



개인 맞춤형 주거 지역 추천 서비스



집탐험대

목차

01. 팀 소개

02. 프로젝트 소개

01. 개발 배경

02. 사이트 구조도

03. 차별점

03. 프로젝트 개발 과정

01. 설계도 및 개발 도구

02. 분석 프로세스

03. 주요 기능

04. 프로젝트 결과

01. 시연 영상

02. 기대효과 및 활용분야

팀 소개

김시은

데이터 수집 및 전처리
데이터 시각화

김지호

웹 디자인
프론트엔드

김효정

백엔드

이현지

데이터 수집 및 전처리
알고리즘

01. 기존 부동산 애플리케이션의 한계점 보완

기존의 부동산 애플리케이션은 집의 위치, 가격, 평수에 치중한 정보만을 제공한다. 하지만 집을 구할 때 우리는 집의 위치, 가격, 평수 외에도 교통, 안전, 직장과의 거리, 편의시설 등 많은 정보를 필요로 한다. 이러한 정보를 제공해주는 서비스의 부재로 대부분의 사람들은 본인의 라이프 스타일에 맞는 주거지를 찾기 위해서 많은 시간과 비용을 투자해야한다.

이 프로젝트를 통해 사용자의 나이, 성별, 라이프 스타일에 적합한 지역을 추천하여 집을 구하는 사용자의 만족도를 향상시키고자 한다.

The image shows a complex filter interface for a real estate application. At the top, there are four filter categories: '관리비' (Management Fee), '방크기' (Room Count), '사용승인일' (Usage Approval Date), and '추가필터' (Additional Filter). Below these, there are buttons for '매물' (Listings) and '분양' (Sales), followed by a dropdown menu for '원룸, 투·쓰리룸, 오피스텔·도시형' (Studio, 2-3 room, Office/Urban) and another dropdown for '월세, 전세, 매매' (Monthly Rent, Full Rent, Buy). A search bar with the placeholder '지역, 지하철역, 학교 검색' (Search by area, subway station, school) is located below the filters. To the right of the search bar, there are three tabs: '전월세 · 금액' (Monthly Rent · Amount), '구조 · 면적' (Structure · Area), and '옵션' (Options). The '전월세 · 금액' tab is currently selected. The '추가필터' section is expanded, showing various filters: '층수' (Floor Count) with options like '전체' (All), '1층', '2층', '3층', '4층', '5층', '6층', '7층 이상', '반지층', and '옥탑방'; '방구조' (Room Structure) with options like '분리형' (Separate), '복층' (Duplex), '투룸' (2-room), and '쓰리룸 이상' (3-room or more); '세대수' (Number of Units) with options like '전체' (All), '100세대 이하' (Below 100 units), '100세대 이상' (Above 100 units), '500세대 이상' (Above 500 units), and '1000세대 이상' (Above 1000 units); '주차대수' (Number of Parking Spaces) with options like '상관없음' (Doesn't matter), '세대당 2대 이상' (More than 2 units per floor), and '세대당 1대 이상' (More than 1 unit per floor); '거래종류' (Transaction Type) with options like '전체' (All), '중개' (Brokerage), and '직거래' (Direct transaction); and '추가 옵션' (Additional Options) with options like '주차가능' (Parking available), '단기임대' (Short-term rental), '풀옵션' (Full option), '엘리베이터' (Elevator), '배란다' (Balcony), '보안/안전시설' (Security/Safety facilities), '360° VR', and '비대면계약' (Non-face-to-face contract). A note at the bottom states: '풀옵션 항목은 세탁기, 냉장고, 에어컨, 가스레인지 또는 인덕션입니다.' (Full option items include washing machine, refrigerator, air conditioner, gas stove, or induction cooktop).

사진1. 기존 부동산 애플리케이션이 제공하는 정보의 한계

02. 주거 환경과 주거지에 대한 중요성 부각

포스트 코로나 시대의 영향으로 '집'의 의미가 변화했다. 팬데믹 이전까지만 해도 집은 안락한 휴식처 혹은 잠깐 머물다가는 숙소에 가까웠다. 하지만 더 이상 집은 거쳐 가는 곳이 아니다. 이제 집은 사회·문화·경제 등과 관련된 일련의 모든 생활 활동이 가능한 복합생활공간과 다를 바 없다.¹⁾ 특히 사람들은 집의 내부만큼이나 인근 동네의 기능에도 큰 관심을 갖기 시작했다. 신조어 '슬세권'²⁾ 등장은 주변 환경의 중요성을 강조한다. 이 프로젝트를 통해 데이터 시각화를 기반으로 주거지역 정보를 제공하여 사람들이 보다 편리하게 만족스러운 주거지역을 찾을 수 있도록 한다.

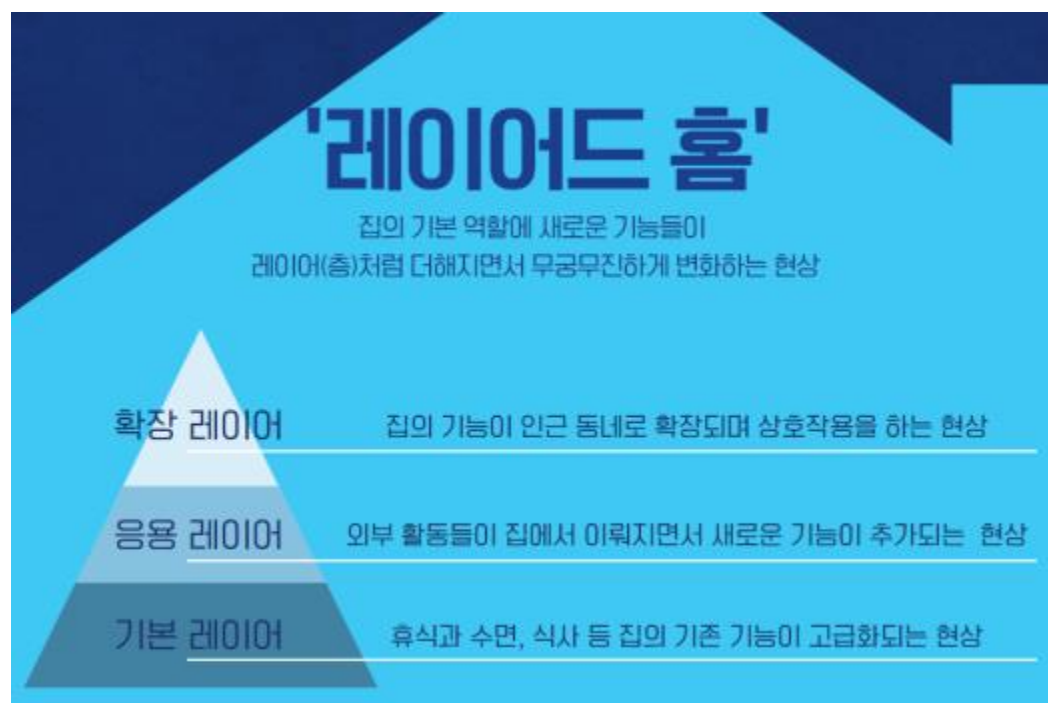


사진2. 코로나 이후 주거 환경 트렌드 '레이어드 홈'

1) 최아름. (2021). '집방'을 통해 본 '집'의 의미와 역할 변화에 대한 고찰. 문화콘텐츠연구,(21), 193-226.
2) 슬리퍼 차림과 같은 편한 복장으로 카페나 편의점, 도서관, 쇼핑몰 같은 편의시설을 사용할 수 있는 주거 권역

사이트 구조도

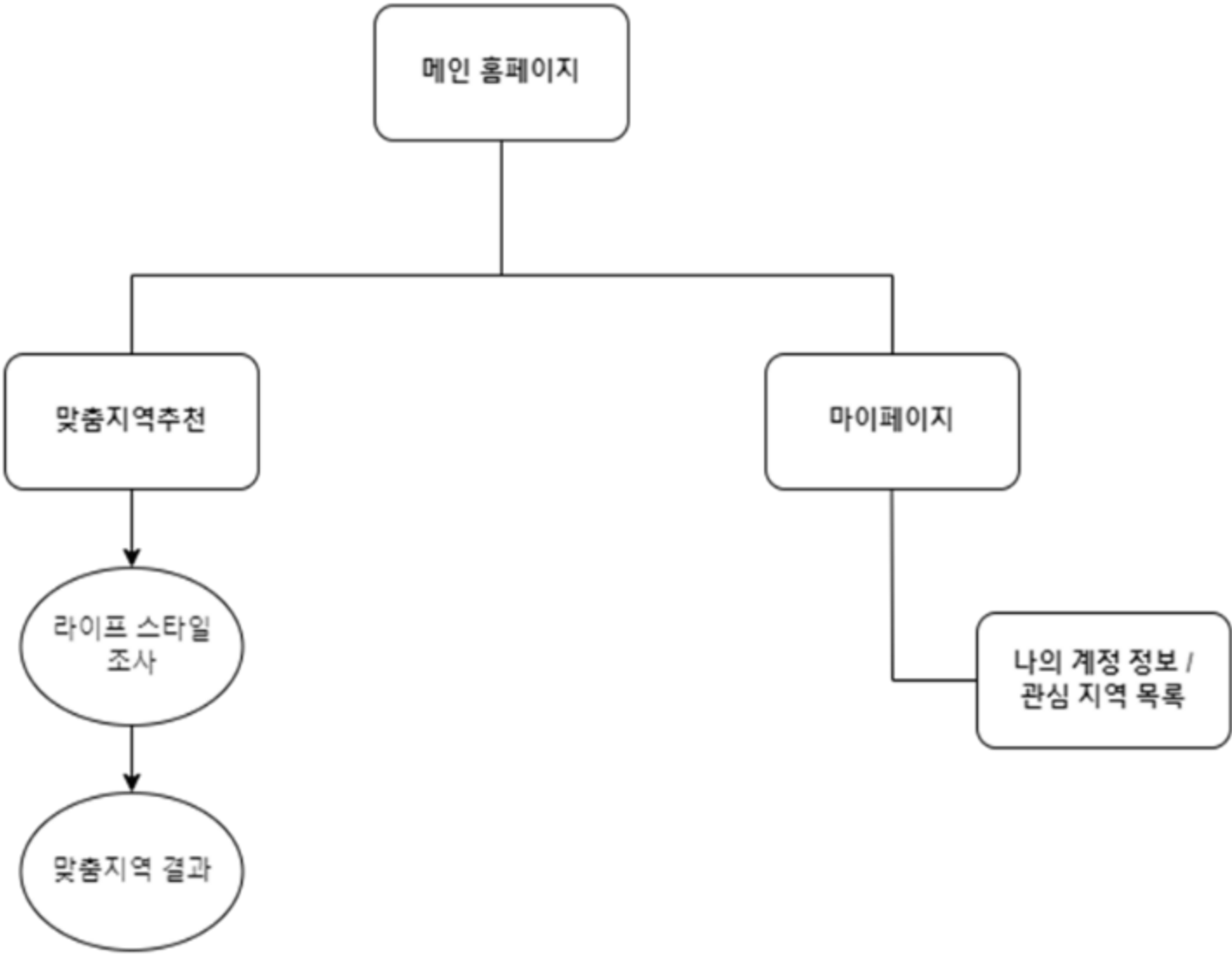
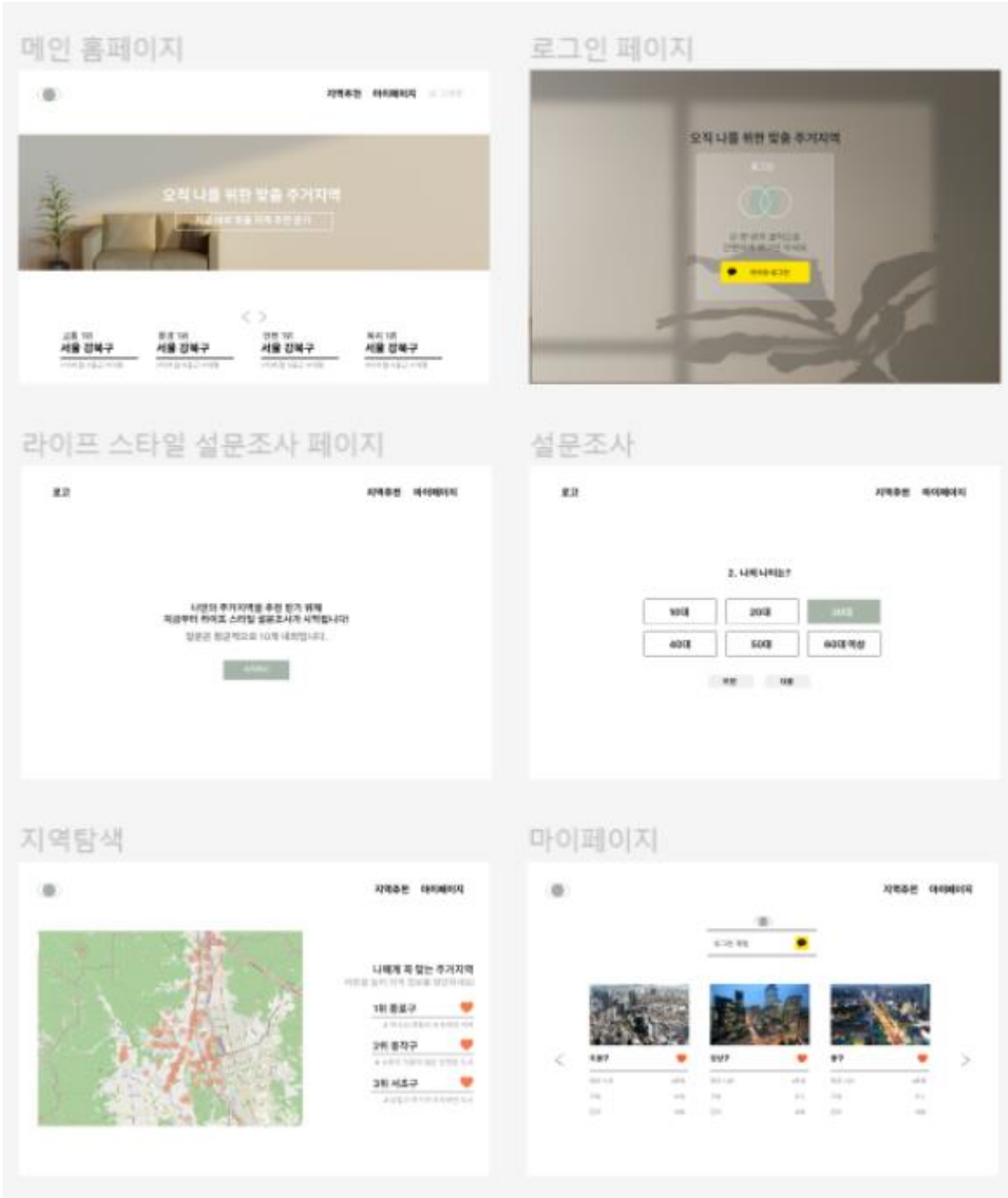


사진3. 사이트 디자인 시안 및 구조도

차별점

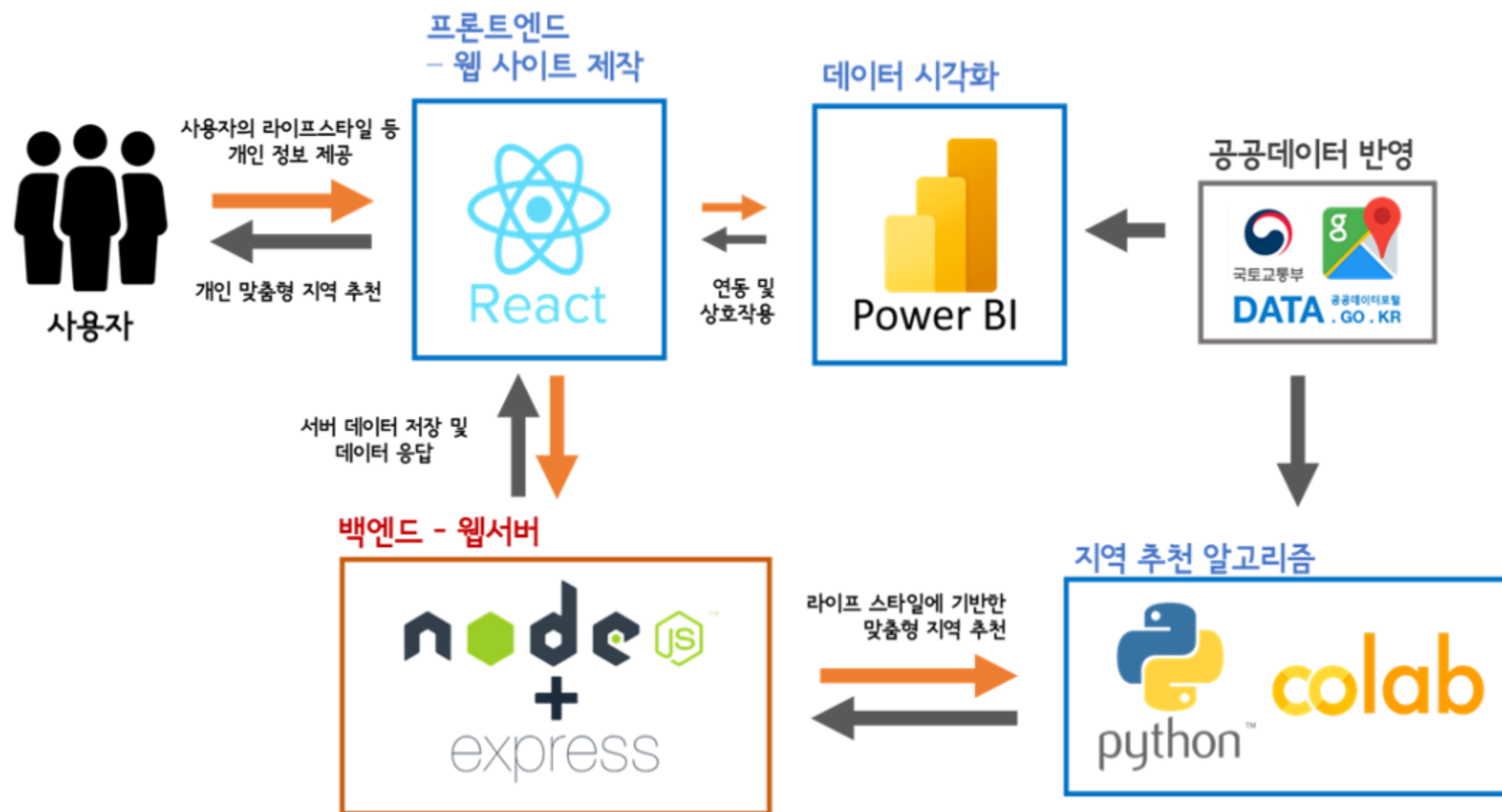
「'나'의 라이프 스타일에 맞는 주거지역 추천

나이, 성별, 부양가족, 취미 등 오로지 '나'의 정보를 기반으로 주거지역을 추천해주는 서비스

「주거지역의 다양한 정보를 한 눈에 비교

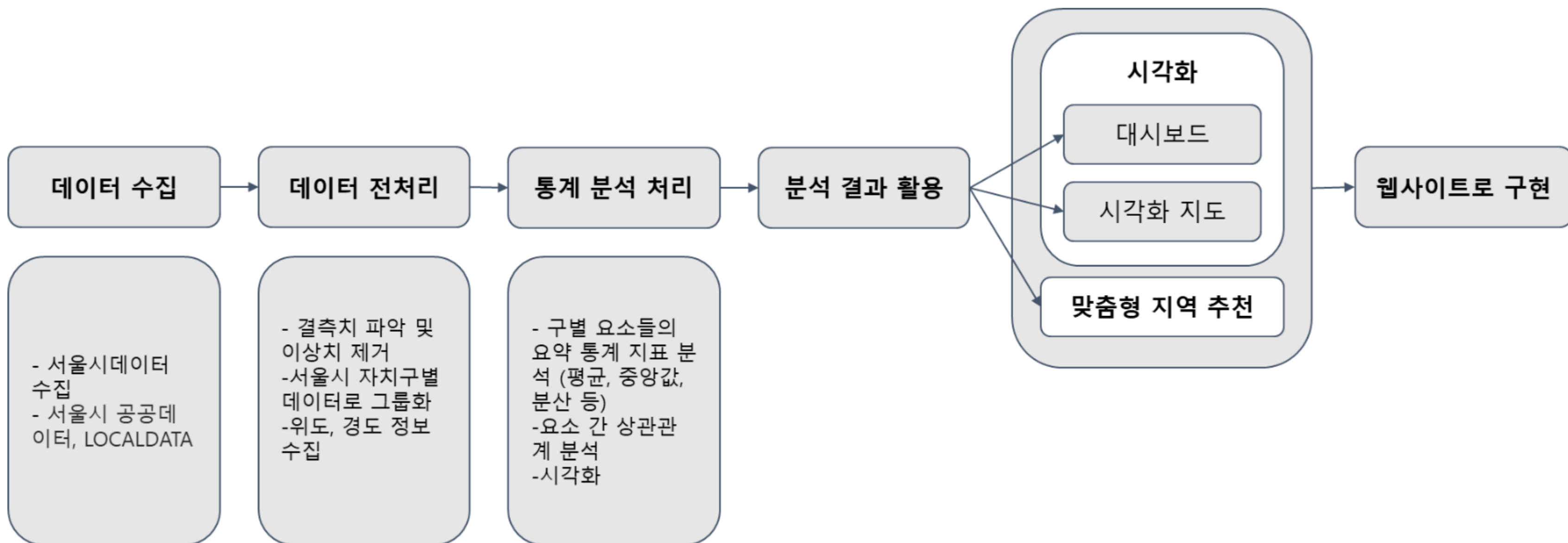
별도로 검색할 필요없이 지역의 모든 정보를 한 눈에 확인 가능

설계도 및 개발도구



| 기능 | 개발도구 |
|-----------|--|
| 데이터 시각화 | - PowerBI |
| 맞춤형 지역 추천 | - Python package (Pandas, Numpy, Scikit-learn) - Colab |
| 프론트엔드 | - React |
| 백엔드 | - Node js |
| 데이터 | - 수집 : 공공데이터 - 전처리 : Pandas |

분석 프로세스



주요 기능

| 구분 | | 기능 |
|-------------|---------|---|
| 데이터 시각화 | 지역 대시보드 | <ul style="list-style-type: none"> - 주거지역의 주변 환경, 복지, 교통 등의 지표를 사용자가 한 눈에 보기 쉽게 대시보드로 시각화 - 특정 지역의 모든 지표를 한 눈에 확인 가능 (구 단위 정보 제공) |
| | 시각화 지도 | <ul style="list-style-type: none"> - 지도 상에 데이터를 시각화 - 특정 요소에 대해 전체 지역의 정보를 한 눈에 볼 수 있음. |
| 맞춤형 지역 추천 | | <ul style="list-style-type: none"> - 추천 알고리즘을 통해 사용자가 입력한 데이터와 매칭되는 지역을 추천 - 사용자는 연령, 성별, 취미와 중요하게 고려하는 요소(대중교통, 복지·의료시설 등)를 입력 |
| 로그인 / 마이페이지 | | <ul style="list-style-type: none"> - 소셜 계정 연동을 통한 편리한 로그인 - 사용자의 라이프 스타일 정보 저장 - 관심 지역 저장 및 지역 간 특징 비교 |

주요 기능

콘텐츠 기반 모델 (CB)

모델 특징 : 사용자 정보, 아이템 정보를 활용하는 모델

모델 강점 : 항목의 특성을 기반으로 분석

- 콜드스타트, 특이 취향 유저에 대한 처리 가능
- 가장 원시적인 모델로 직접적으로 특성에 대한 가중치 조절 가능

사용자 기반 협력 필터링 모델 (CF, Collaborative Filtering) [6]

유사도 행렬을 기반으로 사용자와 유사한 선호도를 가진 다른 사용자가 구매하거나 선호하는 상품을 추천하는 기법



그림1. CF 모델 예시 [5]

[5] 정영윤 외. "개인 맞춤형 음식 추천 알고리즘에 대한 비교 평가 연구." 한국정보통신학회논문지, vol. 27, no. 3, 2023, 318-327.

[6] 손지은 외. "추천 시스템 기법 연구동향 분석." 대한산업공학회지, vol. 41, no. 2, 2015, 185-208. 9

주요 기능

콘텐츠 기반 모델 (CB) 예시

User matrix 가중치 정보

| | 경찰서 | 마트 | 미술관 | 공연장 | 테니스 | 축구 | 초등학교 | 공원 | 미세먼지 | 병원 | 지하철 | 버스 | 노인복지시설 |
|---|-----|----|-----|-----|-----|----|------|----|------|----|-----|----|--------|
| A | 10 | 6 | 1 | 1 | 6 | 1 | 4 | 10 | 3 | 7 | 11 | 11 | 2 |
| B | 7 | 7 | 1 | 1 | 1 | 6 | 3 | 8 | 2 | 8 | 12 | 7 | 4 |
| C | 6 | 7 | 1 | 6 | 1 | 1 | 8 | 7 | 4 | 8 | 6 | 11 | 8 |
| D | 8 | 5 | 6 | 1 | 1 | 1 | 6 | 12 | 8 | 6 | 3 | 3 | 2 |



Item matrix 자치구별 특성 정보

| | 강남구 | 강동구 | 강북구 | 강서구 | 관악구 | 광진구 | 구로구 | 금천구 | 노원구 | 도봉구 |
|--------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| 경찰서 | 0.166040 | 0.504123 | 0.518537 | 0.922622 | 0.539030 | 0.305636 | 0.009182 | 0.888631 | 0.945022 | 0.517929 |
| 마트 | 0.413616 | 0.170858 | 0.122650 | 0.121031 | 0.743667 | 0.987296 | 0.528436 | 0.035151 | 0.381106 | 0.305310 |
| 미술관 | 0.000757 | 0.124618 | 0.925627 | 0.862807 | 0.695232 | 0.804820 | 0.070049 | 0.869554 | 0.753210 | 0.968609 |
| 공연장 | 0.655050 | 0.155790 | 0.482464 | 0.450374 | 0.052364 | 0.996938 | 0.015063 | 0.591317 | 0.172262 | 0.010102 |
| 테니스 | 0.520953 | 0.677668 | 0.193265 | 0.407466 | 0.317512 | 0.682270 | 0.443371 | 0.898093 | 0.205366 | 0.945218 |
| 축구 | 0.201792 | 0.790534 | 0.876274 | 0.529418 | 0.446671 | 0.372449 | 0.887418 | 0.755597 | 0.496506 | 0.592141 |
| 초등학교 | 0.317354 | 0.584853 | 0.697348 | 0.199177 | 0.507345 | 0.040719 | 0.212259 | 0.844177 | 0.530031 | 0.005382 |
| 공원 | 0.645463 | 0.421798 | 0.478593 | 0.336234 | 0.758430 | 0.146772 | 0.963641 | 0.516519 | 0.324349 | 0.641324 |
| 미세먼지 | 0.733647 | 0.394459 | 0.560116 | 0.164915 | 0.995805 | 0.865096 | 0.378240 | 0.802366 | 0.511513 | 0.386464 |
| 병원 | 0.824380 | 0.286702 | 0.503665 | 0.128179 | 0.045627 | 0.955536 | 0.427687 | 0.797893 | 0.026753 | 0.896670 |
| 지하철 | 0.797121 | 0.847622 | 0.362927 | 0.405662 | 0.504514 | 0.508467 | 0.883019 | 0.339916 | 0.070819 | 0.503527 |
| 따릉이 | 0.075610 | 0.193837 | 0.303140 | 0.924731 | 0.646911 | 0.058646 | 0.004158 | 0.769158 | 0.450794 | 0.799366 |
| 노인복지시설 | 0.829940 | 0.458442 | 0.998480 | 0.250926 | 0.310822 | 0.204735 | 0.947169 | 0.566732 | 0.276322 | 0.054862 |



| | 강남구 | 강동구 | 강북구 | 강서구 | 관악구 | 광진구 | 구로구 | 금천구 | 노원구 | 도봉구 |
|---|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| A | 31.525291 | 40.147040 | 46.375280 | 25.018446 | 26.245367 | 47.813658 | 34.462801 | 44.625601 | 46.227770 | 42.904204 |
| B | 30.535635 | 35.860424 | 39.424981 | 24.476356 | 24.415362 | 40.641647 | 30.857115 | 40.718282 | 41.600804 | 40.894238 |
| C | 34.218555 | 37.673022 | 44.729631 | 24.398890 | 28.858180 | 46.085683 | 35.468856 | 41.550753 | 48.605106 | 46.087203 |
| D | 22.557123 | 27.982963 | 34.262560 | 23.662304 | 24.757876 | 39.852693 | 32.774256 | 35.071538 | 41.685719 | 37.477234 |

User-item martix

```
# 사용자 c의 점수가 가장 높은 구 3개 찾기
user_C_scores = user_item_matrix3.loc['C']
top_3_gu = user_C_scores.nlargest(3).index

print("사용자 c에 대한 가장 높은 점수를 가지는 구 3개:")
print(top_3_gu)
```

사용자 c에 대한 가장 높은 점수를 가지는 구 3개:
Index(['노원구', '도봉구', '광진구'], dtype='object')

시연 영상



기대효과 및 활용분야

■ 사용자 측면

- 만족도와 삶의 질 향상
- 많은 시간과 노력을 들이지 않고 최적화된 주거지를 찾을 수 있음

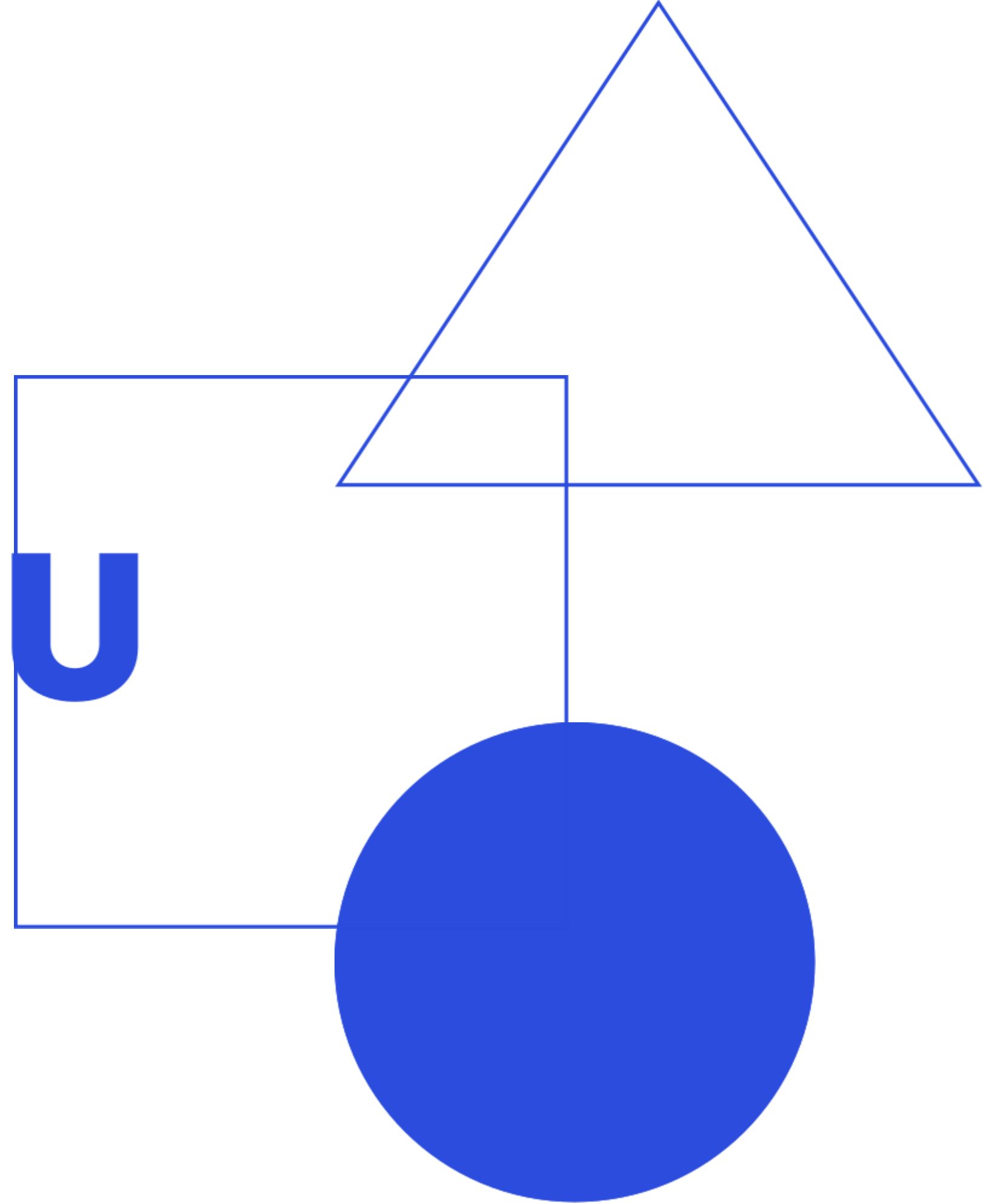
■ 기술적 측면

- 데이터 시각화를 통해 복잡한 데이터를 쉽게 표현
- 데이터 분석을 통해 최적의 주거지를 제안

■ 사회적 측면

- 지속 가능한 도시 발전
- 지역 경제에 긍정적 영향
- 정부, 지자체의 투자 가능성

THANK YOU



< 부록 >

발표 내용 외 추가 기능 설명

설문조사 페이지

사용자가 상반되는 답변 클릭 시 다른 답변 사전 차단

Q.05
나의 자녀는?
(복수선택 가능)

없음

초등학교

중학교

고등학교

특수학교

이전

다음

“없음”

“있음”

초등학교 / 중학교 /
고등학교 / 특수학교

```
<label className='checkboxStyle'>
  <input type="checkbox"
    name="child"
    value="없음"
    style={{ display: "none" }}
    onChange={handleCheckboxChange}
    checked={selectedOptions.includes('없음')}
    disabled={selectedOptions.length > 0 && !selectedOptions.includes('없음')}
    className='test' />
  <span>없음</span>
</label>
```

```
<label className='checkboxStyle'>
  <input type="checkbox"
    name="child"
    value="초등학교"
    style={{ display: "none" }}
    onChange={handleCheckboxChange}
    checked={selectedOptions.includes('초등학교')}
    disabled={selectedOptions.includes('없음')}
    className='test' />
  <span>초등학교</span>
</label>
```

사용자 답변 중 특정 답변이 포함되면 disabled 상태로 전환

설문조사 페이지

답변의 양이 많은 경우 버튼으로 구현하여 사용자의 편리성 향상
화면 상에 사용자의 답변이 바로 보이도록 구성.

Q.07
내가 중요하게 생각하는 요소를 1순위부터 7순위까지 차례대로 눌러주세요.
(모든 요소 선택 이후 다음 페이지로 이동 가능합니다.)

처음부터 다시하기 ↺

| | | | | | | |
|----|------|----|-----------|-----------|-----------|------------------|
| 안전 | 생활시설 | 교육 | 의료 | 환경 | 교통 | 기타 (금액, 평수 등) |
| | | | 1순위 의료 | 2순위 환경 | 3순위 교통 | 4순위 기타 |

이전 다음

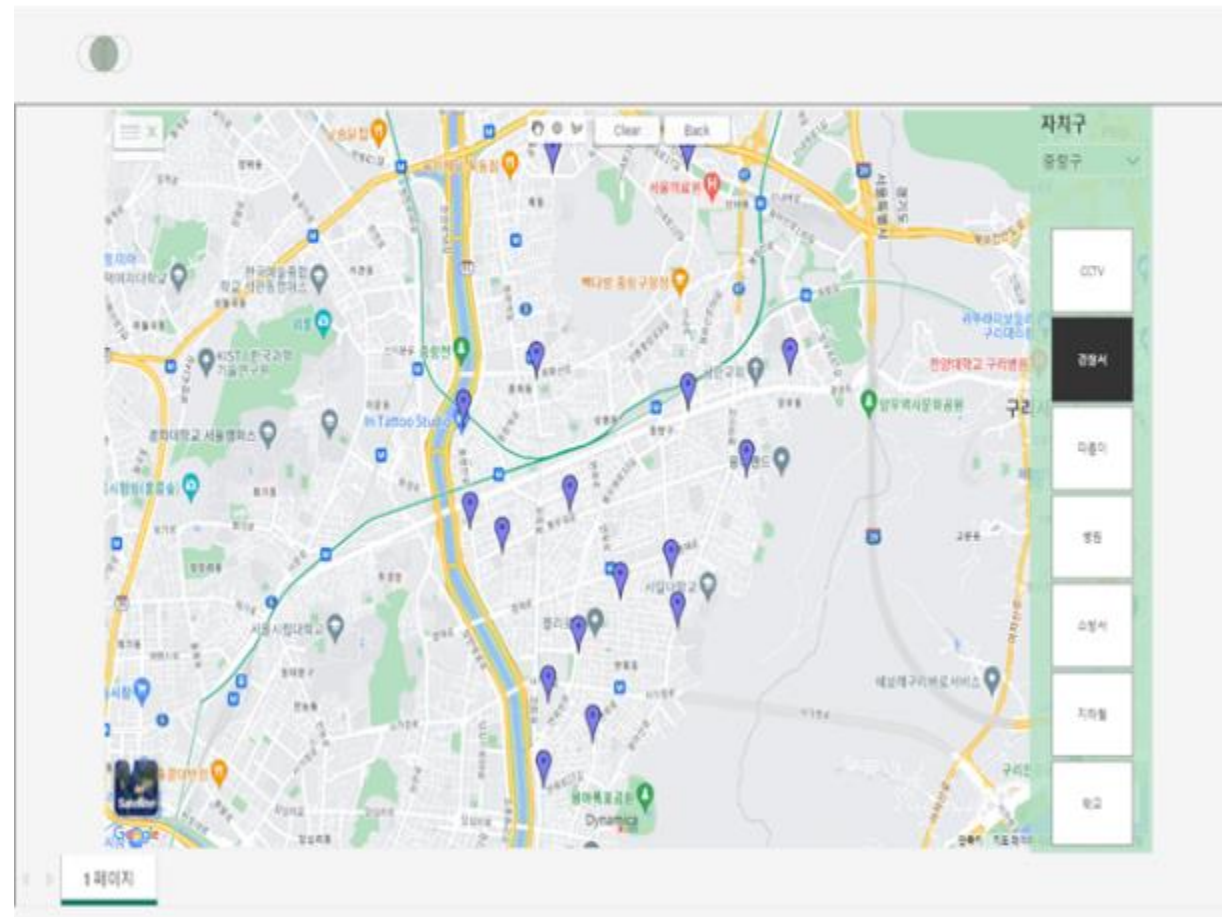
```
<div className='page-priority-rank'>
  {buttonClicks.map((click, index) => (
    <div key={index} className='page-priority-set'>
      <div className='page-priority-value'>{Object.values(click)[0]}순위</div> <div className='page-rank-key'>{Object.keys(click)[0]}</div>
    </div>
  ))}
</div>
```

사용자가 누르는 순서대로 화면에 나열

지역 추천 결과 페이지

맞춤형 주거지역 3곳 추천

PowerBI 프로그램 연동 - 시각화지도 보기, 대시보드 보기



시각화 지도 보기

맞춤지역추천 마이페이지 로그아웃

나에게 딱 맞는 주거지역 순위
버튼을 눌러 지역에 대한 다양한 정보를 탐색해보세요!
지도보기

- 1위 도봉구 #편안한 주거환경과 공원이 많은 지역
- 2위 서대문구 #역사적인 명소와 문화 시설이 풍부한 지역
- 3위 노원구 #주택가와 자연이 조화로운 도시



대시보드 보기

지역 추천 결과 페이지

나에게 딱 맞는 주거지역 순위

버튼을 눌러 지역에 대한 다양한 정보를 탐색해보세요!

 지도 보기

1위 **도봉구**

#편안한 주거환경과 공원이 많은 지역

2위 **서대문구**

#역사적인 명소와 문화 시설이 풍부한 지역

3위 **노원구**

#주택가와 자연이 조화로운 도시





대시보드 보기 / 관심목록 저장하기



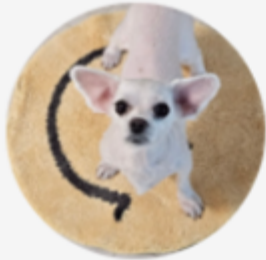
해당 지역에 대한 간략한 특징 소개



라이프 스타일 설문조사 처음부터 다시하기

마이페이지

한 눈에 여러 지역 특징 비교하기



김지호님의
딱 맞는 관심목록 리스트

관심지역이 없습니다.
지금 바로 오직 나를 위한 지역을 추천받아요!

지금 바로 추천받기



김지호님의
딱 맞는 관심목록 리스트

| 노원구 | ❤️ | 강북구 | ❤️ | 종로구 | ❤️ |
|-------------------|--------|----------------|--------|------------------|--------|
| #주택가와 자연이 조화로운 도시 | | #자연경관이 아름다운 도시 | | #역사와 전통이 어우러진 지역 | |
| 평균시세 | n000만원 | 평균시세 | n000만원 | 평균시세 | n000만원 |
| 복지 | n위 | 복지 | n위 | 복지 | n위 |
| 교통 | n위 | 교통 | n위 | 교통 | n위 |

추천 알고리즘

- 항목별 함수 생성: 사용자 설문 결과가 저장된 파일을 불러와서 가중치를 부여하는 것 항목별 함수 생성
- 효율적인 처리 방식: dataframe을 dictionary 형태로 전환하여 key, value를 통해 코드 단순화

```
def update_ten(user,matrix):
    if user == '자연':
        matrix['공원'] += 5
    elif user == '핫플':
        matrix['쇼핑'] += 5

# key와 함수 매핑
key_to_func = {
    'sex': update_gender,
    'age': update_age,
    'family': update_family,
    'marry': update_marry,
    'child': update_child,
    'hobby': update_hobby,
    'sports': update_sports,
    # 'car':
    # 'env':
    'wel': update_wel,
    'ten': update_ten
}

##### 사용자 정보에 따라 matrix 업데이트
for key, update_func in key_to_func.items():
    for user, info in zip(users, user_matrix.index):
        update_func(user[key], user_matrix.loc[info])
```

| name | age | child | hobby | sex | sports | ten | wel | marry | family | location1 | location2 | location3 |
|------|-----|----------|----------|-----|--------|-------|----------|-------|--------|-----------|-----------|-----------|
| 김효정 | 10대 | ["없음"] | ["운동"] | 남성 | ["축구"] | 핫플 도시 | ["필요없음"] | | | | | |
| 김효정 | 20대 | ["고등학교"] | ["쇼핑","기 | 여성 | | 핫플 | ["필요없음"] | 기혼 | 5인 | 강남구 | 관악구 | 서초구 |

| name | 안전 | 생활시설 | 교육 | 의료 | 환경 | 교통 | 기타 |
|------|----|------|----|----|----|----|----|
| 김효정 | 7 | 1 | 2 | 3 | 4 | 5 | 6 |

| name | 광역버스 | 기차 | 따릉이 | 시내버스 | 자차 | 지하철 |
|------|------|----|-----|------|----|-----|
| 김효정 | 0 | 0 | 0 | 5 | 0 | 0 |

| name | 공원 | 미세먼지 | 소음 | 풍수해 |
|------|----|------|----|-----|
| 김효정 | 0 | 0 | 0 | 5 |

중요하게 생각하는
요소에 가중치를 두어
저장

```
]
# 마지막 유저만 추출
last_user = survey_result.iloc[-1:]
last_user
```

| | name | age | child | hobby | sex | sports | ten | wel | marry | family | location1 | location2 | location3 |
|----|------|-----|----------|--------|-----|--------|-----|-----|-------|--------|-----------|-----------|-----------|
| 11 | 김효정 | 50대 | ["고등학교"] | ["쇼핑"] | 여성 | NaN | NaN | NaN | 기혼 | 2인 | 강남구 | 관악구 | 서초구 |

```
] users = last_user.to_dict(orient='records')
users
```

```
[{'name': '김효정',
 'age': '50대',
 'child': '["고등학교"]',
 'hobby': '["쇼핑"]',
 'sex': '여성',
 'sports': nan,
 'ten': nan,
 'wel': nan,
 'marry': '기혼',
 'family': '2인',
 'location1': '강남구',
 'location2': '관악구',
 'location3': '서초구'}]
```

추천 알고리즘

- 항목별 함수 생성: 사용자 설문 결과가 저장된 파일을 불러와서 가중치를 부여하는 것 항목별 함수 생성
- 효율적인 처리 방식: dataframe을 dictionary 형태로 전환하여 key, value를 통해 코드 단순화

```
def update_ten(user,matrix):
    if user == '자연':
        matrix['공원'] += 5
    elif user == '핫플':
        matrix['쇼핑'] += 5

# key와 함수 매핑
key_to_func = {
    'sex': update_gender,
    'age': update_age,
    'family': update_family,
    'marry': update_marry,
    'child': update_child,
    'hobby': update_hobby,
    'sports': update_sports,
    # 'car':
    # 'env':
    'wel': update_wel,
    'ten': update_ten
}

##### 사용자 정보에 따라 matrix 업데이트
for key, update_func in key_to_func.items():
    for user, info in zip(users, user_matrix.index):
        update_func(user[key], user_matrix.loc[info])
```

| name | age | child | hobby | sex | sports | ten | wel | marry | family | location1 | location2 | location3 |
|------|-----|----------|----------|-----|--------|-------|----------|-------|--------|-----------|-----------|-----------|
| 김효정 | 10대 | ["없음"] | ["운동"] | 남성 | ["축구"] | 핫플 도시 | ["필요없음"] | | | | | |
| 김효정 | 20대 | ["고등학교"] | ["쇼핑","기 | 여성 | | 핫플 | ["필요없음"] | 기혼 | 5인 | 강남구 | 관악구 | 서초구 |

| name | 안전 | 생활시설 | 교육 | 의료 | 환경 | 교통 | 기타 |
|------|----|------|----|----|----|----|----|
| 김효정 | 7 | 1 | 2 | 3 | 4 | 5 | 6 |

| name | 광역버스 | 기차 | 따릉이 | 시내버스 | 자차 | 지하철 |
|------|------|----|-----|------|----|-----|
| 김효정 | 0 | 0 | 0 | 5 | 0 | 0 |

| name | 공원 | 미세먼지 | 소음 | 풍수해 |
|------|----|------|----|-----|
| 김효정 | 0 | 0 | 0 | 5 |

중요하게 생각하는
요소에 가중치를 두어
저장

```
]
# 마지막 유저만 추출
last_user = survey_result.iloc[-1:]
last_user
```

| | name | age | child | hobby | sex | sports | ten | wel | marry | family | location1 | location2 | location3 |
|----|------|-----|----------|--------|-----|--------|-----|-----|-------|--------|-----------|-----------|-----------|
| 11 | 김효정 | 50대 | ["고등학교"] | ["쇼핑"] | 여성 | NaN | NaN | NaN | 기혼 | 2인 | 강남구 | 관악구 | 서초구 |

```
] users = last_user.to_dict(orient='records')
users
```

```
[{'name': '김효정',
 'age': '50대',
 'child': '["고등학교"]',
 'hobby': '["쇼핑"]',
 'sex': '여성',
 'sports': nan,
 'ten': nan,
 'wel': nan,
 'marry': '기혼',
 'family': '2인',
 'location1': '강남구',
 'location2': '관악구',
 'location3': '서초구'}]
```


생성형 AI

- 실행 화면

