1. Add a new custom state of your own to the `ChatType` enum. Explain what the state represents and how it can be used in the chatbot.

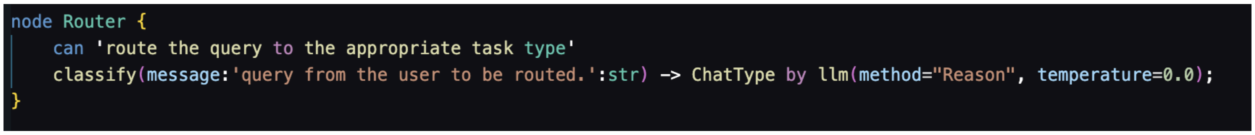
I added an ‘SQ’ or ‘Summarize Question’ ChatType. My goal with this mode was for when the user is asking for a summary or an informative but concise paragraph on one of the documents, this ChatType would be used to help answer the question more effectively.

2. Add your own custom `chat` node to the graph and define its abilities.

My chat node is called ‘SQChat’ and it provides the chatbot with the context of the documents. It also gives a custom prompt to the bot so it can be more focused on the summary aspect of a question.

3. Add your custom state to the routing capabilities of the chatbot. Explain how the chatbot will route the user query to your custom chat model based on the classification.

The chatbot will route the user using the ‘Router’ node with the code below:



```
node Router {
    can 'route the query to the appropriate task type'
    classify(message:'query from the user to be routed.':str) -> ChatType by llm(method="Reason", temperature=0.0);
}
```

Essentially, the classify function queries the LLM to reason about which ChatType will be most effective by considering the question posed and the different ChatTypes.

In my screenshot ‘chat_type_changing.png’, you can see how the ChatType switches depending on the query. First I asked a question specifically requesting a summary, then a question that could be determined from context (not needing to load the documents). Finally, I asked a specific question that required the context, but not a summary.