

AI ASSISTED CODING

Name : K.Abhinav

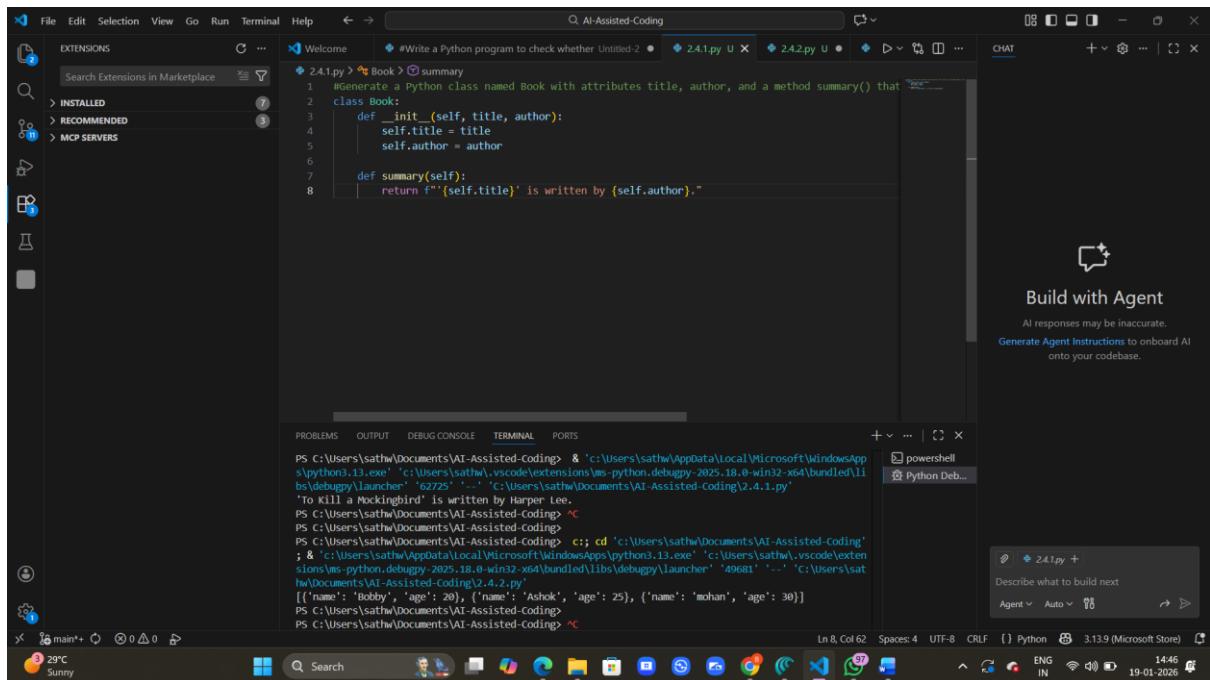
HT No: 2303A51856

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 :



```
#Write a Python program to check whether Untitled-2
# 2.4.1.py ✘ 2.4.2.py ✘
# Generate a Python class named Book with attributes title, author, and a method summary() that returns a formatted string with the title and author.
class Book:
    def __init__(self, title, author):
        self.title = title
        self.author = author

    def summary(self):
        return f'{self.title} is written by {self.author}.'
```

PS C:\Users\sathw\Documents\AI-Assisted-Coding> & 'c:\Users\sathw\appdata\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\sathw\vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debug\Launcher' '62725' '-' 'C:\Users\sathw\Documents\AI-Assisted-Coding\2.4.1.py'
'To Kill a Mockingbird' is written by Harper Lee.
PS C:\Users\sathw\Documents\AI-Assisted-Coding> ^C
PS C:\Users\sathw\Documents\AI-Assisted-Coding> PS C:\Users\sathw\Documents\AI-Assisted-Coding> & 'c:\Users\sathw\appdata\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\sathw\vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debug\Launcher' '49681' '-' 'C:\Users\sathw\Documents\AI-Assisted-Coding\2.4.2.py'
[{'name': 'Bobby', 'age': 20}, {'name': 'Ashok', 'age': 25}, {'name': 'mohan', 'age': 30}]
PS C:\Users\sathw\Documents\AI-Assisted-Coding> PS C:\Users\sathw\Documents\AI-Assisted-Coding> ^C

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 :

The screenshot shows the Microsoft Visual Studio Code interface with the following details:

- File Explorer:** Shows extensions installed, recommended, and MCP servers.
- Editor:** Displays a Python script named `2.4.2.py` containing code to sort a list of dictionaries by age.
- Terminal:** Shows the command line history for running the script.
- Output:** Shows the PowerShell and Python Debug terminals.
- Bottom Status Bar:** Shows the file path as `C:\Users\sathw\Documents\AI-Assisted-Coding\2.4.2.py`, the current line and column as `Ln 1, Col 1`, and the file size as `Spaces: 4`.
- Bottom Taskbar:** Shows the Windows taskbar with various pinned icons.

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:

The screenshot shows the Microsoft Visual Studio Code interface with the following details:

- File Explorer:** Shows extensions installed, recommended, and MCP servers.
- Editor:** Displays a Python script named `2.4.3.py` containing functions for addition, subtraction, multiplication, and division, along with a main block that prompts the user for input and prints the results.
- Terminal:** Shows the command line history for running the script.
- Output:** Shows the PowerShell and Python Debug terminals.
- Bottom Status Bar:** Shows the file path as `C:\Users\sathw\Documents\AI-Assisted-Coding\2.4.3.py`, the current line and column as `Ln 2, Col 1`, and the file size as `Spaces: 4`.
- Bottom Taskbar:** Shows the Windows taskbar with various pinned icons.

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 shows the Visual Studio Code interface with the following details:

- Extensions Bar:** Shows the "AI-Assisted-Coding" extension is active.
- Code Editor:** Displays a Python script named `2.4.4.py` containing code to check if a number is Armstrong. The code uses basic Python constructs like loops, conditionals, and arithmetic.
- Terminal:** Shows the command-line output of running the script with the input `153`. The output indicates that `153` is not an Armstrong number.
- Right Panel:** Features an "AI Assistant" sidebar with a "Build with Agent" button. It includes a note about AI responses being inaccurate and a link to "Generate Agent Instructions".
- Bottom Status Bar:** Shows system information including the date and time (`19-01-2026 14:54`), battery level (`IN`), and network status.