SR UNIVERSITY

AI ASSISTED CODING

BEGALA HASINI

2503A51L13

Lab 9.5 – Documentation Generation: Automatic Documentation and Code Comments

Lab Objectives:

- Inline Comments.
- Docstrings.
- Auto-Documentation Tools.
- AI assisted Summarization.

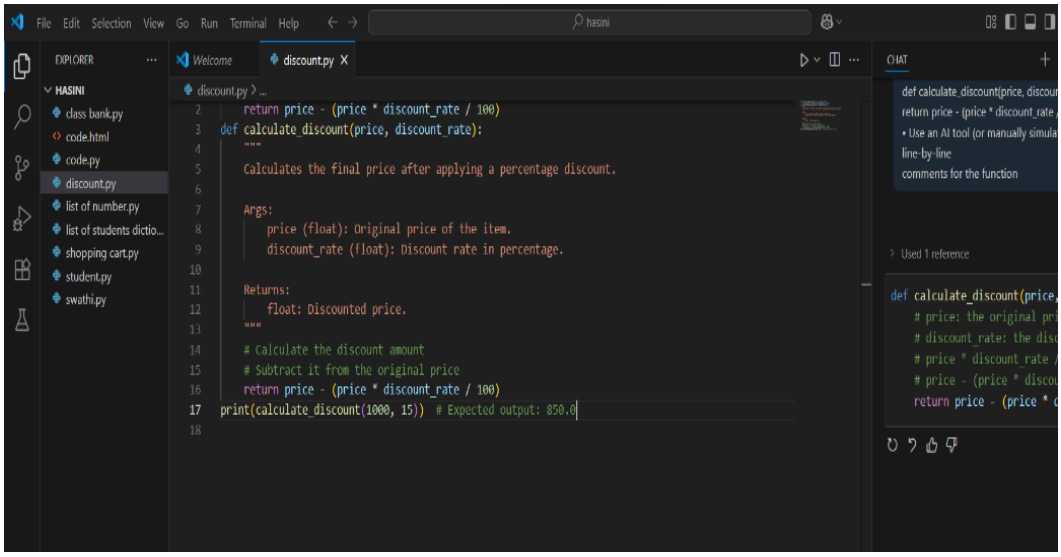
TASK DESCRIPTION #1 (Automatic Code Commenting)

Scenario: You have been given a Python Function without comments .

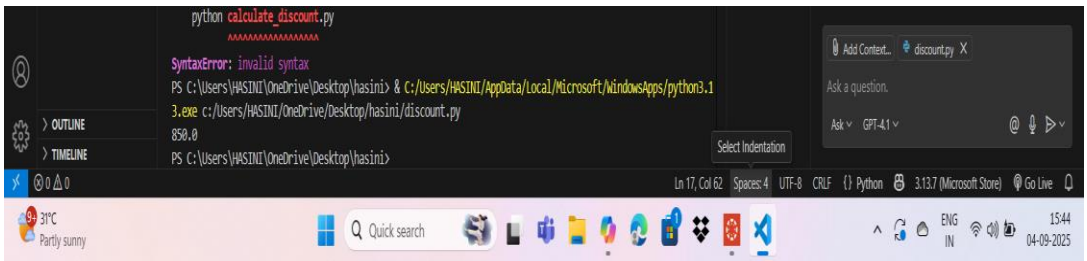
```
def calculate_discount(price ,discount_rate):
    Return price –(price*discount_rate/100)
```

- Use an AI tool (or manually simulate it) to generate line-by-line comments for the function.
- Modify the function so that it includes a docstring in Google-style or NumPy-style format.
- Compare the auto-generated comments with your manually written version.

CODE:



OUTPUT:



TASK DESCRIPTION #2(API Documentation Generator)

Scenario: A Team is building a **Libaray Management System** with multiple functions.

```
def add_book(title, author, year):

  # code to add book

   pass
```

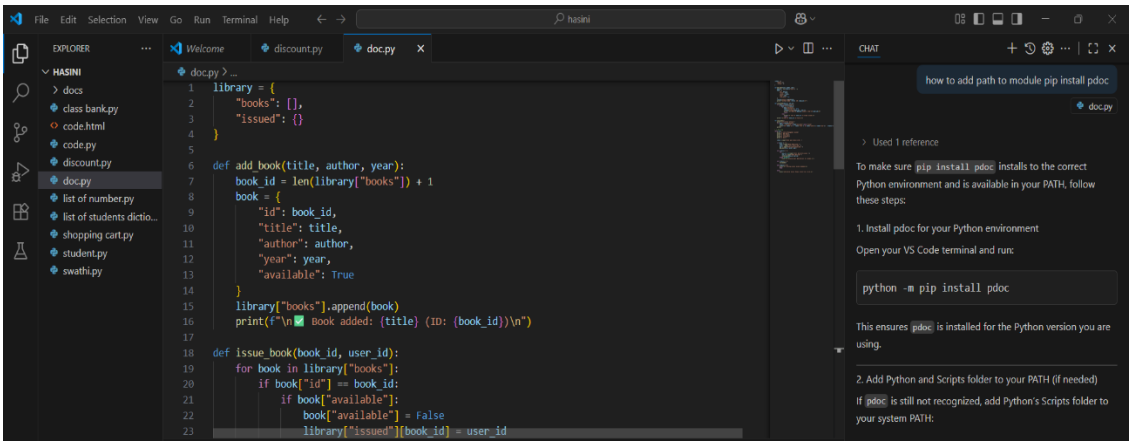
def issue_book(book_id, user_id)

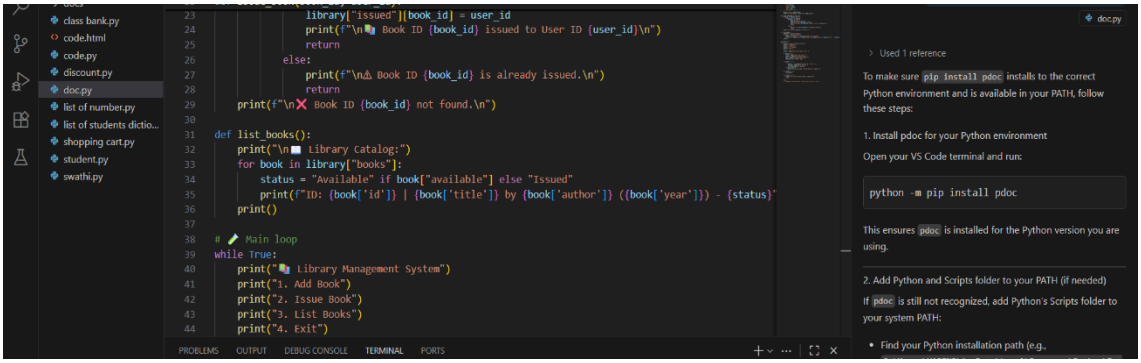
code to issue book

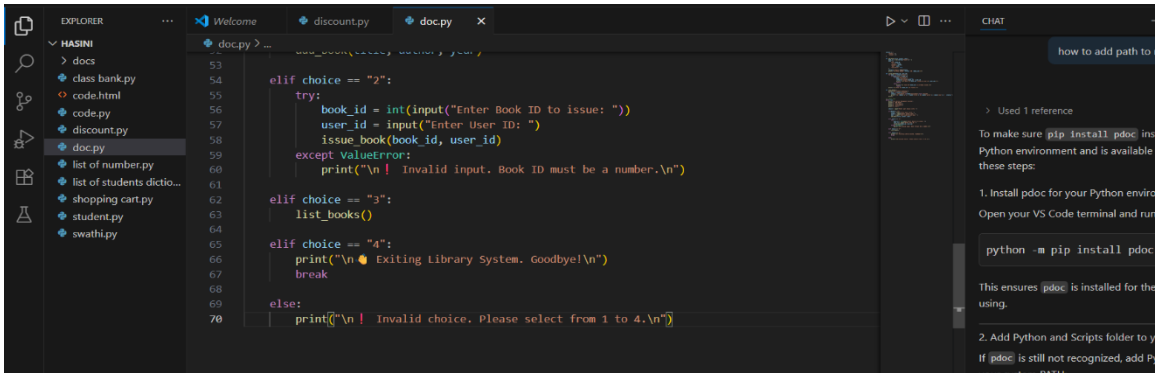
Pass

- Write a Python script that uses docstrings for each function (with input, output, and description).
- Use a documentation generator tool (like pdoc, sphinx, or MkDocs) to automatically create HTML documentation.
- Submit both the code and the generated documentation as output.

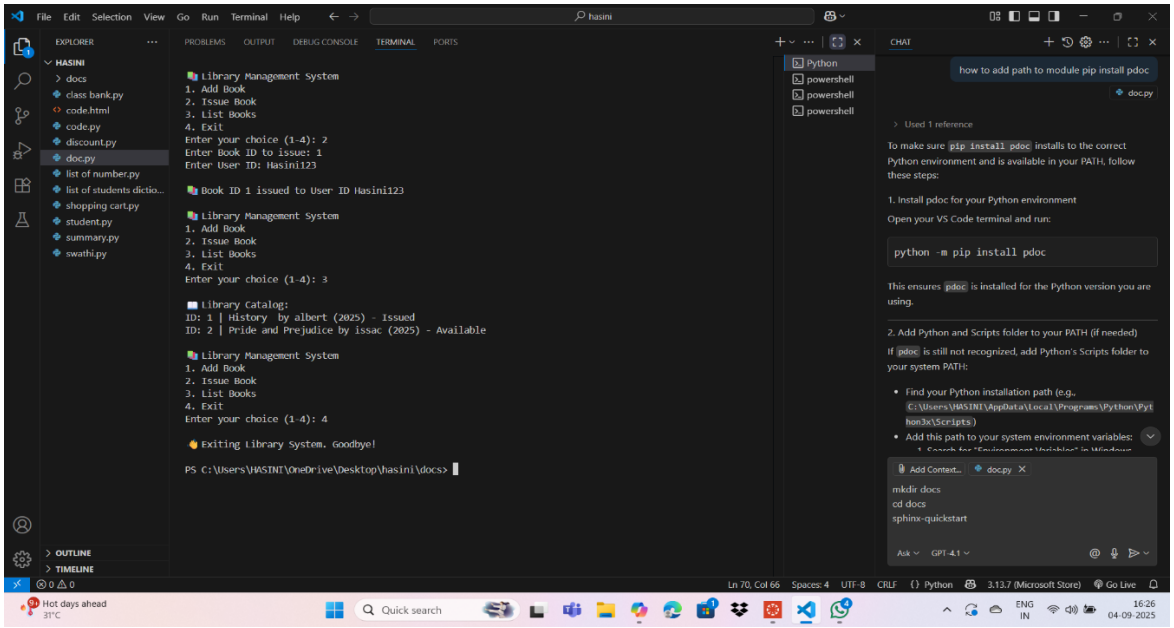
CODE :







OUTPUT :



TASK DESCRIPTION #3 (AI_Assisted Code Summarization)

SCENARIO: You are reviewing a colleagues codebase containing long functions.

def process_sensor_data(data):

Cleansed = [x for x in data if x is not None]

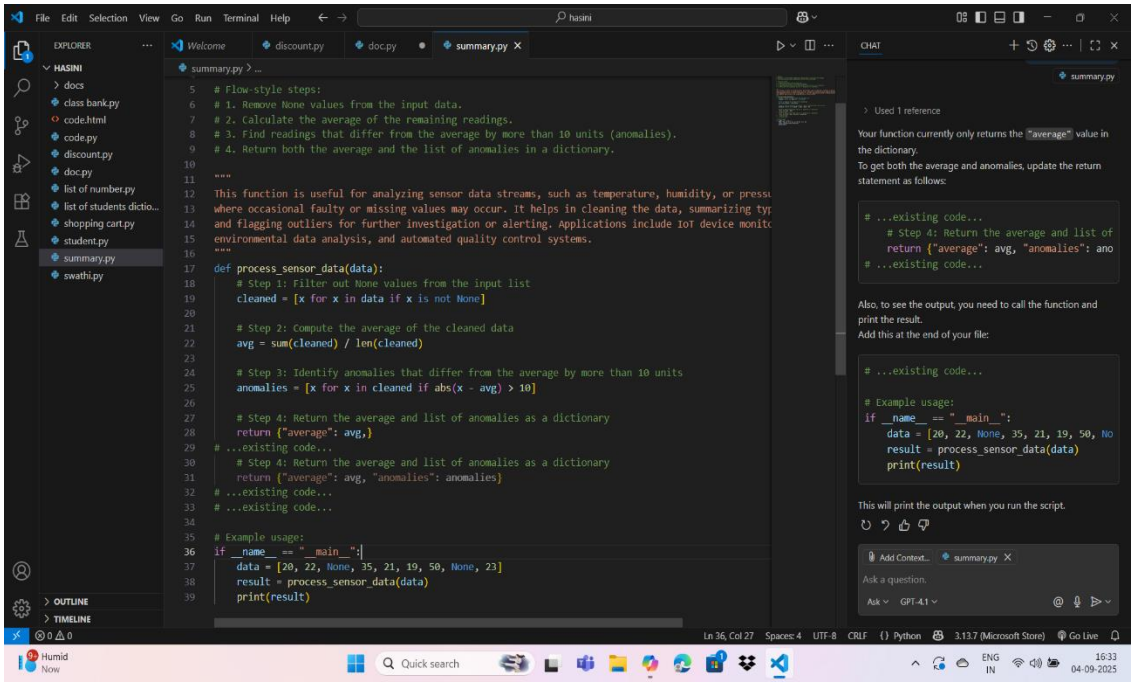
avg = sum (cleaned)/len(cleaned)

anomalies = [x for x in cleaned if abs(x- avg) > 10]

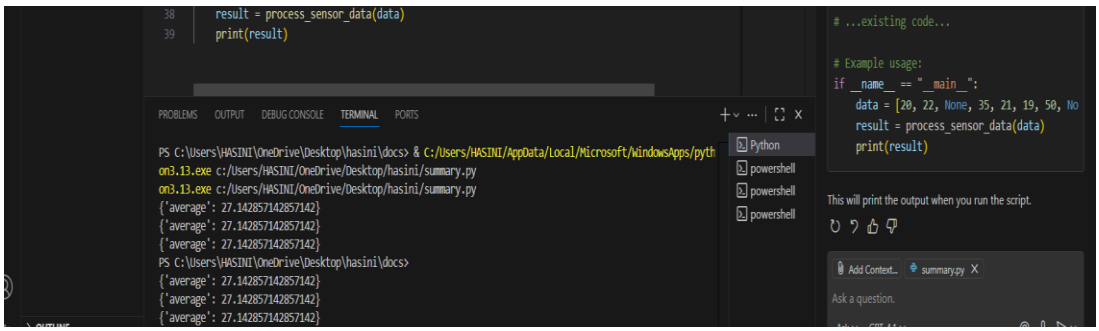
return {“average”: avg, “anomalies”:anomalies}

- Generate a summary comment explaining the purpose of the function in 2-3 lines.
- Create a flow-style comment (step-by-step explanation).
- Write a short paragraph of documentation describing possible use cases of this function in real-world scenarios.

CODE :



OUTPUT:

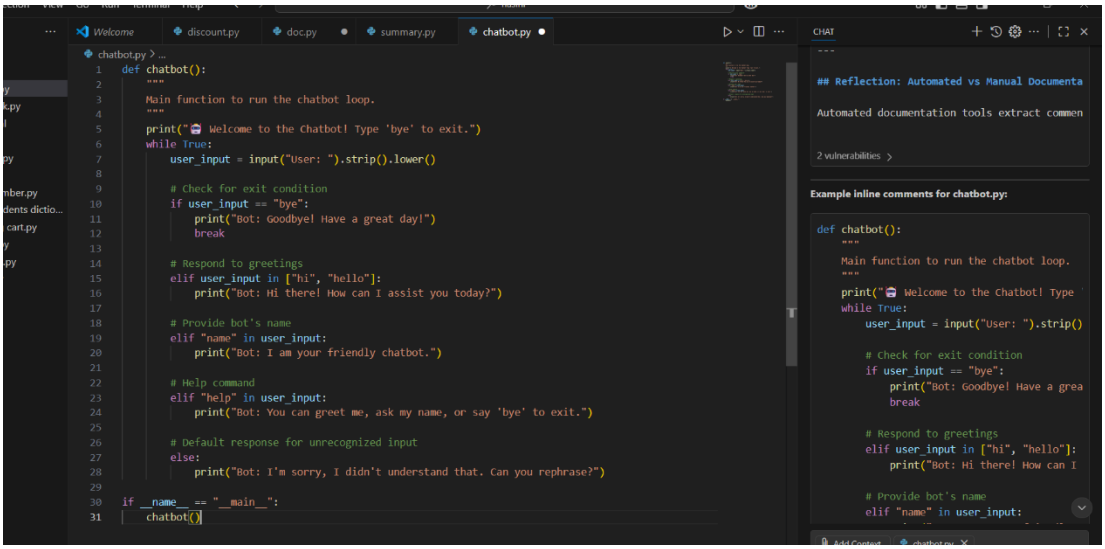


TASK DESCRIPTION #4 (Real-Time Project Documentation)

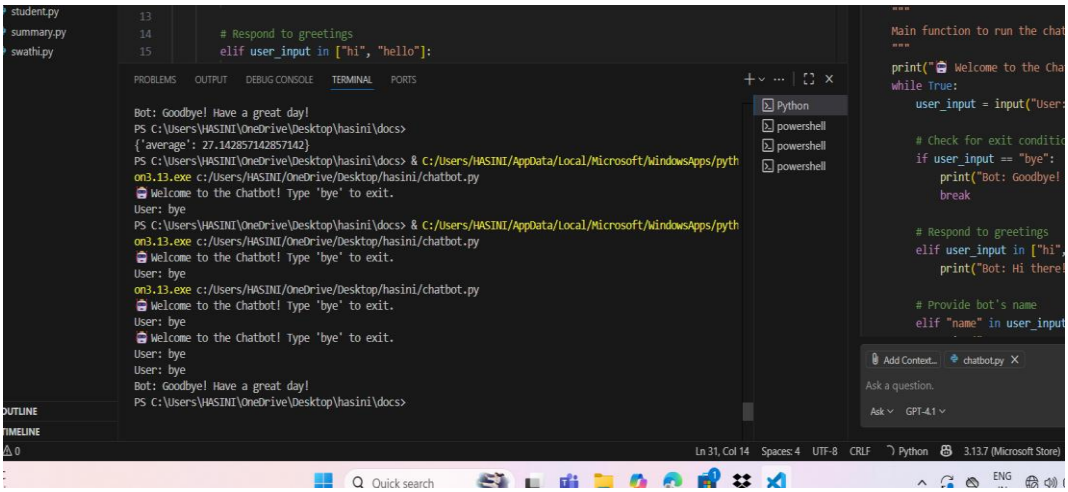
Scenario: You are part of a project team that develops a Chatbot Application. The team needs documentation for maintainability.

- Write a README.md file for the chatbot project (include project description, installation steps, usage, and example).
- Add inline comments in the chatbot's main Python script (focus on explaining logic, not trivial code).
- Use an AI-assisted tool (or simulate it) to generate a usage guide in plain English from your code comments
- Reflect: How does automated documentation help in real-time projects compared to manual documentation?

CODE



OUTPUT



OBSERVATIONS :

By completing this Assignment ,I am able to:

- Compare the auto-generated comments with manually written version using AI Tools.
- Generates the API Documentation Generator using the python scripts .
- Generated the AI_Assisted Code Summarization.
- Implements the Real-Time Project Documentation.