Experiment: 1 Date: 20-12-24

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

AIM: To build mobile application based on the concept activity life cycle with Custom Toast.

EXPERIMENTAL REQUIREMENTS: PC with Android Studio and Internet connection

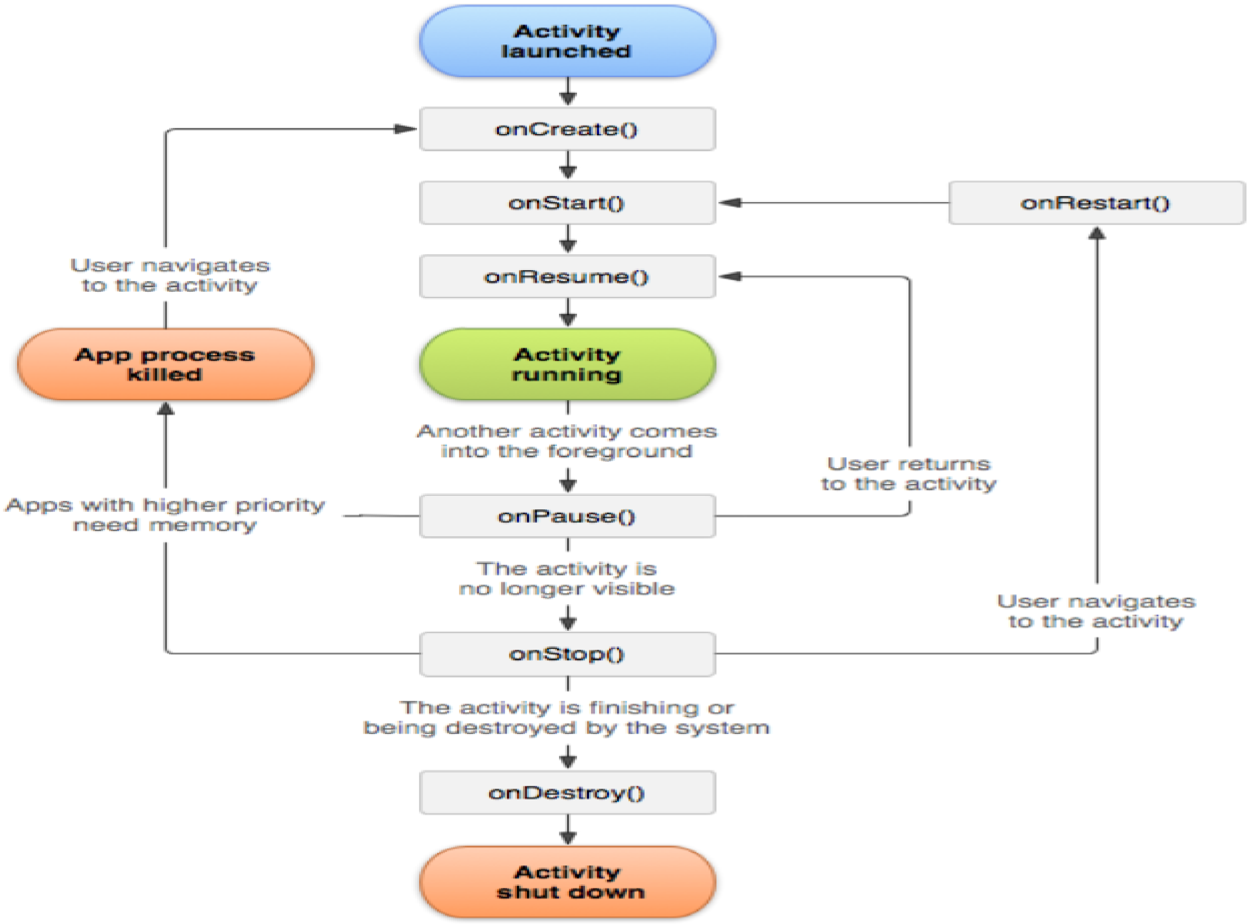
THEORY:

Android applications consist of activities, which serve as the primary components of user interaction. Each activity goes through different states in its lifecycle, managed by the Android system. The **Activity Lifecycle** defines the transitions between these states, ensuring efficient resource management.

An Android activity follows a **lifecycle** from the moment it is created to when it is destroyed. The Android system calls specific methods at different points in the activity's lifecycle, allowing developers to manage resources efficiently.

Key Lifecycle Methods:

1. **onCreate()** – Called when the activity is first created. Used for initializing UI components and logic.
2. **onStart()** – Called when the activity becomes visible but is not yet interactive.
3. **onResume()** – Called when the activity is in the foreground and ready for user interaction.
4. **onPause()** – Called when the activity is partially obscured (e.g., a dialog appears).
5. **onStop()** – Called when the activity is no longer visible. Used for releasing resources.
6. **onRestart()** – Called when an activity is restarted after being stopped.
7. **onDestroy()** – Called before the activity is completely removed from memory.


Activity launched
onCreate()
onStart()
onRestart()
User navigates to the activity
onResume()
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

Toast in Android

A **Toast** is a small, temporary message that appears at the bottom of the screen and disappears automatically after a short duration. It is used to display brief notifications without interrupting the user experience.

Types of Toasts in Android:

1. **Default Toast:**
   - Simple message display using Toast.makeText(context, "Message", Toast.LENGTH_SHORT).show();
2. **Custom Toast:**
   - Uses a custom layout, allowing for images, colors, and text styling.

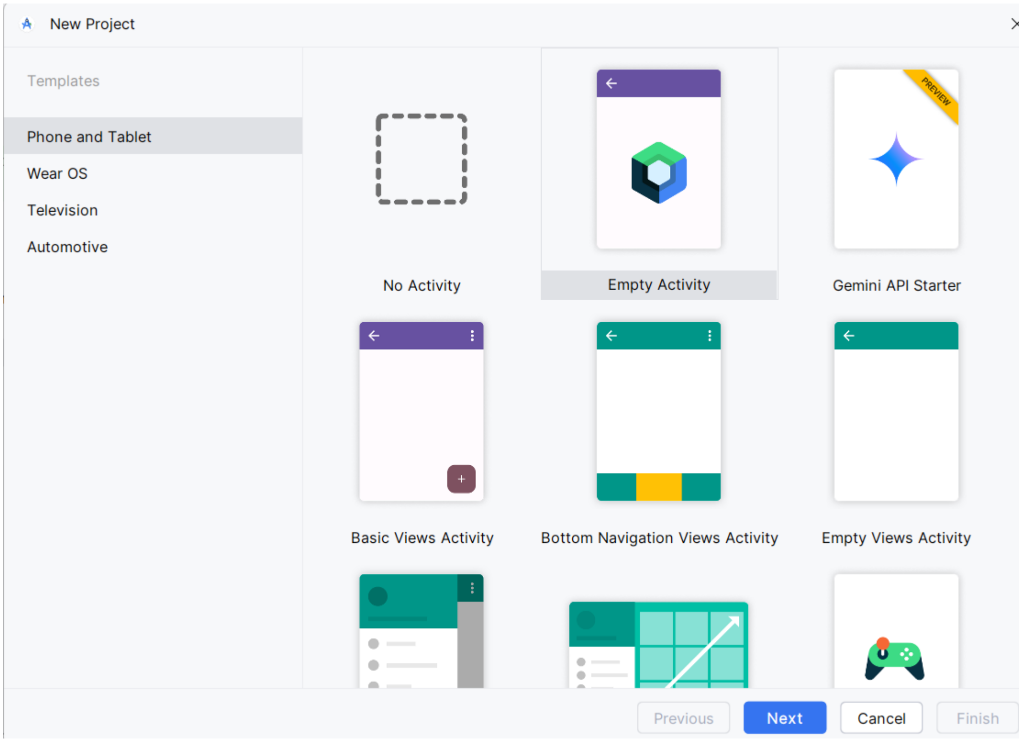
A **Custom Toast** provides a more visually appealing notification compared to a default Toast. It allows developers to use XML layouts to define the appearance of the toast message.

Components of a Custom Toast:

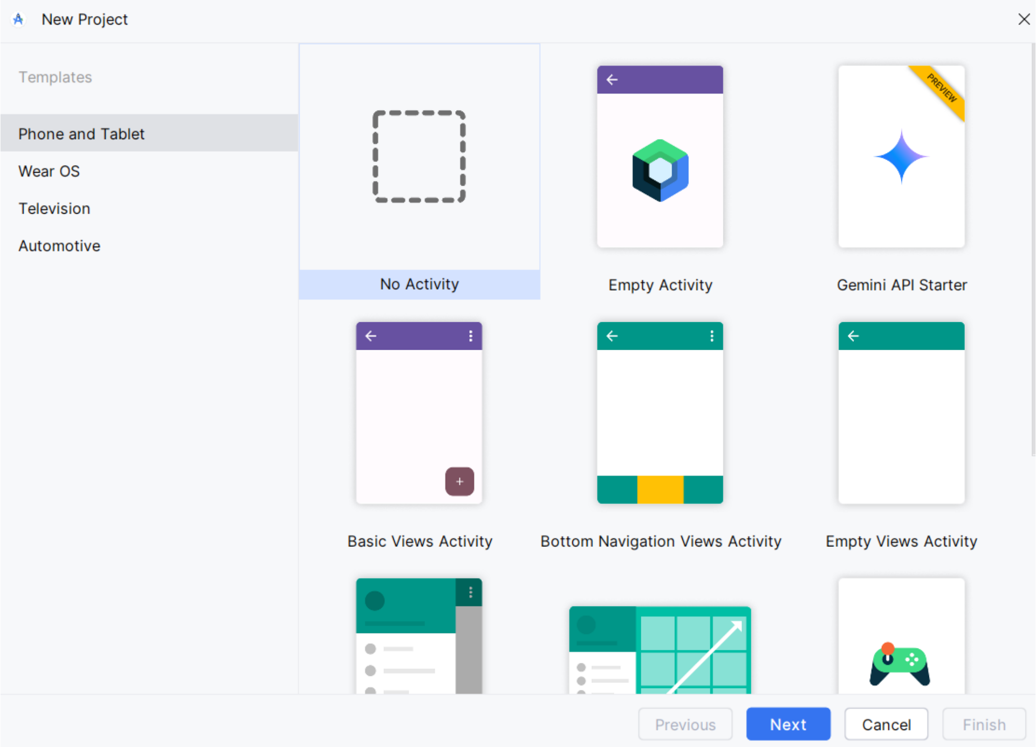
1. **Custom Layout (XML):**
   - A separate XML file is created with TextView, ImageView, or other UI elements.
2. **Inflation in Java Code:**
   - The layout is inflated dynamically using LayoutInflater.
3. **Displaying the Custom Toast:**
   - The custom view is set using Toast.setView() and shown using Toast.show().

PROCEDURE:

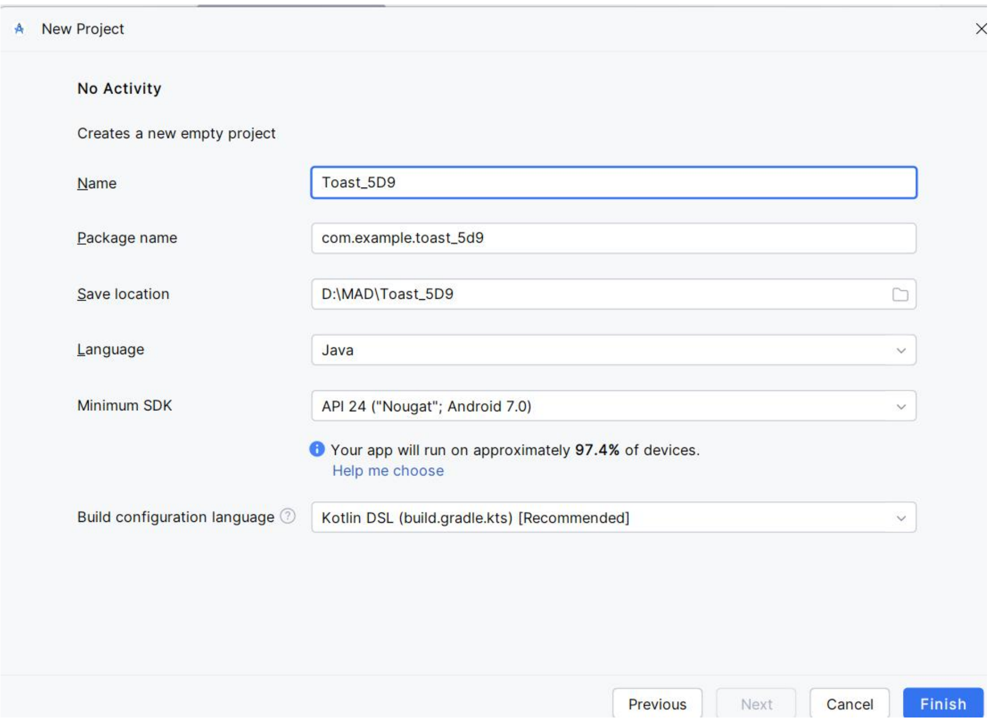
1. Open Android Studio and start a new project.



2. Select "No Views Activity" and click Next.



3. Enter project details (Name: ToastApp, Language: Java) and click Finish.



No Activity

Creates a new empty project

Name: Toast_5D9

Package name: com.example.toast_5d9

Save location: D:\MAD\Toast_5D9

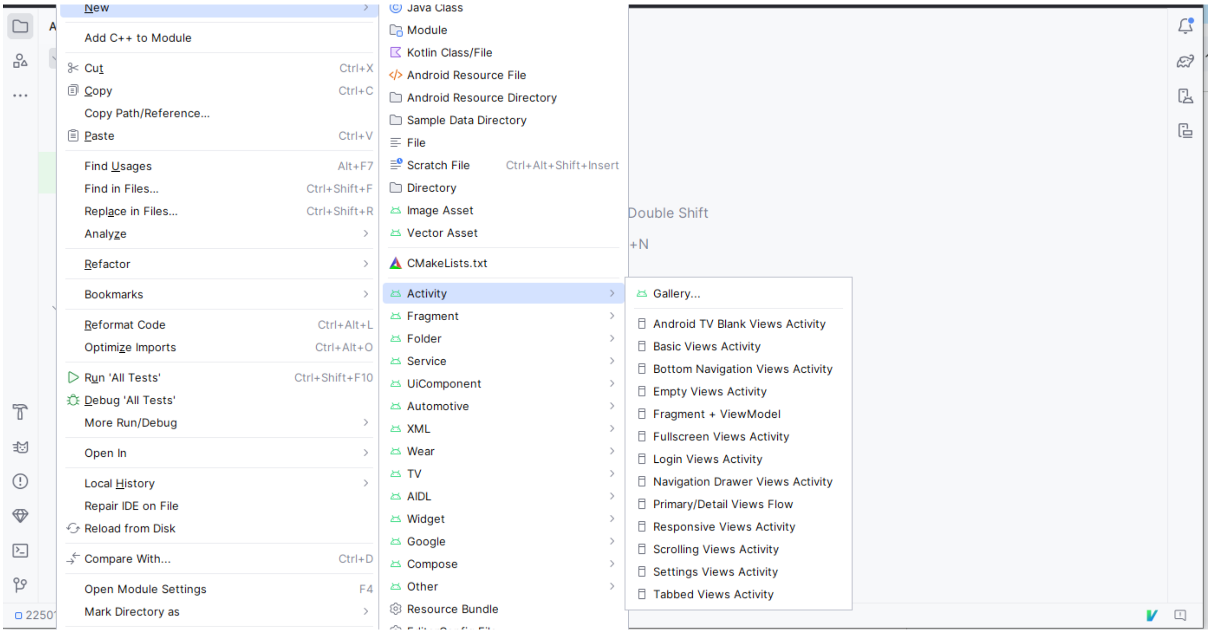
Language: Java

Minimum SDK: API 24 ("Nougat"; Android 7.0)

Your app will run on approximately **97.4%** of devices.
Help me choose

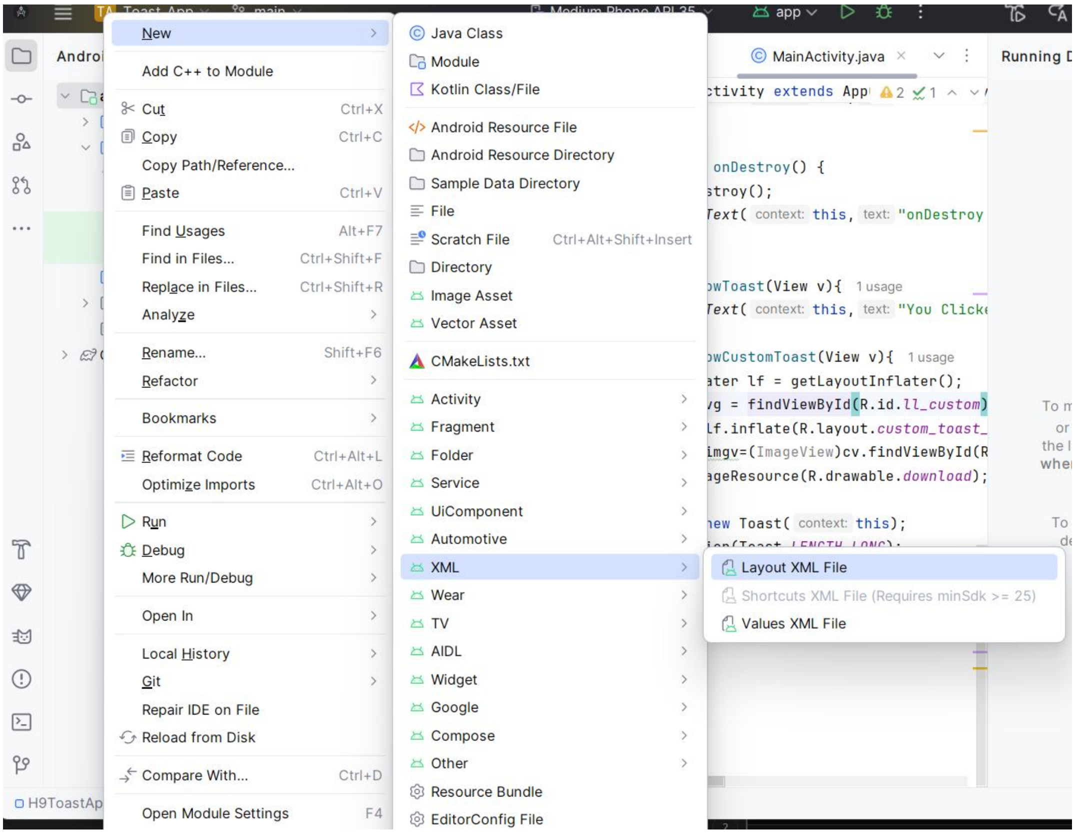
Build configuration language: Kotlin DSL (build.gradle.kts) [Recommended]

4. To create a new activity, right-click on App > New > Activity > Empty Views Activity, name it MainActivity, check the Launcher Activity option, and click Finish.

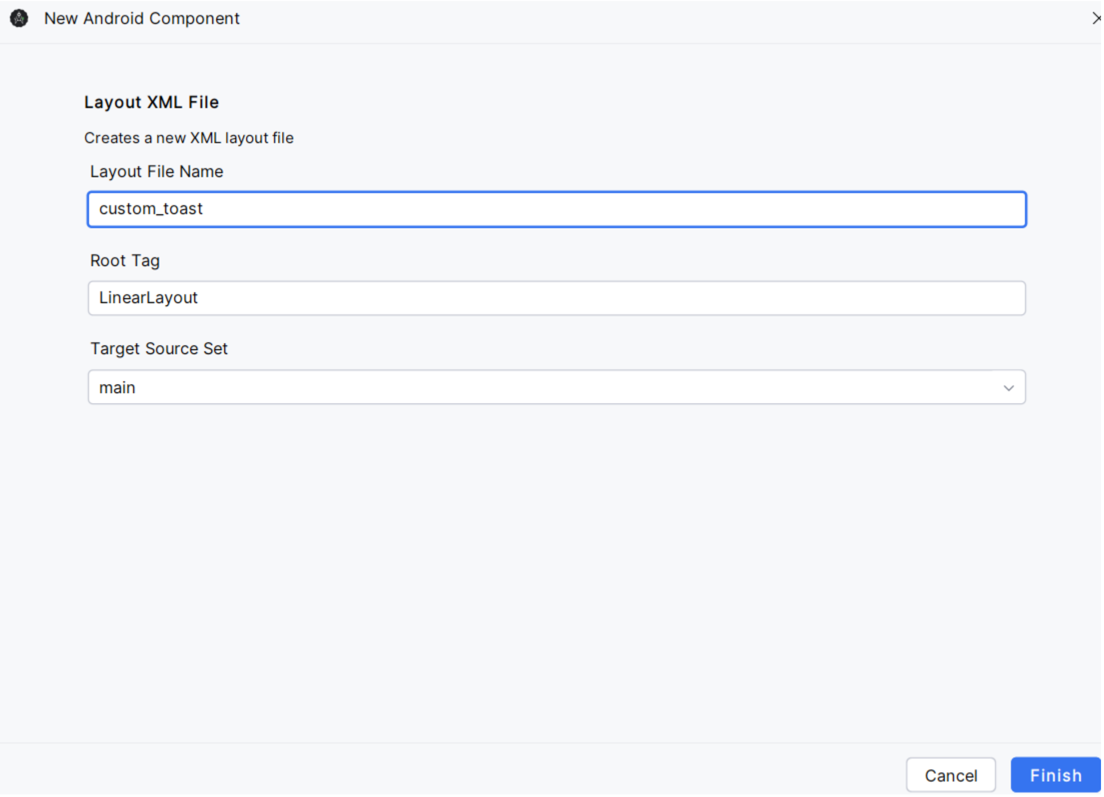


5. Write code in MainActivity.java to implement lifecycle methods and display Custom Toast.

6. To create a new layout file, right-click on App > New > XML > Layout XML file, name it custom_toast_layout, and click Finish.



```
ctivity extends App
onDestroy() {
stroy();
Text( context: this, text: "onDestroy
owToast(View v){  1 usage
Text( context: this, text: "You Click
owCustomToast(View v){  1 usage
ater lf = getLayoutInflater();
vg = findViewById(R.id.ll_custom)
lf.inflate(R.layout.custom_toast_
imgv=(ImageView)cv.findViewById(R
ageResource(R.drawable.download);
new Toast( context: this);
ion(Toast.LENGTH_LONG);
```



New Android Component

Layout XML File

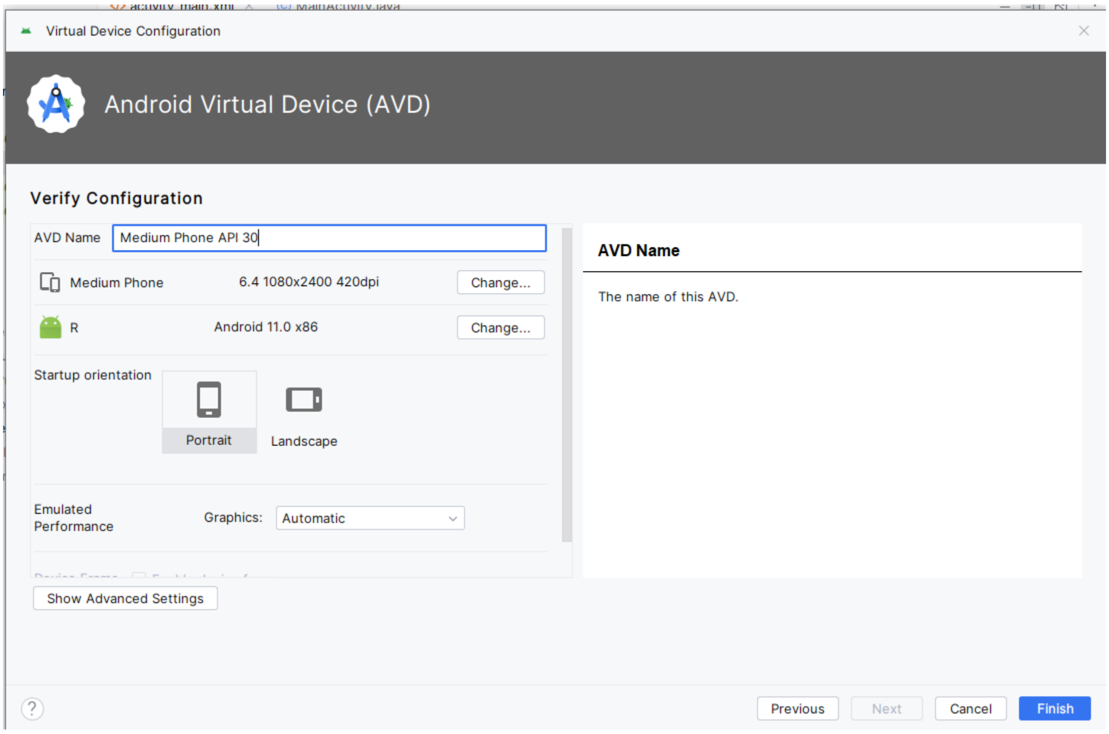
Creates a new XML layout file

Layout File Name: custom_toast

Root Tag: LinearLayout

Target Source Set: main

7. To create an Android Virtual Device (AVD) for testing.
    1. In Android Studio, Click **Device Manager** → Click **"+" (Create Device)**.
    2. Select a **mobile model**, choose a **system image**, configure settings, and click **Finish**.
    3. Click **Play** to launch the emulator.



Virtual Device Configuration

Android Virtual Device (AVD)

Verify Configuration

AVD Name: Medium Phone API 30

Medium Phone 6.4 1080x2400 420dpi

R Android 11.0 x86

Startup orientation: Portrait

Emulated Performance Graphics: Automatic

AVD Name

The name of this AVD.

8. Run the application using the emulator.
9. Observe Custom Toast messages appearing at each lifecycle stage.

PROGRAMS:

MainActivity.java

```
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;

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

public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });
    }
    public void showToast(View v){
        Toast.makeText(this, "Harsha Vardhan", Toast.LENGTH_LONG).show();
    }
    @Override
    protected void onStart(){
        super.onStart();
        Toast.makeText(this,"On start called",Toast.LENGTH_LONG).show();
    }
    @Override
    protected void onRestart(){
        super.onRestart();
        Toast.makeText(this, "On Restart called", Toast.LENGTH_SHORT).show();
    }
    @Override
    protected void onPause(){
        super.onPause();
        Toast.makeText(this,"on Pause called",Toast.LENGTH_LONG).show();
    }
    @Override
    protected void onResume(){
        super.onResume();
        Toast.makeText(this,"on Resume called",Toast.LENGTH_LONG).show();
    }
    @Override
    protected void onStop() {
        super.onStop();
         Toast.makeText(getApplicationContext(), "onStop Called", Toast.LENGTH_LONG).show();
    }

    public void showCustomToast(View v){
        LayoutInflater lf=getLayoutInflater();
        ViewGroup vg=findViewById(R.id.ll_ctoast);
        View cv=lf.inflate(R.layout.custom_toast_layout,vg);
        ImageView imgv=cv.findViewById(R.id.imageView);
        imgv.setImageResource(R.drawable.spidy);
        TextView tv=cv.findViewById(R.id.textView);
        tv.setText("SpiderMan");

        Toast t=new Toast(this);
        t.setDuration(Toast.LENGTH_LONG);
        t.setView(cv);
        t.show();
    }
}
```

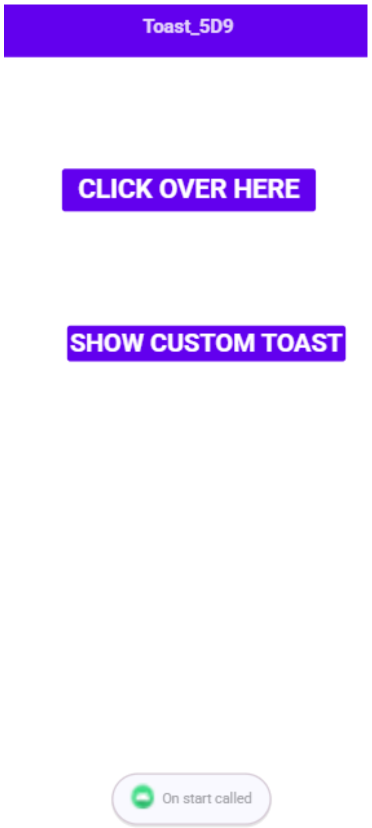
activity_main.xml

```
<?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:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="72dp"
        android:layout_marginTop="132dp"
        android:onClick="showToast"
        android:text="CLICK OVER HERE"
        android:textSize="24sp"
        android:textStyle="bold"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="104dp"
        android:onClick="showCustomToast"
        android:text="SHOW CUSTOM TOAST"
        android:textSize="24sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.494"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/button"
        app:layout_constraintVertical_bias="0.0" />

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

Custom_toast_layout.xml

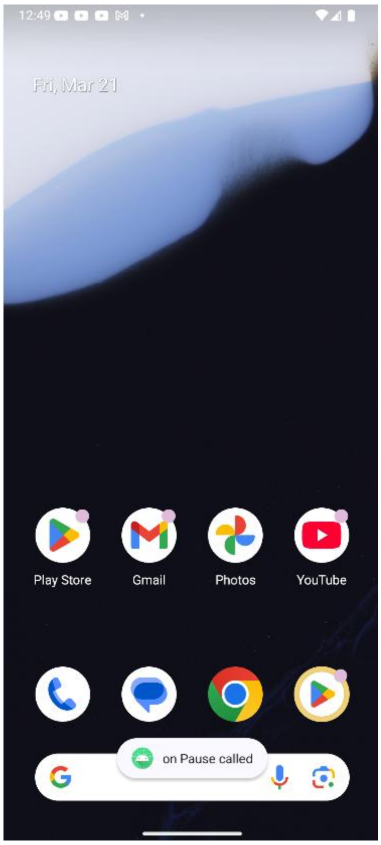
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/ll_ctoast"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        tools:srcCompat="@tools:sample/avatars" />

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="TextView" />
</LinearLayout>
```

OUTPUT:



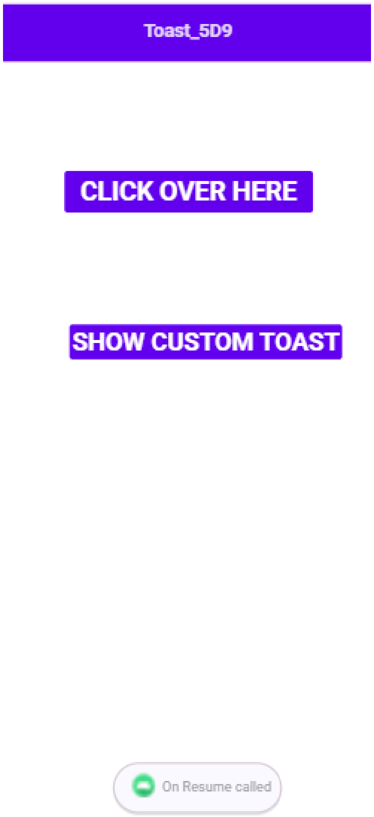
On start called

When device is running
onStart() is evoked



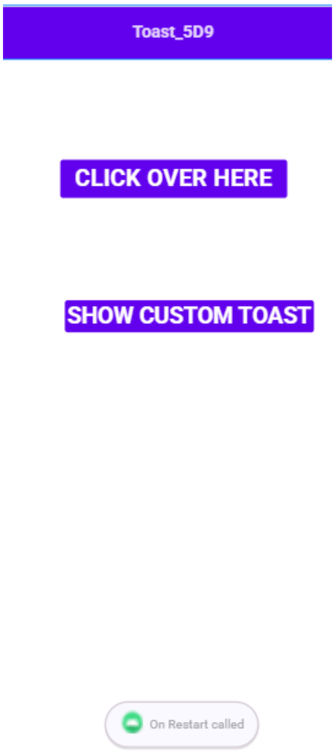
on Pause called

When app is closed
onPause() is evoked



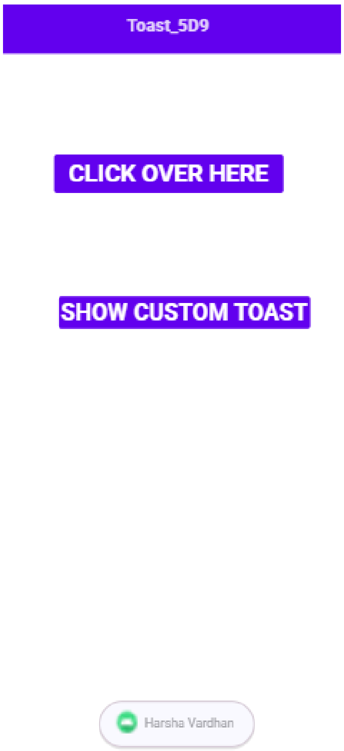
On Resume called

When app is opened again
onResume() is evoked



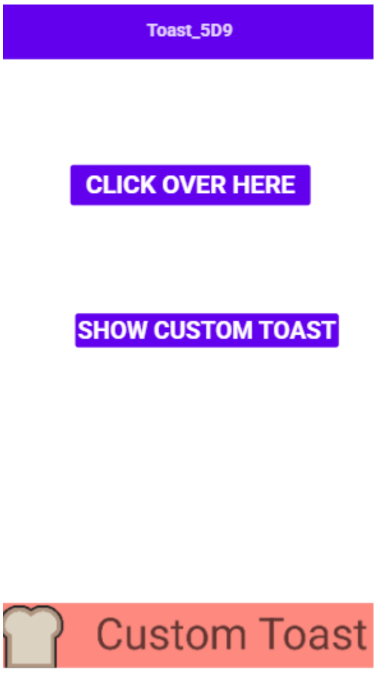
On Restart called

When emulator is switch on
onRestart() is evoked



Harsha Vardhan

When Toast is clicked
showToast() is evoked



Custom Toast

When CUSTOM_TOAST is clicked
showcustom_toast() is evoked

RESULT: Mobile application based on the concept activity life cycle with Custom Toast built successfully.

Experiment: 2 Date:27-12-24

Build mobile application using different layouts

AIM: To build mobile application using different layouts.

EXPERIMENTAL REQUIREMENTS: PC with Android Studio and Internet connection

THEORY:

Mobile applications require well-structured user interfaces (UI) to provide a seamless experience. In Android development, layouts define how UI components are arranged on the screen. Choosing the right layout ensures efficiency, readability, and responsiveness.

A **layout** in Android is a container that organizes UI elements such as buttons, text views, and images. Android provides multiple layouts, each designed for specific use cases. The commonly used layouts include:

a) Linear Layout

- Arranges UI components in a single direction, either **vertically** or **horizontally**.
- Ensures a simple and sequential arrangement of elements.
- Suitable for lists, forms, or stacked elements.
- Requires careful management of screen space as each element takes up its own row or column.

b) Relative Layout

- Positions UI components relative to each other or to the parent container.
- Allows flexibility in arranging elements without a fixed order.
- Reduces nesting of layouts, making UI design more efficient.
- Ideal for placing components dynamically based on screen size and content.

c) Table Layout

- Organizes UI components in a grid-like structure using rows and columns.
- Useful for displaying structured data such as tables, forms, and schedules.
- Each row can contain multiple UI elements, and columns are automatically adjusted based on content.
- Best for applications requiring a tabular representation of information.

d) Frame Layout

- Designed to hold a single child view, though multiple views can be layered.
- Used for displaying overlapping elements, such as images, videos, or floating buttons.
- Helps in creating UI elements that require stacking, like media players or pop-up dialogs.
- Simple and efficient for cases where only one view is needed at a time.

e) Tab Layout

- Provides a tab-based navigation system, allowing users to switch between different sections.

- Enhances the user experience by organizing content into multiple views accessible through tabs.
- Often used with navigation components such as ViewPager or fragments to handle different screens.
- Ideal for applications that require categorized navigation, like social media apps, shopping apps, or settings menus.

PROCEDURE:

1. Open Android Studio and create a new project.
2. Select Empty Activity, enter project details, and click Finish.
3. Create a main activity, as the Launcher Activity.
4. Open res/layout/activity_main.xml and choose a layout type (LinearLayout, RelativeLayout, TableLayout, FrameLayout, or TabLayout).
5. Use Intents in MainActivity.java to navigate between activities.
6. Click Run to test the application on an emulator or device.

PROGRAMS:

MainActivity.java

```
package com.example.layout;

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

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

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });
    }

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

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

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

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

    public void show_tabl_activity(View v){
        Intent i = new Intent(this, TabLayoutActivity.class);
        this.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:id="@+id/main"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity">

<Button
   android:id="@+id/btn_linear"
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:layout_marginTop="160dp"
   android:onClick="show_ll_activity"
   android:text="Linear Layout"
   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/btn_table"
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:layout_marginTop="60dp"
   android:onClick="show_tl_activity"
   android:text="Table Layout"
   app:layout_constraintEnd_toEndOf="parent"
   app:layout_constraintHorizontal_bias="0.511"
   app:layout_constraintStart_toStartOf="parent"
   app:layout_constraintTop_toBottomOf="@+id/btn_linear" />

<Button
   android:id="@+id/btn_frame"
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:layout_marginTop="68dp"
   android:onClick="show_fl_activity"
   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/btn_table" />

<Button
    android:id="@+id/btn_relative"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="64dp"
    android:onClick="show_rl_activity"
    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/btn_frame" />

<Button
    android:id="@+id/btn_tablayout"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="36dp"
    android:onClick="show_tabl_activity"
    android:text="Tab Layout"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/btn_relative" />

<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="84dp"
    android:text="Layout_5D9"
    android:textSize="24sp"
    app:layout_constraintBottom_toTopOf="@+id/btn_linear"
```

```
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.0" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

LinearLayout.java

```
package com.example.layout;

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

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

public class LinearLayoutActivity extends AppCompatActivity {
    private LinearLayout mainLayout;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_linear_layout);

        mainLayout = findViewById(R.id.main);

        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });
    }
    public void changeOrientation(View view) {
        if (mainLayout.getOrientation() == LinearLayout.VERTICAL) {
            mainLayout.setOrientation(LinearLayout.HORIZONTAL);
        } else {
            mainLayout.setOrientation(LinearLayout.VERTICAL);
        }
    }
}
```

activity_linear_layout.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    xmlns:tools="http://schemas.android.com/tools"
    tools:context=".LinearLayoutActivity">

    <Button
        android:id="@+id/button"
        android:layout_width="133dp"
        android:layout_height="91dp"
        android:onClick="changeOrientation"
        android:text="CHANGE ORIENTATION"
        android:textSize="14sp" />

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="159dp"
        android:layout_height="165dp"
        android:src="@drawable/dog" />

    <ImageView
        android:id="@+id/imageView2"
        android:layout_width="125dp"
        android:layout_height="195dp"
        android:src="@drawable/flower" />
</LinearLayout>
```

FrameActivity.java

```
package com.example.layout;

import android.os.Bundle;

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

public class FrameLayoutActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_frame_layout);
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });
    }
}
```

activity_frame.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:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".FrameLayoutActivity">

    <Button
        android:id="@+id/button"
        android:layout_width="297dp"
        android:layout_height="102dp"
        android:layout_margin="50sp"
        android:text="DOG" />

    <ImageView
        android:id="@+id/imageView4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="69sp"
        android:layout_marginTop="250sp"
        app:srcCompat="@drawable/dog" />

</FrameLayout>
```

TablesLayout.java

```
package com.example.layout;

import android.os.Bundle;
import android.widget.ImageView;

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

public class TableLayoutActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_table_layout);
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });
        ImageView imgv1=(ImageView) findViewById(R.id.img1);
        imgv1.setImageResource(R.drawable.img);
        ImageView imgv2=(ImageView) findViewById(R.id.img2);
        imgv2.setImageResource(R.drawable.lion);
        ImageView imgv3=(ImageView) findViewById(R.id.img3);
        imgv3.setImageResource(R.drawable.flower);
        ImageView imgv4=(ImageView) findViewById(R.id.img4);
        imgv4.setImageResource(R.drawable.dog);

    }
}
```

activity_tables_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"
    tools:context=".TableLayoutActivity"
    android:id="@+id/main"
    android:layout_width="409dp"
    android:layout_height="729dp"
    tools:layout_editor_absoluteX="1dp"
    tools:layout_editor_absoluteY="1dp">

    <TableRow
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <ImageView
            android:id="@+id/img1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:src="@drawable/img" />
    </TableRow>

    <TableRow
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <ImageView
            android:id="@+id/img2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:src="@drawable/lion" />
    </TableRow>

    <TableRow
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <ImageView
            android:id="@+id/img3"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:src="@drawable/flower" />
    </TableRow>

    <TableRow
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <ImageView
            android:id="@+id/img4"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:src="@drawable/dog" />
    </TableRow>
</TableLayout>
```

TabsLayout.java

```
package com.example.layout;

import android.os.Bundle;

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

public class TabLayoutActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_tab_layout);
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });
    }
}
```

activity_tabs_layout.xml

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

    <com.google.android.material.tabs.TabLayout
```

```
        android:id="@+id/tab"
        android:layout_width="409dp"
        android:layout_height="wrap_content"
        tools:layout_editor_absoluteX="1dp"
        tools:layout_editor_absoluteY="10dp"
        tools:ignore="MissingConstraints">

        <com.google.android.material.tabs.TabItem
            android:id="@+id/tab_A"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/A" />

        <com.google.android.material.tabs.TabItem
            android:id="@+id/tab_B"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/B" />

        <com.google.android.material.tabs.TabItem
            android:id="@+id/tab_C"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/C" />
    </com.google.android.material.tabs.TabLayout>
</androidx.constraintlayout.widget.ConstraintLayout>
```

RelativeLayout.java

```
package com.example.layout;

import android.os.Bundle;

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

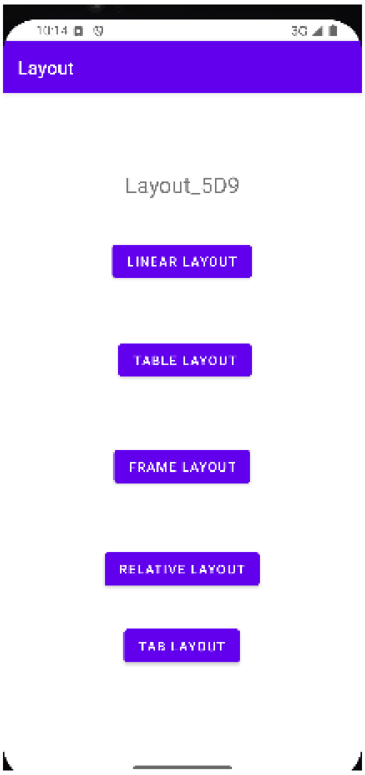
```
public class RelativeLayoutActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_relative_layout);
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });
    }
}
```

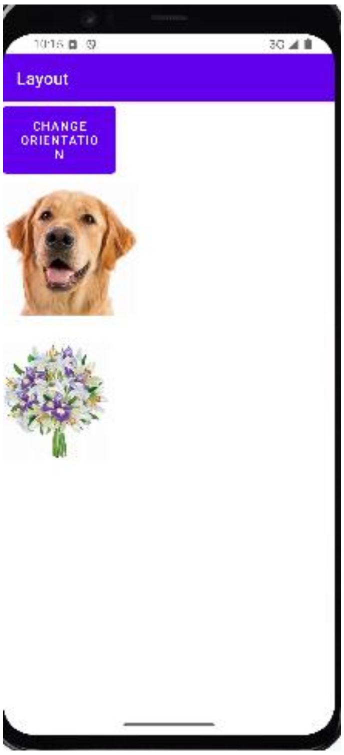
activity_relatives_layout.xml

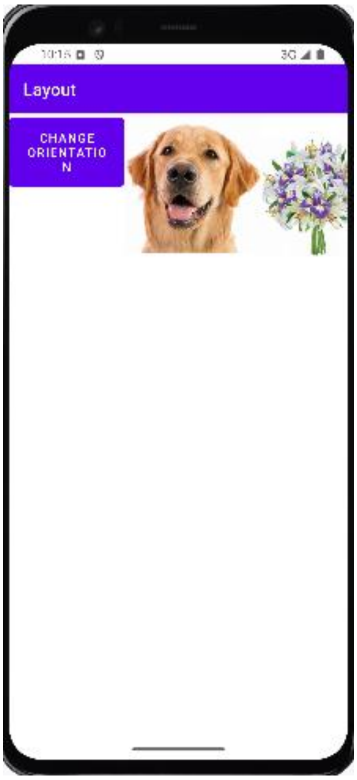
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"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

    <TextView
        android:id="@+id/txt_title"
        android:layout_width="wrap_content"
        android:layout_height="226dp"
        android:layout_alignParentStart="true"
        android:layout_alignParentEnd="true"
        android:layout_marginStart="48dp"
        android:layout_marginTop="50dp"
        android:layout_marginEnd="65dp"
        android:fontFamily="casual"
        android:text="Appilication From"
        android:textSize="25sp"
        android:textStyle="bold" />

    <TextView
        android:id="@+id/text_name1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBottom="@id/txt_title"
        android:layout_marginLeft="60dp"
        android:layout_marginBottom="135dp"
        android:text="name" />

    <EditText
        android:id="@+id/edittext1"
        android:layout_width="139dp"
        android:layout_height="wrap_content"
        android:layout_alignStart="@+id/txt_title"
        android:layout_alignEnd="@+id/txt_title"
        android:layout_alignBottom="@+id/txt_title"
        android:layout_marginStart="74dp"
        android:layout_marginEnd="85dp"
        android:layout_marginBottom="127dp"
        android:ems="10"
        android:inputType="text"
        android:layoutDirection="inherit"
        android:minHeight="48dp"
        android:text="Name" />

    <TextView
        android:id="@+id/text_name2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentStart="true"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginStart="57dp"
```

```
        android:layout_marginEnd="296dp"
        android:layout_marginBottom="532dp"
        android:text="Roll No" />

    <EditText
        android:id="@+id/edittext2"
        android:layout_width="127dp"
        android:layout_height="wrap_content"
        android:layout_alignParentStart="true"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginStart="123dp"
        android:layout_marginEnd="160dp"
        android:layout_marginBottom="527dp"
        android:ems="10"
        android:inputType="text"
        android:minHeight="48dp"
        android:text="Roll No" />

    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentStart="true"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginStart="136dp"
        android:layout_marginEnd="181dp"
        android:layout_marginBottom="441dp"
        android:text="Button" />

</RelativeLayout>
```

OUTPUT:



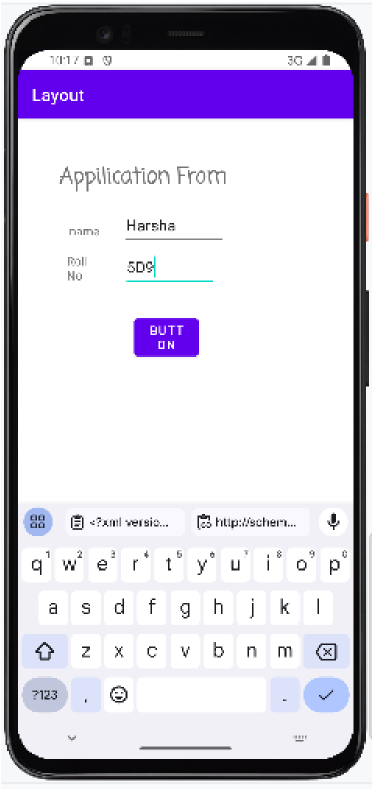




Main Activity

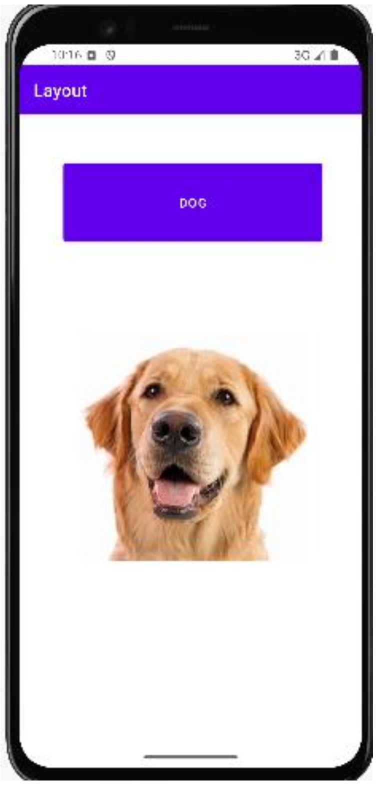
Linear layout in horizontal orientation

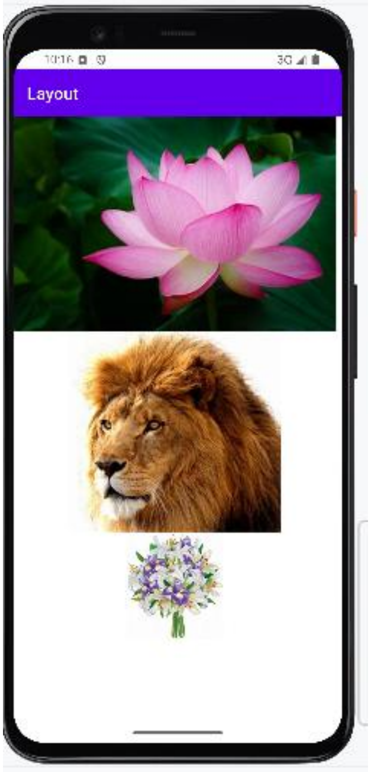
Linear layout in vertical orientation



name Harsha

Roll No 5D9

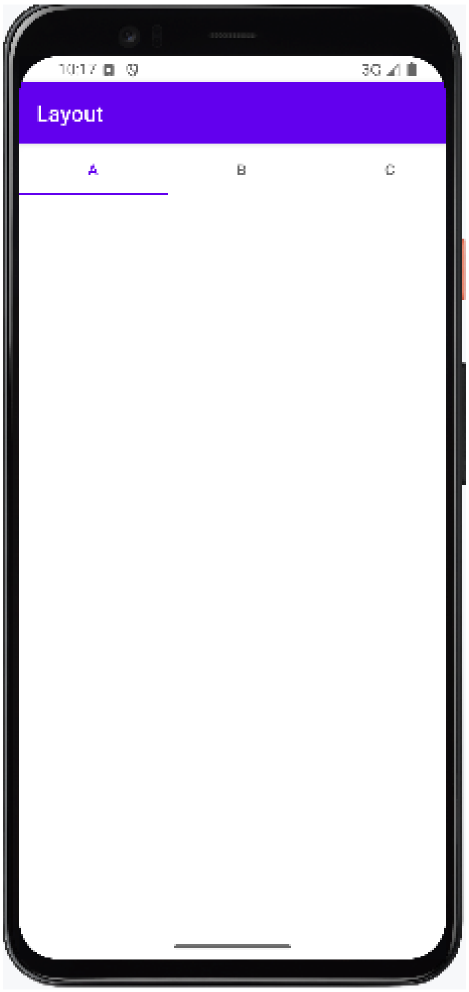




Relative Layout

Frame Layout

Table Layout



Tab Layout

RESULT: Mobile application using different layouts is built successfully.

Experiment: 3 Date:10-1-2025

Build mobile application using different dialogs

AIM: To build mobile application using different dialogs.

EXPERIMENTAL REQUIREMENTS: PC with Android Studio and Internet connection

THEORY:

A **dialog** is a small window that prompts the user to make a decision or enter additional information. A dialog doesn't fill the screen and is normally used for modal events that require users to take an action before they can proceed. The commonly used dialogs include:

a) Alert Dialog

- Android AlertDialog can be used to display the dialog message with OK and Cancel buttons.
- It can be used to interrupt and ask the user about his/her choice to continue or discontinue.
- This Dialog maximum 3 buttons allowed.

b) DatePicker Dialog

- Android DatePickerDialog puts a DatePicker on a Dialog.
- It allows user to select Date(day,month and year) in our application.

c) TimePicker Dialog

- Android TimePicker is a component that permits users to select a time including hour and minute.
- 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 or spinner format.
- Android provides this functionality through TimePicker class.

d) Custom Dialog

- Custom dialog is used to customize a dialog of your needs.
- You can create your own custom dialog with custom characteristics.

PROCEDURE:

1. Open Android Studio and create a new project.
2. Select Empty Activity, enter project details, and click Finish.
3. Create a main activity, as the Launcher Activity.
4. In main activity, create 3 buttons to access Alert dialog, time picker dialog and date picker dialog.
5. After writing the necessary code, click Run to test the application on an emulator or device.

PROGRAMS:

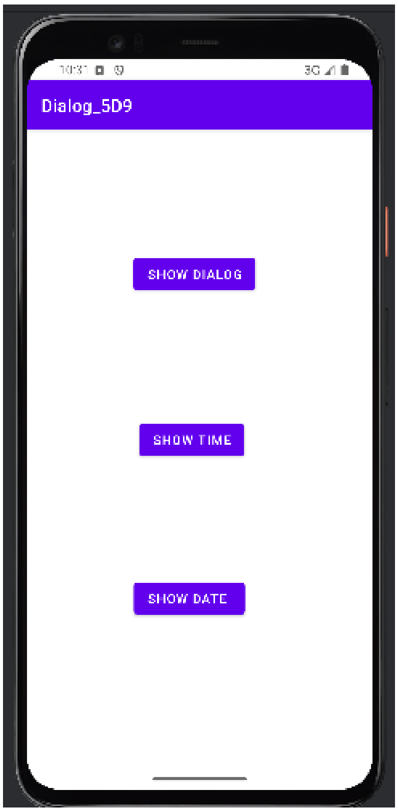
MainActivity.java

```
package com.example.dialog_5d9;

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

import java.util.Calendar;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });
    }
    public void showDialog(View v) {
        AlertDialog.Builder adb = new AlertDialog.Builder(this);
        Toast.makeText(this, "Welcome", Toast.LENGTH_SHORT).show();
        adb.setTitle("Harsha Vardhan");
        adb.setMessage("This is the dialog created by Harsha!!");
        adb.setPositiveButton("Continue", (dialogInterface, i) ->
                Toast.makeText(MainActivity.this, "Thank You", Toast.LENGTH_LONG).show());
        adb.setNegativeButton("Cancel", (dialogInterface, i) ->
                Toast.makeText(MainActivity.this, "Try Again", Toast.LENGTH_LONG).show());
        AlertDialog ad = adb.create();
        ad.show();
    }

    public void showTime(View v) {
        Calendar c = Calendar.getInstance();
        int dh = c.get(Calendar.HOUR);
        int dm = c.get(Calendar.MINUTE);
        TimePickerDialog tpd = new TimePickerDialog(this, (view, hourOfDay, minute) ->
                Toast.makeText(MainActivity.this, hourOfDay + "H: " + minute + "M",
Toast.LENGTH_LONG).show(),
                dh, dm, true);
        tpd.show();
    }

    public void showDate(View v) {
        Calendar c = Calendar.getInstance();
        int dYear = c.get(Calendar.YEAR);
        int dMon = c.get(Calendar.MONTH);
        int dDay = c.get(Calendar.DAY_OF_MONTH);

        DatePickerDialog dpd = new DatePickerDialog(this, (datePicker, i, i1, i2) ->
                Toast.makeText(MainActivity.this, i + "Y:" + i1 + "M:" + i2 + "D",
Toast.LENGTH_LONG).show(),
                dYear, dMon, dDay);
        dpd.setTitle("Pick a date please");
        dpd.show();
    }
}
```

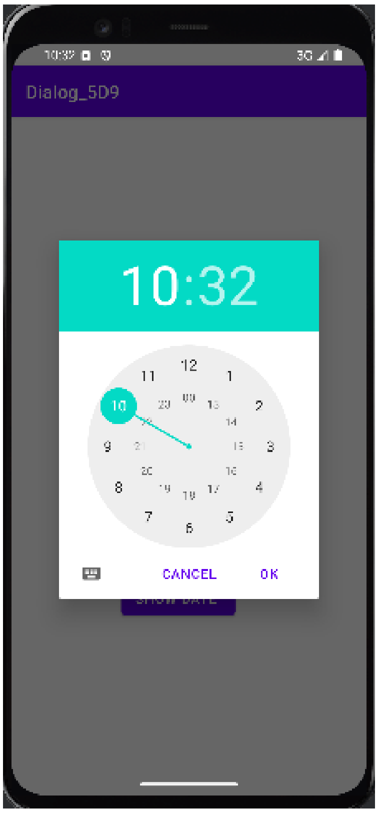
activity_main.xml

```
<?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:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="showDialog"
        android:text="SHOW DIALOG"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.477"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.207" />

    <Button
        android:id="@+id/button5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="140dp"
        android:onClick="showTime"
        android:text="SHOW TIME"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.467"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/button2" />

    <Button
        android:id="@+id/button6"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="132dp"
        android:onClick="showDate"
        android:text="SHOW DATE "
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.456"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/button5"
        app:layout_constraintVertical_bias="0.0" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

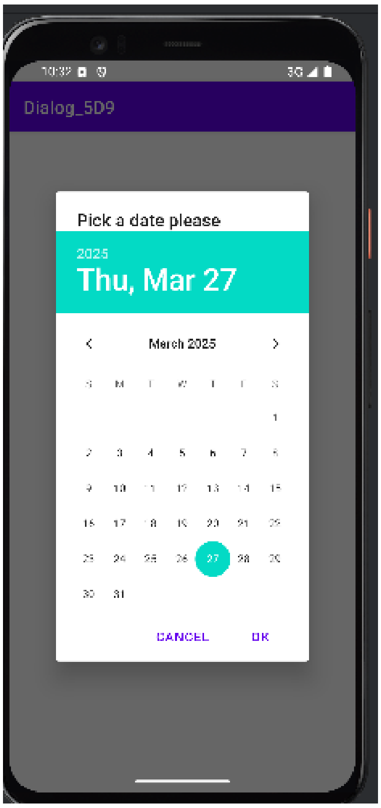
OUTPUT:



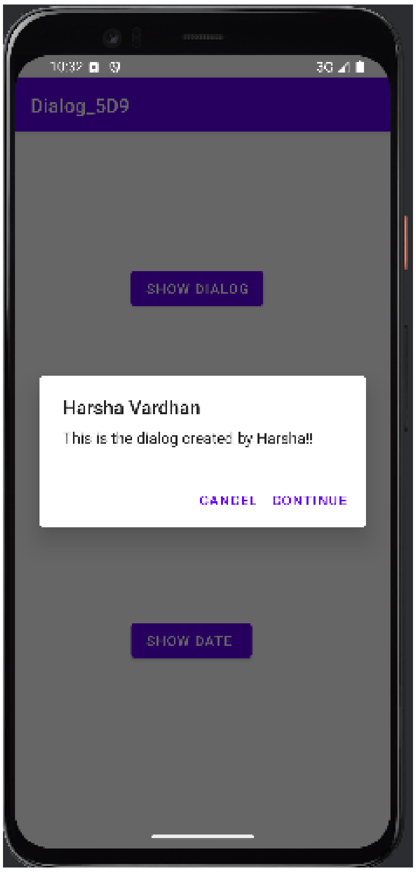
Main Activity



TimePicker Dialog



DatePicker Dialog



Harsha Vardhan

This is the dialog created by Harsha!!

Alert Dialog

RESULT: Mobile application using dialogs has been built successfully.

Experiment: 4 Date: 31-1-2025

Build mobile application using Recycler View

AIM: To build mobile application using Recycler View.

EXPERIMENTAL REQUIREMENTS: PC with Android Studio and Internet connection

THEORY:

In Android, RecyclerView is a powerful and flexible UI component used for displaying large sets of data efficiently by recycling views, making it an improved version of ListView. It works by creating only the necessary number of views that fit on the screen and reusing them as the user scrolls, reducing memory usage and improving performance.

Below are some of the essential components and their functionalities:

a) RecyclerView.Adapter

The Adapter acts as a bridge between the data source and the RecyclerView, binding data to each item in the list. It contains three important methods:

- onCreateViewHolder(ViewGroup parent, int viewType) – Creates new ViewHolder instances that hold references to item views.
- onBindViewHolder(ViewHolder holder, int position) – Binds data to the ViewHolder at the specified position.
- getItemCount() – Returns the total number of items in the dataset.

b) RecyclerView.ViewHolder

ViewHolder is responsible for holding references to individual item views in RecyclerView, reducing the need to repeatedly find views, thus improving performance.

- It extends the RecyclerView.ViewHolder class and holds UI elements inside each row.
- Helps in reusing views to minimize resource consumption.

c) RecyclerView.LayoutManager

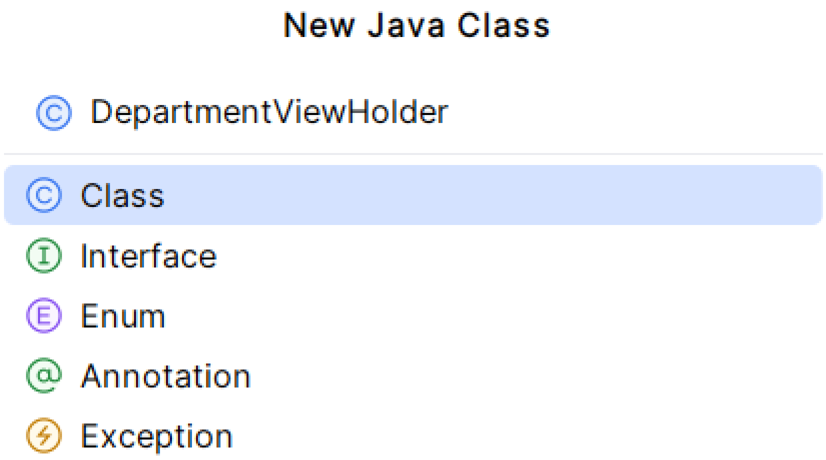
LayoutManager defines how items in the RecyclerView should be arranged on the screen. Android provides three built-in LayoutManagers:

- LinearLayoutManager – Displays items in a vertical or horizontal list (like a ListView).
- GridLayoutManager – Displays items in a grid pattern (like a photo gallery).
- StaggeredGridLayoutManager – Displays items in staggered rows or columns, commonly used for Pinterest-like layouts.

PROCEDURE:

1. Open Android Studio and create a new project.
2. Select Empty Activity, enter project details, and click Finish.
3. Create a main activity, as the Launcher Activity.
4. Create itemlayout_recyclerview xml file as a relative layout to display the view.
5. Create Department, DepartmentViewHolder, DepartmentAdapter and DAdapter java files.

6. To generate a .java file:
    a. Double click on package
    b. Click on New > Java
    c. Give name to the Java file



New Java Class

DepartmentViewHolder

7. After writing the necessary code, click Run to test the application on an emulator or device.

PROGRAMS:

<u>MainActivity.java</u>

```
package com.example.recycler;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.os.Bundle;

import com.example.recycler.Department;
import com.example.recycler.DepartmentAdaptor;
import com.example.recycler.R;

import java.util.ArrayList;
import java.util.List;

public class MainActivity extends AppCompatActivity {
    RecyclerView rcv;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        List<Department> depts = new ArrayList<>();
        depts.add(new Department(R.drawable.cse, "CSE", "210"));
        depts.add(new Department(R.drawable.ece, "ECE", "140"));
        depts.add(new Department(R.drawable.eee, "EEE", "70"));
        rcv = (RecyclerView) this.findViewById(R.id.rcv);
        rcv.setAdapter(new DepartmentAdaptor(getApplicationContext(), depts));
        rcv.setLayoutManager(new LinearLayoutManager(getApplicationContext()));
    }
}
```

activity_main.xml

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

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/rcv"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tools:layout_editor_absoluteX="1dp"
        tools:layout_editor_absoluteY="1dp" />
</FrameLayout>
```

itemlayout_recyclerview.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_dpic"
        android:layout_width="120dp"
        android:layout_height="120dp"
        android:layout_margin="10dp"
        tools:srcCompat="@tools:sample/avatars" />

    <TextView
        android:id="@+id/txtv_dname"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_above="@+id/txtv_dseats"
        android:layout_marginLeft="8dp"
        android:layout_marginBottom="3dp"
        android:layout_toRightOf="@id/imgv_dpic"
        android:text="Dept name"
        android:textSize="24sp" />

    <TextView
        android:id="@+id/txtv_dseats"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBottom="@id/imgv_dpic"
        android:layout_marginLeft="13dp"
        android:layout_marginBottom="18dp"
        android:layout_toRightOf="@id/imgv_dpic"
        android:text="seats"
        android:textSize="24sp" />
</RelativeLayout>
```

Department.java

```
package com.example.recycler;

public class Department {
    int dpic;
    String dname;
    String dseats;
    public Department(int dpic, String dname, String dseats) {
        this.dpic = dpic;
        this.dname = dname;
        this.dseats = dseats;
    }

    public int getDpic() {
        return dpic;
    }

    public String getDname() {
        return dname;
    }

    public String getDseats() {
        return dseats;
    }

    public void setDpic(int dpic) {
        this.dpic = dpic;
    }

    public void setDseats(String dseats) {
        this.dseats = dseats;
    }

    public void setDname(String dname) {
        this.dname = dname;
    }
}
```

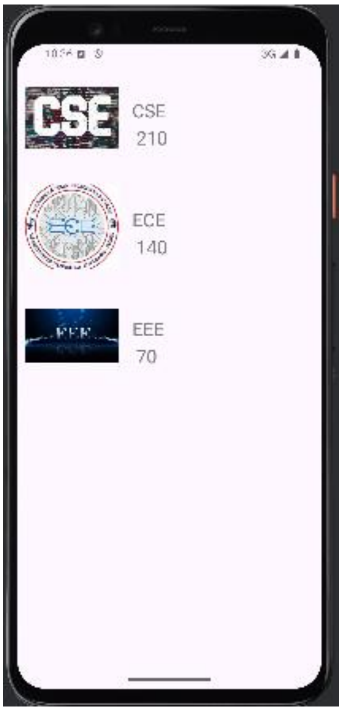
DepartmentAdapter.java

```
package com.example.recycler;

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 DepartmnetAdaptor extends RecyclerView.Adapter<DepartmentViewHolder> {

    Context ct;
    List<Department> depts;

    public DepartmnetAdaptor(Context ct, List<Department> depts) {
        this.ct = ct;
        this.depts = depts;
    }

    @NonNull
    @Override
    public DepartmentViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        return new
DepartmentViewHolder(LayoutInflater.from(ct).inflate(R.layout.itemlayout_recyclerview, parent,
false));
    }

    @Override
    public void onBindViewHolder(@NonNull DepartmentViewHolder holder, int position) {
        holder.dpic.setImageResource(depts.get(position).getDpic());
        holder.dname.setText(depts.get(position).getDname());
        holder.dseats.setText(depts.get(position).getDseats());
    }

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

DepartmentViewHolder.java

```
package com.example.recycler;

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

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

public class DepartmentViewHolder extends RecyclerView.ViewHolder {
    ImageView dpic;
    TextView dname,dseats;

    public DepartmentViewHolder(@NonNull View itemView) {
        super(itemView);
        dpic=(ImageView) itemView.findViewById(R.id.imgv_dpic);
        dname=(TextView) itemView.findViewById(R.id.txtv_dname);
        dseats=(TextView) itemView.findViewById(R.id.txtv_dseats);
    }

}
```

OUTPUT:



CSE
210

ECE
140

EEE
70

Main Activity

RESULT: Mobile application using Recycler View has been built successfully.

Experiment: 5 **Date:** 7-02-2025

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

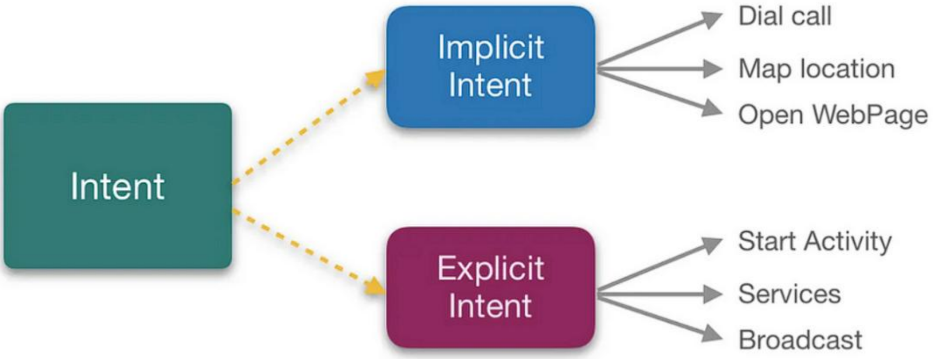
AIM: To build mobile application to switch from one activity to another using Intent.

EXPERIMENTAL REQUIREMENTS: PC with Android Studio and Internet connection

THEORY:

In Android, Intents are a powerful messaging system that allows components within an application—or even across different applications—to communicate and perform actions. Intents enable developers to navigate between activities, start services, broadcast messages, or request actions from other apps, such as opening a web page or sharing content.

There are two main types of intents: explicit intents and implicit intents. Explicit intents are used when the target component is known, such as launching a specific activity within the same app. For example, an explicit intent can be used to move from a login screen to a home screen. On the other hand, implicit intents do not specify a particular component but declare an action, allowing the system to determine which app can handle it. A common example of an implicit intent is opening a URL in a web browser or sharing an image via different apps. Intents also allow passing data between components using extras, making them essential for seamless interaction within Android applications.



PROCEDURE:

1. Open Android Studio and start a new project.
2. Select "No Views Activity" and click Next.
3. Enter project details (Name: SwitchIntent, Language: Java) and click Finish.
4. To create a new activity, right-click on App > New > Activity > Empty Views Activity, name it MainActivity, check the Launcher Activity option, and click Finish.
5. Write code in MainActivity.java to implement intent.
6. To create a new layout file, right-click on App > New > XML > Layout XML file, name it, and click Finish.
7. Run the application using the emulator.

PROGRAMS:

MainActivity.java

```
package com.example.a22501a05g7_exp5;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;
import com.example.a22501a05g7_exp5.R;
import com.example.a22501a05g7_exp5.SecondActivity;
public class MainActivity extends AppCompatActivity {
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
   }
   public void Show_second(View v) {
      Intent i = new Intent(this, SecondActivity.class);
      startActivity(i);
   }
   public void send_email(View v) {
      Intent emailIntent = new Intent(Intent.ACTION_SEND);
      emailIntent.setData(Uri.parse("mailto:"));
      emailIntent.setType("text/plain");
      emailIntent.putExtra(Intent.EXTRA_EMAIL, new String[]{""});
      emailIntent.putExtra(Intent.EXTRA_CC, new String[]{""});
      emailIntent.putExtra(Intent.EXTRA_SUBJECT, "Your subject");
      emailIntent.putExtra(Intent.EXTRA_TEXT, "Email message goes here");
```

```
        try {
            startActivity(Intent.createChooser(emailIntent, "Send mail..."));
            finish();
            Log.i("Finished sending email.", "");
        } catch (android.content.ActivityNotFoundException ex) {
            Toast.makeText(MainActivity.this, "There is no email client installed.",
Toast.LENGTH_SHORT).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:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/SwitchToActivity"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="Show_second"
        android:text="Show Second Activity"
        app:layout_constraintBottom_toTopOf="@+id/button2"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.46"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.557" />

    <Button
```

```
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="448dp"
        android:onClick="send_email"
        android:text="SEND EMIAL"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.46"
        app:layout_constraintStart_toStartOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

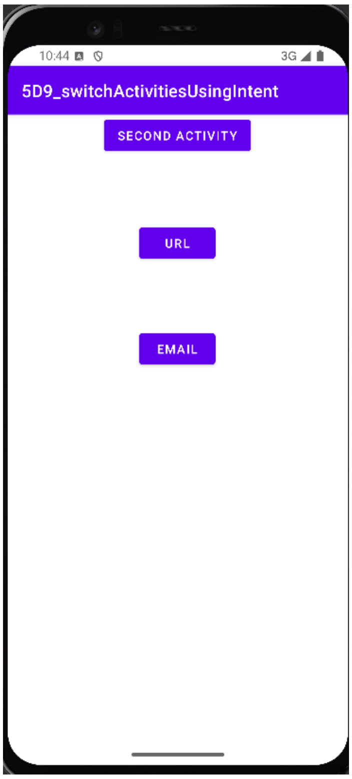
SecondActivity.java

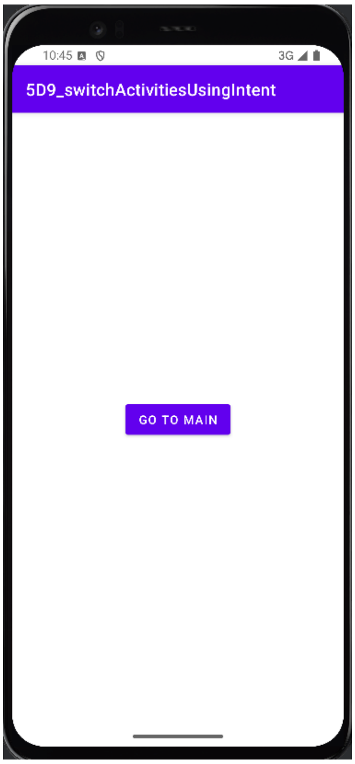
```
package com.example.a22501a05g7_exp5;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;
public class SecondActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_second);
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });
    }
    public void show_main(View v){
        Intent i=new Intent(this,MainActivity.class);
        startActivity(i);
    }
}
```

activity_second.xml

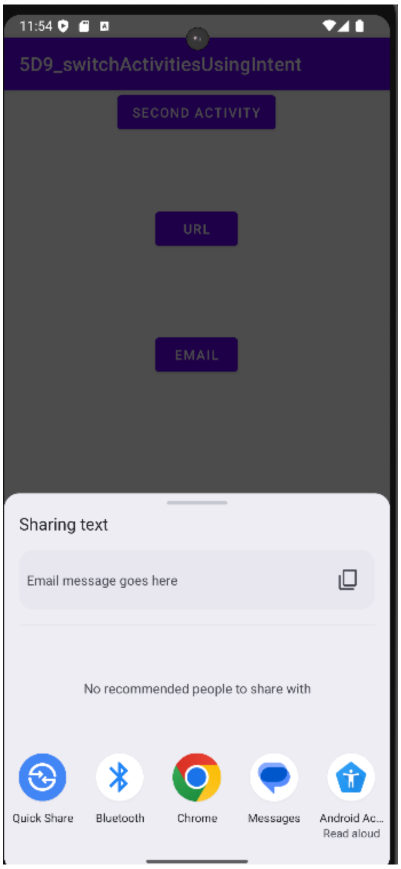
```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"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

    <Button
        android:id="@+id/button3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="show_main"
        android:text="REDIRECT TO MAIN ACTIVITY"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

OUTPUT:





Main Activity Show second activity clicked when we clicked on URL



Sharing text

Email message goes here

No recommended people to share with

When send Email is clicked

RESULT: Mobile application to switch from one activity to another using Intent is built successfully.

Experiment: 6 Date: 14-02-2025

Build mobile application to demonstrate Dynamic Fragments

AIM: To Build mobile application to demonstrate Dynamic Fragments

EXPERIMENTAL REQUIREMENTS: PC with Android Studio and Internet connection

THEORY:

Fragments in Android

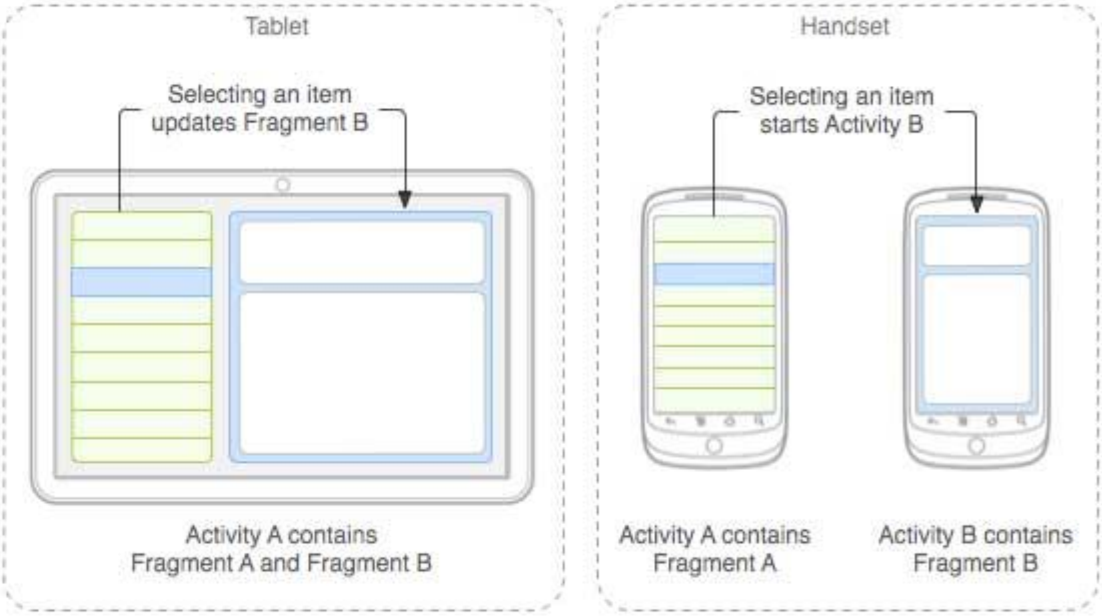
A Fragment represents a reusable portion of an app's UI. It defines and manages its own layout, has its own lifecycle, and can handle its own input events. However, fragments cannot exist independently—they must be hosted by an Activity or another fragment. The fragment's view hierarchy attaches to the host's view hierarchy.

Benefits of Using Fragments

1. **Modular UI Design:** Dividing UI into fragments makes it easier to manage and modify the activity's appearance at runtime.
2. **Better UI Adaptability:** Fragments are ideal for creating UIs that can change dynamically, such as adapting to different screen sizes or orientations.
3. **Improved Code Organization:** Using fragments keeps the UI code more organized, making it easier to maintain and scale.

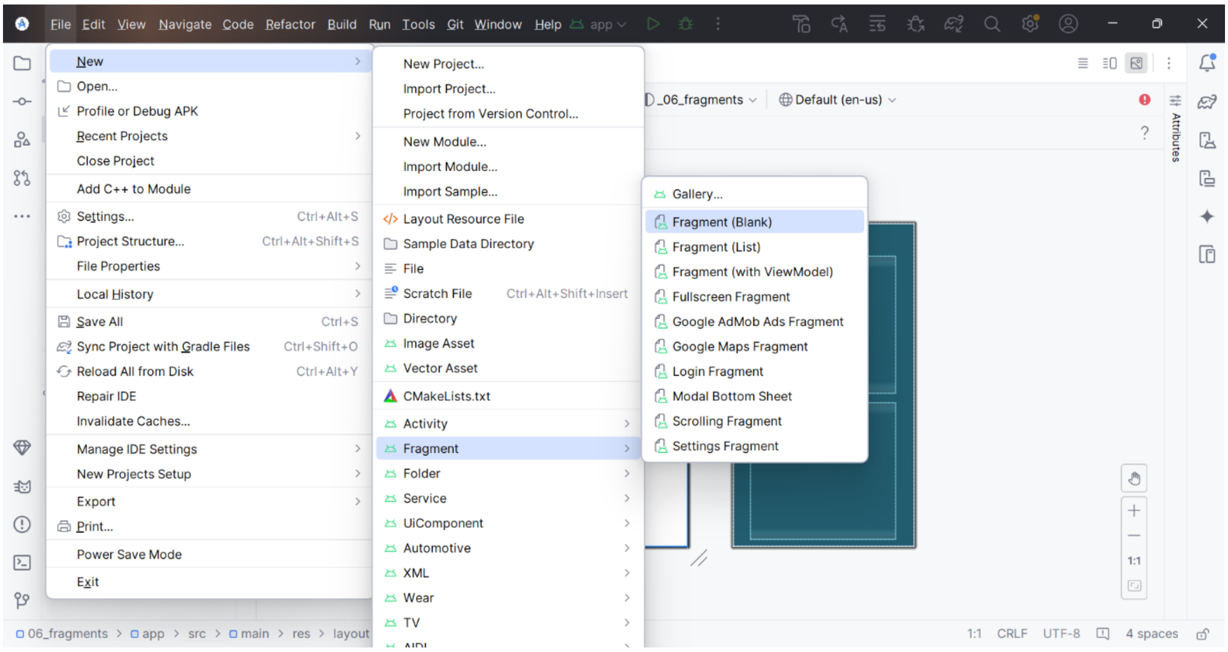
Activity vs. Fragment

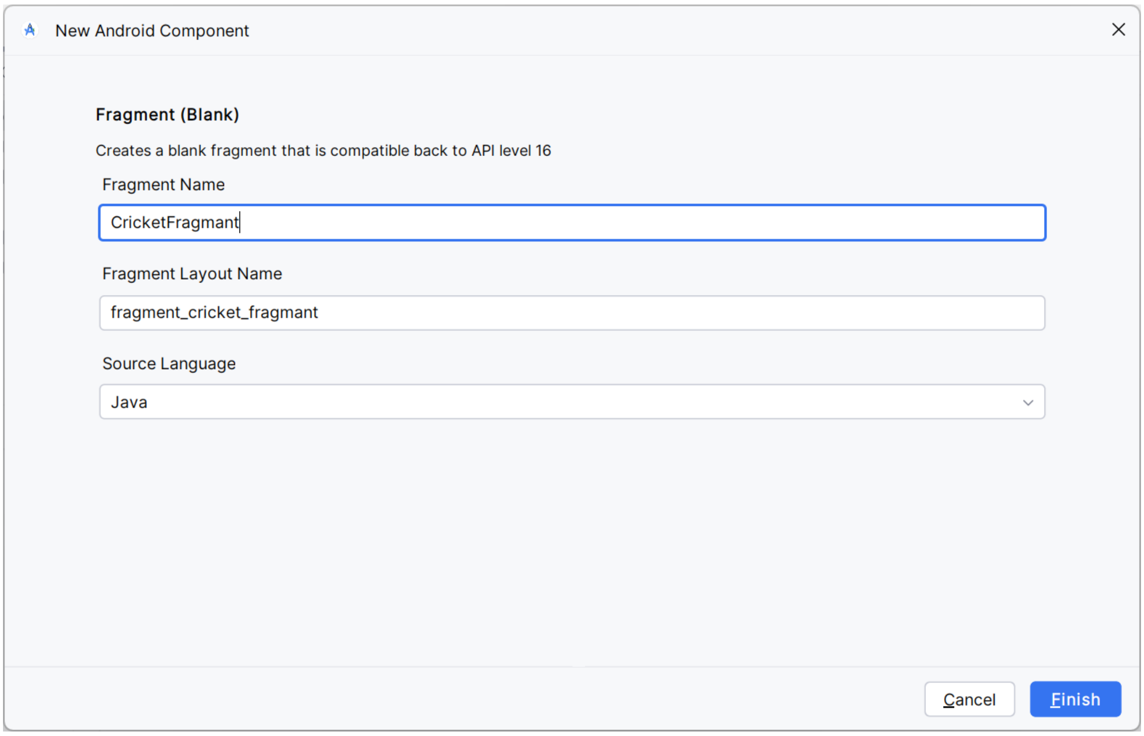
- **Activity**: The best place to manage global UI elements, such as a **navigation drawer**.
- **Fragment**: Best suited for defining and managing the UI of a single screen or a portion of a screen.


Tablet
Selecting an item updates Fragment B
Activity A contains Fragment A and Fragment B
Handset
Selecting an item starts Activity B
Activity A contains Fragment A
Activity B contains Fragment B

PROCEDURE:

1. Open Android Studio and start a new project.
2. Select "No Views Activity" and click Next.
3. Enter project details and click Finish.
4. To create a new activity, right-click on App > New > Activity > Empty Views Activity, name it MainActivity, check the Launcher Activity option, and click Finish.
5. Write code in MainActivity.java to implement intent.
6. To create fragments, right-click on App > New > Fragment > Blank Fragment, name it, and click Finish.




New Android Component

Fragment (Blank)

Creates a blank fragment that is compatible back to API level 16

Fragment Name: CricketFragmant

Fragment Layout Name: fragment_cricket_fragmant

Source Language: Java

7. Run the application using the emulator.

PROGRAMS:

MainActivity.java

```
package com.example.a22501a05g7_exp6;
import android.os.Bundle;
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 androidx.fragment.app.FragmentManager;
import androidx.fragment.app.FragmentResultListener;
import androidx.fragment.app.FragmentTransaction;
import com.example.a22501a05g7_exp6.R;
public class MainActivity extends AppCompatActivity {
    sportsFragment sf;
    FragmentManager fm;
    FragmentTransaction ft;
    BadmintonFragment bf;
    CricketFragment cf;
    FootBallFragment ff;
    HockeyFragment hf;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        sf = new sportsFragment();
        fm = getSupportFragmentManager();
        ft = fm.beginTransaction();
        sf = new sportsFragment();
        ft.add(R.id.ll1,sf);
        ft.commit();
        fm.setFragmentResultListener("si", this, new FragmentResultListener() {
            @Override
            public void onFragmentResult(@NonNull String requestKey, @NonNull Bundle result) {
                int option = result.getInt("si");
                ft = fm.beginTransaction();
                switch (option) {
```

```
                case 0:
                    ft.replace(R.id.ll2, new BadmintonFragment(), String.valueOf(bf));
                    break;
                case 1:
                    ft.replace(R.id.ll2, new HockeyFragment(), String.valueOf(hf));
                    break;
                case 2:
                    ft.replace(R.id.ll2, new FootBallFragment(), String.valueOf(ff));
                    break;
                case 3:
                    ft.replace(R.id.ll2, new CricketFragment(), String.valueOf(cf));
                    break;
            }
            ft.commit();
        }
    });
  }
}
```

activity_main.java

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

  <LinearLayout
    android:id="@+id/ll1"
    android:layout_width="330dp"
    android:layout_height="310dp"
    android:orientation="vertical"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.493"
```

```
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.175" />

    <LinearLayout
        android:id="@+id/ll2"
        android:layout_width="330dp"
        android:layout_height="310dp"
        android:orientation="vertical"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.493"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.961" />

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

BadmintionFragment.java

```
package com.example.a22501a05g7_exp6;

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_badmintion.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/imageView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/badminton" />
</FrameLayout>
```

CricketFragment.java

```
package com.example.a22501a05g7_exp6;

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) {
        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/imageView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/cricket" />
</FrameLayout>
```

FootballFragment.java

```
package com.example.a22501a05g7_exp6;

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_foot_ball, container, false);
    }
}
```

fragment_foot_ball.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/imageView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/football" />
</FrameLayout>
```

HockeyFragment.java

```
package com.example.a22501a05g7_exp6;
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=".CricketFragment">
    <ImageView
        android:id="@+id/imageView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/hockey" />
</FrameLayout>
```

SportsFragment.java

```
package com.example.a22501a05g7_exp6;

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.ArrayAdapter;
import android.widget.ListView;

import java.util.List;

public class sportsFragment extends Fragment implements AdapterView.OnItemClickListener {

    public sportsFragment() {}

    ListView lv;

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

    @Override
    public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState) {
        super.onViewCreated(view, savedInstanceState);
        lv = (ListView) view.findViewById(R.id.lvsports);
        String[] sportsList = getResources().getStringArray(R.array.sports_list);
        ArrayAdapter<String> adapter = new ArrayAdapter<>(requireContext(),
android.R.layout.simple_list_item_1, sportsList);
        lv.setAdapter(adapter);
        lv.setOnItemClickListener(this);
    }
    @Override
    public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {
        Bundle b = new Bundle();
        b.putInt("si", i);
        getParentFragmentManager().setFragmentResult("si", b);
    }
}
```

fragment_sports.xml

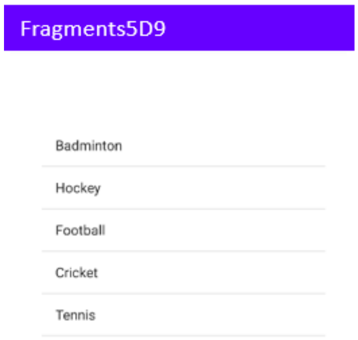
```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".sportsFragment">
    <!-- TODO: Update blank fragment layout -->
    <ListView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/lvsports"
        android:entries="@array/sports_list" />

</FrameLayout>
```

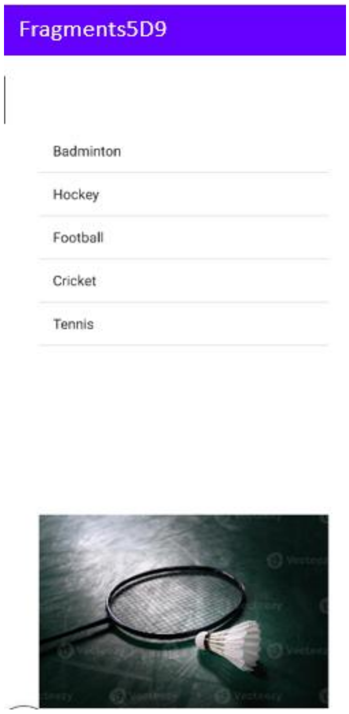
strings.xml

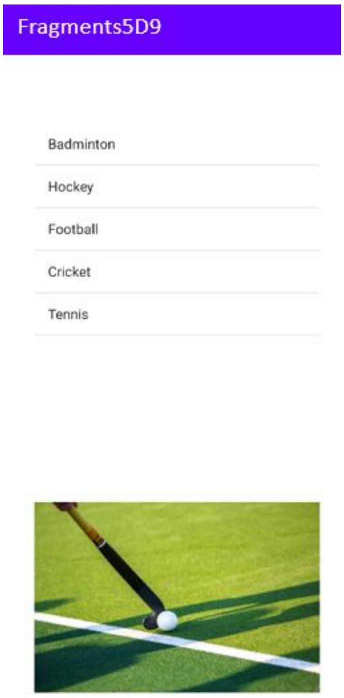
```
<resources>
    <string name="app_name">22501a05g7_exp6</string>
    <!-- TODO: Remove or change this placeholder text -->
    <string name="hello_blank_fragment">Hello blank fragment</string>
    <array name="sports_list">
        <item>Badminton</item>
        <item>Hockey</item>
        <item>Football</item>
        <item>Cricket</item>
        <item>Tennis</item>
    </array>
```

</resources>

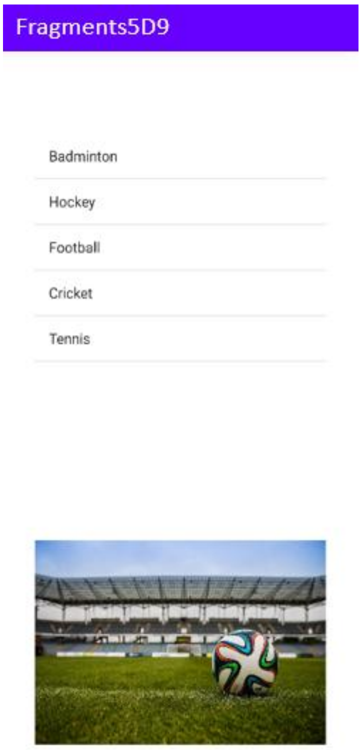
OUTPUT:

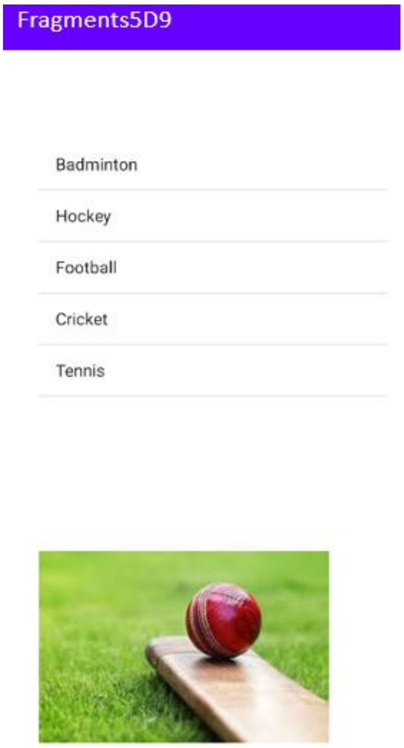






Main Activity After Clicking on Badminton After Clicking on Hockey





After Clicking on Football After Clicking on Cricket

RESULT: Mobile application based on the concept fragment built successfully.

Experiment: 7.1 **Date: 21-2-2025**

Build mobile application serverless database SQLite Database

AIM: To build mobile application serverless database SQLite Database.

EXPERIMENTAL REQUIREMENTS: PC with Android Studio and Internet connection

THEORY:

SQLite Database

SQLite is a lightweight **serverless** database engine integrated into Android. It allows developers to manage structured data efficiently without requiring an internet connection. The **android.database.sqlite** package provides built-in classes and methods for managing SQLite databases in Android applications.

SQLiteOpenHelper

- Helps in creating, upgrading, and managing the database.
- It extends the SQLiteOpenHelper class and overrides two key methods:
  - onCreate(SQLiteDatabase db): Creates tables when the database is first created.
  - onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion): Updates the database structure when required.

SQLiteDatabase

- Provides methods to perform **CRUD operations** (Create, Read, Update, Delete) on the database.
- Common methods:
  - insert() – Adds a new record.
  - query() – Retrieves data from the database.
  - update() – Modifies existing records.
  - delete() – Removes data from the database.
  - execSQL() – Executes raw SQL commands.

Roles of Each File

1. **MainActivity.java**
   - Handles user input, button clicks, and interactions.
   - Calls database methods to insert, update, delete, and retrieve student records.
   - Displays data in a RecyclerView or ListView.
2. **Student.java (Model Class)**
   - Represents a Student entity with attributes like id, name, and grade.
   - Contains getter and setter methods for data handling.
3. **StudentGradeDb.java (Database Helper Class)**
   - Extends SQLiteOpenHelper to create and manage the database.
   - Implements CRUD (Create, Read, Update, Delete) methods.
   - Uses onCreate() to define the database schema.
   - Uses onUpgrade() to update the database when needed.

PROCEDURE:

1. Open Android Studio and create a New Project with Empty Activity.
2. Enter project name, package name, select Java or Kotlin, and click Finish.
3. In the java/com.example.app folder, create a new Java Class named Student.java.
4. Define attributes like id, name, and grade, along with getter and setter methods.
5. In the java/com.example.app folder, create another Java Class named StudentGradeDb.java.
6. Extend SQLiteOpenHelper in StudentGradeDb.java, override onCreate() to define the database schema, and implement CRUD methods (insert, update, delete, query).
7. Open MainActivity.java, design the UI in activity_main.xml with input fields and buttons for database operations.
8. In MainActivity.java, create an instance of StudentGradeDb, call database methods for adding, updating, deleting, and fetching student records.
9. Run the application on an Emulator or Physical Device and test database functionality.

PROGRAMS:

MainActivity.java

```
package com.example.sqlite_5d9;

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

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

public class MainActivity extends AppCompatActivity {
    private EditText edtxt_roll, edtxt_name, edtxt_avg, edtxt_grade;
    private DatabaseHelper dbHelper;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });

        // Initialize UI components
        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);

        Button btnInsert = findViewById(R.id.btn_insert);
        Button btnGet = findViewById(R.id.btn_get);
        Button btnUpdate = findViewById(R.id.btn_update);
        Button btnDelete = findViewById(R.id.btn_delete);

        // Initialize SQLite helper
        dbHelper = new DatabaseHelper(this);

        // Set button click listeners
        btnInsert.setOnClickListener(v -> insertStudent());
        btnGet.setOnClickListener(v -> getStudent());
        btnUpdate.setOnClickListener(v -> updateStudent());
        btnDelete.setOnClickListener(v -> deleteStudent());
    }

    private void insertStudent() {
        String roll = edtxt_roll.getText().toString().trim();
        String name = edtxt_name.getText().toString().trim();
        String avg = edtxt_avg.getText().toString().trim();
        String grade = edtxt_grade.getText().toString().trim();

        if (roll.isEmpty() || name.isEmpty() || avg.isEmpty() || grade.isEmpty()) {
            Toast.makeText(this, "Please fill all fields", Toast.LENGTH_SHORT).show();
            return;
        }

        Student student = new Student(roll, name, avg, grade);
        if (dbHelper.insertStudent(student)) {
            Toast.makeText(this, "Student Inserted", Toast.LENGTH_SHORT).show();
            clearFields();
        } else {
            Toast.makeText(this, "Insertion Failed", Toast.LENGTH_SHORT).show();
        }
    }

    private void getStudent() {
        String roll = edtxt_roll.getText().toString().trim();
        if (roll.isEmpty()) {
            Toast.makeText(this, "Please enter roll number", Toast.LENGTH_SHORT).show();
            return;
        }

        Student student = dbHelper.getStudent(roll);
        if (student != null) {
            edtxt_name.setText(student.getName());
            edtxt_avg.setText(student.getAvg());
            edtxt_grade.setText(student.getGrade());
            Toast.makeText(this, "Student Found: " + student.getName(), Toast.LENGTH_SHORT).show();
        } else {
            Toast.makeText(this, "No student found with roll: " + roll, Toast.LENGTH_SHORT).show();
        }
    }

    private void updateStudent() {
        String roll = edtxt_roll.getText().toString().trim();
        String name = edtxt_name.getText().toString().trim();
        String avg = edtxt_avg.getText().toString().trim();
        String grade = edtxt_grade.getText().toString().trim();

        if (roll.isEmpty() || name.isEmpty() || avg.isEmpty() || grade.isEmpty()) {
            Toast.makeText(this, "Please fill all fields", Toast.LENGTH_SHORT).show();
            return;
        }

        Student student = new Student(roll, name, avg, grade);
        if (dbHelper.updateStudent(student)) {
            Toast.makeText(this, "Student Updated", Toast.LENGTH_SHORT).show();
            clearFields();
        } else {
            Toast.makeText(this, "Update Failed", Toast.LENGTH_SHORT).show();
        }
    }

    private void deleteStudent() {
        String roll = edtxt_roll.getText().toString().trim();
        if (roll.isEmpty()) {
            Toast.makeText(this, "Please enter roll number", Toast.LENGTH_SHORT).show();
            return;
        }

        if (dbHelper.deleteStudent(roll)) {
            Toast.makeText(this, "Student Deleted", Toast.LENGTH_SHORT).show();
            clearFields();
        } else {
            Toast.makeText(this, "Deletion Failed", Toast.LENGTH_SHORT).show();
        }
    }

    private void clearFields() {
        edtxt_roll.setText("");
        edtxt_name.setText("");
        edtxt_avg.setText("");
        edtxt_grade.setText("");
    }
}
```

Student.java

```
package com.example.sqlite_5d9;
public class Student {
    private String roll;
    private String name;
    private String avg;
    private String grade;

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

    // Getters and Setters
    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; }
}
```

DatabaseHelper.java

```
package com.example.sqlite_5d9;

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

public class DatabaseHelper extends SQLiteOpenHelper {
    private static final String DATABASE_NAME = "StudentDB";
    private static final int DATABASE_VERSION = 1;
    private static final String TABLE_NAME = "students";
    private static final String COL_ROLL = "roll";
    private static final String COL_NAME = "name";
    private static final String COL_AVG = "avg";
    private static final String COL_GRADE = "grade";

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

    @Override
    public void onCreate(SQLiteDatabase db) {
       String CREATE_TABLE = "CREATE TABLE " + TABLE_NAME + " (" +
              COL_ROLL + " TEXT PRIMARY KEY, " +
              COL_NAME + " TEXT, " +
              COL_AVG + " TEXT, " +
              COL_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 a student
    public boolean insertStudent(Student student) {
       SQLiteDatabase db = this.getWritableDatabase();
       ContentValues values = new ContentValues();
       values.put(COL_ROLL, student.getRoll());
       values.put(COL_NAME, student.getName());
       values.put(COL_AVG, student.getAvg());
       values.put(COL_GRADE, student.getGrade());
       long result = db.insert(TABLE_NAME, null, values);
       db.close();
       return result != -1; // Return true if insert successful
```

```
    }

    // Get a student by roll number
    public Student getStudent(String roll) {
        SQLiteDatabase db = this.getReadableDatabase();
        Cursor cursor = db.query(TABLE_NAME,
                new String[]{COL_ROLL, COL_NAME, COL_AVG, COL_GRADE},
                COL_ROLL + "=?",
                new String[]{roll}, null, null, null);

        if (cursor != null && cursor.moveToFirst()) {
            Student student = new Student(
                    cursor.getString(0), // roll
                    cursor.getString(1), // name
                    cursor.getString(2), // avg
                    cursor.getString(3)  // grade
            );
            cursor.close();
            return student;
        }
        if (cursor != null) cursor.close();
        return null;
    }

    // Update a student
    public boolean updateStudent(Student student) {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues values = new ContentValues();
        values.put(COL_NAME, student.getName());
        values.put(COL_AVG, student.getAvg());
        values.put(COL_GRADE, student.getGrade());
        int result = db.update(TABLE_NAME, values, COL_ROLL + "=?",
                new String[]{student.getRoll()});
        db.close();
        return result > 0; // Return true if update successful
    }
    // Delete a student
    public boolean deleteStudent(String roll) {
        SQLiteDatabase db = this.getWritableDatabase();
        int result = db.delete(TABLE_NAME, COL_ROLL + "=?",
                new String[]{roll});
        db.close();
        return result > 0; // Return true if delete successful
    }
}
```

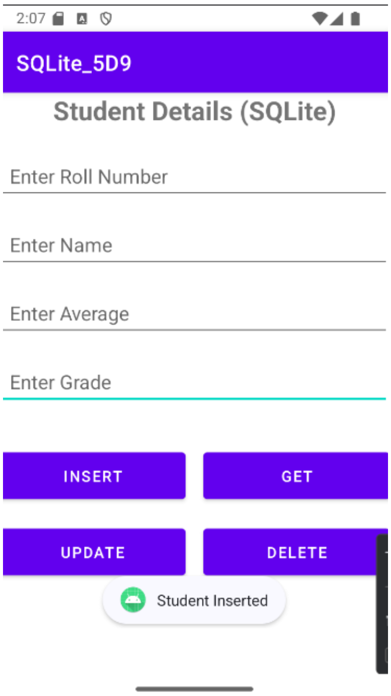
activity_main.xml

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

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Student Details (SQLite)"
        android:textAlignment="center"
        android:textSize="24sp"
        android:textStyle="bold"
        android:layout_marginBottom="20dp"/>

    <EditText
        android:id="@+id/edtxt_roll"
        android:layout_width="match_parent"
        android:layout_height="48dp"
        android:hint="Enter Roll Number"
        android:textSize="18sp"
        android:padding="10dp"
        android:inputType="text"
        android:layout_marginBottom="15dp"/>
```

```
<EditText
    android:id="@+id/edtxt_name"
    android:layout_width="match_parent"
    android:layout_height="48dp"
    android:hint="Enter Name"
    android:textSize="18sp"
    android:padding="10dp"
    android:inputType="text"
    android:layout_marginBottom="15dp"/>

<EditText
    android:id="@+id/edtxt_avg"
    android:layout_width="match_parent"
    android:layout_height="48dp"
    android:hint="Enter Average"
    android:textSize="18sp"
    android:padding="10dp"
    android:inputType="numberDecimal"
    android:layout_marginBottom="15dp"/>

<EditText
    android:id="@+id/edtxt_grade"
    android:layout_width="match_parent"
    android:layout_height="48dp"
    android:hint="Enter Grade"
    android:textSize="18sp"
    android:padding="10dp"
    android:inputType="text"
    android:layout_marginBottom="15dp"/>

<!-- First row: Insert and Get -->
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:gravity="center"
    android:layout_marginTop="20dp">
    <Button
        android:id="@+id/btn_insert"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:minHeight="48dp"
        android:padding="12dp"
        android:text="Insert"
        android:layout_marginEnd="8dp"/>

    <Button
        android:id="@+id/btn_get"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:minHeight="48dp"
        android:padding="12dp"
        android:text="Get"
        android:layout_marginStart="8dp"/>

</LinearLayout>

<!-- Second row: Update and Delete -->
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:gravity="center"
    android:layout_marginTop="15dp">

    <Button
        android:id="@+id/btn_update"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:minHeight="48dp"
```

```
            android:padding="12dp"
            android:text="Update"
            android:layout_marginEnd="8dp"/>

        <Button
            android:id="@+id/btn_delete"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:minHeight="48dp"
            android:padding="12dp"
            android:text="Delete"
            android:layout_marginStart="8dp"/>

    </LinearLayout>
</LinearLayout>
```

OUTPUT:



2:07

SQLite_5D9

Student Details (SQLite)

Enter Roll Number

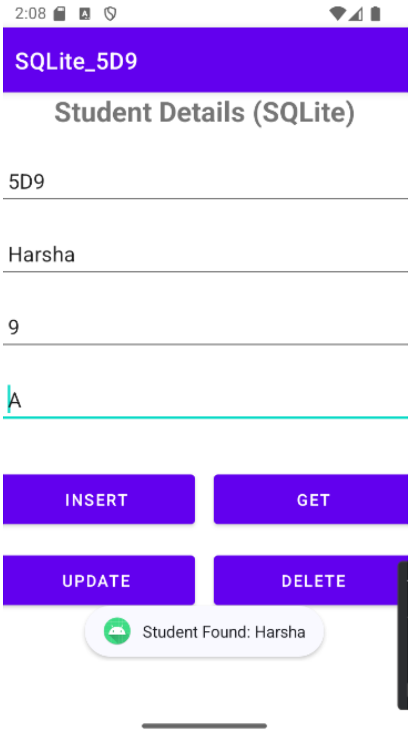
Enter Name

Enter Average

Enter Grade

Student Inserted

Insert operation



2:08

SQLite_5D9

Student Details (SQLite)

5D9

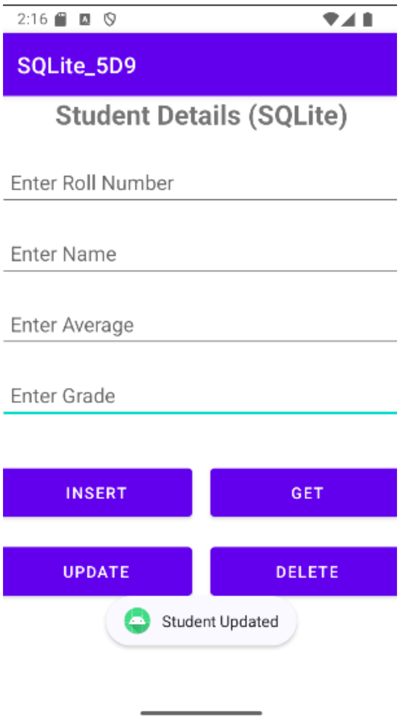
Harsha

9

A

Student Found: Harsha

Select operation



2:16

SQLite_5D9

Student Details (SQLite)

Enter Roll Number

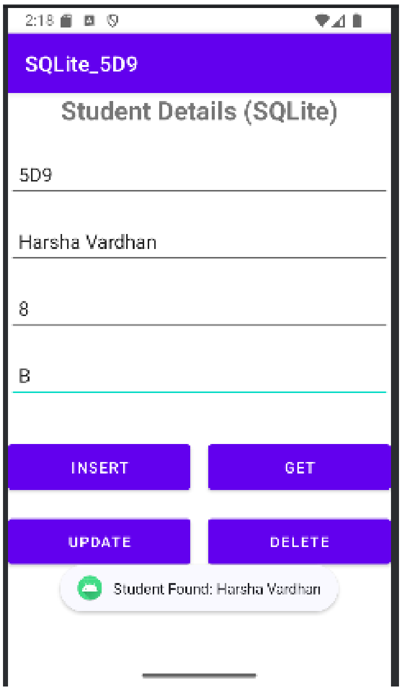
Enter Name

Enter Average

Enter Grade

Student Updated

Update operation



Student Details (SQLite)

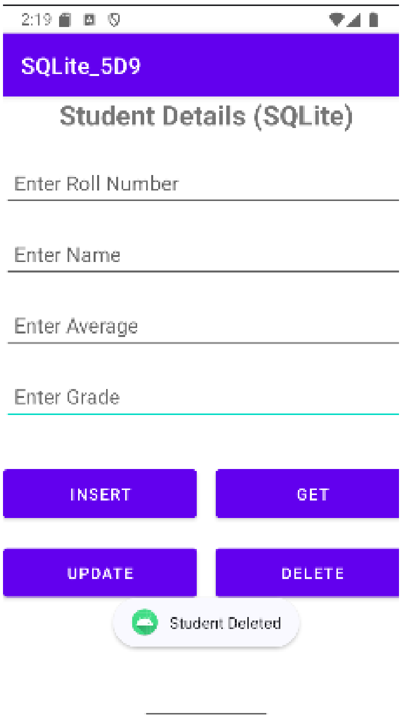
5D9

Harsha Vardhan

8

B

Student Found: Harsha Vardhan



Student Details (SQLite)

Enter Roll Number

Enter Name

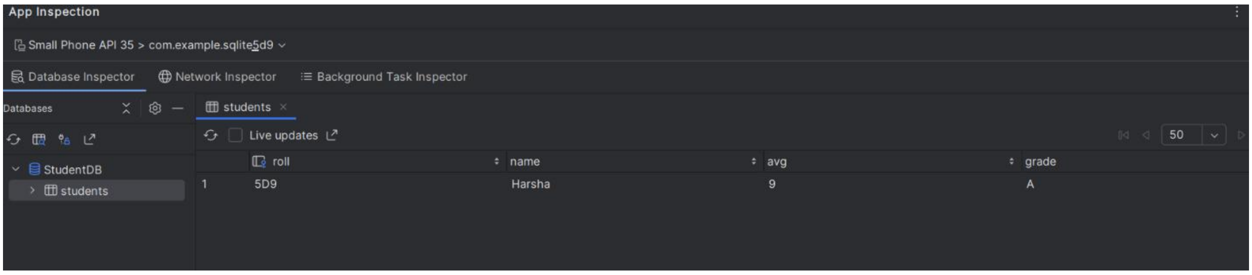
Enter Average

Enter Grade

Student Deleted

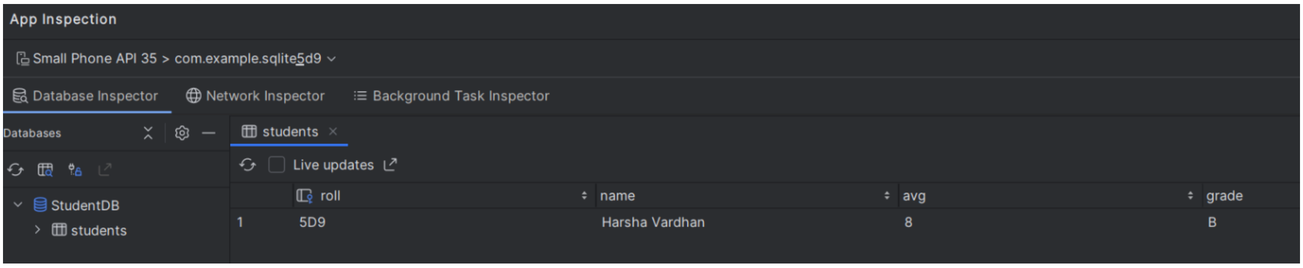
Select after updation Deletion operation

Database After Insertion:



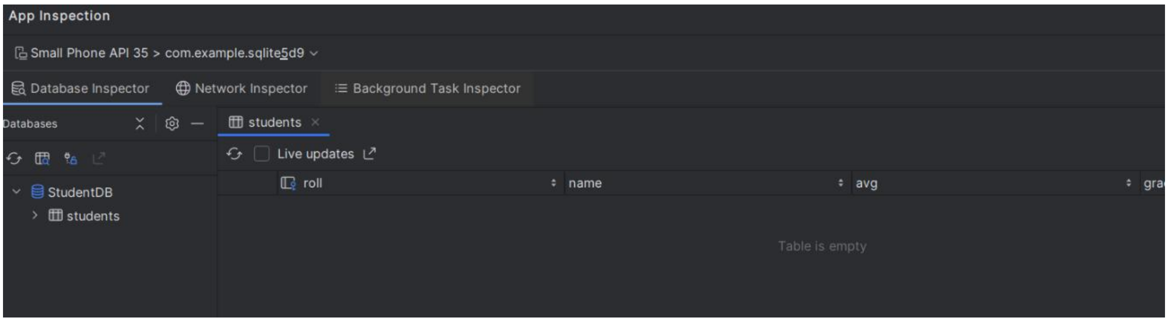
	roll	name	avg	grade
1	5D9	Harsha	9	A

Database After Updation:



	roll	name	avg	grade
1	5D9	Harsha Vardhan	8	B

Database After Deletion:



roll	name	avg	gra

Table is empty

RESULT: Mobile application with serverless SQLite Database is built successfully.

Experiment: 7.2 Date: 28-02-2025

Build mobile application cloud based database Firebase

Firebase

Firebase is a Backend-as-a-Service (BaaS) platform developed by Google that provides tools and infrastructure to build web and mobile applications. It offers services such as database management, authentication, cloud storage, and real-time data synchronization without the need for managing backend servers.

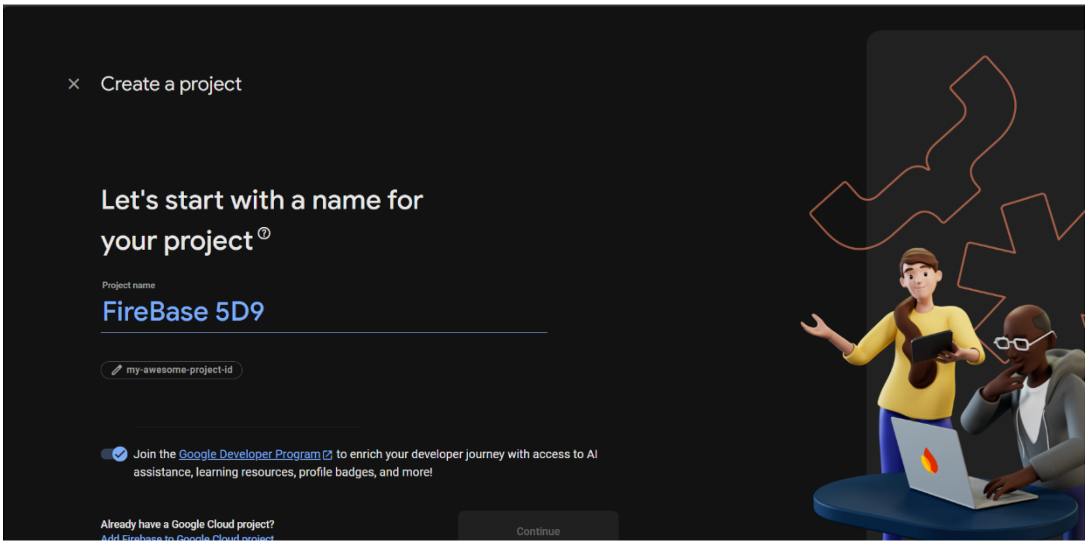
Firebase Realtime Database

- A cloud-hosted NoSQL database that allows real-time data synchronization.
- Stores data in JSON format and updates automatically across connected clients.
- Works offline, syncing changes when the device reconnects to the internet.

PROCEDURE:

Steps to Connect Firebase Using Firebase Assistant in Android Studio

1. Create a New Android Project in Android Studio (Empty Activity, Java/Kotlin, API 21+).
2. Open Firebase Assistant: Click Tools → Firebase in Android Studio.
3. Connect to Firebase:
   a. In Firebase Assistant, go to Realtime Database → Set up Firebase Realtime Database → Connect to Firebase.
   b. This opens the Firebase Console in your browser.
4. Create a Firebase Project:
   a. Click Create a Project.
   b. Enter a Project Name
   c. Click Continue, then enable Google Analytics (optional).
   d. Click Create Project and wait for it to complete.

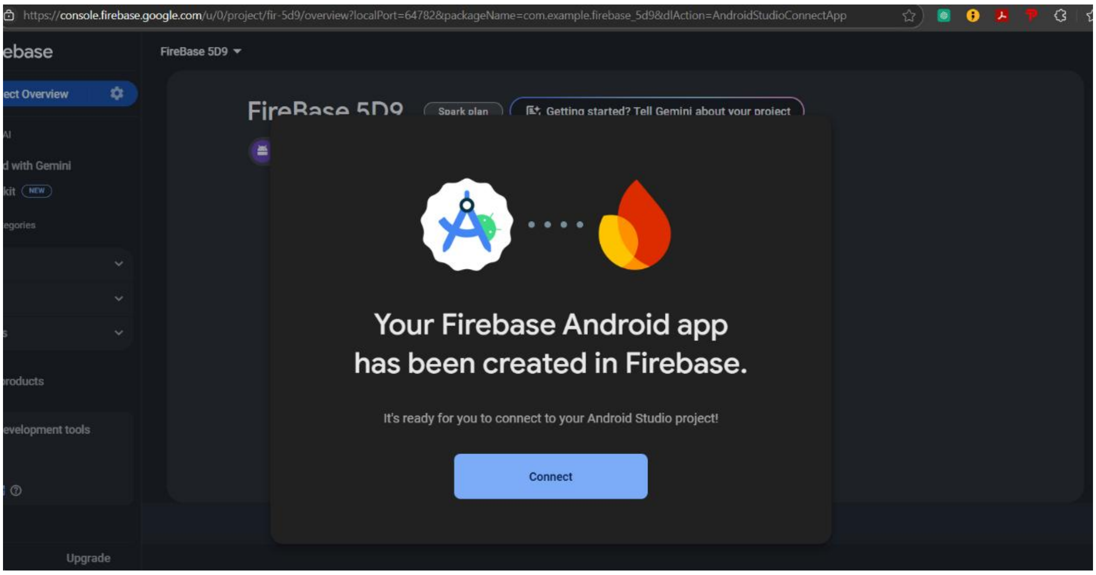


Let's start with a name for your project

Project name: FireBase 5D9

☑ Join the Google Developer Program to enrich your developer journey with access to AI assistance, learning resources, profile badges, and more!

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

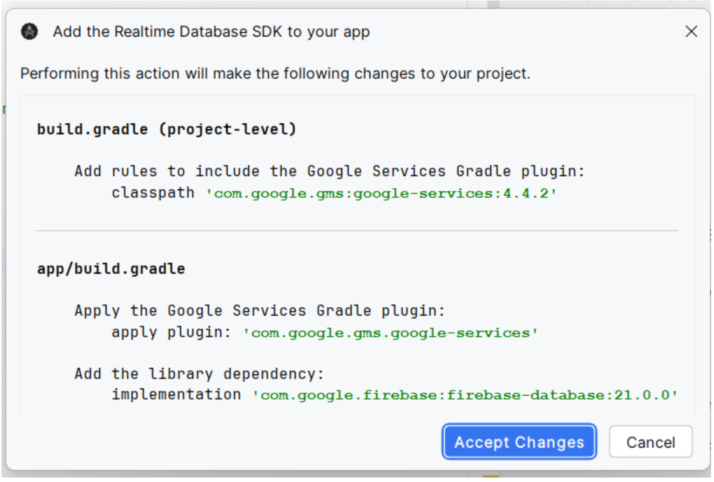

https://console.firebase.google.com/u/0/project/fir-5d9/overview?localPort=64782&packageName=com.example.firebase_5d9&dlAction=AndroidStudioConnectApp
ebase
FireBase 5D9
ect Overview
d with Gemini
kit NEW
egories
roducts
evelopment tools
Upgrade
FireBase 5D9
Spark plan
Getting started? Tell Gemini about your project

Your Firebase Android app has been created in Firebase.

It's ready for you to connect to your Android Studio project!

Connect

5. Add Firebase SDK:
   a. Go back to Android Studio and click "Add Realtime Database to Your App".
   b. Android Studio will automatically add Firebase dependencies in Gradle files.
   c. Click Sync Now to apply changes.

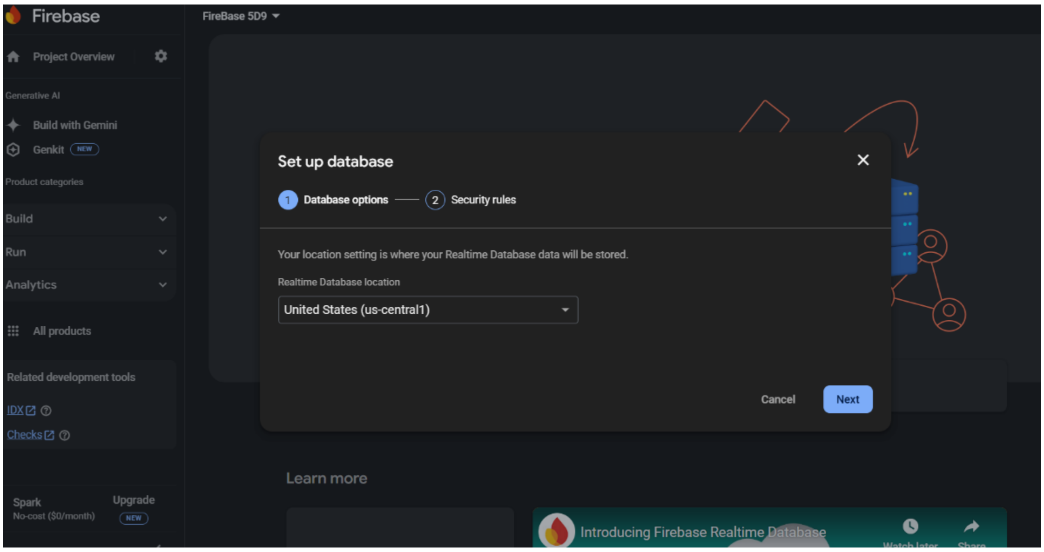


Add the Realtime Database SDK to your app

Performing this action will make the following changes to your project.

```
build.gradle (project-level)

    Add rules to include the Google Services Gradle plugin:
        classpath 'com.google.gms:google-services:4.4.2'

app/build.gradle

    Apply the Google Services Gradle plugin:
        apply plugin: 'com.google.gms.google-services'

    Add the library dependency:
        implementation 'com.google.firebase:firebase-database:21.0.0'
```

6. Enable Realtime Database:
   a. Open Firebase Console → Realtime Database.
   b. Click Create Database.
   c. Choose Database Location (e.g., us-central1, asia-south1).
   d. Select Start in Test Mode (for development, allows unrestricted read/write access).
   e. Click Enable



Set up database

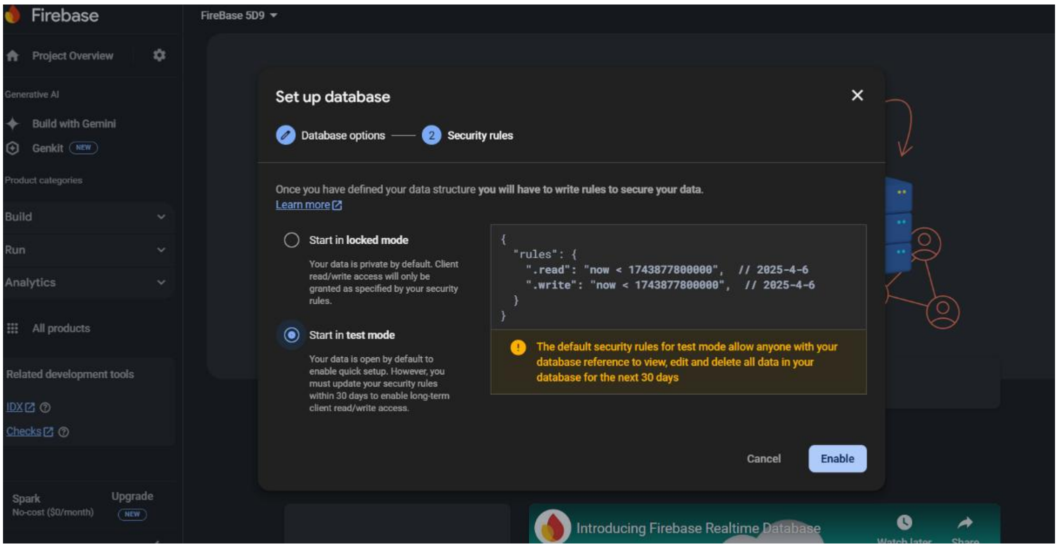
1 Database options —— 2 Security rules

Your location setting is where your Realtime Database data will be stored.

Realtime Database location

United States (us-central1)

Cancel Next



Set up database

Database options —— 2 Security rules

Once you have defined your data structure **you will have to write rules to secure your data.**
Learn more

○ Start in **locked mode**

Your data is private by default. Client read/write access will only be granted as specified by your security rules.

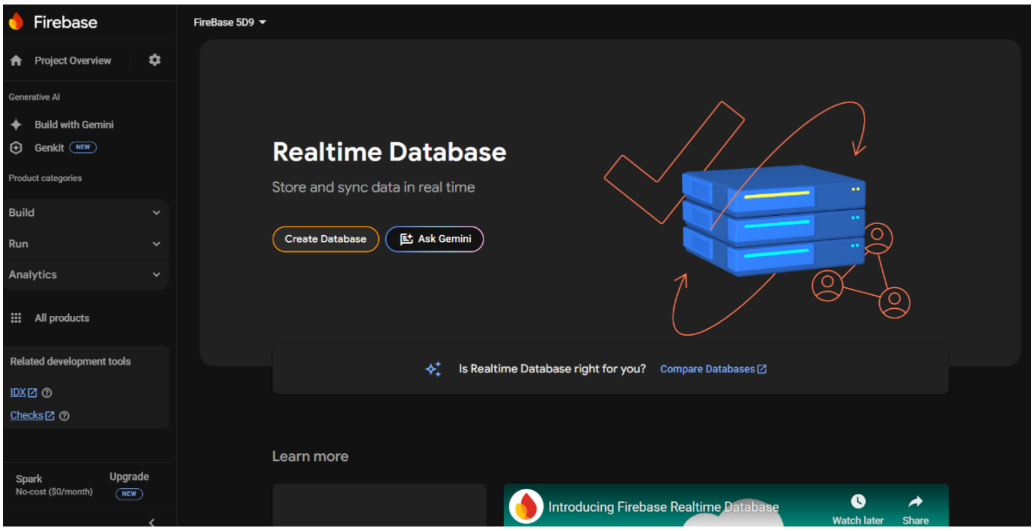
● Start in **test mode**

Your data is open by default to enable quick setup. However, you must update your security rules within 30 days to enable long-term client read/write access.

```
{
  "rules": {
    ".read": "now < 1743877800000",  // 2025-4-6
    ".write": "now < 1743877800000",  // 2025-4-6
  }
}
```

The default security rules for test mode allow anyone with your database reference to view, edit and delete all data in your database for the next 30 days

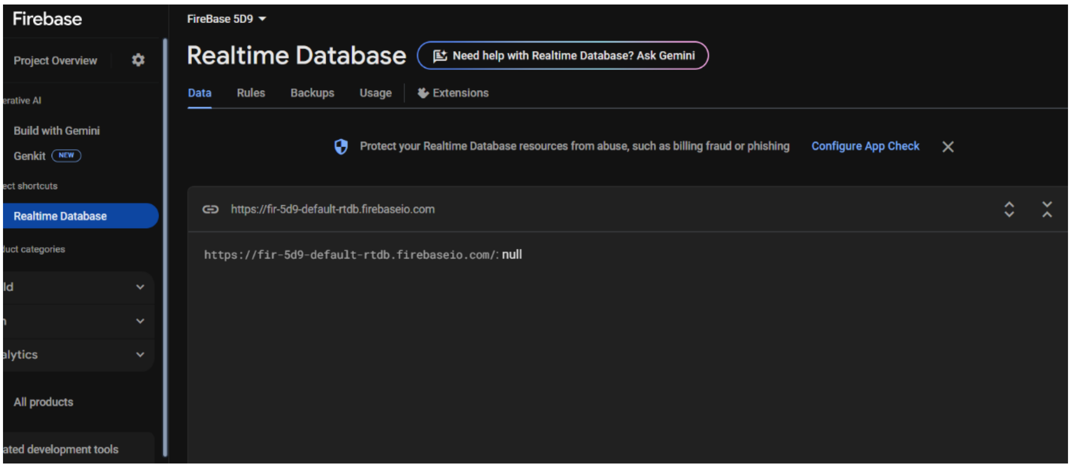
Cancel Enable



Realtime Database

Store and sync data in real time

Is Realtime Database right for you?



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

https://fir-5d9-default-rtdb.firebaseio.com

https://fir-5d9-default-rtdb.firebaseio.com/: null

PROGRAMS:

MainActivity.java

```
package com.example.firebase_5d9;

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

import androidx.activity.EdgeToEdge;
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.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;

public class MainActivity extends AppCompatActivity {
    EditText edtxt_roll, edtxt_name, edtxt_avg, edtxt_grade;
    FirebaseDatabase fdb;
    DatabaseReference studentRef; // Added DatabaseReference for better structure

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });

        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();
        studentRef = fdb.getReference("students"); // Changed to plural "students" for better naming
    }

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

        studentRef.child(roll).setValue(s)
                .addOnSuccessListener(unused ->
                        Toast.makeText(MainActivity.this, "Insertion Successful",
Toast.LENGTH_LONG).show())
                .addOnFailureListener(e ->
                        Toast.makeText(MainActivity.this, "Insertion Failure: " + e.getMessage(),
Toast.LENGTH_LONG).show());
    }

    // Added CRUD operations
    public void getStudent(View v) {
        String roll = edtxt_roll.getText().toString();
        studentRef.child(roll).get()
```

```
                .addOnSuccessListener(dataSnapshot -> {
                    if (dataSnapshot.exists()) {
                        Student s = dataSnapshot.getValue(Student.class);
                        if (s != null) {
                            edtxt_name.setText(s.getName());
                            edtxt_avg.setText(s.getAvg());
                            edtxt_grade.setText(s.getGrade());
                            Toast.makeText(MainActivity.this, "Student Found: " + s.getName(),
Toast.LENGTH_LONG).show();
                        }
                    } else {
                        Toast.makeText(MainActivity.this, "No student found with Roll No: " + roll,
Toast.LENGTH_LONG).show();
                    }
                })
                .addOnFailureListener(e ->
                        Toast.makeText(MainActivity.this, "Retrieval Failure: " + e.getMessage(),
Toast.LENGTH_LONG).show());
    }

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

        studentRef.child(roll).setValue(s)
                .addOnSuccessListener(unused ->
                        Toast.makeText(MainActivity.this, "Update Successful", Toast.LENGTH_LONG).show())
                .addOnFailureListener(e ->
                        Toast.makeText(MainActivity.this, "Update Failure: " + e.getMessage(),
Toast.LENGTH_LONG).show());
    }

    public void deleteStudent(View v) {
        String roll = edtxt_roll.getText().toString();
        studentRef.child(roll).removeValue()
```

```
                .addOnSuccessListener(unused ->
                        Toast.makeText(MainActivity.this, "Deletion Successful",
Toast.LENGTH_LONG).show())
                .addOnFailureListener(e ->
                        Toast.makeText(MainActivity.this, "Deletion Failure: " + e.getMessage(),
Toast.LENGTH_LONG).show());
    }
}
```

student.java

```
package com.example.firebase_5d9;

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

    // Added default constructor required by Firebase
    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 String getName() { return name; }
    public String getAvg() { return avg; }
    public String getGrade() { return grade; }

    // Added setters for updating
    public void setRoll(String roll) { this.roll = roll; }
    public void setName(String name) { this.name = name; }
    public void setAvg(String avg) { this.avg = avg; }
    public void setGrade(String grade) { this.grade = grade; }
}
```

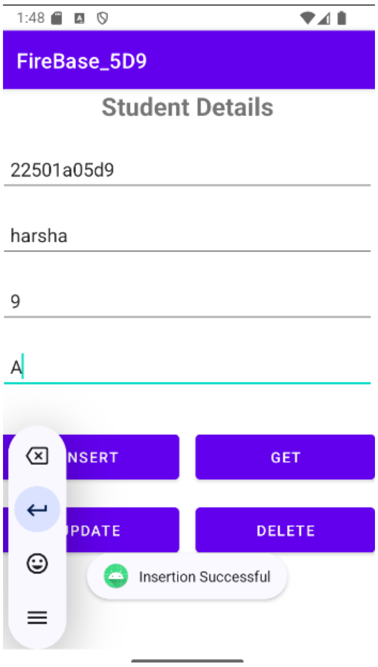
activity_main.xml

```
<?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:id="@+id/main"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:orientation="vertical"
  tools:context=".MainActivity"
  android:padding="16dp">

  <TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Student Details"
    android:textAlignment="center"
    android:textSize="24sp"
    android:textStyle="bold"
    android:layout_marginBottom="20dp"/>

  <EditText
    android:id="@+id/edtxt_roll"
    android:layout_width="match_parent"
    android:layout_height="48dp"
    android:hint="Enter Roll Number"
    android:textSize="18sp"
    android:padding="10dp"
    android:layout_marginBottom="15dp"/>

  <EditText
    android:id="@+id/edtxt_name"
    android:layout_width="match_parent"
    android:layout_height="48dp"
    android:hint="Enter Name"
    android:textSize="18sp"
    android:padding="10dp"
    android:layout_marginBottom="15dp"/>
```

```
<EditText
    android:id="@+id/edtxt_avg"
    android:layout_width="match_parent"
    android:layout_height="48dp"
    android:hint="Enter Average"
    android:textSize="18sp"
    android:padding="10dp"
    android:layout_marginBottom="15dp"/>

<EditText
    android:id="@+id/edtxt_grade"
    android:layout_width="match_parent"
    android:layout_height="48dp"
    android:hint="Enter Grade"
    android:textSize="18sp"
    android:padding="10dp"
    android:layout_marginBottom="15dp"/>

<!-- First row of buttons: Insert and Get -->
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:gravity="center"
    android:layout_marginTop="20dp">

    <Button
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:minHeight="48dp"
        android:onClick="insertStudent"
        android:padding="12dp"
        android:text="Insert"
        android:layout_marginEnd="8dp"/>

    <Button
```

```
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:minHeight="48dp"
            android:onClick="getStudent"
            android:padding="12dp"
            android:text="Get"
            android:layout_marginStart="8dp"/>

    </LinearLayout>

    <!-- Second row of buttons: Update and Delete -->
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:gravity="center"
        android:layout_marginTop="15dp">

        <Button
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:minHeight="48dp"
            android:onClick="updateStudent"
            android:padding="12dp"
            android:text="Update"
            android:layout_marginEnd="8dp"/>

        <Button
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:minHeight="48dp"
            android:onClick="deleteStudent"
            android:padding="12dp"
            android:text="Delete"
            android:layout_marginStart="8dp"/>


        </LinearLayout>
    </LinearLayout>
```

OUTPUT:



FireBase_5D9

Student Details

22501a05d9

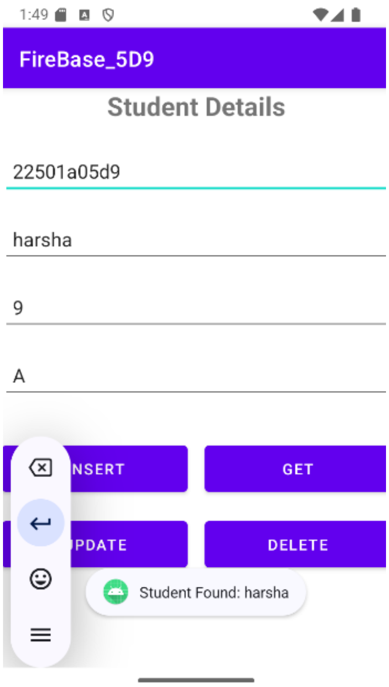
harsha

9

A

Insertion Successful

Insert operation



FireBase_5D9

Student Details

22501a05d9

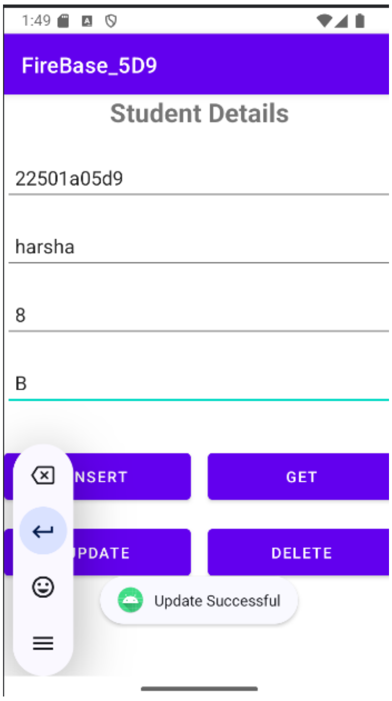
harsha

9

A

Student Found: harsha

Select operation



FireBase_5D9

Student Details

22501a05d9

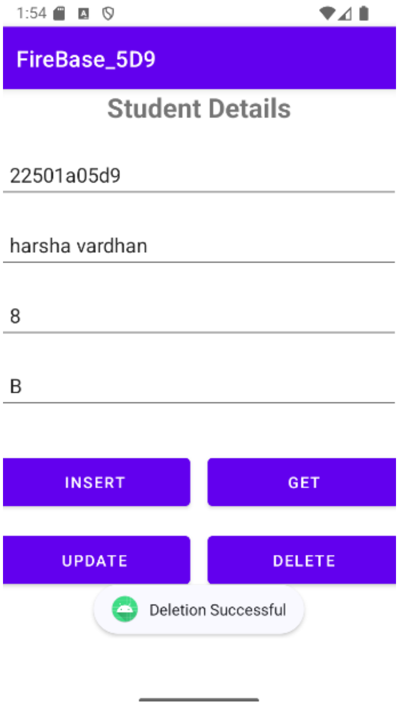
harsha

8

B

Update Successful

Update operation



FireBase_5D9

Student Details

22501a05d9

harsha vardhan

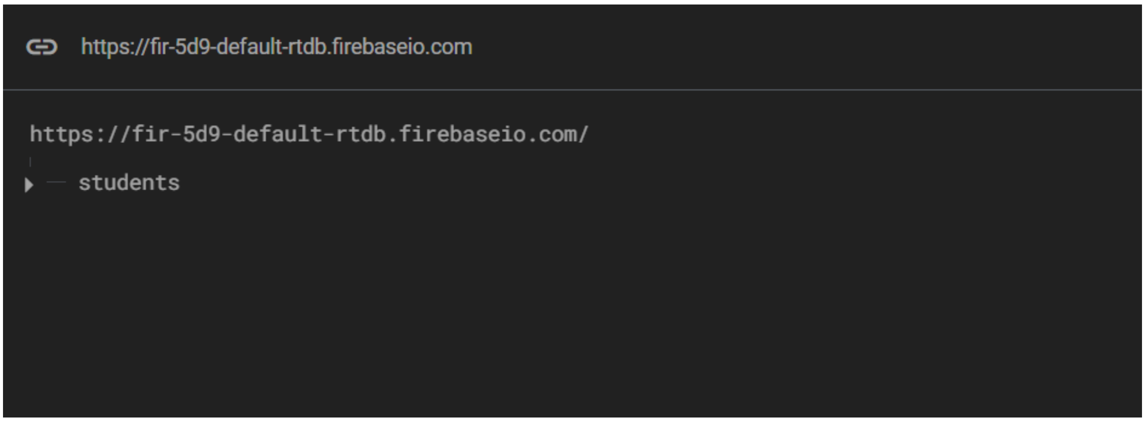
8

B

Deletion Successful

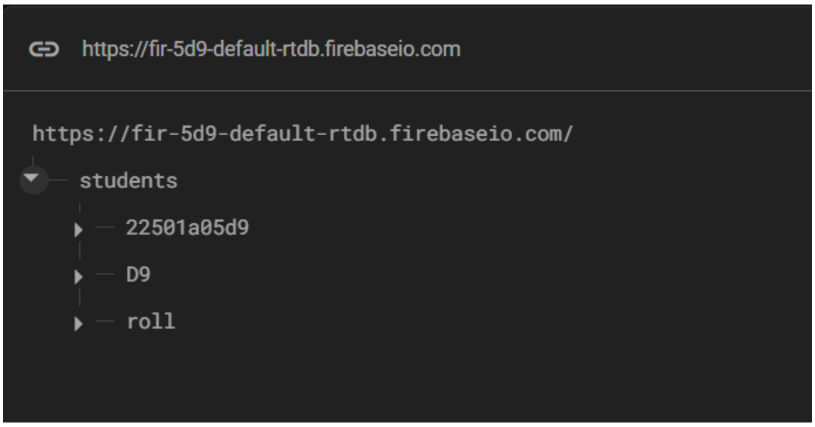
Deletion operation

In FireBase



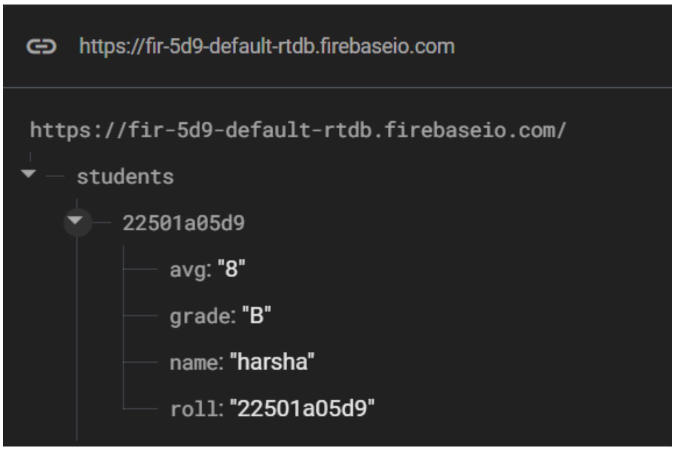
```
https://fir-5d9-default-rtdb.firebaseio.com

https://fir-5d9-default-rtdb.firebaseio.com/
▸ — students
```

Students:



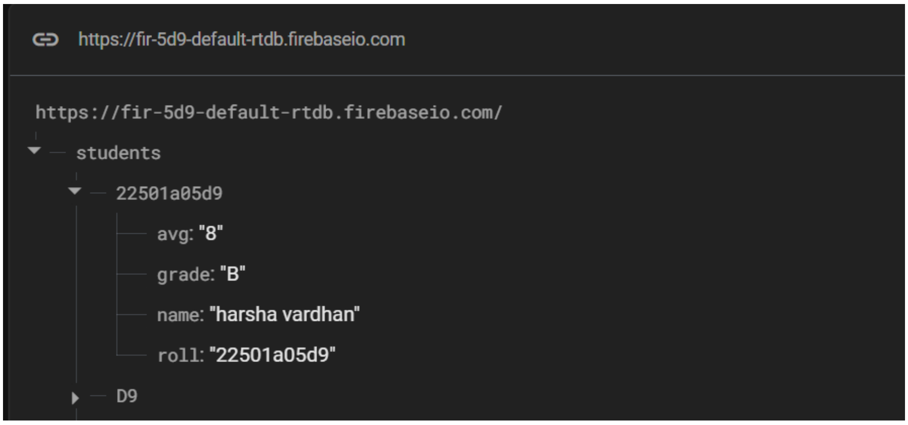
```
https://fir-5d9-default-rtdb.firebaseio.com

https://fir-5d9-default-rtdb.firebaseio.com/
▾ — students
    ▸ — 22501a05d9
    ▸ — D9
    ▸ — roll
```

Database After Insertion:



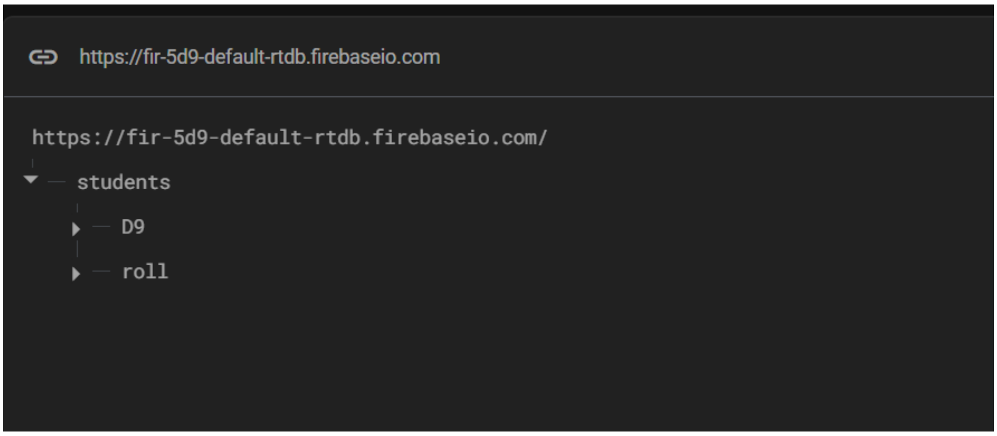
```
https://fir-5d9-default-rtdb.firebaseio.com

https://fir-5d9-default-rtdb.firebaseio.com/
▾ — students
    ▾ — 22501a05d9
          — avg: "8"
          — grade: "B"
          — name: "harsha"
          — roll: "22501a05d9"
```

Database After Updation:



```
https://fir-5d9-default-rtdb.firebaseio.com

https://fir-5d9-default-rtdb.firebaseio.com/
▾ — students
    ▾ — 22501a05d9
          — avg: "8"
          — grade: "B"
          — name: "harsha vardhan"
          — roll: "22501a05d9"
    ▸ — D9
```

Database After Deletion:



https://fir-5d9-default-rtdb.firebaseio.com

```
https://fir-5d9-default-rtdb.firebaseio.com/
▾ — students
    ▸ — D9
    ▸ — roll
```

RESULT: Mobile application with cloud based Firebase is built successfully.

Experiment: 8

Date: 21-3-2025

Build mobile application based on the Google Maps.

AIM: To build mobile application based on the google maps.

EXPERIMENTAL REQUIREMENTS: PC with Android Studio and Internet connection

THEORY:

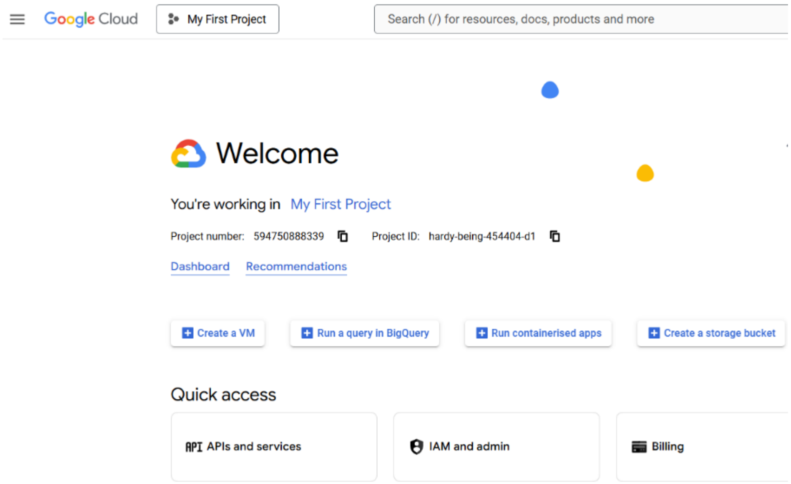
Google Maps is a widely used service that provides mapping and location-based functionalities. Developers can integrate Google Maps into mobile applications using the Google Maps API. This theory explains the fundamental concepts behind Google Maps API integration, including its purpose, benefits, and significance in mobile application development.

Google Maps API is a set of web services that allows developers to incorporate mapping features into applications. It enables functionalities such as location search, real-time navigation, traffic updates, and user location tracking. By utilizing Google Maps API, developers can enhance their applications with geographic and location-based features.

To integrate Google Maps into a mobile application, developers must obtain an API key, configure it within the application, and use relevant SDKs or libraries provided by Google. The integration process involves embedding a map view, customizing map features, and implementing additional functionalities such as markers, route planning, and geolocation tracking.

PROCEDURE:

1. Open Android Studio and click on "New Project" in the Welcome screen. Select "No Activity" under the "Phone and Tablet" category and click "Next."
2. Enter the project details, set the language to Java, and select a Minimum SDK version of API Level 21 or higher. Click "Finish" and wait for Gradle to build the project.
3. Open Google Cloud Console and create a new project.

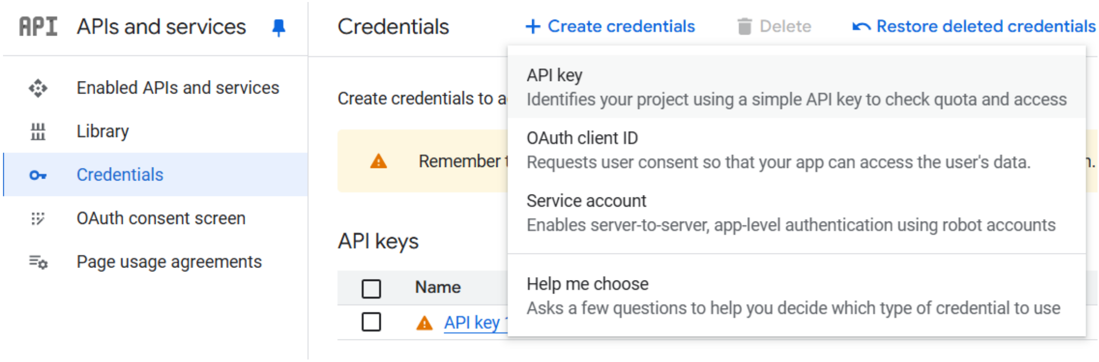


Welcome

You're working in My First Project

Project number: 594750888339 Project ID: hardy-being-454404-d1

4. Navigate to "APIs & Services," enable "Google Maps SDK for Android," and go to the "Credentials" section. Click "Create Credentials" and select "API Key." Restrict the API key for security and copy it for later use.



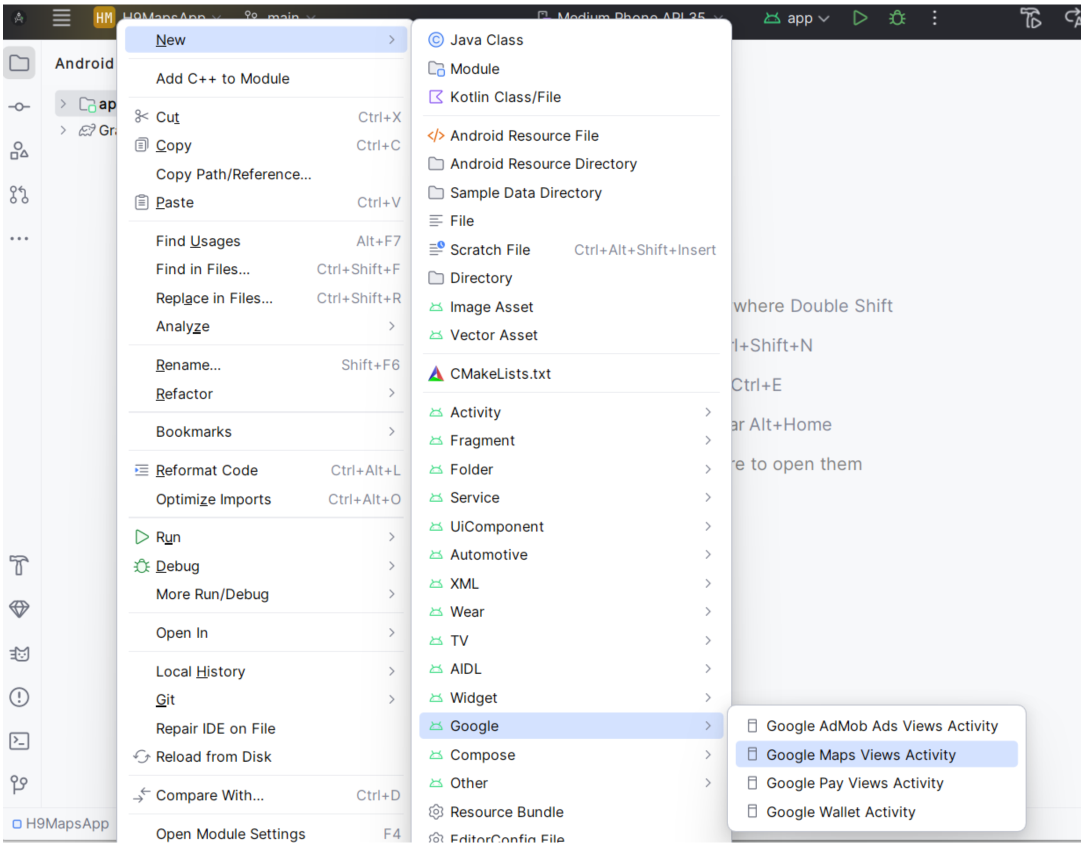
Identifies your project using a simple API key to check quota and access

Requests user consent so that your app can access the user's data.

Enables server-to-server, app-level authentication using robot accounts

Asks a few questions to help you decide which type of credential to use

5. In Android Studio, right-click the "app" folder, go to "New," select "Google," and then choose "Google Maps Views Activity." Check the "Launcher Activity" option and click "Finish."



6. In the file local.properties of the project. Add the API key in the file as MAPS_API_KEY=YOUR_API_KEY.

7. Open the AndroidManifest.xml file and locate or add the <meta-data> tag inside the <application> tag. Set android:value to ${MAPS_API_KEY} to securely reference the API key.

8. Open the module-level build.gradle file and ensure that the Google Maps SDK dependency is added as implementation("com.google.android.gms:play-services-maps:19.0.0").
9. Sync the Gradle files and run the application on an emulator or a physical device to verify that the Google Map is displayed correctly.

PROGRAMS:

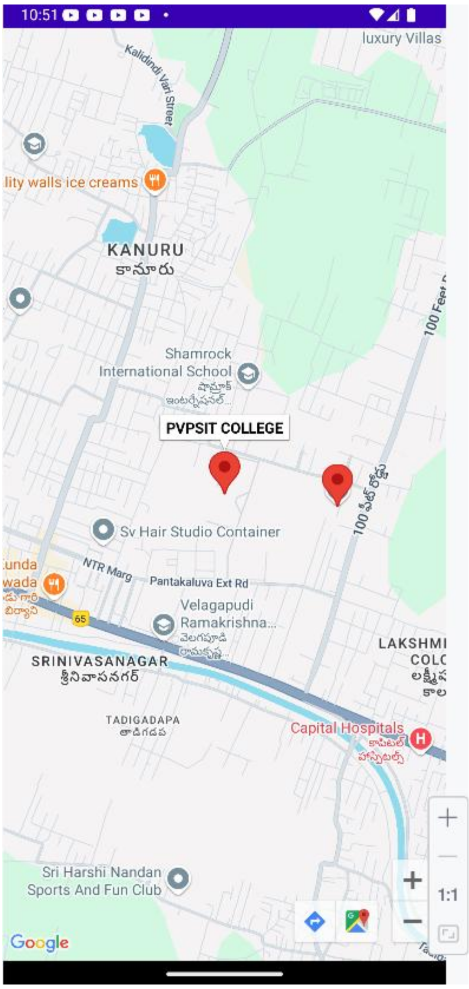
MapsActivity.java

```
package com.example.a22501a05g7_exp8;
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.a22501a05g7_exp8.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 pvp = new LatLng(16.487780, 80.693798);
        MarkerOptions mo=new MarkerOptions();
        mo.position(pvp);
        mo.title("PVPSIT COLLEGE");
        mMap.addMarker(mo);

        mMap.addMarker(new MarkerOptions().position(pvp).title("PVPSIT COLLEGE"));
        mMap.moveCamera(CameraUpdateFactory.newLatLng(pvp));
        mMap.moveCamera(CameraUpdateFactory.newLatLng(pvp));
        mMap.moveCamera(CameraUpdateFactory.zoomTo(17));

        LatLng ion = new LatLng(16.4873, 80.6983);
        mMap.addMarker(new MarkerOptions().position(ion).title("ION DIGITAL"));
        mMap.moveCamera(CameraUpdateFactory.newLatLng(ion));
        mMap.moveCamera(CameraUpdateFactory.newLatLng(ion));
        mMap.moveCamera(CameraUpdateFactory.zoomTo(17));
    }
}
```

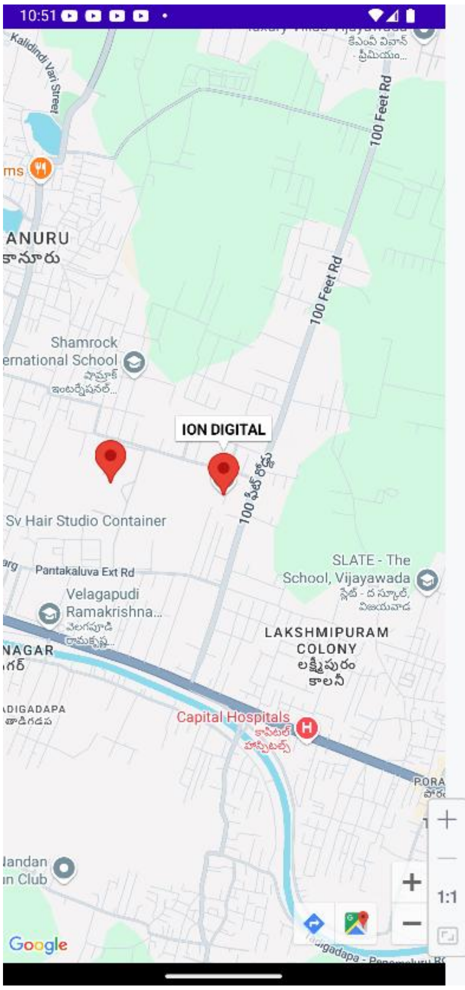
activity_maps.xml

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

OUTPUT:



When 1st location is clicked



When 2nd location is clicked

RESULT: Mobile application based on the Google maps built successfully.