# AI Assisted Coding Lab Assignment (1.2)

Name : K.Poojitha

Roll No : 2403A51185

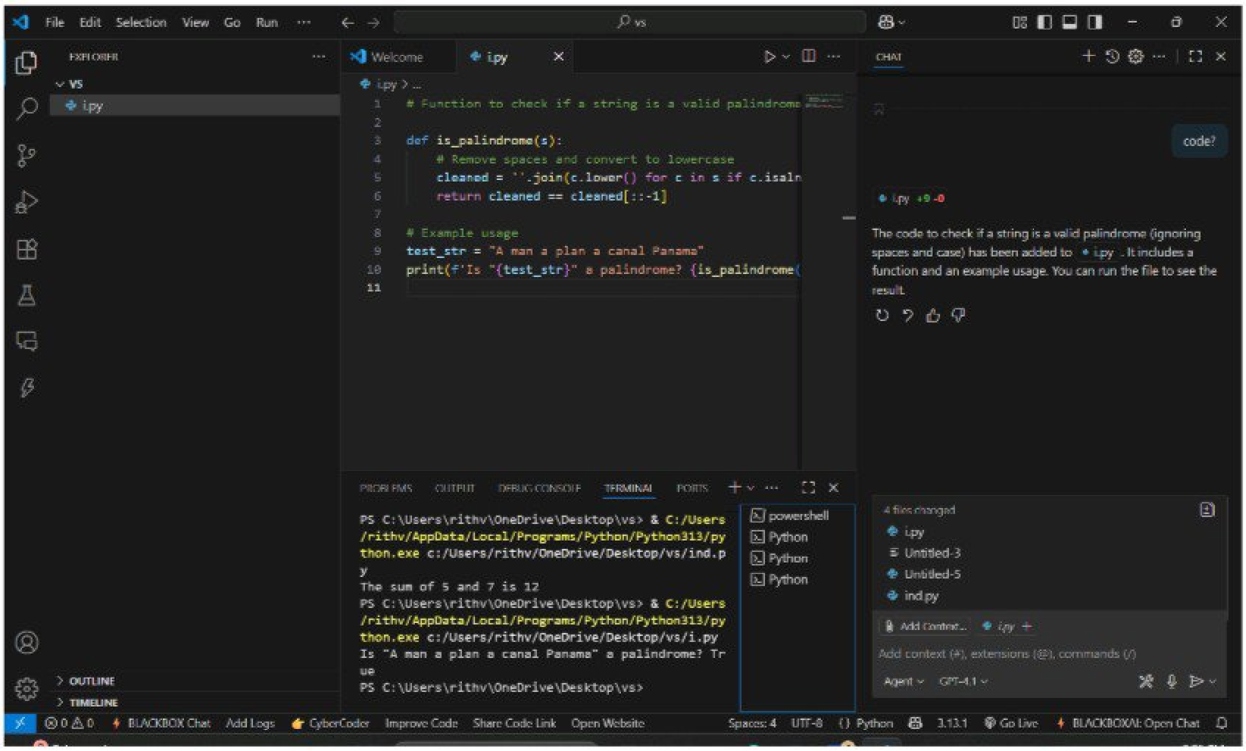
Section : CS-B09

## Task 1:

### Prompt used:

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

Code:



```
# Function to check if a string is a valid palindrome

def is_palindrome(s):
    # Remove spaces and convert to lowercase
    cleaned = ''.join(c.lower() for c in s if c.isaln
    return cleaned == cleaned[::-1]

# Example usage
test_str = "A man a plan a canal Panama"
print(f'Is "{test_str}" a palindrome? {is_palindrome(
```

```
PS C:\Users\rithv\OneDrive\Desktop\vs> & C:/Users
/rithv/AppData/Local/Programs/Python/Python313/py
thon.exe c:/Users/rithv/OneDrive/Desktop/vs/ind.p
y
The sum of 5 and 7 is 12
PS C:\Users\rithv\OneDrive\Desktop\vs> & C:/Users
/rithv/AppData/Local/Programs/Python/Python313/py
thon.exe c:/Users/rithv/OneDrive/Desktop/vs/i.py
Is "A man a plan a canal Panama" a palindrome? Tr
ue
PS C:\Users\rithv\OneDrive\Desktop\vs>
```

code?

i.py +9 -0

The code to check if a string is a valid palindrome (ignoring spaces and case) has been added to i.py . It includes a function and an example usage. You can run the file to see the result.

4 files changed

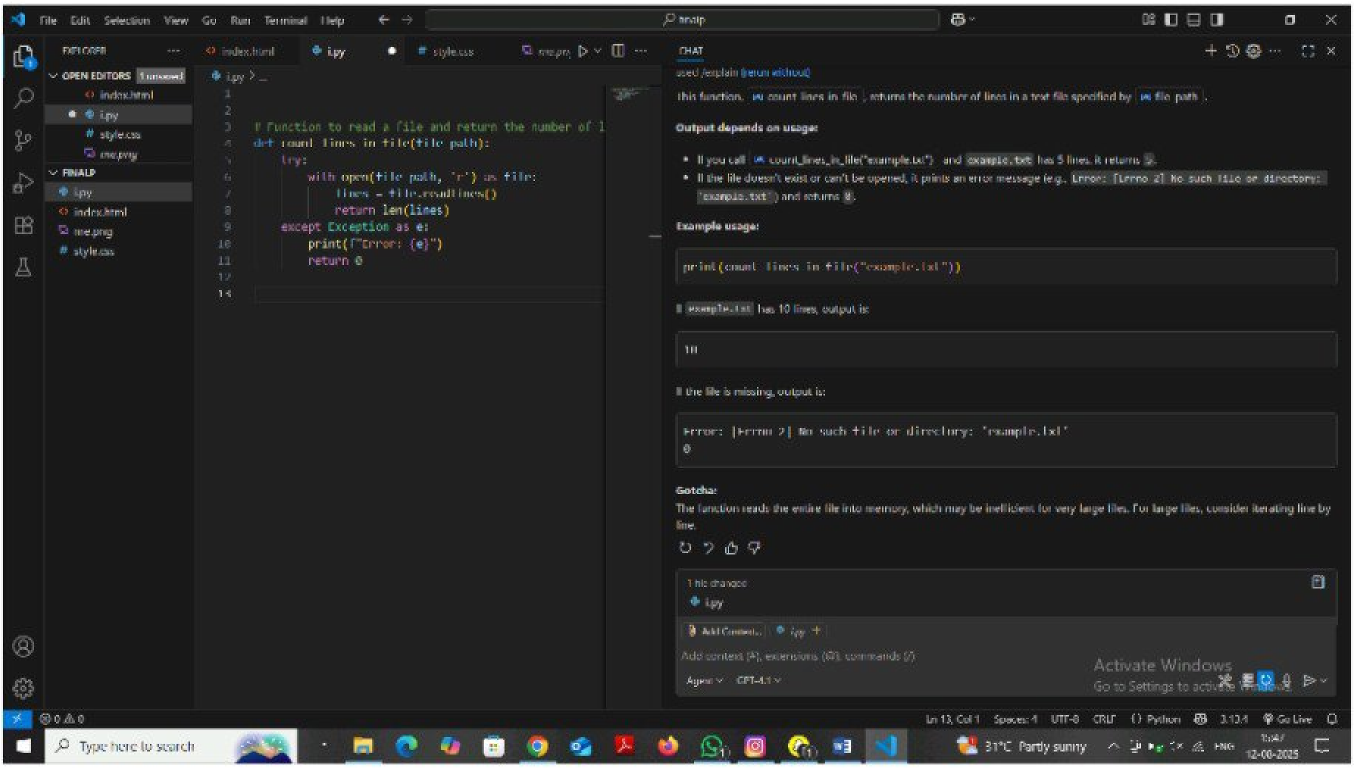
- i.py
- Untitled-3
- Untitled-5
- ind.py

## Task 5:

### Prompt used:

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

# code:



```
# Function to read a file and return the number of l
def count_lines_in_file(file_path):
    try:
        with open(file_path, 'r') as file:
            lines = file.readlines()
            return len(lines)
    except Exception as e:
        print(f"Error: {e}")
        return 0
```

used /explain (rerun without)

This function, count_lines_in_file, returns the number of lines in a text file specified by file_path.

**Output depends on usage:**

- If you call count_lines_in_file("example.txt") and example.txt has 5 lines, it returns 5.
- If the file doesn't exist or can't be opened, it prints an error message (e.g., Error: [Errno 2] No such file or directory: 'example.txt') and returns 0.

**Example usage:**

```
print(count_lines_in_file("example.txt"))
```

If example.txt has 10 lines, output is:

```
10
```

If the file is missing, output is:

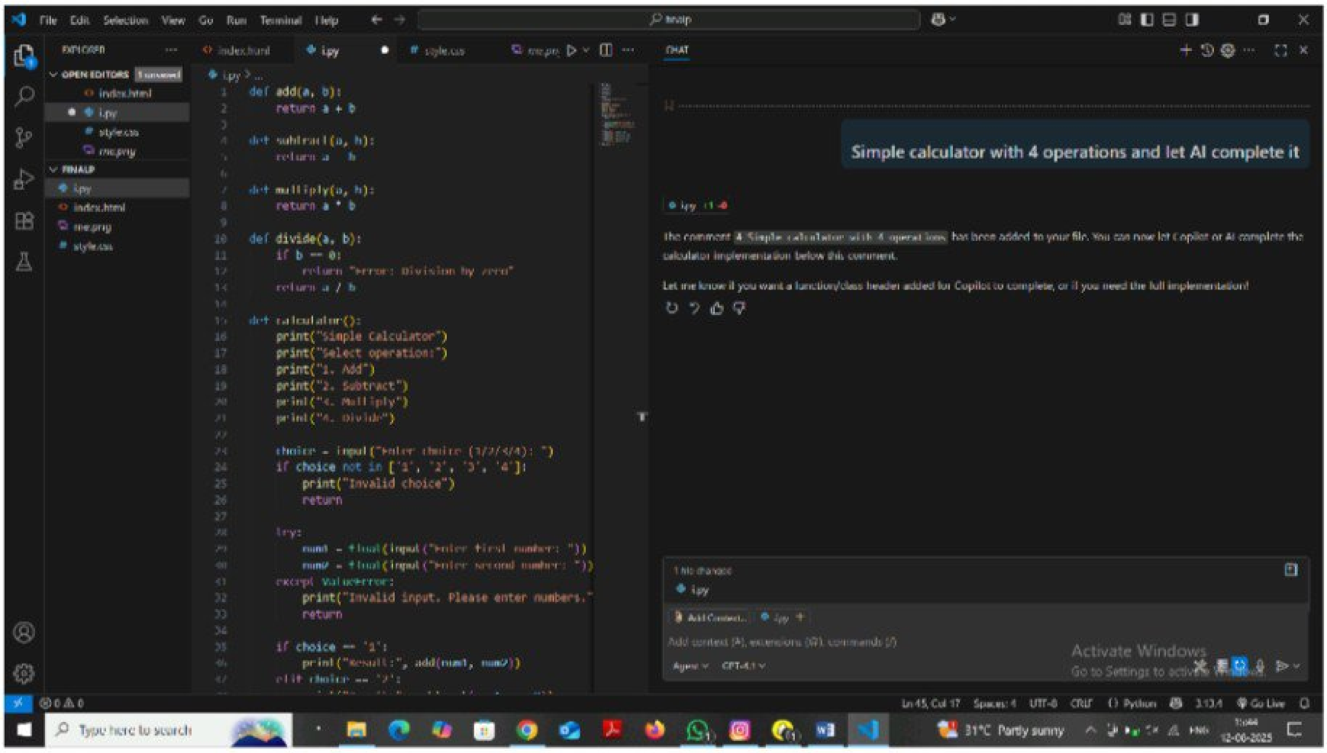
```
Error: [Errno 2] No such file or directory: 'example.txt'
0
```

**Gotcha:**
The function reads the entire file into memory, which may be inefficient for very large files. For large files, consider iterating line by line.

1 file changed
i.py

# 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.

Code:

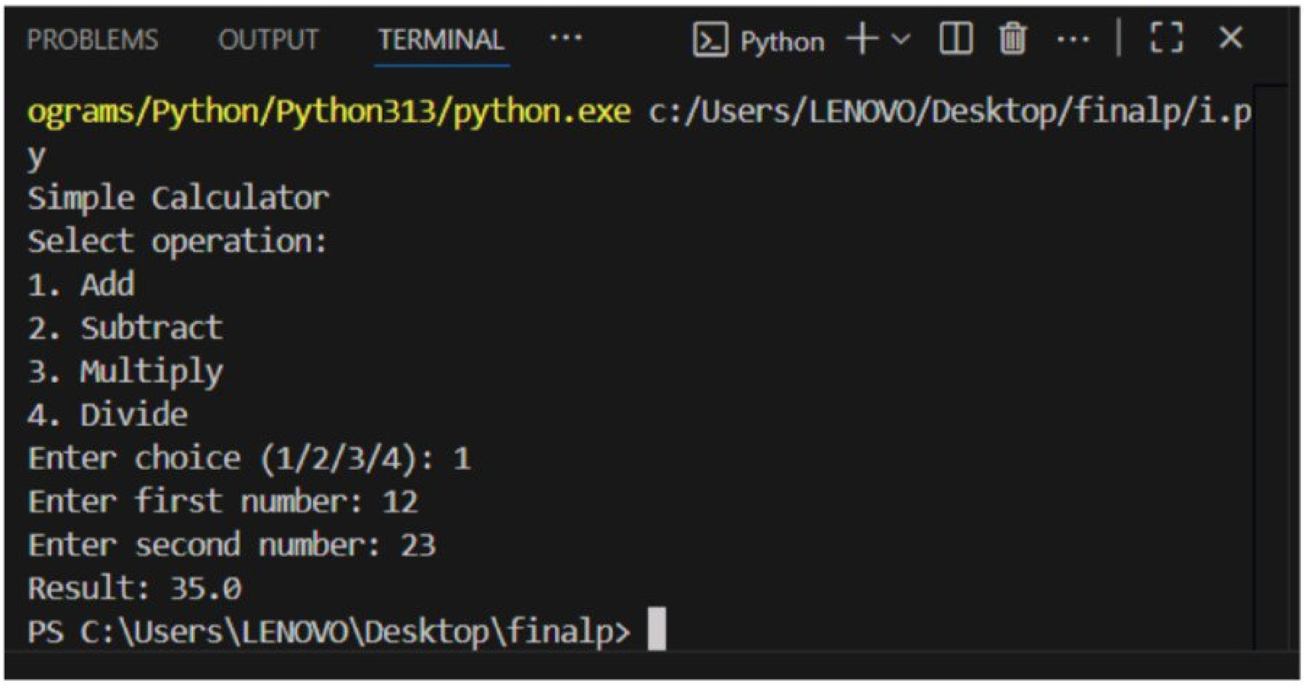


```python
def add(a, b):
    return a + b

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

def multiply(a, b):
    return a * b

def divide(a, b):
    if b == 0:
        return "Error: Division by zero"
    return a / b

def calculator():
    print("Simple Calculator")
    print("Select operation:")
    print("1. Add")
    print("2. Subtract")
    print("3. Multiply")
    print("4. Divide")

    choice = input("Enter choice (1/2/3/4): ")
    if choice not in ['1', '2', '3', '4']:
        print("Invalid choice")
        return

    try:
        num1 = float(input("Enter first number: "))
        num2 = float(input("Enter second number: "))
    except ValueError:
        print("Invalid input. Please enter numbers."
        return

    if choice == '1':
        print("Result:", add(num1, num2))
    elif choice == '2':
```

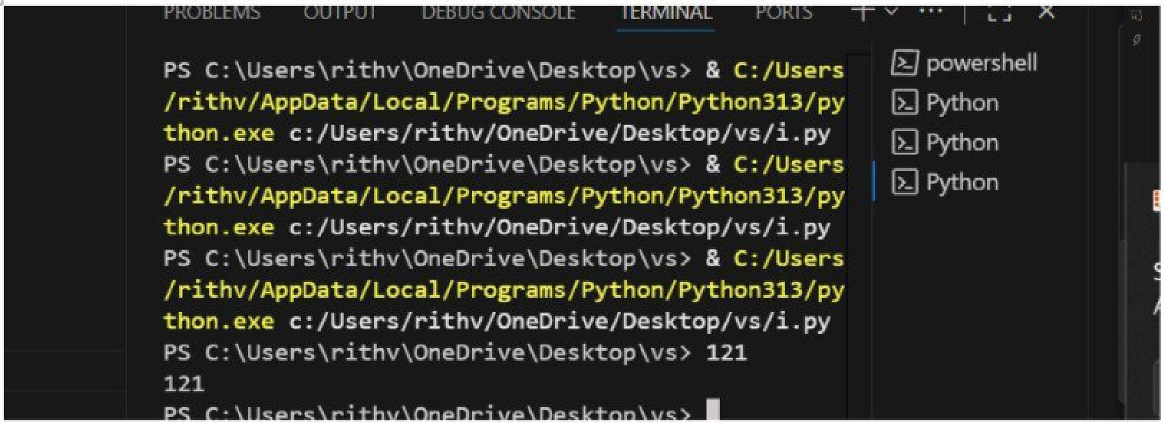
Simple calculator with 4 operations and let AI complete it

The comment # Simple calculator with 4 operations has been added to your file. You can now let Copilot or AI complete the calculator implementation below this comment.

Let me know if you want a function/class header added for Copilot to complete, or if you need the full implementation!



```
ograms/Python/Python313/python.exe c:/Users/LENOVO/Desktop/finalp/i.p
y
Simple Calculator
Select operation:
1. Add
2. Subtract
3. Multiply
4. Divide
Enter choice (1/2/3/4): 1
Enter first number: 12
Enter second number: 23
Result: 35.0
PS C:\Users\LENOVO\Desktop\finalp> 
```



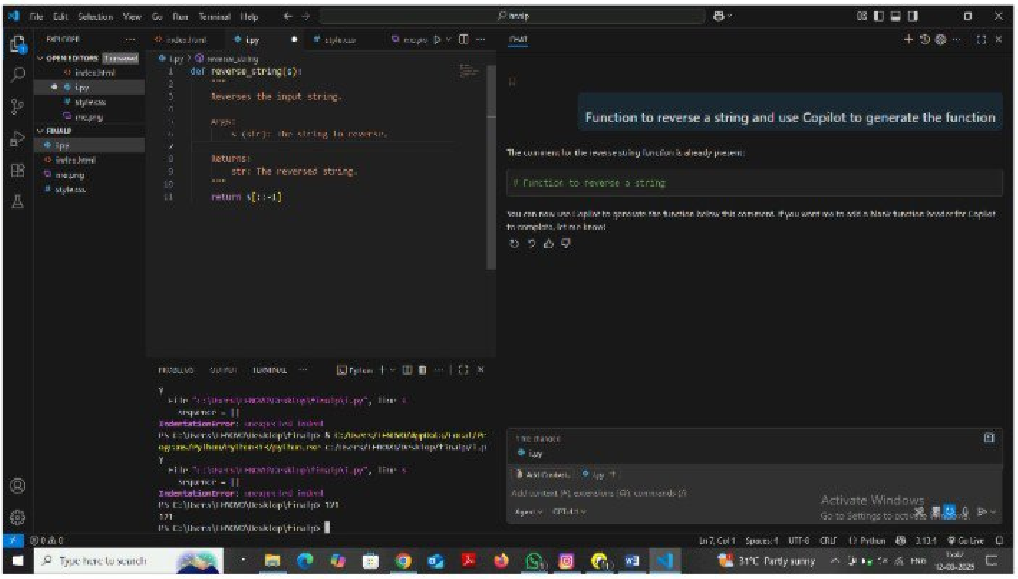
```
PS C:\Users\rithv\OneDrive\Desktop\vs> & C:/Users
/rithv/AppData/Local/Programs/Python/Python313/py
thon.exe c:/Users/rithv/OneDrive/Desktop/vs/i.py
PS C:\Users\rithv\OneDrive\Desktop\vs> & C:/Users
/rithv/AppData/Local/Programs/Python/Python313/py
thon.exe c:/Users/rithv/OneDrive/Desktop/vs/i.py
PS C:\Users\rithv\OneDrive\Desktop\vs> & C:/Users
/rithv/AppData/Local/Programs/Python/Python313/py
thon.exe c:/Users/rithv/OneDrive/Desktop/vs/i.py
PS C:\Users\rithv\OneDrive\Desktop\vs> 121
121
PS C:\Users\rithv\OneDrive\Desktop\vs>
```

## Task 3:

Prompt used:

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

Code:



```python
def reverse_string(s):
    """
    Reverses the input string.

    Args:
        s (str): The string to reverse.

    Returns:
        str: The reversed string.
    """
    return s[::-1]
```

```
    sequence = []
IndentationError: unexpected indent
    sequence = []
IndentationError: unexpected indent
PS C:\Users\[illegible]\Desktop\Final.py> 121
121
PS C:\Users\[illegible]\Desktop\Final.py>
```

Function to reverse a string and use Copilot to generate the function

The comment for the reverse string function is already present:

```
# Function to reverse a string
```

You can now use Copilot to generate the function below this comment. If you want me to add a blank function header for Copilot to complete, let me know!

## Task 4:

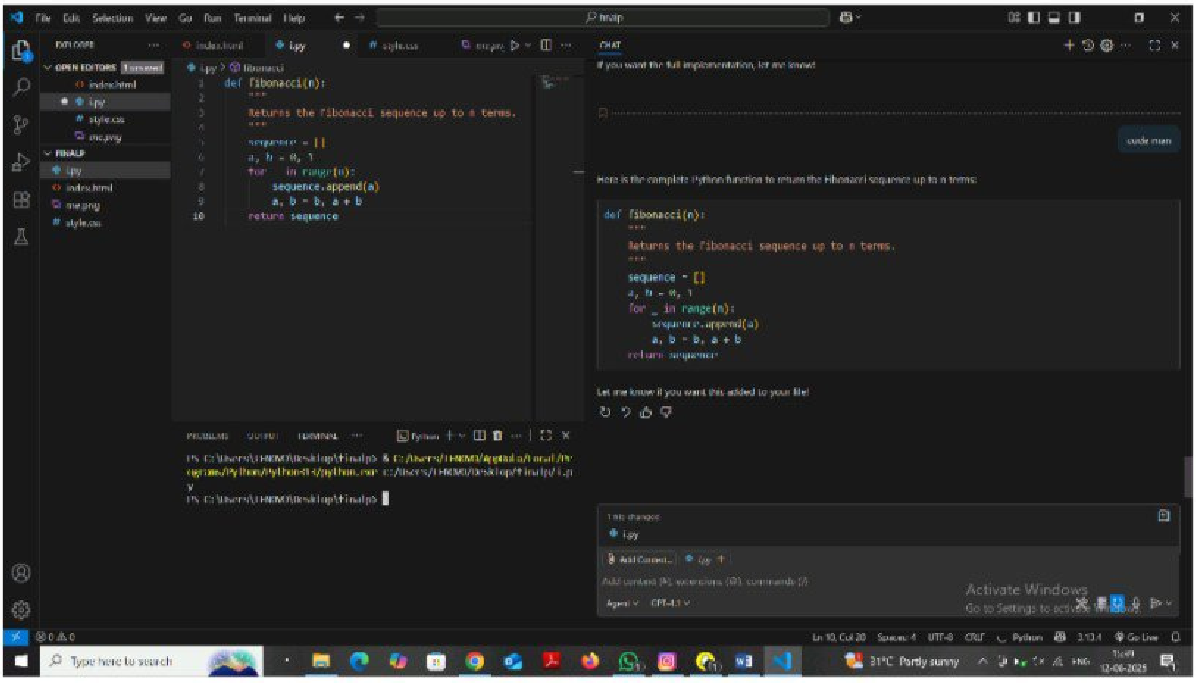
Prompt used:

## Task 2:

## Prompt used:

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

Code:



```python
def fibonacci(n):
    """
    Returns the Fibonacci sequence up to n terms.
    """
    sequence = []
    a, b = 0, 1
    for _ in range(n):
        sequence.append(a)
        a, b = b, a + b
    return sequence
```

```
PS C:\Users\LENOVO\Desktop\finalp> & C:/Users/LENOVO/AppData/Local/Programs/Python/Python313/python.exe c:/Users/LENOVO/Desktop/finalp/1.py
PS C:\Users\LENOVO\Desktop\finalp> 
```

If you want the full implementation, let me know!

code men

Here is the complete Python function to return the Fibonacci sequence up to n terms:

```python
def fibonacci(n):
    """
    Returns the Fibonacci sequence up to n terms.
    """
    sequence = []
    a, b = 0, 1
    for _ in range(n):
        sequence.append(a)
        a, b = b, a + b
    return sequence
```

Let me know if you want this added to your file!

1 file changed
1.py
Add Context...
1.py
Add context (#), extensions (@), commands (/)
Agent
GPT-4.1
Activate Windows
Go to Settings to activate Windows.
Ln 10, Col 20 Spaces: 4 UTF-8 CRLF Python 3.13.4 Go Live
Type here to search
31°C Partly sunny
ENG
12-06-2025