NAME - Anirudh Singh
Reg. No. - 24BCE10259
Slot- C11+C12+C13+F11+F12

Introduction

Managing daily expenses is a common challenge faced by students, working professionals, and households. Without a proper system, tracking where money is spent becomes difficult and often results in poor financial planning.
The **Expense Tracker Application**, developed using **Core Java**, provides a simple console-based solution to record, categorize, and analyze personal expenses. The application enables users to add, view, and delete expenses and also view a summary of their spending habits.

Problem Statement

Users often rely on memory or handwritten notes to track their expenses, leading to poor accuracy and loss of data.
There is a need for a simple, lightweight software tool that helps individuals track expenses without requiring complex installations or advanced technical knowledge.

Functional Requirements

FR1: Add Expense

- Input: amount, category, description, date
- Store expense entry in the system

FR2: View All Expenses

- Display all recorded expenses in tabular format

FR3: Delete Expense

- Delete an expense by its ID

FR4: Filter Expenses (Optional)

- Filter by category or date

FR5: Show Summary

- Show total expenses
- Show category-wise spending

Non-functional Requirements

NFR1: Usability

- Simple menu-driven interface

NFR2: Performance

- Operations should execute within 1 second

NFR3: Reliability

- Expense data stored persistently using text file (if implemented)

NFR4: Maintainability

- Modular design using OOP concepts

NFR5: Portability

- Runs on any machine with Java installed

NFR6: Error Handling

- Input validation for amount, date, and empty fields

System Architecture

Architecture Type:

Three-layer console-based architecture

1. **Presentation Layer**
    - Console UI
    - Menu options (1: Add, 2: View, 3: Delete, 4: Summary, 5: Exit)
2. **Business Logic Layer**
    - Expense Service
    - Validation
    - Calculations for summary
3. **Data Layer**
    - Stores expenses in:
        - ArrayList (runtime), and/or
        - Text file expenses.txt (persistent storage)

Design Decisions & Rationale

Decision	**Rationale**
Core Java console app	Simple, portable, easy for beginners
ArrayList for storage	Fast insertion & retrieval
Text file persistence	Lightweight, no DB setup needed
Menu-driven UI	Easy for non-technical users
Modular classes	Better readability & maintainability

Implementation Details

Language: Java (Core Java)

Key Concepts Used:

- Classes & Objects
- ArrayList
- Exception Handling
- File Handling (optional)
- Methods & Modular structure

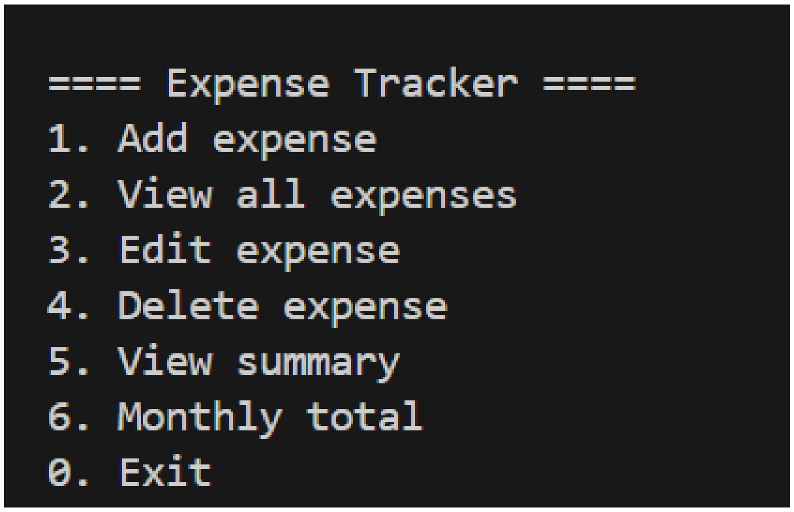
Major Classes

- **Expense.java** → Model class
- **ExpenseService.java** → Handles add/remove/view logic
- **Main.java** → User interface + menu

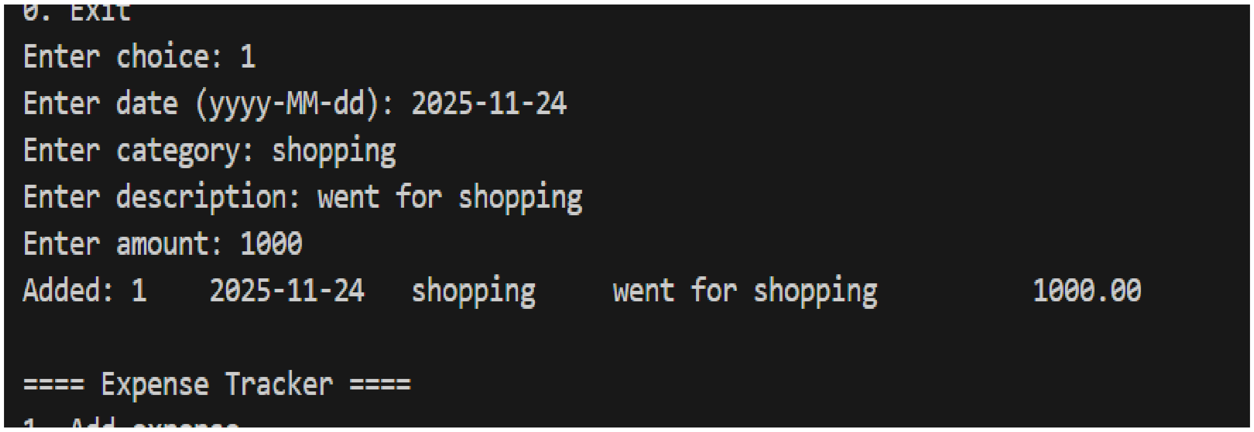
Storage

- Runtime: ArrayList
- Persistent: expenses.txt (optional)

Screenshots / Results



```
==== Expense Tracker ====
1. Add expense
2. View all expenses
3. Edit expense
4. Delete expense
5. View summary
6. Monthly total
0. Exit
```



```
0. Exit
Enter choice: 1
Enter date (yyyy-MM-dd): 2025-11-24
Enter category: shopping
Enter description: went for shopping
Enter amount: 1000
Added: 1    2025-11-24   shopping     went for shopping          1000.00

==== Expense Tracker ====
1. Add expense
```

Testing Approach

Unit Testing

- Add expense function
- Delete expense function
- Summary calculations

Validation Testing

- Negative amount
- Invalid date
- Empty description

File Handling Testing (optional)

- Check if file updates correctly

User Acceptance Testing

- Ensure menu options work as expected

Challenges Faced

- Managing input validation in console
- Handling incorrect user choices
- File handling complexities
- Maintaining clean separation between UI and business logic

Learnings & Key Takeaways

- Improved understanding of Core Java
- Learned modular programming and code reuse
- Better grasp of data structures (ArrayList)
- Understanding of CRUD operations
- Importance of clean UI flow even in console applications
- Learned how to break a problem into smaller modules

Future Enhancements

- Add database support (MySQL/SQLite)
- Add login system for multiple users
- Add monthly budget alerts
- Provide CSV/PDF export
- Build GUI using Swing/JavaFX
- Add graphs using JavaFX charts