

ASSIGNMENT – 13

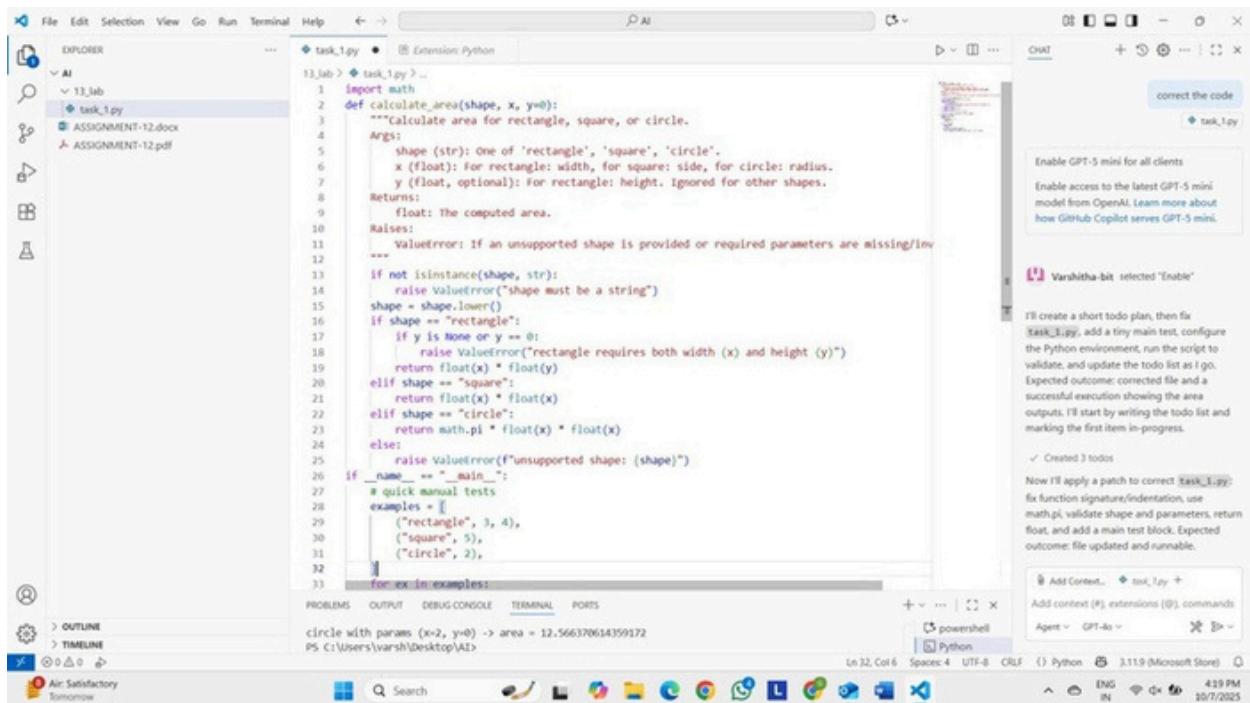
NAME: P. Shreyash

HT.NO: 2403A52048

Batch no:- 03

Task-1: Prompt: correct

the code. Code:



```
13_lab > task_1.py > ...
1 import math
2 def calculate_area(shape, x, y=0):
3     """Calculate area for rectangle, square, or circle.
4     Args:
5         shape (str): One of 'rectangle', 'square', 'circle'.
6         x (float): For rectangle: width, for square: side, for circle: radius.
7         y (float, optional): For rectangle: height. Ignored for other shapes.
8     Returns:
9         float: The computed area.
10    Raises:
11        ValueError: If an unsupported shape is provided or required parameters are missing/inv
12    """
13    if not isinstance(shape, str):
14        raise ValueError("shape must be a string")
15    shape = shape.lower()
16    if shape == "rectangle":
17        if y is None or y == 0:
18            raise ValueError("rectangle requires both width (x) and height (y)")
19        return float(x) * float(y)
20    elif shape == "square":
21        return float(x) * float(x)
22    elif shape == "circle":
23        return math.pi * float(x) * float(x)
24    else:
25        raise ValueError(f"unsupported shape: {shape}")
26 if __name__ == "__main__":
27     # quick manual tests
28     examples = [
29         ("rectangle", 3, 4),
30         ("square", 5),
31         ("circle", 2),
32     ]
33     for ex in examples:
```

circle with params (x=2, y=0) -> area = 12.566370614359172
PS C:\Users\varsh\Desktop\AI>

correct the code
task_1.py

Enable GPT-5 mini for all clients
Enable access to the latest GPT-5 mini model from OpenAI. Learn more about how GitHub Copilot serves GPT-5 mini.

Varshitha-bit selected "Enable"

I'll create a short todo plan, then fix task_1.py, add a tiny main test, configure the Python environment, run the script to validate, and update the todo list as I go. Expected outcome: corrected file and a successful execution showing the area outputs. I'll start by writing the todo list and marking the first item in-progress.

✓ Created 3 todos

Now I'll apply a patch to correct task_1.py: fix function signature/indentation, use math.pi, validate shape and parameters, return float, and add a main test block. Expected outcome: file updated and runnable.

Add Context... task_1.py +

Add context (F), extensions (E), commands

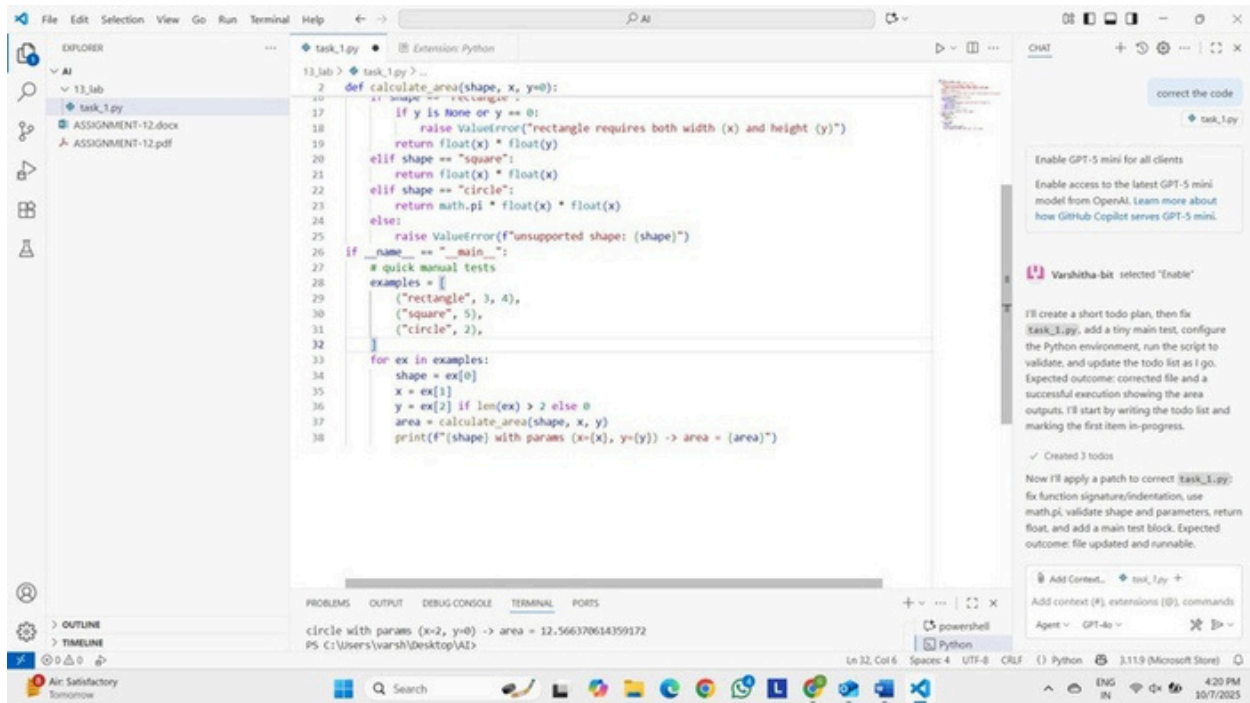
Agent - GPT-4o

Ln 32, Col 6 Spaces: 4 UTF-8 CRLF Python 3.11.9 (Microsoft Store)

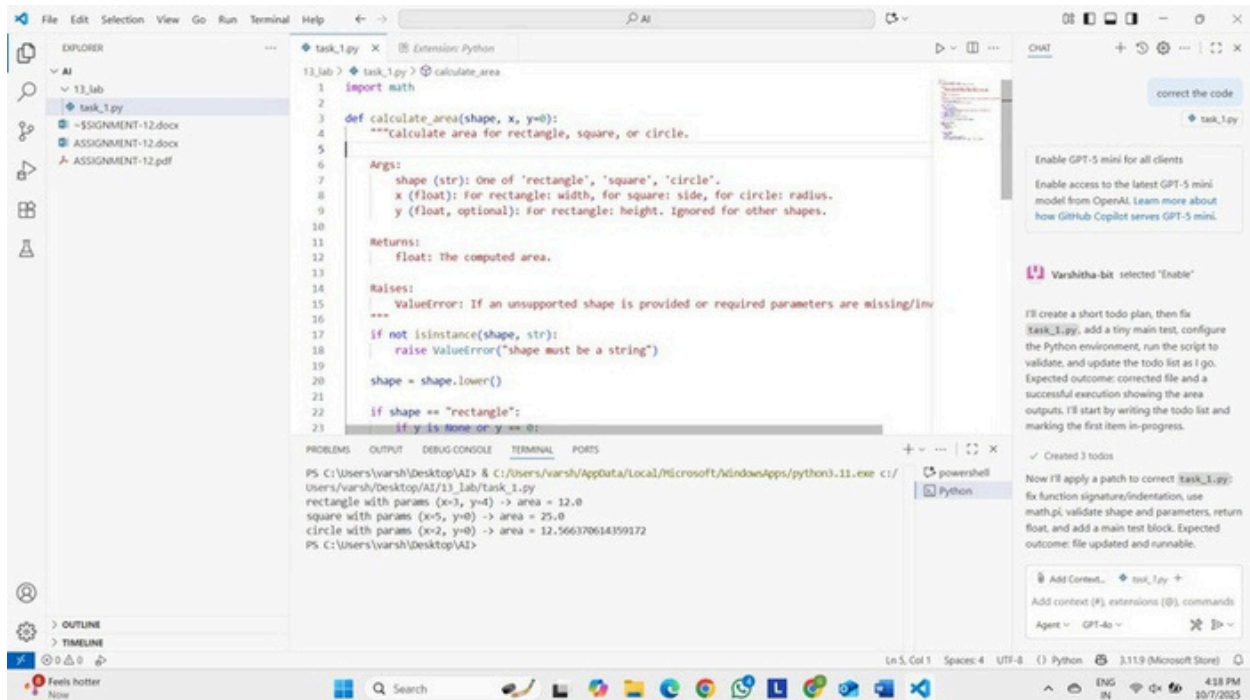
Air Satisfactory Tomorrow

Search

419 PM 10/7/2025



Output:



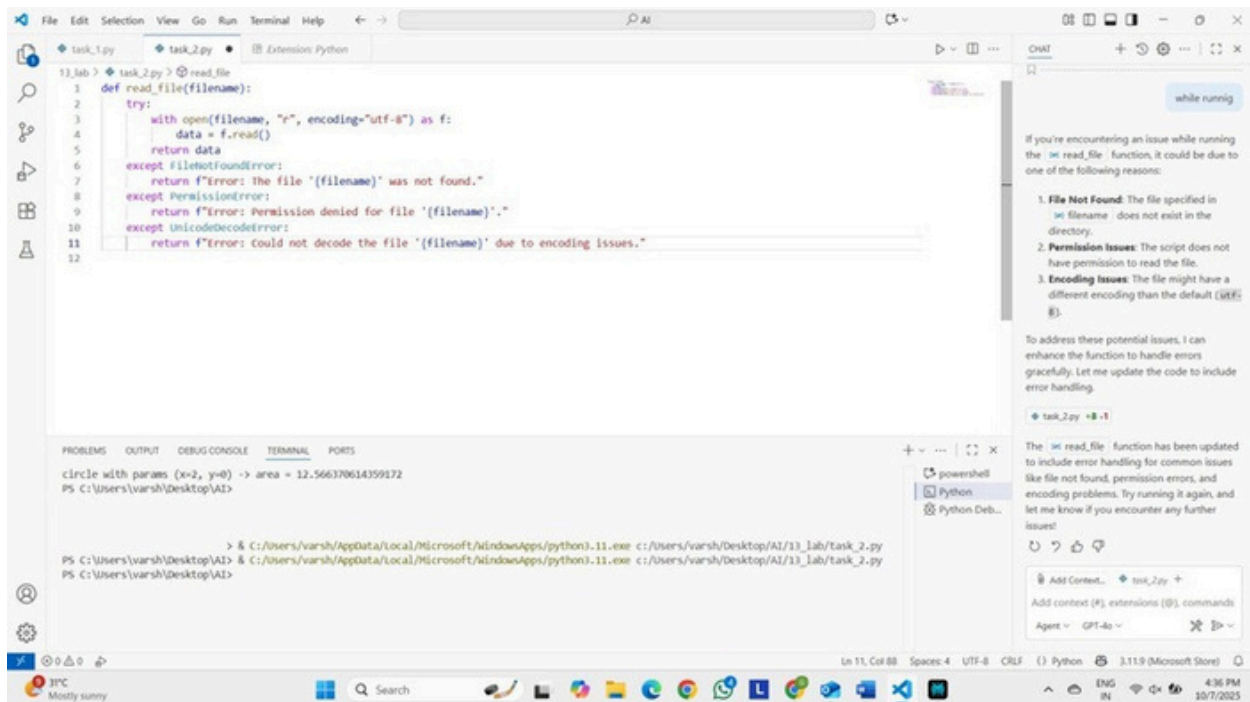
Observation:

First when we write the code in the work space, then when we give the prompt to correct the code it immediately changed the whole code. The original code was hardly 6 lines but the generated code is more lines. It even included comments so that we can understand the code easily. Now the code is in a correct structure.

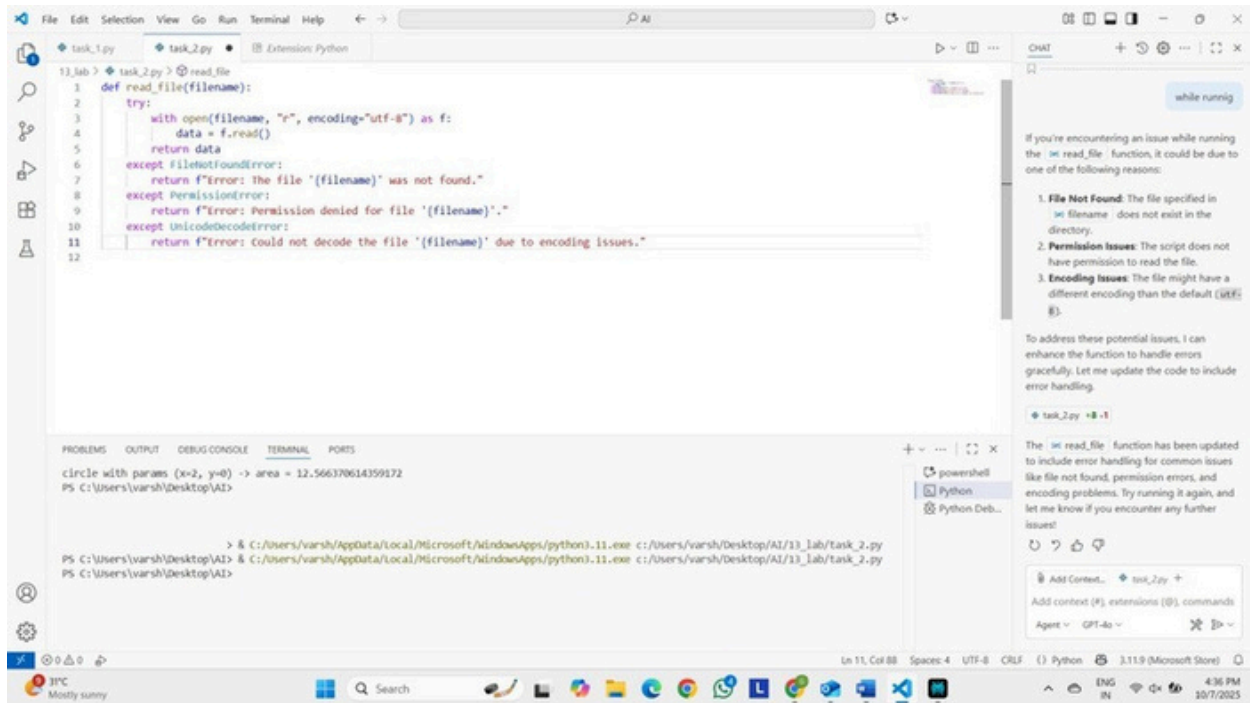
Task-2:

Prompt: Correct my mistake

Code:



Output:



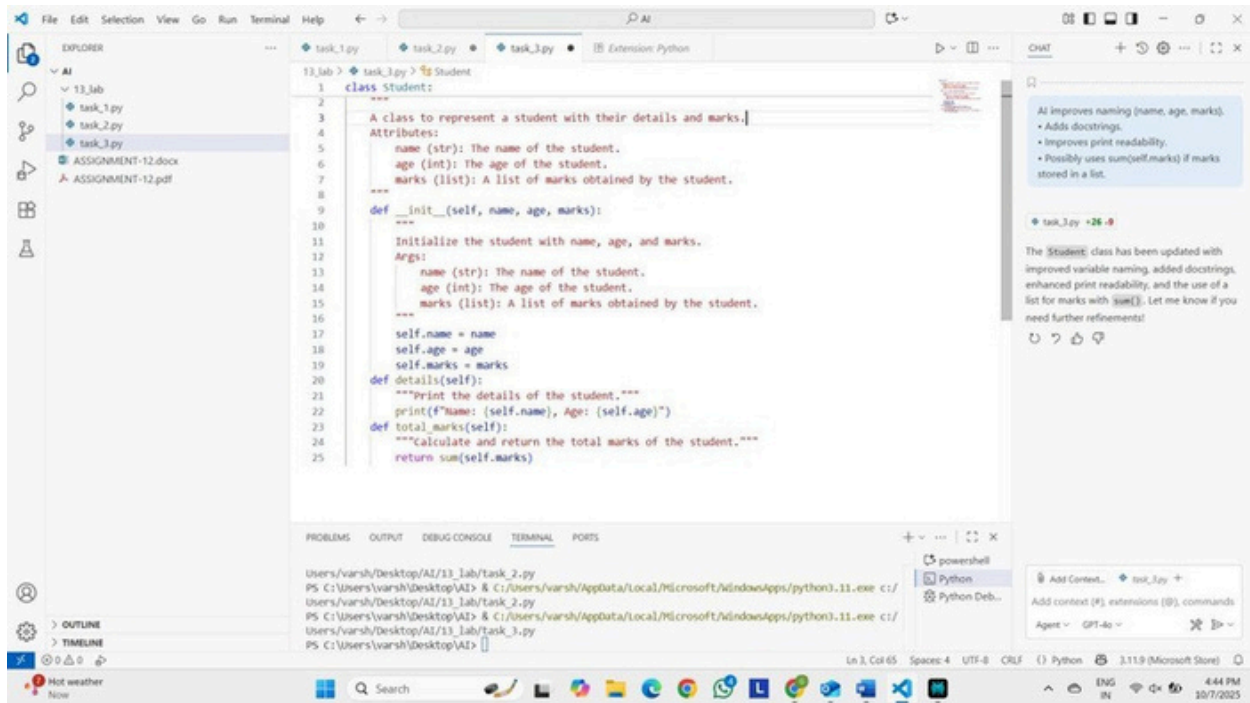
Observation:

In the give code we did not give a code with proper indentation. And file handling was also not perfect. But as we gave the prompt to correct it. It handled all that issues.

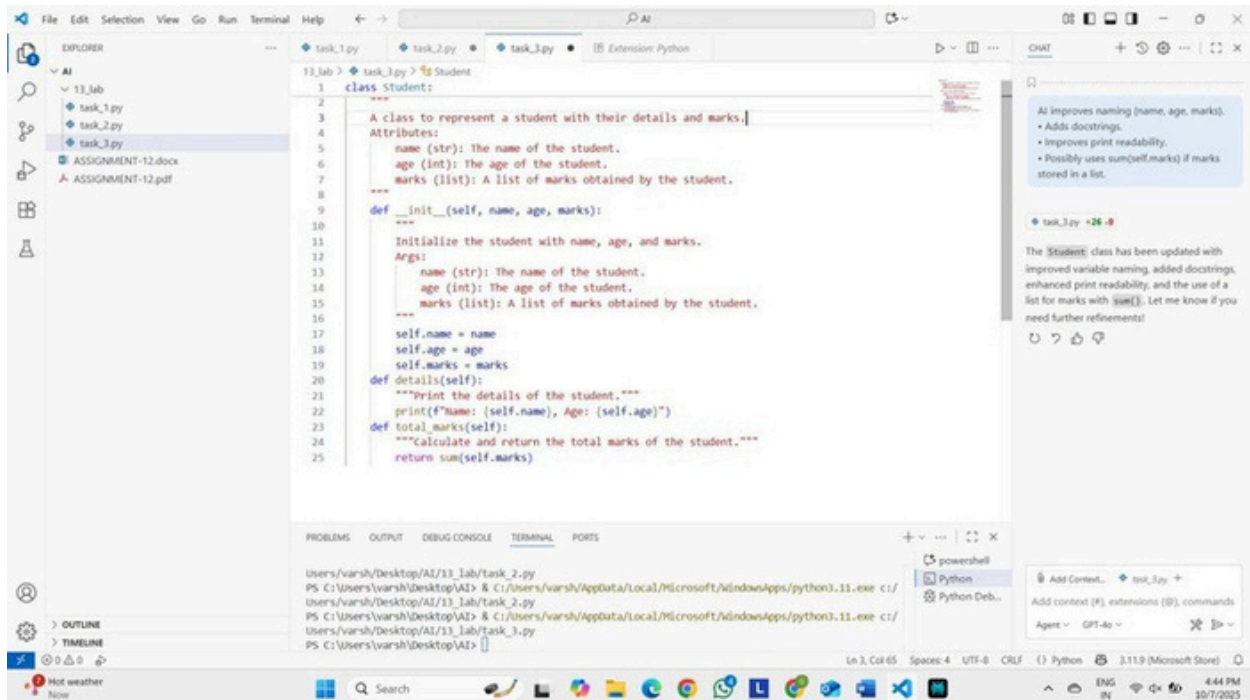
Task-3:

Prompt: correct my code. In this the attributes are name, age and marks.

Code:



Output:



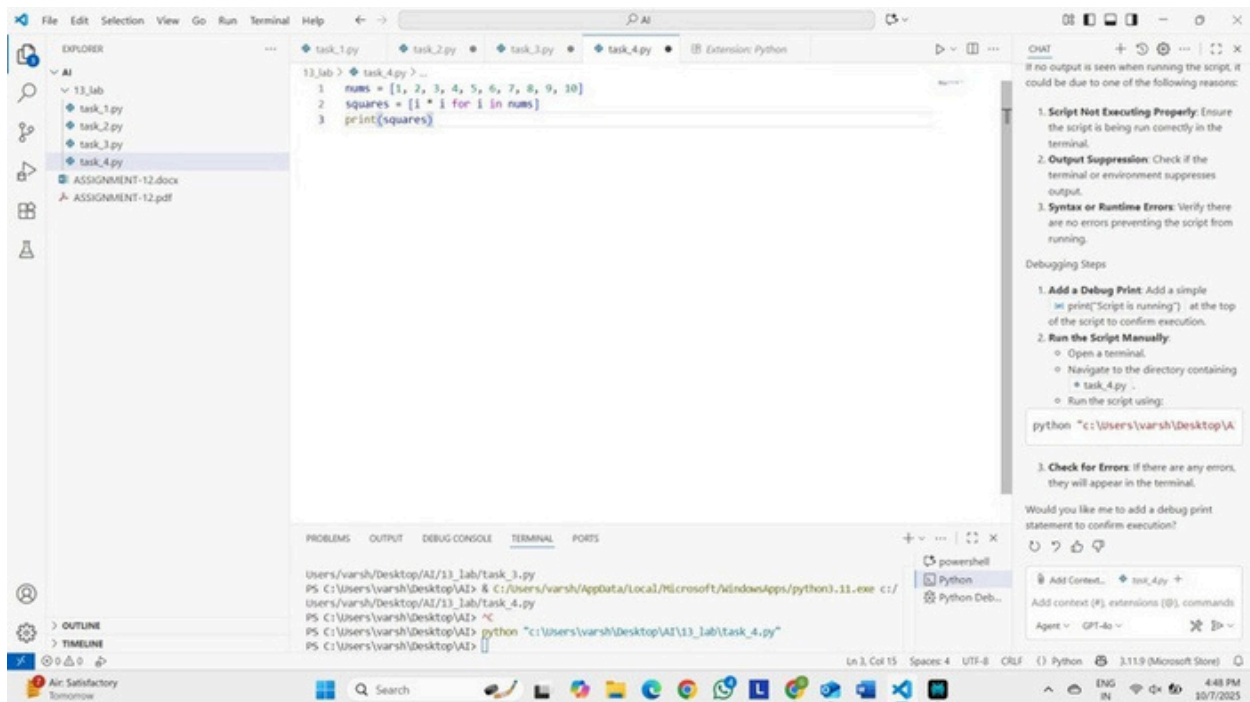
Observation:

First when we write the code in the work space, then when we give the prompt to correct the code it immediately changed the whole code. The original code was hardly 6 lines but the generated code is more lines. It even included comments so that we can understand the code easily. Now the code is in a correct structure

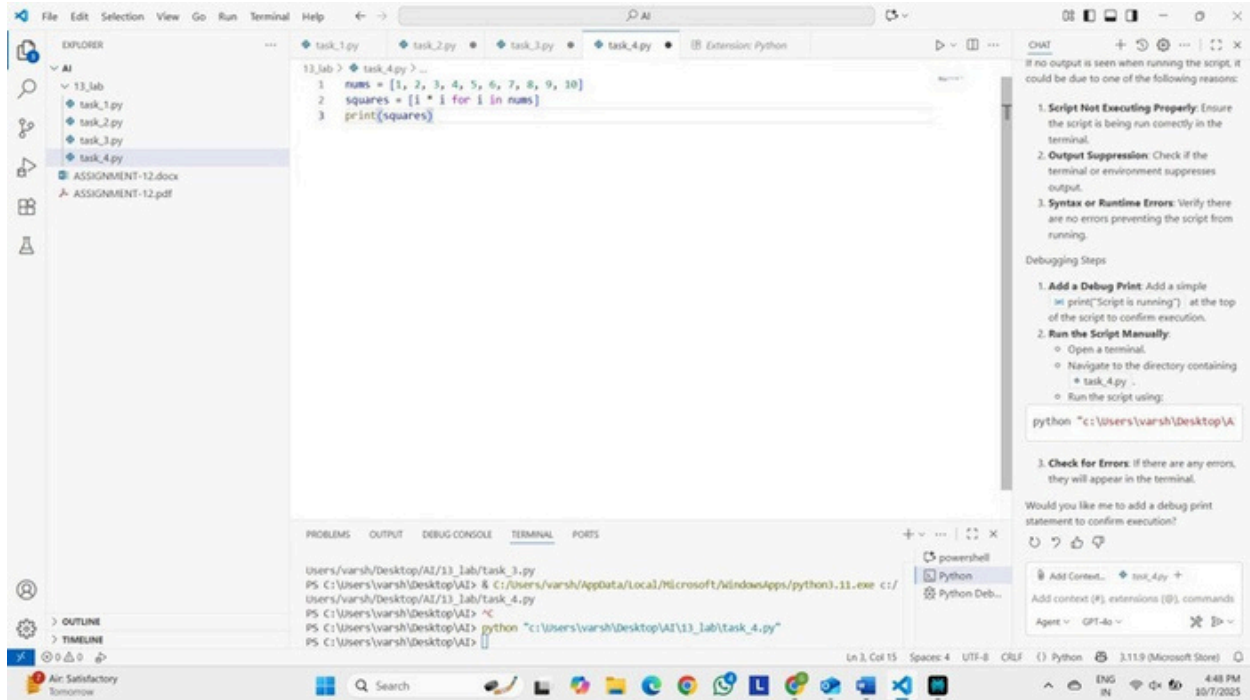
Task-4:

Prompt: Correct my mistake

Code:



Output:



Observation:

First when we write the code in the work space, then when we give the prompt to correct the code it immediately changed the whole code. The original code was hardly 6 lines but the generated code is more lines. It even included comments so that we can understand the code easily. Now the code is in a correct structure