

ASSIGNMENT – 13

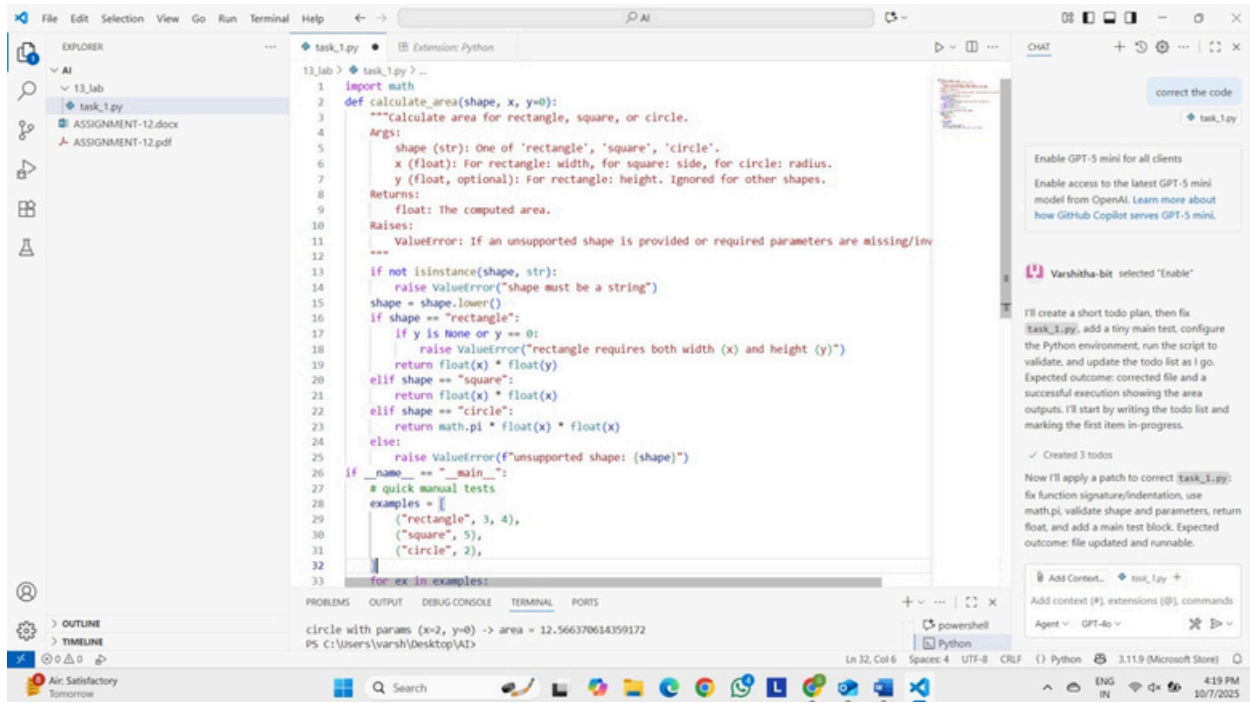
NAME: V.Himesh Ram

HT.NO: 2403A52045

Batch no:- 03

Task-1: Prompt: correct

the code. Code:

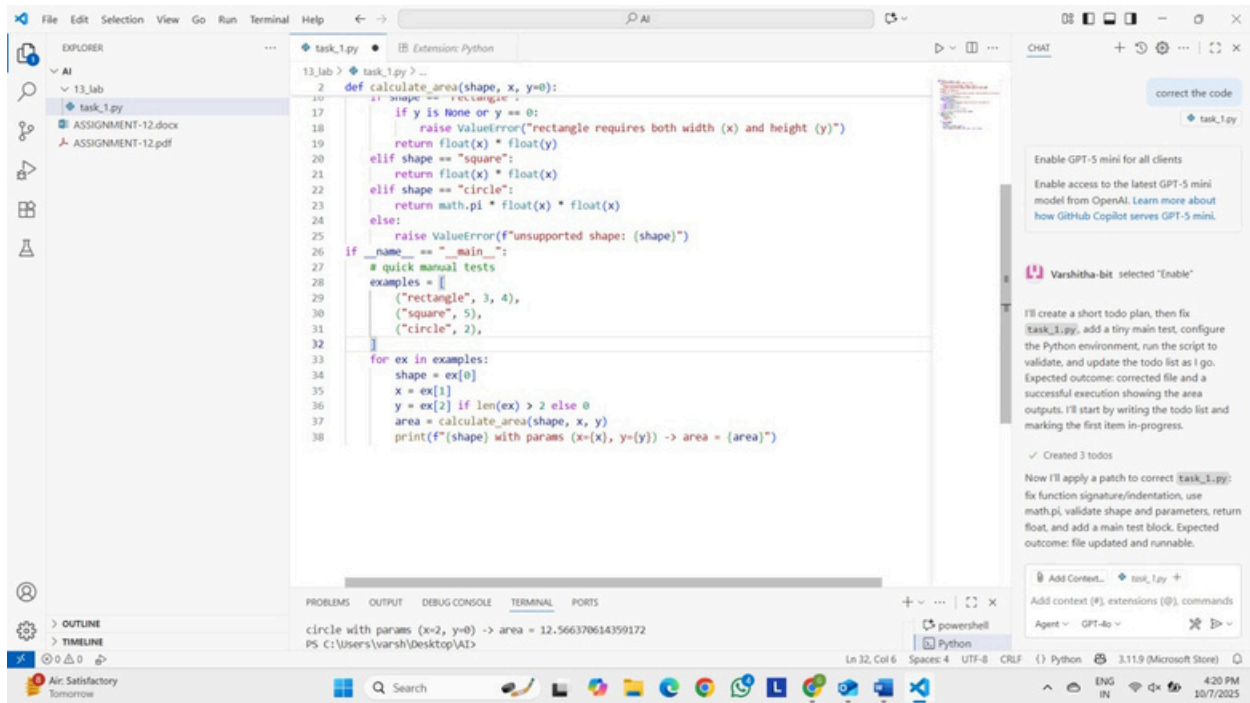


The screenshot shows a Visual Studio Code editor with a Python file named `task_1.py`. The code defines a `calculate_area` function that takes a shape and dimensions as input and returns the area. The function handles rectangles, squares, and circles. It includes docstrings for arguments and returns, and raises a `ValueError` for unsupported shapes or missing parameters. A `__main__` block contains a quick manual test with examples for rectangle, square, and circle.

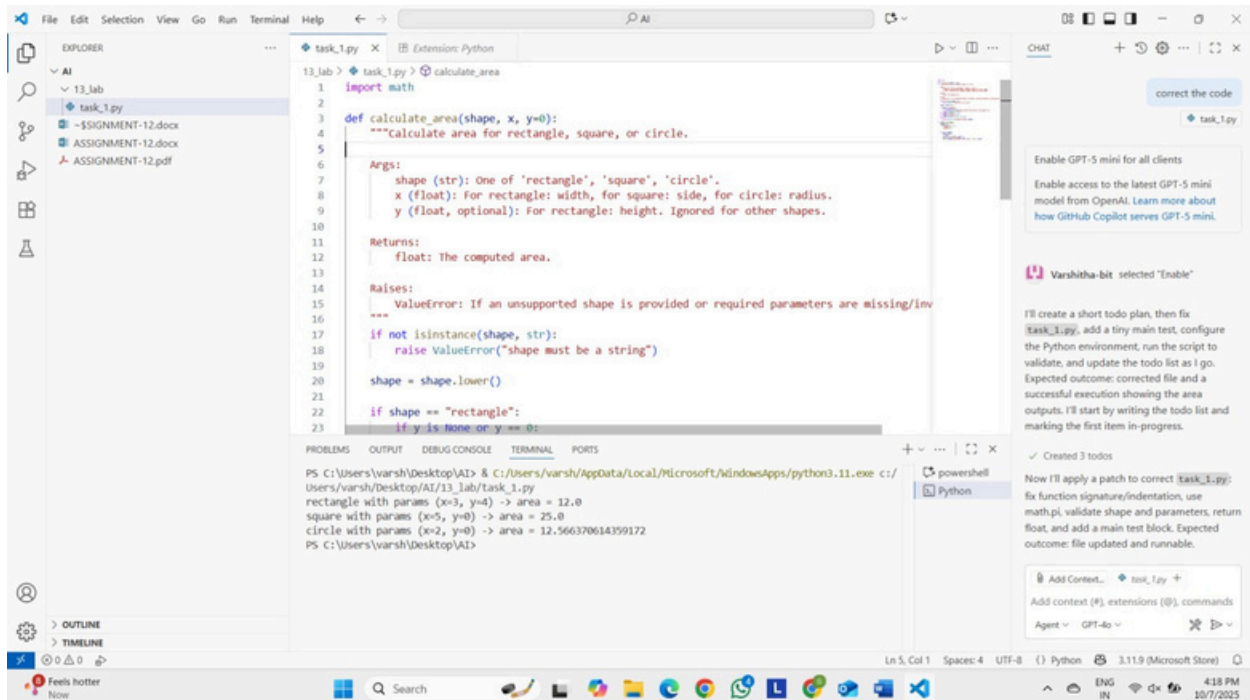
On the right side, there is a chat window with the following content:

- A button labeled "correct the code" with a link to `task_1.py`.
- A message: "Enable GPT-5 mini for all clients".
- A message: "Enable access to the latest GPT-5 mini model from OpenAI. Learn more about how GitHub Copilot serves GPT-5 mini."
- A message: "Varshitha-bit selected 'Enable'".
- A message: "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."
- A message: "Created 3 todos".
- A message: "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."
- A button labeled "Add Context..." with a link to `task_1.py`.
- A message: "Add context (#), extensions (@), commands".
- A dropdown menu showing "Agent" with "GPT-4o" selected.

The terminal at the bottom shows the command `python task_1.py` and the output: `circle with params (x=2, y=0) -> area = 12.566370614359172`.



Output:



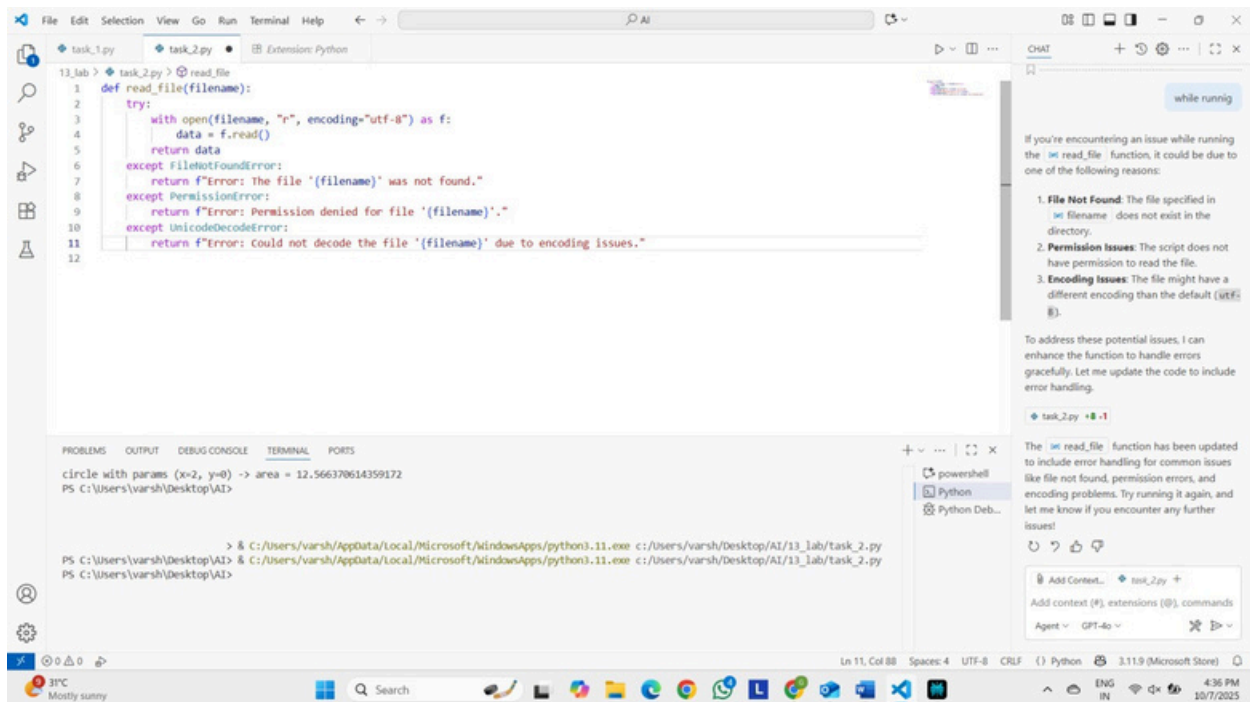
Observation:

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.

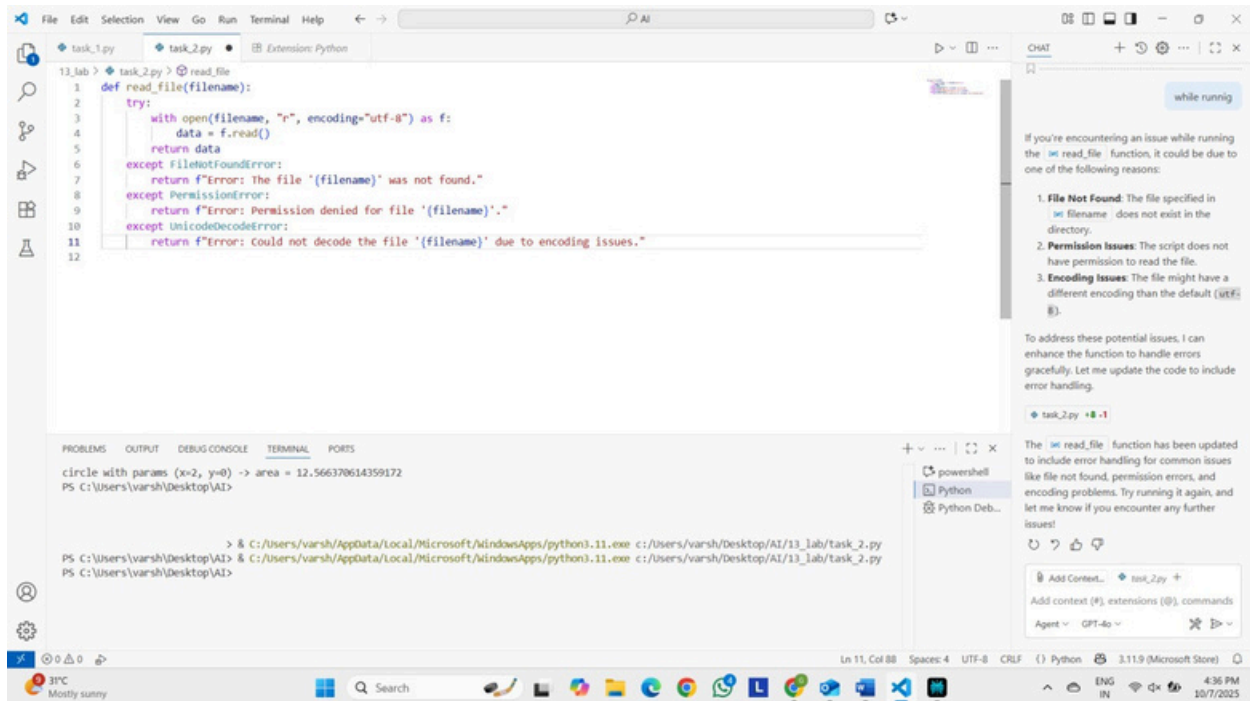
Task-2:

Prompt: Correct my mistake

Code:



Output:



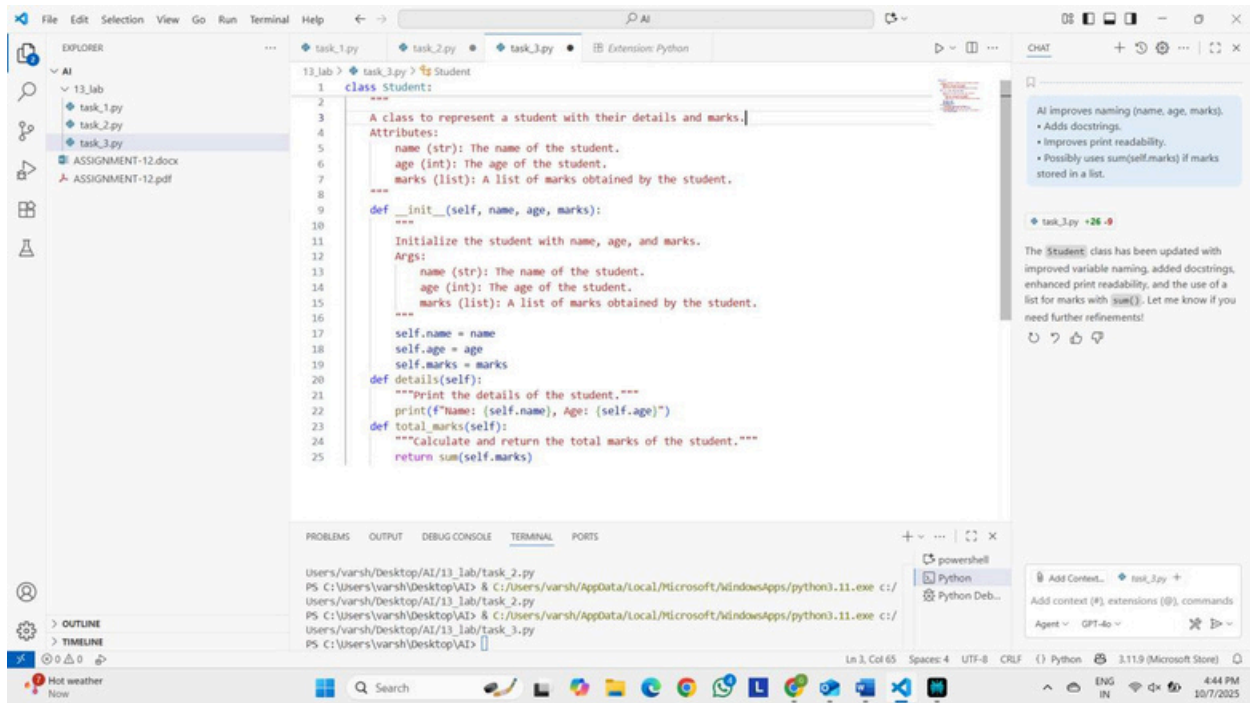
Observation:

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.

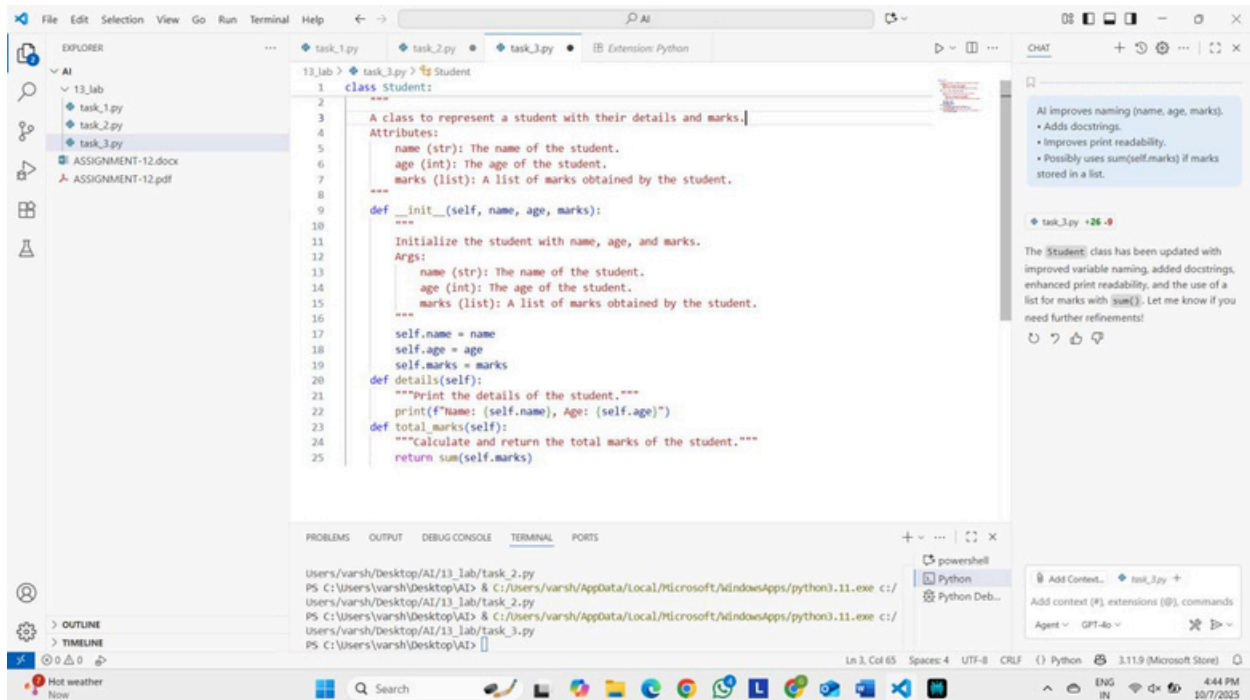
Task-3:

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

Code:



Output:



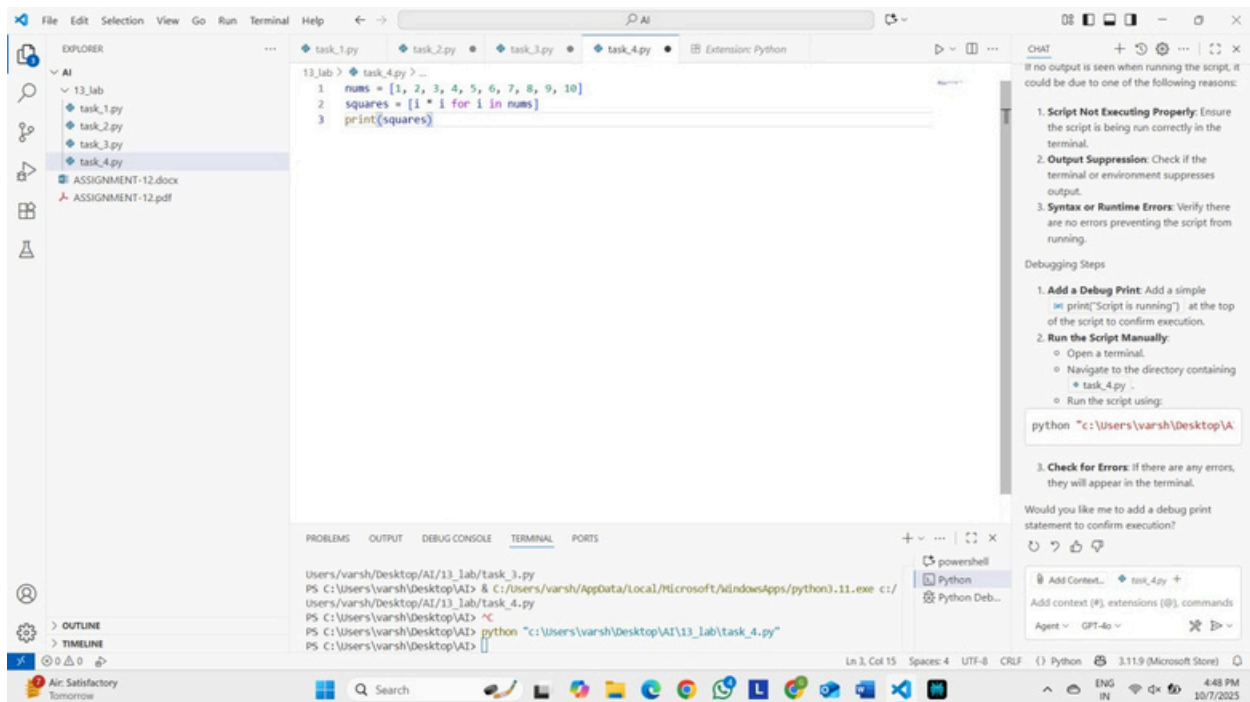
Observation:

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

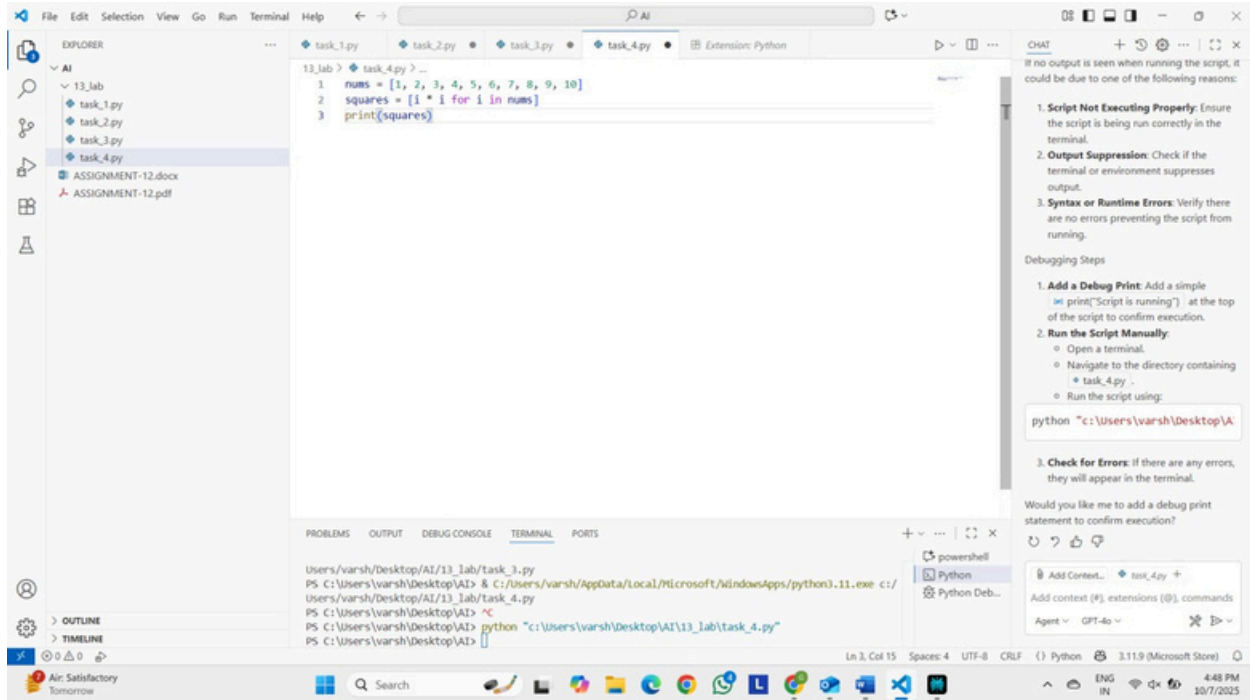
Task-4:

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