



황유빈 Yubin Hwang

안녕하세요.

환경을 빠르게 이해하고, 안정적으로 역할을 채워가는 개발자  
황유빈 입니다.

### 개인정보

생년월일 1999년 09월 03일

학교/전공 강남대학교/소프트웨어응용학부

전화번호 010-3418-8600

이메일 dbqls990909@naver.com

지원분야 Front-End 개발자

### 교육

#### 멋쟁이사자처럼 프론트엔드 스쿨 13기 2025.02 ~ 2025.08

- 기간: 2025.02.05 ~ 2025.08.08
- HTML/CSS, JavaScript, React/Next.js 단계적 심화 교육 이수
- 1rem, 반응형 UI 기반 개인 웹 퍼블리싱 프로젝트
- 7쪽 오락실, JS 기반 게임 인터랙션 구현, Netlify 배포
- 도토리섬, Next.js 전자상거래 서비스, Vercel 배포(팀 프로젝트)

### Links



## 자기소개

### 주어진 환경과 함께하는 개발자

살아온 과정에서 새로운 환경에 놓일 때마다, 먼저 그 안의 분위기와 흐름을 파악하며 적응해 왔습니다.

이러한 성향은 개발 프로젝트에서도 그대로 이어져, 팀 구성과 상황에 따라 직함보다 필요한 역할이 무엇인지부터 고민하게 되었습니다.

UI 프로젝트에서는 팀원으로서 맡은 구현을 책임졌고, JavaScript 프로젝트에서는 PL로서 코드 구조를 점검하며 개선 방향을 제시했습니다.

이후 최종 프로젝트에서는 PM 역할을 맡아 일정과 전체 흐름을 조율하며 팀을 이끌었습니다.

또한 Flutter, Vue, React, Next.js 등 서로 다른 개발 환경을 경험하며, 변화하는 기술 흐름 속에서도 빠르게 적응하고 역할을 수행해 왔습니다.

### 기록과 공유는 필수

이해한 내용을 그냥 넘기기보다, 정리해 두는 편입니다.

그래야 오래 기억에 남고, 필요할 때 다시 꺼내볼 수 있다고 느껴왔기 때문입니다.

이러한 습관은 개발을 시작한 이후에도 이어져, Notion에는 공부한 내용과 필요한 자료들을 정리하며 어떤 과정을 거쳐 해결했는지를 함께 기록해 왔습니다.

프로젝트를 진행하며 마주한 이슈나 트러블슈팅 경험은 Tistory에 정리해 두고, 필요할 때 다시 참고했습니다.

또한 JavaScript 학습을 위해 GitHub 기반 스터디를 운영하며, 매주 공부한 내용을 공유하고 발표하는 시간을 통해 스스로 이해한 내용을 점검해 왔습니다.

이 과정은 스스로 이해한 내용을 점검하는 동시에,

새로운 문제를 마주했을 때 이전 경험을 차분하게 떠올릴 수 있는 기반이 되었습니다.

### 문제를 구조로 바라보는 시선

프론트엔드 프로젝트를 진행하며, 눈앞의 현상만 고치는 방식에는 한계가 있다는 것을 느꼈습니다.

상태 변경이 잦은 화면에서 지연과 불필요한 렌더링 문제가 발생했을 때도,

단순한 코드 수정으로 대응하기보다 화면 흐름과 상태 관리 구조를 나누어 살펴보며 원인을 정리했습니다.

이후 상태 관리 방식을 조정하고 렌더링 흐름을 단순화하는 방향으로 개선을 진행했고,

그 결과 화면 응답 속도가 약 40~50% 개선되는 변화를 확인할 수 있었습니다.

이러한 경험을 통해 문제를 구조로 나누어 바라보는 접근이 기능 구현을 넘어,

서비스의 안정성과 이후 확장성에도 중요하다는 것을 체감했습니다.

## 프로그래밍 역량

### JavaScript / TypeScript

비동기 로직이 많은 환경에서 코드 흐름이 복잡해지는 문제를 겪으며 TypeScript를 도입했습니다.  
데이터 구조와 함수의 역할을 타입으로 정리해 협업 시 오류를 사전에 방지하는 데 활용했습니다.

### React / Next.js

페이지 성격에 따라 서버 컴포넌트와 클라이언트 컴포넌트를 구분해 구성했으며,  
렌더링 방식이 사용자 경험에 미치는 영향을 고려해 SSR과 CSR을 선택적으로 적용했습니다.

### HTML / Tailwind CSS

시맨틱 구조를 바탕으로 접근성을 고려한 마크업을 작성했고,  
컴포넌트 단위로 스타일을 관리해 반응형 환경에서도 일관된 UI를 유지했습니다.

### Git & 협업 도구

PR 기반 협업을 통해 변경 의도를 공유하며 작업했고,  
Vercel을 활용해 배포 환경을 관리했습니다.

### 상태 관리 (Zustand)

불필요한 리렌더링 문제를 해결하기 위해 전역 상태를 도입했고,  
상태 구독 범위를 나누어 인터랙션에 즉각 반응하는 구조를 만들었습니다.

### 비동기 통신

Fetch와 Axios를 상황에 맞게 사용하며 API 호출 로직을 분리했고,  
공통 응답 처리와 에러 흐름을 정리 했습니다.

## 학력

기간	소속	내용
2018.02 ~ 2024.08	강남대학교	소프트웨어 응용학부

## 팀 프로젝트

### 그리니(Greenie)

2023.03 ~ 2023.08

#### ▣ 프로젝트 상세

친환경 커뮤니티 기반 웹 서비스 | 팀 프로젝트

GitHub URL | <https://github.com/YouVin/greenie-frontend>

사용 기술 | Vue 3 · Quasar Framework · TypeScript

- 게시글 목록 추천순/최신순 정렬 및 페이지네이션
- 댓글/대댓글 CRUD 및 좋아요 인터랙션 기능 구현
- 게시글 상세 페이지 UI 및 작성자 정보 표시
- Quasar 기반 카드형 리스트 UI 제작
- Axios 활용 REST API 연동 및 데이터 처리

### 오늘의키워드(Today Keyword)

2022.07 ~ 2022.11

#### ▣ 프로젝트 상세

키워드 기반 교내 앞 맛집 추천 앱 | 팀 프로젝트

GitHub URL | <https://github.com/YouVin/TodayKeyword.FrontEnd>

사용 기술 | Flutter · Dart · Firebase · Provider 상태관리 · REST API 연동

- 현재 위치 기반 지도 화면 구현 (WebView + Geolocator)
- 검색 디바운싱 적용한 가게 검색 기능 개발
- 지도 마커 조회/생성 기능 및 API 연동
- 카테고리/키워드 필터 UI 구현
- 마커 클릭 시 상세 정보 바텀시트 표시

## 멋쟁이사자처럼 프로젝트

킹버거

2025.03 ~ 2025.03

### ▣ 프로젝트 상세

HTML/CSS 기반 반응형 레이아웃 UI 클론 프로젝트 | 팀 프로젝트

GitHub URL | <https://github.com/YouVin/greenie-frontend>

배포 URL | <https://1rem.netlify.app/>

사용 기술 | HTML, CSS, 반응형 웹

- 실제 서비스 UI를 참고하여 반응형 레이아웃 설계
- Flex/Grid 기반 레이아웃 구성 및 구조화
- 재사용 가능한 컴포넌트 스타일 정리
- 웹 접근성 및 사용자 경험(UI/UX) 고려

7짱오락실

2025.05 ~ 2025.05

### ▣ 프로젝트 상세

Vanilla JavaScript 기반 게임 웹서비스 | 팀 프로젝트

GitHub URL | <https://github.com/YouVin/JS-07-7zzang-Arcade>

배포 URL | <https://7zzang-arcade.netlify.app/>

사용 기술 | HTML, CSS, JavaScript · TypeScript

- 메인 홈 UI 제작 및 오락실 레트로 감성 디자인
- 키보드/마우스 입력 기반 메뉴 및 인터랙션 구현
- 제7우주 게임 페이지 제작 및 난이도/충돌 판정 로직 구성
- 점수 시스템 및 순위표 UI 제작
- 사운드 리소스 적용 및 게임 리스트 관리 기능 구현

## 멋쟁이사자처럼 프로젝트

도토리섬 2025.07 ~ 2025.08

### ▣ 프로젝트 상세

Next.js 기반 이커머스 웹 서비스 | 팀 프로젝트

GitHub URL | <https://github.com/YouVin/Final-10-console.10g>

배포 URL | [dotori-island.vercel.app/](https://dotori-island.vercel.app/)

사용 기술 | React, Next.js(App Router), TypeScript, Tailwind CSS, Zustand, react-hook-form

- 상품 리스트·상세, 장바구니, 주문까지 이어지는 쇼핑 플로우 설계 및 구현
- 장바구니 수량 변경에 Optimistic UI와 디바운싱 적용해 응답 속도 체감 개선
- 장바구니/주문 정보를 Zustand 전역 상태로 관리하고, 총 금액·배송비 계산 로직 구현
- 주문 페이지를 서버 컴포넌트, 클라이언트 컴포넌트로 분리해 데이터 패칭 구조화
- 결제 정보·배송지 입력을 react-hook-form으로 관리하며 유효성 검증 및 UX 개선