For my applet, I decided to use local models for my embedding model and the LLM model for answering prompts. I made this decision because using models on AWS or Microsoft Azure would cost money, which I did not want to do since my applet is just for a job application. For my embedding model, I choose the nomic embed model as it was the best embedding model I could find that could be run locally. For my prompt answering model, I choose llama3.1 with 8B parameters quantized (can be found here). I choose this because llama3.1 is considered a state-of-the-art model, and 8B parameters quantized is the upper limit of what I can support on my computer as I only have 8 GB of VRAM. I used Ollama for running local LLM's, as that is what I have experience in, and I used ChromaDB for my vector database as that is what I found most suitable from doing research online (I did not use any generative AI for doing research). For the data I use, I use the publicly available US Army Field Manuals FM 2-0 Intelligence, FM 3-0 Operations, and FM 3-90 Tactics (I think this is a fitting choice since Torch.ai is a government contractor). I only used a small amount of documents since they are there mostly just for testing, but for a real-world application where I don't have to rely on local LLM's I would certainly use much more. For the UI, I kept everything within the command line, since as stated before this is just an applet for a job application. I did rely on this video https://www.youtube.com/watch?v=2TJxpyO3ei4 and its GitHub code a great deal when designing my applet, though I did make my own changes where I felt such was needed. For my applet, I allow the possibility of adding new documents to the vector database and resetting it. I didn't implement the functionality of updating existing documents, as I felt that such would be outside the scope of the assigned task. I did add a chat history that would be for the same session, but I did not add chat history prolonging through multiple sessions of running the applet. I did the latter intentionally because I wanted users to have each session be "clean" for them. For semantic searching, I only get the top 6 searches due to wishing to not have long search times for that and because I think six sources are plenty for the user to have.

I did have to make a few tradeoffs when designing my applet. The biggest one is only using local models, which does limit the accuracy of my applet's responses, and especially how proficient it is at doing a semantic search, but such was necessary considering everything for the applet needed to be free. Also, as stated before I only used a small amount of documents, since I could only use local LLM's and I mostly just wanted the documents used to be there for testing purposes. In addition, I could only use an 8B parameter LLM model instead of something stronger due to the hardware limitations of my laptop. If I had a more powerful device, I certainly would have used a more powerful model. Finally, for

semantic searching I choose to just get the top 6 sources as a good balance between speed and showing helpful references to the user.