

AI ASSISTANT CODING

ASSIGNMENT - 2

Name : L. Sharan Sai

Varshith

HT.NO: 2303A51450

BATCH: 21

LAB 2:

**Exploring Additional AI Coding Tools beyond Copilot – Gemini (Colab)
and Cursor AI**

Task 1: Cleaning Sensor Data

❖ Scenario:

❖ You are cleaning IoT sensor data where negative values are invalid.

❖ Task:

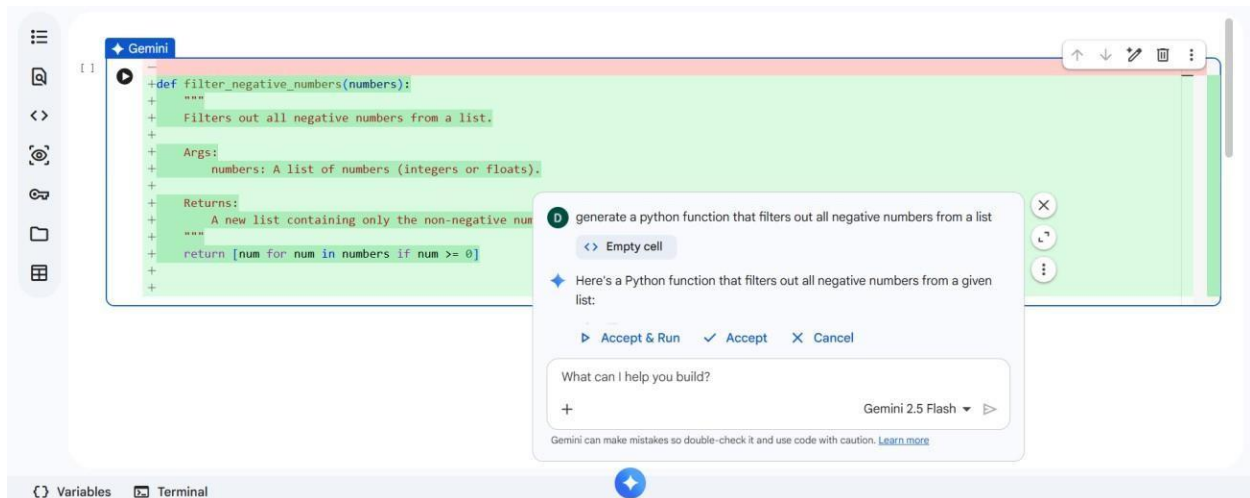
Use Gemini in Colab to generate a function that filters out all negative numbers from a list.

❖ Expected Output:

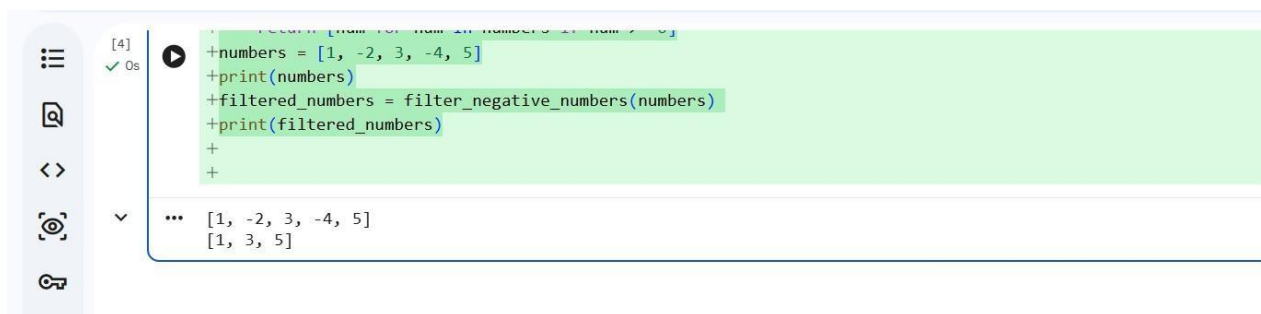
➤ Before/after list

➤ Screenshot of Colab execution

CODE :



OUTPUT:



Task 2: String Character Analysis

❖ Scenario:

You are building a text-analysis feature.

❖ Task:

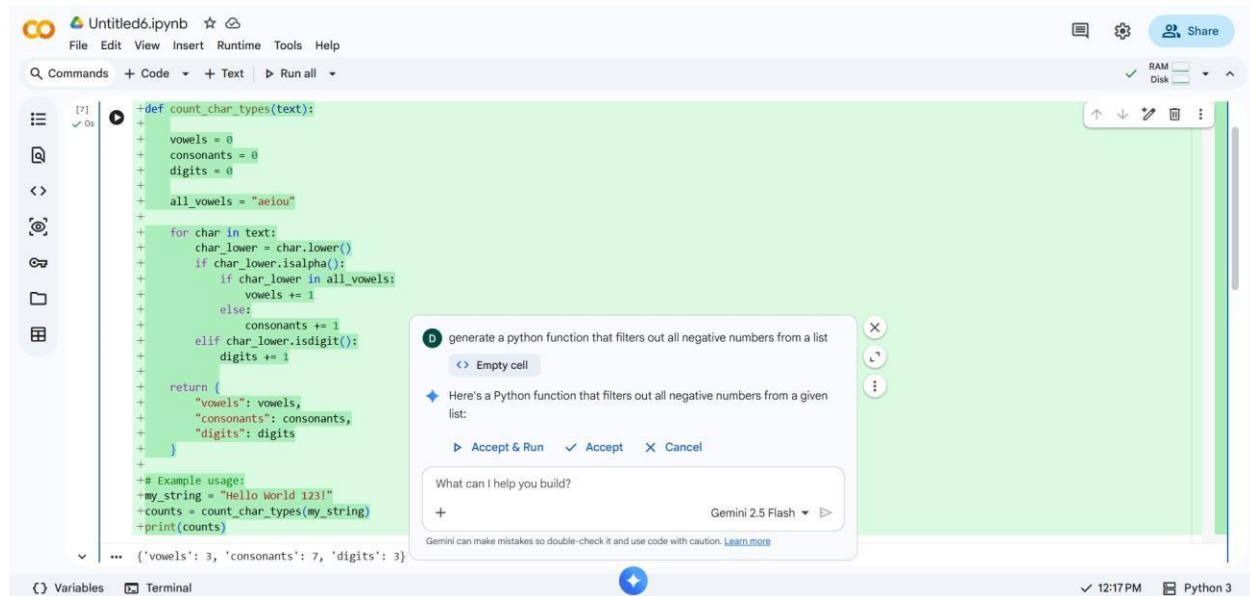
Use Gemini to generate a Python function that counts vowels, consonants, and digits in a string.

❖ Expected Output:

➤ Working function

➤ Sample inputs and outputs

CODE :

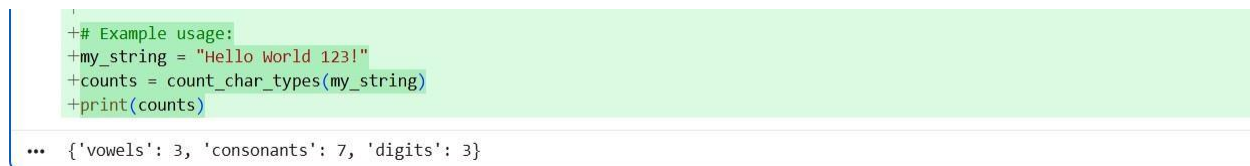


The screenshot shows a Jupyter Notebook titled 'Untitled6.ipynb'. The code in the cell defines a function `count_char_types(text)` that counts vowels, consonants, and digits in a string. The function initializes counters, iterates through the string, and returns a dictionary of counts. An example usage is provided: `my_string = "Hello world 123!"`, `counts = count_char_types(my_string)`, and `print(counts)`. The output of the cell is `{'vowels': 3, 'consonants': 7, 'digits': 3}`. A Gemini AI chat window is open over the code, showing a prompt to generate a function for filtering negative numbers and a response providing a Python function for that purpose.

```
[?] Untitled6.ipynb ☆ ☁
File Edit View Insert Runtime Tools Help
Q Commands + Code + Text ▶ Run all
RAM Disk
[?] 0s
def count_char_types(text):
    vowels = 0
    consonants = 0
    digits = 0
    all_vowels = "aeiou"
    for char in text:
        char_lower = char.lower()
        if char_lower.isalpha():
            if char_lower in all_vowels:
                vowels += 1
            else:
                consonants += 1
        elif char_lower.isdigit():
            digits += 1
    return {
        "vowels": vowels,
        "consonants": consonants,
        "digits": digits
    }
    ... {'vowels': 3, 'consonants': 7, 'digits': 3}

D generate a python function that filters out all negative numbers from a list
<> Empty cell
Here's a Python function that filters out all negative numbers from a given list:
Accept & Run Accept Cancel
What can I help you build?
+ Gemini 2.5 Flash ▶
Gemini can make mistakes so double-check it and use code with caution. Learn more
```

OUTPUT:



The screenshot shows the output of the Python code, which is a dictionary containing the counts for vowels, consonants, and digits. The output is `{'vowels': 3, 'consonants': 7, 'digits': 3}`.

```
... {'vowels': 3, 'consonants': 7, 'digits': 3}
```

Task 3: Palindrome Check – Tool Comparison

❖ Scenario:

You must decide which AI tool is clearer for string logic.

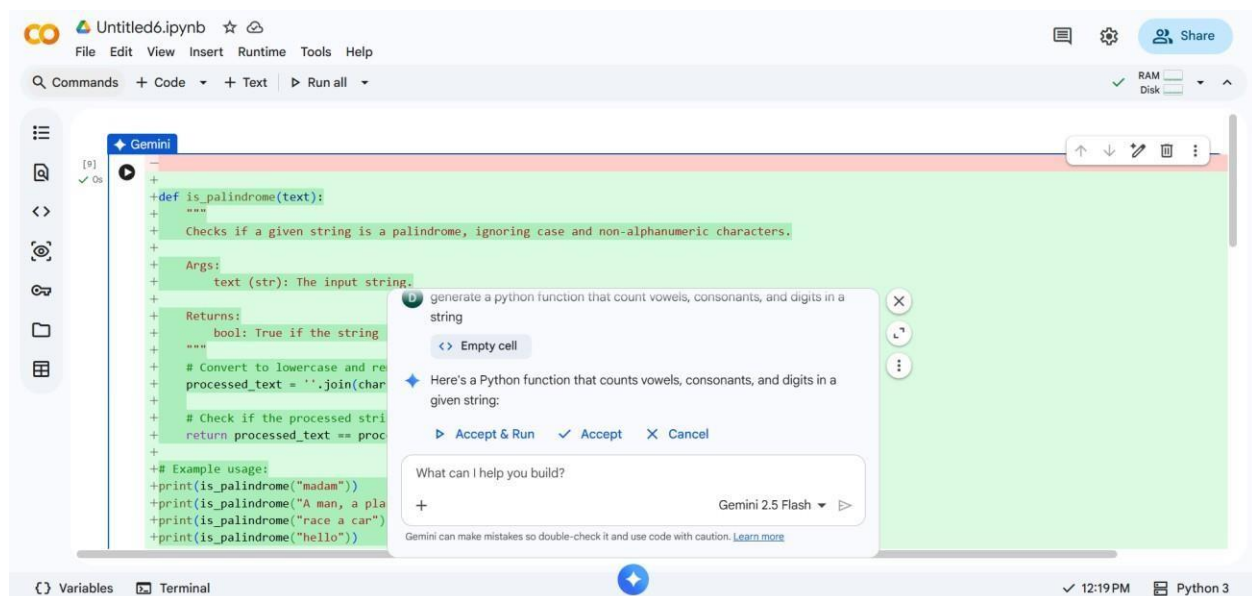
❖ Task:

Generate a palindrome-checking function using Gemini and Copilot, then compare the results.

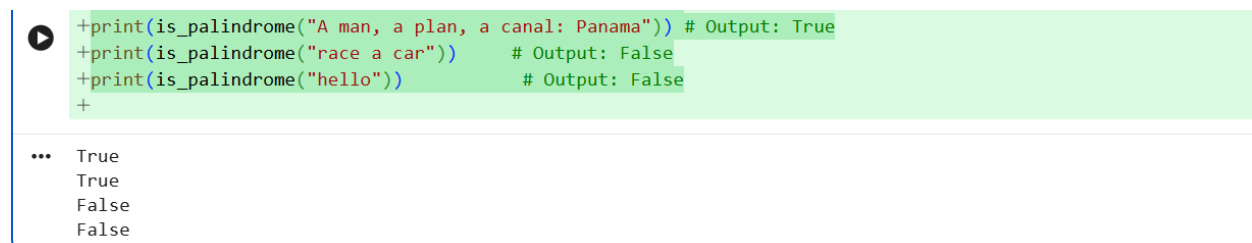
❖ Expected Output:

- Side-by-side code comparison
- Observations on clarity and structure

CODE:



OUTPUT:



Task 4: Code Explanation Using AI

❖ Scenario:

You are reviewing unfamiliar code written by another developer.

❖ Task:

Ask Gemini to explain a Python function (prime check OR palindrome check) line by line.

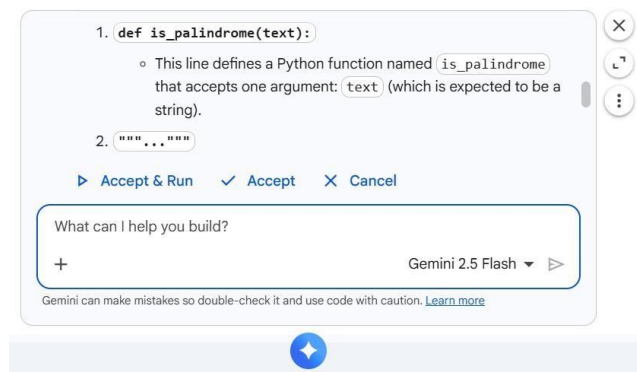
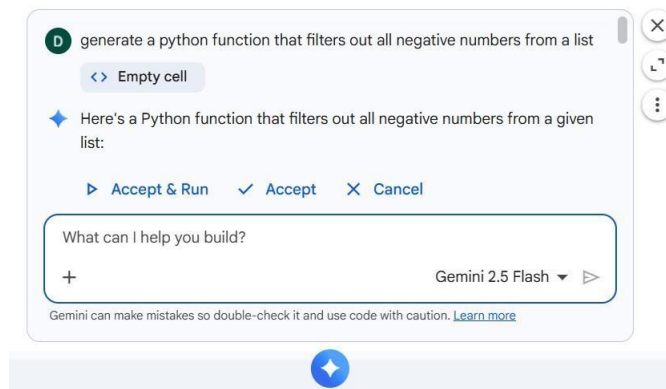
❖ Expected Output:

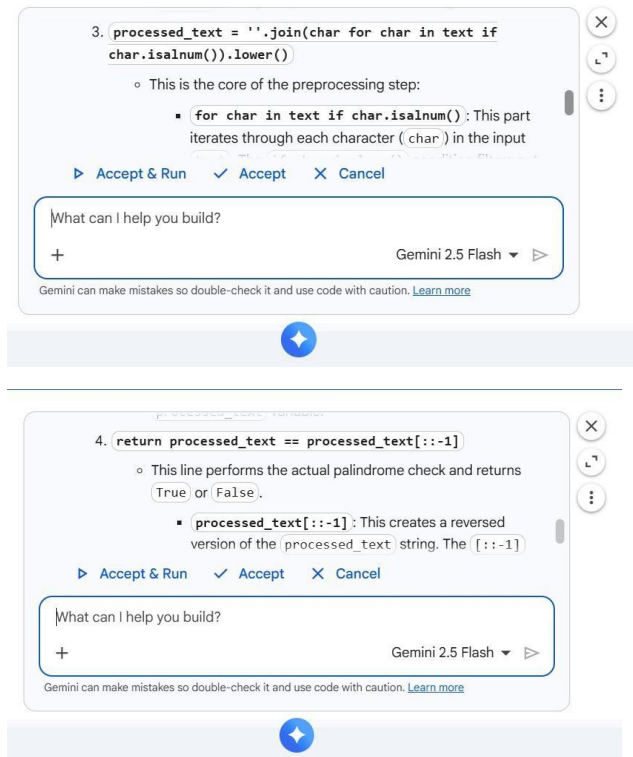
➤ Code snippet

➤ AI explanation

➤ Student comments on understanding

OUTPUT:





My own experience using both Gemini and GitHub Copilot:

While using Gemini in Google Colab, I found the explanations to be very clear and helpful in understanding the logic behind the code. Gemini was especially useful for learning and analyzing Python programs step by step. GitHub Copilot, was faster in generating code directly inside the editor and helped me complete tasks quickly. Copilot felt more suitable for continuous coding, while Gemini was better for conceptual clarity. Overall, using both tools together improved my coding efficiency and understanding.