Name:
k.lokesh

roll no:
2403A53006

SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE	**DEPARTMENT OF COMPUTER SCIENCE ENGINEERING**

ProgramName:B. Tech	**Assignment Type: Lab**	**AcademicYear:**2025-2026

CourseCoordinatorName	Venkataramana Veeramsetty
Instructor(s)Name	Dr. V. Venkataramana (Co-ordinator) Dr. T. Sampath Kumar Dr. Pramoda Patro Dr. Brij Kishor Tiwari Dr.J.Ravichander Dr. Mohammand Ali Shaik Dr. Anirodh Kumar Mr. S.Naresh Kumar Dr. RAJESH VELPULA Mr. Kundhan Kumar Ms. Ch.Rajitha Mr. M Prakash Mr. B.Raju Intern 1 (Dharma teja) Intern 2 (Sai Prasad) Intern 3 (Sowmya) NS_2 (Mounika)

CourseCode	24CS002PC215	**CourseTitle**	AI Assisted Coding
Year/Sem	II/I	**Regulation**	R24
Date and Day of Assignment	Week2 - Wednesday	**Time(s)**	
Duration	2 Hours	**Applicableto Batches**	

AssignmentNumber:2.3(Present assignment number)/**24**(Total number of assignments)

Q.No.	**Question**	***ExpectedTime to complete***
1	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.	Week2 - Wednesday

- 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

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

Expected Output#1

- AI-generated function with no examples provided

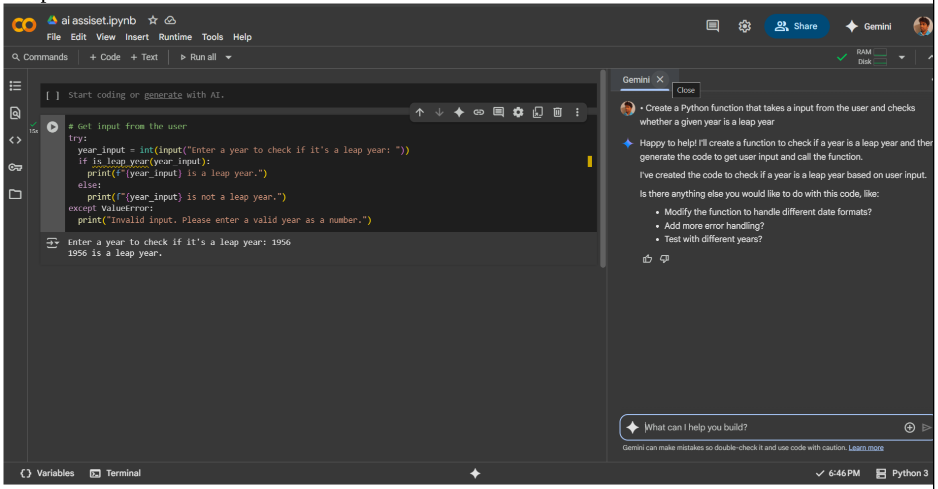
Prompt:

Create a Python program that takes a input from the user and checks whether a given year is a leap year

Code:

```
# Get input from the user
try:
  year_input = int(input("Enter a year to check if it's a leap year: "))
  if is_leap_year(year_input):
    print(f"{year_input} is a leap year.")
  else:
    print(f"{year_input} is not a leap year.")
except ValueError:
  print("Invalid input. Please enter a valid year as a number.")
```

out put:



```
[ ] Start coding or generate with AI.
```

```
# Get input from the user
try:
  year_input = int(input("Enter a year to check if it's a leap year: "))
  if is_leap_year(year_input):
    print(f"{year_input} is a leap year.")
  else:
    print(f"{year_input} is not a leap year.")
except ValueError:
  print("Invalid input. Please enter a valid year as a number.")
```

```
Enter a year to check if it's a leap year: 1956
1956 is a leap year.
```

• Create a Python function that takes a input from the user and checks whether a given year is a leap year

Happy to help! I'll create a function to check if a year is a leap year and ther generate the code to get user input and call the function.

I've created the code to check if a year is a leap year based on user input.

Is there anything else you would like to do with this code, like:

- Modify the function to handle different date formats?
- Add more error handling?
- Test with different years?

What can I help you build?

Gemini can make mistakes so double-check it and use code with caution. Learn more

Task Description#2

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

Expected Output#2

- Function with correct conversion logic

Prompt:

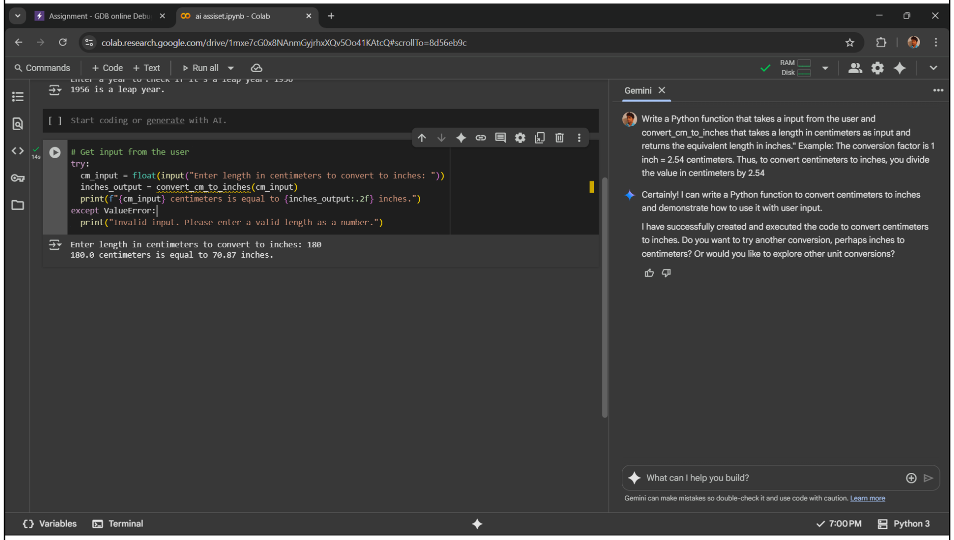
"Write a Python function that takes a input from the user and convert_cm_to_inches that

takes a length in centimeters as input and returns the equivalent length in inches."
Example:
The conversion factor is 1 inch = 2.54 centimeters. Thus, to convert centimeters to inches, you divide the value in centimeters by 2.54

Code:

```
        # Get input from the user
try:
  cm_input = float(input("Enter length in centimeters to convert to inches: "))
  inches_output = convert_cm_to_inches(cm_input)
  print(f"{cm_input} centimeters is equal to {inches_output:.2f} inches.")
except ValueError:
  print("Invalid input. Please enter a valid length as a number.")
```

out put:


Assignment - GDB online Debu
ai assiset.ipynb - Colab
colab.research.google.com/drive/1mxe7cG0x8NAnmGyjrhxXQv5Oo41KAtcQ#scrollTo=8d56eb9c
Commands
+ Code
+ Text
Run all
RAM
Disk
Gemini
Variables
Terminal
7:00 PM
Python 3

```
1956 is a leap year.
```

```
[ ] Start coding or generate with AI.
```

```
# Get input from the user
try:
  cm_input = float(input("Enter length in centimeters to convert to inches: "))
  inches_output = convert_cm_to_inches(cm_input)
  print(f"{cm_input} centimeters is equal to {inches_output:.2f} inches.")
except ValueError:
  print("Invalid input. Please enter a valid length as a number.")
```

```
Enter length in centimeters to convert to inches: 180
180.0 centimeters is equal to 70.87 inches.
```

Write a Python function that takes a input from the user and convert_cm_to_inches that takes a length in centimeters as input and returns the equivalent length in inches." Example: The conversion factor is 1 inch = 2.54 centimeters. Thus, to convert centimeters to inches, you divide the value in centimeters by 2.54

Certainly! I can write a Python function to convert centimeters to inches and demonstrate how to use it with user input.

I have successfully created and executed the code to convert centimeters to inches. Do you want to try another conversion, perhaps inches to centimeters? Or would you like to explore other unit conversions?

What can I help you build?

Gemini can make mistakes so double-check it and use code with caution. Learn more

Task Description#3

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

Expected Output#3

- Well-structured function respecting the examples

Prompt:

create a python function that formats full names as "Last, First".

Example 1

input: enter a name: Lokesh kamatham

output: last name = kamatham

First name =lokesh

Example 2

input: enter a name: Rithvik Gona

output: last name = Gona

First name = Rithvik

Example 3

input: enter a name: Nikhil thatikonda

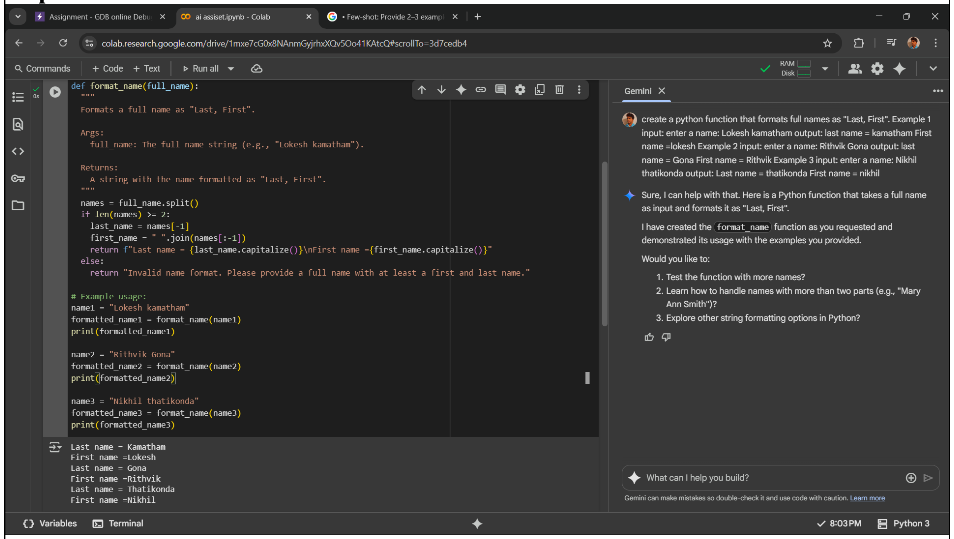
output: Last name = thatikonda

First name = Nikhil

Code:

```
        def format_name(full_name):
  """
  Formats a full name as "Last, First".

  Args:
    full_name: The full name string (e.g., "Lokesh kamatham").

  Returns:
    A string with the name formatted as "Last, First".
  """
  names = full_name.split()
  if len(names) >= 2:
    last_name = names[-1]
    first_name = " ".join(names[:-1])
    return f"Last name = {last_name.capitalize()}\nFirst name ={first_name.capitalize()}"
  else:
    return "Invalid name format. Please provide a full name with at least a first and last
name."

# Example usage:
name1 = "Lokesh kamatham"
formatted_name1 = format_name(name1)
print(formatted_name1)

name2 = "Rithvik Gona"
formatted_name2 = format_name(name2)
print(formatted_name2)

name3 = "Nikhil thatikonda"
formatted_name3 = format_name(name3)
print(formatted_name3)
```

output:


Assignment - GDB online Debu
ai assiset.ipynb - Colab
• Few-shot: Provide 2–3 exampl
colab.research.google.com/drive/1mxe7cG0x8NAnmGyjrhxXQv5Oo41KAtcQ#scrollTo=3d7cedb4
Commands
+ Code
+ Text
Run all
RAM
Disk
0s
Gemini
What can I help you build?
Variables
Terminal
8:03PM
Python 3

```
def format_name(full_name):
  """
  Formats a full name as "Last, First".

  Args:
    full_name: The full name string (e.g., "Lokesh kamatham").

  Returns:
    A string with the name formatted as "Last, First".
  """
  names = full_name.split()
  if len(names) >= 2:
    last_name = names[-1]
    first_name = " ".join(names[:-1])
    return f"Last name = {last_name.capitalize()}\nFirst name ={first_name.capitalize()}"
  else:
    return "Invalid name format. Please provide a full name with at least a first and last name."

# Example usage:
name1 = "Lokesh kamatham"
formatted_name1 = format_name(name1)
print(formatted_name1)

name2 = "Rithvik Gona"
formatted_name2 = format_name(name2)
print(formatted_name2)

name3 = "Nikhil thatikonda"
formatted_name3 = format_name(name3)
print(formatted_name3)
```

```
Last name = Kamatham
First name =Lokesh
Last name = Gona
First name =Rithvik
Last name = Thatikonda
First name =Nikhil
```

create a python function that formats full names as "Last, First". Example 1 input: enter a name: Lokesh kamatham output: last name = kamatham First name =lokesh Example 2 input: enter a name: Rithvik Gona output: last name = Gona First name = Rithvik Example 3 input: enter a name: Nikhil thatikonda output: Last name = thatikonda First name = nikhil

Sure, I can help with that. Here is a Python function that takes a full name as input and formats it as "Last, First".

I have created the `format_name` function as you requested and demonstrated its usage with the examples you provided.

Would you like to:

1. Test the function with more names?
2. Learn how to handle names with more than two parts (e.g., "Mary Ann Smith")?
3. Explore other string formatting options in Python?

Gemini can make mistakes so double-check it and use code with caution. Learn more

Task Description#4

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

Expected Output#4

- Functional output and comparative reflection

Prompt:

1. Write a Python function that takes a string as input and returns the number of vowels (a, e, i, o, u) in the string.

2. Write a Python function that counts the number of vowels (a, e, i, o, u) in a string.

# Example 1
Input: "hello world"
Output: 3

# Example 2
Input: "Python"
Output: 1

# Example 3
Input: "AI is amazing"
Output: 6

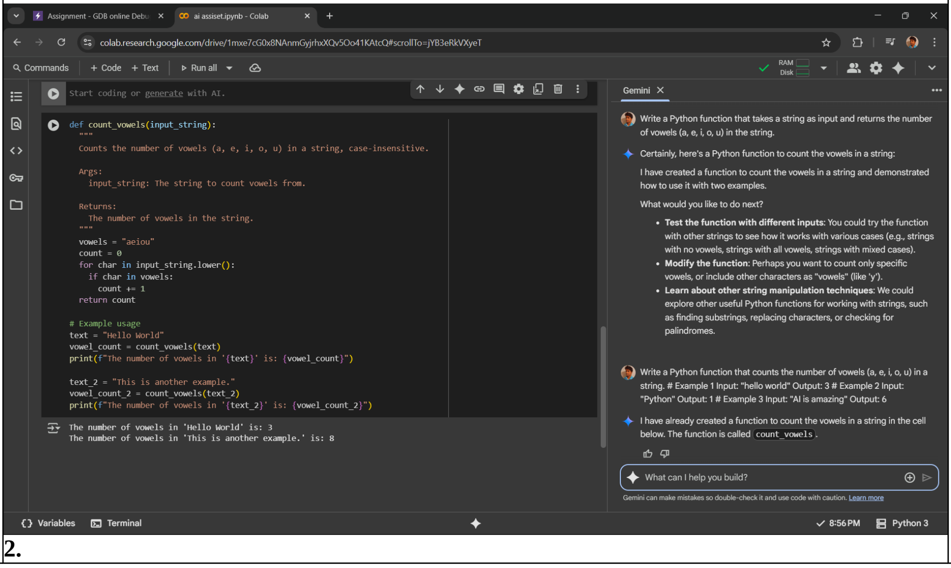
Code:

```
1. def count_vowels(input_string):
  """
  Counts the number of vowels (a, e, i, o, u) in a string, case-insensitive.

  Args:
    input_string: The string to count vowels from.

  Returns:
    The number of vowels in the string.
  """
  vowels = "aeiou"
  count = 0
  for char in input_string.lower():
    if char in vowels:
      count += 1
  return count

# Example usage
text = "Hello World"
vowel_count = count_vowels(text)
print(f"The number of vowels in '{text}' is: {vowel_count}")

text_2 = "This is another example."
vowel_count_2 = count_vowels(text_2)
print(f"The number of vowels in '{text_2}' is: {vowel_count_2}")
```

```
2. def count_vowels(input_string):
  """
  Counts the number of vowels (a, e, i, o, u) in a string, case-insensitive.

  Args:
    input_string: The string to check.

  Returns:
    The number of vowels in the string.
  """
  vowels = "aeiou"
  vowel_count = 0
  for char in input_string.lower():
    if char in vowels:
      vowel_count += 1
  return vowel_count

# Example 1
input_string_1 = "hello world"
vowel_count_1 = count_vowels(input_string_1)
print(f"Input: \"{input_string_1}\"\nOutput: {vowel_count_1}")

# Example 2
input_string_2 = "Python"
vowel_count_2 = count_vowels(input_string_2)
print(f"Input: \"{input_string_2}\"\nOutput: {vowel_count_2}")

# Example 3
input_string_3 = "AI is amazing"
vowel_count_3 = count_vowels(input_string_3)
print(f"Input: \"{input_string_3}\"\nOutput: {vowel_count_3}")
```

Output:

1.



Assignment - GDB online Debu
ai assiset.ipynb - Colab
colab.research.google.com/drive/1mxe7cG0x8NAnmGyjrhxXQv5Oo41KAtcQ#scrollTo=jYB3eRkVXyeT
Commands
+ Code
+ Text
Run all
RAM
Disk
Start coding or generate with AI.
Gemini
What can I help you build?
Variables
Terminal
8:56 PM
Python 3

```
def count_vowels(input_string):
  """
  Counts the number of vowels (a, e, i, o, u) in a string, case-insensitive.

  Args:
    input_string: The string to count vowels from.

  Returns:
    The number of vowels in the string.
  """
  vowels = "aeiou"
  count = 0
  for char in input_string.lower():
    if char in vowels:
      count += 1
  return count

# Example usage
text = "Hello World"
vowel_count = count_vowels(text)
print(f"The number of vowels in '{text}' is: {vowel_count}")

text_2 = "This is another example."
vowel_count_2 = count_vowels(text_2)
print(f"The number of vowels in '{text_2}' is: {vowel_count_2}")
```

```
The number of vowels in 'Hello World' is: 3
The number of vowels in 'This is another example.' is: 8
```

Write a Python function that takes a string as input and returns the number of vowels (a, e, i, o, u) in the string.

Certainly, here's a Python function to count the vowels in a string:

I have created a function to count the vowels in a string and demonstrated how to use it with two examples.

What would you like to do next?

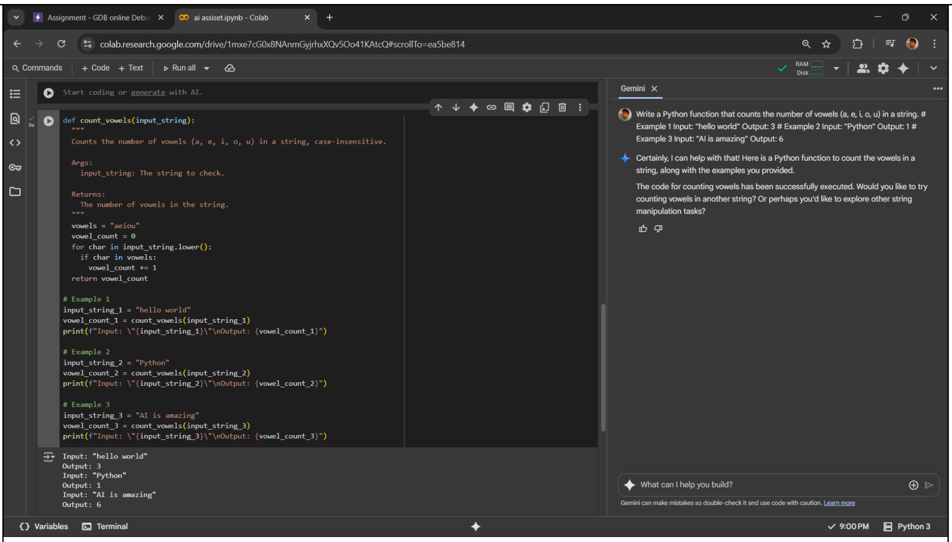
- **Test the function with different inputs**: You could try the function with other strings to see how it works with various cases (e.g., strings with no vowels, strings with all vowels, strings with mixed cases).
- **Modify the function**: Perhaps you want to count only specific vowels, or include other characters as "vowels" (like 'y').
- **Learn about other string manipulation techniques**: We could explore other useful Python functions for working with strings, such as finding substrings, replacing characters, or checking for palindromes.

Write a Python function that counts the number of vowels (a, e, i, o, u) in a string. # Example 1 Input: "hello world" Output: 3 # Example 2 Input: "Python" Output: 1 # Example 3 Input: "AI is amazing" Output: 6

I have already created a function to count the vowels in a string in the cell below. The function is called `count_vowels`.

Gemini can make mistakes so double-check it and use code with caution. Learn more

2.


Assignment - GDB online Debu
ai assiset.ipynb - Colab
colab.research.google.com/drive/1mxe7cG0x8NAnmGyjrhxXQv5Oo41KAtcQ#scrollTo=ea5be814
Commands
+ Code
+ Text
Run all
RAM
Disk
Gemini
Start coding or generate with AI.
Variables
Terminal
9:00 PM
Python 3

```
def count_vowels(input_string):
  """
  Counts the number of vowels (a, e, i, o, u) in a string, case-insensitive.

  Args:
    input_string: The string to check.

  Returns:
    The number of vowels in the string.
  """
  vowels = "aeiou"
  vowel_count = 0
  for char in input_string.lower():
    if char in vowels:
      vowel_count += 1
  return vowel_count

# Example 1
input_string_1 = "hello world"
vowel_count_1 = count_vowels(input_string_1)
print(f"Input: \"{input_string_1}\"\nOutput: {vowel_count_1}")

# Example 2
input_string_2 = "Python"
vowel_count_2 = count_vowels(input_string_2)
print(f"Input: \"{input_string_2}\"\nOutput: {vowel_count_2}")

# Example 3
input_string_3 = "AI is amazing"
vowel_count_3 = count_vowels(input_string_3)
print(f"Input: \"{input_string_3}\"\nOutput: {vowel_count_3}")
```

```
Input: "hello world"
Output: 3
Input: "Python"
Output: 1
Input: "AI is amazing"
Output: 6
```

Write a Python function that counts the number of vowels (a, e, i, o, u) in a string. # Example 1 Input: "hello world" Output: 3 # Example 2 Input: "Python" Output: 1 # Example 3 Input: "AI is amazing" Output: 6

Certainly, I can help with that! Here is a Python function to count the vowels in a string, along with the examples you provided.

The code for counting vowels has been successfully executed. Would you like to try counting vowels in another string? Or perhaps you'd like to explore other string manipulation tasks?

What can I help you build?

Gemini can make mistakes so double-check it and use code with caution. Learn more

Task Description#5

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

Expected Output#5

- Working file-processing function with AI-guided logic

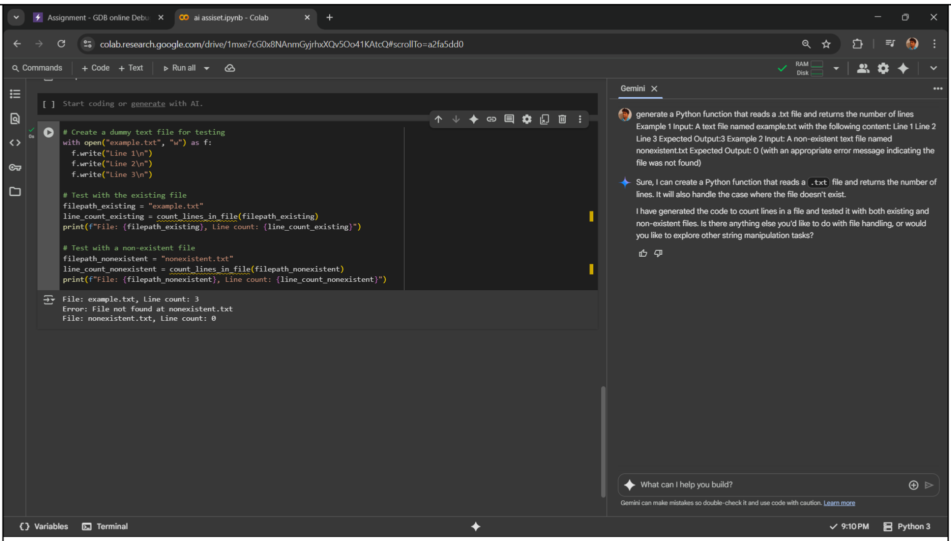
Prompt:

generate a Python function that reads a .txt file and returns the number of lines
Example 1
Input: A text file named example.txt with the following content:
Line 1
Line 2
Line 3
Expected Output:3
Example 2
Input: A non-existent text file named nonexistent.txt
Expected Output:
0 (with an appropriate error message indicating the file was not found)

Code:

```
        # Create a dummy text file for testing
with open("example.txt", "w") as f:
  f.write("Line 1\n")
  f.write("Line 2\n")
  f.write("Line 3\n")

# Test with the existing file
filepath_existing = "example.txt"
line_count_existing = count_lines_in_file(filepath_existing)
print(f"File: {filepath_existing}, Line count: {line_count_existing}")

# Test with a non-existent file
filepath_nonexistent = "nonexistent.txt"
line_count_nonexistent = count_lines_in_file(filepath_nonexistent)
print(f"File: {filepath_nonexistent}, Line count: {line_count_nonexistent}")
```

Output:


Assignment - GDB online Debu
ai assiset.ipynb - Colab
colab.research.google.com/drive/1mxe7cG0x8NAnmGyjrhxXQv5Oo41KAtcQ#scrollTo=a2fa5dd0
Commands
+ Code
+ Text
Run all
RAM
Disk
Gemini
Variables
Terminal
9:10 PM
Python 3

```
[ ] Start coding or generate with AI.
```

```python
# Create a dummy text file for testing
with open("example.txt", "w") as f:
  f.write("Line 1\n")
  f.write("Line 2\n")
  f.write("Line 3\n")

# Test with the existing file
filepath_existing = "example.txt"
line_count_existing = count_lines_in_file(filepath_existing)
print(f"File: {filepath_existing}, Line count: {line_count_existing}")

# Test with a non-existent file
filepath_nonexistent = "nonexistent.txt"
line_count_nonexistent = count_lines_in_file(filepath_nonexistent)
print(f"File: {filepath_nonexistent}, Line count: {line_count_nonexistent}")
```

```
File: example.txt, Line count: 3
Error: File not found at nonexistent.txt
File: nonexistent.txt, Line count: 0
```

generate a Python function that reads a .txt file and returns the number of lines Example 1 Input: A text file named example.txt with the following content: Line 1 Line 2 Line 3 Expected Output:3 Example 2 Input: A non-existent text file named nonexistent.txt Expected Output: 0 (with an appropriate error message indicating the file was not found)

Sure, I can create a Python function that reads a `.txt` file and returns the number of lines. It will also handle the case where the file doesn't exist.

I have generated the code to count lines in a file and tested it with both existing and non-existent files. Is there anything else you'd like to do with file handling, or would you like to explore other string manipulation tasks?

What can I help you build?

Gemini can make mistakes so double-check it and use code with caution. Learn more

Note: Report should be submitted a word document for all tasks in a single document with prompts, comments & code explanation, and output and if required, screenshots

Evaluation Criteria:

Criteria	**Max Marks**
Zero Shot (Task #1)	0.5
One Shot (Task#2)	0.5
Few Shot (Task#3 & Task #5)	1.0
Comparison (Task#4)	0.5
Total	**2.5 Marks**