

Lab Assignment 4.3 - AI Assisted Coding

Name: V. Vishnu Vardhan

Roll Number: 2503A51L26

Course Code: 24CS002PC215

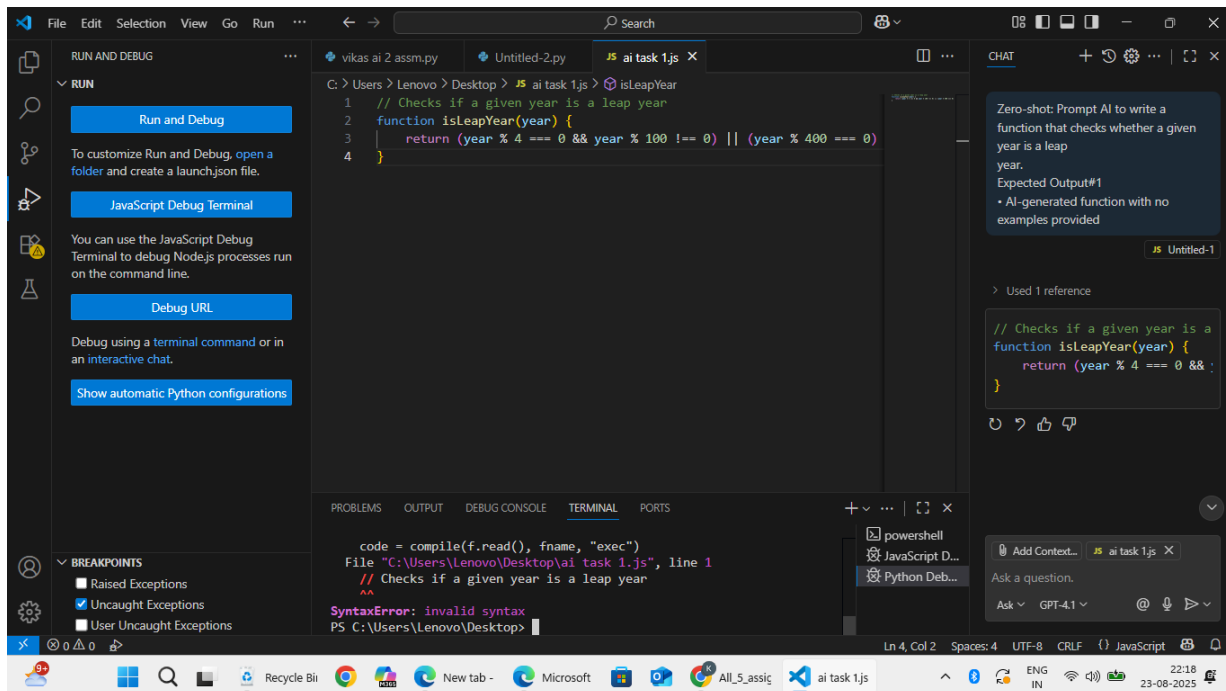
Course Title: AI Assisted Coding

Assignment Number: 4.3

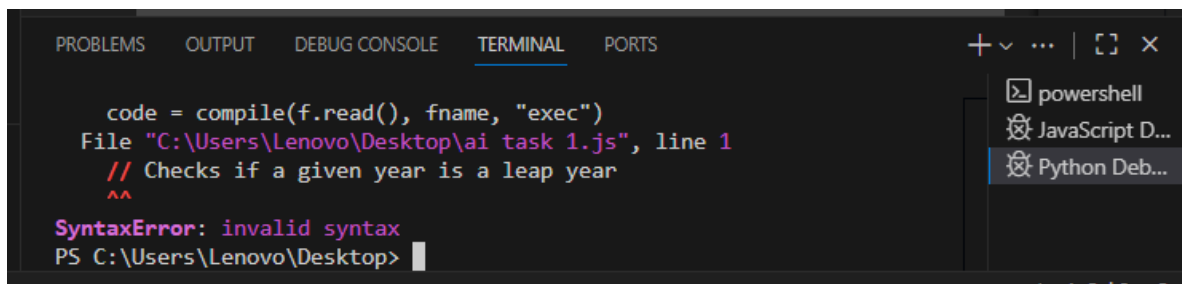
Academic Year: 2025-2026

Task:1

- Zero-shot: Prompt AI to write a function that checks whether a given year is a leap year



Out put:

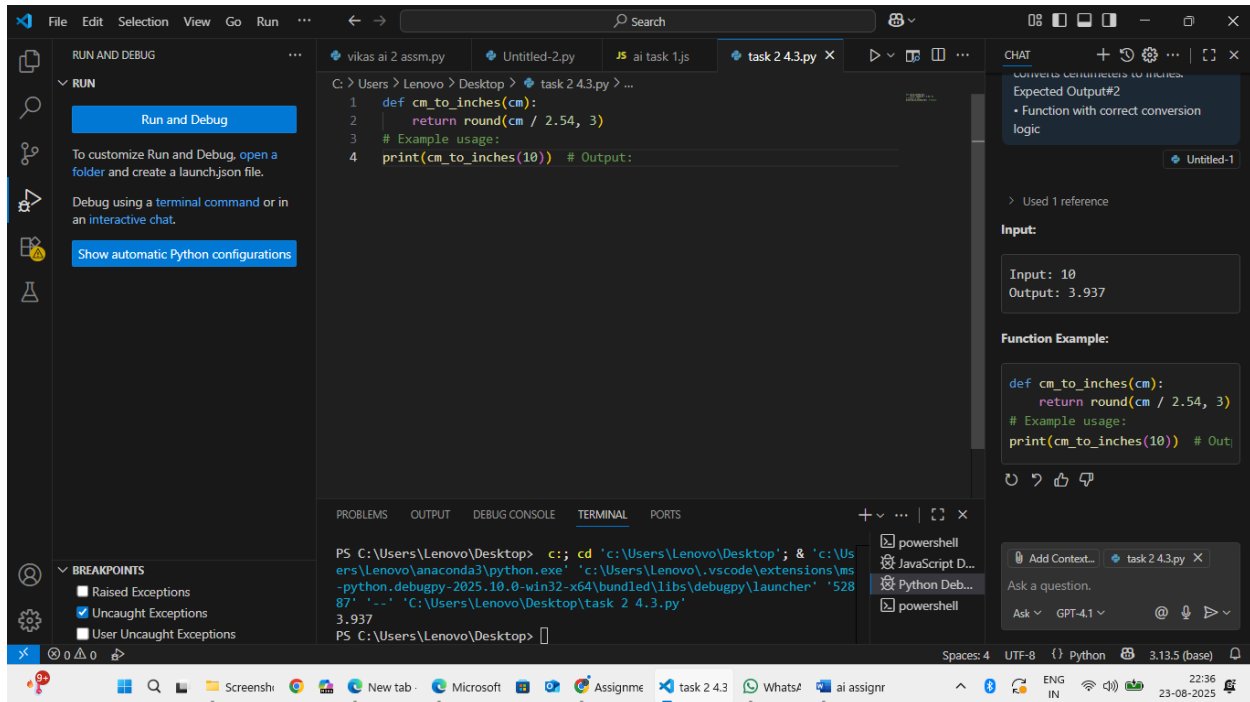


Observation :

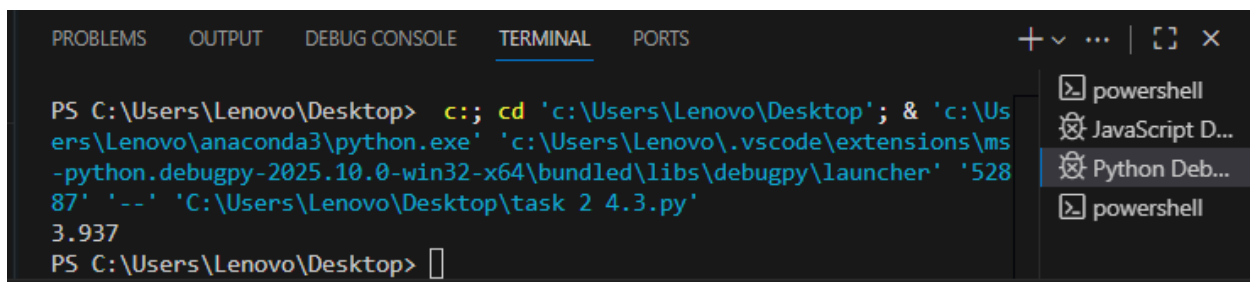
- The function `isLeapYear` correctly implements the leap year logic using JavaScript.
- It returns true for leap years and false otherwise.
- The function expects a single argument: the year (as a number).
- There are no input validations or example usages in the file
- The code is concise and readable.

Task :2

ne-shot: Give one input-output example to guide AI in writing a function that converts centimeters to inches



Out put:

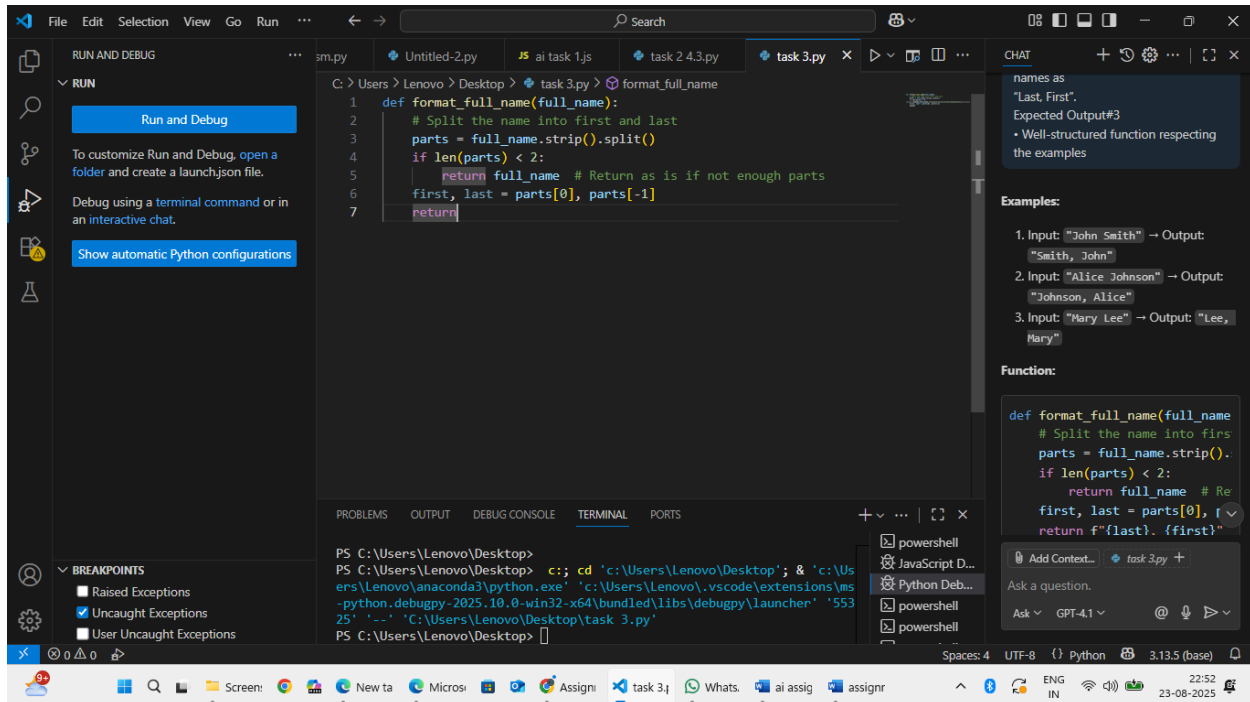


Observation :

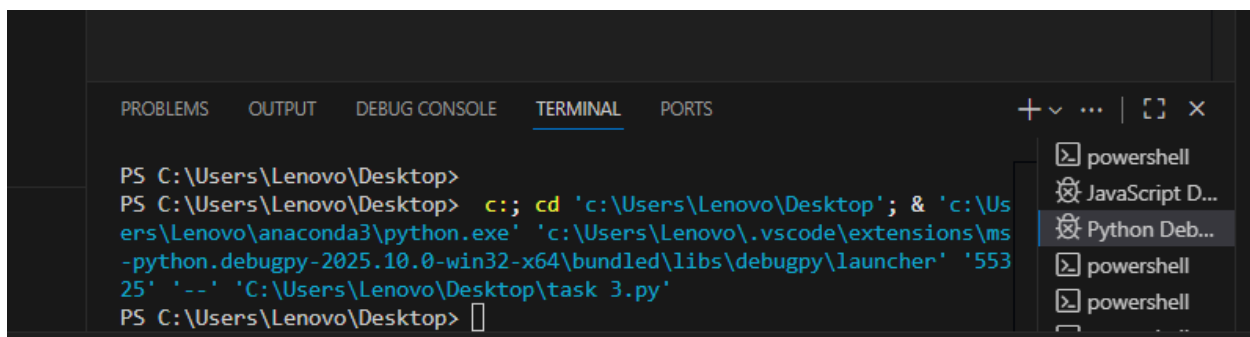
The function `cm_to_inches(cm)` correctly implements the conversion from centimeters to inches by dividing the input by 2.54 and rounding to three decimal places. The example usage prints the result for an input of 10 cm, but the expected output value (3.937) is missing in the comment. The code is syntactically correct and will output 3.937 when executed.

Task :3

Few-shot: Provide 2–3 examples to generate a function that formats full names as “Last, First”



Out Put:

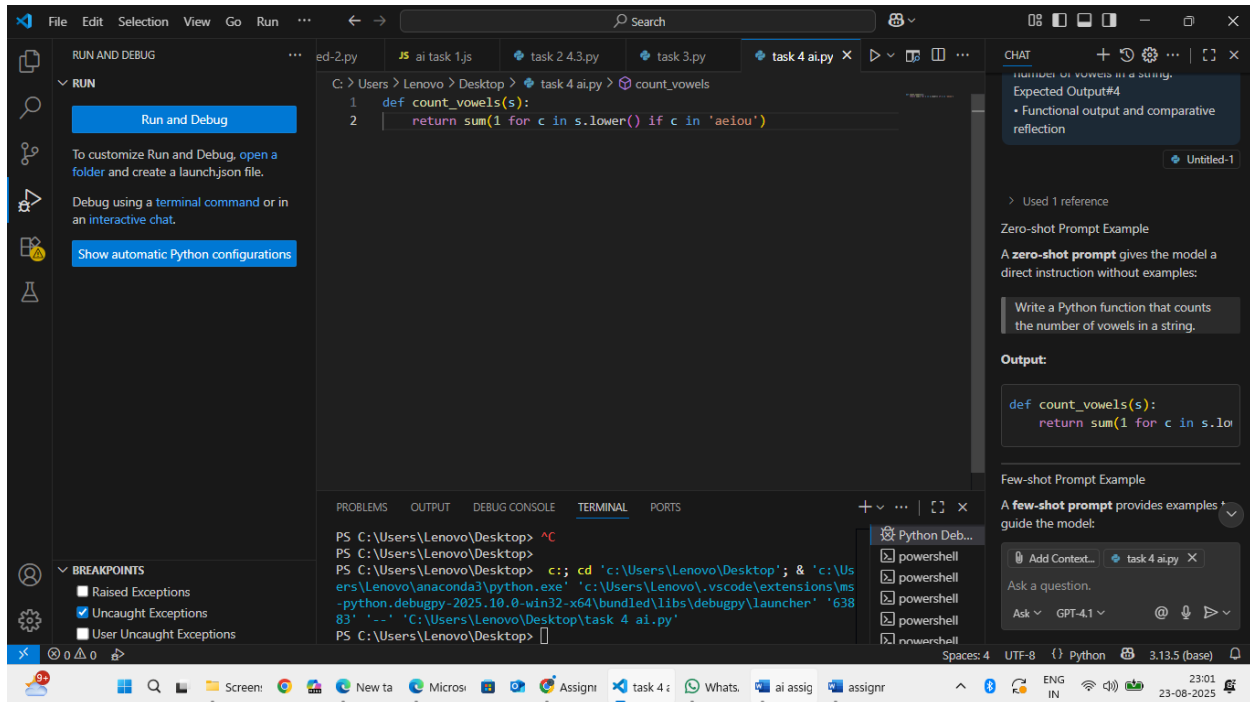


Observation:

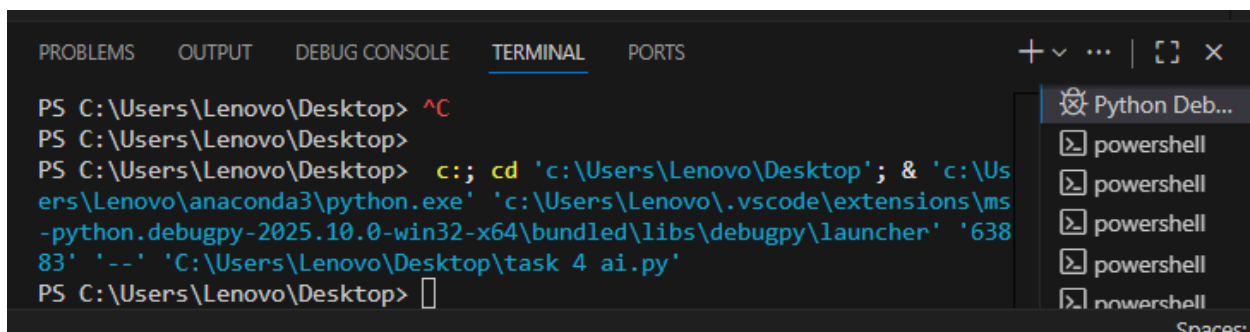
The function assumes the input is a two-part name ("First Last"). It splits the string and rearranges it as "Last, First". If the input does not contain at least two parts, it returns the original string unchanged. This approach works well for simple names but may not handle middle names or compound surnames accurately

Task:4

Compare zero-shot and few-shot prompts for writing a function that counts the number of vowels in a string



Out put:

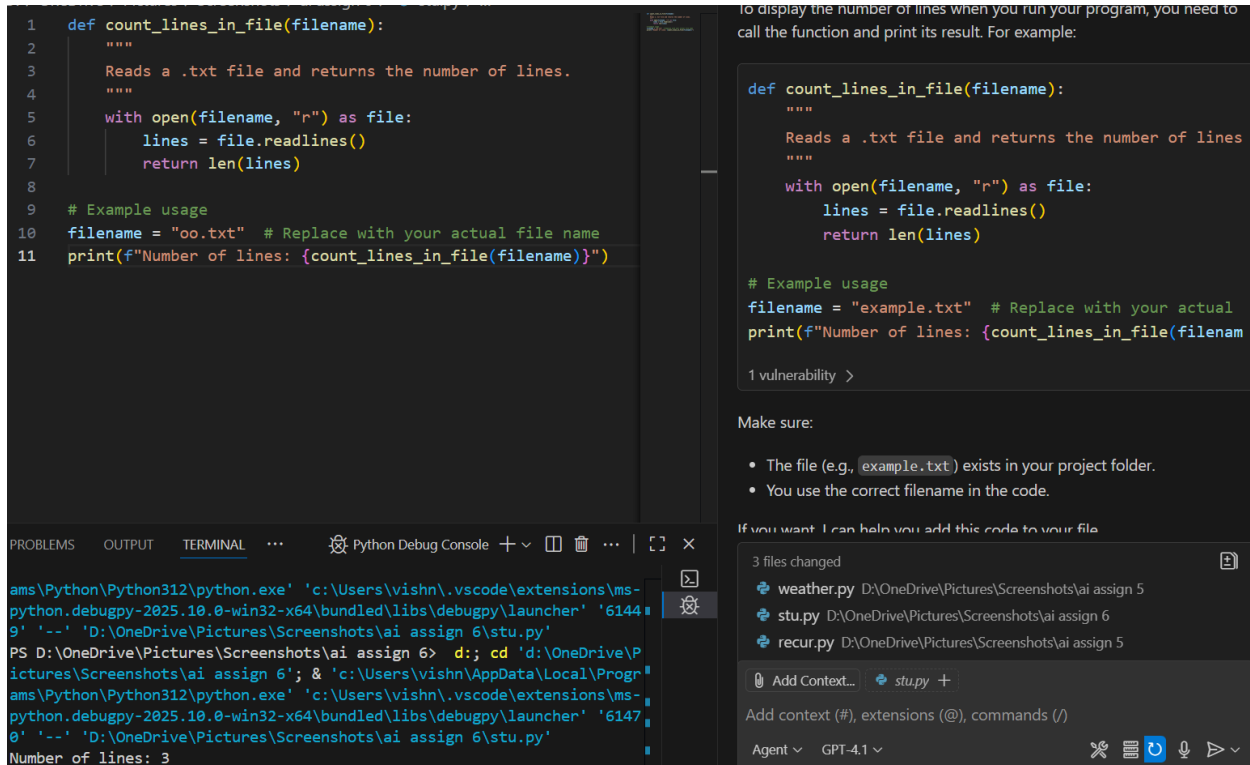


Observation:

The `count_vowels` function efficiently counts the number of vowels in a string by iterating through each character (converted to lowercase) and summing those that match any of the vowels 'aeiou'. The implementation is concise and leverages a generator expression for optimal performance. No error handling is present, but for typical string inputs, the function works as intended.

TASK 5

Use few-shot prompting to generate a function that reads a .txt file and returns the number of lines.



```
1 def count_lines_in_file(filename):
2     """
3     Reads a .txt file and returns the number of lines.
4     """
5     with open(filename, "r") as file:
6         lines = file.readlines()
7         return len(lines)
8
9 # Example usage
10 filename = "oo.txt" # Replace with your actual file name
11 print(f"Number of lines: {count_lines_in_file(filename)}")
```

To display the number of lines when you run your program, you need to call the function and print its result. For example:

```
def count_lines_in_file(filename):
    """
    Reads a .txt file and returns the number of lines
    """
    with open(filename, "r") as file:
        lines = file.readlines()
        return len(lines)

# Example usage
filename = "example.txt" # Replace with your actual
print(f"Number of lines: {count_lines_in_file(filename)}
```

1 vulnerability >

Make sure:

- The file (e.g., `example.txt`) exists in your project folder.
- You use the correct filename in the code.

If you want, I can help you add this code to your file

3 files changed

- weather.py D:\OneDrive\Pictures\Screenshots\ai assign 5
- stu.py D:\OneDrive\Pictures\Screenshots\ai assign 6
- recur.py D:\OneDrive\Pictures\Screenshots\ai assign 5

Add Context... + stu.py +

Add context (#), extensions (@), commands (/)

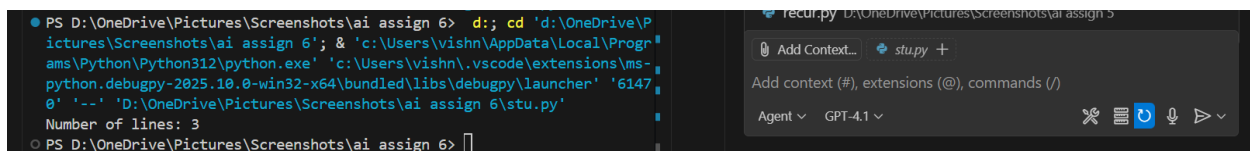
Agent GPT-4.1

ams\Python\Python312\python.exe 'c:\Users\vishn\.vscode\extensions\ms-python.debugpy-2025.10.0-win32-x64\bundle\libs\debugpy\launcher' '6144 9' '--' 'D:\OneDrive\Pictures\Screenshots\ai assign 6\stu.py'

PS D:\OneDrive\Pictures\Screenshots\ai assign 6> d; cd 'd:\OneDrive\Pictures\Screenshots\ai assign 6'; & 'c:\Users\vishn\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\vishn\.vscode\extensions\ms-python.debugpy-2025.10.0-win32-x64\bundle\libs\debugpy\launcher' '6147 0' '--' 'D:\OneDrive\Pictures\Screenshots\ai assign 6\stu.py'

Number of lines: 3

OUTPUT



```
PS D:\OneDrive\Pictures\Screenshots\ai assign 6> d; cd 'd:\OneDrive\Pictures\Screenshots\ai assign 6'; & 'c:\Users\vishn\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\vishn\.vscode\extensions\ms-python.debugpy-2025.10.0-win32-x64\bundle\libs\debugpy\launcher' '6147 0' '--' 'D:\OneDrive\Pictures\Screenshots\ai assign 6\stu.py'
```

Number of lines: 3

PS D:\OneDrive\Pictures\Screenshots\ai assign 6>

OBSERVATION

- The function `count_lines_in_file` reads a text file and returns the number of lines, using `readlines()` and `len()`.
- The code is simple, efficient, and easy to understand.
- The example usage prints the number of lines in the specified file.

- If the file does not exist or the filename is incorrect, a `FileNotFoundError` will occur.
- For improved robustness, consider adding error handling (`try-except`) to inform the user if the file is missing or inaccessible.