AI ASSISTED CODING

LAB TEST-02

NAME:SUNKARI NAGARJUNA REDDY

HTNO:2403A52064

BATCH:03

SUBGROUP-F

F.1 — [S14F1] Add type hints and fix None handling

Scenario (telecom network):

Context:

A helper in telecom network sometimes returns None, causing downstream type errors.

Your Task:

Annotate types and ensure the function always returns str or raises ValueError if not found.

Data & Edge Cases:

Given ids=['a','b'] and target='c' -> expect ValueError('not found').

AI Assistance Expectation:

Use AI to propose a typed signature and non-None return guarantees.

Constraints & Notes:

Include a negative test that expects ValueError.

Sample Input

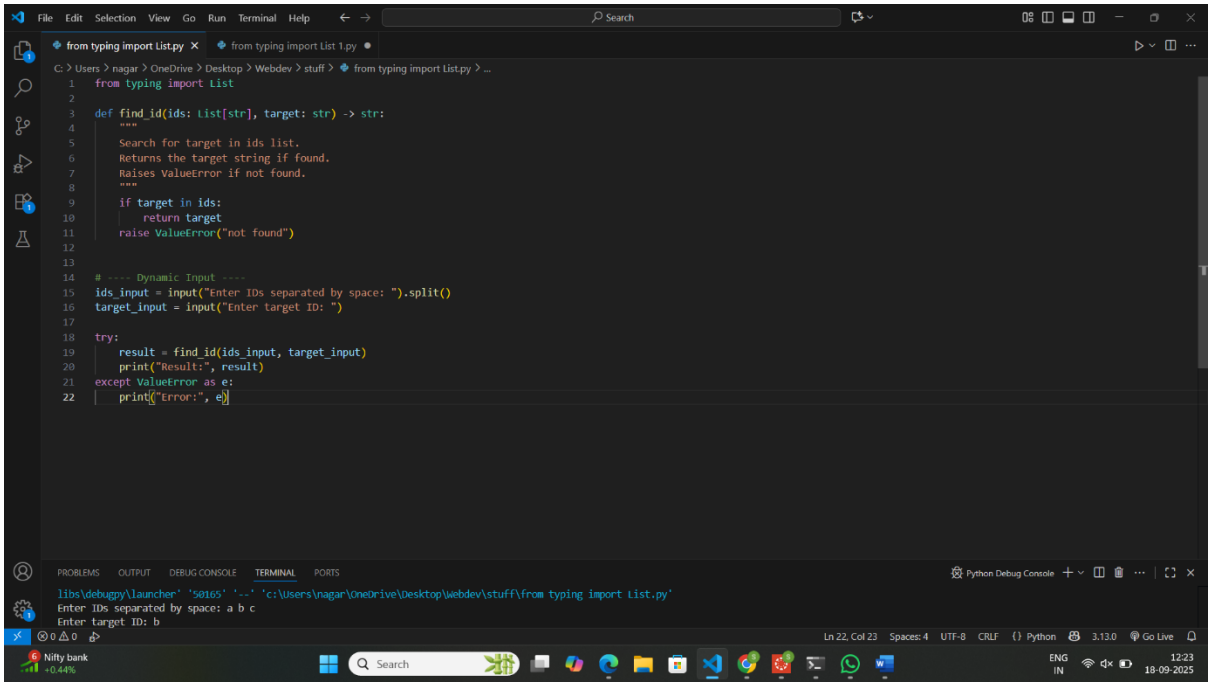
ids=['a','b'], target='c'

Sample Output

ValueError('not found')

Acceptance Criteria: Typed; raises on miss; no implicit None

CODE:



```
from typing import List

def find_id(ids: List[str], target: str) -> str:
    """
    Search for target in ids list.
    Returns the target string if found.
    Raises ValueError if not found.
    """
    if target in ids:
        return target
    raise ValueError("not found")


# ---- Dynamic Input ----
ids_input = input("Enter IDs separated by space: ").split()
target_input = input("Enter target ID: ")

try:
    result = find_id(ids_input, target_input)
    print("Result:", result)
except ValueError as e:
    print("Error:", e)
```

```
libs\debugpy\launcher' '50165' '--' 'c:\Users\nagar\OneDrive\Desktop\Webdev\stuff\from typing import List.py'
Enter IDs separated by space: a b c
Enter target ID: b
```

OUTPUT:

```
PS C:\Users\nagar\OneDrive\Desktop\Webdev\stuff> & 'c:\Users\nagar\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\nagar\.vscode\extensions\ms-python.debugpy-2025.10.0-win32-x64\bundled\
libs\debugpy\launcher' '50165' '--' 'c:\Users\nagar\OneDrive\Desktop\Webdev\stuff\from typing import List.py'
Enter IDs separated by space: a b c
Enter target ID: b
Result: b
PS C:\Users\nagar\OneDrive\Desktop\Webdev\stuff> ^C
PS C:\Users\nagar\OneDrive\Desktop\Webdev\stuff>
PS C:\Users\nagar\OneDrive\Desktop\Webdev\stuff>  c:; cd 'c:\Users\nagar\OneDrive\Desktop\Webdev\stuff'; & 'c:\Users\nagar\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\nagar\.vscode\ex
tensions\ms-python.debugpy-2025.10.0-win32-x64\bundled\libs\debugpy\launcher' '64067' '--' 'c:\Users\nagar\OneDrive\Desktop\Webdev\stuff\from typing import List.py'
Enter IDs separated by space: a b
Enter target ID: z
Error: not found
PS C:\Users\nagar\OneDrive\Desktop\Webdev\stuff> ^C
```

OBSERVATION:

1.If the target is present-prints it.

2.If the target is absent-raises valueerror and prints the error.

F.2 — [S14F2] Optimize membership checks

Scenario (telecom network):

Context:

A streaming job in telecom network checks if IDs are in a large corpus.

Your Task:

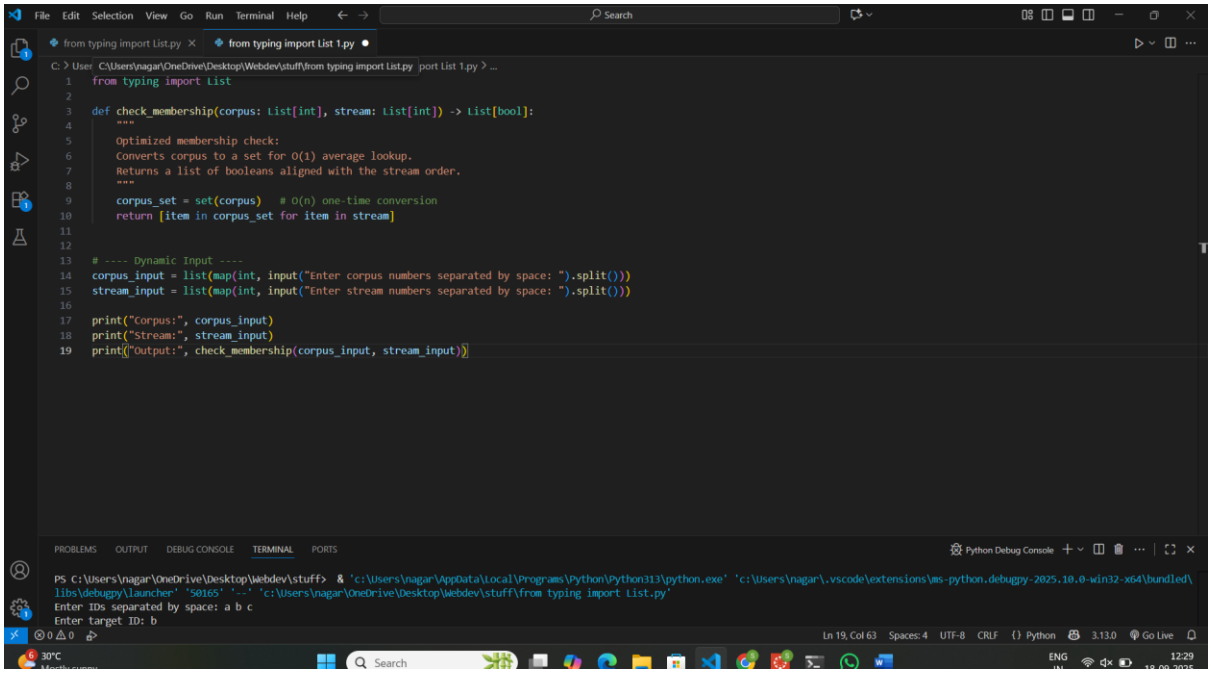
Optimize membership checks by converting corpus to a set once, then mapping stream to
booleans.

Data & Edge Cases:

corpus=[1,2,3,4,5]; stream=[2,5,9] -> [True, True, False].

AI Assistance Expectation:
Ask AI to suggest complexity improvements and micro-bench ideas.
Constraints & Notes:
Return list[bool] aligned to stream order.
Sample Input
corpus=[1,2,3,4,5], stream=[2,5,9]
Sample Output
[True, True, False]
Acceptance Criteria: Uses set; correct Booleans

CODE:



OUTPUT:

OBSERVATION:

The solution uses a set,which makes membership checking very fast and independent of input size.

The output matches the stream order giving the correct sequence of boolen values.