SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE	**DEPARTMENT OF COMPUTER SCIENCE ENGINEERING**

ProgramName:B. Tech	**Assignment Type: Lab**	**AcademicYear:**2025-2026

CourseCoordinatorName	Venkataramana Veeramsetty
Instructor(s)Name	Dr. V. Venkataramana (Co-ordinator) Dr. T. Sampath Kumar Dr. Pramoda Patro Dr. Brij Kishor Tiwari Dr.J.Ravichander Dr. Mohammand Ali Shaik Dr. Anirodh Kumar Mr. S.Naresh Kumar Dr. RAJESH VELPULA Mr. Kundhan Kumar Ms. Ch.Rajitha Mr. M Prakash Mr. B.Raju Intern 1 (Dharma teja) Intern 2 (Sai Prasad) Intern 3 (Sowmya) NS_2 (Mounika)

CourseCode	24CS002PC215	**CourseTitle**	AI Assisted Coding
Year/Sem	II/I	**Regulation**	R24
Date and Day of Assignment	Week1 - Tuesday	**Time(s)**	
Duration	2 Hours	**Applicableto Batches**	24CSBTB01 To 24CSBTB39

AssignmentNumber:1.2(Present assignment number)/**24**(Total number of assignments)

Q.No.	**Question**	***ExpectedTime to complete***
1	Lab 1: Environment Setup – GitHub Copilot and VS Code Integration **Lab Objectives:** ● To install and configure GitHub Copilot in Visual Studio Code.	Week1 - wednesday

- To explore AI-assisted code generation using GitHub Copilot.
- To analyze the accuracy and effectiveness of Copilot's code suggestions.
- To understand prompt-based programming using comments and code context

Lab Outcomes (LOs):
After completing this lab, students will be able to:

- Set up GitHub Copilot in VS Code successfully.
- Use inline comments and context to generate code with Copilot.
- Evaluate AI-generated code for correctness and readability.
- Compare code suggestions based on different prompts and programming styles.

Task Description#1

- Write a comment: # Function to check if a string is a valid palindrome (ignoring spaces and case) and allow Copilot to complete it.

Expected Output#1

- A function that correctly returns True for phrases like "A man a plan a canal Panama"

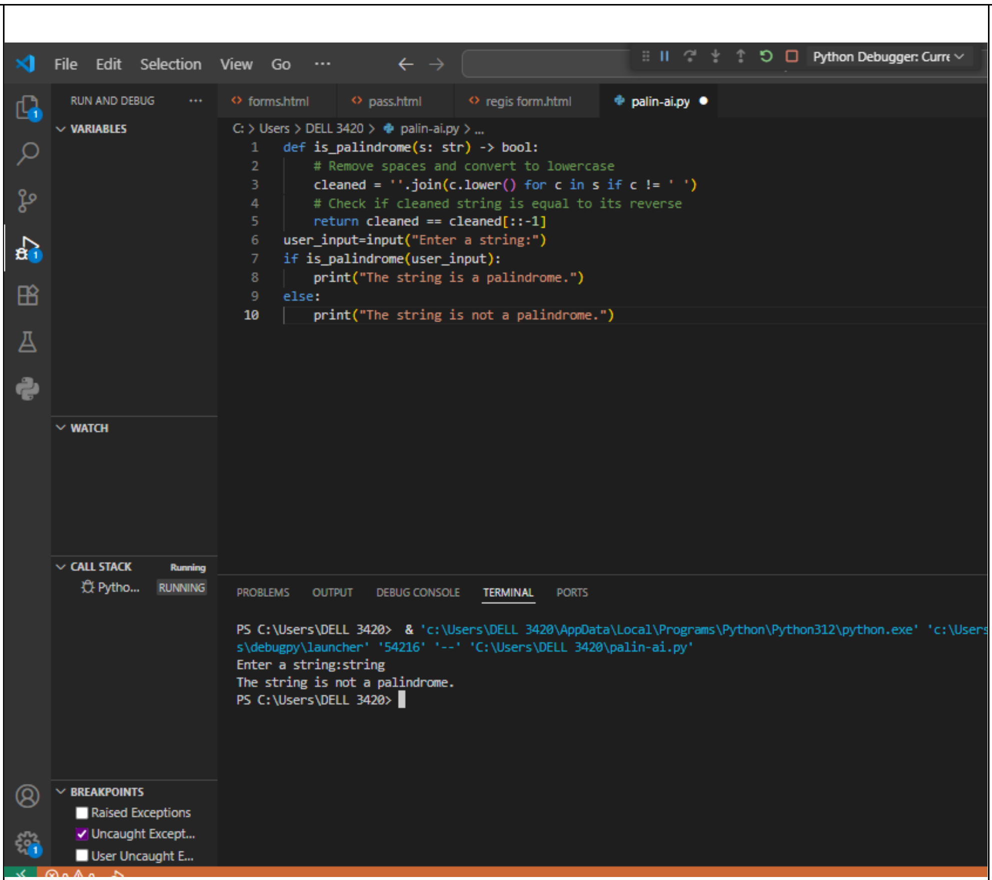
Prompt #1:

1. Write a function to check if a string is a palindrome.
2. Ignore spaces in the string while checking.

Observations:

1. Co-poilot will generate automatic code for palindrome.
2. It should ignore spaces and case while checking.
3. Copilot will likely make a reverse-string check.

CODE:



```python
def is_palindrome(s: str) -> bool:
    # Remove spaces and convert to lowercase
    cleaned = ''.join(c.lower() for c in s if c != ' ')
    # Check if cleaned string is equal to its reverse
    return cleaned == cleaned[::-1]
user_input=input("Enter a string:")
if is_palindrome(user_input):
    print("The string is a palindrome.")
else:
    print("The string is not a palindrome.")
```

```
PS C:\Users\DELL 3420>  & 'c:\Users\DELL 3420\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users
s\debugpy\launcher' '54216' '--' 'C:\Users\DELL 3420\palin-ai.py'
Enter a string:string
The string is not a palindrome.
PS C:\Users\DELL 3420> 
```

Code Explanation:

1. user types a string
2. The program removes spaces and makes everything lowercase
3. It checks if the cleaned string is the same as its reverse
4. If same → prints"valid palindrome".
5. If not → prints "not a palindrome ".

Task Description#2

- Generate a Python function that returns the Fibonacci sequence up to n terms. Prompt with only a function header and docstring

Expected Output#2

- AI completes the function logic using loop or recursion with accurate output

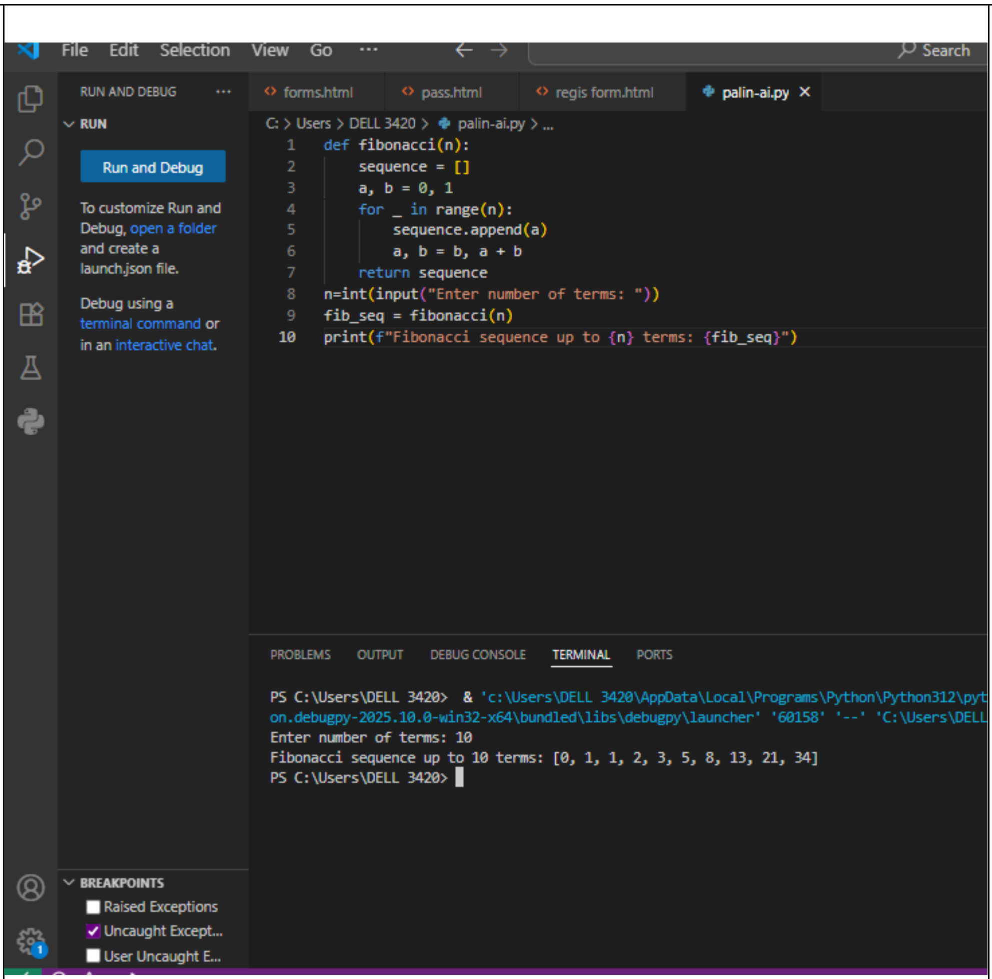
Prompt #2:

1. Take n as number of terms
2.Define a function fionacci(n).
3.Output should be a list of Fibonacci numbers.

Observations:

1. The prompt has only the function header and docstring.
2. Copilot understands it should generate Fibonacci numbers.
3. Output should match the sequence for given n.
4. AI may use 1oop or recursion to solve it.

CODE:



To customize Run and Debug, open a folder and create a launch.json file.

Debug using a terminal command or in an interactive chat.

```python
def fibonacci(n):
    sequence = []
    a, b = 0, 1
    for _ in range(n):
        sequence.append(a)
        a, b = b, a + b
    return sequence
n=int(input("Enter number of terms: "))
fib_seq = fibonacci(n)
print(f"Fibonacci sequence up to {n} terms: {fib_seq}")
```

```
PS C:\Users\DELL 3420>  & 'c:\Users\DELL 3420\AppData\Local\Programs\Python\Python312\pyt
on.debugpy-2025.10.0-win32-x64\bundled\libs\debugpy\launcher' '60158' '--' 'C:\Users\DELL
Enter number of terms: 10
Fibonacci sequence up to 10 terms: [0, 1, 1, 2, 3, 5, 8, 13, 21, 34]
PS C:\Users\DELL 3420> 
```

Code Explanation:

This program starts with 0 and 1, then keeps adding the last two numbers to get the next one, until it prints as many terms as you asked for.

Task Description#3

- Write a comment like # Function to reverse a string and use Copilot to generate the function.

Expected Output#3

- Auto-completed reverse function

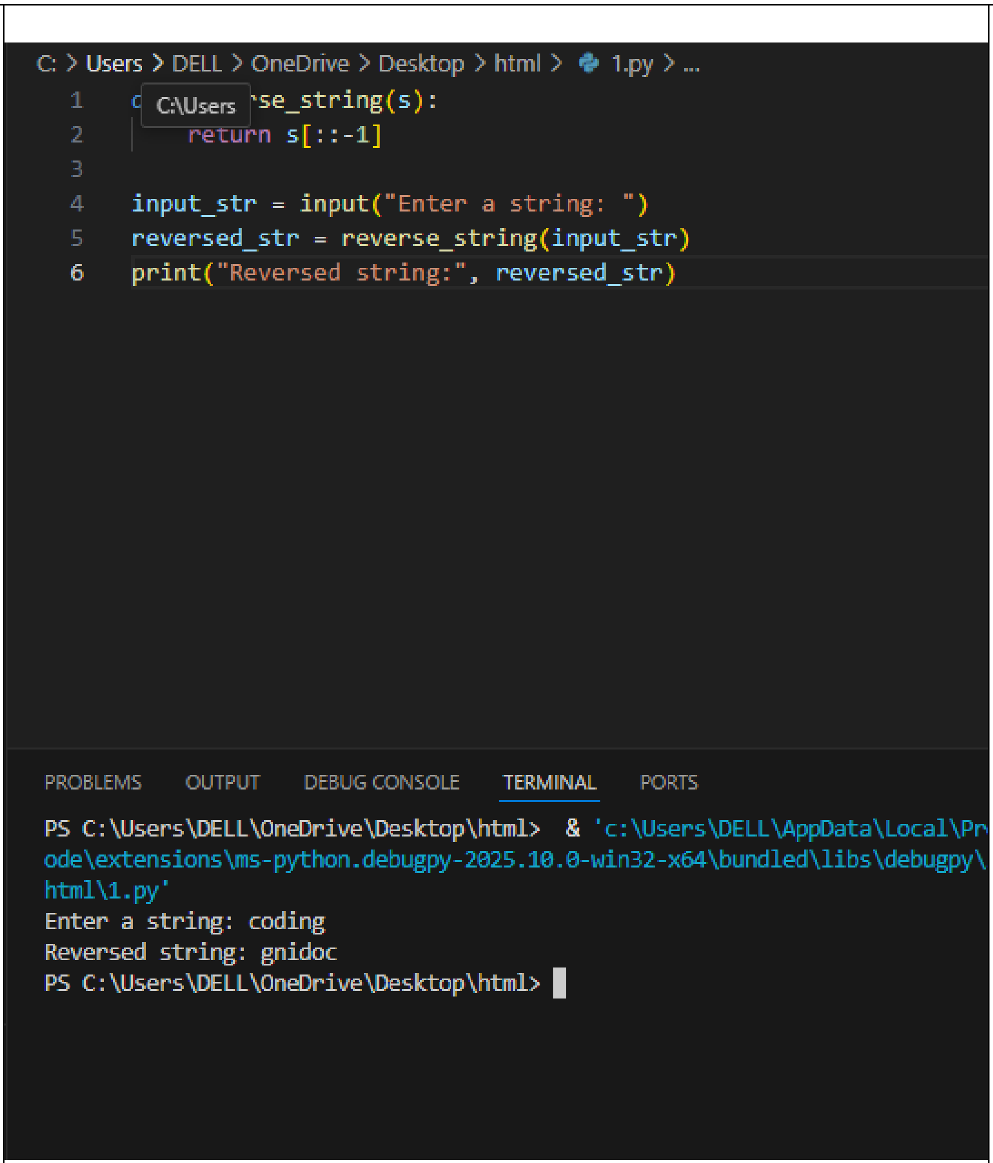
Prompt #3:

1.Function to reverse a string
2.Take input string
3. Return reversed string

Observations:

1. This tells Copilot to reverse a string.
2. Copilot will likely use slicing or a loop.
3. Works for any string input.
4. Should be tested with empty and single-character strings.

CODE:



```python
d         se_string(s):
    return s[::-1]

input_str = input("Enter a string: ")
reversed_str = reverse_string(input_str)
print("Reversed string:", reversed_str)
```

```
PS C:\Users\DELL\OneDrive\Desktop\html>  & 'c:\Users\DELL\AppData\Local\Pr
ode\extensions\ms-python.debugpy-2025.10.0-win32-x64\bundled\libs\debugpy\
html\1.py'
Enter a string: coding
Reversed string: gnidoc
PS C:\Users\DELL\OneDrive\Desktop\html> 
```

Code Explantion:

This program takes a word from you, flips it backward using Python slicing ([::-1]), and shows you the reversed result.

Task Description#4

- Generate a program that simulates a basic calculator (add, subtract, multiply, divide). Write the comment: # Simple calculator with 4 operations and let AI complete it.

Expected Output#4

- Fully working calculator with input/output and operator selection logic

Prompt #4:

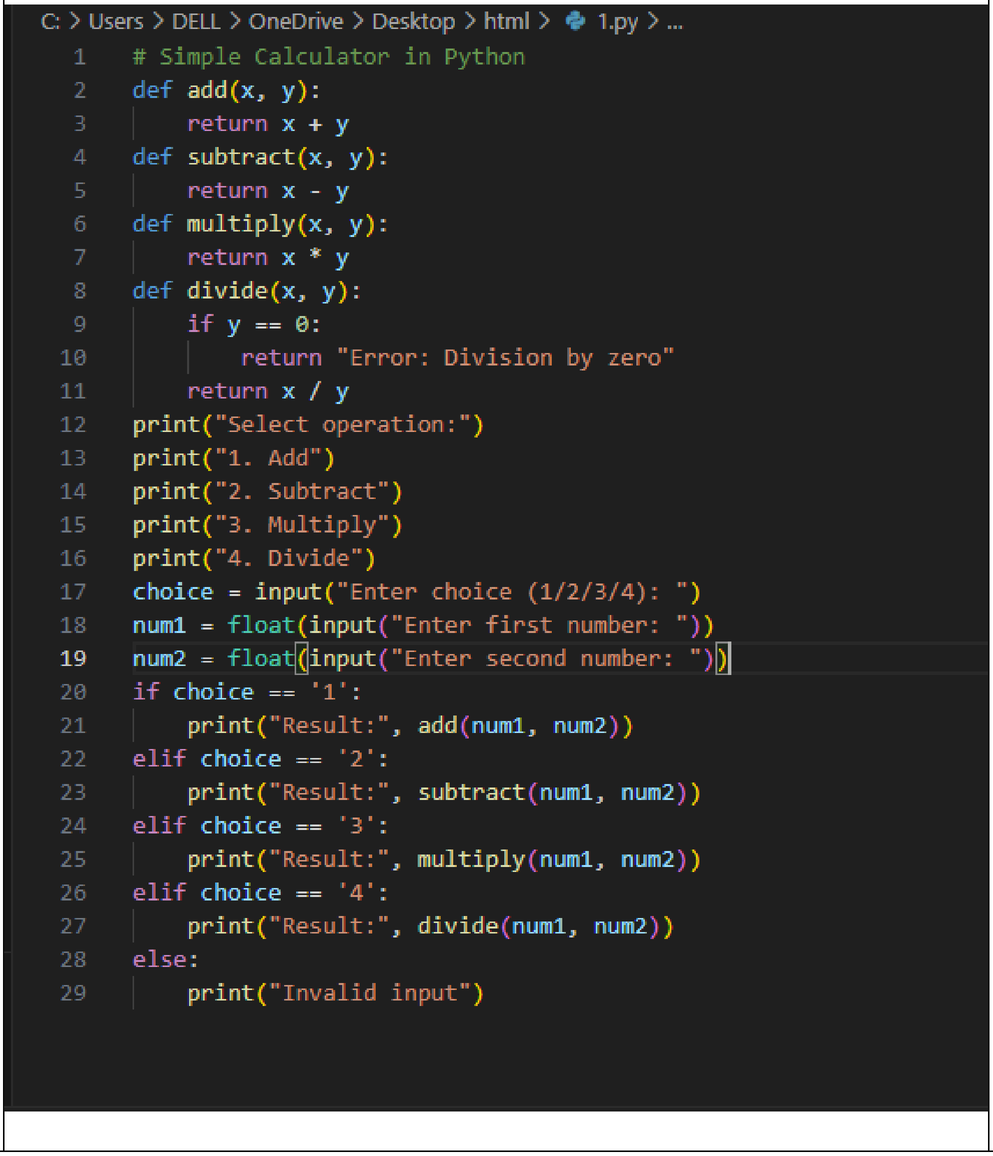
1.Simple calculator with 4 operations
2Add, subtract, multiply, divide
3. Take user input and show result

Observations:

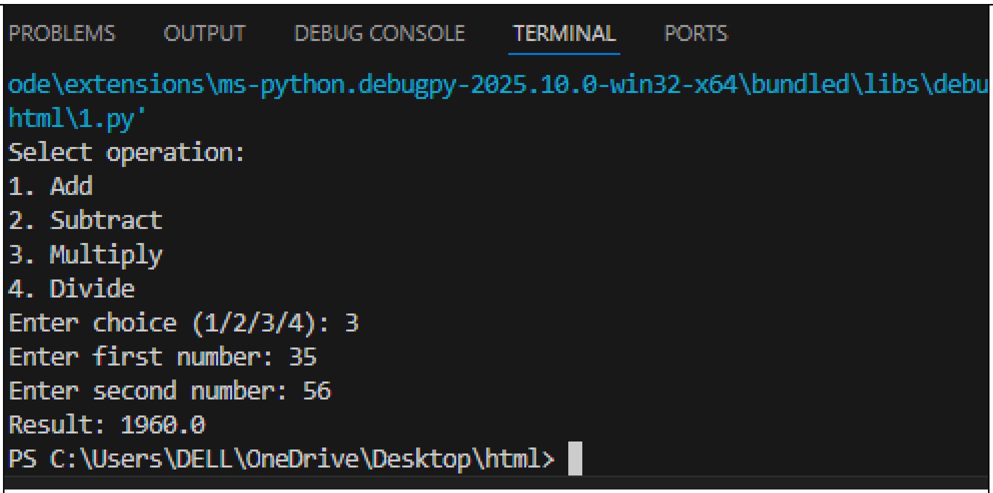
1. This tells Copilot to make a basic calculator.
2. It should support add, subtract, multiply, and divide.
3. Likely will include user input for numbers and operation choice.

4. Needs to handle division by zero.
5. Should be tested with different numbers and operators.

CODE:



```python
# Simple Calculator in Python
def add(x, y):
    return x + y
def subtract(x, y):
    return x - y
def multiply(x, y):
    return x * y
def divide(x, y):
    if y == 0:
        return "Error: Division by zero"
    return x / y
print("Select operation:")
print("1. Add")
print("2. Subtract")
print("3. Multiply")
print("4. Divide")
choice = input("Enter choice (1/2/3/4): ")
num1 = float(input("Enter first number: "))
num2 = float(input("Enter second number: "))
if choice == '1':
    print("Result:", add(num1, num2))
elif choice == '2':
    print("Result:", subtract(num1, num2))
elif choice == '3':
    print("Result:", multiply(num1, num2))
elif choice == '4':
    print("Result:", divide(num1, num2))
else:
    print("Invalid input")
```



```
ode\extensions\ms-python.debugpy-2025.10.0-win32-x64\bundled\libs\debu
html\1.py'
Select operation:
1. Add
2. Subtract
3. Multiply
4. Divide
Enter choice (1/2/3/4): 3
Enter first number: 35
Enter second number: 56
Result: 1960.0
PS C:\Users\DELL\OneDrive\Desktop\html>
```

Code Explanation:

This program asks what math you want to do, takes two numbers, performs the operation using functions, and shows the result.

Task Description#5

- Use a comment to instruct AI to write a function that reads a file and returns the number of lines..

Expected Output#5

- Functional implementation using open() or with open() and readlines()

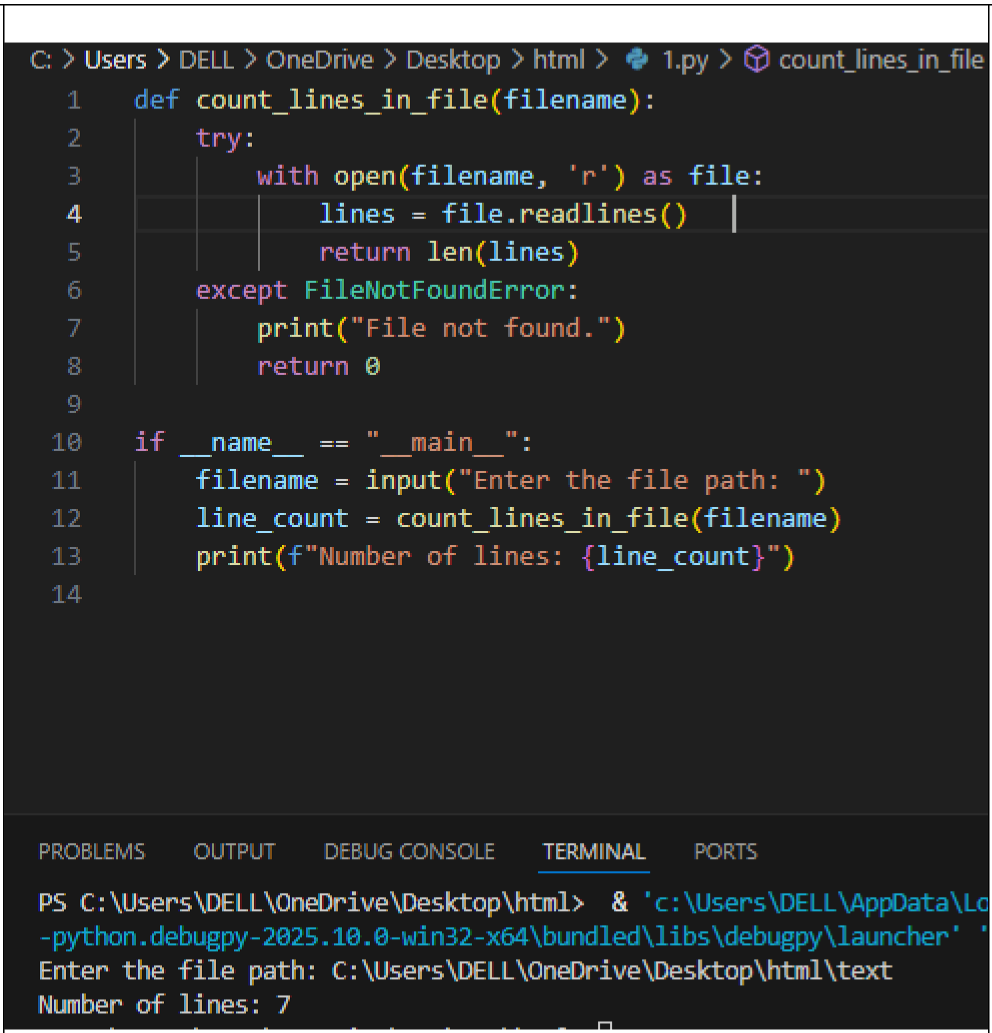
Prompt #5:

1.Function to read a file
2. Count the number of lines
3. Return the count

Observations:

1. This explains the task.
2. AI will likely use open() or with open() to read the file.
3. Counting can be done with len(readlines()) or a loop.
4. Needs testing with empty and multi-line files.

CODE:


C: > Users > DELL > OneDrive > Desktop > html > 1.py > count_lines_in_file
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS

```python
def count_lines_in_file(filename):
    try:
        with open(filename, 'r') as file:
            lines = file.readlines()
            return len(lines)
    except FileNotFoundError:
        print("File not found.")
        return 0

if __name__ == "__main__":
    filename = input("Enter the file path: ")
    line_count = count_lines_in_file(filename)
    print(f"Number of lines: {line_count}")
```

```
PS C:\Users\DELL\OneDrive\Desktop\html>  & 'c:\Users\DELL\AppData\Lo
-python.debugpy-2025.10.0-win32-x64\bundled\libs\debugpy\launcher' '
Enter the file path: C:\Users\DELL\OneDrive\Desktop\html\text
Number of lines: 7
```

Code Explanation:

This program opens a file, counts how many lines it has, and shows the result. If the file doesn't exist, it tells you.

Note: Report should be submitted a word document for all tasks in a single document with prompts, comments & code explanation, and output and if required, screenshots

Evaluation Criteria:

Criteria	Max Marks
Task #1	0.5
Task #2	0.5
Task #3	0.5
Task #4	0.5
Task #5	0.5
Total	**2.5 Marks**