AI ASSISSTANT CODING

ASSIGNMENT -2.2

B.Varun

2303A52020

BATCH- 38

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

PROMPT:

write a python code to extract all the negative numbers from a user given list of numbers and show the new list which contains no negative numbers

Code :

```
def filter_negative_numbers(numbers):
    return [num for num in numbers if num >= 0]

# Example usage 1:
my_list_1 = [1, -2, 3, -4, 5, 0, -6]
filtered_list_1 = filter_negative_numbers(my_list_1)
negative_numbers_1 = [num for num in my_list_1 if num < 0]
print(f"\nExample 1 (Mixed Numbers):")
print(f"Original list: {my_list_1}")
print(f"Filtered list (non-negative numbers): {filtered_list_1}")
print(f"Negative numbers: {negative_numbers_1}")

# Example usage 2 (No Negative Numbers):
my_list_2 = [10, 20, 30, 0, 5]
filtered_list_2 = filter_negative_numbers(my_list_2)
negative_numbers_2 = [num for num in my_list_2 if num < 0]
print(f"\nExample 2 (No Negative Numbers):")
print(f"Original list: {my_list_2}")
print(f"Filtered list (non-negative numbers): {filtered_list_2}")
print(f"Negative numbers: {negative_numbers_2}")
```

Output:

```
Example 1 (Mixed Numbers):
Original list: [1, -2, 3, -4, 5, 0, -6]
Filtered list (non-negative numbers): [1, 3, 5, 0]
Negative numbers: [-2, -4, -6]

Example 2 (No Negative Numbers):
Original list: [10, 20, 30, 0, 5]
Filtered list (non-negative numbers): [10, 20, 30, 0, 5]
Negative numbers: []
```

Explanation :

This program takes a list of numbers from the user and checks which of them are negative.

The user first enters several numbers separated by spaces. This input is taken as a string and then converted into a list of decimal numbers using the split() and float() functions inside a try block. If the user enters anything other than numbers, the program shows an error message and uses an empty list.

After that, the program uses list comprehension to find all the negative numbers in the list. It also creates another list that contains only non-negative numbers, meaning zero and positive values.

At the end, the program displays the original list, the negative numbers, and the list after removing all negative values**.**

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

PROMPT: write a python code that counts vowels, consonants and digits in a string

Code & output:

```python
def count_chars(input_string):
    vowels = 0
    consonants = 0
    digits = 0
    original_input_string = input_string # Store original for length and word count
    input_string_lower = input_string.lower()
    for char in input_string_lower:
        if char.isalpha():
            if char in 'aeiou':
                vowels += 1
            else:
                consonants += 1
        elif char.isdigit():
            digits += 1
    # Calculate word count and string length
    words = original_input_string.split()
    word_count = len(words)
    string_length = len(original_input_string) # Total characters including spaces and punctuation
    return vowels, consonants, digits, word_count, string_length
# Example usage 1:
text1 = "Hello World 123!"
v, c, d, wc, sl = count_chars(text1)
print(f"\nString: '{text1}'")
print(f"Vowels: {v}, Consonants: {c}, Digits: {d}, Words: {wc}, Length: {sl}")
# Example usage 2:
text2 = "Python is fun and easy to learn 2025"
v, c, d, wc, sl = count_chars(text2)
print(f"\nString: '{text2}'")
print(f"Vowels: {v}, Consonants: {c}, Digits: {d}, Words: {wc}, Length: {sl}")
```

```
String: 'Hello World 123!'
Vowels: 3, Consonants: 7, Digits: 3, Words: 3, Length: 16

String: 'Python is fun and easy to learn 2025'
Vowels: 9, Consonants: 16, Digits: 4, Words: 8, Length: 36
```

Explanation:

This program is used to count vowels, consonants, and digits in a given string.

A function called count_chars is created, which takes a string as input. Inside the function, the program checks each character one by one. If the character is a letter, it checks whether it is a vowel or a consonant and increases the count accordingly. If the character is a number, the digit count is increased.

After checking all the characters, the function returns the total number of vowels, consonants, and digits.

In the program, two example strings are given. The function is called for each string and the results are printed.

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

PROMPT: write a python code for palindrome checker by user input

Code & output:

```
def is_palindrome(input_string):
    # 1. Convert to lowercase
    cleaned_string = "".join(char for char in input_string.lower() if char.isalnum())

    # 2. Compare with its reverse
    return cleaned_string == cleaned_string[::-1]

# Example usage:
def print_palindrome_status(text):
    result = is_palindrome(text)
    status = 'palindrome' if result else 'not a palindrome'
    print(f"'{text}' is {status}")
print_palindrome_status('Racecar') # Expected: palindrome
print_palindrome_status('A man, a plan, a canal: Panama') # Expected: palindrome
print_palindrome_status('Hello') # Expected: not a palindrome
print_palindrome_status('No lemon, no melon') # Expected: palindrome
print_palindrome_status('Madam, I\'m Adam') # Expected: palindrome
```

```
'Racecar' is palindrome
'A man, a plan, a canal: Panama' is palindrome
'Hello' is not a palindrome
'No lemon, no melon' is palindrome
'Madam, I'm Adam' is palindrome
```

Explanation:

This program checks whether a given string is a palindrome.

The function removes spaces and converts the string to lowercase, then compares it with its reverse. If both are the same, it returns True; otherwise, it returns 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:**

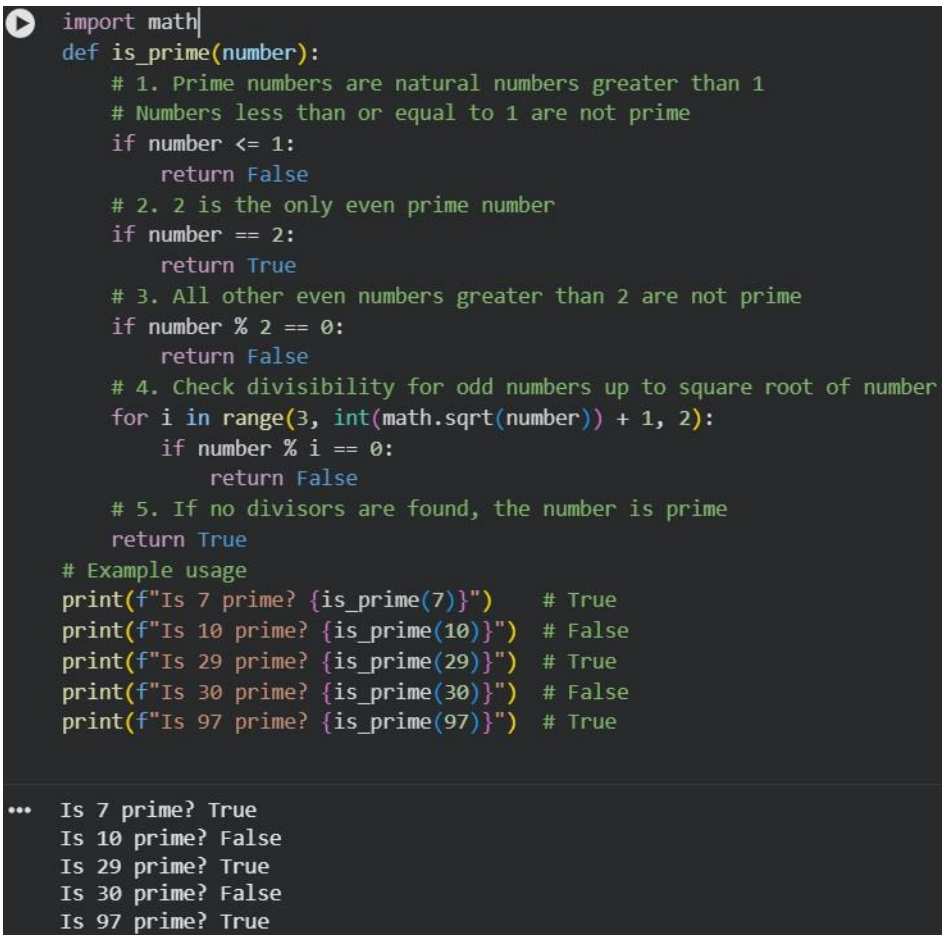
➢ Code snippet

➢ AI explanation

➢ Student comments on understanding

PROMPT: write a python code for Python function (prime check) line by line.

Code & Output:



```
import math
def is_prime(number):
    # 1. Prime numbers are natural numbers greater than 1
    # Numbers less than or equal to 1 are not prime
    if number <= 1:
        return False
    # 2. 2 is the only even prime number
    if number == 2:
        return True
    # 3. All other even numbers greater than 2 are not prime
    if number % 2 == 0:
        return False
    # 4. Check divisibility for odd numbers up to square root of number
    for i in range(3, int(math.sqrt(number)) + 1, 2):
        if number % i == 0:
            return False
    # 5. If no divisors are found, the number is prime
    return True
# Example usage
print(f"Is 7 prime? {is_prime(7)}")    # True
print(f"Is 10 prime? {is_prime(10)}")  # False
print(f"Is 29 prime? {is_prime(29)}")  # True
print(f"Is 30 prime? {is_prime(30)}")  # False
print(f"Is 97 prime? {is_prime(97)}")  # True
```

```
Is 7 prime? True
Is 10 prime? False
Is 29 prime? True
Is 30 prime? False
Is 97 prime? True
```

Explanation:

This program checks whether a given number is a prime number.

The function is_prime first removes numbers less than or equal to 1. It then checks special cases like 2 and even numbers. For the remaining odd numbers, it checks divisibility from 3 up to the square root of the number. If no divisor is found, the number is considered prime.