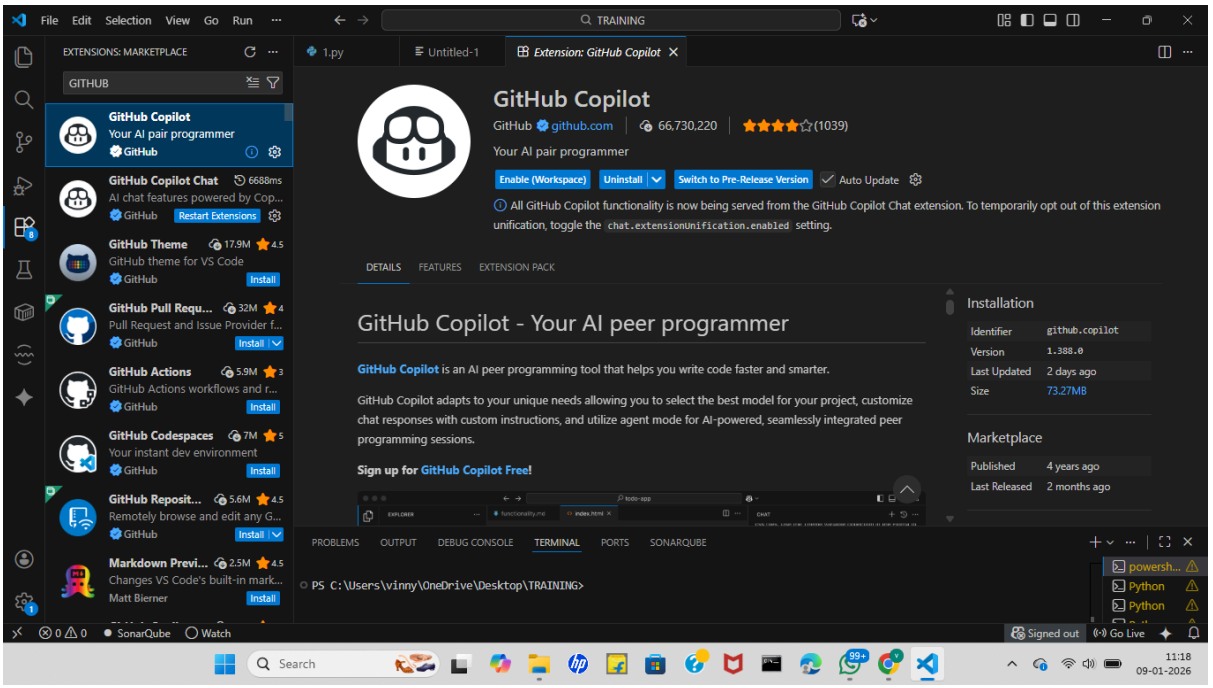
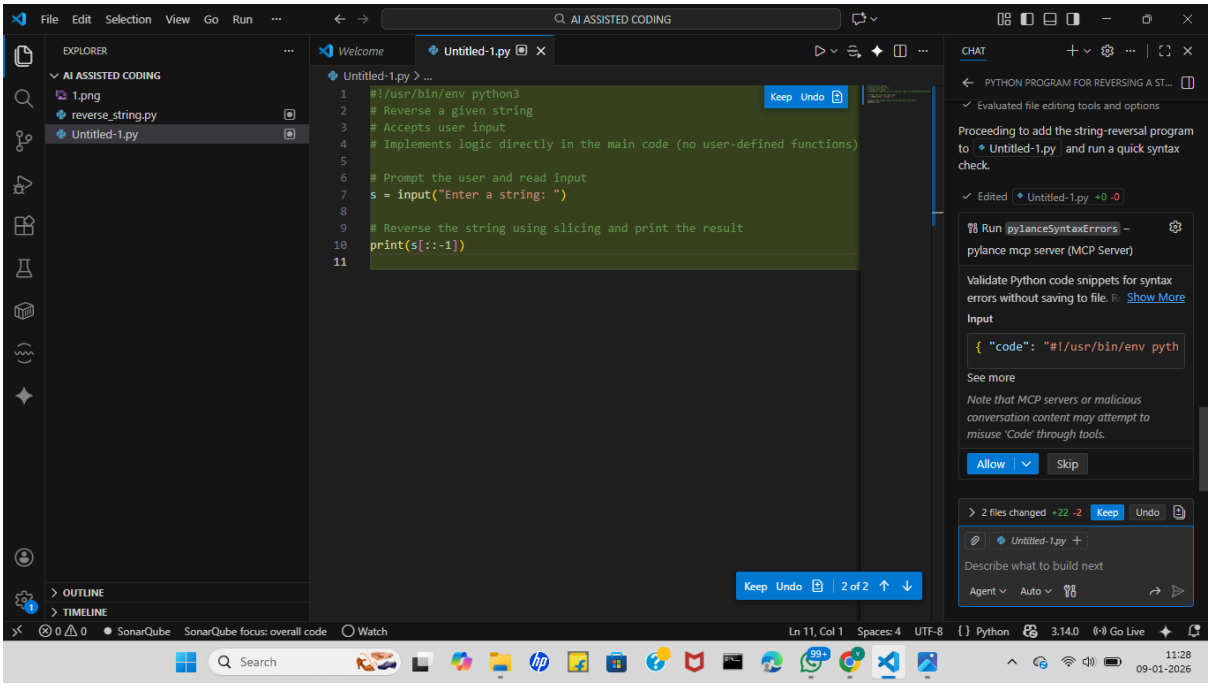
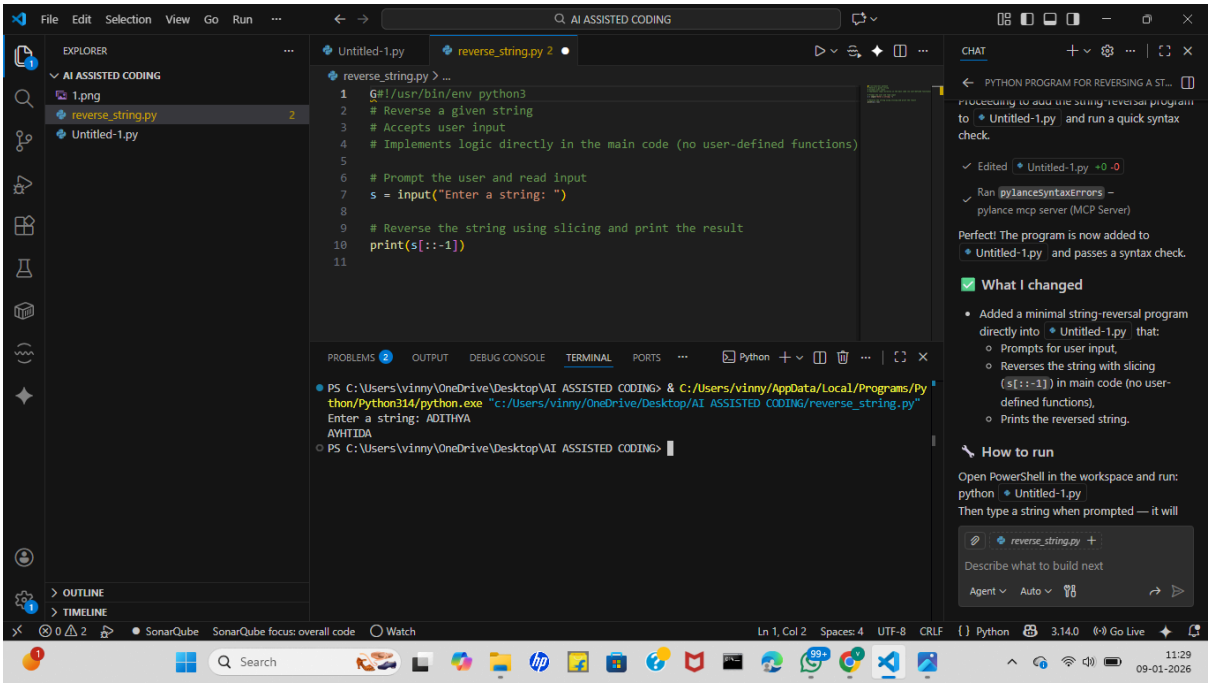
Batch 03
2303A51449

Task Description 1:
Use GitHub Copilot to generate a Python program that:
➢ Reverses a given string
➢ Accepts user input
➢ Implements the logic directly in the main code
➢ Does not use any user-defined functions







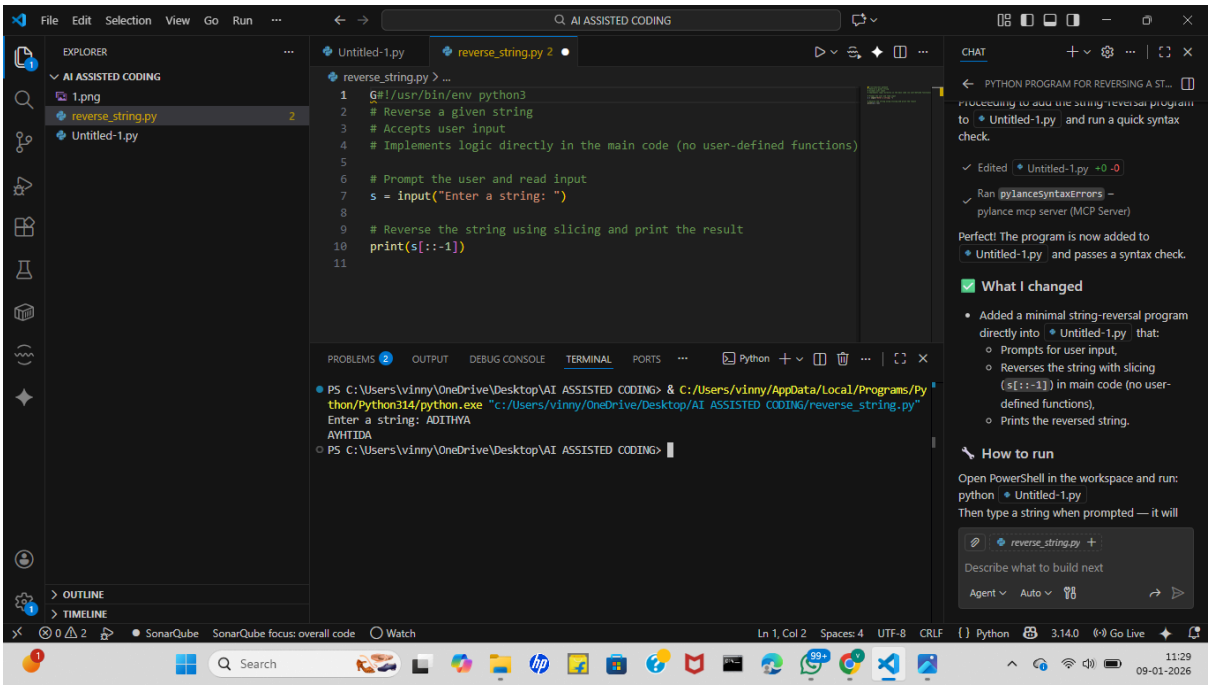

```
Enter a string: ADITHYA
AYHTIDA
```

Task Description 2:

Examine the Copilot-generated code from Task 1 and improve it by:

➢ Removing unnecessary variables
➢ Simplifying loop or indexing logic
➢ Improving readability
➢ Use Copilot prompts like:

- "Simplify this string reversal code"
- "Improve readability and efficiency"

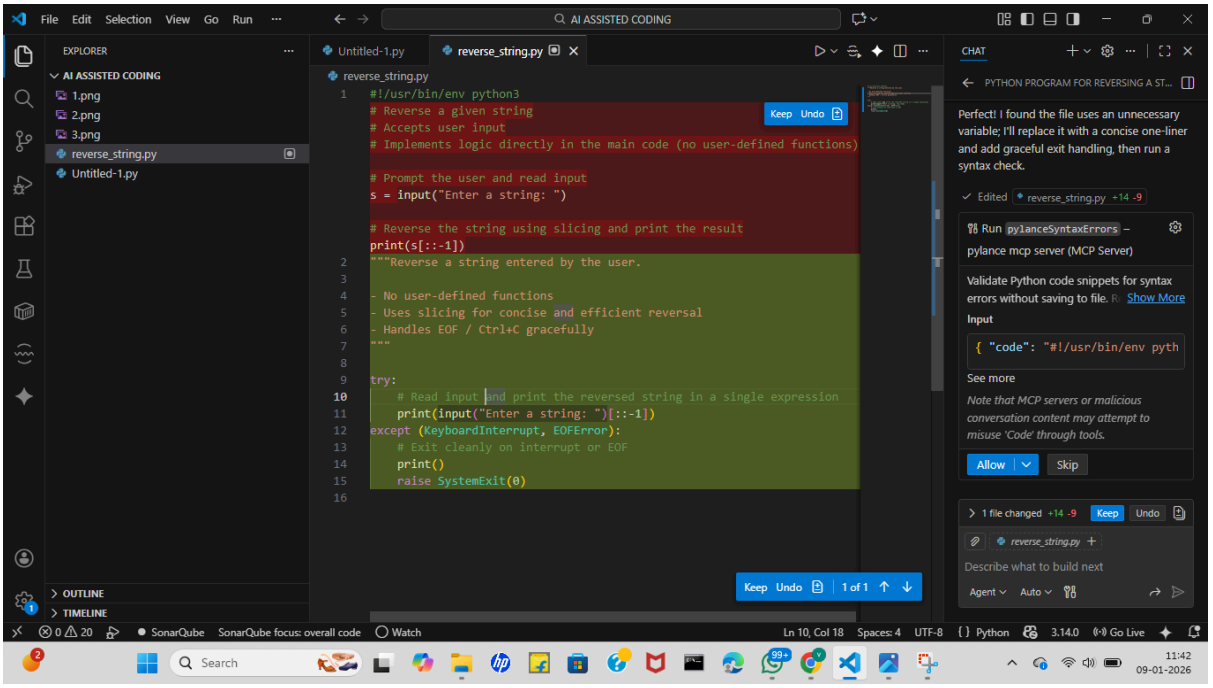


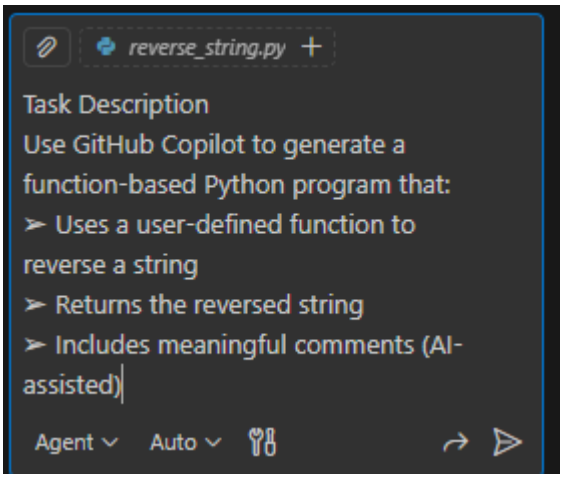
```
PS C:\Users\vinny\OneDrive\Desktop\AI ASSISTED CODING> & C:/Users/vinny/AppData/Local/Programs/Python/Python314/python.exe "c:/Users/vinny/OneDrive/Desktop/AI ASSISTED CODING/reverse_string.py"
Enter a string: ADITHYA
AYHTIDA
PS C:\Users\vinny\OneDrive\Desktop\AI ASSISTED CODING>
```

OPTIMIZE THE CODE,SIMPLIOFT THE LOGIC,MAKE IT MORE READABLE

Task Description 3:

Use GitHub Copilot to generate a function-based Python program that:
➢ Uses a user-defined function to reverse a string
➢ Returns the reversed string
➢ Includes meaningful comments (AI-assisted)






```
Enter a string: ADITHYA
AYHTIDA
```

TASK DESCRIPTION 4:

Compare the Copilot-generated programs:
➢ Without functions (Task 1)
➢ With functions (Task 3)
Analyze them based on:
➢ Code clarity
➢ Reusability
➢ Debugging ease
➢ Suitability for large-scale applications

OUTPUT:

Quick summary table 📊

Criterion	reverse_string.py **(no-function)**	reverse_string_function.py **(function)**
Code clarity	✅ Very concise but minimal context	✅ Explicit and documented
Reusability	❌ Low — hard to import/test	✅ High — straightforward to import/use
Debugging ease	❌ Harder to unit-test / step-debug	✅ Easy to unit-test and inspect
Large-scale suitability	❌ Not suitable for scale	✅ Well-suited for scale & extension

SHORT ANALYTICAL REPORT:

The function-based version (reverse_string_function.py) is clearer, more reusable, and easier to debug, making it the better foundation for production.

The one-liner (reverse_string.py) is concise and fine for quick interactive use, but not suitable for scale.

Analysis (concise)

Code clarity

Function: High — docstrings, type hints, explicit behavior.

One-liner: Low–Medium — very short but lacks context and docs.

Reusability

Function: High — importable and can be reused in other code.

One-liner: Poor — logic embedded in main flow.

Debugging ease :

Function: Easy — pure function, straightforward unit tests.

One-liner: Harder — one expression, hard to step through or test.

Suitability for large-scale apps :

Function: Good (with tests, CI, validation, and monitoring).

One-liner: Unsuitable — no API, tests, or automation hooks.

Task Description 5:

Prompt GitHub Copilot to generate:

➢ A loop-based string reversal approach

➢ A built-in / slicing-based string reversal approach