

웹·모바일 개발자

박주철

AI 문서 분석 플랫폼과 정산 ERP의 풀스택 개발, 물류·주식 업무 웹과 Android 연동 경험.

경력

트리포스(주) / 2026.02 ~ 현재

담당

웹 화면과 서버 API 개발, Android 앱 연동

jukrap628@gmail.comgithub.com/jukrapwww.linkedin.com/in/jukrap/jukrap.vercel.app/ko

주요 경험

01 AI 기반 문서·데이터 분석 플랫폼

Ragas 평가에서는 검증할 문장을 서버에서 고정하고, 모델이 반환한 평가 결과를 문장 ID로 대조하도록 구성했습니다. 긴 입력은 여러 묶음으로 나누고, 원문 인용·근거 일치·답변의 설명 범위를 각각 확인하도록 구성했습니다.

02 업무 정산 ERP·모바일 앱

수정 버전이 오래되면 저장을 거절하되 입력값은 화면에 남겼습니다. 수정 전 값·서버의 최신 값·현재 입력값을 비교해 선택한 뒤, 별도 저장으로 변경을 확정하도록 만들었습니다.

03 물류 운영 웹·출력 앱

화면·Excel 코드를 필요할 때 불러오도록 바꿔 초기 JavaScript 파일 크기를 약 74% 줄였습니다. 프린터 연결 오류와 라벨 잘림은 실기기 로그와 실제 출력물을 확인하며 수정했습니다.

04 AI Agent Playbook

개인 개발 도구. 실행기·스케줄러를 범위에서 제외하고, 설치 실패와 사용자 수정본 보존을 검증했습니다.

이어서 생활정보·개인정보 보호 앱과 사내 문서화 도구, 그 밖의 회사 업무와 개인·팀 프로젝트를 소개하겠습니다.

회사 업무 01

AI 기반 문서·데이터 분석 플랫폼

문서 검색과 질문 응답을 제공하는 Gemma 4 기반 플랫폼의 풀스택 개발을 맡았습니다. 검색 결과에 앞뒤 문맥을 추가하고, 답변 검증과 Ragas 평가 기능을 개선했습니다.

기간

2026.09 ~ 현재

담당

풀스택 개발

사용 기술 React, TypeScript, Tailwind CSS, Vite, TanStack Query, Zustand, Python, Django, MariaDB, SQLGlott, Gemma 4, Ragas, tokenizers, Vitest

해결할 문제

문서에서 근거를 찾아 답하는 AI 기능을 개발하면서, 답변뿐 아니라 평가 결과도 믿을 수 있어야 한다고 판단했습니다. 긴 답변은 평가 입력 한도를 넘었고, 평가 모델이 문장을 다시 쓰면서 검증 대상의 의미가 달라지는 문제가 있었습니다.

평가 대상은 고정하고, 모델의 판정 결과를 검증

Ragas 평가에서는 검증할 문장을 서버에서 고정하고, 모델이 반환한 평가 결과를 문장 ID로 대조하도록 구성했습니다. 긴 입력은 여러 묶음으로 나누고, 원문 인용·근거 일치·답변의 설명 범위를 각각 확인하도록 구성했습니다.

검증

실제 문서 답변에서 추출한 25개 주장을 5개 묶음으로 나누어 끝까지 평가했습니다. ID 누락·중복과 잘못된 응답 형식을 검사해, 형식이 깨진 판정이 정상 결과로 저장되지 않도록 확인했습니다.

모델 대기와 애플리케이션 병목을 분리해서 개선

응답이 느린 원인을 구분하려고 모델 응답을 고정한 상태에서 내부 처리를 측정했습니다. 반복되는 토큰 수 계산을 줄이고, 같은 요청에서 이미 처리한 문서 페이지를 다시 활용했습니다.

검증

워밍업 후 개선 전후 각각 20회를 측정한 결과, 내부 처리 중앙값이 2,969ms에서 672ms로 줄었습니다. 모델 호출 횟수는 유지했고, 실제 모델을 사용하는 근거 검증은 별도로 진행했습니다.

결과

긴 답변의 평가를 완료할 수 있게 했고, 반복 계산으로 발생하던 내부 지연을 줄였습니다. 답변은 먼저 보여 주고 Ragas 평가는 비동기로 이어가도록 바꿔 평가가 끝날 때까지 사용자가 기다리지 않게 했습니다.

문서 분석 처리 시간의 중앙값

2,969ms → 672ms

같은 모델 응답을 사용하는 시험에서 애플리케이션 처리 시간을 비교했습니다. 준비 실행 후 전후 각 20회 측정했으며, 모델·네트워크 대기 시간은 제외했습니다.

입력 한도를 넘던 긴 답변 평가 개선

25개 항목 평가 완료

실제 문서에 대한 Gemma 4 답변에서 검증할 내용을 추출해 5개 묶음으로 평가했습니다. 가장 긴 요청도 8,164토큰으로, 8,192토큰의 입력 한도를 넘지 않았습니다.

품질 평가를 기다리지 않는 답변 제공

답변을 먼저 제공하고 평가는 별도로 실행

Ragas 평가가 끝나기 전에 생성된 답변을 보여 줍니다. 사용자의 새 질문을 처리하는 동안에는 다음 평가 요청을 잠시 미룹니다.

회사 업무 02

업무 정산 ERP 플랫폼

차량·기사 정보부터 운송료·청구·정산까지 관리하는 ERP의 풀스택 개발을 맡았습니다. 여러 사람이 함께 수정하거나 Excel 자료를 일괄 등록해도 입력 내용과 확정된 정산 이력을 지키도록 구현했습니다.

기간

2026.07 ~ 2026.09

담당

풀스택 개발

사용 기술 React, TypeScript, Tailwind CSS, Vite, React Router, Zustand, Spring Boot, MyBatis, MariaDB, Vitest

해결할 문제

여러 담당자가 같은 정산 자료를 수정할 때, 늦게 저장한 내용이 다른 사람의 변경을 덮어쓰면 안 됩니다. 동시에 저장을 거절했다는 이유로 사용자가 작성한 내용까지 잃어서는 안 된다고 판단했습니다.

충돌은 서버에서 막고, 어떤 값을 살릴지는 사용자가 결정

수정 버전이 오래되면 저장을 거절하되 입력값은 화면에 남겼습니다. 수정 전 값·서버의 최신 값·현재 입력값을 비교해 선택한 뒤, 별도 저장으로 변경을 확정하도록 만들었습니다.

검증

개발 환경에서 별도 API 요청으로 먼저 값을 변경한 후, 브라우저의 저장 요청이 409 응답을 받아도 입력값과 DB의 최신 값이 유지되는지 확인했습니다. 이후 선택한 값을 다시 저장하고 DB와 변경 이력을 대조했습니다.

Excel 검증 결과를 저장 시점까지 유지

검증한 데이터를 서버에 보관하고 저장 직전에 권한과 참조 데이터를 다시 확인했습니다. 중간 오류는 전체 트랜잭션을 되돌리고, 같은 저장 요청을 다시 보내면 기존 처리 결과를 반환하도록 구성했습니다.

검증

격리한 MariaDB 환경에서 중간 실패·재요청·동시 요청을 재현해 부분 저장과 중복 반영이 발생하지 않는지 확인했습니다. 검증 화면에서 통과한 데이터가 실제 저장 과정에서도 같은 기준을 따르는지 점검했습니다.

결과

현재 배차 자료와 확정된 정산 이력을 구분하고, 수정 충돌과 일괄 저장 실패를 복구할 수 있는 흐름을 구현했습니다. 저장 성공 여부뿐 아니라 사용자의 입력과 변경 이력이 함께 유지되는지 확인했습니다.

회사 업무 02 / Mobile

ERP 연계 모바일 업무 앱

ERP와 연결되는 모바일 업무 앱에서 채팅 시스템을 제외한 대부분의 개발을 맡았습니다. React 업무 화면과 Spring Boot API를 구현하고, Flutter 앱의 알림·파일 다운로드와 연결했습니다.

기간

2026.07 ~ 2026.09

담당

풀스택 개발 · 채팅 시스템 제외

사용 기술 React, TypeScript, TanStack Query, TanStack Router, Spring Boot, MariaDB, Flutter, Dart, Firebase Cloud Messaging

해결할 문제

정산 ERP와 연결되는 모바일 앱에서는 알림이 늦게 도착하거나 같은 요청이 반복되어도 최신 상태가 유지되어야 했습니다. 웹에서 사용하던 PDF 다운로드 방식도 모바일 실행 환경에 맞게 바꿔야 했습니다.

알림 내용과 읽음 상태를 분리해 동시 요청에 대응

알림 버전과 삭제 상태를 비교해 늦게 도착한 생성 알림이 삭제된 항목을 되살리지 못하게 했습니다. 읽음 정보는 별도로 관리하고, 먼저 갱신한 뒤 없으면 등록하던 처리를 한 번의 UPSERT로 바꿔 최초 확인 시각을 유지했습니다.

검증

격리한 MariaDB에서 읽음 처리의 데드락을 재현한 뒤 동시 요청을 다시 실행했습니다. 알림 순서를 뒤집거나 같은 알림을 반복해서 보내도 최신 내용과 읽음 정보가 유지되는지 확인했습니다.

파일 다운로드 권한을 모바일에 맞게 전달

브라우저 안에서만 유효한 blob URL 대신 사용자가 다운로드할 때 짧게 유효한 서명 URL을 발급했습니다. 발급과 실제 다운로드 시점에 접근 권한을 확인하고, 로그인 토큰을 파일 주소에 넣지 않았습니다.

검증

Chrome에서 내려받은 PDF와 원본의 해시를 비교하고 네이티브 브리지에 전달되는 값을 확인했습니다. Android 에뮬레이터에서는 알림을 누른 뒤 공지 목록을 거쳐 공지 본문을 여는 흐름을 점검했습니다.

결과

문의 처리와 알림, PDF 다운로드를 ERP와 연결했습니다. 채팅 시스템 자체는 담당 범위에서 제외하고, 모바일 화면과 API 연계 및 데이터 처리 대부분을 개발했습니다.

회사 업무 03 / Web

물류 운영 웹

조회·예약·Excel 등록·라벨 출력을 지원하는 물류 웹의 프론트엔드 전체를 개발했습니다. 서버 API와 PC 프린터를 연동하고, Android 앱에서도 같은 업무 화면으로 라벨을 출력하도록 연결했습니다.

기간

2026.04 ~ 2026.06

담당

프론트엔드 전체 개발

사용 기술 React, TypeScript, Vite, React Router, TanStack Query, Zustand, XLSX, Vitest

해결할 문제

운영 웹은 시작할 때 당장 필요하지 않은 화면과 Excel 코드까지 함께 불러왔습니다. 첫 화면에 필요한 코드와 사용자가 기능을 실행할 때 필요한 코드를 구분해 초기 다운로드 크기를 줄이고자 했습니다.

번들 분석으로 불필요한 초기 코드를 찾아 분리

번들 분석으로 첫 화면에서 쓰지 않는 코드의 비중을 확인했습니다. 페이지 코드는 해당 화면에 들어갈 때, Excel 라이브러리는 업로드·다운로드를 실행할 때 불러오도록 변경했습니다.

검증

일반 프로덕션 빌드에서 초기 JavaScript가 2,405.50kB에서 616.60kB로, gzip 기준 815.10kB에서 204.38kB로 줄어든 것을 확인했습니다. 변경 후에도 페이지 이동·데이터 조회·Excel 처리·PC 라벨 출력이 이어지는지 확인했습니다.

미리보기와 실제 출력 결과를 따로 확인

화면에서 보이는 배치만으로 인쇄 품질을 판단하지 않았습니다. 긴 주소를 실제 폭에 맞춰 나누고 글자 크기와 간격을 조절했습니다.

검증

실제 용지에 출력해 긴 주소가 잘리거나 다른 항목과 겹치는 부분을 확인하고 보정했습니다.

결과

첫 진입에 필요한 코드의 크기를 줄이고, 운영 화면에서 선택한 데이터를 실제 라벨로 출력하는 흐름을 정비했습니다.

초기 JavaScript 파일 크기

2,405.50 → 616.60 kB

2026.05 빌드 산출물 기준 약 74% 감소

gzip 압축 크기

815.10 → 204.38 kB

초기 JavaScript 파일의 압축 크기 약 75% 감소

회사 업무 03 / Android

물류 라벨 출력 앱

물류 웹의 라벨 출력 요청을 Android 앱의 네이티브 모듈과 Bluetooth 프린터에 연결했습니다. 사용자가 장비를 선택하고 연결 상태를 확인한 뒤 출력할 수 있도록 구현했습니다.

기간

2026.04 ~ 2026.06

담당

Android 앱 개발

사용 기술 Expo, React Native, Expo Router, WebView, Android 네이티브 모듈, TypeScript, Bluetooth

해결할 문제

Android에서 프린터를 선택할 수 있는데도 연결이 실패했습니다. 또 SDK가 출력 요청을 받았다는 사실만으로 실제 인쇄가 끝났다고 표시하면 사용자에게 잘못된 상태를 전달하게 됩니다.

연결 실패를 SDK 내부 동작까지 추적

실기기 로그를 따라가 연결 과정에서 SDK가 기기 검색을 중단하는 동작을 확인했습니다. 이 동작에도 검색 권한이 필요해, 연결 권한뿐 아니라 검색 권한까지 점검하도록 수정했습니다.

검증

Android 16 실기기에서 권한 요청, 프린터 연결, 실제 라벨 출력을 차례로 확인했습니다.

확인할 수 있는 출력 상태만 사용자에게 전달

SDK의 성공 응답은 요청 접수로 해석하고 화면에 “출력 요청 전송”으로 표시했습니다. 실제 용지가 나왔는지 확인할 수 없는 상태에서 인쇄 완료로 단정하지 않았습니다.

검증

SDK 응답과 실물 출력 결과를 각각 확인해 앱이 표시하는 상태의 의미를 맞췄습니다.

결과

React Native 화면에서 Android 네이티브 SDK를 호출해 라벨을 출력하고, 권한과 연결 실패를 구분해 처리했습니다.

라벨 출력 연결

01

WebView

라벨 데이터·출력 요청

02

Android

브리지·권한·장비 연결

03

Bluetooth

프린터 SDK·라벨 출력

프로젝트 01

AI Agent Playbook

AI 코딩 도구가 작업 세션이 바뀌어도 프로젝트 규칙과 이전 기록을 참고하도록 만든 개발 도구입니다. 필요한 스킬을 골라 설치하고, 기존 파일을 보존하면서 구성을 바꾸거나 복구할 수 있습니다.

기간

2026.06 ~ 현재

담당

1인 개발

사용 기술 JavaScript, Node.js, MCP SDK, Zod, GitHub Actions

해결할 문제

AI 코딩 도구를 바꾸거나 새 세션을 시작할 때도 프로젝트의 결정과 작업 상태를 이어가고 싶었습니다. 이를 위해 무엇을 직접 만들고, 무엇을 기존 도구에 맡길지 먼저 정했습니다.

실행기·스케줄러를 덜어 내고 프로젝트 기록에 집중

기존 코딩 도구와 역할이 겹치는 실행기와 스케줄러는 제품 범위에서 제외했습니다. 패키지 설치, 프로젝트 초기화, 스킬 설치, MCP 연결을 분리해 필요한 기능만 적용할 수 있게 했습니다.

검증

CLI와 MCP가 같은 프로젝트 기록을 다루는지, 각 설치·초기화 경로가 독립적으로 동작하는지 확인하고 Windows·Ubuntu CI에서 검증했습니다.

자동 갱신보다 사용자가 수정한 파일의 보존을 우선

파일의 소유 정보와 해시로 수정 여부를 판단했습니다. 새 스킬의 유효성을 모두 확인한 뒤 기존 파일을 정리하고, 삭제 직전에도 변경 여부를 다시 확인하도록 했습니다. 적용 기록과 백업으로 복구 경로도 마련했습니다.

검증

설치 중 일부 단계의 실패, 파일 충돌, 검사 후 발생한 파일 수정, 복원 시나리오를 테스트했습니다. 실패한 설치가 기존 파일이나 사용자의 수정본을 지우지 않는지 확인했습니다.

결과

프로젝트 기록 구조와 CLI·MCP 서버를 npm 패키지로 배포하고, 소스와 사용 문서를 공개했습니다. 기능을 늘리는 것뿐 아니라 도구의 책임 범위와 실패 시 보존해야 할 데이터를 설계한 사례입니다.

GitHub

github.com/jukrap/ai-agent-playbook

npm

www.npmjs.com/package/ai-agent-playbook

회사 업무 04

주식 업무 관리 웹

계좌·종목·주문을 조회하고 관리하는 React 프론트엔드 전체를 구축했습니다. MSW 모의 API로 화면을 개발하면서 반복되는 테이블과 조회 상태 처리를 공통화했습니다.

기간

2026.02 ~ 2026.03

담당

프론트엔드 전체 개발

사용 기술 React, TypeScript, Vite, React Router, TanStack Query, Zustand, MSW

함께 쓰는 테이블·필터·모달

계좌·종목·주문 화면에 반복되는 목록·검색·상세·등록·수정 UI를 공통 컴포넌트로 구성했습니다. 테이블에는 열 고정·너비 조절·재정렬을 넣고, MSW 모의 API로 조회와 입력 동작을 구현했습니다.

주문내역의 조회 조건 적용

선택 중인 계좌·종목과 목록 조회에 적용한 조건을 따로 관리했습니다. 필터를 바꾸는 중에는 기존 목록을 유지하고, 조회 버튼을 누르면 제출한 조건으로 TanStack Query의 조회 키를 갱신하도록 구현했습니다.

재조회와 더보기

같은 조건으로 다시 조회

기존 조회를 초기화해 첫 페이지부터 다시 불러오도록 처리.

더보기 요청 중복 방지

다음 페이지가 없거나 이미 요청 중일 때 추가 요청을 보내지 않도록 처리.

회사 업무 05

생활정보 앱의 날씨 기능

기존 Android 앱에 공공 데이터 기반 날씨 기능을 추가했습니다. WebView 화면과 서버의 API 연동·캐시 처리를 맡았습니다.

기간

2026.06 ~ 2026.07

담당

프론트엔드·백엔드 기능 개발

사용 기술 Spring MVC, JSP, jQuery, Java, Android

여러 API로 나뉜 날씨 정보

날씨와 대기질, 특보는 조회 지역과 갱신 주기가 달랐습니다. 화면을 열 때마다 모든 정보를 다시 요청하면 같은 지역의 호출이 반복되고, 느린 응답 하나 때문에 다른 정보까지 늦게 표시될 수 있었습니다.

지역과 정보 종류에 따른 캐시

조회한 지역의 정보 재사용

같은 지역의 정보를 반복해서 요청하지 않도록 서버 응답을 지역과 정보 종류별로 캐시했습니다. 각 정보의 갱신 주기에 맞춰 보관 시간을 정했습니다.

동시에 들어온 요청 묶기

같은 지역·정보를 조회하는 요청이 동시에 들어오면 진행 중인 요청의 결과를 함께 사용하도록 했습니다. 외부 API에 같은 요청을 중복해서 보내지 않도록 처리했습니다.

핵심 정보부터 표시

현재 날씨와 핵심 예보를 먼저 표시하고, 대기질·특보 등 부가 정보는 응답이 도착하면 채우도록 나눴습니다. 부가 정보가 늦거나 조회에 실패해도 이미 받은 날씨 정보는 볼 수 있습니다.

오래된 정보를 보여 줄 때의 기준

외부 조회가 실패하면 허용 범위 안에서 이전 값을 표시했습니다. 다만 만료된 ‘특보 없음’은 현재도 특보가 없다는 뜻이 아니므로 조회 불가로 구분하고, 이전에 조회한 특보에는 최근 조회 기준임을 표시했습니다.

회사 업무 06

개인정보 보호 앱

기존 Android 앱에 이메일 유출 여부와 딥페이크 이미지를 검사하는 기능을 추가했습니다. 검사 화면과 서버 API, Android 이미지 선택 기능을 연결해 사용자가 자료를 입력하고 결과를 확인할 수 있도록 했습니다.

기간

2026.06

담당

프론트엔드·백엔드 기능 개발

사용 기술 Android, Cordova, Spring MVC, jQuery, Java

외부 검사 API를 앱 기능으로 연결

사용자가 이메일이나 이미지를 입력하면 서버에서 외부 검사 API를 호출하고, 결과와 오류를 화면에 맞게 반환하도록 만들었습니다. API 인증 정보는 서버에서 관리하고 앱에는 검사에 필요한 결과와 상태를 전달했습니다.

이미지 선택부터 검사 요청까지

입력 검사와 미리보기

선택한 이미지의 파일 형식·용량·해상도를 검사하고, 전송 전에 이미지를 확인할 수 있도록 미리보기를 구현했습니다.

Android 파일 선택 연결

WebView에서 이미지나 파일 선택을 요청하면 Android 선택 창을 열고, 선택한 파일을 다시 WebView로 전달하도록 연결했습니다.

이미지를 다시 선택했을 때

파일을 읽는 도중 다른 이미지를 선택하면 이전 작업이 더 늦게 끝날 수 있습니다. 파일 읽기마다 구분 값을 두고 완료 시 현재 선택과 비교해, 이전 이미지의 결과가 새 미리보기를 덮지 않도록 처리했습니다.

회사 업무 07

AI 보조 프로젝트 문서화 웹

저장소와 프로젝트 착수 자료를 바탕으로 문서 초안을 만들고 검토하는 사내 웹 도구를 개발했습니다. 자료 수집, 요구사항 검토, 표 편집과 내보내기까지 이어지는 화면과 서버 API를 구현했습니다.

기간

2026.04

담당

프론트엔드·백엔드 개발

사용 기술 Node.js, TypeScript, React, AI API, xlsx, Vitest

자료 확인 후 문서 작성

자료를 바로 최종 문서로 만들기보다, 요구사항 후보와 확인할 질문을 먼저 검토하도록 화면을 나눴습니다. 요구사항 문서가 준비된 뒤 기능 정의서를, 선행 문서가 준비된 뒤 화면 설계서를 생성할 수 있도록 진입 조건을 구성했습니다.

표에서 검토하고 필요한 부분만 수정

시트별 문서 편집

요구사항·기능 정의서·화면 설계서를 시트별 표에서 검토하고 수정할 수 있도록 했습니다. 내보내기 파일에도 사용자가 수정한 표의 내용을 반영했습니다.

선택한 내용의 AI 수정

수정할 시트·셀과 주변 내용을 AI에 함께 전달했습니다. 수정할 범위와 유지할 내용을 구분해, 문서 전체를 다시 생성하지 않고 선택한 부분을 고칠 수 있도록 했습니다.

일부 문서 생성이 실패했을 때

문서별 생성 상태를 구분해 일부 작업이 실패해도 성공한 문서는 검토할 수 있도록 했습니다. 전체를 다시 생성하지 않고 필요한 문서를 선택해 다시 작성하고 내보내도록 구성했습니다.

회사 업무

그 밖에 맡은 업무

기존 서비스에 기능을 추가하고, Android 버전과 현장 사용 조건에 맞춰 수정한 작업입니다.

08	회원·문의 관리 웹 및 앱 솔루션	기간 2026.09	플랫폼 Web / Android
핵심 구현 기존 웹과 Android 앱에 회원 검색·수정·중복 통합과 문의·첨부 기능을 추가했습니다. 서버에서 회원 역할과 작성자를 확인해 접근을 제한하고, 앱의 파일 선택과 이미지 미리보기를 연결했습니다. 기술 Spring MVC, Java, MyBatis, Vue 2, MariaDB, Android, Kotlin			
09	택배 업무 앱	기간 2026.08	플랫폼 Web / Android
핵심 구현 지도 마커 선택과 처리 후 지도 복귀를 수정했습니다. 통신이 끊겨도 배달완료 요청과 사진을 앱에 보관하고, 재실행 후 요청을 복원하도록 했습니다. 재전송 전에는 서버의 완료 내역과 대조하도록 구현했습니다. 기술 React, TypeScript, React Native, Expo, Spring Boot, SQLite			
10	차트 미리보기 페이지	기간 2026.03 ~ 2026.04	플랫폼 Web
핵심 구현 개발 중이던 사이트에 차트 설정·데이터 연결·미리보기 기능을 구현했습니다. 차트별 데이터 항목을 연결하고, 불필요한 차트 재생성을 줄여 깜빡임과 스크롤 흔들림을 완화했습니다. 기술 React, TypeScript, Vite, Chart.js, WebGL/GLSL (OpenGL 계열), OGL (WebGL 라이브러리), MSW, Vitest			
11	물류 입출고 PDA 앱	기간 2026.03 ~ 2026.04	플랫폼 Android
핵심 구현 기존 PDA 앱에 입고 조회 조건과 수량 입력 화면을 추가하고 QR·바코드 스캔을 연결했습니다. 업무 모드별 코드 검사·중복 허용·수량 계산을 분리하고, 실기기 입력과 로그로 외부 검증 실패와 앱 내부 오류를 구분했습니다. 기술 Android Java, Gradle/AGP, Scanner SDK			
12	출고·재고 관리 레거시 앱	기간 2026.03	플랫폼 Android
핵심 구현 기존 Android 앱을 다시 빌드할 수 있도록 개발 도구의 호환 버전을 맞췄습니다. OS에 따라 달라지는 권한·파일 접근·뒤로가기 처리를 수정하고, 구형 OS 기기에서 동작을 확인했습니다. 기술 Android Java, Gradle/AGP, RxJava, FileProvider			

프로젝트 02

C. Donghae

동해선 이용객이 실시간 교통 정보와 주변 장소를 지도에서 확인할 수 있는 웹 서비스입니다.

기간

2024.10.04 ~ 2024.10.06

담당

3명 팀 / 프론트엔드 개발

사용 기술 TypeScript, React, Next.js, Tailwind CSS, Google Maps, Storybook, AWS, GitHub Actions, Docker

해커톤의 개발 범위

72시간 안에 열차·날씨·주변 장소처럼 출처가 다른 정보를 지도 한 화면에 모으고 모바일 조작도 구현해야 했습니다.

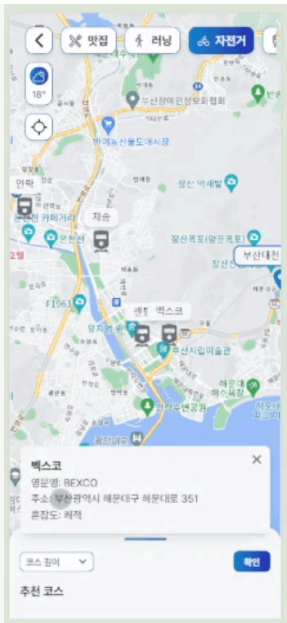
담당 구현

지도와 데이터 연결

백엔드 개발자 2명과 Swagger 명세를 기준으로 API를 연결했습니다. Google Maps 지도에서 교통 정보와 주변 장소를 확인할 수 있도록 화면을 구현했습니다.

모바일 지도 조작

지도를 보면서 장소 목록도 살펴볼 수 있도록 바텀 시트를 만들었습니다. 드래그한 거리와 속도에 따라 목록 영역이 펼쳐지거나 접히도록 구현했습니다.



수상

DIVE 2024 부산테크노파크원장상 수상(발제사 3등).

프로젝트 03

ShareBBy

취미 활동을 공유하고 함께할 사람을 찾는 앱입니다. Android 화면 대응과 게시물·댓글 등 커뮤니티 기능을 담당했습니다.

기간

2024.04 ~ 2024.06

담당

React Native 앱 개발·Firebase 연동

사용 기술 JavaScript, React Native, Firebase, Faster Image

Android 지원과 이미지 갱신

iOS에 맞춰 만든 화면을 Android에서도 사용할 수 있도록 고쳐야 했습니다. 게시물 이미지를 바꿔도 이전 이미지가 보이는 캐시 문제도 있었습니다.

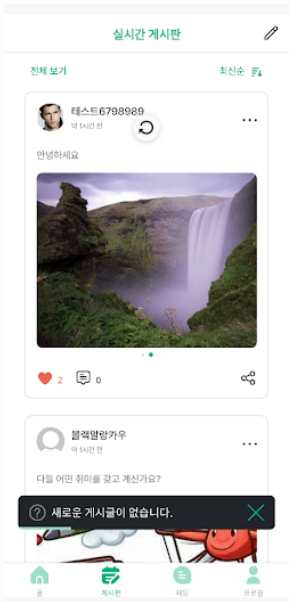
담당 구현

커뮤니티 기능

게시글과 댓글의 작성·조회·수정·삭제 기능을 구현했습니다. 위치 필터와 정렬을 추가하고, 목록을 당겨 새로고침하거나 스크롤해 다음 게시글을 불러올 수 있도록 했습니다.

데이터와 이미지 처리

Firebase에 저장할 데이터 관계를 ERD로 정리했습니다. 게시물 이미지를 바꿔도 이전 이미지가 보이는 문제를 재현한 뒤 이미지 라이브러리를 교체했습니다.



배포

Android 화면과 커뮤니티 기능을 개발했습니다. 팀은 2024년에 앱을 App Store에 배포했습니다.

GitHub

github.com/jukrap/archived-rn-ShareBBy

프로젝트 04

Itzip

개발자 취업 준비생을 위한 15명 팀 프로젝트입니다. 프론트엔드 팀장으로 블로그와 Markdown 편집 화면을 개발하고, 테스트·오류 모니터링·배포 환경을 구성했습니다.

기간

2024.07 ~ 2025.06

담당

프론트엔드 팀장 / DevOps

사용 기술 TypeScript, React, Next.js, Tailwind CSS, React Spring, Jotai, Jest, Prisma, Storybook, Postman, Sentry, AWS, Jenkins, Docker

편집 기능과 팀 개발 환경

편집 화면을 개발하면서 팀원들이 컴포넌트의 동작과 배포 후 오류를 함께 확인할 수 있어야 했습니다.

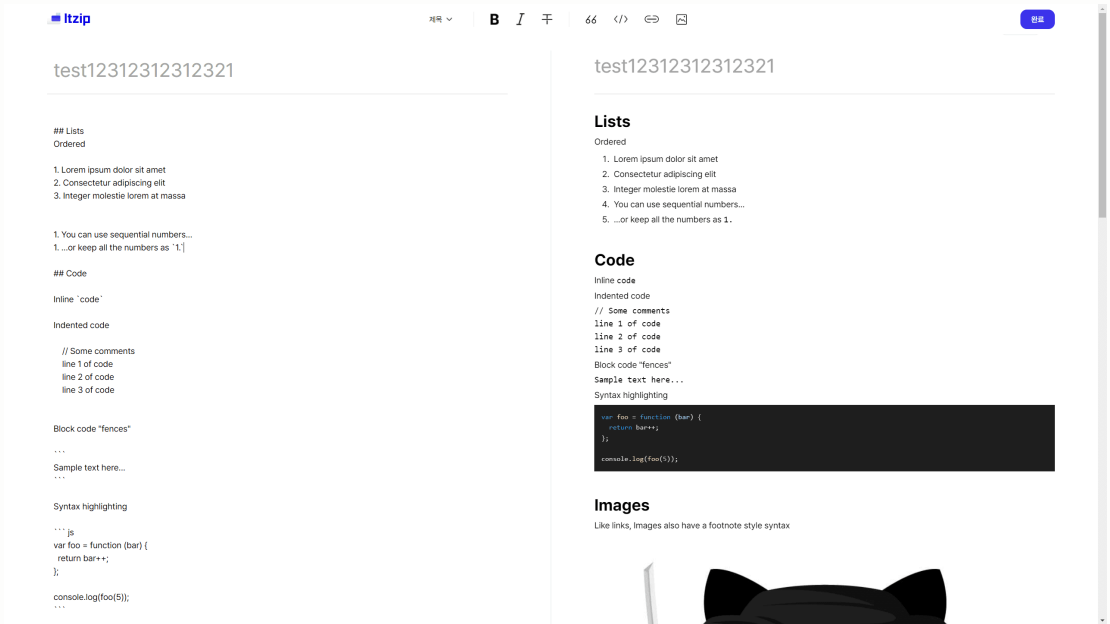
담당 구현

Markdown 편집과 미리보기

글을 작성하면서 결과를 바로 확인할 수 있도록 Markdown 편집기와 실시간 미리보기를 구현했습니다. 프로젝트 전용 문법을 지원하고, 무거운 게시글 컴포넌트는 dynamic import로 분리해 필요할 때 불러오도록 했습니다.

테스트와 오류 관측

Jest로 주요 동작을 테스트하고 Storybook에 컴포넌트의 상태와 사용법을 정리했습니다. 배포 후 발생한 오류는 Sentry에서 확인할 수 있도록 연결했습니다.



Markdown 편집과 미리보기

팀 개발에 적용

블로그와 편집 화면을 구현하고 팀이 테스트·컴포넌트 문서·오류 기록을 함께 확인할 수 있도록 구성했습니다. AWS 실행 환경과 Jenkins·Docker 기반 배포 환경도 구축했습니다.

GitHub

github.com/ITZipProject/itzip _ front

프로젝트 05

Posture Teacher

카메라 영상에서 몸의 주요 지점을 찾아 앉은 자세와 플랭크 자세를 판별하고, 자세를 유지한 시간을 기록하는 Android 앱입니다.

기간

2022.07 ~ 2023.05

담당

2명 팀 / Android 개발

사용 기술 Java, Android, Jetpack, MediaPipe, SQLite

카메라 분석과 화면 조작

카메라 프레임을 계속 분석해도 화면 조작이 멈추지 않아야 했습니다. MediaPipe를 Android 앱에서 사용할 수 있게 빌드하는 작업도 필요했습니다.

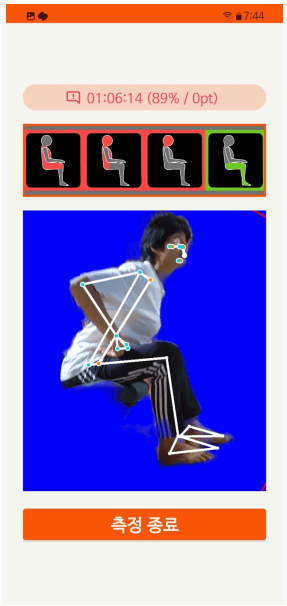
담당 구현

MediaPipe 통합과 자세 판별

Ubuntu에서 MediaPipe를 Android 라이브러리(AAR)로 빌드해 앱에 통합했습니다. 영상에서 찾은 신체 지점 사이의 각도와 길이로 자세를 판별하고, 해당 자세를 유지한 시간을 계산했습니다.

프레임 분석 분리

카메라 프레임을 분석하는 작업을 별도 스레드로 옮겨, 분석 중에도 사용자의 화면 조작을 처리하도록 했습니다.



자세 분석과 측정 기록

측정 결과

프로젝트 당시 OpenCV 기반 구현을 MediaPipe로 바꾼 뒤 초당 처리하는 카메라 프레임 수(FPS)가 이전 구현의 5~10배로 늘어난 것을 확인했습니다.

GitHub

github.com/jukrap/archived-Posture-Teacher

프로필

기술, 교육과 수상

기술

언어

TypeScript, JavaScript, Java, Python, Dart

모바일·하이브리드

React Native, Expo, Flutter, Android, Gradle

도구와 데이터

GitHub Actions, Jenkins, Docker, MariaDB, SQLite

웹

React, Next.js, Vite, TanStack Query, Zustand, Django

테스트·관측

Vitest, MSW, Storybook, Sentry, Ragas

교육·활동

경상국립대학교 컴퓨터과학과

2023.02 졸업

학점 3.64 / 4.5

프로그래머스 데브코스

2023.12 ~ 2024.05

Cloud Application Engineering 과정, React와 React Native

프로그래머스 데브코스 서브멘토

2024.05 ~ 2024.09

개발 정보 공유, 데일리 스크럼 참여와 프로젝트 점검 지원.

수상

DIVE 2024 글로벌 데이터 해커톤

2024.10

부산테크노파크원장상(발제사 3등)

경남소프트웨어 경진대회

2021.10

ESD HotDeal, 최우수상

더 보기

About

jukrap.vercel.app/ko/about

Work

jukrap.vercel.app/ko/work

Side Projects

jukrap.vercel.app/ko/projects