

AI ASSISTED CODING

Name : K.Abhinav

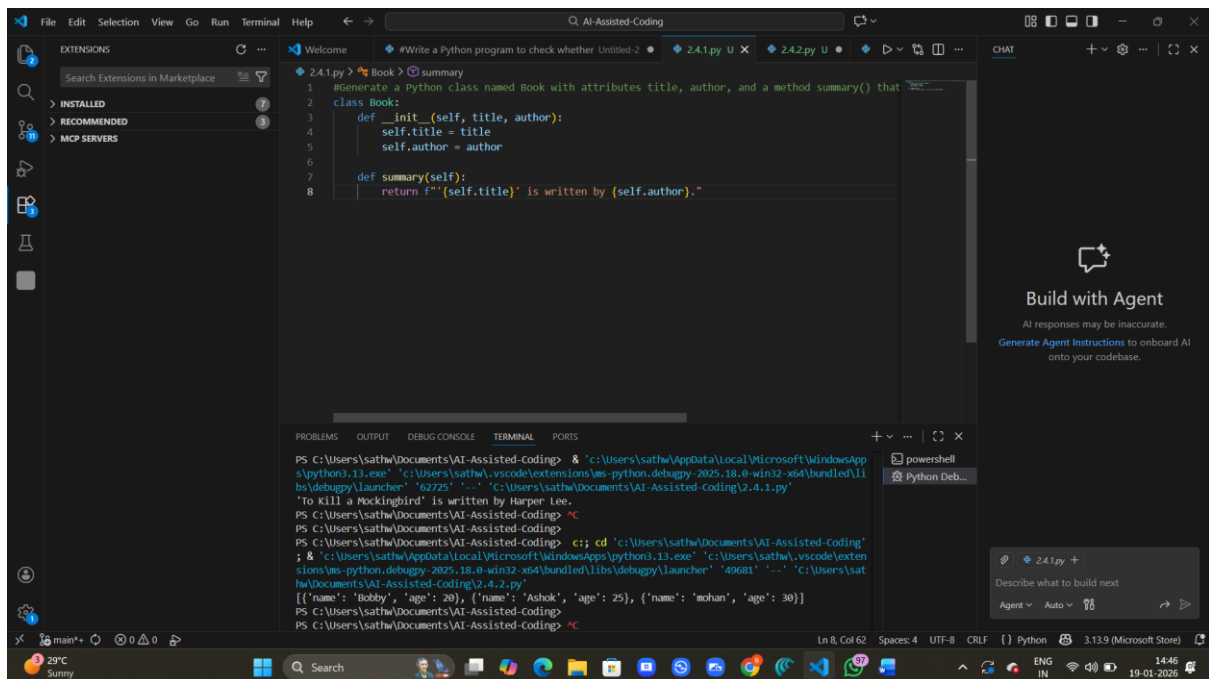
HT No: 2303A51856

ASSIGNMENT – 1

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 :



The screenshot shows the Cursor AI IDE interface. The main editor displays a Python class named `Book` with attributes `title` and `author`, and a method `summary()`. The code is as follows:

```
1 #Generate a Python class named Book with attributes title, author, and a method summary() that
2 class Book:
3     def __init__(self, title, author):
4         self.title = title
5         self.author = author
6
7     def summary(self):
8         return f'{self.title} is written by {self.author}.'
```

The terminal at the bottom shows the execution of the code, which outputs the following list of dictionaries:

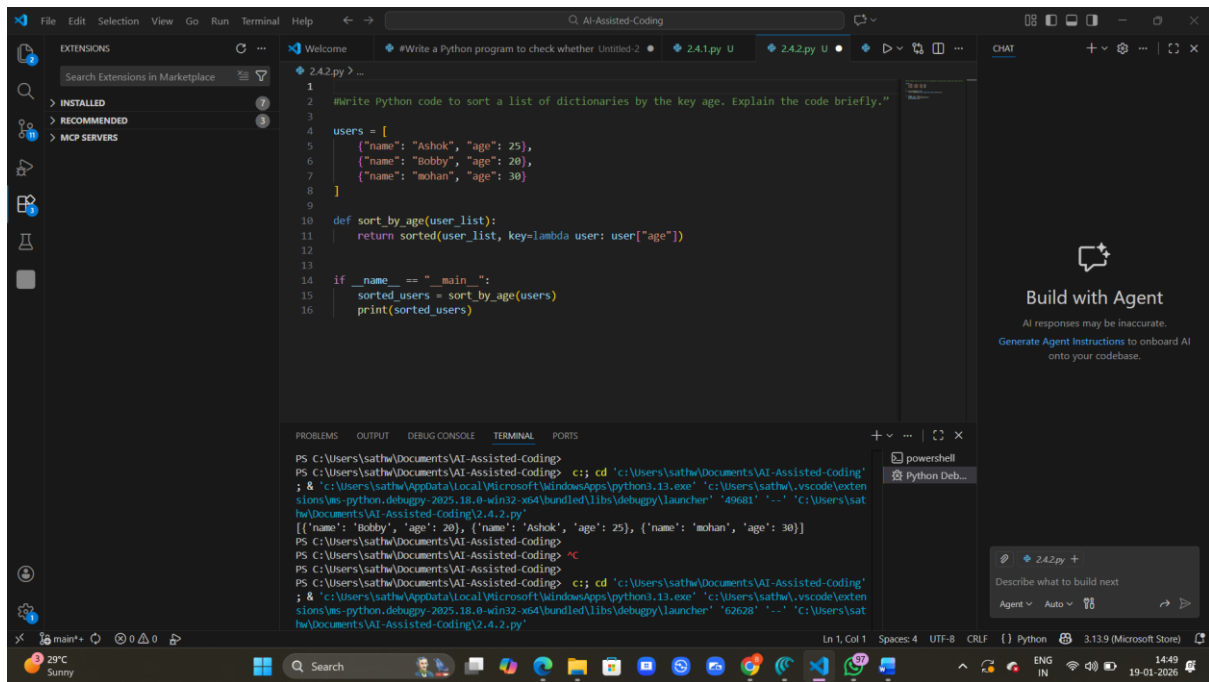
```
[[{'name': 'Bobby', 'age': 20}, {'name': 'Ashok', 'age': 25}, {'name': 'mohan', 'age': 30}]]
```

The right sidebar shows the 'CHAT' panel with the 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." and the response: "To kill a Mockingbird is written by Harper Lee."

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.

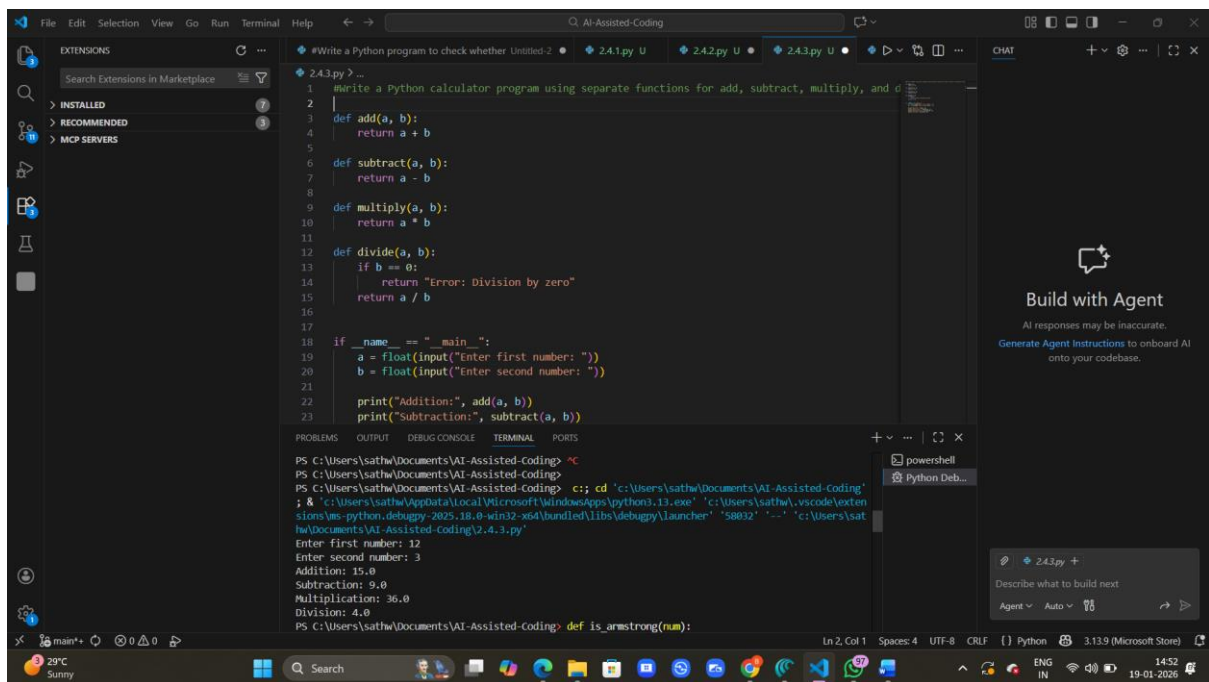
Code and output :



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.

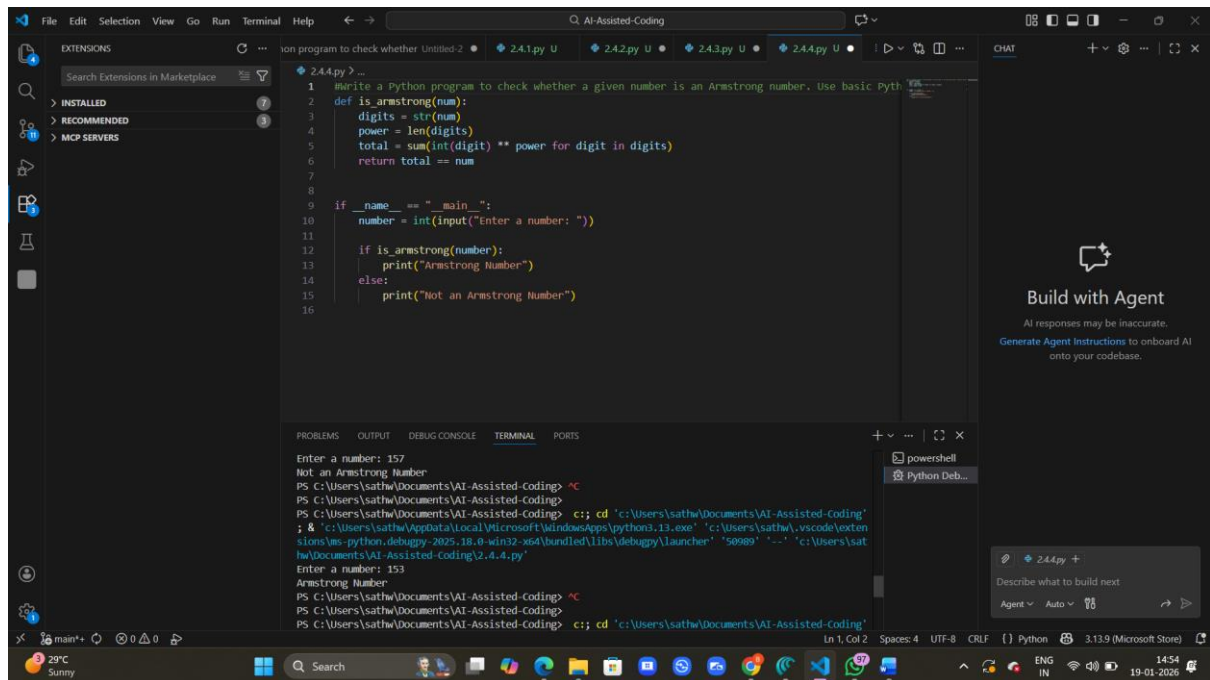
Code and Output:



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:



The screenshot displays the Visual Studio Code interface. The main editor window shows a Python file named `2.4.4.py` with the following code:

```
1 # Write a Python program to check whether a given number is an Armstrong number. Use basic Python constructs and explain briefly.
2 def is_armstrong(num):
3     digits = str(num)
4     power = len(digits)
5     total = sum(int(digit) ** power for digit in digits)
6     return total == num
7
8
9 if __name__ == "__main__":
10     number = int(input("Enter a number: "))
11
12     if is_armstrong(number):
13         print("Armstrong Number")
14     else:
15         print("Not an Armstrong Number")
16
```

Below the code editor, the TERMINAL window shows the program's execution:

```
Enter a number: 157
Not an Armstrong Number
PS C:\Users\sathw\Documents\AI-Assisted-Coding>
PS C:\Users\sathw\Documents\AI-Assisted-Coding> c:: cd 'c:\Users\sathw\Documents\AI-Assisted-Coding'
; & 'c:\Users\sathw\AppData\Local\Microsoft\WindowsApps\python3.13.exe' 'c:\Users\sathw\vscode\extensions\ms-python.debugpy-2025.18.8-win32-x64\bundle\libs\debugpy\launcher' '50989' '-.' 'c:\Users\sathw\Documents\AI-Assisted-Coding\2.4.4.py'
Enter a number: 153
Armstrong Number
PS C:\Users\sathw\Documents\AI-Assisted-Coding>
PS C:\Users\sathw\Documents\AI-Assisted-Coding> c:: cd 'c:\Users\sathw\Documents\AI-Assisted-Coding'
```

On the right side of the interface, there is a 'CHAT' panel with a 'Build with Agent' section, which includes a warning: 'AI responses may be inaccurate. Generate Agent instructions to onboard AI onto your codebase.'