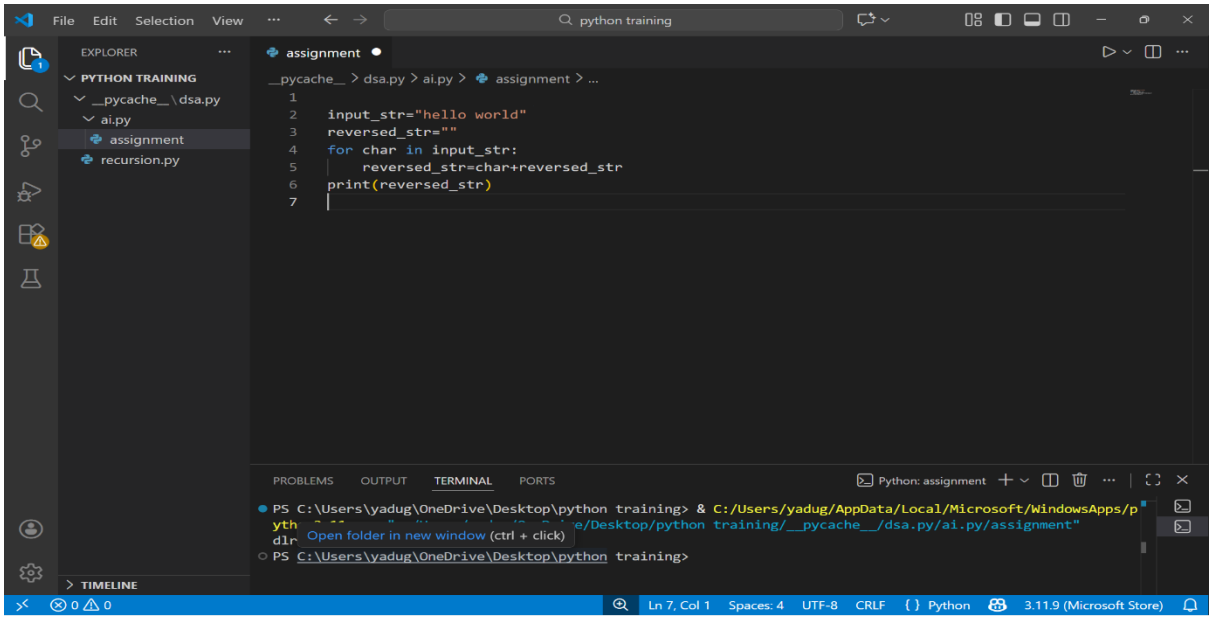
Assignment-1.5

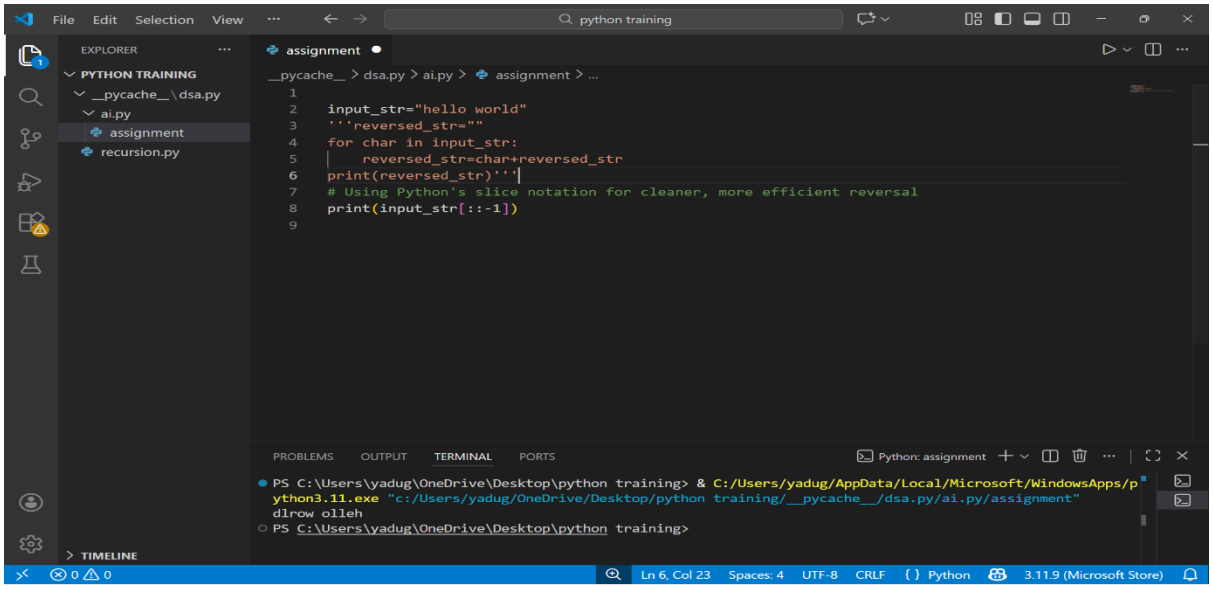
Name: K.Likhith

RollNo:2303A51331

TASK-1 AI-Generated Logic Without Modularization (String Reversal Without Functions)



TASK-2 Efficiency & Logic Optimization (Readability Improvement)



- Explaination: [::-1] tells Python:
    - Start from end

- Move backwards
- Step = -1

- Python internally reverses the string in **one pass**.

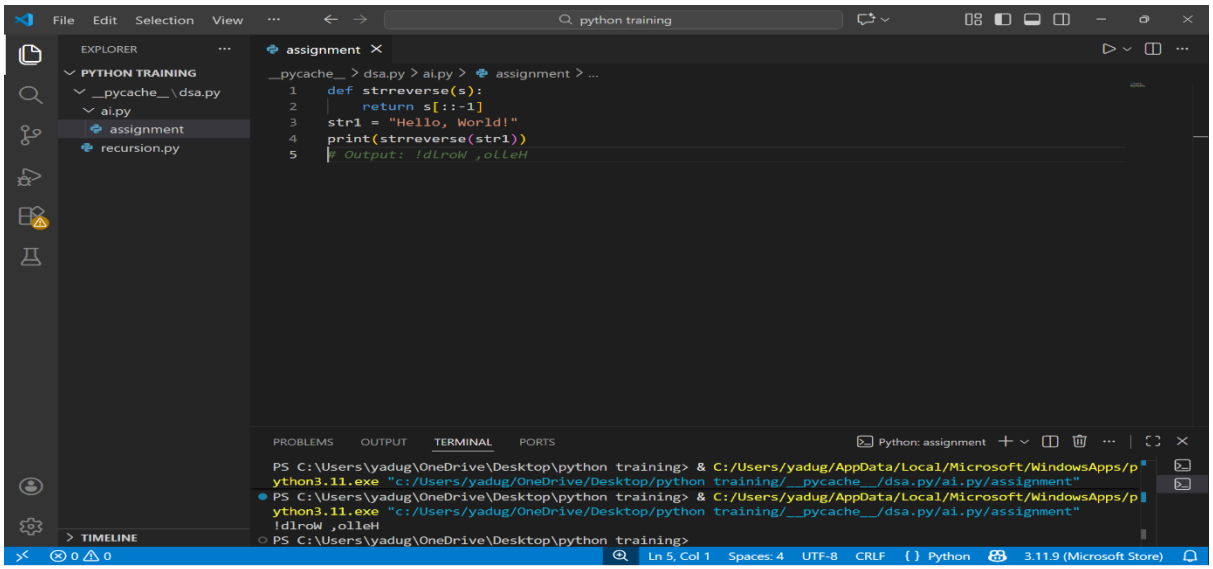
Time Complexity

- Each character is accessed **once**.
- No repeated copying like the loop version.

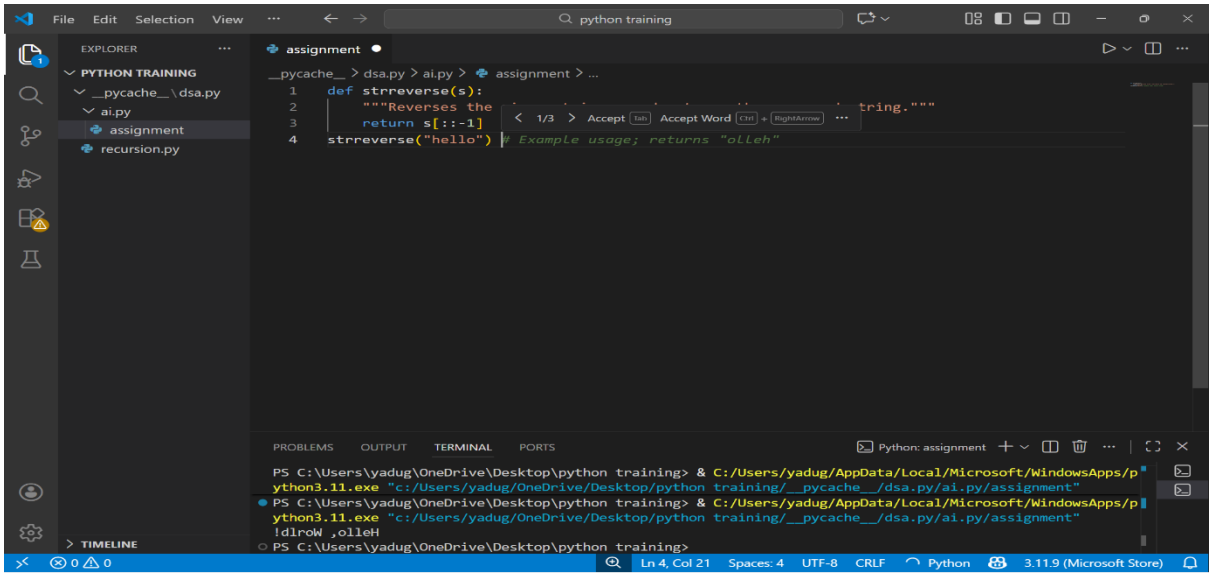
Time Complexity**:** O(n)

Task-3: Modular Design Using AI Assistance (String Reversal Using Functions)

Before optimization



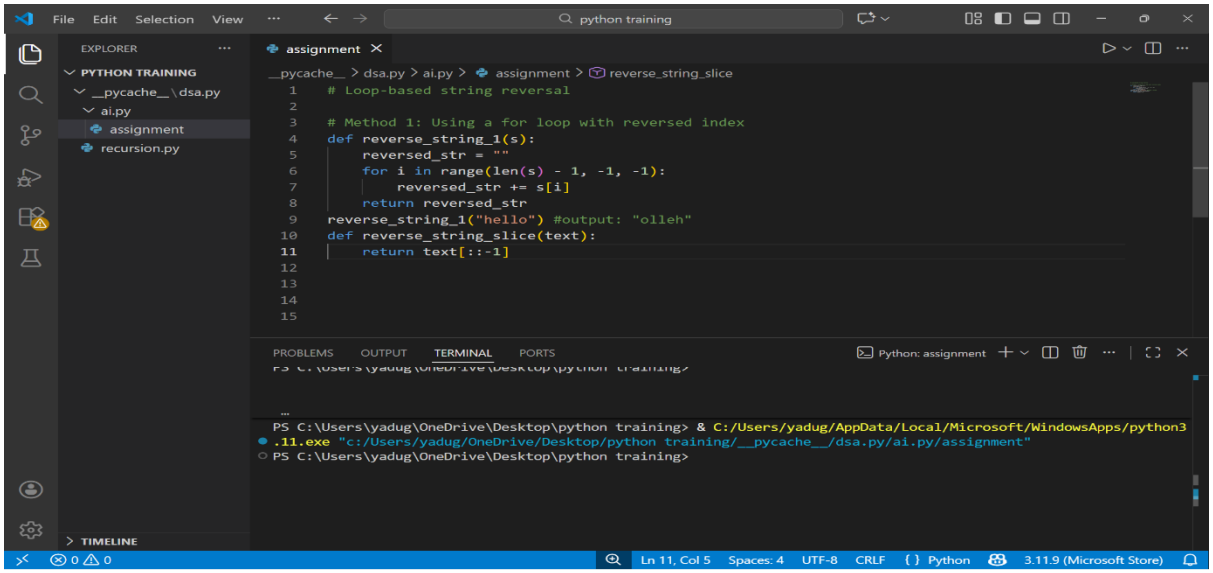
After optimization using copilot



Task-4 Comparative Analysis – Procedural vs Modular Approach (With vs Without Functions)

Criteria	Without Functions (Procedural)	With Functions (Modular)
Code Clarity	Logic mixed with execution, harder to read in large files	Clear separation of logic and execution
Reusability	Code must be rewritten for every use	Function can be reused anywhere
Debugging Ease	Bugs are harder to isolate	Easy to debug by testing the function alone
Scalability	Poor for large programs	Highly suitable for large applications
Maintenance	Changes must be made in multiple places	Changes made once in the function
Testing	Manual testing only	Supports unit testing easily
Readability	Acceptable for small scripts	Excellent for professional codebases
Copilot Assistance	Limited suggestions	Strong AI support for docstrings & logic

Task-5 AI-Generated Iterative vs Recursive Fibonacci Approaches (Different Algorithmic Approaches to String Reversal)



Explaination:

Both approaches correctly reverse strings.

Slicing-based reversal is superior in performance, readability, and scalability.