Expense Sharing System — Project Report

Overview

ExpenseSharing is a simple Python-based command-line tool that helps track and settle shared expenses among a group of people. It uses **equal splitting methodology** and calculates how much each person owes or needs to be reimbursed, ultimately simplifying the final settlement.

The Methodology Used for Expense Splitting:

This system uses a **simple equal contribution model** for splitting expenses among a group of members. Each time an expense is recorded:

- The **payer's** balance increases by the full amount they paid.
- Each **participant's** balance decreases by an equal share of the expense.

Formula Used:

If **amount** is paid by **payer** for **n** people:

- Each person pays **amount / n**
- payer gets credited with the full amount

Dataset Details and Preprocessing Steps:

Input Data:

The system takes real-time inputs from the user:

- **Member Names** (initial list)
- **Expense Entries**:
    - Name of the **payer**
    - **Amount** paid
    - List of **beneficiaries**

Preprocessing:

- Strips extra spaces from names
- Converts amounts to **float** for consistent calculation

Tracks balances in a dictionary:

self.balances = {member: 0 for member in members}

Key Insights and Handling Special Cases

Key Features:

- Tracks how much each member owes or is owed
- Supports multiple transactions and dynamic participants per expense
- Automatically computes final settlement with the **least number of transactions**

Special Cases:

Case	Handling
Refunds	Can be represented as a negative expense by swapping payer and payees.
Missed Payments	Not automatically detected; relies on accurate input by users.
Single-person expense	Treated as fully borne by the payer; no debt generated.
Zero or negative amounts	Not handled explicitly — assumed user inputs valid data.

Results and Insights:

```
Enter the names of Members involved in the Expenses, Separated by commas:  A,B,C
---------------------------------------------------------------------------------------------------------
Enter the name of the payer (if Expenses are finished, the enter 'done') A
Enter the amount paid:  150
Enter the names of the people involved in this transaction, Separated by commas:  A,B,C
Enter the name of the payer (if Expenses are finished, the enter 'done') B
Enter the amount paid:  300
Enter the names of the people involved in this transaction, Separated by commas:  A,B
Enter the name of the payer (if Expenses are finished, the enter 'done') c
c is not a valid member.
Enter the name of the payer (if Expenses are finished, the enter 'done') C
Enter the amount paid:  240
Enter the names of the people involved in this transaction, Separated by commas:  A,b,C
b is not a valid member.
Enter the name of the payer (if Expenses are finished, the enter 'done') C
Enter the amount paid:  240
Enter the names of the people involved in this transaction, Separated by commas:  A,B,C
Enter the name of the payer (if Expenses are finished, the enter 'done') done
---------------------------------------------------------------------------------------------------------

 Individual Balance
A owes Rs.-130.00
B needs to be reimbursed with Rs.20.00
C needs to be reimbursed with Rs.110.00
---------------------------------------------------------------------------------------------------------

 Final Settlement
A owes C: Rs.110.00
A owes B: Rs.20.00
```

Possible Improvements:

- Add support for **weighted splitting** (e.g., based on shares or consumption)
- Include a **GUI or web interface**
- Add **persistent storage** (saving data to a file or database)
- Handle **data validation** (e.g., non-numeric input, missing names)

Challenges Faced:

- Avoiding circular debts (which was solved by the greedy `min(debt, credit)` settlement approach)
- Ensuring accurate float calculations (e.g., rounding errors in money handling)
- Keeping the logic minimal yet correct for all input cases

This project provides a **minimal, effective** solution to track and settle group expenses fairly. It focuses on clarity, usability, and simplicity, and serves as a solid foundation for more advanced financial management tools.