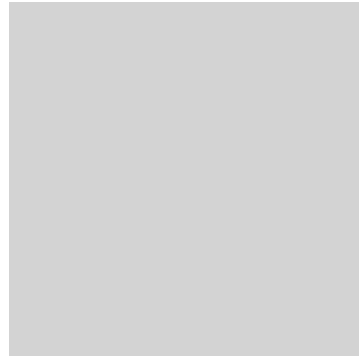
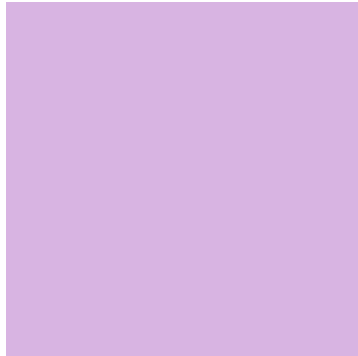


역할	색상 이름	RGB 코드
주색	바이올렛	#6A4C93 → rgb(106, 76, 147)
보조	라벤더	#D8B4E2 → rgb(216, 180, 226)
보조	실버 그레이	#D3D3D3 → rgb(211, 211, 211)
포인트	레드와인	#B00020 → rgb(176, 0, 32)
텍스트	딥 네이비	#2D3142 → rgb(45, 49, 66)



폰트	프리젠테이션 5 Medium
강조 폰트	프리젠테이션 8 ExtraBold
	테스트중

코딩
사각

수정 필요

기업맞춤형 AI-X 융복합 인재 양성 교육
2차 프로젝트

LLM-RAG 기반 Django 자동차보험 추천 플랫폼

AI 기술과 실무 비즈니스 요구사항을 결합한 실용적인 서비스 개발

코딩씩

김진경 / 김민준 / 장선호 / 한창희 / 유성미



Prologue

“ 자동차가 달리는 모든 순간이 아름다운 추억이 되도록
나와 우리가 함께하는 이 길이 안전할 수 있도록
마음과 마음이 이어지는 조용한 약속



자동차 보험”

[어플 이름]

Chapter 01

코딩사

프로젝트 소개

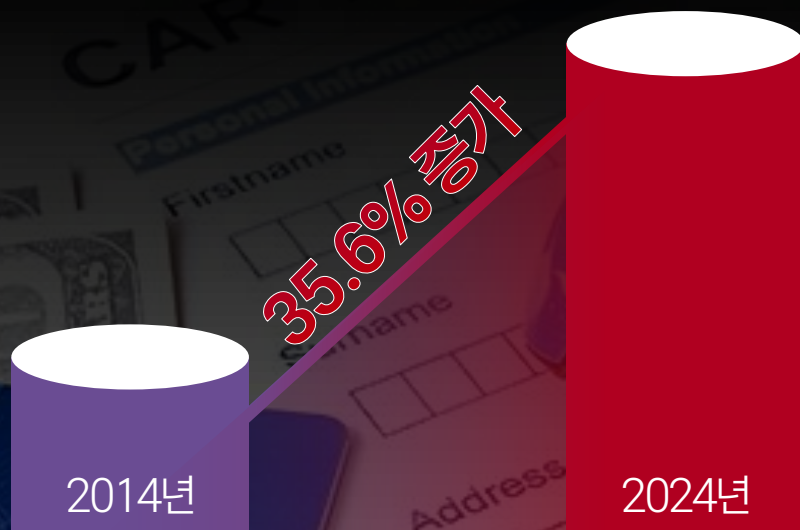
Project Overview

- 01. 주제 선정 및 배경
- 02. 데이터 수집
- 03. 업무분장
- 04. 추진 일정표
- 05. 개발 환경

운전자라면 누구나 겪는 Pain Point

넘쳐나는 보험 상품과 복잡한 약관에 지친 소비자들의 정보 과부하 시대!

자동차 등록 현황



출처 : 국토교통통계누리 - 자동차등록현황보고

자동차 보험 현황

그러나 보험시장은?

- 자동차 보험 시장 규모 -
20조원 규모 (24년 기준)

대규모 어찌고 저찌고

- 보험 수익성 활성화 -
다양한 보험 서비스 제공

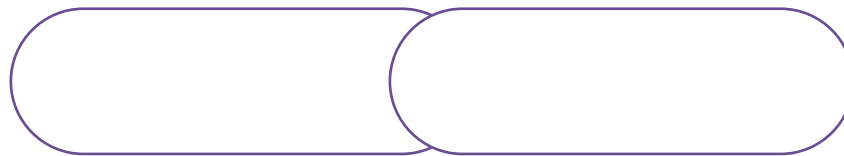
소비자 선택지 다양화

- 가입 방식 변화 -
온라인/비대면 가입 증가

가입 채널의 디지털 전환

운전자라면 누구나 겪는 Pain Point

넘쳐나는 보험 상품과 복잡한 약관에 지친 소비자들의 정보 과부하 시대!



✕ Pain Point

보험사별 정보 파편화 흩어진 약관 정보로 인한 고객 접근성이 낮음

어려운 전문 용어로 작성된 약관, 비전문가들에게 이해도 저하

직접 비교하며 찾아봐야 하는 선택의 다양성으로 피로도 향상

단순한 정보 안내하는 챗봇으로 구체적 답변 부족

✓ Solution

보험사 약관 통합 제공, 챗봇으로 정보 접근성 극대화

약관 내 전문 용어들 상세 해설, 쉽게 이해할 수 있게 명확성 증진

맞춤형 체크리스트와 추천 시스템으로 간편한 상품 선택 지원

LLM 기반 소통형 챗봇 도입, 고객 질문 의도에 맞춰 심층적 상담 제공

운전자라면 누구나 겪는 Pain Point

넘쳐나는 보험 상품과 복잡한 약관에 지친 소비자들의 정보 과부하 시대!

POINT1

차별화 전략 1

LLM 자연어 처리

키워드를 넘어 문맥 이해,
친근한 대화 챗봇

RAG 기반 정확성

LLM 한계 극복,
신뢰 높은 정보 제공

POINT2

차별화 전략 2

개인맞춤형

정형화 NO,
개인별 최적 솔루션

상시운영

시공간 제약 없이
24시간 상시 접근

운전자라면 누구나 겪는 Pain Point

넘쳐나는 보험 상품과 복잡한 약관에 지친 소비자들의 정보 과부하 시대!

왼쪽 주요 문장
다시 정리 필요

1



사회초년생

김초보

20대 중반 / 여성 / 신규 가입

난생 처음 자차 보험 가입,
약관이 너무 복잡하고 어려워요.

복잡한 용어 이해 부족

개인맞춤형 보장 모름

2



베테랑 운전자

박중남

40대 초반 / 남성 / 보험 갱신

보험 갱신 시점, 여러회사 비교중
보험 비교 자료 정리 시간부족

상담원 연결의 비효율성

정확한 비교 정보 부족

3



정보취약계층

어르신

60대 중반 / 여성 / ○○○○

나이가 들어 글씨도 안 보이고,
어려운 단어는 이해가 안됨

디지털 소외 정보 불균형

직접 상담이 어려움

다시 정리 필요

데이터 종류

다시 정리 필요

업무분장

다시 정리 필요

추진일정

다시 정리 필요

개발환경

[어플 이름]

Chapter 01

코딩사

프로젝트 세부 운영

Project -----

01.00000
02.00000
03.00000
04.00000
05.00000



I 서론

1. 주제 선정 및 배경

프로젝트 주제 선정 배경

기술적 도전

LLM + RAG + ML 기술 융합 시스템 구현

고객 Pain Point

복잡한 자동차보험 상품 비교의 어려움

시장 필요성

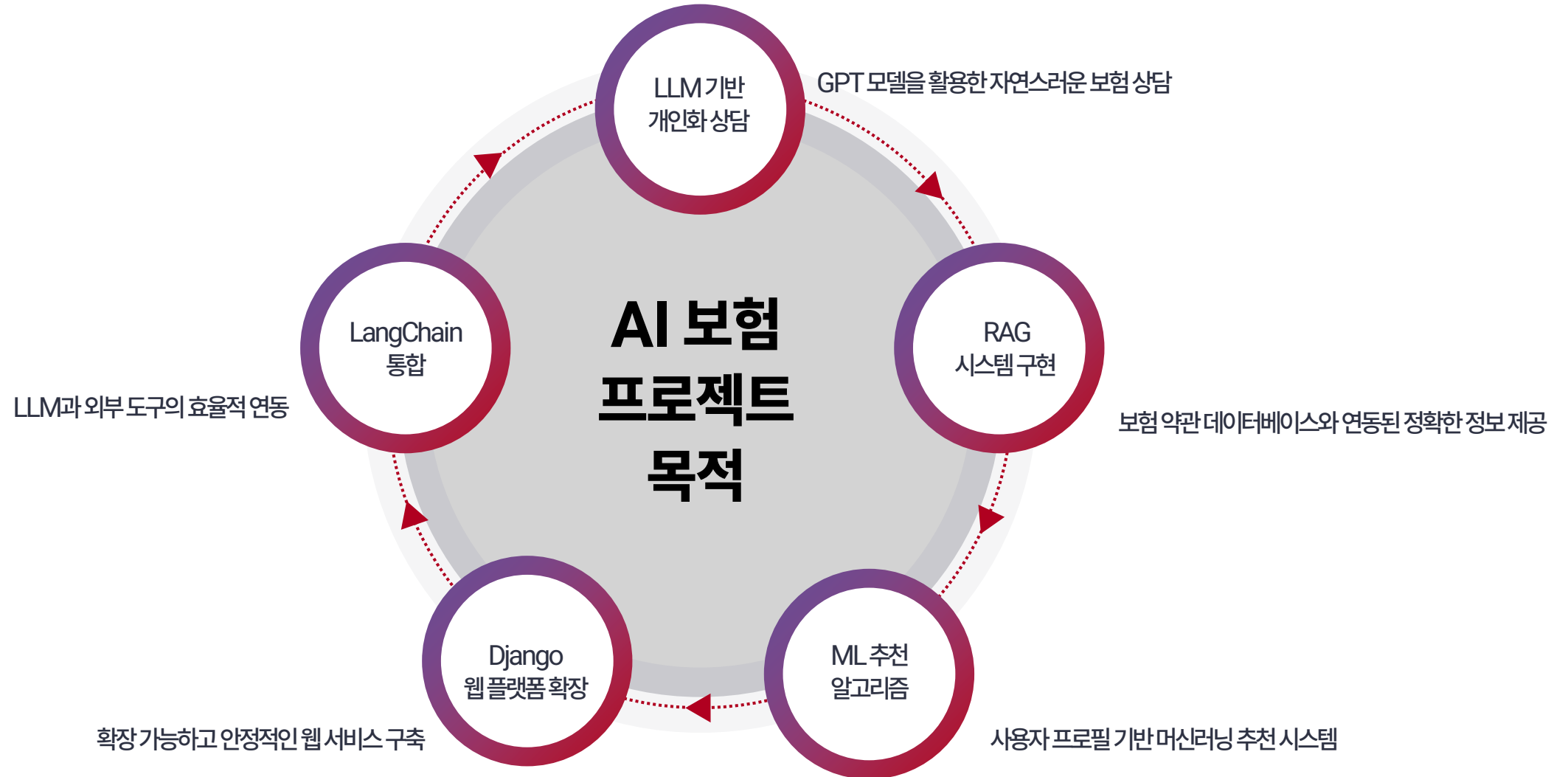
기반 확장 가능한 개인 맞춤형 보험 추천 서비스

AI 기술 활용

LangChain을 통한 실용적 LLM 애플리케이션 개발

I 서론

1. 주제 선정 및 배경



I 서론

2. 데이터 수집

손해보험협회 보험사별 약관 데이터 수집







자동차보험 종합포털
손해보험협회

자동차보험 종합포털이란?

정보포털서비스 약관 상품설명서 정보검색서비스 상담창구 공지사항

HOME > 약관 상품설명서 > 보험사별 약관

보험사별 약관

 메리츠화재 메리츠화재보험주식회사 개인용 약관 →	 한화손해보험 한화손해보험주식회사 개인용 약관 →	 롯데손해보험 롯데손해보험주식회사 개인용 약관 →
 MG손해보험 MG손해보험주식회사 개인용 약관 →	 Heungkuk Fire & Marine Insurance 흥국화재해상보험주식회사 개인용 약관 →	 삼성화재 SAMSUNG 삼성화재해상보험주식회사 개인용 약관 →
 현대해상 현대해상화재보험주식회사 개인용 약관 →	 KB손해보험 KB손해보험주식회사 개인용 약관 →	 DB손해보험 DB손해보험주식회사 개인용 약관 →
 AXA AXA손해보험주식회사 개인용 약관 →	 하나손해보험 하나손해보험주식회사 개인용 약관 →	 Carrot 캐롯손해보험주식회사 개인용 약관 →

DB손해보험

장제약속

프로미카 다이렉트(플랫폼)
개인용자동차보험
약관
2025년 7월 1일

promy car | promy life

본 약관은 「자동차보험법」 제41조 제1항에 따라 DB손해보험(이하 "보험회사"라 함)이 자동차보험의 보험계약의 체결, 보험금 지급에 관한 사항을 규정하고 있습니다. 본 약관은 DB손해보험(이하 "보험회사"라 함)의 자동차보험의 보험계약의 체결, 보험금 지급에 관한 사항을 규정하고 있습니다.

개인용 자동차보험(공동)

● 보통약관
● 특별약관

KB손해보험

KB
개인용 자동차보험
고급형
KB손해보험

자동차보험법 제41조 제1항에 따라 KB손해보험(이하 "보험회사"라 함)이 자동차보험의 보험계약의 체결, 보험금 지급에 관한 사항을 규정하고 있습니다. 본 약관은 KB손해보험(이하 "보험회사"라 함)의 자동차보험의 보험계약의 체결, 보험금 지급에 관한 사항을 규정하고 있습니다.

KB손해보험

캐롯손해보험

캐롯 자동차보험
개인용 약관

자동차보험법 제41조 제1항에 따라 캐롯손해보험(이하 "보험회사"라 함)이 자동차보험의 보험계약의 체결, 보험금 지급에 관한 사항을 규정하고 있습니다. 본 약관은 캐롯손해보험(이하 "보험회사"라 함)의 자동차보험의 보험계약의 체결, 보험금 지급에 관한 사항을 규정하고 있습니다.

캐롯손해보험

hicar

2025. 07. 01 개정

개인용자동차보험 약관
Hyundai Insurance

공공하고 안전한 도로 위해 해나쳐
하이카자동차보험

현대해상화재보험

개인용
자동차보험 약관

2025. 07. 06 개정

I 서론

2. 데이터 수집

데이터 수집 방안

[웹 크롤링]

각 보험사 공식 홈페이지
약관 데이터 수집



[구조화 작업]

PDF / HTML 문서를
구조화된 데이터 변환



[데이터 검증]

수집된 약관의
정확성 및 최신성 검증

데이터 전처리

[텍스트 정제]

불필요한 서식 제거 및 표준화



[백터화]

약관 텍스트의 임베딩 생성



[인덱싱]

Pinecone을 활용한
백터 데이터베이스 구축

I 서론

3. 프로젝트 업무분장

				
김 진 경 프로젝트 매니저 / 백엔드 개발	김 민 준 프론트엔드 개발 / UI/UX	장 선 호 데이터 분석 / AI 모델링	한 창 희 데이터 수집 / 크롤링	유 성 미 데이터베이스 설계 / API 개발
[프로젝트] : 프로젝트 총괄 관리	[프론트엔드 개발] : 백엔드 API 연동 및 상태 관리 : 사용자 화면 구현	[데이터 분석] : 서비스/사용자 로그데이터 분석 및 인사이트 도출	[데이터 수집] : 공공 데이터, 오픈 API를 통한 데이터 수집 및 정제	[데이터베이스 설계] : DB 스키마 설계 및 관계 설정 : 성능 최적화
[백엔드 개발] : 서버 로직 구현(회원가입/로그인)	[UI/UX] : 와이어프레임 및 프로토타입 설계	[모델링] : 모델 설계, 학습, 평가 : 모델 서빙 및 서비스 연동	[크롤링] : 웹 크롤러 개발 : 자동화 스크립트를 통한 데이터 수집 및 학습 스케줄링	[API 개발] : 인증/권한 관리와 API 문서화
[웹페이지] : DB 관리	[웹페이지] : 메인 페이지 제작 및 전체 검수 : 웹 페이지 디자인 제작 관리 총괄	[웹페이지] : 회원가입 페이지 제작	[웹페이지] : 상담페이지 제작	[웹페이지] : 비교 페이지 제작

I 서론

4. 프로젝트 추진 일정표

구분		7/24	25	28	29	30	31	8/1	4	5	6	7	8	11	12	13	14	18	19	20	21	22	25	26	27	28
기획 및 설계	주제 선정	← 아이디어 공모 →			최종 아이디어 확정																					
	세부 기능 정의									← 기능 구상 →		시스템/ DB 설계														
	업무, 일정 관리					← 1차 일정 조율 →				최종 일정 확정																
	시스템 설계도									← 1차 설계 구상 →		최종 설계 확정														
연구 모형 및 조사 설계	연구 가설 설정					← 연구 가설 설정 →		최종 가설 선정																		
	측정 항목					← 항목 설정 →		측정 항목 확정																		
데이터	데이터 수집& 전처리					← 데이터 분석 →		데이터 확보 및 가공		← 데이터 활용 방안 탐구 →																
	기술									← 코딩 →		RAG 시스템 구축		← ML 기획 →		LLM, LangChin 백엔드 개발			← 웹 서비스 기획 →		Django 프론트엔드개발					
연구/검증	기술 통계 분석																									
	기술 검증														← 웹 서비스 기획 →		통합 테스트 및 버그 수정 / 성능 최적화 / PythonAnywhere 배포									
디자인	전체 디자인									← 기능 구상 →		UI,UX 프로토타입 제작														
문서	프로젝트 문서																									

I 서론

5. 프로젝트 개발 환경

개발 환경

OS	Window 10 pro
Language	Python 3.10
IDE	Anaconda Jupyter NoteBook, Vscode, Cursor
Backend & AI Core Stack	Mock Server, LLM, LangChain, RAG, Django,ML
Database & Vector Store	MySQL, PythonAnywhere
Data Processing & ML	easycodfpy,Pandas, Numpy, Scikit-learn, NLTK/spaCy
Frontend	Django Templates, Bootstrap, JavaScript,Chart.js

II 프로젝트 세부 내용

1. 외부 서비스 연동

Mock 서버 기반 보험료 계산 시스템

Mock 서버 선택 배경

- ▶ CODEF API 정식 서비스 신청 시 사업자등록증 필요
- ▶ 개발 초기 단계에서 외부 API 의존성 제거
- ▶ 팀원 모두가 동일한 환경에서 개발 가능
- ▶ API 호출 비용 및 제한 없는 무제한 테스트

Mock 데이터 설계
실제 보험료
계산 로직과 유사한
알고리즘 구현

환경별 전환 지원
개발용 Mock →
운영용 실제 API
쉬운 전환

완전한 기능 구현
Mock 환경에서도
모든 시스템 기능
완벽 구현

```
# 완벽한 자동차보험 Mock 서버 구현

import random
import math
from datetime import datetime, timedelta
from django.conf import settings
from typing import Dict, List, Any

class InsuranceMockServer:
    """
    실제 자동차보험 계산 로직을 시뮬레이션하는 Mock 서버
    실제 보험사들의 요율 계산 방식을 참고하여 구현
    """

    def __init__(self):
        # 11개 보험사 기본 정보
        self.insurance_companies = {
            '삼성화재': {
                'base_rate': 850000,
                'age_multiplier': {'young': 1.3, 'middle': 1.0, 'senior': 0.85},
                'gender_multiplier': {'M': 1.0, 'F': 0.92},
                'region_multiplier': {'서울': 1.1, '부산': 0.95, '대구': 0.9, '기타': 0.88},
                'experience_bonus': 0.85, # 1년당 5% 할인
                'accident_penalty': 0.25, # 사고당 25% 할인
                'car_type_multiplier': {'경차': 0.8, '소형': 0.9, '중형': 1.0, '대형': 1.15, 'SUV': 1.2},
                'coverage_options': {'기본': '표준', '고급': '프리미엄'},
                'special_discount': '무사고 할인 최대 30%'
            },
            '현대해상': {
                'base_rate': 820000,
                'age_multiplier': {'young': 1.25, 'middle': 1.0, 'senior': 0.88},
                'gender_multiplier': {'M': 1.0, 'F': 0.93},
            },
        }
```

```
# 사용 예시
if __name__ == "__main__":
    # Mock 서버 테스트
    mock_server = InsuranceMockServer()

    # 테스트 사용자 프로필
    test_profile = {
        'birth_date': '1990-05-15',
        'gender': 'M',
        'residence_area': '서울',
        'driving_experience': 8,
        'accident_history': 1,
        'annual_mileage': 15000,
        'car_info': {'type': '중형'},
        'coverage_level': '표준'
    }

    # 보험료 계산 테스트
    result = mock_server.calculate_premium(test_profile)

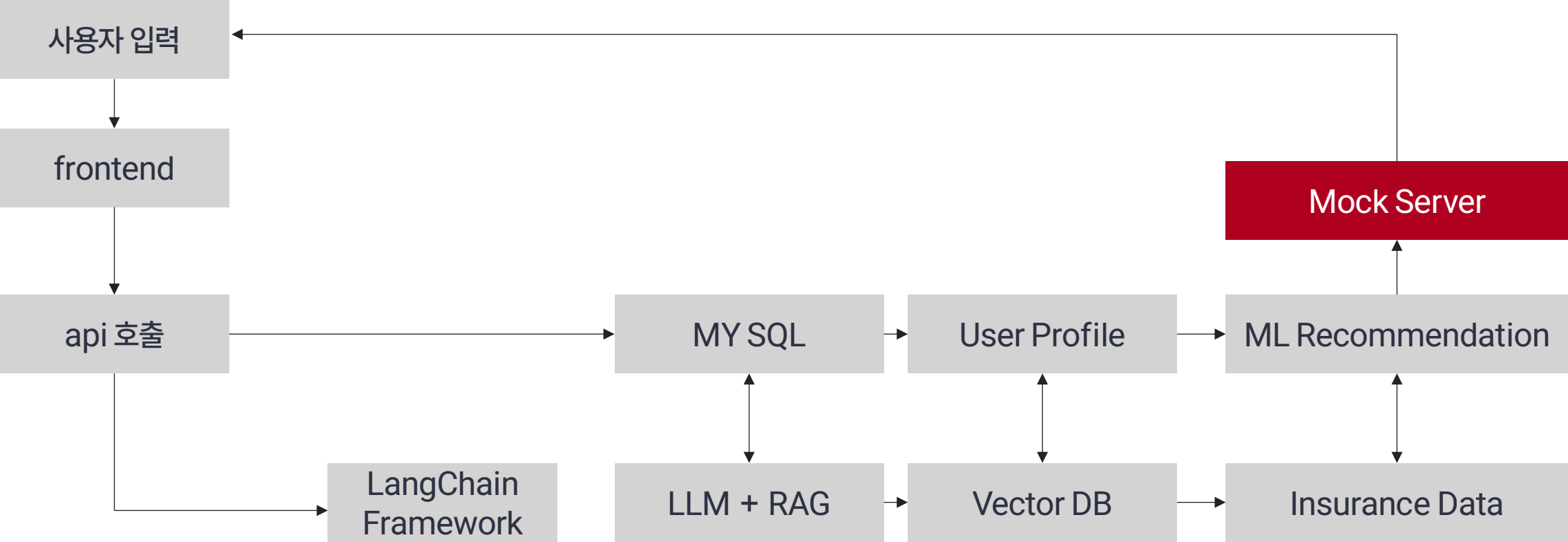
    print("=== 보험료 계산 결과 ===")
    print(f"계산 ID: {result['result']['calculation_id']}")
    print(f"사용자 위험도: {result['user_info']['risk_level']}")
    print(f"추천 보장: {result['user_info']['recommended_coverage']}")
    print(f"평균 보험료: {result['market_analysis']['average_premium']:,}원")
    print(f"가성비 최고: {result['market_analysis']['best_value']}")

    print("\n=== 상위 3개 보험사 견적 ===")
    for i, quote in enumerate(result['quotes'][:3], 1):
        print(f"{i}. {quote['company']}")
        print(f"연간 보험료: {quote['annual_premium']:,}원")
        print(f"월 납입액: {quote['monthly_premium']:,}원")
        print(f"고객만족도: {quote['customer_satisfaction']/5.0}")
```

II 프로젝트 세부 내용

2. 시스템 아키텍처

전체 시스템 구조



II 프로젝트 세부 내용

2. 시스템 아키텍처

회원 관리 시스템

사용자 입력 필수 정보

- ▶ **아이디**: 로그인 식별자
- ▶ **비밀번호**: 암호화 저장(Django 기본 해시)
- ▶ **생년월일**: 보험료 계산 및 위험도 평가

사용자 입력 선택 정보

- ▶ **자동차 번호**: 차량 정보 자동 조회
- ▶ **운전 경력**: 면허 취득일 또는 운전 연수
- ▶ **성별**: 보험료 산정 요소
- ▶ **직업**: 위험도 평가 요소
- ▶ **거주지역**: 지역별 보험료 차등
- ▶ **연간 주행거리**: 사용자 패턴 분석
- ▶ **사고 이력**: 과거 보험 사고 여부
- ▶ **차량 정보**: 연식, 배기량, 차종 등
- ▶ **보험가입 이력**: 현재 가입 보험 및 만료일

II 프로젝트 세부 내용

3. 예상 결과물

웹 애플리 케이선

웹용 기능 목표 페이지

- ▶ **회원가입/로그인**: 사용자 인증 및 프로필 관리
- ▶ **메인 페이지**: 서비스 소개 및 간단 견적
- ▶ **상담 페이지**: AI 챗봇 상담 인터페이스
- ▶ **비교 페이지**: 보험 상품 상세 비교
- ▶ **마이페이지**: 개인정보 및 보험 이력 관리

성과지표

목표 성과 지표

- ▶ **ML 추천 정확도**: 유사 사용자 기반 추천의 정확성
(목표: 85% 이상)
- ▶ **사용자 만족도**: 추천 보험상품에 대한 만족도
(목표: 4.2/5.0 이상)
- ▶ **추천 다양성**: 추천 상품의 다양성 지수 측정
- ▶ **응답 속도**: LLM+RAG+ML 통합 응답 시간
(목표: 5초 이내)
- ▶ **Cold Start 해결률**: 신규 사용자 대상 추천 성공률
(목표: 75% 이상)