Genie Webtoon 개발 보고서

Full-Stack Webtoon Platform

이영진

2025-10-22

목차

1. 서론

1.1 프로젝트 개요

'Genie Webtoon'은 누구나 무료로 즐길 수 있는 웹 기반 웹툰 플랫폼입니다. Next.js + React + Typescript를 사용하여 사용자 경험과 프론트엔드의 성능을 확보하고, 백엔드는 Node.js + express.js + Typescript 기반의 라우터, 컨트롤러, 미들웨어, DB모델 로직을 Next.js API Route + Sequelize ORM + MySQL(AWS RDS)기반의 코드로 리팩토링을 진행하여 백엔드 로직 처리를 담당하도록 하였습니다. 이미지, 쎔네일과같은 정적 데이터는 AWS S3에 보관하여 대용량 이미지 저장 및 사용이 용이하도록 하였습니다.

1.2 추진 배경 및 필요성

과금을 통한 미리보기, 광고 기반의 기존 웹툰 수익구조를 광고 기반의 수익구조로 단순화 시키는 동시에 모든 작품에 무료로 접근할 수 있도록하여 기존 대형 플랫폼(네이버 웹툰, 카카오 웹툰)과 차별점을 만들었습니다. 또한 정형화되어있던 웹사이트 구조에서 벗어나 호버기반 메인사이트 제작, AI챗봇의 웹툰 추천, 웹툰 골든벨을 통한 새로운 사용자 경험이 가능한 웹툰 플랫폼을 제작하였습니다.

1.3 목표

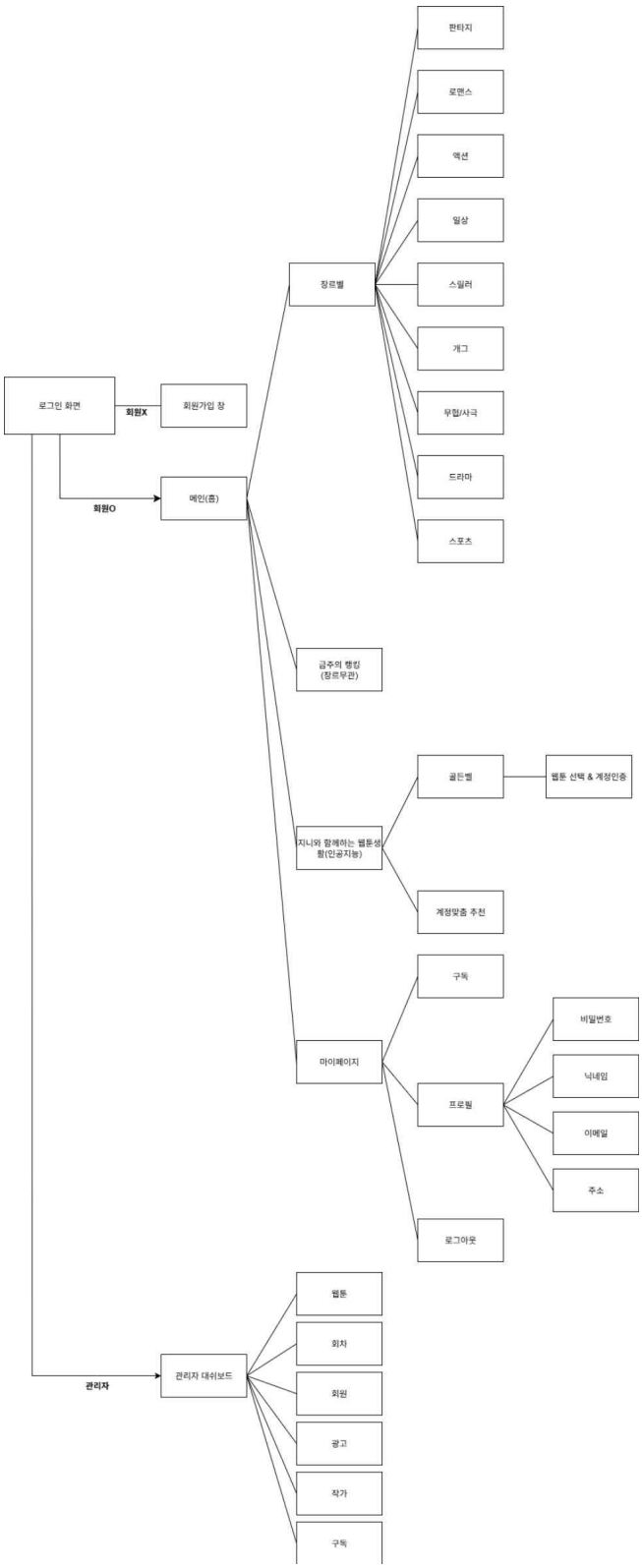
사용자에게 신선한 충격과 경험을 줄 수 있는 기존 틀을 깨는 UI/UX와 더불어 사용자 편의를 생각하는 UI/UX 또한 빼놓지 않고 구현하는 것이 목표입니다. 또한 관리자 대쉬보드 페이지를 제작하여 광고, 작품, 회차, 작가, 멤버, 댓글, 구독등의 여부를 확인할 수 있도록 할 것입니다. 이외에는 웹툰 뷰어(에피소드 뷰어), 구독, 알람, 댓글, 랭킹, 마이페이지, 지니AI(챗봇), 골든벨과 같은 기존 기능과 더불어 지니웹툰만의 특색을 지닌 여러 메뉴와 기능들을 구현할 계획입니다.

2. 시스템 개요

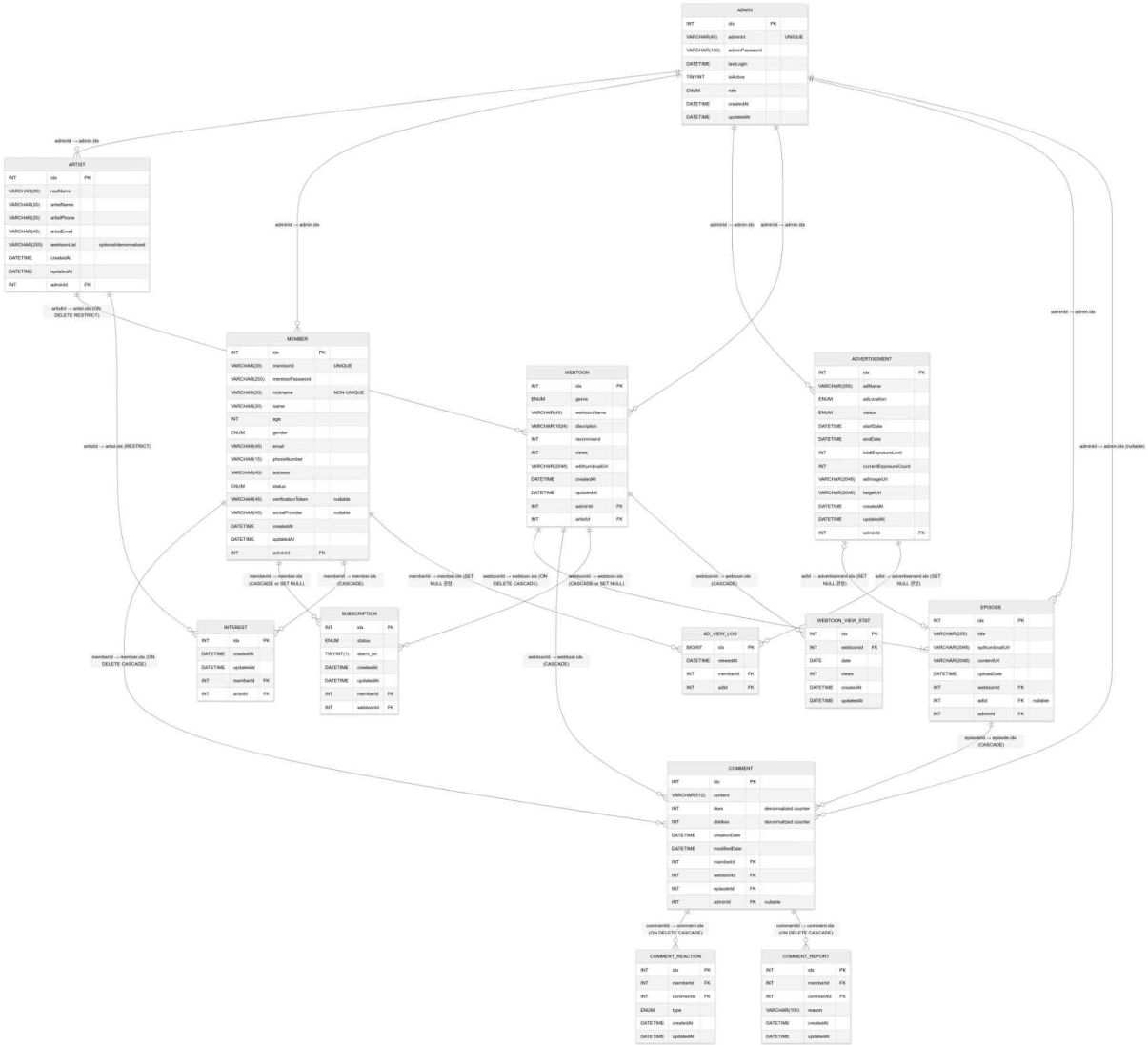
2.1 아키텍처 개요

Next.js(SSR/ISR/Route Handler) + Sequelize ORM + MySQL(AWS RDS) + AWS S3 + NextAuth 기반 풀스택 프로젝트를 구현했습니다. 이때 API는 app/api로 따로 라우터들을 관리하도록 분리하고, middleware.ts로 보호합니다. 또한 컨트롤러/서비스/모델 계층을 구성하여 프런트엔드와 백엔드 코드를 next.js 기반 하나의 폴더 아래에서 모두 관리하도록 구성하였습니다. 썸네일과 이미지등은 AWS S3를 활용하여 저장후 MySQL(AWS RDS)에 해당 이미지의 S3 URL을 테이블에 저장하는 식으로 이미지를 호출합니다. 또한 Figma Pro 버전을 사용하여 화면 설계시 사용된 디자인을 React 태그로 변환하여 코드로서 바로 css 디자인에 적용시켰습니다.

2.2 정보구조(IA)


로그인 화면
회원X
회원가입 창
회원O
메인(홈)
장르별
판타지
로맨스
액션
일상
스릴러
개그
무협/사극
드라마
스포츠
금주의 랭킹
(장르무관)
지니와 함께하는 웹툰생활(인공지능)
골든벨
웹툰 선택 & 계정인증
계정맞춤 추천
마이페이지
구독
프로필
비밀번호
닉네임
이메일
주소
로그아웃
관리자
관리자 대쉬보드
웹툰
회차
회원
광고
작가
구독

2.3 데이터베이스(ERD)



3. 개발환경 및 개발 스택

3.1 개발도구/품질

ESlint 9.33.0버전, Prettier 3.6.2버전, TypeScript strict을 사용해 코드 정합성과 문법 맞춤을 확인하였습니다. Vercel을 통해 배포 및 logging 기능을 통해 로그/모니터링을 실시하였습니다.

3.2 클라우드/스토리지

AWS RDS(MySQL), AWS S3(썸네일/이미지), 사전서명 업로드(Presigned URL)를 클라우드와 스토리지 관련 툴로 사용하여 프로젝트를 구축했습니다. 인증메일을 위한 이메일 관리는 Nodemailer를 사용했습니다.

3.3 인증/보안

NextAuth, Sequelize(Credentials/Email), bcrypt 해시, bfcache를 사용하여 사용자의 로그인,회원가입등의 캐시,세션을 관리하도록 코드를 작성하였습니다. 또한 middleware.ts를 사용하여 admin, member의 role과 status를 보호하도록 하였습니다. 추가적으로 NextAuth(SequelizeAdapter)로 세션을 관리하고, API는 requireAuth/requireAdminAuth가드로 보호한다. 권한 미달은 401/403 JSON 또는 /403 리디렉트로 처리합니다.

3.4 프런트엔드

Next.js 15(App Router), React 19, TypeScript 5버전을 사용하여 사용자가 보는 페이지들을 전부 제작하였습니다. 스타일은 Tailwind CSS, styled-components를 사용하였고 일부 페이지에서는 scss를 사용하여 특정 디자인 적용을 하도록 하였습니다. 상태는 SWR, Zustand, React Hook Form을 사용하여 유지 및 관리하도록 설정하였습니다. 또한 기본적 화면 설계는 figma를 사용하여 디자인하였습니다.

3.5 백엔드

Next.js Route Handlers(API), 컨트롤러, 서비스, 모델, 미들웨어 계층 분리를 통해 유지보수가 용이하도록 제작하였습니다. Sequelize(TypeScript), mysql2 드라이버를 설치후 이를 백엔드 코드 전반에 import하여 DB에 접근 및 수정이 용이하게 제작하였습니다.

4. 요구사항

4.1 비기능 요구사항

4.1.1. 보안 및 접근 제어

공통 미들웨어 /admin과 주요 관리자 경로에 접근한 요청을 검사해 비로그인 사용자는 로그인 페이지로, 권한 없는 사용자는 403 페이지로 리디렉션합니다.

서버 사이드 API는 requireAuth/requireAdminAuth를 통해 토큰에서 최소 정보만 추출한 라이트 세션을 사용하고, 권한이 없을 경우 JSON 401/403을 반환합니다.

NextAuth 설정은 Sequelize 어댑터와 bcrypt 검증을 사용해 관리자/회원 로그인 흐름을 분리하고, 소셜 계정 교차 연동을 차단하며, 리다이렉트·JWT 후처리에서 세션 변조를 막습니다.

관리자/보호 레이아웃은 서버 세션을 직접 확인해 비관리자 접속을 차단하고, 관리자 API 라우터도 동일한 가드를 사용합니다.

회원 관리 기능은 yup/zod 스키마로 입력을 검증하고, 컨트롤러에서 bcrypt 해시·UUID 토큰 생성·중복 검사 등을 수행합니다.

이메일 인증은 DB 트랜잭션으로 계정 상태를 ACTIVE로 전환하고 최소 관리자에게 자동 할당하며, 실패 시 롤백합니다.

비밀번호 찾기 로직은 메일 환경 미구성 시 503을 반환하고, 임시 비밀번호 발급·메일 발송이 성공했을 때만 트랜잭션을 커밋합니다.

4.1.2. 안정성 및 운영

이메일 전송은 Gmail/Naver/Legacy SMTP를 우선순위와 폴백으로 구성하며, 환경 미설정 시 실패 대신 경고만 남깁니다.

/api/health/db엔드포인트가 인증 없이도 DB 연결을 검사해 배포 후 헬스체크에 활용됩니다.

API 핸들러는 공통 오류 래퍼를 통해 미처리 예외를 500 JSON으로 변환합니다.

웹툰 조회 API는 정상 응답 이후 비동기로 조회수·일자별 통계를 업데이트하고, 통계 실패가 사용자 응답에 영향을 주지 않도록 설계되었습니다.

전용 메트릭 엔드포인트는 force-dynamic 및 revalidateTag를 사용해 캐시를 무효화하고 랭킹 데이터를 최신 상태로 유지합니다.

웹툰 조회 컨트롤러는 세션 검증 후 구독 상태를 함께 반환해 프런트가 예외 없이 처리할 수 있게 합니다.

4.1.3. 품질과 유지보수성

TypeScript는 strict, noImplicitAny, incremental등을 활성화해 타입 안정성을 보장합니다.

Next.js 빌드는 ESLint/TS 오류를 허용하지 않으며, 서버 번들 외부 패키지 목록·원격 이미지 도메인 등을 명시해 구성 실수를 방지합니다.

ESLint Flat 구성이 Next.js 추천 규칙과 @typescript-eslint 플러그인을 적용해 일관된 코드 스타일을 유지합니다.

VS Code 설정은 저장 시 Prettier/ESLint를 실행하도록 고정되어 협업 시 포맷 불일치를 줄입니다.

npm 스크립트와 의존성은 lint/build/start 외에도 Turbopack dev 서버 및 S3, Nodemailer 등 운영 필수 패키지를 명시합니다.

보조 스크립트는 모든 API 라우트에 runtime = 'nodejs'를 강제해 Edge 환경에서의 예상치 못한 동작을 방지합니다.

4.1.4. 성능 및 확장성

Sequelize 초기화는 커넥션 풀과 SSL 옵션을 설정하고, Next.js 핫 리로드 시 싱글턴을 재사용해 자원 소모를 줄입니다.

CLI용 설정은 환경별.env를 자동 로드하고 로깅 여부를 분기해 배포·테스트 환경을 구분합니다.

API fetch 유틸은 서버/클라이언트에서 적절한 기본 URL과 쿠키 정책을 적용하며, 실패 시 상세 오류 정보를 포함한 예외를 던집니다.

관리자 HTTP 래퍼는 cache: 'no-store' 와 쿠키 전파로 대시보드 데이터의 신선도를 확보합니다.

대시보드 통계는 Promise.all로 병렬 수집 후 상위 5개만 반환해 네트워크 병목을 완화합니다.

클라이언트 훅은 IntersectionObserver기반 무한스크롤, NextAuth

hydration 상태 캐시로 불필요한 리렌더와 로딩 지연을 최소화합니다.

4.1.5. 사용자 경험 및 접근성

루트 레이아웃은 스킵 링크·한국어 폰트·메타데이터를 설정해 접근성과 SEO를 함께 고려합니다.

헤더는 동적 임포트와 로딩 스켈레톤으로 초기 화면을 가볍게 유지하며, 네비게이션은 ARIA 속성으로 키보드 접근성을 높입니다.

전역 스타일은 다크모드 환경에서도 관리자/뷰어 영역을 라이트 테마로 강제하고 iOS 확대 등 브라우저별 이슈를 제어합니다.

Zustand 기반 UI 스토어는 모바일 내비와 테마 설정을 로컬 스토리지에 영속화해 사용자 선호도를 유지합니다.

앱 프로바이더는 NextAuth 세션과 커스텀 토스트를 전역으로 주입해 로그인 상태나 알림을 일관되게 처리합니다.

4.1.6. 배포 및 환경 구성

Next.js 설정은 S3/CloudFront 이미지 도메인과 서버 번들 외부 패키지를 명시해 AWS RDS·S3 등 외부 리소스 사용을 염두에 둔 구성입니다.

sequelize.config.js는 NODE_ENV에 따라.env.local/.env.production을 자동 로드해 배포 타깃별 DB 정보를 분리합니다.

AWS SDK, Nodemailer, Sequelize 등 배포 환경에서 필요한 런타임 의존성이 패키지 목록에 포함되어 있습니다.

Node.js 런타임 강제 스크립트는 App Router API가 Edge가 아닌 Node 환경에서 실행되도록 보장합니다.

4.2 기능 요구사항(화면 설계서 참고)

구분	항목	비고
로그인(GW_001)	로그인 버튼 클릭시	사용자/관리자 계정에 따른 에러/승인 문구 발생
	회원가입 클릭시	회원가입 페이지로 이동
	아이디 찾기 클릭시	아이디 찾기 페이지로 이동
	비밀번호 찾기 클릭시	비밀번호찾기 페이지로 이동
회원가입(GW_002)	ID, DOUBLE CHECK	아이디 입력칸 및 중복확인
	PW, PW CONFIRM	비밀번호 및 재입력칸, 비밀번호 규칙 표시
	NAME	이름 입력칸
	EMAIL	이메일 입력칸, 이메일 형식 강제
	BIRTH	생년월일 입력칸, 규칙대로 입력되지 않아도 자동완성
메인화면(GW_003)	장르별	클릭시 해당 세부 메뉴 호버 영역 호출
	랭킹	위와 동일
	마이페이지	위와 동일
	지니와 함께하는 웹툰생활	위와 동일
세부메뉴(GW_004)	장르별 세부메뉴	세부메뉴에 있는 메뉴 클릭시 해당 페이지로 이동
	랭킹 세부메뉴	세부메뉴에 있는 메뉴 클릭시 해당 페이지로 이동
	마이페이지 세부메뉴	세부메뉴에 있는 메뉴 클릭시 해당 페이지로 이동
	지니와 함께하는 웹툰 생활 세부메뉴	세부메뉴에 있는 메뉴 클릭시 해당 페이지로 이동
	웹툰 상세페이지	웹툰 제목, 작가이름, 웹툰 설명, 구독여부, 에피소드 목록등 해당

웹툰(GW_005~006)		웹툰의 상세 정보를 한페이지에 볼 수 있도록 제작된 페이지
	웹툰 뷰어	에피소드 본문과 조작 리모콘, 광고, 댓글영역등이 포함된 무한 스크롤 페이지
광고(GW_007)	광고 박스 영역	현재는 에피소드 아래, 에피소드 목록 위 중간 영역에만 광고를 표시하지만 이후에 요구사항이 추가되면 다른 페이지들에도 추가예정
관리자(GW_008)	관리자 대쉬보드	각 영역별 nav 버튼 모음
	회원 대쉬보드	DB에 등록된 메뉴와 관련된 수치, 그래프 대쉬보드
	광고 대쉬보드	
	멤버 대쉬보드	
	댓글 대쉬보드	
	회차 대쉬보드	
	작가 대쉬보드	
	웹툰 대쉬보드	
랭킹(GW_009)	일/주/월/연간 랭킹	각 기간별 누적 조회수 기반 랭킹 페이지 표시, 1~3등은 포디움 위에 4~10등은 아래 나열형 리스트로
마이페이지	구독	웹툰 상세페이지에 있던 구독 버튼을 클릭해놓은 작품들을 모아서 한눈에 확인할 수 있도록 제작한 페이지
	프로필	회원가입시 사용자가 입력한 정보를 확인하고 비밀번호를 수정할 수 있도록 한 페이지, 회원탈퇴 버튼또한 포함중
	로그아웃	사용자의 쿠키를 지우는 동시에 로그아웃 로직을 실행

지니와 함께하는 웹툰생활	추천웹툰	지니AI가 DB에 등록된 웹툰과 사용자의 구독 내역을 바탕으로 사용자가 원하거나 보기 좋은 웹툰 추천
	도전! 골든벨	본인이 구독한 웹툰에 대해서 골든벨 형식으로 문제를 풀어볼 수 있는 페이지

5. 설계

5.1 페이지 개요

페이지군	경로(예시)	목적	보호여부
홈/장르	app/(protected)/home/page.tsx, app/(protected)/genre/[genre]/page.tsx	기본 피드, 장르 그리드/필터, 랭킹 프리뷰	멤버 전용
랭킹	app/(protected)/ranking/[period]/[genre]/page.tsx	일/주/월/년 + 장르별 Top3 포디움, 4~10위 리스트	멤버 전용
웹툰 상세	app/(protected)/webtoon/[id]/page.tsx	작품 메타/썸네일/구독, 에피소드 목록, 조회 추적	멤버 전용
에피소드 뷰어	app/(protected)/webtoon/[id]/episodes/[epId]/page.tsx	본문 보기, 이전/다음, 광고, 댓글 CRUD	멤버 전용
지니AI 추천	app/(protected)/genieai/recommendation/page.tsx	장르/취향 기반 추천, 썸네일·바로가기	멤버 전용
골든벨	app/(protected)/genieai/golden-bell/page.tsx	구독 작품별 4문항 퀴즈, 정오답 피드백	멤버 전용
멤버	app/(protected)/(member)/{profile,bookmarks,interests}/page.tsx	마이페이지 CRUD, 구독·관심·북마크 관리	멤버 전용
어드민	app/(admin)/*	운영(광고/댓글/회원/작품/회차/작가)·대시보드	관리자 전용

5.2 파일 구조(경로)

5.2.1 App Router

구분	주요 경로(예시)	비고
(protected)	ranking/[period]/[genre]/(page)	Client
	webtoon/[id]/(page)	loading
	webtoon/[id]/episodes/[epId]/(page)	CommentSectionClient.tsx
	(member)/{profile,bookmarks,interests}/(page)	loading
	genieai/{recommendation,golden-bell}/page.tsx	추천/퀴즈
(admin)	(admin)/layout.tsx, .../dashboard/page.tsx	전역 레이아웃
	(admin)/{advertisements,comments,reported,webtoons,episodes,artists,members}/**/page.tsx	목록/상세/뷰로그
공통 스타일	styles/{global.css,tailwind.css,expanding-grid.css}, (protected)/styles/public.scss	글로벌/페이지 전용

5.2.2 API Routes

범주	경로(예시)	용도
랭킹	app/api/(protected)/ranking/{daily,weekly,monthly,yearly}/[genre]/route.ts	집계 API
웹툰/에피	`app/api/(protected)/webtoon/([id]/route).ts, .../episode/[webtoonId]/[episodeId]/route.ts, .../episode/navigation/route.ts`	웹툰 상세페이지, 에피소드 뷰어
댓글	app/api/(protected)/comments/route.ts	댓글 CRUD/리액션/신고
멤버	app/api/(protected)/member/*	프로필/북마크/관심/구독
광고	app/api/(protected)/advertisement/view/route.ts	뷰 로그
어드민	app/api/(protected)/admin/*	운영 전반
인증	app/api/auth/[...nextauth]/route.ts, signup/route.ts,verify-email/route.ts, find-id/route.ts, find-password/route.ts	인증·계정
메트릭	app/api/(protected)/metrics/webtoon-view/route.ts	조회수 기록
헬스	app/api/health/db/route.ts	DB 헬스체크

5.2.3 Controllers / Services / Models / 공용

계층	경로(예시)	비고
Controllers	controllers/{webtoon,episode,comment,member,ranking,advertisement,genieai}/**.ts	도메인별
Admin Controllers	controllers/admin/*	대시보드/운영
Services	services/{webtoon,episode,comment,member,ranking,advertisement,artist,auth}.service.ts, admin/dashboard.service.ts	비즈로직
Models	models/{member,admin,artist,webtoon,episode,comment,commentReaction,commentReport,subscription,interest,advertisement,ad_view_log,webtoonViewStat}.ts	Sequelize
Lib	lib/{apiClient,fetcher,auth-client,format,route,toNumericId,validators,middlewares}.*	공용 유틸
Hooks	hooks/{useAdViewLog,useInfiniteScroll,useAuth}.ts	커스텀 훅
Types	types/{next-auth.d.ts,dto.ts}	세션/DTO
Store	store/ui.store.ts	전역 UI

5.3 주요 컴포넌트(프런트)

페이지군	컴포넌트	역할	비고
랭킹	components/public/RankingList.tsx	4~10위 리스트	가상 스크롤 검토
	components/public/Podium.tsx또는 components/ranking/RankingPodium.tsx	Top3 포디움	강조 UI
웹툰 상세	components/cards/WebtoonCard.tsx	메타/썸네일 카드	재사용 빈도 높음
	app/(protected)/webtoon/[id]/ViewTracker.tsx	조회 추적	메트릭 전송
에피소드	components/viewer/EpisodeViewer.tsx(+ CSS 모듈)	본문 이미지 렌더	지연 로딩
	components/viewer/EpisodeNavigator.tsx, components/public/ViewerNav.tsx	이전/목록/다음	경계 비활성
	components/viewer/CommentSection.tsx	댓글 CRUD/리액션/신고	분리 로딩
광고	components/ads/AdBanner.tsx	노출/클릭 로그	useAdViewLog
지니AI	(페이지 내부) 추천 카드	썸네일+바로가기	말풍선 UI
골든벨	(페이지 내부) 문제 카드/배지	정오답 피드백	마지막 상태 초기화
멤버	ProfileFormClient.tsx, BookmarksListClient.tsx	폼·목록	React Hook Form 사용
공용	ui/{SpeechBubble,BackNavigator,ImageFallBack}.tsx, nav/MegaMenu.tsx	스타일·내비	공통 요소

5.4 상태/훅

범주	리소스	역할/용도	비고
세션/권한	lib/auth-client.ts,types/next-auth.d.ts	세션/토큰확장(role, adminRole, status)	보호 라우팅
데이터 패칭	lib/fetcher.ts(api 클라이언트)	공통 GET/POST/PUT/DELETE	에러 공통 처리
랭킹	(클라 유틸) Client.tsx, _fetch.ts	period/genre 기반 로딩·캐시	SWR/캐시
광고 뷰	hooks/useAdViewLog.ts	노출/클릭 전송	sendBeacon 고려
뷰어/댓글	hooks/useInfiniteScroll.ts	무한 스크롤	관찰자 패턴
인증 가드	hooks/useAuth.ts	보호 페이지 접근 보조	리다이렉트
폼 검증	lib/validators.ts, validators/*	Zod/Yup 스키마	RHF 연동

5.5 데이터 플로우(아키텍처)

기능	호출 흐름	핵심 포인트	DTO/검증
랭킹	Page/Client→ /api/(protected)/ranking/*→ Controller → Service → Model	기간경계(Asia/Seoul), unstable_cache	period/genre normalizer
웹툰 상세	Page(SSR)→ /api/.../webtoon/[id]→ Controller → Service → Model	상세 SSR + 에피 목록 CSR	id 유효성
에피소드	Page→ /api/.../episode/[wId]/[epId]+ /navigation	이전/다음 경계 가드	숫자 파라미터
댓글	Client → /api/.../comments→ Controller → Service → Model	CRUD/리액션/신고	길이/금칙어
추천	Client→ /api/(protected)/genieai/recommendation/*	썸네일·경로 제공	옵션 스키마
골든벨	Start/Load/Answer → Controller → Service	구독 확인·채점	정답 인덱스
멤버	Client→ /api/(protected)/member/*	프로필/북마크/관심/구독	인증 필수
어드민	Client→ /api/(protected)/admin/*	권한 (MANAGER /SUPER)	

5.6 권한/보안

리소스	접근 대상	미들웨어/가드	검증	데이터 최소화
(protected)페이지/API	로그인 멤버	middleware.ts+ getToken()	세션유효, status!=='DELETED'	DTO 화이트리스트
(admin)페이지/API	ADMIN(`SUPER, 'MANAGER)		middleware.ts+ role/adminRole	권한 미달 → /403
랭킹 파라미터	멤버	Normalizer	period∈{daily,weekly,...}, genre∈enum	-
댓글/폼 입력	멤버	Zod/Yup	길이/포맷/금칙어	서버에서 escape
파일 업로드	어드민	토큰+사이즈/확장자	이미지 MIME 검사	URL 제한 노출
CSRF/XSS	모든 POST	SameSite/Lax, 서버 렌더 우선	HTML 미허용	텍스트만

5.7 성능/UX 설계

영역	전략	구현 포인트
렌더링	SSR(상세/관리자), CSR(댓글/추천) 혼합	스트리밍/분리 로딩
캐싱	랭킹 unstable_cache	장르·기간 키
리스트	무한스크롤/페이지네이션	useInfiniteScroll/테이블 페이징
이미지	지연 로딩/폴백	ImageFallBack/소형 썸네일
로딩/에러	loading.tsx/error.tsx·Skeleton	EmptyState 분리
네비/버튼	경계 비활성, 추천 버튼 하단 고정	접근성 고려
접근성	포커스·키보드 내비	모바일·데스크톱 최적화

5.8 예외/에러 처리

시나리오	감지/가드	사용자 피드백
잘못된 period/genre/id/epId	라우트·스키마 검증	400/404 + 안내
비로그인 접근	미들웨어·useAuth	로그인 유도
권한 미달(관리자)	role/adminRole 체크	/403리다이렉트
데이터 없음	서비스 0건	Empty 컴포넌트
FK 제약(댓글↔리액션)	DB 에러 매핑	친절한 메시지/대안
네트워크/서버 에러	fetcher 공통	토스트/재시도
업로드 실패	서비스 예외	롤백/재시도/가이드

5.9 로그 추적

이벤트	수집 경로	저장/집계	활용
작품 조회	ViewTracker.tsx→ /api/(protected)/metrics/webtoon-view	뷰 카운트/시간대	랭킹/추천 근거
광고 노출/클릭	useAdViewLog+ AdBanner.tsx→ /api/(protected)/advertisement/view	ad_view_log 집계	대시보드 차트
랭킹 클릭	Podium/List onClick(프론트)	(선택) 이벤트 수집	UX·AB

관리자 활동	컨트롤러/서비스 로그	표준 로그 스키마	운영 감사

5.10 테스트 포인트

범주	핵심 테스트	경계/에러 케이스
권한/인증	(protected)/(admin) 접근 가드	세션 만료·권한 불일치
랭킹	period×genre 응답·정렬 안정성	동률/캐시 경계(자정/주초/월초/연초)
상세/구독	구독 토글 상태 반영·롤백	썸네일 폴백·메트릭 전송
뷰어/댓글	이전/다음 경계·댓글 CRUD/신고	긴 본문 성능·리액션/제약
추천	버튼 상호작용·썸네일/바로가기	잘못된 입력 처리
골든벨	구독 확인·정오답 피드백	마지막 문항 후 초기화
멤버	변경 없음 → 저장 비활성	빈 목록/페이징
어드민	상태 토글/업로드/필터	403/유효성 실패
API 공통	400/404/422/500 응답	타임아웃/재시도

6. 구현

6.1 로그인 페이지

레이어	파일/경로	설명
App Router (Page)	app/login/page.tsx,app/providers.tsx, app/post-login/page.tsx	로그인 화면, 로그인 후 처리 페이지(소셜·세션 반영)
API Routes	app/api/auth/[...nextauth]/route.ts	NextAuth 엔드포인트(크리덴셜/소셜)
Controllers	controllers/auth/nextAuthController.ts	NextAuth 옵션/콜백 래핑
Services	services/auth.service.ts	로그인/세션 연계 헬퍼
Models	models/member.ts,models/admin.ts, models/index.ts	Sequelize 모델 (회원/관리자)
Components	components/forms/PasswordField.tsx, components/forms/TextField.tsx, components/auth/SocialLoginButtons.tsx, components/auth/SessionActions.tsx	입력 UI, 소셜 로그인 버튼
DB	db/migrations/20250602012605-create-member.js, db/seeders/*member*	회원 테이블/시드
Lib/MW/유틸	lib/middlewares/authOptions.ts, lib/auth-client.ts, lib/validators/auth.ts	NextAuth 옵션, 클라이언트 헬퍼, 검증 스키마

6.2 회원가입 페이지

레이어	파일/경로	설명
App Router (Page)	app/signup/page.tsx	회원가입 화면
API Routes	app/api/auth/signup/route.ts, app/api/auth/checkduplicate/route.ts, app/api/auth/verify-email/route.ts	가입, 아이디 중복체크, 이메일 인증
Controllers	controllers/auth/signupController.ts, controllers/auth/checkDuplicateController.ts, controllers/auth/verifyEmailController.ts	가입/중복/이메일 인증 처리
Services	services/auth.service.ts, lib/emailService.ts	가입 로직, 인증 메일 발송
Models	models/member.ts	회원
Components	components/forms/TextField.tsx, components/forms/PasswordField.tsx	폼 입력
DB	db/migrations/20250802-add-verification-	verificationToken추가

	token-to-member.js	
Lib/MW/유틸	lib/validators/auth.ts, lib/middlewares/validate.ts	가입 입력 검증, 라우트 검증 미들웨어

6.3 아이디/비밀번호 찾기 페이지

레이어	파일/경로	설명
App Router (Page)	app/find-id/page.tsx,app/find-password/page.tsx	찾기 화면
API Routes	app/api/auth/find-id/route.ts, app/api/auth/find-password/route.ts	아이디/비번 찾기
Controllers	controllers/auth/findIdController.ts, controllers/auth/findPasswordController.ts	찾기 로직
Services	services/auth.service.ts, lib/emailService.ts	조회/임시비번 발급·메일
Models	models/member.ts	회원
Components	components/forms/TextField.tsx	입력 폼
DB	(회원 테이블 동일)	
Lib/MW/유틸	lib/validators/auth.ts, lib/middlewares/errorHandler.ts	검증/에러 핸들링

6.4 메인화면 페이지

레이어	파일/경로	설명
App Router (Page)	app/page.tsx,app/(protected)/home/page.tsx, app/(protected)/layout.tsx	랜딩/보호 홈
API Routes	app/api/(protected)/webtoon/route.ts	메인 목록 로딩
Controllers	controllers/webtoon/webtoonController.ts, controllers/genre/listGenreController.ts	웹툰 리스트/장르
Services	services/webtoon.service.ts	목록/조회수 등
Models	models/webtoon.ts, models/webtoonViewStat.ts	웹툰/조회수 집계
Components	components/home/ExpandingGrid.tsx, components/cards/WebtoonCard.tsx, components/ui/SpeechBubble.tsx	확장 그리드/카드/말풍선
DB	db/migrations/20250602012636-create-webtoon.js,db/migrations/20250823-create-webtoon-view-stat.ts	웹툰/뷰 통계 테이블

Lib/MW/유틸	lib/fetcher.ts,hooks/useInfiniteScroll.ts, hooks/useAuth.ts	SWR fetcher, 무한 스크롤, 인증

6.5 웹툰 상세

레이어	파일/경로	설명
App Router (Page)	app/(protected)/webtoon/[id]/page.tsx, app/(protected)/webtoon/[id]/loading.tsx	상세·로딩
API Routes	app/api/(protected)/webtoon/[id]/route.ts, app/api/(protected)/member/subscription/route.ts, app/api/(protected)/member/bookmarks/route.ts	상세, 구독, 북마크
Controllers	controllers/webtoon/webtoonController.ts, controllers/member/interestsController.ts, controllers/member/bookmarksController.ts	상세·관심/북마크
Services	services/webtoon.service.ts, services/member.service.ts, services/episode.service.ts	상세·관심/구독·회차
Models	models/webtoon.ts,models/episode.ts, models/subscription.ts, models/interest.ts	웹툰/회차/구독/관심(작가)
Components	components/webtoon/SubscribeControls.tsx, components/webtoon/EpisodeList.tsx, components/webtoon/ReaderControls.tsx, components/cards/WebtoonCard.tsx	구독/회차 리스트/리더 제어
DB	db/migrations/20250619100007-create-subscriptions.js	구독 테이블
Lib/MW/유틸	lib/route.ts, lib/toNumericId.ts, lib/format.ts	라우팅/ID 변환/포맷

6.6 에피소드 뷰어

레이어	파일/경로	설명
App Router (Page)	app/(protected)/webtoon/[id]/episodes/[epId]/page.tsx, app/(protected)/webtoon/[id]/episodes/[epId]/CommentSectionClient.tsx, app/(protected)/webtoon/[id]/ViewTracker.tsx	뷰어/댓글 클라 모듈/뷰 트래커
API Routes	app/api/(protected)/episode/[webtoonId]/[episodeId]/comments/route.ts, app/api/(protected)/comment/*,	댓글 CRUD/리액션/리포트, 조회수 기록

	app/api/(protected)/metrics/webtoon-view/route.ts	
Controllers	controllers/episode/episodeController.ts, controllers/comment/*, controllers/advertisement/advertisementViewController.ts	회차/댓글(좋·싫·답글·신고)/광고뷰
Services	services/episode.service.ts, services/comment.service.ts	회차/댓글 로직
Models	models/episode.ts, models/comment.ts, models/commentReaction.ts, models/commentReport.ts, models/ad_view_log.ts	에피소드/댓글/반응/신고/광고 로그
Components	components/viewer/EpisodeViewer.tsx, components/viewer/EpisodeNavigator.tsx, components/viewer/ScrollButtons.tsx, components/viewer/CommentSection.tsx, components/ads/AdBanner.tsx, components/ui/BackNavigator.tsx	뷰어/네비/스크롤/댓글/광고/뒤로가기
DB	db/migrations/20250602012734-create-episode.js, db/migrations/20250602012738-create-comment.js, db/migrations/20250619100009-create-ad-view-logs.js	회차/댓글/광고뷰 로그
Lib/MW/유틸	hooks/useAdViewLog.ts, lib/fetcher.ts, lib/middlewares/errorHandler.ts	광고뷰 로그 훅, 페처, 에러 처리

6.7 랭킹 페이지

레이어	파일/경로	설명
App Router (Page)	app/(protected)/ranking/[period]/[genre]/page.tsx, app/(protected)/ranking/[period]/[genre]/Client.tsx, app/(protected)/ranking/[period]/[genre]/_fetch.ts	일/주/월/연 + 장르
API Routes	app/api/(protected)/ranking/daily/[genre]/route.ts, .../weekly/[genre]/route.ts, .../monthly/[genre]/route.ts, .../yearly/[genre]/route.ts, app/api/(protected)/ranking/_lib.ts	기간별 랭킹
Controllers	controllers/ranking/dailyRankingController.ts, weeklyRankingController.ts, monthlyRankingController.ts, yearlyRankingController.ts	컨트롤러 4종
Services	services/ranking.service.ts, services/webtoon.service.ts	랭킹 집계/정렬

Models	models/webtoon.ts, models/webtoonViewStat.ts	기준 데이터/집계
Components	components/ranking/RankingPodium.tsx, components/cards/WebtoonCard.tsx, components/feedback/EmptyState.tsx	포디움/카드/빈 상태
DB	db/migrations/20250823-create-webtoon-view-stat.ts	뷰 통계
Lib/MW/유틸	hooks/useRanking.ts, lib/format.ts	클라 훅, 포맷

6.8 마이페이지

레이어	파일/경로	메모
App Router (Page)	app/(protected)/(member)/profile/page.tsx, app/(protected)/(member)/bookmarks/page.tsx, app/(protected)/(member)/interests/page.tsx	프로필/북마크/관심(작가)
API Routes	app/api/(protected)/member/profile/route.ts, .../profile/update/route.ts, .../bookmarks/route.ts, .../interests/route.ts, .../subscription/route.ts, .../subscription/[webtoonId]/alarm/route.ts	마이페이지 관련 API
Controllers	controllers/member/profileController.ts, bookmarksController.ts, interestsController.ts	마이페이지 컨트롤러
Services	services/member.service.ts, services/webtoon.service.ts	회원/웹툰 연동
Models	models/member.ts,models/subscription.ts, models/interest.ts, models/webtoon.ts	회원/구독/관심/웹툰
Components	components/ui/ImageFallBack.tsx, components/cards/WebtoonCard.tsx	썸네일/카드
DB	db/migrations/20250619100007-create-subscriptions.js	구독 스키마
Lib/MW/유틸	hooks/useAuth.ts,hooks/useToast.tsx, lib/validators/auth.ts	인증/토스트/검증

6.9 추천 웹툰 페이지

레이어	파일/경로	메모
App Router (Page)	app/(protected)/genieai/recommendation/page.tsx	추천 UI
API Routes	app/api/(protected)/genieai/recommendation/rout	개인화/장르별 추천

	e.ts, .../for-me/route.ts, .../list-by-genre/route.ts	
Controllers	controllers/genieai/recommendationController.ts	추천 로직(LLM+룰)
Services	services/webtoon.service.ts	후보 추출/정렬
Models	models/webtoon.ts, models/member.ts, models/subscription.ts	후보/사용자 맥락
Components	components/cards/WebtoonCard.tsx, components/ui/SpeechBubble.tsx	추천 카드/말풍선
DB	(웹툰/구독/조회 통계 활용)	
Lib/MW/유틸	lib/apiClient.ts, lib/format.ts, hooks/useAuth.ts	API 클라이언트/포맷/인증

6.10 도전! 골든벨 페이지

레이어	파일/경로	메모
App Router (Page)	app/(protected)/genieai/golden-bell/page.tsx	퀴즈 UI
API Routes	app/api/(protected)/genieai/golden-bell/route.ts, .../start/route.ts, .../[webtoonId]/route.ts, .../answer/route.ts, .../ranking/route.ts	퀴즈 시작/문제/정답/랭킹
Controllers	controllers/genieai/goldenBellController.ts	골든벨 진행
Services	services/webtoon.service.ts, services/member.service.ts	문제 풀기/검증 보조
Models	models/webtoon.ts, models/member.ts	기본 참조
Components	(공용UI)components/ui/SpeechBubble.tsx, components/cards/WebtoonCard.tsx	안내/카드
DB/데이터	data/goldenBell/questions.ts	문제 데이터 소스
Lib/MW/유틸	lib/fetcher.ts,hooks/useToast.tsx, lib/middlewares/auth.ts	페처/토스트/보호

6.11 관리자 페이지

레이어	파일/경로	메모
App Router (Page)	(admin)/layout.tsx,(admin)/dashboard/page.tsx, (admin)/advertisements/page.tsx, (admin)/advertisements/[id]/page.tsx, (admin)/advertisements/[id]/view-logs/page.tsx, (admin)/comments/page.tsx,	관리자 UI 전반

	(admin)/members/page.tsx, (admin)/webtoons/page.tsx, (admin)/episodes/page.tsx, (admin)/artists/page.tsx, (admin)/subscriptions/page.tsx	
API Routes	app/api/(protected)/admin/*(ads, view-logs, artists, comments,reported,members, episodes(+thumbnail),webtoons(+thumbnail), subscriptions)	관리자용 보호 API
Controllers	controllers/admin/advertisementsController.ts, advertisementViewLogsController.ts, artistsController.ts,commentsController.ts, reportedCommentsController.ts, membersController.ts,episodesController.ts, webtoonsController.ts, subscriptionsController.ts	관리자 전 도메인 컨트롤
Services	services/admin/advertisements.service.ts, admin/advertisements-view-logs.service.ts, admin/artists.service.ts, admin/comments.service.ts, admin/reported-comments.service.ts, admin/members.service.ts, admin/episodes.service.ts, admin/webtoons.service.ts, admin/subscriptions.service.ts, admin/dashboard.service.ts	관리자 서비스 계층
Models	models/advertisement.ts,models/ad_view_log.ts, models/webtoon.ts,models/episode.ts, models/member.ts,models/artist.ts, models/subscription.ts, models/comment.ts	전 도메인 모델
Components	components/admin/SimpleBarChart.tsx, components/ads/AdBanner.tsx, components/cards/WebtoonCard.tsx, components/feedback/EmptyState.tsx	대시보드 차트/광고/카드
DB	db/migrations/*advertisement*,*ad-view-logs*, *webtoon*,*episode*,*member*,*artist*, *subscriptions*, *comment*	각 테이블/시드 일괄
Lib/MW/유틸	lib/middlewares/auth.ts(권한), lib/middlewares/db.ts(DB연결), lib/middlewares/errorHandler.ts,lib/validators.ts, lib/route.ts	보호·DB·에러·검증·라우팅

7. 테스트

테스트는 GenieWebtoon-Test-Cases.xlsx 파일에 test case로 동작 기대사항부터 동작여부까지 정리해서 모두 정리해놓았습니다.

8. 배포 및 운영

.env.loacl, .env.production의 분리를 통해(DB, S3, Nodemailer, Auth Secret) 보안 설정이 유지되도록 환경 분리를 해놓았습니다. Node.js LTS, Next.js build/start, Vercel를 기반으로 런타임 설정하여 안정적인 운영이 가능하도록 하였습니다. DB운영은 Sequelize를 활용한 마이그레이션과 시더를 활용하여 추가/유지/보수가 용이하도록 하였습니다. 보안으로는 IAM 최소권한 설정, 비밀번호 bcrypt 해시, Vercel 내부 firewall 설정,계정 역할 기준 접근 방식을 사용하여 강화하였습니다. 관측성을 위해서는 KPI 대시보드(활성 광고/신규 회원/댓글/회차)를 활용하여 관리자가 확인할 수 있도록 하고 Vercel 내부의 logs에서 확인가능하도록 하였습니다. 배포는 github에 전체 소스코드를 push후 이와 연결한 Vercel 프로젝트 관리 페이지에서 연결된 url에 빌드된 프로젝트를 적용하는 방식으로 배포하였습니다.

9. 유저 피드백(더 자세한 자료는 geniewebtoon feedback.xlsx 참고 요망)

총 30여명에게 구글폼 설문지를 배포하여 유저 피드백을 수용했고, 크고작은 여러 아쉬웠던점과 추가됐으면 하는점을 받아볼 수 있었습니다. 인원의 60퍼센트는 it관련 학과 재학중인 대학생 또는 관련 분야에 종사하는 직장인들이였고 나머지 40퍼센트는 이와 관련 없는 대학생 또는 직장인들이였습니다. 나이대는 20대가 76퍼센트로 가장 많았고 나머지 24퍼센트는 전부 30대 이상이였습니다. 사이트 사용 플랫폼으로는 60퍼센트가 모바일 환경이였고 나머지 40퍼센트는 pc환경에서 사용하여 피드백을 제시해주었습니다.

9.1 회원가입, 로그인, 아이디/비밀번호 찾기

회원가입 영역에서 가장 많은 피드백이 들어온 요인은 주소를 입력받을 때 현재 많은 사이트에서 사용되는 다음 주소FORM을 사용해 주소 검색을 통해 간편하게 주소 입력이 가능하도록 되었으면 좋겠다는 피드백이었습니다.

이외에도 몇 가지 피드백으로는 첫째 비밀번호 관련 규칙 명시와 재입력 비밀번호와 비밀번호간의 일치 불일치 여부 제공에 대한 편의성과 관련된 요구사항이 많았습니다.

둘째, 비밀번호는 입력될때 보안을 위해 가려지도록 설정을 하였지만 이를 중간에 확인할 수 있도록(visible) 설정이 가능한 기능이 있으면 좋겠다는 피드백이였습니다.

마지막으로는 이메일 인증을 완료해야 로그인이 정상적으로 가능하다는 문구가 강조되었으면 좋겠다는 피드백이였습니다. 이를 바탕으로 위에 작성된 피드백을 전부 수용하였고 수정 반영하였습니다.

로그인, 아이디/비밀번호 찾기에서는 대부분 좋은 평가 대다수 였어서 특별히 표기할만한 공통된 피드백은 없었습니다.

9.2 메인화면

메인화면 영역에서 가장 많은 피드백이 들어온 요인은 크게 두가지 였습니다.

첫 번째로는, 웹툰 사이트 메인화면인데 네x버,카x오와 같은 기존 대형 플랫폼은 이미지 자료가 풍부하여 시각적으로 내가 웹툰 사이트에 왔다는 인식이 생기는 반면, 제 사이트는 들어오자마자 4분할의 영역과 글씨만 보여 허전한 느낌이 든다는 평가였습니다.

두 번째로는, 웹툰 사이트치고 애매하게 어두운 회색과 흰색 글씨가 주는 밋밋하고 사용자로써 어색하고 불편하는 느낌이 들었다 라는 것이였습니다.

우선 첫 번째 피드백에 대해서는 기존 웹툰 사이트와 차별화되는 독창적인 사용자 경험을 주고자 계획한 호버 및 확대 기능을 구현한 메인화면이여서 웹툰 이미지를 넣기에는 제 의도와는 부적합하는 판단이 섰습니다. 그렇기에 수정을 강행하진 못했지만 사용자의 입장에선 충분히 제기될 수 있는 문제라는 점을 인지하여 시각적 자료가 주는 이점에 대해 다시 생각해볼 수 있는 기회였습니다.

두 번째 피드백에 대해서는 확실한 색체 대비와 임펙트를 주기위해 색상코드 #929292에서 #4f4f4f로 바꾸어 더 진한 회색과 흰색 글씨 대신 색상코드로 네온 오렌지 색깔을 채택하여 확실한 채도 구분과 제 사이트만의 아이덴티티를 확립하고자 하였습니다.

9.3 장르별

장르별에서 가장 많은 피드백이 들어온 요소는 크게 두 가지였습니다. 첫번째는 랜더링 속도에 대한 지적이 가장 많았습니다. 현재 평균적인 랜더링 시간이 google devtools에서 확인되는 바로는 2s정도 소요됩니다. 이는 길다고 보긴 어렵지만 결코 짧다고 볼 수는 없고 LCP 점수 기준으로 높다고 볼순 없습니다.

이를 개선하기 위해선 Vercel의 서버 region이 america로 설정된 것을 aisa/seoul로 바꾸면 눈에 띄게 개선되지만 이에는 년간 3만원이라는 대학생에게는 비교적 부담스러운 금액으로 인해 적용하지 못했습니다. 또한 추가적인 방법으로는 AWS Cloudfront를 사용하여 S3에 저장된 정적 이미지 로딩을 더 빠르도록 해주는 방법이 있지만 이는 현재 랜더링 시간이 2s정도이기에 cloudfront를 사용할 정도로 무거운 사이트 또는 이미지들이 아

니기에 해당 방법을 채용하진 않았습니다.

두 번째로는, 댓글 영역에서 좋아요와 싫어요 아이콘 클릭시 화면에 반응 표시까지 1.5s~2s정도의 시간이 소요되는 것에 대한 피드백이였습니다. 이 또한 위와 마찬가지로 Vercel server region 설정 문제라고 확인하였습니다.

9.4 랭킹

랭킹에서 가장 많이 들어온 피드백은 기간별 랭킹도 좋지만 장르별, 성별, 나이대별 랭킹등을 신설하여 여러 분야의 랭킹들을 보여주면 좋을거 같다는 의견이 많았습니다. 허나 혼자하는 프로젝트의 특성상 설계수정부터 코드 수정까지 정해진 기간 시간 내에 하기에는 무리가 있다고 판단하여 해당 피드백은 수용까지는 진행하지 못하였습니다.

9.5 마이페이지

마이페이지 메뉴 아래 세부 메뉴까지 사용자들에게서 공통적이나 추가 또는 수정을 요구하는 피드백이 없었습니다.

9.6 지니와 함께하는 웹툰생활

지니와 함께하는 웹툰생활에서는 도전! 골든벨에 대한 피드백이 있었습니다. 우선, 결과나 등수 및 상위 몇퍼센트인지 본인의 위치 또는 실력을 확인할 수 있는 지표가 있으면 좋겠다는 피드백이였습니다. 또한, 기존 코드로는 문제가 DB에 등록된 모든 랜덤 문제를 풀어야하는 형식이였기에 웹툰을 선택해서 풀 수 있었으면 좋겠다는 피드백이 들어왔습니다.

우선 첫 번째 피드백은 문제를 풀고 정답인지 아닌지를 알 수 있도록 UI를 추가하고 등수 또는 상위 몇퍼센트인지를 만드는 기능은 피드백이 전부 끝나고 받은 피드백이라 유저 데이터를 형성하는데 어려움이 있어 진행하지는 못했습니다. 허나, 사용자들이 본인의 등수나 성적과 관련한 컨텐츠에 꽤나 큰 흥미와 관심을 갖는다는 중요한 피드백을 수용하게 되었습니다.

두 번째 피드백에 대해선 DB에 등록된 웹툰별로 질문을 4개씩 생성후 본인의 해당 웹툰 구독 여부를 확인하여 구독자일 경우 문제를 풀어볼 수 있도록 코드를 수정하였습니다.

9.7 종합의견, 전반적 피드백

전반적인 사이트에 대한 피드백에선 크게 두 가지 피드백이 있었습니다. 첫 번째로, 기존 Header,Nav,Body,Footer 구조의 웹사이트의 형식이 아닌 호버기반 메인화면 구성으로 인해 정말 여러곳에서 피드백을 받았습니다. 기존 웹사이트가 주는 가장 큰 이점은 Nav를 사용한 메인화면으로의 복귀, 원하는 메뉴로의 상시 접근성입니다. 허나 제 웹툰 사이트느 호버 기반의 메인화면과 그로 인해 과감하게 삭제한 Nav로 인해 사용자들이 적지않은 당혹감에 피드백을 받았습니다.

두 번째로는 웹툰 사이트인데 그와 매칭되지 못하는 페이지 디자인과 색상에 대한 피드백을 받았습니다. 대형 플랫폼이자 현재 업계 선두를 달리는 네x버 웹툰과 카x오웹툰을 살펴보면 메인페이지부터 많은 웹툰 썸네일과 더불어 흰색 배경에 초록색 포인트 컬러, 검은색 배경에 흰색 글씨를 통한 확실한 색체 대비를 사용하여 충분한 시각적 효과를 표현했습니다. 허나 초기 화면설계서를 작성할 당시 회색과 흰색을 사용한 깔끔하고 모던한 느낌의 센세이셔널한 웹툰 사이트를 만들어보겠다는 생각에 기획과 구현을 시작했지만 막상 구현후 그리 좋지 못한 피드백을 받게 되었습니다.

우선 첫 번째 피드백에 대해선 충분히 있을 수 있는 피드백이였다고 생각합니다. 오랫동안 전통적인 웹사이트 구조로서 사용되는 틀에서 벗어나 사용자에게 신선하며 독창적인 경험을 주고자 실험적인 도전을 해보았습니다. 허나 제 생각에는 기존 누구나 알고 만드는 웹사이트를 만드는 것보다 독창적으로 사용자에게 새로운 경험을 줄 수 있는 웹사이트를 만들겠다는 계획에서 시작됐던 지니웹툰 프로젝트이기에 피드백을 수용하지만 기술적인 보안과 더 많은 경험을 통해 추후에는 더 완전한 웹사이트를 제작할 것입니다.

두 번째 피드백에 대해서는 저도 제작하고 피드백을 받으면서 더 깊게 느낀 피드백입니다. 웹툰 사이트스러운 포인트 컬러나 시각적 효과가 좀더 뒷받침이 되었다면 훨씬 사용자의 이목도 끌고 좋은 웹툰 사이트가 될 수 있었다고 생각됩니다.

10. 기대효과 및 활용

10.1 서비스·비즈니스 효과

서비스 및 비즈니스적 기대효과로는 아래와 같은 효과들이 있습니다.

첫째, 무료 감상과 광고 기반 수익모델을 정립함으로써 전면 무료 감상을 기본으로 하고 회차 말미/홈 영역 배너 운영으로 노출·클릭 로그를 축적하여 광고 효율을 정량화합니다.

둘째, AI 추천·퀴즈로 체류시간 증대 효과를 기대해볼 수 있습니다 지니와 함께하는 웹툰생활의 AI 추천(구독/취향 기반)과 '도전! 골든벨' 퀴즈를 통해 탐색–참여–재방문 루프를 형성, 주당 세션 수와 평균 체류시간 상승을 기대합니다.

셋째, 랭킹·장르·에피소드 구조로 탐색 효율 향상을 기대해볼 수 있습니다. 일·주·월·연간 랭킹과 장르 진입 흐름(호버 확대 → 세부 메뉴 → 상세/뷰어)으로 빠른 진입과 심화 탐색을 동시에 지원합니다.

10.2 운영·데이터 효과

운영 및 데이터 기대효과로는 아래와 같은 효과들이 있습니다.

첫째, 데이터 기반 의사결정 파이프라인으로 RDS(MySQL) + Sequelize 모델로 뷰/구독/댓글/광고 로그를 정규화 저장 → 관리자 대시보드 지표화 → 랭킹/광고/콘텐츠 전략에 재투입(AB 운영 가능)이 가능하다는 점입니다.

둘째, 시간대·작품·회원 특성별 성과 비교로 배너 위치/빈도/캠페인 온오프를 탄력 조정하여 광고 효율을 극대화시킬 수 있습니다.

셋째, Next.js App Router, Sequelize, S3 조합의 표준 스택으로 온보딩 비용을 낮추고, 마이그레이션·시더 기반 DB 변경 내성을 확보하여 지속 가능 아키텍처를 구성했습니다.

10.3 UX·접근성 효과

UX 및 접근성 기대효과로는 아래와 같은 효과들이 있습니다.

첫째, 호버 기반 메인·명확한 정보 구조를 통해 '호버→확대→세부 메뉴' 경험으로 차별화된 탐색성을 제공하고, 색채 대비/썸네일 가시성 개선을

지속합니다.

둘째, 반응형·지연 로딩·SSR/ISR 전략을 사용하여 모바일/데스크톱 모두에서 초기 렌더를 가볍게 유지하고 이미지 지연 로딩을 적용, LCP 등 핵심 지표 개선을 지향합니다.

11. 자기평가(초안)

11.1 잘한 점

이번 풀스택 개인 프로젝트를 진행하면서 자기평가를 해봤을때 아래와 같은 잘한 점들이 있었습니다.

첫째, 설계, 구현, 테스트, 배포의 전과정을 혼자 주도적으로 했습니다. IA·ERD·화면설계서·요구사항을 문서화하고 이를 Next.js API/서비스/모델로 일관 매핑하여 유지보수성을 확보했습니다.

둘째, 표준·안정 스택과 엄격한 TS 설정을 적용시켜 TypeScript strict, ESLint/Prettier, 경로 alias, 최신 React/Next/Sequelize/mysql2 조합으로 품질 게이트를 마련했습니다.

셋째, 운영·보안 기본기 준수를 준수하여 작업하였습니다. .env 분리, NextAuth 세션/역할 보호, 비밀번호 해시, S3 프리사인 업로드, 미들웨어 접근 제어를 반영하여 실제 업무의 로직을 최대한 적용하고자 했습니다.

마지막으로 실제 배포 후 사용자들에게 피드백 수렴함으로써 주소 검색, 비밀번호 규칙·가시화, 이메일 인증 흐름 등 실사용 이슈를 빠르게 반영해 UX 균형을 도모했습니다.

11.2 아쉬운 점·한계

이번 풀스택 개인 프로젝트를 진행하면서 자기평가를 해봤을때 위와 같은 잘한 점도 있었지만 아쉬운 점 또한 있었습니다.

첫째, 리전/네트워크 지연과 이미지 최적화가 CDN/리사이즈/캐시 고도화의 일부 항목 일정·비용 제약으로 미뤄졌습니다.

둘째, 기간 외 장르/연령/성별 세분 랭킹까지 세분화된 랭킹 목록으로 확장하지 못했습니다.

마지막으로 퀴즈 리더보드·성과 지표를 구현하지 못했습니다. 이는 사용

자 피드백 단계에서 피드백이 전부 완료된후 확인된 정보라 다시 코드 수정후 사용자들에게 해당 데이터 수집을 하기에는 현실적 문제로 인해 정답률/백분위/누적 포인트 등 경쟁·보상 장치가 구현되지 못했습니다.

11.3 성장·학습 포인트

이번 풀스택 개인 프로젝트를 진행하면서 자기평가를 해봤을때 성장 및 학습 포인트는 아래와 같았습니다.

첫째, 초반 프로젝트 기획 및 구성 단계에서의 문서가 얼마나 중요한지 느낄 수 있었던 프로젝트였습니다. 초반부에 혼자 작성한 기획서와 화면설계서등에서 애매한 워딩이나 자료때문에 구현 단계에서 애를 먹은 적이 정말 많았습니다. 이를 계기로 프로젝트 기획 및 구성 단계가 얼마나 중요하고 프로젝트에 큰 비중을 차지하는지 체감하게 되었습니다.

둘째, 문서 ↔ 코드 정합성 유지 루틴을 확립할 수 있었습니다. IA/ERD/화면 설계서/요구 문서와 폴더링·타입·모듈 구조를 지속 동기화하는 것의 필요성과 중요성을 아주 깊게 느꼈습니다.

셋째, 데이터 운영 역량을 늘릴 수 있었습니다. 마이그레이션/시더 체계로 스키마 진화와 운영 장애(외래키/인덱스/열 변경) 대응을 통해 실제 운영시 발생할 수 있는 상황과 문제들을 미리 경험해볼 수 있었습니다.

마지막으로 TC(테스트 케이스)의 중요성을 다시 한번 느끼게 되었습니다. 대학교 팀플전공 수업에서 진행했던 TC와 다르게 처음부터 끝까지 모든 부분을 혼자서 확인하려다보니 완벽하게 모든 TC를 잡지는 못했었습니다. 그로 인해 사용자로부터 에러에 관한 문의가 사용자 피드백중 두 건이 들어왔었습니다. 이를 통해 TC의 중요성을 다시 한 번 새길 수 있었습니다.

11.4 향후 개선 로드맵

향후 개선 로드맵은 아래와 같이 같습니다.

첫째, 성능적으로는 S3 썸네일 파이프라인(AWS CloudFront)를 사용하여 이미지 로딩 속도를 개선할 예정입니다. 또한 Vercel의 pro 버전을 구매하여 server region을 asia/seoul로 설정하여 보다 빠른 랜더링 속도를 기대

해볼 수 있을 것입니다. 또한 RUM(실제 사용자 모니터링)을 기반으로 LCP(핵심 내용이 화면에 표시되기까지의 시간), INP(사용자 입력에 대한 반응성)측정을 통해 사용자 경험 지표를 백엔드 데이터 처리 속도 향상으로 이끌어 가볼 수 있습니다. 뿐만 아니라 Vercel의 Edge 런타임 캐싱 작업과 ISR 강화를 통한 프런트엔드단의 랜더링 속도 향상 또한 기대해볼 수 있습니다.

둘째, 도전! 골든벨에서 사용자들의 웹툰별 정답률을 수집하여 해당 데이터를 기반으로 각 웹툰별로 문제를 푼 사용자가 상위 몇퍼센트인지를 확인할 수 있는 기능과 맞춘 문제수나 상위 퍼센트에 따라 차등하게 포인트를 지급하여 지니웹툰 내에서 상품 교환을 할 수 있는 기능을 고려해볼 수 있을 것입니다.

셋째, 사용자들에게 많은 피드백을 받았던 디자인적 개선을 위해 전반적인 배경색과 포인트 컬러의 재선정 또는 채도 및 구성 변경을 통해 시각적 효과를 개선해볼 수 있을 것입니다.

12. 참고문헌

12.1 내부 산출물(주요 근거 문서)

Genie Webtoon 개발 보고서: IA, ERD, 아키텍처/요구/테스트/운영 전반의 기준 문서

Genie Webtoon 화면설계서(PPTX): 주요 화면 흐름, 랭킹/광고/댓글/뷰어 동작 명세

Genie Webtoon 계획서(DOCX): 목표/시장분석/요구사항/DB 개념·논리 설계

Genie Webtoon 사용자(UI_UX) 설문지(응답).xlsx: 사용자 피드백 내용이 담긴 xlsx 파일

GenieWebtoon-Test-Cases.xlsx : 테스트 케이스를 적고 작동 환경에 따라 비고와 응답 결과를 기재한 xlsx 파일

프로젝트 설정 파일: tsconfig.json, package.json, ESLint 설정 등(Strict/경로 alias/의존성 기준)

12.2 외부 레퍼런스(기술 문서·모범사례)

Next.js 공식 문서(App Router, Route Handlers, ISR/SSR, 이미지 최적화).

React & TypeScript 공식 문서(Strict Mode, 최적화 패턴)

Sequelize & mysql2 공식 문서(모델링, 마이그레이션/시더, 트랜잭션).

MySQL 8.0 가이드(인덱스/제약/실행계획 튜닝)

AWS RDS for MySQL & S3 베스트 프랙티스(IAM 최소권한, Presigned URL, CORS, 백업/복구)

NextAuth 문서(세션/JWT 콜백, Credentials/Email Provider, Role 기반 보호)

Styled-Components / Tailwind CSS(디자인 시스템, 반응형/접근성 패턴)

SWR / Zustand / React Hook Form / Zod(데이터 패칭·상태·폼·유효성 도구)

Vercel Docs(빌드·배포·로그 관측·Edge/Region 구성)