A MINI PROJECT REPORT

Submitted by

MADHAN RAJ P **2207011148**

In partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE

(AUTONOMOUS) THANDALAM CHENNAI-602105

MAY-2025

BONAFIDE CERTIFICATE

Certified that this project report "**EXPENSE TRACKER**" is the bonafide work of **MADHAN RAJ P(220701148)** who carried out the project work under my supervision.

Submitted for the Practical Examination held on

SIGNATURE

Mr. V.Karthick
Associate professor
Computer Science and Engineering,
Rajalakshmi Engineering College,
(Autonomous),
Thandalam, Chennai - 602 105

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ABSTRACT

The **Expense Tracker** Android application is a personal finance management tool developed using **Android Studio** with **Kotlin**. It enables users to record, manage, and monitor their daily expenses effortlessly. Users can add expense entries by providing details like the amount, category, date, and a brief description. The app maintains a clear list of all transactions, allowing users to keep track of their spending habits over time. Designed with a clean and intuitive user interface, the application ensures ease of use for users of all ages. The data is stored locally, making the app usable even without an internet connection. It aims to promote financial awareness by helping users identify unnecessary expenses and improve their budgeting skills. The project showcases important Android development concepts such as activity lifecycle, RecyclerView handling, state management, and local data persistence. This Expense Tracker app serves as an excellent beginner project for understanding mobile application architecture, user interaction, and data-driven app development.

TABLE OF CONTENTS

CHAPTER 1

1. INTRODUCTION:

Managing personal finances is an essential part of everyday life, yet many individuals struggle to keep track of their daily expenses. The **Expense Tracker** Android application is developed to address this need by providing a simple, efficient, and accessible platform to monitor and record financial transactions. Built using **Kotlin** in **Android Studio**, this application allows users to add, view, and organize their expenses systematically. With a clean and user-friendly interface, the app focuses on enhancing user experience while ensuring essential functionalities like adding expenses with details such as amount, category, and description. By maintaining a detailed log of all transactions, users can analyze their spending habits, make informed financial decisions, and cultivate better budgeting practices. The application functions offline, offering convenience and privacy without relying on internet connectivity. This project serves as a foundational exercise in understanding Android app development, UI design, and local data management.

1.1 IMPLEMENTATION:

The **Expense Tracker** application was implemented using **Kotlin** programming language in **Android Studio**. The design follows a simple architecture consisting of activities and local storage. The main screen displays a list of all recorded expenses using a **RecyclerView**, where each expense includes fields such as the amount, category, date, and description.

A floating action button (FAB) is provided to allow users to quickly add a new expense. When clicked, it navigates to a form where users can input their expense details. Once submitted, the data is saved locally using either in-memory data storage or lightweight storage techniques like **SharedPreferences** or SQLite (if extended). UI elements such as **EditText**, **Button**, and **TextView** are used for input and display purposes, following Android's material design principles. All data updates are dynamically reflected in the RecyclerView. The overall implementation emphasizes modular code structure, intuitive navigation between activities, and real-time updates to enhance the user experience.

CHAPTER 2

SYSTEM SPECIFICATIONS

2.1 HARDWARE SPECIFICATION:

PROCESSOR - Intel® core™ i5-6006U @ 2.00 GHz

RAM - 4GB

OPERATING SYSTEM - Microsoft Windows 11

HARD DISK - 850 GB of free space

SYSTEM TYPE - 64-bit operating system , x64 based processor

2.2 SOFTWARE SPECIFICATION:

PROGRAMMING LANGUAGE: kotlin

OPERATING SYSTEM : Microsoft

Windows 11 SOFTWARE : Android Studio

CHAPTER 3

Source Code

AndroidManifest.xml

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

   <application
       android:allowBackup="true"
       android:dataExtractionRules="@xml/data_extraction_rules"
       android:fullBackupContent="@xml/backup_rules"
       android:icon="@mipmap/ic_launcher"
       android:label="@string/app_name"
       android:roundIcon="@mipmap/ic_launcher_round"
       android:supportsRtl="true"
       android:theme="@style/Theme.ExpenseTracker"
       tools:targetApi="31">
       <activity
           android:name=".MainActivity"
           android:exported="true"
           android:label="@string/app_name"
           android:theme="@style/Theme.ExpenseTracker">
           <intent-filter>
               <action android:name="android.intent.action.MAIN" />

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

</manifest>
```

MainActivity.kt

```
package com.example.expensetracker // Make sure this matches your
actual package name

import android.os.Bundle
import android.widget.Button
import android.widget.EditText
import androidx.appcompat.app.AppCompatActivity
import androidx.recyclerview.widget.LinearLayoutManager
import androidx.recyclerview.widget.RecyclerView
import android.app.AlertDialog
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
```

```
import android.widget.TextView

class MainActivity : AppCompatActivity() {

   override fun onCreate(savedInstanceState: Bundle?) {
       super.onCreate(savedInstanceState)
       setContentView(R.layout.activity_main)

       // Initialize RecyclerView
       val recyclerView: RecyclerView =
findViewById(R.id.expense_list)
       recyclerView.layoutManager = LinearLayoutManager(this)

       // Sample data for RecyclerView
       val expenses = mutableListOf(
           Expense(100.0, "Food", "2025-04-24"),
           Expense(50.0, "Transport", "2025-04-23"),
           Expense(200.0, "Entertainment", "2025-04-22")
       )

       // Set Adapter for RecyclerView
       val adapter = ExpenseAdapter(expenses)
       recyclerView.adapter = adapter

       // Button to add a new expense
       val addButton: Button = findViewById(R.id.add_expense_button)
       addButton.setOnClickListener {
           val builder = AlertDialog.Builder(this)
           builder.setTitle("Add Expense")

           // Create an EditText to get the amount
           val amountInput = EditText(this)
           amountInput.hint = "Enter amount"
           builder.setView(amountInput)

           builder.setPositiveButton("Add") { dialog, _ ->
               val amount =
amountInput.text.toString().toDoubleOrNull()
               if (amount != null) {
                   // Create a new expense
                   val expense = Expense(amount, "Other",
"2025-04-24")
                   expenses.add(expense)

                   // Notify adapter of new data
                   adapter.notifyItemInserted(expenses.size - 1)
               }
               dialog.dismiss()
           }

           builder.setNegativeButton("Cancel") { dialog, _ ->
dialog.dismiss() }
           builder.show()
       }
   }
```

```
    // Data class for Expense
    data class Expense(val amount: Double, val category: String, val
date: String)

    // Adapter for the RecyclerView
    class ExpenseAdapter(private val expenses: List<Expense>) :
RecyclerView.Adapter<ExpenseAdapter.ExpenseViewHolder>() {

        override fun onCreateViewHolder(parent: ViewGroup, viewType:
Int): ExpenseViewHolder {
            val view =
LayoutInflater.from(parent.context).inflate(R.layout.expense_item,
parent, false)
            return ExpenseViewHolder(view)
        }

        override fun onBindViewHolder(holder: ExpenseViewHolder,
position: Int) {
            val expense = expenses[position]
            holder.amount.text = "Amount: ${expense.amount}"
            holder.category.text = "Category: ${expense.category}"
            holder.date.text = "Date: ${expense.date}"
        }

        override fun getItemCount(): Int {
            return expenses.size
        }

        // ViewHolder for Expense
        class ExpenseViewHolder(itemView: View) :
RecyclerView.ViewHolder(itemView) {
            val amount: TextView =
itemView.findViewById(R.id.expense_amount)
            val category: TextView =
itemView.findViewById(R.id.expense_category)
            val date: TextView =
itemView.findViewById(R.id.expense_date)
        }
    }
}
```

Expense.kt

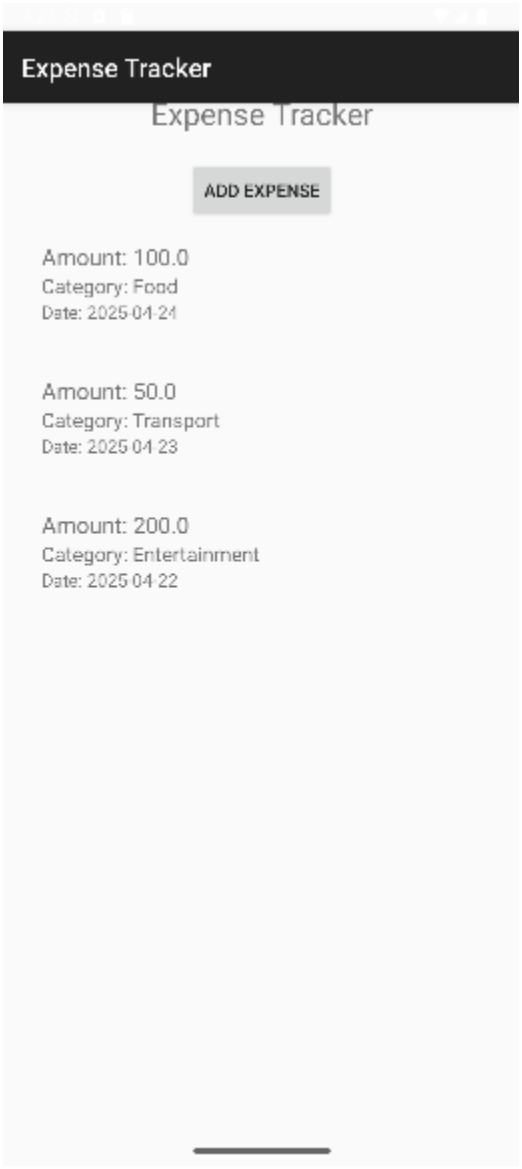
```
data class Expense(
    val amount: Double,
    val category: String,
    val date: String
)
```

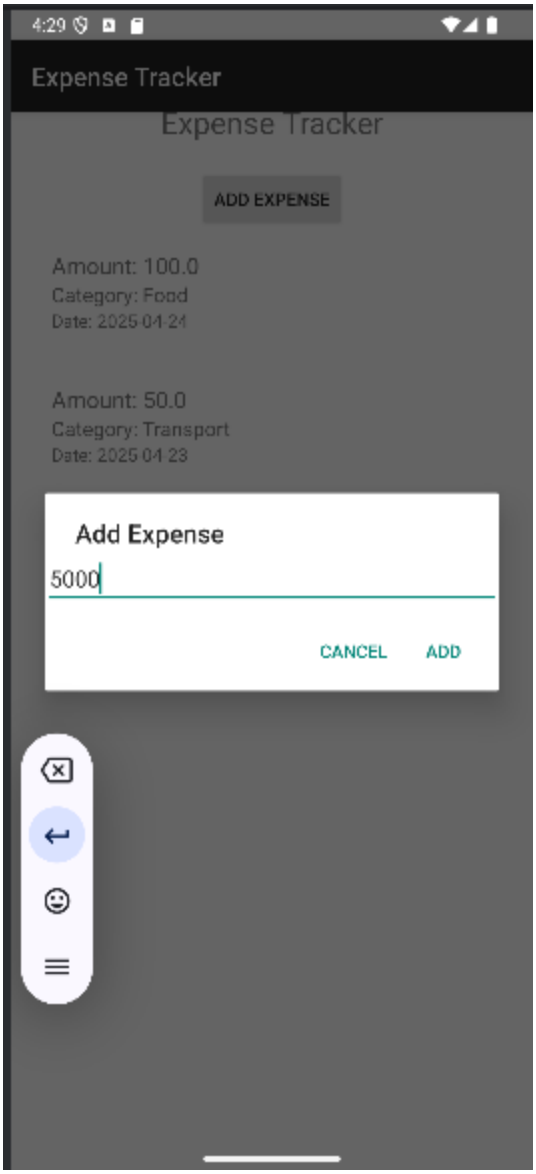
ExpenseAdapter.kt

```
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.TextView
import androidx.recyclerview.widget.RecyclerView

class ExpenseAdapter(private val expenses: List<Expense>) :
   RecyclerView.Adapter<ExpenseAdapter.ExpenseViewHolder>() {

   override fun onCreateViewHolder(parent: ViewGroup, viewType: Int):
ExpenseViewHolder {
       val itemView = LayoutInflater.from(parent.context)
           .inflate(android.R.layout.simple_list_item_2, parent,
false)
       return ExpenseViewHolder(itemView)
   }

   override fun onBindViewHolder(holder: ExpenseViewHolder, position:
Int) {
       val expense = expenses[position]
       holder.amountTextView.text = "Amount: ${expense.amount}"
       holder.categoryTextView.text = "Category: ${expense.category}"
   }

   override fun getItemCount() = expenses.size

   class ExpenseViewHolder(itemView: View) :
RecyclerView.ViewHolder(itemView) {
       val amountTextView: TextView =
itemView.findViewById(android.R.id.text1)
       val categoryTextView: TextView =
itemView.findViewById(android.R.id.text2)
   }
}
```

CHAPTER 4

SNAP SHOTS




4:29
Expense Tracker
Expense Tracker
ADD EXPENSE
Amount: 100.0
Category: Food
Date: 2025-04-24
Amount: 50.0
Category: Transport
Date: 2025-04-23
Add Expense
5000
CANCEL
ADD

CHAPTER 5

CONCLUSION

The **Expense Tracker** application successfully achieves its goal of helping users efficiently manage and monitor their daily financial activities. Through a simple and intuitive interface, users can easily add, view, and keep track of their expenses, encouraging better financial habits and budgeting awareness. The project demonstrates key Android development skills such as user interface design, activity handling, and local data management using Kotlin. By focusing on offline functionality and a lightweight design, the app ensures quick access and privacy without depending on internet connectivity. Overall, this project not only provides a practical solution for personal finance tracking but also serves as a strong foundation for further enhancements like implementing graphical reports, cloud data storage, or expense reminders, paving the way for more advanced Android app development skills.