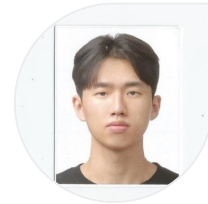


# 박현우

## 신입 프론트엔드 개발자



연락처 +82 01056863892

이메일 pq120wal39@gmail.com

주소 경남 창원시 진해구 천자로 386 창원 마린 푸르지오 1단지

웹이라는 도화지에 vscode라는 붓을 잡고 코드라는 물감을 찍어 하얀 도화지를 채워나가 만든 그림을 사람들에게 보여주는 것이 즐겁고 뿌듯한 프론트엔드 개발자 박현우입니다.

수학에 항상 정답이 있듯이 개발에서도 정답을 찾기위해 끊임없이 고민하여 정답에 가까워지도록 노력하는 것을 좋아하고 어려운 문제에 직면했을 때 그 문제를 해결하기 위해 그 문제를 집중하여 생각하고 해결책을 찾는 것을 좋아합니다.

---

## 자기소개

- 무작정 개발하기보다는 어떻게 설계하면 유지보수가 수월하고 재사용성이 높은지를 스스로 끊임없이 생각하며 개발합니다. 팀프로젝트를 진행하면서 기간안에 설계하는 것도 중요하지만 재사용성이 높은 코드를 만드는 것을 제일 우선시 하였습니다.
- 개발을 했다고해서 끝이아니라 이게 최선의 방법일까라는 고민을 가져, 제 코드를 의심하고 질문하여 정답이라는 단어에 가까워지도록 노력합니다.
- 맡은 일에 대한 책임을 가지고 임하며, 어려운 상황에서도 끝까지 포기하지 않고 문제를 해결해왔습니다.
- 처음 배우는 것이라도 여러가지 매체를 통해 빠르게 지식을 습득하여 활용할 수 있습니다. 기술의 더 깊은 이해를 원할 때는 기술 페이지의 문서를 보고 적용해 보는 것을 좋아합니다.

---

## 기술 스택

HTML/CSS, JavaScript, TypeScript, React, react-query, next.js13, tailwind-css, GitHub, Figma

---

## 포트폴리오

## URL

## 프로젝트

### YOO.GG

개인

2024.08. ~ 진행 중

## 리그오브레전드의 다양한 정보를 제공하는 서비스입니다.

[깃허브](#) | [배포 링크](#) | [디자인](#) | [RIOT API](#)

RIOT API에 웹 정식 등록 요청을 하여 승인을 받았습니다.

### 핵심 사용 기술

[Next.js14](#), [Tailwind-css](#), [React-query](#), [Typescript](#)

### 핵심포인트

- Context API를 활용해 modal, tooltip상태를 전역으로 관리함으로써 애플리케이션의 상태 관리가 더욱 용이해지고 일관성이 유지하였고, 또한 전역 상태를 활용하면 컴포넌트 간의 props 전달을 줄이고, 각 컴포넌트에서 modal, tooltip을 손쉽게 호출할 수 있도록 구현했습니다.
- Next.js의 Route Handlers를 사용하여 지정된 경로에 대한 사용자 지정 요청 처리기를 구현하였습니다.
- 챔피언 데이터, 데이터 버전, 룬 데이터 등 웹 페이지에 필요한 많은 정적 데이터를 서버 컴포넌트에서 관리함으로써, 클라이언트에서의 로딩 시간을 단축시키고, 데이터 일관성을 보장하였습니다.
- [useModal](#), [useOutsideClick](#), [useTooltip](#), [useIntersectionObservRef](#) 등 커스텀 훅을 통해 로직의 중복을 최소화하고, 코드의 재사용성을 높였습니다.
- [useInfinityQuery](#)를 활용하여 더보기 버튼을 통해 사용자에게 끊김 없는 전적 검색 경험을 제공하였습니다. 이로 인해 사용자는 필요에 따라 데이터를 계속해서 불러올 수 있으며, 직관적인 인터페이스를 통해 탐색할 수 있습니다. 또한, 초기 데이터 불러오는 과정에서 Skeleton UI를 적용함으로써, 로딩 중에도 사용자에게 매끄러운 시각적 피드백을 제공하여 불편함을 최소화했습니다.
- [metaData](#)를 적극 활용하여 SEO 최적화를 구현함으로써, 웹 애플리케이션의 가시성과 검색 엔진 순위를 크게 향상시켰습니다. 또한 각 페이지에 적절한 제목, 설명, 키워드, OG 태그 및 기타 메타 정보를 설정하여 사용자가 검색 결과에서 보다 쉽게 해당 페이지를 발견할 수 있도록 하였습니다.
- 다크 모드를 구현함으로써 사용자에게 보다 편리하고 쾌적한 경험을 제공하였습니다.
- [next-intl](#)를 활용하여 다국어 지원을 구현함으로써, 사용자에게 보다 포괄적이고 접근 가능한 경험을 제공하였습니다.

## PORTFOLIO

개인

2024.09. ~ 진행 중

## 저의 개인 포트폴리오 프로젝트입니다.

[깃허브](#) | [배포링크](#) | [디자인](#)

### 핵심 사용 기술

JavaScript

#### 핵심포인트

- JS의 `class` 문법을 사용하여 React의 Component 기반 아키텍처를 흉내 내어, 코드의 재사용성과 유지보수성을 높였습니다.
- 미디어 쿼리와 유연한 레이아웃을 활용해 다양한 디바이스 환경에서도 최적의 사용자 경험을 제공할 수 있도록 반응형 웹을 구축하였습니다

## Pickle Time

코드잇(팀 프로젝트)

2024.05. ~ 2024.06.

## 피클타임(Pickle Time)은 뜻이 맞는 사람들과 함께 자투리 시간을 가치 있게 만들어주는 자기개발 플랫폼입니다.

피클타임이란? 피자를 먹을 때 아주 잠깐의 시간동안 피클을 먹습니다. 이 피클 먹는 시간은 짧지만 피자와의 맛의 조화를 이루어주는 아주 중요한 시간입니다. 이처럼 피클타임은 피클 먹는 시간과 같이 짧은 시간을 가치 있는 시간으로 만들어주는 뜻입니다.

[깃허브](#) | [디자인](#)

서버 비용 문제로 서버 배포가 중단되었습니다.

### 핵심 사용 기술

React , Styled-component , Typescript , React-query , Zustand

#### 인원

프론트엔드(5명), 디자이너(1명)

### 맡은 역할

- 공통 Bottom Sheet Modal을 개발함으로써 사용자에게 일관된 UI 경험을 제공하였습니다. 이 모달은 다양한 상황에서 재사용 가능하도록 설계되어, 애플리케이션 내에서 여러 컴포넌트에서 쉽게 호출할 수 있는 구조를 갖추었습니다. Zustand를 활용하여 모달 상태를 전역으로 관리함으로써, 상태 관리의 복잡성을 줄이고, 컴포넌트 간의 효율적인 데이터 흐름을 유지할 수 있었습니다. 이를 통해 모달의 열림과 닫힘 상태를 중앙에서 관리하며, 다양한 컴포넌트에서 일관된 상태를 유지할 수 있었습니다.
- 메인 페이지의 인기 급상승 피클 UI, 마감 임박 피클 UI를 공통 컴포넌트로 개발하였습니다. 공통 컴포넌트인만큼 재사용성을 중요시하였으며 React의 Suspense와 ErrorBoundary를 직접 구현하여 에러가 발생하거나 로딩 중일때 사용자의 불편을 최소화 하였습니다.

### 3. 피클을 신청하거나 피클을 생성할 때의 마지막 단계인 결제 페이지의 컴포넌트들을 구현하였습니다.

결제하는 디자인은 같기 때문에 범용적으로 사용할 수 있는 이름으로 컴포넌트를 네이밍 하였습니다. 결제 로직도 구현하고 싶었지만 당시 트러블 슈팅으로 저가 구현하지 못해 큰 아쉬움이 남았습니다.

### 4. 내 피클 페이지에 대한 컴포넌트들을 구현하였습니다. 내 피클 카드는 현재 날짜에 맞춰 진행전, 진행중, 종료 중 하나의 상태를 갖게 되어 각 상태마다 UI가 조금씩 다르지만 전체적인 디자인 틀은 같기 때문에 하나의 컴포넌트로 만들어 각 상태에 따라서 그에 맞는 UI가 랜더링 되도록 구현하였습니다.

---

## 온라인 아르바이트, 사장님 매칭 서비스[danchoo]

코드잇(팀 프로젝트)

2024.04. ~ 2024.05.

## 가게에 급한 일손이 필요한 경우 더 높은 페이를 주고 일손을 매칭시켜주는 서비스입니다.

[깃허브](#) | [배포링크](#)

### 핵심 사용 기술

Next.js , tailwind-css , React-query , Typescript

### 인원

프론트엔드(5명)

### 맡은 역할

- 페이지 FLOW를 설계에 앞장섰습니다..** App router를 사용하여 프로젝트 개발에 앞서 페이지 구조를 명확히 정리함으로써 App router의 폴더구조를 수월하게 설계하여 개발을 시작하였습니다.
- 페이지네이션 컴포넌트를 구현하였습니다.** 재사용되는 컴포넌트이고 로직이 중요한 컴포넌트라고 생각하여 페이지네이션 구현 로직은 따로 분리하여 개발을 하였습니다.
- 메인 페이지의 필터 컴포넌트를 구현하였습니다.** 원래는 버튼과 필터 팝업을 분리하고 상태를 공유하는 식으로 개발을 진행하려고 하였으나 다른 부분에서 사용하는 점이 없다는 것을 고려하여 버튼과 필터 팝업을 합쳐 하나의 컴포넌트로 개발을 진행하였습니다.
- 가게 정보 등록 폼 페이지와 가게 정보 수정 폼 페이지를 개발하였습니다.** 폼에 들어가는 Input같은 경우 다른 팀원이 만든 공통 Input컴포넌트를 사용하였으며 수정이 필요하다고 생각하는 부분은 팀원과 소통을 통해 수정하였습니다. 또한 이미지 업로드는 백엔드에 나와있는 AWS를 이용한 이미지 업로드를 사용하였습니다.
- 내 공고 상세 페이지를 개발하였습니다.** 이 페이지의신청자 목록 테이블은 내 프로필의 신청 내역에도 사용되고 있어 공통 컴포넌트로 분리하여 진행하였습니다. 또한 테이블의 상태 열에 들어가는 부분은 따로 로직이 들어가야하는 부분이라고 판단하여 컴포넌트로 분리하여 진행하였습니다.
- 내 프로필 등록 폼 페이지와 수정 페이지를 개발하였습니다.** 상태를 기반으로 하여 한 페이지로 구현할 수도 있지만 내 프로필을 등록하는 것과 수정하는 것은 다른 로직이라고 판단하여 각각 다른 페이지로 구현하였습니다.

## Rolling

코드잇(팀 프로젝트)

2024.02. ~ 2024.05.

추억의 롤링페이퍼를 오프라인에서 쓰는 것이 아닌 온라인으로 간편하게 작성할 수 있는 플랫폼입니다.

[깃허브](#) | [배포링크](#)

### 핵심 사용 기술

React, styled-component, React-router-dom

### 인원

프론트엔드(5명)

### 맡은 역할

- 버튼 공통 컴포넌트를 개발하였습니다.
- 롤링 메시지 작성 폼 페이지를 개발하였습니다.React-quill을 사용하여 메시지 작성 폼을 구현하였습니다.  
또한, 완성된 메시지를 볼 수 있으면 사용자 경험을 향상 시킬 수 있다고 생각하여 메시지 폼을 작성완료하면 폼 제출전에 완성된 메시지 UI를 구현함으로써 사용자에게 직관적이고 편리한 메시지 작성 과정을 제공할 수 있었습니다.
- 롤링 페이퍼 리스트 페이지의 롤링 리스트 컴포넌트의 초기 로딩시 스켈레톤 UI를 적용함으로써 사용자에게 로딩시간에 대한 불편을 최소화 하였습니다.

## 교육

### 창원대학교

대학교(학사) | 수학과(평균 학점 : 3.73)

2017.03. ~ 2023.02. | 졸업

### 코드잇

사설 교육 | 프론트엔드 스프린트 과정

2023.12. ~ 2024.06. | 졸업

## 자격증

### 정보처리기사

한국산업인력공단

2024.09.

## 컴퓨터활용능력

1급 | 대한상공회의소

2022.06.

Powered by Rallit.