

김의현

Front-end Developer

+82 10-8310-8853 rladml91gus@naver.com

Profile

안녕하세요. 프론트엔드 개발자 김의현입니다.

3년간의 실무에서 사용자가 더 편리하게 서비스를 이용할 수 있도록 UI/UX 기능을 개발하고 개선했으며 코드 품질 향상에도 집중했습니다.

팀의 생산성을 높이고 유지보수하기 쉬운 구조를 고민하며 새로운 기술을 빠르게 익혀 적용하는 것을 즐깁니다.

앞으로도 사용자 중심의 서비스를 만들고 함께 일하는 동료들에게 긍정적인 영향을 주는 개발자가 되고자 합니다.

Skills

JavaScript, TypeScript, React.js, Next.js, Sass, Tailwind CSS, MobX, Zustand, Tanstack Query, Jest, Testing Library, Vite

Experiences

(주) 티맥스클라우드

2022.09 - 2024.11 | 정규직

No Code / Low Code 백엔드 개발 스튜디오

- 비즈니스 요구 사항에 맞춰 Java 기반 백엔드 서비스 코드를 자동으로 생성할 수 있는 No Code / Low Code 플랫폼입니다.
- 비개발자도 손쉽게 API를 설계 & 생성할 수 있도록 직관적인 UI/UX와 코드 자동화 기능을 제공합니다.

기술 스택

JavaScript, TypeScript, React.js, MobX, Sass, React DnD, Java, FreeMarker, Tibero, Figma, Slack, Google Sheet

담당 업무 및 성과

1. No Code Select 서비스 기능 개발

이슈

- 비개발자가 SQL 지식 없이 데이터 조회 API를 생성하는 과정이 복잡해 효율적인 파이프라인이 필요했음

해결

- Drag & Drop 기반 UI로 테이블 선택 및 관계 자동 분석 UX 설계
- 선택된 테이블 컬럼 기반 OutDTO 자동 매핑 시스템 구현
- 조건 및 부가 설정 시 컬럼명/연산자 자동 제안 기능 개발
- 자체 개발한 SQL Generator와 FreeMarker를 활용한 동적 쿼리 및 Java 코드 생성

성과

- 평균 10분 내에 Select API를 손쉽게 생성하는 기능 제공

2. 리스트 순서 변경 UX 개선

이슈

- 기존 버튼 기반 순서 변경 방식의 직관성 부족으로 VOC가 지속 발생

해결

- React DnD 라이브러리를 적용하여 Drag & Drop 방식으로 개선
- GPU 가속 및 CSS 트랜지션을 통한 부드러운 이동 효과 구현
- React.memo, useCallback으로 불필요한 렌더링 최소화

성과

- 사용자 만족도 설문 90% 이상 긍정적 피드백
- VOC 감소 및 직관적인 리스트 관리 UX 제공

3. 중앙 집중식 디이얼로그 관리 시스템 구현

이슈

- 프로젝트 규모가 커지면서 디이얼로그 관리가 컴포넌트별 분산되어 타입 오류와 유지보수 비용 증가

해결

- as const + typeof를 활용한 타입 기반 디이얼로그 정의
- MobX 전역 관리로 디이얼로그 상태를 한 곳에서 제어
- 조건부 렌더링기반 디이얼로그 통합 관리 컴포넌트 구현

성과

- 20개 이상 디이얼로그를 단일 컴포넌트로 관리
- 런타임 오류 대폭 감소 및 유지보수성 향상

4. 서비스 등록 및 메시지 핸들링 시스템 개선

이슈

- 서비스 등록 및 메시지 핸들링 로직이 분산되어 있어 신규 서비스 추가 시 코드 누락이 자주 발생
- 런타임 오류 발생으로 디버깅에 오랜 시간이 소요

해결

- 기존 Map 기반 양방향 매핑 구조를 도메인별 객체 기반 서비스 관리 구조로 개선
- Object.keys 기반 자동 매핑 구조를 설계하여, 핸들러 등록 과정을 코드 1줄로 단순화
- 핸들러 누락 시 컴파일 타임 오류를 강제하는 타입 유ти리티 구현

성과

- 디버깅 시간 90% 이상 단축
- 신규 서비스 추가 작업 시간 대폭 감소 및 코드의 안정성 확보

5. React 순환 의존성 문제 해결

이슈

- 컴포넌트 구조가 복잡해지면서 순환 의존성 문제가 반복적으로 발생
- 빌드 및 런타임 오류가 자주 발생

해결

- 컴포넌트간 직접 import를 제거하고, Props 주입 방식으로 전환하여 결합도를 최소화
- 타입/유ти리티/상수 등의 공통 요소를 별로 모듈로 분리하여 구조 개선

성과

- 빌드 및 런타임 오류 제거
- 테스트 코드 작성 용이성과 코드 가독성 향상

티맥스가이아

2022.02 - 2022.09 | 정규직

RDBMS 기반 데이터 모델링 설계 플랫폼

- 데이터베이스 설계부터 ERD 시각화, 마이그레이션까지 지원하는 온라인 플랫폼입니다.
- 실시간 협업과 데이터 마이그레이션 기능을 제공하여 개발 생산성을 높입니다.

기술 스택

JavaScript, TypeScript, React.js, MobX, Sass, React Flow, Jest, Testing Library, Java, Tibero, Figma, Slack

담당 업무 및 성과

1. 공통 컴포넌트 개발

이슈

- UI 일관성 부족 및 낮은 개발 효율성 문제로 재사용 가능한 UI 컴포넌트 필요

해결

- Figma를 통해 디자이너와 협업하여 다양한 화면에서 공통적으로 사용되는 UI를 재사용 가능한 공통 컴포넌트로 추상화
- Props 기반 유연한 동작 정의 및 스타일 확장성 확보

성과

- UI 일관성 확보 및 사용자 경험(UX) 개선
- 팀 내 개발 생산성 및 유지보수성 향상

2. 테스트 코드 도입

이슈

- 코드 품질 향상 및 버그 사전 방지를 위한 테스트 환경 구축 필요

해결

- Jest + React Testing Library 기반 단위 테스트 환경 구축
- Husky를 활용해 Git push 시 자동 테스트 실행

성과

- 코드 품질 및 안정성 향상
- TDD (Test Driven Development)의 중요성 인식 및 점진적으로 반영

3. ERD 툴 편집 모드 기능 개발

이슈

- 생성/읽기 전용 구조로 협업 편집 기능 부족

해결

- UX 일관성 확보와 컴포넌트 재사용을 위해 테이블 생성 및 편집 기능을 하나의 컴포넌트로 통합하여 리팩토링
- React Flow 노드 더블클릭 기반 편집 모드 진입 기능 구현
- Deep Copy 메서드로 데이터 무결성 보장

성과

- 실시간 협업 편집 기능 제공
- UI/UX 일관성 강화 및 사용자 경험(UX) 개선

4. 데이터 마이그레이션 및 플랫폼 통합 기능 개발

이슈

- Tibero DB 데이터를 자사 플랫폼 DB 스키마로 이관 필요

해결

- ETL 프로세스를 구축하여 메타데이터 추출 및 변환
- ERD 시각화를 위한 위치 정보까지 함께 저장

성과

- 데이터 이전 시간 80% 이상 단축
- Tibero 기반 개발자의 학습 비용 최소화

Projects

포트폴리오 웹사이트

2025.07 - 2025.09 | 1인 프로젝트

- Next.js의 SSR(서버 사이드 렌더링)을 직접 학습하고 실제 프로젝트에 적용하며, 기획부터 배포까지의 전체 개발 프로세스를 경험했습니다.
- 반응형 디자인과 인터랙티브한 UI/UX를 구현하여 사용자 경험을 향상시키는 데 집중했습니다.
- SEO(검색엔진 최적화)와 웹 접근성(a11y)에 대한 이해를 바탕으로 프로젝트의 노출과 사용성을 극대화하기 위해 노력했습니다.

기술 스택

TypeScript, React.js, Next.js, Framer Motion, Tailwind CSS, Prisma, Vercel

담당 업무 및 성과

1. 프로젝트 구조 설계 및 기술 스택 선정

이슈

- 유지보수와 확장성을 고려하면서 최신 기술을 경험할 수 있는 스택 선정이 필요

해결

- Next.js의 SSR과 서버 컴포넌트를 직접 학습·적용하며 효율적인 렌더링 구조 설계
- Vercel PostgreSQL과 배포 환경을 활용해 서버 연동 및 배포 과정을 단순화
- Tailwind CSS를 활용해 빠르고 일관된 UI 구현

성과

- Next.js SSR/서버 컴포넌트의 동작 원리 이해 및 실제 프로젝트 적용 능력 향상
- 서버 연동부터 배포까지 전반적인 개발 프로세스 경험

2. UI/UX 디자인 및 인터랙티브 기능 구현

이슈

- 정적인 정보 나열을 넘어 사용자의 흥미를 유발하고 정보를 효과적으로 전달할 수 있는 동적인 UI/UX가 필요

해결

- Framer Motion을 활용하여 스크롤, 등장, 모달 등 다양한 애니메이션 효과를 적용
- Intersection Observer API를 커스텀 흑으로 만들어, 스크롤에 따라 콘텐츠가 동적으로 나타나도록 구현
- hover/click 인터랙션을 추가하여 데스크탑과 모바일 환경 모두에서 긍정적인 사용자 경험을 제공
- 미디어 쿼리와 clamp() 함수를 사용해 화면 크기에 따라 유연하게 변화하는 반응형 웹 디자인을 구현

성과

- 인터랙티브한 사용자 경험(UX)을 구현하여 정보를 직관적으로 전달
- CSS와 애니메이션 최적화를 통해 부드러운 사용자 경험(UX)을 제공

3. SEO 및 웹 접근성(a11y) 최적화

이슈

- SEO 및 웹 접근성(a11y) 확보를 통해 검색 노출을 높이고, 다양한 사용자가 편리하게 이용할 수 있는 웹사이트 환경 구축 필요

해결

- 시맨틱 마크업을 사용하여 콘텐츠 의미를 명확화
- 스크린 리더 사용자의 이해를 돋기 위해 aria-label을 적용
- WCAG 기준에 맞춰 명도 대비를 설정해 가기성을 높임
- 동적 sitemap & robots.txt 추가를 통해 검색 엔진 크롤링 효율을 높여 노출을 극대화

성과

- Google Lighthouse 성능 측정을 통해 모든 지표 100점 달성
- SEO 및 웹 접근성(a11y)에 대한 이해 증가

TaskFlow

2024.12 - 2025.04 | 3인 프로젝트

- 복잡한 프로젝트 관리 도구들의 핵심 기능만을 추려 직관적인 웹 기반 협업 플랫폼으로 구현한 애플리케이션입니다.
- 칸반 보드, 작업 관리, 팀 협업 등 필수 기능들을 심플한 UI/UX로 제공하여 누구나 쉽고 간편하게 이용할 수 있습니다.

기술 스택

TypeScript, React.js, Vite, TanStack Query, Zustand, Tailwind CSS, MongoDB, Notion

담당 업무 및 성과

1. Google 소셜 로그인 및 인증 아키텍처 설계

이슈

- 빠르고 안전하게 적용이 가능한 로그인 기능이 필요
- API 요청 시 인증 토큰을 매번 수동으로 주입해야 하는 관리 구조 개선

해결

- Google OAuth2.0 기반 소셜 로그인 기능 구현
- 인증 로직을 중앙화하고 API 클라이언트에 인터셉터를 추가해 헤더에 자동으로 토큰 삽입되도록 개선

성과

- 회원가입 절차 간소화를 통해 초기 사용자 접근성 및 편의성 향상
- 인증 로직을 API 클라이언트에 통합하여 반복 코드 제거, 유지보수성 향상

2. Project/Task 생성 모달 및 데이터 유효성 검증

이슈

- 프로젝트/タ스크 생성 과정에서 반복적인 입력 UI 개발 발생
- 반복적인 검증 로직 발생으로 인한 비효율성 개선 필요

해결

- 재사용 가능한 입력 컴포넌트(Modal, Input, Datepicker 등)를 설계하여 반복 코드 제거
- UI에 종속되지 않도록 유효성 검증 로직을 별도 Hook으로 분리하여 재사용성 향상

성과

- 재사용 가능한 컴포넌트와 분리된 검증 로직을 통해 중복 코드 제거 및 개발 효율성 향상

3. 우측 패널 개발

이슈

- 칸반 보드에서 각 Task를 바로 수정할 수 있는 기능이 필요

해결

- 페이지 전환 없이 Task를 바로 편집할 수 있는 우측 패널 구현
- **Zustand**를 사용해 칸반 보드와 패널의 데이터를 실시간으로 동기화

성과

- 페이지 이동 없이 바로 Task를 수정할 수 있는 우측 패널을 구현하여 사용자 경험(UX)을 개선
- **Zustand**를 활용한 실시간 데이터 동기화로 보드와 패널 간의 사용성을 증가

4. 구성원 관리 및 권한 제어

이슈

- 프로젝트 내에서 단순한 관리자/멤버 구분을 넘어, 읽기 전용, 댓글 작성 등과 같은 세분화된 권한 관리 기능이 필요
- 구성원 추가/삭제 시 즉각적으로 반영되지 않아 사용자 경험을 저해하는 문제가 발생

해결

- 구성원을 추가/수정/삭제할 수 있는 관리 탭 구현
- 낙관적 업데이트(Optimistic Update)를 통해 사용자 경험(UX) 개선
- 데이터 동기화를 위해 캐시 무효화 전략을 적용하고, 실패 시 룰백 로직을 통해 일관된 상태 유지

성과

- 구성원별로 역할을 세분화하여 프로젝트 협업의 효율성 증가
- 낙관적 업데이트와 룰백 구조를 통해 즉각적인 반응성과 안정적인 사용자 경험(UX)을 제공

Education

충실파워

2016.03 - 2022.08 | 졸업 | 컴퓨터학부

Certifications

정보처리기사

2022.01 | 한국산업인력공단

Links

Portfolio : <https://uihyeon-portfolio.vercel.app/>

Blog : <https://kuh97.tistory.com/>

GitHub : <https://github.com/kuh97>

LinkedIn : <https://www.linkedin.com/in/%EC%9D%98%ED%98%84-%EA%B9%80-64b608335/>