

AI-Assisted Documentation & Docstring Assignment Report

Hall Ticket : 2303a51359

Name : A.Abilashgoud

Task 1: Basic Docstring Generation

Prompt:

Write a Python function to return the sum of even and odd numbers in a list. Add manual Google-style docstring. Generate AI docstring. Compare both.

Implementation with Manual Google-Style Docstring:

```
def sum_even_odd(numbers):
    """Calculate sum of even and odd numbers in a list.

    Args:
        numbers (list[int]): List of integers.

    Returns:
        tuple: (sum_even, sum_odd)

    Raises:
        TypeError: If input is not a list.
    """
    # Validate input type
    if not isinstance(numbers, list):
        raise TypeError("Input must be a list")

    sum_even = 0 # Store even sum
    sum_odd = 0 # Store odd sum

    # Iterate through numbers
    for num in numbers:
        if num % 2 == 0:
            sum_even += num
        else:
            sum_odd += num

    return sum_even, sum_odd
```

AI-Generated Docstring (Simulated):

```
"""Return the sum of even and odd numbers from a list.
```

Parameters:

numbers: List of integers.

Returns:

Tuple containing even sum and odd sum.

```
"""
```

Code Explanation:

The function validates input type, iterates through list elements, classifies numbers as even or odd, and returns their respective sums.

Sample Output:

Input: [1,2,3,4,5]

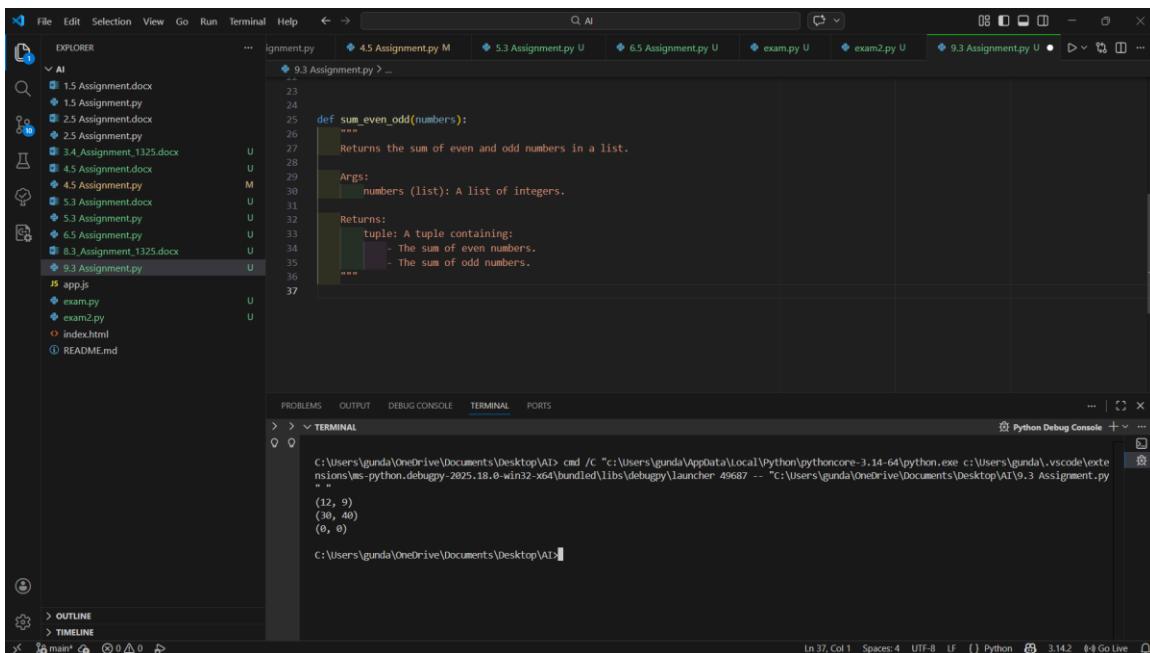
Output: (6, 9)

Comparison:

Manual docstring includes structured sections (Args, Returns, Raises). AI version is concise but lacks error handling details and structured format.

The screenshot shows the Visual Studio Code interface with the following details:

- Explorer View:** Shows a folder named "AI" containing various assignment files (e.g., 1.5 Assignment.docx, 2.5 Assignment.docx, 4.5 Assignment.docx, etc.) and some Python files (e.g., 4.5 Assignment.py, 5.3 Assignment.py, etc.).
- Editor View:** Displays the content of the file "9.3 Assignment.py". The code defines a function `sum_even_odd` that takes a list of numbers, initializes even and odd sums to 0, iterates through the list, and returns a tuple of the two sums. It includes a test case at the bottom.
- Terminal View:** Shows the command line output of running the script. The terminal window shows the command `cmd /C "c:\Users\gunda\AppData\Local\Python\pythoncore-3.14-64\python.exe c:\Users\gunda\vscode\extensions\ms-python.python-2025.18.0-win32-x64\bundled\l1bs\debug\launcher 49687 -- "c:\Users\gunda\OneDrive\Documents\Desktop\AI\9.3 Assignment.py` followed by the expected output `(12, 9)`, `(30, 40)`, and `(0, 0)`.



Task 2: Automatic Inline Comments

Prompt:

Create sru_student class with attributes and methods. Add manual inline comments and compare with AI-generated comments.

Manual Inline Comment Version:

```

class sru_student:
    # Initialize student object
    def __init__(self, name, roll_no, hostel_status):
        self.name = name # Student name
        self.roll_no = roll_no # Student roll number
        self.hostel_status = hostel_status # Hostel status (Yes/No)

    # Update student fee
    def fee_update(self, amount):
        self.fee = amount

    # Display student details
    def display_details(self):
        print("Name:", self.name)
        print("Roll No:", self.roll_no)
        print("Hostel Status:", self.hostel_status)

```

AI-Generated Inline Comments (Simulated):

```
class sru_student:  
    def __init__(self, name, roll_no, hostel_status):  
        # Constructor method  
        self.name = name  
        self.roll_no = roll_no  
        self.hostel_status = hostel_status  
  
    def fee_update(self, amount):  
        # Assign fee value  
        self.fee = amount  
  
    def display_details(self):  
        # Print details  
        print("Name:", self.name)  
        print("Roll No:", self.roll_no)  
        print("Hostel Status:", self.hostel_status)
```

Code Explanation:

The class stores student details, updates fee information, and prints student information.

Sample Output:

Name: Abhilash

Roll No: 101

Hostel Status: True

Fee:5000

Comparison:

Manual comments explain logic clearly. AI comments are shorter and sometimes obvious. AI may miss context or deeper explanation.

A screenshot of the Visual Studio Code (VS Code) interface. The top menu bar includes File, Edit, Selection, View, Go, Run, Terminal, Help, and a search bar. The left sidebar has an Explorer section with files ass.py and b_question.py under a PYTHON folder. The main editor window shows a Python script with code for a student class and its methods. The terminal below the editor shows the command PS C:\Users\anapu\Downloads\python & c:/Users/anapu/AppData/Local/Python/pythoncore-3.14-64/python.exe c:/Users/anapu/Downloads/lab_assignment_9.3.py and the output of the script's execution. The status bar at the bottom indicates the file is lab_assignment_9.3.py, the line number is 20, column 1, and the date is 3/14/2023.

```
1 class sru_student:
2     def __init__(self, name, roll_no, hostel_status):
3         self.name = name
4         self.roll_no = roll_no
5         self.hostel_status = hostel_status
6         self.fee = None
7
8     def fee_update(self, amount):
9         self.fee = amount
10
11     def display_details(self):
12
13         print("Name:", self.name)
14         print("Roll No:", self.roll_no)
15         print("Hostel Status:", self.hostel_status)
16         print("Fee:", self.fee if self.fee is not None else "Not updated")
17
18 student1 = sru_student("Abhilash", 101, True)
19 student1.fee_update(5000)
20 student1.display_details()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\anapu\Downloads> & c:/Users/anapu/AppData/Local/Python/pythoncore-3.14-64/python.exe c:/Users/anapu/Downloads/lab_assignment_9.3.py

Name: Abhilash
Roll No: 101
Hostel Status: True
Fee: 5000

PS C:\Users\anapu\Downloads>

A second screenshot of the VS Code interface, similar to the first but with a different terminal output. The terminal shows the same command and script execution, but the output is slightly different, indicating a change in the script's logic or environment. The status bar at the bottom indicates the file is lab_assignment_9.3.py, the line number is 21, column 1, and the date is 3/14/2023.

```
21 class sru_student:
22     def __init__(self, name, roll_no, hostel_status):
23         self.name = name
24         self.roll_no = roll_no
25         self.hostel_status = hostel_status
26         self.fee = None
27
28     def fee_update(self, amount):
29         self.fee = amount
30
31     def display_details(self):
32
33         print("Name:", self.name)
34         print("Roll No:", self.roll_no)
35         print("Hostel Status:", self.hostel_status)
36         print("Fee:", self.fee if self.fee is not None else "Not updated")
37 student1 = sru_student("Abhilash", 101, True)
38 student1.fee_update(5000)
39 student1.display_details()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

ds/lab_assignment_9.3.py

Name: Abhilash
Roll No: 101
Hostel Status: True
Fee: 5000

PS C:\Users\anapu\Downloads>

Task 3: Module-Level and Function-Level Documentation

Prompt:

Create calculator functions. Add manual NumPy-style docstrings. Generate AI module-level docstring. Compare both.

Manual NumPy-Style Function Docstring:

```
def add(a, b):
    """Add two numbers.
```

Parameters

a : int or float
b : int or float

Returns

int or float
Sum of a and b.

"""

return a + b

```
def divide(a, b):  
    """Divide two numbers.
```

Parameters

a : int or float
b : int or float

Returns

float

Raises

ZeroDivisionError
"""
if b == 0:
 raise ZeroDivisionError("Cannot divide by zero")
return a / b

AI-Generated Module-Level Docstring (Simulated):

"""
This module provides basic arithmetic operations such as addition,
subtraction, multiplication, and division.

11

Code Explanation:

The calculator module performs arithmetic operations with structured documentation for better maintainability.

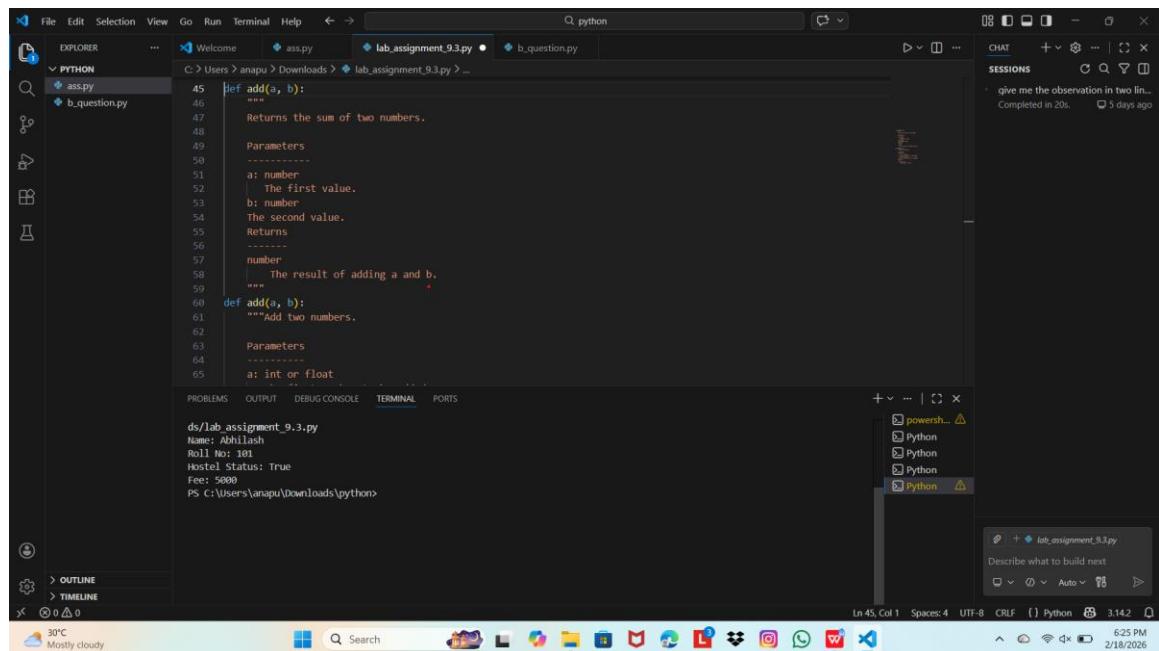
Sample Output:

`add(5,3) → 8`

divide(10,2) → 5.0

Comparison:

Manual NumPy-style docstrings are more structured and detailed. AI-generated module documentation is concise but lacks parameter depth.



The screenshot shows the Visual Studio Code (VS Code) interface. The left sidebar has a 'PYTHON' folder containing 'ass.py' and 'b_question.py'. The main editor window displays the contents of 'ass.py'. The terminal at the bottom shows a session where a script named 'lab_assignment_9.3.py' is running, displaying user information like Name: Abhilash, Roll No: 101, Hostel Status: True, and Fee: 5000.

```
44 def add(a, b):
45     """
46     Returns the sum of two numbers.
47
48     Parameters
49     ----------
50     a: number
51         The first value.
52     b: number
53         The second value.
54
55     Returns
56     -------
57     number
58         The result of adding a and b.
59
60     """
61
62
63
```

```
ds/lab_assignment_9.3.py
Name: Abhilash
Roll No: 101
Hostel Status: True
Fee: 5000
PS C:\Users\anapu\Downloads\python>
```