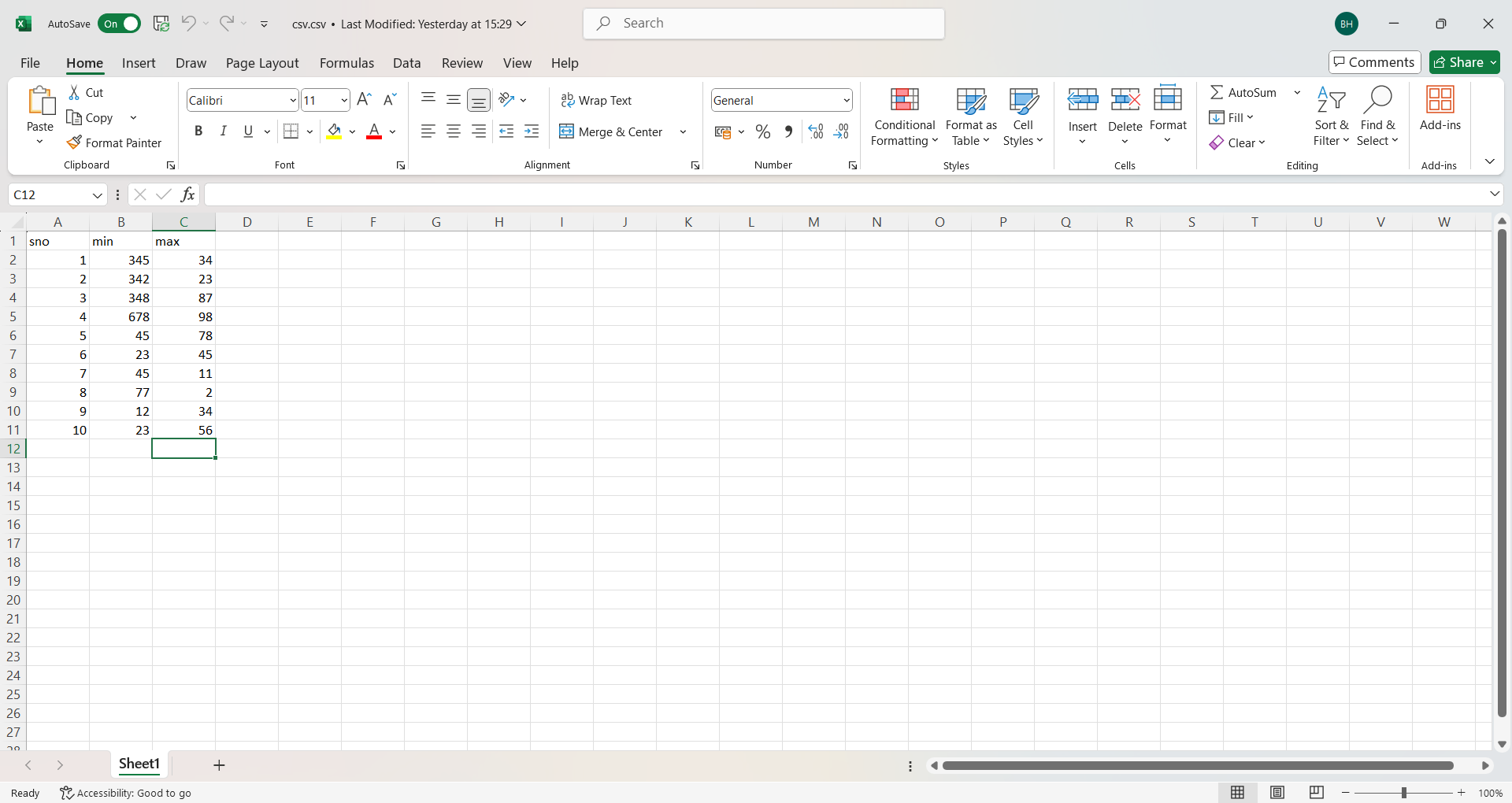
**Lab 2: Exploring Additional AI Coding Tools – Gemini (Collab) and Cursor AI**

**ASSIGNMENT: 2.3**

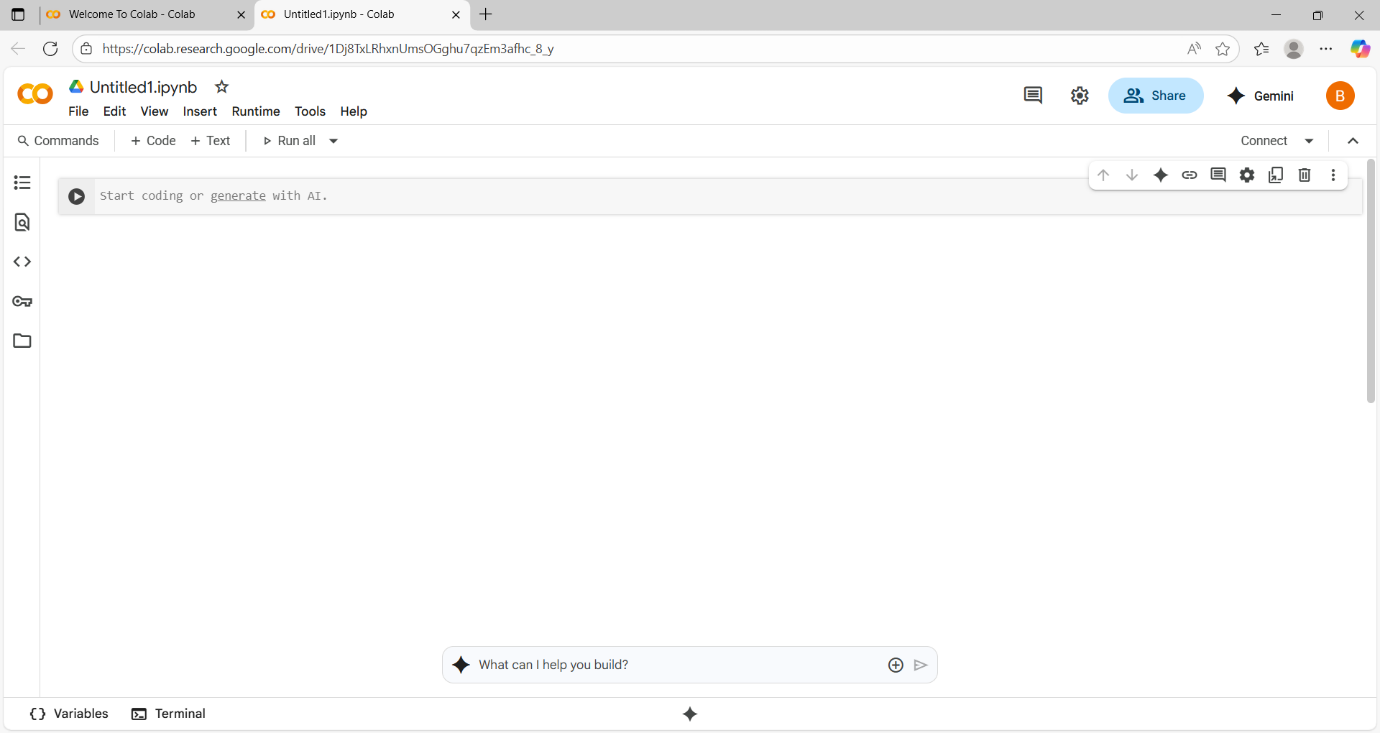
**TASK 1:**

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

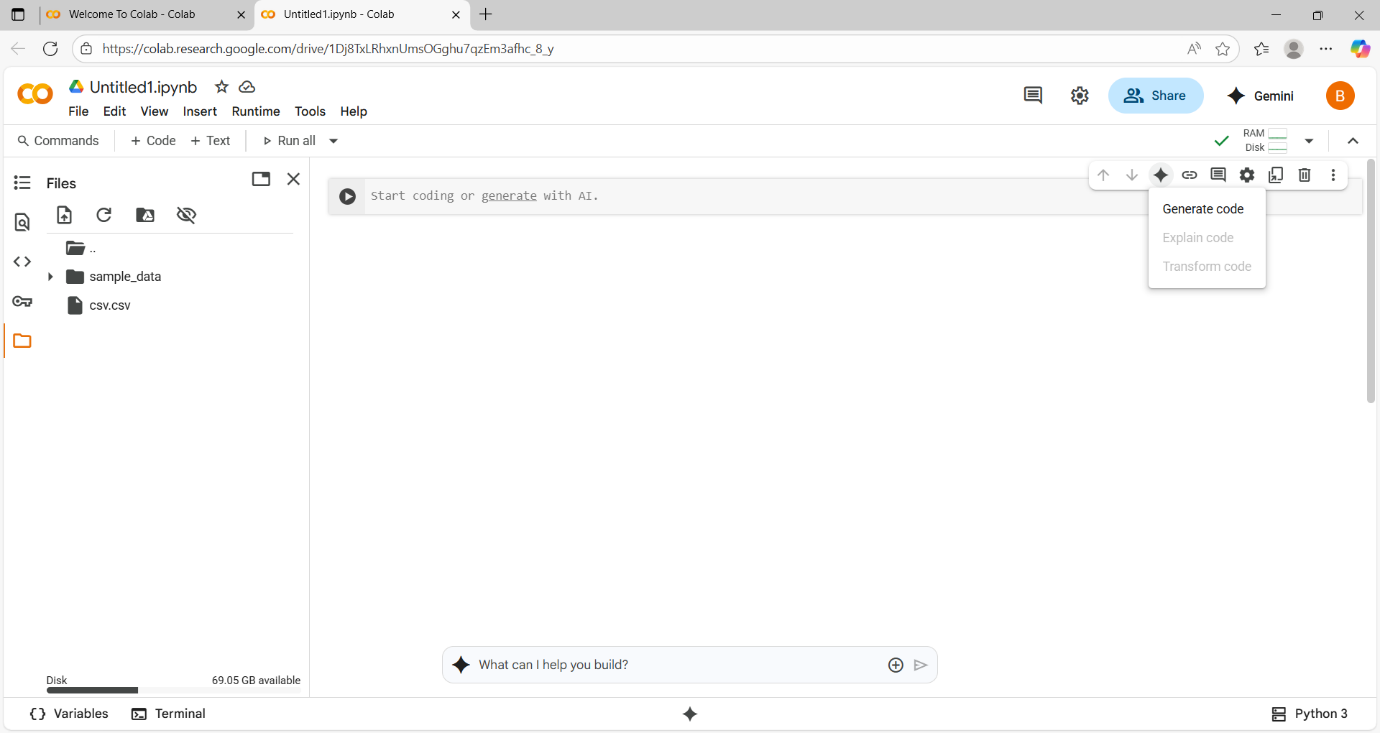
**Step 1:** create a file with .csv extension



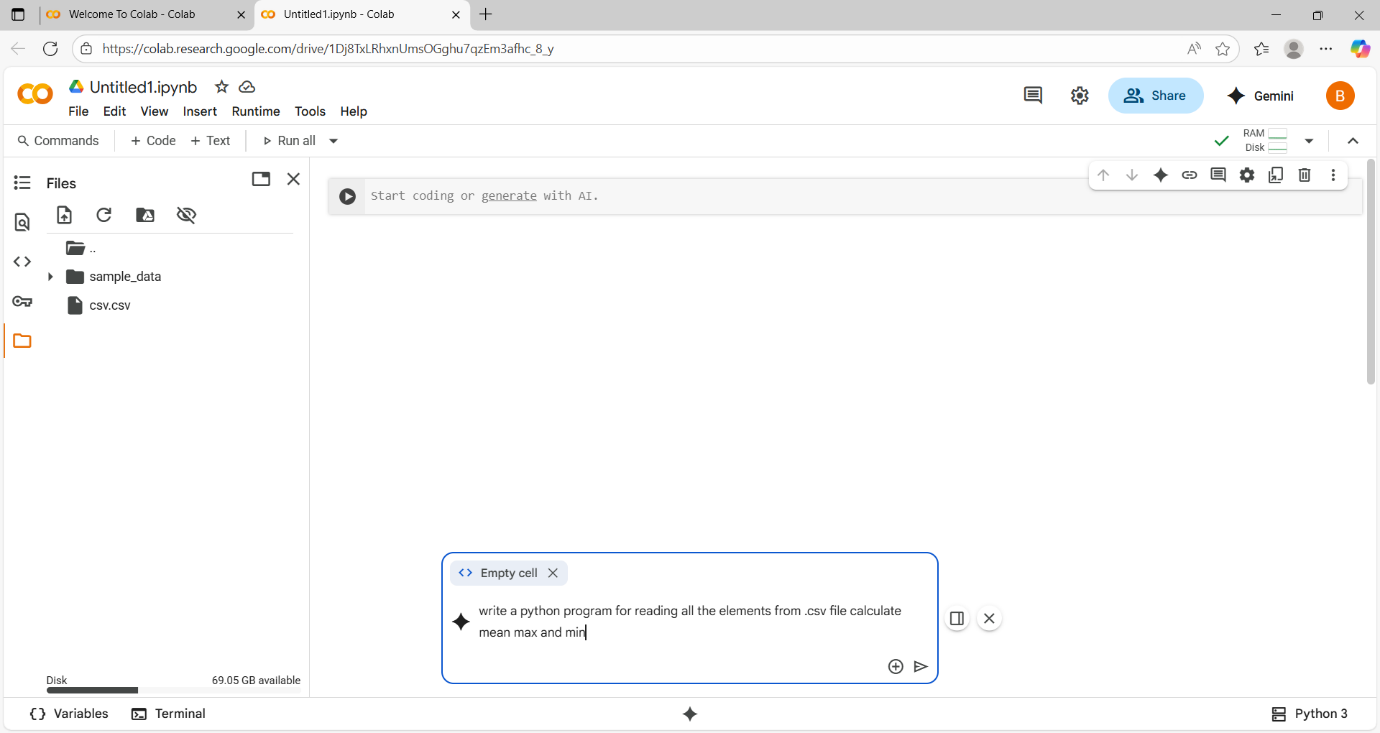
**Step 2:** Open Google Collab and upload the file.



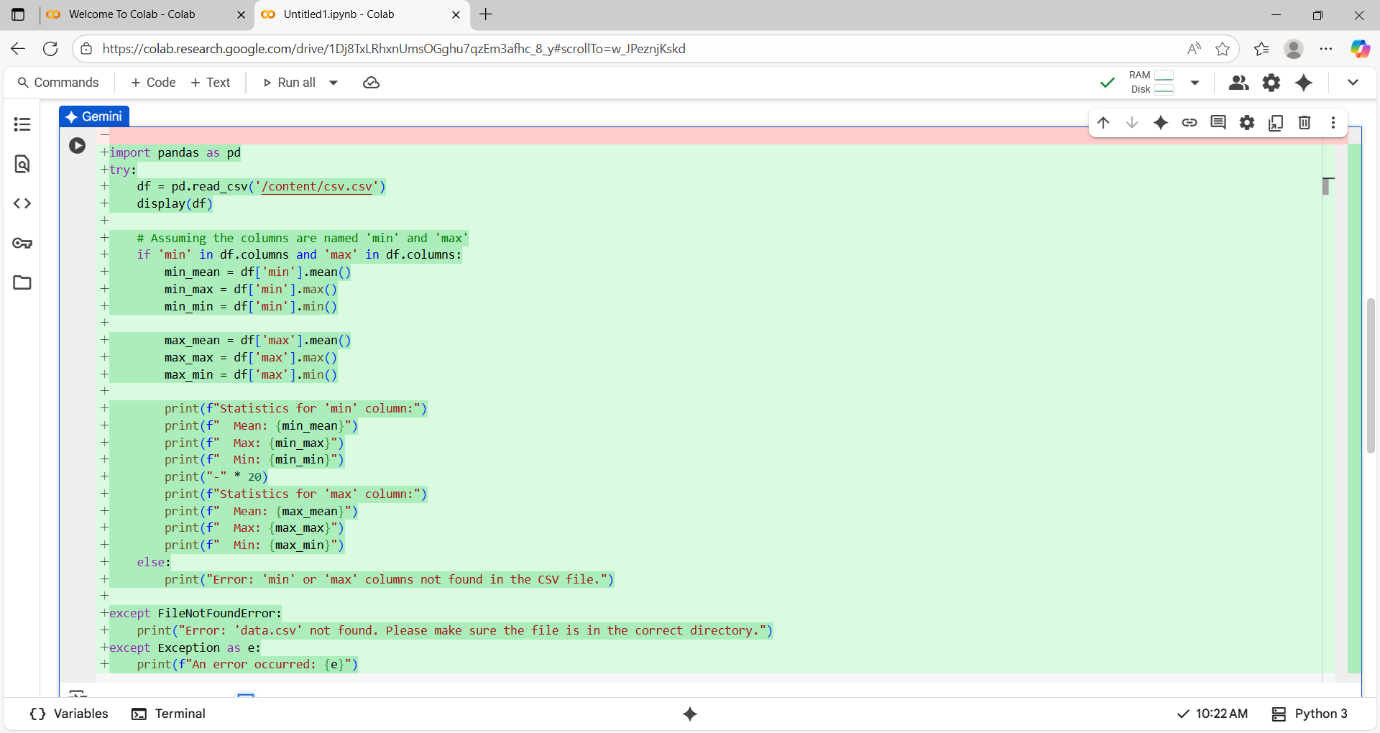
* Upload the file into files section.



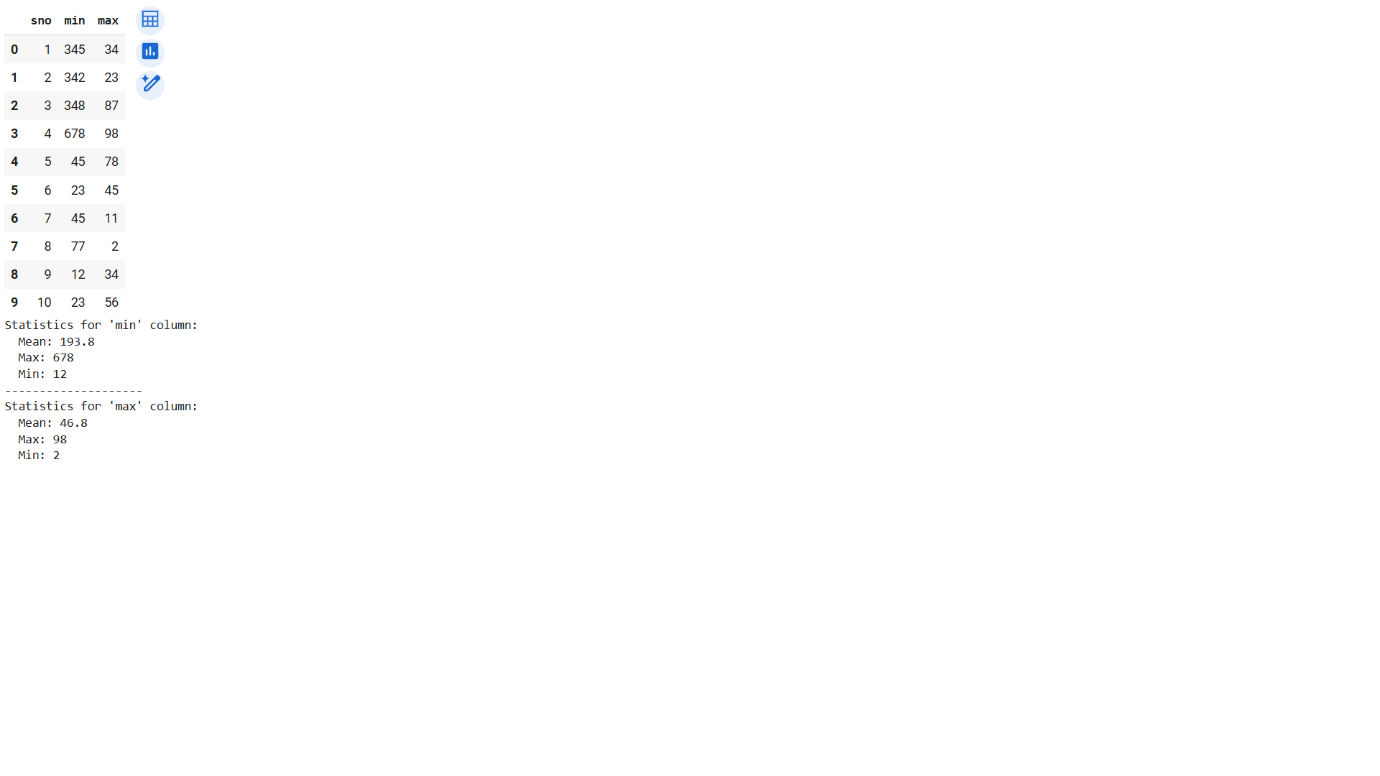
**Step 3:** enter the prompt

**Prompt**: generate a python code for displaying all the elements from .csv file and calculating mean , max, min.

**Code:**



Output:



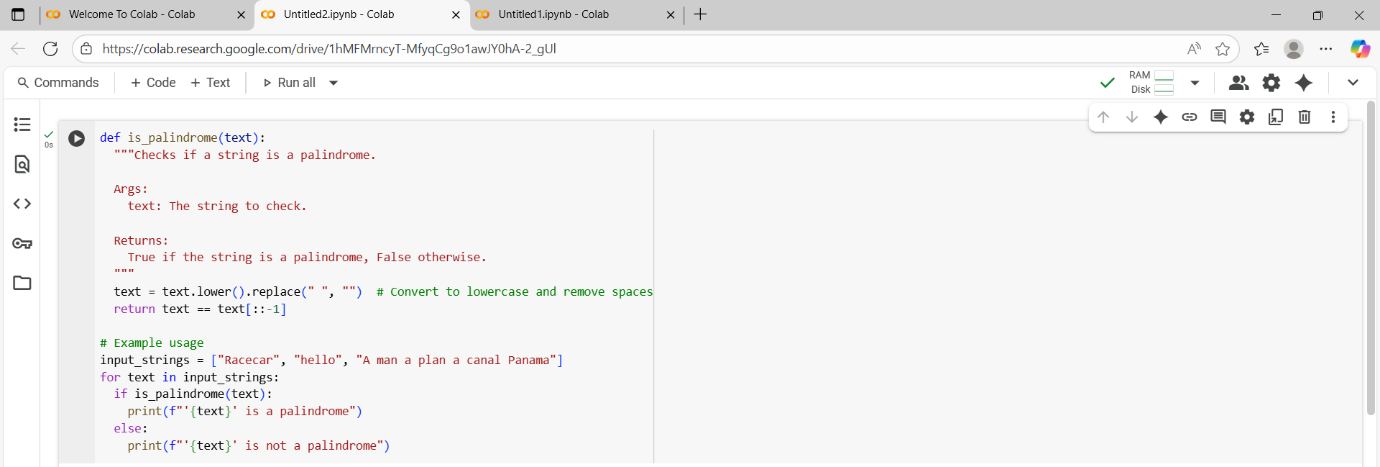
***TASK 2: Compare Gemini and Copilot outputs for a palindrome check function.***

**Prompt:**

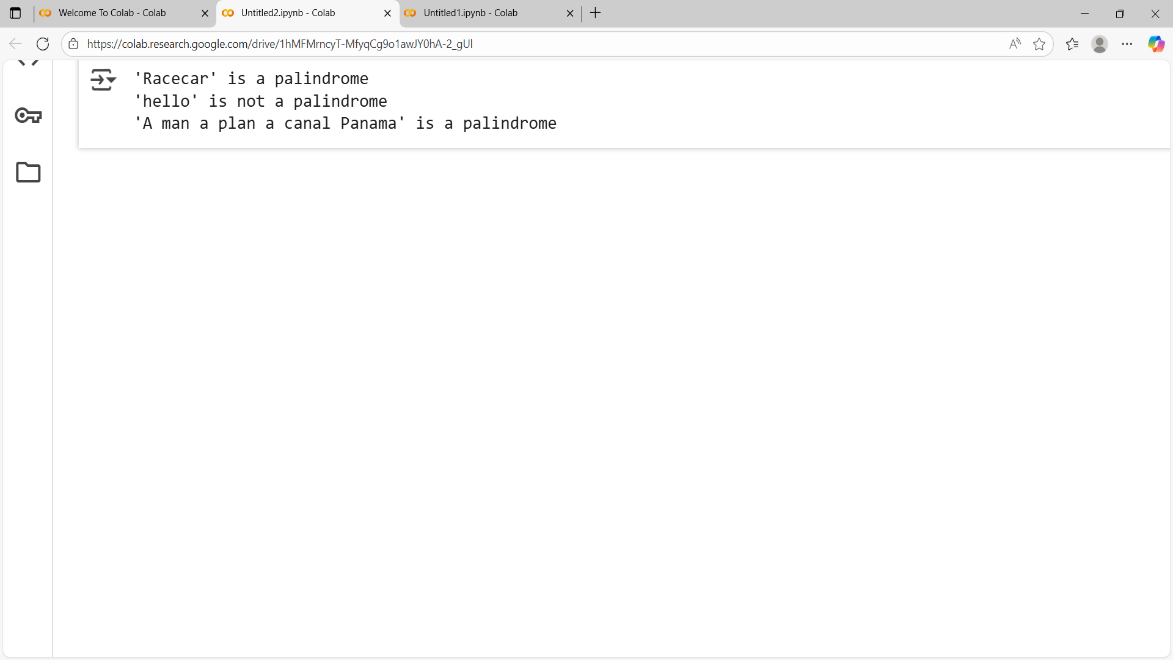
Write a python function for checking palindrome.

GOOGLE COLLAB VERSION:

**Code:**

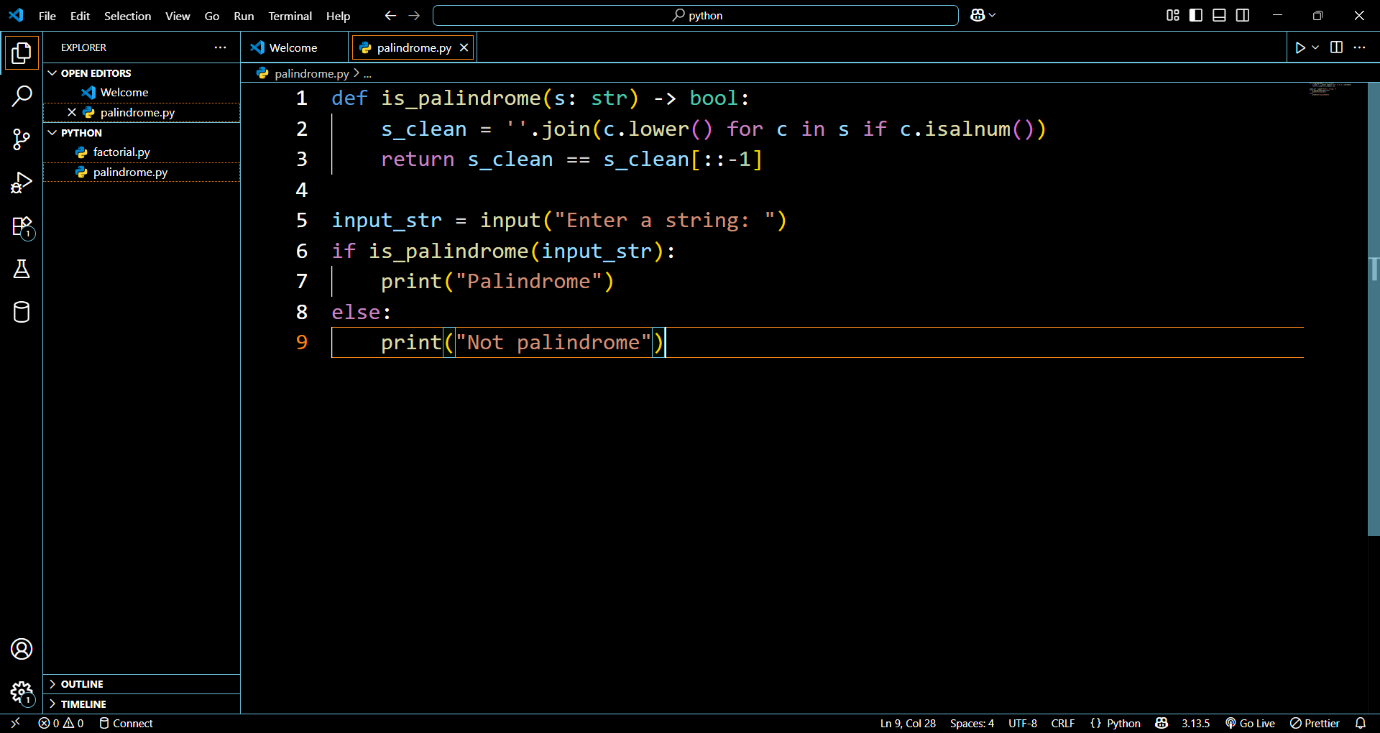


**Output:**

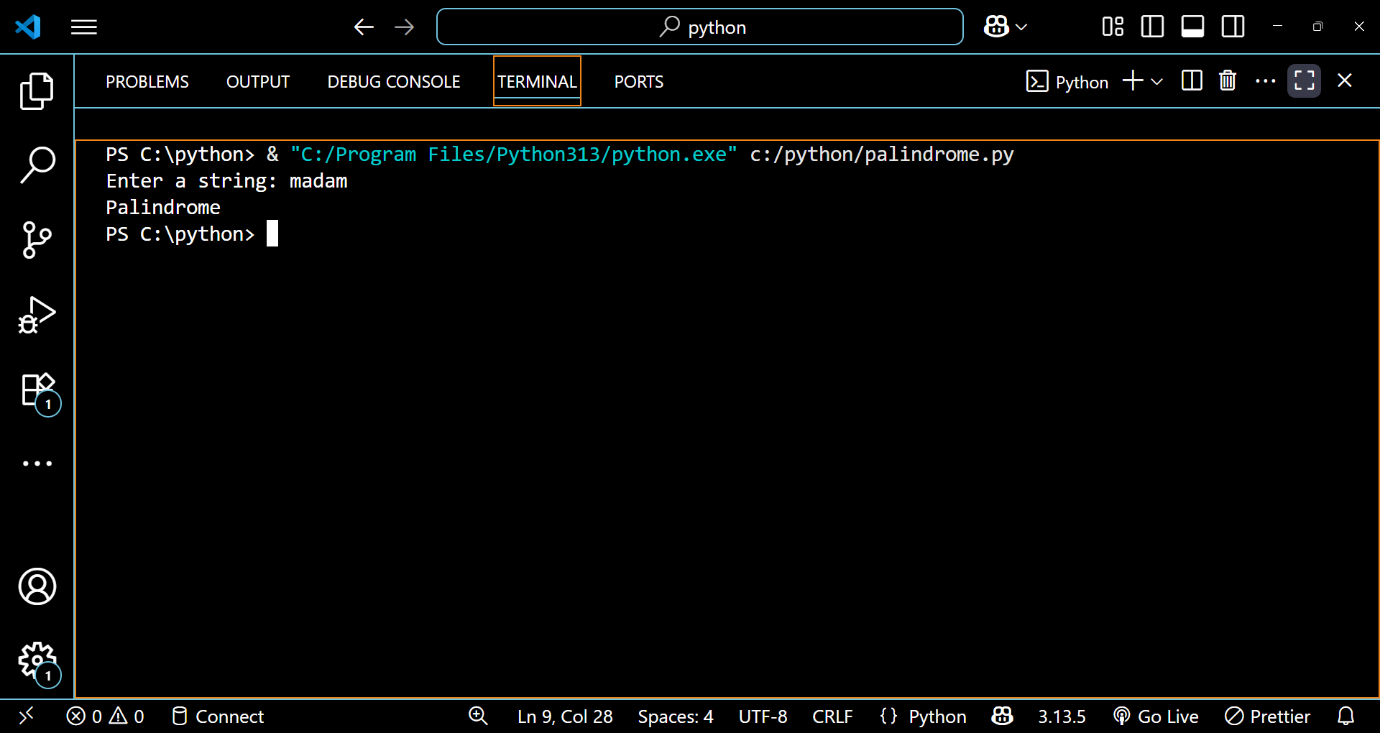


COPILOT – VS CODE VERSION:

**Code:**



**Output:**



**Comparison table of Gemini and Copilot output.**

| **Feature** | **Gemini Output** | **Copilot Output** |
| --- | --- | --- |
| Function Name | is\_palindrome | is\_palindrome |
| Input Cleaning | Lowercase + removes spaces | Lowercase + removes non-alphanumeric chars |
| Case Sensitivity | Not sensitive | Not sensitive |
| Spaces Handling | Yes | Yes |
| Special Characters | Not handled | Handles special characters |
| Code Length | More lines, includes docstring and examples | Shorter, direct function usage |
| Readability | Beginner-friendly with docstring | Concise and practical |
| Robustness | Handles spaces and case, but not special symbols | Handles spaces, case, and special symbols |

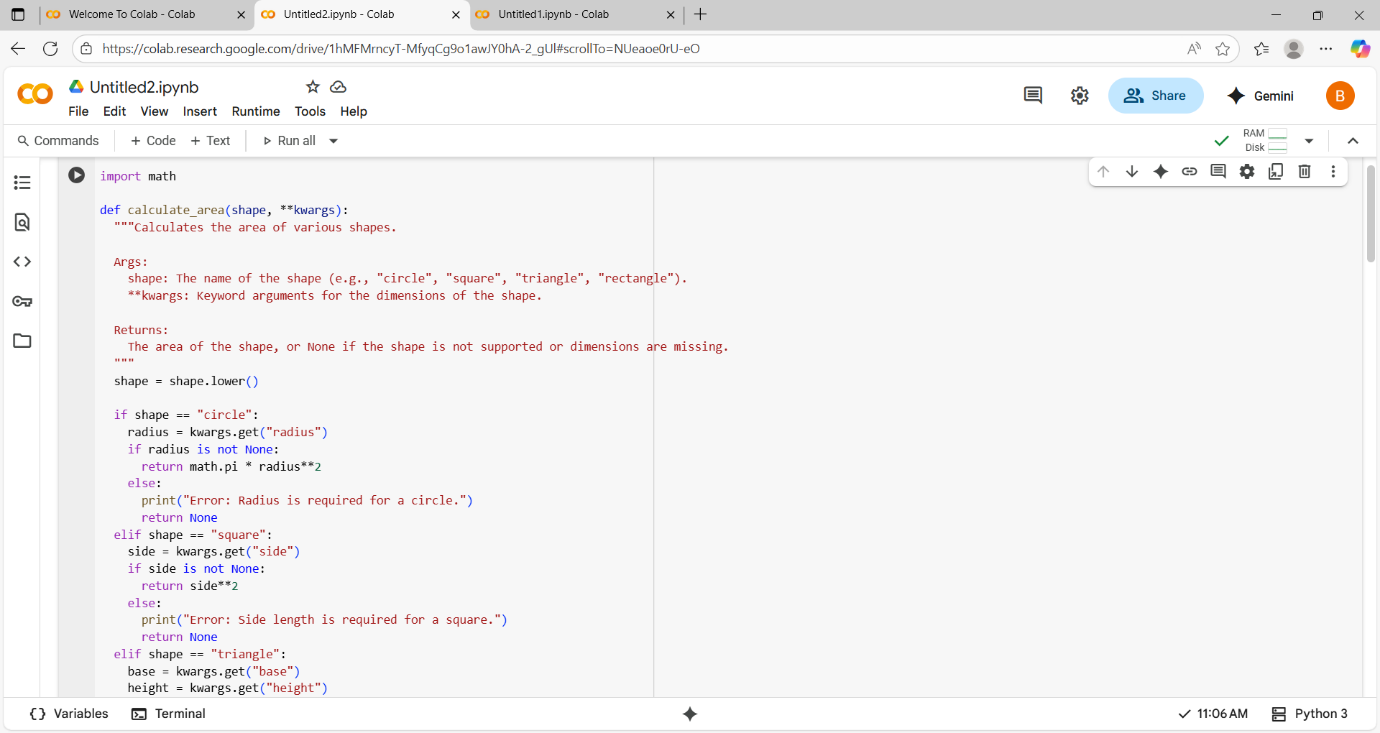
**OBSERVATION:**

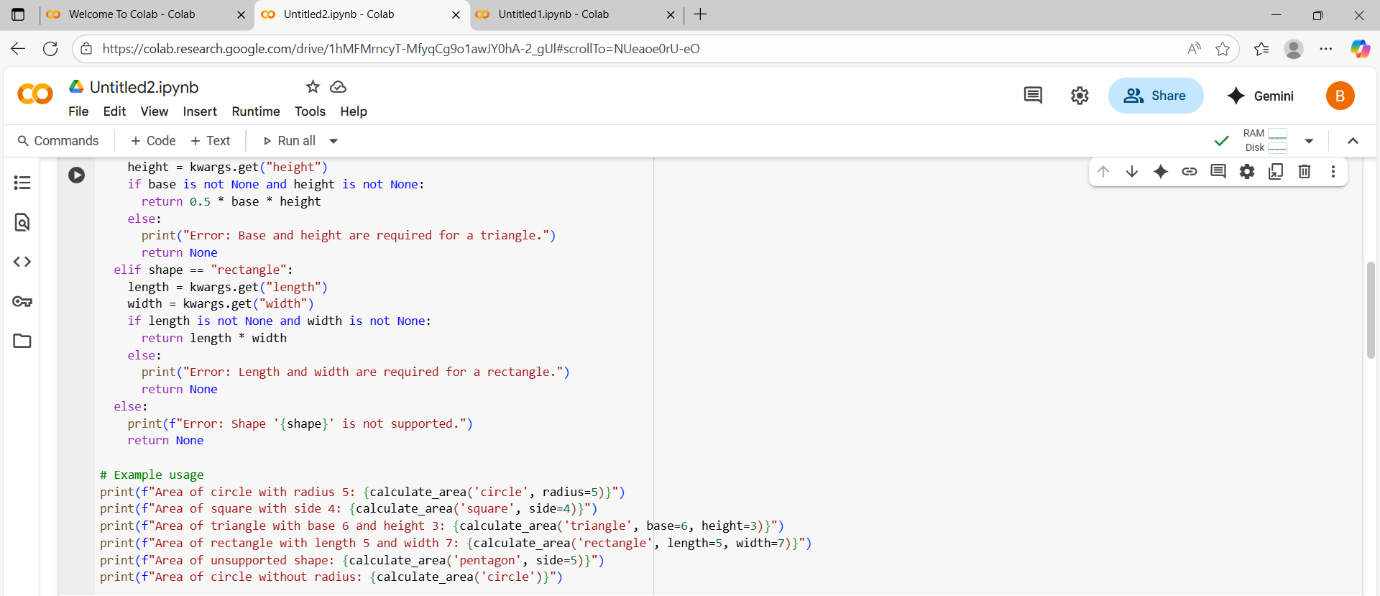
Gemini’s version is way more suitable for beginners whereas Copilot version is more input-safe as it allows users to enter their required input to check. Both are efficient but the copilot code is shorter, but Gemini code is for learners because it has more lines and more examples.

***TASK 3: Ask Gemini to explain a Python function (to calculate area of various shapes) line by line***

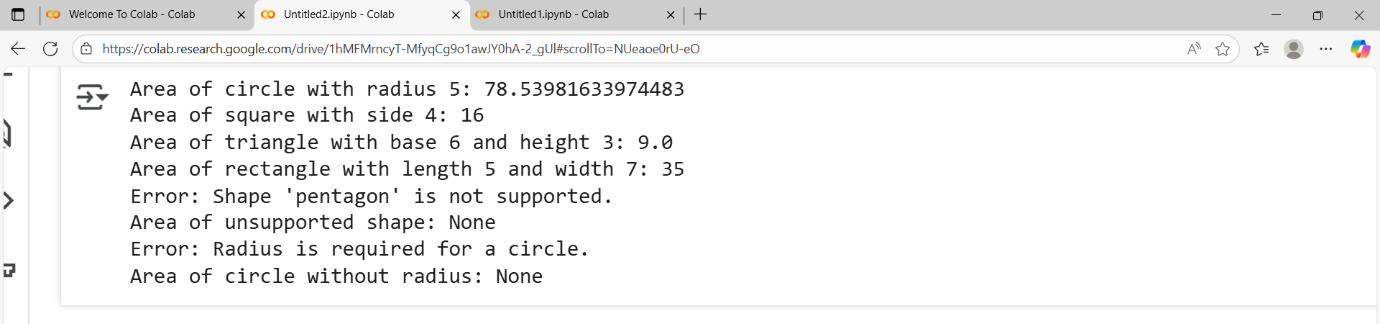
**Prompt:** Explain a python function to calculate are of various shapes line by line.

**CODE:**





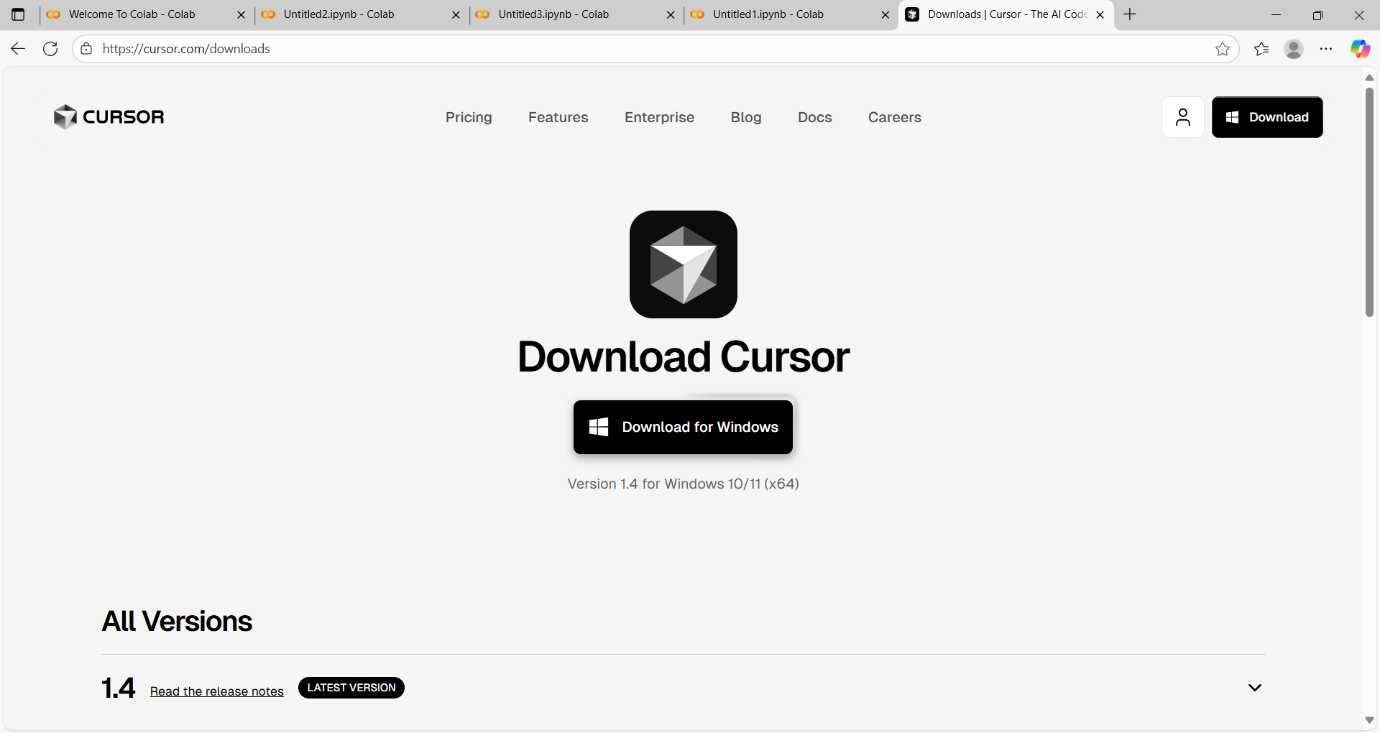
**OUTPUT:**

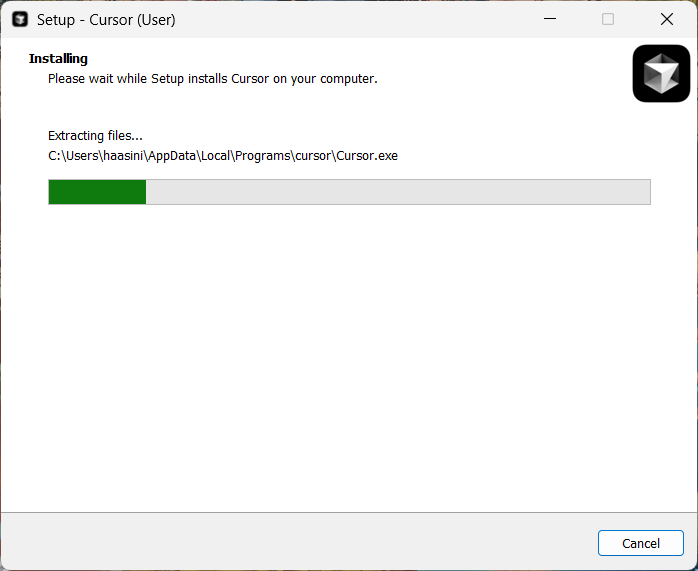


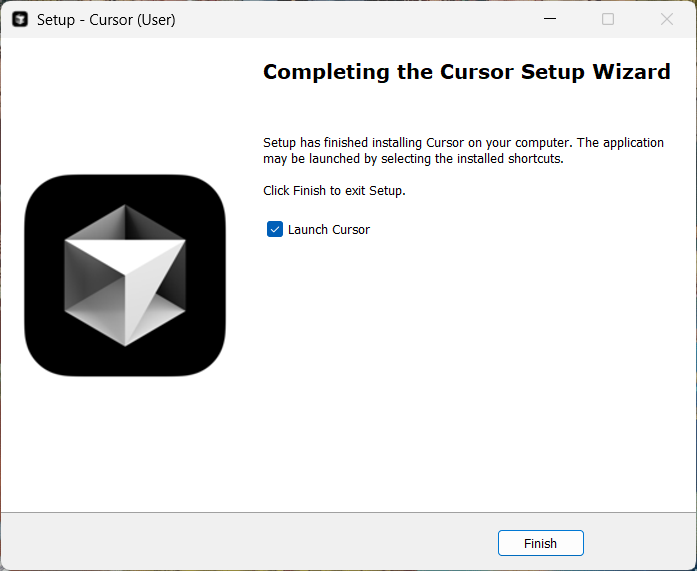
**TASK 4:**

**INSTALLATION OF CURSOR AI**

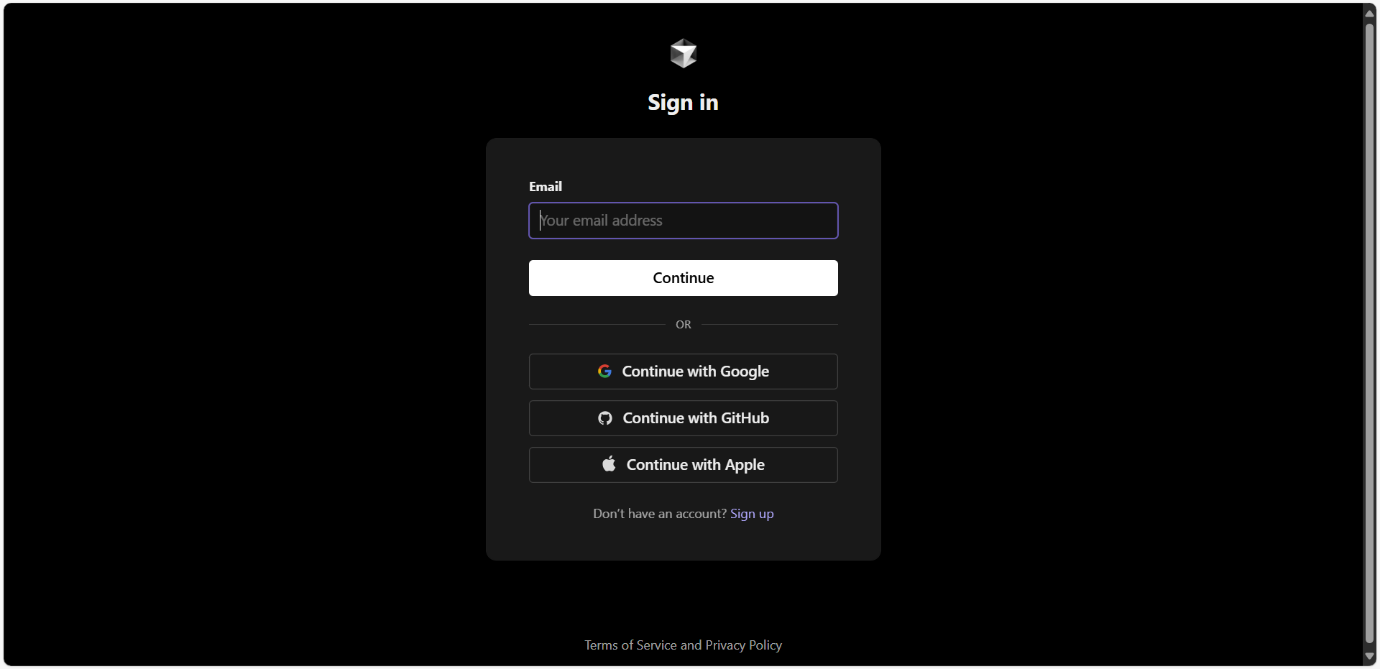
**Here, I am installing Cursor AI for windows**



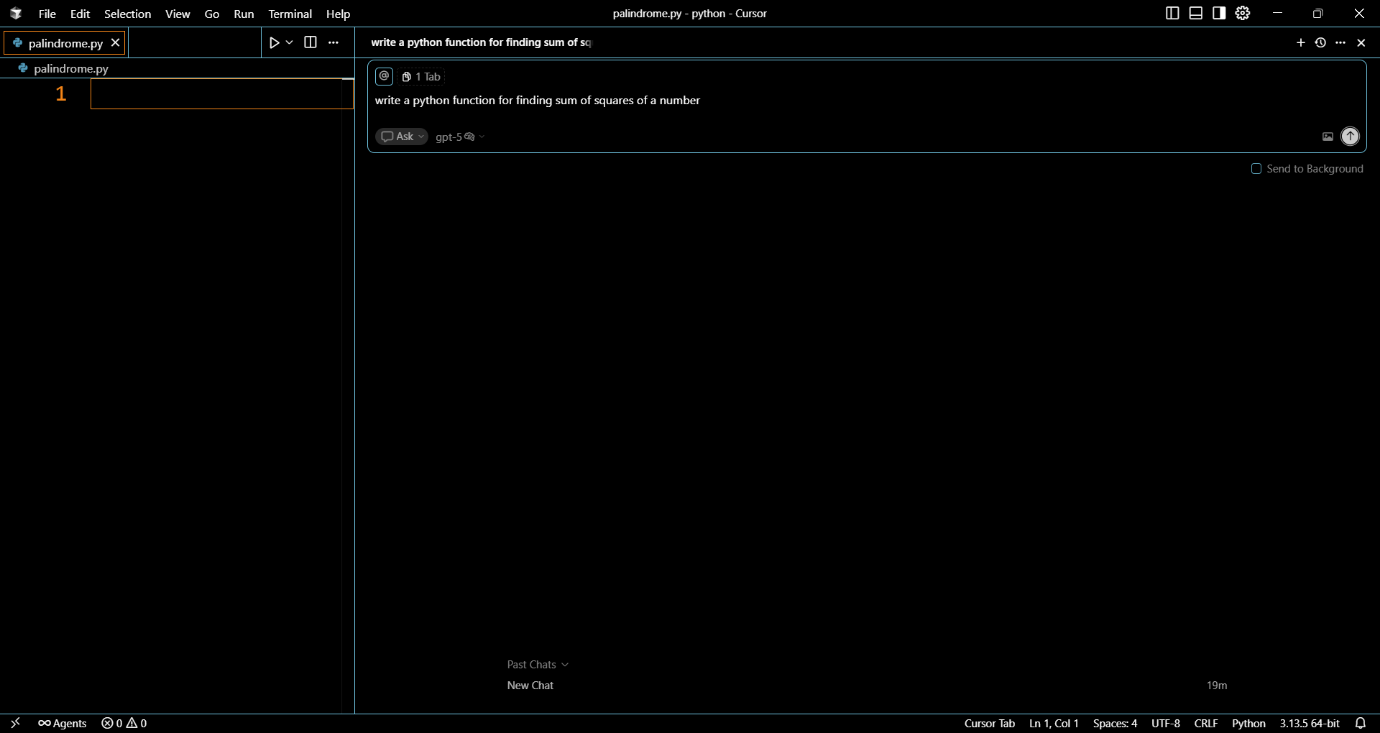




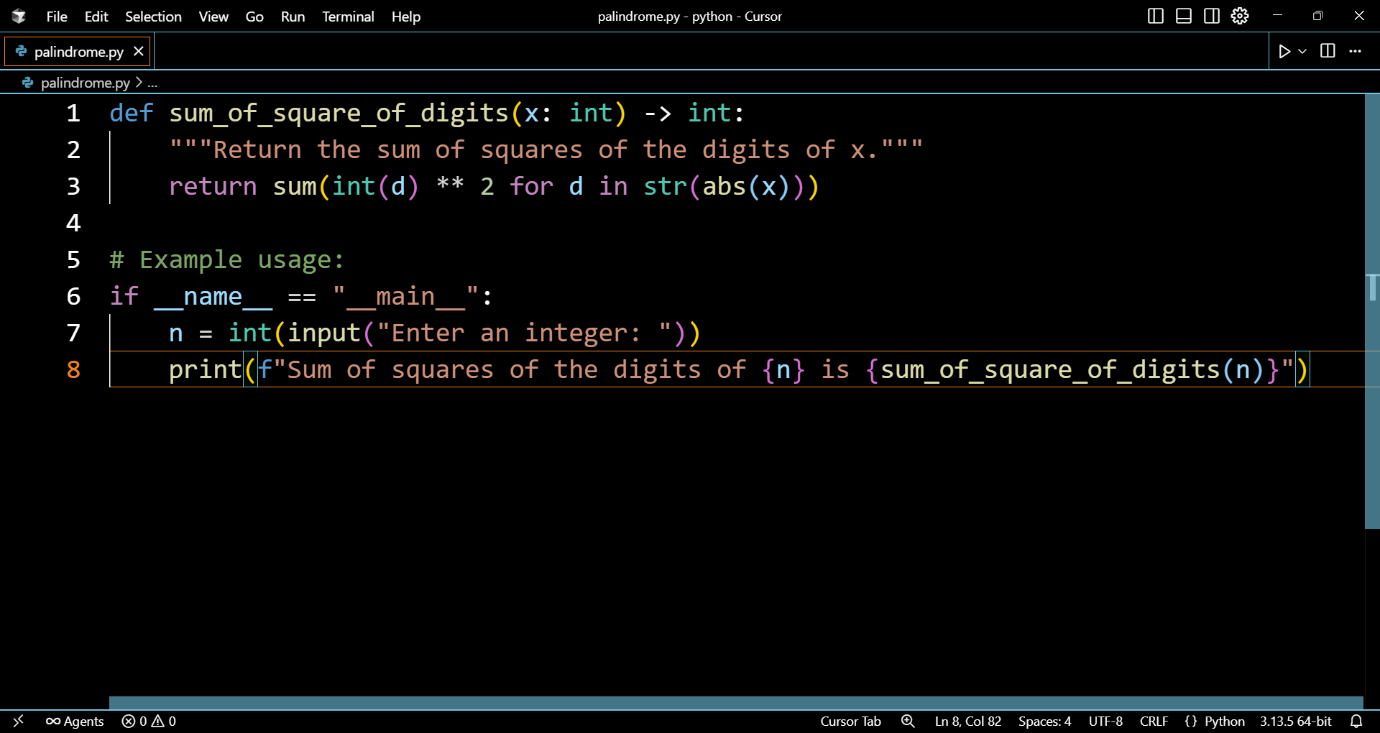
🡪 Enter the mail and login into Cursor Ai



PROMPT: Write a python function to calculate sum of squares of a number.



## CODE:

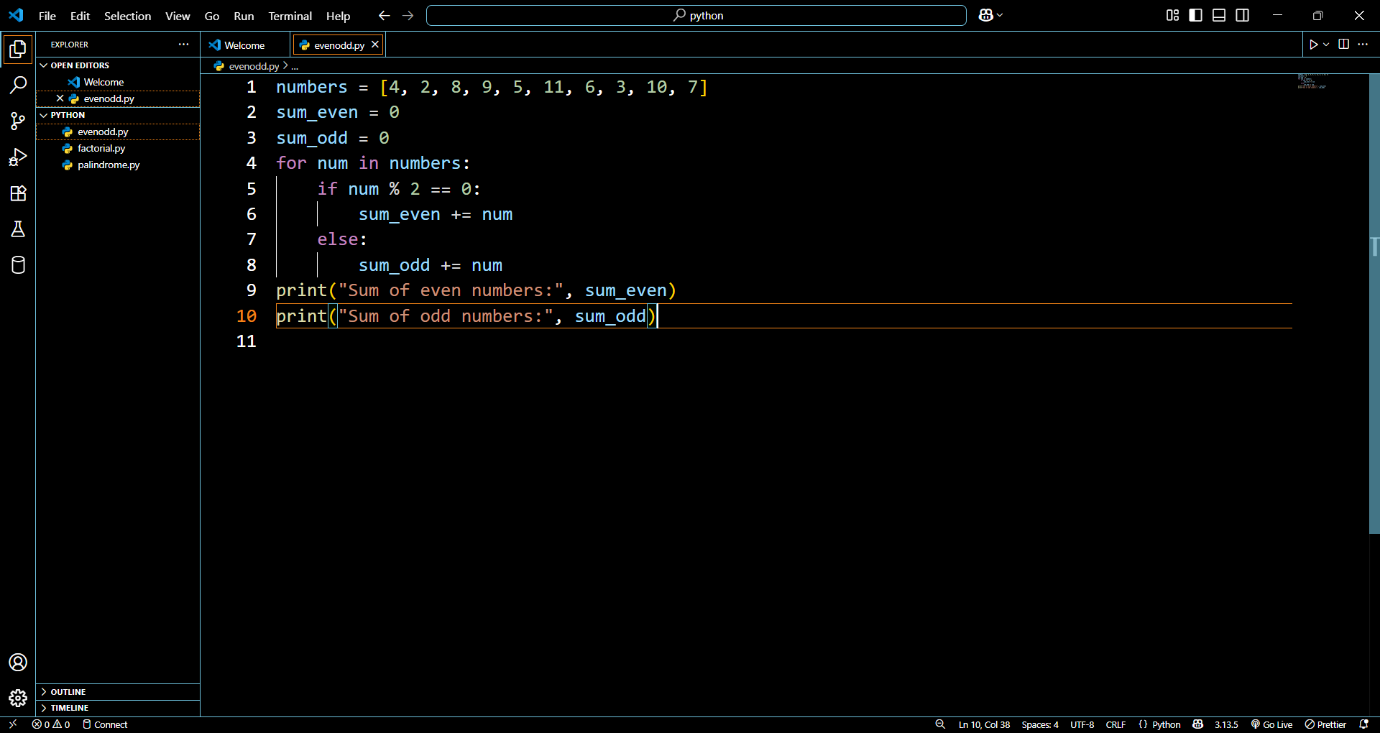


OUTPUT:

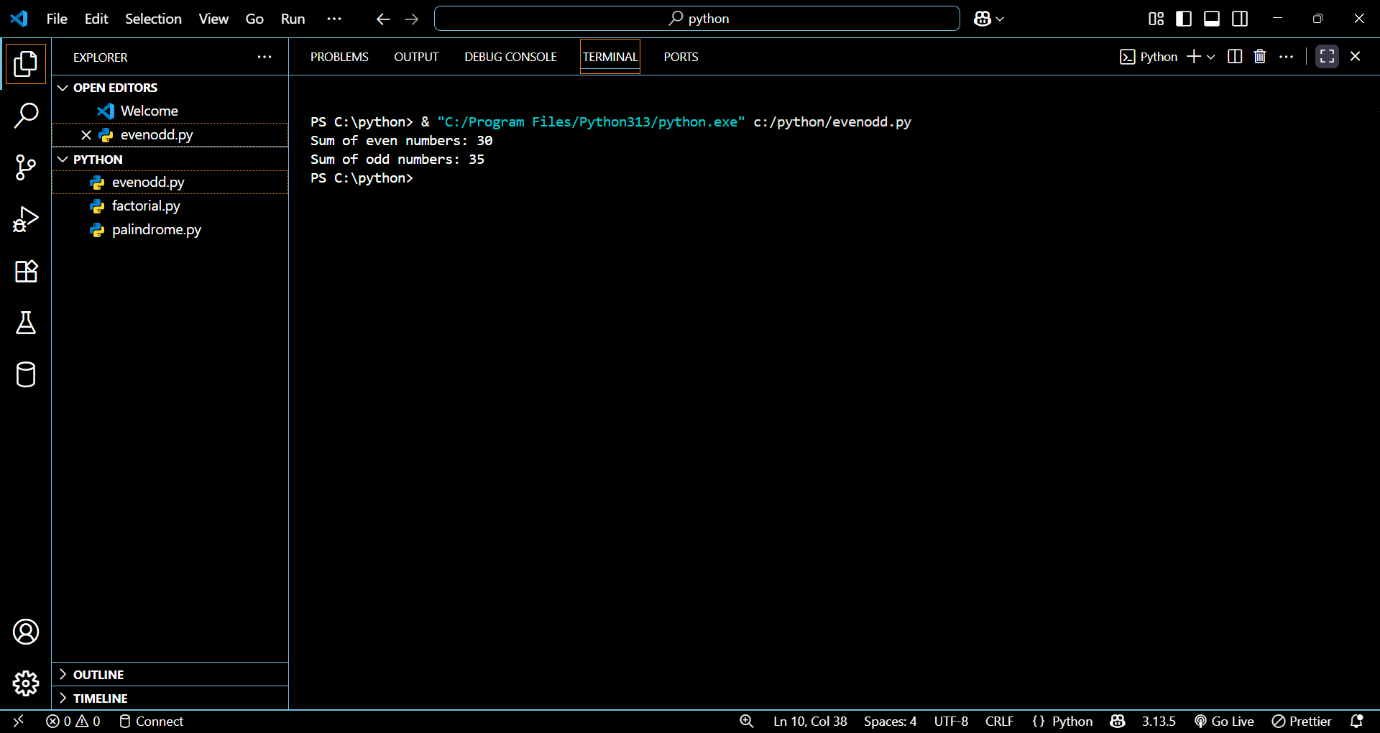


**TASK 5: to write code to calculate sum of add number and even numbers in the list**

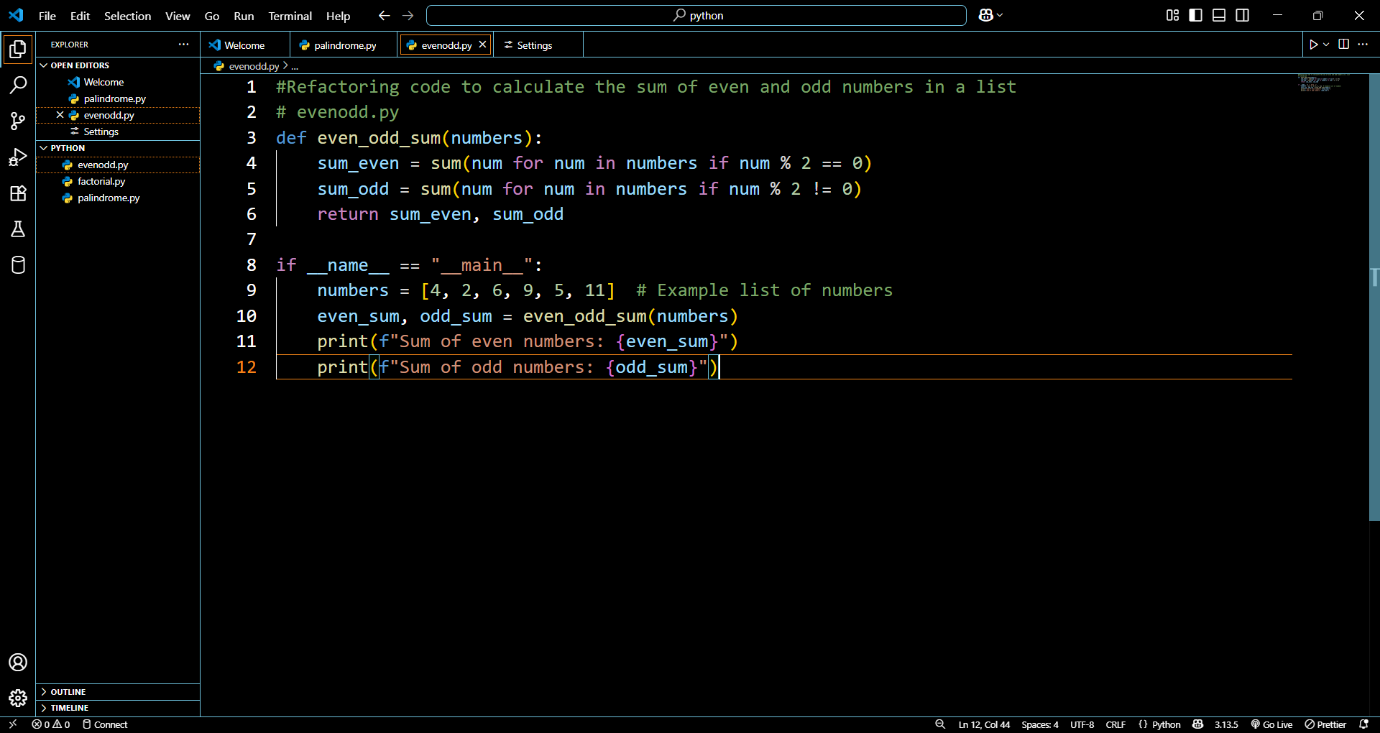
**And refactoring the code.**

ORIGINAL CODE:  


OUTPUT:



REFACTORED CODE:



OUTPUT:



COMPARISION TABLE

| **Feature** | **Original Code** | **Refactored Code** |
| --- | --- | --- |
| Structure | Simple loop | Function + List Comprehension |
| Length | 10 lines | 8 lines |
| Readability | Easier for beginners | Compact and concise |
| Reusability | No | Yes (function can be reused) |
| Efficiency | O(n) | O(n) (same, but cleaner syntax) |

**OBSERVATIONS:**

* The **original version** is more beginner-friendly but longer.
* The **refactored version** is more concise, reusable, and Pythonic.
* Both give the same output, but AI’s approach is better for professional coding standards.