

ASSIGNMENT-4.2

NAME- HABEEBA KHANAM

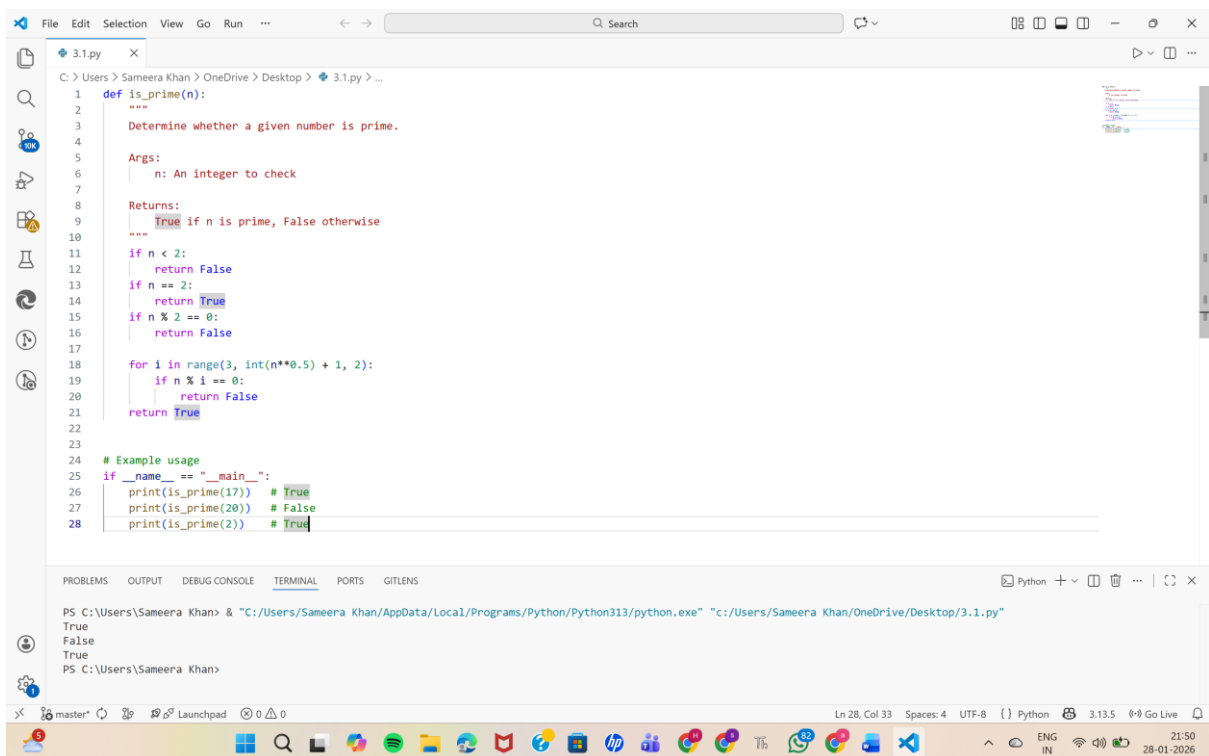
ROLL_NO:2303A51474

BATCH NO:29

TASK-1: ZERO-SHOT PROMPTING

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

CODE:



```
File Edit Selection View Go Run ... Search
3.1.py
C:\Users\Sameera Khan\OneDrive\Desktop> 3.1.py ...
1 def is_prime(n):
2     """
3     Determine whether a given number is prime.
4
5     Args:
6         n: An integer to check
7
8     Returns:
9         True if n is prime, False otherwise
10
11     """
12     if n < 2:
13         return False
14     if n == 2:
15         return True
16     if n % 2 == 0:
17         return False
18
19     for i in range(3, int(n**0.5) + 1, 2):
20         if n % i == 0:
21             return False
22     return True
23
24 # Example usage
25 if __name__ == "__main__":
26     print(is_prime(17)) # True
27     print(is_prime(20)) # False
28     print(is_prime(2)) # True

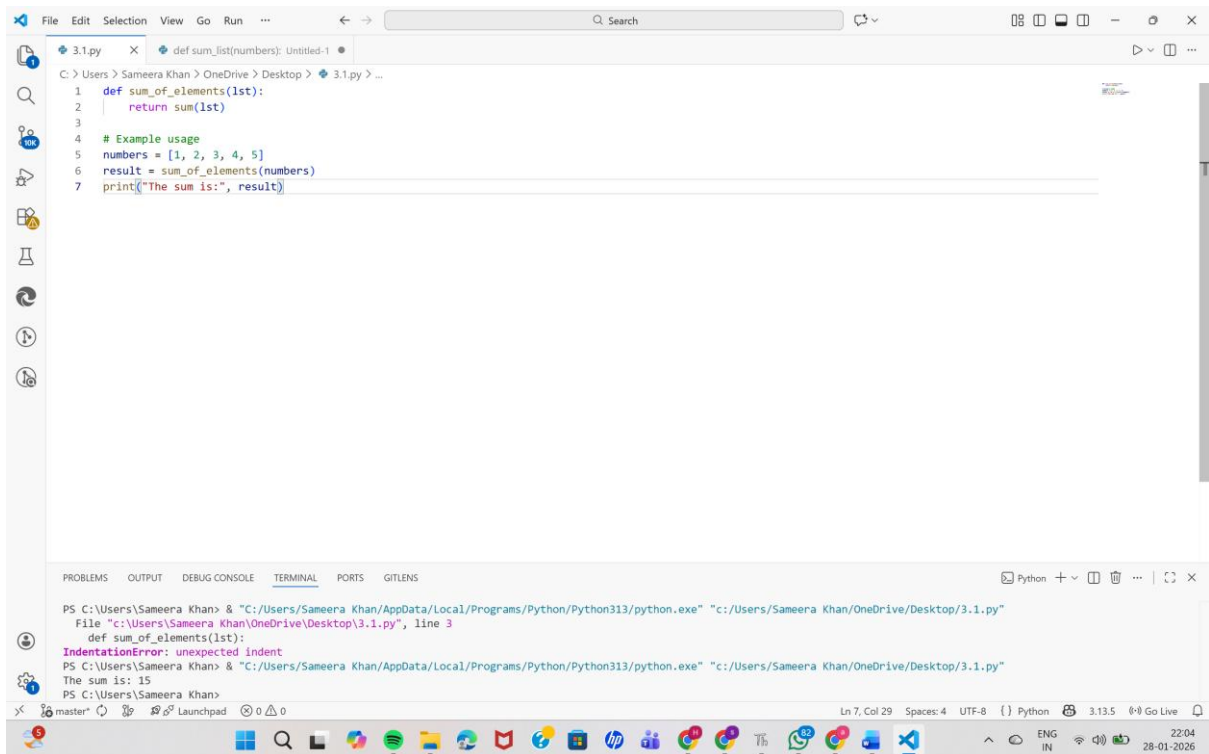
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS
Python + - - - - -
PS C:\Users\Sameera Khan> & "C:/Users/Sameera Khan/AppData/Local/Programs/Python/Python313/python.exe" "c:/Users/Sameera Khan/OneDrive/Desktop/3.1.py"
True
False
True
PS C:\Users\Sameera Khan>
```

OBSERVATION:

The AI model successfully interprets the definition of a prime number solely from the prompt, without relying on examples or extra instructions. It applies correct mathematical logic to distinguish prime numbers from non-prime numbers. Additionally, the model produces Python code that is both syntactically correct and logically accurate, demonstrating effective zero-shot reasoning.

TASK-2

PROMPT: Write a Python function that calculates the sum of elements in a list.



The screenshot shows a Visual Studio Code editor window with a Python file named `3.1.py`. The code defines a function `sum_of_elements` that takes a list `lst` and returns the sum of its elements. An example usage is provided: `numbers = [1, 2, 3, 4, 5]`, `result = sum_of_elements(numbers)`, and `print("The sum is:", result)`. The terminal at the bottom shows the command `python 3.1.py` being executed, which results in the output `The sum is: 15`. The terminal also shows the command `python 3.1.py` being executed again, which results in the output `The sum is: 15`.

```
C:\Users\Sameera Khan> python 3.1.py
1 def sum_of_elements(lst):
2     return sum(lst)
3
4 # Example usage
5 numbers = [1, 2, 3, 4, 5]
6 result = sum_of_elements(numbers)
7 print("The sum is:", result)
```

```
PS C:\Users\Sameera Khan> python 3.1.py
The sum is: 15
PS C:\Users\Sameera Khan> python 3.1.py
The sum is: 15
```

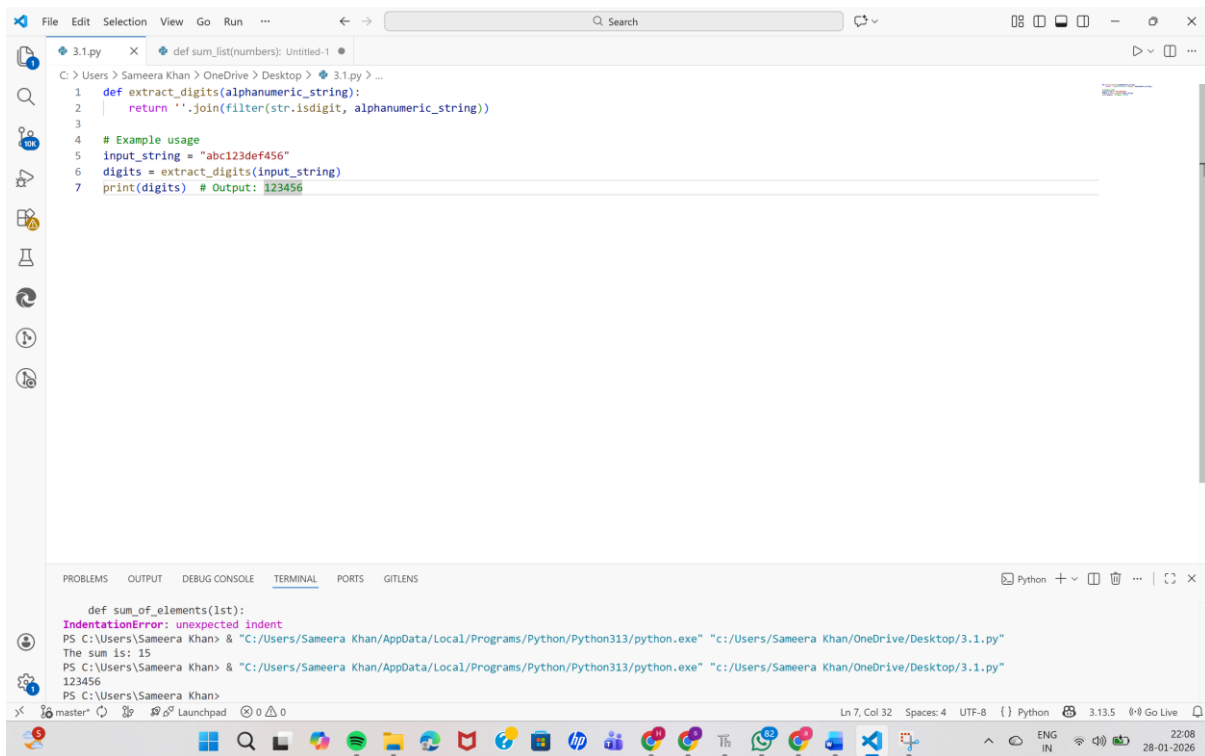
OBSERVATION:

By observing one example, the AI understands how inputs relate to outputs and correctly applies the same pattern to other lists of numbers.

TASK-3

PROMPT: Write a Python function that extracts only digits from an alphanumeric string.

CODE:



```
File Edit Selection View Go Run ... Search
3.1.py x def sum_list(numbers): Untitled-1
C:\Users\Sameera Khan> OneDrive\ Desktop > 3.1.py > ...
1 def extract_digits(alphanumeric_string):
2     return ''.join(filter(str.isdigit, alphanumeric_string))
3
4 # Example usage
5 input_string = "abc123def456"
6 digits = extract_digits(input_string)
7 print(digits) # Output: 123456

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS
def sum_of_elements(lst):
IndentationError: unexpected indent
PS C:\Users\Sameera Khan> & "C:/Users/Sameera Khan/AppData/Local/Programs/Python/Python313/python.exe" "c:/Users/Sameera Khan/OneDrive/Desktop/3.1.py"
The sum is: 15
PS C:\Users\Sameera Khan> & "C:/Users/Sameera Khan/AppData/Local/Programs/Python/Python313/python.exe" "c:/Users/Sameera Khan/OneDrive/Desktop/3.1.py"
123456
PS C:\Users\Sameera Khan>
```

OBSERVATION:

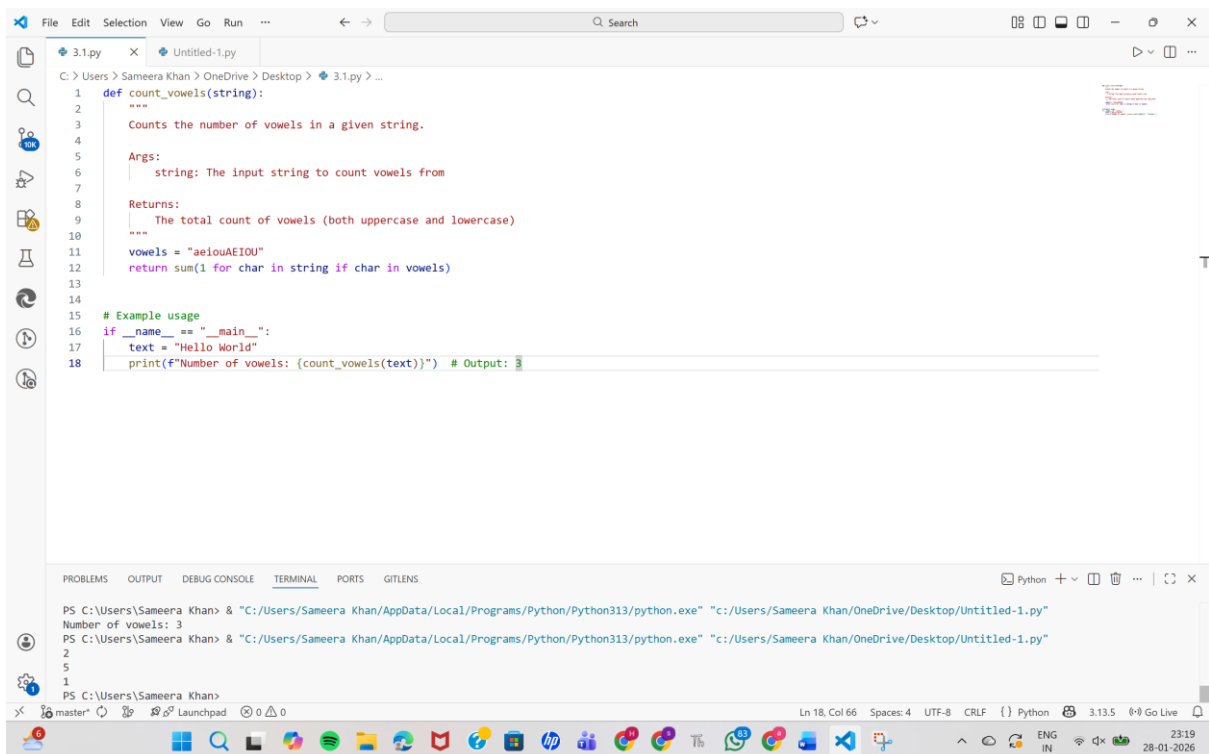
Providing multiple examples helps the AI model precisely identify the required pattern. The model correctly filters out alphabetic characters and focuses

solely on digits. This approach results in more reliable and unambiguous outputs than zero-shot and one-shot prompting.

TASK-4

PROMPT: ZERO-SHOT: Write a Python function that counts the number of vowels in a string.

FEW-SHOT: Write a Python function that counts the number of vowels in a string. Examples: Input: "hello" Output: 2 Input: "AEIOU" Output: 5 Input: "chatgpt" Output: 2



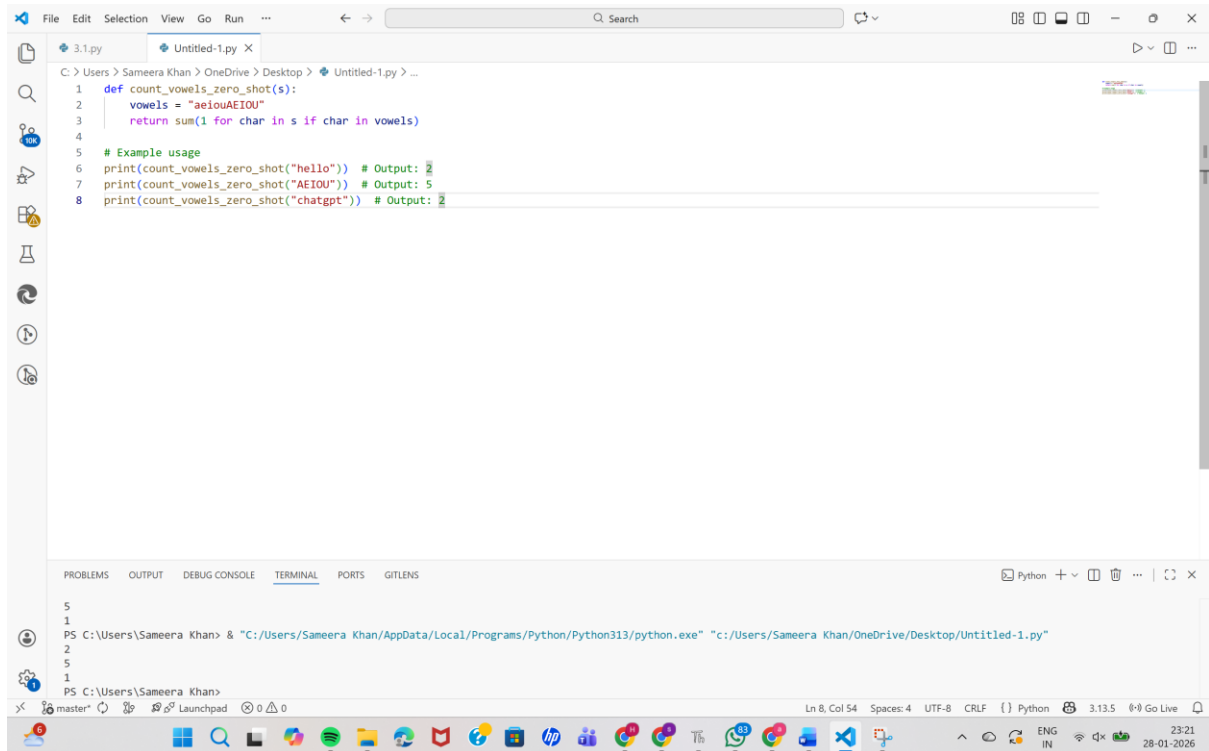
The screenshot shows a Visual Studio Code editor window with a Python file named '3.1.py'. The code defines a function 'count_vowels' that counts the number of vowels in a given string. The function uses a set of vowels 'aeiouAEIOU' and a list comprehension to count them. An example usage is provided in the main block, where the string 'Hello World' is passed to the function, resulting in an output of 3.

```
1 def count_vowels(string):
2     """
3     Counts the number of vowels in a given string.
4
5     Args:
6         string: The input string to count vowels from
7
8     Returns:
9         The total count of vowels (both uppercase and lowercase)
10
11     vowels = "aeiouAEIOU"
12     return sum(1 for char in string if char in vowels)
13
14
15 # Example usage
16 if __name__ == "__main__":
17     text = "Hello World"
18     print(f"Number of vowels: {count_vowels(text)}") # Output: 3
```

The terminal output shows the execution of the script, confirming the output of 3 for the input 'Hello World'.

```
PS C:\Users\Sameera Khan> "C:/Users/Sameera Khan/AppData/Local/Programs/Python/Python313/python.exe" "C:/Users/Sameera Khan/OneDrive/Desktop/Untitled-1.py"
Number of vowels: 3
PS C:\Users\Sameera Khan> "C:/Users/Sameera Khan/AppData/Local/Programs/Python/Python313/python.exe" "C:/Users/Sameera Khan/OneDrive/Desktop/Untitled-1.py"
2
5
1
PS C:\Users\Sameera Khan>
```

ZERO SHOT:



The screenshot shows a Visual Studio Code editor window with a Python file named 'Untitled-1.py'. The code defines a function 'count_vowels_zero_shot(s)' that counts vowels in a string 's'. The function uses a set of vowels 'aeiouAEIOU' and a list comprehension to count them. Below the function, there are three example usage lines with their corresponding outputs. The terminal at the bottom shows the command to run the script and the resulting output.

```
1 def count_vowels_zero_shot(s):
2     vowels = "aeiouAEIOU"
3     return sum(1 for char in s if char in vowels)
4
5 # Example usage
6 print(count_vowels_zero_shot("hello")) # Output: 2
7 print(count_vowels_zero_shot("AEIOU")) # Output: 5
8 print(count_vowels_zero_shot("chatgpt")) # Output: 2
```

```
PS C:\Users\Sameera Khan> & "C:/Users/Sameera Khan/AppData/Local/Programs/Python/Python313/python.exe" "c:/Users/Sameera Khan/OneDrive/Desktop/Untitled-1.py"
5
1
2
5
1
PS C:\Users\Sameera Khan>
```

OBSERVATION:

FEW-SHOT OBSERVATION

The provided examples clearly define what characters should be counted as vowels

The model confidently includes both uppercase and lowercase vowels due to examples ZERO SHOT:

zero shot prompting the AI guesses the intent based on general knowledge which may vary for ambiguous tasks

TASK-5

PROMPT:

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

The screenshot shows a Python IDE with a file named '3.1.py' open. The code defines a function 'minimum_of_three(a, b, c)' that returns the minimum of three numbers using conditional statements. Below the function, there is an example usage with three numbers: num1 = 5, num2 = 3, and num3 = 8. The function is called with these numbers, and the output is printed: 'The minimum of the three numbers is: 3'. The terminal window at the bottom shows the command to run the script and the resulting output.

```
1 def minimum_of_three(a, b, c):
2     if a < b and a < c:
3         return a
4     elif b < a and b < c:
5         return b
6     else:
7         return c
8
9 # Example usage
10 num1 = 5
11 num2 = 3
12 num3 = 8
13 print("The minimum of the three numbers is:", minimum_of_three(num1, num2, num3))
```

```
PS C:\Users\Sameera Khan> & "C:/Users/Sameera Khan/AppData/Local/Programs/Python/Python313/python.exe" "c:/Users/Sameera Khan/OneDrive/Desktop/Untitled-1.py"
2
5
1
PS C:\Users\Sameera Khan> & "C:/Users/Sameera Khan/AppData/Local/Programs/Python/Python313/python.exe" "c:/Users/Sameera Khan/OneDrive/Desktop/3.1.py"
The minimum of the three numbers is: 3
PS C:\Users\Sameera Khan>
```

OBSERVATION:

The examples help the AI understand how to compare values to find the minimum. The model correctly accounts for equal values and implements the logic using conditional statements rather than built-in functions.

