AI ASSISTANT CODING ASSIGNMENT - 2

NAME: K.LAXMAN

HT.NO: 2303A51428

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:**

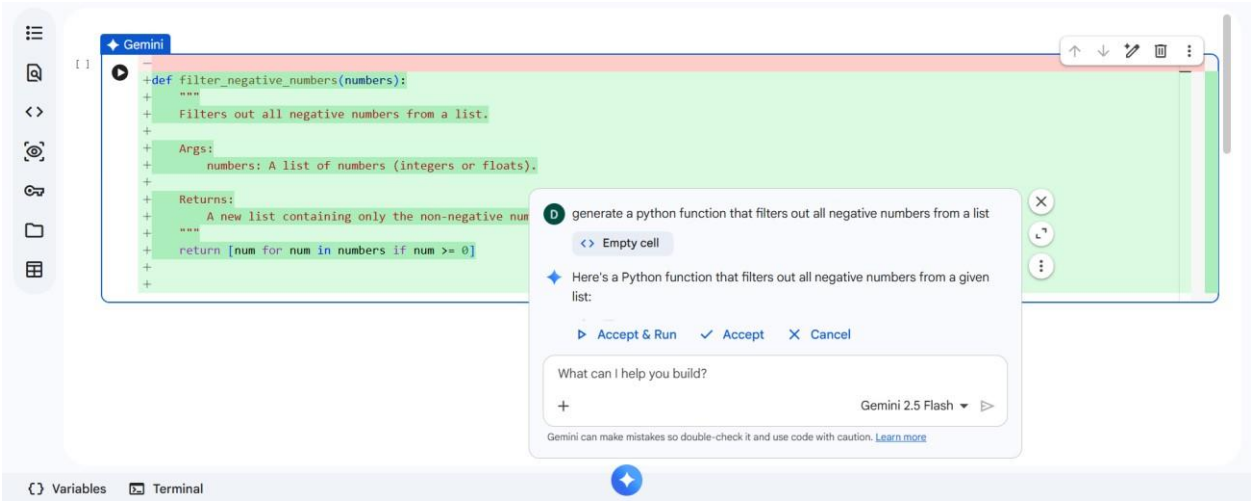
Use Gemini in Colab to generate a function that filters out all negative numbers from a list.

❖ **Expected Output:**

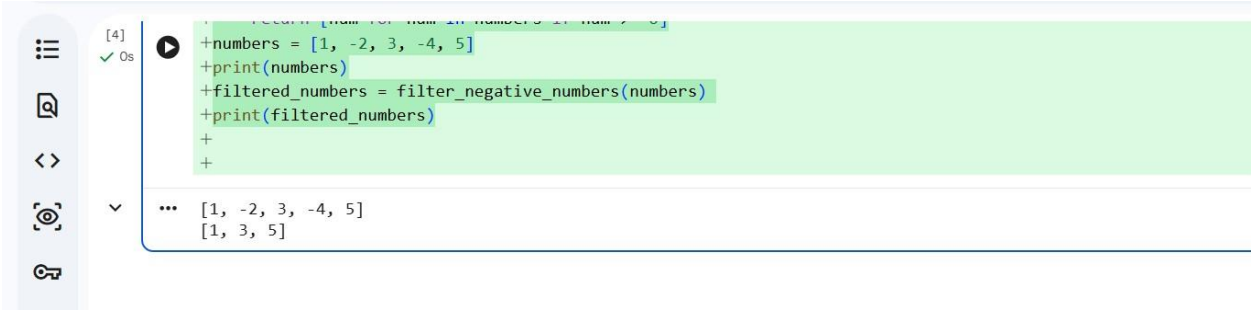
➢ **Before/after list**

➢ **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,

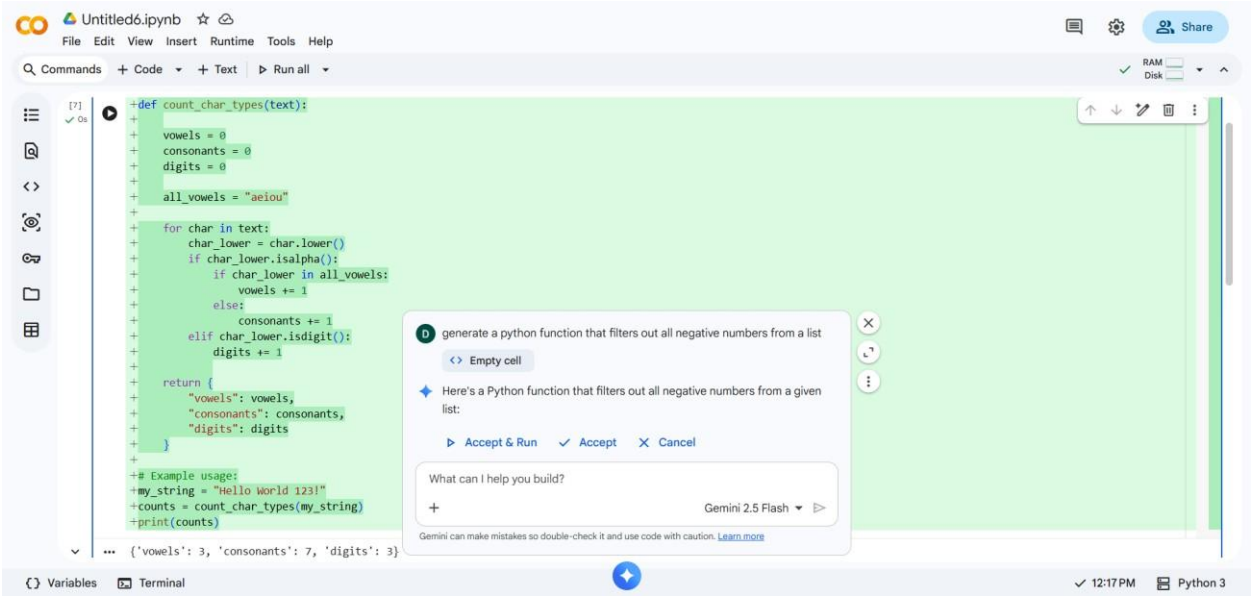
and digits in a string.

❖ **Expected Output:**

➢ **Working function**

➢ **Sample inputs and outputs**

CODE :



OUTPUT:

```
+# Example usage:
+my_string = "Hello World 123!"
+counts = count_char_types(my_string)
+print(counts)
```

```
{'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:**

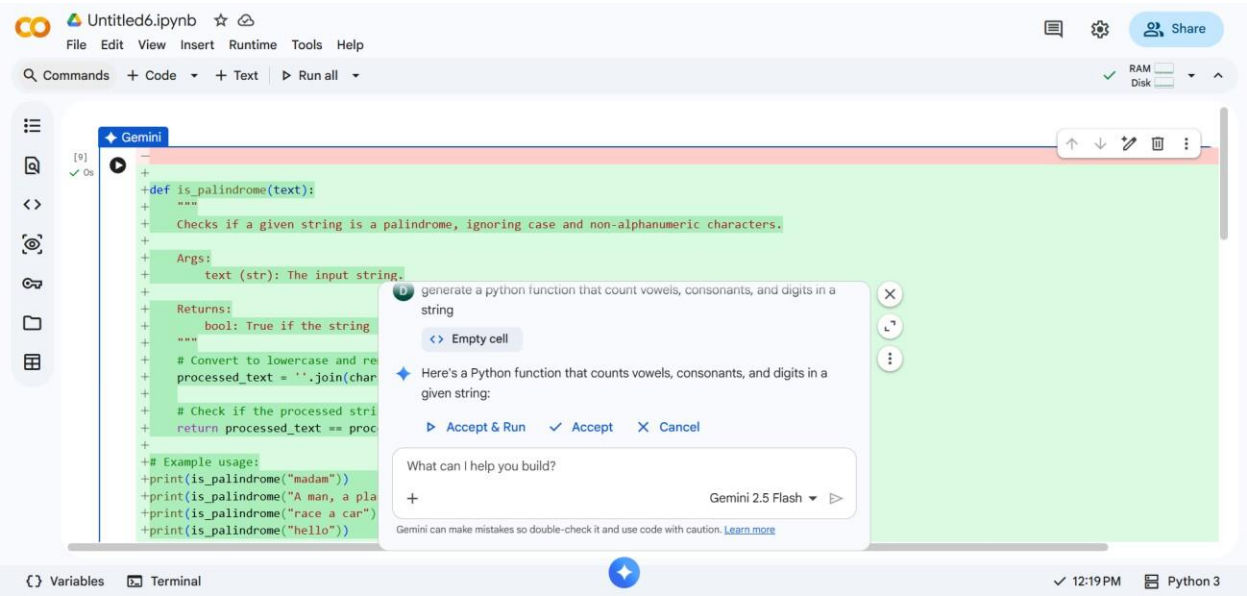
Generate a palindrome-checking function using Gemini and Copilot, then compare the results.

❖ Expected Output:

➢ Side-by-side code comparison

➢ Observations on clarity and structure

CODE:



OUTPUT:

```
+print(is_palindrome("A man, a plan, a canal: Panama")) # Output: True
+print(is_palindrome("race a car"))     # Output: False
+print(is_palindrome("hello"))          # Output: False
+
```

```
True
True
False
False
```

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.

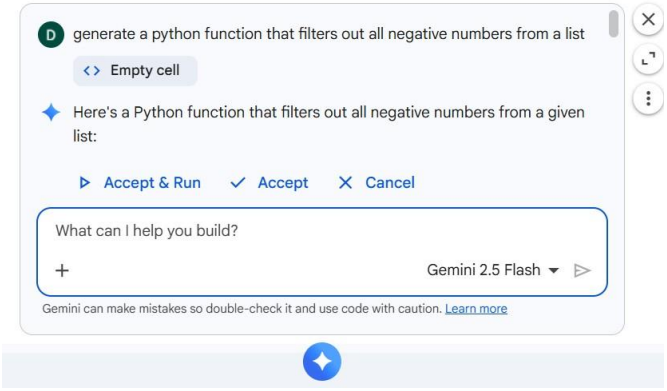
❖ **Expected Output:**

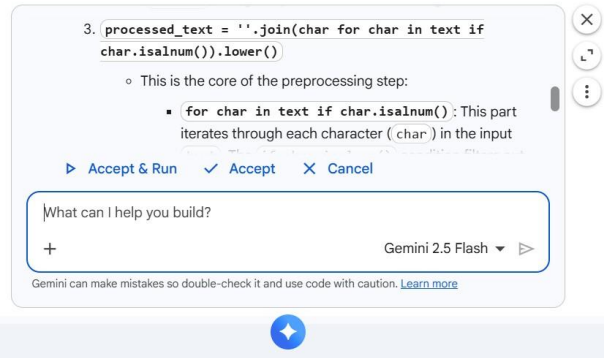
➢ **Code snippet**

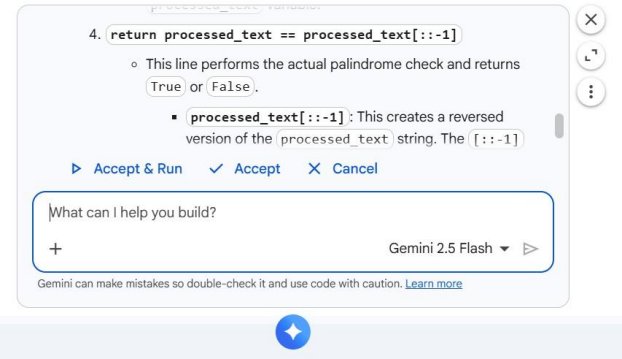
➢ **AI explanation**

➢ **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.