION },
LOCATION_PERMISSION_REQUEST_CODE);
        } else {
            getLocation();
        }
    }

    @SuppressLint("MissingPermission")
    private void getLocation() {
        try {
            locationManager = (LocationManager)
getSystemService(LOCATION_SERVICE);

locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER,
5000, 5, this);
        } catch (Exception e) {
            e.printStackTrace();
            Toast.makeText(this, "Error getting location",
Toast.LENGTH_SHORT).show();
        }
    }

    @Override
    public void onLocationChanged(@NonNull Location location) {
        double latitude = location.getLatitude();
        double longitude = location.getLongitude();
        textView.setText("Latitude: " + latitude + "\nLongitude: " + longitude);
    }
    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull String[]
permissions, @NonNull int[] grantResults) {
        if (requestCode == LOCATION_PERMISSION_REQUEST_CODE) {
            if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
                getLocation();
            } else {
                Toast.makeText(this, "Location permission denied",
Toast.LENGTH_SHORT).show();
            }
        }
    }

    // Other required LocationListener methods (empty implementations)
    @Override public void onStatusChanged(String provider, int status, Bundle
extras) { }
    @Override public void onProviderEnabled(@NonNull String provider) { }
    @Override public void onProviderDisabled(@NonNull String provider) { }
}
```

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:padding="10dp"
    android:orientation="vertical"
    android:gravity="center_horizontal"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:id="@+id/textView"
        android:textSize="18sp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Location will be displayed here"
        android:gravity="center"
        android:padding="10dp"/>
    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Check Location" />
</LinearLayout>
```

AndroidManifest.xml

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

    <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:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true">
        <activity android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

6 - Create an Android application that displays the current location of your device from longitude and latitude values (Reverse Geocoding).

Mainactivity.java

```
package com.example.reversegeocoding;

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

import android.Manifest;
import android.content.pm.PackageManager;
import android.location.Address;
import android.location.Geocoder;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import java.io.IOException;
import java.util.List;
import java.util.Locale;

public class MainActivity extends AppCompatActivity implements LocationListener
{

    private LocationManager locationManager;
    private TextView tvAddress;
    private Button btnGetAddress;
    private static final int LOCATION_PERMISSION_REQUEST_CODE = 101;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        tvAddress = findViewById(R.id.tv_address);
        btnGetAddress = findViewById(R.id.btn_getAddress);

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

    private void checkPermissionAndFetchLocation() {
        if (ContextCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(this,
                    new String[]{Manifest.permission.ACCESS_FINE_LOCATION},
LOCATION_PERMISSION_REQUEST_CODE);
        } else {
            getCurrentLocation();
        }
    }

    private void getCurrentLocation() {
        locationManager = (LocationManager)
getSystemService(LOCATION_SERVICE);
        try {

locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER,
5000, 5, this);
        } catch (SecurityException e) {
            e.printStackTrace();
        }
    }

    @Override
    public void onLocationChanged(Location location) {
        double lat = location.getLatitude();
        double longi = location.getLongitude();

        try {
            Geocoder geocoder = new Geocoder(getApplicationContext(),
Locale.getDefault());
            List<Address> addressList = geocoder.getFromLocation(lat, longi, 1);
            if (addressList != null && !addressList.isEmpty()) {
                Address address = addressList.get(0);
                String currentAddress = address.getAddressLine(0);
                tvAddress.setText("Address: " + currentAddress);
            } else {
                tvAddress.setText("No Address Found");
            }
        } catch (IOException e) {
            e.printStackTrace();
            Toast.makeText(this, "Unable to get address. Check network connection.",
Toast.LENGTH_SHORT).show();
        }
    }

    @Override
    public void onRequestPermissionsResult(int requestCode, String[] permissions,
int[] grantResults) {
        if (requestCode == LOCATION_PERMISSION_REQUEST_CODE) {
            if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
                getCurrentLocation();
            } else {
                Toast.makeText(this, "Location Permission Denied",
Toast.LENGTH_SHORT).show();
            }
        }
    }

    @Override public void onStatusChanged(String provider, int status, Bundle
extras) {}
    @Override public void onProviderEnabled(String provider) {}
    @Override public void onProviderDisabled(String provider) {}

}
```

acitivity_main.xml

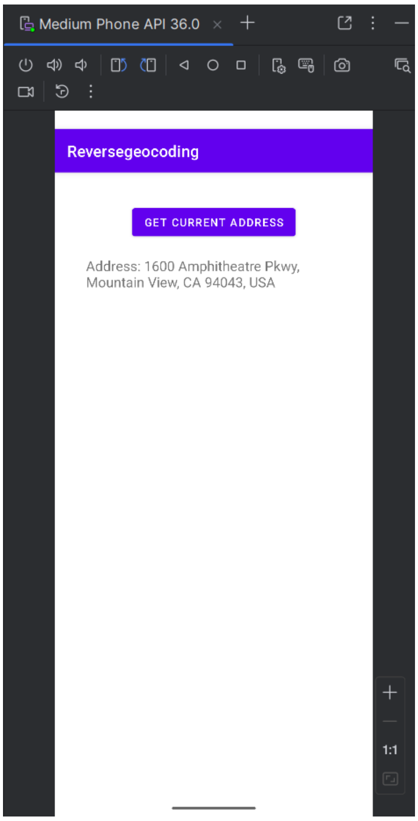
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:padding="20dp"
   android:orientation="vertical"
   android:gravity="center_horizontal"
   android:layout_width="match_parent"
   android:paddingTop="100dp"
   android:layout_height="match_parent">

   <Button
      android:layout_marginTop="100dp"
      android:id="@+id/btn_getAddress"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Get Current Address" />

   <TextView
      android:paddingTop="100dp"
      android:id="@+id/tv_address"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:text="Address will be shown here"
      android:padding="20dp"
      android:textSize="18sp" />
</LinearLayout>
```

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">

   <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.Reversegeocoding">
        <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>
```

Output:



Address: 1600 Amphitheatre Pkwy,
Mountain View, CA 94043, USA

7. Create an Android application that accepts longitude and latitude from the user and marks that location on google map.

MainActivity.java

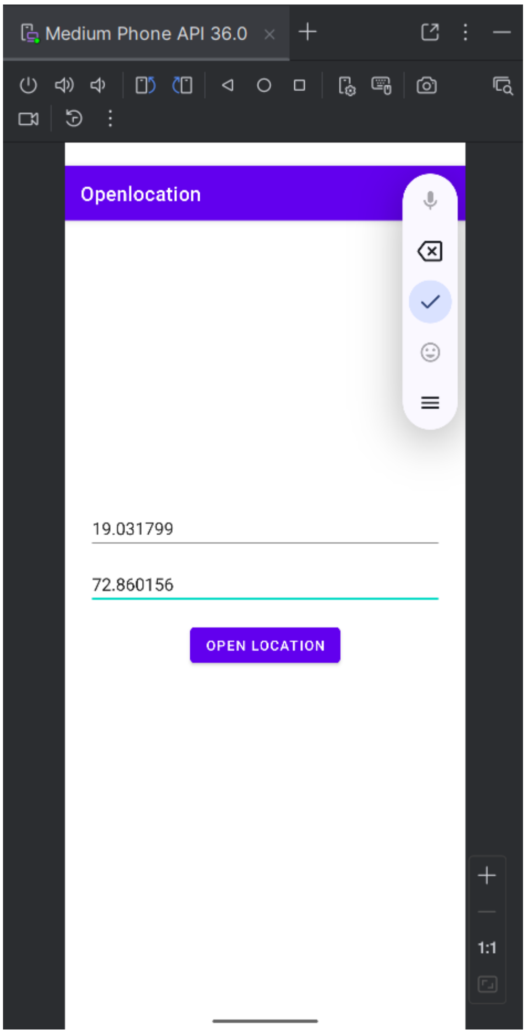
```
package com.example.openlocation;

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

public class MainActivity extends AppCompatActivity {

    EditText etLatitude, etLongitude;
    Button btnOpenMap;

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

        etLatitude = findViewById(R.id.etLatitude);
        etLongitude = findViewById(R.id.etLongitude);
        btnOpenMap = findViewById(R.id.btnOpenMap);

        btnOpenMap.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String latStr = etLatitude.getText().toString().trim();
                String lonStr = etLongitude.getText().toString().trim();

                if (TextUtils.isEmpty(latStr) || TextUtils.isEmpty(lonStr)) {
                    Toast.makeText(MainActivity.this, "Please enter latitude and longitude",
Toast.LENGTH_SHORT).show();
                    return;
                }

                double lat, lon;
                try {
                    lat = Double.parseDouble(latStr);
                    lon = Double.parseDouble(lonStr);
                } catch (NumberFormatException e) {
                    Toast.makeText(MainActivity.this, "Invalid input format",
Toast.LENGTH_SHORT).show();
                    return;
                }

                // Create geo URI
                String geoUriString = "geo:" + lat + "," + lon + "?q=" + lat + "," + lon +
"(Marker)";
                Uri geoUri = Uri.parse(geoUriString);

                // Create intent to open map app
                Intent intent = new Intent(Intent.ACTION_VIEW, geoUri);
                intent.setPackage("com.google.android.apps.maps"); // Optional: ensure
only Google Maps opens

                if (intent.resolveActivity(getPackageManager()) != null) {
                    startActivity(intent);
                } else {
                    Toast.makeText(MainActivity.this, "Google Maps app is not installed",
Toast.LENGTH_SHORT).show();
                }
            }
        });
    }
}
```

Activity_main.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:gravity="center_vertical|center_horizontal"
    android:orientation="vertical"
    android:padding="24dp"
```

```
  android:layout_width="match_parent"
  android:layout_height="match_parent">

  <EditText
    android:id="@+id/etLatitude"
    android:hint="Enter Latitude"
    android:inputType="numberDecimal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />

  <EditText
    android:id="@+id/etLongitude"
    android:hint="Enter Longitude"
    android:inputType="numberDecimal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="12dp" />

  <Button
    android:id="@+id/btnOpenMap"
    android:text="Open Location"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp" />

</LinearLayout>
```

Output :



19.031799

72.860156

Experiment No – 4

REST API integration

1. Create an Android application to demonstrate JSON data parsing using OkHttp. Use https://api.github.com/users JSON data.

Mainactivity.java

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

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

import java.io.IOException;

import okhttp3.Call;
import okhttp3.Callback;
import okhttp3.OkHttpClient;
import okhttp3.Request;
import okhttp3.Response;

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

public class MainActivity extends AppCompatActivity {

    private Button btnFetchUsers;
    private TextView tvUserList;
    private OkHttpClient client;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        btnFetchUsers = findViewById(R.id.btnFetchUsers);
        tvUserList = findViewById(R.id.tvUserList);

        client = new OkHttpClient();

        btnFetchUsers.setOnClickListener(v -> fetchGitHubUsers());
    }

    private void fetchGitHubUsers() {
        String url = "https://api.github.com/users";

        Request request = new Request.Builder()
                .url(url)
                .build();

        client.newCall(request).enqueue(new Callback() {
            @Override
            public void onFailure(Call call, IOException e) {
                runOnUiThread(() -> tvUserList.setText("Failed to fetch users"));
            }

            @Override
            public void onResponse(Call call, Response response) throws IOException {
                if (!response.isSuccessful()) {
                    runOnUiThread(() -> tvUserList.setText("Unexpected response: " +
response));
                    return;
                }

                String responseData = response.body().string();

                try {
                    JSONArray usersArray = new JSONArray(responseData);

                    StringBuilder userListBuilder = new StringBuilder();

                    for (int i = 0; i < usersArray.length(); i++) {
                        JSONObject userObj = usersArray.getJSONObject(i);
                        String login = userObj.getString("login");
                String url = userObj.getString("html_url");
                userListBuilder.append(login).append(": ").append(url).append("\n");
              }

              String userList = userListBuilder.toString();

              runOnUiThread(() -> tvUserList.setText(userList));

            } catch (JSONException e) {
              e.printStackTrace();
              runOnUiThread(() -> tvUserList.setText("Error parsing JSON"));
            }
          }
        });
    }
}
```

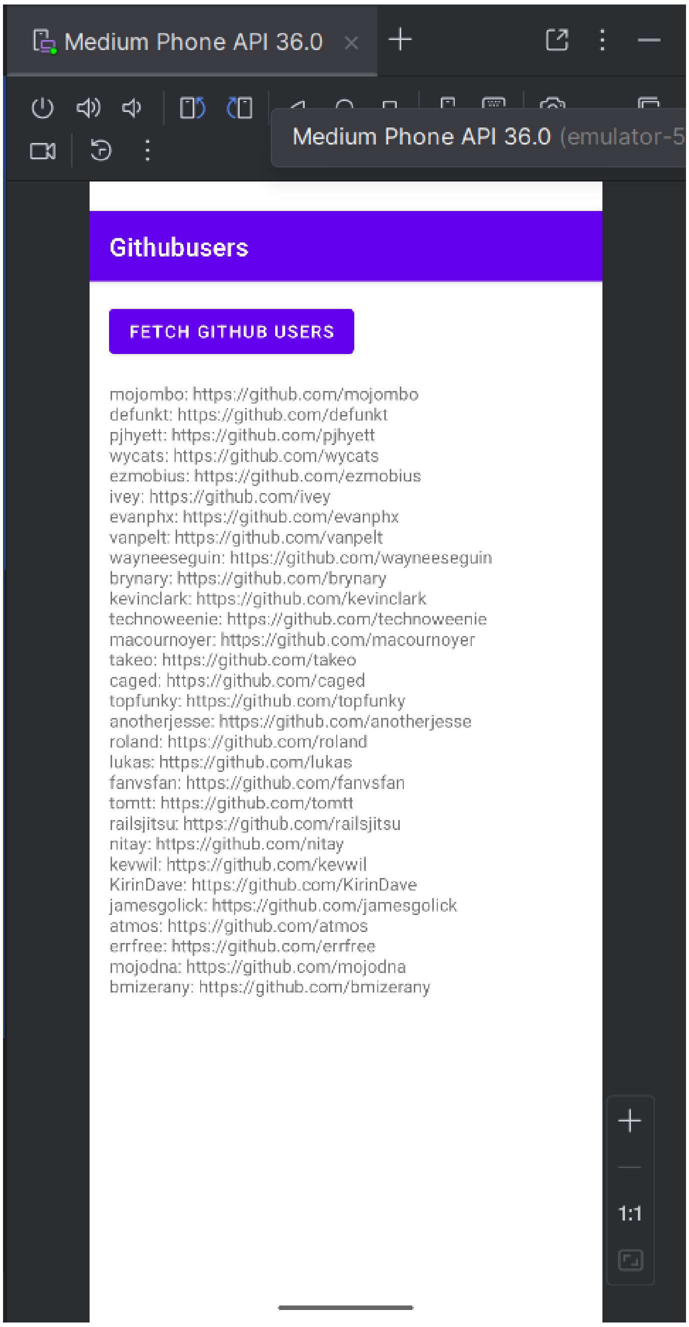
activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
  android:orientation="vertical"
  android:padding="16dp"
  android:layout_marginTop="80dp"
  android:layout_width="match_parent"
  android:layout_height="match_parent">

  <Button
    android:id="@+id/btnFetchUsers"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Fetch GitHub Users" />

  <TextView
    android:id="@+id/tvUserList"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:paddingTop="16dp" />
</LinearLayout>
```

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">

    <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.Githubusers">
        <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>
```

Output :



```
mojombo: https://github.com/mojombo
defunkt: https://github.com/defunkt
pjhyett: https://github.com/pjhyett
wycats: https://github.com/wycats
ezmobius: https://github.com/ezmobius
ivey: https://github.com/ivey
evanphx: https://github.com/evanphx
vanpelt: https://github.com/vanpelt
wayneeseguin: https://github.com/wayneeseguin
brynary: https://github.com/brynary
kevinclark: https://github.com/kevinclark
technoweenie: https://github.com/technoweenie
macournoyer: https://github.com/macournoyer
takeo: https://github.com/takeo
caged: https://github.com/caged
topfunky: https://github.com/topfunky
anotherjesse: https://github.com/anotherjesse
roland: https://github.com/roland
lukas: https://github.com/lukas
fanvsfan: https://github.com/fanvsfan
tomtt: https://github.com/tomtt
railsjitsu: https://github.com/railsjitsu
nitay: https://github.com/nitay
kevwil: https://github.com/kevwil
KirinDave: https://github.com/KirinDave
jamesgolick: https://github.com/jamesgolick
atmos: https://github.com/atmos
errfree: https://github.com/errfree
mojodna: https://github.com/mojodna
bmizerany: https://github.com/bmizerany
```

2 Create an Android application to demonstrate JSON data parsing using Volley. Use https://api.github.com/users JSON data.

Mainactivity.java

```
package com.example.githubusers;

import androidx.appcompat.app.AppCompatActivity;

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

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.JsonArrayRequest;
import com.android.volley.toolbox.Volley;

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

public class MainActivity extends AppCompatActivity {

    private Button btnFetchUsers;
    private TextView tvUserList;
    private RequestQueue requestQueue;

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

        btnFetchUsers = findViewById(R.id.btnFetchUsers);
        tvUserList = findViewById(R.id.tvUserList);

        requestQueue = Volley.newRequestQueue(this);
        btnFetchUsers.setOnClickListener(v -> fetchGitHubUsers());
    }

    private void fetchGitHubUsers() {
        String url = "https://api.github.com/users";

        JsonArrayRequest jsonArrayRequest = new JsonArrayRequest(
                Request.Method.GET, url, null,
                new Response.Listener<JSONArray>() {
                    @Override
                    public void onResponse(JSONArray response) {
                        StringBuilder userListBuilder = new StringBuilder();

                        try {
                            for (int i = 0; i < response.length(); i++) {
                                JSONObject userObj = response.getJSONObject(i);
                                String login = userObj.getString("login");
                                String htmlUrl = userObj.getString("html_url");
                                String userType = userObj.getString("type");
                                String avatarUrl = userObj.getString("avatar_url");

                                userListBuilder.append("Username: 
").append(login).append("\n")
                                        .append("Profile URL: ").append(htmlUrl).append("\n\n");
                            }

                        } catch (JSONException e) {
                            e.printStackTrace();
                            userListBuilder.append("Error parsing JSON");
                        }

                        tvUserList.setText(userListBuilder.toString());
                    }
                },
                new Response.ErrorListener() {
                    @Override
                    public void onErrorResponse(VolleyError error) {
                        tvUserList.setText("Failed to fetch data: " + error.getMessage());
                    }
```

```
            }
        );

        requestQueue.add(jsonArrayRequest);
    }
}
```

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:padding="16dp"
    android:layout_marginTop="80dp"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <Button
        android:id="@+id/btnFetchUsers"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Fetch GitHub Users" />

    <TextView
        android:id="@+id/tvUserList"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="16sp"
        android:paddingTop="16dp"
        android:paddingBottom="16dp" />
</LinearLayout>
```

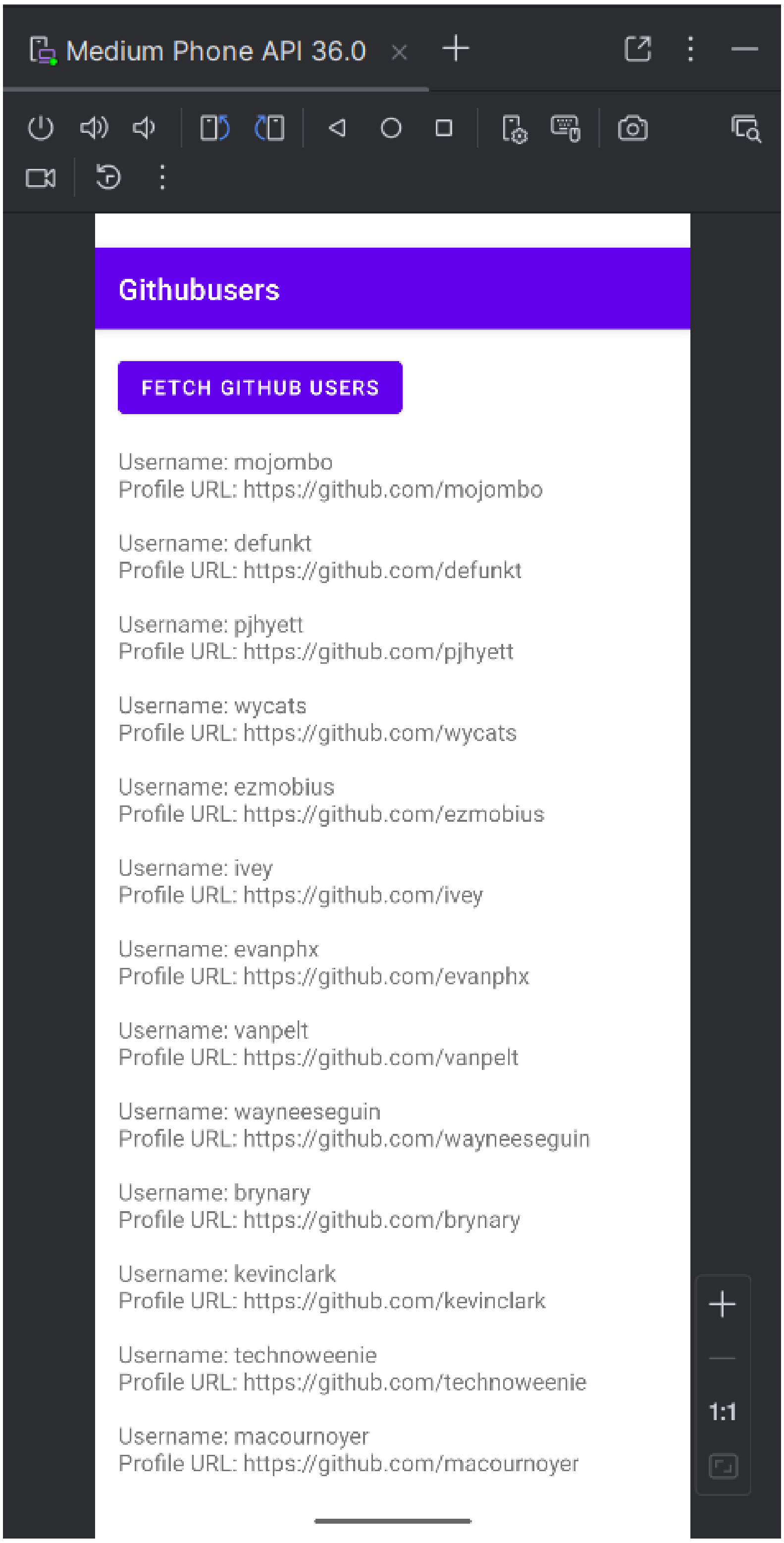
androidManifest.xml

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

    <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.Githubusers">
        <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>
```

Build.gradle.kts (:app) - Level

```
plugins {
    alias(libs.plugins.android.application)
}

android {
    namespace = "com.example.githubusers"
    compileSdk = 36

    defaultConfig {
        applicationId = "com.example.githubusers"
        minSdk = 24
        targetSdk = 36
        versionCode = 1
        versionName = "1.0"
        testInstrumentationRunner = "androidx.test.runner.AndroidJUnitRunner"
    }

    buildTypes {
        release {
            isMinifyEnabled = false
            proguardFiles(
                getDefaultProguardFile("proguard-android-optimize.txt"),
                "proguard-rules.pro"
            )
        }
    }
    compileOptions {
        sourceCompatibility = JavaVersion.VERSION_11
        targetCompatibility = JavaVersion.VERSION_11
    }
}

dependencies {
    implementation("com.android.volley:volley:1.2.1")
    implementation(libs.appcompat)
    implementation(libs.material)
    implementation(libs.activity)
    implementation(libs.constraintlayout)
    testImplementation(libs.junit)
    androidTestImplementation(libs.ext.junit)
    androidTestImplementation(libs.espresso.core)
}
```

Output :



Githubusers

FETCH GITHUB USERS

```
Username: mojombo
Profile URL: https://github.com/mojombo

Username: defunkt
Profile URL: https://github.com/defunkt

Username: pjhyett
Profile URL: https://github.com/pjhyett

Username: wycats
Profile URL: https://github.com/wycats

Username: ezmobius
Profile URL: https://github.com/ezmobius

Username: ivey
Profile URL: https://github.com/ivey

Username: evanphx
Profile URL: https://github.com/evanphx

Username: vanpelt
Profile URL: https://github.com/vanpelt

Username: wayneeseguin
Profile URL: https://github.com/wayneeseguin

Username: brynary
Profile URL: https://github.com/brynary

Username: kevinclark
Profile URL: https://github.com/kevinclark

Username: technoweenie
Profile URL: https://github.com/technoweenie

Username: macournoyer
Profile URL: https://github.com/macournoyer
```

3. Create an Android application to demonstrate JSON data parsing using Retrofit. Use https://api.github.com/users JSON data.

Mainactivity.java

```
package com.example.githubusers;

import androidx.appcompat.app.AppCompatActivity;

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

import java.util.List;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;
import retrofit2.Retrofit;
import retrofit2.converter.gson.GsonConverterFactory;

public class MainActivity extends AppCompatActivity {

    private Button btnFetchUsers;
    private TextView tvUserList;
    private GitHubApiService apiService;

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

        btnFetchUsers = findViewById(R.id.btnFetchUsers);
        tvUserList = findViewById(R.id.tvUserList);

        Retrofit retrofit = new Retrofit.Builder()
                .baseUrl("https://api.github.com/")
                .addConverterFactory(GsonConverterFactory.create())
                .build();
    apiService = retrofit.create(GitHubApiService.class);

    btnFetchUsers.setOnClickListener(v -> fetchGitHubUsers());
  }

  private void fetchGitHubUsers() {
    Call<List<GitHubUser>> call = apiService.getUsers();

    call.enqueue(new Callback<List<GitHubUser>>() {
      @Override
      public void onResponse(Call<List<GitHubUser>> call,
Response<List<GitHubUser>> response) {
        if (!response.isSuccessful()) {
          tvUserList.setText("Code: " + response.code());
          return;
        }

        List<GitHubUser> users = response.body();
        if (users != null) {
          StringBuilder userListBuilder = new StringBuilder();
          for (GitHubUser user : users) {
            userListBuilder.append("Username:
").append(user.getLogin()).append("\n");
            userListBuilder.append("Profile URL:
").append(user.getHtmlUrl()).append("\n\n");
          }
          tvUserList.setText(userListBuilder.toString());
        }
      }

      @Override
      public void onFailure(Call<List<GitHubUser>> call, Throwable t) {
        Toast.makeText(MainActivity.this, "Failed to load users",
Toast.LENGTH_SHORT).show();
      }
    });
  }
}
```

githubUser.java

```
package com.example.githubusers;

import com.google.gson.annotations.SerializedName;

public class GitHubUser {
   @SerializedName("login")
   private String login;

   @SerializedName("html_url")
   private String htmlUrl;

   @SerializedName("avatar_url")
   private String avatarUrl;

   // Getters
   public String getLogin() {
      return login;
   }

   public String getHtmlUrl() {
      return htmlUrl;
   }

   public String getAvatarUrl() {
      return avatarUrl;
   }
}
```

GithubApiService.java

```
package com.example.githubusers;

import java.util.List;

import retrofit2.Call;
import retrofit2.http.GET;

public interface GitHubApiService {
   @GET("users")
```

```
    Call<List<GitHubUser>> getUsers();
}
```

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:padding="16dp"
    android:layout_width="match_parent"
    android:layout_marginTop="80dp"
    android:layout_height="match_parent">

    <Button
        android:id="@+id/btnFetchUsers"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Fetch GitHub Users" />

    <TextView
        android:id="@+id/tvUserList"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:paddingTop="16dp" />
</LinearLayout>
```

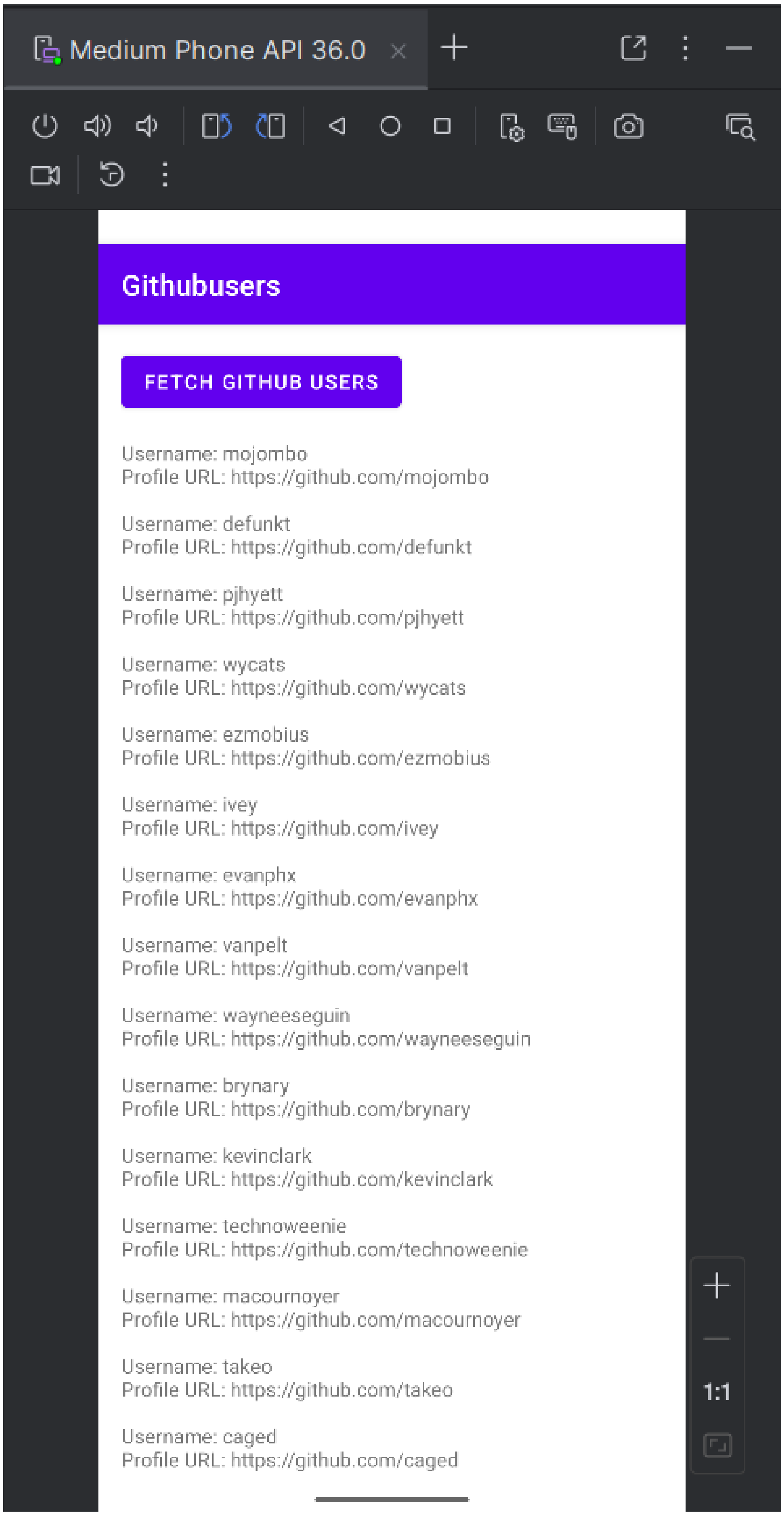
AndroidManifest.xml

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

    <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.Githubusers">
        <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>
```

Build.gradle.kts (:app) – Level

```
plugins {
    alias(libs.plugins.android.application)
}

android {
    namespace = "com.example.githubusers"
    compileSdk = 36

    defaultConfig {
        applicationId = "com.example.githubusers"
        minSdk = 24
        targetSdk = 36
        versionCode = 1
        versionName = "1.0"

        testInstrumentationRunner = "androidx.test.runner.AndroidJUnitRunner"
    }

    buildTypes {
        release {
            isMinifyEnabled = false
            proguardFiles(
                getDefaultProguardFile("proguard-android-optimize.txt"),
                "proguard-rules.pro"
            )
        }
    }
    compileOptions {
        sourceCompatibility = JavaVersion.VERSION_11
        targetCompatibility = JavaVersion.VERSION_11
    }
}

dependencies {
    implementation("com.squareup.retrofit2:retrofit:2.9.0")
    implementation("com.squareup.retrofit2:converter-gson:2.9.0")

    implementation(libs.appcompat)
    implementation(libs.material)
    implementation(libs.activity)
    implementation(libs.constraintlayout)
    testImplementation(libs.junit)
    androidTestImplementation(libs.ext.junit)
    androidTestImplementation(libs.espresso.core)
}
```

Output :



Githubusers

FETCH GITHUB USERS

```
Username: mojombo
Profile URL: https://github.com/mojombo

Username: defunkt
Profile URL: https://github.com/defunkt

Username: pjhyett
Profile URL: https://github.com/pjhyett

Username: wycats
Profile URL: https://github.com/wycats

Username: ezmobius
Profile URL: https://github.com/ezmobius

Username: ivey
Profile URL: https://github.com/ivey

Username: evanphx
Profile URL: https://github.com/evanphx

Username: vanpelt
Profile URL: https://github.com/vanpelt

Username: wayneeseguin
Profile URL: https://github.com/wayneeseguin

Username: brynary
Profile URL: https://github.com/brynary

Username: kevinclark
Profile URL: https://github.com/kevinclark

Username: technoweenie
Profile URL: https://github.com/technoweenie

Username: macournoyer
Profile URL: https://github.com/macournoyer

Username: takeo
Profile URL: https://github.com/takeo

Username: caged
Profile URL: https://github.com/caged
```

Experiment No - 5:

Introduction to Dart and Flutter

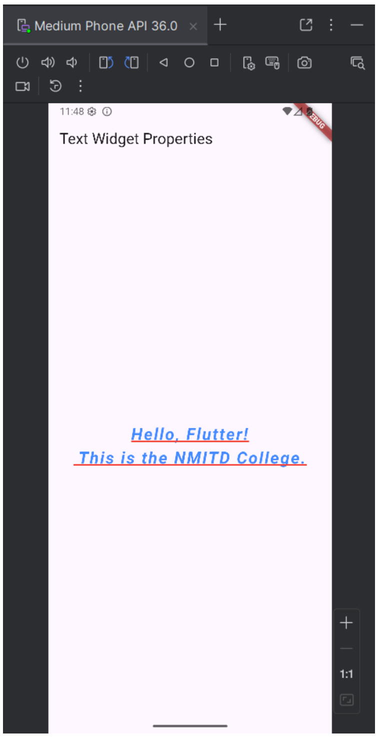
1. Write a Flutter program to demonstrate the Text widget and its properties.

Code :

Main.dart

```
import 'package:flutter/material.dart';

void main() {
 runApp(TextDemoApp());
}

class TextDemoApp extends StatelessWidget {
 @override
 Widget build(BuildContext context) {
  return MaterialApp(
   title: 'Text Widget Demo',
   home: Scaffold(
    appBar: AppBar(
     title: Text('Text Widget Properties'),
    ),
    body: Center(
     child: Text(
      'Hello, Flutter!\nThis is a Text widget demo.',
      textAlign: TextAlign.center,
      style: TextStyle(
       color: Colors.blueAccent,
              fontSize: 24,
              fontWeight: FontWeight.bold,
              fontStyle: FontStyle.italic,
              letterSpacing: 2.0,
              decoration: TextDecoration.underline,
              decorationColor: Colors.red,
              decorationThickness: 2,
            ),
            maxLines: 3,
            overflow: TextOverflow.ellipsis,
          ),
        ),
      ),
    );
  }
}
```

Output :



Text Widget Properties

Hello, Flutter!
This is the NMITD College.

2. Write a Flutter program to display dog names demonstrating stateless widget and column widgets.

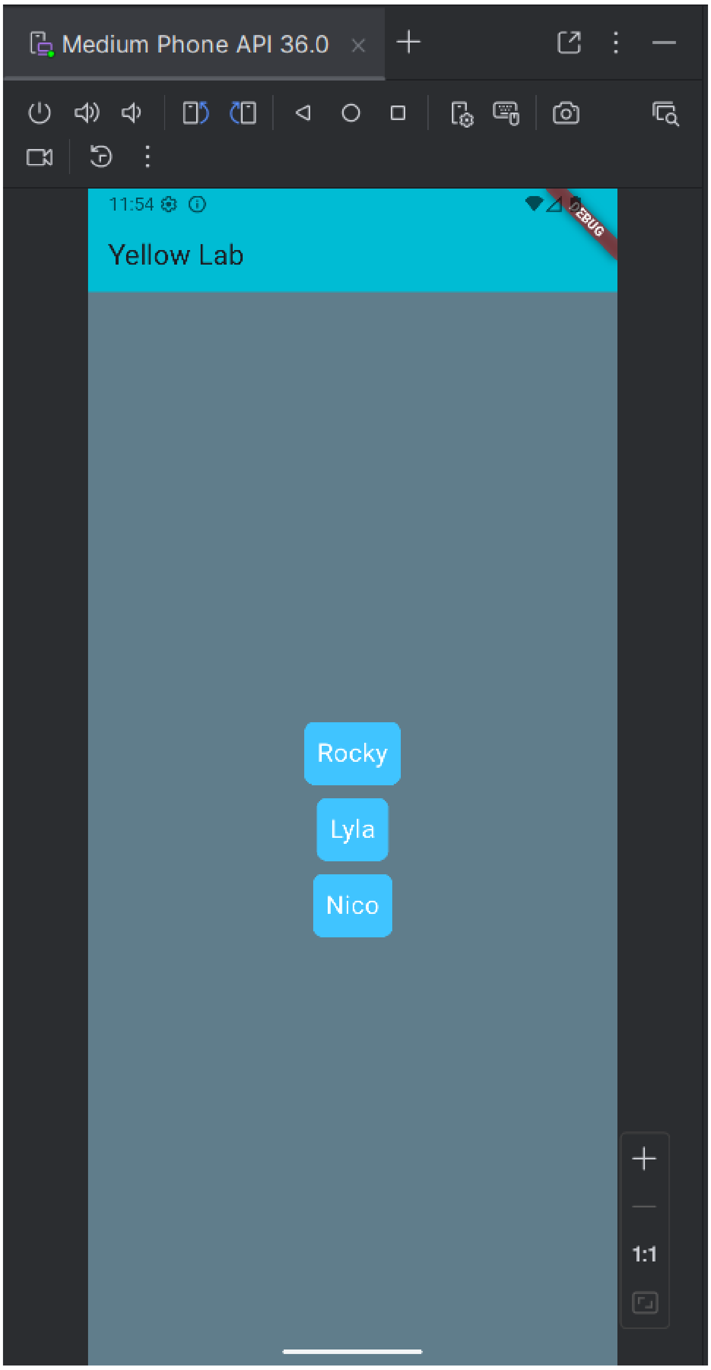
Code :

Main.dart

```
import 'package:flutter/material.dart';
void main() {
  runApp(DogApp());
}

class DogApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'My Dog App',
      home: Scaffold(
        backgroundColor: Colors.blueGrey,
        appBar: AppBar(
          backgroundColor: Colors.cyan,
          title: Text('Yellow Lab'),
        ),
        body: Center(
          child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              DogName('Rocky'),
              SizedBox(height: 10.0),
              DogName('Lyla'),
              SizedBox(height: 10.0),
              DogName('Nico'),
            ],
          ),
        ),
      ),
    );
  }
}

class DogName extends StatelessWidget {
  final String name;
```

```
 const DogName(this.name, {Key? key}) : super(key: key);

 @override
 Widget build(BuildContext context) {
  return DecoratedBox(
   decoration: BoxDecoration(
    color: Colors.lightBlueAccent,
    borderRadius: BorderRadius.circular(8),
   ),
   child: Padding(
    padding: const EdgeInsets.all(10.0),
    child: Text(
     name,
     style: TextStyle(
      fontSize: 20,
      color: Colors.white,
     ),
    ),
   ),
  );
 }
}
```

Output :


Medium Phone API 36.0
11:54
DEBUG
Yellow Lab
Rocky
Lyla
Nico
1:1

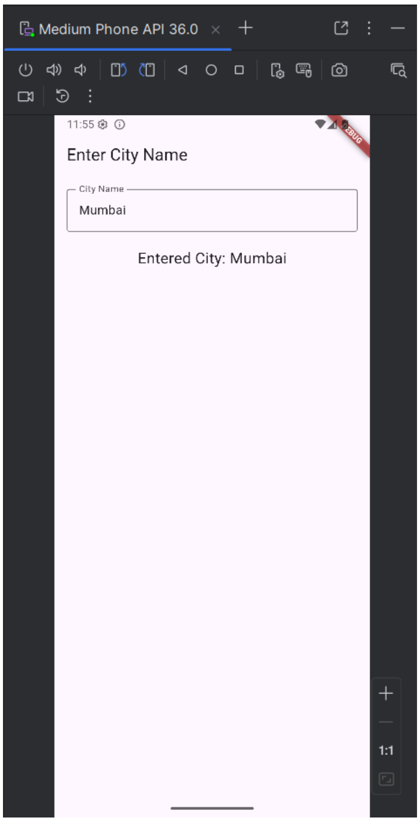
3. Write a Flutter program that allows the user to enter a city in a text field and displays the city name demonstrating stateful widget.

Code :

Main.dart

```
import 'package:flutter/material.dart';

void main() {
  runApp(CityInputApp());
}

class CityInputApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'City Input Demo',
      home: Scaffold(
        appBar: AppBar(
          title: Text('Enter City Name'),
        ),
        body: CityInputWidget(),
      ),
    );
  }
}

class CityInputWidget extends StatefulWidget {
  @override
  _CityInputWidgetState createState() => _CityInputWidgetState();
}

class _CityInputWidgetState extends State<CityInputWidget> {
  String _cityName = '';

  @override
  Widget build(BuildContext context) {
    return Padding(
      padding: const EdgeInsets.all(16.0),
      child: Column(
        children: [
          TextField(
            decoration: InputDecoration(
              labelText: 'City Name',
              border: OutlineInputBorder(),
            ),
            onChanged: (text) {
              setState(() {
                _cityName = text;
              });
            },
          ),
          SizedBox(height: 20),
          Text(
            'Entered City: $_cityName',
            style: TextStyle(fontSize: 20),
          ),
        ],
      ),
    );
  }
}
```

Output :



Enter City Name

City Name

Mumbai

Entered City: Mumbai

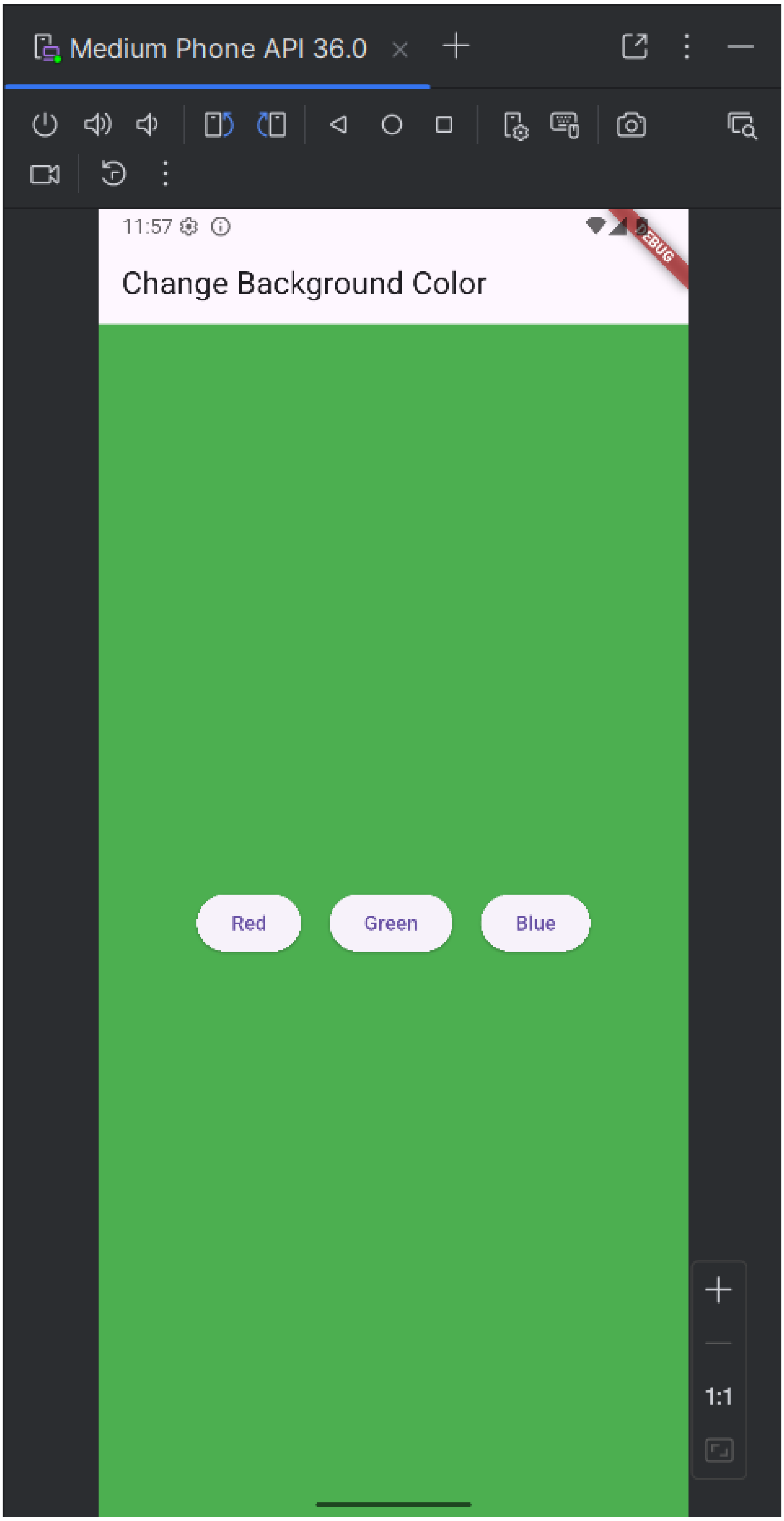
4 - Write a Flutter program to change the background color demonstrating stateful widget.

Code:

Main.dart

```
import 'package:flutter/material.dart';

void main() {
  runApp(BackgroundColorChangerApp());
}

class BackgroundColorChangerApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Background Color Changer',
      home: BackgroundColorChanger(),
    );
  }
}

class BackgroundColorChanger extends StatefulWidget {
  @override
  _BackgroundColorChangerState createState() =>
_BackgroundColorChangerState();
}

class _BackgroundColorChangerState extends State<BackgroundColorChanger> {
  Color _backgroundColor = Colors.white;

  void _changeColor(Color color) {
    setState(() {
      _backgroundColor = color;
    });
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: _backgroundColor,
      appBar: AppBar(
        title: Text('Change Background Color'),
      ),
      body: Center(
        child: Row(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            ElevatedButton(
              onPressed: () => _changeColor(Colors.red),
              child: Text('Red'),
            ),
            SizedBox(width: 20),
            ElevatedButton(
              onPressed: () => _changeColor(Colors.green),
              child: Text('Green'),
            ),
            SizedBox(width: 20),
            ElevatedButton(
              onPressed: () => _changeColor(Colors.blue),
              child: Text('Blue'),
            ),
          ],
        ),
      ),
    );
  }
}
```

Output:



Change Background Color

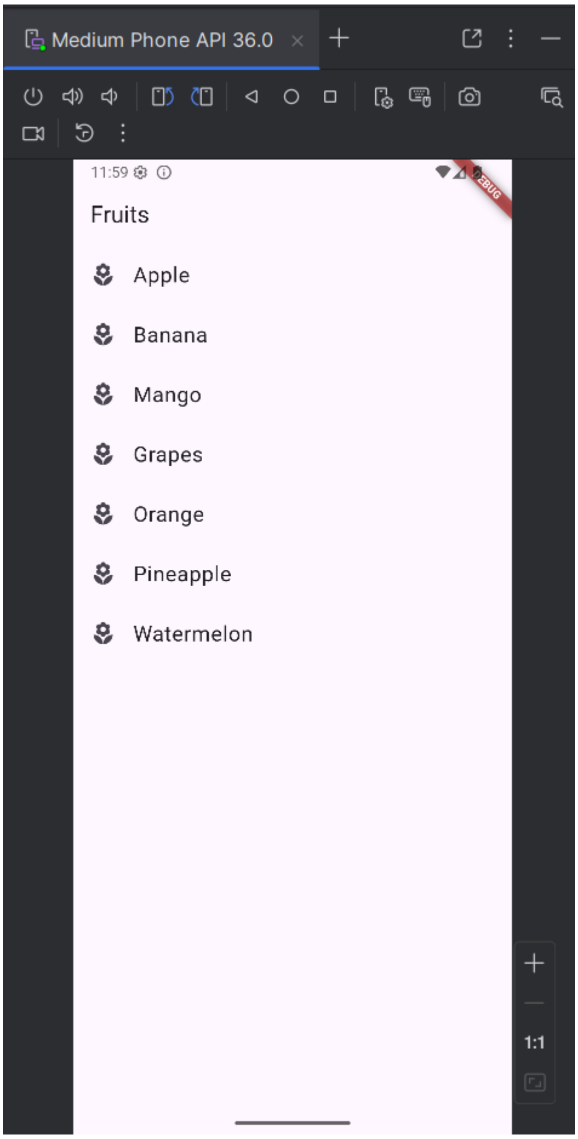
5 - Write a Flutter program to display a fruit list using ListView.

Code :

Main.dart

```
import 'package:flutter/material.dart';

void main() {
  runApp(FruitListApp());
}

class FruitListApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Fruit List Demo',
      home: Scaffold(
        appBar: AppBar(
          title: Text('Fruits'),
        ),
        body: FruitList(),
      ),
    );
  }
}

class FruitList extends StatelessWidget {
  final List<String> fruits = [
    'Apple',
    'Banana',
    'Mango',
    'Grapes',
    'Orange',
    'Pineapple',
    'Watermelon',
  ];

  @override
  Widget build(BuildContext context) {
    return ListView.builder(
      itemCount: fruits.length,
      itemBuilder: (context, index) {
        return ListTile(
          leading: Icon(Icons.local_florist),
          title: Text(
            fruits[index],
            style: TextStyle(fontSize: 20),
          ),
        );
      },
    );
  }
}
```

Output:



Fruits

Apple

Banana

Mango

Grapes

Orange

Pineapple

Watermelon

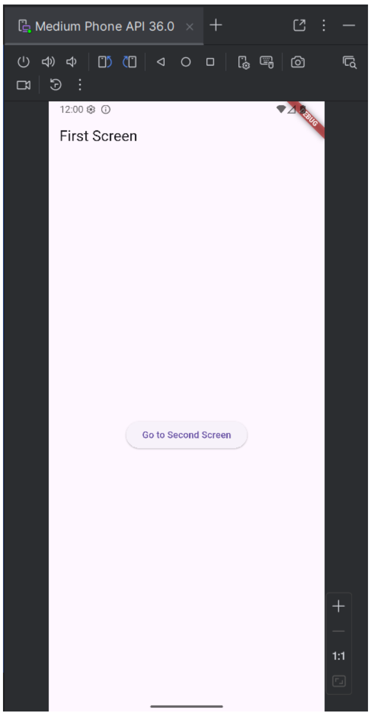
6. Write a Flutter program to demonstrate navigation where the user should be navigated from the first screen to the second screen.

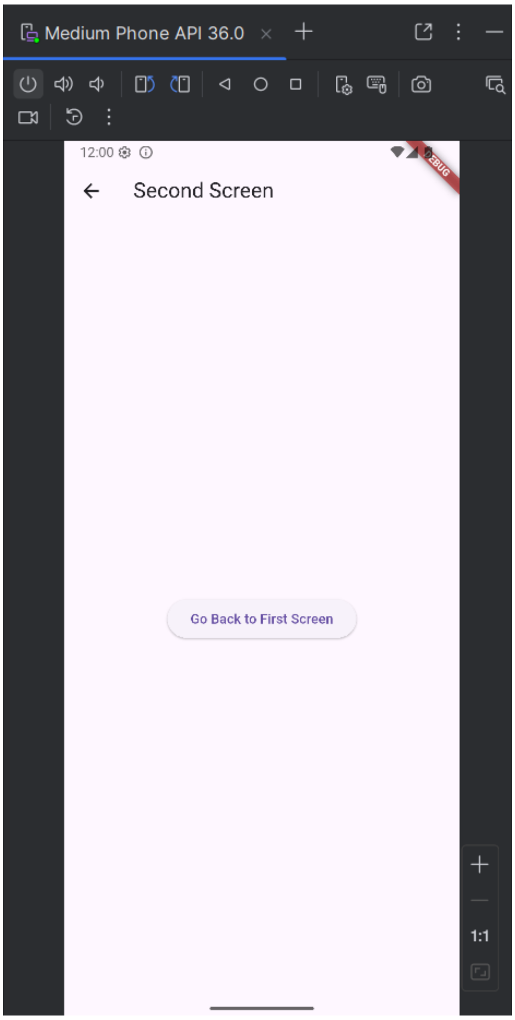
Code :

Main.dart

```
import 'package:flutter/material.dart';

void main() {
  runApp(NavigationDemoApp());
}

class NavigationDemoApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Navigation Demo',
      home: FirstScreen(),
    );
  }
}

class FirstScreen extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('First Screen'),
      ),
      body: Center(
        child: ElevatedButton(
          child: Text('Go to Second Screen'),
          onPressed: () {
            Navigator.push(
              context,
              MaterialPageRoute(builder: (context) => SecondScreen()),
            );
          },
        ),
      ),
    );
  }
}

class SecondScreen extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Second Screen'),
      ),
      body: Center(
        child: ElevatedButton(
          child: Text('Go Back to First Screen'),
          onPressed: () {
            Navigator.pop(context);
          },
        ),
      ),
    );
  }
}
```

<u>**Output :**</u>





Second Screen

Go Back to First Screen

7 - Write a Flutter program to design a Login form using TextField, Check Box, Buttons, Drop down, Switch, etc.

Code :

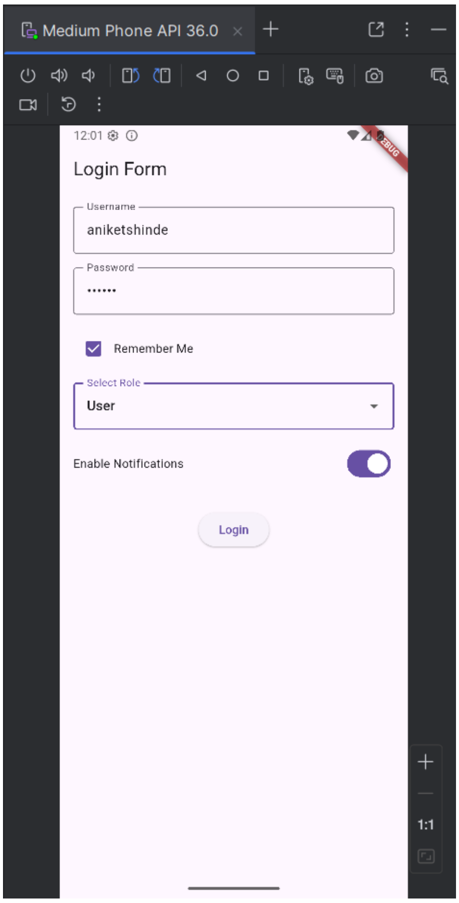
Main.dart

```
import 'package:flutter/material.dart';

void main() {
  runApp(LoginFormApp());
}

class LoginFormApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Login Form Demo',
      home: Scaffold(
        appBar: AppBar(
          title: Text('Login Form'),
        ),
        body: LoginForm(),
      ),
    );
  }
}

class LoginForm extends StatefulWidget {
  @override
  _LoginFormState createState() => _LoginFormState();
}

class _LoginFormState extends State<LoginForm> {
  final _usernameController = TextEditingController();
  final _passwordController = TextEditingController();

  bool _rememberMe = false;
  bool _isSwitchOn = false;

  String _selectedRole = 'User';
```

```
final List<String> _roles = ['User', 'Admin', 'Guest'];

@override
Widget build(BuildContext context) {
 return Padding(
  padding: const EdgeInsets.all(16.0),
  child: SingleChildScrollView(
   child: Column(
    children: [
     TextField(
      controller: _usernameController,
      decoration: InputDecoration(
       labelText: 'Username',
       border: OutlineInputBorder(),
      ),
     ),
     SizedBox(height: 16),
     TextField(
      controller: _passwordController,
      obscureText: true, // Hides password input
      decoration: InputDecoration(
       labelText: 'Password',
       border: OutlineInputBorder(),
      ),
     ),
     SizedBox(height: 16),
     Row(
      children: [
       Checkbox(
        value: _rememberMe,
        onChanged: (bool? value) {
         setState(() {
          _rememberMe = value ?? false;
         });
        },
       ),
       Text('Remember Me'),
      ],
     ),
     SizedBox(height: 16),
     DropdownButtonFormField<String>(
              value: _selectedRole,
              items: _roles
                  .map((role) =>
                  DropdownMenuItem(value: role, child: Text(role)))
                  .toList(),
              onChanged: (String? newValue) {
                setState(() {
                  _selectedRole = newValue ?? _selectedRole;
                });
              },
              decoration: InputDecoration(
                labelText: 'Select Role',
                border: OutlineInputBorder(),
              ),
            ),
            SizedBox(height: 16),
            Row(
              mainAxisAlignment: MainAxisAlignment.spaceBetween,
              children: [
                Text('Enable Notifications'),
                Switch(
                  value: _isSwitchOn,
                  onChanged: (bool value) {
                    setState(() {
                      _isSwitchOn = value;
                    });
                  },
                ),
              ],
            ),
            SizedBox(height: 30),
            ElevatedButton(
              onPressed: () {
                // For demonstration, shows input values in a dialog
                showDialog(
                  context: context,
                  builder: (context) => AlertDialog(
                    title: Text('Login Data'),
                    content: Text(
                        'Username: ${_usernameController.text}\nPassword:
${_passwordController.text}\nRemember Me: $_rememberMe\nRole:
```

```
$_selectedRole\nNotifications: $_isSwitchOn'),
                  actions: [
                    TextButton(
                      onPressed: () => Navigator.pop(context),
                      child: Text('OK'),
                    )
                  ],
                ),
              );
            },
            child: Text('Login'),
          ),
        ],
      ),
    ),
  );
 }

 @override
 void dispose() {
   _usernameController.dispose();
   _passwordController.dispose();
   super.dispose();
 }
}
```

Output :



Login Form

Username
aniketshinde

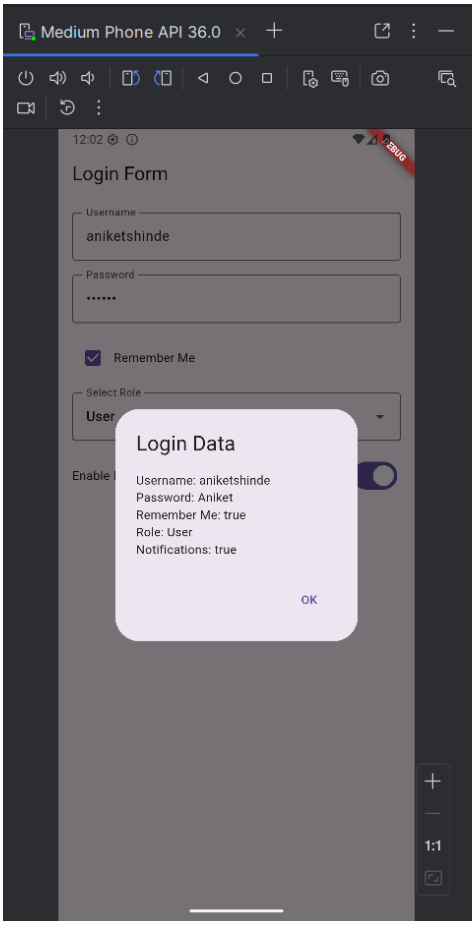
Password
••••••

☑ Remember Me

Select Role
User

Enable Notifications

Login



Login Form

Username
aniketshinde

Password
••••••

☑ Remember Me

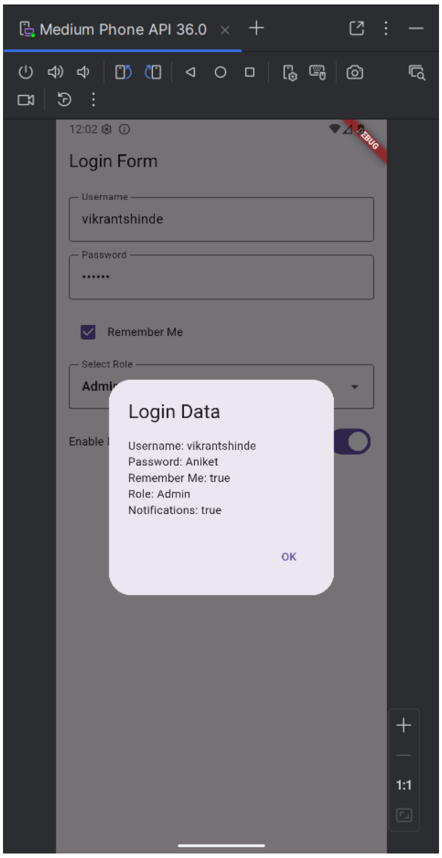
Select Role
User

Enable

Login Data

Username: aniketshinde
Password: Aniket
Remember Me: true
Role: User
Notifications: true

OK



Login Form

Username: vikrantshinde

Password: ••••••

☑ Remember Me

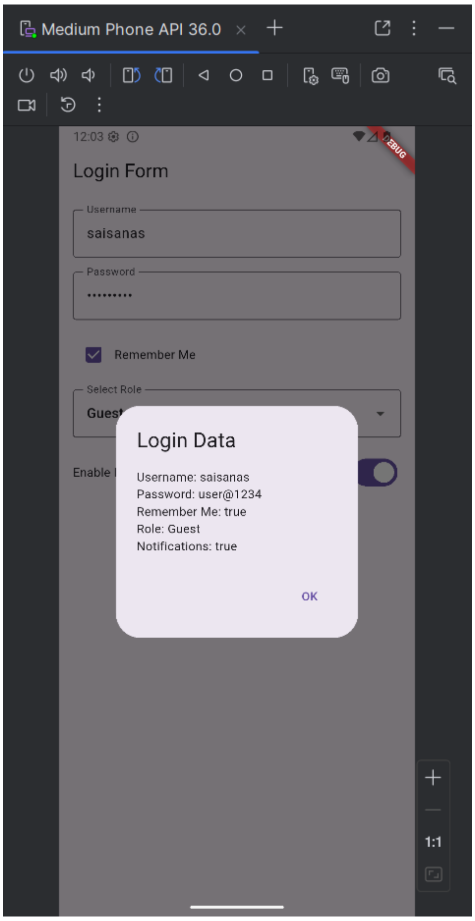
Select Role: Admi

Enable

Login Data

Username: vikrantshinde
Password: Aniket
Remember Me: true
Role: Admin
Notifications: true

OK



Login Form

Username: saisanas

Password: •••••••••

☑ Remember Me

Select Role: Guest

Enable

Login Data

Username: saisanas
Password: user@1234
Remember Me: true
Role: Guest
Notifications: true

OK

Experiment No - 6

Data Handling in Flutter

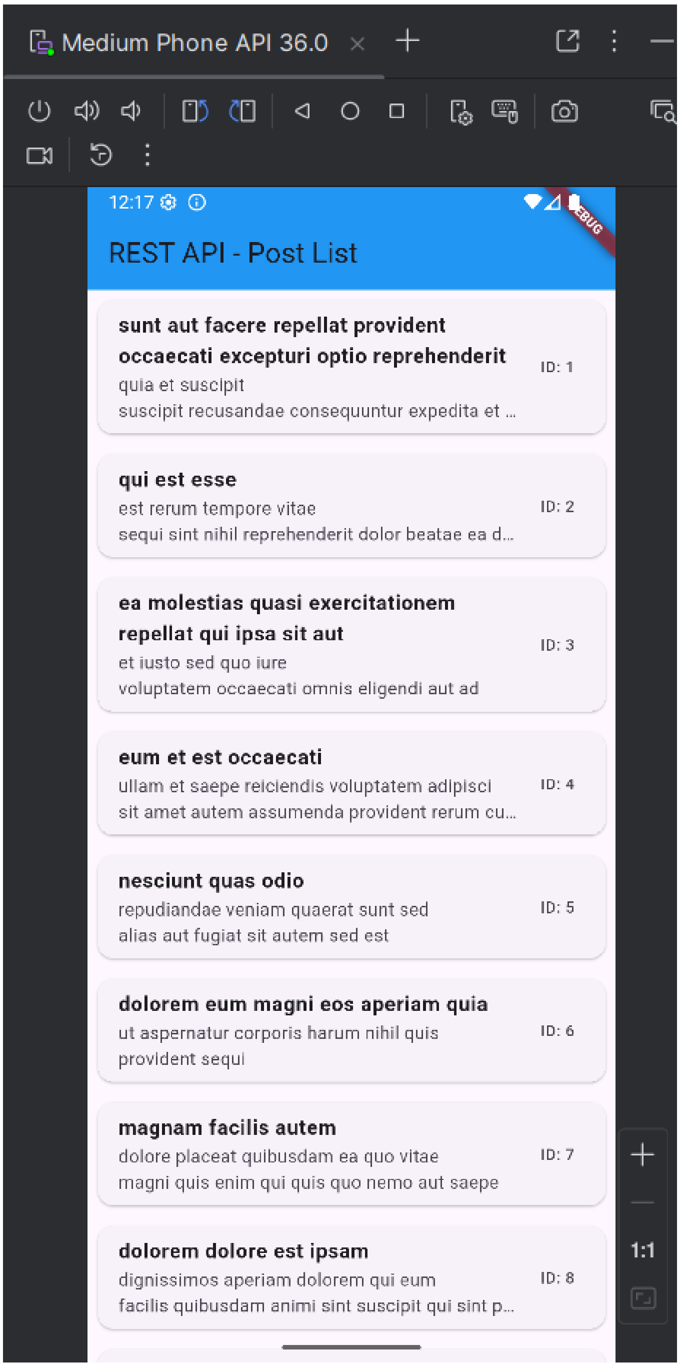
1. Write a Flutter program based on REST API to fetch data.

Code :

Main.dart

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

void main() {
  runApp(const MaterialApp(
    home: HomePage(),
  ));
}

class HomePage extends StatefulWidget {
  const HomePage({Key? key}) : super(key: key);

  @override
  HomePageState createState() => HomePageState();
}

class HomePageState extends State<HomePage> {
  late List<dynamic> data;
  bool isLoading = true;
  String errorMessage = '';

  Future<String> getData() async {
    try {
      var response = await http.get(
        Uri.parse("https://jsonplaceholder.typicode.com/posts"),
        headers: {
          "Accept": "application/json"
        },
      );

      if (response.statusCode == 200) {
        setState(() {
          data = json.decode(response.body);
          isLoading = false;
          errorMessage = '';
        });
        return "Success!";
      } else {
        setState(() {
          isLoading = false;
          errorMessage = 'Failed to load data: ${response.statusCode}';
        });
        return "Error: ${response.statusCode}";
      }
    } catch (e) {
      setState(() {
        isLoading = false;
        errorMessage = 'Error: $e';
      });
      return "Error: $e";
    }
  }

  @override
  void initState() {
    super.initState();
    getData();
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text("REST API - Post List"),
        backgroundColor: Colors.blue,
      ),
      body: isLoading
          ? const Center(
      child: CircularProgressIndicator(),
     )
       : errorMessage.isNotEmpty
       ? Center(
      child: Text(
       errorMessage,
       style: const TextStyle(color: Colors.red, fontSize: 16),
      ),
     )
       : ListView.builder(
      itemCount: data.length,
      itemBuilder: (BuildContext context, int index) {
       return Card(
        margin: const EdgeInsets.all(8.0),
        child: ListTile(
         title: Text(
          data[index]["title"] ?? "No Title",
          style: const TextStyle(fontWeight: FontWeight.bold),
         ),
         subtitle: Text(
          data[index]["body"] ?? "No Body",
          maxLines: 2,
          overflow: TextOverflow.ellipsis,
         ),
         trailing: Text(
          "ID: ${data[index]["id"]}",
          style: const TextStyle(fontSize: 12),
         ),
        ),
       );
      },
     ),
    );
  }
}
```

Output :



REST API - Post List

sunt aut facere repellat provident occaecati excepturi optio reprehenderit
quia et suscipit
suscipit recusandae consequuntur expedita et ...
ID: 1

qui est esse
est rerum tempore vitae
sequi sint nihil reprehenderit dolor beatae ea d...
ID: 2

ea molestias quasi exercitationem repellat qui ipsa sit aut
et iusto sed quo iure
voluptatem occaecati omnis eligendi aut ad
ID: 3

eum et est occaecati
ullam et saepe reiciendis voluptatem adipisci
sit amet autem assumenda provident rerum cu...
ID: 4

nesciunt quas odio
repudiandae veniam quaerat sunt sed
alias aut fugiat sit autem sed est
ID: 5

dolorem eum magni eos aperiam quia
ut aspernatur corporis harum nihil quis
provident sequi
ID: 6

magnam facilis autem
dolore placeat quibusdam ea quo vitae
magni quis enim qui quis quo nemo aut saepe
ID: 7

dolorem dolore est ipsam
dignissimos aperiam dolorem qui eum
facilis quibusdam animi sint suscipit qui sint p...
ID: 8

2. Write a Flutter program to demonstrate JSON serialization and deserialization.

Code :

Main.dart

```
import 'dart:convert';
import 'package:flutter/material.dart';
import 'usermodel.dart';

void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'JSON Serialization & Deserialization',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: const HomePage(),
    );
  }
}

class HomePage extends StatefulWidget {
  const HomePage({Key? key}) : super(key: key);

  @override
  State<HomePage> createState() => _HomePageState();
}

class _HomePageState extends State<HomePage> {
  final UserModel userObject = UserModel(
    id: "1",
    fullname: "Aniket Shinde",
    email: "aniketshinde@gmail.com",
  );

  final String userJSON =
      '{"id": "1", "fullname": "Aniket Shinde", "email": "aniketshinde@gmail.com"}';

  String serializationResult = '';
  String deserializationResult = '';

  void serializeUser() {
    try {
      Map<String, dynamic> userMap = userObject.toMap();
      String json = jsonEncode(userMap);

      setState(() {
        serializationResult = json;
      });

      debugPrint("Serialization Result: $json");
    } catch (e) {
      setState(() {
        serializationResult = "Error: ${e.toString()}";
      });
    }
  }

  void deserializeUser() {
    try {
      var decode = jsonDecode(userJSON);
      Map<String, dynamic> userMap = decode;
      UserModel newUser = UserModel.fromMap(userMap);

      setState(() {
        deserializationResult =
        "ID: ${newUser.id}\nName: ${newUser.fullname}\nEmail: ${newUser.email}";
      });

      debugPrint("Deserialization Result: ${newUser.toString()}");
    } catch (e) {
      setState(() {
        deserializationResult = "Error: ${e.toString()}";
```

```
   });
  }
 }

 @override
 Widget build(BuildContext context) {
  return Scaffold(
   appBar: AppBar(
    title: const Text("JSON Serialization & Deserialization"),
    backgroundColor: Colors.blue,
   ),
   body: SingleChildScrollView(
    child: Padding(
     padding: const EdgeInsets.all(16.0),
     child: Column(
      crossAxisAlignment: CrossAxisAlignment.center,
      children: <Widget>[
       const SizedBox(height: 20),
       Row(
        mainAxisAlignment: MainAxisAlignment.center,
        children: <Widget>[
         ElevatedButton(
          onPressed: serializeUser,
          style: ElevatedButton.styleFrom(
           backgroundColor: Colors.green,
           padding: const EdgeInsets.symmetric(
            horizontal: 30,
            vertical: 15,
           ),
          ),
          child: const Text(
           "Serialize",
           style: TextStyle(fontSize: 16, color: Colors.white),
          ),
         ),
         const SizedBox(width: 20),
         ElevatedButton(
          onPressed: deserializeUser,
          style: ElevatedButton.styleFrom(
           backgroundColor: Colors.orange,
        padding: const EdgeInsets.symmetric(
          horizontal: 30,
          vertical: 15,
        ),
      ),
      child: const Text(
        "Deserialize",
        style: TextStyle(fontSize: 16, color: Colors.white),
      ),
    ),
  ],
),
const SizedBox(height: 40),
Card(
  elevation: 5,
  child: Padding(
    padding: const EdgeInsets.all(16.0),
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        const Text(
          "Serialization Result:",
          style: TextStyle(
            fontSize: 16,
            fontWeight: FontWeight.bold,
            color: Colors.green,
          ),
        ),
        const SizedBox(height: 10),
        Container(
          width: double.infinity,
          padding: const EdgeInsets.all(12),
          decoration: BoxDecoration(
            border: Border.all(color: Colors.grey),
            borderRadius: BorderRadius.circular(8),
            color: Colors.grey[100],
          ),
          child: Text(
            serializationResult.isEmpty
```

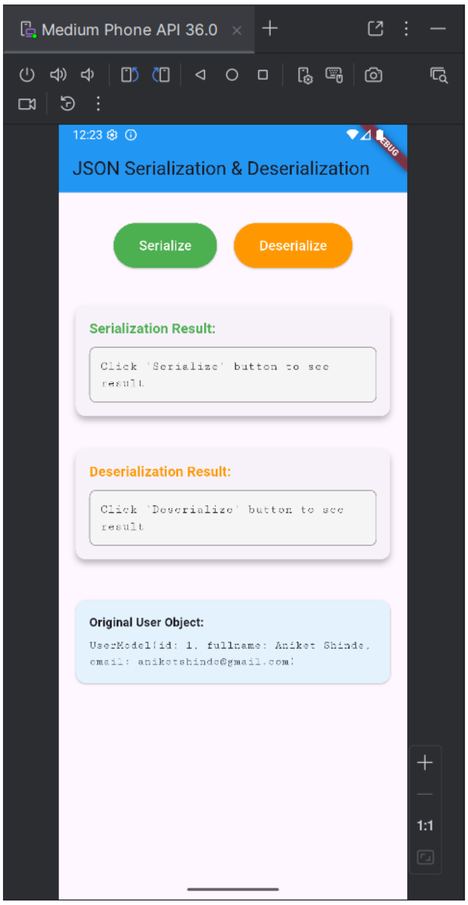
```
              ? "Click 'Serialize' button to see result"
              : serializationResult,
            style: const TextStyle(
              fontSize: 14,
              fontFamily: 'Courier',
            ),
          ),
        ),
      ],
    ),
  ),
),
const SizedBox(height: 30),
Card(
  elevation: 5,
  child: Padding(
    padding: const EdgeInsets.all(16.0),
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        const Text(
          "Deserialization Result:",
          style: TextStyle(
            fontSize: 16,
            fontWeight: FontWeight.bold,
            color: Colors.orange,
          ),
        ),
        const SizedBox(height: 10),
        Container(
          width: double.infinity,
          padding: const EdgeInsets.all(12),
          decoration: BoxDecoration(
            border: Border.all(color: Colors.grey),
            borderRadius: BorderRadius.circular(8),
            color: Colors.grey[100],
          ),
          child: Text(
            deserializationResult.isEmpty
```

```
                ? "Click 'Deserialize' button to see result"
                : deserializationResult,
            style: const TextStyle(
              fontSize: 14,
              fontFamily: 'Courier',
            ),
          ),
        ),
      ],
    ),
  ),
),
const SizedBox(height: 40),
Card(
  color: Colors.blue[50],
  child: Padding(
    padding: const EdgeInsets.all(16.0),
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        const Text(
          "Original User Object:",
          style: TextStyle(
            fontSize: 14,
            fontWeight: FontWeight.bold,
          ),
        ),
        const SizedBox(height: 8),
        Text(
          userObject.toString(),
          style: const TextStyle(
            fontSize: 13,
            fontFamily: 'Courier',
          ),
        ),
      ],
    ),
  ),
),
      ],
     ),
    ),
   ),
  );
 }
}
```

usermodel.dart

```
class UserModel {
 String id;
 String fullname;
 String email;

 UserModel({
  required this.id,
  required this.fullname,
  required this.email,
 });

 // Map to Object (Deserialization)
 factory UserModel.fromMap(Map<String, dynamic> map) {
  return UserModel(
   id: map["id"] ?? "",
   fullname: map["fullname"] ?? "",
   email: map["email"] ?? "",
  );
 }

 // Object to Map (Serialization)
 Map<String, dynamic> toMap() {
  return {
   "id": id,
   "fullname": fullname,
   "email": email,
  };
 }

 @override
 String toString() {
  return 'UserModel(id: $id, fullname: $fullname, email: $email)';
  }
}
```

Output :



JSON Serialization & Deserialization

Serialization Result:

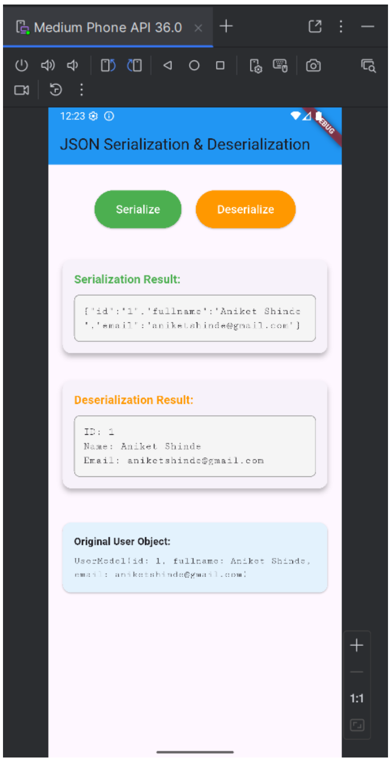
```
Click 'Serialize' button to see
result
```

Deserialization Result:

```
Click 'Deserialize' button to see
result
```

Original User Object:

```
UserModel(id: 1, fullname: Aniket Shinde,
email: aniketshinde@gmail.com)
```



JSON Serialization & Deserialization

Serialization Result:

```
{"id":"1","fullname":"Aniket Shinde
","email":"aniketshinde@gmail.com"}
```

Deserialization Result:

```
ID: 1
Name: Aniket Shinde
Email: aniketshinde@gmail.com
```

Original User Object:

```
UserModel(id: 1, fullname: Aniket Shinde,
email: aniketshinde@gmail.com)
```

3. Write a Flutter program to perform CRUD operations using SQLite (sqflite).

Code :

Main.dart

```
import 'package:flutter/material.dart';
import 'database_helper.dart';
import 'user_model.dart';

void main() {
 runApp(const MyApp());
}

class MyApp extends StatelessWidget {
 const MyApp({Key? key}) : super(key: key);

 @override
 Widget build(BuildContext context) {
  return MaterialApp(
   title: 'SQLite CRUD App',
   theme: ThemeData(
    primarySwatch: Colors.blue,
   ),
   home: const HomePage(),
  );
 }
}

class HomePage extends StatefulWidget {
 const HomePage({Key? key}) : super(key: key);

 @override
 State<HomePage> createState() => _HomePageState();
}

class _HomePageState extends State<HomePage> {
 final dbHelper = DatabaseHelper();
 final TextEditingController nameController = TextEditingController();
 final TextEditingController emailController = TextEditingController();
 final TextEditingController phoneController = TextEditingController();
```

```
List<UserModel> users = [];
UserModel? editingUser;

@override
void initState() {
  super.initState();
  _loadUsers();
}

Future<void> _loadUsers() async {
  final userList = await dbHelper.getAllUsers();
  setState(() {
    users = userList;
  });
}

Future<void> _addUser() async {
  if (nameController.text.isEmpty ||
      emailController.text.isEmpty ||
      phoneController.text.isEmpty) {
    ScaffoldMessenger.of(context).showSnackBar(
      const SnackBar(content: Text('Please fill all fields')),
    );
    return;
  }

  if (editingUser == null) {
    // CREATE
    final newUser = UserModel(
      name: nameController.text,
      email: emailController.text,
      phone: phoneController.text,
    );
    await dbHelper.insertUser(newUser);
    ScaffoldMessenger.of(context).showSnackBar(
      const SnackBar(content: Text('User added successfully')),
    );
  } else {
    // UPDATE
    final updatedUser = UserModel(
      id: editingUser!.id,
      name: nameController.text,
      email: emailController.text,
      phone: phoneController.text,
    );
    await dbHelper.updateUser(updatedUser);
    ScaffoldMessenger.of(context).showSnackBar(
      const SnackBar(content: Text('User updated successfully')),
    );
    editingUser = null;
  }

  nameController.clear();
  emailController.clear();
  phoneController.clear();
  _loadUsers();
}

Future<void> _deleteUser(int id) async {
  await dbHelper.deleteUser(id);
  ScaffoldMessenger.of(context).showSnackBar(
    const SnackBar(content: Text('User deleted successfully')),
  );
  _loadUsers();
}

void _editUser(UserModel user) {
  nameController.text = user.name;
  emailController.text = user.email;
  phoneController.text = user.phone;
  editingUser = user;
  setState(() {});
}

void _clearForm() {
  nameController.clear();
  emailController.clear();
  phoneController.clear();
  editingUser = null;
  setState(() {});
```

```
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: const Text('SQLite CRUD Operations'),
      backgroundColor: Colors.blue,
    ),
    body: SingleChildScrollView(
      child: Padding(
        padding: const EdgeInsets.all(16.0),
        child: Column(
          children: [
            // Input Form
            Card(
              elevation: 5,
              child: Padding(
                padding: const EdgeInsets.all(16.0),
                child: Column(
                  crossAxisAlignment: CrossAxisAlignment.start,
                  children: [
                    Text(
                      editingUser == null ? 'Add New User' : 'Edit User',
                      style: const TextStyle(
                        fontSize: 18,
                        fontWeight: FontWeight.bold,
                      ),
                    ),
                    const SizedBox(height: 16),
                    TextField(
                      controller: nameController,
                      decoration: InputDecoration(
                        labelText: 'Name',
                        border: OutlineInputBorder(
                          borderRadius: BorderRadius.circular(8),
                        ),
                      ),
                    ),
                    const SizedBox(height: 12),
                    TextField(
                      controller: emailController,
                      decoration: InputDecoration(
                        labelText: 'Email',
                        border: OutlineInputBorder(
                          borderRadius: BorderRadius.circular(8),
                        ),
                      ),
                    ),
                    const SizedBox(height: 12),
                    TextField(
                      controller: phoneController,
                      decoration: InputDecoration(
                        labelText: 'Phone',
                        border: OutlineInputBorder(
                          borderRadius: BorderRadius.circular(8),
                        ),
                      ),
                    ),
                    const SizedBox(height: 16),
                    Row(
                      children: [
                        Expanded(
                          child: ElevatedButton(
                            onPressed: _addUser,
                            style: ElevatedButton.styleFrom(
                              backgroundColor: Colors.green,
                              padding: const EdgeInsets.symmetric(vertical: 12),
                            ),
                            child: Text(
                              editingUser == null ? 'Add' : 'Update',
                              style: const TextStyle(
                                fontSize: 16,
                                color: Colors.white,
                              ),
                            ),
                          ),
                        ),
                        const SizedBox(width: 12),
                        if (editingUser != null)
```

```
              Expanded(
                child: ElevatedButton(
                  onPressed: _clearForm,
                  style: ElevatedButton.styleFrom(
                    backgroundColor: Colors.grey,
                    padding: const EdgeInsets.symmetric(vertical: 12),
                  ),
                  child: const Text(
                    'Cancel',
                    style: TextStyle(
                      fontSize: 16,
                      color: Colors.white,
                    ),
                  ),
                ),
              ),
          ],
        ),
      ],
    ),
  ),
),
const SizedBox(height: 24),

// Users List
const Text(
  'Users List',
  style: TextStyle(
    fontSize: 18,
    fontWeight: FontWeight.bold,
  ),
),
const SizedBox(height: 12),
users.isEmpty
    ? const Center(
  child: Padding(
    padding: EdgeInsets.all(32.0),
    child: Text('No users found'),
  ),
)
```

```
          : ListView.builder(
        shrinkWrap: true,
        physics: const NeverScrollableScrollPhysics(),
        itemCount: users.length,
        itemBuilder: (context, index) {
          final user = users[index];
          return Card(
            margin: const EdgeInsets.symmetric(vertical: 8),
            child: ListTile(
              title: Text(user.name),
              subtitle: Column(
                crossAxisAlignment: CrossAxisAlignment.start,
                children: [
                  Text('Email: ${user.email}'),
                  Text('Phone: ${user.phone}'),
                ],
              ),
              trailing: SizedBox(
                width: 100,
                child: Row(
                  mainAxisAlignment: MainAxisAlignment.end,
                  children: [
                    IconButton(
                      icon: const Icon(Icons.edit, color: Colors.blue),
                      onPressed: () => _editUser(user),
                    ),
                    IconButton(
                      icon: const Icon(Icons.delete, color: Colors.red),
                      onPressed: () => _deleteUser(user.id!),
                    ),
                  ],
                ),
              ),
            ),
          );
        },
      ),
    ],
  ),
),
      ),
    );
  }

  @override
  void dispose() {
    nameController.dispose();
    emailController.dispose();
    phoneController.dispose();
    super.dispose();
  }
}
```

user_model.dart

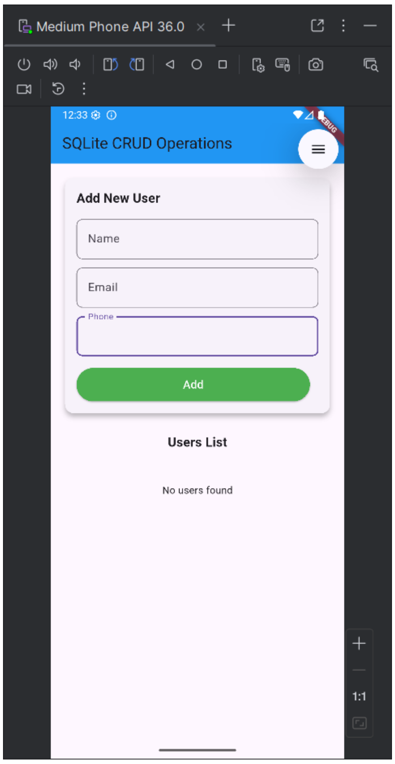
```
class UserModel {
  int? id;
  String name;
  String email;
  String phone;

  UserModel({
    this.id,
    required this.name,
    required this.email,
    required this.phone,
  });

  Map<String, dynamic> toMap() {
    return {
      'id': id,
      'name': name,
      'email': email,
      'phone': phone,
    };
  }

  factory UserModel.fromMap(Map<String, dynamic> map) {
    return UserModel(
      id: map['id'],
      name: map['name'] ?? '',
      email: map['email'] ?? '',
      phone: map['phone'] ?? '',
    );
  }

  @override
  String toString() => 'UserModel(id: $id, name: $name, email: $email, phone:
$phone)';
}
```

database_helper.dart

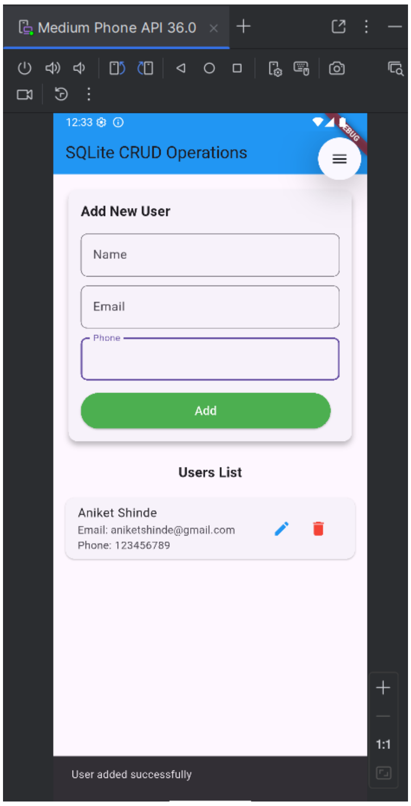
```
import 'package:sqflite/sqflite.dart';
import 'package:path/path.dart';
import 'user_model.dart';

class DatabaseHelper {
  static final DatabaseHelper _instance = DatabaseHelper._internal();

  factory DatabaseHelper() {
    return _instance;
  }

  DatabaseHelper._internal();

  static Database? _database;

  Future<Database> get database async {
    if (_database != null) return _database!;
    _database = await _initDatabase();
    return _database!;
  }

  Future<Database> _initDatabase() async {
    String path = join(await getDatabasesPath(), 'users.db');
    return await openDatabase(
      path,
      version: 1,
      onCreate: _createTable,
    );
  }
```

```
Future<void> _createTable(Database db, int version) async {
  await db.execute('''
    CREATE TABLE users (
      id INTEGER PRIMARY KEY AUTOINCREMENT,
      name TEXT NOT NULL,
      email TEXT NOT NULL,
      phone TEXT NOT NULL
    )
  ''');
}

// CREATE
Future<int> insertUser(UserModel user) async {
  final db = await database;
  return await db.insert(
    'users',
    user.toMap(),
    conflictAlgorithm: ConflictAlgorithm.replace,
  );
}

// READ - Get all users
Future<List<UserModel>> getAllUsers() async {
  final db = await database;
  final List<Map<String, dynamic>> maps = await db.query('users');
  return List.generate(
    maps.length,
        (i) => UserModel.fromMap(maps[i]),
  );
}

// READ - Get single user
Future<UserModel?> getUser(int id) async {
  final db = await database;
  final List<Map<String, dynamic>> maps = await db.query(
    'users',
    where: 'id = ?',
    whereArgs: [id],
   );
   if (maps.isNotEmpty) {


     return UserModel.fromMap(maps.first);
   }
   return null;
 }

 // UPDATE
 Future<int> updateUser(UserModel user) async {
   final db = await database;
   return await db.update(
     'users',
     user.toMap(),
     where: 'id = ?',
     whereArgs: [user.id],
   );
 }

 // DELETE
 Future<int> deleteUser(int id) async {
   final db = await database;
   return await db.delete(
     'users',
     where: 'id = ?',
     whereArgs: [id],
   );
 }

 // DELETE ALL
 Future<void> deleteAllUsers() async {
   final db = await database;
   await db.delete('users');
 }
}
```

Output :

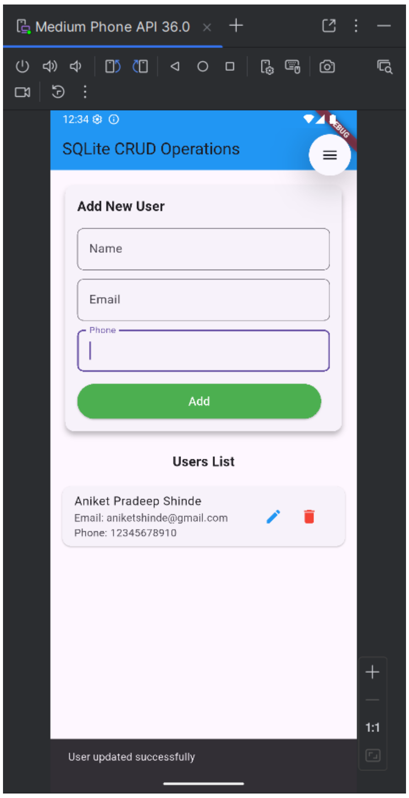


No users found



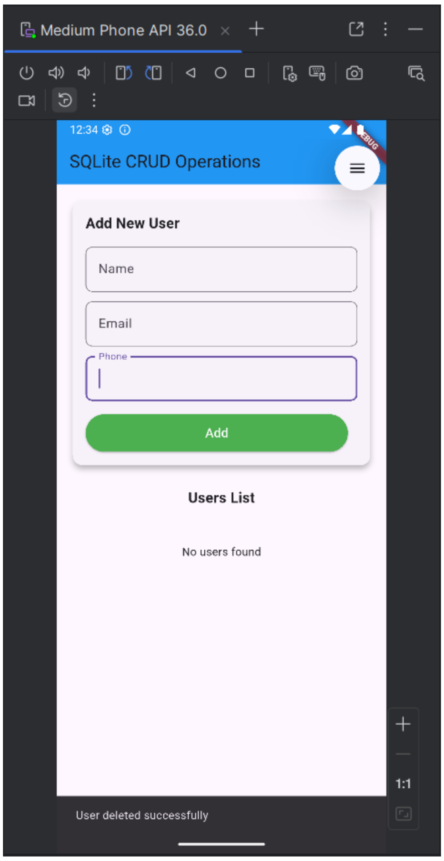
Aniket Shinde
Email: aniketshinde@gmail.com
Phone: 123456789

User added successfully



Aniket Pradeep Shinde
Email: aniketshinde@gmail.com
Phone: 12345678910

User updated successfully



No users found

User deleted successfully