



# Check-In Home

체크리스트 작성부터  
인테리어 추천까지

SSAFY 11기 광주 5반 고민호

SSAFY 11기 광주 5반 박선희

# 목차

---

- 01 WHY – 서비스의 개요
- 02 WHAT – 서비스 소개
- 03 HOW – 구현 방법
- 04 Retrospect – 기대 및 회고
- 05 Appendix

# Check-In Home

A blue-tinted photograph of a tennis court from an elevated perspective. The court is divided into three sections by a central net. Two players are visible on the outer sections. The word 'WHY' is superimposed in large white letters across the center of the court.

WHY

# 제안배경

001



## 계약 전 방을 꼼꼼히 점검해야 하는 필요성 부각

- 늘어나는 전월세사기 (2023년 사기 피해액: 4조 (출처: 동아일보))
- 실제 입주했을 시, 당시 발견하지 못했던 각종 하자로 인한 피해 발생
- 입주자의 **선제적인 예방 및 꼼꼼한 체크리스트** 관리 필요

002



## 맞춤형 인테리어 추천 서비스 수요 증가

- 내 집이 아니더라도 새 집처럼 꾸미려는 MZ세대의 욕구 증가
- 인테리어는 사용자의 취향, 예산 등에 따라 천차만별
- 사용자의 **거주환경, 라이프스타일, 선호 디자인** 등 초개인화 맞춤형 서비스 필요성 증가

# 프롭테크 서비스 분석

	네이버 부동산	직방	다방
장점	<ul style="list-style-type: none"><li>- 매물이 많음</li><li>- 필터가 가장 상세</li></ul>	<ul style="list-style-type: none"><li>- 매물이 다소 많음</li></ul>	<ul style="list-style-type: none"><li>- 검색하기 편리하고 직관적인 <b>UI</b></li></ul>
단점	<ul style="list-style-type: none"><li>- 매물 이미지 없는 경우 다수</li></ul>	<ul style="list-style-type: none"><li>- 검색이 세분화되어 있지 않음</li><li>- <b>UI</b> 불편</li></ul>	<ul style="list-style-type: none"><li>- 매물이 타사에 비해 많지 않음</li></ul>

네이버 부동산  
직방, 다방



VS.

Check-In Home

주택 검색, 중개거래 등 **주택 매매 서비스 제공**에만 치중



(

주택을 구매하기까지의 전 과정을 함께하는 동반자

)





A red lighthouse with a black metal framework stands on a rocky shore. The lighthouse has a red roof and a black lantern room. A black metal walkway extends from the lighthouse towards the right. The background is a clear blue sky and a blue ocean. The word "WHAT" is written in white capital letters across the middle of the image.

# WHAT

# 고객 여정 지도/차별점

## 1. 거주지 탐색

- 다양한 집을 검색하여 후보군 선정
- 법정동 기반 검색, Drawing기반 검색
- 매매가, 주위 편의시설 등 다양한 상세정보

## 3. 주택 방문하여 둘러보기

- 집을 방문하여 체크리스트 작성 및 비교
- 사용자 니즈에 따라 체크리스트 템플릿 제작
- 체크리스트를 작성한 아파트 한 번에 조회

## 2. 매물 검색, 판매자 연락

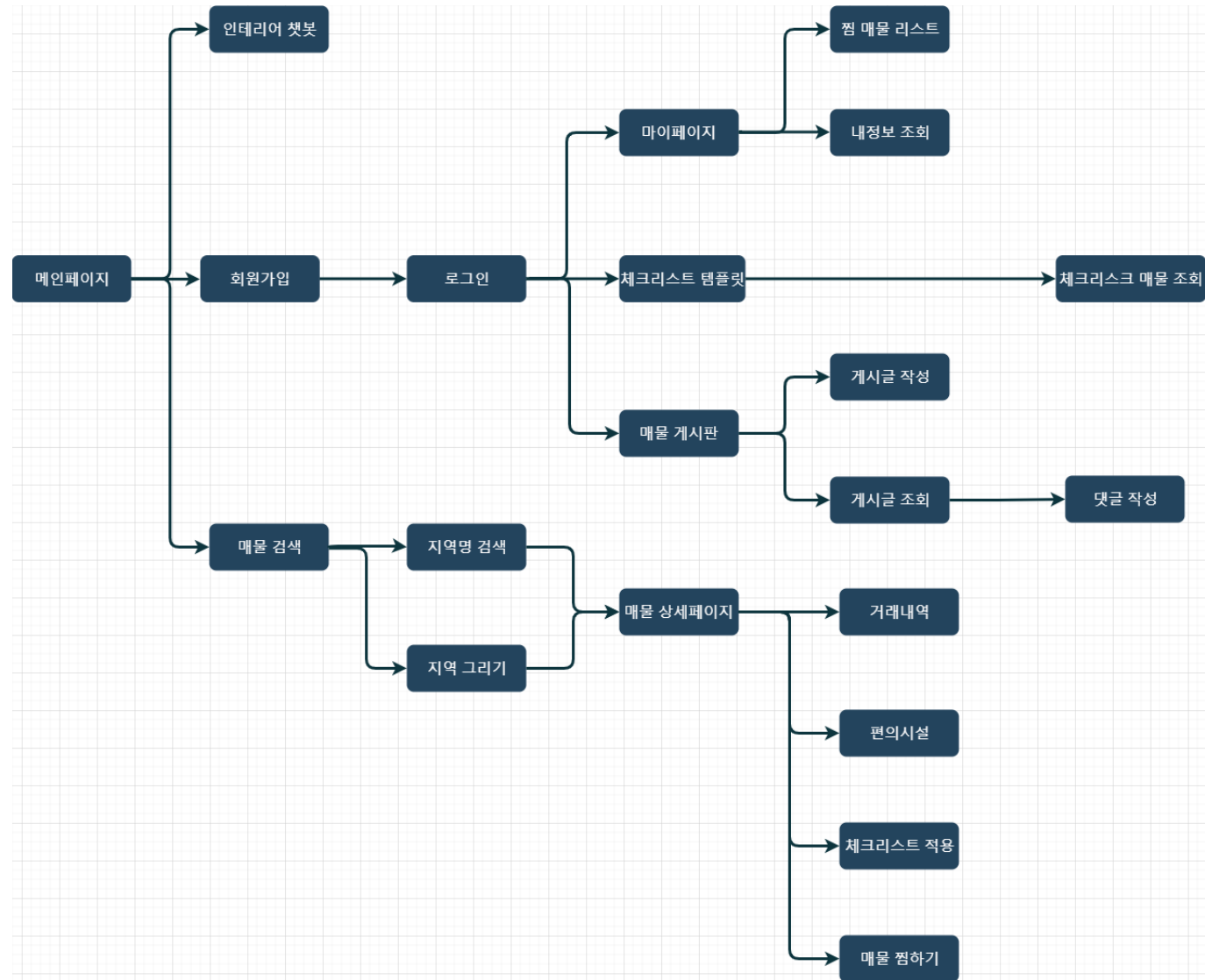
- 거래중인 매물 중에서 후보군 선정
- 매물 게시판에서 판매자와 거래
- 댓글을 통해 판매자와 소통

## 4. 인테리어 꾸미기

- 주택 구매 후 인테리어 시공
- 인테리어 추천 챗봇을 통해 맞춤형 인테리어 추천

“ All-In-One Solution ”

# Screen Flow Chart



## 체크인홈은 당신의 집을 찾고 꾸미는 과정까지의 여정을 책임집니다.

검색을 통해 아파트에 대한 정보를 얻어보세요!

[매물 검색 바로가기](#)





A serene tropical beach scene. The foreground shows a sandy beach with gentle waves washing onto it, creating white foam. The water is a vibrant turquoise color, extending to a clear horizon. The sky is a deep blue with scattered, soft white clouds. On the left side, a small strip of land features several palm trees and lush greenery. The overall atmosphere is peaceful and idyllic.

# HOW

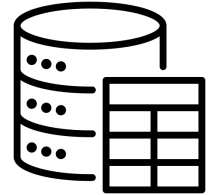
# BASE

Data & Fundamental

# 1. Base

## 1. 8개년 실거래가 데이터 직접 수집(2016 ~ 2024)

- 아파트데이터약4.5만개, 실거래데이터약400만개 직접수집및가공(활용데이터:국토교통부아파트실거래가)
- 최신화된데이터및데이터갱신을위함



## 2. Leaflet.js 지도 Library 사용

- 카카오, 네이버 지도 API를 사용하지 않고 커스텀하기 위해 오픈소스 지도 라이브러리 활용
- Polygon을 그리기 위한 *turf*, 드로잉을 하기 위한 *freeDraw* 등 third-party 라이브러리 활용



## 3. CheckList 구현

- 방을 구할 때 필요한 체크리스트 자동 관리
- 개개인 체크리스트 템플릿 관리, 회원 생성 시 자동 생성
- UserId, aptId를 기반으로 회원 체크리스트 템플릿을 아파트마다 개별 적용



# INDEX

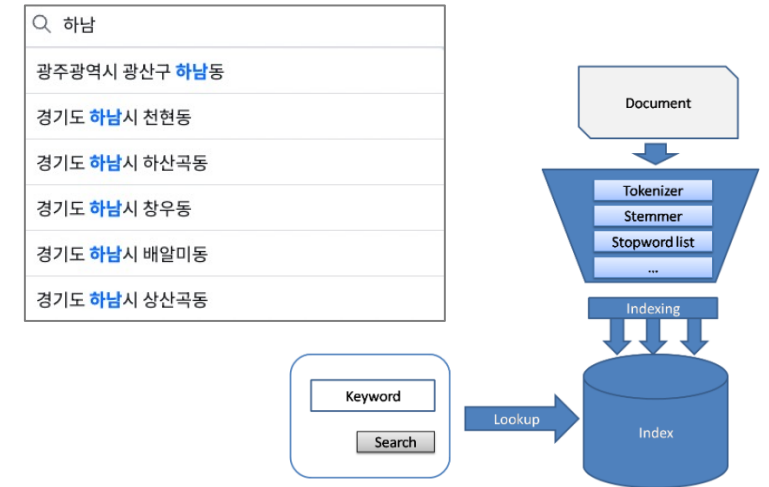
Full-Text Index / Spatial Index



# 2. Index

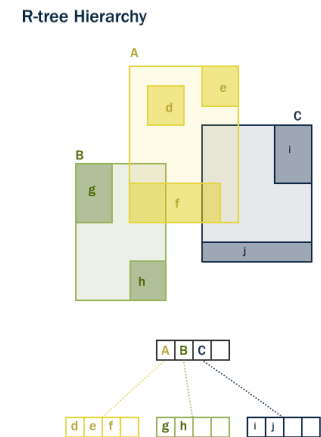
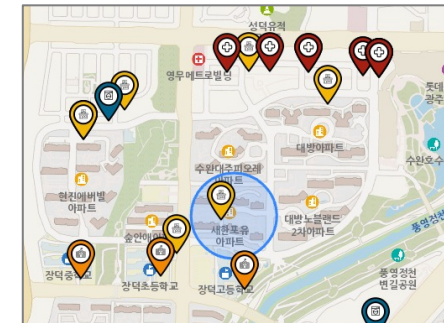
## 1. Full-Text Index

- 첫글자뿐만아닌, 중간의단어나문장으로도 인덱스를 생성하여 빠른 검색 결과제공
- MySQL에서 문자열 검색을 위해 LIKE이외에 Full-Text Search 제공
- **검색어 자동완성** 기능에 해당 Index를 적용하여 빠르고 정확한 검색결과제공



## 2. Spatial Index

- 위도와 경도로 이루어진 좌표값과 같은 공간데이터를 빠르게 검색, 활용하도록 함
- **R-Tree** 기반 알고리즘으로 작동
- **드로잉을 활용한 아파트 검색, 아파트 반경 이내 편의시설 검색** 때 사용
- 10만개의 편의시설데이터에서 반경 500m 이내의 편의시설을 검색하는데 10배 이상 시간 단축  
(공간인덱스 사용 전: 0.425초 -> 사용 후: 0.04초)



# RAG

Retrieval Augmented Generation

### 3. Retrieval Augmented Generation



「*“내가 지금 원룸에 살고 있는데,  
공간감있는 인테리어 추천해줘”*  
라고 묻는다면?

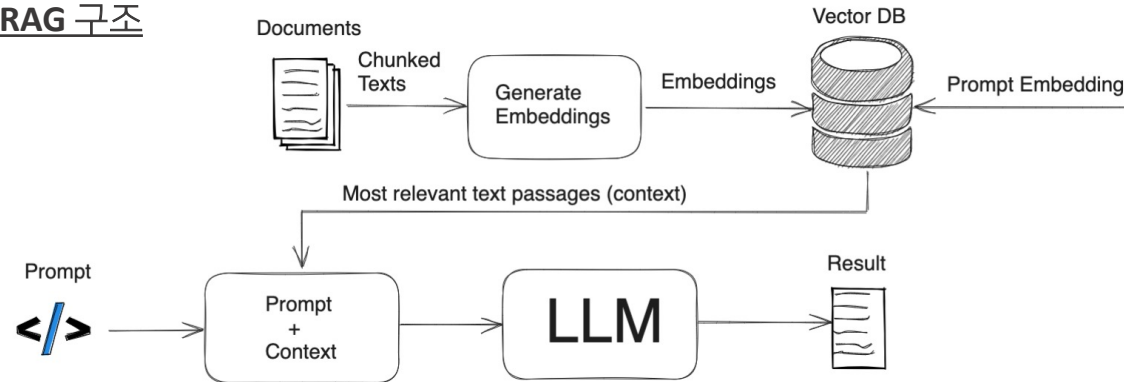
」

# 3. Retrieval Augmented Generation

## RAG(Retrieval Augmented Generation)

- **맞춤형 데이터**를 활용하여 대규모 언어 모델(LLM) 애플리케이션의 효율성을 개선할 수 있는 아키텍처 접근 방식
- 모델을 다시 훈련할 필요 없이 LLM의 원래 지식 기반 보완함으로써 검색 결과의 정확도 개선
- LLM의 Hallucination(환각) 현상을 예방

### \* RAG 구조



출처 : <https://wandb.ai/cosmo3769/RAG/reports/A-Gentle-Introduction-to-Retrieval-Augmented-Generation-RAG---VmIldzo1MjM4Mjk1>

RAG를 활용하여 **인테리어 추천 챗봇** 구현

# 3. Retrieval Augmented Generation

## 프로젝트 적용

### 인테리어 데이터 수집

약 2,700개의 데이터 수집

### Embedding 추출

 **Hugging Face**

컨텐츠 제목, 내용을 기반으로  
임베딩 벡터 추출

### 데이터 적재

 **MongoDB.**  
Atlas

Vector Database로 활용

### Prompt Engineering




인테리어 챗봇 프롬프트 튜닝

### LangChain 기반 서버 구축

 **LangChain**

LLM과 애플리케이션 통합 간소화

 인테리어에 대한 모든 것을 물어보세요!  
Online

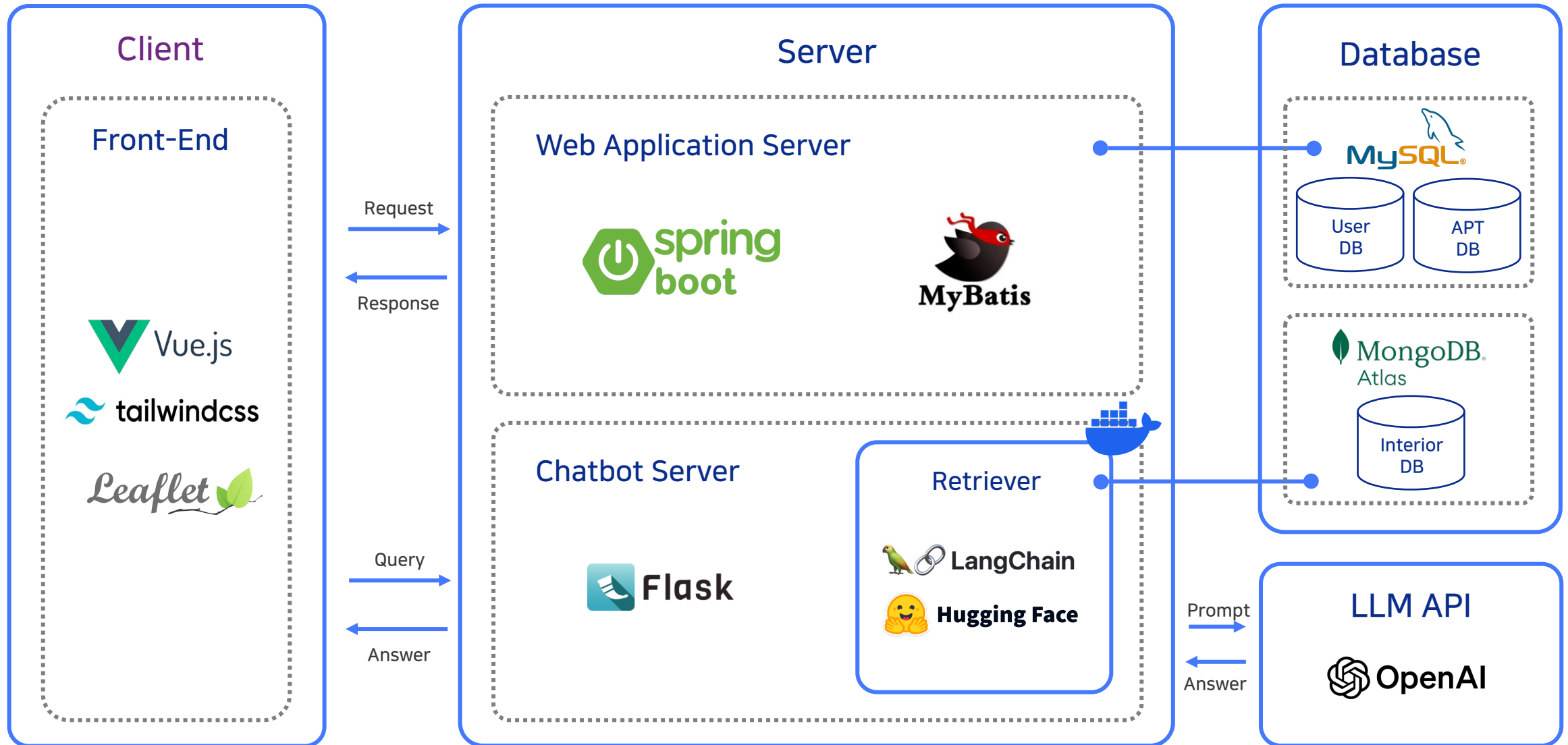
안녕하세요. 무엇을 도와드릴까요?

웜톤 인테리어 추천해줘



### 인테리어 챗봇 구현

# 서비스 아키텍처



# 일정관리

## 일정 관리

### 체크인 홈

고민호 박선희

📅 타임라인 🔥 By Priority \* All projects

» 2024년 5월

📅 Notion 캘린더에서 열기

주 < 오늘 >

화 14

수 15

목 16

금 17

토 18

일 19

월 20

화 21

수 22

목 23

← JWT 로그인/로그아웃

아파트 거래내역 및 상세정보 페이지

매물 게시판 및 댓글

인테리어 챗봇

마이페이지 및 매물 찜하기

체크리스트

TailWind Css 디자인

+ 새로 만들기



# Retrospect



# 기대효과



체크리스트 활용하여  
올바른 의사결정에 도움



챗봇을 통한  
맞춤형 인테리어 추천



다양한 검색 기능 등으로  
사용자 편의성 극대화

# 회고

“ 이번 관통 프로젝트를 하면서 소감 한마디 말씀해주세요 ”



고민호

짧다면 짧고 길다면 긴 일주일동안 개발하며 힘들지만 결과물을 보며  
즐거움을 느낄 수 있었습니다

수업시간에 배운 프레임워크 및 다양한 기술을 활용하여 재미있고  
의미있는 서비스를 만들 수 있었습니다.



박선홍



# Appendix

# Prompt

1. You are an interior chatbot based on the database of our interior site, which lets you know which interior is appropriate when users ask questions.
2. you must analyze user needs and recommend the most suitable interior design.
3. You are given two interior content. Analyze and you can recommend the most appropriate content or you can both recommend two content if two content is highly related two the user query.
4. you must the following
  - 4.1 Most of all, consider the area of the house where the user lives.
  - 4.2 Next, consider what kind of interior and furniture the user thinks of.
  - 4.3 Next, consider the user's occupation and lifecycle.
  - 4.4 As the last subordinate, consider the overall color and tone that the user wants.
5. When generating an answer, you should keep in mind as follows.
  - 5.1 You should answer the interior design adequated to user's preference. In this case, you should retrieving the information in the DB.
  - 5.2 And you should answer by connecting your answer with the interior posts in the DB.

Example) You say that you like to watch movies, how about the interior using a beam projector? The post uploaded by Ronald say that he decorated the room using interior and so he can watch movies on the bed through a beam projector.
6. If you feel that the information is insufficient, you should ask additional questions for more information. It is not right to make inaccurate interior recommendations. Keep in mind that it may cause damage to your company.
  - 6.1 At this time, simply don't ask. And never ask after taking out unnecessary information from the database you are retrieving. The information in the database is not related to the user you are currently asking.
7. When you generate a reply, you should add the enter key('\n') every sentence to seperate the sentence.
8. This content of the database you are retrieving is not corrlated the user you are chatting. So, Do not refer the anything like name with using the database content.
9. you can chat freely if the query is not correlated with interior. Make PingPong like other chatbot
10. You Should refer the content title which content you refer.
11. You should should answer in Korean It's important because the user is korean.

## [Information]

## {context}

## [Task]

## {question}

1~2 챗봇의 역할 설정

3~6 : 인테리어 추천 규약 설정

7~11 : 기타 답변 형식 규약 등

\* 영어로 **Prompting** 하는 이유?

=> 더 높은 성능을 기대할 수 있음

(GPT3 기준 학습 데이터셋의 90% 이상이 영어 – 출처)

# Chatbot - Server

```
def question(query_text, prompt_template):
    try:
        # OpenAI LLM 설정
        if not OPENAI_API_KEY:
            raise ValueError("OPENAI_API_KEY 환경 변수를 설정하세요.")
        # 환경 변수와 초기 설정 로직...

        if not MONGODB_URI:
            raise ValueError("MONGO_URI 환경 변수를 설정하세요.")

        print("embedding function")
        print(MONGODB_URI)
        print(MONGODB_DATABASE + "." + MONGODB_COLLECTION)
        print(type(model_huggingface))
        # 아래 DB 만 변경
        vector_search = MongoDBAtlasVectorSearch.from_connection_string(
            MONGODB_URI,
            MONGODB_DATABASE + "." + MONGODB_COLLECTION, # 사용 collection
            model_huggingface, # 사용 모델
            index_name="default", # 사용 index
            embedding_key="embeddings", # 사용 임베딩 키
            text_key="content" # 텍스트
        )
        print("mongodb conn")

        # Retriever 설정
        qa_retriever = vector_search.as_retriever(search_type="similarity", search_kwargs={"k": 2}) # 47~53.2
        print("retriever complete")
        # 어떤 LLM 사용할 것인지
        llm = ChatOpenAI(api_key=OPENAI_API_KEY, model="gpt-3.5-turbo") # 또는 원하는 OpenAI 모델 사용
```

```
PROMPT = PromptTemplate(
    template=prompt_template, input_variables=["context", "question"]
)
print("llm complete")
qa = RetrievalQA.from_chain_type(
    llm=llm,
    chain_type="stuff",
    retriever=qa_retriever,
    return_source_documents=True, # 소스 문서 반환 설정
    chain_type_kwargs={"prompt": PROMPT},
    verbose=True,
)
print("qa complete")
# 전송 및 응답 받아옴
result = qa({"query": query_text})
# Answer
answer = result['result']
# 어떤 문서 참고했는지!
source_documents = result['source_documents']

urls = [doc.metadata['url'] for doc in source_documents if 'url' in doc.metadata]
titles = [doc.metadata['title'] for doc in source_documents if 'title' in doc.metadata]
images = [doc.metadata['image'] for doc in source_documents if 'image' in doc.metadata]

# 챗봇 결과 반환
return {
    "answer": answer,
    "urls": urls,
    "titles": titles,
    "images": images,
}

except Exception as e:
    return {"error": f"오류 발생: {e}"}
```

## - Chatbot 서버의 RAG 부분 코드 캡처



A hand with five fingers extended, showing colorful nail polish: thumb is purple, index is blue, middle is yellow, ring is purple, and pinky is blue. A black wristwatch with a black strap and a black face with colorful dots is visible on the wrist. The background is a solid blue color. The text 'Q&A' is overlaid in white, bold, sans-serif font in the center of the image.

Q&A

감사합니다