

Kakhkhorov Shokhjakhon (Sean)

풀스택 개발자 • DevOps 엔지니어 • AI 엔지니어 • 에이전틱 엔지니어

3년 이상의 실무 경험을 가진 풀스택 개발자입니다. MongoDB, PostgreSQL, MySQL, Redis를 다루고, NestJS와 Express로 타입 안전한 GraphQL·REST API를 만들고, React와 Next.js로 반응형 인터페이스를 구현한 뒤 직접 배포하고 운영합니다.



엔드 투 엔드 개발

데이터 모델, API, 인증, 실시간 기능부터 UI까지



배포와 운영

Docker, Nginx, DNS, TLS와 GitHub Actions CI



AI 통합

사이트 콘텐츠에 근거한 Gemini 기능, 코딩 에이전트와 함께 개발

TS TypeScript

React

Next.js

Node.js

NestJS

Express

GraphQL

Socket.io

MongoDB

PostgreSQL

MySQL

Redis

Docker

Nginx

GitHub Actions

Gemini API

Claude Code

대한민국 수원 · kakhkhorov97@gmail.com · github.com/kakhkhorov97



● 최신 프로젝트 · creatorhub.site

포트폴리오 구성

01



자기소개

배경과 일하는 방식

03쪽

03



기술

핵심 스택과 기타 도구

06쪽

05



인터페이스

탐색, 프로필, 서비스, 이벤트, 로그인

10쪽

07



AI 파인더

사이트 콘텐츠에 근거한 Gemini 어시스턴트

22쪽

09



사례 연구

중복 요청, AI 제한, 스스로 복구하는 배포

29쪽

11



Sofiya Bookstore

이전 프로젝트: React·Express 온라인 서점

35쪽

02



경력

IT Local, Atlantis Investment

04쪽

04



CreatorHub

최신 프로젝트: 개요와 기능

07쪽

06



워크플로

문의, 관리자 영역, 검수, 워커

17쪽

08



엔지니어링

아키텍처, 보안, 테스트, CI/CD, 배포

24쪽

10



데이터 모델

CreatorHub 데이터 구조

32쪽

12



연락처

이메일, 전화번호, 링크

38쪽

코드부터 운영까지, 풀스택 개발

Express와 MongoDB API를 만드는 백엔드 개발자로 시작해, 2023년 6월부터 2026년 1월까지 IT Local에서 풀스택 개발자로 일했습니다. 4인 이상 개발팀의 일원으로 20개 이상 고객사를 맡았고, 데이터베이스 설계와 인증부터 배포, DNS, 장애 대응까지 담당하면서 코드를 어떻게 작성하는지뿐 아니라 배포 후 어떻게 동작하는지까지 신경 쓰게 되었습니다.

최신 프로젝트 **CreatorHub**는 creatorhub.site에서 운영 중입니다. 프론트엔드와 백엔드를 모두 혼자 개발했고, 사이트의 공개 콘텐츠를 바탕으로 방문자의 질문에 답하는 Gemini 기반 어시스턴트를 추가했습니다.

Claude Code, Codex 같은 코딩 에이전트를 명문화된 저장소 규칙 안에서 활용합니다. 모든 변경은 제가 직접 승인하고, 병합 전에는 자동 검사를 거칩니다.



안정성

모니터링, 로그 분석,
신속한 장애 대응



보안

인증, 입력값 검증,
코드 리뷰



문서화

API와 배포 절차 문서화



AI 워크플로

검토 단계를 갖춘 코딩
에이전트



Kakhkhorov Shokhjakhon (Sean)

풀스택 개발자 · DevOps 엔지니어
AI 엔지니어 · 에이전틱 엔지니어

— 경력

풀스택 웹 개발자

IT Local · 2023.06 – 2026.01

백엔드 개발자

Atlantis Investment · 2022.05 – 2022.12

— 거주지

대한민국 경기도 수원시

— 연락처

github.com/kakhkhorov97
kakhkhorov97@gmail.com

근무 경력

풀스택 웹 개발자

2023.06 - 2026.01

IT Local · 우즈베키스탄

웹 개발, 서버·컴퓨터 조립, 네트워크 구성, CCTV 설치 서비스를 제공하는 IT 서비스 회사.

- ◆ 4인 이상 개발팀에서 건설·유통·요식·기업/공공기관 등 20개 이상 고객사 웹사이트 개발·유지보수
- ◆ 인증, 권한 부여, 입력값 검증을 갖춘 API와 서버 측 비즈니스 로직 개발
- ◆ 데이터베이스 스키마 설계, 인덱싱·쿼리 최적화로 페이지·API 응답 속도 개선
- ◆ Ubuntu 서버 5~15대 구축 및 SSH 접근·방화벽 규칙 등 보안 설정
- ◆ Nginx 뒤에서 PM2와 Docker로 Node.js 서비스 운영, Cloudflare로 DNS·TLS 관리
- ◆ 수동 배포를 CI 기반 자동 배포로 전환
- ◆ 애플리케이션 상태 모니터링, 로그 분석, 장애 및 성능 문제 해결
- ◆ 팀원 코드 리뷰, API와 배포 절차 문서화
- ◆ React, Bootstrap, Tailwind CSS로 반응형 페이지 개발
- ◆ 퇴사 전 후임자 온보딩 및 모든 시스템 인수인계

백엔드 개발자

2022.05 - 2022.12

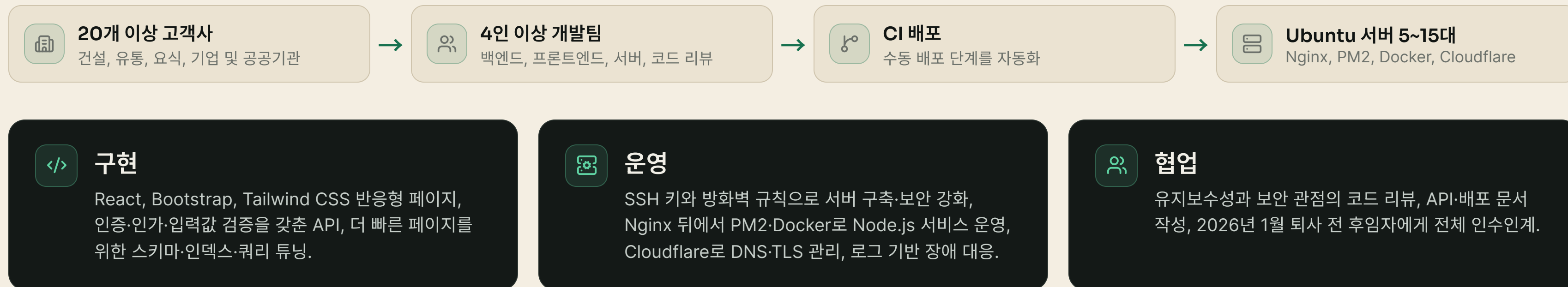
Atlantis Investment · 우즈베키스탄

대학 학업과 병행.

- ◆ Express.js로 프론트엔드와 백엔드를 잇는 RESTful API 구축
- ◆ Mongoose를 활용한 MongoDB 데이터베이스 설계·유지 및 데이터 모델링
- ◆ MVC 아키텍처로 관심사를 분리해 유지보수와 확장이 쉬운 코드 구성
- ◆ 세션과 JWT 기반 회원가입·로그인 구현, bcryptjs로 비밀번호 해싱
- ◆ Multer로 이미지·문서 파일 업로드 엔드포인트 구축
- ◆ 요청 데이터 검증과 일관된 에러 처리 및 상태 코드 적용
- ◆ 검색·필터링·페이지네이션을 위한 MongoDB 쿼리와 인덱스 작성
- ◆ Postman으로 엔드포인트 테스트·문서화, 프론트엔드와 요청·응답 형식 협의

고객사 웹사이트, 개발부터 서버까지 .

IT Local은 다른 기업의 웹사이트를 개발하고 운영합니다. 4인 이상 개발팀에서 첫 페이지부터 그 페이지가 돌아가는 서버까지 전 과정을 맡았습니다.



고객사 사이트는 IT Local 고객의 소유이므로 여기에 실지 않았습니다.

기술 스택

☒ 핵심 · ☒ 매일 사용 ☐ 사용 경험 있음

프론트엔드

React Next.js TypeScript JavaScript Apollo Client Redux Toolkit MUI Tailwind CSS SCSS HTML
CSS jQuery Axios TanStack Query React Router Socket.IO Client Framer Motion Three.js Swiper
Anime.js Typed.js TUI Editor Bootstrap Chakra UI shadcn/ui Figma

백엔드

Node.js NestJS Express GraphQL REST API Socket.io JWT & OAuth 역할 기반 접근 제어 Python EJS
express-session bcryptjs Multer

데이터베이스

MongoDB Mongoose PostgreSQL MySQL Redis 인덱싱·쿼리 최적화 트랜잭션 MongoDB Compass

DevOps

Docker Docker Compose Nginx GitHub Actions Cloudflare DNS·TLS PM2 방화벽 설정 Git & GitHub
GitHub Pages Render

AI 및 에이전트

Gemini API OpenAI API Claude Code Codex Cursor 콘텐츠 검색

테스트 및 도구

Jest Supertest ESLint Prettier 통합 테스트 API 계약 테스트 PostHog Postman Yarn npm NVM

CreatorHub

👤 개인 프로젝트: 프론트엔드, 백엔드, 인프라를 직접 설계·개발·배포했습니다.

방문자가 한국 각지의 독립 크리에이터를 찾고, 그들의 프로젝트와 서비스를 살펴보며, 비공개 프로젝트 문의를 보낼 수 있는 포트폴리오 탐색 플랫폼입니다.

- ◆ 카테고리, 지역, 원격 근무로 검색
- ◆ 관리자 승인을 거치는 크리에이터 워크스페이스
- ◆ 실시간 채팅이 가능한 비공개 문의

프론트엔드

Next.js

React

TypeScript

백엔드

NestJS

GraphQL

Socket.io

데이터

MongoDB

Mongoose

인프라

Docker

Nginx

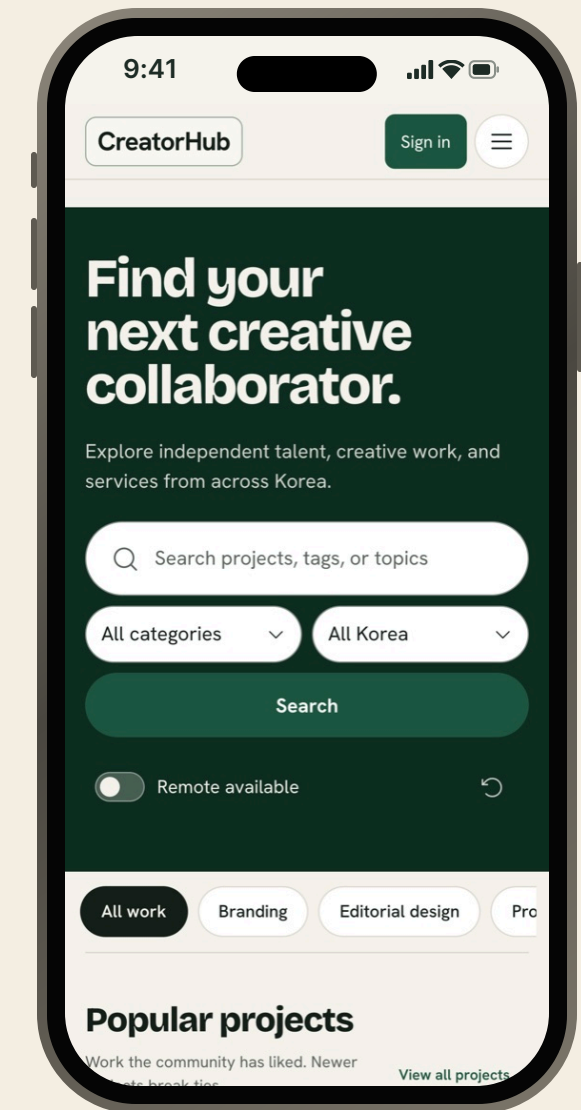
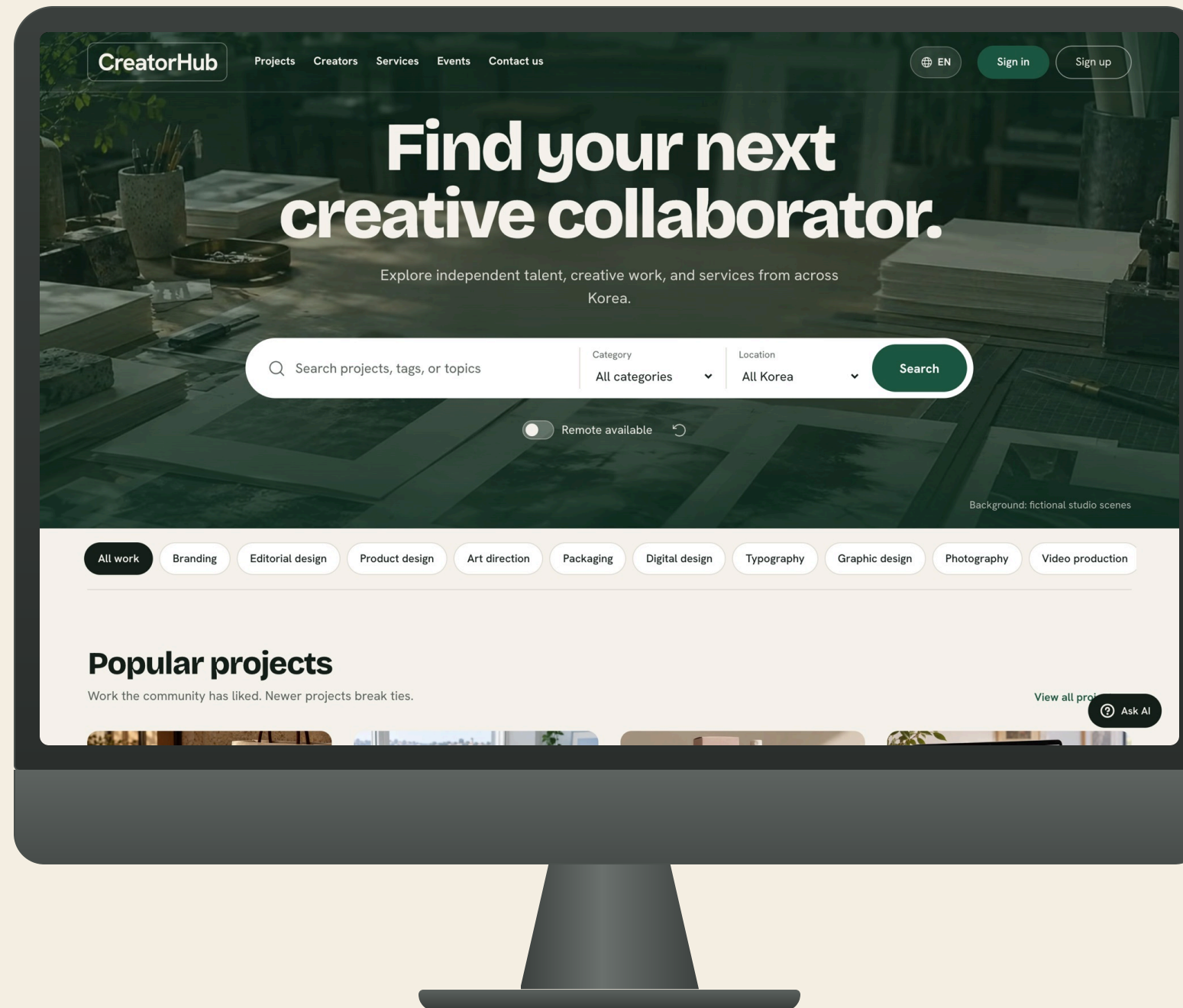
GitHub Actions

AI

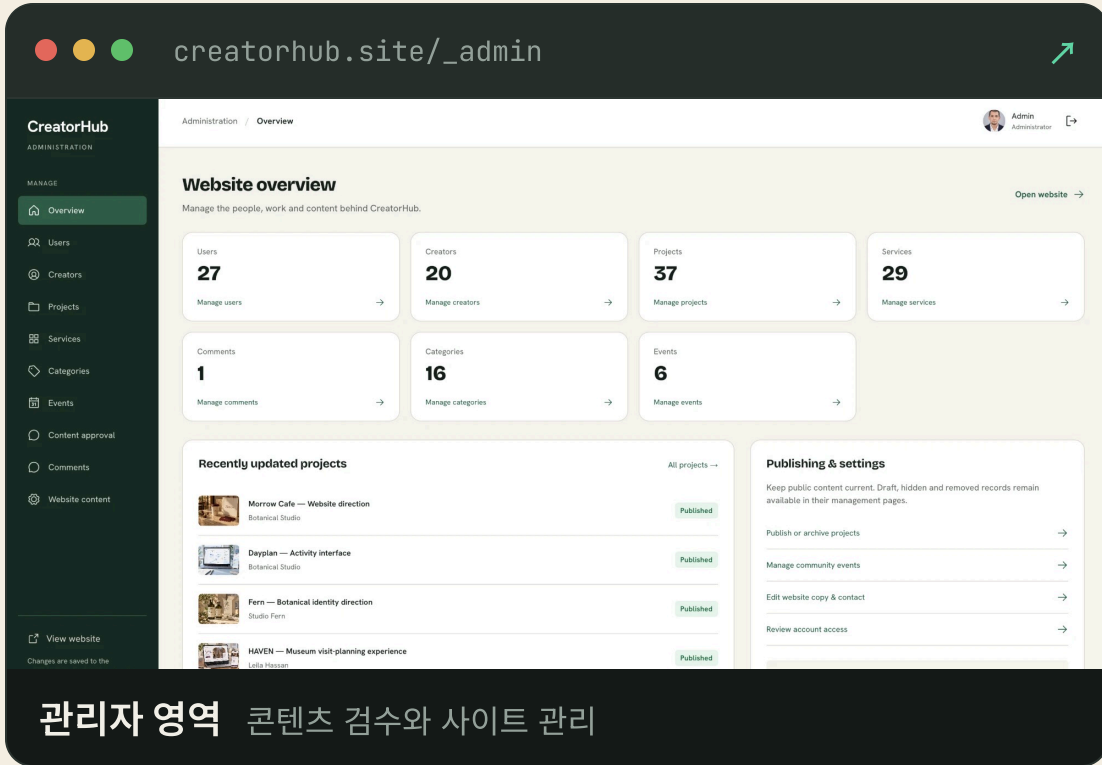
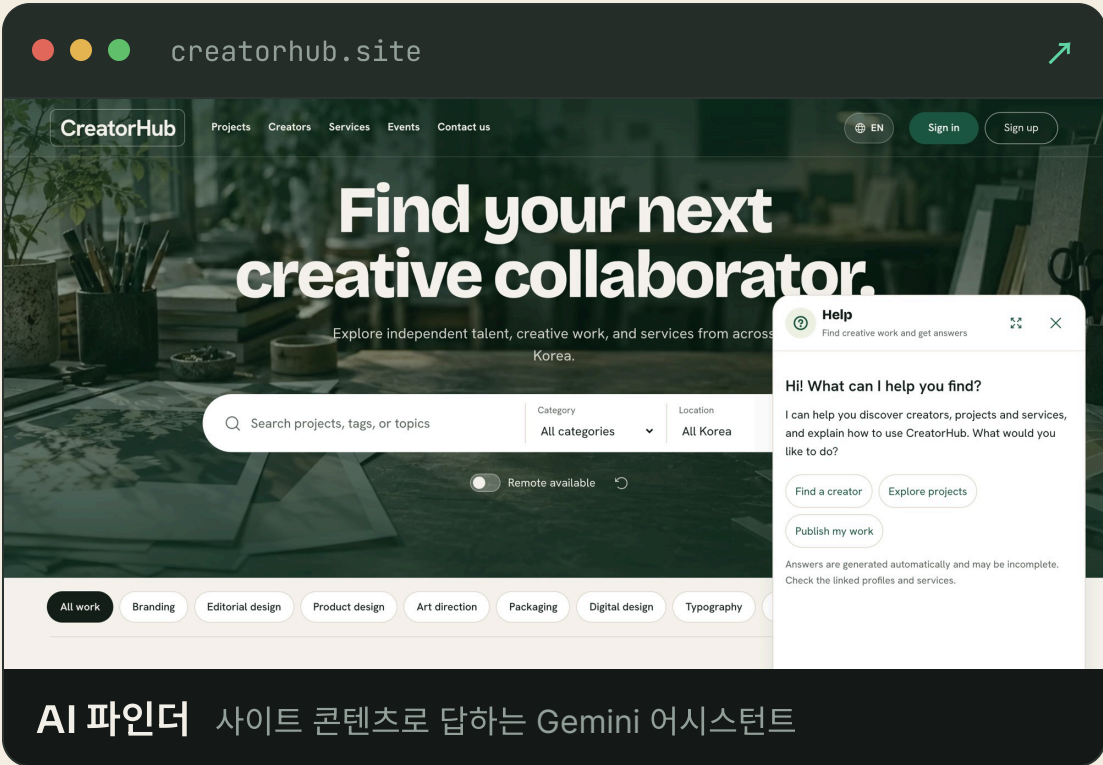
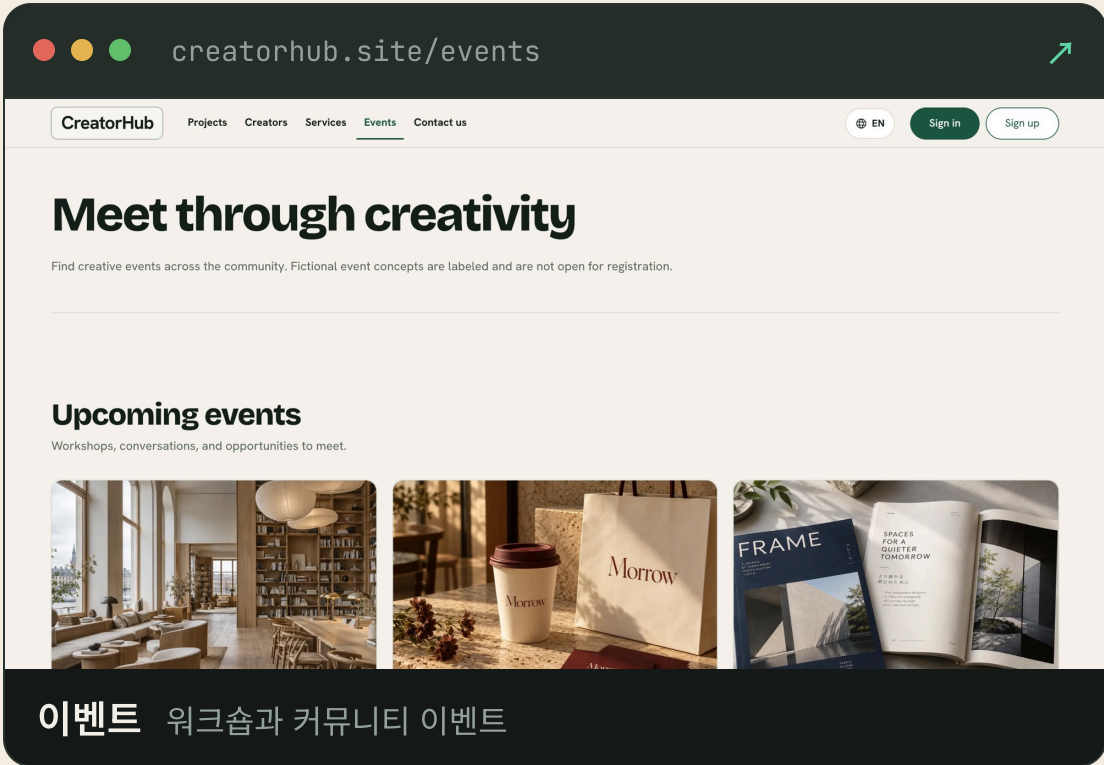
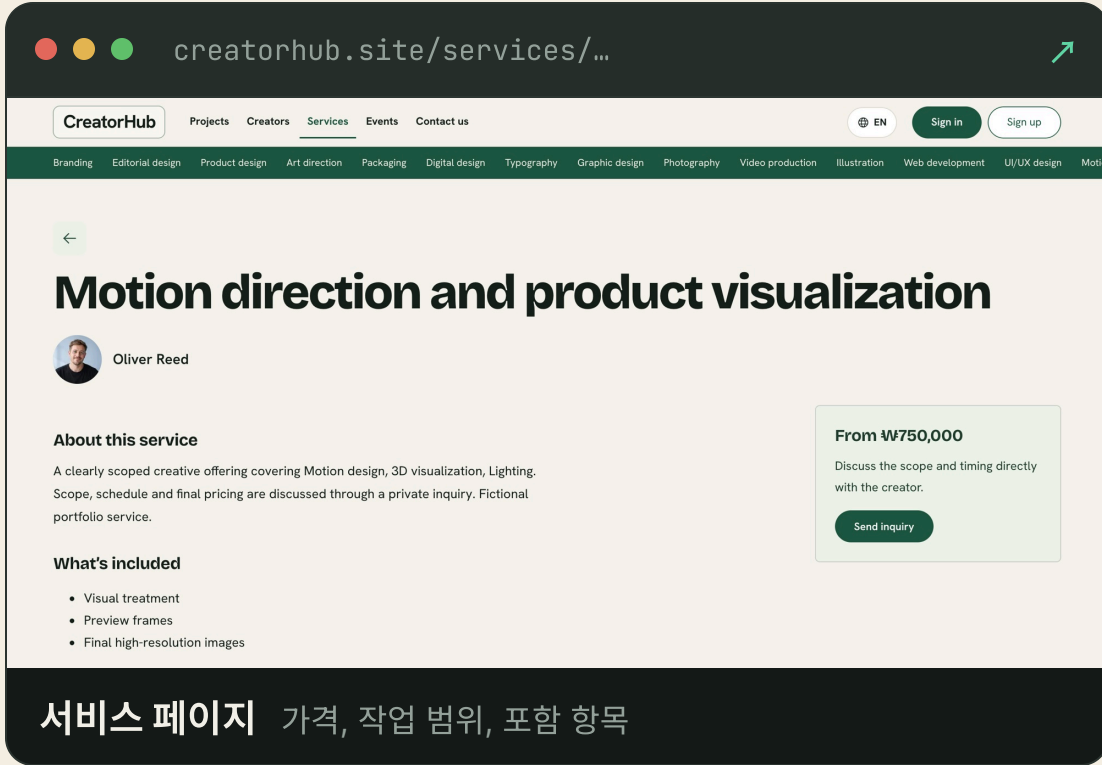
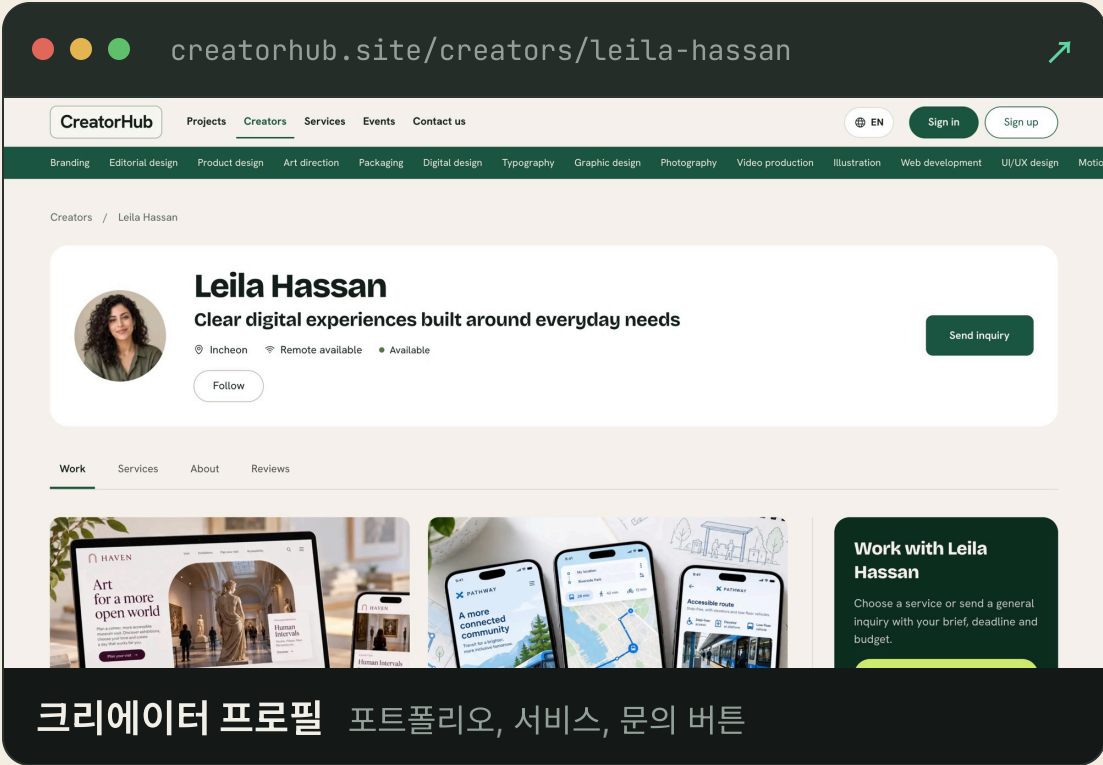
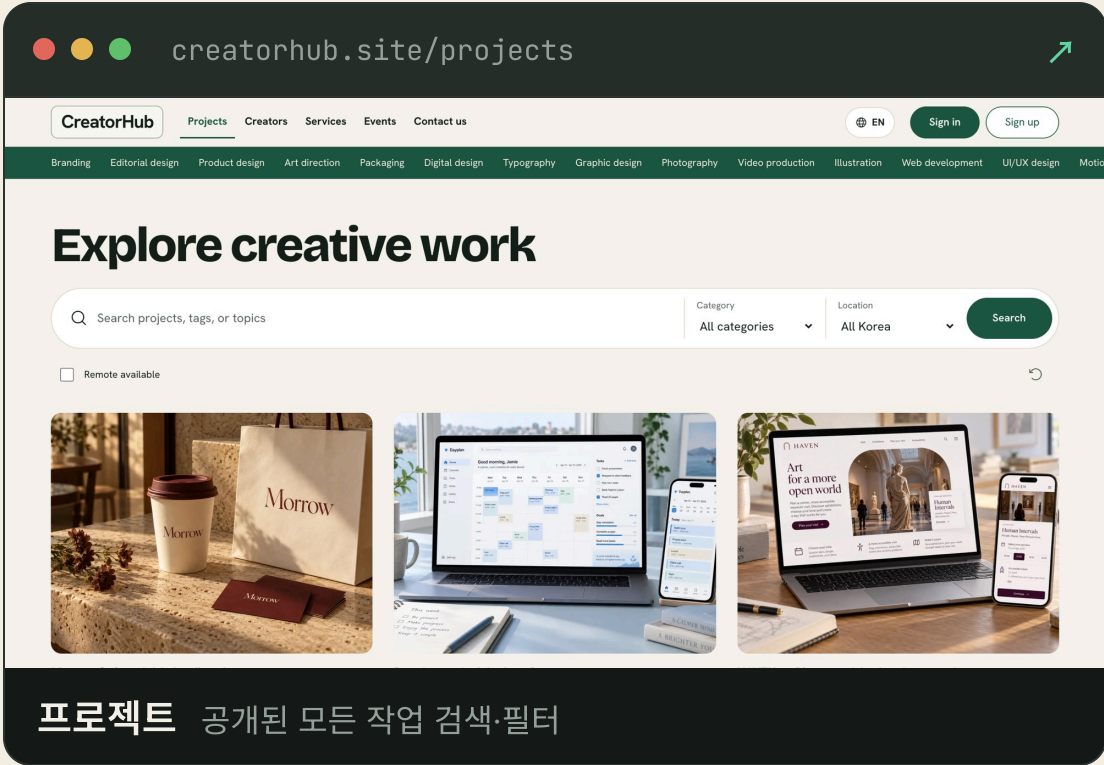
Gemini API

사이트 방문 ↗

저장소: [CreatorHub](#) · [CreatorHub-next](#)

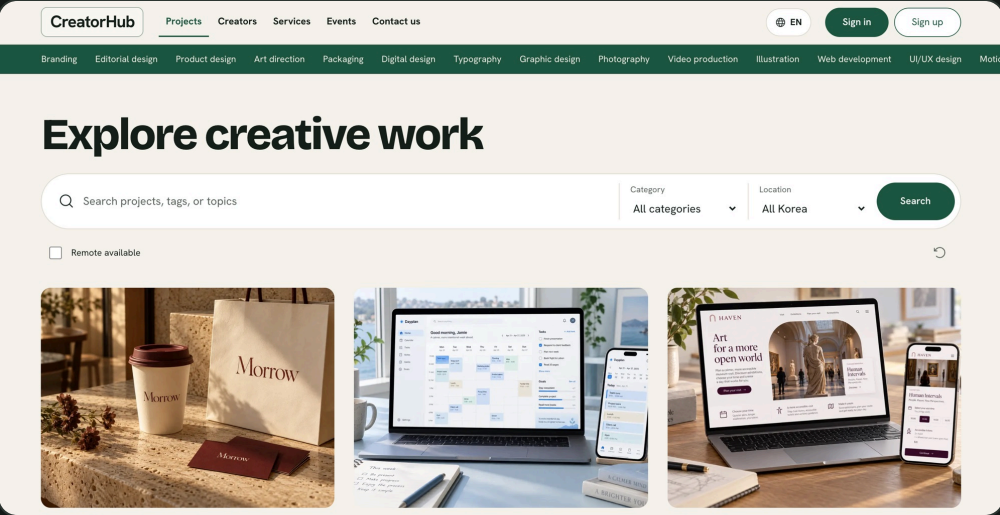


라이브 사이트 둘러보기



플랫폼의 주요 기능

 방문자



 탐색

카테고리, 지역, 원격 근무로 검색

 AI 파인더

사이트 자체 콘텐츠 기반 답변

 커뮤니티

좋아요, 저장, 댓글, 크리에이터 팔로우

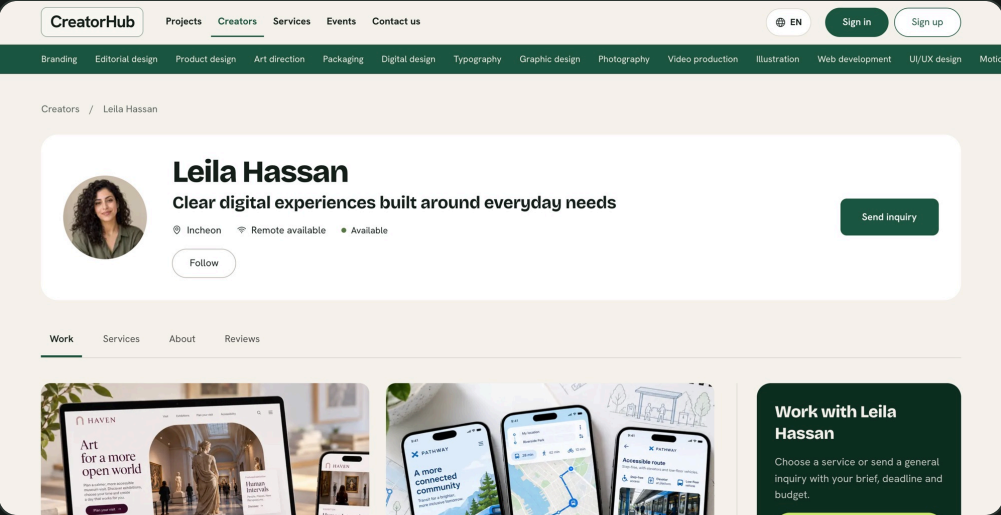
 로그인 방식

이메일, Google, Telegram, Kakao

 3개 언어 지원

영어, 한국어, 우즈베크어

 크리에이터



 워크스페이스

프로필, 프로젝트, 서비스 관리

 비공개 문의

의리 내용과 1:1 실시간 채팅

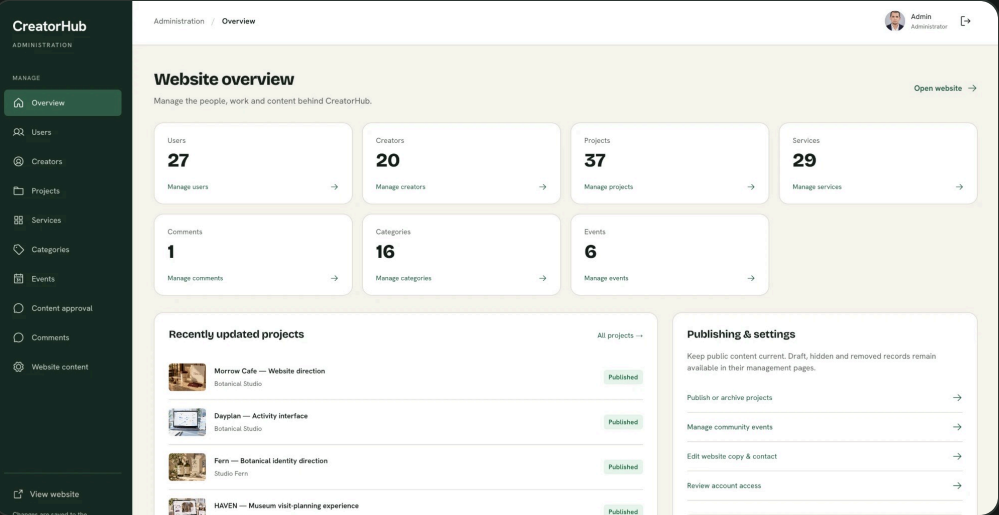
 리뷰

완료된 문의당 평점 1회

 팔로워 알림

새 작업을 팔로워에게 알림

 관리자



 승인

검토 후 새 작업 공개

 안전한 수정

승인 전까지 변경 사항 비공개

 사용자와 역할

계정, 역할, 댓글 관리

 사이트 콘텐츠

카테고리, 이벤트, 사이트 문구

크리에이티브 작업 탐색

방문자는 공개된 모든 작업을 검색하고 카테고리, 도시, 원격 가능 여부로 범위를 좁힐 수 있습니다.

구현 방식

- ◆ 하나의 MongoDB 집계로 필터링·정렬·페이지 나누기를 처리하고, 같은 쿼리에서 전체 개수도 반환
- ◆ 공개 상태이며 계정이 활성화된 크리에이터의 작업만 노출
- ◆ 검색어는 대소문자 구분 없는 매칭 전에 이스케이프 처리
- ◆ 검색어, 필터, 페이지 번호를 URL에 저장해 뒤로 가기 시 같은 위치로 복귀

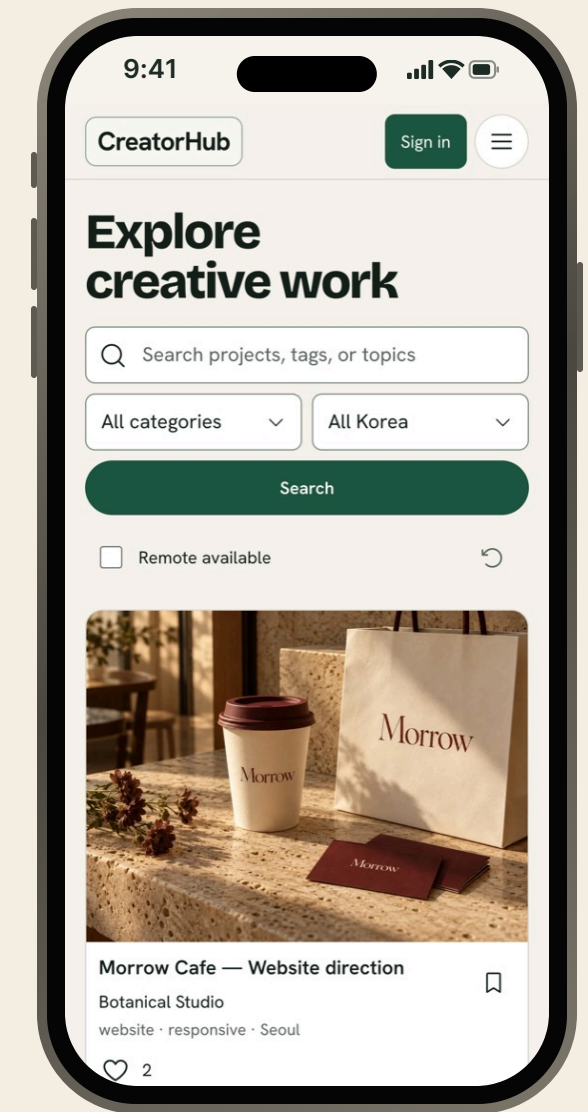
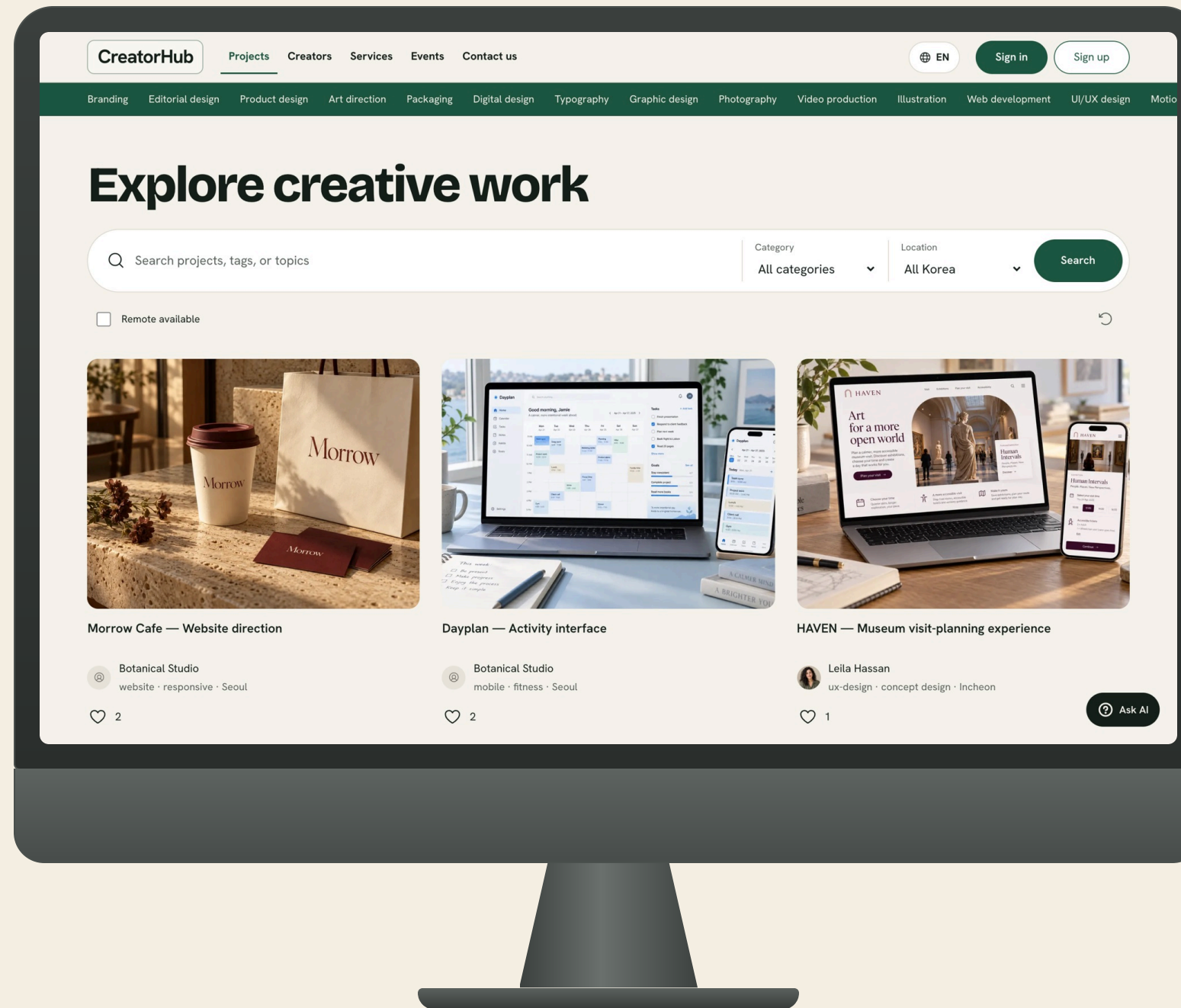
🌿 MongoDB 집계

🔗 GraphQL

🔗 Apollo Client

🟢 Next.js

creatorhub.site/projects ↗



프로젝트 페이지

프로젝트마다 갤러리, 태그, 좋아요, 크리에이터에게 비공개 문의를 보내는 버튼이 있는 전용 페이지가 있습니다.

구현 방식

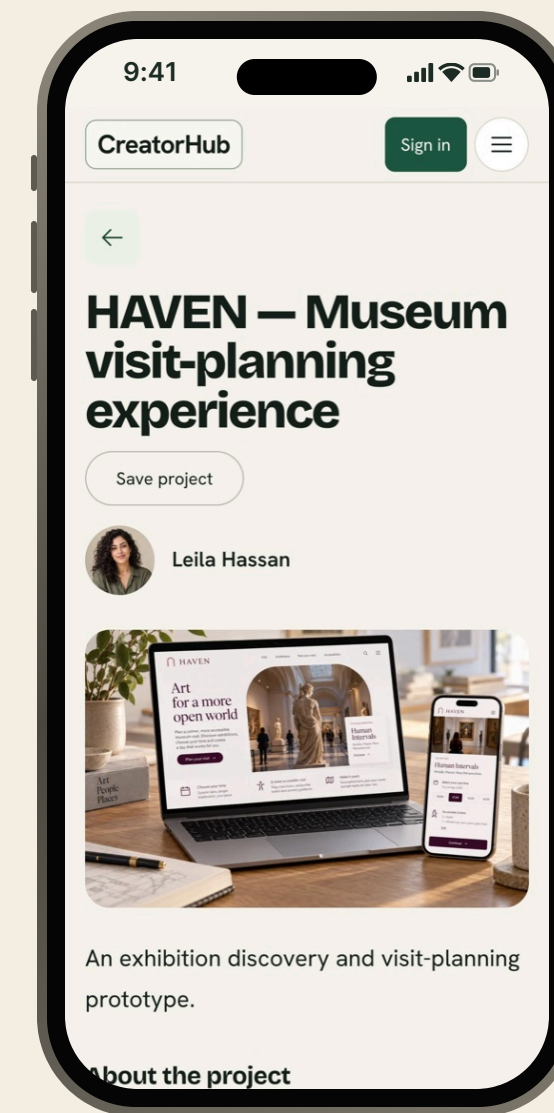
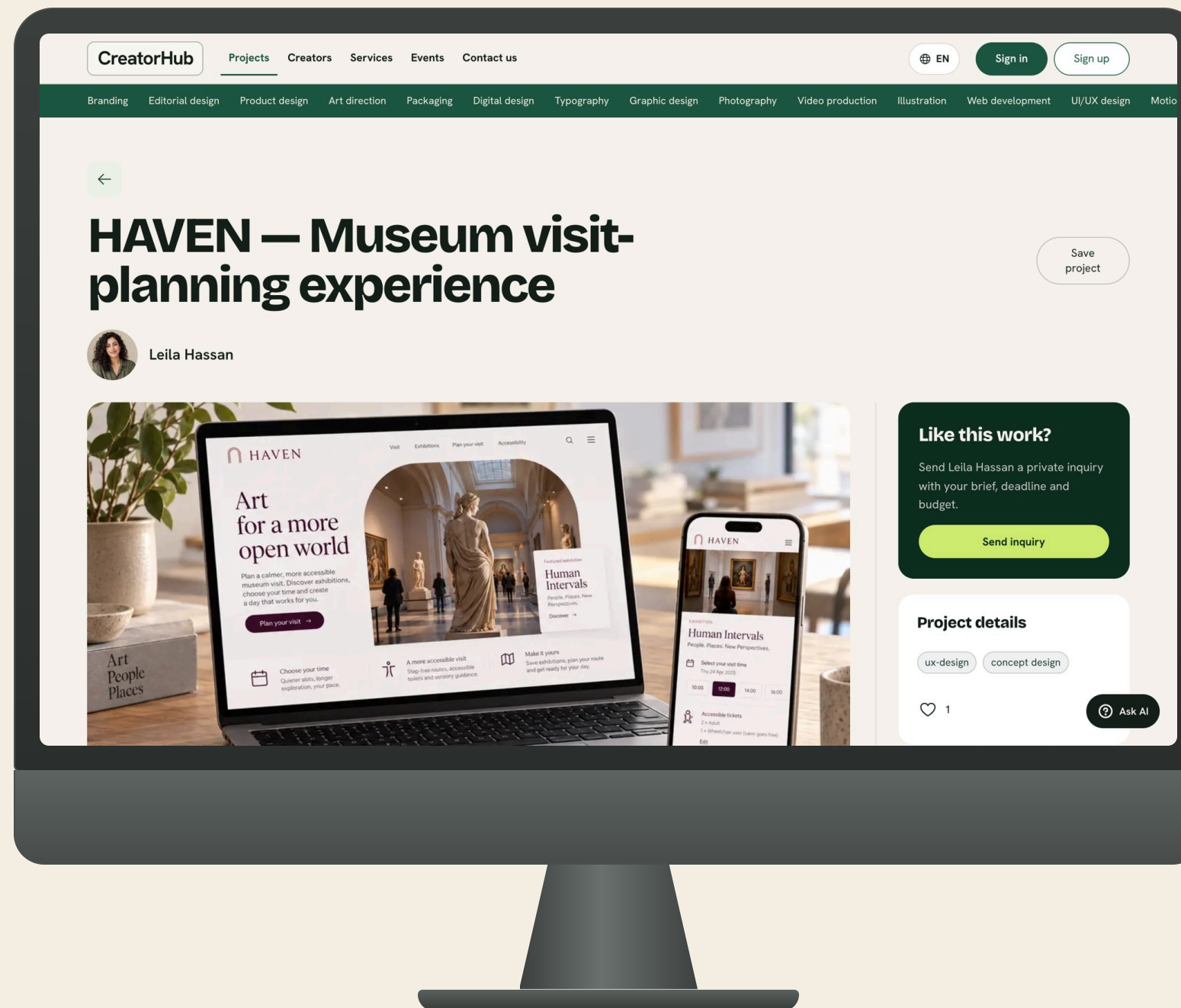
- ◆ 공유 링크용 미리보기 태그(제목, 요약, 커버)를 서버에서 렌더링하고, 없는 ID는 실제 404를 반환
- ◆ 좋아요는 회원·프로젝트당 한 번만 저장되며, 활동 이벤트와 같은 트랜잭션에서 기록
- ◆ 자신의 작업에는 좋아요 불가

 Next.js SSR

 GraphQL

 MongoDB 트랜잭션

이 프로젝트 보기 ↗



크리에이터 프로필

헤드라인, 도시, 작업 가능 여부를 보여 주는 공개 프로필로, 작업·서비스·리뷰·소개 탭이 있습니다.

구현 방식

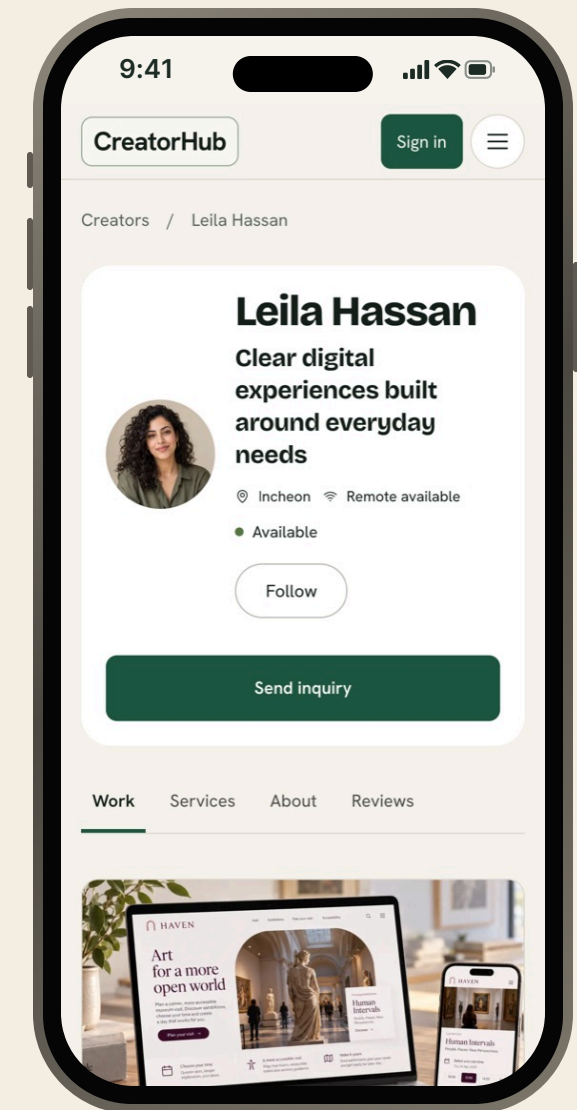
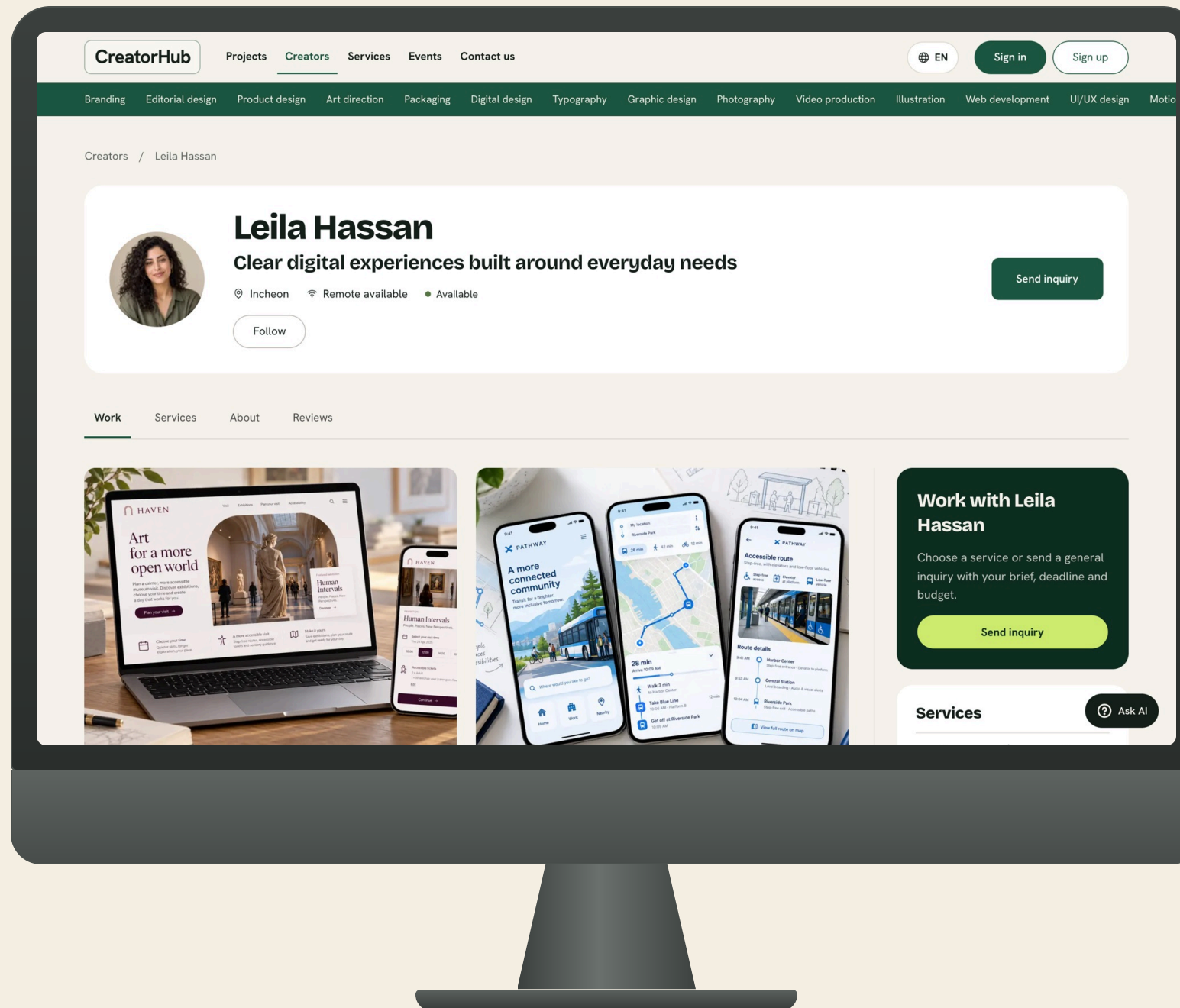
- ◆ 데이터베이스 ID 대신 읽기 쉬운 슬러그로 프로필 조회
- ◆ 공개 상태이며 계정이 활성화된 프로필만 열람 가능
- ◆ 팔로우는 트랜잭션으로 처리하며, 중복 생성 대신 기존 팔로우를 다시 활성화
- ◆ 로그아웃 상태의 방문자는 로그인 후 같은 프로필로 복귀

 GraphQL

 MongoDB

 Next.js SSR

이 프로필 보기 ↗



— CREATORHUB · 인터페이스

크리에이터와 서비스

카테고리, 도시, 원격 근무, 예산으로 필터링할 수 있는 크리에이터 디렉터리와 서비스 마켓플레이스입니다.

— 구현 방식

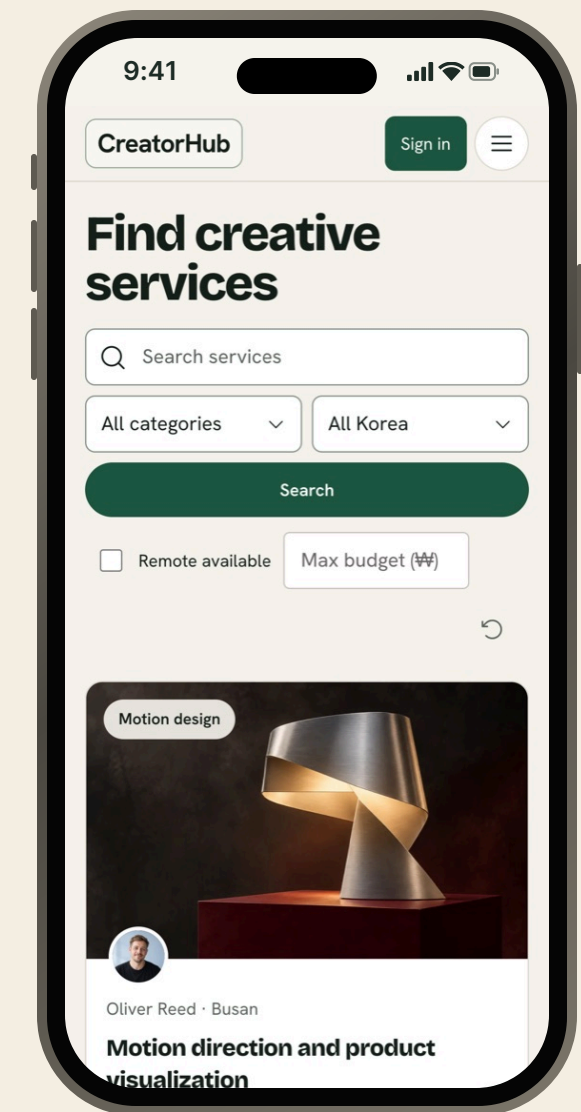
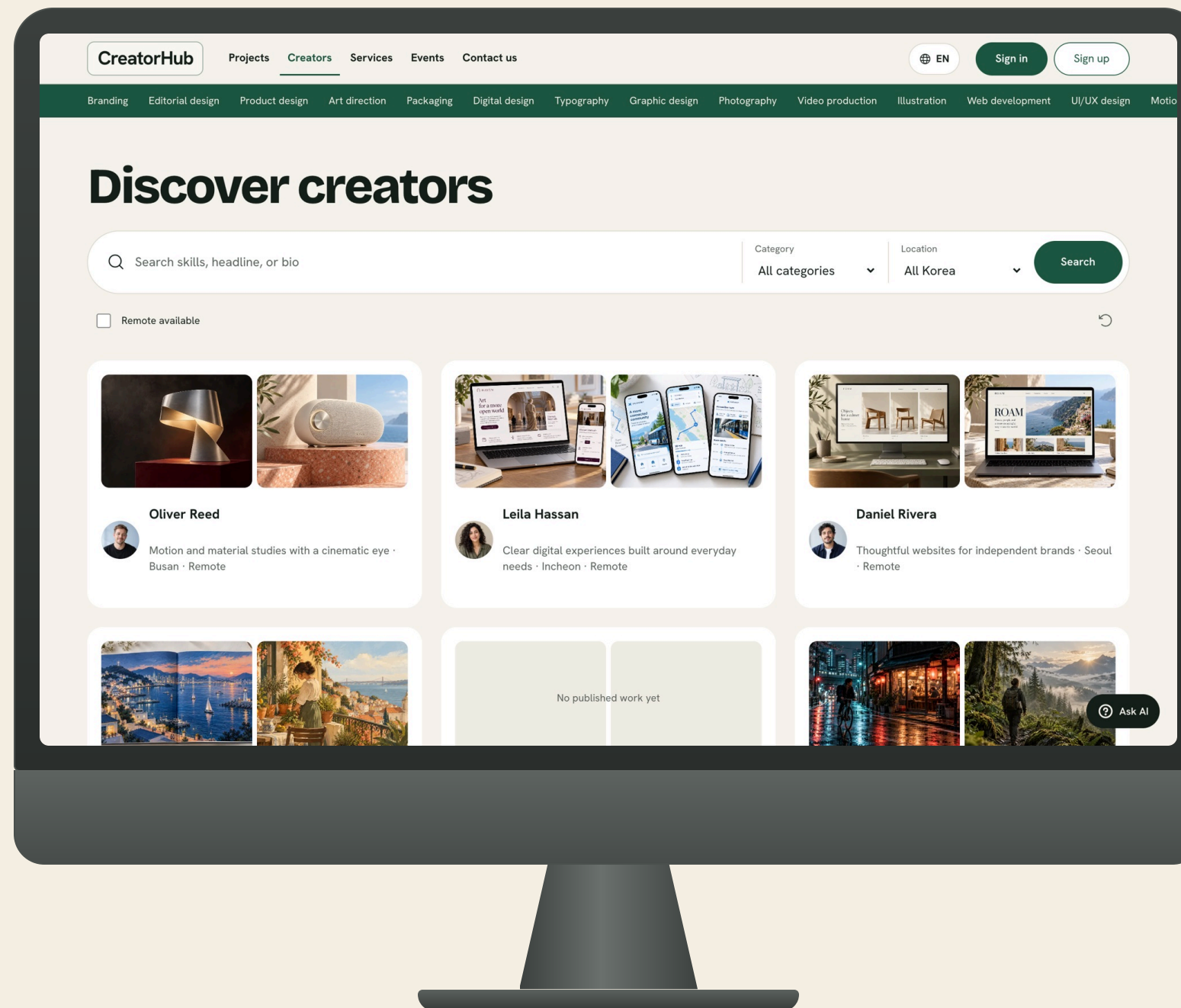
- ◆ 예산 필터는 한도 이하 서비스와 모든 맞춤 견적 서비스를 표시
- ◆ 크리에이터 카드에는 최신 커버 3개를 표시하며, 현재 페이지의 항목만 조회
- ◆ 두 목록 모두 프로젝트 페이지와 같은 검색·노출 규칙 사용

🌿 MongoDB 집계

🔗 GraphQL

🧪 i18n

creatorhub.site/creators ↗



서비스 페이지

작업 범위, 결과물, 원화 시작 가격 또는 맞춤 견적을 보여 줍니다. 최종 조건은 비공개 문의로 협의합니다.

구현 방식

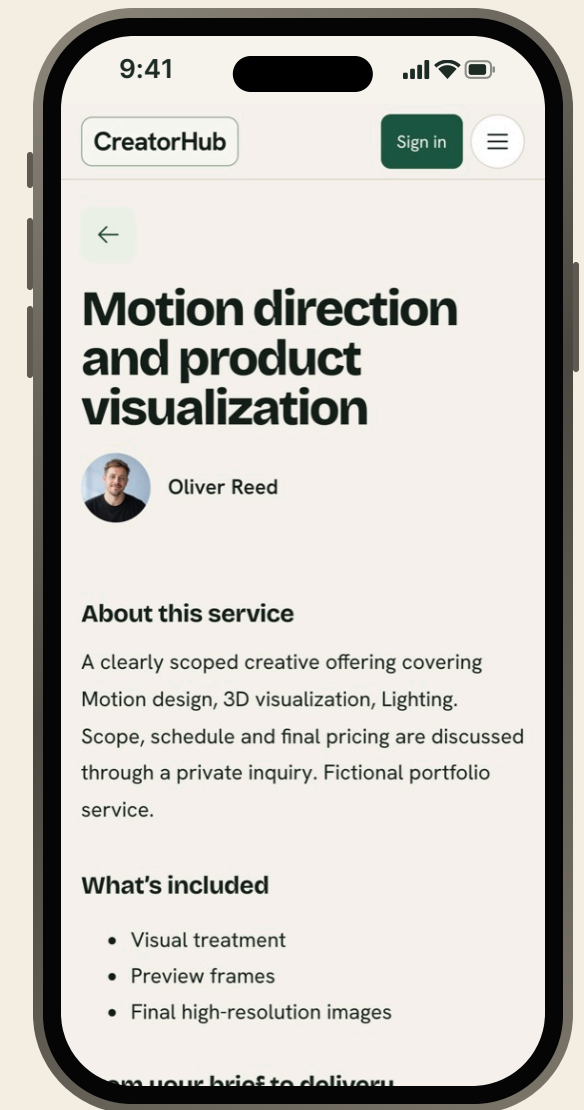
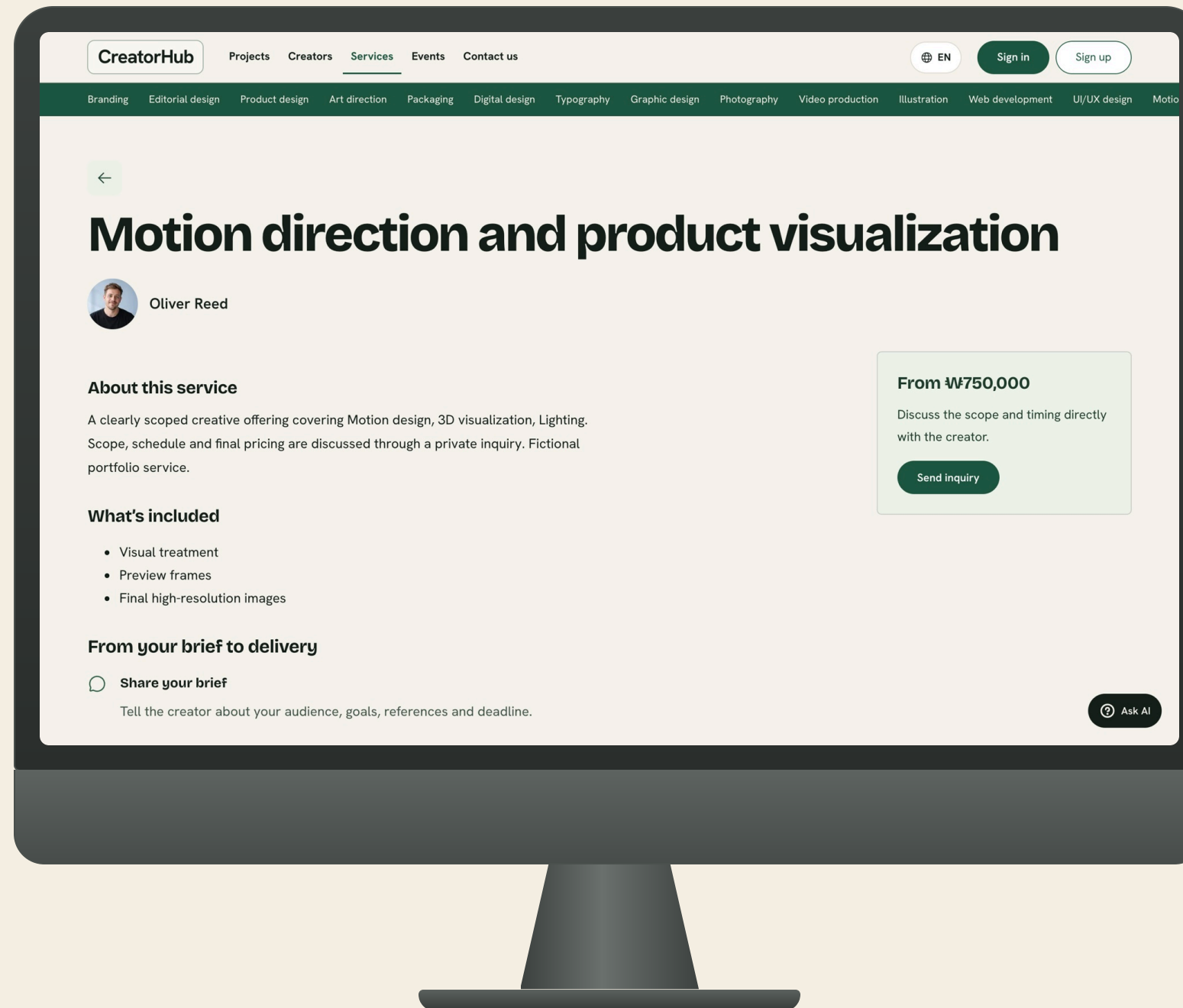
- ◆ 두 가지 가격 유형: 시작 가격 또는 맞춤 견적
- ◆ 문의를 제출하면 서비스 제목, 가격, 기간, 결과물이 문의에 복사
- ◆ 이후 서비스를 수정해도 이미 보낸 조건은 바뀌지 않음

GraphQL

MongoDB

내장 스냅샷

이 서비스 보기 ↗



크리에이티브 이벤트

관리자가 운영하는 크리에이터 커뮤니티의 워크숍, 토크, 모임입니다. 데모 이벤트는 가상임을 표시합니다.

구현 방식

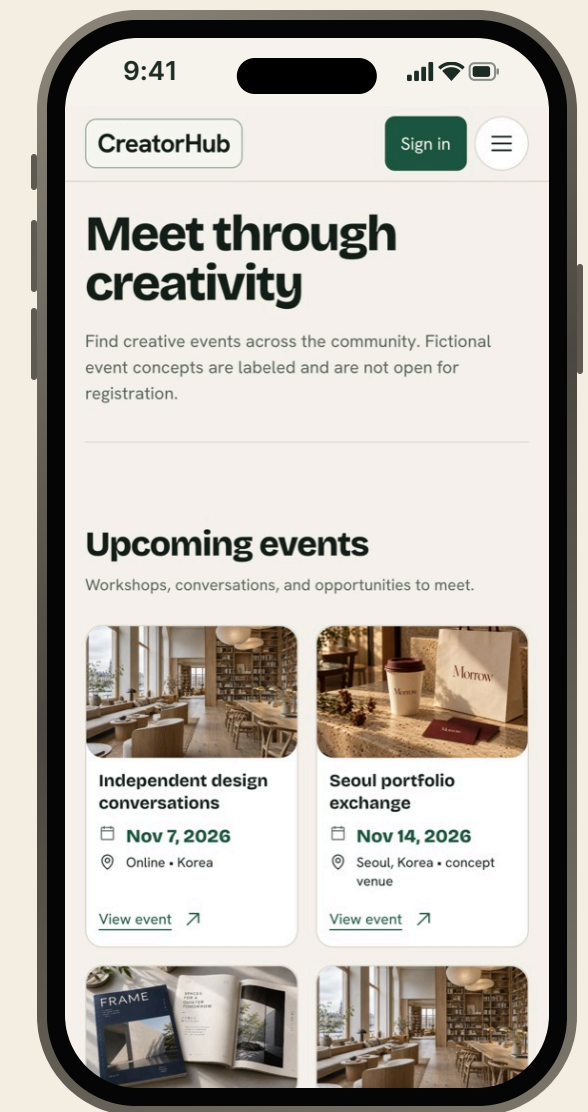
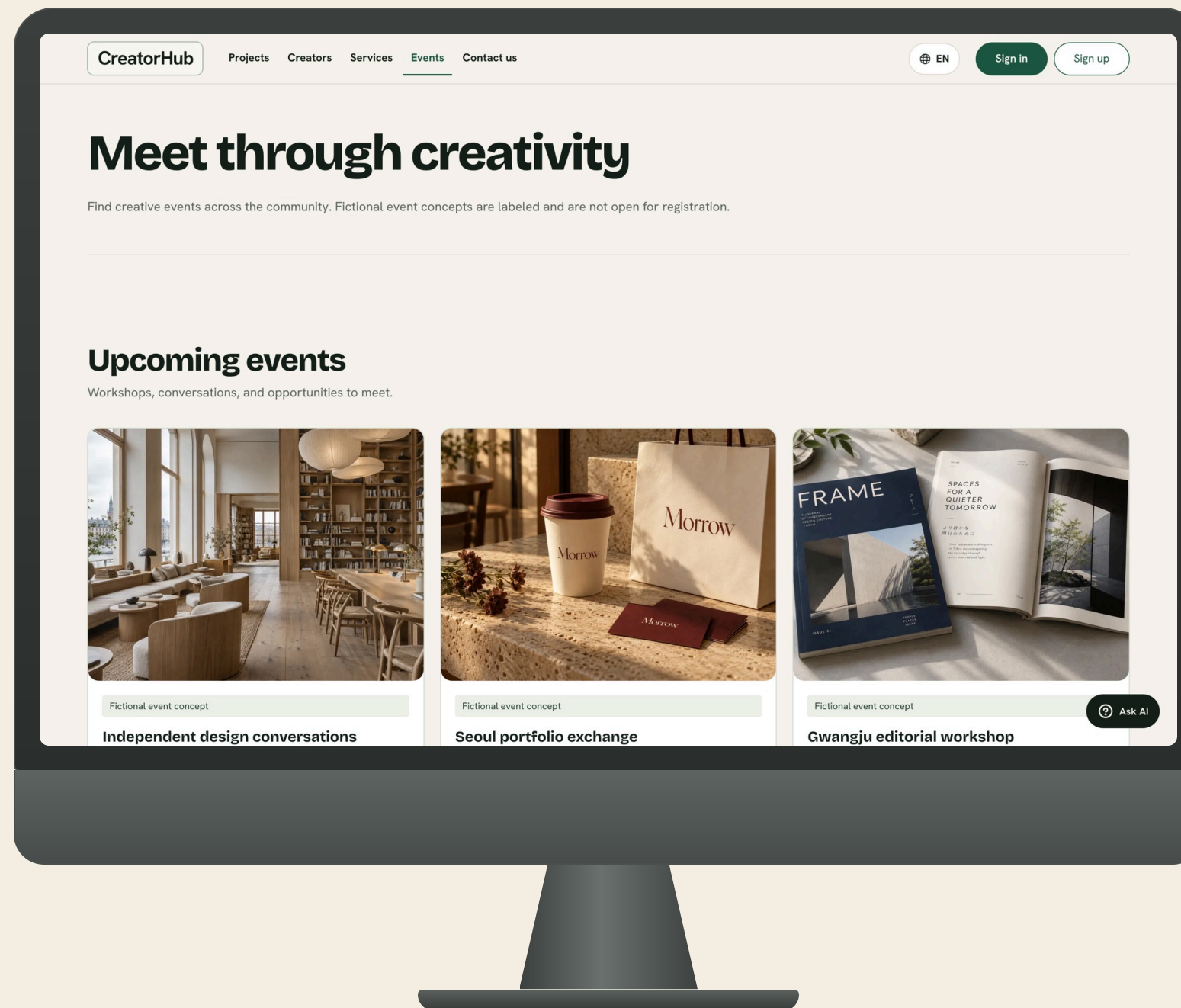
- ◆ 국내 도시 또는 온라인에서 열리는 공개 예정 이벤트만 가까운 순으로 표시
- ◆ 관리자는 시작 이후의 종료 시간과 HTTPS 링크를 입력해야 함
- ◆ 데모 플래그로 가상 예시 이벤트 표시

 GraphQL

 MongoDB

관리자 영역

creatorhub.site/events ↗



로그인 방식

이메일과 비밀번호, 또는 Google, Kakao, Telegram으로 로그인합니다. 새 소셜 계정은 이름과 전화번호를 입력해 가입을 마칩니다.

구현 방식

- ◆ Google, Kakao, Telegram 로그인은 각 제공자의 공개 키로 검증하고, Kakao는 일회용 state·nonce 값도 사용
- ◆ 비밀번호는 bcrypt로 해싱하고, 액세스 토큰은 15분간 유효
- ◆ 리프레시 토큰은 사용할 때마다 교체되며 해시로만 저장



JWT

OAuth / OpenID

bcrypt

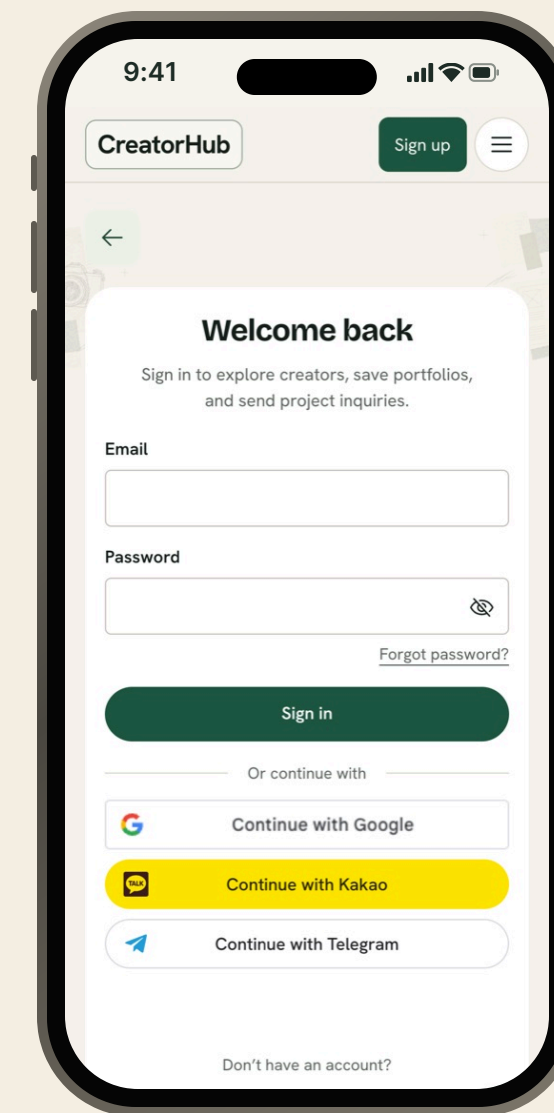
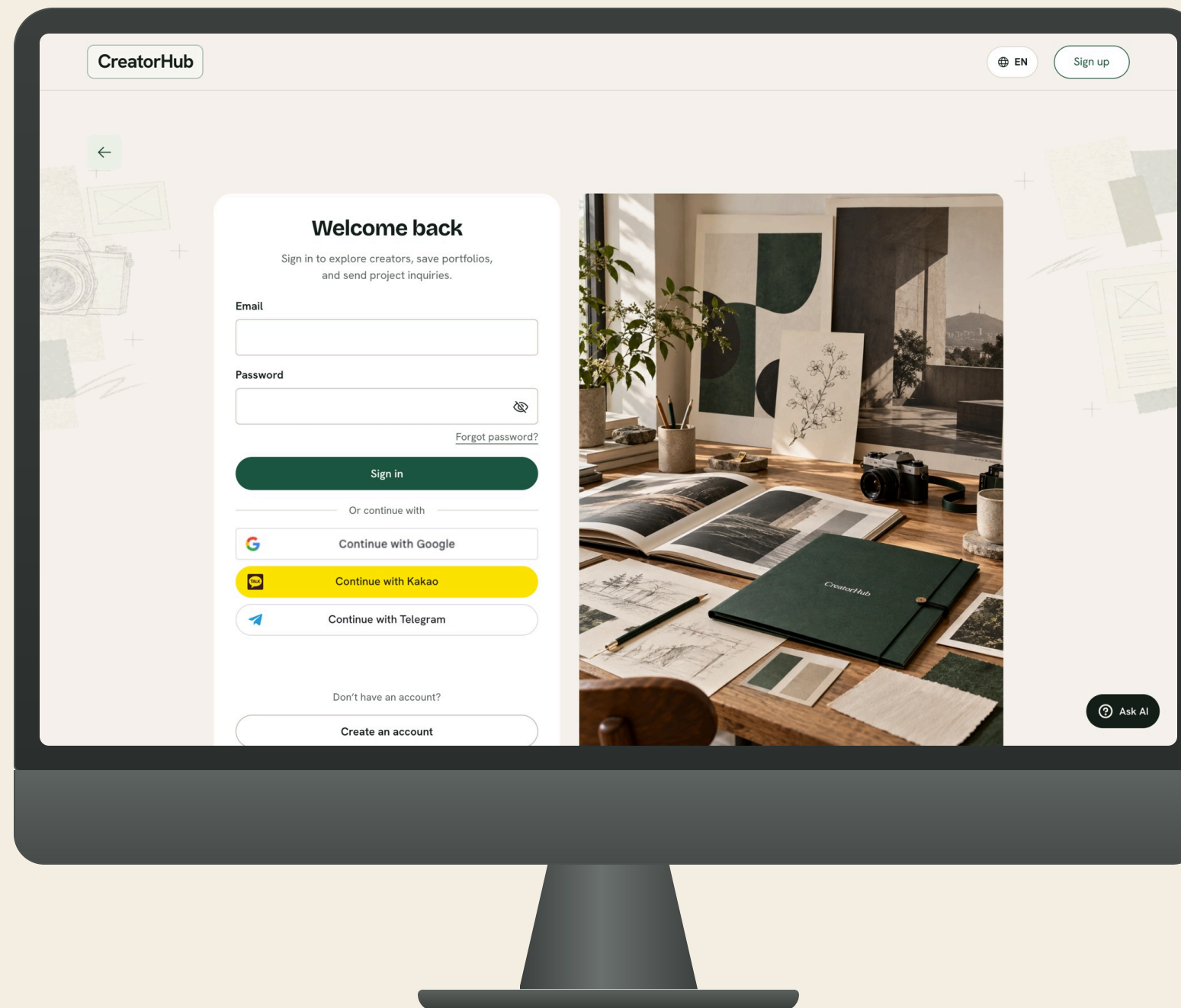


Kakao



Telegram

creatorhub.site/account/sign-in ↗



비공개 문의와 실시간 채팅

클라이언트는 예산과 마감일을 담은 의뢰서를 크리에이터에게 보내고, 비공개 채팅으로 논의합니다.



기본적으로 비공개

문의는 클라이언트와 크리에이터만 열 수 있습니다. 크리에이터는 작성 중인 초안을 볼 수 없습니다.

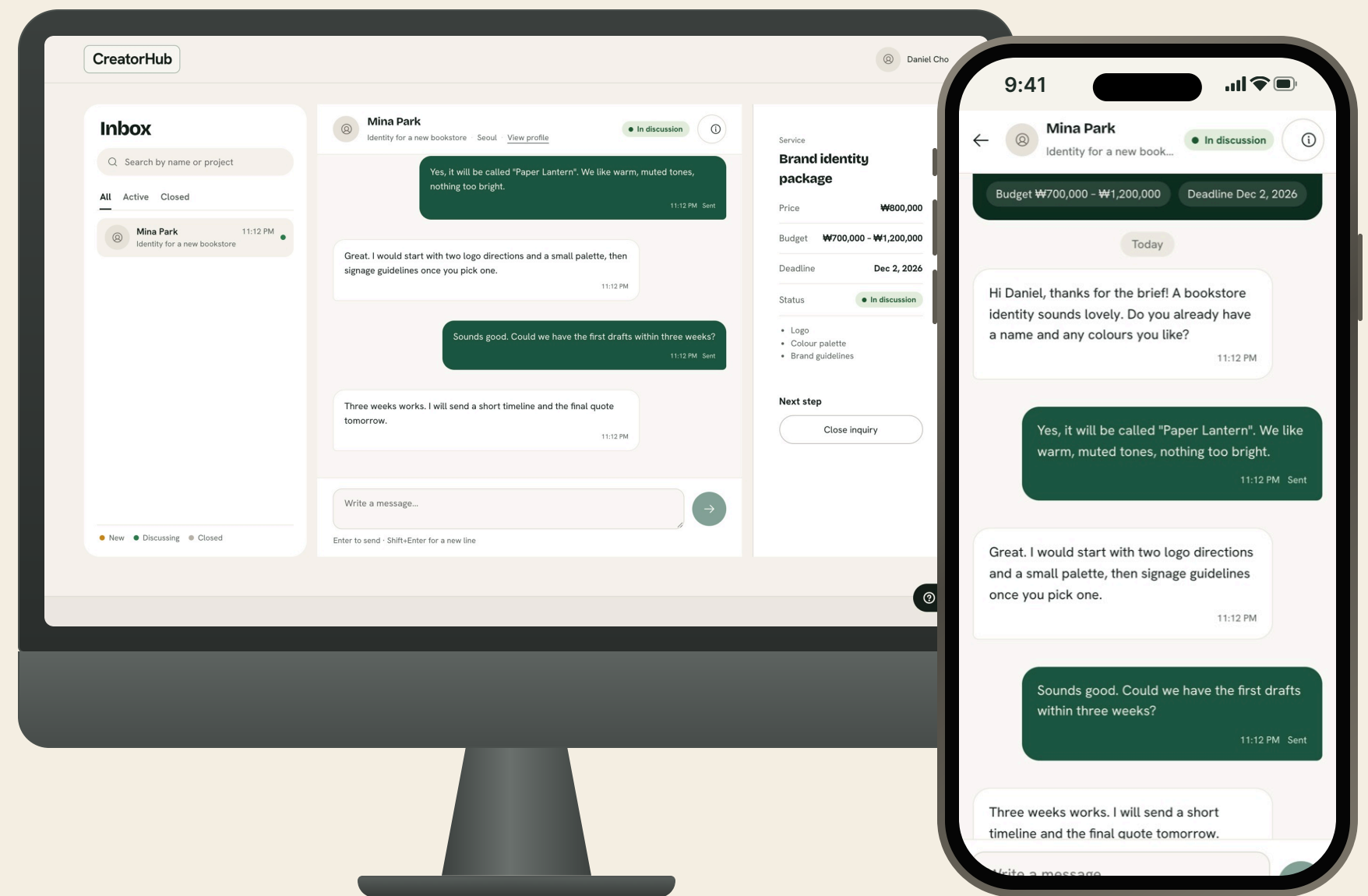
중복 방지

더블 클릭이나 네트워크 재시도로 문의나 메시지가 중복 생성되지 않습니다.

실시간 채팅

Socket.io 방도 같은 접근 검사를 거치며, 데이터베이스가 기준 데이터입니다.

현재 버전의 앱을 로컬에서 실행해 가상의 계정과 메시지로 캡처했습니다.



관리자: 개요와 사용자

별도의 관리자 영역에서 모든 데이터 유형의 합계를 확인하고 계정과 역할을 관리합니다.

구현 방식

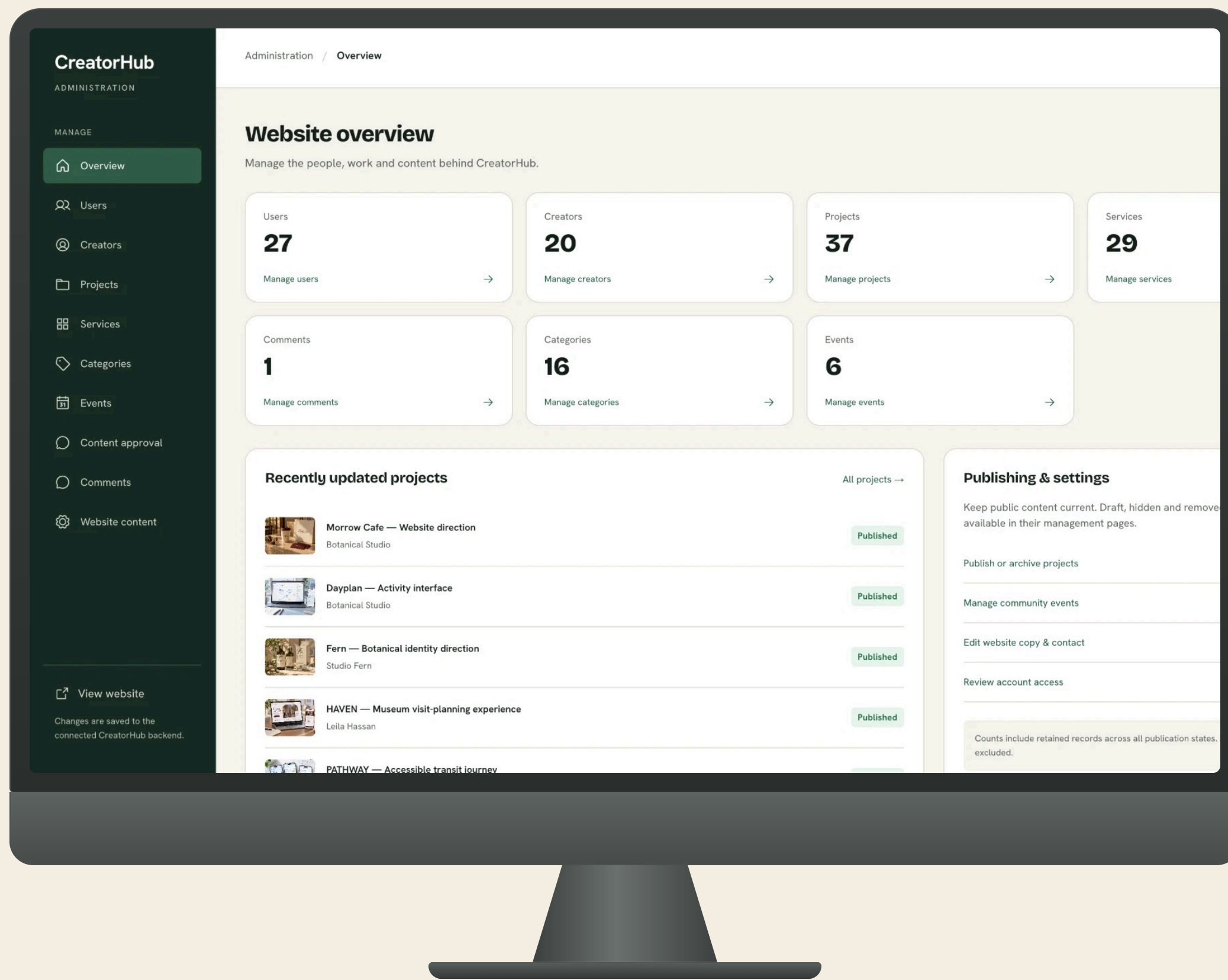
- ◆ 모든 관리자 쿼리는 메뉴 숨김이 아닌 서버 측 역할 가드로 보호
- ◆ 계정을 정지하거나 삭제하면 같은 트랜잭션에서 모든 세션 종료
- ◆ 관리자는 자신의 역할이나 상태를 바꿀 수 없어 스스로 접근을 잃지 않음
- ◆ 수정 시 열람 이후 변경 여부를 확인해 두 관리자가 서로 덮어쓰지 않음

🐉 역할 가드

세션 폐기

낙관적 동시성 제어

🌿 MongoDB 트랜잭션



관리자: 검수와 콘텐츠

관리자는 숨김·삭제된 항목을 포함한 모든 크리에이터, 프로젝트, 서비스를 검토하고 사이트 문구를 수정합니다.

구현 방식

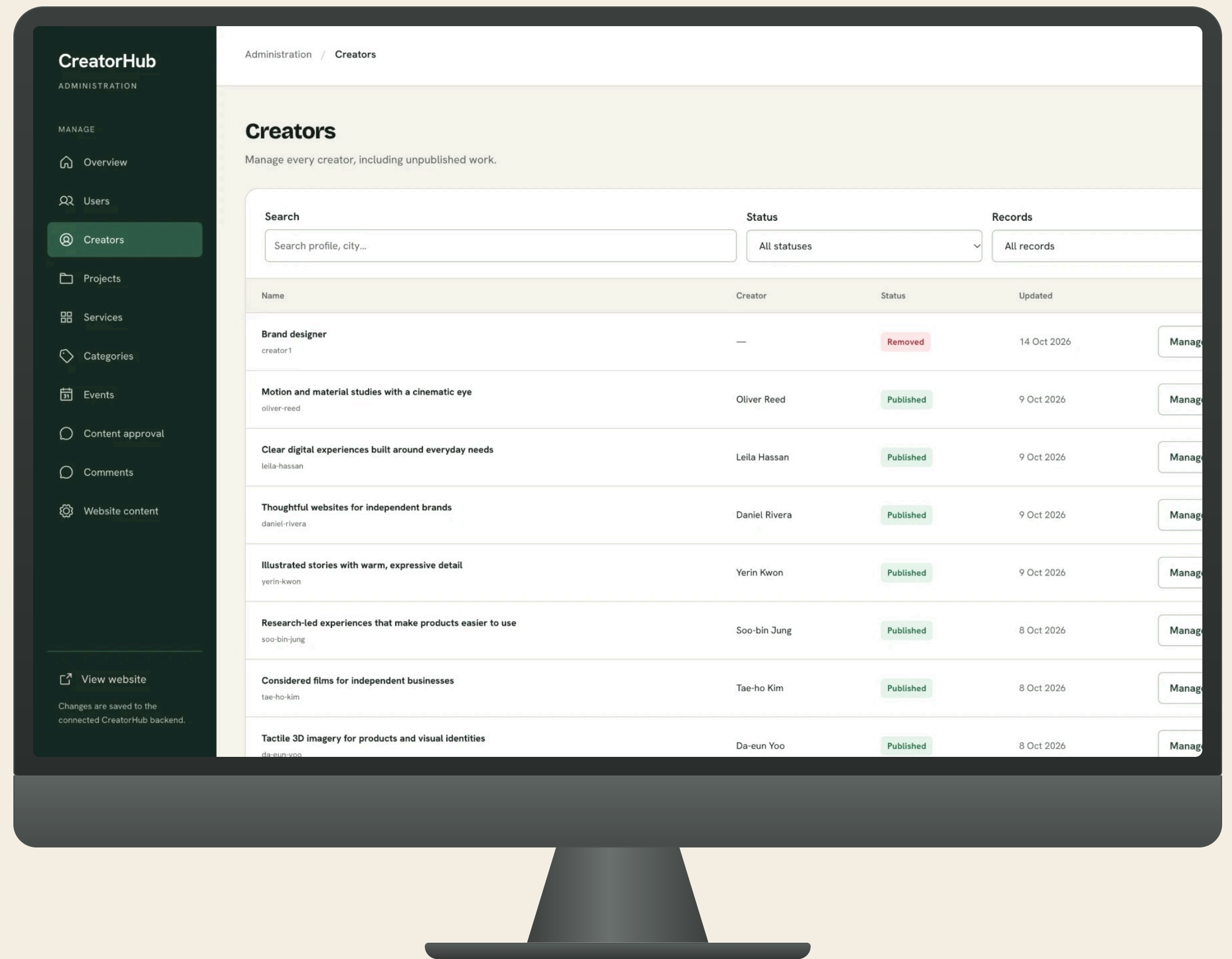
- ◆ 관리자는 댓글을 숨겼다가 다시 표시할 수 있으며, 작성자가 삭제한 댓글은 복구되지 않음
- ◆ 삭제는 소프트 삭제이며, 복원해도 자동으로 다시 공개되지 않음
- ◆ 홈, 소개, 푸터 문구를 데이터베이스에 저장해 재배포 없이 반영

콘텐츠 검수

소프트 삭제

 GraphQL

 MongoDB



크리에이터 워크스페이스와 검수

크리에이터는 비공개 워크스페이스에서 작업을 수정합니다. 관리자가 승인하기 전에는 아무것도 공개되지 않습니다.

- 1 크리에이터가 수정 사항을 저장합니다. 공개 중인 버전은 그대로 유지됩니다.
- 2 검토 대기 항목이 저장됩니다. 항목당 하나이며, 다시 저장하면 교체됩니다.
- 3 관리자가 승인 또는 거절합니다. 거절 시에는 사유를 작성해야 합니다.

공개 체크리스트

카테고리, 설명, 커버,
미디어가 필요하며 승인 시
다시 확인합니다.

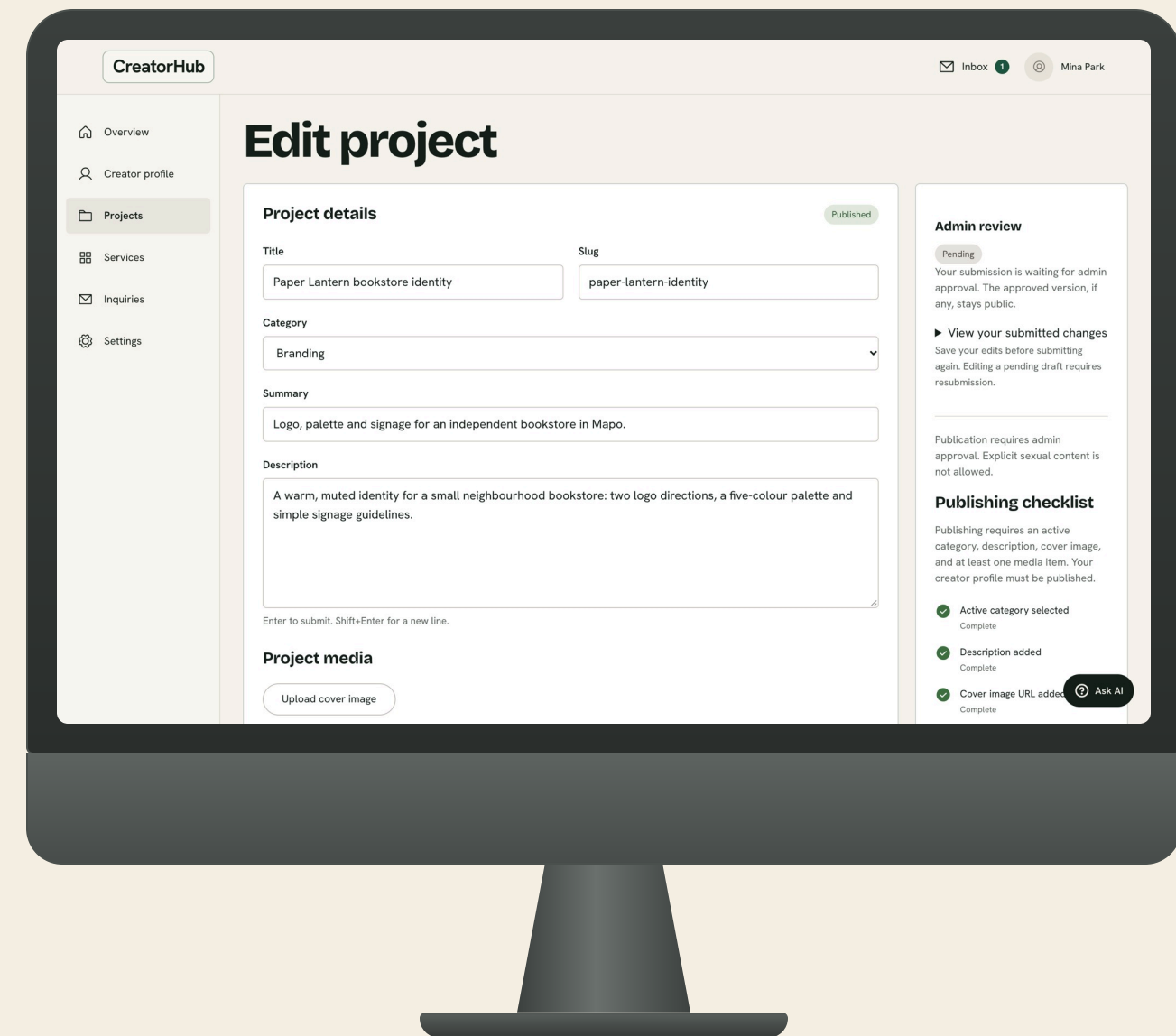
관리자 전용 결정

관리자 역할은 화면뿐
아니라 서버에서도
확인합니다.

오래된 승인 방지

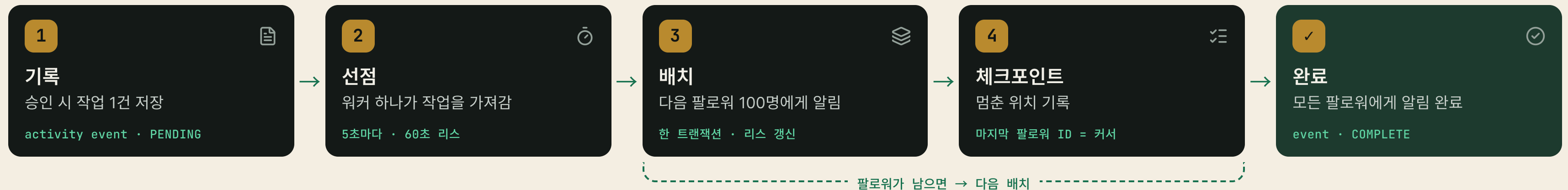
검토 저장 이후 항목이
바뀌면 승인이 실패합니다.

현재 버전의 앱을 로컬에서 실행해 가상의 계정과 콘텐츠로 캡처했습니다.



워커가 재시작되어도 유지되는 팔로워 알림

문제. 크리에이터가 새 작업을 공개하면 모든 팔로워에게 알림이 갑니다. 워커가 중간에 멈추면 일부 팔로워는 알림을 두 번 받거나 아예 받지 못할 수 있습니다.



⚡ **중단이나 오류가 나면?** 오류가 나면 30초 후 재시도하도록 작업을 풀어 주고, 워커가 멈추면 60초 리스가 만료됩니다. 어느 경우든 다음 작업은 저장된 커서부터 이어갑니다.



중복 방지

알림은 팔로워·이벤트마다 하나뿐이라, 배치가 반복돼도 같은 사람에게 두 번 알리지 않습니다.



알려진 한계

MongoDB 폴링으로 별도 큐 서비스는 필요 없지만 몇 초의 지연이 생깁니다. 아직 재시도 횟수 제한이 없어, 다음 단계로 일정 횟수 후 중단하고 관리자에게 표시할 계획입니다.



검증

통합 테스트로 롤백된 배치가 중복 없이 재시도되는지, 첫 공개 때만 팔로워에게 알리는지 확인합니다.

범위: 앱 내 알림만 해당하며, 팔로워·이벤트당 최대 1건입니다. 이메일이나 푸시는 보내지 않습니다.

Gemini 기반 AI 파인더

모든 페이지에 있는 어시스턴트입니다. 방문자가 필요한 것을 일상 언어로 설명하면, 맞는 크리에이터와 서비스 링크가 담긴 답변을 받습니다.

💬 “원격으로 일할 수 있는 UI/UX 디자이너가 필요해요. 누구를 보면 될까요?”

- ◆ 사이트 언어에 맞춰 영어, 한국어, 우즈베크어로 답변
- ◆ 답변마다 해당 검색 결과 링크 제공
- ◆ 데스크톱에서는 사이드 패널, 휴대폰에서는 전체 화면

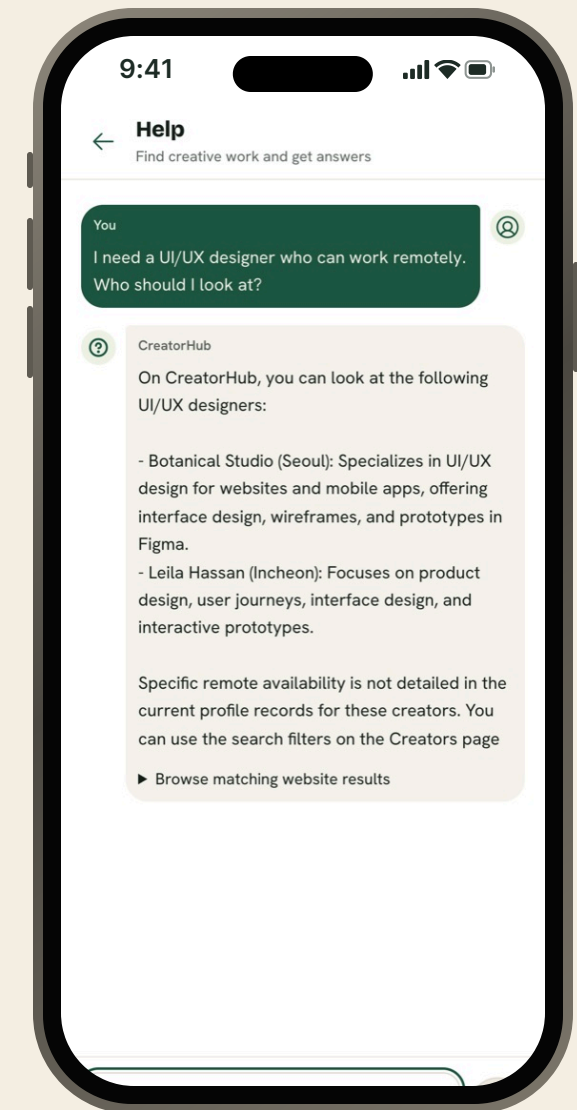
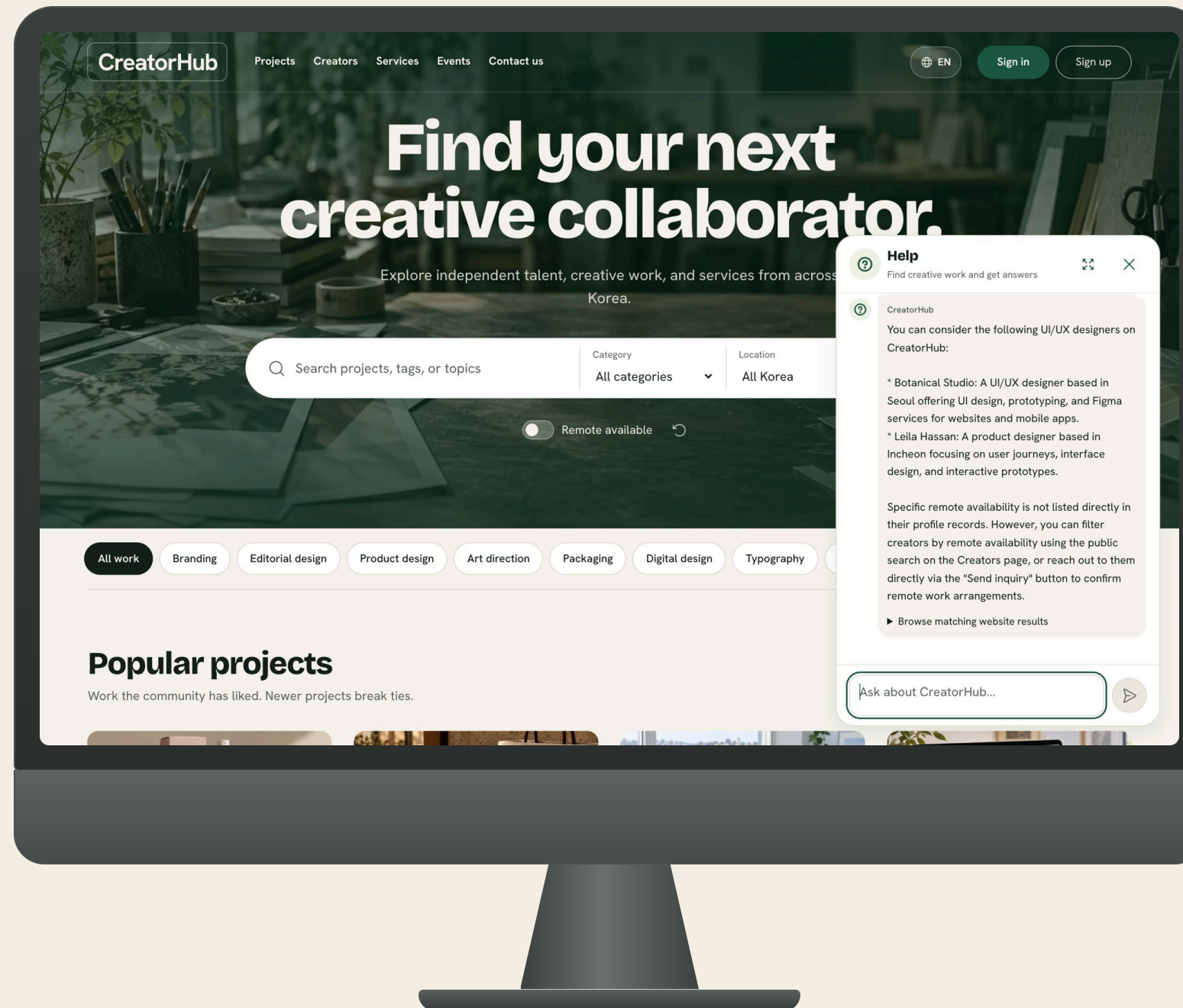
🌟 Gemini API

Next.js API 라우트

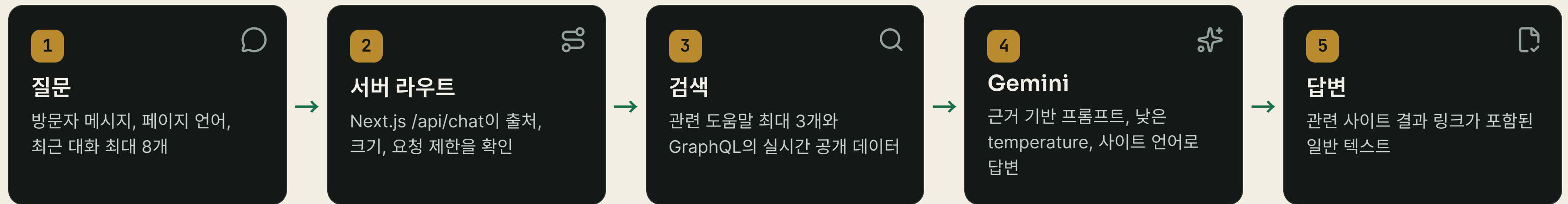
🔗 GraphQL

AI 파인더 열기 ↗

creatorhub.site에서 캡처한 실제 답변입니다. 답변에 나온 크리에이터는 사이트의 데모 프로필입니다.



AI 파인더의 답변 과정



안전

- ◆ API 키는 서버에만 보관
- ◆ 공개·승인된 데이터만 조회
- ◆ 소스와 대화 기록은 지시가 아닌 데이터로 처리

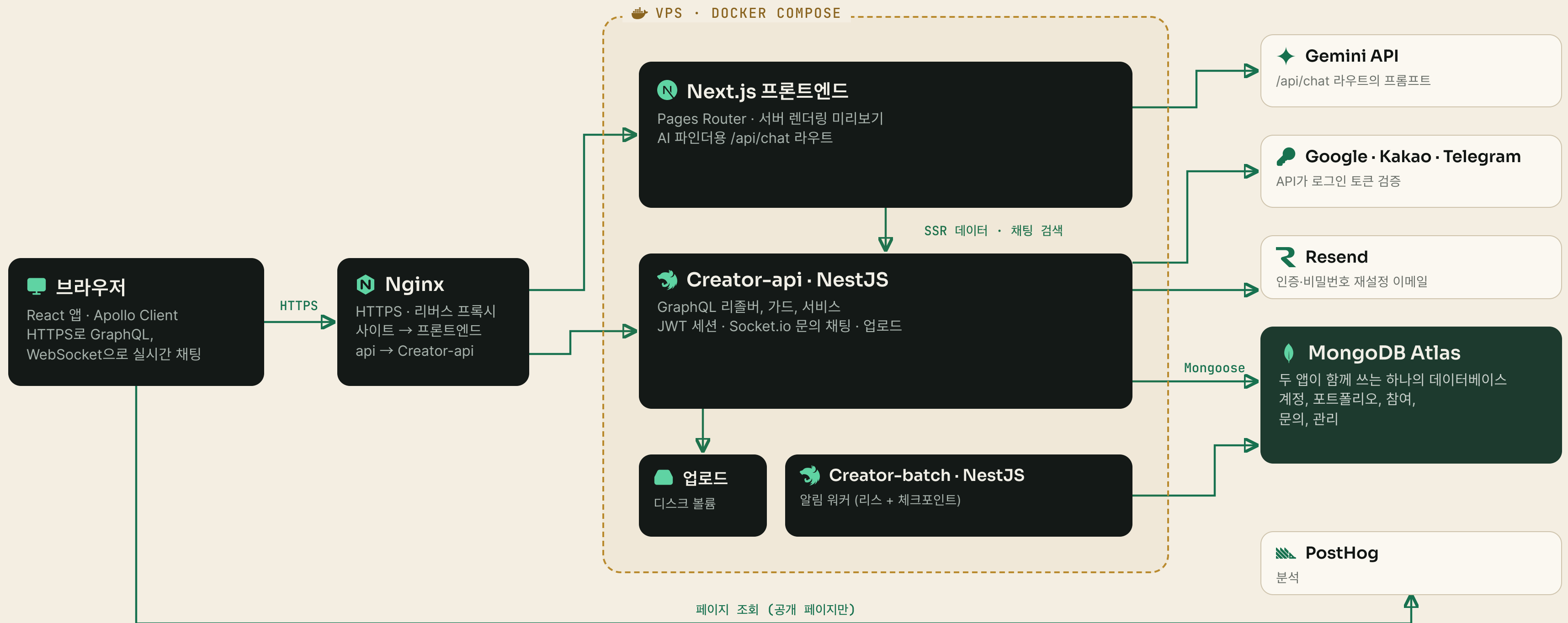
제한

- ◆ 같은 사이트 요청만 허용
- ◆ 메시지는 최대 1,500자
- ◆ 방문자당 분당 8개, 일일 한도 별도

복원력

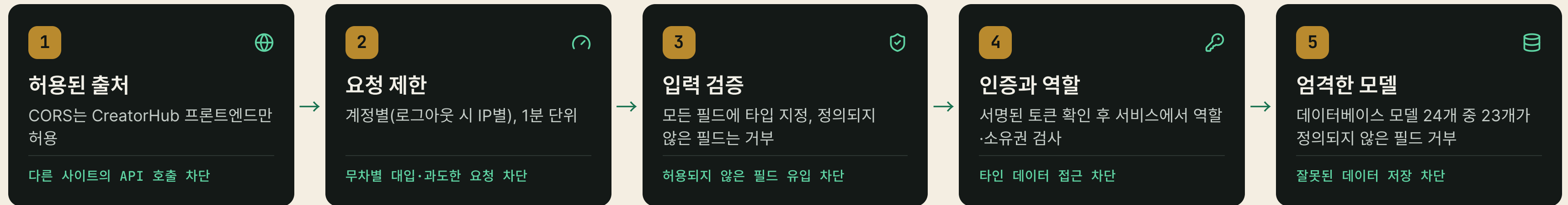
- ◆ 조회는 3.5초 후 중단
- ◆ 모델이 바쁘면 대체 모델로 한 번 재시도
- ◆ 키가 없으면 도움말로 안내

세 개의 애플리케이션, 하나의 데이터 모델



모든 요청을 보호하는 방식.

모든 GraphQL 요청은 데이터베이스에 닿기 전에 같은 검사를 이 순서대로 거칩니다.



변경 내용 유실 방지
문의와 수정 대상 데이터에는 버전이 있어, 오래된 사본으로 저장하면 최신 변경을 덮어쓰지 않고 충돌을 반환합니다.

안전한 업로드
파일당 10MB, 최대 10개까지 업로드합니다. 미디어 링크는 소유자 폴더 안의 허용된 형식만 가능합니다.

비밀 정보는 서버에만
브라우저에는 공개 ID와 URL만 전달됩니다. API 키, 토큰 비밀값, 비밀번호 해시는 백엔드 밖으로 나가지 않습니다.

로그인 세션, 비공개 문의, AI 제한은 각 페이지에서 다룹니다.

테스트와 CI/CD

두 저장소 모두 푸시할 때마다 테스트합니다. 배포는 수동으로 시작하는 별도 워크플로를 거칩니다.

— 테스트 범위

백엔드 단위 · Jest

비즈니스 규칙: 문의 상태, 비밀번호, 요청 제한

API 통합 · Jest

일회용 MongoDB 레플리카 셋에서 실제 GraphQL 요청

프론트엔드 · Node.js

세션, 로그인, AI 채팅 라우트, 번역

정적 검사

TypeScript, ESLint, 모든 쿼리를 스키마와 대조

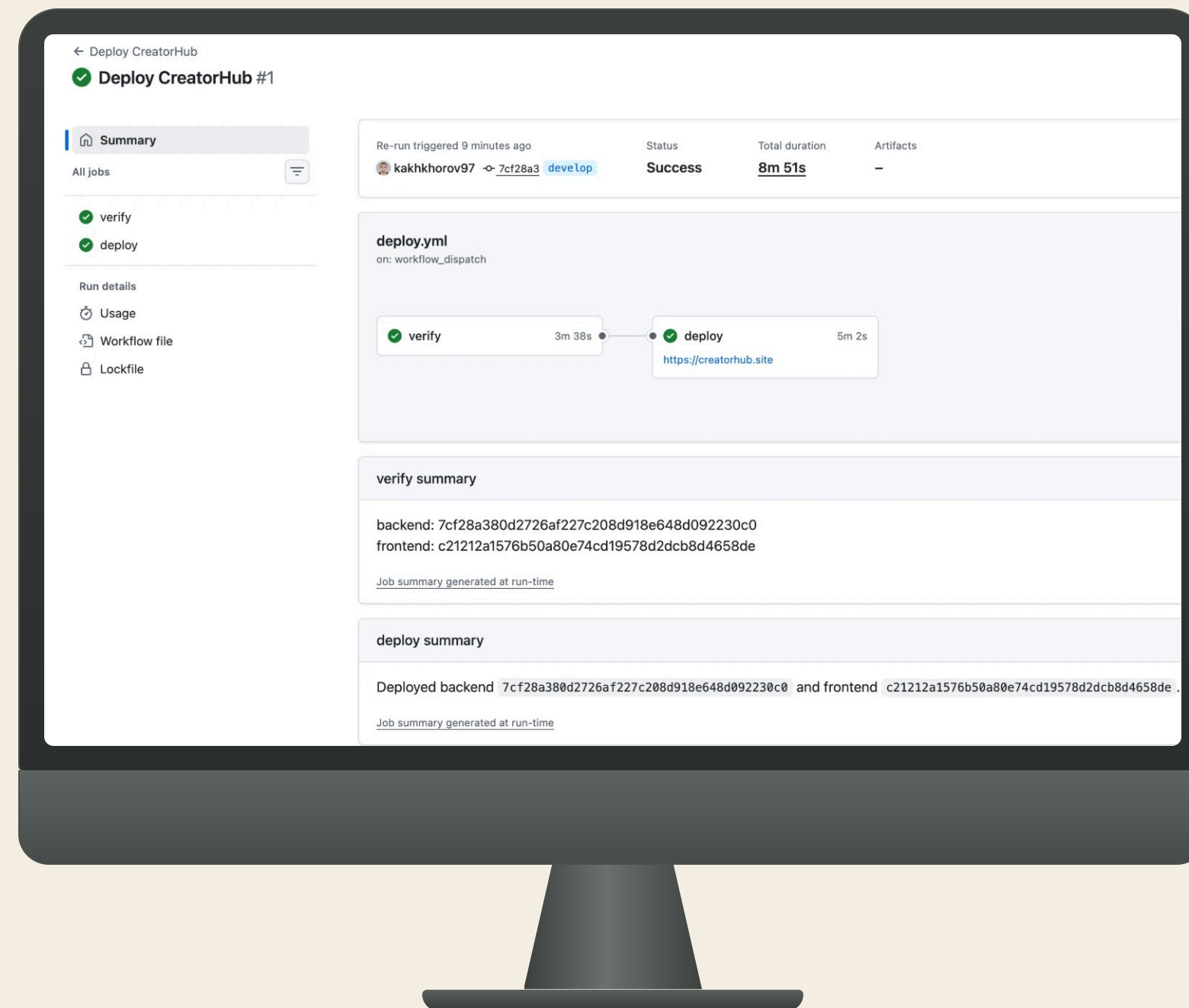
local re-run · 17 Oct 2026

```
$ backend jest ✓ 162/162 tests · 16 suites
$ backend deploy scripts ✓ 23/23
$ frontend node --test ✓ 108/108
$ both typecheck · lint · contract (119 ops) ✓ passed
```

— 배포 과정

- 1 **develop-main에 푸시할 때마다** 설치, 타입 검사, 린트, 테스트, 빌드를 실행합니다.
- 2 **배포 워크플로**는 수동으로 시작하며, 백엔드·프론트엔드 커밋 한 쌍에 대해 모든 검사를 다시 실행합니다.
- 3 **제한된 SSH**로 “deploy”와 커밋 ID 두 개만 받는 배포 전용 계정에 접속합니다.
- 4 **헬스 체크 전환**: 새 이미지를 실행하고 최대 180초간 상태를 확인하며, 실패하면 이전 이미지로 되돌립니다.

화면: 첫 Deploy CreatorHub 실행(2026년 10월 18일). verify와 deploy가 모두 통과해 creatorhub.site에 배포되었습니다.



운영 환경에서의 실행

프론트엔드, API, 워커는 Nginx와 HTTPS 뒤의 Linux VPS에서 Docker Compose 컨테이너로 실행되며, 데이터베이스는 MongoDB Atlas에 있습니다.



강화된 컨테이너

루트가 아닌 사용자, 읽기 전용 파일 시스템, 모든 Linux 권한 제거, 권한 상승 차단. 비밀 정보는 이미지가 아닌 실행 시 env 파일에서 읽습니다.



헬스 체크

컨테이너마다 30초마다 상태를 확인하고, 세 번 실패하면 비정상으로 표시하며, 중단된 컨테이너는 재시작됩니다. 헬스 체크는 앱의 응답 여부만 확인할 뿐 모든 기능을 보장하지는 않습니다.



릴리스와 롤백

빌드마다 고정 태그를 붙이고 헬스 체크를 통과해야 하며, GitHub 배포 경로는 실패 시 이전 이미지로 되돌립니다. 데이터베이스 변경은 롤백하지 않습니다.

2026년 10월 17-18일 확인

- ✓ creatorhub.site가 HTTPS로 페이지 제공
- ✓ API 헬스 엔드포인트가 ok 반환
- ✓ Compose 파일에서 컨테이너 설정 확인
- ✓ 첫 GitHub 배포 실행 통과 (2026년 10월 18일)

다음 단계

- 서버 외부 사본과 복원 테스트를 포함한 Docker용 백업
- 외부 가동 시간 알림

PostHog 기반 공개 페이지 분석

페이지 조회는 공개 페이지 15개에서만 기록합니다. 계정, 워크스페이스, 문의, 관리자 페이지는 추적하지 않습니다.

- ◆ IP 주소와 위치 조회 없음
- ◆ 세션 녹화, 히트맵, 설문, 개인 프로필 없음
- ◆ 분석 오류는 처리되어 페이지 이동을 방해하지 않음



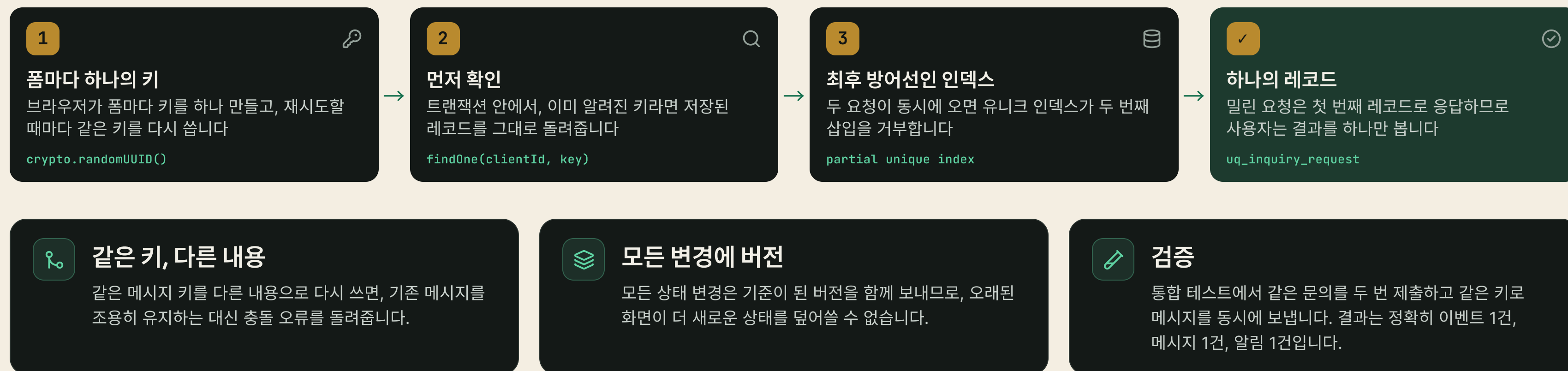
PostHog · Activity · creatorhub.site 페이지 조회 기록

EVENT	PERSON	URL / SCREEN	LIBRARY	TIME	
Pageview	01a144d0-b072-77a1-a99f-7b53868d3100	https://creatorhub.site/	web	17 minutes ago	
Pageview	01a14024-d5e3-72fc-a711-2da1f4583527	https://creatorhub.site/	web	18 minutes ago	
Pageview	01a14455-106d-71dc-99f8-b01e27eaa469	https://creatorhub.site/	web	3 hours ago	
Pageview	01a14392-656b-70bf-ab20-81cb7031588c	https://creatorhub.site/	web	6 hours ago	
Pageview	01a1438d-0dd6-7ecb-a865-97c03a9eb366	https://creatorhub.site/	web	6 hours ago	
Pageview	01a1437a-24e0-73eb-bc57-540ae9c46a2f	https://creatorhub.site/	web	7 hours ago	
Pageview	01a142a3-448e-70bb-b111-1e40cd50e0cc	https://creatorhub.site/	web	10 hours ago	
October 15, 2026					
Pageview	01a143b2-be7f-7851-b968-543a727c341f	https://creatorhub.site/faq	web	16 hours ago	
Pageview	01a143b2-be7f-7851-b968-543a727c341f	https://creatorhub.site/	web	16 hours ago	

2026년 10월 16일 캡처. 수치는 고유 방문자가 아닌 페이지 조회 수이며, 제 테스트 방문이 포함될 수 있습니다.

한 번 클릭, 하나의 문의: 재시도에 안전한 요청

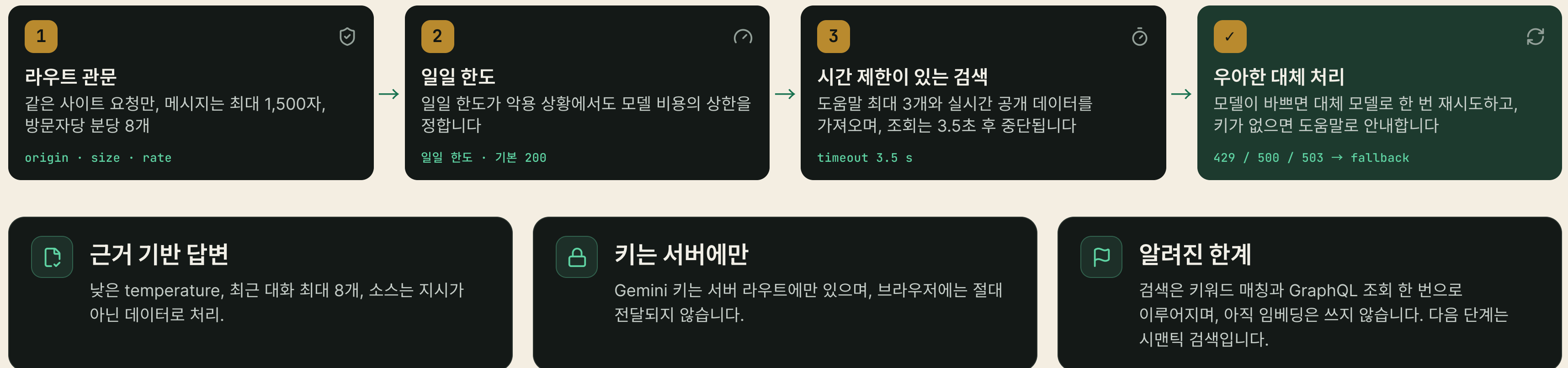
문제. 더블 클릭이나 끊긴 연결 때문에 같은 문의나 채팅 메시지가 두 번 전송될 수 있었습니다. 서버는 이를 새 요청 두 개로 보고 둘 다 저장해, 크리에이터에게 알림도 두 번 보낼 수 있었습니다.



좋아요, 저장, 팔로우, 댓글 재시도에도 같은 패턴을 적용했습니다.

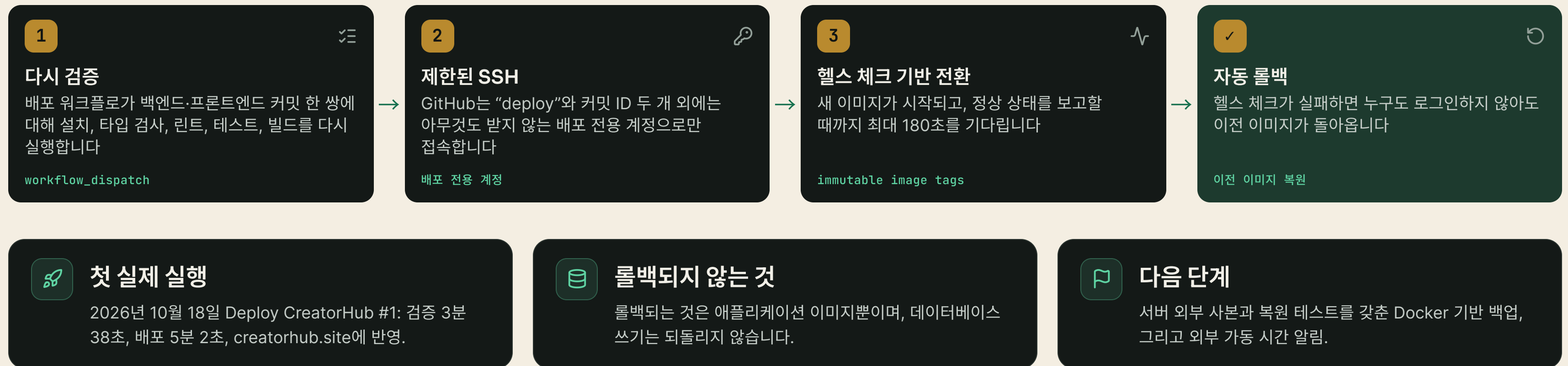
AI 파인더를 빠르고, 비용 부담 없이, 안전하게 .

문제. AI 파인더는 모든 방문자에게 열려 있고, 답변 하나하나가 유료 Gemini 호출입니다. 제한이 없으면 스크립트 하나로 요청이 폭주하고, 느린 조회는 방문자를 기다리게 하며, 모델이 바쁘면 오류가 표시됩니다.



스스로 롤백하는 배포

문제. CreatorHub는 VPS 한 대에서 운영됩니다. 잘못된 배포는 누군가 알아차리고 직접 고칠 때까지 라이브 사이트를 멈추게 할 수 있었습니다.



CreatorHub ER model

All collections expanded · MongoDB · lines connect each parent _id to the referencing field

SOLID LINE = ObjectId REFERENCE

DASHED = EMBEDDED SCHEMA

ACCOUNTS

users		
PK	_id	objectId
	displayName	string
	phoneNumber	string
	avatarUrl	string
	role	enum
	status	enum
	preferredLocale	enum
	lastLoginAt	date
	deletedAt	date
	createdAt	date
	updatedAt	date

authIdentities		
PK	_id	objectId
FK	userId	objectId
	provider	enum
	providerSubject	string
	passwordHash	string
	verifiedAt	date
	revokedAt	date
	lastLoginAt	date
	createdAt	date
	updatedAt	date

authSessions		
PK	_id	objectId
FK	userId	objectId
FK	authIdentityId	objectId
	refreshTokenHash	string
	expiresAt	date
	revokedAt	date
	lastUsedAt	date
	createdAt	date
	updatedAt	date

authTokens		
PK	_id	objectId
FK	authIdentityId	objectId
	purpose	enum
	tokenHash	string
	previousEmail	string
	replacementEmail	string
	expiresAt	date
	revokedAt	date
	consumedAt	date
	createdAt	date

PORTFOLIO

creatorProfiles		
PK	_id	objectId
FK	userId	objectId
	slug	string
	headline	string
	bio	string
	city	string
	remoteAvailable	bool
	availability	enum
FK	categoryIds	objectId[]
	skills	string[]
	socialLinks	SocialLink[]
	status	enum
	publishedAt	date
	deletedAt	date
	createdAt	date
	updatedAt	date

SocialLink		
	label	string
	uri	string

categories		
PK	_id	objectId
FK	slug	string
	name	string
	description	string
	sortOrder	int
	isActive	bool
	createdAt	date
	updatedAt	date

portfolioProjects		
PK	_id	objectId
FK	creatorId	objectId
FK	categoryId	objectId
	title	string
	slug	string
	summary	string
	description	string
	coverImageUrl	string
	media	PortfolioMedia[]
	tags	string[]
	externalUri	string
	status	enum
	publishedAt	date
	deletedAt	date
	createdAt	date
	updatedAt	date

PortfolioMedia		
	type	enum
	uri	string
	altText	string
	sortOrder	int

services		
PK	_id	objectId
FK	creatorId	objectId
FK	categoryId	objectId
	title	string
	description	string
	imageUrl	string
	pricingType	enum
	startingPrice	int
	currency	enum
	estimatedDays	int
	deliverables	string[]
	status	enum
	deletedAt	date
	createdAt	date
	updatedAt	date

ENGAGEMENT

creatorFollows		
PK	_id	objectId
FK	followerId	objectId
FK	creatorId	objectId
	isActive	bool
	followedAt	date
	createdAt	date
	updatedAt	date

projectLikes		
PK	_id	objectId
FK	userId	objectId
FK	projectId	objectId
	createdAt	date

savedProjects		
PK	_id	objectId
FK	userId	objectId
FK	projectId	objectId
	createdAt	date

projectComments		
PK	_id	objectId
FK	userId	objectId
FK	projectId	objectId
	clientCommentKey	string
	content	string
	status	enum
	editedAt	date
	deletedAt	date
	createdAt	date
	updatedAt	date

activityEvents		
PK	_id	objectId
FK	performedById	objectId
FK	creatorId	objectId
FK	projectId	objectId
FK	commentId	objectId
	actionKey	string
	deliveryStatus	enum
	deliveryCursor	string
	leaseUntil	date
	nextAttemptAt	date
	attempts	number
	completedAt	date
	createdAt	date
	updatedAt	date

INQUIRIES

inquiries		
PK	_id	objectId
	clientRequestKey	string
FK	clientId	objectId
FK	creatorId	objectId
FK	servicId	objectId
	subject	string
	brief	string
	budgetMin	int
	budgetMax	int
	currency	enum
	desiredDeadline	date
	serviceSnapshot	InquiryServiceSnapshot
	status	enum
	outcome	enum
FK	closedById	objectId
	closedAt	date
	submittedAt	date
	lastMessageAt	date
	version	int
	createdAt	date
	updatedAt	date

InquiryServiceSnapshot		
	title	string
	pricingType	enum
	startingPrice	int
	currency	enum
	estimatedDays	int
	deliverables	string[]

inquiryMessages		
PK	_id	objectId
FK	inquiryId	objectId
FK	senderId	objectId
	clientMessageKey	string
	body	string
	readAt	date
	createdAt	date

inquiryEvents		
PK	_id	objectId
FK	inquiryId	objectId
FK	performedById	objectId
	type	enum
	fromStatus	enum
	toStatus	enum
FK	messageId	objectId
	version	int
	createdAt	date

serviceFeedback		
PK	_id	objectId
FK	inquiryId	objectId
FK	clientId	objectId
FK	creatorId	objectId
FK	servicId	objectId
	rating	int
	feedback	string
	createdAt	date
	updatedAt	date

ADMINISTRATION

contentReviews		
PK	_id	objectId
	kind	enum
	targetId	objectId
FK	ownerId	objectId
	title	string
	snapshot	mixed
	baseUpdatedAt	date
	status	enum
	reason	string
FK	reviewedById	objectId
	createdAt	date
	updatedAt	date

notifications		
PK	_id	objectId
FK	recipientId	objectId
FK	inquiryEventId	objectId
FK	activityEventId	objectId
	inboxReadAt	date
	readAt	date
	createdAt	date

creativeEvents		
PK	_id	objectId
	title	string
	description	string
	startsAt	date
	endsAt	date
	location	string
	eventUrl	string
	status	enum
	isDemo	bool
	createdAt	date
	updatedAt	date

siteContent		
PK	_id	objectId
	key	enum
	homeHeading	string
	homeSubheading	string
	aboutText	string
	contactEmail	string
	contactText	string
	footerCopy	string
	createdAt	date
	updatedAt	date

Inquiries, 필드 단위로

PK 기본 키 FK → 참조 컬렉션  내장 스키마

inquiries		
PK	_id	objectId
	clientRequestKey	string
FK	clientId	→ users
FK	creatorId	→ creatorProfiles
FK	serviceId	→ services
	subject	string
	brief	string
	budgetMin	int
	budgetMax	int
	currency	enum
	desiredDeadline	date
	serviceSnapshot	embedded
	status	enum
	outcome	enum
FK	closedById	→ users
	closedAt	date
	submittedAt	date
	lastMessageAt	date
	version	int
	createdAt	date
	updatedAt	date

InquiryServiceSnapshot		
embedded in inquiries.serviceSnapshot		
	title	string
	pricingType	enum
	startingPrice	int
	currency	enum
	estimatedDays	int
	deliverables	string[]

inquiryMessages		
PK	_id	objectId
FK	inquiryId	→ inquiries
FK	senderId	→ users
	clientMessageKey	string
	body	string
	readAt	date
	createdAt	date

inquiryEvents		
PK	_id	objectId
FK	inquiryId	→ inquiries
FK	performedById	→ users
	type	enum
	fromStatus	enum
	toStatus	enum
FK	messageId	→ inquiryMessages
	version	int
	createdAt	date

serviceFeedback		
PK	_id	objectId
FK	inquiryId	→ inquiries
FK	clientId	→ users
FK	creatorId	→ creatorProfiles
FK	serviceId	→ services
	rating	int
	feedback	string
	createdAt	date
	updatedAt	date

데이터 설계 방식

API와 워커가 함께 쓰는 하나의 MongoDB 데이터베이스에 21개 컬렉션이 있습니다. 참조는 서비스에서 검사하고, 절대 깨지면 안 되는 규칙은 유니크 인덱스로 강제합니다.

계정

users authIdentities
authSessions authTokens

- ◆ 로그인 방식마다 별도 레코드라, 한 사람이 이메일·Google·Kakao를 함께 사용 가능
- ◆ 리프레시·이메일 토큰은 해시로만 저장되고 자동 만료
- ◆ 세션이 레코드로 저장되어, 사용자 정지 시 모든 세션 종료

포트폴리오

creatorProfiles categories
portfolioProjects services

- ◆ 프로필과 크리에이터별 프로젝트에 읽기 쉬운 고유 슬러그
- ◆ 소셜 링크와 프로젝트 미디어는 항상 상위 문서와 함께 읽으므로 임베드
- ◆ 인덱스는 공개 피드에 맞춰 상태, 최신순으로 구성

참여

creatorFollows projectLikes
savedProjects projectComments
activityEvents

- ◆ 좋아요·저장·팔로우는 사람·항목당 하나로, 유니크 인덱스로 강제
- ◆ 언팔로우는 삭제하지 않고 팔로우를 비활성화
- ◆ 활동 이벤트에 워커의 리스와 커서를 저장해 안전하게 전달

문의

inquiries inquiryMessages
inquiryEvents serviceFeedback

- ◆ 서비스 조건은 임베드된 스냅샷으로 문의에 복사
- ◆ 재시도 키와 부분 유니크 인덱스로 문의·메시지 중복 방지
- ◆ 모든 상태 변경은 번호가 매겨진 이벤트이며, 완료된 문의당 리뷰는 하나

관리

contentReviews notifications
creativeEvents siteContent

- ◆ 프로필·프로젝트·서비스당 검토 대기는 최대 하나
- ◆ 알림은 수신자·이벤트마다 하나
- ◆ 사이트 문구는 키가 있는 문서 하나라 수정 시 재배포 불필요

모든 컬렉션의 전체 필드는 이전 페이지에 있습니다.

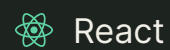
— 이전 프로젝트

Sofiya Bookstore

독자가 장르와 형식별로 책을 둘러보고, 장바구니에 담고, 주문을 확인하는 온라인 서점입니다. 카탈로그는 별도의 관리자 패널에서 관리합니다.

— 구현 방식

- ◆ React·Redux Toolkit 스토어프론트와 Express REST API 연동
- ◆ 두 가지 로그인 방식: 독자는 JWT 쿠키, 관리자 패널은 MongoDB에 저장된 세션
- ◆ 관리자 패널은 EJS로 서버 렌더링, 비밀번호는 bcrypt로 해싱



React



Redux Toolkit



MUI

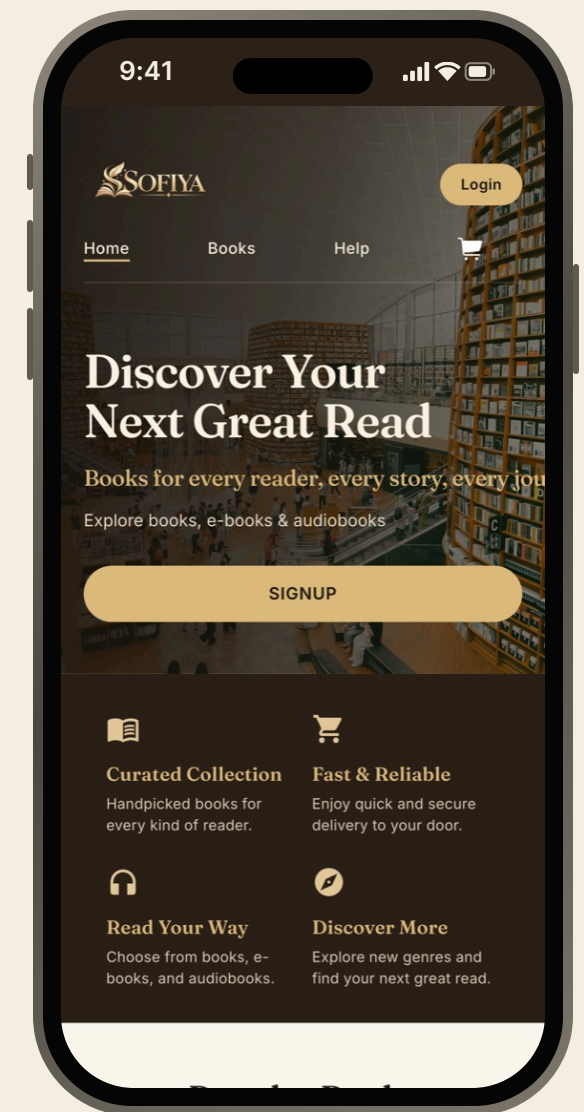
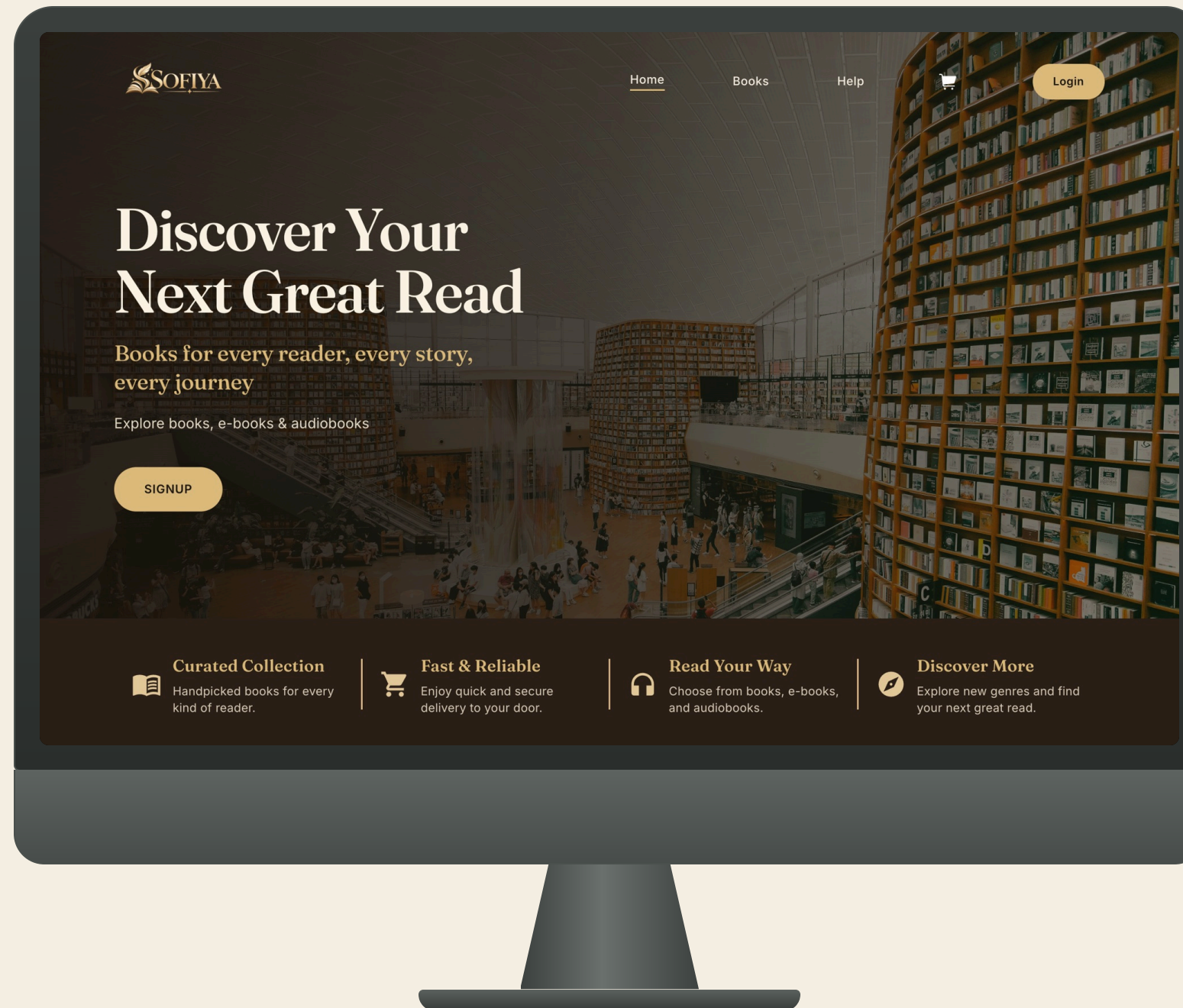


Express



MongoDB

저장소: [Sofiya-react](#) · [Sofiya-Bookstore](#)



탐색, 검색, 주문

9개 장르와 4가지 형식, 검색과 페이지 나누기, 슬라이더와 장바구니가 있는 상품 페이지를 갖추었고, 같은 화면이 휴대폰에도 맞춰집니다.

구현 방식

- ◆ 주문은 주문 항목과 함께 저장되며 보류, 처리 중, 배송 완료 상태를 거침
- ◆ 상품 조회 수는 회원당 한 번만 집계
- ◆ 커버·프로필 이미지 업로드는 Multer로 처리

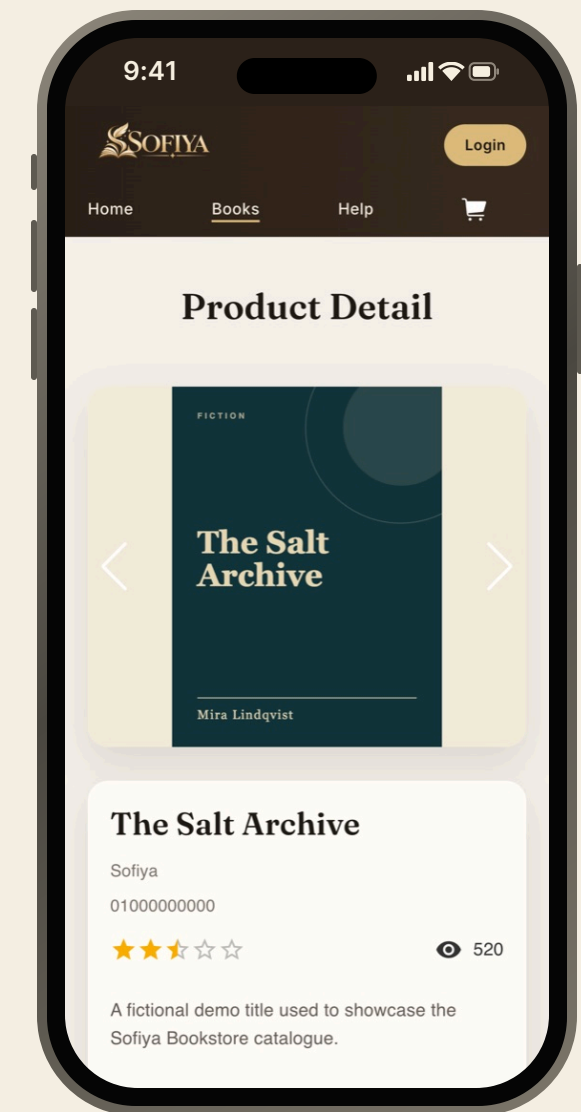
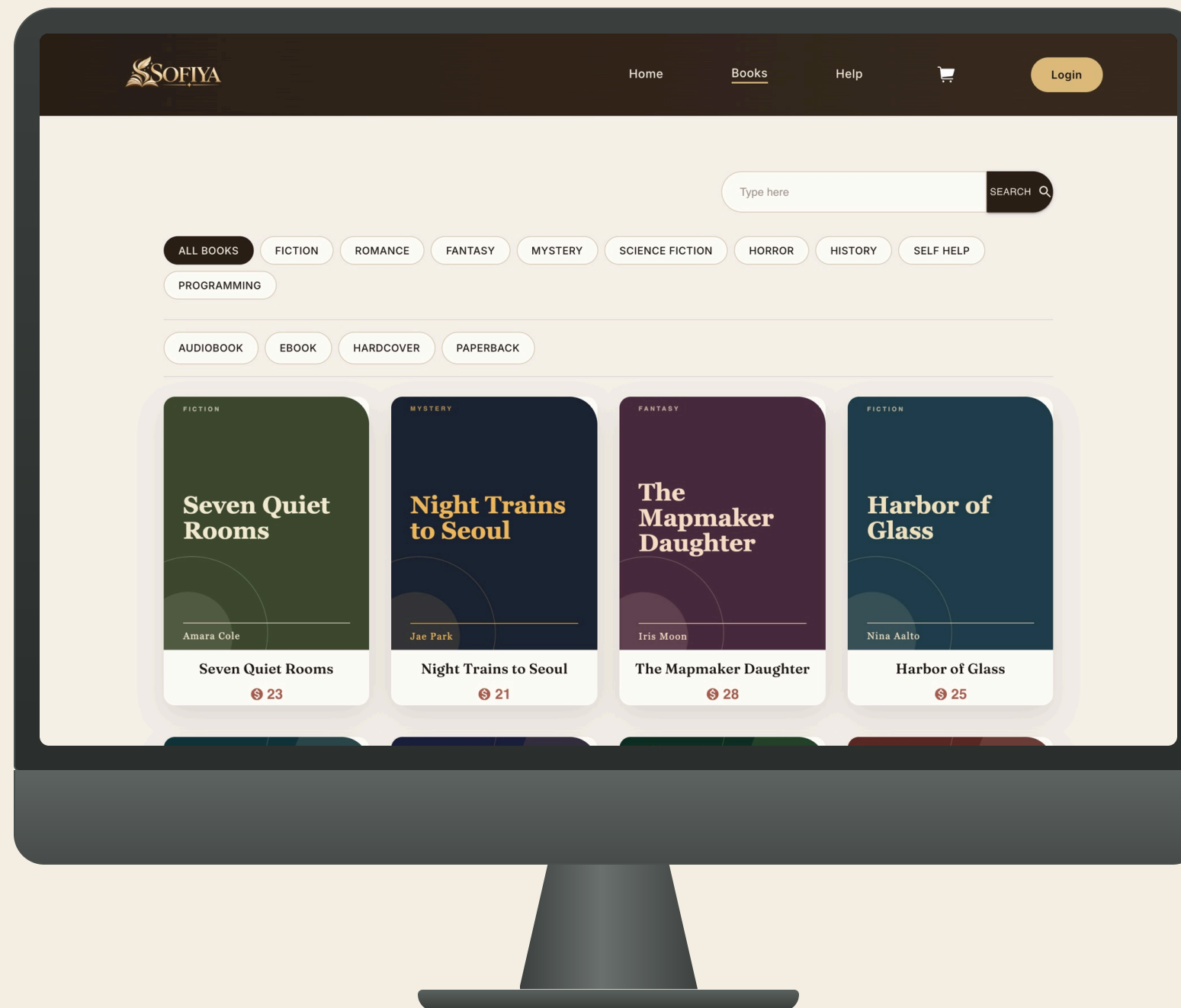
🌿 Mongoose

🔼 Axios

Multer

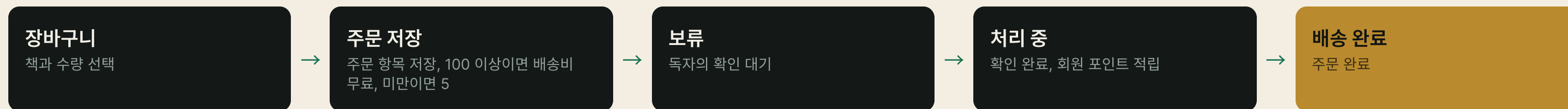
🔗 React Router

화면 속 도서 제목, 저자, 표지는 로컬 실행용 가상 데모 데이터입니다.



관리자 패널, 주문, 로그인

모든 주문 조회는 로그인한 회원 기준으로 필터링되므로, 독자는 자신의 주문만 보고 변경할 수 있습니다.



관리자 패널

서버에서 렌더링하는 EJS 페이지: 로그인, 상품 목록, 이미지 최대 5장의 신규 상품 등록, 상품 수정, 회원 상태를 바꿀 수 있는 회원 목록.



두 가지 로그인 경로

독자는 24시간 유효한 JWT를 쿠키로 받고, 모든 API 호출에서 검증됩니다. 관리자 패널은 MongoDB에 저장되는 6시간 세션을 쓰며, 스토어 계정만 가드를 통과합니다.



지금이라면 바꿀 점

상품 가격이 브라우저에서 넘어오고 결제 단계는 목업입니다. 지금이라면 서버에서 합계를 다시 계산하고 실제 결제 서비스를 연동하겠습니다.

아이디어부터 운영까지

✉ 이메일

kakhkhorov97@gmail.com

☎ 전화

[010-7901-7555](tel:010-7901-7555)

💬 카카오톡

kakhkhorov_A

🐙 GITHUB

github.com/kakhkhorov97

🌐 라이브 프로젝트

creatorhub.site

📍 주소

대한민국 경기도 수원시



라이브 프로젝트
creatorhub.site



GitHub
github.com/kakhkhorov97