Task #1 – Zero-Shot Prompting with Conditional Validation

Objective

Use zero-shot prompting to instruct an AI tool to generate a function

that validates an Indian mobile number.

Requirements

 The function must ensure the mobile number:

o Starts with 6, 7, 8, or 9

o Contains exactly 10 digits

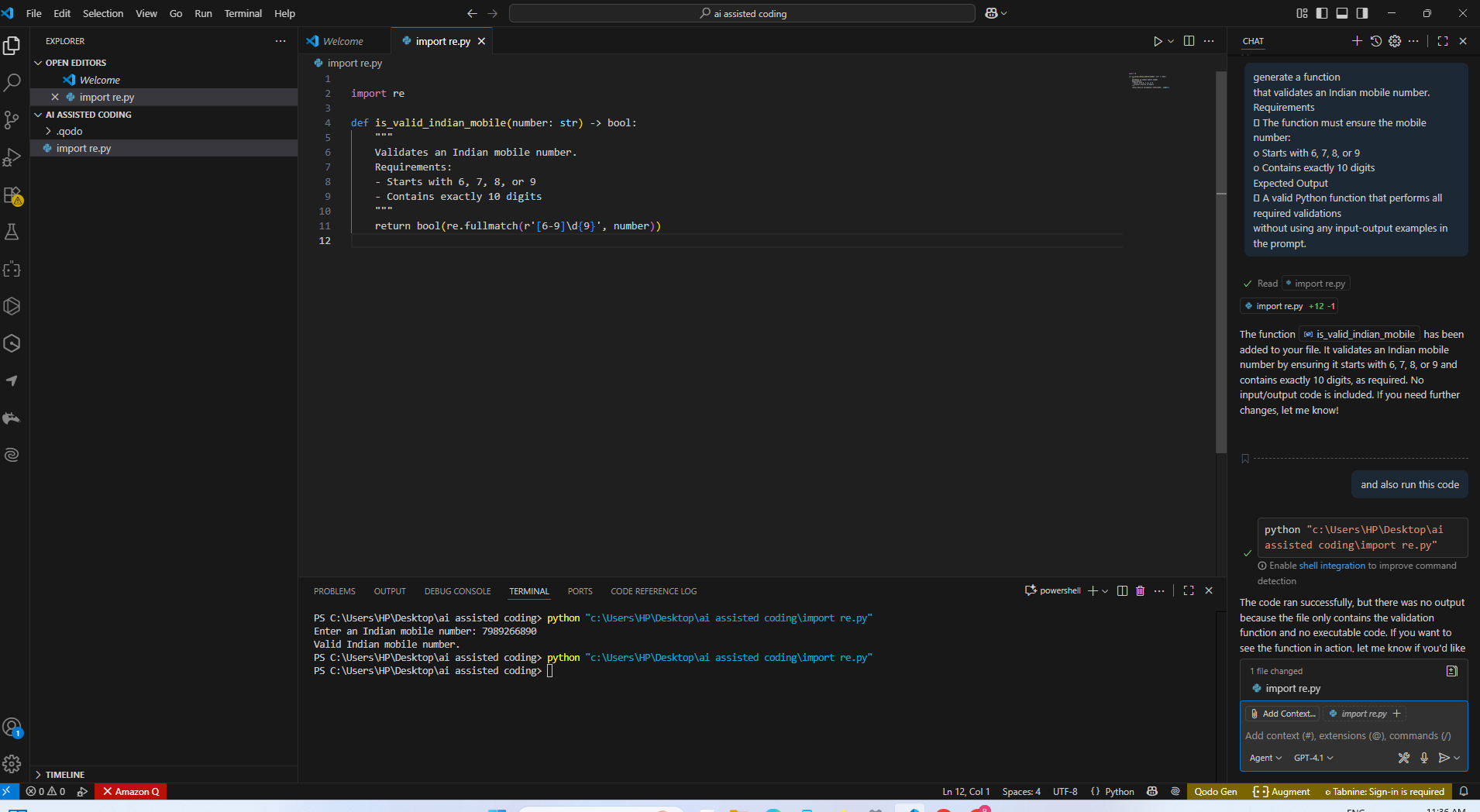
Expected Output

 A valid Python function that performs all required validations

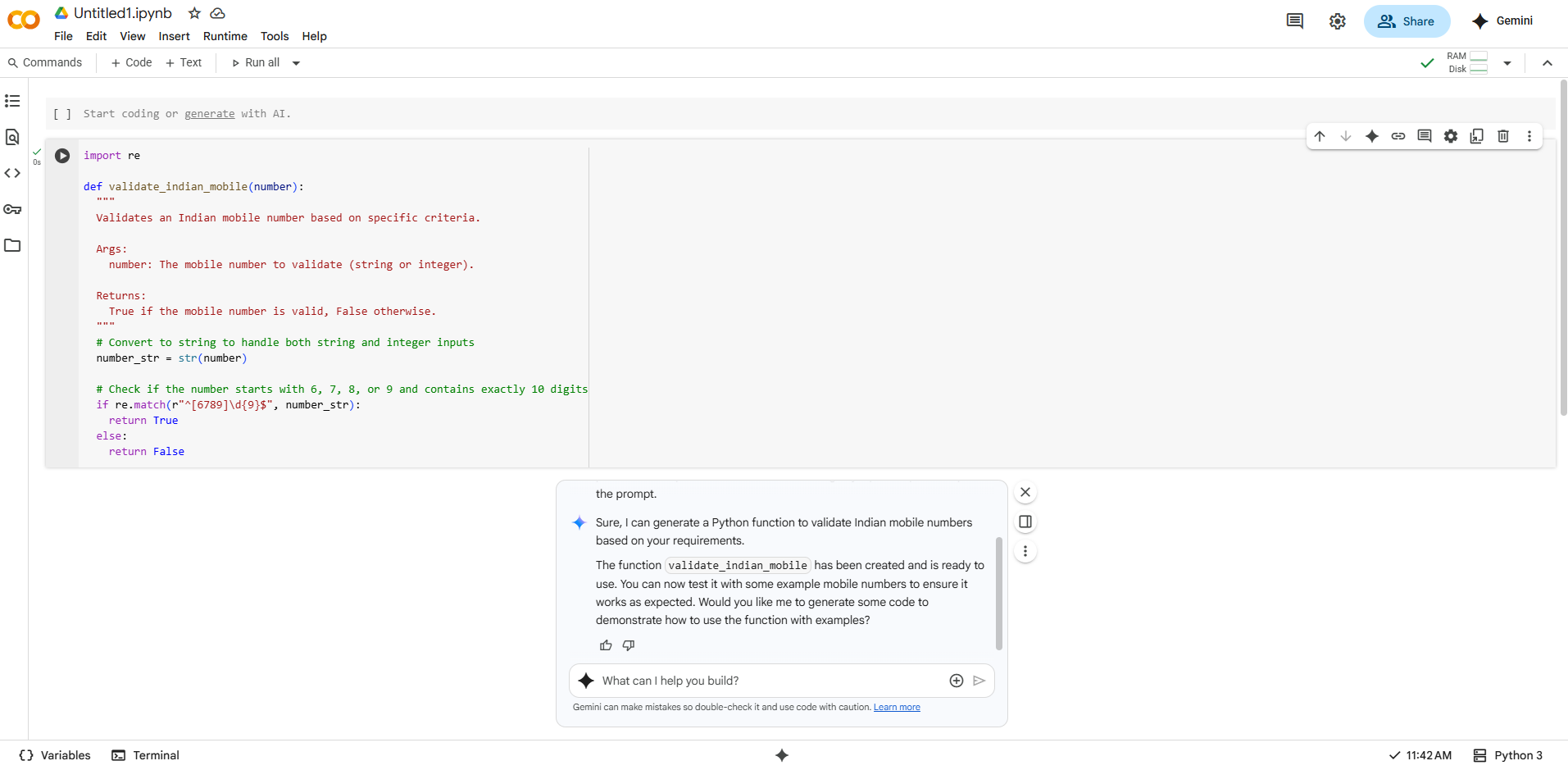
without using any input-output examples in the prompt.

ANS :

It has not given any output coz the file contains validation function not the executable code



ANS2: WHERE AS IN GOOGLE COLAB ITS ASKING TO GIVE ANY EXAMPLE TEST CASES BEFORE EXECUTING



Task #2 – One-Shot Prompting with Edge Case Handling

Objective

Use one-shot prompting to generate a Python function that calculates

the factorial of a number.

Requirements

 Provide one sample input-output pair in the prompt to guide the

AI.

 The function should handle:

o 0! correctly

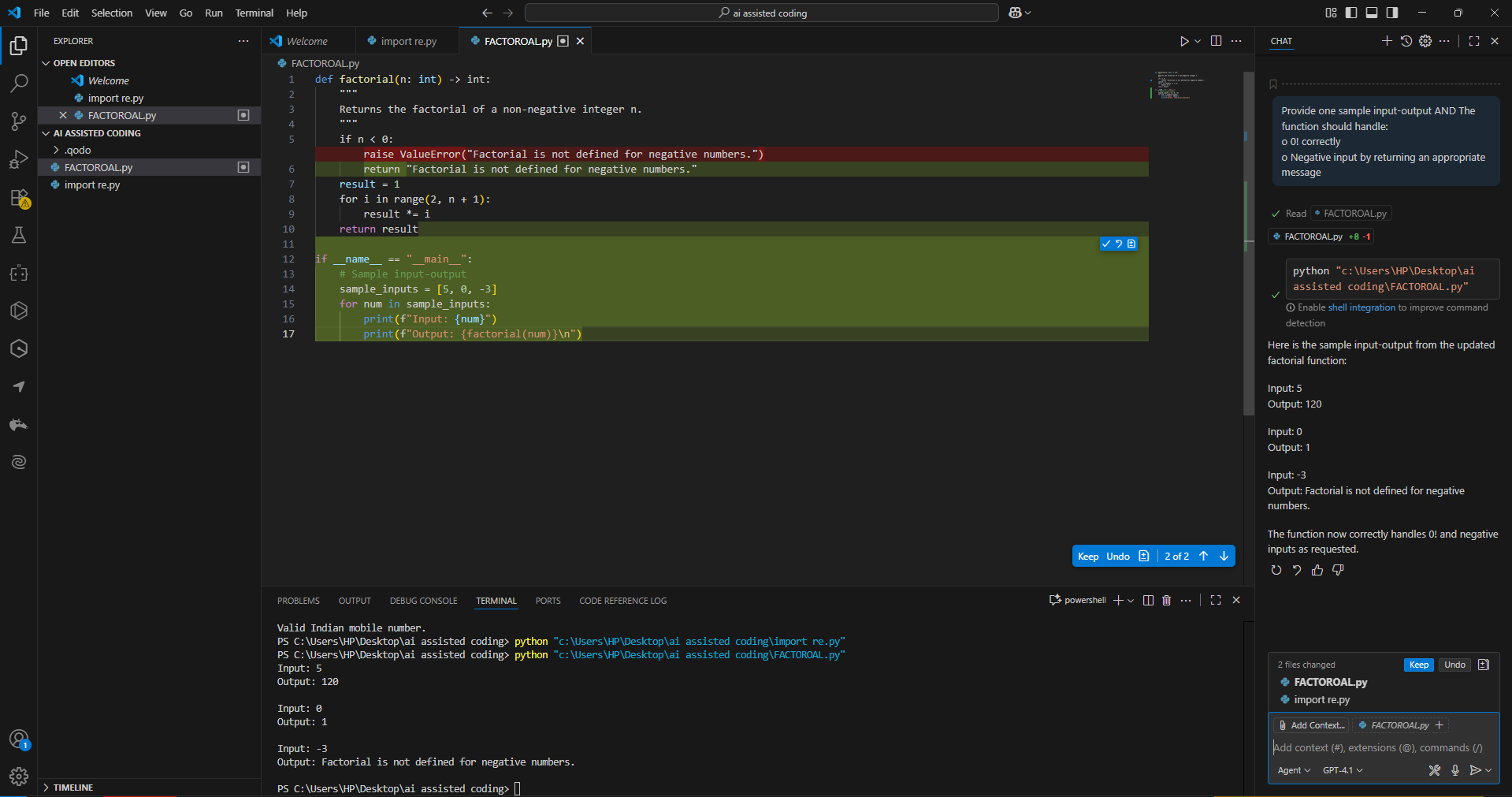
o Negative input by returning an appropriate message

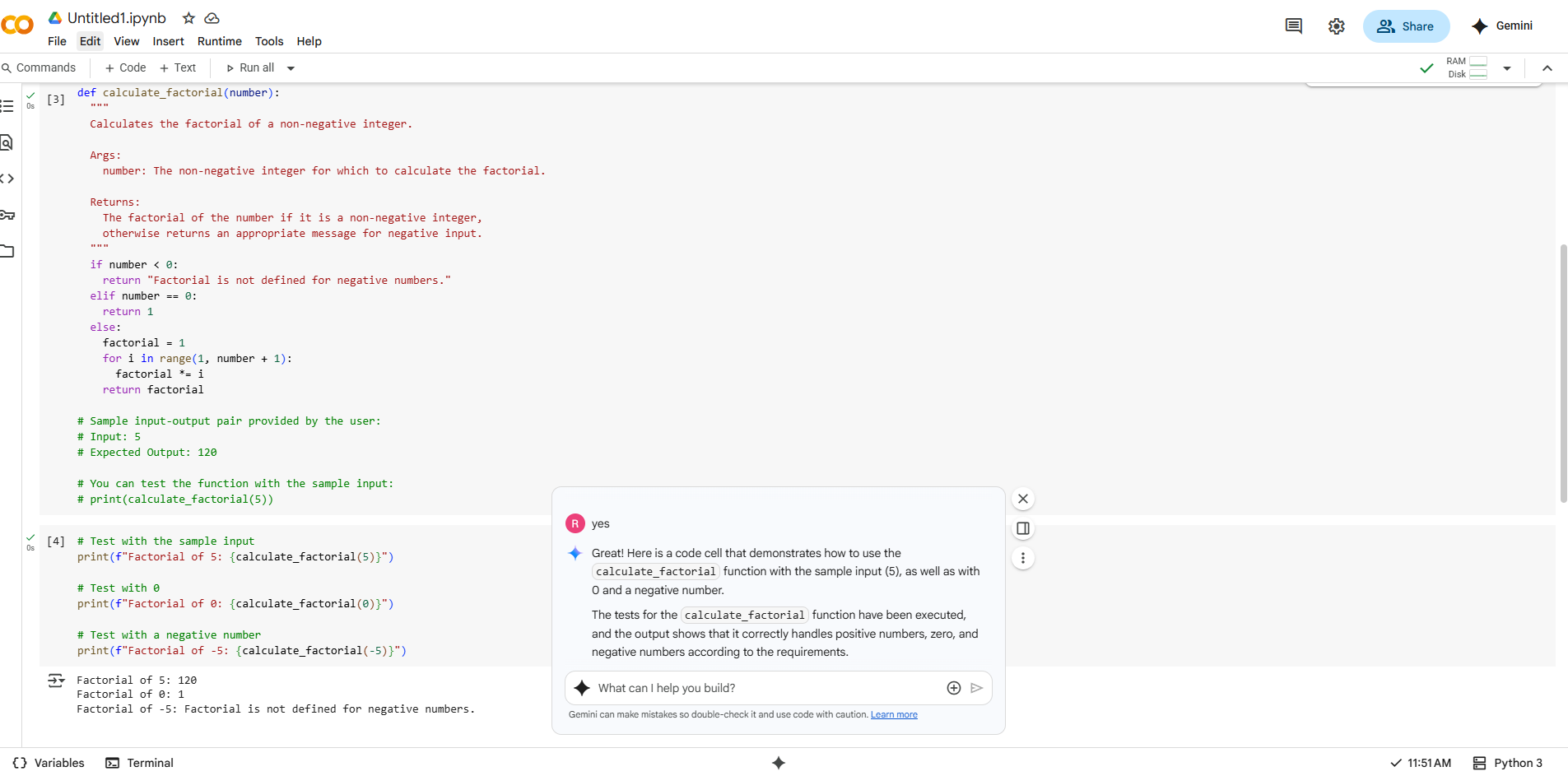
Expected Output

 A Python function with correct factorial logic and edge case

handling, generated from a single example.

solution: in vs code its giving the excact output as given in the prompt

****

ans 2: ****in google colab its asking another time after giving the prompt in a perfect way of prompting

Task #3 – Few-Shot Prompting for Nested Dictionary Extraction

Use few-shot prompting (2–3 examples) to instruct the AI to create a

function that parses a nested dictionary representing student

information.

Requirements

 The function should extract and return:

o Full Name

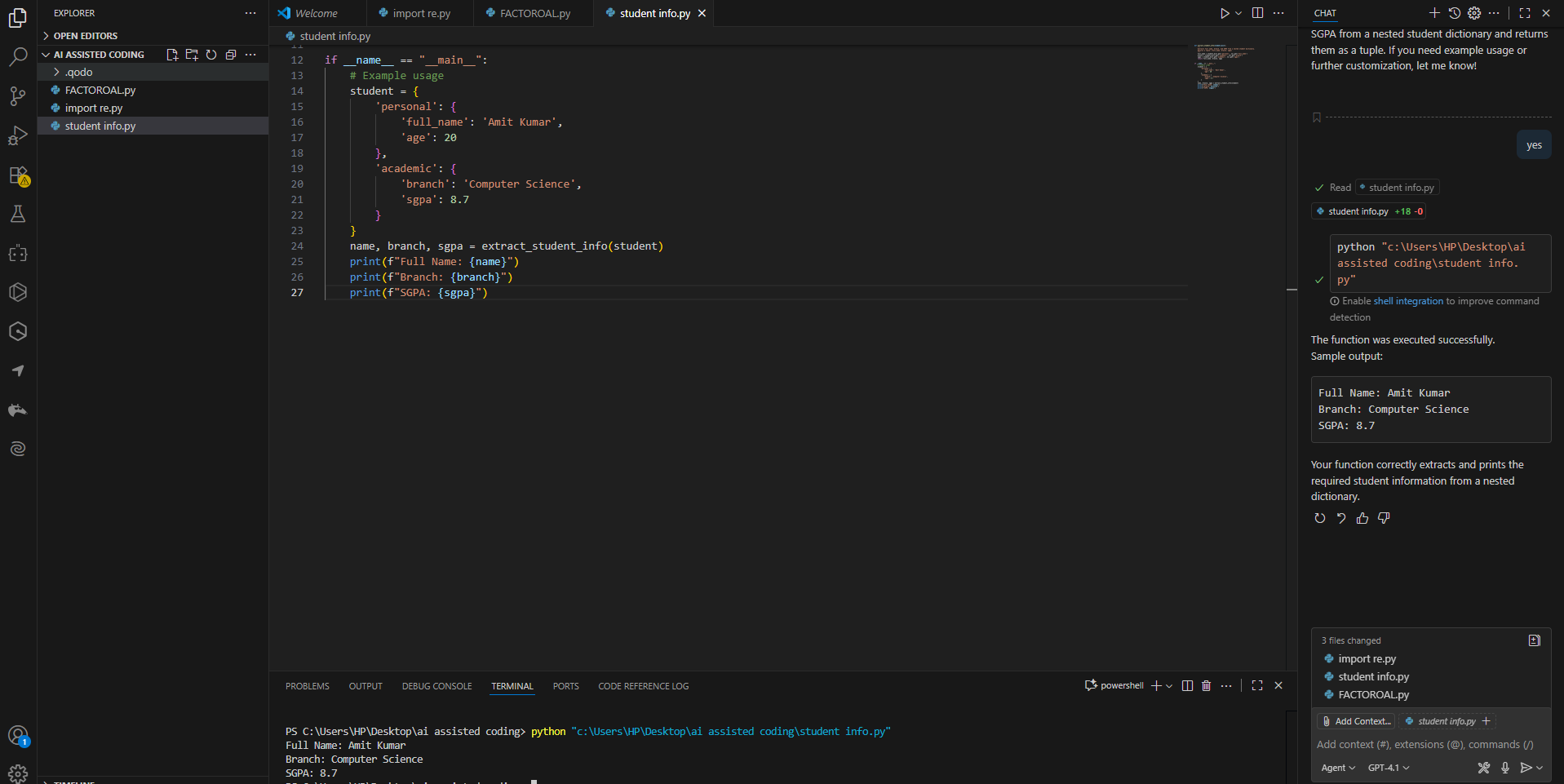
o Branch

o SGPA

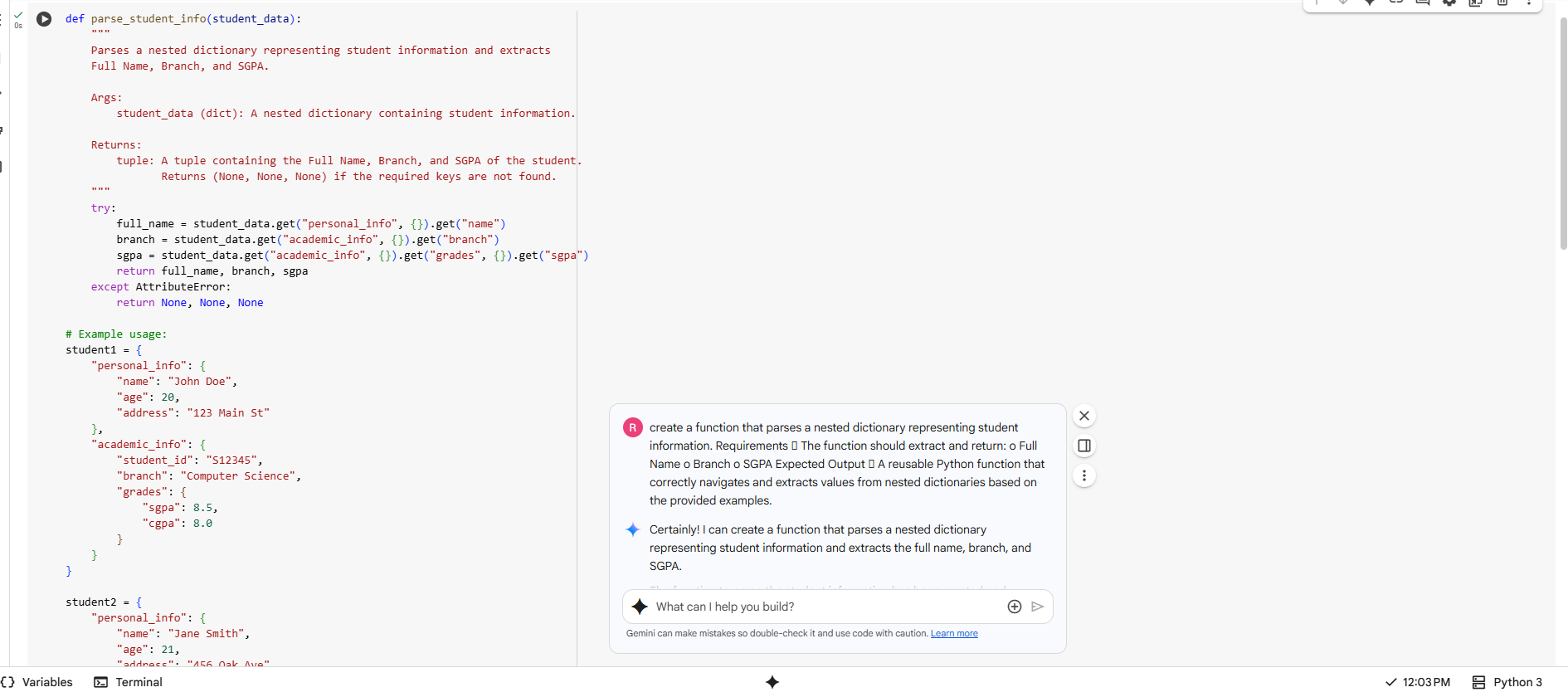
Expected Output

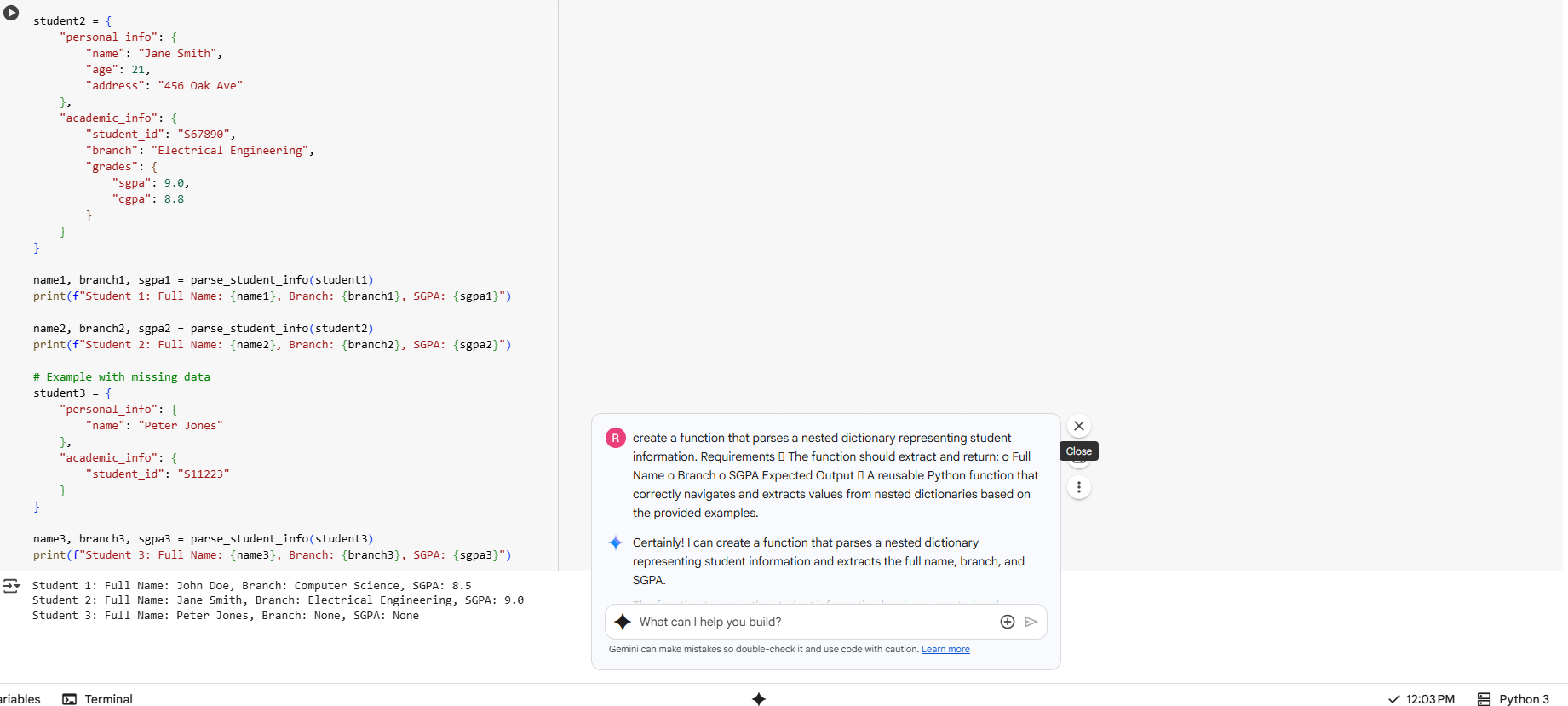
 A reusable Python function that correctly navigates and extracts

values from nested dictionaries based on the provided examples.

solution 1: in vs code its asking for another time weather to give example output even after giving the prompt saying that: give me output with an example to check weather the code is working or not ****

solution 2: in google colab after giving the prompt it directly gave code with an input and output example

****

****

Task #4 – Comparing Prompting Styles for File Analysis

Objective

Experiment with zero-shot, one-shot, and few-shot prompting to

generate functions for CSV file analysis.

Requirements

 Each generated function should:

o Read a .csv file

o Return the total number of rows

o Count the number of empty rows

o Count the number of words across the file

Expected Output

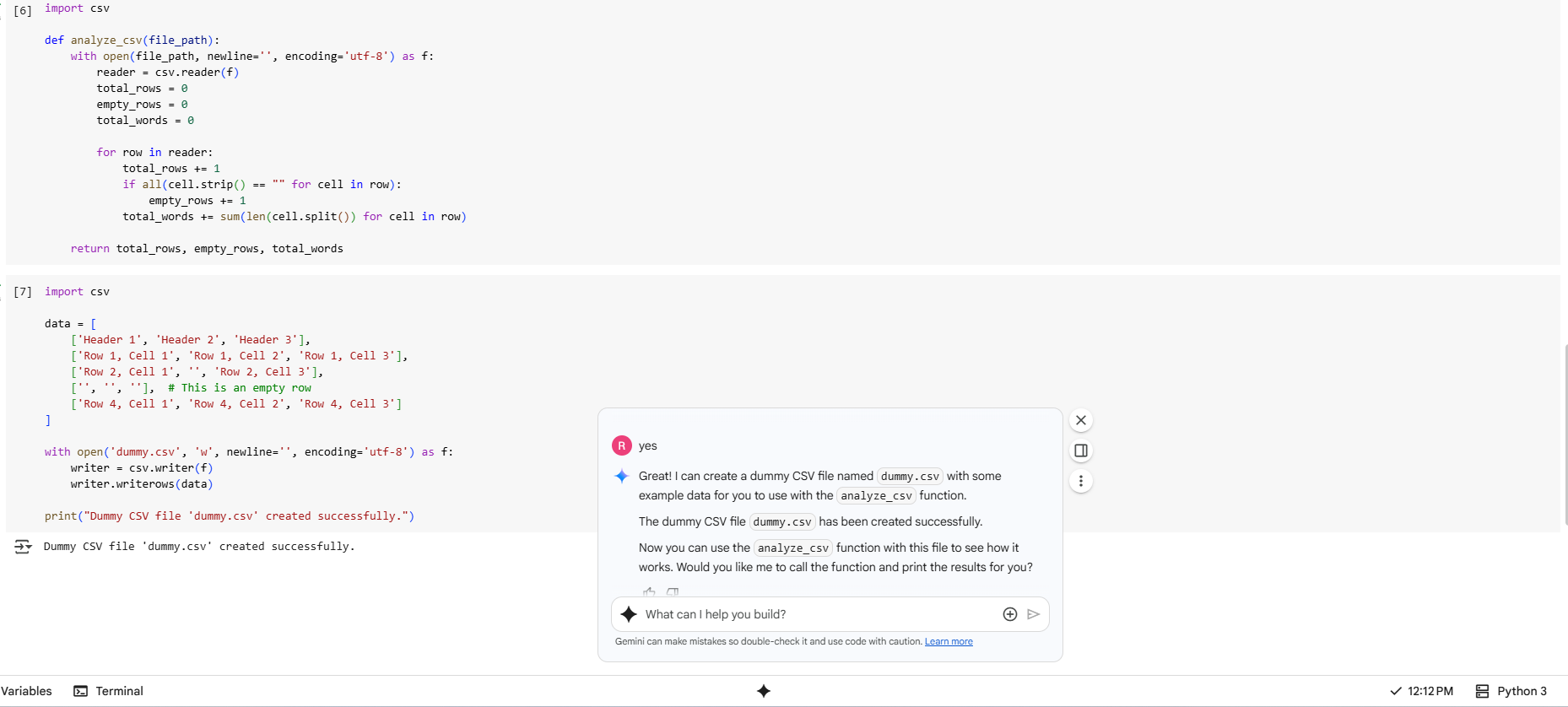
 Working Python functions for each prompting style, with a brief

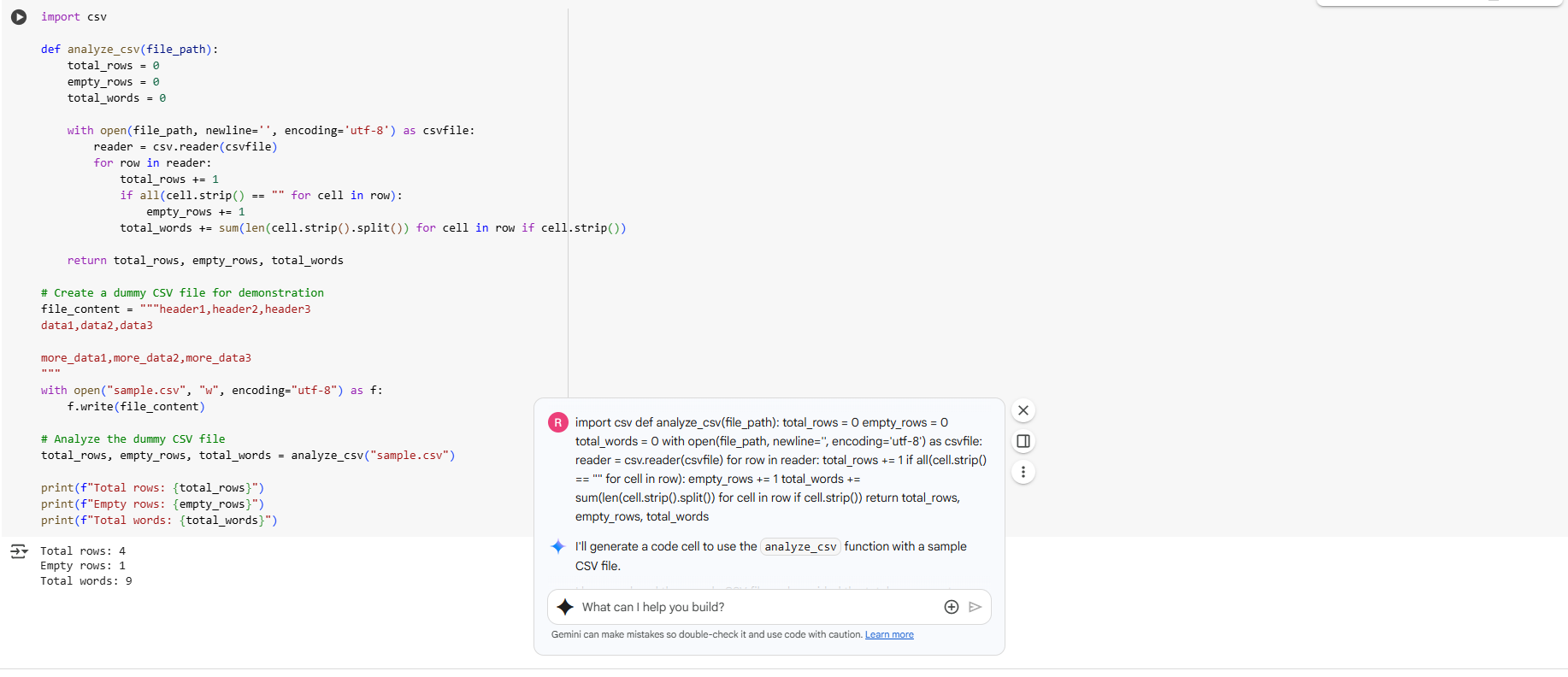
reflection comparing their accuracy, clarity, and efficiency.

solution : zero shot prompt:- Write a Python function that reads a CSV file and returns:

1. Total number of rows
2. Number of empty rows (all cells empty)
3. Total number of words across all cells.

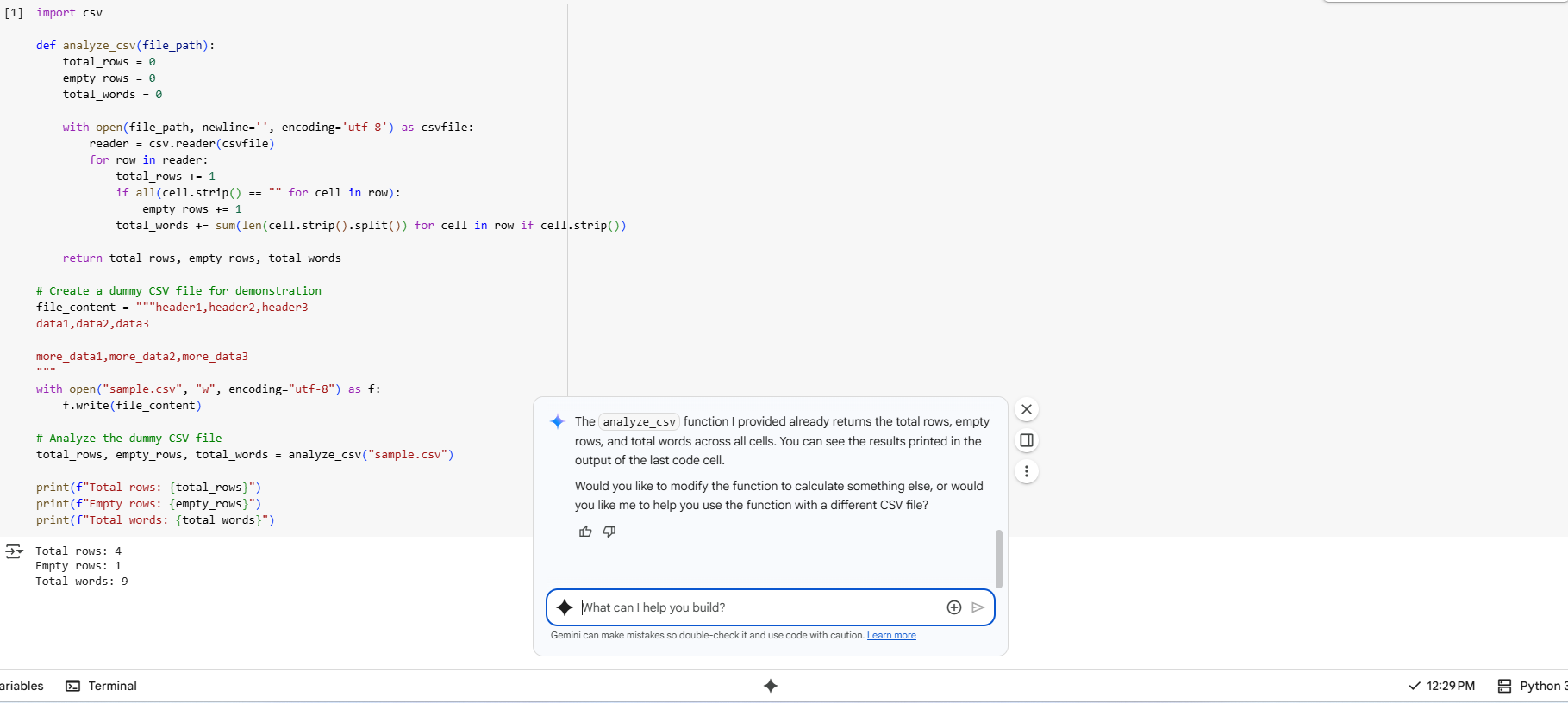
With few lines of code and by creating dummy csv file , successfully the dummy.csv created

****

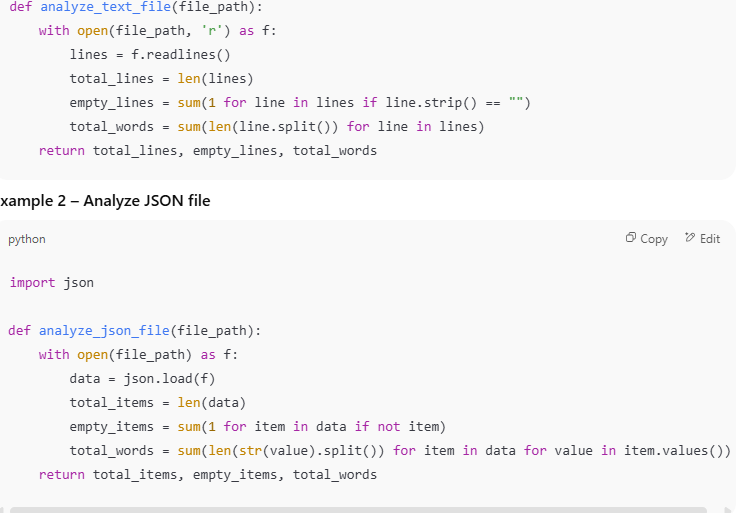
****

one shot prompting : Now write a similar function for a CSV file that returns:

* total rows
* empty rows
* total words across all cells
* its asking to modify because the output is same, but the prompting is different

****

****

few shot prompting:- its giving short and different codes ****

Task #5 – Few-Shot Prompting for Text Processing and Word

Frequency

Objective

Use few-shot prompting (with at least 3 examples) to generate a

Python function that processes text and analyzes word frequency.

Requirements

The function must:

 Accept a paragraph as input

 Convert all text to lowercase

 Remove punctuation

 Return the most frequently used word

Expected Output

 A functional Python script that performs text cleaning,

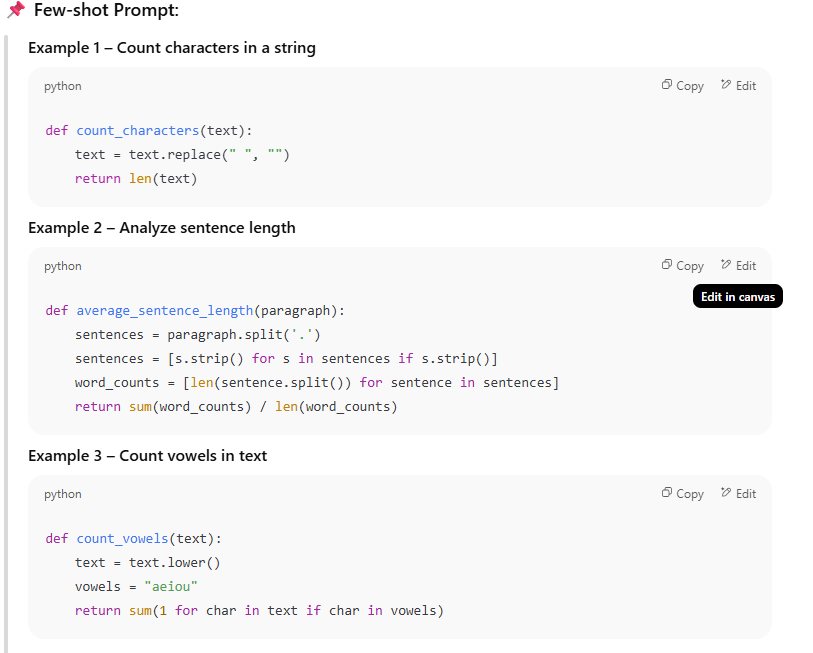
tokenization, and returns the most common word using only the

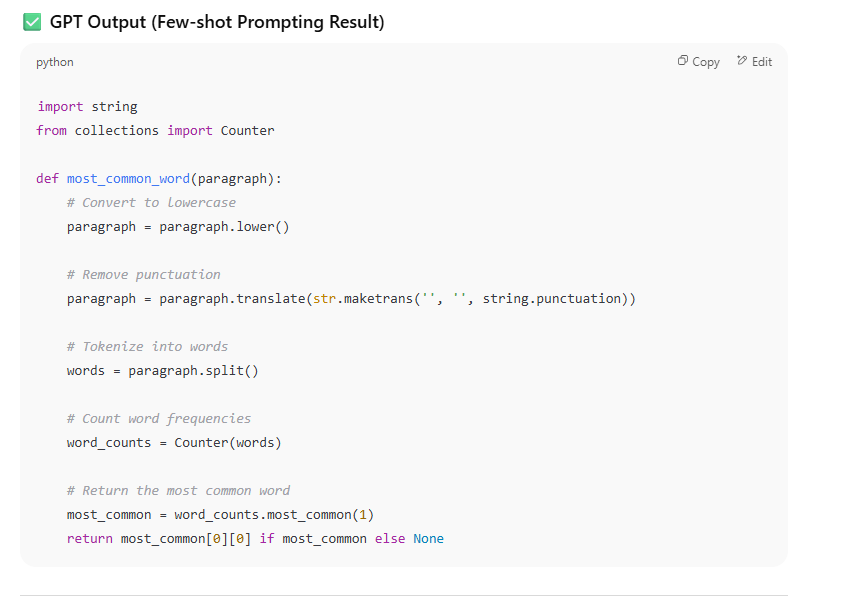
examples provided in the prompt

solution: few shot prompting :-  **str.maketrans('', '', string.punctuation)** efficiently strips all punctuation.

 **Counter(words).most\_common(1)** returns the most frequent word.

 The function is safe for empty input and handles edge cases (e.g. returns None if the paragraph is empty).

****

****

