AI ASSISTANT CODING ASSIGNMENT 2

Name: Kommu Madhupriya

Hall ticket: 2303A51583

Batch:22

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

and Cursor AI

Lab Objectives:

❖ To explore and evaluate the functionality of Google Gemini for AI-

Week1 -

Monday

assisted coding within Google Colab.

❖ To understand and use Cursor AI for code generation, explanation,

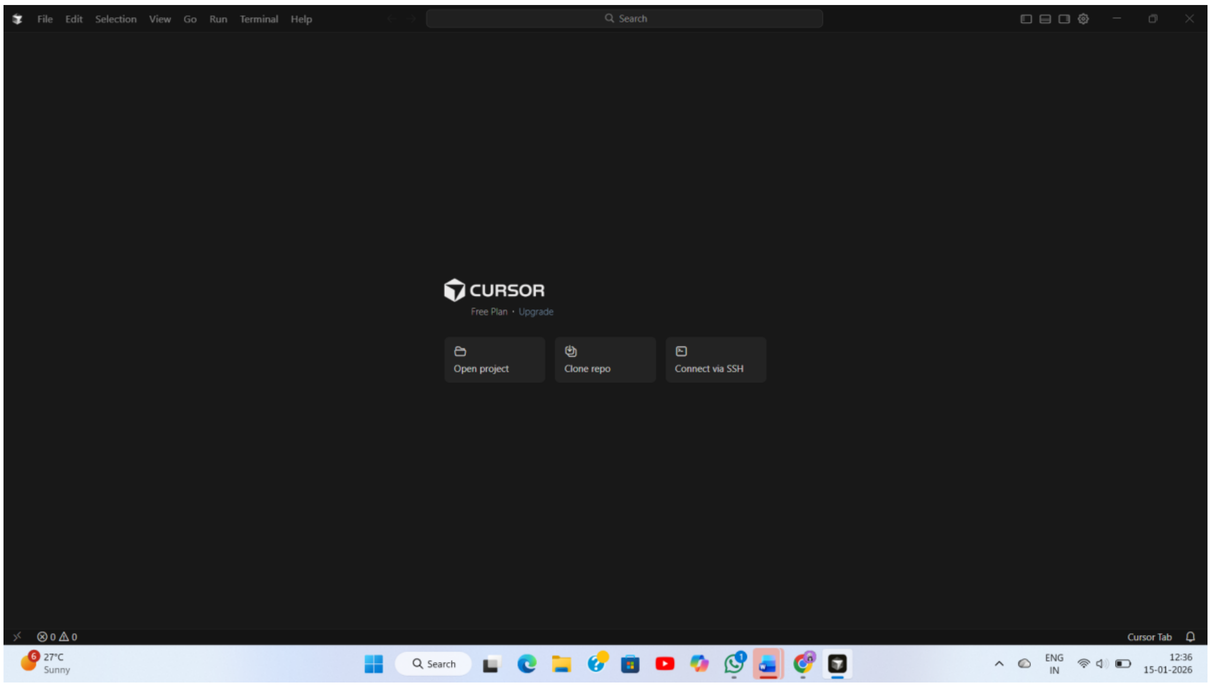
and refactoring.

❖ To compare outputs and usability between Gemini, GitHub Copilot,

and Cursor AI.

❖ To perform code optimization and documentation using AI tools.

Lab Outcomes (LOs):



After completing this lab, students will be able to:

❖ Generate Python code using Google Gemini in Google Colab.

❖ Analyze the effectiveness of code explanations and suggestions by Gemini.

❖ Set up and use Cursor AI for AI-powered coding assistance.

❖ Evaluate and refactor code using Cursor AI features.

❖ Compare AI tool behavior and code quality across different platform.

Task 1: Cleaning Sensor Data

❖ Scenario:

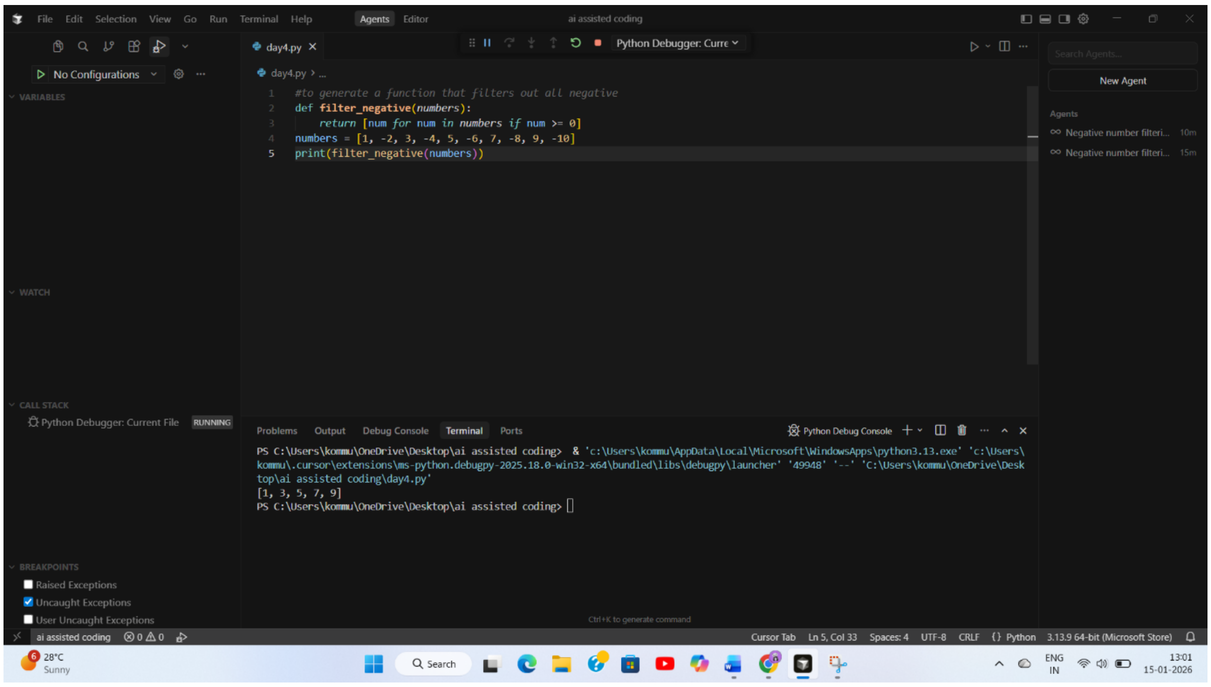
❖ You are cleaning IoT sensor data where negative values are invalid.

❖ Task:

Use Gemini in Colab to generate a function that filters out all negative

numbers from a list.

Code/output:


```python
#to generate a function that filters out all negative
def filter_negative(numbers):
    return [num for num in numbers if num >= 0]
numbers = [1, -2, 3, -4, 5, -6, 7, -8, 9, -10]
print(filter_negative(numbers))
```

```
PS C:\Users\kommu\OneDrive\Desktop\ai assisted coding>  & 'c:\Users\kommu\AppData\Local\Microsoft\WindowsApps\python3.13.exe' 'c:\Users\
kommu\.cursor\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '49948' '--' 'C:\Users\kommu\OneDrive\Desk
top\ai assisted coding\day4.py'
[1, 3, 5, 7, 9]
PS C:\Users\kommu\OneDrive\Desktop\ai assisted coding> 
```

➢ Before/after list

➢ Screenshot of Colab execution.

Task 2: String Character Analysis

❖ **Scenario:**

You are building a text-analysis feature.

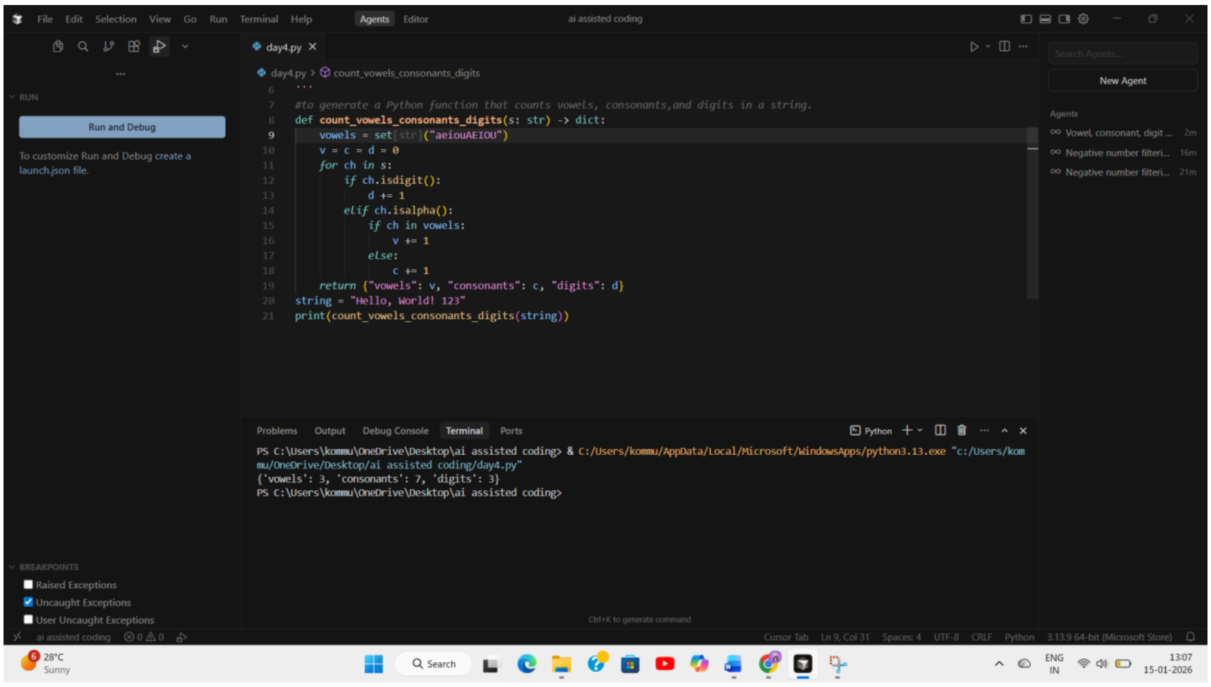
❖ Task:

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

❖ Expected Output:

➢ Working function

➢ Sample inputs and outputs.



```python
'''
#to generate a Python function that counts vowels, consonants,and digits in a string.
def count_vowels_consonants_digits(s: str) -> dict:
    vowels = set[str]("aeiouAEIOU")
    v = c = d = 0
    for ch in s:
        if ch.isdigit():
            d += 1
        elif ch.isalpha():
            if ch in vowels:
                v += 1
            else:
                c += 1
    return {"vowels": v, "consonants": c, "digits": d}
string = "Hello, World! 123"
print(count_vowels_consonants_digits(string))
```

```
PS C:\Users\kommu\OneDrive\Desktop\ai assisted coding> & C:/Users/kommu/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/kom
mu/OneDrive/Desktop/ai assisted coding/day4.py"
{'vowels': 3, 'consonants': 7, 'digits': 3}
PS C:\Users\kommu\OneDrive\Desktop\ai assisted coding>
```

File Edit Selection View Go Run Terminal Help
Agents Editor
ai assisted coding
day4.py
day4.py > count_vowels_consonants_digits
RUN
Run and Debug
To customize Run and Debug create a launch.json file.
BREAKPOINTS
Raised Exceptions
Uncaught Exceptions
User Uncaught Exceptions
Problems Output Debug Console Terminal Ports
New Agent
Agents
Vowel, consonant, digit ... 2m
Negative number filteri... 16m
Negative number filteri... 21m
Ctrl+K to generate command
Cursor Tab Ln 9, Col 31 Spaces: 4 UTF-8 CRLF Python 3.13.9 64-bit (Microsoft Store)
28°C Sunny
Search
ENG IN
13:07 15-01-2026

Task 3: Palindrome Check – Tool Comparison

❖ **Scenario:**

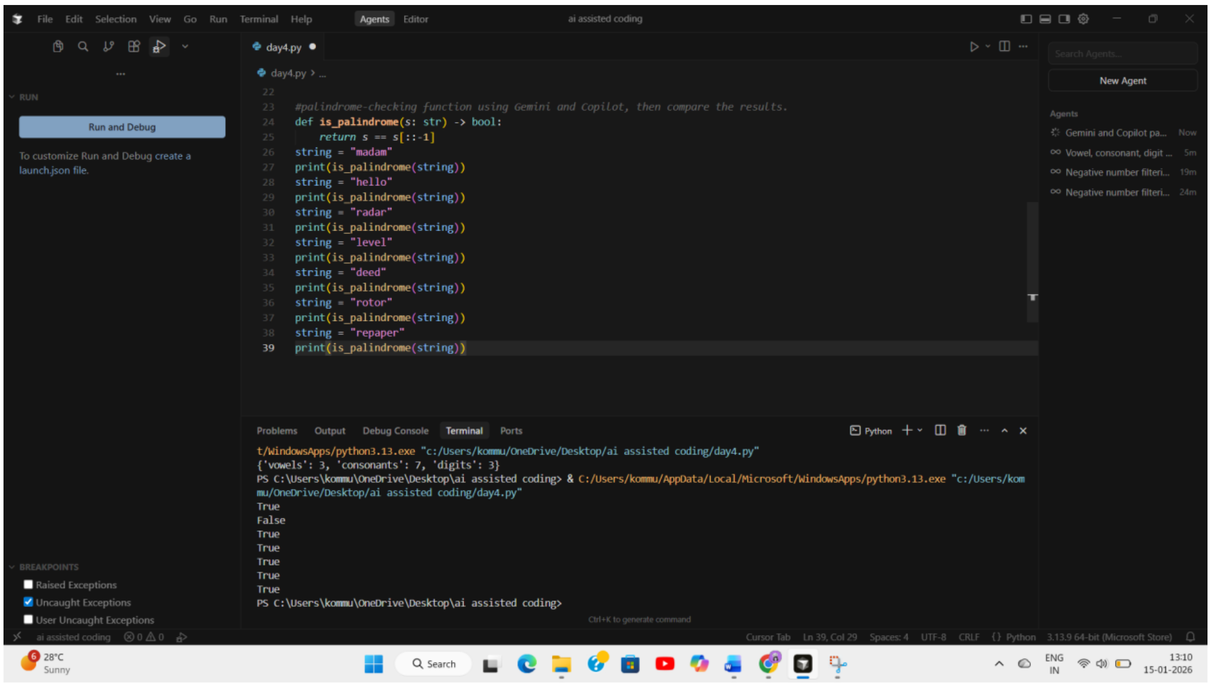
You must decide which AI tool is clearer for string logic.

❖ Task:

Generate a palindrome-checking function using Gemini and Copilot, then

compare the results.

❖ Expected Output:



```python
#palindrome-checking function using Gemini and Copilot, then compare the results.
def is_palindrome(s: str) -> bool:
    return s == s[::-1]
string = "madam"
print(is_palindrome(string))
string = "hello"
print(is_palindrome(string))
string = "radar"
print(is_palindrome(string))
string = "level"
print(is_palindrome(string))
string = "deed"
print(is_palindrome(string))
string = "rotor"
print(is_palindrome(string))
string = "repaper"
print(is_palindrome(string))
```

```
t/WindowsApps/python3.13.exe "c:/Users/kommu/OneDrive/Desktop/ai assisted coding/day4.py"
{'vowels': 3, 'consonants': 7, 'digits': 3}
PS C:\Users\kommu\OneDrive\Desktop\ai assisted coding> & C:/Users/kommu/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/kom
mu/OneDrive/Desktop/ai assisted coding/day4.py"
True
False
True
True
True
True
True
PS C:\Users\kommu\OneDrive\Desktop\ai assisted coding>
```

➢ Side-by-side code comparison

➢ Observations on clarity and structure

Task 4: Code Explanation Using AI

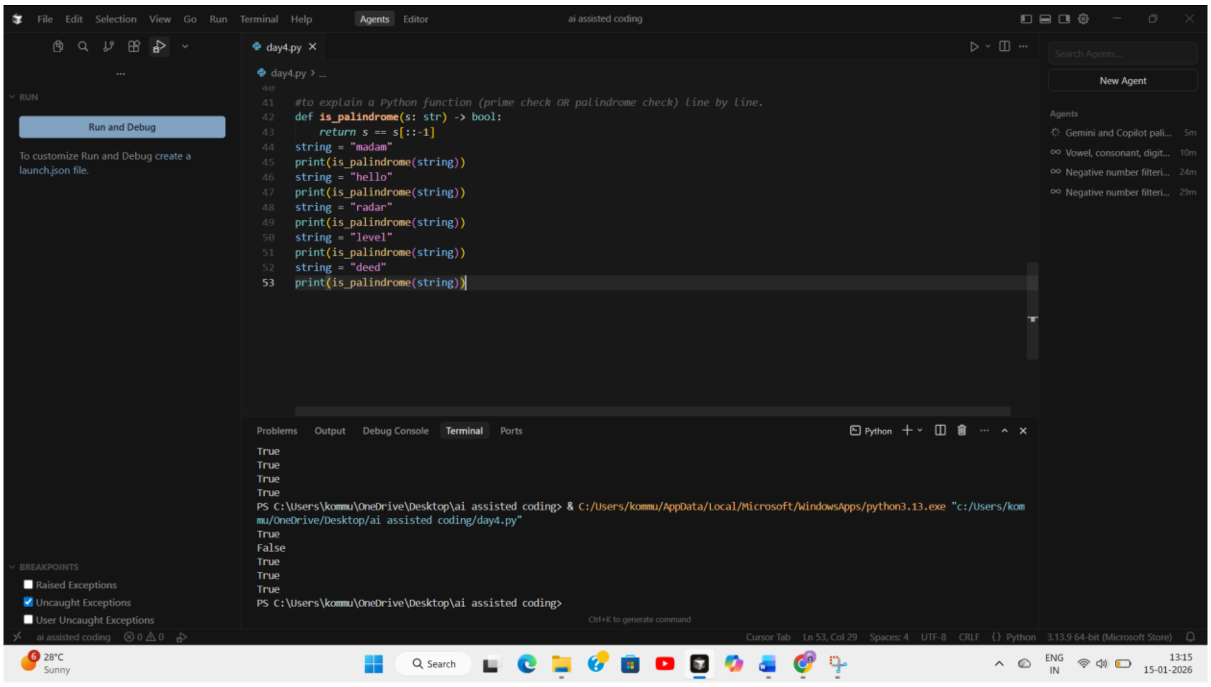
❖ **Scenario:**

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:

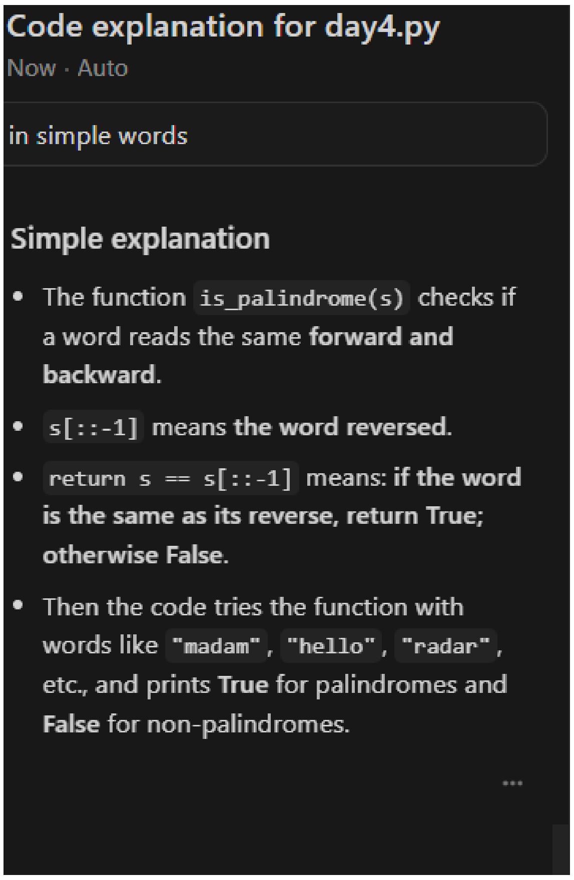


```python
#to explain a Python function (prime check OR palindrome check) line by line.
def is_palindrome(s: str) -> bool:
    return s == s[::-1]
string = "madam"
print(is_palindrome(string))
string = "hello"
print(is_palindrome(string))
string = "radar"
print(is_palindrome(string))
string = "level"
print(is_palindrome(string))
string = "deed"
print(is_palindrome(string))
```

```
True
True
True
True
PS C:\Users\kommu\OneDrive\Desktop\ai assisted coding> & C:/Users/kommu/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/kom
mu/OneDrive/Desktop/ai assisted coding/day4.py"
True
False
True
True
True
PS C:\Users\kommu\OneDrive\Desktop\ai assisted coding>
```

➢ Code snippet

➢ **AI explanation**



Code explanation for day4.py

Now · Auto

in simple words

Simple explanation

- The function `is_palindrome(s)` checks if a word reads the same **forward and backward**.
- `s[::-1]` **means the word reversed.**
- `return s == s[::-1]` means: **if the word is the same as its reverse, return True; otherwise False.**
- Then the code tries the function with words like `"madam"`, `"hello"`, `"radar"`, etc., and prints **True** for palindromes and **False** for non-palindromes.

➢ Student comments on understanding

My own experience using **both Gemini and Cursor Ai and GitHub Copilot**:

While using Cursor Ai, I found the explanations to be very clear and helpful in understanding the logic behind the code. Cursor 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