AI-ASSISTED CODING (24CS002PC215)

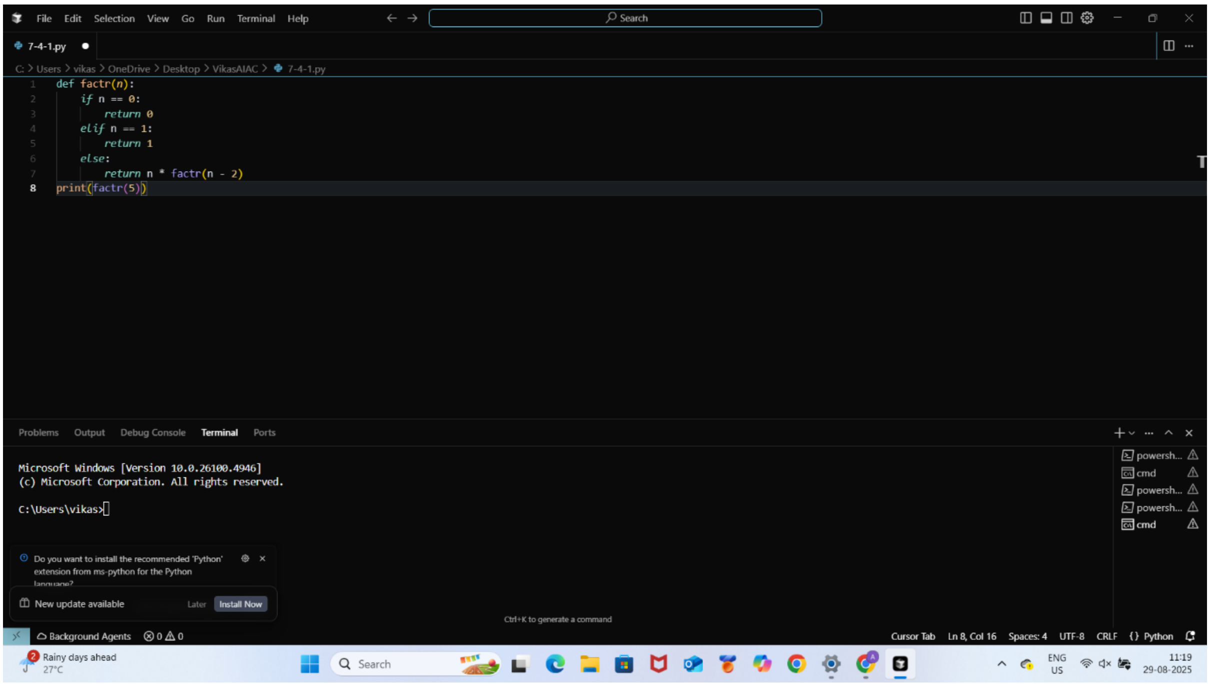
LAB ASSIGNMENT-7.4

Student Name: Amgoth Vikas Nayak

HALL-TICKET NO: 2403A51410

TASK 1:

Buggy Python function that calculates the factorial of a number using recursion.



File
Edit
Selection
View
Go
Run
Terminal
Help
Search
7-4-1.py
C: > Users > vikas > OneDrive > Desktop > VikasAIAC > 7-4-1.py

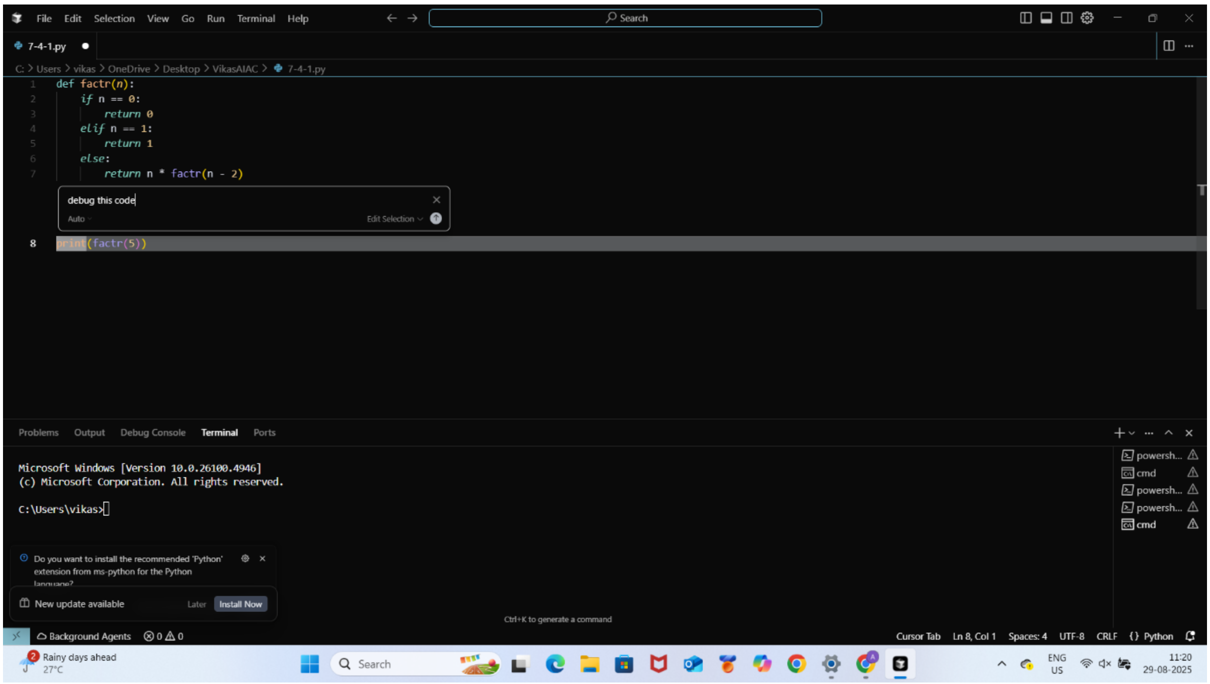
```python
def factr(n):
    if n == 0:
        return 0
    elif n == 1:
        return 1
    else:
        return n * factr(n - 2)
print(factr(5))
```

```
Microsoft Windows [Version 10.0.26100.4946]
(c) Microsoft Corporation. All rights reserved.

C:\Users\vikas>
```

Do you want to install the recommended 'Python' extension from ms-python for the Python language?

New update available

Cursor AI to detect and fix the logical or syntax errors.



```
def factr(n):
    if n == 0:
        return 0
    elif n == 1:
        return 1
    else:
        return n * factr(n - 2)
print(factr(5))
```

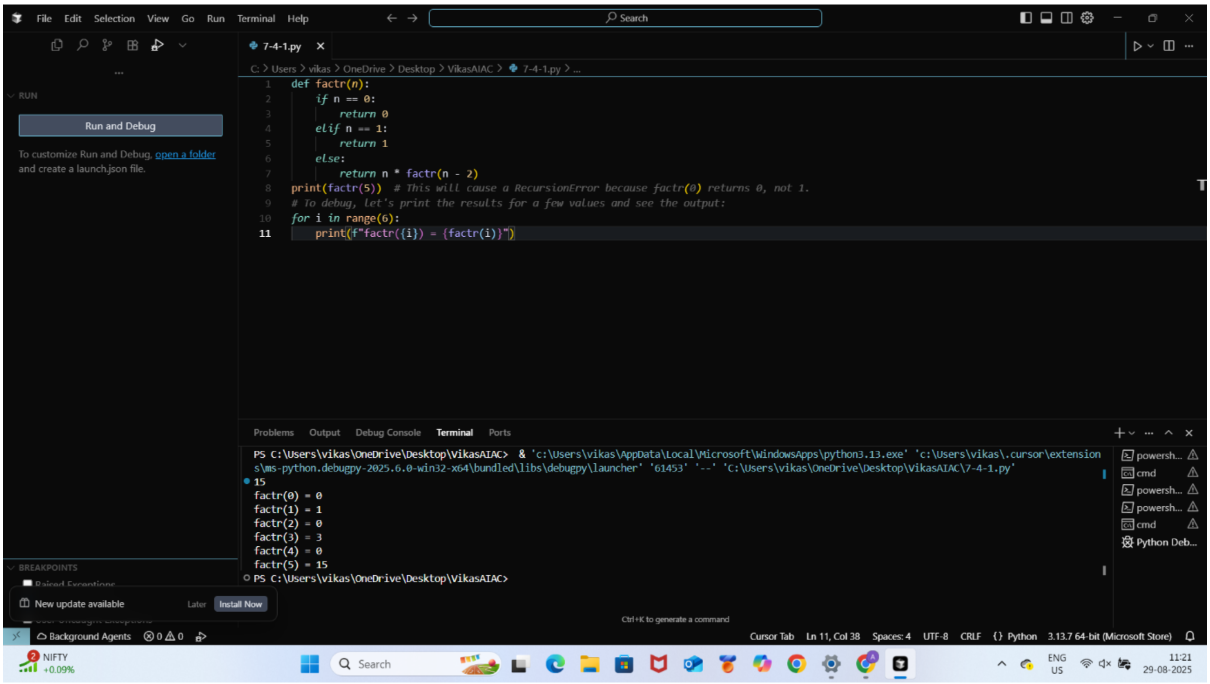
debug this code

```
Microsoft Windows [Version 10.0.26100.4946]
(c) Microsoft Corporation. All rights reserved.

C:\Users\vikas>
```

Do you want to install the recommended 'Python' extension from ms-python for the Python language?

New update available

Cursor AI correctly identifies missing base condition or incorrect recursive call and suggests a functional factorial implementation.



```
def factr(n):
    if n == 0:
        return 0
    elif n == 1:
        return 1
    else:
        return n * factr(n - 2)
print(factr(5))  # This will cause a RecursionError because factr(0) returns 0, not 1.
# To debug, let's print the results for a few values and see the output:
for i in range(6):
    print(f"factr({i}) = {factr(i)}")
```

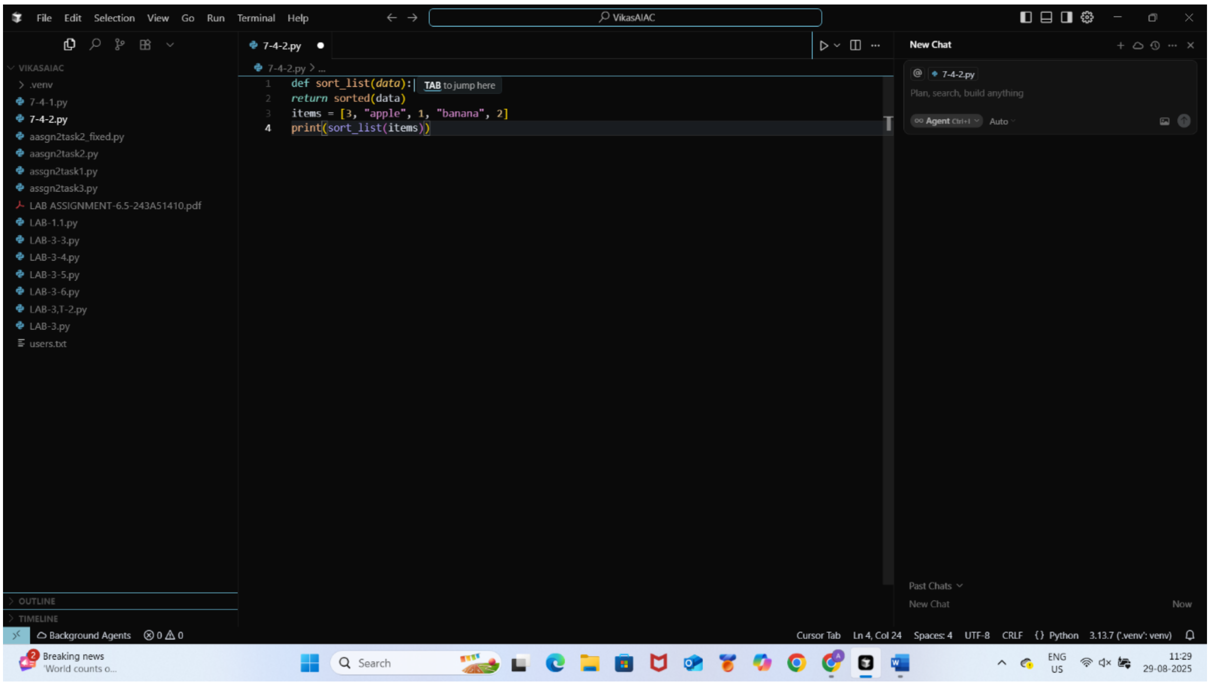
```
PS C:\Users\vikas\OneDrive\Desktop\VikasAIAC>  & 'c:\Users\vikas\AppData\Local\Microsoft\WindowsApps\python3.13.exe' 'c:\Users\vikas\.cursor\extension
s\ms-python.debugpy-2025.6.0-win32-x64\bundled\libs\debugpy\launcher' '61453' '--' 'C:\Users\vikas\OneDrive\Desktop\VikasAIAC\7-4-1.py'
15
factr(0) = 0
factr(1) = 1
factr(2) = 0
factr(3) = 3
factr(4) = 0
factr(5) = 15
PS C:\Users\vikas\OneDrive\Desktop\VikasAIAC>
```

To customize Run and Debug, open a folder and create a launch.json file.

New update available

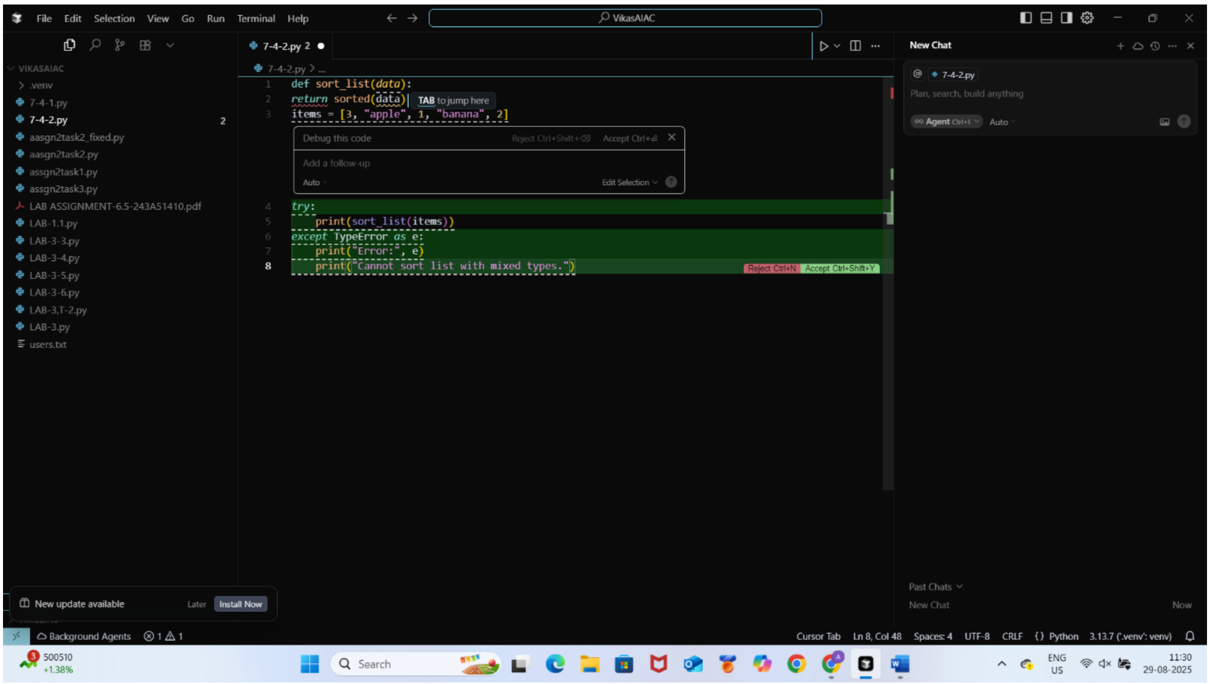
TASK 2:

a list sorting function that fails due to a type error (e.g., sorting list with mixed integers and strings).



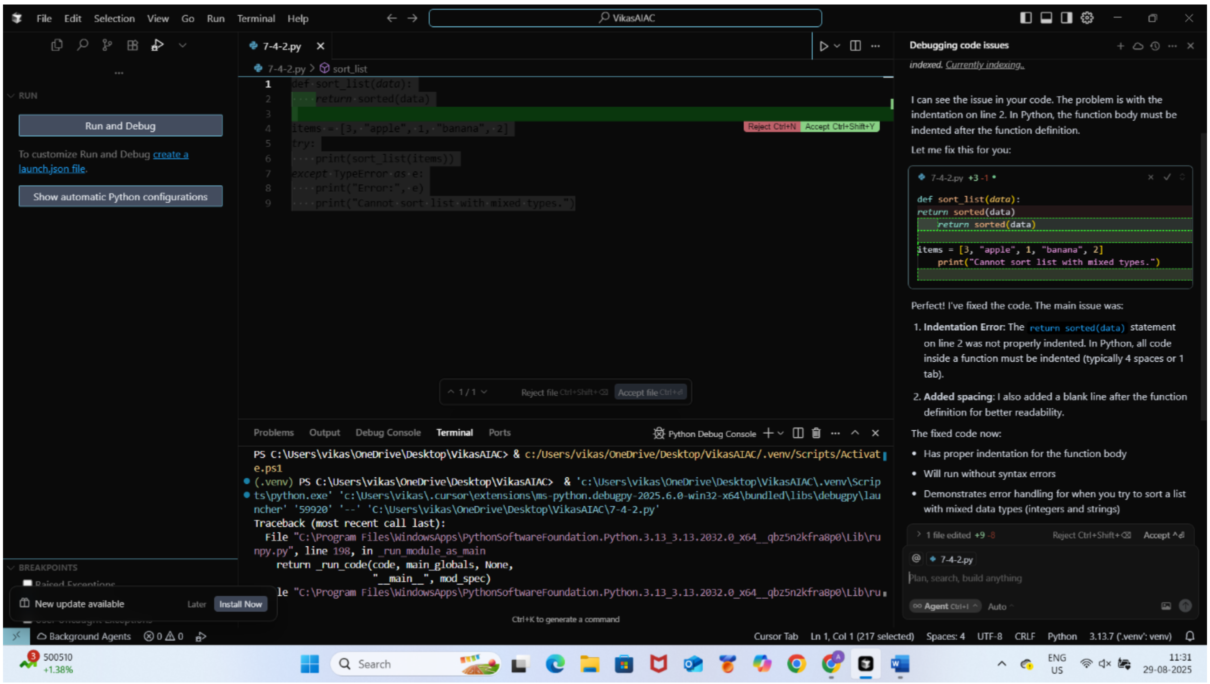
```python
def sort_list(data):
return sorted(data)
items = [3, "apple", 1, "banana", 2]
print(sort_list(items))
```

Prompt AI to detect the issue and fix the code for consistent sorting.



Debug this code

```python
def sort_list(data):
return sorted(data)
items = [3, "apple", 1, "banana", 2]
try:
    print(sort_list(items))
except TypeError as e:
    print("Error:", e)
    print("Cannot sort list with mixed types.")
```



To customize Run and Debug create a launch.json file.

```python
def sort_list(data):
    return sorted(data)

items = [3, "apple", 1, "banana", 2]
try:
    print(sort_list(items))
except TypeError as e:
    print("Error:", e)
    print("Cannot sort list with mixed types.")
```

```
PS C:\Users\vikas\OneDrive\Desktop\VikasAIAC> & c:/Users/vikas/OneDrive/Desktop/VikasAIAC/.venv/Scripts/Activat
e.ps1
(.venv) PS C:\Users\vikas\OneDrive\Desktop\VikasAIAC>  & 'c:\Users\vikas\OneDrive\Desktop\VikasAIAC\.venv\Scrip
ts\python.exe' 'c:\Users\vikas\.cursor\extensions\ms-python.debugpy-2025.6.0-win32-x64\bundled\libs\debugpy\lau
ncher' '59920' '--' 'C:\Users\vikas\OneDrive\Desktop\VikasAIAC\7-4-2.py'
Traceback (most recent call last):
  File "C:\Program Files\WindowsApps\PythonSoftwareFoundation.Python.3.13_3.13.2032.0_x64__qbz5n2kfra8p0\Lib\ru
npy.py", line 198, in _run_module_as_main
    return _run_code(code, main_globals, None,
                     "__main__", mod_spec)
le "C:\Program Files\WindowsApps\PythonSoftwareFoundation.Python.3.13_3.13.2032.0_x64__qbz5n2kfra8p0\Lib\ru
```

indexed. Currently indexing..

I can see the issue in your code. The problem is with the indentation on line 2. In Python, the function body must be indented after the function definition.

Let me fix this for you:

7-4-2.py +3 -1

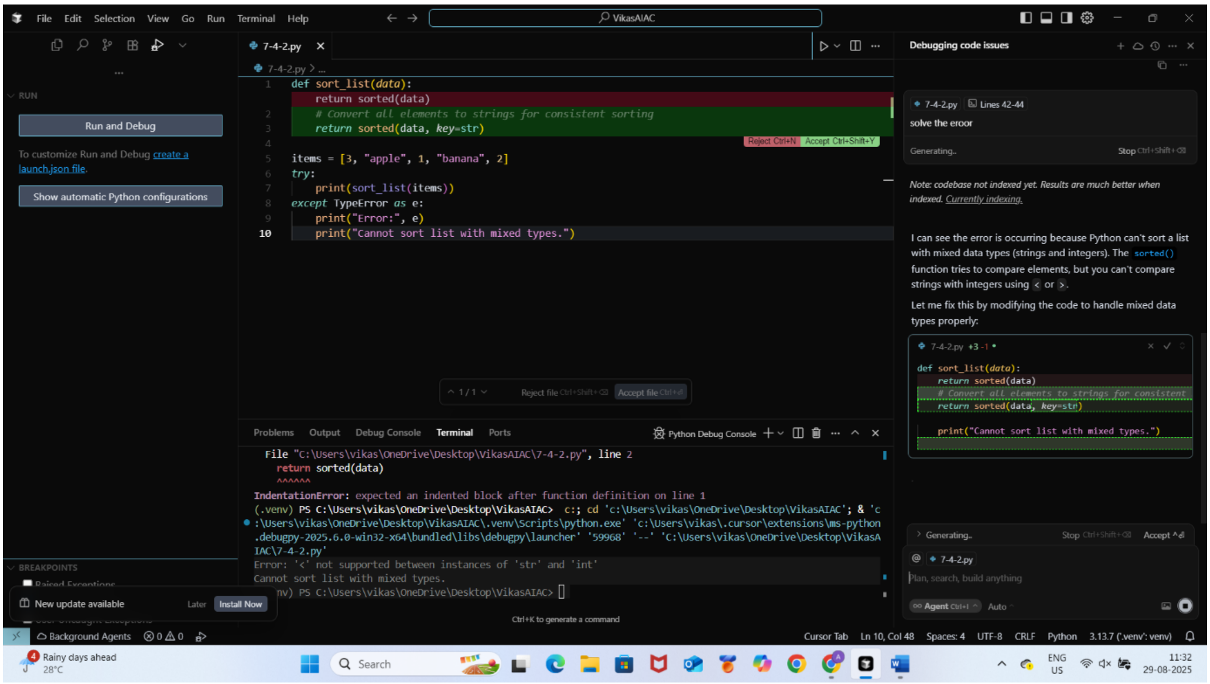
```python
def sort_list(data):
return sorted(data)
    return sorted(data)

items = [3, "apple", 1, "banana", 2]
    print("Cannot sort list with mixed types.")
```

Perfect! I've fixed the code. The main issue was:

1. **Indentation Error:** The `return sorted(data)` statement on line 2 was not properly indented. In Python, all code inside a function must be indented (typically 4 spaces or 1 tab).
2. **Added spacing:** I also added a blank line after the function definition for better readability.

The fixed code now:

- Has proper indentation for the function body
- Will run without syntax errors
- Demonstrates error handling for when you try to sort a list with mixed data types (integers and strings)



To customize Run and Debug create a launch.json file.

```python
def sort_list(data):
    return sorted(data)
    # Convert all elements to strings for consistent sorting
    return sorted(data, key=str)

items = [3, "apple", 1, "banana", 2]
try:
    print(sort_list(items))
except TypeError as e:
    print("Error:", e)
    print("Cannot sort list with mixed types.")
```

```
  File "C:\Users\vikas\OneDrive\Desktop\VikasAIAC\7-4-2.py", line 2
    return sorted(data)
    ^^^^^^
IndentationError: expected an indented block after function definition on line 1
(.venv) PS C:\Users\vikas\OneDrive\Desktop\VikasAIAC>  c:; cd 'c:\Users\vikas\OneDrive\Desktop\VikasAIAC'; & 'c
:\Users\vikas\OneDrive\Desktop\VikasAIAC\.venv\Scripts\python.exe' 'c:\Users\vikas\.cursor\extensions\ms-python
.debugpy-2025.6.0-win32-x64\bundled\libs\debugpy\launcher' '59968' '--' 'C:\Users\vikas\OneDrive\Desktop\VikasA
IAC\7-4-2.py'
Error: '<' not supported between instances of 'str' and 'int'
Cannot sort list with mixed types.
nv) PS C:\Users\vikas\OneDrive\Desktop\VikasAIAC>
```

solve the eroor

Note: codebase not indexed yet. Results are much better when indexed. Currently indexing.

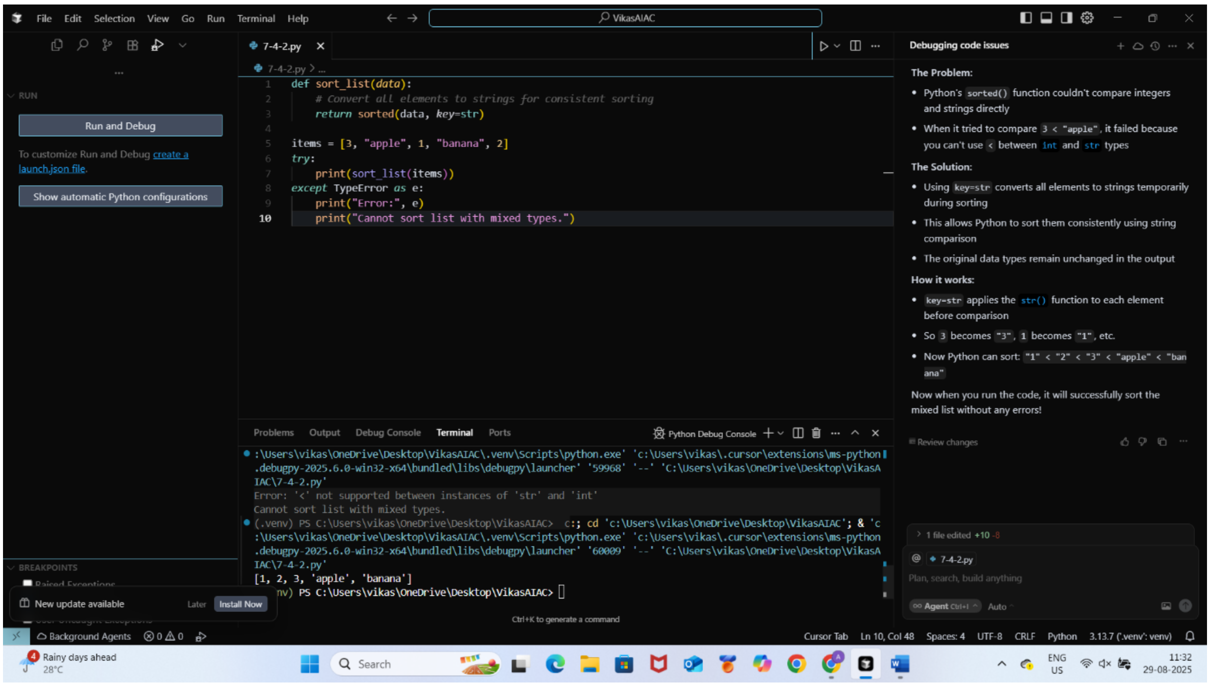
I can see the error is occurring because Python can't sort a list with mixed data types (strings and integers). The `sorted()` function tries to compare elements, but you can't compare strings with integers using `<` or `>`.

Let me fix this by modifying the code to handle mixed data types properly:

7-4-2.py +3 -1

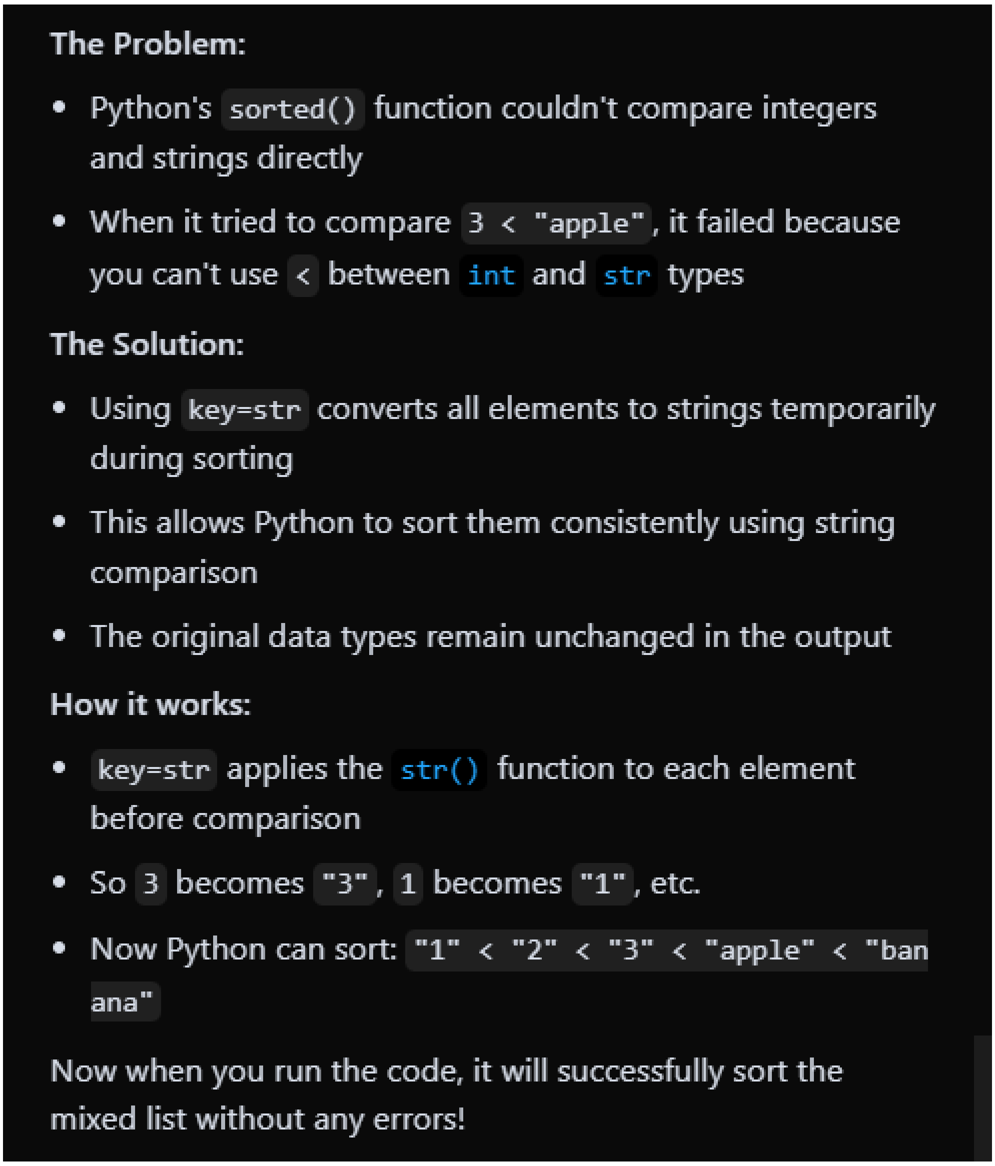
```python
def sort_list(data):
    return sorted(data)
    # Convert all elements to strings for consistent
    return sorted(data, key=str)

    print("Cannot sort list with mixed types.")
```

AI detects the type inconsistency and either filters or converts list elements, ensuring successful
sorting without a crash.



```python
def sort_list(data):
    # Convert all elements to strings for consistent sorting
    return sorted(data, key=str)

items = [3, "apple", 1, "banana", 2]
try:
    print(sort_list(items))
except TypeError as e:
    print("Error:", e)
    print("Cannot sort list with mixed types.")
```

```
:\Users\vikas\OneDrive\Desktop\VikasAIAC\.venv\Scripts\python.exe' 'c:\Users\vikas\.cursor\extensions\ms-python
.debugpy-2025.6.0-win32-x64\bundled\libs\debugpy\launcher' '59968' '--' 'C:\Users\vikas\OneDrive\Desktop\VikasA
IAC\7-4-2.py'
Error: '<' not supported between instances of 'str' and 'int'
Cannot sort list with mixed types.
(.venv) PS C:\Users\vikas\OneDrive\Desktop\VikasAIAC>  c:; cd 'c:\Users\vikas\OneDrive\Desktop\VikasAIAC'; & 'c
:\Users\vikas\OneDrive\Desktop\VikasAIAC\.venv\Scripts\python.exe' 'c:\Users\vikas\.cursor\extensions\ms-python
.debugpy-2025.6.0-win32-x64\bundled\libs\debugpy\launcher' '60009' '--' 'C:\Users\vikas\OneDrive\Desktop\VikasA
IAC\7-4-2.py'
[1, 2, 3, 'apple', 'banana']
nv) PS C:\Users\vikas\OneDrive\Desktop\VikasAIAC>
```



The Problem:

- Python's `sorted()` function couldn't compare integers and strings directly
- When it tried to compare `3 < "apple"`, it failed because you can't use `<` between `int` and `str` types

The Solution:

- Using `key=str` converts all elements to strings temporarily during sorting
- This allows Python to sort them consistently using string comparison
- The original data types remain unchanged in the output

How it works:

- `key=str` applies the `str()` function to each element before comparison
- So `3` becomes `"3"`, `1` becomes `"1"`, etc.
- Now Python can sort: `"1" < "2" < "3" < "apple" < "banana"`

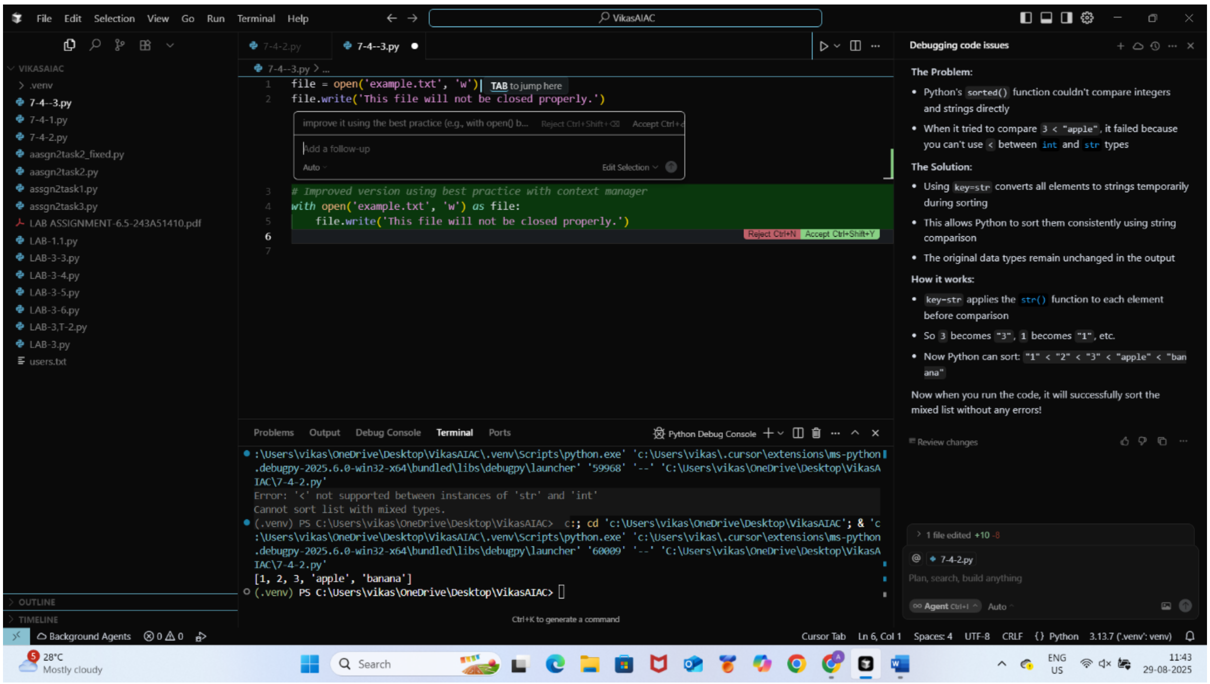
Now when you run the code, it will successfully sort the mixed list without any errors!

TASK 3:

Python snippet for file handling that opens a file but forgets to close it.

Ask Copilot or

Cursor AI to improve it using the best practice (e.g., with open() block)



```python
file = open('example.txt', 'w')
file.write('This file will not be closed properly.')
```

improve it using the best practice (e.g., with open() b...

```python
# Improved version using best practice with context manager
with open('example.txt', 'w') as file:
    file.write('This file will not be closed properly.')
```

The Problem:

- Python's `sorted()` function couldn't compare integers and strings directly
- When it tried to compare `3 < "apple"`, it failed because you can't use `<` between `int` and `str` types

The Solution:

- Using `key=str` converts all elements to strings temporarily during sorting
- This allows Python to sort them consistently using string comparison
- The original data types remain unchanged in the output

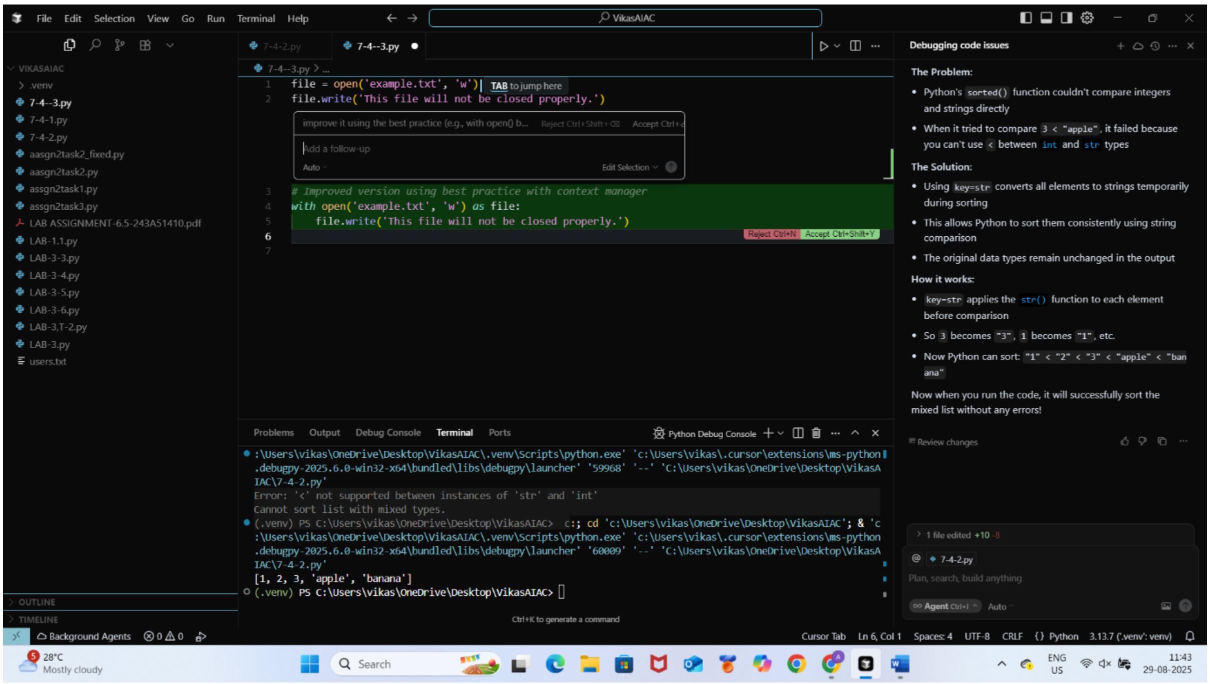
How it works:

- `key-str` applies the `str()` function to each element before comparison
- So `3` becomes `"3"`, `1` becomes `"1"`, etc.
- Now Python can sort: `"1" < "2" < "3" < "apple" < "banana"`

Now when you run the code, it will successfully sort the mixed list without any errors!

```
:\Users\vikas\OneDrive\Desktop\VikasAIAC\.venv\Scripts\python.exe' 'c:\Users\vikas\.cursor\extensions\ms-python
.debugpy-2025.6.0-win32-x64\bundled\libs\debugpy\launcher' '59968' '--' 'C:\Users\vikas\OneDrive\Desktop\VikasA
IAC\7-4-2.py'
Error: '<' not supported between instances of 'str' and 'int'
Cannot sort list with mixed types.
(.venv) PS C:\Users\vikas\OneDrive\Desktop\VikasAIAC>  c:; cd 'c:\Users\vikas\OneDrive\Desktop\VikasAIAC'; & 'c
:\Users\vikas\OneDrive\Desktop\VikasAIAC\.venv\Scripts\python.exe' 'c:\Users\vikas\.cursor\extensions\ms-python
.debugpy-2025.6.0-win32-x64\bundled\libs\debugpy\launcher' '60009' '--' 'C:\Users\vikas\OneDrive\Desktop\VikasA
IAC\7-4-2.py'
[1, 2, 3, 'apple', 'banana']
(.venv) PS C:\Users\vikas\OneDrive\Desktop\VikasAIAC> 
```

AI refactors the code to use a context manager, preventing resource leakage and runtime warnings



```
file = open('example.txt', 'w')
file.write('This file will not be closed properly.')
# Improved version using best practice with context manager
with open('example.txt', 'w') as file:
    file.write('This file will not be closed properly.')
```

Debugging code issues

The Problem:

- Python's `sorted()` function couldn't compare integers and strings directly
- When it tried to compare `3 < "apple"`, it failed because you can't use `<` between `int` and `str` types

The Solution:

- Using `key=str` converts all elements to strings temporarily during sorting
- This allows Python to sort them consistently using string comparison
- The original data types remain unchanged in the output

How it works:

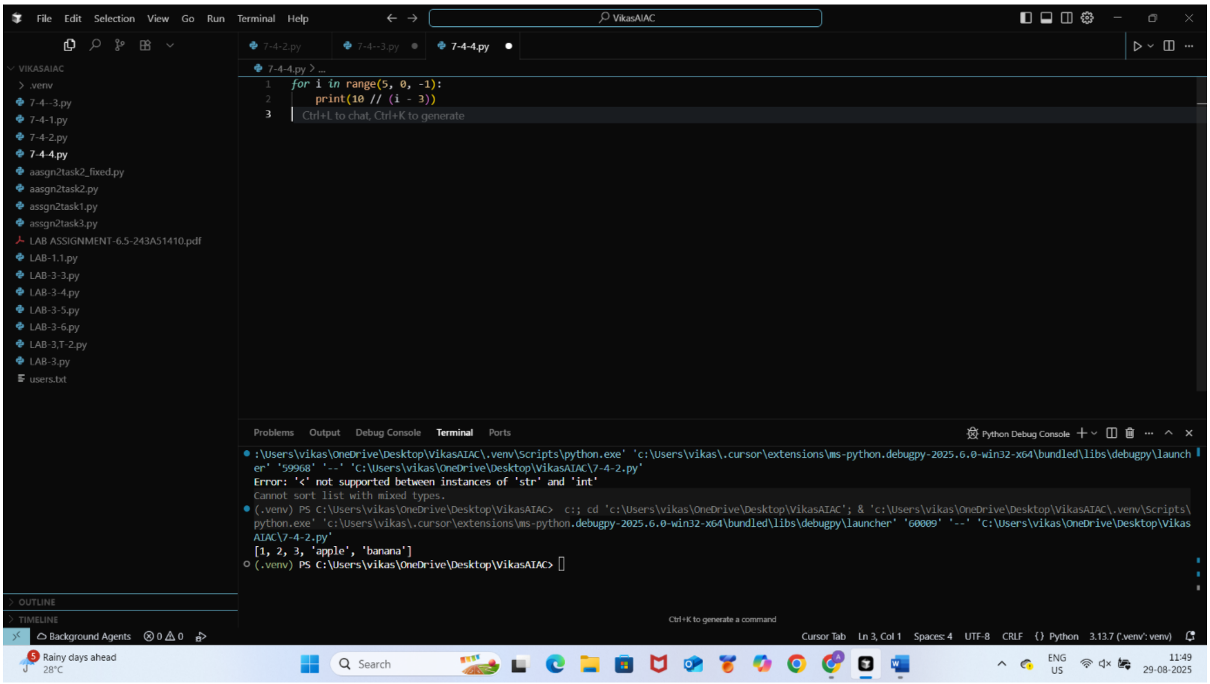
- `key-str` applies the `str()` function to each element before comparison
- So `3` becomes `"3"`, `1` becomes `"1"`, etc.
- Now Python can sort: `"1" < "2" < "3" < "apple" < "banana"`

Now when you run the code, it will successfully sort the mixed list without any errors!

```
:\Users\vikas\OneDrive\Desktop\VikasAIAC\.venv\Scripts\python.exe' 'c:\Users\vikas\.cursor\extensions\ms-python
.debugpy-2025.6.0-win32-x64\bundled\libs\debugpy\launcher' '59968' '--' 'C:\Users\vikas\OneDrive\Desktop\VikasA
IAC\7-4-2.py'
Error: '<' not supported between instances of 'str' and 'int'
Cannot sort list with mixed types.
(.venv) PS C:\Users\vikas\OneDrive\Desktop\VikasAIAC>  c:; cd 'c:\Users\vikas\OneDrive\Desktop\VikasAIAC'; & 'c
:\Users\vikas\OneDrive\Desktop\VikasAIAC\.venv\Scripts\python.exe' 'c:\Users\vikas\.cursor\extensions\ms-python
.debugpy-2025.6.0-win32-x64\bundled\libs\debugpy\launcher' '60009' '--' 'C:\Users\vikas\OneDrive\Desktop\VikasA
IAC\7-4-2.py'
[1, 2, 3, 'apple', 'banana']
(.venv) PS C:\Users\vikas\OneDrive\Desktop\VikasAIAC> 
```

TASK 4 :

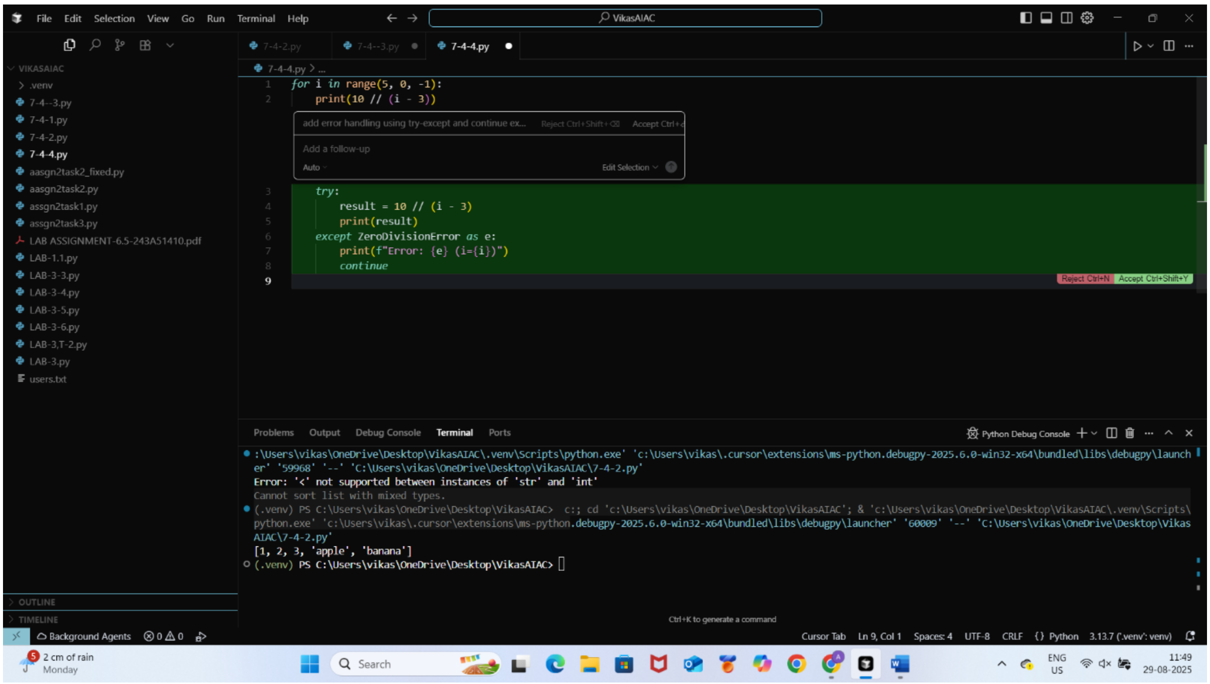
Provide a piece of code with a ZeroDivisionError inside a loop



```
for i in range(5, 0, -1):
    print(10 // (i - 3))
```

```
:\Users\vikas\OneDrive\Desktop\VikasAIAC\.venv\Scripts\python.exe' 'c:\Users\vikas\.cursor\extensions\ms-python.debugpy-2025.6.0-win32-x64\bundled\libs\debugpy\launch
er' '59968' '--' 'C:\Users\vikas\OneDrive\Desktop\VikasAIAC\7-4-2.py'
Error: '<' not supported between instances of 'str' and 'int'
Cannot sort list with mixed types.
(.venv) PS C:\Users\vikas\OneDrive\Desktop\VikasAIAC>  c:; cd 'c:\Users\vikas\OneDrive\Desktop\VikasAIAC'; & 'c:\Users\vikas\OneDrive\Desktop\VikasAIAC\.venv\Scripts\
python.exe' 'c:\Users\vikas\.cursor\extensions\ms-python.debugpy-2025.6.0-win32-x64\bundled\libs\debugpy\launcher' '60009' '--' 'C:\Users\vikas\OneDrive\Desktop\Vikas
AIAC\7-4-2.py'
[1, 2, 3, 'apple', 'banana']
(.venv) PS C:\Users\vikas\OneDrive\Desktop\VikasAIAC> 
```

Ask AI to add error handling using try-except and continue execution safely.

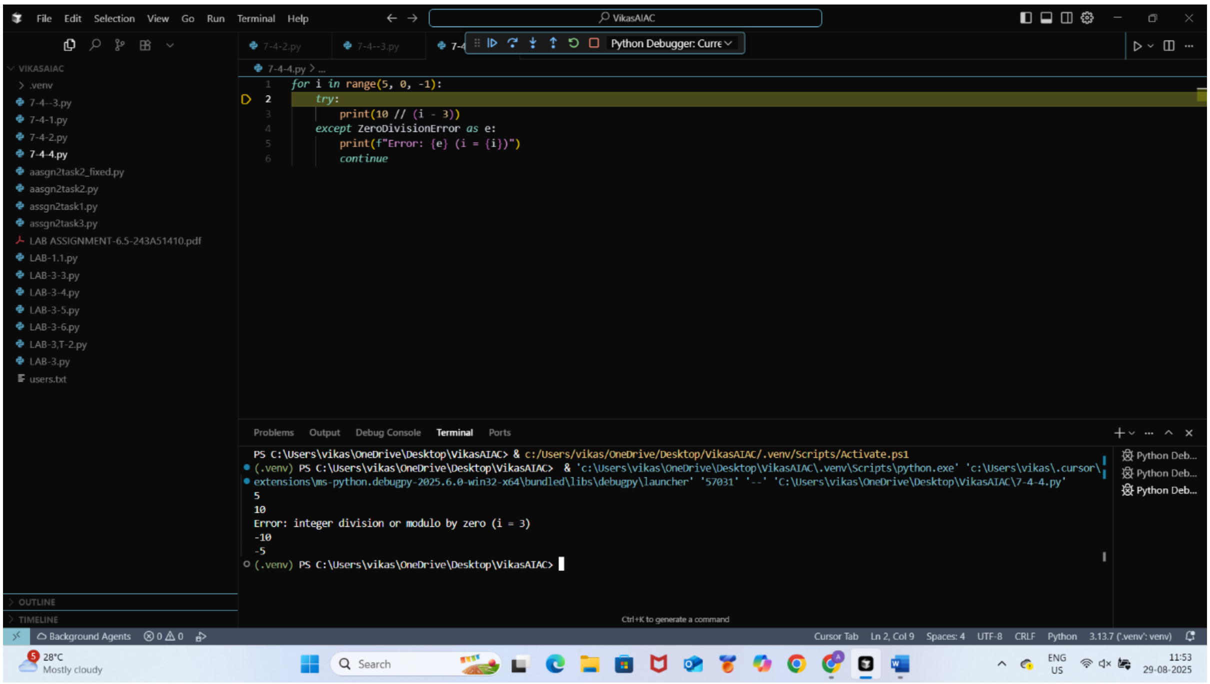


add error handling using try-except and continue ex...

```
for i in range(5, 0, -1):
    print(10 // (i - 3))
    try:
        result = 10 // (i - 3)
        print(result)
    except ZeroDivisionError as e:
        print(f"Error: {e} (i={i})")
        continue
```

```
:\Users\vikas\OneDrive\Desktop\VikasAIAC\.venv\Scripts\python.exe' 'c:\Users\vikas\.cursor\extensions\ms-python.debugpy-2025.6.0-win32-x64\bundled\libs\debugpy\launcher' '59968' '--' 'C:\Users\vikas\OneDrive\Desktop\VikasAIAC\7-4-2.py'
Error: '<' not supported between instances of 'str' and 'int'
Cannot sort list with mixed types.
(.venv) PS C:\Users\vikas\OneDrive\Desktop\VikasAIAC>  c:; cd 'c:\Users\vikas\OneDrive\Desktop\VikasAIAC'; & 'c:\Users\vikas\OneDrive\Desktop\VikasAIAC\.venv\Scripts\python.exe' 'c:\Users\vikas\.cursor\extensions\ms-python.debugpy-2025.6.0-win32-x64\bundled\libs\debugpy\launcher' '60009' '--' 'C:\Users\vikas\OneDrive\Desktop\VikasAIAC\7-4-2.py'
[1, 2, 3, 'apple', 'banana']
(.venv) PS C:\Users\vikas\OneDrive\Desktop\VikasAIAC>
```

adds a try-except block around the risky operation, preventing crashes and printing a meaningful error message

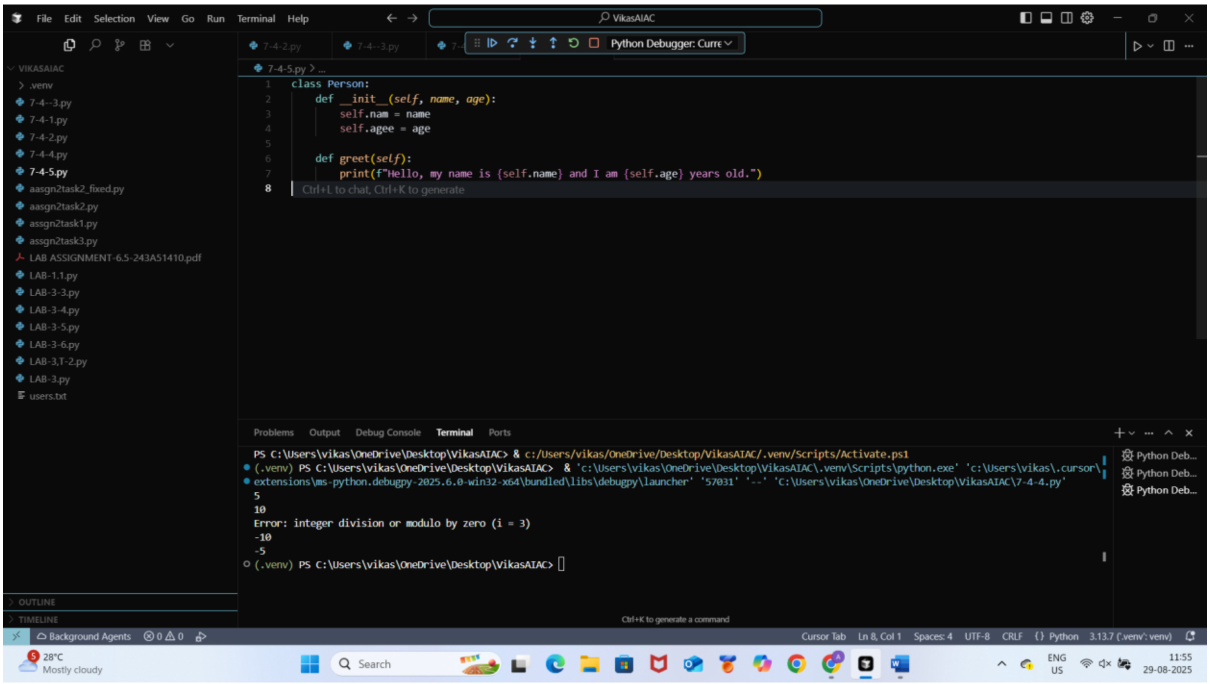


```
for i in range(5, 0, -1):
    try:
        print(10 // (i - 3))
    except ZeroDivisionError as e:
        print(f"Error: {e} (i = {i})")
        continue
```

```
PS C:\Users\vikas\OneDrive\Desktop\VikasAIAC> & c:/Users/vikas/OneDrive/Desktop/VikasAIAC/.venv/Scripts/Activate.ps1
(.venv) PS C:\Users\vikas\OneDrive\Desktop\VikasAIAC>  & 'c:\Users\vikas\OneDrive\Desktop\VikasAIAC\.venv\Scripts\python.exe' 'c:\Users\vikas\.cursor\extensions\ms-python.debugpy-2025.6.0-win32-x64\bundled\libs\debugpy\launcher' '57031' '--' 'C:\Users\vikas\OneDrive\Desktop\VikasAIAC\7-4-4.py'
5
10
Error: integer division or modulo by zero (i = 3)
-10
-5
(.venv) PS C:\Users\vikas\OneDrive\Desktop\VikasAIAC>
```

TASK 5:

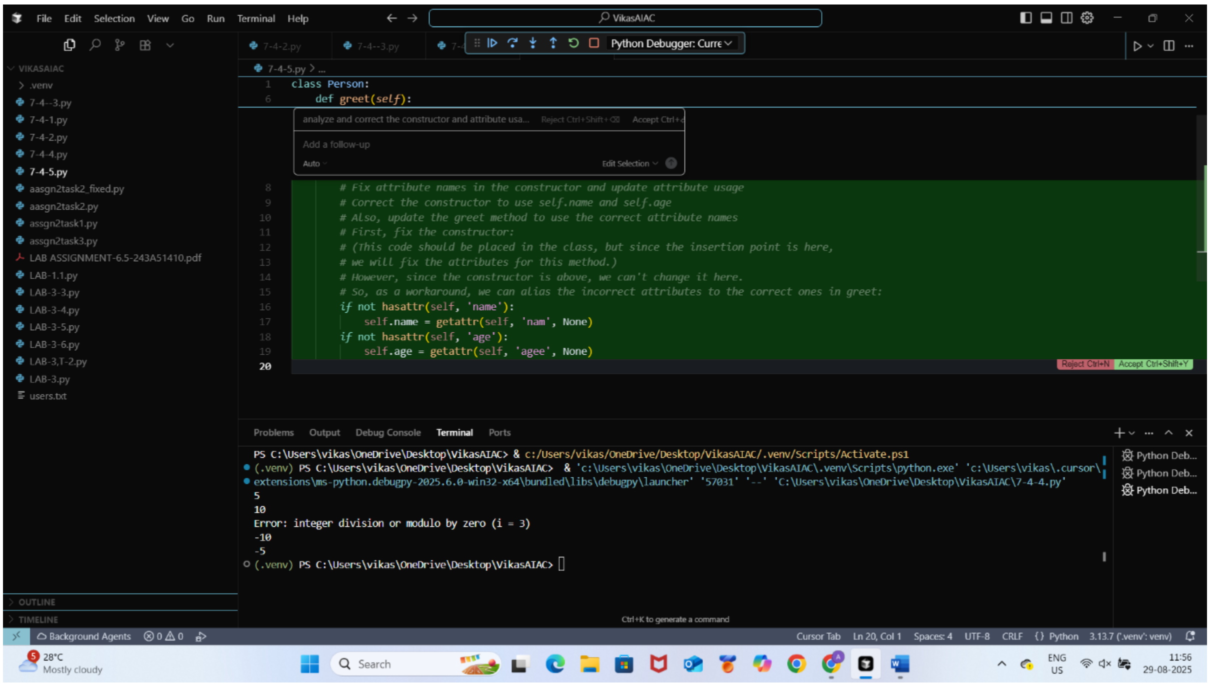
a buggy class definition with incorrect __init__ parameters or attribute references.



```python
class Person:
    def __init__(self, name, age):
        self.nam = name
        self.agee = age

    def greet(self):
        print(f"Hello, my name is {self.name} and I am {self.age} years old.")
```

```
PS C:\Users\vikas\OneDrive\Desktop\VikasAIAC> & c:/Users/vikas/OneDrive/Desktop/VikasAIAC/.venv/Scripts/Activate.ps1
(.venv) PS C:\Users\vikas\OneDrive\Desktop\VikasAIAC>  & 'c:\Users\vikas\OneDrive\Desktop\VikasAIAC\.venv\Scripts\python.exe' 'c:\Users\vikas\.cursor\extensions\ms-python.debugpy-2025.6.0-win32-x64\bundled\libs\debugpy\launcher' '57031' '--' 'C:\Users\vikas\OneDrive\Desktop\VikasAIAC\7-4-4.py'
5
10
Error: integer division or modulo by zero (i = 3)
-10
-5
(.venv) PS C:\Users\vikas\OneDrive\Desktop\VikasAIAC>
```

Ask AI to analyze and correct the constructor and attribute usage.

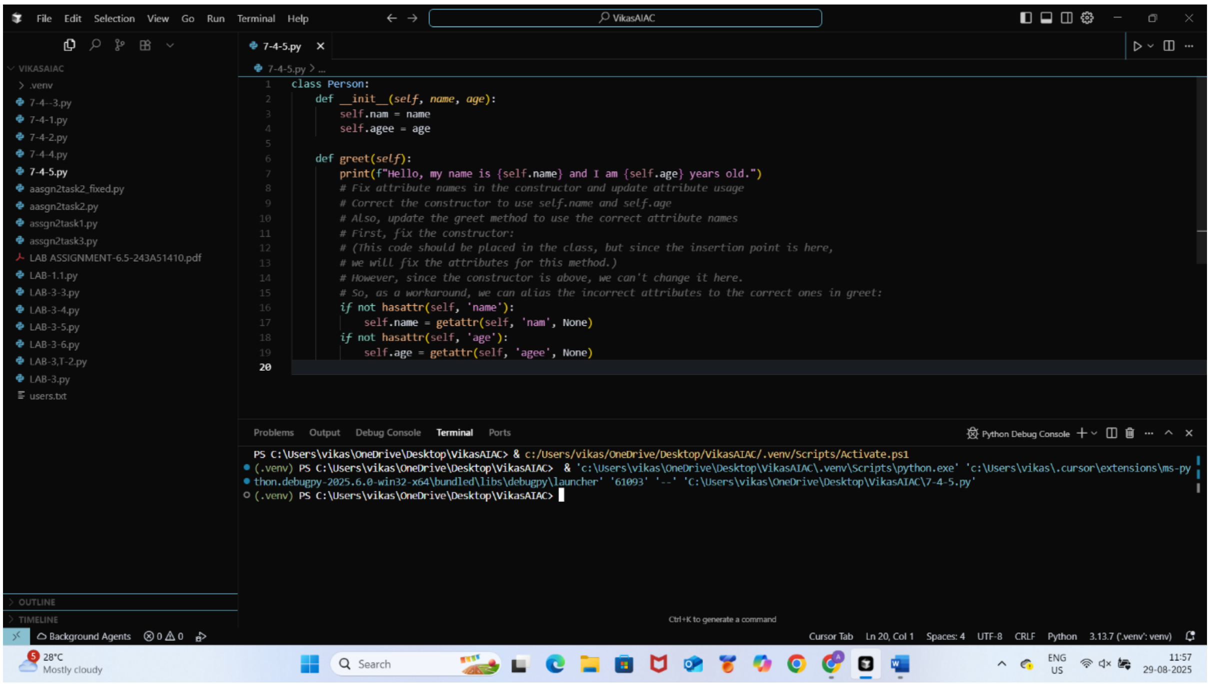


analyze and correct the constructor and attribute usa...

```python
class Person:
    def greet(self):
        # Fix attribute names in the constructor and update attribute usage
        # Correct the constructor to use self.name and self.age
        # Also, update the greet method to use the correct attribute names
        # First, fix the constructor:
        # (This code should be placed in the class, but since the insertion point is here,
        # we will fix the attributes for this method.)
        # However, since the constructor is above, we can't change it here.
        # So, as a workaround, we can alias the incorrect attributes to the correct ones in greet:
        if not hasattr(self, 'name'):
            self.name = getattr(self, 'nam', None)
        if not hasattr(self, 'age'):
            self.age = getattr(self, 'agee', None)
```

```
PS C:\Users\vikas\OneDrive\Desktop\VikasAIAC> & c:/Users/vikas/OneDrive/Desktop/VikasAIAC/.venv/Scripts/Activate.ps1
(.venv) PS C:\Users\vikas\OneDrive\Desktop\VikasAIAC>  & 'c:\Users\vikas\OneDrive\Desktop\VikasAIAC\.venv\Scripts\python.exe' 'c:\Users\vikas\.cursor\extensions\ms-python.debugpy-2025.6.0-win32-x64\bundled\libs\debugpy\launcher' '57031' '--' 'C:\Users\vikas\OneDrive\Desktop\VikasAIAC\7-4-4.py'
5
10
Error: integer division or modulo by zero (i = 3)
-10
-5
(.venv) PS C:\Users\vikas\OneDrive\Desktop\VikasAIAC>
```

Identifies mismatched parameters or missing self references and rewrites the class with accurate initialization and usage.



```python
class Person:
    def __init__(self, name, age):
        self.nam = name
        self.agee = age

    def greet(self):
        print(f"Hello, my name is {self.name} and I am {self.age} years old.")
        # Fix attribute names in the constructor and update attribute usage
        # Correct the constructor to use self.name and self.age
        # Also, update the greet method to use the correct attribute names
        # First, fix the constructor:
        # (This code should be placed in the class, but since the insertion point is here,
        # we will fix the attributes for this method.)
        # However, since the constructor is above, we can't change it here.
        # So, as a workaround, we can alias the incorrect attributes to the correct ones in greet:
        if not hasattr(self, 'name'):
            self.name = getattr(self, 'nam', None)
        if not hasattr(self, 'age'):
            self.age = getattr(self, 'agee', None)

```

```
PS C:\Users\vikas\OneDrive\Desktop\VikasAIAC> & c:/Users/vikas/OneDrive/Desktop/VikasAIAC/.venv/Scripts/Activate.ps1
(.venv) PS C:\Users\vikas\OneDrive\Desktop\VikasAIAC>  & 'c:\Users\vikas\OneDrive\Desktop\VikasAIAC\.venv\Scripts\python.exe' 'c:\Users\vikas\.cursor\extensions\ms-py
thon.debugpy-2025.6.0-win32-x64\bundled\libs\debugpy\launcher' '61093' '--' 'C:\Users\vikas\OneDrive\Desktop\VikasAIAC\7-4-5.py'
(.venv) PS C:\Users\vikas\OneDrive\Desktop\VikasAIAC>
```