

# 목차

1. 프로젝트 설명
  - a. 프로젝트 폴더 계층 구조
  - b. 사용 기술 및 언어
2. **ERD**
3. 메인 페이지
4. 책
  - a. 상세정보
  - b. 검색
5. 장바구니
6. 주문

## 1-a. 프로젝트 폴더 계층 구조

**설명 :** sg폴더가 본인이 한것입니다.

- **src/main/java/**
  - **com/bookmarket/app**
    - **controller** : MVC 컨트롤러.
    - **dto** : MVC DTO.
    - **mapper** : MVC DAO.
    - **restcontroller** : API 용 컨트롤러.
    - **service** : MVC 모델 서비스 레이어.
    - **BookMarketApplication.java** : 프로그램 진입점.
- **src/main/resources/**
  - **mapper** : SQL 코드들.
- **src/main/webapp/**
  - **resources** : css, js, Multipart 등
  - **WEB-INF/views** : MVC View (JSP)

## 1-b. 사용 기술 및 언어

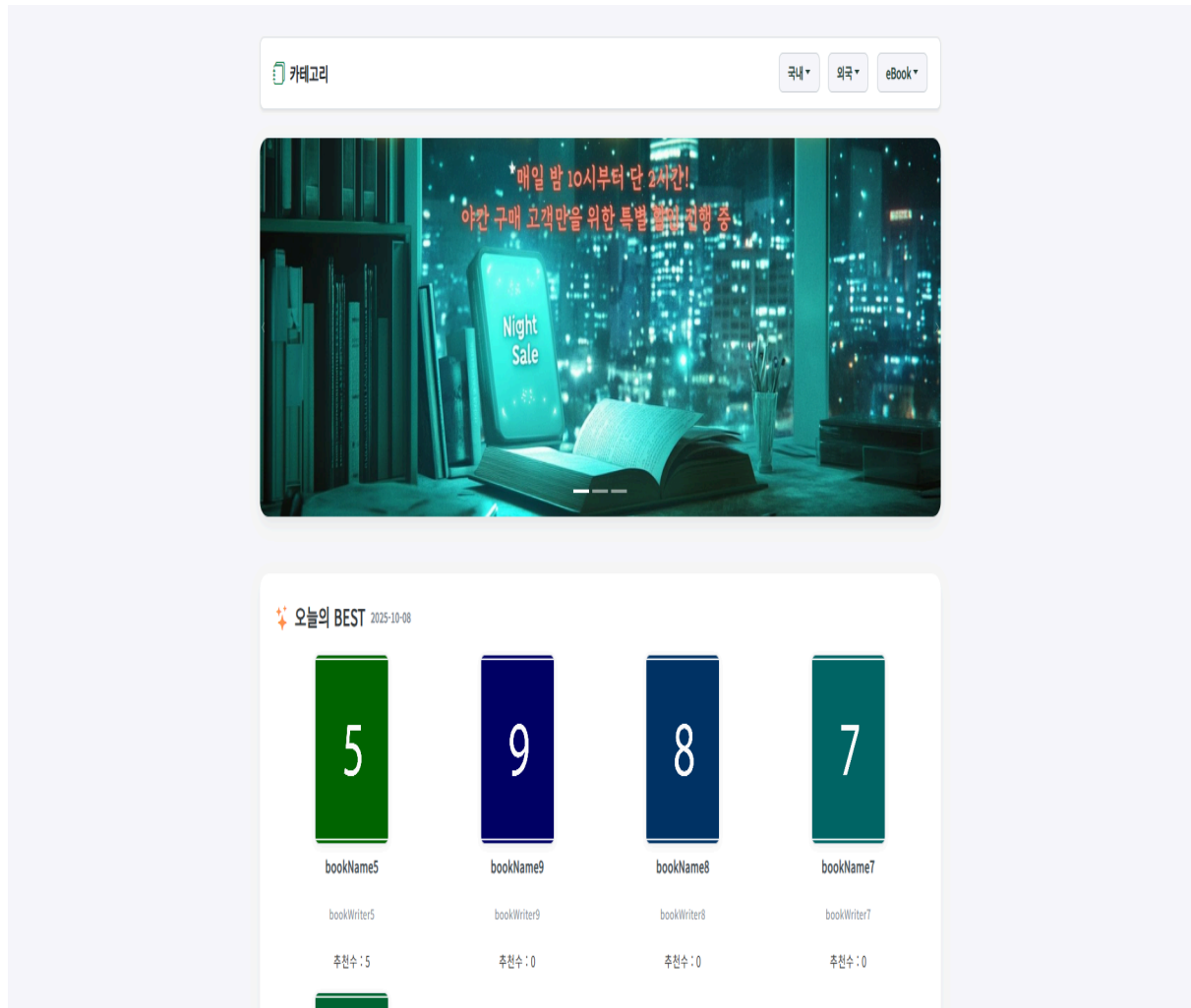
- **기술** : Spring boot, mybatis, oracle sql, java, ajax, jsp
- **언어** : SQL, JAVA, JS

## 2-a. ERD

ERD 주소 : <https://www.erdcloud.com/d/CptyXbQ9G42t5ZCfi>

설명 : 출판사가 책 등록을 해서 판매하고 사용자가 사는 상황을 고려하여 만들었습니다.

### 3. 메인페이지



책 추천 설명 : 판매가 가능한 책들 중에서 오늘날짜 - 7 부터 오늘날짜까지 추천수를 기준으로 상위 5개만 가져오도록 하였습니다.

### 4-a. 책 상세정보

1	book-desc-container	
	bookId: 1	
	추천수: 0	
	수량: 1	1000 원
	장바구니	추천하기

book-info-container

#### 설명 :

- **수량 표시** : **if**문으로 작성한게 아닌 효율성 및 가독성을 위해 정규식 사용하였습니다. 초기 수량을 **1**로 표시를 해 수량의 변화가 발생하고 해당 변화가 정규식에 부합하면 해당 값으로 표시하고 정규식에 부합하지 않으면 롤백처리를 통해 이전 값을 표시합니다. 이전 값은 정상처리된 값이기 때문에 롤백처리가 됩니다.
- **가격 표시** : 수량 표시와 동일한 원리이며 원화표시 정규식을 사용하였습니다.
- **추천하기** : 추천 테이블에서 오늘날짜에 해당 데이터가 있는지 없는지를 확인하여 구현하였습니다. 있으면 실패 없으면 성공입니다. **1일 1회**로 제한하였습니다.
- **장바구니** : 물건 넣기 기능입니다. 추천하기 기능과 비슷하게 해당 데이터가 있는지 없는지에 대한 판별을 기준으로 구현하였습니다. 따라서 장바구니에 중복으로 책을 넣을 수가 없습니다.
- **절판표시** : 재고가 **0**이거나 절판되면 책을 장바구니에 넣을 수 없도록 하였습니다.

## 4-b. 책 검색



검색어를 입력하세요



로그인

검색 결과: book

<div>9</div> <div>bookName9</div> <div>저자: bookWriter9</div> <div>9000원</div> <div>상세보기</div>	<div>8</div> <div>bookName8</div> <div>저자: bookWriter8</div> <div>8000원</div> <div>상세보기</div>	<div>7</div> <div>bookName7</div> <div>저자: bookWriter7</div> <div>7000원</div> <div>상세보기</div>
<div>6</div> <div>bookName6</div> <div>저자: bookWriter6</div> <div>6000원</div> <div>상세보기</div>	<div>5</div> <div>bookName5</div> <div>저자: bookWriter5</div> <div>5000원</div> <div>상세보기</div>	<div>4</div> <div>bookName4</div> <div>저자: bookWriter4</div> <div>4000원</div> <div>상세보기</div>

**설명** : 판매신청된 책들중에서 판매신청완료된 책들을 **WHERE %keyword%** 필터링을 통한뒤, 최신순으로 페이징 처리를 하여 보여주도록 하였습니다. **1**페이지당 **15**개 기준입니다.

## 5. 장바구니



메인

주문신청목록

주문완료목록

주문반려목록

주문번호: 23

2025-05-11

주문 도서 목록

3

bookName3 주문수량 : 1권    단가 : 3,000원    총 합계 : 3,000원

2

bookName2 주문수량 : 1권    단가 : 2,000원    총 합계 : 2,000원

1

bookName1 주문수량 : 1권    단가 : 1,000원    총 합계 : 1,000원

총 주문신청금액 : 6,000원

주문신청취소

주문번호: 22

2025-05-11

주문 도서 목록

**주문처리과정** : 장바구니에서 주문을 하면 주문데이터를 입력하게 되는데 주문데이터를 입력시 주문과 주문상세를 **INSERT ALL**을 이용해서 하나의 트랜잭션으로 묶어서 처리하였습니다. 이렇게 하나로 묶지 않고 따로 처리하게 되면 주문과 주문상세의 데이터가 일치하지 않는 경우가 발생할 수 있기에 이렇게 묶어서 처리하였습니다.

**주문확인** : 주문한 것을 확인할 때마다 테이블에 총주문금액 컬럼을 추가하지 않아도 주문에 있는 상품의 총합을 구하는 연산을 통해 구할 수 있습니다. 하지만 이렇게 하면 주문을 확인할 때마다 연산을 해야하는 비효율적인 과정이 생기게 됩니다. 따라서 주문 테이블에 총주문금액 컬럼을 추가하여 이러한 연산과정을 없애보았습니다.