
Lab Assignment # 2.2

Program	: B. Tech (CSE)
Specialization	:
Course Title	: AI Assisted coding
Course Code	:
Semester	II
Academic Session	: 2025-2026
Name of Student	: Kothapally Srikanth
Enrollment No.	: 2403A51L42
Batch No.	52
Date	:15-01-2026

Submission Starts here :

AI LAB – 2.2

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.

Code Snippet:

```
def remove_negative_values(data):  
    cleaned_data = []  
    for value in data:  
        if value >= 0:  
            cleaned_data.append(value)  
    return cleaned_data  
  
sensor_data = [12, -5, 7, -3, 25, 0, -10]  
print("Before Cleaning:", sensor_data)  
print("After Cleaning:", remove_negative_values(sensor_data))
```

Output:

```
Before Cleaning: [12, -5, 7, -3, 25, 0, -10]  
After Cleaning: [12, 7, 25, 0]
```

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.

Code Snippet:

```
def analyze_string(text):
    vowels = "aeiouAEIOU"
    vowel_count = 0
    consonant_count = 0
    digit_count = 0

    for char in text:
        if char.isdigit():
            digit_count += 1
        elif char.isalpha():
            if char in vowels:
                vowel_count += 1
            else:
                consonant_count += 1

    return vowel_count, consonant_count, digit_count

text = "Hello123World"
result = analyze_string(text)

print("Input String:", text)
print("Vowels:", result[0])
print("Consonants:", result[1])
print("Digits:", result[2])
```

[How can I install Python](#)

Output:

```
Input String: Hello123World
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.

Code Snippet:

```
def is_palindrome_copilot(text):
    text = text.lower()
    left = 0
    right = len(text) - 1

    while left < right:
        if text[left] != text[right]:
            return False
        left += 1
        right -= 1

    return True
```

```
def is_palindrome_gemini(text):
    text = text.lower().replace(" ", "")
    return text == text[::-1]
```

Output:

```
Input: "madam"
Output: True
```

Observations on clarity and structure:

- The **Gemini-generated code** is concise and uses Python slicing, making it easy to read and understand.
- The **Copilot-generated code** uses a loop-based approach, which clearly explains the logic step by step.
- Gemini's solution is more suitable for beginners due to its simplicity.
- Copilot's solution is better for understanding algorithmic thinking and logic flow.

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.

Code Snippet:

```
def is_palindrome(text):  
    text = text.lower()  
    reversed_text = text[::-1]  
    return text == reversed_text
```

AI Explanation:

- `def is_palindrome(text):`
This line defines a function named `is_palindrome` that accepts a string as input.
- `text = text.lower()`
Converts all characters in the string to lowercase to avoid case-sensitive mismatches.
- `reversed_text = text[::-1]`
Reverses the string using Python slicing.
- `return text == reversed_text`
Compares the original string with the reversed string and returns `True` if they are the same; otherwise, returns `False`.

Students Comments on understanding:

- The AI explanation is clear and easy to understand.
- I learned how Python slicing is used to reverse a string.
- The step-by-step explanation helped me understand palindrome logic.
- I can now confidently read and explain similar Python programs.