

```

import csv
from datetime import datetime

# Global list to store expenses
expenses = []

def add_expense():
    """Prompt the user to input expense details and add them to the expenses list."""
    try:
        date = input("Enter the date (YYYY-MM-DD): ")
        datetime.strptime(date, "%Y-%m-%d") # Validate date format
        category = input("Enter the category (e.g., Food, Travel): ")
        amount = float(input("Enter the amount spent: "))
        description = input("Enter a brief description: ")

        expense = {
            'date': date,
            'category': category,
            'amount': amount,
            'description': description
        }
        expenses.append(expense)
        print("Expense added successfully!\n")
    except ValueError as e:
        print(f"Error: {e}. Please try again.\n")

def view_expenses():
    """Display all stored expenses."""
    if not expenses:
        print("No expenses to display.\n")
        return

    print("Expenses:\n")
    for expense in expenses:
        if all(key in expense for key in ('date', 'category', 'amount', 'description')):
            print(f'Date: {expense['date']}, Category: {expense['category']}, Amount: {expense['amount']}, Description: {expense['description']}\n')
        else:
            print("Incomplete expense entry detected.\n")

def set_and_track_budget():
    """Allow the user to set a budget and track expenses against it."""
    try:
        budget = float(input("Enter your monthly budget: "))

```

```

total_expenses = sum(expense['amount'] for expense in expenses)
remaining_budget = budget - total_expenses

if total_expenses > budget:
    print(f"You have exceeded your budget by {abs(remaining_budget):.2f}!\n")
else:
    print(f"You have {remaining_budget:.2f} left for the month.\n")
except ValueError:
    print("Invalid input. Please enter a numeric value.\n")

def save_expenses_to_file():
    """Save all expenses to a CSV file."""
    try:
        with open("expenses.csv", "w", newline="") as file:
            writer = csv.DictWriter(file, fieldnames=['date', 'category', 'amount', 'description'])
            writer.writeheader()
            writer.writerows(expenses)
        print("Expenses saved successfully!\n")
    except Exception as e:
        print(f"Error saving expenses: {e}\n")

def load_expenses_from_file():
    """Load expenses from a CSV file into the expenses list."""
    try:
        with open("expenses.csv", "r") as file:
            reader = csv.DictReader(file)
            for row in reader:
                row['amount'] = float(row['amount']) # Convert amount to float
                expenses.append(row)
        print("Expenses loaded successfully!\n")
    except FileNotFoundError:
        print("No previous expenses found. Starting fresh.\n")
    except Exception as e:
        print(f"Error loading expenses: {e}\n")

def display_menu():
    """Display the interactive menu and handle user choices."""
    while True:
        print("""
Personal Expense Tracker
1. Add Expense
2. View Expenses
3. Track Budget
4. Save Expenses

```

5. Exit

""")

try:

choice = int(input("Choose an option: "))

if choice == 1:

add\_expense()

elif choice == 2:

view\_expenses()

elif choice == 3:

set\_and\_track\_budget()

elif choice == 4:

save\_expenses\_to\_file()

elif choice == 5:

save\_expenses\_to\_file()

print("Goodbye!")

break

else:

print("Invalid choice. Please try again.\n")

except ValueError:

print("Invalid input. Please enter a number.\n")

# Load expenses from file at program start

load\_expenses\_from\_file()

# Start the interactive menu

display\_menu()