School of Computer Science and Artificial Intelligence

Lab Assignment # 4.2

Program   : B. Tech (CSE)

Specialization   : AIML

Course Title   : AI Assisted

Coding Course Code: 23CS002PC304

Semester   : VI

Academic Session   : 2025-2026

Name of Student   : P. SricharanGoud

Enrollment No.   : 2303A52100

Batch No.   : 33

Date   : 20/01/26

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

Lab Objectives

- To explore and apply different levels of prompt examples in AI-assisted code generation.
- To understand how zero-shot, one-shot, and few-shot prompting affect AI output quality.
- To evaluate the impact of context richness and example quantity on AI performance.
- To build awareness of prompt strategy effectiveness for different problem types.

Lab Outcomes (LOs)

After completing this lab, students will be able to:

- Use zero-shot prompting to instruct AI with minimal context.
- Use one-shot prompting with a single example to guide AI code generation.
- Apply few-shot prompting using multiple examples to improve AI responses.
- Compare AI outputs across the three prompting strategies.

Task Description – 1: Zero-shot Prompting

Problem: Write a Python function to determine whether a given number is prime.

Prompt Used (Zero-shot)

"Write a Python function to check whether a given number is prime."

Python Code:

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

Output

Returns True if the number is prime, otherwise False.

Task Description – 2: One-shot Prompting

Problem: Generate a function that calculates the sum of elements in a list.

Prompt Used (One-shot)

"Write a Python function that returns the sum of all elements in a list.\n Example:\n Input: [1, 2, 3, 4]\n Output: 10"

Python Code:

```
def list_sum(arr):
    total = 0
    for num in arr:
        total += num
        return total
```

Output

For input [1, 2, 3, 4], output is 10.

Task Description – 3: Few-shot Prompting

Problem: Create a function that extracts digits from an alphanumeric string.

Prompt Used (Few-shot)

"Extract digits from an alphanumeric string.\n Examples:\n Input: 'a1b2c3' → Output: '123'\n Input: 'x9y8z' → Output: '98'\n Input: 'abc123' → Output: '123'"

Python Code:

```
def extract_digits(text):
    digits = ''
    for ch in text:
        if ch.isdigit():
            digits += ch
            return digits
```

Output

Returns only the numeric characters from the input string.

Task Description – 4: Zero-shot vs Few-shot Comparison

Problem: Generate a function that counts the number of vowels in a string.

Zero-shot Prompt

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

```
def count_vowels_zero(s):
    vowels = 'aeiouAEIOU'
    count = 0
    for ch in s:
        if ch in vowels:
            count += 1
    return count
```

Few-shot Prompt

"Count vowels in a string.\nExamples:\nInput: 'hello' → Output: 2\nInput: 'AI Model' → Output: 4\nInput: 'xyz' → Output: 0"

```
def count_vowels_few(s):
    vowels = 'aeiouAEIOU'
    return sum(1 for ch in s if ch in vowels)
```

Comparison & Explanation

- Zero-shot produced a correct but more verbose solution.

- Few-shot produced a concise and optimized solution.
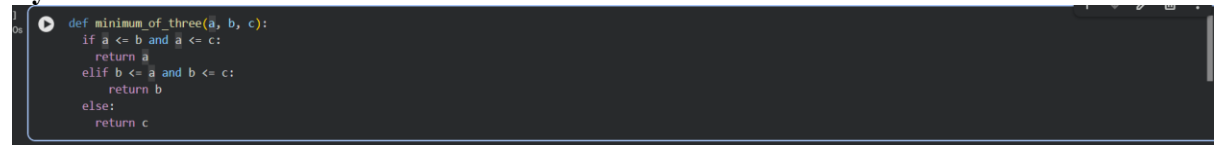- Examples helped the model understand edge cases and improve efficiency.

Task Description – 5: Few-shot Prompting
Problem: Determine the minimum of three numbers without using min().

Prompt Used (Few-shot)
"Find the minimum of three numbers without using built-in functions.\nExamples:\nInput: (3, 5, 1) → Output: 1\nInput: (10, 2, 7) → Output: 2\nInput: (4, 4, 9) → Output: 4"

Python Code:

```
def minimum_of_three(a, b, c):
    if a <= b and a <= c:
        return a
    elif b <= a and b <= c:
        return b
    else:
        return c
```

Output
Correctly returns the minimum value for all input combinations.

Conclusion
This lab demonstrated that:
- Zero-shot prompting works well for simple tasks.
- One-shot prompting helps align logic with expected output.
- Few-shot prompting significantly improves accuracy, structure, and optimization.
- Providing examples enhances AI understanding and output quality.