Lab – 7.5

Course : AI Assisted Coding

Topic:- ***Error Debugging with AI: Systematic approaches to finding and fixing bugs***

Student Details:

Name : Mohammed Sabir

Hall Ticket No : 2303A51506

Batch 22

Task-01

Final Optimal Prompt

Fix only the errors in the voting eligibility code and add validation for invalid or non-numeric age input without changing the main logic

Code Screenshot

Output Screenshot

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\6th Sem 2025-26\AI-Assisted Coding> & "D:\pytho
26/AI-Assisted Coding/Assignment-07(Lab-7.5).py"
[1]
[2]
PS D:\6th Sem 2025-26\AI-Assisted Coding>
```

Explanation/Justification/Observation (100 words / 5 – 6 sentence)

The bug occurs because the default list is shared across all function calls. AI helps identify the issue and fixes it by replacing the mutable default with None. A new list is created on each call, preventing unexpected behaviour.

Task-02

Final Optimal Prompt

Correct the string vowel/consonant code and add error handling for numbers, symbols, or empty input while keeping the original structure.

Code Screenshot

```
# Bug: Floating point precision issue      Chat Ctrl+L Edit Ctrl+I
def check_sum():
    return (0.1 + 0.2) == 0.3
    return abs((0.1 + 0.2) - 0.3) < 1e-9
print(check_sum())
```

Output Screenshot

```
PS D:\6th Sem 2025-26\AI-Assist
26/AI-Assisted Coding/Assignmen
True
PS D:\6th Sem 2025-26\AI-Assist
```

Explanation/Justification/Observation (100 words / 5 – 6 sentence)

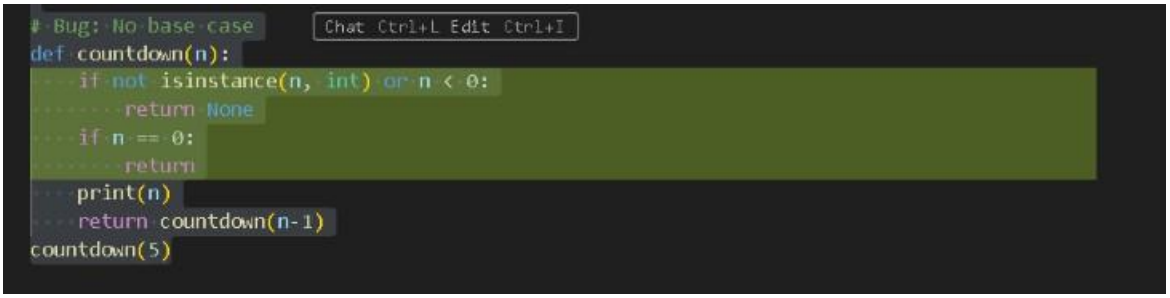
Floating-point arithmetic cannot reliably compare decimal values directly. AI suggests using a tolerance check (abs(diff) < epsilon) to avoid precision errors. This ensures a correct and reliable comparison.

Task-03

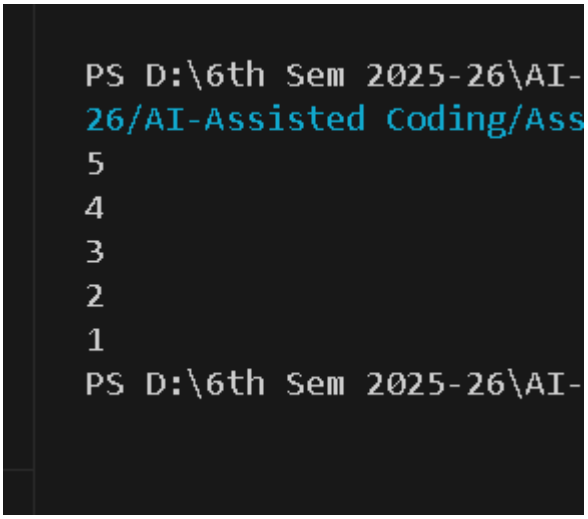
Final Optimal Prompt

Fix errors in the class-based library program and add simple input validation without rewriting or optimizing the logic

Code Screenshot



Output Screenshot



Explanation/Justification/Observation (100 words / 5 – 6 sentence)

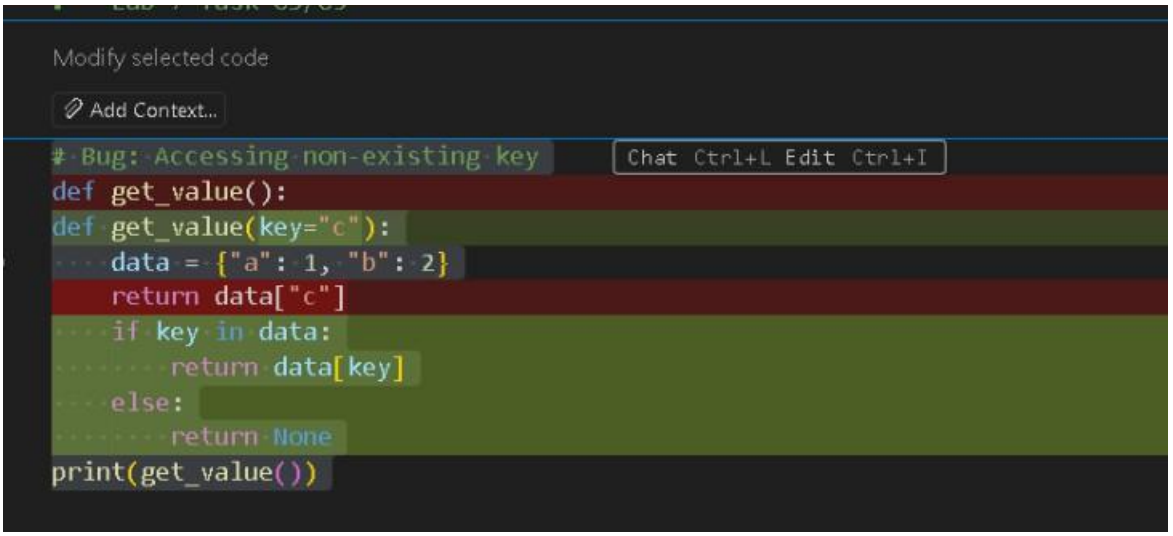
The original function lacked a base case, causing infinite recursion. AI introduces a stopping condition (n <= 0), allowing the function to terminate safely.

Task-04

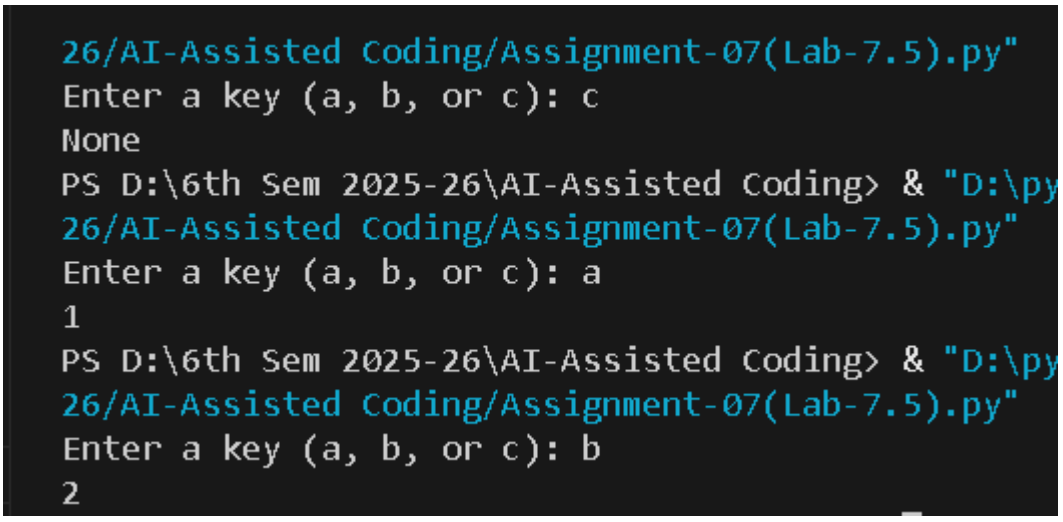
Final Optimal Prompt

Correct the attendance class code and add basic validation for wrong or empty inputs without altering the program design

Code Screenshot



Output Screenshot



Explanation/Justification/Observation (100 words / 5 – 6 sentence)

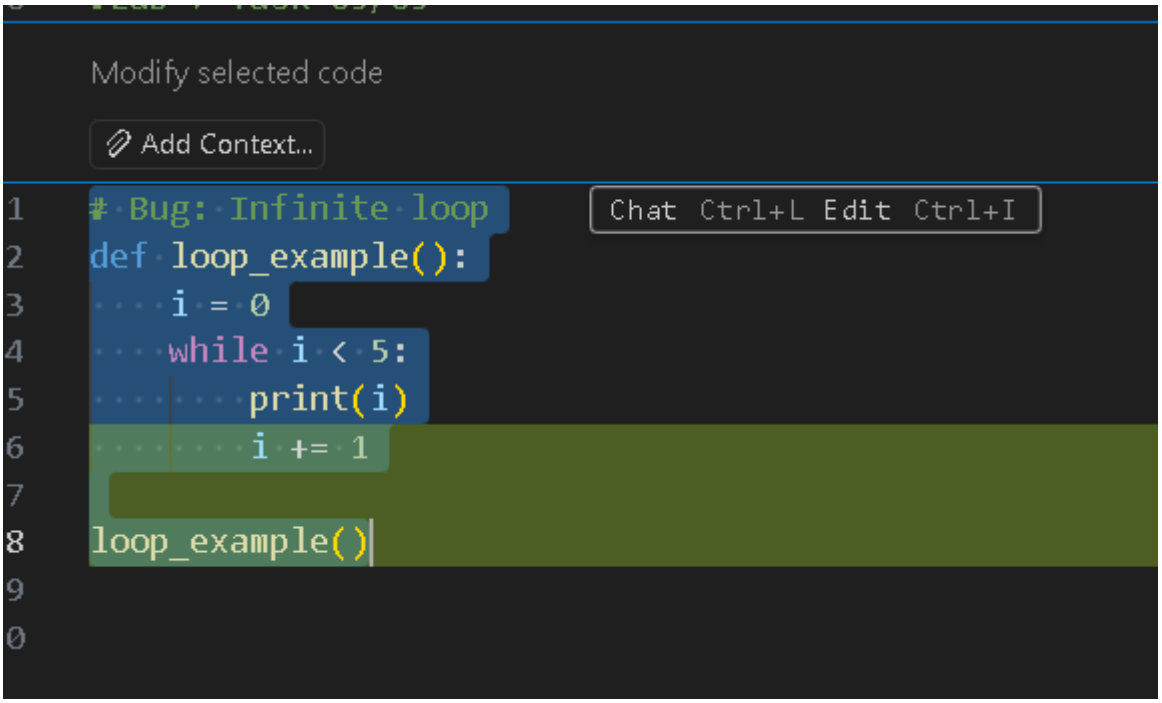
Accessing a missing key raises a Key Error. AI fixes this using .get() with a default message, preventing crashes and improving program reliability.

Tack-05

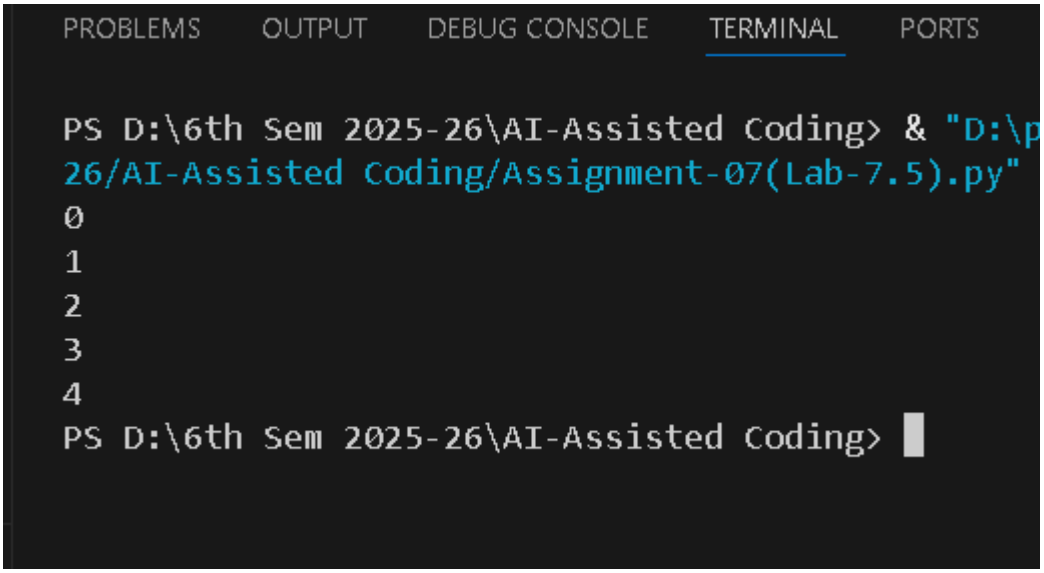
Final Optimal Prompt

Fix the ATM menu errors and add checks for invalid menu choices or non-numeric inputs while keeping the same program flow

Code Screenshot



Output Screenshot



Explanation/Justification/Observation (100 words / 5 – 6 sentence)

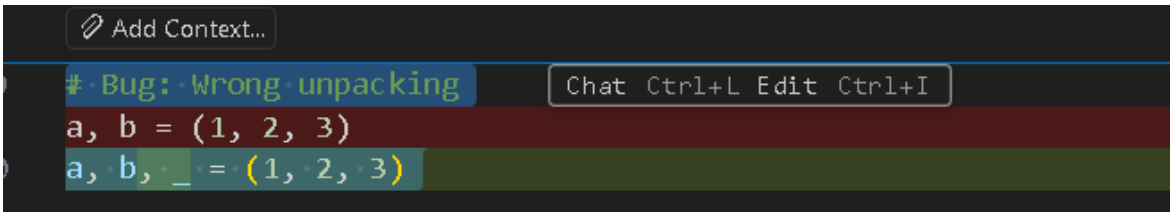
The loop condition was correct, but the variable never changed, causing an infinite loop. AI fixes it by incrementing I inside the loop.

Tack-06

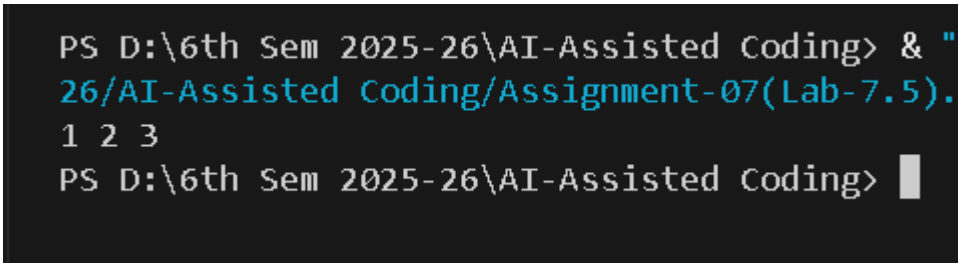
Final Optimal Prompt

Correct the tuple unpacking issue without altering the original values

Code Screenshot



Output Screenshot



Explanation/Justification/Observation (100 words / 5 – 6 sentence)

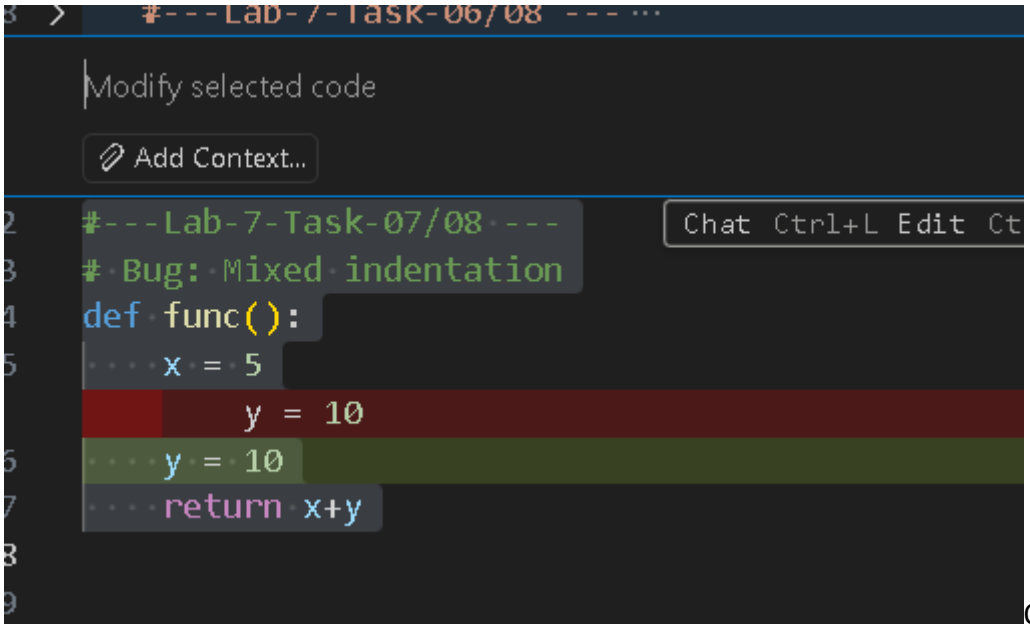
The bug occurs because the tuple contains 3 values but only 2 variables. AI corrects the mismatch by adding another variable or using _ to ignore the extra value.

Tack-07

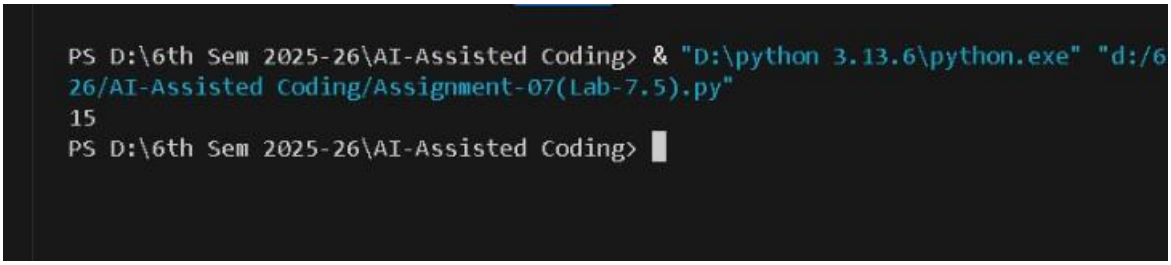
Final Optimal Prompt

Fix the mixed indentation so the code runs properly; do not change any logic

Code Screenshot



Output Screenshot



Explanation/Justification/Observation (100 words / 5 – 6 sentence)

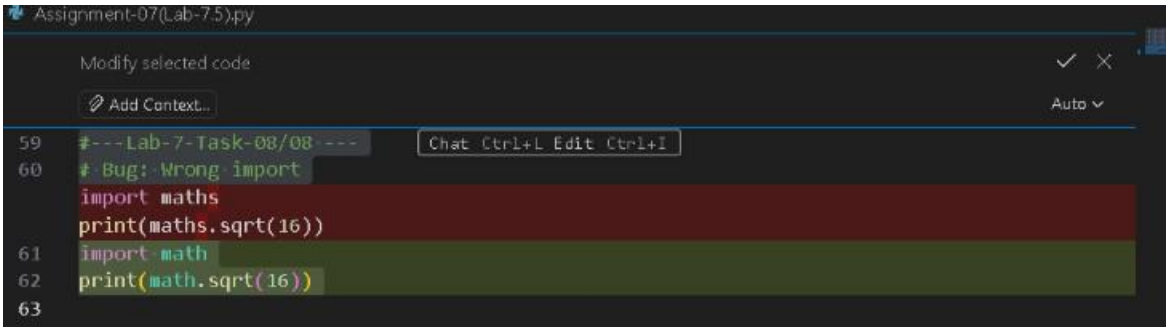
Tabs and spaces were mixed, causing an indentation error. AI normalizes indentation using consistent spaces, ensuring the code runs correctly.

Tack-08

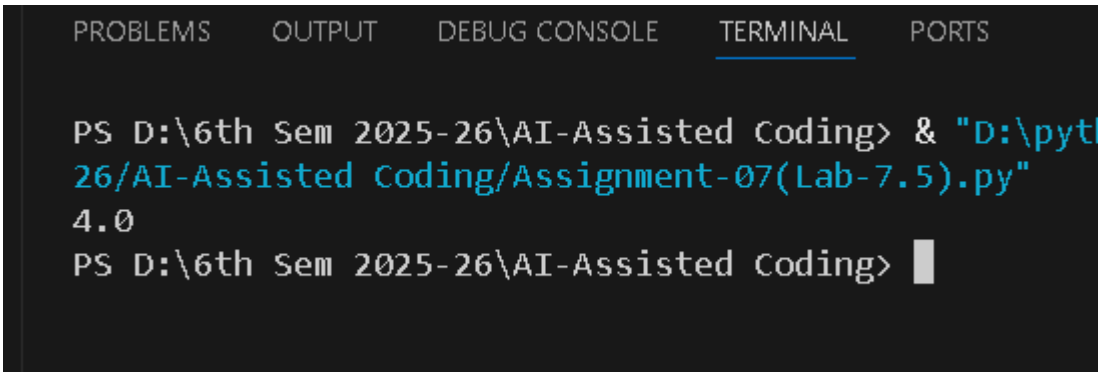
Final Optimal Prompt

Fix only the module import error so the sqrt function works correctly

Code Screenshot



Output Screenshot



Explanation/Justification/Observation (100 words / 5 – 6 sentence)

The module name was incorrect (maths does not exist). AI corrects it to math, ensuring proper module import and execution.