



정혜연 | 鄭慧娟 | Jung HyeYeon

전화번호 010-2277-6954 이메일 jhy20306@gmail.com 링크 [Portfolio](#) [GitHub](#)

사용자의 경험을 중심으로 웹 서비스를 개발해온 정혜연입니다.

단순히 기능을 만드는 것에서 그치지 않고, 속도 개선이나 SEO, 데이터 처리 구조처럼 서비스 품질을 높이는 부분까지 고민해왔습니다. 개발 과정에서 예상치 못한 오류가 발생해도 배포 환경과 로컬 환경을 비교하며 원인을 끝까지 추적해 해결했던 경험이 있습니다. 계획에만 머무르기 보다 '일단 실행해보자'는 태도로 빠르게 구현하고, 빠르게 검증하는 방식으로 성장해왔습니다.

입사 후에도 주어진 역할에 한정되지 않고, 서비스 전반을 이해하는 웹개발자로 기여하겠습니다.

경력

Conscience partners | 스타트업 SI&AI 개발사

정규직 - 2025.08 ~ 2026.01

Typescript, React, Next.js(App), Tanstack query, Zustand, Zod, Tailwind CSS, Socket.io, FCM, Cloudflare, Python, Django, Capacitor

● 웹 기반의 앱 개발 및 python 기반의 RPA 개발

- 총 6개의 프로젝트에 참여
- 웹 기반의 앱인 마펫즈와 뽀는 단독 진행으로 full-stack + application 개발 전반 담당
- RAG 프론트엔드 개발로 AI 프롬프트 처리 경험
- python, C++(AutoHotKey) 기반의 RPA 개발 - 90% 바이브코딩
- 1차 MVP 완성된 프로젝트 고도화 개발

마펫즈(MPZ) | 유기동물 입양 커뮤니티

2025.08 ~ 2025.10

Full-stack

Typescript, React, Next.js, Tanstack Query, Tailwind CSS, python, Django, Gradle

[웹사이트 링크](#)

● 웹·앱 서비스 풀스택 개발

- React / Next.js 기반 웹 서비스와 Capacitor 패키징으로 iOS, Android 앱 개발을 병행
- 프론트엔드 UI 구현부터 API 연동, 상태 관리, 서버 로직 일부까지 풀스택으로 참여
- 일반 유저, 센터(구독 / 미구독), 전문가, 훈련가 등 다양한 사용자 유형에 따른 조건 분기 처리 구현

● 복잡한 권한·상태 기반 기능 구현

- 사용자 권한 및 상태에 따라 화면 노출, 기능 접근, 데이터 처리 로직을 세분화
- 단순 분기 처리 수준이 아닌, 실제 운영 시 발생하는 케이스를 고려한 예외 처리 로직 설계
- 사용자 유형 증가에 따른 구조 복잡도를 고려하여 유지보수 가능한 형태로 정리

● QA 중심 개발 및 반복 검증

- 총 12차 이상 QA를 반복 수행하며 기능 안정성 및 예외 케이스 검증
- QA 과정에서 발견된 이슈를 단순 수정에 그치지 않고, 기능 구조 개선 및 추가 개발까지 병행
- QA → 수정 → 재검증 사이클을 반복하며 서비스 완성도 향상

● Firebase 기반 알림 기능 연동

- Firebase를 활용한 앱 푸시 알림(FCM) 기능 연동
- 사용자 상태 및 이벤트에 따른 알림 흐름을 고려하여 알림 구조 설계

● AI 기반 동물 매칭 기능 개발 경험

- 8~9가지 설문 응답을 기반으로 AI를 활용한 동물 매칭 기능(프론트엔드) 구현
- 사용자 입력 데이터를 바탕으로 매칭 결과를 도출하는 흐름까지 개발 완료
- 현재 서비스 정책 변경으로 기능은 주석 처리되었으나, 구조 및 로직 설계 경험 보유

뽀(BBU) | 미혼, 기혼, 이혼자들을 위한 매칭 어플리케이션

2025.10 ~ 2025.12

Full-stack

Typescript, React, Next.js,
Tanstack Query, Socket.io
Tailwind CSS, python,
Django, FCM, Capacitor

웹사이트 링크

● 웹·앱 서비스 풀스택 개발

- 서비스 기획 단계부터 참여하여 사용자 흐름을 기준으로 화면 구조와 기능 동선을 설계
- 단순 CRUD 수준이 아닌, 실시간 상호작용과 운영 로직이 결합된 서비스 구조 구현

● 서비스 로직 자동화 및 정책 경험

- 매칭 리스트를 매일 자정 자동 갱신하도록 스케줄링하여 운영 자동화
- 특정 회사 차단, 조건별 노출 제한 등 서비스 정책을 반영한 로직 구현
- 전체 회원 데이터를 기준으로 한 인기도 지표 산출 로직을 설계·적용

● 운영 서비스 관점의 안정화

- 토스페이 연동을 포함한 결제 기능 구현 및 결제 흐름 안정화
- 실제 서비스 운영 중 발생하는 이슈 대응 및 기능 개선을 통해 서비스 완성도 유지
- 기능 추가, 정책 변경, 안정화 작업을 반복하며 서비스 완성도 유지

RPA | 손해사정사의 단순 업무 보조를 위한 RPA

2025.10

Python

Python, C++, Typescript, React,
Next.js, Tailwind CSS

● 손해사정 업무 자동화 프로그램 설계 및 운영

- 손해사정 업무 전반에서 반복적으로 발생하는 조회·이동·입력 작업을 분석하여, 실제 현업에서 사용 가능한 자동화 프로그램을 설계·개발
- 고객사 내부 보안 정책으로 인해 Python 기반 자동화가 차단되는 환경에서, AutoHotkey(AHK) 기반으로 자동화 방식을 전환하여 재설계
- 번들번호를 기준으로 여러 작업 화면을 동시에 확인해야 하는 업무 특성을 반영하여, 멀티 창 중심의 자동화 흐름을 구현

● 보안·사용성 제약을 고려한 현실적인 자동화 구현

- 완전 자동화가 어려운 환경에서 사용자 입력 이벤트를 활용한 반자동화 방식을 적용하여 실무 활용도를 확보
- 다양한 자동화 접근 방식을 검토하고, 보안·안정성·사용성을 종합적으로 고려해 운영 가능한 최적의 방식 선택
- 실제 직원 대상 배포 및 사용성 테스트를 통해 현업 피드백을 반영하며 자동화 흐름을 지속적으로 개선

RAG

2025.10

Frontend

Typescript, React, Next.js,
Tanstack Query, Tailwind CSS

● RAG 기반 AI 서비스 프론트엔드 개발

- RAG 구조의 AI 서비스를 대상으로 프론트엔드 전반을 설계·구현
- AI 프롬프트 입력, 실시간 응답 스트리밍 처리 등 **AI 응답 흐름을 고려한 UI 로직** 구현
- 백엔드 및 모델 영역은 제외하고, 사용자와 AI 사이의 상호작용에 집중된 프론트엔드 개발 수행

● AI 응답 관리 및 프롬프트 커스터마이징 기능 구현

- AI가 생성한 응답을 비교·관리할 수 있는 UI를 구성하여 결과 분석이 가능하도록 설계
- 개인별 프롬프트 커스터마이징, 버전 기록 및 업데이트가 가능한 구조로 구현
- 반복 테스트와 개선이 필요한 AI 서비스 특성을 고려하여, 프롬프트 관리 중심의 화면 구성

● 사용량 기반 데이터 시각화 및 서비스 확장성 고려

- 사용자 및 프로젝트별 **AI 사용량**을 기준으로 다양한 **그래프 시각화 UI** 구현
- 별도의 **UI/UX 기획이 없는** 상황에서, React 기반 그래프 라이브러리를 활용해 화면 흐름을 자체 설계
- 프로젝트 단위로 서비스를 분리하여, 특정 기업에 특화된 **AI 서비스로 활용 가능한 구조**로 구현

CodeStation | 스타트업 외주개발사(웹 에이전시)

인턴 - 2025.05 ~ 2025.06

Tetherdrop 풀스택 개발

Typescript, React, Next.js(App),
Server Actions, Zustand, Zod,
Tailwind CSS, Framer-motion,
Prisma, Supabase

웹사이트 링크

● Prisma/Server Actions 기반 데이터 관리

- **Prisma 스키마** 활용 findMany·create·update·delete 등 CRUD 쿼리 구현
- orderBy, skip/take를 활용한 **페이징 처리 및 관계 매핑** 로직 작성
- **Server Actions**를 적용하여 클라이언트 폼 제출 시 서버에서 실시간 Prisma 쿼리 실행
- **데이터 흐름/모델링/관리 전체를 주도적으로 담당**

● API 통신 구조화 및 상태 관리

- useState로 출금 현황을 **실시간 폴링 및 n초 주기 자동 갱신** 구현
- Zustand 전역 스토어(useWithdrawStore, useUIDStore)로 컴포넌트 간 데이터 공유 최적화
- 서비스 레이어 분리로 비즈니스 로직 모듈화, **각 최상단 스크립트는 서버 사이드 실행 구조**로 설계

● 단계별 구조 페이지 작업

- 단일 컴포넌트와 useState로 **단계별 구조를 관리**, 재접근 시 초기 단계 진입 및 상태 복원 설계
- 단계별 입력값을 **Zustand 전역 스토어**에 저장하고, **Zod로 유효성 검증**하여 데이터 무결성 보장
- **각 단계별 UI를 독립 컴포넌트로 모듈화**하여 유지보수성 강화
- 1차 완성 후 QA리스트 기반으로 누락 기능, UI/UX, 데이터 표출 등 **주요 항목을 실제 점검 및 개선**

프로젝트

GlobalNomad | 취미/소모임 활동을 주치 및 참여할 수 있는 웹 플랫폼

2025.03 ~ 2025.04

팀프로젝트 | Frontend | 총 6명

Typescript, React, Next.js(App),
Tanstack query, Axios, Zustand,
Module CSS, Storybook,
Framer-motion

[Github](#)

[배포 링크](#)

● 프론트엔드 아키텍처 개선 및 인증 기반 API 구조 설계

- React 기반 예약 현황 시각화 및 상태 중심 UI 구성, 복잡한 모달 내 다중 endpoint 구조를 단순화 하며 유지보수성 개선
- useState 중심 로직을 React Query 기반 캐시 구조로 리팩토링하여 상태 흐름 안정화 및 사용자 경험 개선
- **Lighthouse** 분석을 기반으로 성능 최적화 진행, 데스크탑·모바일 점수 90점 이상 달성
- React Query 커스텀 훅 설계 및 QueryKey/QueryFn 구조 통일로 API 로직 추상화 및 재사용성 강화
- JWT 인증 + Axios instance 기반 자동 토큰 처리 구조 구현, SSR prefetch 과정에서 인증 누락 이슈를 분석하며 SSR/CSR 인증 처리 차이 이해
- 팀장으로서 초기 세팅, 기술 스택 선정, 코드 컨벤션 및 협업 기준 수립 등 프로젝트 전반 리딩

WHYNE | 와인에 대해 소통할 수 있는 웹 커뮤니티

2025.01 ~ 2025.02

팀프로젝트 | Frontend | 총 5명

Typescript, React, Next.js(Page),
Restful API-Axios, module CSS

[Github](#)

[배포 링크](#)

● 프로젝트 기반 설계 및 사용자 경험 개선

- 초기 환경 구성, 폴더 구조, Lint 설정 등 프로젝트 기반 세팅 주도 및 개발 흐름 정립
- 팀원들로부터 초기 구조 설계에 대한 긍정적 피드백을 받으며 **비공식 리더십 경험**
- 공통 Button 컴포넌트 및 Global 스타일 설계, 타입·사이즈 옵션화로 재사용성과 유지보수성 강화
- 마이페이지 UI 구현 및 삭제 확인 모달 도입 등 사용자 실수 방지 중심 UX 개선
- 와인 리뷰 수정 기능 및 상세 페이지 연동 구현, 데이터 전달 구조 개선 및 진입 시 **refetch** 처리로 안정성 보완

Rolling | 익명/비익명으로 타인에게 메시지를 남길 수 있는 웹 커뮤니티

2024.11 ~ 2024.12

팀프로젝트 | Frontend | 총 5명

Javascript, React, Restful API,
Styled-components, Module CSS

[Github](#)

[배포 링크](#)

● 협업 인식 기반 컴포넌트 설계 및 폼 구조 개선 경험

- 팀 구성원으로 참여하며 협업 방식을 관찰하고, 이후 리딩 시 어려움을 선제적으로 공유하는 소통 방식의 중요성 체감
- 팀 내 심리적 거리감 해소와 명확한 커뮤니케이션 구조의 필요성 인식
- styled-components 기반 Button 컴포넌트 설계, 아이콘·형태를 props로 제어하여 재사용성과 확장성 확보
- 롤링페이퍼 생성 폼 구현, 필수 입력 검증 및 사용자 피드백 중심의 폼 처리 로직 설계
- 폼 구조화 미흡으로 인한 코드 중복 경험을 통해 폼 로직 공통화 및 커스텀 훅 분리 전략의 필요성 체감

스킬

Frontend

- React
- Next.js
- JavaScript / TypeScript
- HTML5 / CSS3

Application

- Capacitor
- FCM
- Socket.io

스타일

- Styled-components
- Tailwind CSS
- CSS Module

Backend

- Prisma
- Python
- Django

상태관리/Data

- TanStack Query
- Restful API
- Axios instance
- Zustand
- Server Actions

협업/기타

- Git / GitHub
- Slack / Notion
- Storybook
- Figma/XD

학력

아주대학교 | 졸업

2019.03 - 2025.02

(인문대학) 불어불문학과

(소프트웨어융합대학) (복)디지털미디어학과

외국어

영어 고급회화

프랑스어 기초회화

일본어 일상회화

DEL F B1