ssignment-personal-expense-tracker

June 7, 2025

0.0.1 Problem statement:

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.

0.0.2 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

0.0.3 Import the required Labraries

[1]:
```
import csv
```

0.0.4 Decalre the golable varables

[2]:
```
expense_stor=[]
```

0.0.5 1. Add an expense:

- add the deatil expense like date,category,amount,description

- Create a function to prompt the user for expense details. Ensure you ask for: o The date of the expense in the format YYYY-MM-DD o The category of the expense, such as Food or Travel o The amount spent o A brief description of the expense • Store the expense in a list as a dictionary, where each dictionary includes the date, category, amount, and description as key-value pairs Example: {'date': '2024-09-18', 'category': 'Food', 'amount': 15.50, 'description': 'Lunch with friends'}

[3]:

```
def add_expense(expense_stor):
    """This function is adding the expense and store into the list as key value␣
 ↪pair"""
    user_input_date=input("Please enter the date in the format YYYY-MM-DD")
    user_input_category=input("please enter the category of the expense, such␣
 ↪as Food or Travel")
    user_amount=float(input("please enter the spent amount"))
    user_expense_desc=input("please enter the brief description of the expense")
    expense_stor.append({"date" :user_input_date,"category":␣
 ↪user_input_category,"amount":user_amount,"description":user_expense_desc})
    print("entered epenses have been added successfully")
```

[4]:

```
#expense_stor
```

0.0.6 2. View expenses:

- View the added expenses
- Validate the added expenses

[5]:

```
def view_expenses(expenseDetails):
    """ This function is validating the required details and displaing the␣
 ↪expense """
    if not expenseDetails:
        print("No any expenses added in the list")
    else:
        for expense in expenseDetails:
            if all(key in expense for key in␣
 ↪['date','category','amount','description']):
                print(f"expense details : \n    date: {expense['date']} \n   ␣
 ↪category: {expense['category']} \n    amount: {expense['amount']} \n   ␣
 ↪description: {expense['description']} ")
            else:
                print(f"invalid expense please enter again {expense} ")
```

[6]:

```
#view_expenses(expense_stor)
```

0.0.7 3. Set and track the budget:

- Create a function that allows the user to input a monthly budget. Prompt the user to:Enter the total amount they want to budget for the month.

- Create another function that calculates the total expenses recorded so far
- Compare the total with the user's monthly budget
- If the total expenses exceed the budget, display a warning (Example: You have exceeded your budget!)
- If the expenses are within the budget, display the remaining balance (Example: You have 150 left for the month)

[7]:
```
#This function is allows the user to input a monthly budget.
def set_budget():
    """ This function is capturing the monthly total budget """
    return float(input('please enter your monthly budget : '))
```

[8]:
```
#set_budget()
```

[9]:
```
def track_budget(expenseDetails,monthlyBudget):
    """this function is help to track the budget and give the worning␣
 ↪accordingly"""
    #Calculate the total expenses
    total_expenses=sum(expense["amount"] for expense in expenseDetails)
    print(f"Total Expenses : {total_expenses}")
    # compare the total expenses with assigned budget and provide the woring␣
 ↪accordingly.
    if(total_expenses > monthlyBudget ):
        print(f"WARNING : You have exceeded your budget. \n Your Total Monthly␣
 ↪budget is : {monthlyBudget} \n your current total expenses is :␣
 ↪{total_expenses}")
    else:
        print(f" You are within the budget. \n Your Total reaming budget is :␣
 ↪{monthlyBudget - total_expenses} ")
```

[10]:
```
#track_budget(expense_stor,set_budget())
```

0.0.8 4. Save and load expenses:

####Save - Function to save all expenses to a CSV file, with each row containing the date, category, amount, and description of each expense

####Load - Function to load expenses from the CSV file. When the program starts,it will Read the saved data from the file - it back into the list of expenses so the user can see their previous expenses and continue from where they left off

[11]:
```
def save_expenses(expenses,fileName='expense.csv'):
    """This function is saving the expense into a file"""
    with open(fileName,'w',newline='') as file:
        writer=csv.writer(file)
        writer.writerow(["Date","Category","Amount","Description"])
        list(map(lambda expense: writer.
↪writerow([expense["date"],expense["category"],expense["amount"],expense["description"]]),␣
↪expenses))
       # for expense in expenses:
           # writer.
↪writerow([expense["date"],expense["category"],expense["amount"],expense["description"]])
    print(f" Total number of { len(expenses)}  expenses saved successfully in␣
↪the file")
```

[12]:

```
#save_expenses(expense_stor,fileName="expense.csv")
```

[13]:

```
def load_expenses(fileName='expense.csv'):
    """This function is loading the expenses from the file"""
    expenses=[]
    try:
        with open(fileName,'r') as file:
            reader=csv.DictReader(file)
            for row in reader:
                if all(key in row for key in␣
↪["Date","Category","Amount","Description"]):
                    row["Amount"]=float(row["Amount"])
                    expense={'date':row['Date'],'category':
↪row['Category'],'amount':row['Amount'],'description':row['Description']}
                    expenses.append(expense)
                else:
                    print(f"skipping the Invalid expense record : \n {row}")
    except FileNotFoundError :
        print("No existing expenses records file found . please add the expenes␣
↪in the file")

    return expenses
```

[14]:

```
#load_expenses(fileName='expense.csv')
```

0.0.9 5. Create an interactive menu:

- This function to display a menu with the following options:
- 1. Add expense
- 2. View expenses
- 3. Track budget
- 4. Save expenses
- 5. Exit
- Allow the user to enter a number to choose an option

- It provide the following conditions to user input:
- If the user selects option 1, call the function to add an expense
- If the user selects option 2, call the function to view expenses
- If the user selects option 3, call the function to track the budget
- If the user selects option 4, call the function to save expenses to the file
- If the user selects option 5, save the expenses and exit the program

[15]:

```
def main():
    expenses=load_expenses()
    budget=set_budget()

    while True:
        print("\n Personal Expense Tracker")
        print("1. Add expense")
        print("2. View expenses")
        print("3. Track budget")
        print("4. Save expenses")
        print("5. Exit")
        choice=int(input("please enter your choice"))

        if(choice==1):
            add_expense(expense_stor)
        elif(choice==2):
            view_expenses(expense_stor)

        elif(choice==3):
            track_budget(expense_stor,budget)

        elif(choice==4):
            save_expenses(expense_stor,fileName='expense.csv')

        elif(choice==5):
            save_expenses(expense_stor,fileName='expense.csv')
            print("existing...")
            break
        else:
            print("Inavlid choice please select the correct choice")
```

Run Main function

[16]:

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

```
please enter your monthly budget :  5000
```

```
 Personal Expense Tracker
1. Add expense
2. View expenses
3. Track budget
4. Save expenses
5. Exit

please enter your choice 1
Please enter the date in the format YYYY-MM-DD 2024-06-07
please enter the category of the expense, such as Food or Travel Travel
please enter the spent amount 2000
please enter the brief description of the expense Travel for business meetings

entered epenses have been added successfully


 Personal Expense Tracker
1. Add expense
2. View expenses
3. Track budget
4. Save expenses
5. Exit

please enter your choice 2

expense details :
    date: 2024-06-07
    category: Travel
    amount: 2000.0
    description: Travel for business meetings


 Personal Expense Tracker
1. Add expense
2. View expenses
3. Track budget
4. Save expenses
5. Exit

please enter your choice 3

Total Expenses : 2000.0
 You are within the budget.
 Your Total reaming budget is : 3000.0


 Personal Expense Tracker
1. Add expense
2. View expenses
3. Track budget
4. Save expenses
5. Exit

please enter your choice 4
```

```
 Total number of 1  expenses saved successfully in the file

 Personal Expense Tracker
1. Add expense
2. View expenses
3. Track budget
4. Save expenses
5. Exit

please enter your choice 1
Please enter the date in the format YYYY-MM-DD 2024-06
please enter the category of the expense, such as Food or Travel food
please enter the spent amount 1000
please enter the brief description of the expense lunch with product owner

entered epenses have been added successfully

 Personal Expense Tracker
1. Add expense
2. View expenses
3. Track budget
4. Save expenses
5. Exit

please enter your choice 2

expense details :
    date: 2024-06-07
    category: Travel
    amount: 2000.0
    description: Travel for business meetings
expense details :
    date: 2024-06
    category: food
    amount: 1000.0
    description: lunch with product owner

 Personal Expense Tracker
1. Add expense
2. View expenses
3. Track budget
4. Save expenses
5. Exit

please enter your choice 3

Total Expenses : 3000.0
 You are within the budget.
 Your Total reaming budget is : 2000.0

 Personal Expense Tracker
```

```
1. Add expense
2. View expenses
3. Track budget
4. Save expenses
5. Exit

please enter your choice 4

 Total number of 2  expenses saved successfully in the file


 Personal Expense Tracker
1. Add expense
2. View expenses
3. Track budget
4. Save expenses
5. Exit

please enter your choice 1
Please enter the date in the format YYYY-MM-DD 2024-02-23
please enter the category of the expense, such as Food or Travel Travel
please enter the spent amount 1000
please enter the brief description of the expense client travel

entered epenses have been added successfully


 Personal Expense Tracker
1. Add expense
2. View expenses
3. Track budget
4. Save expenses
5. Exit

please enter your choice 2

expense details :
    date: 2024-06-07
    category: Travel
    amount: 2000.0
    description: Travel for business meetings
expense details :
    date: 2024-06
    category: food
    amount: 1000.0
    description: lunch with product owner
expense details :
    date: 2024-02-23
    category: Travel
    amount: 1000.0
    description: client travel


 Personal Expense Tracker
```

```
1. Add expense
2. View expenses
3. Track budget
4. Save expenses
5. Exit

please enter your choice 3

Total Expenses : 4000.0
 You are within the budget.
 Your Total reaming budget is : 1000.0

 Personal Expense Tracker
1. Add expense
2. View expenses
3. Track budget
4. Save expenses
5. Exit

please enter your choice 4

 Total number of 3  expenses saved successfully in the file

 Personal Expense Tracker
1. Add expense
2. View expenses
3. Track budget
4. Save expenses
5. Exit

please enter your choice 1
Please enter the date in the format YYYY-MM-DD 2024-03-06
please enter the category of the expense, such as Food or Travel food
please enter the spent amount 2000
please enter the brief description of the expense break fast with client

entered epenses have been added successfully

 Personal Expense Tracker
1. Add expense
2. View expenses
3. Track budget
4. Save expenses
5. Exit

please enter your choice 1
Please enter the date in the format YYYY-MM-DD 2025-03-08
please enter the category of the expense, such as Food or Travel travel
please enter the spent amount 500
please enter the brief description of the expense bus ticket for client
```

```
entered epenses have been added successfully

 Personal Expense Tracker
1. Add expense
2. View expenses
3. Track budget
4. Save expenses
5. Exit

please enter your choice 2

expense details :
    date: 2024-06-07
    category: Travel
    amount: 2000.0
    description: Travel for business meetings
expense details :
    date: 2024-06
    category: food
    amount: 1000.0
    description: lunch with product owner
expense details :
    date: 2024-02-23
    category: Travel
    amount: 1000.0
    description: client travel
expense details :
    date: 2024-03-06
    category: food
    amount: 2000.0
    description: break fast with client
expense details :
    date: 2025-03-08
    category: travel
    amount: 500.0
    description: bus ticket for client

 Personal Expense Tracker
1. Add expense
2. View expenses
3. Track budget
4. Save expenses
5. Exit

please enter your choice 3

Total Expenses : 6500.0
WARNING : You have exceeded your budget.
 Your Total Monthly budget is : 5000.0
 your current total expenses is : 6500.0
 Personal Expense Tracker
1. Add expense
2. View expenses
3. Track budget
4. Save expenses
5. Exit

please enter your choice 4

 Total number of 5  expenses saved successfully in the file

 Personal Expense Tracker
1. Add expense
2. View expenses
3. Track budget
4. Save expenses
5. Exit

please enter your choice 5

 Total number of 5  expenses saved successfully in the file
existing…
```