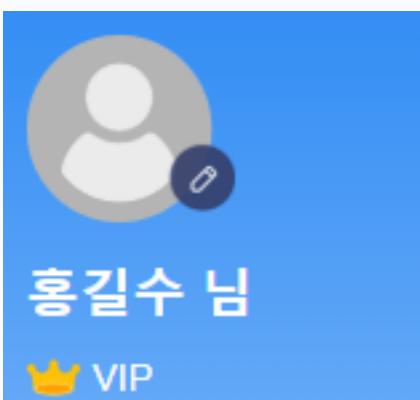


마이페이지

 / 마이 / 쇼핑내역 / 주문배송목록 / 주문정보 / 배송정보

-페이지 계층 구조를 시각적으로 표현, 이동 가능
추가



- 핵심 개인 정보를 한눈에 확인 가능
 - 이름 클릭시 수정 페이지로 이동 가능



- 관심 도서와 등록된 배송지
개수를 실시간으로 표시

```
<div class="table">
    <c:forEach var="wish" begin="0" end="3" items="${wishlistDtos }">
        <div>
            
            <p>${wish.author }<br>${wish.title }</p>
        </div>
    </c:forEach>
</div>
```

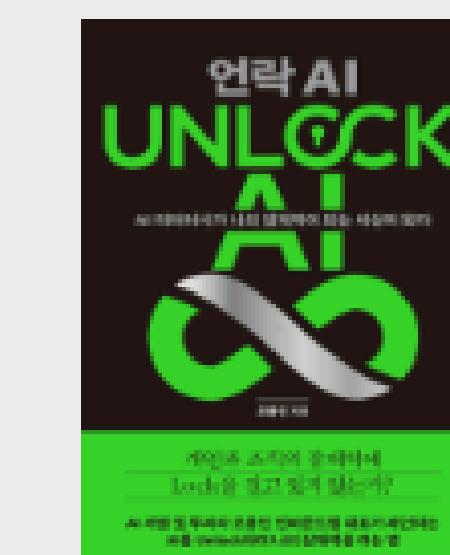
-각 도서 클릭시 도서 상세페이지로 이동

The screenshot shows a mobile application interface with a blue header bar. The top left corner displays the text '내 도서 / 마이' (My Books). Below the header is a user profile section with a placeholder icon, the name '홍길수 님', and a 'VIP' status indicator. To the right of the profile are two buttons: a green rounded rectangle labeled '주전 도서 리스트' (Main Document List) and a dark blue rounded rectangle labeled '인기 도서 리스트' (Popular Document List). The main content area features three document cards. The first card, titled '소년미 온다' (Coming of Age), shows a QR code, a small thumbnail image of a building, and the text '10% 13,500원'. The second card, titled '작별하지 않는다' (Does Not Say Goodbye), shows a thumbnail image of a person, and the text '10% 15,120원'. The third card, titled '채식주의자' (Vegetarian), shows a thumbnail image of a person, and the text '10% 13,500원'. Each card has a light gray background with a subtle shadow effect.

- 쇼핑내역
- 주문/배송목록
- 선물함
- 매장 구매 내역
- 영수증 조회/후 적립

- 회원정보
- 회원정보 수정
- 로그인 설정
- 마케팅 수신 설정
- 배송 주소록

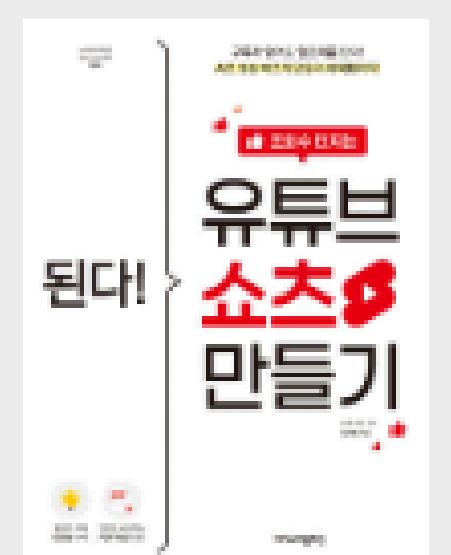
예시리스트



조용민
UNLOCK AI 언락



온진주
모든 게 자연인 미련



최지영



암투안 드 생트리에리
어린 여자

위시리스트로 이동



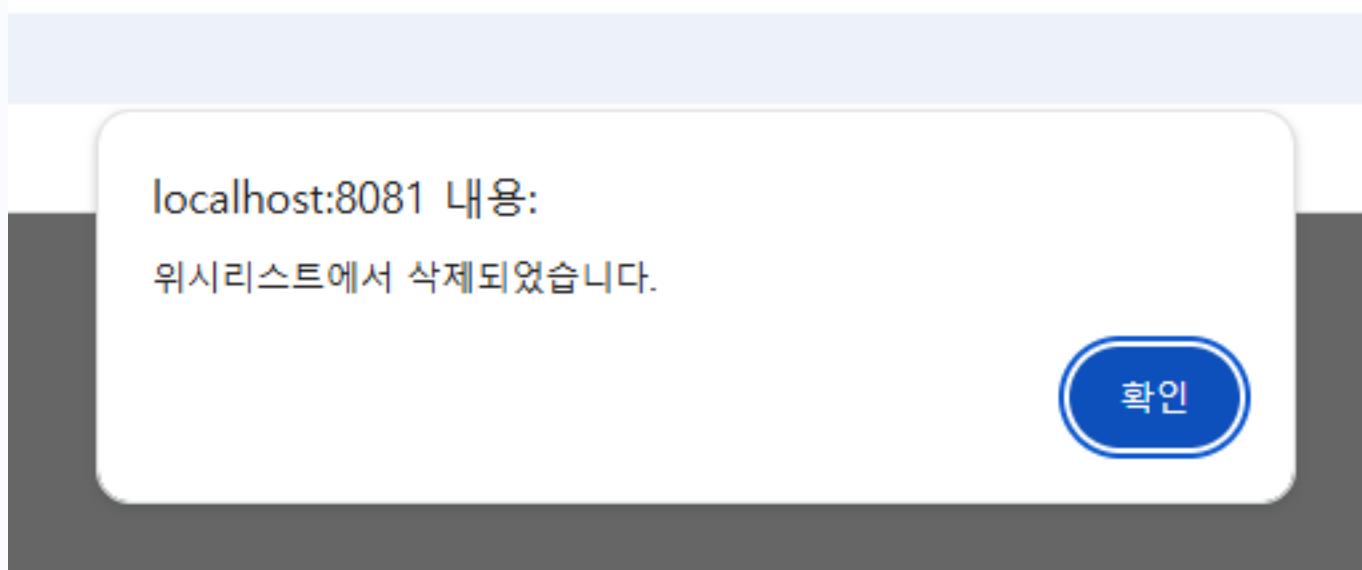
위시리스트

-사용자가 지정한 카테고리 기준으로 필터링하여 확인할 수 있는 기능

```
<form action="/00BookJSP/08wishlist/wishlistDB.jsp" method="get">
<input type="hidden" name="book_id" value="${wish.book_id}">
<button type="submit" class="detail-btn"></button>

wishlistDao.deleteWishList(user_id, book_id);
session.setAttribute("wishAdded", false);
```

-위시리스트에 등록된 도서 하단에 ❤ 아이콘을 배치하여, 클릭 시 해당 도서를 위시리스트에서 즉시 삭제



-위의 알림창이 뜨고 위시리스트 목록에서 삭제

위시리스트

전체 소설 경제 만화 IT/프로그래밍 인문

홍길수 님 VIP 평내역

마이 라이브러리 위시리스트

- ▶ 등록
- ▶ 편집
- ▶ 삭제
- ▶ 회원정보
- ▶ 회원정보 수정
- ▶ 로그인 설정
- ▶ 마케팅 수신 설정
- ▶ 배송 주소록
- ▶ 라이브러리
- ▶ 메인
- ▶ 리스트
- ▶ 코멘트
- ▶ 문장수집
- ▶ 위시리스트
- ▶ 구독
- ▶ e-라이브러리
- ▶ 문의내역
- ▶ 1:1 문의

The Sentence of Death ANTHONY HODGESWELL 앤서니 호로비츠 슬겨진 전 죽음

LONG SHADOWS 기억을 되살리는 남자

모든 게 처음인 브랜드의 무기들

어린왕자

된다! 유튜브 쇼츠 만들기

된다! 최지영 된다! 조희수 터지는 ...

조용민 UNLOCK AI 언락 AI

→ 원하는 메뉴로 이동

배송 주소록 - 추가기능

- 전체 배송지 데이터 확인이 가능
- 주소 입력 후 등록 버튼 클릭시 DB 저장과 동시에 목록에 행이 즉시 추가되는 기능 구현

```

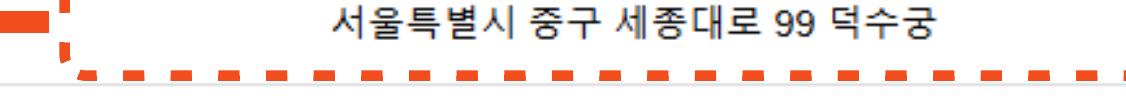
request.setCharacterEncoding("UTF-8");
response.setContentType("text/html; charset=UTF-8");

String user_id = (String) session.getAttribute("user_id");
String address = request.getParameter("address");

ShipDto dto = new ShipDto(user_id, address);
ShipDao.insert(dto);

```

배송지 목록

목록	주소	삭제	수정
1	서울특별시 강남구 테헤란로 123	<button>삭제</button>	<button>수정</button>
2	서울특별시 광진구 자양동1-2	<button>삭제</button>	<button>수정</button>
3	서울특별시 종로구 사직로 161	<button>삭제</button>	<button>수정</button>
+ 	서울특별시 중구 세종대로 99 덕수궁	 	
목록	주소	삭제	수정
1	서울특별시 강남구 테헤란로 123	<button>삭제</button>	<button>수정</button>
2	서울특별시 광진구 자양동1-2	<button>삭제</button>	<button>수정</button>
3	서울특별시 종로구 사직로 161	<button>삭제</button>	<button>수정</button>
4	서울특별시 중구 세종대로 99 덕수궁	  	
+ 	주소 입력		

배송 주소록 - 수정기능

- '수정' 버튼 클릭 시 주소 수정 페이지로 이동
- 수정 후 저장 시 배송지 목록에 변경 사항 실시간 반영 가능 구현

```

long ship_id = (Long) session.getAttribute("ship_id");
String address = request.getParameter("address");

ShipDao.update(ship_id, address);

response.sendRedirect("shiplist.jsp");
    
```

배송지 목록			
목록	주소	삭제	수정
1	서울특별시 강남구 테헤란로 123	[삭제]	[수정]
2	서울특별시 광진구 자양동1-2	[삭제]	[수정]
3	서울특별시 종로구 사직로 161	[삭제]	[수정]
4	서울특별시 종구 정동 5-1	[삭제]	[수정]
+	주소 입력		[추가]

배송지 목록

목록	주소	삭제	수정
1	서울특별시 강남구 테헤란로 123	[삭제]	[수정]
2	서울특별시 광진구 자양동1-2	[삭제]	[수정]
3	서울특별시 종로구 사직로 161	[삭제]	[수정]
4	서울특별시 종구 세종대로 99 덕수궁	[삭제]	[수정]
+	주소 입력		[추가]

수정 누르면 '주소 수정하기' 페이지로 이동

주소 수정하기

목록	주소
주소	서울특별시 종구 세종대로 99 덕수궁
변경	서울특별시 종구 정동 5-1



배송 주소록 - 삭제기능

-주소 목록에서 '삭제' 버튼 클릭 시, 해당 주소의 고유 식별자(ID)가 서버로 전송되어 해당 데이터를 삭제하고, 삭제 결과가 테이블에 즉시 반영되는 처리 기능

배송지 목록

목록	주소	삭제	수정
1	서울특별시 강남구 테헤란로 123	삭제	수정
2	서울특별시 광진구 자양동1-2	삭제	수정
3	서울특별시 종로구 사직로 161	삭제	수정
+		주소 입력	
		추가	

배송지 목록

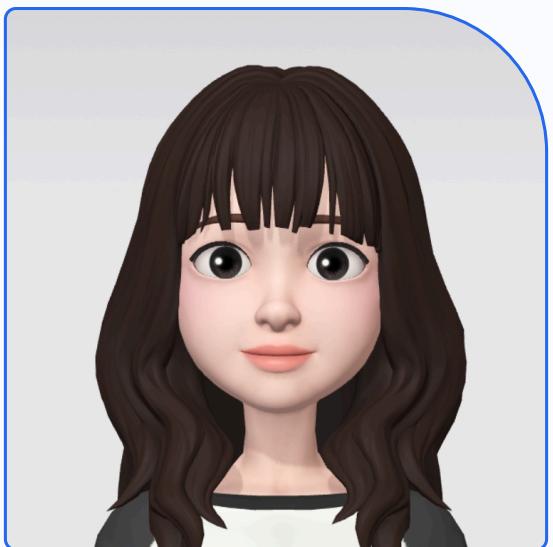
목록	주소	삭제	수정
1	서울특별시 강남구 테헤란로 123	삭제	수정
2	서울특별시 광진구 자양동1-2	삭제	수정
3	서울특별시 종로구 사직로 161	삭제	수정
4	서울특별시 종로구 정동 5-1	삭제	수정
+		주소 입력	
		추가	

```
public static void delete(long ship_id) {
    String sql = String.format("delete from ship where ship_id=%d",
        ship_id);
    DBConn.statementUpdate(sql);
}
```

```
long ship_id = Long.parseLong(request.getParameter("ship_id"));
ShipDao.delete(ship_id);
response.sendRedirect("shiplist.jsp");
```



자체 피드백



박채원

- 도서 메타데이터 구축
- 배송지 CRUD 기능
- 배송 정보/진행 연동 기능
- 주문 배송 연동 기능
- 주문번호 운송장 자동 생성 기능
- 마이페이지(사용자 정보, 주문, 위시리스트) 통합 구현 기능
- 관심 도서 목록 UI

좋았던 점

팀 분위기가 좋아서 서로 자유롭게 의견을 나눌 수 있었습니다. 부족한 부분이나 모르는 점이 생기면 서로 도와주면서 함께 해결해 나갔고, 그 과정에서 많은 것을 새롭게 배울 수 있었습니다. 개선하거나 변경이 필요하다고 느끼는 부분이 있을 때는 팀원 모두가 적극적으로 의견을 제시했고, 그 중 가장 합리적인 방향을 함께 찾아가는 과정이 재미있고 의미있었습니다.

어려웠던 점

주문한 책 정보를 기반으로 주문번호와 운송장번호를 생성하는 과정에서, 중복되지 않는 고유한 번호를 만드는 방식에 대한 고민이 많았고, 생성된 번호를 활용해 주문상세 페이지를 구성할 때는 도서 수량만큼 정보가 중복 출력되는 문제가 발생했습니다. 또한 장바구니에서 배송지를 선택한 후, 해당 배송 정보와 주문 도서 정보를 동시에 전송하는 과정에서는 form 태그 중첩으로 인해 데이터 전달 오류가 발생했고, 배송지 테이블에서 삭제, 수정, 추가 기능을 하나의 테이블에 통합하여 구현하면서 각각의 정보를 DB로 정확히 전달하는 데에도 어려움이 있었습니다.

개선할 부분

우선, 기본 프로필 이미지 대신 사용자가 원하는 이미지를 업로드해 저장할 수 있는 기능을 추가하고 싶습니다. 배송지 입력은 주소 검색API를 연동해 도로명 주소를 쉽게 검색, 선택할 수 있도록 개선하고 싶고, 마이페이지의 추천 도서는 모든 사용자에게 동일하게 제공되는데, 개인 취향이나 구매 이력 기반으로 개인화된 추천이 가능하도록 개선하고 싶습니다. 장바구니에서도 배송지를 등록할 수 있는 기능을 추가하고, 누적 구매 금액에 따라 회원등급을 나눈 시스템도 도입해 다양한 혜택을 제공할 수 있도록 개선할 계획입니다.



장바구니

- 체크박스 선택 유무로 주문 및 항목 삭제 가능
- 총가격 버튼 누를 시 선택된 항목들 가격 합산 가능
- 주문하면 주문완료 페이지로 넘어가면서 주문번호, 주문시간, 주문목록, 총가격을 띄움

```

Long totalPrice = 4000L; // 기본배송비 설정
String[] Ids = request.getParameterValues("cart_id[]");
request.setAttribute("totalPrice", totalPrice);
if (Ids != null) { // 체크박스가 선택되었을 때만 실행
    for (String cart_id : Ids) {
        try {
            int cartIdInt = Integer.parseInt(cart_id); // 숫자로 변환 (안전성 확보)
            String sql = "select amount, price FROM cart, books" +
                " where cart.book_id = books.book_id and cart_id = " + cartIdInt;
            ResultSet rs = DBConn.createStatementQuery(sql);

            if (rs.next()) {
                // 결과가 있을 경우에만 처리
                Long amount = rs.getLong("amount");
                Long price = rs.getLong("price");
                totalPrice = totalPrice+(amount * price);
            }
        }
    }
}

```

"주문이 정상적으로 완료되었습니다."

장바구니

번호	제목	수량	가격
<input checked="" type="checkbox"/> 1	행복은 먹고 자고 기다리고 5	1 <button>수정</button>	7,200원
<input checked="" type="checkbox"/> 2	마루는 강쥐 2	2 <button>수정</button>	14,400원
<input checked="" type="checkbox"/> 3	커서 AI	1 <button>수정</button>	22,500원

배송지선택

서울특별시 종로구 우정국로

주문하기 **선택 항목 삭제**

상품 추가하기

총가격 62,500원

주문 번호	20250410000184
주문 시간	2025-04-10 12:03:00
주문 목록	
행복은 먹고 자고 기다리고 5	1
마루는 강쥐 2	2
커서 AI	1
총 가격 : 62500 원	

※ 주문 내역 및 배송에 관한 안내는 **주문/배송목록**을 통하여 확인 가능합니다.

→ 주문목록으로 이동



장바구니

-주문을 완료하면 기존에 장바구니에서는 사라짐

장바구니			
번호	제목	수량	가격
			총가격 원

배송지선택

서울시 강남구 테헤란로 123

```
// 주문전송 눌렀을때 장바구니에서는 삭제
String[] deleteId = request.getParameterValues("cart_id[]");
if(deleteId != null){
    List<Long> deleteIdList = new ArrayList<>();
    for(String id : deleteId){
        deleteIdList.add(Long.parseLong(id));
    }

    CartBookDao cartBookDao = new CartBookDao();
    boolean deleteCartId = cartBookDao.delete(deleteIdList);
}

response.sendRedirect("../05orders/ordersSelectId.jsp");
```

```
public boolean delete(List<Long> deleteId) {
    try {
        for(Long cart_id : deleteId){
            String sql = "delete from cart where cart_id = " + cart_id;
            int result = DBConn.createStatementUpdate(sql);
        }
    }
```



주문목록

- 주문번호, 주문시간이 나오며 삭제 기능 구현
- 주문번호를 누르면 주문상세정보가 뜸
- 주문상품과 총 주문가격 확인 가능

회원 가입 시 첫구매 100원 이벤트

 Rocket bookstore

검색어를 입력하세요

홍길수 님 [로그아웃](#) | [주문배송](#) | [장바구니](#) | [고객센터](#)

[전체](#) [소설](#) [경제](#) [만화](#) [IT/프로그래밍](#) [인문](#)

[▶/ 마이 / 쇼핑내역 / 주문배송목록 / 주문정보](#)

주문정보

주문번호 : 20250408000125
운송장번호 : RK20250408000125
배송지
서울시 강남구 테헤란로 123

주문상품

어린 왕자	2
꿈빛 파티시엘 신장판 3	1
한 권으로 끝내는 회계와 재무제표	2
총 가격 : 73000 원	

쇼핑내역

- 주문/배송목록
- 선물함
- 매장 구매 내역
- 영수증 조회/후 적립

회원정보

- 회원정보 수정
- 로그인 설정
- 마케팅 수신 설정

회원 가입 시 첫구매 100원 이벤트

 Rocket bookstore

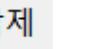
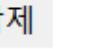
검색어를 입력하세요

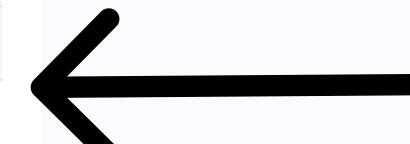
홍길수 님 [로그아웃](#) | [주문배송](#) | [장바구니](#) | [고객센터](#)

[▶/ 마이 / 쇼핑내역 / 주문배송목록 / 주문정보](#)

주문목록

목록	주문번호	주문시간	삭제
1	20250408000125	2025-04-08	
2	20250407000122	2025-04-07	
3	20250407000102	2025-04-07	
4	20250331000062	2025-03-31	
5	20250331000061	2025-03-31	
6	20250331000060	2025-03-31	





배송정보 및 현황

- 주문배송목록에 들어가면 배송상황과 기본 정보가 뜸
- 관리자가 배송현황을 바꿔주면 사용자 화면에서도 바뀜

배송정보

등기 및 소포 우편물의 배송정보를 조회할 수 있습니다. (배송 문의 가능 시간 : 18시까지)

본 서비스인 배송조회는 우편물의 배송상태를 고객이 신속히 확인할 수 있도록 제공하는 것이 목적으로 모든 배송정보가 표시되지 않을 수 있습니다.
우편물과 관련된 법적 증거 자료가 필요한 경우, 로켓법 시행규칙 제25조 제1항 제4호 다목 및 제59조에 따른 배달증명서비스를 이용해 주시기 바랍니다.
로켓서점의 배송은 자사가 직접 운영하며, 주문 즉시 상품을 준비하고 배송을 진행하오니 이 점 참고하여 주시기 바랍니다.
※ 주문 취소 및 배송지 변경이 어려우므로 신중하게 주문해 주시기 바라며, 자세한 사항은 고객센터로 문의해 주세요.
(모니터 뒤에 사람이 있으니, 욕설은 지양해주세요^^)

❶ 배송진행 상황



상품준비중
배송준비중
배송중
배송완료

배송상태				
상품 접수 완료				

❷ 기본 정보

운송장번호	보내는 분/접수일자	받는분	수령인/배달일자	배송현황
RK20250408000180	주식회사:로켓서점 20250408	홍길동	홍길동 20250408	상품 접수 완료

배송정보

등기 및 소포 우편물의 배송정보를 조회할 수 있습니다. (배송 문의 가능 시간 : 18시까지)

본 서비스인 배송조회는 우편물의 배송상태를 고객이 신속히 확인할 수 있도록 제공하는 것이 목적으로 모든 배송정보가 표시되지 않을 수 있습니다.
우편물과 관련된 법적 증거 자료가 필요한 경우, 로켓법 시행규칙 제25조 제1항 제4호 다목 및 제59조에 따른 배달증명서비스를 이용해 주시기 바랍니다.
로켓서점의 배송은 자사가 직접 운영하며, 주문 즉시 상품을 준비하고 배송을 진행하오니 이 점 참고하여 주시기 바랍니다.
※ 주문 취소 및 배송지 변경이 어려우므로 신중하게 주문해 주시기 바라며, 자세한 사항은 고객센터로 문의해 주세요.
(모니터 뒤에 사람이 있으니, 욕설은 지양해주세요^^)

❶ 배송진행 상황



상품준비중
배송준비중
배송중
배송완료

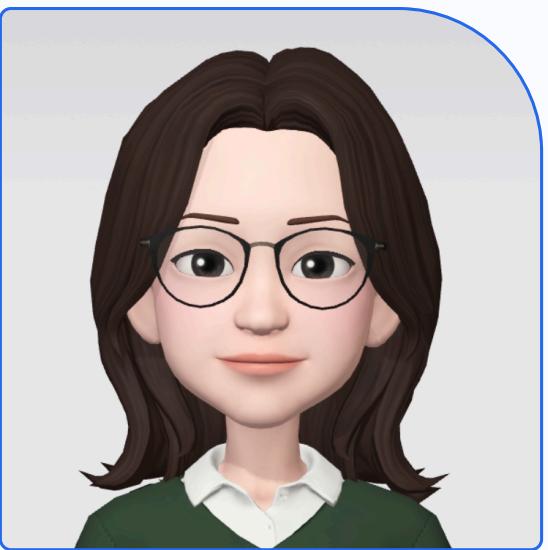
배송상태				
배송 준비 중				

❷ 기본 정보

운송장번호	보내는 분/접수일자	받는분	수령인/배달일자	배송현황
RK20250408000180	주식회사:로켓서점 20250408	홍길동	홍길동 20250408	배송 준비 중

Rocket Bookstore

자체 피드백



선승희

- 장바구니 항목 삭제 및 주문 기능
- 총 금액 계산 로직 구현
- 카테고리 페이지 반응형 웹 적용
- 카테고리 페이지네이션
- 주문목록 삭제 기능
- 로고 디자인

좋았던 점

이번 프로젝트를 통해 실제 개발 경험을 쌓으며 전반적인 개발 역량을 높일 수 있었습니다. 기능 구현 과정에서 발생한 다양한 문제들을 직접 분석하고 해결하는 과정을 거치며, 문제 해결 능력 또한 자연스럽게 향상되었습니다. 팀원들과의 협업을 통해 커뮤니케이션 능력도 함께 성장할 수 있었던 의미 있는 경험이었습니다.

어려웠던 점

도서페이지에서 장바구니 추가 버튼을 눌러도 장바구니에 insert되지 않았습니다. cart와 book을 조인된 결과는 두 테이블의 데이터를 합쳐서 보여주는 것이지, 실제로는 두 개의 테이블이니까 select, update, delete만 가능하고 insert는 안됐었던 것입니다. cart에 book_id를 insert한 후 두 테이블을 조인한 select문을 쓰니 문제가 해결되었습니다.

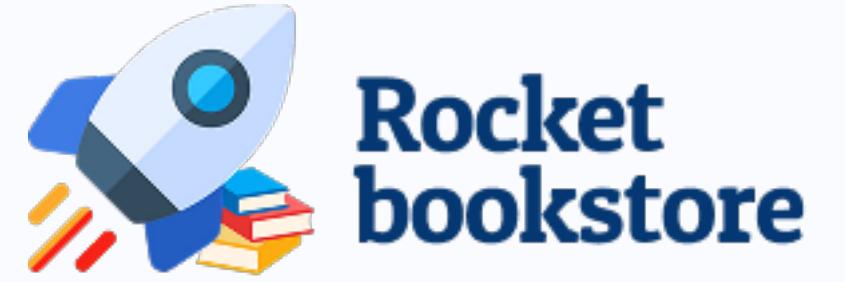
체크박스로 여러개를 선택하고 주문, 삭제, 총가격 계산 기능을 만들었는데 셋 중 한가지만 가능 했습니다. form태그를 나눠주니 제대로 작동했습니다.

CSS 작업에 대한 경험이 부족해 UI 구현 과정에서 다소 어려움을 겪었습니다. 향후에는 CSS에 대한 이해를 높여 보다 세련된 인터페이스 구현이 가능하도록 보완할 계획입니다.

개선할 부분 - 장바구니 가격 자동 계산

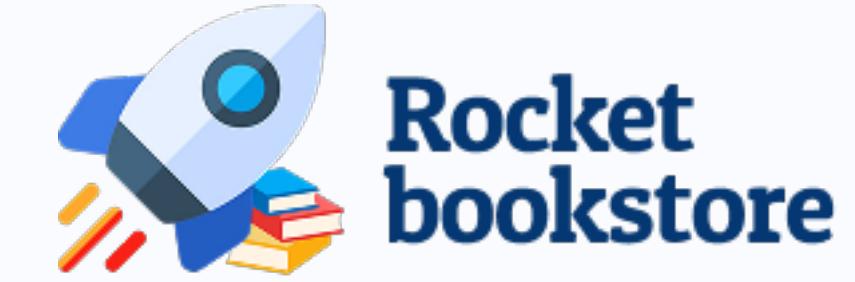
장바구니에서 수량을 증감시켜도 우측의 가격은 자동으로 업데이트되지 않습니다. 총가격을 계산할때도 체크박스로 선택을 하고 버튼을 눌러야 계산되고 계산 후 체크박스가 해제되어 주문하기를 하려면 다시 선택해야하므로 번거롭습니다. 총가격이 자동으로 계산되면 사용자가 더 편하고 효율적으로 구매를 할 수 있었을 것이라 생각됩니다.





관리자 페이지

Rocket bookstore



관리자 배송지 관리

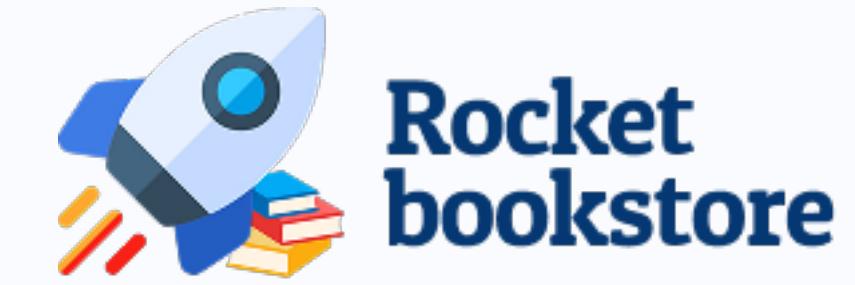
사용자 ID를 기준으로 데이터베이스에 저장된 배송지 정보를 조회하고, 조회된 데이터를 테이블 형태로 출력하여 사용자의 배송지 목록을 직관적으로 확인할 수 있도록 구현

Rocket bookstore
-admin-

[배송지 전체](#) [재고 관리](#) [회원 관리](#) [도서 관리](#) [주문 관리](#)

목록	사용자ID	주소
1	hong	서울시 강남구 테헤란로 123
2	U002	부산시 해운대구 센텀로 456





관리자 회원 관리

데이터베이스에 저장된 회원 정보를 조회하여 이름, 아이디, 이메일, 전화번호 등의 주요 정보를 테이블 형식으로 출력하고, 관리자가 회원 목록을 직관적이고 편리하게 조회할 수 있도록 구성

Rocket bookstore				
-admin-				
배송지 전체	재고 관리	<u>회원 관리</u>	도서 관리	주문 관리
<hr/>				
이름	아이디	이메일	전화번호	
admin	관리자	admin@gmail.com	01012345678	
hong	홍길남	hong@gmail.com	01012340000	
U002	김철수	kim@gmail.com	01098765432	
U003	이영희	lee@gmail.com	01056781234	



관리자 도서 관리

DB에 저장된 전체 도서 목록을 조회해 테이블 형태로 출력하고, 관리자가 도서를 추가, 수정, 삭제할 수 있도록 기능을 구현함. 도서 추가 버튼 클릭 시 신규 도서 등록 폼으로, 수정 버튼 클릭 시 해당 도서 정보 수정 폼으로 이동하도록 구성함. 삭제 버튼 클릭 시 선택한 도서를 데이터베이스에서 삭제 처리할 수 있도록 구현함.

Rocket bookstore
-admin-

배송지 전체 재고 관리 회원 관리 도서 관리 주문 관리

도서 추가

도서 ID	제목	저자	카테고리 ID	카테고리 이름	가격	재고	설명	이미지	관리
1001	UNLOCK AI 언락 AI	조용민	2	경제	16200	50	AI 리터러시가 나의 잠재력이 되는 세상이 왔다		수정 삭제
1002	생각이 많은 당신을 위한 말하기 수업	사이토 다카시, 최지현(옮김)	2	경제	16200	50	고민을 줄이고 대화가 쉬워지는 방법		수정 삭제

DB에서 조회한 데이터를 JSTL forEach문을 사용해 테이블 형태로 반복 출력되도록 구성

```
<c:forEach var="books" items="${booksList}">
    <tr>
        <td>${books.book_id}</td>
        <td>${books.title}</td>
        ...
    </tr>
</c:forEach>
```

관리자 도서 관리

도서 추가는 폼 입력 데이터를 DTO 객체에 담아 처리하고, DAO 패턴을 이용해 데이터베이스에 insert하도록 구성함. 등록 후에는 도서 목록 페이지로 리다이렉트하도록 처리함. 도서 수정은 수정 폼으로 이동하도록 구성하고, 수정된 데이터를 DAO를 이용해 데이터베이스에 업데이트하도록 처리함.

제목:

저자:

가격:

재고:

카테고리 :

설명:

전송

새 카테고리:

새 카테고리 추가

추가

책 ID: 1001 수정

제목: UNLOCK AI 언락 AI

저자: 조용민

가격: 16200

재고: 50

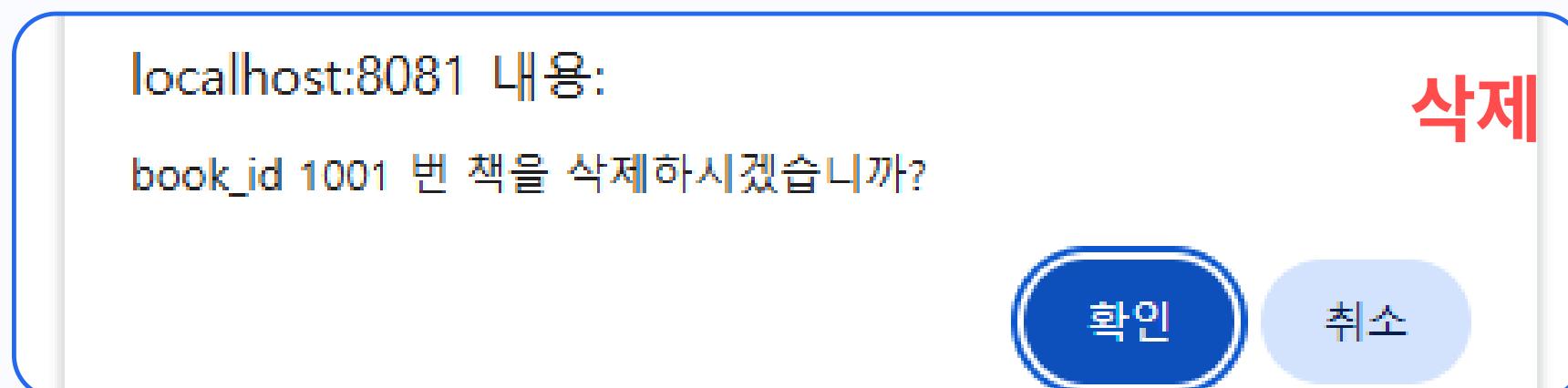
카테고리 : 경제 ▾

설명: AI 리터러시가 나의 잠재력이 도전장을 내밀었다.

전송

관리자 도서 관리

삭제 버튼 클릭 시 confirm 알림창을 띄워 사용자에게 한번 더 확인 후 삭제 처리도록 함. book_id를 파라미터로 전달받아 해당 도서 ID 값을 가져온 후 DAO를 통해 데이터베이스에서 삭제 처리한 후, 삭제 완료 후에는 관리자 도서 목록 페이지로 리다이렉트 처리하여 전체 목록이 갱신되도록 구성



```
<%@page contentType="text/html; charset=UTF-8"
%>

<%
    Long deleteid=Long.parseLong(request.getParameter("book_id")); // 도서 ID 파라미터 전달받기
    com.the.dao.BooksDao dao=new com.the.dao.BooksDao(); // DAO 객체 생성
    dao.delete(deleteid); // 도서 삭제 처리

    response.sendRedirect("adminSelect.jsp"); // 도서 리스트 페이지로 리다이렉트
%>
```

관리자 재고 관리

DB에 저장된 입·출고 이력을 조회해 도서별 입고 및 출고 현황을 추적하고, 현재 재고 수량을 테이블 형태로 출력하도록 구성함. 도서별 최신 재고 데이터를 식별하기 위해 log_id 기준 최신 값을 저장해 관리함. 상단 입력 폼을 통해 도서 재고를 실시간으로 추가할 수 있으며, 입력한 수량만큼 stock_logs 테이블에 데이터가 추가되고, books 테이블의 재고 수량이 자동으로 갱신되도록 처리함.

Rocket bookstore
-admin-

- 배송지 전체
- 재고 관리**
- 회원 관리
- 도서 관리
- 주문 관리

입고 현황 입력

도서 ID:
 입고 수량:

log_id	Book_id	책 제목	입고 수량	주문 번호	출고 수량	현재 재고
19	1113	나는 왜 남들보다 쉽게 지칠까	0	20250407000041	-1	-1
18	1004	인생에 달리기가 필요한 시간	0	20250407000042	-1	-2
17	1004	인생에 달리기가 필요한 시간	0	20250404000027	-1	-1
16	1072	봄밤의 모든 것	0	20250404000027	-1	-2
15	1093	어린 왕자	0	20250404000027	-1	-1
14	1003	한 권으로 끝내는 회계와 재무제표	0	20250404000026	-1	-1
13	1072	봄밤의 모든 것	0	20250404000026	-1	-1

```
Map<Long, Long> latestLogIdMap = new HashMap<>();
for (StockLogsDto dto : dtos) {
    Long bookId = dto.getBook_id();
    Long logId = dto.getLog_id();
    if (!latestLogIdMap.containsKey(bookId) || logId > latestLogIdMap.get(bookId)) {
        latestLogIdMap.put(bookId, logId);
    }
}
```

```
// 출고량 0, change_stock 초기값은 입고 수량
StockLogsDto dto = new StockLogsDto(0L, book_id, in_stock, 0L, in_stock);
// DAO의 insert 메서드에서 책이 존재하지 않으면 Exception이 발생함
StockLogsDao.insert(dto);
```



관리자 주문 관리

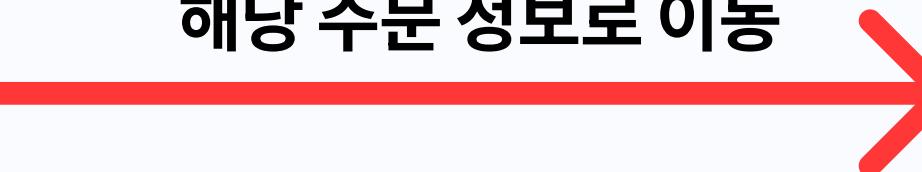
사용자들의 주문 정보를 조회해 주문 내역을 테이블 형태로 출력하고, 관리자가 주문 상태를 확인하고 추적할 수 있도록 구성함. 모든 주문 데이터를 가져오고 이전 주문 ID와 비교하여 중복된 주문을 배제하여 효율적인 데이터 관리를 제공.

Rocket bookstore
-admin-

배송지 전체 재고 관리 회원 관리 도서 관리 주문 관리

목록	주문번호	사용자ID	주문날짜	책	수량	운송장번호
1	20250407000004	hong	2025-04-07	생각이 많은 당신을 위한 말하기 수업	2	RK20250407000004
2	20250407000008	hong	2025-04-07	코딩 자율학습 잔재미코딩의 파이썬 데이터 분석	2	RK20250407000008

주문 번호 클릭 시
해당 주문 정보로 이동



주문정보

회원 ID : hong
회원 이름 : 홍길남
회원 이메일 : hong@gmail.com
회원 전화번호 : 01012340000

주문번호 : 20250404000027
운송장번호 : RK20250404000027
배송지
서울시 강남구 테헤란로 123
주문상품
어린 왕자 1
봄밤의 모든 것 1
인생에 달리기가 필요한 시간 1
총 가격 : 47320 원

```
// 전체 주문 리스트 가져오기
ArrayList<OrdersReceiptDto> receiptList = OrdersReceiptDao.selectAll();

// 중복 제거된 리스트
ArrayList<OrdersReceiptDto> filteredList = new ArrayList<>();
String lastOrderId = ""; // 이전 주문번호

for (OrdersReceiptDto item : receiptList) {
    if (!item.getOrder_id().equals(lastOrderId)) { // 이전 주문번호와 다르면 추가
        filteredList.add(item);
        lastOrderId = item.getOrder_id(); // 현재 주문번호를 저장
    }
}
```



관리자 주문 관리

각 배송 상태에 따라 다른 이미지가 동적으로 로드되어 관리자가 배송 상태를 직관적으로 확인하고 수정할 수 있게 하며, 이를 통해 관리자의 작업 효율성을 높임. 동적인 이미지 적용은 사용자 경험(UX)을 향상시키고, 관리자의 실수 방지와 시각적 피드백을 통해 각 상태를 빠르게 인식할 수 있게 함

Rocket bookstore
-admin-

배송지 전체 재고 관리 회원 관리 도서 관리 주문 관리

목록	주문번호	사용자ID	주문날짜	책	수량	운송장번호
1	2025040700004	hong	2025-04-07	생각이 많은 당신을 위한 말하기 수업	2	RK20250407000004
2	2025040700008	hong	2025-04-07	코딩 자율학습 잔재미코딩의 파이썬 데이터 분석	2	RK20250407000008

운송장 번호 클릭 시
배송 정보로 이동



상품준비중 배송준비중 배송중 배송완료

배송상태

상품 접수 완료

※1=상품준비중 2=배송준비중 3=배송중 4=배송완료

```

class="delivery_process">
    iv>
        .png"
        <p>상품준비중</p>
    div>
    iv>
        .png"
        <p>배송준비중</p>
    div>
    iv>
        .png"
        <p>배송중</p>
    div>
    iv>
        .png"
        <p>배송완료</p>
    div>

```

자체 피드백



김환희

- 재고관리 데이터베이스 설계 및 구축
- 메인 페이지 레이아웃 설계 및 스타일링
- 메인 페이지 반응형 웹 적용
- 프로젝트 전역 스타일 통일화 및 CSS 리팩토링
- Favicon 디자인 및 적용

좋았던 점

개발 과정에서 팀원들과 적극적으로 소통하고 협력하며 프로젝트를 진행할 수 있었습니다. 특히 서로 모르는 부분이나 막히는 부분에 대해 적극적으로 의견을 나누고 해결해 나가는 과정이 좋았습니다. 각자의 역할에 최선을 다하며 프로젝트 완성도를 높일 수 있었던 점이 좋았습니다.

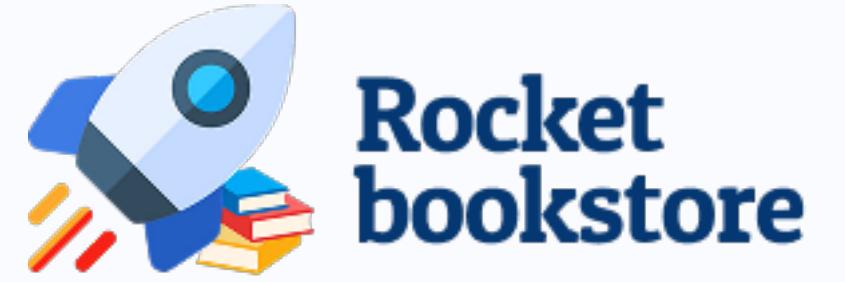
어려웠던 점

재고 관리 파트를 맡아 개발하면서 입출고 내역과 도서 재고 수량을 자동으로 연동하는 부분이 가장 고민이 많았던 부분이었습니다. 입고나 출고 처리 시 재고가 자동으로 변경되고, 주문 발생 시 출고 처리와 연결되도록 구현하는 과정이 쉽지 않았지만 여러 번 수정하고 테스트하면서 해결해 나갔습니다. 앞으로도 이런 데이터 처리나 로직 설계 부분을 더 많이 공부하고, 다양한 상황에서도 안정적으로 동작하는 코드를 작성할 수 있도록 실력을 키워나가고 싶습니다.

개선할 점

Header, Footer 등 공통 UI 요소를 별도의 파일로 모듈화하고 include 방식으로 관리하였다면 코드 재사용성과 유지보수 측면에서 더 효율적이고 일관된 개발이 가능했을 것으로 생각됩니다. 또한 `request.getContextPath()`를 활용하여 프로젝트 루트 경로를 동적으로 처리하면 경로 수정에 대한 의존성을 줄이고 확장성과 유지보수성을 높일 수 있을 것으로 판단됩니다.





프로젝트 시연회

Rocket bookstore



4. 프로젝트 개선할 점

Rocket bookstore



반응형 웹 대응

개발 초기 디자인 설계 단계에서 다양한 디바이스 해상도를 고려한 레이아웃 구성 및 UI 컴포넌트 설계를 진행했다면, 이후 미디어쿼리 적용 및 레이아웃 수정 과정에서의 리팩토링 비용을 줄이고 유지보수성을 높일 수 있었을 것으로 판단됨.

리뷰/평점 기능 추가

사용자 리뷰 및 평점 기능을 추가함으로써 사용자 간 피드백 공유 및 커뮤니케이션이 가능해지고, 추천 기반 데이터 확보를 통해 책 추천 로직 고도화 및 사용자 구매 전환율 개선에 기여할 수 있었을 것으로 보임.

쿠폰/할인 기능 추가

쿠폰 할인 기능을 추가하여 사용자가 특정 조건(이벤트, 등급, 첫구매 등)에 따라 할인을 적용받을 수 있도록 하면, 구매유도와 재방문율 향상에도 기여할 수 있을 것으로 기대함.

이미지 동적 처리

도서의 이미지 관리 및 출력 방식을 정적 파일 참조 방식이 아닌 DB 또는 파일 서버 연동을 통한 동적 처리로 개선한다면, 이미지 수정 및 추가 작업 시 배포 없이 관리가 가능해지고, 다양한 디바이스 해상도에 대응하는 최적화된 이미지 제공 및 유지보수 효율성을 높일 수 있었을 것으로 판단됨.

배송원 매칭 기능

현재 시스템에는 배송원 매칭 기능이 포함되지 않아, 실시간 배송 상황이 추적이 어려움. 프로젝트의 목표에 맞춰, 1:1 배송원 매칭 시스템을 추가하고, 배송원의 위치와 예상 배송 시간을 실시간으로 확인할 수 있도록 개선하는 것이 필요함.

총 가격 자동 계산

도서 선택 시 총 금액이 자동으로 갱신되지 않고 별도의 버튼 클릭을 통해 총 금액이 계산되는 방식으로 구현됨. 추후 JavaScript 이벤트 처리 및 동적 렌더링을 적용하여 사용자의 선택 변경에 따라 총 금액이 실시간으로 자동 갱신되도록 개선할 필요가 있다고 판단됨.