



# 김서연

Frontend Developer

1998.11.10 (만 25세)

**Github** : <https://github.com/aeong98>

**Blog** : <https://velog.io/@aeong98>

Phone 010-5027-5301 | Email sy9815@gmail.com | Address 성남시 위례중앙로 216

## 자기소개

'도전'과 '성장'을 추구하는 사람으로, 새로운 기술을 습득하고 실제 결과물을 도출해내는 과정에서 가장 큰 보람을 느낍니다. 협업을 통해 성장하는 것을 즐깁니다. 최근에는 요구사항 변경에 유연하게 대응할 수 있는 디자인 패턴에 관심을 가지고 공부하고 있습니다.

## 경력사항

재직 기간	2021.08-2022.08 (1년)
회사명 / 직무	(주)젠틀에너지 ICT팀 프론트엔드 개발자
사용 기술	Next.js, Typescript, Redux Toolkit, React Query, React Echart.js, SASS, Storybook, Github Action, Vercel, Azure VM, Azure Function, Jquery
주요 업무	<ul style="list-style-type: none"><li>- 스마트팩토리 솔루션 웹대시보드 팩토로이드 <b>프론트엔드 개발</b></li><li>- 팩토로이드 <b>가동 현황, 가동 이력 조회 기능, 예지보전</b> 기능 개발</li><li>- 백오피스용 <b>어드민 페이지</b> 개발</li><li>- 차트 라이브러리 커스텀을 통한 <b>데이터 시각화</b></li><li>- 디자인 시스템 적용, <b>재사용가능한 컴포넌트</b> 개발 (MonoRepo)</li><li>- 팩토로이드 기능 모듈화 및 컴포넌트 설계 (기획 단계 참여)</li><li>- Springboot 기반 <b>레거시 프로젝트(Vanilla JS, Jquery) Next.js 로 마이그레이션</b> 및 기능 개선</li></ul>

## 동아리 및 부트캠프 이력

기간	2021.07-2021.08 (2개월)
부트캠프명 / 역할	2021 SiliconValley Internship 부트캠프 팀장/서버개발
내용	<ul style="list-style-type: none"><li>- 사용기술 : React.js, Flask, PostgreSQL, ElasticSearch, Docker</li><li>- 그림책 추천 서비스 iHome 서버 개발 및 팀장 역할 수행</li><li>- Flask API 개발, ElasticSearch 벡터 유사도 검색을 활용한 추천 쿼리 개발</li><li>- RabbitMQ, celery사용한 비동기 처리 아키텍처 구현</li></ul>
기간	2021.04-2021.12 (8개월)
동아리명 / 역할	멋쟁이 사자처럼 서강대 9기 프론트엔드 개발
내용	<ul style="list-style-type: none"><li>- 사용기술 : React.js, Django, AWS RDS,EC2, Docker</li><li>- 인테리어 커뮤니티 웹서비스 iHome 프론트엔드 개발</li><li>- Django 교육세션 참여 및 방학 프론트엔드 스터디 참여 (노마드 코딩)</li></ul>

## 핵심역량

### 1. React.js, Next.js / redux(redux toolkit), recoil, react query

- Hooks 라이프사이클을 이해하고 개발 (custom hook 개발)
- 재사용 가능한 컴포넌트를 설계 및 storybook 명세화
- Redux, recoil 을 사용해 클라이언트 전역 상태관리 / React Query 사용 가능
- Next.js를 사용한 SSR 클라이언트 개발.
- Monorepo를 사용한 공통 패키지 개발, 디자인시스템 적용

### 2. Vanilla JS(ES6), Typescript, SASS

- Vanilla JS(ES6), webpack, babel만으로 이루어진 SPA 개발 가능
- Typescript 를 사용해 정적 타이핑 및 에러 핸들링이 가능
- SASS @mixin을 사용해 반복되는 스타일링 모듈화

### 3. Serverless Backend (AWS, Firebase)

- AWS lambda, Firebase를 사용한 서버리스 백엔드 개발
- Next.js API 서버(Express.js)를 사용해 에러 로깅 API 개발
- slack webhook, notion API 연동한 로깅, 모니터링 자동화
- PostgreSQL, ElasticSearch 쿼리 개발 및 활용
- Docker 사용한 프로젝트 세팅 및 배포

## 프로젝트 이력

프로젝트 기간	2022.01-2022.08
인원	(프론트엔드 3명)
프로젝트 명	(주)젠틀에너지 Factoroid 웹대시보드 리뉴얼 프로젝트
사용 기술	<p><b>Next.js</b> : SSR 캐싱 기능 사용 및 proxy API 개발을 위해 Next.js 프레임워크 사용</p> <p><b>Typescript</b> : 데이터 정적 타입 정의 및 에러 핸들링을 위해 사용</p> <p><b>Redux toolkit, React Query</b> : 클라이언트, 서버 상태 분리를 위해 사용</p> <p><b>React Echart.js</b> : 데이터 시각화 및 차트 커스터마이징을 위해 사용</p> <p><b>SASS</b> : @mixin 을 사용한 스타일 모듈화, 자동완성 지원</p> <p><b>l8n</b> : 다국어 지원을 위해 사용</p> <p><b>React Table</b> : 테이블 기능 고도화 및 스타일 커스터마이징이 용이해 사용</p> <p><b>Storybook</b> : 디자인 시스템 적용 이후, 컴포넌트 명세화와 커뮤니케이션에 사용</p> <p><b>Github Action, Vercel, Chromatic</b> : 프로젝트 배포 및 운영을 위해 사용</p>
주요 업무	<p><b>1) 가동현황 (실시간 공장 현황) 대시보드 개발 및 기능 모듈화</b></p> <p>1-a. polling 방식의 실시간 대시보드 구현</p> <ul style="list-style-type: none"> <li>- react query refetchInterval을 사용 했을 때 이전 요청의 응답 대기로 인한 실시간 주기 지연 현상 이슈 발생</li> <li>- setInterval web API을 사용해 일정한 주기의 실시간 대시보드기능 구현. Redux 에 stack 자료구조 만들어 실시간 데이터 관리</li> <li>- Promise.all 병렬처리를 사용해 데이터 호출, 렌더링 횟수 최적화</li> </ul> <p>1-b. React Echart.js 사용 전력/빛/진동 등 센터 타입 별 그래프 개발</p> <ul style="list-style-type: none"> <li>- redux toolkit 사용해 실시간 데이터 관리 (실시간 데이터 저장소)</li> <li>- 센서 타입 별 단위 표시, 데이터 표시 형식 util 개발 및 모듈화</li> <li>-</li> </ul> <p><b>2) 가동히스토리 대시보드 개발 및 기능 모듈화</b></p> <p>2-a. Next.js SSR(getServersideProps) 사용해 조회 시간 단축 및 페이지 캐싱 적용</p> <ul style="list-style-type: none"> <li>- React Query prefetchQuery, dehydrate 사용해 서버에서 사전에 데이터를 패칭하고 클라이언트단에서 공유할 수 있도록 설계</li> <li>- 조직 / 조회 기간 / 가동 상태 필터링, 정렬 기능 개발,</li> <li>- Next.js useRouter사용해 조회 조건 저장. 링크 연결 및 뒤로가기 기능 고도화 (커스텀 혹은 개발해 query parameter 와 관련된 상태관리 및 UI 로직 분리)</li> <li>- 전력사용량 바 그래프, 가동상태 시계열 그래프 개발</li> </ul>

### 3) 백오피스용 어드민 페이지 개발

3-a. React Table 라이브러리를 사용해 재사용 가능한 Table 컴포넌트 개발

- 테이블 기능 고도 (필터링, 정렬, 리사이징, 페이지네이션, 체크박스 등 기능 구현)
- useTable 커스텀 훅 개발해 반복되는 보일러 플레이트 코드 줄이고, UI 관련 로직을 재사용할 수 있도록 모듈화

3-b. 입력 폼 유효성 검증을 위한 커스텀 훅 (useForm.tsx) 및 validation util 개발

3-c. 재사용 가능한 컴포넌트 개발, 디자인 시스템 적용

- Select 컴포넌트 (검색기능, 다중 선택 기능, 스타일 설정)
- Input, Button, Modal 컴포넌트 등 (컴포넌트 상태, 스타일 설정)

3-d. kakaoMAP API 를 사용한 지도 뷰 및 주소 검색 기능 개발 (툴팁 커스텀)

### 4) 프로젝트 세팅 및 storybook, 테스트 툴 연결

4-a. storybook, chromatic 세팅 및 github action으로 CI/CD 파이프라인 개발

4-b. jest, react-testing-library 세팅 redux 관련 unit test 진행.

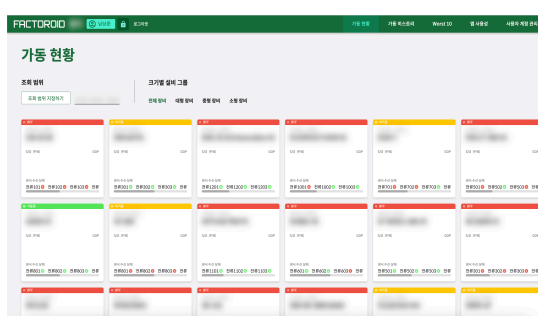
4-c. 모노레포 세팅(yarn workspace, turborepo) UI 패키지 분리 및

4-d. i18n 을 사용한 다국어 지원 세팅

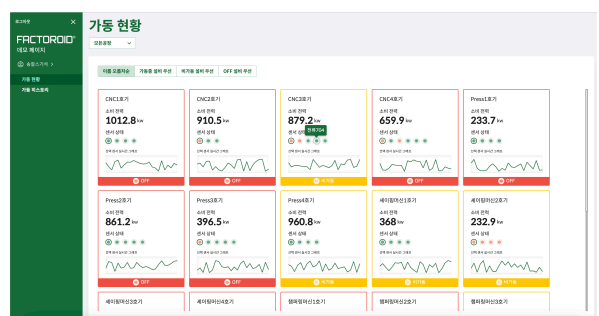
### 주요 성과

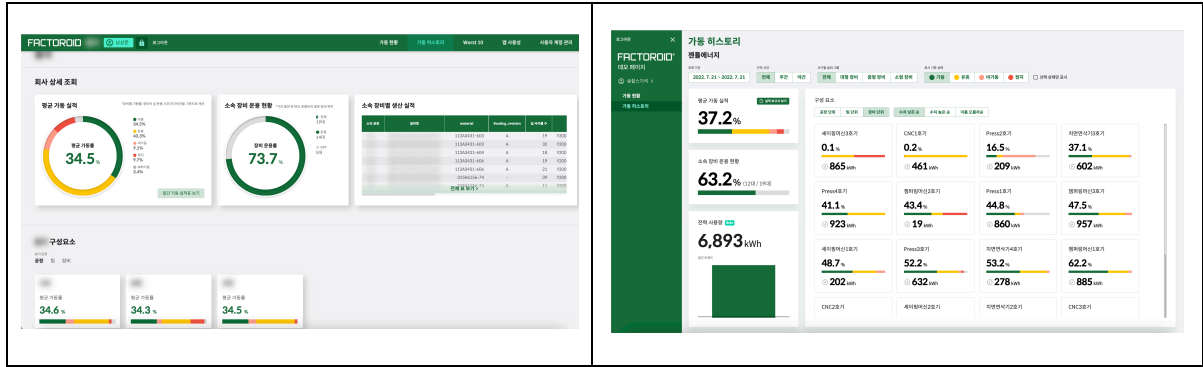
- 5개 이상의 고객사 온보딩
- 고객사별로 프로젝트 만들던 기존의 개발 프로젝트 구조 개선
- springboot 환경에서의 레거시 프로젝트 분리, Next.js 로 마이그레이션 (고객사마다 새로 페이지를 개발해야 했던 비효율적인 개발 구조 개선)
- 기능 모듈화를 통해, 고객사마다 다이나믹한 페이지를 볼 수 있도록 구현
- 재사용가능한 컴포넌트 개발 (UI 패키지 구현)을 통한 개발 생산성 향상

### 리뉴얼 이전



### 리뉴얼 이후







프로젝트 기간	2021.08-2021.12
인원	(프론트엔드 2명)
프로젝트 명	(주)젠틀에너지 기존 Factoroid 추가 기능 개발 및 유지보수 (고객사 3 개 참여)

사용 기술	<p><b>Vanilla JS(ES6), JQuery</b> : SpringBoot 환경에서의 레거시 프로젝트 유지보수</p> <p><b>Webpack, Babel</b> : JS 모듈 번들러로 사용, hot reload 서버로 개발</p> <p><b>Chart.js</b> : 차트 개발 및 데이터 시각화를 위해 사용</p> <p><b>Stomp.js</b> : 웹소켓 통신을 위해 사용</p> <p><b>Panzoom.js</b> : SVG 파일 애니메이션/인터랙션 적용을 위해 사용</p> <p><b>Google Analytics</b> : 사용성 분석 및 사용자 로깅을 위해 사용</p>
-------	--

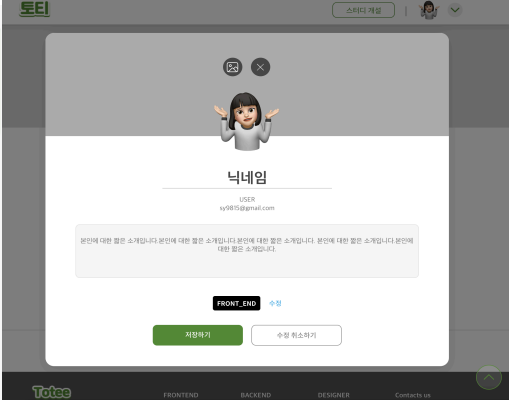
주요 업무	<p><b>1) X사 : 공장 현황 조회 실시간 대시보드 기능 고도화 / 로그인 기능 개선</b></p> <p>1-a. SVG 도면을 사용한 실시간 공장 현황 대시보드 구현</p> <ul style="list-style-type: none"> <li>- Panzoom.js 를 사용해 SVG 도면 zoom-in, zoom-out, 등 인터랙션 구현 및 실시간 대시보드 기능 고도화</li> <li>- socket 통신(Stomp.js) 을 사용한 실시간 데이터 페칭 구현</li> </ul> <p>1-b. Vanilla JS 클래스형으로 재사용 가능한 컴포넌트 개발</p> <ul style="list-style-type: none"> <li>- 프레임워크 없이 Vanilla JS 클래스형으로 상태관리 및 재사용 가능한 컴포넌트 개발 (Modal, Filter, Card 등 ..)</li> <li>- 옵저버 패턴 적용해 상태 업데이트됐을 경우, 리렌더링 될 수 있도록 구현</li> <li>- 이벤트 위임 (Event Delegation) 기법을 사용해 메모리 절약, 코드 모듈화</li> </ul> <p>1-c. 로그인 기능 개선</p> <ul style="list-style-type: none"> <li>- 로그인 페이지 UI 개선 및 로그인 자동완성 기능 추가 (localStorage 이용)</li> <li>- 비밀번호 찾기 기능 추가 (인증코드 메일링 마크업 개발)</li> </ul> <p><b>2) Google Analytics 로 사용자 방문 이력 관리 기능 개발, 로깅 기능 추가</b></p> <p>2-a. Azure Function 사용해 GA 로깅 서버리스 API 개발</p> <p>2-b. GA 연동, 사용자 이용 현황 조회 기능 개발</p>
-------	---

	<b>3) 고객사 전달용 서버리스 API 개발</b> 3.a. Azure Function 사용해 python으로 고객사 전달용 raw data 조회 API 개발 3.b. 보안을 위한 API Key 추가. 예외처리 및 에러메시지 정의
<b>주요 성과</b>	<ul style="list-style-type: none"> <li>- X사, Y사 계약 성사</li> <li>- 투자 유치</li> <li>- 에러 로깅 및 조회 시스템 구현</li> <li>- 팩토리드 초기 핵심 기능 개발 및 고도화</li> <li>- 기존 팩토리드 UI/UX 개선</li> </ul>


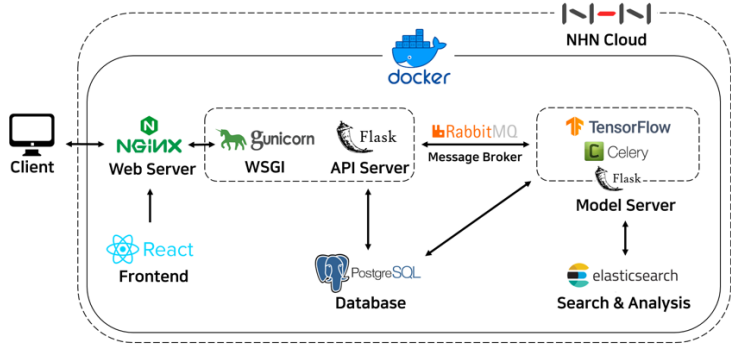
가동현황 - X사	가동현황 - Y사
	

## 사이드 프로젝트 이력

<b>프로젝트 기간</b>	2022.06 ~ 진행중
<b>프로젝트 명</b>	멘토 멘티 매칭 웹사이트 Totee
<b>Github / 홈페이지</b>	Github Repo: <a href="https://github.com/Totee-Side-Project/totee-fe">https://github.com/Totee-Side-Project/totee-fe</a> Project: <a href="https://totee-fe-omega.vercel.app/">https://totee-fe-omega.vercel.app/</a>

 <p>메인.화면</p>	 <p>마이페이지 회원정보 수정 모달</p>
--	--

사용 기술	<p><b>React.js(v18)</b> : Automatic Batching 으로 렌더링 횟수 최적화 지원으로 인해 사용</p> <p><b>Vite</b> : 빠른 빌드 지원으로 인해 사용</p> <p><b>Typescript</b> : 데이터 정적 타입 정의 및 에러 핸들링을 위해 사용</p> <p><b>Recoil, React Query</b> : 높은 생산성으로 사용/ 서버상태 관리를 위해 사용,</p> <p><b>SASS</b> : 중첩, 변수 선언을 통한 코드 가동성 향상을 위해 사용</p> <p><b>ReactQuill, SweetAlert</b> : 게시판 기능 구현 / 알람 기능 구현을 위해 사용</p> <p><b>Eslint</b> : 일관된 코딩 컨벤션 적용을 위해 사용</p>
주요 업무	<ol style="list-style-type: none"> <li>1) Totee 주요 기능 구현. <ul style="list-style-type: none"> <li>● 메인페이지 배너, 카테고리 및, 정렬 필터링 기능 구현 및 UI 개</li> <li>● 프로필 사진등록 등 정보 입력을 위한 온보딩 모달 개발</li> <li>● 댓글 / 대댓글 기능 구현 및 UI 개발</li> <li>● 마이페이지 회원 정보 수정 기능 개발</li> <li>● JWT 로그인 및 소셜로그인 연동 (axios interceptor, localStorage 를 사용해 토큰 관리)</li> </ul> </li> <li>2) 자동 완성 검색 기능 개발 <ul style="list-style-type: none"> <li>● React Query 에서는 axios 호출 함수에 debounce 를 걸 수 없다는 이슈 발생,</li> <li>● useDebounce 커스텀 훅 개발해, React Query 에 key 로 들어가는 value 값의 상태 변경 횟수를 제한. api 호출 횟수 최적화</li> </ul> </li> <li>3) 재사용 가능한 컴포넌트, 커스텀 훅 개발 <ul style="list-style-type: none"> <li>● Input, Button, Modal, Select 등 재사용 가능한 컴포넌트 개발</li> <li>● useProfileImage, useOutsideAlerter 등 반복되는 UI 로직을 커스텀 훅으로 모듈화 및 재사용</li> </ul> </li> <li>4) 협업 프로세스 성립 <ul style="list-style-type: none"> <li>● Github Issue 템플릿 지정, feature 단위 별 개발 프로세스 성립</li> <li>● Postman, Swagger 를 사용한 백엔드와의 협업</li> <li>● 서비스 기능 기획 및 디자인 과정 참여</li> </ul> </li> </ol>
주요 성과	

프로젝트 기간	2021.06-2021.07
프로젝트 명	객체 인식 기반 AI 그림책 추천 서비스 iHome
Github	<a href="https://github.com/SiliconValleyLorax/i-home">https://github.com/SiliconValleyLorax/i-home</a>
 	<p>iHome UI</p> <p>소프트웨어 아키텍처</p>
사용 기술	<p><b>React.js</b> : SPA 개발을 위해 사용</p> <p><b>Flask</b> : 빠른 PoC 프로젝트 개발, 높은 생산성, 낮은 러닝커브로 인해 선택</p> <p><b>PostgreSQL</b>: UUID, JSON 등 다양한 함수, 타입 지원으로 인해 선택</p> <p><b>ElasticSearch</b> : 추천 검색 엔진으로 사용하기 위해 사용</p> <p><b>RabbitMQ</b> : 비동기 처리를 위한 메시지큐로 사용하기 위해 도입</p> <p><b>Docker</b> : 개발 환경 세팅 및 개발 인스턴스 배포 편의성을 높이기 위해 도입</p>
주요 업무	<ol style="list-style-type: none"> <li>프로젝트 아키텍처 설계 및 세팅 담당 <ul style="list-style-type: none"> <li>모델 처리 과정이 오래 걸릴 경우 파이썬 싱글스레드의 한계로 서버 부하가 증가하고 응답이 지연되는 이슈 발생</li> <li>API 서버, 모델 서버를 분리해 서버 부하 분산</li> <li>메세지큐 (RabbitMQ, celery)를 사용한 비동기처리 아키텍처 구현</li> </ul> </li> <li>추천 검색 엔진 개발을 위한 ElasticSearch 도입 <ul style="list-style-type: none"> <li>Tensorflow 오픈 소스(Universal Sentence Encoder)를 사용해 그림책 텍스트 데이터 벡터화 및 검색 엔진 활용</li> <li>ElasticSearch cosine similarity 검색 쿼리개발, 추천 시스템 구현</li> </ul> </li> <li>docker 컨테이너 구성 (docker-compose 작성)</li> <li>팀장으로 일정 조율 및 창업자와의 커뮤니케이션 담당</li> </ol>
주요 성과	<ul style="list-style-type: none"> <li>베타 테스트 설문조사 (50 명) 만족도 95% 달성</li> <li>창업 제안</li> </ul>



## 교육

2017-2022	서강대학교 경영학과 / 융합소프트웨어
-----------	-------------------------

## 자격증·시험

2020.06	정보처리기사1급
2020.05	TOEIC 945