Lab Assignment (9.1)

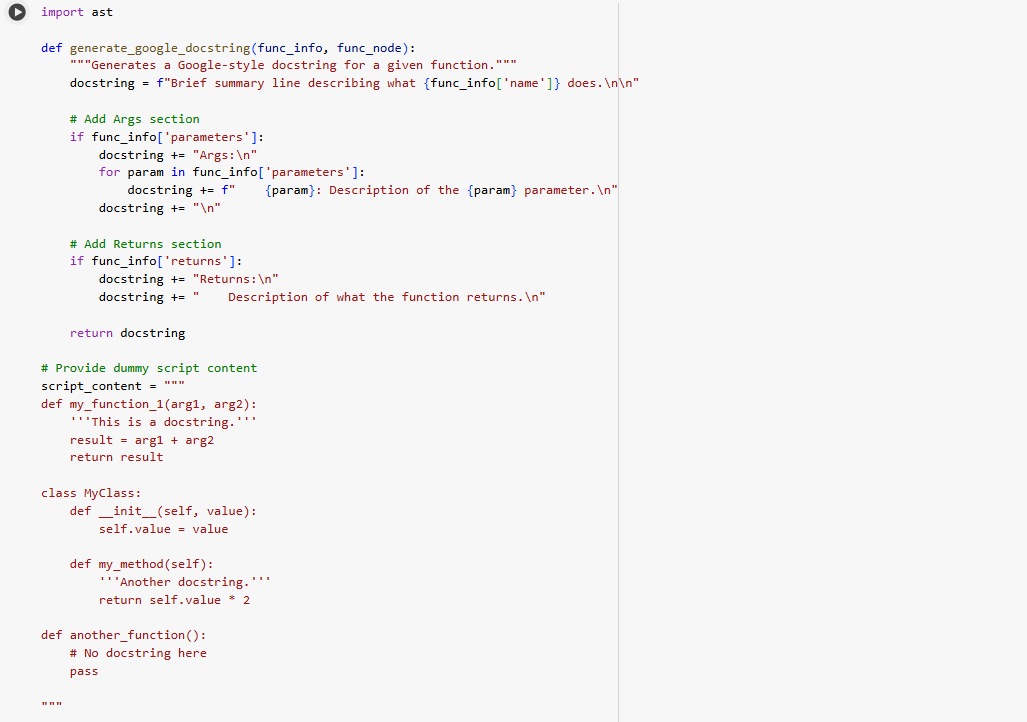
Name: Saketh

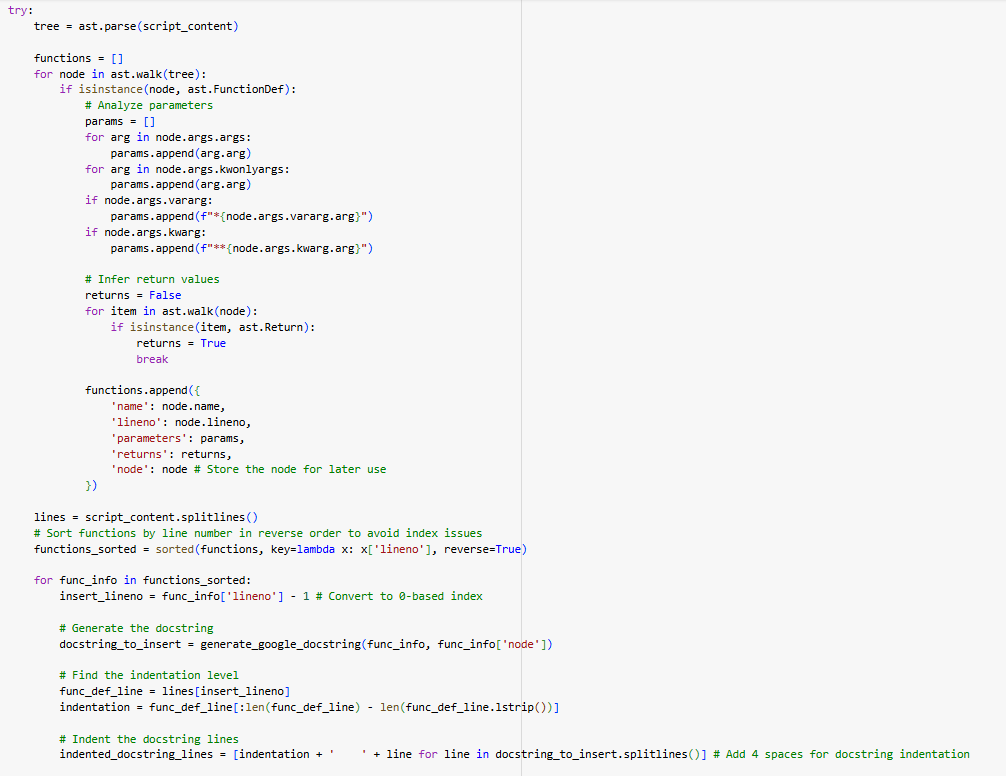
H.no: 2403a51337

Batch: 14

Task Description #1 (Documentation – Google-Style Docstrings for Python Functions)

Task: Use AI to add Google-style docstrings to all functions in a given Python script.







Observation:

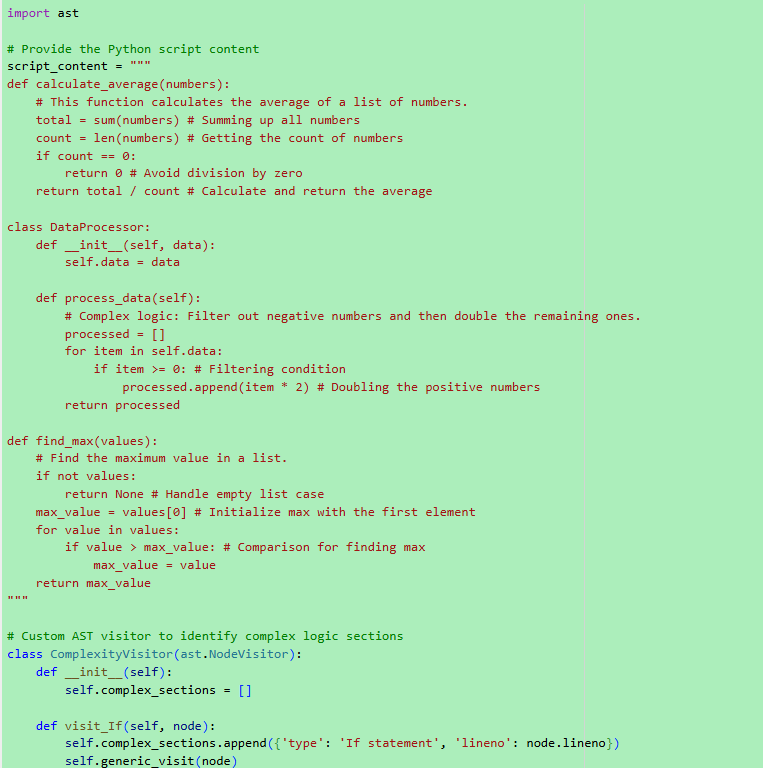
a. Adding **Google-style docstrings** to a Python script means documenting each function using a clear, structured format that improves readability and maintainability.

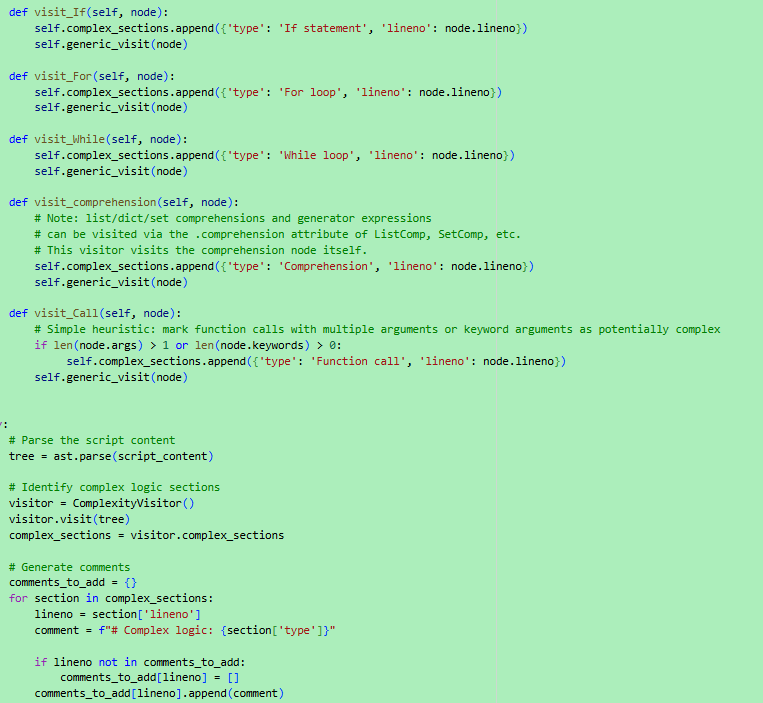
2. These docstrings include a brief summary, detailed descriptions of parameters (Args:), return values (Returns:), and optionally raised exceptions (Raises:).

3. This style is widely used in professional Python development and works well with documentation tools like Sphinx or PyDoc.

Task Description #2 (Documentation – Inline Comments for Complex  
Logic)  
• Task: Use AI to add meaningful inline comments to a Python.

Code:

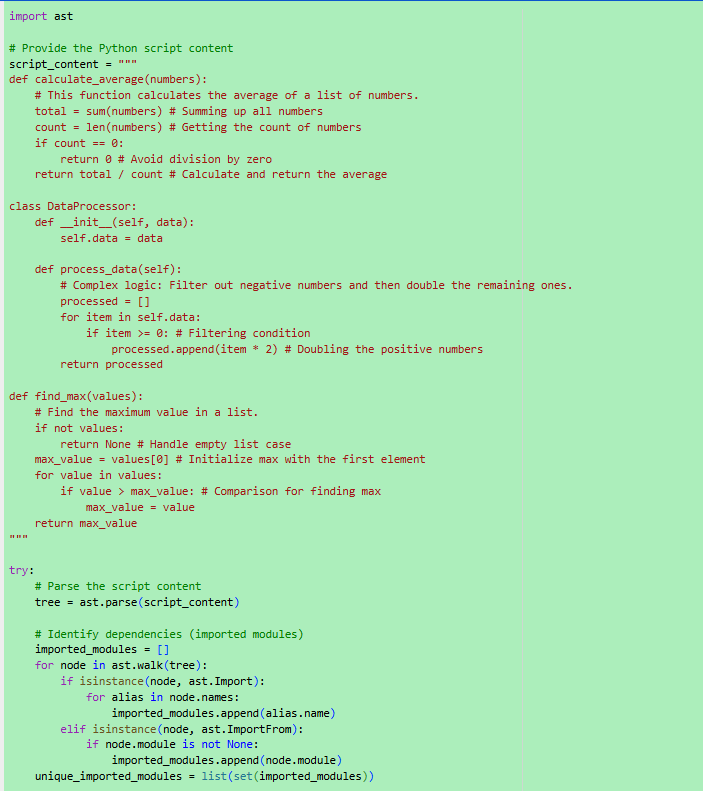


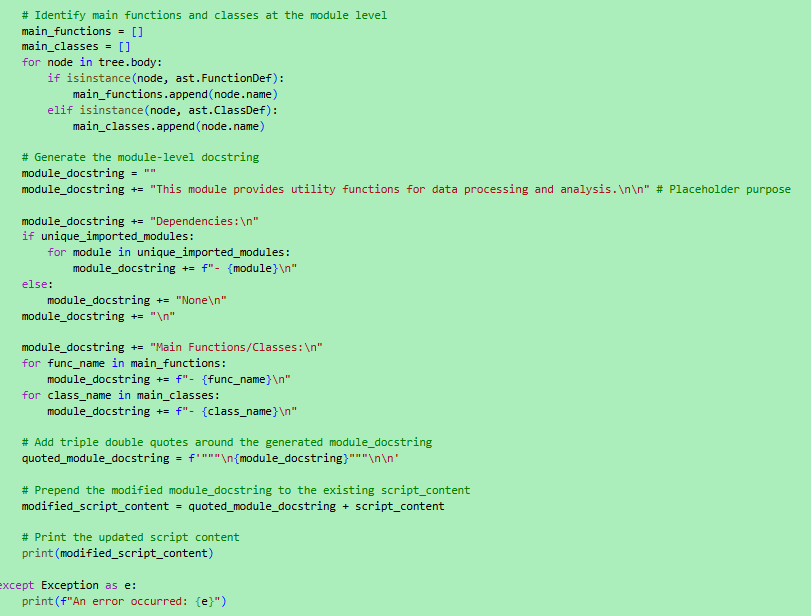


Observation:

1.A module-level docstring sits at the top of a Python file and briefly explains what the module does.  
2 .It lists key dependencies (like imported libraries) and highlights the main functions or classes defined in the file.  
3 .This helps other developers quickly understand the module’s role and how to use it.  
4 .It’s especially useful in large projects or when generating documentation automatically.

Task Description #3 (Documentation – Module-Level Documentation)  
• Task: Use AI to create a module-level docstring summarizing the purpose, dependencies, and main functions/classes of a Python  
file.





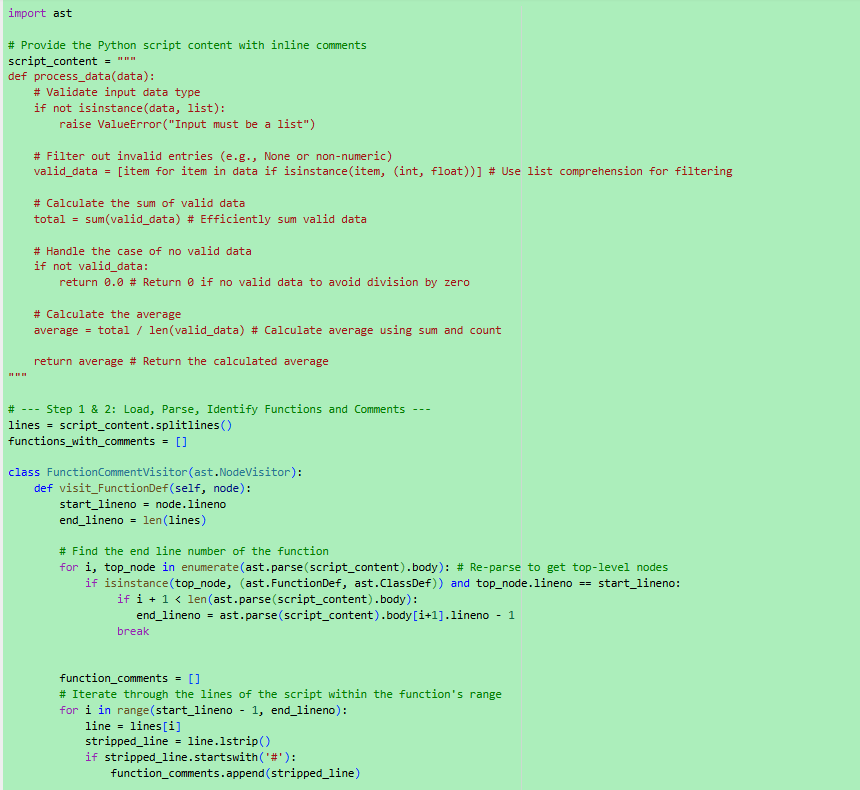
Observation:

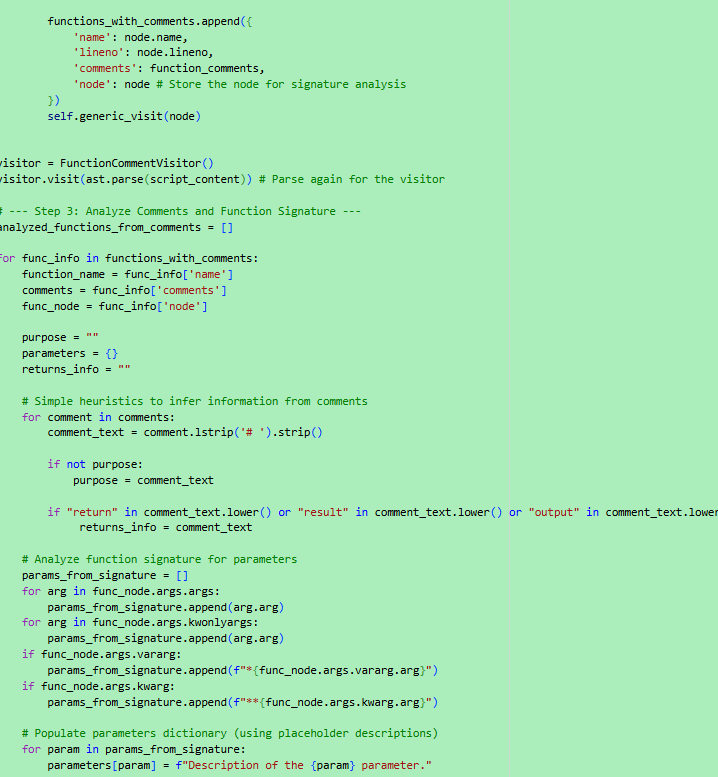
**Purpose**: What the code is designed to do.

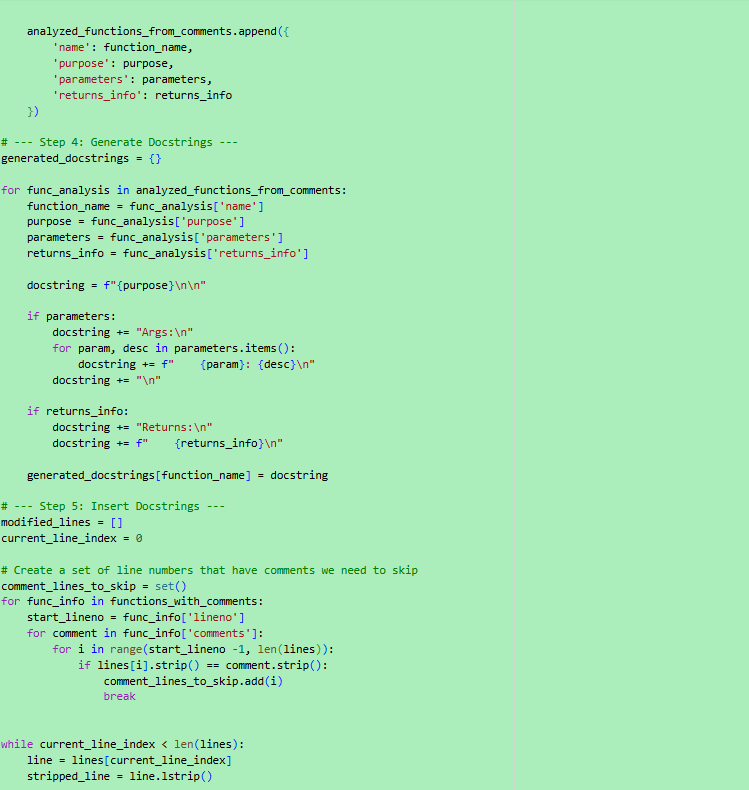
* **Logic**: How it works, step by step.
* **Functions & Classes**: What each one does and how they interact.
* **Dependencies**: What libraries or modules it uses and why.
* **Improvements**: Suggestions to make it cleaner, faster, or more readable.

Task Description #4 (Documentation – Convert Comments to  
Structured Docstrings)  
• Task: Use AI to transform existing inline comments into  
structured function docstrings following Google style.

Code:









Observation:

1.Google-style docstrings provide structured documentation for each function, detailing its purpose, parameters, return values, and exceptions.  
2.Inline comments clarify complex logic by explaining *why* certain code blocks exist, not just *what* they do.

3.Docstrings help with external documentation and code navigation, while inline comments aid direct code comprehension.  
4.Together, they make your Python code more readable, maintainable, and collaborative

Task Description #5 (Documentation – Review and Correct  
Docstrings)  
• Task: Use AI to identify and correct inaccuracies in existing docstrings.

Code:



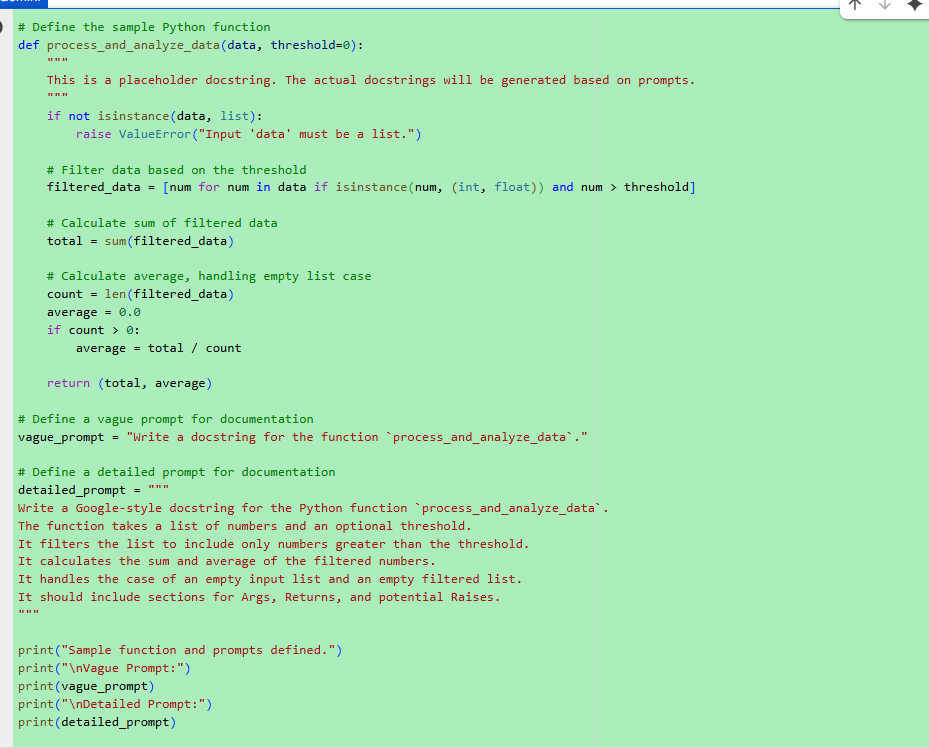
Observation:

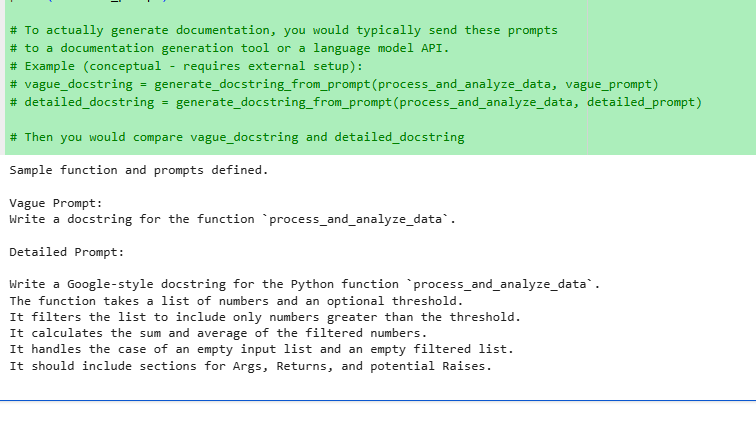
1.Identifying and correcting inaccuracies in docstrings ensures that documentation matches the actual behavior of the code.  
2. IT involves checking for missing parameters, incorrect return descriptions, or outdated summaries.  
3. Accurate docstrings improve code readability, debugging, and collaboration across teams.  
 4.This practice is essential for maintaining clean, trustworthy, and well-documented Python projects.

Task Description #6 (Documentation – Prompt Comparison  
Experiment)

Task: Compare documentation output from a vague prompt and a detailed prompt for the same Python function.

Code:





Observation:

1.When using a vague prompt, documentation tends to be generic and lacks clarity about parameters, return types, or function Behaves.  
2.A detailed prompt, on the other hand, produces precise docstrings with accurate descriptions of inputs, outputs, and edge cases.  
3.Clear prompts help capture the function’s intent and logic, making the documentation more useful for developers.  
4.This comparison highlights the importance of specificity when generating or refining code comments.  
5.Better prompts lead to better documentation, which leads to better collaboration and maintainability.