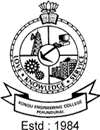
** KONGU ENGINEERING COLLEGE**

**(AUTONOMOUS)**

**PERUNDURAI – 638060**

**DEPARTMENT OF INFORMATION TECHNOLOGY**

**TO-DO LIST ANDROID BASED APPLICATION USING ANDROID STUDIO**

### KARTHIKEYAN R

**22ITR043**

### NIRAJANA SRI S

**22ITR070**

### IMMANUVEL R

**22ITR035**

**GUIDED BY**

****

**ABSTRACT**

### ANDROID BASED TO-DO LIST APPLICATION

This project showcases the successful creation of a comprehensive To-Do List application using Android Studio, catering to the specific needs of users seeking a streamlined task management solution. The application encompasses a suite of functionalities, including task creation, editing, and deletion, providing users with a versatile platform to manage their daily responsibilities.The development process involved harnessing the capabilities of Android Studio, the preferred integrated development environment (IDE) for Android applications.

The app adheres to Material Design principles, ensuring an aesthetically pleasing and intuitive user interface. Leveraging the power of Android Studio, the development environment was finely tuned to deliver a responsive and visually appealing application.A critical aspect of the app is its integration with SQLite, a robust and efficient relational database management system. This integration facilitates seamless data storage and retrieval, allowing users to store their tasks securely and ensuring consistent information across multiple devices.

The app provides users with the ability to mark tasks as completed, introducing a visual cue for tracking progress and enhancing user engagement.The project emphasizes the importance of meeting evolving user demands, and the To-Do List app is positioned as a valuable organizational tool designed to adapt to the dynamic requirements of modern life. The app's user-friendly interface, coupled with the robust functionality enabled by Android Studio, ensures a reliable and efficient task management experience for users, ultimately contributing to enhanced productivity and time management in their daily lives.

TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| S. No: | TITLE | Page. No: |

|  |  |  |
| --- | --- | --- |
| 1. | ABSTRACT | ii |
| 2. | INTRODUCTION | 1 |
| 3. | OBJECTIVES | 1 |
| 4. | SYSTEM SPECIFICATION  4.1 HARDWARE SPECIFICATION  4.2 SOFTWARE SPECIFICATION | 2 |
| 5. | CODE | 3 |
| 6. | MODULE DESCRIPTION | 14 |
| 7. | SCREENSHOTS | 15 |
| 8. | CONCLUSION | 22 |
| 9. | REFERENCE | 23 |

### 1. INTRODUCTION

**1.1 A PREVIEW**

In an era marked by increasing demands on our time and attention, effective task management has become an indispensable aspect of our daily lives. The advent of mobile applications has revolutionized the way we organize and execute our responsibilities. This project introduces a sophisticated solution to the challenge of efficient task management: a To-Do List app meticulously developed using the versatile Android Studio.With a focus on user-centric design and robust functionality, this app is tailored to enhance the productivity of individuals navigating the complexities of modern life. Offering a seamless blend of intuitive features such as task creation, deletion, and editing, coupled with a powerful database integration, our To-Do List app aspires to be a cornerstone in the realm of digital task management

### OBJECTIVES

The objectives of this project are as follows:

* User-Friendly Interface: Develop an intuitive, visually appealing interface that simplifies navigation, ensuring an accessible and user-friendly experience for individuals with varying levels of technological proficiency.
* Comprehensive Task Functionality: Implement robust task management features, including creation, deletion, and editing, providing users with a dynamic platform to efficiently organize and modify their to-do lists.
* Database Integration: Utilize SQLite to establish an efficient database system, enabling seamless storage and retrieval of task data. This integration ensures data consistency and synchronization across multiple devices for a unified user experience.
* Material Design Principles: Adhere to Material Design principles to create a cohesive and aesthetically pleasing app interface. This not only enhances visual appeal but also contributes to a consistent and intuitive user interaction model.
* Cross-Device Synchronization: Implement a synchronization mechanism enabling users to access and manage tasks across different devices. This ensures real-time updates, promoting a seamless and integrated experience regardless of the user's preferred device.
* Task Priority and Categorization: Integrate features for setting task priorities and categorizing tasks, empowering users to organize their to-do lists effectively. This functionality adds a layer of customization.

**3. SYSTEM SPECIFICATIONS**

**3.1.1 HARDWARE SPECIFICATIONS**

Processor Type : AMD RYZEN 5

RAM : 8 GB RAM

HARD DISK : 512 GB SSD

**3.1.2 SOFTWARE REQUIREMENTS:**

OPERATING SYSTEM : WINDOWS 11

FRONT END : ANDROID STUDIO

LANGUAGE : JAVA

DATABASE : SQLite

**3.2 SOFTWARE REQUIREMENTS:**

ANDROID STUDIO:

Android Studio stands as the authoritative integrated development environment (IDE) for the creation of Android applications. Supporting both Java and Kotlin, it accommodates developers' language preferences, offering flexibility in the choice of programming tools. The IDE features a visual layout editor, facilitating the design of user interfaces through an intuitive drag-and-drop interface, easing the complexities of manual XML coding. Seamless integration with the Android Software Development Kit (SDK) ensures access to a comprehensive set of tools, libraries, and APIs, keeping developers abreast of the latest features for building robust applications.

1. **CODING**

MainActivity.java

package com.example.todoapp;

import androidx.appcompat.app.AppCompatActivity;

import androidx.recyclerview.widget.ItemTouchHelper;

import androidx.recyclerview.widget.LinearLayoutManager;

import androidx.recyclerview.widget.RecyclerView;

import android.content.DialogInterface;

import android.os.Bundle;

import android.view.View;

import com.example.todoapp.Adapter.ToDoAdapter;

import com.example.todoapp.Model.ToDoModel;

import com.example.todoapp.Utils.DataBaseHelper;

import com.google.android.material.floatingactionbutton.FloatingActionButton;

import java.util.ArrayList;

import java.util.Collections;

import java.util.List;

public class MainActivity extends AppCompatActivity implements OnDialogCloseListner {

private RecyclerView mRecyclerview;

private FloatingActionButton fab;

private DataBaseHelper myDB;

private List<ToDoModel> mList;

private ToDoAdapter adapter;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

mRecyclerview = findViewById(R.id.recyclerview);

fab = findViewById(R.id.fab);

myDB = new DataBaseHelper(MainActivity.this);

mList = new ArrayList<>();

adapter = new ToDoAdapter(myDB , MainActivity.this);

mRecyclerview.setHasFixedSize(true);

mRecyclerview.setLayoutManager(new LinearLayoutManager(this));

mRecyclerview.setAdapter(adapter);

mList = myDB.getAllTasks();

Collections.reverse(mList);

adapter.setTasks(mList);

fab.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

AddNewTask.newInstance().show(getSupportFragmentManager() , AddNewTask.TAG);

}

});

ItemTouchHelper itemTouchHelper = new ItemTouchHelper(new RecyclerViewTouchHelper(adapter));

itemTouchHelper.attachToRecyclerView(mRecyclerview);

}

@Override

public void onDialogClose(DialogInterface dialogInterface) {

mList = myDB.getAllTasks();

Collections.reverse(mList);

adapter.setTasks(mList);

adapter.notifyDataSetChanged();

}

}

activity\_main.xml

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

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

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

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

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".MainActivity">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:id="@+id/textview"

android:text="To Do Tasks"

android:textSize="32sp"

android:textColor="#000"

android:textStyle="bold"

android:layout\_centerHorizontal="true"

android:layout\_marginTop="10dp"/>

<androidx.recyclerview.widget.RecyclerView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:id="@+id/recyclerview"

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

android:nestedScrollingEnabled="true"/>

<com.google.android.material.floatingactionbutton.FloatingActionButton

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentEnd="true"

android:layout\_alignParentBottom="true"

android:id="@+id/fab"

android:src="@drawable/ic\_baseline\_add\_24"

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

</RelativeLayout>

Activity\_splash.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/colorPrimaryDark"

tools:context=".SplashActivity">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.5"

app:layout\_constraintStart\_toStartOf="parent"

android:text="To Do App"

android:textColor="@color/colorWhite"

android:textSize="40sp"

android:textStyle="bold"

android:drawablePadding="15dp"

android:drawableRight="@drawable/ic\_baseline\_done\_all\_24"

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

<TextView

android:id="@+id/textView"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="80dp"

android:text="Designed and developed by Karthikeyan"

android:textColor="#FFFFFF"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.49"

app:layout\_constraintStart\_toStartOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

SplashActivity.java

package com.example.todoapp;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;

import android.os.Bundle;

import android.os.Handler;

public class SplashActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_splash);

Handler handler = new Handler();

handler.postDelayed(new Runnable() {

@Override

public void run() {

startActivity(new Intent(SplashActivity.this , MainActivity.class));

finish();

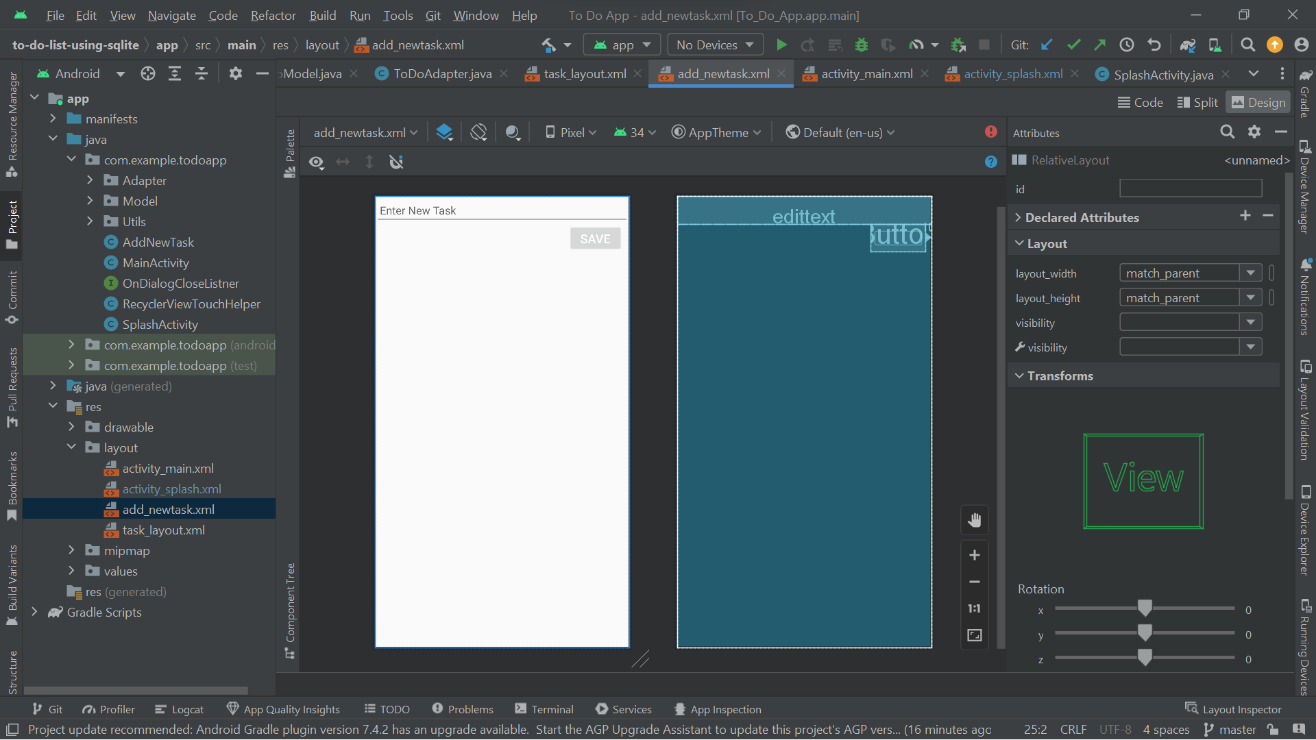
}

} , 4000);

}

}

add\_newtask.xml



ic\_baseline\_add\_24.xml

<vector android:height="24dp" android:tint="#FFFFFF"

android:viewportHeight="24" android:viewportWidth="24"

android:width="24dp" xmlns:android="http://schemas.android.com/apk/res/android">

<path android:fillColor="@android:color/white" android:pathData="M19,13h-6v6h-2v-6H5v-2h6V5h2v6h6v2z"/>

</vector>



ic\_baseline\_delete.xml

<vector android:height="24dp" android:tint="#FFFFFF"

android:viewportHeight="24" android:viewportWidth="24"

android:width="24dp" xmlns:android="http://schemas.android.com/apk/res/android">

<path android:fillColor="@android:color/white" android:pathData="M6,19c0,1.1 0.9,2 2,2h8c1.1,0 2,-0.9 2,-2V7H6v12zM19,4h-3.5l-1,-1h-5l-1,1H5v2h14V4z"/>

</vector>



ic\_baseline\_done\_all\_24.xml

<vector android:height="55dp" android:tint="#FFFFFF"

android:viewportHeight="24" android:viewportWidth="24"

android:width="55dp" xmlns:android="http://schemas.android.com/apk/res/android">

<path android:fillColor="@android:color/white" android:pathData="M18,7l-1.41,-1.41 -6.34,6.34 1.41,1.41L18,7zM22.24,5.59L11.66,16.17 7.48,12l-1.41,1.41L11.66,19l12,-12 -1.42,-1.41zM0.41,13.41L6,19l1.41,-1.41L1.83,12 0.41,13.41z"/>

</vector>



ic\_baseline\_edit.xml

<vector android:height="24dp" android:tint="#FFFFFF"

android:viewportHeight="24" android:viewportWidth="24"

android:width="24dp" xmlns:android="http://schemas.android.com/apk/res/android">

<path android:fillColor="@android:color/white" android:pathData="M3,17.25V21h3.75L17.81,9.94l-3.75,-3.75L3,17.25zM20.71,7.04c0.39,-0.39 0.39,-1.02 0,-1.41l-2.34,-2.34c-0.39,-0.39 -1.02,-0.39 -1.41,0l-1.83,1.83 3.75,3.75 1.83,-1.83z"/>

</vector>



RecyclerViewTouchHelper.java

package com.example.todoapp;

import android.app.AlertDialog;

import android.content.DialogInterface;

import android.graphics.Canvas;

import android.graphics.Color;

import androidx.annotation.NonNull;

import androidx.core.content.ContextCompat;

import androidx.recyclerview.widget.ItemTouchHelper;

import androidx.recyclerview.widget.RecyclerView;

import com.example.todoapp.Adapter.ToDoAdapter;

import it.xabaras.android.recyclerview.swipedecorator.RecyclerViewSwipeDecorator;

public class RecyclerViewTouchHelper extends ItemTouchHelper.SimpleCallback {

private ToDoAdapter adapter;

public RecyclerViewTouchHelper(ToDoAdapter adapter) {

super(0, ItemTouchHelper.LEFT | ItemTouchHelper.RIGHT);

this.adapter = adapter;

}

@Override

public boolean onMove(@NonNull RecyclerView recyclerView, @NonNull RecyclerView.ViewHolder viewHolder, @NonNull RecyclerView.ViewHolder target) {

return false;

}

@Override

public void onSwiped(@NonNull RecyclerView.ViewHolder viewHolder, int direction) {

final int position = viewHolder.getAdapterPosition();

if (direction == ItemTouchHelper.RIGHT){

AlertDialog.Builder builder = new AlertDialog.Builder(adapter.getContext());

builder.setTitle("Delete Task");

builder.setMessage("Are You Sure ?");

builder.setPositiveButton("Yes", new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

adapter.deletTask(position);

}

});

builder.setNegativeButton("Cancel", new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

adapter.notifyItemChanged(position);

}

});

AlertDialog dialog = builder.create();

dialog.show();

}else{

adapter.editItem(position);

}

}

@Override

public void onChildDraw(@NonNull Canvas c, @NonNull RecyclerView recyclerView, @NonNull RecyclerView.ViewHolder viewHolder, float dX, float dY, int actionState, boolean isCurrentlyActive) {

new RecyclerViewSwipeDecorator.Builder(c, recyclerView, viewHolder, dX, dY, actionState, isCurrentlyActive)

.addSwipeLeftBackgroundColor(ContextCompat.getColor(adapter.getContext() , R.color.colorPrimaryDark))

.addSwipeLeftActionIcon(R.drawable.ic\_baseline\_edit)

.addSwipeRightBackgroundColor(Color.RED)

.addSwipeRightActionIcon(R.drawable.ic\_baseline\_delete)

.create()

.decorate();

super.onChildDraw(c, recyclerView, viewHolder, dX, dY, actionState, isCurrentlyActive);

}

}

AddNewTask.java

package com.example.todoapp;

import android.app.Activity;

import android.content.DialogInterface;

import android.graphics.Color;

import android.os.Bundle;

import android.text.Editable;

import android.text.TextWatcher;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.Button;

import android.widget.EditText;

import androidx.annotation.NonNull;

import androidx.annotation.Nullable;

import com.example.todoapp.Model.ToDoModel;

import com.example.todoapp.Utils.DataBaseHelper;

import com.google.android.material.bottomsheet.BottomSheetDialogFragment;

public class AddNewTask extends BottomSheetDialogFragment {

public static final String TAG = "AddNewTask";

//widgets

private EditText mEditText;

private Button mSaveButton;

private DataBaseHelper myDb;

public static AddNewTask newInstance(){

return new AddNewTask();

}

@Nullable

@Override

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

View v = inflater.inflate(R.layout.add\_newtask , container , false);

return v;

}

@Override

public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState) {

super.onViewCreated(view, savedInstanceState);

mEditText = view.findViewById(R.id.edittext);

mSaveButton = view.findViewById(R.id.button\_save);

myDb = new DataBaseHelper(getActivity());

boolean isUpdate = false;

final Bundle bundle = getArguments();

if (bundle != null){

isUpdate = true;

String task = bundle.getString("task");

mEditText.setText(task);

if (task.length() > 0 ){

mSaveButton.setEnabled(false);

}

}

mEditText.addTextChangedListener(new TextWatcher() {

@Override

public void beforeTextChanged(CharSequence s, int start, int count, int after) {

}

@Override

public void onTextChanged(CharSequence s, int start, int before, int count) {

if (s.toString().equals("")){

mSaveButton.setEnabled(false);

mSaveButton.setBackgroundColor(Color.GRAY);

}else{

mSaveButton.setEnabled(true);

mSaveButton.setBackgroundColor(getResources().getColor(R.color.colorPrimary));

}

}

@Override

public void afterTextChanged(Editable s) {

}

});

final boolean finalIsUpdate = isUpdate;

mSaveButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

String text = mEditText.getText().toString();

if (finalIsUpdate){

myDb.updateTask(bundle.getInt("id") , text);

}else{

ToDoModel item = new ToDoModel();

item.setTask(text);

item.setStatus(0);

myDb.insertTask(item);

}

dismiss();

}

});

}

@Override

public void onDismiss(@NonNull DialogInterface dialog) {

super.onDismiss(dialog);

Activity activity = getActivity();

if (activity instanceof OnDialogCloseListner){

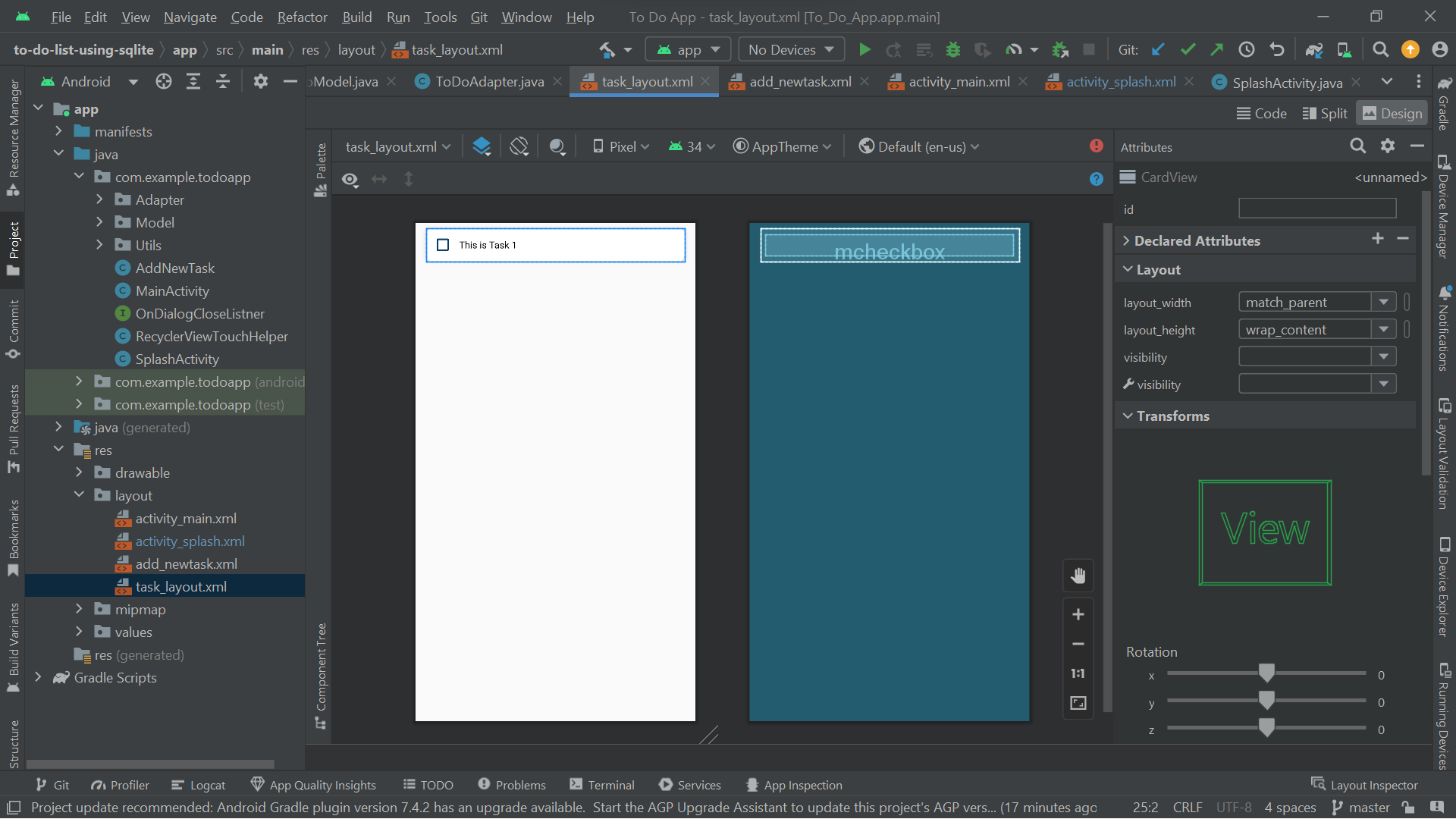
((OnDialogCloseListner)activity).onDialogClose(dialog);

}

}

}

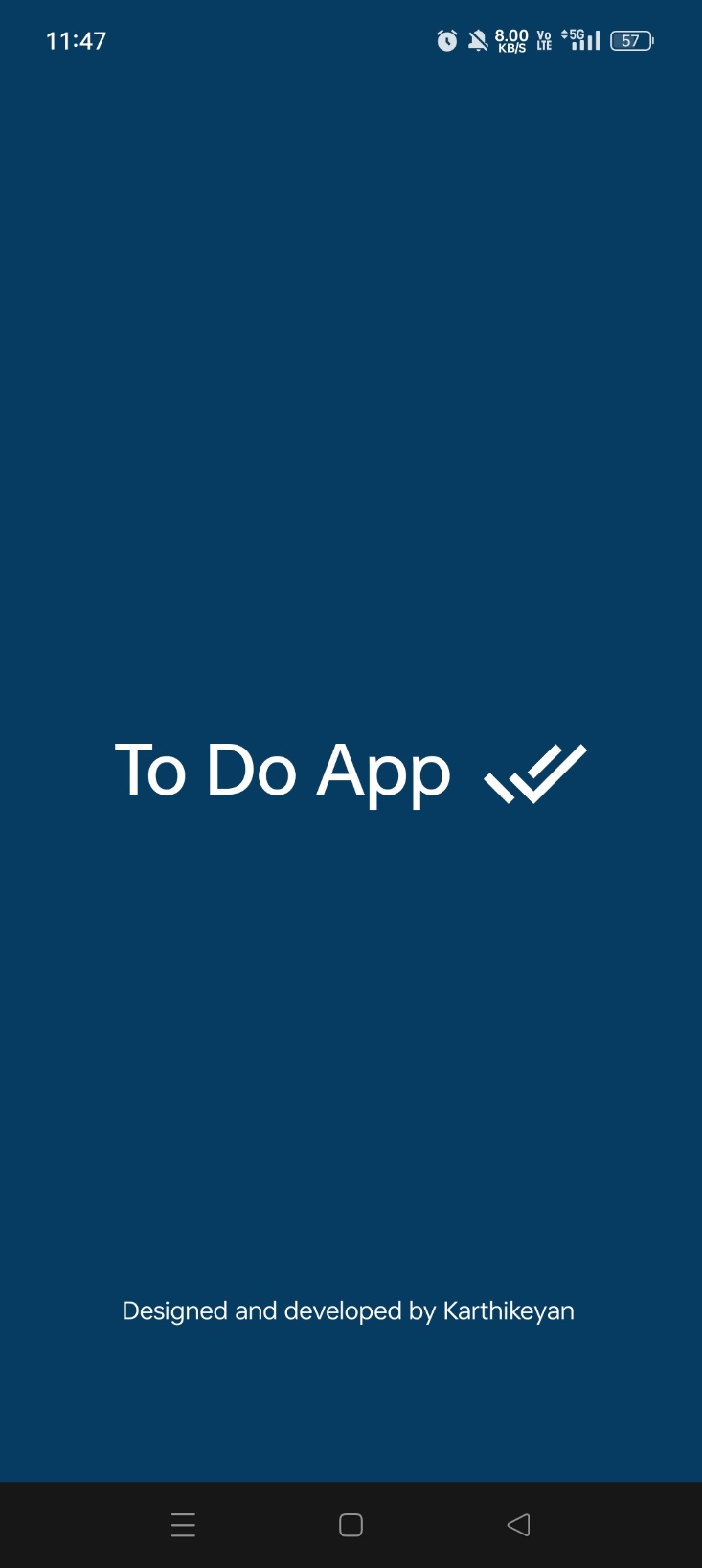
task\_layout.xml



**5. MODULE DESCRIPTION**

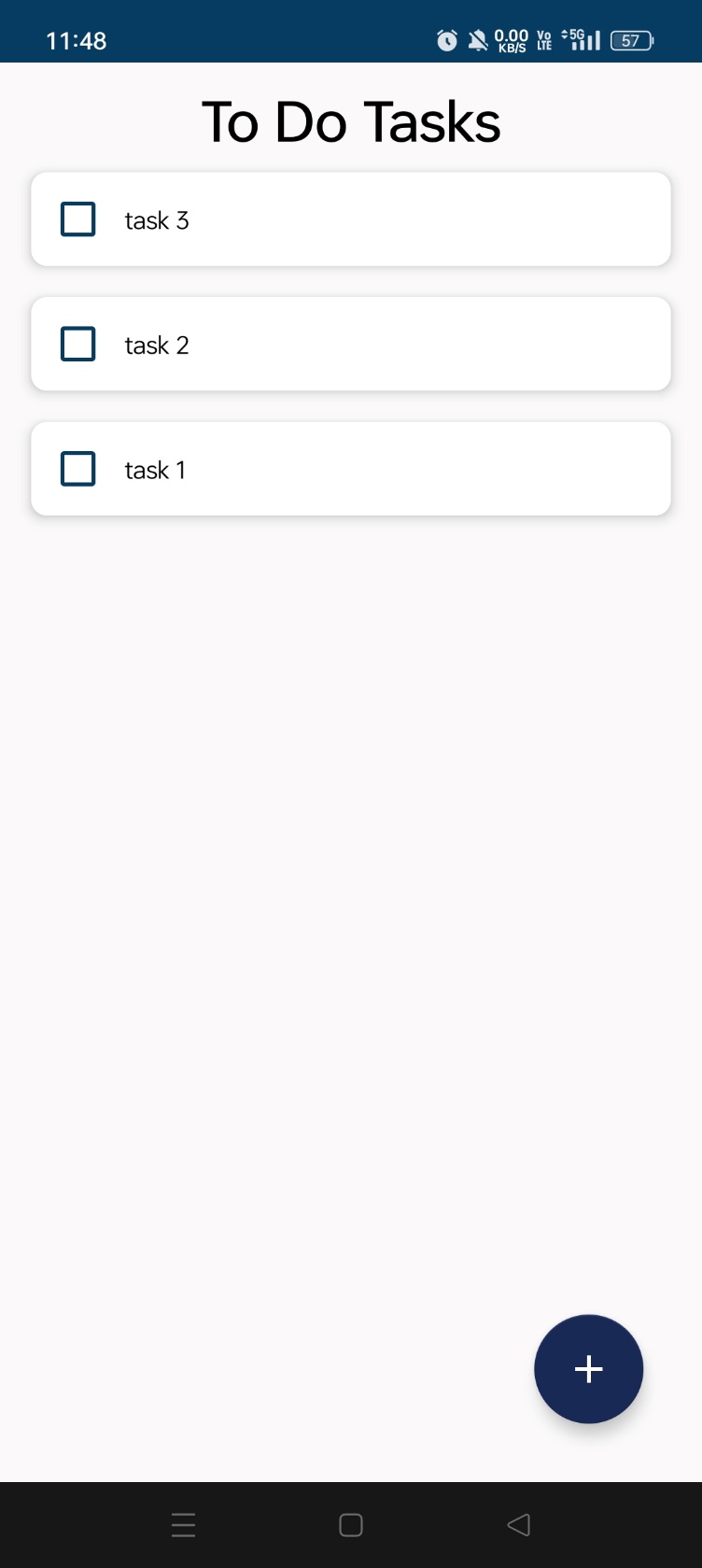
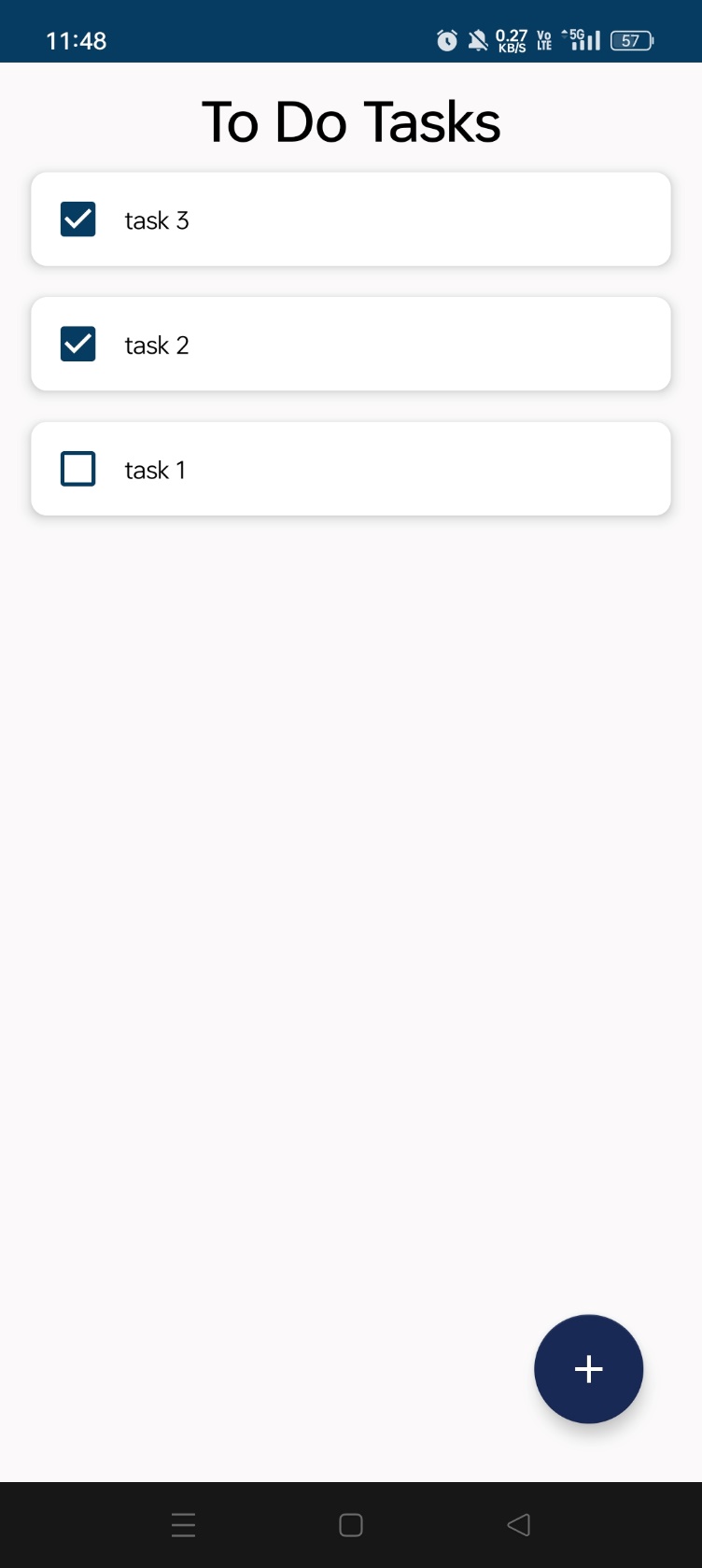
The to-do list app simplifies task management with a user-friendly interface that encompasses key functionalities. Users initiate the process by effortlessly creating new tasks, entering titles. As tasks are completed, a straightforward checkbox mechanism allows users to mark them as done, visually signifying their completion. Conversely, users can uncheck tasks to reactivate them if necessary. The app supports task modification through an intuitive editing feature, enabling users to update titles, or other details as needed. When tasks become obsolete, a straightforward deletion process, often involving a delete button or gesture, ensures seamless removal from the list. This streamlined functionality empowers users to efficiently manage their to-do lists, providing a digital space where tasks can be added, completed, modified, and removed with ease.

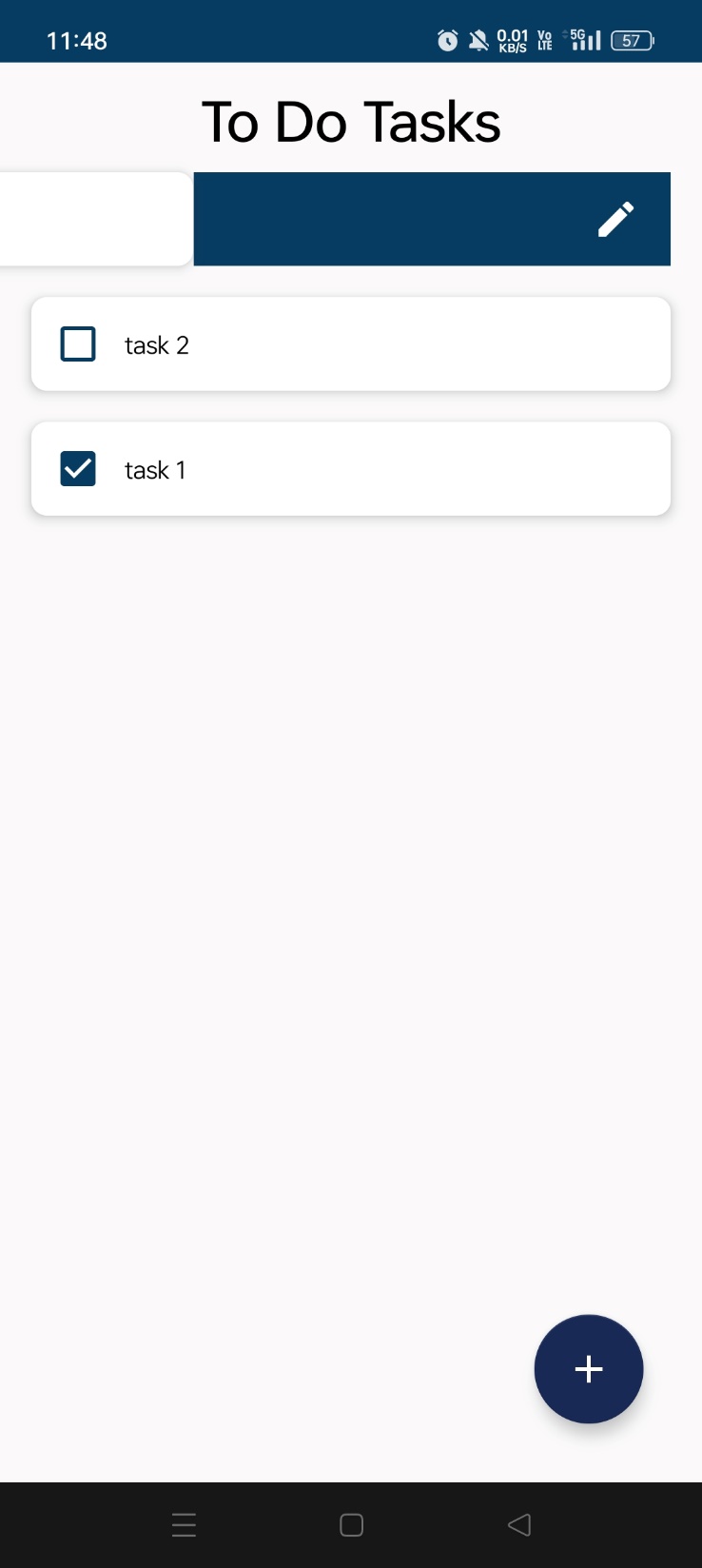
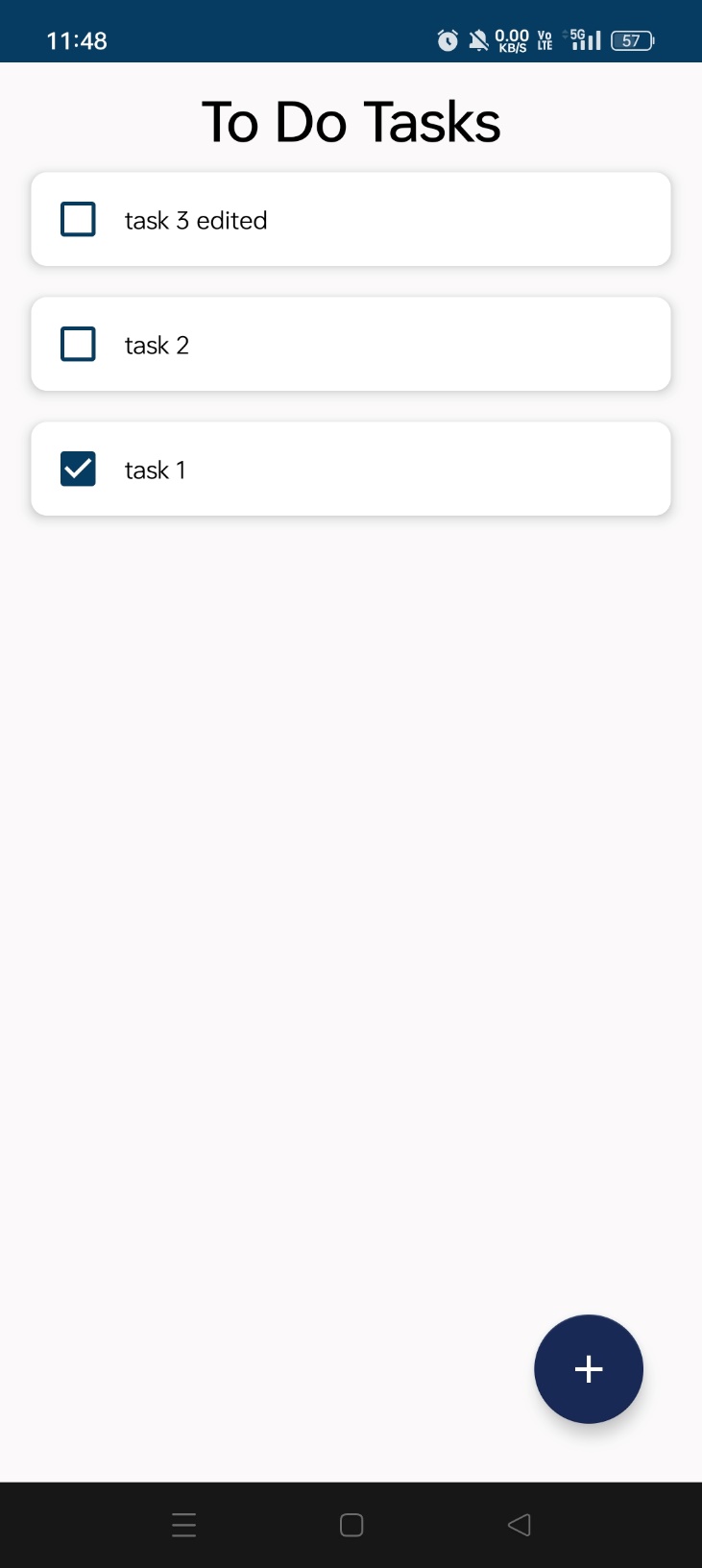
1. **SCREENSHOTS**

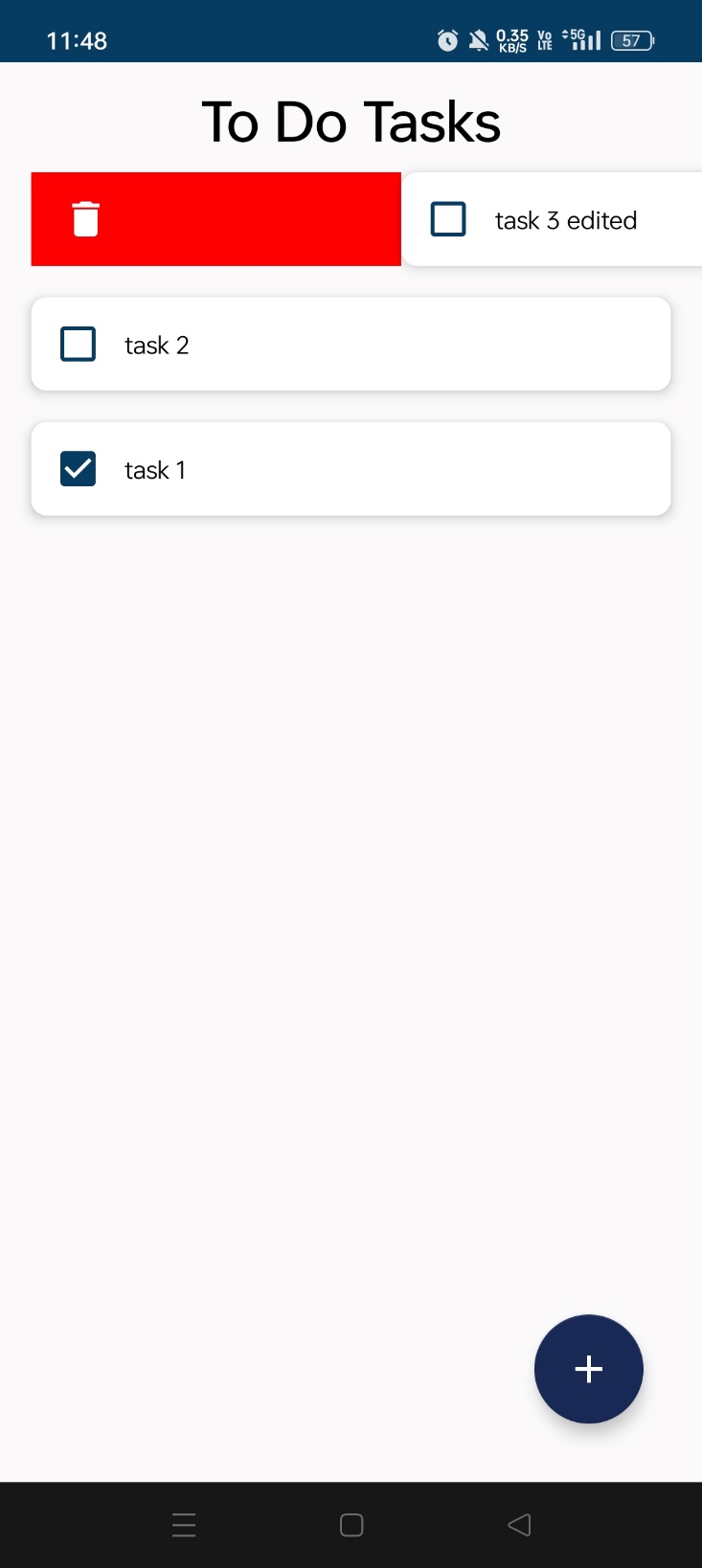
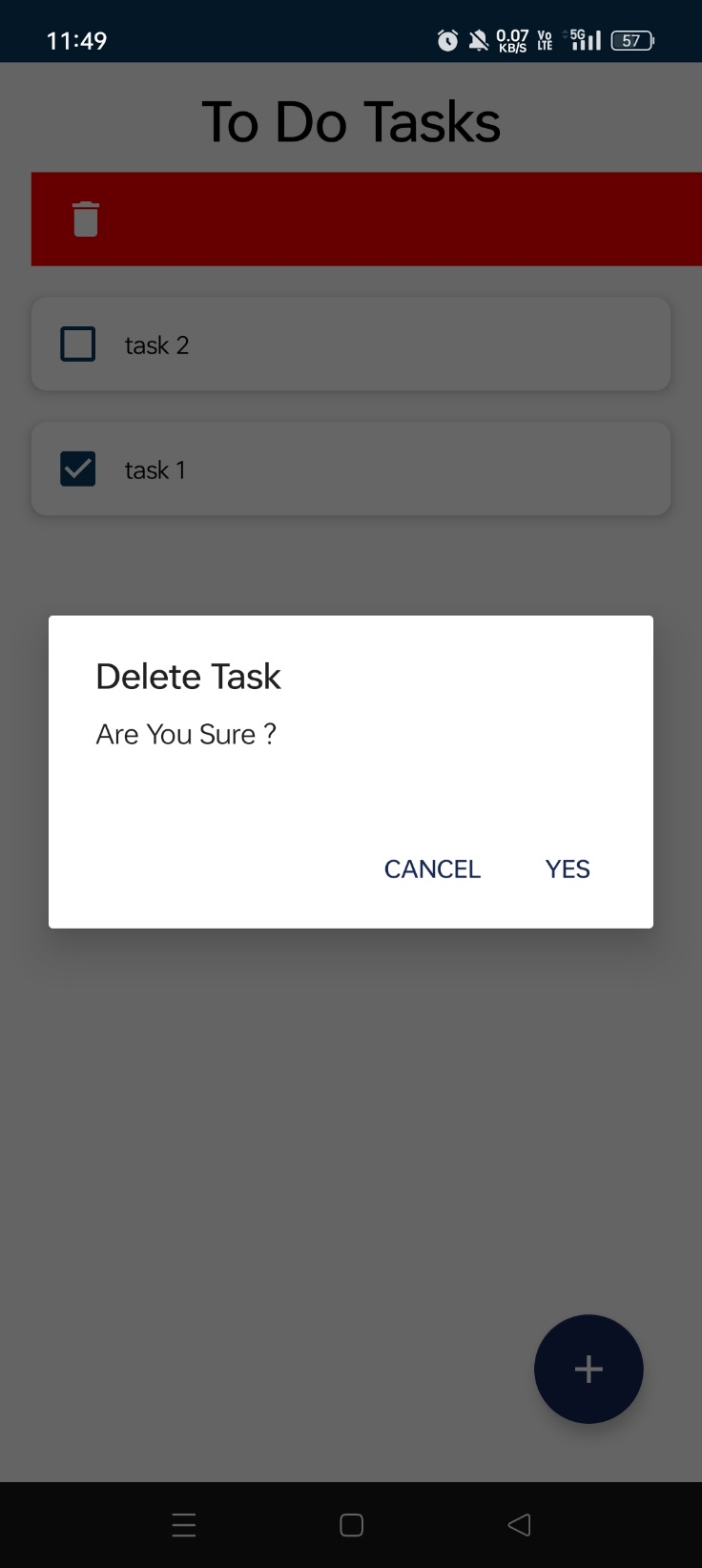
** **



1. **CONCLUSION**

In conclusion, the to-do list app represents a highly effective and user-centric solution for streamlined task management. With its intuitive interface, users can effortlessly create, modify, complete, and delete tasks, providing a digital workspace that adapts seamlessly to their dynamic schedules. The app's emphasis on simplicity and functionality not only enhances productivity but also fosters an organized approach to daily activities. By incorporating features such as checking and unchecking tasks, editing capabilities, and efficient deletion processes, the app ensures a user-friendly experience that aligns with the diverse needs of individuals seeking a reliable tool for effective task management. As we navigate an increasingly digital and fast-paced world, the to-do list app stands as a valuable ally in helping users stay organized, focused, and in control of their daily responsibilities.

1. **REFERENCES:**
   1. Android Developer Documentation. (n.d.). Retrieved from <https://developer.android.com/docs>
   2. Material Design Guidelines. (n.d.). Retrieved from <https://material.io/design>
   3. <https://youtu.be/9Qlvrvu5Kk8?si=Sz0vWmjM6Yfiy0GK>