**1) Was using stuff technique for summarization**

Initially I was using stuff techniwue for for summarization which exceeds the text limit of context window. So used map-reduce technique which basically summurise doc in chunkwise format and then provide final output

**2) time complexity was high for chat with data feature was reloading whole vectordatabase after each query**

To reduce the time complexity and improve the responsiveness of your Streamlit app, especially when handling queries, you can implement a few optimizations.

* Cache the vectorstore: Used Streamlit's caching mechanism to avoid reprocessing the PDF file every time the app reruns. This will significantly speed up subsequent queries.
* st.cache is a decorator that can be used to optimize the performance of your Streamlit app. When you apply st.cache to a function, it tells Streamlit to store the results of that function in a cache. The next time the function is called with the same inputs, Streamlit will return the cached result instead of recomputing it.
* Optimize the text splitting: Adjust the chunk size and overlap in the text splitter to balance between accuracy and speed.

**3) The problem was that our chat history was being shared between different pages (chat with data and chat with website) because I was using the same session state key for messages across both pages.**

To fix this, we need to create separate chat histories for each page. Changed the session state key from messages to web\_chat\_messages to create a separate chat history for this page.