데이터 조회 프로그램

SKN Family AI Camp 14기 : 최종 프로젝트 1팀

프로젝트 주제	LLM 활용 내부 고객 업무 효율성 향상을 위한 구글 API 전문 개발자 지원 AI 기반 문서 검색 시스템
원본 데이터 **(텍스트)**	구글 **API** 공식 문서 (Google for developers) - OAuth 2.0 / Google Identity - People API - Google Drive API - Google Sheets API - Gmail API - API Reference - YouTube Data API - Google Maps - Firestore REST API - Firebase Authentication - BigQuery API 기업 내부 문서 **:** OpenAI 합성 데이터셋(자체 생성)
데이터 전처리 과정	**[**구글 **API** 공식 문서**]** OPEN AI API를 사용해서, 구글 API 문서 데이터를 QA셋(jsonl파일)로 변환하였습니다. 구글 API 문서 데이터 원문을 벡터 DB에 넣게 되었을 때, 코드나 문맥이 잘리면서 데이터의 품질이 떨어질 수 있기 때문에 QA셋으로 가공을 하기로 결정하였습니다. 원문 데이터를 벡터 DB에 넣는 작업도 따로 진행 중이므로 QA셋과 원문 데이터를 각각 벡터 DB에 넣은 후 검색 성능을 비교해볼 예정입니다. (1) 각 문서 1개당 최대 10개의 Q&A셋을 생성 ● OPEN AI API를 통해 구글 API 문서 데이터(txt 파일)을 순회 ● 각 문서 1개당 최대 10개의 Q&A셋을 생성하고, 생성된 Q&A셋을 JSONL 파일로 저장 ➔ 문서 길이 등을 고려하지 않고, 한 문서당 최대 10개의 Q&A셋을 만들다 보니, 문서 길이가 긴 경우 문서의 핵심 정보가 Q&A 생성 과정에서 누락되는 문제가 발생했습니다. (2) 페어 단위 **Q&A**셋 생성 - 최종 선택 방식 ● 청크 분할 및 오버랩 적용: 위 문제 해결을 위해, 긴 문서도 정보 손실 없이 처리할 수 있도록 청크(Chunk) 단위로 분할하고 경계 영역 정보를 보존하기 위해 오버랩(Overlap)을 적용했습니다. ● 정보 누락 최소화 및 정확성 확보: 페어 단위로 처리하여 청크 간 중복을 검토하고,질문-답변 생성이 불필요한 페어는 저장하지 않음으로써 최종 결과물의 정확성과 유용성을 확보했습니다.
DB 사용 용도	**[VectorDB - Chroma]** API 문서 및 내부 문서의 임베딩 벡터 저장 RAG(검색 증강 생성) 시스템의 핵심 요소로, 사용자 질문에 가장 관련성 높은 문서를 찾아 LLM에 제공하는 역할 유사도 검색 지원 사내 내부 문서 권한 기반 필터링 적용 구글 api 주제에 대한 메타 필터링 적용 **[MySQL (**관계형 **DB)]**

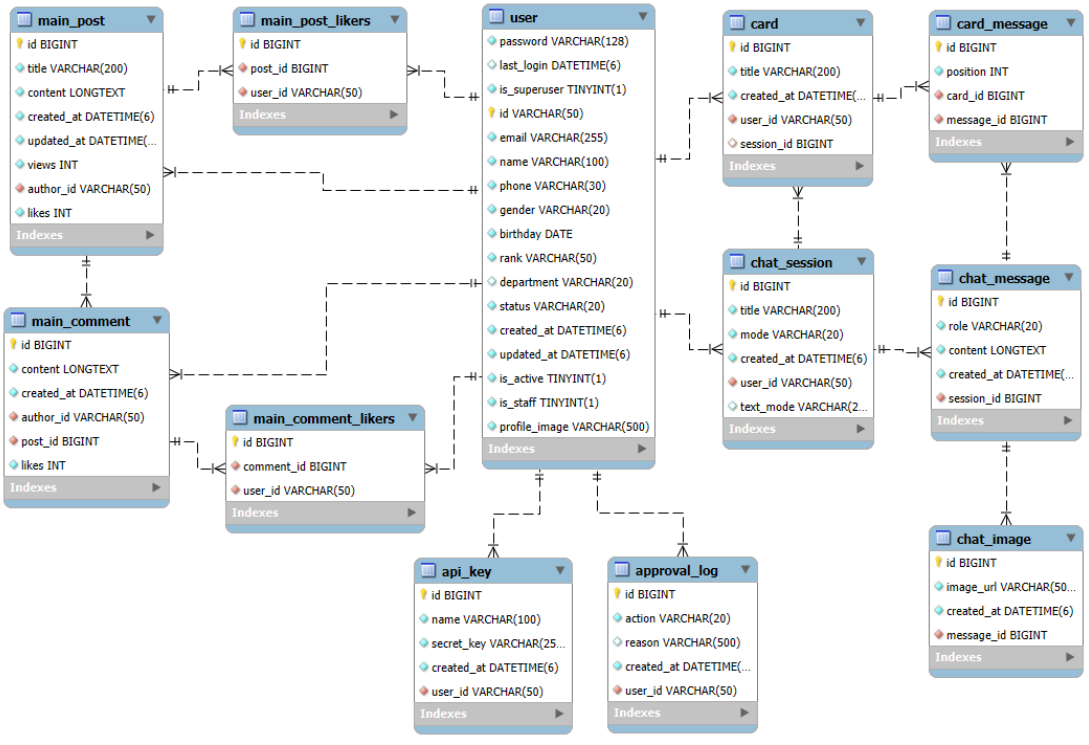
	사용자 계정, 직급, 권한 관리 (sLLM(Qwen3-8B)의 권한 기반 필터링에 사용) 채팅 저장 및 관리 (GPT-4o 및 sLLM과의 전체 대화 내역) 커뮤니티 Q&A, 대화 내역 카드 저장, API 키 관리
사용 데이터	외부 데이터: Google Developers 공식 문서 • OAuth 2.0 / Google Identity • People API • Google Drive API • Google Sheets API • Gmail API • YouTube Data API • Google Maps • Firestore REST API • Firebase Authentication • BigQuery API 내부 데이터: 기업 내부 문서 ~ 4개의 팀별로 각각 15개의 문서 RAG용 전처리 데이터: Google Developers 공식 문서 QA
Vector DB 구조	**[**구글 **API** 원문**]** [1] id: 6788a274-a664-4466-9c89-191c48191303 문서: Q: BigQuery에서 새로운 데이터셋을 생성하는 방법은 무엇인가요? A: 새로운 데이터셋을 생성하려면 다음의 API 메서드를 사용합니다: `insert` 메서드. 요청은 다음과 같이 구성됩니다: ``` POST /bigquery/v2/projects/{projectId}/datasets ``` 여기서 `{projectId}`는 데이터셋을 생성할 프로젝트의 ID입니다. 메타데이터: {'last_verified': '2025-08-19', 'source_file': 'bigquery_docs_reference_rest.txt', 'source': '["https://cloud.google.com/bigquery/docs/reference/rest"]', 'tags': 'bigquery'} **[**구글 **API** 공식 문서 **QA]** [2] id=58ef51d2-61f3-40db-ae23-81e6a7e0b1ad similarity=72.1% (distance=0.2791) Q: Google의 CardDAV 프로토콜을 사용하여 연락처를 삽입하는 방법은 무엇인가요? A: 클라이언트 애플리케이션은 VCard 3.0 형식의 새 연락처가 포함된 POST 요청을 실행하여 연락처를 삽입할 수 있습니다. 응답에는 새 연락처의 ID가 포함됩니다. -> sources: ["https://developers.google.com/people/carddav?hl=ko#synchronizing_contacts"] -> tags: people -> last_verified: 2025-08-19 -> source_file: people_v1_troubleshoot-authentication-authorization.txt [3] id=cf76f249-b1bd-4233-92ad-337210e2278f similarity=71.4% (distance=0.2862) Q: Google의 CardDAV API를 사용하여 새 연락처를 생성하는 방법은 무엇인가요? A: 클라이언트 애플리케이션은 VCard 3.0 형식의 새 연락처가 포함된 POST 요청을 실행하여 새 연락처를 생성할 수 있습니다. 응답에는 새 연락처의 ID가 포함됩니다. -> sources: ["https://developers.google.com/people/carddav"] -> tags: people -> last_verified: 2025-08-19 -> source_file: people_v1_troubleshoot-authentication-authorization.txt **[**사내 내부 문서**]**

```
(codenova) ➜  google-api-auto-crawer git:(crawer) ✗ python chroma_rest_select.py
쿼리 : 백엔드
임계값 : 0.1, 최대개수 : 3
```

문서	유사도 점수

<!-- 회사 : 코드노바 | 대상 : 사원(백엔드) | 작성일 : 2025-08-29 -->
# 백엔드팀 주간 업무 계획 	
 	
작성일: 2025-08-29
작성자: [작성자 이름]
팀: 백엔드팀
 	
--- 	
 	
## 1. 주간 목표 	
 	
- 서비스 안정성 강화
- 신규 기능 개발 및 배포
- 기술 부채 관리 및 리팩토링
 	
## 2. 주요 업무 	
 	
### 2.1 서비스 안정성 강화 	
- **서버 모니터링**:
  - [ ] 로그 및 오류 모니터링 도구 점검
  - [ ] 주요 지표(응답 시간, 오류율) 확인
  - [ ] 이상 징후 발견 시 즉각 대응 방안 마련
 	
### 2.2 신규 기능 개발 및 배포 	
- **기능 A 개발**:
  - [ ] 요구사항 정의서 작성 완료
  - [ ] API 설계 문서 작성
  - [ ] 개발 일정 수립 및 팀원 배정
  - [ ] 기능 구현 후 코드 리뷰 및 테스트 진행
  - [ ] 배포 일정 조율 및 배포 준비
	0.448

Django 내부
RDB 구조



RDB 조회

```
-- 2. 가입이 '승인(approved)'된 사용자만 조회
SELECT id, name, email, status FROM user WHERE status = 'approved';
```

2. 가입이 '승인(approved)'된 사용자만 조회

id	name	email	status
aaaa	안윤지	aa@aa.com	approved
admin	Admin User	admin@example.com	approved
anda	이원지희	can@naver.com	approved
car_	이원지희	can@naver.com	approved
Ella	Ella Lee	can@naver.com	approved
frontend	Jungi Kim	skn14f1@gmail.com	approved
guest	송유나	yuna@email.com	approved
iiiiiiiii	김의령	my_cookies_@naver.com	approved
kan_	이원지희	can@naver.com	approved
kjw0902	xxxx	asiudb@naver.com	approved
lee_	이원지희	can@naver.com	approved
miny0913	정민영	miny0913@naver.com	approved
rwr9857	아무개	skn14f1@gmail.com	approved
rwr99999	김준기	admin@gmail.com	approved
skn14-1	codenova	topteam@naver.com	approved

```
-- 2. 사용자의 가입 승인/반려 기록 조회
SELECT user_id, action, reason, created_at FROM approval_log;
```

2. 사용자의 가입 승인/반려 기록 조회

user_id	action	reason	created_at
lee_	approved	<null>	2025-09-19 02:35:08.515967
rwr9857	approved	<null>	2025-09-19 05:59:30.525610
guest	approved	<null>	2025-09-22 01:43:07.199109
jmy0913	rejected	<null>	2025-09-22 02:32:48.368863
aaaa	approved	<null>	2025-09-22 03:07:51.738854
jenny_	pending	<null>	2025-09-22 04:57:01.366374
anda	approved	당신은 외부인이야	2025-09-22 06:23:17.462527
car_	approved	<null>	2025-09-22 06:47:44.927928
kjw0902	pending	<null>	2025-09-22 07:10:23.146741
kan_	approved	외부인입니다.	2025-09-25 00:41:16.710169
rwr99999	approved	<null>	2025-09-25 06:42:04.640201
Ella	approved	<null>	2025-09-26 03:00:46.538865
iiiiiiiii	approved	<null>	2025-09-29 18:45:07.057478
frontend	approved	<null>	2025-09-30 07:37:11.440220
miny0913	approved	<null>	2025-09-30 08:05:14.186901

```sql
-- 3. 특정 게시물과 그 댓글들을 함께 조회
SELECT
    p.title AS '게시물 제목',
    c.content AS '댓글 내용',
    c.author_id AS '댓글 작성자'
FROM
    main_post p
JOIN
    main_comment c  ON p.id = c.post_id
WHERE
    p.id = 7;
```

출력 | 3. 특정 게시물(예: id=5)과 ...함께 조회 (수정된 테이블 이름 적용)

	`게시물 제목`	`댓글 내용`	`댓글 작성자`
1	연봉이 언제 오를까요???	이직하세요	rwr9857
2	연봉이 언제 오를까요???	공부하기	skn14-1
3	연봉이 언제 오를까요???	ㄹㄹㄹ	admin
4	연봉이 언제 오를까요???	ㅁㄴㅇㄹ	admin
5	연봉이 언제 오를까요???	흠	admin

```sql
-- 4. '좋아요(likes)'를 많이 받은 순서대로 게시물 정렬
SELECT title, content, likes, author_id FROM main_post ORDER BY likes DESC;
```

출력 | 4. '좋아요(likes)'를 많이 받은 순서대로 게시물 정렬

	title	content	likes	author_id
1	14	14	15	rwr9857
2	히히히히힣히히히힣히히히힣히	히힣힣히히히힣히히히힣히힣힣히히히힣히히히힣히힣힣히히히힣히히히힣히힣힣히히히힣히히히힣히힣힣히히히	11	kjw0902
3	관리자 계정 포스트 테스트용	1111111111111111111111	10	skn14-1
4	정짜장님 화이팅!!	응원의 메시지를 보냅니다.	10	lee_
5	연봉이 언제 오를까요???	연봉올리는법!ㅇㅇ	10	anda
6	바른말 고운말 사용합시다.!	여러분 우리 고운말 바른말 사용해요	10	lee_
7	상사 정짜장에대해 알려드립	상사 정짜장이 맨날 들이 닥침.... 조심하쇼	3	lee_
8	안녕	나 관리자야~~	2	admin
9	메인 챗봇이 잘 안나와요	open ai API 키 문제인지... 에러메시지만 나옵니다.처리 중 오류가 발생했습니다: Error code	1	rwr9857
10	22	22	1	rwr9857
11	20	20	1	rwr9857
12	12	12	1	rwr9857
13	안녕	ㄴㅇㄹ	1	lee_
14	13	13	0	rwr9857
15	11	11	0	rwr9857
16	15	15	0	rwr9857
17	19	19	0	rwr9857
18	10	10	0	rwr9857
19	아~ 오늘 점심 뭐먹지?	어디가지?	0	rwr9857