NAME : B.SRISHANTH ROLL NO : 2403A510G3

ASSIGNMENT : 9.1 SUBJECT : AI ASSISTED CODING

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.

• Instructions:

o Prompt AI to generate docstrings without providing any input-output examples.

o Ensure each docstring includes:

▪ Function description

▪ Parameters with type hints

▪ Return values with type hints

▪ Example usage

o Review the generated docstrings for accuracy and formatting.

• Expected Output #1:

o A Python script with all functions documented using correctly formatted Google-style docstrings

PROMPT :

WRITE A PYTHON SCRIPT TO ADD Google-style docstrings to all functions

IT SHOULD CONTAIN :

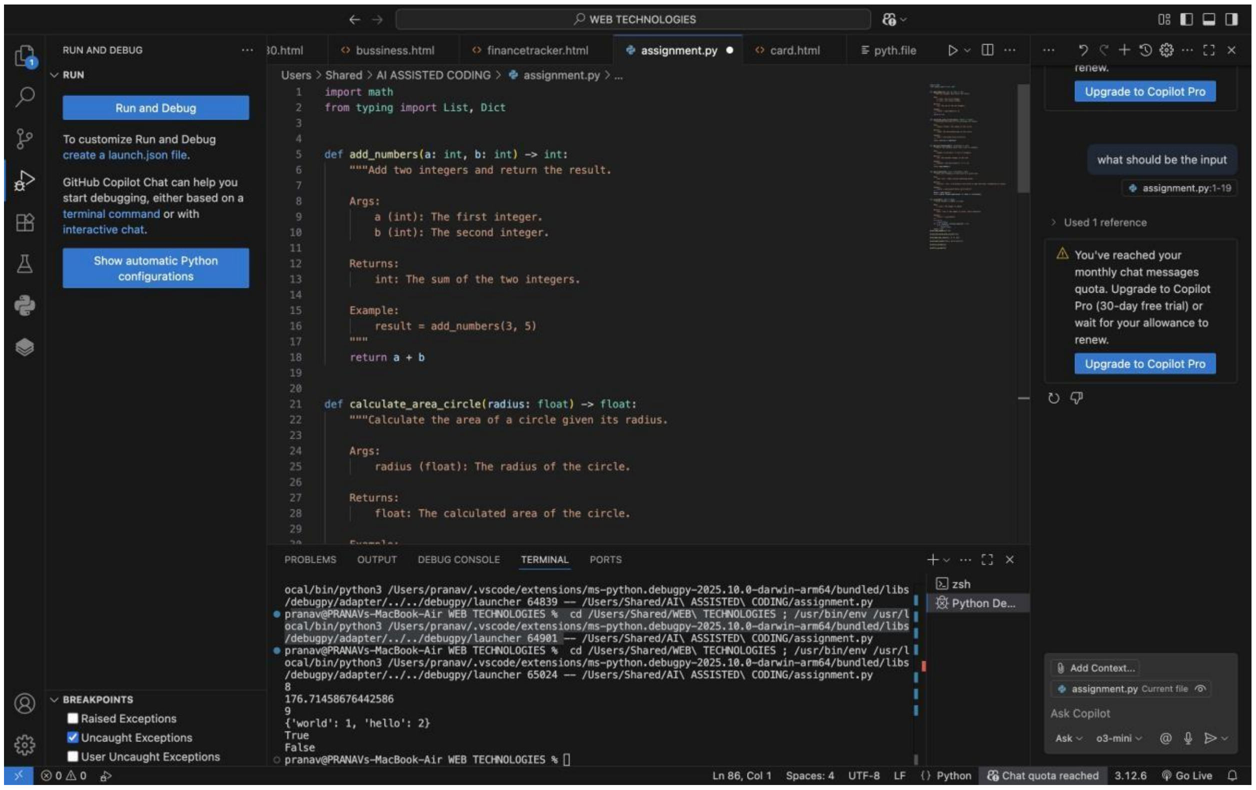
Function description

▪ Parameters with type hints

▪ Return values with type hints

▪ Example usage

CODE GENERATED :



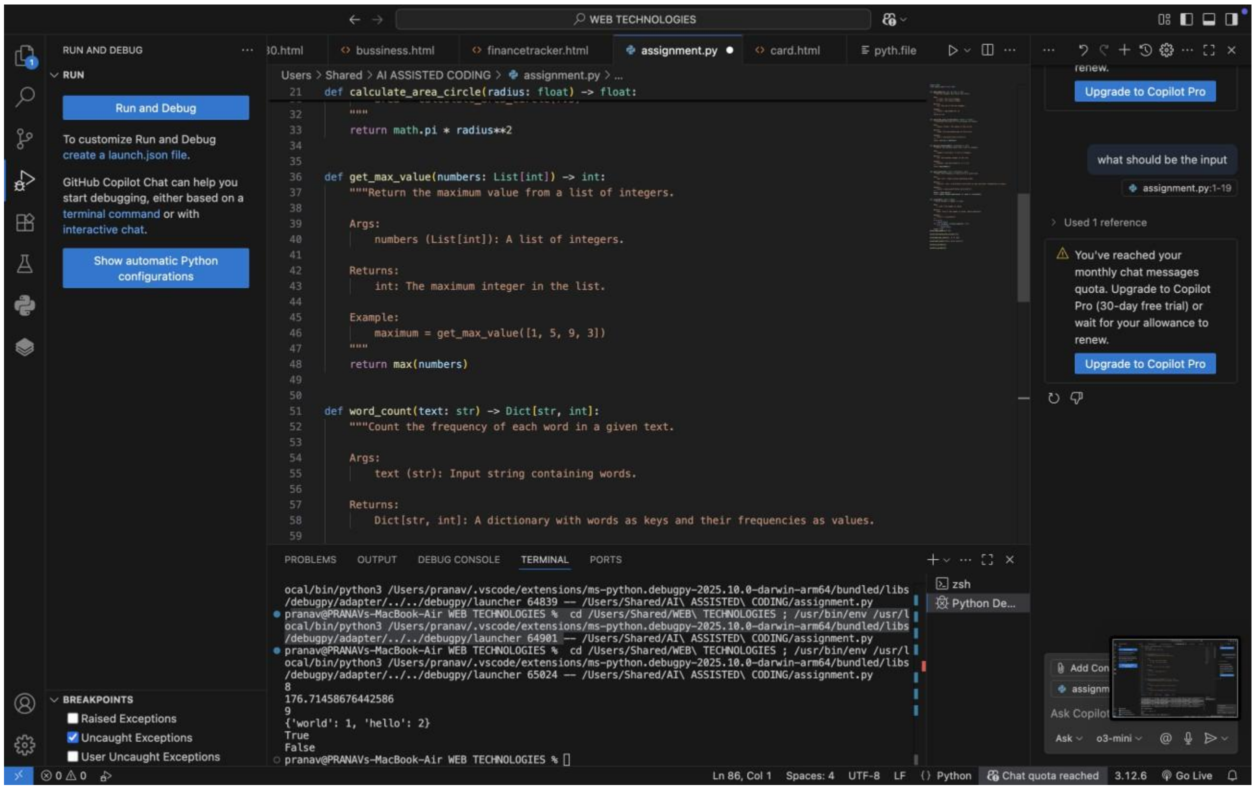
```python
import math
from typing import List, Dict


def add_numbers(a: int, b: int) -> int:
    """Add two integers and return the result.

    Args:
        a (int): The first integer.
        b (int): The second integer.

    Returns:
        int: The sum of the two integers.

    Example:
        result = add_numbers(3, 5)
    """
    return a + b


def calculate_area_circle(radius: float) -> float:
    """Calculate the area of a circle given its radius.

    Args:
        radius (float): The radius of the circle.

    Returns:
        float: The calculated area of the circle.

```

```
ocal/bin/python3 /Users/pranav/.vscode/extensions/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs
/debugpy/adapter/../../debugpy/launcher 64839 -- /Users/Shared/AI\ ASSISTED\ CODING/assignment.py
pranav@PRANAVs-MacBook-Air WEB TECHNOLOGIES %  cd /Users/Shared/WEB\ TECHNOLOGIES ; /usr/bin/env /usr/l
ocal/bin/python3 /Users/pranav/.vscode/extensions/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs
/debugpy/adapter/../../debugpy/launcher 64901 -- /Users/Shared/AI\ ASSISTED\ CODING/assignment.py
pranav@PRANAVs-MacBook-Air WEB TECHNOLOGIES %  cd /Users/Shared/WEB\ TECHNOLOGIES ; /usr/bin/env /usr/l
ocal/bin/python3 /Users/pranav/.vscode/extensions/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs
/debugpy/adapter/../../debugpy/launcher 65024 -- /Users/Shared/AI\ ASSISTED\ CODING/assignment.py
8
176.71458676442586
9
{'world': 1, 'hello': 2}
True
False
pranav@PRANAVs-MacBook-Air WEB TECHNOLOGIES % 
```

what should be the input

You've reached your monthly chat messages quota. Upgrade to Copilot Pro (30-day free trial) or wait for your allowance to renew.

OUTPUT :



```python
def calculate_area_circle(radius: float) -> float:
    """
    return math.pi * radius**2


def get_max_value(numbers: List[int]) -> int:
    """Return the maximum value from a list of integers.

    Args:
        numbers (List[int]): A list of integers.

    Returns:
        int: The maximum integer in the list.

    Example:
        maximum = get_max_value([1, 5, 9, 3])
    """
    return max(numbers)


def word_count(text: str) -> Dict[str, int]:
    """Count the frequency of each word in a given text.

    Args:
        text (str): Input string containing words.

    Returns:
        Dict[str, int]: A dictionary with words as keys and their frequencies as values.

```

```
ocal/bin/python3 /Users/pranav/.vscode/extensions/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs
/debugpy/adapter/../../debugpy/launcher 64839 -- /Users/Shared/AI\ ASSISTED\ CODING/assignment.py
pranav@PRANAVs-MacBook-Air WEB TECHNOLOGIES %  cd /Users/Shared/WEB\ TECHNOLOGIES ; /usr/bin/env /usr/l
ocal/bin/python3 /Users/pranav/.vscode/extensions/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs
/debugpy/adapter/../../debugpy/launcher 64901 -- /Users/Shared/AI\ ASSISTED\ CODING/assignment.py
pranav@PRANAVs-MacBook-Air WEB TECHNOLOGIES %  cd /Users/Shared/WEB\ TECHNOLOGIES ; /usr/bin/env /usr/l
ocal/bin/python3 /Users/pranav/.vscode/extensions/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs
/debugpy/adapter/../../debugpy/launcher 65024 -- /Users/Shared/AI\ ASSISTED\ CODING/assignment.py
8
176.71458676442586
9
{'world': 1, 'hello': 2}
True
False
pranav@PRANAVs-MacBook-Air WEB TECHNOLOGIES % 
```

what should be the input

You've reached your monthly chat messages quota. Upgrade to Copilot Pro (30-day free trial) or wait for your allowance to renew.

Observation for the Code

1. The script defines **five independent functions**, each performing a specific task:
   a. add_numbers(a, b) → Adds two integers.
   b. calculate_area_circle(radius) → Computes the area of a circle using the formula πr².
   c. get_max_value(numbers) → Finds the maximum element from a list of integers.
   d. word_count(text) → Returns a dictionary of word frequencies from a given string.
   e. is_prime(n) → Checks if a number is prime.
2. Each function is documented with **Google-style docstrings**, which include:
   a. Function description.
   b. Parameters with type hints.
   c. Return type with description.
   d. Example usage for clarity.
3. The code uses **type hints** (int, float, List[int], Dict[str, int], bool) to improve readability and make it easier to catch type-related errors.
4. The script imports **standard libraries** only (math for π and square root, and typing for type hints).
5. The script is written in a **modular** way:
   a. Functions can be reused independently in other programs.
   b. Running the script directly will not produce output (since no main block or print statements are included).
6. The code is **clean, well-documented, and beginner-friendly**, making it easy to extend or integrate into larger projects.

Task Description #2 (Documentation – Inline Comments for Complex Logic)

• Task: Use AI to add meaningful inline comments to a Python program explaining only complex logic parts.

• Instructions:

o Provide a Python script without comments to the AI.

o Instruct AI to skip obvious syntax explanations and focus only on tricky or non-intuitive code sections.

o Verify that comments improve code readability and maintainability.

• Expected Output #2:

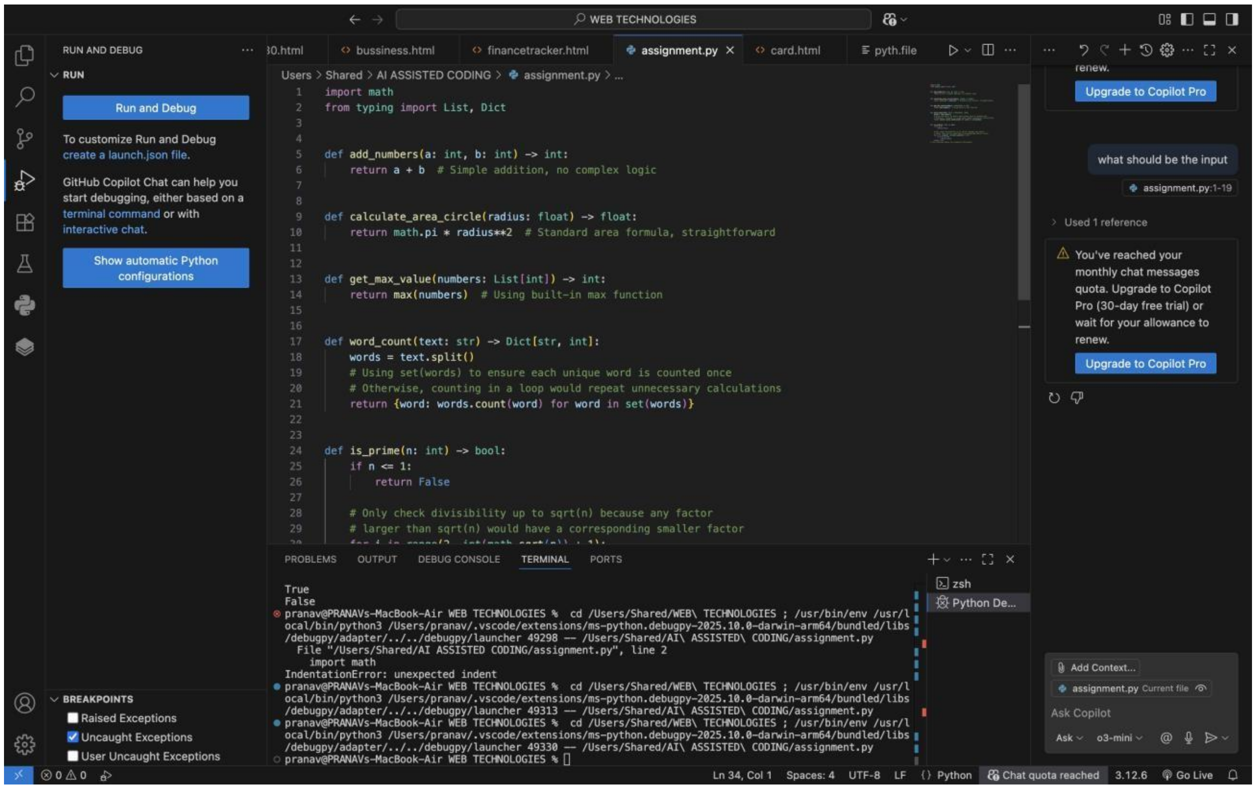
o Python code with concise, context-aware inline comments for complex logic blocks

PROMPT USED :

Write a code to add meaningful inline comments to a python program explaining only complex logic parts .

Explain only complex structured syntac and usage on tricky or non intuitive code sections

CODE GENERATED :

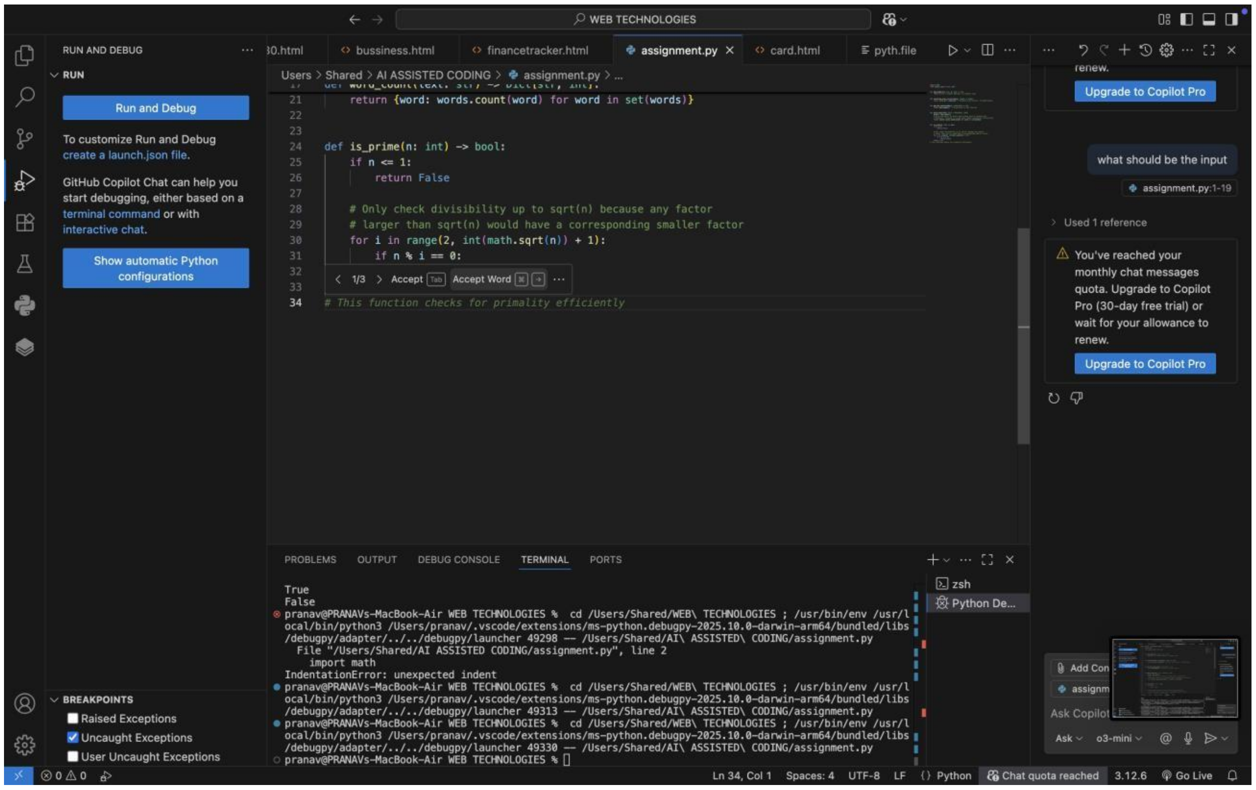


```python
import math
from typing import List, Dict


def add_numbers(a: int, b: int) -> int:
    return a + b  # Simple addition, no complex logic


def calculate_area_circle(radius: float) -> float:
    return math.pi * radius**2  # Standard area formula, straightforward


def get_max_value(numbers: List[int]) -> int:
    return max(numbers)  # Using built-in max function


def word_count(text: str) -> Dict[str, int]:
    words = text.split()
    # Using set(words) to ensure each unique word is counted once
    # Otherwise, counting in a loop would repeat unnecessary calculations
    return {word: words.count(word) for word in set(words)}


def is_prime(n: int) -> bool:
    if n <= 1:
        return False

    # Only check divisibility up to sqrt(n) because any factor
    # larger than sqrt(n) would have a corresponding smaller factor
```

```
True
False
pranav@PRANAVs-MacBook-Air WEB TECHNOLOGIES %  cd /Users/Shared/WEB\ TECHNOLOGIES ; /usr/bin/env /usr/l
ocal/bin/python3 /Users/pranav/.vscode/extensions/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs
/debugpy/adapter/../../debugpy/launcher 49298 -- /Users/Shared/AI\ ASSISTED\ CODING/assignment.py
  File "/Users/Shared/AI ASSISTED CODING/assignment.py", line 2
    import math
IndentationError: unexpected indent
pranav@PRANAVs-MacBook-Air WEB TECHNOLOGIES %  cd /Users/Shared/WEB\ TECHNOLOGIES ; /usr/bin/env /usr/l
ocal/bin/python3 /Users/pranav/.vscode/extensions/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs
/debugpy/adapter/../../debugpy/launcher 49313 -- /Users/Shared/AI\ ASSISTED\ CODING/assignment.py
pranav@PRANAVs-MacBook-Air WEB TECHNOLOGIES %  cd /Users/Shared/WEB\ TECHNOLOGIES ; /usr/bin/env /usr/l
ocal/bin/python3 /Users/pranav/.vscode/extensions/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs
/debugpy/adapter/../../debugpy/launcher 49330 -- /Users/Shared/AI\ ASSISTED\ CODING/assignment.py
pranav@PRANAVs-MacBook-Air WEB TECHNOLOGIES % 
```

what should be the input

You've reached your monthly chat messages quota. Upgrade to Copilot Pro (30-day free trial) or wait for your allowance to renew.



```python
    return {word: words.count(word) for word in set(words)}


def is_prime(n: int) -> bool:
    if n <= 1:
        return False

    # Only check divisibility up to sqrt(n) because any factor
    # larger than sqrt(n) would have a corresponding smaller factor
    for i in range(2, int(math.sqrt(n)) + 1):
        if n % i == 0:


# This function checks for primality efficiently
```

```
True
False
pranav@PRANAVs-MacBook-Air WEB TECHNOLOGIES %  cd /Users/Shared/WEB\ TECHNOLOGIES ; /usr/bin/env /usr/l
ocal/bin/python3 /Users/pranav/.vscode/extensions/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs
/debugpy/adapter/../../debugpy/launcher 49298 -- /Users/Shared/AI\ ASSISTED\ CODING/assignment.py
  File "/Users/Shared/AI ASSISTED CODING/assignment.py", line 2
    import math
IndentationError: unexpected indent
pranav@PRANAVs-MacBook-Air WEB TECHNOLOGIES %  cd /Users/Shared/WEB\ TECHNOLOGIES ; /usr/bin/env /usr/l
ocal/bin/python3 /Users/pranav/.vscode/extensions/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs
/debugpy/adapter/../../debugpy/launcher 49313 -- /Users/Shared/AI\ ASSISTED\ CODING/assignment.py
pranav@PRANAVs-MacBook-Air WEB TECHNOLOGIES %  cd /Users/Shared/WEB\ TECHNOLOGIES ; /usr/bin/env /usr/l
ocal/bin/python3 /Users/pranav/.vscode/extensions/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs
/debugpy/adapter/../../debugpy/launcher 49330 -- /Users/Shared/AI\ ASSISTED\ CODING/assignment.py
pranav@PRANAVs-MacBook-Air WEB TECHNOLOGIES % 
```

what should the input

You've reached your monthly chat messages quota. Upgrade to Copilot Pro (30-day free trial) or wait for your allowance to renew.

OBSERVATIONS :

1. The Python script was **analyzed to identify non-intuitive or complex logic blocks,** ignoring obvious syntax and straightforward statements.
2. Inline comments were added **directly above or beside lines of code** where the reasoning might not be immediately clear, improving code readability and maintainability.
3. Examples of complex logic that were commented include:
    a. In word_count, using set(words) to avoid redundant counting of the same word multiple times.
    b. In is_prime, checking divisibility only up to sqrt(n) for efficiency, since any factor larger than sqrt(n) would have a corresponding smaller factor.
4. Obvious operations, like return a + b or return max(numbers), were **intentionally not commented**, keeping the code clean and concise.
5. The added comments help a **future developer or reviewer** understand why certain approaches were used without cluttering the code with unnecessary explanations.

6. Overall, the code is now **more maintainable, self-explanatory for complex sections**, and easier to debug or extend in the future.

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.

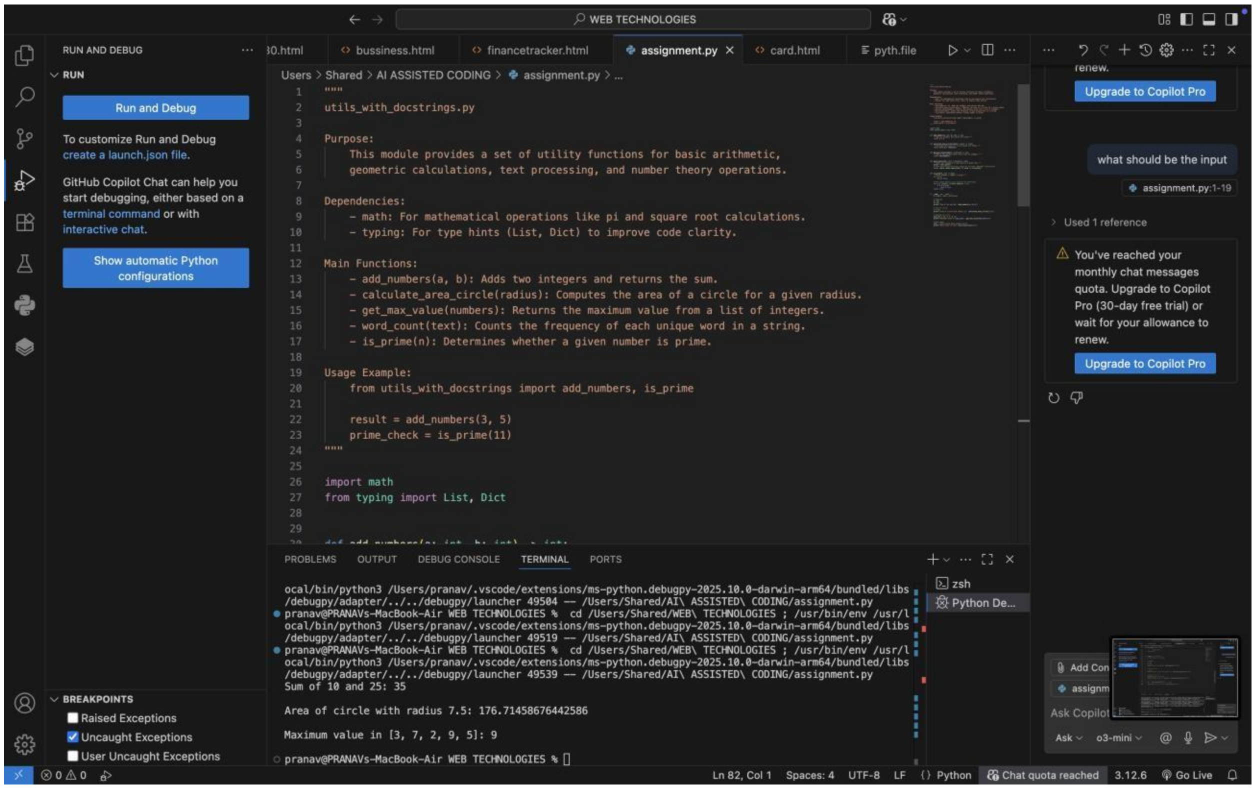
• Instructions:
o Supply the entire Python file to AI.
o Instruct AI to write a single multi-line docstring at the top of the file.
o Ensure the docstring clearly describes functionality and usage without rewriting the entire code.
• Expected Output #3:
o A complete, clear, and concise module-level docstring at the beginning of the file.

PROMPT USED :

WRITE A PURPOSE , DEPENDENCIES, MAIN FUNCTIONS/CLASSES OF A PYTHON FILE WHICH IS ALREADY EXISTING

Write a single multi-line docstring at the top of the file.

CODE GENERATED :

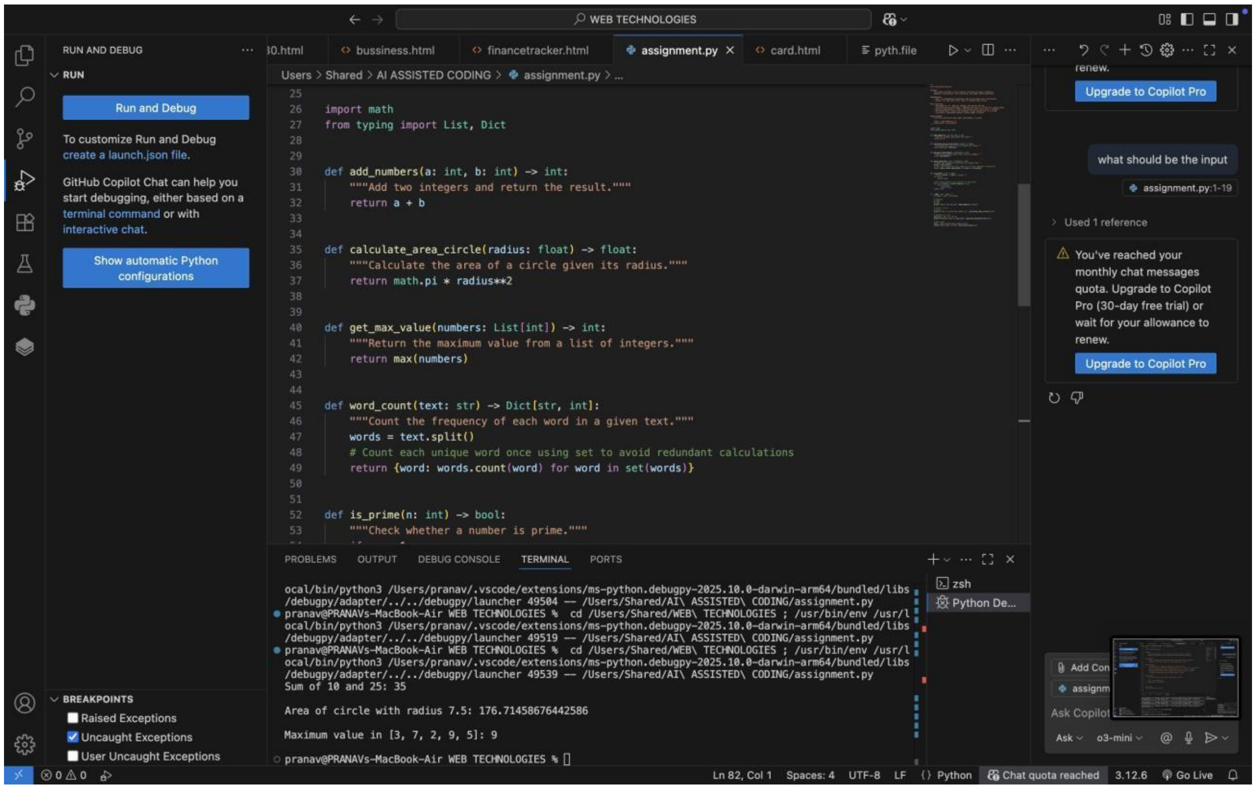


```
"""
utils_with_docstrings.py

Purpose:
    This module provides a set of utility functions for basic arithmetic,
    geometric calculations, text processing, and number theory operations.

Dependencies:
    - math: For mathematical operations like pi and square root calculations.
    - typing: For type hints (List, Dict) to improve code clarity.

Main Functions:
    - add_numbers(a, b): Adds two integers and returns the sum.
    - calculate_area_circle(radius): Computes the area of a circle for a given radius.
    - get_max_value(numbers): Returns the maximum value from a list of integers.
    - word_count(text): Counts the frequency of each unique word in a string.
    - is_prime(n): Determines whether a given number is prime.

Usage Example:
    from utils_with_docstrings import add_numbers, is_prime

    result = add_numbers(3, 5)
    prime_check = is_prime(11)
"""

import math
from typing import List, Dict
```

```
ocal/bin/python3 /Users/pranav/.vscode/extensions/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs
/debugpy/adapter/../../debugpy/launcher 49504 -- /Users/Shared/AI\ ASSISTED\ CODING/assignment.py
pranav@PRANAVs-MacBook-Air WEB TECHNOLOGIES %  cd /Users/Shared/WEB\ TECHNOLOGIES ; /usr/bin/env /usr/l
ocal/bin/python3 /Users/pranav/.vscode/extensions/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs
/debugpy/adapter/../../debugpy/launcher 49519 -- /Users/Shared/AI\ ASSISTED\ CODING/assignment.py
pranav@PRANAVs-MacBook-Air WEB TECHNOLOGIES %  cd /Users/Shared/WEB\ TECHNOLOGIES ; /usr/bin/env /usr/l
ocal/bin/python3 /Users/pranav/.vscode/extensions/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs
/debugpy/adapter/../../debugpy/launcher 49539 -- /Users/Shared/AI\ ASSISTED\ CODING/assignment.py
Sum of 10 and 25: 35

Area of circle with radius 7.5: 176.71458676442586

Maximum value in [3, 7, 2, 9, 5]: 9

pranav@PRANAVs-MacBook-Air WEB TECHNOLOGIES % 
```

what should be the input

You've reached your monthly chat messages quota. Upgrade to Copilot Pro (30-day free trial) or wait for your allowance to renew.

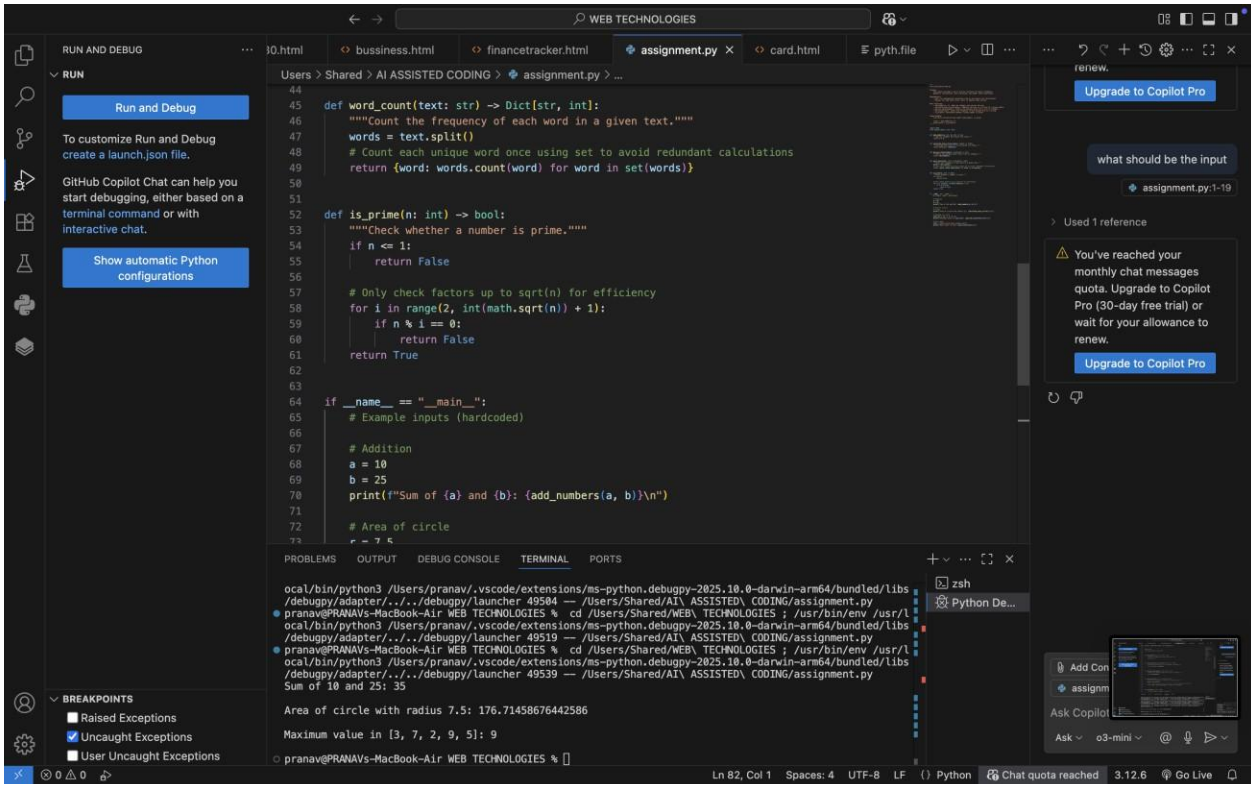


```
import math
from typing import List, Dict


def add_numbers(a: int, b: int) -> int:
    """Add two integers and return the result."""
    return a + b


def calculate_area_circle(radius: float) -> float:
    """Calculate the area of a circle given its radius."""
    return math.pi * radius**2


def get_max_value(numbers: List[int]) -> int:
    """Return the maximum value from a list of integers."""
    return max(numbers)


def word_count(text: str) -> Dict[str, int]:
    """Count the frequency of each word in a given text."""
    words = text.split()
    # Count each unique word once using set to avoid redundant calculations
    return {word: words.count(word) for word in set(words)}


def is_prime(n: int) -> bool:
    """Check whether a number is prime."""
```

```
ocal/bin/python3 /Users/pranav/.vscode/extensions/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs
/debugpy/adapter/../../debugpy/launcher 49504 -- /Users/Shared/AI\ ASSISTED\ CODING/assignment.py
pranav@PRANAVs-MacBook-Air WEB TECHNOLOGIES %  cd /Users/Shared/WEB\ TECHNOLOGIES ; /usr/bin/env /usr/l
ocal/bin/python3 /Users/pranav/.vscode/extensions/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs
/debugpy/adapter/../../debugpy/launcher 49519 -- /Users/Shared/AI\ ASSISTED\ CODING/assignment.py
pranav@PRANAVs-MacBook-Air WEB TECHNOLOGIES %  cd /Users/Shared/WEB\ TECHNOLOGIES ; /usr/bin/env /usr/l
ocal/bin/python3 /Users/pranav/.vscode/extensions/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs
/debugpy/adapter/../../debugpy/launcher 49539 -- /Users/Shared/AI\ ASSISTED\ CODING/assignment.py
Sum of 10 and 25: 35

Area of circle with radius 7.5: 176.71458676442586

Maximum value in [3, 7, 2, 9, 5]: 9

pranav@PRANAVs-MacBook-Air WEB TECHNOLOGIES % 
```

what should be the input

You've reached your monthly chat messages quota. Upgrade to Copilot Pro (30-day free trial) or wait for your allowance to renew.

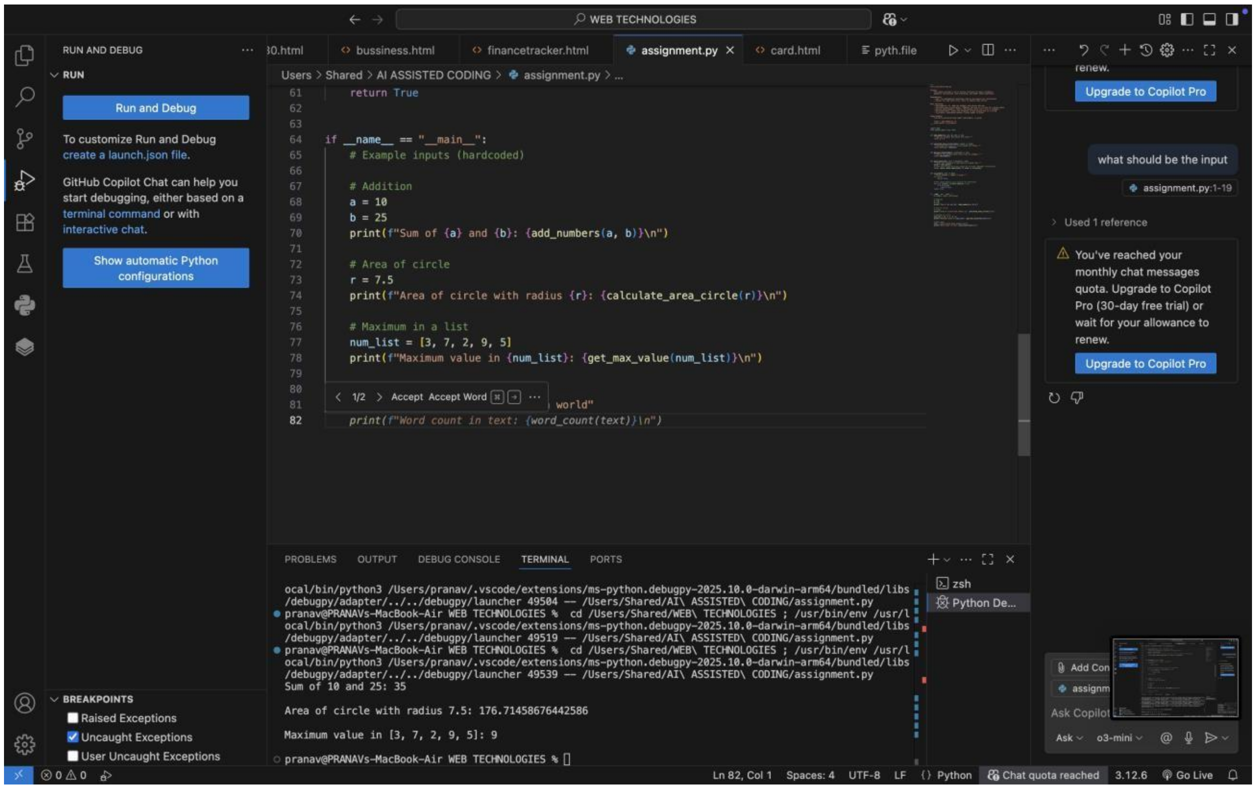


```python
def word_count(text: str) -> Dict[str, int]:
    """Count the frequency of each word in a given text."""
    words = text.split()
    # Count each unique word once using set to avoid redundant calculations
    return {word: words.count(word) for word in set(words)}


def is_prime(n: int) -> bool:
    """Check whether a number is prime."""
    if n <= 1:
        return False

    # Only check factors up to sqrt(n) for efficiency
    for i in range(2, int(math.sqrt(n)) + 1):
        if n % i == 0:
            return False
    return True


if __name__ == "__main__":
    # Example inputs (hardcoded)

    # Addition
    a = 10
    b = 25
    print(f"Sum of {a} and {b}: {add_numbers(a, b)}\n")

    # Area of circle
```

```
ocal/bin/python3 /Users/pranav/.vscode/extensions/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs
/debugpy/adapter/../../debugpy/launcher 49504 -- /Users/Shared/AI\ ASSISTED\ CODING/assignment.py
pranav@PRANAVs-MacBook-Air WEB TECHNOLOGIES %  cd /Users/Shared/WEB\ TECHNOLOGIES ; /usr/bin/env /usr/l
ocal/bin/python3 /Users/pranav/.vscode/extensions/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs
/debugpy/adapter/../../debugpy/launcher 49519 -- /Users/Shared/AI\ ASSISTED\ CODING/assignment.py
pranav@PRANAVs-MacBook-Air WEB TECHNOLOGIES %  cd /Users/Shared/WEB\ TECHNOLOGIES ; /usr/bin/env /usr/l
ocal/bin/python3 /Users/pranav/.vscode/extensions/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs
/debugpy/adapter/../../debugpy/launcher 49539 -- /Users/Shared/AI\ ASSISTED\ CODING/assignment.py
Sum of 10 and 25: 35

Area of circle with radius 7.5: 176.71458676442586

Maximum value in [3, 7, 2, 9, 5]: 9

pranav@PRANAVs-MacBook-Air WEB TECHNOLOGIES %
```

what should be the input

You've reached your monthly chat messages quota. Upgrade to Copilot Pro (30-day free trial) or wait for your allowance to renew.



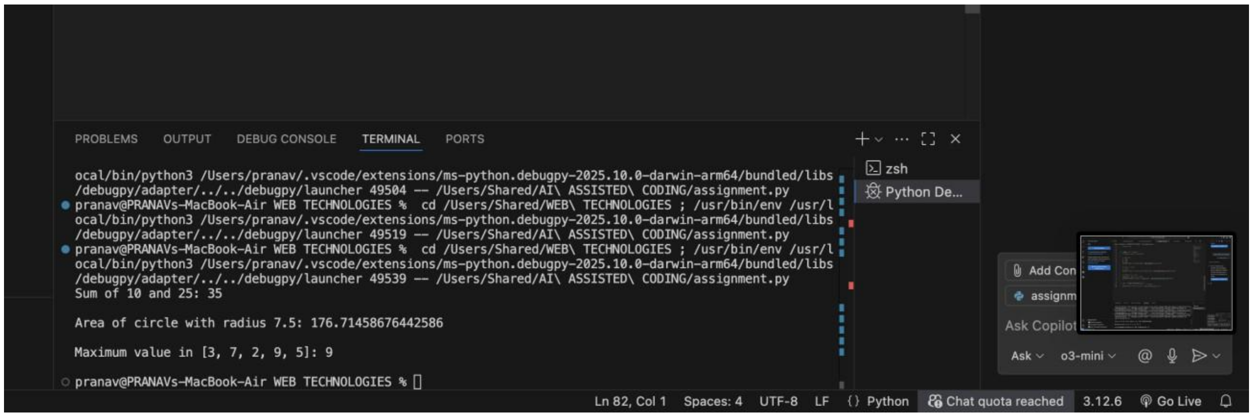
```python
    return True


if __name__ == "__main__":
    # Example inputs (hardcoded)

    # Addition
    a = 10
    b = 25
    print(f"Sum of {a} and {b}: {add_numbers(a, b)}\n")

    # Area of circle
    r = 7.5
    print(f"Area of circle with radius {r}: {calculate_area_circle(r)}\n")

    # Maximum in a list
    num_list = [3, 7, 2, 9, 5]
    print(f"Maximum value in {num_list}: {get_max_value(num_list)}\n")


                                     world"
    print(f"Word count in text: {word_count(text)}\n")
```

```
ocal/bin/python3 /Users/pranav/.vscode/extensions/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs
/debugpy/adapter/../../debugpy/launcher 49504 -- /Users/Shared/AI\ ASSISTED\ CODING/assignment.py
pranav@PRANAVs-MacBook-Air WEB TECHNOLOGIES %  cd /Users/Shared/WEB\ TECHNOLOGIES ; /usr/bin/env /usr/l
ocal/bin/python3 /Users/pranav/.vscode/extensions/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs
/debugpy/adapter/../../debugpy/launcher 49519 -- /Users/Shared/AI\ ASSISTED\ CODING/assignment.py
pranav@PRANAVs-MacBook-Air WEB TECHNOLOGIES %  cd /Users/Shared/WEB\ TECHNOLOGIES ; /usr/bin/env /usr/l
ocal/bin/python3 /Users/pranav/.vscode/extensions/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs
/debugpy/adapter/../../debugpy/launcher 49539 -- /Users/Shared/AI\ ASSISTED\ CODING/assignment.py
Sum of 10 and 25: 35

Area of circle with radius 7.5: 176.71458676442586

Maximum value in [3, 7, 2, 9, 5]: 9

pranav@PRANAVs-MacBook-Air WEB TECHNOLOGIES %
```

what should be the input

You've reached your monthly chat messages quota. Upgrade to Copilot Pro (30-day free trial) or wait for your allowance to renew.

OUTPUT GENERATED :



```
ocal/bin/python3 /Users/pranav/.vscode/extensions/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs
/debugpy/adapter/../../debugpy/launcher 49504 -- /Users/Shared/AI\ ASSISTED\ CODING/assignment.py
pranav@PRANAVs-MacBook-Air WEB TECHNOLOGIES %  cd /Users/Shared/WEB\ TECHNOLOGIES ; /usr/bin/env /usr/l
ocal/bin/python3 /Users/pranav/.vscode/extensions/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs
/debugpy/adapter/../../debugpy/launcher 49519 -- /Users/Shared/AI\ ASSISTED\ CODING/assignment.py
pranav@PRANAVs-MacBook-Air WEB TECHNOLOGIES %  cd /Users/Shared/WEB\ TECHNOLOGIES ; /usr/bin/env /usr/l
ocal/bin/python3 /Users/pranav/.vscode/extensions/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs
/debugpy/adapter/../../debugpy/launcher 49539 -- /Users/Shared/AI\ ASSISTED\ CODING/assignment.py
Sum of 10 and 25: 35

Area of circle with radius 7.5: 176.71458676442586

Maximum value in [3, 7, 2, 9, 5]: 9

pranav@PRANAVs-MacBook-Air WEB TECHNOLOGIES % 
```

Task Description #4 (Documentation – Convert Comments to
Structured Docstrings)

• Task: Use AI to transform existing inline comments into
structured function docstrings following Google style.

• Instructions:

o Provide AI with Python code containing inline comments.

o Ask AI to move relevant details from comments into
function docstrings.

o Verify that the new docstrings keep the meaning intact
while improving structure.

• Expected Output #4:

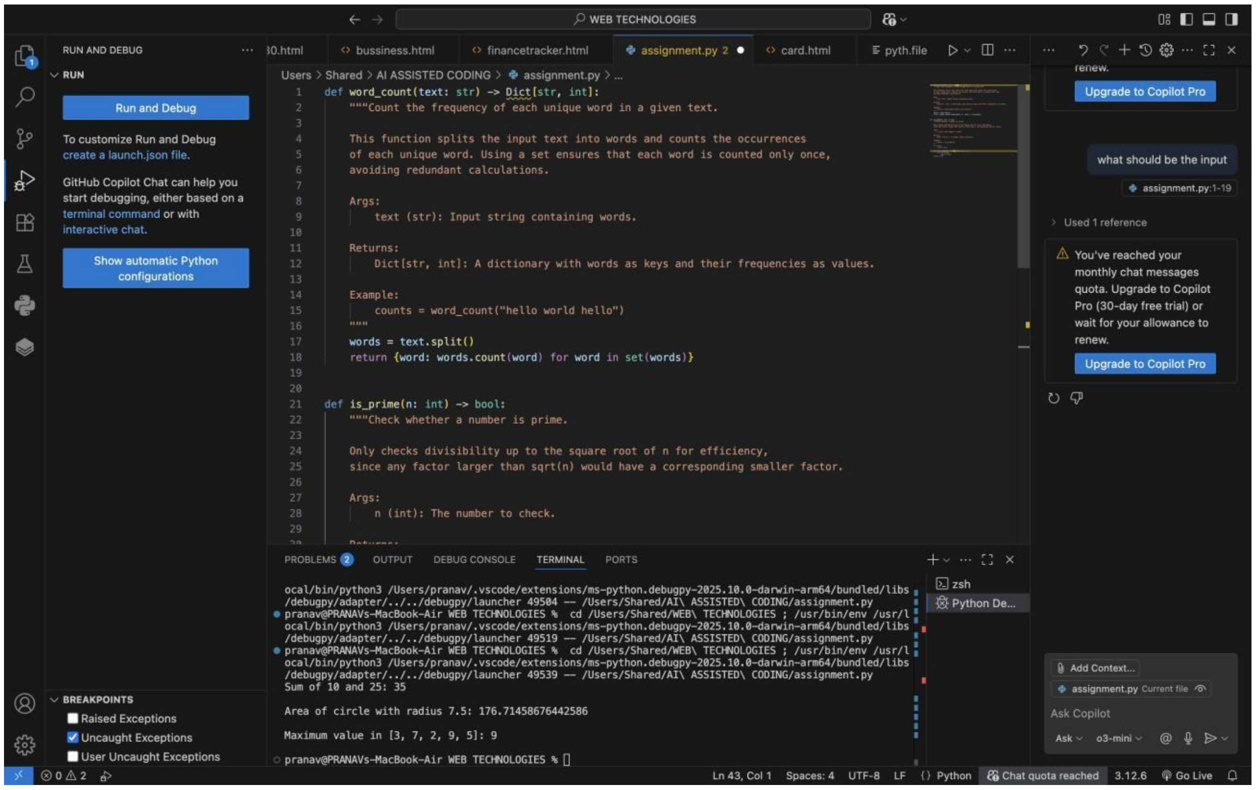
o Python code with comments replaced by clear,
standardized docstrings

PROMPT USED :

WRITE A structured function docstrings using existing inline comments

Comments to be changed to function docstrings.

CODE GENERATED :



Users > Shared > AI ASSISTED CODING > assignment.py > ...

```python
def word_count(text: str) -> Dict[str, int]:
    """Count the frequency of each unique word in a given text.

    This function splits the input text into words and counts the occurrences
    of each unique word. Using a set ensures that each word is counted only once,
    avoiding redundant calculations.

    Args:
        text (str): Input string containing words.

    Returns:
        Dict[str, int]: A dictionary with words as keys and their frequencies as values.

    Example:
        counts = word_count("hello world hello")
    """
    words = text.split()
    return {word: words.count(word) for word in set(words)}


def is_prime(n: int) -> bool:
    """Check whether a number is prime.

    Only checks divisibility up to the square root of n for efficiency,
    since any factor larger than sqrt(n) would have a corresponding smaller factor.

    Args:
        n (int): The number to check.

```

```
ocal/bin/python3 /Users/pranav/.vscode/extensions/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs
/debugpy/adapter/../../debugpy/launcher 49504 -- /Users/Shared/AI\ ASSISTED\ CODING/assignment.py
pranav@PRANAVs-MacBook-Air WEB TECHNOLOGIES %  cd /Users/Shared/WEB\ TECHNOLOGIES ; /usr/bin/env /usr/l
ocal/bin/python3 /Users/pranav/.vscode/extensions/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs
/debugpy/adapter/../../debugpy/launcher 49519 -- /Users/Shared/AI\ ASSISTED\ CODING/assignment.py
pranav@PRANAVs-MacBook-Air WEB TECHNOLOGIES %  cd /Users/Shared/WEB\ TECHNOLOGIES ; /usr/bin/env /usr/l
ocal/bin/python3 /Users/pranav/.vscode/extensions/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs
/debugpy/adapter/../../debugpy/launcher 49539 -- /Users/Shared/AI\ ASSISTED\ CODING/assignment.py
Sum of 10 and 25: 35

Area of circle with radius 7.5: 176.71458676442586

Maximum value in [3, 7, 2, 9, 5]: 9

pranav@PRANAVs-MacBook-Air WEB TECHNOLOGIES % 
```

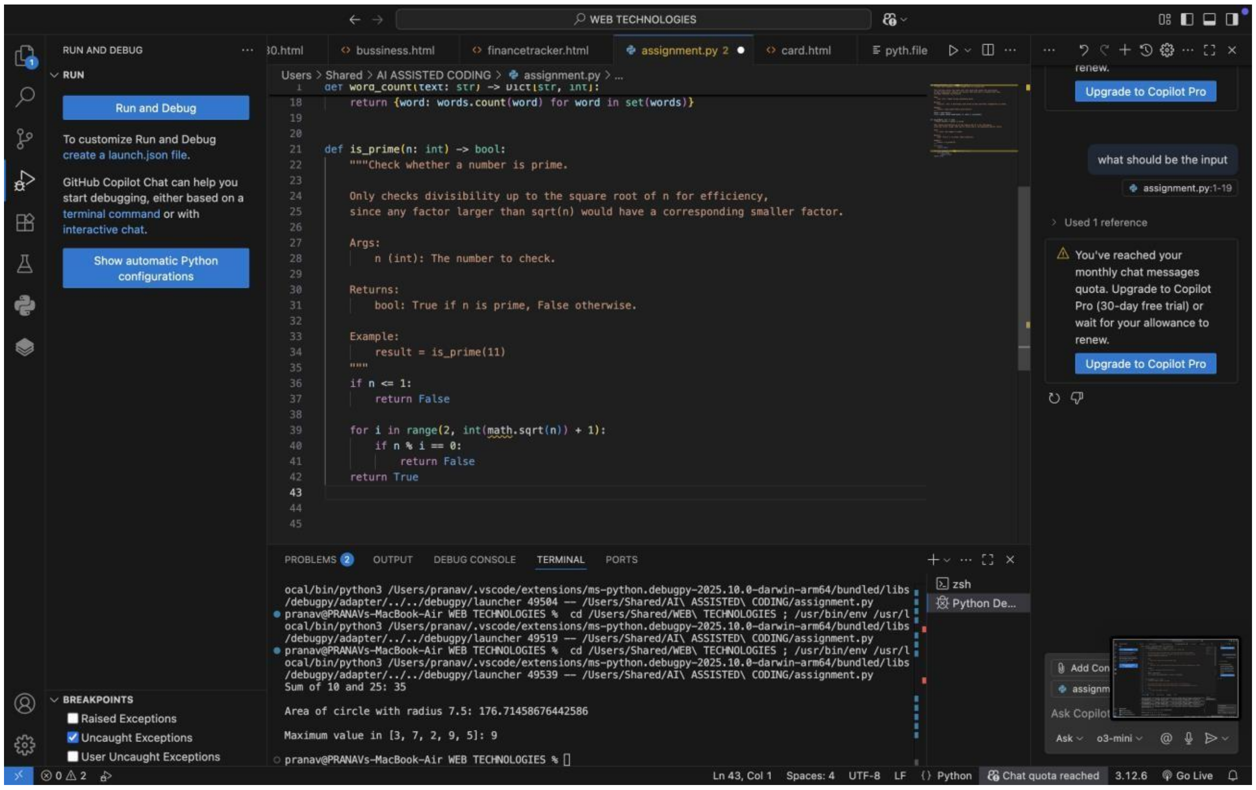
what should be the input

assignment.py:1-19

Used 1 reference

You've reached your monthly chat messages quota. Upgrade to Copilot Pro (30-day free trial) or wait for your allowance to renew.

Upgrade to Copilot Pro



Users > Shared > AI ASSISTED CODING > assignment.py > ...

```python
def word_count(text: str) -> Dict[str, int]:
    return {word: words.count(word) for word in set(words)}


def is_prime(n: int) -> bool:
    """Check whether a number is prime.

    Only checks divisibility up to the square root of n for efficiency,
    since any factor larger than sqrt(n) would have a corresponding smaller factor.

    Args:
        n (int): The number to check.

    Returns:
        bool: True if n is prime, False otherwise.

    Example:
        result = is_prime(11)
    """
    if n <= 1:
        return False

    for i in range(2, int(math.sqrt(n)) + 1):
        if n % i == 0:
            return False
    return True


```

```
ocal/bin/python3 /Users/pranav/.vscode/extensions/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs
/debugpy/adapter/../../debugpy/launcher 49504 -- /Users/Shared/AI\ ASSISTED\ CODING/assignment.py
pranav@PRANAVs-MacBook-Air WEB TECHNOLOGIES %  cd /Users/Shared/WEB\ TECHNOLOGIES ; /usr/bin/env /usr/l
ocal/bin/python3 /Users/pranav/.vscode/extensions/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs
/debugpy/adapter/../../debugpy/launcher 49519 -- /Users/Shared/AI\ ASSISTED\ CODING/assignment.py
pranav@PRANAVs-MacBook-Air WEB TECHNOLOGIES %  cd /Users/Shared/WEB\ TECHNOLOGIES ; /usr/bin/env /usr/l
ocal/bin/python3 /Users/pranav/.vscode/extensions/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs
/debugpy/adapter/../../debugpy/launcher 49539 -- /Users/Shared/AI\ ASSISTED\ CODING/assignment.py
Sum of 10 and 25: 35

Area of circle with radius 7.5: 176.71458676442586

Maximum value in [3, 7, 2, 9, 5]: 9

pranav@PRANAVs-MacBook-Air WEB TECHNOLOGIES % 
```

what should be the input

assignment.py:1-19

Used 1 reference

You've reached your monthly chat messages quota. Upgrade to Copilot Pro (30-day free trial) or wait for your allowance to renew.

Upgrade to Copilot Pro

OBSERVATIONS:

- The given Python file originally contained **inline comments** to explain non-trivial logic (e.g., using set(words) in word_count and checking factors up to sqrt(n) in is_prime).
- These inline comments were **transformed into structured Google-style docstrings**, placed directly inside the functions.
- Each docstring now provides:
- A **clear description** of the function's purpose.
- **Parameter definitions** with type hints.
- The **return type and meaning**.
- An **example usage** for clarity.
- Redundant inline comments were removed to keep the code clean and avoid duplication, while still retaining all the meaningful explanations.
- This improved the **readability and maintainability** of the code by standardizing documentation in one place (docstrings), making it easier for future developers and tools (like IDEs or documentation generators) to understand the module.
- Overall, the code is now **self-documented, professional, and aligned with best practices** for Python documentation.

Task Description #5 (Documentation – Review and Correct Docstrings)

• Task: Use AI to identify and correct inaccuracies in existing docstrings.

• Instructions:

o Provide Python code with outdated or incorrect docstrings.

o Instruct AI to rewrite each docstring to match the current code behavior.

o Ensure corrections follow Google-style formatting.

• Expected Output #5:

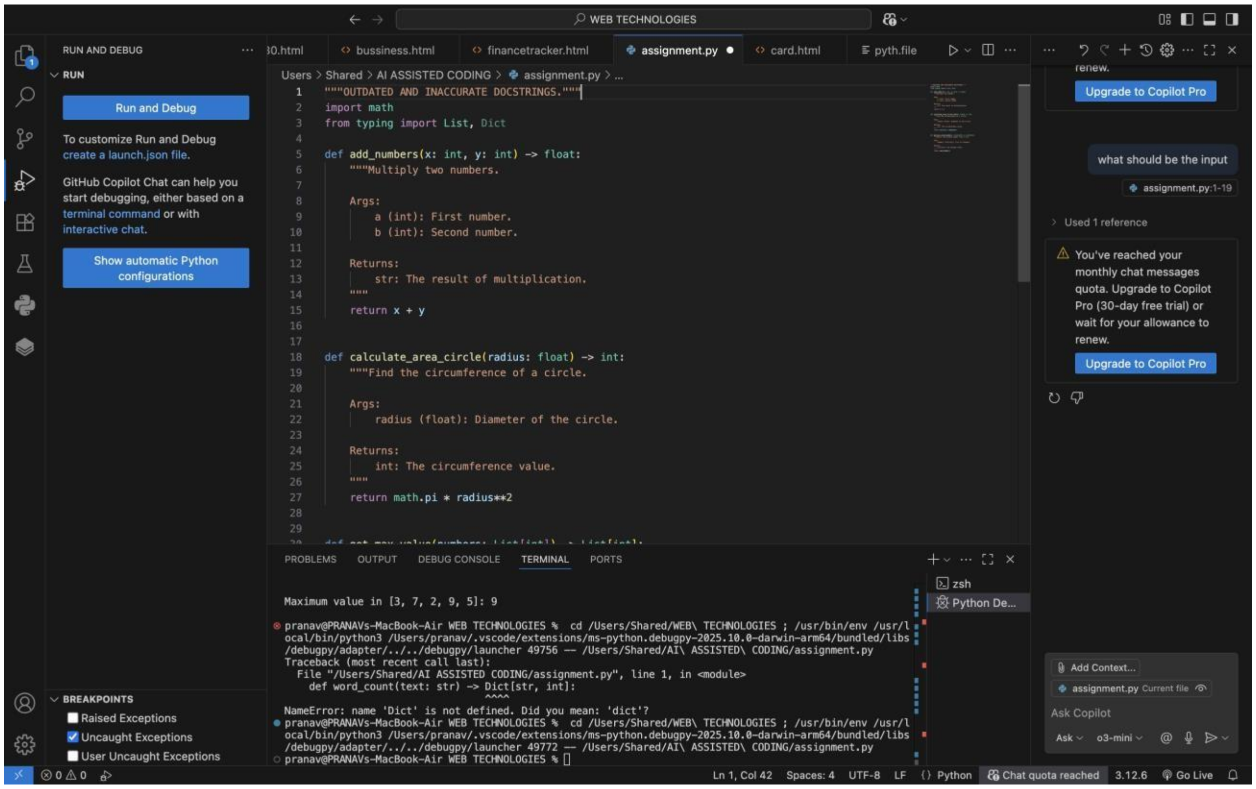
o Python file with updated, accurate, and standardized docstrings

PROMPT USED :

To identify and correct inaccuracies in existing docstrings.

Ensure corrections follow Google-style formatting.

OUTDATED OR INACCURATE DOCSTRINGS :



To customize Run and Debug create a launch.json file.

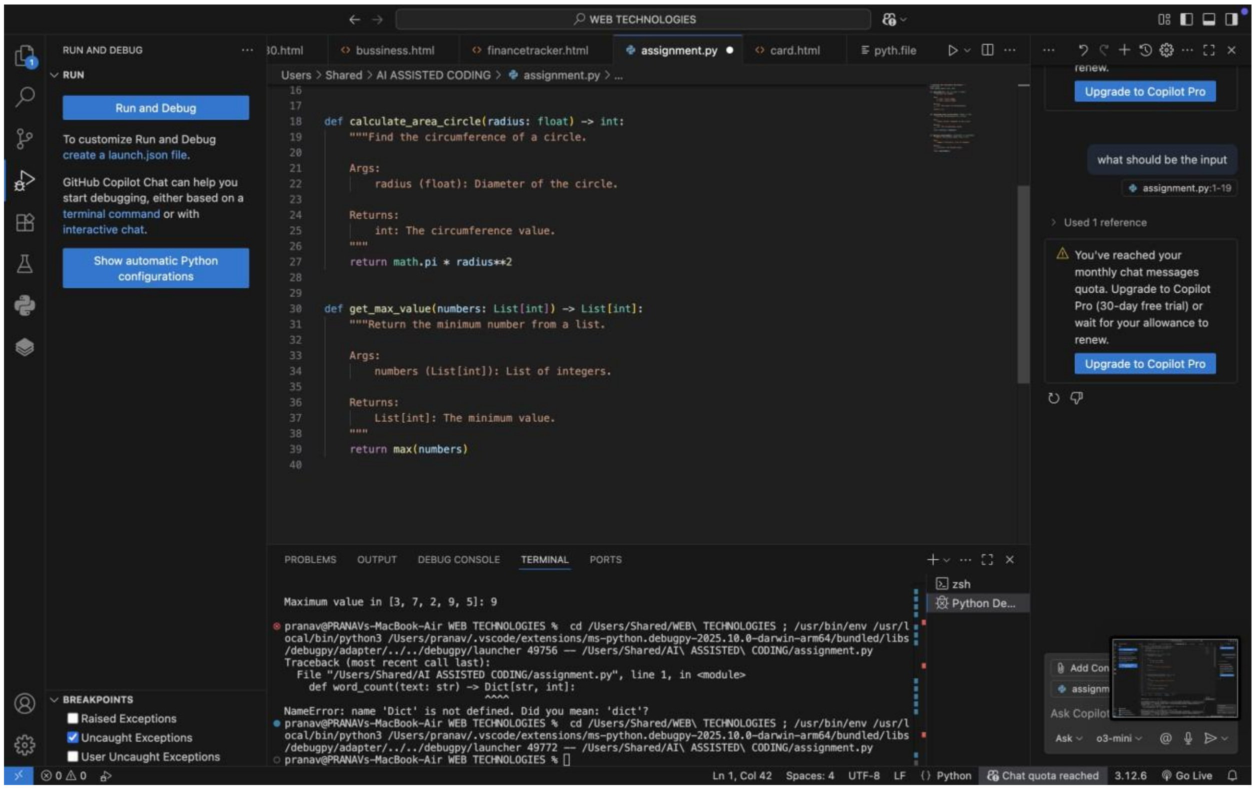
GitHub Copilot Chat can help you start debugging, either based on a terminal command or with interactive chat.

```python
"""OUTDATED AND INACCURATE DOCSTRINGS."""
import math
from typing import List, Dict

def add_numbers(x: int, y: int) -> float:
    """Multiply two numbers.

    Args:
        a (int): First number.
        b (int): Second number.

    Returns:
        str: The result of multiplication.
    """
    return x + y


def calculate_area_circle(radius: float) -> int:
    """Find the circumference of a circle.

    Args:
        radius (float): Diameter of the circle.

    Returns:
        int: The circumference value.
    """
    return math.pi * radius**2
```

```
Maximum value in [3, 7, 2, 9, 5]: 9

pranav@PRANAVs-MacBook-Air WEB TECHNOLOGIES %  cd /Users/Shared/WEB\ TECHNOLOGIES ; /usr/bin/env /usr/l
ocal/bin/python3 /Users/pranav/.vscode/extensions/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs
/debugpy/adapter/../../debugpy/launcher 49756 -- /Users/Shared/AI\ ASSISTED\ CODING/assignment.py
Traceback (most recent call last):
  File "/Users/Shared/AI ASSISTED CODING/assignment.py", line 1, in <module>
    def word_count(text: str) -> Dict[str, int]:
                                 ^^^^
NameError: name 'Dict' is not defined. Did you mean: 'dict'?
pranav@PRANAVs-MacBook-Air WEB TECHNOLOGIES %  cd /Users/Shared/WEB\ TECHNOLOGIES ; /usr/bin/env /usr/l
ocal/bin/python3 /Users/pranav/.vscode/extensions/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs
/debugpy/adapter/../../debugpy/launcher 49772 -- /Users/Shared/AI\ ASSISTED\ CODING/assignment.py
pranav@PRANAVs-MacBook-Air WEB TECHNOLOGIES % 
```

what should be the input

You've reached your monthly chat messages quota. Upgrade to Copilot Pro (30-day free trial) or wait for your allowance to renew.



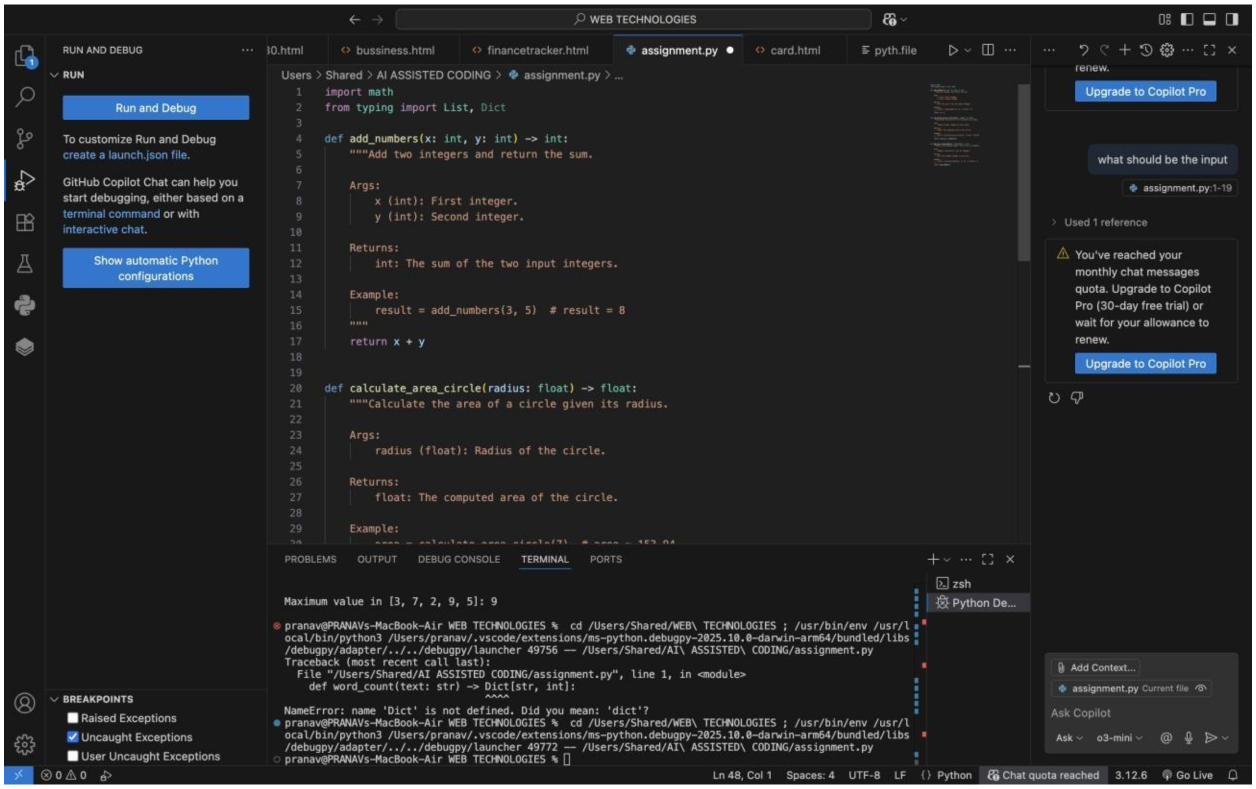
```python
def calculate_area_circle(radius: float) -> int:
    """Find the circumference of a circle.

    Args:
        radius (float): Diameter of the circle.

    Returns:
        int: The circumference value.
    """
    return math.pi * radius**2


def get_max_value(numbers: List[int]) -> List[int]:
    """Return the minimum number from a list.

    Args:
        numbers (List[int]): List of integers.

    Returns:
        List[int]: The minimum value.
    """
    return max(numbers)
```

```
Maximum value in [3, 7, 2, 9, 5]: 9

pranav@PRANAVs-MacBook-Air WEB TECHNOLOGIES %  cd /Users/Shared/WEB\ TECHNOLOGIES ; /usr/bin/env /usr/l
ocal/bin/python3 /Users/pranav/.vscode/extensions/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs
/debugpy/adapter/../../debugpy/launcher 49756 -- /Users/Shared/AI\ ASSISTED\ CODING/assignment.py
Traceback (most recent call last):
  File "/Users/Shared/AI ASSISTED CODING/assignment.py", line 1, in <module>
    def word_count(text: str) -> Dict[str, int]:
                                 ^^^^
NameError: name 'Dict' is not defined. Did you mean: 'dict'?
pranav@PRANAVs-MacBook-Air WEB TECHNOLOGIES %  cd /Users/Shared/WEB\ TECHNOLOGIES ; /usr/bin/env /usr/l
ocal/bin/python3 /Users/pranav/.vscode/extensions/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs
/debugpy/adapter/../../debugpy/launcher 49772 -- /Users/Shared/AI\ ASSISTED\ CODING/assignment.py
pranav@PRANAVs-MacBook-Air WEB TECHNOLOGIES % 
```

what should be the input

You've reached your monthly chat messages quota. Upgrade to Copilot Pro (30-day free trial) or wait for your allowance to renew.

UPDATED DOCSTRING CODE :

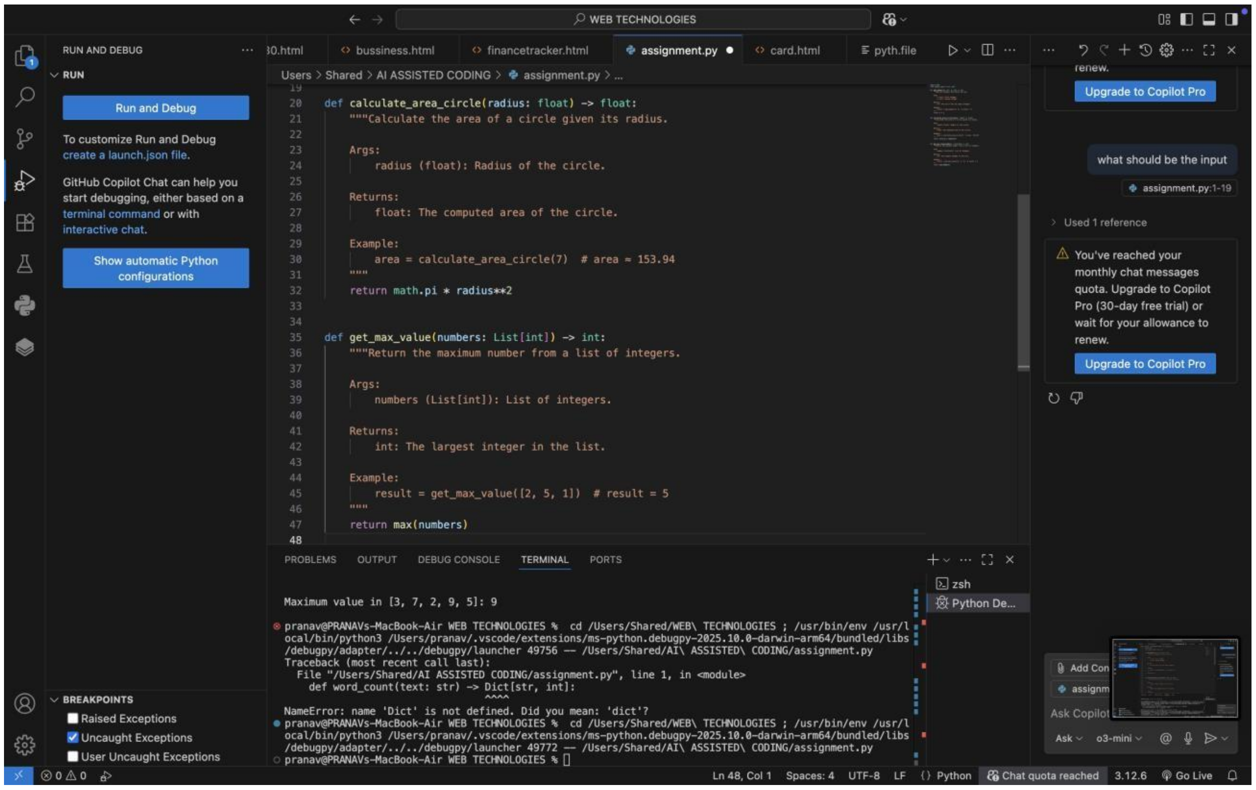


```python
import math
from typing import List, Dict

def add_numbers(x: int, y: int) -> int:
    """Add two integers and return the sum.

    Args:
        x (int): First integer.
        y (int): Second integer.

    Returns:
        int: The sum of the two input integers.

    Example:
        result = add_numbers(3, 5)  # result = 8
    """
    return x + y


def calculate_area_circle(radius: float) -> float:
    """Calculate the area of a circle given its radius.

    Args:
        radius (float): Radius of the circle.

    Returns:
        float: The computed area of the circle.

    Example:
```

```
Maximum value in [3, 7, 2, 9, 5]: 9

pranav@PRANAVs-MacBook-Air WEB TECHNOLOGIES %  cd /Users/Shared/WEB\ TECHNOLOGIES ; /usr/bin/env /usr/l
ocal/bin/python3 /Users/pranav/.vscode/extensions/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs
/debugpy/adapter/../../debugpy/launcher 49756 -- /Users/Shared/AI\ ASSISTED\ CODING/assignment.py
Traceback (most recent call last):
  File "/Users/Shared/AI ASSISTED CODING/assignment.py", line 1, in <module>
    def word_count(text: str) -> Dict[str, int]:
                                 ^^^^
NameError: name 'Dict' is not defined. Did you mean: 'dict'?
pranav@PRANAVs-MacBook-Air WEB TECHNOLOGIES %  cd /Users/Shared/WEB\ TECHNOLOGIES ; /usr/bin/env /usr/l
ocal/bin/python3 /Users/pranav/.vscode/extensions/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs
/debugpy/adapter/../../debugpy/launcher 49772 -- /Users/Shared/AI\ ASSISTED\ CODING/assignment.py
pranav@PRANAVs-MacBook-Air WEB TECHNOLOGIES % 
```

what should be the input

You've reached your monthly chat messages quota. Upgrade to Copilot Pro (30-day free trial) or wait for your allowance to renew.


RUN AND DEBUG
RUN
Run and Debug
Show automatic Python configurations
BREAKPOINTS
Raised Exceptions
Uncaught Exceptions
User Uncaught Exceptions
WEB TECHNOLOGIES
Users > Shared > AI ASSISTED CODING > assignment.py > ...
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Upgrade to Copilot Pro
Used 1 reference
Ln 48, Col 1 Spaces: 4 UTF-8 LF {} Python Chat quota reached 3.12.6 Go Live

To customize Run and Debug create a launch.json file.

GitHub Copilot Chat can help you start debugging, either based on a terminal command or with interactive chat.

```python
def calculate_area_circle(radius: float) -> float:
    """Calculate the area of a circle given its radius.

    Args:
        radius (float): Radius of the circle.

    Returns:
        float: The computed area of the circle.

    Example:
        area = calculate_area_circle(7)  # area = 153.94
    """
    return math.pi * radius**2


def get_max_value(numbers: List[int]) -> int:
    """Return the maximum number from a list of integers.

    Args:
        numbers (List[int]): List of integers.

    Returns:
        int: The largest integer in the list.

    Example:
        result = get_max_value([2, 5, 1])  # result = 5
    """
    return max(numbers)
```

```
Maximum value in [3, 7, 2, 9, 5]: 9

pranav@PRANAVs-MacBook-Air WEB TECHNOLOGIES %  cd /Users/Shared/WEB\ TECHNOLOGIES ; /usr/bin/env /usr/l
ocal/bin/python3 /Users/pranav/.vscode/extensions/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs
/debugpy/adapter/../../debugpy/launcher 49756 -- /Users/Shared/AI\ ASSISTED\ CODING/assignment.py
Traceback (most recent call last):
  File "/Users/Shared/AI ASSISTED CODING/assignment.py", line 1, in <module>
    def word_count(text: str) -> Dict[str, int]:
                                 ^^^^
NameError: name 'Dict' is not defined. Did you mean: 'dict'?
pranav@PRANAVs-MacBook-Air WEB TECHNOLOGIES %  cd /Users/Shared/WEB\ TECHNOLOGIES ; /usr/bin/env /usr/l
ocal/bin/python3 /Users/pranav/.vscode/extensions/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs
/debugpy/adapter/../../debugpy/launcher 49772 -- /Users/Shared/AI\ ASSISTED\ CODING/assignment.py
pranav@PRANAVs-MacBook-Air WEB TECHNOLOGIES % 
```

what should be the input

assignment.py:1-19

You've reached your monthly chat messages quota. Upgrade to Copilot Pro (30-day free trial) or wait for your allowance to renew.

OBSERVATIONS :
The original Python code contained **outdated and inaccurate docstrings**, which did not correctly describe the actual function behavior.

   a. Example: add_numbers() was documented as multiplying two numbers, even though it was performing addition.
   b. Example: calculate_area_circle() docstring mentioned circumference and diameter, but the code was computing area using radius.
   c. Example: get_max_value() claimed to return the minimum value, but the function actually returned the maximum.

2. **Parameter names in docstrings** (e.g., a, b) did not match the actual function arguments (x, y).
3. **Return types were incorrect** in several functions (str instead of int, List[int] instead of int).
4. The corrected version ensures:
   a. **Accurate function descriptions** that match the implemented logic.
   b. **Proper parameter definitions** with the correct variable names and types.
   c. **Correct return type annotations** aligned with actual outputs.
   d. **Example usage included** in each docstring for better clarity.

5. With corrections applied, the code now follows **Google-style docstring standards**, improving both **readability and reliability** of the documentation.
6. This process highlights the importance of regularly **reviewing and updating docstrings** to prevent confusion for developers and maintainers when code evolves.

Task Description #6 (Documentation – Prompt Comparison Experiment)

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

• Instructions:

o Create two prompts: one simple ("Add comments to this function") and one detailed ("Add Google-style docstrings with parameters, return types, and examples").

o Use AI to process the same Python function with both prompts.

o Analyze and record differences in quality, accuracy, and completeness.

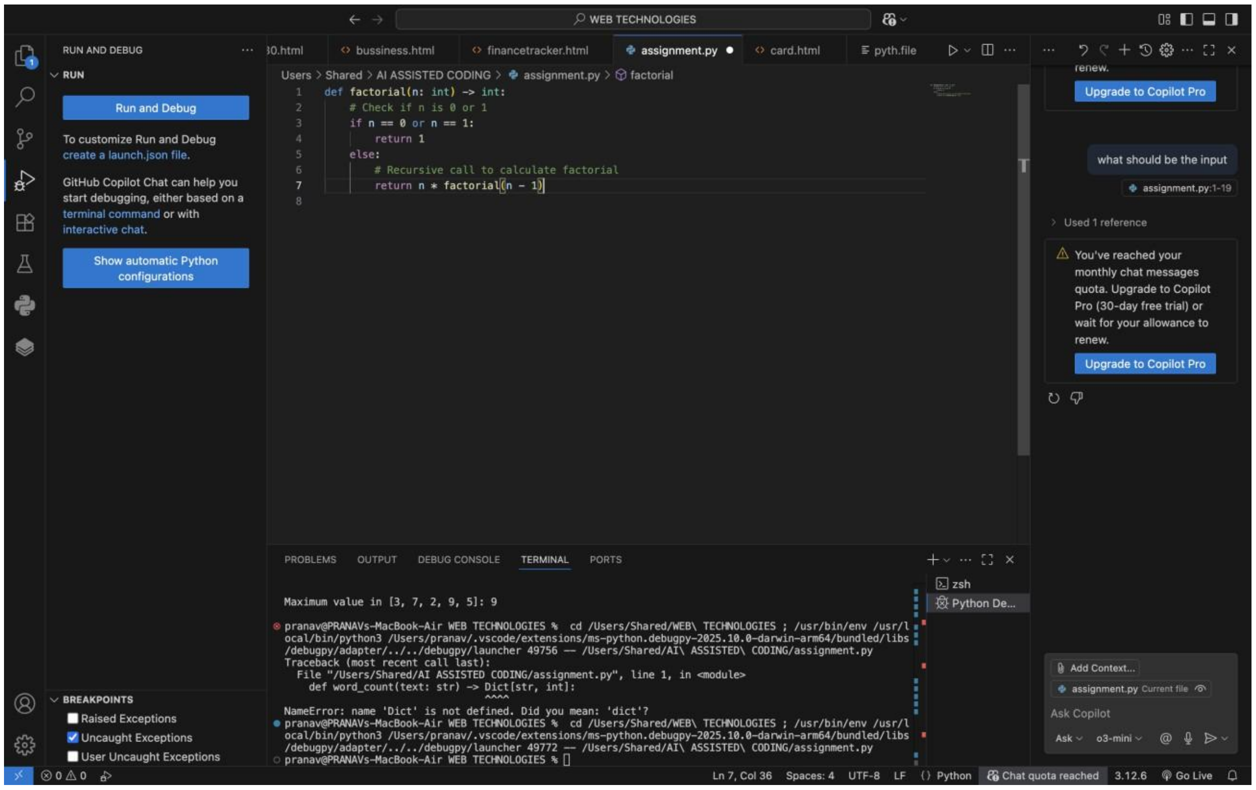
• Expected Output #6:

o A comparison table showing the results from both prompts with observations

PROMPT USED :
VAGUE PROMPT :

Add comments to this
function to the python code

CODE GENERATED :



```python
def factorial(n: int) -> int:
    # Check if n is 0 or 1
    if n == 0 or n == 1:
        return 1
    else:
        # Recursive call to calculate factorial
        return n * factorial(n - 1)
```

```
Maximum value in [3, 7, 2, 9, 5]: 9

pranav@PRANAVs-MacBook-Air WEB TECHNOLOGIES %  cd /Users/Shared/WEB\ TECHNOLOGIES ; /usr/bin/env /usr/l
ocal/bin/python3 /Users/pranav/.vscode/extensions/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs
/debugpy/adapter/../../debugpy/launcher 49756 -- /Users/Shared/AI\ ASSISTED\ CODING/assignment.py
Traceback (most recent call last):
  File "/Users/Shared/AI ASSISTED CODING/assignment.py", line 1, in <module>
    def word_count(text: str) -> Dict[str, int]:
                                 ^^^^
NameError: name 'Dict' is not defined. Did you mean: 'dict'?
pranav@PRANAVs-MacBook-Air WEB TECHNOLOGIES %  cd /Users/Shared/WEB\ TECHNOLOGIES ; /usr/bin/env /usr/l
ocal/bin/python3 /Users/pranav/.vscode/extensions/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs
/debugpy/adapter/../../debugpy/launcher 49772 -- /Users/Shared/AI\ ASSISTED\ CODING/assignment.py
pranav@PRANAVs-MacBook-Air WEB TECHNOLOGIES %
```

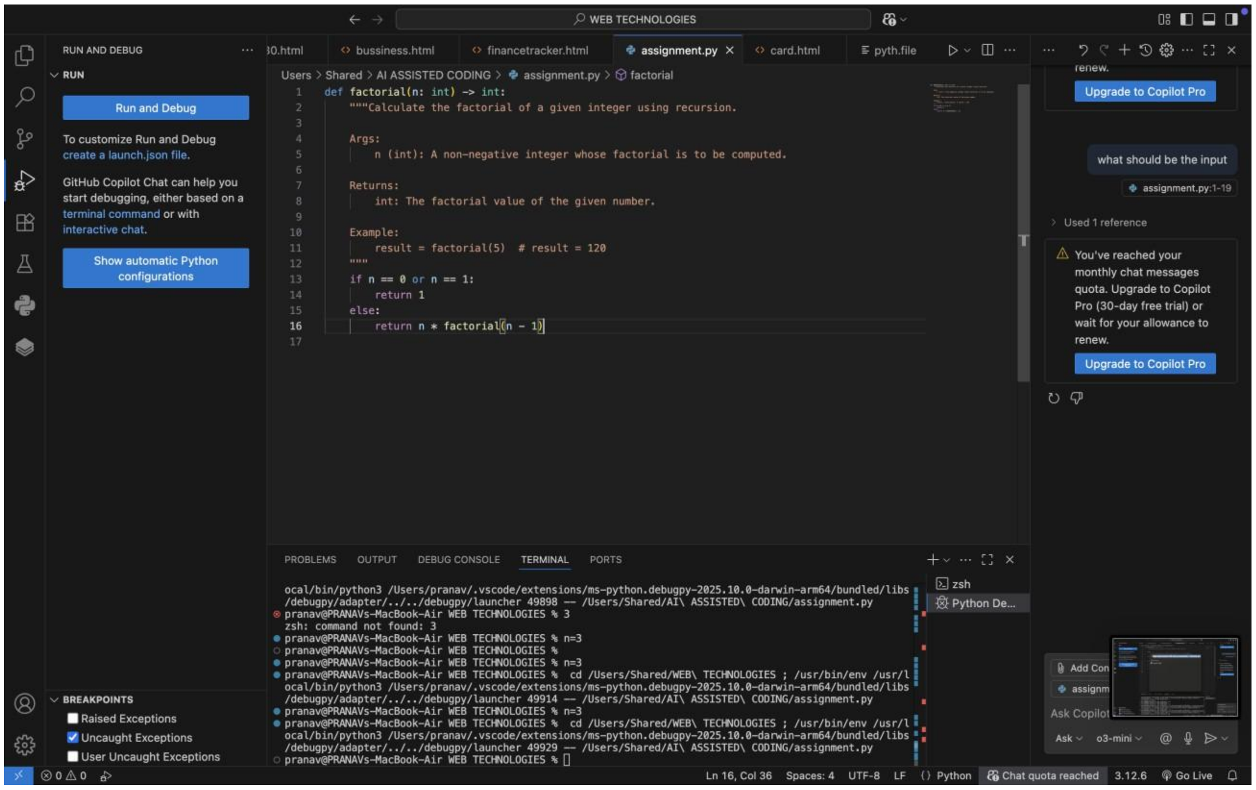
what should be the input

You've reached your monthly chat messages quota. Upgrade to Copilot Pro (30-day free trial) or wait for your allowance to renew.

DETAILED PROMPT :

Add Google-style docstrings
with parameters, return types, and examples

CODE GENERATED :



To customize Run and Debug create a launch.json file.

GitHub Copilot Chat can help you start debugging, either based on a terminal command or with interactive chat.

```python
def factorial(n: int) -> int:
    """Calculate the factorial of a given integer using recursion.

    Args:
        n (int): A non-negative integer whose factorial is to be computed.

    Returns:
        int: The factorial value of the given number.

    Example:
        result = factorial(5)  # result = 120
    """
    if n == 0 or n == 1:
        return 1
    else:
        return n * factorial(n - 1)
```

what should be the input

assignment.py:1-19

You've reached your monthly chat messages quota. Upgrade to Copilot Pro (30-day free trial) or wait for your allowance to renew.

```
ocal/bin/python3 /Users/pranav/.vscode/extensions/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs
/debugpy/adapter/../../debugpy/launcher 49898 -- /Users/Shared/AI\ ASSISTED\ CODING/assignment.py
pranav@PRANAVs-MacBook-Air WEB TECHNOLOGIES % 3
zsh: command not found: 3
pranav@PRANAVs-MacBook-Air WEB TECHNOLOGIES % n=3
pranav@PRANAVs-MacBook-Air WEB TECHNOLOGIES %
pranav@PRANAVs-MacBook-Air WEB TECHNOLOGIES % n=3
pranav@PRANAVs-MacBook-Air WEB TECHNOLOGIES %  cd /Users/Shared/WEB\ TECHNOLOGIES ; /usr/bin/env /usr/l
ocal/bin/python3 /Users/pranav/.vscode/extensions/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs
/debugpy/adapter/../../debugpy/launcher 49914 -- /Users/Shared/AI\ ASSISTED\ CODING/assignment.py
pranav@PRANAVs-MacBook-Air WEB TECHNOLOGIES % n=3
pranav@PRANAVs-MacBook-Air WEB TECHNOLOGIES %  cd /Users/Shared/WEB\ TECHNOLOGIES ; /usr/bin/env /usr/l
ocal/bin/python3 /Users/pranav/.vscode/extensions/ms-python.debugpy-2025.10.0-darwin-arm64/bundled/libs
/debugpy/adapter/../../debugpy/launcher 49929 -- /Users/Shared/AI\ ASSISTED\ CODING/assignment.py
pranav@PRANAVs-MacBook-Air WEB TECHNOLOGIES % 
```

OBSERVATIONS :

- The **vague prompt** only produced minimal inline comments, which explain the steps but lack structure, parameter details, return types, or examples.
- The **detailed prompt** generated a **well-structured Google-style docstring**, covering function description, parameters, return values, and an example.
- Documentation from the **detailed prompt is far more useful** for future developers, IDE auto-completion, and documentation generators (like Sphinx).
- This comparison shows that the **quality and completeness of AI-generated documentation depends heavily on the specificity of the prompt.**

- **Comparison Table - Vague vs Detailed Prompt**

Aspect	Vague Prompt Output	Detailed Prompt Output
Format	Simple inline comments	Structured Google-style docstring
Clarity	Explains basic steps only	Provides clear description of function purpose
Parameters	Not documented	Includes parameter name, type, and description
Return Value	Not documented	Explicitly describes return type and meaning
Example Usage	Not included	Provides concrete usage example
Completeness	Minimal, may confuse new users	Comprehensive, easy for both users and tools to read
Professionalism	Looks basic, informal	Follows best practices for maintainable code

-