Day 1: March 20th, 2024

Agenda Item: Initial Setup

- Action Items:
    - Erkang to review setup instructions and ensure all necessary tools and libraries are installed.
- Tasks:
    - Utilize the source codes provided in the provided drive link.
    - Ensure usage of Python scripts and other essential tools within the Linux environment.
    - Document setup steps and requirements for running scripts in the submission document.
    - Refer to the Readme.md file in the Drive folder for setup instructions, list of libraries, their versions, and installation steps required for Parts 1 and 2 of Lab 6.
    - Use App.py and html.py as a reference for the lab, and employ any tools for creating a GUI for the chatbot, including PHP, Python, JavaScript, HTML, StreamLit, etc.

Day 2: March 21st, 2024

Agenda Item: Domain-Specific Chatbot with Open-Source Resources

- Action Items:
    - Jason to research and experiment with open-source text embedding models to replicate the solution for Part 1.
- Tasks:
    - Replicate the solution for Part 1 using only open-source tools and models.
    - Utilize open-source local embedding models and LLMs to produce similar outputs as OpenAI embeddings.
    - Explore available open-source text embedding models at Hugging Face Models.
    - Additional guidance and information can be found at SBERT, along with a helpful video summary on open-source text embedding models.

Day 3: March 22nd, 2024

Agenda Item: Web Interface Design

- Action Items:
    - Riten to begin designing the web interface, focusing on user-friendly features and implementing necessary functionalities.
- Tasks:
    - Focus on designing a user-friendly chatbot interface using HTML and CSS.
    - Implement JavaScript for handling user interactions and communicating with the chatbot.
    - Implement a sidebar option for uploading single or multiple PDF documents for analysis.
    - Execute Python script to extract text, create chunks, generate word embeddings, and store in the vector database for PDF analysis.
    - Create a chat window with an input field for users to ask questions based on the PDF and display messages as conversations.
    - Store questions and messages in chat history for future reference.
    - Refer to html.py for starter code on displaying user and bot messages.