



# 전종훈

Frontend Engineer

☎ 010-9227-4072  
✉ jonghoon.jeon.0209@gmail.com  
🌐 <https://github.com/JunJongHun>

## 소개

React, TypeScript, React-Query를 활용하여 **사용자 경험을 최우선**으로 고려하여 어플리케이션을 구현합니다.

CloudFront, S3를 이용해 **FE 인프라**를 구축하고 관리하고, GitHub Actions를 통해 **지속적 배포**를 수행할 수 있습니다.

End-To-End 프로젝트를 수행한 경험과 AWS Lambda를 사용해 **Serverless**를 구현 할 수 있습니다.

## 프로젝트

### 주말내집 / Frontend

[Web](#) [Github](#)

듀얼 라이프를 꿈꾸고 실현하는 사람들 간의 소통의 공간 부족과 지역 사회의 빈집 문제 해결을 위한 플랫폼

기술 스택 | React TypeScript Zustand React-Query Vite Sass

#### [성능 및 UX 개선]

- 반응형 UI 개발과 Throttle 적용을 통해 resize 이벤트의 과도한 호출을 방지하며, 성능 최적화된 웹 인터페이스를 구현
- 번들 사이즈 증가함에 따라 Route Level에서 **Code Splitting**을 적용해 **초기 로딩 속도 2s → 1s로 개선**
- React-Query의 캐싱을 사용하여 '좋아요' 기능에 Optimistic UI 업데이트를 구현하여 서버 응답을 기다리지 않고 인터페이스 반응 속도를 즉각적으로 구현
  - 이슈: 서버 요청을 실패해도 성공적으로 UI 업데이트가 진행되어 DB 데이터와 캐싱된 데이터 불일치 문제 발생
  - 해결: 이전 쿼리 상태의 SnapShot을 활용해 API 요청에 실패할 경우 **Rollback** 적용하여 **데이터 불일치 문제 해결**

#### [이미지 최적화]

- CloudFront + Lambda@Edge를 적용하여 **이미지 5MB→316KB 95%최적화** 및 캐싱 적용하여 동일한 이미지 요청에 대해서 **2s → 43ms로 97% 감소**
- 모바일 화면에서의 빠른 이미지 로딩을 위해 srcset을 사용하여 해상도에 맞는 이미지를 제공하여 1s → 0.4s로 감소

#### [로그인, 회원가입 및 인증 서비스 개발]

- 많은 입력 폼들을 제어 컴포넌트로 관리하는 대신 비제어 컴포넌트 기반인 React-Hook-Form을 사용해 유효성 검사 및 에러 핸들링 로직을 코드 가독성과 유지보수를 고려하여 구현
- 로컬 스토리지와 httpOnly 쿠키를 통해 JWT 토큰을 안전하게 관리

#### [상태관리]

- Zustand를 사용한 전역 상태 관리 및 persist 미들웨어 적용으로 페이지 새로고침 시 데이터 손실 방지
- React Query를 사용하여 서버 데이터를 캐싱하여 서버 요청을 줄이고, 사용자에게 캐싱된 데이터를 제공하여 사용자 경험 향상
  - 이슈: 게시글 데이터를 수정하고 캐시를 업데이트했지만, 연관된 데이터의 캐시가 갱신되지 않아 데이터 불일치가 발생
  - 해결: 데이터 무효화 로직을 도입하여 백그라운드에서 데이터를 자동 갱신하여 **최신의 데이터와 정확성을 보장**

기술 스택 | [React](#) [TypeScript](#) [React-Query](#) [Vite](#) [Chakra UI](#) [CloudFront](#) [S3](#) [Api Gateway](#) [Lambda](#) [DynamoDB](#)

### [성능 및 UX 개선]

- **Intersection Observer API**를 사용하여 사용자가 페이지 끝에 도달하기 전에 추가 콘텐츠를 자동으로 로드하도록 구현
- **ErrorBoundary**를 사용해 UI의 일부분에서 발생하는 자바스크립트 에러가 **전체 애플리케이션 중지 하지 않도록 구현**
- API 응답 지연시 사용자에게 예측 가능한 Skeleton UI를 보여줌으로써 **렌더링 시간에 대한 체감 감소**  
→ 고민: 너무 짧은 로딩에도 Skeleton UI를 보여줌으로써 오히려 사용자 경험이 떨어질 수 있다는 판단  
→ 해결: 무조건 Skeleton UI를 보여주는 것은 사용자 경험에 좋지 않을 수 있기에 응답 속도 0.3s 기준으로 렌더링

### [비즈니스 로직과 UI 로직 분리]

- 컴포넌트 내 비즈니스 로직을 **Custom Hook**으로 추상화 하여 **재사용성과 UI에 개발에 집중** 할 수 있도록 분리

### [클라우드 서비스 배포 및 CD 적용]

- CloudFront + S3로 배포하여 **초기 로딩속도 85%이상 감소**
- Github Actions를 사용하여 프론트엔드 CD 파이프라인 구축 및 **캐시 무효화 적용**

### [API 서버 구축]

- Api Gateway + Lambda + DynamoDB를 사용해 Serverless 아키텍처 구현
- 이벤트 브릿지를 활용하여 Cold Start 문제 해결하여 **응답속도 평균 1.2s → 200ms으로 감소** ( CloudWatch 로그를 통해 Init Duration 항목 확인 )

## 스킬

- **Language** : TypeScript, JavaScript
- **Bundling tools** : Vite
- **Containerization**: Docker
- **Infra**: S3, CloudFront
- **Frontend Dev** : React, NextJS
- **Styling Tools** : Sass, Styled-Component, Tailwind
- **State Manage Lib** : Zustand, React-Query

## 경험

- **[Techeer] 2023 WINTER 실리콘밸리 부트캠프 ( Frontend 기술 멘토 )** | 2023.12 - 2024.02
- **원티드 프론트엔드 프리온보딩 인턴십** | 2023.04 - 2023.05
  - 원티드에서 진행하는 동료학습 기반 기업 과제 훈련 교육과정
- **Techeer** | 2023.02 - (진행중)
  - 실리콘밸리 개발자의 **SW 개발자 커리어 그룹**
  - 브라우저 렌더링 과정 및 성능개선 테크톡 진행
  - 기술 세션, 해커톤, 프로젝트, 스터디 등 다양한 개발 및 네트워킹 활동에 집중하는 그룹
- **[티타임즈 x Techeer] 2022 WINTER 실리콘밸리 부트캠프** | 2022.12 - 2023.02
- **ACM-ICPC ( 전국 대학생 프로그래밍 경시대회 ) 본선 진출** | 2021.11

## 학력

- 성결대학교 컴퓨터공학과 전공 | 2018.03 - 2024.02