COURSE: AI Assisted Coding

NAME : PRAVALIKA MUTHOJU

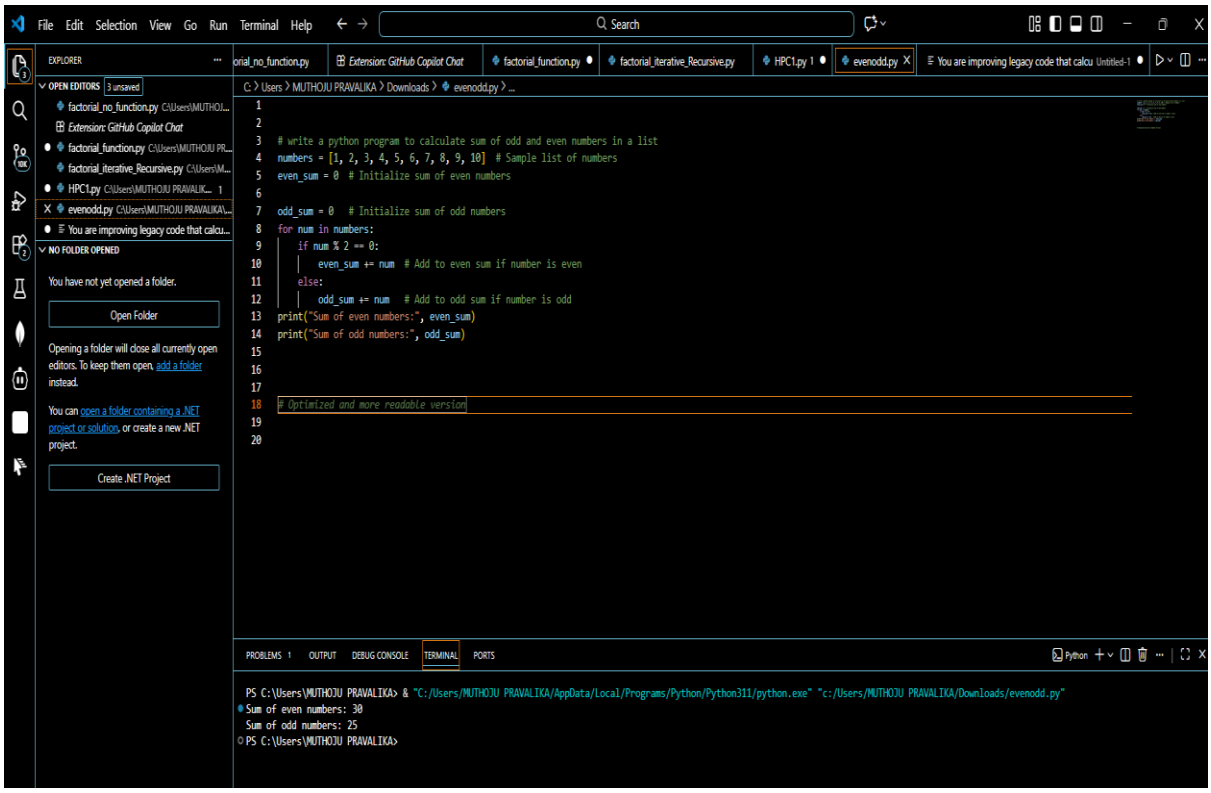
BATCH-34

HALLTICKET.NO : 2303A52347

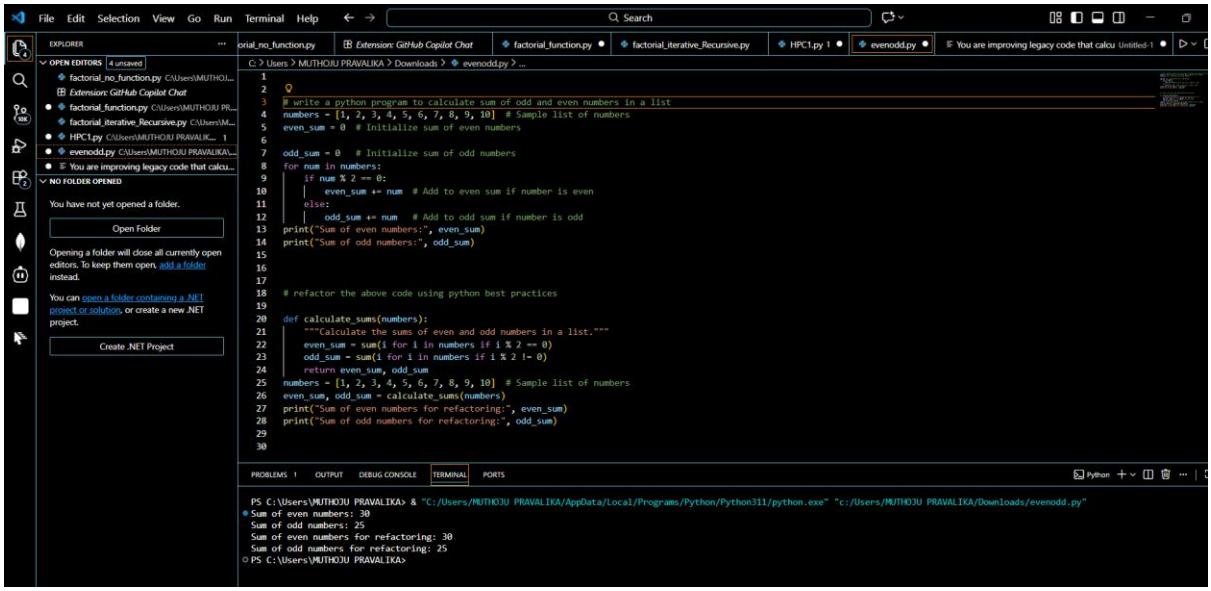
Lab 2: Exploring Additional AI Coding Tools beyond Copilot – Gemini (Colab) and Cursor AI

Task 1: Refactoring Odd/Even Logic (List Version)

- **Scenario:**
- # write a python program to calculate sum of odd and even numbers in a list



refactor the above code using python best practices



Task 2: Area Calculation Explanation

Scenario

You are onboarding a junior developer and need to explain how a function calculates the area of different shapes.

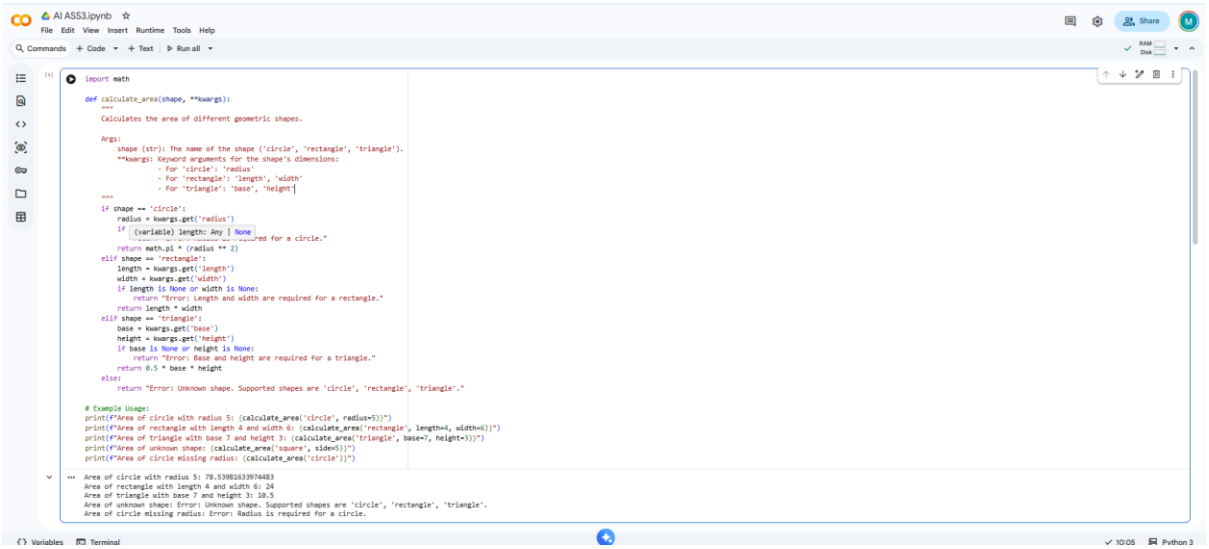
❖ Task:

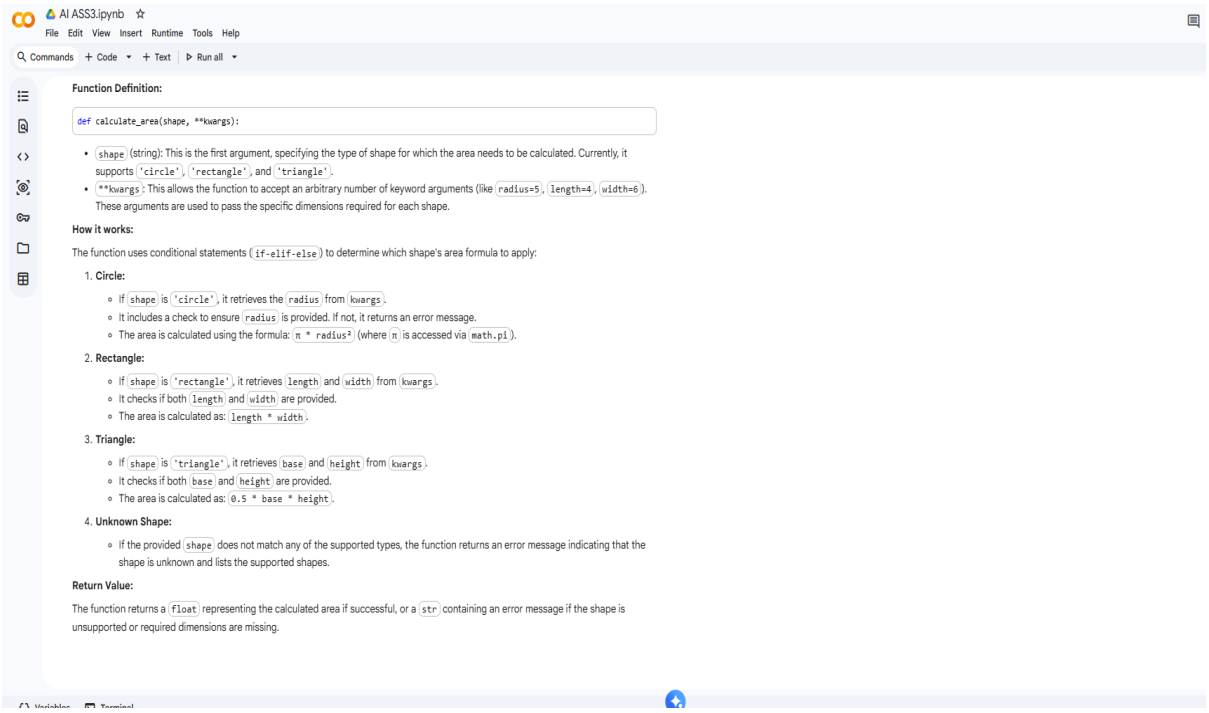
Prompt Used in Google Gemini (Colab)

“Explain a Python function that calculates the area of different shapes like circle, rectangle, and triangle.”

❖ Expected Output:

➢ Code





Explanation

- The function calculate_area() is used to calculate the area of various shapes.
- The parameter shape specifies the type of shape (circle, rectangle, or triangle).
- value1 and value2 represent the required dimensions of the shape.
- For a **circle**, the area is calculated using the formula π × radius².
- For a **rectangle**, the area is calculated as length × breadth.
- For a **triangle**, the area is calculated as ½ × base × height.
- If an unknown shape is passed, the function returns "Invalid shape".

Task 3: Prompt Sensitivity Experiment

Scenario

You are testing how Cursor AI responds to different prompts for the same problem and observing how the generated code changes.

Problem Selected

Calculate the factorial of a number.

Prompt List and Code Variations (Using Cursor AI)

Prompt 1

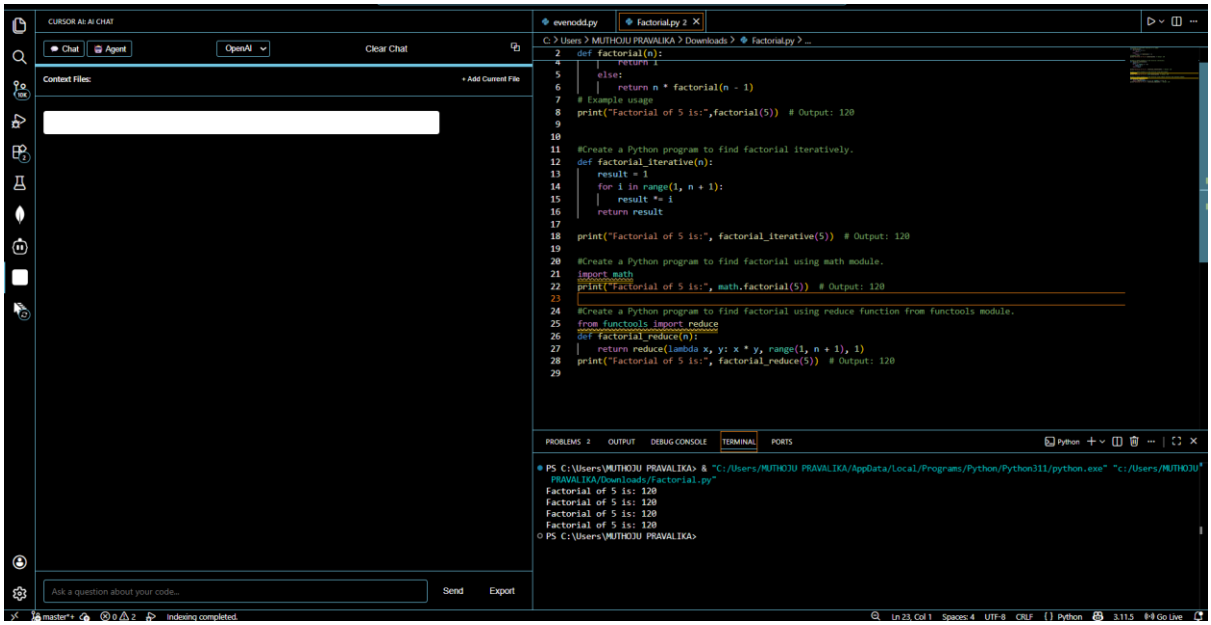
Prompts:

#Write a Python program to calculate factorial of a number."

#Create a Python program to find factorial iteratively.

#Create a Python program to find factorial using math module.

#Create a Python program to find factorial using reduce function from functools module.



Task 4: Tool Comparison Reflection

❖ Scenario:

You must recommend an AI coding tool.

❖ Task:

Based on your work in this topic, compare Gemini, Copilot, and Cursor AI

for usability and code quality.

❖ Expected Output:

Short written reflection

Tool Comparison Reflection

During this experiment with AI-assisted coding, I explored three major AI coding tools: **Gemini, GitHub Copilot, and Cursor AI**. Each tool has strengths and limitations in terms of **usability** and **code quality**.

1. **Gemini**
    - **Usability:** Gemini provides a clear interface for generating code and explanations. It's beginner-friendly, and the AI can respond to natural language prompts directly.
    - **Code Quality:** Gemini often generates readable code with proper comments and stepwise logic. However, sometimes it includes extra steps that may not be necessary, slightly reducing efficiency.
    - **Best Use Case:** Learning and understanding code logic, especially for beginners.
2. **GitHub Copilot**
    - **Usability:** Copilot integrates seamlessly into VS Code. Code suggestions appear inline, making coding faster. Accepting or cycling through suggestions is intuitive.
    - **Code Quality:** Copilot usually produces functional code that follows common programming practices. It is strong for generating boilerplate code and common algorithms. However, complex or niche problems may require manual adjustments.
    - **Best Use Case:** Professional coding and rapid prototyping in known programming languages.
3. **Cursor AI**
    - **Usability:** Cursor AI allows prompt-based code generation and editing directly in VS Code. It supports various ways to give instructions, like comments or selected code blocks, making it flexible.
    - **Code Quality:** Cursor AI generates concise and efficient code, often with multiple variations based on prompts. It is particularly useful for experimenting with different coding approaches and learning alternative methods.
    - **Best Use Case:** Testing code variations, experimenting with different programming approaches, and learning multiple solutions for the same problem.

Recommendation:
Based on this experience, **Cursor AI** is the most versatile tool for experimentation and learning because it provides multiple code variations and responds well to diverse prompts. **GitHub Copilot** is

best for rapid coding and industry-level projects, while **Gemini** excels in educational scenarios and step-by-step code explanation.