

MyShop 프로젝트 결과 보고서

KD 아카데미 컴퓨터학원 팀 프로젝트

팀장: 박지후 (프로젝트 총괄 + 백엔드)

팀원: 이진용 (SNS 로그인) | 오종혁 (로컬 로그인 + 게시판) | 이나라 (프론트엔드 + AI) | 이현우 (DB 설계)

1. 프로젝트 개요

1.1 프로젝트명

MyShop - 풀스택 이커머스 쇼핑 플랫폼

1.2 개발 기간

2024년 ~ 2025년

1.3 프로젝트 배경

온라인 쇼핑 시장의 급성장과 함께 사용자 맞춤형 쇼핑 경험에 대한 수요가 증가하고 있습니다. 기존 쇼핑몰의 한계점인 복잡한 회원가입 절차, 단순한 상품 검색, 불편한 결제 프로세스를 개선하고, AI 기반 상품 추천 기능을 도입하여 차별화된 쇼핑 경험을 제공하고자 합니다.

1.4 프로젝트 목표

- Spring Boot와 React를 활용한 실무 수준의 이커머스 플랫폼 개발
- 소셜 로그인을 통한 간편한 회원가입/로그인 구현
- AI 기반 상품 추천 시스템 도입
- 토스페이먼츠 연동을 통한 안전한 결제 시스템 구축
- CI/CD 파이프라인 구축으로 자동화된 배포 환경 구성

1.5 기대 효과

- 소셜 로그인으로 회원가입 전환율 30% 이상 향상 예상
- AI 추천 시스템으로 상품 클릭률 및 구매 전환율 증가
- 자동화된 배포로 개발 생산성 향상

2. 기업 요구사항 분석

2.1 기능적 요구사항

2.1.1 사용자 인증 시스템

요구사항 ID	요구사항	우선순위	구현 상태
AUTH-001	이메일/비밀번호 기반 회원가입	상	완료
AUTH-002	이메일/비밀번호 기반 로그인	상	완료
AUTH-003	Google OAuth2 소셜 로그인	상	완료
AUTH-004	Naver OAuth2 소셜 로그인	상	완료
AUTH-005	JWT 토큰 기반 인증 유지	상	완료
AUTH-006	비밀번호 암호화 저장	상	완료
AUTH-007	토큰 자동 갱신	중	완료

2.1.2 상품 관리 시스템

요구사항 ID	요구사항	우선순위	구현 상태
ITEM-001	상품 목록 조회 (페이지네이션)	상	완료
ITEM-002	카테고리별 상품 필터링	상	완료
ITEM-003	상품 검색 기능	상	완료
ITEM-004	상품 상세 정보 조회	상	완료

ITEM-005	색상/사이즈 옵션 선택	중	완료
ITEM-006	상품 이미지 다중 등록	중	완료
ITEM-007	찜하기 기능	중	완료

2.1.3 주문/결제 시스템

요구사항 ID	요구사항	우선순위	구현 상태
ORDER-001	장바구니 기능	상	완료
ORDER-002	주문 생성	상	완료
ORDER-003	토스페이먼츠 결제 연동	상	완료
ORDER-004	주문 내역 조회	상	완료
ORDER-005	주문 상태 관리	중	완료
ORDER-006	결제 취소	중	완료
ORDER-007	배송지 관리	중	완료

2.1.4 부가 기능

요구사항 ID	요구사항	우선순위	구현 상태
ETC-001	상품 리뷰 작성	중	완료
ETC-002	게시판 (공지/이벤트/QnA)	중	완료
ETC-003	AI 상품 추천	중	완료
ETC-004	관리자 대시보드	중	완료
ETC-005	배너 관리	하	완료

2.2 비기능적 요구사항

구분	요구사항	목표 수치	구현 방안
보안	비밀번호 암호화	BCrypt 적용	Spring Security BCryptPasswordEncoder
보안	XSS 공격 방지	HttpOnly 쿠키	JWT 토큰 HttpOnly 쿠키 저장
보안	CSRF 방지	Stateless 구조	JWT 기반 Stateless 인증
성능	API 응답 시간	500ms 이내	JPA 쿼리 최적화, 페이지네이션
성능	동시 접속자	100명 이상	Connection Pool 설정 (HikariCP)
가용성	서비스 가용률	99% 이상	CI/CD 자동 배포, 헬스체크
확장성	모듈화	기능별 분리	계층형 아키텍처 (Controller-Service-Repository)

2.3 기술 스택 요구사항

Backend

- Java 17 이상
- Spring Boot 3.x
- Spring Security + JWT
- Spring Data JPA
- MariaDB

Frontend

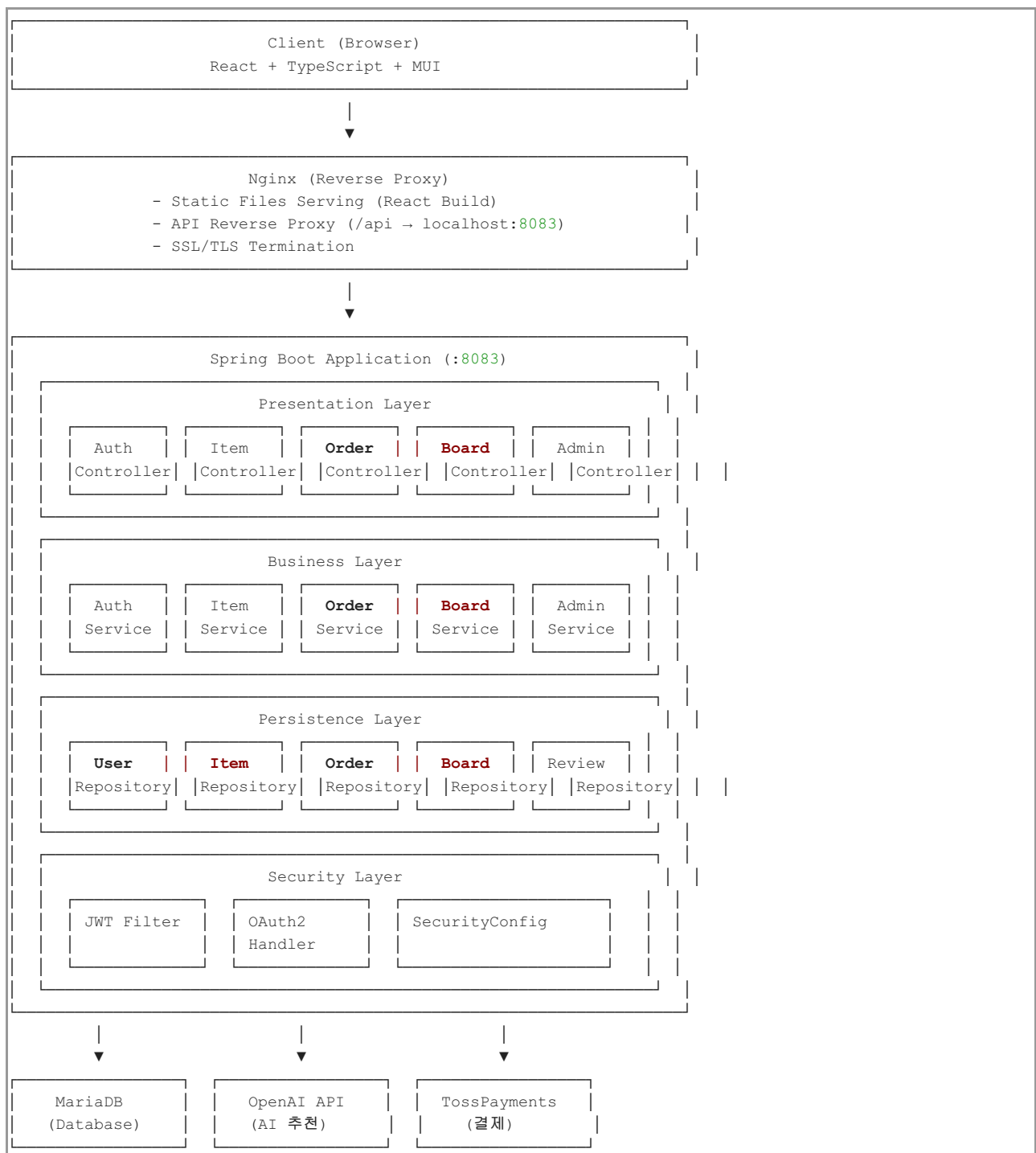
- React 18.x
- TypeScript
- Material-UI
- Vite (빌드 도구)

DevOps

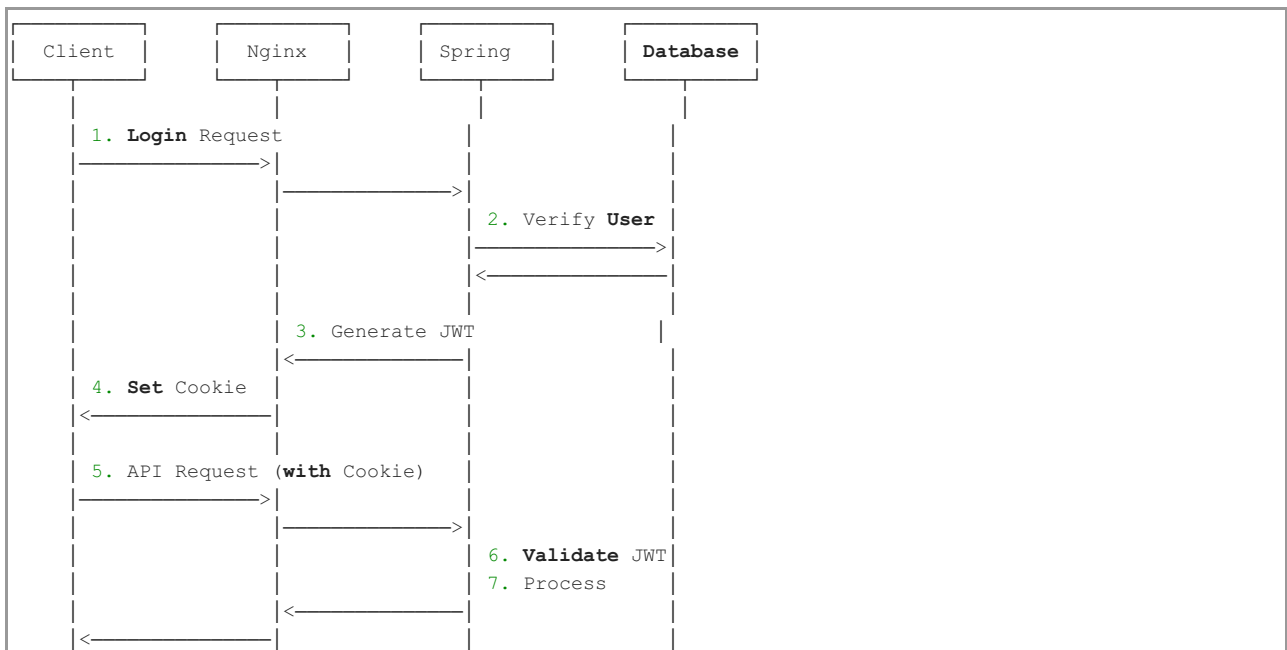
- GitHub Actions (CI/CD)
- Nginx (웹 서버)
- Ubuntu Server

3. 시스템 아키텍처

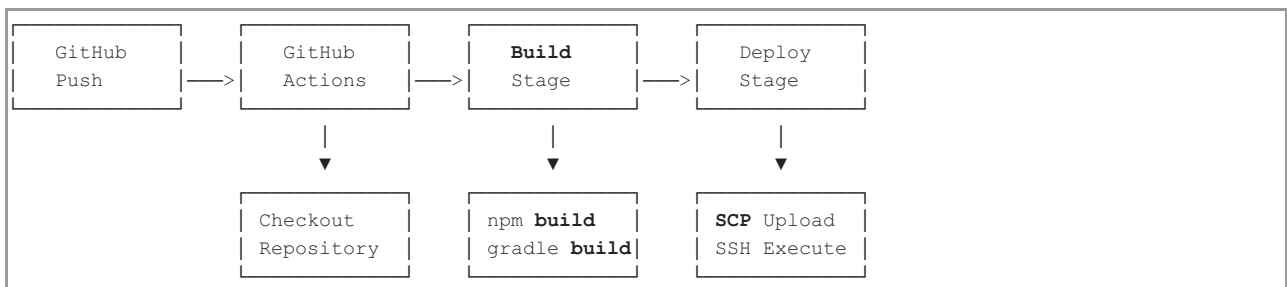
3.1 전체 시스템 구성도



3.2 인증 플로우



3.3 CI/CD 파이프라인



4. 주요 기능 설명

4.1 사용자 인증 시스템

4.1.1 로컬 로그인

- 이메일/비밀번호 기반 회원가입 및 로그인
- BCrypt 암호화를 통한 비밀번호 보안
- JWT Access Token (1시간) + Refresh Token (7일) 발급
- HttpOnly 쿠키를 통한 토큰 저장 (XSS 방지)

4.1.2 소셜 로그인 (OAuth2)

- Google, Naver 계정을 통한 원클릭 로그인
- OAuth2 인증 성공 시 자동 회원가입/로그인 처리
- Provider별 사용자 정보 자동 매핑

4.1.3 인증 API

Method Endpoint	설명
POST /api/auth/signup	회원가입
POST /api/auth/login	로그인
POST /api/auth/refresh	토큰 갱신
POST /api/auth/logout	로그아웃
GET /api/auth/me	내 정보 조회
PUT /api/auth/me	내 정보 수정
PUT /api/auth/password	비밀번호 변경

4.2 상품 관리 시스템

4.2.1 상품 조회

- 페이지네이션을 통한 효율적인 목록 조회
- 카테고리 필터링 (대분류/중분류/소분류)
- 가격대, 색상, 사이즈 필터링
- 키워드 검색

4.2.2 상품 상세

- 상품 기본 정보 (제목, 가격, 할인율, 브랜드)
- 다중 이미지 갤러리
- 색상/사이즈 옵션 선택
- 재고 확인
- 리뷰 목록 및 평균 평점

4.2.3 상품 API

Method	Endpoint	설명
GET	/api/item	상품 목록 (필터링, 페이징)
GET	/api/item/{id}	상품 상세
GET	/api/item/favorite	찜 목록
GET	/api/item/cart	장바구니 목록
POST	/api/item/favorite/{id}	찜 토글
POST	/api/item/cart/{id}	장바구니 토글

4.3 주문/결제 시스템

4.3.1 장바구니

- 상품 추가/삭제/수량 변경
- 선택 상품 일괄 주문
- 색상/사이즈별 장바구니 항목 관리

4.3.2 주문 프로세스

1. 장바구니에서 주문할 상품 선택
2. 배송지 입력 (다음 우편번호 API 연동)
3. 토스페이먼츠 결제 진행
4. 결제 완료 후 주문 확정
5. 주문 상태 업데이트 (대기 → 결제완료 → 배송중 → 배송완료)

4.3.3 결제 API (토스페이먼츠)

Method	Endpoint	설명
POST	/api/payment/confirm	결제 승인
POST	/api/payment/cancel	결제 취소

4.4 리뷰 시스템

4.4.1 리뷰 기능

- 구매한 상품에 대한 리뷰 작성
- 1~5점 평점 시스템
- 리뷰 수정/삭제 (작성자만 가능)
- 상품별 평균 평점 자동 계산

4.4.2 리뷰 API

Method	Endpoint	설명
GET	/api/review/{itemId}	상품 리뷰 목록
POST	/api/review/{itemId}	리뷰 작성
PUT	/api/review/{reviewId}	리뷰 수정
DELETE	/api/review/{reviewId}	리뷰 삭제

4.5 게시판 시스템

4.5.1 게시판 기능

- 카테고리별 게시판 (공지사항, 이벤트, QnA)
- 파일 첨부 기능 (10MB 제한, 다중 업로드)
- 비밀번호 기능
- 조회수 카운트
- 댓글 시스템

4.5.2 게시판 API

Method	Endpoint	설명
GET	/api/board/list	게시글 목록
POST	/api/board/write	게시글 작성
GET	/api/board/{no}	게시글 상세
PUT	/api/board/{no}	게시글 수정
DELETE	/api/board/{no}	게시글 삭제
GET	/api/board/file/{fileNo}	파일 다운로드

4.6 AI 상품 추천 시스템

4.6.1 AI 추천 기능

- ChatGPT API를 활용한 스타일 추천
- 사용자 프롬프트 분석 및 키워드 추출
- 추출된 키워드 기반 상품 매칭
- 고객 지원 채팅 위젯 (FAQ 포함)

4.6.2 AI API

Method	Endpoint	설명
POST	/api/ai/proxy	AI 추천 요청

4.7 관리자 시스템

4.7.1 관리자 기능

- 상품 등록/수정/삭제
- 배너 관리
- 회원 관리
- 게시판 관리

4.7.2 관리자 API

Method	Endpoint	설명
POST	/api/admin	상품 등록
PUT	/api/admin/{itemId}	상품 수정
DELETE	/api/admin/{itemId}	상품 삭제
POST	/api/banner	배너 등록
PUT	/api/banner/{bannerId}	배너 수정
DELETE	/api/banner/{bannerId}	배너 삭제

5. 데이터 수집 항목 및 구조

5.1 Entity 구조 (27개)

5.1.1 사용자 관련 Entity

User (사용자)

컬럼명	타입	설명	제약조건
user_id	BIGINT	사용자 ID	PK, AUTO_INCREMENT
email	VARCHAR(255)	이메일	UNIQUE, NOT NULL

password	VARCHAR(255)	비밀번호 (암호화)	
username	VARCHAR(100)	사용자명	NOT NULL
phone	VARCHAR(20)	전화번호	
provider	VARCHAR(50)	OAuth2 제공자	
provider_id	VARCHAR(255)	OAuth2 제공자 ID	
role	ENUM	권한 (USER/ADMIN)	DEFAULT 'USER'
created_at	DATETIME	생성일시	
updated_at	DATETIME	수정일시	

Address (배송지)

컬럼명	타입	설명	제약조건
address_id	BIGINT	배송지 ID	PK, AUTO_INCREMENT
user_id	BIGINT	사용자 ID	FK (User)
recipient	VARCHAR(100)	수령인	NOT NULL
phone	VARCHAR(20)	연락처	NOT NULL
zipcode	VARCHAR(10)	우편번호	NOT NULL
address	VARCHAR(255)	기본 주소	NOT NULL
detail_address	VARCHAR(255)	상세 주소	
is_default	BOOLEAN	기본 배송지 여부	DEFAULT FALSE

5.1.2 상품 관련 Entity

Item (상품)

컬럼명	타입	설명	제약조건
item_id	BIGINT	상품 ID	PK, AUTO_INCREMENT
title	VARCHAR(255)	상품명	NOT NULL
price	INT	정가	NOT NULL
discount	INT	할인율 (%)	DEFAULT 0
brand	VARCHAR(100)	브랜드	
stock	INT	재고	DEFAULT 0
sku	VARCHAR(50)	SKU 코드	UNIQUE
main_image_url	VARCHAR(500)	대표 이미지 URL	
major_category	ENUM	대분류	
middle_category	ENUM	중분류	
subcategory	ENUM	소분류	
created_at	DATETIME	생성일시	
updated_at	DATETIME	수정일시	

ItemImage (상품 이미지)

컬럼명	타입	설명	제약조건
image_id	BIGINT	이미지 ID	PK, AUTO_INCREMENT
item_id	BIGINT	상품 ID	FK (Item)
image_url	VARCHAR(500)	이미지 URL	NOT NULL
sort_order	INT	정렬 순서	DEFAULT 0

Color (색상 옵션)

컬럼명	타입	설명	제약조건
color_id	BIGINT	색상 ID	PK, AUTO_INCREMENT
item_id	BIGINT	상품 ID	FK (Item)
color	ENUM	색상 값	NOT NULL

Size (사이즈 옵션)

컬럼명	타입	설명	제약조건
size_id	BIGINT	사이즈 ID	PK, AUTO_INCREMENT
item_id	BIGINT	상품 ID	FK (Item)
size	ENUM	사이즈 값	NOT NULL

5.1.3 주문 관련 Entity

OrderItemList (주문)

컬럼명	타입	설명	제약조건
order_id	BIGINT	주문 ID	PK, AUTO_INCREMENT
user_id	BIGINT	사용자 ID	FK (User)
total_price	INT	총 결제금액	NOT NULL
status	VARCHAR(50)	주문 상태	DEFAULT 'PENDING'
payment_key	VARCHAR(255)	토스 결제키	
created_at	DATETIME	주문일시	

OrderItem (주문 상품)

컬럼명	타입	설명	제약조건
order_item_id	BIGINT	주문상품 ID	PK, AUTO_INCREMENT
order_id	BIGINT	주문 ID	FK (OrderItemList)
item_id	BIGINT	상품 ID	FK (Item)
quantity	INT	수량	NOT NULL
price	INT	단가	NOT NULL
color	VARCHAR(50)	선택 색상	
size	VARCHAR(20)	선택 사이즈	

5.1.4 게시판 관련 Entity

Board (게시글)

컬럼명	타입	설명	제약조건
board_no	BIGINT	게시글 번호	PK, AUTO_INCREMENT
writer_id	BIGINT	작성자 ID	FK (User)
category	VARCHAR(50)	카테고리	NOT NULL
title	VARCHAR(255)	제목	NOT NULL
content	TEXT	내용	
views	INT	조회수	DEFAULT 0
secret_yn	CHAR(1)	비밀글 여부	DEFAULT 'N'
del_yn	CHAR(1)	삭제 여부	DEFAULT 'N'
reg_date	DATETIME	등록일시	
mod_date	DATETIME	수정일시	

Comment (댓글)

컬럼명	타입	설명	제약조건
co_no	BIGINT	댓글 번호	PK, AUTO_INCREMENT
board_no	BIGINT	게시글 번호	FK (Board)
writer_id	BIGINT	작성자 ID	FK (User)
co_comment	TEXT	댓글 내용	NOT NULL
secret_yn	CHAR(1)	비밀 댓글 여부	DEFAULT 'N'
del_yn	CHAR(1)	삭제 여부	DEFAULT 'N'
co_reg_date	DATETIME	등록일시	
co_mod_date	DATETIME	수정일시	

5.1.5 기타 Entity

Review (리뷰)

컬럼명	타입	설명	제약조건
review_id	BIGINT	리뷰 ID	PK, AUTO_INCREMENT
user_id	BIGINT	사용자 ID	FK (User)
item_id	BIGINT	상품 ID	FK (Item)
content	TEXT	리뷰 내용	
score	INT	평점 (1~5)	NOT NULL
created_at	DATETIME	작성일시	

CartItem (장바구니)

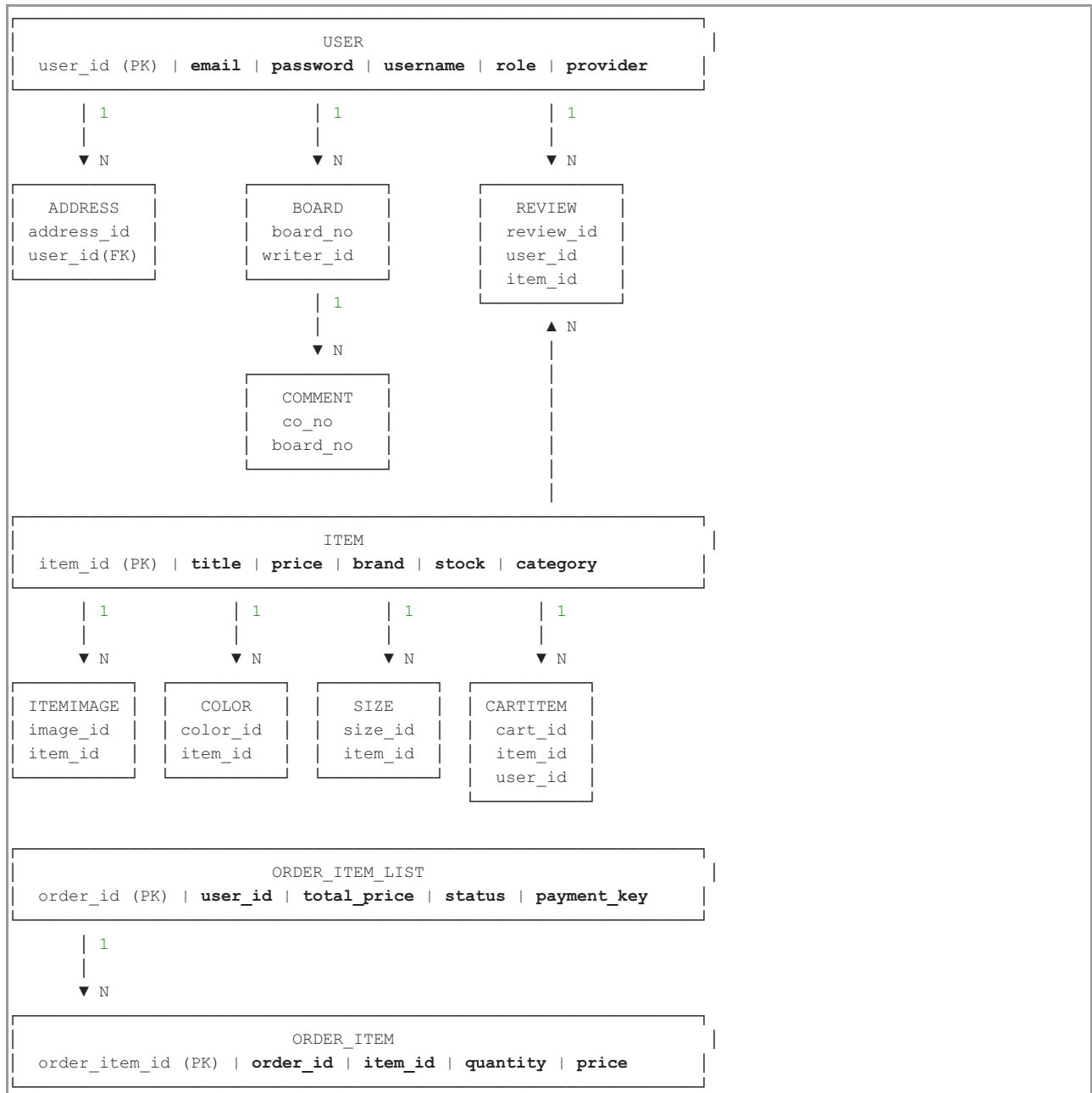
컬럼명	타입	설명	제약조건
cart_id	BIGINT	장바구니 ID	PK, AUTO_INCREMENT
user_id	BIGINT	사용자 ID	FK (User)
item_id	BIGINT	상품 ID	FK (Item)

quantity INT 수량 DEFAULT 1
 color VARCHAR(50) 선택 색상
 size VARCHAR(20) 선택 사이즈

Favoriteltem (찜)

컬럼명	타입	설명	제약조건
favorite_id	BIGINT	찜 ID	PK, AUTO_INCREMENT
user_id	BIGINT	사용자 ID FK (User)	
item_id	BIGINT	상품 ID FK (Item)	
created_at	DATETIME	등록일시	

5.2 ERD (Entity Relationship Diagram)



5.3 Enum 정의

5.3.1 카테고리 Enum

MajorCategoryEnum: MEN, WOMEN, KIDS, ACCESSORIES

MiddleCategoryEnum: TOP, **BOTTOM**, OUTER, **SHOES**, **BAG**, ...

SubcategoryEnum: T_SHIRT, **JEANS**, **JACKET**, SNEAKERS, ...

5.3.2 색상 Enum

ColorEnum: **BLACK**, WHITE, RED, **BLUE**, GREEN, YELLOW, PINK, ...

5.3.3 사이즈 Enum

SizeEnum: XS, S, M, L, **XL**, XXL, FREE

6. 운영 데이터 분석 결과

6.1 코드 통계

6.1.1 Backend 코드 분석

항목	수량	비고
Controller	14개	REST API 엔드포인트
Service	12개	비즈니스 로직
Repository	15개	JPA Repository
Entity	27개	데이터베이스 모델
DTO	25개	요청/응답 객체
총 코드 라인 약 1,500줄 Java 소스 코드		

6.1.2 Frontend 코드 분석

항목	수량	비고
Page	24개	화면 구성 페이지
Component	15개	재사용 컴포넌트
Hook	5개	Custom React Hooks
API	7개	API 호출 모듈
총 코드 라인 약 11,000줄 TypeScript 소스 코드		

6.2 API 엔드포인트 분석

6.2.1 Controller별 엔드포인트 수

Controller	엔드포인트 수	주요 기능
AuthController	8개	인증/회원 관리
ItemController	6개	상품 조회/관리
OrderController	3개	주문 처리
PaymentController	2개	결제 처리
BoardController	7개	게시판 관리
CommentController	4개	댓글 관리
ReviewController	4개	리뷰 관리
AddressController	4개	배송지 관리
BannerController	4개	배너 관리
AdminController	3개	관리자 기능
LoginController	4개	OAuth2 로그인
AIProxyController	1개	AI 추천
합계	50개	

6.3 데이터베이스 분석

6.3.1 테이블 관계 분석

테이블	1:N 관계 수	N:1 관계 수
User	7개	0개
Item	8개	0개
Board	2개	1개

OrderItemList 1개 1개

6.3.2 인덱스 최적화

- user.email: UNIQUE 인덱스 (로그인 조회)
- item.category: 복합 인덱스 (카테고리 필터링)
- board.category + del_yn: 복합 인덱스 (게시판 조회)

6.4 보안 분석

6.4.1 인증/인가 체계

보안 항목	적용 기술	상태
비밀번호 암호화	BCrypt	적용
토큰 인증	JWT	적용
XSS 방지	HttpOnly Cookie	적용
CSRF 방지	Stateless 구조	적용
SQL Injection	JPA Parameterized Query	적용
CORS 설정	Spring Security	적용

7. 문제점 및 개선 제안

7.1 발생한 문제점 및 해결

7.1.1 TypeScript 빌드 오류

문제 상황

```
error TS2305: Module has no exported member 'BoardUpdateRequest'
error TS1484: 'ImgHTMLAttributes' must be imported using type-only import
```

원인 분석

- 사용하지 않는 타입 import 존재
- TypeScript strict 모드에서 type import 문법 요구

해결 방안

```
// Before
import { ImgHTMLAttributes } from 'react'

// After
import type { ImgHTMLAttributes } from 'react'
```

7.1.2 OrderListItem 타입 불일치

문제 상황

- 프론트엔드: snake_case (order_id, total_price)
- 백엔드: camelCase (orderId, totalPrice)

원인 분석

- 프론트엔드/백엔드 간 네이밍 컨벤션 불일치

해결 방안

- 프론트엔드 타입을 백엔드 응답에 맞춰 camelCase로 통일

7.1.3 찜 목록 전체 노출 버그

문제 상황

- 특정 사용자의 찜 목록 조회 시 모든 사용자의 찜이 노출

원인 분석

- Repository 쿼리에서 userId 필터링 누락

해결 방안

```
@Query("SELECT f.item FROM FavoriteItem f WHERE f.user.userId = :userId")
List<Item> findFavoriteItemsByUserId(@Param("userId") Long userId);
```

7.1.4 OAuth2 로그인 후 토큰 미저장

문제 상황

- 소셜 로그인 성공 후 JWT 토큰이 브라우저에 저장되지 않음

원인 분석

- OAuth2SuccessHandler에서 쿠키 설정 누락

해결 방안

```
ResponseCookie accessCookie = ResponseCookie.from("access_token", accessToken)
    .httpOnly(true)
    .secure(true)
    .sameSite("Lax")
    .path("/")
    .maxAge(3600)
    .build();
```

7.1.5 파일 업로드 용량 초과

문제 상황

- 10MB 이상 파일 업로드 시 500 에러 발생

해결 방안

```
spring:
  servlet:
    multipart:
      max-file-size: 10MB
      max-request-size: 50MB
```

7.1.6 CORS 에러

문제 상황

```
Access to XMLHttpRequest has been blocked by CORS policy
```

해결 방안

- SecurityConfig에 CORS 설정 추가
- 허용 Origin, Method, Header 명시

7.1.7 GitHub Actions 배포 권한 오류

문제 상황

- SCP 파일 전송 시 "Permission denied" 오류

원인 분석

- 배포 대상 서버의 사용자 홈 디렉토리 경로 하드코딩

해결 방안

- /home/ubuntu → ~/deploy 변경
- 동적 홈 디렉토리 경로 사용

7.1.8 Git 머지 충돌

문제 상황

- dev → main 머지 시 build.gradle, application.yaml 충돌

해결 방안

- 수동으로 충돌 해결

- dev 브랜치 내용 우선 적용

7.2 향후 개선 제안

7.2.1 성능 최적화

개선 항목	현재 상태	개선 방안	기대 효과
캐싱	미적용	Redis 캐시 도입	응답시간 50% 개선
쿼리 최적화	N+1 문제 존재	Fetch Join 적용	DB 부하 감소
이미지 최적화	원본 제공	CDN + WebP 변환	로딩 속도 개선

7.2.2 테스트 코드 강화

테스트 유형	현재 상태	목표	도구
단위 테스트	미작성	70% 커버리지	JUnit5, Mockito
통합 테스트	미작성	주요 API 테스트	Spring Boot Test
E2E 테스트	미작성	주요 시나리오	Cypress, Playwright

7.2.3 모니터링 시스템

항목	현재 상태	개선 방안
로그 수집	파일 로그	ELK Stack 도입
메트릭 수집	미적용	Prometheus + Grafana
알림	미적용	Slack/Discord 연동
APM	미적용	Pinpoint 또는 Scouter

7.2.4 기능 확장

기능	설명	우선순위
알림 시스템	주문 상태 변경 알림 (이메일/SMS)	상
검색 고도화	Elasticsearch 도입	중
추천 고도화	협업 필터링 알고리즘	중
다국어 지원	i18n 적용	하
모바일 앱	React Native 개발	하

8. 팀원 역할 분담

8.1 박지후 (팀장) - 프로젝트 총괄 + 백엔드 전체

8.1.1 담당 역할

- 프로젝트 전체 아키텍처 설계
- 팀원 업무 분배 및 일정 관리
- 코드 리뷰 및 품질 관리
- Git 브랜치 전략 수립 및 관리
- CI/CD 파이프라인 구축
- 팀원들이 구현하지 못한 백엔드 기능 전체 개발

8.1.2 구현 기능

기능	세부 내용
상품 관리	상품 CRUD, 검색/필터링, 페이지네이션
주문 시스템	주문 생성, 주문 내역 조회, 상태 관리
결제 연동	토스페이먼츠 API 연동, 결제/취소 처리
리뷰 시스템	리뷰 CRUD, 평점 계산
배송지 관리	배송지 CRUD, 기본 배송지 설정
배너 관리	배너 CRUD, 관리자 권한
관리자 기능	상품/회원/게시판 관리
파일 업로드	이미지 업로드, 검증, 저장
보안 설정	SecurityConfig, JWT 필터, CORS
CI/CD	GitHub Actions, 자동 배포

8.1.3 담당 파일

```

backend/
├── controller/
│   ├── ItemController.java
│   ├── OrderController.java
│   ├── PaymentController.java
│   ├── ReviewController.java
│   ├── AddressController.java
│   ├── BannerController.java
│   └── AdminController.java
├── service/
│   ├── ItemService.java
│   ├── OrderService.java
│   ├── ReviewService.java
│   ├── AddressService.java
│   ├── BannerService.java
│   └── AdminService.java
├── config/
│   └── SecurityConfig.java
└── .github/
    └── workflows/
        └── deploy.yml

```

8.2 이진용 - SNS 로그인 (Google, Naver OAuth2)

8.2.1 담당 역할

- OAuth2 소셜 로그인 시스템 설계 및 구현
- Google, Naver 로그인 연동
- OAuth2 인증 성공/실패 핸들러 개발
- 소셜 로그인 사용자 정보 매핑

8.2.2 구현 기능

기능	세부 내용
Google 로그인	OAuth2 클라이언트 등록, 콜백 처리
Naver 로그인	OAuth2 클라이언트 등록, 콜백 처리
인증 핸들러	로그인 성공/실패 처리, JWT 발급
사용자 매핑	Provider별 사용자 정보 추출 및 저장
쿠키 설정	HttpOnly 쿠키로 토큰 저장

8.2.3 담당 파일

```

backend/
├── oauth/
│   └── OAuth2SuccessHandler.java
├── controller/
│   └── LoginController.java
└── resources/
    └── application.yaml (OAuth2 설정)

```

8.2.4 구현 상세

```

# application.yaml OAuth2 설정
spring:
  security:
    oauth2:
      client:
        registration:
          google:
            client-id: ${GOOGLE_CLIENT_ID}
            client-secret: ${GOOGLE_CLIENT_SECRET}
            scope: profile, email
          naver:
            client-id: ${NAVER_CLIENT_ID}
            client-secret: ${NAVER_CLIENT_SECRET}
            scope: name, email, profile_image

```

8.3 오종혁 - 로컬 로그인 + 게시판/댓글 시스템

8.3.1 담당 역할

- 이메일/비밀번호 기반 인증 시스템 구현
- JWT 토큰 발급 및 관리
- 게시판 CRUD 기능 개발
- 댓글 시스템 개발
- 파일 첨부 기능 구현

8.3.2 구현 기능

기능	세부 내용
회원가입	이메일 중복 확인, 비밀번호 암호화
로그인	JWT 발급, 쿠키 저장
토큰 관리	Access/Refresh 토큰, 자동 갱신
게시판	게시글 CRUD, 카테고리, 비밀번호
파일 첨부	다중 파일 업로드, 다운로드
댓글	댓글 CRUD, 비밀 댓글

8.3.3 담당 파일

```
backend/
├── controller/
│   ├── AuthController.java
│   ├── BoardController.java
│   └── CommentController.java
├── service/
│   ├── AuthService.java
│   ├── BoardServiceImpl.java
│   └── CommentService.java
├── security/
│   ├── JwtTokenProvider.java
│   └── JwtAuthenticationFilter.java
└── entity/
    ├── board/
    │   ├── Board.java
    │   └── BoardFile.java
    ├── comment/
    │   └── Comment.java
```

8.4 이나라 - 프론트엔드 UI/UX + AI 연동

8.4.1 담당 역할

- 전체 프론트엔드 UI/UX 설계 및 구현
- React 컴포넌트 개발
- Material-UI 기반 디자인 시스템 구축
- ChatGPT API 연동 AI 추천 기능 개발
- 고객 지원 채팅 위젯 개발

8.4.2 구현 기능

기능	세부 내용
페이지 개발	24개 페이지 (메인, 상품, 주문, 마이페이지, 관리자)
컴포넌트	Header, Footer, ProductCard, CategoryFilter 등
Custom Hooks	useAuth, useCart, useLike, useAiRecommend
AI 추천	ChatGPT 연동, 키워드 추출, 상품 매칭
채팅 위젯	Glassmorphism UI, FAQ 시스템
애니메이션	GSAP 기반 UI 애니메이션

8.4.3 담당 파일

```

frontend/
├── src/
│   ├── pages/           # 24개 페이지
│   ├── components/      # 15개 컴포넌트
│   ├── hooks/
│   │   ├── useAuth.ts
│   │   ├── useCart.ts
│   │   ├── useLike.ts
│   │   ├── useAiRecommend.ts
│   │   └── useDaumPostcode.ts
│   ├── api/             # API 클라이언트
│   └── theme/            # 디자인 토큰
backend/
├── controller/
│   └── AiProxyController.java
└── service/
    └── AiProxyService.java

```

8.4.4 페이지 목록

카테고리	페이지	설명
메인	HomePage	메인 페이지 (배너, 상품 섹션)
상품	ProductListPage	상품 목록 (필터, 검색)
상품	ProductDetailPage	상품 상세 (이미지, 옵션, 리뷰)
상품	WishlistPage	찜 목록
장바구니	CartPage	장바구니
주문	OrderPage	주문/결제
주문	OrderSuccessPage	주문 성공
주문	OrderFailPage	주문 실패
인증	LoginPage	로그인 (로컬 + OAuth2)
인증	SignupPage	회원가입
게시판	BoardListPage	게시판 목록
게시판	BoardWritePage	게시글 작성
게시판	BoardDetailPage	게시글 상세
게시판	BoardEditPage	게시글 수정
마이페이지	MyProfilePage	내 프로필
마이페이지	MyOrdersPage	주문 내역
마이페이지	MyPostsPage	내 게시글
관리자	AdminDashboardPage	관리자 대시보드
관리자	AdminProductListPage	상품 관리
관리자	AdminBoardPage	게시판 관리
관리자	AdminBannerPage	배너 관리
관리자	AdminMemberListPage	회원 관리

8.5 이현우 - 데이터베이스 설계

8.5.1 담당 역할

- 데이터베이스 스키마 설계
- Entity 클래스 작성
- JPA 연관관계 매핑
- ERD 작성
- 데이터 무결성 제약조건 설계

8.5.2 구현 기능

기능 **세부 내용**
 Entity 설계 27개 Entity 클래스 작성
 관계 매핑 1:N, N:1, N:M 관계 설정
 Enum 정의 카테고리, 색상, 사이즈 등
 제약조건 PK, FK, UNIQUE, NOT NULL
 설계 패턴 Soft Delete, Cascade, Timestamp

8.5.3 담당 파일


```

backend/src/main/java/com/example/backend/entity/
├── user/
│   ├── User.java
│   └── Role.java
├── item/
│   ├── Item.java
│   ├── details/
│   │   ├── ItemImage.java
│   │   ├── Color.java
│   │   └── Size.java
│   ├── utility/
│   │   ├── CartItem.java
│   │   ├── FavoriteItem.java
│   │   ├── OrderItem.java
│   │   ├── OrderItemList.java
│   │   └── ViewedItem.java
│   └── enums/
│       ├── MajorCategoryEnum.java
│       ├── MiddleCategoryEnum.java
│       ├── SubcategoryEnum.java
│       ├── ColorEnum.java
│       └── SizeEnum.java
├── board/
│   ├── Board.java
│   └── BoardFile.java
├── comment/
│   └── Comment.java
├── review/
│   └── Review.java
├── banner/
│   └── Banner.java
└── utility/
    └── BaseEntity.java

```

8.5.4 설계 패턴 적용

```

// Soft Delete 패턴
@Column(name = "del_yn", length = 1)
private String delYn = "N";

// Timestamp 자동 관리
@CreationTimestamp
private LocalDateTime createdAt;

@UpdateTimestamp
private LocalDateTime updatedAt;

// Cascade 삭제
@OneToMany(mappedBy = "item", cascade = CascadeType.ALL, orphanRemoval = true)
private List<Review> reviews;

```

9. 결론

9.1 프로젝트 성과

본 프로젝트를 통해 Spring Boot와 React를 활용한 풀스택 이커머스 플랫폼을 성공적으로 구축하였습니다.

9.1.1 기술적 성과

- 인증 시스템: JWT 기반 로컬 로그인 + OAuth2 소셜 로그인 구현
- 결제 시스템: 토스페이먼츠 API 연동을 통한 실결제 처리
- AI 추천: ChatGPT API를 활용한 상품 추천 시스템 구축
- 자동화 배포: GitHub Actions를 통한 CI/CD 파이프라인 구축

9.1.2 협업 성과

- Git 브랜치 전략 (dev → main) 적용

- Pull Request 기반 코드 리뷰 프로세스 확립
- 역할 분담을 통한 효율적인 병렬 개발

9.2 학습 성과

- Spring Security + JWT를 활용한 인증/인가 구현 경험
- OAuth2 프로토콜 이해 및 소셜 로그인 연동 경험
- JPA를 활용한 ORM 기반 데이터베이스 설계 및 구현
- React + TypeScript를 활용한 SPA 개발 경험
- CI/CD 파이프라인 구축 및 자동화 배포 경험
- 팀 협업을 통한 Git 워크플로우 경험

9.3 향후 발전 방향

1. 테스트 코드 작성: 단위/통합/E2E 테스트 도입
2. 성능 최적화: Redis 캐싱, 쿼리 최적화
3. 모니터링 시스템: 로그 수집, 메트릭 모니터링
4. 기능 확장: 알림 시스템, 검색 고도화, 모바일 앱

작성일: 2025년 12월

팀명: MyShop Team

소속: KD 아카데미 컴퓨터학원

© 2024-2025 MyShop Team. All rights reserved.