SK네트웍스 Family AI과정 14기

데이터 수집 및 저장 데이터 베이스 설계서

◘ 개요

- 산출물 단계 : 데이터 수집 및 저장
- 평가 산출물 : 데이터베이스 설계서
- 제출 일자 : 2025.10.01
- 깃허브 경로 : SKN14-Final-1Team

개요	• 소개 • 시스템 개요
구조	• 시스템 아키텍처 • 요구사항 매트릭스
설계	• 데이터 설계 • 설계 근거 • ERD

소개	**- RDB목적** - 본 데이터베이스는 사용자 가입/승인 흐름, 게시글 및 댓글 관리, 문서/모드별 챗 세션과 메시지, 즐겨찾기(카드), 그리고 사용자별 API 키 관리까지 포함한 AI 기반 문서 검색 시스템의 핵심 데이터를 저장·관리하기 위해 설계되었습니다. **- RDB범위** - 사용자 계정과 상태 관리, 승인 로그 이력 관리, 세션·메시지·첨부 이미지 보관, 게시글(Post) 및 댓글(Comment) 관리, 즐겨찾기 카드/카드&메시지 매핑, 선택적 API 키 발급 및 사용 권한 범위 관리 등을 포함합니다. **- Vector DB목적** - Google API 및 사내 내부 문서에 대한 RAG(검색 증강 생성) 기술을 적용하여, 사용자의 질문에 가장 정확하고 연관성 높은 정보를 제공하기 위해 설계되었습니다. **- Vector DB 범위** - GoogleAPI의 범위는 OAuth 2.0, YouTube Data API, People API, Maps Platform, Drive API, Sheets API, Gmail API, Calendar API . Firestore REST API ,Firestore API, Firebase Authentication, BigQuery API 총 11개로 구성되어 있습니다. - 사내 내부 문서는 OpenAI API를 활용하여 자체 생성한 합성 데이터셋(Synthetic Dataset)을 사용합니다. 이는 사용자의 직급 및 부서에 따른 문서 접근 권한 제어 기능을 테스트하기 위한 목적으로 구성되어 있습니다. **- Vector DB 내용** 1. 적용된 메타필터링의 목적: Google API 문서와 사내 내부 문서를 벡터화(ChromaDB 저장)한 뒤, 단순한 유사도 검색(RAG)만 하는 것이 아니라 사용자 질문의 종류(api 종류) / 사용자의 속성(직급, 부서, 권한 등)에 따라 접근 가능한 문서를 제어하기 위한 방식으로 메타필터링이 적용됩니다.

	2. 메타필터링이 적용되는 데이터 종류 **Google API 문서** (OAuth, YouTube Data API, People API, Maps, Drive, Sheets, Gmail, Calendar, Firestore, Firebase, BigQuery 등 총 11종) **사내 내부 문서** (Synthetic Dataset) → OpenAI API를 활용해 합성한 문서이며, **사용자 직급 및 부서 정보**를 기반으로 접근 권한이 제어됨 3. 메타필터링 구조 - 사내 내부 문서 벡터 db 메타 필터링 Django의 mysql RDB(User 테이블)의 rank (직급), department (부서) 존재. 이 값들이 메타정보로 활용되어 벡터 검색 시 필터 조건으로 활용 - 구글 api 문서(원문/QA) 벡터 DB 메타 필터링 사용자의 질문에서 api 종류를 추출해서, 이후 벡터 db 검색 시 메타 필터링으로 활용 - Vector DB(ChromaDB) [사내 내부 문서] 문서 임베딩 저장 시, 해당 문서가 속한 팀/부서/권한 정보를 메타데이터로 함께 저장. (role) 검색 시 metadata filter 를 걸어, 현재 사용자와 일치하지 않는 문서는 제외. [구글 api 문서] 문서 임베딩 저장 시, 해당 문서의 api 종류를 메타 데이터로 함께 저장(tags) 사용자의 질문에서 추출된 api 종류를 메타 필터링(tags)하여 검색
시스템 개요	**- 시스템 역할** - 가입 사용자의 승인 상태(pending/approved/rejected) 관리 - 채팅 모드별 ChatSession 생성 및 ChatMessage 기록 - 메시지에 첨부된 이미지(ChatImage) 관리 - 게시판 기능: 게시글(Post) 및 댓글(Comment) 작성·조회, 좋아요 관리 - 메시지 선택을 통한 카드(Card) 생성 및 즐겨찾기 관리 - 사용자별 API Key 생성·보관

	- API 전문 어시스턴트: OpenAI의 GPT-4o 모델을 기반으로, 구글 API 문서에 대한 질의응답, 코드 예시 제공, 이미지/음성을 통한 질문 기능을 지원합니다. - 사내 문서 전문 챗봇: Qwen3-8B 소형 언어 모델(sLLM)을 기반으로, 사내 문서에 대한 질의응답을 수행하며, 사용자의 선택에 따라 '공손 말투'와 '친구 말투'로 소통합니다. **- 주요 기능** - 사용자 인증 및 상태 변경 이력 보관(ApprovalLog) - 세션 단위의 대화 흐름 저장(ChatSession, ChatMessage) - 즐겨찾기 카드 관리(Card, CardMessage) - API 키 관리(ApiKey) - 게시글/댓글 관리 및 좋아요 기능(Post, Comment) - 챗팅 저장
시스템 아키텍처	**- 데이터베이스 구조** - 관계형 데이터베이스(RDBMS)로 설계되며, 핵심 테이블은 다음과 같습니다. - user : 사용자 기본 정보/조직 정보/상태 - approval_log : 가입 승인/거절 이력 - chat_session : 세션 메타(제목/모드/소유자) - chat_message : 세션 내 메시지(역할/본문/시각) - chat_image : 메시지에 첨부된 이미지 - card : 즐겨찾기 카드(사용자 소유, 선택적으로 세션 연결) - card_message : 카드-메시지 매핑 및 순서 - api_key : 사용자별 API Key와 권한 범위 - post : 게시글 정보(제목/내용/작성자/조회수/좋아요) - post_likers : 게시글 좋아요 관계 - comment : 댓글 정보(내용/작성자/좋아요) - comment_likers : 댓글 좋아요 관계 **- 적용한 데이터베이스** -사내 내부 문서의 경우 chromaDB에 벡터화 하여 저장함

요구사항 매트릭스

요구사항 및 관련 테이블

요구사항	관련 테이블
사용자 계정 및 상태 관리	user
가입 승인/반려 이력 저장	approval_log
모드별 챗 세션 관리	chat_session
세션 내 메시지 저장	chat_message
메시지 첨부 이미지 관리	chat_image
즐겨찾기 카드 생성/조회	card
카드에 포함된 메시지와 순서	card_message
사용자별 API 키 관리	api_key
게시글 관리	post
댓글 관리	comment
게시글 좋아요 관리	post_likers
댓글 좋아요 관리	comment_likers

데이터 설계

- 테이블 설명

1) user : 사용자 계정 정보

컬럼명	데이터 타입	설명	제약
id	VARCHAR(50)	사용자 (PK), 로그인 ID	PK, NOT NULL
email	VARCHAR(255)	이메일	NOT NULL
password	VARCHAR(255)	해시 비밀번호	NOT NULL
name	VARCHAR(100)	이름	NOT NULL
phone	VARCHAR(30)	전화번호	NOT NULL
gender	VARCHAR(20)	성별	NOT NULL
birthday	DATE	생년월일	NOT NULL
rank	VARCHAR(50)	직급	NOT NULL
department	VARCHAR(100)	부서	NULL

status	ENUM('pending','approved', 'rejected')	가입 상태	NOT NULL, DEFAULT 'pending'
last_login	DATETIME	최근 로그인 시각	NULL
created_at	DATETIME	생성 시각	NOT NULL
updated_at	DATETIME	수정 시각	NOT NULL
is_active	BOOLEAN	계정 활성 여부	DEFAULT TRUE
is_staff	BOOLEAN	스태프 여부	DEFAULT FALSE
profile_image	VARCHAR(500)	프로필 이미지 URL	DEFAULT 기본 이미지

2) approval_log : 계정 승인/거절 기록

컬럼명	데이터 타입	설명	제약
id	INT	로그 PK	PK, AUTO INCREMENT
user_id	VARCHAR(50)	대상 사용자	FK → user.id, NOT NULL, ON DELETE CASCADE
action	ENUM('pending', 'approved', 'rejected')	처리 종류	NOT NULL
reason	VARCHAR(500)	사유	NULL
created_at	DATETIME	기록 시각	NOT NULL

3) chat_session : 채팅 세션

컬럼명	데이터 타입	설명	제약
id	INT	세션 PK	PK, AUTO INCREMENT
user_id	INT	세션 소유자	FK → user.id, NOT NULL, ON DELETE CASCADE
title	VARCHAR(200)	세션 제목	NOT NULL
mode	ENUM('api','internal')	세션 모드	NOT NULL
text_mode	ENUM('formal','informal')	말투 모드	NULL
created_at	DATETIME	생성 시각	NOT NULL

4) chat_message : 채팅 메시지

컬럼명	데이터 타입	설명	제약
id	INT	메시지 PK	PK, AUTO INCREMENT
session_id	INT	소속 세션	FK → chat_session.id, NOT NULL, ON DELETE CASCADE
role	ENUM('user','assistant','tool_calls','tool_responses')	role	NOT NULL
content	TEXT	본문	NOT NULL
created_at	DATETIME	생성 시각	NOT NULL

5) chat_image : 메시지 첨부 이미지

컬럼명	데이터 타입	설명	제약
id	INT	카드 PK	PK, AUTO INCREMENT
message_id	INT	연결된 메시지	FK → chat_message.id, NOT NULL
image_url	VARCHAR(500)	이미지 URL	NOT NULL
created_at	DATETIME	생성 시각	NOT NULL

6) card : 즐겨찾기 카드

컬럼명	데이터 타입	설명	제약
id	INT	카드 PK	PK, AUTO INCREMENT
user_id	VARCHAR(50)	카드 소유자	FK → user.id, NOT NULL, ON DELETE CASCADE
session_id	INT	관련 세션	FK → chat_session.id, NULL, ON DELETE CASCADE
title	VARCHAR(200)	카드 제목	NOT NULL
created_at	DATETIME	생성 시각	NOT NULL

7) card_message : 카드-메시지 매핑

컬럼명	데이터 타입	설명	제약
id	INT	매핑 PK	PK, AUTO INCREMENT
card_id	INT	대상 카드	FK → card.id, NOT NULL, ON DELETE CASCADE
message_id	INT	대상 메시지	FK → chat_message.id, NOT NULL, ON DELETE CASCADE
position	INT	카드 내 순서	NOT NULL

8) api_key : API 키 관리

컬럼명	데이터 타입	설명	제약
id	INT	키 PK	PK, AUTO INCREMENT
user_id	VARCHAR(50)	키 소유자	FK → user.id, NOT NULL, ON DELETE CASCADE
name	VARCHAR(100)	키 별칭	NOT NULL
secret_key	VARCHAR(255)	시크릿 키	NOT NULL
created_at	DATETIME	생성 시각	NOT NULL

9) post : 게시물

컬럼명	데이터 타입	설명	제약
id	INT	게시물의 고유 식별자	Primary Key
title	VARCHAR(200)	게시물의 제목	NOT NULL
content	TEXT	게시물의 내용	NOT NULL
author_id	VARCHAR(50)	작성자의 ID	Foreign Key, NOT NULL

created_at	DATETIME	생성 시각	NOT NULL
updated_at	DATETIME	수정 시각	NOT NULL
views	INT	조회수	Not Null, Default: 0
likes	INT	좋아요 수	Not Null, Default: 0

9-1) post_likers : 게시물 좋아요

컬럼명	데이터 타입	설명	제약
id	INT	고유 식별자	Primary Key
post_id	INT	게시물의 ID	FK → post.id, NOT NULL
user_id	VARCHAR(50)	사용자의 ID	FK → user.id, NOT NULL

10) comment : 댓글

컬럼명	데이터 타입	설명	제약
id	INT	댓글의 고유 식별자	Primary Key
post_id	INT	게시물의 ID	FK → post.id, NOT NULL
author_id	VARCHAR(50)	작성자의 ID	FK → user.id, NOT NULL
content	TEXT	댓글 내용	Not Null
created_at	DATETIME	생성 시각	Not Null
likes	INT	좋아요 수	Not Null, Default: 0

10-1) comment_likers : 댓글 좋아요

컬럼명	데이터 타입	설명	제약
id	INT	관계의 고유 식별자	Primary Key
comment_id	INT	'좋아요'를 받은 댓글 ID	FK → comment.id, NOT NULL
user_id	VARCHAR(50)	'좋아요'를 누른 사용자 ID	FK → user.id, NOT NULL

설계 근거

- **데이터 무결성 보장**
  - 외래 키(FK)를 활용하여 테이블 간 관계를 명확히 정의함.
  - 삭제 전파는 기본적으로 CASCADE로 설정하여 참조 무결성 위반 레코드 방지

ERD