

# 서집사

20 ~ 30대 1인 가구를 위한 AI 추천 부동산 웹 서비스



2025.01.13

팀장 김정훈

# 목차

---

## 01 팀 소개

- SKN04-FINAL-1Team
- 

## 03 진행 현황

- 진행 방식
  - 데이터 수집부터 전처리까지
  - 모델 현황
- 

## 05 프로젝트 시연

---

## 02 프로젝트 개요

- 주제 및 기획 의도
  - 서비스 주요 기능
- 

## 04 진행 계획

- 남은 일정
  - 최종 비즈니스 모델
-

## SKN04-FINAL-1Team



**팀장 김정훈**

- Backend
- Data



**팀원 김태욱**

- Frontend



**팀원 박진호**

- AI (프롬프트 설계 및 결과 최적화)



**팀원 박화랑**

- AI (LangGraph 설계 및 LLM 작동 구조 구현)

# 프로젝트 배경

# 대학 기숙사 1인실 전환

실시간

동아대, 지방대학 활성화 사업 총합평가 A등급

검색어를 입력해 주세요

권익위 "대학 기숙사 다인실, 1인실로" 개선 권고

| 1인실로 개선하면 대학평가에 반영

윤광원 기자

| news@dhinsight.co.kr

기자의 다른기사 보기

f

t

m

l

u

s

[이슈]

입력 2024.11.04 14:30

수정 2024.11.04 14:42

댓글 0

+

가

-

## □ 내 · 외부 동향

- 1) 국민권익위원회의 국민생각함 설문조사에 의하면 학생 응답자의 94.3%가 개인공간 및 프라이버시 추구 등을 이유로 1인실을 선호한다고 답하였음
- 대한민국 정책브리핑, (2024.11.04.), “대학 기숙사, 독립공간 보장하는 공유형으로”, (https://www.korea.kr/news/policyNewsView.do?newsId=148935810)

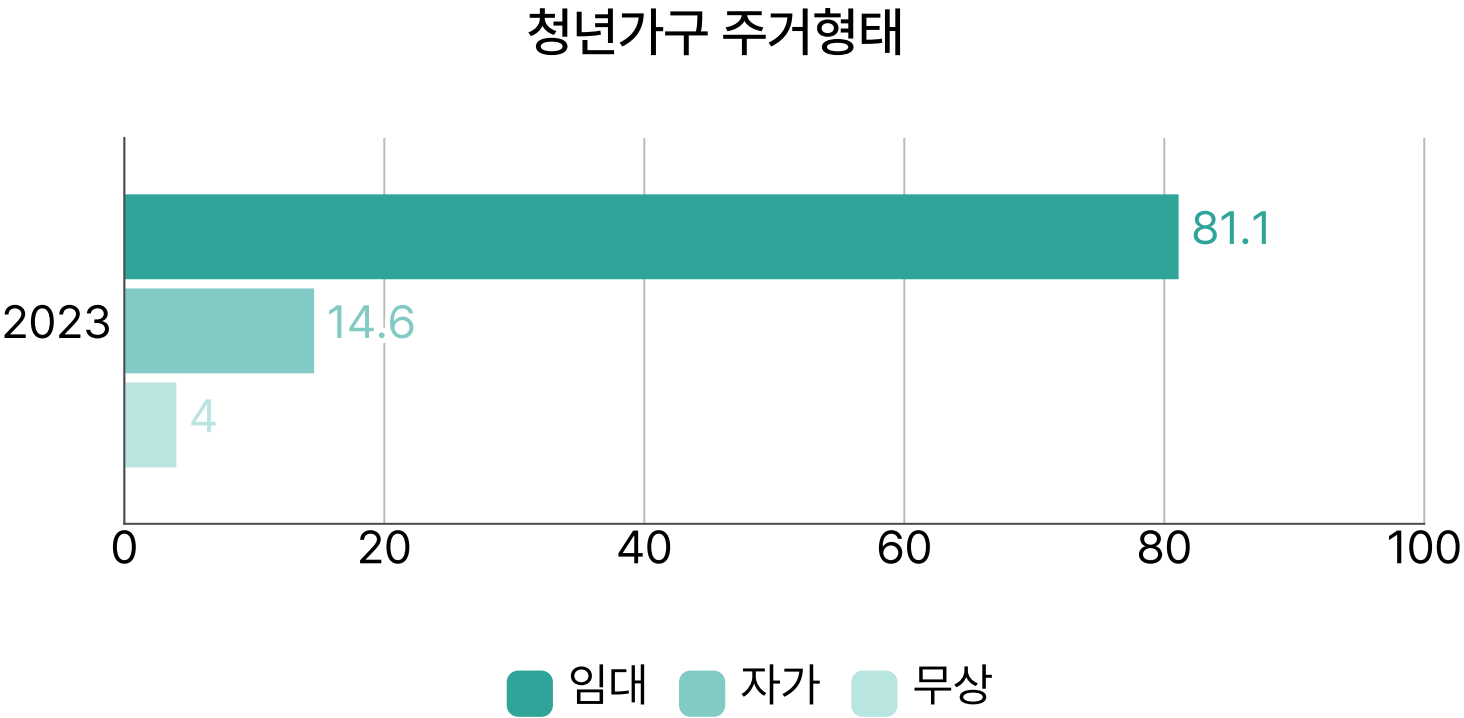
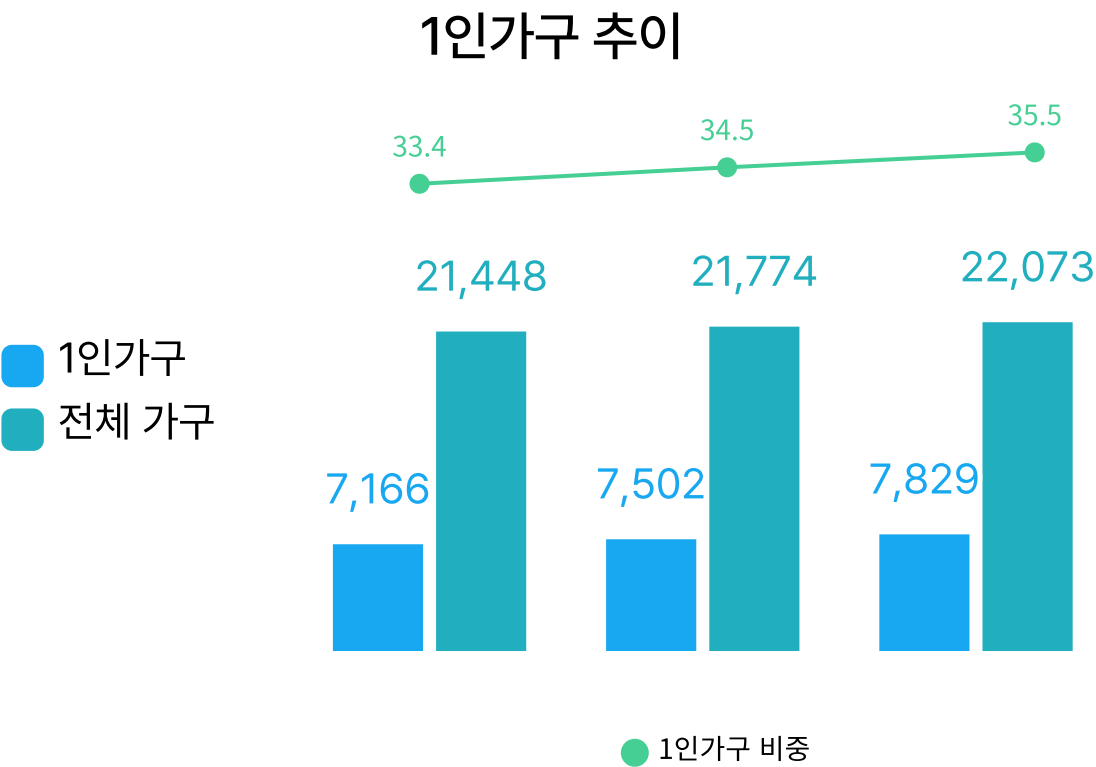
HOME > 뉴스 > 경제

## 기숙사도 혼자 살고 싶은 MZ세대, 독립 생활공간 확대된다

차미경 | 승인 2024.11.05 15:06 | 댓글 0

| ‘대학기숙사생활관 주거환경 개선 방안’ 의결...교육부 등에 권고

# 프로젝트 배경



## 계속되는 1인가구의 증가

- 통계청이 발표한 '2024 통계로 보는 1인 가구'에 따르면, 2023년 기준 1인 가구는 전체 가구의 35.5%를 차지하며, 전년 대비 1% 증가해 가장 높은 비율을 기록하고 있습니다.
- 특히, 20~30대 1인 가구는 전체 1인 가구의 약 36%를 차지합니다.

## 20~30의 주거형태

- 청년가구는 임차로 거주하는 비율이 81.1%로 매우 높음
- 특히, 수도권 지역에 거주하는 청년가구의 경우 임차 비율이 83.3%로 매우 높음

## 꾸준한 1인가구 부동산 관심

- 1인 가구가 전체 가구의 35.5%를 차지하며, 특히 20~30대 청년층은 임차 비율이 81.1%로 높아 소형 주택 및 임대 시장의 성장 가능성이 보임
- 수도권 중심의 주거 수요는 지역경제 활성화와 맞춤형 주택 공급 확대를 이끌 것으로 기대

# 프로젝트 주제 및 주요 기능

복잡한 검색 과정 해소

여러 단계의 검색 과정을 거쳐야 하는 번거로움을 해소하고, 대화형 인터페이스를 통해 원하는 매물을 쉽게 찾을 수 있도록 지원합니다.

용어의 어려움 해결

- 부동산 관련 용어는 전문적이고 복잡해 유저들이 이해하기 어려움
- 어려운 용어를 쉬운 언어로 설명하고, 예시를 제공하여,부동산 정보를 쉽게 이해할 수 있도록 돕습니다.



LLM 활용 부동산 추천 웹 서비스

- AI 기반 맞춤형 추천
  - 개인의 선호도, 라이프스타일, 예산 등을 고려하여 최적의 부동산을 추천합니다.
- 대화형 인터페이스
  - 자연어 처리 기술을 통해 유저와 대화하며 원하는 정보를 쉽게 찾도록 도와줍니다.
- 데이터 기반 인사이트 제공
  - 데이터 분석을 통해 부동산 시장 트렌드와 유저 선호도를 파악하고, 유용한 정보를 제공합니다.
- 어려운 용어 설명
  - 부동산 용어에 대한 설명과 예시를 제공하여 유저의 이해를 돕습니다.

# 타 부동산과의 차별점

유저가 직접 조건을 입력하고  
수동으로 매물을 검색해야 함.

일반적인 조건(지역, 평수, 가격)에 따라  
일괄적인 검색 결과를 제공

전문적인 용어와  
복잡한 매물 정보를 그대로 보여줌.

방대한 매물 정보가 나열되어 유저가  
필요 없는 정보까지 확인해야 함.

단순 매물 정보 나열에 그침.

유저가 원하는 조건 입력 시 매물을 즉시 추천  
불필요한 검색 단계를 줄임

LLM 기반 AI가 예산, 관심 지역등을 학습해 개인화된 추천 제공.  
상호작용을 통해 시간이 지날수록 정확도 향상.

초보자도 쉽게 이해할 수 있도록  
어려운 부동산 용어를 쉬운 언어로 수정 기능 제공.

맞춤형 필터링 : 유저의 요구에 맞는 정보만 제공.  
요약 및 하이라이트: 매물의 핵심 정보만 간추려 보여주어  
빠르게 판단 가능.

유저 데이터를 분석해 부동산 시장 트렌드와 선호도를 파악.  
이를 통해 맞춤형 추천 개선과 더불어 새로운 매물 수요를 발굴.

# 일정관리



	12월				1월				2월
진행사항	1주차	2주차	3주차	4주차	1주차	2주차	3주차	4주차	1주차
요구사항정의서									
프로젝트기획서									
데이터 수집 및 저장									
데이터 전처리 및 모델링 평가									
시스템아키텍처 및 테스트									
모델 배포 및 미비점 보완									
소스코드 업로드 및 발표 자료									
프로젝트 소스코드, 최종 소스코드 깃허브 업로드									



문서관리



Confluence

홈

최근

스페이스

팀

앱

템플릿

만들기

결제

세부 정보 추가

검색

SKN04-FINAL-1Team

모든 콘텐츠

캘린더

스페이스 설정

콘텐츠

제목으로 검색

프로젝트 산출물

1주차

2주차

프로젝트 기획서 초...

AI Scenario 초안

3주차, 4주차

백엔드

프론트엔드

AI

5주차

백엔드\_02

게시함 — v. 4

SKN04-FINAL-1Team / ERD 수정 및 스키마 1 JIRA 링크

진행 중

ERD 수정 및 스키마

소유자: 김정훈(jeonhun6084) 1월 03, 2025에 마지막 업데이트 • 조회한 사용자 6명

댓글

Address

+id: Integer  
+name: String  
+latitude: Float  
+longitude: Float  
+created\_at: DateTime

API Search Data

+id: Integer  
+api\_name: String  
+parameter: String  
+response: Text  
+created\_at: DateTime

Cultural Facility

+facility\_id: Integer  
+address\_id: Integer  
+facility\_name: String  
+origin\_date: String  
+end\_date: String  
+created\_at: DateTime

Cultural Festival

+festival\_id: Integer  
+address\_id: Integer  
+festival\_name: String  
+origin\_date: String  
+end\_date: String  
+created\_at: DateTime

Subway Station

+station\_id: Integer  
+address\_id: Integer  
+line\_info: String  
+created\_at: DateTime

Crime Stats

+stat\_id: Integer  
+address\_id: Integer  
+reference\_date: String  
+total\_population: Float  
+crime\_category: String  
+crime\_subcategory: String  
+incident\_count: Integer  
+crime\_rate: Float  
+created\_at: DateTime

Population Stats

+stat\_id: Integer  
+address\_id: Integer  
+collection\_time: DateTime  
+population\_level: String  
+population\_message: Text  
+max\_population: Integer  
+male\_ratio: Float  
+female\_ratio: Float  
+non\_resident\_ratio: Float  
+created\_at: DateTime

Location Distance

+id: Integer  
+property\_id: Integer  
+address\_id: Integer  
+distance: Float

Location

+property\_id: Integer  
+location\_id: Integer  
+vdo: String  
+gungu: String  
+dong: String  
+latitude: DECIMAL  
+longitude: DECIMAL

Property

+property\_id: Integer  
+property\_location\_id: Integer  
+property\_type: Enum  
+description: Text  
+status: JSON  
+heating\_type: Enum  
+room\_count: Integer  
+bathroom\_count: Integer  
+parking\_count: Integer  
+floor: Integer  
+total\_area: Integer  
+land\_area: DECIMAL  
+exclusive\_area: DECIMAL

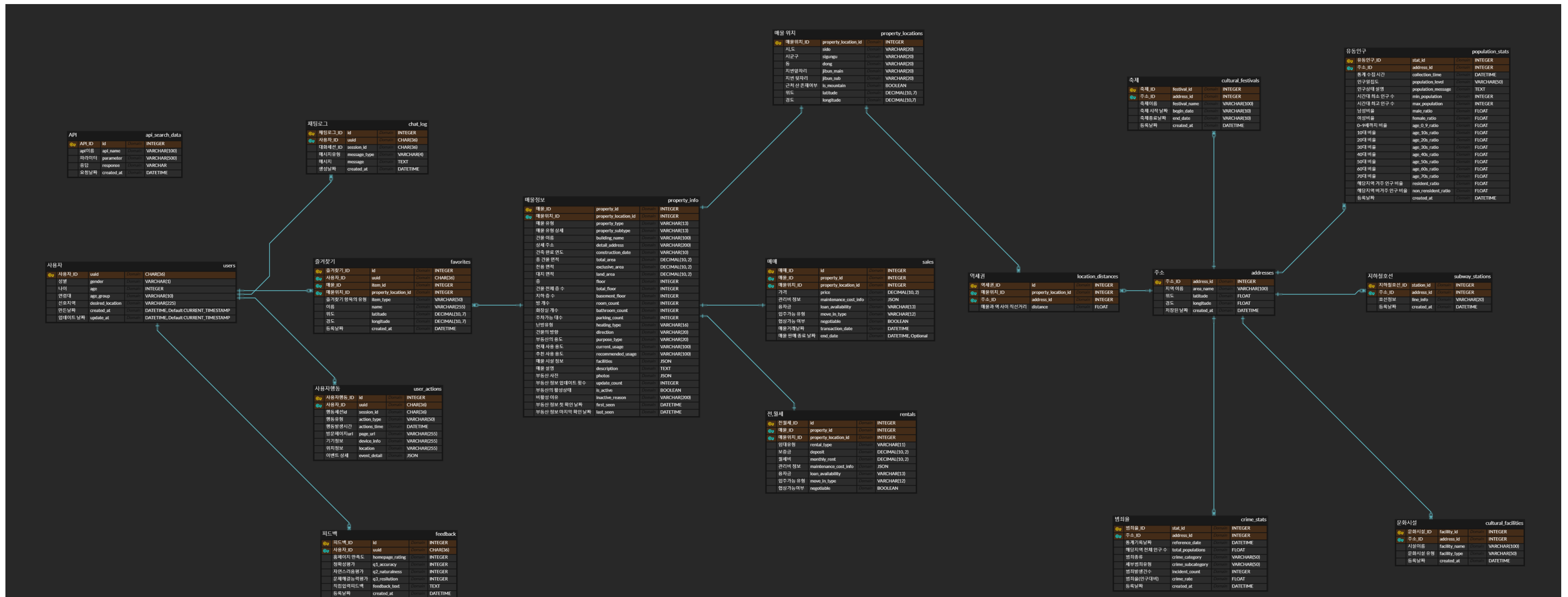
User

+id: UUID  
+gender: Enum  
+age: Integer  
+desired\_location: String  
+created\_at: DateTime  
+updated\_at: DateTime

9.page

### 03. 진행현황

# 데이터 수집



# 데이터 조회



DBeaver 24.1.5 - crime\_stats

File Edit Navigate Search SQL Editor Database Window Help

Database Navigator

Enter a part of object name here

Tables

- addresses
- api\_search\_data
- chat\_logs
- crime\_stats
- cultural\_facilities
- cultural\_festivals
- favorites
- feedback
- location\_distances
- notices
- population\_stats
- property\_info
- property\_locations
- rentals
- sales
- subway\_stations
- user\_actions
- users

Views

Indexes

Sequences

Table Triggers

Data Types

postgres - django-practice-test.cf6koesacoh6.ap-northeast-1.amazonaws.com

real\_estate.db

Properties Data ER Diagram

crime\_stats

Enter a SQL expression to filter results (use Ctrl+Space)

	stat_id	address_id	reference_date	total_population	crime_category	crime_subcategory	incident_count	crime_rate	created_at
1	1	5,974	20250103	224,653.7104	강력범죄	살인기수	2	0.8902590553	2025-01-03 08:4
2	2	5,974	20250103	224,653.7104	강력범죄	살인미수등	1	0.4451295277	2025-01-03 08:4
3	3	5,974	20250103	224,653.7104	강력범죄	강도	5	2.2256476384	2025-01-03 08:4
4	4	5,974	20250103	224,653.7104	강력범죄	강간	25	11.1282381918	2025-01-03 08:4
5	5	5,974	20250103	224,653.7104	강력범죄	유사강간	10	4.4512952767	2025-01-03 08:4
6	6	5,974	20250103	224,653.7104	강력범죄	강제추행	204	90.806423645	2025-01-03 08:4
7	7	5,974	20250103	224,653.7104	강력범죄	기타 강간 강제추행등	0	0	2025-01-03 08:4
8	8	5,974	20250103	224,653.7104	폭력범죄	상해	127	56.5314500143	2025-01-03 08:4
9	9	5,974	20250103	224,653.7104	폭력범죄	폭행	905	402.8422225427	2025-01-03 08:4
10	10	5,974	20250103	224,653.7104	폭력범죄	협박	116	51.6350252099	2025-01-03 08:4
11	11	5,974	20250103	224,653.7104	폭력범죄	폭력행위등	32	14.2441448855	2025-01-03 08:4
12	12	5,974	20250103	224,653.7104	폭력범죄	손괴	332	147.7830031869	2025-01-03 08:4
13	13	5,974	20250103	224,653.7104	지능범죄	사기	2,032	904.5032002285	2025-01-03 08:4
14	14	5,975	20250103	201,218.5746	강력범죄	살인기수	0	0	2025-01-03 08:4
15	15	5,975	20250103	201,218.5746	강력범죄	살인미수등	3	1.4909160379	2025-01-03 08:4
16	16	5,975	20250103	201,218.5746	강력범죄	강도	6	2.9818320759	2025-01-03 08:4
17	17	5,975	20250103	201,218.5746	강력범죄	강간	30	14.9091603793	2025-01-03 08:4
18	18	5,975	20250103	201,218.5746	강력범죄	유사강간	4	1.9878880506	2025-01-03 08:4
19	19	5,975	20250103	201,218.5746	강력범죄	강제추행	182	90.4489063009	2025-01-03 08:4
20	20	5,975	20250103	201,218.5746	강력범죄	기타 강간 강제추행등	0	0	2025-01-03 08:4
21	21	5,975	20250103	201,218.5746	폭력범죄	상해	127	63.1154456056	2025-01-03 08:4
22	22	5,975	20250103	201,218.5746	폭력범죄	폭행	1,001	497.4689846551	2025-01-03 08:4
23	23	5,975	20250103	201,218.5746	폭력범죄	협박	131	65.1033336562	2025-01-03 08:4
24	24	5,975	20250103	201,218.5746	폭력범죄	폭력행위등	40	19.8788805057	2025-01-03 08:4
25	25	5,975	20250103	201,218.5746	폭력범죄	손괴	286	142.1339956157	2025-01-03 08:4
26	26	5,975	20250103	201,218.5746	지능범죄	사기	2,094	1,040.6593944732	2025-01-03 08:4
27	27	5,976	20250103	259,461.1516	강력범죄	살인기수	1	0.3854141531	2025-01-03 08:4
28	28	5,976	20250103	259,461.1516	강력범죄	살인미수등	2	0.7708283062	2025-01-03 08:4
29	29	5,976	20250103	259,461.1516	강력범죄	강도	3	1.1562424592	2025-01-03 08:4
30	30	5,976	20250103	259,461.1516	강력범죄	강간	45	17.3436368884	2025-01-03 08:4

Refresh Save Cancel Export data 200 200+ 200 row(s) fetched - 0.002s (0.001s fetch), on 2025-01-06 at 14:33:43

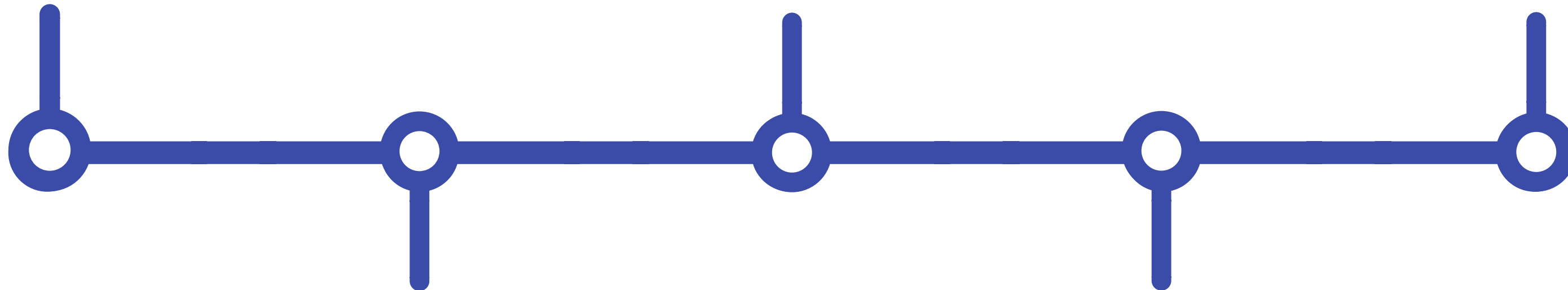
KST en

## 데이터 전처리

1. 결측값 처리

3. 중복제거

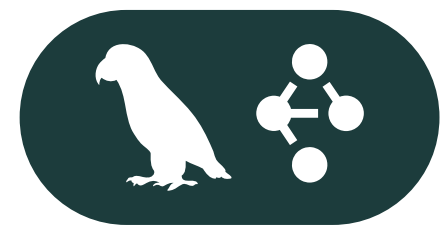
5. 데이터 저장



2. 데이터 변환

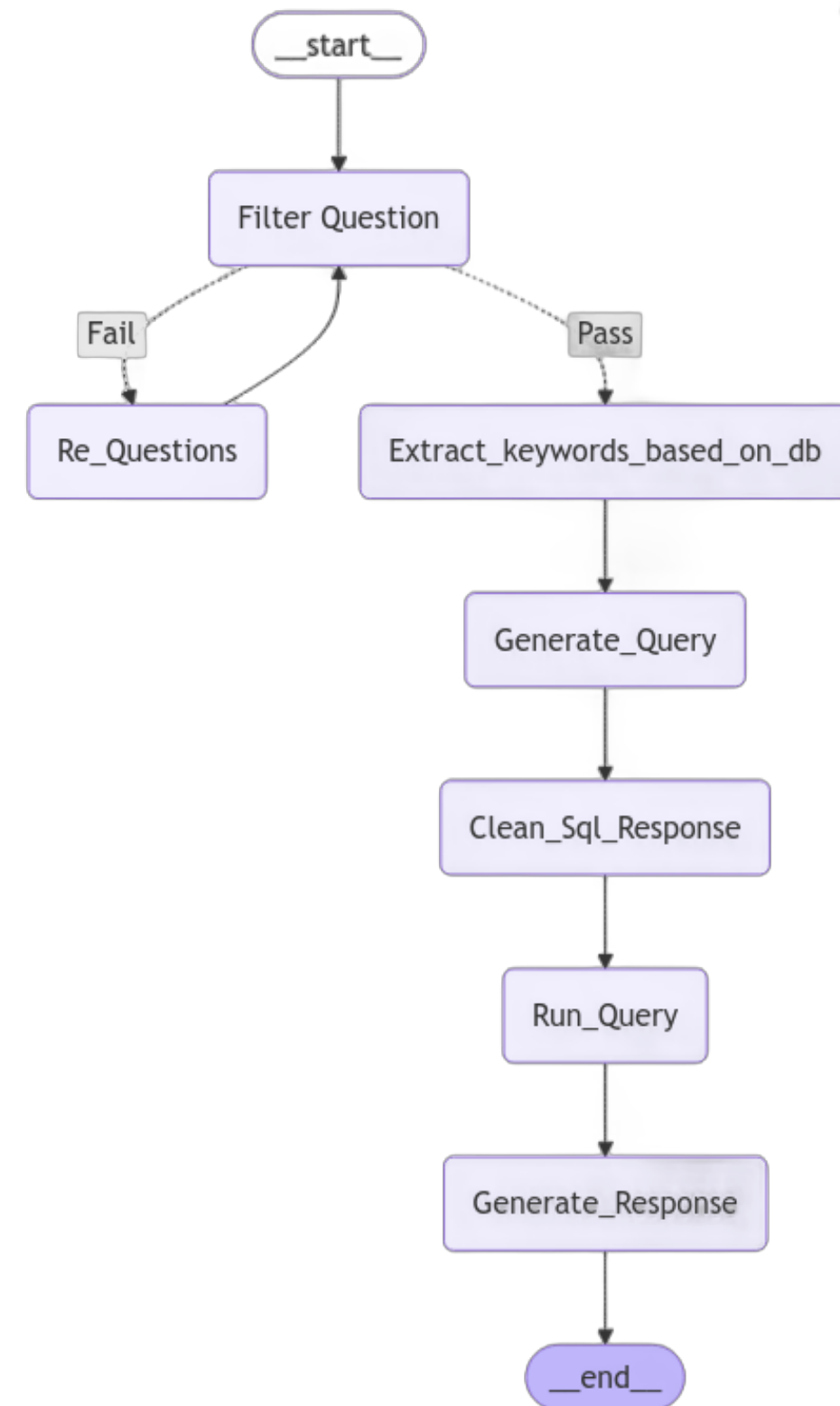
4. 데이터 검증

## 모델 설계



# LangGraph

## 그래프 단계별로 진행



# 모델 설계



## 1. Filter Node with OpenAI

```
21
22 ## 필터링 노드
23 def filter_node(state: RealEstateState) -> RealEstateState :
24     filter_prompt = PromptTemplate(
25         input_variables=["questions"],
26         template=""" '{questions}'이 부동산과 관련된 질문인지를 파악하고 다음과 같은 규칙을 지켜라
27         1. 부동산 관련 질문이 아니면 'None'을 반환해라
28         2. 부동산 관련 질문 중 에서 매매면 '매매'를 반환해라
29         3. 전세 그리고 월세랑 관련된 질문이면 '전월세'를 반환해라.
30         """
31     )
32     prompt=filter_prompt.format(questions=state["questions"])
33     response = llm(messages=[{"role": "system", "content": prompt}])
34     real_estate_type = response.content
35     return RealEstateState(real_estate_type=real_estate_type)
```

기존 프롬프트

필터 단계에서 사용자의 질문을 '한번에'  
매매와 전월세로 분리

```
filter_system_prompt : |
Classify if a given question is related to real estate. If the question is related to topics such as property transactions, re

# Output Format
- Return Pass if the question is real estate-related, otherwise return Fail.

# Examples
- 입력: '서울 아파트 매매 가격이 어떻게 되나요?'
  출력: Pass

- 입력: '이 동네 전세 시세 알려주세요'
  출력: Pass

- 입력: '서울에서 월세 계약 조건이 어떻게 되나요?'
  출력: Pass

- 입력: '서울 아파트 매매가 얼마인가요?'
  출력: Pass

- 입력: '대치동에서 버스정류장과 지하철이 가장 가까운 곳으로 알려줘'
  출력: Pass

- 입력: '여자 혼자 살기 좋은 곳 추천해줘'
  출력: Pass
```

바뀐 프롬프트

부동산 질문 여부이지만 필터링  
다양한 부동산 질문대응

## 모델 설계

## 2. Extract Keywords Node with OpenAI

```
keyword_system_prompt : |
Extract relevant keywords from a user's question based on the database schema. Match the keywords to the database schema.

1. **Location**:
- Table: `Location`
- Columns: `sido`, `sigungu`, `dong`

2. **Property Type**:
- Table: `Property`
- Column: `property_type` (e.g., apartment, villa, officetel)

3. **Price**:
- Tables: `Sale`, `Rental`
- Columns: `price`, `deposit`, `monthly_rent`

4. **Transaction Type**:
- Tables: `Sale`, `Rental`
- Columns: `rental_type` (e.g., 매매, 전세, 월세)

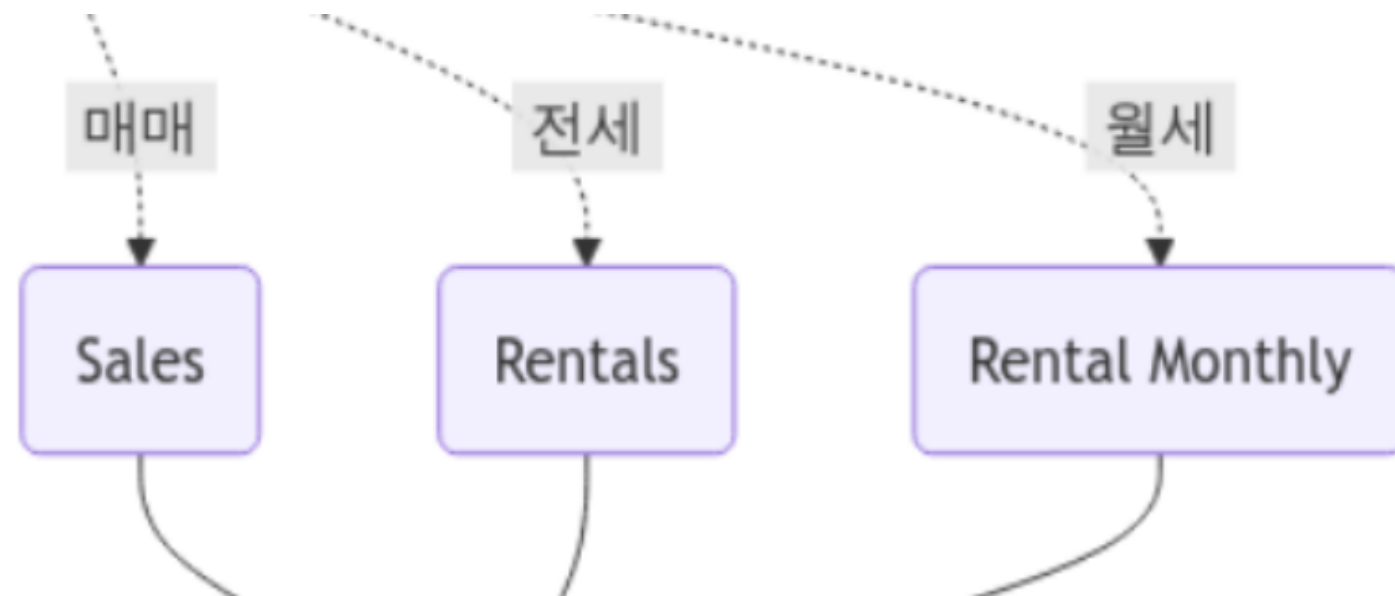
5. **Property Features**:
- Table: `Property`
- Columns: `room_count`, `bathroom_count`, `parking_count`, `exclusive_area`, `floor`
```

사용자 질문이 들어오면  
종류별 키워드를 추출함



## 모델 설계

### 3. Generate Query Node with OpenAI



기존 Generate Query Node

기존에는 추출된 키워드로 매매, 전세, 월세  
 다 다르게 쿼리문을 생성  
 => 반복되는 프롬프트가 많음

```
def generate_query(state: RealEstateState) -> RealEstateState:
    print("[generate_query] 열심히 데이터베이스 쿼리문을 작성중입니다...")

    if state['keywordlist'] == '매매':
        prompt = prompts['base_prompt'] + prompts['sales_prompt']
        table = db.get_table_info(table_names=["addresses", "sales", "property"])
        prompt = prompt.format(
            table = table,
            top_k=5,
            user_query=state['messages'][-1].content
        )
    else:
        prompt = prompts['base_prompt'] + prompts['rentals_prompt']
        table = db.get_table_info(table_names=["addresses", "rentals", "property"])
        prompt = prompt.format(
```

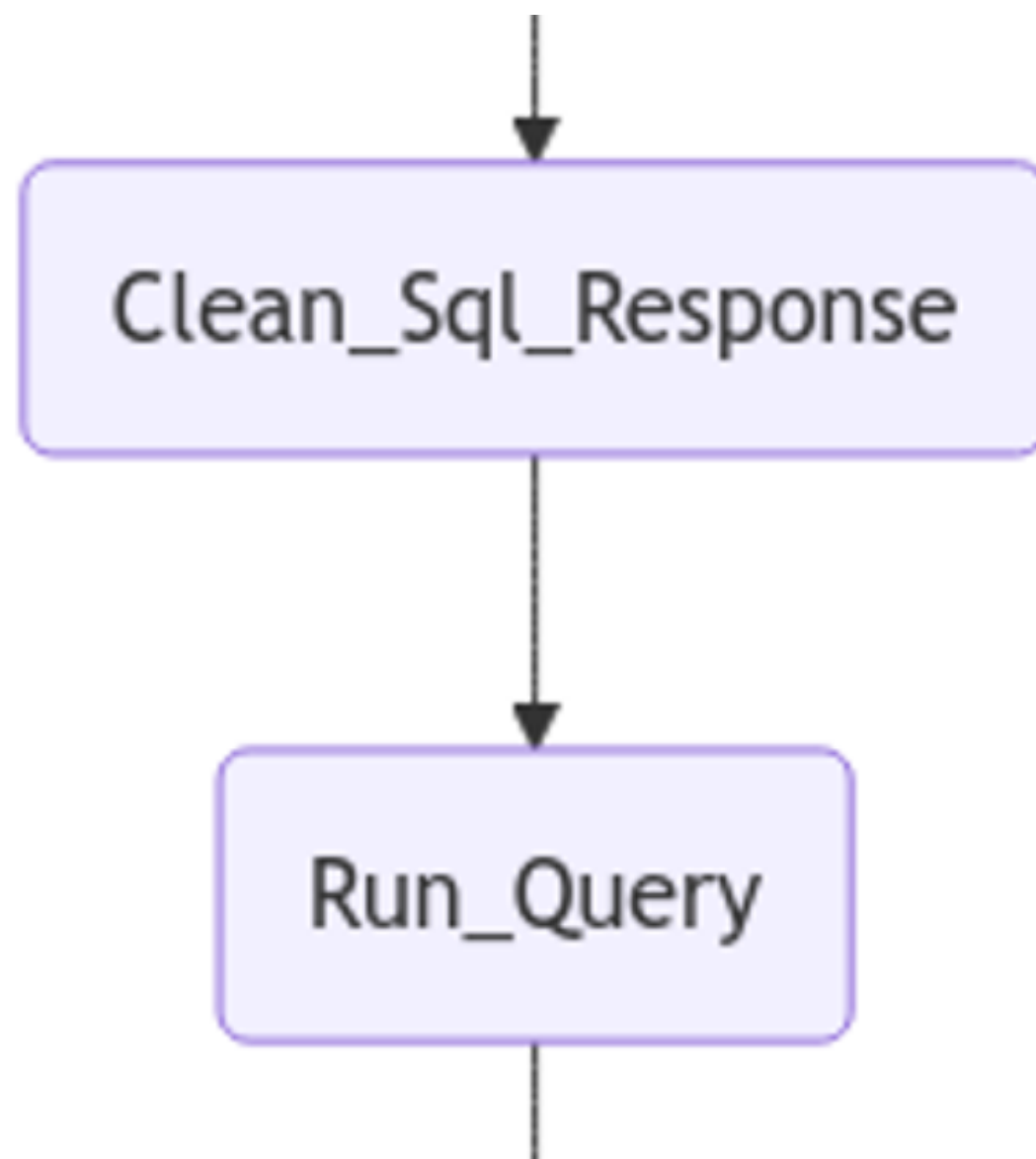
바뀐 Generate Query Node

겹치는 프롬프트는 base\_prompt로  
 만들고 다른 부분을 종류별로 만들어  
 if문을 사용해서 붙여넣기



## 모델 설계

### 4. Clean Query & Run Query



불필요하게 나온 문자열 부분을 다듬어 줌

잘 다듬은 sql쿼리문을 실행

# 모델 설계



## 5. Generate Response with OpenAI

```
system_prompt = f"""
당신은 부동산 추천 전문가이자 세계 지식을 갖춘 AI입니다.
주어진 정보와 세계 지식을 결합하여 아래 양식에 맞춰서 사용자의 질문에 답변해주세요.
구분선 이후 간단한 추천이유도 적어줍니다.
조회한 결과 매물 데이터 전부를 양식에 맞춰서 나열해줍니다.

정보: {state['results']}

## 출력 양식:
**매물번호**: {{property_id}}

**특징**:
{{특징}}

____

**추천 이유**:
{{추천 이유}}
"""
```

```
**매물번호**: 6949640

**특징**:
◆ 전세 금액: 10억
◆ 2025년 1월 입주 가능
◆ 50평, 방 5개 욕실 2개
◆ 대형 거실과 넓은 침실
◆ 구조 좋은 남향
◆ 관리비 약 40만원
◆ 신논현역, 언주역 도보 10분 거리
◆ 쾌적하고 조용한 1동 아파트

____

**추천 이유**: 이 매물은 강남역에서 도보로 접근 가능한 신논현역과 언주역 가까이에 위치하며, 넓은 평수와 방 수를 제공하여
```

프롬프트로 양식을 지정해줘서 쿼리로 나온 결과값을 양식에 맞게 바꿔줌

## 도전과제

---

### Frontend

- UX/UI 최적화

### Backend

- 파이프라인 설립 및 CI/CD

### RAG

- 질문 - 쿼리, 유저정보 Vector DB화

### PROMPT

- 도보 거리 계산 등 구체적인 조건 프롬프팅

# KPIs

측정방법

- 사용자 성장/참여도: DB에 남는 로그를 기반으로 측정
- 사용자 만족도/품질: 홈페이지 피드백을 이용해서 결과 종합
- 비즈니스 성과: 광고 DB에 남는 로그를 기반으로 측정

	KPI 지표	목표치	측정 주기
사용자 성장/참여도	재방문율	출시 후 3개월 내 30% 이상	월간
사용자 만족도/품질	사용자 만족도 (5점 척도)	출시 후 3개월 내 평균 3.5점 이상	분기별
사용자 만족도/품질	챗봇 응답 정확도	75% 이상	분기별
비즈니스 성과	광고 클릭율	3% 이상	월간
GOAL	출시 후 1년 이내 월간 사용자 수 50,000명 이상		

# 비즈니스 모델

광고 및 스폰서십 모델	데이터 기반 인사이트 판매
<div>플랫폼 내에 부동산 광고를 게재, 스폰서십 수익 창출</div> <div><ul style="list-style-type: none"><li>부동산 개발업체, 인테리어 업체, 금융기관 등과 광고 계약 체결.</li><li>타겟 광고를 통해 유저에게 관련성 높은 광고 노출.</li></ul></div>	<div>유저 데이터 분석 후 부동산 시장 트렌드, 유저 선호도 등을 부동산, 연구기관에 판매</div> <div><ul style="list-style-type: none"><li>익명화된 데이터를 기반으로 시장 보고서 작성.</li><li>부동산 관련 기업이나 연구기관에 데이터 판매.</li></ul></div>
파트너십 및 제휴	
<div>금융기관, 인테리어 업체, 이사 서비스 업체 등과 제휴 유저에게 통합 서비스를 제공 후 수수료 받기</div> <div><ul style="list-style-type: none"><li>모기지 대출, 보험, 인테리어 등 관련 서비스와의 제휴 계약.</li><li>서비스를 이용하는 유저에게 할인 혜택 제공 및 제휴 수익 확보.</li></ul></div>	

# 프로젝트 시연

**감사합니다**