



TEAM06: Drop the Git

커피빈 주문 관리 시스템

스프링 MVC 기반 웹 주문 시스템 개발 프로젝트를 소개합니다.

팀원 소개



이강현 (팀장)

- 사용자 장바구니
- 실시간 수량 변경 기능



김도윤

- 회원가입 및 로그인 구현
- Spring Security 인가 처리



안세희

- 주문 내역 조회
- 주문 취소 프로세스



이초롱

- 주문 페이지 제작
- 주문 및 주문 확인



황영준

- 관리자 화면 재고 관리
- 전체 주문 관리 대시보드

프로젝트 소개

커피빈 주문 관리 시스템

Spring MVC 기반
웹 주문 시스템입니다.

주문 프로세스

- 회원가입 및 로그인
- 상품 선택
- 주문정보 입력
- 결제 및 주문 완료

회원

- 주문 내역 조회
- 주문 취소
- 장바구니

관리자

- 재고 관리
- 회원의 주문 내역 조회
- 회원의 주문 취소

기술 스택



Backend

Java 21, Spring MVC,
MyBatis



Database

MySQL 8.x



View

JSP, JSTL,
Materialize, CSS



개발 환경

Maven, IntelliJ IDEA,
Git, GitHub

협업 과정

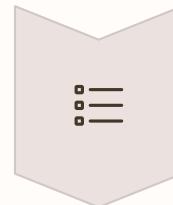
Trello

zep





주요 기능



회원가입 및 로그인



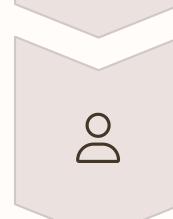
상품 선택 및 주문 정보 입력



장바구니 또는 결제(주문)

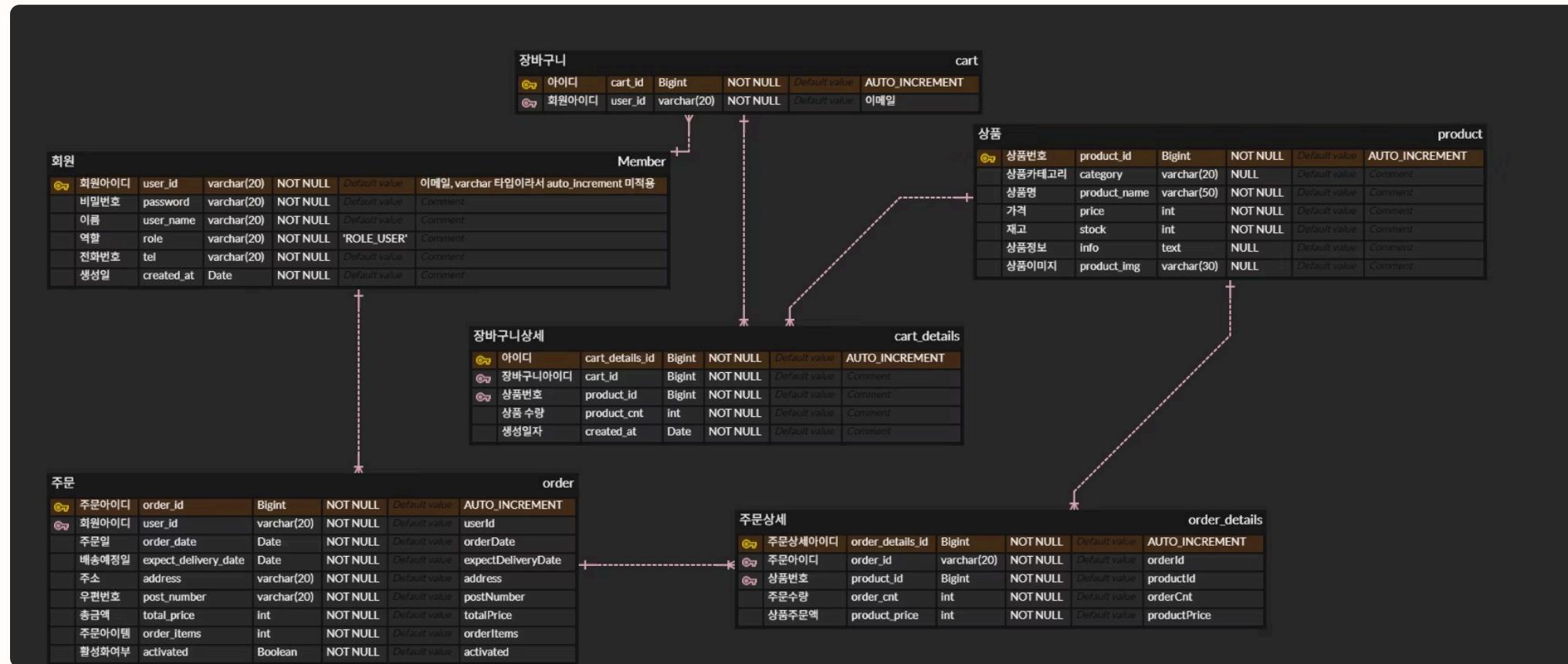


고객의 주문상품 조회 및 주문 취소



관리자의 재고 관리 및 고객 주문 관리

ERD 및 DB 설계



ERD 및 테이블 구조

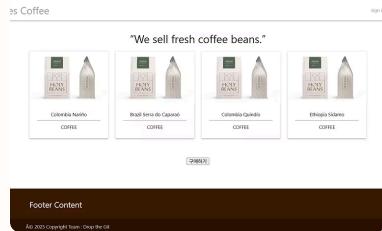
테이블명	주요 컬럼	관계	설명
회원	회원ID(PK)	1:N → 주문	회원가입 시 입력 정보 저장
주문	주문ID(PK), 회원ID(FK)	N:1 ← 회원 1:N → 주문상세	회원별 주문 정보 관리
주문상세	주문상세ID(PK), 주문ID(FK)	N:1 ← 주문	주문 시점의 상품 정보와 수량 저장
상품	상품ID(PK)	참조됨	상품 정보(이름, 가격) 관리
장바구니	장바구니ID(PK), 회원ID(FK)	N:1 ← 회원	결제 전 임시 보관 정보

각 테이블은 비즈니스 로직에 맞게 연결되어 있습니다. 주문과 상품은 주문상세 테이블을 통해 연결됩니다.

API 명세서

구분	기능명	메서드	경로	설명
메인	메인화면	GET	/	웹사이트 첫 페이지 로드
로그인/회원가입	로그인	GET/POST	member/signin	사용자 인증 처리
로그인/회원가입	회원가입	GET/POST	member/signup	신규 사용자 등록
로그인/회원가입	회원 확인	GET	api/member/exists/{id}	아이디 중복 확인
주문	주문 화면	GET	order	상품 주문 화면
주문	주문	GET/POST	order/order	상품 주문 처리
장바구니	장바구니 조회 및 수량 조정	GET/POST	cartList	사용자 장바구니 목록 표시 및
장바구니	상품 삭제	POST	cartList/delete	장바구니에서 상품 제거
장바구니 주문	주문 결과	GET/POST	order/cartOrderComplete	장바구니 상품 주문 완료
회원	주문 내역	GET	member/mypage	사용자 주문 이력 조회
회원	주문 취소	DELETE	member/order/{orderId}/cancel	주문 취소 요청 처리
회원	주문 취소 결과	GET	member/order/cancel/result	취소 처리 결과 화면
관리자	관리자 회원가입	GET/POST	admin/signup	관리자 계정 등록
관리자	대시보드	GET	admin/dashboard	관리자 페이지 메인화면
관리자	주문취소	DELETE	admin/orderList	관리자 권한 주문 취소

주요 화면 (UI)



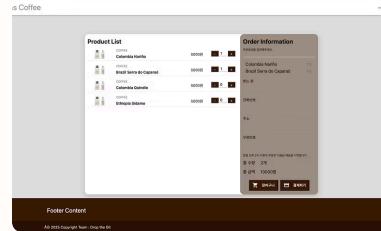
메인 화면

상품 목록을 확인할 수 있고,
상품 주문, 또는 회원가입 / 로그인 페이지로
연결됩니다.



로그인 및 회원가입

이메일, 비밀번호, 이름, 전화번호를 작성하여
회원가입을 하고 로그인할 수 있습니다.



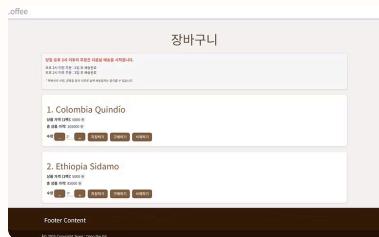
상품 주문 페이지

주문하고자 하는 상품의 수량을 조절하고,
수령 정보를 입력하여 상품을
장바구니에 담거나 주문할 수 있습니다.



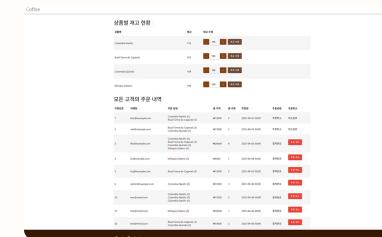
주문 완료 페이지

주문 번호, 주문일자, 상품 목록,
총 결제 금액을 표시합니다.



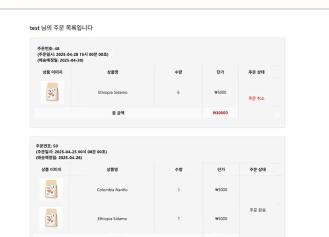
장바구니

주문하고자 하는 상품들을 담아둔다.



관리자 페이지

재고를 관리하고, 모든 고객의 주문을
확인 및 취소할 수 있습니다.



주문내역 조회 및 취소

주문내역을 조회하고,
주문을 취소할 수 있습니다.

주요 코드 흐름

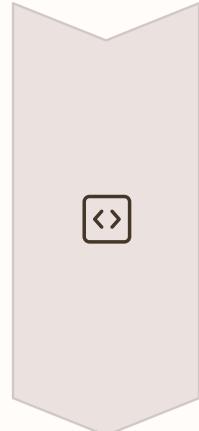


팀원별 트러블슈팅 & 해결 과정

프로젝트 진행 중 발생한 다양한 문제들과 해결 방법을 정리했습니다.

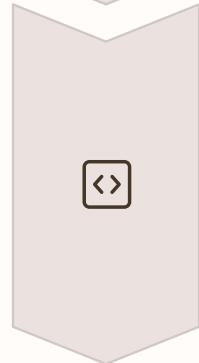
팀원	발생한 문제	해결 방법
도윤	<ul style="list-style-type: none">Dto에 name이 null로 반환	<ul style="list-style-type: none">dto 변수명을 테이블 열과 일치시킴(productName으로 변경)
세희	<ul style="list-style-type: none">resultMap 설정 오류로 테이블 구조 꼬임주문 취소 시 GET 매팅과 리다이렉트 경로 설정 오류로 500 에러 다수 발생	<ul style="list-style-type: none">Dto 필드명을 정확히 매팅하고 동적으로 값 할당되도록 수정한 개씩 값을 바꿔가며 논리적 흐름에 맞게 경로 설정 조정
강현	<ul style="list-style-type: none">로그인 후 user_id null값 반환장바구니→결제 화면 데이터 전달 실패	<ul style="list-style-type: none">커스텀 Principal에서 Authentication으로 변경RedirectAttributes 대신 ModelAttribute로 데이터 전달 방식 수정
영준	<ul style="list-style-type: none">DTO와 column 이름 불일치로 매팅 실패	<ul style="list-style-type: none">resultMap에 Dto 입력해 오류 방지하고 수동 매팅
초롱	<ul style="list-style-type: none">Mapper.xml과 Mapper.java 혼용OrderDetails 저장 시 product_name null 발생order_id null 저장으로 DB 제약조건 에러	<ul style="list-style-type: none">Mapper.xml만 사용하여 Repository 연결ProductDto→OrderDetailsDto 변환 시 null 체크 강화Order 저장 후 생성된 order_id를 OrderDetails에 매팅

느낀점 및 개선 방향



느낀점

- 실사용 흐름을 의식한 설계가 중요함
- DB 제약조건과 연동 로직의 사전 설계 필요성
- Controller → Service → Repository 흐름 분리의 중요성
- 깃 브랜치 전략의 중요성성



개선 방향

- 상품 재고(stock) 차감 로직 추가
- 주문 완료 후 이메일 발송 기능
- JPA/Hibernate를 활용한 버전 개선