COSC 1104 Assignment 2

Part 1: Identify the Problem

Project 1: Python Group Expense Splitter Script

Managing group expenses during trips can be challenging, especially when each participant pays for different things at different times. This often leads to manual calculations, which can be tedious and prone to errors. Developing a Python script to handle this process automates the calculations, ensuring fair and accurate expense distribution.

The group expense splitter script works by taking in a data file (e.g., YAML) containing information about who paid for what and how much each expense was. It then processes the data, calculates individual balances, and shows how much each participant owes or is owned. This kind of script is valuable as it provides transparency and convenience for groups who may not want to rely on external apps or sites. It allows customization to suit personal or group specific needs.

Potential Python Libraries for Use:
"yaml " : For reading the input data file.
" pandas " : To handle data structures and perform calculations.
" argparse " : To provide commandline functionality if needed.
" tabulate " : For displaying output in a structured format.

Complexity and Feasibility:

The script is of moderate difficulty due to data parsing and balance calculations. Once set up, it's easy to customize or extend, such as adding a web interface later.

https://lexdsolutions.com/2024/02/python-group-expense-splitter-script/

Project 2: Price and Review Comparison Tool for Ecommerce

Online shopping often involves comparing prices across various platforms like Amazon, eBay, and Best Buy. A Pythonbased price and review comparison tool automates this process by fetching product data, including prices and reviews, from different ecommerce websites and presenting it in an easytoread format. This helps users make informed decisions and potentially save money by choosing the best deal.

The tool's primary function would be to:
Fetch product data from multiple sources using web scraping.
Clean and process this data for comparison.
Display the prices, ratings, and product information in a single interface.
Notify users via SMS or email when a product's price drops.

Potential Python Libraries for Use:
"BeautifulSoup" and "requests" : For web scraping to gather product data.
"pandas" : For data cleaning and processing.
"matplotlib" or "plotly" : To visualize price trends.

COSC 1104 Assignment 2

"smtplib" or "Twilio API" : For sending notifications.
"selenium" : For handling dynamic web pages.

Complexity and Feasibility:

Developing this tool can be moderately challenging, especially due to the intricacies of web scraping and maintaining compatibility with ecommerce websites. Using tools like "webautomation.io" can simplify the scraping process. A basic version of the tool could be implemented relatively easily, while more advanced features like realtime monitoring and notifications might add complexity.

https://webautomation.io/blog/how-to-create-price-comparison-tool-with-beautiful-soup/

Part 2 : Solve the Problem Using Python

Problem Summary

The objective is to write a Python script that computes the sums owing or credited to each member of a group based on a YAML file that contains a list of expenses paid by each member of the group. The final credits and debts should be shown in the script, along with recommendations for who should pay whom to settle the outstanding balance.

Approach to the Solution
Finding a fair way to divide the costs entails reading the data from a YAML file, processing it, and calculating the total expenses for each individual. The script ought to manage possible mistakes such names that don't match.

Python Libraries

- "yaml" : To read the input YAML file.
- "pandas" (optional) : To organize and manipulate data easily.
- "tabulate" : For formatted output (optional).
- "sys" : To handle command-line arguments.
- "argparse" : To make the script more user-friendly.

The group expense-splitting procedure is automated by this script, which speeds it up and eliminates errors. For wider use, it can be expanded to incorporate functions like file exporting and web-based interface integration.

COSC 1104 Assignment 2

Reflection on the Group Expense Splitter Script Development

The decision to create a group expense splitter was both sensible and difficult, offering a worthwhile undertaking that fit in nicely with typical real-world situations. Building my own solution from scratch provided a unique opportunity to explore programming thinking and improve my Python skills, even if there are already websites and mobile apps for expense splitting. In order to integrate control structures, data validation, and computations while maintaining the code's usability and efficiency, this project struck a balance between simplicity and complexity.

"Difficult Aspects"

- Did you pick a suitably challenging problem to solve? Or was it too easy, or too hard?

The implementation of strong error handling and validation logic was the most difficult aspect of this project. It was essential to make sure that all references were legitimate and that the input data was formatted consistently. For instance, there should be a clear and easy-to-use error message if a person in the "split_with" section did not match any of the "people" provided. It took careful planning to manage these validations while preserving the modularity and readability of the code.

Determining how to accurately assess and balance expenses when they were divided among several group members was another difficulty. To guarantee accuracy and fairness in the product, it required rigorous consideration of how to allocate funds and precisely track credits and obligations.

“Valuable Lessons”

- What was the most valuable thing you learned from this assignment?

The significance of data validation and modular coding in programming was reaffirmed by this effort. The code became easier to manage and more maintainable by decomposing the solution into functions like "read_expenses," "validate_expenses," and "calculate_expenses." Additionally, using Python libraries like "yaml" served as a helpful refresher in handling configuration files and parsing structured data.

I learned a lot about using fundamental arithmetic in programming for practical situations and effectively managing data structures from the logic used to determine the net balance (who owes or is owed). I also learned how basic scripts might be expanded into more complete solutions that would be able to connect with more intricate financial systems or web-based apps.

COSC 1104 Assignment 2

“Enjoyment and Engagement”

- Did you *enjoy* the experience of working on this assignment? Why?

I had a great time working on this task. The project called for creativity and critical thinking and was a good mix of problem-solving and coding. Seeing the code develop from a basic idea to a working script that could process actual data inputs was satisfying to me. Developing a script that may streamline a manual and error-prone process gave a certain sense of accomplishment.

The issue was not simple, and I valued the chance to solve problems that came up throughout development, such addressing disparities in the "split_with" section or edge instances when no expenses were provided. Every obstacle I overcame strengthened my problem-solving abilities and increased my sense of accomplishment.

In summary, this experience was enlightening and brought to mind the reasons I like coding: it provides the opportunity to solve real-world issues, create tools that make life easier, and participate in an educational process that is constantly open to new insights. In addition to developing a potentially helpful tool, this project helped me strengthen fundamental Python programming abilities that I can apply in other projects.

REFERENCE:

https://webautomation.io/blog/how-to-create-price-comparison-tool-with-beautiful-soup/

https://lexdsolutions.com/2024/02/python-group-expense-splitter-script/