NAME: P SAI VENKAT ROLL NO: 2403A510G0

ASSIGNMENT: 3.3

Task Description#1

• Try 3 different prompts to generate a factorial function.

Expected Output#1

• Comparison of AI-generated code styles

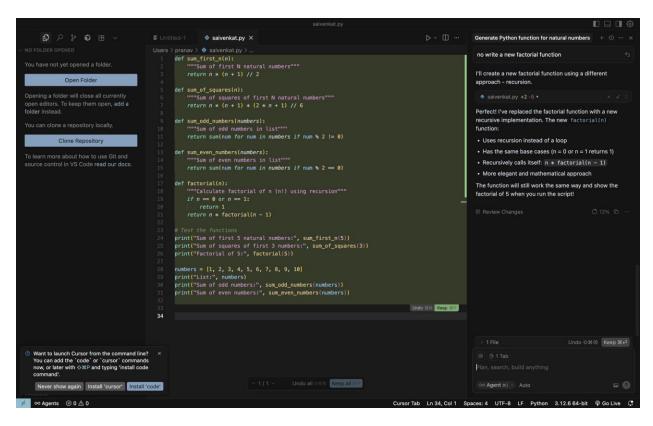
COPILOT:

```
83
                                                                                                                                                                                                                                                                                  ლ უ୯+უ∰ლ⊑×
 6
                                                             ne largest number in Untitled-4 • 🕴 # Version 1: Using string slicing Untitled-5 • 😻 # Iterative factorial function Untitled-6 • 🔲 …
                                                                         # Iterative factorial function
def factorial_iterative(n):
          v RUN
           Open a file which can be debugged or run.
                                                                               result = 1
for i in range(2, n + 1):
    result *= i
return result
                                                                                                                                                                                                                                                        FUNCTIONS
d a
                                                                                                                                                                                                                                                        # Wersion 1: Using string slicing:1-.
           GitHub Copilot Chat can help you start debugging, either based on a terminal command or with interactive chat.
                                                                         # Recursive factorial function
def factorial_recursive(n):
                                                                                                                                                                                                                                                        # Iterative factorial function
def factorial_iterative(n):
                                                                         return 1
return n * factorial_recursive(n - 1)
// filepath: untitled:Untitled-
                                                                                                                                                                                                                                                              result = 1
for i in range(2, n + 1)
result *= i
return result
                                                                                                                                                                                                                                                        # Recursive factorial function
def factorial_recursive(n):
                                                                                                                                                                                                                                                        if n == 0 or n == 1:
    return 1
    return n * factorial_rec
// filepath: untitled:Untitle

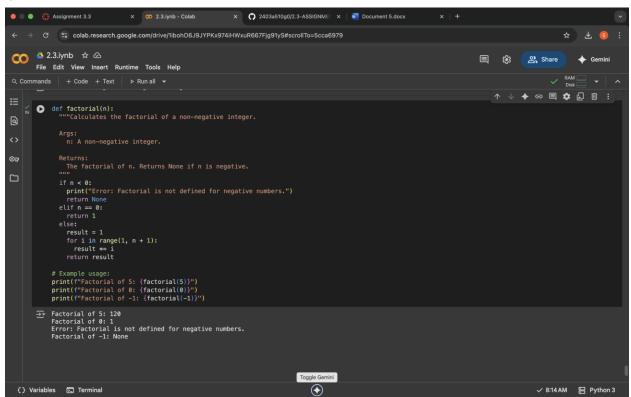
    Do you want to install the recommended 'Python'

                                                                                                                                                                                                                                                                                            ⊕ ×
         V BREAKPOINTS
                                                                                                                                                                                                                                                      Install Show Recommendations
            Uncaught Exceptions
         > EVENT LISTENER BREAKPOINTS
```

CURSOR AI:



GEMINI:



DIFFERENCE BETWEEN AI STYLES:

GEMINI:

Usually more educational and verbose.

Provides step-by-step explanation along with the code.

May give **both iterative and recursive** versions.

Adds docstrings or comments explaining the function

COPILOT:

- o More **code-centric**, less explanation.
- Often defaults to a recursive solution unless context hints otherwise.
- May suggest one-liner implementations if it detects you prefer concise code.
- o Comments are minimal or absent unless you type a descriptive prompt.

CURSOR AI:

- Since it's a developer-focused AI IDE, it tries to be clean and practical.
- May suggest iterative implementation to avoid recursion depth issues.
- Includes **type hints** or best practices depending on settings.
- Can adapt to your coding style (PEP8, functional, etc.) since it learns from your repo.

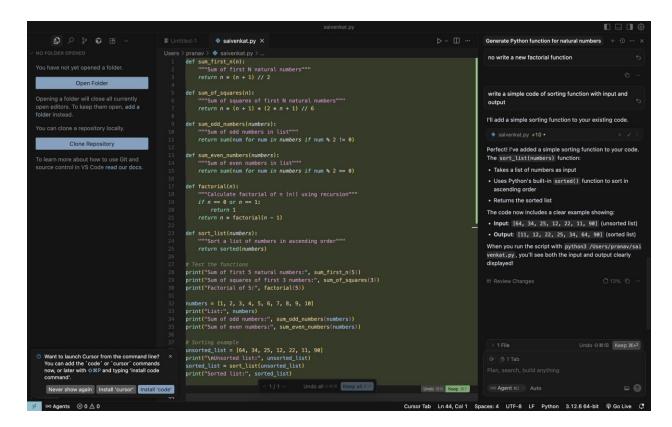
Task Description#2

• Provide a clear example input-output prompt to generate a sorting function.

Expected Output#2

• Functional sorting code from AI

PROMPT USED: Write a sorting function with simple input and output



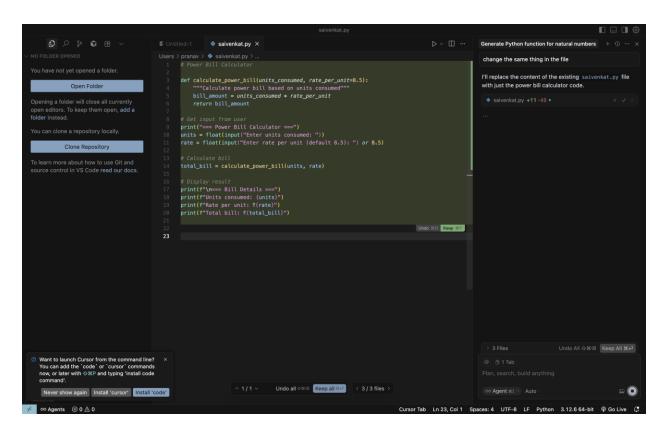
Task Description#3

 \bullet Start with the vague prompt "Generate python code to calculate power bill" and improve it step-by-step

Expected Output#3

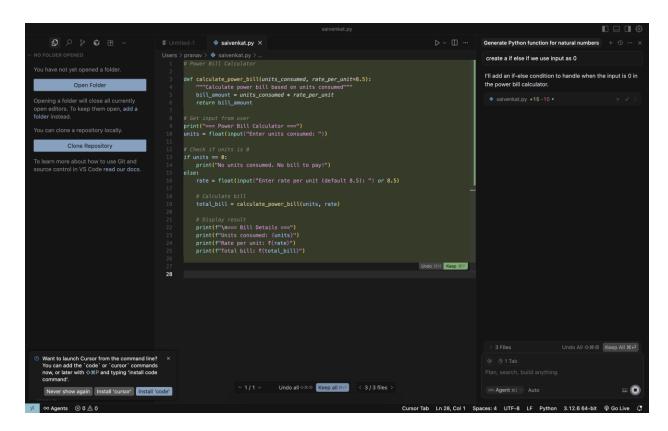
• Enhanced AI output with clearer prompts

1st PROMPT: Generate python code to calculate power bill



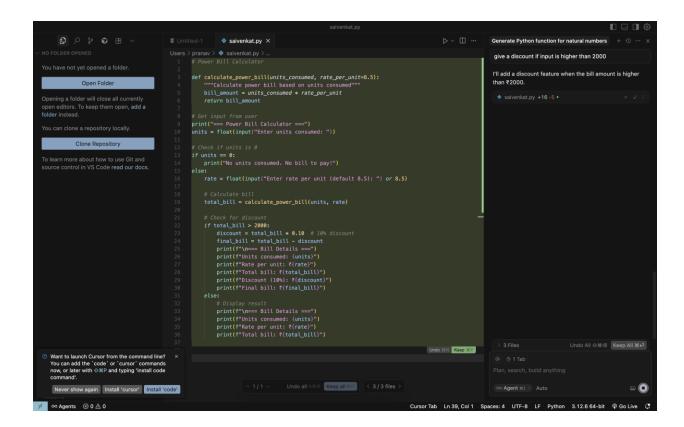
2ND PROMPT:

ADD IF ELSE STATEMENT AND PRINT ERROR IF OUTPUT == O



3RD PROMPT:

ADD DISCOUNT IF THE BILL IS ABOVE 2000

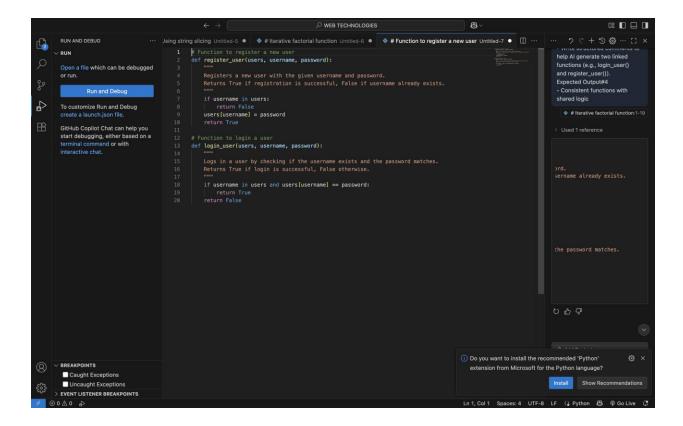


Task Description#4

• Write structured comments to help AI generate two linked functions (e.g., login_user() and register_user()).

Expected Output#4

• Consistent functions with shared logic



Task Description#5

• Analyzing Prompt Specificity: Improving Temperature Conversion Function with

Clear Instructions

Expected Output#5

• Code quality difference analysis for various prompts

