AI-Assisted Coding-Assignment- 2

HTNO:2303A52228
Batch:37

Lab 2: Exploring Additional AI Coding Tools beyond Copilot – Gemini (Colab)and Cursor AI

Task 1: Cleaning Sensor Data
Prompt: Generate a function that filters out all negative numbers from a list.
CODE AND OUTPUT:

```
def filter_negative_numbers_loop(input_list):
    filtered_list = []
    for num in input_list:
        if num >= 0:
            filtered_list.append(num)
    return filtered_list
original_list = [1, -2, 3, -4, 0, 5, -6]
print(f"Original list: {original_list}")
filtered_list_loop = filter_negative_numbers_loop(original_list)
print(f"Filtered list (non-negative numbers only, using loop): {filtered_list_loop}")
def filter_negative_numbers_comprehension(input_list):
    return [num for num in input_list if num >= 0]
filtered_list_comprehension = filter_negative_numbers_comprehension(original_list)
print(f"Filtered list (non-negative numbers only, using comprehension): {filtered_list_comprehension}")
```

```
Original list: [1, -2, 3, -4, 0, 5, -6]
Filtered list (non-negative numbers only, using loop): [1, 3, 0, 5]
Filtered list (non-negative numbers only, using comprehension): [1, 3, 0, 5]
```

Explanation: This program takes a list of numbers and removes all the negative ones. The first function checks each number one by one and keeps only the positive values and zero.The second function does the same job but in a shorter and smarter way using list comprehension.Both methods return a new list that contains only non-negative numbers.

Task 2: String Character Analysis
Prompt: Use Gemini to generate a Python function that counts vowels, consonants,and digits in a string.
CODE AND OUTPUT:

```
def analyze_user_string():
    def count_chars(input_string):
        vowels = "aeiouAEIOU"
        vowel_count = 0
        consonant_count = 0
        digit_count = 0
        for char in input_string:
            if char.isalpha():
                if char in vowels:
                    vowel_count += 1
                else:
                    consonant_count += 1
            elif char.isdigit():
                digit_count += 1
        return {
            'vowels': vowel_count,
            'consonants': consonant_count,
            'digits': digit_count
        }
    user_string = input("Enter a string to analyze: ")
    print(f"Analyzing string: '{user_string}'")
    analysis_result = count_chars(user_string)
    print("Analysis:")
    for key, value in analysis_result.items():
        print(f"  {key.capitalize()}: {value}")
analyze_user_string()
```

```
Enter a string to analyze: dwnambe1i2y4i14
Analyzing string: 'dwnambe1i2y4i14'
Analysis:
  Vowels: 4
  Consonants: 6
  Digits: 5
```

Explanation: This program asks you to type any string and then studies each character in it.Inside the program, it counts how many vowels, consonants, and digits the string contains.Letters are checked one by one—vowels go to the vowel count, other letters go to consonants, and numbers are counted as digits.Finally, it prints how many vowels, consonants, and digits were found in the string.

Task 3: Palindrome Check – Tool Comparison
Prompt: Generate a palindrome-checking function using Gemini and Copilot, then compare the results.
CODE AND OUTPUT:

```
def check_user_palindrome():
    user_input_string = input("Enter a string to check for palindrome: ")
    print(f"Analyzing: '{user_input_string}'")
    cleaned_s = ''.join(char.lower() for char in user_input_string if char.isalnum())
    if cleaned_s == cleaned_s[::-1]:
        print(f"'{user_input_string}' IS a palindrome.")
    else:
        print(f"'{user_input_string}' IS NOT a palindrome.")
check_user_palindrome()
```

```
Enter a string to check for palindrome: madam
Analyzing: 'madam'
'madam' IS a palindrome.
```

Explanation: This program asks you to enter a word or sentence and then checks if it reads the same backward.It first removes spaces and symbols and converts everything to lowercase so the check is fair.Then it compares the cleaned string with its reverse using slicing.If both match, it prints that it is a palindrome; otherwise, it says it is not.

Task 4: Code Explanation Using AI

Prompt: Ask Gemini to explain a Python function prime check line by line .

CODE AND OUTPUT:

```
def is_prime(number):
    if number < 2:
        return False
    for i in range(2, int(number**0.5) + 1):
        if number % i == 0:
            return False
    return True
try:
    user_num_str = input("Enter a number to check if it's prime: ")
    user_num = int(user_num_str)
    print(f"Is {user_num} prime? {is_prime(user_num)}")
except ValueError:
    print(f"Invalid input: '{user_num_str}' is not a valid integer.")
```

```
Enter a number to check if it's prime: 5
Is 5 prime? True
```

Explanation: This program asks you for a number and checks whether it is prime.
A number less than 2 is automatically not prime, and other numbers are tested for any divisor.It tries dividing the number by values from 2 up to its square root; if any divide evenly, it's not prime.Finally, it prints True or False, and if you enter something that isn't a valid number, it shows an error message.