



9

[ BOOKREALM ] 이영진, 김단아, 안우용, 최원규



# 목차

## 01 팀 구성 및 역할

## 02 프로젝트 수행 절차 및 방법

기술 스택

WBS

화면설계도

## 03 프로젝트 수행 결과

DB설계

서비스 아키텍처

백엔드 수행결과

시연영상

## 04 느낀 점

추후 고려사항

팀원들의 한마디



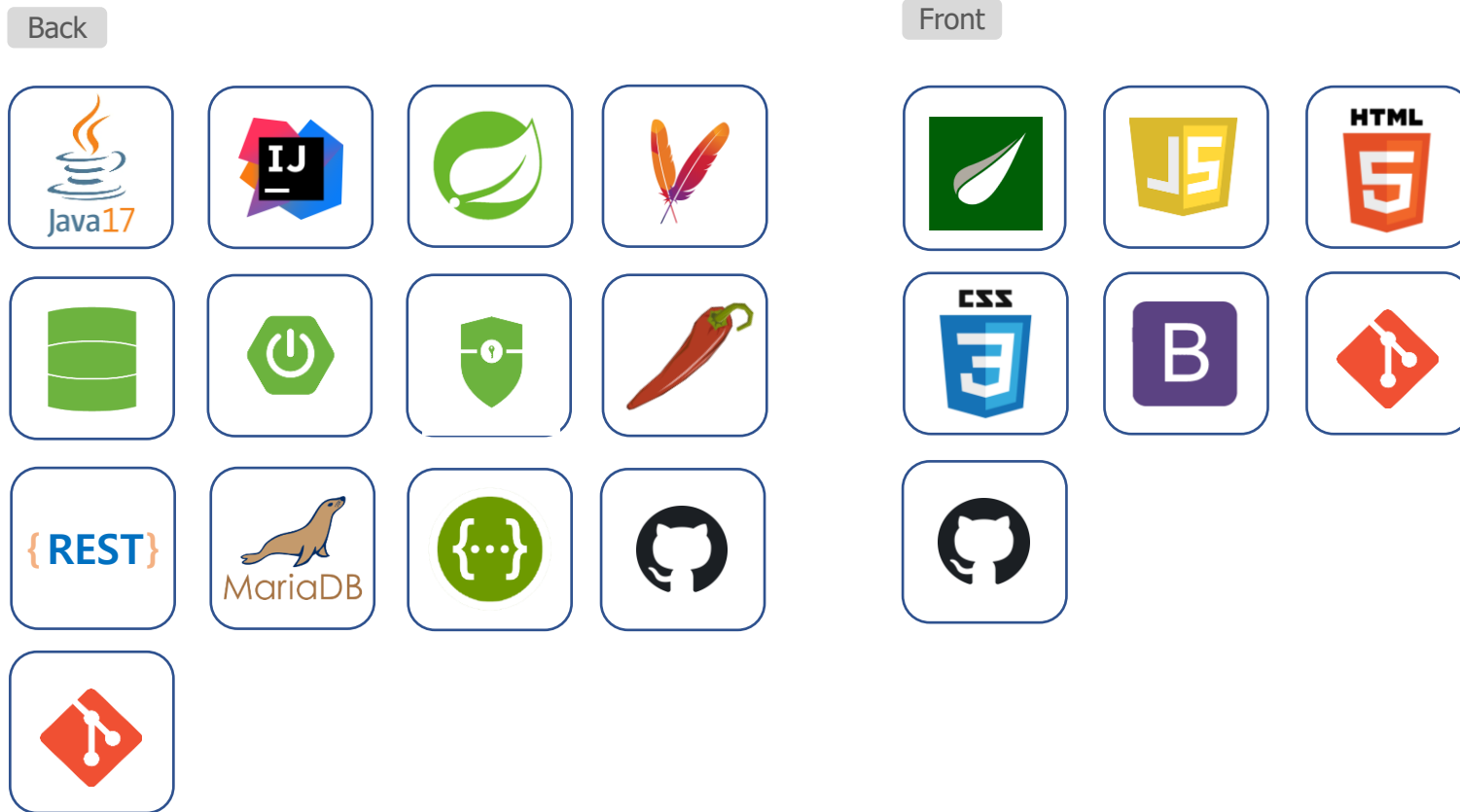
# 팀 구성 및 역할

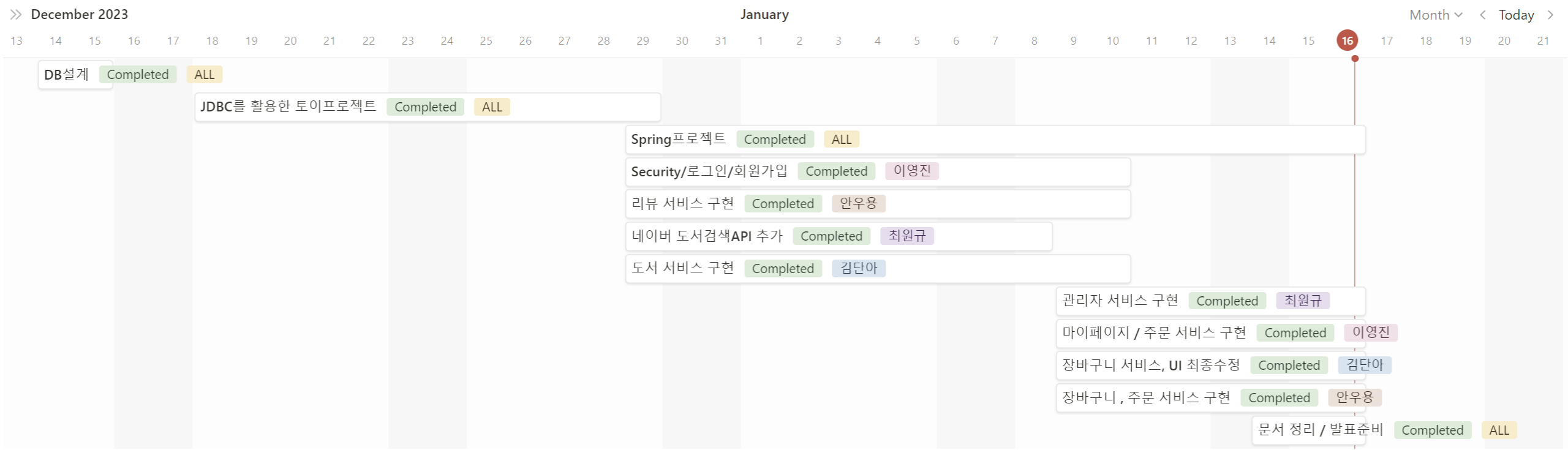
## 팀 구성 및 역할

팀원 명		역할
BE	이영진	프로젝트 총괄책임, Spring Security 적용, 로그인/회원가입 구현 마이페이지 구현, 주문 서비스 구현
	김단아	UI총괄책임, 도서 서비스 구현, 장바구니 서비스 구현
	안우용	문서총괄책임, 리뷰 서비스 구현, 장바구니 서비스 구현, 주문 서비스 구현
	최원규	GitHub 총괄책임, 관리자 서비스 구현, 네이버 도서검색 API 적용
공통		DB설계 HTML, CSS, Thymeleaf, Bootstrap, JavaScript

# 프로젝트 수행 절차 및 방법

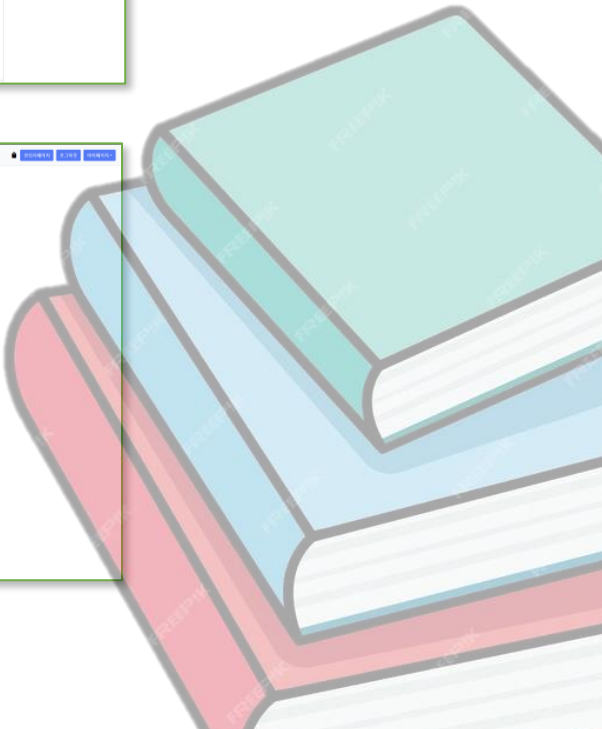
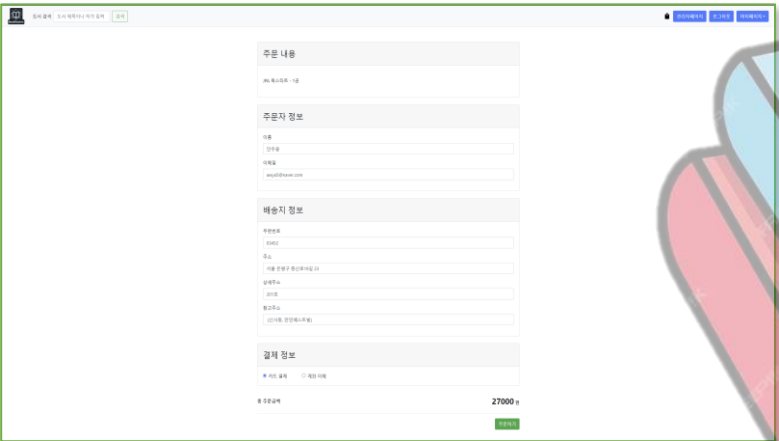
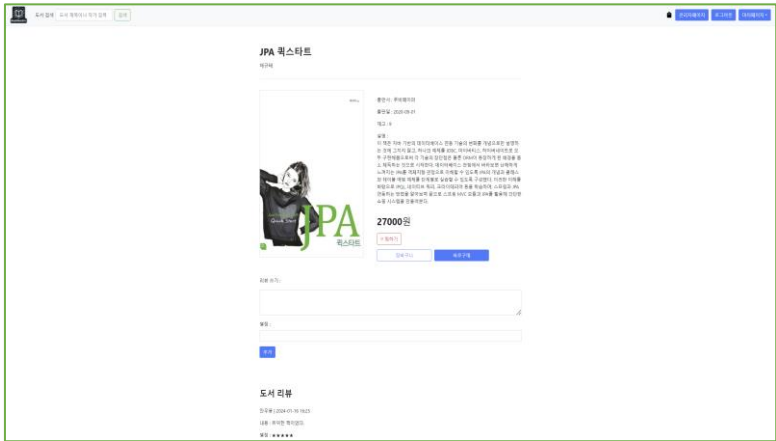
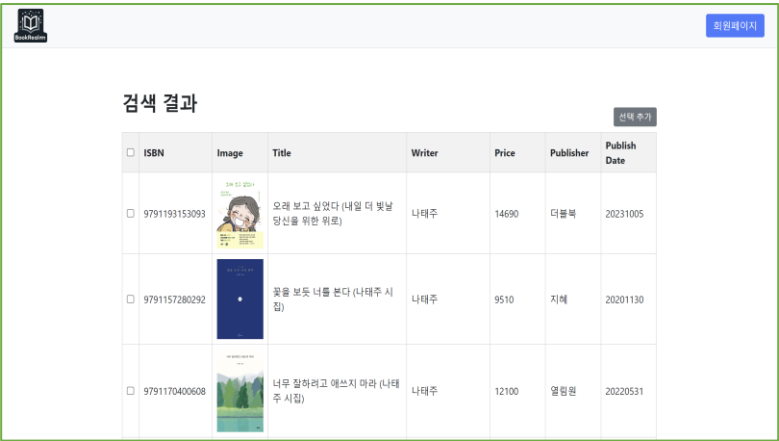
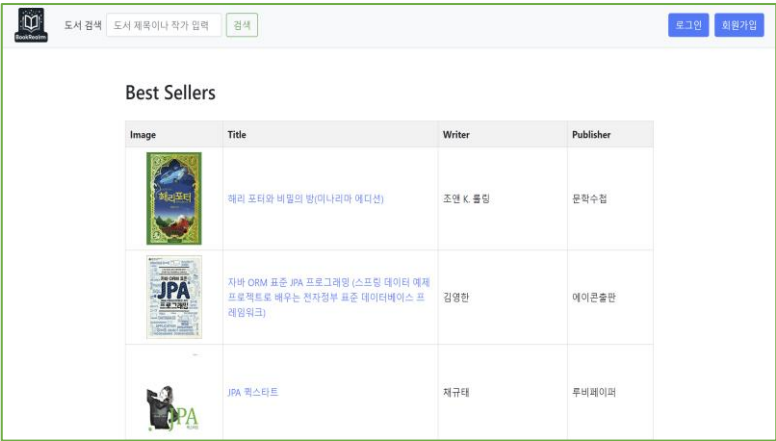
## 기술 스택





# 프로젝트 수행 절차 및 방법

## 화면설계도

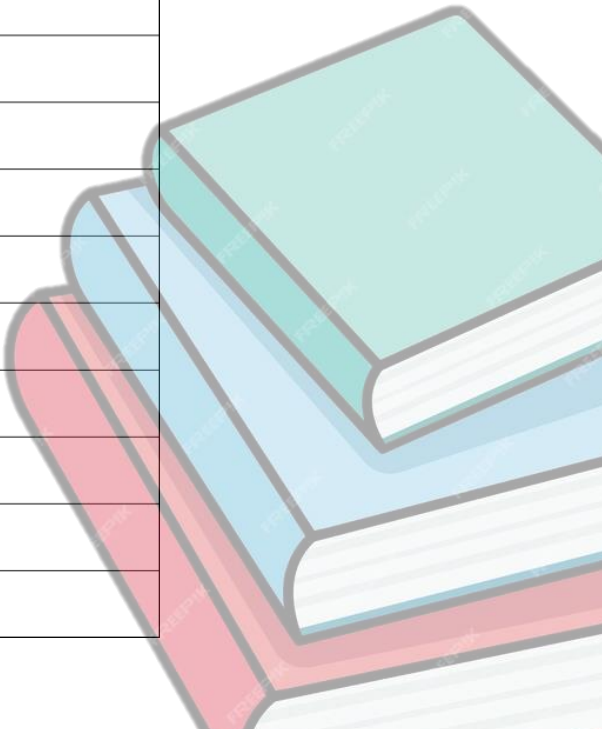


# 프로젝트 수행 결과

DB설계

## 요구사항분석

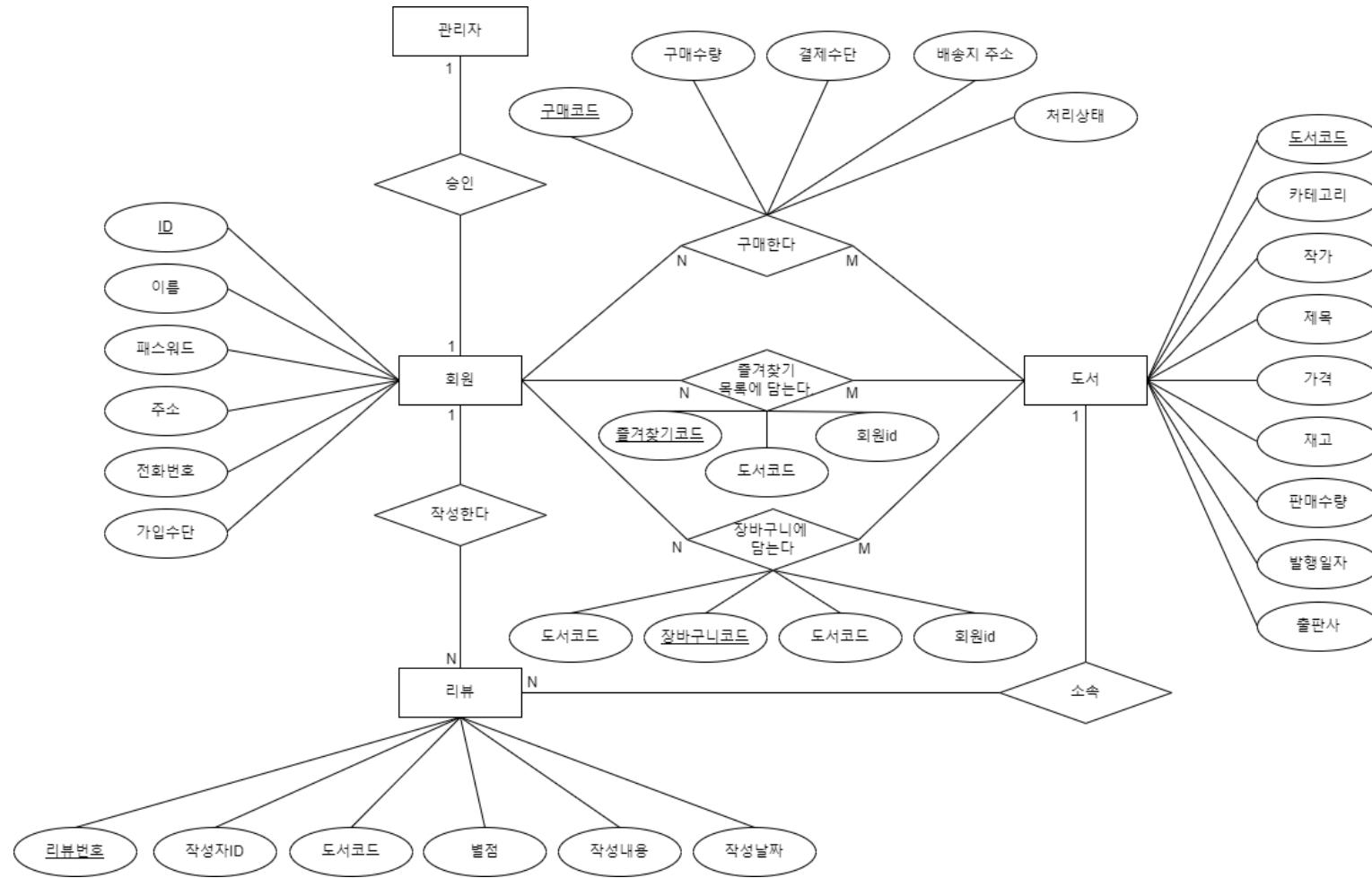
항목	설명
회원	도서를 구매할 수 있다.
	제목,작가,베스트셀러 여부, 카테고리 기준으로 검색 할 수 있다.
	장바구니에 도서를 저장할 수 있다.
	도서를 마이리스트에 저장할 수 있다.
	장바구니에 있거나 검색한 도서를 구매할 수 있다.
	구매한 도서의 리뷰를 작성하고 별점을 줄 수 있다.
	구매한 도서를 교환, 환불 신청할 수 있다.
	도서의 상세 정보를 확인할 수 있다.(제목,작가,카테고리, 가격, 수량, 별점, 리뷰)
비회원	제목,작가,베스트셀러 여부, 카테고리 기준으로 검색 할 수 있다.
	장바구니에 있거나 검색한 도서를 구매할 수 있다.
	도서의 상세 정보를 확인할 수 있다.(제목,작가,카테고리, 가격, 수량, 별점, 리뷰)
관리자	도서를 추가, 삭제, 수정, 조회할 수 있다.
	구매완료된 도서를 조회할 수 있다.
	회원 정보를 관리할 수 있다.



# 프로젝트 수행 결과

DB설계

## 개념적설계

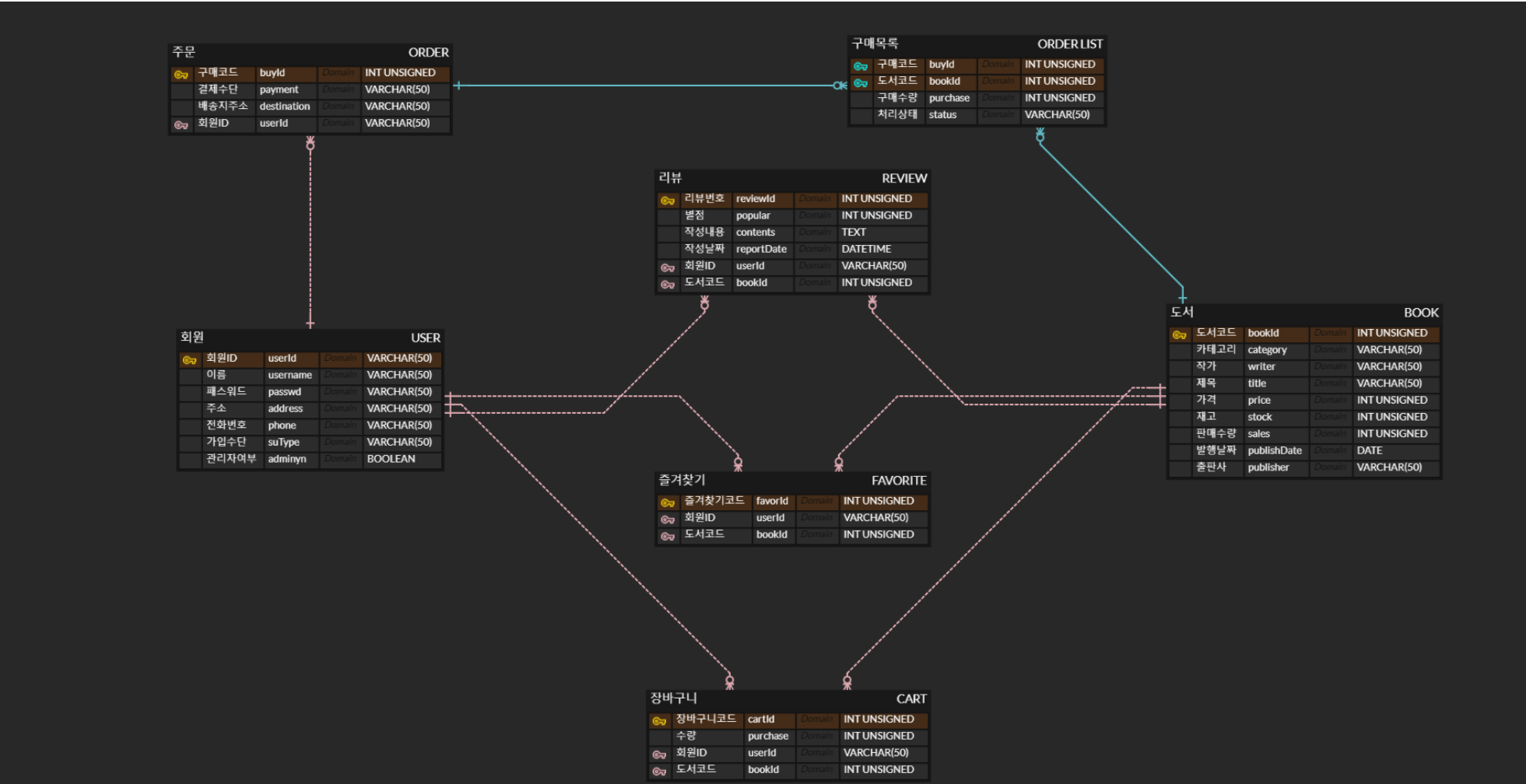




# 프로젝트 수행 결과

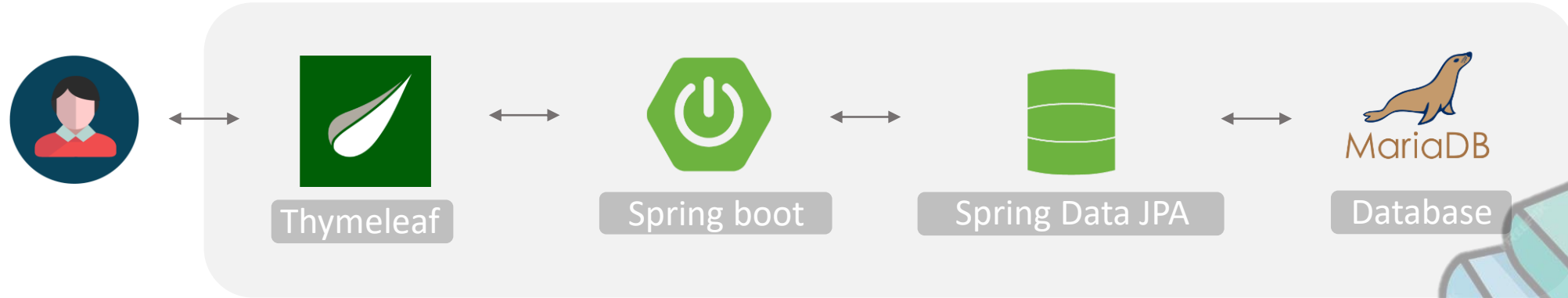
DB설계

논리적설계



# 프로젝트 수행 결과

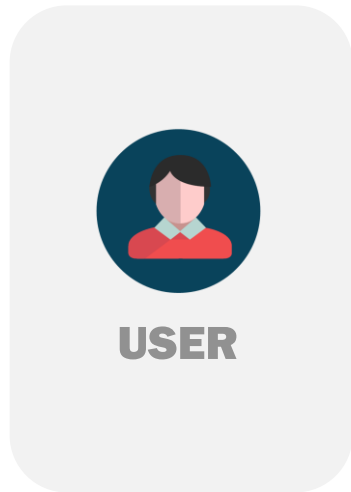
## 서비스 아키텍처



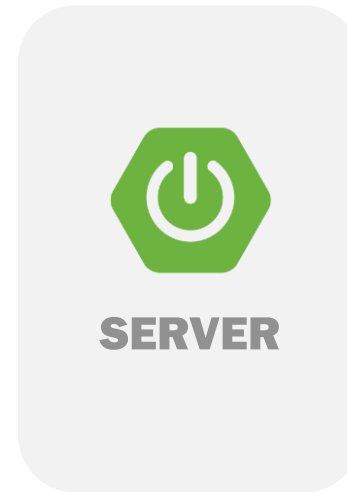
# 프로젝트 수행 결과

Back

## Security 프로세스



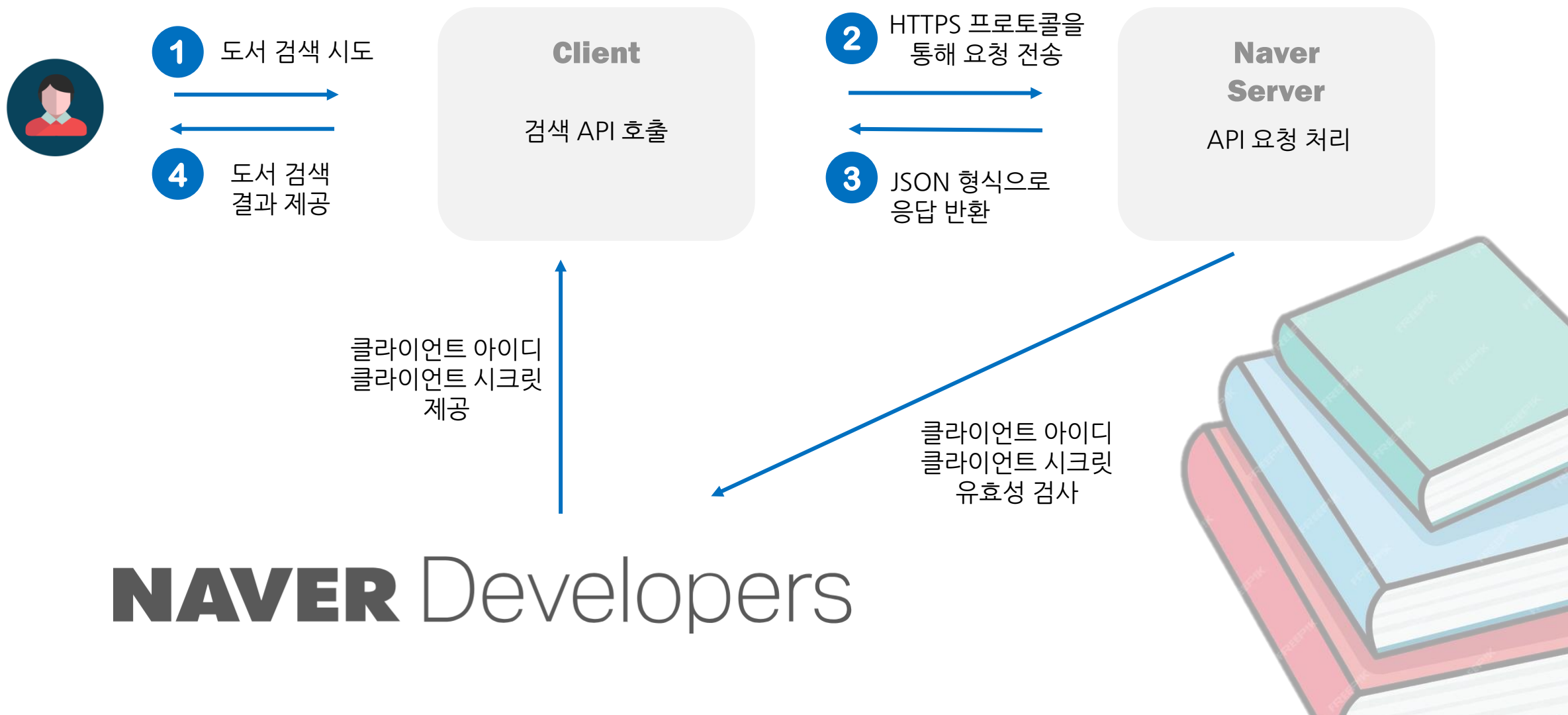
로그인 요청 (post)



# 프로젝트 수행 결과

Back


## Naver 검색 API 흐름



# 프로젝트 수행 결과

Back

## RESTful API

 **Swagger.**  
Supported by SMARTBEAR

/v3/api-docs

Explore

### Bookrealm REST API

1.0.0 OAS 3.0

/v3/api-docs

Bookrealm 프로젝트에서 사용된 REST API 명세

Servers

http://localhost:8080 - Generated server url

#### 관리자 API

관리자 페이지 기능 API

POST

/admin/member/role

관리자 권한 관리

▼

POST

/admin/member/delete

회원 삭제

▼

POST

/admin/book/save

도서 목록 저장

▼

POST

/admin/book/delete

도서 목록 삭제

▼

# 프로젝트 수행 결과

Back

## Test Code

```
Run service in bookrealm x
[Icons]
✓ Tests passed: 16 of 16 tests - 1 sec 412 ms
service (com.bookrealm) 1 sec 412 ms
  ReviewServiceTest 605 ms
    유저별리뷰검색() 540 ms
    도서별리뷰검색() 13 ms
    리뷰삭제() 23 ms
    리뷰수정() 15 ms
    리뷰쓰기() 14 ms
  BookServiceTest 68 ms
    도서_상세_조회() 8 ms
    베스트셀러_조회() 27 ms
    도서_제목작가_검색() 20 ms
    도서_추가() 13 ms
  OrderServiceTest 91 ms
    구매() 59 ms
    구매총액() 18 ms
    결제수단선택및결제() 14 ms
  CartServiceTest 104 ms
    장바구니_수량업데이트() 69 ms
    장바구니_담기() 13 ms
    장바구니_제거() 22 ms
  AdminServiceImplTest 544 ms
    SAVEALL_도서검색() 544 ms
    2024-01-16T16:37:05.860+09:00 DEBUG 26160 --- [main] org.hibernate.SQL : insert into member (address,det
    Hibernate: insert into member (address,detail_address,extra_address,postcode,email,passwd,phone,role,su_type,username) values (?,?,,?,,?
    2024-01-16T16:37:05.861+09:00 INFO 26160 --- [main] p6spy : #1705390625861 | took 0ms | sta
    insert into member (address,detail_address,extra_address,postcode,email,passwd,phone,role,su_type,username) values (?,?,,?,,?,,?,,?)
    insert into member (address,detail_address,extra_address,postcode,email,passwd,phone,role,su_type,username) values ('시흥시','대은로','62'
    2024-01-16T16:37:05.863+09:00 DEBUG 26160 --- [main] org.hibernate.SQL : insert into book (category,desc
    Hibernate: insert into book (category,description,image,price,publish_date,publisher,sales,stock,title,writer) values (?,?,,?,,?,,?,,?)
    2024-01-16T16:37:05.864+09:00 INFO 26160 --- [main] p6spy : #1705390625864 | took 0ms | sta
    insert into book (category,description,image,price,publish_date,publisher,sales,stock,title,writer) values (?,?,,?,,?,,?,,?)
    insert into book (category,description,image,price,publish_date,publisher,sales,stock,title,writer) values ('소셜','1999년 <해리 포터와 마
    2024-01-16T16:37:05.867+09:00 DEBUG 26160 --- [main] org.hibernate.SQL : insert into cart (book_id,membe
    Hibernate: insert into cart (book_id,member_id,purchase) values (?,?,?)
    2024-01-16T16:37:05.868+09:00 INFO 26160 --- [main] p6spy : #1705390625868 | took 0ms | sta
    insert into cart (book_id,member_id,purchase) values (?,?,?)
    insert into cart (book_id,member_id,purchase) values (114,40,2);
    2024-01-16T16:37:05.872+09:00 DEBUG 26160 --- [main] org.hibernate.SQL : insert into orders (address,det
    Hibernate: insert into orders (address,detail_address,extra_address,postcode,member_id,order_date,payment,total_amount) values (?,?,,?,,?
    2024-01-16T16:37:05.873+09:00 INFO 26160 --- [main] p6spy : #1705390625873 | took 0ms | sta
    insert into orders (address,detail_address,extra_address,postcode,member_id,order_date,payment,total_amount) values (?,?,,?,,?,,?,,?)
    insert into orders (address,detail_address,extra_address,postcode,member_id,order_date,payment,total_amount) values ('서울시','갈현로4길',
    2024-01-16T16:37:05.880+09:00 DEBUG 26160 --- [main] org.hibernate.SQL : select c1_0.cart_id,c1_0.book_i
    Hibernate: select c1_0.cart_id,c1_0.book_id,c1_0.member_id,c1_0.purchase from cart c1_0 left join member m1_0 on m1_0.member_id=c1_0.membe
    2024-01-16T16:37:05.881+09:00 INFO 26160 --- [main] p6spy : #1705390625881 | took 0ms | sta
    select c1_0.cart_id,c1_0.book_id,c1_0.member_id,c1_0.purchase from cart c1_0 left join member m1_0 on m1_0.member_id=c1_0.member_id where
    select c1_0.cart_id,c1_0.book_id,c1_0.member_id,c1_0.purchase from cart c1_0 left join member m1_0 on m1_0.member_id=c1_0.member_id where
```

시연 영상

Book Realm

localhost:8080

도서 검색





도서 제목이나 작가 입력

검색

로그인

회원가입

### Best Sellers

Image	Title	Writer	Publisher
	내가 한 말을 내가 오해하지 않기로 함 (빠더너스 문상훈 책)	문상훈	위너스북
	마흔에 읽는 쇼펜하우어 (마흔의 위기를 다스리는 철학 수업)	강용수	유노북스
	혼한남매 15	백난도	미래엔아이세움
	트렌드 코리아 2024 (2024 대한민국 소비트렌드 전망)	김난도, 전미영, 최지혜, 이수진, 권정윤, 한다혜, 이준영, 이향은, 이혜원, 추예린, 전다현	미래의창



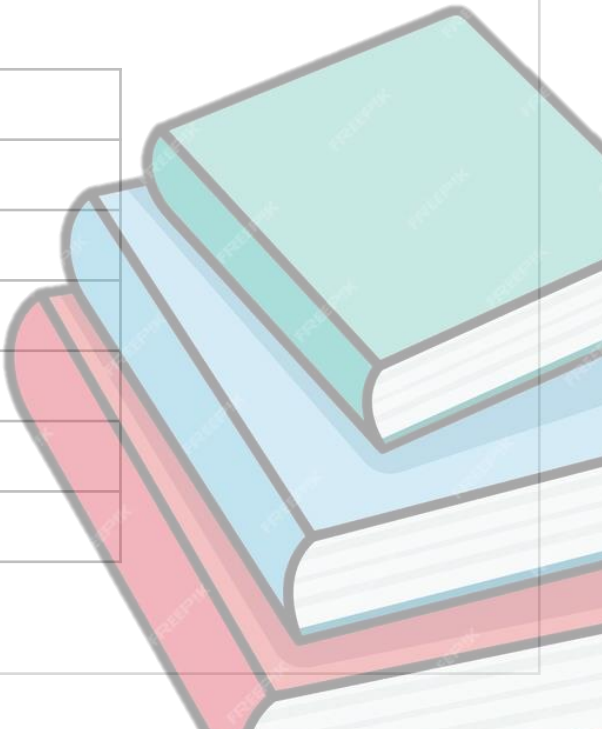
# 추후 고려사항

## ▼ 아쉬운 점

백엔드	JWT 미적용
	테스트 코드 작성이 미흡하여, 테스트 주도 개발 미적용
	결제시스템, SNS로그인 등 일부기능 미 구현
	Exception Handling 균일화 필요

## ▼ 향후 개발 사항

향후 개발 사항	결제 시스템 구현 , SNS 로그인 구현
	테스트 주도 개발 적용, Exception Handling 균일화
	바로 구매 시 수량변경 기능 구현
	중고거래 기능 추가
	찜목록 기능 구현, 리뷰수정 변경(Ajax 활용)
	Front-End 구현
	JWT 적용





## 팀원들의 한 마디



### 영진

처음 해보는 프로젝트인 만큼 새로 경험하는 것들과 나 자신의 부족한 것들이 많이 느껴지는 프로젝트였습니다. 하지만 팀원들과 함께 같이 문제를 해결해 보고 같이 성장하면서 많은 것들을 배울 수 있었고 더 열심히 공부해야겠다고 느꼈습니다. 하지만 배울 때와 달리 직접 프로젝트를 해보니 재미있었습니다.

### 단아

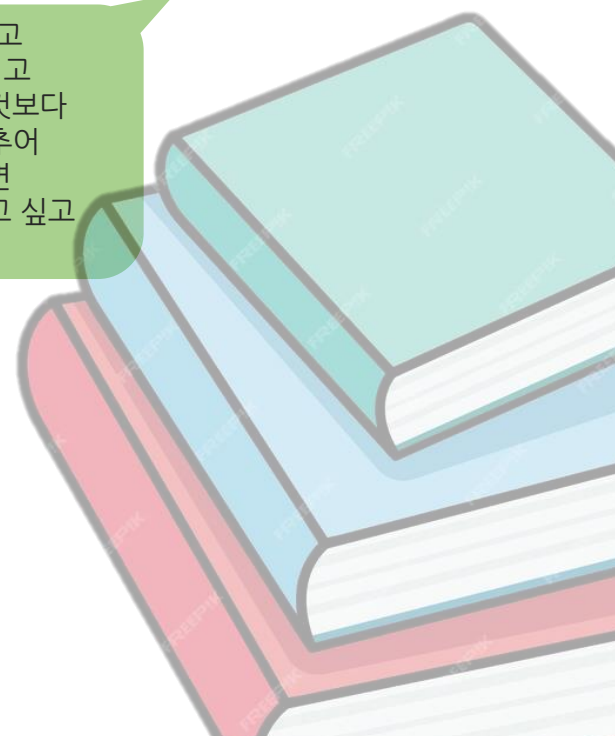
처음 코드를 작성했을 때 막막하고, 수많은 에러에 당황하기도 하고 답답하기도 했지만 계속 공부하고 고민하며 에러에서 성공으로 바뀌는 순간 너무 기뻐던 것 같습니다. 부족한 부분도 많고 어려운 부분도 많았지만 팀 프로젝트이기에 책임감을 가지고 마무리할 수 있었던 것 같습니다. 파트 간의 지속적인 의사소통으로 점점 좋아지는 결과물을 보며 뿌듯함을 느꼈습니다. 이번 프로젝트는 팀워크와 배려심을 배울 수 있는 값진 시간이었습니다.

### 우용

프로젝트를 통해 팀원 간 소통의 중요성을 깨달았습니다. Spring 이 해도 부족으로 Front-end에서 원하는 기능을 구현하지 못한 점이 아쉬웠지만, 기능 완성 후의 성취감은 컸습니다. 앞으로는 Spring 공부와 Front-end 기술습득 필요성을 느꼈습니다. 이번 프로젝트 경험을 통해, 팀원과의 협업과 성과를 통해 새로운 도전에 대한 자신감이 높아졌습니다.

### 원규

처음에는 시간도 충분하고 여유있으면 무엇이든 다 해보자고 욕심을 부렸으나 결국에는 갈 수록 시간도 부족하다고 느끼고 여러모로 잘 되지 않는 부분들도 많아서 오히려 생각했던 것보다 부족하게 완성한 것 같아 아쉬웠습니다. 그래도 시간에 맞추어 주요 기능들을 잘 구현한 것 같아 만족스럽고 기회가 된다면 부족한 부분을 채워 더 완성도 있는 프로젝트로 만들어 보고 싶고 그러기 위해 더 노력해야 할 것 같습니다.



**THANK YOU**

