AI ASSISTED CODING

P.Aneesh Reddy **2303A51120**

BATCH – 03 **20 – 01 – 2026**

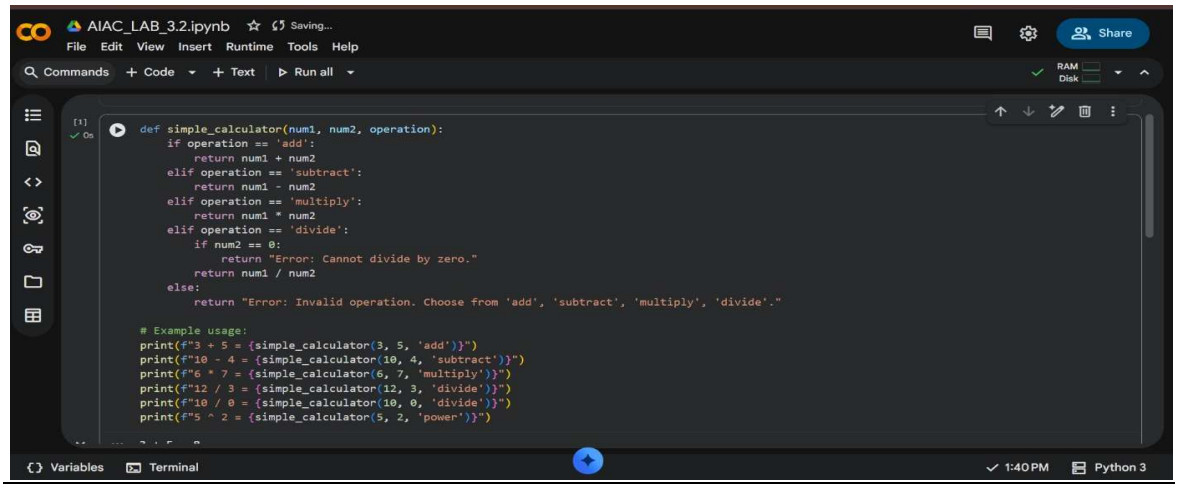
ASSIGNMENT – 3.2

LAB – 03 : Prompt Engineering – Improving Prompts and Context Management.

TASK - 01 : Progressive Prompting for Calculator Design.

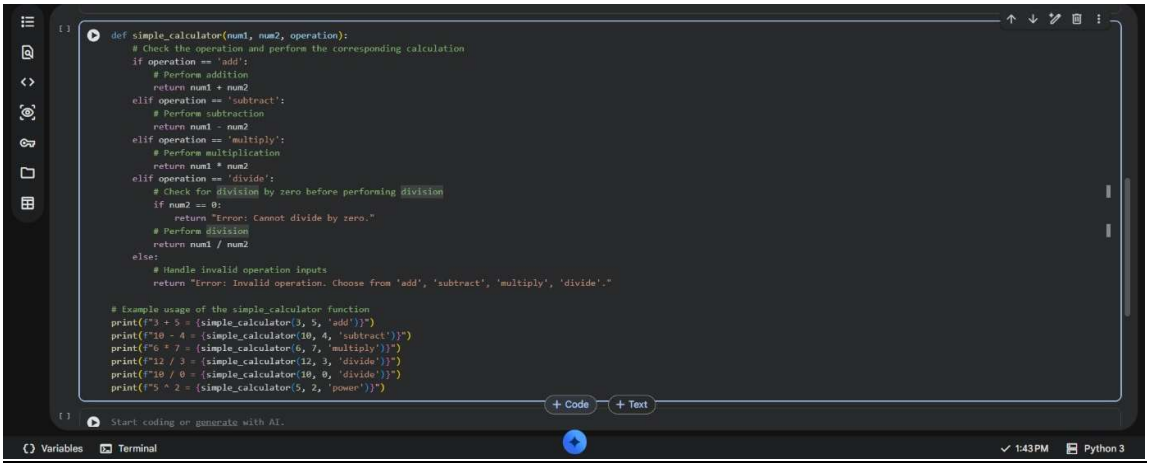
Prompt – 01: Create a Python function named simple_calculator.

Code :



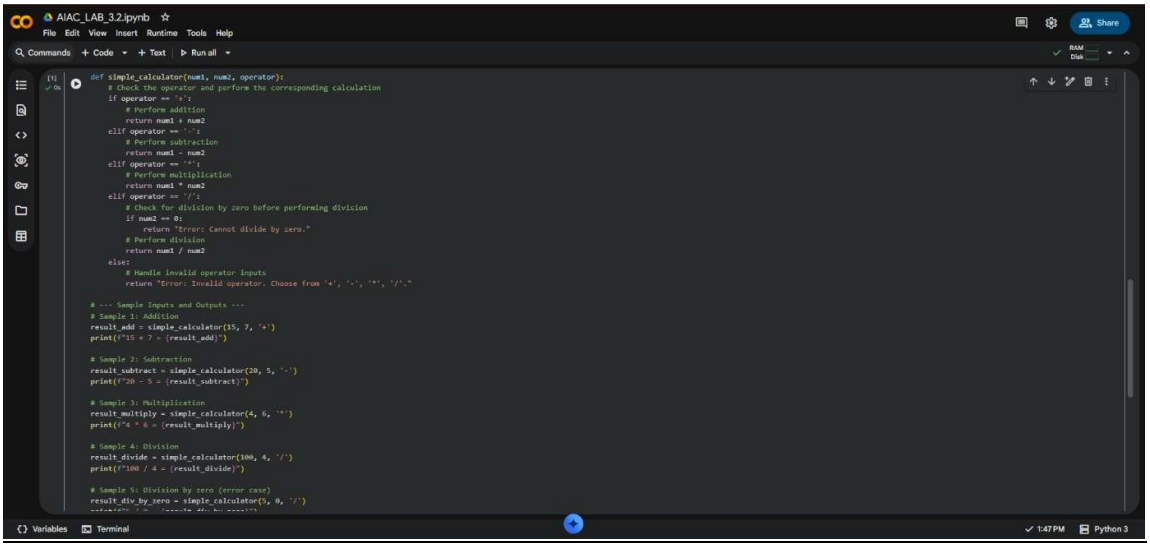
Prompt – 02 : Create a Python function named simple_calculator that performs basic arithmetic operations (addition, subtraction, multiplication, division). Add comments explaining each step.

Code:



Prompt – 03 : Create a Python function named simple_calculator that takes two numbers and an operator (+, -, , /) as input and returns the result. Include comments and provide at least two sample inputs with outputs.

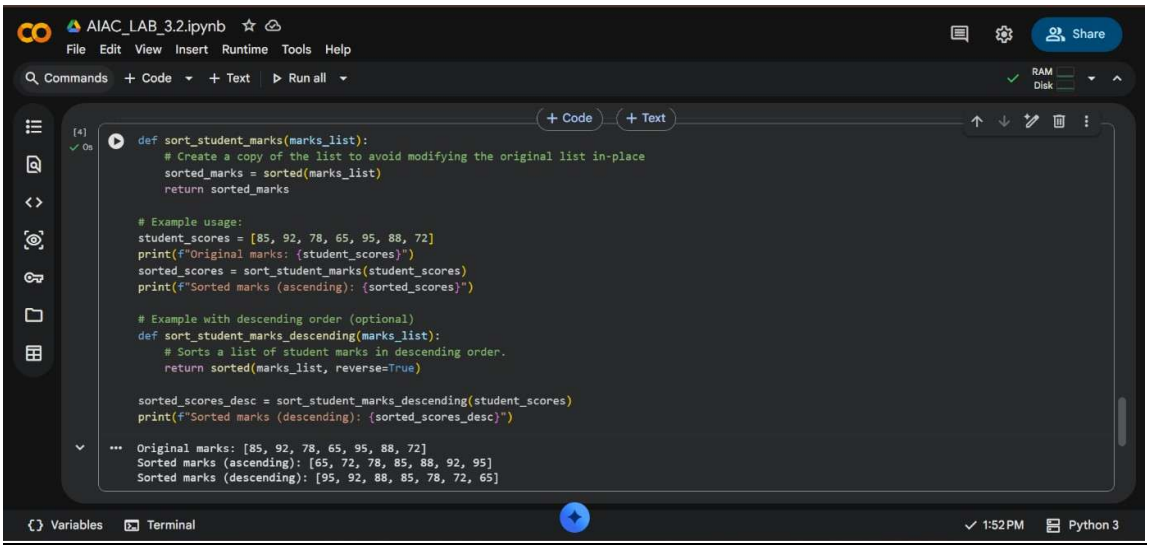
Code :



TASK – 02 : Refining Prompts for Sorting Logic

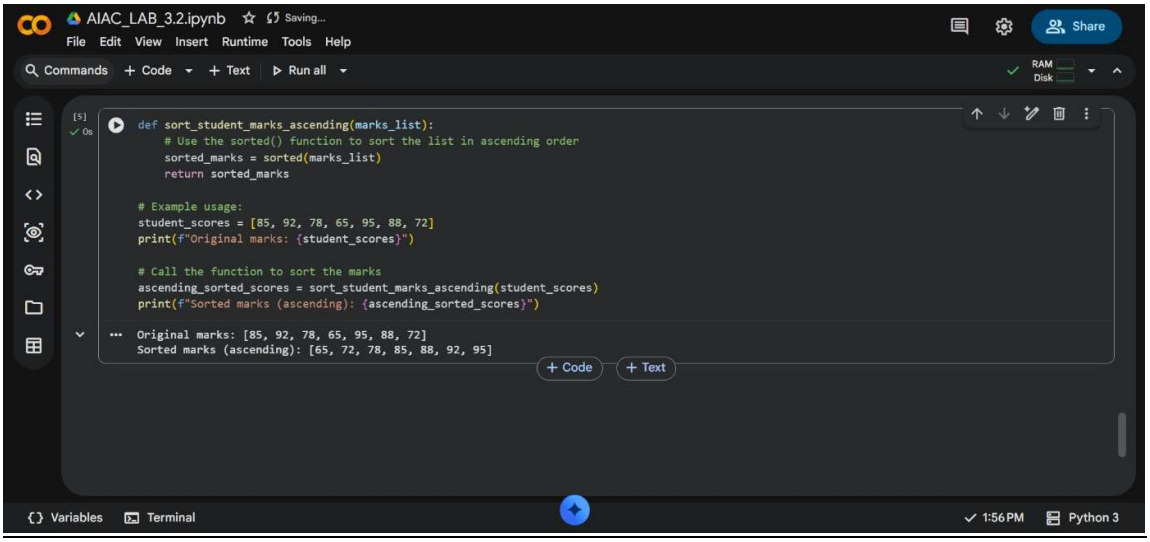
Prompt – 01 : Write a Python Function to sort the Student Marks.

Code :



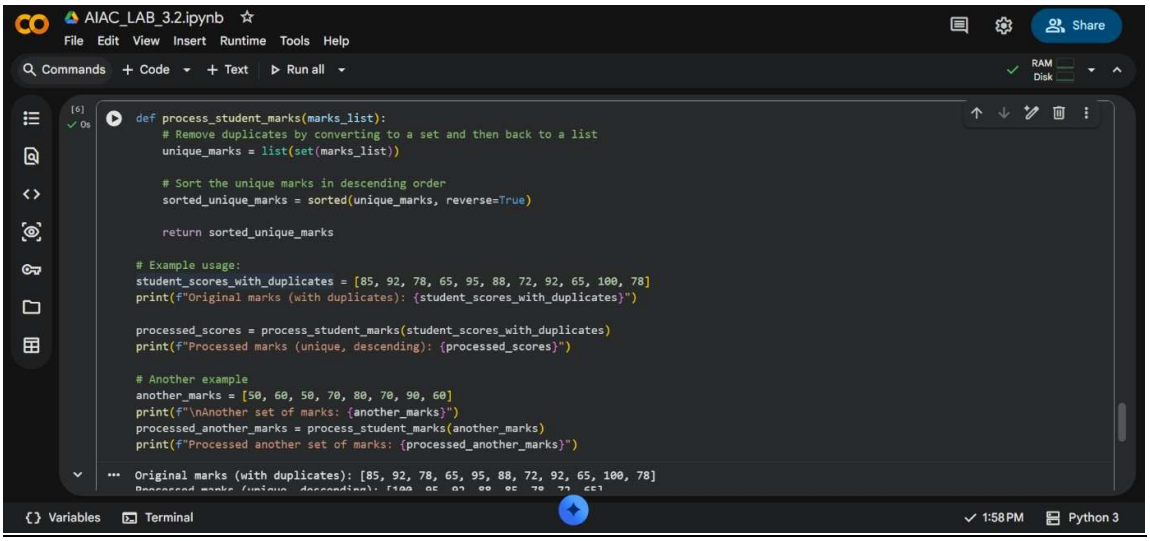
Prompt – 02 : Write a Python function that takes a list of student marks and sorts them in ascending order.

Code :



Prompt – 03 : Write an efficient Python function that takes a list of student marks (0–100), sorts them in descending order, removes duplicates, and returns the sorted list.

Code :



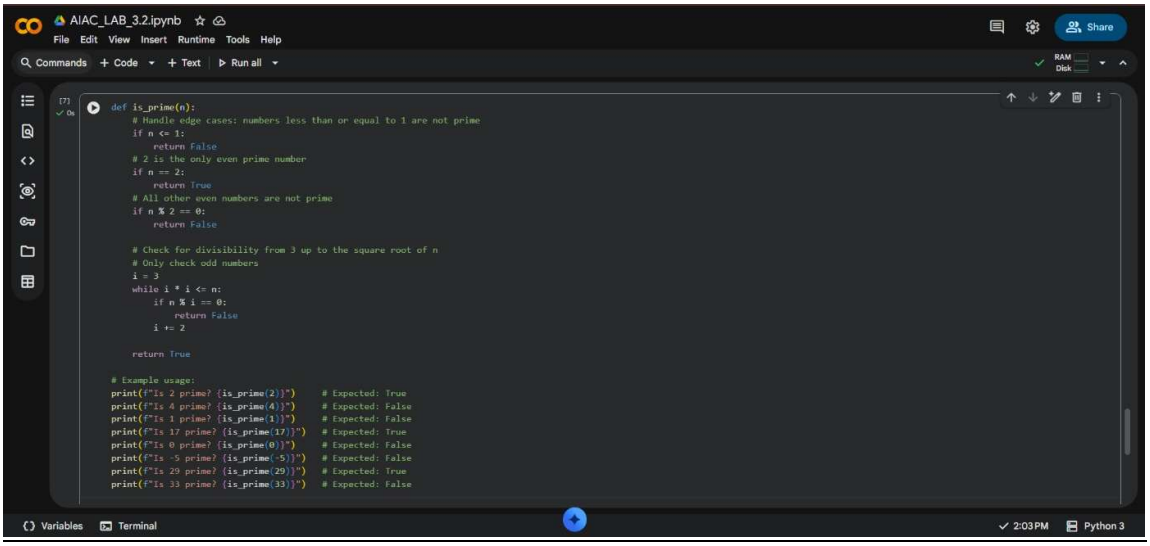
TASK – 03 : Few-Shot Prompting for Prime Number Validation. **Prompt :** Write a Python function is_prime(n) that checks whether a given is Prime or Not, Using Examples:

I/P: 2 -> O/P True

I/P: 4 -> O/P False

Handle edge cases like negative numbers and 0 properly.

Code:



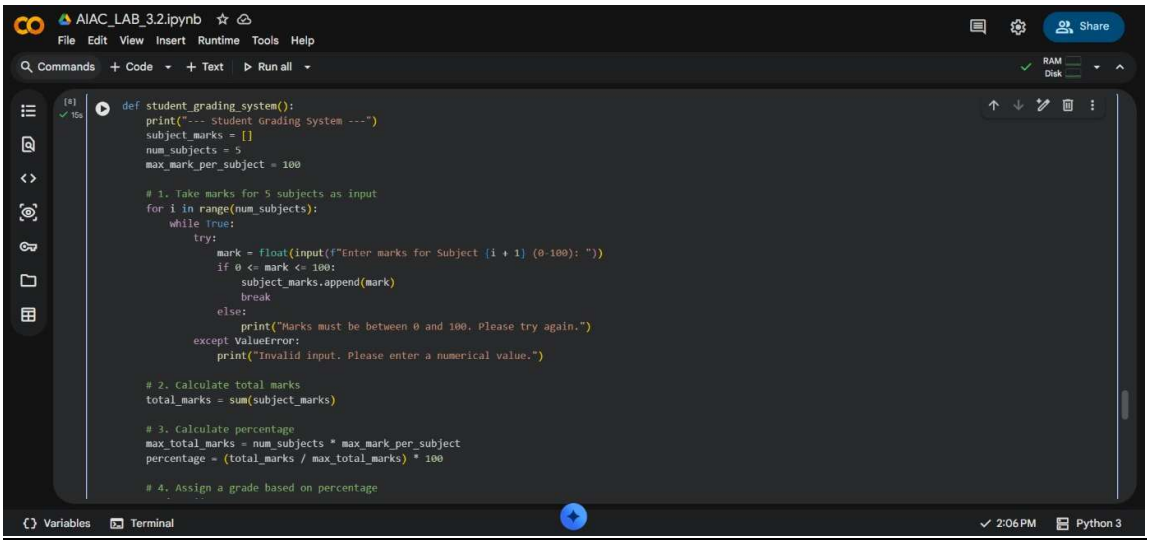
TASK – 04 : Prompt-Guided UI Design for Student Grading System.

Prompt : Design a simple Python-based user interface (CLI or GUI) for a student grading system.

The program should:

- Take marks for 5 subjects as input
- Calculate total marks
- Calculate percentage
- Assign a grade based on percentage
- Display all results clearly.

Code :



```
        # 4. Assign a grade based on percentage
        grade = ''
        if percentage >= 90:
            grade = 'A+'
        elif percentage >= 80:
            grade = 'A'
        elif percentage >= 70:
            grade = 'B+'
        elif percentage >= 60:
            grade = 'B'
        elif percentage >= 50:
            grade = 'C'
        elif percentage >= 40:
            grade = 'D'
        else:
            grade = 'F'

        # 5. Display all results clearly
        print("\n--- Results ---")
        for i, mark in enumerate(subject_marks):
            print(f"Subject {i + 1} Marks: {mark:.2f}")
        print(f"Total Marks: {total_marks:.2f} / {max_total_marks}")
        print(f"Percentage: {percentage:.2f}%")
        print(f"Grade: {grade}")
        print("----------------------")

    # Run the grading system
    student_grading_system()
```

Variables Terminal 2:06 PM Python 3

Output:

AIAC_LAB_3.2.ipynb

File Edit View Insert Runtime Tools Help

Share

Commands + Code + Text Run all

RAM Disk

```
        print(f"Total Marks: {total_marks:.2f} / {max_total_marks}")
        print(f"Percentage: {percentage:.2f}%")
        print(f"Grade: {grade}")
        print("----------------------")

    # Run the grading system
    student_grading_system()
```

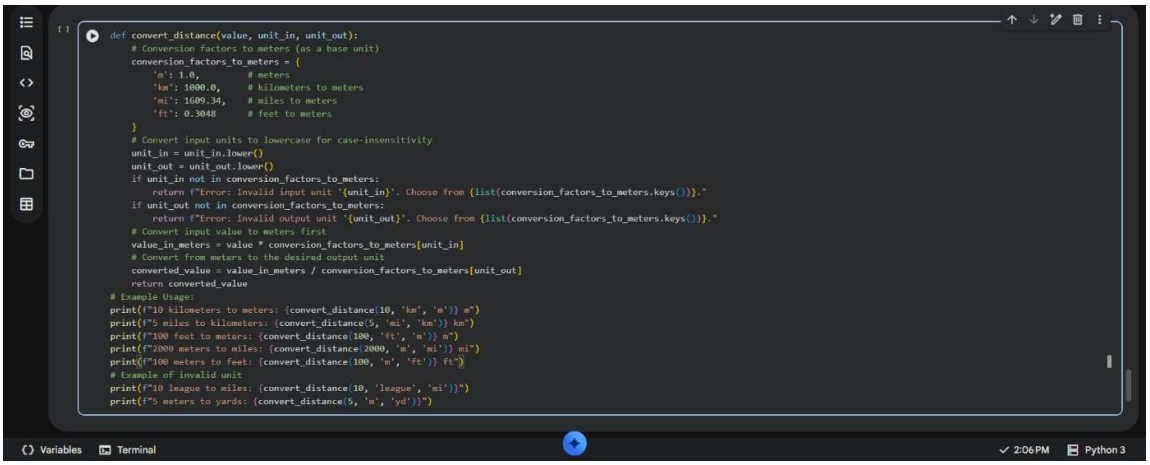
```
--- Student Grading System ---
Enter marks for Subject 1 (0-100): 98
Enter marks for Subject 2 (0-100): 54
Enter marks for Subject 3 (0-100): 99
Enter marks for Subject 4 (0-100): 45
Enter marks for Subject 5 (0-100): 32

--- Results ---
Subject 1 Marks: 98.00
Subject 2 Marks: 54.00
Subject 3 Marks: 99.00
Subject 4 Marks: 45.00
Subject 5 Marks: 32.00
Total Marks: 328.00 / 500
Percentage: 65.60%
Grade: B
----------------------
```

Variables Terminal 2:06 PM Python 3

<u>TASK – 05 :</u> Analyzing Prompt Specificity in Unit Conversion Functions.

<u>Prompt –</u> 01 : Write a function to convert distance.

<u>Code:</u>

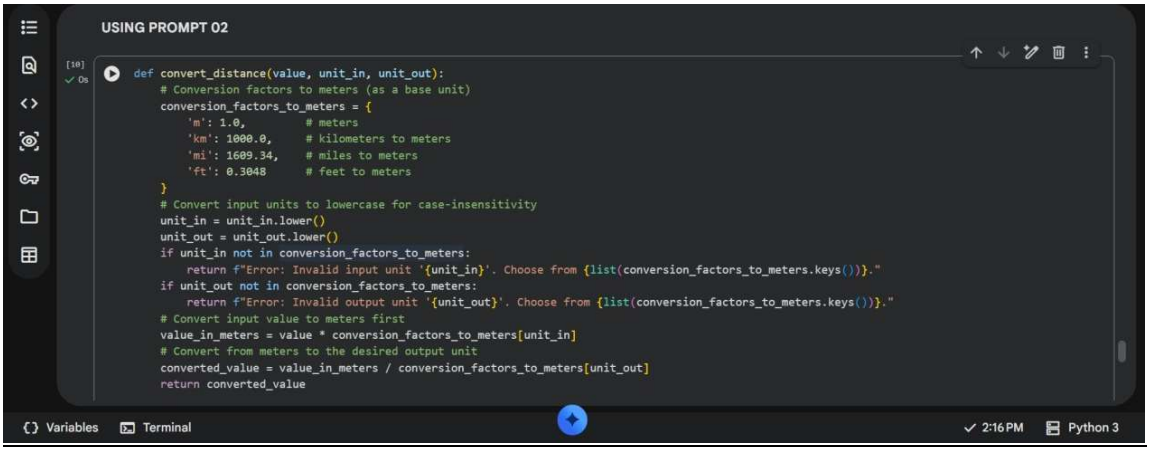


```
def convert_distance(value, unit_in, unit_out):
    # Conversion factors to meters (as a base unit)
    conversion_factors_to_meters = {
        'm': 1.0,         # meters
        'km': 1000.0,     # kilometers to meters
        'mi': 1609.34,    # miles to meters
        'ft': 0.3048      # feet to meters
    }
    # Convert input units to lowercase for case-insensitivity
    unit_in = unit_in.lower()
    unit_out = unit_out.lower()
    if unit_in not in conversion_factors_to_meters:
        return f"Error: Invalid input unit '{unit_in}'. Choose from {list(conversion_factors_to_meters.keys())}."
    if unit_out not in conversion_factors_to_meters:
        return f"Error: Invalid output unit '{unit_out}'. Choose from {list(conversion_factors_to_meters.keys())}."
    # Convert input value to meters first
    value_in_meters = value * conversion_factors_to_meters[unit_in]
    # Convert from meters to the desired output unit
    converted_value = value_in_meters / conversion_factors_to_meters[unit_out]
    return converted_value
# Example Usage:
print(f"10 kilometers to meters: {convert_distance(10, 'km', 'm')} m")
print(f"5 miles to kilometers: {convert_distance(5, 'mi', 'km')} km")
print(f"100 feet to meters: {convert_distance(100, 'ft', 'm')} m")
print(f"2000 meters to miles: {convert_distance(2000, 'm', 'mi')} mi")
print(f"100 meters to feet: {convert_distance(100, 'm', 'ft')} ft")
# Example of invalid unit
print(f"10 league to miles: {convert_distance(10, 'league', 'mi')}")
print(f"5 meters to yards: {convert_distance(5, 'm', 'yd')}")
```

Variables Terminal 2:06 PM Python 3

Prompt – 02: Write a Python function to convert kilometers to miles.

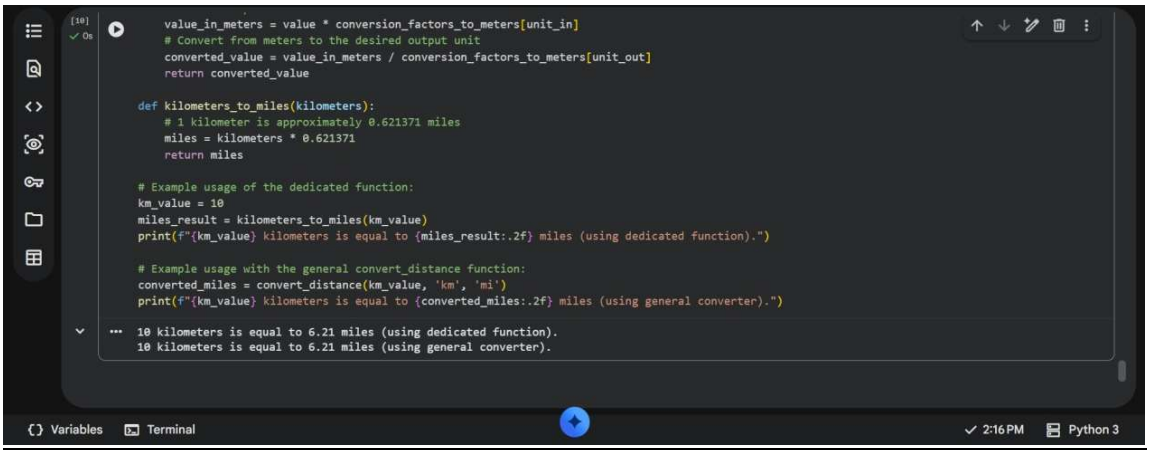
Code :



USING PROMPT 02

```
def convert_distance(value, unit_in, unit_out):
    # Conversion factors to meters (as a base unit)
    conversion_factors_to_meters = {
        'm': 1.0,         # meters
        'km': 1000.0,     # kilometers to meters
        'mi': 1609.34,    # miles to meters
        'ft': 0.3048      # feet to meters
    }
    # Convert input units to lowercase for case-insensitivity
    unit_in = unit_in.lower()
    unit_out = unit_out.lower()
    if unit_in not in conversion_factors_to_meters:
        return f"Error: Invalid input unit '{unit_in}'. Choose from {list(conversion_factors_to_meters.keys())}."
    if unit_out not in conversion_factors_to_meters:
        return f"Error: Invalid output unit '{unit_out}'. Choose from {list(conversion_factors_to_meters.keys())}."
    # Convert input value to meters first
    value_in_meters = value * conversion_factors_to_meters[unit_in]
    # Convert from meters to the desired output unit
    converted_value = value_in_meters / conversion_factors_to_meters[unit_out]
    return converted_value
```

Variables Terminal 2:16 PM Python 3



```
    value_in_meters = value * conversion_factors_to_meters[unit_in]
    # Convert from meters to the desired output unit
    converted_value = value_in_meters / conversion_factors_to_meters[unit_out]
    return converted_value

def kilometers_to_miles(kilometers):
    # 1 kilometer is approximately 0.621371 miles
    miles = kilometers * 0.621371
    return miles

# Example usage of the dedicated function:
km_value = 10
miles_result = kilometers_to_miles(km_value)
print(f"{km_value} kilometers is equal to {miles_result:.2f} miles (using dedicated function).")

# Example usage with the general convert_distance function:
converted_miles = convert_distance(km_value, 'km', 'mi')
print(f"{km_value} kilometers is equal to {converted_miles:.2f} miles (using general converter).")
```

```
10 kilometers is equal to 6.21 miles (using dedicated function).
10 kilometers is equal to 6.21 miles (using general converter).
```

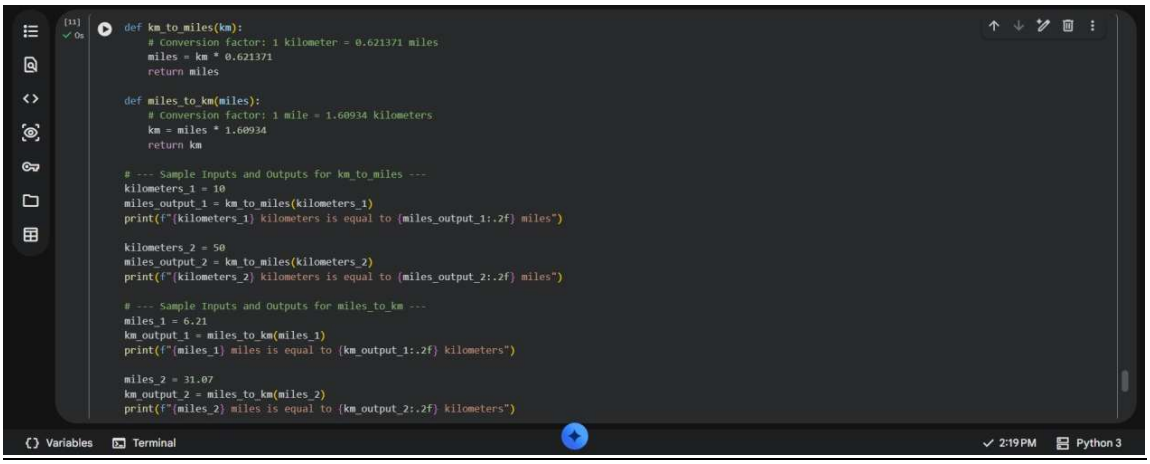
Variables Terminal 2:16 PM Python 3

Prompt – 03 : Write two Python functions:

1. km_to_miles(km)

2. miles_to_km(miles)
   Use correct conversion factors, add comments, and provide sample inputs and outputs.

Code :



```
[11]
✓ 0s

def km_to_miles(km):
    # Conversion factor: 1 kilometer = 0.621371 miles
    miles = km * 0.621371
    return miles

def miles_to_km(miles):
    # Conversion factor: 1 mile = 1.60934 kilometers
    km = miles * 1.60934
    return km

# --- Sample Inputs and Outputs for km_to_miles ---
kilometers_1 = 10
miles_output_1 = km_to_miles(kilometers_1)
print(f"{kilometers_1} kilometers is equal to {miles_output_1:.2f} miles")

kilometers_2 = 50
miles_output_2 = km_to_miles(kilometers_2)
print(f"{kilometers_2} kilometers is equal to {miles_output_2:.2f} miles")

# --- Sample Inputs and Outputs for miles_to_km ---
miles_1 = 6.21
km_output_1 = miles_to_km(miles_1)
print(f"{miles_1} miles is equal to {km_output_1:.2f} kilometers")

miles_2 = 31.07
km_output_2 = miles_to_km(miles_2)
print(f"{miles_2} miles is equal to {km_output_2:.2f} kilometers")

Variables  Terminal                                  2:19 PM  Python 3
```

THANK YOU!!