

# 종합프로젝트 기획안

2025년 6월 05일

과정명 : **LG**유플러스 유레카 백엔드 개발자

팀명	1조 : 눈 떠보니 유플러스 개발자?!
팀원	조장: 권영태 조원: 김원석, 김희진, 박기정, 이가인, 조윤주, 한현우, 황지연
프로젝트 주제	유플: 개인 맞춤 통신 요금제 추천 서비스

## 프로젝트 목적

## 가. 프로젝트 배경

가.휴대폰 용도별·사업자별 회선 수

(단위:회선)

구분		2023.7월	2024.6월
1-1-1. 휴대폰	고객용	SKT	23,164,492
		KT	13,619,689
		LGU+	11,046,604
		MVNO	8,231,441
		소 계	56,062,226
	기타 회선	SKT	4,722
		KT	121
		LGU+	40
		MVNO	2,694
		소 계	7,577
	합계	SKT	23,169,214
		KT	13,619,810
		LGU+	11,046,644
		MVNO	8,234,135
		합계	56,069,803

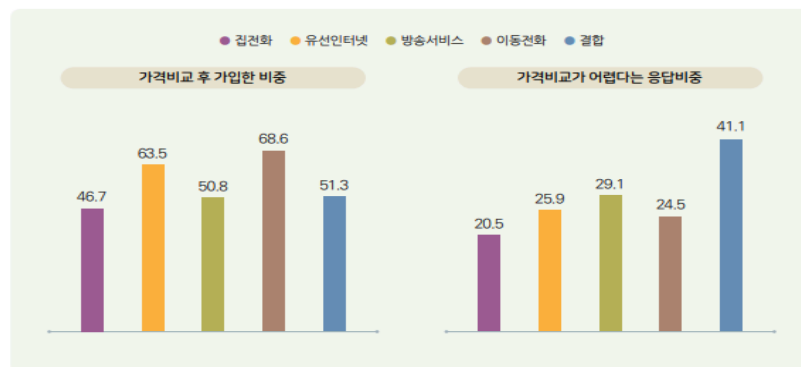
※ '24년 1월부터 휴대폰 회선 중 통신사 내부용 회선, 인바운드 로밍 회선 등을 휴대폰-기타회선으로 재분류

최근 요금이 저렴하고 상품 구조가 단순한 알뜰폰(MVNO)의 인기가 높아지면서 이동통신 3사는 알뜰폰과의 가격 경쟁보다는 차별화된 부가가치를 제공하는 전략을 택하고 있다.

OTT 결합 요금제, 멤버십 혜택, 구독형 서비스 개편 등 다양한 프리미엄 서비스를 통해 충성 고객을 확보하고, 요금 외적인 요소에서 경쟁력을 강화하고자 하는 것이다. LG유플러스 역시 모바일 사업이 전체 매출의 약 40% 이상을 차지하는 핵심 수익원으로, 기존 고객 유지 및 신규 고객 유입을 위한 서비스 다각화에 집중하고 있다.

통신이용자의 요금 이해 관련 설문조사 결과

단위: %



자료: KISDI 통신이용자 설문조사(2021)

하지만 요금제가 점점 복잡해지면서 고객은 적절한 요금제를 찾기 어려워지고, 이는 정보 탐색 피로와 요금제 선택 실패에 대한 불안으로 이어져 만족도 저하 및 이탈로 연결된다.

정보통신정책연구원(KISDI)에 따르면, 통신 요금체계의 복잡성이 증가할 경우 이용자의 합리적인 선택이 저해될 수 있으며, 실제 국내 이용자 대상 설문조사에서도 소비자들이 통신요금의 탐색, 비교, 선택 과정에서 어려움을 겪고 있는 것으로 나타났다.

이러한 상황을 타파하고자 LG유플러스는 Rule 기반의 AI 요금제 추천 챗봇 'ixi 기반 chat agent'를 도입하였다.

## 나. 문제 정의

요금제의 복잡성과 다양성은 고객에게 정보 탐색의 부담과 최적 선택 실패에 대한 불안감을 야기하며, 이는 결국 고객 만족도 저하 및 이탈로 이어질 수 있다. 실제로 고객의 성향, 라이프스타일, 멤버십 정보, 할인 내역 등은 요금제 선택에 큰 영향을 미치지만, 기존 **Rule** 기반 챗봇은 이러한 복합적인 니즈를 반영한 추천을 제공하지 못한다는 한계가 있다.

## 다. 목표

본 프로젝트의 목표는 고객의 개인 통신 성향을 정확히 파악하여 최적의 요금제를 추천함으로써 고객 만족도를 향상시키는 것이다.

### 1. 고객의 통신성향 파악

- 멀티턴 대화를 통해 고객의 통신 성향을 점진적으로 분석하고, 프롬프트 엔지니어링으로 핵심 정보를 실시간 추출하여 고객의 명시적·암묵적 통신 성향을 파악한다.

### 2. 고객 맞춤형 요금제 추천 제공

- **RAG** 기반 **LLM** 챗봇을 이용하여 고객의 통화/데이터 사용 패턴, 라이프스타일, 선호도 등을 바탕으로 최적의 요금제를 추천한다.

### 3. 고객 의사결정 지원 강화

- 요금제 선택 과정에서 발생하는 정보 탐색 부담을 해소하여 고객이 요금제를 쉽게 비교 및 이해할 수 있도록한다.

## 라. 타겟층

### 1. 전화 상담 및 대리점 방문이 어려운 고객

- 24시간 자연어 기반 처리로 대응 가능한 챗봇으로 해결 가능

### 2. 신규 가입을 고려하는 고객

- 타 통신사에서 **LG U+**로 이동을 고려하는 고객

### 3. 현재 요금제에 불만이 있는 기존 고객

- 약정 만료 또는 요금제 사용성 불일치 고객

## 마. 제공가치

### 1. 실시간 개인 맞춤 요금제 추천

- 사용자의 실제 통신 데이터 및 간단한 대화를 기반으로 요금제를 추천하여, 복잡한 요금제 탐색의 어려움을 해소한다.

### 2. 대화 맥락 기반 추천 및 비교 기능

- 고객의 다양한 문의를 이해하고, 유사 요금제에 대한 실시간 비교를 제공한다.

### 3. 상담 비용 절감

- 단순 문의 및 요금제 추천 업무를 자동화하여, 고객 상담센터 인력 운영의 효율성을 제고한다.

### 4. 서비스 이용 데이터 기반 전략 수립

- 사용자의 선택 패턴과 니즈를 분석하여, 향후 요금제 개선 및 마케팅 전략에 반영 가능하다.

## 바. 타당성 분석

### 1. 기술적 타당성

#### 1.1 최신 LLM 활용

- GPT, Claude 3, Gemini 등 최신 LLM을 통해 불완전/추상적 질문 이해 및 복잡한 요금제 조건(데이터, 통화, 할인 등) 종합 분석 후 맞춤형 요금제를 추천할 수 있다.
- T-Mobile, SoftBank 등 글로벌 통신사 상용 챗봇 사례 다수있다.
- Spring Boot의 RestTemplate, FastAPI의 httpx, Lang chain등을 사용해 최신 LLM API을 쉽게 연결하여 사용할 수 있다.

#### 1.2 LangChain 프레임워크 기반 LLM 활용 극대화

- 파이썬 기반 랭체인 프레임워크는 풍부한 레퍼런스가 존재한다.
- System/Role Promoting, Step-Back & CoT와 같은 프롬프트 엔지니어링 기법으로 프롬프트를 구조화하여 작성할 수 있다.

### 1.3 강화된 콘텐츠 안전성 확보

- 금치어 감지 및 **LangChain API** 내 안전 옵션 (**safety\_settings** 등)을 활용할 수 있다.
- 명확한 지침으로 부적절/비윤리적 응답 사전 차단한다.
- 백엔드에서 추가로 필터링하여 안전한 서비스 환경을 구축한다
- **LangChain API** 옵션과 간단한 조건문 또는 정규식 필터로 구현 가능하다.

### 1.4 pg vector를 통해 요금제 검색 최적화

- 단순 키워드 매칭을 넘어선 정교한 의미 기반 검색으로 추천 정확도를 향상이 필요하다. **pgvector** 기반 임베딩과 유사도 검색 기술을 사용하여 이를 실현할 수 있다.

### 1.5 RAG 파이프라인

- LLM의 답변 생성 능력과 **pgvector** 기반 외부 정보 검색을 결합한 **RAG** 구조는 **LangChain**에서 기본적으로 지원된다.
- **FastAPI**로 서버화하여 신뢰도 높은 요금제 정보를 실시간으로 사용자에게 제공할 수 있다.

### 1.6 확장 가능한 이중 서버 백엔드 시스템

#### 1.6.1 Spring Boot 서버(애플리케이션 로직 및 데이터 관리)

- 자동 설정 및 내장 서버 제공과 **IOC/DI** 등 특징을 통해 짧은 프로젝트 기간 동안 대규모 애플리케이션을 유연하고 확장 가능하게 설계할 수 있다.

#### 1.6.2 FastAPI 서버 (LLM 연동 및 AI 처리)

- **Python AI** 생태계 활용(LLM API 호출, **LangChain** 파이프라인, 임베딩, **pg vector**연동)을 **FastAPI**, **LangChain**, **OpenAI/Google AI** 등 **Python** 라이브러리를 사용하여 구현가능하다.

## 2. 경제적 타당성

### 2.1 LLM, 임베딩 사용 비용

- **gpt, gemini, claude** 등의 LLM모델은 API호출형 서비스로 사용량에 따른 종량제 구조를 가진다.

- 질문 당 토큰수를 제한하거나, 프롬프트를 효율적으로 작성하면 비용을 절감한다.
- Claude sonnet 4는 1M tokens당 약 \$6.00, 배치 입력은 \$0.30, 배치 출력은 \$15.00 정도로 책정되어 있어 예산에 충족한다.
- 임베딩 모델 Open AI의 text-embedding-3-small 모델이 1000개 토큰 당 \$0.00002로 다른 모델에 비해 비용이 저렴하여 예산 안에서 사용할 수 있다.

## 2.2 오픈소스 기술 활용으로 개발비 절감

- Spring Boot, LangChain, FastAPI, PostgreSQL, pgvector 등의 기술 스택이 오픈소스로 제공된다.
- 별도 라이선스 비용이 들지 않으며, 개발 및 유지에 필요한 프레임워크/라이브러리 비용이 없다.

## 2.3 클라우드 기반 유연한 인프라 운영 가능

- 예산 5만원의 지원금이 AWS의 EC2 및 RDS와 같은 필요한 클라우드 서비스의 운영에 사용된다. 이를 통해 초기 비용 부담을 줄이고, 개발 및 운영에 집중할 수 있는 기반을 마련할 수 있다.

# 3. 조직성 타당성

## 3.1 데이터 전처리 역량

- 최적 위치 탐색 개발, 코로나19 감염병 확산 모델링 등의 정량적 데이터 처리 및 결과 해석 경험을 보유하고 있다.
- 이러한 역량은 추천 요금제 데이터 전처리 시 생산성 증가를 기대할 수 있다.

## 3.2 프론트엔드 역량

- 팀원들은 Vue.js 및 React 기반 프론트엔드 개발 경험을 보유하고 있다.
- 사용자 친화적인 챗봇 UI 및 관리자 대시보드를 직접 설계하고 구현할 수 있다.

## 3.3 백엔드 역량

- 팀원 모두 Spring Boot, Spring Security, JPA 등 프레임워크 경험을 보유하고 있다.
- CI/CD 자동화 구축 경험을 통해 지속 가능한 서비스 운영 및 유지보수 체계 수립이 가능하다.
- PostgreSQL, pgvector 등 RDB 기반 구조 설계 및 운영에도 익숙하여 데이터 연동에 강점이 있다.

## 3.4 LLM 활용 역량

- GPT, Claude, Gemini 등 최신 LLM 기반 개인화 시스템 구축 경험을 보유하고 있다.

프롬프트 설계, LangChain 사용, 벡터 검색(RAG) 기반 구조 설계 등 실제 사용자 조건을 반영한 LLM 연동 프로젝트 경험이 있어, 요금제 추천 챗봇 개발을 안정적으로 수행 가능하다.

## 사. 해결 방안

### 1. RAG(Retrieval-Augmented Generation) 기반 요금제 추천 챗봇 개발

- RAG 기반 요금제 추천 챗봇 개발을 통해 기존 Rule 기반 챗봇의 한계를 극복하고자 한다.
- RAG 시스템을 활용하여 최신 요금제 정보 반영과 정확한 정보 제공을 보장하며, 고객의 통신 성향, 라이프스타일, 할인 혜택 등의 다차원적 데이터를 종합 분석하여 최적의 요금제를 추천한다.

### 2. LLM 기반 대화형 AI 인터페이스 구현

- 대화형 AI 인터페이스를 활용하여 고객과의 자연스러운 대화를 통해 숨겨진 니즈를 파악하고, 실시간으로 개인 맞춤형 요금제를 제안한다.
- 이를 통해 복잡한 요금제 구조를 고객이 쉽게 이해할 수 있도록 직관적인 비교 분석 기능을 제공한다.

## 아. 기대 효과

### 1. 고객 경험 개선

- Rule 기반 챗봇이 제공하지 못했던 자연스러운 대화와 복합적인 요금제 비교 기능을 통해 실제 상담사와 대화하는 것과 같은 경험을 제공한다.

### 2. 고객 이탈 방지 및 충성 고객 유도

- 요금제 불만족으로 인한 해지를 줄이고, 맞춤 추천을 통해 고객 충성도를 향상시킨다.
- 특히 약정 종료 시점 고객의 이탈을 방지하고, 신규 고객 유입도 기대할 수 있다.

프로젝트 수행 방향

(개발일정 및  
역할분담)

● 일정

	Task	Start Date	Due Date	2025											
				05/29	05/30	06/01	06/02	06/03	06/04	06/05	06/06	06/07	06/08	06/09	06/10
1	프로젝트 기획 및 요구 사항 분석	25/05/29	25/06/03												
2	DB Modeling	25/06/03	25/06/03												
3	시스템 설계 및 초기 개발 환경 구성	25/06/04	25/06/08												
4	AWS 배포 환경 구성	25/06/07	25/06/09												
5	요구사항 기능 구현	25/06/09	25/06/16												
6	요구사항 기능 고도화	25/06/16	25/06/20												
7	통합 테스트 시나리오 작성 및 수행	25/06/18	25/06/20												

	Task	Start Date	Due Date	2025								
				06/11	06/12	06/13	06/14	06/15	06/16	06/17	06/18	06/19
1	프로젝트 기획 및 요구 사항 분석	25/05/29	25/06/03									
2	DB Modeling	25/06/03	25/06/03									
3	시스템 설계 및 초기 개발 환경 구성	25/06/04	25/06/08									
4	AWS 배포 환경 구성	25/06/07	25/06/09									
5	요구사항 기능 구현	25/06/09	25/06/16									
6	요구사항 기능 고도화	25/06/16	25/06/20									
7	통합 테스트 시나리오 작성 및 수행	25/06/18	25/06/20									

05/29 ~ 06/03: 프로젝트 기획 및 요구 사항 분석

06/03 ~ 06/03: DB Modeling

06/04 ~ 06/08: 시스템 설계 및 초기 개발 환경 구성

06/07 ~ 06/09: AWS 배포 환경 구성

06/09 ~ 06/16: 요구사항 기능 구현

06/16 ~ 06/20: 요구사항 기능 고도화

06/18 ~ 06/20: 통합 테스트 시나리오 작성 및 수행

● 역할분담

이름	팀 역할	개발 역할
권영태 (조장)	팀장, WBS 관리	어드민 대시보드 페이지, 챗봇 리뷰 저장, 어드민 요금제 목록 API 리뷰 정보 요약(LLM)
김원석	백엔드 보안 관리자	로그인 페이지, 요금제 페이지,



			로그인 <b>API</b> 및 인증/인가, <b>LLM</b> 멀티턴 엔지니어링
	김희진	서기	챗봇 대화 페이지, 챗봇 메시지 다건 조회 <b>API</b> , <b>LLM</b> 프롬프트 엔지니어링, 금칙어 필터링( <b>LLM</b> )
	박기정	<b>QA</b>	데이터 전처리
	이가인	산출물 관리자	어드민 리뷰 페이지, 어드민 요금제 생성 <b>API</b> , 어드민 요금제 삭제 <b>API</b> , 어드민 요금제 지표 조회 <b>API</b> , 어드민 챗봇 리뷰 조회 <b>API</b> , 데이터 전처리( <b>RAG</b> )
	조윤주	서기	어드민 요금제 관리 페이지, 백엔드 금칙어 필터링, 벡터 임베딩( <b>RAG</b> )
	한현우	스크럼 리더	요금제 상세 페이지, 요금제 목록 <b>API</b> , 요금제 상세 <b>API</b> , 데이터 전처리( <b>RAG</b> )
	황지연	인프라 담당	챗봇 리뷰 페이지, 챗봇 채팅 메시지 저장 <b>API</b> , 데이터 수정 대응( <b>RAG</b> )
프로젝트 수행 도구	1. 아키텍처		



### 3.4 Chart.js

- 비교적 가볍고 간단한 데이터 시각화 라이브러리를 통해 어드민 대시보드 페이지 내 막대그래프를 구현한다.

### 3.5 Tailwind CSS

- 유틸리티 클래스 기반 스타일링을 통해 UI 구성 요소를 구현한다.

## 4. Back end

### 4.1 Java JDK 17

- record, pattern matching으로 멀티턴 대화 모델링을 간결하게 한다. 또한, Spring Framework 6 기반인 Spring Boot 3.x은 최소 Java JDK 17을 요구한다.

### 4.2 Spring Boot 3.3.12

- 높은 안정성, 활발한 지원, 친클라우드성, 외부 라이브러리 호환성을 가진 Spring Boot 3.X 버전을 채택한다. 그 중 3.4.X와 3.5.X도 어느정도 안정화가 되었지만, 비교적 더 안정성을 가진 3.3.12 버전을 채택한다.

### 4.3 Spring Security

- Security Filter Chain을 통한 인증/인가, 권한 처리 쉽게 관리한다.

### 4.4 Spring Data JPA

- ORM을 통한 객체 관계 매핑을 통해 RDB 또는 NoSQL 접근을 편리하게 한다.

### 4.5 Lombok

- 반복적이고 중복적인 코드 작업(Getter, Setter 등)을 줄이고, 편리하게 구현한다.

### 4.6 OpenFegin/QueryDSL

- Java 문법 기반 쿼리 작성을 통해 재사용성 을 높이고, 동적 조건절을 편리하게 구현한다.

## 5. AI Chatbot

### 5.1 LangChain

- RAG기반 챗봇 시스템의 파이프라인을 연결해준다.
- `prompt_template`를 통해 프롬프트 엔지니어링 기법을 편리하게 적용할 수 있다.
- LLM 모델 변경성과 확장성이 뛰어나다.

### 5.2 FastAPI

- RAG 파이프라인은 네트워크 I/O 중심의 직렬 구조이므로, 비동기 처리가 가능한 웹 프레임 워크가 요구된다.
- FastAPI는 Python의 `async/await` 문법을 기반으로 작동하며, ASGI 서버 위에서 실행되어 `non-blocking I/O` 처리가 가능한 환경을 제공한다.

## 6. DataBase

### 6.1 MongoDB

- JSON 형태의 문서 구조로 데이터를 직관적으로 저장할 수 있고, 채팅 및 요금제와 같은 비정형 데이터를 유연하게 처리할 수 있도록 지원한다.

### 6.2 PostgreSQL + pgvector

- 회원 정보, 채팅방 정보, 데이터/통화/문자 사용량, 모바일 기기의 데이터를 저장하기 위한 RDBMS로 PostgreSQL을 사용한다.
- PostgreSQL의 확장 기능인 `pgvector`를 사용함으로써 요금제 정보의 벡터 데이터의 영속성을 보장한다.
- 벡터 인덱싱을 활용하여 고속의 유사도 검색이 가능하다.

### 6.3 Redis

- TTL설정을 두어 민감한 정보를 간편하게 만료시킬 수 있다.

- **In-Memory** 구조로 메모리에서 데이터를 가져오기 때문에 빠르게 **Refresh Token**을 가져올 수 있다.
- **RTR**방식을 사용할 때 유리하다. 토큰을 재발급할 때 **DEL old\_token** 처럼 간편히 토큰을 제거하고 새로운 토큰을 발급할 수 있다.

## 7. Infra

### 7.1 AWS EC2

- 클라우드 컴퓨팅 환경에서 **WAS(Web Application Server)**를 배포하고 운영하여 안정적인 서비스를 제공한다.

### 7.2 AWS RDS(PostgreSQL + pgvector)

- 관계형 데이터베이스를 클라우드에서 관리하고, **pgvector** 확장을 통해 벡터 검색 기능을 지원한다.

### 7.3 Docker

- 각 백엔드 구성 요소(Spring Boot, Redis, MongoDB)를 개별 컨테이너로 분리하여 실행 환경의 일관성과 배포 자동화를 지원한다.

### 7.4 Cloudflare Page

- **Vue.js**로 개발된 프론트엔드 애플리케이션을 정적 사이트로 배포한다. **HTTPS**, 트래픽 관리 기능을 통해 서비스 안정성과 성능을 강화한다.

### 7.5 Github Actions

- 코드 변경 시 자동 빌드 및 테스트 후, **Docker** 컨테이너를 **EC2**에 배포하여 **CI/CD** 파이프라인을 구성한다.

## 8. Communication Tools

### 8.1 Git / Github

- 프로젝트 소스 코드를 버전 별로 관리하고, 협업을 위한 **Branch** 및 **Pull Request** 기반의 코드 리뷰를 수행한다.

### 8.2 Jira

- 프로젝트 일정과 작업 진행 상황을 체계적으로 관리하고, 이슈 트래킹을 통해 업무 분담을 명확히 한다.

	<p><b>8.3 Slack</b></p> <ul style="list-style-type: none"><li>- 팀원 간의 실시간 소통과 <b>Github PR</b> 알림 등 외부 툴 연동을 통한 알림 관리에 활용한다.</li></ul> <p><b>8.4 Notion</b></p> <ul style="list-style-type: none"><li>- 기획안, 회의록, <b>API</b> 명세서, 기술 문서 등 프로젝트 관리 문서를 통합적으로 관리한다.</li></ul> <p><b>8.5 Discord</b></p> <ul style="list-style-type: none"><li>- 비대면 상황에서도 원활한 실시간 회의를 위한 커뮤니케이션 도구로 활용한다.</li></ul>
<p>필수 기능</p> <p>(주요 구현 기능)</p>	<p>1. 회원가입/로그인/로그아웃 기능</p> <p>2. 요금제 추천 챗봇</p> <ul style="list-style-type: none"><li>- <b>LLM</b>을 이용하여 사용자 질문의 도메인 판별</li><li>- 사용자의 실제 통신 데이터 및 대화를 기반으로 맞춤형 요금제 추천</li><li>- 안정적이고 일관적인 어투의 답변</li><li>- 사용자와의 최근 질문과 응답 내역을 프롬프트에 함께 포함하여 자연스러운 멀티턴 대화</li><li>- 질문과 응답에 대한 금칙어(욕설, 해킹) 필터링</li><li>- 사용자 별 챗봇 질의응답 저장</li></ul> <p>3. 챗봇의 상품 추천에 대한 리뷰</p> <ul style="list-style-type: none"><li>- 상품 추천 시 사용자에게 상품 추천에 대한 만족도를 1점 ~ 5점 단위로 선택 및 제출</li><li>- 상품 추천에 대한 최대 300자 범위내의 텍스트 리뷰를 작성 및 제출</li><li>- 금칙어(욕설, 해킹) 필터링</li></ul> <p>4. 관리자의 요금제 관리</p> <ul style="list-style-type: none"><li>- 요금제 상품 등록</li><li>- 요금제 상품 삭제</li><li>- 인기 있는 요금제 상품을 지표(막대그래프)로 조회</li></ul> <p>5. 관리자의 챗봇 상품 추천에 대한 리뷰 조회</p> <ul style="list-style-type: none"><li>- 각 리뷰가 이루어지기까지의 질문-답변 흐름의 요약과, 챗봇이 추천한 상품의 핵심 내용을 함께 제공</li><li>- 리뷰 평점과 텍스트 리뷰는 원문으로 조회</li></ul>

## 1. 임베딩 기술 및 도구

임베딩 기술 및 도구	설명
<b>Embedding</b> * OpenAI text-embedding-3-small model	요금제 정보와 사용자 질의를 각각 벡터 데이터로 변환하기 위해 임베딩 을 사용한다.  이때, 빠른 응답 속도와 낮은 비용 대비 우수한 성능을 제공하는 <b>text-embedding-3-small</b> 모델을 사용한다.
<b>PostgreSQL + pgvector</b>	요금제 데이터의 벡터 데이터 저장 및 검색을 위한 데이터베이스로, 유사도 기반으로 사용자 질문에 가장 유사한 데이터를 효율적으로 검색할 수 있다.
<b>RAG</b>	사용자의 요금제 질의(Query)를 임베딩 한 후, <b>pgvector</b> 를 활용해 유사한 요금제 데이터를 검색하여 자연어 처리 기반 <b>LLM</b> 에게 전달함으로써 신뢰도 있는 응답을 생성할 수 있다.

## 2. 프롬프트 엔지니어링 기법

프롬프트 엔지니어링 기법	설명
<b>Zero-Shot Prompting</b>	사전 예시 없이도 단순 명령을 이해하고 실행할 수 있어, 비회원 사용자 대응이나 빠른 반응이 필요한 상황에 활용한다.
<b>One-Shot / Few-Shot Prompting</b>	1~2개의 예시만으로 말투와 응답 방식을 모방할 수 있어, 자연스러운 챗봇 어투 유도나 말투 제어에 활용한다.
<b>System Prompting</b>	챗봇의 기본 역할과 말투를 지정하여 일관된 챗봇의 말투를 유지하도록 활용한다.

	Contextual Prompting	유저 상황을 반영한 응답을 유도할 수 있어, 로그인한 회원 정보를 활용한 개인화 추천에 활용한다.
	Role Prompting	챗봇에게 역할을 명확히 부여함으로써, 전문적이고 신뢰감 있는 어투를 유지하도록 한다.
	Step-Back Prompting	바로 답변하지 않고 고객의 요청을 먼저 요약하도록 유도함으로써, 의도 파악의 정확도를 높이고 멀티턴 대화 흐름의 최적화에 활용한다.
	CoT(Chain of Thought)	복잡한 질문에 대해 단계별로 사고한 뒤 응답하는 방식으로, 고객 조건 확인부터 요금제 후보 선정, 장단점 비교, 최종 추천까지의 절차를 통해 요금제 비교의 높은 정확도를 위해서 활용한다.