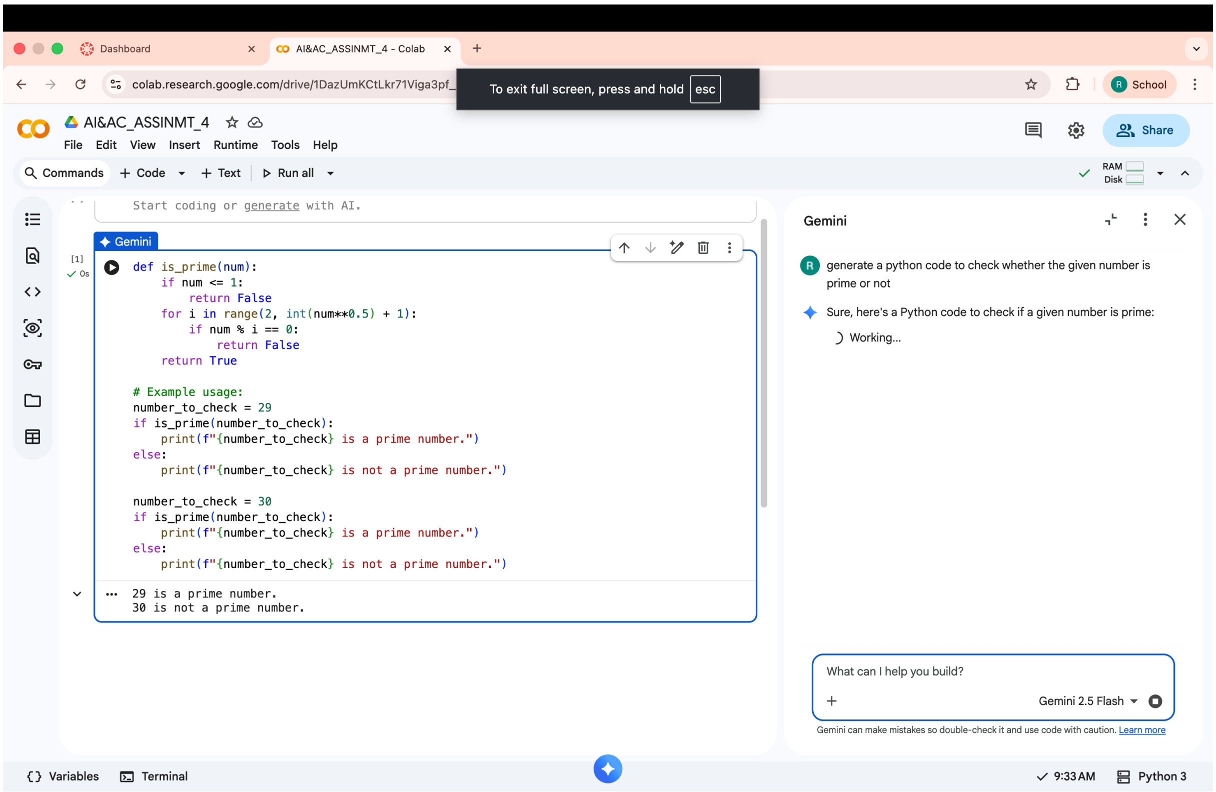
AI ASSISTED CODING - ASSIGNMENT -4

NAME : R.RONITH REDDY
HTNO: 2303A52280

TASK - 1 :

ZERO-SHOT: PROMPT AI WITH ONLY THE INSTRUCTION. WRITE A PYTHON FUNCTION TO DETERMINE WHETHER A GIVEN NUMBER IS PRIME

PROMPT : GENERATE A PYTHON CODE TO CHECK WHETHER THE GIVEN NUMBER IS PRIME OR NOT


Dashboard
AI&AC_ASSINMT_4 - Colab
colab.research.google.com/drive/1DazUmKCtLkr71Viga3pf_
To exit full screen, press and hold esc
School
AI&AC_ASSINMT_4
File Edit View Insert Runtime Tools Help
Share
Commands
+ Code
+ Text
Run all
RAM
Disk
Start coding or generate with AI.
Gemini
[1]
0s
Variables
Terminal
9:33 AM
Python 3
What can I help you build?
Gemini 2.5 Flash
Gemini can make mistakes so double-check it and use code with caution. Learn more

```
def is_prime(num):
    if num <= 1:
        return False
    for i in range(2, int(num**0.5) + 1):
        if num % i == 0:
            return False
    return True

# Example usage:
number_to_check = 29
if is_prime(number_to_check):
    print(f"{number_to_check} is a prime number.")
else:
    print(f"{number_to_check} is not a prime number.")

number_to_check = 30
if is_prime(number_to_check):
    print(f"{number_to_check} is a prime number.")
else:
    print(f"{number_to_check} is not a prime number.")
```

```
29 is a prime number.
30 is not a prime number.
```

Gemini

generate a python code to check whether the given number is prime or not

Sure, here's a Python code to check if a given number is prime:

Working...

EXPLANATION:
THE AL 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.

TASK - 2 :

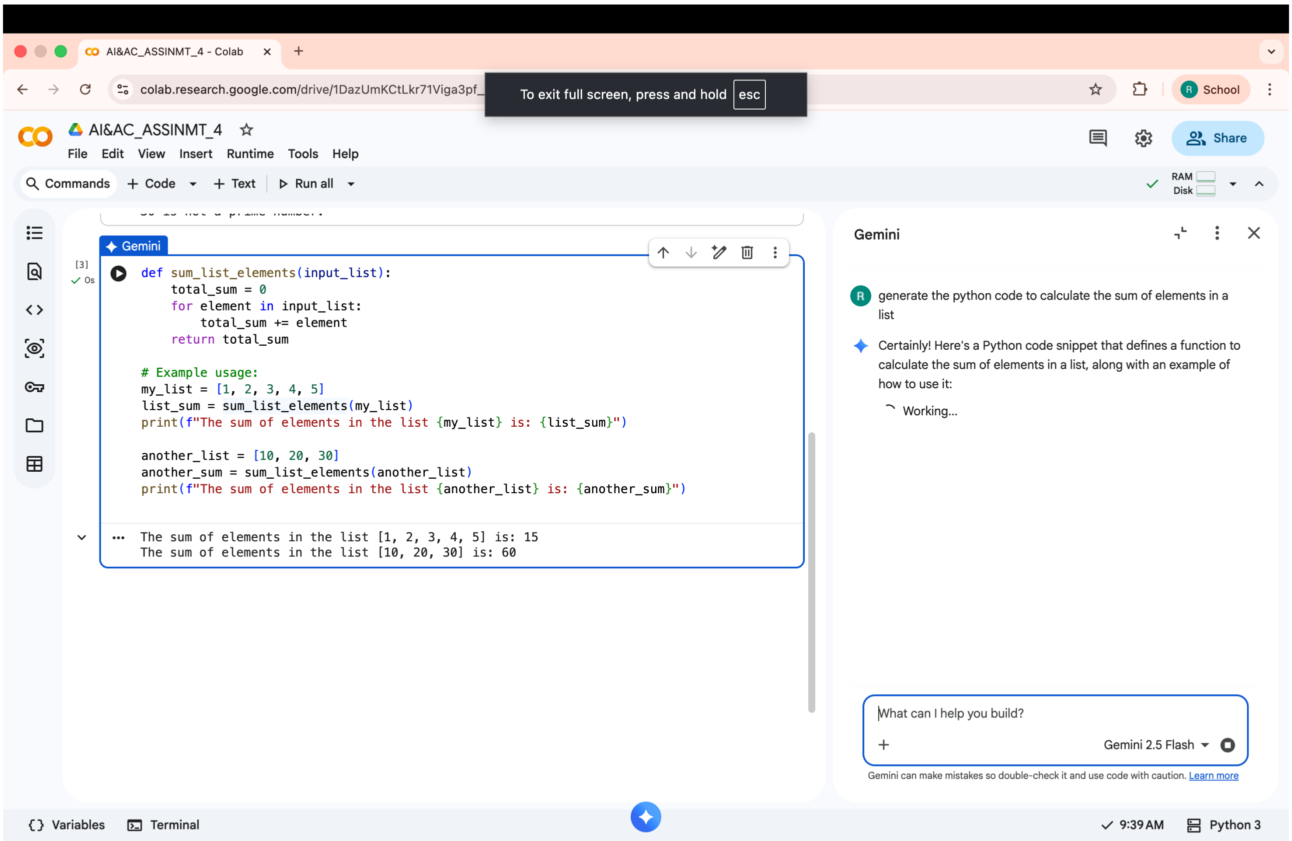
ONE-SHOT: PROVIDE ONE EXAMPLE: INPUT: [1, 2, 3, 4], OUTPUT: 10 TO HELP AI GENERATE A FUNCTION THAT CALCULATES THE SUM OF ELEMENTS IN A LIST.

PROMPT :

GENERATE THE PYTHON CODE TO CALCULATE THE SUM OF ELEMENTS IN A LIST

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.


AI&AC_ASSINMT_4 - Colab
colab.research.google.com/drive/1DazUmKCtLkr71Viga3pf_
To exit full screen, press and hold esc
School
AI&AC_ASSINMT_4
File Edit View Insert Runtime Tools Help
Share
Commands + Code + Text Run all
RAM Disk
Gemini
[3] 0s
Gemini
Variables Terminal
9:39 AM Python 3

```python
def sum_list_elements(input_list):
    total_sum = 0
    for element in input_list:
        total_sum += element
    return total_sum

# Example usage:
my_list = [1, 2, 3, 4, 5]
list_sum = sum_list_elements(my_list)
print(f"The sum of elements in the list {my_list} is: {list_sum}")

another_list = [10, 20, 30]
another_sum = sum_list_elements(another_list)
print(f"The sum of elements in the list {another_list} is: {another_sum}")
```

```
The sum of elements in the list [1, 2, 3, 4, 5] is: 15
The sum of elements in the list [10, 20, 30] is: 60
```

generate the python code to calculate the sum of elements in a list

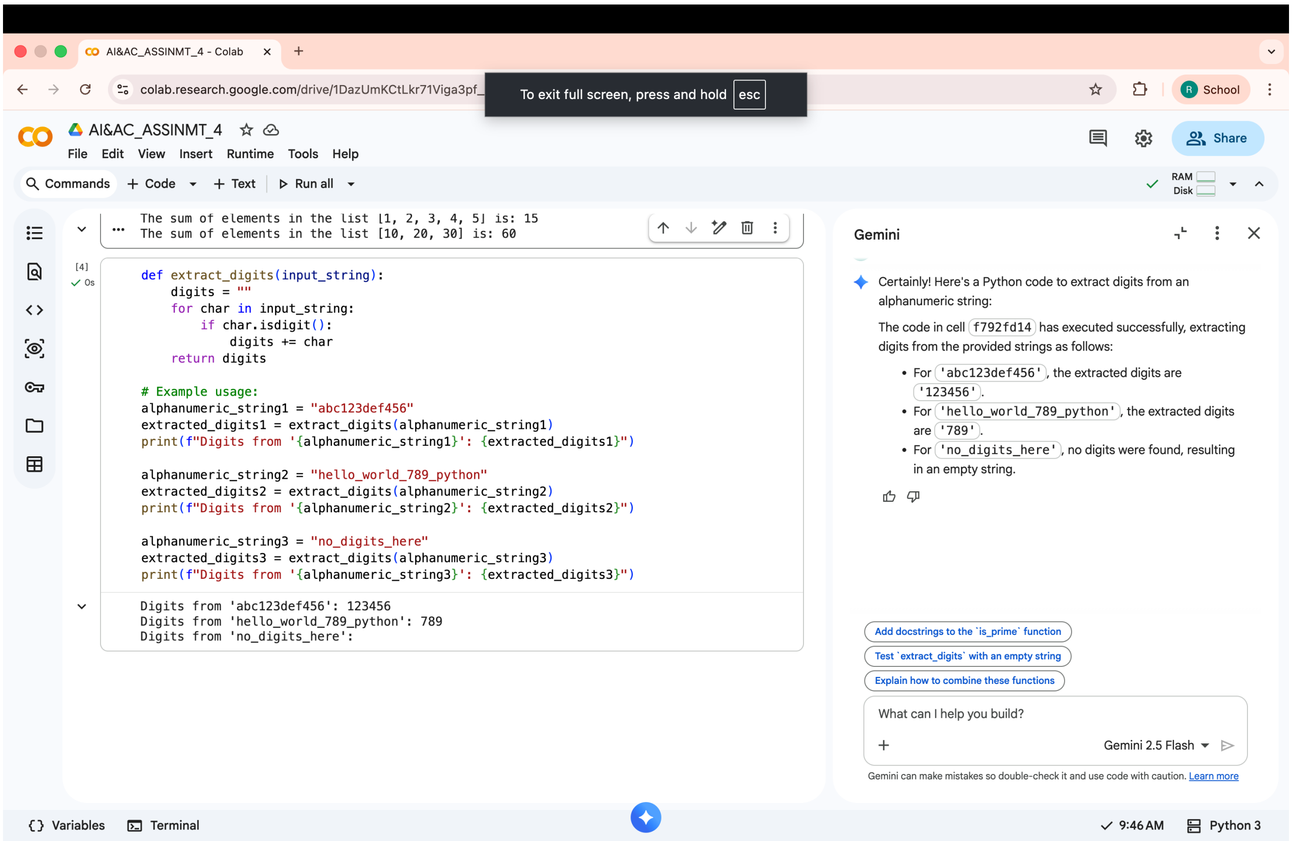
Certainly! Here's a Python code snippet that defines a function to calculate the sum of elements in a list, along with an example of how to use it:

Working...

What can I help you build?
Gemini 2.5 Flash
Gemini can make mistakes so double-check it and use code with caution. Learn more

TASK - 3 : FEW-SHOT: GIVE 2–3 EXAMPLES TO CREATE A FUNCTION THAT EXTRACTS DIGITS FROM AN ALPHANUMERIC STRING.

PROMPT :
CREATE A PYTHON CODE TO EXTRACT DIGITS FROM ALPHANUMERIC STRING


AI&AC_ASSINMT_4 - Colab
colab.research.google.com/drive/1DazUmKCtLkr71Viga3pf_
To exit full screen, press and hold esc
School
AI&AC_ASSINMT_4
File Edit View Insert Runtime Tools Help
Share
Commands + Code + Text Run all
RAM Disk
[4] 0s
Gemini
Variables Terminal
9:46 AM Python 3

```
The sum of elements in the list [1, 2, 3, 4, 5] is: 15
The sum of elements in the list [10, 20, 30] is: 60
```

```python
def extract_digits(input_string):
    digits = ""
    for char in input_string:
        if char.isdigit():
            digits += char
    return digits

# Example usage:
alphanumeric_string1 = "abc123def456"
extracted_digits1 = extract_digits(alphanumeric_string1)
print(f"Digits from '{alphanumeric_string1}': {extracted_digits1}")

alphanumeric_string2 = "hello_world_789_python"
extracted_digits2 = extract_digits(alphanumeric_string2)
print(f"Digits from '{alphanumeric_string2}': {extracted_digits2}")

alphanumeric_string3 = "no_digits_here"
extracted_digits3 = extract_digits(alphanumeric_string3)
print(f"Digits from '{alphanumeric_string3}': {extracted_digits3}")
```

```
Digits from 'abc123def456': 123456
Digits from 'hello_world_789_python': 789
Digits from 'no_digits_here':
```

Certainly! Here's a Python code to extract digits from an alphanumeric string:

The code in cell `f792fd14` has executed successfully, extracting digits from the provided strings as follows:

- For `'abc123def456'`, the extracted digits are `'123456'`.
- For `'hello_world_789_python'`, the extracted digits are `'789'`.
- For `'no_digits_here'`, no digits were found, resulting in an empty string.

Add docstrings to the `is_prime` function
Test `extract_digits` with an empty string
Explain how to combine these functions
What can I help you build?
Gemini 2.5 Flash
Gemini can make mistakes so double-check it and use code with caution. Learn more

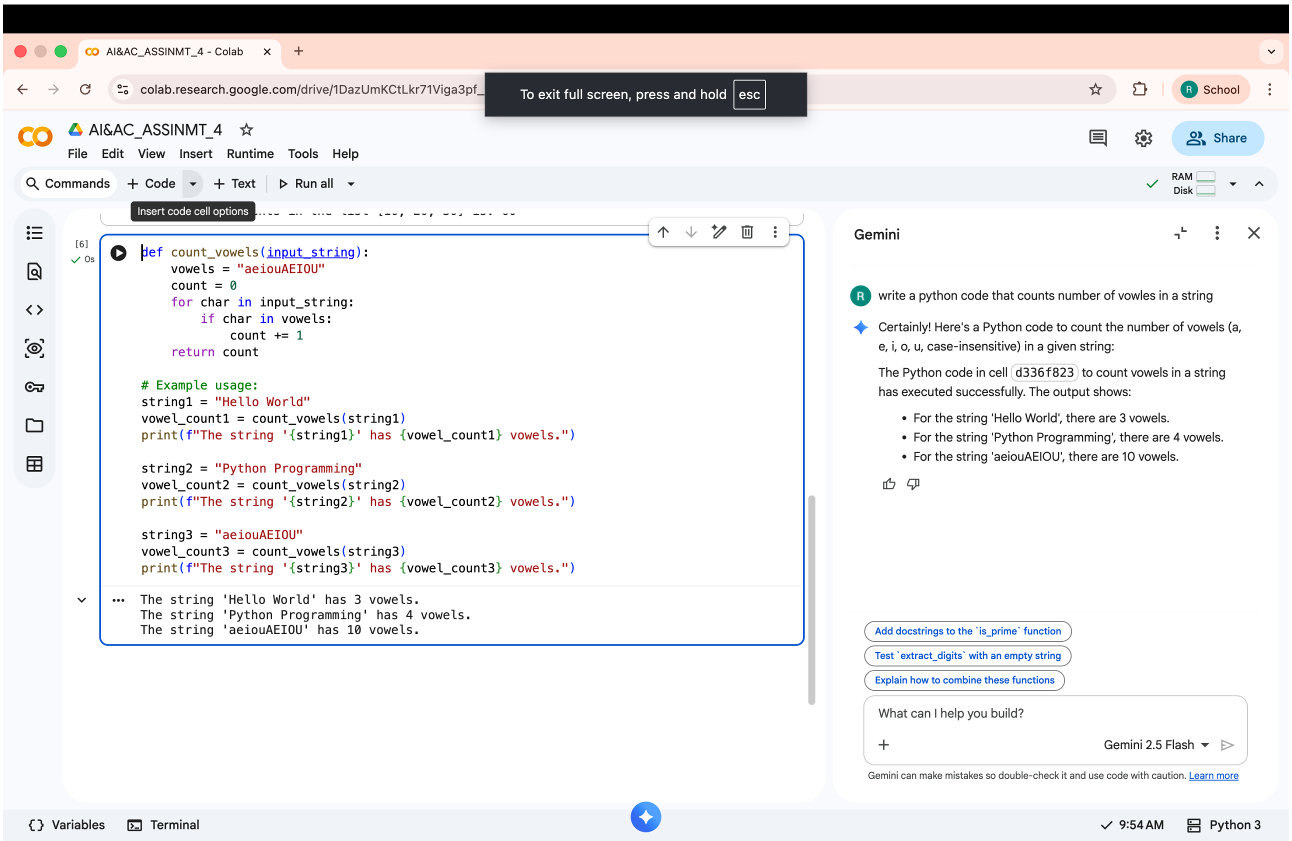
EXPLANATION :
GEMINI AI GENERATED A FUNCTION THAT LOOPS THE STRING AND CHECKS DIGITS AND OUTPUT SHOWS THE RESULT STRING AND RETURNED .

TASK - 4 :

COMPARE ZERO-SHOT VS FEW-SHOT PROMPTING FOR GENERATING A FUNCTION THAT COUNTS THE NUMBER OF VOWELS IN A STRING.

ZERO SHOT :

WRITE A PYTHON CODE THAT COUNTS NUMBER OF VOWLES IN A STRING


AI&AC_ASSINMT_4 - Colab
colab.research.google.com/drive/1DazUmKCtLkr71Viga3pf_
To exit full screen, press and hold esc
School
AI&AC_ASSINMT_4
File Edit View Insert Runtime Tools Help
Share
Commands + Code + Text Run all
Insert code cell options
RAM Disk
[6]
0s
Variables Terminal
9:54 AM Python 3

```
def count_vowels(input_string):
    vowels = "aeiouAEIOU"
    count = 0
    for char in input_string:
        if char in vowels:
            count += 1
    return count

# Example usage:
string1 = "Hello World"
vowel_count1 = count_vowels(string1)
print(f"The string '{string1}' has {vowel_count1} vowels.")

string2 = "Python Programming"
vowel_count2 = count_vowels(string2)
print(f"The string '{string2}' has {vowel_count2} vowels.")

string3 = "aeiouAEIOU"
vowel_count3 = count_vowels(string3)
print(f"The string '{string3}' has {vowel_count3} vowels.")
```

```
The string 'Hello World' has 3 vowels.
The string 'Python Programming' has 4 vowels.
The string 'aeiouAEIOU' has 10 vowels.
```

Gemini

write a python code that counts number of vowles in a string

Certainly! Here's a Python code to count the number of vowels (a, e, i, o, u, case-insensitive) in a given string:

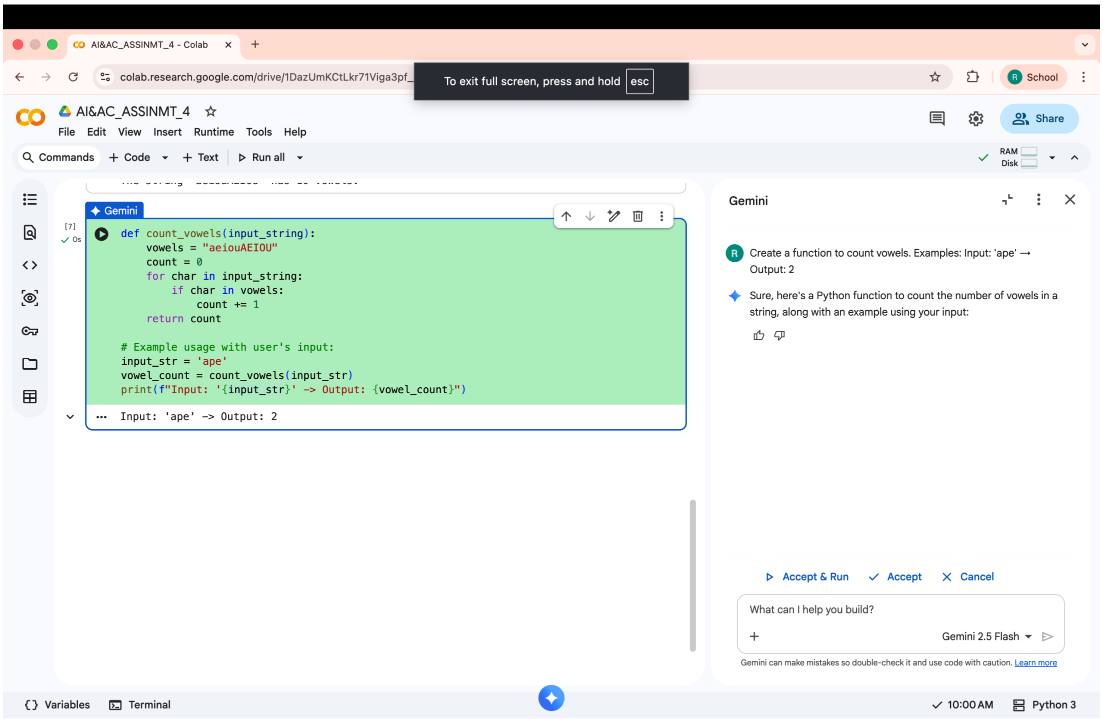
The Python code in cell `d336f823` to count vowels in a string has executed successfully. The output shows:

- For the string 'Hello World', there are 3 vowels.
- For the string 'Python Programming', there are 4 vowels.
- For the string 'aeiouAEIOU', there are 10 vowels.

Add docstrings to the `is_prime` function
Test `extract_digits` with an empty string
Explain how to combine these functions
What can I help you build?
Gemini 2.5 Flash
Gemini can make mistakes so double-check it and use code with caution. Learn more

FEW SHOT :

CREATE A FUNCTION TO COUNT VOWELS. EXAMPLES: INPUT: 'APE' → OUTPUT: 2


AI&AC_ASSINMT_4 - Colab
colab.research.google.com/drive/1DazUmKCtLkr71Viga3pf_
To exit full screen, press and hold esc
School
AI&AC_ASSINMT_4
File Edit View Insert Runtime Tools Help
Share
Commands + Code + Text Run all
RAM Disk
Gemini
[7]
0s
Accept & Run
Accept
Cancel
What can I help you build?
Gemini 2.5 Flash
Gemini can make mistakes so double-check it and use code with caution. Learn more
Variables Terminal
10:00 AM Python 3

```
def count_vowels(input_string):
    vowels = "aeiouAEIOU"
    count = 0
    for char in input_string:
        if char in vowels:
            count += 1
    return count

# Example usage with user's input:
input_str = 'ape'
vowel_count = count_vowels(input_str)
print(f"Input: '{input_str}' -> Output: {vowel_count}")
```

```
Input: 'ape' -> Output: 2
```

Gemini

Create a function to count vowels. Examples: Input: 'ape' → Output: 2

Sure, here's a Python function to count the number of vowels in a string, along with an example using your input:

EXPLANATION :

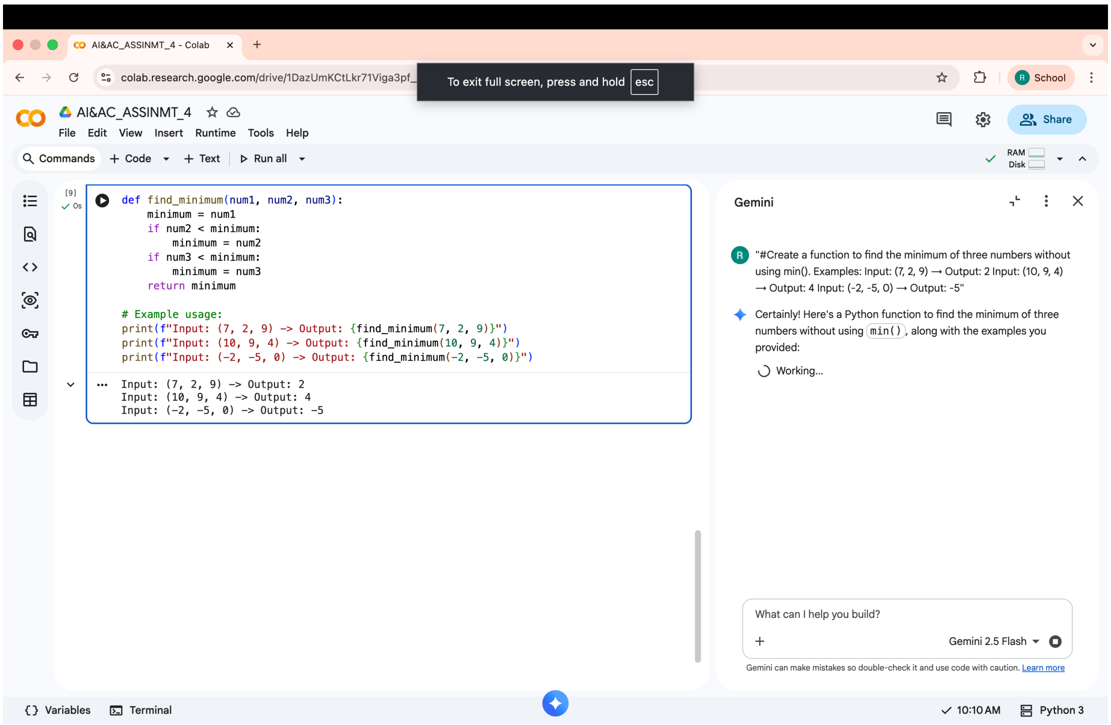
- ZERO-SHOT PROMPTING USES BASIC LOGIC FOR CLARITY.
- FEW-SHOT PROMPTING IS MORE OPTIMIZED IN TERMS OF CLARITY.
- BOTH METHODS ARE CONSIDERED CORRECT REGARDING ACCURACY.

TASK - 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 :

CREATE A FUNCTION TO FIND THE MINIMUM OF THREE NUMBERS WITHOUT USING MIN(). EXAMPLES: INPUT: (7, 2, 9) → OUTPUT: 2 INPUT: (10, 9, 4) → OUTPUT: 4 INPUT: (-2, -5, 0) → OUTPUT: -5"


AI&AC_ASSINMT_4 - Colab
colab.research.google.com/drive/1DazUmKCtLkr71Viga3pf_
To exit full screen, press and hold esc
School
AI&AC_ASSINMT_4
File
Edit
View
Insert
Runtime
Tools
Help
Share
Commands
+ Code
+ Text
Run all
RAM
Disk
[9]
0s
Gemini
What can I help you build?
Gemini 2.5 Flash
Gemini can make mistakes so double-check it and use code with caution. Learn more
Variables
Terminal
10:10 AM
Python 3

```python
def find_minimum(num1, num2, num3):
    minimum = num1
    if num2 < minimum:
        minimum = num2
    if num3 < minimum:
        minimum = num3
    return minimum

# Example usage:
print(f"Input: (7, 2, 9) -> Output: {find_minimum(7, 2, 9)}")
print(f"Input: (10, 9, 4) -> Output: {find_minimum(10, 9, 4)}")
print(f"Input: (-2, -5, 0) -> Output: {find_minimum(-2, -5, 0)}")
```

```
Input: (7, 2, 9) -> Output: 2
Input: (10, 9, 4) -> Output: 4
Input: (-2, -5, 0) -> Output: -5
```

"#Create a function to find the minimum of three numbers without using min(). Examples: Input: (7, 2, 9) → Output: 2 Input: (10, 9, 4) → Output: 4 Input: (-2, -5, 0) → Output: -5"

Certainly! Here's a Python function to find the minimum of three numbers without using `min()`, along with the examples you provided:

Working...

EXPLANATION:

THE FUNCTION ASSUMES THE FIRST NUMBER IS SMALLEST AND COMPARES IT WITH OTHERS. THE SMALLEST VALUE IS UPDATED AND RETURNED.

OBSERVATION :

- AI PERFORMANCE: THE AI SUCCESSFULLY INFERRED THE CORRECT LOGIC.
- METHODOLOGY: THIS SUCCESS WAS ACHIEVED USING FEW-SHOT EXAMPLES.
- CONSTRAINT ADHERENCE: THE SOLUTION FOLLOWED INSTRUCTIONS BY AVOIDING THE USE OF BUILT-IN FUNCTIONS.