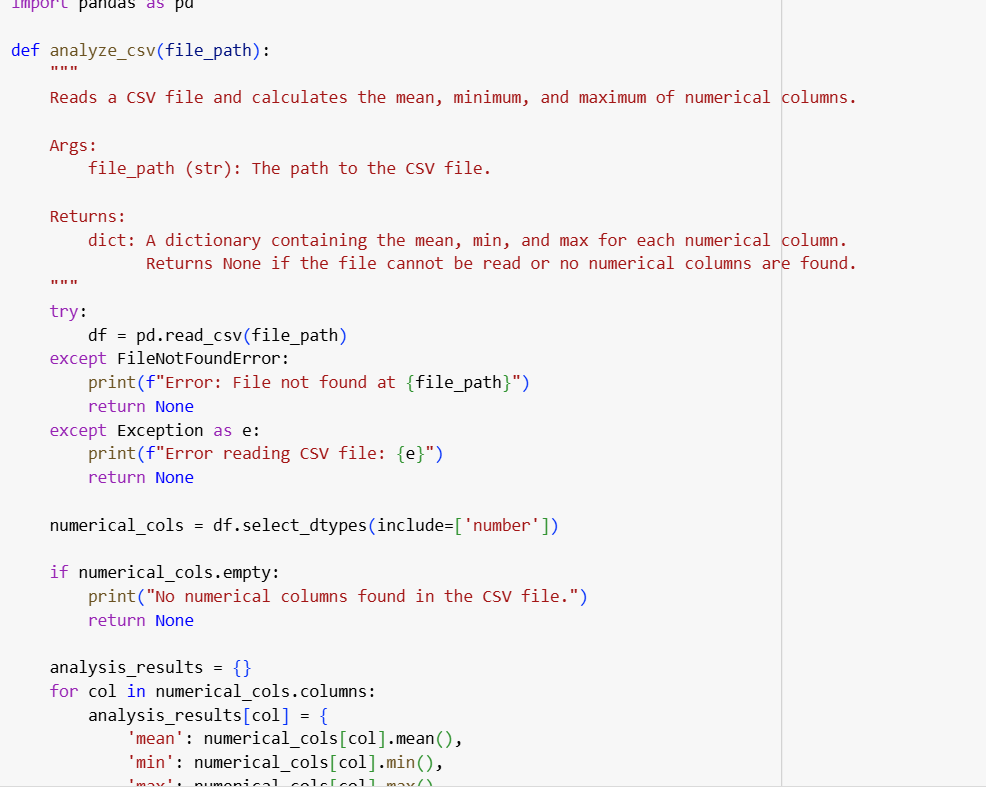
Task Description#1

● Use Google Gemini in Colab to write a function that reads a CSV file and calculates

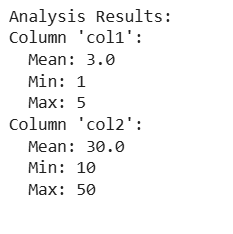
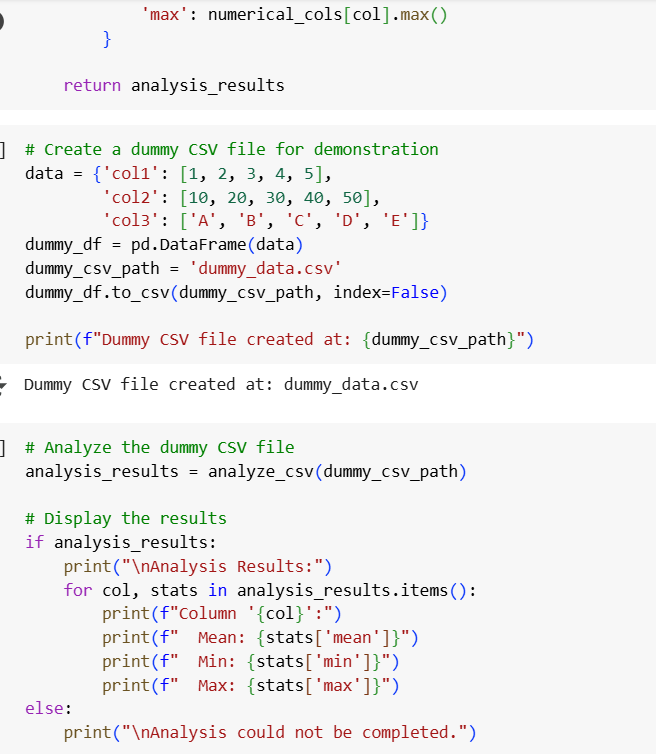
mean, min, max.

Expected Output#1

● Functional code with output and screenshot

By creating the dummy file it reads the code automatically

Here dummy\_data.cv is the dummy file



Task Description#2

● Compare Gemini and Copilot outputs for a palindrome check function.

Expected Output#2

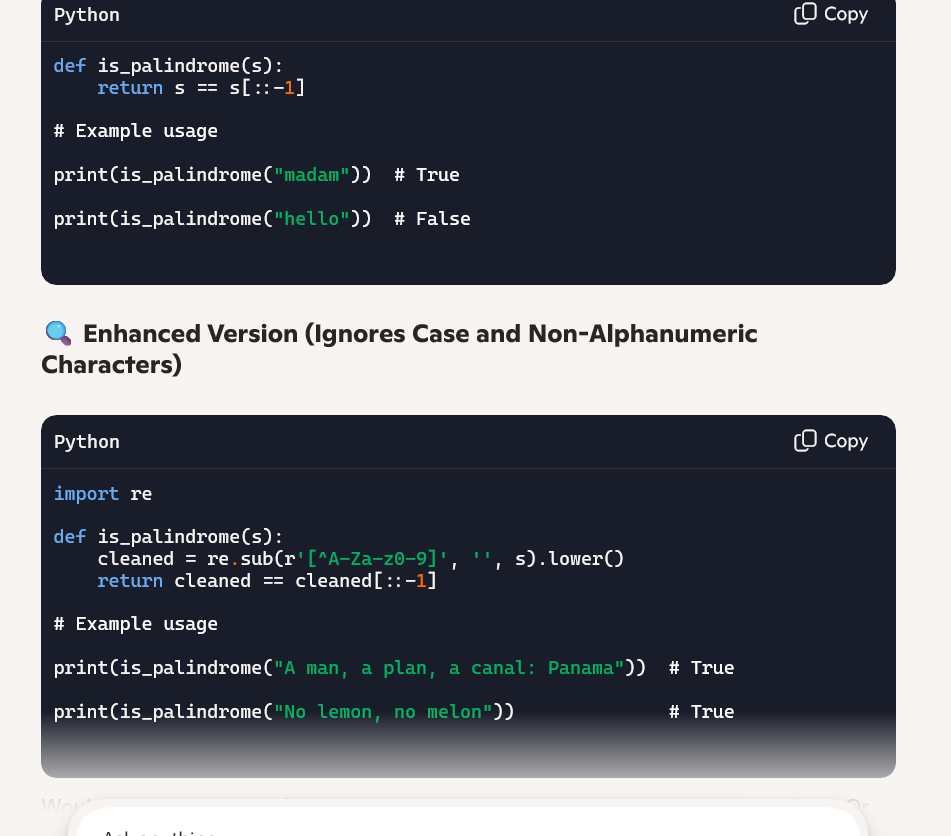
● Side-by-side comparison and observations

Sol: The code and the output given by gemini.ai is



Palindrome function check given by copilot

It gives inbuilt examples in copilot whereas in geminiai it doesn’t shows the inbuild examples we need to specify them



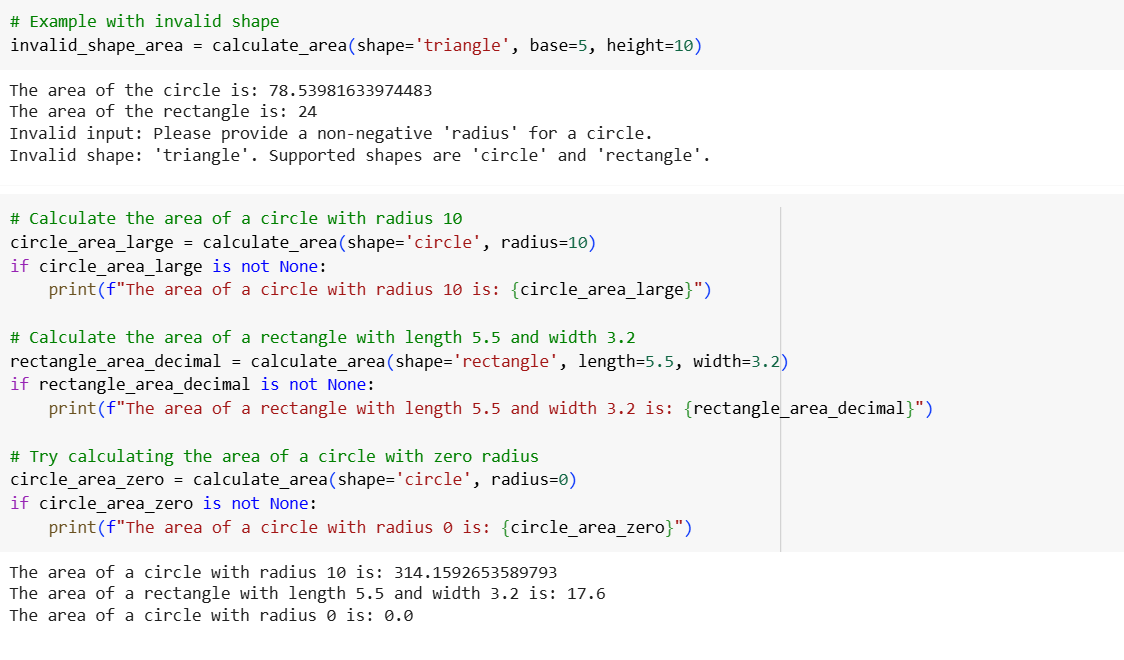
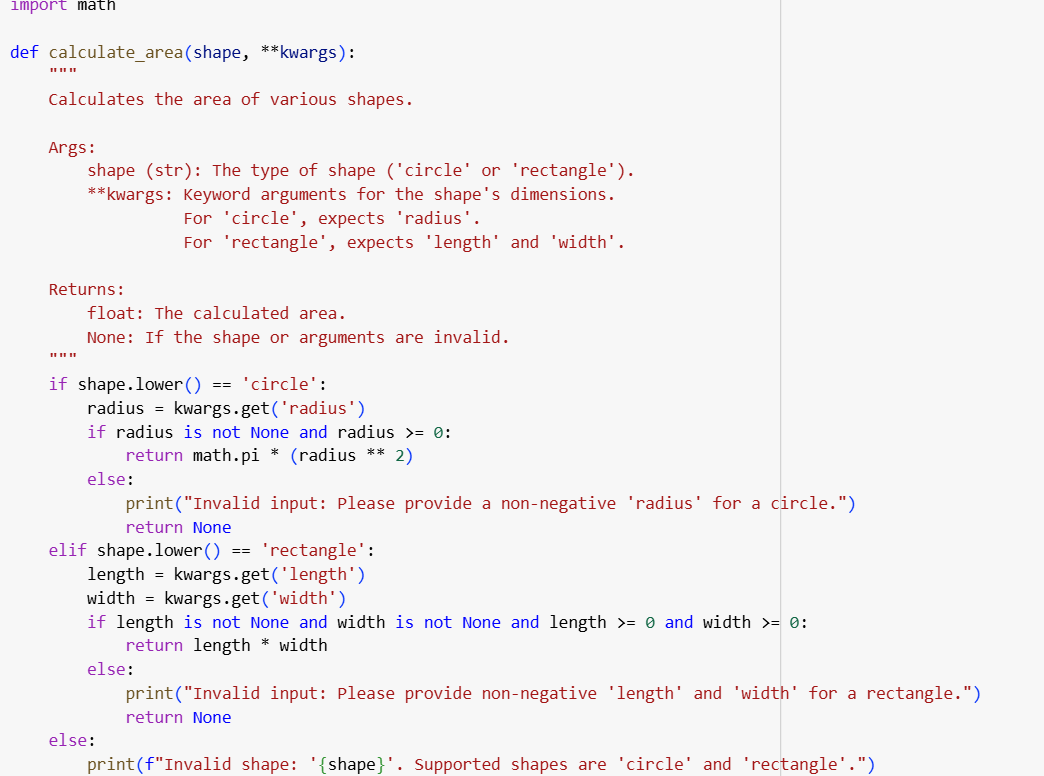
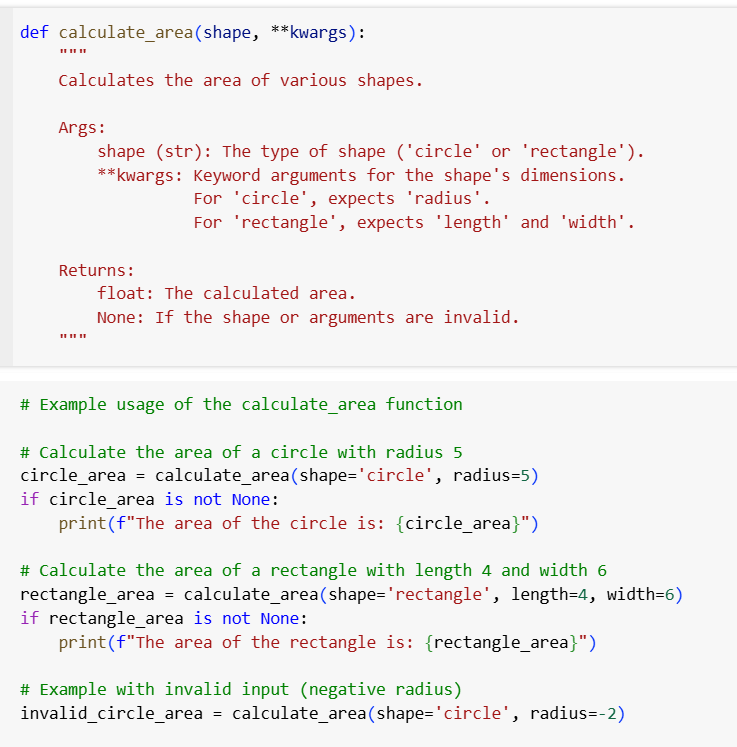
Task Description#3

● Ask Gemini to explain a Python function (to calculate area of various shapes) line by

line..

Expected Output#3

● Detailed explanation with code snippet



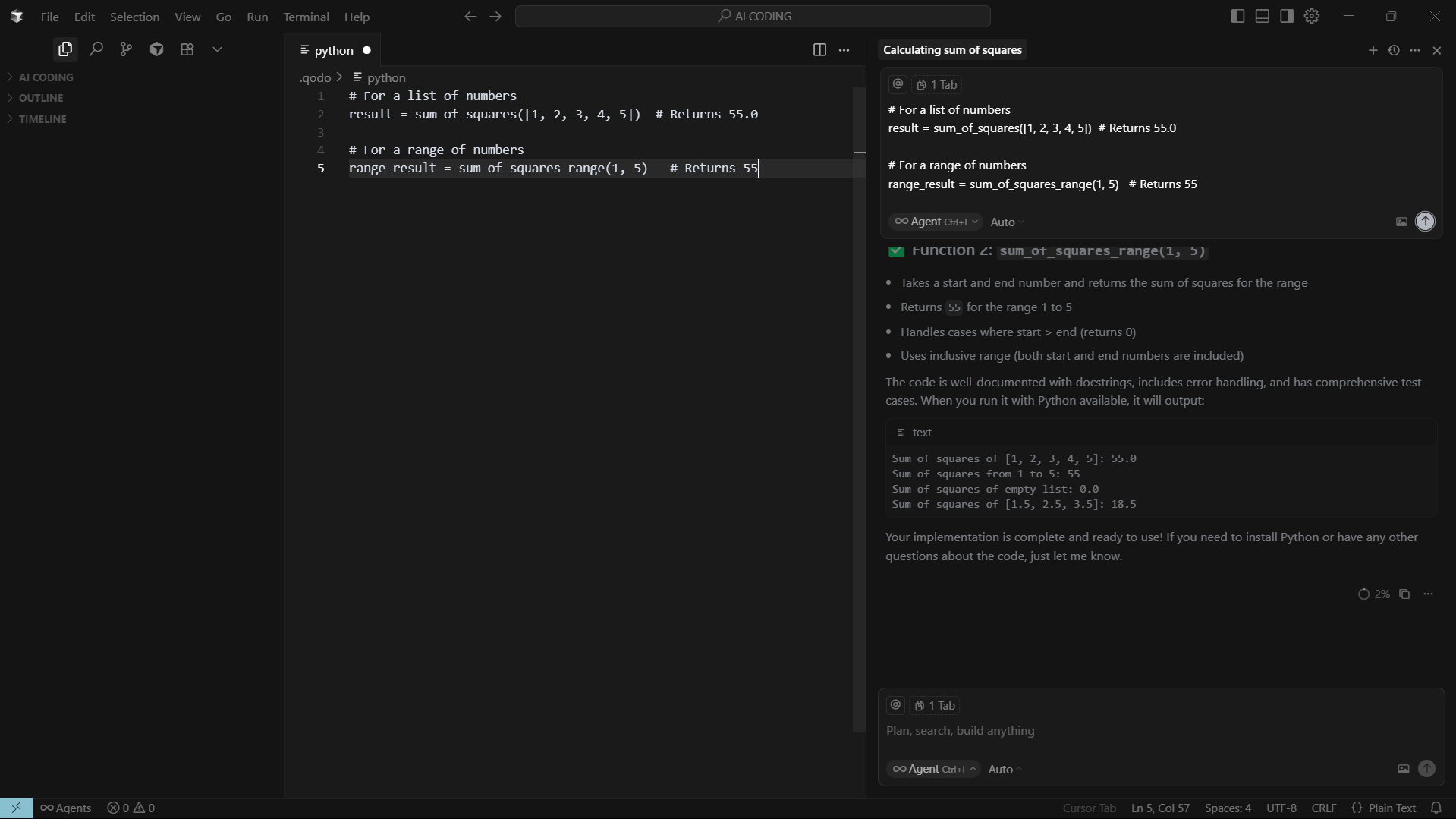
Task Description#4

● Install and configure Cursor AI. Use it to generate a Python function (e.g., sum of

squares).

Expected Output#4

● Screenshots of working environments with few prompts to generate python code



Task Description#5

● Student need to write code to calculate sum of add number and even numbers in the

list

Expected Output#5

● Refactored code written by student with improved logic

Code:

def sum\_odd\_even(numbers):

    """

    Calculates the sum of odd and even numbers in a list.

    Args:

        numbers (list): A list of numbers.

    Returns:

        tuple: A tuple containing the sum of odd numbers and the sum of even numbers (sum\_odd, sum\_even).

    """

    sum\_odd = 0

    sum\_even = 0

    for number in numbers:

        if isinstance(number, (int, float)): # Check if the element is a number

            if number % 2 == 0:

                sum\_even += number

            else:

                sum\_odd += number

        else:

            print(f"Warning: Skipping non-numeric element: {number}")

    return sum\_odd, sum\_even