

AI 모듈 연동 명세서

1. 기술 스택

- AI Framework: LangChain
- API 서버: LangServe/FastAPI
- Vector DB: FAISS
- LLM: OpenAI GPT-3.5-turbo

2. 프로젝트 구조

```
ai/
├── nodes/
│   ├── analyze_sleep.py #수면 시간에 따른 운동 영향
│   ├── format_output.py #최종 답변 출력 형식
│   ├── generate_cheering.py #응원 멘트 생성
│   ├── generate_diet.py #식단 생성
│   ├── generate_todo.py #todo 운동 3-4 개 생성
│   ├── search_tip.py #todo 운동 3-4 개 관련 팁 생성(rag 검색 후
검색되지 않은 부분만 gpt)
│   └── time_check.py #답변 생성 시간 체크
├── rag/
│   ├── diet_rag.py #식단 문서 검색
│   │   ├── doc_to_vector.py #문서를 벡터화(vectorstores)
│   │   ├── sleep_rag.py #수면에 따른 운동 영향 문서 검색
│   │   └── tip_rag.py #운동 팁 문서 검색
├── data/
│   ├── Effects of Diet on Sleep Quality.pdf #식단 문서(diet_rag.py)
│   ├── sleep_performance.pdf #수면에 따른 운동 영향 문서(sleep_rag.py)
│   └── workout_tips.txt #운동 팁 문서(tip_rag.py)
├── server/
│   ├── main.py
│   └── routes.py
└── Dockerfile
```

3. 환경 설정

OPENAI_API_KEY=
LANGCHAIN_API_KEY=
DATABASE_URL=

4. 주요 API

4.1 맞춤 운동/ 식단 추천 API

- POST /healthai/invoke
- POST /healthai/stream

5. RAG 시스템

문서 처리

1. 파일 로드: PyPDFLoader, TextLoader
2. 텍스트 분할: RecursiveCharacterTextSplitter
3. 임베딩 생성 : OpenAIEmbeddings
4. 벡터 저장: FAISS

5.1 수면 시간 기반 식단 RAG

5.1.1 질의응답

1. 문서 벡터화 - vectorstores 에 존재하는지 확인 후 없을 경우 문서 벡터화 진행
(*Effects of Diet on Sleep Quality.pdf*)
2. 질문 프롬프트 생성 - 사용자 수면 시간(sleep)을 바탕으로 질문 프롬프트 생성
3. 질문 임베딩 및 문서 검색 - 유사한 상위 2 개 문서 검색
4. 컨텍스트 구성 - 검색된 문서를 묶어 프롬프트로 생성

5.2 수면 시간 기반 운동 영향 RAG

5.2.1 질의응답

1. 문서 벡터화 - vectorstores 에 존재하는지 확인 후 없을 경우 문서 벡터화 진행
(*sleep_performance.pdf*)
2. 질문 임베딩 및 문서 검색 - 질문을 벡터화하여 유사한 상위 문서 검색
3. 컨텍스트 구성 - 검색된 문서 컨텍스트로 구성해 LLM 에 전달
4. LLM 답변 생성 - RetrievalQA 체인을 통해 최종 답변 생성(수면시간에 따른 운동 영향)

5.3 운동 TIP RAG

5.3.1 질의응답

1. 문서 벡터화 - vectorstores 에 존재하는지 확인 후 없을 경우 문서 벡터화 진행
(*workout_tips.txt*)
2. 질문 임베딩 및 문서 검색 - 질문을 벡터화하여 유사한 상위 문서 검색
3. 컨텍스트 구성 - 검색된 문서 컨텍스트로 구성해 LLM 에 전달
4. LLM 답변 생성 - RetrievalQA 체인을 통해 최종 답변 생성(운동 팁)

6. 벡터 DB 설정

- FAISS 인덱스: IndexFlatL2
- 임베딩 모델: text-embedding-3-small
- 청크 크기: 1000 자
- 오버랩: 200 자

7. Backend 연동

7.1 HTTP 통신

Spring Boot → AI Module

POST http://ai-server:8003/healthai/invoke

```
{
  "input": {
    "user_id": 1,
    "goal": "근력 향상",
    "diseases": ["당뇨", "고혈압"],
    "records": [
      {
        "date": "2025-06-27",
        "sleep": 6.5,
        "weight": 70.2,
        "fat": 17.5,
        "muscle": 32.1,
        "bmr": 1580,
        "bmi": 23.5,
        "vai": 1.2
      },
      {

```

```

        "date": "2025-06-28",
        "sleep": 6.5,
        "weight": 70.2,
        "fat": 17.5,
        "muscle": 32.1,
        "bmr": 1580,
        "bmi": 23.5,
        "vai": 1.2
    },
    {...7 일치 데이터}
],
"todolists": [
    {
        "date": "2025-06-27",
        "items": [
            { "todo": "유산소 50 분", "complete": true },
            { "todo": "스트레칭 10 분", "complete": false }
        ]
    },
    {
        "date": "2025-06-28",
        "items": [
            { "todo": "윗몸 일으키기 30 개", "complete": true },
            { "todo": "스트레칭 10 분", "complete": true }
        ]
    },
    {...7 일치 데이터}
],
"prompt": "오늘은 등 운동을 할거야",
"place": "헬스장"
}
}

```

AI Module → Spring Boot 응답

```

{
    "todolists": [
        {
            "todoItem": "벤치프레스 - 4 세트 x 12 회",
            "tip": "1. 바벨을 잡을 때 어깨 너비로 벌리고, 손목은 팔꿈치와  
일직선이 되도록 합니다.Wn2. 등을 벤치에 굽히지 않고 고정하며, 엉덩이와 상체는  
벤치에 밀착시킵니다.Wn3. 내려갈 때는 가슴을 힘껏 펴고, 올라올 때는 팔꿈치를

```

완전히 펴지 않고 약간 굽히며 수행합니다.Wn4. 숨을 들이마시고 바벨을 내리면서 내쉬고, 올라올 때 숨을 내쉬며 힘을 주어 수행합니다."

```
},  
{
```

"todoItem": "덤벨 플라이 - 3 세트 x 15 회",

"tip": "1. 어깨를 바닥에서 멀리 떨어지지 않도록 유지하며 팔을 약간 굽힌 상태로 시작하세요.Wn2. 팔을 천천히 벌리면서 가슴을 넓게 펴고 최대한 깊게 내려가도록 노력하세요.Wn3. 팔을 원위치할 때까지 가슴을 꼭 쥐고, 팔을 완전히 모으는 것을 목표로 하세요."

```
},  
{
```

"todoItem": "케이블 크로스오버 - 3 세트 x 12 회",

"tip": "1. 케이블 크로스오버를 할 때는 어깨 너비로 발을 벌리고 안정적인 자세를 유지해야 합니다.Wn2. 팔을 굽히고 펴는 동작을 할 때는 팔꿈치를 고정시키고 근육을 집중적으로 사용해야 합니다.Wn3. 숨을 들이마시고 팔을 앞으로 뻗어 상체를 굽힌 후 천천히 원래 자세로 돌아오는 동작을 반복해야 합니다."

```
},  
{
```

"todoItem": "체스트 프레스 - 3 세트 x 10 회",

"tip": "1. 체스트 프레스를 할 때는 어깨 너비로 바벨을 잡고 안정적인 자세를 유지해야 합니다.Wn2. 호흡을 잘 조절하며 천천히 내리고 올리는 것이 중요합니다.Wn3. 팔꿈치를 너무 벌리거나 모으지 않도록 주의해야 하며, 균형있는 힘을 유지해야 합니다.Wn4. 체스트 프레스를 할 때는 적절한 중량을 선택하여 자신의 체력에 맞게 운동을 진행해야 합니다.Wn5. 운동 후에는 스트레칭을 통해 근육을 풀어주고, 충분한 수분을 섭취하여 근육 회복을 도와야 합니다."

```
}
```

```
],
```

"diet": [

```
{
```

"breakfast": "- 닭가슴살 샐러드 (100g): 탄수화물 2g / 단백질 20g / 지방 2gWn- 아보카도 (1/2 개): 탄수화물 6g / 단백질 2g / 지방 15gWn 설명: 고단백 저지방 식단으로 근력 향상에 도움이 됩니다."

```
},
```

```
{
```

"lunch": "- 그릴 된 연어 (150g): 탄수화물 0g / 단백질 30g / 지방 10gWn- 시금치 샐러드 (100g): 탄수화물 4g / 단백질 2g / 지방 1gWn 설명: 고품질 단백질과 오메가-3 지방산이 체중 감량에 도움이 됩니다."

```
},
```

```
{
```

```
        "dinner": "- 닭가슴살 볶음 (100g): 탄수화물 1g / 단백질 25g / 지방 3g\n- 콩나물무침 (100g): 탄수화물 4g / 단백질 2g / 지방 0g\n설명: 단백질 섭취로 근육을 보호하고 체지방을 감소시킬 수 있습니다."
    },
    "cheering": "지난 7 일간의 노력이 64%의 운동 수행률을 이루었지만, 오늘도 포기하지 말고 목표에 한 걸음 더 다가갈 수 있도록 벤치프레스를 열심히 해보세요! 함께 성취해나갈 자신을 믿어요!"
}
```

8. 지원 파일 형식

- PDF
- TXT

9. LangGraph 흐름



```
graph = StateGraph(GenState)
```

```
graph.add_node("수면분석", analyze_sleep.analyze_sleep)
graph.add_node("TODO 생성", generate_todo.generate_todo)
graph.add_node("TIP 탐색", search_tip.search_tip_from_db)
graph.add_node("TIP 생성", search_tip.generate_tip_from_gpt)
graph.add_node("식단생성", generate_diet.generate_diet)
graph.add_node("멘트생성", generate_cheering.generate_cheering)
graph.add_node("출력포맷", format_output.format_output)
```

```
graph.set_entry_point("수면분석")
graph.add_edge("수면분석", "TODO 생성")
graph.add_edge("TODO 생성", "TIP 탐색")
graph.add_conditional_edges("TIP 탐색", tip_exists, {
    "yes": "식단생성",
    "no": "TIP 생성"
})
```

```
graph.add_edge("TIP 생성", "식단생성")
graph.add_edge("식단생성", "멘트생성")
graph.add_edge("멘트생성", "출력포맷")
graph.set_finish_point("출력포맷")
```

10. 테스트

10.1 단위 테스트

- 문서 처리 기능 테스트
- 벡터 검색 정확도 테스트
- API 엔드포인트 테스트

10.2 통합 테스트

- Backend-AI Module 통신 테스트
- 전체 RAG 파이프라인 테스트

11. 배포

- Docker 컨테이너(Dockerfile)

```
FROM python:3.12-slim
```

```
RUN apt-get update && apt-get install -y build-essential && rm -rf  
/var/lib/apt/lists/*
```

```
WORKDIR /app
```

```
COPY pyproject.toml poetry.lock* ./
```

```
RUN pip install --upgrade pip && pip install poetry
```

```
RUN poetry config virtualenvs.create false && poetry install --no-root
```

```
COPY . .
```

```
EXPOSE 8003
```

```
CMD ["uvicorn", "server.main:app", "--host", "0.0.0.0", "--port", "8003"]
```