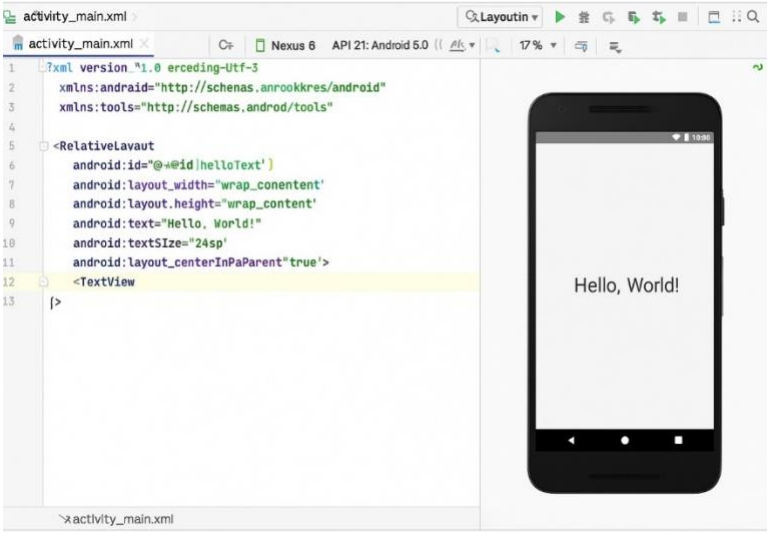
1.Creating an Android Project 1. Create an Android project with a minimum SDK of 21. Design a simple app that displays "Hello, World!" on the screen.

Code :

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

xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   tools:context=".MainActivity">

   <TextView
      android:id="@+id/helloKamlesh"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Hello, World!"
      android:textSize="24sp"
      android:layout_centerInParent="true"
/>
</RelativeLayout>
```

```
package com.example.helloworldapp;


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


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

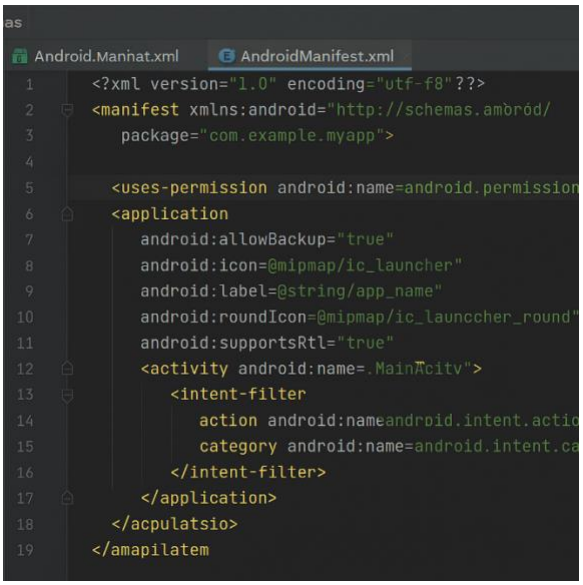


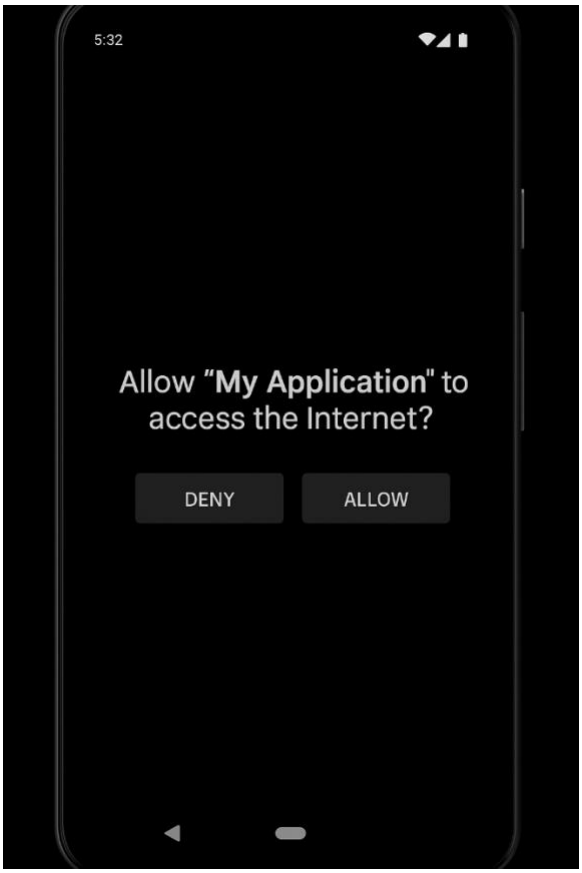
2. Project Structure 2. Identify and explain the purpose of the AndroidManifest.xml file in your project. Modify it to request INTERNET permission.

Code –

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

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

    <!-- Permission to access the internet -->
    <uses-permission
android:name="android.permission.INTERNET" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"

android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
android:theme="@style/Theme.MyApp">

        <activity
android:name=".MainActivity">
            <intent-filter>
                <action
android:name="android.intent.action.MAIN" />
                <category
android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

    </application>

</manifest>
```


as
Android.Manıat.xml
AndroidManifest.xml
<?xml version="1.0" encoding="utf-8"??>
<manifest xmlns:android="http://schemas.amòród/
package="com.example.myapp">
<uses-permission android:name=android.permissior
<application
android:allowBackup="true"
android:icon=@mipmap/ic_launcher"
android:label=@string/app_name"
android:roundIcon=@mipmap/ic_launccher_round"
android:supportsRtl="true"
<activity android:name=.MainAcitv">
<intent-filter
action android:nameandroid.intent.actic
category android:name=android.intent.ca
</intent-filter>
</application>
</acpulatsio>
</amapilatem


5:32
Allow "My Application" to access the Internet?
DENY
ALLOW

3. Activity and Activity Life Cycle

3. Create an activity that logs lifecycle method calls (onCreate, onStart, etc.) in the Logcat.

4. Design an app with two activities and use lifecycle methods to save and restore a counter value during orientation changes.

Code :

✅ 1. MainActivity.java

```
package com.example.lifecycleapp;

import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.widget.Button;
import android.widget.TextView;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    private static final String TAG = "MainActivity";
    private static final String COUNTER_KEY = "counter_value";

    private int counter = 0;
    private TextView counterTextView;
    private Button incrementButton, nextButton;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        Log.d(TAG, "onCreate");

setContentView(R.layout.activity_main);

        counterTextView = findViewById(R.id.counterText);
        incrementButton = findViewById(R.id.incrementButton);
        nextButton = findViewById(R.id.nextButton);

        if (savedInstanceState != null) {
            counter = savedInstanceState.getInt(COUNTER_KEY);
        }
```

```
        updateCounterText();

        incrementButton.setOnClickListener(v -
> {
            counter++;
            updateCounterText();
        });

        nextButton.setOnClickListener(v -> {
            Intent intent = new
Intent(MainActivity.this,
SecondActivity.class);
            startActivity(intent);
        });
    }
    private void updateCounterText() {
        counterTextView.setText("Counter: " +
counter);
    }

    @Override
    public void onStart() {
        super.onStart();
        Log.d(TAG, "onStart");
    }

    @Override
    public void onResume() {
        super.onResume();
        Log.d(TAG, "onResume");
    }@Override
    public void onPause() {
        super.onPause();
        Log.d(TAG, "onPause");
    }

    @Override
    public void onStop() {
        super.onStop();
        Log.d(TAG, "onStop");
    }

    @Override
    public void onDestroy() {
        super.onDestroy();
        Log.d(TAG, "onDestroy");
    }

    @Override
    public void
onSaveInstanceState(@NonNull Bundle
outState) {
```

```
        outState.putInt(COUNTER_KEY, counter);
        super.onSaveInstanceState(outState);
    }
```

✅ 2. SecondActivity.java

```
package com.example.lifecycleapp;

import android.os.Bundle;

import android.util.Log;

import androidx.appcompat.app.AppCompatActivity;

public class SecondActivity extends AppCompatActivity {


    private static final String TAG = "SecondActivity";


    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);


setContentView(R.layout.activity_second);

        Log.d(TAG, "onCreate");

    }


    @Override

protected void onStart() {

    super.onStart();

    Log.d(TAG, "onStart");

}

@Override

protected void onResume() {

    super.onResume();

    Log.d(TAG, "onResume");

}

@Override

protected void onPause() {

    super.onPause();

    Log.d(TAG, "onPause");

}


@Override

protected void onStop() {

    super.onStop();

    Log.d(TAG, "onStop");

}


@Override

protected void onDestroy() {

    super.onDestroy();

    Log.d(TAG, "onDestroy");

}
```

}✅ **3. activity_main.xml**

xml

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

    <TextView
        android:id="@+id/counterText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Counter: 0"
        android:textSize="24sp"
        android:textStyle="bold"
        android:layout_marginBottom="24dp"
/>

    <Button
        android:id="@+id/incrementButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Increment"
        android:layout_marginBottom="16dp"
/>

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

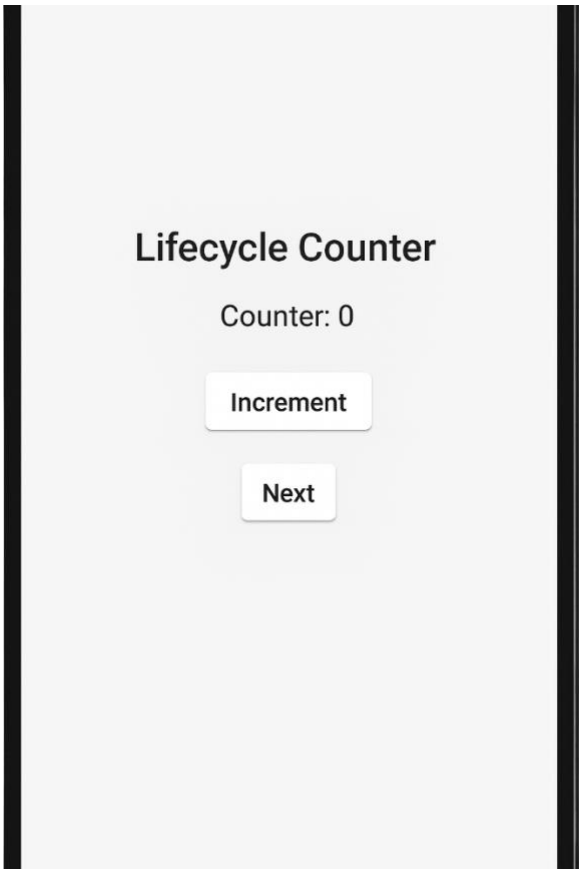
✅ 4. activity_second.xml

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

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```

```
    android:text="Welcome to Second Activity"
    android:textSize="20sp"
    android:textStyle="bold" />
</LinearLayout>
```

✅ 5. AndroidManifest.xml

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

  <application
    android:allowBackup="true"
    android:theme="@style/Theme.AppCompat.Light.NoActionBar"
    android:supportsRtl="true"
    android:label="@string/app_name">

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

</manifest>
```



4. Fragment and Fragment Life Cycle

5. Create an activity with two fragments (e.g., Fragment A and Fragment B) that communicate with each other.

6. Implement lifecycle logging for a fragment and display the logs in a TextView inside the activity

📁 MainActivity.java

```
package com.example.fragmentlifecycle;

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

public class MainActivity extends AppCompatActivity implements LogListener {

    private TextView logTextView;
    private StringBuilder logBuilder = new StringBuilder();

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

setContentView(R.layout.activity_main);

        logTextView = findViewById(R.id.logTextView);

        FragmentManager fragmentManager = getSupportFragmentManager();
        FragmentTransaction fragmentTransaction = fragmentManager.beginTransaction();

        FragmentA fragmentA = new FragmentA();
        FragmentB fragmentB = new FragmentB();

        fragmentA.setLogListener(this);
        fragmentB.setLogListener(this);
```

```
fragmentTransaction.add(R.id.fragment_container_a, fragmentA);

fragmentTransaction.add(R.id.fragment_container_b, fragmentB);
        fragmentTransaction.commit();
    }


    @Override
    public void onLogReceived(String message) {

logBuilder.append(message).append("\n");

logTextView.setText(logBuilder.toString());
    }
}
```

📁 FragmentA.java

```
package com.example.fragmentlifecycle;

import android.content.Context;
import android.os.Bundle;
import android.util.Log;
import androidx.fragment.app.Fragment;

public class FragmentA extends Fragment {
    private LogListener logListener;

    @Override
    public void onAttach(Context context) {
        super.onAttach(context);
        log("Fragment A onAttach");
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        log("Fragment A onCreate");
    }

    @Override
    public void onStart() {
        super.onStart();
        log("Fragment A onStart");
    }

    private void log(String message) {
        Log.d("FragmentA", message);
        if (logListener != null) logListener.onLogReceived(message);
    }
```

```
    public void setLogListener(LogListener
listener) {
        this.logListener = listener;
    }
}
```

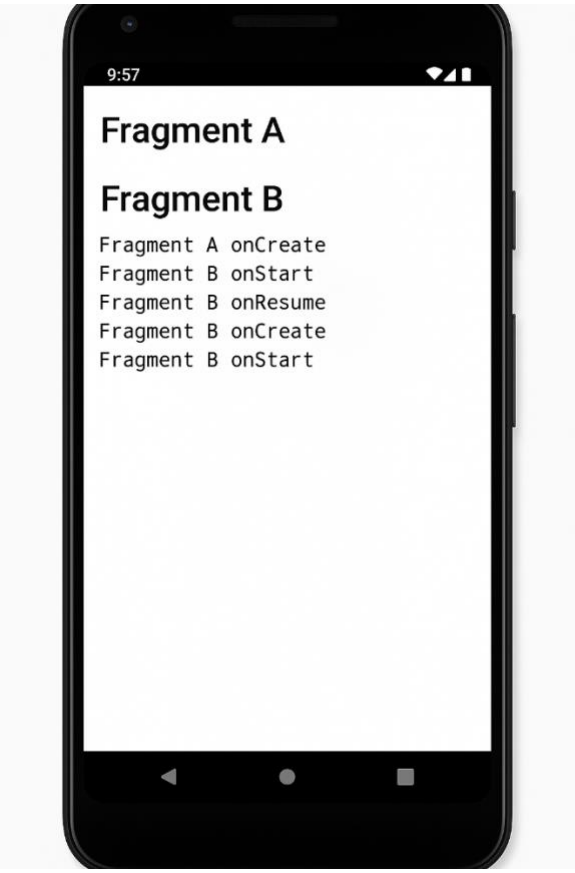
FragmentB.java

```
package com.example.fragmentlifecycle;

import android.content.Context;
import android.os.Bundle;
import android.util.Log;
import androidx.fragment.app.Fragment;

public class FragmentB extends Fragment {
    private LogListener logListener;

    @Override
    public void onAttach(Context context) {
        super.onAttach(context);
        log("Fragment B onAttach");
    }

    @Override
    public void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);
        log("Fragment B onCreate");
    }

    @Override
    public void onStart() {
        super.onStart();
        log("Fragment B onStart");
    }

    @Override
    public void onResume() {
        super.onResume();
        log("Fragment B onResume");
    }

    private void log(String message) {
        Log.d("FragmentB", message);
        if (logListener != null)
logListener.onLogReceived(message);
    }

    public void setLogListener(LogListener
listener) {
        this.logListener = listener;
```

}

}📂 activity_main.xml

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

    <FrameLayout

android:id="@+id/fragment_container_a"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1" />

    <FrameLayout

android:id="@+id/fragment_container_b"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1" />

    <TextView
        android:id="@+id/logTextView"
        android:layout_width="match_parent"
        android:layout_height="200dp"
        android:background="#EEEEEE"
        android:padding="8dp"
        android:textSize="14sp"
        android:textColor="#000000" />
</LinearLayout>
```



5. Views and View Groups

7. Design a layout using a LinearLayout to display a list of user profiles (name, email, and photo).

8. Use ConstraintLayout to create a responsive layout 10 that adapts to different screen sizes

✅ 1. **LinearLayout Version** (activity_main.xml)

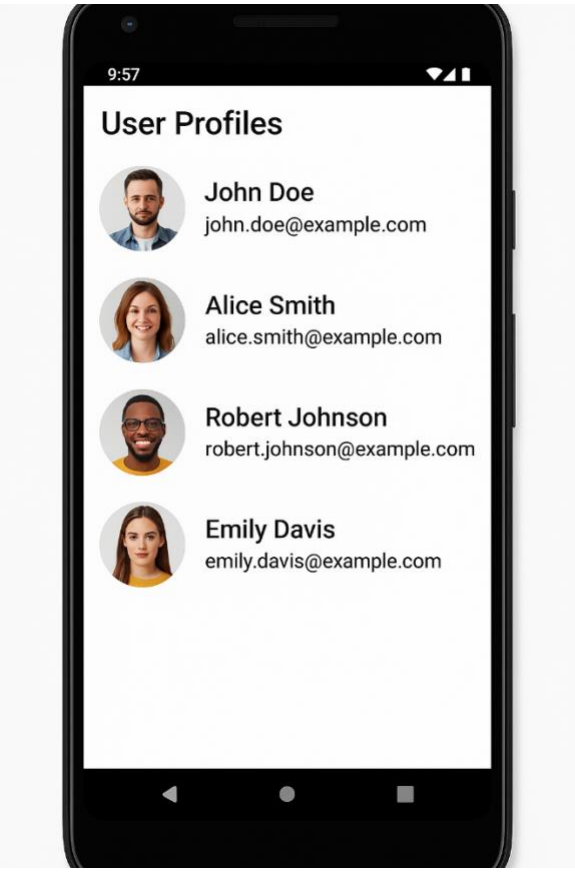
```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

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

        <TextView
            android:text="User Profiles"
            android:textSize="24sp"
            android:textStyle="bold"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginBottom="16dp"/>

        <!-- Repeating profile block -->
        <LinearLayout
            android:orientation="horizontal"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginBottom="16dp">

            <ImageView
                android:src="@drawable/user1"
                android:layout_width="64dp"
                android:layout_height="64dp"
                android:layout_marginEnd="12dp"
                android:scaleType="centerCrop"
android:background="@drawable/circle_background" />

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

```
        <TextView
            android:text="John Doe"
            android:textSize="18sp"
            android:textStyle="bold"
android:layout_width="wrap_content"
android:layout_height="wrap_content" />

        <TextView
android:text="john.doe@example.com"
android:layout_width="wrap_content"
android:layout_height="wrap_content" />
      </LinearLayout>
    </LinearLayout>

    <!-- Repeat more profile blocks as needed -->

  </LinearLayout>
</ScrollView>
```

✅ 2. **ConstraintLayout Version** (Responsive)

```
<?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"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:padding="16dp">

  <TextView
    android:id="@+id/title"
    android:text="User Profile"
    android:textSize="24sp"
    android:textStyle="bold"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintStart_toStartOf="parent" />

  <ImageView
    android:id="@+id/user_image"
    android:src="@drawable/user1"
    android:layout_width="80dp"
    android:layout_height="80dp"
    android:scaleType="centerCrop"
```

```
android:background="@drawable/circle_background"

app:layout_constraintTop_toBottomOf="@id/title"

app:layout_constraintStart_toStartOf="parent"
    android:layout_marginTop="16dp" />

  <TextView
    android:id="@+id/user_name"
    android:text="John Doe"
    android:textSize="18sp"
    android:textStyle="bold"
    android:layout_width="0dp"
    android:layout_height="wrap_content"

app:layout_constraintStart_toEndOf="@id/user_image"

app:layout_constraintTop_toTopOf="@id/user_image"

app:layout_constraintEnd_toEndOf="parent"
    android:layout_marginStart="16dp"/>

  <TextView
    android:id="@+id/user_email"
    android:text="john.doe@example.com"
    android:layout_width="0dp"
    android:layout_height="wrap_content"

app:layout_constraintTop_toBottomOf="@id/user_name"

app:layout_constraintStart_toStartOf="@id/user_name"

app:layout_constraintEnd_toEndOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```



6. Intents and Filters

9. Create an app with an Explicit Intent to navigate from one activity to another and pass a message between them.

10. Implement an Implicit Intent to open a web page using the default browser..

Code :

✅ 1. MainActivity.java

```
package com.example.intentsandfilters;

import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.widget.Button;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    Button openActivityButton, openWebButton;

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

setContentView(R.layout.activity_main);

        openActivityButton = findViewById(R.id.btnOpenActivity);
        openWebButton = findViewById(R.id.btnOpenWeb);


openActivityButton.setOnClickListener(v -> {
            Intent intent = new Intent(MainActivity.this, SecondActivity.class);
            intent.putExtra("message", "Hello from MainActivity!");
            startActivity(intent);
        });

        openWebButton.setOnClickListener(v -> {
```

```
        Intent intent = new
Intent(Intent.ACTION_VIEW,
Uri.parse("https://www.example.com"));
        startActivity(intent);
    });
  }
}
```

✅ 2. SecondActivity.java

```
package com.example.intentsandfilters;

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

import
androidx.appcompat.app.AppCompatActivit
y;

public class SecondActivity extends
AppCompatActivity {

  TextView msgTextView;

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

setContentView(R.layout.activity_second);

    msgTextView =
findViewById(R.id.txtMessage);
    String message =
getIntent().getStringExtra("message");
    msgTextView.setText(message);
  }
}
```

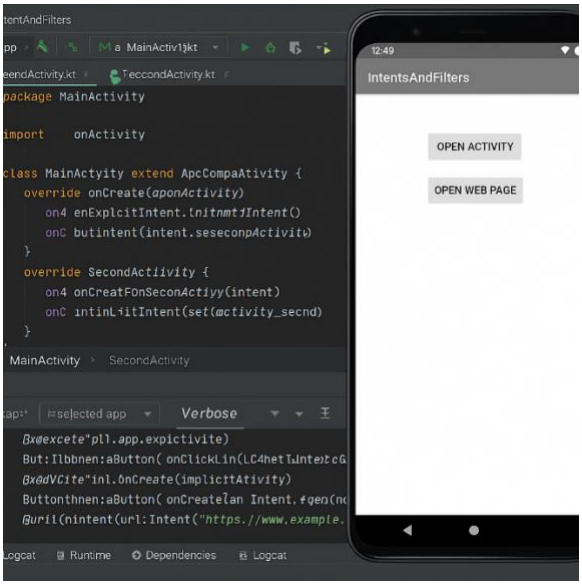
✅ 3. activity_main.xml

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

  <Button
    android:id="@+id/btnOpenActivity"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="OPEN ACTIVITY" />

  <Button
```

```
    android:id="@+id/btnOpenWeb"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="OPEN WEB PAGE"
    android:layout_marginTop="20dp"/>
</LinearLayout>
```

✅ 4. activity_second.xml

```
package com.example.intentsandfilters;

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

import
androidx.appcompat.app.AppCompatActivit
y;

public class SecondActivity extends
AppCompatActivity {

    TextView msgTextView;

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

        msgTextView =
findViewById(R.id.txtMessage);
        String message =
getIntent().getStringExtra("message");
        msgTextView.setText(message);
    }
}
```



7. Adapters

11.Create a RecyclerView to display a list of items (e.g., a to-do list). Use a custom adapter to bind the data.

12. Implement a Spinner with an ArrayAdapter to display a list of countries.

Code :

1. activity_main.xml

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

   <TextView
      android:text="Adapters"
      android:textSize="24sp"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
android:layout_marginBottom="16dp"/>

<androidx.recyclerview.widget.RecyclerView
      android:id="@+id/recyclerView"
      android:layout_width="match_parent"
android:layout_height="wrap_content"/>

   <Spinner
      android:id="@+id/spinner"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:layout_marginTop="16dp"/>

</LinearLayout>
```

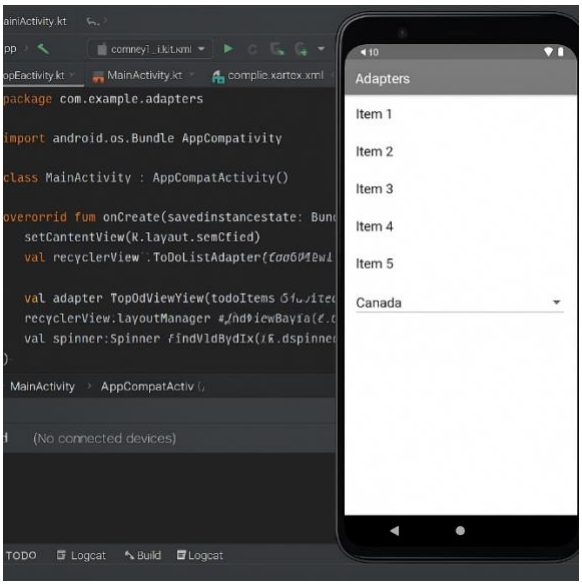
2. MainActivity.kt

```
package com.example.adapters

import android.os.Bundle
import android.widget.ArrayAdapter
import android.widget.Spinner
```

```
import androidx.appcompat.app.AppCompatActivity

import androidx.recyclerview.widget.LinearLayoutManager

import androidx.recyclerview.widget.RecyclerView


class MainActivity : AppCompatActivity() {


    private lateinit var recyclerView: RecyclerView

    private lateinit var spinner: Spinner


    override fun onCreate(savedInstanceState: Bundle?) {

        super.onCreate(savedInstanceState)

        setContentView(R.layout.activity_main)


        // Sample To-Do List

        val todoItems = listOf("Item 1", "Item 2", "Item 3", "Item 4", "Item 5")


        recyclerView = findViewById(R.id.recyclerView)

        recyclerView.layoutManager = LinearLayoutManager(this)

        recyclerView.adapter = ToDoListAdapter(todoItems)


        // Spinner - Countries

        val countries = arrayOf("India", "USA", "Canada", "UK", "Australia")

        spinner = findViewById(R.id.spinner)

        val adapter = ArrayAdapter(this, android.R.layout.simple_spinner_dropdown_item, countries)

        spinner.adapter = adapter

    }

}
```

3. ToDoListAdapter.kt

```
package com.example.adapters


import android.view.LayoutInflater

import android.view.View

import android.view.ViewGroup

import android.widget.TextView

import androidx.recyclerview.widget.RecyclerView


class ToDoListAdapter(private val items: List<String>) : RecyclerView.Adapter<ToDoListAdapter.ViewHolder>() {
```




```
   class ViewHolder(itemView: View) :
RecyclerView.ViewHolder(itemView) {

      val itemText: TextView =
itemView.findViewById(android.R.id.text1)

   }


   override fun onCreateViewHolder(parent:
ViewGroup, viewType: Int): ViewHolder {

      val view =
LayoutInflater.from(parent.context)


.inflate(android.R.layout.simple_list_item_1
, parent, false)

      return ViewHolder(view)

   }


   override fun getItemCount(): Int =
items.size


   override fun onBindViewHolder(holder:
ViewHolder, position: Int) {

      holder.itemText.text = items[position]

   }

}
```

8. Dialogs

13.Create an AlertDialog with "Yes" and "No" buttons. Display a Toast message based on the user's choice.

14. Implement a DatePickerDialog and display the selected date in a TextView.

Code :

MainActivity.java

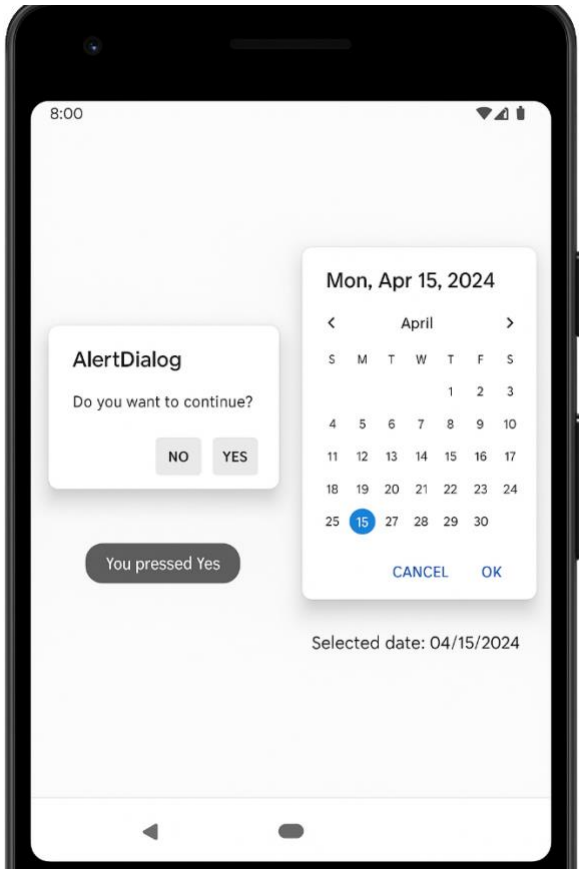
```
package com.example.dialogdemo;


import android.app.DatePickerDialog;
import android.content.DialogInterface;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.DatePicker;
import android.widget.TextView;
import android.widget.Toast;
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import java.util.Calendar;


public class MainActivity extends AppCompatActivity {
    Button alertBtn, dateBtn;
    TextView dateTextView;

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


        alertBtn = findViewById(R.id.alertBtn);
        dateBtn = findViewById(R.id.dateBtn);
        dateTextView = findViewById(R.id.dateTextView);


        // AlertDialog
        alertBtn.setOnClickListener(view -> {
            AlertDialog.Builder builder = new AlertDialog.Builder(MainActivity.this);
            builder.setTitle("Confirmation")
```

```
            .setMessage("Do you want to continue?")
            .setPositiveButton("Yes", (dialog, which) -> {

Toast.makeText(MainActivity.this, "You clicked Yes", Toast.LENGTH_SHORT).show();
            })
            .setNegativeButton("No", (dialog, which) -> {

Toast.makeText(MainActivity.this, "You clicked No", Toast.LENGTH_SHORT).show();
            })
            .show();
    });


    // DatePickerDialog
    dateBtn.setOnClickListener(view -> {
        final Calendar calendar = Calendar.getInstance();
        int year = calendar.get(Calendar.YEAR);
        int month = calendar.get(Calendar.MONTH);
        int day = calendar.get(Calendar.DAY_OF_MONTH);


        DatePickerDialog datePickerDialog = new DatePickerDialog(MainActivity.this,
            (view1, year1, month1, dayOfMonth) -> {
                String selectedDate = dayOfMonth + "/" + (month1 + 1) + "/" + year1;
                dateTextView.setText("Selected Date: " + selectedDate);
            }, year, month, day);
        datePickerDialog.show();
    });
  }
}
```

activity_main.xml

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

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

```
    android:layout_height="wrap_content"
    android:text="Show Alert Dialog" />

<Button
    android:id="@+id/dateBtn"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:text="Pick a Date" />

<TextView
    android:id="@+id/dateTextView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:text="Selected Date:"
    android:textSize="18sp" />

</LinearLayout>
```



9. Menus

15.Add an Options Menu to an activity with actions like "Settings" and "Logout." Handle click events for these options.

16. Implement a Popup Menu that appears when a button is clicked.

Code :

1. MainActivity.java

```
package com.example.menus;


import android.os.Bundle;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.PopupMenu;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;


public class MainActivity extends AppCompatActivity {


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

setContentView(R.layout.activity_main);
    }


    // Options Menu
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        MenuInflater inflater = getMenuInflater();
        inflater.inflate(R.menu.options_menu, menu);
        return true;
    }


    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        switch (item.getItemId()) {
            case R.id.settings:
                Toast.makeText(this, "Settings clicked", Toast.LENGTH_SHORT).show();
                return true;
            case R.id.logout:
```

```
            Toast.makeText(this, "Logout clicked", Toast.LENGTH_SHORT).show();
            return true;
        default:
            return super.onOptionsItemSelected(item);
    }
  }

  // Popup Menu
  public void showPopup(View view) {
    PopupMenu popup = new PopupMenu(this, view);

popup.getMenuInflater().inflate(R.menu.popup_menu, popup.getMenu());


popup.setOnMenuItemClickListener(item -> {
        switch (item.getItemId()) {
            case R.id.edit:
                Toast.makeText(this, "Edit clicked", Toast.LENGTH_SHORT).show();
                return true;
            case R.id.delete:
                Toast.makeText(this, "Delete clicked", Toast.LENGTH_SHORT).show();
                return true;
            case R.id.share:
                Toast.makeText(this, "Share clicked", Toast.LENGTH_SHORT).show();
                return true;
            default:
                return false;
        }
    });

    popup.show();
  }
}
```

2. activity_main.xml

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

xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  tools:context=".MainActivity">

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

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="SHOW MENU"
    android:layout_centerInParent="true"
    android:onClick="showPopup"/>
</RelativeLayout>
```

3. res/menu/popup_menu.xml

```
<menu
xmlns:android="http://schemas.android.com/apk/res/android">
   <item
      android:id="@+id/edit"
      android:title="Edit" />
   <item
      android:id="@+id/delete"
      android:title="Delete" />
   <item
      android:id="@+id/share"
      android:title="Share" />
</menu>
```

10. Notifications

17.Create a simple notification that displays when a button is clicked.

18. Implement a notification with a pending intent that opens a new activity when clicked

Code :

MainActivity.java

```
package com.example.notificationdemo;


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


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


public class MainActivity extends AppCompatActivity {

    private static final String CHANNEL_ID = "demo_channel";


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

        setContentView(R.layout.activity_main);
        createNotificationChannel();


        Button simpleBtn = findViewById(R.id.simpleNotificationBtn);
        Button intentBtn = findViewById(R.id.pendingIntentBtn);


        simpleBtn.setOnClickListener(view -> showSimpleNotification());


        intentBtn.setOnClickListener(view -> showNotificationWithIntent());
    }
```

```
    private void showSimpleNotification() {
        NotificationCompat.Builder builder = new NotificationCompat.Builder(this, CHANNEL_ID)
                .setSmallIcon(R.drawable.ic_launcher_foreground)
                .setContentTitle("Simple Notification")
                .setContentText("This is a basic notification.")
                .setPriority(NotificationCompat.PRIORITY_DEFAULT);

        NotificationManager manager = getSystemService(NotificationManager.class);
        manager.notify(1, builder.build());
    }

    private void showNotificationWithIntent() {
        Intent intent = new Intent(this, SecondActivity.class);
        PendingIntent pendingIntent = PendingIntent.getActivity(this, 0, intent, PendingIntent.FLAG_IMMUTABLE);

        NotificationCompat.Builder builder = new NotificationCompat.Builder(this, CHANNEL_ID)
                .setSmallIcon(R.drawable.ic_launcher_foreground)
                .setContentTitle("PendingIntent Notification")
                .setContentText("Tap to open SecondActivity")
                .setContentIntent(pendingIntent)
                .setAutoCancel(true)
                .setPriority(NotificationCompat.PRIORITY_HIGH);

        NotificationManager manager = getSystemService(NotificationManager.class);
        manager.notify(2, builder.build());
    }

    private void createNotificationChannel() {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            CharSequence name = "Demo Channel";
            String description = "Channel for basic demo notifications";
```

```
        int importance =
NotificationManager.IMPORTANCE_DEFAUL
T;


        NotificationChannel channel = new
NotificationChannel(CHANNEL_ID, name,
importance);
        channel.setDescription(description);


        NotificationManager manager =
getSystemService(NotificationManager.class
);

manager.createNotificationChannel(channel
);
      }
   }
}
```

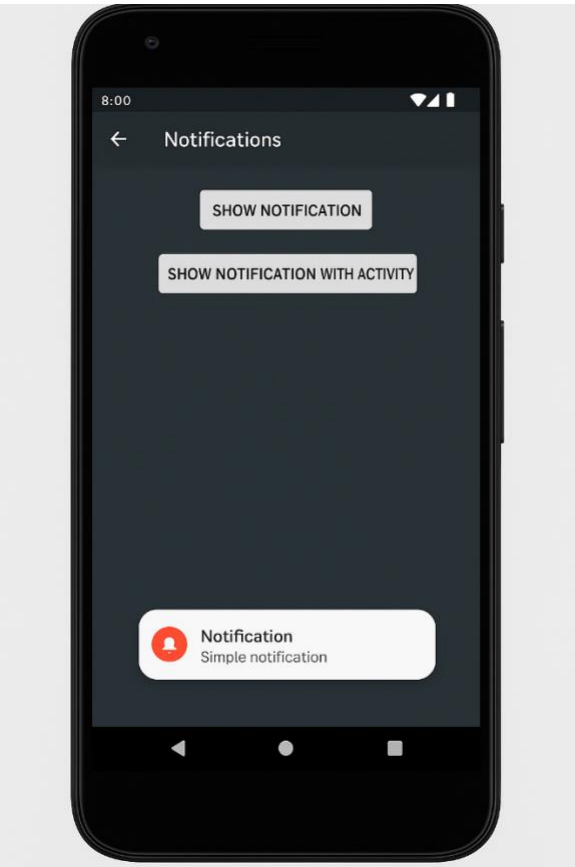
SecondActivity.java

```
package com.example.notificationdemo;

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

import
androidx.appcompat.app.AppCompatActivit
y;

public class SecondActivity extends
AppCompatActivity {
   @Override
   protected void onCreate(Bundle
savedInstanceState) {
      super.onCreate(savedInstanceState);
      TextView tv = new TextView(this);
      tv.setText("You opened SecondActivity
via notification!");
      tv.setTextSize(22);
      setContentView(tv);
   }
}
```

res/layout/activity_main.xml

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

   <Button
```

```
android:id="@+id/simpleNotificationBtn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Show Simple
Notification" />


    <Button
        android:id="@+id/pendingIntentBtn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:text="Show Notification with
Intent" />
</LinearLayout>
```



11. Introduction to Database (SQLite and Firebase)

19.Create an SQLite database to store user information (name, email). Create a form to add data and a ListView to display it.

20.Use Firebase Realtime Database to create an app that allows users to post and view messages.

Code :

✅ activity_main.xml

```
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <EditText
        android:id="@+id/editName"
        android:hint="Name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
/>

    <EditText
        android:id="@+id/editEmail"
        android:hint="Email"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
/>

    <Button
        android:id="@+id/btnAdd"
        android:text="ADD"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
android:layout_gravity="center_horizontal"
        android:layout_marginTop="10dp" />

    <ListView
        android:id="@+id/listView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
/>
</LinearLayout>
```

✅ MainActivity.java

```
public class MainActivity extends AppCompatActivity {

    EditText editName, editEmail;
    Button btnAdd;
    ListView listView;
    ArrayList<String> userList = new ArrayList<>();
    ArrayAdapter<String> adapter;
    DBHelper db;

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

setContentView(R.layout.activity_main);

        editName = findViewById(R.id.editName);
        editEmail = findViewById(R.id.editEmail);
        btnAdd = findViewById(R.id.btnAdd);
        listView = findViewById(R.id.listView);
        db = new DBHelper(this);

        adapter = new ArrayAdapter<>(this, android.R.layout.simple_list_item_1, userList);
        listView.setAdapter(adapter);
        loadData();

        btnAdd.setOnClickListener(v -> {
            String name = editName.getText().toString();
            String email = editEmail.getText().toString();
            db.insertUser(name, email);
            loadData();
            editName.setText("");
            editEmail.setText("");
        });
    }

    private void loadData() {
        userList.clear();
        Cursor cursor = db.getUsers();
        while (cursor.moveToNext()) {
            userList.add(cursor.getString(1) + "\n" + cursor.getString(2));
        }
        adapter.notifyDataSetChanged();
    }
```

```
}
```

✅ DBHelper.java

```
public class DBHelper extends
SQLiteOpenHelper {


    public DBHelper(Context context) {
        super(context, "UserDB", null, 1);
    }


    @Override
    public void onCreate(SQLiteDatabase db)
{
        db.execSQL("CREATE TABLE users(id
INTEGER PRIMARY KEY AUTOINCREMENT,
name TEXT, email TEXT)");
    }


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


    public void insertUser(String name, String
email) {
        SQLiteDatabase db =
this.getWritableDatabase();
        ContentValues cv = new
ContentValues();
        cv.put("name", name);
        cv.put("email", email);
        db.insert("users", null, cv);
    }


    public Cursor getUsers() {
        SQLiteDatabase db =
this.getReadableDatabase();
        return db.rawQuery("SELECT * FROM
users", null);
    }
}
```

✅ activity_main.xml

```
<LinearLayout
xmlns:android="http://schemas.android.co
m/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">


    <EditText
        android:id="@+id/messageInput"
```

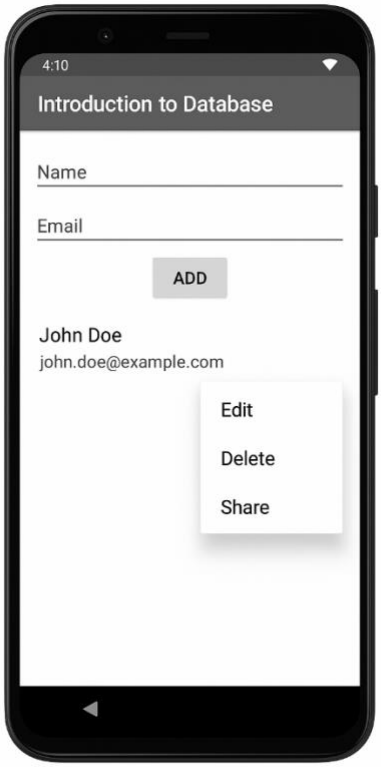
```
    android:hint="Enter your message"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
/>

  <Button
    android:id="@+id/sendBtn"
    android:text="Send"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
/>

  <TextView
    android:id="@+id/messageView"
    android:text="Messages will appear
here"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:paddingTop="20dp" />
</LinearLayout>
```

✅ MainActivity.java

```
public class MainActivity extends
AppCompatActivity {

  EditText messageInput;
  Button sendBtn;
  TextView messageView;
  DatabaseReference databaseReference;

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

setContentView(R.layout.activity_main);

    messageInput =
findViewById(R.id.messageInput);
    sendBtn = findViewById(R.id.sendBtn);
    messageView =
findViewById(R.id.messageView);

    databaseReference =
FirebaseDatabase.getInstance().getReferenc
e("Messages");

    sendBtn.setOnClickListener(v -> {
      String message =
messageInput.getText().toString();
      String id =
databaseReference.push().getKey();

databaseReference.child(id).setValue(messa
ge);
```

```
        messageInput.setText("");
    });
```

```
databaseReference.addValueEventListener(
new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull
DataSnapshot snapshot) {
            StringBuilder messages = new
StringBuilder();
            for (DataSnapshot ds :
snapshot.getChildren()) {

messages.append(ds.getValue(String.class)).
append("\n");
            }

messageView.setText(messages.toString());
        }

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



12. Cursors and Content Values

21.Write a query to fetch all rows from an SQLite table and display them in a RecyclerView using a Cursor.

22.Use ContentValues to insert a new record into an 6 SQLite database

Code :

MyDatabaseHelper.java

```
public class MyDatabaseHelper extends SQLiteOpenHelper {

    private static final String DB_NAME = "students.db";
    private static final int DB_VERSION = 1;

    public static final String TABLE_NAME = "students";
    public static final String COL_ID = "_id";
    public static final String COL_NAME = "name";

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

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

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

    public void insertStudent(String name) {
        SQLiteDatabase db = this.getWritableDatabase();
```

```
        ContentValues values = new
ContentValues();
        values.put(COL_NAME, name);
        db.insert(TABLE_NAME, null, values);
    }

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

🧠 MainActivity.java

```
public class MainActivity extends
AppCompatActivity {

    MyDatabaseHelper dbHelper;
    RecyclerView recyclerView;
    StudentAdapter adapter;
    ArrayList<String> studentList;

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

setContentView(R.layout.activity_main);

        recyclerView =
findViewById(R.id.recyclerView);
        dbHelper = new
MyDatabaseHelper(this);

        // Insert a student using ContentValues
        dbHelper.insertStudent("Kamlesh");

        // Load students using Cursor
        loadStudents();

        recyclerView.setLayoutManager(new
LinearLayoutManager(this));
        adapter = new
StudentAdapter(studentList);
        recyclerView.setAdapter(adapter);
    }

    private void loadStudents() {
        studentList = new ArrayList<>();
        Cursor cursor =
dbHelper.getAllStudents();
        if (cursor.moveToFirst()) {
            do {
                String name =
cursor.getString(cursor.getColumnIndexOrT
hrow("name"));
```

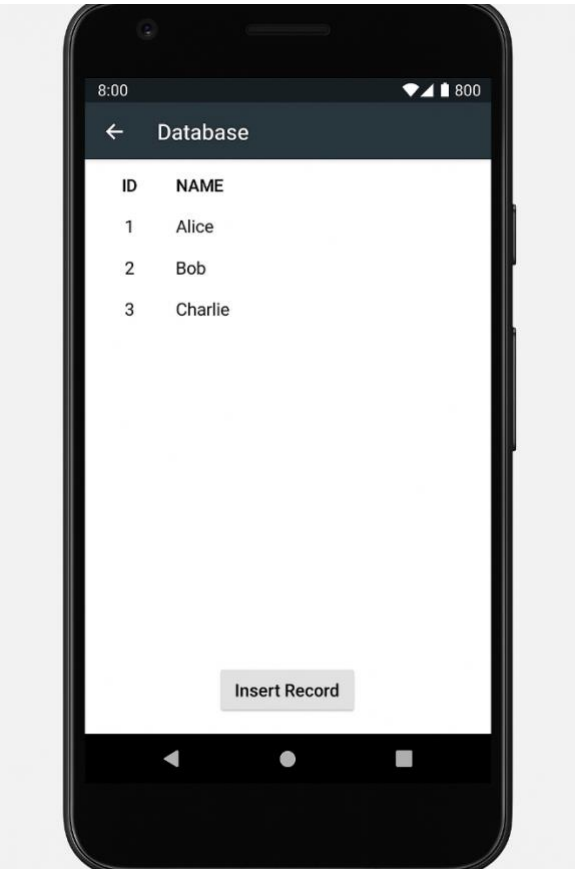
```
            studentList.add(name);
        } while (cursor.moveToNext());
    }
    cursor.close();
  }
}
```

🧠 StudentAdapter.java

```
public class StudentAdapter extends RecyclerView.Adapter<StudentAdapter.ViewHolder> {

  ArrayList<String> studentList;

  public StudentAdapter(ArrayList<String> studentList) {
    this.studentList = studentList;
  }

  @NonNull
  @Override
  public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
    View view = LayoutInflater.from(parent.getContext()).inflate(android.R.layout.simple_list_item_1, parent, false);
    return new ViewHolder(view);
  }

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

holder.textView.setText(studentList.get(position));
  }

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

  class ViewHolder extends RecyclerView.ViewHolder {
    TextView textView;
    ViewHolder(View view) {
      super(view);
      textView = view.findViewById(android.R.id.text1);
    }
  }
}
```

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


<androidx.recyclerview.widget.RecyclerView
        android:id="@+id/recyclerView"
        android:layout_width="0dp"
        android:layout_height="0dp"

app:layout_constraintTop_toTopOf="parent
"

app:layout_constraintBottom_toBottomOf="parent"

app:layout_constraintStart_toStartOf="parent"

app:layout_constraintEnd_toEndOf="parent"/>

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



13. CURD Operations

23. Implement a complete SQLite CRUD operation:

o Add a new record.

o View all records in a RecyclerView.

o Update a record.

o Delete a record.

Code :

🔧 Step 1: SQLite Database Helper

```
public class DBHelper extends SQLiteOpenHelper {
    private static final String DB_NAME = "UserDB";
    private static final int DB_VERSION = 1;

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

    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL("CREATE TABLE Users(id INTEGER PRIMARY KEY AUTOINCREMENT, name TEXT, email TEXT)");
    }

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

    public void insertUser(String name, String email) {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues cv = new ContentValues();
        cv.put("name", name);
        cv.put("email", email);
        db.insert("Users", null, cv);
    }
```

```
    public void updateUser(int id, String name, String email) {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues cv = new ContentValues();
        cv.put("name", name);
        cv.put("email", email);
        db.update("Users", cv, "id=?", new String[]{String.valueOf(id)});
    }

    public void deleteUser(int id) {
        SQLiteDatabase db = this.getWritableDatabase();
        db.delete("Users", "id=?", new String[]{String.valueOf(id)});
    }

    public Cursor getAllUsers() {
        SQLiteDatabase db = this.getReadableDatabase();
        return db.rawQuery("SELECT * FROM Users", null);
    }
}
```

📱 Step 2: Model Class

```
public class User {
    int id;
    String name;
    String email;

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

🎨 Step 3: User Adapter (RecyclerView)

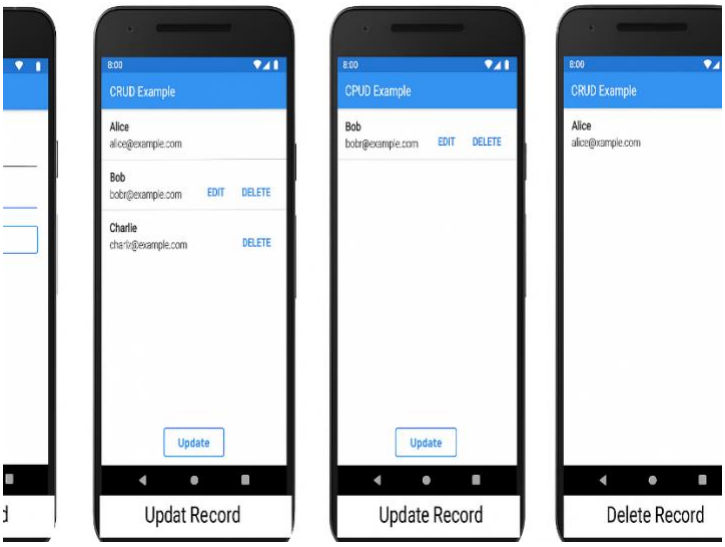
```
public class UserAdapter extends RecyclerView.Adapter<UserAdapter.UserViewHolder> {
    private ArrayList<User> userList;
    private Context context;
    private OnUserClickListener listener;

    public interface OnUserClickListener {
        void onEditClick(User user);
        void onDeleteClick(User user);
    }
```

```
    public UserAdapter(Context context,
ArrayList<User> userList,
OnUserClickListener listener) {
        this.context = context;
        this.userList = userList;
        this.listener = listener;
    }

    @NonNull
    @Override
    public UserViewHolder
onCreateViewHolder(@NonNull ViewGroup
parent, int viewType) {
        View view =
LayoutInflater.from(context).inflate(R.layout
.user_row, parent, false);
        return new UserViewHolder(view);
    }

    @Override
    public void onBindViewHolder(@NonNull
UserViewHolder holder, int position) {
        User user = userList.get(position);
        holder.name.setText(user.name);
        holder.email.setText(user.email);

        holder.editBtn.setOnClickListener(v ->
listener.onEditClick(user));
        holder.deleteBtn.setOnClickListener(v -
> listener.onDeleteClick(user));
    }

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

    public static class UserViewHolder
extends RecyclerView.ViewHolder {
        TextView name, email;
        Button editBtn, deleteBtn;

        public UserViewHolder(@NonNull View
itemView) {
            super(itemView);
            name =
itemView.findViewById(R.id.tvName);
            email =
itemView.findViewById(R.id.tvEmail);
            editBtn =
itemView.findViewById(R.id.btnEdit);
            deleteBtn =
itemView.findViewById(R.id.btnDelete);
        }
    }
}
```

🧠 Step 5: MainActivity

```
public class MainActivity extends AppCompatActivity {
    EditText etName, etEmail;
    Button btnAddUpdate;
    RecyclerView recyclerView;
    DBHelper dbHelper;
    ArrayList<User> userList;
    UserAdapter adapter;

    int updateUserId = -1;

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

setContentView(R.layout.activity_main);

        etName = findViewById(R.id.etName);
        etEmail = findViewById(R.id.etEmail);
        btnAddUpdate = findViewById(R.id.btnAddUpdate);
        recyclerView = findViewById(R.id.recyclerView);
        dbHelper = new DBHelper(this);
        recyclerView.setLayoutManager(new LinearLayoutManager(this));
        userList = new ArrayList<>();
        loadUsers();

        btnAddUpdate.setOnClickListener(v -> {
            String name = etName.getText().toString();
            String email = etEmail.getText().toString();

            if (updateUserId == -1) {
                dbHelper.insertUser(name, email);
            } else {
dbHelper.updateUser(updateUserId, name, email);
                updateUserId = -1;
                btnAddUpdate.setText("Add");
            }

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

```
    private void loadUsers() {
        userList.clear();
        Cursor cursor = dbHelper.getAllUsers();
        while (cursor.moveToNext()) {
            userList.add(new User(
                cursor.getInt(0),
                cursor.getString(1),
                cursor.getString(2)
            ));
        }


        adapter = new UserAdapter(this,
userList, new
UserAdapter.OnUserClickListener() {
            @Override
            public void onEditClick(User user) {
                etName.setText(user.name);
                etEmail.setText(user.email);
                updateUserId = user.id;
                btnAddUpdate.setText("Update");
            }


            @Override
            public void onDeleteClick(User user) {
                dbHelper.deleteUser(user.id);
                loadUsers();
            }
        });

        recyclerView.setAdapter(adapter);
    }
}
```



24. Create an app that fetches data from a public API (e.g., OpenWeatherMap) and displays it in a TextView.

Code :

💻 activity_main.xml

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

    <TextView
        android:id="@+id/weatherText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Fetching weather..."
        android:textSize="20sp"/>
</LinearLayout>
```

🧠 MainActivity.java

```
package com.example.weatherapp;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

import android.os.StrictMode;

import android.widget.TextView;


import org.json.JSONObject;


import java.io.BufferedReader;

import java.io.InputStreamReader;

import java.net.HttpURLConnection;

import java.net.URL;


public class MainActivity extends AppCompatActivity {

    TextView weatherText;

    String apiKey = "your_api_key"; // Replace with your actual API key

    String city = "London";


    @Override
```

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

setContentView(R.layout.activity_main);
        weatherText =
findViewById(R.id.weatherText);

        // Allow network on main thread (not
recommended for production)
        StrictMode.setThreadPolicy(new
StrictMode.ThreadPolicy.Builder().permitAll(
).build());

        fetchWeather();
    }

    private void fetchWeather() {
        try {
            String urlString =
"https://api.openweathermap.org/data/2.5
/weather?q=" + city + "&appid=" + apiKey +
"&units=metric";
            URL url = new URL(urlString);
            HttpURLConnection conn =
(HttpURLConnection) url.openConnection();

            conn.setRequestMethod("GET");
            BufferedReader reader = new
BufferedReader(new
InputStreamReader(conn.getInputStream())
);
            StringBuilder result = new
StringBuilder();
            String line;

            while ((line = reader.readLine()) !=
null) {
                result.append(line);
            }

            JSONObject jsonObject = new
JSONObject(result.toString());
            JSONObject main =
jsonObject.getJSONObject("main");
            JSONObject weather =
jsonObject.getJSONArray("weather").getJSO
NObject(0);

            double temp =
main.getDouble("temp");
            String condition =
weather.getString("main");
            String description =
weather.getString("description");
```

```
            String output = "Temperature: " + temp + " °C\nCondition: " + condition + "\nDescription: " + description;

            weatherText.setText(output);


        } catch (Exception e) {

            weatherText.setText("Failed to fetch weather.");

            e.printStackTrace();

        }

    }

}
```

✅ **Requirements**

1. Internet permission in AndroidManifest.xml
2. API key from https://openweathermap.org/api