

# 강민재

010-5519-7425 · towndev@naver.com · <https://minjaekang.dev/>

대학교를 휴학하고 창업에 도전하며 서비스를 직접 구현하기 위해 개발을 시작했습니다. 이후 오토바이 정비·라이딩 플랫폼 라이더타운과 협업 여행 플랫폼 저스트트립을 기획하고, React·Next.js·React Native 기반의 웹과 앱으로 개발해 출시·운영했습니다.

프론트엔드부터 백엔드, 인프라까지 직접 구축해 본 풀스택 경험을 바탕으로 시스템 전체의 데이터 흐름을 폭넓게 이해하며, 깊이 있는 소통과 집요한 실행력으로 조직과 서비스의 성장에 기여하겠습니다.

## 경력 신입

### 저스트트립 - AI 기반 크로스플랫폼 협업 여행 일정/기록 서비스

2026.03 - 2026.10 · 서비스 기획 · 프론트엔드 개발 · 백엔드 개발자 1명과 2인 협업

Next.js, React Native (Expo), TypeScript, TanStack Query, Redux Toolkit, Axios, SQLite, Azure Container App, MCP

- 삭제한 항목이 다시 나타나는 문제를 해결하고, 실행 취소가 가능한 삭제 흐름을 설계해 연속 삭제 시 재조회를 1회로 축소
  - 삭제 직후 응답에 서버 복제 지연으로 옛 데이터가 섞여 지운 항목이 되살아남
  - 4초 지연 커밋 + 대기·전송 중·최근 커밋 3단계로 삭제 ID를 추적해 응답에서 필터링, 백엔드 변경 없이 7개 도메인 적용
- 여행 상세 화면의 첫 로딩 용량을 절반으로 줄이고(890kB → 452kB, 49%↓), 지도 조작 시 불필요한 재렌더링 제거
  - next/dynamic만 쓰면 닫혀 있어도 청크를 받아, 처음 열릴 때 마운트·유지하는 LazyMount를 만들어 18곳에 적용
  - 첫 클릭 지연은 requestIdleCallback 프리페치로 보완, 일정 목록에만 React.memo + props 메모이제이션
- Document AI 기반 바우처, 영수증 자동입력을 백엔드와 설계하고 단일/일괄 폼 자동 전환을 구현
  - 입력 항목 항공 10·숙소 6개 → 파일 업로드 1회, 지출 10개 → 업로드 1회 + 3개 선택
  - 파싱 정확도 95%, 다구간 등록 시간 80% 단축, 가계부 입력 공수 90% 이상 절감
- 요청이 동시에 몰릴 때 생기는 검색 결과 뒤바뀔과 중복 요청 문제를 상황에 맞는 방식으로 해결
  - 장소 자동완성: SDK가 AbortController를 지원하지 않아 요청 순번으로 오래된 응답 무시
  - 토큰 갱신: Axios 인터셉터에서 refresh 1회만 실행, 동시 실패 요청은 대기 큐에서 재시도
- 비행기 모드나 네트워크가 없는 지역에서도 한 번 연 여행의 일정과 바우처를 볼 수 있는 오프라인 모드 구현
  - 일정은 Redux·React Query + AsyncStorage, 용량 큰 바우처 PDF·이미지는 기기 파일 + SQLite로 나눠 저장
  - NetInfo로 오프라인을 감지하면 저장된 데이터를 바로 표시, 재연결 시 최신화
- 여행 참가자만 사진·바우처를 볼 수 있도록 접근 권한을 백엔드와 설계하고 클라이언트에 적용
  - URL만 알면 누구나 열리던 구조 → 여행 단위 Signed Cookie로 참가자만 열람
  - 발급 중인 Promise 공유로 동시 요청에도 1회 발급, 만료 2분 전 자동 갱신

### 라이더타운 V2 - 라이딩 기록 서비스 피벗 & 서버리스 전환

2024.06 - 2024.12 · 공동창업 / 풀스택 개발 · 영업 담당 공동창업자와 2인 팀

React Native, Node.js, GCP, Firebase, react-native-background-geolocation, Mapbox

- AWS MSA(EC2·RDS·VPC)를 Firebase 서버리스로 전면 전환해 월 고정 인프라 비용 90% 이상 절감
  - 백엔드 개발자 이탈 후 혼자 운영할 수 있는 구조를 기준으로 모놀리식·Lambda·Firebase를 비교
  - Node.js 기반 Firebase로 전환해 프론트엔드부터 백엔드까지 JavaScript로 통일

- 네이티브 백그라운드 위치 수집과 속도·정지 상태 기반 동적 전송 주기로 화면이 꺼져도 끊기지 않는 주행 기록 구현, 경로 전송량 70% 이상 절감
  - 기본 Geolocation은 백그라운드에서 멈춰 네이티브 background-geolocation으로 교체
  - 속도·정지 상태별 동적 전송 주기 + 거리 샘플링(20/40/60m) + polyline 인코딩
- Realtime Database onDisconnect로 앱 강제 종료·배터리 방전 시에도 친구 지도에 "주행 중"으로 남는 유령 라이더 제거

## 라이더타운 V1 - 오토바이 정비 플랫폼 창업 · 라이딩 기록 및 그룹 음성 통화

2022.11 - 2023.12 · 공동창업 / 풀스택 개발 · 영업 담당·백엔드 개발자와 3인 창업팀

React, Next.js, React Native, Redux, Node.js, Express, WebSocket, Mediasoup v3, WebRTC, Java, Spring Boot, Redis

- React SPA를 Next.js SSR로 전환해 초기 번들 2.5MB → 550KB(약 78%↓), 웹뷰 첫 화면 3.5초 → 0.8초(약 77%↓) 단축
  - 라이브러리 교체·dynamic import·웹뷰 프리로딩으로는 첫 로딩이 줄지 않아 SSR로 전면 전환
  - 웹뷰에서는 postMessage로 네이티브 화면 이동, 브리지로 앱 다크모드 동기화
- Mediasoup v3 SFU 서버를 직접 구축해 그룹 음성 통화의 단말당 업로드 스트림을 3개 → 1개(4인 기준 약 67%↓)로 고정
  - WebRTC 메시 구조에서 3~4인부터 오디오 끊김과 발열 발생 → 재인코딩 없이 패킷만 전달하는 SFU 선택, 비정상 단절 시 서버 리소스 자동 회수
- 결제 중 슬롯을 5분간 원자적으로 선점하는 Redis 락 방식을 제안·설계해 인기 시간대 정비 예약 중복 문제 해결
  - 결제 앱 전환 시 웹소켓이 끊겨, 클라이언트 연결과 무관한 서버 측 TTL 락을 백엔드 개발자와 설계

## AI 활용 경험

- Azure Document Intelligence 연동으로 항공·숙소 바우처와 영수증을 자동 파싱해 가계부 입력 공수 90% 절감 (파싱 정확도 95%)
- 백엔드 Swagger 기반 38개 도구를 제공하는 원격 MCP 서버를 구축해 AI 에이전트가 실제 API로 검증하며 개발하는 환경 구성
- YOLOv8 합성 데이터셋 파이프라인과 모델 학습으로 mAP 92% 객체 인식 모델 개발
- 건강 설문 구조화와 LLM 기반 영양 성분 분석·추천 앱 비타 출시
- OpenAI API 감성 공감 프롬프트 설계로 사용자 감정에 맞춘 데일리 알림 메시지를 생성하는 앱 HappyWhale 출시 (유저 1,300여 명)

## 기술 역량

- Frontend: React, Next.js, TypeScript, Redux, React-query, Tailwind CSS
- Mobile: React Native, Expo, React Native Webview
- Backend: Java (Spring Boot), Node.js (Express), RESTful API 설계
- Database & Cache: MySQL, PostgreSQL, Redis, Firestore
- Cloud & DevOps: AWS, Docker, GCP, Firebase, Vercel, Git-flow 기반 협업

## 수상

청년창업사관학교 (창업성공패키지) 선정 및 수료

2024.03

오토바이 라이딩 DB를 통한 소모품 관리 및 정비 플랫폼 [1억 원 규모]

예비창업패키지 선정 및 수료

2023.05

오토바이 정비 견적 서비스 및 커뮤니티 플랫폼 [4,000만 원 규모]

## 학력

---

광운대학교

2018.03 - 2026.08 · 전자공학과 학사 졸업

## 링크

---

포트폴리오 · <https://minjaekang.dev/>