ASSIGNMENT

NAME: B. Akhira Nandhini

HTNO:2303A51516

BATCH:22

AI ASSITANT CODING

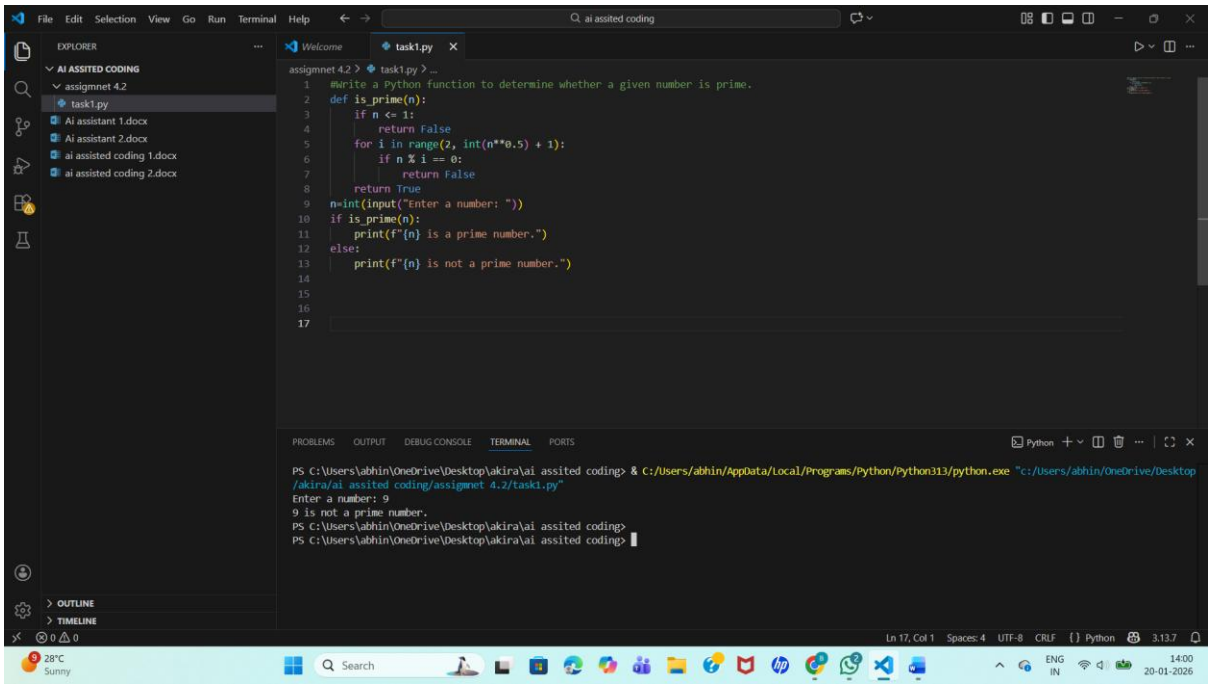
Task Description-1

• 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

INPUT and OUTPUT:



Explanation:

- A number that is less than or equal to 1 cannot be a prime number, so first we check for that condition.

- Then, to find whether the number is prime or not, we divide it by numbers starting from 2 up to the square root of the given number.
- If the number gets divided by any of these values, it means it has factors and is not a prime number.
- If no divisor is found in this range, then the number has only two factors (1 and itself), so it is a prime number.

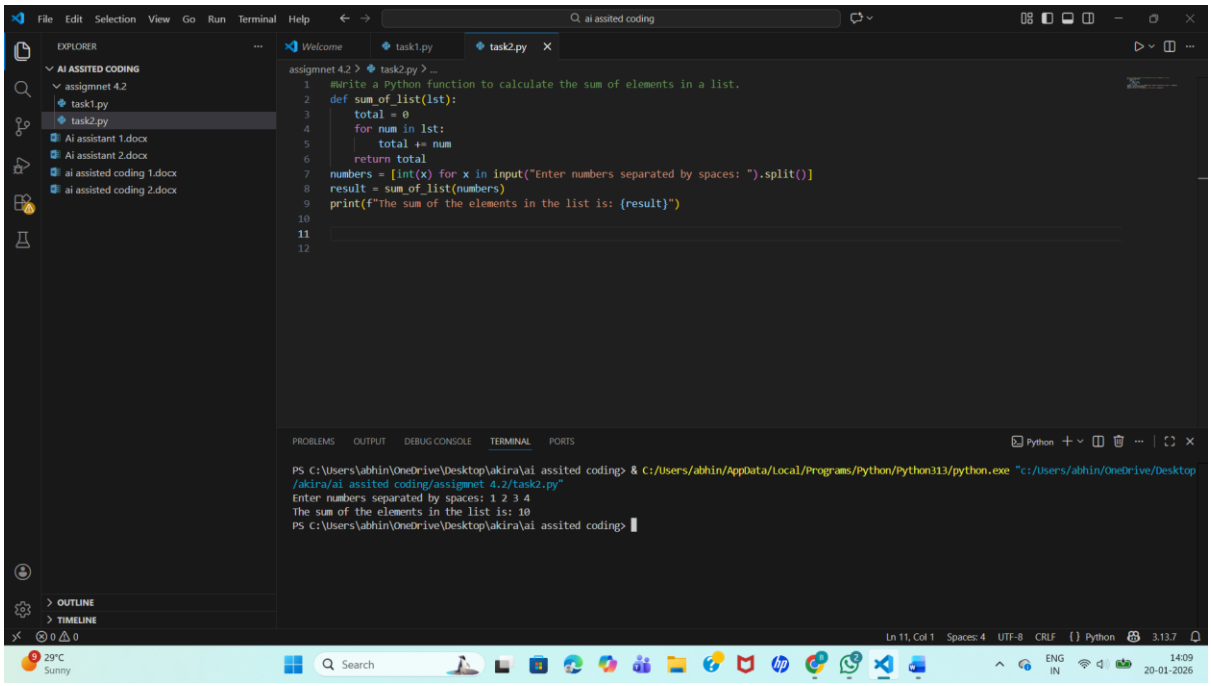
Task Description-2

• 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.

INPUT and OUTPUT:



EXPLANATION:

- In this task, the function is used to find the sum of all elements present in a list.
- First, a variable is initialized to store the total sum.
- Then, the function goes through each element in the list one by one using a loop and adds each value to the total.

- After all the elements are added, the final sum is returned as the output.
- This approach is simple and clearly shows how the sum of a list can be calculated using basic logic without directly using any built-in function.

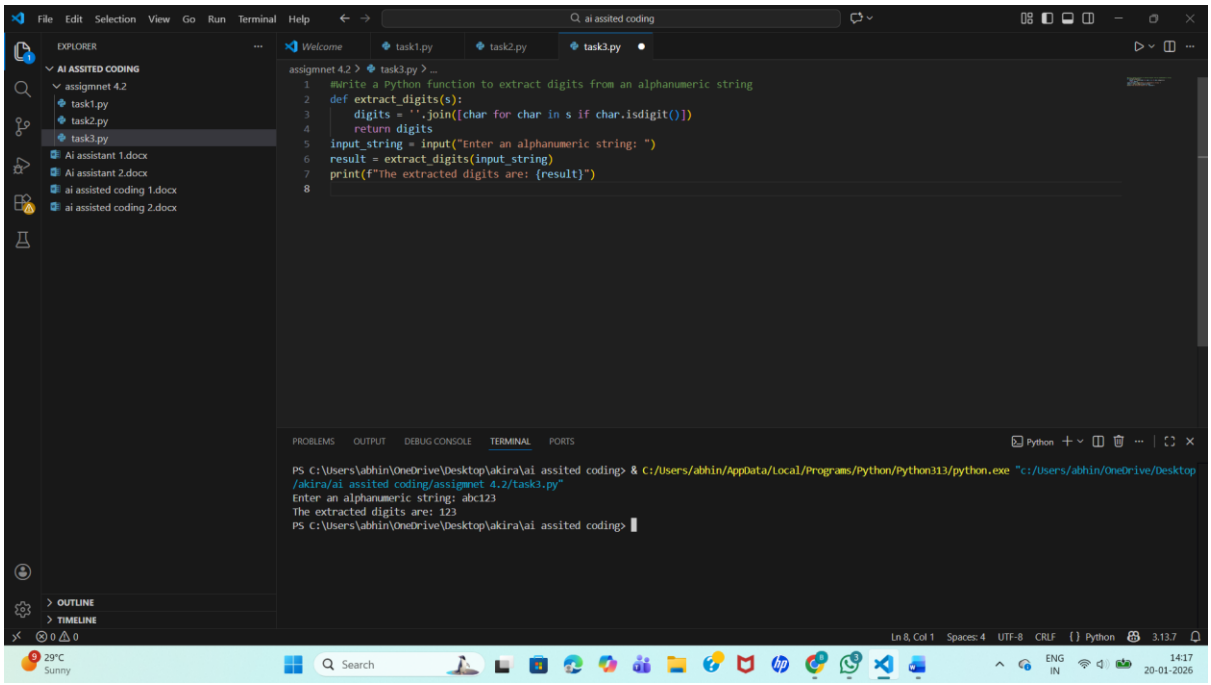
Task Description-3

• 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.

INPUT AND OUTPUT:



EXPLANATION:

- In this task, the function is designed to extract only the digits from an alphanumeric string.
- The function checks each character of the given string one by one.
- If the character is a number, it is added to a new string, and if it is a letter or any other symbol, it is ignored.
- By the end of the loop, the new string contains only the digits from the original input.

- This method makes it easy to separate numbers from text and works correctly for any alphanumeric string**.**

Task Description-4

• Compare zero-shot vs few-shot prompting for generating a function that counts the

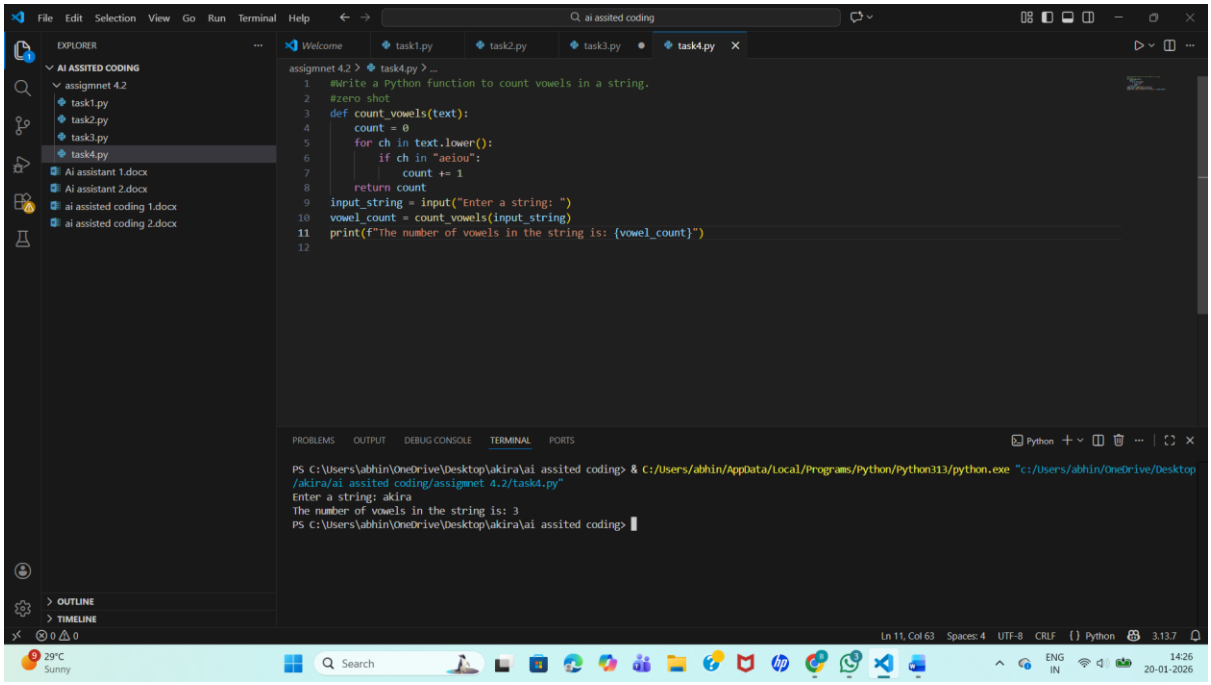
number of vowels in a string.

Expected Output-4

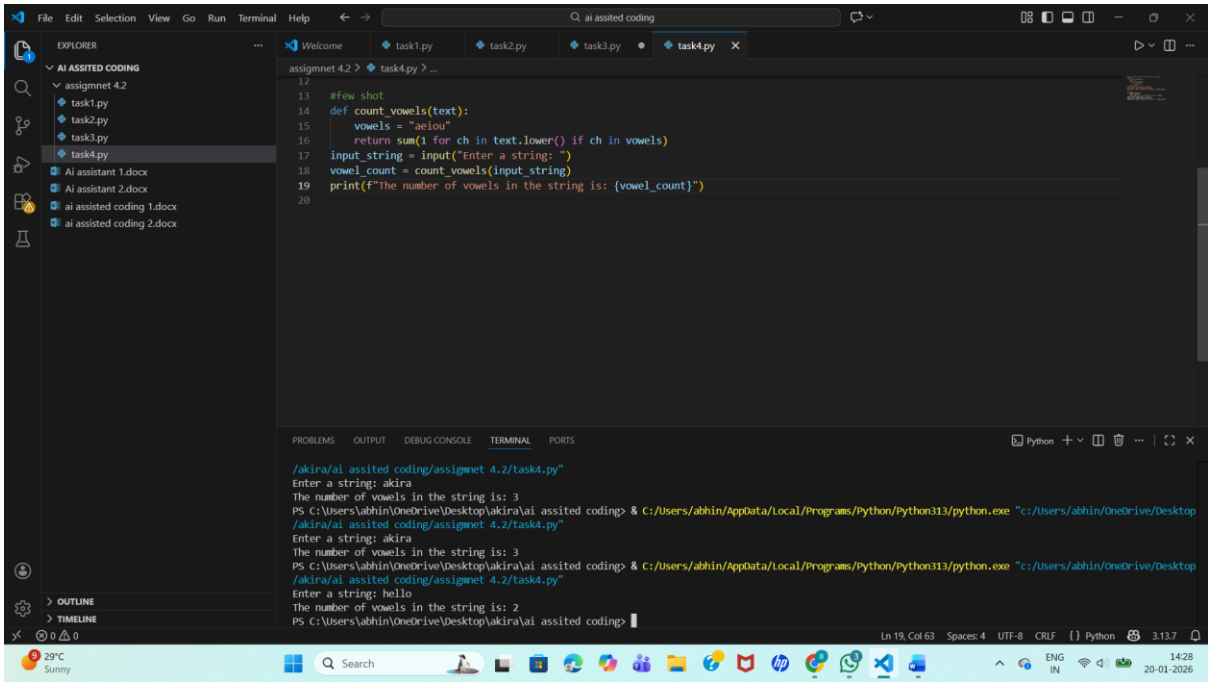
• Output comparison + student explanation on how examples helped the model.

INPUT AND OUTPUT:

Zero shot:



Few shot:



EXPLANATION:

- In this task, zero-shot and few-shot prompting are compared using a function that counts the number of vowels in a string.
- In the zero-shot approach, only the instruction is given, so the function is written using basic logic with loops and conditions.
- In the few-shot approach, examples are provided, which help in understanding the pattern more clearly and allow the function to be written in a more optimized and concise way.
- By comparing both outputs, it is observed that while both methods give correct results, few-shot prompting improves clarity, efficiency, and overall quality of the generated code because the examples guide the model better.

Task Description-5

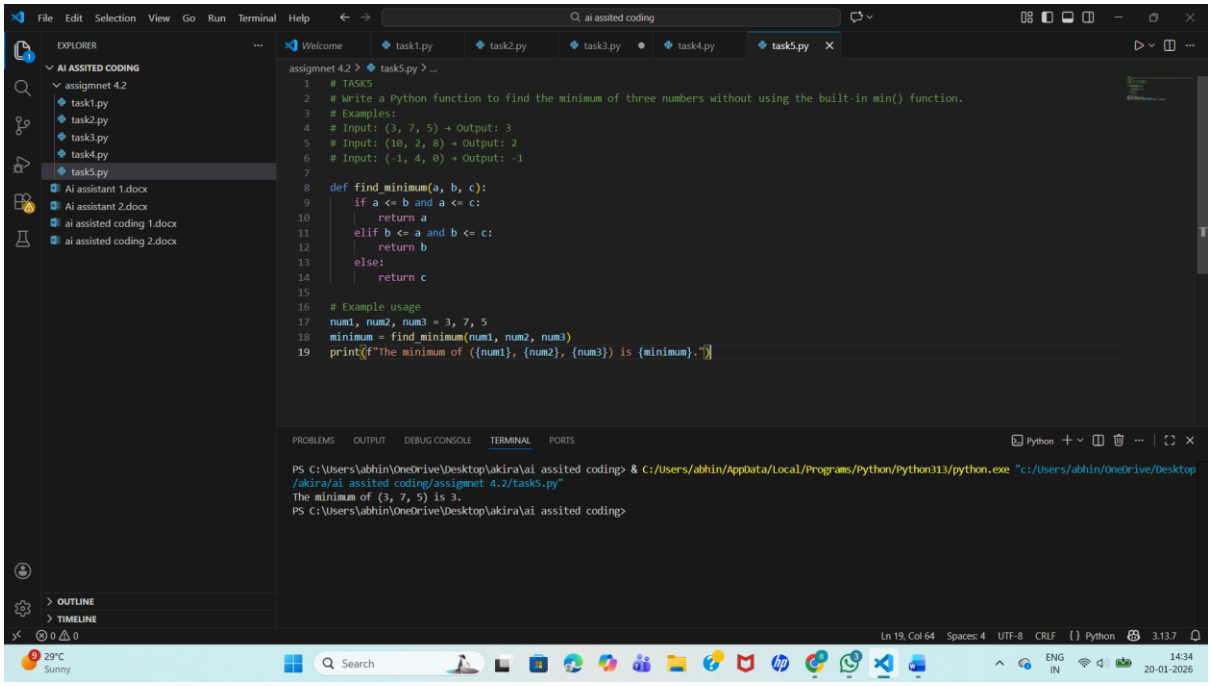
• Use few-shot prompting with 3 sample inputs to generate a function that determines

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.

INPUT AND OUTPUT:



EXPLANATION:

- In this task, the function is written to find the minimum of three numbers without using the built-in min() function.
- The function compares the three values using conditional statements. First, it checks whether the first number is smaller than or equal to the other two.
- If not, it then checks the second number in the same way. If neither of these conditions is true, the third number is considered the smallest.
- This logic ensures that all possible cases are handled correctly, and the correct minimum value is returned based on the given examples.