Final Project: Personal Finance Tracker App

Description

This project involves creating an interactive, text-based **Personal Finance Tracker App** that helps users manage and analyze their spending habits. Users can import a CSV file containing transaction data, perform operations like viewing, adding, editing, and deleting transactions, and analyze spending patterns. The app also includes data visualization capabilities, displaying monthly spending trends and top spending categories. The project emphasizes Python programming, pandas for data handling, and matplotlib for visualization, along with GitHub for collaboration and version control.

Objective

- Build a Python-based interactive app simulating personal finance management.
- Implement file handling for importing, editing, and saving transaction data.
- Use pandas for data manipulation and analysis.
- Add data visualization using matplotlib for a better understanding of spending trends.
- Use GitHub for collaborative development and version control.

Features

1. **File Import Functionality**
    - Users can read a CSV file containing transaction data (fields: date, category, description, amount) `sampledata.csv`. (If your bank provides a CSV file of your transactions, you can use it directly with this app.)
    - Handles errors like incorrect file format or missing columns.
2. **Data Management**

   - **View Transactions by Date Range**: Filter and display transactions within a specified date range.
   - **Add a Transaction**: Add a new transaction with details like date, category, description, and amount.
   - **Edit a Transaction**: Modify details of an existing transaction (date, category, description, amount).
   - **Delete a Transaction**: Remove a specific transaction by its index.
3. **Spending Analysis**
   - **Analyze Spending by Category**: Display total spending for each category.
   - **Calculate Average Monthly Spending**: Show average spending per month.
   - **Show Top Spending Category**: Identify the category with the highest total spending.
4. **Data Visualization**
   - **Monthly Spending Trend**: Visualize spending trends over time using a line chart.
   - **Spending by Category**: Create a bar chart showing total spending by category.
   - **Percentage Distribution**: Generate a pie chart representing the distribution of spending across categories.
5. **Save Transactions to CSV**
   - Save the updated list of transactions to a CSV file, maintaining a record of financial data.
6. **User Interaction**
   - A fully text-based interface with prompts for input at each stage, allowing users to manage transactions and view analysis results.
   - A menu-driven interface that simplifies navigation through the app's features.
7. **GitHub Collaboration**

- Teams use GitHub to manage the project collaboratively:
    - Create branches for different modules (data management, analysis, visualization).
    - Use pull requests to merge completed work into the main branch after review.
    - Document the project setup, instructions, and usage in a README file.

Workflow and Requirements

- **Technologies**: Python, pandas, matplotlib, GitHub (for collaboration).
- **Dataset**: Students can use a sample CSV file with columns "Date," "Category," "Description," and "Amount." or if your bank provides a CSV file of your transactions, you can use it directly with this app.
- **Modules**:
    - **Data Management Module**: Handles file import, viewing, adding, editing, and deleting transactions.
    - **Data Analysis Module**: Includes spending analysis by category, average monthly spending, and identifying the top spending category.
    - **Visualization Module**: Creates line, bar, and pie charts to visualize spending trends and distribution.
- **Deliverables**:
    - Complete code on GitHub with documented commits and a structured README file.
    - A presentation demonstrating the app’s functionality, insights, and visualizations.
    - A description of collaborative development, challenges, and solutions.

Evaluation Criteria

- **Functionality**: How well does the app meet the requirements, including file handling, data management, analysis, and visualizations?

- **User Interaction**: Is the app easy to use and navigate with clear prompts and outputs?
- **Code Quality**: Is the code well-structured, modular, and readable? Are pandas and matplotlib used effectively?
- **Collaboration**: Did the team use GitHub effectively, with clear commits, pull requests, and issue management?
- **Presentation**: How well do the students present their work, including app functionality, challenges faced, and key insights?

Sample Output Behavior for the Personal Finance Tracker App

This section demonstrates the expected output behavior of the app. **This is a sample guide—students can be creative and improve their own versions.**

Initial Menu Display

When users start the app, the main menu is displayed:

```
=== Personal Finance Tracker ===
0. Import a CSV File
1. View All Transactions
2. View Transactions by Date Range
3. Add a Transaction
4. Edit a Transaction
5. Delete a Transaction
6. Analyze Spending by Category
7. Calculate Average Monthly Spending
8. Show Top Spending Category
9. Visualize Monthly Spending Trend
10. Save Transactions to CSV
11. Exit
Choose an option (1-11):
```

1. View All Transactions

Displays the current transaction list:

```
--- All Transactions ---
        Date   Category       Description   Amount
0 2024-10-01       Food          Grocery     50.75
1 2024-10-02       Rent      Monthly Rent  1200.00
2 2024-10-02  Utilities  Electricity Bill    60.00
3 2024-10-03       Food           Dinner     30.00
4 2024-10-04  Transport       Bus Ticket      2.75
```

2. View Transactions by Date Range

Prompts for a start and end date:

```
Enter the start date (YYYY-MM-DD): 2024-10-02
Enter the end date (YYYY-MM-DD): 2024-10-03

--- Transactions from 2024-10-02 to 2024-10-03 ---
        Date   Category       Description   Amount
1 2024-10-02       Rent      Monthly Rent  1200.00
2 2024-10-02  Utilities  Electricity Bill    60.00
3 2024-10-03       Food           Dinner     30.00
```

If no transactions are found:

```
No transactions found in this date range.
```

3. Add a Transaction

Prompts for new transaction details:

```
Enter the date (YYYY-MM-DD): 2024-10-05
Enter the category (e.g., Food, Rent): Food
Enter a description: Breakfast
Enter the amount: 15.00
```

```
Transaction added successfully!
```

4. Edit a Transaction

Prompts for the index of the transaction to edit:

```
Enter the index of the transaction to edit: 0

Current Transaction Details:
Date          2024-10-01 00:00:00
Category                  Food
Description            Grocery
Amount                    50.75

Enter new date (YYYY-MM-DD) or press Enter to keep current:
Enter new category or press Enter to keep current: Food
Enter new description or press Enter to keep current: Groceries
Enter new amount or press Enter to keep current: 55.00

Transaction updated successfully!
```

If the index is invalid:

```
Invalid index.
```

5. Delete a Transaction

Prompts for the index of the transaction to delete:

```
Enter the index of the transaction to delete: 2

Transaction deleted successfully!
```

If the index is invalid:

```
Invalid index.
```

6. Analyze Spending by Category

Displays total spending for each category:

```
--- Total Spending by Category ---
Category
Food          95.75
Rent        1200.00
Transport      2.75
```

7. Calculate Average Monthly Spending

Calculates and displays the average monthly spending:

```
--- Average Monthly Spending ---
337.88
```

8. Show Top Spending Category

Displays the top spending category:

```
--- Top Spending Category ---
Rent with 1200.00 total spending.
```

9. Visualize Monthly Spending Trend

Displays a line chart showing total spending over months:

```
[Line Chart Opens]
```

The line chart visualizes monthly spending, helping users track trends.

10. Save Transactions to CSV

Prompts to enter a file name for saving:

```
Enter file name to save (e.g., 'transactions.csv'): my_transactions.csv

Transactions saved to my_transactions.csv successfully!
```

11. Exit

Exits the program:

```
Exiting the Personal Finance Tracker. Goodbye!
```