UNIVERSITY OF WESTMINSTER

INFORMATICS INSTITUTE OF TECHNOLOGY

TEST PLAN SUMMARY

STUDENT NAME : S.GOKULA NANDHAN

STUDENT ID : 20233027

UOW ID :20820910

ASSIGNMENT RELATED DETAILS

Course	:Level 04(SE/CS) 2024 January Intake
Module	:4COS006C0.2 Software Development
Assignment No	:02
Assignment type	:Individual
Issued date	: 30th March 2024
Submission due date	:on or before 15th April 2024
Module leader	:Mr.Guhanathan Poravi

STUDENT DETAILS

Student Name	: S.Gokula Nandhan
Student ID	: 20233027
UOW number	: 20820910
Course	:B.Eng(Hons) Software Engineering
Email address	:gokula.20233027@gmail.com

CONTENTS

ABSTRACT

Mr. Guhanathan Poravi, the module leader, has assigned the job of creating a Personal Finance Tracker utilising the 4COSC006C0.2 Software Development module. This report acts as thorough documentation for the project, which entails using the basic of Python programming and JSON serialisation to create a Dictionary-based application. The problem statement is examined in the documentation, and a workable solution is suggested to address it. Internal testing processes have also been carried out, and the report contains thorough test cases and validations to guarantee the application's accuracy and resilience.

ACKNOWLEDGEMENT

My heartfelt appreciation goes out to everyone who contributed to the creation of this assignment report for the application of Personal Financial Tracker.Above all, I would want to express my sincere gratitude to the module leader, Sir ,Mr. Guhanathan Poravi, and his team for providing me an great opportunity to work on this assignment. They provided guidance, inspiration, and constant support throughout the entire assignment, which allowed it to conclude successfully.
In addition, I would like to express my gratitude to my peers and colleagues who actively contributed constructive criticism and perceptive remarks during the review process. The report's quality and accuracy were enhanced by their perceptive criticisms and suggestions.
Though I have tried my best to express my gratitude to everyone of the contributors.

Thankyou

SPECIFICATION OF THE TASK

TASK OVERVIEW

The goal of this assignment is to use Python to create a Personal Finance Tracker while concentrating on core programming concepts including input/output, functions, loops, dictionaries, and input validation.The application will now manage financial transactions using dictionaries rather than lists, as the original specification used lists to represent different types of expenses and values being lists of expenses.The motive of this update intends to improve comprehension of programme design, testing, and data handling in a real-world setting, with an emphasis on file input/output for large-scale data processing.

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

These are the key factors to design this application.

1. Apply essential Python programming constructs.
2. Manage data using dictionary manipulations for more efficient data retrieval.
3. Implement CRUD operations on financial transactions, leveraging JSON serialization for data storage.
4. Introduce file operations for bulk reading of transactions from a text file, enhancing the program's usability for larger data sets.

5. Conduct thorough testing and validation to ensure program robustness.

2.DESIGN THE PROGRAMME

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

Psuedocode as follows

Psuedocode for the programme

```
Start
IMPORT json

INITIALIZE transactions={}

FUNCTION load_transactions()
    INITIALIZE global transactions
    OPENFILE " transactions.json" for  read
    INITIALIZE a variable "load_file" to
      serialize python list into JSON data
    CLOSEFILE
    DISPLAY Transaction loaded successfully!
ENDFUNCTION

FUNCTION save_transactions()
    OPENFILE "transactions.json" for write
    INITIALIZE a variable "save_file" to
    Diserilize the transactions into python
    CLOSEFILE
    DISPLAY Transactions saved successfully!
ENDFUNCTION

FUNCTION read_bulk_transactions_from_file()
    OPENFILE "exepense.txt" as expense for read
    INITIALIZE a variable "text_file" for expense
    IF transaction_type in transactions THEN
        Append transactions[transaction_type] with values
    ELSE
        GET transactions[transaction_type]=Amount,date
    ENDIF
END FUNCTION
```

```
FUNCTION add_transactions()
        Display   Enter Transaction Details!
        GET transaction_type
        DO WHILE true
           GET amount
           IF amount>=0 THEN
                BREAK
           ELSE
                DISPLAY Negative value does not get for an amount
           ENDIF
        GET date
        IF transaction_type not in transactions THEN
                Transactions[transaction_type] is empty list
        ENDIF
        APPEND amount and date to transactions
        PRINT Transaction added successfully!
ENDFUNCTION



FUNCTION view_transactions()
        IF transactions=[] THEN
                DISPLAY No more transactions to view!
        ELSE
            FOR  transaction_type in transactions
              DISPLAY transaction_type
              FOR sub_transaction_dictionary in transactions[transaction_type]
                DISPLAY(sub_trasaction_dictionary)
        ENDIF
ENDFUNCTION


FUNCTION update_transaction()
        IF transactions=[] THEN
```

```
                DISPLAY No more transactions to update!
        ELSE
                DISPLAY check Transaction type and index you want to update
                FUNCTION view_update()
                ENDFUNCTION
                DISPLAY Now you can update!
                GET transaction_type
                IF transaction_type in transactions THEN
                   DOWHILE true
                      GET new_amount
                       IF new_amount>=0 THEN
                          BREAK
                       ELSE
                          DISPLAY Negative value does not get for an amount!
                   GET new_date
                   ADD new_amount,new_date to the global dictionary
                   DISPLAY Transaction updated successfully!
                ELSE
                   DISPLAY Transaction not found!
                ENDIF
END FUNCTION


FUNCTION delete_transaction()
        IF transactions=[] THEN
                DISPLAY No more transactions to delete!
        ELSE
                DISPLAY Check transaction type and index you want to delete
                FUNCTION view_transactions()
                END FUNCTION
                DISPLAY now you can delete it!
                GET transaction_type
                IF transaction_type in transactions THEN
                   DOWHILE true
                   GET index
                IF 0<=index<len(transactions[transaction_type]) THEN
             DELETE transaction mentioned index
             DISPLAY Transaction deleted successfully!
             FUNCTION save_transactions()
                BREAK
             ELSE
               DISPLAY Invalid index.Please try again!
             ENDIF
      ELSE
          DISPLAY Type of expense not found!
      ENDIF
END FUNCTION


FUNCTION display_summary()
      total_expense=0
      FOR transaction_type in transactions DO
          FOR sub_transaction_dictionary in transactions[transaction_type]
             total_expense+=sub_transaction_dictionary[Amount]
DISPLAY your total expense is "total_expense"
END FUNCION


FUNCTION main_menu()
load_transactions()
read_bulk_transactions_from_file()
DOWHILE True
      Display ---Personal Finance Tracker---
      Display 1 - Add Transaction
      Display 2 - View Transactions
      Display 3 - Update Transaction
      Display 4 - Delete Transaction
      Display 5 - Display Summary
      Display 6 – Exit
      Prompt for  choice
```

```
Get choice
IF choice =1 THEN
         add_transaction()
   ELSE IF choice=2 THEN
            view_transactions()
     ELSE IF choice=3 THEN
               update_transaction()
        ELSE IF choice=4 THEN
                 delete_transaction()
           ELSE IF choice=5 THEN
                     display_summary()
             ELSE IF choice=6 THEN
                    save_transactions()
                    Display -EXIT-
                    break
         ELSE
                  Display Invalid choice.Please try again!
                  ENDIF
             ENDIF
           ENDIF
        ENDIF
     ENDIF
  ENDIF
ENDIF
ENDWHILE
ENDFUNCTION
main_menu()
END
```

3.IMPLIMENTATION AND DOCUMENTATION

After the designing phase the following code would be executed

PYTHON CODE

```
import json

#Global dictionary to store transactions
transactions={}



'''File handling functions'''

#function to load transactions from a json file
def load_transactions():
  global transactions
  load_file=open("transactions.json","r")
  transactions=json.load(load_file)
  load_file.close()
  print("Transactions loaded successfully!")


#function to save transactions to a json file
def save_transactions():
  save_file=open("transactions.json","w")
  json.dump(transactions,save_file)
  save_file.close()
  print("Transactions saved successfully!")


#read transactions from a text file and add them to global dictionary
def read_bulk_transactions_from_file(expense="expense.txt"):
  try:
    text_file=open(expense,"r")
    for values in text_file:
      transaction_type,amount,date=values.strip().split(",")
      if transaction_type in transactions:
        transactions[transaction_type].append({"Amount":float(amount,2),"Date":date})
        else:
            transactions[transaction_type]=[{"Amount":float(amount,2),"Date":date}]
    except FileNotFoundError:
        print("File not found!")



'''Feature implementations'''

#add a new transaction to global dictionary
def add_transaction():
    print("\nEnter Transaction Details!")
    transaction_type=input("\nEnter the type of expense :")

    while True:
        try:
            amount=float(input("\nEnter the amount :"))
            if amount>=0:
                break
            else:
                print("Negative value does not get for an amount!")
        except ValueError as amount_error:
            print(amount_error)

    date=input("\nEnter the date(YYYY-MM-DD) :")

    if transaction_type not in transactions:
        transactions[transaction_type]=[]

    transactions[transaction_type].append({"Amount":amount,"Date":date})
    print("\nTransaction added successfully!")
    save_transactions()



#view all transactions stored in global dictionary
def view_transactions():
    if not transactions:#empty global dictionary
        print("No more transactions to view!")
    else:
        for transaction_type in transactions:
            print(transaction_type)
            for sub_transaction_dictionary in transactions[transaction_type]:
                print(sub_transaction_dictionary)
```

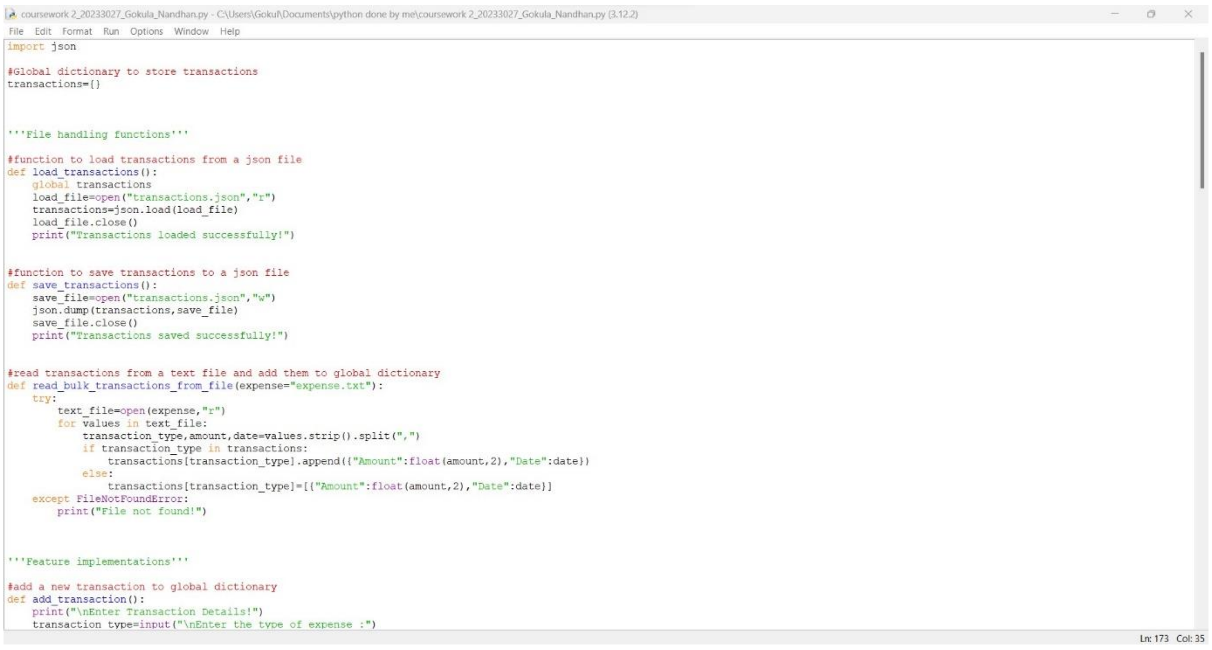
```
#update an existing transaction in the transactions(global dictionary)
def update_transaction():
  if not transactions:
    print("No more transactions to update!")
  else:
    print("check Transaction type and index you want to update!")
    view_transactions()
    print("\nNow you can update it!")

    transaction_type=input("\nEnter type of your expense :")
    if transaction_type in transactions:
      while True:
        try:
          index=int(input("Enter index from purticular type of expense :"))
          if 0 <= index <len(transactions[transaction_type]):
            break
          else:
            print("Invalid index.Please try again!")
        except ValueError as error:
          print(error)


      while True:
        try:
          new_amount=float(input("Enter new amount :"))
          if new_amount >= 0:
            break
          else:
            print("Negative value does not get for an amount!")
        except ValueError as new_amount_error:
          print(new_amount_error)


      new_date=input("Enter new date(YYYY-MM-DD) :")

      #will update new date and amount in nested dictionaries
      transactions[transaction_type][index]["Amount"]=new_amount
      transactions[transaction_type][index]["Date"]=new_date

      print("Transaction updated successfully!")
      save_transactions()
    else:
      print("Transaction not found!")
```

```
#delete an transaction from global dictionary
def delete_transaction():
    if not transactions:
        print("No more transactions to delete!")
    else:
        print("check Transaction type and index you want to delete!")
        view_transactions()
        print("\nNow you can delete it!")

        transaction_type=input("\nEnter type of your expense :")
        if transaction_type in transactions:
            while True:
                    try:
                        index = int(input("Enter index from purticular type of expense :"))

                        if 0 <= index < len(transactions[transaction_type]):
                            del transactions[transaction_type][index]
                            print("Transaction deleted successfully!")
                            save_transactions()
                            break
                        else:
                            print("Invalid index.Please try again!")
                    except ValueError as error:
                        print(error)
        else:
            print("Type of expense not found!")

#function for display summary of total expenses
def display_summary():
    total_expense=0
    for transaction_type in transactions:
        for sub_transaction_dictionary in transactions[transaction_type]:
            total_expense+=sub_transaction_dictionary["Amount"]
    print("Your total expense is",total_expense)


#user interactional function
def main_menu():
    #call functions to load transactions from file
    load_transactions()
    read_bulk_transactions_from_file()
```

```
#main programme loop
while True:
  print("\n---Personal Finance Tracker---")
  print("\n1 - Add Transaction")
  print("2 - View Transactions")
  print("3 - Update Transaction")
  print("4 - Delete Transaction")
  print("5 - Display Summary")
  print("6 - Exit")


  try:
    #prompt user for call the feature implimentation functions
    choice=int(input("\nEnter your choice:"))
    if 1<=choice<=6:
      if choice==1:
        add_transaction()
      elif choice==2:
        view_transactions()
      elif choice==3:
        update_transaction()
      elif choice==4:
        delete_transaction()
      elif choice==5:
        display_summary()
      else:
        choice==6
        save_transactions()
        print("-EXIT-")
        break
    else:
      print("Invalid choice.Please try again!")
  except ValueError as error:
    print(error)


#programme start and calling for main menu function
if __name__=="__main__":
  main_menu()
```

SCREEN SHOTS OF EXECUTED CODE

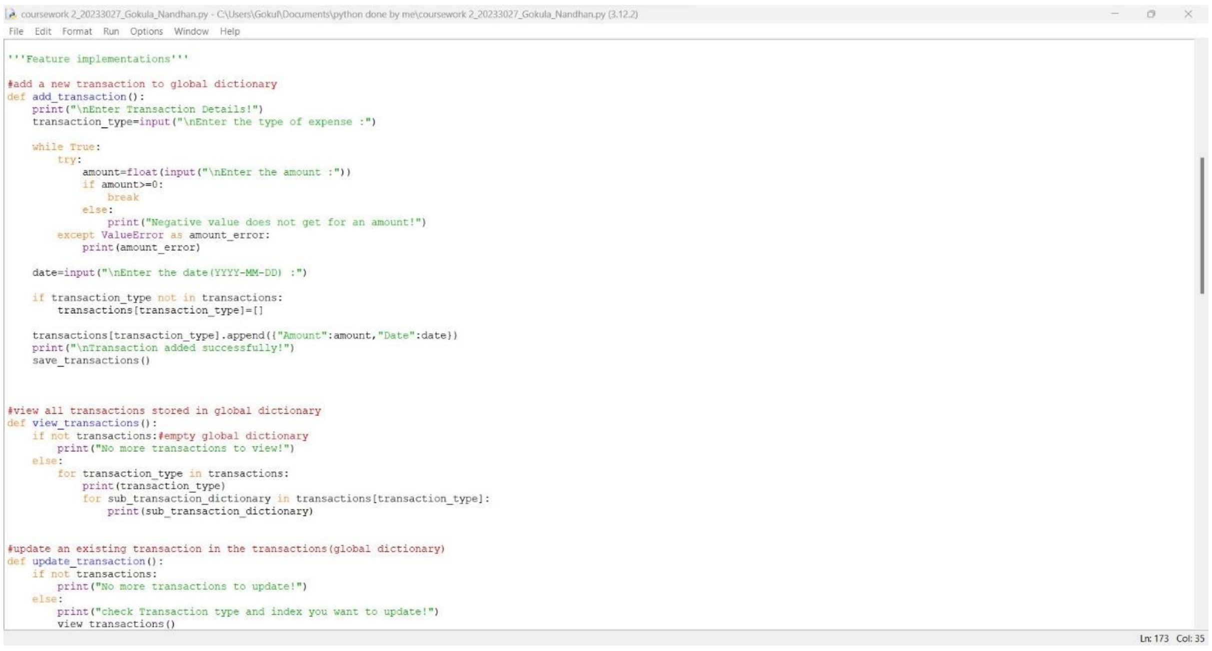
PAGE 1


coursework 2_20233027_Gokula_Nandhan.py - C:\Users\Gokul\Documents\python done by me\coursework 2_20233027_Gokula_Nandhan.py (3.12.2)

```python
import json

#Global dictionary to store transactions
transactions={}



'''File handling functions'''

#function to load transactions from a json file
def load_transactions():
    global transactions
    load_file=open("transactions.json","r")
    transactions=json.load(load_file)
    load_file.close()
    print("Transactions loaded successfully!")


#function to save transactions to a json file
def save_transactions():
    save_file=open("transactions.json","w")
    json.dump(transactions,save_file)
    save_file.close()
    print("Transactions saved successfully!")


#read transactions from a text file and add them to global dictionary
def read_bulk_transactions_from_file(expense="expense.txt"):
    try:
        text_file=open(expense,"r")
        for values in text_file:
            transaction_type,amount,date=values.strip().split(",")
            if transaction_type in transactions:
                transactions[transaction_type].append({"Amount":float(amount,2),"Date":date})
            else:
                transactions[transaction_type]=[{"Amount":float(amount,2),"Date":date}]
    except FileNotFoundError:
        print("File not found!")



'''Feature implementations'''

#add a new transaction to global dictionary
def add_transaction():
    print("\nEnter Transaction Details!")
    transaction_type=input("\nEnter the type of expense :")
```

Ln: 173 Col: 35

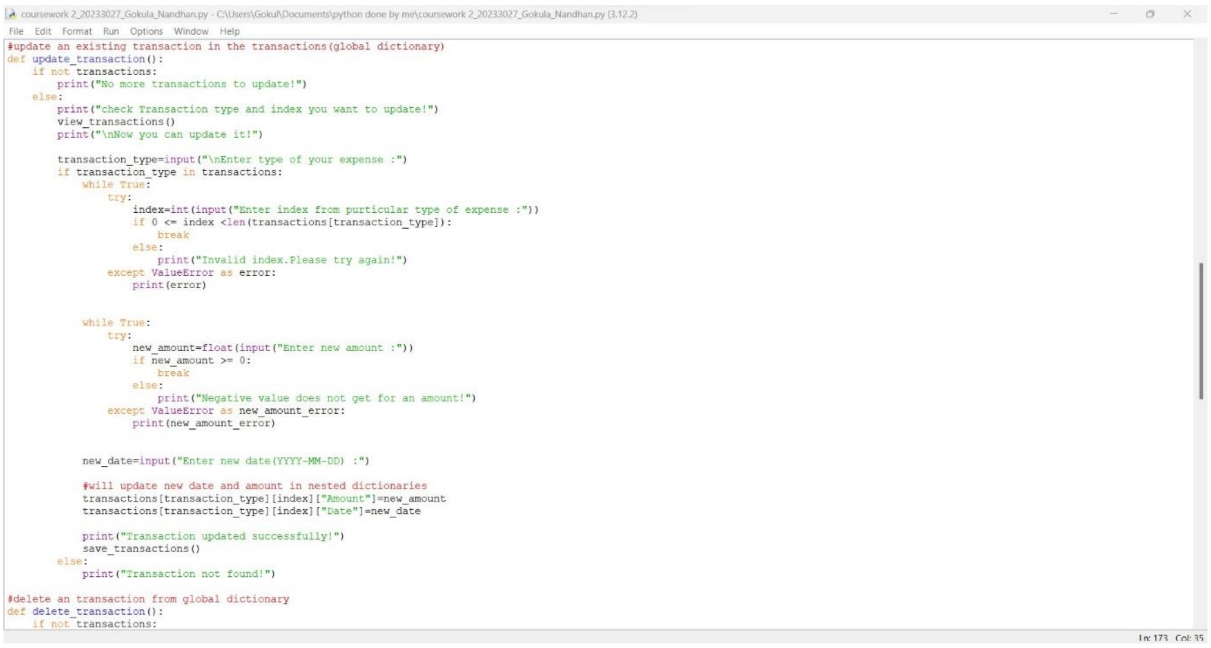
PAGE 2


coursework 2_20233027_Gokula_Nandhan.py - C:\Users\Gokul\Documents\python done by me\coursework 2_20233027_Gokula_Nandhan.py (3.12.2)

```python
'''Feature implementations'''

#add a new transaction to global dictionary
def add_transaction():
    print("\nEnter Transaction Details!")
    transaction_type=input("\nEnter the type of expense :")

    while True:
        try:
            amount=float(input("\nEnter the amount :"))
            if amount>=0:
                break
            else:
                print("Negative value does not get for an amount!")
        except ValueError as amount_error:
            print(amount_error)

    date=input("\nEnter the date(YYYY-MM-DD) :")

    if transaction_type not in transactions:
        transactions[transaction_type]=[]

    transactions[transaction_type].append({"Amount":amount,"Date":date})
    print("\nTransaction added successfully!")
    save_transactions()


#view all transactions stored in global dictionary
def view_transactions():
    if not transactions:#empty global dictionary
        print("No more transactions to view!")
    else:
        for transaction_type in transactions:
            print(transaction_type)
            for sub_transaction_dictionary in transactions[transaction_type]:
                print(sub_transaction_dictionary)


#update an existing transaction in the transactions(global dictionary)
def update_transaction():
    if not transactions:
        print("No more transactions to update!")
    else:
        print("check Transaction type and index you want to update!")
        view_transactions()
```

Ln: 173 Col: 35

PAGE 3



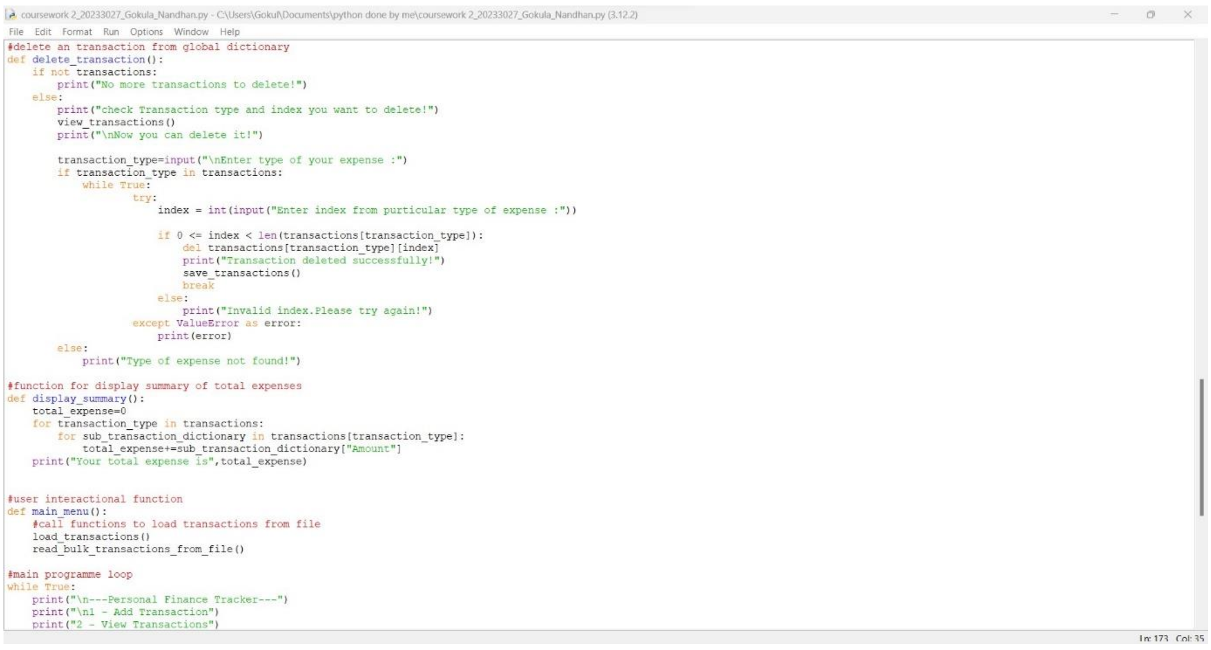
coursework 2_20233027_Gokula_Nandhan.py - C:\Users\Gokul\Documents\python done by me\coursework 2_20233027_Gokula_Nandhan.py (3.12.2)

File Edit Format Run Options Window Help

```python
#update an existing transaction in the transactions(global dictionary)
def update_transaction():
    if not transactions:
        print("No more transactions to update!")
    else:
        print("check Transaction type and index you want to update!")
        view_transactions()
        print("\nNow you can update it!")

        transaction_type=input("\nEnter type of your expense :")
        if transaction_type in transactions:
            while True:
                try:
                    index=int(input("Enter index from purticular type of expense :"))
                    if 0 <= index <len(transactions[transaction_type]):
                        break
                    else:
                        print("Invalid index.Please try again!")
                except ValueError as error:
                    print(error)


            while True:
                try:
                    new_amount=float(input("Enter new amount :"))
                    if new_amount >= 0:
                        break
                    else:
                        print("Negative value does not get for an amount!")
                except ValueError as new_amount_error:
                    print(new_amount_error)


            new_date=input("Enter new date(YYYY-MM-DD) :")

            #will update new date and amount in nested dictionaries
            transactions[transaction_type][index]["Amount"]=new_amount
            transactions[transaction_type][index]["Date"]=new_date

            print("Transaction updated successfully!")
            save_transactions()
        else:
            print("Transaction not found!")

#delete an transaction from global dictionary
def delete_transaction():
    if not transactions:
```

Ln: 173 Col: 35

PAGE 4



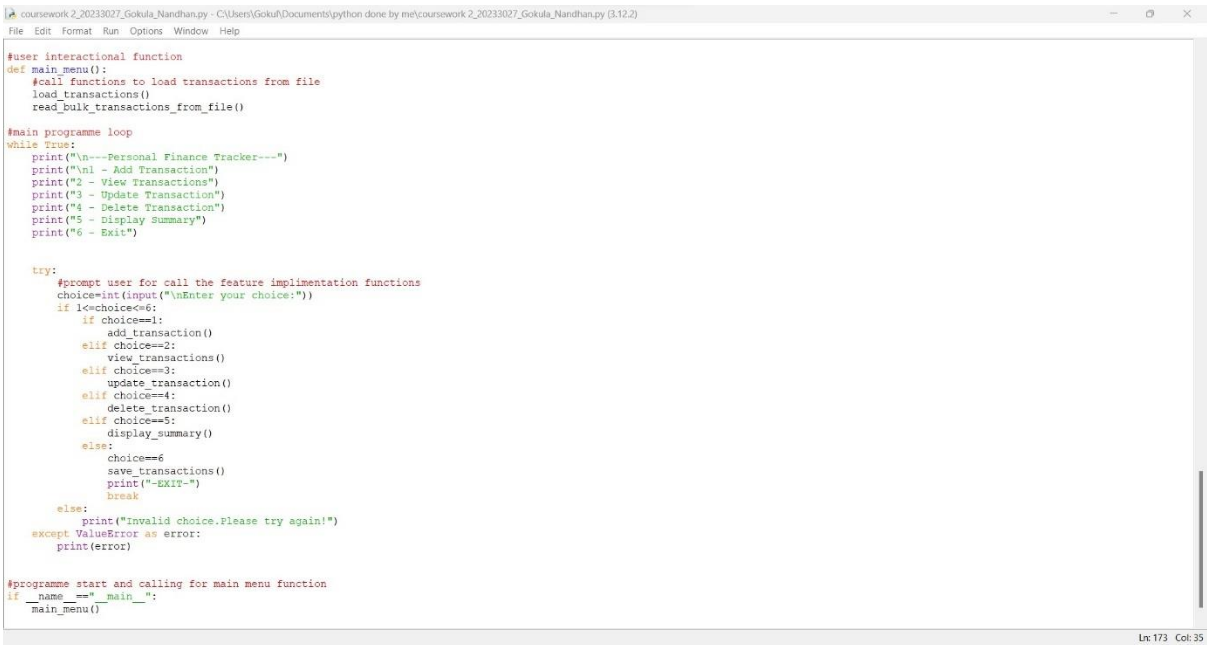
coursework 2_20233027_Gokula_Nandhan.py - C:\Users\Gokul\Documents\python done by me\coursework 2_20233027_Gokula_Nandhan.py (3.12.2)

File Edit Format Run Options Window Help

```python
#delete an transaction from global dictionary
def delete_transaction():
    if not transactions:
        print("No more transactions to delete!")
    else:
        print("check Transaction type and index you want to delete!")
        view_transactions()
        print("\nNow you can delete it!")

        transaction_type=input("\nEnter type of your expense :")
        if transaction_type in transactions:
            while True:
                    try:
                        index = int(input("Enter index from purticular type of expense :"))

                        if 0 <= index < len(transactions[transaction_type]):
                            del transactions[transaction_type][index]
                            print("Transaction deleted successfully!")
                            save_transactions()
                            break
                        else:
                            print("Invalid index.Please try again!")
                    except ValueError as error:
                        print(error)
        else:
            print("Type of expense not found!")

#function for display summary of total expenses
def display_summary():
    total_expense=0
    for transaction_type in transactions:
        for sub_transaction_dictionary in transactions[transaction_type]:
            total_expense+=sub_transaction_dictionary["Amount"]
    print("Your total expense is",total_expense)


#user interactional function
def main_menu():
    #call functions to load transactions from file
    load_transactions()
    read_bulk_transactions_from_file()

#main programme loop
while True:
    print("\n---Personal Finance Tracker---")
    print("\n1 - Add Transaction")
    print("2 - View Transactions")
```

Ln: 173 Col: 35

PAGE 5


coursework 2_20233027_Gokula_Nandhan.py - C:\Users\Gokul\Documents\python done by me\coursework 2_20233027_Gokula_Nandhan.py (3.12.2)
File Edit Format Run Options Window Help
Ln: 173 Col: 35

```python
#user interactional function
def main_menu():
    #call functions to load transactions from file
    load_transactions()
    read_bulk_transactions_from_file()

#main programme loop
while True:
    print("\n---Personal Finance Tracker---")
    print("\n1 - Add Transaction")
    print("2 - View Transactions")
    print("3 - Update Transaction")
    print("4 - Delete Transaction")
    print("5 - Display Summary")
    print("6 - Exit")


    try:
        #prompt user for call the feature implimentation functions
        choice=int(input("\nEnter your choice:"))
        if 1<=choice<=6:
            if choice==1:
                add_transaction()
            elif choice==2:
                view_transactions()
            elif choice==3:
                update_transaction()
            elif choice==4:
                delete_transaction()
            elif choice==5:
                display_summary()
            else:
                choice==6
                save_transactions()
                print("-EXIT-")
                break
        else:
            print("Invalid choice.Please try again!")
    except ValueError as error:
        print(error)


#programme start and calling for main menu function
if __name__=="__main__":
    main_menu()
```

4.TEST

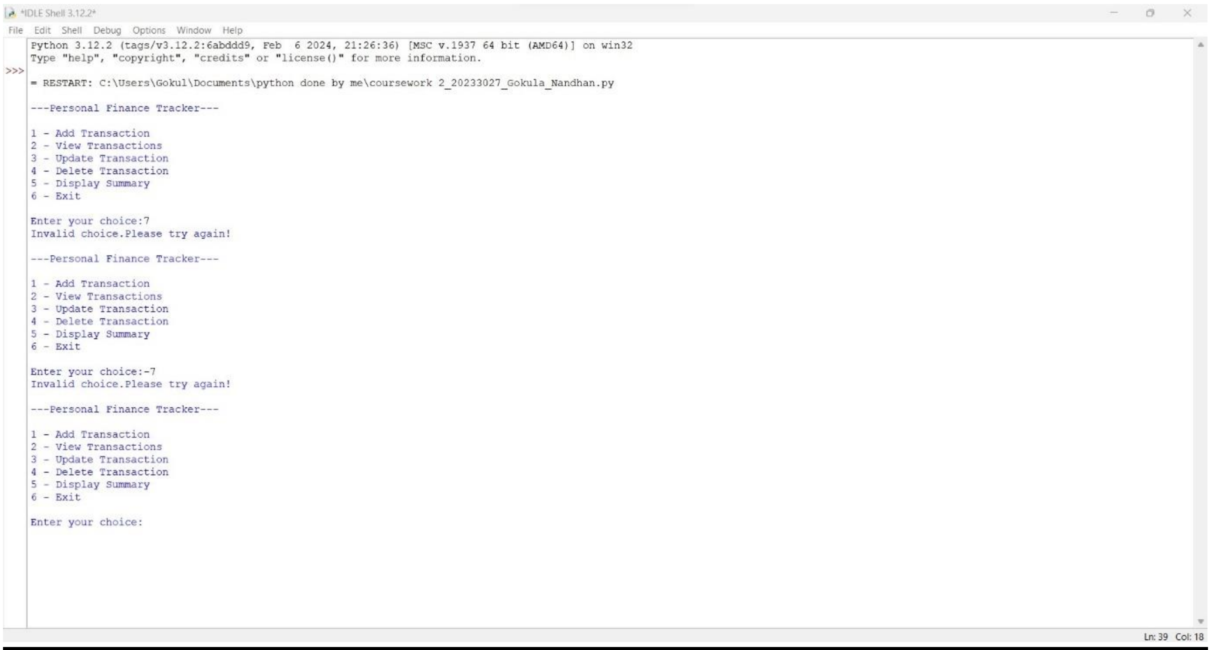
THE FOLLOWING CHARTS SHOWS THE ENTIRE TEST PHASE OF THE EXECUTED CODE

Test case	Discription	Inputs	Expected Output	Actual Output	Remarks
01	Choice random (without a range 1<=choice<=6)	Input a choice=7 Choice=-7	Display Invalid choice.Plese try again	Did expected output	Test pass
02	Choice random (without string data type(string,float,boolean))	Choice=ant Choice=True Choice=3.5	invalid literal for int()	Did expected output	Test pass
03	Choice=1 (add transactions())	Choice=1	Collect the data and display "Transaction added successfully!"	Did expected output	Test pass
04	Choice=1 (add_transaction())	Input negative value for amount	Display error message	Did expected output	Test pass
05	Choice=2 (view_transactions())	Choice=2 After some added transactions	Display transactions	Did expected output	Test pass
06	Choice=3 (update_transaction())	Choice=3 and typing all the needs from application properly	Save all data and show the message	Did expected output	Test pass
07	Choice=3 (update transaction())	input irrilevant key in dictionary	Display error message	Did expected output	Test pass

08	Choice=3 (update transaction())	Input invalid index for nested dictionary	Display error message	Showed expected output	Test pass
09	Choice=3 (update transaction())	Input negative value for new amount	Display error message	Showed Expected output	Test pass
10	Choice=4 (delete transaction())	Input valid data for positive output	Save relevant data and output success message	Showed expected output	Test pass
11	Choice=4 (delete transaction())	Input invalid index for nested dictionary	Display error message	Showed expected output	Test pass
12	Choice=4 (delete transaction())	input irrilevant key in dictionary	Display error message	Showed expected output	Test pass
13	Choice=5 (display_summary())	Choice=5	Display the sum of all expenses	Showed expected output	Test pass
14	Choice=6 (exit())	Choice=6	Save and exit	Showed expected output	Test pass

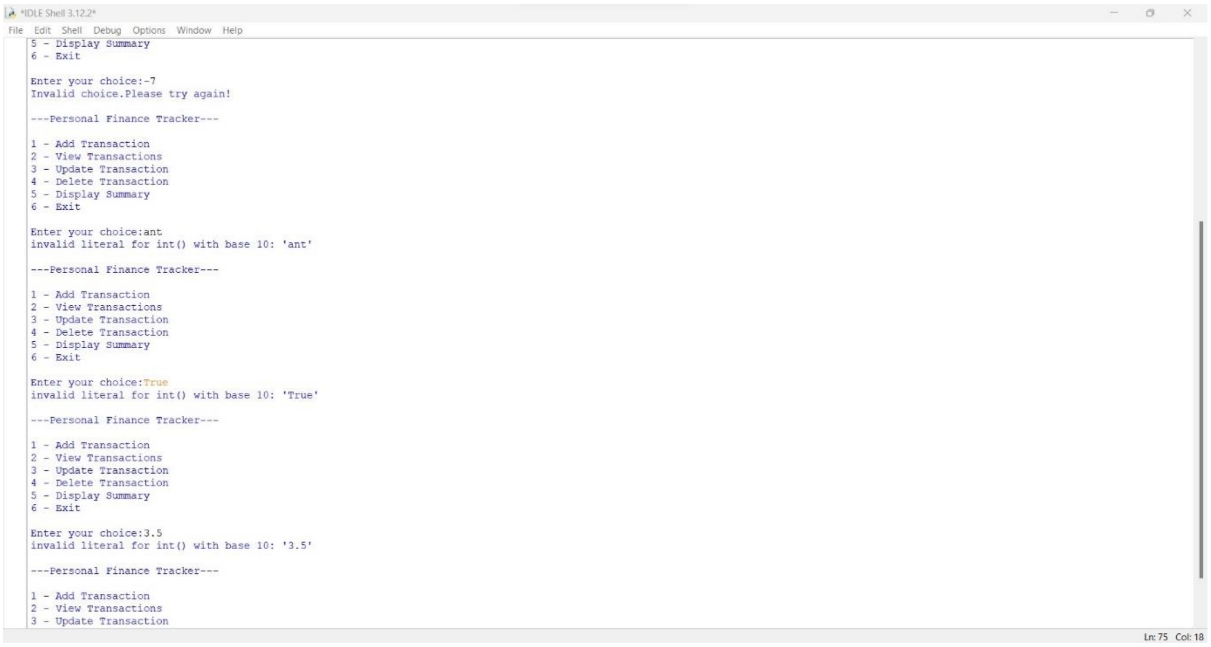
SCREEN SHOTS OF TEST SUMMARY

TEST CASE 01



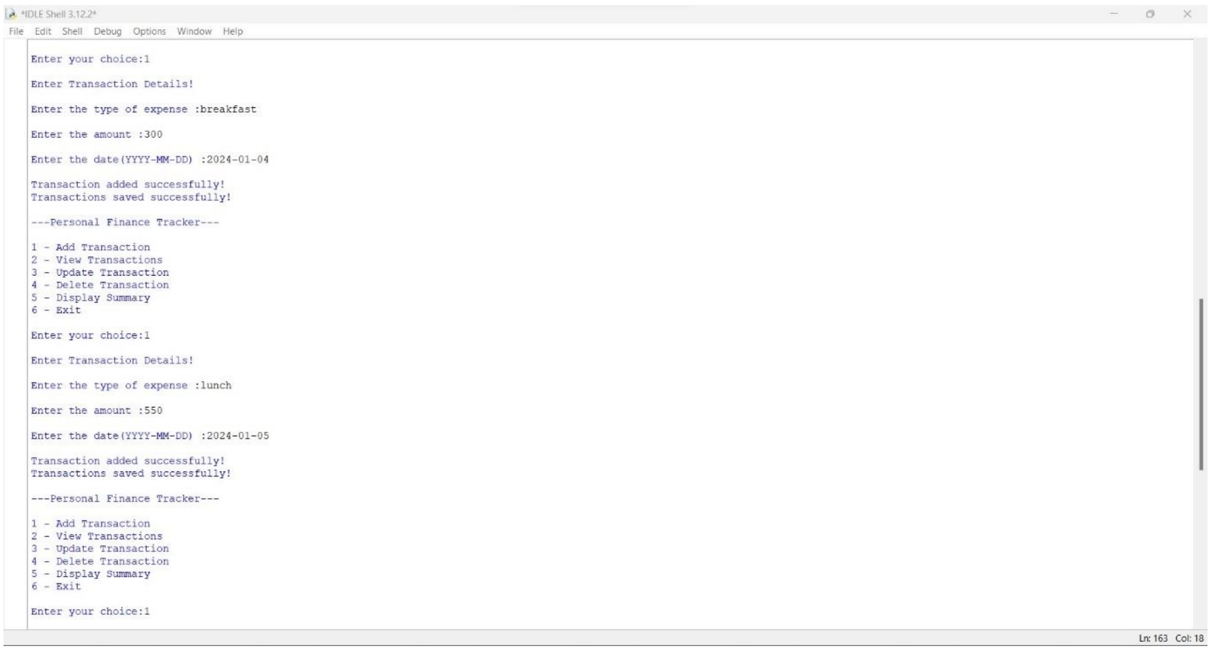
```
Python 3.12.2 (tags/v3.12.2:6abddd9, Feb  6 2024, 21:26:36) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\Gokul\Documents\python done by me\coursework 2_20233027_Gokula_Nandhan.py

---Personal Finance Tracker---

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

Enter your choice:7
Invalid choice.Please try again!

---Personal Finance Tracker---

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

Enter your choice:-7
Invalid choice.Please try again!

---Personal Finance Tracker---

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

Enter your choice:
```

TEST CASE NO 2

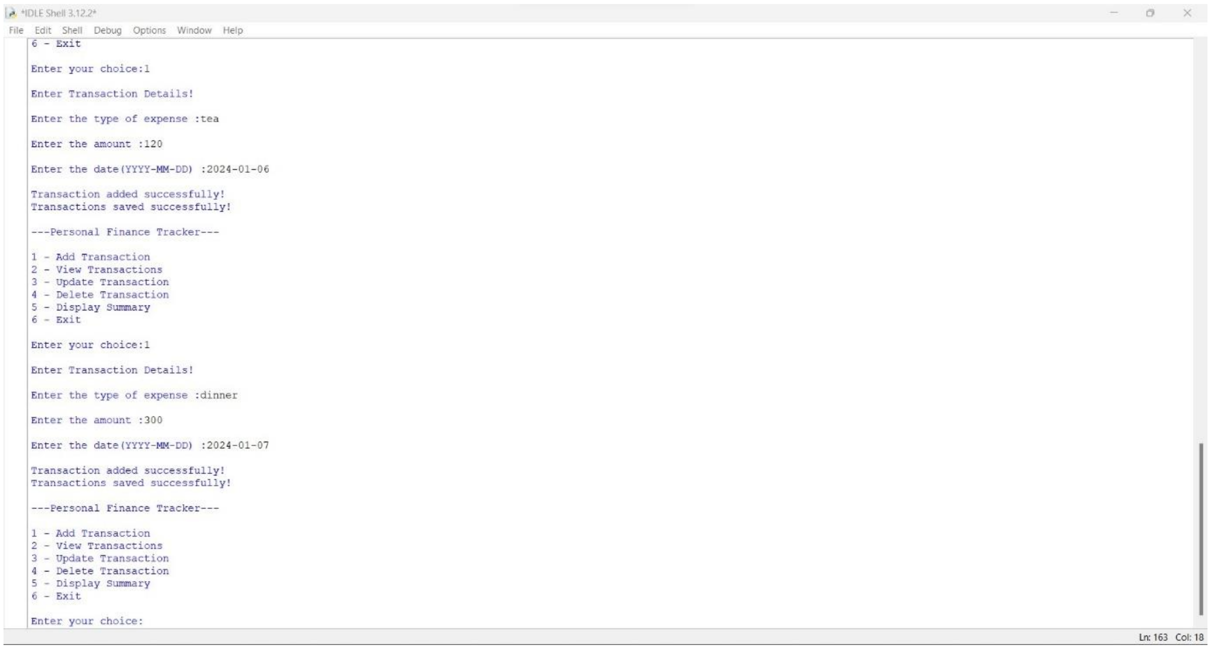


```
5 - Display Summary
6 - Exit

Enter your choice:-7
Invalid choice.Please try again!

---Personal Finance Tracker---

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

Enter your choice:ant
invalid literal for int() with base 10: 'ant'

---Personal Finance Tracker---

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

Enter your choice:True
invalid literal for int() with base 10: 'True'

---Personal Finance Tracker---

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

Enter your choice:3.5
invalid literal for int() with base 10: '3.5'

---Personal Finance Tracker---

1 - Add Transaction
2 - View Transactions
3 - Update Transaction
```

TEST CASE NO 3

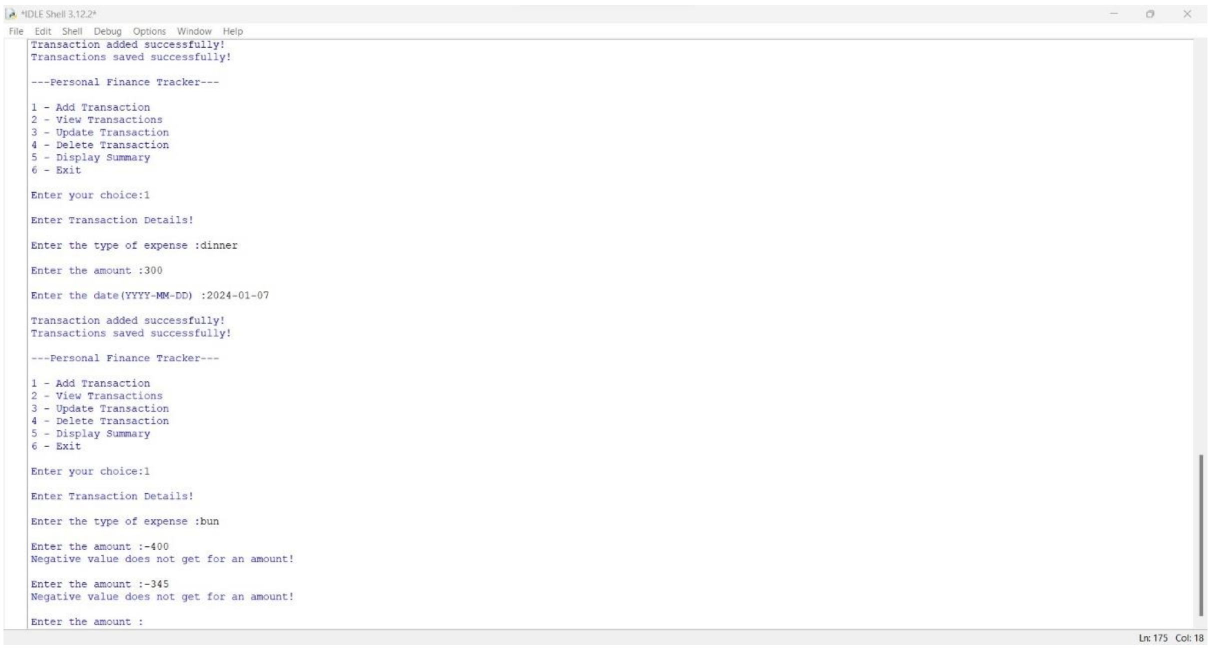


```
Enter your choice:1

Enter Transaction Details!

Enter the type of expense :breakfast

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

Enter your choice:1

Enter Transaction Details!

Enter the type of expense :lunch

Enter the amount :550

Enter the date(YYYY-MM-DD) :2024-01-05

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

Enter your choice:1
```



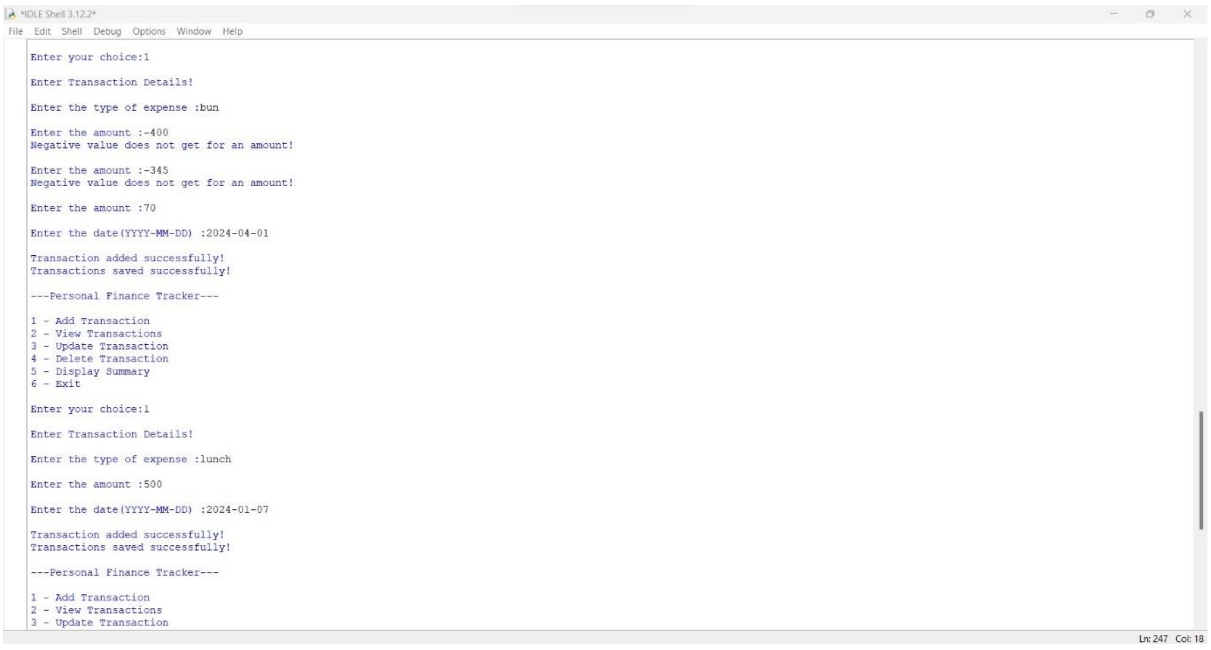
```
6 - Exit

Enter your choice:1

Enter Transaction Details!

Enter the type of expense :tea

Enter the amount :120

Enter the date(YYYY-MM-DD) :2024-01-06

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

Enter your choice:1

Enter Transaction Details!

Enter the type of expense :dinner

Enter the amount :300

Enter the date(YYYY-MM-DD) :2024-01-07

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

Enter your choice:
```

TEST CASE NO 4

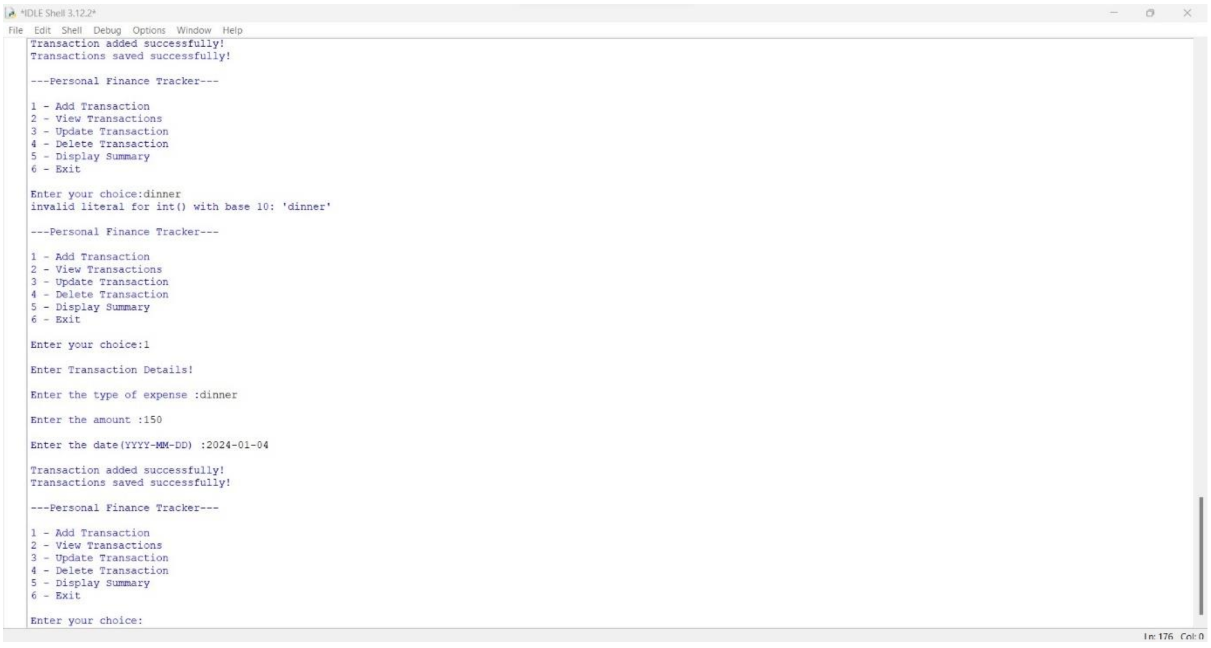


```
Transaction added successfully!
Transactions saved successfully!

---Personal Finance Tracker---

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

Enter your choice:1

Enter Transaction Details!

Enter the type of expense :dinner

Enter the amount :300

Enter the date(YYYY-MM-DD) :2024-01-07

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

Enter your choice:1

Enter Transaction Details!

Enter the type of expense :bun

Enter the amount :-400
Negative value does not get for an amount!

Enter the amount :-345
Negative value does not get for an amount!

Enter the amount :
```

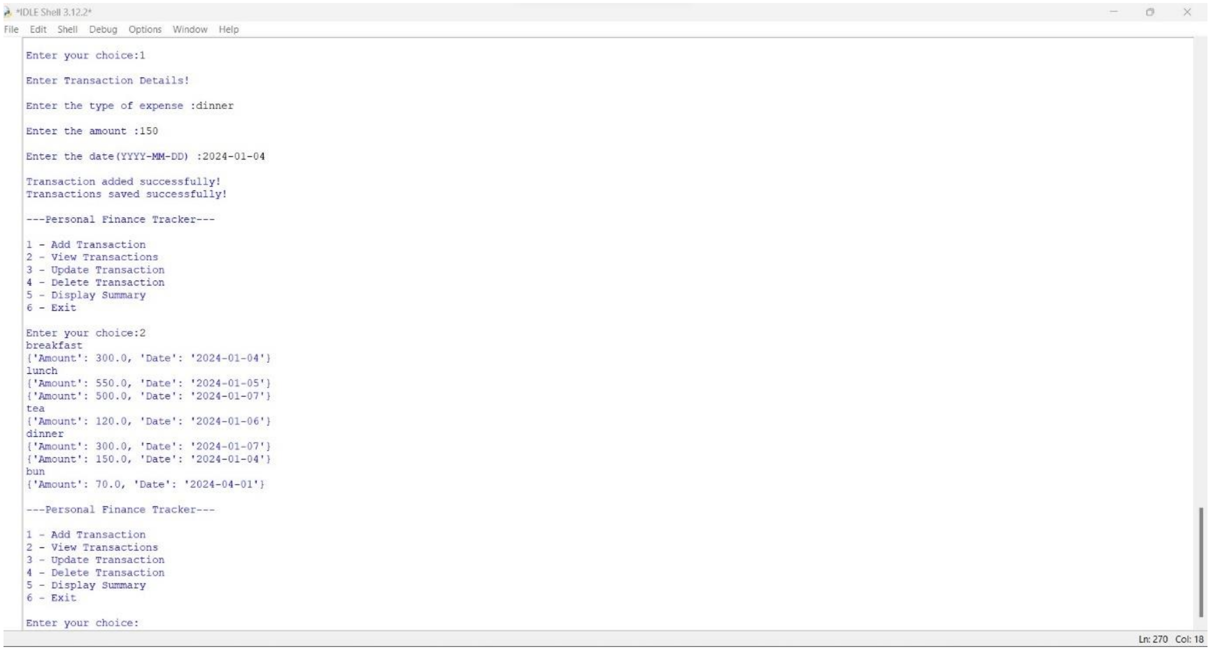
TEST CASE NO 5



```
Enter your choice:1

Enter Transaction Details!

Enter the type of expense :bun

Enter the amount :-400
Negative value does not get for an amount!

Enter the amount :-345
Negative value does not get for an amount!

Enter the amount :70

Enter the date(YYYY-MM-DD) :2024-04-01

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

Enter your choice:1

Enter Transaction Details!

Enter the type of expense :lunch

Enter the amount :500

Enter the date(YYYY-MM-DD) :2024-01-07

Transaction added successfully!
Transactions saved successfully!

---Personal Finance Tracker---

1 - Add Transaction
2 - View Transactions
3 - Update Transaction
```

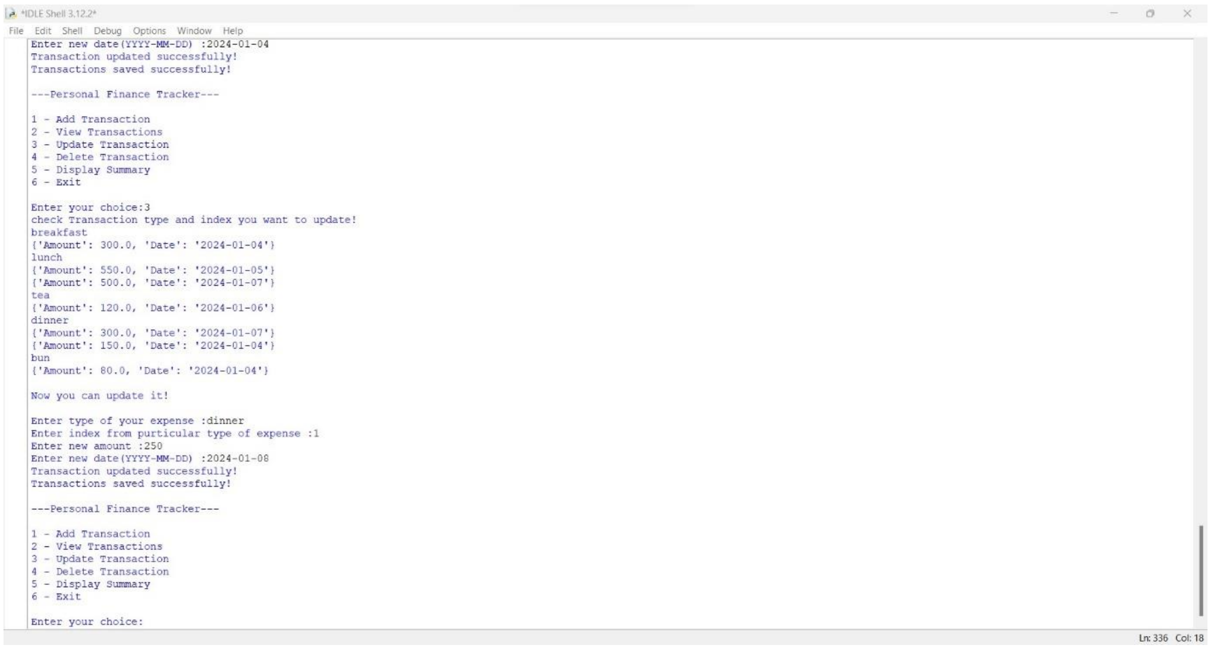


```
Transaction added successfully!
Transactions saved successfully!

---Personal Finance Tracker---

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

Enter your choice:dinner
invalid literal for int() with base 10: 'dinner'

---Personal Finance Tracker---

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

Enter your choice:1

Enter Transaction Details!

Enter the type of expense :dinner

Enter the amount :150

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

Enter your choice:
```

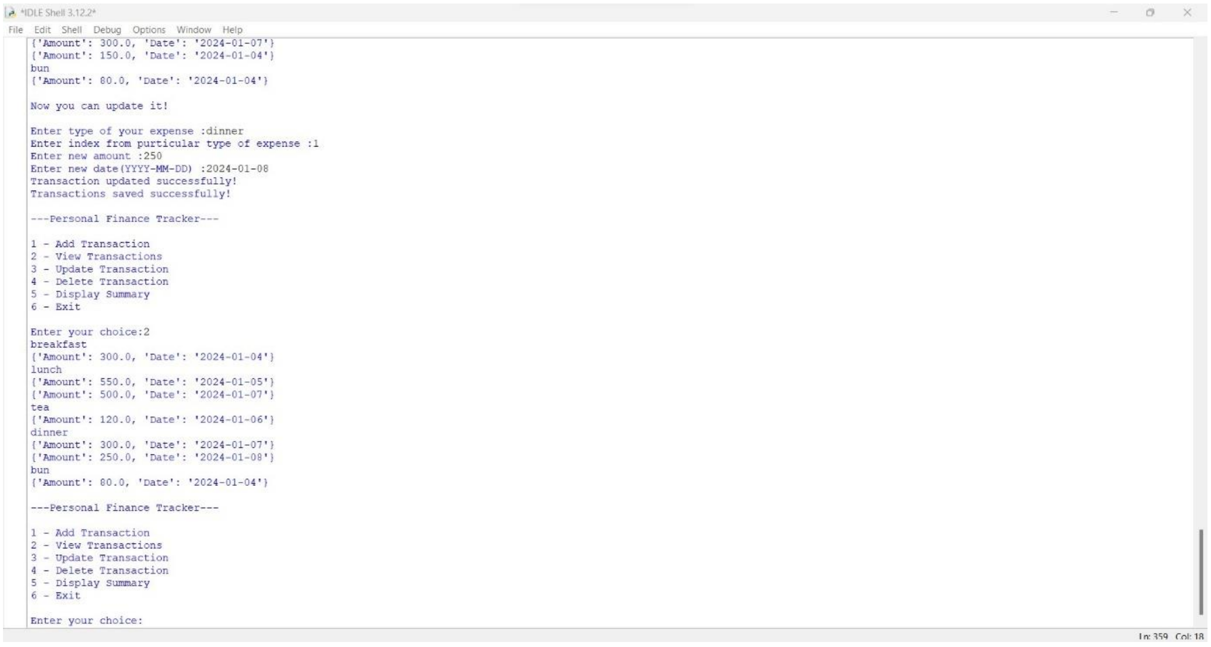


```
Enter your choice:1

Enter Transaction Details!

Enter the type of expense :dinner

Enter the amount :150

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

Enter your choice:2
breakfast
{'Amount': 300.0, 'Date': '2024-01-04'}
lunch
{'Amount': 550.0, 'Date': '2024-01-05'}
{'Amount': 500.0, 'Date': '2024-01-07'}
tea
{'Amount': 120.0, 'Date': '2024-01-06'}
dinner
{'Amount': 300.0, 'Date': '2024-01-07'}
{'Amount': 150.0, 'Date': '2024-01-04'}
bun
{'Amount': 70.0, 'Date': '2024-04-01'}

---Personal Finance Tracker---

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

Enter your choice:
```

TEST CASE NO 6

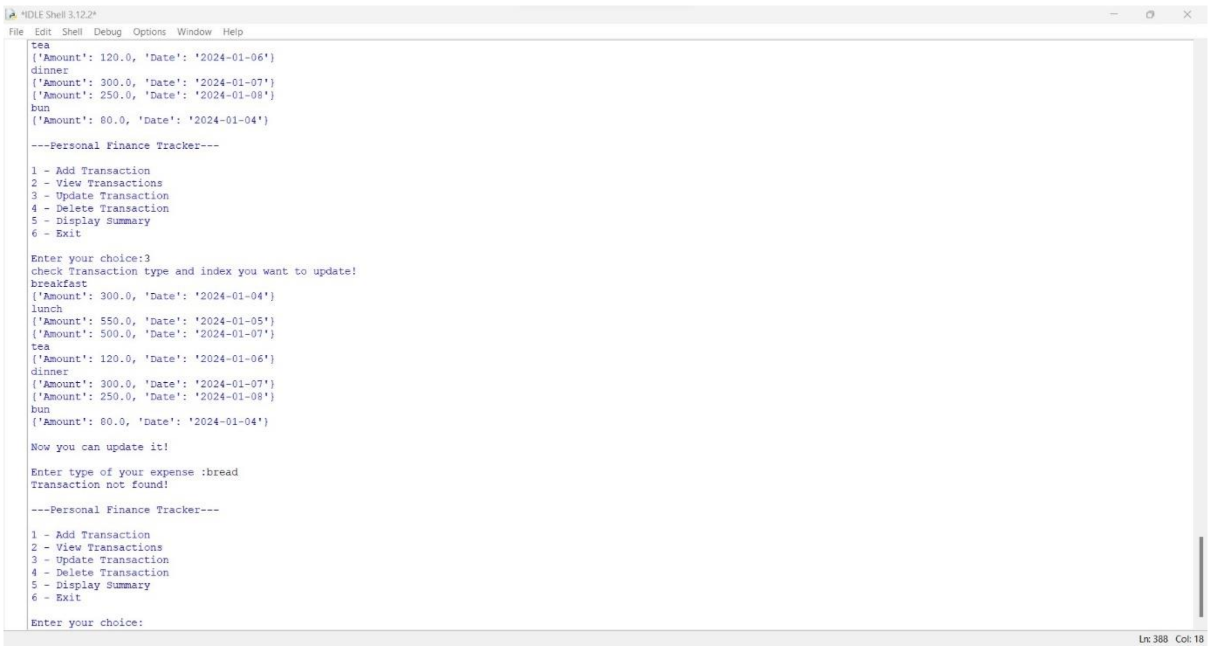


```
Enter new date(YYYY-MM-DD) :2024-01-04
Transaction updated successfully!
Transactions saved successfully!

---Personal Finance Tracker---

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

Enter your choice:3
check Transaction type and index you want to update!
breakfast
{'Amount': 300.0, 'Date': '2024-01-04'}
lunch
{'Amount': 550.0, 'Date': '2024-01-05'}
{'Amount': 500.0, 'Date': '2024-01-07'}
tea
{'Amount': 120.0, 'Date': '2024-01-06'}
dinner
{'Amount': 300.0, 'Date': '2024-01-07'}
{'Amount': 150.0, 'Date': '2024-01-04'}
bun
{'Amount': 80.0, 'Date': '2024-01-04'}

Now you can update it!

Enter type of your expense :dinner
Enter index from purticular type of expense :1
Enter new amount :250
Enter new date(YYYY-MM-DD) :2024-01-08
Transaction updated successfully!
Transactions saved successfully!

---Personal Finance Tracker---

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

Enter your choice:
```

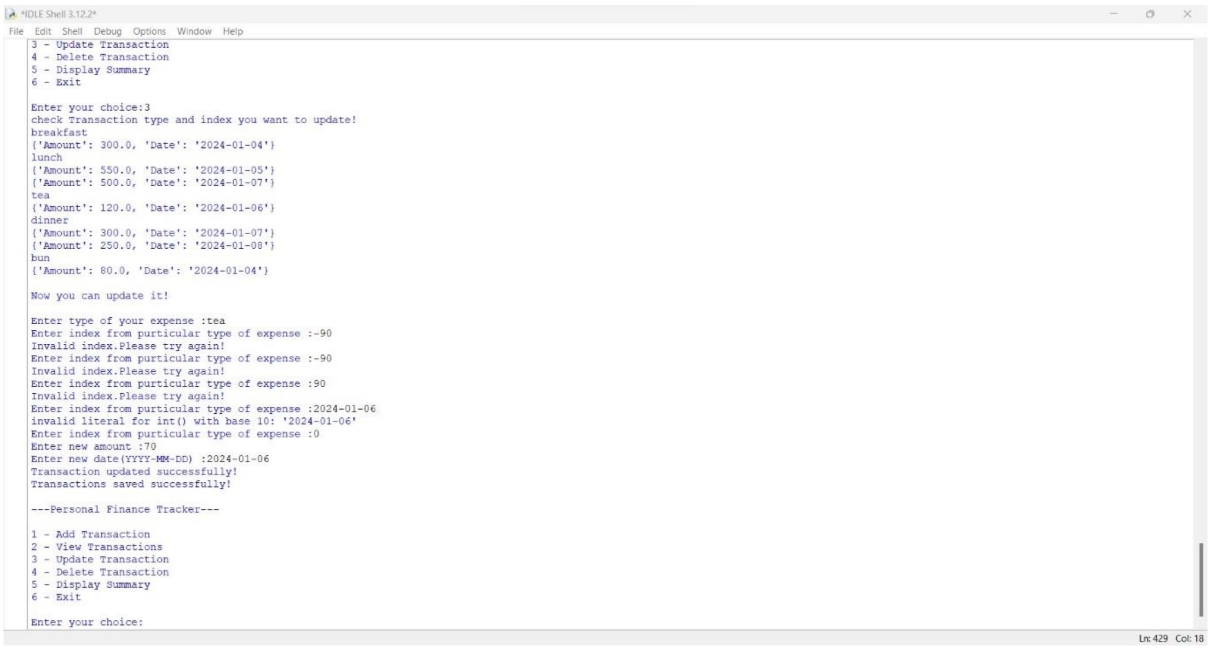


```
{'Amount': 300.0, 'Date': '2024-01-07'}
{'Amount': 150.0, 'Date': '2024-01-04'}
bun
{'Amount': 80.0, 'Date': '2024-01-04'}

Now you can update it!

Enter type of your expense :dinner
Enter index from purticular type of expense :1
Enter new amount :250
Enter new date(YYYY-MM-DD) :2024-01-08
Transaction updated successfully!
Transactions saved successfully!

---Personal Finance Tracker---

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

Enter your choice:2
breakfast
{'Amount': 300.0, 'Date': '2024-01-04'}
lunch
{'Amount': 550.0, 'Date': '2024-01-05'}
{'Amount': 500.0, 'Date': '2024-01-07'}
tea
{'Amount': 120.0, 'Date': '2024-01-06'}
dinner
{'Amount': 300.0, 'Date': '2024-01-07'}
{'Amount': 250.0, 'Date': '2024-01-08'}
bun
{'Amount': 80.0, 'Date': '2024-01-04'}

---Personal Finance Tracker---

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

Enter your choice:
```

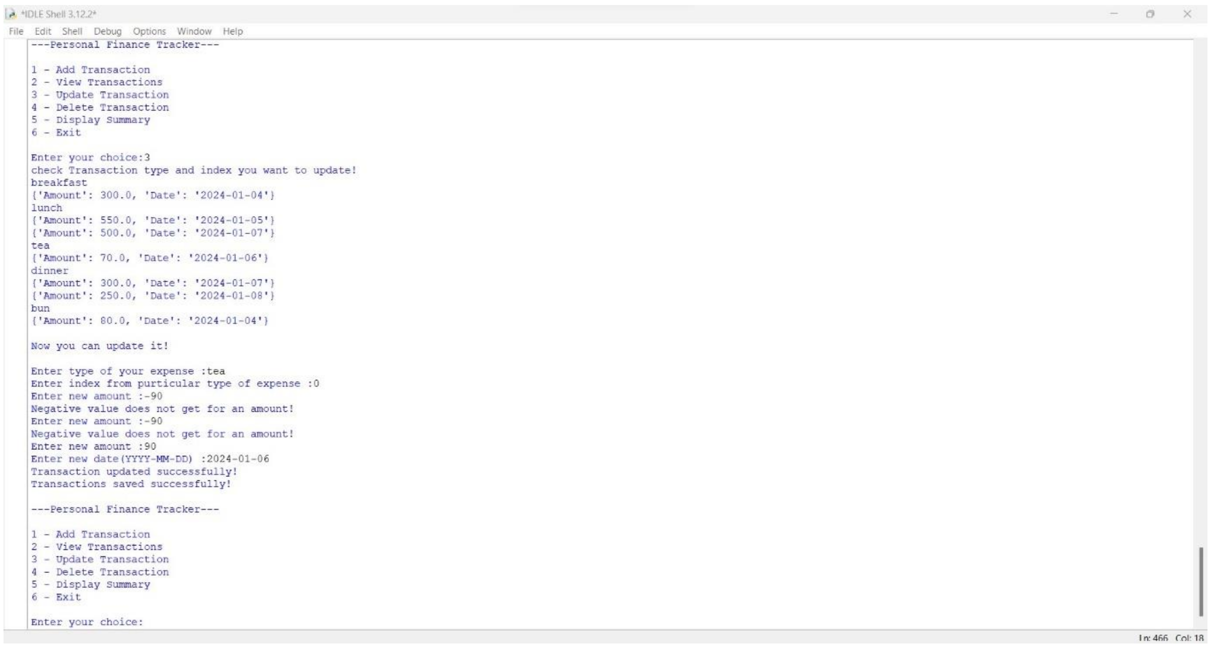
TEST CASE NO 7


```
tea
{'Amount': 120.0, 'Date': '2024-01-06'}
dinner
{'Amount': 300.0, 'Date': '2024-01-07'}
{'Amount': 250.0, 'Date': '2024-01-08'}
bun
{'Amount': 80.0, 'Date': '2024-01-04'}

---Personal Finance Tracker---

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

Enter your choice:3
check Transaction type and index you want to update!
breakfast
{'Amount': 300.0, 'Date': '2024-01-04'}
lunch
{'Amount': 550.0, 'Date': '2024-01-05'}
{'Amount': 500.0, 'Date': '2024-01-07'}
tea
{'Amount': 120.0, 'Date': '2024-01-06'}
dinner
{'Amount': 300.0, 'Date': '2024-01-07'}
{'Amount': 250.0, 'Date': '2024-01-08'}
bun
{'Amount': 80.0, 'Date': '2024-01-04'}

Now you can update it!

Enter type of your expense :bread
Transaction not found!

---Personal Finance Tracker---

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

Enter your choice:
```

TEST CASE NO 8

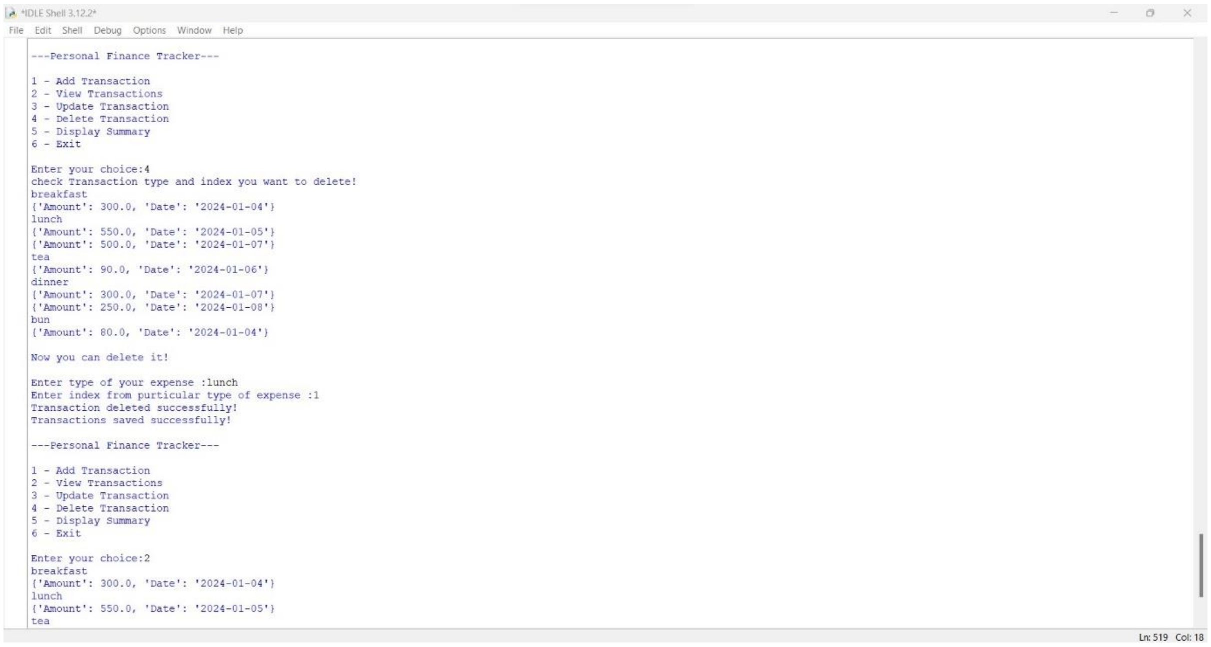

```
3 - Update Transaction
4 - Delete Transaction
5 - Display Summary
6 - Exit

Enter your choice:3
check Transaction type and index you want to update!
breakfast
{'Amount': 300.0, 'Date': '2024-01-04'}
lunch
{'Amount': 550.0, 'Date': '2024-01-05'}
{'Amount': 500.0, 'Date': '2024-01-07'}
tea
{'Amount': 120.0, 'Date': '2024-01-06'}
dinner
{'Amount': 300.0, 'Date': '2024-01-07'}
{'Amount': 250.0, 'Date': '2024-01-08'}
bun
{'Amount': 80.0, 'Date': '2024-01-04'}

Now you can update it!

Enter type of your expense :tea
Enter index from purticular type of expense :-90
Invalid index.Please try again!
Enter index from purticular type of expense :-90
Invalid index.Please try again!
Enter index from purticular type of expense :90
Invalid index.Please try again!
Enter index from purticular type of expense :2024-01-06
invalid literal for int() with base 10: '2024-01-06'
Enter index from purticular type of expense :0
Enter new amount :70
Enter new date(YYYY-MM-DD) :2024-01-06
Transaction updated successfully!
Transactions saved successfully!

---Personal Finance Tracker---

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

Enter your choice:
```

TEST CASE NO 9

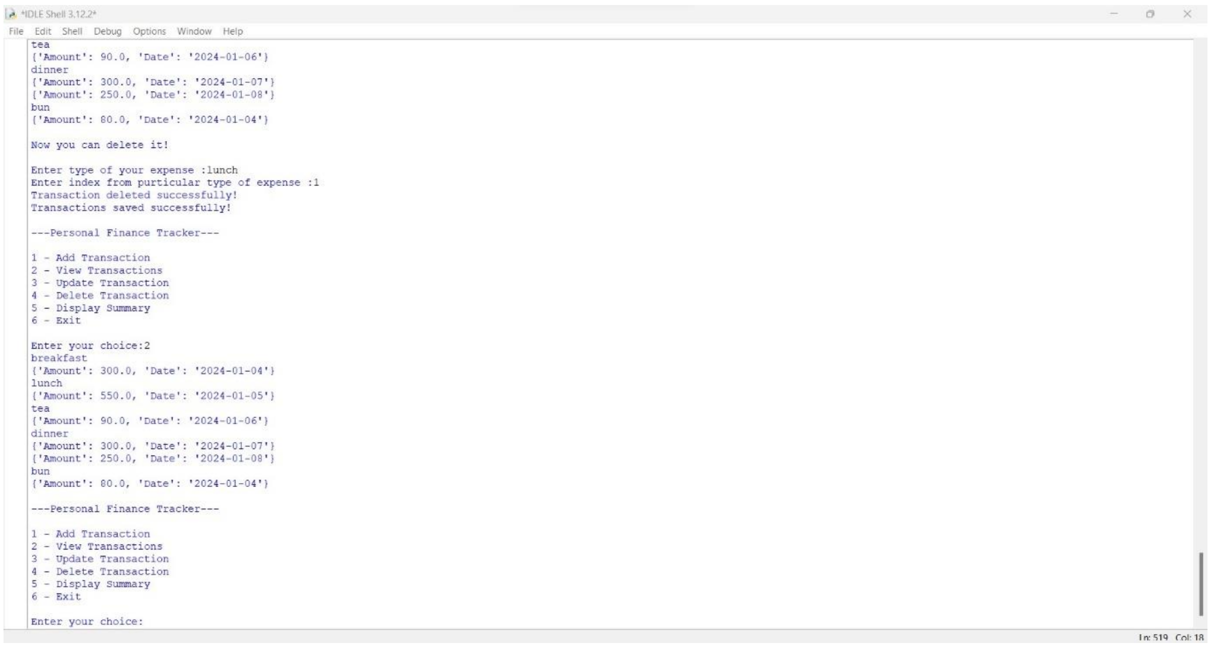


```
---Personal Finance Tracker---

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

Enter your choice:3
check Transaction type and index you want to update!
breakfast
{'Amount': 300.0, 'Date': '2024-01-04'}
lunch
{'Amount': 550.0, 'Date': '2024-01-05'}
{'Amount': 500.0, 'Date': '2024-01-07'}
tea
{'Amount': 70.0, 'Date': '2024-01-06'}
dinner
{'Amount': 300.0, 'Date': '2024-01-07'}
{'Amount': 250.0, 'Date': '2024-01-08'}
bun
{'Amount': 80.0, 'Date': '2024-01-04'}

Now you can update it!

Enter type of your expense :tea
Enter index from purticular type of expense :0
Enter new amount :-90
Negative value does not get for an amount!
Enter new amount :-90
Negative value does not get for an amount!
Enter new amount :90
Enter new date(YYYY-MM-DD) :2024-01-06
Transaction updated successfully!
Transactions saved successfully!

---Personal Finance Tracker---

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

Enter your choice:
```

TEST CASE NO 10

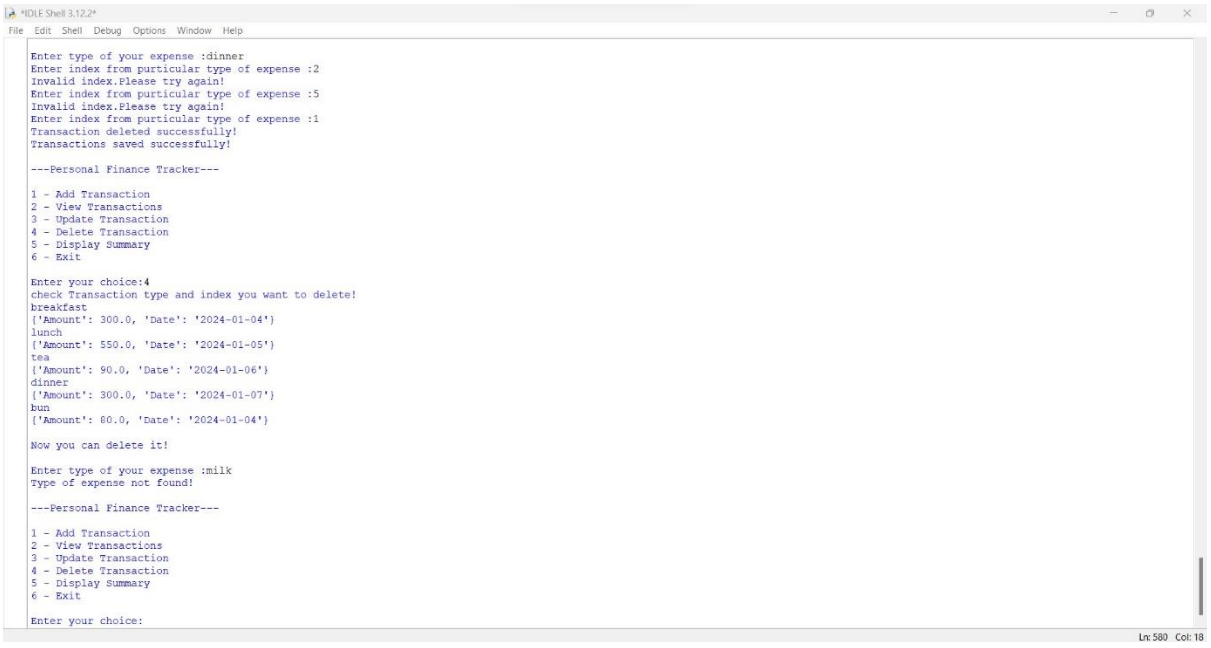


```
---Personal Finance Tracker---

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

Enter your choice:4
check Transaction type and index you want to delete!
breakfast
{'Amount': 300.0, 'Date': '2024-01-04'}
lunch
{'Amount': 550.0, 'Date': '2024-01-05'}
{'Amount': 500.0, 'Date': '2024-01-07'}
tea
{'Amount': 90.0, 'Date': '2024-01-06'}
dinner
{'Amount': 300.0, 'Date': '2024-01-07'}
{'Amount': 250.0, 'Date': '2024-01-08'}
bun
{'Amount': 80.0, 'Date': '2024-01-04'}

Now you can delete it!

Enter type of your expense :lunch
Enter index from purticular type of expense :1
Transaction deleted successfully!
Transactions saved successfully!

---Personal Finance Tracker---

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

Enter your choice:2
breakfast
{'Amount': 300.0, 'Date': '2024-01-04'}
lunch
{'Amount': 550.0, 'Date': '2024-01-05'}
tea
```



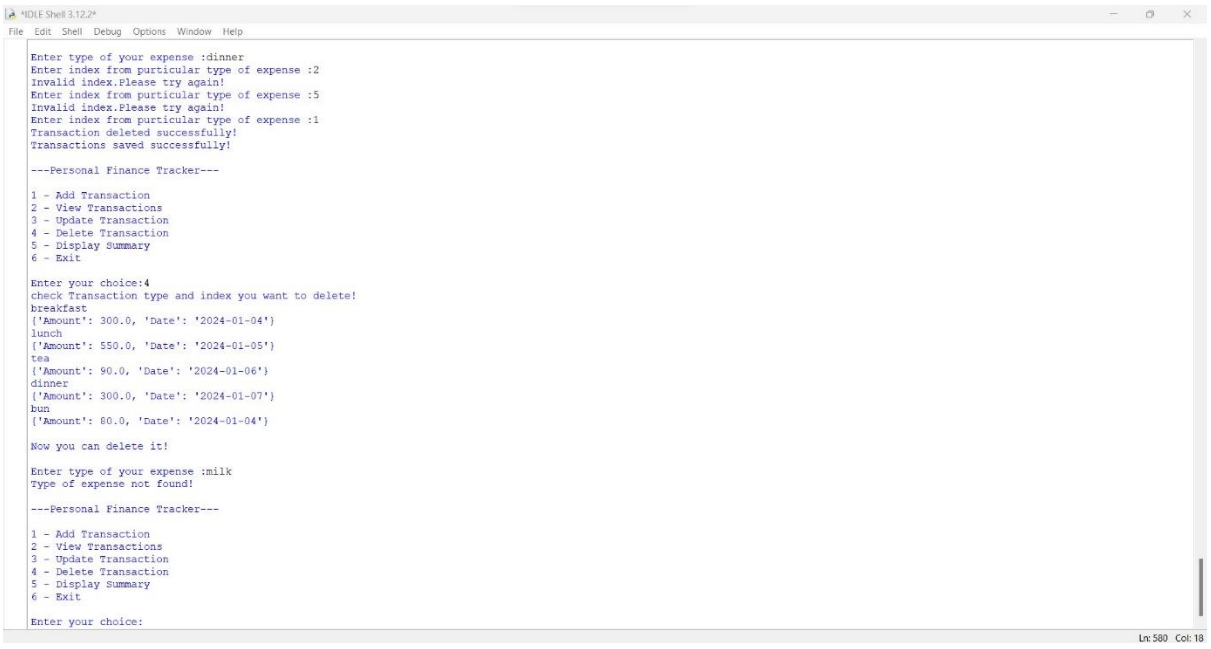
```
tea
{'Amount': 90.0, 'Date': '2024-01-06'}
dinner
{'Amount': 300.0, 'Date': '2024-01-07'}
{'Amount': 250.0, 'Date': '2024-01-08'}
bun
{'Amount': 80.0, 'Date': '2024-01-04'}

Now you can delete it!

Enter type of your expense :lunch
Enter index from purticular type of expense :1
Transaction deleted successfully!
Transactions saved successfully!

---Personal Finance Tracker---

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

Enter your choice:2
breakfast
{'Amount': 300.0, 'Date': '2024-01-04'}
lunch
{'Amount': 550.0, 'Date': '2024-01-05'}
tea
{'Amount': 90.0, 'Date': '2024-01-06'}
dinner
{'Amount': 300.0, 'Date': '2024-01-07'}
{'Amount': 250.0, 'Date': '2024-01-08'}
bun
{'Amount': 80.0, 'Date': '2024-01-04'}

---Personal Finance Tracker---

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

Enter your choice:
```

TEST CASE NO 11



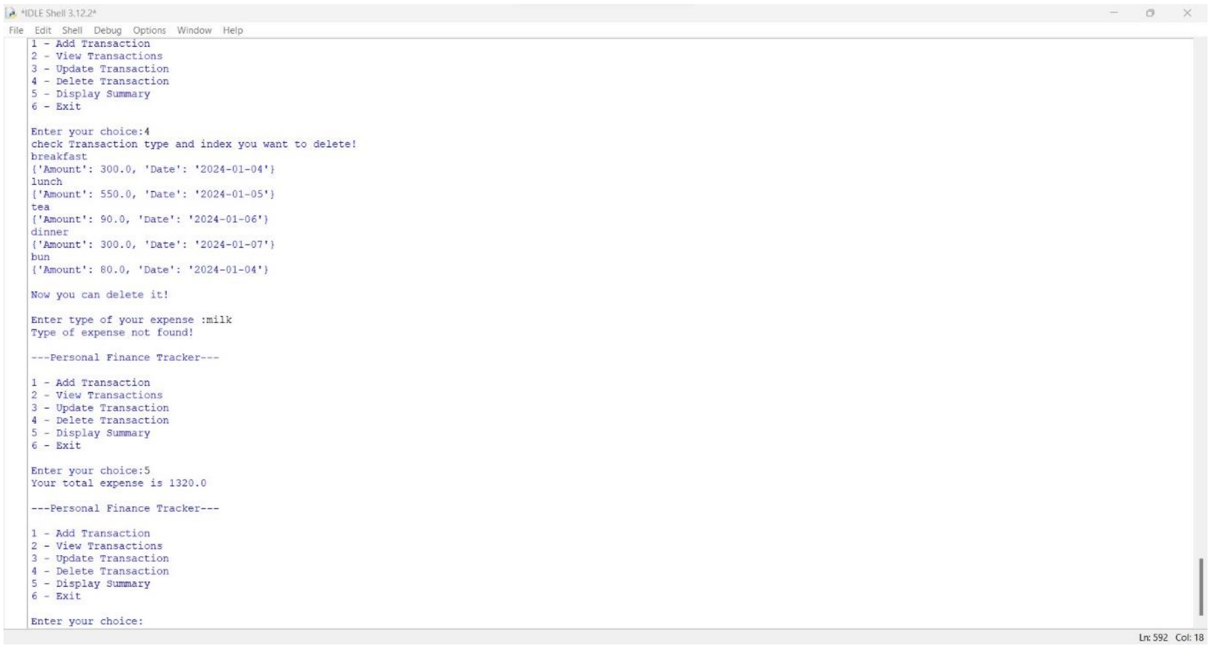
```
Enter type of your expense :dinner
Enter index from purticular type of expense :2
Invalid index.Please try again!
Enter index from purticular type of expense :5
Invalid index.Please try again!
Enter index from purticular type of expense :1
Transaction deleted successfully!
Transactions saved successfully!

---Personal Finance Tracker---

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

Enter your choice:4
check Transaction type and index you want to delete!
breakfast
{'Amount': 300.0, 'Date': '2024-01-04'}
lunch
{'Amount': 550.0, 'Date': '2024-01-05'}
tea
{'Amount': 90.0, 'Date': '2024-01-06'}
dinner
{'Amount': 300.0, 'Date': '2024-01-07'}
bun
{'Amount': 80.0, 'Date': '2024-01-04'}

Now you can delete it!

Enter type of your expense :milk
Type of expense not found!

---Personal Finance Tracker---

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

Enter your choice:
```

TEST CASE NO 12



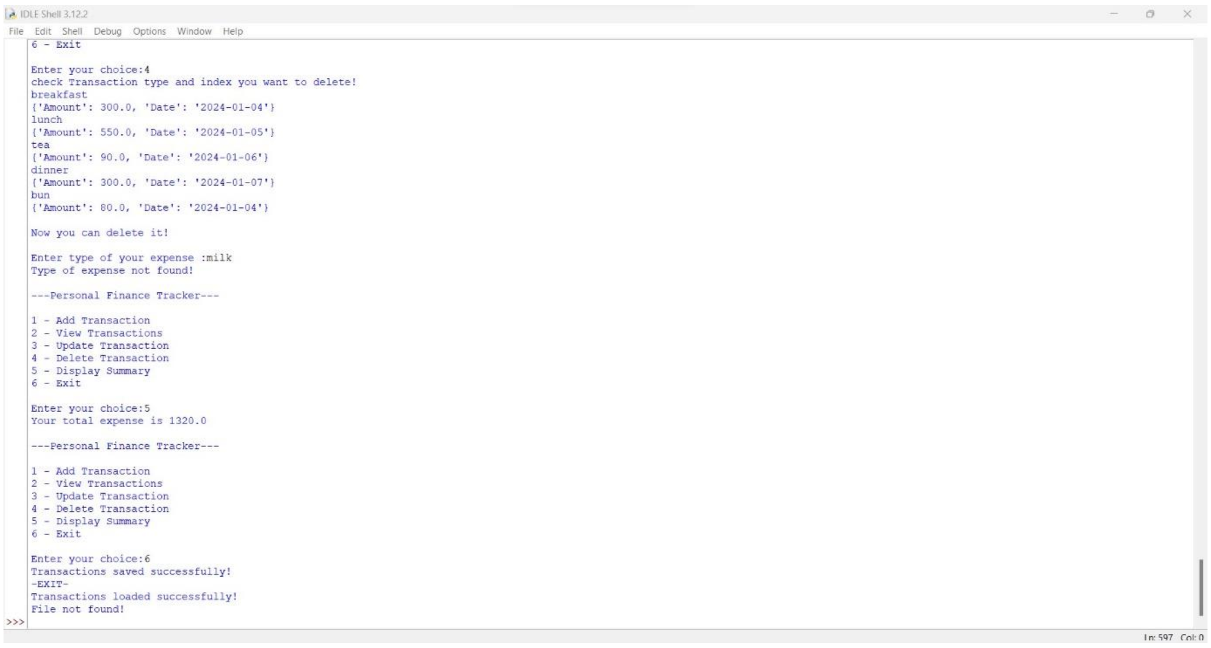
```
Enter type of your expense :dinner
Enter index from purticular type of expense :2
Invalid index.Please try again!
Enter index from purticular type of expense :5
Invalid index.Please try again!
Enter index from purticular type of expense :1
Transaction deleted successfully!
Transactions saved successfully!

---Personal Finance Tracker---

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

Enter your choice:4
check Transaction type and index you want to delete!
breakfast
{'Amount': 300.0, 'Date': '2024-01-04'}
lunch
{'Amount': 550.0, 'Date': '2024-01-05'}
tea
{'Amount': 90.0, 'Date': '2024-01-06'}
dinner
{'Amount': 300.0, 'Date': '2024-01-07'}
bun
{'Amount': 80.0, 'Date': '2024-01-04'}

Now you can delete it!

Enter type of your expense :milk
Type of expense not found!

---Personal Finance Tracker---

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

Enter your choice:
```

TEST CASE NO 13



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

Enter your choice:4
check Transaction type and index you want to delete!
breakfast
{'Amount': 300.0, 'Date': '2024-01-04'}
lunch
{'Amount': 550.0, 'Date': '2024-01-05'}
tea
{'Amount': 90.0, 'Date': '2024-01-06'}
dinner
{'Amount': 300.0, 'Date': '2024-01-07'}
bun
{'Amount': 80.0, 'Date': '2024-01-04'}

Now you can delete it!

Enter type of your expense :milk
Type of expense not found!

---Personal Finance Tracker---

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

Enter your choice:5
Your total expense is 1320.0

---Personal Finance Tracker---

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

Enter your choice:
```

TEST CASE NO 14



```
6 - Exit

Enter your choice:4
check Transaction type and index you want to delete!
breakfast
{'Amount': 300.0, 'Date': '2024-01-04'}
lunch
{'Amount': 550.0, 'Date': '2024-01-05'}
tea
{'Amount': 90.0, 'Date': '2024-01-06'}
dinner
{'Amount': 300.0, 'Date': '2024-01-07'}
bun
{'Amount': 80.0, 'Date': '2024-01-04'}

Now you can delete it!

Enter type of your expense :milk
Type of expense not found!

---Personal Finance Tracker---

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

Enter your choice:5
Your total expense is 1320.0

---Personal Finance Tracker---

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

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