TECHNIK NEST
INNOVATIVE MINDS, NESTING SUCCESS

Name: Saad Bin Haroon:

Intern ID: TN/IN02/PY/026:

Task no: week 2 task:

Internship domain: python language:

Date: 6 August 2025:

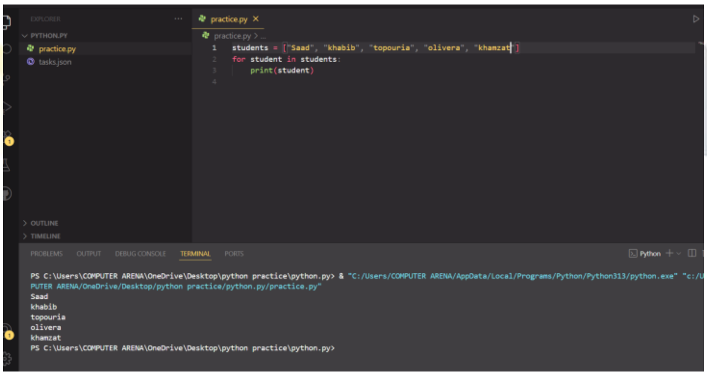
Task 1:

1. Store 5 student names & print each.

   Code:

   ```
   students = ["Saad", "khabib", "topouria", "olivera", "khamzat"]
   for student in students:
       print(student)
   ```

   Output:

   ```
   students = ["Saad", "khabib", "topouria", "olivera", "khamzat"]
   for student in students:
       print(student)
   ```

   ```
   PS C:\Users\COMPUTER ARENA\OneDrive\Desktop\python practice\python.py> & "C:/Users/COMPUTER ARENA/AppData/Local/Programs/Python/Python313/python.exe" "c:/U
   PUTER ARENA/OneDrive/Desktop/python practice/python.py/practice.py"
   Saad
   khabib
   topouria
   olivera
   khamzat
   PS C:\Users\COMPUTER ARENA\OneDrive\Desktop\python practice\python.py>
   ```

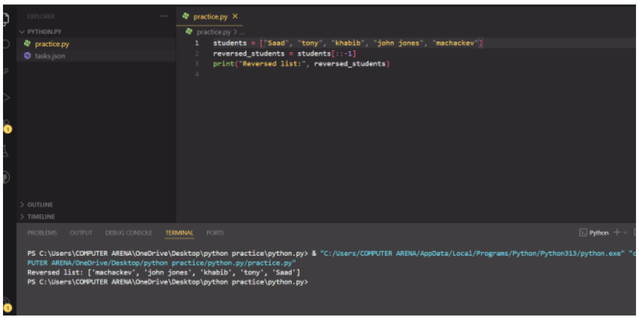
2. Reverse list without reverse()

Code:

```
students = ["Saad", "tony", "khabib", "john jones", "machackev"]

reversed_students = students[::-1]
```

```
print("Reversed list:", reversed_students)
```

Output:



```
students = ["Saad", "tony", "khabib", "john jones", "machackev"]
reversed_students = students[::-1]
print("Reversed list:", reversed_students)
```

```
PS C:\Users\COMPUTER ARENA\OneDrive\Desktop\python practice\python.py> & "C:/Users/COMPUTER ARENA/AppData/Local/Programs/Python/Python313/python.exe" "c
PUTER ARENA/OneDrive/Desktop/python practice/python.py/practice.py"
Reversed list: ['machackev', 'john jones', 'khabib', 'tony', 'Saad']
PS C:\Users\COMPUTER ARENA\OneDrive\Desktop\python practice\python.py>
```

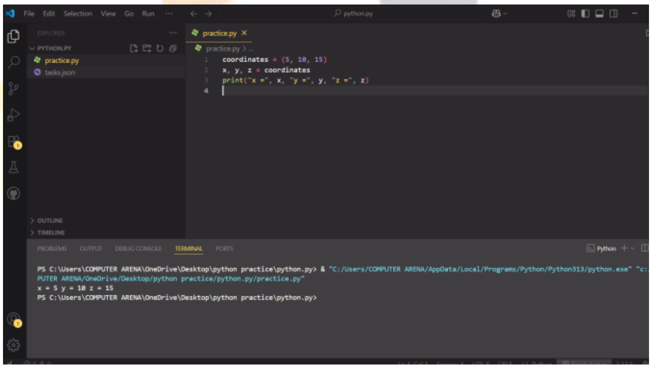
Task 2:

1. Store 3 coordinates & unpack.

Code:

```
coordinates = (5, 10, 15)
x, y, z = coordinates
print("x =", x, "y =", y, "z =", z)
```

Output:



```
coordinates = (5, 10, 15)
x, y, z = coordinates
print("x =", x, "y =", y, "z =", z)
```

```
PS C:\Users\COMPUTER ARENA\OneDrive\Desktop\python practice\python.py> & "C:/Users/COMPUTER ARENA/AppData/Local/Programs/Python/Python313/python.exe" "c:
PUTER ARENA/OneDrive/Desktop/python practice/python.py/practice.py"
x = 5 y = 10 z = 15
PS C:\Users\COMPUTER ARENA\OneDrive\Desktop\python practice\python.py>
```

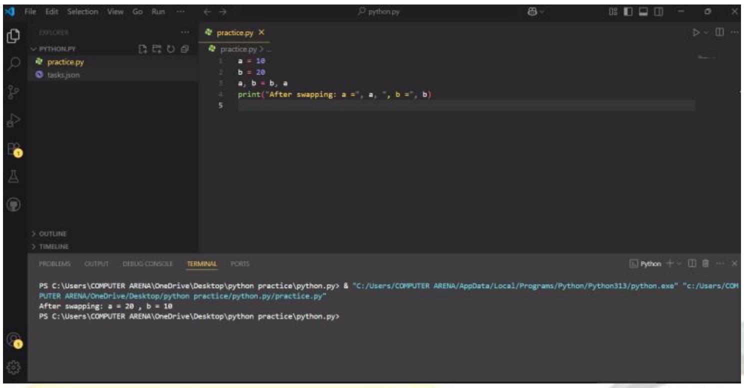
2. Swap vars using tuple assignment.

Code:

```
a = 10
b = 20
a, b = b, a
```

```
print("After swapping: a =", a, ", b =", b)
```

Output:



```
a = 10
b = 20
a, b = b, a
print("After swapping: a =", a, ", b =", b)
```

```
PS C:\Users\COMPUTER ARENA\OneDrive\Desktop\python practice\python.py> & "C:/Users/COMPUTER ARENA/AppData/Local/Programs/Python/Python313/python.exe" "c:/Users/COM
PUTER ARENA/OneDrive/Desktop/python practice/python.py/practice.py"
After swapping: a = 20 , b = 10
PS C:\Users\COMPUTER ARENA\OneDrive\Desktop\python practice\python.py>
```

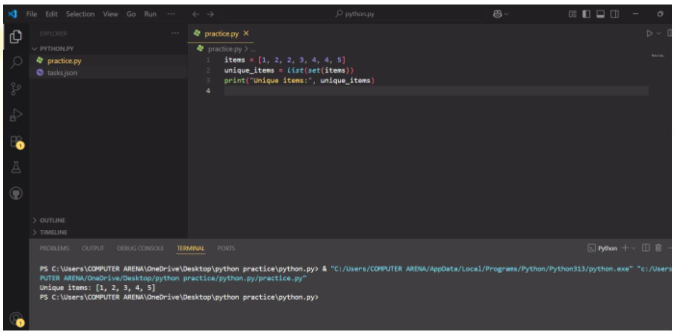
Task 3:

1. Remove duplicates from list.

Code:

```
items = [1, 2, 2, 3, 4, 4, 5]
unique_items = list(set(items))
print("Unique items:", unique_items)
```

Output:



```
items = [1, 2, 2, 3, 4, 4, 5]
unique_items = list(set(items))
print("Unique items:", unique_items)
```

```
PS C:\Users\COMPUTER ARENA\OneDrive\Desktop\python practice\python.py> & "C:/Users/COMPUTER ARENA/AppData/Local/Programs/Python/Python313/python.exe" "c:/Users
PUTER ARENA/OneDrive/Desktop/python practice/python.py/practice.py"
Unique items: [1, 2, 3, 4, 5]
PS C:\Users\COMPUTER ARENA\OneDrive\Desktop\python practice\python.py>
```

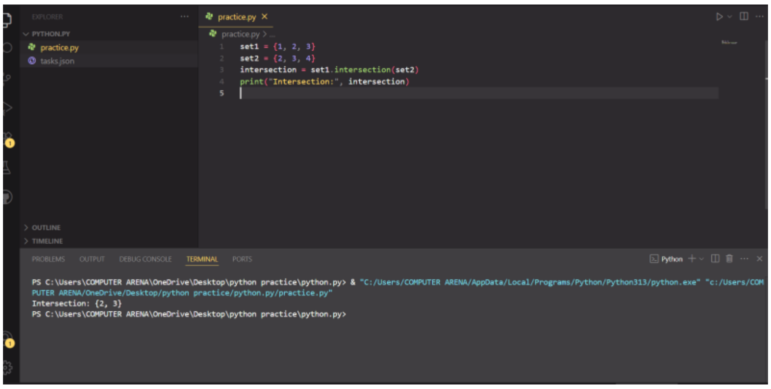
2. Find intersection of two sets.

Code:

```
set1 = {1, 2, 3}
set2 = {2, 3, 4}
intersection = set1.intersection(set2)
```

```
print("Intersection:", intersection)
```

Output:



```
set1 = {1, 2, 3}
set2 = {2, 3, 4}
intersection = set1.intersection(set2)
print("Intersection:", intersection)
```

```
PS C:\Users\COMPUTER ARENA\OneDrive\Desktop\python practice\python.py> & "C:/Users/COMPUTER ARENA/AppData/Local/Programs/Python/Python313/python.exe" "c:/Users/COMPUTER ARENA/OneDrive/Desktop/python practice/python.py/practice.py"
Intersection: {2, 3}
PS C:\Users\COMPUTER ARENA\OneDrive\Desktop\python practice\python.py>
```

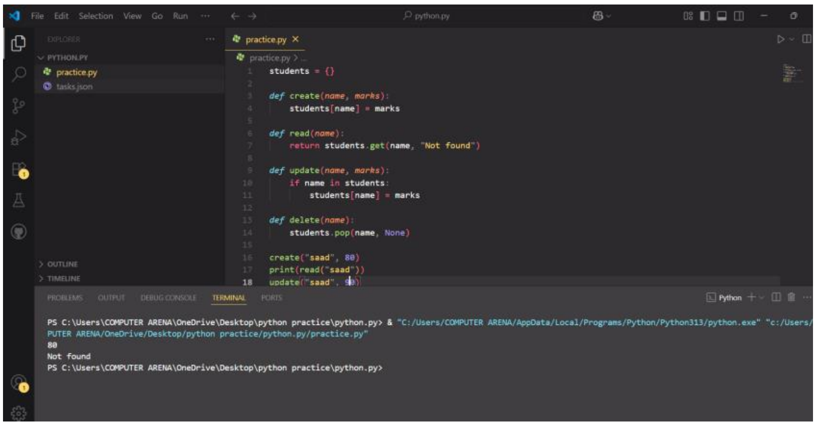
Task 4:

1. Student record CRUD in dict.

Code:

```
students = {}

def create(name, marks):
    students[name] = marks

def read(name):
    return students.get(name, "Not found")

def update(name, marks):
    if name in students:
        students[name] = marks

def delete(name):
    students.pop(name, None)

create("saad", 80)
print(read("saad"))
update("saad", 90)
delete("saad")
print(read("saad"))  # Should print 'Not found'
```

Output:



File Edit Selection View Go Run ···
python.py
EXPLORER
PYTHON.PY
practice.py
tasks.json
OUTLINE
TIMELINE
practice.py
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Python

```
students = {}

def create(name, marks):
    students[name] = marks

def read(name):
    return students.get(name, "Not found")

def update(name, marks):
    if name in students:
        students[name] = marks

def delete(name):
    students.pop(name, None)

create("saad", 80)
print(read("saad"))
update("saad", 90)
```

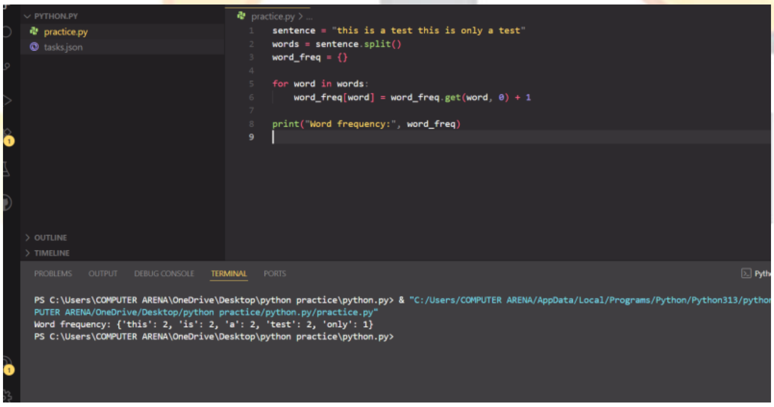
```
PS C:\Users\COMPUTER ARENA\OneDrive\Desktop\python practice\python.py> & "C:/Users/COMPUTER ARENA/AppData/Local/Programs/Python/Python313/python.exe" "c:/Users/
PUTER ARENA/OneDrive/Desktop/python practice/python.py/practice.py"
80
Not found
PS C:\Users\COMPUTER ARENA\OneDrive\Desktop\python practice\python.py>
```

2. Count word frequency in sentence.

Code:

```
sentence = "this is a test this is only a test"
words = sentence.split()
word_freq = {}

for word in words:
    word_freq[word] = word_freq.get(word, 0) + 1

print("Word frequency:", word_freq)
```

Output:



```
sentence = "this is a test this is only a test"
words = sentence.split()
word_freq = {}

for word in words:
    word_freq[word] = word_freq.get(word, 0) + 1

print("Word frequency:", word_freq)
```

```
PS C:\Users\COMPUTER ARENA\OneDrive\Desktop\python practice\python.py> & "C:/Users/COMPUTER ARENA/AppData/Local/Programs/Python/Python313/pytho
PUTER ARENA/OneDrive/Desktop/python practice/python.py/practice.py"
Word frequency: {'this': 2, 'is': 2, 'a': 2, 'test': 2, 'only': 1}
PS C:\Users\COMPUTER ARENA\OneDrive\Desktop\python practice\python.py>
```

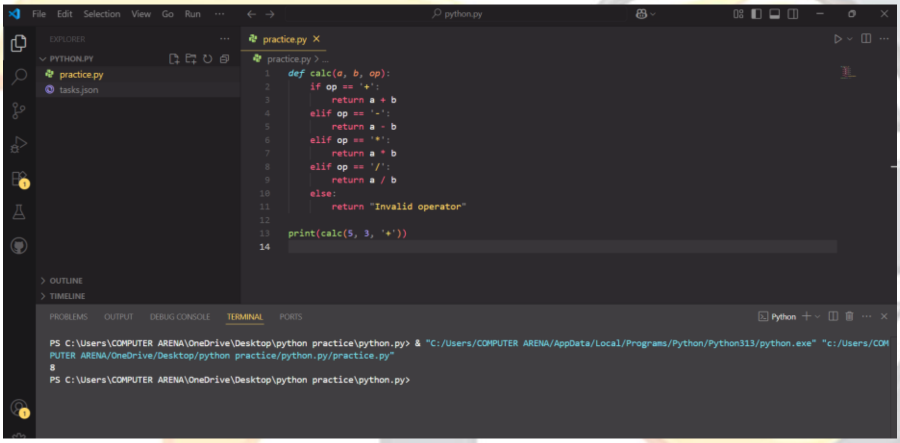
Task 5:

1. Write calc(a,b,op).

Code:

```
def calc(a, b, op):
    if op == '+':
        return a + b
    elif op == '-':
        return a - b
    elif op == '*':
        return a * b
    elif op == '/':
        return a / b
    else:
        return "Invalid operator"

print(calc(5, 3, '+'))
```

Output:



File Edit Selection View Go Run ...
python.py
EXPLORER
PYTHON.PY
practice.py
tasks.json
OUTLINE
TIMELINE
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Python

```
def calc(a, b, op):
    if op == '+':
        return a + b
    elif op == '-':
        return a - b
    elif op == '*':
        return a * b
    elif op == '/':
        return a / b
    else:
        return "Invalid operator"

print(calc(5, 3, '+'))
```

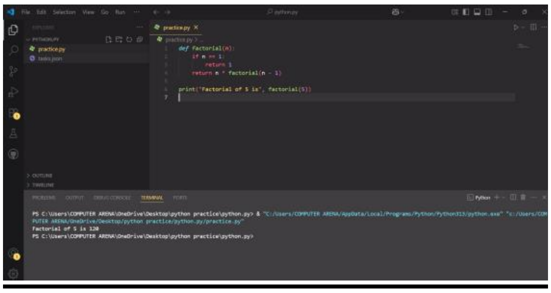
```
PS C:\Users\COMPUTER ARENA\OneDrive\Desktop\python practice\python.py> & "C:/Users/COMPUTER ARENA/AppData/Local/Programs/Python/Python313/python.exe" "c:/Users/COM
PUTER ARENA/OneDrive/Desktop/python practice/python.py/practice.py"
8
PS C:\Users\COMPUTER ARENA\OneDrive\Desktop\python practice\python.py>
```

2. Write factorial(n) recursive.

Code:

```
def factorial(n):
    if n == 1:
        return 1
    return n * factorial(n - 1)

print("Factorial of 5 is", factorial(5))
```

Output:



```
def factorial(n):
    if n == 1:
        return 1
    return n * factorial(n - 1)

print("Factorial of 5 is", factorial(5))
```

```
PS C:\Users\COMPUTER ARENA\OneDrive\Desktop\python practice\python.py> & "C:/Users/COMPUTER ARENA/AppData/Local/Programs/Python/Python313/python.exe" "c:/Users/COM
PUTER ARENA/OneDrive/Desktop/python practice/python.py/practice.py"
Factorial of 5 is 120
PS C:\Users\COMPUTER ARENA\OneDrive\Desktop\python practice\python.py>
```

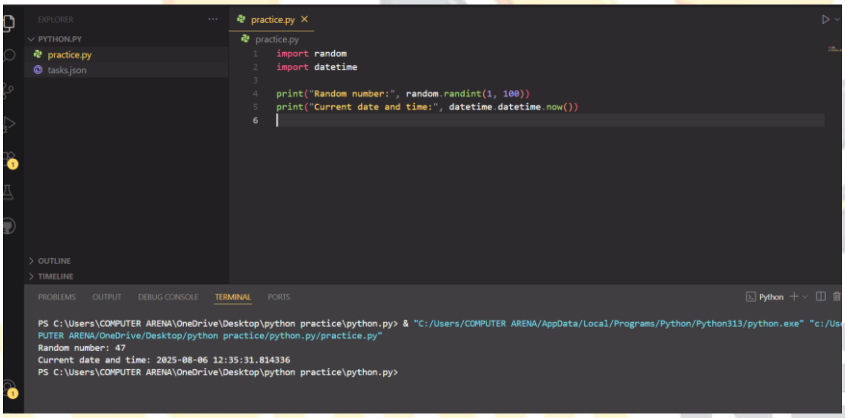
Task 6:

1. Use random & datetime in script.

Code:

```
import random
import datetime

print("Random number:", random.randint(1, 100))
print("Current date and time:", datetime.datetime.now())
```

Output:



```
import random
import datetime

print("Random number:", random.randint(1, 100))
print("Current date and time:", datetime.datetime.now())
```

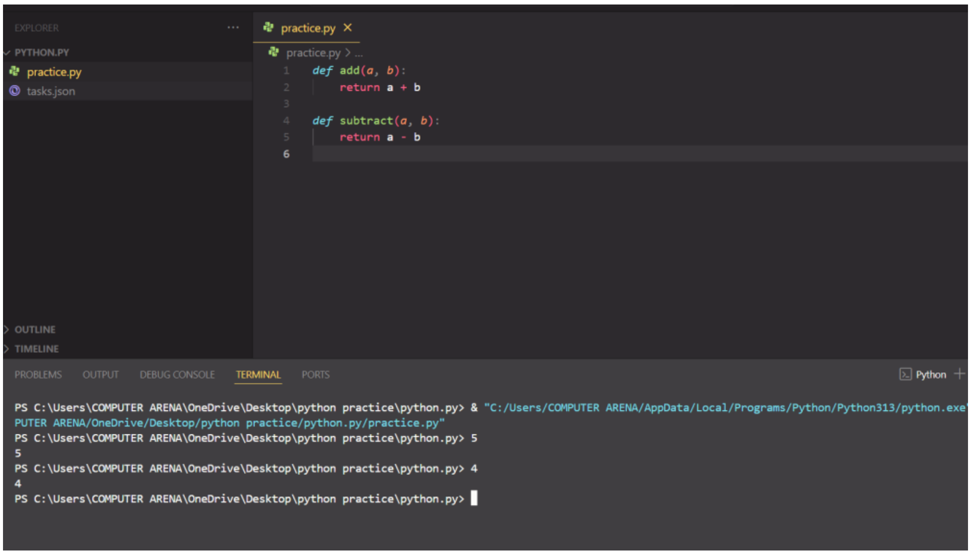
```
PS C:\Users\COMPUTER ARENA\OneDrive\Desktop\python practice\python.py> & "C:/Users/COMPUTER ARENA/AppData/Local/Programs/Python/Python313/python.exe" "c:/Use
PUTER ARENA/OneDrive/Desktop/python practice/python.py/practice.py"
Random number: 47
Current date and time: 2025-08-06 12:35:31.814336
PS C:\Users\COMPUTER ARENA\OneDrive\Desktop\python practice\python.py>
```

2. Create math_utils module & import.

Code:

```
def add(a, b):
    return a + b

def subtract(a, b):
    return a - b
```

Output:



```python
def add(a, b):
    return a + b

def subtract(a, b):
    return a - b
```

```
PS C:\Users\COMPUTER ARENA\OneDrive\Desktop\python practice\python.py> & "C:/Users/COMPUTER ARENA/AppData/Local/Programs/Python/Python313/python.exe"
PUTER ARENA/OneDrive/Desktop/python practice/python.py/practice.py"
PS C:\Users\COMPUTER ARENA\OneDrive\Desktop\python practice\python.py> 5
5
PS C:\Users\COMPUTER ARENA\OneDrive\Desktop\python practice\python.py> 4
4
PS C:\Users\COMPUTER ARENA\OneDrive\Desktop\python practice\python.py>
```

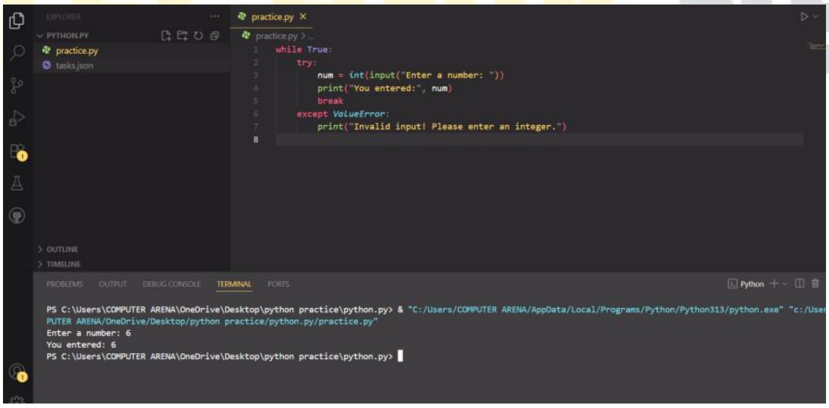
Task 7:

1. Safe int input loop.

Code:

```
while True:
    try:
        num = int(input("Enter a number: "))
        print("You entered:", num)
        break
    except ValueError:
        print("Invalid input! Please enter an integer.")
```

Output:



```python
while True:
    try:
        num = int(input("Enter a number: "))
        print("You entered:", num)
        break
    except ValueError:
        print("Invalid input! Please enter an integer.")
```

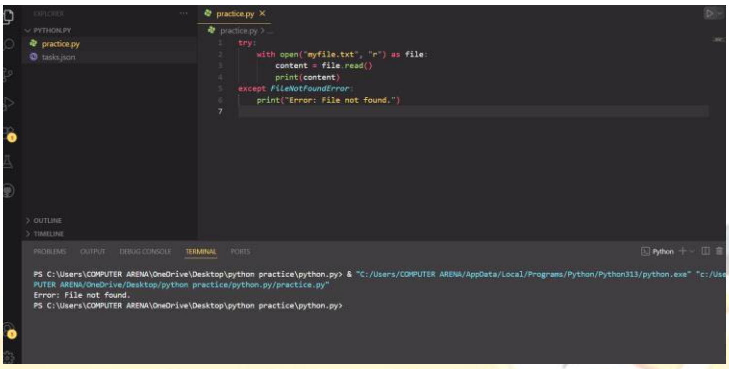
```
PS C:\Users\COMPUTER ARENA\OneDrive\Desktop\python practice\python.py> & "C:/Users/COMPUTER ARENA/AppData/Local/Programs/Python/Python313/python.exe" "c:/User
PUTER ARENA/OneDrive/Desktop/python practice/python.py/practice.py"
Enter a number: 6
You entered: 6
PS C:\Users\COMPUTER ARENA\OneDrive\Desktop\python practice\python.py>
```

2. File open with error message.

Code:

```
try:
    with open("myfile.txt", "r") as file:
        content = file.read()
        print(content)
except FileNotFoundError:
    print("Error: File not found.")
```

Output:



```
try:
    with open("myfile.txt", "r") as file:
        content = file.read()
        print(content)
except FileNotFoundError:
    print("Error: File not found.")
```

```
PS C:\Users\COMPUTER ARENA\OneDrive\Desktop\python practice\python.py> & "C:/Users/COMPUTER ARENA/AppData/Local/Programs/Python/Python313/python.exe" "c:/Use
PUTER ARENA/OneDrive/Desktop/python practice/python.py/practice.py"
Error: File not found.
PS C:\Users\COMPUTER ARENA\OneDrive\Desktop\python practice\python.py>
```

Task 8: hard:

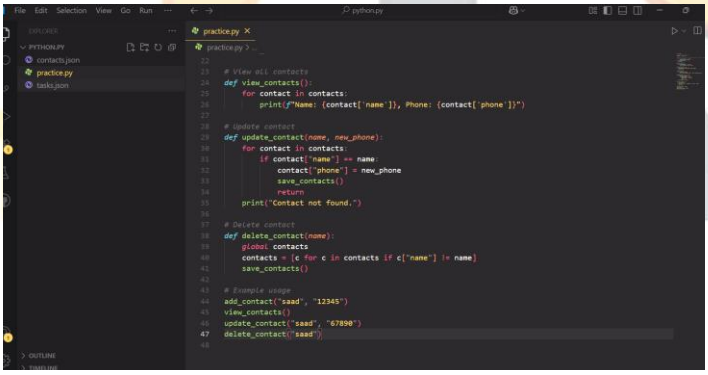
Phonebook App: CRUD contacts dict <-> JSON file storage.

Code:

```
import json

# JSON file to store contact data
filename = "contacts.json"

# Load contacts from file (or start with empty list if file doesn't exist)
try:
    with open(filename, "r") as file:
        contacts = json.load(file)
except FileNotFoundError:
    contacts = []

# Save contacts to file
def save_contacts():
    with open(filename, "w") as file:
        json.dump(contacts, file, indent=4)

# Add a new contact
def add_contact(name, phone):
    contacts.append({"name": name, "phone": phone})
    save_contacts()
```

```
# View all contacts
def view_contacts():
    for contact in contacts:
        print(f"Name: {contact['name']}, Phone: {contact['phone']}")

# Update contact
def update_contact(name, new_phone):
    for contact in contacts:
        if contact["name"] == name:
            contact["phone"] = new_phone
            save_contacts()
            return
    print("Contact not found.")

# Delete contact
def delete_contact(name):
    global contacts
    contacts = [c for c in contacts if c["name"] != name]
    save_contacts()

# Example usage
add_contact("saad", "12345")
view_contacts()
update_contact("saad", "67890")
delete_contact("saad")
```

Output:



```
# View all contacts
def view_contacts():
    for contact in contacts:
        print(f"Name: {contact['name']}, Phone: {contact['phone']}")

# Update contact
def update_contact(name, new_phone):
    for contact in contacts:
        if contact["name"] == name:
            contact["phone"] = new_phone
            save_contacts()
            return
    print("Contact not found.")

# Delete contact
def delete_contact(name):
    global contacts
    contacts = [c for c in contacts if c["name"] != name]
    save_contacts()

# Example usage
add_contact("saad", "12345")
view_contacts()
update_contact("saad", "67890")
delete_contact("saad")
```

Learning from the task:

From these tasks, I learned how to work with Python's core data structures like lists, tuples, sets, and dictionaries, and how to use them in real-world applications. I understood how to perform operations like reversing a list, unpacking tuples, removing duplicates with sets, and managing student records using dictionaries. I practiced writing functions, including a calculator and recursive factorial, and got introduced to working with modules like random and datetime. I also explored creating and importing custom modules. Exception handling was covered to make programs more robust against user and file errors. Lastly, through the Phonebook app, I learned how to implement CRUD operations and store data using JSON files, which is a key step toward building real-world Python applications with persistent storage.