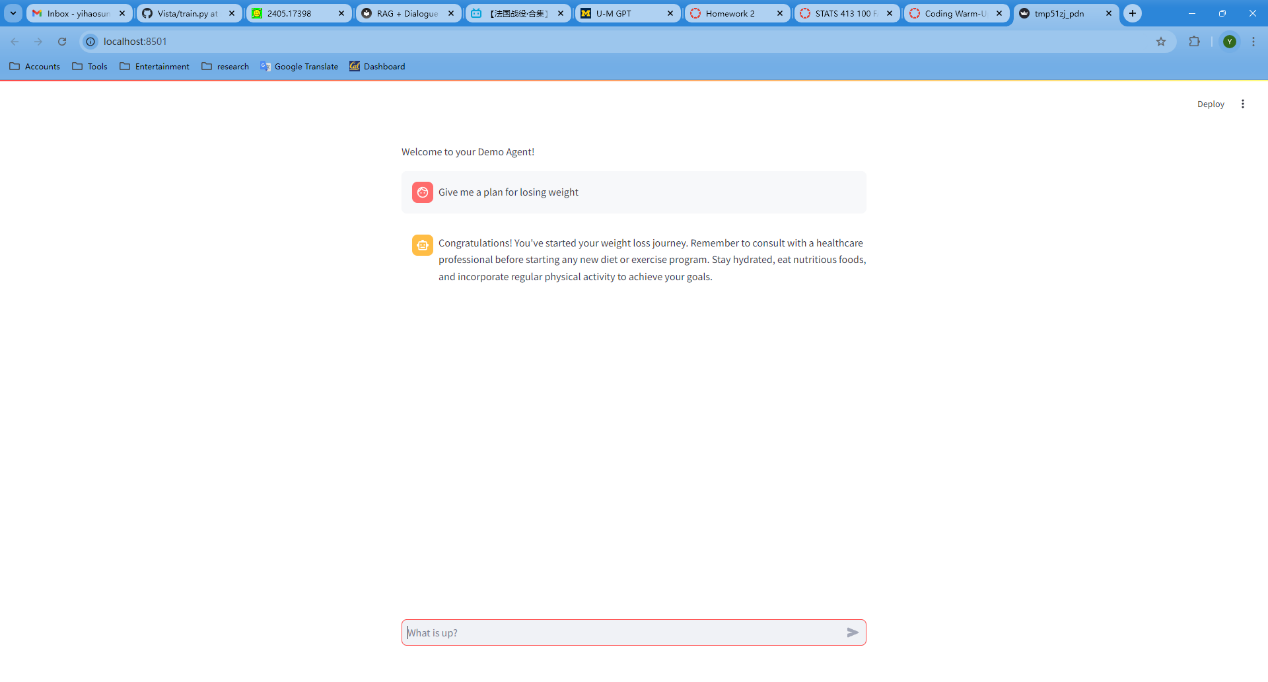
How things went:

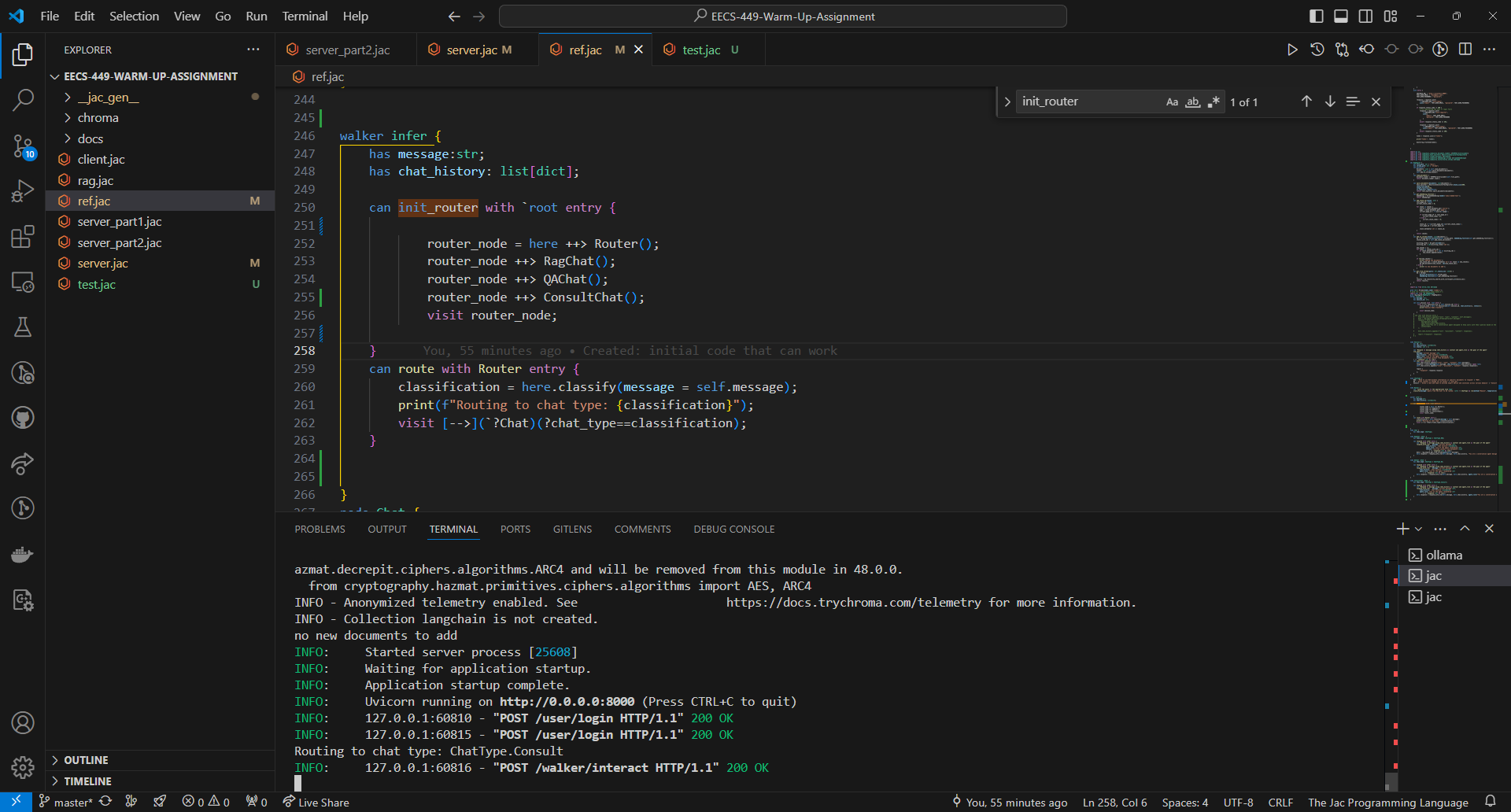
Originally, I decided to first replicate everything in the guide before working on the additional tasks in the canvas assignment. However, as I found that the second part requires me to delete everything in the server.jac, I chose to work on the assignment before continuing.

While working on the first part, I met some trouble about the environment. The problem was that the commands provided in the guide cannot be run in windows powershell. For example, when running ‘DATABASE\_HOST=mongodb://localhost:27017/?replicaSet=my-rs jac serve server.jac’, powershell would report a ‘command not found’ error. As a result, I had to switch to git bash terminal to run these commands. And it took me some time to get conda available in git bash by adding conda to the PATH in the ‘~/.bashrc’ file. The task to add walkers did not bother me a lot. I created one walker taking ‘time’, a Boolean variable, to output ‘good morning’ or ‘good evening’. And that is the reason why I add two screenshots to show the function of the first walker. The other walker is simply a difference calculator.

As for the second part, I first followed the guide and attempted to use chat-gpt api. However, since I do not have a premium account, I could not use that api to generate any response. So, I switched to ollama library and used llama3.1 and qwen2.5 after the initial commit in part 2. Similar to that in the part 1, I had to add ollama into the PATH in the ‘~/.bashrc’ file to make everything work in git bash.

In the third part, I met some trouble running the script after the modification. It kept reporting error complaining the not existence of attribute ‘response’. After trying for several hours, it turned out that the Router is not updated after the classification is correctly done. So, I removed the line “visit [-->](`?Router) else” to make it run each time the server starts. During the debugging process, I created two new files, test.jac and ref.jac to store progress and do some little experiments to find out the root problem. After the tasks were completed, I deleted these files. Another thing worth mentioning is that I changed my chat-type half way from “Consult” to “Comfort”. It was mainly an issue about the quality of the response.





As is shown in the second screenshot above, the input “Give me a plan for losing weight” is correctly classified into “Consult”. However, the actual response given by the chatbot in the first screenshot was weird since it did not yield any actual plan as expected. On the other hand, the “Comfort” works well in its response as is shown in the screenshot in the folder. As a result, I switched to “Comfort” instead of “Consult”.

GitHub repo link: <https://github.com/Alexcel-Harry/EECS-449-Warm-Up-Assignment>