**1)Animations :**

**Description:**

Created an Android application that showcases various **text animations** applied to the word **"Nidharsan"** using Java. The app includes:

* A central **TextView** displaying the text "Nidharsan" in **white color**.
* **10 different buttons**, each triggering a unique animation when clicked.
* Animation types implemented include:
  + **Slide**
  + **Bulge**
  + **Shrink**
  + **Fade**
  + **Rotate**
  + **Bounce**
  + **Alpha**
  + **Scale**
  + **Translate**
  + **Slide Up**

These animations enhance user interaction and demonstrate core Android animation techniques using the Animation and AnimationUtils classes with XML-defined effects.

**MainActivity.java:**

package com.example.myapplication;  
  
import android.annotation.SuppressLint;  
import android.os.Bundle;  
import android.view.animation.Animation;  
import android.view.animation.AnimationUtils;  
import android.widget.Button;  
import android.widget.TextView;  
import androidx.appcompat.app.AppCompatActivity;  
  
import com.example.myapplication.R;  
  
public class MainActivity extends AppCompatActivity {  
  
 TextView textView;  
 Button buttonSlide, buttonBulge, buttonShrink, buttonFade, buttonRotate, buttonBounce, buttonAlpha, buttonScale, buttonTranslate, buttonSlideUp;  
  
 @SuppressLint("MissingInflatedId")  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 textView = findViewById(R.id.*textView*); // Add a TextView in your layout for "Nidharsan"  
 buttonSlide = findViewById(R.id.*button\_slide*);  
 buttonBulge = findViewById(R.id.*button\_bulge*);  
 buttonShrink = findViewById(R.id.*button\_shrink*);  
 buttonFade = findViewById(R.id.*button\_fade*);  
 buttonRotate = findViewById(R.id.*button\_rotate*);  
 buttonBounce = findViewById(R.id.*button\_bounce*);  
 buttonAlpha = findViewById(R.id.*button\_alpha*);  
 buttonScale = findViewById(R.id.*button\_scale*);  
 buttonTranslate = findViewById(R.id.*button\_translate*);  
 buttonSlideUp = findViewById(R.id.*button\_slide\_up*);  
  
 buttonSlide.setOnClickListener(v -> applySlideAnimation());  
 buttonBulge.setOnClickListener(v -> applyBulgeAnimation());  
 buttonShrink.setOnClickListener(v -> applyShrinkAnimation());  
 buttonFade.setOnClickListener(v -> applyFadeAnimation());  
 buttonRotate.setOnClickListener(v -> applyRotateAnimation());  
 buttonBounce.setOnClickListener(v -> applyBounceAnimation());  
 buttonAlpha.setOnClickListener(v -> applyAlphaAnimation());  
 buttonScale.setOnClickListener(v -> applyScaleAnimation());  
 buttonTranslate.setOnClickListener(v -> applyTranslateAnimation());  
 buttonSlideUp.setOnClickListener(v -> applySlideUpAnimation());  
 }  
  
 private void applySlideAnimation() {  
 Animation slide = AnimationUtils.*loadAnimation*(this, R.anim.*slide\_in*);  
 textView.startAnimation(slide);  
 }  
  
 private void applyBulgeAnimation() {  
 Animation bulge = AnimationUtils.*loadAnimation*(this, R.anim.*bulge*);  
 textView.startAnimation(bulge);  
 }  
  
 private void applyShrinkAnimation() {  
 Animation shrink = AnimationUtils.*loadAnimation*(this, R.anim.*shrink*);  
 textView.startAnimation(shrink);  
 }  
  
 private void applyFadeAnimation() {  
 Animation fade = AnimationUtils.*loadAnimation*(this, R.anim.*fade*);  
 textView.startAnimation(fade);  
 }  
  
 private void applyRotateAnimation() {  
 Animation rotate = AnimationUtils.*loadAnimation*(this, R.anim.*rotate*);  
 textView.startAnimation(rotate);  
 }  
  
 private void applyBounceAnimation() {  
 Animation bounce = AnimationUtils.*loadAnimation*(this, R.anim.*bounce*);  
 textView.startAnimation(bounce);  
 }  
  
 private void applyAlphaAnimation() {  
 Animation alpha = AnimationUtils.*loadAnimation*(this, R.anim.*alpha*);  
 textView.startAnimation(alpha);  
 }  
  
 private void applyScaleAnimation() {  
 Animation scale = AnimationUtils.*loadAnimation*(this, R.anim.*scale*);  
 textView.startAnimation(scale);  
 }  
  
 private void applyTranslateAnimation() {  
 Animation translate = AnimationUtils.*loadAnimation*(this, R.anim.*translate*);  
 textView.startAnimation(translate);  
 }  
  
 private void applySlideUpAnimation() {  
 Animation slideUp = AnimationUtils.*loadAnimation*(this, R.anim.*slide\_up*);  
 textView.startAnimation(slideUp);  
 }  
}

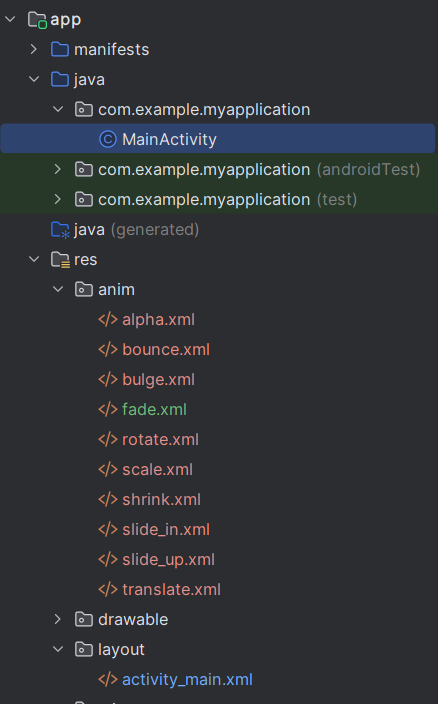
**Activity\_main.xml :**

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:padding="16dp"  
 android:gravity="center">  
  
 <!-- TextView to display the name "Nidharsan" -->  
 <TextView  
 android:id="@+id/textView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Nidharsan"  
 android:textSize="24sp"  
 android:layout\_marginBottom="20dp"  
 android:textColor="#FFFFFF"  
 android:gravity="center"/>  
  
  
 <Button  
 android:id="@+id/button\_slide"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Slide Animation" />  
  
 <Button  
 android:id="@+id/button\_bulge"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Bulge Animation" />  
  
 <Button  
 android:id="@+id/button\_shrink"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Shrink Animation" />  
  
 <Button  
 android:id="@+id/button\_fade"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Fade Animation" />  
  
 <Button  
 android:id="@+id/button\_rotate"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Rotate Animation" />  
  
 <Button  
 android:id="@+id/button\_bounce"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Bounce Animation" />  
  
 <Button  
 android:id="@+id/button\_alpha"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Alpha Animation" />  
  
 <Button  
 android:id="@+id/button\_scale"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Scale Animation" />  
  
 <Button  
 android:id="@+id/button\_translate"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Translate Animation" />  
  
 <Button  
 android:id="@+id/button\_slide\_up"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Slide Up Animation" />  
  
</LinearLayout>

**AndroidMainfest.xml:**

<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools">  
  
 <application  
 android:allowBackup="true"  
 android:dataExtractionRules="@xml/data\_extraction\_rules"  
 android:fullBackupContent="@xml/backup\_rules"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.MyApplication"  
 tools:targetApi="31">  
 <activity  
 android:name=".MainActivity"  
 android:exported="true">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity>  
 </application>  
  
</manifest>

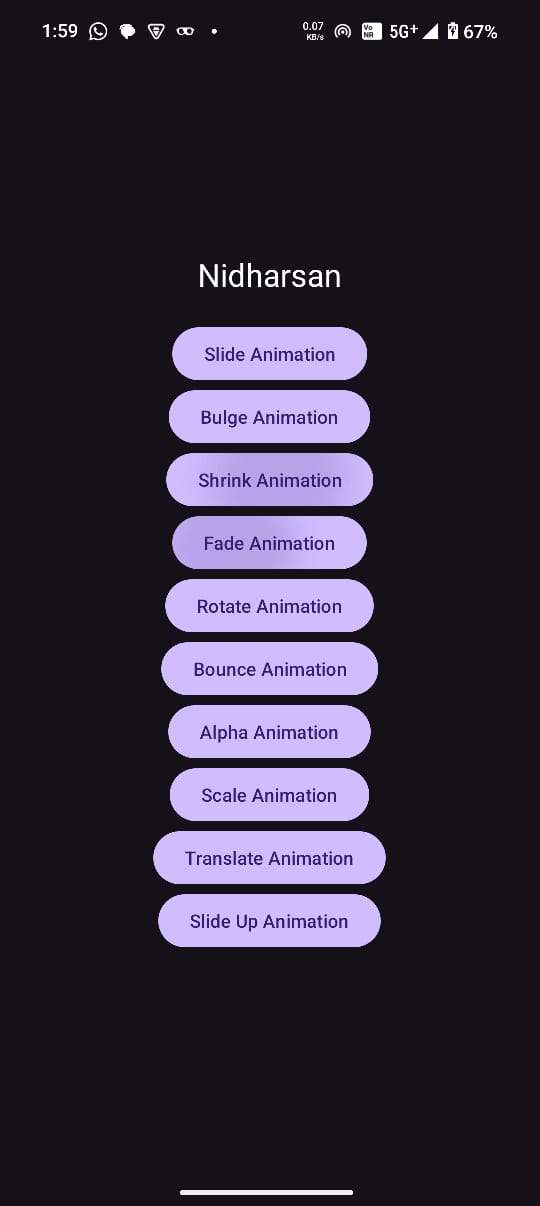
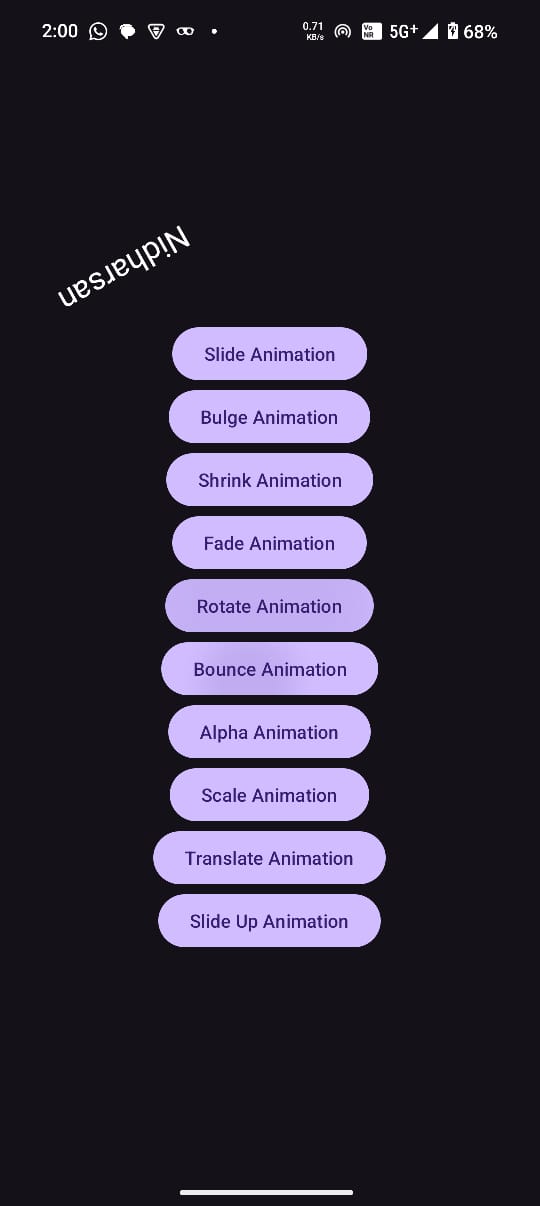
**FileStructure:**



**Project Link :**

<https://github.com/NIDHARSAN-V/Android/tree/master/Animation>

**Output:**

**2)Bundle :**

**Description:**

As part of the project, inter-activity communication was implemented in the Android application using **Intent and Bundle** to transfer data from one activity to another. This technique is essential for sharing data such as user inputs or application state between different screens. The following steps outline the implementation:

* An Intent object was created in the first activity to initiate a transition to the second activity:

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

* A key-value pair was attached to the intent using the putExtra() method, which allows primitive data types or strings to be passed:

intent.putExtra("username", "Nidharsan");

* The second activity was launched using the startActivity(intent) method:

startActivity(intent);

* In the second activity, the passed data was retrieved by accessing the intent and extracting the value using the corresponding key:

Intent intent = getIntent();

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

* The received value was then utilized as needed—such as displaying it in a TextView or integrating it into business logic.

This approach demonstrates a fundamental and widely used method in Android development for managing data transfer between components in an application.

**MainActivity.java:**

package com.example.bundle;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import androidx.appcompat.app.AppCompatActivity;  
  
import com.example.bundle.R;  
import com.example.bundle.SecondActivity;  
  
public class MainActivity extends AppCompatActivity implements View.OnClickListener {  
  
 Button btnPassBundles, btnNoPassBundle;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 btnPassBundles = findViewById(R.id.*btnPassBundles*);  
 btnNoPassBundle = findViewById(R.id.*btnNoPassBundle*);  
  
 // one button will pass the bundle and other button  
 // will not pass the bundle  
 btnPassBundles.setOnClickListener(this);  
 btnNoPassBundle.setOnClickListener(this);  
 }  
  
 @Override  
 public void onClick(View view) {  
 int id = view.getId();  
 if (id == R.id.*btnPassBundles*) {  
 Bundle bundle = new Bundle();  
 bundle.putString("key1", "Passing Bundle From Main Activity to 2nd Activity");  
 Intent intent = new Intent(MainActivity.this, SecondActivity.class);  
 intent.putExtras(bundle);  
 startActivity(intent);  
 } else if (id == R.id.*btnNoPassBundle*) {  
 Bundle bundle = new Bundle();  
 bundle.putString("key1", "Not passing Bundle From Main Activity");  
 bundle.clear();  
 Intent intent = new Intent(MainActivity.this, SecondActivity.class);  
 intent.putExtras(bundle);  
 startActivity(intent);  
 }  
 }  
  
}

**SecondActivity.java:**

package com.example.bundle;  
  
import androidx.appcompat.app.AppCompatActivity;  
import android.os.Bundle;  
import android.widget.TextView;  
  
public class SecondActivity extends AppCompatActivity {  
  
 TextView txtString;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_second*);  
  
 txtString = findViewById(R.id.*txtString*);  
  
 // getting the bundle from the intent  
 Bundle bundle = getIntent().getExtras();  
  
 // setting the text in the textview  
 txtString.setText(bundle.getString("key1", "No value from the MainActivity"));  
 }  
}

**Activity\_main.xml :**

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:padding="10dp"  
 tools:context=".MainActivity">  
  
 <Button  
 android:id="@+id/btnPassBundles"  
 android:layout\_width="275dp"  
 android:layout\_height="101dp"  
 android:layout\_marginTop="250dp"  
 android:text="Pass Data Into Bundle"  
 android:textSize="24sp"  
 app:layout\_constraintHorizontal\_bias="0.498"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <Button  
 android:id="@+id/btnNoPassBundle"  
 android:layout\_width="277dp"  
 android:layout\_height="92dp"  
 android:layout\_marginBottom="220dp"  
 android:layout\_marginTop="75dp"  
 android:text="Pass No Data/Empty BUNDLE"  
 android:textSize="24sp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/btnPassBundles" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>

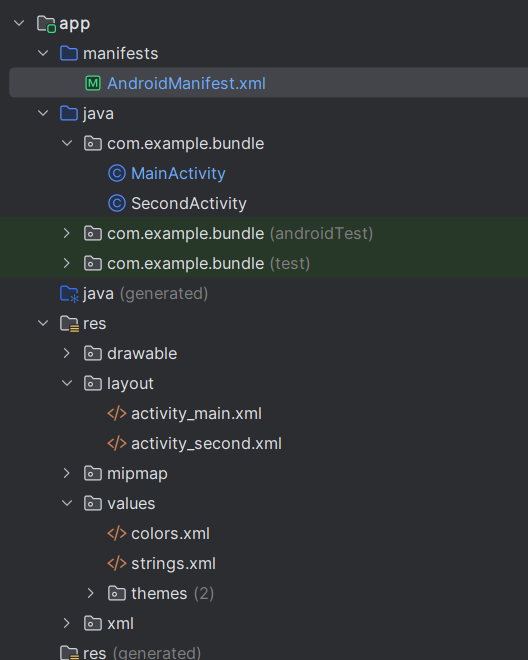
**Activity\_second.xml :**

<?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:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".SecondActivity">  
  
 <TextView  
 android:id="@+id/txtString"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="348dp"  
 android:text="String from MainActivity"  
 android:textSize="40sp"  
 android:textStyle="bold"  
 android:gravity="center"  
 android:textColor="#008000"  
 app:layout\_constraintHorizontal\_bias="0.428"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>

**AndroidMainfest.xml:**

<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 package="com.example.bundle">  
  
 <application  
 android:allowBackup="true"  
 android:dataExtractionRules="@xml/data\_extraction\_rules"  
 android:fullBackupContent="@xml/backup\_rules"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.Bundle"  
 tools:targetApi="31">  
  
 <activity android:name=".SecondActivity" />  
  
 <activity  
 android:name=".MainActivity"  
 android:exported="true">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity>  
  
 </application>  
  
</manifest>

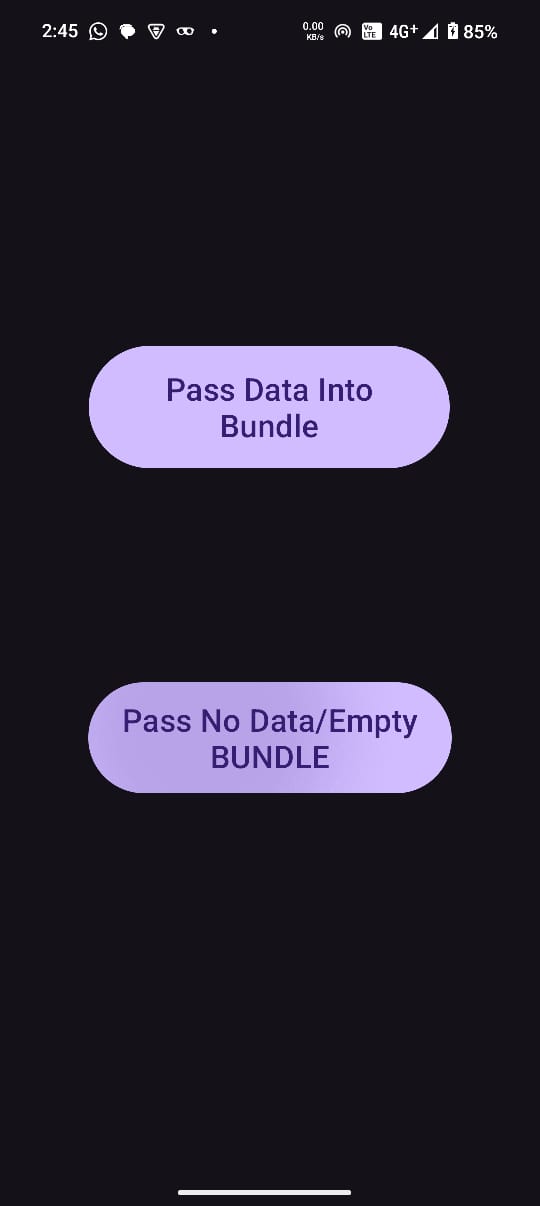
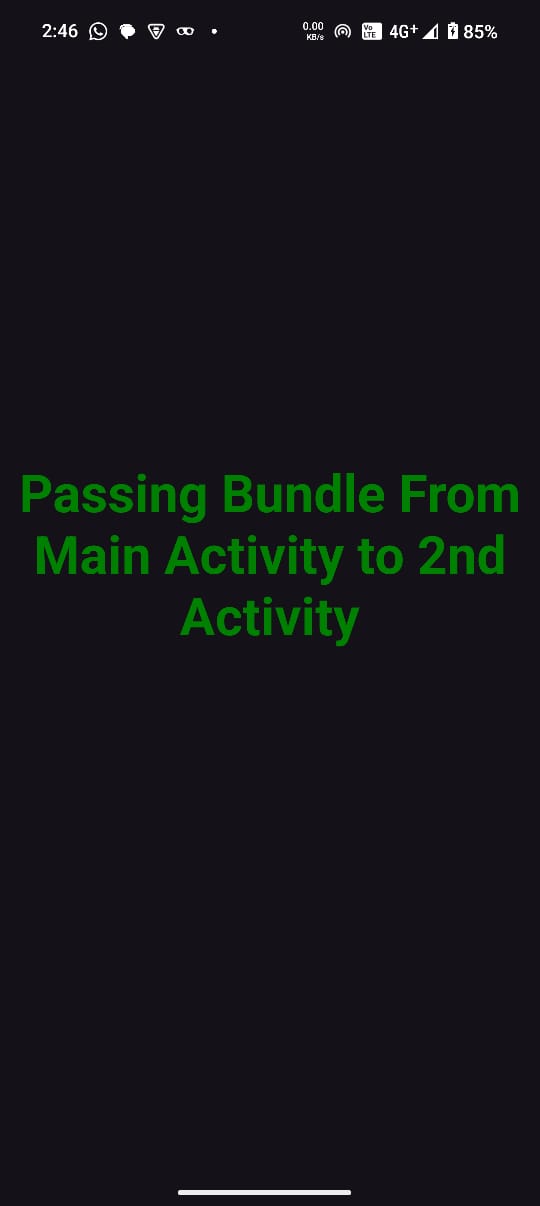
**FileStructure:**

****

**Project Link :**

<https://github.com/NIDHARSAN-V/Android/tree/master/Bundle>

**Output:**

**3)PhoneCall feature using Intent:**

**Description:**

This module adds a direct calling feature to the Android application, allowing users to initiate a phone call by entering a number and pressing a button.

**Key Features:**

* **User Input Interface**:  
  The application provides a text input field for users to enter a phone number and a button to initiate the call.
* **Intent-Based Call Handling**:  
  When the button is pressed, the app uses an intent to initiate a phone call action via the device's calling service.
* **Runtime Permission Management**:  
  Since calling is a sensitive operation, the app dynamically checks for and requests the necessary CALL\_PHONE permission from the user at runtime.
* **Validation and Feedback**:  
  The app ensures the entered phone number is not empty before proceeding. User feedback is provided through toast messages indicating whether the required permission was granted or denied.

**MainActivity.java:**

package com.example.calling;  
  
import android.Manifest;  
import android.content.Intent;  
import android.content.pm.PackageManager;  
import android.net.Uri;  
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.app.ActivityCompat;  
import androidx.core.graphics.Insets;  
import androidx.core.view.ViewCompat;  
import androidx.core.view.WindowInsetsCompat;  
  
public class MainActivity extends AppCompatActivity {  
  
  
 private EditText phone\_number;  
 private Button call\_button;  
  
 private static final int *CALL\_PHONE\_PERMISSION\_CODE* = 100;  
  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 EdgeToEdge.*enable*(this);  
 setContentView(R.layout.*activity\_main*);  
 call\_from\_phone();  
  
 }  
  
  
 private void call\_from\_phone()  
 {  
 phone\_number = findViewById(R.id.*edit\_phone*);  
 call\_button = findViewById(R.id.*call\_button*);  
  
 call\_button.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 String phonenumber = phone\_number.getText().toString();  
  
 if(!phonenumber.isEmpty())  
 {  
 if(ActivityCompat.*checkSelfPermission*(MainActivity.this, Manifest.permission.*CALL\_PHONE*)== PackageManager.*PERMISSION\_GRANTED*)  
 {  
 Intent phoneintent = new Intent(Intent.*ACTION\_CALL*);  
  
 phoneintent.setData(Uri.*parse*("tel:" + phonenumber));  
 startActivity(phoneintent);  
  
 }  
 else {  
 requestCallPermission();  
 }  
  
 }  
 }  
 });  
  
  
 }  
  
  
  
 private void requestCallPermission() {  
 ActivityCompat.*requestPermissions*(this,  
 new String[]{Manifest.permission.*CALL\_PHONE*}, *CALL\_PHONE\_PERMISSION\_CODE*);  
 }  
  
  
 @Override  
 public void onRequestPermissionsResult(int requestCode, String[] permissions, int[] grantResults) {  
 super.onRequestPermissionsResult(requestCode, permissions, grantResults);  
  
 if (requestCode == *CALL\_PHONE\_PERMISSION\_CODE*) {  
 if (grantResults.length > 0 && grantResults[0] == PackageManager.*PERMISSION\_GRANTED*) {  
 Toast.*makeText*(this, "Permission Granted", Toast.*LENGTH\_SHORT*).show();  
 } else {  
 Toast.*makeText*(this, "Permission Denied", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
 }  
}

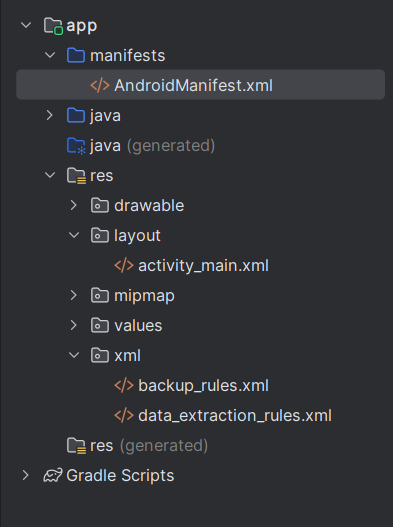
**Activity\_main.xml :**

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:id="@+id/main"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:gravity="center"  
 android:padding="24dp"  
 android:background="@color/black"  
 tools:context=".MainActivity">  
  
 <!-- CardView for Input Field -->  
 <androidx.cardview.widget.CardView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="16dp"  
 app:cardCornerRadius="12dp"  
 app:cardElevation="6dp">  
  
 <EditText  
 android:id="@+id/edit\_phone"  
 android:layout\_width="match\_parent"  
 android:layout\_height="56dp"  
 android:hint="Enter phone number..."  
 android:inputType="phone"  
 android:padding="16dp"  
 android:textSize="16sp"  
 android:background="@null"/>  
 </androidx.cardview.widget.CardView>  
  
 <!-- Call Button -->  
 <Button  
 android:id="@+id/call\_button"  
 android:layout\_width="match\_parent"  
 android:layout\_height="56dp"  
 android:backgroundTint="@color/green"  
 android:text="Dial"  
 android:textColor="@color/white"  
 android:textSize="18sp"  
   
 android:fontFamily="sans-serif-medium"  
 android:layout\_marginTop="16dp"  
 android:background="@drawable/rounded\_button"/>  
</LinearLayout>

**AndroidMainfest.xml:**

<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools">  
  
 <uses-feature  
 android:name="android.hardware.telephony"  
 android:required="false"/>  
  
 <uses-permission android:name="android.permission.CALL\_PHONE"/>  
  
 <application  
 android:allowBackup="true"  
 android:dataExtractionRules="@xml/data\_extraction\_rules"  
 android:fullBackupContent="@xml/backup\_rules"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.Calling"  
 tools:targetApi="31">  
  
 <activity  
 android:name=".MainActivity"  
 android:exported="true">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity>  
 </application>  
  
</manifest>

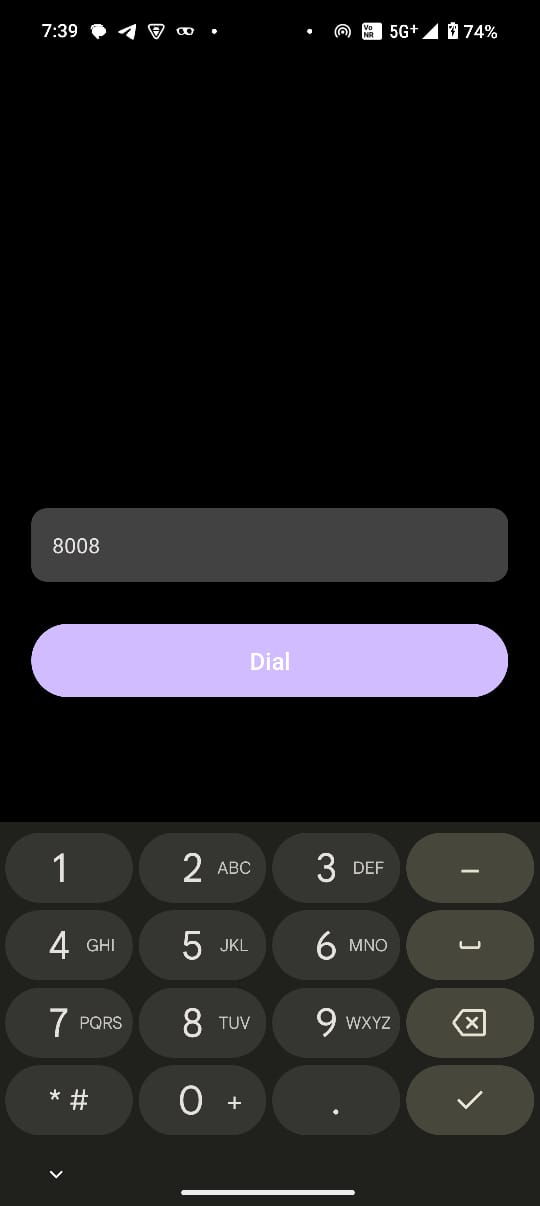
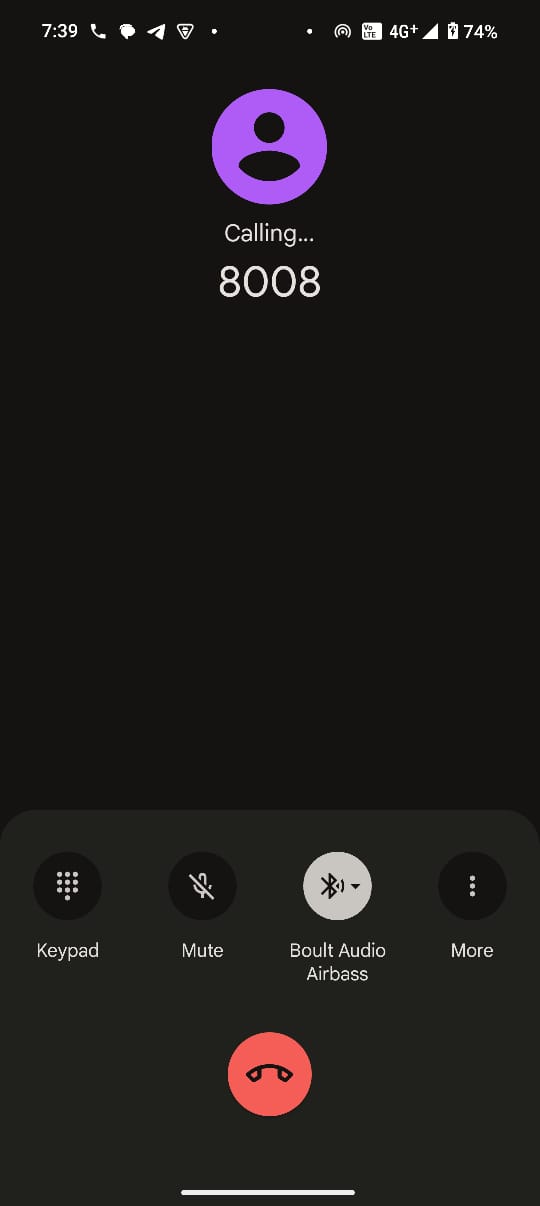
**FileStructure:**

****

**Project Link:**

<https://github.com/NIDHARSAN-V/Android/tree/master/Calling>

**Output:**

**4)** **Android Contact Manager using SQLite:**

**Description:**

This project is a simple Contact Manager Android application that allows users to add, edit, and delete contact information using **SQLite**, the native database provided by Android. It demonstrates full **CRUD (Create, Read, Update, Delete)** operations and efficient data handling through a clean and user-friendly interface.

**Add Contacts**  
Users can input a contact name and phone number, then tap a button to save the contact. The information is stored in a local SQLite database.

**Display Contact List**  
All saved contacts are retrieved from the database and displayed in a ListView using an ArrayAdapter. Each entry in the list shows the contact's name.

**Context Menu (Long Press)**  
A context menu appears when the user long-presses on a contact item in the list. The menu provides the following options:

* **Edit** – Launches a new screen to modify the contact’s details.
* **Delete** – Removes the contact from the database.

**Edit Contacts**  
The Edit screen is accessed using an Intent, and the selected contact's ID is passed as an extra. The user can change the name and phone number, and the updated data is saved back into the SQLite database.

**Delete Contacts**  
Contacts can be removed directly from the main list using the context menu. A toast message provides feedback after deletion.

**SQLite Integration**  
The project uses a custom SQLiteOpenHelper class to manage the database lifecycle, including creation and versioning.

**MainActivity.java:**

package com.example.editdelete;  
  
import android.content.ContentValues;  
import android.content.Intent;  
import android.database.Cursor;  
import android.database.sqlite.SQLiteDatabase;  
import android.os.Bundle;  
import android.view.ContextMenu;  
import android.view.MenuItem;  
import android.view.View;  
import android.widget.AdapterView;  
import android.widget.ArrayAdapter;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.ListView;  
import android.widget.Toast;  
import androidx.appcompat.app.AppCompatActivity;  
import java.util.ArrayList;  
  
public class MainActivity extends AppCompatActivity {  
 ListView listView;  
 EditText nameInput, phoneInput;  
 Button addContactButton;  
 ArrayList<Contact> contactList;  
 ArrayAdapter<String> adapter;  
 DatabaseHelper dbHelper;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 // Initialize views  
 listView = findViewById(R.id.*listView*);  
 nameInput = findViewById(R.id.*nameInput*);  
 phoneInput = findViewById(R.id.*phoneInput*);  
 addContactButton = findViewById(R.id.*addContactButton*);  
 dbHelper = new DatabaseHelper(this);  
 contactList = new ArrayList<>();  
  
 loadContacts();  
 registerForContextMenu(listView);  
  
 // Handle Add Contact Button Click  
 addContactButton.setOnClickListener(v -> {  
 String name = nameInput.getText().toString().trim();  
 String phone = phoneInput.getText().toString().trim();  
 if (!name.isEmpty() && !phone.isEmpty()) {  
 addContact(name, phone);  
 } else {  
 Toast.*makeText*(this, "Please enter name and phone", Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
 }  
  
 private void loadContacts() {  
 contactList.clear();  
 SQLiteDatabase db = dbHelper.getReadableDatabase();  
 Cursor cursor = db.rawQuery("SELECT \* FROM contacts", null);  
  
 ArrayList<String> names = new ArrayList<>();  
 while (cursor.moveToNext()) {  
 int id = cursor.getInt(0);  
 String name = cursor.getString(1);  
 String phone = cursor.getString(2);  
 contactList.add(new Contact(id, name, phone));  
 names.add(name);  
 }  
 cursor.close();  
 adapter = new ArrayAdapter<>(this, android.R.layout.*simple\_list\_item\_1*, names);  
 listView.setAdapter(adapter);  
 }  
  
 private void addContact(String name, String phone) {  
 SQLiteDatabase db = dbHelper.getWritableDatabase();  
 ContentValues values = new ContentValues();  
 values.put("name", name);  
 values.put("phone", phone);  
 db.insert("contacts", null, values);  
// db.close();  
 loadContacts(); // Refresh List  
 nameInput.setText(""); // Clear input fields  
 phoneInput.setText("");  
 Toast.*makeText*(this, "Contact Added", Toast.*LENGTH\_SHORT*).show();  
 }  
  
 @Override  
 public void onCreateContextMenu(ContextMenu menu, View v, ContextMenu.ContextMenuInfo menuInfo) {  
 super.onCreateContextMenu(menu, v, menuInfo);  
 getMenuInflater().inflate(R.menu.*context\_menu*, menu); // Ensure context\_menu.xml exists!  
 }  
  
  
  
  
 @Override  
 public boolean onContextItemSelected(MenuItem item) {  
 AdapterView.AdapterContextMenuInfo info = (AdapterView.AdapterContextMenuInfo) item.getMenuInfo();  
 Contact selectedContact = contactList.get(info.position);  
  
 if (item.getItemId() == R.id.*edit*) {  
 Intent intent = new Intent(this, EditContactActivity.class);  
 intent.putExtra("CONTACT\_ID", selectedContact.getId());  
 startActivity(intent);  
 return true;  
 } else if (item.getItemId() == R.id.*delete*) {  
 dbHelper.deleteContact(selectedContact.getId());  
 loadContacts(); // Refresh List  
 Toast.*makeText*(this, "Contact Deleted", Toast.*LENGTH\_SHORT*).show();  
 return true;  
 }  
 return super.onContextItemSelected(item);  
 }  
}

**EditContactActivity.java :**

package com.example.editdelete;  
  
import android.content.ContentValues;  
import android.database.Cursor;  
import android.database.sqlite.SQLiteDatabase;  
import android.os.Bundle;  
import android.widget.EditText;  
import android.widget.Button;  
import androidx.appcompat.app.AppCompatActivity;  
  
  
  
public class EditContactActivity extends AppCompatActivity {  
 EditText nameEdit, phoneEdit;  
 Button saveButton;  
 DatabaseHelper dbHelper;  
 int contactId;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_edit\_contact*);  
  
 nameEdit = findViewById(R.id.*nameEdit*);  
 phoneEdit = findViewById(R.id.*phoneEdit*);  
 saveButton = findViewById(R.id.*saveButton*);  
 dbHelper = new DatabaseHelper(this);  
  
 contactId = getIntent().getIntExtra("CONTACT\_ID", -1);  
 loadContactData(contactId);  
  
 saveButton.setOnClickListener(v -> {  
 updateContact(contactId, nameEdit.getText().toString(), phoneEdit.getText().toString());  
 finish();  
 });  
 }  
  
 private void loadContactData(int id) {  
 SQLiteDatabase db = dbHelper.getReadableDatabase();  
 Cursor cursor = db.rawQuery("SELECT \* FROM contacts WHERE id=?", new String[]{String.*valueOf*(id)});  
 if (cursor.moveToFirst()) {  
 nameEdit.setText(cursor.getString(1));  
 phoneEdit.setText(cursor.getString(2));  
 }  
 cursor.close();  
 }  
  
 private void updateContact(int id, String name, String phone) {  
 SQLiteDatabase db = dbHelper.getWritableDatabase();  
 ContentValues values = new ContentValues();  
 values.put("name", name);  
 values.put("phone", phone);  
 db.update("contacts", values, "id=?", new String[]{String.*valueOf*(id)});  
 db.close();  
 }  
}

**DatabaseHelper.java :**

package com.example.editdelete;  
  
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* = "contacts.db";  
 private static final int *DATABASE\_VERSION* = 1;  
 private static final String *TABLE\_NAME* = "contacts";  
  
 public DatabaseHelper(Context context) { // Use Context instead of MainActivity  
 super(context, *DATABASE\_NAME*, null, *DATABASE\_VERSION*);  
 }  
  
 @Override  
 public void onCreate(SQLiteDatabase db) {  
 String createTable = "CREATE TABLE " + *TABLE\_NAME* + " (id INTEGER PRIMARY KEY AUTOINCREMENT, name TEXT, phone TEXT)";  
 db.execSQL(createTable);  
 }  
  
 @Override  
 public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {  
 db.execSQL("DROP TABLE IF EXISTS " + *TABLE\_NAME*);  
 onCreate(db);  
 }  
  
 // Method to delete a contact  
 public void deleteContact(int id) {  
 SQLiteDatabase db = this.getWritableDatabase();  
 db.delete(*TABLE\_NAME*, "id=?", new String[]{String.*valueOf*(id)});  
 db.close();  
 }  
  
 // Method to retrieve all contacts  
 public Cursor getReadableData() {  
 SQLiteDatabase db = this.getReadableDatabase();  
 return db.rawQuery("SELECT \* FROM " + *TABLE\_NAME*, null);  
 }  
}

**Contact.java :**

package com.example.editdelete;  
  
public class Contact {  
 private int id;  
 private String name;  
 private String phone;  
  
 public Contact(int id, String name, String phone) {  
 this.id = id;  
 this.name = name;  
 this.phone = phone;  
 }  
  
 public int getId() { return id; }  
 public String getName() { return name; }  
 public String getPhone() { return phone; }  
}

**Activity\_main.xml :**

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:orientation="vertical"  
 android:padding="16dp"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <!-- Name Input -->  
 <EditText  
 android:id="@+id/nameInput"  
 android:hint="Enter Name"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"/>  
  
 <!-- Phone Input -->  
 <EditText  
 android:id="@+id/phoneInput"  
 android:hint="Enter Phone"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:inputType="phone"/>  
  
 <!-- Add Contact Button -->  
 <Button  
 android:id="@+id/addContactButton"  
 android:text="Add Contact"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"/>  
  
 <!-- Contact List -->  
 <ListView  
 android:id="@+id/listView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"/>  
</LinearLayout>

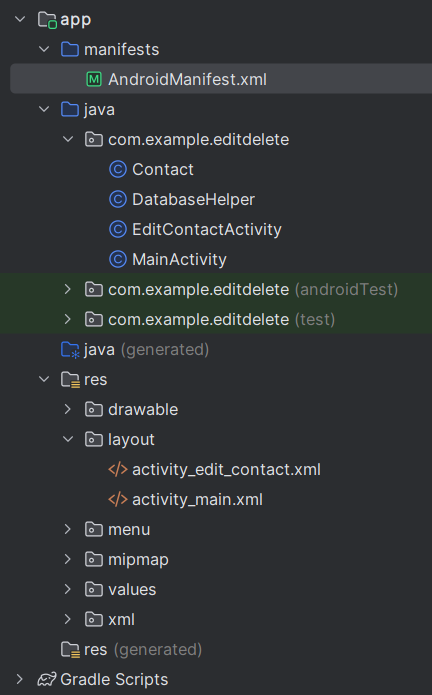
**Activity\_edit\_contact.xml :**

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:orientation="vertical"  
 android:padding="16dp"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <EditText android:id="@+id/nameEdit"  
 android:hint="Name"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"/>  
  
 <EditText android:id="@+id/phoneEdit"  
 android:hint="Phone"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"/>  
  
 <Button android:id="@+id/saveButton"  
 android:text="Save"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"/>  
</LinearLayout>

**AndroidMainfest.xml:**

<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools">  
  
 <application  
 android:allowBackup="true"  
 android:dataExtractionRules="@xml/data\_extraction\_rules"  
 android:fullBackupContent="@xml/backup\_rules"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.EditDelete"  
 tools:targetApi="31">  
 <activity  
 android:name=".MainActivity"  
 android:exported="true">  
  
  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity>  
 <activity  
 android:name="com.example.editdelete.EditContactActivity"  
 android:exported="false" />  
 </application>  
  
</manifest>

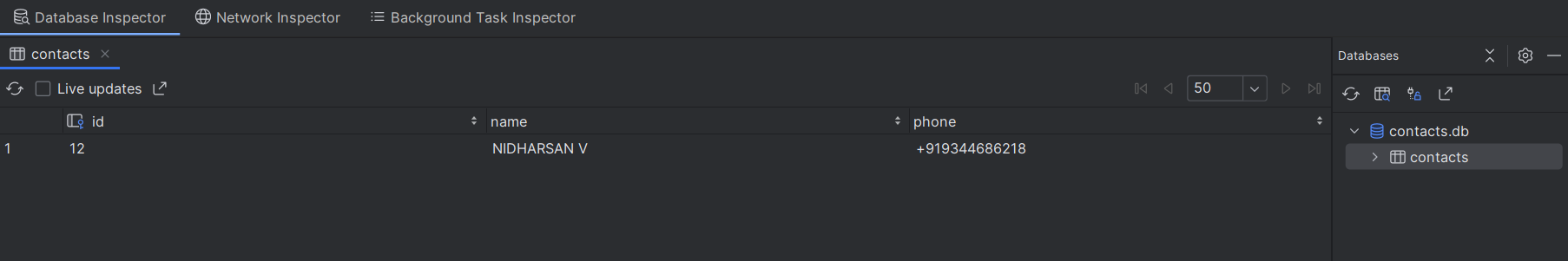
**FileStructure:**

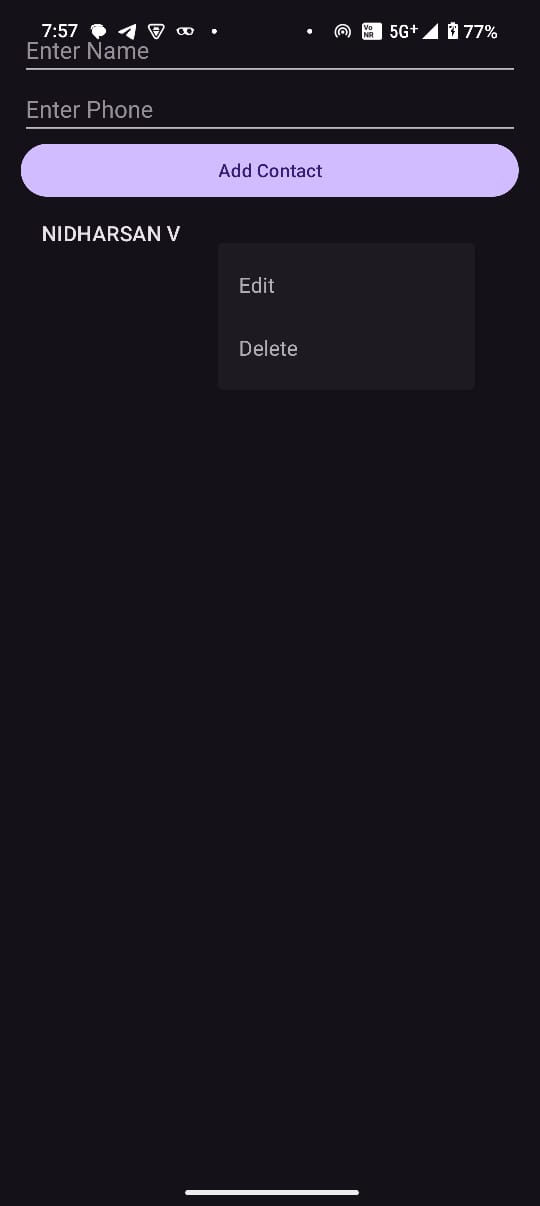
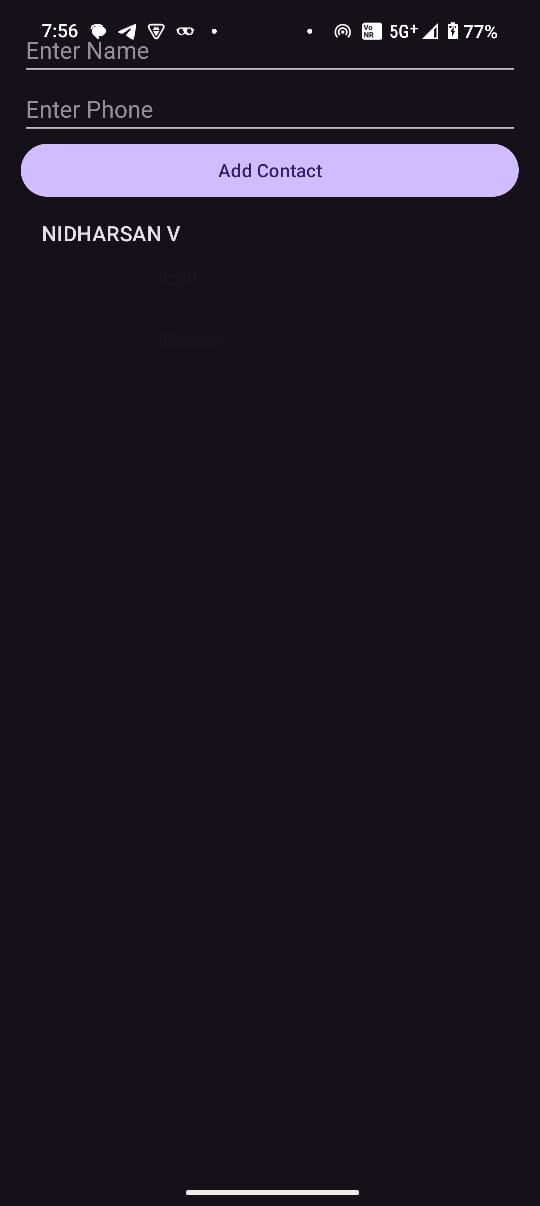
****

**Project Link:**

<https://github.com/NIDHARSAN-V/Android/tree/master/EditDelete>

**Output:**



**5)** **Android Notification App:**

**Description:**

This project is an Android application that demonstrates the implementation of push-style **local notifications** using the **Android Notification API**. It includes the creation of a custom notification channel and displays a notification when the user taps a button.

**Notification Permission Handling**  
For Android 13 (API 33) and above, the app dynamically requests the POST\_NOTIFICATIONS permission using ActivityCompat to comply with runtime permission requirements.

**Notification Channel Creation**  
A NotificationChannel is created programmatically for Android 8.0 (API 26) and above. The channel includes:

* High importance for priority visibility
* Vibration pattern for custom tactile feedback
* Green light color for LED notifications
* Public lock screen visibility

**Triggering Notifications**  
Upon clicking the button, the app sends a notification containing:

* An icon
* A title and message
* Auto-dismiss behavior
* Default sound/vibration settings
* A PendingIntent to reopen the app when the notification is tapped

**MainActivity.java:**

package com.example.notification;  
  
import android.Manifest;  
import android.app.NotificationChannel;  
import android.app.NotificationManager;  
import android.app.PendingIntent;  
import android.content.Context;  
import android.content.Intent;  
import android.content.pm.PackageManager;  
import android.graphics.Color;  
import android.os.Build;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
  
import androidx.activity.EdgeToEdge;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.core.app.ActivityCompat;  
import androidx.core.app.NotificationCompat;  
import androidx.core.app.NotificationManagerCompat;  
  
public class MainActivity extends AppCompatActivity {  
  
 private String channel\_id = "nid\_notifications";  
 private String desc = "Hello Notification from Nidharsan";  
 private int noti\_id = 1;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 EdgeToEdge.*enable*(this);  
 setContentView(R.layout.*activity\_main*);  
  
 Button btn = findViewById(R.id.*btn*);  
 createChannel();  
  
 btn.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*TIRAMISU*) {  
 if (ActivityCompat.*checkSelfPermission*(MainActivity.this, Manifest.permission.*POST\_NOTIFICATIONS*) != PackageManager.*PERMISSION\_GRANTED*) {  
 ActivityCompat.*requestPermissions*(MainActivity.this, new String[]{Manifest.permission.*POST\_NOTIFICATIONS*}, 101);  
 return;  
 }  
 }  
 send();  
 }  
 });  
 }  
  
 private void createChannel() {  
 if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*O*) {  
 NotificationChannel notificationChannel = new NotificationChannel(  
 channel\_id,  
 "Nidharsan Notifications",  
 NotificationManager.*IMPORTANCE\_HIGH* );  
  
 notificationChannel.enableLights(true);  
 notificationChannel.setLightColor(Color.*GREEN*);  
 notificationChannel.enableVibration(true);  
 notificationChannel.setVibrationPattern(new long[]{100, 200, 300, 400,500,600});  
 notificationChannel.setLockscreenVisibility(NotificationCompat.*VISIBILITY\_PUBLIC*);  
  
 NotificationManager notificationManager = (NotificationManager) getSystemService(Context.*NOTIFICATION\_SERVICE*);  
  
 if (notificationManager != null) {  
 notificationManager.createNotificationChannel(notificationChannel);  
 }  
 }  
 }  
  
 private void send() {  
 // PendingIntent to open the app when clicked  
 Intent intent = new Intent(this, MainActivity.class);  
 PendingIntent pendingIntent = PendingIntent.*getActivity*(this, 0, intent, PendingIntent.*FLAG\_UPDATE\_CURRENT* | PendingIntent.*FLAG\_IMMUTABLE*);  
  
 NotificationCompat.Builder builder = new NotificationCompat.Builder(this, channel\_id)  
 .setSmallIcon(R.mipmap.*ic\_launcher*) // Required icon  
 .setContentTitle("New Notification")  
 .setContentText(desc)  
 .setPriority(NotificationCompat.*PRIORITY\_HIGH*)  
 .setAutoCancel(true)  
 .setContentIntent(pendingIntent) // Open app when clicked  
 .setDefaults(NotificationCompat.*DEFAULT\_ALL*) // Enable sound, vibration, etc.  
 .setVibrate(new long[]{100, 200, 300, 400});  
  
 NotificationManagerCompat notificationManager = NotificationManagerCompat.*from*(this);  
  
 if (ActivityCompat.*checkSelfPermission*(this, Manifest.permission.*POST\_NOTIFICATIONS*) == PackageManager.*PERMISSION\_GRANTED*) {  
 notificationManager.notify(noti\_id, builder.build());  
 }  
 }  
}

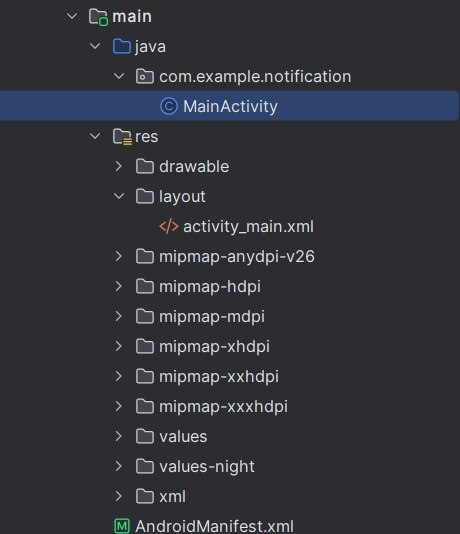
**Activity\_main.xml :**

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@color/white"  
 tools:context=".MainActivity">  
  
 <Button  
 android:id="@+id/btn"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerInParent="true"  
 android:text="Send Notification"  
 android:backgroundTint="@color/green"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>

**AndroidMainfest.xml:**

<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools">  
  
 <uses-permission android:name="android.permission.POST\_NOTIFICATIONS" />  
 <application  
 android:allowBackup="true"  
 android:dataExtractionRules="@xml/data\_extraction\_rules"  
 android:fullBackupContent="@xml/backup\_rules"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.Notification"  
 tools:targetApi="31">  
 <activity  
 android:name=".MainActivity"  
 android:exported="true">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity>  
 </application>  
  
</manifest>

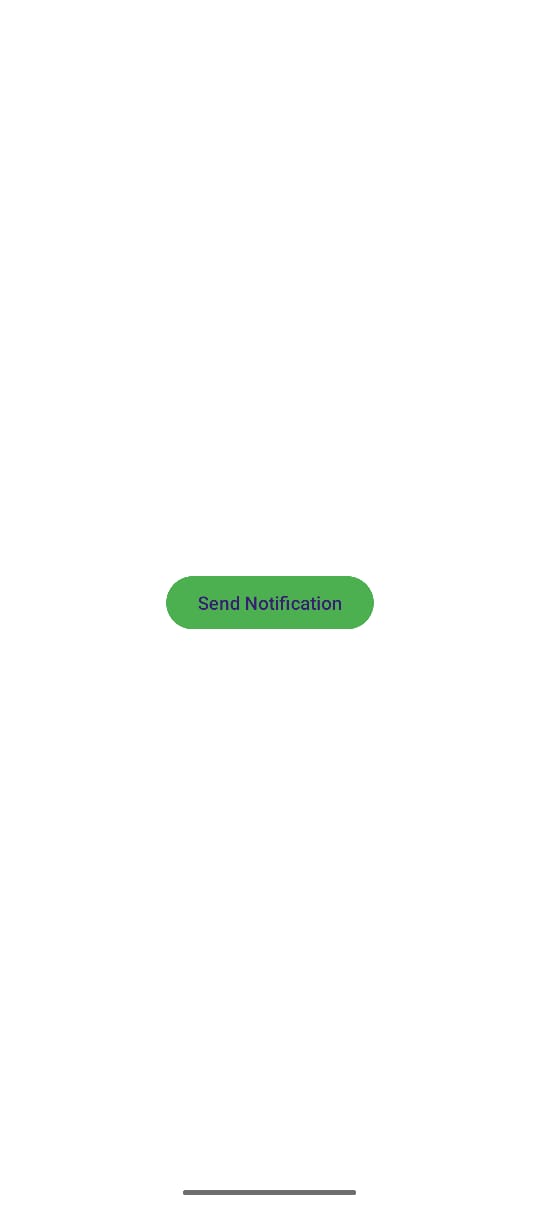
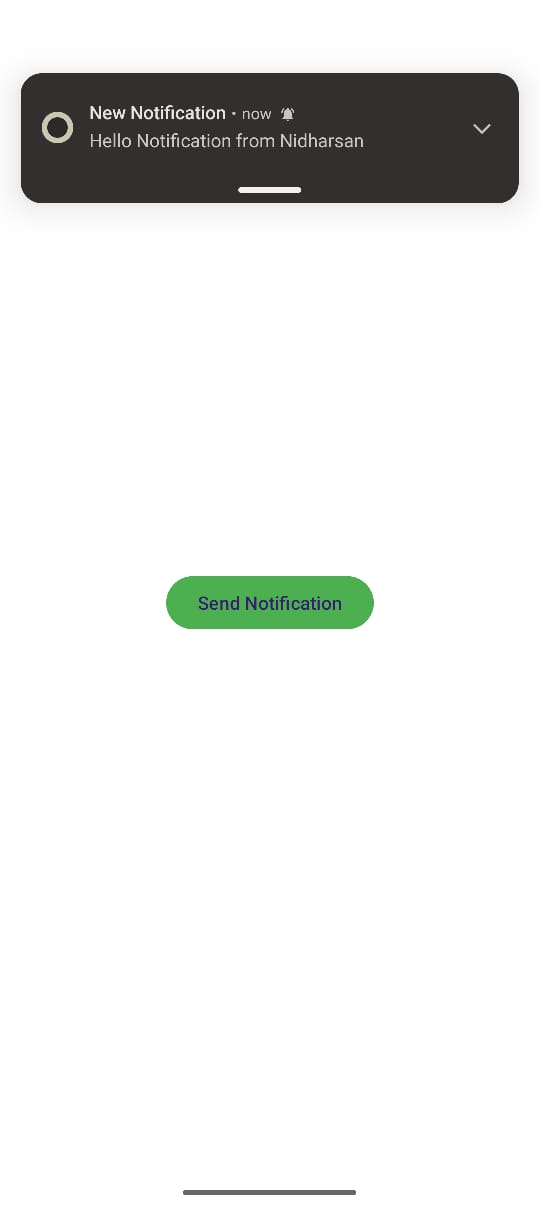
**FileStructure:**

****

**Project Link:**

<https://github.com/NIDHARSAN-V/Android/tree/master/Calling>

**Output:**

**6)** **Repeating Notification App with Dismiss Action:**

**Description:**

This Android application demonstrates how to send **periodic local notifications** with an actionable **dismiss button**, using NotificationManager, NotificationChannel, and BroadcastReceiver. The app uses a handler to trigger a notification every 5 seconds and allows users to dismiss the current notification directly.

**Repeating Notifications**  
The app triggers a new notification every 5 seconds using a Handler and Runnable, simulating timed alerts such as work-break reminders.

**Notification with Action**  
Each notification includes a **Dismiss** button. When tapped, it sends a broadcast via a PendingIntent, which is caught by a BroadcastReceiver (DismissReceiver) to cancel the notification.

**Runtime Permission Handling**  
For Android 13 (API 33) and above, the app checks and requests POST\_NOTIFICATIONS permission to ensure compliance with system requirements.

**Notification Channel (API 26+)**  
Although the channel creation is commented out in the provided code, the app is structured to support Android O+ behavior with a notification channel named break\_reminder\_channel.

**MainActivity.java:**

package com.example.notificationinterveldismiss;  
  
import android.Manifest;  
import android.app.Notification;  
import android.app.NotificationChannel;  
import android.app.NotificationManager;  
import android.app.PendingIntent;  
import android.content.BroadcastReceiver;  
import android.content.Context;  
import android.content.Intent;  
import android.content.pm.PackageManager;  
import android.os.Build;  
import android.os.Bundle;  
import android.os.Handler;  
import android.graphics.Color;  
  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.core.app.ActivityCompat;  
  
public class MainActivity extends AppCompatActivity {  
 public static final String *CHANNEL\_ID* = "break\_reminder\_channel";  
 private static final int *NOTIFICATION\_ID* = 1;  
 private final Handler handler = new Handler();  
 private final int INTERVAL = 5000; // 5 seconds  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
// createNotificationChannel();  
  
 // Request notification permission for Android 13+  
 if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*TIRAMISU*) {  
 if (ActivityCompat.*checkSelfPermission*(this, Manifest.permission.*POST\_NOTIFICATIONS*) != PackageManager.*PERMISSION\_GRANTED*) {  
 ActivityCompat.*requestPermissions*(this, new String[]{Manifest.permission.*POST\_NOTIFICATIONS*}, 101);  
 }  
 }  
  
 // Start the repeating notification  
 startRepeatingNotification();  
 }  
  
 private void startRepeatingNotification() {  
 handler.postDelayed(new Runnable() {  
 @Override  
 public void run() {  
 sendNotification(MainActivity.this);  
 handler.postDelayed(this, INTERVAL); // Repeat every 5 seconds  
 }  
 }, INTERVAL);  
 }  
  
  
  
 private void sendNotification(Context context) {  
 NotificationManager notificationManager = (NotificationManager) context.getSystemService(Context.*NOTIFICATION\_SERVICE*);  
  
 if (ActivityCompat.*checkSelfPermission*(context, Manifest.permission.*POST\_NOTIFICATIONS*) == PackageManager.*PERMISSION\_GRANTED*) {  
 if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*O*) {  
  
 // Create an intent for the dismiss action  
 Intent dismissIntent = new Intent(context, DismissReceiver.class);  
 PendingIntent dismissPendingIntent = PendingIntent.*getBroadcast*(  
 context, 0, dismissIntent, PendingIntent.*FLAG\_UPDATE\_CURRENT* | PendingIntent.*FLAG\_IMMUTABLE*);  
  
 // Build the notification with a dismiss button  
 Notification notification = new Notification.Builder(context, *CHANNEL\_ID*)  
 .setSmallIcon(R.mipmap.*ic\_launcher*)  
 .setContentTitle("Take a Break")  
 .setContentText("You've been working for a while. Take a short break!")  
 .setAutoCancel(true)  
 .addAction(android.R.drawable.*ic\_delete*, "Dismiss", dismissPendingIntent) // Dismiss Button  
 .build();  
  
 notificationManager.notify(*NOTIFICATION\_ID*, notification);  
 }  
 }  
 }  
  
  
 public static class DismissReceiver extends BroadcastReceiver {  
 @Override  
 public void onReceive(Context context, Intent intent) {  
 NotificationManager notificationManager = (NotificationManager) context.getSystemService(Context.*NOTIFICATION\_SERVICE*);  
 if (notificationManager != null) {  
 notificationManager.cancel(*NOTIFICATION\_ID*);  
 }  
 }  
 }  
}

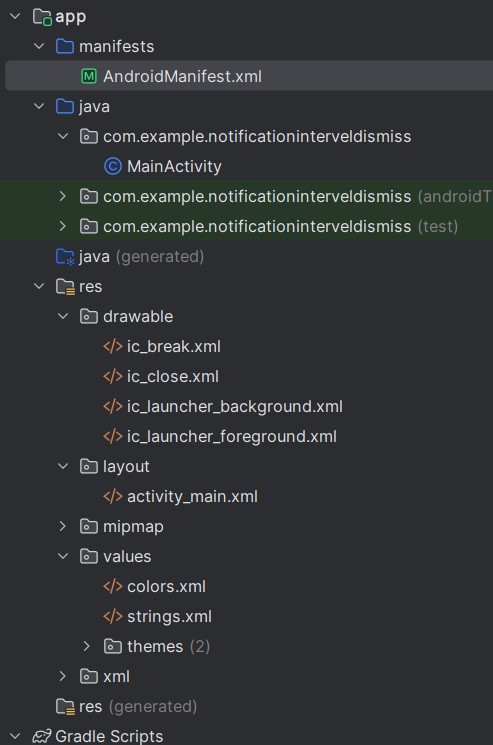
**Activity\_main.xml :**

<?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="20dp">  
  
 <TextView  
 android:id="@+id/textView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Break Reminder App"  
 android:textSize="22sp"  
 android:textStyle="bold"  
 android:textColor="#000000"/>  
  
 <TextView  
 android:id="@+id/textViewDesc"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="This app will remind you to take a break every 5 seconds."  
 android:textSize="16sp"  
 android:layout\_marginTop="10dp"/>  
</LinearLayout>

**AndroidMainfest.xml:**

<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools">  
  
 <!-- Permission required for notifications (Android 13+) -->  
 <uses-permission android:name="android.permission.POST\_NOTIFICATIONS" />  
  
 <application  
 android:allowBackup="true"  
 android:dataExtractionRules="@xml/data\_extraction\_rules"  
 android:fullBackupContent="@xml/backup\_rules"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.NotificationIntervelDismiss"  
 tools:targetApi="31">  
  
 <!-- Main Activity -->  
 <activity  
 android:name=".MainActivity"  
 android:exported="true">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity>  
  
 <!-- Broadcast Receiver for dismissing notification -->  
 <receiver android:name=".MainActivity$DismissReceiver"  
 android:exported="false"/>  
 </application>  
  
</manifest>

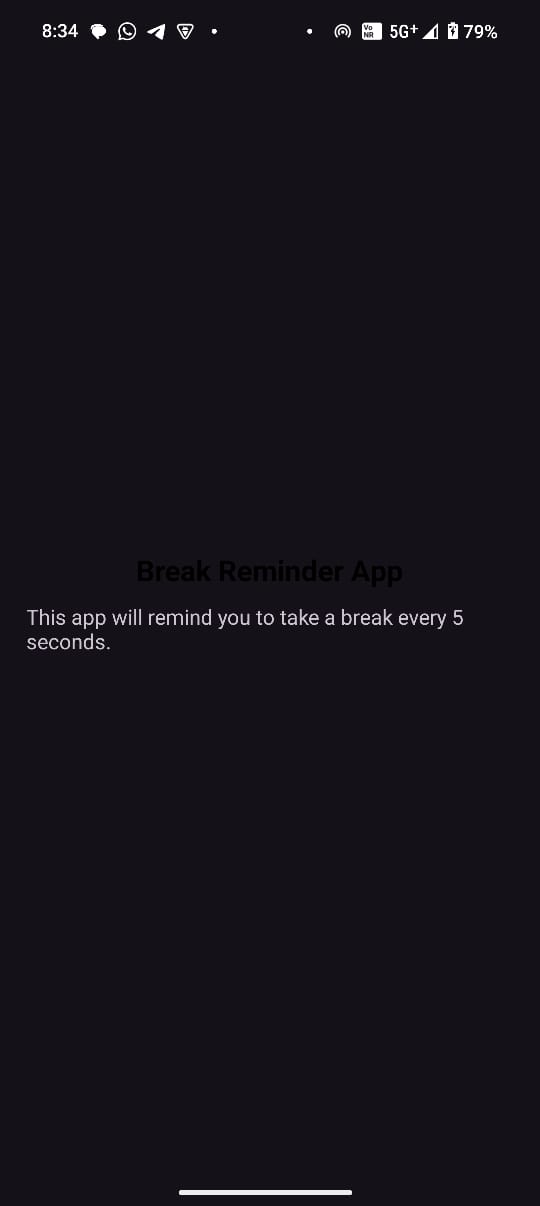
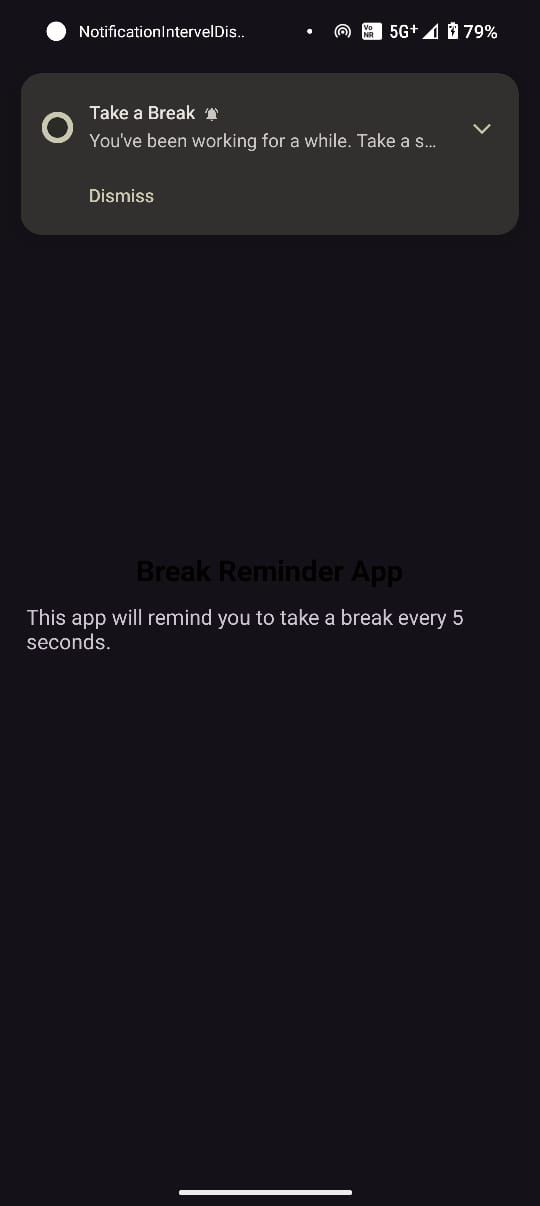
**FileStructure:**

****

**Project Link:**

<https://github.com/NIDHARSAN-V/Android/tree/master/NotificationBroadcast>

**Output:**

**7)** **URL Opener with WebView:**

**Description:**

This Android application allows users to input a URL and open the corresponding webpage inside the app using a WebView. It demonstrates the usage of editable text input, basic validation for URLs, and embedded browsing functionality within an Android app.

**Open URLs via WebView**  
Users can enter any URL, and the app will load the webpage directly within the app interface using a WebView.

**Automatic URL Correction**  
If the entered URL doesn’t start with http:// or https://, the app automatically prepends https:// to ensure it is loaded correctly.

**Embedded Browsing**  
A WebViewClient is used to ensure links stay within the app instead of launching an external browser, providing a seamless experience.

**JavaScript Support**  
JavaScript is enabled within the WebView settings, allowing modern web pages to function properly.

**MainActivity.java:**

package com.example.urlopen;  
  
import android.os.Bundle;  
import android.view.View;  
import android.webkit.WebSettings;  
import android.webkit.WebView;  
import android.webkit.WebViewClient;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.TextView;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
public class MainActivity extends AppCompatActivity {  
  
 EditText editText;  
 Button button;  
 TextView textView;  
 WebView webView;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 editText = findViewById(R.id.*urltxt*);  
 button = findViewById(R.id.*gotoURL*);  
 textView = findViewById(R.id.*textview*);  
 webView = findViewById(R.id.*webview*);  
  
 // Enable JavaScript  
 WebSettings webSettings = webView.getSettings();  
 webSettings.setJavaScriptEnabled(true);  
 webView.setWebViewClient(new WebViewClient()); // Open links inside WebView  
  
 button.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 String url = editText.getText().toString();  
  
 // Ensure a valid URL  
 if (!url.startsWith("http://") && !url.startsWith("https://")) {  
 url = "https://" + url; // Add HTTPS if missing  
 }  
  
 webView.setVisibility(View.*VISIBLE*);  
 webView.loadUrl(url); // Load URL inside WebView  
 }  
 });  
 }  
}

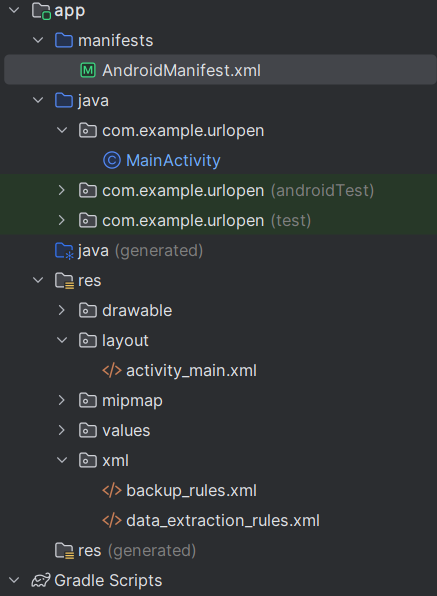
**Activity\_main.xml :**

<?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">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerHorizontal="true"  
 android:id="@+id/layout"  
 android:orientation="horizontal"  
 android:padding="10dp">  
  
 <TextView  
 android:textSize="20dp"  
 android:layout\_marginTop="5dp"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="https://www."  
 android:id="@+id/textview" />  
  
 <EditText  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:minHeight="48dp"  
 android:id="@+id/urltxt"  
 android:textColor="@color/black"  
 android:layout\_marginStart="10dp"  
 android:alpha="0.7"  
 android:textSize="20dp"  
 android:padding="10dp"  
 android:hint="Enter URL" />  
  
 </LinearLayout>  
  
 <Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Open in WebView"  
 android:layout\_below="@+id/layout"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="10dp"  
 android:id="@+id/gotoURL" />  
  
 <WebView  
 android:id="@+id/webview"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_below="@+id/gotoURL"  
 android:visibility="gone"/> <!-- Initially hidden -->  
</RelativeLayout>

**AndroidMainfest.xml:**

<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools">  
  
  
  
 <uses-permission android:name="android.permission.INTERNET"/>  
 <uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE"/>  
  
 <application  
 android:allowBackup="true"  
 android:dataExtractionRules="@xml/data\_extraction\_rules"  
 android:fullBackupContent="@xml/backup\_rules"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.UrlOpen"  
 tools:targetApi="31">  
  
  
  
  
 <activity  
 android:name=".MainActivity"  
 android:exported="true">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity>  
 </application>  
  
</manifest>

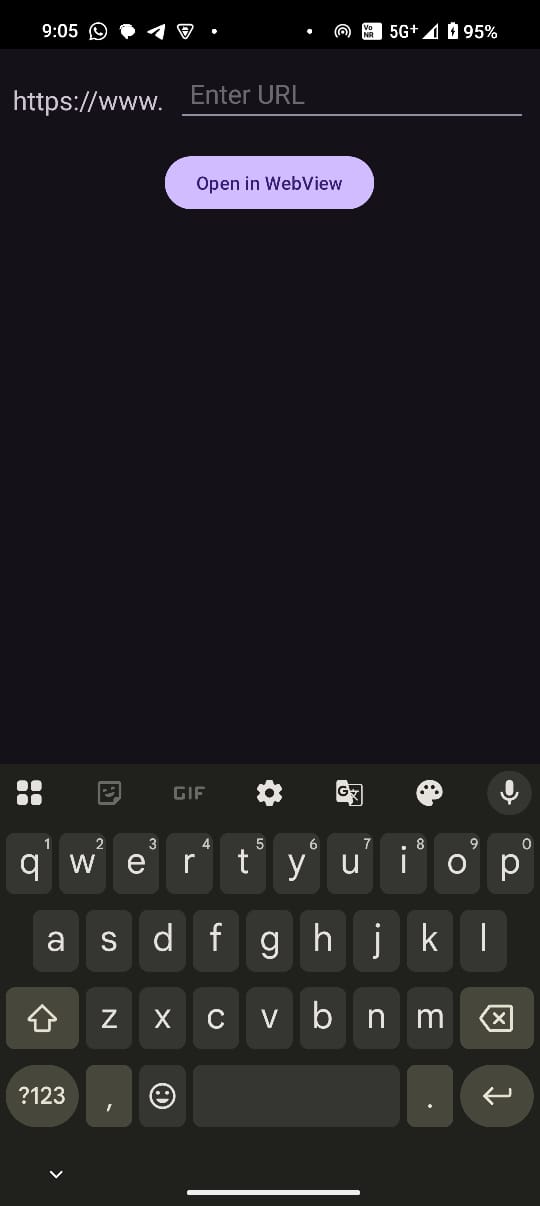
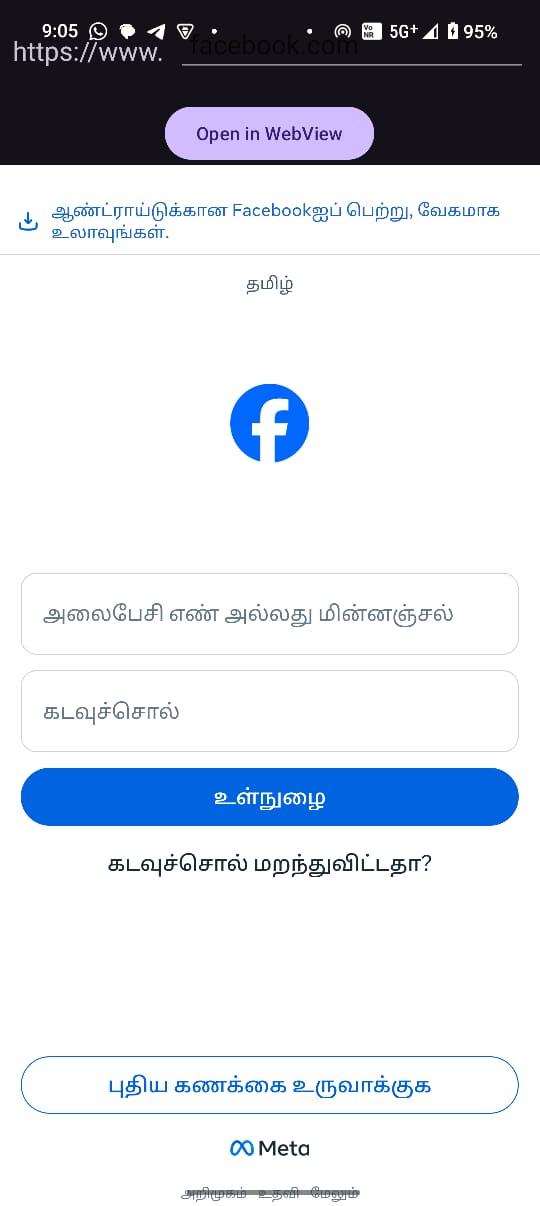
**FileStructure:**

****

**Project Link:**

<https://github.com/NIDHARSAN-V/Android/tree/master/UrlOpen>

**Output:**

**8)** **Android Notification App:**

**Description:**

**MainActivity.java:**

**Activity\_main.xml :**

**AndroidMainfest.xml:**

**FileStructure:**

**Project Link:**

<https://github.com/NIDHARSAN-V/Android/tree/master/Calling>

**Output:**