AI ASSISTED CODING

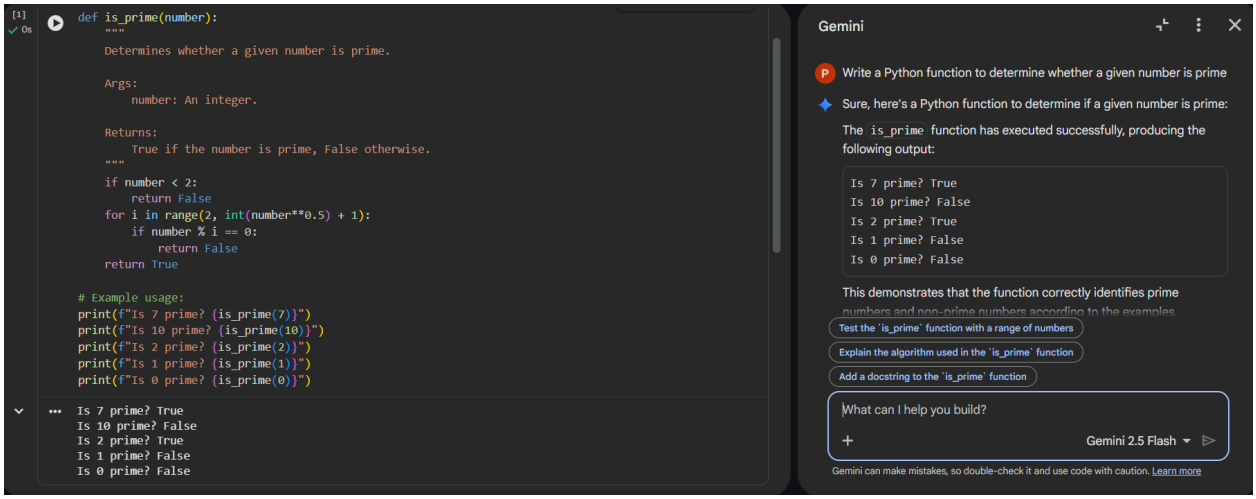
Name: P.Swaran Raj **Roll.No:** 2303A52247
Batch: 37 **Assignment-** 4

Task Description 1:

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

Prompt used:

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



Explanation:

The AI generated a prime-checking function using only instructions.
It checks numbers less than or equal to 1 and verifies divisibility up to the square root of the number.

Output:

Is 7 prime? True
Is 10 prime? False

Is 2 prime? True
Is 1 prime? False
Is 0 prime? False

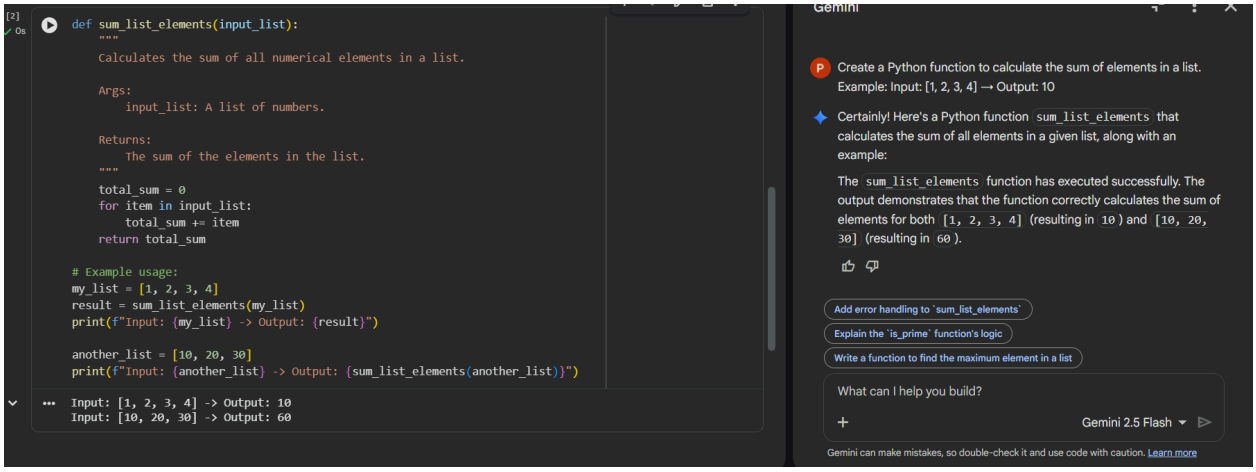
Observation:

The AI correctly generated logic without any examples.
Zero-shot prompting works well for simple and well-defined problems.

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.

Prompt used:

"#Create a Python function to calculate the sum of elements in a list. Example: Input: [1, 2, 3, 4] → Output: 10"



Explanation:

The function initializes a total variable and iterates through the list. Each element is added to the total and the final sum is returned.

Output:

Input: [1, 2, 3, 4] -> Output: 10

Input: [10, 20, 30] -> Output: 60

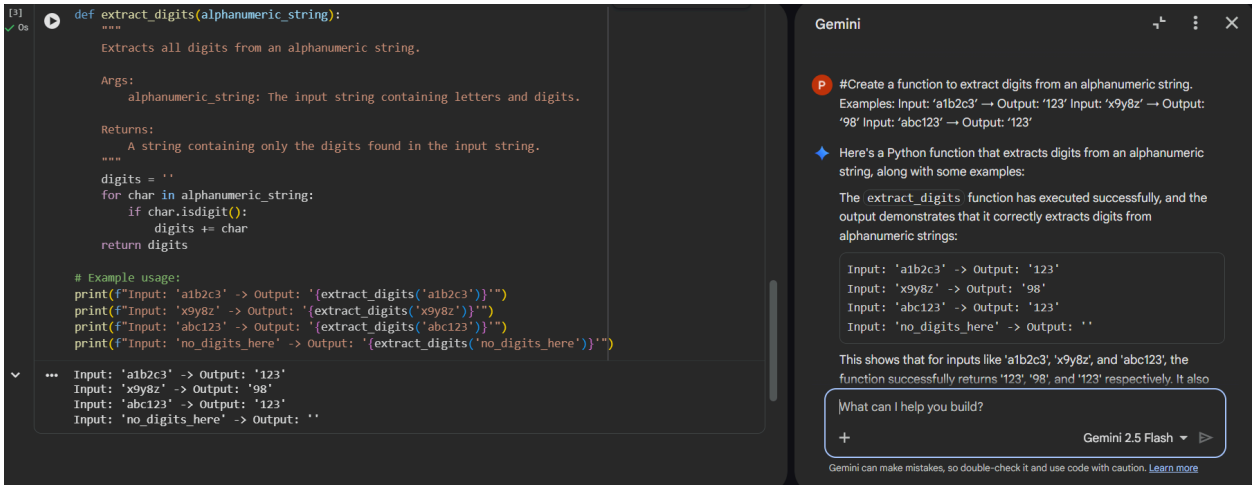
Observation:

Providing one example clearly guided the AI. One-shot prompting improved clarity and correctness.

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

Prompt used:

“#Create a function to extract digits from an alphanumeric string.
Examples:
Input: ‘a1b2c3’ → Output: ‘123’
Input: ‘x9y8z’ → Output: ‘98’
Input: ‘abc123’ → Output: ‘123’”



Explanation:

The AI generated a function that loops through the string and checks digits using isdigit(). Digits are appended to the result string and returned.

Output:

Input: 'a1b2c3' -> Output: '123'
Input: 'x9y8z' -> Output: '98'
Input: 'abc123' -> Output: '123'

Observation:

Multiple examples improved accuracy and logic understanding.
Few-shot prompting produced more reliable code.

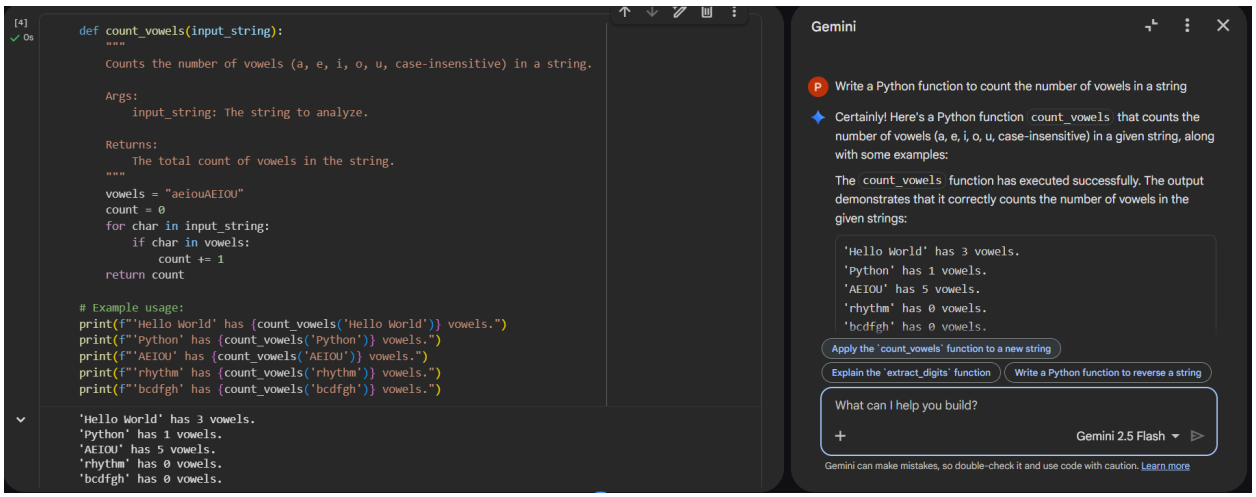
Task Description 4:

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

Problem: Generate a function to count the number of vowels in a string.

Zero-Shot prompting for generating a function that counts the number of vowels in a string.:

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



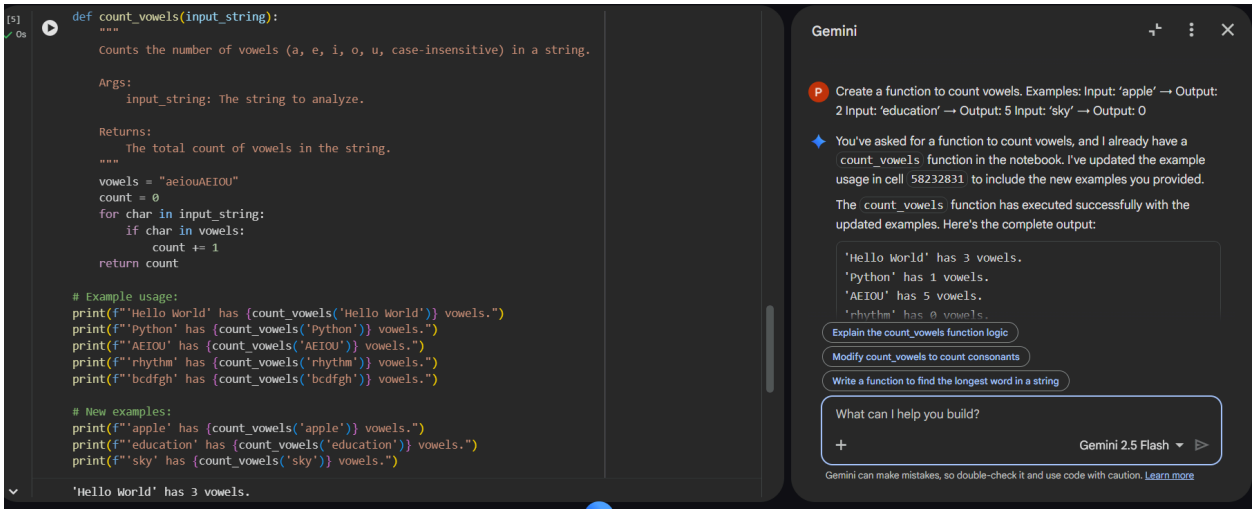
Output:

'Hello World' has 3 vowels.

'Python' has 1 vowels.
'AEIOU' has 5 vowels.
'rhythm' has 0 vowels.
'bcdfgh' has 0 vowels.

Few-shot prompting for generating a function that counts the number of vowels in a string:

“Create a function to count vowels.
Examples:
Input: ‘apple’ → Output: 2
Input: ‘education’ → Output: 5
Input: ‘sky’ → Output: 0”



Output:

'Hello World' has 3 vowels.
'Python' has 1 vowels.
'AEIOU' has 5 vowels.
'rhythm' has 0 vowels.
'bcdfgh' has 0 vowels.
'apple' has 2 vowels.
'education' has 5 vowels.
'sky' has 0 vowels.

Comparison & Explanation:

Aspect	Zero-Shot	Few-Shot
Clarity	Basic logic	More optimized
Accuracy	Correct	Correct

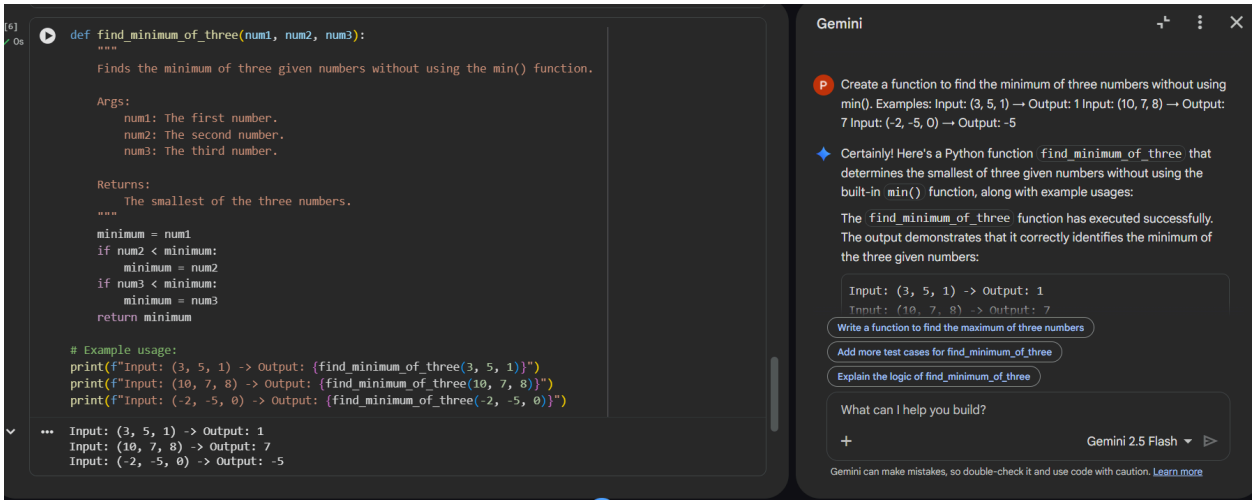
Observation:

Few-shot prompting produced clearer and optimized logic.
Examples significantly improved AI understanding.

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.

Prompt used:

"#Create a function to find the minimum of three numbers without using `min()`.
Examples:
Input: (3, 5, 1) → Output: 1
Input: (10, 7, 8) → Output: 7
Input: (-2, -5, 0) → Output: -5"



Explanation:

The function assumes the first number is smallest and compares it with others. The smallest value is updated and returned.

Output:

Input: (3, 5, 1) -> Output: 1
Input: (10, 7, 8) -> Output: 7
Input: (-2, -5, 0) -> Output: -5

Observation:

Few-shot examples helped the AI infer comparison logic correctly.
The solution avoided built-in functions as instructed.

Conclusion:

• Zero-shot prompting works well for simple tasks but may lack optimization.
• One-shot prompting improves clarity using examples.
• Few-shot prompting provides the best accuracy, structure, and optimized solutions.