I used Profiling Tool - DevTools Performance tab, whichconfirmed the backend crashed during the search request. (Kloda, 2025a) It was caused because the return object format didn’t match with the front end trying to parse it.   
Refactored SemanticSearchBar.js reduced bug reports in subsequent sprints - test update. (Kloda, 2025b)

Python debugger determined the point of failure, it allowed me to pause execution and inspect the exact input sent to prompt\_lama, helping you identify formatting or logic issues. This insight enabled me to refactor how recommendations were generated and cleaned, ensuring accurate and properly formatted output. (Kloda, 2025c; Kloda, 2025d)

Initially, the codebase was structured to use a GPU for model inference, however, my development environment does not have a dedicated GPU, so these GPU calls were failing. (Kloda, 2025e)

I refactored the code to explicitly use the CPU instead, removing GPU-specific logic and optimizing for CPU inference. After the changes, I observed change of API response time. (Kloda, 2025f)

# References

Kloda (2025). [online] github. Available at: https://github.com/Jkloda/LOJuliaKloda/blob/main/Assets/DevTools.png [Accessed 21 May 2025].

Kloda (2025). [online] github. Available at: https://github.com/Jkloda/LOJuliaKloda/blob/main/Assets/JestTest.png [Accessed 21 May 2025].

Kloda (2025). [online] github. Available at: https://github.com/Jkloda/LOJuliaKloda/blob/main/Assets/Debugger.png [Accessed 21 May 2025].

Kloda (2025). [online] github. Available at: https://github.com/Jkloda/LOJuliaKloda/blob/main/Assets/CPU.png [Accessed 21 May 2025].

Kloda (2025). [online] github. Available at: https://github.com/Jkloda/LOJuliaKloda/blob/main/Assets/FaissFail.png [Accessed 21 May 2025].

Kloda (2025). [online] github. Available at: https://github.com/Jkloda/LOJuliaKloda/blob/main/Assets/Performance.png [Accessed 21 May 2025].