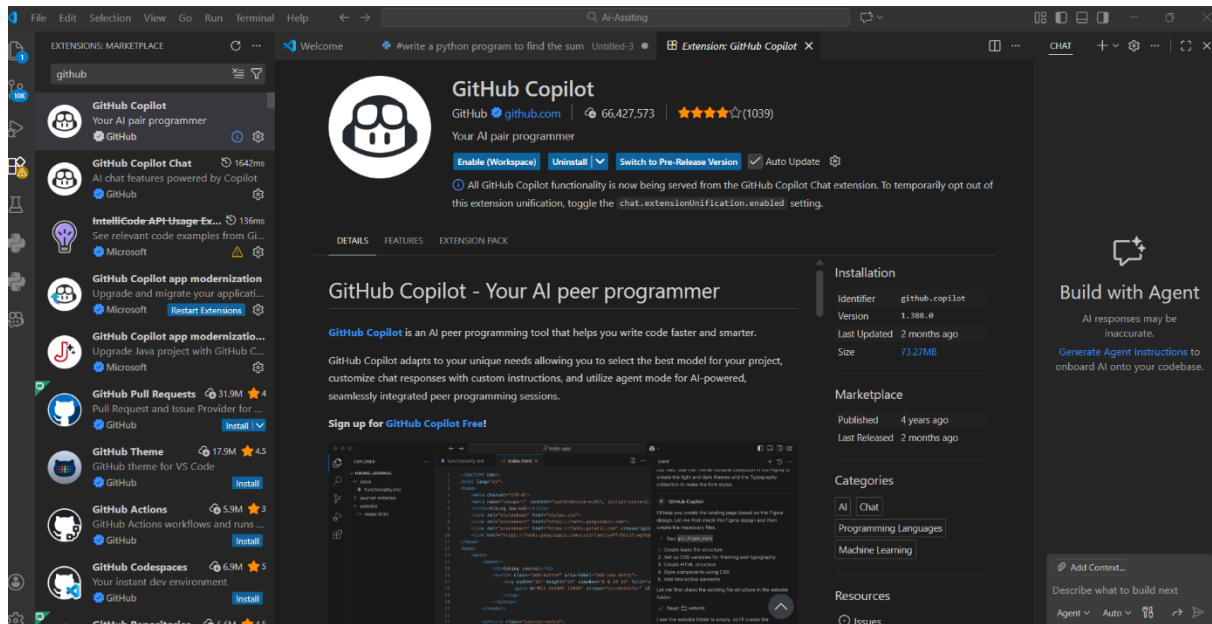


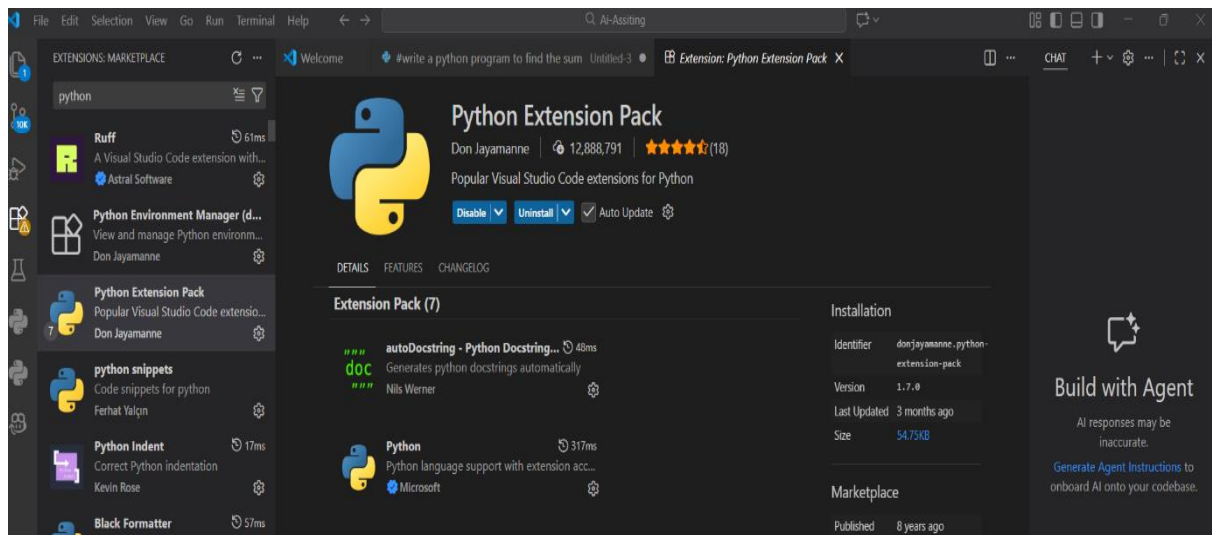
R Akshay

Lab Assignment - 1.3

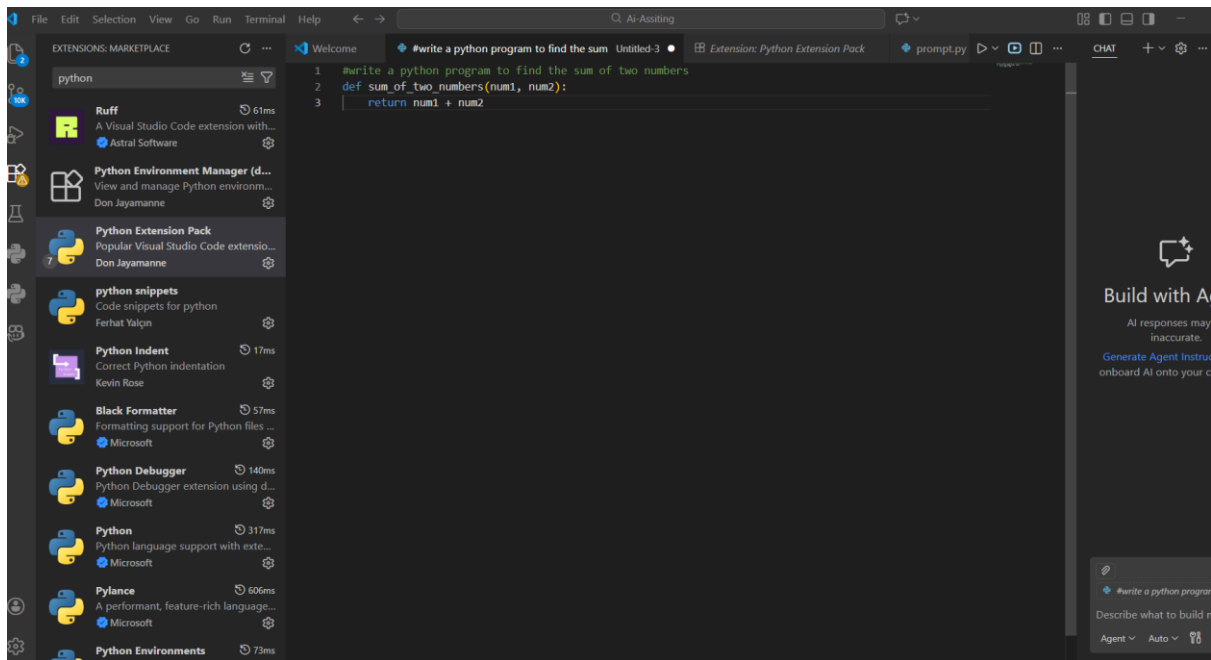
- Downloaded the GitHub Copilot extension in vscode



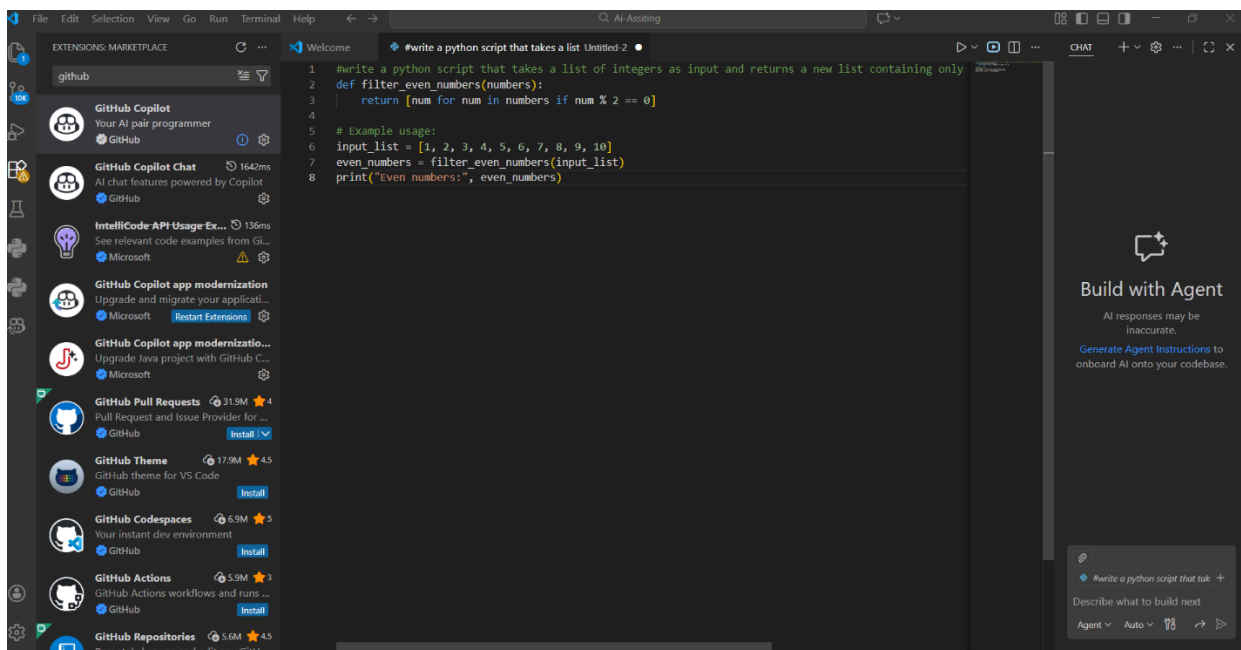
- Downloaded the Python Extension pack in vscode



- Implementation of code with the extension pack



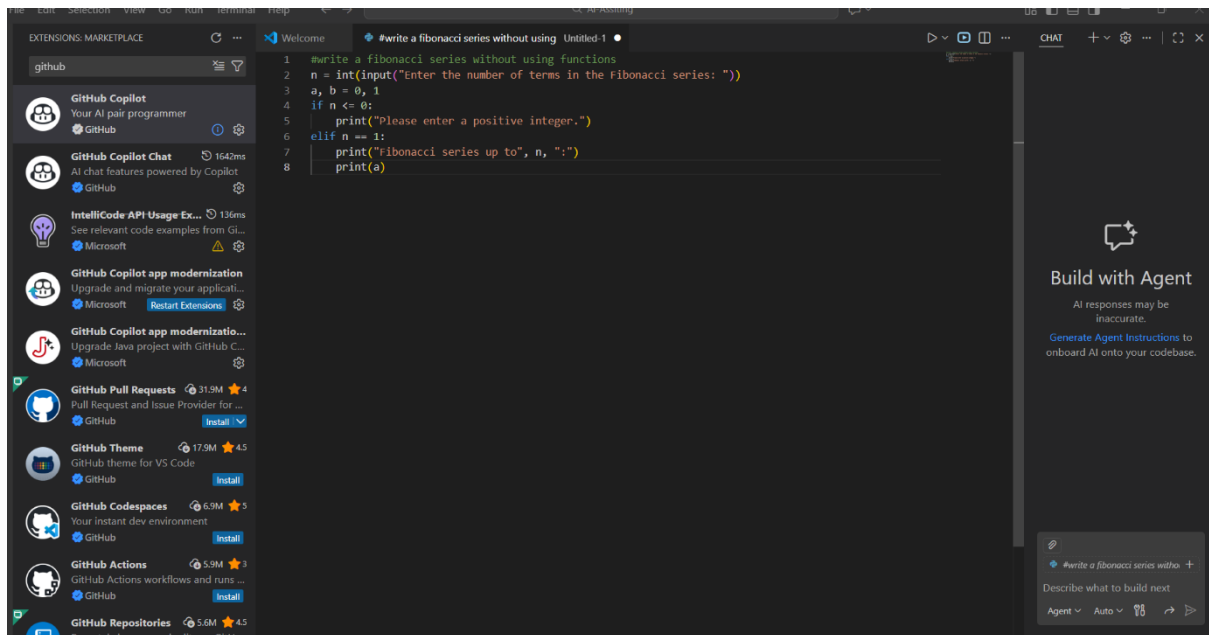
- Implementation of code using github copilot extension



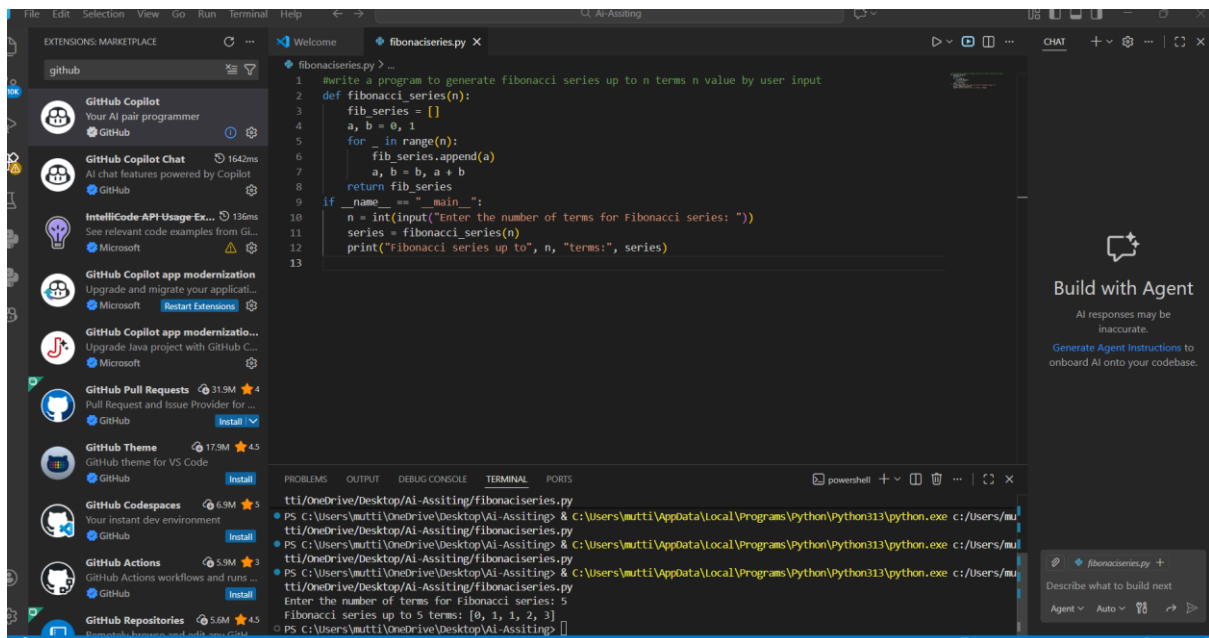
- You are asked to write a quick numerical sequence generator for a learning platform prototype.

Fibonacci series 5 steps

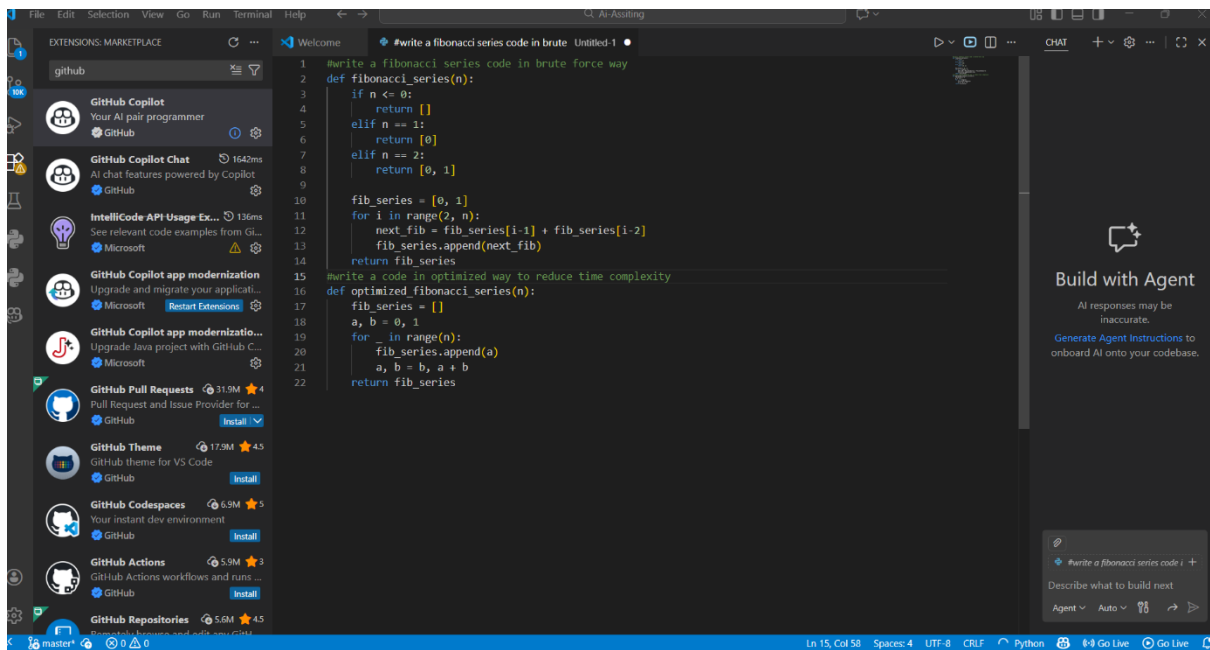
Task 1: AI-Generated Logic Without Modularization (Fibonacci Sequence Without Functions)



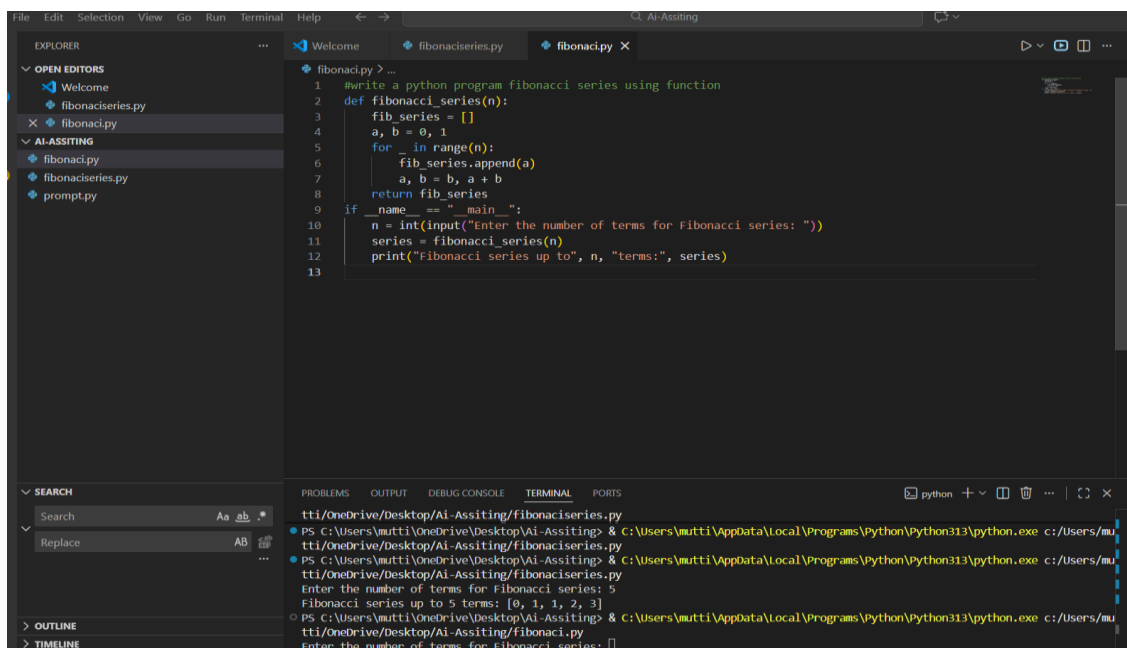
#output for the code of fibonacci series with using functions



Task 2: AI Code Optimization & Cleanup (Improving Efficiency)



Task 3: Modular Design Using AI Assistance (Fibonacci Using Functions)



Task 4: Comparative Analysis – Procedural vs Modular Fibonacci Code

The screenshot shows a Visual Studio Code editor with a Python file named `fibonacci.py`. The code is divided into two sections: one for an iterative Fibonacci series and another for a recursive one. The iterative section uses a loop to calculate the series, while the recursive section uses a function call. The terminal window at the bottom shows the execution of the script, which prompts the user to enter the number of terms for the Fibonacci series. The output shows the series up to 5 terms: [0, 1, 1, 2, 3].

```
1 #write a python program fiboonacci series with function
2 def fibonacci_series(n):
3     fib_series = []
4     a, b = 0, 1
5     for _ in range(n):
6         fib_series.append(a)
7         a, b = b, a + b
8     return fib_series
9 if __name__ == "__main__":
10    n = int(input("Enter the number of terms for Fibonacci series: "))
11    series = fibonacci_series(n)
12    print("Fibonacci series up to", n, "terms:", series)
13 #write a python program fiboonacci series without function
14 n = int(input("Enter the number of terms for Fibonacci series: "))
15 a, b = 0, 1
16 fib_series = []
17 for _ in range(n):
18     fib_series.append(a)
19     a, b = b, a + b
20 print("Fibonacci series up to", n, "terms:", fib_series)
21
```

Task 5: AI-Generated Iterative vs Recursive Fibonacci Approaches (Different Algorithmic Approaches for Fibonacci Series)

The screenshot shows a Visual Studio Code editor with a Python file named `fibonacci.py`. The code is a recursive implementation of the Fibonacci series. It uses a function `fibonacci(n)` that returns the series up to `n` terms. The terminal window at the bottom shows the execution of the script, which prompts the user to enter the number of terms for the Fibonacci series. The output shows the series up to 5 terms: [0, 1, 1, 2, 3].

```
1 #write a python program fibonacci series using recursion
2 def fibonacci(n):
3     if n <= 0:
4         return []
5     elif n == 1:
6         return [0]
7     elif n == 2:
8         return [0, 1]
9     else:
10        fib_series = fibonacci(n - 1)
11        fib_series.append(fib_series[-1] + fib_series[-2])
12        return fib_series
13 if __name__ == "__main__":
14    n = int(input("Enter the number of terms for Fibonacci series: "))
15    series = fibonacci(n)
16    print("Fibonacci series up to", n, "terms:", series)
17
```