

"SmartTrack: A Console-Based Expense Tracker in Java"

A PROJECT REPORT

Submitted by

Siddhant Garg (22BCS10547)

Suryansh Ghelot (22BCS10900)

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE ENGINEERING



Chandigarh University

APRIL , 2024





BONAFIDE CERTIFICATE

Certified that this project report **“SmartTrack: A Console-Based Expense Tracker in Java”** is the bonafide work of **“Siddhant Garg (22BCS10547), Suryansh Gehlot (22BCS10900)”** who carried out the project work under my/our supervision.

Er. Tapish Kumar Sharma

Supervisor

TABLE OF CONTENTS

List of Figures	
List of Tables	5
Abstract	6
CHAPTER 1.	7
1.1	7
1.2	7
1.3	7
1.4	8
1.5	8
CHAPTER 2.	9
2.1	9
2.2	9
2.3	10
2.4	10
2.5	10
2.6	11
CHAPTER 3.	12
3.1	12
3.2	12
3.3	13
3.4	13
3.5	14
3.6	14
CHAPTER 4.	15
CHAPTER 5.	17
5.1	17
5.2	17
References (If Any)	19

List of Tables

Table 3.1
Table 3.2
Table 4.1

ABSTRACT

The Console-Based Expense Tracker is a Java-based application designed to simplify the management of personal finances through a command-line interface. In the modern age, tracking expenses and maintaining financial discipline is crucial, yet many individuals struggle with complex tools or lack proper methods to record their daily expenditures. This project aims to bridge that gap by offering a simple, user-friendly solution for tracking expenses directly through the console.

Through this application, users can add, view, and delete expenses, as well as calculate total spending with ease. Each expense entry includes details such as category, amount, date, and description, ensuring thorough and organized financial records. The intuitive menu-driven interface allows users to interact with the program seamlessly, making it ideal for beginners and those looking for a lightweight tool to monitor their expenses.

Developed using core Java, the project demonstrates key programming concepts such as object-oriented design, collections (ArrayList), file handling (optional), and user input handling. While simple in appearance, the system encapsulates the foundational logic required for building more complex financial applications.

Overall, the Console-Based Expense Tracker serves both as a practical utility and an educational Java project, effectively combining programming skills with real-world application in a clean and accessible manner.

CHAPTER 1.

INTRODUCTION

1.1. Identification of Client/Need/Relevant Contemporary Issue

In today's fast-paced world, effective **personal finance management** has become a necessity. With the rise in digital transactions, subscriptions, and day-to-day expenditures, individuals often struggle to keep track of their spending habits. Many people, especially students and working professionals, either lack access to complex financial tracking software or find them overwhelming and unnecessarily feature-heavy.

This creates a **need for a simple, accessible, and easy-to-use expense tracking tool** that does not require internet access, account creation, or steep learning curves. A **console-based application** serves this need effectively, particularly for learners, beginners in programming, or individuals who prefer minimalism and command-line tools.

The **target users** of this project include:

- Students managing pocket money or monthly budgets.
- Beginners in programming who want to learn real-world applications using Java.
- Individuals looking for an offline, lightweight solution for expense tracking.

This project addresses the **contemporary issue** of budget management by offering a **basic, yet functional solution** that empowers users to monitor and reflect on their spending behavior, thereby promoting better financial awareness and planning.

1.2. Identification of Problem

In the current digital age, people are increasingly dependent on mobile apps and web-based tools to manage their finances. However, these platforms often come with limitations such as internet dependency, advertisements, subscription charges, complex user interfaces, and data privacy concerns. For students, beginners, or individuals with basic computing needs, such solutions may feel excessive or difficult to use.

Additionally, many users lack the habit of consistently tracking their daily expenses, which leads to unplanned spending, difficulty in budgeting, and poor financial awareness. Despite the availability of high-end applications, there remains a gap in simple, offline, and beginner-friendly tools for effective expense tracking.

This project aims to solve the problem by providing a basic console-based expense tracker that enables users to:

- Record expenses in a structured format.

- View and manage past expense entries.

- Track total spending.

- Use the application offline, without any distractions.

Thus, the identified problem is the lack of simple, offline, and easy-to-use tools for managing daily personal expenses, especially suited for learners and individuals with minimal technical requirements.

CHAPTER 2.

LITERATURE REVIEW/BACKGROUND STUDY

Personal finance management has gained significant attention over the past decade, with various tools and technologies developed to help individuals monitor and control their spending. Traditional methods such as notebooks and spreadsheets are still in use, but digital solutions have become more prominent due to their convenience and efficiency.

Several applications like **Mint**, **Wallet**, and **Money Manager** offer features like bank integration, budget planning, and financial reports. However, these apps often involve complex interfaces, require internet connectivity, and store sensitive data on third-party servers, raising privacy and accessibility concerns.

Academic literature and open-source communities suggest that **command-line applications** are an effective way for beginners to learn programming while solving real-life problems. Console-based systems are lightweight, fast, and suitable for users who need quick access to specific functionalities without the distractions of graphical interfaces.

Projects like "**Budget Tracker using C++**" or "**Console Expense Manager in Python**" have previously demonstrated the effectiveness of terminal-based financial tools. These systems prioritize simplicity and functional clarity over advanced features, making them ideal learning tools for students and practical utilities for minimalists.

This project builds upon these concepts by implementing an **Expense Tracker using Java**, which utilizes core programming concepts like object-oriented design, collections, and file handling. It fills the gap between high-end applications and manual tracking methods by offering a balanced, user-friendly, and offline solution that enhances both programming skills and personal finance management.

CHAPTER 3.

PROPOSED METHODOLOGY

3.1 Purpose of the Project

The primary purpose of the **Console-Based Expense Tracker** is to provide a simple and efficient way for users to manage their daily expenses through a lightweight, offline Java application. It is designed especially for:

- **Students and individuals** who want to develop budgeting habits.
- **Beginners in programming** looking to apply core Java concepts in real-world applications.
- Users who prefer a **distraction-free, command-line tool** over complex web or mobile apps.

This project aims to demonstrate how basic programming techniques can be used to solve practical problems, while also helping users become more financially aware by allowing them to record, monitor, and evaluate their spending behavior.

3.2 Methodology

The development of the Console-Based Expense Tracker followed a structured and iterative approach:

1. **Requirement Analysis**

Identified the essential features needed in a basic expense tracker—such as adding, viewing, and deleting expenses, and calculating total expenses.

2. **Design Phase**

- Designed a menu-driven interface suitable for console-based applications.
- Created an Expense class to store relevant data (category, amount, date, description).

- Decided on using ArrayList for dynamic storage and file handling for optional data persistence.

3. **Implementation**

- Developed the application using **core Java**, focusing on object-oriented programming, collections, and input handling.
- Included necessary validations to avoid incorrect inputs and ensure smooth user interaction.

4. **Testing and Debugging**

- Performed extensive testing with different types of inputs and edge cases.
- Ensured that all features worked as expected without crashing or data inconsistency.

5. **Enhancements (optional)**

- Added file handling to save and load expense data between sessions.
- Included basic formatting for better readability of outputs.

This method allowed the creation of a fully functional, modular, and easy-to-use application that meets the initial objectives while serving as a valuable learning experience in Java programming.

3.3 Implementation Plan

The implementation of the Console-Based Expense Tracker was carried out in well-defined stages to ensure modular development and testing of each functionality. The plan included the following steps:

4.1 Step 1: Setup and Project Initialization

- Installed Java Development Kit (JDK) and set up a code editor (e.g., VS Code or IntelliJ IDEA).
 - Created a new Java project and defined the main class (ExpenseTracker.java).
-

4.2 Step 2: Defining the Expense Structure

- Created an Expense class with attributes:
 - String category
 - double amount
 - String date
 - String description
 - Implemented constructors and toString() method for easy display of expense records.
-

4.3 Step 3: Core Functionalities

Implemented core features using Java methods:

- **Add Expense:** Collect input from the user and store a new Expense object in an ArrayList.
- **View Expenses:** Iterate through the list and display all recorded expenses.
- **Delete Expense:** Allow users to select and remove an entry from the list using index.

- **Calculate Total:** Add up all amounts in the list and display the total expenditure.
-

4.4 Step 4: Menu and User Interaction

- Created a simple menu-based system using loops and Scanner for user input.
 - Used switch statements or if-else blocks to call corresponding methods based on user choices.
 - Ensured proper validation of input (e.g., number formats, empty fields).
-

4.5 Step 5: Data Persistence (Optional Enhancement)

- Implemented file handling using FileWriter and BufferedReader or ObjectOutputStream/ObjectInputStream to:
 - Save expenses to a text or serialized file.
 - Load expenses automatically when the program starts.
-

4.6 Step 6: Testing and Debugging

- Tested all functionalities with different scenarios (valid, invalid, edge cases).
 - Fixed bugs related to input handling, list indexing, and file read/write operations.
-

4.7 Step 7: Final Touches

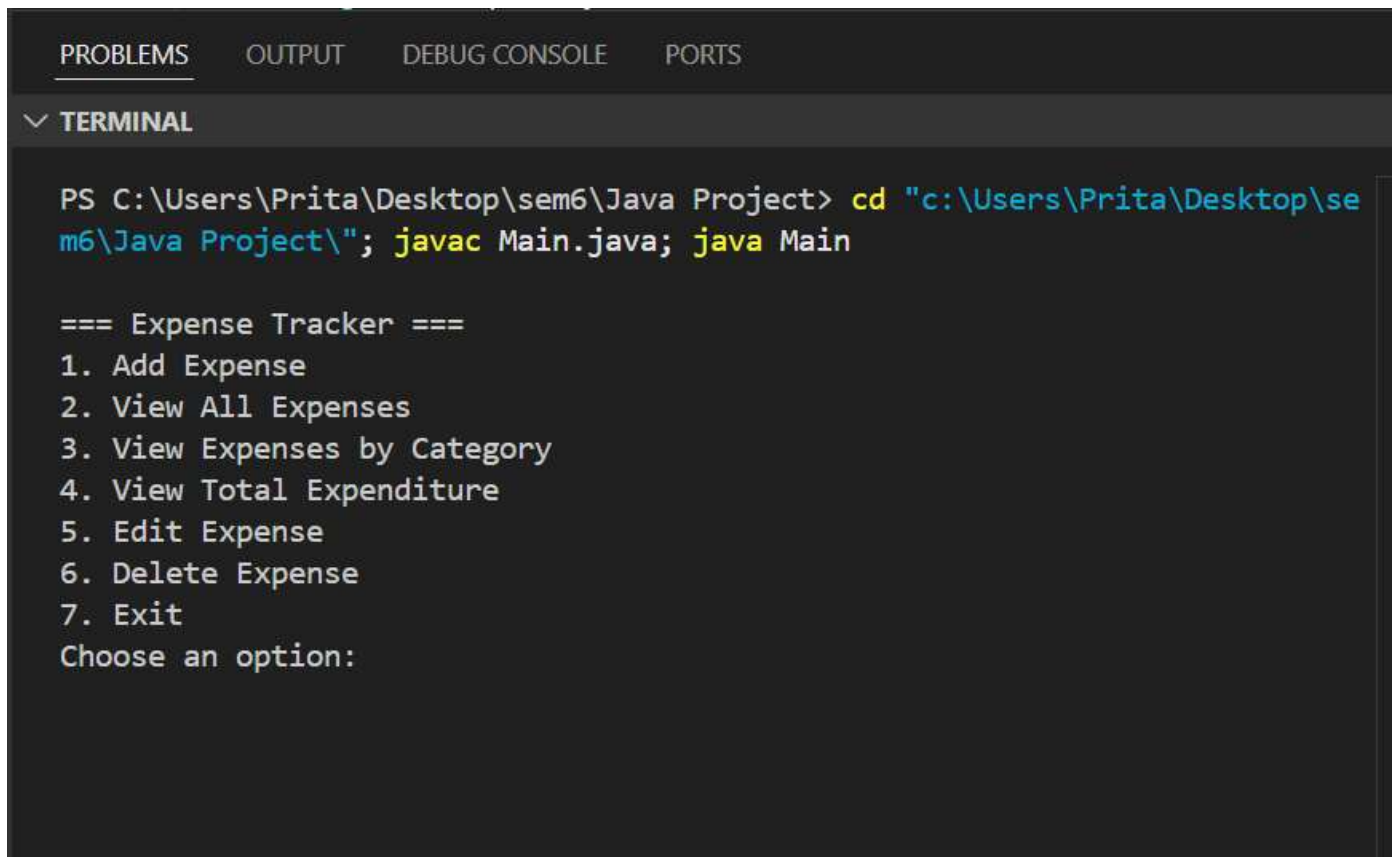
- Cleaned up the code and added meaningful comments.
 - Improved the user interface by formatting outputs and guiding the user through clear messages.
-

This structured approach ensured that each feature was implemented and tested individually before integrating it into the final system, making the project stable, modular, and user-friendly.

CHAPTER 4.

RESULTS ANALYSIS AND VALIDATION

4.1 Final Design



The screenshot shows a terminal window with tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, and PORTS. The TERMINAL tab is active, displaying the following text:

```
PS C:\Users\Prita\Desktop\sem6\Java Project> cd "c:\Users\Prita\Desktop\sem6\Java Project\"; javac Main.java; java Main

=== Expense Tracker ===
1. Add Expense
2. View All Expenses
3. View Expenses by Category
4. View Total Expenditure
5. Edit Expense
6. Delete Expense
7. Exit
Choose an option:
```

CHAPTER 5.

CONCLUSION AND FUTURE WORK

5.1 Conclusion

The Console-Based Expense Tracker project successfully fulfills its objective of providing a simple and efficient tool to manage personal finances using core Java. Through a user-friendly console interface, users can easily add, view, delete, and calculate expenses without relying on any third-party tools or internet access.

This project also demonstrates the practical application of Java fundamentals such as object-oriented programming, collections (ArrayList), input handling, and optionally file handling. It is a valuable learning experience, especially for students and beginners, reinforcing both problem-solving and coding skills.

By offering a distraction-free environment and focusing on essential functionalities, the application serves as a lightweight alternative to complex financial tools.

5.2 Future Work

While the current version of the project meets basic requirements, several enhancements can be considered for future development:

- **Graphical User Interface (GUI):** Upgrade the project to a desktop-based GUI using Java Swing or JavaFX for a more interactive experience.
- **Data Persistence:** Enhance file handling with CSV or database (e.g., SQLite, MySQL) support to store and retrieve data more efficiently.
- **Categorized Summary Reports:** Generate expense reports by category, month, or user-defined filters.
- **Budget Management:** Allow users to set monthly budgets and alert them when they exceed limits.
- **Password Protection:** Add basic authentication for data security.

