SIMPLE EXPENSE TRACKER

Submitted By - Hrituraj Saha
Enrollment ID - PD25100001
Date of Submission - 11/09/2025

1. PROJECT OVERVIEW

The Simple Expense Tracker is a Python-based console application designed to help users efficiently record and analyze their daily expenses. This project was created with the intent of demonstrating not only functional requirements (adding, viewing, and analyzing expenses) but also industry-level non-functional requirements such as readability, scalability, error handling, and user experience. The tracker allows users to categorize their spending, view total expenses, and receive budget alerts when category limits are exceeded. Additionally, password protection ensures basic security, and a summary report is generated before exit. This project is highly extendable, making it suitable for both beginners and as a foundation for advanced financial management tools.

2. FEATURES IMPLEMENTED

The following features have been successfully implemented as part of this project -

1. Add Expenses – Users can add expenses under a chosen category.
2. View Total Expenses – Displays the total of all expenses added so far.
3. View Expenses by Category – Provides detailed expense reports by category.
4. Exit Gracefully – Exits with a summary report.
5. Password-Protected Login – Simple login system requiring authentication.
6. Budget Alerts – Alerts the user if they exceed pre-set category budgets.
7. Summary Report – Displays highest and lowest spending categories on exit.

3. CODE EXPLANATION

The program is designed in a modular structure using functions for clarity and maintainability.

- Global Variables
  - CATEGORIES → List of expense categories.
  - BUDGETS → Dictionary storing category budget limits.
  - EXPENSES → List that stores all expenses (amount + category).
- Functions
  - login() → Implements password-protected login with 3 attempts.
  - add_expense() → Handles adding an expense under a chosen category with validation.
  - view_total() → Displays total spending across categories.
  - view_by_category() → Breaks down spending category-wise.
  - summary_report() → Generates a final summary of total, highest, and lowest expenses.
  - main_menu() → Controls the user interface loop.
- Logic Flow
  - a. Program starts → user must log in.
  - b. Main menu displays options.
  - c. Based on input, corresponding functions are called.
  - d. Program exits gracefully with a final summary.

4. ERROR HANDLING & EDGE CASES

The program includes several robust error-handling mechanisms -

- Invalid Inputs → If a user enters a string instead of a number, the program uses try-except to catch the error.
- Invalid Category Choice → If the user selects a non-existent category, an error message is shown instead of crashing.
- Negative/Zero Amounts → Expenses with <= 0 values are rejected.
- Login Security → Only 3 login attempts are allowed. If all fail, the program exits.
- Empty Records → If no expenses are recorded, the summary report handles this gracefully by displaying "No expenses recorded."

These measures ensure that the application never crashes unexpectedly and always provides meaningful feedback to the user.

5. CHALLENGES & LEARNINGS

Challenges Faced -

1. User Input Validation
   - Initially, the program crashed when users entered invalid values (e.g., text instead of numbers).
   - Solution - Implemented try-except blocks and logical checks.
2. Budget Limit Alerts
   - It was tricky to implement real-time budget checking after each expense.
   - Solution - Added a calculation that checks the total spent in a category each time a new expense is added.

Learnings -

- Learned the importance of modular coding with functions for clarity and reuse.
- Gained hands-on experience in error handling and input validation.
- Improved understanding of Python dictionaries and list comprehensions for aggregating data efficiently.
- Learned how to structure a user-friendly console menu system.

6. FUTURE SCOPE

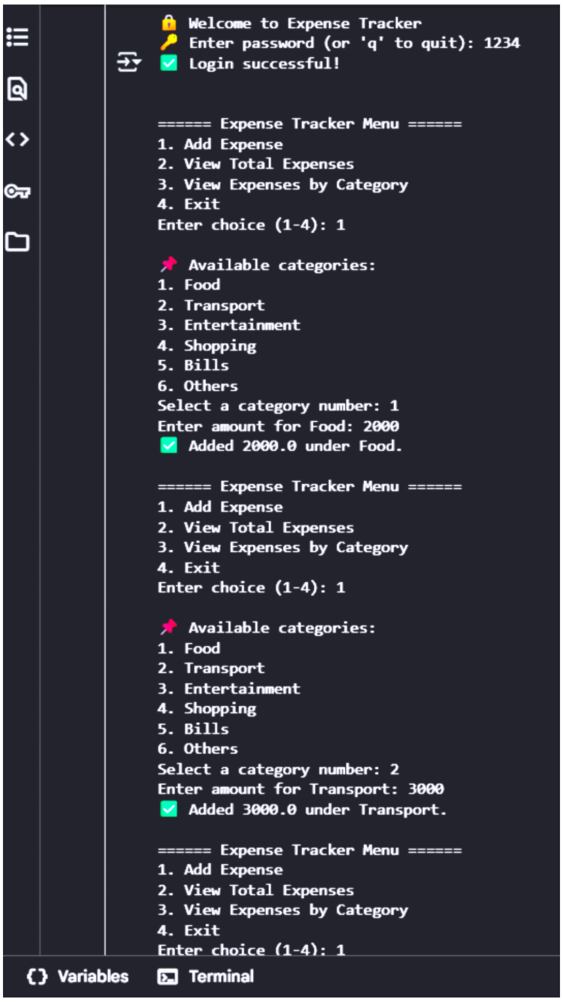
If given more time, the following features could enhance the project -

- Data Persistence – Save expenses to a CSV/JSON file so data is not lost after restart.
- Visualizations – Graphical charts (bar/pie) to show category-wise spending.
- Custom Budgets – Allow users to set their own category limits dynamically.
- Multi-user Support – Each user can log in with a separate account and track their expenses.

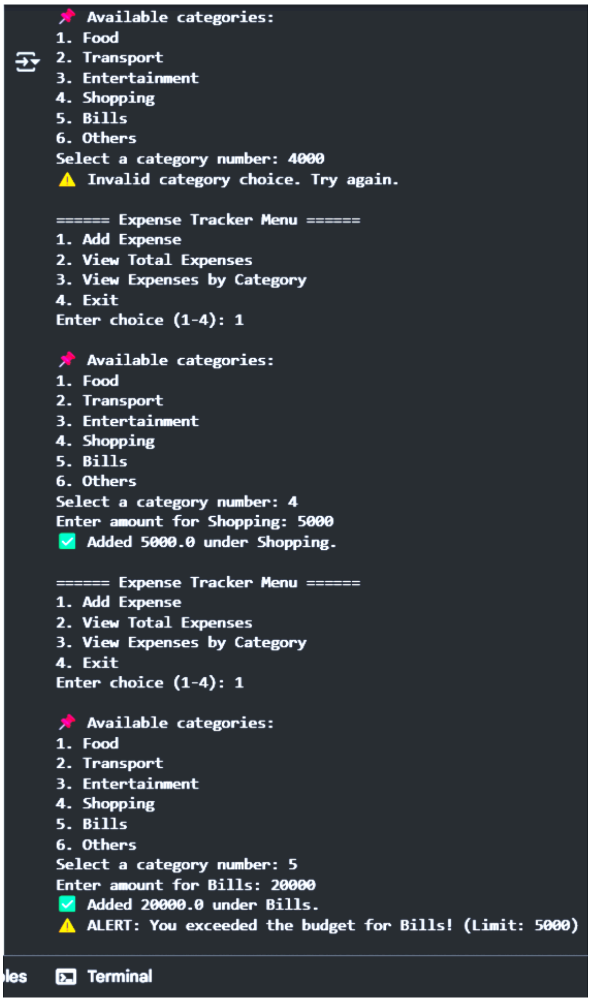
7. CONCLUSION

The Simple Expense Tracker successfully demonstrates both functional and non-functional software engineering requirements. It is a small but extendable project that balances practicality with scalability. By completing this project, I gained valuable insights into Python programming best practices, error handling, modular design and user experience.

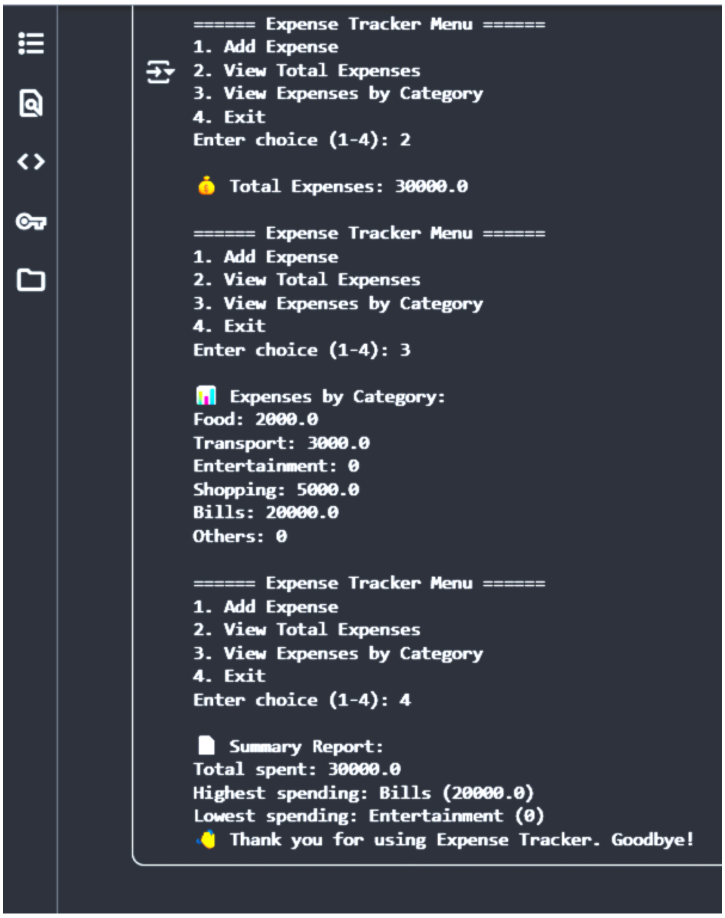
8. OUTPUT SCREENSHOTS



```
🔒 Welcome to Expense Tracker
🔑 Enter password (or 'q' to quit): 1234
✅ Login successful!

====== Expense Tracker Menu ======
1. Add Expense
2. View Total Expenses
3. View Expenses by Category
4. Exit
Enter choice (1-4): 1

📌 Available categories:
1. Food
2. Transport
3. Entertainment
4. Shopping
5. Bills
6. Others
Select a category number: 1
Enter amount for Food: 2000
✅ Added 2000.0 under Food.

====== Expense Tracker Menu ======
1. Add Expense
2. View Total Expenses
3. View Expenses by Category
4. Exit
Enter choice (1-4): 1

📌 Available categories:
1. Food
2. Transport
3. Entertainment
4. Shopping
5. Bills
6. Others
Select a category number: 2
Enter amount for Transport: 3000
✅ Added 3000.0 under Transport.

====== Expense Tracker Menu ======
1. Add Expense
2. View Total Expenses
3. View Expenses by Category
4. Exit
Enter choice (1-4): 1
```



```
📌 Available categories:
1. Food
2. Transport
3. Entertainment
4. Shopping
5. Bills
6. Others
Select a category number: 4000
⚠️ Invalid category choice. Try again.

====== Expense Tracker Menu ======
1. Add Expense
2. View Total Expenses
3. View Expenses by Category
4. Exit
Enter choice (1-4): 1

📌 Available categories:
1. Food
2. Transport
3. Entertainment
4. Shopping
5. Bills
6. Others
Select a category number: 4
Enter amount for Shopping: 5000
✅ Added 5000.0 under Shopping.

====== Expense Tracker Menu ======
1. Add Expense
2. View Total Expenses
3. View Expenses by Category
4. Exit
Enter choice (1-4): 1

📌 Available categories:
1. Food
2. Transport
3. Entertainment
4. Shopping
5. Bills
6. Others
Select a category number: 5
Enter amount for Bills: 20000
✅ Added 20000.0 under Bills.
⚠️ ALERT: You exceeded the budget for Bills! (Limit: 5000)
```



```
====== Expense Tracker Menu ======
1. Add Expense
2. View Total Expenses
3. View Expenses by Category
4. Exit
Enter choice (1-4): 2

💰 Total Expenses: 30000.0

====== Expense Tracker Menu ======
1. Add Expense
2. View Total Expenses
3. View Expenses by Category
4. Exit
Enter choice (1-4): 3

📊 Expenses by Category:
Food: 2000.0
Transport: 3000.0
Entertainment: 0
Shopping: 5000.0
Bills: 20000.0
Others: 0

====== Expense Tracker Menu ======
1. Add Expense
2. View Total Expenses
3. View Expenses by Category
4. Exit
Enter choice (1-4): 4

📄 Summary Report:
Total spent: 30000.0
Highest spending: Bills (20000.0)
Lowest spending: Entertainment (0)
👋 Thank you for using Expense Tracker. Goodbye!
```