A white object with a black background

Description automatically generated

Course :Bsc(Hons) Computer Science

Module : Software Development 01

Module Leader : Mr. Guhanathan P

Assignment Type : Individual

Issue Date : 06th March 2024

Hand – in – Date : 18 th March 2024

Deadline : on or before 1.00 PM

Student Id : 20230469

Email : Disandu.20230469@iit.ac.lk

[**Programme Design** 3](#_Toc161631208)

[**Algorithm** 3](#_Toc161631209)

[**Data Structure Design** 5](#_Toc161631210)

[**Pesudocode Design** 5](#_Toc161631211)

[Function load\_transactions() 5](#_Toc161631212)

[Function save\_transactions() 5](#_Toc161631213)

[Function add\_transactions() 6](#_Toc161631214)

[Function View\_transactions() 6](#_Toc161631215)

[Function update\_transactions() 7](#_Toc161631216)

[Function Delete\_tranasactions() 8](#_Toc161631217)

[Function Display\_Summary() 8](#_Toc161631218)

[Function Main() 9](#_Toc161631219)

[**Test cases** 10](#_Toc161631220)

[1.Main Function 10](#_Toc161631221)

[1.1 Screen Shots For Test Cases 10](#_Toc161631222)

[2. Add Function 12](#_Toc161631223)

[2.1 Screen Shots Of Test Cases 12](#_Toc161631224)

[3.View Transactions 14](#_Toc161631225)

[3.1 Screenshots Of Test Cases 14](#_Toc161631226)

[4. Update Transactions 15](#_Toc161631227)

[4.1 Screen Shots of test Cases 15](#_Toc161631228)

[5. Delete Transactions 17](#_Toc161631229)

[5.1 ScreenShots Of test cases 17](#_Toc161631230)

[**Error Report** 19](#_Toc161631231)

[**Resolved Error Report** 20](#_Toc161631232)

[1.Displaying an error when opening empty json file 20](#_Toc161631233)

[2. When Transactions List Is Empty 20](#_Toc161631234)

[3. Entering Negative values & zero for the amount 21](#_Toc161631235)

[4. User Enter “Income”or “Expense” with spelling Mistakes 22](#_Toc161631236)

# **Programme Design**

# **Algorithm**

1.Start

2.Initialize transactions as an Empty List

3.define load\_transaction

3.1open json file in read Mode

4.define save\_transaction

4.1 Asking Input from the user

4.2 Open json file in Write Mode

4.3 Check User enters Yes or No

4.4 Store Data into file according to users input

5. define add\_transaction

5.1Ask from user to enter amount,description,type,& date of transaction

5.2append data entered by user to list

5.3 call save\_transaction

6.define view\_transaction

6.1 Call load\_transaction

6.2 display the transactions saved in the list

7.define update\_transaction

7.1 Load view\_transaction

7.2 initialize num=None

7.3 ask user to enter the transaction Number

7.4 Ask user what want to update

7.5 Ask user to enter the new value

7.6 call the function to save\_transaction

8. Define delete\_transaction

8.1 Call delete

8.2initialize num=None

8.3 Ask the transaction number from user

8.4 Select the correspond transaction from list then delete it

8.5 call the function save\_transaction

9. Define display\_summary

9.1initialize tot\_income=0

9.2 initialize tot\_expense=0

9.3 Iterates through list to find “income”&”expense”

9.4 disaplay the total income

9.5 display total expenses

10.define main\_menu

10.1 display 1 as add transaction

10.2 display 2 as view transaction

10.3 display 3 as update transaction

10.4 display 4 as delete transactiom

10.5 display 5 as display summary

10.6 display 6 as exit

10.7 if user enters 1 load add\_transaction function

10.8 if user enters 2 load view transaction function

10.9 if user enters 3 load update transaction function

10.10 if user enters 4 losd delete transaction function

10.11if user enter 5 load display\_summary function

10.12if user enters 6 exit from the programme

11.End

# **Data Structure Design**

transactions=[ ]

//Every transaction saved as a list into the main transactions list

//[amount,type,description,date]

Ex:-transactions[

[1500,”expense” ,”fee”,”12/03/2023]

[4500,”Income”,”salary”,15/03/2023] ]

# **Pesudocode Design**

Start

Import json

Transactions=[]

## Function load\_transactions()

Set file\_path=data.json

Open file\_path in readmode as json\_file

Set transaction=Json.parse(json\_file)

End Function

## Function save\_transactions()

Prompt user Do you want to save(Yes/No)?

If user enters “Yes” then

Set file\_path=data.json

Open file path in write mode as json\_file

JSON,parse(transactions,json\_file)

Else If user enters “No” then

Display Transaction details are not saved

Else

Disaplay Invalid Option Enter a valid Option

End If

End Function

## Function add\_transactions()

Prompt user to enter the transaction Amount

While True Do,

Prompt user to enter the transaction Type(Income/Expense)

If users input is not “income”or “expense”

Display invalid Input please Input “Income”or “Expense”

Else:

Prompt user to enter the transaction purpose

Prompt user to enter the transaction Date

Append users inputs to transactions list

Call save Transaction to save the Transactions

Break The loop

End While

End Function

## Function View\_transactions()

Calling the load transaction to the transactions to load transactions

If transactions list empty

Display Sorry there aren’t Any Transactions

Else

For each transaction in transactions list with index start From 1

Display Transaction Details

End If

End Function

## Function update\_transactions()

Call load\_transaction to load transactions

Call view\_transactions to view exsisting list of transations

If transaction list is empty

Display the transaction list is empty

Else

Prompt user to the enter the transaction number to update

If transaction number is Invalid

Display Invalid transaction number

Return

Prompt user to What you want to Update(Amount/Type/Description/Date)

If user enters amount

Prompt user to enter the new amount

Update it in the list

Else If user enters Type

Prompt user to enter the new type

Update it in list

Else If user enters the Description

Prompt user to Enter the new date

Update it in list

Else If user Enters the Date

Prompt User to enter the new date

Update it in the list

Else

Display Invalid Field

Return

End If

Load save\_transactions to save the Updates

End Function

## Function Delete\_tranasactions()

Load Load\_transactions to load exsisiting transactions

Load view\_transactions to print transaction list

If transaction number is Invalid

Display Invalid transaction number

Return

Prompt user to enter the number of the transaction to delete

Del the list of transaction which user enter from Transactions List

Load save\_transactions to prompt user to transaction is deleted

End Function

## Function Display\_Summary()

Calling load\_transactions to load exsisting Transactions

Initializing tot\_income=0

Intializing tot\_expense=0

Intializing Net\_balance=0

If transaction list is empty

Display there isn’t any transactions to Display

Else

For each transaction in transactions list

If transaction type is”Income”

tot\_income=tot\_income+amount in the transaction in transactions List

Else

tot\_expense=tot+expense+amount in the transaction in transactions List

End If

End For

End If

Net\_balance=tot\_income-tot\_expenses

Display Net\_balance,tot\_expense,tot\_income

End Function

## Function Main()

Call load\_transactions to load transactions from the file

While True Do

Display Personal Finance Tracker

Display 1. Add Transaction

Display 2. View Transactions

Display 3. Update Transactions

Display 4. Delete Transactions

Display 5. Display Summary

Display 6.Exit

Prompt user to choose one out of Following

End While

If users Choice is’ 1’

Call add\_transactions

Else If Users Choice is’2’

Call view\_transactions

Else If Users Choice is’3’

Call Update\_transactions

Else If users Choice is’4’

Call delete\_Transactions

Else If users choice is ‘5’

Call display\_summary

Else If users Choice is’6’

Display Exiting the programme…

Break

Else

Display Invalid Choice Try Again

End If

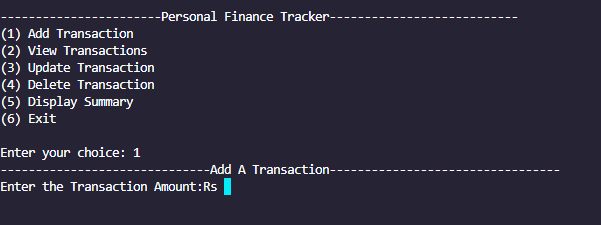
End Function

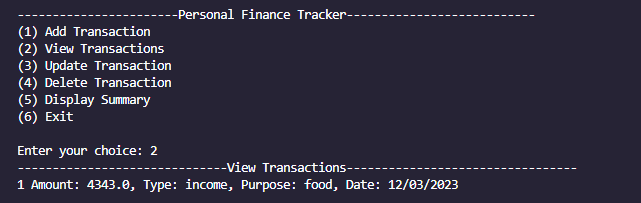
# **Test cases**

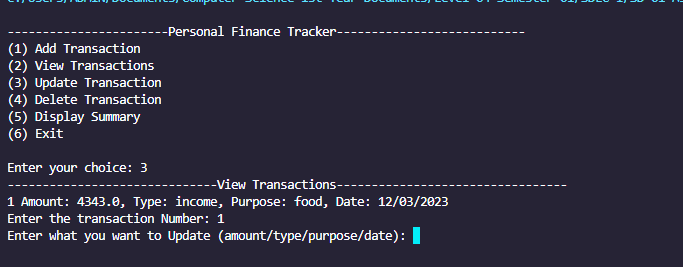
## Main Function

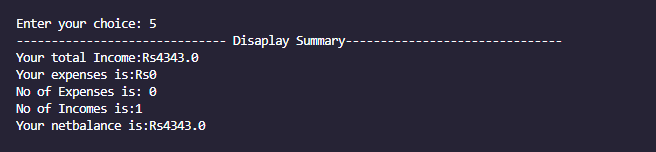
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test case** | **Input** | **Expected Output** | **Actual Output** | **Remarks** |
| 01 | Enter choice option 1 | Run the add\_transaction function | Run the add\_transaction function | Pass |
| 02 | Enter choice option 2 | Runt the view\_transactions  function | Runt the view\_transactions  function | Pass |
| 03 | Enter choice option 3 | Run the update transaction Function | Run the update transaction Function | Pass |
| 04 | Enter choice option 4 | Run the delete transaction function | Run the delete transaction function | Pass |
| 05 | Enter choice option 5 | Run the display summary Function | Run the display summary Function | pass |
| 06 | Enter choice option 6 | Exit from the programme | Exit from the programme | Pass |
| 07 | Enter a choice without 1 to 6 | Printing Error message invalid choice | Printing Error message invalid choice | Pass |

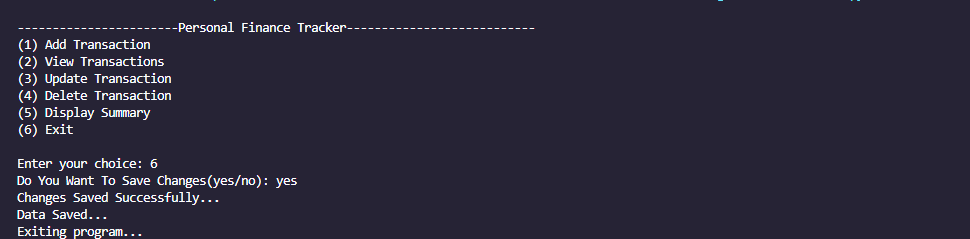
### 1.1 Screen Shots For Test Cases

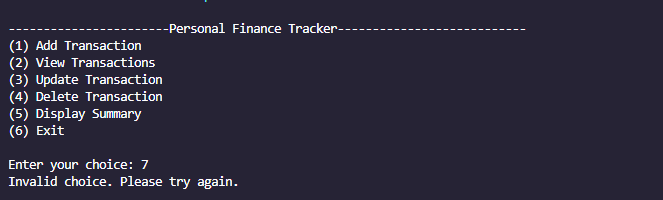








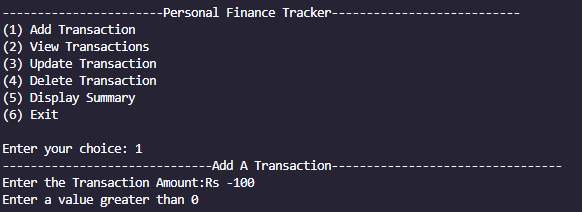


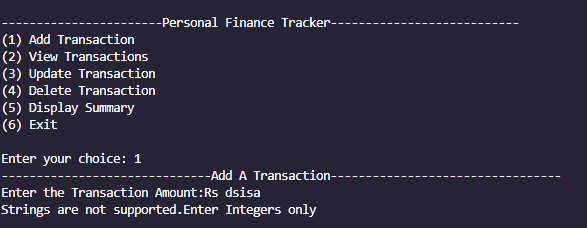


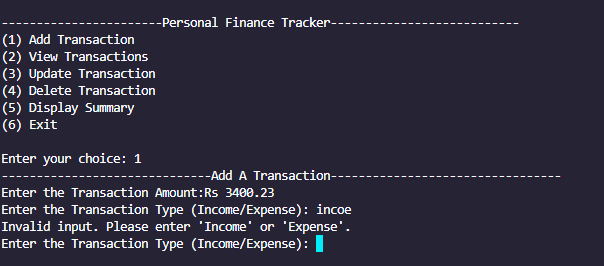
## Add Function

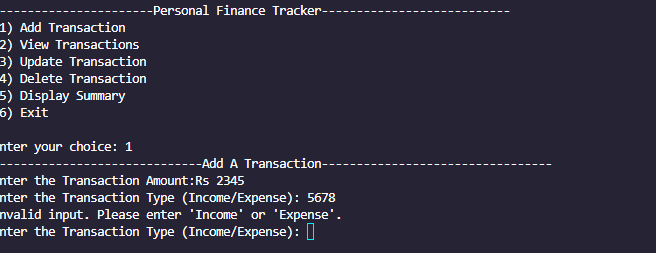
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test case** | **Input** | **Expected Output** | **Actual Output** | **Remarks** |
| 01 | Entering Negative Amount | Printing Error message input must greater than 0 | Printing Error message input must greater than 0 | Pass |
| 02 | Entering String value to the amount | Printing a string error message string values are not supported | Printing a string error message string values are not supported | Pass |
| 03 | Entering transaction type except “input” and “expense” | Printing error message “Income”&”Expense” are only allowed | Printing error message “Income”&”Expense” are only allowed | Pass |
| 04 | Entering a number to transaction type | Printing error message “Income”&”Expense” are only allowed | Printing error message “Income”&”Expense” are only allowed | Pass |

## 2.1 Screen Shots Of Test Cases





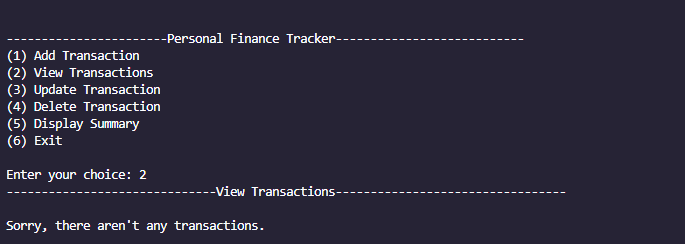




## View Transactions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test case** | **Input** | **Expected Output** | **Actual Output** | **Remarks** |
| 01 | Calling view transaction function without having any data in the data file | Error message which prints there is’t any transactions to view | Error message which prints there is’t any transactions to view | Pass |

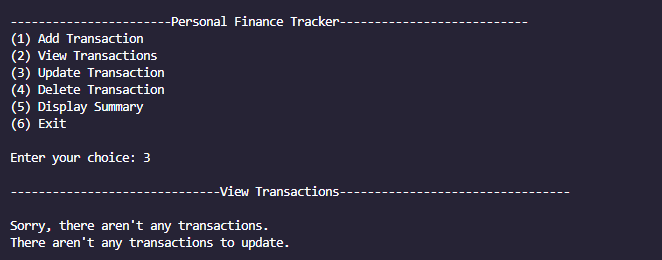
### Screenshots Of Test Cases

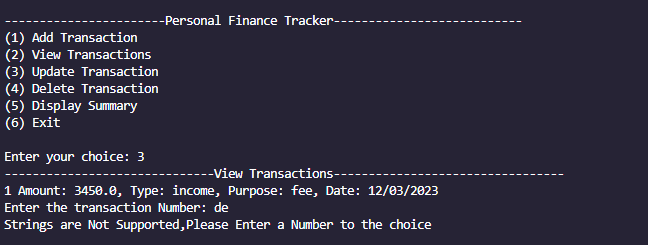


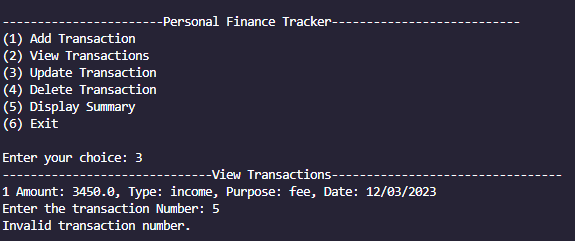
## Update Transactions

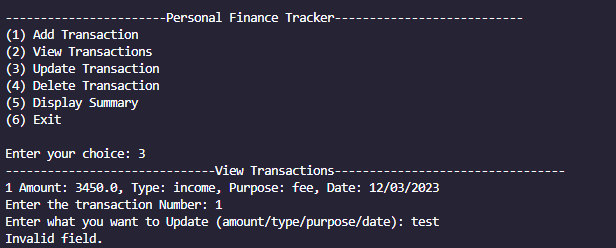
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test case** | **Input** | **Expected Output** | **Actual Output** | **Remarks** |
| 01 | Calling view transaction function without having any data in the data file | Error message which prints there isn’t any transactions to view | Error message which prints there isn’t any transactions to view | Pass |
| 02. | Entering a string value to the transaction Number | Print Strings are not supported please enter a Number | Print Strings are not supported please enter a Number | Pass |
| 03. | Inserting a transaction number not exsisting | Print Invalid Transaction Number | Print Invalid Transaction  Number | Pass |
| 04. | Entering a field except amount,type,purpose,date | Print error message Invalid field entered | Print error message Invalid field entered | Pass |

### 4.1 Screen Shots of test Cases





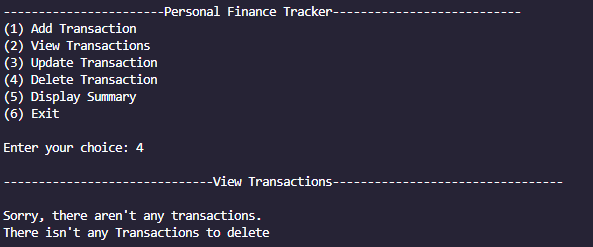


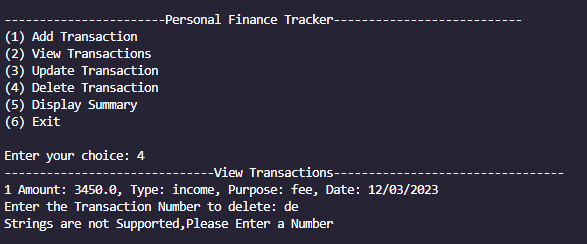


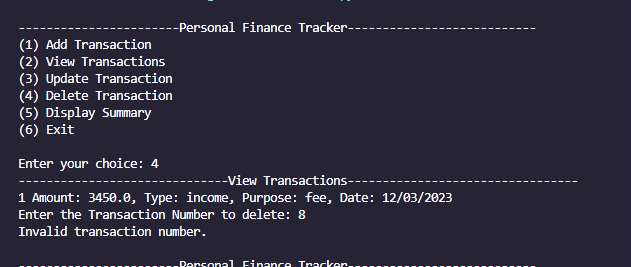
## Delete Transactions

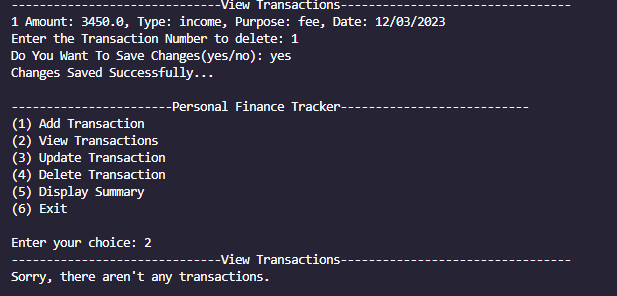
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test case** | **Input** | **Expected Output** | **Actual Output** | **Remarks** |
| 01 | Calling view transaction function without having any data in the data file | Error message which prints there is’t any transactions to view | Error message which prints there isn’t any transactions to view | Pass |
| 02. | Entering a string value to the transaction Number | Print Strings are not supported please enter a Number | Print Strings are not supported please enter a Number | Pass |
| 03. | Inserting a transaction number not exsisting | Print Invalid Transaction Number | Print Invalid Transaction  Number | Pass |
| 04 | Find the correct transaction corresponds what want to delete | Remove the entered transaction from corresponding list | Remove the entered transaction from corresponding list | Pass |

### 5.1 ScreenShots Of test cases









# **Error Report**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test case** | **Input** | **Expected Output** | **Actual Output** | **Remarks** |
| 01 | Open a file with programme empty json file | Programme  Runs Smoothly | Decode Error | Fail |
| 02. | Enter a choice Not exsisting | Display Invalid choice enter a correct choice | Displaying  Invalid Choice | Pass |
| 03. | Choose view transactions when there isn’t any transactions in the json file | Display Error Message | Programme returns to the start | Fail |
| 04. | User enters there input from mix cases | Run programme without errors | Run programme  without any errors | Pass |
| 05. | Entering Strings for Transaction amount | Displays an error message | Displays an  Error message | pass |
| 06. | Entering negative values and 0 for the amount | Printing Error Message | Saving negative values and zero in the list | Fail |
| 07 | User enters income or expense with spelling mistakes | Printing an Error message | Saving spelling mistaken word in the list | Fail |
| 08. | Users enters a wrong transaction number to Update or delete Which not exsisting | Print error as Invalid transaction number | Print error as invalid transaction Number | Pass |

# **Resolved Error Report**

## 1.Displaying an error when opening empty json file

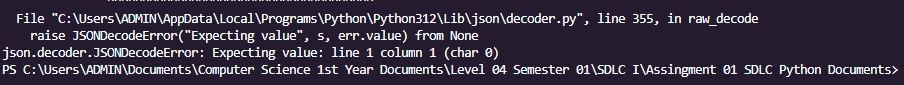
1. Description

When we open the programme first time with empty Json file programme disaplays a JSONdecode error

1. Steps to reproduce

Using error handling to handle the JSONDecodeError

1. Screen Shot of Error



1. Code Snippet to resolve the Error

*#Decode json file if there isn't any data in the file when programme intilialy start*

    except json.decoder.JSONDecodeError:

        print("JSONDecodeError: The file does not contain valid JSON content.")

        transactions = []*#if the file not found initalize transactions list as empty*

## 2. When Transactions List Is Empty

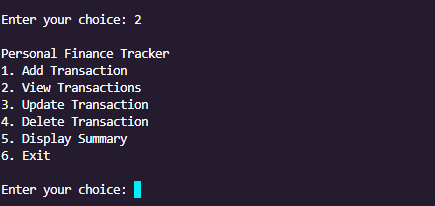
1. Description

When we call the view transaction when empty list is exsisting programme returns to the start

1. Step to Reproduce

Use if not check is there any transactions in the list

1. Screen Shot of the error



1. Code Snippet to resolve the error

*#check is there any transactions exsisting*

if not transactions:

      print("Sorry, there aren't any transactions.")

else:

*#iterate through transactions list from starting 1*

   for index, trans in enumerate(transactions, *start*=1):

   print(f"{index} Amount: {trans[0]}, Type: {trans[1]}, Purpose: {trans[2]}, Date: {trans[3]}")

## 3. Entering Negative values & zero for the amount

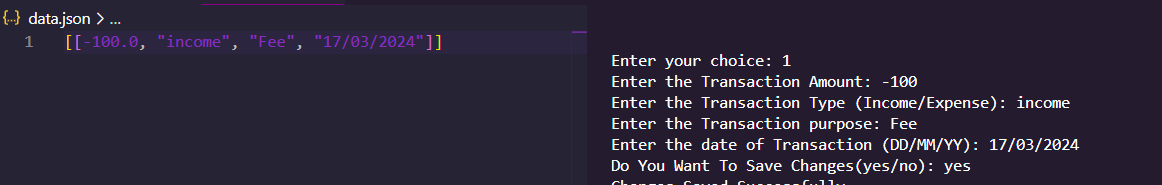
1. Description

If user enters zero or negative value for the amount it stored in the list

1. Step to Reproduce

Using if statement to check users input is lesser than or equals to zero

1. Screen Shot of the error



1. Code to resolve the error

if amount>0:

   while True:

*#prompt user for type of transactions*

   trs\_type = input("Enter the Transaction Type(Income/Expense):").lower()

  if trs\_type not in ("income", "expense"):

      print("Invalid input. Please enter 'Income' or'Expense'.")

   else:

*#prompt user for transaction purpose*

       purpose = input("Enter the Transaction purpose: ")

*#prompt user for data of transaction*

       date =input("Enter the date of Transaction (DD/MM/YY):")

*#append user enterd data in the main transactions list*

        transactions.append([amount, trs\_type, purpose, date])

*#calling save transaction function to save the transaction details user enterd*

        save\_transactions()

        break

   else:

      print("Enter a value greater than 0")

## 4. User Enter “Income”or “Expense” with spelling Mistakes

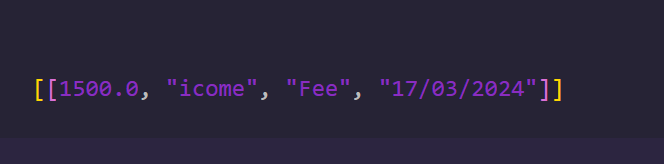
1. Description

If user Enters words income and expense with spelling mistakes it stored in the list without displaying error message

1. Step to reproduce

Check using if condition if the user enters “income” or “expense”

1. Screen Shot of the error



1. Code to Resolve the error

*#prompt user for tpe of transactions*

trs\_type = input("Enter the Transaction Type(Income/Expense): ").lower()

if trs\_type not in ("income", "expense"):

    print("Invalid input. Please enter 'Income' or'Expense'.")

 else: