**TA's 솔루션 해설**

* <https://huq8hcz9ktqsatygmrjsdm.streamlit.app/>
* <https://github.com/fullstack-gpt-python/assignment-15/blob/main/app.py>

1. 사용자의 OpenAI API 키 이용하기
   * st.text\_input을 사용하여 사용자의 OpenAI API 키를 입력받습니다.
   * 입력받은 API 키를 ChatOpenAI와 OpenAIEmbeddings 클래스를 사용할 때 openai\_api\_key 매개변수로 넘깁니다.
2. 파일 업로드
   * st.file\_uploader를 사용하여 사용자가 파일을 업로드할 수 있도록 합니다.
   * 업로드할 수 있는 파일의 확장자는 pdf, txt, docx로 지정합니다.
   * 업로드된 파일을 임베딩하고 vectorstore에 저장한 후, 이를 retriever로 변환하여 체인에서 사용합니다.
   * 이전과 같은 파일을 선택했을 때 임베딩 과정을 다시 하지 않도록 하기 위해 embed\_file 함수에 st.cache\_data 데코레이터를 추가하였습니다. ([st.cache\_data 공식 문서](https://docs.streamlit.io/develop/api-reference/caching-and-state/st.cache_data))
3. 채팅 기록
   * 채팅 기록을 저장하기 위해 Session State를 사용합니다.
   * 솔루션에서는 st.session\_state["messages"]를 리스트로 초기화하고, 메시지를 추가하는 방법으로 구현했습니다. (save\_message 함수 참고)
   * 저장된 채팅 기록을 페이지에 출력하기 위해 st.session\_state["messages"] 리스트에 있는 메시지들을 하나씩 출력합니다. (paint\_history 함수 참고)
4. 결론

* 이전 과제에서 구현한 RAG 파이프라인을 Streamlit을 활용하여 재구현하면서, Streamlit 사용에 익숙해지고 지난 과제 내용을 효과적으로 복습할 수 있었습니다.