SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE		DEPARTMENT OF COMPUTER SCIENCE ENGINEERING	
ProgramName:B. Tech		**Assignment Type: Lab**	**AcademicYear:**2025-2026
CourseCoordinatorName	Venkataramana Veeramsetty		
Instructor(s)Name	Dr. V. Venkataramana (Co-ordinator) Dr. T. Sampath Kumar Dr. Pramoda Patro Dr. Brij Kishor Tiwari Dr.J.Ravichander Dr. Mohammand Ali Shaik Dr. Anirodh Kumar Mr. S.Naresh Kumar Dr. RAJESH VELPULA Mr. Kundhan Kumar Ms. Ch.Rajitha Mr. M Prakash Mr. B.Raju Intern 1 (Dharma teja) Intern 2 (Sai Prasad) Intern 3 (Sowmya) NS_2 (Mounika)		
CourseCode	24CS002PC215	**CourseTitle**	AI Assisted Coding
Year/Sem	II/I	**Regulation**	R24
Date and Day of Assignment	Week4 - Wednesday	**Time(s)**	
Duration	2 Hours	**Applicableto Batches**	
AssignmentNumber:9.3(Present assignment number)/**24**(Total number of assignments)			

Q.No.	Question	*ExpectedTime to complete*
1	Lab 8: Documentation Generation: Automatic documentation and code comments **Lab Objectives:** • To understand the importance of documentation and code comments in software development. • To explore how AI-assisted coding tools can generate meaningful documentation and	Week4 - Wednesday

inline comments.

- To practice generating function-level and module-level docstrings automatically.
- To evaluate the quality, accuracy, and limitations of AI-generated documentation.
- To develop a small automated tool for documentation generation in Python..

Lab Outcomes (LOs):
After completing this lab, students will be able to:

- Apply AI-assisted coding tools to generate docstrings and inline comments for Python code.
- Critically analyze AI-generated documentation for correctness, completeness, and readability.
- Create structured documentation (function-level, module-level) following standard formats.

- Design and implement a mini documentation generator tool to automate code commenting and docstring creation.

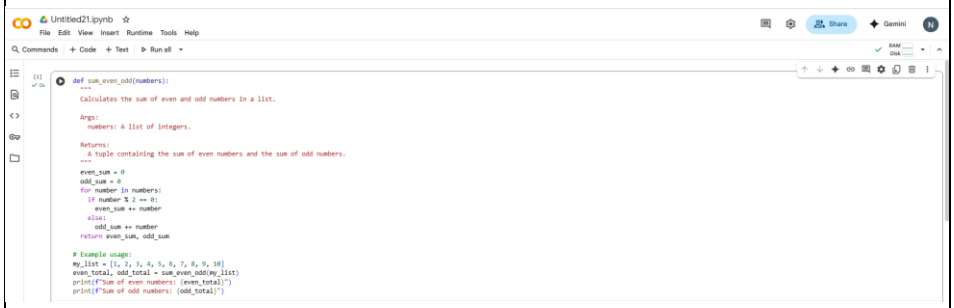
Task Description#1 Basic Docstring Generation

- Write python function to return sum of even and odd numbers in the given list.
- Incorporate manual **docstring** in code with Google Style
- Use an AI-assisted tool (e.g., Copilot, Cursor AI) to generate a docstring describing the function.
- Compare the AI-generated docstring with your manually written one.

Expected Outcome#1: Students understand how AI can produce function-level documentation.

PROMPT1:write a python code to return the sum of even and odd numbers in the list using function
PROMPT2:take list numbers after the execution
Prompt3:generate the docstring describing the code

Code:



OUTPUT:

```
Sum of even numbers: 30
Sum of odd numbers: 25
```

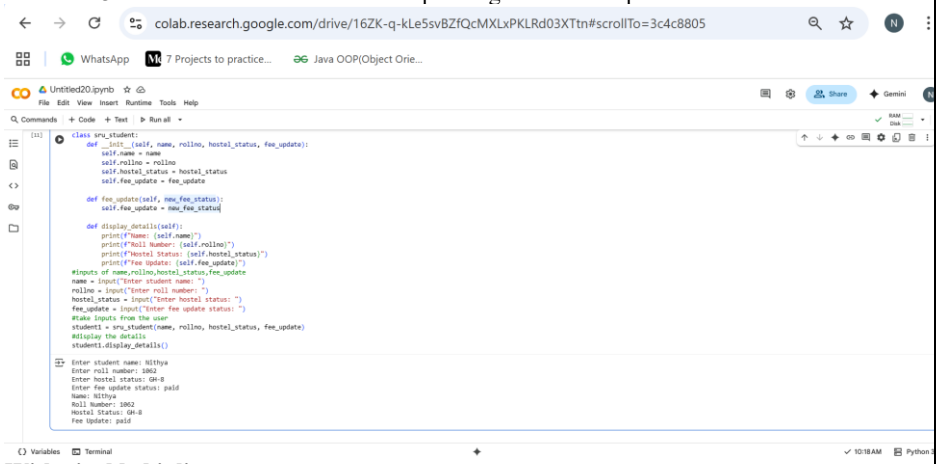
OBSERVATION:
In the above the code in the input I will be some integer numbers that contains all the numbers in the list. Then it will identify the which number is even and which number is off from the list then in the Output it print the even numbers in a list and also the odd

numbers in a list.

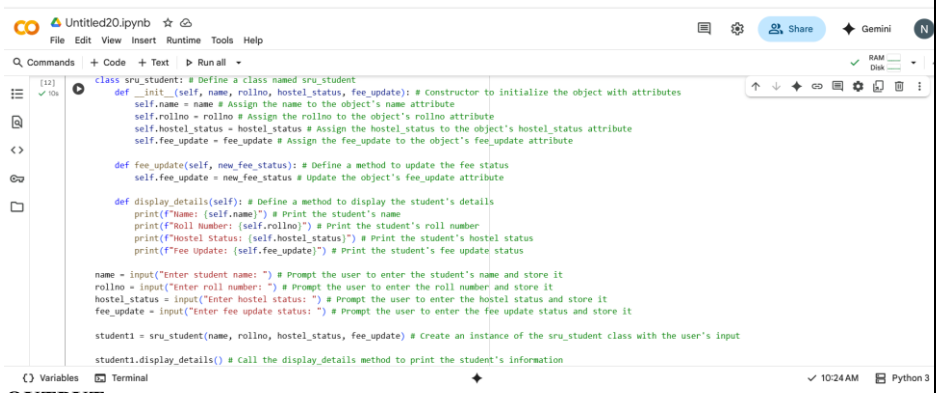
Task Description#2 Automatic Inline Comments

- Write python program for **sru_student** class with attributes like name, roll no., hostel_status and **fee_update** method and **display_details** method.
- Write comments manually for each line/code block
- Ask an AI tool to add inline comments explaining each line/step.
- Compare the AI-generated comments with your manually written one.

Expected Output#2: Students critically analyze AI-generated code comments.
PROMPT1:write a python code class sru_student with attributes name, rollno, hostel_status, fee_update method and diplay_details method.
PROMPT2:Take inputs from the user.
PROMPT3: Please add inline comments explaining each line/step of the code



With ai added inline comments



OUTPUT:

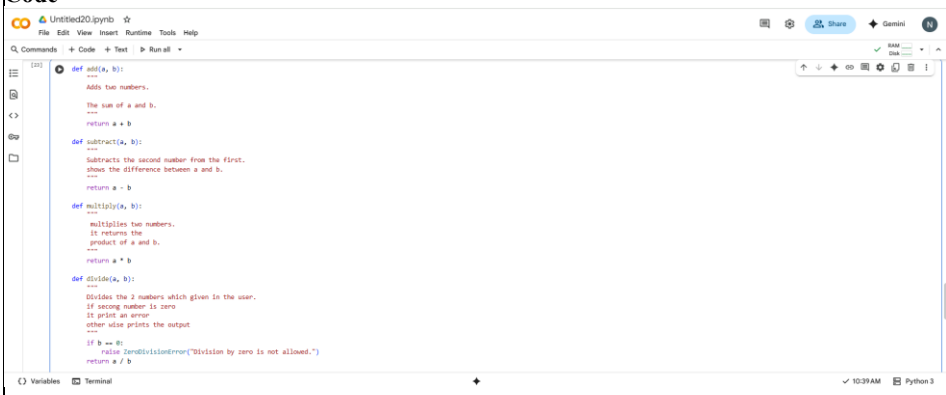
```
Enter student name: Nithya
Enter roll number: 1062
Enter hostel status: GH-8
Enter fee update status: paid
Name: Nithya
Roll Number: 1062
Hostel Status: GH-8
Fee Update: paid
```

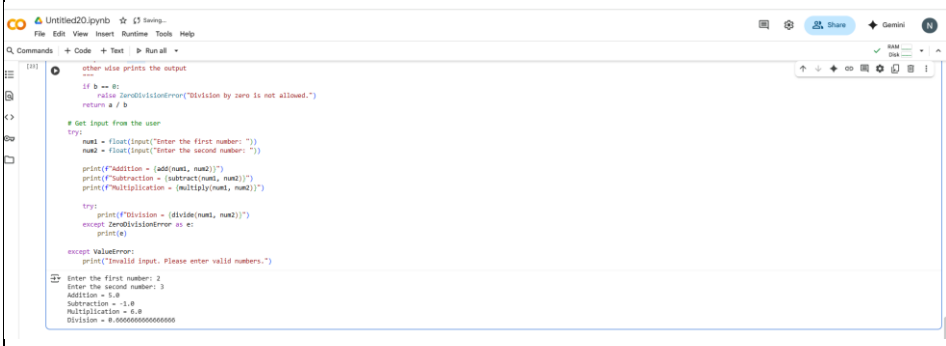
OBSERVATION:
From the above code I observed that I given the code to print the name,rollno , hostel, fee_update from that it collected the inputs and given the output. And also I writhen the comments by my own without ai as shown In first screenshoot. Coming to 2nd screnshoot it given by ai with inline commenst of each line or also we can say given the code of commenting with the each step.

Task Description#3

- Write a Python script with 3–4 functions (e.g., calculator: add, subtract, multiply, divide).
- Incorporate manual **docstring** in code with NumPy Style
- Use AI assistance to generate a module-level docstring + individual function docstrings.
- Compare the AI-generated docstring with your manually written one.

Expected Output#3: Students learn structured documentation for multi-function scripts
PROMPT1: write a python code to add,subtract,multiply,divde using functions.
Writing in manual docstring
PROMPT2: Form the above code generate the a module-level docstring + individual function docstrings.
Code





From ai generated module-level docstring + individual function docstrings.

```
"""
This module provides basic arithmetic functions: addition, subtraction,
multiplication, and division.
"""

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

    Args:
        a: The first number.
        b: The second number.

    Returns:
        The sum of a and b.
    """
    return a + b

def subtract(a, b):
    """
    Subtracts the second number from the first.

    Args:
        a: The number to subtract from.
        b: The number to subtract.

    Returns:
        The difference between a and b.
    """
    return a - b

def multiply(a, b):
    """
     multiplies two numbers.
```

```
    """
    Divides the first number by the second.

    Args:
        a: The dividend.
        b: The divisor.

    Returns:
        The result of the division.

    Raises:
        ZeroDivisionError: If the divisor (b) is zero.
    """
    if b == 0:
        raise ZeroDivisionError("Division by zero is not allowed.")
    return a / b

# Get input from the user
try:
    num1 = float(input("Enter the first number: "))
    num2 = float(input("Enter the second number: "))

    print(f"Addition = {add(num1, num2)}")
    print(f"Subtraction = {subtract(num1, num2)}")
    print(f"Multiplication = {multiply(num1, num2)}")

    try:
        print(f"Division = {divide(num1, num2)}")
    except ZeroDivisionError as e:
        print(e)

except ValueError:
    print("Invalid input. Please enter valid numbers.")
```

OUTPUT:

```
Enter the first number: 2
Enter the second number: 3
Addition = 5.0
Subtraction = -1.0
Multiplication = 6.0
Division = 0.6666666666666666
```

OBSERVATION:

From the above task I observed that I wrriten the code with manually comments .the code is to take the two numbers as first number and second number and prints the addition,subtraction,multiplication and division of the two numbers. Then I given my code to generate the module-level docstring + individual function docstrings. Then it given in the understandable and step by step comments .

Push documentation whole workspace as .md file in GitHub Repository

Note: Report should be submitted a word document for all tasks in a single document with prompts, comments & code explanation, and output and if required, screenshots