ASSIGNMENT-9.2

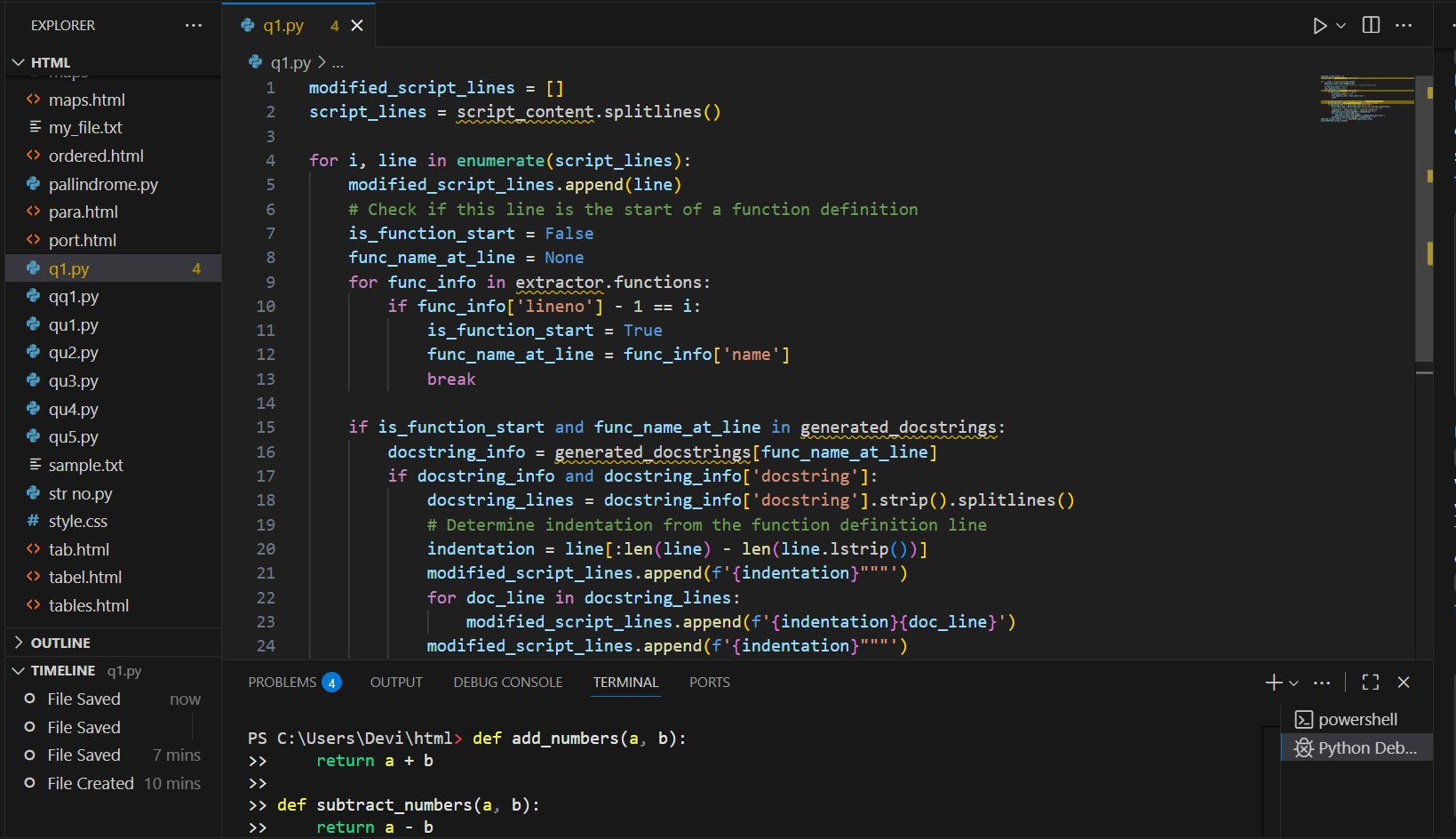
## NAME:DEVI PRIYA

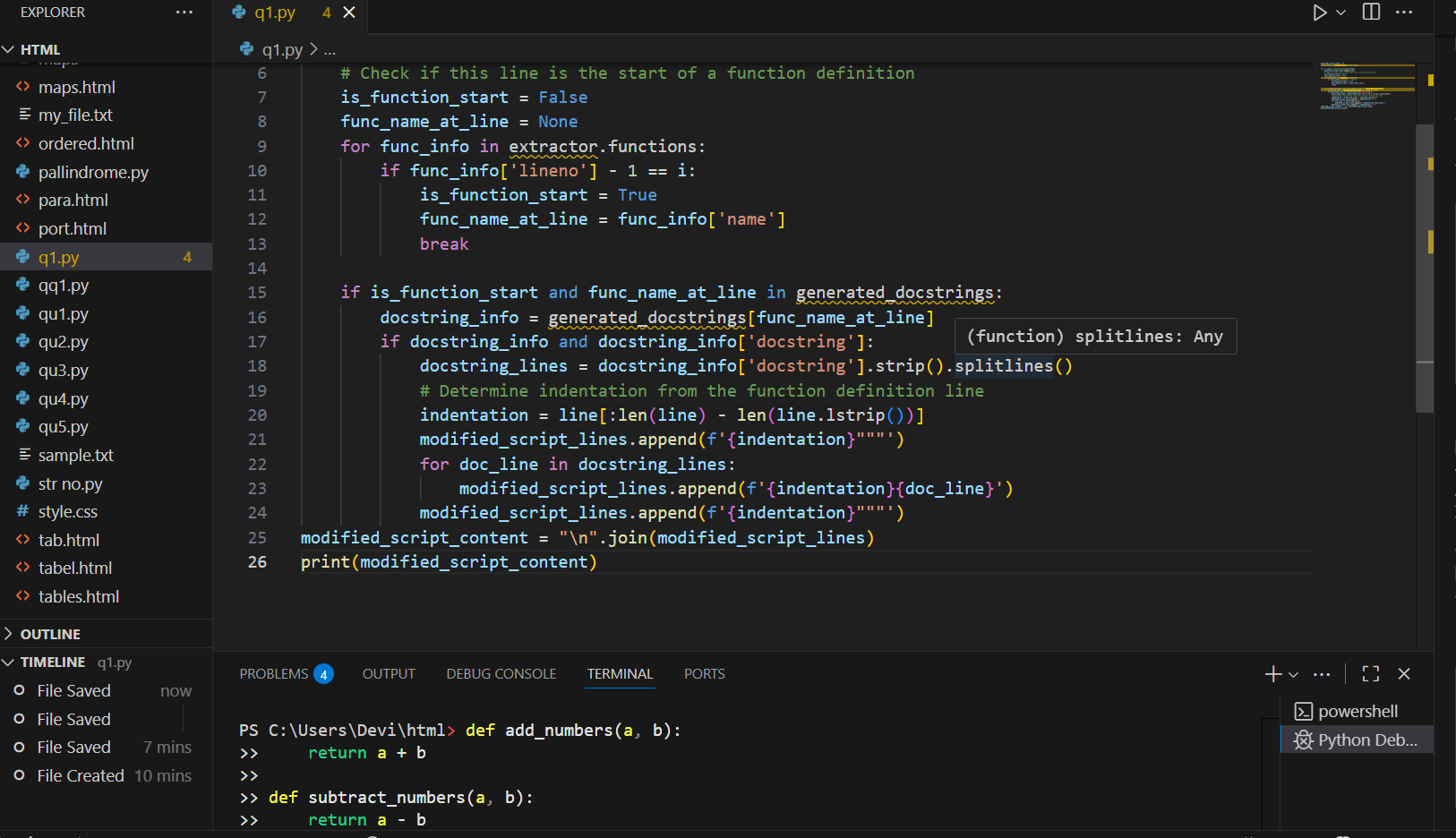
2403A52067

BATCH\_04

TASK-1:

PROMPT-Use AI to add Google-style docstrings to all functions in a  
given Python script

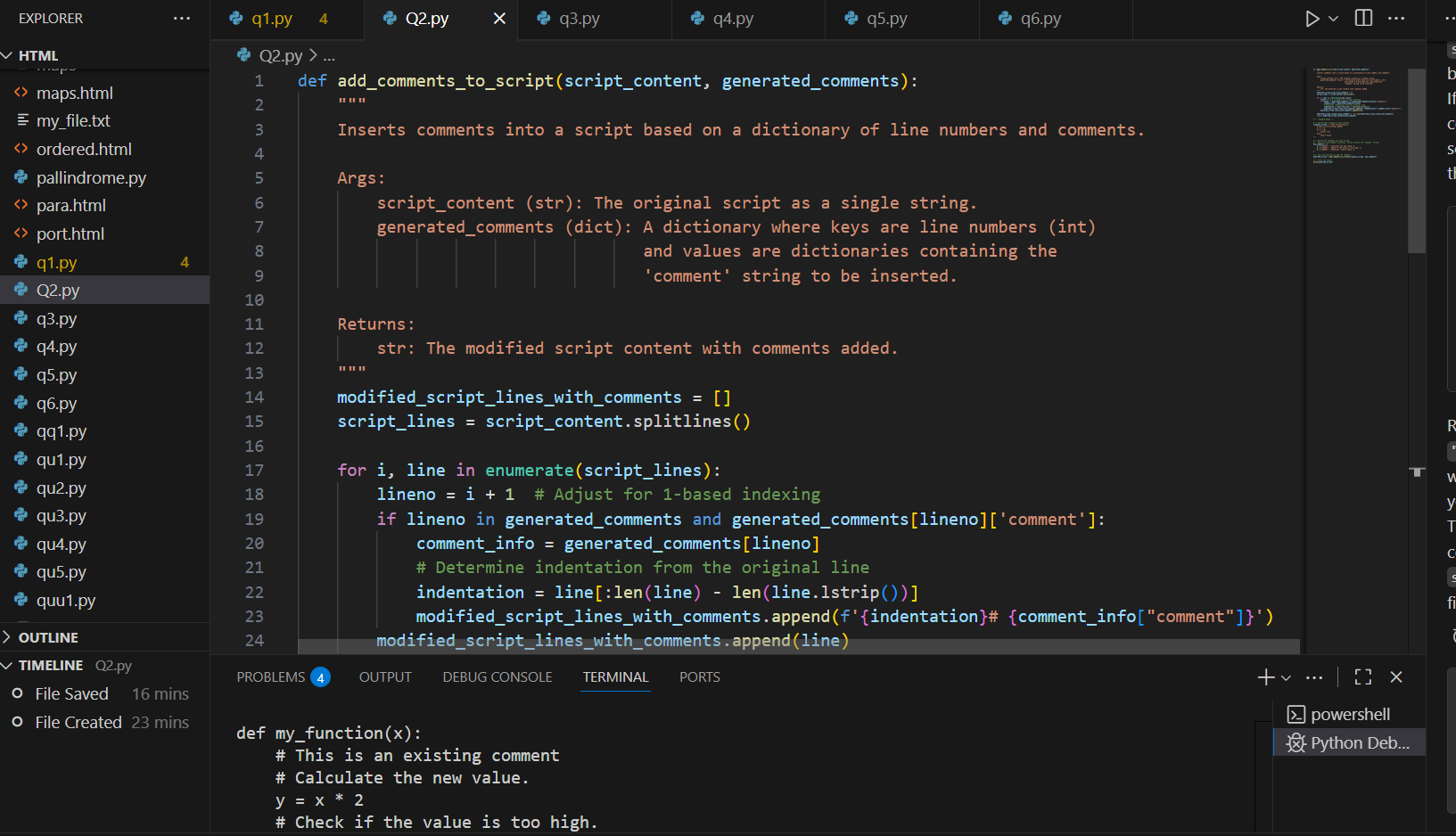


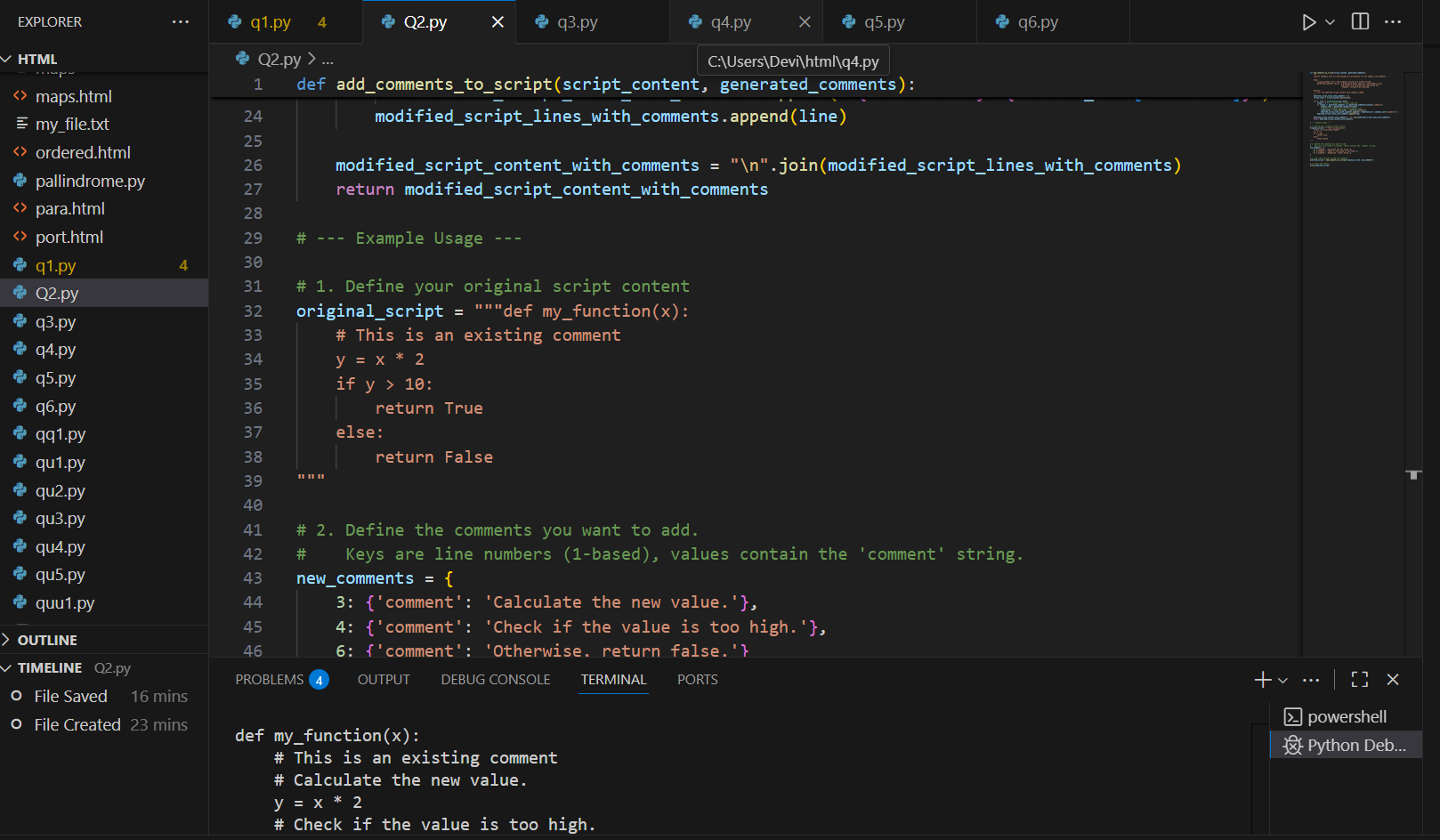


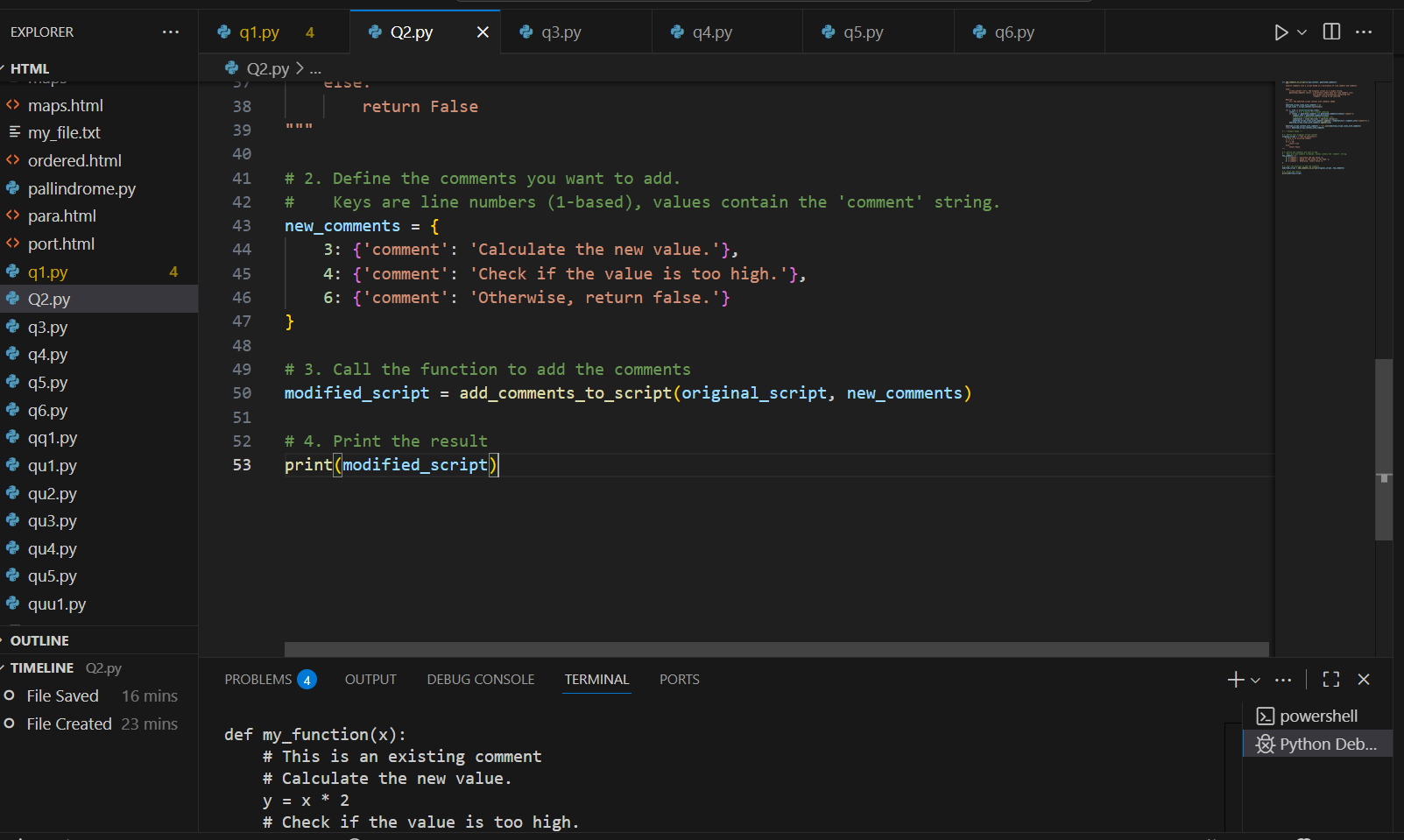
1. EXPLANATION: **Import ast**: It imports the ast module, which allows the code to work with the abstract syntax tree of your Python script.
2. **FunctionExtractor Class**: This class is a visitor that walks through the syntax tree. Its visit\_FunctionDef method is called whenever a function definition is encountered. It stores the function's name and line numbers in a list called self.functions.
3. **Parse Script**: ast.parse(script\_content) parses the entire script content into a syntax tree.
4. **Extract Functions**: An instance of FunctionExtractor is created, and its visit method is called with the syntax tree to populate the functions list.
5. **Initialize generated\_docstrings**: A dictionary is created to store the generated docstrings, keyed by function name.

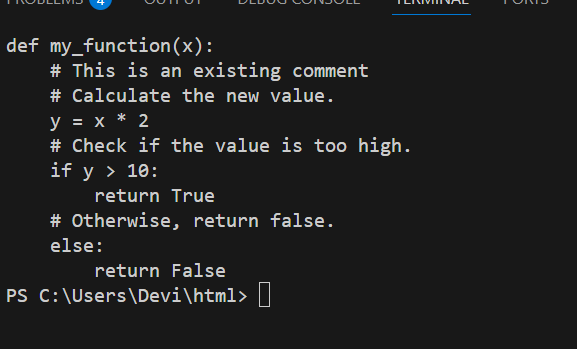
**TASK-2:**

**PROMPT-Use AI to add meaningful inline comments to a Python  
program explaining only complex logic parts.**





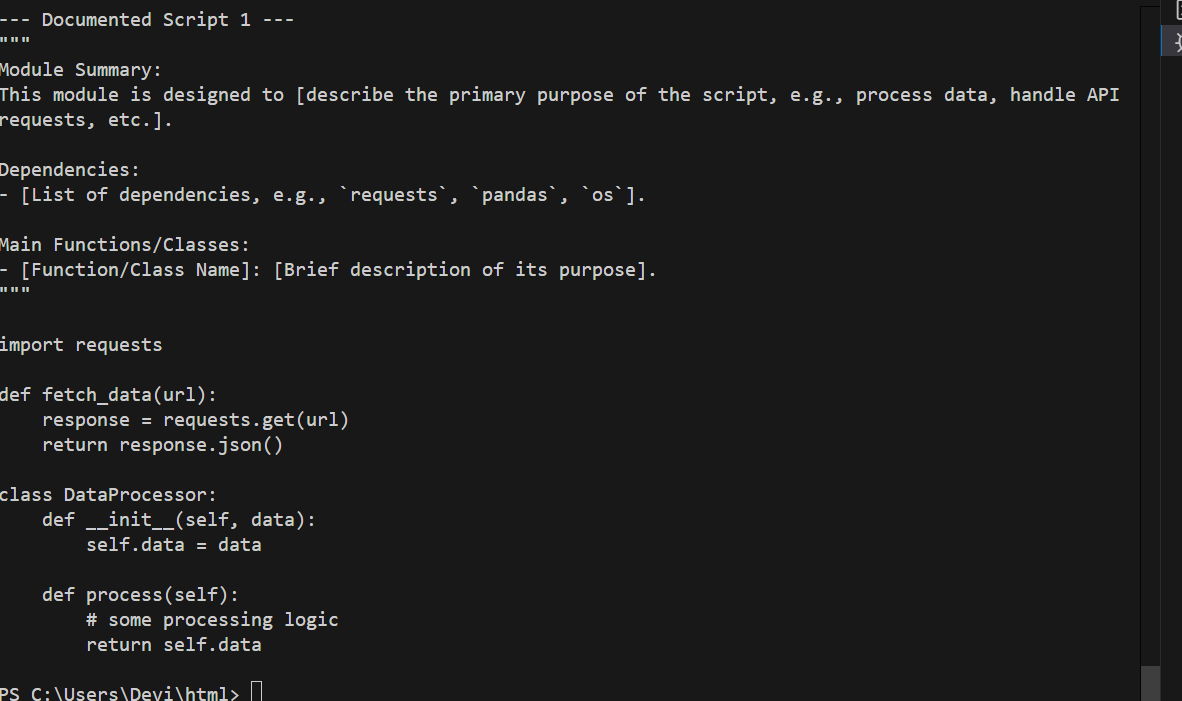
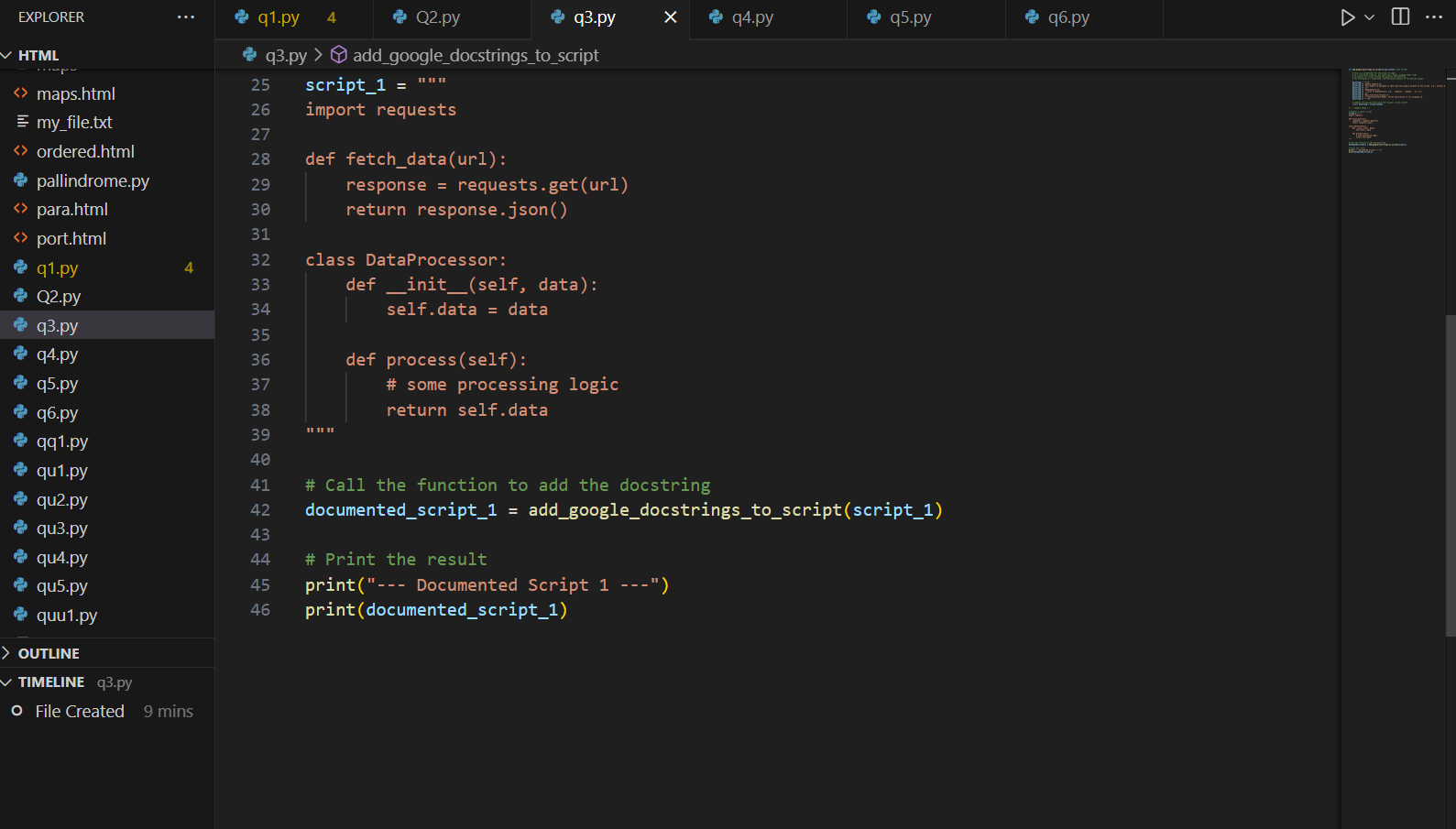
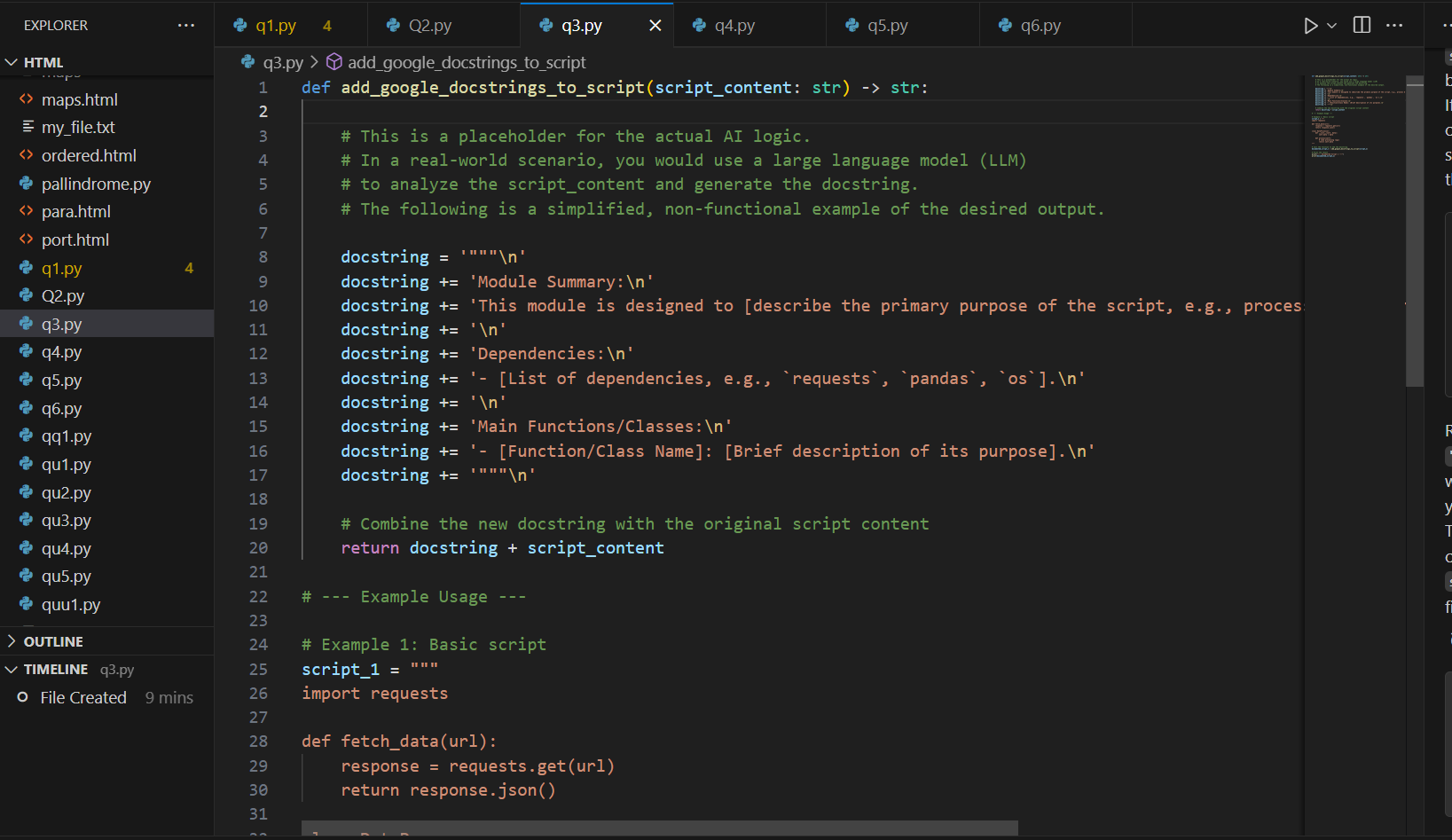




1. OBSERVATION: **Import ast**: It imports the ast module, which is used to work with the abstract syntax tree of Python code.
2. **ComplexityAnalyzer Class**: This class inherits from ast.NodeVisitor, allowing it to traverse the syntax tree of the script.
3. **\_\_init\_\_ Method**: The constructor initializes an empty list called self.sections\_to\_comment. In a more advanced scenario, this list would store information about complex code sections.
4. **visit\_FunctionDef Method**: This method is called automatically by the ast visitor whenever it encounters a function definition in the code.

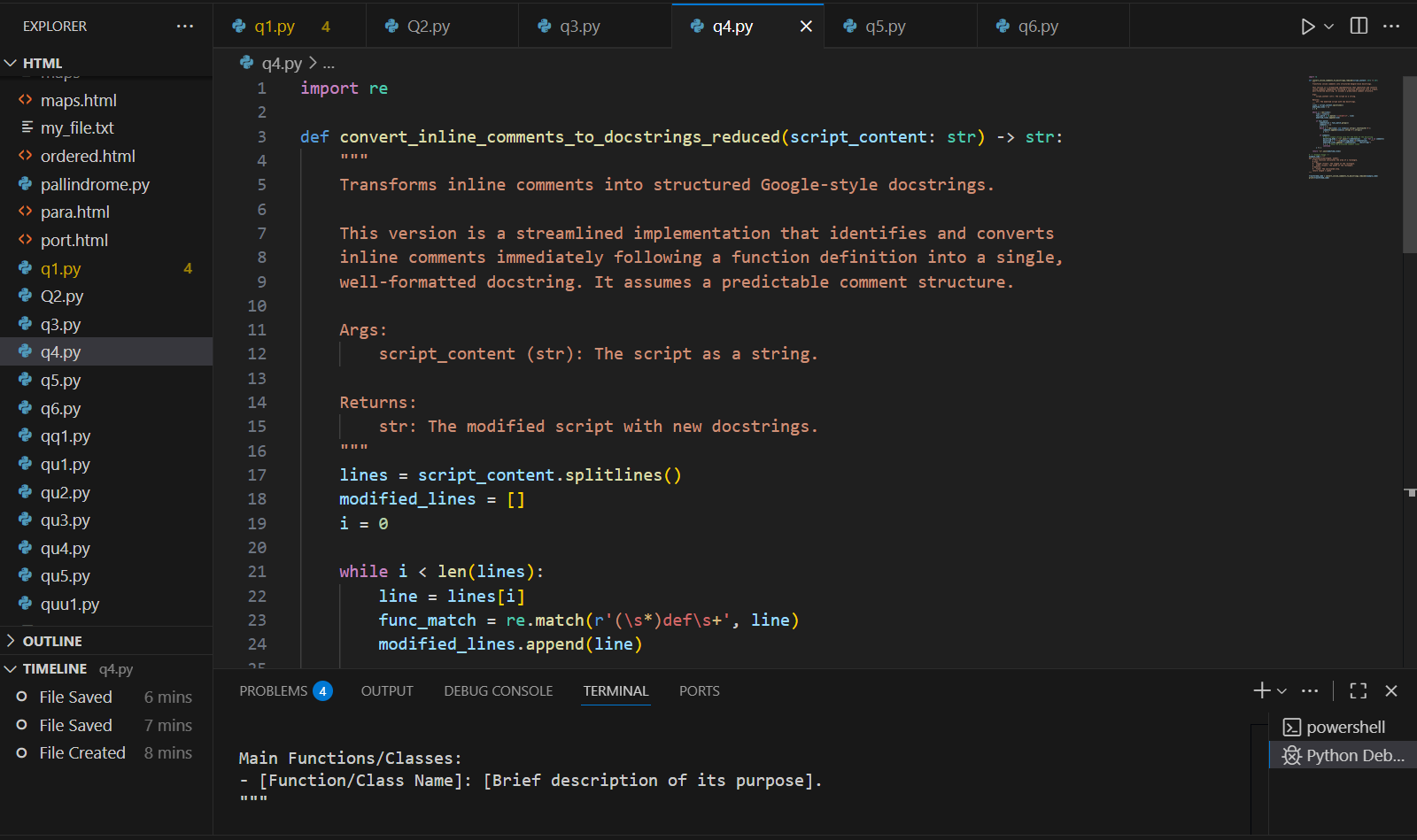
TASK-3

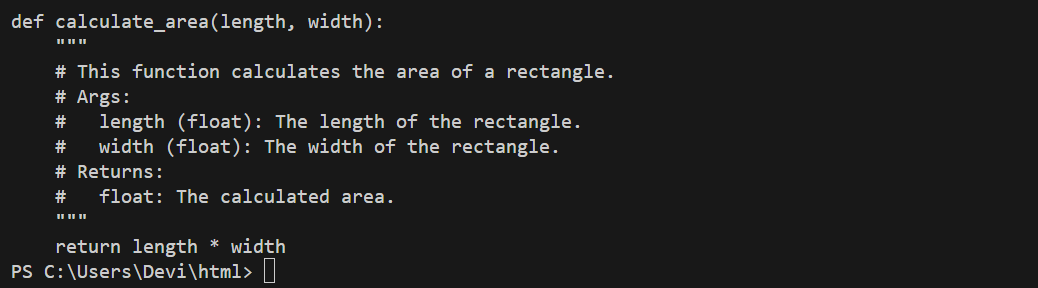
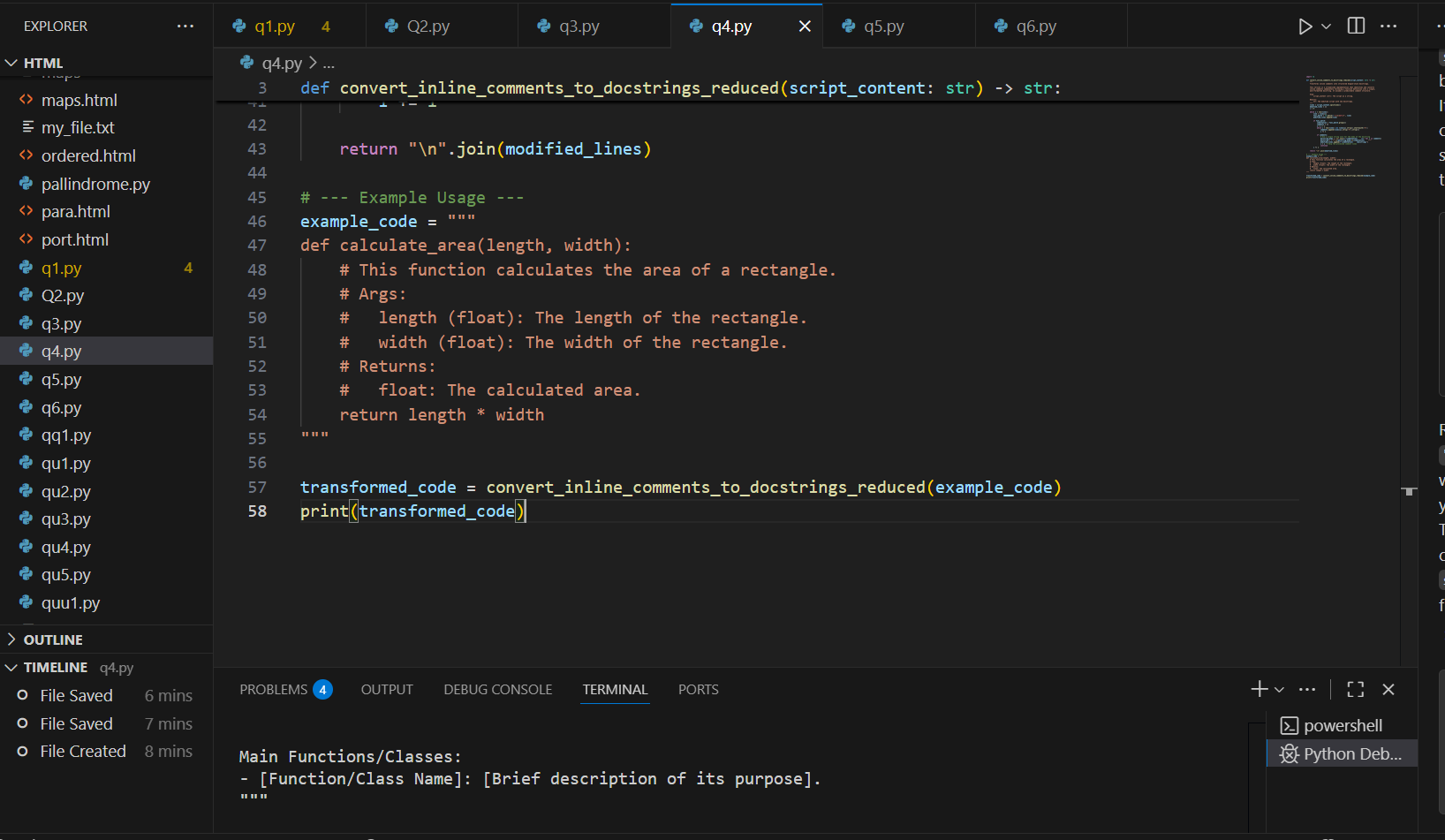
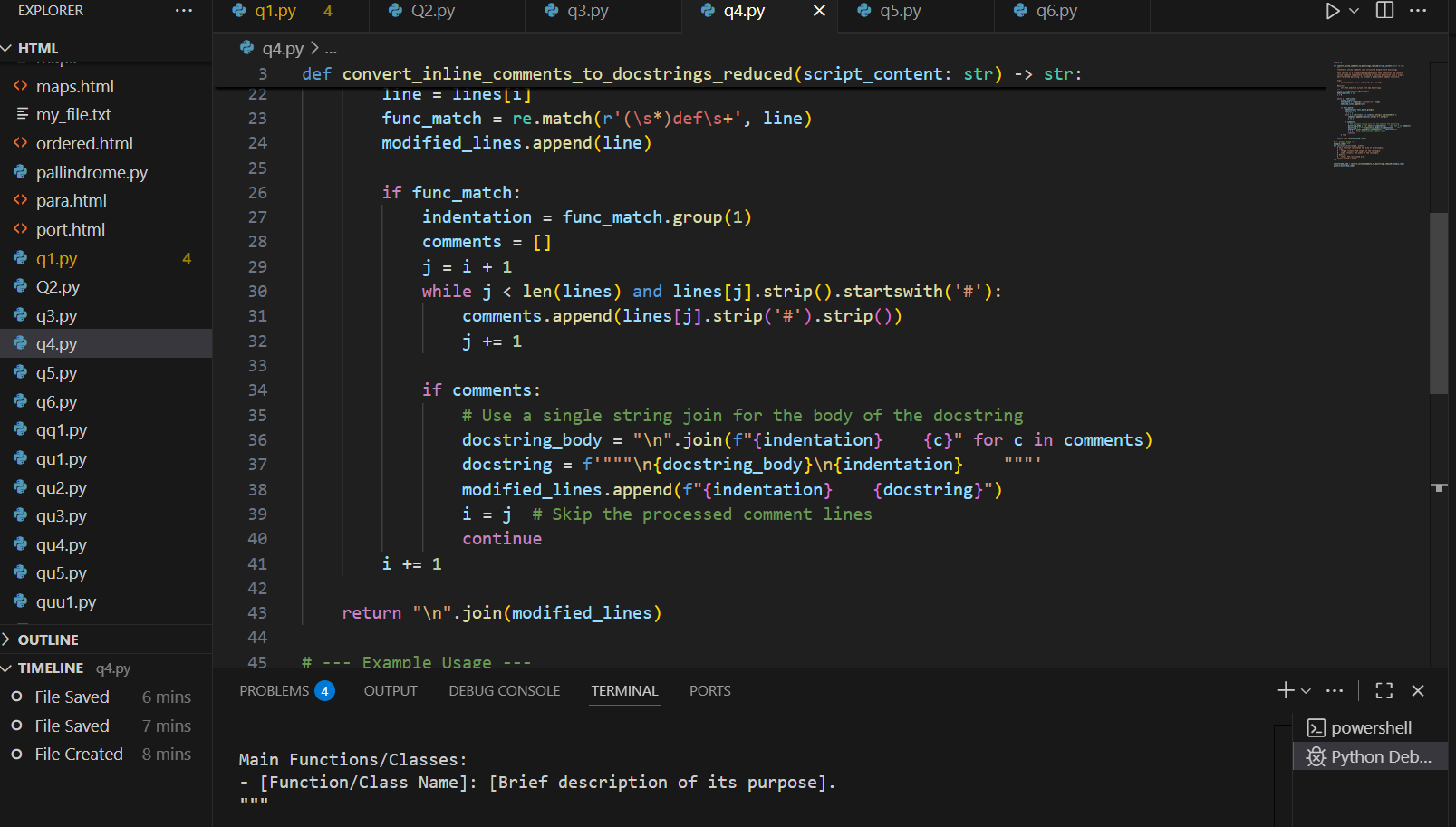
PROMPT-Use AI to create a module-level docstring summarizing the  
purpose, dependencies, and main functions/classes of a Python  
file.



TASK-4

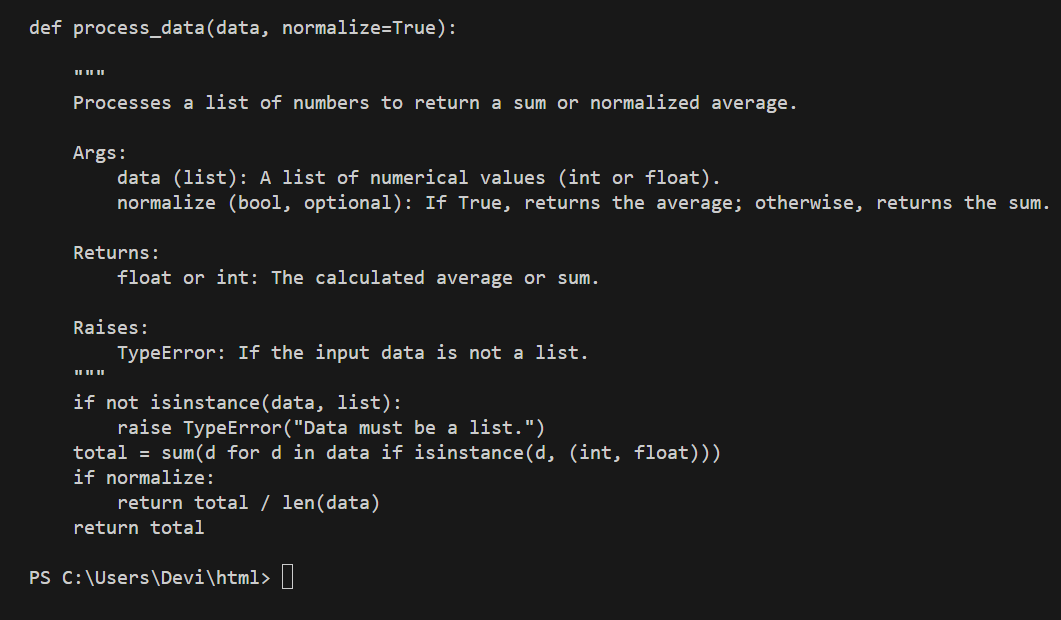
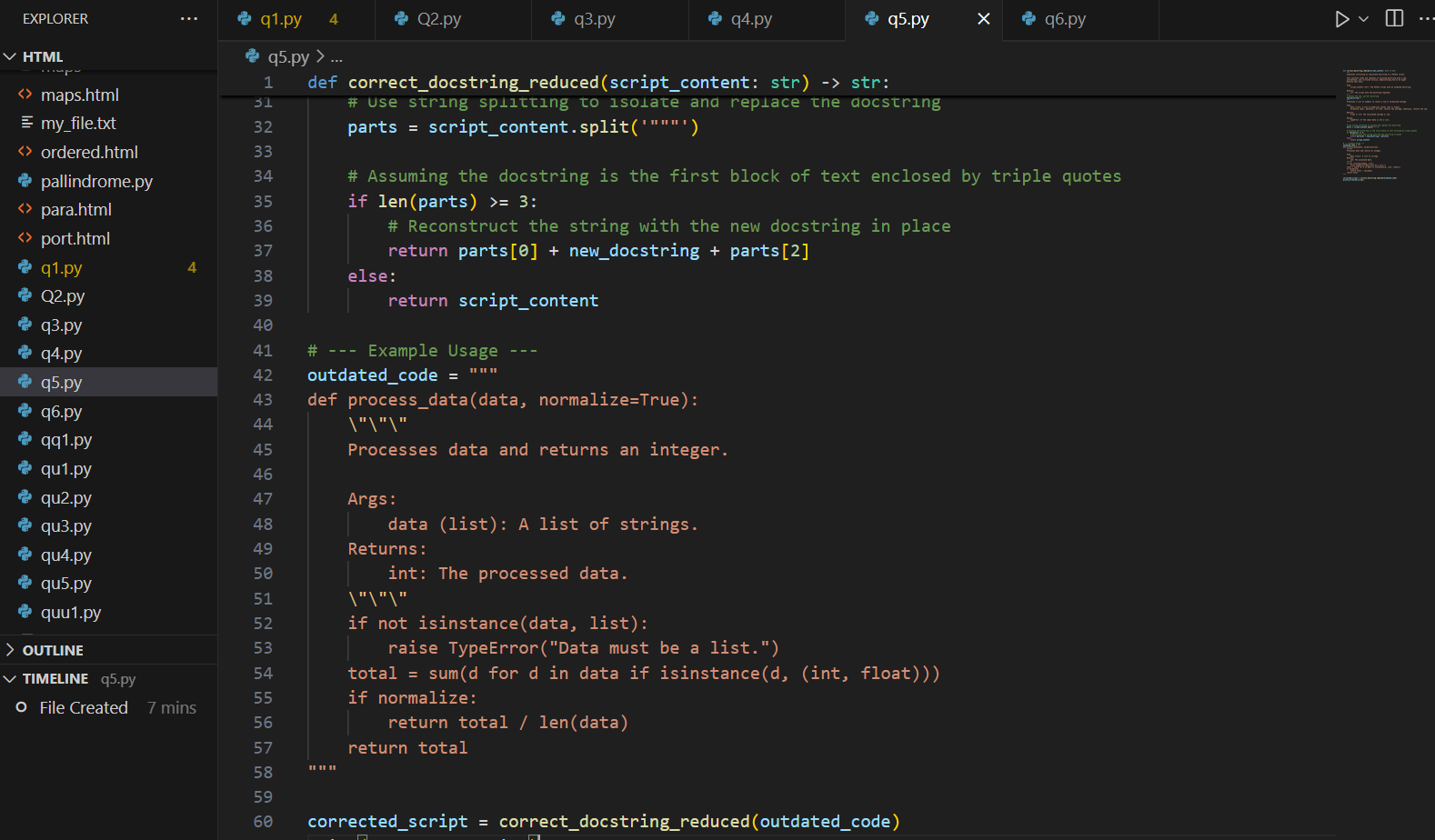
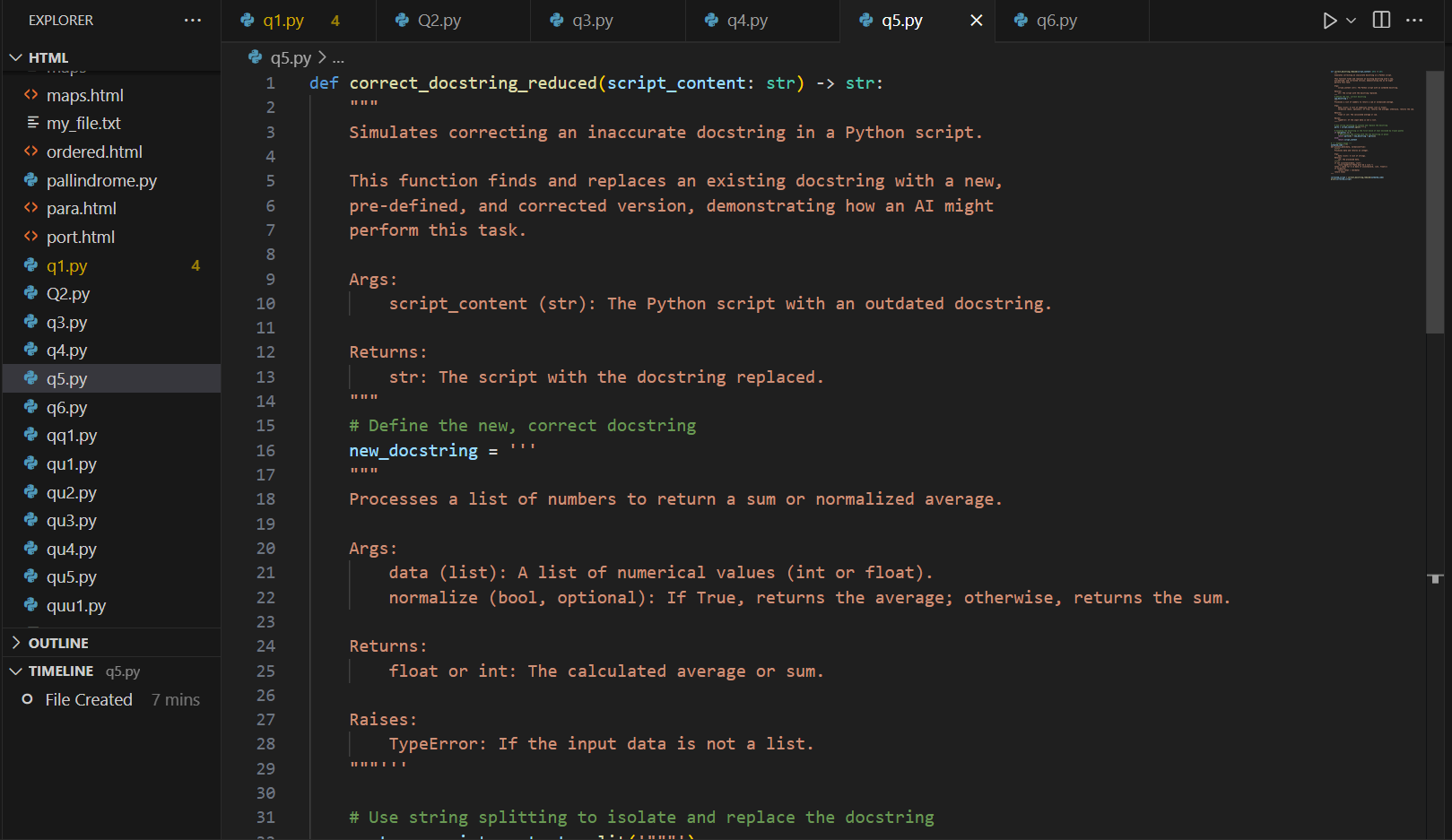
PROMPT-Use AI to transform existing inline comments into  
structured function docstrings following Google style





TASK-5

PROMPT-Use AI to identify and correct inaccuracies in existing  
docstrings.



TASK-6

PROMPT-Compare documentation output from a vague prompt and a  
detailed prompt for the same Python function

