ASSIGNMENT 10.2

H.T.N.O : 2403A51292

BATCH : 12

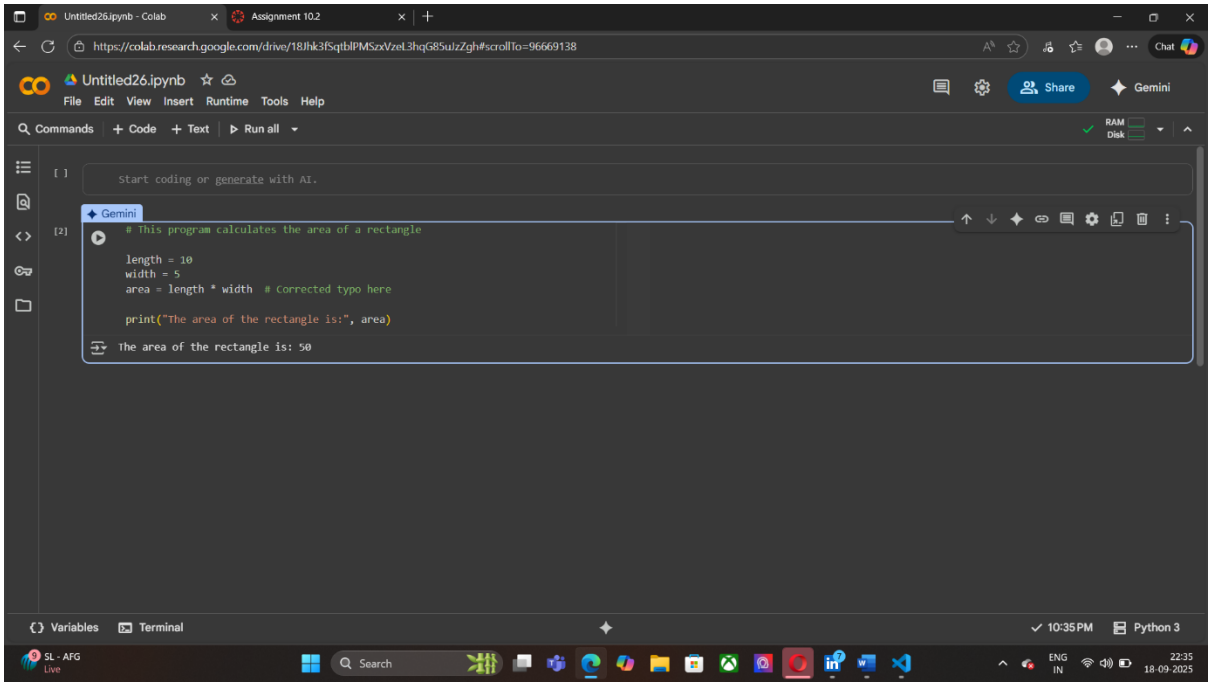
Task Description 1 :

AI-Assisted Code Review (Basic Errors)

• Write python program as shown below.

• Use an AI assistant to review and suggest corrections.

Expected Outcome#1: Students need to submit corrected code with comments.



Prompt:

Write a Python program as instructed.

- Use an AI assistant to review your code and suggest corrections for any errors or improvements.
- Submit the corrected code with comments explaining the changes.

Task Description 2:

Automatic Inline Comments

• Write the Python code for Fibonacci as shown below and execute.

• Ask AI to improve variable names, add comments, and apply PEP8 formatting (cleaned up).

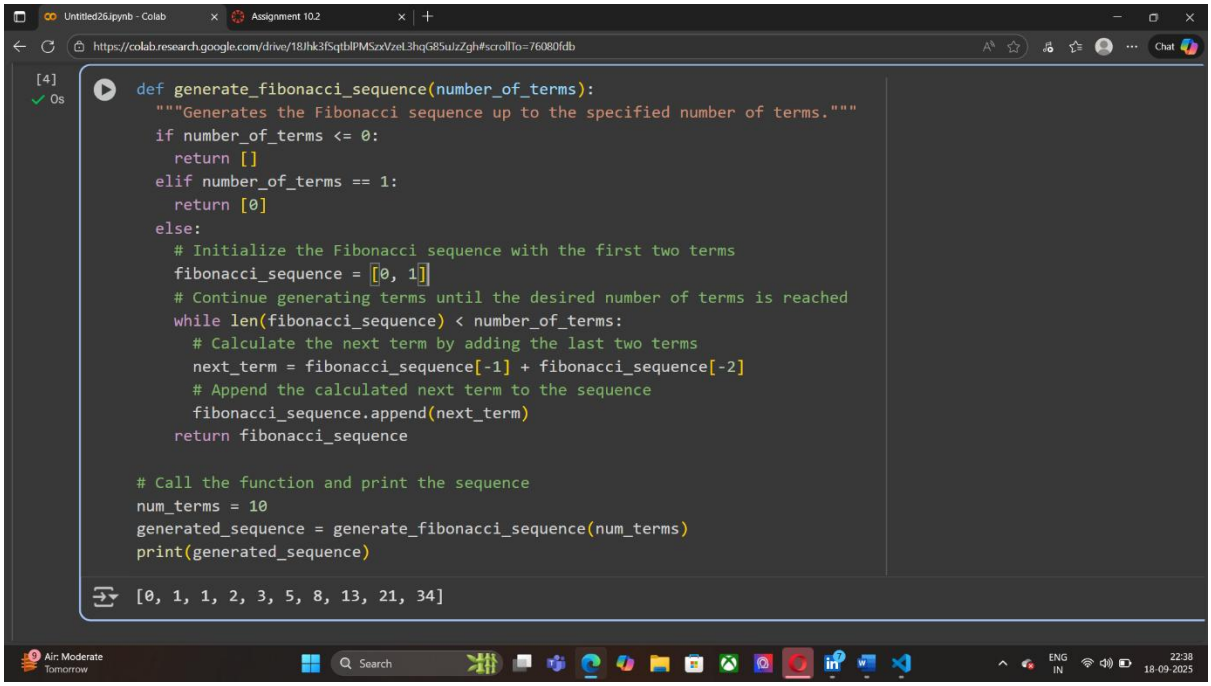
• Students evaluate which suggestions improve readability most. one.

Expected Output#2: Clean format python code with much readability.

Prompt:

Write the Python code for generating Fibonacci numbers as shown below and execute it.

- Ask an AI assistant to improve variable names, add meaningful inline comments, and apply PEP8 formatting for better readability.
- Evaluate which AI suggestions most improve the code's clarity and maintainabilitY



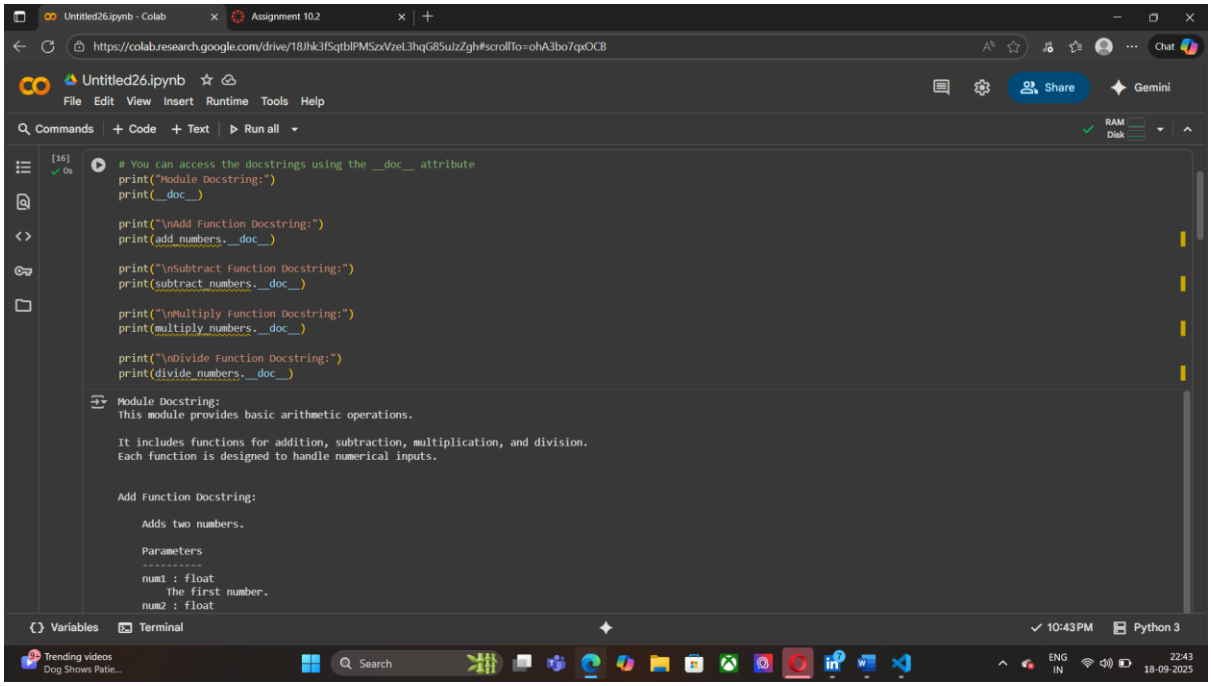
Task Description 3:

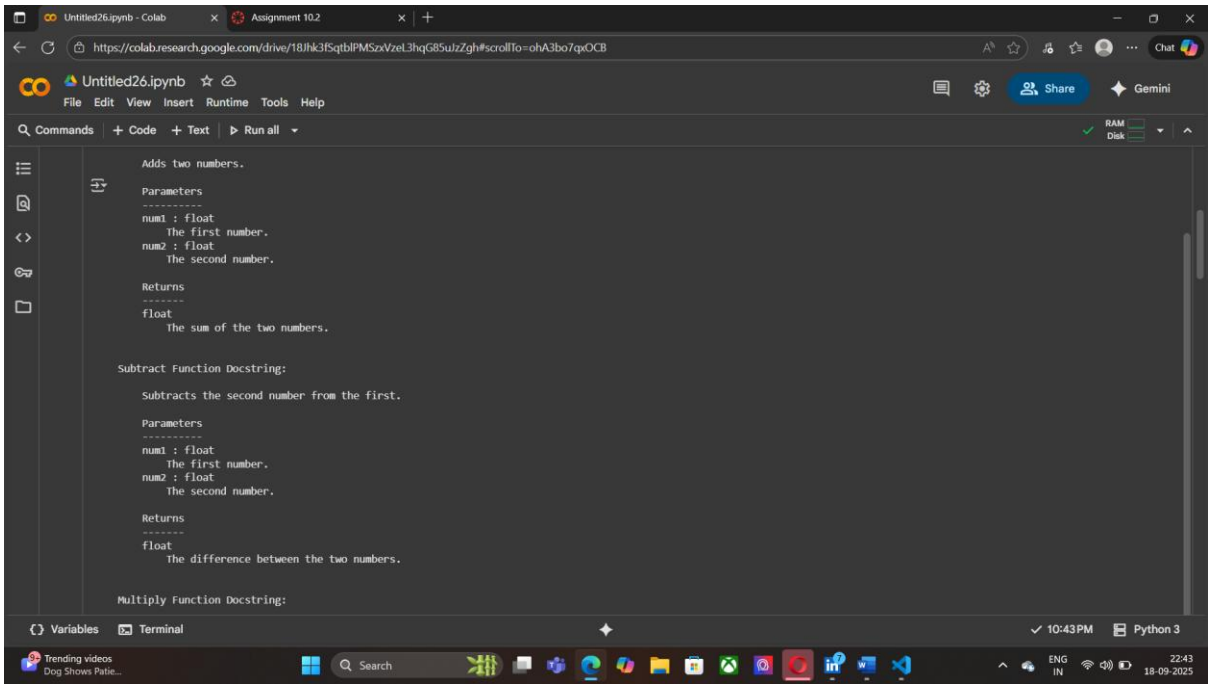
• Write a Python script with 3–4 functions (e.g., calculator: add, subtract, multiply, divide).
• Incorporate manual docstring in code with NumPy Style
• Use AI assistance to generate a module-level docstring + individual function docstrings.
• Compare the AI-generated docstring with your manually written one.
Common Examples of Code Smells
• Long Function – A single function tries to do too many things.
• Duplicate Code – Copy-pasted logic in multiple places.
• Poor Naming – Variables or functions with confusing names (x1, foo, data123).
• Unused Variables – Declaring variables but never using them.
• Magic Numbers – Using unexplained constants (3.14159 instead of PI).
• Deep Nesting – Too many if/else levels, making code hard to read.
• Large Class – A single class handling too many responsibilities.
Why Detecting Code Smells is Important
• Makes code easier to read and maintain.
• Reduces chance of bugs in future updates.
• Helps in refactoring (improving structure without changing behavior).
• Encourages clean coding practices
Dead Code – Code that is never executed.
Expected Output#3: Students learn structured documentation for multi-function scripts
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

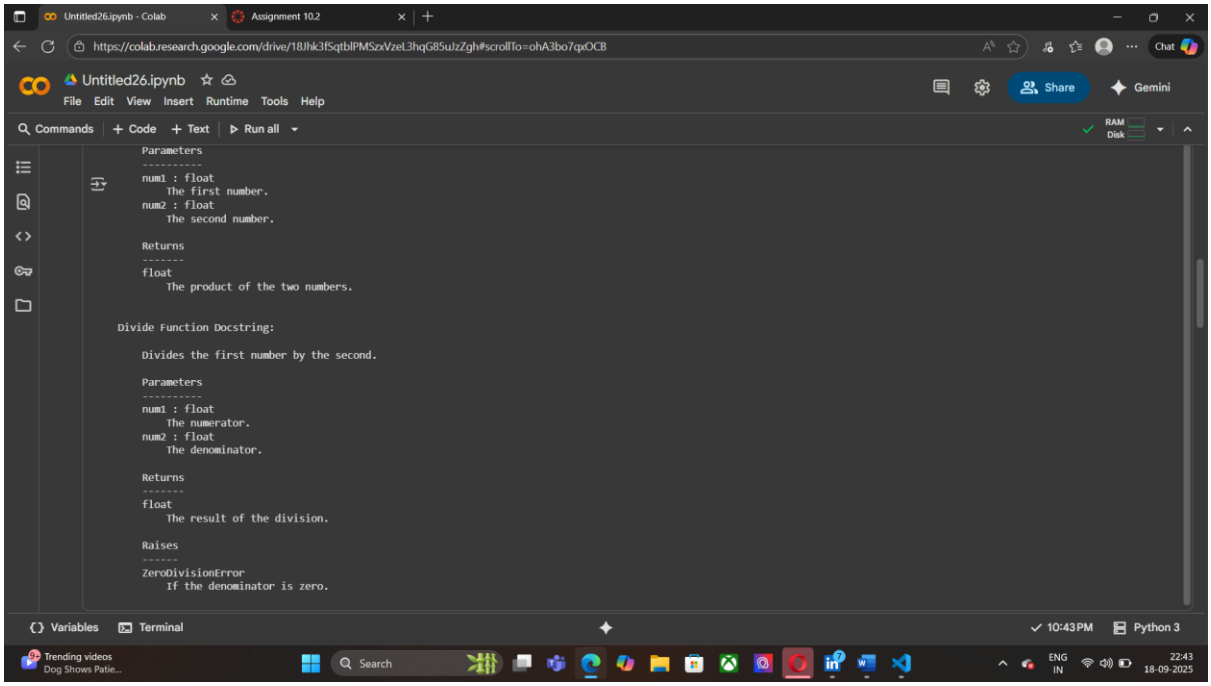
Prompt:

Write a Python script with 3–4 functions (e.g., calculator: add, subtract, multiply, divide).

- Manually add NumPy-style docstrings to each function.
- Use AI assistance to generate a module-level docstring and individual function docstrings.
- Compare the AI-generated docstrings with your manually written ones, noting differences in structure, clarity, and completeness.


Untitled26.ipynb - Colab
Assignment 10.2
https://colab.research.google.com/drive/18Jhk3fSqtblPMSzxVzeL3hqG85uJzZgh#scrollTo=ohA3bo7qxOCB
Untitled26.ipynb
File Edit View Insert Runtime Tools Help
Share
Gemini
Commands + Code + Text Run all
RAM Disk
[16] 0s
# You can access the docstrings using the __doc__ attribute
print("Module Docstring:")
print(__doc__)
print("\nAdd Function Docstring:")
print(add_numbers.__doc__)
print("\nSubtract Function Docstring:")
print(subtract_numbers.__doc__)
print("\nMultiply Function Docstring:")
print(multiply_numbers.__doc__)
print("\nDivide Function Docstring:")
print(divide_numbers.__doc__)
Module Docstring:
This module provides basic arithmetic operations.
It includes functions for addition, subtraction, multiplication, and division.
Each function is designed to handle numerical inputs.
Add Function Docstring:
Adds two numbers.
Parameters
num1 : float
The first number.
num2 : float
Variables Terminal
10:43PM Python 3
Trending videos
Dog Shows Patie...
Search
ENG IN
22:43
18-09-2025


Untitled26.ipynb - Colab
Assignment 10.2
https://colab.research.google.com/drive/18Jhk3fSqtblPMSzxVzeL3hqG85uJzZgh#scrollTo=ohA3bo7qxOCB
Untitled26.ipynb
File Edit View Insert Runtime Tools Help
Share
Gemini
Commands + Code + Text Run all
RAM Disk
Adds two numbers.
Parameters
num1 : float
The first number.
num2 : float
The second number.
Returns
float
The sum of the two numbers.
Subtract Function Docstring:
Subtracts the second number from the first.
Parameters
num1 : float
The first number.
num2 : float
The second number.
Returns
float
The difference between the two numbers.
Multiply Function Docstring:
Variables Terminal
10:43PM Python 3
Trending videos
Dog Shows Patie...
Search
ENG IN
22:43
18-09-2025


Untitled26.ipynb - Colab
Assignment 10.2
https://colab.research.google.com/drive/18Jhk3fSqtblPMSzxVzeL3hqG85uJzZgh#scrollTo=ohA3bo7qxOCB
Untitled26.ipynb
File Edit View Insert Runtime Tools Help
Share
Gemini
Commands
Code
Text
Run all
RAM
Disk
Parameters
num1 : float
The first number.
num2 : float
The second number.
Returns
float
The product of the two numbers.
Divide Function Docstring:
Divides the first number by the second.
Parameters
num1 : float
The numerator.
num2 : float
The denominator.
Returns
float
The result of the division.
Raises
ZeroDivisionError
If the denominator is zero.
Variables
Terminal
10:43PM
Python 3
Trending videos
Dog Shows Patie...
Search
ENG
IN
22:43
18-09-2025