

Personal Expense Tracker

Course-End Project Problem Statement
(Python Fundamentals)

Submitted By

Nazrul Islam Choudhury

Description:

In today's fast-paced world, individuals need to track and manage their expenses effectively. Your task is to build a personal expense tracker that allows users to log daily expenses, categorize them, and track spending against a monthly budget. The tracker should also be able to save and load expenses from a file for future reference.

Objectives:

1. Design and implement a personal expense tracker that enables users to manage their expenses
2. Allow users to categorize expenses and set monthly budgets
3. Implement file-handling functionality to save and load expense data
4. Create an interactive, menu-driven interface for ease of use

Features and Functionality:

Core Features:

1. Expense Logging

- Users can input daily expenses with details such as date, amount, category, and description.
- Data is stored in a structured format for easy retrieval and analysis.

2. Expense Categorization

- Expenses are grouped into categories (e.g., Food, Transport, Entertainment).
- Users can create custom categories to tailor the tracker to their needs.

3. Budget Management

- Users can set monthly budgets for specific categories as well as an overall monthly budget.
- The tracker calculates and displays the remaining budget for each category.

4. Expense Visualization

- Provides insights into spending patterns by category and overall budget utilization.

5. File Handling

- Save expenses and budgets to a file for future reference.
- Load previously saved data to continue tracking seamlessly.

6. Interactive Interface

- A menu-driven Command-Line Interface (CLI) for ease of use.
- Options to add expenses, view logs, set budgets, save/load data, and more.

Implementation Plan

1. Data Structure Design

- **Expense Entry:**
 - Date (YYYY-MM-DD)
 - Amount
 - Category
 - Description (optional)
- **Budget Management:**
 - Individual category budgets.
 - Overall monthly budget.

2. Menu Options

1. Add Expense

- Input date, amount, category, and description.
- Append the data to the expense list.

2. View Expenses

- Display all expenses with filters for date and category.

3. Set Monthly Budget

- Allow users to define budgets for categories and overall spending.

4. View Budget Status

- Show remaining budget per category and overall.
- Highlight overspending in any category.

5. Save to File

- Export data to a JSON or CSV file.

6. Load from File

- Import previously saved data for continued tracking.

7. Exit

- Terminate the program.

Technology Stack

- **Programming Language:** Python
- **Libraries:**
 - json for file handling.
 - datetime for date validation and manipulation.
 - Optionally, matplotlib for visualizing spending trends.

Sample Code Outline

1. Data Structures

```
expenses = []  
budget = {}
```

2. Add Expense Function

```
def add_expense():  
    date = input("Enter date (YYYY-MM-DD): ")  
    amount = float(input("Enter amount: "))  
    category = input("Enter category: ")  
    description = input("Enter description (optional): ")  
    expenses.append({"date": date, "amount": amount, "category": category, "description": description})  
    print("Expense added successfully!")
```

3. View Expenses Function

```
def view_expenses():  
    print("\n--- Expenses ---")  
    for expense in expenses:  
        print(f"{expense['date']}: {expense['amount']} ({expense['category']}) - {expense['description']}")
```

4. Set & View Budget Function

```
def set_budget():  
    global budget  
    budget = {}  
    print("Set monthly budget for categories. Type 'done' to finish.")  
    while True:  
        category = input("Enter category: ")  
        if category.lower() == 'done':  
            break  
        amount = float(input(f"Enter budget for {category}: "))  
        budget[category] = amount  
    total_budget = float(input("Enter total monthly budget: "))  
    budget["Total"] = total_budget  
    print("Budget set successfully!")  
  
def view_budget_status():  
    print("\n--- Budget Status ---")  
    category_totals = {cat: 0 for cat in budget if cat != "Total"}  
    total_spent = 0  
    for expense in expenses:  
        if expense['category'] in category_totals:  
            category_totals[expense['category']] += expense['amount']  
        total_spent += expense['amount']  
    for category, spent in category_totals.items():  
        print(f"{category}: Spent {spent}, Remaining {budget[category] - spent}")  
    print(f"Total: Spent {total_spent}, Remaining {budget['Total'] - total_spent}")
```

5. File Handling Functions

```
def save_to_file():
    with open("D:\\IITM\\IITM AG\\Python Fundamentals\\Python_Project\\expenses.json", "w") as file:
        json.dump({"expenses": expenses, "budget": budget}, file)
    print("Data saved successfully!")

def load_from_file():
    global expenses, budget
    try:
        with open("D:\\IITM\\IITM AG\\Python Fundamentals\\Python_Project\\expenses.json", "r") as file:
            data = json.load(file)
            expenses = data["expenses"]
            budget = data["budget"]
        print("Data loaded successfully!")
    except FileNotFoundError:
        print("No saved data found.")
```

6. Main Function

```
def main():
    while True:
        print("\n--- Personal Expense Tracker ---")
        print("1. Add Expense")
        print("2. View Expenses")
        print("3. Set Monthly Budget")
        print("4. View Budget Status")
        print("5. Save to File")
        print("6. Load from File")
        print("7. Exit")
        choice = input("Choose an option: ")
        if choice == "1":
            add_expense()
        elif choice == "2":
            view_expenses()
        elif choice == "3":
            set_budget()
        elif choice == "4":
            view_budget_status()
        elif choice == "5":
            save_to_file()
        elif choice == "6":
            load_from_file()
        elif choice == "7":
            print("Goodbye!")
            break
        else:
            print("Invalid option, please try again.")
```

```
if __name__ == "__main__":
    main()
```

Enhancements and Future Scope:

1. **Data Validation:** Ensure proper formatting for dates and amounts.
2. **Visualization:** Integrate graphical representations of expenses using libraries like matplotlib.
3. **Reports:** Generate monthly spending reports and export them as PDFs.
4. **Cloud Integration:** Enable syncing data across devices.
5. **Mobile App Integration:** Expand to a mobile platform for convenience.

Conclusion:

The Personal Expense Tracker is a versatile and practical tool for managing personal finances. By implementing features such as expense logging, categorization, budget tracking, and file handling, it provides users with an effective solution to monitor and control their spending. With future enhancements, this tracker can be expanded into a comprehensive financial management application.