Expense Tracker – Version 1 Review

1. Project Overview

The Expense Tracker is a Python-based application designed to help users record, view, edit, and analyze their daily expenses. It provides features such as data filtering, category summaries, monthly breakdowns, and exporting reports. Version 1 was implemented as a command-line interface (CLI) application, using CSV files to store data, pandas for data management, and tabulate for clean table display.

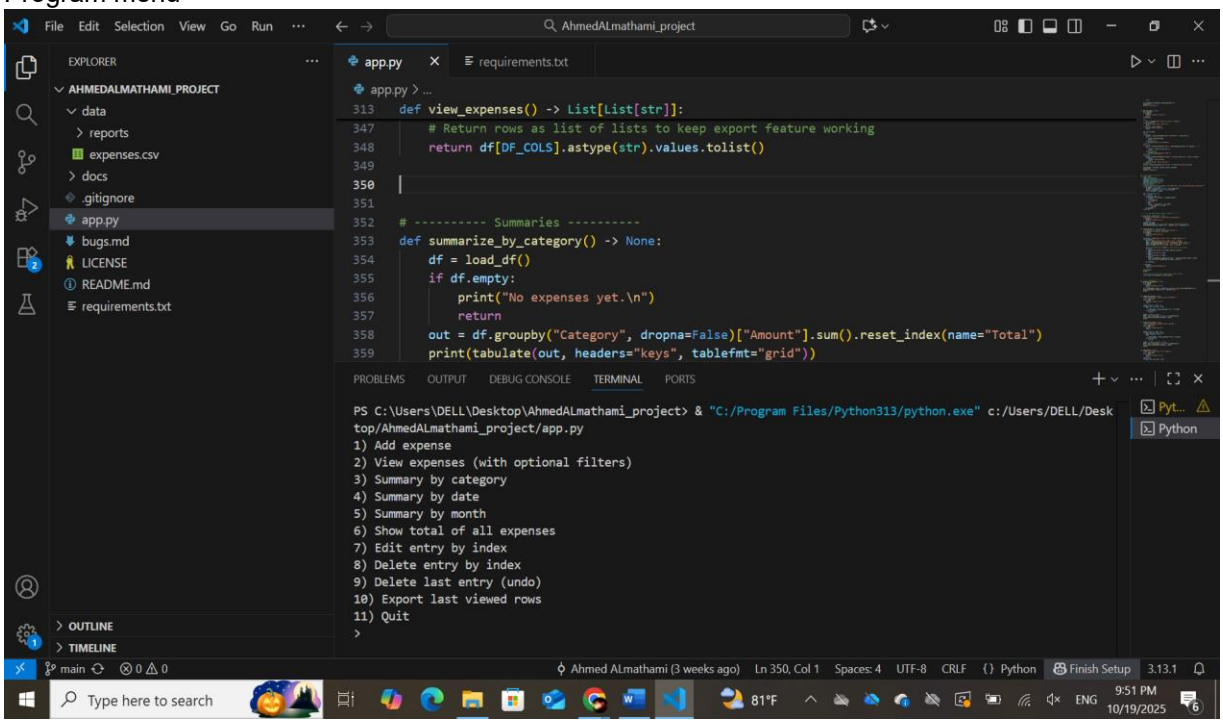
2. Features Implemented

- Add new expense entries (Date, Amount, Category, Description).
- View all expenses with optional filters (category, date range, keyword search).
- Edit or delete any expense entry by index.
- Summaries by category, date, and month.
- Export filtered reports to timestamped CSV files.
- Graceful exit handling (Ctrl+C).
- Data stored and loaded using pandas DataFrame for improved performance and flexibility.

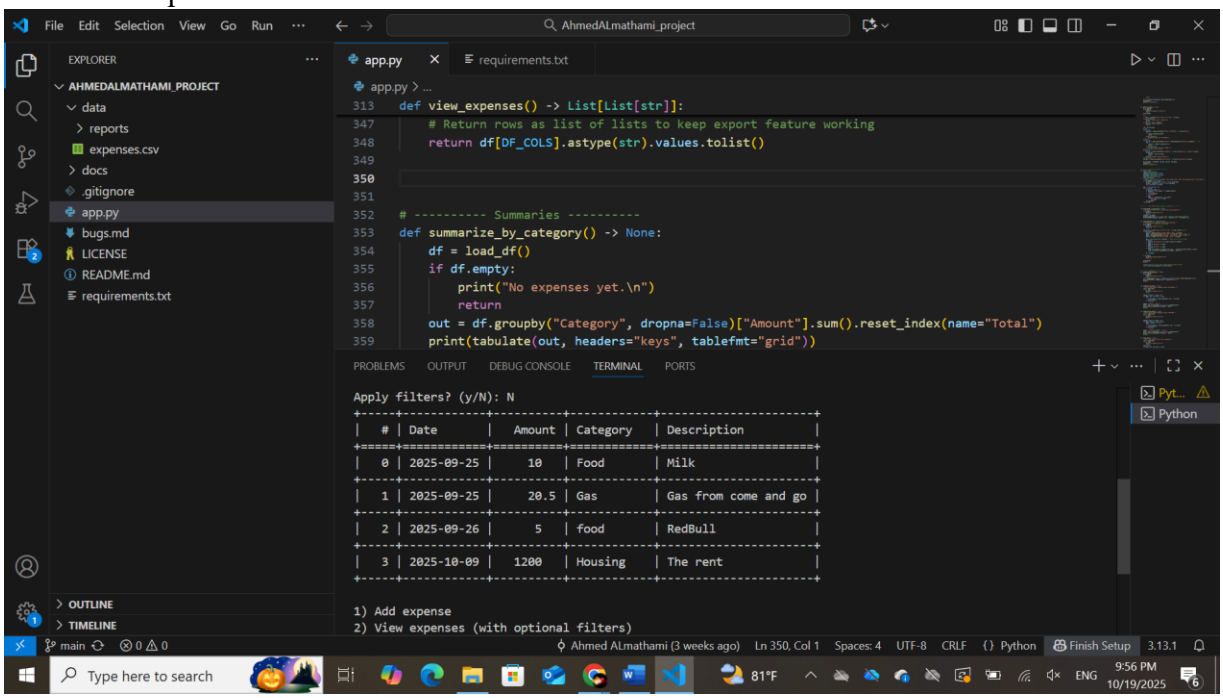
3. Demonstration

The following screenshots demonstrate the working state of the program:

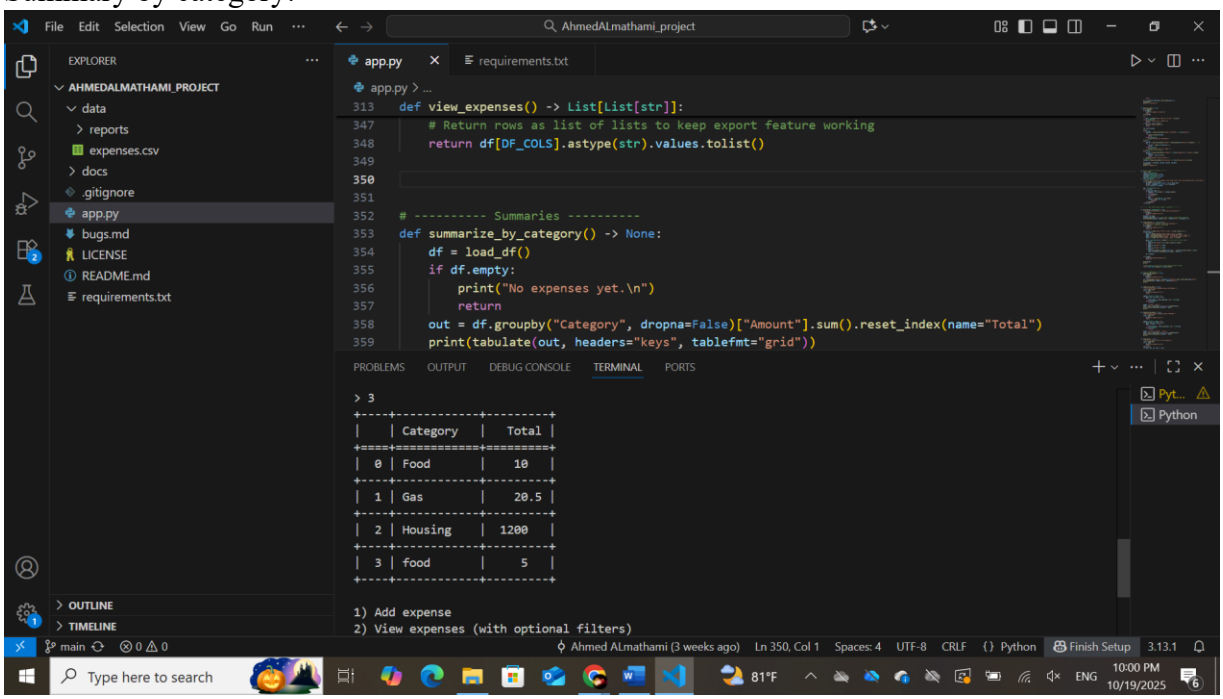
- Program menu



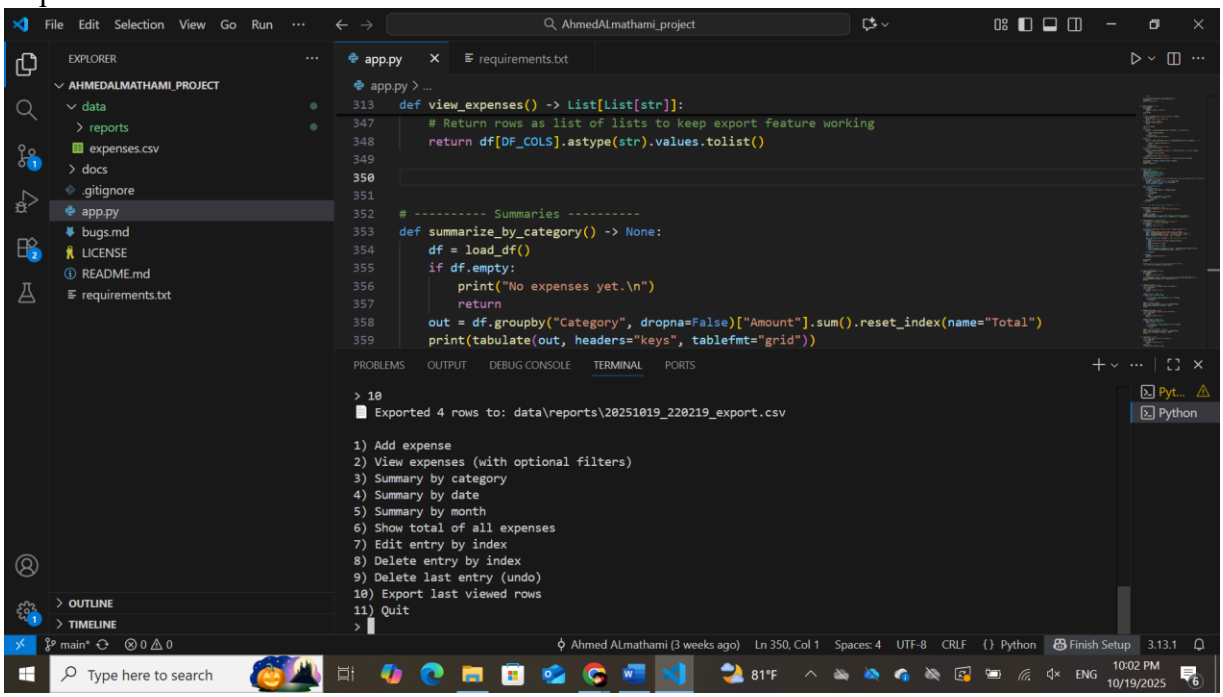
- Tabulated expense view:



- Summary by category:



- Export confirmation:



4. Issues I faced and Solutions:

- Data structure: Initially used nested lists, which made filtering difficult. Solved by switching topandas DataFrame.
- Misaligned text output: Resolved by using tabulate module for clean table formatting.
- Category input: Previously allowed blanks, now validated to avoid empty values.
- Graceful exit: Added sys.exit() handling to prevent abrupt program termination.

5. Help Needed:

I just would like your advice about your recommendations for structuring Streamlet components for Version 2 GUI.

6. Milestones for Version 2:

- Milestone 1 : Streamlit Setup: Create Streamlit interface with sidebar input fields for adding expenses.
- Milestone 2 : Data Integration: Connect Streamlit interface with pandas-based backend for live updates.
- Milestone 3 : Visualization: Add summary charts (totals by category, monthly trends).

- Milestone 4 :Export/Download: Implement export and download buttons directly in Streamlit.
- Milestone 5 : Polish and Testing: Improve layout, add icons, and perform usability testing.

7. Self Reflection

I am satisfied with my progress so far. I was able to complete all core features and make my code more organized and maintainable. Using pandas and tabulate was a big improvement from my earlier approach. Some tasks took longer than expected, especially understanding pandas operations, but I learned a lot from the process. I believe I can finish the Streamlit version within the next few weeks. The biggest 'light bulb' moment was realizing how pandas can simplify filtering and summarizing operations that were previously done manually.