

✦ WELCOME TO THE

HYEON IVERSE

화면 너머의 동작까지 확인하는 프론트엔드 개발자

TypeScript · React · Next.js를 주력으로,
12개의 프로젝트가 궤도를 도는 김정현의 작은 우주입니다.

12

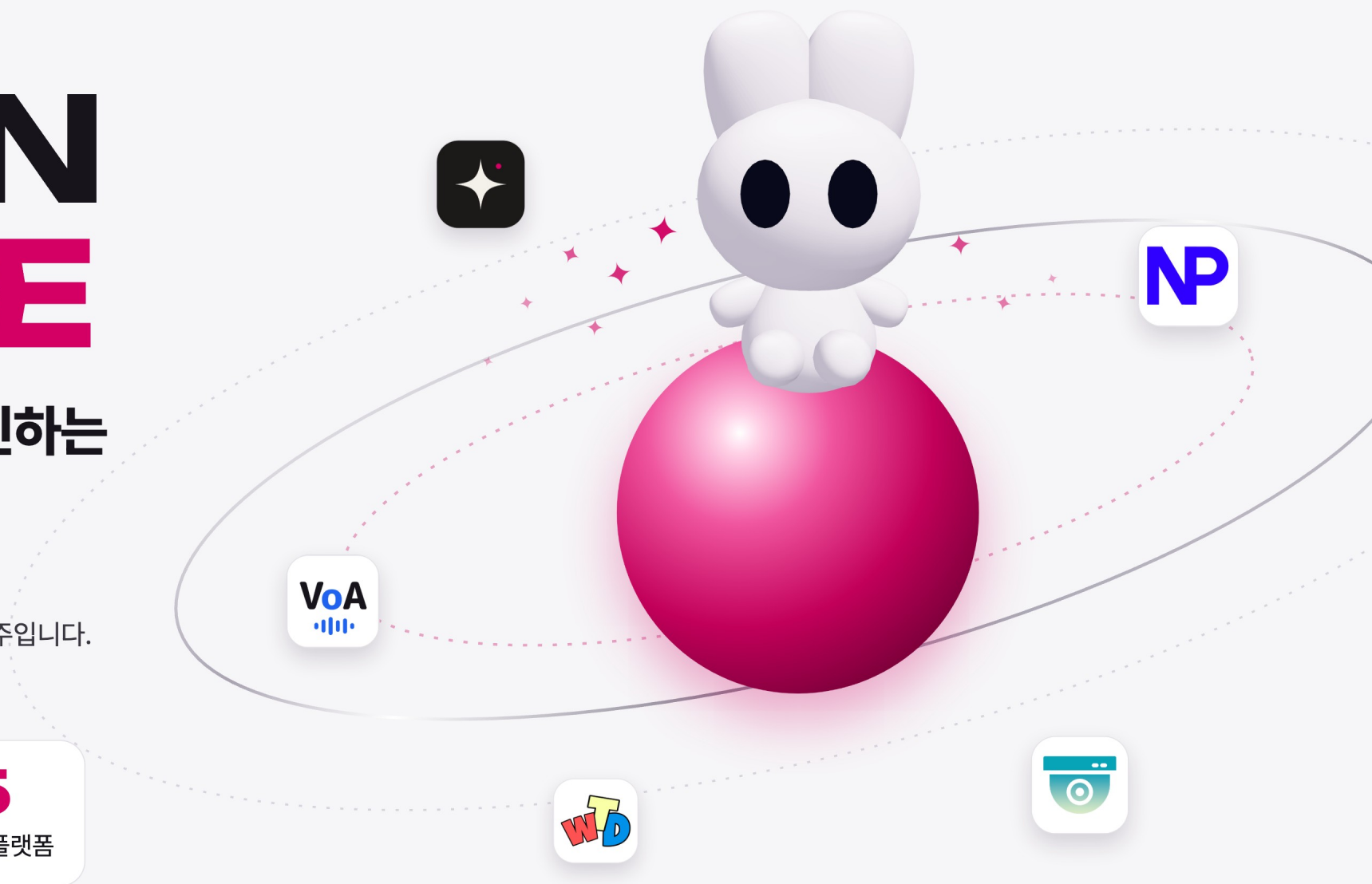
완료 프로젝트

05

배포 서비스

05

다뤄 본 플랫폼





MAIL hyeoniverse.dev@gmail.com

WEB hyeoniverse.com

GIT github.com/hyeoniverse

TEL 010-7191-2644

김정현

FRONTEND DEV

Kim JeongHyeon · 프론트엔드 개발자 · 서울 거주

화면을 만드는 데서 그치지 않고 API 연동·테스트·배포까지 끝내는 프론트엔드 개발자입니다. 데브코스에서 TypeScript 풀 사이클 개발을 익힌 뒤 Next.js로 개인 사이트를 운영 중이고, 학부 시절에는 Android·iOS·데스크톱·Unity까지 플랫폼을 가리지 않고 문제를 정의해 구현했습니다.

학력

서울여자대학교

정보보호학과 · 소프트웨어융합전공 복수 전공

2020.02 – 2025.02 졸업 · 3.9 / 4.5

졸업논문: 온라인 시험 부정행위 탐지 서비스 (2023 춘계학술대회)

일하는 방식 · 목표

AI와 함께 빠르게 만들고, 직접 검증합니다

초안과 반복 작업은 AI로 빠르게, 구조와 동작은 직접 읽고 테스트해 완성도를 끌어올립니다.

기록으로 협업합니다

결정 사항과 ERD·API 명세를 Notion과 GitHub에 남겨 같은 그림을 보게 합니다.

목표 — 화면 너머를 이해하는 프론트엔드 개발자

UI뿐 아니라 API·배포·운영까지 흐름을 알고, 프로젝트 전체에 책임을 지는 개발자로 성장하고 싶습니다.

주력 기술, 어디까지 써 봤는지



FRONTEND

01

언어 · 프레임워크 TypeScript · JavaScript · React · Next.js

NP 공통 컴포넌트 · 커스텀 **혹 NewPick**

반복 UI는 공통 컴포넌트로, 요청 로직은 커스텀 혹은 분리

🌟 리렌더링 최적화 **HYEONIVERSE**

작업물 편집기의 글자별 렌더링을 줄여 React 렌더 387→98ms

02

상태 · 데이터 React Query · Context · Axios · Zustand · Redux

NP 서버 / 클라이언트 상태 분리 **NewPick**

초기 데이터는 서버에서, 상호작용 데이터만 React Query로

🎨 전역 상태 · 데이터 동기화 **WTD**

Context로 상태 공유, 비로그인 할 일은 로그인 시 서버 동기화

03

UI · 인터랙션 CSS Modules · Styled-components · Tailwind · GSAP · Framer Motion

🌟 디자인 시스템 **HYEONIVERSE**

사이트 전체의 디자인 시스템을 설계하고 문서 페이지로 정리

📺 창 관리 인터랙션 **MacFolio**

창 드래그·리사이즈·포커스 순서를 직접 구현, 폰에선 홈 화면으로 전환

04

테스트 · 배포 Unit · E2E · Playwright · CI · Vercel · Cloudflare

🌟 CI · E2E 테스트 **HYEONIVERSE**

PR마다 자동 테스트, 운영 빌드 E2E로 라우트 35개 검증

📺 CI 파이프라인 **MacFolio**

lint·타입·테스트·빌드·E2E를 한 줄로, 데스크톱·iPhone·Android 3종 검증

프로젝트에서 직접 다뤄본 기술



EXPERIENCE

백엔드 · DB NestJS · Express · Prisma · Supabase · PostgreSQL · MySQL · MongoDB

NP 인증 API 수정 NewPick

NestJS·Prisma 관심사 API 3개를 직접 수정

API 서버 구축 MacFolio

NestJS·Prisma로 스키마와 API를 직접 설계,
PostgreSQL 운영

모바일 Android (Java) · Swift · UIKit · SpriteKit · Firebase

iOS 화면 구성 모두의 키오스크

CollectionView·TableView로 메뉴 화면 구성

인터랙션 · 인증 PurrFectday

SpriteKit 고양이 인터랙션, Firebase 인증 연동

데스크톱 · 게임 PyQt5 · Unity · C# · WebGL

데스크톱 GUI Voice of All

PyQt5로 음성 안내 열차 예매 전 과정 구성

게임 로직 Sprout Farm

무한 타일맵과 A* 경로 탐색 AI, WebGL 빌드와
서버리스 랭킹까지 배포

AI · 데이터 MediaPipe · PyTorch · HuggingFace · Pandas

컴퓨터 비전 ChasEye

MediaPipe 얼굴 수·홍채 비율로 시선 이탈 판별

자연어 분류 악성댓글 분류

KcELECTRA 파인튜닝, 분류 정확도 92%

협업 Notion · Slack · GitHub · Postman

NP 문서 공유 NewPick

ERD·API 명세를 Notion·Postman으로 팀과 공유

NP 개발 프로세스 NewPick

이슈·PR 템플릿을 만들어 백엔드 저장소까지 적용

수상·자격·활동

수상·자격·어학

2026.09	TOEIC 775 영문 기술 문서 독해 가능
2024.06	정보처리기사 한국산업인력공단
2023.12	프로보노 공모전 동상 수상 과학기술정보통신부 주최 · IITP·한국정보방송통신대연합 주관 · Voice of All
2022.12	졸업 프로젝트 평가회 우수상 수상 서울여대 정보보호학과 · ChasEye
2022.11	TOEFL iBT 83

교육·활동

2026.02 - 현재	HYEONIVERSE 개인 사이트 개발·운영 Next.js 16 · React 19 · Supabase · Vercel
2024.08 - 현재	MacFolio 개발·운영 React 19 · NestJS · PostgreSQL · Cloudflare · Oracle Cloud
2025.10 - 2026.06	가족 카페 운영 영업 정상화·인력 채용, 월매출 900 → 2,100만 원
2024.08 - 2025.02	K-Digital Training 프로그래머스 데브코스 수료 타입스크립트로 함께하는 웹 풀 사이클 개발 · (주)그렙 · 875시간
2023.09 - 10	AI+Security 아이디어 공모전 KISA · 본선(발표평가) 진출 · 그린라이트
2023.04 - 05	춘계학술대회 논문 게재 — ChasEye 온라인 시험 부정행위 탐지 서비스
2023.04 - 11	ICT멘토링(한이음) — Voice of All 3인 팀 · 시각장애인용 키오스크 SW
2023.01 - 03	GDSC Solution Challenge 4인 팀 · ChatBuddy Android 앱

문제를 정의하고, 동작하는 화면으로 답합니다.

01 문제 정의

사용자의 불편을 듣고
요구사항으로 정리합니다

Voice of All 전맹 시각장애인을 직접 인터뷰해
요구사항을 분석하고, 반영·미반영 이유까지 문서
로 기록

NewPick 관심 분야 뉴스를 받아 보는 흐름을 기
준으로 메인·리스트·상세·마이페이지 구조 설계

모두의 키오스크 키오스크가 낯선 50대 이상이 집
에서 연습할 수 있도록 실제 주문 흐름을 그대로
재현

02 동작

외부 시스템과 연결해
실제로 동작하게 만듭니다

NewPick 백엔드(NestJS) API와 통신하는 프론
트엔드 API 모듈 9개 중 8개를 작성해 구독·북마
크·마이페이지 기능 연결

Voice of All TAGO 열차정보 API를 단계별로
호출하고, Google STT로 받은 음성 입력을 예매
흐름에 연결

ChatBuddy ChatGPT API로 상담 응답을 받
고, 위험 표현이 감지되면 지원 안내로 분기.
Firebase로 Google 로그인 연동

03 화면

사용자가 쓰는
화면으로 완성합니다

NewPick · QRU 다크모드와 PC·모바일 반응형
으로 어떤 기기에서도 같은 화면 구성 유지

HYEONIVERSE KO/EN 다국어와 스크롤 인터
랙션을 넣으면서도 Lighthouse 성능 94 · 접근성
96 유지

PurrFectday 2인 팀에서 iOS 구현을 주도,
UIKit·SpriteKit으로 투두와 고양이 인터랙션 화
면 제작

✦ SYSTEM MAP

프로젝트 한눈에

MAIN 안쪽 궤도 · 4

직접 기획·구현한 웹 프로젝트. 트러블 슈팅까지 깊게.

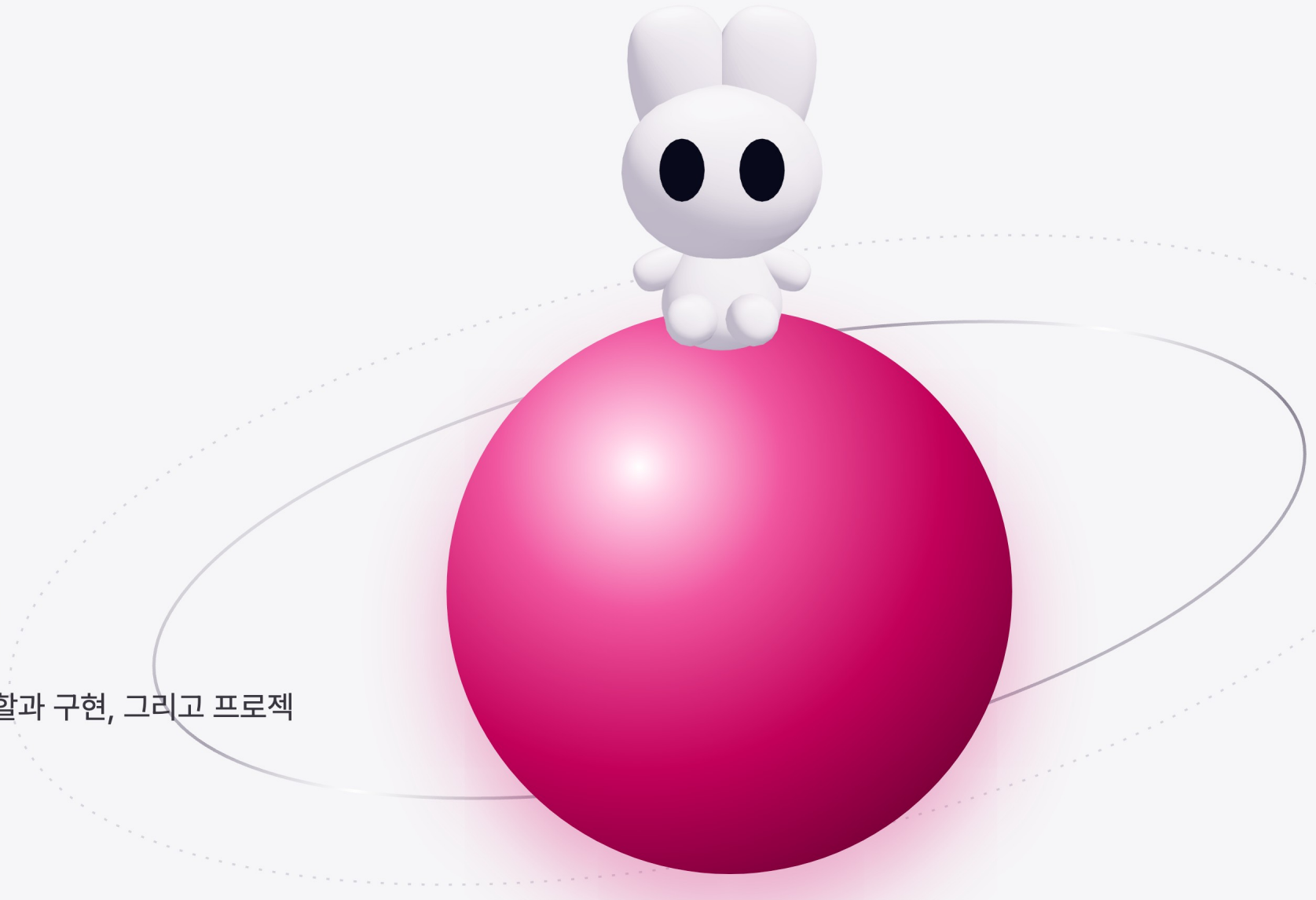
SIDE 바깥 궤도 · 8

데스크톱·모바일·게임·AI와 수상작. 핵심만 가볍게.



주요 프로젝트

직접 기획하고 구현한 네 개의 웹 프로젝트. 역할과 구현, 그리고 프로젝트마다 트러블슈팅을 함께 정리했습니다.





MAIN · ORBIT 01

개인 · 2026.02 – 운영 중 · hyeoniverse.com

HYEONIVERSE

혼자 만든 서비스에서 성능·검증을 어디까지 끌어올렸는가

흩어진 산출물을 한 곳에서 보여 주고 계속 갱신하기 위한 개인 포트폴리오 사이트. 공개 사이트와 관리자 CMS를 한 저장소에서 함께 운영하며, 디자인 시스템부터 배포까지 직접 설계·구현.

담당 기획 · 디자인 · 프론트엔드 · 배포 전체

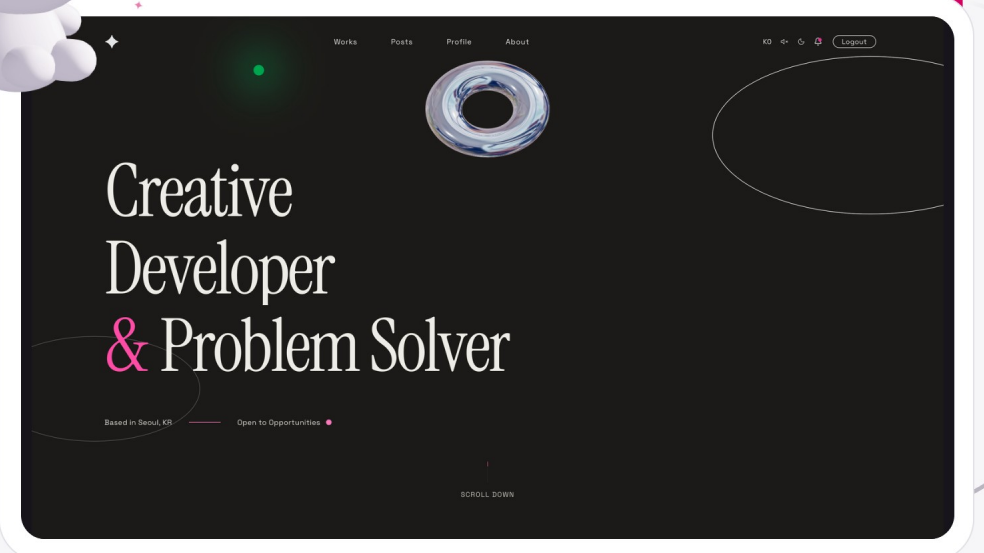
기술 Next.js 16 · React 19 · TypeScript · CSS Modules · Plate.js · Framer Motion · GSAP · Three.js · Supabase · Vercel

지표 Lighthouse **98** · LCP **1.9초** · 초기 번들 **449KB**

- 01** 페이지 Home · Works · Posts · Profile · About, Works 레이아웃 6종
- 02** 디자인 시스템 3계층 토큰(Raw → Semantic → Component) · 색은 전부 OKLCH · stylelint로 강제
- 03** 에디터 Plate.js 기반 CMS — 다이어그램 · 코드 실행 · 수식 · .md 양방향 동기화 · 개정 이력
- 04** 인증·권한 Supabase RLS 4단계 권한 · 로그인 5회 실패 시 잠금 · 새 기기는 이메일 승인
- 05** 검증 PR마다 타입 검사·린트·단위 테스트를 CI에서 자동 실행, E2E는 운영 빌드로 따로 돌려 공개 19 · 관리자 16개 라우트 확인



01



↳ 메인 화면 — Creative Developer & Problem Solver 히어로

배운 점

혼자 개발하면 코드 리뷰를 받을 수 없어, PR마다 타입 검사·린트·단위 테스트가 자동으로 돌게 하고 운영 빌드로 E2E를 따로 실행했습니다. 변경 후 기존 기능이 깨지지 않았는지 확인하는 과정을 개발 과정에 포함시켰습니다.

HYEONIVERSE — 글자 하나에 무거워지던 편집기

작업물 편집기에서 글자를 칠 때마다 입력이 밀림 — 원인 세 개를 따로 찾아 따로 해결



PROBLEM

① 섹션 전체 리렌더

글자 하나에 섹션이 통째로 다시 그려짐

② 배열을 매번 새로 생성

첨부 이미지 배열을 글자마다 새로 만들어 하위가 바뀐 것으로 인식

③ CSS :has(:hover) 선택자

입력 51번 중 10번, 요소 6천여 개를 재계산



SOLVE

섹션 메모이제이션

바뀐 섹션만 다시 그려지게 함

참조 유지

내용이 같으면 같은 배열을 넘김

선택자 수정

페이지 전체 재계산이 일어나지 않게 함



RESULT

입력 지연 p90

120→72ms

React 렌더 (51자 합계)

387→98ms

페이지 전체 스타일 재계산

10→0회

입력 지연: CPU 4배 감속 · 51자 입력 · 3회 중앙값

배운 점

느린 원인이 React 렌더링에만 있지 않았습니다. 렌더와 스타일 계산을 나눠 측정하고 나서야 CSS 선택자까지 찾을 수 있었습니다.

HYEONIVERSE — 이미지를 받고도 보이지 않던 첫 화면



PROBLEM

원인 두 개가 직렬로 걸려 있었음

- ① useSearchParams 때문에 Suspense 경계가 프리렌더에서 빠져, 서버 HTML에 배너 없음
- ② 이미지를 받아도 하이드레이션 뒤까지 opacity 0으로 가려짐



SOLVE

두 원인을 모두 풀어야 수치가 움직임

배너가 서버 HTML에 포함되도록 경계를 나누고, 로드 전까지 이미지를 가리던 처리 제거. 하나만 고쳤을 때는 LCP가 거의 줄지 않음



RESULT

LCP 7.8초 → 2.9초

BEFORE

이미지 다운로드 2.2초

받았지만 opacity 0 — 하이드레이션 뒤 onLoad까지 가려짐

LCP 7.8초

AFTER

LCP 2.9초

0s

2s

4s

6s

8s

배운 점

원인이 여러 개 겹쳐 있으면 하나만 고쳐서는 수치가 움직이지 않습니다. 수치가 그대로라고 해서 고친 것이 틀린 것은 아니었고, 남은 원인을 계속 찾아야 했습니다.



MAIN · ORBIT 02

5인 팀 · 2024.12 – 2025.02

NewPick

팀에서 백엔드와 조율하며 인증·데이터 흐름을 해결한 경험

관심 분야를 고르면 최신 뉴스를 수집·AI 요약해 메일로 보내 주는 뉴스레터 웹 서비스. 기획부터 실제 도메인 배포까지 팀으로 완성. 6주 개발 · 21회 릴리스. 6개 카테고리 뉴스를 매일 AI로 요약해 주 1회 자동 발송.

담당 서비스 기획 · IA · 프론트엔드 환경 및 구조 구축 · 공통 컴포넌트 설계 · 메인화면 등 일부 화면 설계 · API 연동 · 인증 및 구독 서비스 · 백엔드 일부

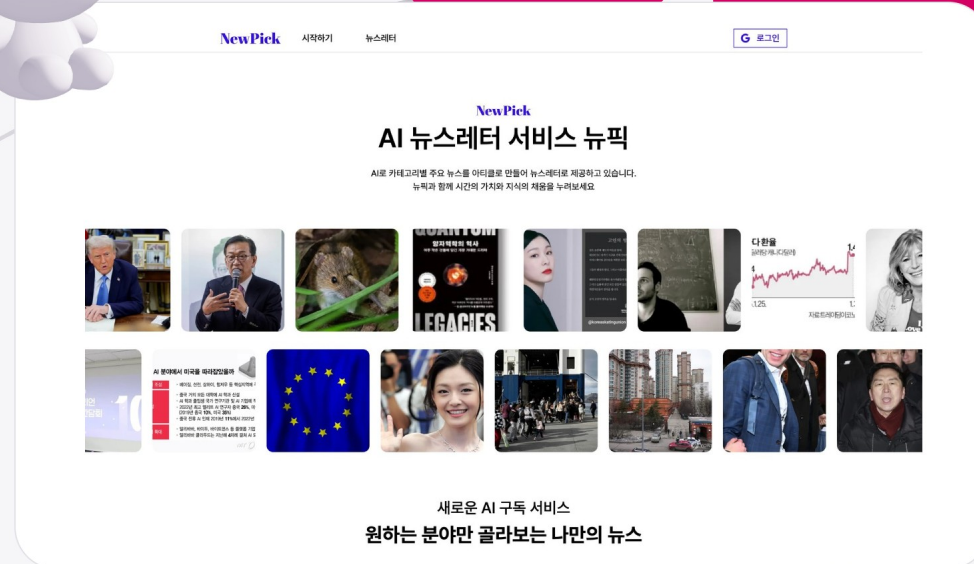
기술 **FE** Next.js · TypeScript · Styled-components · React Query · Vercel / **BE** NestJS · Prisma · MySQL · MongoDB

링크 www.newpick.site · github.com/Devcourse-NewPick/front

- 01** 기획·IA 메인 / 로그인 / 뉴스레터 리스트·상세 / 마이페이지 구조와 화면 흐름 정의, 다크모드 · PC/모바일 반응형 적용
- 02** API 연동 프론트엔드 API 모듈 9개 중 8개 작성, page.tsx(서버 컴포넌트)에서 fetch 하는 SSR 구조로 전환
- 03** 인증 전환 HttpOnly 쿠키 기반 인증으로 전환, 백엔드 담당자와 범위를 합의하고 API 3개 직접 수정
- 04** 협업 이슈·PR 템플릿을 만들어 백엔드 저장소까지 적용, Notion·Slack·GitHub·Postman으로 ERD·API 명세 공유



02



→ 메인 화면 — 관심 분야별 AI 뉴스레터

배운 점

관심사 API가 공유 없이 바뀌어 구독 설정이 저장되지 않았습니다. 백엔드와 수정 범위를 맞춰 해결하고, API가 바뀌면 명세를 먼저 공유하자는 규칙을 팀과 정했습니다. 이후 크롤링 API 경로가 바뀌었을 때는 공유가 먼저 와서 바로 반영할 수 있었습니다.

NewPick — 로그인하지 않았는데 로그인한 것처럼 보이던 화면



PROBLEM

01 새로그침

로그인이 풀림

다시 조회하는 데 필요한 사용자 ID가 스토어와 함께 사라짐

02 로그아웃 → 새로그침

로그인한 것처럼 보임

재발급 실패가 에러로 처리되지 않아 401 응답이 사용자 정보로 저장됨

03 쿠키 없음

로딩이 끝나지 않음

조회가 한 번도 실행되지 않아 pending에 머물



SOLVE

로그인 때 ID를 쿠키에 저장하고, 마운트 뒤 읽어 다시 조회

재발급 실패 시 에러를 던지고, enabled: !!user로 조회 조건 추가

요청을 기다리지 않고 바로 비로그인으로 확정



RESULT

로그인 유지

로그인 버튼 정상 표시

로딩 없이 비로그인 화면

새로그침 중 하이드레이션 오류 없음 · 이후 httpOnly 쿠키로 옮기며 로그인 여부는 서버가 첫 화면에서 판단(SSR)

배운 점

로그인 상태는 "사용자 값이 있는가"가 아니라 "확인을 마쳤고 성공했는가"로 판단해야 합니다. 세 버그 모두 이 구분이 없어서 생겼습니다.

NewPick — 로그아웃 상태에서도 나가던 요청



PROBLEM

로그아웃 상태에서도 요청이 나가고, 실패할 때마다 요청이 연달아 늘어남

로그아웃 방문자가 사용자·북마크 요청 → 401 실패 → 공통 요청 코드가 토큰 재발급 요청 → 로그아웃 요청 = 요청 1번에 3건

React Query가 실패한 요청을 자동으로 다시 시도함 (사용자 정보 2번, 북마크 4번) → 6번 × 3건 = 최대 18건

마이페이지 탭: 화면 속 두 영역이 같은 기사 목록을 따로 요청하고, 카테고리 목록을 받은 뒤 또 요청해서 처음 열 때 최대 8건



SOLVE

로그인 시에만 요청, 받아 둔 데이터는 재사용

enabled: !!user로 로그인한 경우에만 사용자·북마크 요청. 마이페이지는 홈과 같은 쿼리 키(5분 캐시)로 직접 만든 트렌드 API의 목록을 재사용



RESULT

당시 코드 기준으로 센 요청 수

로그아웃 방문자가 카드 페이지를 열 때	최대 18건	→ 0건
마이페이지 탭을 처음 열 때	최대 8건	→ 0~1건
마이페이지 탭을 다시 열 때	2건	→ 0건 (5분 안)

배운 점

실패한 요청 하나가 토큰 재발급과 로그아웃 요청을 부르고, 거기에 자동 재시도까지 붙으면서 요청 수가 여러 배로 늘었습니다. 이후로는 요청 전에 실행 조건을 먼저 확인하고, 같은 데이터는 하나의 쿼리 키로 받아 공유합니다.



MAIN · ORBIT 03

개인 · 2024.08 – 운영 중

MacFolio

API 서버와 DB, 배포 파이프라인까지 직접 만들어 운영 중

브라우저 안에 macOS 데스크톱을 구현한 포트폴리오. 창·Dock·메뉴바를 직접 만들고, 폰에서는 같은 콘텐츠가 iOS 홈 화면으로 바뀝니다. 프론트와 API를 한 모노레포에서 함께 관리합니다.

담당 기획 · 프론트엔드 · API 서버 · DB 설계 · 인프라 · CI 전체**기술** **FE** React 19 · TypeScript · Vite · Milkdown / **BE** NestJS · Prisma 7 · PostgreSQL · Swagger / **빌드** pnpm workspaces · Turborepo**링크** macfolio.hyeoniverse.com · [API 문서](#) · github.com/hyeoniverse/MacFolio

- 01** 데스크톱 창 드래그·리사이즈·포커스 순서와 Dock·메뉴바를 직접 구현, 폰에선 iOS 홈 화면
- 02** 앱 Memo · Messages · Safari · Terminal · 설정을 앱 단위로 분리, Sprout Farm도 안에서 실행
- 03** Memo 폴더·검색·찾아 바꾸기, 초안과 예약 발행, 버전 50개 보관과 30일 휴지통으로 되돌리기
- 04** API GitHub OAuth 직접 구현(state 검증) · 세션은 해시로만 저장 · 화면·API·DB 세 겹 검증 · Swagger 자동 문서
- 05** CI 포맷·린트·타입·단위·빌드·API e2e·브라우저 E2E 8단계, main은 전부 통과해야 머지



03



↳ 메뉴 막대 · 창 관리 · Dock을 브라우저에서 그대로 구현한 화면

배운 점

GitHub API는 토큰 없이 시간당 60회라, 방문자가 늘면 화면이 먼저 깨 집니다. 응답을 서버에서 캐시해 동시 요청을 하나로 묶고, GitHub가 멈추면 마지막 값을 내주게 했습니다. 외부에 기대는 화면은 그쪽이 실패할 때를 먼저 정해 줘야 합니다.

MacFolio — 지원이 끝난 빌드 도구 위에 올라가 있던 저장소



PROBLEM

한 줄 고치고 결과를 볼 때마다 대기

Create React App은 2025년 2월 공식 지원이 끝났습니다. 개발 서버 기동에 5초, 빌드에 6초가 걸렸고, 의존성 1,438개 중 무엇이 왜 필요한지 설명하기 어려운 상태였습니다.



SOLVE

Vite로 옮기면서 의존성을 하나씩 확인

번들러를 Vite로 옮기고, 그 김에 패키지를 전부 다시 훑어 실제로 쓰는 것만 남겼습니다. 테스트는 Vitest, E2E는 Playwright로 통일하고 과정은 docs/migration-cra-to-vite.md에 기록했습니다.



RESULT

네 수치가 함께 내려갔습니다

개발 서버 기동

5.05초 → 0.58초

8.7배

프로덕션 빌드

6.33초 → 1.88초

3.4배

JS 번들 (gzip)

138.2 → 127.7KB

7.6% 감소

의존성 패키지

1,438 → 171개

88% 감소

배운 점

기능이 늘지 않는 작업이지만, 기동이 5초에서 0.58초로 줄면서 수정과 확인을 반복하는 비용이 크게 낮아졌습니다. 의존성을 1,438개에서 171개로 줄이며 각 패키지가 필요한 이유도 다시 확인했습니다.

MacFolio — 포트를 열지 않고 API 서버를 공개하기



PROBLEM

API를 공개하려면 포트를 열어야 한다

클라우드 VM에 API를 올리고 브라우저에서 호출하려면 공인 IP와 포트를 그대로 노출하게 됩니다. 비밀번호를 받는 방명록과 GitHub OAuth 관리자 로그인에 붙어 있어 그대로 열어 둘 수 없었습니다.



SOLVE

연결은 서버가 먼저 걸고, 저장은 원문 없이

Cloudflare Tunnel로 VM이 바깥으로 먼저 연결을 걸게 해 SSH 외에는 포트를 열지 않았습니다. 저장하는 값은 원문을 복원할 필요가 없는 형태로 바꾸고, 그 결과를 API e2e 테스트로 확인했습니다.

EDGE

Cloudflare Workers



TUNNEL

아웃바운드 연결만



API

Oracle Cloud VM · Docker



DATA

PostgreSQL



RESULT

네 가지 구성으로 정리했습니다

서버 노출

열린 포트는 **SSH뿐**
공인 IP 비노출

세션

토큰은 **SHA-256 해시**만
쿠키 httpOnly · Secure

개인정보

방명록 비밀번호 **script**
IP는 **HMAC**으로만 저장

권한·제한 검증

401·400·429를
API e2e 테스트로 확인

배운 점

세션 토큰은 같은지만 확인하면 되고, IP는 같은 사람인지만 알면 됩니다. 되워일 일이 없는 값은 원문을 들고 있을 이유가 없습니다.



MacFolio — 기본값대로 두면 그대로 약점이 되는 자리들

서버를 직접 띄우면 흔히 쓰는 쪽을 그냥 따라가기 쉬운 자리가 생깁니다. 무엇을 고르지 않았는지도 같이 적었습니다.

결정	대안	고른 것	막는 위험
세션 저장	토큰 원문을 DB에 보관	쿠키에만 원문, DB에는 SHA-256 해시만	DB가 새어도 세션을 가로챌 수 없음
관리자 식별	GitHub username	바뀌지 않는 계정 숫자 ID	이름을 바꿔도 권한이 끊기지 않음
방문자 식별	로그인 요구, IP 원문 저장	쿠키 토큰과 IP는 HMAC으로만 저장	로그인 없이도 본인 댓글만 삭제
서버 공개	포트를 열고 nginx·인증서 운영	Cloudflare Tunnel로 나가는 연결만	열린 포트는 SSH뿐, 인증서 관리 없음
필수값 검증	화면에서만 막기	화면 · API · DB의 CHECK 제약 세 겹	화면을 건너뛴 요청도 깨진 값을 못 넣음
업로드 검증	확장자로 형식 판별	파일 앞부분(매직 바이트)으로 확인	확장자만 바꾼 파일이 통과하지 못함
글 삭제	바로 지우기	30일 휴지통과 버전 50개 보관	실수로 지워도 되돌릴 수 있음

배운 점

흔히 쓰는 쪽을 고르지 않은 자리마다 이유가 필요했습니다. 대안을 같이 적어 두면 나중에 왜 그랬는지 다시 캐내지 않아도 됩니다.



MAIN · ORBIT 04

7인 팀(프론트엔드 4 · 백엔드 3) · 2024.10

WTD

화면에 보이는 상태와 서버의 상태를 맞추는 일을 맡은 경험

What To Do — 할 일과 세부 할 일을 드래그 앤 드롭으로 관리하고 루틴으로 반복 일정을 돌리는 투두 웹. 데브코스 첫 팀 프로젝트로 React + Express 풀스택 구성.

담당 백엔드 연동 · API 관리 · 상태 관리(Context) · README

기술 React · JavaScript · Tailwind CSS · Context API · Axios · react-beautiful-dnd / BE Node.js · Express

링크 what-to-do-chi.vercel.app · github.com/Devcourse-WhatToDo

결과 API 12개 연동 (할 일 8 · 루틴 4)

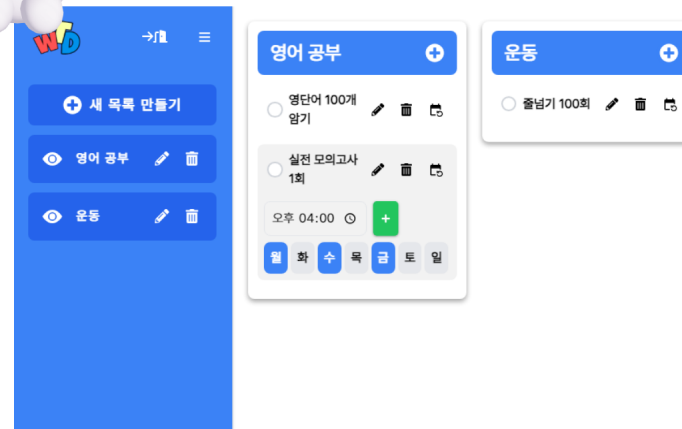
01 API 모듈화 할 일·하위 할 일·루틴 CRUD 요청을 taskApi.js·routineApi.js로 모아 Axios로 통신

02 상태 관리 TaskContext·RoutineContext로 드로어·카드·대시보드 데이터 공유

03 오류 처리·동기화 통신 실패 시 안내, 비로그인 할 일은 로컬 스토리지에 보존했다가 로그인하면 서버로 일괄 동기화



04



↳ 목록별 할 일 카드 · 루틴 요일/시간 설정 화면

배운 점

체크 토글이 상태 객체를 직접 바꿔서, 제목을 고치면 체크가 초기화됐습니다. 필드별 불변 업데이트로 나눠 해결했습니다. 상태를 직접 바꾸면 수정끼리 서로를 덮어씹습니다.

WTD — 매분 초기화되는 루틴, 화면에만 남아 있던 체크



PROBLEM

폴링을 넣자 오래된 목록으로 요청

- ① setInterval 콜백이 처음 렌더 때의 목록을 계속 붙잡음(오래된 값 참조)
- ② 스케줄러와 같은 순간 조회하면 초기화 전 데이터를 받음



SOLVE

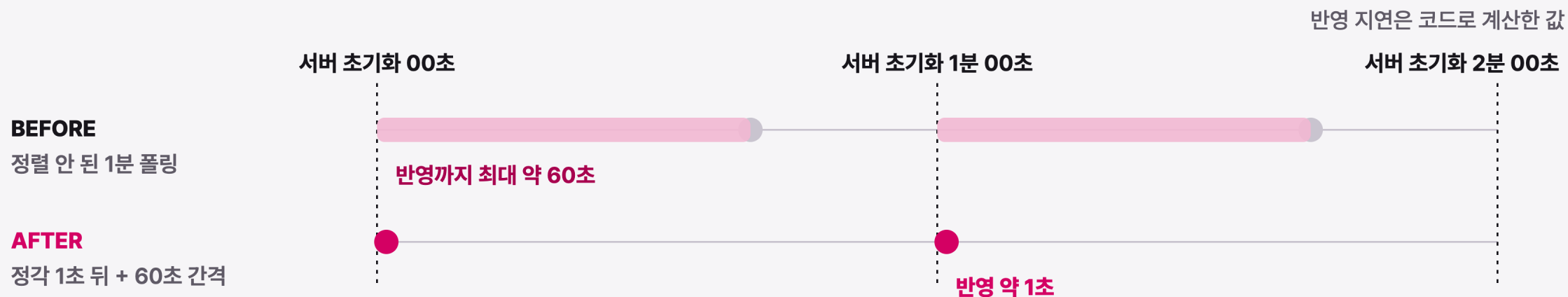
최신 목록을 읽고, 조회를 서버보다 1초 늦춤

콜백이 호출될 때마다 최신 목록을 읽고, 첫 조회를 다음 정각 1초 뒤로 맞춘 뒤 60초마다 조회



RESULT

반영 지연 최대 약 60초 → 약 1초



배운 점

한 번 등록한 타이머 콜백은 그 시점의 값을 계속 들고 있습니다. 지금이라면 useRef로 최신 값을 참조하고, 로그아웃할 때 타이머를 정리하겠습니다.

WTD — 로그인하면 사라지던 비로그인 할 일



PROBLEM

비로그인 할 일과 로그인 사용자의 할 일이 같은 저장 공간을 써서, 로그인하면 사라지거나 섞임



SOLVE

저장 공간을 나누고, 로그인하면 서버로 옮기는 흐름을 만들

guestTasks

비로그인 할 일
(localStorage)



로그인



일괄 등록

서버에 한 번에 저장



다시 불러오기

userTasks_{id}



guestTasks 삭제



FOLLOW-UP

일괄 등록 뒤 하위 할 일이 저장되지 않음

프론트는 응답을 `insertIds`로 읽고 백엔드는 `insertedIds`로 보내 하위 할 일의 `listId`가 비어 있음 → 필드명을 맞춰 해결



RESULT

로그인 전후로 할 일과 하위 할 일이 그대로 유지

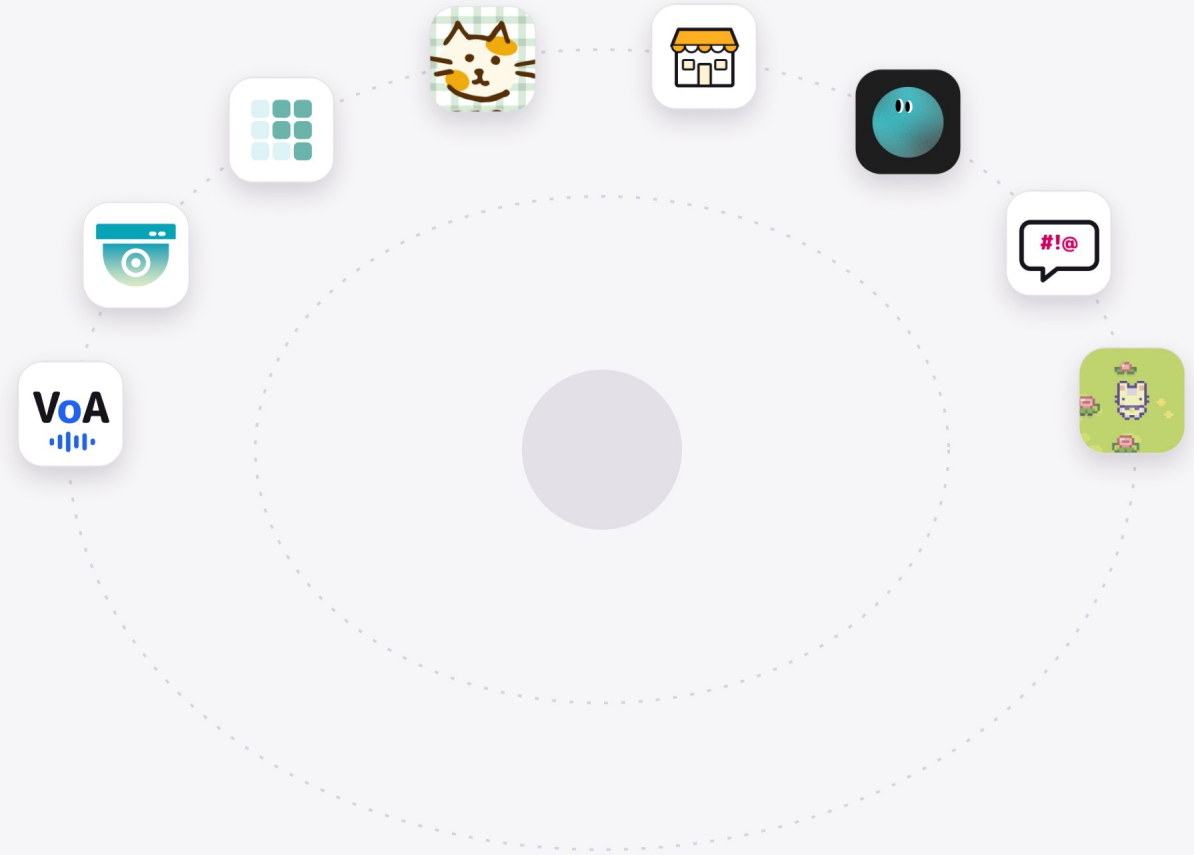
비로그인으로 만든 할 일이 로그인한 계정으로 이동

배운 점

필드명이 하나 다른 것은 에러로 잡히지 않고 `undefined`로 조용히 넘어갑니다. 이후로는 API 응답 형식을 명세와 먼저 대조한 뒤 연동합니다.

사이드 프로젝트

데스크톱 · 웹 · 모바일 · 게임 · AI까지 플랫폼을 넓게 경험한 여덟 개 프로젝트. 두 개씩 묶어 핵심만 정리했습니다.





Voice of All

3인 팀 · 2023.04 - 11 · 프로보노 공모전 동상



카메라로 사용자를 감지하고 음성 안내로 열차를 예약하는 시각장애인을 위한 키오스크 SW. 최종 확인은 터치로 받고 예약 정보는 문자로 전송. ICT멘토링(한 이음).

- 전맹 시각장애인 인터뷰를 반영/미반영 이유까지 기록
- 예약 전 과정 PyQt5 구성, 시각 피드백 + 음성 안내
- TAGO 열차정보 API 단계별 호출, 날짜·시각 필터링

기술 Python · PyQt5 · MediaPipe · STT/TTS · TAGO API



ChasEye

3인 팀 · 2022 - 2023.05 · 우수상 · **논문 게재**



실시간 웹캠·네트워크 분석으로 비대면 시험 중 이상행위를 탐지하는 데스크톱 GUI. 졸업 프로젝트.

- 얼굴 수·홍채 비율로 부재·다자 응시·시선 이탈 판별
- 패킷 급증으로 검색 추정, 단발성 신호는 누적 시 경고
- 기능별 각 50회 테스트: 정면 100% · 우 96% · 좌 97% · 패킷 86%

기술 Python · PyQt5 · MediaPipe · Npcap



PurrFectday

2인 팀 · 2024.04 - 06 · 졸업 프로젝트



고양이를 길들이고 방을 꾸미며 할 일을 완수하게 만드는 게이미피케이션 투두 리스트 iOS 앱.

- 월/주/전체 캘린더 투두, 반복 루틴과 완료 포인트
- 고양이 인터랙션, 달성률에 따라 방 청결도 변화
- 250종+ 인테리어 상점, 이웃 방문·응원 메시지

기술 Swift · UIKit · SpriteKit · Firebase(인증, 데이터)



모두의 키오스크

3인 팀 · 팀장 · 2022.07 - 08 · GURU2 해커톤



키오스크가 낯선 50대 이상 사용자가 집에서 미리 연습할 수 있는 카페 주문 시뮬레이터 iOS 앱.

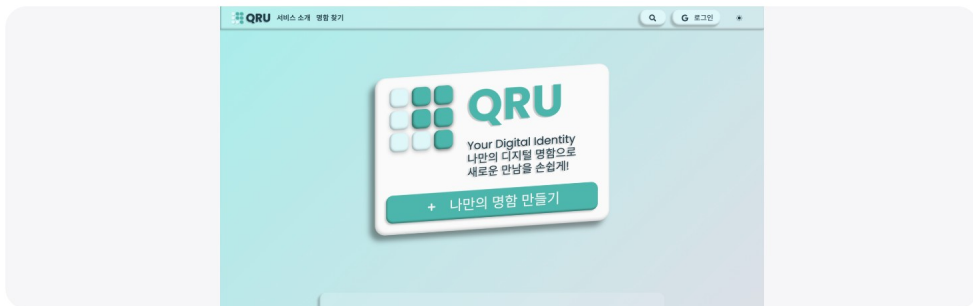
- 매장/포장부터 결제·영수증까지 실제 키오스크 흐름 재현
- 메뉴·옵션 화면: UICollectionView, 다중 섹션 TableView
- 옵션 취소 시 상태 복귀, 장바구니 수량·금액 집계

기술 Swift · UIKit · CocoaPods(Parchment)



QRU

개인 · 2024.11 제작 · 2026.09 개편 · qryou-app.web.app



QR 코드로 내 정보를 담아 공유하는 디지털 명함 웹. 회원가입 없이 익명 인증으로 카드를 만들고, 구글 로그인 시 그대로 이어받음.

- 항목별 공개·비공개 분리 저장 — 비공개 값은 공개 문서에 아예 없고, 카드 페이지는 noindex
- 별도 서버 없이 Firestore 보안 규칙으로 권한 처리, 시리얼 번호 조회용 문서 분리
- 셔플로 공개 카드 5장 추천(성별·MBTI·SNS 필터), 다크/라이트·반응형, GitHub Actions 배포

기술 TypeScript · React 18(Vite) · styled-components · Redux Toolkit · React Query · Firebase(Firestore · Auth · Hosting) · Playwright



Sprout Farm

개인 · 2024.06 · 2026.09 웹 배포 · sprout-farm-beta.vercel.app



자정 전까지 도망친 동물 20마리를 되찾는 탐다운 2D 픽셀 게임. Unity로 만들고 WebGL로 빌드해 웹에 배포.

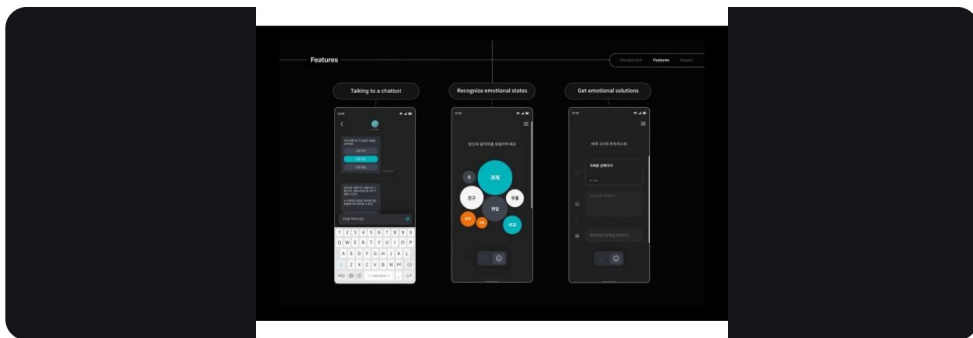
- 20×20 타일맵 4장이 플레이어 따라다니는 무한 맵, 청크 좌표 해시로 같은 지형 재생성
- A* 경로 탐색으로 도망·추적하는 동물 AI, 체력과 낮/밤 시간 압박
- 서버리스 함수와 Redis로 상위 10위 랭킹, 점수는 서버에서 다시 계산해 검증

기술 Unity 6 · C# · URP 2D · A* Pathfinding · WebGL(Brotli) · Vercel Serverless · Redis



ChatBuddy

4인 팀 · 2023.01 – 03 · GDSC Solution Challenge



우울감을 겪는 청년을 위한 상담 챗봇 Android 앱. Java 네이티브 앱으로 팀과 함께 화면과 외부 API를 연동했습니다.

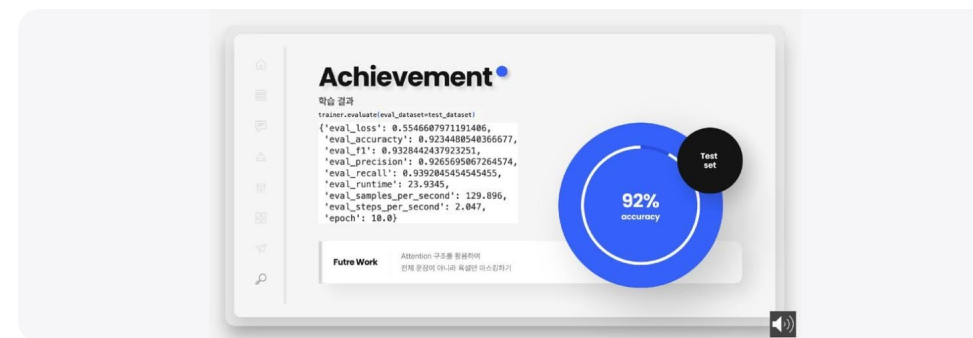
- Google 로그인·챗봇·자주 쓴 단어 버블·계정관리 화면
- ChatGPT API 응답, 위험 표현 감지 시 지원 안내로 분기
- Firebase로 Google 로그인 인증과 사용자 데이터 저장

기술 Java · Android Studio · ChatGPT API · Firebase



악성댓글 분류 모델

개인 · 딥러닝 및 응용 기말 · 정확도 92%



한국어 댓글 데이터셋으로 KcELECTRA를 파인튜닝해 댓글의 악성 여부를 분류한 모델.

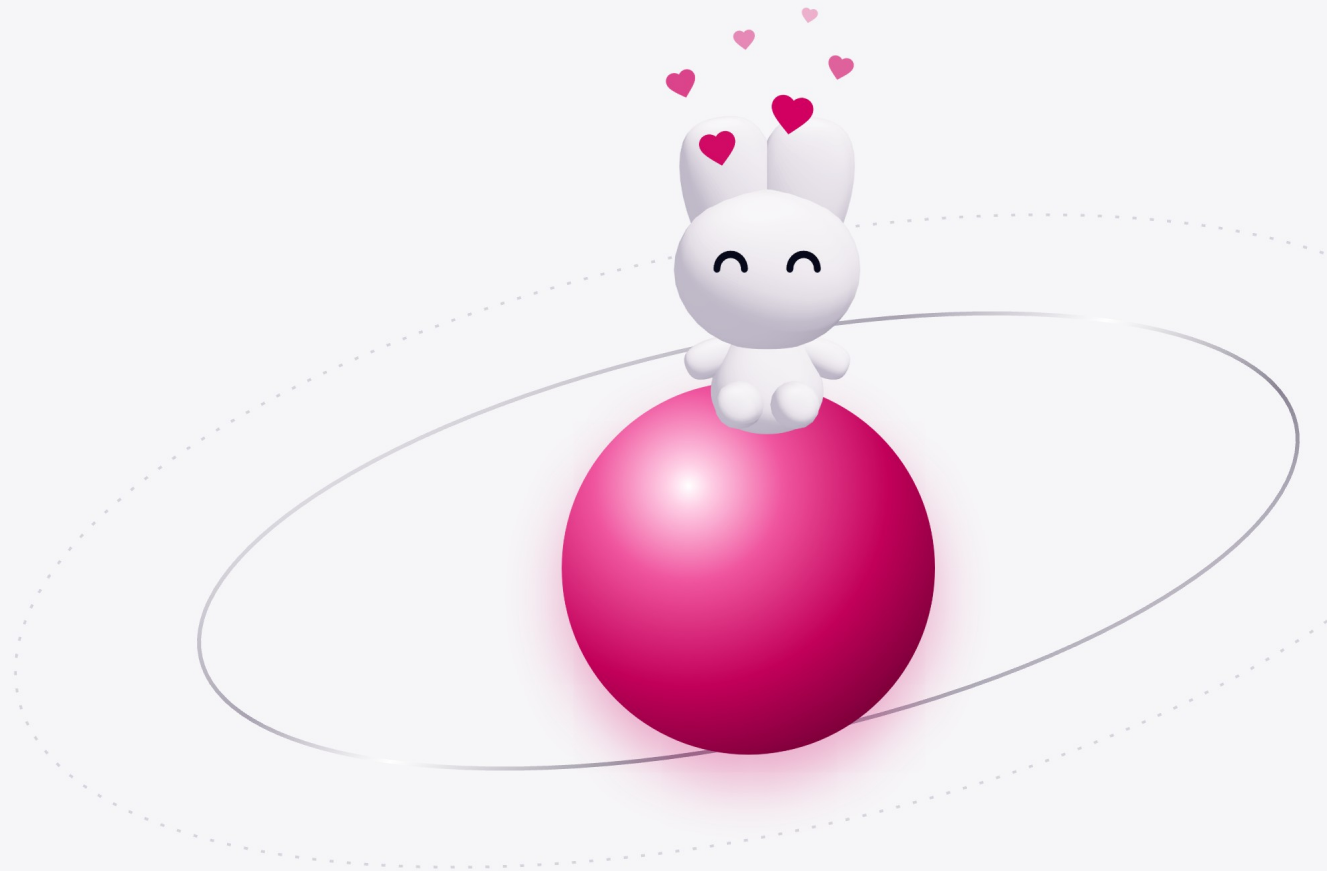
- 15,825개 통합 → 중복 제거 14,837개, Train/Test 8:2
- 오타자·신조어에 맞는 KcELECTRA 토큰나이저로 토큰화
- Trainer API로 학습, accuracy 92% · F1·recall 기록

기술 Python · PyTorch · HuggingFace · Colab

✦ SEE YOU IN THE HYEONIVERSE

감사합니다.

끝까지 봐 주셔서 감사합니다. 프로젝트와 코드는 아래 링크에서 더 자세히 보실 수 있습니다.



MAIL

hyeoniverse.dev@gmail.com

WEB

hyeoniverse.com

GITHUB

github.com/hyeoniverse

TEL

010-7191-2644