```
import csv
from datetime import datetime

expenses = []
monthly_budget = 0.0

# Function to add an expense
def add_expense()
    global expenses
    date = input(Enter the date of the expense (YYYY-MM-DD) )
    category = input(Enter the category of the expense (e.g., Food, Travel) )
    amount = float(input(Enter the amount spent ))
    description = input(Enter a brief description of the expense )

    expense = {
        ’date’ date,
        ’category’ category,
        ’amount’ amount,
        ’description’ description
    }

    expenses.append(expense)
    print(Expense added successfully!)


def view_expenses()
    global expenses
    if not expenses
        print(No expenses recorded yet.)
        return

    print(All recorded expenses)
    for expense in expenses
        if all(expense.values())  # Check if all values are present
            print(fDate {expense[’date’]}, Category {expense[’category’]}, Amount ${expense[’amount’]},
Description {expense[’description’]})
        else
            print(Incomplete expense entry found. Skipping...)


def set_budget()
    global monthly_budget
    monthly_budget = float(input(Enter your monthly budget ))
    print(fYour budget for the month is set to ${monthly_budget})

def track_budget()
    global expenses, monthly_budget
    total_expenses = sum(expense[’amount’] for expense in expenses)
    print(fTotal expenses so far ${total_expenses})

    if total_expenses  monthly_budget
        print(You have exceeded your budget!)
    else
        remaining_balance = monthly_budget - total_expenses
        print(fYou have ${remaining_balance} remaining for the month.)
```

```
# Function to save expenses to a CSV file
def save_expenses()
    global expenses
    with open(’expenses.csv’, ’w’, newline=’’) as csvfile
        fieldnames = [’date’, ’category’, ’amount’, ’description’]
        writer = csv.DictWriter(csvfile, fieldnames=fieldnames)

        writer.writeheader()
        for expense in expenses
            writer.writerow(expense)

    print(Expenses have been saved to ’expenses.csv’.)

# Function to load expenses from a CSV file
def load_expenses()
    global expenses
    try
        with open(’expenses.csv’, ’r’) as csvfile
            reader = csv.DictReader(csvfile)
            # Ensure we have the right columns
            if ’date’ not in reader.fieldnames or ’category’ not in reader.fieldnames or ’amount’ not in
reader.fieldnames or ’description’ not in reader.fieldnames
                print(Warning CSV file is missing expected columns!)
                return

            expenses = []
            for row in reader
                try
                    # Ensure the ’amount’ field is a float and handle invalid data
                    row[’amount’] = float(row[’amount’])
                    expenses.append(row)
                except ValueError
                    print(fSkipping invalid row (invalid amount) {row})
            print(Expenses have been loaded from ’expenses.csv’.)
    except FileNotFoundError
        print(No previous expenses file found. Starting fresh.)

# Function to display the menu and get user input
def display_menu()
    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 = input(Enter your choice (1-5) )

        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 the program. Goodbye!)
            break
        else
            print(Invalid choice. Please try again.)

# Main program entry point
def main()
    load_expenses()  # Load previous expenses if any
    display_menu()   # Start the interactive menu

# Run the program
if __name__ == __main__
    main()
```