

PORTFOLIO.

사용자 경험을 중심으로 개발하는 프론트엔드 개발자

프론트엔드 개발자 | 김민석

2025

About Me.



사용자를 생각하는 인터페이스,
좋은 경험을 만드는 프론트엔드 개발자 김민석입니다.
저는 직관적이고 친화적인 웹 인터페이스 구현에 관심이 많습니다.
새로운 환경에도 두려움 없이 도전하며,
빠르게 변화하는 프론트엔드 트렌드에 맞춰
꾸준히 학습하고 책임감을 가지고 성장하고 있습니다.

Education.

성결대학교 학사

2022년 졸업

LG유플러스 유레카 프론트엔드 교육과정

2025.01~2025.08

Tech Stack



React

팀 프로젝트에서 SPA 구조 설계,
컴포넌트 재사용성 최적화 경험



Tailwind CSS

반응형 UI 구현 및
빠른 개발 속도를 위해 활용



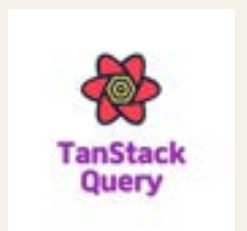
Redux

전역 상태 관리 적용,
복잡한 데이터 흐름 개선



TypeScript

타입 안정성 확보 및 유지보수 효율성 개선



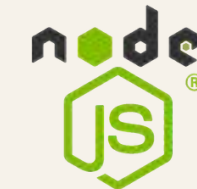
TanStack Query

API 연동 시 캐싱과 비동기 데이터 관리를 적용해,
중복 요청을 줄이고 사용자 경험을 개선함



SpringBoot

간단한 로그인 및 CRUD 기능 구현 경험



Node.js(express)

RESTful API 설계 및 CRUD 기능 개발 경험



GitHub

협업 시 브랜치 전략, 코드 리뷰 및 효율적인 프로젝트 경험



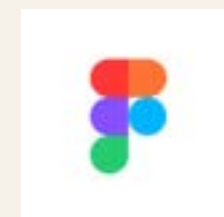
Jira

효율적인 프로젝트 관리 및 업무 우선순위 설정 경험



Slack

팀 커뮤니케이션 및 일정 공유



Figma

와이어프레임 제작 및 화면 설계를
추후 디자이너와 의사소통 능력 향상

PROJECT 01

로드픽

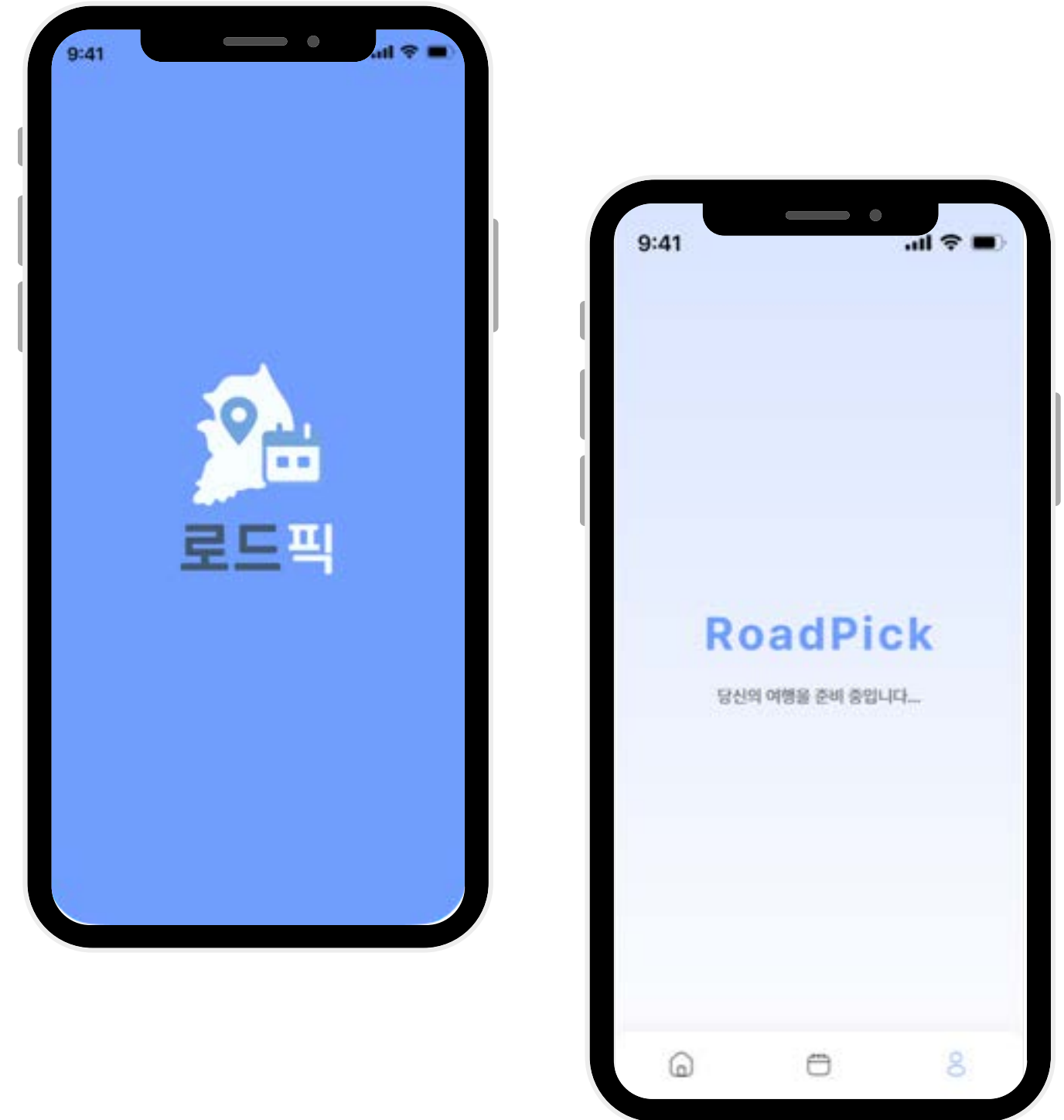
나만의 여행을 설계하는 스마트 여행 일정 플랫폼

로드픽은 사용자가 맞춤형 여행 일정을 손쉽게 설계할 수 있도록 여행 정보, 축제/행사 추천 등 다양한 콘텐츠를 제공하는 스마트 여행 일정 관리 플랫폼입니다.

Frontend: React, CSS Module, React Query 활용

Backend: Node.js, Express

협업: GitHub, Slack, Notion



공공 API 정리부터 호출 최적화까지 – 커스텀 Hook 기반 통합 관리

API 호출 구조 설계 및 최적화 핵심 포인트

```
import { getLocationTourList } from '@apis/getLocationTourList'
import { useQuery } from '@tanstack/react-query'

export const useLocationTourList = (mapX, mapY, radius, contentTypeId) => {
  return useQuery({
    queryKey: ['LocationTourList', mapX, mapY, radius, contentTypeId],
    queryFn: () => getLocationTourList({ mapX, mapY, radius, contentTypeId }),
    enabled: !!mapX && !!mapY, // mapX, mapY는 필수 없으면 요청 안 보냄
    staleTime: 1000 * 60 * 5, // 5분 동안 fresh 상태 유지
  })
}
```

React Query 설정으로 데이터 캐싱 처리

- API 요청 구조 설계 및 커스텀 Hook 구현
- 반복 요청으로 인한 429 에러
→ React Query 캐싱 적용
- staleTime 설정으로 재요청 최소화, 안정성 확보
- 서버 부하 감소 & 사용자 응답 속도 향상

디테일 페이지 구현



백제문화단지

★ 4.6 (59)

충청남도 부여군 규암면 백제문화로 455

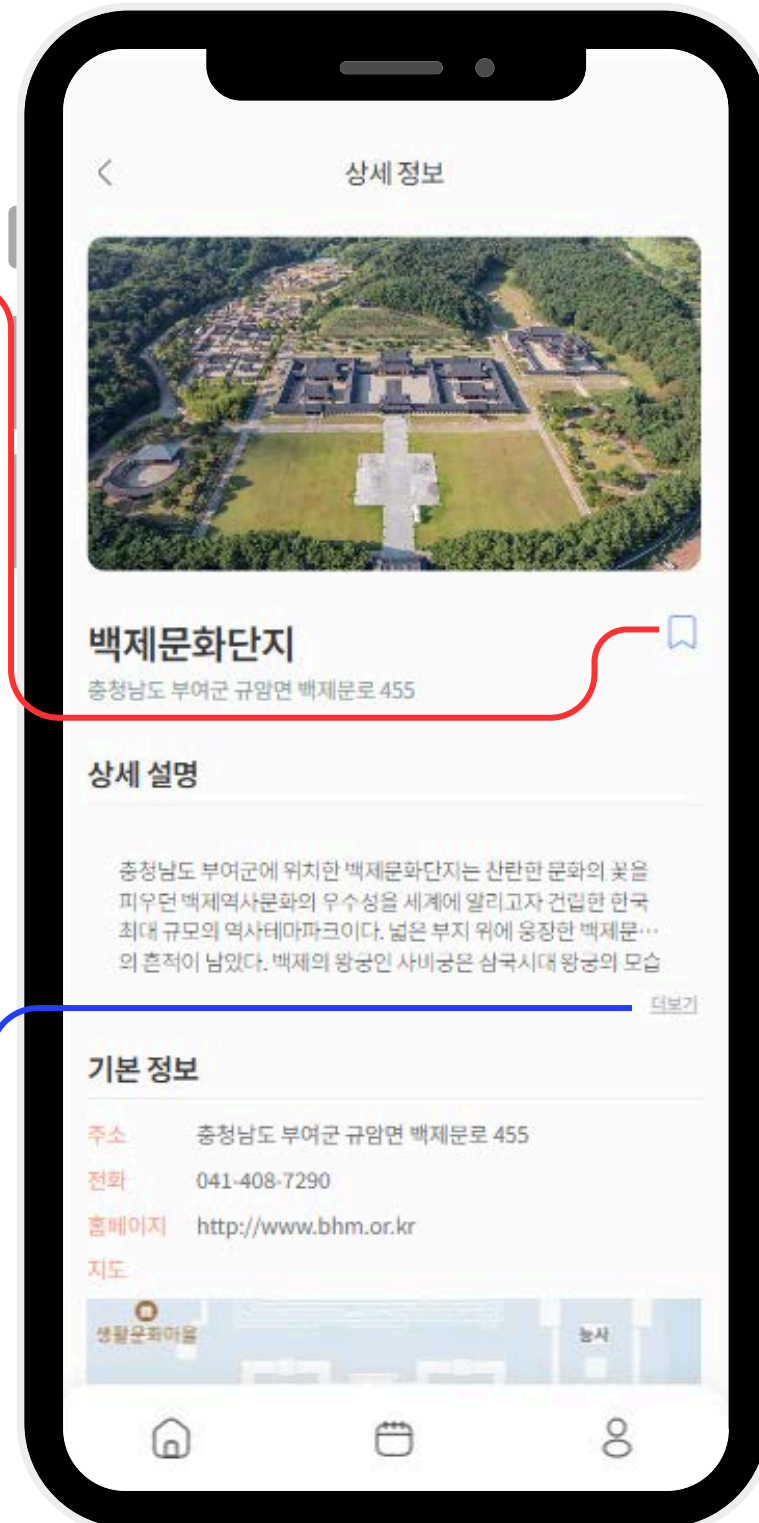
저장/삭제 동일하게 가능

상세 설명

충청남도 부여군에 위치한 백제문화단지는 찬란한 문화의 꽃을 피우던 백제역사문화의 우수성을 세계에 알리고자 건립한 한국 최대 규모의 역사테마파크이다. 넓은 부지 위에 웅장한 백제문화의 흔적이 남았다. 백제의 왕궁인 사비궁은 삼국시대 왕궁의 모습을 최초로 재현했다. 백제의 왕실 사찰 능사는 부여읍 능산리 유적을 실물 크기로 복원했다. 중문인 대동문을 지나 마주하는 능사 오층 석탑은 단지에서 가장 두드러지는 건축물이다. 위례성은 한성백제 시기의 성으로 사비궁과는 다른 시대 백제의 모습을 살펴볼 수 있다. 제형루에 올라하면 단지 전체가 한눈에 들어온다.

간략히

더보기/간략히



해당 지도 이동

관련 축제 정보

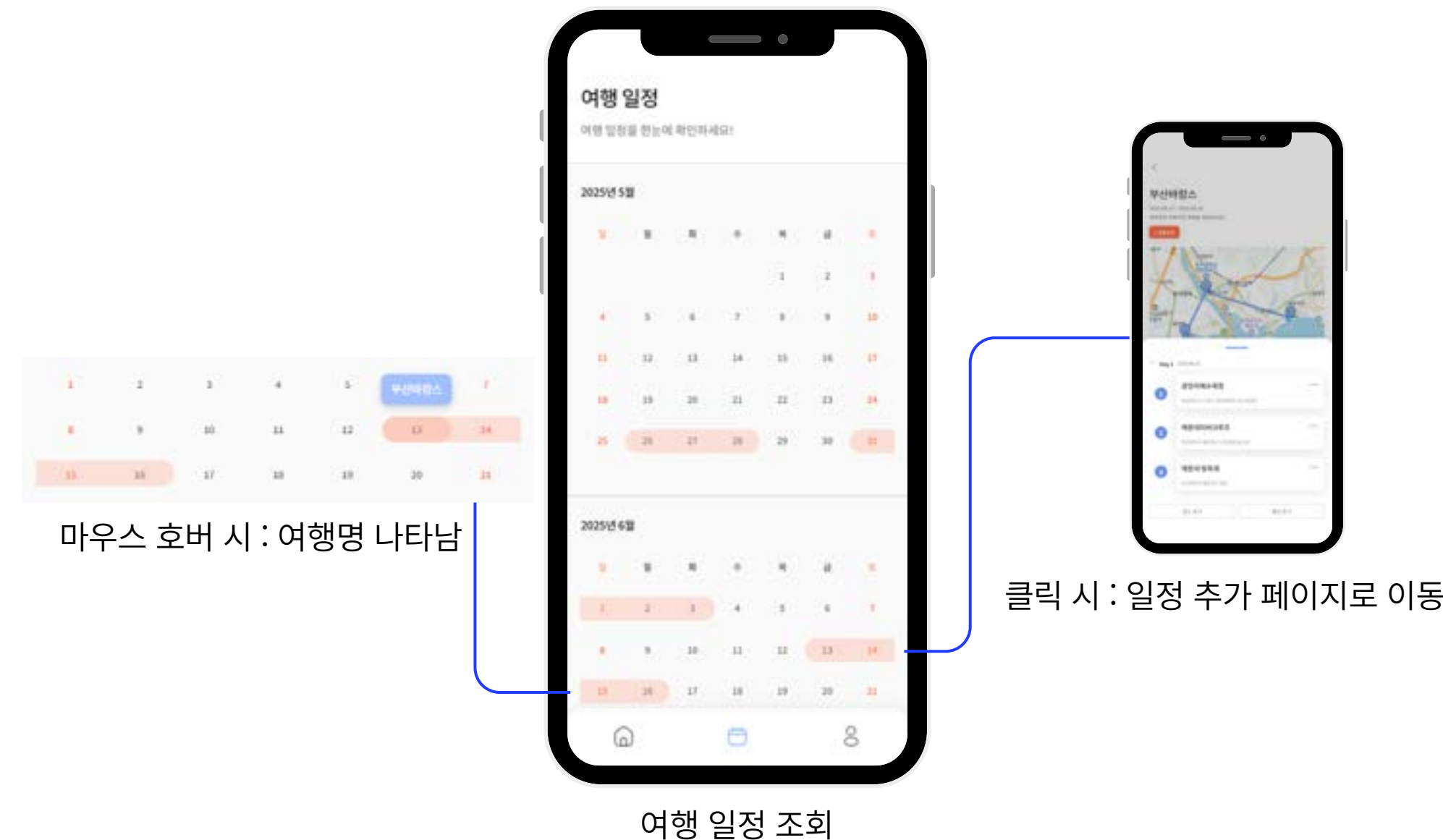
관련 축제 정보가 없습니다.

축제 정보 없을 때

캘린더 페이지 – 일정 생성 및 조회 기능 구현

- 독립 캘린더 컴포넌트로 재사용 일정 생성·조회 구현
- 출발/도착 날짜 선택 시 자동 기간 산출
→ 일정 생성 활성화
- 생성: 여행명/기간 자동 입력, 필요 시 편집 가능
- 조회: 캘린더 Hover로 여행명 미리보기
Click시 상세 작성 페이지 이동

캘린더 페이지



여행 계획 페이지 – 장소 저장, 날짜별 정리 및 지도 시각화

- 날짜별 데이터 분리 구조 → 많은 호출로 429 Error 발생
- 개선: 데이터 일괄 조회 후 프론트에서 분류 하여 호출을 한번만 하도록 개선
- id와 name 위도 경도를 가진 planMap 객체로 Kakao Map 연동
- KakaoMap Api PolyLine으로 일정 간 시각적 연결 구현

```
// 날짜별로 그룹화하는 유틸 함수
const groupActivitiesByDate = activities => {
  if (!Array.isArray(activities)) return {}
  return activities.reduce((acc, activity) => {
    const date = formatDateToLocalString(activity.visitDate)
    if (!acc[date]) acc[date] = []
    acc[date].push(activity)
    return acc
  }, {})
}

// 전체 일정 한번에 불러오기
const fetchAllDetailPlans = async () => {
  if (!schedule?.tripId) return
  try {
    const allActivities = await getScheduleDetailList(schedule.tripId)
    const grouped = groupActivitiesByDate(allActivities)
    setActivitiesByDate(grouped)
  } catch (err) {
    console.error('일정 전체 조회 실패:', err)
  }
}
```

You, 2개월 전 • 플랜 페이지 기능 추가 ...

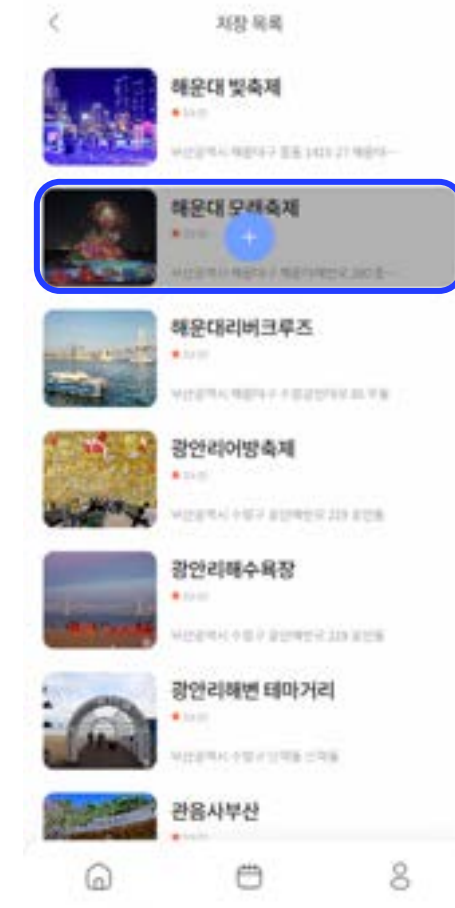
여행 계획 페이지 – 장소 저장, 날짜별 정리 및 지도 시각화

주요 기능 요약

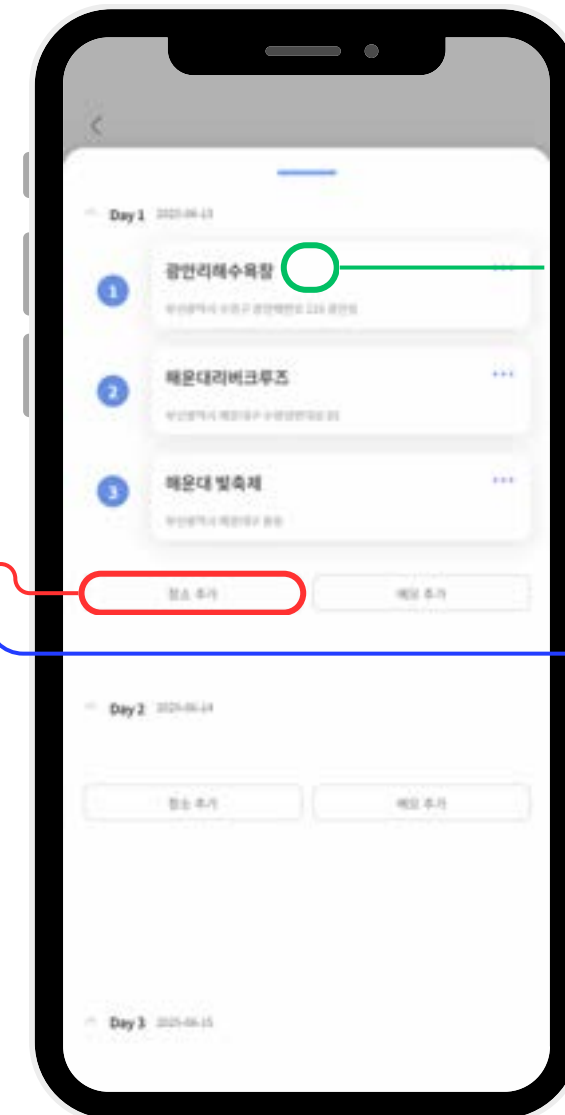
- 지도에 동선 시각화: 사용자가 입력한 장소를 경로 순서대로 선으로 연결
- 날짜별 일정 시트 작성: 날짜별 장소를 테이블 형식으로 입력 및 수정
- 모달로 장소 추가 및 설명 입력
- 장소 리스트로 빠른 추가 가능



순서에 맞는 동선을 나타낸 지도



저장 목록에서
장소 불러오기



바텀 시트 올릴 시



장소 추가 모달

트러블 슈팅

1. 페이지 로딩 지연 대응

문제

- 첫 페이지 진입 시 컴포넌트 및 이미지 로딩 지연
- 사용자 체감 속도 저하 및 초기 렌더 지연 발생

해결

- React.lazy, Suspense 활용 → 필요 시점에 컴포넌트 동적 로딩
- 이미지에 loading="lazy" 적용 → 화면 밖 이미지 지연 로딩
- 초기 화면에 보이는 요소에는 적용 제외

효과

- 초기 진입 속도 개선 → 빠른 첫 화면 응답
- UX 개선 및 Core Web Vitals 점수 향상

2. API 호출 오류 대응 + 캐싱 처리

문제

- 외부 API 과도 호출로 인한 429 Too Many Requests 에러
- 간헐적 실패 시 빈 화면 or 데이터 미노출 현상

해결

- 정상 호출 시 데이터 → sessionStorage에 캐싱
- 오류 발생 시 캐시 데이터로 대체 렌더링
- TanStack Query (React Query) 사용

프로젝트 경험 & 느낀 점

1. Figma 사용 및 기획

- 기획부터 참여 : 요구사항 정의의 중요성 체감
- Figma 설계 경험 : UI/UX 중심 사고 확립
- 변경 대응 경험 : 문서화 · 버전 관리 필요성 인식
- 팀 협업 배움 : 코드보다 기획과 소통이 핵심
- 성장 포인트 : 문제 해결력 & 전체 프로세스 이해 확장

2. 공통 컴포넌트의 중요성 및 팀원과의 소통의 중요성

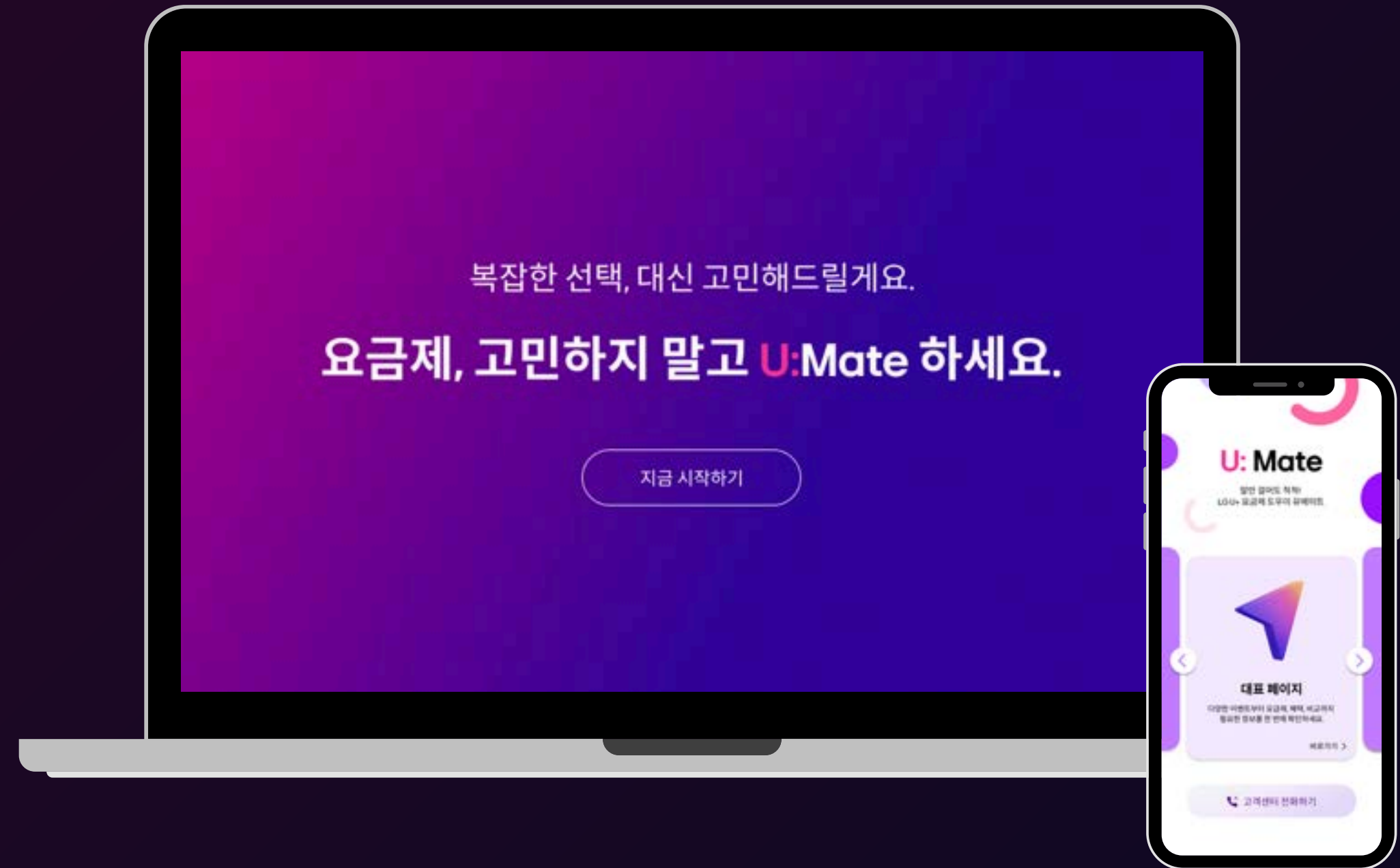
- 공통 컴포넌트의 중요성 : 중복 개발을 줄이고, 유지보수 효율을 높임
- UI 일관성 확보 : 공통 컴포넌트 사용으로 사용자 경험 향상
- 팀원과의 소통 : 기획 변경 시 빠른 피드백으로 충돌 최소화
- 협업 프로세스 개선 : 문서화 · 규칙 합의 → 생산성 향상

PROJECT 02

U+ 모든 세대의 디지털 친구
사용자 맞춤 요금제 도우미

U: Mate

AI챗봇과 요금제 조회·비교·리뷰 기능을 통해 합리적인 선택을 돕고,
디지털 취약 계층(시니어)도 쉽게 이용할 수 있도록
접근성과 편의성을 강화하는 것이 목적입니다.



공통 컴포넌트 개발

01 헤더

Props를 이용해 페이지별 맞춤 헤더 적용, 반응형 적용
로그인 상태 반영, Outlet Context 연동을 사용해
상위 컴포넌트에 전달하여 헤더 즉시 업데이트

02 인풋 박스

프로젝트 전역에서 재사용 가능한 입력 필드 UI 제공
Variant와 Props로 유연하게 제어
다양한 페이지에서 동일한 UX와 스타일 보장
기능 확장이 용이하도록 구현하였음.

03 요금제 카드

레이아웃은 분리 유지 하되
중복되는 요소만 모듈화(별점, 가격 블록, 공통 컨테이너)

```
const setHeaderConfig = useOutletContext<(config: HeaderProps) => void>();  
useEffect(() => {  
  setHeaderConfig({  
    title: '마이페이지',  
    showBackButton: true,  
    showSearch: false,  
    hasShadow: true,  
  });  
}, []);  
// React Hook useEffect has a missing dependency: 'setHeaderConfig'. Eit
```

비밀번호

휴대폰 번호 입력 ex) 01012345678

5G 프리미어 플러스 > ★ 3.0 (15)
데이터 무제한
테더링+웨어링 100GB
월정액 105,000원
약정할인가 73,500원

★ 3.0 (15)
5G 프리미어 에센셜 >
데이터 무제한
테더링+웨어링 70GB
월 85,000원

약정 할인 시 월 58,500원

비교하기

변경하기

회원가입 페이지 구현

01 유효성 검사

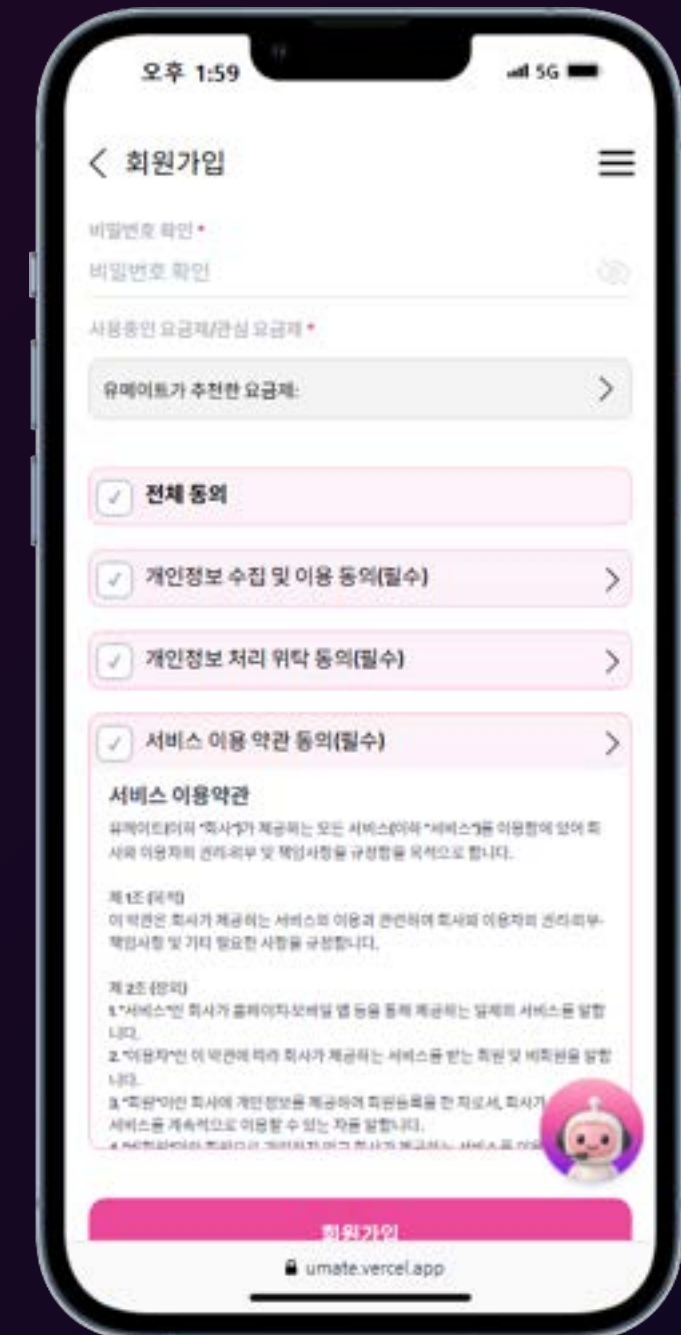
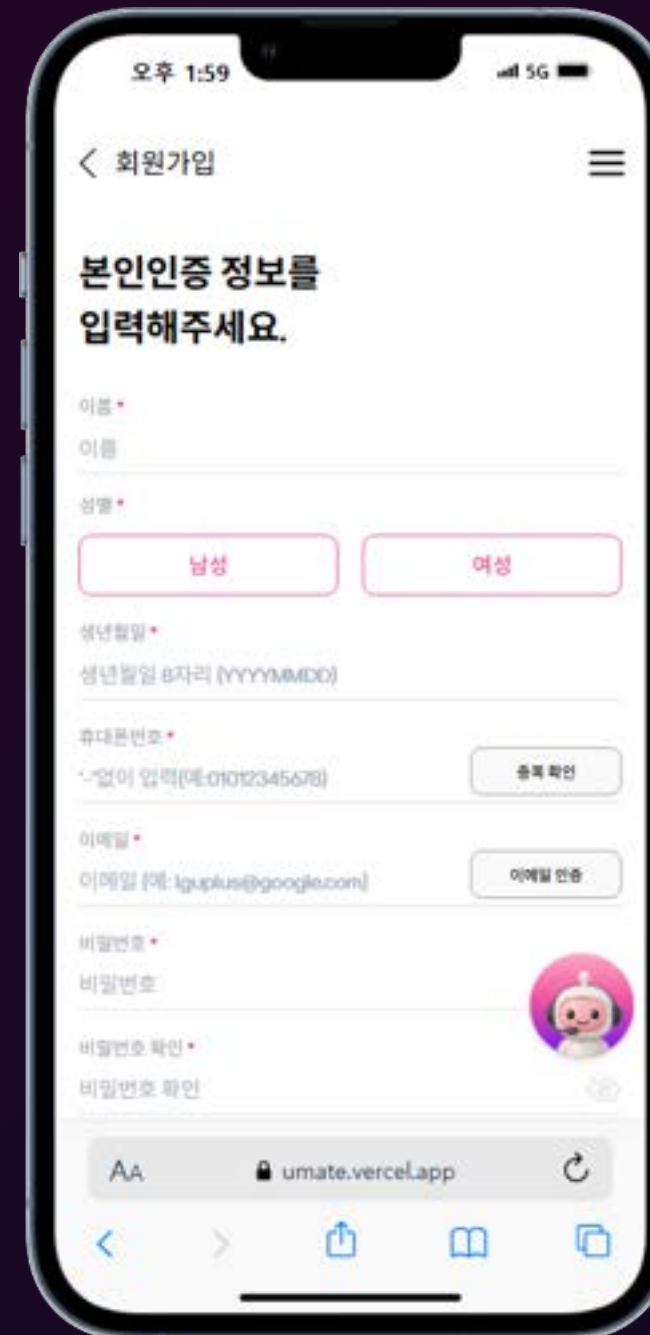
- 입력값 즉시 검증 & 피드백
- 실패 시 자동 포커스 이동 → 사용자 경험 향상
- 모든 필드 검증 후 요청 전송 → 서버 부하 방지

02 휴대폰 중복,이메일 인증

- 휴대폰 DB 중복 확인 → 오류 차단
- 이메일 인증: 코드 발송·입력 → 실시간 인증

03 사용(관심) 요금제 선택,이용약관

- 관심사 기반 요금제 추천
- 선택 후 약관 동의 → 회원가입 완료



회원가입 페이지 구현

01 유효성 검사

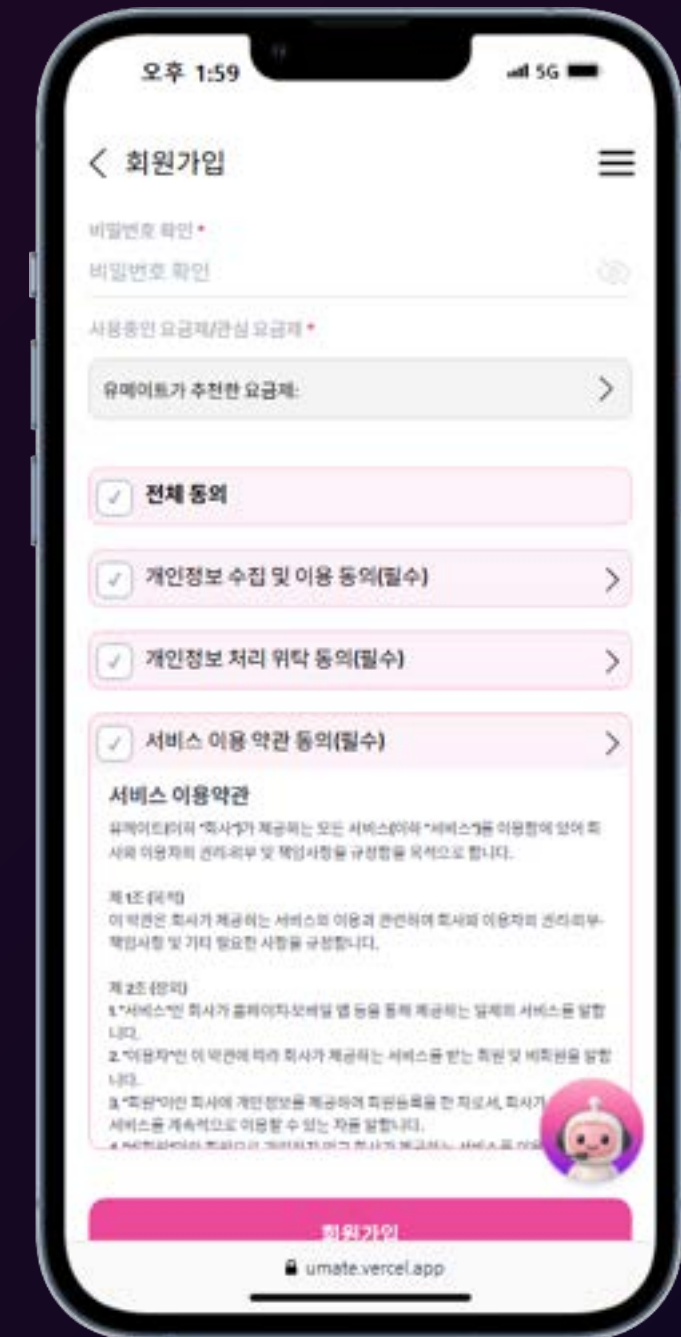
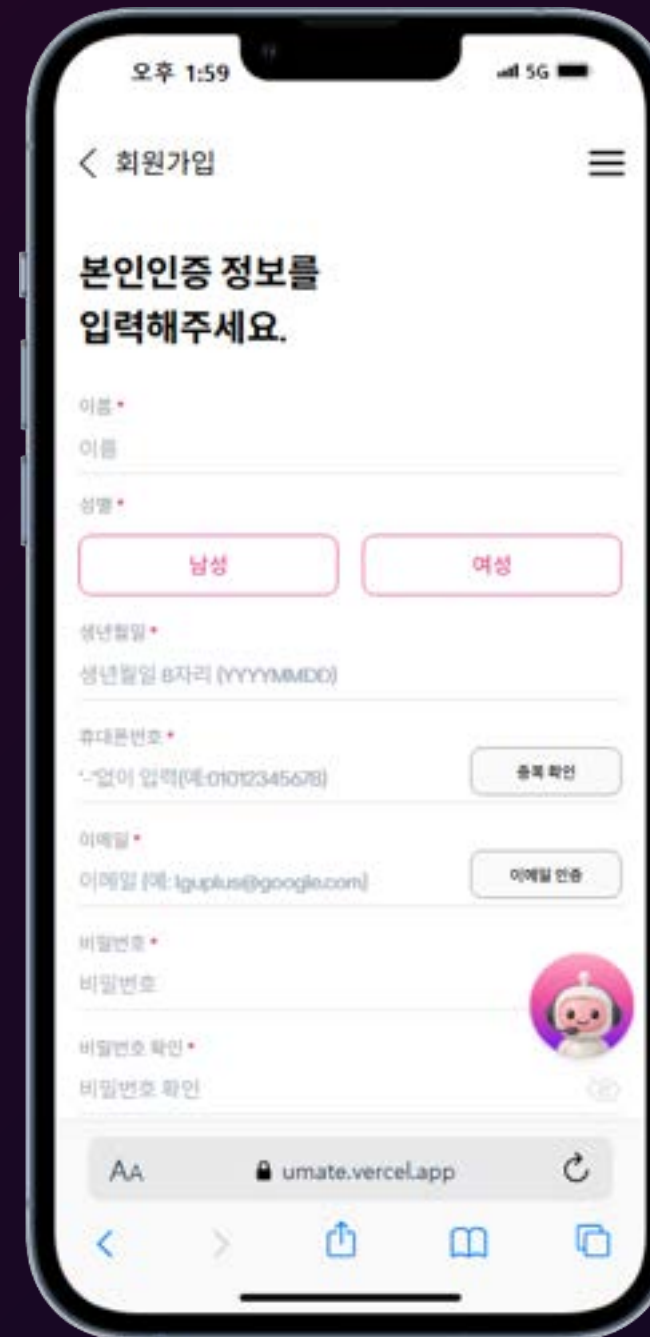
- 입력값 즉시 검증 & 피드백
- 실패 시 자동 포커스 이동 → 사용 편의성 ↑
- 모든 필드 검증 후 서버 전송 → 불필요한 요청 ↓

02 휴대폰 중복, 이메일 인증

- 휴대폰/이메일 중복 확인 → 오류 차단
- 인증 코드 발송·입력 → 실시간 인증 완료

03 사용(관심) 요금제 선택, 이용약관

- 관심사 기반 요금제 추천 → 맞춤형 경험 제공
- 전체 약관 동의 후 회원가입 완료



로그인 페이지 구현

01 유효성 검사

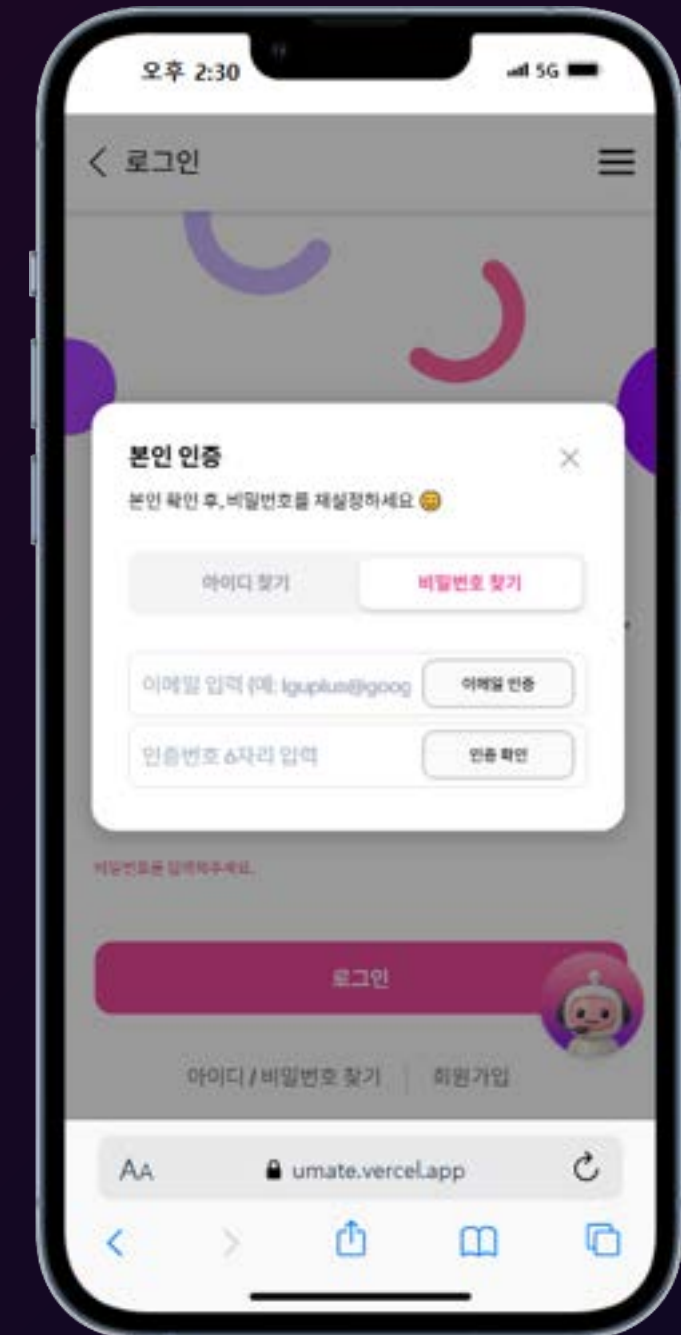
- 로그인/인증 입력값 실시간 검증
- 자동 포커스 이동 → 오류 즉시 수정 가능
- 불필요한 서버 요청 최소화 → UX 향상

02 아이디, 비밀번호 찾기

- 휴대폰 번호 기반 아이디 확인
- 이메일 인증 → 코드 발송 & 입력 → 실시간 확인

03 비밀번호 리셋

- 이메일 인증 후 새 비밀번호 재설정
- 보안성 강화 + 로그인 지속성 확보

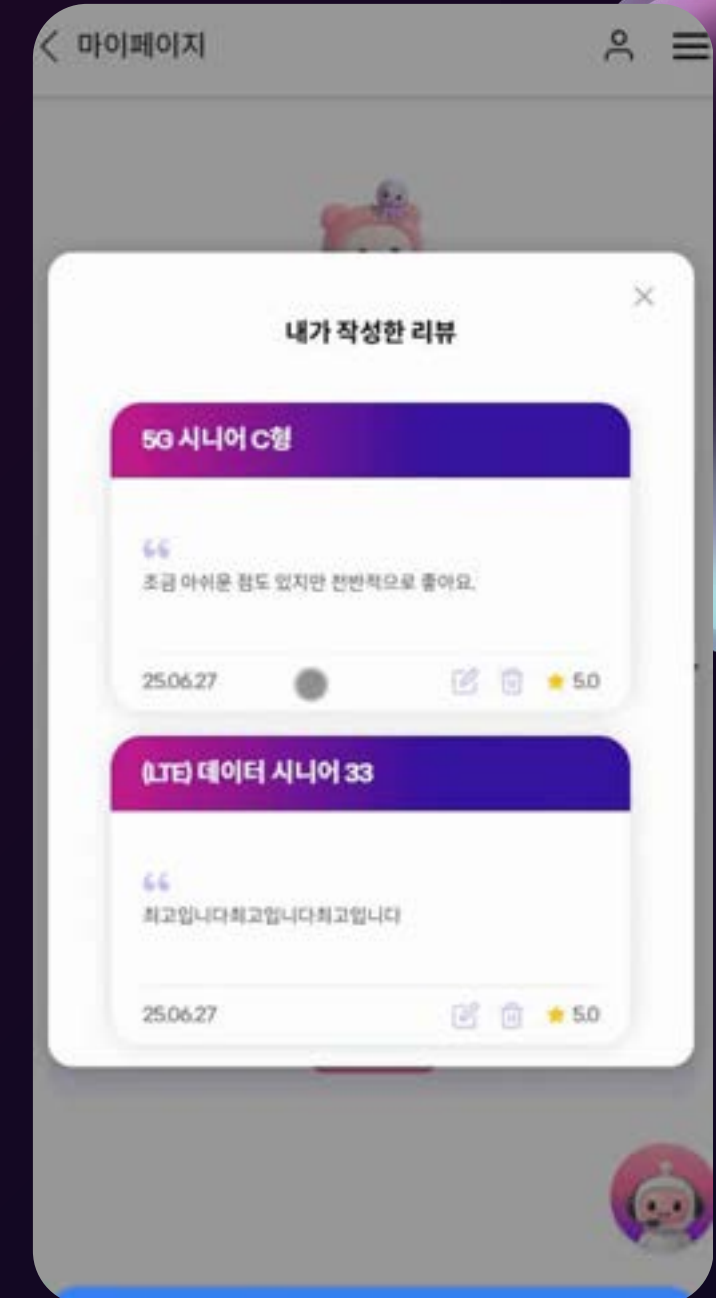
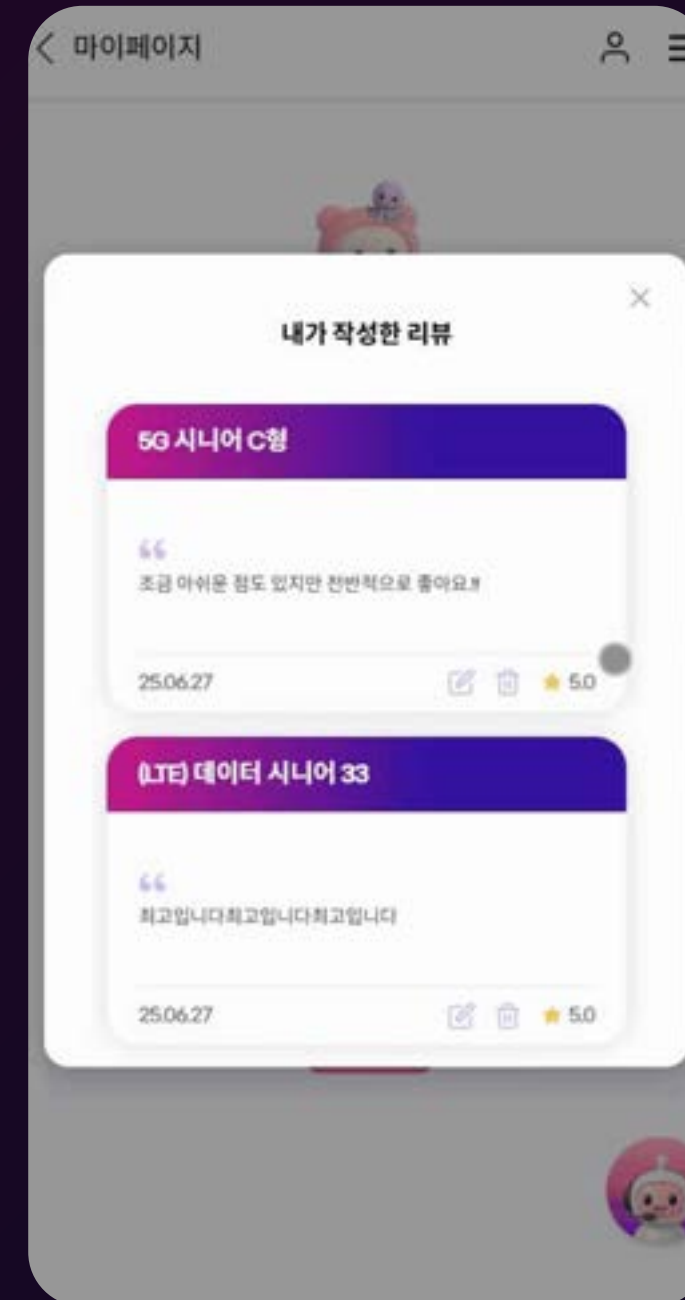
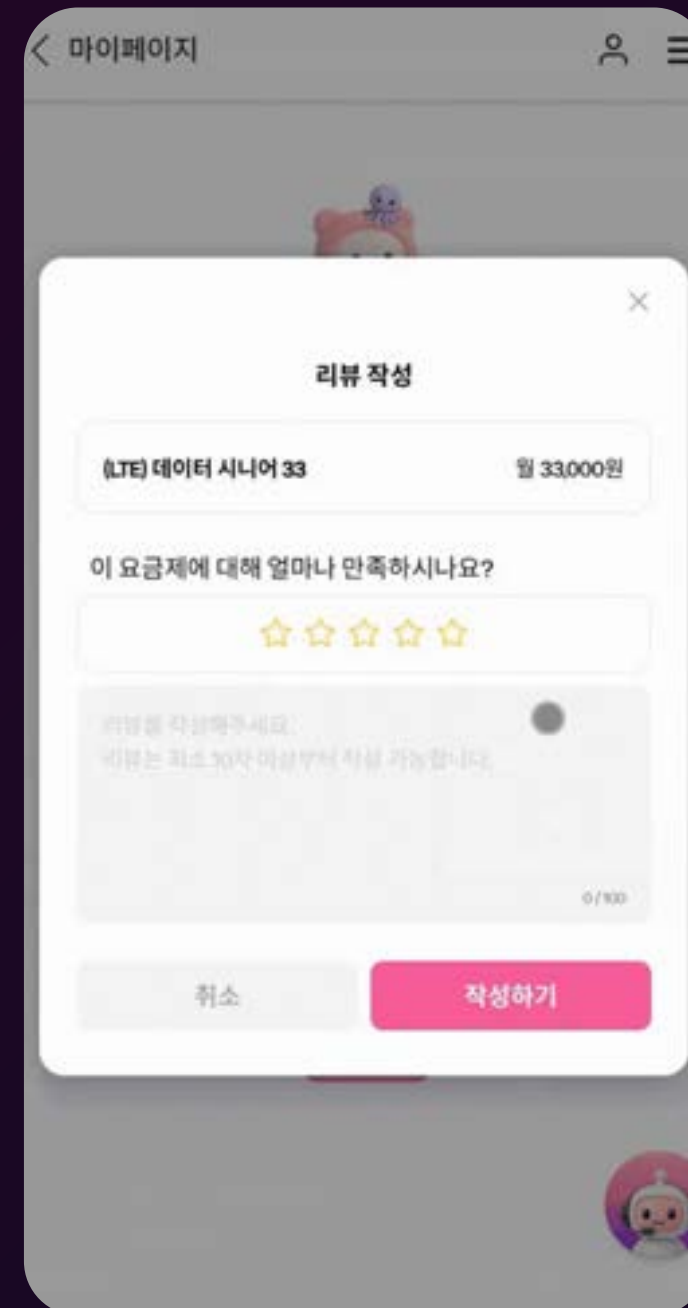


마이 페이지 -요금제별 사용자 리뷰 작성/조회/삭제/수정

01 사용자가 사용하고 있는 요금제에 대한 별점과 후기 작성

02 작성된 리뷰는 요금제 상세 페이지에서 다른 사용자와 공유

03 사용자가 작성한 리뷰를 마이페이지에서 언제든지 수정 또는 삭제 가능



마이 페이지 - 사용자 정보 확인 및 관리

01

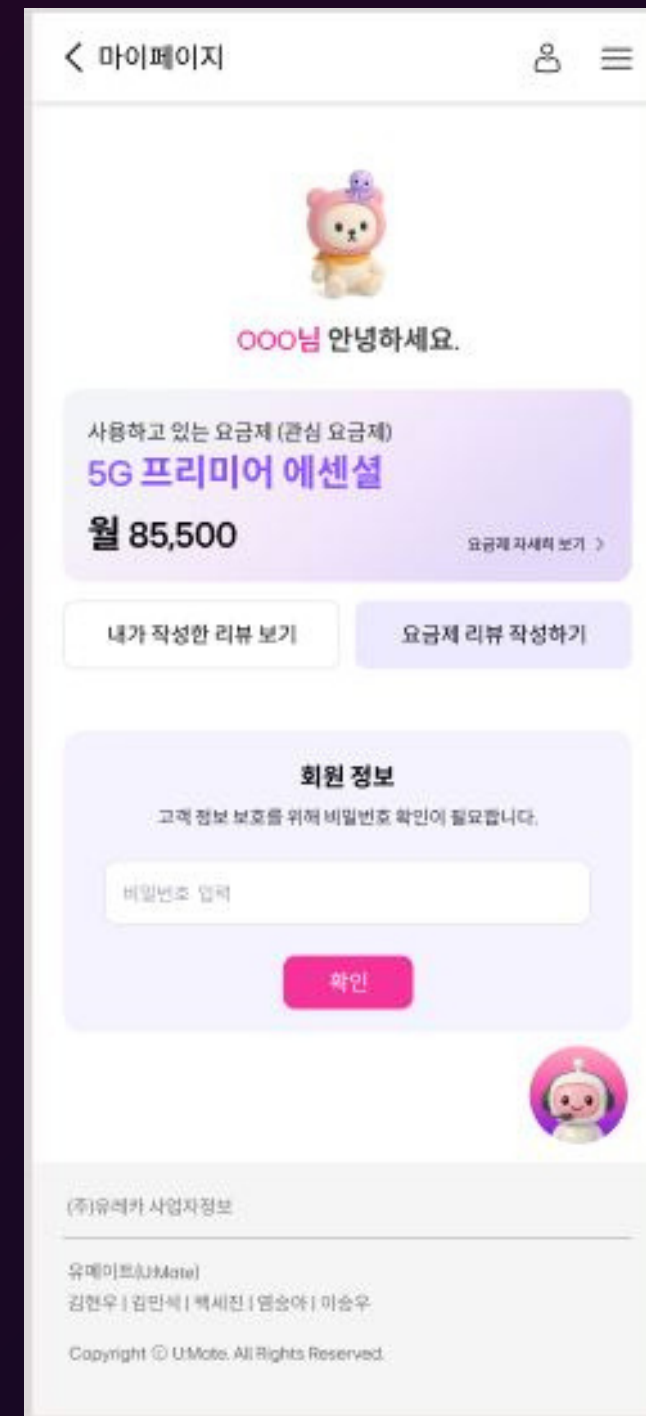
사용자의 현재 사용중인 요금제를 확인
할 수 있으며 해당 요금제로 이동가능

02

회원 정보는 비밀번호를 입력해 비밀
번호가 맞다면 정보를 볼 수 있음

03

확인된 정보 중 비밀번호는 재설정이
가능하며 회원탈퇴도 가능함



트러블 슈팅 - 요금제 카드 단일 컴포넌트화

문제 인식

- 목적: 두 카드(작은 카드/큰 카드)를 하나의 컴포넌트로 통합해 재사용성·유지보수성 향상
- 실제 이슈: 레이아웃 구조가 상이하여 조건 분기남발 → 가독성 저하, 스타일 충돌

원인 분석

- 정보 계층이 다름
- DOM 트리 깊이/순서/간격이 달라 단일 템플릿으로 커버 어려움

결론

- 레이아웃은 분리 유지
- 중복되는 요소만 모듈화(별점, 가격 블록, 공통 컨테이너)
- 레이아웃 자유도 유지 + 중복 코드 감소(내부 요소 공유)

5G 프리미어 플러스 >
★ 3.0 (15)

데이터 무제한
테더링+쉐어링 100GB

월정액

105,000원

약정할인가

73,500원

★ 3.0 (15)
5G 프리미어 에센셜 >

데이터 무제한
테더링+쉐어링 70GB

월 85,000원
약정 할인 시 월 58,500원

비교하기

변경하기

프로젝트 경험 & 느낀점

- 아쉬운 점
 - → 주요 요구사항보다 UI/UX 디테일에만 집중한 점
- 배운 점
 - → 디테일을 신경 쓸수록 사용자 경험 개선에 효과적임을 체감
- 협업 경험 ①
 - → GitHub PR 템플릿으로 작업 상황을 투명하게 공유
- 협업 경험 ②
 - → 매일 아침 회의로 진행 상황 공유 → 일정 관리 용이
- 성과
 - → 팀원 간 원활한 소통 덕분에 프로젝트를 즐겁게 진행

PROJECT 03

지도 안의 중요한 혜택

지중해

지중해는 통신사 멤버십 제휴처 안내 지도 서비스로
혜택을 잘 모르는 사용자가 현재 위치를 기반으로
주변 제휴처와 혜택을 추천받고, 동네 사람들과 정보를 공유하며,
게이미피케이션 요소로 사용자의 접근성을 높이도록 설계한 서비스입니다.



지도페이지 - 검색 및 필터링 기능

- 검색/필터링 지원: 카테고리, 키워드, 혜택 유형 등 조건별 매장 검색 가능
- 빠른 탐색: 사용자가 원하는 매장을 즉시 찾을 수 있도록 직관적 인터페이스 제공
- 주변/전체 구분: 내 주변 매장 / 전체 검색 분리 → 정확한 정보 제공
- 실시간 반영: 조건 선택 시 결과가 즉시 지도에 반영되어 사용자 경험 향상



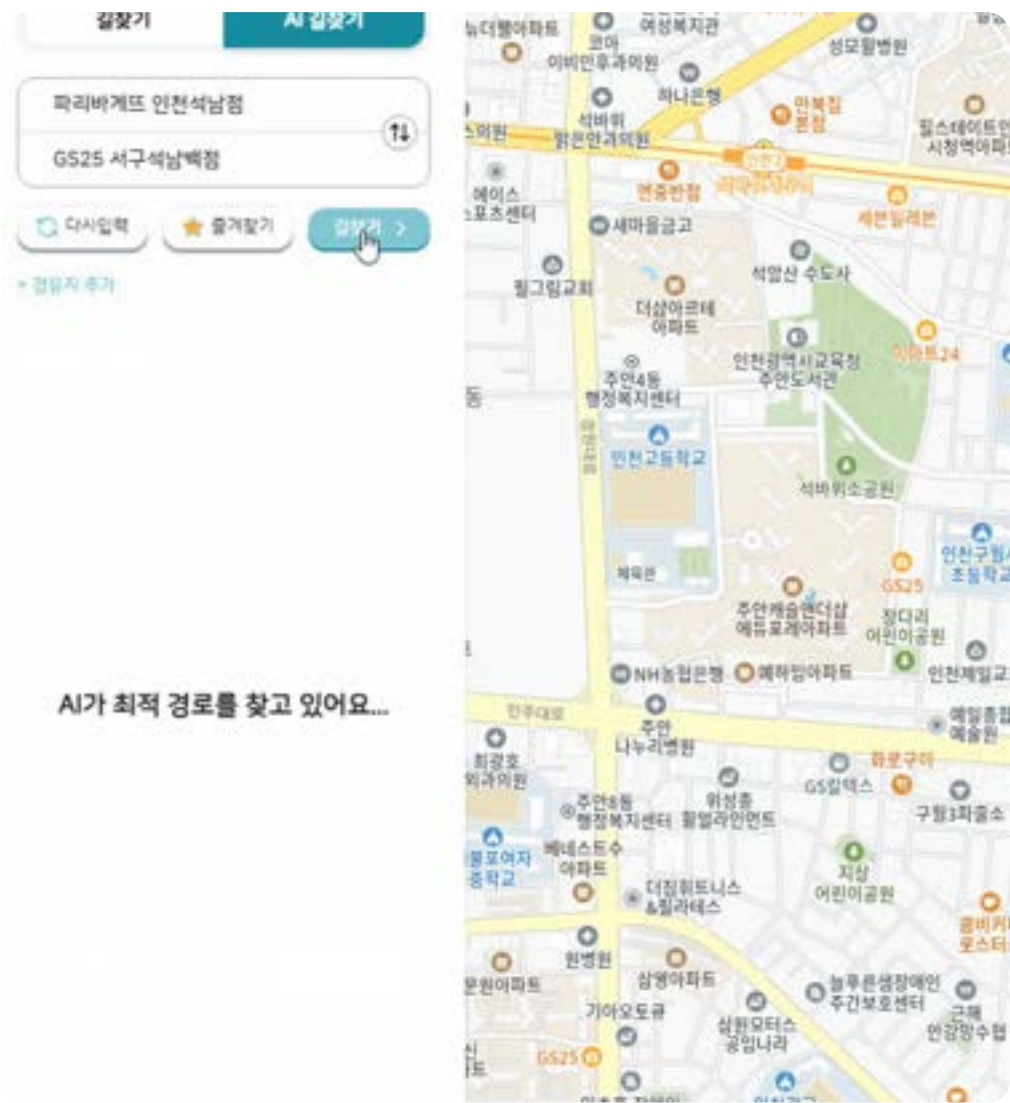
지도페이지 - 개인화된 정보 제공

- 맞춤 정보 제공: 즐겨찾기한 제휴처, 경로, 상세 정보 등을 한눈에 확인 가능
- Hover/Overlay 활용: 마우스를 올리면 간단 정보 제공, 클릭 시 상세 혜택·도감 현황 확인
- 실시간 현황 체크: 제휴처 이용 시 브랜드별 기록(도감처럼 별도 표시) → 사용자 경험 강화
- 차별화 포인트: 단순 지도 표시가 아니라, 이용 이력과 혜택을 개인 맞춤형으로 제공



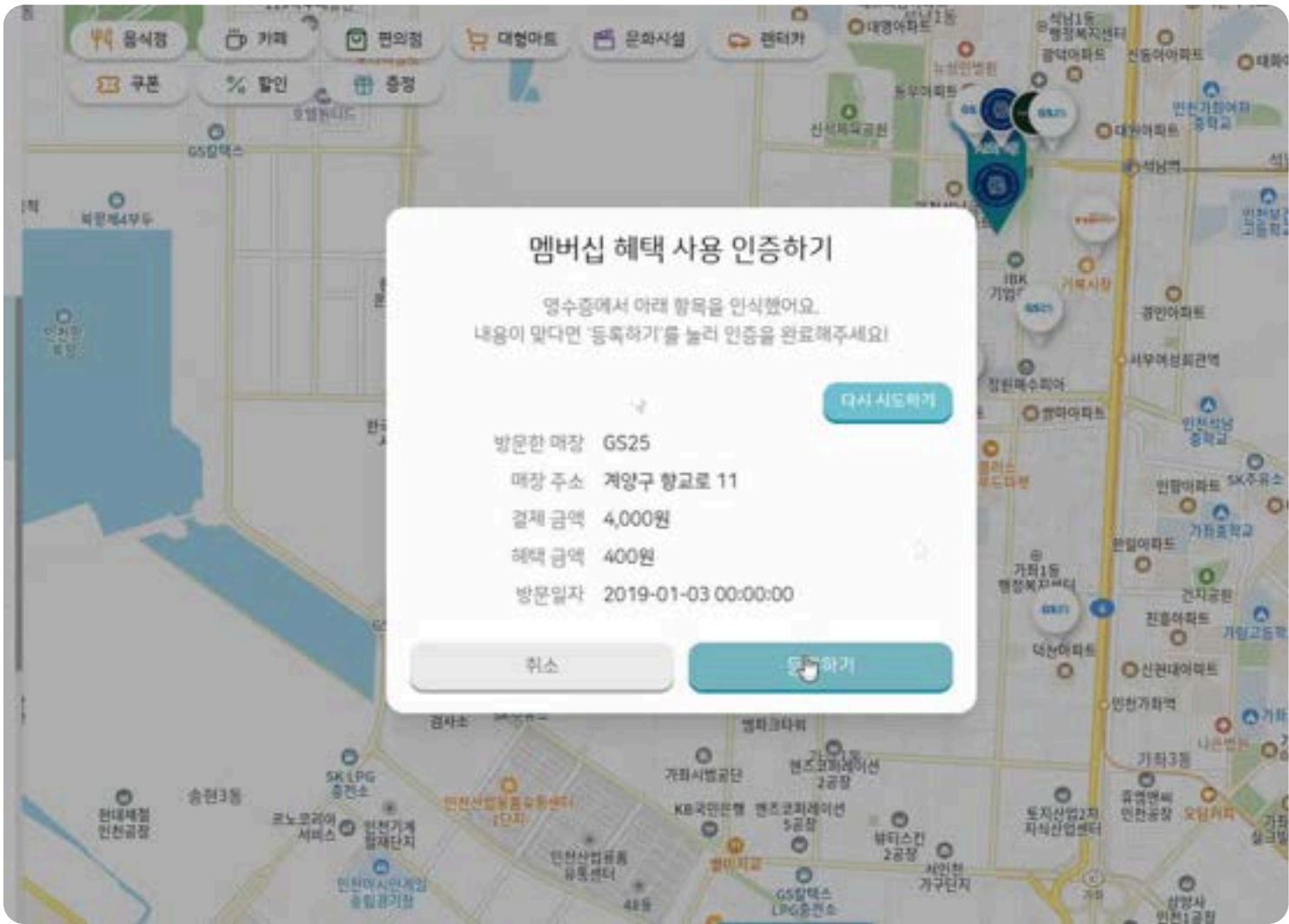
지도페이지 - AI 기반 길찾기 기능

- AI 추천 경로: 사용자 이용 패턴(시간대·이용 내역)을 분석해 최적 경로와 제휴처 추천 제공
- 사용자 맞춤 탐색: 출발지·경유지·도착지를 직접 선택해 개인화된 경로 탐색 가능
- 실시간 교통 반영: 카카오 모빌리티 API 연동으로 실시간 교통 상황 즉시 지도 반영
- 가이드 제공: 제휴처 경유가 포함된 경로를 상세 안내
→ 사용자 맞춤형 길찾기 경험 강화
- 시각화: Polyline으로 실제 지도 위에 경로를 직관적으로 표현



지도페이지 - OCR 기능

- 자동 인증: 사용자가 혜택 영수증을 업로드하면 Google Vision OCR이 텍스트를 추출 → Gemini API로 전달
- 혜택 판별: 추출된 데이터로 실제 멤버십 혜택이 적용된 영수증인지 자동 확인
- 데이터 처리: 금액·제휴처 정보가 JSON 형태로 프론트엔드에 전달
→ 등록 버튼 클릭 시 바로 등록
- 사용자 편의성: 수동 입력 없이 간단 업로드만으로 인증 완료
→ 빠르고 편리한 경험 제공



지도페이지 - AI 추천 제휴처 제공

- AI 기반 맞춤 추천: OpenAI API를 활용해 사용자의 이용 패턴 + 현재 위치를 분석 → 최적 제휴처 자동 추천
- 구분 표시: AI 추천 제휴처는 일반 제휴처와 시각적으로 차별화하여 쉽게 확인 가능
- 검색 편의성: 사용자가 직접 검색하지 않아도 가장 적합한 매장을 바로 확인
- UX 개선 효과: 불필요한 탐색 과정 제거 → 사용자 편의성·만족도 크게 향상

근처 제휴처

전체 검색

Q

근처 제휴처를 검색해봐요!

X

AI의 픽! ⓘ



파리바게뜨 부산재송점

부산 해운대구 재반로 78-1

출발

도착

☆



반올림피자 부산센텀점

부산 해운대구 재반로 71

출발

도착

☆



뚜레쥬르 연산LG점

부산 여제구 투곡로 29

출발

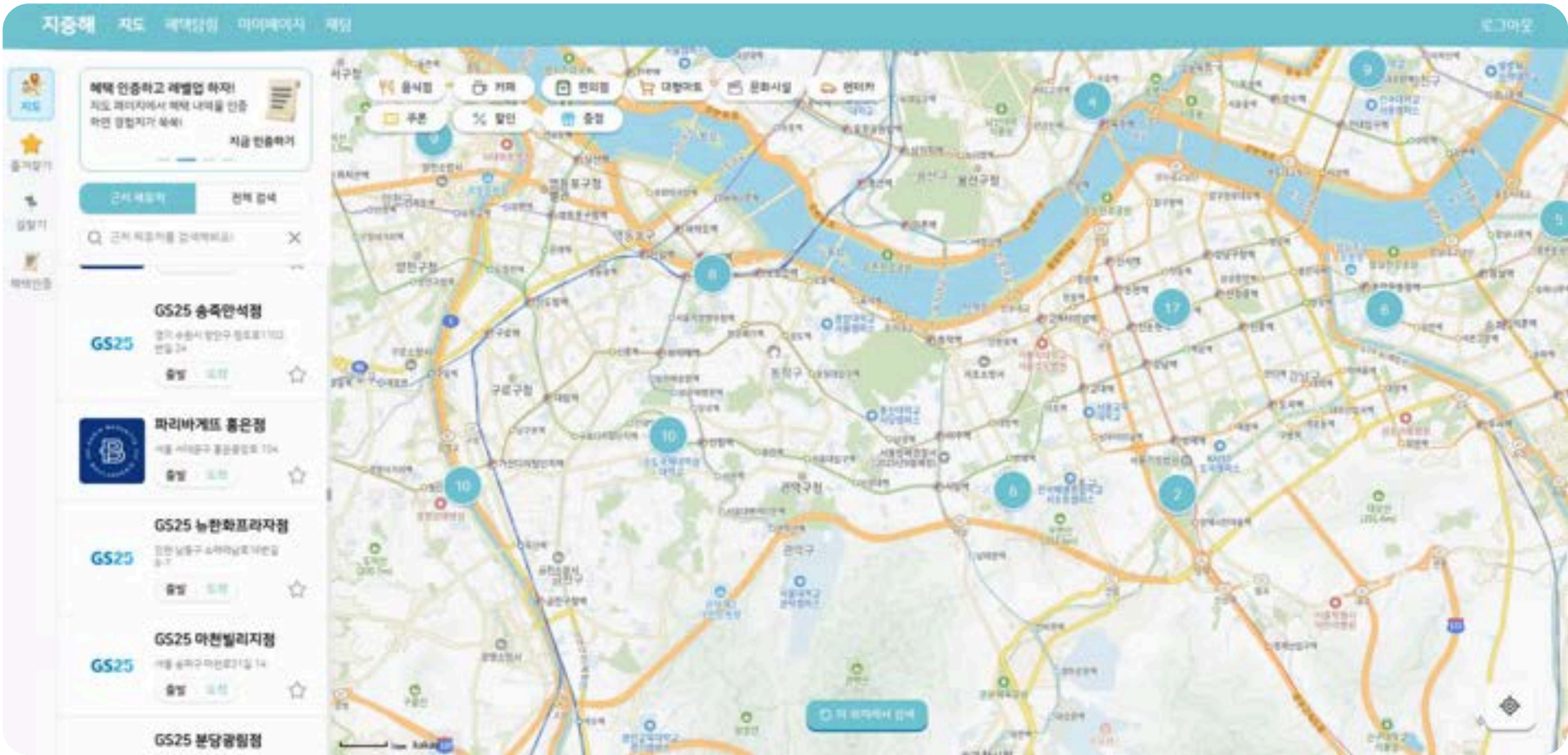
도착

☆



지도페이지 - 마커 최적화

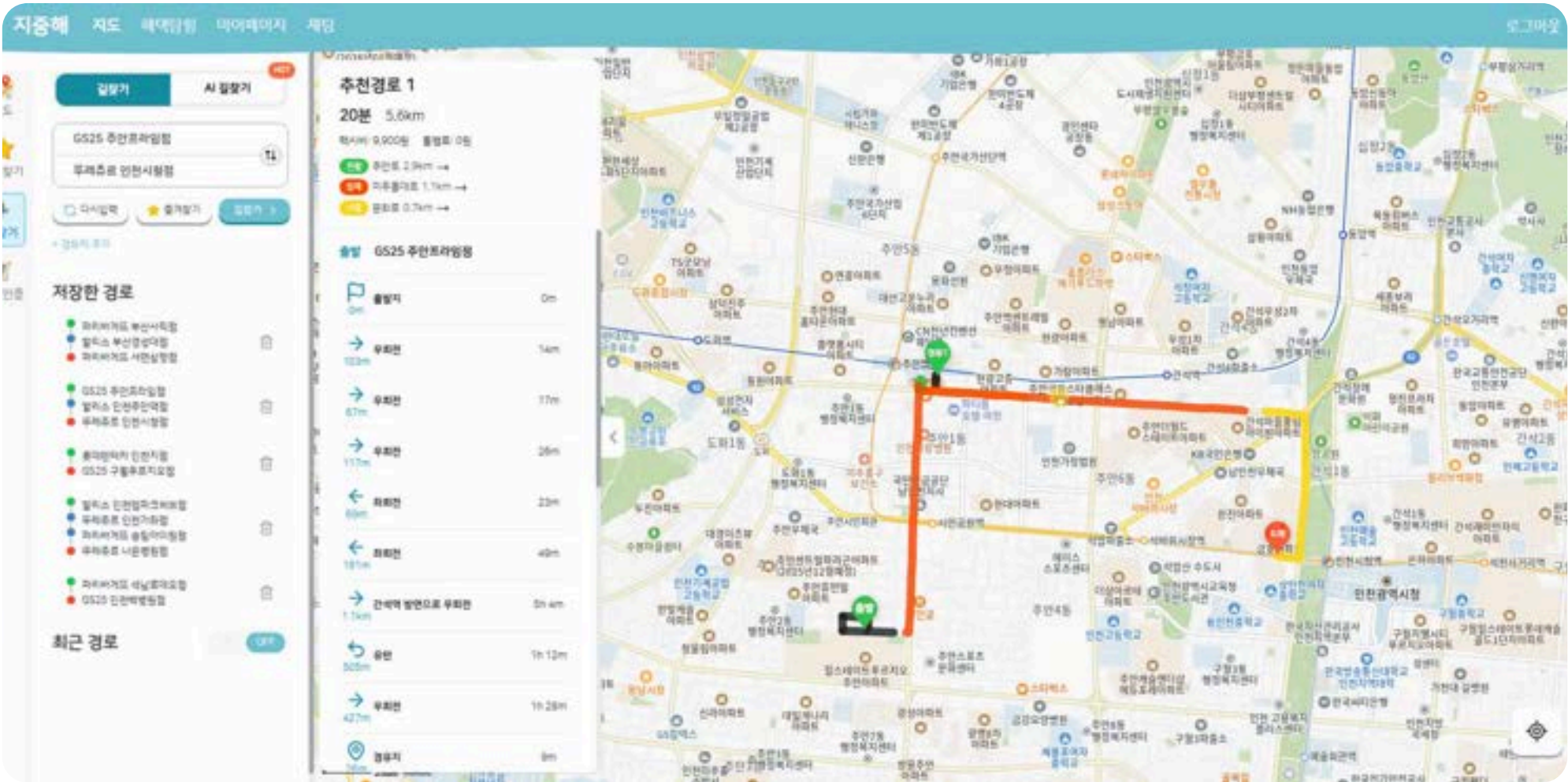
- 로드 속도 개선
 - 지도와 메뉴 상태에 따라 실시간 마커 관리
 - 지도 레벨별 마커 개수 제한 + 클러스터링 적용 → 성능 & 가독성 향상
- 효율적 정보 제공
 - 중심점과 가까운 매장을 우선 표시하여 직관적 정보 전달
 - 마커에는 핵심 정보만 추출해 불필요한 데이터 최소화
- 최적화 기법
 - useMemo 활용 → 불필요한 연산 제거
 - 성능 부하 감소 & 사용자 경험 개선



```
const Markers = useMemo(() => {
  if (!map) return [];
  if (panel.menu === '길찾기') return [];
  const level = map.getLevel?.() ?? 5;
  let candidates = stores;
  if (level <= 6) {
    candidates = stores
      .map((m) => ({
        marker: m,
        distance: getDistance(center, { lat: m.latitude, lng: m.longitude }),
      }))
      .sort((a, b) => a.distance - b.distance)
      .slice(0, level <= 2 ? 60 : level <= 4 ? 40 : 30)
      .map((item) => item.marker);
  } else {
    candidates = stores.slice(0, 100);
  }
  return candidates.map((m) => ({
    id: m.id,
    lat: m.latitude,
    lng: m.longitude,
    imageUrl: m.brandImageUrl ?? '',
    isRecommended: m.isRecommended ?? '',
  }));
}, [stores, map, panel.menu, center]);
```

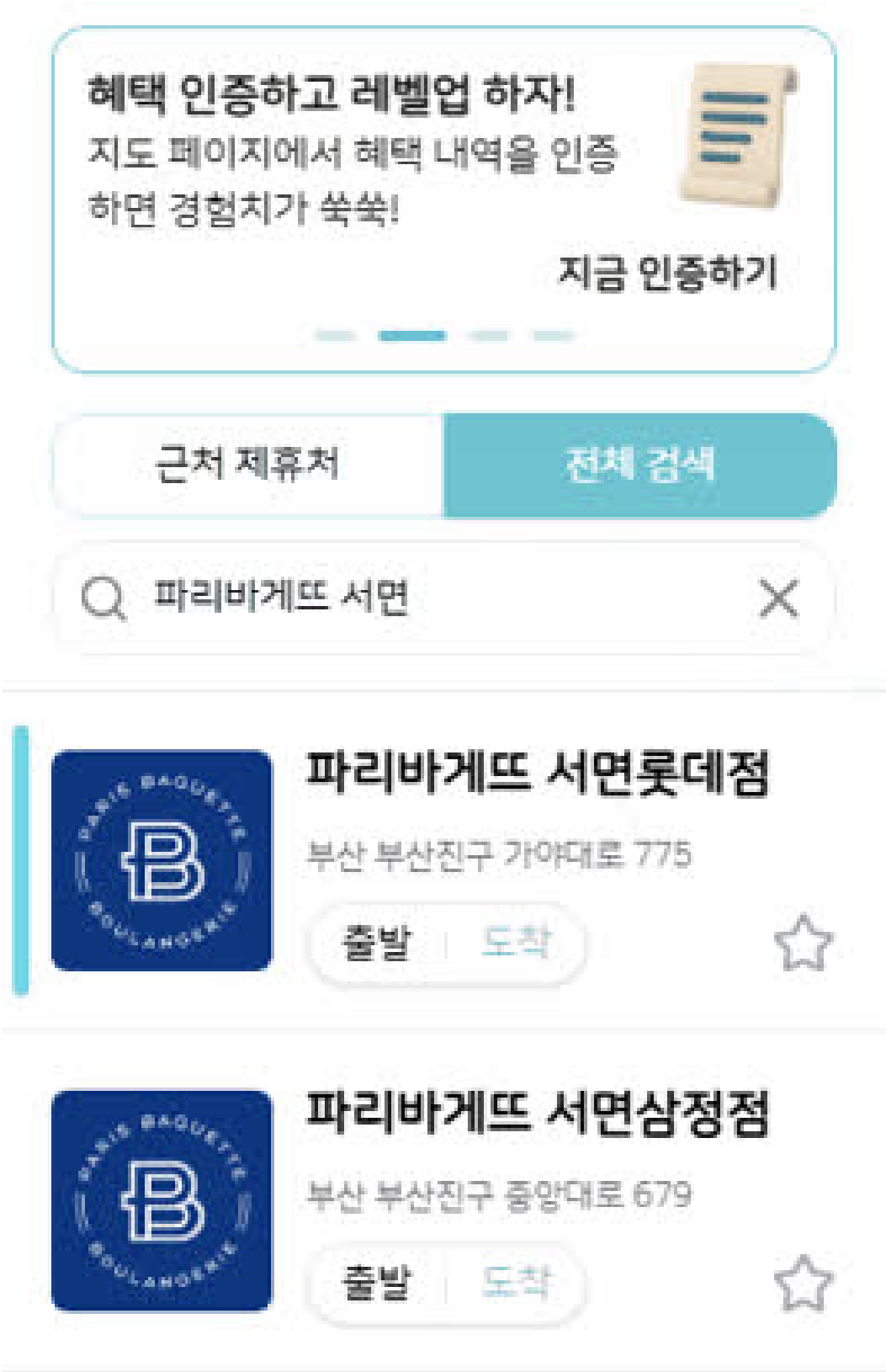

지도페이지 - 경로 고정 문제

- 문제 상황
 - 지도 중심 좌표만 바뀌어도 모든 Polyline(경로)이 불필요하게 리렌더링
 - 지도 이동 시 성능 저하 & 사용자 경험 악화
- 해결 방법
 - Polyline을 별도 컴포넌트로 분리
 - React.memo 적용 → 선택된 경로가 바뀌지 않으면 리렌더링 방지
- 효과
 - 지도 드래그·이동 시에도 경로선 고정 유지
 - 렌더링 성능 최적화 & 사용자 경험 개선



지도페이지 - 근처 제휴처만 볼 수 있는 문제 개선

- 문제 상황
 - 기존에는 현재 위치 기반 검색만 가능
 - 전국 단위 검색이 불가해 사용자가 원하는 특정 지역 제휴처 찾기 어려움
- 해결 방법
 - 신규 API 개발 → 전국 단위 제휴처 검색 지원
 - 근처 제휴처 모드와 전체 검색 모드를 완전히 분리
- 효과
 - 사용자는 원하는 지역 매장을 쉽게 검색 가능
 - 검색 편의성과 활용성 대폭 향상



프로젝트 경험 & 느낀점

- RESTful API 구현
 - 데이터 흐름을 명확히 정의, 프론트-백 간 통신 원활
- Swagger 활용 경험
 - API 문서화로 사용·테스트가 편리해짐
- 아이디어 제안 & 기능 확장
 - 진행 중 필요한 API/기능을 직접 제안하고 반영
- 기획 단계 소통 & 페어 프로그래밍
 - 백엔드와 협의하며 API 스펙 불일치 최소화, 빠른 피드백
- 아쉬운 점
 - 기능 개발보다 지도 최적화에 더 집중했으면 좋았을 것 같음

CONTACT ME!

Github : <https://github.com/nas7062>

Email : nas7062@naver.com

P.H : 010-9314-7062

velog : <https://velog.io/@10012/posts>

PortFolio: <https://portpolio-10012.vercel.app/>

THANK YOU

봐주셔서 감사합니다