Lab Report 9.2

Name:B.RAJAGOPAL

Roll Number: 2503A51L01

Course Code: 24CS002PC215

Course Title: AI Assisted Coding

Assignment Number: 9.2

Academic Year: 2025-2026

Branch :cse

Lab Objectives

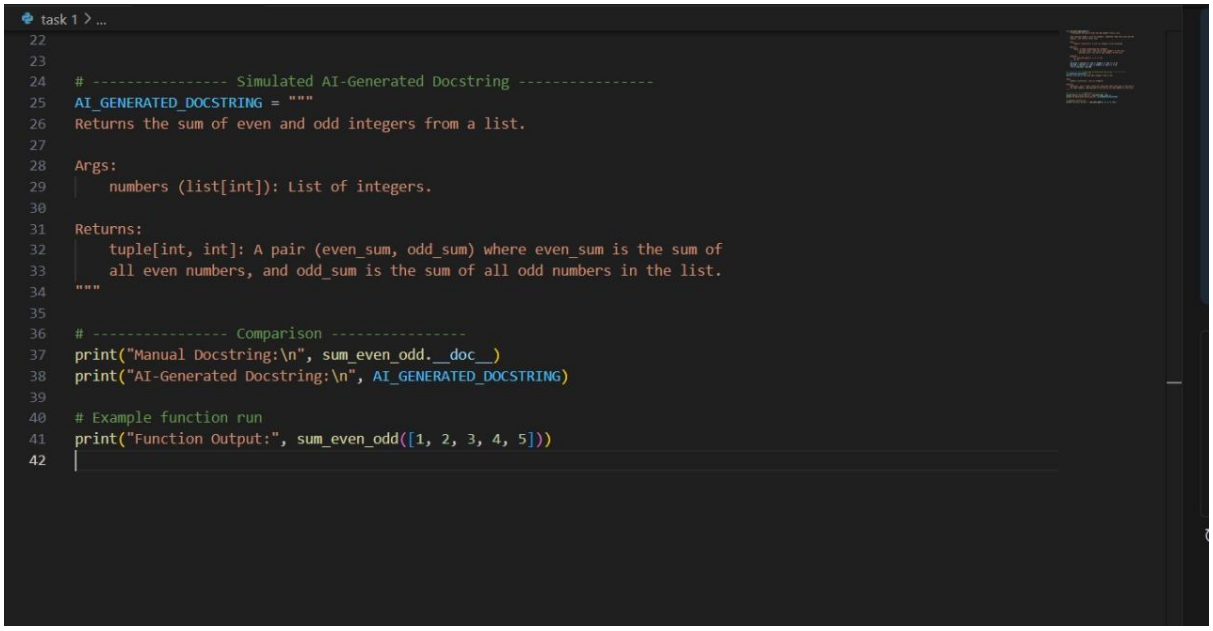
- To explore AI-powered auto-completion features for core Python constructs.
- To analyze how AI suggests logic for class definitions, loops, and conditionals.
- To evaluate the completeness and correctness of code generated by AI assistants.

Lab Outcomes

- Use AI tools to generate and complete class definitions and methods.
- Understand and assess AI-suggested loops for iterative tasks.
- Generate conditional statements through prompt-driven suggestions.
- Critically evaluate AI-assisted code for correctness and clarity.

TASK :-1

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

    This function takes a list of integers, separates them into even and odd
    numbers, and returns their sums.

    Args:
        numbers (list[int]): A list of integers to be processed.

    Returns:
        tuple: A tuple containing two integers:
            - sum_even (int): The sum of even numbers in the list.
            - sum_odd (int): The sum of odd numbers in the list.

    Example:
        >>> sum_even_odd([1, 2, 3, 4, 5])
        (6, 9)
    """
    sum_even = sum(num for num in numbers if num % 2 == 0)
    sum_odd = sum(num for num in numbers if num % 2 != 0)
    return sum_even, sum_odd


# ---------------- Simulated AI-Generated Docstring ----------------
AI_GENERATED_DOCSTRING = """
Returns the sum of even and odd integers from a list.

Args:
```



```python
task 1 > ...


# ---------------- Simulated AI-Generated Docstring ----------------
AI_GENERATED_DOCSTRING = """
Returns the sum of even and odd integers from a list.

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

Returns:
    tuple[int, int]: A pair (even_sum, odd_sum) where even_sum is the sum of
    all even numbers, and odd_sum is the sum of all odd numbers in the list.
"""

# ---------------- Comparison ----------------
print("Manual Docstring:\n", sum_even_odd.__doc__)
print("AI-Generated Docstring:\n", AI_GENERATED_DOCSTRING)

# Example function run
print("Function Output:", sum_even_odd([1, 2, 3, 4, 5]))

```

Observation:-

Manual (Google Style):

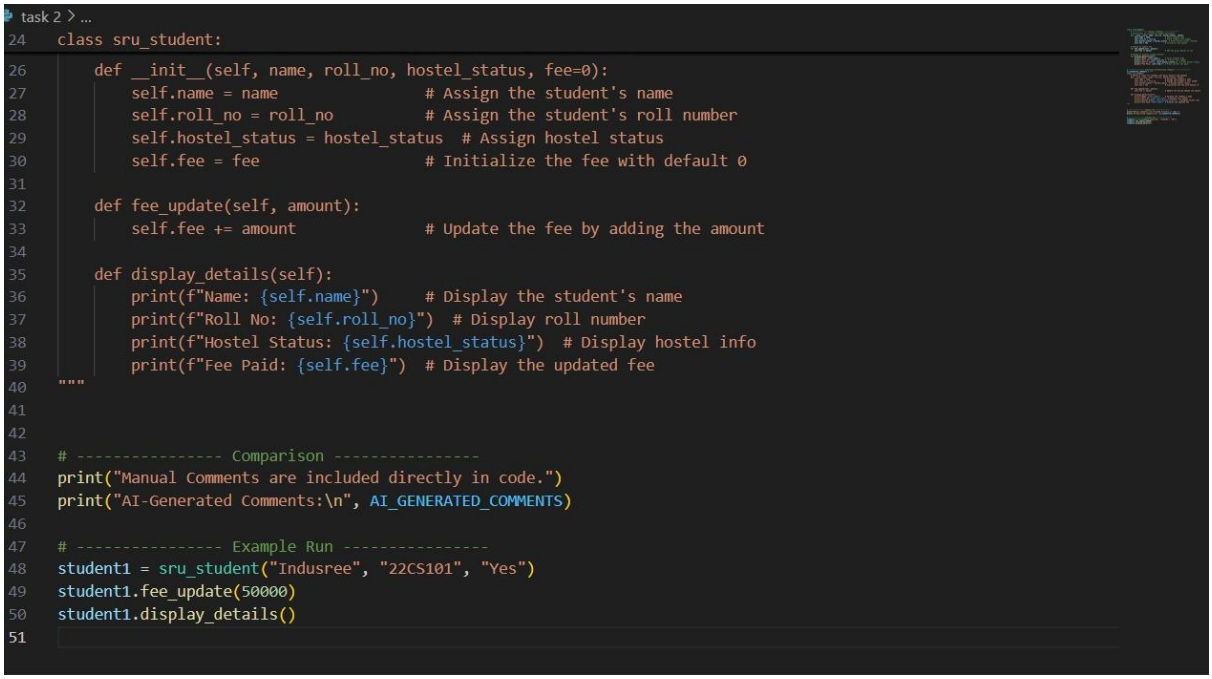
- More **structured and detailed**.
- Includes **Examples** section (useful for quick understanding).
- Explicitly states return values with explanation.

Output:-

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS                    Python

Returns the sum of even and odd integers from a list.

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

Returns:
    tuple[int, int]: A pair (even_sum, odd_sum) where even_sum is the sum of
    all even numbers, and odd_sum is the sum of all odd numbers in the list.

Function Output: (6, 9)
PS C:\Users\indus\OneDrive\Desktop\lab 9.3>
```

TASK:-2

```
task 2 > ...
1   class sru_student:
2       # ---------------- Manual Comments ----------------
3       # Constructor to initialize student attributes
4       def __init__(self, name, roll_no, hostel_status, fee=0):
5           self.name = name                # Store student name
6           self.roll_no = roll_no          # Store student roll number
7           self.hostel_status = hostel_status  # Store hostel status (Yes/No)
8           self.fee = fee                  # Initialize fee amount
9
10      # Method to update fee
11      def fee_update(self, amount):
12          self.fee += amount              # Add the given amount to fee
13
14      # Method to display student details
15      def display_details(self):
16          print(f"Name: {self.name}")     # Print student name
17          print(f"Roll No: {self.roll_no}")  # Print roll number
18          print(f"Hostel Status: {self.hostel_status}")  # Print hostel status
19          print(f"Fee Paid: {self.fee}")  # Print current fee paid
20
21
22  # ---------------- Simulated AI-Generated Comments ----------------
23  AI_GENERATED_COMMENTS = """
24  class sru_student:
25      # Define a class for student with basic details and methods
26      def __init__(self, name, roll_no, hostel_status, fee=0):
27          self.name = name                # Assign the student's name
28          self.roll_no = roll_no          # Assign the student's roll number
29          self.hostel_status = hostel_status  # Assign hostel status
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS                    Python
```



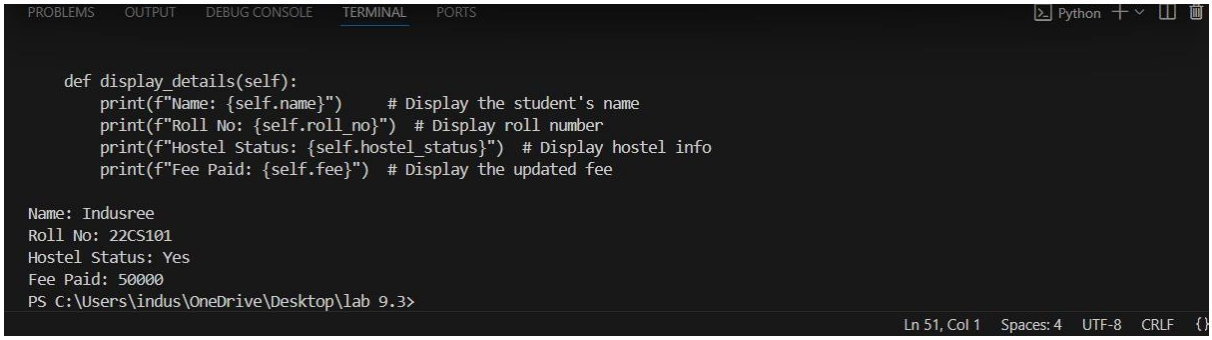
Observation:-

1. **Manual Comments:**
    - **Concise and to the point.**
    - **Written in human-friendly language (like a teacher explaining).**
    - **Sometimes slightly less formal but quicker to read.**
2. **AI-Generated Comments:**
    - **More verbose and descriptive, explaining each line in detail.**
    - **Uses technical phrasing (e.g., *"assign the provided value"*) instead of simpler wording.**
    - **Good for beginners, but may feel repetitive for experienced programmers.**

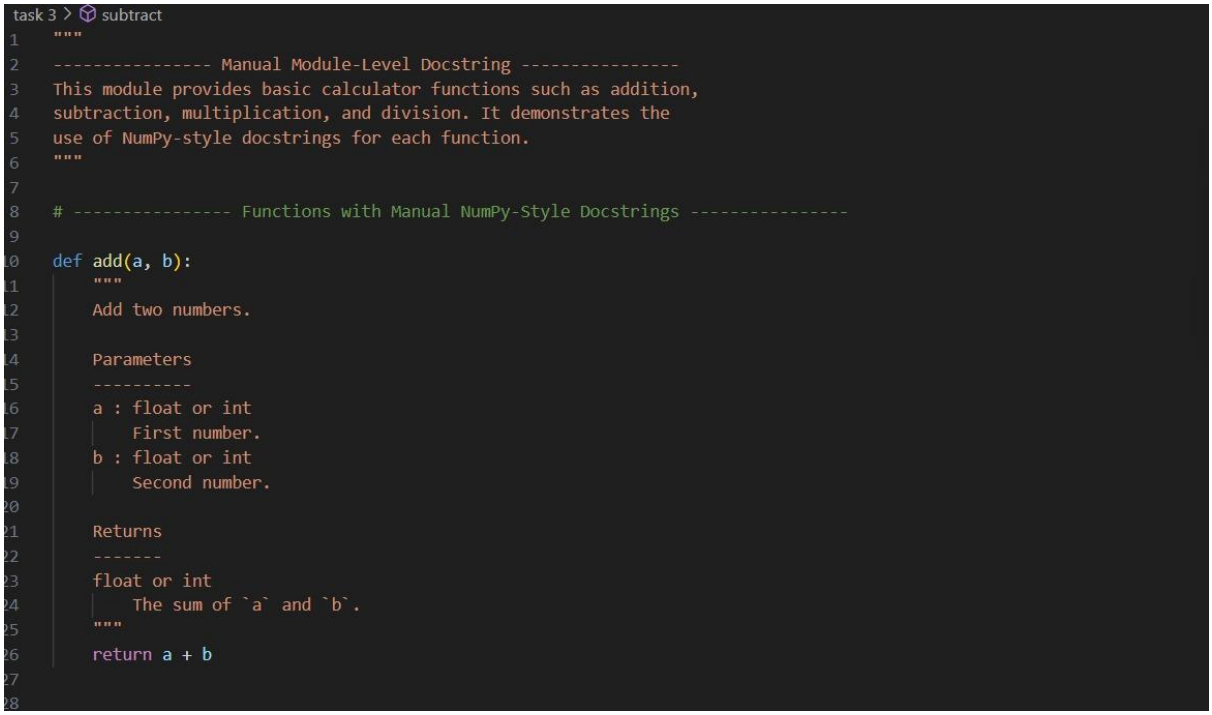
Conclusion:

- **Manual comments are better for readability and give a summary explanation.**
- **AI-generated comments are thorough and line-by-line, which can be helpful for learning or code reviews, but may be overly detailed in larger programs.**

Output:-



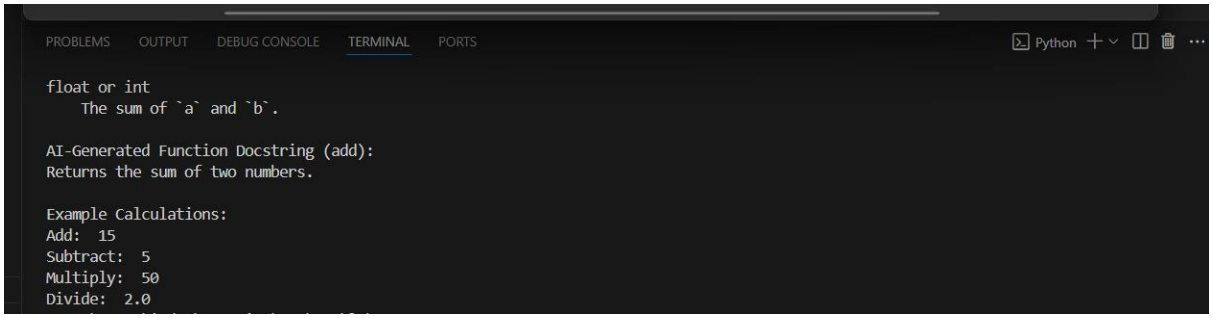
TASK:-3



Observation:-

- **Well-structured with Parameters, Returns, Raises, Examples.**
- **Very useful for both developers and users of the code.**
- **Consistent across all functions.**
- **Slightly verbose, but excellent for documentation standards.**

Output:-


PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Python
float or int
The sum of `a` and `b`.
AI-Generated Function Docstring (add):
Returns the sum of two numbers.
Example Calculations:
Add: 15
Subtract: 5
Multiply: 50
Divide: 2.0