Assignment-07

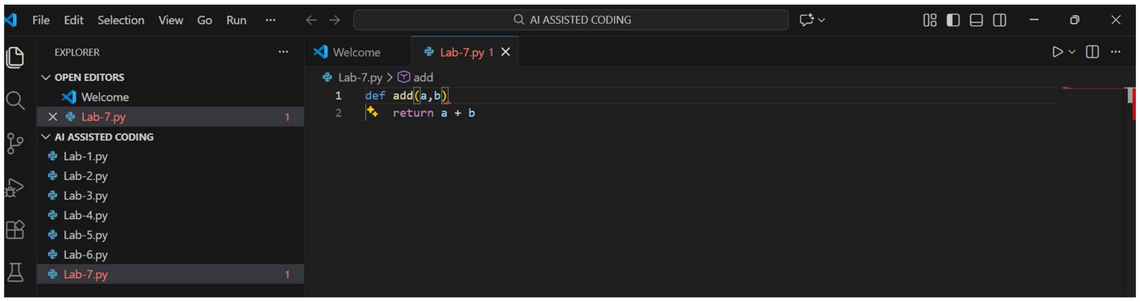
Gujja Pranitha

2303A52171

Batch 41

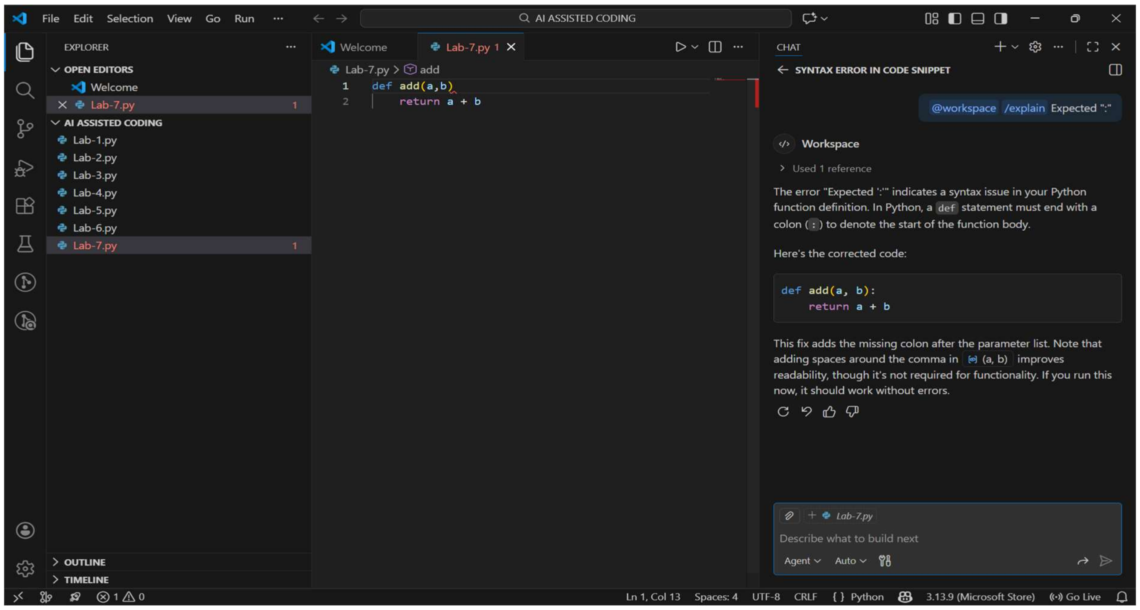
Task 1: Fixing Syntax Errors

Fix:



```
def add(a,b)
    return a + b
```

Review:



```
def add(a,b)
    return a + b
```

SYNTAX ERROR IN CODE SNIPPET

@workspace /explain Expected ":"

Workspace

Used 1 reference

The error "Expected ":"" indicates a syntax issue in your Python function definition. In Python, a `def` statement must end with a colon (`:`) to denote the start of the function body.

Here's the corrected code:

```
def add(a, b):
    return a + b
```

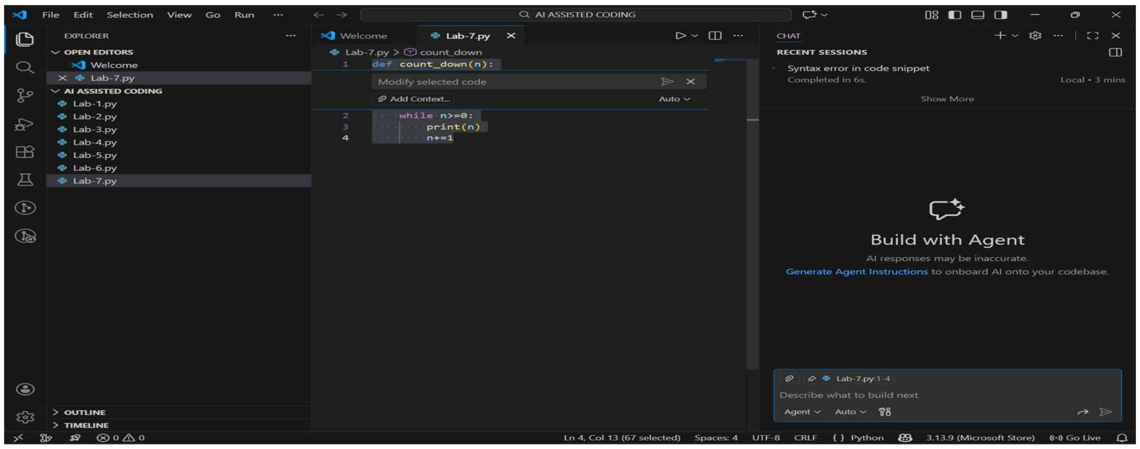
This fix adds the missing colon after the parameter list. Note that adding spaces around the comma in `(a, b)` improves readability, though it's not required for functionality. If you run this now, it should work without errors.

Justification:

The program had a small syntax error because the colon (:) was missing after the function definition. Python needs this colon to know where the function body starts. The AI tool detected the error, explained the reason clearly, and fixed it by adding the colon. After that, the function worked correctly without any errors.

Task 2: Debugging Logic Errors in Loop

Modify:



```
def count_down(n):
    while n>=0:
        print(n)
        n+=1
```

Modify selected code

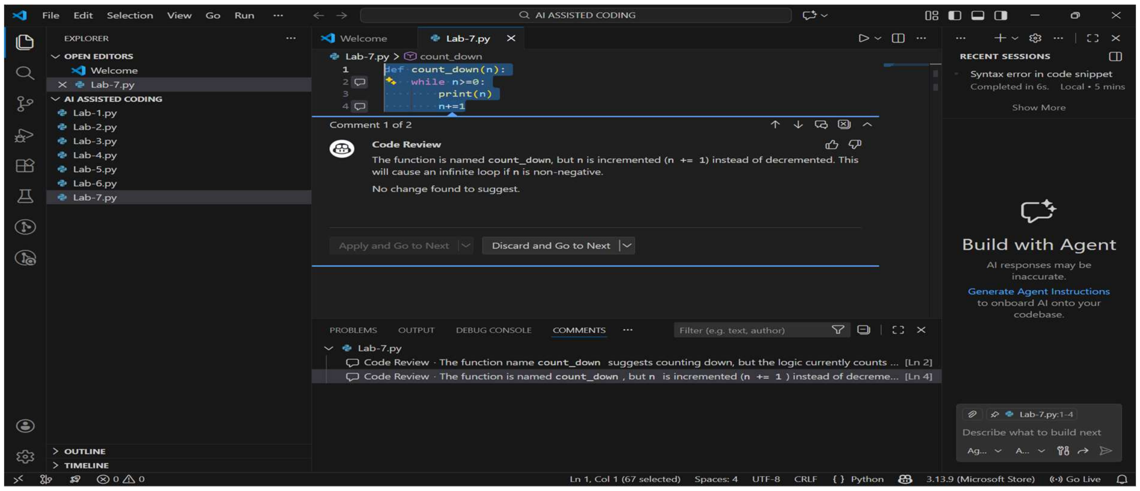
RECENT SESSIONS
Syntax error in code snippet
Completed in 6s. Local • 3 mins

Build with Agent
AI responses may be inaccurate.
Generate Agent Instructions to onboard AI onto your codebase.

Describe what to build next

Ln 4, Col 13 (67 selected) Spaces: 4 UTF-8 CRLF { } Python 3.13.9 (Microsoft Store)

Review:



```
def count_down(n):
    while n>=0:
        print(n)
        n+=1
```

Comment 1 of 2

Code Review

The function is named count_down, but n is incremented (n += 1) instead of decremented. This will cause an infinite loop if n is non-negative.

No change found to suggest.

Apply and Go to Next | Discard and Go to Next

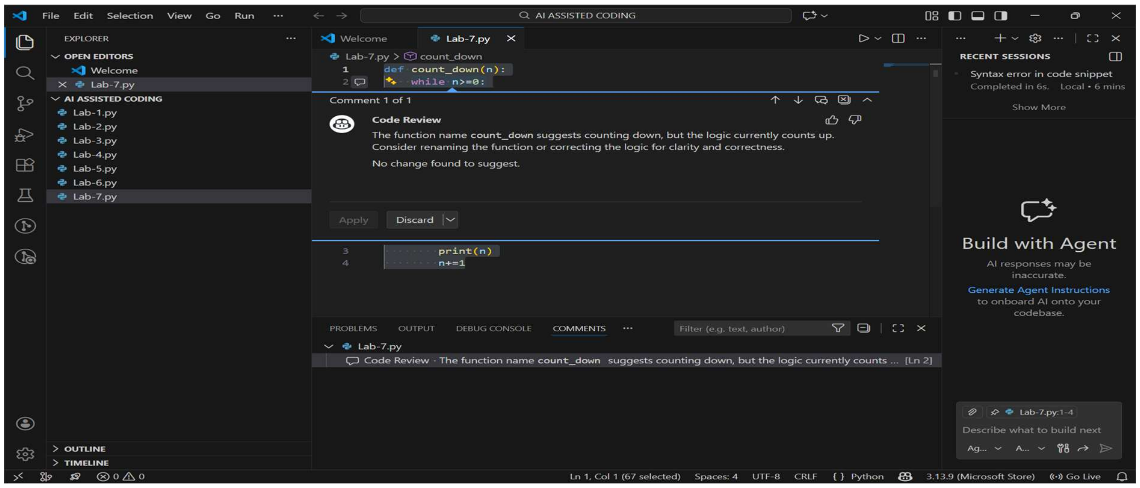
COMMENTS

Lab-7.py
- Code Review · The function name count_down suggests counting down, but the logic currently counts ... [Ln 2]
- Code Review · The function is named count_down , but n is incremented (n += 1) instead of decreme... [Ln 4]

RECENT SESSIONS
Syntax error in code snippet
Completed in 6s. Local • 5 mins

Build with Agent
AI responses may be inaccurate.
Generate Agent Instructions to onboard AI onto your codebase.

Ln 1, Col 1 (67 selected) Spaces: 4 UTF-8 CRLF { } Python 3.13.9 (Microsoft Store)



```
def count_down(n):
    while n>=0:
```

Comment 1 of 1

Code Review

The function name count_down suggests counting down, but the logic currently counts up. Consider renaming the function or correcting the logic for clarity and correctness.

No change found to suggest.

Apply | Discard

```
        print(n)
        n+=1
```

COMMENTS

Lab-7.py
- Code Review · The function name count_down suggests counting down, but the logic currently counts ... [Ln 2]

RECENT SESSIONS
Syntax error in code snippet
Completed in 6s. Local • 6 mins

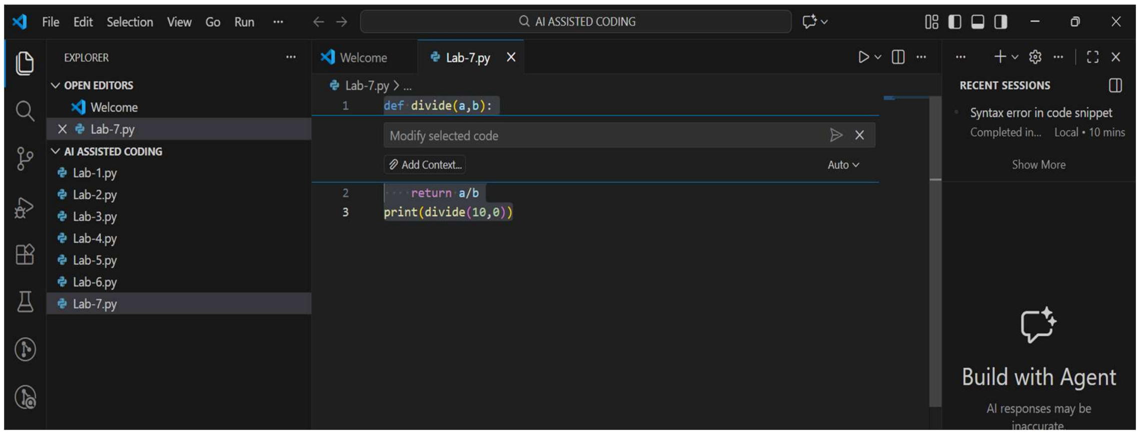
Build with Agent
AI responses may be inaccurate.
Generate Agent Instructions to onboard AI onto your codebase.

Ln 1, Col 1 (67 selected) Spaces: 4 UTF-8 CRLF { } Python 3.13.9 (Microsoft Store)

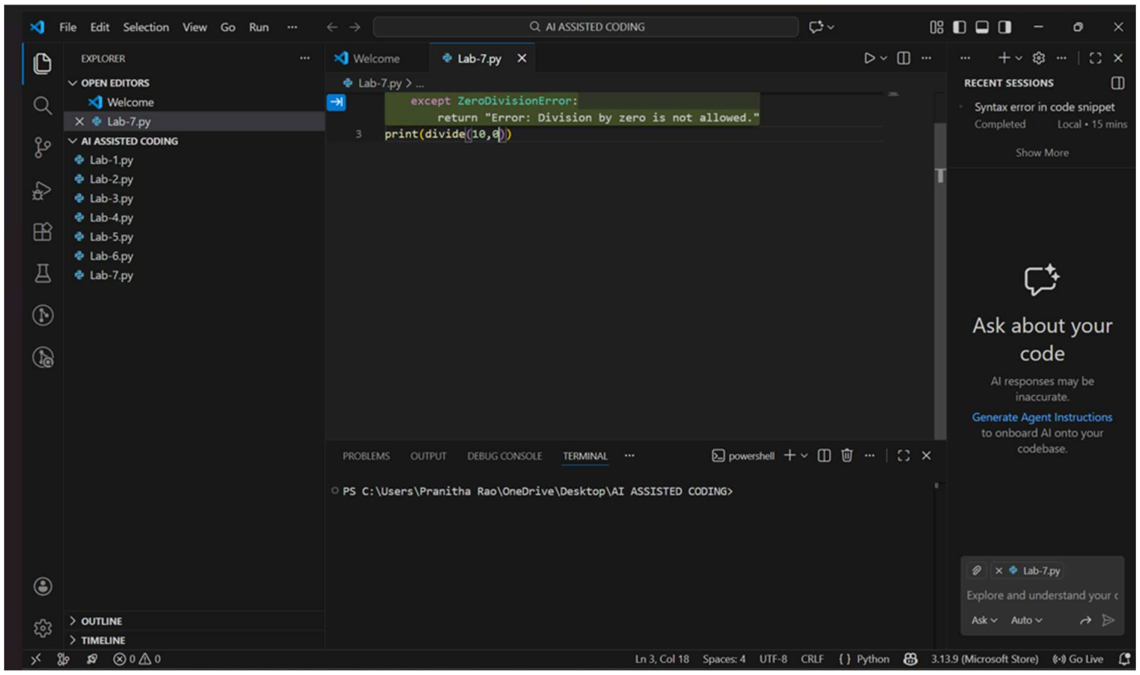
Justification: In this case, the loop was running infinitely because of a logical error in how the variable was updated. The function was named count_down, but instead of decreasing the value, it was increasing it (n += 1), so the loop condition never became false. The AI tool identified this mismatch between the function name and the logic and explained why it caused an infinite loop. It then fixed the issue by correcting the increment/decrement logic so the value changes in the right direction. After the fix, the loop worked as expected and stopped correctly.

Task 3: Handling Runtime Errors (Division by Zero)

Fix:



```
def divide(a,b):
    return a/b
print(divide(10,0))
```

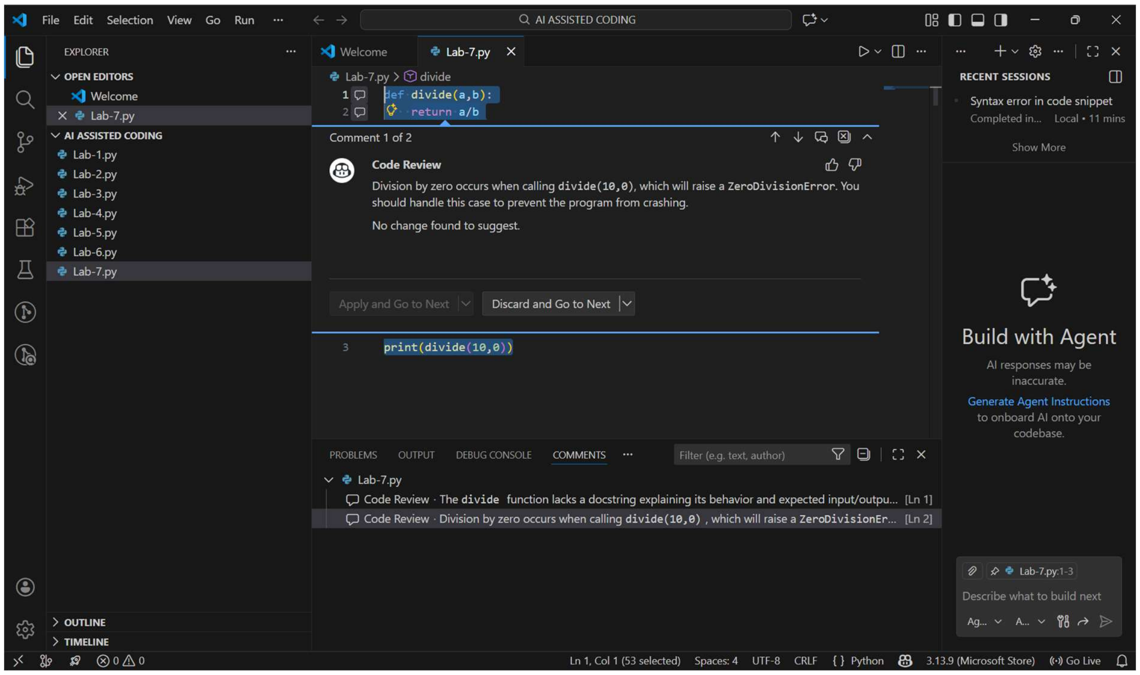


```
        except ZeroDivisionError:
            return "Error: Division by zero is not allowed."
print(divide(10,0))
```

```
PS C:\Users\Pranitha Rao\OneDrive\Desktop\AI ASSISTED CODING>
```

RECENT SESSIONS
Syntax error in code snippet
Ask about your code
AI responses may be inaccurate.
Generate Agent Instructions to onboard AI onto your codebase.
Ln 3, Col 18 Spaces: 4 UTF-8 CRLF Python 3.13.9 (Microsoft Store)

Review:


File Edit Selection View Go Run
AI ASSISTED CODING
EXPLORER
OPEN EDITORS
Welcome
Lab-7.py
AI ASSISTED CODING
Lab-1.py
Lab-2.py
Lab-3.py
Lab-4.py
Lab-5.py
Lab-6.py
Lab-7.py
OUTLINE
TIMELINE
Lab-7.py > divide
Comment 1 of 2
Apply and Go to Next
Discard and Go to Next
PROBLEMS OUTPUT DEBUG CONSOLE COMMENTS
Filter (e.g. text, author)
RECENT SESSIONS
Syntax error in code snippet
Completed in... Local • 11 mins
Show More
Build with Agent
AI responses may be inaccurate.
Generate Agent Instructions to onboard AI onto your codebase.
Lab-7.py:1-3
Describe what to build next
Ln 1, Col 1 (53 selected) Spaces: 4 UTF-8 CRLF { } Python 3.13.9 (Microsoft Store) Go Live

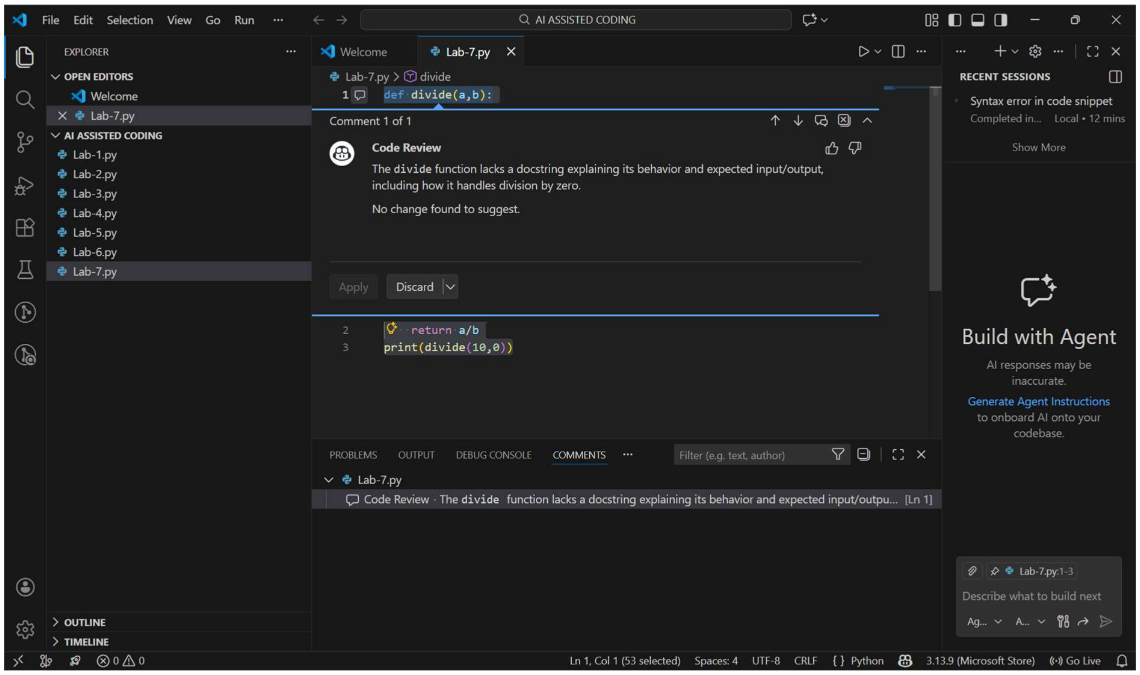
```
def divide(a,b):
    return a/b
print(divide(10,0))
```

Code Review

Division by zero occurs when calling `divide(10,0)`, which will raise a `ZeroDivisionError`. You should handle this case to prevent the program from crashing.

No change found to suggest.

Lab-7.py
- Code Review · The `divide` function lacks a docstring explaining its behavior and expected input/outpu... [Ln 1]
- Code Review · Division by zero occurs when calling `divide(10,0)`, which will raise a `ZeroDivisionEr`... [Ln 2]


File Edit Selection View Go Run
AI ASSISTED CODING
EXPLORER
OPEN EDITORS
Welcome
Lab-7.py
AI ASSISTED CODING
Lab-1.py
Lab-2.py
Lab-3.py
Lab-4.py
Lab-5.py
Lab-6.py
Lab-7.py
OUTLINE
TIMELINE
Lab-7.py > divide
Comment 1 of 1
Apply
Discard
PROBLEMS OUTPUT DEBUG CONSOLE COMMENTS
Filter (e.g. text, author)
RECENT SESSIONS
Syntax error in code snippet
Completed in... Local • 12 mins
Show More
Build with Agent
AI responses may be inaccurate.
Generate Agent Instructions to onboard AI onto your codebase.
Lab-7.py:1-3
Describe what to build next
Ln 1, Col 1 (53 selected) Spaces: 4 UTF-8 CRLF { } Python 3.13.9 (Microsoft Store) Go Live

```
def divide(a,b):
    return a/b
print(divide(10,0))
```

Code Review

The `divide` function lacks a docstring explaining its behavior and expected input/output, including how it handles division by zero.

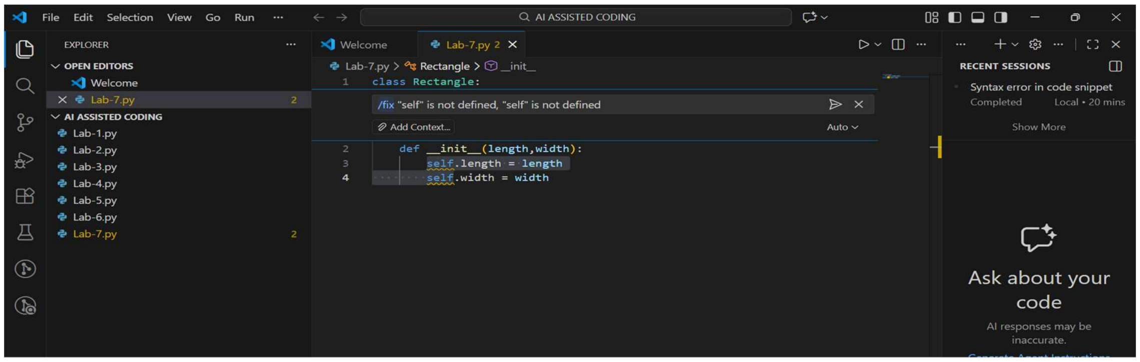
No change found to suggest.

Lab-7.py
- Code Review · The `divide` function lacks a docstring explaining its behavior and expected input/outpu... [Ln 1]

Justification: In this scenario, the division function caused a runtime error because it tried to divide a number by zero, which is not allowed in Python. The AI tool identified this issue when the function was executed and explained that it would raise a ZeroDivisionError. To fix this, the AI added a try-except block to handle the error safely. With this change, the program no longer crashes and instead shows a clear message when division by zero occurs, making the code safer and more reliable.

Task 4: Debugging Class Definition Errors

Fix:



```
class Rectangle:
```

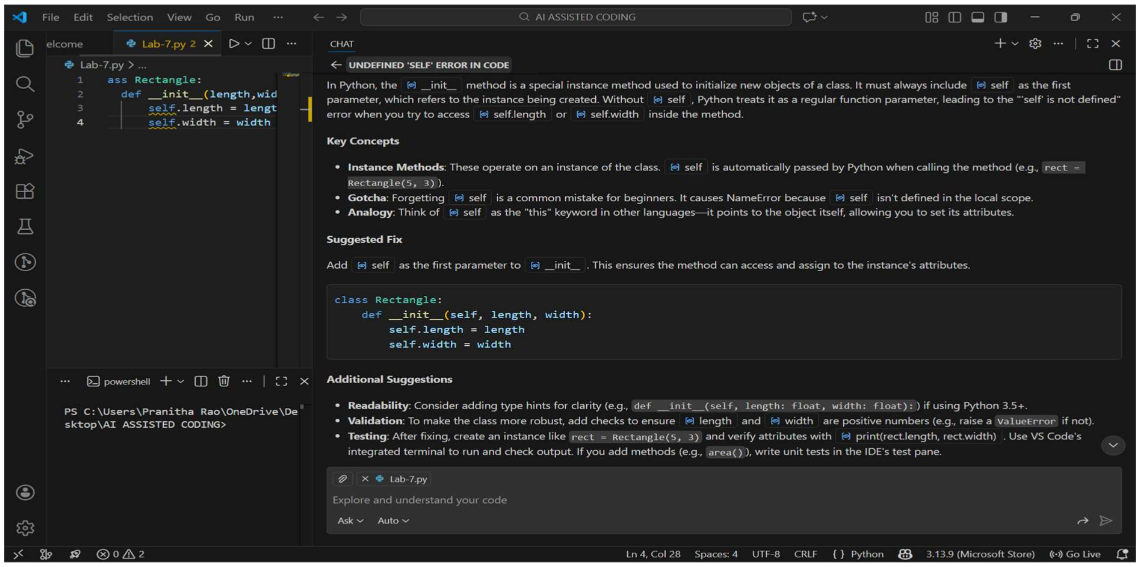
/fix "self" is not defined, "self" is not defined

```
    def __init__(length,width):
        self.length = length
        self.width = width
```

RECENT SESSIONS

Syntax error in code snippet — Completed — Local • 20 mins

Explain:



```
class Rectangle:
    def __init__(length,wid
        self.length = lengt
        self.width = width
```

```
PS C:\Users\Pranitha Rao\OneDrive\Desktop\AI ASSISTED CODING>
```

CHAT

UNDEFINED 'SELF' ERROR IN CODE

In Python, the __init__ method is a special instance method used to initialize new objects of a class. It must always include self as the first parameter, which refers to the instance being created. Without self , Python treats it as a regular function parameter, leading to the "'self' is not defined" error when you try to access self.length or self.width inside the method.

Key Concepts

- **Instance Methods**: These operate on an instance of the class. self is automatically passed by Python when calling the method (e.g., `rect = Rectangle(5, 3)`).
- **Gotcha**: Forgetting self is a common mistake for beginners. It causes NameError because self isn't defined in the local scope.
- **Analogy**: Think of self as the "this" keyword in other languages—it points to the object itself, allowing you to set its attributes.

Suggested Fix

Add self as the first parameter to __init__ . This ensures the method can access and assign to the instance's attributes.

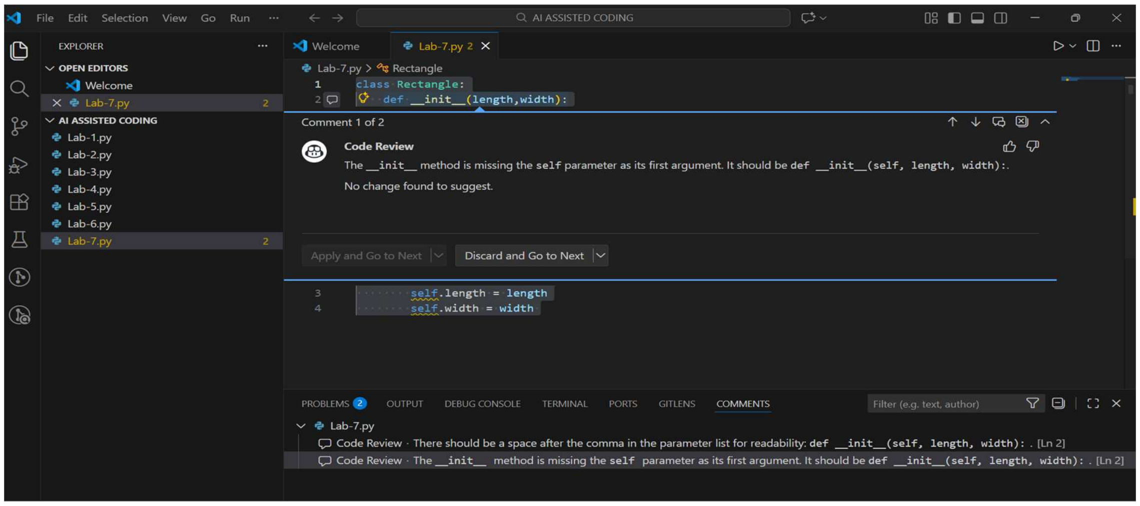
```
class Rectangle:
    def __init__(self, length, width):
        self.length = length
        self.width = width
```

Additional Suggestions

- **Readability**: Consider adding type hints for clarity (e.g., `def __init__(self, length: float, width: float):`) if using Python 3.5+.
- **Validation**: To make the class more robust, add checks to ensure length and width are positive numbers (e.g., raise a `ValueError` if not).
- **Testing**: After fixing, create an instance like `rect = Rectangle(5, 3)` and verify attributes with print(rect.length, rect.width) . Use VS Code's integrated terminal to run and check output. If you add methods (e.g., `area()`), write unit tests in the IDE's test pane.

Ln 4, Col 28   Spaces: 4   UTF-8   CRLF   { } Python   3.13.9 (Microsoft Store)   Go Live

Review:



```
class Rectangle:
    def __init__(length,width):
```

Comment 1 of 2

Code Review

The __init__ method is missing the self parameter as its first argument. It should be def __init__(self, length, width):.

No change found to suggest.

Apply and Go to Next | Discard and Go to Next

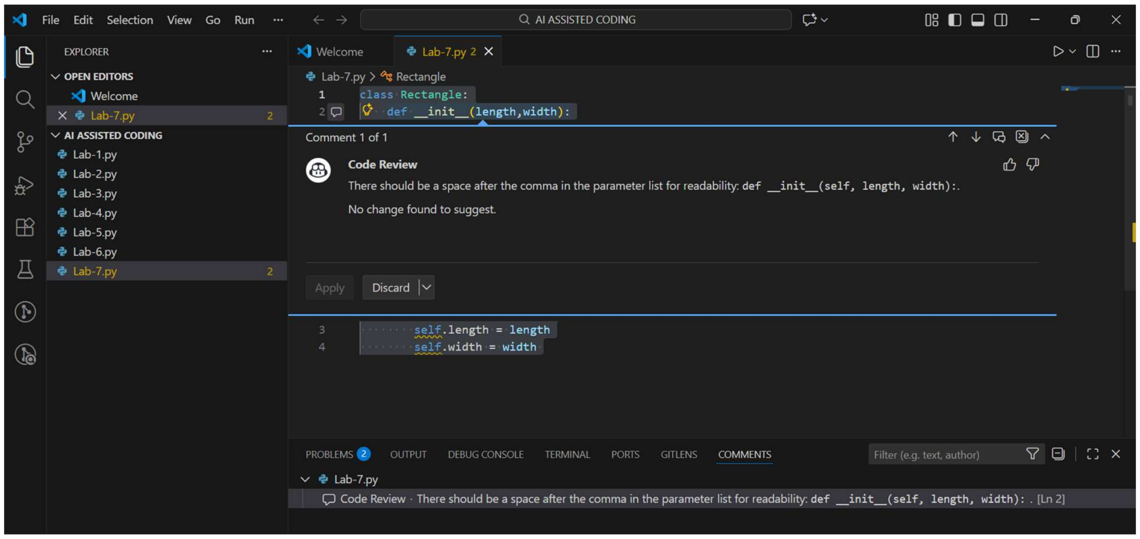
```
        self.length = length
        self.width = width
```

PROBLEMS 2 | OUTPUT | DEBUG CONSOLE | TERMINAL | PORTS | GITLENS | COMMENTS

Lab-7.py

Code Review · There should be a space after the comma in the parameter list for readability: def __init__(self, length, width): . [Ln 2]

Code Review · The __init__ method is missing the self parameter as its first argument. It should be def __init__(self, length, width): . [Ln 2]



```python
class Rectangle:
    def __init__(length,width):
        self.length = length
        self.width = width
```

Comment 1 of 1

Code Review

There should be a space after the comma in the parameter list for readability: `def __init__(self, length, width):`.

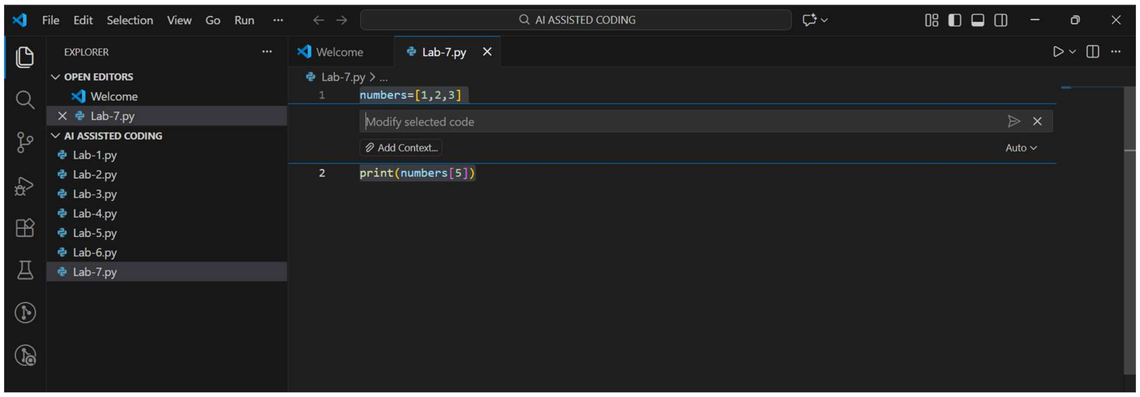
No change found to suggest.

Code Review · There should be a space after the comma in the parameter list for readability: `def __init__(self, length, width):` . [Ln 2]

Justification: In this scenario, the class constructor had an error because the self parameter was missing in the __init__() method. The AI tool detected this problem and explained that self is required to refer to the current object and store instance variables. It then corrected the class by adding self to the constructor and using it properly for attributes. After the fix, the class worked correctly, showing how self is essential in Python class definitions.

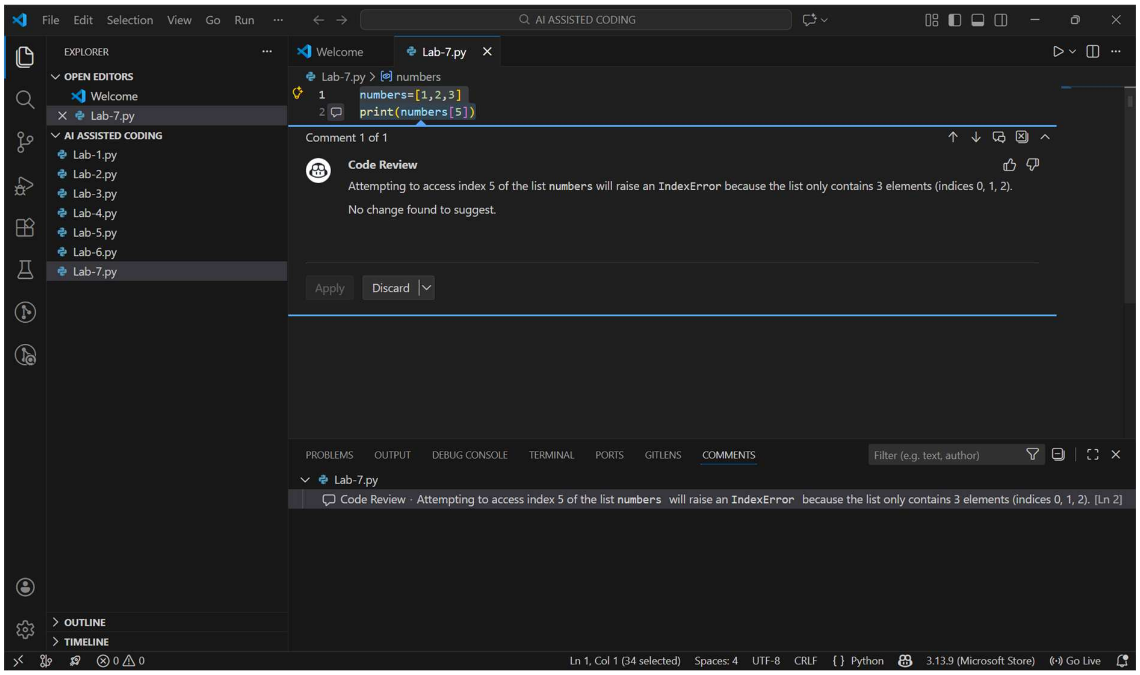
Task 5: Resolving Index Errors in Lists

Modify:



```python
numbers=[1,2,3]
print(numbers[5])
```

Review:



```python
numbers=[1,2,3]
print(numbers[5])
```

Comment 1 of 1

Code Review

Attempting to access index 5 of the list `numbers` will raise an `IndexError` because the list only contains 3 elements (indices 0, 1, 2).

No change found to suggest.

Lab-7.py

Code Review · Attempting to access index 5 of the list `numbers` will raise an `IndexError` because the list only contains 3 elements (indices 0, 1, 2). [Ln 2]

Justification: In this scenario, the program crashed because it tried to access a list index that does not exist, which caused an IndexError. The AI tool identified this problem and explained that list indices must be within the valid range. To fix it, the AI suggested safe access methods such as checking the list length before accessing an index or using a try-except block to handle the error. After applying these changes, the program ran safely without crashing.