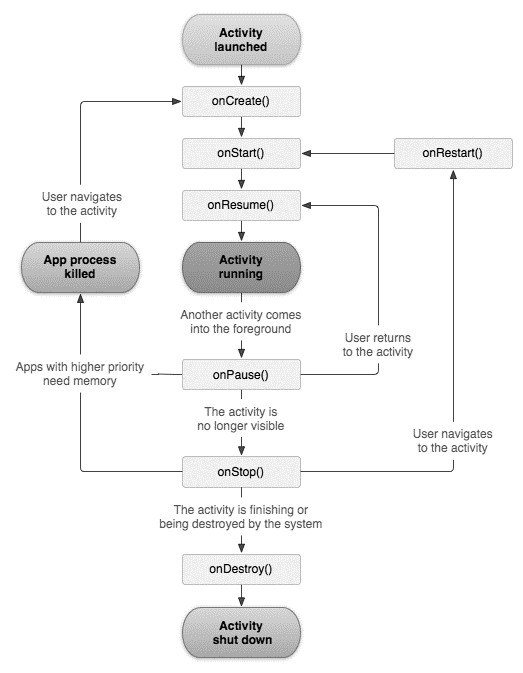
**Experiment – 1**

**Aim:** Build mobile application based on the concept activity life cycle with Custom Toast.

**Description:**

Android Activity Lifecycle is controlled by 7 methods of android.app.Activity class. The android Activity is the subclass of ContextThemeWrapper class.

|  |  |
| --- | --- |
| **Method** | **Description** |
| **onCreate** | called when activity is first created. |
| **onStart** | called when activity is becoming visible to the user. |
| **onResume** | called when activity will start interacting with the user. |
| **onPause** | called when activity is not visible to the user. |
| **onStop** | called when activity is no longer visible to the user. |
| **onRestart** | called after your activity is stopped, prior to start. |
| **onDestroy** | called before the activity is destroyed. |



**Syntax:**

To create a Toast object:

Toast toast = Toast.makeText(getApplicationContext(),"MESSAGE", Duration);

To set gravity for the toast message:

toast.setGravity(gravity, xOffset, yOffset);

To change the duration of a toast message:

toast.setDuration(Toast.*LENGTH\_LONG*);

To display the toast message on android screen:  
toast.show();

**Program**

**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">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Hello World"

android:layout\_marginLeft="70dp"

android:layout\_marginTop="200dp"

android:textSize="40dp"

android:textColor="#C11818"

/>

</LinearLayout>

**MainActivity.java**

package com.example.exp1;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

import android.util.Log;

import android.view.Gravity;

import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

Toast toast = Toast.makeText(getApplicationContext(), "onCreate called", Toast.LENGTH\_LONG);

toast.setGravity(Gravity.TOP,90,90);

toast.show();

}

@Override

protected void onStart(){

super.onStart();

Toast toast = Toast.makeText(getApplicationContext(), "onStart called", Toast.LENGTH\_LONG);

toast.setGravity(Gravity.BOTTOM,90,0);

toast.show();

}

@Override

protected void onPause(){

super.onPause();

Toast toast = Toast.makeText(getApplicationContext(), "onPause called", Toast.LENGTH\_LONG);

toast.setGravity(Gravity.CENTER\_HORIZONTAL,0,0);

toast.show();

}

@Override

protected void onResume(){

super.onResume();

Toast toast = Toast.makeText(getApplicationContext(), "onResume called", Toast.LENGTH\_LONG);

toast.setGravity(Gravity.CENTER\_VERTICAL,0,0);

toast.show();

}

@Override

protected void onStop(){

super.onStop();

Toast toast = Toast.makeText(getApplicationContext(), "onStop called", Toast.LENGTH\_LONG);

toast.setGravity(Gravity.CENTER|Gravity.END,0,0);

toast.show();

}

@Override

protected void onRestart(){

super.onRestart();

Toast toast = Toast.makeText(getApplicationContext(), "onRestart called", Toast.LENGTH\_LONG);

toast.setGravity(Gravity.BOTTOM|Gravity.END,0,0);

toast.show();

}

@Override

protected void onDestroy(){

super.onDestroy();

Toast toast = Toast.makeText(getApplicationContext(), "onDestroy called", Toast.LENGTH\_LONG);

toast.setGravity(Gravity.TOP|Gravity.END,0,0);

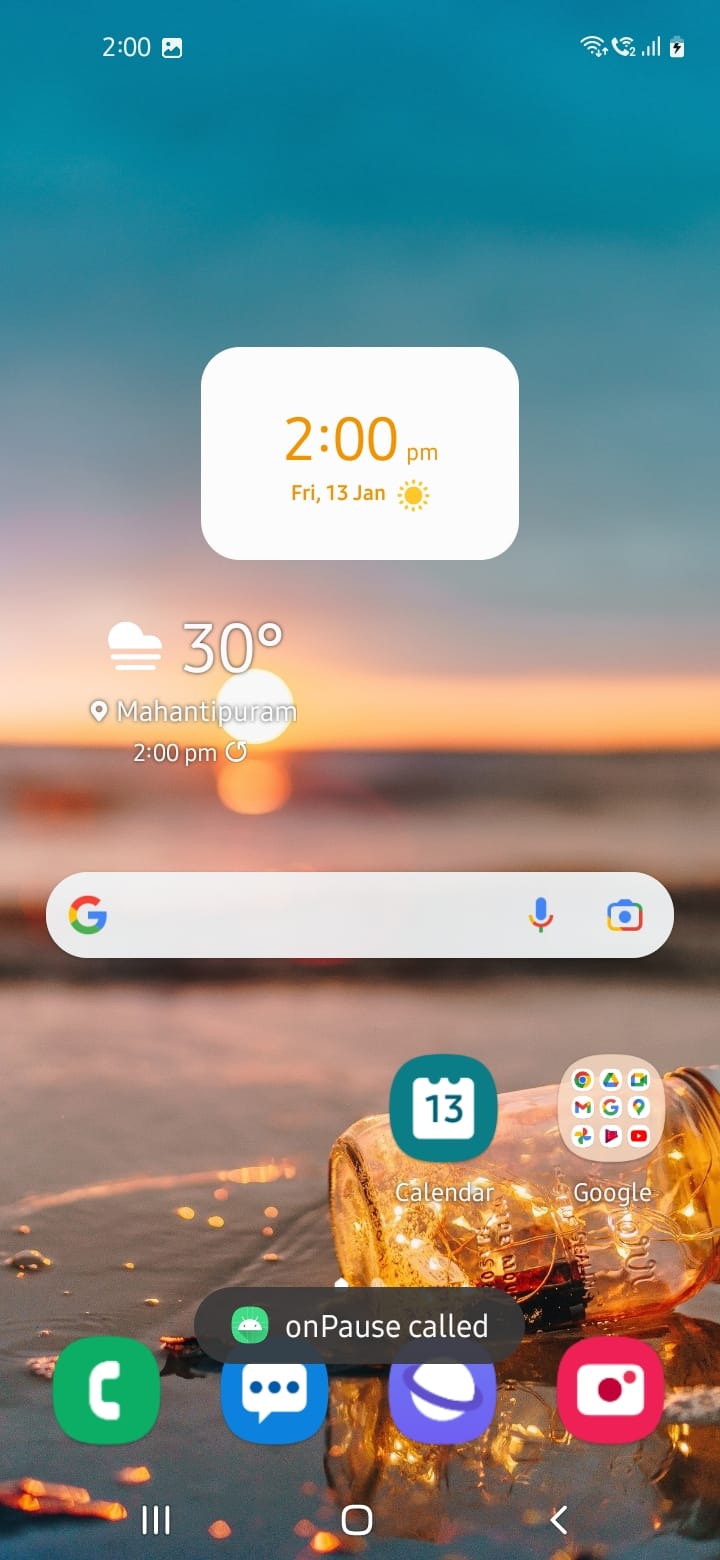
toast.show();

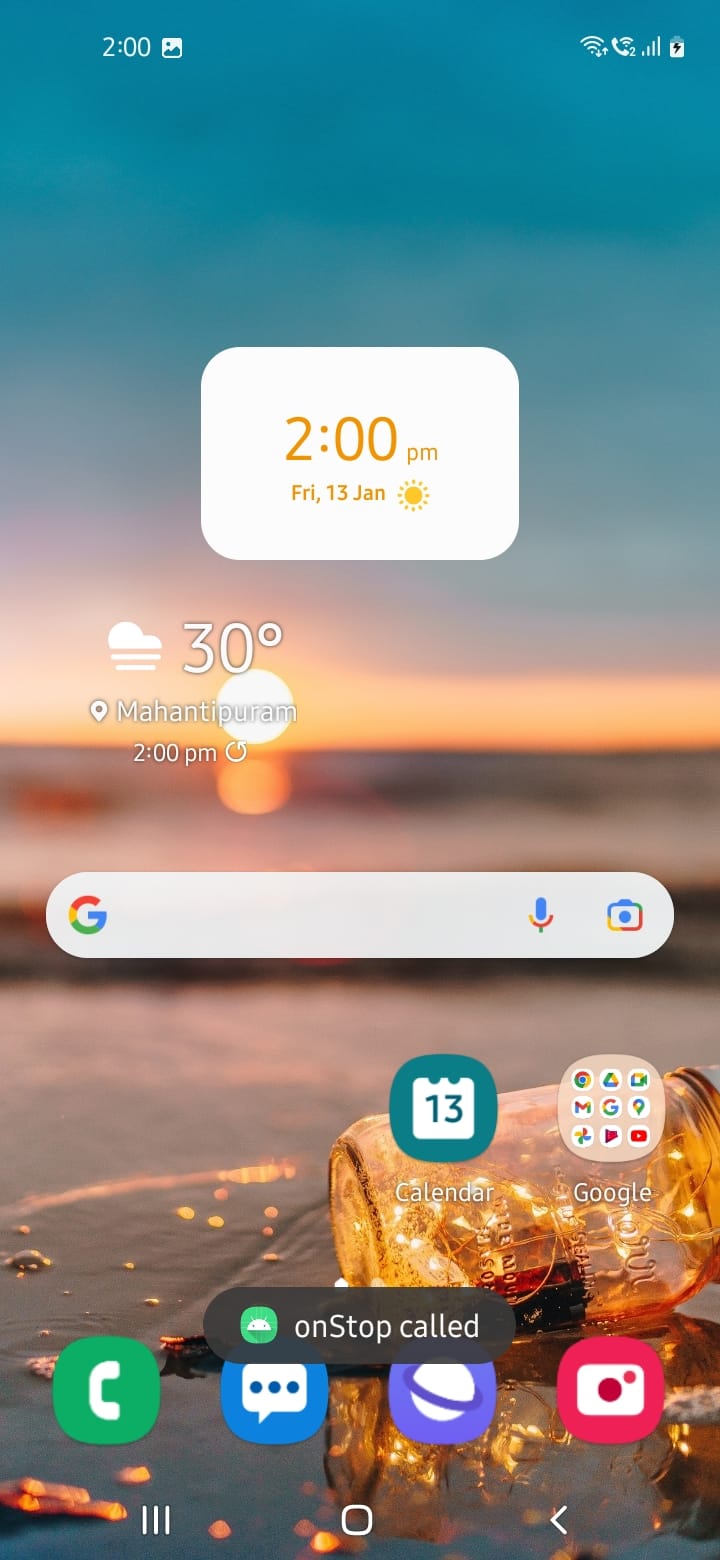
}

}

**Output**:

**Experiment – 2**

**Aim:** Build mobile application using different layouts (use any 3 layouts)

**I. Linear Layout**

**Description:**

Linear layout is a simple layout used in android for layout designing. In the Linear layout all the elements are displayed in linear fashion means all the children/elements of a linear layout are displayed according to its orientation. The value for orientation property can be either horizontal or vertical.

**Syntax:**

<LinearLayout

xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical" >

…………………….

……………………..

……………………..

</LinearLayout>

**Program:**

**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" >

<TextView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="175dp"

android:layout\_marginBottom="30dp"

android:text="Are you"

android:textAlignment="center"

android:textColor="#0000FF"

android:textSize="50sp" />

<TextView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Sure?"

android:textAlignment="center"

android:textColor="#0000FF"

android:textSize="50sp"

/>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_marginTop="50dp"

android:orientation="horizontal">

<Button

android:layout\_width="wrap\_content"

android:layout\_height="75dp"

android:layout\_marginLeft="70dp"

android:layout\_marginRight="20dp"

android:text="Yes"

android:textSize="30sp" />

<Button

android:layout\_width="wrap\_content"

android:layout\_height="75dp"

android:text="Cancel"

android:textSize="30sp"

android:layout\_marginRight="20dp" />

</LinearLayout>

</LinearLayout>

**MainActivity.java**

package com.example.linearlayout;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

}

}

**Output:**



**II. Relative Layout**

**Description:**

The Relative Layout is very flexible layout used in android for custom layout designing. It gives us the flexibility to position our component/view based on the relative or sibling component’s position. Just because it allows us to position the component anywhere, we want so it is considered as most flexible layout. For the same reason Relative layout is the most used layout after the Linear Layout in Android. It allows its child view to position relative to each other or relative to the container or another container.

**Syntax:**

<RelativeLayout

xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

>

…………………….

……………………..

……………………..

</RelativeLayout>

**Program:**

**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"

>

<Button

android:id="@+id/btn1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentLeft="true"

android:text="Button 1" />

<Button

android:id="@+id/btn2"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentRight="true"

android:layout\_centerVertical="true"

android:text="Button 2" />

<Button

android:id="@+id/btn3"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentLeft="true"

android:layout\_centerVertical="true"

android:text="Button 3" />

<Button

android:id="@+id/btn4"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_alignParentBottom="true"

android:text="Button 4" />

<Button

android:id="@+id/btn5"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignBottom="@+id/btn2"

android:layout\_centerHorizontal="true"

android:text="Button 5" />

<Button

android:id="@+id/btn6"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_above="@+id/btn4"

android:layout\_centerHorizontal="true"

android:text="Button 6" />

<Button

android:id="@+id/btn7"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_toEndOf="@+id/btn1"

android:layout\_toRightOf="@+id/btn1"

android:layout\_alignParentRight="true"

android:text="Button 7" />

</RelativeLayout>

**MainActivity.java**

package com.example.relativelayout;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

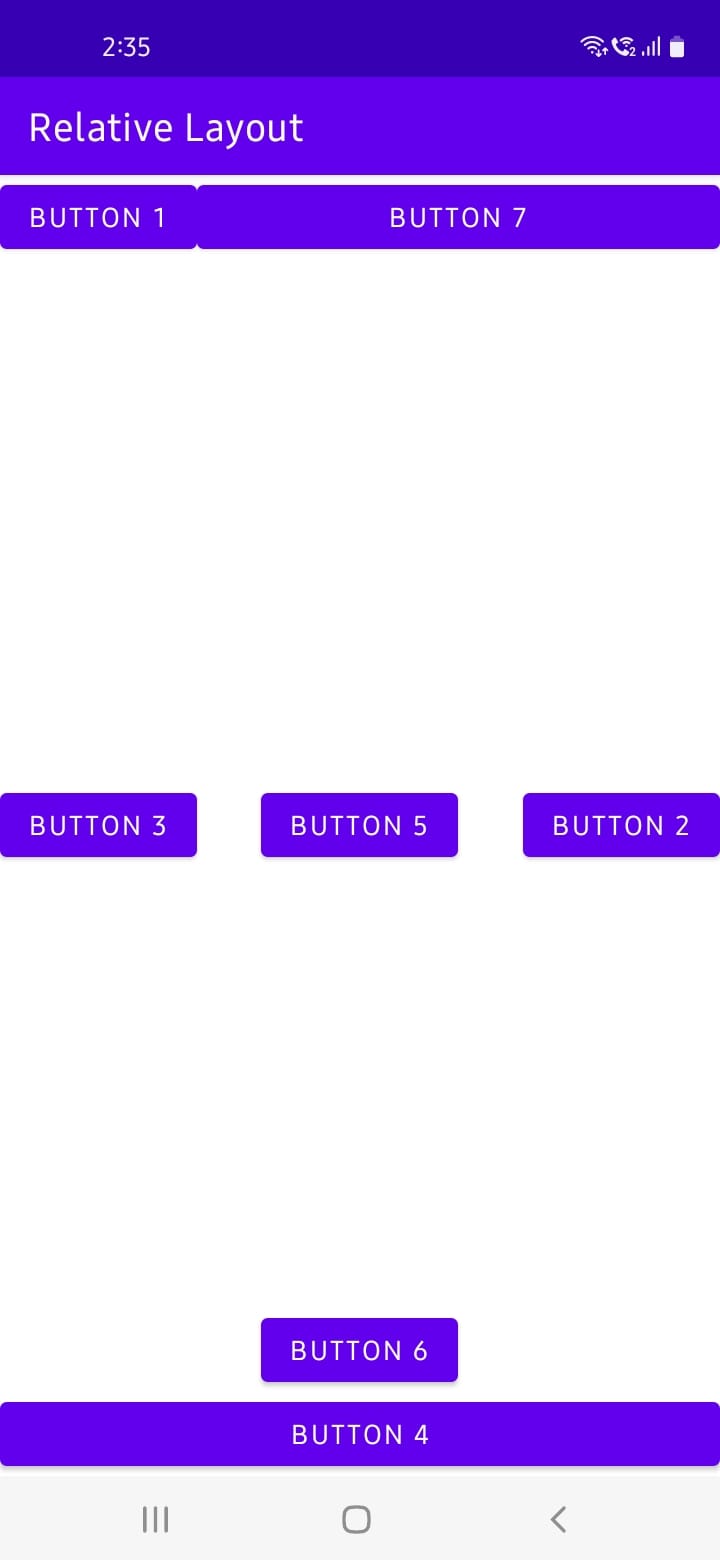
super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

}

}

**Output:**



**III. Table Layout**

**Description:**

In Android, Table Layout is used to arrange the group of views into rows and columns. Table Layout containers do not display a border line for their columns, rows or cells. A Table will have as many columns as the row with the most cells.

**Syntax:**

<TableLayout

xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

>

<TableRow>

……………..

……………..

</TableRow>

<TableRow>

……………..

……………..

</TableRow>

…………………….

……………………..

……………………..

</TableLayout>

**Program:**

**main\_activity.java**

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

<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_marginTop="100dp"

android:paddingLeft="10dp"

android:paddingRight="10dp" >

<TableRow android:background="#0079D6" android:padding="5dp">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:text="UserId" />

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:text="Name" />

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:text="Location" />

</TableRow>

<TableRow android:background="#DAE8FC" android:padding="5dp">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:text="101" />

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:text="Ravi" />

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:text="Vijayawada" />

</TableRow>

<TableRow android:background="#DAE8FC" android:padding="5dp">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:text="102" />

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:text="Ramu" />

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:text="Guntur" />

</TableRow>

<TableRow android:background="#DAE8FC" android:padding="5dp">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:text="103" />

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:text="Raju" />

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:text="Vishakapatnam" />

</TableRow>

</TableLayout>

**MainActivity.java**

package com.example.tablelayout;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

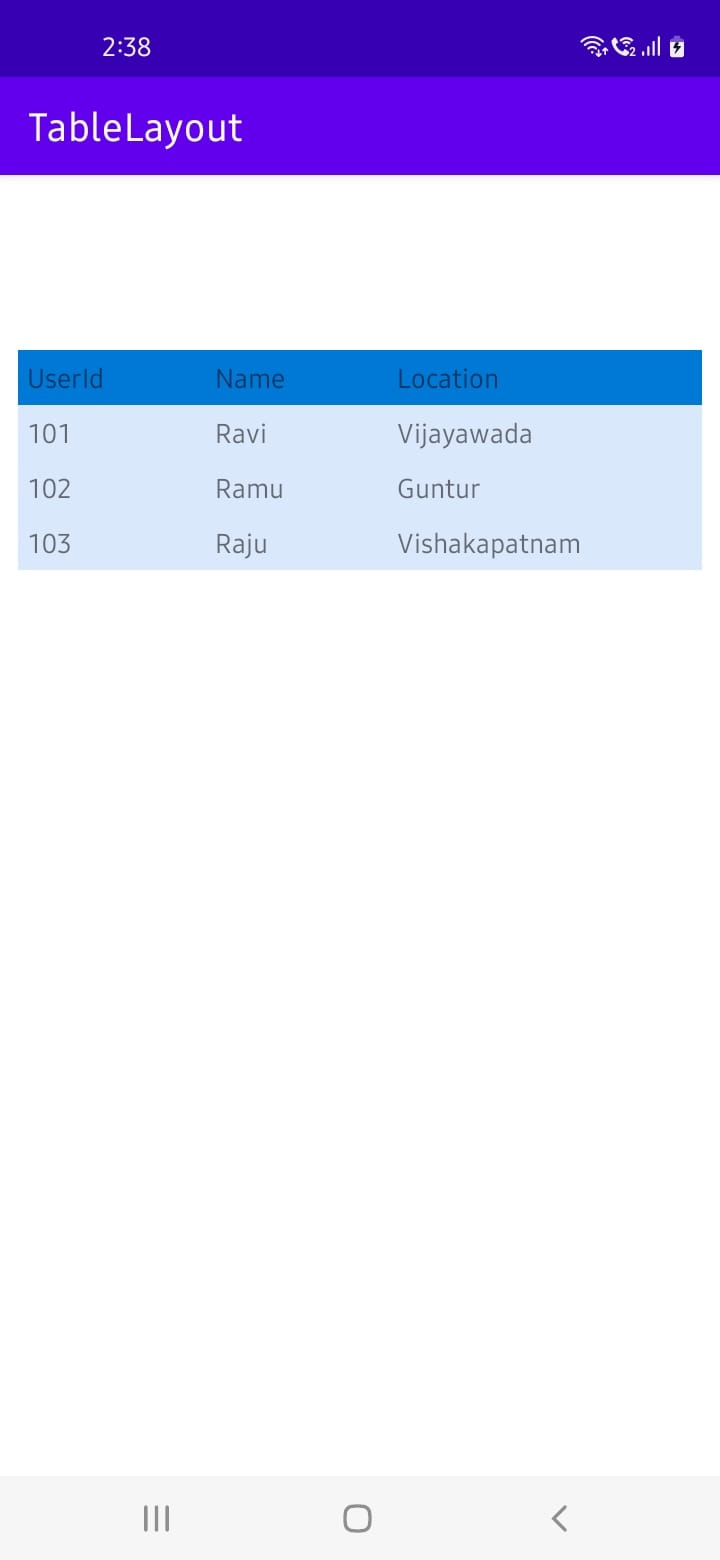
super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

}

}

**Output:**



**Experiment – 3**

**Aim:** Build mobile application using different dialogs (use any 2 dialogs)

**Description:**

A dialog is a small window that prompts the user to make a decision or enter additional information. A dialog does not fill the screen and is normally used for events that require users to take an action before they can proceed.

In android, you can create following types of Dialogs:

* Alert Dialog
* DatePicker Dialog
* TimePicker Dialog
* Custom Dialog

**I. TimePicker Dialog**

**Description:**

Android Time Picker allows you to select the time of day in either 24 hour or AM/PM mode. The time consists of hours, minutes and clock format. Android provides this functionality through TimePicker class.

**Syntax:**

To create a view in xml file:

<TimePicker

android:id="@+id/timePicker1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content" />

To create a TimePicker object in java file:

import android.widget.TimePicker;

TimePicker timePicker1;

timePicker1 = (TimePicker) findViewById(R.id.timePicker1);

To get the current time:

int hour = timePicker1.getCurrentHour();

int min = timePicker1.getCurrentMinute();

**Program:**

**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">

<TimePicker

android:id="@+id/timePicker1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_centerHorizontal="true"

android:layout\_marginTop="20dp" />

<Button

android:id="@+id/button1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_below="@+id/timePicker1"

android:layout\_marginTop="10dp"

android:layout\_marginLeft="160dp"

android:text="CLICK HERE" />

<TextView

android:id="@+id/textView1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_below="@+id/button1"

android:layout\_marginLeft="120dp"

android:layout\_marginTop="10dp"

android:textStyle="bold"

android:textSize="18dp"/>

</RelativeLayout>

**MainActivity.java**

package com.example.timepicker;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Build;

import android.os.Bundle;

import android.view.\*;

import android.widget.\*;

public class MainActivity extends AppCompatActivity {

TimePicker picker;

Button btnGet;

TextView tvw;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

tvw=(TextView)findViewById(R.id.textView1);

picker=(TimePicker)findViewById(R.id.timePicker1);

picker.setIs24HourView(true);

btnGet=(Button)findViewById(R.id.button1);

btnGet.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

int hour, minute;

String am\_pm;

if (Build.VERSION.SDK\_INT >= 23 ){

hour = picker.getHour();

minute = picker.getMinute();

}

else{

hour = picker.getCurrentHour();

minute = picker.getCurrentMinute();

}

if(hour > 12) {

am\_pm = "PM";

hour = hour - 12;

}

else

{

am\_pm="AM";

}

tvw.setText("Selected Time: "+ hour +":"+ minute+" "+am\_pm);

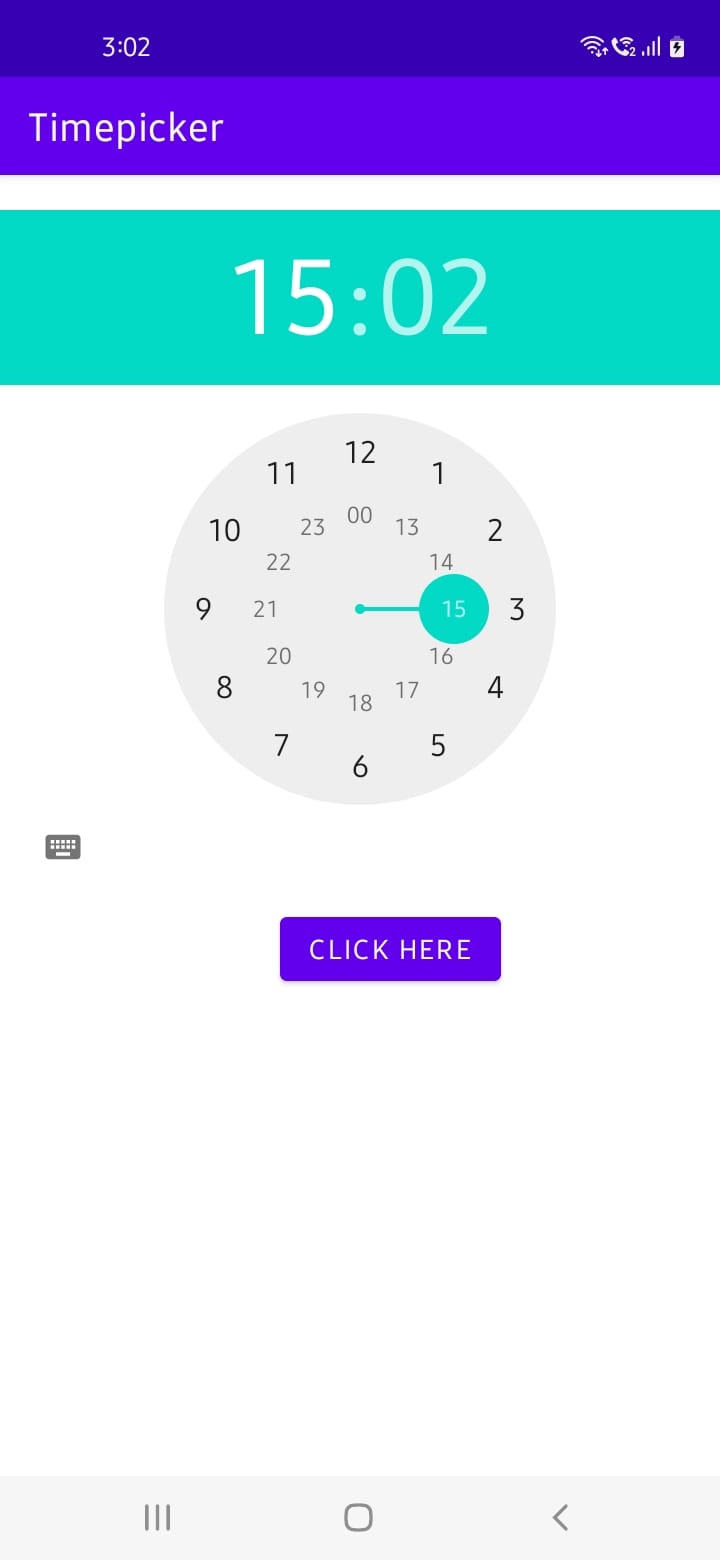
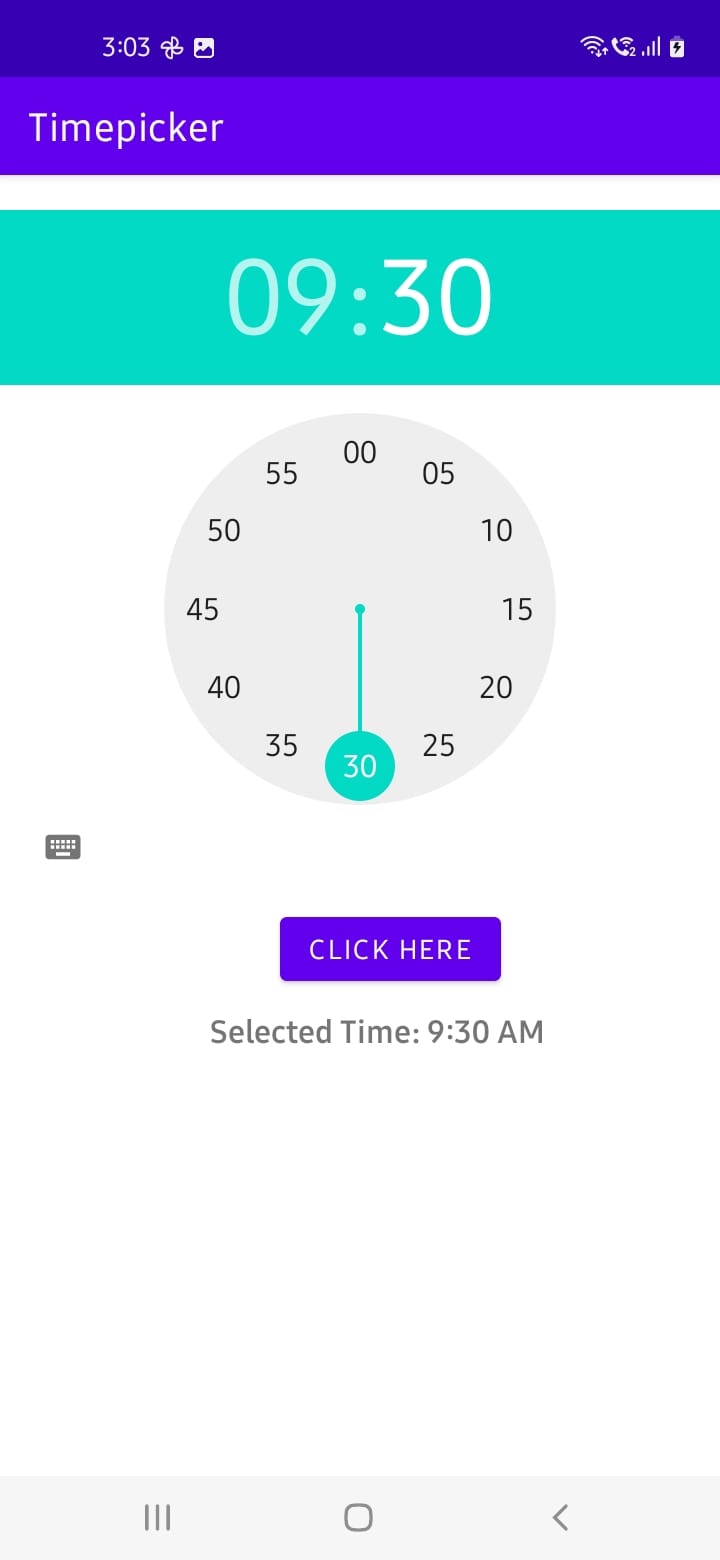
}

});

}

}

**Output:**

**II. DatePicker Dialog**

**Description:**

Android DatePicker is a user interface control that is used to select the date by day, month, and year in the android application. DatePicker is used to ensure that the users will select a valid date.

Android Date Picker allows you to select the date consisting of day, month and year in your custom user interface. For this functionality android provides DatePicker and DatePickerDialog components.

**Syntax:**

To create a view in xml file:

<DatePicker

android:id="@+id/date"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"/>

To create a DatePicker object in java file:

import android.widget.DatePicker;

DatePicker d1 = (DatePicker) findViewById(R.id.date);

To get the selected date:

int hour = timePicker1.getCurrentHour();

int min = timePicker1.getCurrentMinute();

int month = d1.getMonth() + 1;

int year = d1.getYear();

int date = d1.getDayOfMonth();

**Program:**

**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"

>

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Choose the Date"

android:textStyle="bold"

android:textSize="30dp"

android:textColor="#C53434"

android:layout\_marginTop="30dp"

android:layout\_marginLeft="80dp"

/>

<DatePicker

android:id="@+id/date"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="30dp"

android:layout\_marginTop="50dp" />

<Button

android:id="@+id/button1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Enter"

android:layout\_marginLeft="140dp"

android:backgroundTint="@color/button\_background\_color"

/>

<TextView

android:id="@+id/text"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="80dp"

android:textSize="25dp"

android:textStyle="bold"

android:textColor="#C83A3A"

/>

</LinearLayout>

**MainActivity.java**

package com.example.datepicker;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

import android.view.\*;

import android.widget.\*;

public class MainActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

DatePicker d1 = (DatePicker) findViewById(R.id.date);

Button b1 = (Button) findViewById(R.id.button1);

TextView t1 = (TextView) findViewById(R.id.text);

b1.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

int month = d1.getMonth() + 1;

int year = d1.getYear();

int date = d1.getDayOfMonth();

if(2022 - year >= 18)

t1.setText("You are eligible to Vote");

else

t1.setText("You are not eligible to Vote");

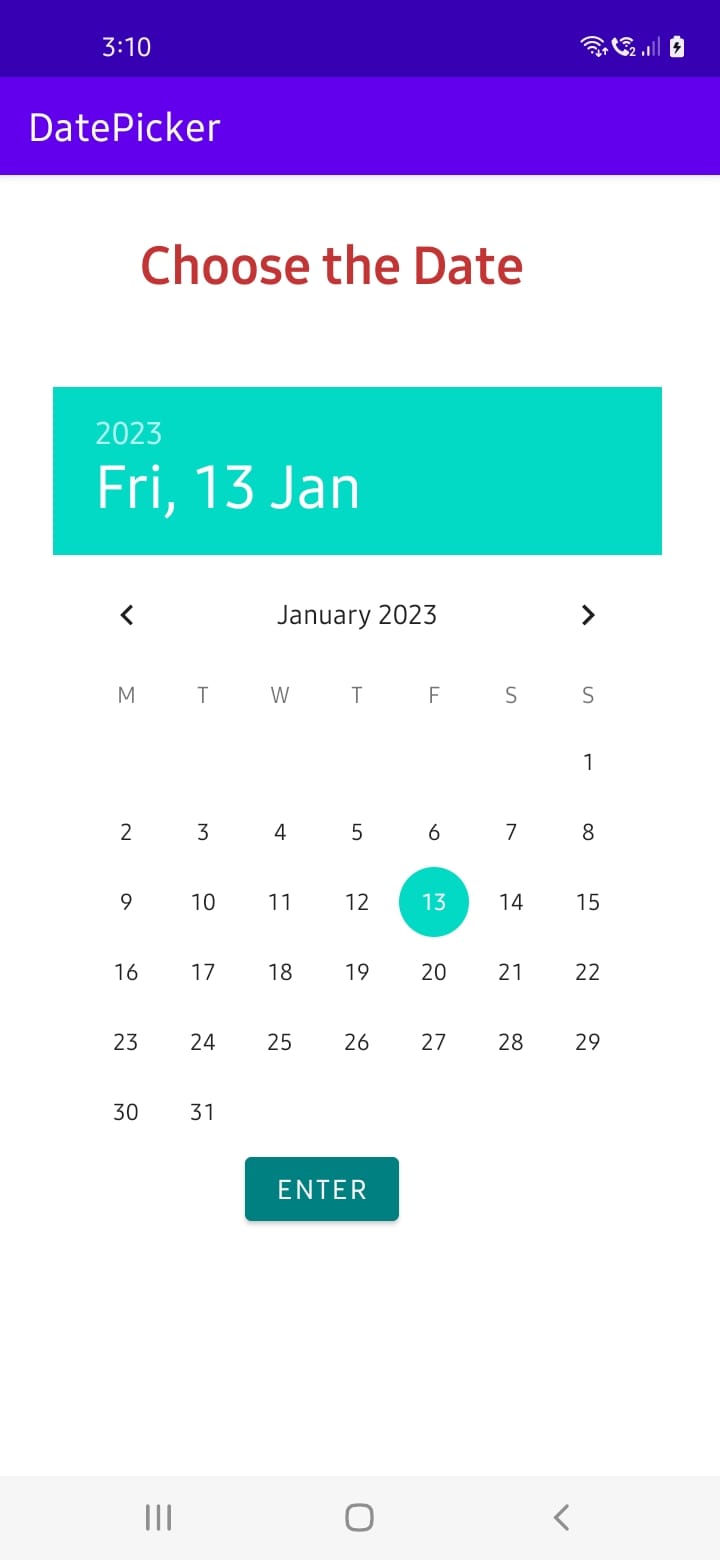
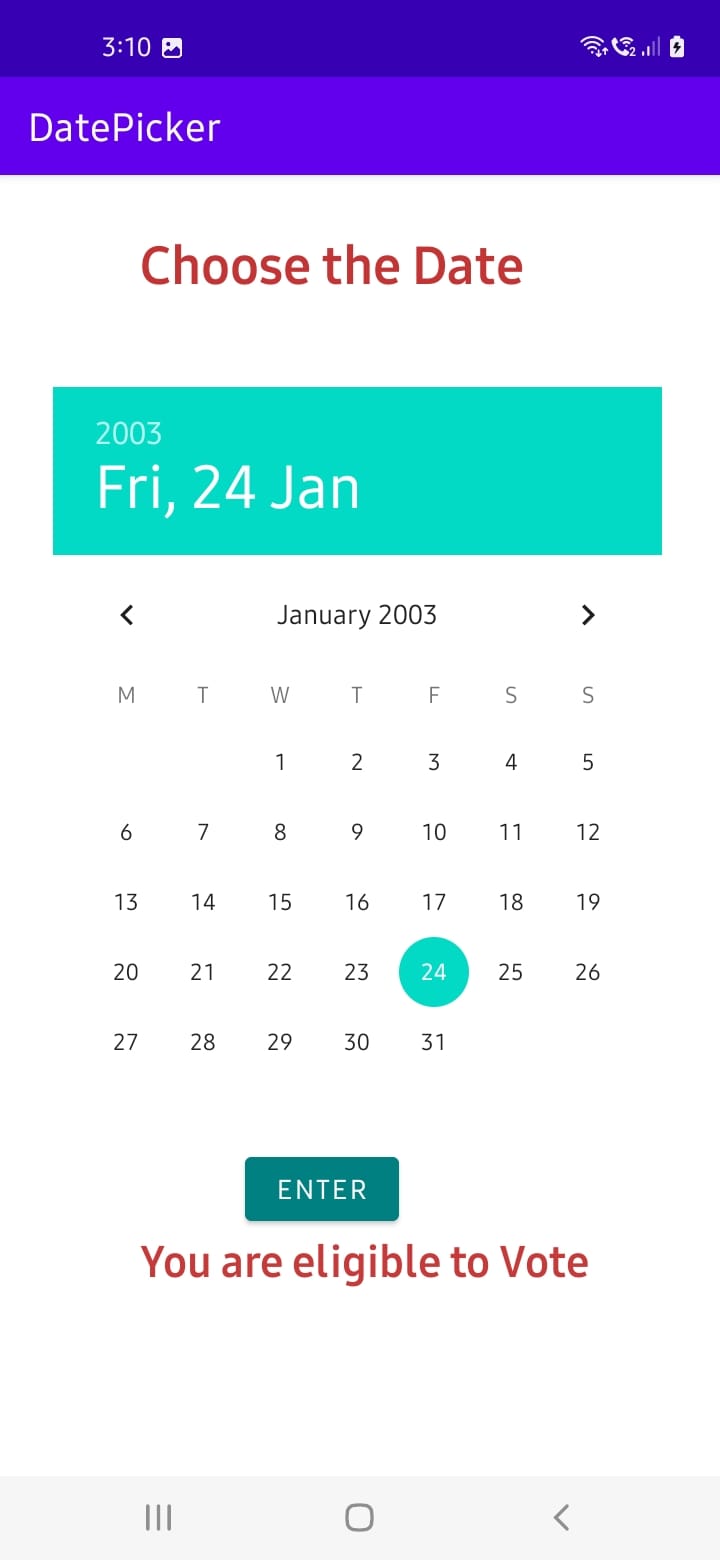
}

});

}

}

**Output:**

**Experiment – 4**

**Aim:** Build mobile application using Recycler View

**Description:**

RecyclerView is a ViewGroup added to the android studio as a successor of the GridView and ListView. It is an improvement on both of them and can be found in the latest v-7 support packages. It has been created to make possible construction of any lists with XML layouts as an item which can be customized vastly while improving on the efficiency of ListViews and GridViews. This improvement is achieved by recycling the views which are out of the visibility of the user. For example, if a user scrolled down to a position where items 4 and 5 are visible; items 1, 2, and 3 would be cleared from the memory to reduce memory consumption.

**Syntax:**

recyclerView.setLayoutManager(new LinearLayoutManager(this));

recyclerView.setAdapter(new MyAdapter(getApplicationContext(),items));

**Program:**

Step – 1: Open Gradle Scripts > build.gradle (Module) and add the following implementation in the dependencies.

implementation 'androidx.recyclerview:recyclerview:1.1.0'

Step – 2:

**activity\_main.xml**

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

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

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".MainActivity">

<androidx.recyclerview.widget.RecyclerView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:id="@+id/recyclerview"/>

</RelativeLayout>

**item\_view.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="wrap\_content"

android:layout\_marginTop="10dp"

xmlns:tools="http://schemas.android.com/tools">

<ImageView

android:layout\_width="48dp"

android:layout\_height="48dp"

android:id="@+id/imageview"

android:layout\_marginRight="10dp"

android:layout\_centerVertical="true"

/>

<TextView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:id="@+id/name"

android:layout\_toEndOf="@id/imageview"

android:textColor="@color/black"

android:textSize="20dp"

tools:text="Name"/>

<TextView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:id="@+id/email"

android:layout\_toEndOf="@id/imageview"

android:layout\_below="@id/name"

android:textColor="@color/black"

tools:text="emial@email.com"/>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="2dp"

android:layout\_below="@id/imageview"

android:layout\_marginTop="10dp"

android:background="#DAD8D8"/>

</RelativeLayout>

**strings.xml** (res 🡪 values 🡪 strings.xml)

<resources>

<string name="app\_name">MyRecyclerViewDemo</string>

</resources>

**MainActivity.java**

package com.example.myrecyclerviewdemo;

import androidx.appcompat.app.AppCompatActivity;

import androidx.recyclerview.widget.LinearLayoutManager;

import androidx.recyclerview.widget.RecyclerView;

import android.os.Bundle;

import java.util.ArrayList;

import java.util.List;

public class MainActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

RecyclerView recyclerView = findViewById(R.id.recyclerview);

List<Item> items = new ArrayList<Item>();

items.add(new Item("John wick","john.wick@email.com",R.drawable.a));

items.add(new Item("Robert j","robert.j@email.com",R.drawable.b));

items.add(new Item("James Gunn","james.gunn@email.com",R.drawable.c));

items.add(new Item("Ricky tales","rickey.tales@email.com",R.drawable.d));

items.add(new Item("Micky mose","mickey.mouse@email.com",R.drawable.e));

items.add(new Item("Pick War","pick.war@email.com",R.drawable.f));

items.add(new Item("Leg piece","leg.piece@email.com",R.drawable.g));

items.add(new Item("Apple Mac","apple.mac@email.com",R.drawable.g));

items.add(new Item("John wick","john.wick@email.com",R.drawable.a));

items.add(new Item("Robert j","robert.j@email.com",R.drawable.b));

items.add(new Item("James Gunn","james.gunn@email.com",R.drawable.c));

items.add(new Item("Ricky tales","rickey.tales@email.com",R.drawable.d));

items.add(new Item("Micky mose","mickey.mouse@email.com",R.drawable.e));

items.add(new Item("Pick War","pick.war@email.com",R.drawable.f));

items.add(new Item("Leg piece","leg.piece@email.com",R.drawable.g));

items.add(new Item("Apple Mac","apple.mac@email.com",R.drawable.g));

recyclerView.setLayoutManager(new LinearLayoutManager(this));

recyclerView.setAdapter(new MyAdapter(getApplicationContext(),items));

}

}

**Item.java**

package com.example.myrecyclerviewdemo;

public class Item {

String name;

String email;

int image;

public Item(String name, String email, int image) {

this.name = name;

this.email = email;

this.image = image;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public int getImage() {

return image;

}

public void setImage(int image) {

this.image = image;

}

}

**MyAdapter.java**

package com.example.myrecyclerviewdemo;

import android.content.Context;

import android.view.LayoutInflater;

import android.view.ViewGroup;

import java.util.List;

import androidx.annotation.NonNull;

import androidx.recyclerview.widget.RecyclerView;

public class MyAdapter extends RecyclerView.Adapter<MyViewHolder> {

Context context;

List<Item> items;

public MyAdapter(Context context, List<Item> items) {

this.context = context;

this.items = items;

}

@NonNull

@Override

public MyViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {

return new MyViewHolder(LayoutInflater.from(context).inflate(R.layout.item\_view,parent,false));

}

@Override

public void onBindViewHolder(@NonNull MyViewHolder holder, int position) {

holder.nameView.setText(items.get(position).getName());

holder.emailView.setText(items.get(position).getEmail());

holder.imageView.setImageResource(items.get(position).getImage());

}

@Override

public int getItemCount() {

return items.size();

}

}

**MyViewHolder.java**

package com.example.myrecyclerviewdemo;

import android.view.View;

import android.widget.ImageView;

import android.widget.TextView;

import androidx.annotation.NonNull;

import androidx.recyclerview.widget.RecyclerView;

public class MyViewHolder extends RecyclerView.ViewHolder {

ImageView imageView;

TextView nameView,emailView;

public MyViewHolder(@NonNull View itemView) {

super(itemView);

imageView = itemView.findViewById(R.id.imageview);

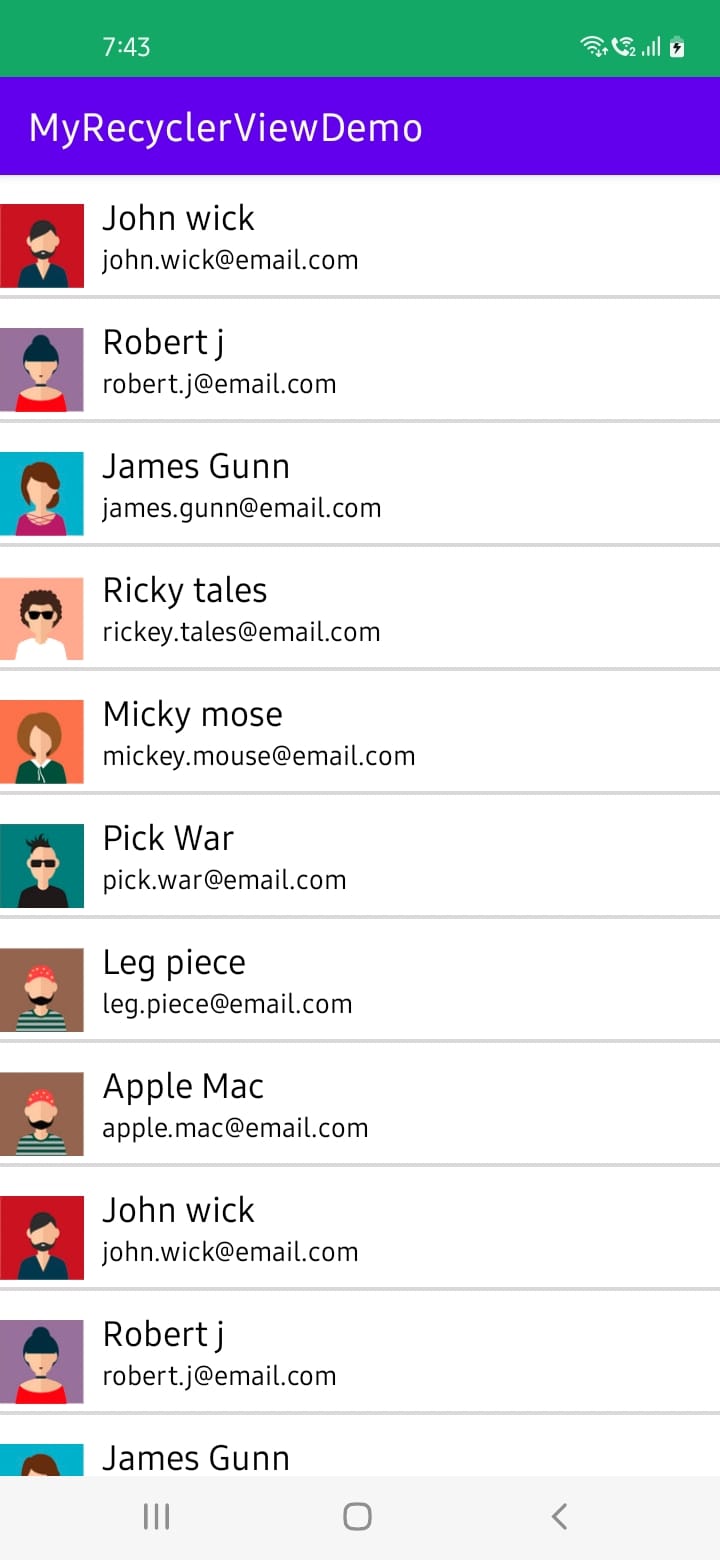
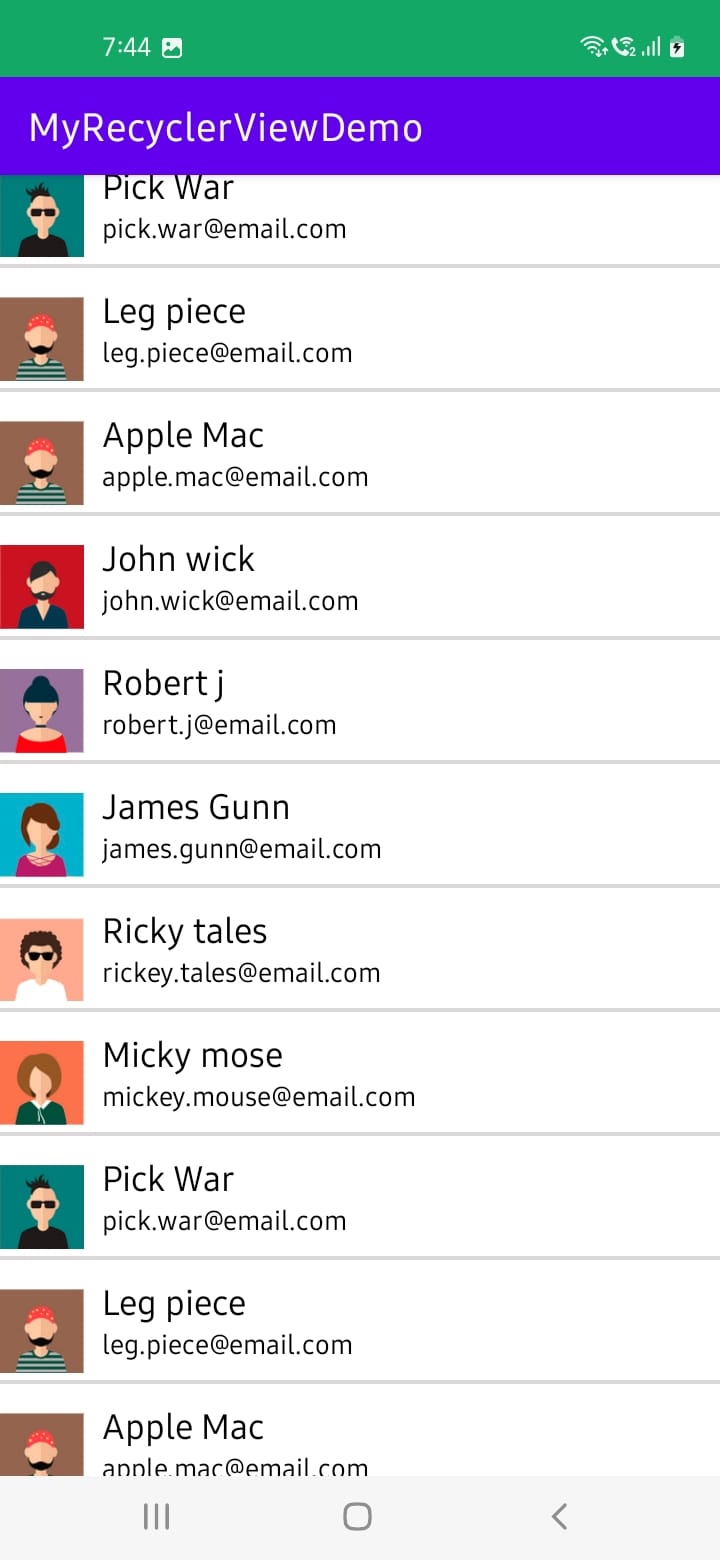
nameView = itemView.findViewById(R.id.name);

emailView = itemView.findViewById(R.id.email);

}

}

**Output:**

**Experiment – 5**

**Aim:** Build mobile application to switch from one activity to another using Intent.

**Description:**

An intent is to perform an action on the screen. It is mostly used to start activity, send broadcast receiver, start services and send message between two activities. There are two intents available in android as Implicit Intents and Explicit Intents. Here is a sample example to start new activity with old activity.

**I. Implicit Intents**

An implicit intent allows you to start an activity in another app by describing an action you

intend to perform, such as "share an article", "view a map", or "take a picture".

An implicit intent specifies an action and may provide data with which to perform the action.

Implicit intents do not specify the target activity class, just the intended action.

Android runtime matches the implicit intent request with registered intent handlers.

If there are multiple matches, an App Chooser will open to let the user decide.

In Implicit Intents we do not need to specify the name of the component. We just specify the Action which has to be performed and further this action is handled by the component of another application.

**Syntax:**

Intent intent = new Intent(Intent.ACTION\_VIEW);

intent.setData(Uri.parse("https://www.google.com"));

startActivity(intent);

**Program:**

**activity\_main.xml**

<LinearLayout

android:layout\_height="match\_parent"

android:layout\_width="match\_parent"

android:orientation="vertical"

xmlns:android="http://schemas.android.com/apk/res/android">

<TextView

android:text="Implicit Intents"

android:gravity="center"

android:textSize="30sp"

android:layout\_marginTop="100dp"

android:textColor="#FDFBFB"

android:background="#D80E0E"

android:layout\_width="match\_parent"

android:layout\_height="50dp"

android:layout\_marginLeft="10sp"

android:layout\_marginRight="10sp"/>

<EditText

android:id="@+id/input"

android:layout\_marginTop="100dp"

android:hint="Enter the URL"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="5sp"/>

<Button

android:onClick="submit"

android:layout\_gravity="center"

android:text="Submit"

android:layout\_marginTop="10dp"

android:id="@+id/btn"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"/>

</LinearLayout>

**MainActivity.java**

package com.example.implicitintents;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;

import android.net.Uri;

import android.os.Bundle;

import android.view.View;

import android.widget.EditText;

public class MainActivity extends AppCompatActivity {

EditText url;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

url = findViewById(R.id.input);

}

public void submit(View view) {

String input = url.getText().toString();

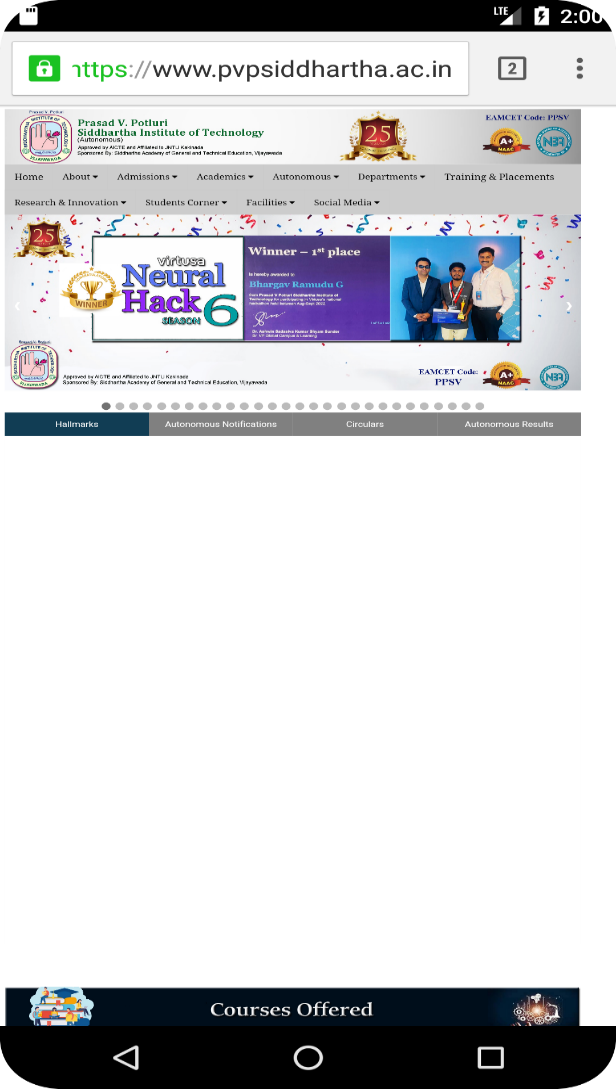
Intent intent = new Intent(Intent.ACTION\_VIEW, Uri.parse(input));

startActivity(intent);

}

}

**Output**:

**II. Explicit Intents**

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

invoked.

An explicit intent is used to launch a specific app component. Eg. a service in the app or a

particular activity.

To create an explicit intent, we need to define the component name for the intent object (all

other intent properties are optional).

The Intent (Context, Class) constructor supplies the app Context and the component a Class

object. As such, this intent explicitly starts the DownloadService class in the app.

**Syntax:**

To pass the data to second activity:

Intent intent = new Intent(MainActivity.this, SecondActivity.class);

intent.putExtra("Name", id);

startActivity(intent);

To receive the data from first activity:

Intent intent = getIntent();

String name = intent.getStringExtra("Name");

**Program:**

**activity\_main.xml**

<LinearLayout

android:layout\_height="match\_parent"

android:layout\_width="match\_parent"

android:orientation="vertical"

xmlns:android="http://schemas.android.com/apk/res/android" >

<TextView

android:text="Average Calculator"

android:gravity="center"

android:textSize="30sp"

android:layout\_marginTop="50dp"

android:textColor="#FDFBFB"

android:background="#6777D1"

android:layout\_width="match\_parent"

android:layout\_height="50dp"

android:layout\_marginLeft="10sp"

android:layout\_marginRight="10sp"/>

<LinearLayout

android:layout\_marginLeft="10dp"

android:layout\_marginRight="10dp"

android:layout\_marginTop="50dp"

android:orientation="horizontal"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content">

<TextView

android:textSize="20sp"

android:text="Student Name: "

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"/>

<EditText

android:id="@+id/sid"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"/>

</LinearLayout>

<LinearLayout

android:layout\_marginLeft="10dp"

android:layout\_marginRight="10dp"

android:layout\_marginTop="5dp"

android:orientation="horizontal"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content">

<TextView

android:textSize="20sp"

android:text="Subject 1: "

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"/>

<EditText

android:id="@+id/sub1"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"/>

</LinearLayout>

<LinearLayout

android:layout\_marginLeft="10dp"

android:layout\_marginRight="10dp"

android:layout\_marginTop="5dp"

android:orientation="horizontal"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content">

<TextView

android:textSize="20sp"

android:text="Subject 2: "

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"/>

<EditText

android:id="@+id/sub2"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"/>

</LinearLayout>

<LinearLayout

android:layout\_marginLeft="10dp"

android:layout\_marginRight="10dp"

android:layout\_marginTop="5dp"

android:orientation="horizontal"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content">

<TextView

android:textSize="20sp"

android:text="Subject 3: "

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"/>

<EditText

android:id="@+id/sub3"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"/>

</LinearLayout>

<Button

android:onClick="percent"

android:layout\_gravity="center"

android:text="Calculate Average"

android:layout\_marginTop="10dp"

android:id="@+id/btn"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"/>

</LinearLayout>

**MainActivity.java**

package com.example.explicitintents;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;

import android.os.Bundle;

import android.util.Log;

import android.view.View;

import android.widget.EditText;

public class MainActivity extends AppCompatActivity {

EditText et1, et2, et3, et4, et5, et6;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

et1 = findViewById(R.id.sid);

et2 = findViewById(R.id.sub1);

et3 = findViewById(R.id.sub2);

et4 = findViewById(R.id.sub3);

}

public void percent(View view) {

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

int sub1 = Integer.parseInt(et2.getText().toString());

int sub2 = Integer.parseInt(et3.getText().toString());

int sub3 = Integer.parseInt(et4.getText().toString());

int total = sub1+sub2+sub3;

double per = (double)total/3;

Log.d("act",total+" "+per);

String res = String.valueOf(per);

Intent intent = new Intent(this, SecondActivity.class);

intent.putExtra("Percent",res);

intent.putExtra("sid",id);

startActivity(intent);

}

}

**activity\_second.xml**

<LinearLayout

android:layout\_height="match\_parent"

android:layout\_width="match\_parent"

android:orientation="vertical"

xmlns:android="http://schemas.android.com/apk/res/android" >

<TextView

android:layout\_marginLeft="10dp"

android:layout\_marginTop="50dp"

android:textSize="30sp"

android:textColor="#4CAF50"

android:text="Student Average"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"/>

<TextView

android:layout\_marginRight="10dp"

android:layout\_marginLeft="10dp"

android:layout\_marginTop="30dp"

android:id="@+id/result"

android:layout\_width="match\_parent"

android:layout\_height="50dp"/>

</LinearLayout>

**SecondActivity.java**

package com.example.explicitintents;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;

import android.os.Bundle;

import android.widget.TextView;

import android.widget.Toast;

public class SecondActivity extends AppCompatActivity {

TextView tv;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_second);

tv = findViewById(R.id.result);

Intent intent = getIntent();

String sid = intent.getStringExtra("sid");

String per = intent.getStringExtra("Percent");

tv.setText("Student "+sid+" : "+per);

}

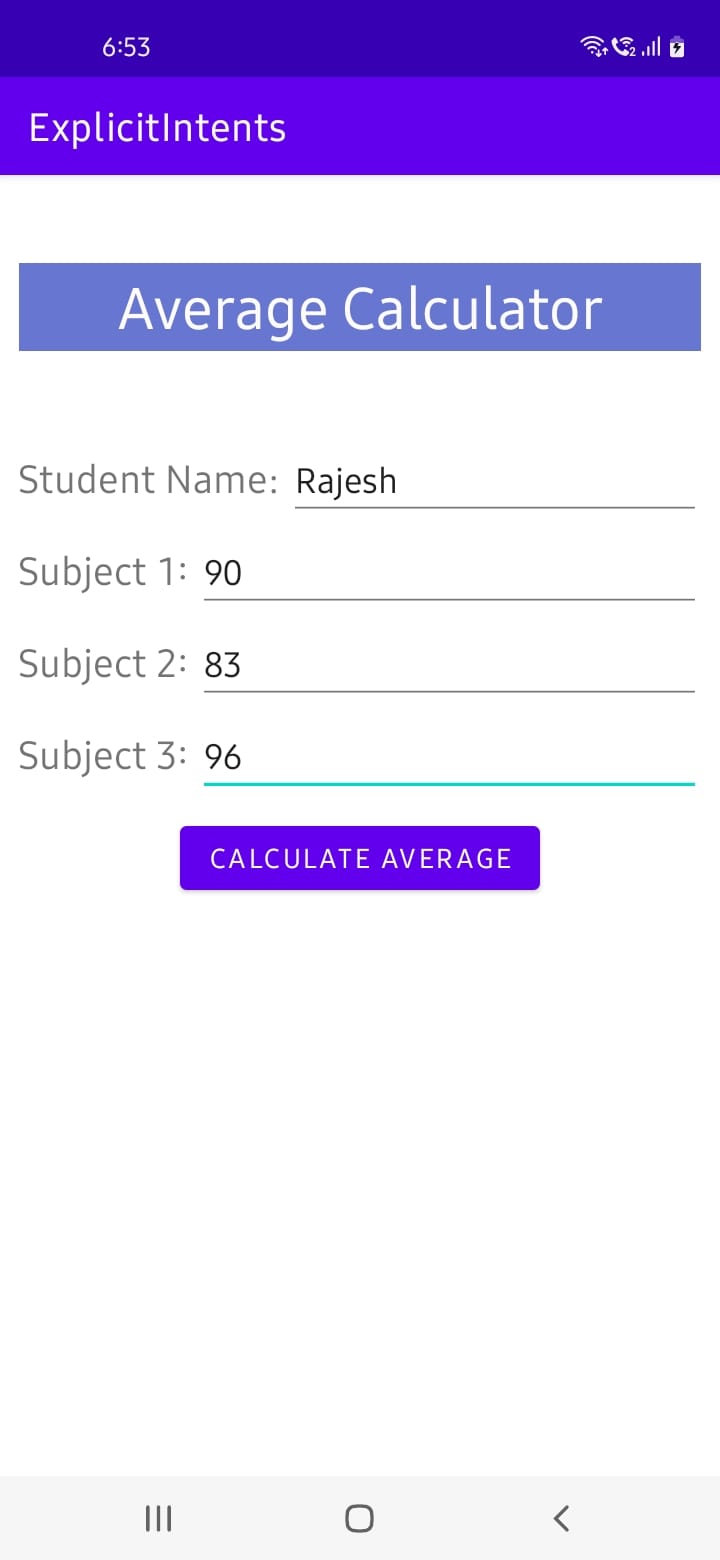
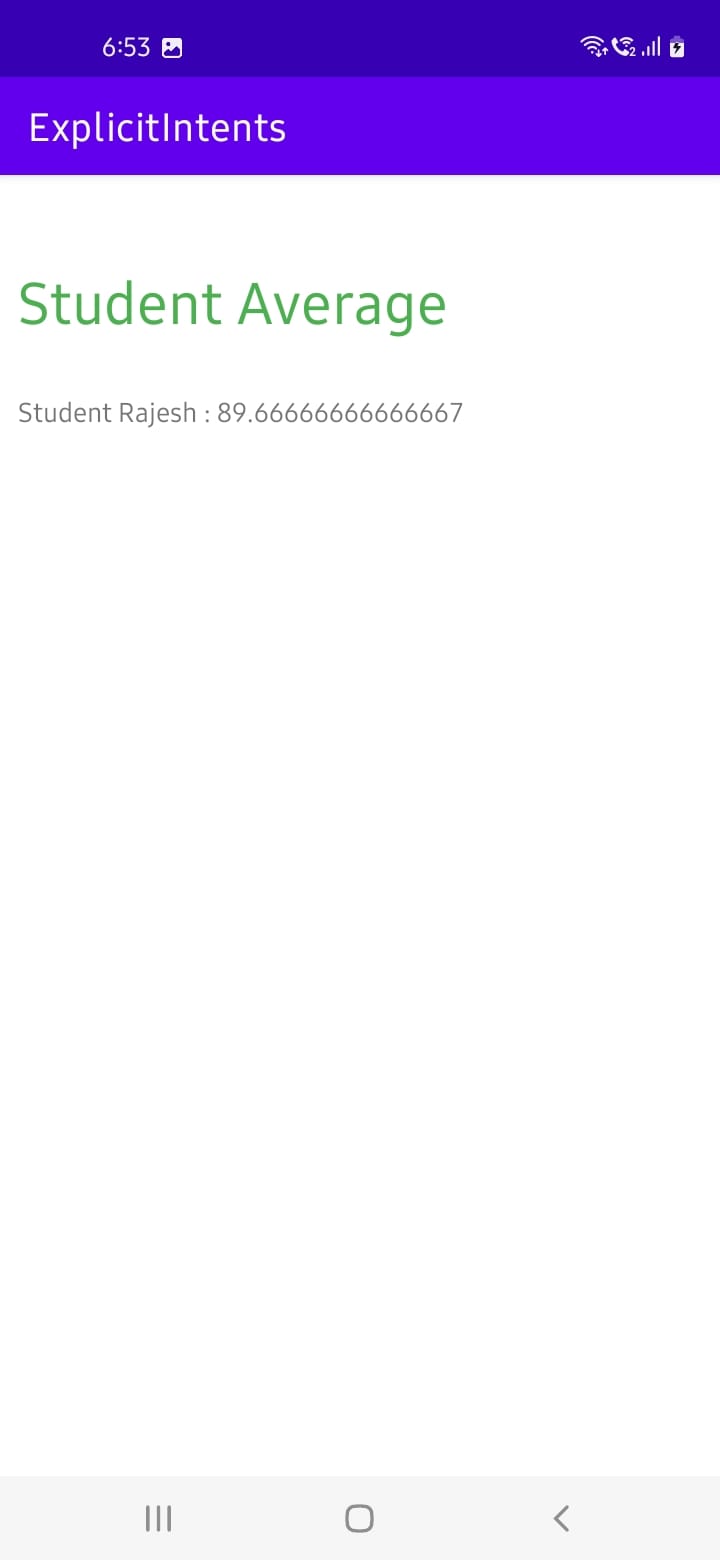
@Override

public void onBackPressed() {

Toast.makeText(this,"You are not allowed to go back",Toast.LENGTH\_LONG); }

}

**Output:**

**Experiment – 6**

**Aim:** Build mobile application to demonstrate Dynamic Fragments.

**Description:**

Fragments are more often used for user interface purposes. Fragments are used when the user wants to see two different views of two different classes on the same screen. Fragments were added with Android Honeycomb. So, if you are developing an application only for Android 3.0 (HoneyComb) or higher, then Android provides you access to the Fragments class. You can also access the FragmentManager by calling the getFragmnetManager() method in your Activities.

**Syntax:**

To add the fragment:

FragmentManager fm=getSupportFragmentManager();

FragmentTransaction ft=fm.beginTransaction();

ft.add(R.id.f1,new frag1());

ft.addToBackStack(null);

ft.commit();

To remove the fragment:

FragmentManager fm=getSupportFragmentManager();

FragmentTransaction ft=fm.beginTransaction();

if(fm.getBackStackEntryCount()>0) {

fm.popBackStack();

Toast.makeText(this, "Fragment is deleted!!", Toast.LENGTH\_SHORT).show();

}

ft.commit();

To replace the fragment:

FragmentManager fm=getSupportFragmentManager();

FragmentTransaction ft=fm.beginTransaction();

ft.replace(R.id.f1,new frag2());

Toast.makeText(this, "Fragment is replaced!!", Toast.LENGTH\_SHORT).show();

ft.addToBackStack(null);

ft.commit();

**Program:**

**activity\_main.xml**

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

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".MainActivity"

android:orientation="vertical">

<fragment

android:id="@+id/f1"

android:layout\_width="match\_parent"

android:layout\_height="300dp"

class="com.example.dynamicfragment.frag2"

android:layout\_margin="30dp"/>

<Button

android:id="@+id/add"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Add the fragment"

android:onClick="add"/>

<Button

android:id="@+id/delete"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Remove the fragment"

android:onClick="delete"/>

<Button

android:id="@+id/replace"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Replace the fragment"

android:onClick="replace"/>

</LinearLayout>

**MainActivity.java**

package com.example.dynamicfragment;

import androidx.appcompat.app.AppCompatActivity;

import androidx.fragment.app.FragmentManager;

import androidx.fragment.app.FragmentTransaction;

import android.os.Bundle;

import android.view.View;

import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

}

public void add(View v)

{

FragmentManager fm=getSupportFragmentManager();

FragmentTransaction ft=fm.beginTransaction();

ft.add(R.id.f1,new frag1());

Toast.makeText(this, "Fragment is added!!", Toast.LENGTH\_SHORT).show();

ft.addToBackStack(null);

ft.commit();

}

public void delete(View v)

{

FragmentManager fm=getSupportFragmentManager();

FragmentTransaction ft=fm.beginTransaction();

if(fm.getBackStackEntryCount()>0) {

fm.popBackStack();

Toast.makeText(this, "Fragment is deleted!!", Toast.LENGTH\_SHORT).show();

}

ft.commit();

}

public void replace(View v)

{

FragmentManager fm=getSupportFragmentManager();

FragmentTransaction ft=fm.beginTransaction();

ft.replace(R.id.f1,new frag2());

Toast.makeText(this, "Fragment is replaced!!", Toast.LENGTH\_SHORT).show();

ft.addToBackStack(null);

ft.commit();

}

}

**fragment\_frag1.xml**

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

<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".frag1"

android:background="#C18383">

<TextView

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:text="Hello"

android:textSize="50dp"

android:textColor="#3F51B5"

android:gravity="center"/>

</FrameLayout>

**frag1.java**

package com.example.dynamicfragment;

import android.os.Bundle;

import androidx.fragment.app.Fragment;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

public class frag1 extends Fragment {

public frag1() {

}

@Override

public View onCreateView(LayoutInflater l, ViewGroup container,

Bundle savedInstanceState) {

// Inflate the layout for this fragment

View v= l.inflate(R.layout.fragment\_frag1, container, false);

return v;

}

}

**fragment\_frag2.xml**

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

<FrameLayout

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".frag2"

android:background="@color/teal\_200">

<TextView

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:text="World"

android:textColor="#3F51B5"

android:textSize="50dp"

android:gravity="center"

/>

</FrameLayout>

**frag2.java**

package com.example.dynamicfragment;

import android.os.Bundle;

import androidx.fragment.app.Fragment;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

public class frag2 extends Fragment {

public frag2() {

}

@Override

public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {

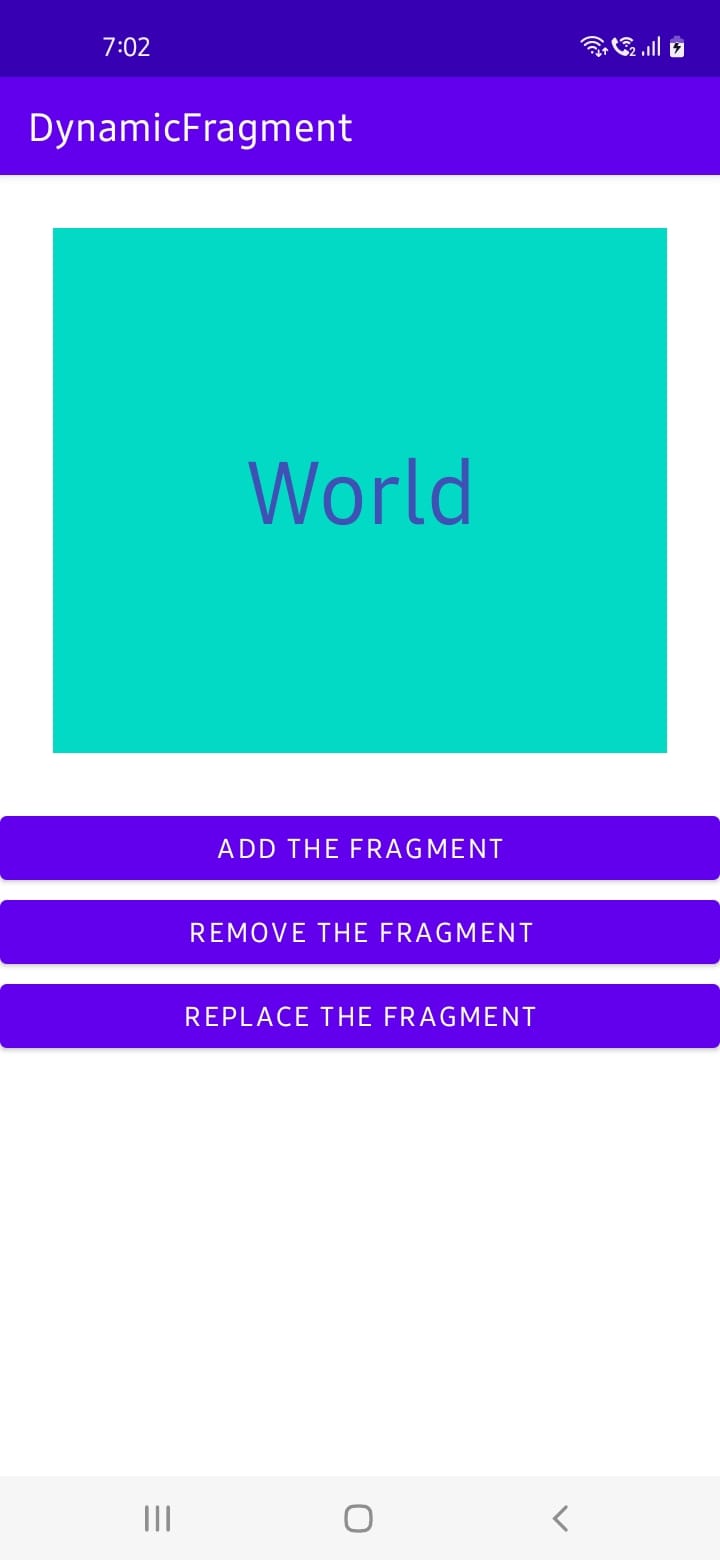
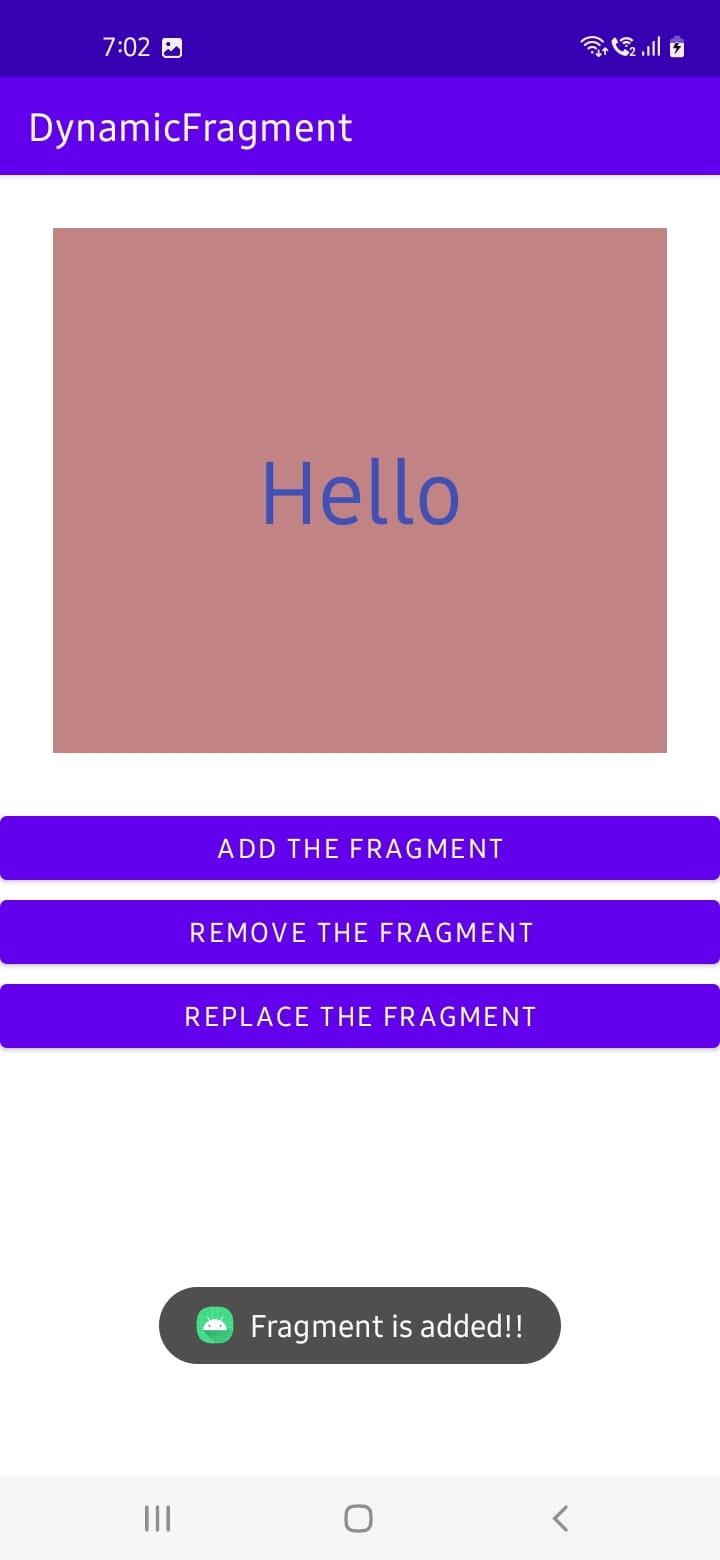
View v=inflater.inflate(R.layout.fragment\_frag2, container, false);

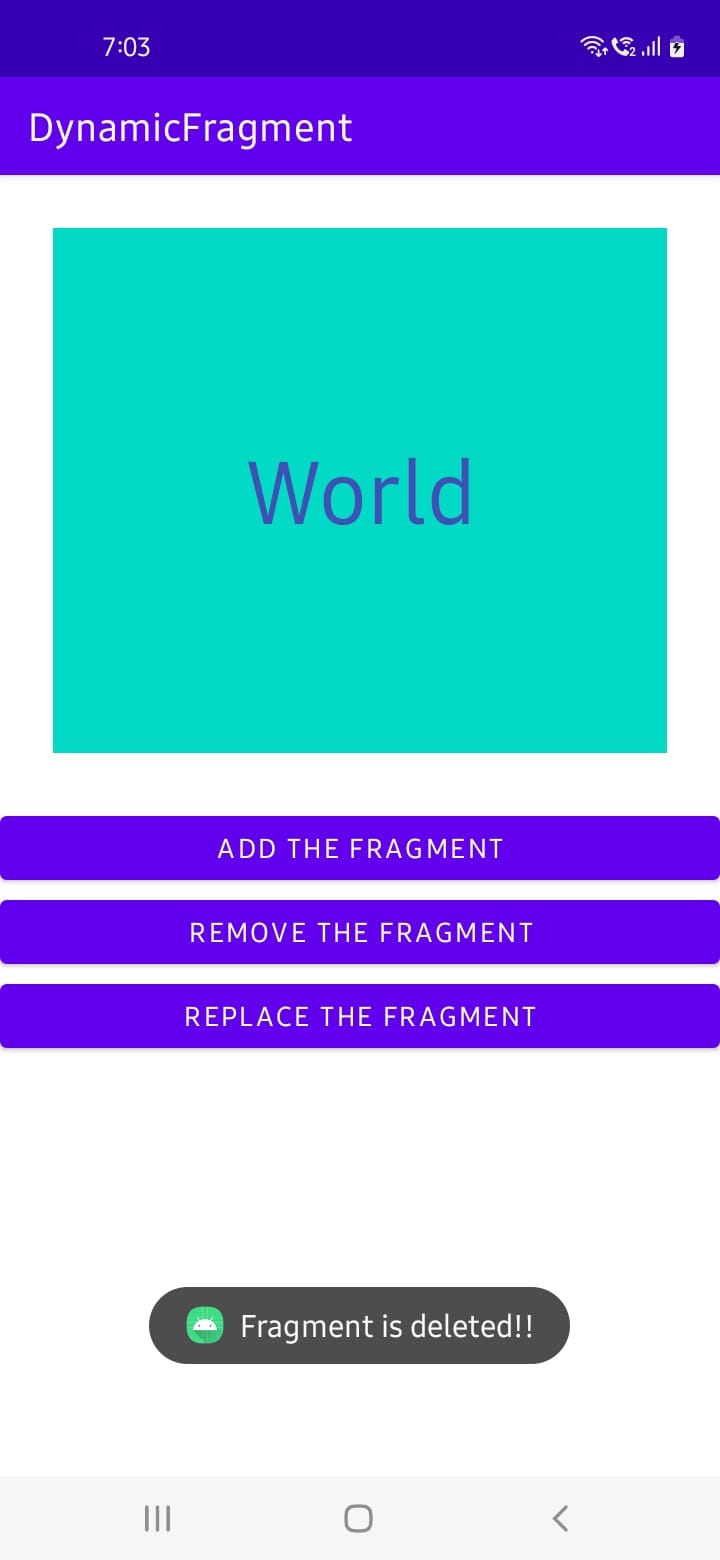
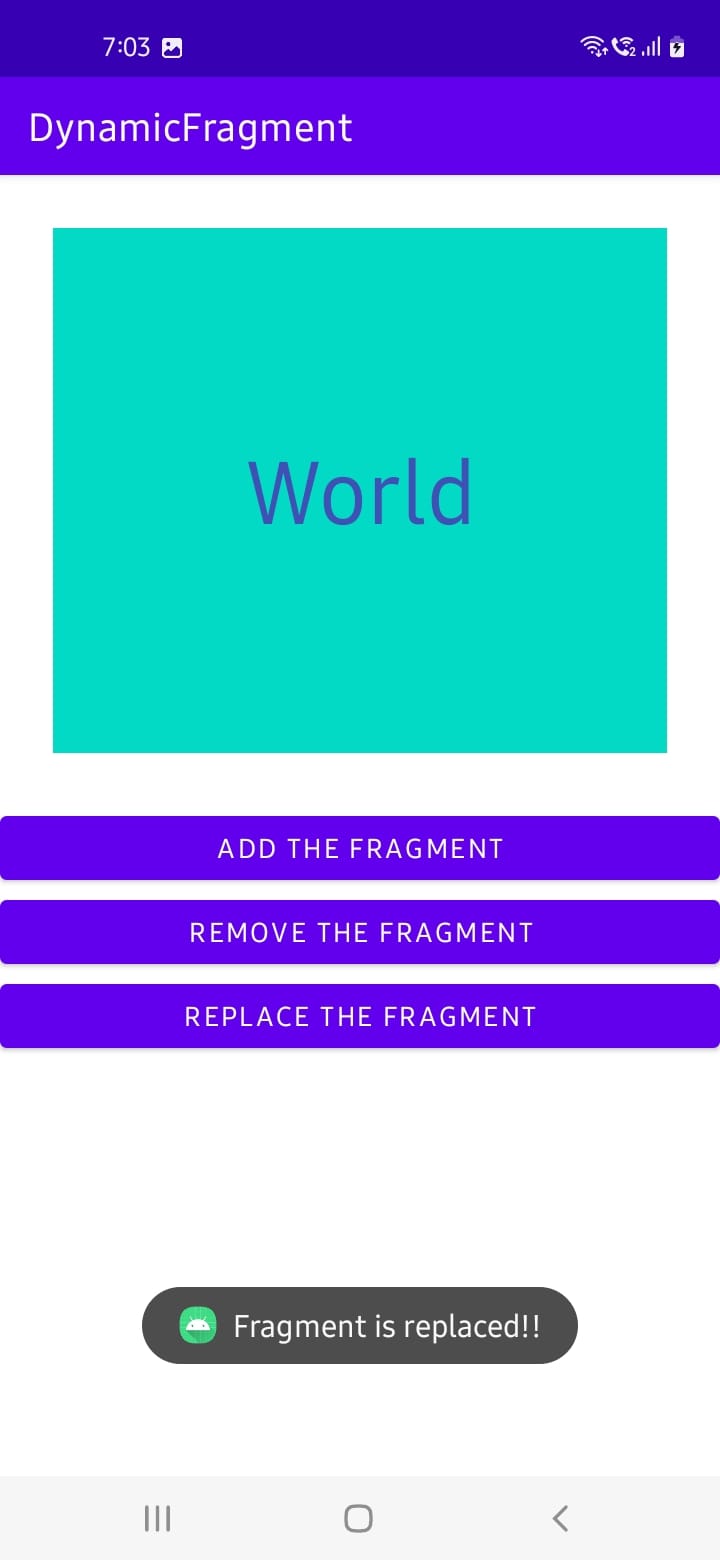
return v;

}

}

**Output:**

**Experiment – 7**

**Aim:** Build mobile application serverless database SQLite Database, Firebase (cloud-hosted database).

**I. SQLite**

**Description:**

SQLite is an open-source relational database that is used to perform database operations on android devices such as storing, manipulating or retrieving persistent data from the database. It is embedded in android by default. So, there is no need to perform any database setup or administration task.

SQLite is a software library that implements SQL database engine that is:

• Self-contained (requires no other components)

• Serverless (requires no server backend)

• Zero-configuration (does not need to be configured for your application)

• Transactional (changes within a single transaction in SQLite either occurs completely or not at all)

**Program (CRUD Operations)**

**activity\_main.xml**

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

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".MainActivity">

<EditText

android:id="@+id/editTextTextPersonName"

android:layout\_width="312dp"

android:layout\_height="68dp"

android:layout\_marginTop="24dp"

android:ems="10"

android:hint="Course"

android:inputType="textPersonName"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.494"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/editTextTextPersonName4" />

<EditText

android:id="@+id/editTextTextPersonName3"

android:layout\_width="312dp"

android:layout\_height="68dp"

android:layout\_marginTop="24dp"

android:ems="10"

android:hint="Name"

android:inputType="textPersonName"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.494"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

<EditText

android:id="@+id/editTextTextPersonName4"

android:layout\_width="312dp"

android:layout\_height="68dp"

android:layout\_marginTop="24dp"

android:ems="10"

android:hint="Section"

android:inputType="textPersonName"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.494"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/editTextTextPersonName3" />

<EditText

android:id="@+id/editTextTextPersonName5"

android:layout\_width="312dp"

android:layout\_height="68dp"

android:layout\_marginTop="24dp"

android:ems="10"

android:hint="Year"

android:inputType="textPersonName"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.494"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/editTextTextPersonName" />

<Button

android:id="@+id/button"

android:layout\_width="209dp"

android:layout\_height="58dp"

android:layout\_marginTop="20dp"

android:onClick="insert\_Data"

android:text="CREATE"

android:textSize="20sp"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/editTextTextPersonName5" />

<Button

android:id="@+id/button2"

android:layout\_width="209dp"

android:layout\_height="58dp"

android:layout\_marginTop="8dp"

android:onClick="update\_Data"

android:text="UPDATE"

android:textSize="20sp"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/button3" />

<Button

android:id="@+id/button3"

android:layout\_width="209dp"

android:layout\_height="58dp"

android:layout\_marginTop="8dp"

android:onClick="delete\_Data"

android:text="DELETE"

android:textSize="20sp"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/button4" />

<Button

android:id="@+id/button4"

android:layout\_width="209dp"

android:layout\_height="58dp"

android:layout\_marginTop="8dp"

android:onClick="view\_Data"

android:text="READ"

android:textSize="20sp"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/button" />

</androidx.constraintlayout.widget.ConstraintLayout>

**MainActivity.java**

package com.example.myapplication;

import androidx.appcompat.app.\*;

import android.database.\*;

import android.os.Bundle;

import android.view.View;

import android.widget.\*;

public class MainActivity extends AppCompatActivity {

//initialize variables

EditText name, age, address, course;

Button insert, update, delete, view;

DBHelper DB;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

//EditTexts UI

name = findViewById(R.id.editTextTextPersonName3);

age = findViewById(R.id.editTextTextPersonName4);

address = findViewById(R.id.editTextTextPersonName5);

course = findViewById(R.id.editTextTextPersonName);

//Buttons UI

insert = findViewById(R.id.button);

update = findViewById(R.id.button2);

delete = findViewById(R.id.button3);

view = findViewById(R.id.button4);

//initialize Database

DB = new DBHelper(this);

}

public void insert\_Data(View v) {

String nameTXT = name.getText().toString();

String ageTXT = age.getText().toString();

String courseTXT = course.getText().toString();

String addressTXT = address.getText().toString();

Boolean checkInsertData = DB.insertuserdata(nameTXT, ageTXT, addressTXT, courseTXT);

if (checkInsertData)

Toast.makeText(MainActivity.this, "Data Inserted", Toast.LENGTH\_SHORT).show();

else

Toast.makeText(MainActivity.this, "Data Not Inserted", Toast.LENGTH\_SHORT).show();

DB.close();

}

public void update\_Data(View v){

String nameTXT = name.getText().toString();

String ageTXT = age.getText().toString();

String addressTXT = address.getText().toString();

String courseTXT = course.getText().toString();

Boolean checkUpdateData = DB.updateuserdata(nameTXT, ageTXT, addressTXT, courseTXT);

if(checkUpdateData)

Toast.makeText(MainActivity.this, "Entry Updated", Toast.LENGTH\_SHORT).show();

else

Toast.makeText(MainActivity.this, "New Entry Not Updated", Toast.LENGTH\_SHORT).show();

DB.close();

}

public void delete\_Data(View v){

String nameTXT = name.getText().toString();

Boolean checkDeleteData = DB.deletedata(nameTXT);

if(checkDeleteData)

Toast.makeText(MainActivity.this, "Entry Deleted", Toast.LENGTH\_SHORT).show();

else

Toast.makeText(MainActivity.this, "Entry Not Deleted", Toast.LENGTH\_SHORT).show();

DB.close();

}

public void view\_Data(View v){

Cursor res = DB.getdata();

if(res.getCount()==0){

Toast.makeText(MainActivity.this, "No Entry Exists", Toast.LENGTH\_SHORT).show();

return;

}

StringBuffer buffer = new StringBuffer();

while(res.moveToNext()){

buffer.append("Name: "+res.getString(0)+"\n");

buffer.append("Section: "+res.getString(1)+"\n");

buffer.append("Course: "+res.getString(3)+"\n");

buffer.append("Year: "+res.getString(2)+"\n\n");

}

//alert pop-up for viewing all data

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

builder.setCancelable(true);

builder.setTitle("User Entries");

builder.setMessage(buffer.toString());

builder.show();

DB.close();

}

}

**DBHelper.java**

package com.example.myapplication;

import android.content.\*;

import android.database.Cursor;

import android.database.sqlite.\*;

public class DBHelper extends SQLiteOpenHelper {

public DBHelper(Context context) {

super(context, "Userdata.db", null, 1);

}

@Override

public void onCreate(SQLiteDatabase DB) {

DB.execSQL("create Table Userdetails(name TEXT primary key, age TEXT, address TEXT, course TEXT)");

}

@Override

public void onUpgrade(SQLiteDatabase DB, int i, int ii) {

DB.execSQL("drop Table if exists Userdetails");

}

public Boolean insertuserdata(String name, String age, String address, String course)

{

SQLiteDatabase DB = this.getWritableDatabase();

ContentValues contentValues = new ContentValues();

contentValues.put("name", name);

contentValues.put("age", age);

contentValues.put("address", address);

contentValues.put("course", course);

long result=DB.insert("Userdetails", null, contentValues);

if(result==-1){

return false;

}else{

return true;

}

}

public Boolean updateuserdata(String name, String age, String address, String course)

{

SQLiteDatabase DB = this.getWritableDatabase();

ContentValues contentValues = new ContentValues();

contentValues.put("age", age);

contentValues.put("address", address);

contentValues.put("course", course);

Cursor cursor = DB.rawQuery("Select \* from Userdetails where name = ?", new String[]{name});

if (cursor.getCount() > 0) {

long result = DB.update("Userdetails", contentValues, "name=?", new String[]{name});

if (result == -1) {

return false;

} else {

return true;

}

} else {

return false;

}

}

public Boolean deletedata (String name)

{

SQLiteDatabase DB = this.getWritableDatabase();

Cursor cursor = DB.rawQuery("Select \* from Userdetails where name = ?", new String[]{name});

if (cursor.getCount() > 0) {

long result = DB.delete("Userdetails", "name=?", new String[]{name});

if (result == -1) {

return false;

} else {

return true;

}

} else {

return false;

}

}

public Cursor getdata ()

{

SQLiteDatabase DB = this.getWritableDatabase();

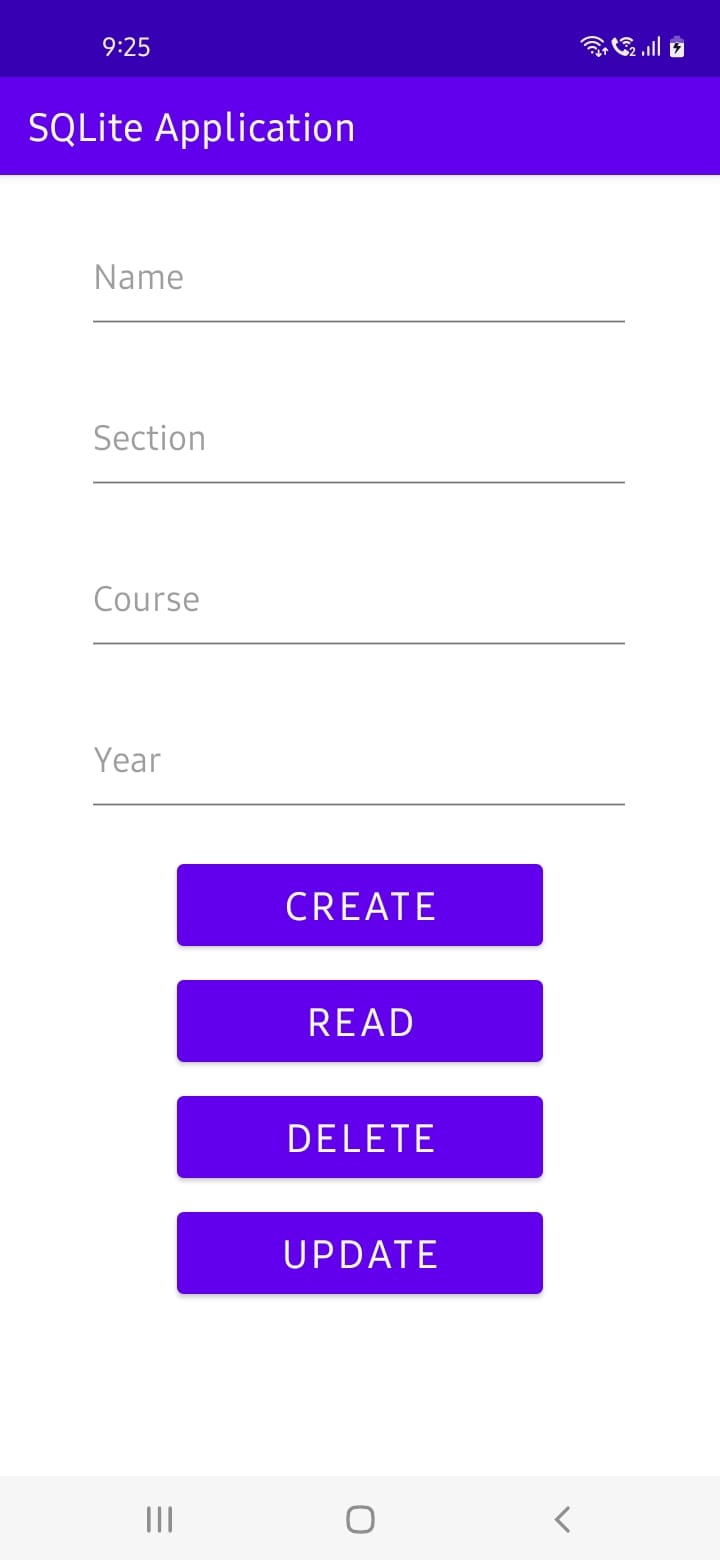
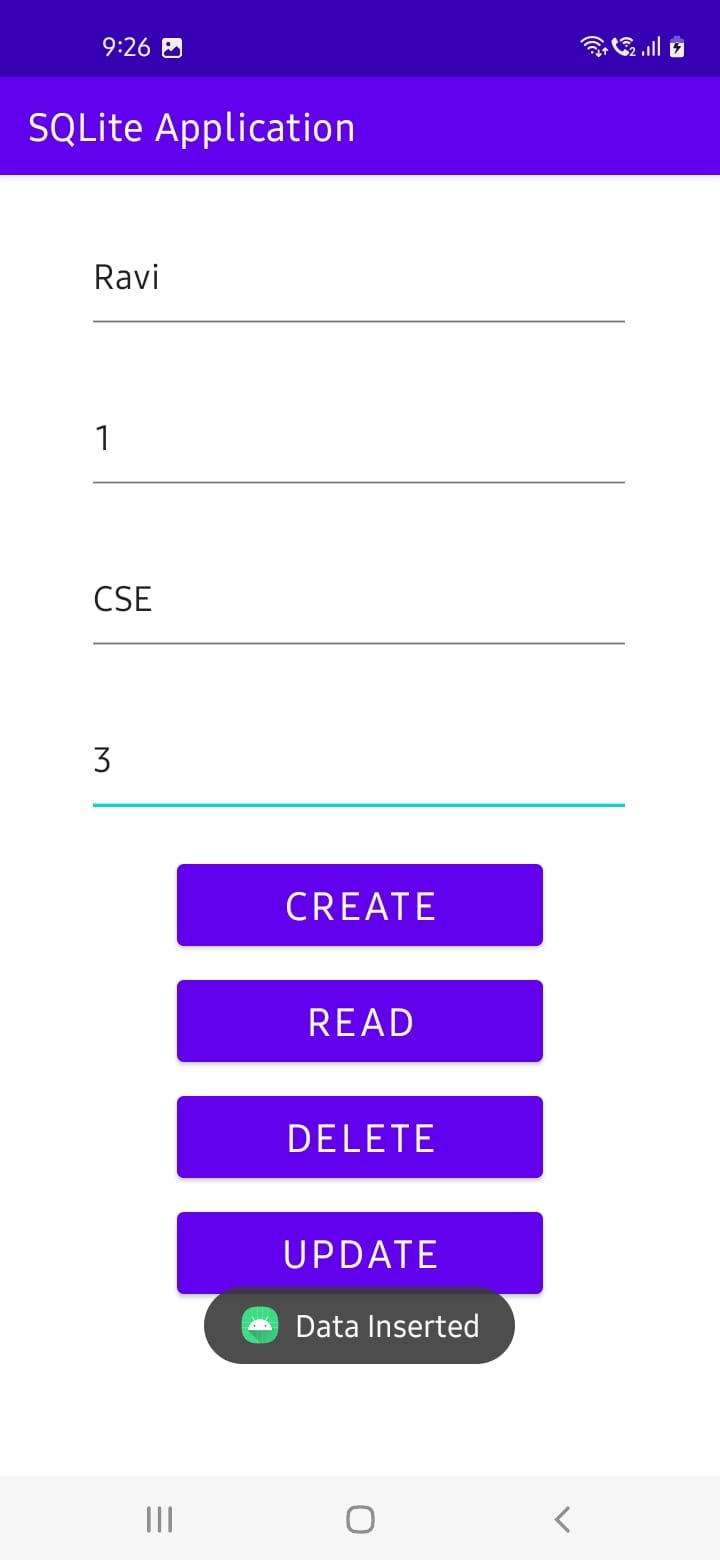
Cursor cursor = DB.rawQuery("Select \* from Userdetails", null);

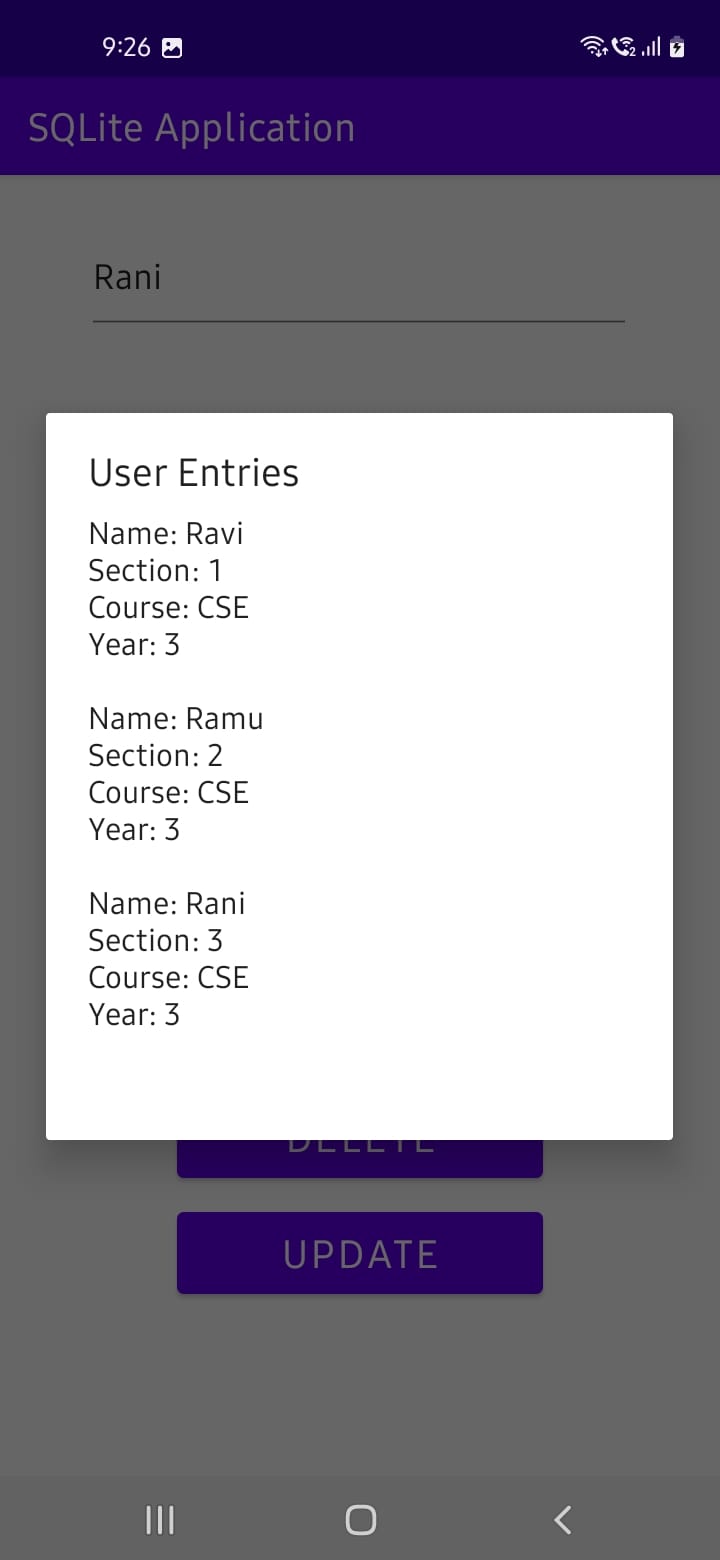
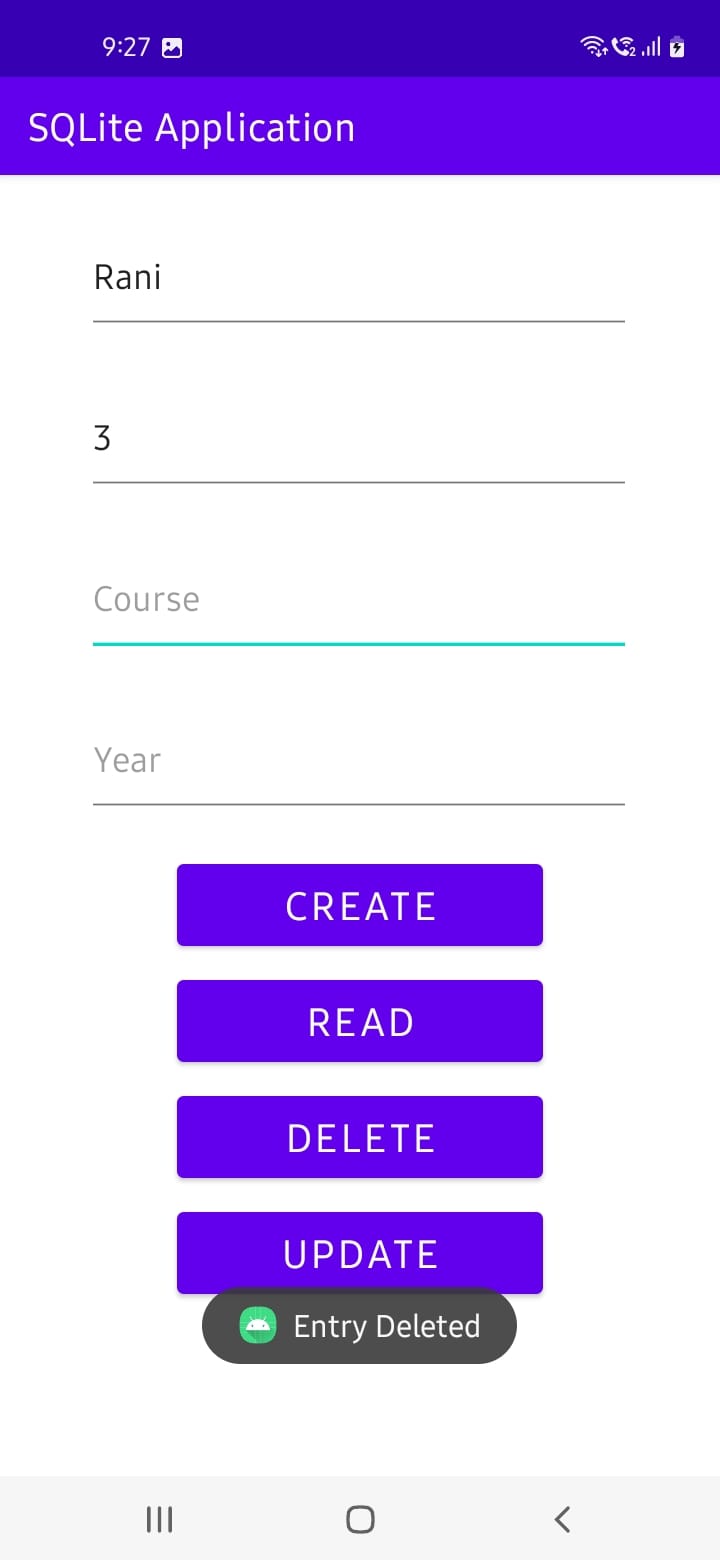
return cursor;

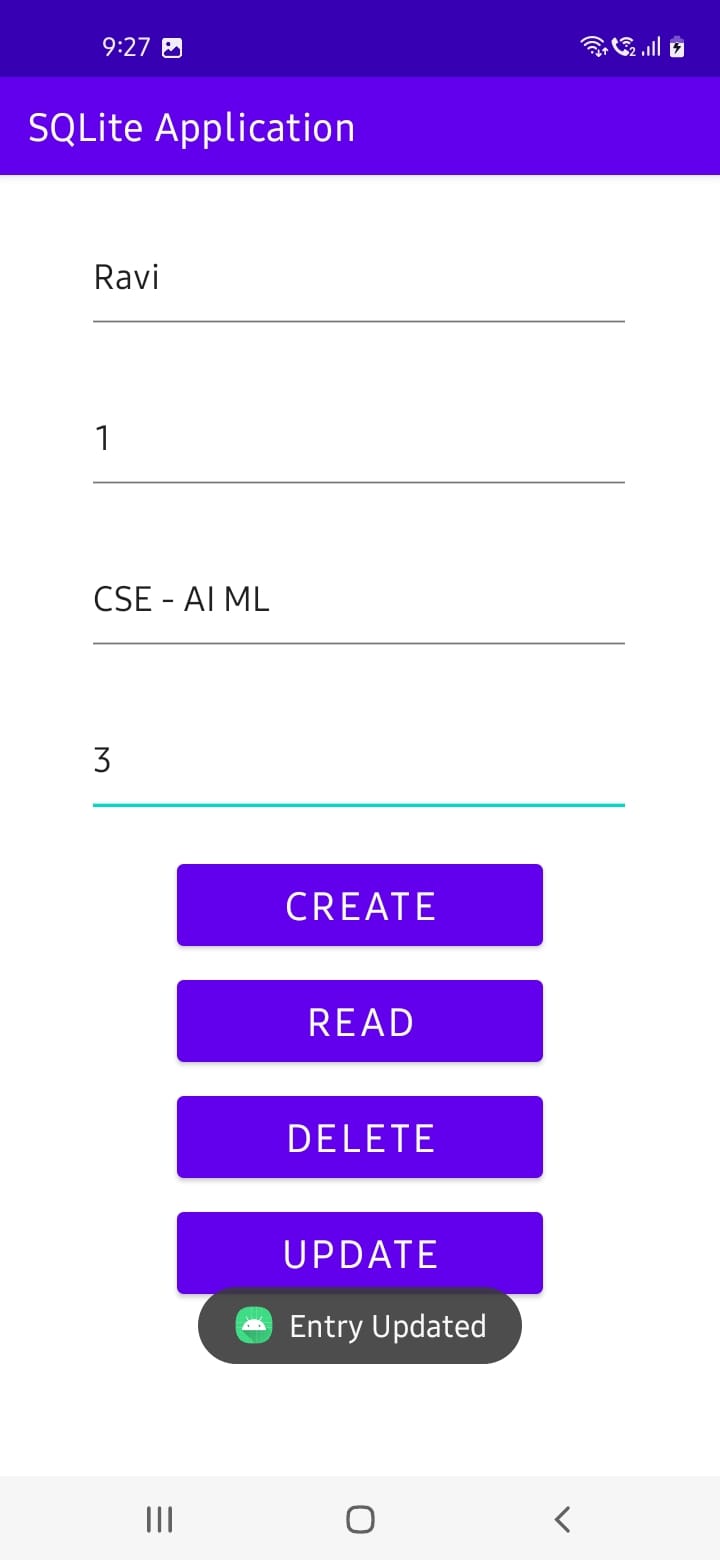
}

}

**Output:**



**II. Firebase**

**Description:**

The Firebase Realtime Database is a cloud-hosted database in which data is stored as JSON. The data is synchronized in real-time to every connected client. All of our clients share one Realtime Database instances and automatically receive updates with the newest data, when we build cross-platform applications with our iOS, and JavaScript SDKs.

The Firebase Realtime Database is a NoSQL database from which we can store and sync the data between our users in real-time. It is a big JSON object which the developers can manage in real-time. By using a single API, the Firebase database provides the application with the current value of the data and updates to that data. Real-time syncing makes it easy for our users to access their data from any device, be it web or mobile.

**Program:** (**CRUD Operations using Firebase)**

activity\_main.xml

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

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

tools:context=".MainActivity">

<androidx.recyclerview.widget.RecyclerView

android:id="@+id/recyclerView"

android:layout\_width="match\_parent"

android:layout\_height="0dp"

android:layout\_weight="1" />

<androidx.appcompat.widget.AppCompatButton

android:id="@+id/buttonAdd"

android:layout\_width="match\_parent"

android:layout\_height="50dp"

android:background="@android:color/holo\_green\_dark"

android:text="ADD NEW USER"

android:textColor="@android:color/white"

android:textStyle="bold"/>

</LinearLayout>

**alert\_dialog\_add\_new\_user.xml**

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

<androidx.cardview.widget.CardView xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

android:layout\_width="300dp"

android:layout\_height="wrap\_content"

android:layoutDirection="ltr"

app:cardBackgroundColor="@android:color/white"

app:cardCornerRadius="5dp">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

android:padding="10dp">

<EditText

android:id="@+id/textName"

android:layout\_width="match\_parent"

android:layout\_height="40dp"

android:layout\_gravity="center"

android:layout\_margin="3dp"

android:gravity="center"

android:hint="User name"

android:padding="5dp"

android:textColor="@android:color/black"

android:textColorHint="@android:color/black"

android:textSize="18sp"

android:textStyle="bold" />

<EditText

android:id="@+id/textEmail"

android:layout\_width="match\_parent"

android:layout\_height="40dp"

android:layout\_gravity="center"

android:layout\_margin="3dp"

android:gravity="center"

android:hint="User email"

android:padding="5dp"

android:textColor="@android:color/black"

android:textColorHint="@android:color/black"

android:textSize="18sp"

android:textStyle="bold" />

<EditText

android:id="@+id/textCountry"

android:layout\_width="match\_parent"

android:layout\_height="40dp"

android:layout\_gravity="center"

android:layout\_margin="3dp"

android:gravity="center"

android:hint="User Country"

android:padding="5dp"

android:textColor="@android:color/black"

android:textColorHint="@android:color/black"

android:textSize="18sp"

android:textStyle="bold" />

<androidx.appcompat.widget.AppCompatButton

android:id="@+id/buttonAdd"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_marginTop="15sp"

android:background="@android:color/holo\_blue\_dark"

android:text="Add"

android:textColor="@android:color/black"

android:textSize="20dp"

android:textStyle="bold" />

<androidx.appcompat.widget.AppCompatButton

android:id="@+id/buttonCancel"

android:layout\_width="match\_parent"

android:layout\_height="50sp"

android:layout\_marginTop="5dp"

android:background="@android:color/holo\_red\_light"

android:text="Exit"

android:textColor="@android:color/black"

android:textSize="20dp"

android:textStyle="bold" />

</LinearLayout>

</androidx.cardview.widget.CardView>

**user\_item.xml**

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

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:padding="4sp"

android:layout\_margin="4sp"

android:orientation="vertical"

android:layout\_height="wrap\_content">

<TextView

android:id="@+id/textName"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:textColor="@android:color/black"

android:textStyle="bold"

android:textSize="18sp"

android:text="User name"/>

<TextView

android:id="@+id/textEmail"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:textColor="@android:color/black"

android:textStyle="italic"

android:textSize="18sp"

android:text="User email"/>

<TextView

android:id="@+id/textCountry"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:textColor="@android:color/black"

android:textStyle="italic"

android:textSize="18sp"

android:text="User country"/>

<LinearLayout

android:layout\_width="match\_parent"

android:orientation="horizontal"

android:layout\_margin="4sp"

android:padding="4sp"

android:layout\_height="wrap\_content">

<androidx.appcompat.widget.AppCompatButton

android:id="@+id/buttonDelete"

android:layout\_width="0dp"

android:layout\_height="40dp"

android:layout\_weight="1"

android:layout\_margin="3dp"

android:background="@android:color/holo\_red\_light"

android:text="DELETE"

android:textColor="@android:color/black"/>

<androidx.appcompat.widget.AppCompatButton

android:id="@+id/buttonUpdate"

android:layout\_width="0dp"

android:layout\_height="40dp"

android:layout\_weight="1"

android:layout\_margin="3dp"

android:background="@android:color/holo\_blue\_dark"

android:text="UPDATE"

android:textColor="@android:color/white"/>

</LinearLayout>

</LinearLayout>

**view\_dialog\_confirm\_delete.xml**

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

<androidx.cardview.widget.CardView xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

android:layout\_width="300dp"

android:layout\_height="wrap\_content"

android:layoutDirection="ltr"

app:cardBackgroundColor="@android:color/white"

app:cardCornerRadius="5dp">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

android:padding="10dp">

<TextView

android:id="@+id/textTitle"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:gravity="center"

android:padding="5dp"

android:text="Are you sure ?"

android:textColor="@android:color/black"

android:textSize="18sp"

android:textStyle="bold" />

<androidx.appcompat.widget.AppCompatButton

android:id="@+id/buttonDelete"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_marginTop="15sp"

android:background="@android:color/holo\_red\_light"

android:text="Yes, Delete"

android:textColor="@android:color/black"

android:textSize="20dp"

android:textStyle="bold" />

<androidx.appcompat.widget.AppCompatButton

android:id="@+id/buttonCancel"

android:layout\_width="match\_parent"

android:layout\_height="50sp"

android:layout\_marginTop="5dp"

android:background="@android:color/holo\_blue\_dark"

android:text="No, Exit"

android:textColor="@android:color/black"

android:textSize="20dp"

android:textStyle="bold" />

</LinearLayout>

</androidx.cardview.widget.CardView>

**MainActivity.java**

package com.example.realtimedbaddreadupdatedelete;

import androidx.annotation.\*;

import androidx.appcompat.app.AppCompatActivity;

import androidx.recyclerview.widget.\*;

import android.annotation.SuppressLint;

import android.app.\*;

import android.content.\*;

import android.graphics.\*;

import android.graphics.drawable.ColorDrawable;

import android.os.Bundle;

import android.view.\*;

import android.widget.\*;

import com.google.firebase.database.\*;

import java.util.\*;

public class MainActivity extends AppCompatActivity {

DatabaseReference databaseReference;

RecyclerView recyclerView;

ArrayList<UsersItem> usersItemArrayList;

UsersRecyclerAdapter adapter;

Button buttonAdd;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

FirebaseDatabase.getInstance().setPersistenceEnabled(true); // work offline

Objects.requireNonNull(getSupportActionBar()).hide();

databaseReference = FirebaseDatabase.getInstance().getReference();

recyclerView = findViewById(R.id.recyclerView);

recyclerView.setHasFixedSize(true);

recyclerView.setLayoutManager(new LinearLayoutManager(this));

usersItemArrayList = new ArrayList<>();

buttonAdd = findViewById(R.id.buttonAdd);

buttonAdd.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

ViewDialogAdd viewDialogAdd = new ViewDialogAdd();

viewDialogAdd.showDialog(MainActivity.this);

}

});

readData();

}

private void readData() {

databaseReference.child("USERS").orderByChild("userName").addValueEventListener(new ValueEventListener() {

@SuppressLint("NotifyDataSetChanged")

@Override

public void onDataChange(@NonNull DataSnapshot snapshot) {

usersItemArrayList.clear();

for (DataSnapshot dataSnapshot : snapshot.getChildren()) {

UsersItem users = dataSnapshot.getValue(UsersItem.class);

usersItemArrayList.add(users);

}

adapter = new UsersRecyclerAdapter(MainActivity.this, usersItemArrayList);

recyclerView.setAdapter(adapter);

adapter.notifyDataSetChanged();

}

@Override

public void onCancelled(@NonNull DatabaseError error) {

}

});

}

public class ViewDialogAdd {

public void showDialog(Context context) {

final Dialog dialog = new Dialog(context);

dialog.requestWindowFeature(Window.FEATURE\_NO\_TITLE);

dialog.setCancelable(false);

dialog.setContentView(R.layout.alert\_dialog\_add\_new\_user);

EditText textName = dialog.findViewById(R.id.textName);

EditText textEmail = dialog.findViewById(R.id.textEmail);

EditText textCountry = dialog.findViewById(R.id.textCountry);

Button buttonAdd = dialog.findViewById(R.id.buttonAdd);

Button buttonCancel = dialog.findViewById(R.id.buttonCancel);

buttonAdd.setText("ADD");

buttonCancel.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

dialog.dismiss();

}

});

buttonAdd.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

String id = "user" + new Date().getTime();

String name = textName.getText().toString();

String email = textEmail.getText().toString();

String country = textCountry.getText().toString();

if (name.isEmpty() || email.isEmpty() || country.isEmpty()) {

Toast.makeText(context, "Please Enter All data...", Toast.LENGTH\_SHORT).show();

} else {

databaseReference.child("USERS").child(id).setValue(new UsersItem(id, name, email, country));

Toast.makeText(context, "DONE!", Toast.LENGTH\_SHORT).show();

dialog.dismiss();

}

}

});

dialog.getWindow().setBackgroundDrawable(new ColorDrawable(Color.TRANSPARENT));

dialog.show();

}

}

}

**UsersRecyclerAdapter.java**

package com.example.realtimedbaddreadupdatedelete;

import android.app.Dialog;

import android.content.Context;

import android.graphics.Color;

import android.graphics.drawable.ColorDrawable;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.view.Window;

import android.widget.Button;

import android.widget.EditText;

import android.widget.TextView;

import android.widget.Toast;

import androidx.annotation.NonNull;

import androidx.recyclerview.widget.RecyclerView;

import com.google.firebase.database.DatabaseReference;

import com.google.firebase.database.FirebaseDatabase;

import java.util.ArrayList;

public class UsersRecyclerAdapter extends RecyclerView.Adapter<UsersRecyclerAdapter.ViewHolder> {

Context context;

ArrayList<UsersItem> usersItemArrayList;

DatabaseReference databaseReference;

public UsersRecyclerAdapter(Context context, ArrayList<UsersItem> usersItemArrayList) {

this.context = context;

this.usersItemArrayList = usersItemArrayList;

databaseReference = FirebaseDatabase.getInstance().getReference();

}

@NonNull

@Override

public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {

LayoutInflater layoutInflater = LayoutInflater.from(context);

View view = layoutInflater.inflate(R.layout.user\_item, parent, false);

return new ViewHolder(view);

}

@Override

public void onBindViewHolder(@NonNull ViewHolder holder, int position) {

UsersItem users = usersItemArrayList.get(position);

holder.textName.setText("Name : " + users.getUserName());

holder.textEmail.setText("Email : " + users.getUserEmail());

holder.textCountry.setText("Country : " + users.getUserCountry());

holder.buttonUpdate.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

ViewDialogUpdate viewDialogUpdate = new ViewDialogUpdate();

viewDialogUpdate.showDialog(context, users.getUserID(), users.getUserName(), users.getUserEmail(), users.getUserCountry());

}

});

holder.buttonDelete.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

ViewDialogConfirmDelete viewDialogConfirmDelete = new ViewDialogConfirmDelete();

viewDialogConfirmDelete.showDialog(context, users.getUserID());

}

});

}

@Override

public int getItemCount() {

return usersItemArrayList.size();

}

public static class ViewHolder extends RecyclerView.ViewHolder {

TextView textName;

TextView textEmail;

TextView textCountry;

Button buttonDelete;

Button buttonUpdate;

public ViewHolder(@NonNull View itemView) {

super(itemView);

textName = itemView.findViewById(R.id.textName);

textEmail = itemView.findViewById(R.id.textEmail);

textCountry = itemView.findViewById(R.id.textCountry);

buttonDelete = itemView.findViewById(R.id.buttonDelete);

buttonUpdate = itemView.findViewById(R.id.buttonUpdate);

}

}

public class ViewDialogUpdate {

public void showDialog(Context context, String id, String name, String email, String country) {

final Dialog dialog = new Dialog(context);

dialog.requestWindowFeature(Window.FEATURE\_NO\_TITLE);

dialog.setCancelable(false);

dialog.setContentView(R.layout.alert\_dialog\_add\_new\_user);

EditText textName = dialog.findViewById(R.id.textName);

EditText textEmail = dialog.findViewById(R.id.textEmail);

EditText textCountry = dialog.findViewById(R.id.textCountry);

textName.setText(name);

textEmail.setText(email);

textCountry.setText(country);

Button buttonUpdate = dialog.findViewById(R.id.buttonAdd);

Button buttonCancel = dialog.findViewById(R.id.buttonCancel);

buttonUpdate.setText("UPDATE");

buttonCancel.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

dialog.dismiss();

}

});

buttonUpdate.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

String newName = textName.getText().toString();

String newEmail = textEmail.getText().toString();

String newCountry = textCountry.getText().toString();

if (name.isEmpty() || email.isEmpty() || country.isEmpty()) {

Toast.makeText(context, "Please Enter All data...", Toast.LENGTH\_SHORT).show();

} else {

if (newName.equals(name) && newEmail.equals(email) && newCountry.equals(country)) {

Toast.makeText(context, "you don't change anything", Toast.LENGTH\_SHORT).show();

} else {

databaseReference.child("USERS").child(id).setValue(new UsersItem(id, newName, newEmail, newCountry));

Toast.makeText(context, "User Updated successfully!", Toast.LENGTH\_SHORT).show();

dialog.dismiss();

}

}

}

});

dialog.getWindow().setBackgroundDrawable(new ColorDrawable(Color.TRANSPARENT));

dialog.show();

}

}

public class ViewDialogConfirmDelete {

public void showDialog(Context context, String id) {

final Dialog dialog = new Dialog(context);

dialog.requestWindowFeature(Window.FEATURE\_NO\_TITLE);

dialog.setCancelable(false);

dialog.setContentView(R.layout.view\_dialog\_confirm\_delete);

Button buttonDelete = dialog.findViewById(R.id.buttonDelete);

Button buttonCancel = dialog.findViewById(R.id.buttonCancel);

buttonCancel.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

dialog.dismiss();

}

});

buttonDelete.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

databaseReference.child("USERS").child(id).removeValue();

Toast.makeText(context, "User Deleted successfully!", Toast.LENGTH\_SHORT).show();

dialog.dismiss();

}

});

dialog.getWindow().setBackgroundDrawable(new ColorDrawable(Color.TRANSPARENT));

dialog.show();

}

}

}

**UsersItem.java**

package com.example.realtimedbaddreadupdatedelete;

public class UsersItem {

String userID;

String userName;

String userEmail;

String userCountry;

public UsersItem() {

}

public UsersItem(String userID, String userName, String userEmail, String userCountry) {

this.userID = userID;

this.userName = userName;

this.userEmail = userEmail;

this.userCountry = userCountry;

}

public String getUserID() {

return userID;

}

public void setUserID(String userID) {

this.userID = userID;

}

public String getUserName() {

return userName;

}

public void setUserName(String userName) {

this.userName = userName;

}

public String getUserEmail() {

return userEmail;

}

public void setUserEmail(String userEmail) {

this.userEmail = userEmail;

}

public String getUserCountry() {

return userCountry;

}

public void setUserCountry(String userCountry) {

this.userCountry = userCountry;

}

}

**Output:**

