

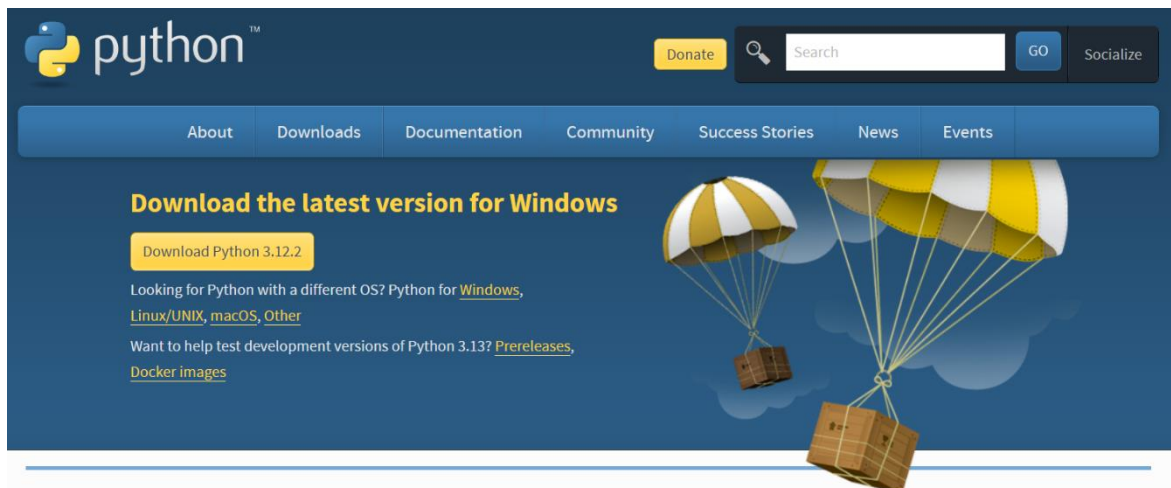
Enhanced Personal Finance Tracker (GUI Implementation with Tkinter and OOP)

Overview

The Enhanced Personal Finance Tracker is a user-friendly Python-based program equipped with graphical user interface (GUI) developed using the Tkinter library, designed to facilitate efficient management to help users manage their expenses.

This application enables users to seamlessly add, view, update, and delete transactions, and display a summary of all transactions and as well as by implementing essential features such as viewing, searching, and sorting transactions loaded from the JSON file (Expense.json) allowing to visualize in a structured format within the GUI. With ability that empowers users to quickly find relevant information by searching transactions based on specific criteria. Additionally, its sorting functionality allows users to organize transactions by date, type, or amount, enabling a clearer understanding of their financial history. The program supports bulk transaction load from a file, enhancing its functionality by enabling users to quickly import multiple transactions at once.

Overall, this application provides a practical tool for basic personal finance tracking with use of Tkinter's GUI capabilities.



Setup Instructions

Follow these steps to setup Personal Finance Tracker:

1. Install Python
 - Make sure you have installed the latest version of python on your system. If not, download and install python using this link: [Download Python | Python.org](https://www.python.org/downloads/)

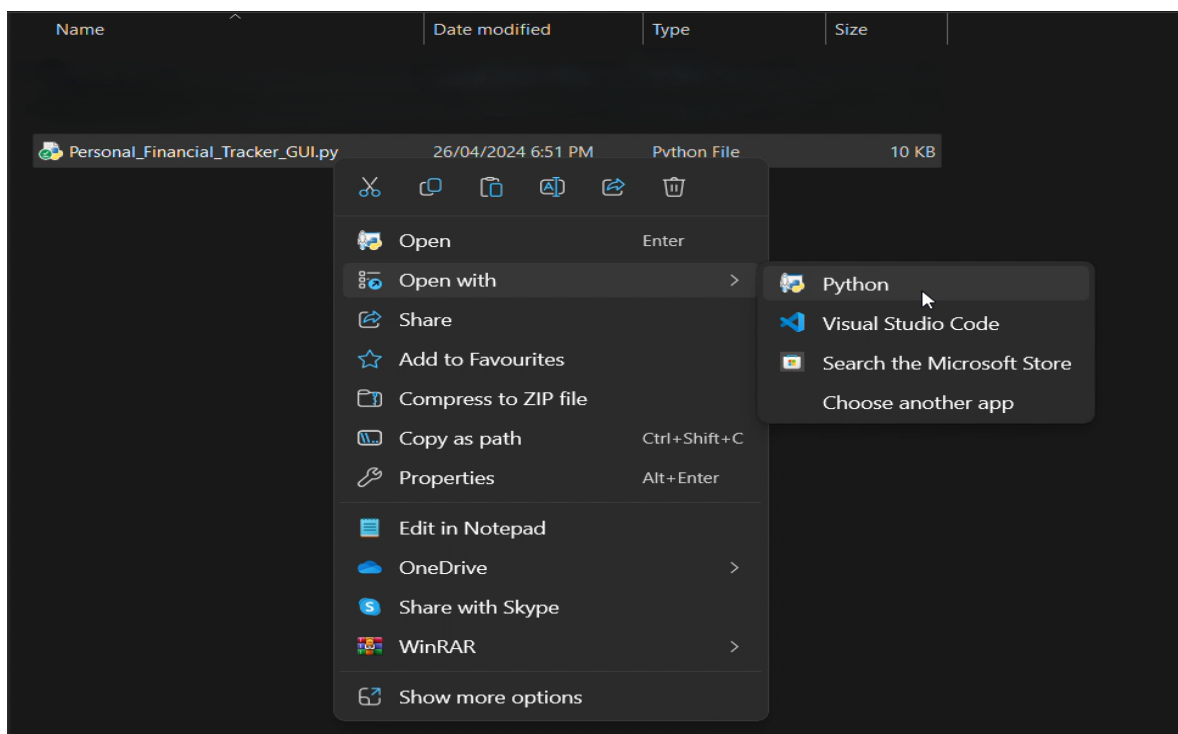
- Tkinter Dependency: Tkinter is included with Python by default, so no separate installation is required. However, if you're using Python 2.x, you may need to install Tkinter separately. You can do this using the following Command on your computer command prompt - pip install tk

2. Get Python Script File.

- Download the python script file ('Personal_Financial_Tracker_GUI.py') from the source and save it to the previous script saves location.

3. Run the program:

- Run the python script file ('Personal_Financial_Tracker_GUI.py') with python



4. Follow the instructions.

- Once the program starts, follow to instructions to use it's features.

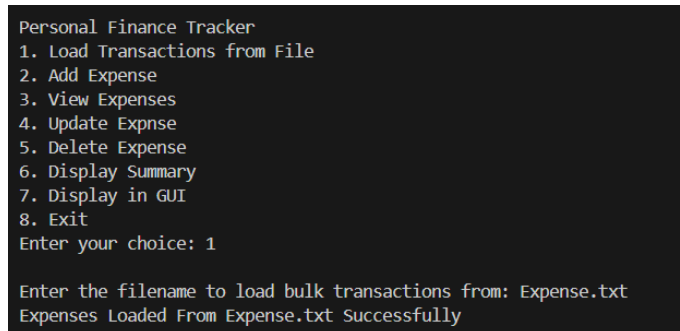
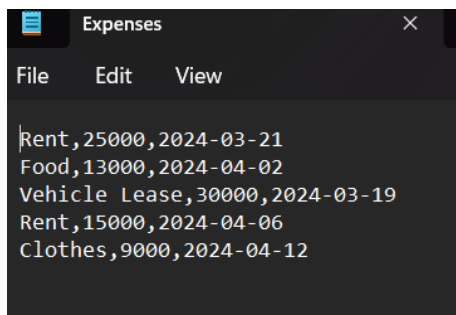
```

Personal Finance Tracker
1. Load Transactions from File
2. Add Expense
3. View Expenses
4. Update Expnse
5. Delete Expense
6. Display Summary
7. Display in GUI
8. Exit
Enter your choice: |
  
```

Features & Usage

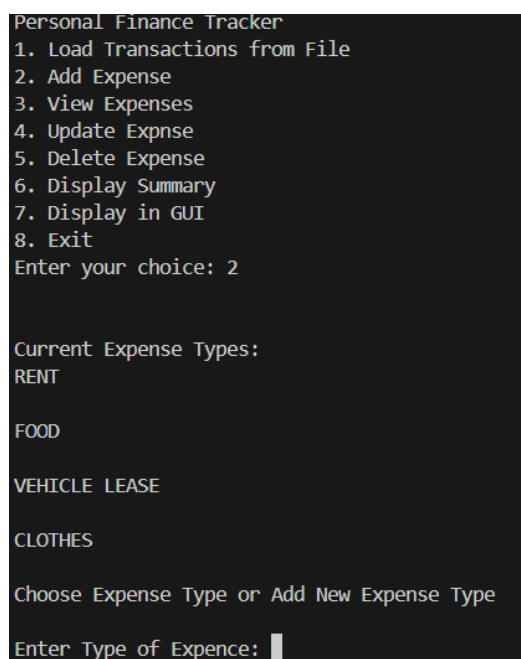
1. Load Transactions From File

- Allows the user to load bulk transactions from a file and the file should be in a specific format with each line containing comma-separated value for expense type, amount, and date



2. Add Transaction

- Shows the existing expense types and Allows the user to add new expense type to the tracker or add to an existing type. Users are prompted to enter expense type, Amount, Date.



3. View Transactions

- Users can view all existing expenses grouped by expense type. Each expense is displayed with its corresponding amount and date.

```
Personal Finance Tracker
1. Load Transactions from File
2. Add Expense
3. View Expenses
4. Update Expnse
5. Delete Expense
6. Display Summary
7. Display in GUI
8. Exit
Enter your choice: 3

RENT:
Expense 1) Amount: 25000.0, Date: 2024-03-21

Expense 2) Amount: 15000.0, Date: 2024-04-06

FOOD:
Expense 1) Amount: 13000.0, Date: 2024-04-02

VEHICLE LEASE:
Expense 1) Amount: 30000.0, Date: 2024-03-19

CLOTHES:
Expense 1) Amount: 9000.0, Date: 2024-04-12
```

4. Update Transaction

- Allows the user to update expenses and can choose the expense type and then specify which expense to update by its number. Options are to update amount and date.

```
Enter your choice: 4

Current Expense Types

RENT:
Expense 1) Amount: 25000.0, Date: 2024-03-21

Expense 2) Amount: 15000.0, Date: 2024-04-06

FOOD:
Expense 1) Amount: 13000.0, Date: 2024-04-02

VEHICLE LEASE:
Expense 1) Amount: 30000.0, Date: 2024-03-19

CLOTHES:
Expense 1) Amount: 9000.0, Date: 2024-04-12

Enter the type of expense to update: rent
Enter the Expense number to be updated:1
Current Expense details - Amount: 25000.0, Date: 2024-03-21

What would you like to update? (Amount/Date):amount

Enter new amount:26000
New Amount Is Successfully Updated.

Would you like to update another choice? (Yes/No):no

Would you like to update another expense type? (Yes/No):no

Expenses Updated Successfully!
```

5. Delete Transaction

- Users can delete existing expenses and select the expense type and then specify which expense to delete by its number. If Expense type is empty, user can delete the entire expense type

Enter your choice: 5

Current Expense Types

RENT:

Expense 1) Amount: 4500.0, Date: 2024-03-21

Expense 2) Amount: 15000.0, Date: 2024-04-06

Expense 3) Amount: 5000.0, Date: 2012-12-14

FOOD:

Expense 1) Amount: 13000.0, Date: 2024-04-02

VEHICLE LEASE:

Expense 1) Amount: 30000.0, Date: 2024-03-19

CLOTHES:

Expense 1) Amount: 9000.0, Date: 2024-04-12

Enter the type of expense to delete: food

Enter the Expense number of the to delete: 1

Transaction deleted successfully.

NO Expenses left in FOOD

Would you like to delete the entire expense type? (Yes/No):yes

Entire expense type and Expenses have been deleted successfully.

Would you like to delete another expense type? (Yes/No):no

Expenses Deleted Successfully!

6. Display Summary

- Display a summary of the user's Expense data. The summary includes total expenses for each expense type and overall total across all types.

```
Enter your choice: 6

Summary of Expenses:

RENT: Total Expenses = 41000.0
FOOD: Total Expenses = 13000.0
VEHICLE LEASE: Total Expenses = 30000.0
CLOTHES: Total Expenses = 9000.0

Total expenses in all types: 93000.0
```

7. Display in GUI

- Users can easily view their financial transactions which was updated on the json file within the application's graphical user interface, providing a comprehensive overview of their financial activities. The GUI is equipped with sorting and searching functions.

```
Personal Finance Tracker
1. Load Transactions from File
2. Add Expense
3. View Expenses
4. Update Expense
5. Delete Expense
6. Display Summary
7. Display in GUI
8. Exit
Enter your choice: 7

GUI Loaded
-----
GUI Closed
```

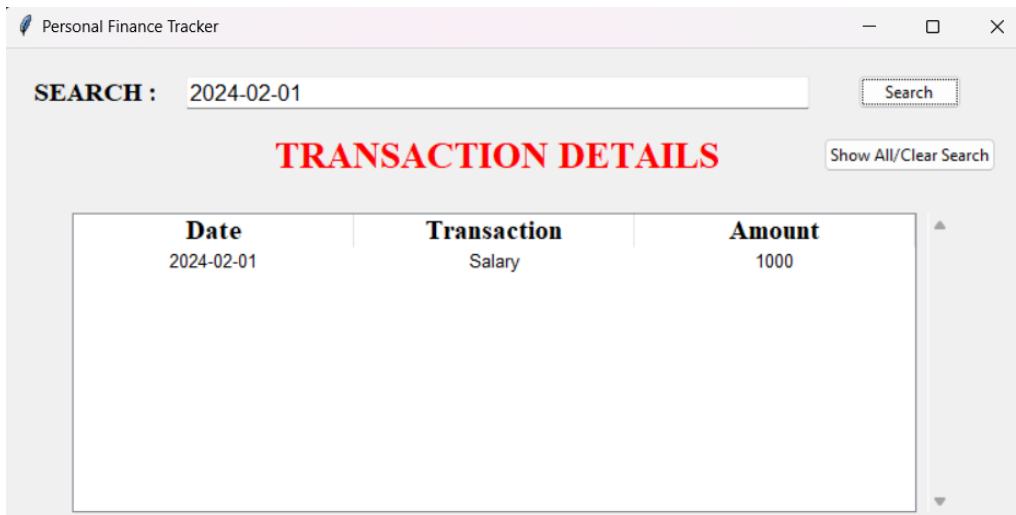


The screenshot shows a window titled "Personal Finance Tracker". At the top, there is a "SEARCH:" label followed by a text input field and a "Search" button. Below this is a red heading "TRANSACTION DETAILS" and a "Show All/Clear Search" button. The main content area contains a table with three columns: "Date", "Transaction", and "Amount". The table lists three transactions: "2024-02-03" for "Groceries" with an amount of "150", "2024-02-15" for "Groceries" with an amount of "75", and "2024-02-01" for "Salary" with an amount of "1000".

Date	Transaction	Amount
2024-02-03	Groceries	150
2024-02-15	Groceries	75
2024-02-01	Salary	1000

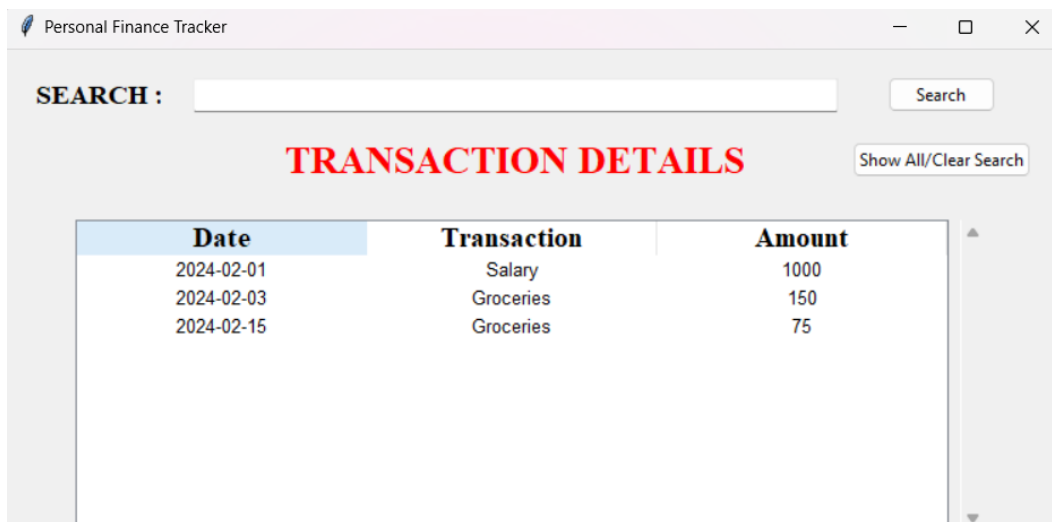
8. Searching Transaction

- The GUI allows users to search for specific transactions based on customized criteria such as date, transaction type, or amount, allowing for precise filtering of transactions based on their unique preferences and requirements, enhancing their ability to locate relevant information quickly and effectively.



9. Sorting Transactions

- Users have the option to sort transactions between ascending and descending order by clicking on the date, transaction type, or amount heading enabling them to organize their financial data in a structured manner for better analysis and understanding.



10. Exit

- Users can exit the program by selecting this option. A confirmation prompt ensures that the user intends to exit.

```
Personal Finance Tracker
1. Load Transactions from File
2. Add Expense
3. View Expenses
4. Update Expense
5. Delete Expense
6. Display Summary
7. Display in GUI
8. Exit
Enter your choice: 8

Are you sure you want to exit the program?(Yes/No): yes
Exiting program.
```

Code Overview

GUI Overview

- Imports and Initialization: The code starts by importing necessary modules (tkinter, ttk, messagebox, and json). It also initializes a dictionary transactions to store transaction data.
- FinanceTrackerGUI Class: This class is responsible for creating and managing the GUI. It includes methods for creating widgets, loading transactions from a JSON file, displaying transactions, searching transactions, and sorting transactions by different criteria.
- Widget Creation: The create_widgets method sets up the GUI layout, including a search bar, buttons for searching and displaying all transactions, and a Treeview widget for displaying transaction details.
- Loading and Displaying Transactions: The load_transactions method reads transaction data from a JSON file, and display_transactions method displays the loaded transactions in the Treeview widget.

- **Searching Transactions:** The `search_transactions` method allows users to search for transactions by date, transaction type, or amount.
- **Sorting Transactions:** The `sort_by_column` method and its related methods enable sorting transactions by date, transaction type, or amount.
- **Running the GUI:** The `gui_run` function initializes the Tkinter application and starts the main event loop.

CLI Overview

- **Transaction Management Functions:** The CLI part of the application includes functions for loading transactions from a JSON file (`load_transactions`), saving transactions to a JSON file (`save_transactions`), adding new transactions (`add_transaction`), viewing all transactions (`view_transactions`), updating existing transactions (`update_transaction`), deleting transactions (`delete_transaction`), and displaying a summary of all transactions (`display_summary`).
- **Main Menu:** The `main_menu` function provides a text-based interface for users to interact with the application. It allows users to load transactions, add new transactions, view transactions, update transactions, delete transactions, display a summary of transactions, and switch to the GUI mode.
- **Execution:** The application starts by calling the `main_menu` function, which presents the user with a menu to choose between CLI and GUI modes or to exit the application.

NOTE

- Ensure Python (version 3.x recommended) and the Tkinter library are installed on your system to run the application smoothly.
- Make sure that in the text file (.txt) expense type, amount, date is entered in the correct order to retain from any errors.
- The JSON file will be created if it does not exist with indication.
- The program utilizes a JSON file (Expense. Json) to store and load transactions.
- Expense types are case-insensitive; they are automatically converted to uppercase for consistency.
- Dates are expected in the format YYYY-MM-DD.
- The program provides error handling for invalid inputs and file operations.
- The application provides flexibility in searching transactions based on specific criteria such as date, transaction type, or amount.