

1. What challenges did you face during the project?

The first big challenge was integrating the Cohere API correctly. At first, I had a lot of trouble because ChatGPT, which I was using to write code, didn't fully understand how the Cohere API worked. I kept getting errors, and I had to rewrite code many times. Eventually, I realized it was easier to just ask ChatGPT to check the official documentation online, and after that it gave the correct code right away.

Another challenge was running the final tests through the terminal. Some tests simply wouldn't run in my environment with Python 3.14, which is experimental for some libraries. ChatGPT explained that such failures are known to happen in this setup. Because of that, I had to come up with a different testing approach, mock some things, and adjust the code to make the tests pass.

Finally, the original lab code was designed for Anthropic models, so I had to refactor the entire codebase to work with Cohere. That made the project take a lot longer than expected, because I had to carefully adapt token counting, API calls, and fallback handling.

2. How did you solve these challenges?

For the Cohere integration, the solution was to consult the official API documentation online with ChatGPT, which helped me get the working code.

For the testing issue, I refactored the `test_summarizer.py` to mock rate-limiting and token counting, which allowed all tests to pass without relying on the terminal environment.

And for Anthropic → Cohere, I systematically rewrote all calls, token counting, and fallback logic to match Cohere's API and constraints.

3. What did you learn from this project?

I learned how to work with multiple LLM providers, handle rate-limiting, count tokens, track API costs, and write robust unit tests that mock external services. I also gained practical experience debugging experimental Python environments and understanding the importance of reading and using official documentation.

4. What ideas for improvement do you have?

One improvement would be to design the lab assignment for Cohere from the start, so students don't have to refactor everything. Another idea could be to provide pre-configured mocks and templates to make testing and API integration easier and faster. Also, clearer instructions for async vs sync processing and rate-limiting strategies could save time.