AI ASSISTED CODING

ASSIGNMENT-4

B.Sri Laxmi Gayathri

2303a52033

Batch:38

Lab 4: Advanced Prompt Engineering – Zero-shot, One-shot, and Few-shot Techniques

Task Description-1

• Zero-shot: Prompt AI with only the instruction. Write a Python function to determine whether a given number is prime

Expected Output-1

• A basic Python function to check if a number is prime, demonstrating correct logical conditions without relying on examples or additional context

Prompt:

Develop a Python function that returns True if a number is prime, otherwise False.

Comments:

1. The function first checks if the number is less than or equal to 1, which cannot be prime.
2. It uses a loop to test divisibility from 2 up to n-1.
3. If any divisor is found, the function returns False; otherwise, it returns True.

Explanation:

The function is_prime checks whether a given number is prime by first eliminating numbers less than or equal to 1, since they cannot be prime. It then uses a loop to test if the number is divisible by any integer from 2 up to one less than the number itself. If the number is divisible by any of these values, it means the number has more than two factors and is not prime. In such cases, the function immediately returns False. If no divisor is found after completing the loop, the function returns True, indicating that the number is prime.

Code & Output :

```python
def is_prime(num):
    if num <= 1:
        return False
    for i in range(2, int(num**0.5) + 1):
        if num % i == 0:
            return False
    return True

# Example usage:
print(f"Is 7 prime? {is_prime(7)}")
print(f"Is 10 prime? {is_prime(10)}")
print(f"Is 1 prime? {is_prime(1)}")
print(f"Is 2 prime? {is_prime(2)}")
```

```
Problems  Output  Debug Console  Terminal  Ports
PS C:\Users\srila> & C:/Users/srila/AppData/Local/Python/bin/python.exe c:/Users/srila/OneDrive/Documents/cursor5.py
Is 7 prime? True
Is 10 prime? False
Is 1 prime? False
Is 2 prime? True
```

Task Description-2

• One-shot: Provide one example: Input: [1, 2, 3, 4], Output: 10 to help AI generate a function that calculates the sum of elements in a list.

Expected Output-2

• A correct conversion function guided by the single example.

Prompt:

Write a Python function to calculate the sum of elements in a list.

Comments:

1. The function initializes a variable total to store the sum of elements.
2. It iterates through each element in the list and adds it to total.
3. After the loop completes, the function returns the final sum.

Explanation:

The function sum_of_list calculates the total of all elements in a given list. It starts by initializing a variable total with the value zero. Using a loop, each element of the list is added to total one by one. This approach follows the logic

demonstrated in the given example input and output. Finally, the function returns the accumulated sum of the list elements.

Code & Output:

```python
#write a program to implement the sum of elements in a list
def sum_list_elements(input_list):
    total_sum = 0
    for item in input_list:
        total_sum += item
    return total_sum

# Example usage:
input_list = [1, 2, 3, 4]
output = sum_list_elements(input_list)
print(f"Input: {input_list}, Output: {output}")
```

```
PS C:\Users\srila> & C:/Users/srila/AppData/Local/Python/bin/python.exe c:/Users/srila/OneDrive/Documents/cursor6.py
15
PS C:\Users\srila> & C:/Users/srila/AppData/Local/Python/bin/python.exe c:/Users/srila/OneDrive/Documents/cursor6.py
Input: [1, 2, 3, 4], Output: 10
```

Task Description-3

• Few-shot: Give 2–3 examples to create a function that extracts digits from an alphanumeric string.

Expected Output-3

• Accurate function that returns only the digits from alphanumeric string.

Prompt:

write a program to implement the extraction of digits from a string

Comments:

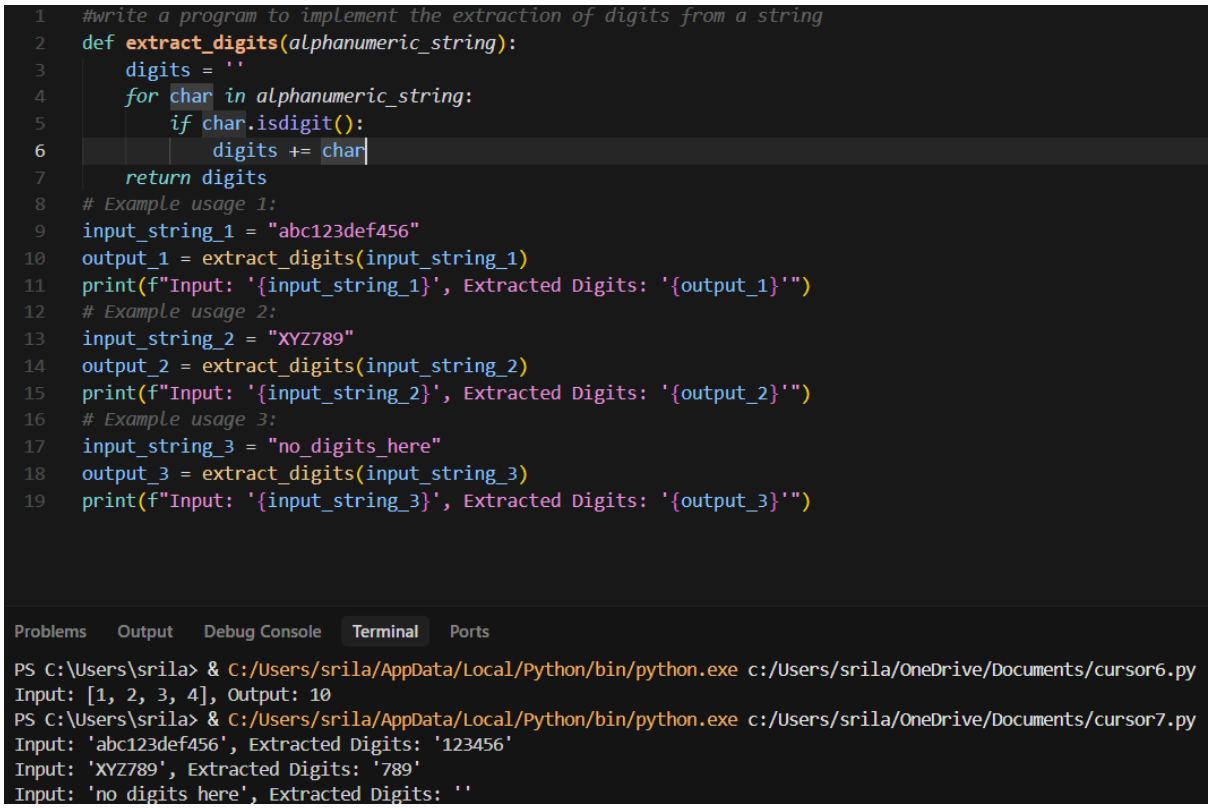
1. The function initializes an empty string to store the extracted digits.
2. It checks each character of the input string using the isdigit() method.
3. Only numeric characters are added to the result string, ignoring letters and symbols.

Explanation:

The function extract_digits takes an alphanumeric string as input and extracts only numeric characters. It scans each character in the string one by one. If a

character is a digit, it is added to the result string. The examples help the model understand that non-digit characters must be ignored. Finally, the function returns a string containing only digits in the original order.

Code & Output:



```python
#write a program to implement the extraction of digits from a string
def extract_digits(alphanumeric_string):
    digits = ''
    for char in alphanumeric_string:
        if char.isdigit():
            digits += char
    return digits
# Example usage 1:
input_string_1 = "abc123def456"
output_1 = extract_digits(input_string_1)
print(f"Input: '{input_string_1}', Extracted Digits: '{output_1}'")
# Example usage 2:
input_string_2 = "XYZ789"
output_2 = extract_digits(input_string_2)
print(f"Input: '{input_string_2}', Extracted Digits: '{output_2}'")
# Example usage 3:
input_string_3 = "no_digits_here"
output_3 = extract_digits(input_string_3)
print(f"Input: '{input_string_3}', Extracted Digits: '{output_3}'")
```

```
Problems  Output  Debug Console  Terminal  Ports
PS C:\Users\srila> & C:/Users/srila/AppData/Local/Python/bin/python.exe c:/Users/srila/OneDrive/Documents/cursor6.py
Input: [1, 2, 3, 4], Output: 10
PS C:\Users\srila> & C:/Users/srila/AppData/Local/Python/bin/python.exe c:/Users/srila/OneDrive/Documents/cursor7.py
Input: 'abc123def456', Extracted Digits: '123456'
Input: 'XYZ789', Extracted Digits: '789'
Input: 'no digits here', Extracted Digits: ''
```

Task Description-4

• Compare zero-shot vs few-shot prompting for generating a function that counts the number of vowels in a string.

Expected Output-4

• Output comparison + student explanation on how examples helped the model.

Prompt:

Write a Python function to count the number of vowels in a string.

Comments:

1.Zero-shot prompting provides only an instruction, so the model relies on general knowledge.

2.Few-shot prompting includes examples that clearly show how vowels should be counted.

3.The presence of examples improves accuracy and reduces ambiguity in the generated function.

Explanation:

In the zero-shot approach, the AI relies only on the instruction and may produce a basic solution without handling all cases clearly. In the few-shot approach, the given examples guide the AI to correctly identify vowels and count them accurately.

Code & Output :

```python
#write a program to implement the counting of vowels in a string
def count_vowels_zero_shot(input_string):
    """
    Counts the number of vowels (a, e, i, o, u) in a string, case-insensitively.
    Args:
        input_string (str): The string to analyze.
    Returns:
        int: The total count of vowels in the string.
    """
    vowels = "aeiou"
    vowel_count = 0
    for char in input_string:
        if char.lower() in vowels:
            vowel_count += 1
    return vowel_count
# Example usage:
print(f"'Hello World' has {count_vowels_zero_shot('Hello World')} vowels.")
print(f"'Python Programming' has {count_vowels_zero_shot('Python Programming')} vowels.")
print(f"'AEIOU' has {count_vowels_zero_shot('AEIOU')} vowels.")
print(f"'Rhythm' has {count_vowels_zero_shot('Rhythm')} vowels.")
```

```
PS C:\Users\srila> & C:/Users/srila/AppData/Local/Python/bin/python.exe c:/Users/srila/OneDrive/Documents/cursor8.py
'Hello World' has 3 vowels.
'Python Programming' has 4 vowels.
'AEIOU' has 5 vowels.
'Rhythm' has 0 vowels.
```

Task Description-5

• Use few-shot prompting with 3 sample inputs to generate a function that determines the minimum of three numbers without using the built-in min() function.

Expected Output-5

• A function that handles all cases with correct logic based on example patterns.

Prompt:

Write a Python function to find the minimum of three numbers without using the built-in min() function.

Comments:

The function compares three numbers using conditional statements instead of the built-in min() function.

It checks all possible conditions to ensure the smallest value is correctly identified.

The example inputs guide the logic to handle equal and unequal values correctly.

Explanation:

The function minimum_of_three determines the smallest value among three numbers using conditional statements. It compares the first number with the other two and returns it if it is the smallest. If not, it checks whether the second number is smaller than the remaining values. Otherwise, the third number is returned as the minimum. The examples guide the logic to correctly handle equal values and different number combinations.

Code & Output:

```python
#write a program to generate a function that determines
#the minimum of three numbers without using the built-in min() function.
def find_minimum_few_shot(a, b, c):
    minimum = a
    if b < minimum:
        minimum = b
    if c < minimum:
        minimum = c
    return minimum
print(find_minimum_few_shot(5, 10, 3))
print(find_minimum_few_shot(1, 1, 1))
print(find_minimum_few_shot(7, 2, 9))
print(find_minimum_few_shot(100, 50, 75))
print(find_minimum_few_shot(-5, -2, -8))
```

```
PS C:\Users\srila> & C:/Users/srila/AppData/Local/Python/bin/python.exe c:/Users/srila/OneDrive/Documents/cursor9.py
3
1
2
50
-8
```