



SeungJae Ahn

FRONTEND DEVELOPER

ABOUT ME

TECH SKILL

EXPERIENCE

PROJECT

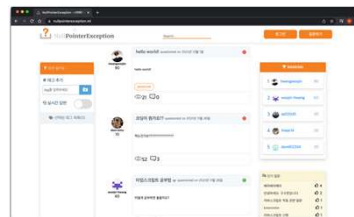
EDUCATION

## 개발자를 위한 실시간 Q&A 서비스(Typescript, Next.js, GraphQL)

**소개** Stackoverflow와 같은 개발자 Q&A와 실시간 답변 기능

**본인 역할** Markdown기반 질문글 작성, Socket기반 채팅 **배포 링크** <https://nullpointerexception.ml/>

**주요 기능** 커뮤니티 / 글 작성 / 동시 편집 / 채팅등 Q&A에 필요한 기능과 실시간으로 답변할 수 있는 기능 구현



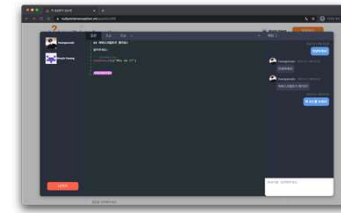
### 메인 화면

메인 화면에서 질문글들을 볼 수 있고 검색, 필터링이 가능. 질문 글은 무한 스크롤링으로 불러오도록 구현



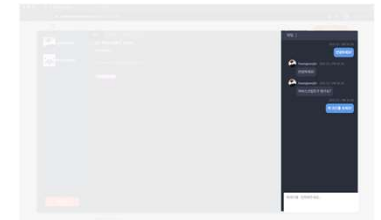
### 글 작성

마크다운 방식으로 글을 작성할 수 있도록 구현



### 동시 편집

질문글, 코드등을 동시에 편집할 수 있는 기능. 실시간으로 답변이 가능



### 채팅

실시간 음성 채팅과 사이드바에서 메시지와 같은 채팅을 할 수 있는 기능

## 문제 해결 경험

### 서버사이드 렌더링을 지원하지 않는 npm 모듈

- npm 모듈을 불러와서 사용할 때 'window is not defined' 라는 에러가 발생. 서버사이드 렌더링이기 때문에 window라는 브라우저에서 제공하는 객체가 없어서 생긴 문제였음.
- 실제 해당 npm 모듈의 Github페이지에서도 서버사이드 렌더링을 지원하지 않는다는 내용이 있었음
- dynamic import를 통해 필요한 부분만 클라이언트 렌더링으로 바꿀 수 있는 것을 배우게 됨. 클라이언트 렌더링이 필요한 컴포넌트에만 dynamic import적용

### 기대대로 되지 않았던 아토믹 디자인 적용

- 처음 기대했던 컴포넌트의 재사용이라는 장점은 시간이 지날수록 사라져갔다.
- Button이라는 Atom컴포넌트를 만들어도 서비스 전체에 쓰이는 서로 다른 버튼을 전부 커버할 수 없었음.
- 결국 Atom, Molecule등의 하위 컴포넌트를 재사용하는 것보다 직접 새로 만드는게 더 편하다는 의견이 앞서게됨.

## 적용 기술



코드스타일 통일을 위해 ESLint와 Prettier 사용

Next.js를 이용하여 서버사이드 렌더링 구현



리액트 구조로 아토믹 디자인 적용



Typescript 적용