Of my contributions to the project, one of my largest contributions was the development of the user experience and website design of the project. This task was something that built of skills in my initial assessment, that I've developed in my co-ops, and in other coursework. This task allowed me to learn more about frontend development from nothing. I learned how to use the React JavaScript framework, create multiple page routes, connect a frontend to a RESTful API, and use lifecycle hooks for components in a single page application. As a whole, the frontend was well designed and developed and no issues from a user's standpoint. The biggest success was creating the chat window and messages for our chat application with no issues of load times or lag. One of the big obstacles I faced was the restructuring of the chat page route. As we developed our early prototype, we had the chat window readily available off the home page but later decided to put a page in between to allow the user to select different sessions. Restructuring this path was a challenge, and I had to learn how to pass function references and properties through multiple components to allow the chat page to still access them. This was eventually solved for our final product, and I had no issues fully restoring functionality.

One of my other contributions to the project was the development of the conversation logic, writing tools and conditionals to handle the conversation agenda and building the system instructions. My goal for this project was to improve my backend development skills and this task allowed me to do that. In our project, every user message in the chat gets routed to our backend server to a pipeline of functions that handles the conversation logic. To build the system instructions, I had to implement Django caching to quickly retrieve Json information on the server, and then extract individual data based on the status of the conversation agenda status. I stored the status in the conversation object as a list of 0s, 1s, and 2s, using an enumerate to translate to not started, current, and completed. I also engineered prompts and implemented tools that allow the OpenAI API to call functions that performed actions such as updated agenda items to complete and picking new agenda items. The biggest obstacle I encountered was testing the validity of these prompts and instructions. Testing LLM instructions cannot be done by normal code testing but must be rigorously tested and attempted to be broken. This task tool a significant amount of time, but a satisfactory threshold was able to be met for our final product.