

Expense Tracker – Revised Project Specification

1. General Description :

The Expense Tracker project is a Python-based application designed to help users record, manage, and analyze their expenses. The initial version was built as a command-line interface (CLI) program using CSV storage, and Version 1 successfully integrated pandas and tabulate for improved data handling and display. Version 2 will focus on transitioning the application into a Streamlit-based web interface for a more user-friendly experience.

2. Implemented Features :

- Data management migrated from nested lists to pandas DataFrame for easier filtering and summaries.
- Integrated tabulate for clean, formatted tables in the terminal.
- Improved input validation (no blank categories, decimal normalization for Amount).
- Added delete confirmation and graceful exit (Ctrl+C, sys.exit).
- Enhanced export confirmation messages and file structure (data/reports/).
- Organized code and repository structure for easier version control.

3. Proposed Features :

The next phase (Version 2) aims to build a Streamlit-based GUI application that enhances usability and visualization capabilities. This interface will allow users to add expenses, filter data, and view visual summaries through an interactive dashboard.

Planned new features include:

- Streamlit interface with sidebar form for adding new expenses.
- Dynamic table view showing all or filtered expenses (using pandas).
- Summary visualization: category totals and monthly spending trends via charts.
- Download/export current view as CSV directly from the GUI.
- Optional login screen (single-user mode for simplicity).

4. Technical Flow :

- Data Storage: CSV file (data/expenses.csv) remains the main persistent data source.
- Data Handling: pandas for reading, writing, and summarizing data.
- CLI Version: Data displayed using tabulate for text output.
- GUI Version (Proposed): Streamlit frontend connected to pandas backend for real-time updates.
- Charts: matplotlib or plotly integrated within Streamlit for visual summaries.
- Export: Ability to download filtered or full datasets through Streamlit download button.

5. Milestones for Version 2 :

- Milestone 1: Streamlit Setup: Initialize app structure with basic sidebar and layout.
- Milestone 2: Data Integration: Connect pandas DataFrame logic with Streamlit UI.
- Milestone 3: Visualization: Add charts for category and monthly summaries.
- Milestone 4: Export/Download: Add CSV download feature for filtered views.
- Milestone 5: Testing & Polish: Improve interface layout and conduct testing.

6. Self-Assessment and Feasibility

The revised plan is feasible for the remaining weeks. The core data logic is complete, and transitioning to a Streamlit GUI primarily involves designing the interface and connecting it to the existing pandas backend. Potential challenges include setting up Streamlit components and managing interactive charts, but available documentation and tutorials make these achievable goals.