

Personal Finance Tracker – Project Documentation

Introduction:

The Personal Finance Tracker is a Python-based console application designed to help users manage their personal finances efficiently. The program enables users to log their daily transactions, categorize them as either Income or Expense, and view detailed summaries within a specified date range. Additionally, it provides a graphical representation of financial trends using Matplotlib.

The application stores all records in a CSV file (finance_data.csv), which acts as the database. This ensures portability and easy import into spreadsheet tools like Excel or Google Sheets.

Objectives:

Track daily income and expenses in a structured format.

Provide summaries of financial activity in a chosen date range.

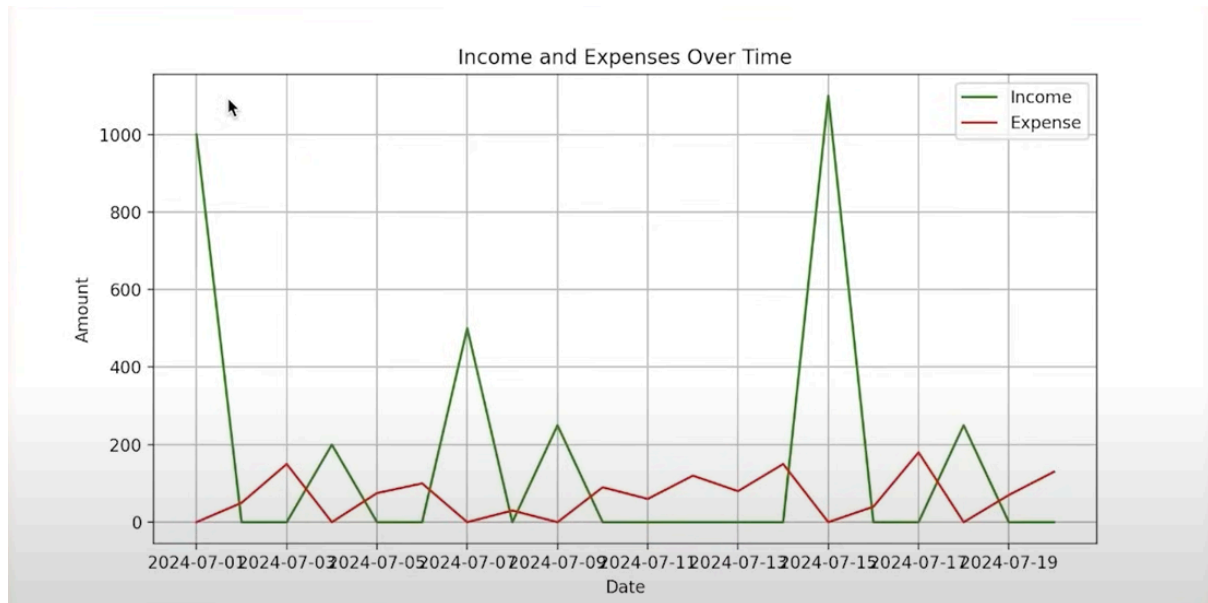
Visualize income and expenses using plots.

Using Python concepts such as:

- File handling (CSV read/write)
- Error handling and validation
- Object-Oriented Programming (OOP)
- Data analysis with Pandas
- Data visualization with Matplotlib

Functional Requirements:

1. Add New Transaction
 - User inputs: Date, Amount, Category (Income/Expense), Description.
 - The transaction is validated and saved into the CSV file.
2. View Transactions & Summary
 - User provides a start and end date.
 - Application filters transactions and displays them.
 - Outputs:
 - Transaction table
 - Total Income
 - Total Expense
 - Net Savings
3. Plot Transactions
 - After viewing transactions, the user can choose to plot them.
 - The line chart displays Income (green) and Expenses (red) over time.



4. Exit Program

- Ends the application gracefully.

Module Description:

data_entry.py

Handles user input and validation.

- get_date() -> Validates date format (dd-mm-yyyy).
- get_amount() -> Ensures entered amount is numeric and positive.
- get_category() -> Restricts category to Income/Expense.
- get_description() -> Takes an optional description.

main.py

Class: CSV

- initialize_csv() -> Creates the CSV file with required headers if not already present.
- add_entry() -> Appends a new transaction (dictionary) into the CSV file.
- get_transactions() -> Retrieves transactions within a date range, calculates totals, and prints a summary.

Functions

- add() -> Coordinates input from data_entry.py and inserts a new record.
- plot_transactions(df) -> Plots income vs expense using Matplotlib.
- main() -> Main menu loop providing options:
 1. Add transaction
 2. View transactions + summary (with optional plot)

3. Exit

Flow chart:

