Project Summary

The project involved developing a chatbot assistant capable of answering questions based on scraped web information and manual documents. The solution was implemented using modern technologies such as Flask for creating the web application, Gunicorn for serving the application in production, and various components of LangChain for handling natural language processing.

The approach began with creating a basic Flask application to serve a chat interface. Then, web scraping functionality was integrated to obtain updated data from the company's website, "Promtior." The scraped data was combined with pre-defined manual documents to create a vector index using FAISS, enabling efficient semantic searches.

One of the main challenges was managing dependencies and library versions, especially with LangChain, which had several recent updates and deprecations. To resolve this, we opted not to specify exact versions in the requirements.txt file, allowing pip to automatically resolve compatible versions.

Another significant difficulty was deploying on Railway, where we encountered issues with insufficient memory for the container. This was addressed by optimizing memory usage in the code and adjusting the container configurations in Railway.

Diagram

The user asks a question in the web interface, which is sent to the Flask server. The server queries data from the web scraper or manual documents and stores this information in FAISS Vector Store. The OpenAI model uses the context from FAISS to generate a response, which the server sends back to the frontend and is displayed to the user. The entire process is managed with Gunicorn on Railway to ensure efficiency and accessibility.