SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE		DEPARTMENT OF COMPUTER SCIENCE ENGINEERING	
Program Name: M. Tech/MCA		**Assignment Type: Lab**	**AcademicYear:**2025-2026
Course Coordinator Name		Venkataramana Veeramsetty	
Course Code		**Course Title**	AI Assisted Problem Solving Using Python
Year/Sem	I/I	**Regulation**	R24
Date and Day of Assignment	Week1 - TUESDAY	**Time(s)**	
Duration	2 Hours	**Applicable to Batches**	M. Tech/MCA
AssignmentNumber:2.3(Present assignment number)/**24**(Total number of assignments)			

Q.No.	Question	*Expected Time to complete*
1	Lab 2: Exploring Additional AI Coding Tools – Gemini (Colab) and Cursor AI **Lab Objectives:** ● To explore and evaluate the functionality of Google Gemini for AI-assisted coding within Google Colab. ● To understand and use Cursor AI for code generation, explanation, and refactoring. ● To compare outputs and usability between Gemini, GitHub Copilot, and Cursor AI. ● To perform code optimization and documentation using AI tools. **Lab Outcomes (LOs):** After completing this lab, students will be able to: ● Generate Python code using Google Gemini in Google Colab. ● Analyze the effectiveness of code explanations and suggestions by Gemini. ● Set up and use Cursor AI for AI-powered coding assistance. ● Evaluate and refactor code using Cursor AI features. ● Compare AI tool behavior and code quality across different platforms. **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	Week1 - TuesDay

Task Description#2

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

Expected Output#2

- Side-by-side comparison and observations

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

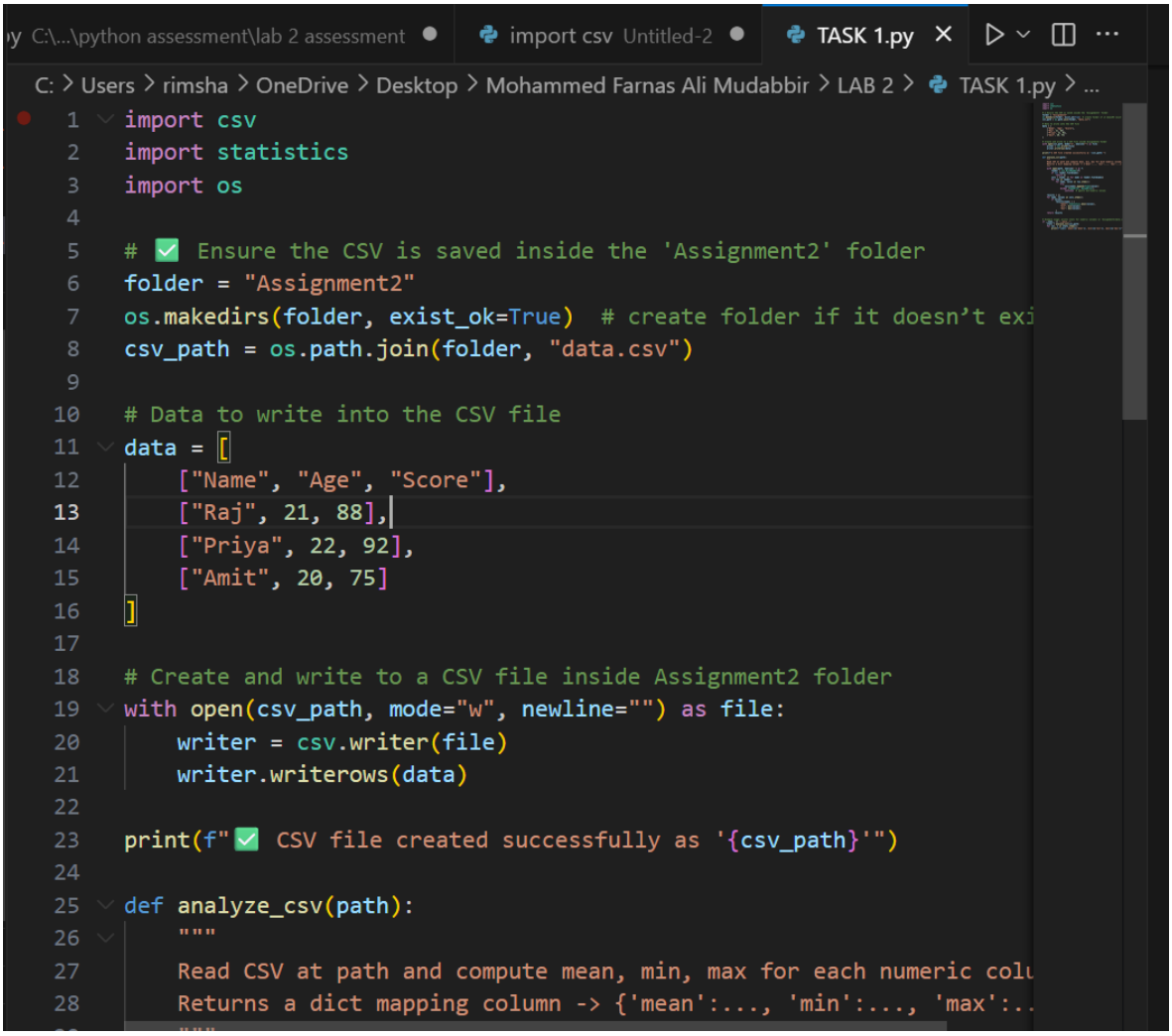
Note: Report should be submitted a word document for all tasks in a single document with prompts, comments & code explanation, and output and if required, screenshots

Evaluation Criteria:

Criteria	Max Marks
Successful Use of Gemini in Colab (Task#1 & #2)	2.5
Code Explanation Accuracy (Gemini) (Task#3)	2.5
Cursor AI Setup and Usage (Task#4)	2.5
Refactoring and Improvement Analysis (Task#5)	2.5
Total	**10 Marks**

TASK DESCRIPTION -1

Use Google Gemini in Colab to write a function that reads a CSV file and calculates mean, min, max.



```
    """
    with open(path, newline='') as f:
        reader = csv.DictReader(f)
        if not reader.fieldnames:
            return {}
        cols = {name: [] for name in reader.fieldnames}
        for row in reader:
            for name, value in row.items():
                try:
                    cols[name].append(float(value))
                except (TypeError, ValueError):
                    continue  # ignore non-numeric values

    results = {}
    for name, values in cols.items():
        if values:
            results[name] = {
                "mean": statistics.mean(values),
                "min": min(values),
                "max": max(values),
            }
    return results


# Example usage: prints stats for numeric columns in 'Assignment2/dat
if __name__ == "__main__":
    stats = analyze_csv(csv_path)
```

C: > Users > rimsha > OneDrive > Desktop > Mohammed Farnas Ali Mudabbir > LAB 2 > TASK 1.py > ...

```
def analyze_csv(path):
    for name, values in cols.items():
        if values:
            results[name] = {
                "mean": statistics.mean(values),
                "min": min(values),
                "max": max(values),
            }
    return results


# Example usage: prints stats for numeric columns in 'Assignment2/dat
if __name__ == "__main__":
    stats = analyze_csv(csv_path)
    for col, s in stats.items():
        print(f"{col}: mean={s['mean']}, min={s['min']}, max={s['max'
```

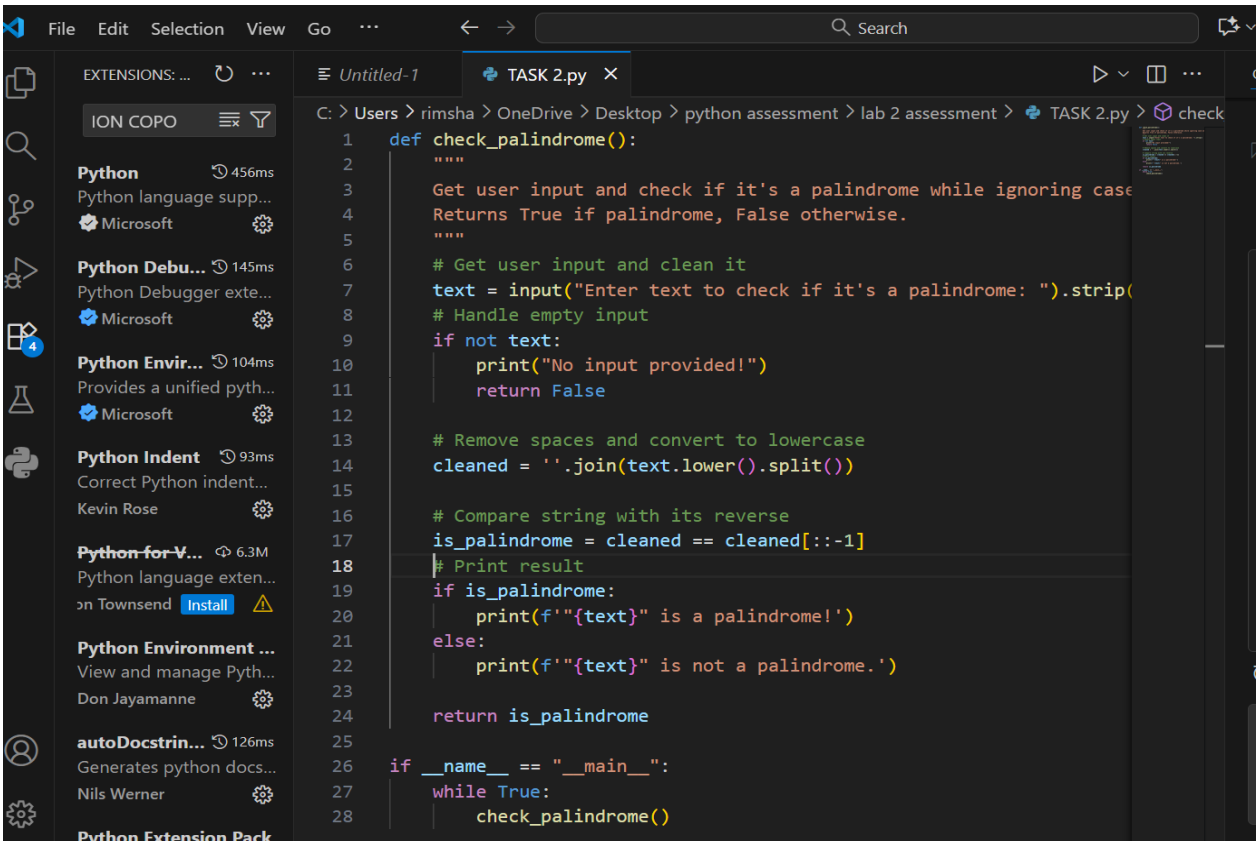
Expected Output
Functional code with output and screenshot

Practical Output:

```
PS C:\Users\rimsha> python -u "c:\Users\rimsha\OneDrive\Desktop\Moha
mmed Farnas Ali Mudabbir\LAB 2\TASK 1.py"
✅ CSV file created successfully as 'Assignment2\data.csv'
Age: mean=21.0, min=20.0, max=22.0
Score: mean=85.0, min=75.0, max=92.0
PS C:\Users\rimsha>
```

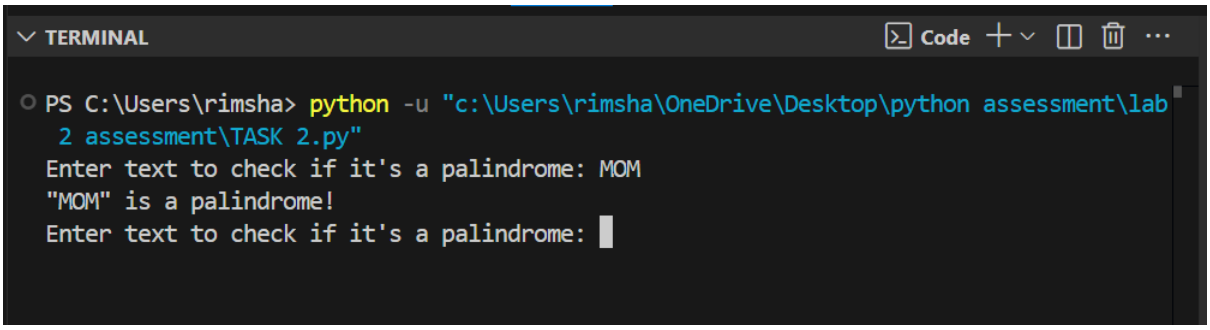
TASK DESCRIPTION -2

Compare Gemini and Copilot outputs for a palindrome check function.
Prompt: write a user input palindrome function

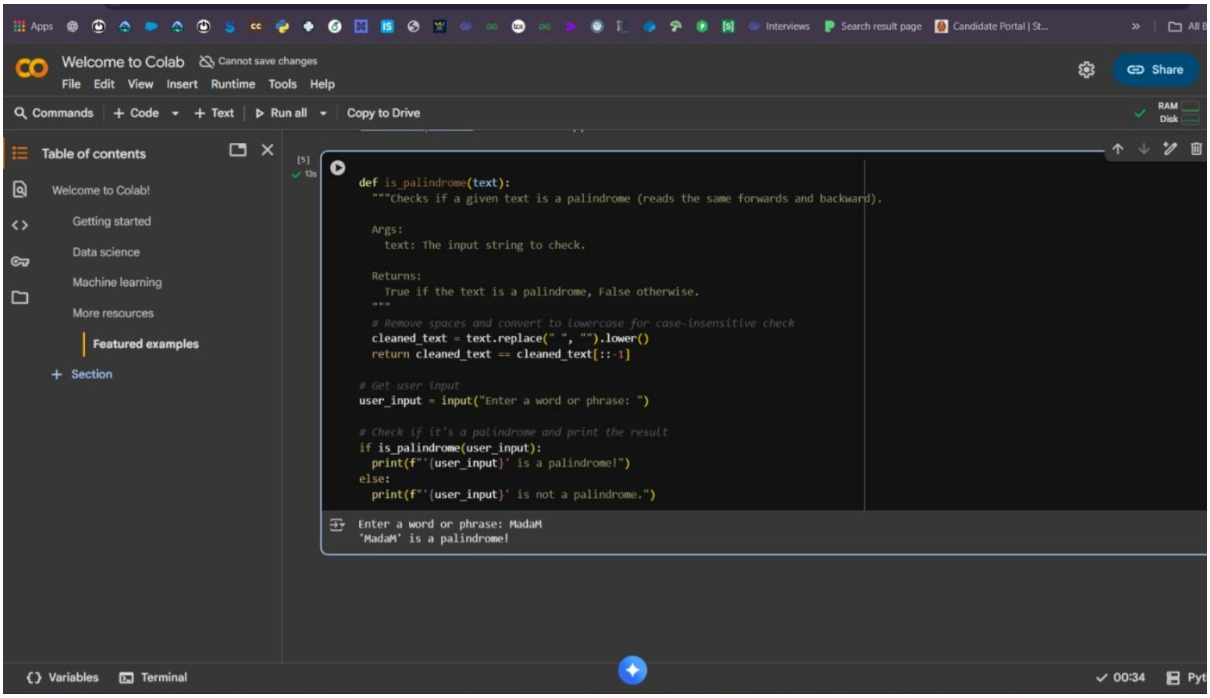


Expected Output:
Side-by-side comparison and observations

Practical Output:



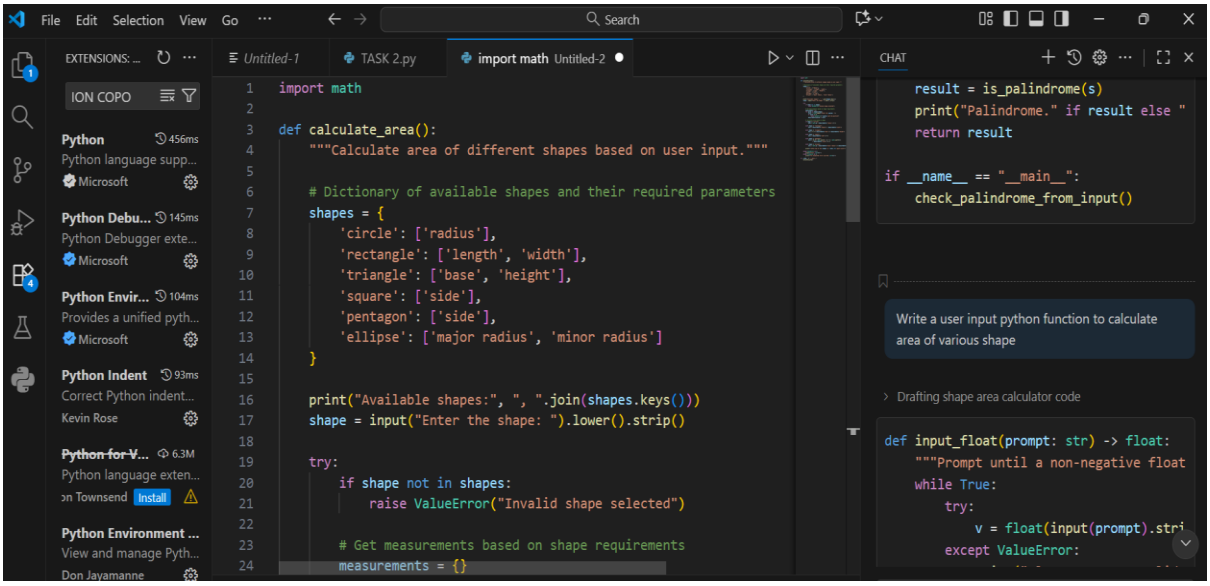
Google colab:

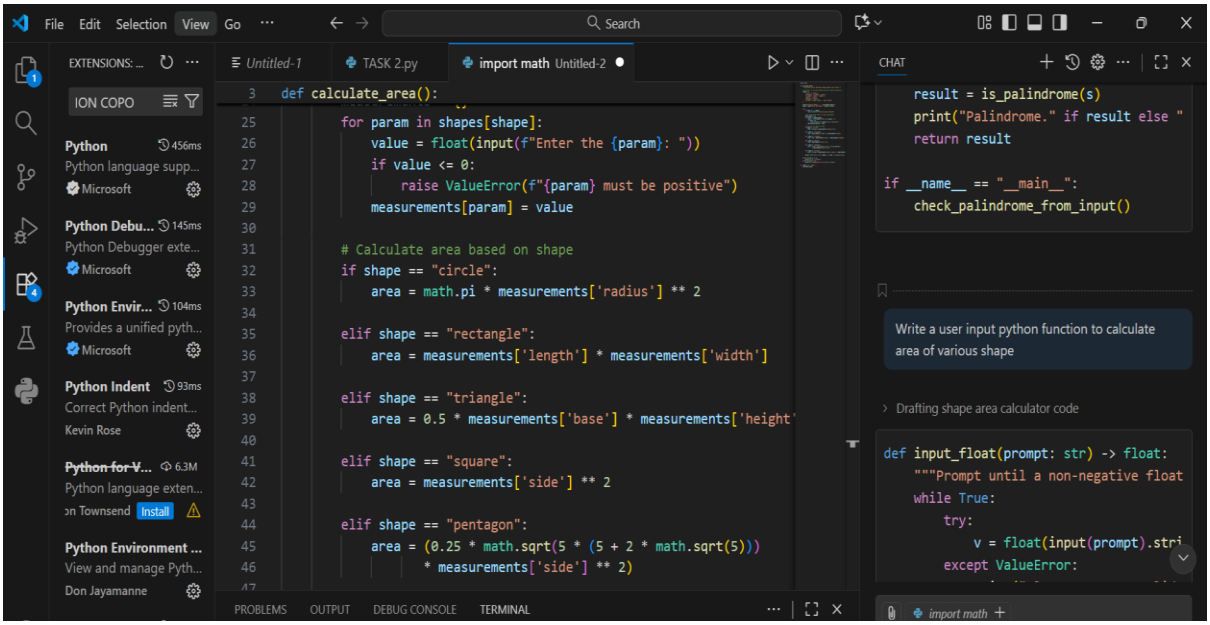


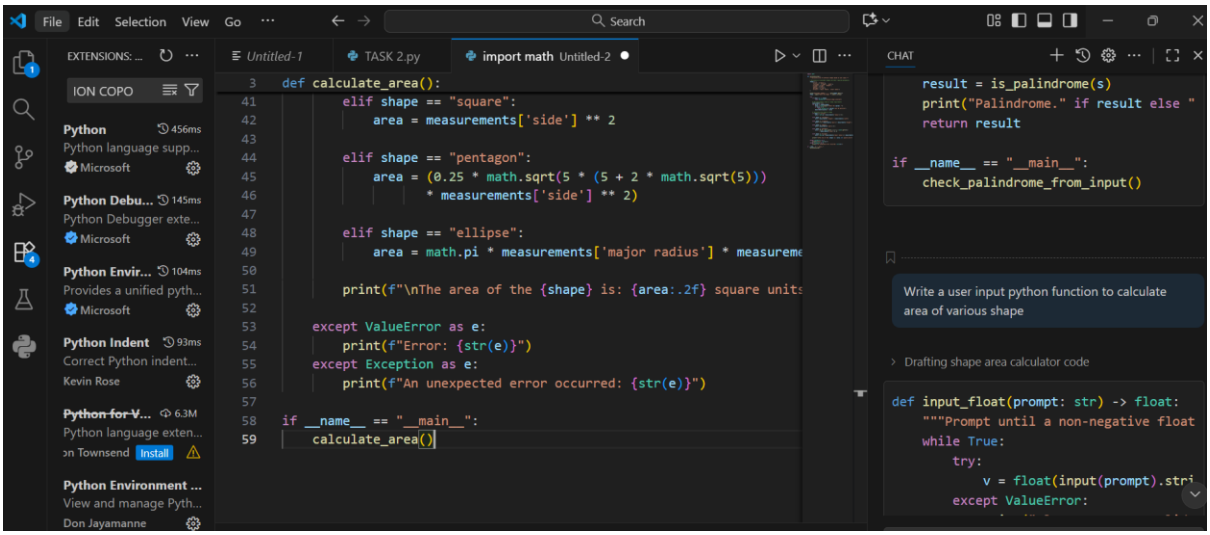
TASK DESCRIPTION -3

Detailed explanation with code snippet

Prompt: Write a user input python function to calculate area of various shape



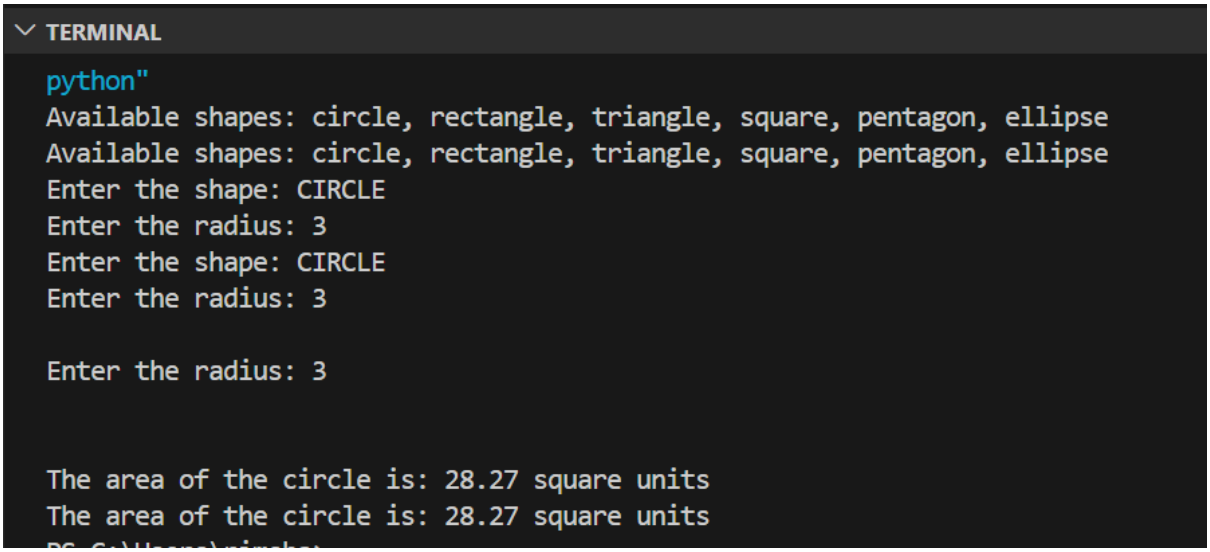




Expected Output:

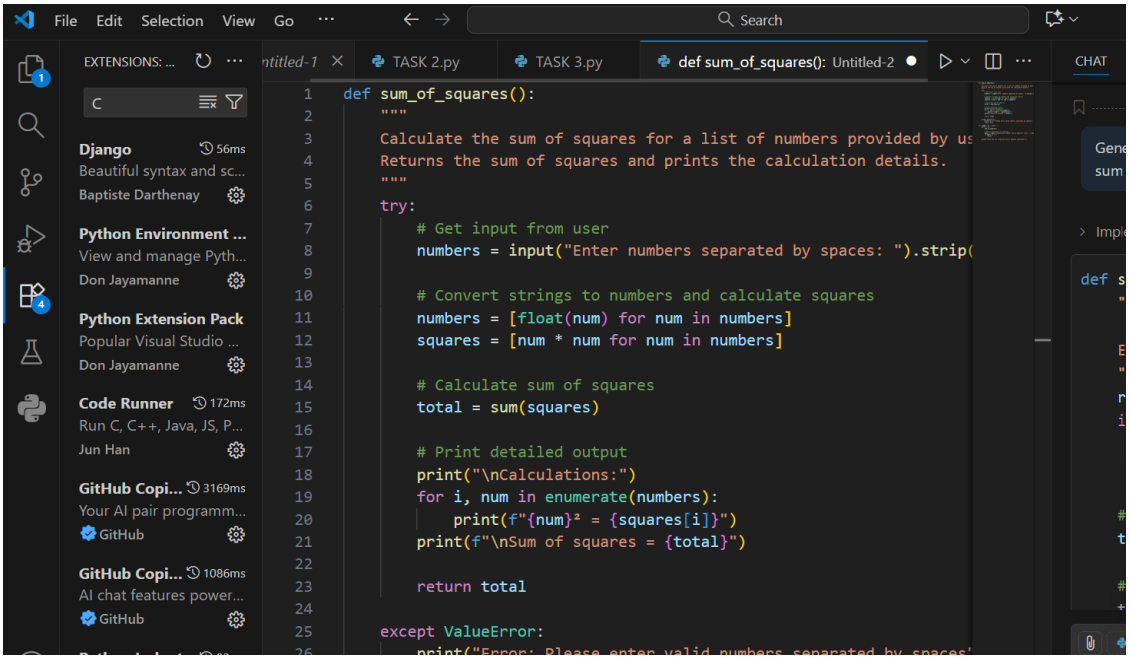
Detailed explanation with code snippet

Practical Output:



TASK DESCRIPTION -4

Install and configure Cursor AI. Use it to generate a Python function (e.g., sum of squares).
Prompt: Generate a user input python function to calculate sum of square

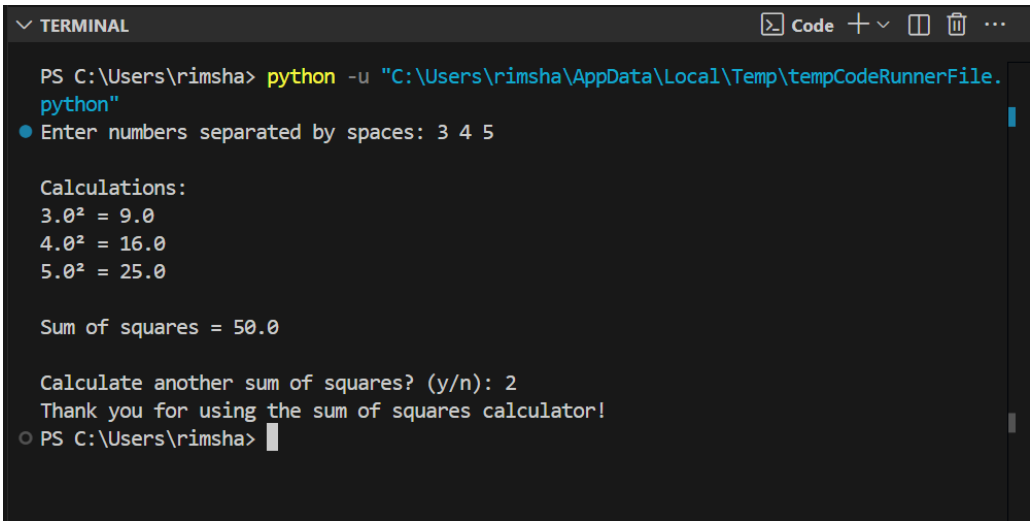



```
        for i, num in enumerate(numbers):
            print(f"{num}² = {squares[i]}")
        print(f"\nSum of squares = {total}")

        return total

    except ValueError:
        print("Error: Please enter valid numbers separated by spaces")
        return None

if __name__ == "__main__":
    while True:
        sum_of_squares()

        # Ask if user wants to continue
        again = input("\nCalculate another sum of squares? (y/n): ")
        if again != 'y':
            break

    print("Thank you for using the sum of squares calculator!")
```

Expected Output

Screenshots of working environments with few prompts to generate python code

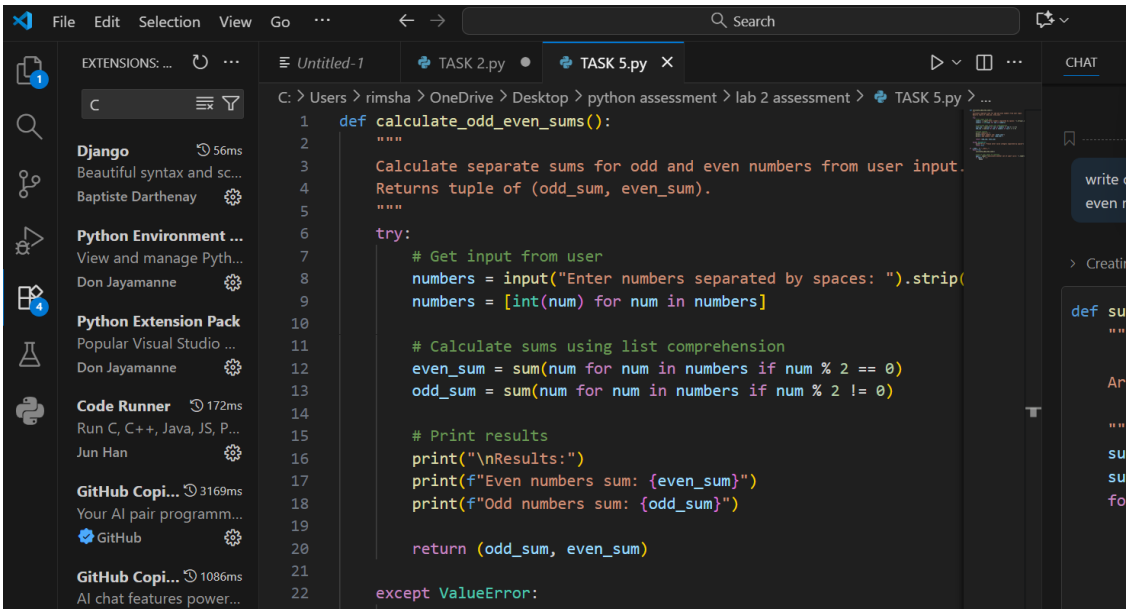
Practical Output:

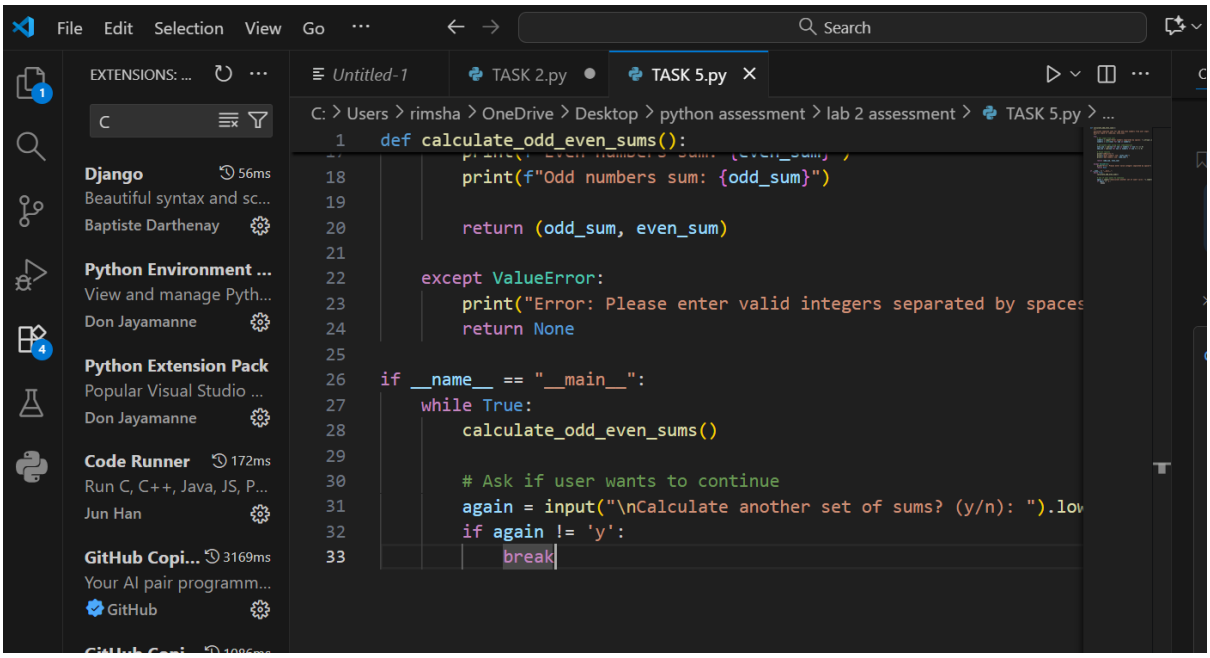


TASK DESCRIPTION -5

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

Prompt: write code to calculate sum of add number and even numbers in the list





Expected Output:
Refactored code written by student with improved logic

Practical Output:

```
python"
Enter numbers separated by spaces: 4 7 2

Results:
Even numbers sum: 6
Odd numbers sum: 7

Results:
Even numbers sum: 6
Even numbers sum: 6
Odd numbers sum: 7
```