**HALL TICKET NUMBER: 2403A51365**

**NAME: RAYAPUDI HARITHA**

**BATCH: 24BTCAICSB14**

**AssignmentNumber:4.3**

**Lab 4: Advanced Prompt Engineering – Zero-shot, One-shot, and Few-shot Techniques**

**Lab Objectives:**

* **To explore and apply different levels of prompt examples in AI-assisted code generation.**
* **To understand how zero-shot, one-shot, and few-shot prompting affect AI output quality.**
* **To evaluate the impact of context richness and example quantity on AI performance.**
* **To build awareness of prompt strategy effectiveness for different problem types.**

**Lab Outcomes (LOs):**

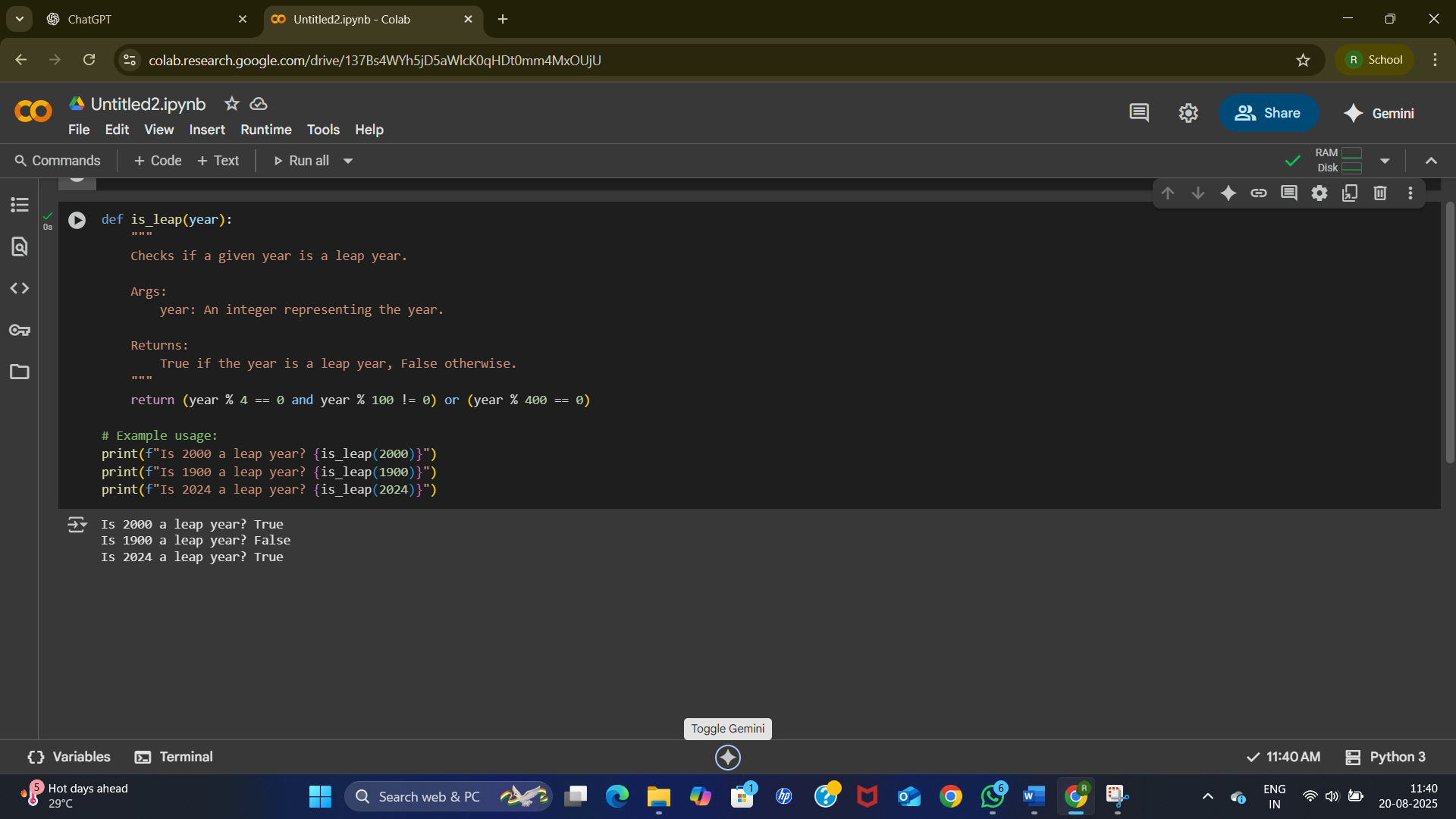
**After completing this lab, students will be able to:**

* **Use zero-shot prompting to instruct AI with minimal context.**
* **Use one-shot prompting with a single example to guide AI code generation.**
* **Apply few-shot prompting using multiple examples to improve AI responses.**
* **Compare AI outputs across the three prompting strategies.**

**Task Description#1**

**Prompt:-Zero-shot: Prompt AI to write a function that checks whether a given year is a leap year.**

**Code and output:-**

****

**code explanation:-**

**The code defines a function is\_leap that determines if a year is a leap year. It follows the rules: a year is a leap year if it's divisible by 4 but not by 100, unless it's also divisible by 400. The example usage then demonstrates how to call this function for the years 2000, 1900, and 2024, printing the result for each.**

**Task Description#2**

**Prompt:-One-shot: Give one input-output example to guide AI in writing a function that converts centimeters to inches.**

**Code and output:-**

A screenshot of a computer

AI-generated content may be incorrect.

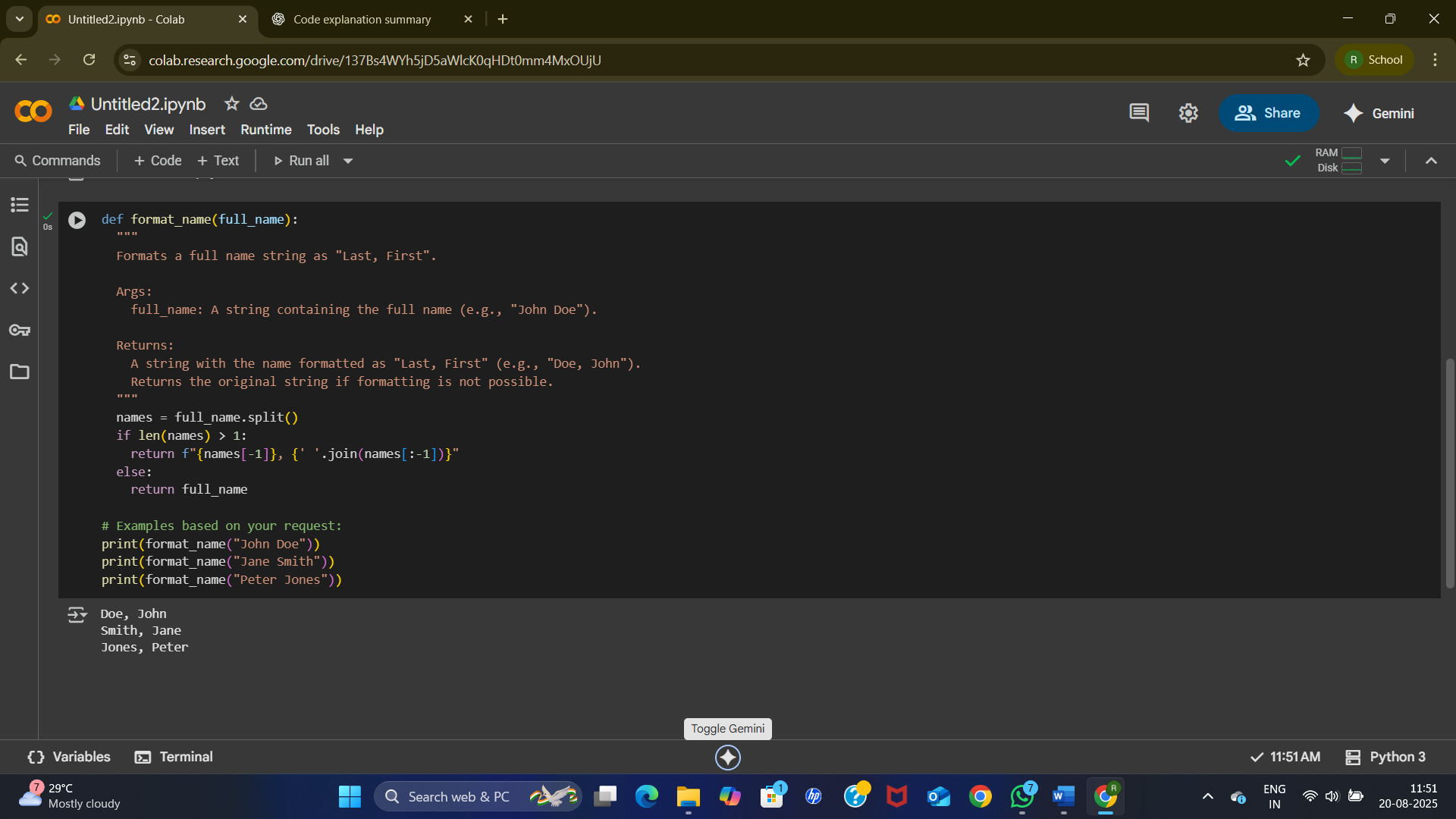
**code explanation:-**

This code defines a function **cm\_to\_inches(cm)** that converts a value in centimeters to inches using the conversion factor 1 cm = 0.393701 inches.  
It then demonstrates the function by converting 10 cm into inches.  
The result is stored in the variable inches and printed using an **f-string** with formatting up to 2 decimal places.  
Finally, it outputs: **"10 centimeters is equal to 3.94 inches."**

**Task Description#3**

Prompt:-Few-shot: Provide 2–3 examples to generate a function that formats full names as “Last, First”.

Code and output:-



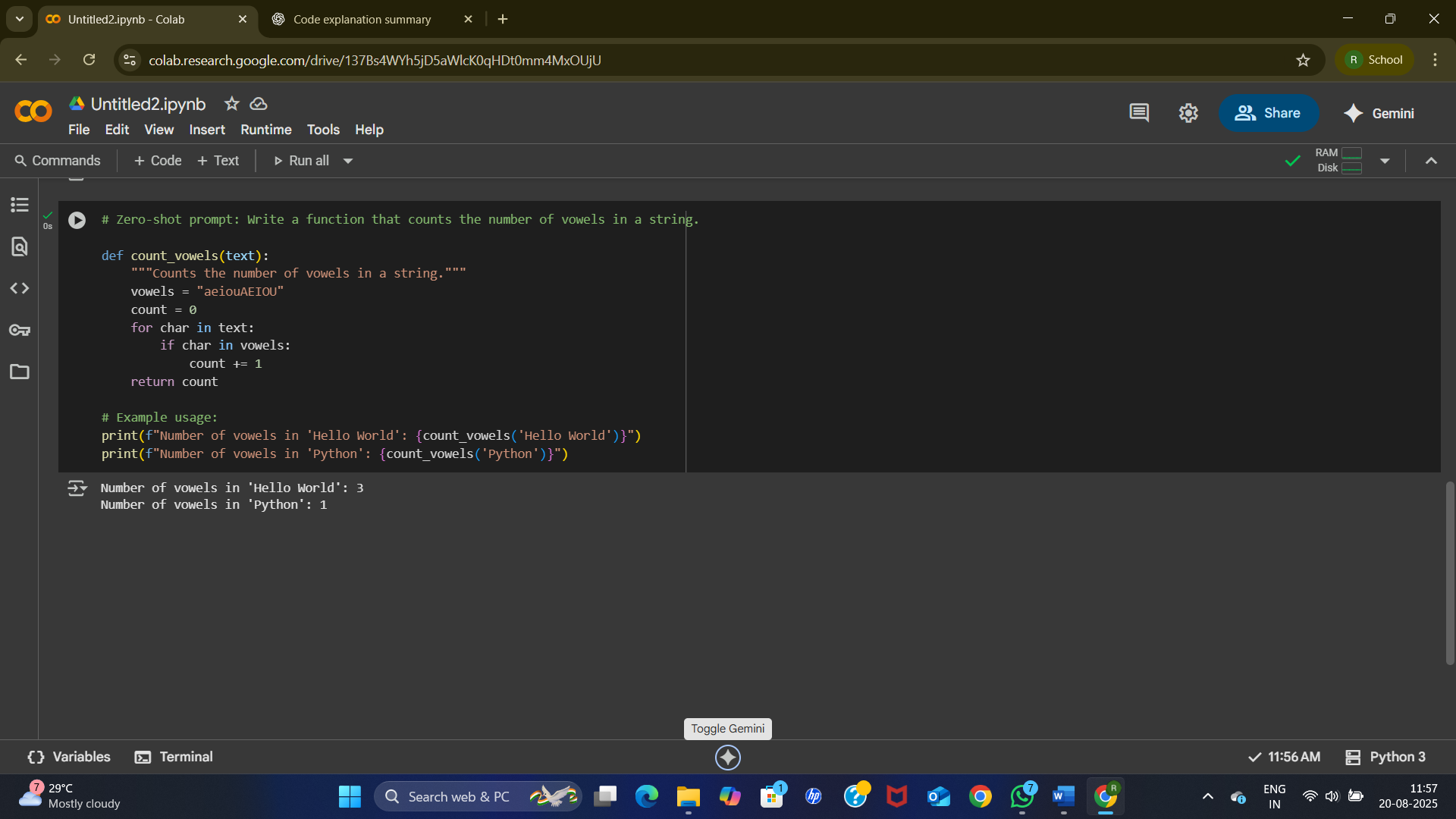
**code explanation:-**

This code defines a function format\_name that takes a full name string. It splits the name into parts, and if there's more than one part, it rearranges them to put the last name first, followed by a comma and the rest of the name. If the name can't be split (e.g., only a single name), it returns the original input. The example usage shows how it works for "John Doe", "Jane Smith", and "Peter Jones".

**Task Description#4**

Prompt:-Compare zero-shot and few-shot prompts for writing a function that counts the number of vowels in a string.

Code and output:-



**code explanation:-**

This code defines a function count\_vowels that takes a string as input. It initializes a counter and a string containing all vowels (both lowercase and uppercase). It then iterates through each character in the input string, and if the character is found within the vowels string, the counter is incremented. Finally, it returns the total count of vowels. The example usage demonstrates calling the function with "Hello World" and "Python".

**Comparison Table:-**

| Feature | Zero-Shot Learning | Few-Shot Learning |
| --- | --- | --- |
| **Number of Examples** | Zero examples are provided in the prompt. The model relies entirely on its pre-trained knowledge. | A small number of examples (typically 1 to 5) are provided in the prompt. |
| **Prompt Structure** | The prompt consists of a direct instruction or question. ➡️ **Example:** "Translate the following sentence into French: 'Hello, how are you?'" | The prompt includes examples of input-output pairs before the final instruction. ➡️ **Example:** "Translate the following sentences into French. 'Dog' -> 'Chien' 'Cat' -> 'Chat' 'Hello, how are you?' ->" |
| **Performance** | Performance can be less reliable, especially for complex or highly specific tasks. It is best for simple, straightforward tasks. | Performance is generally better and more reliable, as the examples provide the model with a clear pattern to follow. |
| **Purpose** | Used to test a model's foundational knowledge and capability on a given task without any explicit guidance. | Used to guide a model to learn a specific pattern, style, or format for a task, which it might not know from its training data. |
| **Efficiency & Cost** | More efficient and lower cost, as the prompt is shorter and requires fewer tokens. | Less efficient and more costly, as the prompt is longer and requires more tokens. |
| **Use Cases** | Basic tasks like simple translation, summarization of a common topic, or straightforward question-answering. | Complex tasks that require a specific output format, style, or nuanced understanding, such as sentiment analysis with specific labels, custom data extraction, or code generation with a particular structure. |
| **Analogy** | Like asking a child to draw a dog without showing them a picture, based only on their general knowledge of what a dog is. | Like asking a child to draw a dog after showing them a few pictures of different dogs to get the idea. |

**Task Description#5**

Prompt:- Use few-shot prompting to generate a function that reads a .txt file and returns the number of lines.

# Few-shot prompt: Write a function to count lines in a text file.

# Example 1:

# Input: "file.txt" containing:

# Line 1

# Line 2

# Output: 2

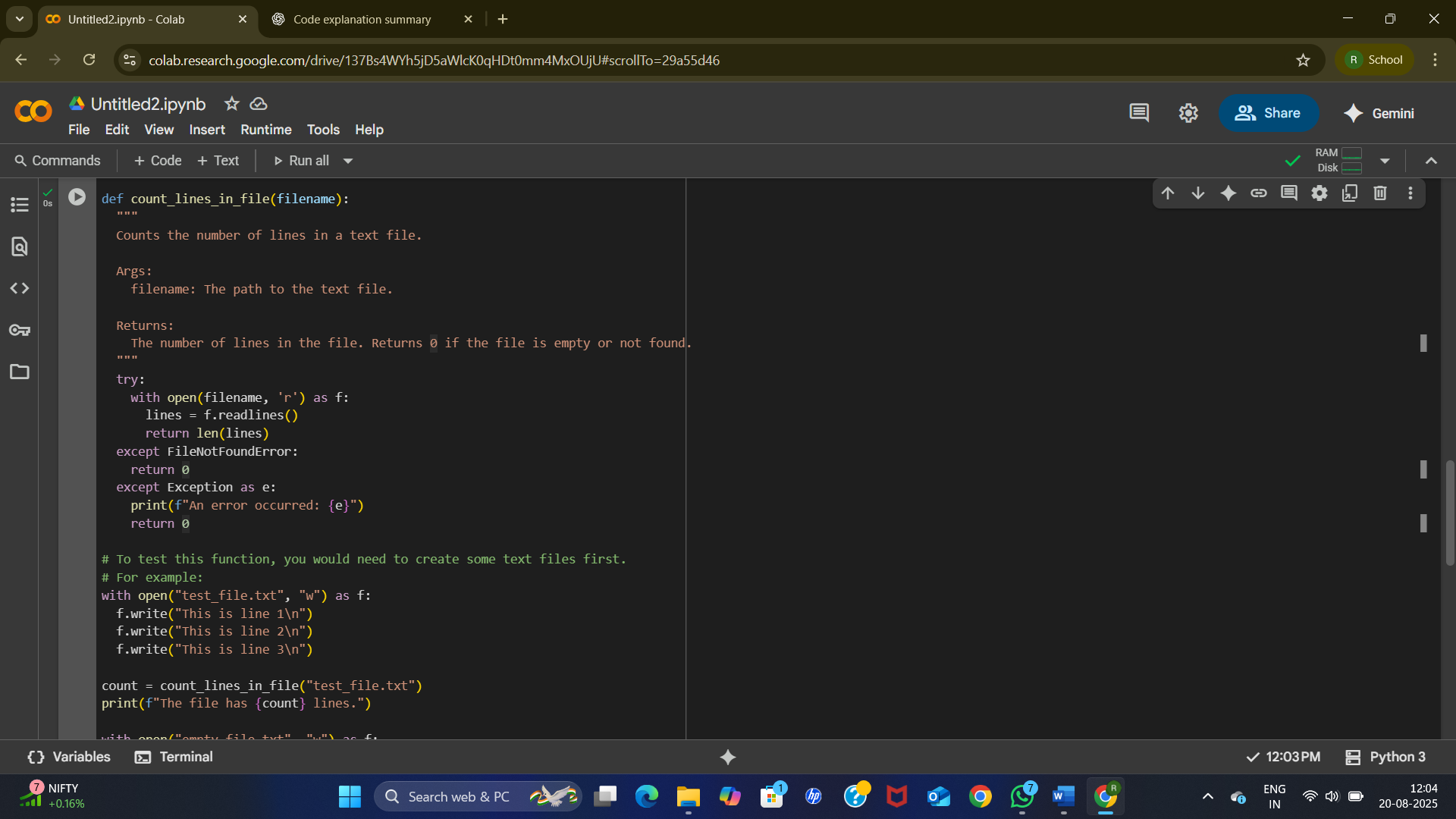
# Example 2:

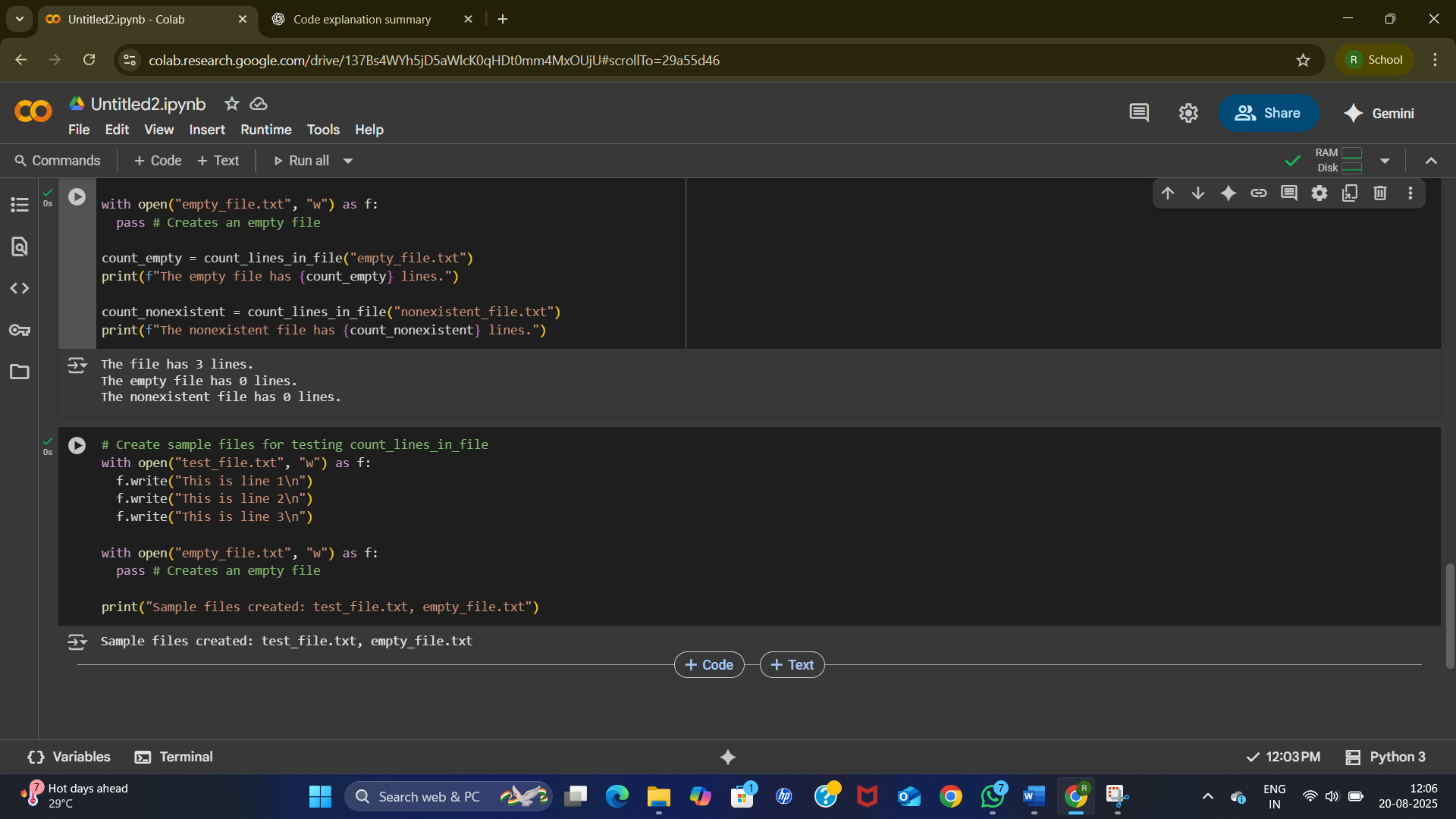
# Input: "empty.txt" containing:

# (empty file)

# Output: 0

Code and output;-

****



**code explanation:-**

 This code defines a function count\_lines\_in\_file that takes a filename and attempts to open and read it. It uses a try...except block to handle potential FileNotFoundError. If the file is opened successfully, it reads all lines using readlines() and returns the count of lines. If the file is not found, it returns 0. The example usage demonstrates this by creating and testing with a sample file, an empty file, and a non-existent file.