Date: 21-12-2024

Experiment – 1

Aim:

Build mobile application based on the concept activity life cycle with custom Toast.

Description:

This experiment demonstrates the **Android Activity Lifecycle** by displaying a **Custom Toast** message each time an activity transitions through different lifecycle stages.

- **onCreate()** – Initializes the activity when it is first created.
- **onStart()** – Prepares the activity as it becomes visible to the user.
- **onResume()** – Activates interaction when the activity is in the foreground.
- **onPause()** – Triggers when the activity is partially visible or in the background.
- **onStop()** – Stops execution when the activity is completely hidden.
- **onRestart()** – Restores the activity just before it becomes visible again.
- **onDestroy()** – Cleans up resources before the activity is permanently closed.

Programs:

MainActivity.java

```
package com.example.a22501a0557_customtoast;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;
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 showToast(View v) {
        Toast.makeText(this, "Hi I'm 557's Toast", Toast.LENGTH_LONG).show();
    }

    public void showCustomToast(View v) {
        LayoutInflater lf = getLayoutInflater();
        View layout = lf.inflate(R.layout.cutomtoastlayout, (ViewGroup)
findViewById(R.id.ll_ctoast));

        ImageView img = layout.findViewById(R.id.imgv_ctoast);
        img.setImageResource(R.drawable.img); // Set the image from drawable

        TextView txt = layout.findViewById(R.id.txtv_ctoast);
        txt.setText("Hi, I'm Ram, welcome to Custom Toast");

        Toast toast = new Toast(getApplicationContext());
        toast.setDuration(Toast.LENGTH_LONG);
        toast.setView(layout);
        toast.show();
    }

    @Override
```

```
    protected void onStart() {
        super.onStart();
        Toast.makeText(this, "APP HAS STARTED", Toast.LENGTH_LONG).show();
    }

    @Override
    protected void onResume() {
        super.onResume();
        Toast.makeText(this, "APP IS NOW RUNNING", Toast.LENGTH_LONG).show();
    }

    @Override
    protected void onPause() {
        super.onPause();
        Toast.makeText(this, "APP IS NOW PAUSED", Toast.LENGTH_LONG).show();
    }

    @Override
    protected void onStop() {
        super.onStop();
        Toast.makeText(this, "APP IS NOW STOPPED", Toast.LENGTH_LONG).show();
    }

    @Override
    protected void onRestart() {
        super.onRestart();
        Toast.makeText(this, "APP IS RESTARTING", Toast.LENGTH_LONG).show();
    }
}
```

activity_main.xml

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

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

```
        android:layout_marginTop="300dp"
        android:onClick="showToast"
        android:text="Press to Toast"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.548"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/ctoast_btn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="400dp"
        android:onClick="showCustomToast"
        android:text="Press to custom Toast"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.606"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

customtoastlayout.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:id="@+id/ll_ctoast"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <ImageView
        android:id="@+id/imgv_ctoast"
        android:layout_width="387dp"
        android:layout_height="196dp"
        android:layout_weight="1"
        app:srcCompat="@drawable/img" />
    <TextView
        android:id="@+id/txtv_ctoast"
        android:layout_width="wrap_content"
        android:layout_height="198dp"
        android:layout_weight="1"
        android:text="Hi, I'm 22501A0557. Welcome to Custom Toast!!!"
        android:textSize="25sp" />
</LinearLayout>
```

Outputs:

Create New Project with “No Activity”

Naming it and Selecting Java as Language

Building Gradle

Creating new Empty Views Activity

Selecting it as Launcher Activity

We now have MainActivity java and xml files

Selecting the button from the tables

Dragging and Dropping the Button

Renaming the Button id to “toast_btn”

Respective xml code we get automatically

We can import the class, which we can have inbuilt

After importing that class, we get no error

Further Editing the button

Editing the java file

We now try to verify the button and it's working

We successfully getting toast for starting

Successful toast

Successful toast

Custom Toast:

Creating an new xml file

Filling the necessary required fields

Editing the new xml

Editing the new xml

Editing the new xml

Editing the new xml

Editing the new xml

Editing the new xml

Successfully rerunning the app

App representing the Custom Toast

Date: 04-01-2025

Experiment – 2

Aim:

Build mobile application using different layouts (use any 3 layouts)

Description:

This experiment explores **four different Android layouts**—**LinearLayout, TableLayout, FrameLayout, and RelativeLayout**—to understand their structure, positioning, and usability in mobile UI design.

1. **LinearLayout** – Aligns child elements **vertically or horizontally** in a linear sequence.
2. **TableLayout** – Organizes UI components **in rows and columns**, like a table structure.
3. **FrameLayout** – Stacks views **on top of each other**, useful for overlays or fragment loading.
4. **RelativeLayout** – Positions elements **relative to each other** or to the parent container for flexible UI

Programs:

MainActivity.java

```
package com.example.a22501a0557_layouts;

import androidx.appcompat.app.AppCompatActivity;

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

public class MainActivity extends AppCompatActivity {

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

    public void showLLActivity(View v) {
        Intent i = new Intent(this, LinearLayoutActivity.class);
        startActivity(i);
    }

    public void showTLActivity(View v) {
```

```
        Intent i = new Intent(this, TableLayoutActivity.class);
        startActivity(i);
    }

    public void showFLActivity(View v) {
        Intent i = new Intent(this, FrameLayoutActivity.class);
        startActivity(i);
    }

    public void showRLActivity(View v) {
        Intent i = new Intent(this, RelativeLayoutActivity.class);
        startActivity(i);
    }
}
```

activity_main.xml

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

    <TextView
        android:id="@+id/titletv"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Layouts App"
        android:textSize="40sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.497"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.095" />

    <Button
        android:id="@+id/llbtn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="100dp"
        android:onClick="showLLActivity"
```

```
        android:text="Linear Layout"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/titletv" />

    <Button
        android:id="@+id/tlbtn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="60dp"
        android:onClick="showTLActivity"
        android:text="Table Layout"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/llbtn" />

    <Button
        android:id="@+id/flbtn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="60dp"
        android:onClick="showFLActivity"
        android:text="Frame Layout"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/tlbtn" />

    <Button
        android:id="@+id/rlbtn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="60dp"
        android:onClick="showRLActivity"
        android:text="Relative Layout"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/flbtn" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

LinearLayoutActivity.java

```
package com.example.a22501a0557_layouts;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.ImageView;
import android.widget.LinearLayout;

public class LinearLayoutActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_linear_layout);
        ImageView imageView1 = findViewById(R.id.imageView1);
        ImageView imageView2 = findViewById(R.id.imageView2);

        imageView1.setImageResource(R.drawable.img);
        imageView2.setImageResource(R.drawable.img_1);
    }

    public void changeOrientation(View v) {
        LinearLayout ll = findViewById(R.id.ll);
        if (ll.getOrientation() == LinearLayout.HORIZONTAL) {
            ll.setOrientation(LinearLayout.VERTICAL);
        } else {
            ll.setOrientation(LinearLayout.HORIZONTAL);
        }
    }
}
```

activity_linear_layout.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/ll"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal"
    android:padding="20dp">

    <!-- Cat Image -->
    <ImageView
        android:id="@+id/imageView1"
```

```
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:src="@drawable/img"
        android:contentDescription="Cat Image"/>

    <!-- Change Orientation Button -->
    <Button
        android:layout_width="192dp"
        android:layout_height="wrap_content"
        android:onClick="changeOrientation"
        android:text="Change Orientation" />

    <!-- Check Icon -->
    <ImageView
        android:id="@+id/imageView2"
        android:layout_width="88dp"
        android:layout_height="100dp"
        android:contentDescription="Check Icon"
        android:src="@drawable/img_1" />
</LinearLayout>
```

TableLayoutActivity.java

```
package com.example.a22501a0557_layouts;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

public class TableLayoutActivity extends AppCompatActivity {

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

activity_table_layout.xml

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

<!-- Row 1 -->
<TableRow
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <Button
        android:id="@+id/btn1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="1" />

    <Button
        android:id="@+id/btn2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="2" />

    <Button
        android:id="@+id/btn3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="3" />
</TableRow>

<!-- Row 2 -->
<TableRow
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <Button
        android:id="@+id/btn4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="4" />

    <Button
        android:id="@+id/btn5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="5" />

    <Button
```

```
        android:id="@+id/btn6"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="6" />
    </TableRow>

    <!-- Row 3 -->
    <TableRow
      android:layout_width="match_parent"
      android:layout_height="wrap_content">

      <Button
        android:id="@+id/btn7"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="7" />

      <Button
        android:id="@+id/btn8"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="8" />

      <Button
        android:id="@+id/btn9"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="9" />
    </TableRow>

</TableLayout>
```

FrameLayoutActivity.java

```
package com.example.a22501a0557_layouts;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

public class FrameLayoutActivity extends AppCompatActivity {

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

```
        setContentView(R.layout.activity_frame_layout);
    }
}
```

activity_frame_layout.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"
    tools:context=".FrameLayoutActivity">

    <!-- TextView for the title -->
    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Welcome to PVPSIT 557"
        android:textSize="30sp"
        android:textStyle="bold"
        android:textColor="@android:color/black"
        android:layout_gravity="top|center"
        android:textAlignment="center"/>

    <!-- ImageView for the logo -->
    <ImageView
        android:id="@+id/pvp_imgv"
        android:layout_width="match_parent"
        android:layout_height="479dp"
        android:layout_gravity="center"
        android:scaleType="fitCenter"
        android:src="@drawable/img_2" />

</FrameLayout>
```

RelativeLayoutActivity.java

```
package com.example.a22501a0557_layouts;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

public class RelativeLayoutActivity extends AppCompatActivity {
```

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

activity_relative_layout.xml

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

    <!-- TextView at the top -->
    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Welcome to Relative Layout"
        android:textSize="20sp"
        android:textStyle="bold"
        android:textColor="@android:color/black"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="50dp"/>

    <!-- Button below TextView -->
    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Click Me"
        android:layout_below="@id/textView"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="20dp"/>

</RelativeLayout>
```

Outputs:

Creating a new Project with No Activity

Filling the necessary details

Creating an Empty Views Activity

Selecting the Launcher Activity for the main activity

Editing the Main java file

Editing the xml file

Editing the xml file

Editing the xml file

Creating a new activity

Editing the java file

Editing the xml file

Successfully running the app

We find out there is nothing yet in second activity

Editing the xml file

Editing the xml file

Editing the xml file

Editing the xml file

Verifying the changes made

Verifying the successful working of the feature

Editing the xml file

Creating a new Activity

Creating the other activity

Added New button in our App

Verifying the changes

Adding a new button

Creating a new Activity

Filling the necessary details

Added new button

Verifying the necessary changes

Creating a new Empty Views Activity

Filling the necessary details

Editing the xml file

Adding the new Button

Successfully verified the changes

Date: 08-02-2025

Experiment – 3

Aim:

Build mobile application using different dialogs (use any 2 dialogs)

Description:

A **Dialog** is a small interactive window that does not fill the entire screen and prompts the user to make a decision or provide input before proceeding.

Types of Dialogs in Android

- **Alert Dialog**: Displays a message with action buttons like OK, Cancel, or Retry for user confirmation.
- **DatePicker Dialog**: Provides a calendar-based interface for selecting a date.
- **TimePicker Dialog**: Allows the user to choose a specific time using a clock-style UI.
- **Custom Dialog**: A user-defined dialog with custom layouts, images, and input fields, offering more flexibility.

Programs:

MainActivity.java

```
package com.example.a22501a0557_dialogs;

import androidx.appcompat.app.AppCompatActivity;

import android.app.AlertDialog;
import android.app.DatePickerDialog;
import android.app.TimePickerDialog;
import android.content.DialogInterface;
import android.os.Bundle;
import android.view.View;
import android.widget.DatePicker;
import android.widget.TimePicker;
import android.widget.Toast;

import java.sql.Time;
import java.util.Calendar;

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

```
    public void showAlertDialog(View v) {
        AlertDialog.Builder builder = new AlertDialog.Builder(this);

        builder.setTitle("Alert");
        builder.setMessage("Are you sure you want to continue?");
        builder.setIcon(R.drawable.img);

        builder.setPositiveButton("OK", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                Toast.makeText(MainActivity.this, "You clicked OK!",
Toast.LENGTH_SHORT).show();
            }
        });

        builder.setNegativeButton("Cancel", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                dialog.dismiss();
            }
        });

        builder.create().show();
    }
    public void showTimeDialog(View v) {
        Calendar cal = Calendar.getInstance();
        int hr = cal.get(Calendar.HOUR_OF_DAY);
        int min = cal.get(Calendar.MINUTE);

        TimePickerDialog tpd = new TimePickerDialog(this, new
TimePickerDialog.OnTimeSetListener() {
            @Override
            public void onTimeSet(TimePicker timePicker, int selectedHour, int selectedMinute)
{
                Toast.makeText(MainActivity.this, selectedHour + "hr : " + selectedMinute +
"min", Toast.LENGTH_LONG).show();
            }
        }, hr, min, true);
        tpd.show();
    }
    public void showCalenderDialog(View v){
        Calendar cal = Calendar.getInstance();
        int dm = cal.get(Calendar.MONTH);
        int dd = cal.get(Calendar.DAY_OF_MONTH);
```

```
        int dy = cal.get(Calendar.YEAR);
        DatePickerDialog dpd = new DatePickerDialog(this, new
DatePickerDialog.OnDateSetListener() {
            @Override
            public void onDateSet(DatePicker datePicker, int i, int i1, int i2) {
                Toast.makeText(MainActivity.this,
i2+"/"+i1+"/"+i,Toast.LENGTH_LONG).show();
            }
        },dy,dm,dd);
        dpd.setTitle("Pick a date please:");
        dpd.setMessage("Hello I'm GVSR");
        dpd.show();
    }
}
```

activity_main.xml

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

    <Button
        android:id="@+id/dbtn1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="300dp"
        android:onClick="showAlertDialog"
        android:text="alert"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/dbtn2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="showTimeDialog"
        android:text="Time"
```

```
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/dbtn1"
        app:layout_constraintVertical_bias="0.188" />

    <Button
        android:id="@+id/dbtn3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="showCalenderDialog"
        android:text="Calendar"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/dbtn2"
        app:layout_constraintVertical_bias="0.316" />

    <TextView
        android:id="@+id/dtitle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Dialogs App"
        android:textColor="@color/black"
        android:textColorHint="@color/black"
        android:textSize="50sp"
        app:layout_constraintBottom_toTopOf="@+id/dbtn1"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Outputs:

Open the Android Studio

Creating a project with No Activity

Filling the necessary details

Wait till the Gradle is build

Creating a new Empty Views Activity

Selecting the main Activity as the Launcher Activity

Editing the xml file

Editing the xml file

Editing the xml file

Editing the xml file

Verifying the xml file

Basic Overview of the App

Successfully getting the alert

Retrieving the clock dialog

Retrieving the Calendar dialogue

App is now Successfully Working

Date: 01-02-2025

Experiment – 4

Aim:

Build mobile application using Recycler View

Description:

RecyclerView is an advanced and flexible **view group** in Android that efficiently displays large sets of data in a **scrollable list or grid** while reusing views to optimize performance.

Components of RecyclerView

- **RecyclerView**: The container for displaying a list of items.
- **ViewHolder**: Holds references to UI components for each item.
- **Adapter**: Binds data to the ViewHolder and manages list updates.
- **LayoutManager**: Defines how items are displayed (Linear, Grid, or Staggered).

Implementation Steps

- Add RecyclerView to the layout.
- Create a model class to represent data items.
- Implement a ViewHolder class to bind UI components.
- Create an Adapter to manage data and bind it to the ViewHolder.
- Set up RecyclerView in MainActivity with a LayoutManager and Adapter.

Programs:

MainActivity.java

```
package com.example.a22501a0557_recyclerview;

import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
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);

        // Creating the list of cricketers
        List<Cricketer> players = new ArrayList<>();
        players.add(new Cricketer(R.drawable.download, "Virat Kohli", "Batsman"));
```

```
        players.add(new Cricketer(R.drawable.dhoni, "MS Dhoni", "Wicket-Keeper"));
        players.add(new Cricketer(R.drawable.jadeja, "Ravindra Jadeja", "All-Rounder"));

        // Setting up RecyclerView
        RecyclerView rcv = findViewById(R.id.rcv);
        rcv.setAdapter(new Cricketer_adapter(this, players));
        rcv.setLayoutManager(new LinearLayoutManager(this));
    }
}
```

activity_main.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"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/rcv"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />
</FrameLayout>
```

recyclelayout.xml

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

    <ImageView
        android:id="@+id/imgv_cpic"
        android:layout_width="100dp"
        android:layout_height="100dp"
        tools:srcCompat="@tools:sample/avatars" />

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

```
        android:layout_height="wrap_content"
        android:layout_toRightOf="@id/imgv_cpic"
        android:text="Cricketer Name"
        android:textSize="18sp"
        android:textStyle="bold"/>

    <TextView
        android:id="@+id/txtv_crole"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/txtv_cname"
        android:layout_toRightOf="@id/imgv_cpic"
        android:text="Role" />
</RelativeLayout>
```

Cricketer.java

```
package com.example.a22501a0557_recyclerview;

public class Cricketer {
    private int cpic;
    private String cname, crole;

    public Cricketer(int cpic, String cname, String crole) {
        this.cpic = cpic;
        this.cname = cname;
        this.crole = crole;
    }

    public int getCpic() { return cpic; }
    public String getCname() { return cname; }
    public String getCrole() { return crole; }
}
```

Cricketer_NameHolder.java

```
package com.example.a22501a0557_recyclerview;

import android.view.View;
import android.widget.ImageView;
import android.widget.TextView;
import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

public class Cricketer_NameHolder extends RecyclerView.ViewHolder {
    ImageView imgv_cpic;
```

```
    TextView txtv_cname, txtv_crole;

    public Cricketer_NameHolder(@NonNull View itemView) {
        super(itemView);
        imgv_cpic = itemView.findViewById(R.id.imgv_cpic);
        txtv_cname = itemView.findViewById(R.id.txtv_cname);
        txtv_crole = itemView.findViewById(R.id.txtv_crole);
    }
}
```

Cricketer_adapter.java

```
package com.example.a22501a0557_recyclerview;
import android.content.Context;
import android.view.LayoutInflater;
import android.view.ViewGroup;
import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;
import java.util.List;

public class Cricketer_adapter extends RecyclerView.Adapter<Cricketer_NameHolder> {
    Context ct;
    List<Cricketer> players;

    public Cricketer_adapter(Context ct, List<Cricketer> players) {
        this.ct = ct;
        this.players = players;
    }
    @NonNull
    @Override
    public Cricketer_NameHolder onCreateViewHolder(@NonNull ViewGroup parent, int
viewType) {
        return new
Cricketer_NameHolder(LayoutInflater.from(ct).inflate(R.layout.recyclelayout, parent, false));
    }
    @Override
    public void onBindViewHolder(@NonNull Cricketer_NameHolder holder, int position) {
        holder.imgv_cpic.setImageResource(players.get(position).getCpic());
        holder.txtv_cname.setText(players.get(position).getCname());
        holder.txtv_crole.setText(players.get(position).getCrole());
    }

    @Override
    public int getItemCount() {
        return players.size();
```

```
    }
}
```

Outputs:

Creating a new project with No Activity

Filling the required details

Creating a new Empty Views Activity

Creating the main activity and selecting it as the launcher activity

Navigating to create a new xml file

Creating a new xml file

Editing the xml file

Editing the main java file

Creating a new java file

Editing the Cricketer java file

Editing the main java file

Navigating to create a java file

Creating an holder java file

Editing the holder java file

Navigating to create a new class

Creating a java class for Cricketer_adapter

Editing the Cricketer_adapter java file

Editing the Main java file

Successfully Got the output

Date: 08-02-2025

Experiment – 5

Aim:

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

Description:

An **Intent** in Android is a messaging object used to **navigate between activities, pass data, or trigger system actions** like opening a browser or calling a phone number.

Types of Intents

- **Explicit Intent**: Directly specifies the target activity to launch.
- **Implicit Intent**: Requests an action to be performed by any suitable application.

Implementation Steps

- Create two activities (MainActivity and SecondActivity).
- Add a button in MainActivity to trigger the intent.
- Use an **Explicit Intent** to start SecondActivity.
- Modify AndroidManifest.xml to declare both activities.

Programs:

MainActivity.java

```
package com.example.gvsr557_intents;

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;
import android.widget.Toast;
import android.net.Uri;

public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
    public void sendEmail(View v) {
        EditText et = findViewById(R.id.etxt_name);
        String recipientEmail = et.getText().toString().trim();

        if (recipientEmail.isEmpty()) {
            et.setError("Please enter an email");
            return;
        }

        Intent emailIntent = new Intent(Intent.ACTION_SEND);
        emailIntent.setType("message/rfc822");
        emailIntent.putExtra(Intent.EXTRA_EMAIL, new String[]{recipientEmail});
        emailIntent.putExtra(Intent.EXTRA_SUBJECT, "Your Subject");
        emailIntent.putExtra(Intent.EXTRA_TEXT, "Email message goes here");

        try {
            startActivity(Intent.createChooser(emailIntent, "Send mail..."));
            Log.i("Email", "Finished sending email.");
        } catch (android.content.ActivityNotFoundException ex) {
            Toast.makeText(MainActivity.this, "No email client installed.",
Toast.LENGTH_SHORT).show();
        }
    }
```

```
    public void LaunchActivity(View v) {
        EditText et = findViewById(R.id.etxt_name);
        String name = et.getText().toString().trim();

        if (name.isEmpty()) {
            et.setError("Please enter your name");
            return;
        }

        Intent i = new Intent(getApplicationContext(), SecondActivity.class);
        Bundle bundle = new Bundle();
        bundle.putString("user_name", name);
        i.putExtras(bundle);
        startActivity(i);
    }
    public void browseLink(View v) {
        EditText et = findViewById(R.id.etxt_name);
        String url = et.getText().toString().trim();

        if (url.isEmpty()) {
            et.setError("Please enter a URL");
            return;
        }

        if (!url.startsWith("http://") && !url.startsWith("https://")) {
            url = "http://" + url;
        }

        Intent browserIntent = new Intent(Intent.ACTION_VIEW, Uri.parse(url));
        startActivity(browserIntent);
    }

}
```

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"
    android:gravity="center"
    android:padding="24dp"
```

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

<TextView
    android:id="@+id/intentsTitle"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:paddingBottom="30dp"
    android:text="Intents App"
    android:textColor="@android:color/black"
    android:textSize="36sp"
    android:textStyle="bold" />

<EditText
    android:id="@+id/etxt_name"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Enter your name"
    android:padding="14dp"
    android:background="@android:drawable/editbox_background"
    android:minHeight="56dp"
    android:inputType="textPersonName" />

<Button
    android:id="@+id/secondActivityButton"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:onClick="LaunchActivity"
    android:text="Show Second Activity"
    android:textSize="16sp"
    android:padding="14dp"
    android:minHeight="56dp"/>

<Button
    android:id="@+id/sendEmail"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:onClick="sendEmail"
    android:text="Send Email"
    android:textSize="16sp"
    android:padding="14dp"
    android:minHeight="56dp"/>
```

```
    <Button
        android:id="@+id/browseLinkButton"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:onClick="browseLink"
        android:text="Browse Link"
        android:padding="12dp"
        android:minHeight="48dp"/>
</LinearLayout>
```

SecondActivity.java

```
package com.example.gvsr557_intents;

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

public class SecondActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_second);

        TextView textView = findViewById(R.id.txtv_display);

        Bundle bundle = getIntent().getExtras();
        if (bundle != null) {
            String name = bundle.getString("user_name", "User");
            textView.setText("Welcome, " + name);
        }
    }
}
```

activity_second.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/sdtxtv"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".SecondActivity">
```

```
    <Button
        android:id="@+id/showMainActivity"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="show_main"
        android:text="showMainActivity"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/txtv_display"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="300dp"
        android:text="TextView"
        android:textSize="30sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

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">

    <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.Gvsr557_intents"
        tools:targetApi="31">

        <activity
            android:name=".SecondActivity"
            android:exported="false" />
```

```
        <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>
```

Outputs:

Opening the Android Studio

Creating a project with No Activity

Filling the Required fields

Budling the Gradle

Creating a new Empty View Activity

Selecting the Launcher Activity for main one

Editing the main java file

Creating another Empty View Activity

Filling the required data and should not select the Launcher activity(As main is already selected)

Editing the activities

This is the second activity we created

Editing the Second Activity

Editing the second xml file

Editing the second xml file

Editing the second xml file

Editing the second xml file

Editing the second xml file

Editing the second xml file

Editing the second xml file

Editing the second xml file

Editing the second xml file

Editing the second xml file

Runing the app

The app installed successfully

We navigated to the second activity

Trying to send mail

Successfully working mail

Trying to access the browser

Successfully retrieved my GitHub website

Date: 15-02-2025

Experiment – 6

Aim:

Build mobile application to demonstrate Dynamic Fragments

Description:

A **Dynamic Fragment** is a UI component that can be added, replaced, or removed at **runtime** using Java/Kotlin code. It provides flexibility for building responsive and modular user interfaces.

Types of Fragment Usage

- **Static Fragment**: Declared directly in the XML layout.
- **Dynamic Fragment**: Added or replaced programmatically during runtime.

Implementation Steps

- Create a layout with a **FrameLayout** as the fragment container.
- Create two fragment classes (e.g., FirstFragment, SecondFragment).
- Add buttons in the activity to trigger fragment changes.
- Use **FragmentManager** and **FragmentTransaction** to:
  – add() or replace() fragments dynamically.
- Handle fragment transactions inside **button click listeners**.

Programs:

MainActivity.java

```
package com.example.a22501a0557_fragments;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.fragment.app.FragmentManager;
import androidx.fragment.app.FragmentResultListener;
import androidx.fragment.app.FragmentTransaction;

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

public class MainActivity extends AppCompatActivity {
    SportsFragment sf;
    FragmentManager fm;
    FragmentTransaction ft;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        sf=new SportsFragment();
        fm=getSupportFragmentManager();
        ft=fm.beginTransaction();
        ft.add(R.id.topll,sf);
        ft.commit();
        fm.setFragmentResultListener("si", this, new FragmentResultListener() {
            @Override
            public void onFragmentResult(@NonNull String requestKey, @NonNull Bundle
result) {
                int options=result.getInt("si");
                ft= fm.beginTransaction();
                switch (options){
                    case 0:
                        ft.replace(R.id.bottomll,new BadmintonFragment());
                        break;
                    case 1:
                        ft.replace(R.id.bottomll,new CricketFragment());
                        break;
                    case 2:
                        ft.replace(R.id.bottomll,new FootballFragment());
                        break;
                    case 3:
```

```
                ft.replace(R.id.bottomll,new HockeyFragment());
                break;
            default:
                ft.replace(R.id.bottomll,new CricketFragment());
        }
        ft.commit();
    }
});

}
}
```

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">

    <LinearLayout
        android:id="@+id/topll"
        android:layout_width="385dp"
        android:layout_height="298dp"
        android:orientation="horizontal"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.076"></LinearLayout>

    <LinearLayout
        android:id="@+id/bottomll"
        android:layout_width="382dp"
        android:layout_height="333dp"
        android:orientation="horizontal"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/topll"></LinearLayout>
</androidx.constraintlayout.widget.ConstraintLayout>
```

BadmintonFragment.java

```
package com.example.a22501a0557_fragments;

import android.os.Bundle;

import androidx.fragment.app.Fragment;

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

public class BadmintonFragment extends Fragment {
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_badminton, container, false);
    }
}
```

fragment_badminton.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"
    tools:context=".BadmintonFragment">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/badminton" />
</FrameLayout>
```

CricketFragment.java

```
package com.example.a22501a0557_fragments;

import android.os.Bundle;

import androidx.fragment.app.Fragment;

import android.view.LayoutInflater;
import android.view.View;
```

```
import android.view.ViewGroup;

public class CricketFragment extends Fragment {
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_cricket, container, false);
    }
}
```

fragment_cricket.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"
    tools:context=".CricketFragment">

    <ImageView
        android:id="@+id/imageView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/cricket" />
</FrameLayout>
```

FootballFragment.java

```
package com.example.a22501a0557_fragments;

import android.os.Bundle;

import androidx.fragment.app.Fragment;

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
public class FootballFragment extends Fragment {
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_football, container, false);
```

```
    }
}
```

fragment_football.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"
    tools:context=".FootballFragment">

    <ImageView
        android:id="@+id/imageView3"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:src="@drawable/football" />
</FrameLayout>
```

HockeyFragment.java

```
package com.example.a22501a0557_fragments;

import android.os.Bundle;

import androidx.fragment.app.Fragment;

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
public class HockeyFragment extends Fragment {
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_hockey, container, false);
    }
}
```

fragment_hockey.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"
    tools:context=".HockeyFragment">

    <ImageView
        android:id="@+id/imageView4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/hockey" />
</FrameLayout>
```

SportsFragment.java

```
package com.example.a22501a0557_fragments;

import android.os.Bundle;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.ListView;

public class SportsFragment extends Fragment implements
AdapterView.OnItemClickListener
{  ListView lv;


    public SportsFragment()
    {

    }

    @Override
    public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState)
{
        super.onViewCreated(view, savedInstanceState);
        lv=(ListView)  view.findViewById(R.id.lv_sports);
        lv.setOnItemClickListener(this);
    }

    @Override
```

```
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        return inflater.inflate(R.layout.fragment_sports, container, false);
    }

    @Override
    public void onItemClick(AdapterView<?> parent, View view, int i, long l) {
        Bundle b= new Bundle();
        b.putInt("si",i);
        getParentFragmentManager().setFragmentResult("si",b);
    }
}
```

fragment_sports.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=".SportsFragment">

    <ListView
        android:id="@+id/lv_sports"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:entries="@array/sports_list" />
</FrameLayout>
```

Outputs:

Opening the Android Studio

Selecting the No Activity

Filling the Required fields

Wait till the Gradle building is successfully done

Create an Empty View Activity

Creating the Main Activity with Launcher Activity Enabled

We get java and xml files for MainActivity

Adding the necessary things in xml

Editing the necessary things in xml

Editing the necessary things in xml

Creating a new Fragment

Naming the Fragment

Editing the fragment

Editing the xml file

Editing strings.xml if required

Added Strings to strings.xml

Editing the xml file

Verifying the toasts of the app

Verifying the toast again

Creating the other fragments

Creating the Badminton Fragment

Editing the java file

Editing the xml file

Editing the MainActivity

Fixing the error by the suggestions

Implementing the methods

Displaying the Badminton Fragment

Displaying the Cricket Fragment

Displaying the Football Fragment

Displaying the Hockey Fragment

Date: 01-03-2025

Experiment – 7.1

Aim:

Build mobile application serverless database SQLite Database, Firebase (cloud-hosted database)

Description:

SQLite Database in Android

SQLite is a lightweight, serverless, self-contained relational database engine built into Android. It allows local data storage directly on the device without needing a server, making it ideal for offline apps.

Key Features

- No server setup required (serverless)
- Stores structured data in tables using SQL
- Fast and efficient for mobile devices
- Good for caching, user data, and small applications

Implementation Steps

- Create a class that extends SQLiteOpenHelper
- Override onCreate() to define database schema
- Use SQLiteDatabase methods to insert(), update(), delete(), and query()
- Create a UI with input fields and buttons for interaction
- Display data using ListView or RecyclerView if needed

Programs:

MainActivity.java

```
package com.example.a22501a0557_sqlite;

import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {
    // Database and com.example.a22501a0557_databases.Student object
    StudentGradeDB sgdb;
    Student s;

    // UI Components
```

```
EditText edttxtv_roll, edttxtv_sname, edttxtv_avg, edttxtv_grade;
int roll;
String sname, grade;
float avg;

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

    // Initialize Database
    sgdb = new StudentGradeDB(getApplicationContext());

    // Link UI elements
    edttxtv_roll = findViewById(R.id.editTxt_roll);
    edttxtv_sname = findViewById(R.id.editTxt_name);
    edttxtv_avg = findViewById(R.id.editTxt_avg);
    edttxtv_grade = findViewById(R.id.editTxt_grade);
}

// Insert com.example.a22501a0557_databases.Student Data
public void insertStudent(View v) {
    roll = Integer.parseInt(edttxtv_roll.getText().toString());
    sname = edttxtv_sname.getText().toString();
    avg = Float.parseFloat(edttxtv_avg.getText().toString());
    grade = edttxtv_grade.getText().toString();

    s = new Student(roll, sname, avg, grade);
    sgdb.addStudent(s);

    Toast.makeText(this, "Insertion Successful", Toast.LENGTH_SHORT).show();
}

// Delete com.example.a22501a0557_databases.Student Data
public void delete_student(View v) {
    try {
        roll = Integer.parseInt(edttxtv_roll.getText().toString());
        sgdb.deleteStudent(roll);
        Log.d("GVSR557 good", "delete_student: Successful");
    } catch (Exception ex) {
        Log.d("PROBLEM GVSR557:", "delete_student: " + ex.getMessage());
    }
}
```

```
    // Retrieve com.example.a22501a0557_databases.Student Data
    public void get_student(View v) {
        try {
            roll = Integer.parseInt(edttxtv_roll.getText().toString());
            String c = sgdb.getStudent(roll);
            Toast.makeText(this, c, Toast.LENGTH_SHORT).show();
            Log.d("GVSR557 good", "select_student: Successful" + c);
        } catch (Exception ex) {
            Log.d("PROBLEM GVSR557:", "select_student: " + ex.getMessage());
        }
    }

    // Update com.example.a22501a0557_databases.Student Data
    public void updateStudent(View v) {
        try {
            roll = Integer.parseInt(edttxtv_roll.getText().toString());
            sname = edttxtv_sname.getText().toString();
            avg = Float.parseFloat(edttxtv_avg.getText().toString());
            grade = edttxtv_grade.getText().toString();

            s = new Student(roll, sname, avg, grade);
            sgdb.updateStudent(s);

            Toast.makeText(this, "Updation Successful", Toast.LENGTH_SHORT).show();
            Log.d("GVSR557 good", "update_student: Successful");
        } catch (Exception ex) {
            Log.d("GVSR557 problem", "update_student: " + ex.getMessage());
        }
    }
}
```

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">

    <TableLayout
        android:layout_width="344dp"
        android:layout_height="646dp"
```

```
    android:layout_margin="20dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent">

    <!-- Heading (Centered) -->
    <TableRow>
        <TextView
            android:id="@+id/TxtV_Heading"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_span="2"
            android:gravity="center"
            android:padding="10dp"
            android:text="@string/student_grade_details"
            android:textAlignment="center"
            android:textColor="@android:color/holo_red_dark"
            android:textSize="25sp"
            android:textStyle="bold" />
    </TableRow>

    <!-- Roll Number -->
    <TableRow>
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:padding="5dp"
            android:text="@string/roll" />

        <EditText
            android:id="@+id/editTxt_roll"
            android:layout_width="230dp"
            android:layout_height="wrap_content"
            android:autofillHints="username"
            android:hint="@string/enter_roll"
            android:inputType="number"
            android:minHeight="48dp" />
    </TableRow>

    <!-- Name -->
    <TableRow>
        <TextView
            android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
        android:padding="5dp"
        android:text="@string/name" />

    <EditText
        android:id="@+id/editTxt_name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:autofillHints="name"
        android:hint="@string/enter_name"
        android:inputType="textPersonName"
        android:minHeight="48dp" />
</TableRow>

<!-- Average -->
<TableRow>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="5dp"
        android:text="@string/average" />

    <EditText
        android:id="@+id/editTxt_avg"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:autofillHints="text"
        android:hint="@string/enter_average"
        android:inputType="numberDecimal"
        android:minHeight="48dp" />
</TableRow>

<!-- Grade -->
<TableRow>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="5dp"
        android:text="@string/grade" />

    <EditText
        android:id="@+id/editTxt_grade"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
```

```
        android:autofillHints="text"
        android:hint="@string/enter_grade"
        android:inputType="text"
        android:minHeight="48dp" />
    </TableRow>

    <!-- Buttons spanning two columns -->
    <TableRow>
      <Button
        android:id="@+id/btn_insert"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_span="2"
        android:backgroundTint="@color/design_default_color_primary"
        android:minHeight="48dp"
        android:onClick="insertStudent"
        android:text="@string/insert"
        android:textColor="@android:color/white" />
    </TableRow>

    <TableRow>
      <Button
        android:id="@+id/btn_delete"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_span="2"
        android:backgroundTint="@color/design_default_color_primary"
        android:minHeight="48dp"
        android:onClick="delete_student"
        android:text="@string/delete"
        android:textColor="@android:color/white" />
    </TableRow>

    <TableRow>
      <Button
        android:id="@+id/btn_select"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_span="2"
        android:backgroundTint="?attr/colorPrimary"
        android:minHeight="48dp"
        android:onClick="get_student"
        android:text="@string/select"
        android:textColor="@android:color/white" />
```

```
        </TableRow>

        <TableRow>
            <Button
                android:id="@+id/btn_update"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_span="2"
                android:backgroundTint="?attr/colorPrimary"
                android:minHeight="48dp"
                android:onClick="updateStudent"
                android:text="@string/update"
                android:textColor="@android:color/white" />
        </TableRow>

    </TableLayout>

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

StudentGradeDB.java

```
package com.example.a22501a0557_sqlite;

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

public class StudentGradeDB extends SQLiteOpenHelper {

    // Database details
    private static final int DATABASE_VERSION = 1;
    private static final String DATABASE_NAME = "students";
    private static final String TABLE_NAME = "StudentGrade";

    // Table columns
    private static final String KEY_ID = "Roll";
    private static final String KEY_NAME = "sname";
    private static final String KEY_AVG = "average";
    private static final String KEY_GRADE = "grade";

    public StudentGradeDB(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }
```

```
    @Override
    public void onCreate(SQLiteDatabase db) {
        String CREATE_TABLE = "CREATE TABLE " + TABLE_NAME + " ("
                + KEY_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, "
                + KEY_NAME + " TEXT NOT NULL, "
                + KEY_AVG + " FLOAT, "
                + KEY_GRADE + " TEXT );";
        db.execSQL(CREATE_TABLE);
    }

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

    // Insert Student Data
    public boolean addStudent(Student student) {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues values = new ContentValues();
        values.put(KEY_NAME, student.getSname());
        values.put(KEY_ID, student.getRoll());
        values.put(KEY_AVG, student.getAverage());
        values.put(KEY_GRADE, student.getGrade());

        long result = db.insert(TABLE_NAME, null, values);
        db.close();
        return result != -1;
    }

    // Retrieve Student Data
    public String getStudent(int roll) {
        SQLiteDatabase db = this.getReadableDatabase();
        Cursor c = db.query(TABLE_NAME, new String[]{KEY_ID, KEY_NAME, KEY_AVG,
KEY_GRADE},
                "Roll=?", new String[]{String.valueOf(roll)}, null, null, null);

        if (c.moveToFirst()) {
            int r = c.getInt(0);
            String sname = c.getString(1);
            float avg = c.getFloat(2);
            String grade = c.getString(3);
            c.close();
```

```
            return "Roll=" + r + "\nName=" + sname + "\nAverage=" + avg + "\nGrade=" +
grade;
        } else {
            c.close();
            return "No record available";
        }
    }

    // Delete Student Data
    public boolean deleteStudent(int roll) {
        SQLiteDatabase db = this.getWritableDatabase();
        boolean deleted = db.delete(TABLE_NAME, "Roll=?", new
String[]{String.valueOf(roll)}) > 0;
        db.close();
        return deleted;
    }

    // Update Student Data
    public boolean updateStudent(Student student) {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues values = new ContentValues();
        values.put(KEY_NAME, student.getSname());
        values.put(KEY_AVG, student.getAverage());
        values.put(KEY_GRADE, student.getGrade());

        boolean updated = db.update(TABLE_NAME, values, "Roll=?", new
String[]{String.valueOf(student.getRoll())}) > 0;
        db.close();
        return updated;
    }
}
```

Student.java

```
package com.example.a22501a0557_sqlite;

public class Student {
    private int roll;
    private String sname;
    private float average;
    private String grade;

    // Constructor
    public Student(int roll, String sname, float average, String grade) {
        this.sname = sname;
```

```
        this.roll = roll;
        this.average = average;
        this.grade = grade;
    }

    // Getter Methods
    public int getRoll() { return roll; }
    public String getSname() { return sname; }
    public float getAverage() { return average; }
    public String getGrade() { return grade; }

    // Setter Methods
    public void setRoll(int roll) { this.roll = roll; }
    public void setSname(String sname) { this.sname = sname; }
    public void setGrade(String grade) { this.grade = grade; }
    public void setAverage(float average) { this.average = average; }
}
```

strings.xml

```
<resources>
    <string name="app_name">22501A0557_SQLite</string>
    <string name="student_grade_details">Student Grade Details</string>
    <string name="roll">Roll</string>
    <string name="enter_roll">Enter Roll</string>
    <string name="name">Name of student</string>
    <string name="enter_name">Enter Name</string>
    <string name="average">Average</string>
    <string name="enter_average">Enter Average</string>
    <string name="grade">Grade</string>
    <string name="enter_grade">Enter Grade</string>
    <string name="insert">INSERT</string>
    <string name="delete">DELETE</string>
    <string name="select">SELECT</string>
    <string name="update">UPDATE</string>
</resources>
```

Outputs:

Creating a new project

Creating a project with No Activity

Filling the details about new Project

New Project is successfully Created

Creating new Empty View Activity (MainActivity)

Filling the Details and Selecting Launcher Activity

MainActivity is successfully Created

Creating a table layout

Changing Strings.xml relating to our data

Adding Necessary Buttons to the layout

Creating a new Java class

Creating a student class

Creating the Database Java class

Overall File Structure

Successful App Execution

Data Successfully Inserted into Database

Updating the Data

Selecting the data from the Database

After Deleting the Data from the Database

Date: 08-03-2025

Experiment – 7.2

Aim:

Build mobile application serverless database Firebase (cloud-hosted database)

Description:

Firebase Realtime Database in Android

Firebase Realtime Database is a cloud-hosted, serverless NoSQL database that lets you store and sync data in real time across all connected clients. It is ideal for apps requiring live updates, such as chat apps, user profiles, or collaborative tools.

Key Features

- Cloud-hosted and serverless
- Stores data in JSON tree structure
- Real-time sync across devices
- Automatically handles offline caching and syncing
- Easy integration with Firebase Authentication and other services

Implementation Steps

- Create a Firebase project at firebase.google.com
- Connect your Android app to Firebase using Android Studio
- Add the Firebase Realtime Database dependency in build.gradle
- Enable Realtime Database and set security rules
- Use FirebaseDatabase.getInstance().getReference() to read and write data
- Structure your data in key-value pairs or nested objects

Program:

MainActivity.java

```
package com.example.a22501a0557_firebase0;


import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;

import androidx.activity.EdgeToEdge;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;
```

```
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;


public class MainActivity extends AppCompatActivity {

    EditText edtxt_roll, edtxt_name, edtxt_avg, edtxt_grade;
    FirebaseDatabase fdb;

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

        edtxt_roll = findViewById(R.id.edtxt_roll);
        edtxt_name = findViewById(R.id.edtxt_name);
        edtxt_avg = findViewById(R.id.edtxt_avg);
        edtxt_grade = findViewById(R.id.edtxt_grade);

        fdb = FirebaseDatabase.getInstance();
    }

    public void insertStudent(View v) {
        Student s = new Student(
                edtxt_roll.getText().toString(),
                edtxt_name.getText().toString(),
                edtxt_avg.getText().toString(),
                edtxt_grade.getText().toString()
        );

        fdb.getReference("students").push().setValue(s)
                .addOnSuccessListener(unused ->
                        Toast.makeText(MainActivity.this, "Insertion Success",
Toast.LENGTH_SHORT).show()
                )
                .addOnFailureListener(e ->
                        Toast.makeText(MainActivity.this, "Insertion Failed",
Toast.LENGTH_SHORT).show()
                );
    }
```

```
    public void viewStudent(View v) {
        String inputRoll = edtxt_roll.getText().toString().trim();

        if (inputRoll.isEmpty()) {
            Toast.makeText(MainActivity.this, "Please enter a roll number",
Toast.LENGTH_SHORT).show();
            return;
        }

        fdb.getReference("students").addListenerForSingleValueEvent(new
ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                boolean found = false;

                for (DataSnapshot dss : snapshot.getChildren()) {
                    Student st = dss.getValue(Student.class);
                    if (st != null) {
                        Log.d("FIREBASE", "Checking roll: " + st.getRoll());
                        if (inputRoll.equals(st.getRoll().trim())) {
                            edtxt_name.setText(st.getName());
                            edtxt_avg.setText(st.getAvg());
                            edtxt_grade.setText(st.getGrade());
                            found = true;
                            break;
                        }
                    }
                }

                if (!found) {
                    Toast.makeText(MainActivity.this, "Student not found",
Toast.LENGTH_SHORT).show();
                }
            }

            @Override
            public void onCancelled(@NonNull DatabaseError error) {
                Toast.makeText(MainActivity.this, "Error fetching 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"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:orientation="vertical"
    android:padding="16dp"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <EditText
        android:id="@+id/edtxt_roll"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter Roll Number"
        android:textSize="20sp"
        android:inputType="number" />

    <EditText
        android:id="@+id/edtxt_name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter Name"
        android:textSize="20sp" />

    <EditText
        android:id="@+id/edtxt_avg"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter Average Marks"
        android:textSize="20sp"
        android:inputType="numberDecimal" />

    <EditText
        android:id="@+id/edtxt_grade"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter Grade"
        android:textSize="20sp" />
```

```
    <Button
        android:id="@+id/btn_insert"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:onClick="insertStudent"
        android:text="Insert Student"
        android:textSize="20sp" />

    <Button
        android:id="@+id/btn_view"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:onClick="viewStudent"
        android:text="View Student"
        android:textSize="20sp" />
</LinearLayout>
```

Student.java

```
package com.example.a22501a0557_firebase0;

public class Student {
    private String roll,name,avg,grade;

    public Student() {
    }

    public Student(String roll, String name, String avg, String grade) {
        this.roll = roll;
        this.name = name;
        this.avg = avg;
        this.grade = grade;
    }

    public String getRoll() {
        return roll;
    }

    public void setRoll(String roll) {
        this.roll = roll;
    }

    public String getName() {
```

```
        return name;
    }

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

    public String getAvg() {
        return avg;
    }

    public void setAvg(String avg) {
        this.avg = avg;
    }

    public String getGrade() {
        return grade;
    }

    public void setGrade(String grade) {
        this.grade = grade;
    }
}
```

Output:

Creating a new project

Selecting No Activity

Filling Required fields

Creating a new Activity

Creating MainActivity

Selecting the Firebase

Selecting RealTime Database

Getting Started with RealTime Database

Creating a new Project

Adding Details

After Continuing Everything, We'll Create a Project

Our FireBase Project is ready

Connecting The Firebase and Android Studio Project

Successful Connection Established

Verifying in Android Studio

Adding The Realtime Database SDK to our App

Successfully Added Realtime database SDK

Go to Firebase Console

Creating The Database

Selecting The Nearest Datacentre

Enabling Test Mode since we are not working on Confidential Project

Realtime Database Is Created Successfully

Writing The code to Required fields

Creating a Student Class

Generating Constructor

Inserting a student into the database

Trying to retrieve the student

Received the students' details when pressed view Student by giving only Roll Number

Details of the Student in the database

Date: 22-03-2025

Experiment – 8

Aim:

Build mobile application based on the Google Maps

Description:

We can build a mobile application based on **Google Maps** using **Android Studio** and **Java** by integrating the **Google Maps SDK for Android**. This allows you to display maps, add markers, track user location, and much more. Android Studio provides a built-in template for Google Maps, making it easy to get started.

Key Steps:

- Create a new project in **Android Studio** using the **Google Maps Activity** template.
- Get a **Google Maps API Key** from the Google Cloud Console.
- Enable **Maps SDK for Android** in your Google Cloud project.
- Add the API key to your app's google_maps_api.xml file.
- Update your AndroidManifest.xml with required permissions.
- Customize your map using the GoogleMap object in Java.

Programs:

local.properties

```
## This file is automatically generated by Android Studio.
# Do not modify this file -- YOUR CHANGES WILL BE ERASED!
#
# This file should *NOT* be checked into Version Control Systems,
# as it contains information specific to your local configuration.
#
# Location of the SDK. This is only used by Gradle.
# For customization when using a Version Control System, please read the
# header note.
sdk.dir=C\:\\Users\\G VENKATA SAI RAM\\AppData\\Local\\Android\\Sdk
MAPS_API_KEY=AIzaSyCqlAFb6pSaIB1GSDGrHjg0AG64EIxJSTY
```

Build.gradle.kts (:app)

```
plugins {
   alias(libs.plugins.android.application)
   alias(libs.plugins.google.android.libraries.mapsplatform.secrets.gradle.plugin)
}

android {
   namespace = "com.example.a22501a0557_google_maps"
   compileSdk = 35
```

```
    defaultConfig {
        applicationId = "com.example.a22501a0557_google_maps"
        minSdk = 24
        targetSdk = 35
        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
    }
    buildFeatures {
        viewBinding = true
    }
}

dependencies {

    implementation("com.google.android.gms:play-services-maps:19.1.0")
    implementation(libs.appcompat)
    implementation(libs.material)
    implementation(libs.play.services.maps)
    implementation(libs.constraintlayout)
    testImplementation(libs.junit)
    androidTestImplementation(libs.ext.junit)
    androidTestImplementation(libs.espresso.core)
}
```

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" />
    <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._22501A0557_Google_Maps"
        tools:targetApi="31" >

        <!--
            TODO: Before you run your application, you need a Google Maps API key.

            To get one, follow the directions here:

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

            Once you have your API key (it starts with "AIza"), define a new property in your
            project's local.properties file (e.g. MAPS_API_KEY=Aiza...), and replace the
            "YOUR_API_KEY" string in this file with "${MAPS_API_KEY}".
        -->
        <meta-data
            android:name="com.google.android.geo.API_KEY"
            android:value="${MAPS_API_KEY}" />

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

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

</manifest>
```

MapsActivity.java

```
package com.example.a22501a0557_google_maps;

import androidx.fragment.app.FragmentActivity;
import android.os.Bundle;

import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;
import com.example.a22501a0557_google_maps.databinding.ActivityMapsBinding;

public class MapsActivity extends FragmentActivity implements OnMapReadyCallback {

    private GoogleMap mMap;
private ActivityMapsBinding binding;

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

     binding = ActivityMapsBinding.inflate(getLayoutInflater());
     setContentView(binding.getRoot());

        // Obtain the SupportMapFragment and get notified when the map is ready to be used.
        SupportMapFragment mapFragment = (SupportMapFragment)
getSupportFragmentManager()
                .findFragmentById(R.id.map);
        mapFragment.getMapAsync(this);
    }

    /**
     * Manipulates the map once available.
     * This callback is triggered when the map is ready to be used.
     * This is where we can add markers or lines, add listeners or move the camera. In this
case,
     * we just add a marker near Sydney, Australia.
     * If Google Play services is not installed on the device, the user will be prompted to install
     * it inside the SupportMapFragment. This method will only be triggered once the user has
     * installed Google Play services and returned to the app.
     */
    @Override
```

```
    public void onMapReady(GoogleMap googleMap) {
        mMap = googleMap;
        mMap.getUiSettings().setZoomControlsEnabled(true);
        // Add a marker in Sydney and move the camera
        LatLng pvpsit = new LatLng(16.4877 , 80.6941);
        mMap.addMarker(new MarkerOptions().position(pvpsit).title("Marker in PVPSIT"));
        mMap.moveCamera(CameraUpdateFactory.newLatLng(pvpsit));
        MarkerOptions mo=new MarkerOptions();
        LatLng ion= new LatLng(16.487505338049363, 80.69832749516166);
        mo.position(ion);
        mo.title("Test Centre");
        mMap.addMarker(mo);
    }
}
```

activity_maps.xml

```
<?xml version="1.0" encoding="utf-8"?>
<fragment xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:map="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/map"
    tools:context=".MapsActivity"
    android:name="com.google.android.gms.maps.SupportMapFragment"/>
```

Outputs:

Creating new Project in Google Cloud for maps

Successfully Created the new Project

Viewing the Navigation menu Provided by google

Selecting Google maps Platform

Viewing the Google maps

Creating new credentials

Selecting the recent Project

Creating the credentials

We successfully Generated the Api key

Creating a new Android Project

Selecting No Activity

Filling the Required Fields

Creating the Google Maps Views Activity

Creating the MapsActivity which is Launcher

On MapReady is a method to retrieve the cloud into our map

Since we did not code much, there is the blank screen

Opening Maps SDK Website

Adding the google maps Dependency

After Adding Api Key, we successfully get output

Successfully Located PVPSIT in maps

Successfully Added Two markers in our Google Maps