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**	Week1 - TuesDay

- Functional code with output and screenshot

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

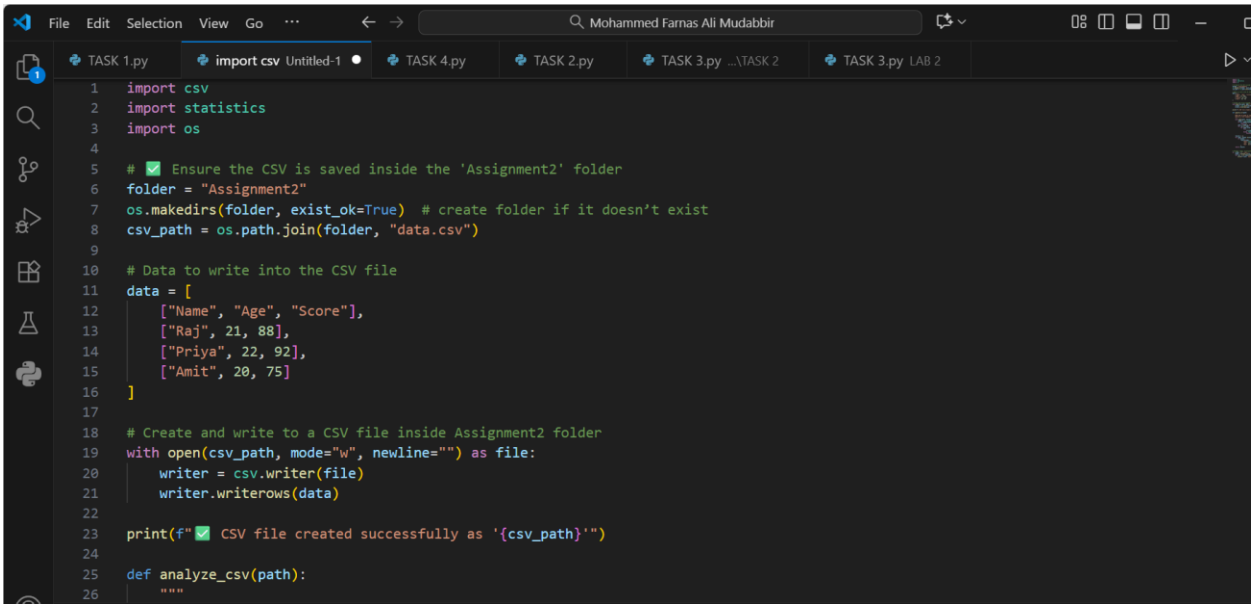
Note: Report should be submitted a word document for all tasks in a single document with prompts, comments & code explanation, and output and if required, screenshots

Evaluation Criteria:

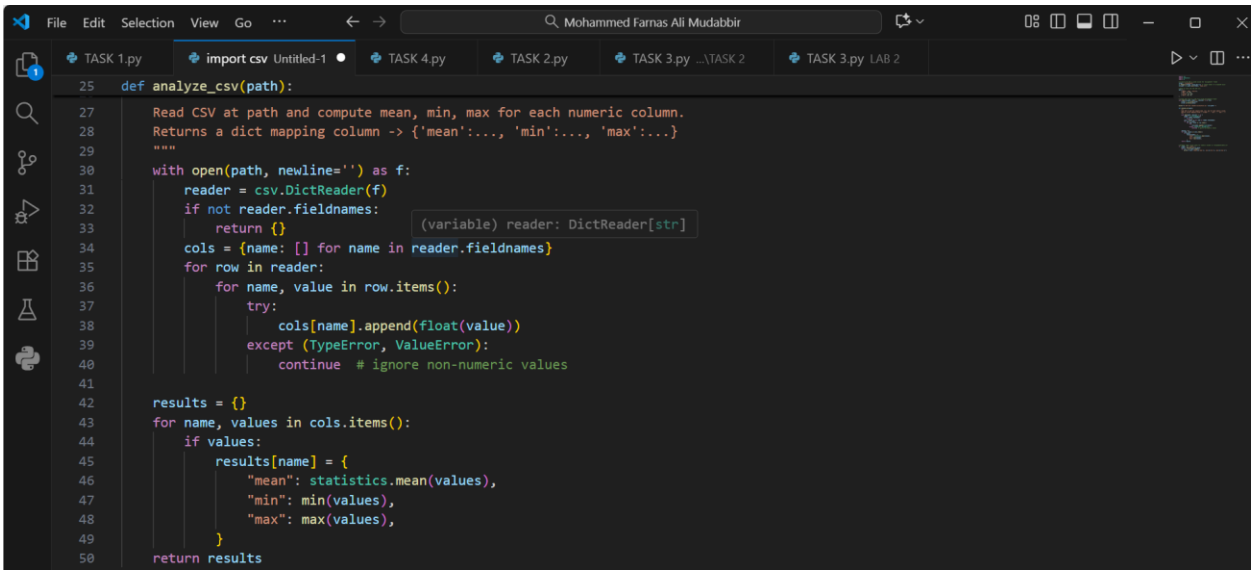
Criteria	Max Marks
Successful Use of Gemini in Colab (Task#1 & #2)	2.5
Code Explanation Accuracy (Gemini) (Task#3)	2.5
Cursor AI Setup and Usage (Task#4)	2.5
Refactoring and Improvement Analysis (Task#5)	2.5
Total	**10 Marks**

TASK DESCRIPTION -1

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



```python
import csv
import statistics
import os

# ✅ Ensure the CSV is saved inside the 'Assignment2' folder
folder = "Assignment2"
os.makedirs(folder, exist_ok=True)  # create folder if it doesn't exist
csv_path = os.path.join(folder, "data.csv")

# Data to write into the CSV file
data = [
    ["Name", "Age", "Score"],
    ["Raj", 21, 88],
    ["Priya", 22, 92],
    ["Amit", 20, 75]
]

# Create and write to a CSV file inside Assignment2 folder
with open(csv_path, mode="w", newline="") as file:
    writer = csv.writer(file)
    writer.writerows(data)

print(f"✅ CSV file created successfully as '{csv_path}'")

def analyze_csv(path):
    """
```



```python
def analyze_csv(path):
    Read CSV at path and compute mean, min, max for each numeric column.
    Returns a dict mapping column -> {'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
```

```
            }
    return results


# Example usage: prints stats for numeric columns in 'Assignment2/data.csv'
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

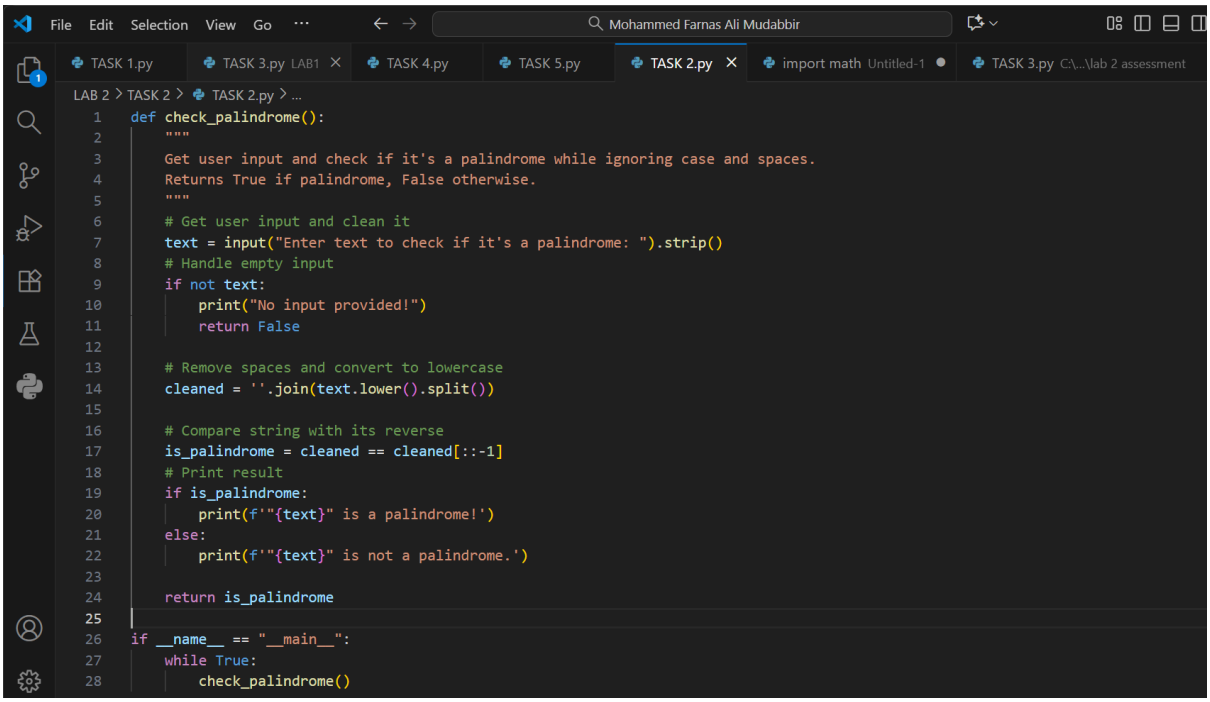
Practical output:

```
dabbir\tempCodeRunnerFile.python"
dabbir\tempCodeRunnerFile.python"
✅ CSV file created successfully as 'Assignment2\data.csv'
✅ 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
Score: mean=85.0, min=75.0, max=92.0
PS C:\Users\rimsha\OneDrive\Desktop\Mohammed Farnas Ali Mudabbir>
```

TASK DESCRIPTION -2

Compare Gemini and Copilot outputs for a palindrome check function.

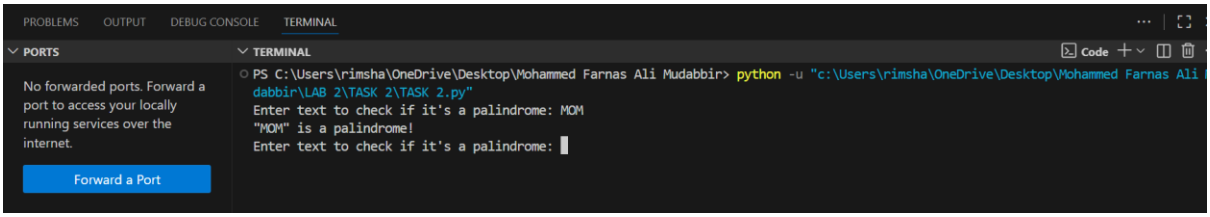
Prompt: write a user input palindrome function



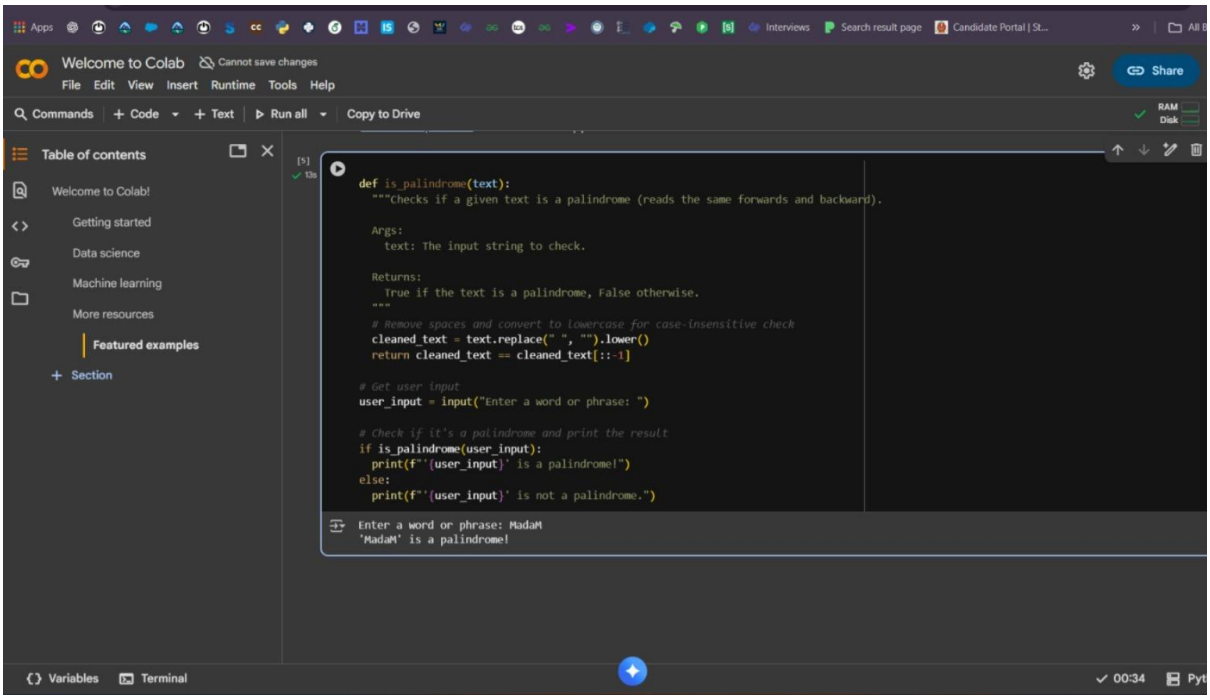
Expected Output:

Side-by-side comparison and observations

Practical output:



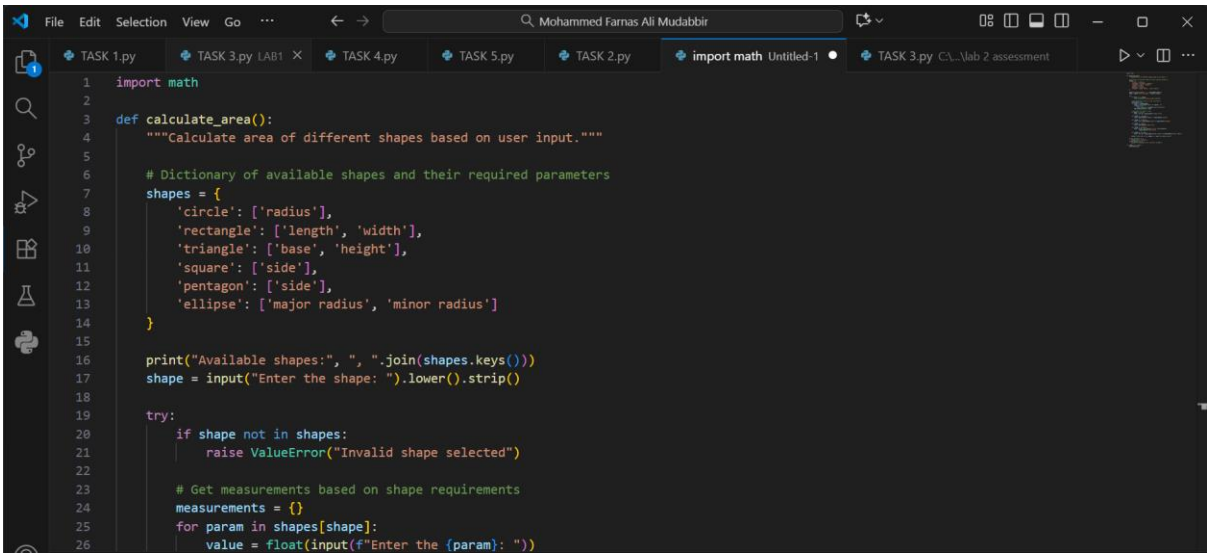
Google colab:

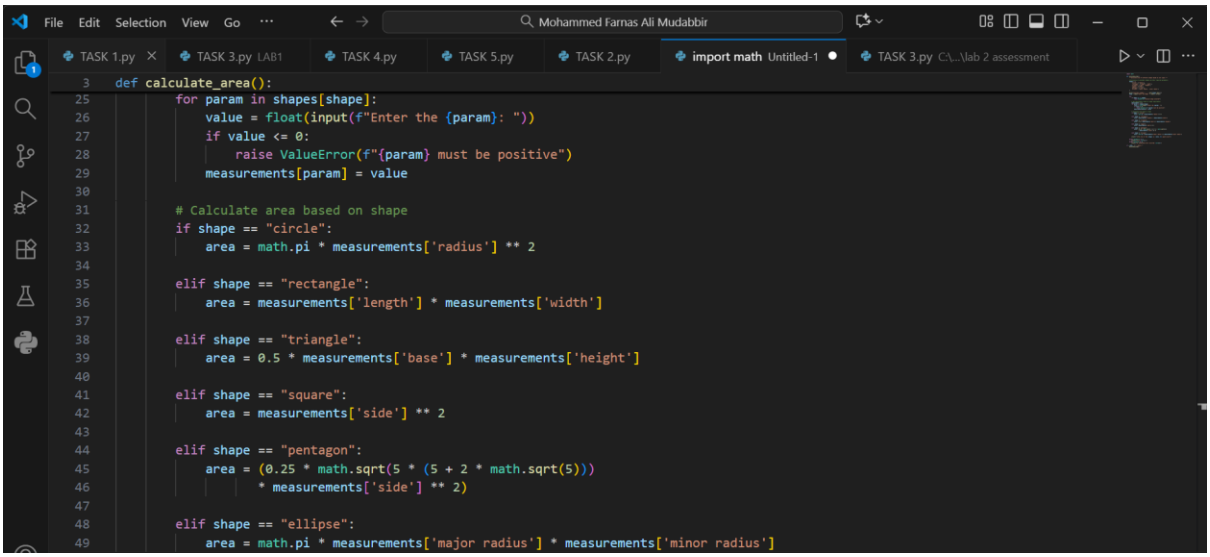


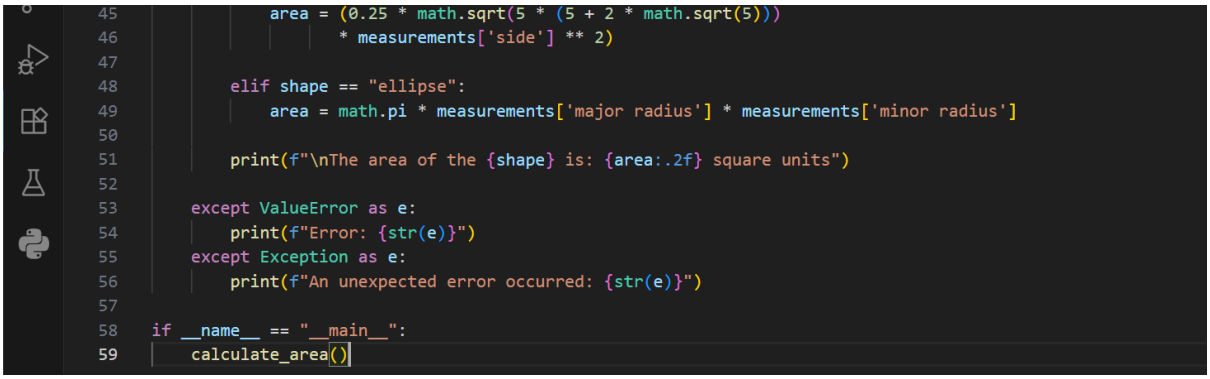
TASK DESCRIPTION -3

Detailed explanation with code snippet

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



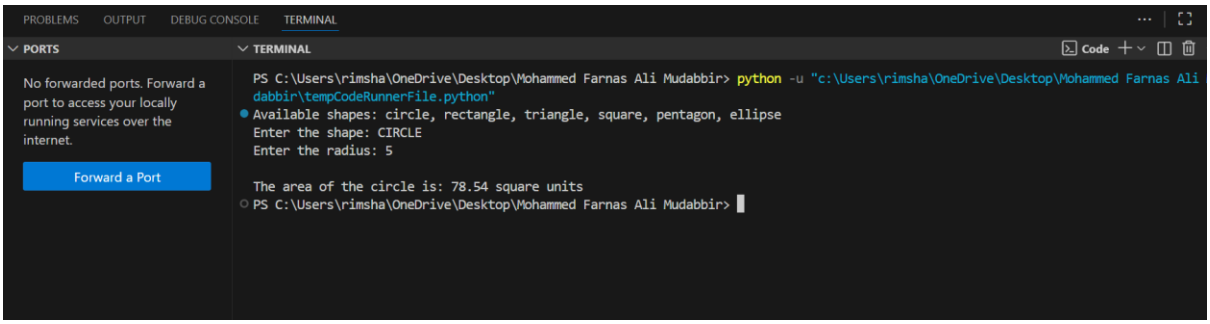




Expected Output:

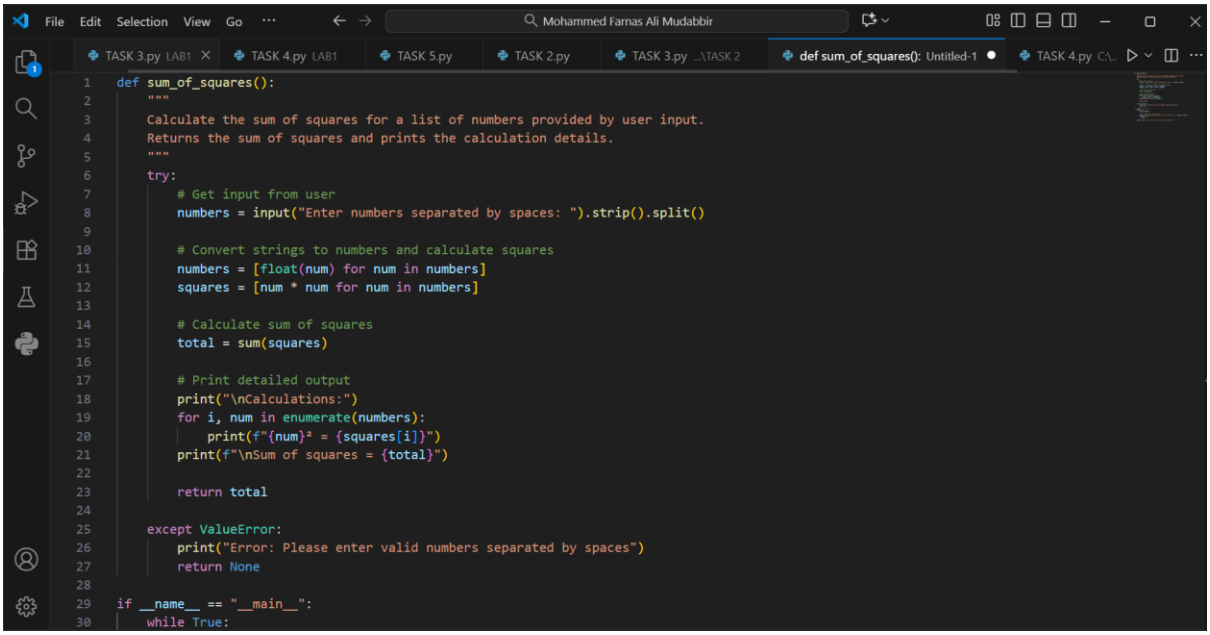
Detailed explanation with code snippet

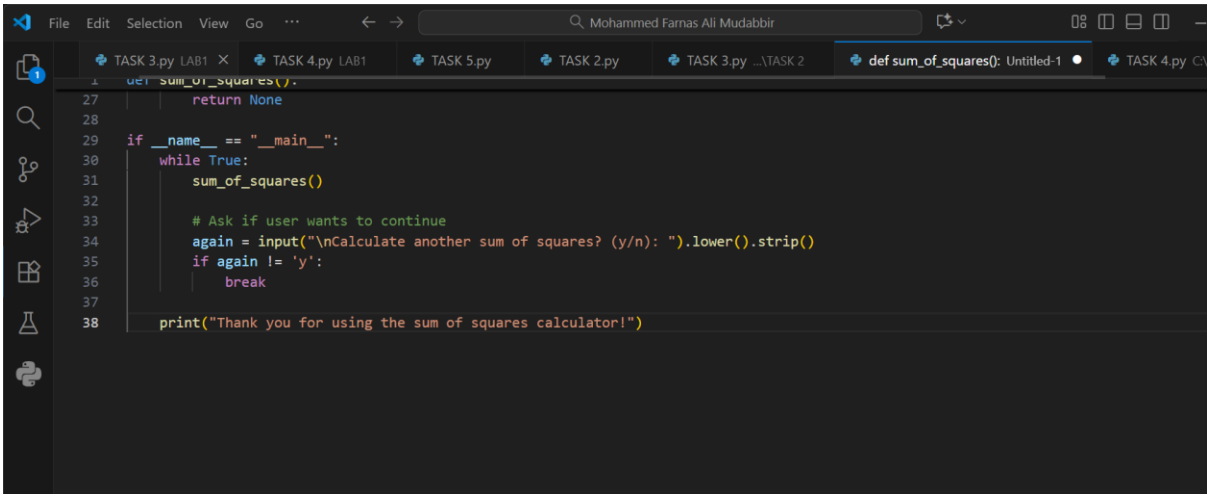
Practical output:



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





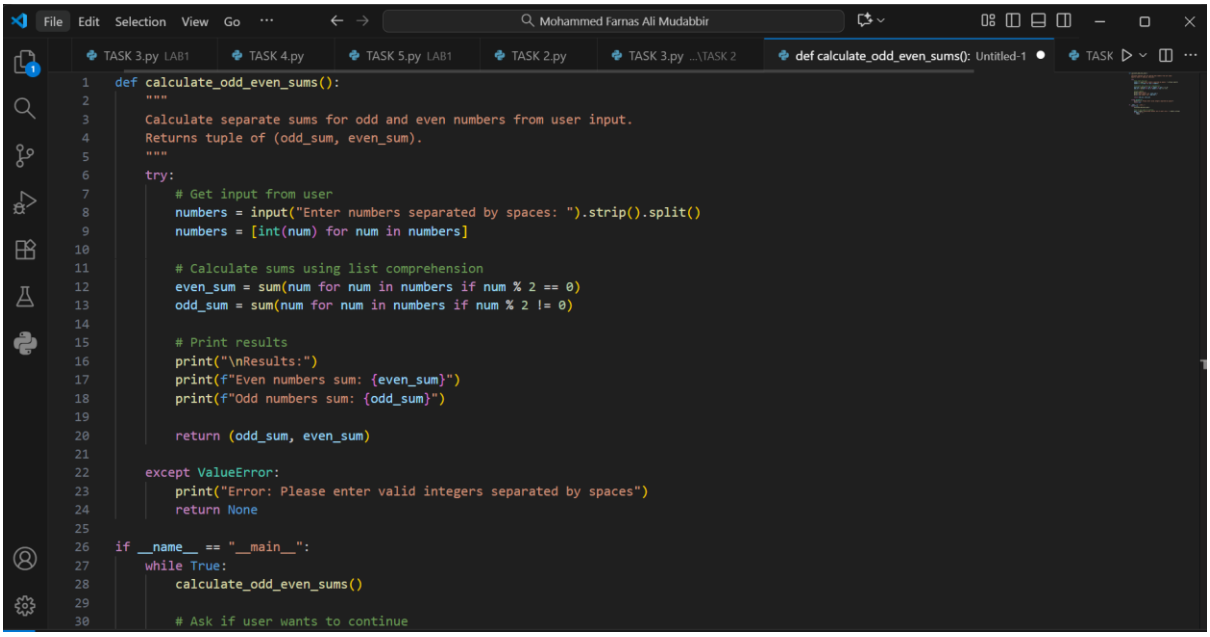
Expected Output

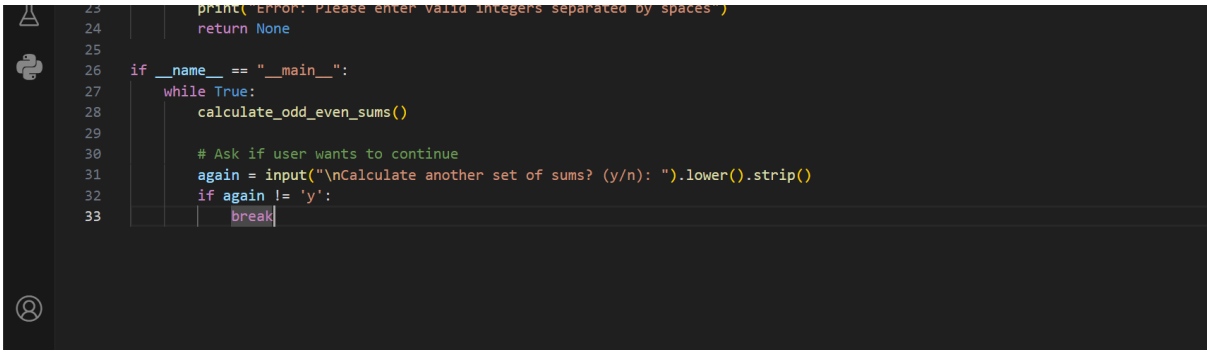
Screenshots of working environments with few prompts to generate python code

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: