

전하성, Frontend Developer

Introduction

끊임없이 배움을 추구하며 “어제보다 오늘 더 나은 개발자”가 되고자 하는 4년차 프론트엔드 개발자 전하성입니다.

대학에 기술경영으로 입학했으나, 우연히 수강하게 된 객체지향 프로그래밍 수업에서 문제를 논리적으로 분석하고 코드로 구현하는 과정에 큰 성취감과 즐거움을 느껴 컴퓨터공학을 복수전공으로 선택하게 되었습니다. 본격적으로 개발자의 길에 들어선 이후 React, Next.js, TypeScript 기반으로 다수의 웹 서비스를 구현하고 배포하며 실무 경험을 쌓아왔습니다.

WorkOnward에서는 지도 기반 채용 검색 서비스의 핵심 기능을 개발하여 최대 **500개** 공고를 지도에 실시간으로 표시하고, **MongoDB** 쿼리 최적화로 쿼리 속도를 **30%** 개선한 바 있습니다. 또한 **디자인 시스템**과 **공통 컴포넌트 구조**를 관리 및 구축하여 팀 생산성과 코드 일관성을 높였습니다.

현재는 Aicy에서 **AI 기반 재무제표 분석 챗봇**을 개발하여 SSE 기반 스트리밍으로 즉각적인 응답 경험을 제공하였고, **빌드 사이즈 최적화**로 페이지 번들 사이즈와 로드 속도를 대폭 개선했습니다.

저는 "작동하는 코드"와 "좋은 코드"의 차이를 고민합니다. 당장의 기능 구현도 중요하지만, 사용자 경험을 해치는 요소는 없는지, 이 코드를 6개월 후에도 쉽게 이해하고 수정할 수 있을지를 함께 생각하며 개발합니다.

Skills

- Frontend: **React.js, Next.js, Typescript**
- State Management: **React Query, Zustand, MobX**
- Styling: **Tailwind CSS, Ant Design, Mantine**

Experience

Aicy

2025.06 - 재직중 | 풀스택 개발자 | Seoul, South Korea

AI 재무제표 분석 챗봇 서비스

프로젝트 설명: Gemini API 기반으로 사용자가 업로드하는 다양한 포맷의 재무제표를 분석해 대화형으로 제공하는 웹 챗봇 서비스.

기술 스택: Next.js, Typescript, React, NestJS, Gemini API

핵심 역할

- PDF, Excel, 이미지 형식의 재무제표 업로드 및 AI 기반 채팅 인터페이스 개발
- SSE(Server-Sent Events)를 활용한 실시간 AI 응답 스트리밍 구현
- Gemini 및 OpenAI API 통합 및 Markdown 형식의 AI 응답 시각화

이슈

- 100장 이상의 대용량 PDF 파싱 시 **proxy timeout**으로 인한 업로드 실패가 빈번하게 발생
- 여러 페이지에서 비슷하거나 같은 AI 채팅 기능이 필요해 SSE 연결 및 메시지 상태 관리 로직이 중복됨

해결

- 서버에서 백그라운드 파싱 작업 처리 및 프론트엔드에서 **polling** 방식으로 파싱 완료 상태 체크
- 커스텀 훅으로 SSE 연결 및 스트리밍 관리 로직을 구현하고, 재사용 가능한 메시지 공통 컴포넌트 개발

성과

- 100장 이상의 대용량 PDF 업로드 **성공률 90% 이상** 유지
- 3개 이상의 페이지에서 일관된 채팅 경험 제공 및 코드 중복 제거

빌드 사이즈 최적화

프로젝트 설명: Next.js 기반 웹앱의 초기 번들 사이즈 및 로드 속도를 최적화하기 위한 빌드 구조 개선 프로젝트. Tree-shaking 버그를 해결하여 초기 로딩 성능 개선.

기술 스택: Next.js, Lighthouse, Bundle Analyzer, Webpack

핵심 역할

- Next.js 빌드 분석을 통한 번들 사이즈 병목 지점 파악 및 최적화 전략 수립
- Tree-shaking 버그 수정 및 dynamic import를 활용한 코드 스플리팅 구현
- 페이지별 번들 사이즈 감소 및 Core Web Vitals 개선

이슈

- 빌드 분석 결과 최대 **1.1MB** 페이지 발견, 간단한 login 페이지도 **500KB** 초과
- Tree-shaking이 제대로 동작하지 않아 불필요한 코드가 번들에 포함됨
- echarts 등 대용량 라이브러리가 초기 로드 시 무조건 포함되어 성능 저하
- 폰트 및 이미지 최적화 미적용으로 LCP/FCP 지연

해결

- package.json에 **sideEffects** 설정을 추가하여 tree-shaking 활성화
- echarts 등 대용량 라이브러리와 조건부 렌더링 컴포넌트(모달 등)에 **dynamic import** 적용
- **next/font**를 활용한 폰트 최적화 및 **next/image**를 통한 이미지 최적화 구현

성과

- 전체 번들 사이즈 **65% 감소**
- Lighthouse Performance 점수 **12점 증가 (68 → 80)**
- **LCP 3초 단축**

WorkOnward

2022.05 - 2025.03 | 풀스택 개발자 | Remote, US

지도 기반 채용 검색 서비스

프로젝트 설명: 하이퍼로컬 채용 플랫폼의 지도 UI 및 채용 데이터 렌더링 구조를 프론트엔드에서 주도적으로 개발. GraphQL 기반 데이터 패칭 구조를 최적화하고, 실시간 인터랙션 UX를 강화.

기술 스택: Next.js, Typescript, React, GraphQL, MongoDB, MobX

핵심 역할

- 지도 드래그/줌 시 viewport 기반 동적 데이터 로딩 로직 개발
- GraphQL과 MongoDB aggregation을 활용한 데이터 페칭 최적화
- URL 기반 채용 공고 상세 모달 시스템 구현

이슈

- 지도 영역 변경 시 불필요한 API 호출 발생으로 과도한 서버 요청 및 사용자 경험 저하
- **500개 이상의 마커 렌더링 시 과도한 리렌더링 및 데이터 오버페칭 발생**
- 페이지네이션에서 카운트 조회와 데이터 조회를 별도로 요청하여 쿼리 시간 증가
- 채용 공고 클릭 시 페이지 전환으로 지도가 사라지면서 하이퍼로컬 플랫폼의 핵심 가치인 지역 기반 탐색 경험이 단절되는 **치명적인 UX 문제 발생**

해결

- 디바운싱을 적용하여 지도 영역 변경이 완료된 후에만 API 호출
- **GraphQL을 활용하여 지도 마커에 필요한 최소한의 필드만 선택적으로 요청하여 데이터 오버페칭 방지**
- **useMemo**를 활용하여 동일 위치의 채용 공고를 하나의 마커로 클러스터링, 불필요한 리렌더링 최적화
- **MongoDB aggregation pipeline과 facets**를 사용하여 페이지네이션 데이터와 전체 카운트를 단일 쿼리로 조회
- **Next.js Intercepting/Parallel Routes**를 활용한 URL 기반 채용 공고 상세 모달 구현으로 지도 위에 모달을 오버레이하여 지도 컨텍스트 유지, 새로고침 시에도 모달과 지도 상태 보존

성과

- 일일 사용자의 **50%**가 사용하는 핵심 기능으로 성장
- 지도 이동 시 응답 속도 **1-3초 내 유지**
- 페이지네이션 쿼리 시간 **30-50% 단축**
- URL 기반 모달로 채용 공고 딥링크 공유 가능 및 **SEO 최적화 달성**

협업 및 개발 프로세스 개선

이슈

- 디자이너/기획자, 개발팀, QA 팀으로 구성된 개발 프로세스에서, 개발 완료 후 QA 단계에서 기획자가 고려하지 못한 엣지 케이스와 예외 상황이 지속적으로 발견됨

- QA에서 발견된 이슈로 인한 재작업이 반복되어 출시 일정 지연 및 개발 리소스 낭비

해결

- 개발 착수 전 디자이너/기획자, 개발자, QA가 참여하는 **리뷰 세션**을 도입하여 유저 플로우와 **예상 엣지 케이스**를 사전에 논의
- 개발 단계를 여러 **마일스톤**으로 나누고, 각 마일스톤마다 간단한 리뷰 세션을 통해 구현된 기능 검증 및 엣지 케이스 조기 발견

성과

- 마지막 QA 단계에서 발생하는 **추가적인 엣지 케이스 대폭 감소**
- 엣지 케이스 사전 발견으로 급한 분기 처리 및 예외 코드가 최소화되어 **코드 품질 및 유지보수성이 향상**

Education

Stony Brook University

2022.05 졸업 | 컴퓨터공학 · 기술경영

Contact

LinkedIn: [linkedin.com/in/hasungjun](https://www.linkedin.com/in/hasungjun)

GitHub: github.com/iianjun

Portfolio: hasungjun.com