

Assignment-9.3

Name:B.Shravya

HallTicket No:2303A51492

Batch:08

Lab 9: Documentation Generation – Automatic Documentation and Scenario

You are developing a utility function that processes numerical lists and must be properly documented for future maintenance.

Task 1: Basic Docstring Generation

A.Manual Implementation

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

    Args:
        numbers (list of int): A list containing integer values.

    Returns:
        tuple: A tuple containing two values:
            - sum_even (int): Sum of all even numbers.
            - sum_odd (int): Sum of all odd numbers.

    Raises:
        TypeError: If the input is not a list of integers.
    """
    if not isinstance(numbers, list):
        raise TypeError("Input must be a list")

    sum_even = 0
    sum_odd = 0

    for num in numbers:
        if not isinstance(num, int):
            raise TypeError("All elements must be integers")

        if num % 2 == 0:
            sum_even += num
        else:
            sum_odd += num

    return sum_even, sum_odd
```

B.Prompt Used for AI Generation

Generate a Google Style docstring for the following Python function.

Include Args, Returns, and Raises sections.

```
def sum_even_odd(numbers):
    """
    Returns the sum of even and odd numbers from a list.

    Parameters:
        numbers (list): List of integers.

    Returns:
        tuple: (even_sum, odd_sum)
    """
    even_sum = 0
    odd_sum = 0
    for num in numbers:
        if num % 2 == 0:
            even_sum += num
        else:
            odd_sum += num
    return (even_sum, odd_sum)
```

Task 2: Automatic Inline Comments

Scenario

Develop a student management module that is easy for new developers to understand.

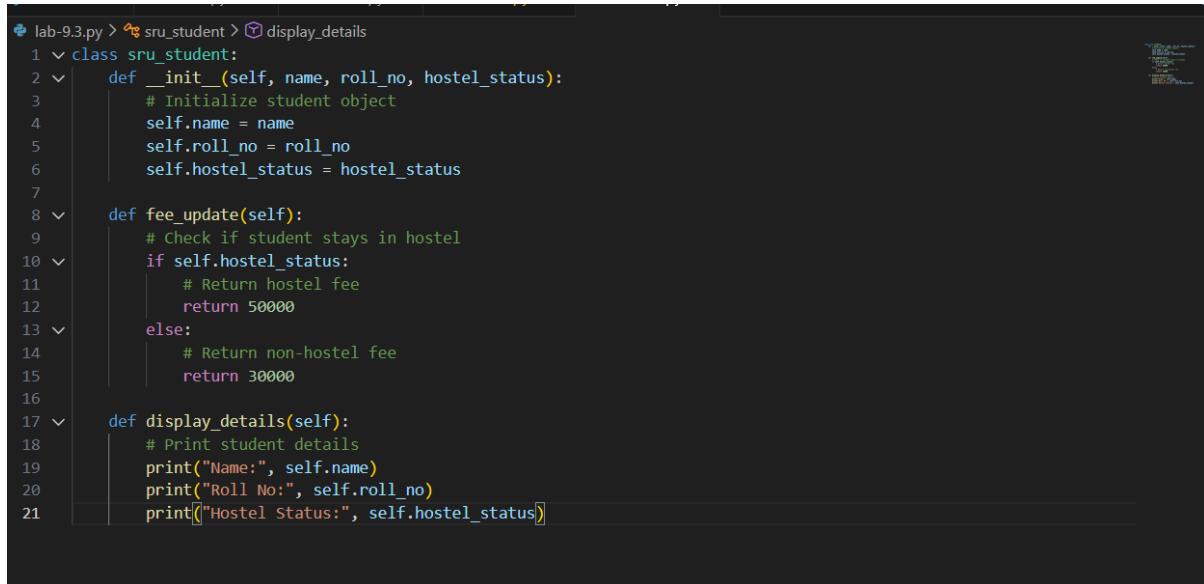
A.Manual Inline Comments

```
lab-9.3.py > ...
1  class sru_student:
2      # Constructor to initialize student details
3      def __init__(self, name, roll_no, hostel_status):
4          self.name = name          # Student name
5          self.roll_no = roll_no    # Student roll number
6          self.hostel_status = hostel_status # Hostel status (True/False)
7
8      # Method to update student fee based on hostel status
9      def fee_update(self):
10         if self.hostel_status:
11             return 50000 # Fee for hostel students
12         else:
13             return 30000 # Fee for day scholars
14
15     # Method to display student details
16     def display_details(self):
17         print("Name:", self.name)
18         print("Roll No:", self.roll_no)
19         print("Hostel Status:", self.hostel_status)
20 # Example usage
```

B.Prompt Used:

Add meaningful inline comments to the following Python class.

Explain each logical block clearly.



```
lab-9.3.py > srstudent > display_details
1 < class srstudent:
2   < def __init__(self, name, roll_no, hostel_status):
3     # Initialize student object
4     self.name = name
5     self.roll_no = roll_no
6     self.hostel_status = hostel_status
7
8   < def fee_update(self):
9     # Check if student stays in hostel
10    if self.hostel_status:
11      # Return hostel fee
12      return 50000
13    else:
14      # Return non-hostel fee
15      return 30000
16
17   < def display_details(self):
18     # Print student details
19     print("Name:", self.name)
20     print("Roll No:", self.roll_no)
21     print("Hostel Status:", self.hostel_status)
```

Task 3: Module-Level & Function-Level Documentation

Scenario

Develop a reusable calculator module with structured documentation.

A. Manual NumPy Style Documentation

```
lab-9.3.py > add
1  C:\Users\shraw\OneDrive\Desktop\AI Assistant Coding\lab-9.3.py
2  calculator_module.py
3
4  A simple calculator module providing basic arithmetic operations.
5  """
6  def add(a, b):
7      """
8          Add two numbers.
9
10         Parameters
11         -----
12         a : int or float
13             First number.
14         b : int or float
15             Second number.
16
17         Returns
18         -----
19         int or float
20             Sum of a and b.
21         """
22         return a + b
23     def subtract(a, b):
24         """
25             Subtract two numbers.
26
27             Parameters
28             -----
29             a : int or float
30             b : int or float
31
32             Returns
33             -----
34             int or float
35             Difference of a and b.
36         """
37         return a - b
```

```

lab-9.3.py > add
38 def multiply(a, b):
39     """
40     Multiply two numbers.
41
42     Parameters
43     -----
44     a : int or float
45     b : int or float
46
47     Returns
48     -----
49     int or float
50     |
51     | Product of a and b.
52     |
53     return a * b
54 def divide(a, b):
55     """
56     Divide two numbers.
57
58     Parameters
59     -----
60     a : int or float
61     b : int or float
62
63     Returns
64     -----
65     float
66     |
67     | Result of division.
68
69     Raises
70     -----
71     ZeroDivisionError
72     |
73     | If b is zero.
74     |
75     if b == 0:
76         raise ZeroDivisionError("Cannot divide by zero")
77     return a / b

```

B.Prompt Used:

Generate NumPy Style docstrings and a professional module-level

docstring for this calculator module.

```

lab-9.3.py > add
1 """
2 This module provides basic arithmetic operations such as addition,
3 subtraction, multiplication, and division. It can be reused in
4 multiple projects requiring simple mathematical calculations.
5 """
6 def add(a, b):
7     """Return the sum of a and b."""
8     return a + b

```

Conclusion:

This lab demonstrated the importance of structured documentation in software development.

Key Learnings:

- AI-assisted tools help generate documentation quickly.
- AI-generated comments are useful but require human review.
- Manual documentation ensures completeness and accuracy.
- Google and NumPy styles improve readability and maintainability.
- AI is a productivity tool, not a replacement for developer responsibility.

Overall, combining manual expertise with AI assistance produces the best quality documentation.