



TEST PLAN SUMMARY

STUDENT NAME : S.GOKULA NANDHAN

STUDENT ID : 20233027

UOW ID : 20820910

EMAIL : gokula.20233027@iit.ac.lk

COURSEWORK DETAILS

Course	: B.Eng(Hons) Software Engineering
Level	: 04
Module	: 4COS006C0.2 Software Development
Coursework No	: 03
Coursework type	: Individual
Issued date	: 04 th April 2024
Submission due date	: on or before 29 th of April 2024
Module Leader	: Mr.Guhanathan Poravi

CONTENTS

1. Abstract

2. Acknowledgement

3. Specification of the task

- Task overview

- Task executing steps

4. Detailed steps of tasks

- Analyse the task

- Design the programme

- Implimentation and documentation

- Test

ABSTRACT

The module leader, Mr. Guhanadhan Poravi, has tasked me to enhance the Personal Finance Tracker by developing a graphical user interface (GUI) using Tkinter utilizing the software development module. This advanced version should not only display the information from a provided JSON file but also incorporate object-oriented programming (OOP) concepts for the GUI components. This report serves as comprehensive documentation for the project. In the documentation, the problem statement is reviewed and a feasible solution is offered to solve it. To ensure the accuracy and resilience of the GUI application, internal testing procedures have also been conducted, and the report includes comprehensive test cases and validations.

ACKNOWLEDGEMENT

All of the people that helped create this test report for the Personal Financial Tracker GUI project have my sincere gratitude. I want to thank Sir Mr. Guhanathan Poravi, the module leader, and his team sincerely for giving me this wonderful opportunity to work on this assignment. They enabled the assignment to be completed successfully by offering direction, motivation, and unwavering support throughout. Additionally, I want to thank my peers and colleagues for their active participation in the review process and for their insightful comments and constructive criticism. Their insightful comments and ideas improved the report's quality and accuracy. I have made every effort to convey my appreciation to each and every reference and contributors.

Thankyou

SPECIFICATION OF THE TASK

TASK OVERVIEW

The task is to enhance the Personal Finance Tracker by developing a graphical user interface (GUI) using Tkinter. This advanced version should not only display the information from a provided JSON file but also incorporate object-oriented programming (OOP) concepts for the GUI components. Additionally, The application will include a search function and a sorting feature to manage and analyze financial transactions more effectively.

TASK EXECUTING STEPS

1. Analyse the task
2. Design the programme
3. Implimentation and documentation
4. Test

DETAILED STEPS OF THE TASK

1.ANALYSE THE TASK

Key objectives of the task

1. Integrate a GUI using Tkinter and OOP concepts.
2. Load and display data from a JSON file upon GUI invocation.
3. Implement search and sorting functionalities within the GUI.
4. Ensure the application is user-friendly and robust.

2.DESIGN THE PROGRAMME

After analyzing above key objectives I have designed a documentation plan with pseudocode.

Pseudocode as follows

FinanceTrackerGUI CLASS :

Function __init__

Root folder

Title "Personal Finance Tracker"

Call Function create_widgets

Load json file

Button for sort column

Function create_widgets

#frame for table and scrollbar

Widget frame fill

#treeview

Widget headings of treeview

Write text topic for every heading

Insert widget headings data

#scroll bar

Scroll bar size,orientation

Insert scroll bar

#search bar and button

Search bar orientation,place

Insert search bar

Search button button text,place

Insert search button

Function load_transactions

```
    Try
    Open file in read mode
    transactions=load the file
    return transactions
except
    return { }
```

Function display_transactions

```
    For entry in self.treeview.get_children():
        Delete(entry)
```

```
for transaction_type,transactions_data in self.transactions.items():
    for data in transactions_data:
        insert(values=(transaction_type,["Date"],["Amount"])
```

Function search_transactions

```
    INITIALIZE query=self.search_entry.get()
    processed_transactions = {}
    for transaction_type,transactions_data in self.transactions.items():
        processed_data=[data for data in transactions_data if
            query in data["Amount"] or
            query in transaction_type or
            query in data["Date"]]
    IF processed_data THEN
        processed_transactions[transaction_type]=processed_data
```



```

display_transactions(processed_transactions)

i=1
for transaction_type,transactions_data in processed_transactions.items():
    print(f"{i}.Expense_type :{transaction_type}")
    j=1
    for details in transactions_data:
        print(f"{j}.Amount :{data['Amount']}\n Date :{data['Date']}")
        j+=1
    i+=1

```

Function column_click()

```

col_id=treeview.identify_column(event.x)
IF col_id THEN
col=col_id.split("#")[-1]
self.sort_by_column(col)

```

Function sort_by_column

```

tems=treeview.get_children("")
data=[(self.treeview.set(child,col),child)for child in items]
from data sort
for index,(val,child) in enumerate(data):
    from treeview move(child,"",index)

```

Function main

```

root = tk.Tk()
app = FinanceTrackerGUI(root)

```

```
app.display_transactions(app.transactions)
root.mainloop()
```

3.Implimentation and documentation

Screen shots of implemented code

PAGE 01

```
coursework_3_20233027.py - C:\Users\Goku\Documents\python done by me\assignment 3\coursework_3_20233027.py (3.12.2)
File Edit Format Run Options Window Help

import tkinter as tk
from tkinter import ttk
import json

class FinanceTrackerGUI:
    def __init__(self, root):
        # Create a new instance
        self.root = root
        self.root.title("Personal Finance Tracker")
        self.create_widgets()
        self.transactions = self.load_transactions("transactions.json")
        self.treeview.bind("<Button-1>", self.column_click)

    def create_widgets(self):
        # Frame for table and scrollbar
        self.frame = ttk.Frame(self.root)
        self.frame.pack(fill=tk.BOTH, expand=True)

        # Treeview for displaying transactions
        self.treeview = ttk.Treeview(self.frame, columns= ("Expense", "Date", "Amount"), show="headings")
        self.treeview.heading("Expense", text="Type of expense")
        self.treeview.heading("Date", text="Date")
        self.treeview.heading("Amount", text="Amount")
        self.treeview.pack(side=tk.LEFT, fill=tk.BOTH, expand=True)

        # Scrollbar for the Treeview
        self.scrollbar = ttk.Scrollbar(self.frame, orient=tk.VERTICAL, command=self.treeview.yview)
        self.treeview.configure(yscrollcommand=self.scrollbar.set)
        self.scrollbar.pack(side=tk.RIGHT, fill=tk.Y)

        # Search bar and button
        self.search_entry = ttk.Entry(self.root)
        self.search_entry.pack(padx=5, pady=5, fill=tk.BOTH)

        # search button
        self.search_button = ttk.Button(self.root, text="Search", command=self.search_transactions)
        self.search_button.pack(padx=5, pady=5, side=tk.RIGHT)

    def load_transactions(self, filename):
        # open the file transactions.json
        try:
            load_file = open(filename, "r")
            transactions = json.load(load_file)
            return transactions
        except FileNotFoundError:
            return {}

Ln 80 Col: 0
```

PAGE 02

```
coursework_3_20233027.py - C:\Users\Goku\Documents\python done by me\assignment 3\coursework_3_20233027.py (3.12.2)
File Edit Format Run Options Window Help

    return {}

def display_transactions(self, transactions):
    # Remove existing entries
    for entry in self.treeview.get_children():
        self.treeview.delete(entry)

    # Add transactions to the treeview
    for transaction_type, transactions_data in self.transactions.items():
        for data in transactions_data:
            self.treeview.insert("", "end", values=(transaction_type, data["Date"], data["Amount"]))

def search_transactions(self):
    # Placeholder for search functionality
    query = self.search_entry.get().lower()
    processed_transactions = {}
    # to search transactions data based from the search query
    for transaction_type, transactions_data in self.transactions.items():
        processed_data = [data for data in transactions_data if
                           query in str(data["Amount"]).lower() or
                           query in transaction_type.lower() or
                           query in data["Date"].lower()]
        if processed_data:
            processed_transactions[transaction_type] = processed_data

    self.display_transactions(processed_transactions)
    i = 1
    for transaction_type, transactions_data in processed_transactions.items():
        print(f"[{i}].Expense_type : {transaction_type}")
        j = 1
        for details in transactions_data:
            print(f"[{j}].Amount : {data['Amount']}\n Date : {data['Date']}")
            j += 1
        i += 1

def column_click(self, event):
    # to get column headings for toggle sort
    col_id = self.treeview.identify_column(event.x)
    if col_id:
        col = col_id.split("#")[-1]
        self.sort_by_column(col)

def sort_by_column(self, col, reverse=None):
    # place holder for sorting functionality

Ln: 92 Col: 48
```

PAGE 03

```
coursework_3_20233027.py - C:\Users\Goku\Documents\python done by me\assignment 3\coursework_3_20233027.py (3.12.2)
File Edit Format Run Options Window Help

        processed_transactions[transaction_type] = processed_data

    self.display_transactions(processed_transactions)
    i = 1
    for transaction_type, transactions_data in processed_transactions.items():
        print(f"[{i}].Expense_type : {transaction_type}")
        j = 1
        for details in transactions_data:
            print(f"[{j}].Amount : {data['Amount']}\n Date : {data['Date']}")
            j += 1
        i += 1

def column_click(self, event):
    # to get column headings for toggle sort
    col_id = self.treeview.identify_column(event.x)
    if col_id:
        col = col_id.split("#")[-1]
        self.sort_by_column(col)

def sort_by_column(self, col, reverse=None):
    # place holder for sorting functionality
    order = None
    if order == False:
        order = True
    else:
        order = False

    items = self.treeview.get_children("")
    data = [(self.treeview.set(child, col), child) for child in items]
    data.sort(reverse=order)
    for index, (val, child) in enumerate(data):
        self.treeview.move(child, "", index)

def main():
    # main function
    root = Tk()
    app = FinanceTrackerGUI(root)
    app.display_transactions(app.transactions)
    root.mainloop()

if __name__ == "__main__":
    main()

Ln: 84 Col: 33
```

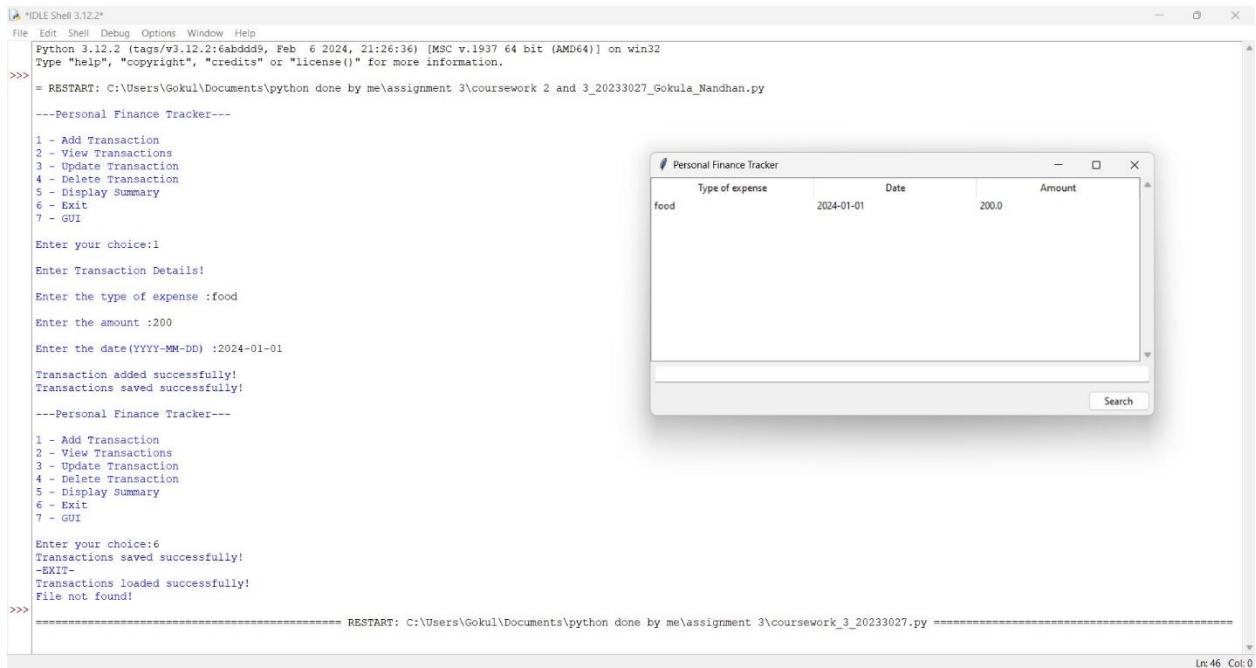
4.Test

THE FOLLOWING CHARTS SHOWS THE ENTIRE TEST PHASE OF THE EXECUTED CODE

Test case	Discription	Inputs	Expected Output	Actual Output	Remarks
01	Open the file in GUI	Input transactions in coursework 2 python code	If run GUI python code Display in GUI	Did expected output	Test pass
02	Add more transactions	Input transactions in coursework 2 python code	If run GUI python code Display in GUI	Did expected output	Test pass
03	Check the scroll bar in GUI	Inpu more transactions to the coursework 2 python code	Newly created a scroll bar	Did expected output	Test pass
04	Sort function	Click the type of expense and date in GUI	Sorted by date	Did expected output	Test pass
05	Search a type of expense	Type in searchbar And search	Display searched row	Did not expected output	Test Fail
06	Open GUI in coursework 2	Click the coursework file with python launcher and choice=7	Open GUI	Did expected output	Test pass

SCREEN SHOT OF THE TEST CASES

TEST 01



```
Python 3.12.2 (tags/v3.12.2:6abddd9, Feb 6 2024, 21:26:36) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\Gokul\Documents\python done by me\assignment 3\coursework 2 and 3_20233027_Gokula_Nandhan.py

---Personal Finance Tracker---

1 - Add Transaction
2 - View Transactions
3 - Update Transaction
4 - Delete Transaction
5 - Display Summary
6 - Exit
7 - GUI

Enter your choice:1

Enter Transaction Details!

Enter the type of expense :food

Enter the amount :200

Enter the date(YYYY-MM-DD) :2024-01-01

Transaction added successfully!
Transactions saved successfully!

---Personal Finance Tracker---

1 - Add Transaction
2 - View Transactions
3 - Update Transaction
4 - Delete Transaction
5 - Display Summary
6 - Exit
7 - GUI

Enter your choice:6
Transactions saved successfully!
-EXIT-
Transactions loaded successfully!
File not found!

>>>
===== RESTART: C:\Users\Gokul\Documents\python done by me\assignment 3\coursework 3_20233027.py =====
```

Type of expense	Date	Amount
food	2024-01-01	200.0

TEST 02

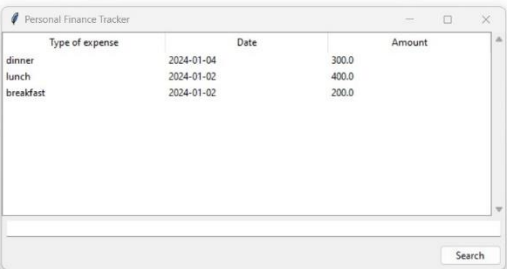
```
"IDLE Shell 3.12.2"
File Edit Shell Debug Options Window Help
Python 3.12.2 (tags/v3.12.2:6abddd9, Feb 6 2024, 21:26:36) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\Gokul\Documents\python done by me\assignment 3\coursework 2 and 3_20233027_Gokula_Nandhan.py

---Personal Finance Tracker---
1 - Add Transaction
2 - View Transactions
3 - Update Transaction
4 - Delete Transaction
5 - Display Summary
6 - Exit
7 - GUI

Enter your choice:1
Enter Transaction Details!
Enter the type of expense :dinner
Enter the amount :300
Enter the date(YYYY-MM-DD) :2024-01-04
Transaction added successfully!
Transactions saved successfully!

---Personal Finance Tracker---
1 - Add Transaction
2 - View Transactions
3 - Update Transaction
4 - Delete Transaction
5 - Display Summary
6 - Exit
7 - GUI

Enter your choice:1
Enter Transaction Details!
Enter the type of expense :lunch
Enter the amount :400
Enter the date(YYYY-MM-DD) :2024-01-02
```



Type of expense	Date	Amount
dinner	2024-01-04	300.0
lunch	2024-01-02	400.0
breakfast	2024-01-02	200.0

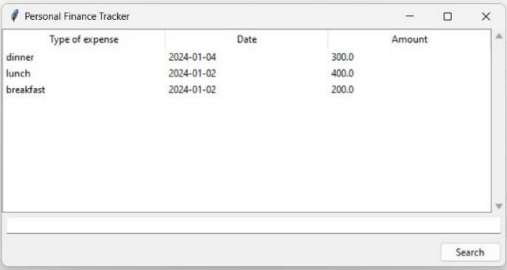
```
"IDLE Shell 3.12.2"
File Edit Shell Debug Options Window Help
6 - Exit
7 - GUI

Enter your choice:1
Enter Transaction Details!
Enter the type of expense :lunch
Enter the amount :400
Enter the date(YYYY-MM-DD) :2024-01-02
Transaction added successfully!
Transactions saved successfully!

---Personal Finance Tracker---
1 - Add Transaction
2 - View Transactions
3 - Update Transaction
4 - Delete Transaction
5 - Display Summary
6 - Exit
7 - GUI

Enter your choice:1
Enter Transaction Details!
Enter the type of expense :breakfast
Enter the amount :200
Enter the date(YYYY-MM-DD) :2024-01-02
Transaction added successfully!
Transactions saved successfully!

---Personal Finance Tracker---
1 - Add Transaction
2 - View Transactions
3 - Update Transaction
4 - Delete Transaction
5 - Display Summary
6 - Exit
```



Type of expense	Date	Amount
dinner	2024-01-04	300.0
lunch	2024-01-02	400.0
breakfast	2024-01-02	200.0

TEST 03

Overview

Type of expense	Date	Amount
amal	2024-01-01	2000.0
phone	2024-01-01	32000.0
car service	2024-01-02	7000.0
guild money	2024-01-03	400.0
cycle	2024-01-03	1000.0
breakfast	2024-01-04	300.0
breakfast	2024-02-09	200.0
lunch	2024-01-04	400.0
dinner	2024-01-04	200.0
wifi	2024-01-05	1500.0
tv	2024-01-06	2000.0
medicine mom	2024-01-07	3000.0
medicine dad	2024-01-08	4500.0
gym	2024-01-10	3000.0
loan	2024-01-11	20000.0
holiday	2024-04-09	4000.0
wedding	2024-02-04	1000.0
birthday	2024-03-01	1200.0
perahera	2024-08-09	5000.0
sipadha	2024-04-14	5000.0
newyear	2024-04-10	12000.0
cricket	2024-03-09	3000.0
accident	2024-02-05	9000.0
umbrella	2024-05-01	450.0
waterbottle	2024-01-22	1200.0

After creation of scroll bar

```
"IDLE Shell 3.12.2"
File Edit Shell Debug Options Window Help

Enter the date(YYYY-MM-DD) :2024-05-01
Transaction added successfully!
Transactions saved successfully!

---Personal Finance Tracker---

1 - Add Transaction
2 - View Transactions
3 - Update Transaction
4 - Delete Transaction
5 - Display Summary
6 - Exit
7 - GUI

Enter your choice:1

Enter Transaction Details!

Enter the type of expense :waterbottle

Enter the amount :1200

Enter the date(YYYY-MM-DD) :2024-01-22
Transaction added successfully!
Transactions saved successfully!

---Personal Finance Tracker---

1 - Add Transaction
2 - View Transactions
3 - Update Transaction
4 - Delete Transaction
5 - Display Summary
6 - Exit
7 - GUI

Enter your choice:6
Transactions saved successfully!
-EXIT-
Transactions loaded successfully!
File not found!

>>> = RESTART: C:\Users\Gokul\Documents\python done by me\assignment 3\coursework_3_20233027.py
```

Type of expense	Date	Amount
amal	2024-01-01	2000.0
phone	2024-01-01	32000.0
car service	2024-01-02	7000.0
guild money	2024-01-03	400.0
cycle	2024-01-03	1000.0
breakfast	2024-01-04	300.0
breakfast	2024-02-09	200.0
lunch	2024-01-04	400.0
dinner	2024-01-04	200.0
wifi	2024-01-05	1500.0


```
"IDLE Shell 3.12.2"
File Edit Shell Debug Options Window Help

Enter the date(YYYY-MM-DD) :2024-05-01
Transaction added successfully!
Transactions saved successfully!
---Personal Finance Tracker---
1 - Add Transaction
2 - View Transactions
3 - Update Transaction
4 - Delete Transaction
5 - Display Summary
6 - Exit
7 - GUI
Enter your choice:1
Enter Transaction Details!
Enter the type of expense :waterbottle
Enter the amount :1200
Enter the date(YYYY-MM-DD) :2024-01-22
Transaction added successfully!
Transactions saved successfully!
---Personal Finance Tracker---
1 - Add Transaction
2 - View Transactions
3 - Update Transaction
4 - Delete Transaction
5 - Display Summary
6 - Exit
7 - GUI
Enter your choice:6
Transactions saved successfully!
-EXIT-
Transactions loaded successfully!
File not found!
>>>
= RESTART: C:\Users\Gokul\Documents\python done by me\assignment 3\coursework_3_20233027.py

Ln: 681 Col: 0
```

Type of expense	Date	Amount
wifi	2024-01-05	1500.0
tv	2024-01-06	2000.0
medicine mom	2024-01-07	3000.0
medicine dad	2024-01-08	4500.0
gym	2024-01-10	3000.0
loan	2024-01-11	20000.0
holiday	2024-04-09	4000.0
wedding	2024-02-04	1000.0
birthday	2024-03-01	1200.0
perahera	2024-08-09	5000.0

```
"IDLE Shell 3.12.2"
File Edit Shell Debug Options Window Help

Enter the date(YYYY-MM-DD) :2024-05-01
Transaction added successfully!
Transactions saved successfully!
---Personal Finance Tracker---
1 - Add Transaction
2 - View Transactions
3 - Update Transaction
4 - Delete Transaction
5 - Display Summary
6 - Exit
7 - GUI
Enter your choice:1
Enter Transaction Details!
Enter the type of expense :waterbottle
Enter the amount :1200
Enter the date(YYYY-MM-DD) :2024-01-22
Transaction added successfully!
Transactions saved successfully!
---Personal Finance Tracker---
1 - Add Transaction
2 - View Transactions
3 - Update Transaction
4 - Delete Transaction
5 - Display Summary
6 - Exit
7 - GUI
Enter your choice:6
Transactions saved successfully!
-EXIT-
Transactions loaded successfully!
File not found!
>>>
= RESTART: C:\Users\Gokul\Documents\python done by me\assignment 3\coursework_3_20233027.py

Ln: 681 Col: 0
```

Type of expense	Date	Amount
holiday	2024-04-09	4000.0
wedding	2024-02-04	1000.0
birthday	2024-03-01	1200.0
perahera	2024-08-09	5000.0
snipadha	2024-04-14	5000.0
newyear	2024-04-10	12000.0
cricket	2024-03-09	3000.0
accident	2024-02-05	9000.0
umbrella	2024-05-01	450.0
waterbottle	2024-01-22	1200.0

TEST 04

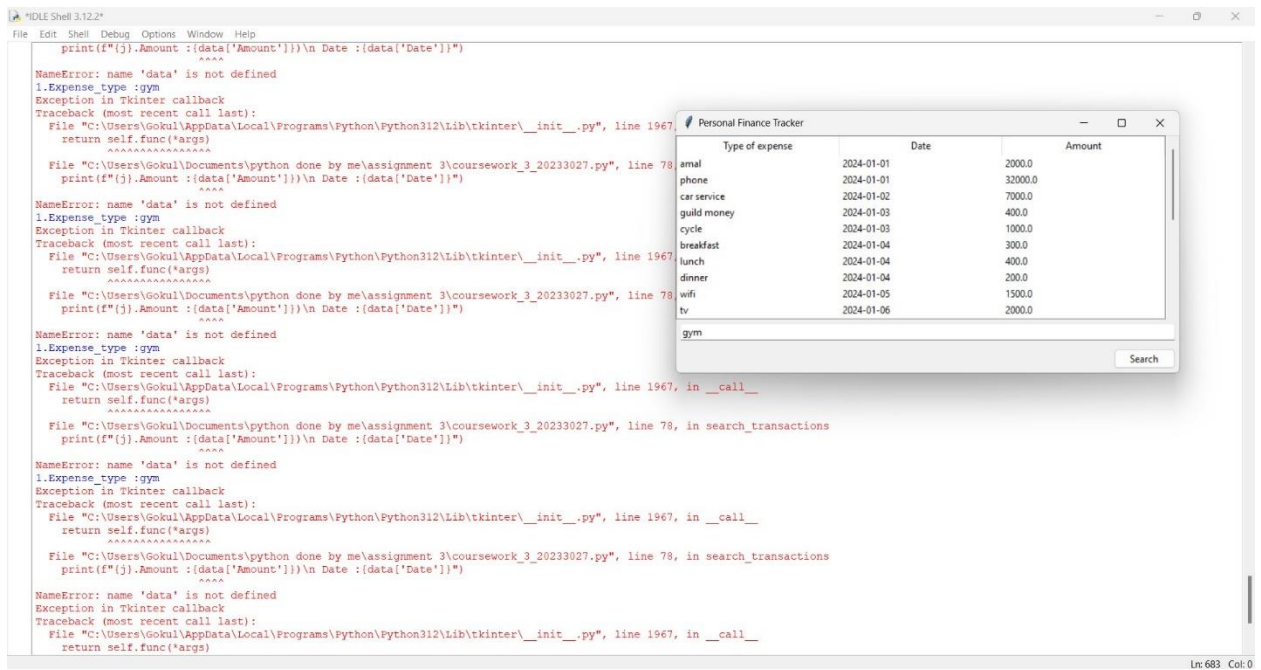
BEFORE SORT

Personal Finance Tracker					
Type of expense	Date		Amount		
amal	2024-01-01		2000.0		
phone	2024-01-01		32000.0		
car service	2024-01-02		7000.0		
guild money	2024-01-03		400.0		
cycle	2024-01-03		1000.0		
breakfast	2024-01-04		300.0		
breakfast	2024-02-09		200.0		
lunch	2024-01-04		400.0		
dinner	2024-01-04		200.0		
wifi	2024-01-05		1500.0		
tv	2024-01-06		2000.0		
medicine mom	2024-01-07		3000.0		
medicine dad	2024-01-08		4500.0		
gym	2024-01-10		3000.0		
loan	2024-01-11		20000.0		
holiday	2024-04-09		4000.0		
wedding	2024-02-04		1000.0		
birthday	2024-03-01		1200.0		
perahera	2024-08-09		5000.0		
stripadha	2024-04-14		5000.0		
newyear	2024-04-10		12000.0		
cricket	2024-03-09		3000.0		
accident	2024-02-05		9000.0		
umbrella	2024-05-01		450.0		
waterbottle	2024-01-22		1200.0		

AFTER SORT

Personal Finance Tracker					
Type of expense	Date		Amount		
amal	2024-01-01		2000.0		
phone	2024-01-01		32000.0		
car service	2024-01-02		7000.0		
guild money	2024-01-03		400.0		
cycle	2024-01-03		1000.0		
breakfast	2024-01-04		300.0		
lunch	2024-01-04		400.0		
dinner	2024-01-04		200.0		
wifi	2024-01-05		1500.0		
tv	2024-01-06		2000.0		
medicine mom	2024-01-07		3000.0		
medicine dad	2024-01-08		4500.0		
gym	2024-01-10		3000.0		
loan	2024-01-11		20000.0		
waterbottle	2024-01-22		1200.0		
wedding	2024-02-04		1000.0		
accident	2024-02-05		9000.0		
breakfast	2024-02-09		200.0		
birthday	2024-03-01		1200.0		
cricket	2024-03-09		3000.0		
holiday	2024-04-09		4000.0		
newyear	2024-04-10		12000.0		
stripadha	2024-04-14		5000.0		
umbrella	2024-05-01		450.0		
perahera	2024-08-09		5000.0		

Test 05



TEST 06

