

Expense Management

Tracker Project Report

Project Title: Expense Tracker for College Students

Course: Introduction to Problem Solving and Programming

Name: Anusha Sinha

Registration No. 25BAI10793

Institution: VIT Bhopal University

Date: 23 November 2025

1. Introduction

Managing daily expenses is crucial for student life, especially for college students living away from home. Proper financial management helps students stick to their budgets and keep track of their spending. This project, "Expense Tracker for College Students," offers an easy and effective way to record, manage, and monitor daily expenses using a Python-based backend system that stores data in a CSV file. The project meets the course requirement for creating an original project that shows an understanding of Python basics: file handling, functions, loops, conditional statements, and program structure.

2. Problem Statement

Many college students struggle to keep track of their daily expenses and often overspend, leading to poor financial management. Without a good tracking system, they cannot:

- Record daily spending
- Review past expenses
- Analyze spending patterns

A simple, lightweight system is needed to help users store, view, and analyze expenses without the complexity of complicated software or databases.

3. Objectives

The main objectives of this project are:

- To design a straightforward Python tool for managing expenses.
- To provide essential features such as adding expenses, viewing all expenses, and calculating monthly totals.
- To store data in a CSV file for easy access and readability.
- To help college students understand their spending habits.

4. Functional Requirements

The project includes the following key functional modules:

.1 Add Expense

This allows the user to enter the date, category, amount, and notes and saves this information in a CSV file.

.2 View All Expenses

This displays all expenses saved in the CSV file.

.3 Monthly Expense Summary

This calculates and shows the total expenses for a chosen month and year.

5. Non-Functional Requirements

At least four non-functional requirements from the project guidelines are included:

.1 Usability

The program has a simple, menu-driven interface for ease of use.

.2 Reliability

Data saved in CSV files remains accessible even after the program is closed.

.3 Maintainability

The program is organized with modular functions, making it easy to update or expand.

.4 Performance

The operations are light and execute quickly, even on basic hardware.

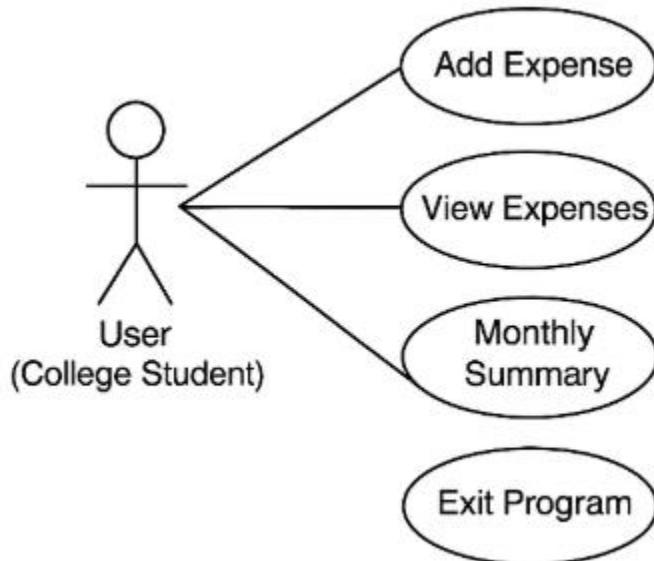
6. System Architecture

The system consists of three main components:

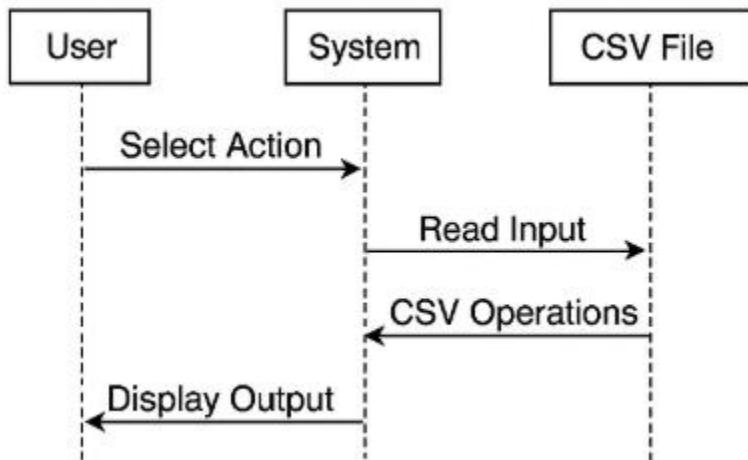
- Input Module: Collects user input for expenses.
- Processing Module: Calculates totals and checks data.
- Storage Module: Uses a CSV file for persistent data storage.

7. Design Diagrams

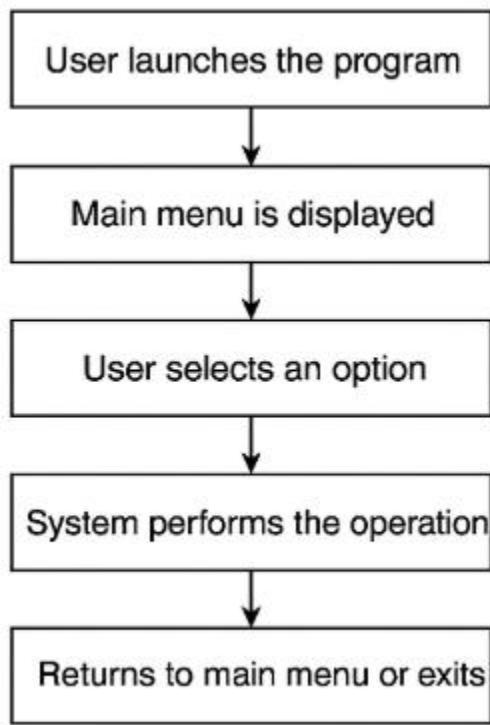
1 Use Case Diagram



2 Workflow Diagram



3 Sequence Diagram



4 ER Diagram

EXPENSES
Date
Category
Amount
Note

8. Design Decisions & Rationale

CSV Storage: Chosen for its simplicity and first-semester constraints.

Python Functions: Improve readability and maintainability.

Menu-Driven Program: Easiest for beginners and provides clear navigation.

No External Libraries: Fits project restrictions and supports basic Python skill development.

9. Implementation Details

The program is built in Python using:

- File handling (open, read, write)
- CSV module
- Functions and loops
- Conditional statements

The main modules include:

- expenses_add()
- see_expenses()
- monthly_expenses()

10. Screenshots / Results

Step 1

```
PS C:\Users\Home\EXPENSE TRACKER> & C:/Users/Home/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/Home/EXPENSE TRACKER/main.py"
1. Add Expense
2. View Expenses
3. Check Monthly Total
4. Exit
Choose option: 1
```

Step 2

```
PS C:\Users\Home\EXPENSE TRACKER> & C:/Users/Home/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/Home/EXPENSE TRACKER/main.py"
1. Add Expense
2. View Expenses
3. Check Monthly Total
4. Exit
Choose option: 1

--- Add New Expense ---
Enter date (YYYY-MM-DD): 2023-10-01
```

Step 3

```
--- Add New Expense ---
Enter date (YYYY-MM-DD): 2025-12-03
Enter category (Food, Travel, Study, etc.): Food
Enter amount: 100
Enter note (optional): Hostel Lunch
Expense added successfully!
```

Step 4

```
--- All Expenses ---
['date', 'category', 'amount', 'note']
['2025-01-10', 'Food', '120', 'Canteen lunch']
['2025-01-10', 'Travel', '40', 'College bus']
['2025-01-11', 'Study', '250', 'Notebook + pen']
['2025-01-12', 'Snacks', '60', 'Tea + samosa']
['2025-01-15', 'Entertainment', '150', 'Movie']
['2025-01-18', 'Food', '100', 'Hostel dinner']
['2025-01-20', 'Travel', '80', 'Auto fare']
['2025-02-19', 'Food', '75', 'Bus fare']
['2025-12-03', 'Food', '100', 'Hostel Lunch']
```

Step 5

```
1. Add Expense
2. View Expenses
3. Check Monthly Total
4. Exit
Choose option: 2

--- All Expenses ---
['date', 'category', 'amount', 'note']
['2025-01-10', 'Food', '120', 'Canteen lunch']
['2025-01-10', 'Travel', '40', 'College bus']
['2025-01-11', 'Study', '250', 'Notebook + pen']
['2025-01-12', 'Snacks', '60', 'Tea + samosa']
['2025-01-15', 'Entertainment', '150', 'Movie']
['2025-01-18', 'Food', '100', 'Hostel dinner']
['2025-01-20', 'Travel', '80', 'Auto fare']
['2025-02-19', 'Food', '75', 'Bus fare']
['2025-12-03', 'Food', '100', 'Hostel Lunch']
-----
```

Step 6

```
1. Add Expense
2. View Expenses
3. Check Monthly Total
4. Exit
Choose option: 3

--- Total Monthly Expenses ---
Enter month (MM): 01
Enter year (YYYY): 2025
Money spent in 01/2025: ₹800.0
```

Step 7

```
1. Add Expense  
2. View Expenses  
3. Check Monthly Total  
4. Exit  
Choose option: 4  
PS C:\Users\Home\EXPENSE TRACKER> █
```

11. Testing Approach

The system was tested using:

- Multiple sample inputs
- Boundary cases (empty note, large amount)
- Invalid month/year inputs
- Checking CSV file updates

All functionalities performed as expected.

12. Challenges Faced

Understanding how to add data to CSV without overwriting

Handling date format consistency

Ensuring monthly totals matched the correct data

Designing a simple but effective menu-driven interface

13. Learnings & Key Takeaways

Through this project, I learned:

- Practical use of Python's CSV module
- Modular programming
- File handling operations
- Designing user-friendly console applications
- The importance of documentation and planning

14. Future Enhancements

Potential improvements include:

- Adding an expense search feature
- Editing or deleting entries
- Category-wise spending report
- Graphical visualization (using a GUI or charts)
- User authentication for privacy

15. References

Python Official Documentation (CSV module)

Course lecture notes and materials