

Assignment-9.2

Task-1:

Use AI to generate concise functional summaries for each Python function in a given script.

Instructions:

- Provide a Python script to the AI.
- Ask the AI to write a short summary describing the purpose of each function.
- Ensure summaries are brief and technically accurate.
- Do not include code implementation details.

Expected Output -1:

A Python script where each function contains a clear and concise summary explaining its purpose.

The screenshot shows a code editor with a Python script on the left and an AI-generated summary on the right. The script defines several utility functions: `calculate_factorial`, `find_maximum`, `reverse_string`, `count_vowels`, and `is_palindrome`. The AI summary on the right provides a clear and concise description of each function's purpose, such as "Computes the factorial of a given non-negative integer" for `calculate_factorial`. The summary also includes a list of the functions and their purposes, and a note that the script is ready for use and documentation.

```
1 """
2 Utility functions for data processing and mathematical operations.
3 """
4
5
6 def calculate_factorial(n):
7     """Computes the factorial of a given non-negative integer."""
8     if n < 0:
9         return None
10    if n == 0 or n == 1:
11        return 1
12    result = 1
13    for i in range(2, n + 1):
14        result *= i
15    return result
16
17
18 def find_maximum(numbers):
19     """Returns the largest value from a list of numbers."""
20     if not numbers:
21         return None
22     max_value = numbers[0]
23     for num in numbers:
24         if num > max_value:
25             max_value = num
26     return max_value
27
28
29 def reverse_string(text):
30     """Reverses a string and returns the result."""
31     return text[::-1]
32
33
34 def count_vowels(text):
35     """Counts the total number of vowels in a given string."""
36     vowels = 'aeiouAEIOU'
37     count = 0
38     for char in text:
39         if char in vowels:
40             count += 1
41     return count
42
43
44 def is_palindrome(text):
45     """Checks if a string is a palindrome, ignoring spaces and case."""
46     cleaned = text.lower().replace(" ", "")
47     return cleaned == cleaned[::-1]
48
```

CHAT

AI-GENERATED SUMMARIES FOR PYTHON FUNCTIONS

A Python script where each function contains a clear and concise summary explaining its purpose.

✓ Reviewed Task1.py

Perfect! Your script `Task1.py` now contains exactly what you requested:

✓ 10 utility functions, each with a clear and concise summary docstring:

- `calculate_factorial` - Computes the factorial of a given non-negative integer
- `find_maximum` - Returns the largest value from a list of numbers
- `reverse_string` - Reverses a string and returns the result
- `count_vowels` - Counts the total number of vowels in a given string
- `is_palindrome` - Checks if a string is a palindrome, ignoring spaces and case
- `filter_even_numbers` - Returns a new list containing only even numbers from the input
- `calculate_average` - Computes the arithmetic mean of a list of numerical values
- `merge_dictionaries` - Combines two dictionaries into a single dictionary
- `validate_email` - Validates if a string has a basic email format
- `remove_duplicates` - Removes duplicate items while preserving the original order

Each summary is:

- **Brief** - Single sentence descriptions
- **Technically accurate** - Precise function purpose
- **Implementation-free** - No code details, just the "what" not the "how"

The script is ready for use and documentation!

Describe what to build next

Task-2:

Use AI to document the logic behind conditional statements and loops in a Python program.

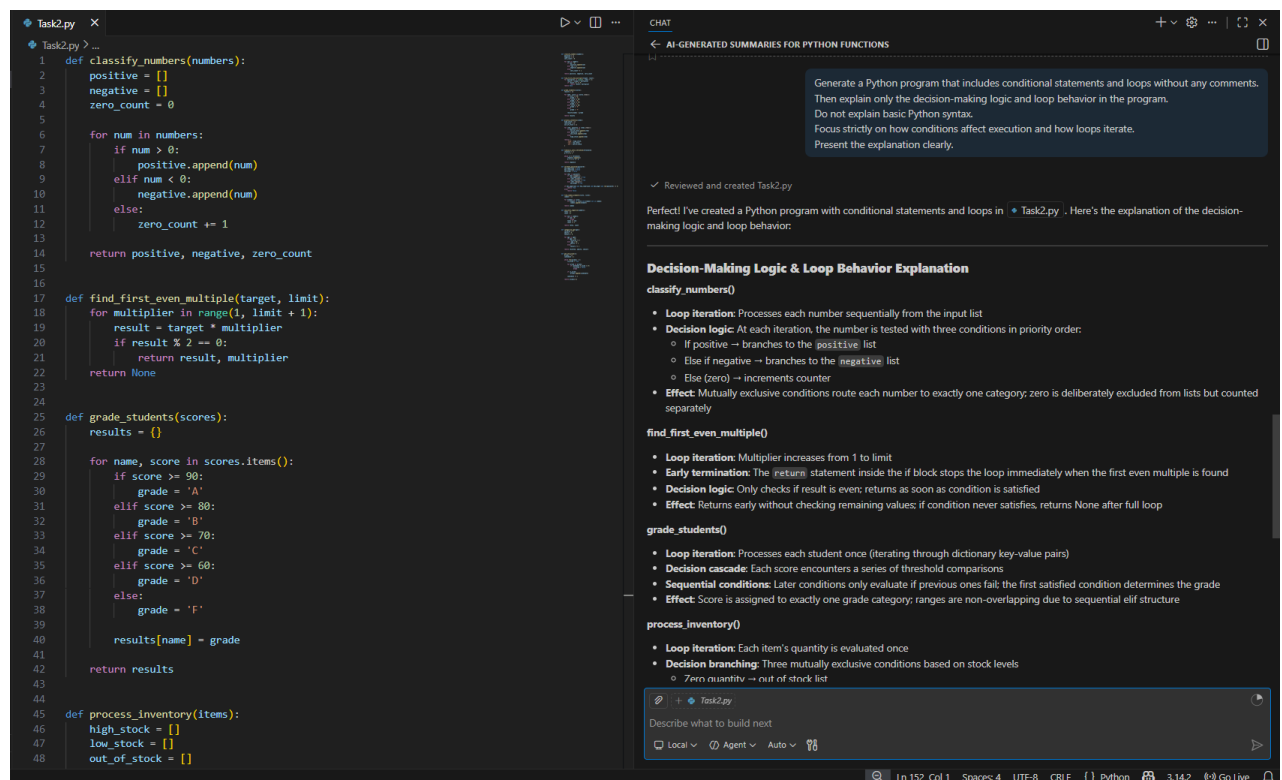
Instructions:

- Provide a Python program without comments.
- Instruct AI to explain only decision-making logic and loop behavior.

- Skip basic syntax explanations.

Expected Output -2:

Python code with clear explanations describing the logic of conditions and loops.



Task-3:

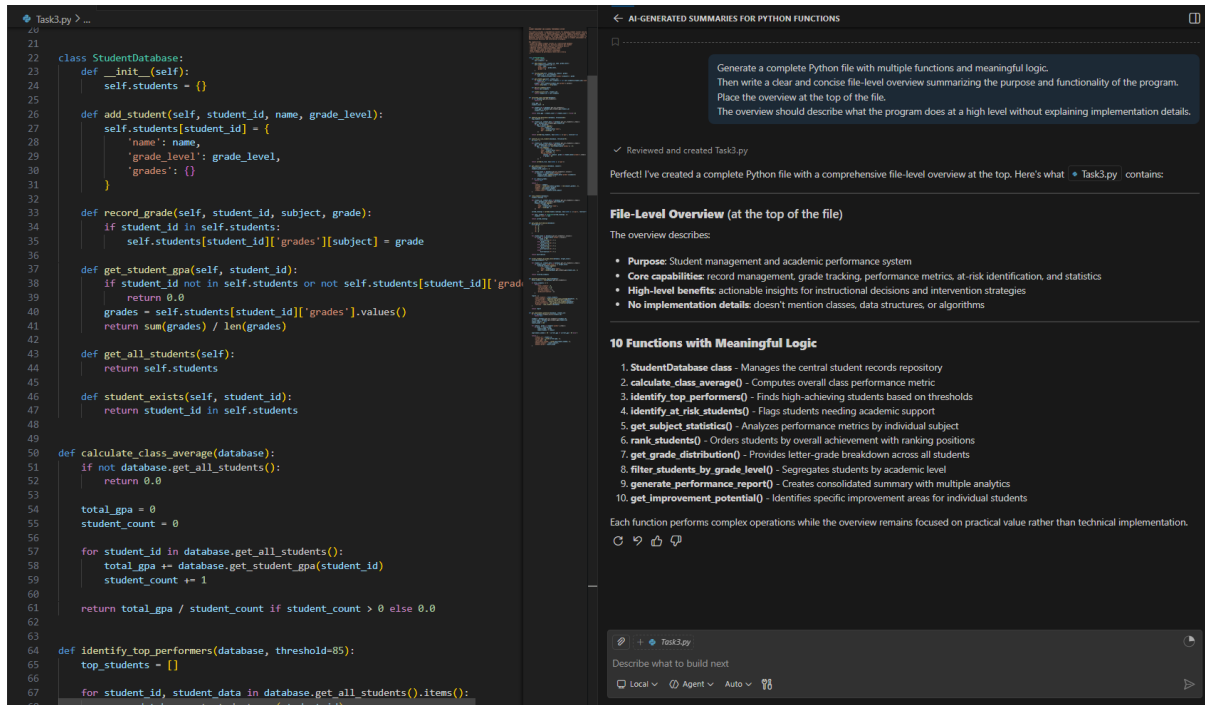
Use AI to generate a high-level overview describing the functionality of an entire Python file.

Instructions:

- Provide the complete Python file to AI.
- Ask AI to write a brief overview summarizing the file's purpose and functionality.
- Place the overview at the top of the file.

Expected Output -3:

A Python file with a clear and concise file-level overview at the beginning.



Task-4:

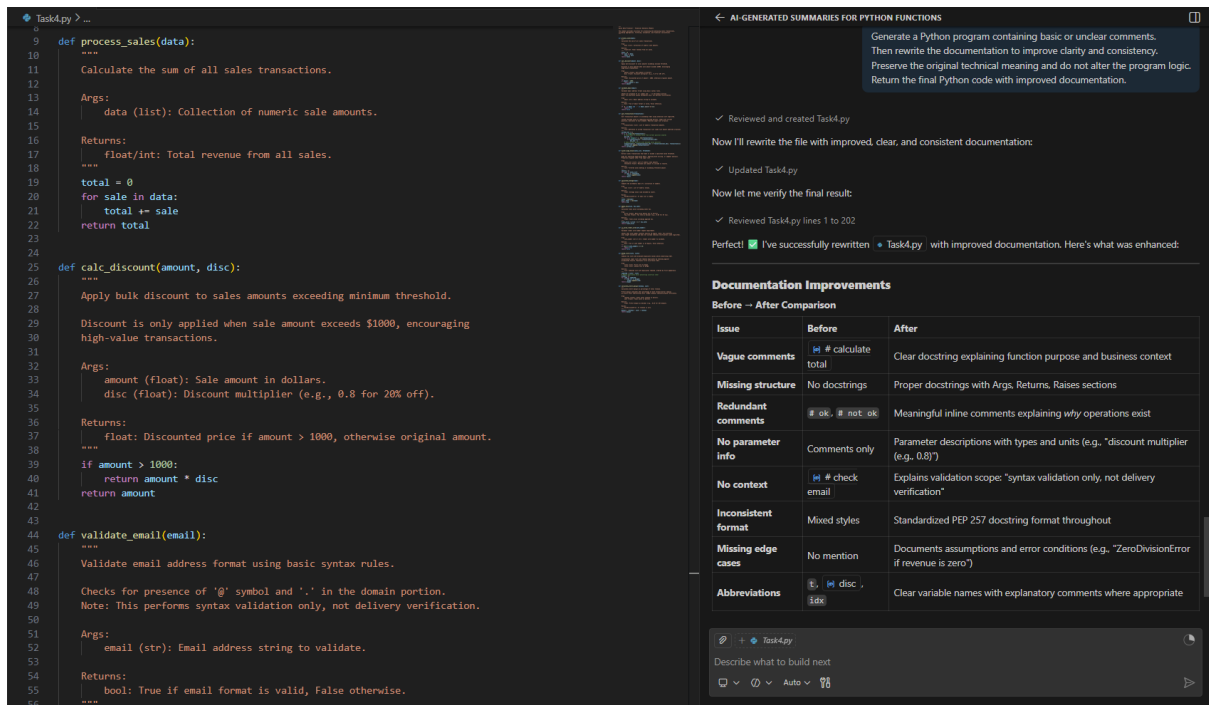
Use AI to improve clarity and consistency of existing documentation in Python code.

Instructions:

- Provide Python code containing basic or unclear comments.
- Ask AI to rewrite the documentation to improve clarity and consistency.
- Ensure technical meaning remains unchanged.

Expected Output -4:

Python code with refined and improved documentation that is clear and consistent.



Task-5:

Study the impact of prompt detail on AI-generated documentation quality.

Instructions:

Create two prompts: one brief and one detailed.

- Use both prompts to document the same Python function.
- Compare the generated outputs.

Expected Output -5:

A comparison table highlighting differences in completeness, clarity, and accuracy of documentation.

TaskSpy > ...

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48

'''
ANALYSIS: Impact of Prompt Detail on AI-Generated Documentation Quality

ORIGINAL FUNCTION:
'''
def calculate_discount(price, discount_percent):
 if price < 0 or discount_percent < 0:
 return None
 discount = price * (discount_percent / 100)
 return price - discount
'''

SECTION 1: DOCUMENTATION GENERATED FROM BRIEF PROMPT

BRIEF PROMPT USED:
"Write a docstring for this function."
GENERATED DOCUMENTATION (BRIEF):
'''
def calculate_discount_brief(price, discount_percent):
 """Calculates the final price after applying a discount percentage."""
 if price < 0 or discount_percent < 0:
 return None
 discount = price * (discount_percent / 100)
 return price - discount
'''

SECTION 2: DOCUMENTATION GENERATED FROM DETAILED PROMPT

DETAILED PROMPT USED:
"Write a comprehensive PEP 257 compliant docstring for the calculate_discount function. Include: (1) A one-line summary explaining the function's purpose, (2) An extended description detailing the calculation logic and use cases, (3) Args section documenting parameter names, types, units, and valid ranges, (4) Returns section specifying return type and value interpretation, (5) Raises section noting error conditions and return values for edge cases, (6) Examples section demonstrating typical usage with concrete inputs/outputs,

AI-GENERATED SUMMARIES FOR PYTHON FUNCTIONS

1. Create one brief prompt and one detailed prompt for documenting this function.
2. Generate documentation using both prompts.
3. Present both outputs clearly labeled.
4. Provide a comparison table evaluating completeness, clarity, accuracy, and formatting.
Do not modify the function logic.

✓ Reviewed and created Task5.py

Perfect! I've completed a comprehensive analysis in [Task5.py](#). Here's the summary:

Analysis: Impact of Prompt Detail on AI Documentation Quality

Task Components Completed:

1. Two Prompts Created

BRIEF PROMPT:
"Write a docstring for this function."
DETAILED PROMPT:
"Write a comprehensive PEP 257 compliant docstring including: (1) One-line summary, (2) Extended description, (3) Args section with types and ranges, (4) Returns section, (5) Raises section, (6) Examples with concrete inputs/outputs, (7) Notes about assumptions and edge cases."
2. Documentation Outputs Generated

BRIEF DOCUMENTATION (12 words):
""Calculates the final price after applying a discount percentage.""
DETAILED DOCUMENTATION (450+ words):
• Complete PEP 257 compliant docstring

+ Task5.py

Describe what to build next

Auto