9/22/2025

GROUP 9

PYTHON ADVANCED COHORT 28

EXPENSE SPLITTER APP

INTRODUCTION

Our project the expense splitter app provides a simple interface for members of a group:

* Add participants.
* Record expenses by specifying payer, participants, and amount.
* View the resulting balances for each group member.
* Generates a minimal set of “who owes who” transactions to settle debts.

The design is simple and ensures users can track and settle group expenses efficiently whatever it may be for.

MVP IMPLEMENTATION

Our MVP was defined as the ability to create a group, record expenses, and view simplified settlements. We achieved this by following some steps:

* Core logic: Implemented Expense, Group, and Settlement classes to encapsulate functionality.
* Settlement algorithm: Wrote a compute\_settlemnets () function that calculates each member balance and produces the fewest transactions needed to clear debt.
* User Interface: A minimal GUI form was created to add expenses, display balances, and provide settlement suggestions.
* Repository Setup: Our GitHub repository included a README, pipenv environment, role assignments, and issue tracker usage. Frequent commits were made with descriptive messages to show progress.

By midday day 3, we recorded a demo video of our core logic functioning. On day 5, we finalized most of our important logics and on day 6 we recorded the final demo video and finalized the core logic and README.

TESTING

We wrote three pytests for core settlement logic that verifies the correctness of balances after multiple expenses. Also, to ensure settlement suggestions matched the minimal transaction criteria.

The test was successful and assured us in the correctness of our algorithm

CHALLENGES AND SOLUTIONS

* Algorithm Complexity: computing and ensuring the core settlement algorithm was tricky. We solved this by isolating the core logic in a dedicated function and testing it.
* Group Coordination: communicating and managing tasks across multiple days was challenging. Using GitHub issues helped us stay organized.
* UI Clarity: Many ideas for how the UI should be was talked about and was contradicting. We finally agreed on a simple design that worked for everyone.

PEER REVIEW AND FEEDBACK

Our group carried out a peer review on day 6, giving our input on other groups project and receiving constructive feedback on ours. We emphasized on the core logic working and the clarity of the UI and test coverage while reviewing others and received feedback on our project and improved it where needed.

REFELCTION AND LEARNING OUTCOMES

This project helped us strengthen skills in:

* Object oriented Programming
* Collaboration and Version Control
* Testing as an integral part of development
* Time Management

CONCLUSION

The expense splitter app successfully met the criteria needed and demonstrated the programming principles taught in the course. We learned a lot throughout and expanded in knowledge and creativity.