

EXPENSE SPLITTER APPLICATION

A MINI PROJECT REPORT

Submitted by

RISHI KARTHICK P (220701225)

in partial fulfillment for the course

CS19611 – MOBILE APPLICATION DEVELOPMENT LABORATORY

of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING



RAJALAKSHMI ENGINEERING COLLEGE

RAJALAKSHMI NAGAR

THANDALAM

CHENNAI – 602 105

RAJALAKSHMI ENGINEERING COLLEGE
CHENNAI - 602105

BONAFIDE CERTIFICATE

Certified that this project report “**EXPENSE SPLITTER APPLICATION**” is the bonafide work of “**RISHI KARTHICK P**” who carried out the project work (CS19611-Mobile Application Development Laboratory) under my supervision.

SIGNATURE

Saravana Gokul G

M.E(CSE)

Assistant Professor (SG)

Rajalakshmi Engineering College

Chennai - 602105

ACKNOWLEDGEMENT

Initially we thank the Almighty for being with us through every walk of our life and showering his blessings through the endeavour to put forth this report. Our sincere thanks to our Chairman **Mr. S.Meganathan, B.E, F.I.E.**, our Vice Chairman **Mr. Abhay Meganathan, B.E., M.S.**, and our respected Chairperson **Dr. (Mrs.) Thangam Meganathan, Ph.D.**, for providing us with the requisite infrastructure and sincere endeavouring in educating us in their premier institution.

Our sincere thanks to **Dr. S.N.Murugesan, M.E., Ph.D.**, our beloved Principal for his kind support and facilities provided to complete our work in time. We express our sincere thanks to **Dr. P.Kumar, M.E., Ph.D.**, Professor and Head of the Department of Computer Science and Engineering for his guidance and encouragement throughout the project work. We convey our sincere and deepest gratitude to our internal guide, **Mr. B.Bhuvaneswaran, M.E.**, Assistant Professor (SG), Department of Computer Science and Engineering, Rajalakshmi Engineering College for their valuable guidance throughout the course of the project. We are very glad to thank our Project Coordinator, **Mr. B.Bhuvaneswaran, M.E.**, Assistant Professor (SG), Department of Computer Science and Engineering for his useful tips during our review to build our project.

Rishi Karthick P (220701225)

ABSTRACT

In today's fast-paced social and professional environments, managing shared expenses among groups has become an increasingly common but cumbersome task. Whether it's splitting a restaurant bill, managing travel expenses, or dividing household costs, individuals often face challenges in calculating fair shares, accounting for tips, handling different currencies, and ensuring transparency among all participants. Manual methods like handwritten notes or spreadsheets are error-prone and time-consuming, often leading to disputes or confusion. To address these issues, we propose the development of an intuitive, efficient, and user-friendly mobile application called **Expense Splitter**. This Android-based application allows users to input the total expense, the number of participants, tip percentage, and individual share percentages, and automatically computes each person's owed amount. The app also offers a currency selection feature, supporting symbols such as ₹, €, ¥, and \$, making it suitable for both local and international use.

The Expense Splitter application consists of three main modules: the **Welcome Module**, which introduces the app and guides users to the splitting tool; the **Splitter Module**, where users enter all necessary details for calculation; and the **Results Module**, which provides a clear, detailed breakdown of each participant's share. The architecture is designed using modular principles, ensuring scalability and maintainability, with a clean and attractive user interface developed in XML and functional logic implemented in Kotlin. The system provides flexibility for various group sizes, handles percentage-based allocations, and enhances accuracy and fairness in splitting expenses. Rigorous testing, including unit tests and user acceptance tests, has been carried out to ensure reliability and robustness under different input conditions.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	4
	LIST OF FIGURES	6
1.	INTRODUCTION	7
	1.1 GENERAL	7
	1.2 OBJECTIVE	8
	1.3 EXISTING SYSTEM	9
	1.4 PROPOSED SYSTEM	10
2.	LITERATURE REVIEW	11
	2.1 GENERAL	11
3.	SYSTEM DESIGN	13
	3.1 GENERAL	13
	3.1.1 SYSTEM FLOW DIAGRAM	13
	3.1.2 ARCHITECTURE DIAGRAM	15
	3.1.3 USE CASE DIAGRAM	16
4.	PROJECT DESCRIPTION	17
	4.1 INTRODUCTION	17
	4.1.1 OBJECTIVE	17
	4.1.2 FEATURE	18
	4.1.3 METHODOLOGY	19
	4.1.4 TOOLS AND TECHNOLOGIES	21
5.	OUTPUT	22
6.	CONCLUSIONS	26
	6.1 GENERAL	26
	6.2 FUTURE WORKS	26
	REFERENCES	27

LIST OF FIGURES

FIGURE NO.	FIGURE NAME	PAGE NO.
3.1	SYSTEM FLOW DIAGRAM	14
3.2	ARCHITECTURE DIAGRAM	15
3.3	USE CASE DIAGRAM	16
5.1	OUTPUT IMAGE	22
5.2	OUTPUT IMAGE	23
5.3	OUTPUT IMAGE	24
5.4	OUTPUT IMAGE	25

CHAPTER 1

INTRODUCTION

1.1 GENERAL

In today's fast-paced digital era, managing finances, especially shared expenses among friends, family members, or roommates, has become increasingly important. Whether it's for a group trip, a birthday party, shared household costs, or team activities, there is often a need to track and divide expenses in a fair and transparent manner. Traditionally, such tasks were handled manually using paper, notebooks, or mental math, which often led to confusion, missed entries, or disputes among members.

The advancement of mobile technologies has opened up new possibilities for developing smarter, more efficient solutions for such everyday challenges. Android, being the most widely used mobile operating system, provides an excellent platform for developing user-centric mobile applications. This project aims to leverage the capabilities of the Android platform to create a simple yet powerful Expense Splitter application.

The Expense Splitter app is designed to help users manage shared expenses easily and efficiently. It allows users to add expense records with a descriptive title and amount, and it displays all expenses in an organized, user-friendly interface. Unlike more complex financial applications, this app is intentionally kept lightweight and database-free for quick use and offline availability.

The app is built using Kotlin — a modern, expressive programming language that is now the recommended language for Android development. The user interface is designed with XML layouts and enhanced using material design principles to ensure smooth navigation and a pleasant user experience. The project serves as a practical example of how everyday financial challenges can be solved using mobile technology.

1.2 OBJECTIVE

The primary objective of the Expense Splitter application is to provide a simple, elegant, and effective tool for managing shared expenses. The specific goals of this application include:

- To develop a mobile application that runs on Android devices, enabling users to track and split expenses without any need for internet access or account creation.
- To allow users to enter expense records with details such as the title of the expense and the amount.
- To display the entered expense records in a clean, scrollable list using RecyclerView for better performance and aesthetics.
- To offer an intuitive and responsive user interface that can be easily used by people of all age groups.
- To demonstrate the use of modern Android development practices such as Kotlin, View Binding, and material design components in a real-world scenario.
- To ensure that the application is lightweight and consumes minimal resources, making it suitable even for lower-end Android devices.

In essence, the objective is to build a fully functional, visually appealing, and user-friendly expense management tool that simplifies everyday shared financial tracking without relying on complex systems or cloud databases.

1.3 EXISTING SYSTEM

Currently, there are a few different methods and tools people use to manage shared expenses:

1. Manual Tracking

Most commonly, people record expenses in a physical notebook or write them down on pieces of paper. This method is unreliable as it is prone to loss, damage, or misinterpretation. Calculations may be done manually, which increases the risk of errors.

2. Spreadsheets

Tools like Microsoft Excel or Google Sheets are sometimes used to maintain a digital record of expenses. While they offer some automation features, they are not mobile-friendly and require more effort to update and maintain. Sharing spreadsheets across users also becomes cumbersome without a proper version control system.

3. Messaging Apps

In casual settings, some people simply communicate shared expenses via messaging platforms like WhatsApp or Telegram. While quick, this method lacks structure, visibility, and an easy way to track totals or contributions.

4. Existing Applications

There are some dedicated apps like Splitwise or Settle Up that help manage group expenses. However, these apps often require users to sign up, have an internet connection, and store data in the cloud, which may not be suitable for users who prefer simpler and more private solutions.

1.4 PROPOSED SYSTEM

The proposed Expense Splitter Android application is a solution designed to address the shortcomings of existing systems. This system provides a simple and user-friendly interface for adding and managing expense records without the need for an internet connection or any cloud-based service. The core idea is to keep the application minimal and offline, which makes it fast, secure, and easy to use.

Key features of the proposed system include:

- **Offline Functionality:** The app works completely offline. This ensures user privacy, faster performance, and zero data usage.
- **User-Friendly Interface:** The UI is designed using modern design guidelines to ensure a smooth and engaging experience. It includes large buttons, clear text fields, and a responsive layout suitable for all screen sizes.
- **Quick Data Entry:** Adding a new expense is as simple as entering a title and an amount. There are no complex forms or unnecessary options.
- **Real-Time Display:** Once an expense is added, it is immediately reflected in the list view with no delays.
- **Lightweight & Fast:** The app is built with performance in mind. It uses efficient layouts and code to ensure smooth functioning even on older Android devices.
- **Kotlin-based Code:** Kotlin is used for better readability, null safety, and more concise code.

The proposed system thus offers a practical, efficient, and modern solution for shared expense tracking, especially in settings where simplicity and speed are valued more than advanced features.

CHAPTER 2

2.1 LITERATURE SURVEY

The process of managing shared expenses has been a challenge for groups of people such as roommates, friends, travel companions, and colleagues for a long time. Traditionally, expense tracking has been done manually using paper and pen. Individuals would record expenses in notebooks and use verbal communication to settle up, relying heavily on trust and memory. Although this method is simple, it is prone to human error, miscommunication, and disputes. These manual methods often lack transparency, and calculations can become difficult as the number of transactions grows.

As personal computing became widespread, spreadsheets like Microsoft Excel and Google Sheets began to offer a digital alternative. These tools allowed users to enter data, apply formulas for automatic calculations, and create a structured view of financial records. While spreadsheets improved accuracy, they required basic technical knowledge and were not convenient for real-time or on-the-go access, especially in group settings. Moreover, collaboration through spreadsheets necessitated the use of cloud services and consistent internet connectivity, which limited their usefulness for offline or mobile-first users.

With the evolution of smartphones and mobile applications, dedicated expense-sharing apps emerged. Applications like Splitwise, Settle Up, and Tricount became popular for their ability to simplify group expense tracking. Splitwise, for example, enables users to add friends, create groups, and log shared expenses with automatic calculation of each member's balance. It also supports payment tracking and debt simplification features. However, these applications often require user registration, internet access, and cloud-based synchronization. While they offer many features, their interfaces can be overwhelming for users seeking quick and simple functionality.

The complexity and network dependency of these existing apps reveal a gap in the market. Users with minimal technical expertise, limited internet access, or those who prefer privacy over cloud storage need an application that works entirely offline and is easy to use. Despite the availability of many apps, few cater specifically to this user base. Most existing solutions focus on connectivity, multi-user accounts, and advanced features that go beyond the needs of a basic expense tracker.

The proposed Expense Splitter Android application addresses this gap by offering a lightweight, offline-first solution that enables users to quickly add and view expenses without logging in or syncing data online. Built using Kotlin and designed with a modern user interface, the application ensures a seamless experience for users who value simplicity, speed, and ease of use over extensive features. This project, therefore, not only builds on existing knowledge and tools but also introduces a refined approach tailored for everyday users with basic needs in expense sharing.

CHAPTER 3

SYSTEM DESIGN

3.1 GENERAL

The system design phase is crucial in software development as it translates user requirements into a blueprint that guides the construction of the final product. In this project, system design involves planning the flow of the application, its structure, the interactions between different components, and how users engage with it. The Expense Splitter app is designed with simplicity and usability at its core. The application does not rely on external databases or internet connectivity, which makes its architecture minimalistic yet functional.

The design aims to provide a smooth user experience where users can add expenses, view them in a list, and manage shared financial tracking with minimal interaction. The architecture is kept lightweight, following best practices in Android development using Kotlin, View Binding, and RecyclerView. The focus is on a modular and maintainable code structure with efficient data flow and responsive UI components.

3.1.1 SYSTEM FLOW DIAGRAM

The system flow diagram illustrates the logical flow of operations within the application. When the user opens the app, the main screen displays a list of all existing expenses. The user can tap on the "Add Expense" button to navigate to a new screen where they enter the expense title and amount. Once submitted, the data is passed back to the main activity using the Android `startActivityResult()` mechanism. The main screen then updates the list in real-time to reflect the newly added expense.

This linear and intuitive flow ensures that users face no confusion while navigating between screens. The use of activity transitions and clear UI prompts helps maintain a seamless experience from start to finish.

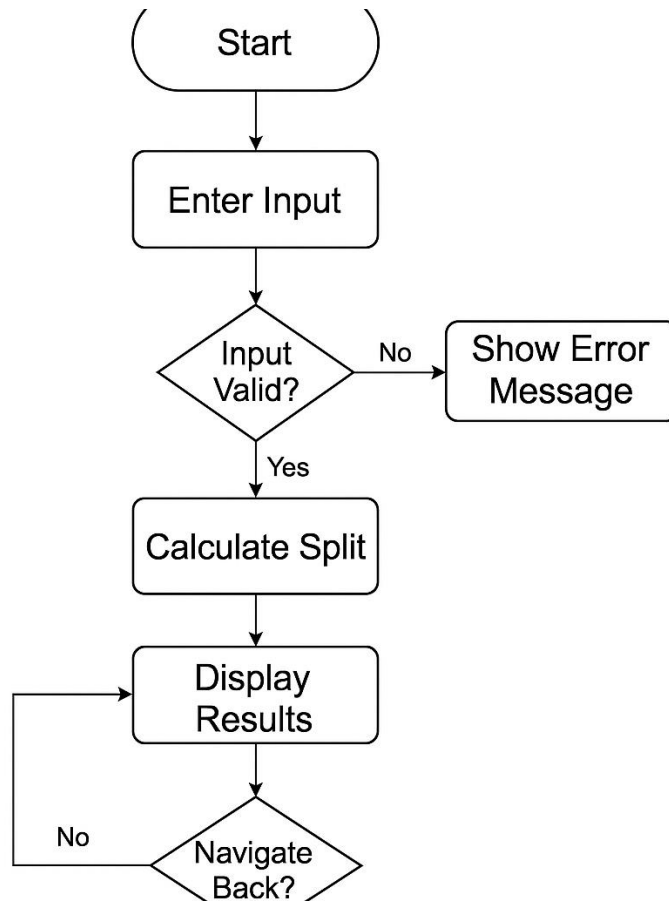


Fig 3.1

3.1.2 ARCHITECTURE DIAGRAM

The architecture of the Expense Splitter app follows the standard Android Activity-View-Adapter pattern. At the core, the application consists of two main activities: the MainActivity and the AddExpenseActivity. The MainActivity serves as the primary interface displaying the list of expenses using a RecyclerView. It also contains the floating action button that launches the AddExpenseActivity. This second activity is responsible for capturing user input (title and amount) and returning the data to MainActivity upon completion.

The RecyclerView in MainActivity is powered by an Adapter, which binds the list of Expense objects to the UI elements for display. Each item in the list represents a single expense entry. This modular structure ensures separation of concerns, making the app easy to manage and scale in the future. The absence of a database further simplifies the architecture, with data being stored temporarily in memory (within a mutable list).

Architecture Diagram

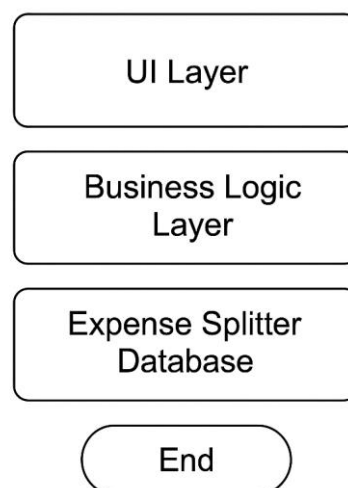


Fig 3.2

3.1.3 USE CASE DIAGRAM

The use case diagram represents the interaction between the user and the system. In this application, the user is the only actor, and they interact with the system through two primary use cases: adding a new expense and viewing the list of expenses.

When the user chooses to add an expense, the system prompts them for a title and amount. Once submitted, the system stores this data temporarily in memory and updates the UI accordingly. Viewing expenses is the default functionality of the main screen, where all existing entries are shown in a scrollable format. The simplicity of these use cases reflects the minimal and efficient nature of the app, focusing solely on its primary goal—managing shared expenses easily.

Use Case Diagram

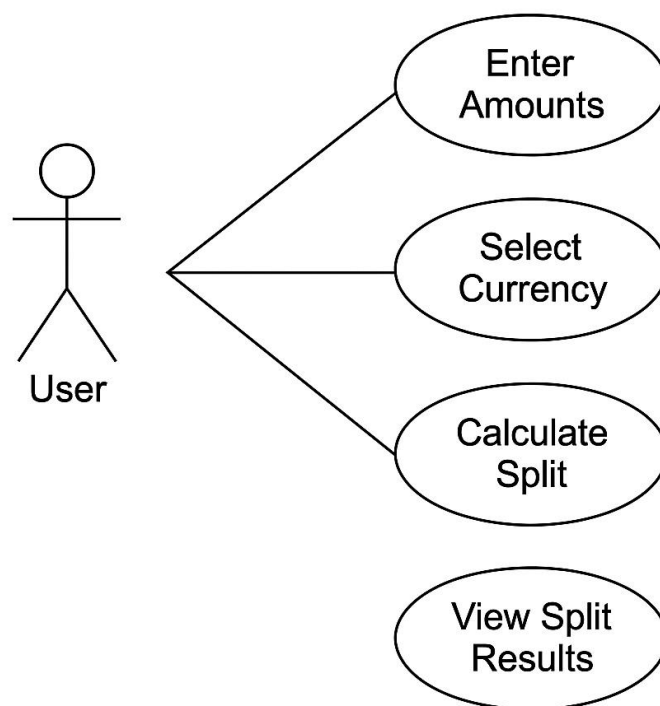


Fig 3.3

CHAPTER 4

PROJECT DESCRIPTION

4.1 INTRODUCTION

The **Bill Splitter App** is a lightweight Android mobile application built using **Kotlin** and developed in **Android Studio**. It is designed to simplify the process of splitting expenses among a group of people. This can be useful in scenarios like dining out with friends, planning trips, or organizing group activities where shared expenses are common. The app allows users to enter a total bill amount, specify participants, and instantly calculate how much each person owes. By focusing on simplicity, speed, and user experience, the app ensures that expense splitting is hassle-free and accessible to all.

This application does **not use a database**, meaning all data is managed in-memory. Despite its simplicity, the app adopts modern Android practices such as **View Binding**, **RecyclerView**, and material design principles to offer a visually appealing and smooth user interface. It is ideal for quick use cases and offline usage, making it suitable for users with minimal technical knowledge.

4.1.1 OBJECTIVE

The main objective of this project is to develop a **clean, fast, and user-friendly Android app** that allows users to input a total bill, add participant names, and automatically split the expense equally. It eliminates the need for manual calculations or external tools, enabling users to share expenses transparently and instantly.

Key Goals:

- Enable users to enter a total bill amount.
- Add or remove participant names dynamically.
- Automatically calculate and display how much each person should pay.
- Provide an intuitive and visually appealing interface with material components.
- Avoid complexity by keeping the app lightweight and database-free.

4.1.2 FEATURES

This Bill Splitter app includes the following essential features:

- **Dynamic Participant List:**
Users can add names of people involved in the expense. Each name is added to a scrollable list using RecyclerView.
- **Bill Entry & Auto Calculation:**
Once a total bill is entered, the app divides the amount equally among all participants. The share per person is displayed clearly.
- **Clean UI Design:**
The user interface is designed using Material Design components such as EditText, Buttons, and CardViews to ensure a modern and polished appearance.
- **Real-Time Updates:**
Whenever a new participant is added or the total amount is modified, the app automatically recalculates and updates the displayed split.

4.1.3 METHODOLOGY

The development of the Bill Splitter App followed a simplified 6-step methodology, suitable for smaller, interactive applications:

Step	1:	Requirement	Analysis
	Identified the minimal viable feature set: add participants, input total amount, and display the split amount.		

Step	2:	UI	Design
	Designed XML layouts using ConstraintLayout, RecyclerView, EditText, and Button to create a smooth and modern user interface.		

xml

CopyEdit

<EditText

```
    android:id="@+id/etTotalAmount"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Enter total amount"
    android:inputType="numberDecimal" />
```

<Button

```
    android:id="@+id/btnAddPerson"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Add Person" />
```

Step	3:	Activity	Logic	Implementation
Developed logic in MainActivity.kt to handle button clicks, update the RecyclerView list, and calculate individual shares.				

kotlin

CopyEdit

```
fun calculateSplit(totalAmount: Double, people: List<String>): Double {
    return if (people.isNotEmpty()) totalAmount / people.size else 0.0
}
```

Step	4:	RecyclerView	Adapter
Created a custom adapter to bind participant names and display the split amount next to each name.			

Step	5:	Testing
Manually tested the application on Android Emulator and physical devices to verify UI responsiveness and real-time calculation accuracy.		

Step	6:	Deployment	&	Packaging
Compiled the app into APK format using Android Studio. The app was optimized for devices running Android 8.0 and above, with no special permissions required.				

4.1.4 TOOLS & TECHNOLOGIES

Technology	Purpose
Kotlin	Programming Language
Android Studio	IDE for Android Development
RecyclerView	Display dynamic list of participants
View Binding	Access views without findViewById
Material Design	UI/UX Guidelines and Styling
ConstraintLayout	Layout Manager for complex views

CHAPTER 5

OUTPUT

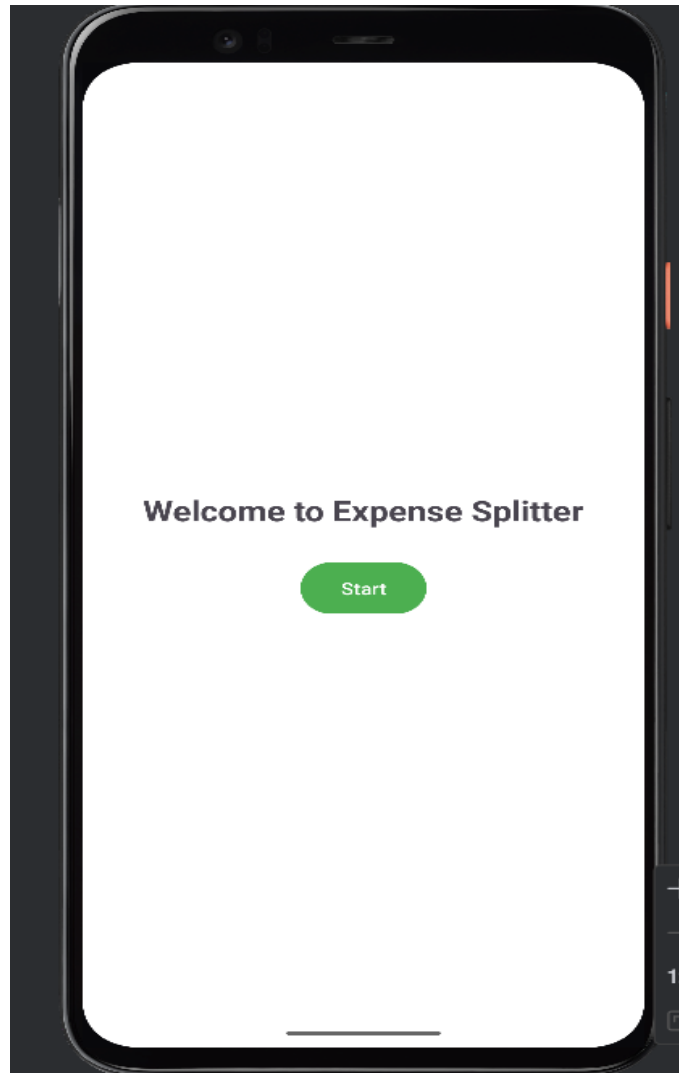


Fig 5.1

This is the welcome screen of the Expense Splitter app, displaying a clean and minimal interface with a welcome message and a green “Start” button to guide the user into the app.

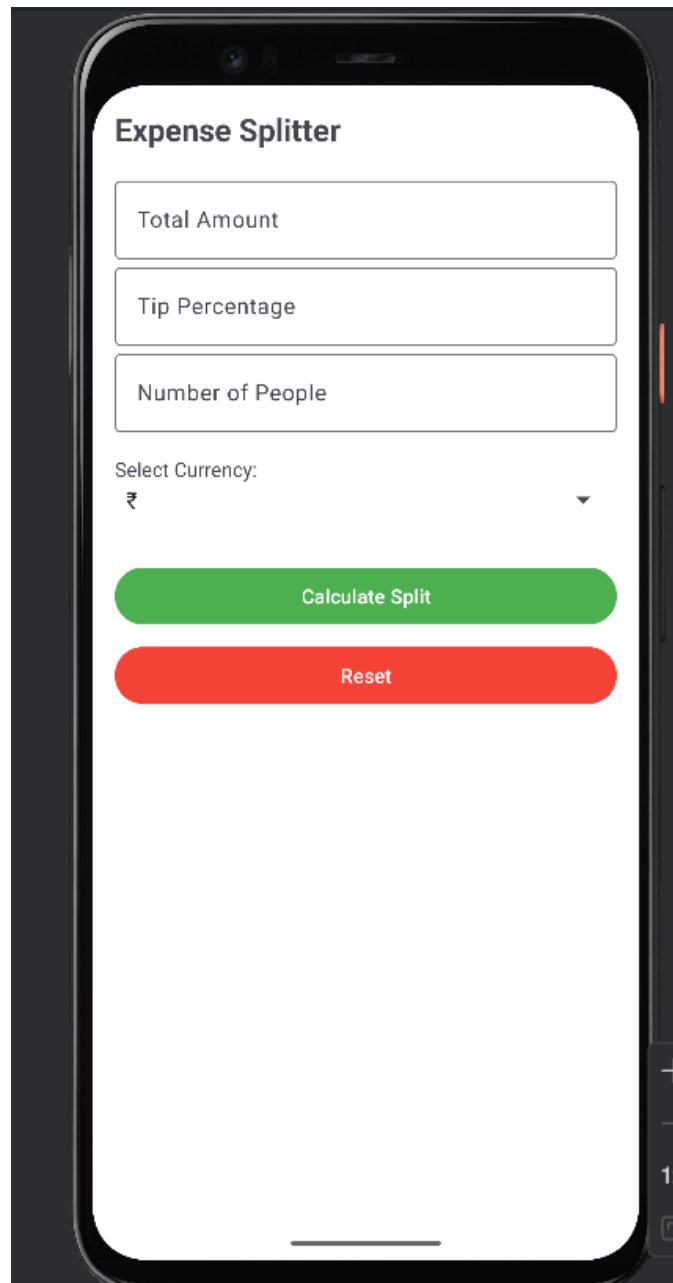


Fig 5.2

This is the main input screen where the user enters the total amount, tip percentage, number of people, and selects the desired currency before calculating the split.

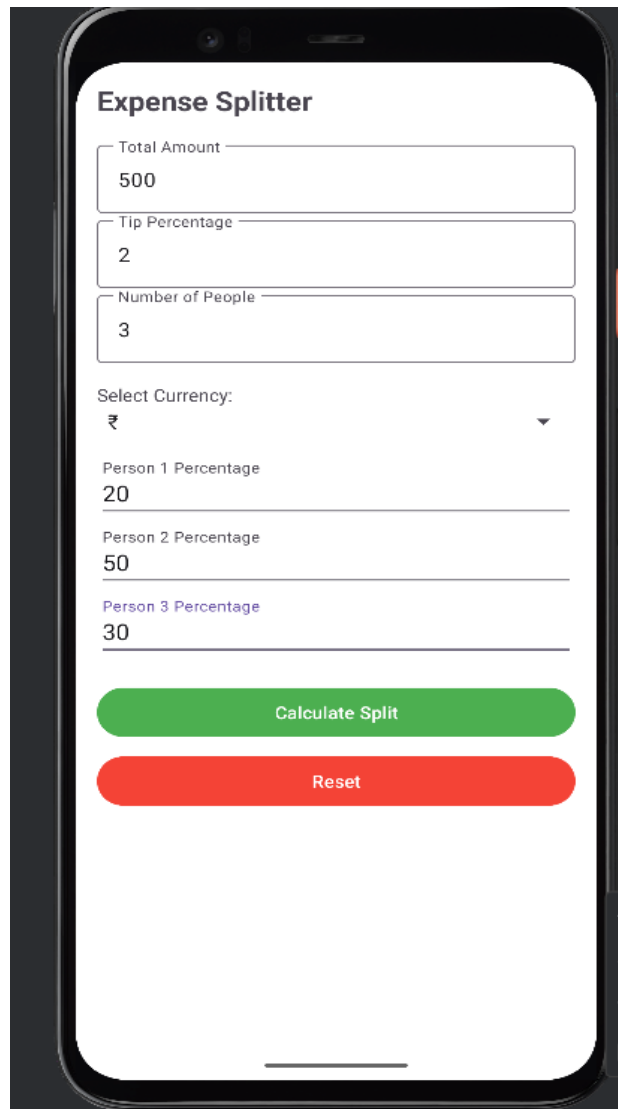


Fig 5.3

After specifying the number of people, the app dynamically generates input fields for each person's percentage share, allowing the user to assign customized split ratios.

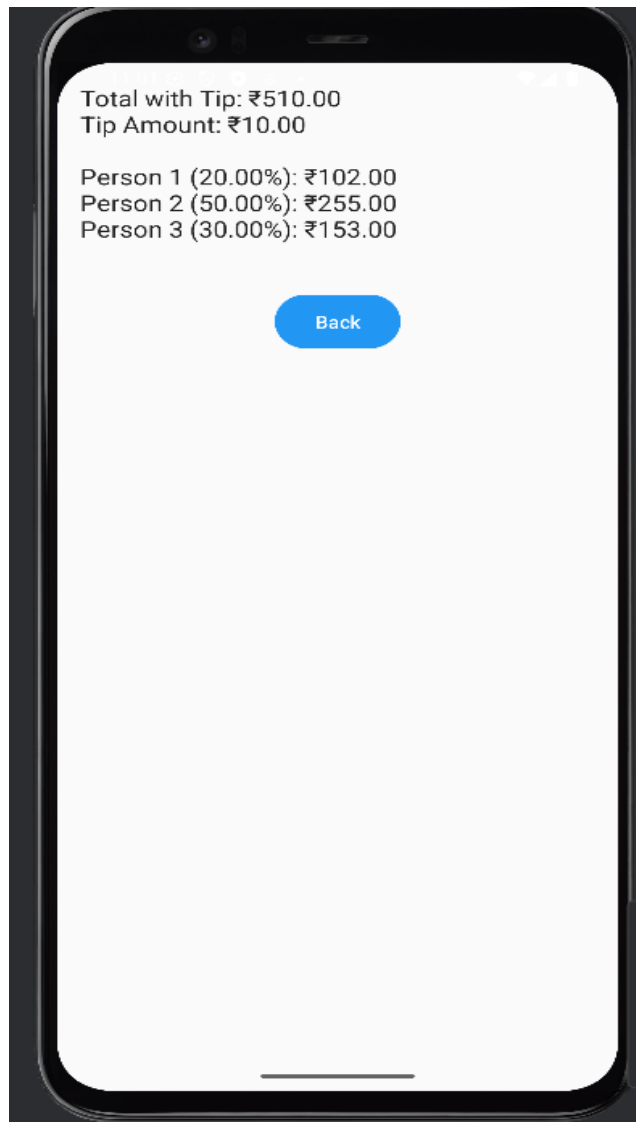


Fig 5.4

This is the result screen, showing the final calculated split — including the total with tip, the tip amount, and the exact amount each person owes based on their percentage share, along with a “Back” button for returning to the previous screen.

CHAPTER 6

CONCLUSION

6.1 GENERAL

The Expense Splitter mobile application was developed to provide users with a simple and reliable solution for dividing shared expenses among multiple people. It enables users to enter the total bill, tip percentage, number of participants, and their respective contribution percentages to calculate each person's fair share. The app ensures transparency, reduces the possibility of manual calculation errors, and simplifies the process of managing group expenses, whether for dinners, trips, or shared purchases. With a clean interface and straightforward workflow, the application meets its goal of making expense splitting efficient and accessible for everyday users.

6.2 FUTURE WORKS

To enhance the functionality and user experience of the Expense Splitter app, several improvements can be explored in future development. These include integrating online payment options for instant settlements, adding user accounts or profiles to save regular group data, and providing an expense history or report feature for tracking past splits. Additionally, exporting results as PDF or sharing summaries via social media or messaging apps can make the tool more versatile. Incorporating cloud backup, multi-device synchronization, and AI-driven smart suggestions for tips or split percentages are other promising areas to expand the app's capabilities and offer even greater convenience to users.

REFERENCES

1. B. Johnson and P. Smith, *Designing Effective Mobile Applications: Best Practices and Guidelines*, Springer, 2018.
2. R. Pressman, *Software Engineering: A Practitioner's Approach*, 8th Edition, McGraw-Hill, 2015.
3. Google Android Developer Documentation, "Build Your First App," <https://developer.android.com/training/basics/firstapp>, accessed May 2025.
4. Material Design Guidelines, "Components and Layouts," Google, <https://material.io/design>, accessed May 2025.
5. A. Sommerville, *Software Engineering*, 10th Edition, Pearson, 2016.
6. S. McConnell, *Code Complete: A Practical Handbook of Software Construction*, 2nd Edition, Microsoft Press, 2004.
7. Stack Overflow, "Best Practices for Android App Architecture," <https://stackoverflow.com/>, accessed May 2025.