

Semester-long Project Iteration 1——Expenses Audit Tool

Yanshi Liu

10/25/2025

1 Project Description and Motivation

Managing expense reports can be frustrating and time-consuming. People sometimes make mistakes—like submitting the same invoice twice, entering dates that fall on weekends, or spending just below the approval limit to avoid extra review. These issues make it hard to track spending accurately and fairly.

My project, “Expenses Audit Tool,” is designed to solve this problem. It acts like a small robot auditor that automatically reviews expense data and flags anything suspicious. The program reads information from CSV files, detects duplicate or weekend transactions, and identifies high-value expenses that may require extra attention.

The motivation behind this project is to make financial auditing faster, more reliable, and less tedious. Instead of manually checking hundreds of records, users can rely on this tool to catch potential problems instantly. It’s a simple, honest, and tireless helper that supports better financial transparency.

2 Project Functionalities

My project includes five main features that work together to make expense auditing easier and more accurate:

1. Read CSV Data Automatically – The program can read expense files from different systems and handle various encodings, such as UTF-8 or ISO-8859-1. This ensures that the data is loaded correctly even if the file format is inconsistent.
2. Find Duplicate Invoices –It checks for repeated invoice numbers or identical expense descriptions. This helps detect cases where the same invoice may have been submitted twice.
3. Flag Weekend Transactions – The tool identifies expenses dated on Saturdays or Sundays, which are unusual for most business activities and might need review.

4. Check High-Value Expenses – The program highlights any transactions that exceed a certain limit (for example, \$5,000) or are suspiciously close to it.
5. Web Interface with Flask – A simple and user-friendly web application allows users to view results, filter flagged records, and download reports easily.

3 Challenges and Solutions

During the development of my Expenses Audit Tool, I faced several challenges that helped me improve both my technical and problem-solving skills.

1. File Encoding Issues

Some CSV files failed to open correctly because of different text encodings. For example, files saved from Excel might use UTF-8, UTF-16, or ISO-8859-1. To solve this, I implemented a function that automatically tries multiple encodings until it successfully reads the file. This made the program more reliable and user-friendly.

2. Messy and Inconsistent Data

Many expense files contained extra symbols like dollar signs (\$), commas, or inconsistent date formats. These caused errors during calculation or comparison. I created data cleaning functions to remove unwanted characters, convert currency values to floats, and standardize date formats using Python's "datetime" module.

3. Flask Web Interface Setup

Building a web app was completely new to me. At first, connecting backend logic with the web interface was confusing. After reading Flask documentation and tutorials, I learned how to use routes, templates, and dynamic rendering. In the end, I successfully created a simple but functional web page to display and download flagged results.

4. Debugging and Error Handling

While testing, I often encountered missing keys or type errors caused by bad data inputs. I solved this by adding exception handling (try-except blocks) and meaningful error messages, which made debugging much faster.

4 Future Work

In the future, I plan to expand the Expenses Audit Tool with more data-cleaning and analysis features. One idea is to add a “Missing Fields Check” that automatically flags records with blank or incomplete information, so users can fix data issues faster.

Another improvement is a currency conversion feature, which can convert all expenses into U.S. dollars for easier comparison across different regions.

I also plan to create a summary report function that shows total spending, average expense, and key statistics in one simple view. Beyond these, I hope to connect the tool with real accounting systems and make the web interface even more user-friendly, so anyone — even without programming skills — can use it easily and trust the results.

5 Acknowledgments

I would like to thank my instructor and teaching assistants for their guidance and support throughout this project. Their feedback helped me understand both programming logic and real-world problem solving. I also want to thank my classmates for their encouragement and ideas, which made this learning experience more enjoyable and meaningful.