AI ASSISTED CODING-LAB ASSIGNMENT

ASSIGNMENT – 2.4

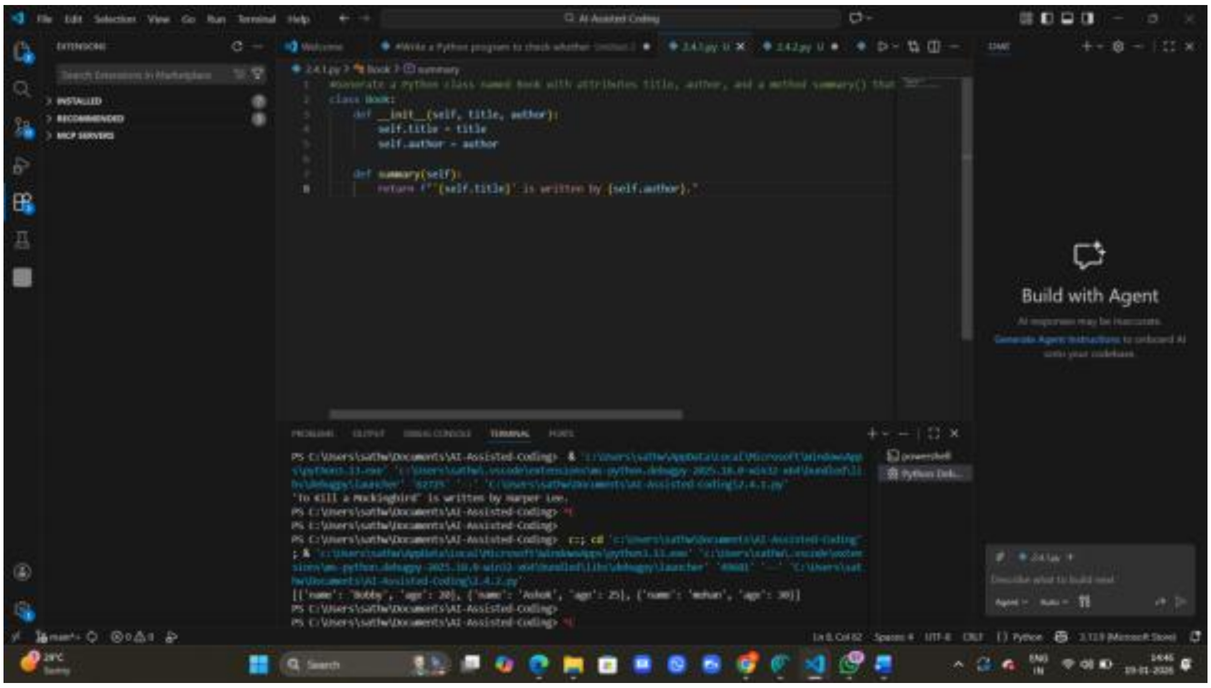
Name:Ch.venugopal

Batch:13

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 :



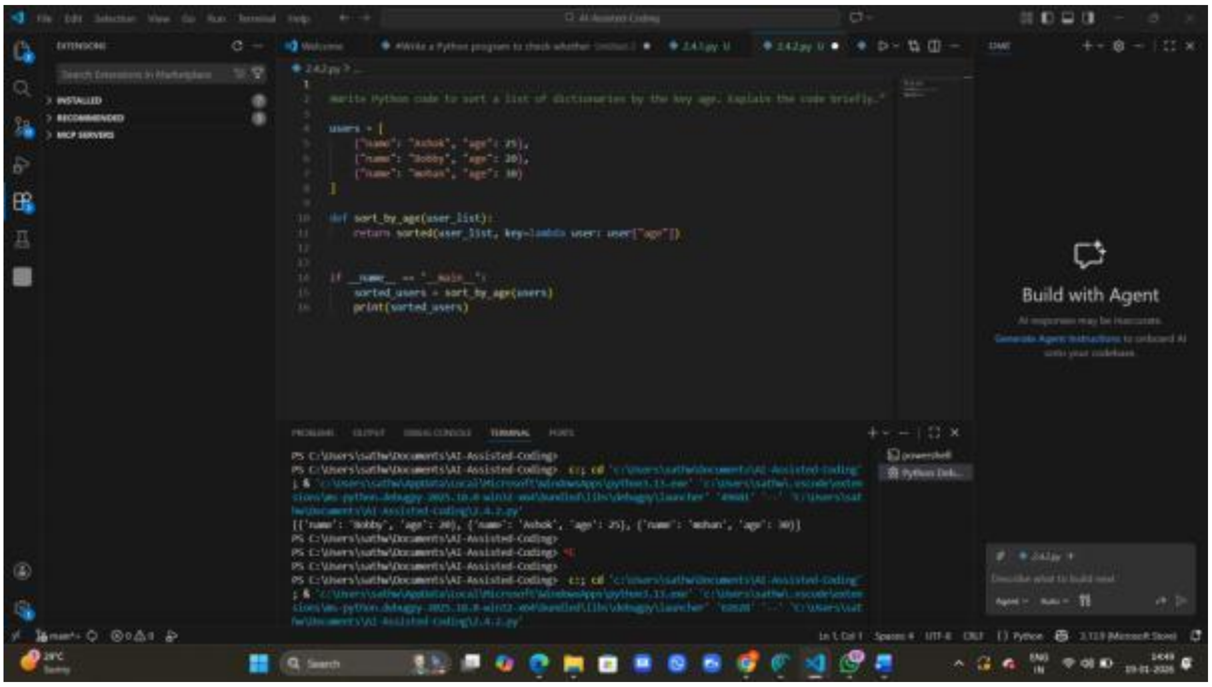
```python
#Generate a Python class named Book with attributes title, author, and a method summary() that
class Book:
    def __init__(self, title, author):
        self.title = title
        self.author = author

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

```
PS C:\Users\sathw\Documents\AI-Assisted-Coding> & [illegible] '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:; cd 'c:\Users\sathw\Documents\AI-Assisted-Coding'; & [illegible] '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 sorta list of dictionaries by the key age. Explain the code briefly.

Code and output :



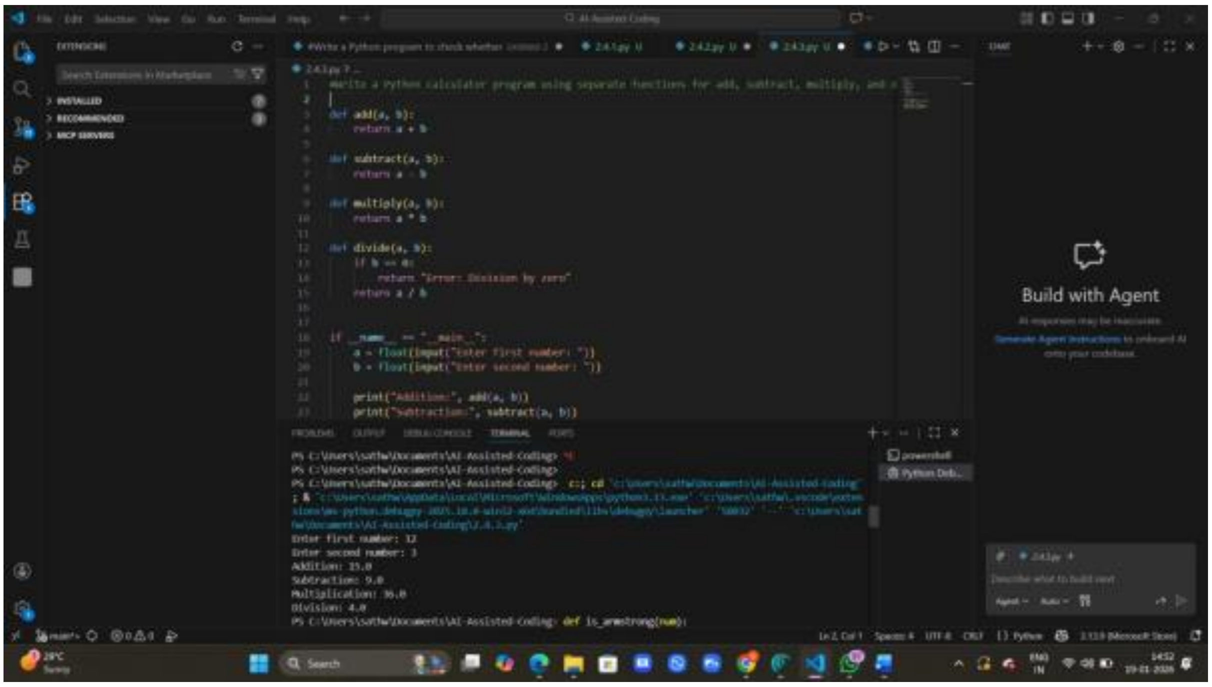
```python
#Write Python code to sort a list of dictionaries by the key age. Explain the code briefly."

users = [
    {"name": "Ashok", "age": 25},
    {"name": "Bobby", "age": 20},
    {"name": "mohan", "age": 30}
]

def sort_by_age(user_list):
    return sorted(user_list, key=lambda user: user["age"])


if __name__ == "__main__":
    sorted_users = sort_by_age(users)
    print(sorted_users)
```

```
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\exten
sions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '49681' '--' 'c:\Users\sat
hw\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
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\exten
sions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '63528' '--' 'c:\Users\sat
hw\Documents\AI-Assisted-Coding\2.4.2.py'
```

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:



```python
#Write a Python calculator program using separate functions for add, subtract, multiply, and d

def add(a, b):
    return a + b

def subtract(a, b):
    return a - b

def multiply(a, b):
    return a * b

def divide(a, b):
    if b == 0:
        return "Error: Division by zero"
    return a / b


if __name__ == "__main__":
    a = float(input("Enter first number: "))
    b = float(input("Enter second number: "))

    print("Addition:", add(a, b))
    print("Subtraction:", subtract(a, b))
```

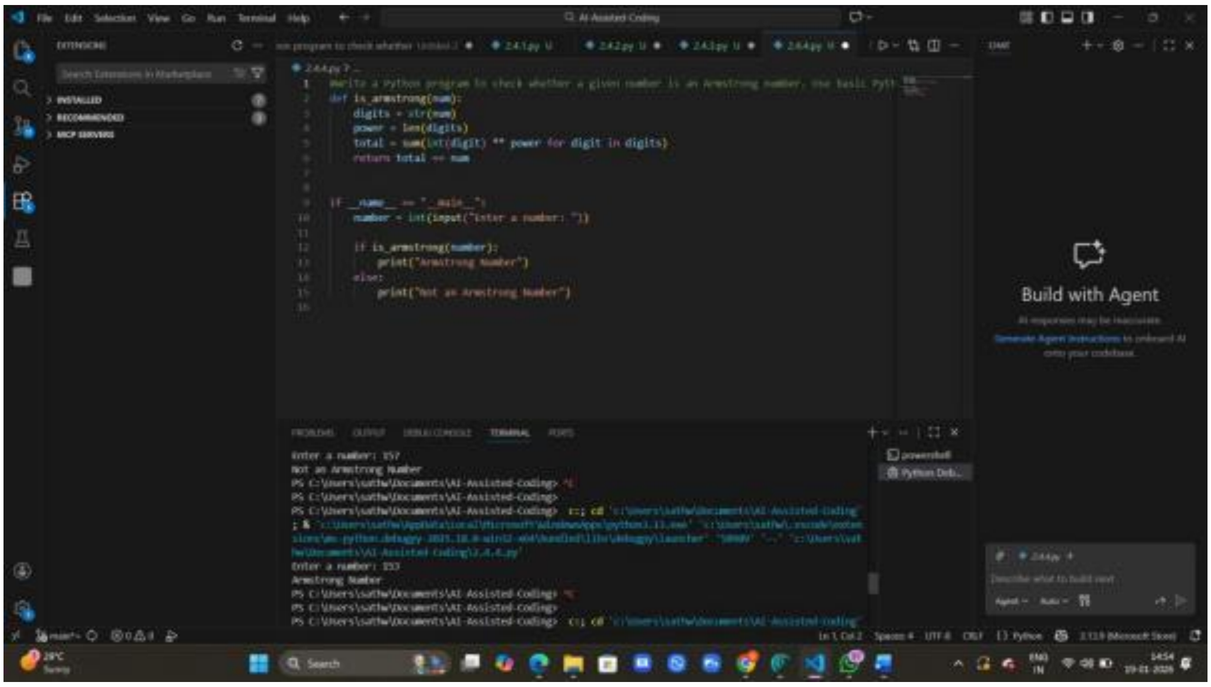
```
PS C:\Users\sathw\Documents\AI-Assisted-Coding> ^C
PS C:\Users\sathw\Documents\AI-Assisted-Coding>
PS C:\Users\sathw\Documents\AI-Assisted-Coding> c:; cd 'c:\Users\sathw\Documents\AI-Assisted-Coding'
; & 'c:\Users\sathw\AppData\Local\Microsoft\WindowsApps\python3.13.exe' 'c:\Users\sathw\.vscode\exten
sions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '50032' '--' 'c:\Users\sat
hw\Documents\AI-Assisted-Coding\2.4.3.py'
Enter first number: 12
Enter second number: 3
Addition: 15.0
Subtraction: 9.0
Multiplication: 36.0
Division: 4.0
PS C:\Users\sathw\Documents\AI-Assisted-Coding> def is_armstrong(num):
```

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:



```python
#Write a Python program to check whether a given number is an Armstrong number. Use basic Pyth
def is_armstrong(num):
    digits = str(num)
    power = len(digits)
    total = sum(int(digit) ** power for digit in digits)
    return total == num


if __name__ == "__main__":
    number = int(input("Enter a number: "))

    if is_armstrong(number):
        print("Armstrong Number")
    else:
        print("Not an Armstrong Number")
```

```
Enter a number: 157
Not an Armstrong Number
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:; cd 'c:\Users\sathw\Documents\AI-Assisted-Coding'
; & 'c:\Users\sathw\AppData\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\sathw\.vscode\exten
sions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '[illegible]' '--' 'c:\Users\sat
hw\Documents\AI-Assisted-Coding\2.4.4.py'
Enter a number: 153
Armstrong Number
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:; cd 'c:\Users\sathw\Documents\AI-Assisted-Coding'
```