ASSIGNMENT 2.1

Name:T.Rakshitha

Hall ticket No:2303a51172

Batch:18

Task 1 :

Statistical Summary for Survey Data

❖ Scenario:

You are a data analyst intern working with survey responses stored as numerical lists.

❖ Task:

Use Google Gemini in Colab to generate a Python function that reads a list of numbers and calculates the mean, minimum, and maximum values.

❖ Expected Output:

➢ Correct Python function

➢ Output shown in Colab

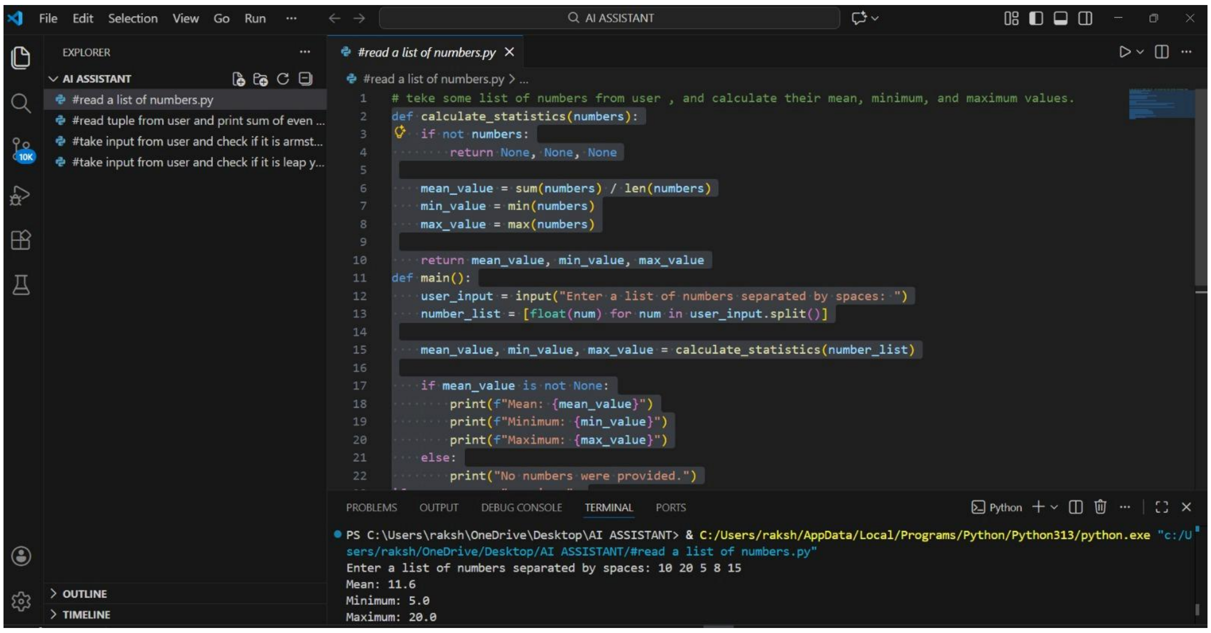
➢ Screenshot of Gemini prompt and result

Prompt :

#read a list of numbers

#calculates the mean, minimum, and maximum values.

Code :



EXPLORER
AI ASSISTANT
#read a list of numbers.py
#read tuple from user and print sum of even ...
#take input from user and check if it is armst...
#take input from user and check if it is leap y...
OUTLINE
TIMELINE

```python
# teke some list of numbers from user , and calculate their mean, minimum, and maximum values.
def calculate_statistics(numbers):
    if not numbers:
        return None, None, None

    mean_value = sum(numbers) / len(numbers)
    min_value = min(numbers)
    max_value = max(numbers)

    return mean_value, min_value, max_value
def main():
    user_input = input("Enter a list of numbers separated by spaces: ")
    number_list = [float(num) for num in user_input.split()]

    mean_value, min_value, max_value = calculate_statistics(number_list)

    if mean_value is not None:
        print(f"Mean: {mean_value}")
        print(f"Minimum: {min_value}")
        print(f"Maximum: {max_value}")
    else:
        print("No numbers were provided.")
```

```
PS C:\Users\raksh\OneDrive\Desktop\AI ASSISTANT> & C:/Users/raksh/AppData/Local/Programs/Python/Python313/python.exe "c:/U
sers/raksh/OneDrive/Desktop/AI ASSISTANT/#read a list of numbers.py"
Enter a list of numbers separated by spaces: 10 20 5 8 15
Mean: 11.6
Minimum: 5.0
Maximum: 20.0
```

Analysis

1.Gemini generated correct code.

2.Output was accurate.

3. Easy to use in vs.

Task 2 :

Scenario:

You are evaluating AI tools for numeric validation logic.

❖ Task:

Generate an Armstrong number checker using Gemini and GitHub Copilot.

Compare their outputs, logic style, and clarity.

❖ Expected Output:

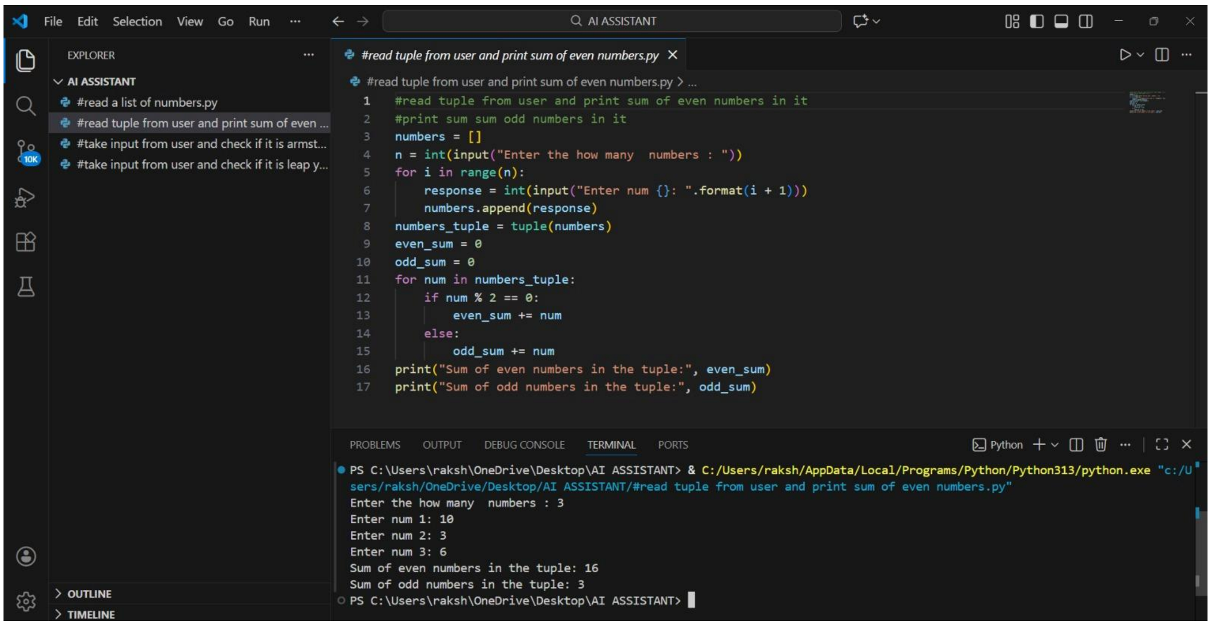
➢ Side-by-side comparison table

➢ Screenshots of prompts and generated code

Prompt :

#take input from user and check if it is armstrong number

Code :


File
Edit
Selection
View
Go
Run
AI ASSISTANT
EXPLORER
AI ASSISTANT
#read a list of numbers.py
#read tuple from user and print sum of even ...
#take input from user and check if it is armst...
#take input from user and check if it is leap y...
OUTLINE
TIMELINE
#read tuple from user and print sum of even numbers.py
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Python

```python
#read tuple from user and print sum of even numbers in it
#print sum sum odd numbers in it
numbers = []
n = int(input("Enter the how many  numbers : "))
for i in range(n):
    response = int(input("Enter num {}: ".format(i + 1)))
    numbers.append(response)
numbers_tuple = tuple(numbers)
even_sum = 0
odd_sum = 0
for num in numbers_tuple:
    if num % 2 == 0:
        even_sum += num
    else:
        odd_sum += num
print("Sum of even numbers in the tuple:", even_sum)
print("Sum of odd numbers in the tuple:", odd_sum)
```

```
PS C:\Users\raksh\OneDrive\Desktop\AI ASSISTANT> & C:/Users/raksh/AppData/Local/Programs/Python/Python313/python.exe "c:/U
sers/raksh/OneDrive/Desktop/AI ASSISTANT/#read tuple from user and print sum of even numbers.py"
Enter the how many  numbers : 3
Enter num 1: 10
Enter num 2: 3
Enter num 3: 6
Sum of even numbers in the tuple: 16
Sum of odd numbers in the tuple: 3
PS C:\Users\raksh\OneDrive\Desktop\AI ASSISTANT>
```

Analysis

1.Both tools gave correct logic.

2.Gemini was easy to understand.

3.Copilot was faster.

Task 3 :

Leap Year Validation Using Cursor AI

❖ **Scenario:**

You are validating a calendar module for a backend system.

❖ **Task:**

Use Cursor AI to generate a Python program that checks whether a given year is a leap year.

Use at least two different prompts and observe changes in code.

❖ **Expected Output:**

➢ **Two versions of code**

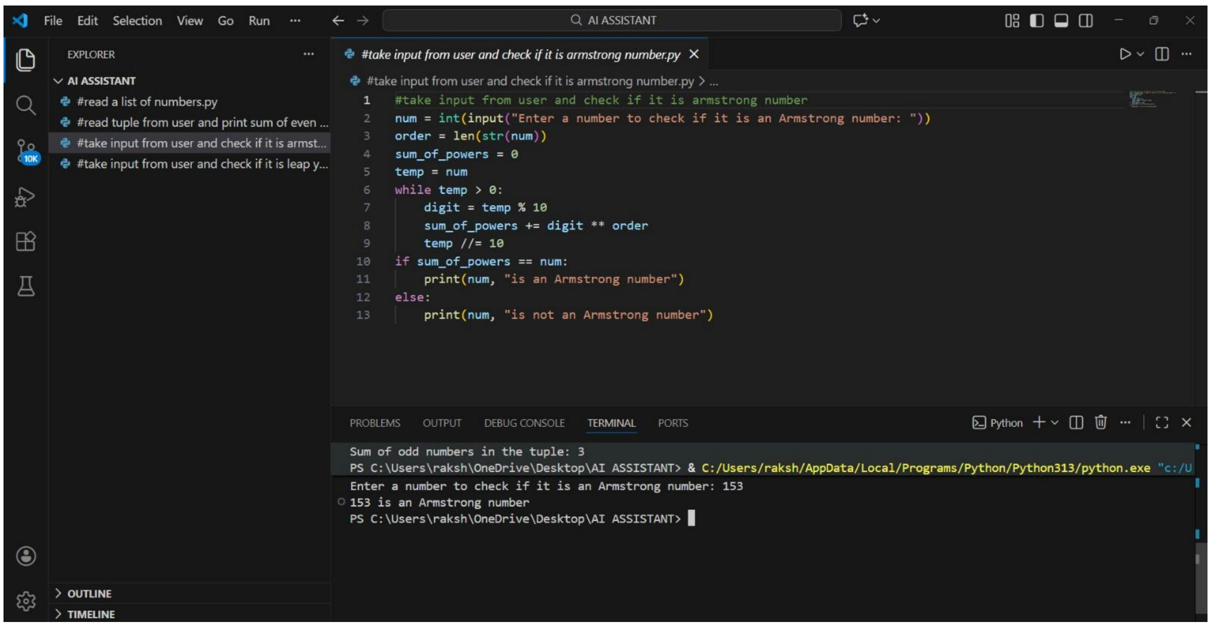
➢ **Sample inputs/outputs**

➢ Brief comparison

Prompt :

#take input from user and check if it is leap year

Code :


File
Edit
Selection
View
Go
Run
AI ASSISTANT
EXPLORER
AI ASSISTANT
#read a list of numbers.py
#read tuple from user and print sum of even ...
#take input from user and check if it is armst...
#take input from user and check if it is leap y...
#take input from user and check if it is armstrong number.py
OUTLINE
TIMELINE
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Python

```python
#take input from user and check if it is armstrong number
num = int(input("Enter a number to check if it is an Armstrong number: "))
order = len(str(num))
sum_of_powers = 0
temp = num
while temp > 0:
    digit = temp % 10
    sum_of_powers += digit ** order
    temp //= 10
if sum_of_powers == num:
    print(num, "is an Armstrong number")
else:
    print(num, "is not an Armstrong number")
```

```
Sum of odd numbers in the tuple: 3
PS C:\Users\raksh\OneDrive\Desktop\AI ASSISTANT> & C:/Users/raksh/AppData/Local/Programs/Python/Python313/python.exe "c:/U
Enter a number to check if it is an Armstrong number: 153
153 is an Armstrong number
PS C:\Users\raksh\OneDrive\Desktop\AI ASSISTANT> 
```

Analysis

1.Cursor AI used with VS Code.

2. Different prompts gave different code.

3. Logic was correct.

Task 4 :

Student Logic + AI Refactoring (Odd/Even Sum)

❖ Scenario:

Company policy requires developers to write logic before using AI.

❖ Task:

Write a Python program that calculates the sum of odd and even numbers in a tuple, then refactor it using any AI tool.

❖ Expected Output:

➢ **Original code**

➢ **Refactored code**

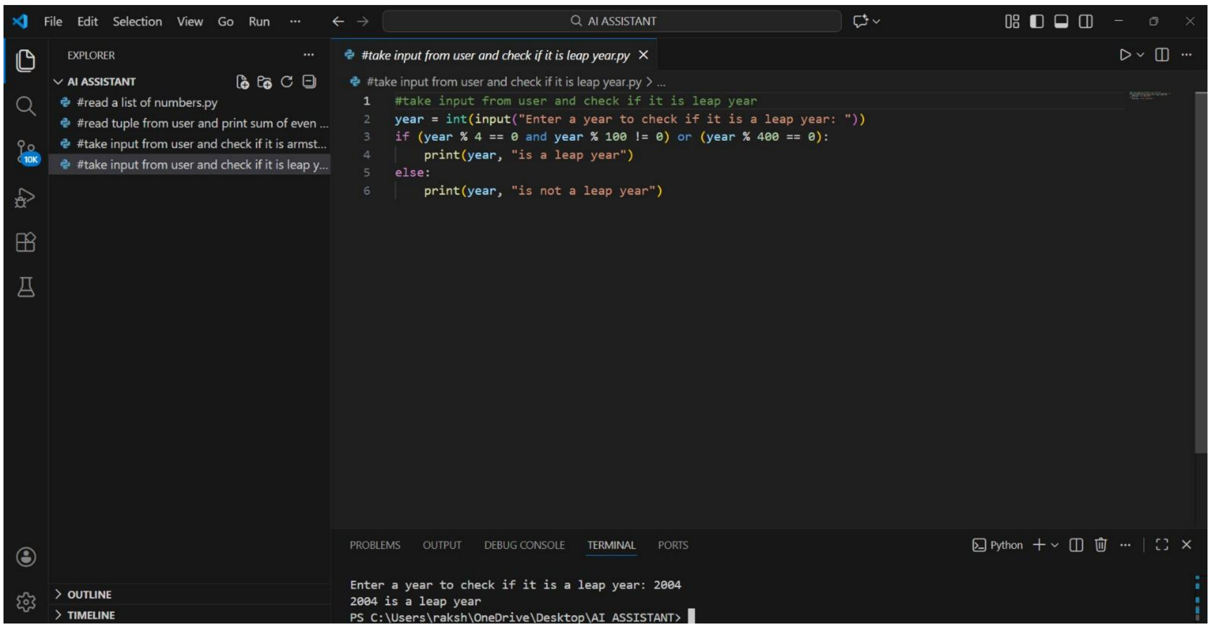
➢ **Explanation of improvements**

Prompt :

#read tuple from user and print sum of even numbers in it

#print sum sum odd numbers in it

Code :


File Edit Selection View Go Run ···
AI ASSISTANT
EXPLORER
AI ASSISTANT
#read a list of numbers.py
#read tuple from user and print sum of even ...
#take input from user and check if it is armst...
#take input from user and check if it is leap y...
#take input from user and check if it is leap year.py
OUTLINE
TIMELINE
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Python

```python
#take input from user and check if it is leap year
year = int(input("Enter a year to check if it is a leap year: "))
if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):
    print(year, "is a leap year")
else:
    print(year, "is not a leap year")
```

```
Enter a year to check if it is a leap year: 2004
2004 is a leap year
PS C:\Users\raksh\OneDrive\Desktop\AI ASSISTANT>
```

Analysis :

1. Code written manually in VS Code.

2. AI refactored the code.

3. Code became clean and readable