ExpenseTracker

May 12, 2025

[2]:

```python
#import all libraries

import csv
import os
from datetime import datetime

expenses = []
monthly_budget = 0.0
expense_file = 'expenses.csv'
```

[3]:

```python
#This cell defines the function `load_expenses`
def load_expenses():
    global expenses
    if os.path.exists(expense_file):
        with open(expense_file, mode='r', newline='') as file:
            reader = csv.DictReader(file)
            for row in reader:
                try:
                    expenses.append({
                        'date': row['date'],
                        'category': row['category'],
                        'amount': float(row['amount']),
                        'description': row['description']
                    })
                except (ValueError, KeyError):
                    print("Skipping corrupted entry in file.")
    else:
        print("No previous expense file found. Starting fresh.")
```

[4]:

```python
#This cell defines the function `save_expenses`
def save_expenses():
    with open(expense_file, mode='w', newline='') as file:
        fieldnames = ['date', 'category', 'amount', 'description']
        writer = csv.DictWriter(file, fieldnames=fieldnames)
        writer.writeheader()
        for expense in expenses:
            writer.writerow(expense)
    print("Expenses saved successfully.")
```

[5]:

```python
#This cell defines the function `add_expense`
def add_expense():
    try:
        date = input("Enter date (YYYY-MM-DD): ")
        datetime.strptime(date, '%Y-%m-%d')
        category = input("Enter category (e.g., Food, Travel): ").strip()
        amount = float(input("Enter amount: "))
        description = input("Enter description: ").strip()

        expenses.append({
            'date': date,
            'category': category,
            'amount': amount,
            'description': description
        })
        print("Expense added successfully.")
    except ValueError:
        print("Invalid input. Please try again.")
```

[6]:

```python
#This cell defines the function `view_expenses`
def view_expenses():
    if not expenses:
        print("No expenses to show.")
        return
    print("\nAll Expenses:")
    for exp in expenses:
        if all(k in exp for k in ['date', 'category', 'amount', 'description']):
            print(f"{exp['date']} | {exp['category']} | ${exp['amount']:.2f} |␣
 ↪{exp['description']}")
        else:
            print("Incomplete entry skipped.")
```

[7]:

```python
#This cell defines the function `track_budget`
def track_budget():
    global monthly_budget
    if monthly_budget == 0:
        try:
            monthly_budget = float(input("Enter your monthly budget: "))
        except ValueError:
            print("Invalid budget input.")
            return

    total_spent = sum(exp['amount'] for exp in expenses)
    print(f"\nTotal spent: ${total_spent:.2f}")
    if total_spent > monthly_budget:
        print("Warning: You have exceeded your budget!")
    else:
        remaining = monthly_budget - total_spent
        print(f"You have ${remaining:.2f} left for the month.")
```

[8]:

```
#This cell defines the function `display_menu`
def display_menu():
    print("\n--- Personal Expense Tracker Menu ---")
    print("1. Add expense")
    print("2. View expenses")
    print("3. Track budget")
    print("4. Save expenses")
    print("5. Exit")
```

[9]:

```
#This cell defines the function `main`
def main():
    load_expenses()
    while True:
        display_menu()
        choice = input("Choose an option (1-5): ").strip()
        if choice == '1':
            add_expense()
        elif choice == '2':
            view_expenses()
        elif choice == '3':
            track_budget()
        elif choice == '4':
            save_expenses()
        elif choice == '5':
            save_expenses()
            print("Exiting. Goodbye!")
            break
        else:
            print("Invalid choice. Please select from 1 to 5.")
```

[10]:

```
#This cell calls the `main()` function to run the program.
main()
```

```
No previous expense file found. Starting fresh.

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

Choose an option (1-5):  1
```

```
Enter date (YYYY-MM-DD):  2024-04-02
Enter category (e.g., Food, Travel):  Food
Enter amount:  5000
Enter description:  Grocery

Expense added successfully.


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

Choose an option (1-5):  1
Enter date (YYYY-MM-DD):  2025-04-05
Enter category (e.g., Food, Travel):  Loan
Enter amount:  30000
Enter description:  Loan Emi

Expense added successfully.


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

Choose an option (1-5):  1
Enter date (YYYY-MM-DD):  2025-04-10
Enter category (e.g., Food, Travel):  MF
Enter amount:  20000
Enter description:  MF

Expense added successfully.


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

Choose an option (1-5):  1
Enter date (YYYY-MM-DD):  2025-04-15
Enter category (e.g., Food, Travel):  Travel
Enter amount:  15000
Enter description:  Travelling

Expense added successfully.
```

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

Choose an option (1-5):  1
Enter date (YYYY-MM-DD):  2025-04-25
Enter category (e.g., Food, Travel):  Party
Enter amount:  3000
Enter description:  Office Party

Expense added successfully.

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

Choose an option (1-5):  2


All Expenses:
2024-04-02 | Food | $5000.00 | Grocery
2025-04-05 | Loan | $30000.00 | Loan Emi
2025-04-10 | MF | $20000.00 | MF
2025-04-15 | Travel | $15000.00 | Travelling
2025-04-25 | Party | $3000.00 | Office Party

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

Choose an option (1-5):  3
Enter your monthly budget:  130000


Total spent: $73000.00
You have $57000.00 left for the month.

--- Personal Expense Tracker Menu ---
1. Add expense
2. View expenses
3. Track budget
```

```
4. Save expenses
5. Exit

Choose an option (1-5):  4

Expenses saved successfully.

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

Choose an option (1-5):  1
Enter date (YYYY-MM-DD):  2025-04-27
Enter category (e.g., Food, Travel):  Medical
Enter amount:  10000
Enter description:  Vaccination

Expense added successfully.

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

Choose an option (1-5):  4

Expenses saved successfully.

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

Choose an option (1-5):  3


Total spent: $83000.00
You have $47000.00 left for the month.

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

```
Choose an option (1-5):  5

Expenses saved successfully.
Exiting. Goodbye!
```

```
[ ]:
```