ASSIGNMENT-4.3

2303A51546

BATCH-10

❖ TASK-1:

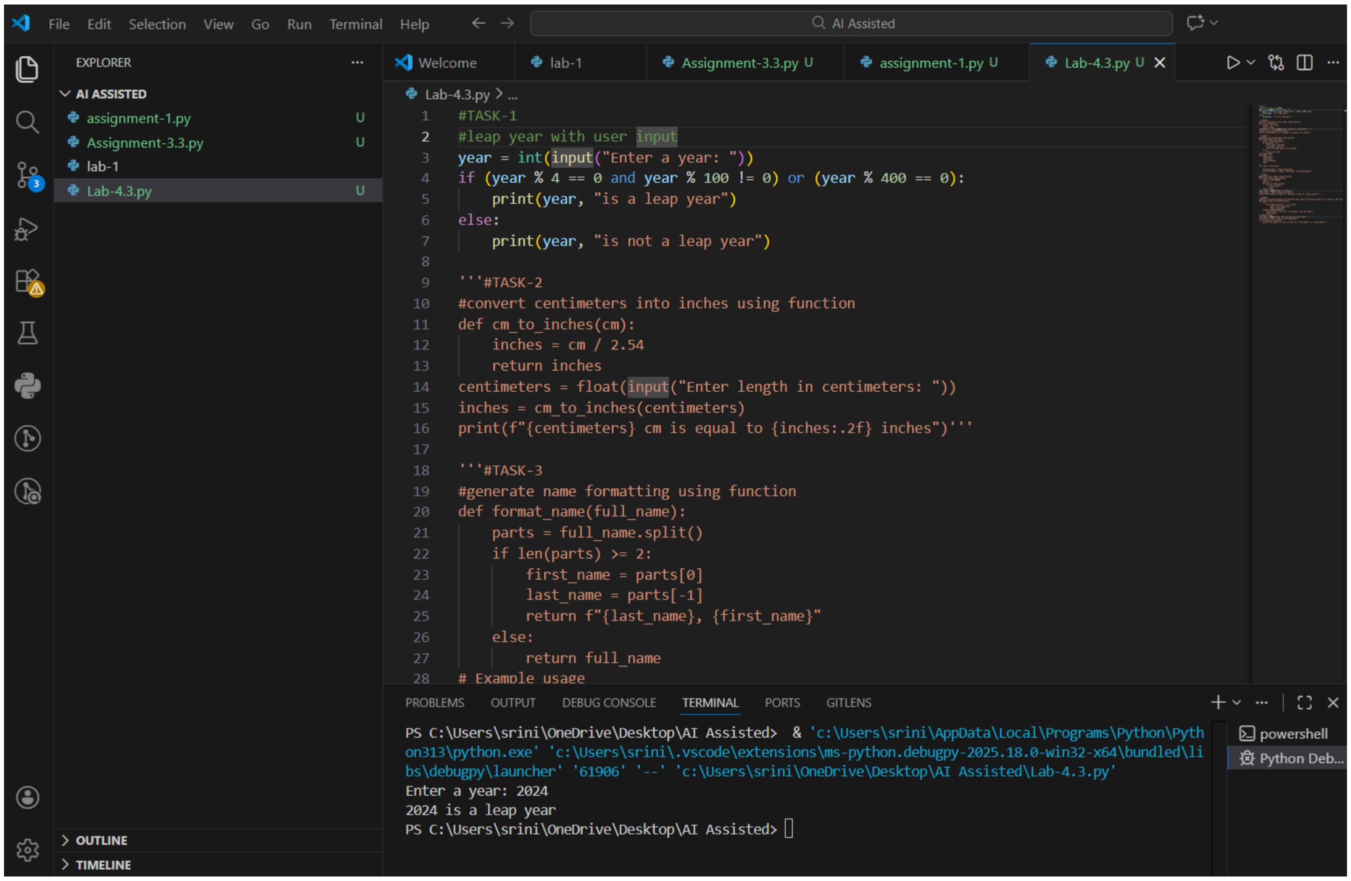
PROMPT:

LEAP YEAR WITH USER INPUT

CODE:

```
year = int(input("Enter a year: "))
if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):
print(year, "is a leap year")
else:
print(year, "is not a leap year")
```

OUTPUT:



Lab-4.3.py > ...

```
#TASK-1
#leap year with user input
year = int(input("Enter a year: "))
if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):
    print(year, "is a leap year")
else:
    print(year, "is not a leap year")

'''#TASK-2
#convert centimeters into inches using function
def cm_to_inches(cm):
    inches = cm / 2.54
    return inches
centimeters = float(input("Enter length in centimeters: "))
inches = cm_to_inches(centimeters)
print(f"{centimeters} cm is equal to {inches:.2f} inches")'''

'''#TASK-3
#generate name formatting using function
def format_name(full_name):
    parts = full_name.split()
    if len(parts) >= 2:
        first_name = parts[0]
        last_name = parts[-1]
        return f"{last_name}, {first_name}"
    else:
        return full_name
# Example usage
```

```
PS C:\Users\srini\OneDrive\Desktop\AI Assisted>  & 'c:\Users\srini\AppData\Local\Programs\Python\Pyth
on313\python.exe' 'c:\Users\srini\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\li
bs\debugpy\launcher' '61906' '--' 'c:\Users\srini\OneDrive\Desktop\AI Assisted\Lab-4.3.py'
Enter a year: 2024
2024 is a leap year
PS C:\Users\srini\OneDrive\Desktop\AI Assisted> 
```

EXPLANATION:

- The program asks the user to type a year and stores it as a number.
- It checks the leap year rule:

divisible by 4 **and not** divisible by 100, **or** divisible by 400.

- If the rule is true, it prints that the year is a leap year.
- If not, it prints that the year is not a leap year.

❖ TASK-2:

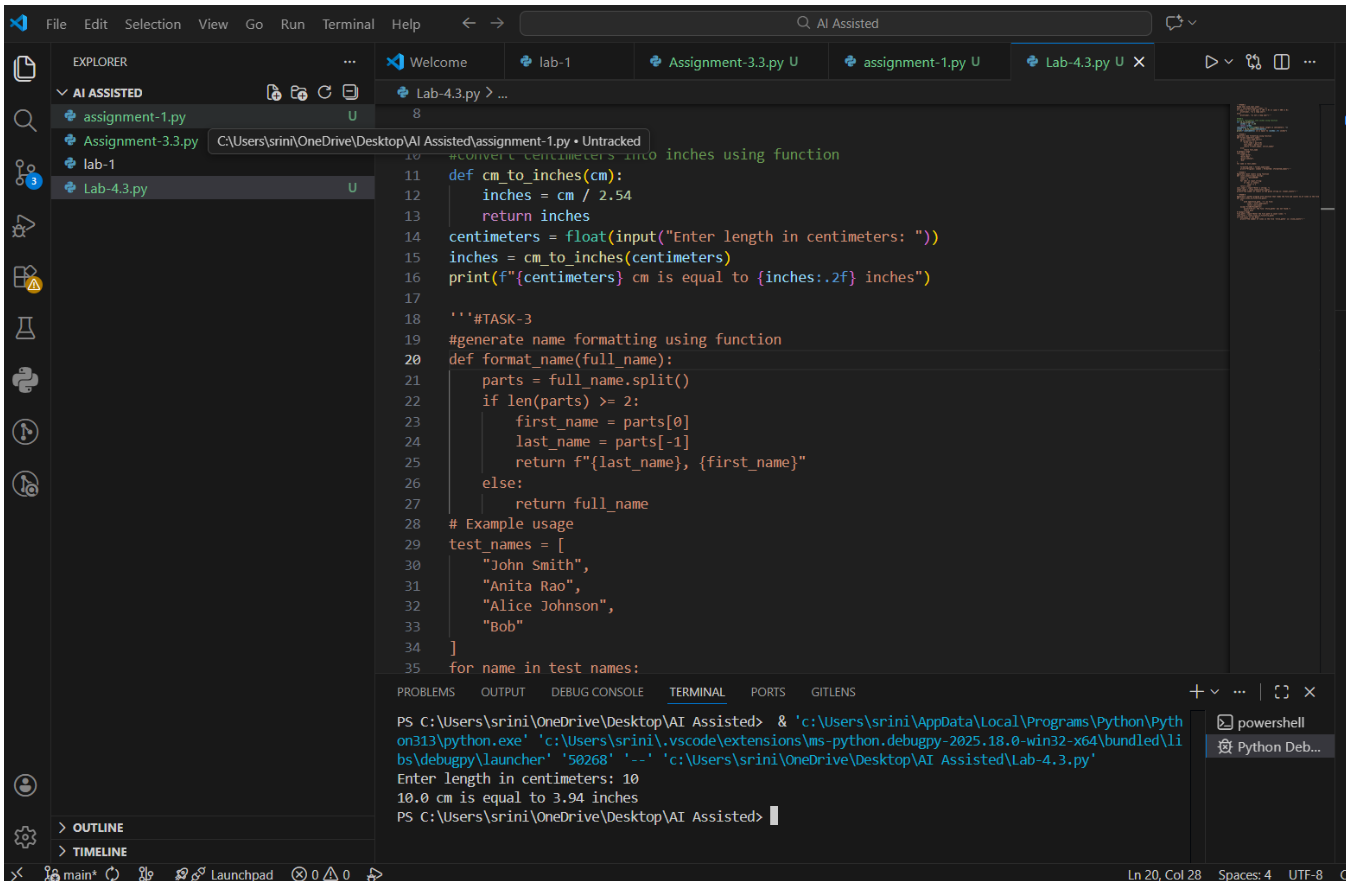
PROMPT:

convert centimeters into inches using function

CODE:

```
def cm_to_inches(cm):
    inches = cm / 2.54
    return inches
centimeters = float(input("Enter length in centimeters: "))
inches = cm_to_inches(centimeters)
print(f"{centimeters} cm is equal to {inches:.2f} inches")
```

OUTPUT:



```
#convert centimeters into inches using function
def cm_to_inches(cm):
    inches = cm / 2.54
    return inches
centimeters = float(input("Enter length in centimeters: "))
inches = cm_to_inches(centimeters)
print(f"{centimeters} cm is equal to {inches:.2f} inches")

'''#TASK-3
#generate name formatting using function
def format_name(full_name):
    parts = full_name.split()
    if len(parts) >= 2:
        first_name = parts[0]
        last_name = parts[-1]
        return f"{last_name}, {first_name}"
    else:
        return full_name
# Example usage
test_names = [
    "John Smith",
    "Anita Rao",
    "Alice Johnson",
    "Bob"
]
for name in test names:
```

```
PS C:\Users\srini\OneDrive\Desktop\AI Assisted>  & 'c:\Users\srini\AppData\Local\Programs\Python\Pyth
on313\python.exe' 'c:\Users\srini\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\li
bs\debugpy\launcher' '50268' '--' 'c:\Users\srini\OneDrive\Desktop\AI Assisted\Lab-4.3.py'
Enter length in centimeters: 10
10.0 cm is equal to 3.94 inches
PS C:\Users\srini\OneDrive\Desktop\AI Assisted>
```

EXPLANATION:

- The function `cm_to_inches(cm)` uses the formula `cm / 2.54` to convert centimeters into inches.
- The program asks the user to enter a length in centimeters and stores it as a floating-point number.
- That value is passed into the function, which calculates and returns the result in inches.
- Finally, the program prints the conversion result, showing the answer rounded to two decimal places.

❖ TASK-3:

PROMPT:

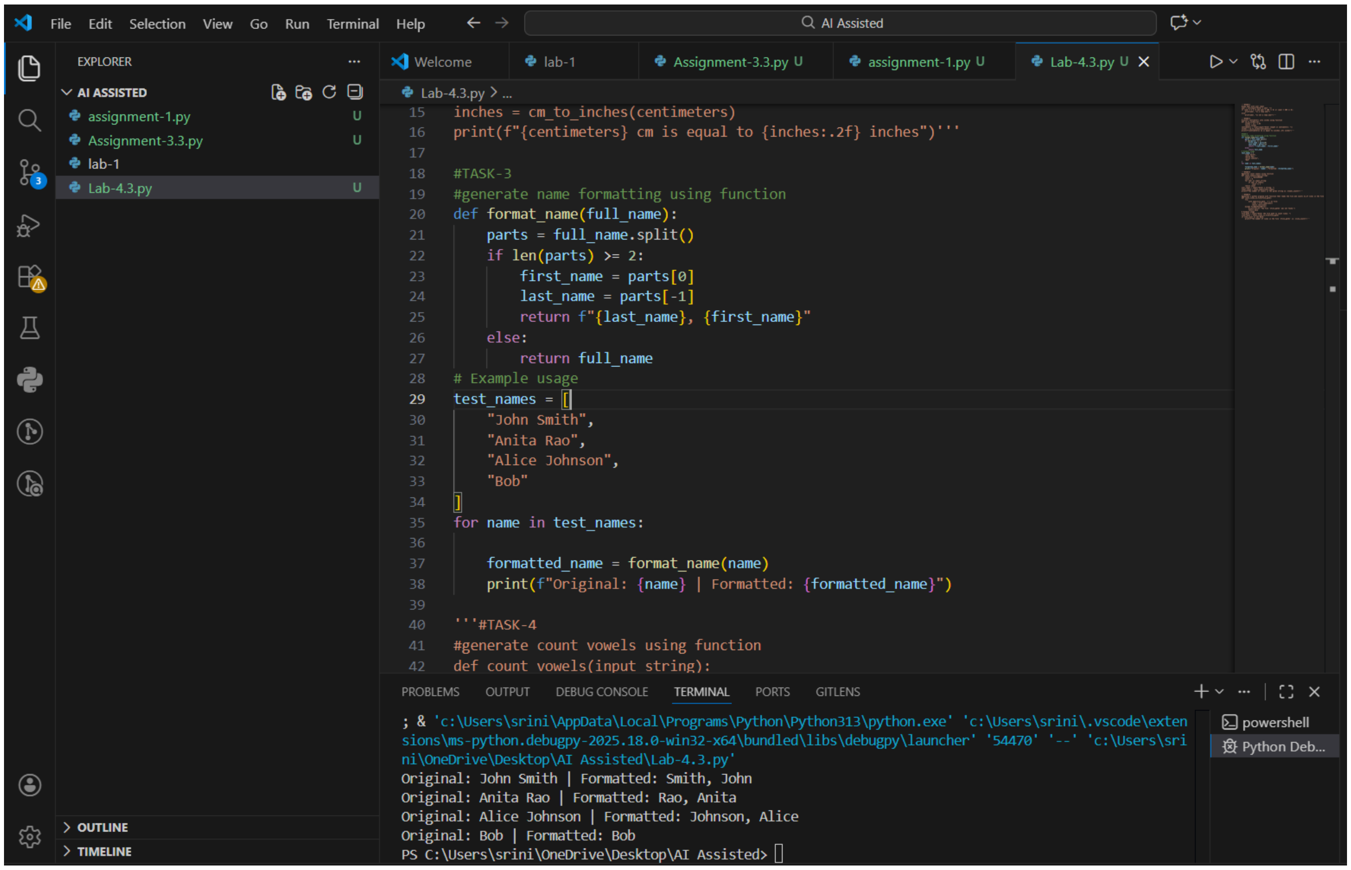
Generate name formatting using function

CODE:

```
def format_name(full_name):
    parts = full_name.split()
    if len(parts) >= 2:
        first_name = parts[0]
        last_name = parts[-1]
        return f"{last_name}, {first_name}"
    else:
        return full_name
# Example usage
test_names = [
    "John Smith",
    "Anita Rao",
    "Alice Johnson",
    "Bob"
]
for name in test_names:

    formatted_name = format_name(name)
    print(f"Original: {name} | Formatted: {formatted_name}")
```

OUTPUT:



```python
inches = cm_to_inches(centimeters)
print(f"{centimeters} cm is equal to {inches:.2f} inches")'''

#TASK-3
#generate name formatting using function
def format_name(full_name):
    parts = full_name.split()
    if len(parts) >= 2:
        first_name = parts[0]
        last_name = parts[-1]
        return f"{last_name}, {first_name}"
    else:
        return full_name
# Example usage
test_names = [
    "John Smith",
    "Anita Rao",
    "Alice Johnson",
    "Bob"
]
for name in test_names:

    formatted_name = format_name(name)
    print(f"Original: {name} | Formatted: {formatted_name}")

'''#TASK-4
#generate count vowels using function
def count vowels(input string):
```

```
; & 'c:\Users\srini\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\srini\.vscode\exten
sions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '54470' '--' 'c:\Users\sri
ni\OneDrive\Desktop\AI Assisted\Lab-4.3.py'
Original: John Smith | Formatted: Smith, John
Original: Anita Rao | Formatted: Rao, Anita
Original: Alice Johnson | Formatted: Johnson, Alice
Original: Bob | Formatted: Bob
PS C:\Users\srini\OneDrive\Desktop\AI Assisted>
```

EXPLANATION:

- The function `format_name(full_name)` splits the name into words using spaces.
- If there are at least two words, it takes the first word as the first name and the last word as the last name, then rearranges them into `"LastName, FirstName"`.
- If there is only one word (like "Bob"), it just returns the same name without changes.
- The program tests this function on a list of names and prints both the original and formatted versions.

❖ TASK-4:

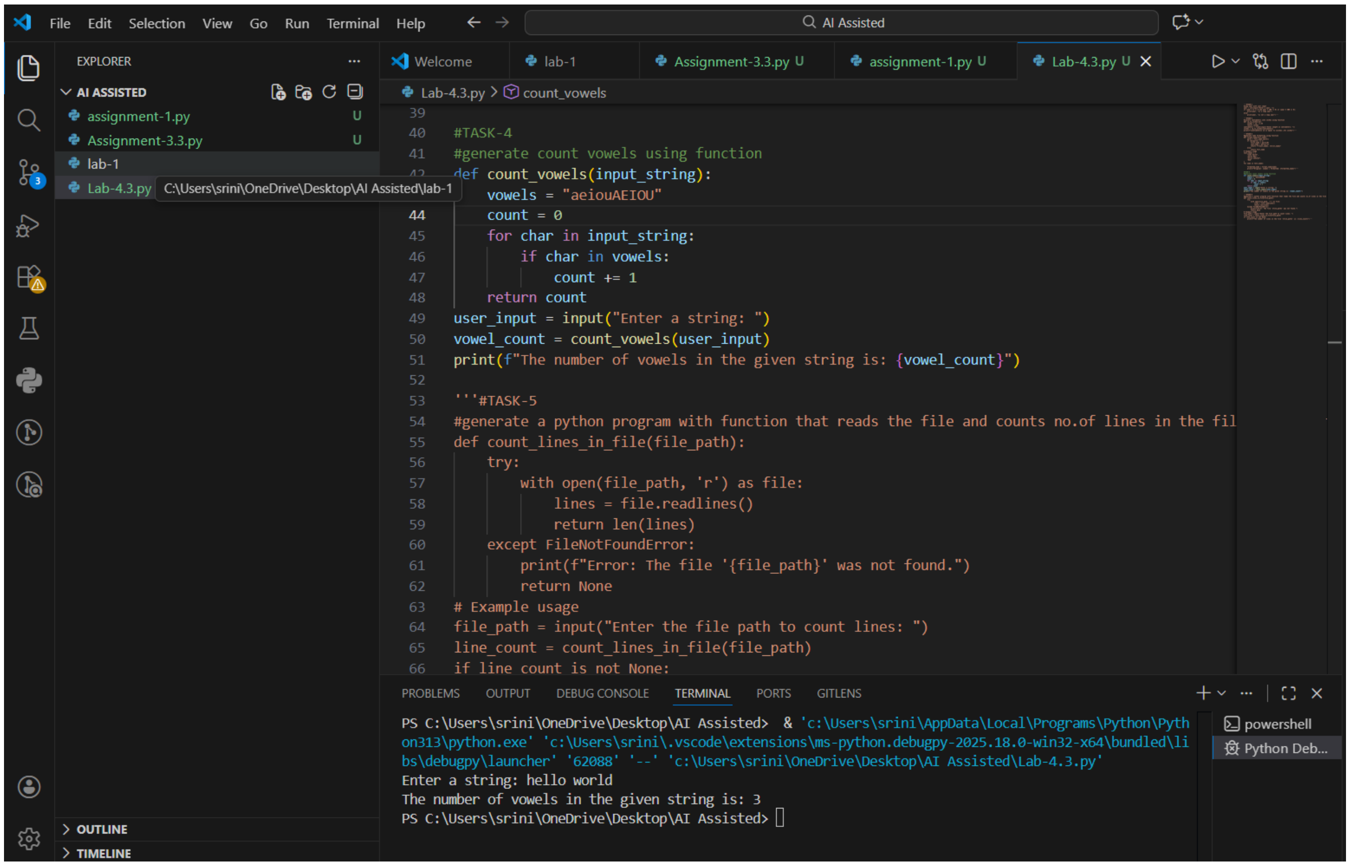
PROMPT:
Generate count vowels using function

CODE:

```python
def count_vowels(input_string):
    vowels = "aeiouAEIOU"
    count = 0
    for char in input_string:
        if char in vowels:
            count += 1
    return count
user_input = input("Enter a string: ")
vowel_count = count_vowels(user_input)
```

```
    print(f"The number of vowels in the given string is: {vowel_count}")
```

OUTPUT:


File Edit Selection View Go Run Terminal Help
AI Assisted
EXPLORER
AI ASSISTED
assignment-1.py U
Assignment-3.3.py U
lab-1
Lab-4.3.py C:\Users\srini\OneDrive\Desktop\AI Assisted\lab-1
Welcome
Lab-4.3.py > count_vowels
OUTLINE
TIMELINE
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS
powershell
Python Deb...

```
#TASK-4
#generate count vowels using function
def count_vowels(input_string):
    vowels = "aeiouAEIOU"
    count = 0
    for char in input_string:
        if char in vowels:
            count += 1
    return count
user_input = input("Enter a string: ")
vowel_count = count_vowels(user_input)
print(f"The number of vowels in the given string is: {vowel_count}")

'''#TASK-5
#generate a python program with function that reads the file and counts no.of lines in the fil
def count_lines_in_file(file_path):
    try:
        with open(file_path, 'r') as file:
            lines = file.readlines()
            return len(lines)
    except FileNotFoundError:
        print(f"Error: The file '{file_path}' was not found.")
        return None
# Example usage
file_path = input("Enter the file path to count lines: ")
line_count = count_lines_in_file(file_path)
if line count is not None:
```

```
PS C:\Users\srini\OneDrive\Desktop\AI Assisted>  & 'c:\Users\srini\AppData\Local\Programs\Python\Pyth
on313\python.exe' 'c:\Users\srini\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\li
bs\debugpy\launcher' '62088' '--' 'c:\Users\srini\OneDrive\Desktop\AI Assisted\Lab-4.3.py'
Enter a string: hello world
The number of vowels in the given string is: 3
PS C:\Users\srini\OneDrive\Desktop\AI Assisted> 
```

EXPLANATION:

- The function `count_vowels(input_string)` goes through each character of the given string.
- It checks if the character is a vowel (`a, e, i, o, u` in both uppercase and lowercase).
- If it is a vowel, the counter increases by 1.
- Finally, the program prints the total number of vowels found in the user's input string.

❖ TASK-5:

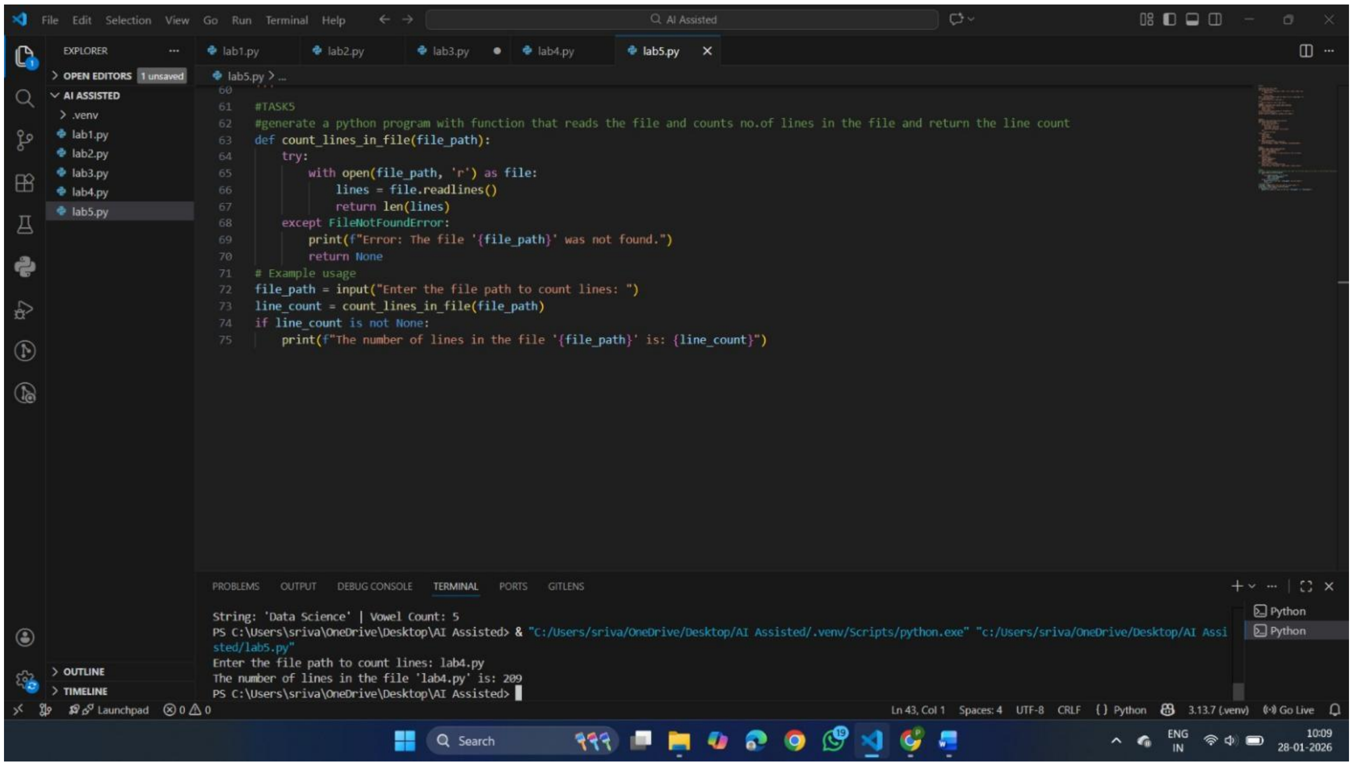
PROMPT:

generate a python program with function that reads the file and counts no.of lines in the file and return the line count

CODE:

```
def count_lines_in_file(file_path):
    try:
        with open(file_path, 'r') as file:
            lines = file.readlines()
            return len(lines)
    except FileNotFoundError:
        print(f"Error: The file '{file_path}' was not found.")
        return None
# Example usage
file_path = input("Enter the file path to count lines: ")
line_count = count_lines_in_file(file_path)
if line_count is not None:
    print(f"The number of lines in the file '{file_path}' is: {line_count}")
```

OUTPUT:



```
#TASK5
#generate a python program with function that reads the file and counts no.of lines in the file and return the line count
def count_lines_in_file(file_path):
    try:
        with open(file_path, 'r') as file:
            lines = file.readlines()
            return len(lines)
    except FileNotFoundError:
        print(f"Error: The file '{file_path}' was not found.")
        return None
# Example usage
file_path = input("Enter the file path to count lines: ")
line_count = count_lines_in_file(file_path)
if line_count is not None:
    print(f"The number of lines in the file '{file_path}' is: {line_count}")
```

```
String: 'Data Science' | Vowel Count: 5
PS C:\Users\sriva\OneDrive\Desktop\AI Assisted> & "C:/Users/sriva/OneDrive/Desktop/AI Assisted/.venv/Scripts/python.exe" "c:/Users/sriva/OneDrive/Desktop/AI Assi
sted/lab5.py"
Enter the file path to count lines: lab4.py
The number of lines in the file 'lab4.py' is: 209
PS C:\Users\sriva\OneDrive\Desktop\AI Assisted>
```

EXPLANATION:

- The function `count_lines_in_file(file_path)` opens the file in read mode and reads all its lines.
- It counts the number of lines using `len(lines)` and returns that value.
- If the file is not found, it shows an error message instead of crashing.
- Finally, the program prints the total number of lines in the file entered by the user.