

당신의 건강을 이해하는 챗봇

Azure 기반 개인 맞춤형 영 양제 추천 챗봇 개발



INDEX

목차

프로젝트 소개

영양제 추천 챗봇 시스템 개발 목적 소개

관련 기술 및 이론

사용된 기술과 기반 이론 설명

시스템 설계

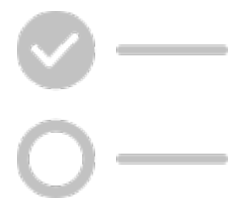
시스템 구성 요소 및 데이터 흐름 구조

구현 내용

백엔드, 프론트엔드, DB 등 주요 구현 요약

테스트 및 결과 분석

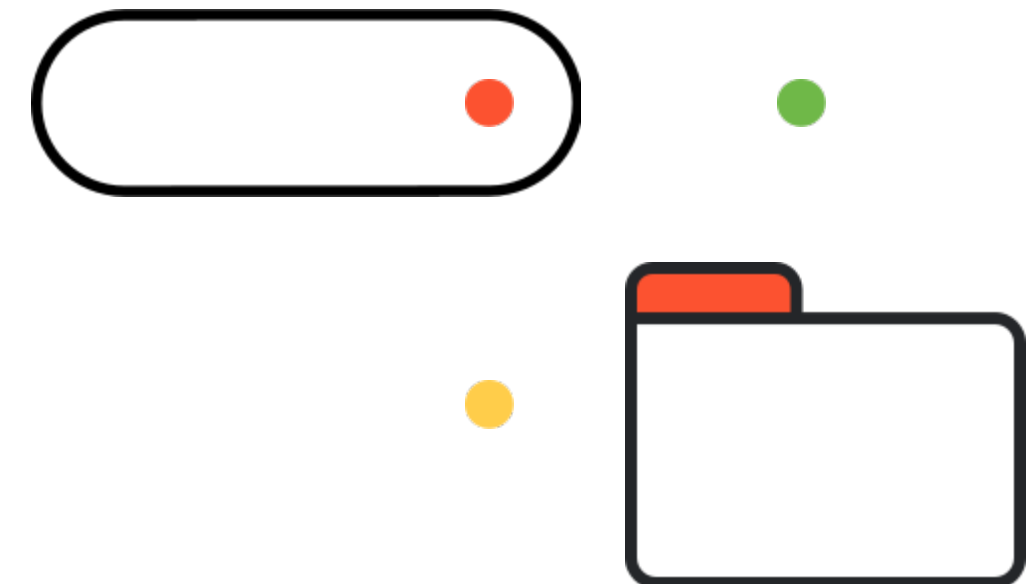
시연 기반 응답 결과 및 분석



01

프로젝트 소개

본 프로젝트는 개인 생체 정보를 기반으로
영양제를 추천하고 복용 방법까지 안내하는
AI 챗봇을 개발하는 것을 목표로 합니다.





팀원 소개

팀장 이정현 Azure 챗봇 엔진 담당

ICT융합학부 미디어테크놀로지 전공

팀원

김상봉 / 인공지능학과 프론트엔드

박성철 / 수리데이터사이언스학과 백엔드

김은영 / 경제학부 Azure 챗봇 엔진 담당

정진욱 / 인공지능학과 프론트엔드

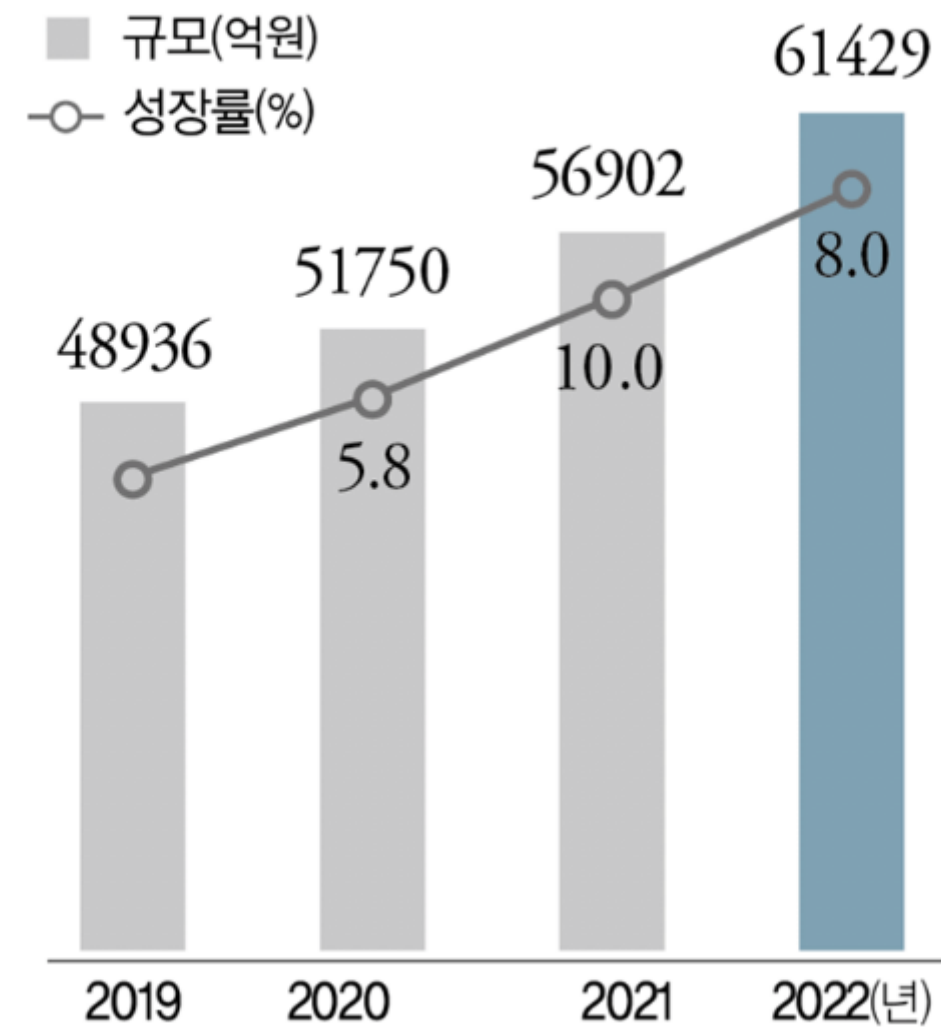
정희찬 / 수리데이터사이언스학과 디자인담당

김동욱/ 컴퓨터학부 ppt 담당

채희준 / 로봇공학과 백엔드

연구 배경

국내 건강기능식품 시장 규모



출처 주간조선

웰빙 관심 증가, 영양제 시장 성장

소비자들은 제품 선택에 어려움을 겪음

영양제 정보의 비대칭성 → 중복 복용, 비용 낭비

건강에 대한 관심은 높아졌지만, 수많은 영양제 중 내게 맞는 것을
고르는 건 여전히 어렵습니다.

연구의 필요성

- 맞춤형 추천 시스템의 필요성

- 모든 사람에게 동일한 영양제를 권장하는 것은 비효율적이며, 개인의 생체 정보에 기반한 맞춤형 추천이 필요합니다.

- 생체 정보 기반 상담의 장점

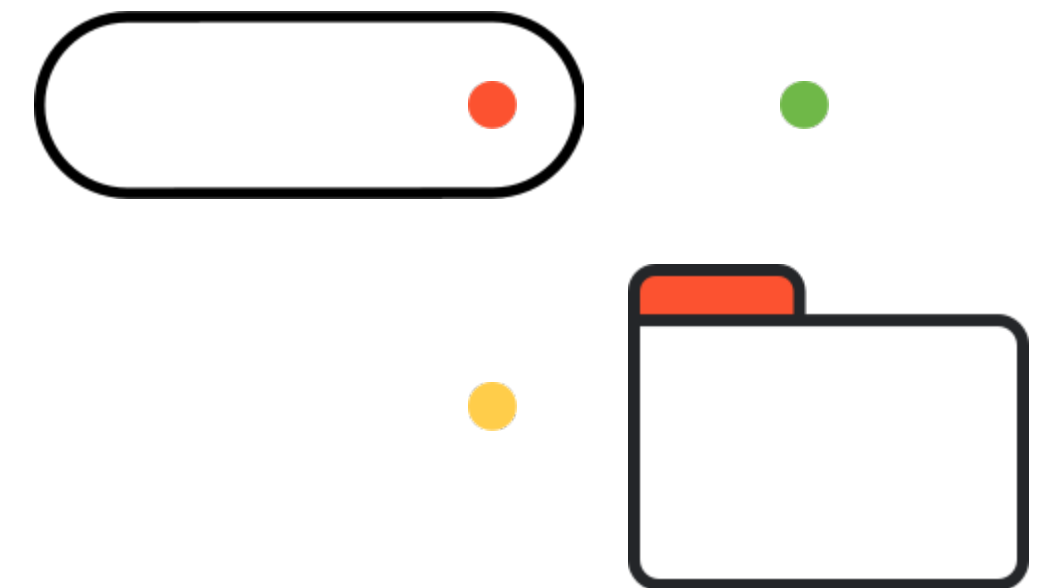
- 키, 몸무게, 나이, 성별 등은 복용 대상에 큰 영향을 미치며, 챗봇은 이를 반영해 정확한 제품과 복용 방법을 제안할 수 있습니다.

- 기존 검색 대비 AI 챗봇의 이점

- 포털 검색은 방대한 정보 속에서 직접 선택해야 하지만, 챗봇은 요약된 맞춤 답변을 바로 제공하여 시간과 효율을 모두 향상시킵니다.

02

관련 기술 및 이론



챗봇 시스템 구현을 위한 핵심 기술과 구성 요소 소개

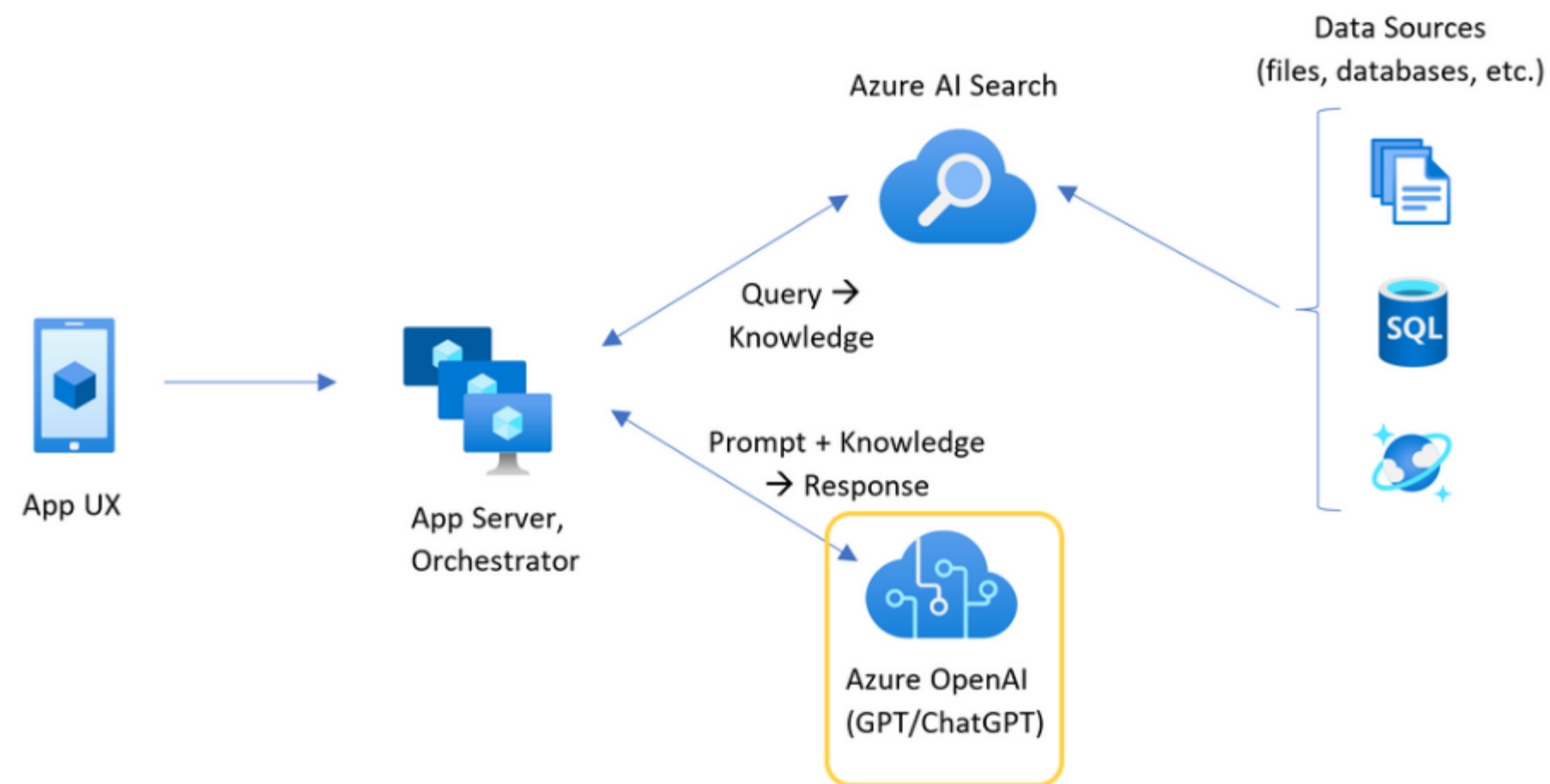
인공지능 챗봇 개요

GPT(Generative Pre-trained Transformer)는 방대한 텍스트 데이터를 기반으로 학습된 고성능 언어 모델로, 사용자 입력의 문맥을 이해하고 자연스러운 대화를 생성합니다.

자연어 처리(NLP) 기술을 활용해 사용자의 질문을 분석하고 적절한 답변 생성

GPT-4 등 최신 모델은 단순 질의응답을 넘어 추론, 요약, 창작 등 다양한 언어 작업 가능

Azure OpenAI API 개요



출처 Learn Microsoft

Microsoft의 Azure 플랫폼을 통해 GPT 모델에 안전하고 쉽게 접근할 수 있는 클라우드 기반 AI 서비스입니다.

별도 서버 없이 GPT API를 호출해 응답 생성 가능

보안성, 확장성, 안정성 확보 (Azure 인프라 기반)

웹 애플리케이션 구조

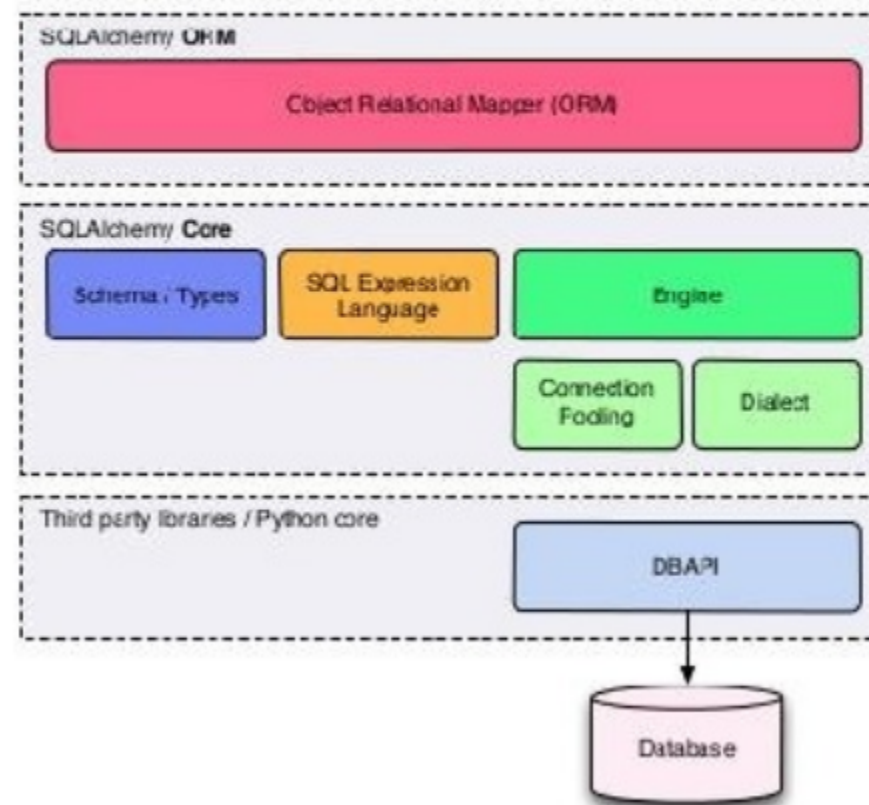
GPT 챗봇은 전통적인 클라이언트-서버 구조를 따릅니다.

- lask (백엔드): 사용자 입력 처리, GPT API 호출
- HTML/JavaScript (프론트엔드): UI 구성, 사용자 입력 전송
- RESTful API로 프론트와 백엔드 연결
- 유연하고 가벼운 구조로 빠른 배포 및 유지보수 가능

데이터베이스 및 ORM

SQLAlchemy Overview

SQLAlchemy consists of the **Core** and the **ORM**



출처 Tutorialspoint

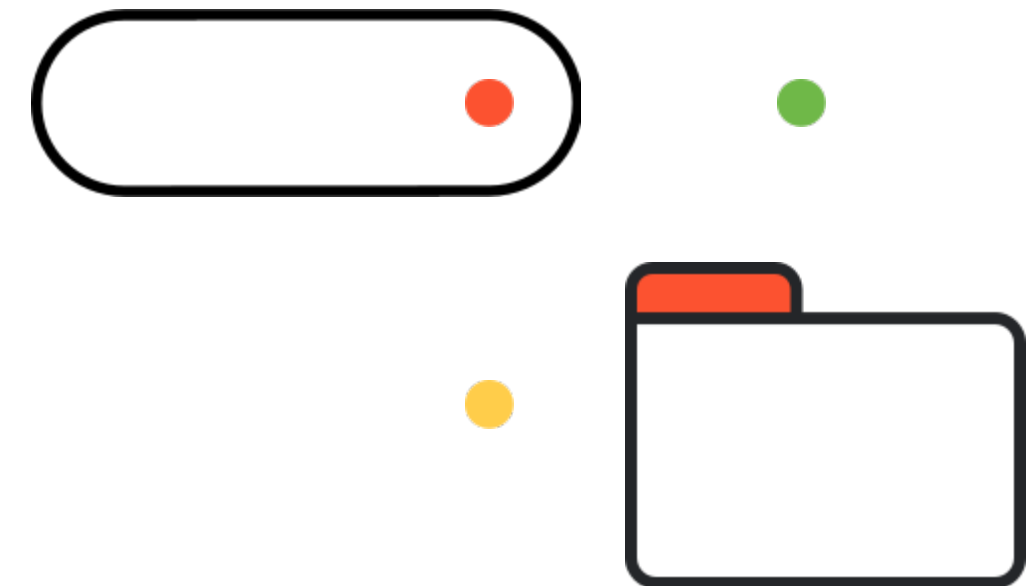
SQLAlchemy를 사용해 데이터베이스를 객체 지향적으로 다룸

- SQL 없이도 데이터 추가/조회/수정이 간편
- 데이터 구조와 로직이 분리되어 유지보수가 쉬움
- 영양제 정보 저장 및 추천 응답 생성 시 활용

03

시스템 설계

사용자 입력부터 맞춤형 응답까지, 챗봇 시스템의 작동 원리



시스템 구성 개요

본 시스템은 총 4개의 구성 요소로 이루어집니다:

프론트엔드 (HTML/JavaScript)

→ 사용자 입력을 받아 서버로 전달

백엔드 (Flask)

→ 사용자 정보를 처리하고 GPT API에 요청

GPT API (Azure OpenAI)

→ 입력 정보를 바탕으로 챗봇 응답 생성

내부 데이터베이스 (SQLAlchemy)

→ 영양제 정보를 저장하고 추천에 활용

사용자 입력 및 응답 처리 흐름

핵심 흐름 설명

사용자 입력: 키, 몸무게, 나이, 성별, 질문

전송 방식: /chatbot 경로로 POST 요청

Flask 서버:

- JSON 데이터 파싱
- 시스템 프롬프트와 함께 GPT에 전달

GPT 응답:

- 사용자 정보 기반 상담 결과 생성
- 응답을 클라이언트로 전달하여 출력



시스템 프롬프트 설계

프롬프트 핵심 역할

- GPT에게 "영양제 박사" 역할을 부여해, 사용자에게 맞춤형 영양제 및 복용 시간/타이밍까지 안내하도록 유도

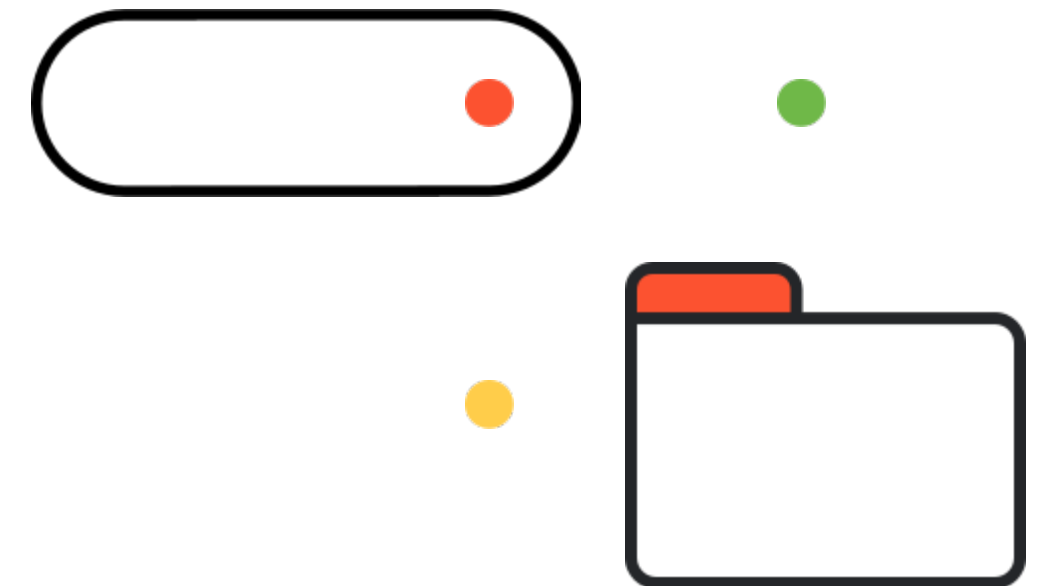
실제 프롬프트 예시

- "너는 지금부터 영양제 박사야. 고객이 영양제 추천이나 상담을 요청하면, "
- "다음 정보(키, 몸무게, 나이, 성별)를 바탕으로 답변해. "
- "또한 복용하면 좋은 시간(아침, 점심, 저녁, 취침 전)과 "
- "공복 또는 식후 중 어떤 타이밍이 적절한지도 설명해줘."

04

구현 내용

프론트엔드부터 백엔드, 데이터베이스까지 구현 과정을
단계별로 살펴보다



시스템 개요 및 구성

- 백엔드: Flask + Azure OpenAI, /chatbot POST API
- 프론트엔드: HTML + JS 폼, 사용자 입력 후 GPT 응답 출력
- 전체 흐름: 사용자 입력 → Flask → GPT → 응답 출력

백엔드 구현 요약

```
# Flask 앱 설정
app = Flask(__name__, static_folder=os.path.join(os.path.dirname(__file__), 'static'))
app.config['SECRET_KEY'] = 'asdf#FGSgvasgf$5$WGT'
app.config['SQLALCHEMY_DATABASE_URI'] = f"sqlite:/// {os.path.join(os.path.dirname(__file__), 'database', 'app.db')}"
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False

# DB 초기화
db.init_app(app)
with app.app_context():
    db.create_all()

# 블루프린트 등록 (import) user_bp: Any
app.register_blueprint(user_bp, url_prefix='/api')
app.register_blueprint(supplement_bp, url_prefix='/api/supplements')

# Azure OpenAI 설정
client = AzureOpenAI(
    api_key="DACG8zSfz6wi4BtnHPzaYYNkScF3jh1b5ZEoJ7MY0WQ0ZPeUIFHHJQQJ99BEACNns7RXJ3w3AAABACOGxCyB",
    api_version="2023-07-01-preview",
    azure_endpoint="https://leejunghyun.openai.azure.com/"
)
```

사용자 정보(JSON) 수신 후 GPT에 전달

Azure OpenAI chat.completions.create() 사용

UTF-8 JSON 형태로 결과 반환

샘플 삽입 스크립트: add_sample_supplements.py 실행 시
DB 초기화 및 데이터 재삽입

프론트엔드 구현 요약

```
<html lang="ko">
<body>
  <main class="main">
    <section id="home" class="section active">
      <div class="container">
        <div class="hero">
          <h2>개인 맞춤형 영양제 추천 서비스</h2>
          <p>당신의 건강한 삶을 위한 최적의 영양제를 찾아보세요</p>
          <button class="btn btn-primary btn-large" onclick="showSection('trending')">시작하기</button>
        </div>
        <div class="features">
          <div class="feature-card">
            <h3>🔥 트렌딩 영양제</h3>
            <p>요즘 가장 인기 있는 영양제들을 확인해보세요</p>
          </div>
          <div class="feature-card">
            <h3>💡 맞춤 추천</h3>
            <p>성별과 나이에 맞는 영양제를 추천받으세요</p>
          </div>
          <div class="feature-card">
            <h3>🤖 AI 상담</h3>
            <p>AI 챗봇과 상담하여 궁금한 점을 해결하세요</p>
          </div>
        </div>
      </div>
    </section>
  </main>
</body>
</html>
```

HTML + JS로 입력 폼 제공

JSON 형태로 Flask 서버에 전송

응답 결과를 브라우저에 출력

데이터 모델 및 활용

```
@supplement_bp.route('/trending', methods=['GET'])
def get_trending_supplements():
    """요즘 떠오르는 영양제 추천"""
    try:
        supplements = Supplement.query.filter_by(is_trending=True).all()
        return jsonify([supplement.to_dict() for supplement in supplements]), 200
    except Exception as e:
        return jsonify({'error': str(e)}), 500

@supplement_bp.route('/by-demographics', methods=['GET'])
def get_supplements_by_demographics():
    """성별/나이대별 영양제 추천"""
    try:
        gender = request.args.get('gender') # 'male' or 'female'
        age_group = request.args.get('age_group') # '20s', '30s', '40s', '50s+'

        query = Supplement.query

        if gender:
            query = query.filter_by(category=gender)

        if age_group:
            query = query.filter_by(age_group=age_group)

        supplements = query.all()
        return jsonify([supplement.to_dict() for supplement in supplements]), 200
    except Exception as e:
        return jsonify({'error': str(e)}), 500
```

SQLAlchemy 기반 Supplement 모델

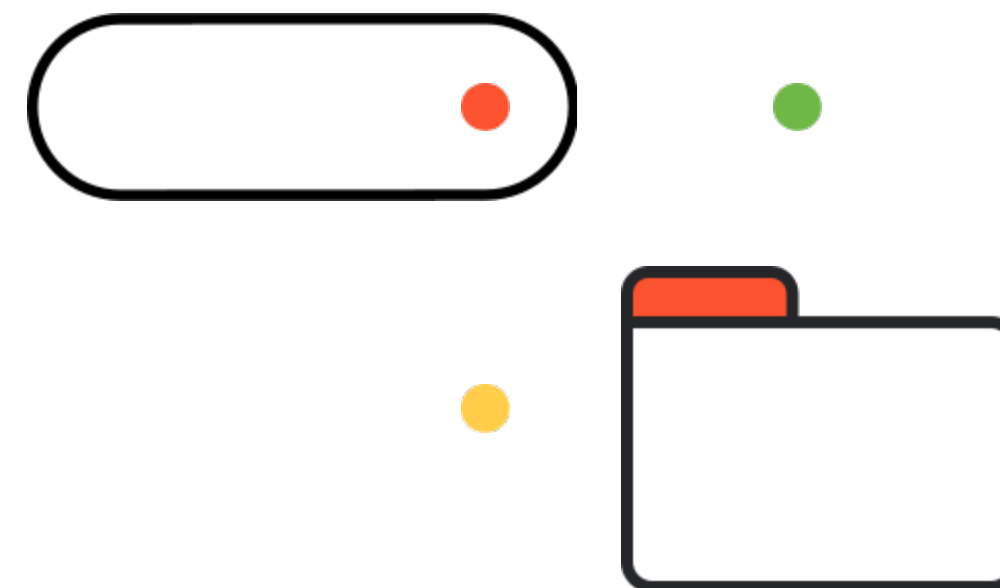
주요 필드: 이름, 설명, 카테고리, 연령대, 가격 등

모델 정보는 GPT가 보다 정확한 상담 응답 생성에 사용됨



05

테스트 시나리오 및 프로젝트 요약



테스트 시나리오 및 시스템 요약

테스트 시나리오 및 시스템 요약

AI 영양제 상담

안녕하세요! 영양제 관련 궁금한 점이 있으시면 언제든지 물어보세요.

메시지를 입력하세요...

전송

테스트 시나리오 개요

사용자: 키, 몸무게, 나이, 성별, 질문 입력

백엔드: Flask 서버가 JSON 데이터 수신 → GPT에 전달

응답: 영양제 추천 + 복용 시간 + 섭취 방식 안내 포함

실행 시연 안내

테스트는 발표자가 직접 코드 실행 시연 예정

실제 응답을 통해 챗봇의 작동 흐름과 결과 확인

예상 응답 예시 (요약 문장으로)

“비타민 B 복합체가 피로 회복에 도움됩니다. 아침 식후 복용을 권장합니다.”

프로젝트 요약

Flask + GPT 기반 챗봇 시스템 구현

사용자 생체 정보를 바탕으로 맞춤형 영양제 추천

데이터베이스, 백엔드, 프론트엔드가 유기적으로 통합됨

개선 방향 및 향후 계획

추가 정보나 참고 내용을 여기에 입력하세요.

구분	개선 방향	향후 발전 계획
항목 1	영양제 데이터 고도화 <ul style="list-style-type: none">다양한 연령대와 건강상태를 반영한 DB 확장	프론트엔드 고도화: <ul style="list-style-type: none">기존 HTML UI → React UI 전환
항목 2	상담 이력 저장 기능 <ul style="list-style-type: none">개인별 상담 맥락 유지 → 연속 대화 실현	카카오톡 연동: <ul style="list-style-type: none">메시징 플랫폼과 챗봇 연결 → 접근성 향상
항목 3	다국어 지원 추가 <ul style="list-style-type: none">영어, 일본어 등 멀티랭귀지 → 글로벌 확장 가능	백오피스 시스템 구축: <ul style="list-style-type: none">영양제 데이터 자동 수집 및 관리 → 운영 효율 + 신뢰도 강화



감사합니다.

궁금하신 점 있으시면 질문 부탁드립니다

HYAI