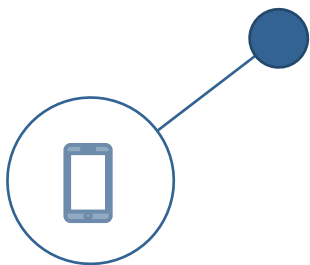


도서 관리 시스템

현인의 서재

이푸름

CONTENTS



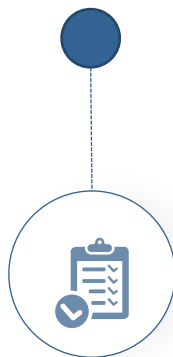
프로젝트 소개

- 프로젝트 목표
- 요구사항 정의



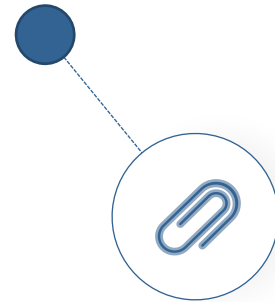
프로젝트 일정

- 로드맵



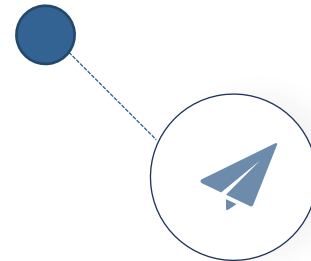
설계

- 아키텍처
- 화면설계
- DB 설계
- 통합 테스트 설계



구현

- 소스 코드



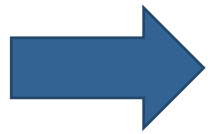
시연

- 통합 테스트

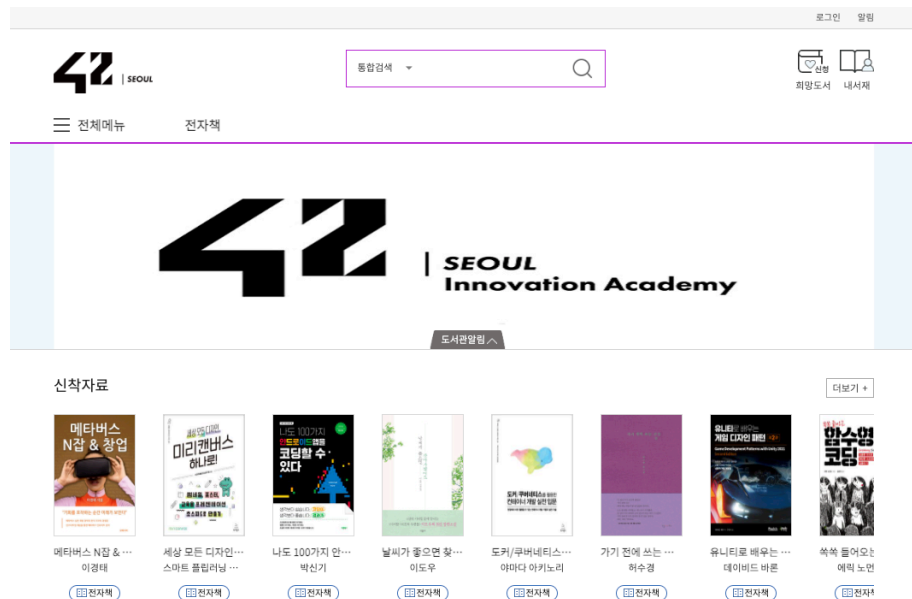
프로젝트 소개

프로젝트 목표 및 요구사항 정의

기업 내 도서 및 자료를 효율적으로 관리하기 위한 도서 관리 시스템



전자 도서관



전체메뉴

전자책

HOME > 전자책 > IT/프로그래밍

전자책

IT/프로그래밍

- 전체
- IT일반/교양
- 컴퓨터입문/활용
- 컴퓨터수험서
- 컴퓨터공학
- 데이터베이스/아키텍처
- OS/네트워크
- 코딩/프로그래밍/언어
- OA (사무 보조 프로그램)
- 웹사이트/홈페이지/블로그
- 그래픽/디자인
- 게임

✓ 최신순 인기순 제목순 저자순

전체도서는 총 517 개의 자료가 있습니다.

1 / 26 이동하기 < > 20개씩 보기



교보문고

TCP/IP 교과서: TCP/IP를 중심으로 네트워크 원리를 파악한다!
조 케서드 | 길벗 | 2023-03-10

메신저로 다른 사람과 대화를 주고받거나 유튜브 및 기타 플랫폼에서 실시간 스트리밍을 통해 소통하는 등 우리는 일상에서 당연하게 인터넷 네트워크를 활용하고 있다. 인터넷과 관련된 최신 기술이 계속 나오고 있지만 그중에서도...
[대출: 0/2 | 예약: 0]

대출

❤️ 찜



교보문고

누구나 자료 구조와 알고리즘
제이 원그로우 | 길벗 | 2023-02-10

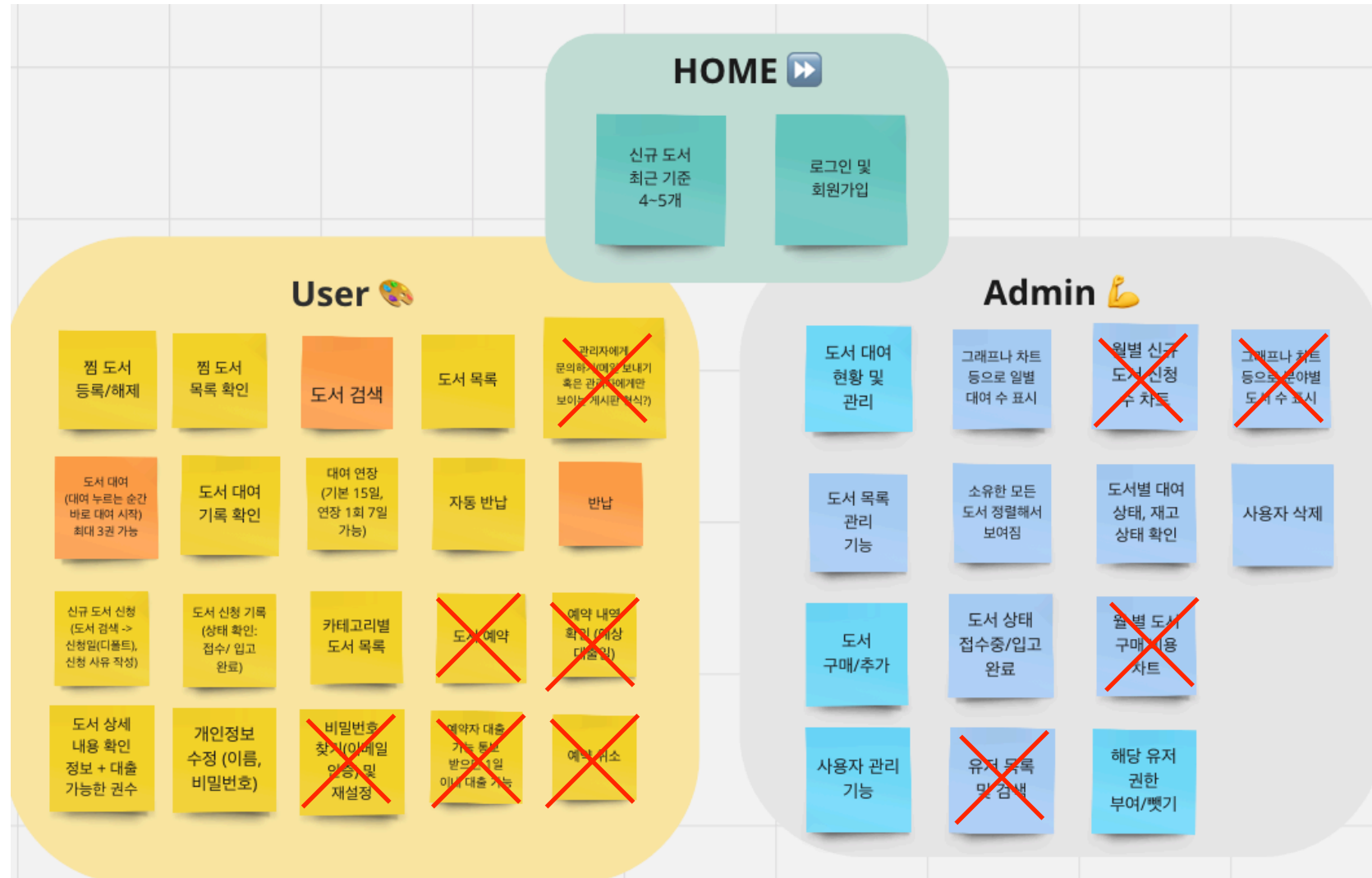
사칙 연산으로 복잡한 알고리즘을 쉽게 이해해보자 수학 용어와 전문 용어가 아니어도 이해한다 이 분야의 책은 대부분 컴퓨터 공학 전공자를 대상으로 쓰였거나 고등학교 수학을 잘 한다고 가정하고 있다. 쉽게 설명했다는 책도 전문...
[대출: 0/2 | 예약: 0]

대출

❤️ 찜

프로젝트 소개

프로젝트 목표 및 요구사항 정의



프로젝트 소개

프로젝트 목표 및 요구사항 정의

항목	요구사항 ID	요구사항명	기능 ID	기능명	상세 설명	필수 데이터
MEMBER	MEM01	로그인	MEM01_LOGIN01	로그인	도서 기능을 이용하기 위해서는 로그인 필수 로그인 성공 시, BOOK01로 이동 (로그인 안했을 시 MAIN01로 이동 가능)	아이디, 비밀번호
	MEM02	회원가입	MEM02_SIGNUP01	회원가입	회원가입	이름, 이메일(~@gmail.com 형식), 비밀번호, 비밀번호 확인
	MEM03	개인 정보 수정	MEM03_MODIFY01	개인 정보 수정	이름, 비밀번호 수정 현재 비밀번호 확인 후 수정 가능	현재 비밀번호, 변경할 이름 또는 비밀번호
	MEM04	로그아웃	MEM04_LOGOUT01	로그아웃	로그아웃 시 MAIN01로 이동	
MAIN	MAIN01	메인 페이지	MAIN01_NEW01	메인 페이지	신규 도서 4권 확인 가능, 현대오토 멤버 로고 클릭시 이동 가능	
BOOK	BOOK01	전체 도서 조회	BOOK01_BOOKLIST01	전체 도서 목록	사내 보유한 전체 도서 목록 조회 도서별 이미지와 제목, 저자, 출판사, 출판일, 도서 설명 제공 원인의 서재 클릭 시 이동 가능	
			BOOK01_BOOKLIST02	카테고리별 도서	선택한 카테고리별 도서 목록 확인 가능	특정 카테고리
			BOOK01_BOOKLIST03	도서 검색	제목 또는 저자 키워드로 검색 가능, BOOK02로 이동	도서 제목 또는 저자
			BOOK01_BOOKLIST04	찜 등록/해제	특정 도서 찜 등록 및 해제 가능	
			BOOK01_BOOKLIST05	도서 상세 정보	특정 도서 제목 또는 대출하러 가기 버튼을 통해 BOOK03로 이동 가능	
	BOOK02	도서 검색	BOOK02_SEARCH01	도서 검색 결과	사내 보유한 도서중 검색 키워드(제목 또는 저자)에 부합하는 도서 조회 가능 도서별 이미지와 제목, 저자, 출판사, 출판일, 도서 설명 제공	
			BOOK02_SEARCH02	도서 상세 정보	검색된 도서 결과의 제목 또는 대출하러 가기 버튼을 통해 BOOK03로 이동 가능	
	BOOK03	도서 상세 정보	BOOK03_BOOKDETAIL01	상세 도서 정보 확인	선택한 도서의 상세 정보를 확인 이미지, 제목, 카테고리, 저자, 출판사, 출판일, ISBN, 대출 가능한 수량 및 보유 수량, 도서 소개 정보 제공	
			BOOK03_BOOKDETAIL02	도서 대출	기본 7일 도서 대여 3권까지 대여 가능 회원의 상태에 따라 대출하기/대출중 확인 가능	
			BOOK03_BOOKDETAIL03	찜 등록/해제	도서 찜 등록 및 해제	
	BOOK04	대출 관리	BOOK04_LOANHISTORY01	대출 기록 조회	회원의 대출 기록 제공, 도서가 삭제되더라도 대출 기록은 확인 가능 이미지, 제목, 저자, 대출일, 반납일/반납예정일, 상태(반납 완료/반납/연장) 확인 가능	
			BOOK04_LOANHISTORY02	대출 기간 연장	대출일부터 1회에 한해, 7일 연장 가능	
			BOOK04_LOANHISTORY03	도서 반납	대출한 도서 반납	
			BOOK04_LOANHISTORY04	대출중인 도서	대출 기록 중 버튼을 통해 현재 대출중인 도서만 조회 가능	
	BOOK05	자주 도서 반납	BOOK05_DAILYRETURN01	도서 반납	매일 지정 대출 기간이 끝난 도서 강제 반납 처리	
	BOOK06	도서 신청	BOOK06_REQ01	신규 도서 신청	사내에 없는 도서 키워드 검색으로 선택 후, 신청 가능 신청 성공시 BOOK06으로 이동	ISBN, 도서 제목, 200자 이내의 신청 사유
	BOOK07	도서 신청 내역	BOOK07_REQHISTORY01	도서 신청 내역	신청 도서 내역 및 상태 확인 가능, 총 요청 횟수 확인 가능	
			BOOK08_REQHISTORY02	신청 내역 상세 정보	신청사유를 모달창으로 확인 가능 도서 제목과 처리상태, 신청일 정보 제공	
	BOOK08	찜한 도서	BOOK09_BOOKMARK01	찜한 도서 내역	찜한 도서 내역 조회 이미지, 제목, 저자, 출판사, 출판일, 도서 설명	
			BOOK10_BOOKMARK02	도서 상세 정보	도서 제목 또는 대출하러 가기 버튼을 통해 BOOK03로 이동 가능	
			BOOK11_BOOKMARK03	찜 해제	찜한 도서 해제 가능	
ADMIN	ADMIN01	사내 도서 목록 관리	ADMIN01_BOOKLIST01	사내 도서 목록	보유한 도서 목록 조회 가능	
			ADMIN01_BOOKLIST02	카테고리별 도서	선택한 카테고리별 도서 목록 확인 가능	
			ADMIN01_BOOKLIST03	도서 검색	제목 또는 저자 키워드로 검색 가능	도서 제목 또는 저자
			ADMIN01_BOOKLIST04	재고 수량 변경	재고 수량 변경	수량
			ADMIN01_BOOKLIST05	도서 삭제	도서 삭제 (도서가 삭제되어도, 회원은 해당 도서를 대출했던 이력, 신청한 이력은 확인 가능)	
	ADMIN02	대출 현황	ADMIN02_LOANSTATUS01	대출 내역	모든 대출 내역 최근 순으로 조회 가능 상태(대출중/연장중/반납완료), 이용자 이름, 제목, 저자, 대출일, 반납일/예정일 정보 제공	
			ADMIN02_LOANSTATUS02	5일간 대출 빈도수	5일간 대출 빈도수 그래프 제공	
	ADMIN03	도서 요청 처리	ADMIN03_REQSTATUS01	도서 요청 목록	도서 추가 요청된 목록 확인 가능 요청자, 제목, 저자, 신청일, 상태(요청중/입고완료), 모달창을 통해 상세 도서 정보와 신청 사유 확인 가능	
			ADMIN03_REQSTATUS02	요청된 도서 입고 처리	요청된 도서를 입고 수량 입력과 카테고리 선택을 통해 입고시키기가 가능 동일한 도서 요청건 모두 입고 완료 처리	입고 수량, 카테고리
	ADMIN04	신규 도서 추가	ADMIN04_NEWBOOK01	신규 도서 추가	추가할 도서를 키워드 검색 및 선택을 통해 추가 가능	입고 수량, 카테고리
	ADMIN05	회원 관리	ADMIN05_MEMBER01	회원 목록	회원 목록 조회 가능(신규 가입일 기준 정렬) 이름, 아이디, 권한, 가입 날짜 정보 제공	
			ADMIN05_MEMBER02	회원 정보 수정	이름, 권한 수정 가능	이름 또는 권한
			ADMIN05_MEMBER03	회원 삭제	회원 삭제 (회원 삭제시 대출 이력, 도서 추가 요청 이력, 찜 이력 모두 삭제)	

프로젝트 소개

프로젝트 목표 및 요구사항 정의

회원

메인 페이지

로그인

회원 가입

신규 도서

전체 도서

도서 정보

카테고리

찜

검색

대출 가능 여부

도서 상세

상세 정보

대출 하기

찜

대출 기록

대출 정보

반납

연장

도서 요청

신청

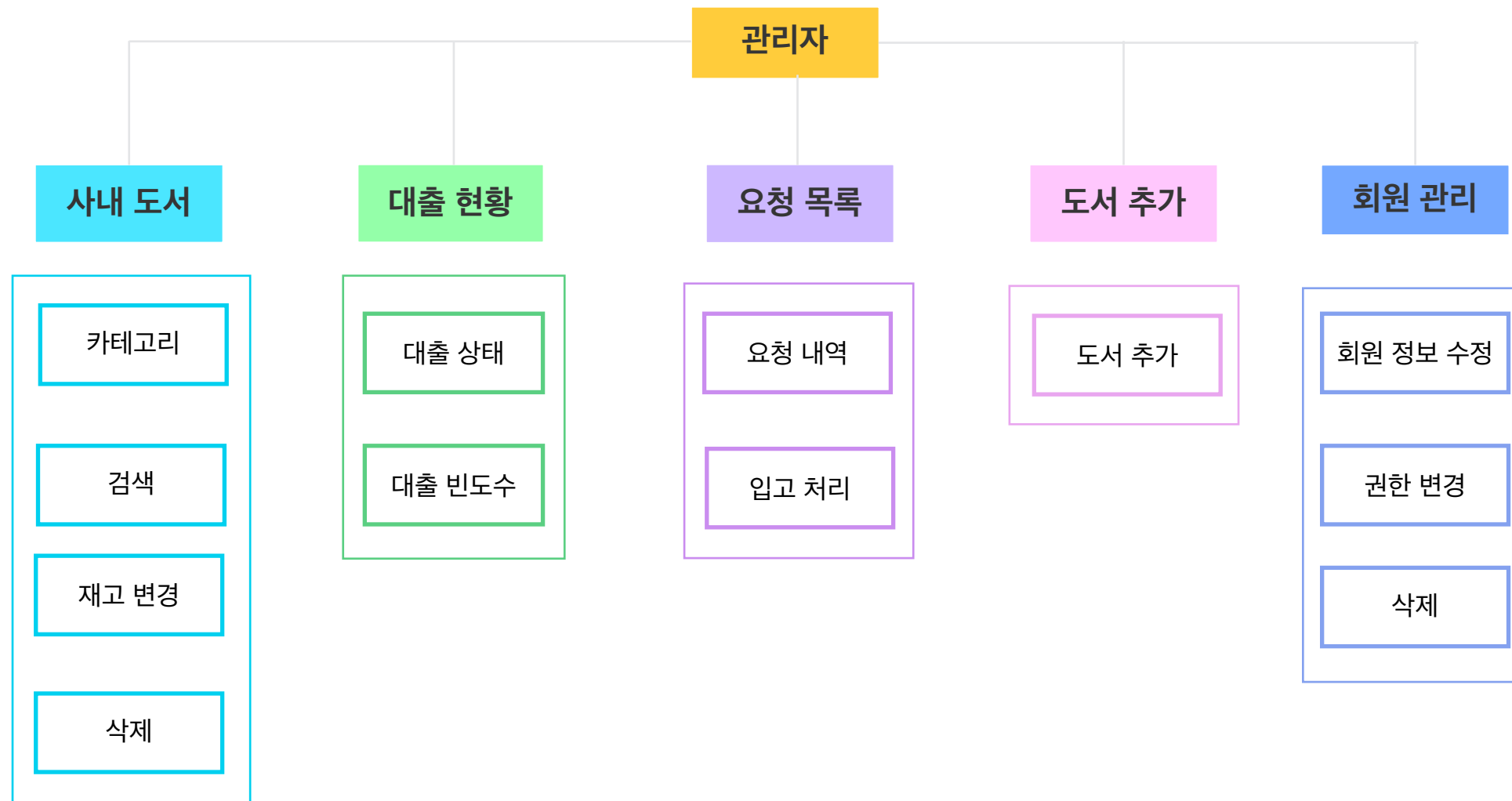
요청 기록

마이 페이지

개인 정보 수정

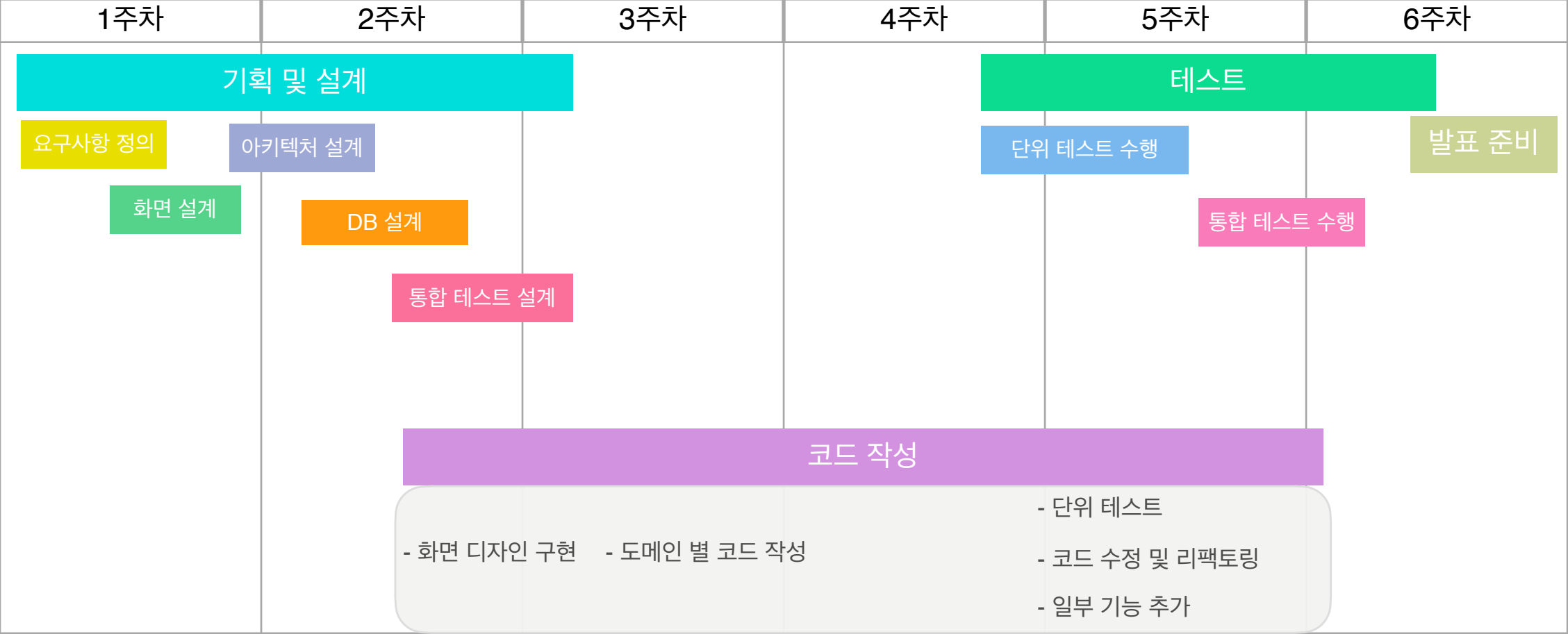
프로젝트 소개

프로젝트 목표 및 요구사항 정의



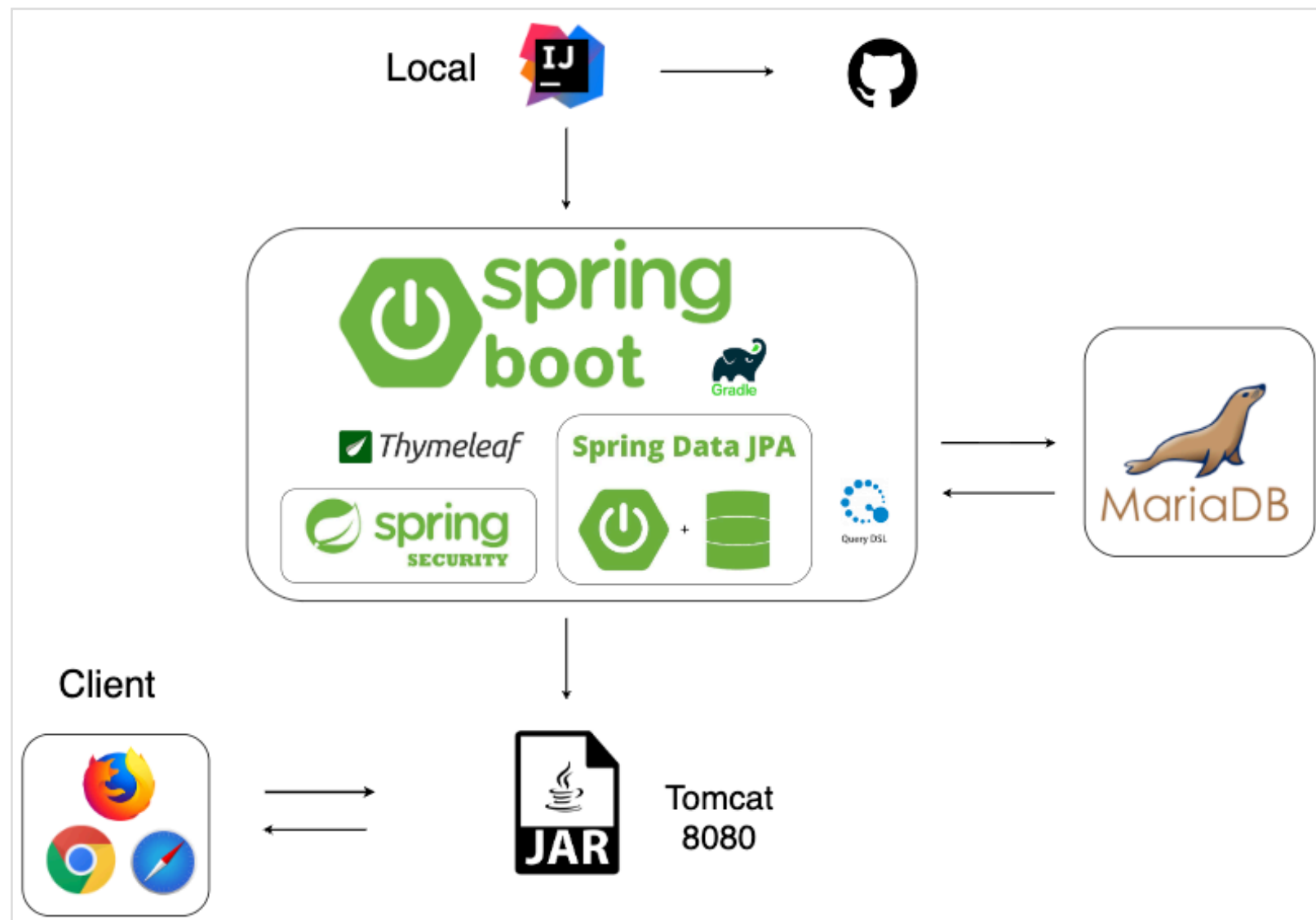
프로젝트 일정

로드맵



설계

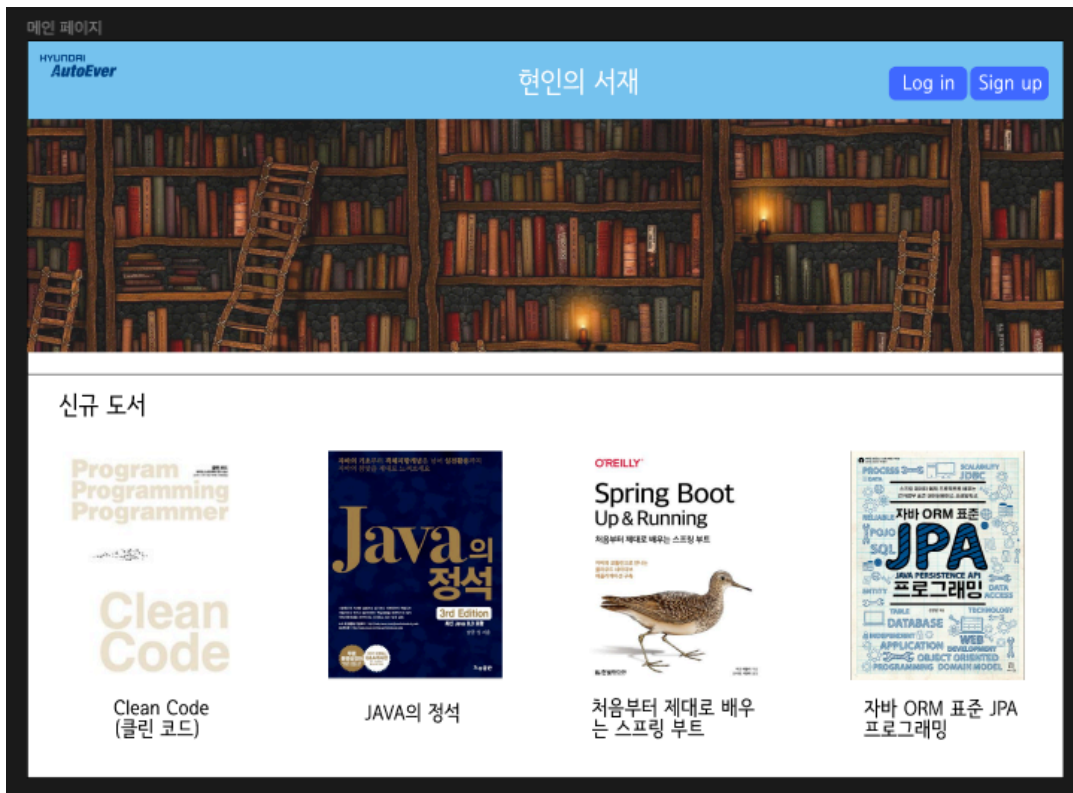
아키텍처



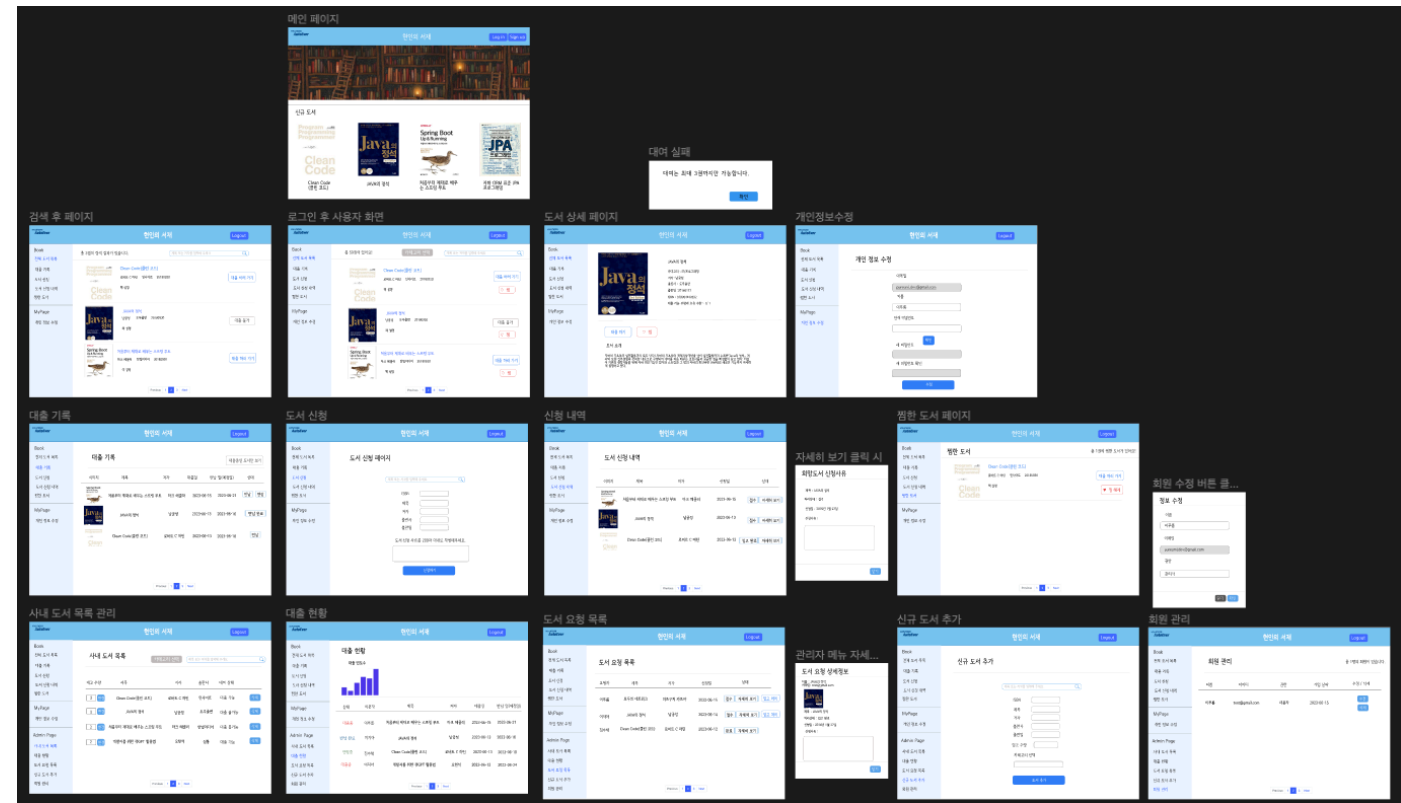
설계

화면 설계

-메인 페이지

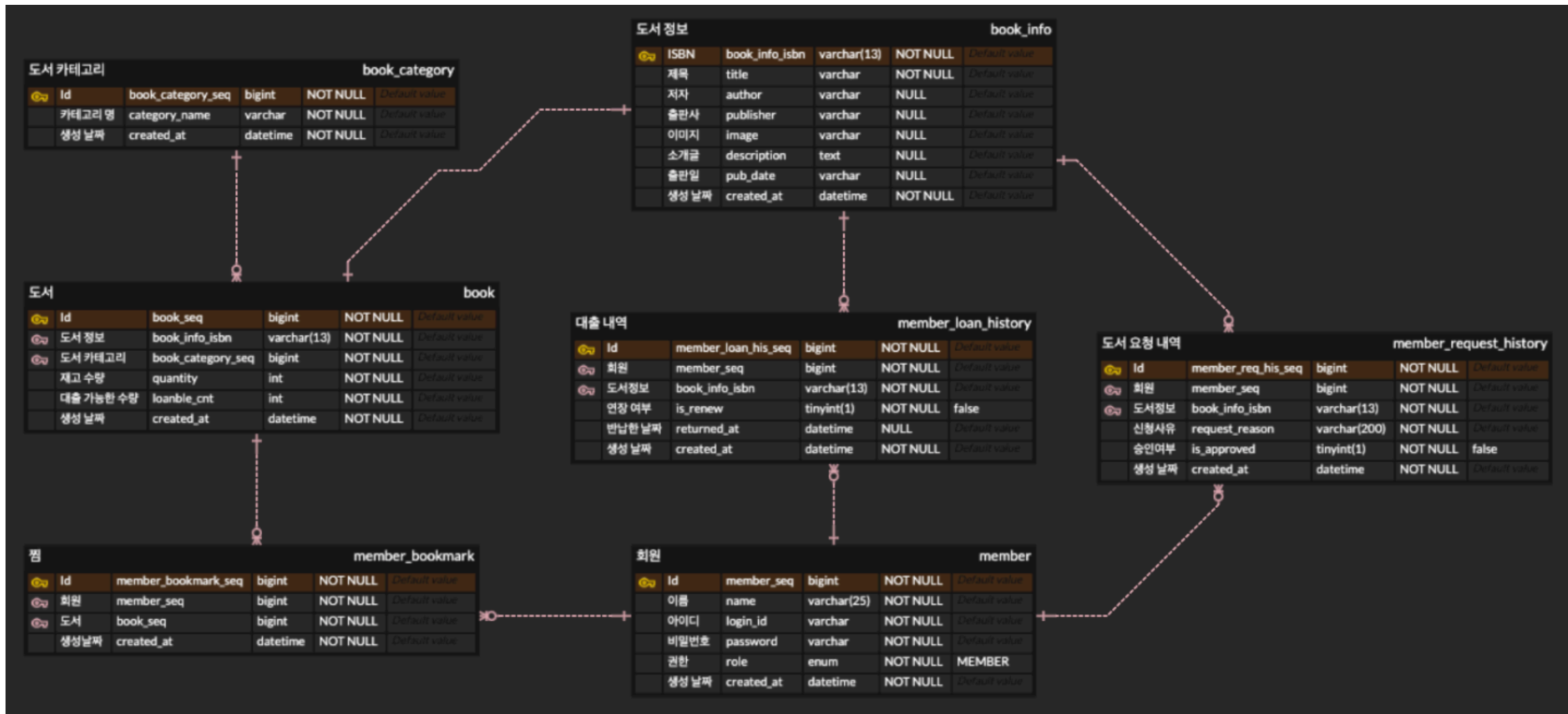


-Figma를 사용한 화면 설계



설계

Database



설계

통합 테스트 시나리오

분류			케이스	체크리스트	결과
1depth	2depth	3depth			
회원가입	이름,이메일, 비밀번호, 비밀번호 확인 입력		이름이 한글 또는 영어가 아닌경우	이름은 한글 또는 영어로만 입력해야 합니다. 메시지가 뜨는가?	○
			이름이 1-25자 이내가 아닌 경우	이름은 1~25자 이내여야 합니다. 메시지가 뜨는가?	○
			이메일이 ~@gmail.com 형식이 아닌경우	이메일은 @gmail.com 형식이어야 합니다. 메시지가 뜨는가?	○
			비밀번호가 6-20자 이내가 아닌 경우	비밀번호는 6~20자 이내여야 합니다. 메시지가 뜨는가?	○
			비밀번호와 비밀번호 확인이 일치하지 않는 경우	비밀번호가 일치하지 않습니다. 메시지가 뜨는가?	○
			이메일이 이미 존재하는 이메일인 경우	이미 존재하는 아이디입니다. 메시지가 뜨는가?	○
			값들이 유효한 경우	회원가입이 정상적으로 진행되는가?	○
			취소 버튼 클릭 시	전체 도서 목록 페이지로 이동하는가?	○
로그인	이메일, 비밀번호 입력		계정을 잘못 입력한 경우	아이디 또는 비밀번호가 맞지 않습니다. 다시 확인해 주세요. 메시지가 뜨는가?	○
			비밀번호가 맞지 않는 경우	아이디 또는 비밀번호가 맞지 않습니다. 다시 확인해 주세요. 메시지가 뜨는가?	○
			로그인 버튼 클릭 시	로그인이 정상적으로 진행되는가?	○
				전체도서목록 페이지로 이동하는가?	○
전체 도서 조회	전체 도서 목록 화면		전체 도서 목록 메뉴 클릭 시	이미지, 제목, 저자, 출판사, 출판일, 도서 소개 모든 정보가 올바르게 표시되는가?	○
			대출하기 버튼 클릭/ 도서 제목 클릭 시	전체 도서의 수량이 올바르게 표시되는가?	○
				사용자의 찜 등록 여부에 따라 찜/찜 해제 버튼이 올바르게 표시되는가?	○
			대출 가능한 수량이 없는 경우	도서 상세페이지로 이동하는가?	○
			찜 등록/해제 버튼 클릭 시	대출불가 버튼이 뜨는가?	○
				각 요청에 맞게 처리되며, 처리된 후 반대 버튼으로 변경되는가?	○
	키워드 입력		올바른 키워드 검색 시	해당 키워드가 제목 또는 저자에 해당하는 도서 정보가 올바르게 보이는가?	○
				검색한 키워드가 올바르게 표시되는가?	○
				키워드 검색 결과의 도서 수량이 올바르게 표시되는가?	○
				대출하러가기 버튼 또는 도서의 제목을 눌렀을때 도서 상세페이지로 이동하는가?	○
	카테고리 선택		검색어가 공백이거나 입력값이 없는 경우	대출 가능한 수량이 없는 경우, 대출불가의 비활성화 버튼이 뜨는가?	○
				검색어를 입력해주세요. 알림창이 뜨는가?	○
			카테고리 버튼 클릭 시	보유한 카테고리가 보이는가?	○
			특정 카테고리 클릭 시	해당 카테고리의 이미지, 제목, 저자, 출판사, 출판일, 도서 소개 모든 정보가 올바르게 표시되는가?	○
				카테고리 목록에 전체 목록이 생기는가?	○
			찜 등록/해제 버튼 클릭 시	사용자의 찜 등록 여부에 따라 찜/찜 해제 버튼이 올바르게 표시되는가?	○
				각 요청에 맞게 처리되며, 처리된 후 반대 버튼으로 변경되는가?	○
			대출하기 버튼 클릭/ 도서 제목 클릭 시	해당 카테고리와 페이지 번호가 그대로 유지되는가?	○
				도서 상세페이지로 이동하는가?	○
페이지네이션			대출 가능한 수량이 없는 경우	대출불가 버튼이 뜨는가?	○
			Previous 버튼 클릭시(첫번째 페이지인 경우)	비활성화 되어있는가?	○
			Previous 버튼 클릭시	이전 페이지로 이동하는가?	○
			페이지 번호가 10개 이상인 경우	10개 단위로 나오는가?	○
			페이지 번호가 10개 미만인 경우	있는 번호 페이지까지만 번호가 나오는가?	○
			값이 없는 경우	페이지번호가 안뜨는가?	○
			Next 버튼 클릭시	다음 페이지로 이동하는가?	○
			Next 버튼 클릭시(마지막 페이지인 경우)	비활성화 되어있는가?	○

설계

통합 테스트 시나리오

관리자-신규 도서 추가	신규 도서 추가 화면	도서 선택 모달 창	검색 버튼 클릭 시	도서 선택 모달창에 뜨는가?	O
			공백 또는 키워드 입력 없이 검색 버튼 클릭 시	키워드를 입력해주세요. 알림창이 뜨는가?	O
			검색 버튼 클릭 시	입력한 키워드에 부합하는 도서가 조회되는가?	O
			선택 버튼 클릭 시	모달창이 닫히며, 선택한 도서의 정보가 신규 도서 추가 화면에 올바르게 표시되는가?	O
				입고 수량을 입력할 수 있는 칸과 카테고리 선택, 신청하기 버튼이 뜨는가?	O
		신청 버튼 클릭	입고 수량 미입력 시	입고 수량을 작성하라는 메시지가 뜨는가?	O
			입고 수량에 유효하지 않은 값을 입력 시	입고 수량을 작성하라는 메시지가 뜨는가?	O
			카테고리 선택 안했을 시	카테고리를 선택해주세요. 알림창이 뜨는가?	O
			이미 사내에서 보유한 도서인 경우	이미 존재하는 책입니다. 알림창이 뜨는가?	O
			올바른 입력 상태인 경우	도서가 추가되었습니다. 알림창이 뜨며, 올바르게 추가 되었는가?	O
관리자-회원 관리	회원 관리 화면	수정 버튼 클릭	회원 관리 메뉴 클릭 시	가입된 총 회원의 수가 표시되는가?	O
				회원들의 이름, 아이디, 권한, 가입 날짜를 올바르게 조회할 수 있는가?	O
				회원의 이름, 아이디, 권한 정보를 올바르게 담은, 정보 수정 모달창이 뜨는가?	O
			아이디 부분 클릭 시	아이디는 수정 불가능한가?	O
			변경 없이 확인 버튼 클릭 시	수정된 정보가 없습니다. 알림창이 뜨는가?	O
		삭제 버튼 클릭 시	이름을 한글과 영어 이외의 값으로 수정하려는 경우	이름은 한글 또는 영어로만 입력해야 합니다. 알림창이 뜨는가?	O
			이름이 1-25자 이내가 아닌 경우	이름은 1~25자 이내여야 합니다. 알림창이 뜨는가?	O
			권한을 변경한 경우	권한 변경이 올바르게 진행되는가?	O
			취소 버튼 클릭 시	이전 상태로 되돌아 가는가?	O
				대출중인 도서는 자동 반납 처리되며, 회원의 대출 이력과, 신규 도서 요청 내역 모두 삭제됩니다. 정말로 삭제하시겠습니까? 메시지의 알림창이 뜨는가?	O
		페이지네이션		회원의 대출 내역, 도서 요청 목록, 찜한 기록 모두 사라지며 회원 삭제가 잘 이루어지는가?	O
				대출중인 도서가 있었다면 대출 가능한 수량이 올바르게 증가하는 가?	O
				요청한 도서 중 입고되지 않고, 다른 회원의 요청 없었던 도서인 경우, bookInfo도 사라지는가?	O
				요청한 도서 중, 다른 회원도 요청한 도서인 경우 bookInfo가 사라지지 않는가?	O
				요청한 도서 중 입고된 도서는 사라지지 않는가?	O
URL 접근 제한	관리자 페이지	회원 페이지	권한 없는 사용자가 접근을 시도한 경우	죄송합니다. 권한이 없어 접근 불가능한 페이지입니다. 메시지를 담은 페이지가 뜨며, 접근 불가능 한가?	O
			로그인 되지 않은 사용자가 서비스 이용을 시도한 경우	로그인페이지가 뜨며, 접근 불가능한가?	O
	없는 페이지		존재하지 않는 접근을 시도한 경우	죄송합니다.요청하신 페이지를 찾을 수 없습니다. 방문하시려는 페이지의 주소가 잘못 입력되었거나,페이지의 주소가 변경 혹은 삭제되어 요청하신 페이지를 찾을 수 없습니다. 입력하신 주소가 정확한지 다시 한번 확인해 주시기 바랍니다. 메시지를 담은 페이지가 뜨는가?	O

구현

자동 반납 처리

```
@Slf4j
@EnableScheduling
@RequiredArgsConstructor
@Component
public class BookReturnScheduler {

    private final BookService bookService;

    /* pupureum */
    @Scheduled(cron = "0 0 0 * * *") // 매일 자정(0시 0분 0초)에 실행
    public void processBookReturnAtMidnight() {
        LocalDateTime scheduledAt = LocalDate.now().atTime(LocalTime.MIDNIGHT);
        log.info("processBookReturn scheduled at {}", scheduledAt);

        try {
            int returnedSize = bookService.processDailyBookReturn(scheduledAt);
            log.info("SUCCESS processBookReturnAtMidnight: {} books returned", returnedSize);
        } catch (Exception e) {
            log.error("ERROR processBookReturnAtMidnight error: {}", e.getMessage());
        }
    }
}
```

```
/**
 * 스케줄러를 통해 매일 자정 대출 기간이 끝난 도서 반납을 처리합니다.
 *
 * @param scheduledAt 일정 시간
 * @return 처리된 연체된 대출 기록의 수
 * @throws NoSuchElementException 대출 기록에 해당하는 도서가 존재하지 않는 경우
 * @throws IllegalStateException 도서의 대출 가능한 수가 올바르지 않은 경우
 */
/* pupureum */
@Override
@Transactional
public int processDailyBookReturn(LocalDateTime scheduledAt) {
    // 요청된 시간 기준으로 연체된 대출 기록 조회
    List<MemberLoanHistory> overdueHistory = memberLoanHisRepository.searchOverdueHistory(
        LoanHistorySearchCondition
            .builder()
            .time(scheduledAt)
            .build());

    // 연체된 기록이 있는 경우
    if (!overdueHistory.isEmpty()) {
        log.info("overdueHistory exist");

        // 중복된 도서를 그룹화하여 수량 계산 및 해당 기록 반납 처리
        Map<String, Integer> bookInfoCount = processLoanHistory(overdueHistory, scheduledAt);

        // 도서 정보를 통해 반납 처리한 도서 조회
        List<Book> booksToUpdate = bookRepository.findByBookInfoIsbnIn(bookInfoCount.keySet());

        // 반납 처리한 대출 기록에 존재하지 않는 도서가 있는 경우 예외 처리
        validateBookInfoCount(booksToUpdate, bookInfoCount);

        // 반납 처리한 도서의 대출 가능한 수량 증가
        increaseLoanableCnt(booksToUpdate, bookInfoCount);

        memberLoanHisRepository.saveAll(overdueHistory);
        bookRepository.saveAll(booksToUpdate);
    }

    return overdueHistory.size();
}
```

```
public interface BookRepositoryCustom {  
    1 implementation  👤 pupureum  
    Page<Book> search(BookSearchCondition condition, Pageable pageable);  
}
```

```
@RequiredArgsConstructor  
@Repository  
public class BookRepositoryCustomImpl implements BookRepositoryCustom {  
  
    private final JPAQueryFactory queryFactory;  
  
    👤 pupureum *  
    @Override  
    public Page<Book> search(BookSearchCondition condition, Pageable pageable) {  
        List<Book> results = queryFactory  
            .select(book)  
            .from(book)  
            .leftJoin(book.bookInfo, bookInfo)  
            .where(  
                searchExpression(condition)  
            )  
            .orderBy(sortBook(pageable))  
            .offset(pageable.getOffset())  
            .limit(pageable.getPageSize())  
            .fetch();  
  
        JPAQuery<Long> countQuery = queryFactory  
            .select(book.count())  
            .from(book)  
            .leftJoin(book.bookInfo, bookInfo)  
            .where(  
                searchExpression(condition)  
            )  
            .fetchOne();  
  
        // results 와 pageable 를 확인하여 상황에 따라 count 쿼리 호출  
        return PageableExecutionUtils.getPage(results, pageable, countQuery::fetchOne);  
    }  
}
```


구현

QueryDSL

```
/**
 * 도서를 정렬하기 위한 OrderSpecifier 를 생성하는 메서드입니다.
 *
 * @param pageable Pageable 객체
 * @return OrderSpecifier 객체로서 적절한 정렬 조건을 포함
 */
1 usage new *
private OrderSpecifier<?> sortBook(Pageable pageable) {
    // Pageable 객체에 정렬 조건이 없는 경우
    if (pageable.getSort().isEmpty()) {
        return OrderByNull.getDefault();
    }

    // 정렬 조건 처리
    for (Sort.Order order : pageable.getSort()) {
        Order direction = order.getDirection().isAscending() ? Order.ASC : Order.DESC;

        switch (order.getProperty()) {
            case "createdAt" -> {
                return new OrderSpecifier<>(direction, book.createdAt);
            }
            case "title" -> {
                return new OrderSpecifier<>(direction, book.bookInfo.title);
            }
        }
    }

    return OrderByNull.getDefault();
}
```

```
/**
 * 검색 조건을 기반으로 BooleanBuilder 를 생성하는 메서드입니다.
 *
 * @param condition 검색 조건
 * @return 검색 조건에 따라 구성된 BooleanBuilder 객체를 반환
 */
2 usages 1 pupureum *
private BooleanBuilder searchExpression(BookSearchCondition condition) {
    BooleanBuilder builder = new BooleanBuilder();

    return builder
        .or(titleLike(condition.isTitle(), condition.getKeyword()))
        .or(authorLike(condition.isAuthor(), condition.getKeyword()))
        .and(categoryEq(condition.getCategoryId()));
}

1 usage 1 pupureum
private BooleanExpression titleLike(boolean title, String keyword) {
    return title ? book.bookInfo.title.containsIgnoreCase(keyword) : null;
}

1 usage 1 pupureum
private BooleanExpression authorLike(boolean author, String keyword) {
    return author ? book.bookInfo.author.containsIgnoreCase(keyword) : null;
}

1 usage 1 pupureum
private BooleanExpression categoryEq(Long categoryId) {
    return categoryId != null ? book.bookCategory.id.eq(categoryId) : null;
}
```

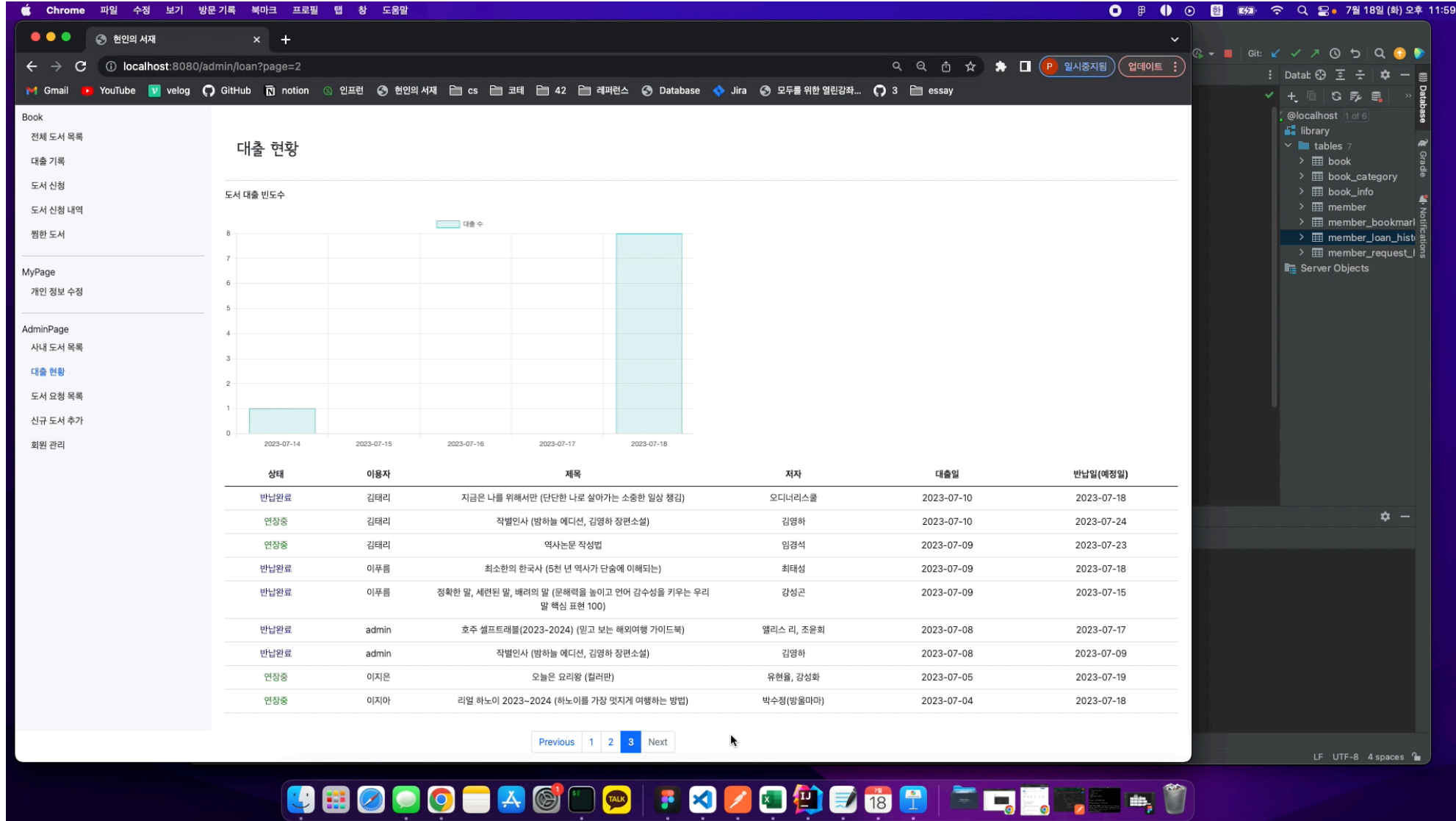

시연

자동 반납 기능

시연 링크

시연

자동 반납 기능



감사합니다

현인의 서재