AI ASSISTED CODING
ASSIGNMENT – 2.4

Name: V. Vyshnavi

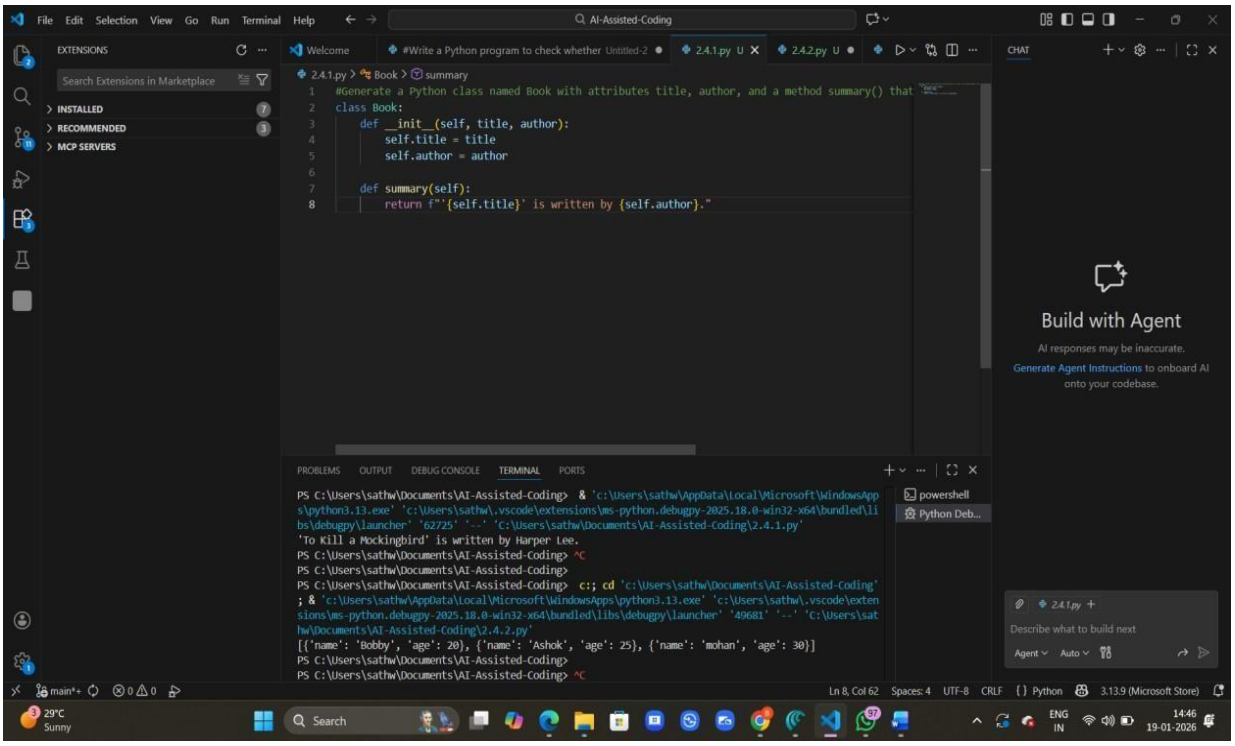
Hallticket No. : 2303A51968

Batch-14

Task 1: Use Cursor AI to generate a Python class Book with attributes title, author, and a summary () method.

Prompt : "Generate a Python class named Book with attributes title, author, and a method summary() that returns a formatted string with the title and author."

Code and output :



Justication:

Object-Oriented Programming helps model real-world entities like books using classes and objects

Encapsulation allows data (title, author) and behavior (summary) to be bundled together logically.

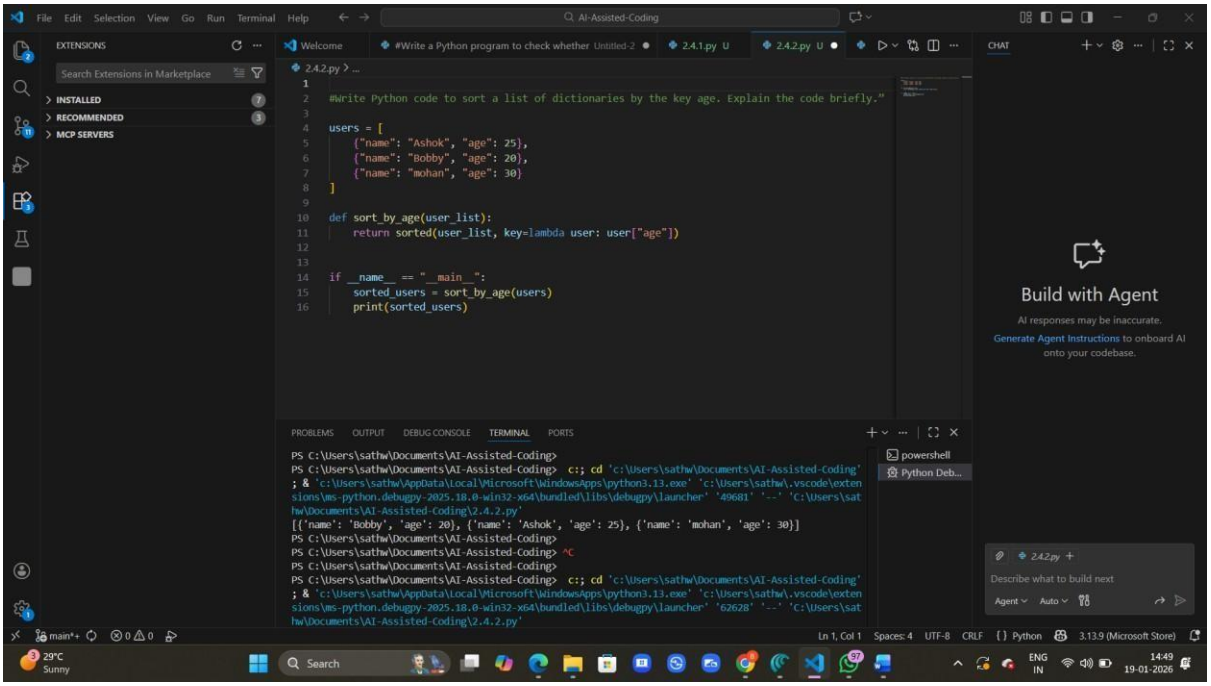
This approach improves code organization, scalability, and ease of maintenance in software systems.

Task 2: Use Gemini and Cursor AI to generate code that sorts a list of dictionaries

by a key.

Prompt: Write Python code to sort a list of dictionaries by the key age. Explain the code briefly.

Code and Output :



Justification:

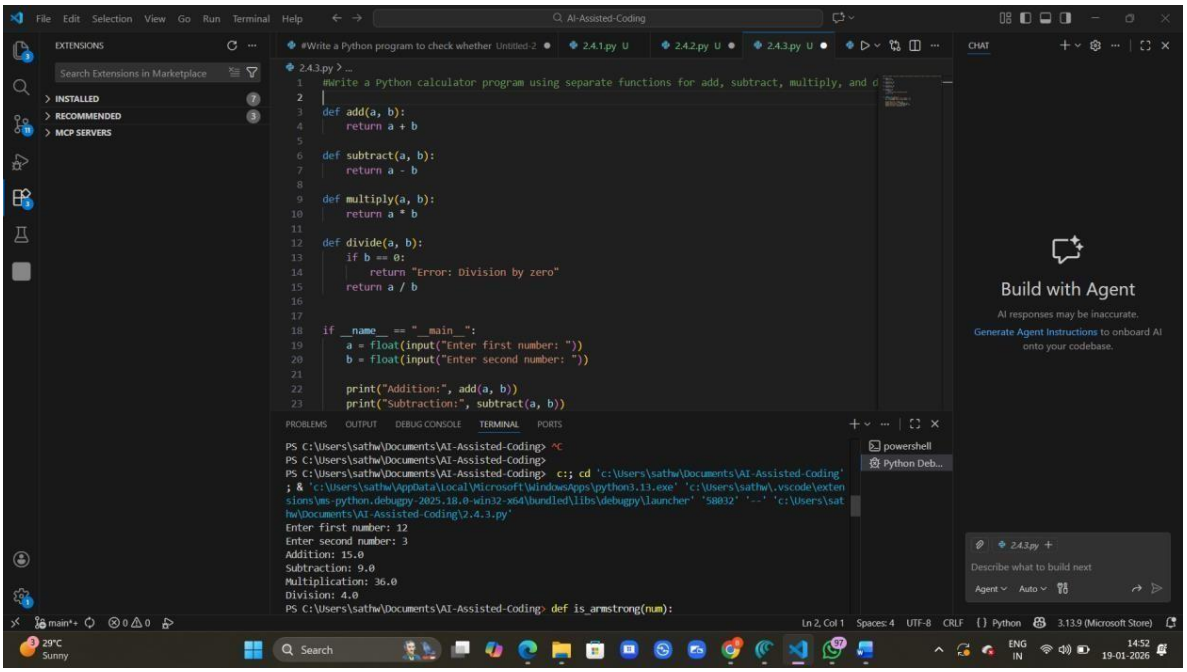
Gemini's code is more concise and easy to understand for quick tasks.

Cursor's approach is slightly more structured and reusable due to function encapsulation.

Task 3: Ask Gemini to generate a calculator using functions and explain how it works.

Prompt: Write a Python calculator program using separate functions for add, subtract, multiply, and divide. Then explain how the program works step by step.

Code and Output:



Justification:

Each arithmetic operation is written as a separate function, improving modularity.

Functions take two inputs, perform the operation, and return the result.

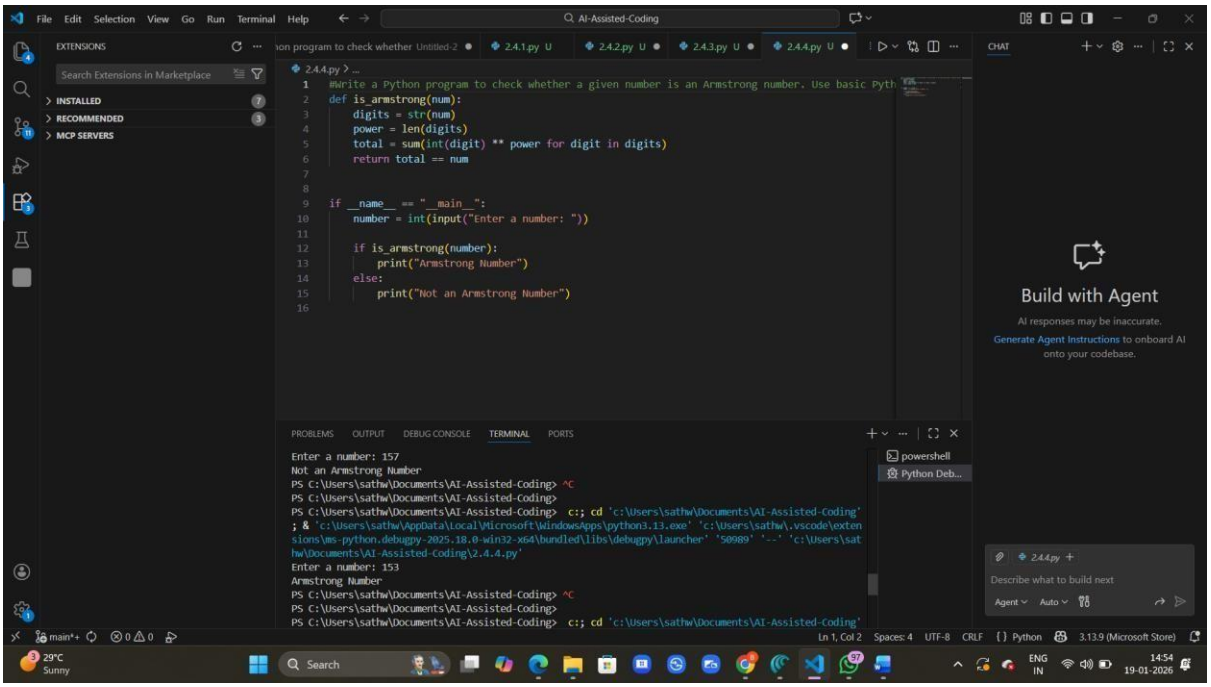
Division includes a safety check to prevent division by zero errors.

Task 4: Generate an Armstrong number program using Gemini, then improve it using

Cursor AI.

Prompt: Write a Python program to check whether a given number is an Armstrong number. Use basic Python constructs and explain briefly.

Code and Input:



Justification:

The optimized version avoids string conversion, reducing unnecessary overhead.

It uses arithmetic operations, which are more efficient and closer to low-level computation.

The logic is clearer for understanding numeric processing and improves overall performance.