DU-도전학기 결과보고서

과제명	AI기반 검색 증강 생성(Retrieval-Augmented Generation) 기술을 이용한 대구대학교 학칙 데이터 분석 및 응답 시스템 개발		
참여자	성명	소속	학번
		AI학과	
		AI학과	
지도교수 의견	이번 프로젝트는 학칙 데이터 분석 및 자동 응답 시스템 개발에 FastAPI를 활용해 RAG 시스템과 앱을 성공적으로 연동했으며, 사용자 질문을 처리하고 응답을 제공하는 어플리케이션 구현도 원활하게 이루어졌다. 최종적으로 UI를 채팅형으로 전환하고 멀티 질문을 안정적으로 처리하는 기능을 더해 시스템의 실용성을 높였다. 도전학기 교육목표에 부합되는 활동으로 학점 부여 및 경쟁력 확보에 기여하였음을 확인하였다. 또한, 이번 프로젝트는 AI 기술을 활용한 데이터 분석과 응답 시스템 개발의 가능성을 확인했으며, 향후 추가적인 성능 향상을 통해 더욱 발전할 가능성이 기대된다. (소속) AI학부		

1. 도전 과제 내용

프로젝트: RAG 기반 대구대학교 학칙 데이터 응답 챗봇

대구대학교 학칙 데이터를 기반으로 검색 증강 생성(Retrieval-Augmented Generation) 기법을 적용하여 사용자 질문에 신뢰성 있는 답변을 제공하는 시스템을 개발하여 이를 Flutter를 통해 사용자가 편리하게 이용할 수 있도록 개발하는 것이 목표이다.

2. 도전 과제 수행 결과 및 성과

① 학칙 데이터 수집 및 전처리 과정

대구대학교 학칙 및 학생 행정 규정 데이터를 효과적으로 활용하기 위해 PDF 파일에서 텍스트를 추출하고 통합하는 작업을 수행하였습니다. PyPDFLoader 모듈을 활용하여 지정된 폴더 내 모든 PDF 파일을 불러온 후, 각 파일의 텍스트를 페이지 단위로 추출하여 하나의 Document 객체로 구성하였습니다. 이렇게 구성된 Document 객체에는 각 파일의 출처 정보와 모든 텍스트 내용이 포함되었으며, 이를 documents 리스트에 저장하여 관리하였습니다. 마지막으로, 파일별 데이터가 정상적으로 수집되었는지 출력 결과를 통해 검증하였습니다.

이후, 데이터 검색 및 처리를 효율적으로 수행하기 위해 추출된 텍스트를 일정한 크기의 청크로 나누는 작업을 진행하였습니다. RecursiveCharacterTextSplitter를 활용하여 각 텍스트 청크의 크기를 1000자로 설정하였으며, 청크 간 자연스러운 연결성을 유지하기 위해 300자의 중첩 구간(Overlap)을 두었습니다. 이러한 설정을 통해 문서가 불연속적으로 나뉘는 것을 방지하고, 문맥 정보를 최대한 보존할 수 있도록 설계하였습니다. 또한, 각 청크에는 출처 정보와 원본 문서의 텍스트 위치를 포함하여 데이터의 신뢰성을 높였습니다. 이를 통해 대구대학교 학칙 데이터가 검색과 처리를 위한 준비를 완료하였으며, 신뢰성과 효율성을 겸비한 데이터 기반을 구축할 수 있었습니다.

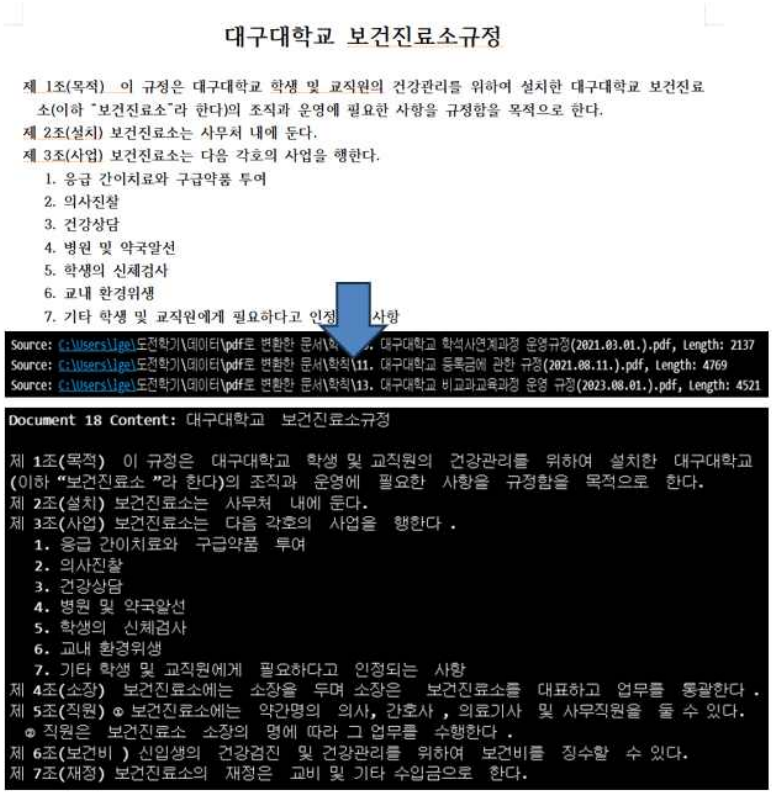

그림 1 출력 일부

② 데이터 저장 및 활용

전처리된 텍스트 데이터를 Chroma Vector DB에 저장하여 사용자 입력 질문에 대해 신속하고 효율적인 검색이 가능하도록 구성하였습니다. 텍스트 데이터의 전처리와 청킹 과정을 통해 대용량 데이터를 체계적으로 관리하고 검색 속도와 정확성을 개선하였으며, 학칙 및 행정 규정과 관련된 다양한 질문에 대해 신뢰성 높은 응답을 제공할 수 있도록 최적화하였습니다.

특히, 검색 결과의 품질과 다양성을 확보하기 위해 MMR(Minimal Marginal Relevance) 방식을 적용하였습니다. MMR은 사용자의 질문과 높은 연관성을 가지는 결과를 우선적

으로 제공하면서도 중복 정보를 최소화하는 기법으로, 프로젝트의 목적에 적합한 최적의 검색 환경을 구현하는 데 기여하였습니다. 이를 통해 검색 결과는 다양한 출처에서 제공되는 고유한 정보를 포함하며, 사용자 질문의 맥락에 부합하는 정확하고 유의미한 데이터를 반환할 수 있었습니다.

Chroma Vector DB와 MMR 방식을 결합한 본 시스템은 대구대학교 학칙 데이터를 기반으로 한 검색 체계의 신뢰성과 실효성을 높이는 데 성공하였으며, 데이터 검색 및 응답 생성의 정확성과 효율성을 증명하였습니다

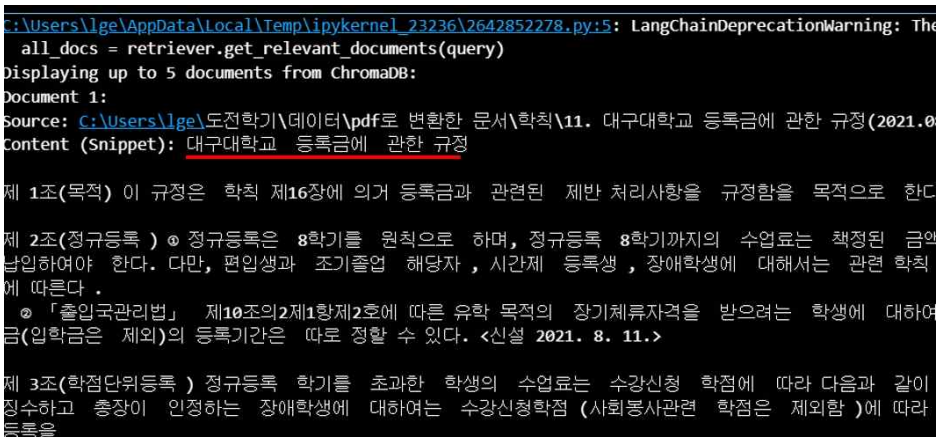

그림 2 등록금에 관해서 검색하였을 때 출력 결과

③ 최종 RAG기반 응답 시스템 개발

MMR(Minimal Marginal Relevance) 기법과 Chroma Vector DB를 결합하여 사용자 질문에 대한 신뢰성 높은 정보를 검색하고, OpenAI의 GPT-4 모델을 활용하여 문맥 기반의 정확한 응답을 제공하는 구조를 설계하였습니다.

응답 생성 단계에서는 OpenAI의 GPT-4 모델을 사용하여, 제공된 문맥에서만 답변을 생성하도록 프롬프트 템플릿을 설계하였습니다. 이를 통해 GPT-4 모델이 문맥에 포함되지 않은 정보를 생성하지 않도록 제한하여 환각 현상을 방지하였고, 질문에 대한 정확하고 신뢰성 있는 답변을 생성할 수 있었습니다. 또한, InMemoryChatMessageHistory를 활용해 사용자 대화 기록을 저장함으로써 연속적인 대화 흐름을 유지하고, 사용자 경험을 개선하였습니다.

```python
from langchain_core.chat_history import InMemoryChatMessageHistory
from langchain_core.output_parsers import StrOutputParser
from langchain_core.prompts import ChatPromptTemplate
from langchain_core.runnables.history import RunnableWithMessageHistory
from langchain_openai import ChatOpenAI
from langchain_community.vectorstores.chroma import Chroma
from langchain_openai import OpenAIEmbeddings

# ChromaDB 벡터 저장소 로드 및 MMR 설정
vectorstore = Chroma.from_documents(documents=split_docs, embedding=OpenAIEmbeddings(openai_api_key=api_key))
retriever = vectorstore.as_retriever(search_type="mmr")

# 프롬프트 템플릿 정의
template = """
당신은 사용자에게 도움이 되는 학칙기반 응답 어시스턴트입니다.
질문에 답하기 위해, 제공된 맥락의 모든 부분을 주의 깊게 고려해야 합니다.
맥락에 명시되지 않은 어떠한 정보도 답변에 포함시키지 마세요.
제공된 문맥에서 답변할 수 없는 경우 '해당 문서에서는 답을 찾을 수 없습니다'라고 답하세요.
만약 해당 내용이 존재한다면 그 내용을 언급하며 알려줘.
Answer the question based only on the following context:
{context}

Question: {question}
"""

# 프롬프트 구조
prompt = ChatPromptTemplate.from_template(template)

# 인메모리 채팅 기록 저장소
history = InMemoryChatMessageHistory()

def get_history():
    return history

# 체인 생성 (MMR과 프롬프트 기반)
chain = prompt | ChatOpenAI(model_name="gpt-4", temperature=0, openai_api_key=api_key) | StrOutputParser()

wrapped_chain = RunnableWithMessageHistory(
    chain,
```

```python
question = "조기 졸업에 관한 규정 알려주세요"
k = 3  # 출력할 관련 문서의 개수 설정
answer = ask_question_with_history_and_retrieve_k(question, k)
print(f"Answer: {answer}")
```

```
Answer: 조기 졸업에 관한 규정은 다음과 같습니다.

1. 5학기 이상을 이수한 자로서 조기졸업을 희망하는 자는 조기졸업 하는 학기 개강 후 3주 이내에 조기졸업 신청

2. 조기졸업을 할 수 있는 자는 다음 각 호의 요건을 모두 갖추어야 합니다.
   - 학업성적 총평점평균이 4.20이상이어야 합니다. 다만, 학.석사 연계과정자는 학·석사연계과정운영에관한내규
   - 6학기 이상 이수하여야 합니다.
   - 대학별 졸업 이수학점 이상을 이수하여야 합니다.
   - 졸업논문에 합격하여야 합니다.
   - 교육과정별 졸업사정기준에 도달하여야 합니다.
```

그림 4 RAG기반 응답시스템 질문 및 답변구현

④ RAGAS 평가

1	ave;new 본거지 어디야	[본디 8장이 더 있었으나 현재 분실된 상태이다. 북유럽 신화 연구의 중요한 1차 자료이며, 이 필사본에방에 존재하지 않는 고대 시가도 상당하다. 1643년 이전의 이 필사본의 행방에 대해서는 아무 것도 알려진 바가 없다. 1643년에 당시 스칼홀트 주교 브라뇰푸르 스베인손이 이 문서들 입수하여, 1662년에 덴마크의 프레데리크 3세에게 진상했다. 이런 전차로 '왕의 서'라고 불리게 되었다. 이후 문서는 1971년 4월 21일까지 코펜하겐 왕립도서관에 소장되어 있었고, 이후 레이캬비크로 반환되어 현재 아르니 마그누손 연구소에서 보관중이다. 이런 보물을 운반하기에 항공편은 미덥지 못하다고 여겨, 군대의 호위들 받으면서 선박 편으로 이송되었다.]	아르니 마그누손 연구소에서 보관중이다.	도쿄 치요다구	0.794838	0.000000	0.0	0.0
2	사카이 다다요시의 아버지가 누구지	[사카이 다다요시(일본어: 酒井忠休, 1714년 9월 24일 ~ 1787년 6월 3일)는 일본 에도 시대의 다이묘로, 데와노쿠니 마쓰야마 번의 3대 번주이다. 어릴적 이름은 헤이지로(平次郎)이며, 관위는 종5위하, 야마시로노카미, 이와미노카미이다. 쇼토쿠 4년(1714년) 8월 16일, 쇼나이 번사로 사카이 이에쓰구의 후손인 사카이 나오타카(酒井直隆)의 아들로 태어났다. 교호 17년(1732년) 7월에 마쓰야마 번의 2대 번주인 사카이 다다아스의 친아들인 사카이 다다히데가 폐적되었기 때문에, 그해 8월 29일에 다다아스의 양자가 되었으며, 같은해 11월 25일 다다아스가 은거하면서 번주직을 계승하였다. 번 통치 개혁의 일환으로 가리아게 제(借上制)을 시행하였으나, 양자로서 번주가 된 다다요시에 대한 가신단의 반발로 인해 실패하고, 막부로부터 3천 냥을 빌려서 곤궁함을 면하였다. 호레키 10년(1760년) 8월, 와카도시요리에 임명되었고, 안에이 8년(1779년) 12월 15일에 고즈케 국 내 영지 5천 석을 추가로 받아, 번 영지는 2만 5천 석이 되었다. 덴메이 7년(1787년) 4월 18일, 74세로 사망하였다.]	사카이 다다요시의 아버지는 사카이 나오타카(酒井直隆)입니다.	사카이 나오타카(酒井直隆)	0.796111	1.000000	1.0	1.0
3	일반성면의 면적이 얼마야	[일반성면은 동부 5개 면의 교통, 문화, 교육, 상업의 중심지로서 일찍부터 상업이 발달한 곳으로 날로 번성 재래시장이 번성하고 있는 고장이다. 면적은 19.41 km²로 전주시 16개 읍면동 중 가장 적은 면이지만 인근에 경남 산림환경연구원이 있어 그곳을 찾는 관광객이 이곳 일반성면을 지나간다. 2012년 1월 1일 기준으로 인구 3,233명 (남 : 1,556명, 여 : 1,677명) 1,413세대로 6개 법정리 19개 자연마을 31개 반으로 구성되어 있다. [1]]	일반성면의 면적은 19.41 km²입니다.	19.41 km²	0.912862	1.000000	1.0	1.0
		[구족을 돈목하게 하고 사색의 당파들 평등하게 기용하였으며, 요행의 문로를 막고 언론의						

그림 5 평가구조에 맞게 전처리한 평가 데이터셋 일부 출력

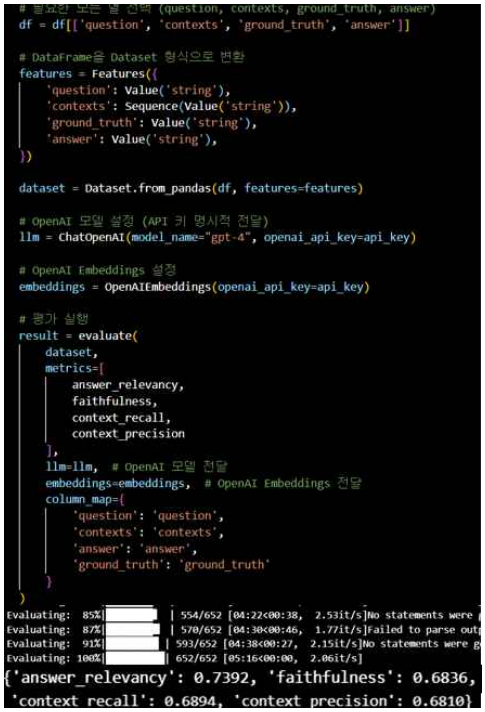

그림 6 RAGAS 평가

전처리된 AI허브 WIKI 데이터셋을 활용하여 RAGAS 평가를 실행하였습니다. RAGAS 평가는 OpenAI의 GPT-4 모델을 활용하여 데이터셋에 포함된 질문에 대한 응답을 생성한 후, 생성된 응답과 데이터셋에 정의된 문맥 및 정답(ground_truth)을 비교하는 방식으로 진행되었습니다.
평가 과정에서는 다음과 같은 주요 지표를 설정하여 시스템의 성능을 다각적으로 측정하였습니다:
- Faithfulness(정확성): 생성된 응답이 데이터셋의 정답과 얼마나 일치하는지를 평가.
- Answer Relevancy(응답 적합성): 사용자의 질문과 생성된 응답 간의 관련성을 평가.
- Context Precision(문맥 정밀도): 응답이 검색된 문맥과 관련된 중요한 정보를 포함하고 있는지 평가.
- Context Recall(문맥 재현율): 필요한 정보를 얼마나 완전하게 검색 및 응답에 반영했는지 평가.
위 RAGAS 평가를 통해 RAG 시스템의 성능을 객관적으로 검증할 수 있음을 입증하였습니다.

FastAPI와 Flutter를 활용하여 학칙 데이터를 효율적으로 처리하고 사용자에게 직관적인 인터페이스를 제공하는 시스템을 구현하였습니다.

FastAPI는 백엔드에서 대규모 학칙 데이터를 처리하고, ChromaDB와 연동하여 신속하고

정확한 검색 결과를 제공합니다. FastAPI는 사용자의 질문을 실시간으로 받아들이고, 이를 ChromaDB에 전달하여 관련 문맥을 검색한 뒤 GPT-4 모델을 통해 최적의 답변을 생성하도록 설계 하였습니다. 이 과정에서 FastAPI의 비동기 처리 기능을 활용하여 응답 시간을 최소화하고 시스템의 안정성을 높이려고 하였습니다.

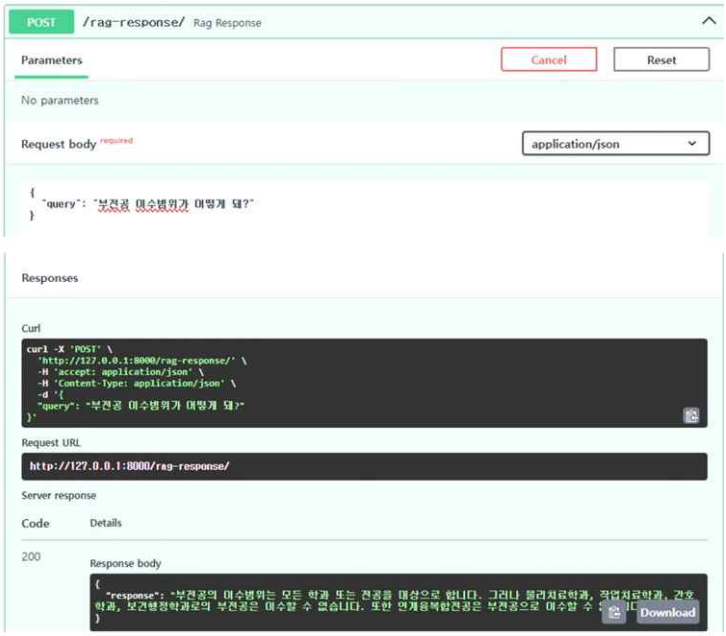

그림 7 FastAPI 연동 후 질문하였을 때 질의응답 결과

Flutter는 이러한 FastAPI 백엔드와 연동되어 사용자 인터페이스를 구현하는 데 활용되었다. Flutter 기반 UI는 사용자들이 이용할 때 편리하고 직관적일 수 있도록 사용하는 것에 초점을 맞췄다. 사용자는 Flutter 앱에서 질문을 입력하면, FastAPI가 이를 처리해 생성한 답변을 앱 화면에 표시합니다.

이 시스템은 FastAPI와 Flutter의 연동을 통해 백엔드와 프론트엔드 간의 효율적인 상호작용을 할 수 있도록 구현하였습니다. FastAPI는 안정적이고 빠른 데이터 처리와 검색 기능을 제공하여 사용자 요청에 신속히 응답하였으며, Flutter는 이러한 데이터를 직관적이고 일관된 방식으로 시각화하여 사용자 경험을 개선하였습니다.
(Flutter 구현 모습은 4번 최종결과물을 참고하여 주시기 바랍니다.)

또한, FastAPI에서 응답 시간을 측정하는 코드를 추가하여 각 질문 처리에 걸리는 시간을 분석할 수 있도록 하였습니다. 이를 통해 시스템의 성능을 확인하고 답변이 나오는 데 걸리는 시간을 확인할 수 있습니다.

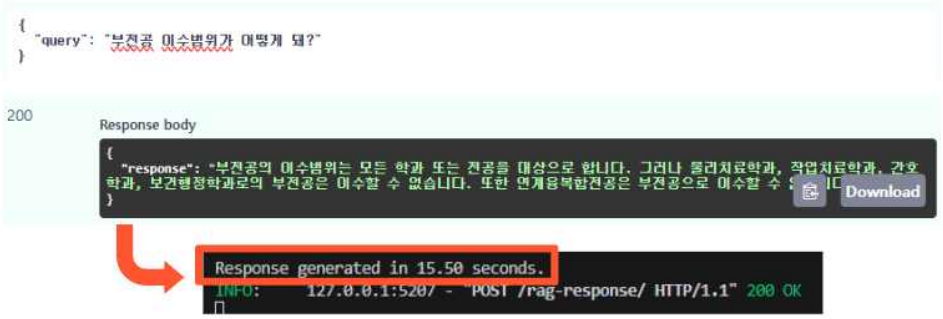

그림 8 질문하였을 때 걸린 응답 시간 확인한 모습

3. 자기 평가

: 이번 도전학기 프로젝트를 하며 FastAPI에 대해 조금 더 알아가고, Flutter 구현 능력을 더욱 향상시키는 것이 목표였습니다. FastAPI를 통해 백엔드를 구성하면서 API 설계의 중요성과 데이터 처리의 신속성을 체감할 수 있었고, Flutter를 활용한 앱 개발에서는 UI/UX 설계와 기술 구현의 균형을 맞추는 것이 얼마나 중요한지 깨닫게 되었습니다. 이 과정에서 팀원들과 협업하며 문제를 해결하고 새로운 기술을 습득하는 기회가 되었습니다. 다만, 처음 시도하는 기술들을 단기간에 익히고 적용하다보니 적용하고 싶었던 기술을 이용하지 못하였다는 아쉬운 점과 미흡함 점이 있었습니다. 이러한 경험을 바탕으로 앞으로의 프로젝트에서는 더욱 체계적인 계획을 수립하고 이용할 것입니다. 향후 목표로는 프로젝트에서 익힌 기술들을 더욱 심화하고, 다양한 프로젝트에 응용하여 실력 향상 시키는 것입니다. 특히, FastAPI와 Flutter의 심화 기능을 학습하고, 다른 모델 및 기술과의 연동 가능성을 탐구하여 더욱 실용적이고 창의적인 서비스를 개발하고자 합니다. 이번 도전학기 프로젝트는 이러한 목표를 향해 나아가는 데 중요한 발판이 되었다고 생각합니다.

:이번 도전학기 프로젝트를 통해 대구대학교 학칙 데이터를 기반으로 한 RAG 응답 시스템을 개발하며 많은 것을 배우고 성장할 수 있었습니다. 처음에는 데이터 전처리와 검색 시스템 설계 과정이 익숙하지 않아 시행착오가 많았지만, PyPDFLoader를 활용한 텍스트 추출과 RecursiveCharacterTextSplitter를 사용한 텍스트 청킹 과정을 하나씩 해결해 나가며 데이터를 체계적으로 관리하는 방법을 익힐 수 있었습니다. 이런 과정에서 데이터의 신뢰성을 유지하면서도 효율적으로 검색하고 응답을 생성하는 데 필요한 기본기를 다질 수 있었습니다.
RAG 기술을 직접 구현해 보며, 검색 단계와 생성 단계가 어떻게 유기적으로 결합되어 사용자에게 신뢰성 있는 정보를 제공할 수 있는지를 깊이 이해할 수 있었습니다. 특히 Chroma Vector DB를 활용해 데이터의 유사도를 분석하고, MMR(Minimal Marginal Relevance) 기법을 적용하여 중복 정보를 줄이면서도 다양한 정보를 제공할 수 있었던 점이 흥미로웠습니다. 이 과정에서 사용자의 질문과 가장 관련성이 높은 정보를 반환하는 것이 얼마나 중요한지, 그리고 그것을 구현하기 위해 어떤

기법이 필요한지를 체감할 수 있었습니다.
RAGAS 평가를 진행하며, Faithfulness, Answer Relevancy, Context Precision, Context Recall과 같은 지표를 통해 제가 설계한 시스템의 강점과 한계를 확인할 수 있었습니다. 데이터를 기반으로 시스템의 성능을 객관적으로 평가하고, 부족한 부분을 개선해야 한다는 것을 알게 되었고, 이를 통해 앞으로 어떤 방향으로 나아가야 할지 구체적인 계획도 세울 수 있었습니다. 특히, Faithfulness와 Context Precision에서 더 나아지기 위해 어떤 개선이 필요한지 스스로 고민하게 된 점이 큰 배움이었습니다.
이번 프로젝트는 제가 AI 기술을 단순히 배우는 것을 넘어, 직접 설계하고 구현하며 문제를 해결하는 과정에서 제 스스로 성장할 수 있는 기회가 되었습니다. 처음에는 어려움도 많았지만, 하나씩 해결해 나가는 과정에서 점점 자신감이 생겼고, 제가 배운 것을 실질적으로 활용할 수 있는 능력을 키울 수 있었습니다. 앞으로도 이번 경험을 바탕으로 더 신뢰성 있고 유용한 시스템을 만들어 보고 싶습니다. 무엇보다 이번 프로젝트를 통해 얻은 경험과 배움은 제게 큰 자산이 되었다고 생각합니다.

4. 최종 결과물

해당 프로그램은 사용자가 학칙 데이터에 대해 모르는 것이 있을 때 관련 질문을 물어보면 RAG 기술을 이용한 학칙 데이터를 기반으로 하여 답변을 주는 챗봇입니다.
FastAPI 와 기술을 연동하여 Flutter UI를 통해 사용자가 채팅을 통해 응답을 받아 볼 수 있습니다.

그림 9 Flutter로 구현한 최종 UI 모습