AI Assisted Coding

## ASSIGNEMNT-9.4

## K.Sindhu meenan(2403A51250)-Batch11

Task-1

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.

Prompt:

Add line-by-line comments to the function calculate\_discount(price, discount\_rate).  
Also write a docstring in Google-style or NumPy-style.  
Compare AI-generated comments with manual comments.

Code:

Original function:

A black background with yellow lights

AI-generated content may be incorrect.

AI-Generated Comments (Simulated)-

A screen shot of a computer

AI-generated content may be incorrect.

Manual Comments-

A computer screen with text and numbers

AI-generated content may be incorrect.

Function with NumPy-Style Docstring

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Comparison

AI comments: short and repeats code, little explanation.

Manual comments: detailed, explain parameters, calculation, and reasoning.

Docstring: structured, includes types, return value, and example usage.

Task-2:

A team is building a Library 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)

Prompt:

Write a Python script for a Library Management System with functions add\_book(title, author, year) and issue\_book(book\_id, user\_id).  
Add **docstrings** for each function describing input parameters, output, and function purpose.  
Include print() statements to show example usage.

Code:

A screen shot of a computer

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

Task-3

Scenario: You are reviewing a colleague’s codebase containing long

functions.

def process\_sensor\_data(data):

cleaned = [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.

Prompt:

Explain this function in 2–3 lines, add step-by-step comments, write a short real-life use case paragraph, add a Google-style docstring, and include print statements to show cleaned data, average, anomalies, and final result:

def process\_sensor\_data(data):

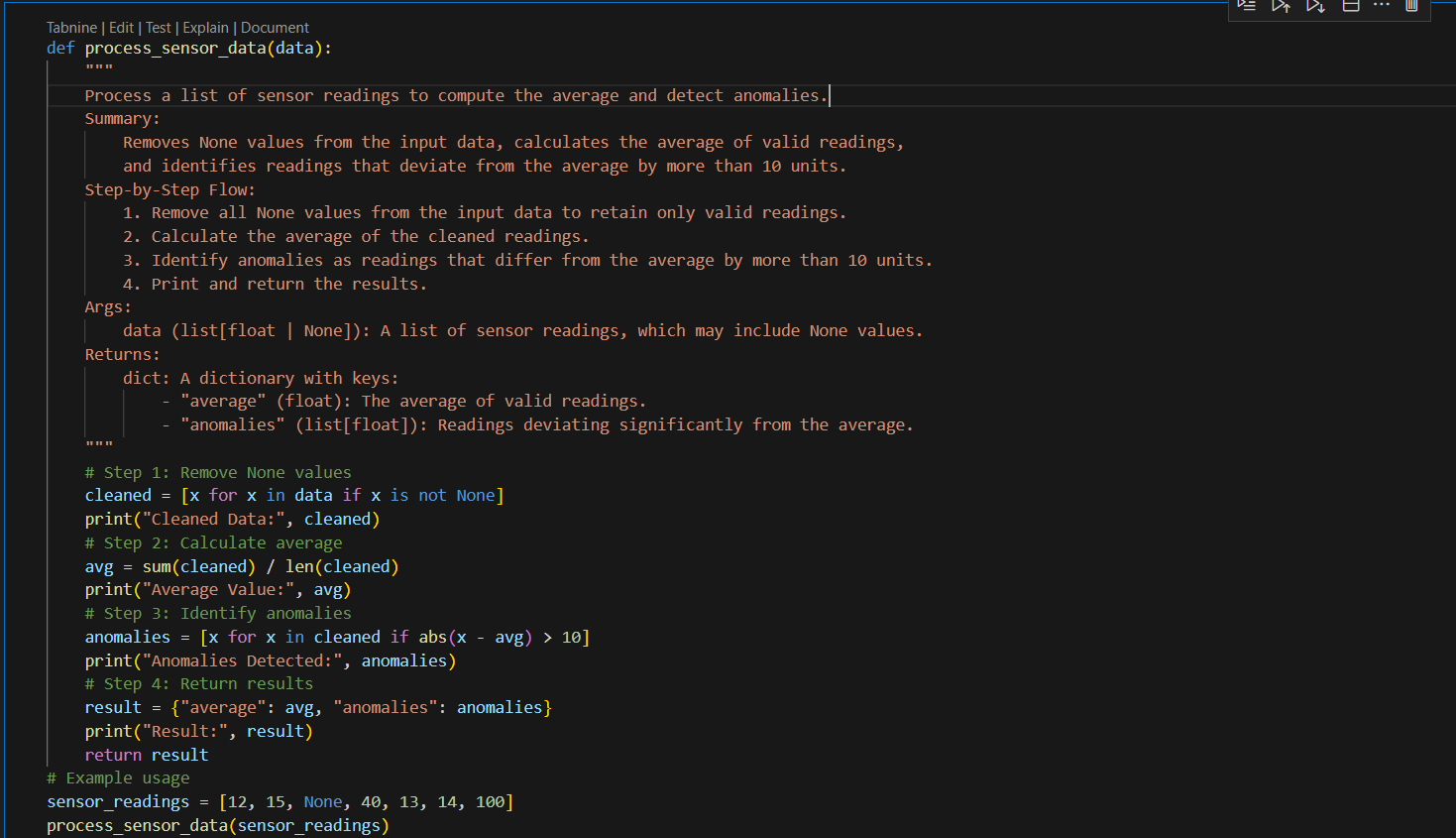
cleaned = [x for x in data if x is not None]

avg = sum(cleaned)/len(cleaned)

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

return {"average": avg, "anomalies": anomalies}

Code:



Output:

A black background with white text

AI-generated content may be incorrect.