AI Assisted Coding – Assignment 4.2

Course Code: 23CS002PC304
Course Title: AI Assisted Coding
Lab: Advanced Prompt Engineering – Zero-shot, One-shot, Few-shot Techniques
Student Name: Ravula seetharam Reddy

Roll no : 2303A52440

Batch : 34
Year/Sem: III / II
Regulation: R23

Task Description – 1: Zero-shot Prompting

Prompt Used (Zero-shot)

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

Code

```
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
```

Explanation

- Zero-shot prompting provides only instructions without examples.
- The function checks divisibility from 2 to $\sqrt{n}$.
- If any divisor exists, the number is not prime.

Sample Output

- is_prime(7) → True

- is_prime(10) → False

Task Description – 2: One-shot Prompting

Prompt Used (One-shot)

> "Input: [1, 2, 3, 4], Output: 10. Write a Python function that calculates the sum of elements in a list."

Code

```
def sum_list(lst):
    total = 0
    for num in lst:
        total += num
    return total
```

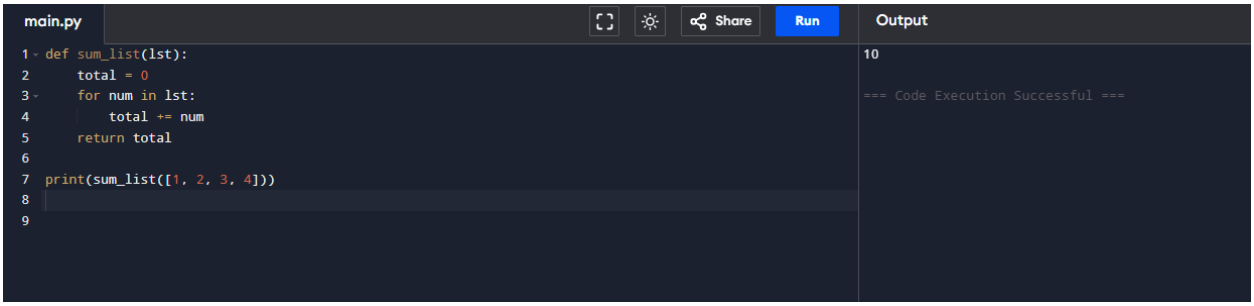
Explanation

- One example guides the AI toward summation logic.
- Iterates through the list and accumulates the total.

Sample Output

- sum_list([1, 2, 3, 4]) → 10

Screenshot Placeholder:



Task Description – 3: Few-shot Prompting

Prompt Used (Few-shot)

"Input: 'a1b2c3' → Output: '123'
Input: 'abc456' → Output: '456'
Input: '9x8y' → Output: '98'
Write a Python function to extract digits from an alphanumeric string."

Code

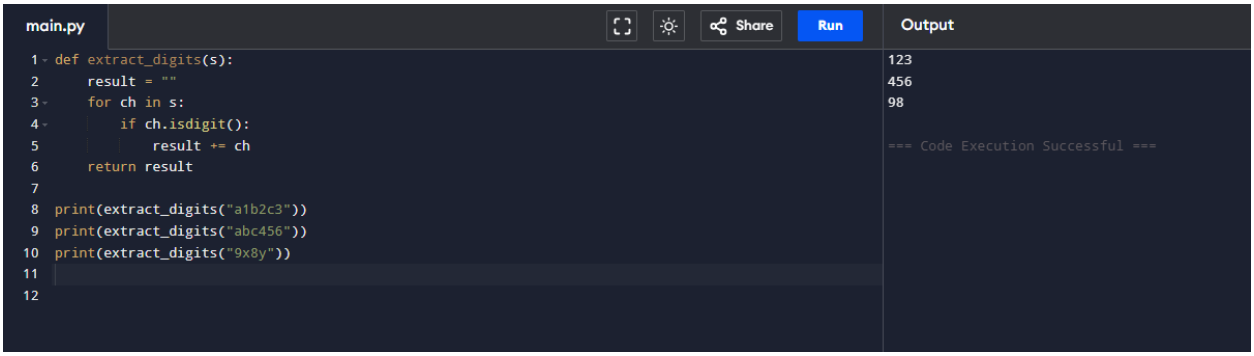
```
def extract_digits(s):
    result = ""
    for ch in s:
        if ch.isdigit():
            result += ch
    return result
```

Explanation

- Multiple examples clarify the pattern.
- Digits are identified using `isdigit()`.

Sample Output

- extract_digits("a1b2c3") → "123"



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

Zero-shot Prompt

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

Zero-shot Code

```
def count_vowels_zero(s):
    vowels = "aeiouAEIOU"
    count = 0
    for ch in s:
        if ch in vowels:
```

```
            count += 1
    return count
```

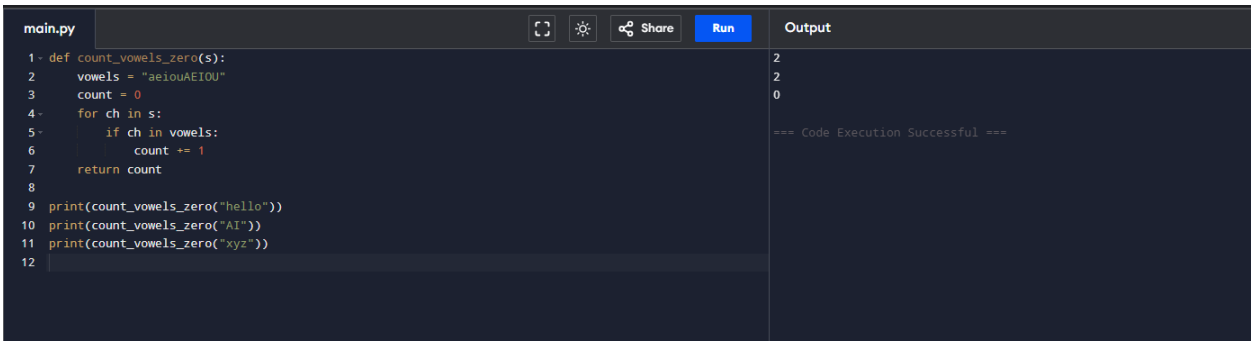
Few-shot Prompt

"Input: 'hello' → Output: 2
Input: 'AI' → Output: 2
Input: 'xyz' → Output: 0
Write a Python function to count vowels in a string."

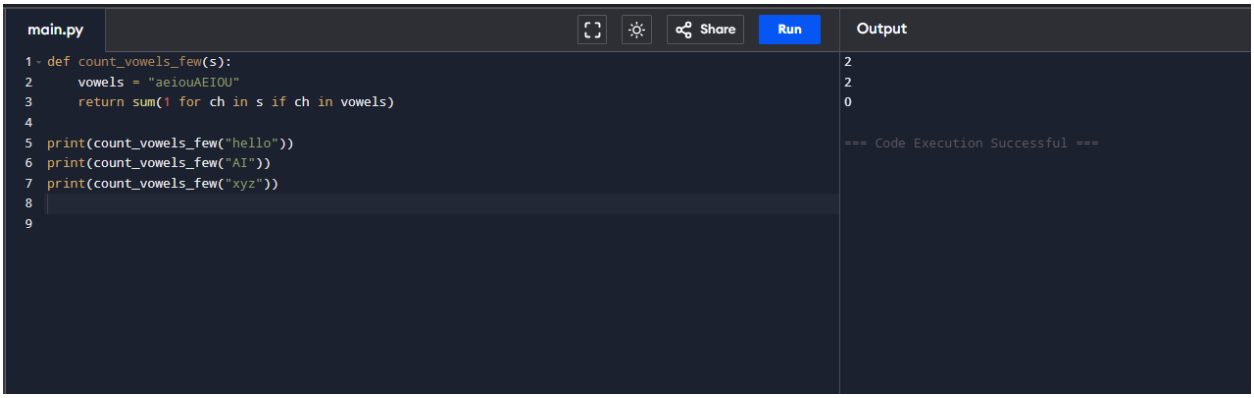
Few-shot Code

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

📸 **Screenshot Placeholder (Zero-shot vowel count):** > Paste screenshot of zero-shot code execution and output here.



📸 **Screenshot Placeholder (Few-shot vowel count):**



Comparison & Explanation

- Zero-shot gives a basic loop-based solution.
- Few-shot improves clarity and efficiency.
- Examples help AI understand edge cases.

Task Description – 5: Few-shot Prompting (Minimum of Three Numbers)

Prompt Used (Few-shot)

> "Input: (3, 5, 1) → Output: 1
> Input: (10, 2, 7) → Output: 2
> Input: (4, 4, 9) → Output: 4
> Write a Python function to find the minimum of three numbers without using min()."

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
```

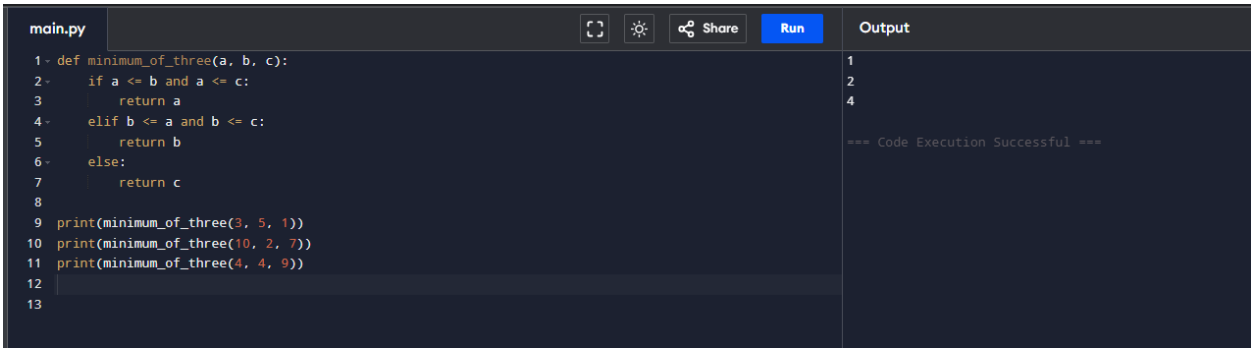
Explanation

- Conditional comparisons handle all cases.
- No built-in `min()` function is used.

Sample Output

- minimum_of_three(3, 5, 1) → 1

📸 **Screenshot Placeholder:**



Conclusion

- Zero-shot prompting works for simple logic.
- One-shot improves accuracy with guidance.
- Few-shot produces the most reliable and optimized results.

- Prompt richness directly impacts output quality.