

소소행

: RAG기반 소도시 여행지
추천 통합 플랫폼



12팀 내가그린
강다혜, 김희서, 윤가빈

목차

1. Pain Point

2. Solution

3. 차별점

4. 전체 가능 및
시스템 아키텍쳐

5. 핵심 기술 구현

6. 기대효과

1. Pain Point



2. Solution

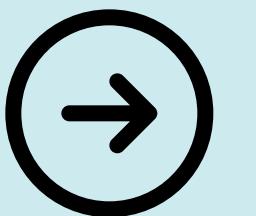
선순환 구조 설계

① AI 기반 초개인화 여행지 추천
(RAG & LLM)

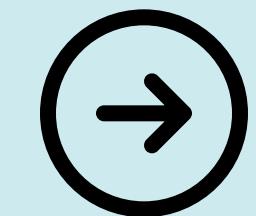
② 실시간 지역 축제 정보 제공
(LBS)

③ 지역 상생 로컬 커머스 통합
(Market)

검증된 한국관광공사 데이터를
기반으로 AI가 사용자의 의도를
해석하고 대화하듯 여행지를 추천

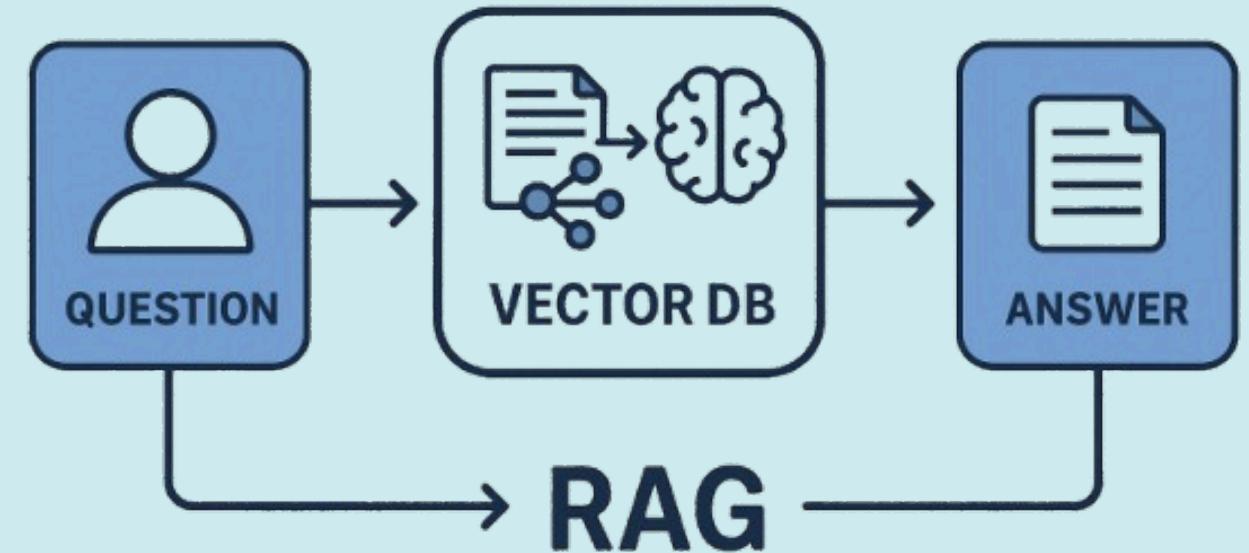


여행지 근처에서의 실시간 축제
정보를 제공받아 더욱 알찬
지역 경험의 기회를 제공



추천받은 지역의 특산물을 바로
구매할 수 있는 마켓 기능을 연동
하여 지역 경제 활성화까지 도모

3. 차별점



단순 나열 vs 근거 있는 추천 (Trustworthy AI)

기존 서비스는 필터링 된 정보를 단순 나열하지만, 소소행은 RAG 기술을 통해 "왜 이 장소를 추천하는지"에 대한 명확한 근거(최신 데이터, 후기 반영)를 함께 제시하여 신뢰도를 높였습니다.

파편화 vs 올인원 (All-in-One Experience)

지도 앱, 블로그, 쇼핑몰을 오갈 필요 없이 [여행지 추천 - 축제 정보 - 특산물 구매]가 하나의 앱에서 이루어지는 끊김 없는 (Seamless) 경험을 제공합니다.

일회성 방문 vs 지속 가능한 연결 (Economic Cycle)

단순히 여행지를 방문하고 끝나는 것이 아니라, 여행 후에도 지역 특산물을 재구매하거나 찜 기능을 통해 지역 사회와 지속적인 관계를 맺도록 유도합니다.

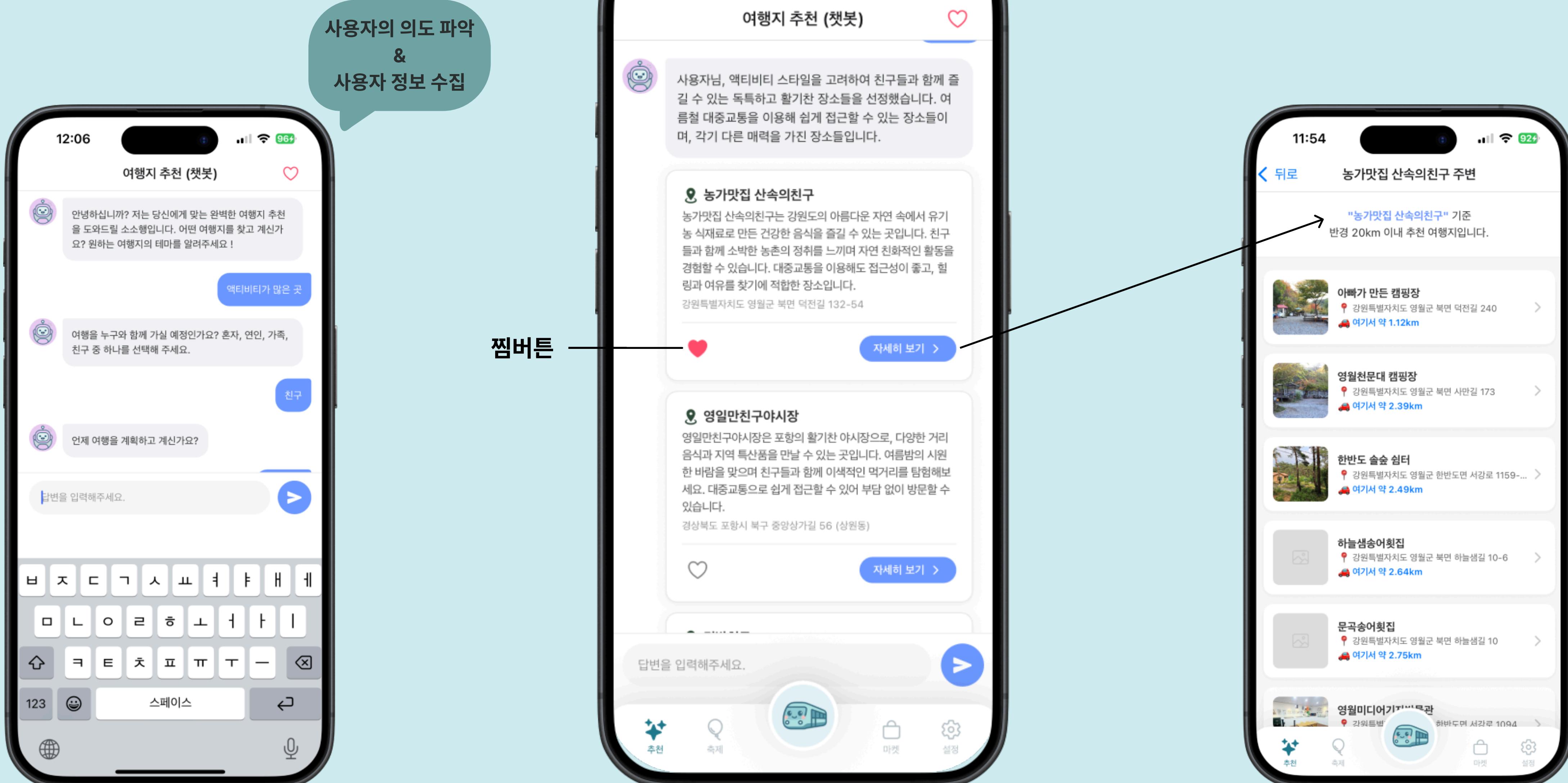
4. 전체 기능 및 시스템 아키텍처

서비스의 흐름이 어떻게 선순환을 만드는지 강조합니다.



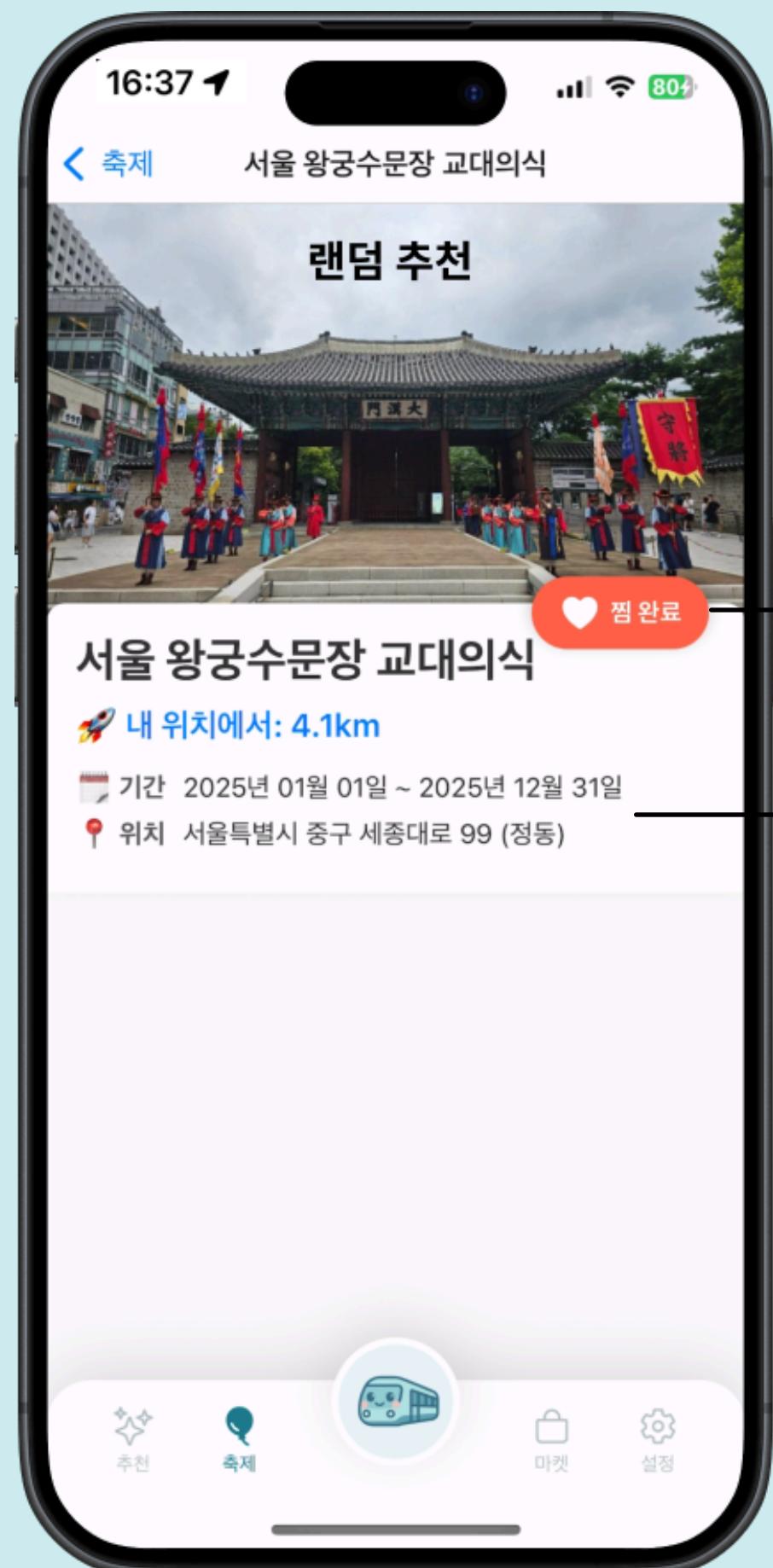
4-1. 전체 기능

기능1: RAG 기반 여행지 추천

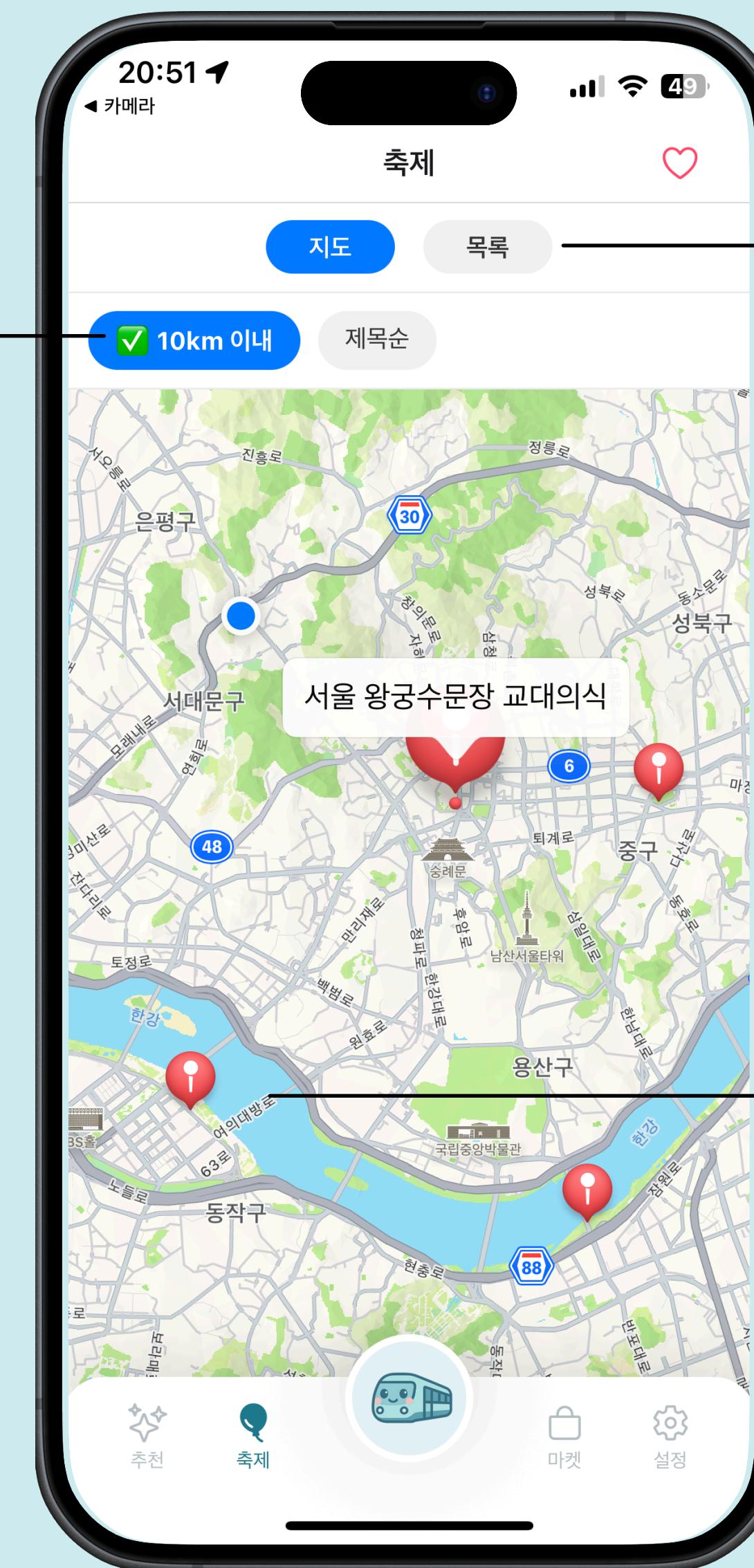


4-1. 전체 기능

기능2: 실시간 축제 정보 제공

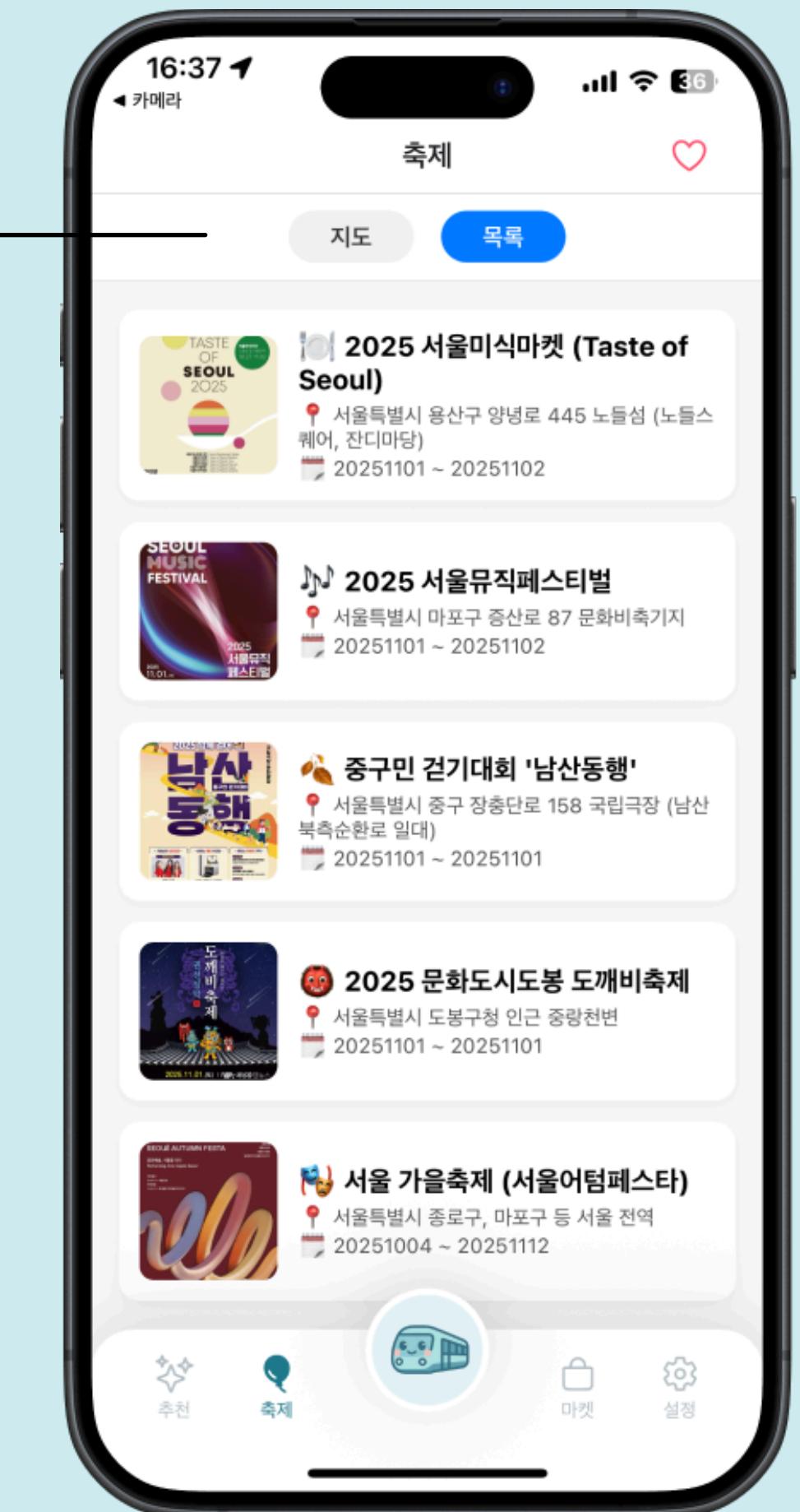


정렬 (거리순/이름순)



지도 뷰 예시

목록뷰 예시



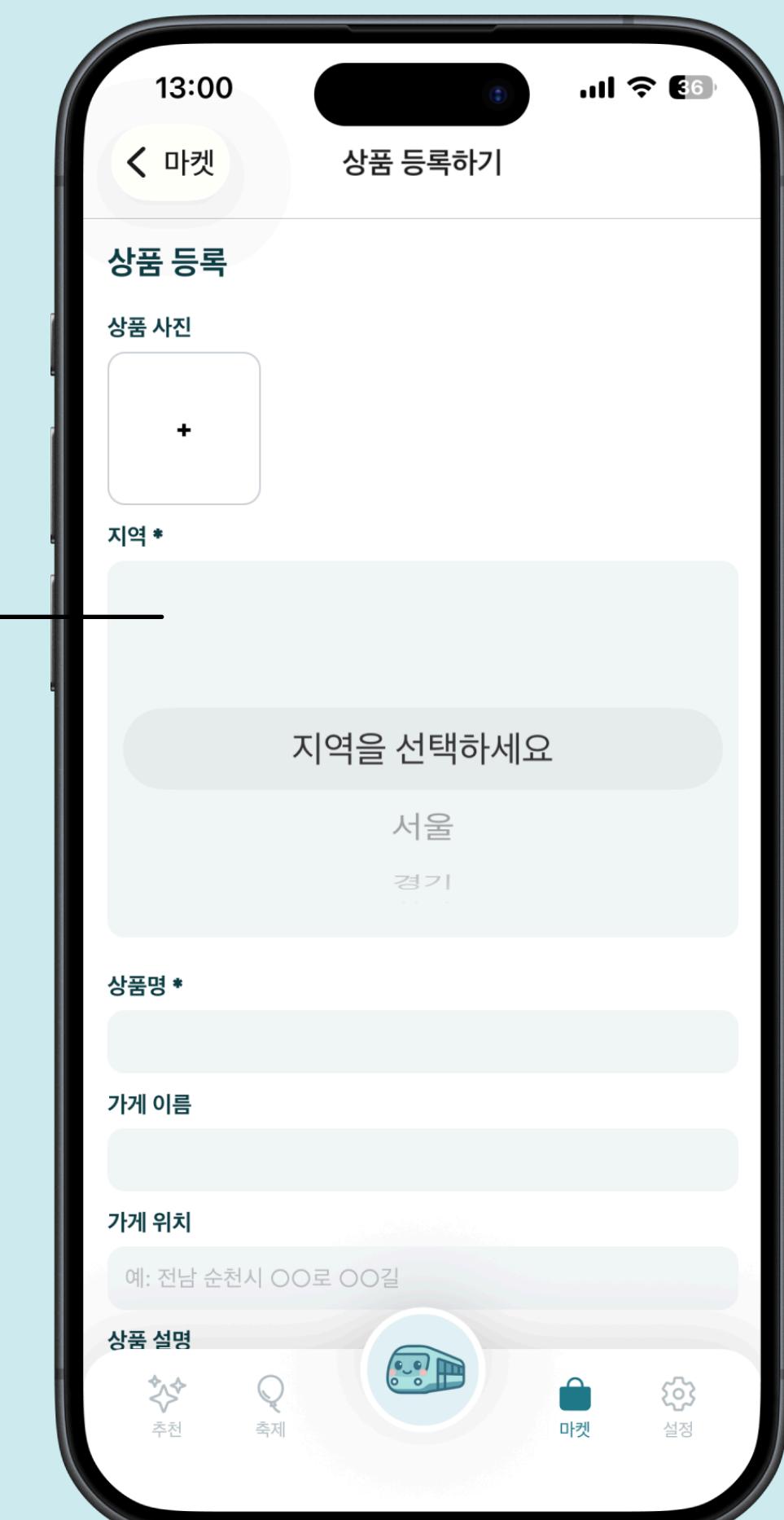
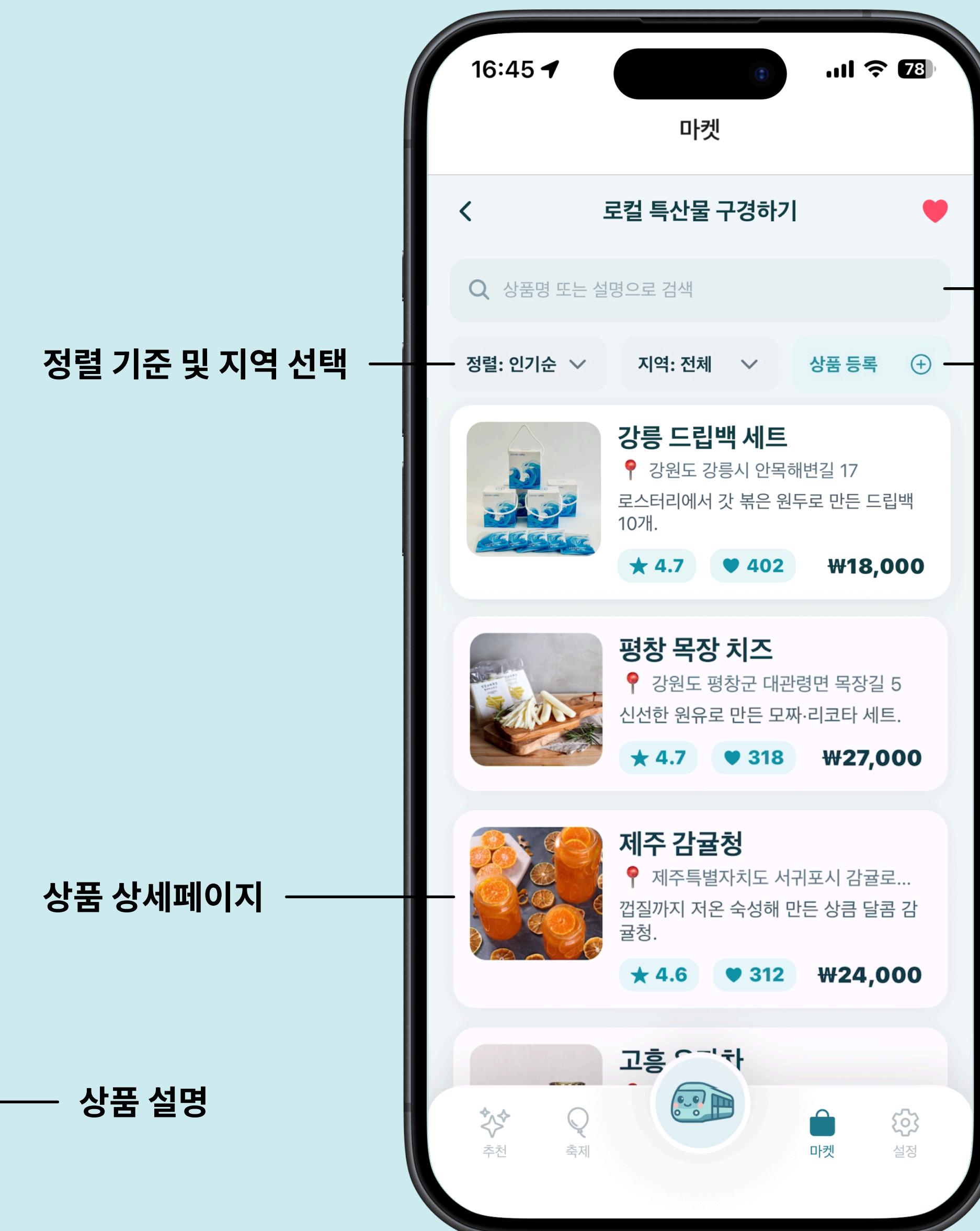
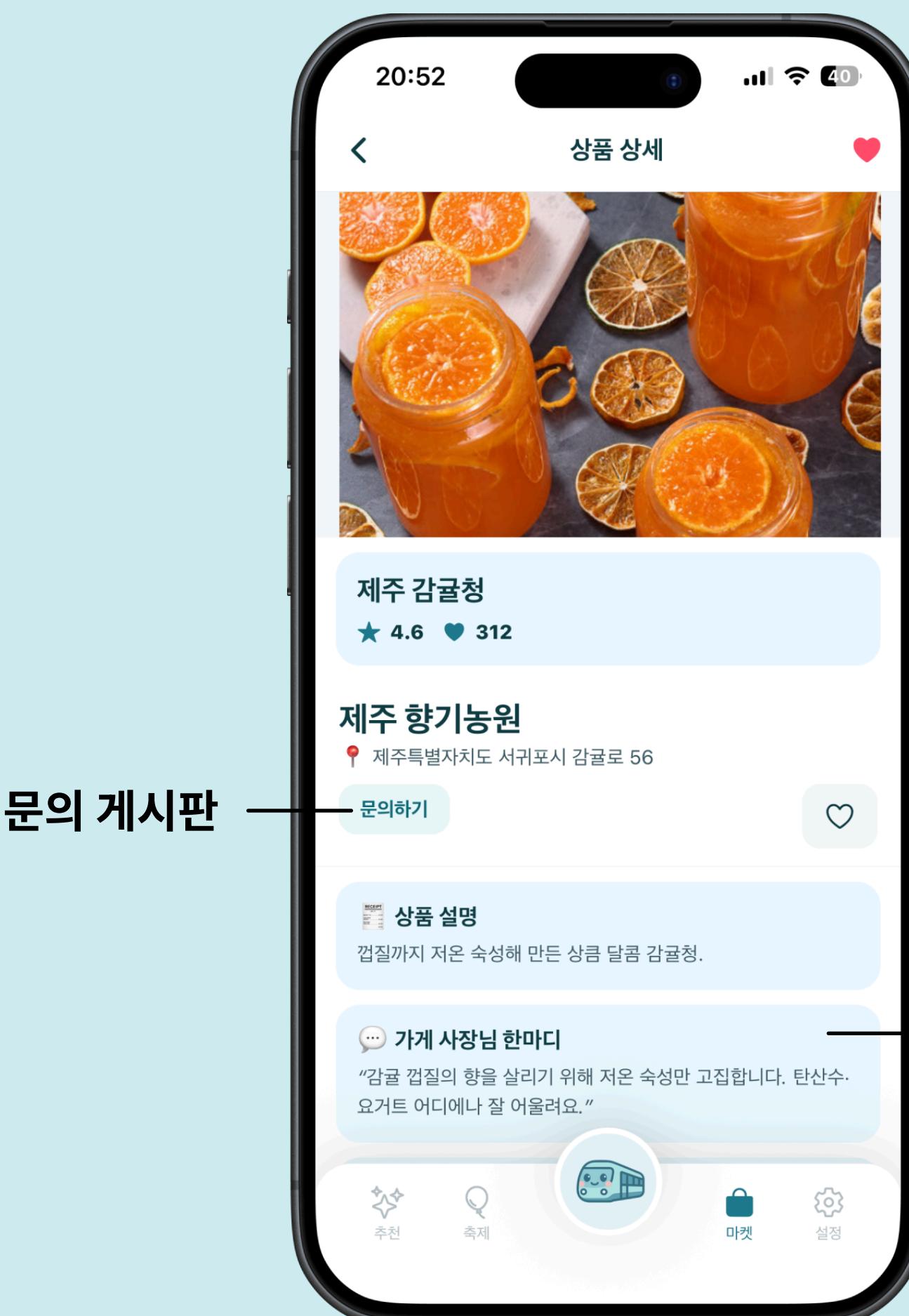
찜버튼

상세 정보 확인 가능

핀버튼 위치 표시

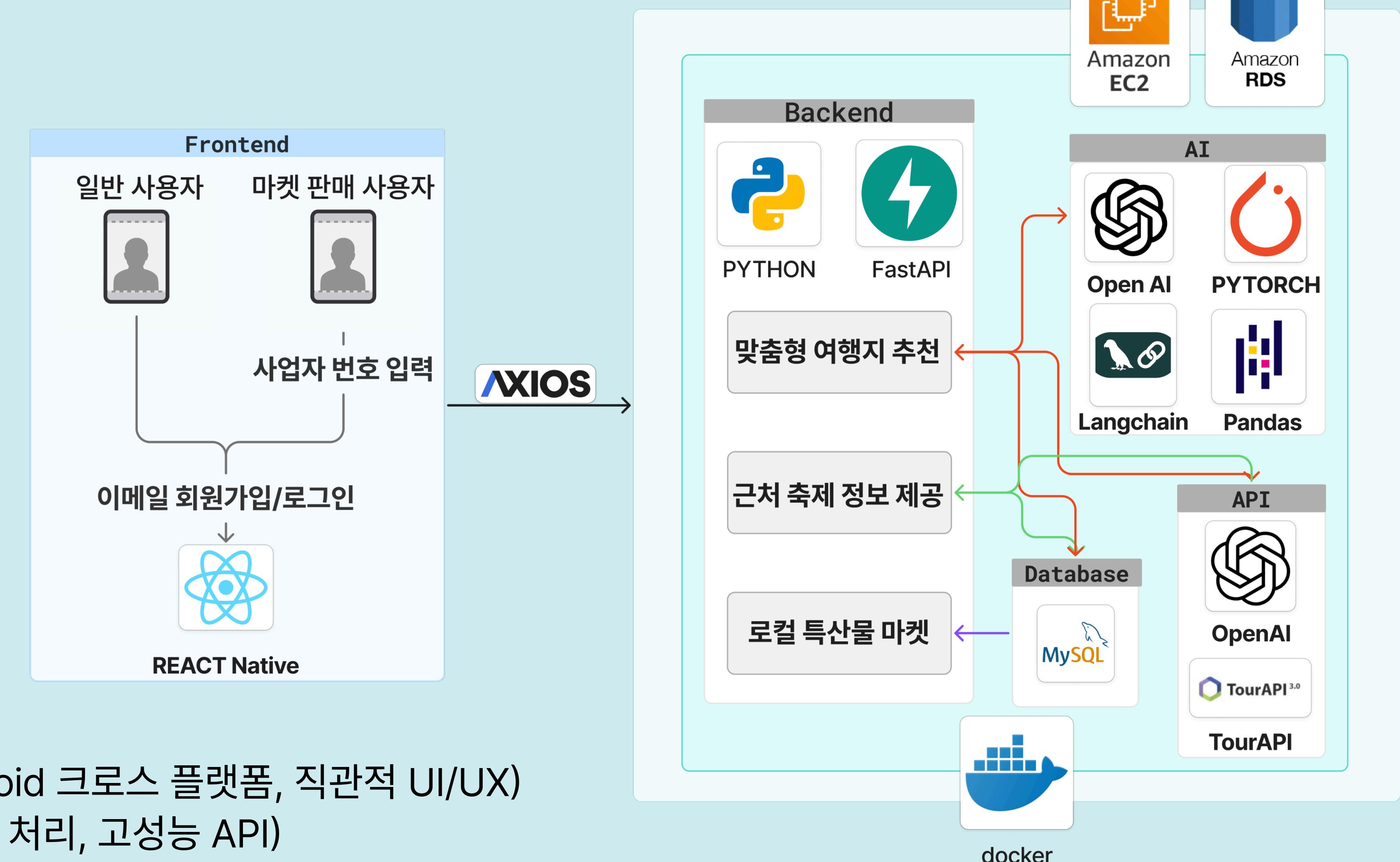
4-1. 전체 기능

기능3: 로컬 특산물 마켓



4-2. 시스템 아키텍처

[SW 구조도 요약]



Client: React Native (iOS/Android 크로스 플랫폼, 직관적 UI/UX)

Server: Python FastAPI (비동기 처리, 고성능 API)

AI Engine: LangChain + OpenAI GPT-4o + FAISS (Vector DB)

Database: MySQL (관계형 데이터), SQLAlchemy ORM

External API: 한국관광공사 TourAPI (데이터 최신성 확보)

5. 핵심 기술 구현

(1) RAG 기반 추천 엔진 Flow

사용자 질문 → 검색 키워드 추출 :

```
system_prompt_final = f"""
[Role]
당신은 소도시 여행 전문가입니다. 사용자 프로필({profile})을 고려하여 추천 멘트를 작성합니다.

[Context Data]
{spots_context}

[Mission]
위 [Context Data]의 3개 여행지에 대해 각각 맞춤형 추천 이유를 작성하고, 전체적인 소개말을 작성하십시오.

[Output Format (JSON Only)]
반드시 아래 JSON 형식을 준수하십시오.
{{

    "intro_message": "사용자님, {profile.get('style')} 스타일을 고려하여 ~한 곳들을 선정했습니다. (전체 요약 1~2문장)",
    "recommendations_detail": [
        {
            "contentid": "여행지 ID (Context Data와 동일해야 함)",
            "ai_summary": "이 여행지를 추천하는 구체적인 이유와 매력 포인트 (3~4문장)"
        },
    ]
}}"""

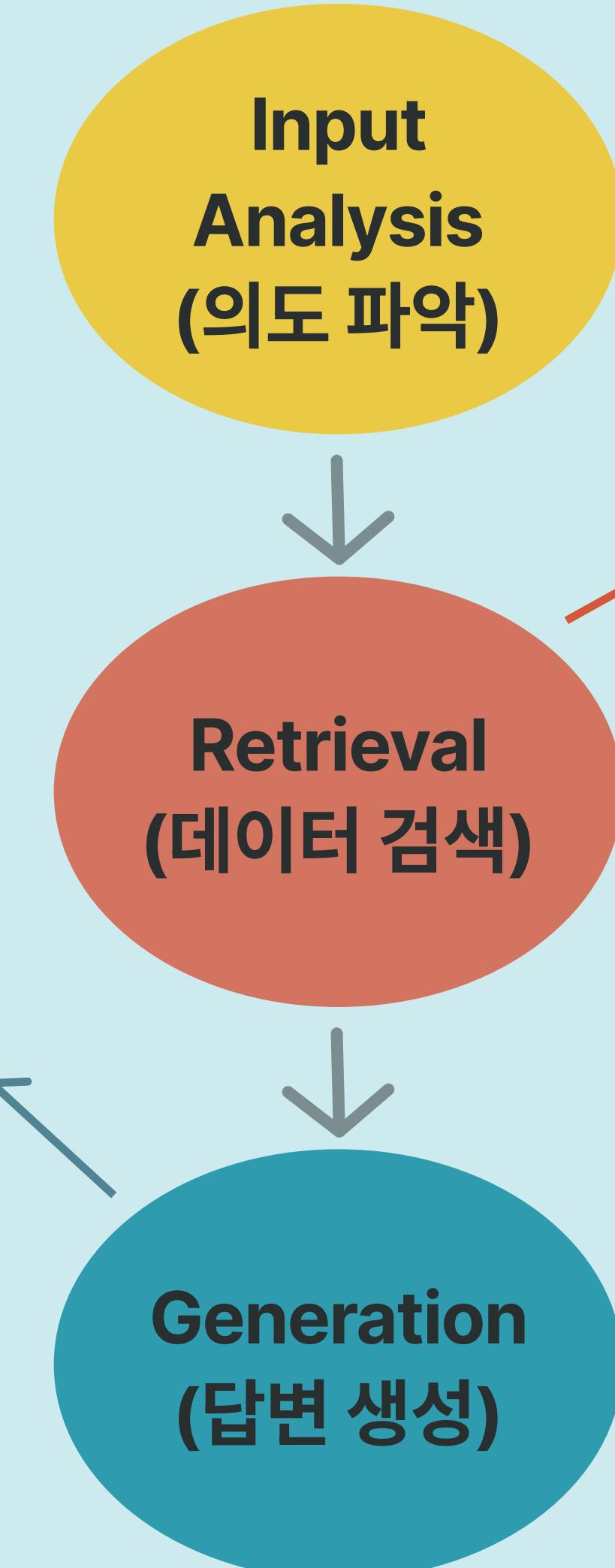
# 1. 소도시 정의: 대도시 제외
exclude_cities = ["서울", "부산", "대구", "인천", "광주", "대전", "울산", "제주"]

query = db.query(RecommendTourInfo)
for city in exclude_cities:
    query = query.filter(RecommendTourInfo.addr1.notlike(f"%{city}%"))

# 2. 키워드 검색 (OR 조건) -> keywords 중 하나라도 포함되면 결과에 포함
conditions = []
for kw in keywords:
    conditions.append(RecommendTourInfo.title.like(f"%{kw}%"))
    conditions.append(RecommendTourInfo.addr1.like(f"%{kw}%"))

if conditions:
    query = query.filter(or_(*conditions))

# 3. 결과 제한 (너무 많으면 AI 토큰 초과)
results = query.limit(3).all()
```



1. 소도시 정의: 대도시 제외
exclude_cities = ["서울", "부산", "대구", "인천", "광주", "대전", "울산", "제주"]

query = db.query(RecommendTourInfo)
for city in exclude_cities:
 query = query.filter(RecommendTourInfo.addr1.notlike(f"%{city}%"))

2. 키워드 검색 (OR 조건) -> keywords 중 하나라도 포함되면 결과에 포함
conditions = []
for kw in keywords:
 conditions.append(RecommendTourInfo.title.like(f"%{kw}%"))
 conditions.append(RecommendTourInfo.addr1.like(f"%{kw}%"))

if conditions:
 query = query.filter(or_(*conditions))

3. 결과 제한 (너무 많으면 AI 토큰 초과)
results = query.limit(3).all()

: 추출된 키워드 → DB 검색 (소도시 필터링)

```
{ "contentid": "2763801", "title": "담양 죽녹원", "addr1": "전라남도 담양군 담양읍 죽녹원로 119", "cat1": "A01", "cat3": "A01010400", "firstimage": "http://tong.visitkorea.../img.jpg", "mapx": 126.990, "mapy": 35.323 },  
{ "contentid": "127492", "title": "메타세콰이어 길", "addr1": "전라남도 담양군 담양읍 학동리 578-4", "cat1": "A01", "cat3": "A01010700", "firstimage": "http://tong.visitkorea.../img2.jpg", "mapx": 127.003, "mapy": 35.324 }
```

: 검색된 여행지 정보(Context) + 사용자 프로필
→ LLM 답변 생성

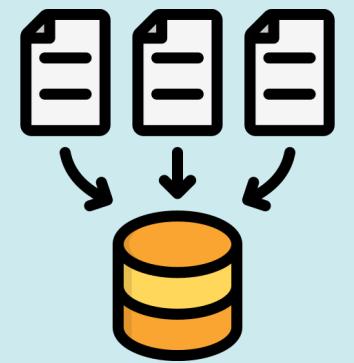
5. 핵심 기술 구현

(2) 프롬프트 엔지니어링



Role (페르소나)

전문가 역할 부여



Context (근거 데이터)

DB에서 가져온 정보 제공



Constraints (제약 조건)

JSON 형식 준수, 언어 스타일 등

- ① [Context Data]: DB에서 조회한 실제 여행지 데이터를 프롬프트에 직접 주입 → 환각 억제

```
[Context Data]
{spots_context}
```

[Mission]

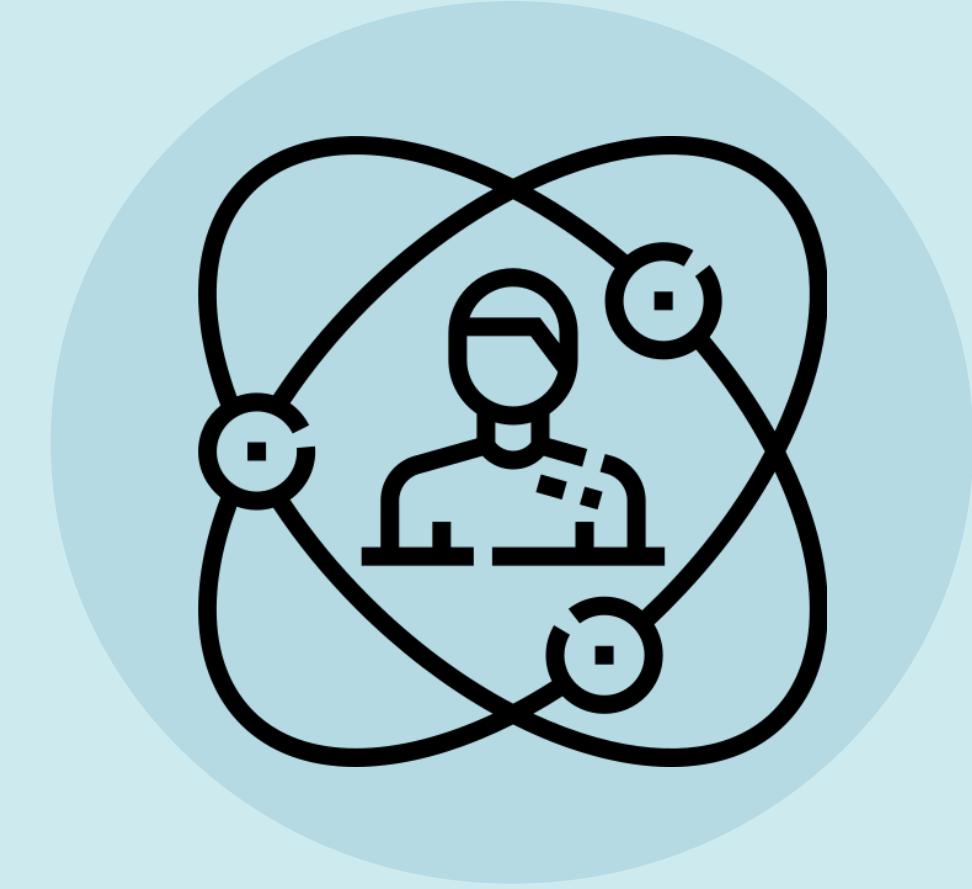
위 [Context Data]의 3개 여행지에 대해 각각 맞춤형 추천 이유를 작성하고, 전체적인 소개말을 작성하십시오.

- ② [Output Format] JSON 강제화 → 파싱 에러를 방지

```
[Output Format (JSON Only)]
반드시 아래 JSON 형식을 준수하십시오.

{{"
  "intro_message": "사용자님, {profile.get('style')} 스타일을 고려하여 ~한 곳들을 선정했습니다. (전체 요약 1~2문장)",
  "recommendations_detail": [
    {
      "contentid": "여행지 ID (Context Data와 동일해야 함)",
      "ai_summary": "이 여행지를 추천하는 구체적인 이유와 매력 포인트 (3~4문장)"
    },
  ]
}}
```

6. 기대효과



사용자 경험 혁신 (Traveler)

- 복잡한 검색 없이 대화만으로 내 취향에 딱 맞는 '인생 여행지'를 발견할 수 있습니다. (정보 탐색 시간 단축, 여행 만족도 증대)
- 실패 없는 여행 계획 수립이 가능해집니다.



지역 경제 활성화 (Economy)

- 소외되었던 소도시의 숨은 명소를 발굴하여 관광객을 분산시키고 지역 균형 발전에 기여합니다.
- 관광 수익이 숙박/식음을 넘어 지역 특산물 구매로 이어지는 경제적 낙수 효과를 창출합니다.



소상공인 상생 (Local Seller)

- 별도의 복잡한 마케팅 없이도 관광객에게 상품을 노출할 수 있는 디지털 판로를 확보합니다.
- 일회성 방문객을 단골 고객으로 전환할 수 있는 기회를 제공합니다.

Demo

demo start