**AIPP LAB 2**

**Name: MAHVISH ISAQ**

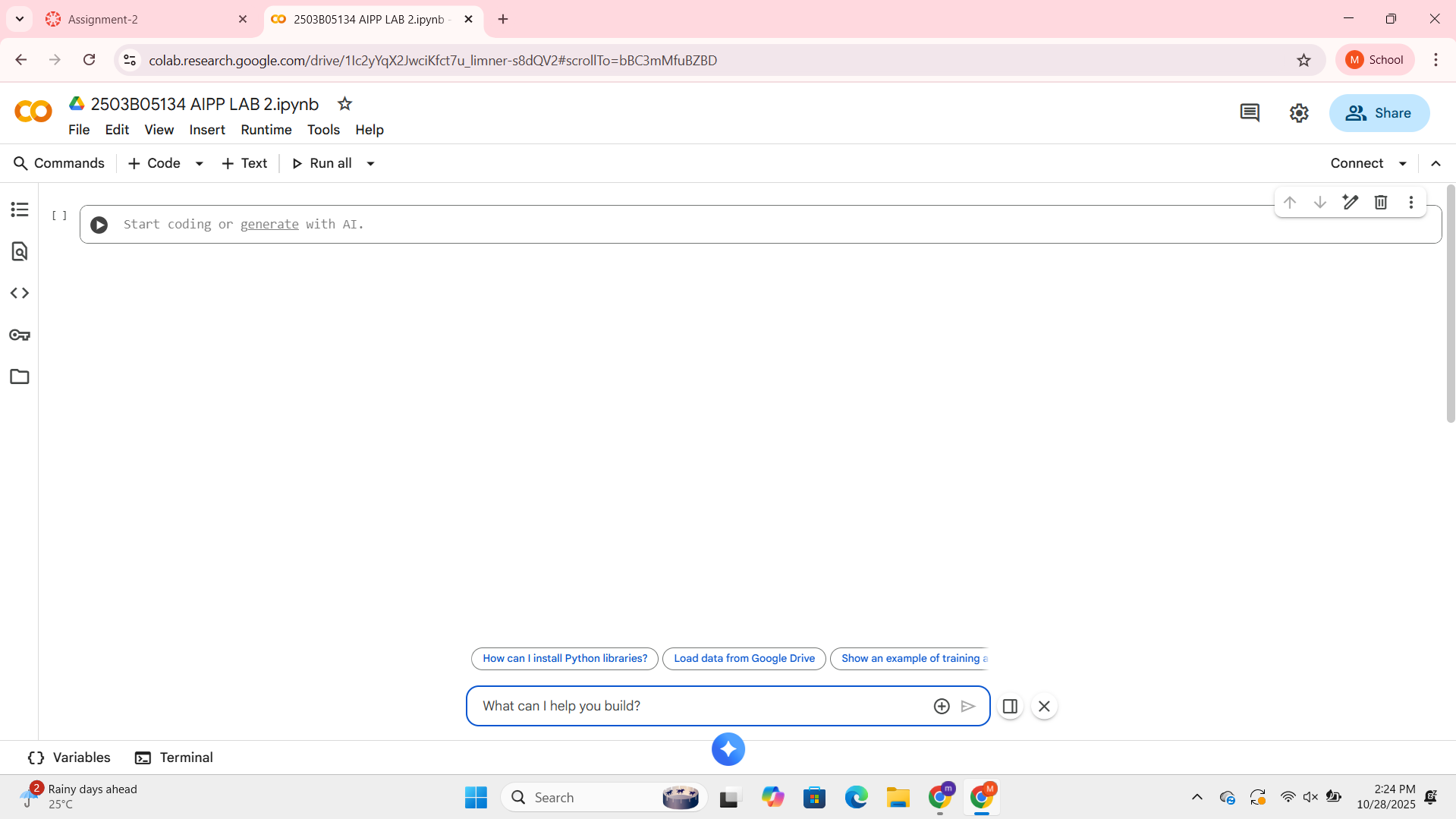
**Hall Ticket No.: 2503B05134**

**Date: 28/10/2025 (Week 1)**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Task 1:**

**Use Google Gemini in Colab to write a function that reads a CSV file and calculates Mean, Min, Max.**



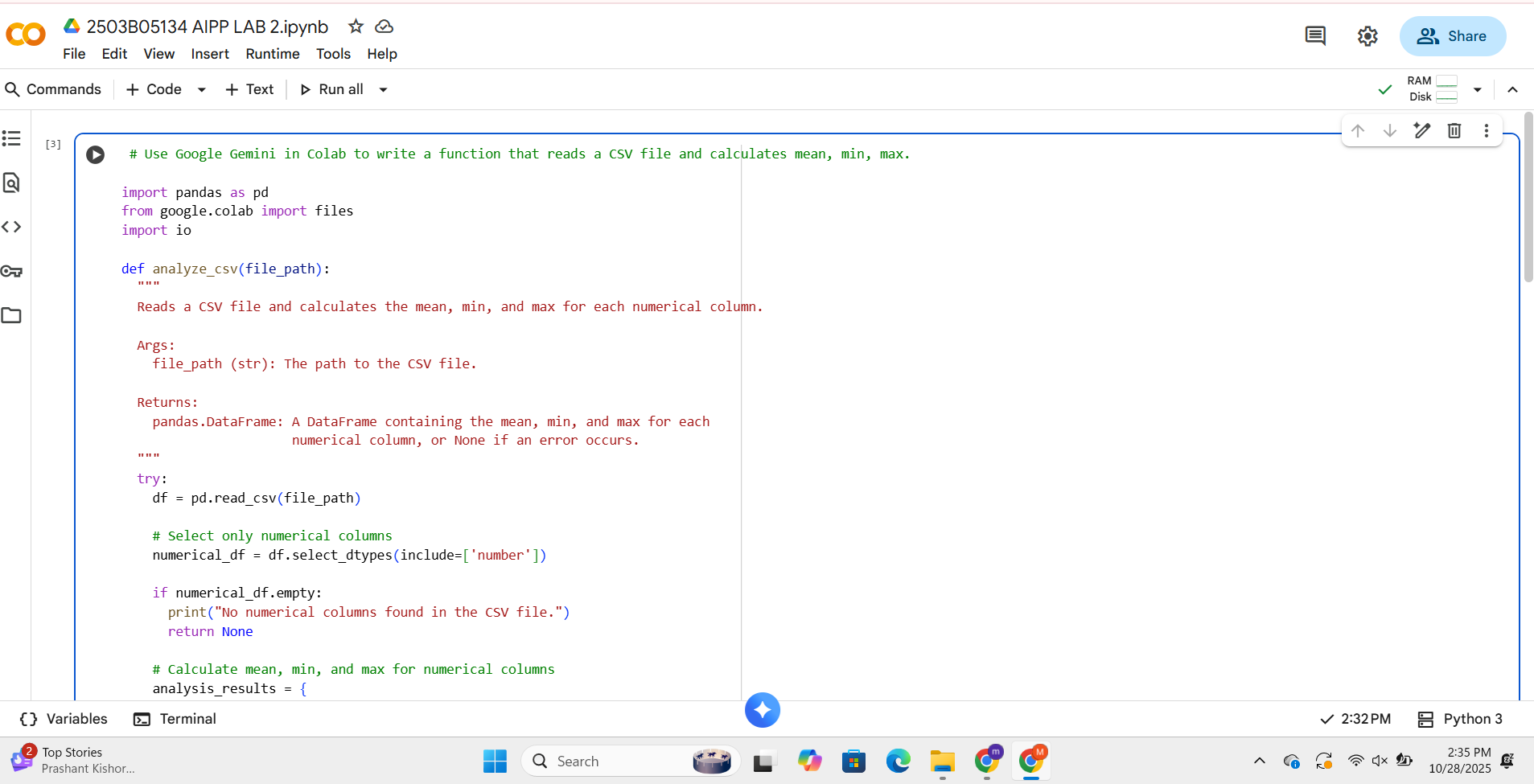
***PROMPTS:***

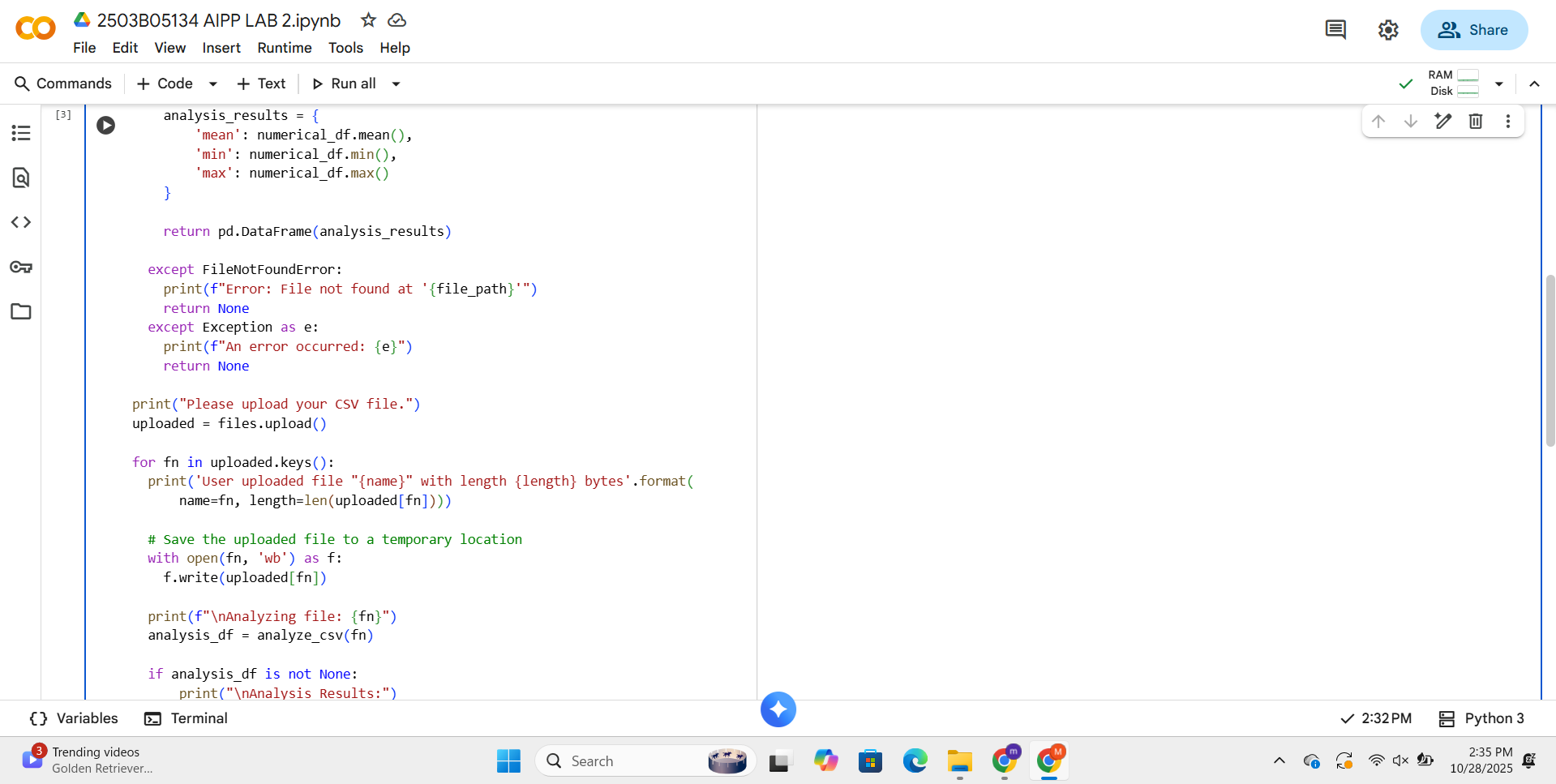
**1) Read csv file and calculate mean, min and max.**

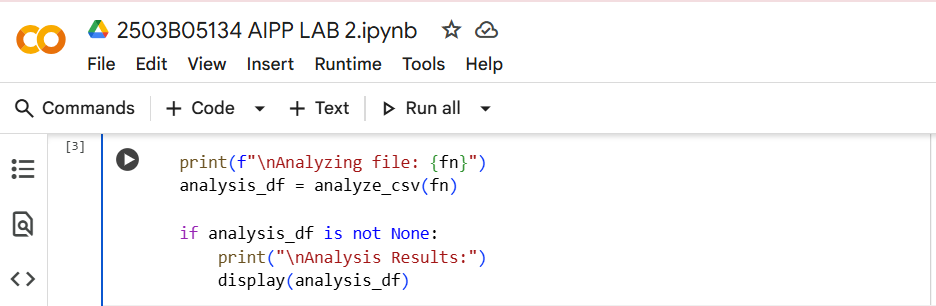
**2) Update code to give options to upload the csv file.**

**3) Give output in a tabular form.**

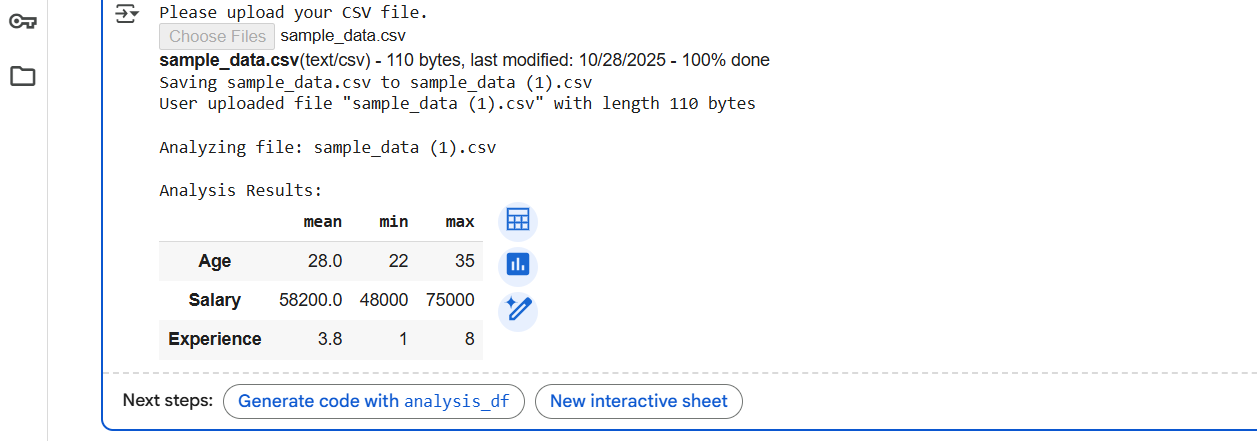
***CODE:***







***OUTPUT:***



**Task 2:**

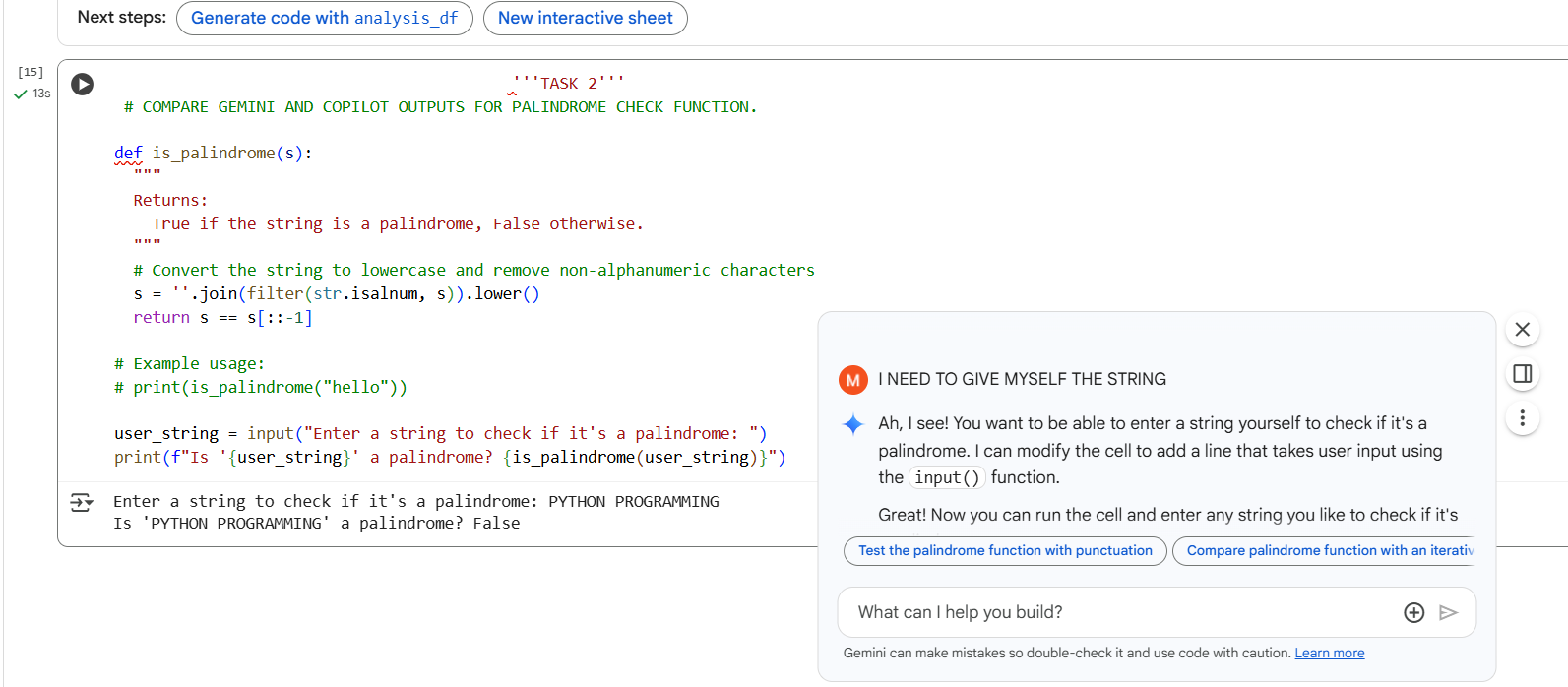
**Compare Gemini and Copilot output for a Palindrome check function.**

***PROMPTS:***

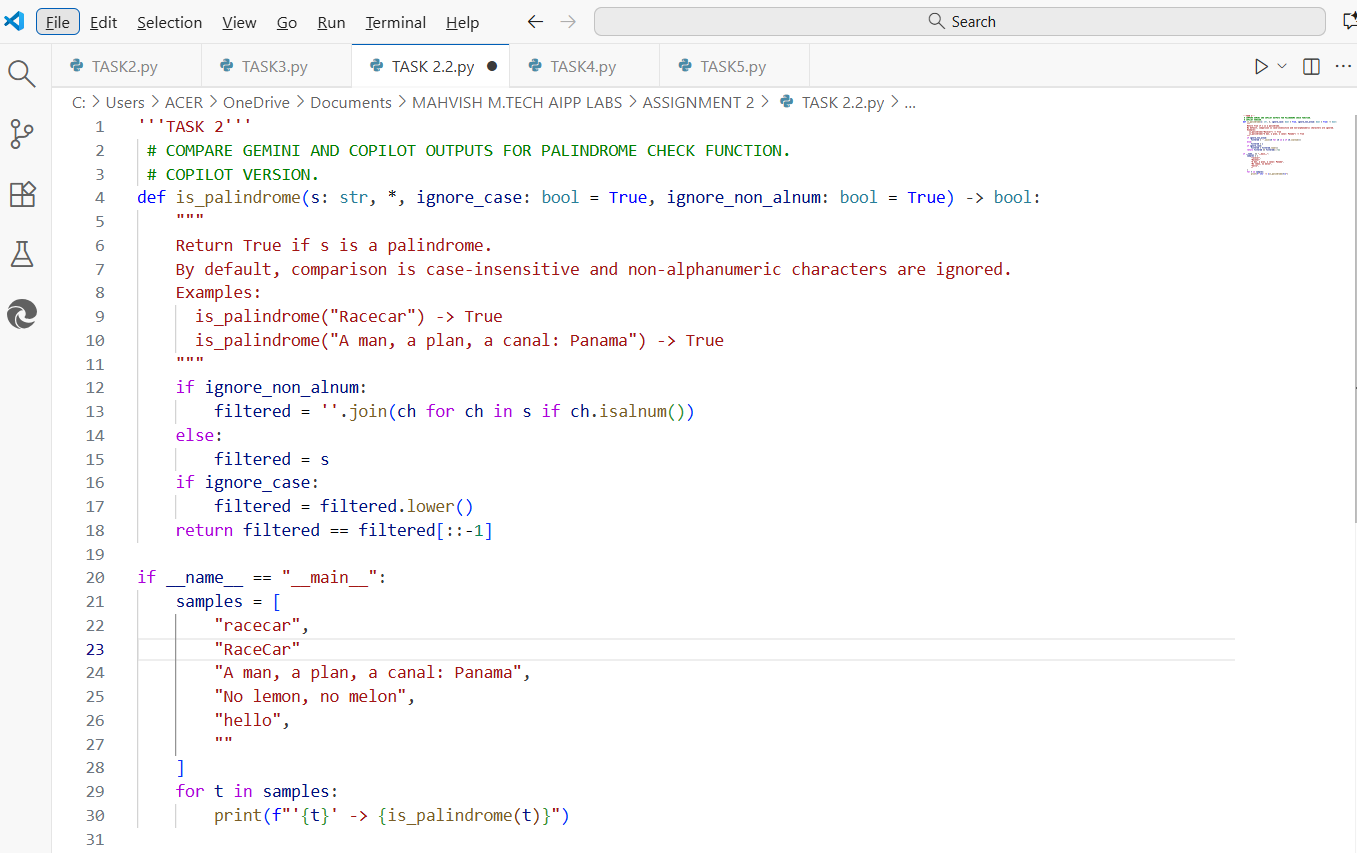
**1) Write function to check if string is Palindrome.**

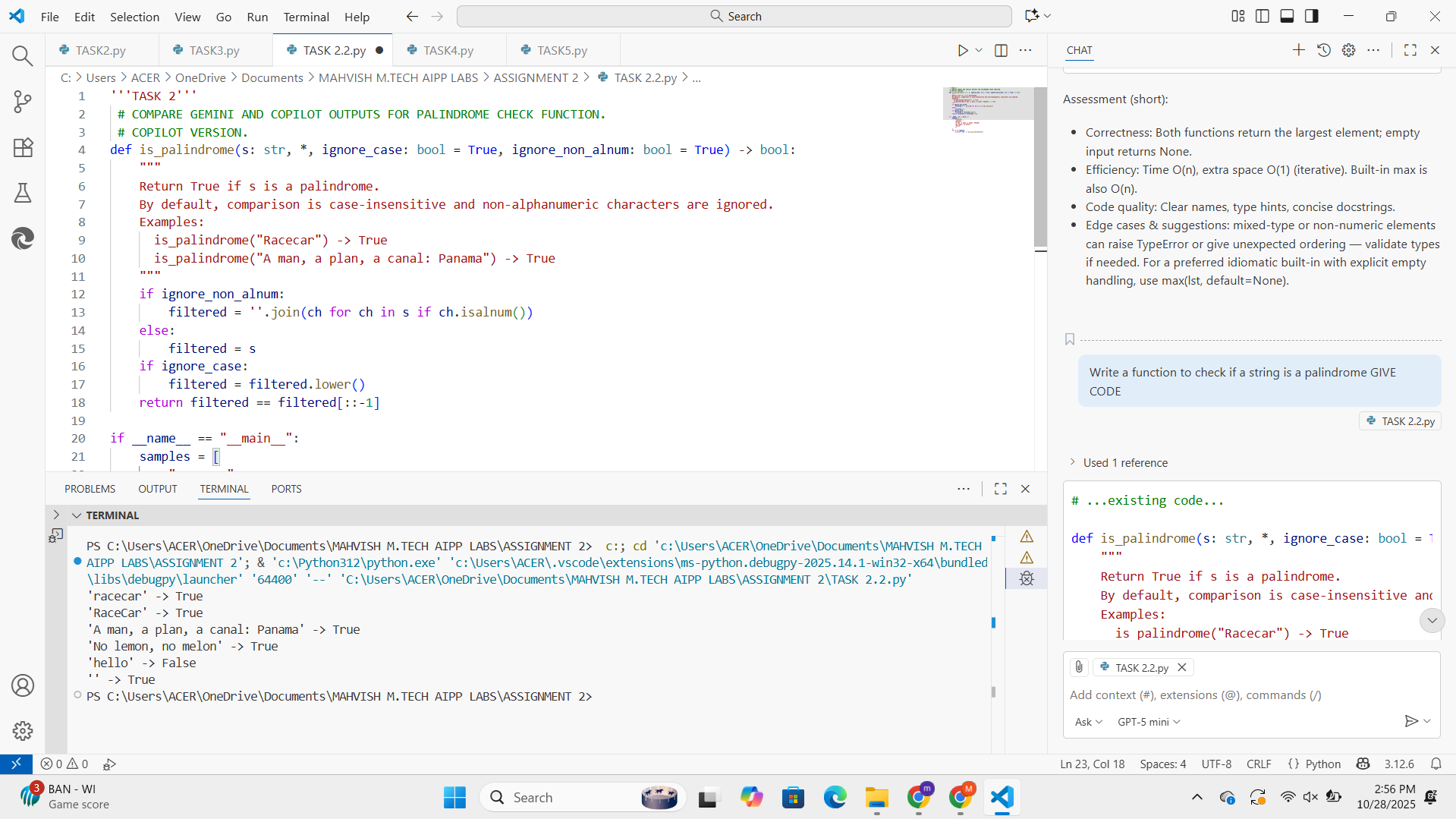
**2) I want to give the string input myself.**

***CODE WITH OUTPUT (GEMINI):***



***CODE WITH OUTPUT (COPILOT):***





***OBSERVATIONS AND COMPARISON:***

**GITHUB COPILOT (VS CODE)**-

More Advanced and Modular

Includes optional arguments for customization.

Can handle case sensitivity and non-alphanumeric filtering independently.

Suitable for reuse in larger programs.

Automatically test multiple strings, showing batch results.

**GEMINI (GOOGLE COLAB)** -

Simpler and User-Friendly

Easy for beginners.

Lets the user enter their own string.

The code is short and minimal.

Good for interactive use or learning, not necessarily for advanced reuse.

**Task 3:**

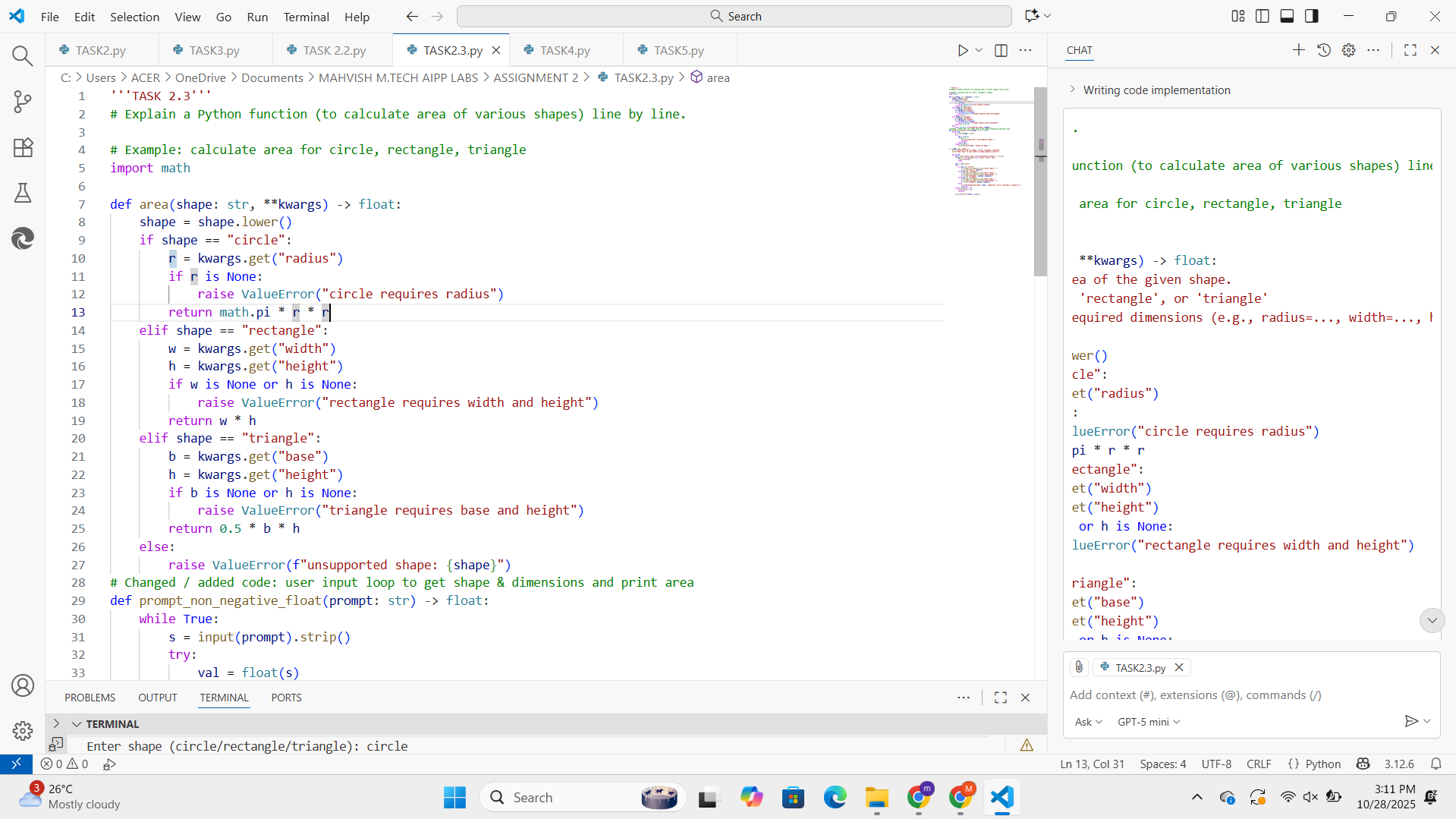
**Ask Gemini to explain a python function (to calculate area of various shapes) line by line.**

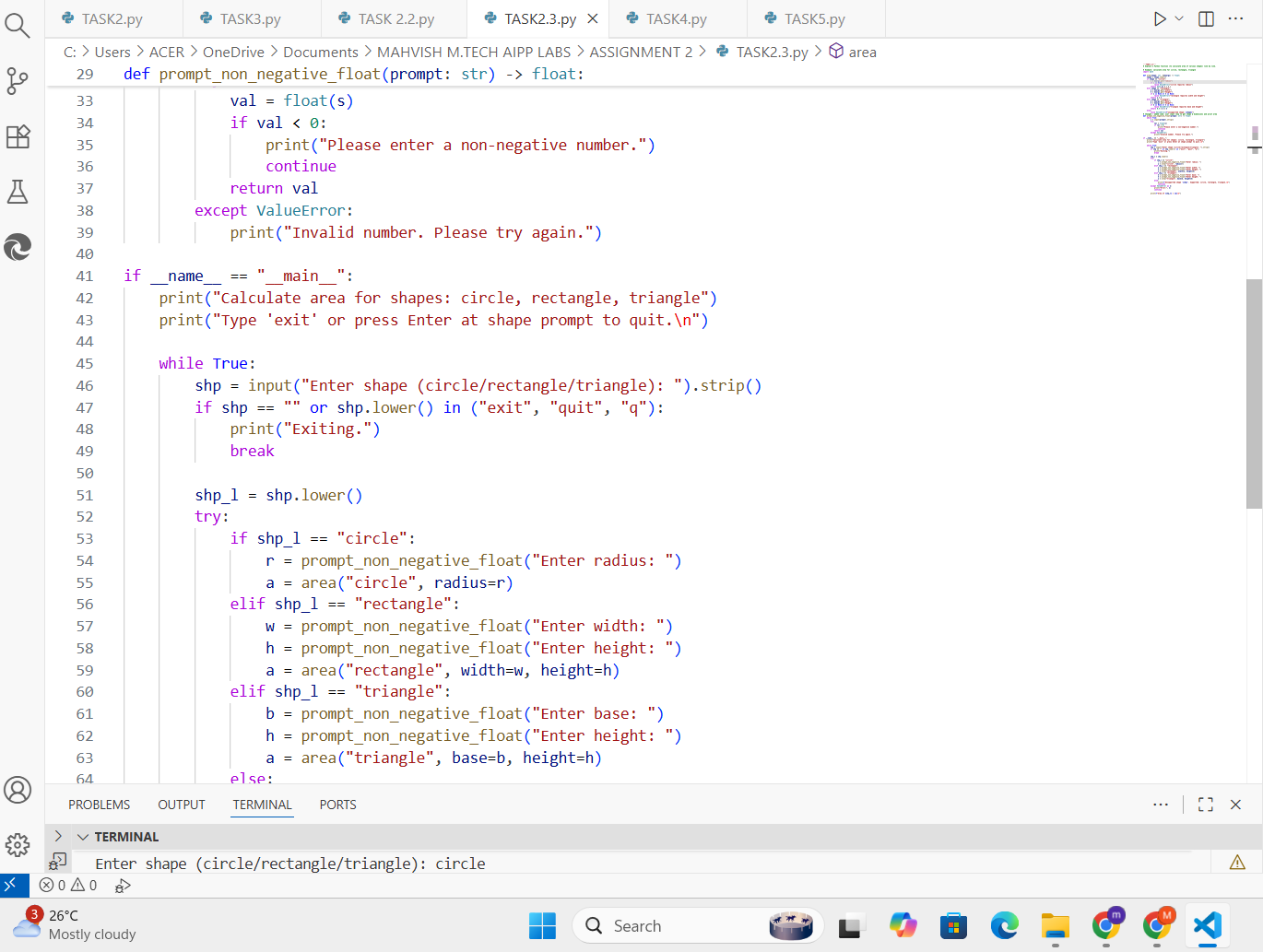
***PROMPTS:***

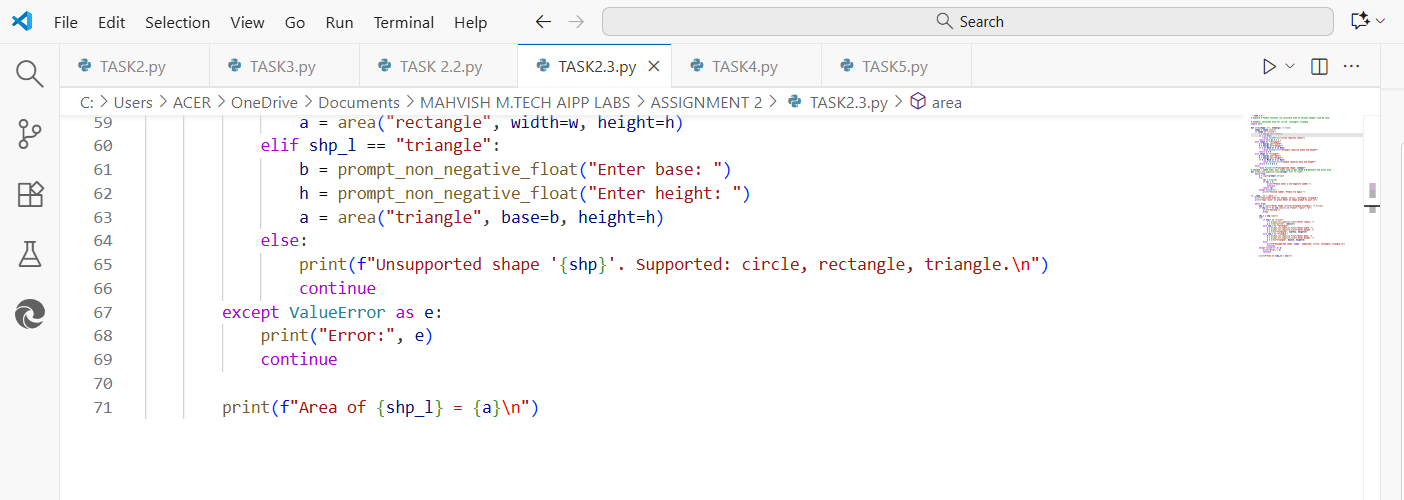
**1) Give Dynamic code for shapes and print the area.**

**2) Explain python functions to calculate the area of shapes.**

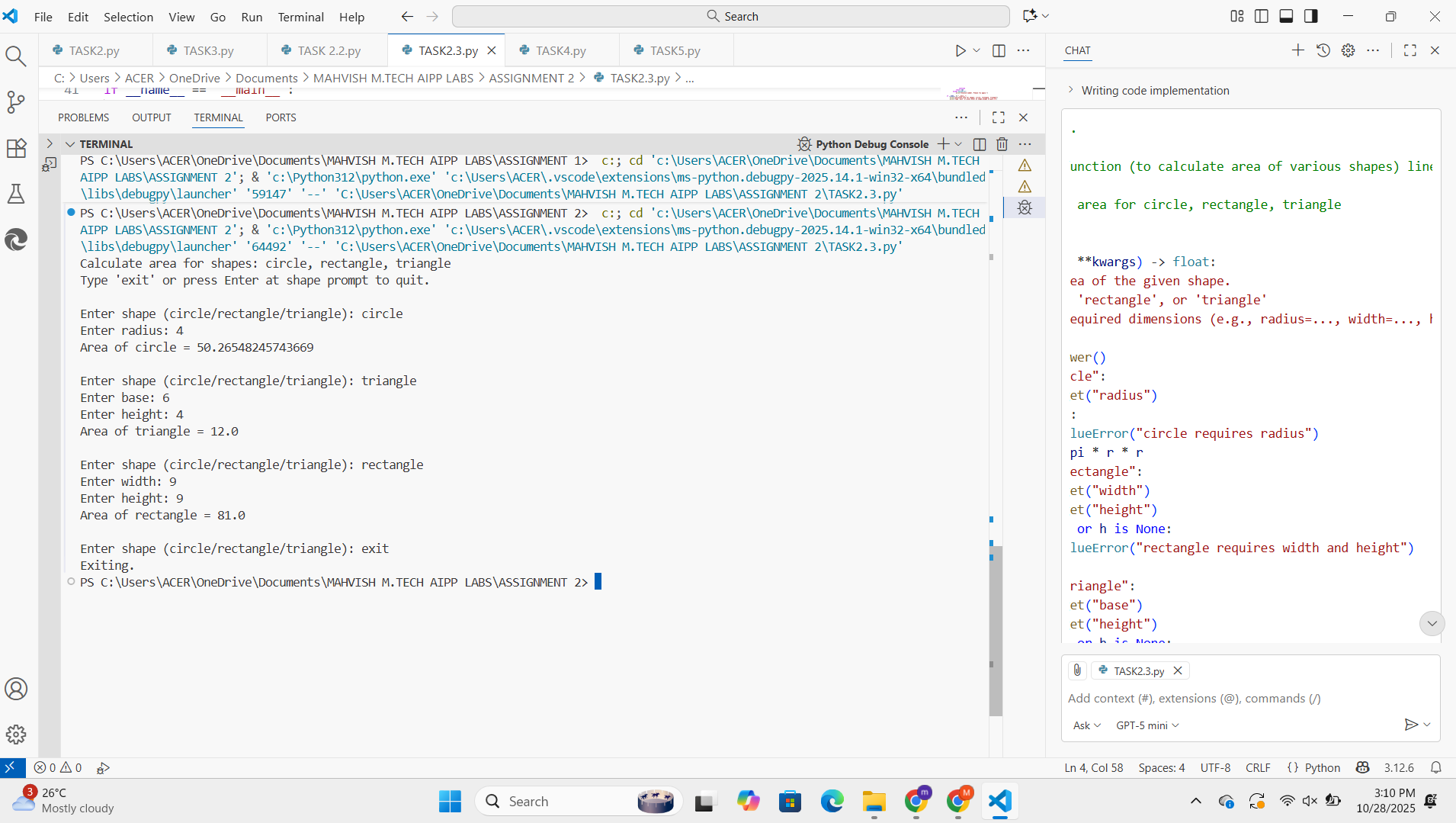
***CODE:***







***OUTPUT:***



***EXPLANATION:***

1) Import math module: Import Python’s standard math module so the code can use constants/functions such as math.pi.

2) Define function area: Declare function area and give hint that *shape* should be a string and \*\*kwargs usage means the function accepts additional keyword arguments (e.g., x=5, w=2, h=4). The computed area is returned as a float.

3) Raise ValueError: Test whether required parameters are missing.

4) Define prompt\_non\_negative\_float: Declare a helpful function that prompts the user and returns a non-negative float.  
 It starts with a loop, so it will continue until a valid non-negative float is entered.

5) if \_\_name\_\_ == "\_\_main\_\_": This is a standard Python module guard.  
 The code under this block runs only when the file is executed directly (not imported).

6) Compute area and display: Compute the area and store it in a variable, then return to the prompt at the end for shape.  
 If everything is successful, print the area.

**Task 4:**

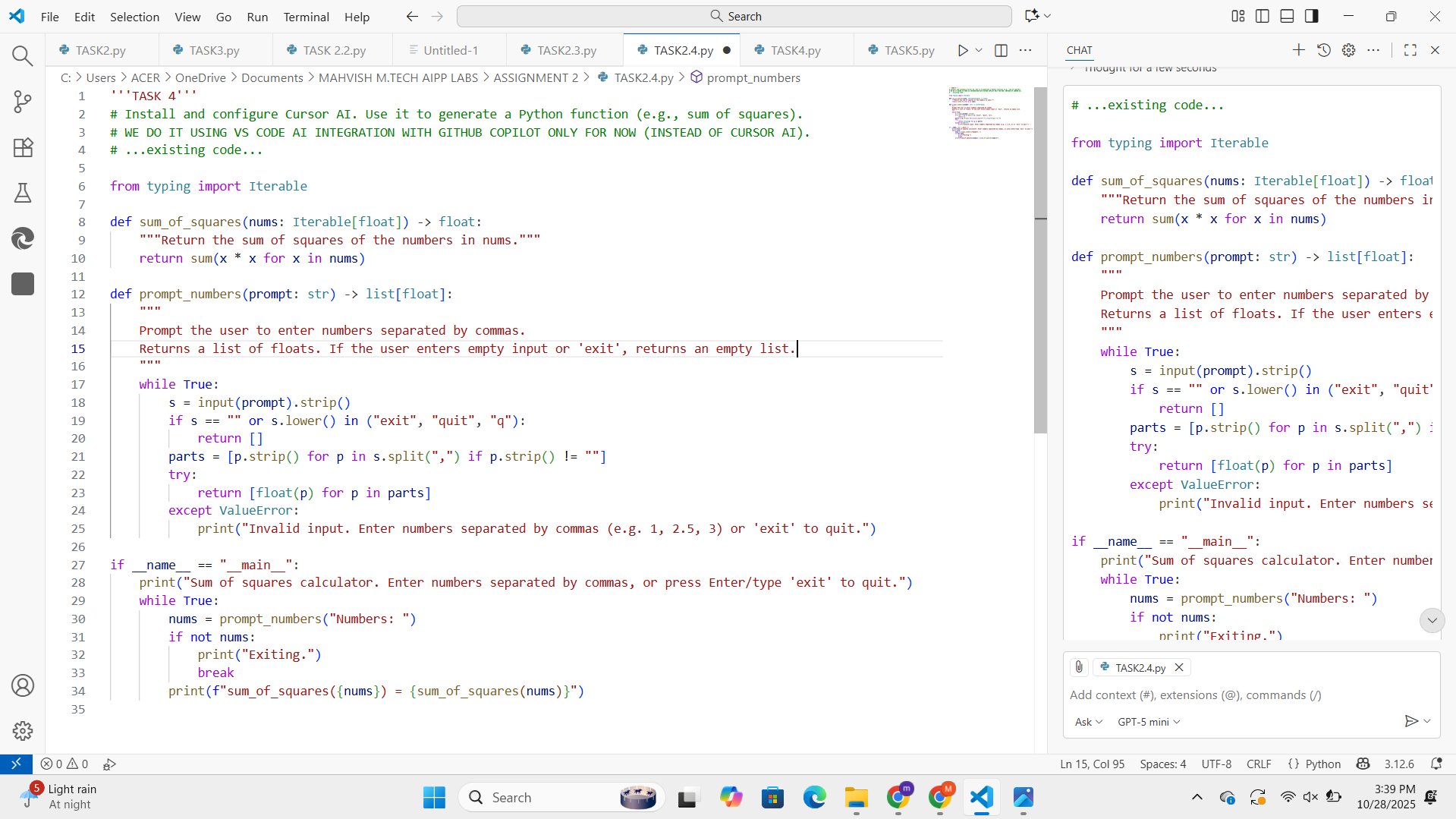
**Install and configure Cursor AI. Use it to generate a python function (e.g. Sum of Squares).**

***We will execute this normally using VS code and Copilot integration; Cursor AI can be installed in later assignments.***

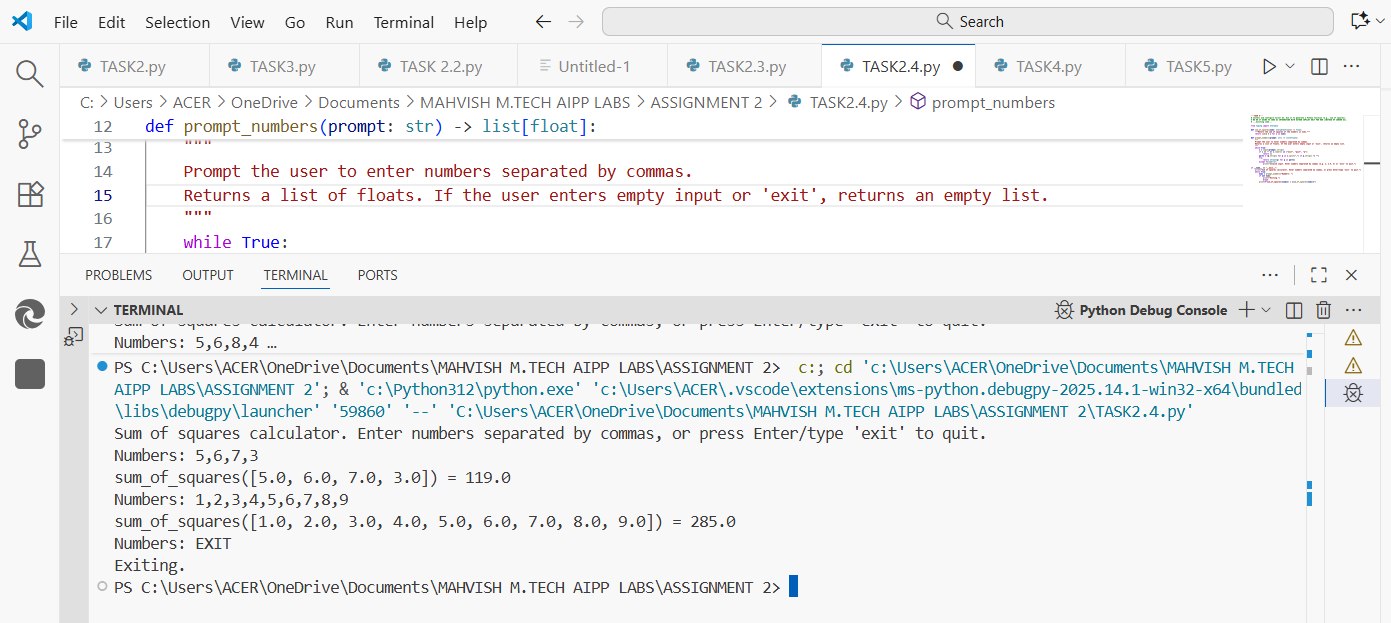
***PROMPTS:***

**1) Generate sum of squares and input should be given by user.**

***CODE:***



***OUTPUT:***



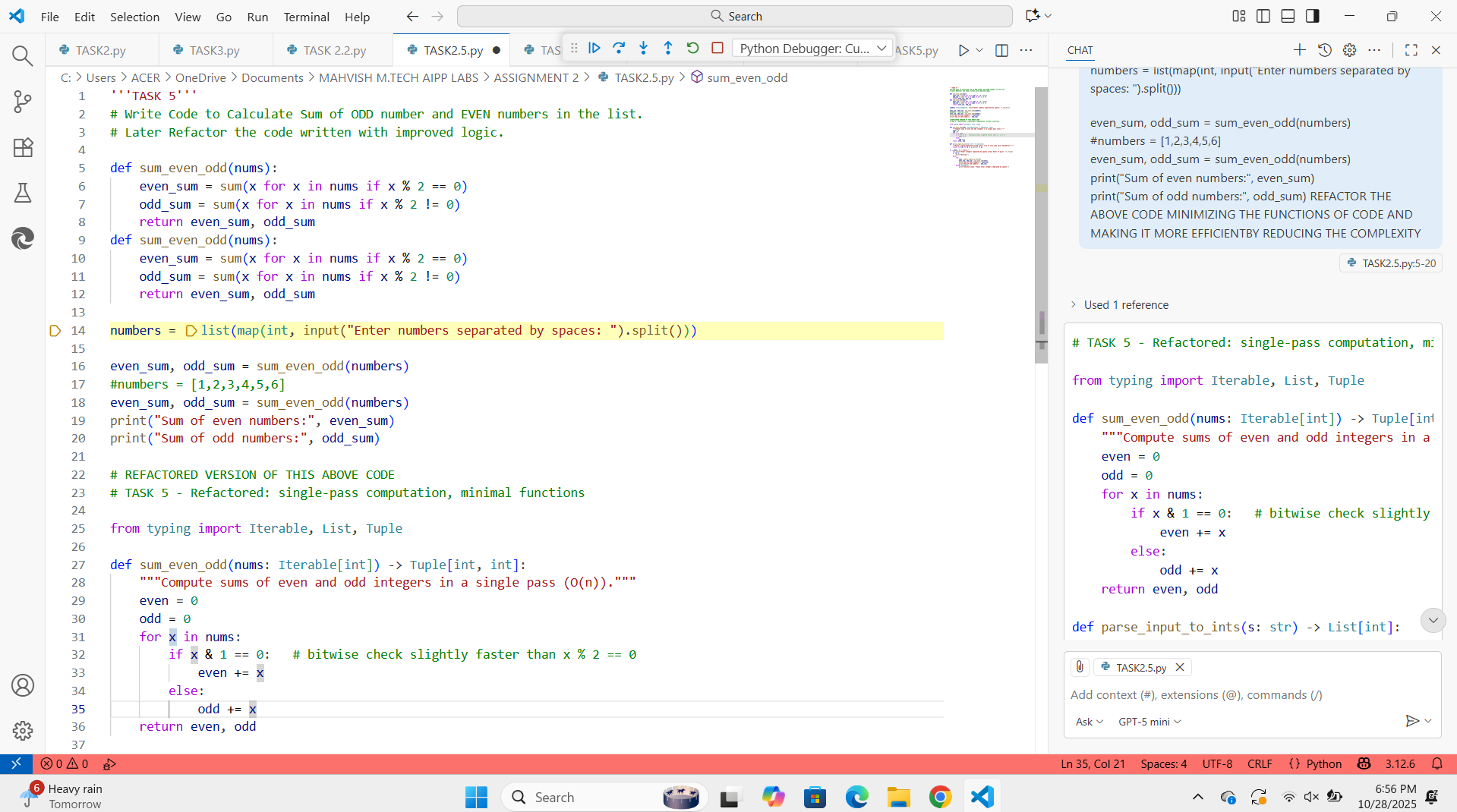
**Task 5:**

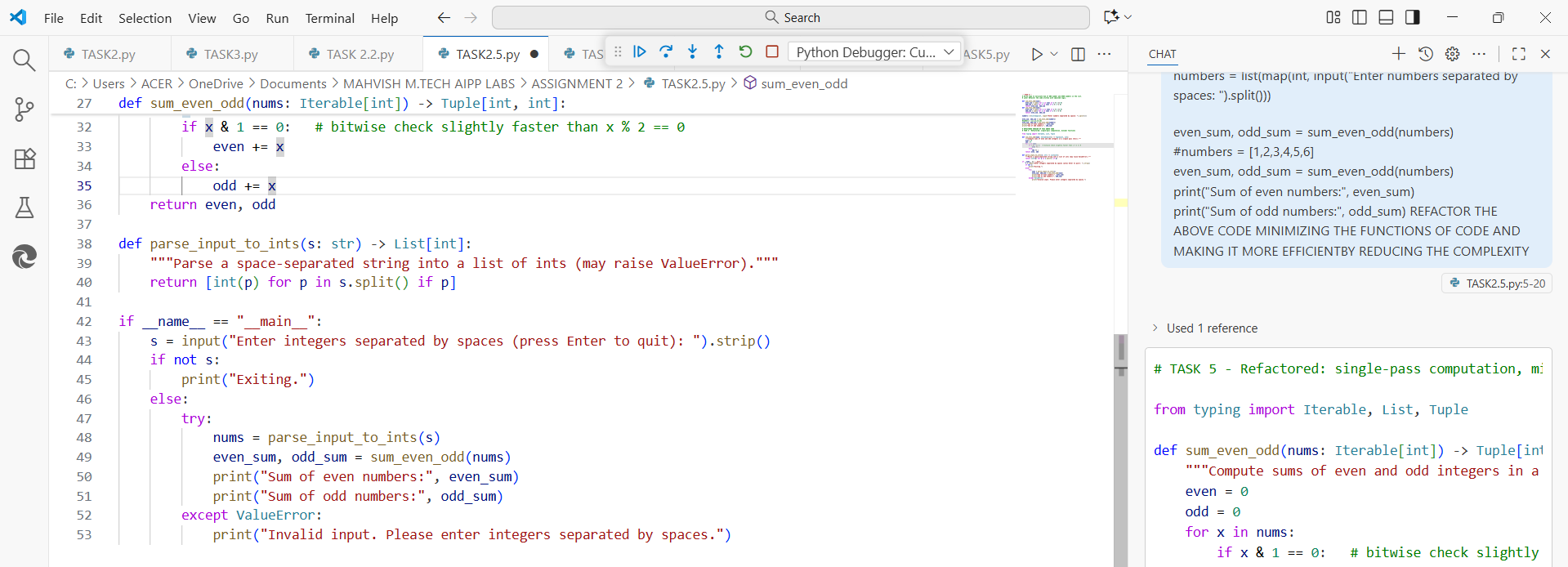
**Students need to write code to calculate the sum of odd numbers and even numbers in the list. Later, Refactor the code with improved logic.**

***PROMPTS:***

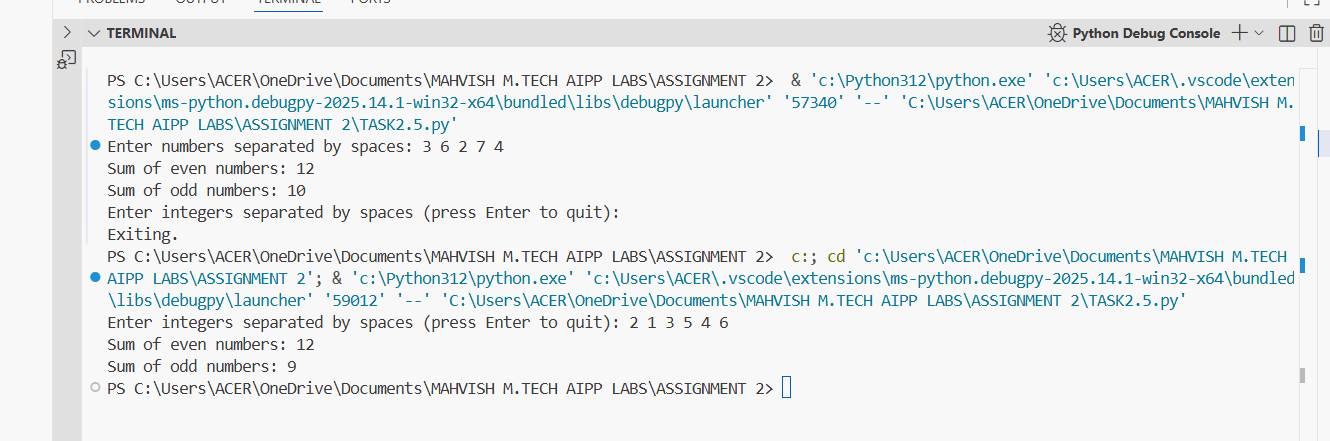
**1) Refactor the above written code of sum of odd and even numbers to make it more efficient and reducing complexity.**

***CODE:***





***OUTPUT:***



**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**