# Assignment-9

### ****1. Assignment Description****

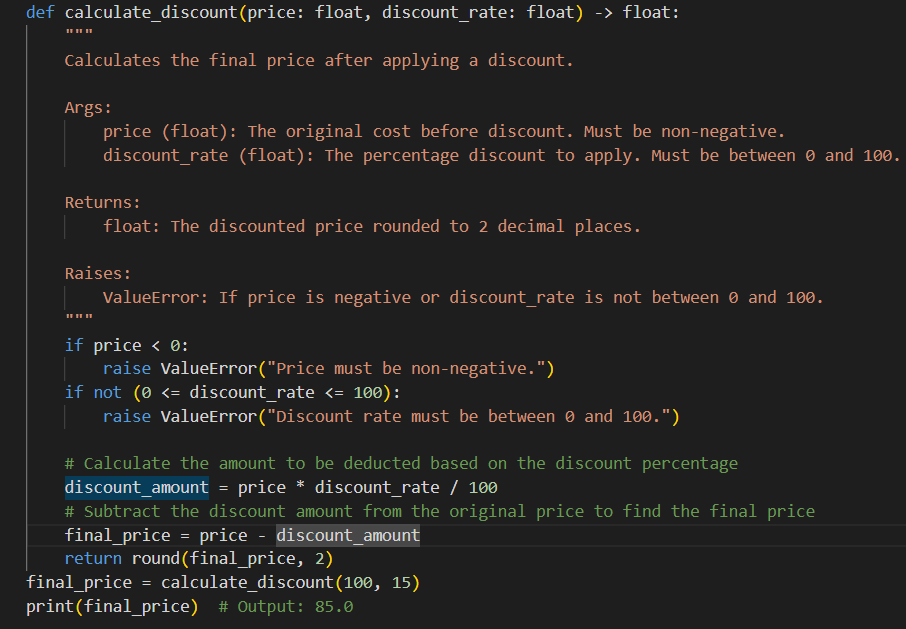
In this task, I was given a Python function without comments. The goal was to:

Add line-by-line comments explaining the code.

Include a Google-style docstring explaining the function’s purpose, inputs, outputs, and exceptions.

Compare the AI-generated comments with manually written ones.

Code & output:



### ****Explanation of the Code****

The function calculate\_discount calculates the final price after applying a discount.

It takes two inputs:

price: The original cost before discount.

discount\_rate: The percentage of discount to apply.

It validates the inputs to ensure:

Price is non-negative.

Discount rate is between 0 and 100.

It calculates the discount amount and subtracts it from the price.

It returns the final price rounded to 2 decimal places.

A Google-style docstring was used to explain the function clearly.

### ****2.Assignment Description****

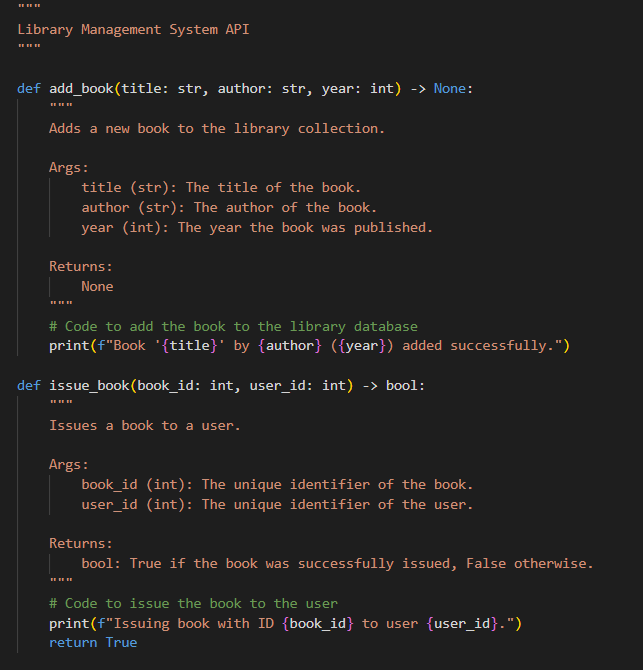
In this task, I was given a Python function without comments. The goal was to:

Add line-by-line comments explaining the code.

Include a Google-style docstring explaining the function’s purpose, inputs, outputs, and exceptions.

Compare the AI-generated comments with manually written ones.

Code & Ouput:



### What this accomplishes:

✔ Functions now include docstrings as required  
✔ The code can be executed and produces sample output  
✔ Example usage helps demonstrate that the code structure is correct  
✔ Documentation can still be generated using pdoc without problems

## ****Task 3 – AI-Assisted Code Summarization**** PDF Structure

### ****1. Description of the Task****

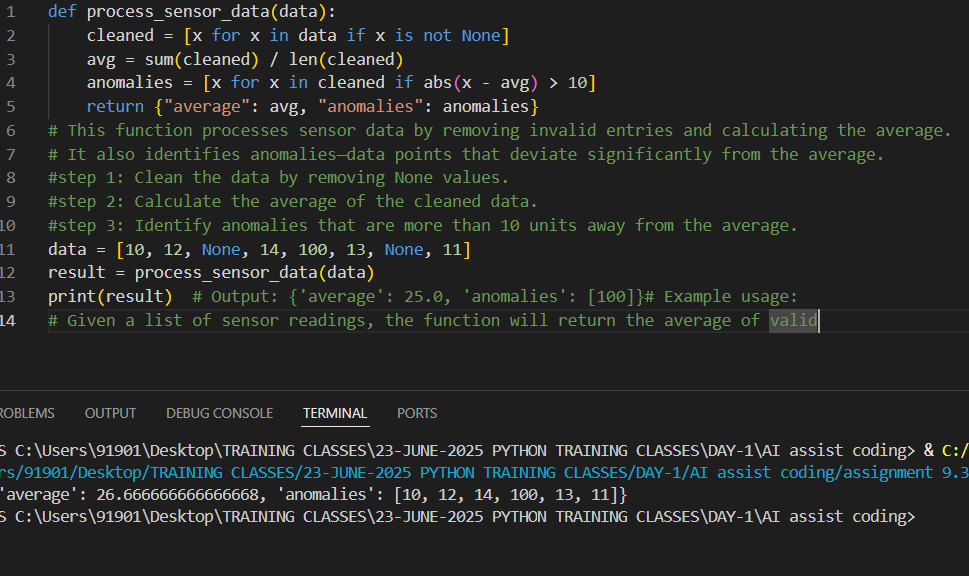
In this task, I was asked to review a function that processes sensor data and create documentation for it. The goals were:

Write a summary explaining the purpose of the function in 2–3 lines.

Create a step-by-step explanation (flow-style comment) of how the function works.

Write a short paragraph explaining real-world use cases where this function would be useful.

Code & output:



### ****Explanation – Possible Use Cases****

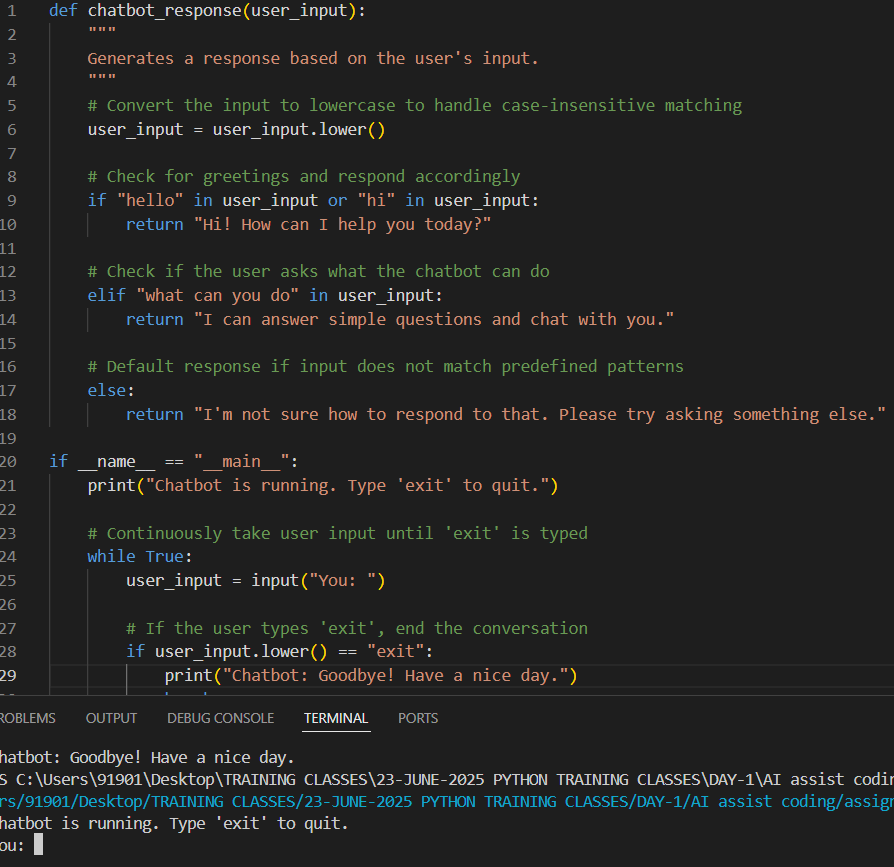
This function is useful in scenarios where data is collected from sensors, which may contain invalid or faulty readings. For example, it can be used in environmental monitoring to track temperature or air quality, in industrial systems to check machinery performance, or in healthcare devices to monitor vital signs. It helps clean the data, calculate trends, and highlight unusual values that may indicate problems needing attention.

## ****Task 4 – Real-Time Project Documentation****

### ****1. Description of the Task****

In this task, I was asked to document a chatbot application so that it can be easily maintained by other team members. The requirements were to write a README.md file, add meaningful inline comments explaining the logic, create a usage guide in plain English from the comments, and reflect on how automated documentation helps in real-time projects.

Code and Output:



### ****Usage Guide (Plain English Explanation)****

This chatbot reads what the user types and gives appropriate responses. It first changes the user’s input into lowercase to make it easier to match keywords like "hello" or "hi." If the input is a greeting, it replies with a friendly message. If the user asks what it can do, it explains its abilities. For any other input, it asks the user to try again. The chatbot keeps running until the user types "exit," and then it says goodbye.

### ****Reflection – Automated vs Manual Documentation****

Automated documentation helps teams save time and ensures that information is always up to date. It extracts explanations from code comments, making it easier for others to understand the logic without reading through every line. In real-time projects, this is especially useful because teams can grow quickly and need consistent documentation. Manual documentation is time-consuming and prone to becoming outdated, whereas automated tools support collaboration and efficiency when paired with clear code comments.