Python programming internship

Project name

EXPENSE TACKER

Name: sandhya pal

Introduction

As we embark on Week 3 of your internship program, I am excited to introduce the project that will further enhance your skills in Python Programming. The upcoming task is the creation of a "Expense Tracker" as part of your Python Programming internship.

The Expense Tracker project is designed to reinforce your understanding of Python programming concepts and enhance your skills in building practical applications. In this project, you will be developing an Expense Tracker application that allows users to manage their expenses efficiently. This real-world application will involve handling data, user input, and implementing key functionalities.

OBJECTIVES

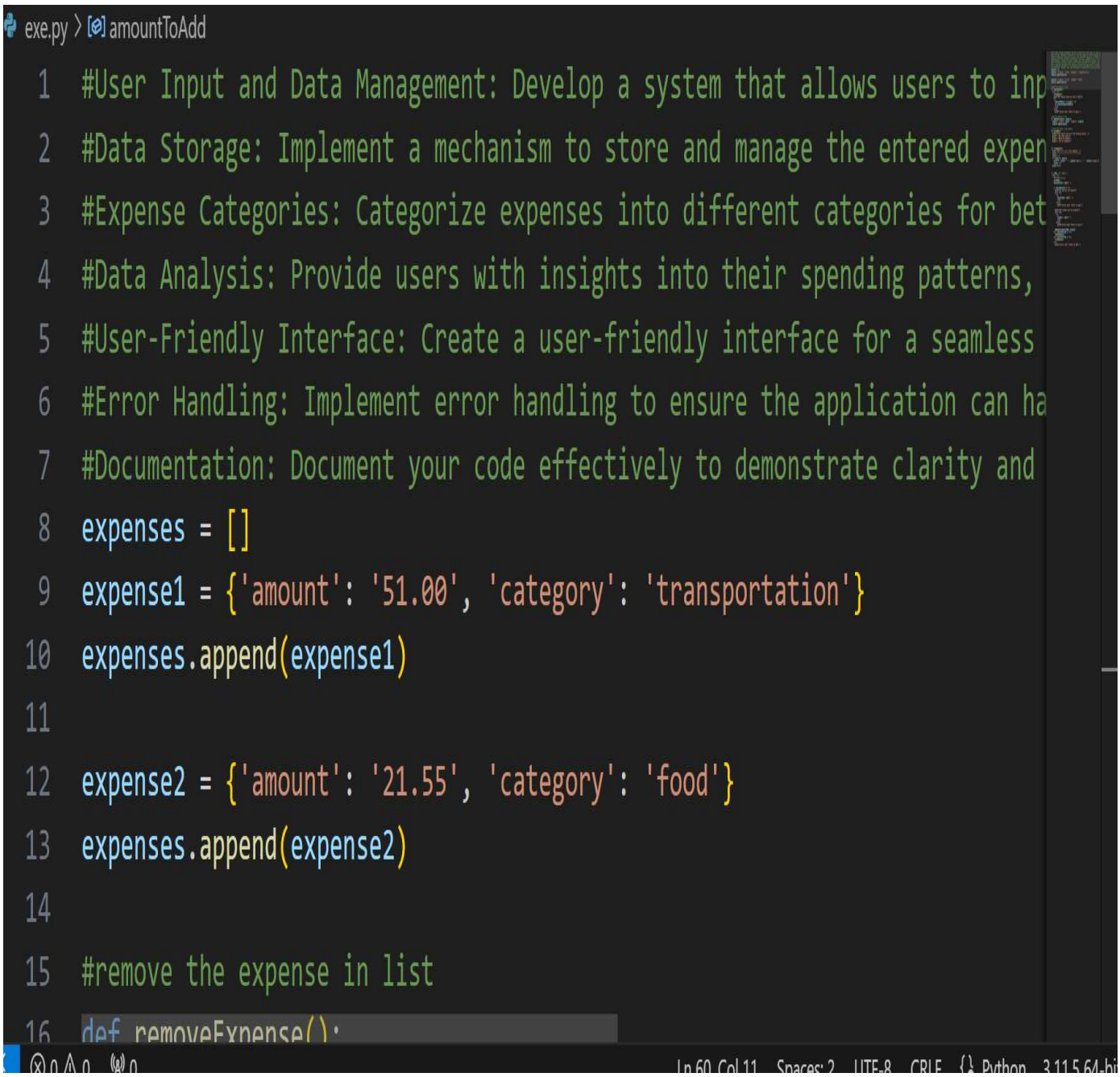
To accurately outline the scope of work required for a project, it is crucial to first identify its objectives. Pinpointing what the project hopes to accomplish will assist in determining its inclusions and limitations.

- **User Input and Data Management:** Develop a system that allows users to input their daily expenses.
- **Data Storage:** Implement a mechanism to store and manage the entered expense data.
- **Expense Categories:** Categorize expenses into different categories for better organization.
- **Data Analysis:** Provide users with insights into their spending patterns, such as monthly summaries and category-wise expenditure.
- **User-Friendly Interface:** Create a user-friendly interface for a seamless user experience.
- **Error Handling:** Implement error handling to ensure the application can handle unexpected inputs gracefully.
- **Documentation:** Document your code effectively to demonstrate clarity and understanding.

Requirements and features

- **User Input:** Allow users to input their daily expenses, including the amount spent and a brief description.
- **Data Storage:** Use appropriate data structures or file handling techniques to store and retrieve expense data.
- **Expense Categories:** Implement the ability for users to categorize their expenses (e.g., food, transportation, entertainment).
- **Data Analysis:** Provide users with the option to view summaries of their monthly expenses and category-wise expenditure.
- **User Interface:** Create a simple and intuitive user interface to interact with the Expense Tracker.
- **Error Handling:** Include error handling mechanisms to address potential issues during user interaction.
- **Documentation:** Provide clear documentation for your code, explaining the logic behind key functions and overall program structure.

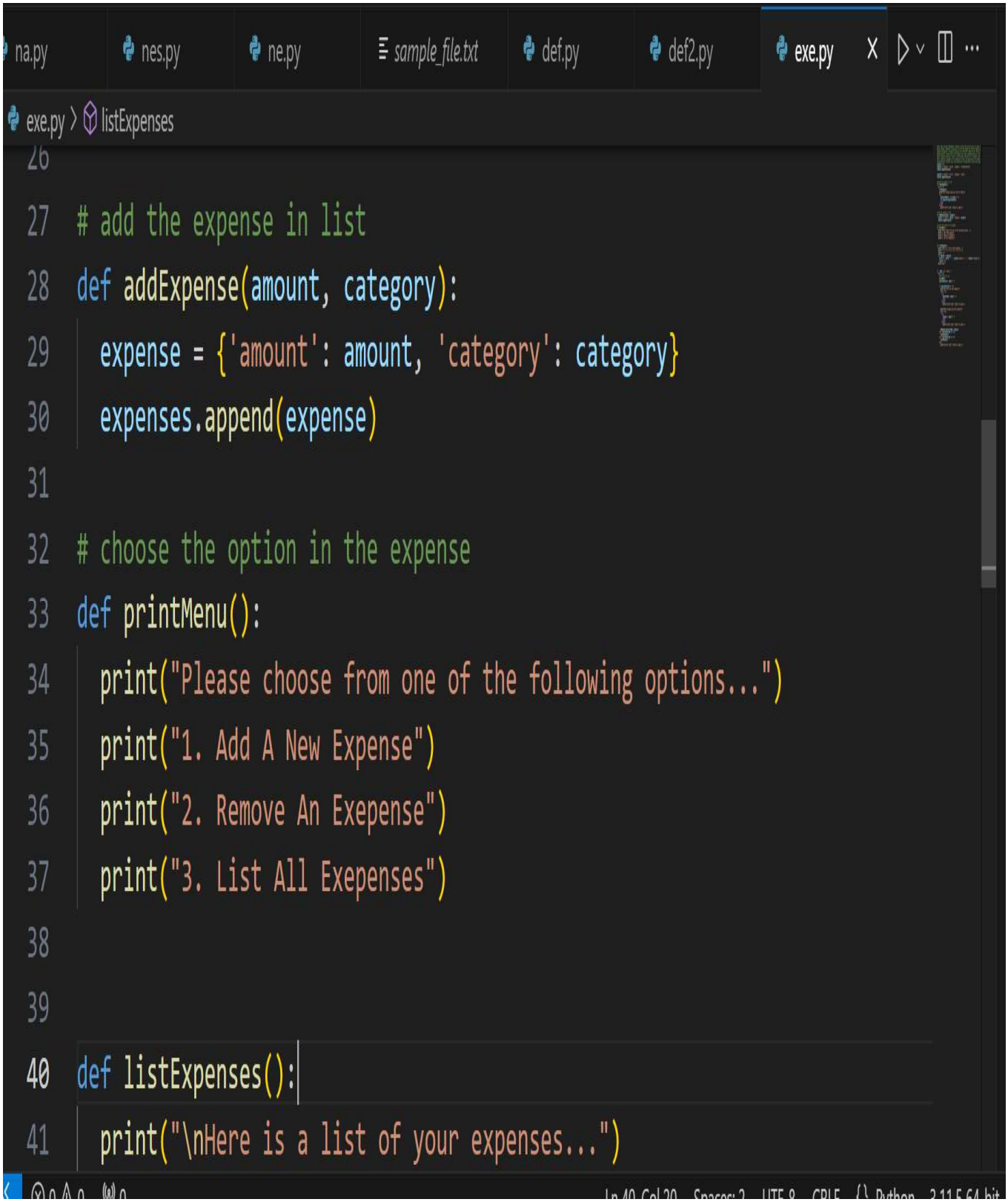
CODEING:

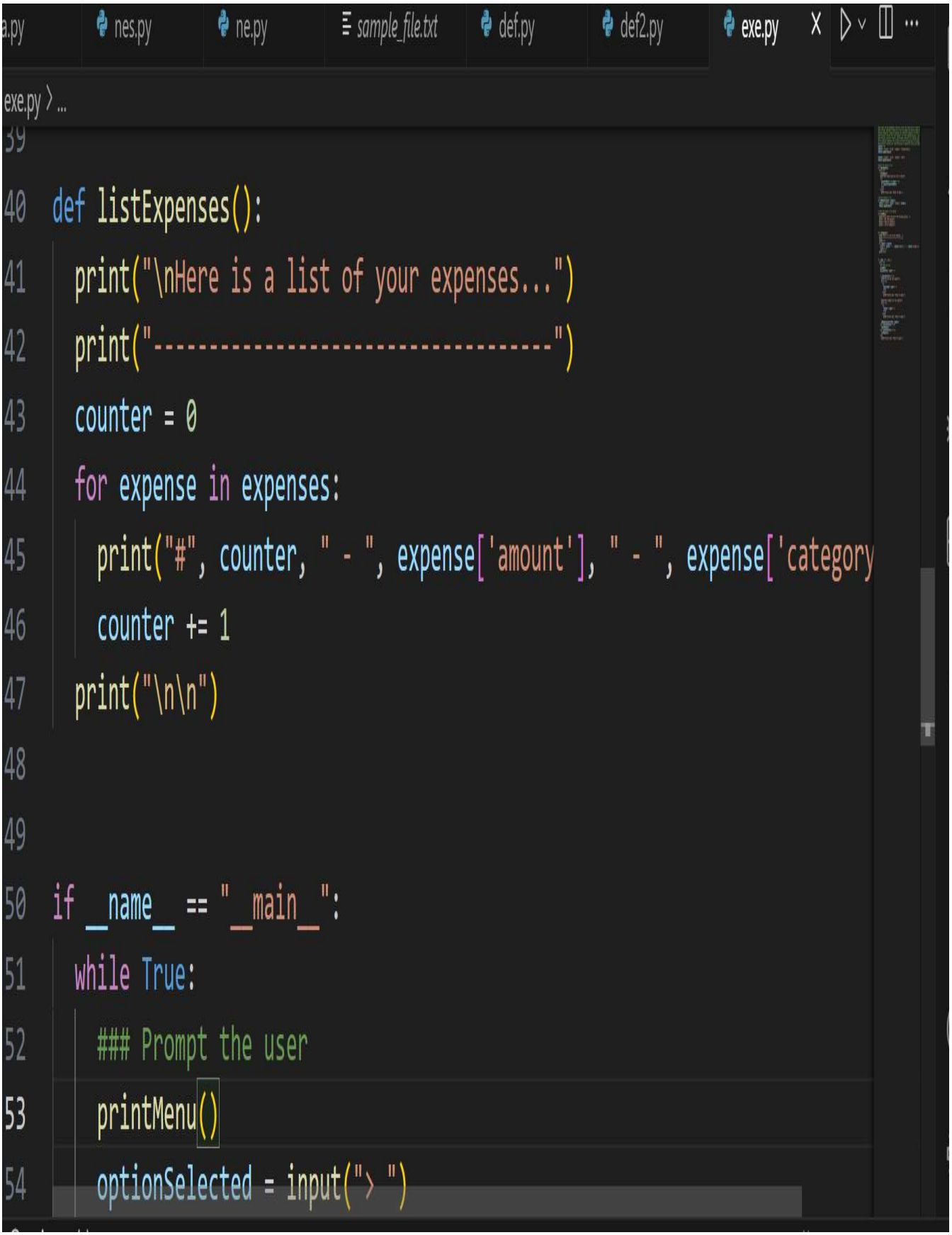

exe.py > amountToAdd
#User Input and Data Management: Develop a system that allows users to inp
#Data Storage: Implement a mechanism to store and manage the entered expen
#Expense Categories: Categorize expenses into different categories for bet
#Data Analysis: Provide users with insights into their spending patterns,
#User-Friendly Interface: Create a user-friendly interface for a seamless
#Error Handling: Implement error handling to ensure the application can ha
#Documentation: Document your code effectively to demonstrate clarity and
expenses = []
expense1 = {'amount': '51.00', 'category': 'transportation'}
expenses.append(expense1)
expense2 = {'amount': '21.55', 'category': 'food'}
expenses.append(expense2)
#remove the expense in list
def removeExpense():

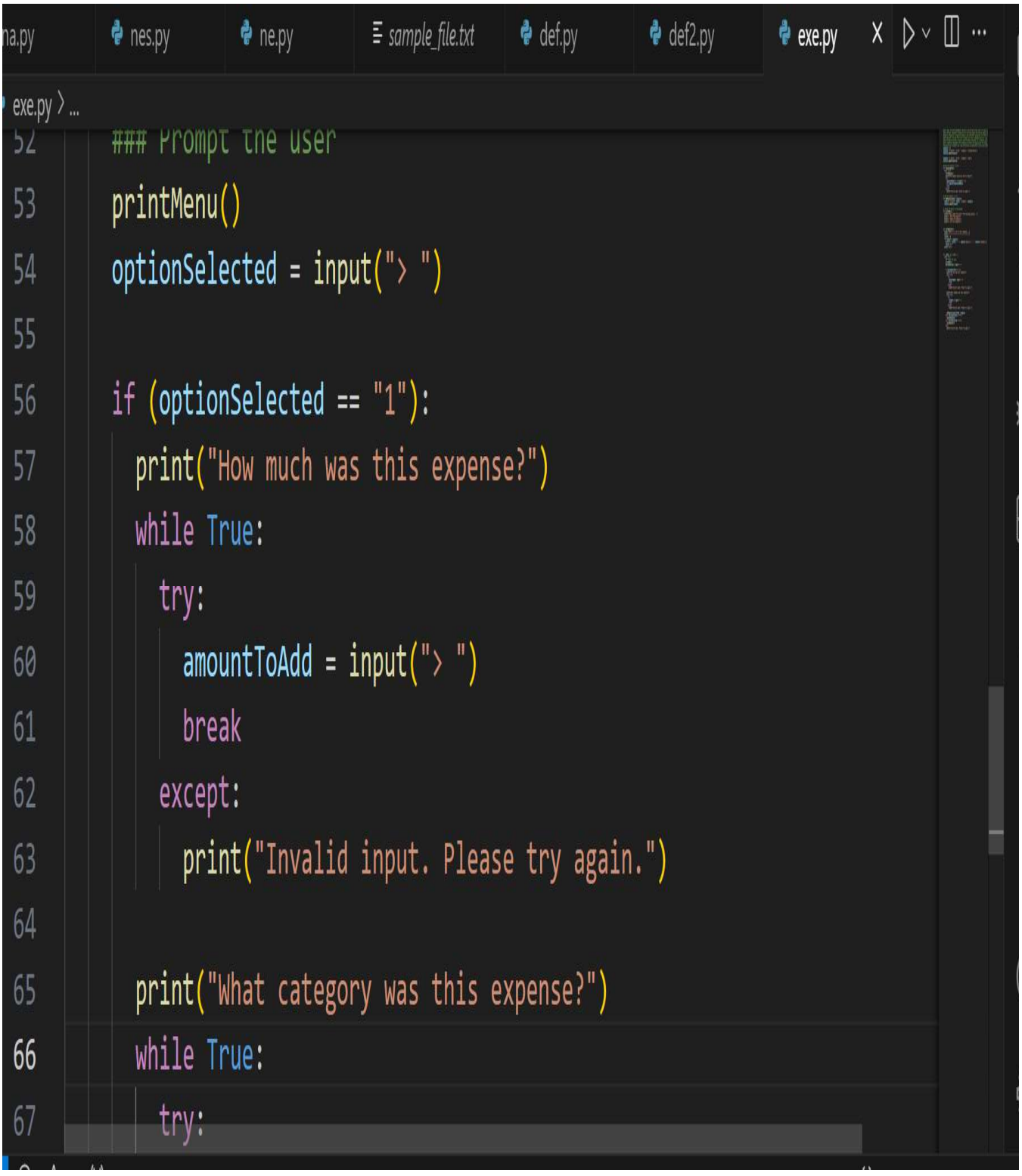
exe.py > addExpense > amount

```
14
15  #remove the expense in list
16  def removeExpense():
17      while True:
18          listExpenses()
19          print("What expense would you like to remove?")
20          try:
21              expenseToRemove = int(input("> "))
22              del expenses[expenseToRemove]
23              break
24          except:
25              print("Invalid input. Please try again.")
26
27  # add the expense in list
28  def addExpense(amount, category):
29      expense = {'amount': amount, 'category': category}
```

(parameter) category: Any

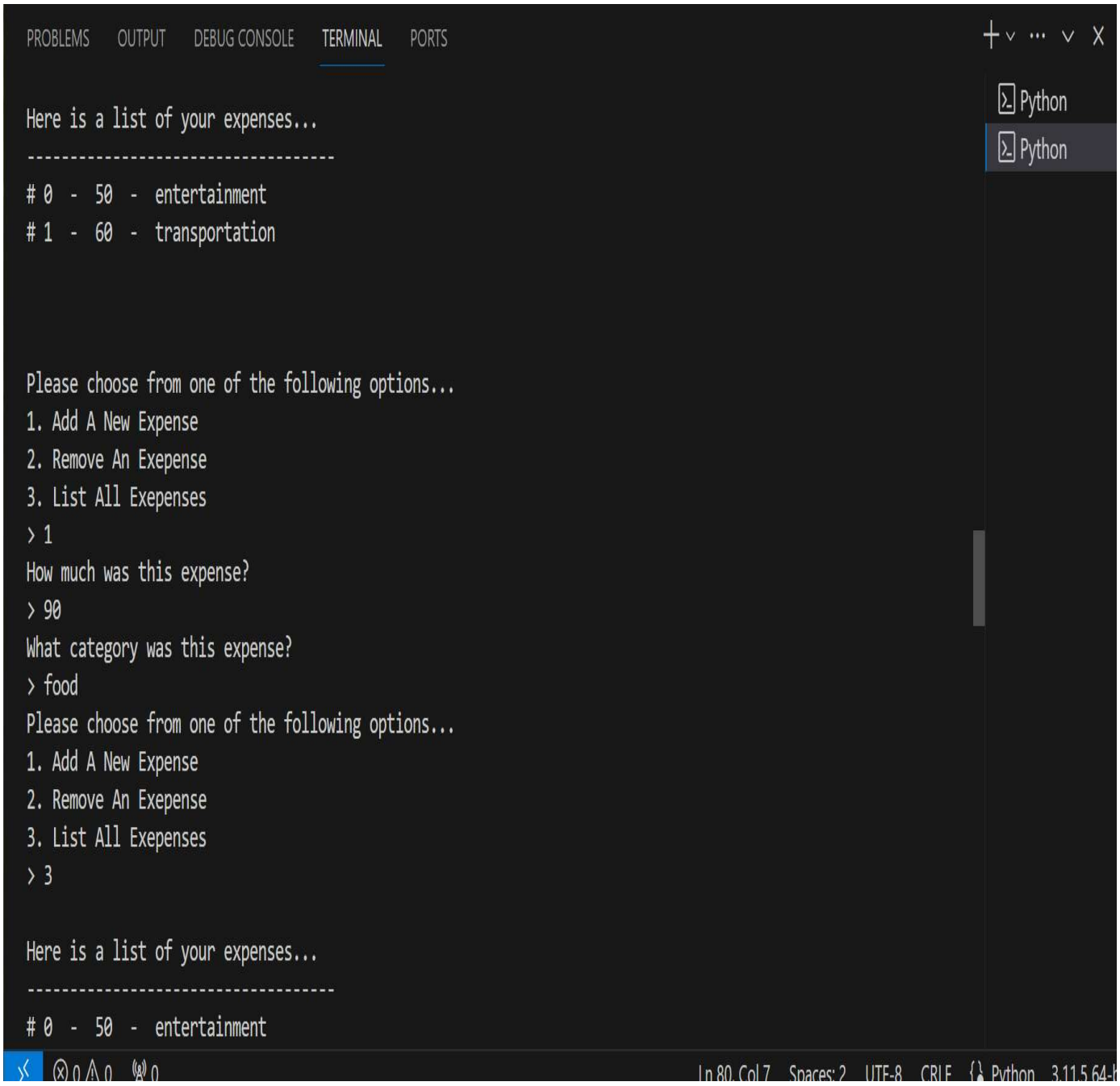

na.py
nes.py
ne.py
sample_file.txt
def.py
def2.py
exe.py
exe.py > listExpenses
26
27 # add the expense in list
28 def addExpense(amount, category):
29 expense = {'amount': amount, 'category': category}
30 expenses.append(expense)
31
32 # choose the option in the expense
33 def printMenu():
34 print("Please choose from one of the following options...")
35 print("1. Add A New Expense")
36 print("2. Remove An Exepense")
37 print("3. List All Exepenses")
38
39
40 def listExpenses():
41 print("\nHere is a list of your expenses...")

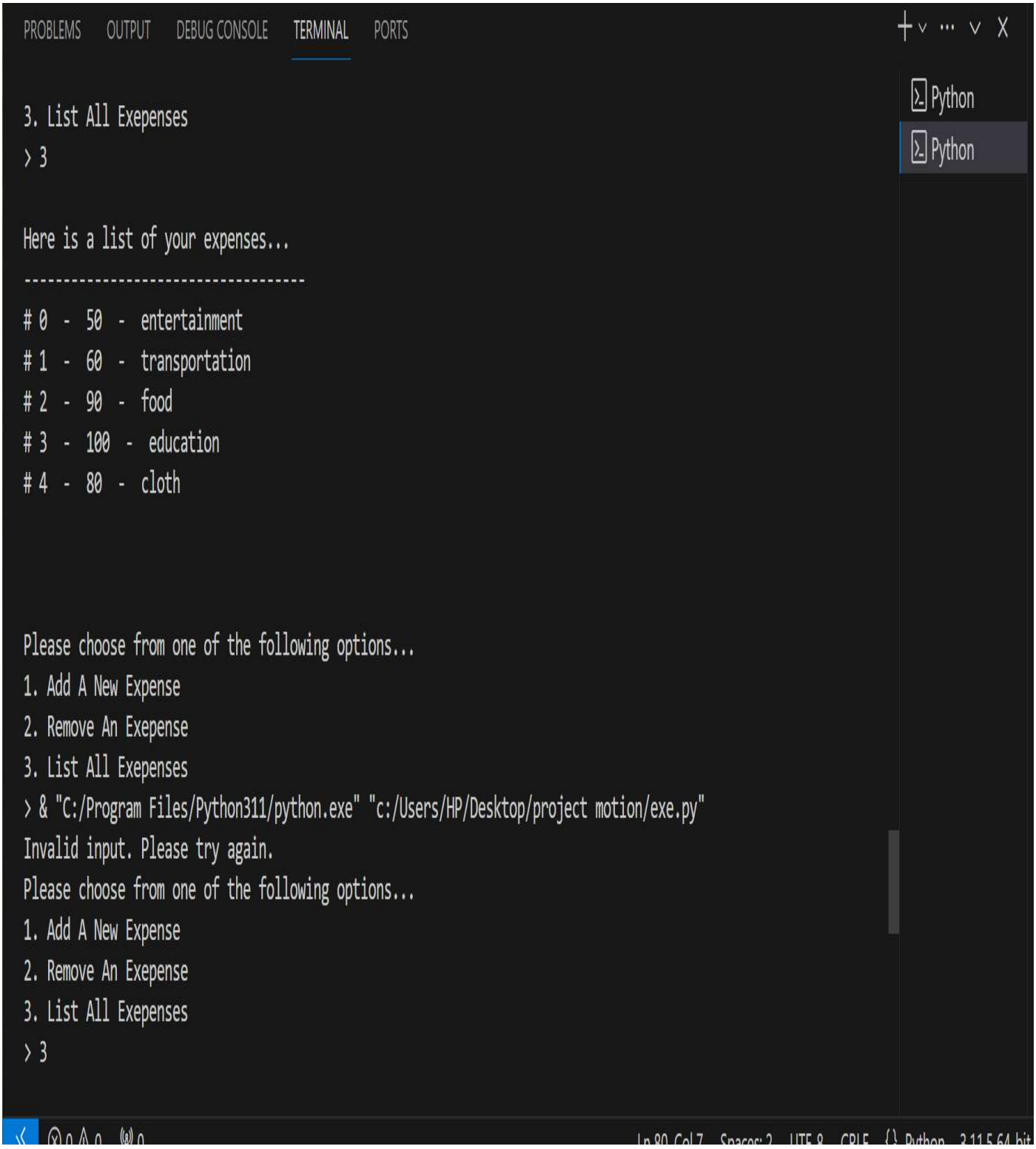

a.py
nes.py
ne.py
sample_file.txt
def.py
def2.py
exe.py
exe.py > ...
39
40 def listExpenses():
41 print("\nHere is a list of your expenses...")
42 print("--------------------------------")
43 counter = 0
44 for expense in expenses:
45 print("#", counter, " - ", expense['amount'], " - ", expense['category
46 counter += 1
47 print("\n\n")
48
49
50 if __name__ == "__main__":
51 while True:
52 ### Prompt the user
53 printMenu()
54 optionSelected = input("> ")

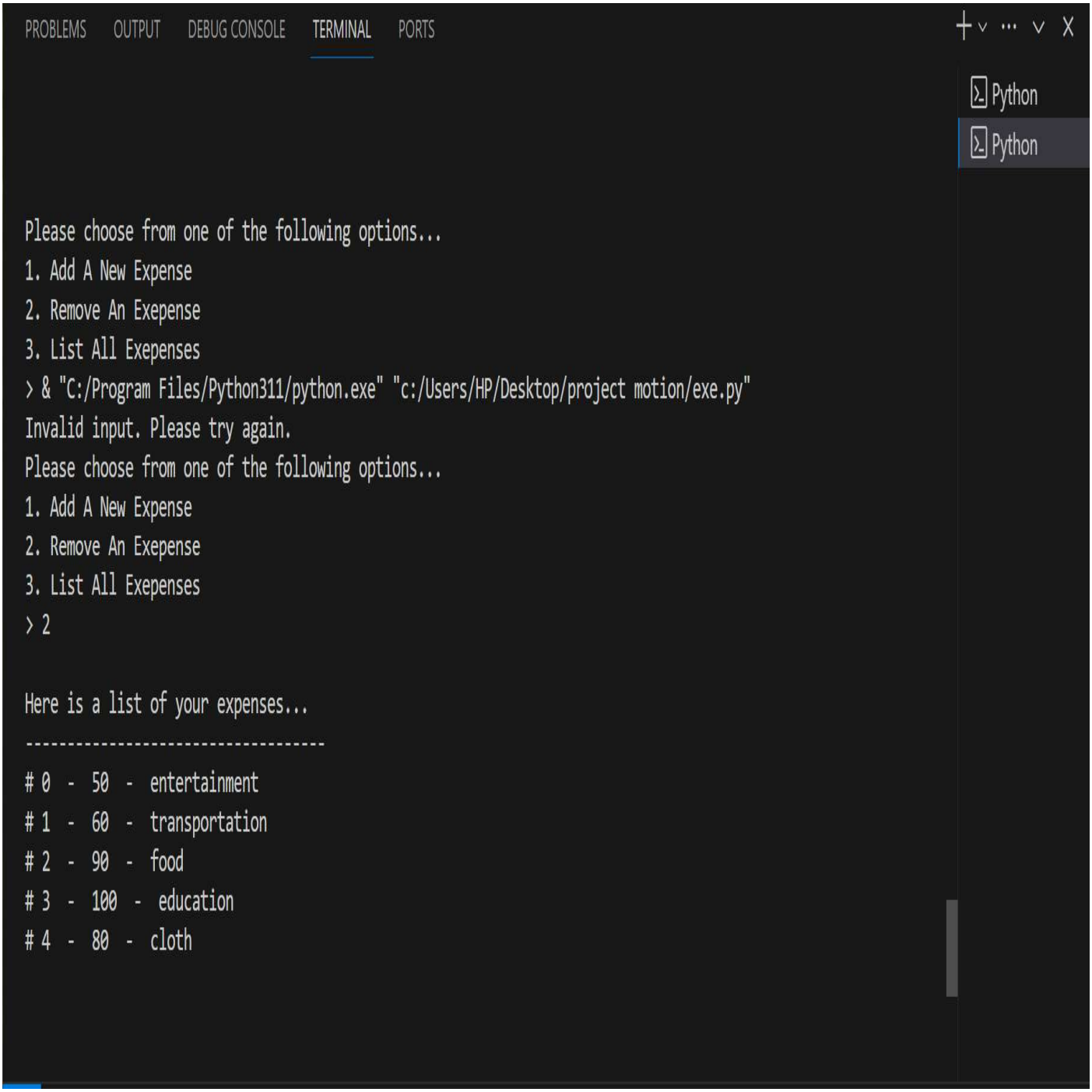

na.py
nes.py
ne.py
sample_file.txt
def.py
def2.py
exe.py
exe.py > ...
Prompt the user
printMenu()
optionSelected = input("> ")
if (optionSelected == "1"):
print("How much was this expense?")
while True:
try:
amountToAdd = input("> ")
break
except:
print("Invalid input. Please try again.")
print("What category was this expense?")
while True:
try:

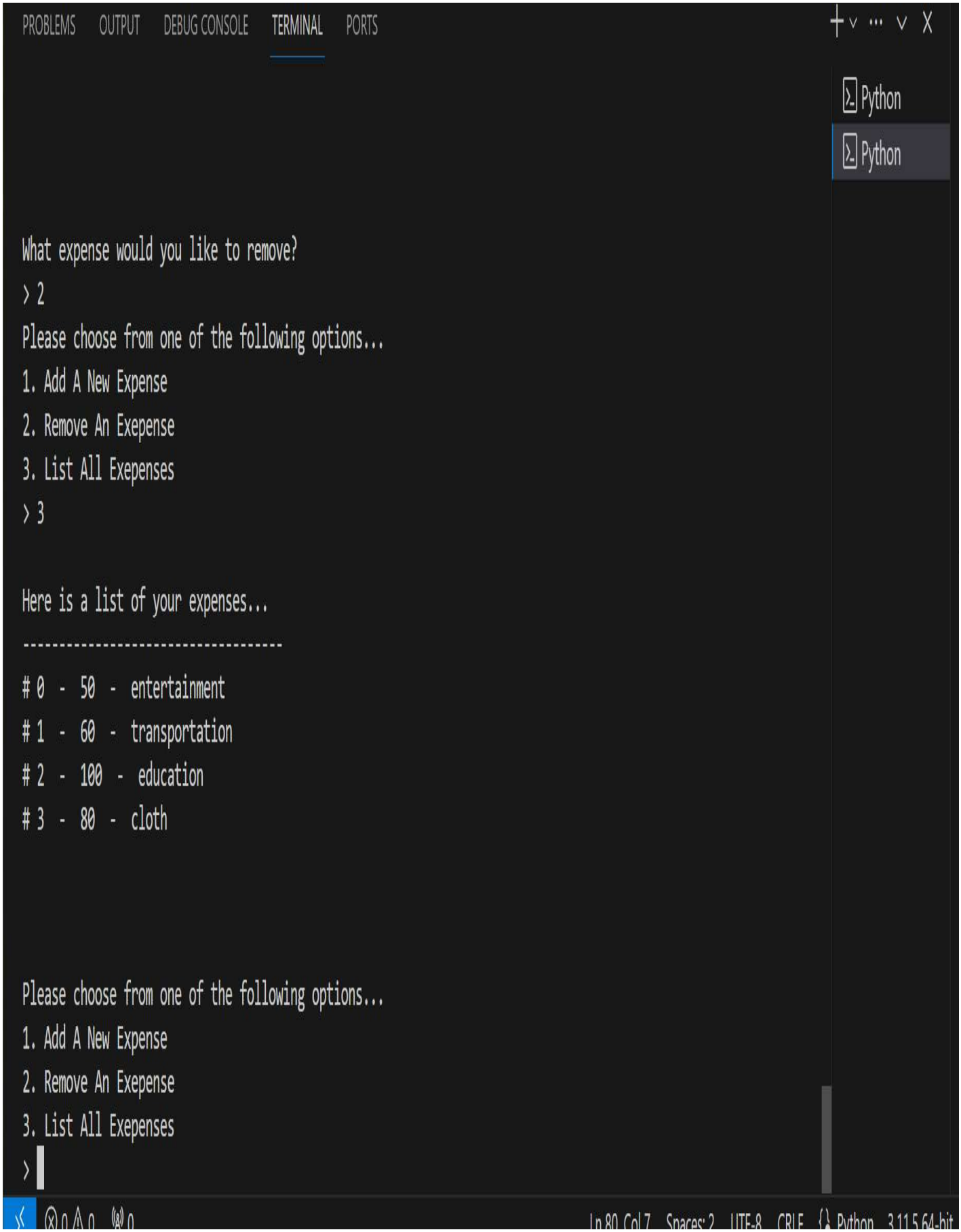
exe.py > ...

```
66        while True:
67          try:
68            category = input("> ")
69            break
70          except:
71            print("Invalid input. Please try again.")
72
73          addExpense(amountToAdd, category)
74        elif (optionSelected == "2"):
75          removeExpense()
76        elif (optionSelected == "3"):
77          listExpenses()
78        else:
79          print("Invalid input. Please try again.")
80
```


PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Python
Python
Here is a list of your expenses...
---
0 - 50 - entertainment
1 - 60 - transportation
Please choose from one of the following options...
1. Add A New Expense
2. Remove An Exepense
3. List All Exepenses
> 1
How much was this expense?
> 90
What category was this expense?
> food
Please choose from one of the following options...
1. Add A New Expense
2. Remove An Exepense
3. List All Exepenses
> 3
Here is a list of your expenses...
---
0 - 50 - entertainment


PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Python
Python
3. List All Exepenses
> 3
Here is a list of your expenses...
0 - 50 - entertainment
1 - 60 - transportation
2 - 90 - food
3 - 100 - education
4 - 80 - cloth
Please choose from one of the following options...
1. Add A New Expense
2. Remove An Exepense
3. List All Exepenses
> & "C:/Program Files/Python311/python.exe" "c:/Users/HP/Desktop/project motion/exe.py"
Invalid input. Please try again.
Please choose from one of the following options...
1. Add A New Expense
2. Remove An Exepense
3. List All Exepenses
> 3


PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Python
Python
Please choose from one of the following options...
1. Add A New Expense
2. Remove An Exepense
3. List All Exepenses
> & "C:/Program Files/Python311/python.exe" "c:/Users/HP/Desktop/project motion/exe.py"
Invalid input. Please try again.
Please choose from one of the following options...
1. Add A New Expense
2. Remove An Exepense
3. List All Exepenses
> 2
Here is a list of your expenses...
0 - 50 - entertainment
1 - 60 - transportation
2 - 90 - food
3 - 100 - education
4 - 80 - cloth


PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Python
Python
What expense would you like to remove?
> 2
Please choose from one of the following options...
1. Add A New Expense
2. Remove An Exepense
3. List All Exepenses
> 3
Here is a list of your expenses...
---
0 - 50 - entertainment
1 - 60 - transportation
2 - 100 - education
3 - 80 - cloth
Please choose from one of the following options...
1. Add A New Expense
2. Remove An Exepense
3. List All Exepenses
>

Thank you