

HELLO WORLD!

안녕하세요,

개발자 정혜원입니다 😎

목차

목차	
3p	인적사항
4p	프로젝트 소개 - 메모 프로젝트
5p	프로젝트 소개 - SNS 프로젝트
6~7p	프로젝트 소개 - space

space 프로젝트	
8p	space 프로젝트를 진행하며 느낀 점
9p	기술 정보
10p~	진행 단계
15p~	기능 소개

space 진행 단계	
11p	기획
12p	DB 설계
13p	URL 설계
14p	일정 설계

space 기능 소개	
15p~	회원가입 / 로그인
17p	홈
18p~	페이지
20p~	내 서재 / 읽어볼까
22p	독서노트
23p	독서캘린더
24p~	프로필 관리 / 회원탈퇴

인적사항



정혜원
Jeong Hyewon

📞 010 4853 3602
✉️ ffaikrw@gmail.com
🐱 <https://github.com/ffaikrw>

더 나은 사용자 경험을 위해 끊임없이 고민하고 학습하는 백엔드 개발자입니다.

JAVA와 SpringBoot를 이용하여 구축한 서비스를 더욱 효율적인 코드 작성을 위해, 보다 편리하고 보안 상 안전한 기능을 구현하기 위해 끊임없이 고민하고 학습하고 있습니다. 수많은 사용자가 이용하는 서비스를 안정적으로 개발하여 사용자에게 최대한 불편을 끼치지 않도록 계속해서 개선하고, 이를 전세계 사용자들이 찾는 서비스로 만드는 백엔드 개발자가 되는 것이 목표입니다.

Skill

JAVA
SpringBoot
MySQL
JavaScript
CSS
HTML
JQuery
JSTL

학력사항

2022.02
동덕여자대학교 일본어과 졸업
2018.03
동덕여자대학교 일본어과 입학
2018.02
복자여자고등학교 졸업

프로젝트 소개

Memo

정혜원 님 [로그아웃](#)

메모 목록 ❤

No.	제목	시간
8	파일 없는 메모	2022-03-22 15:44:53
7	파일 업로드 테스트	2022-03-22 15:36:27
5	다섯번째 메모	2022-03-21 17:09:03
4	test4	2022-03-21 16:34:40
3	test3	2022-03-21 16:34:26
2	test2	2022-03-21 16:33:35
1	test	2022-03-21 15:53:39

글쓰기

메모 프로젝트

회원가입을 하면 사용할 수 있는 메모 웹페이지입니다.
메모 작성 및 수정, 삭제가 가능하고 이미지 첨부도
가능합니다.



<https://github.com/ffaikrw/Memo>

Copyright 2022. Memo all rights reserved.

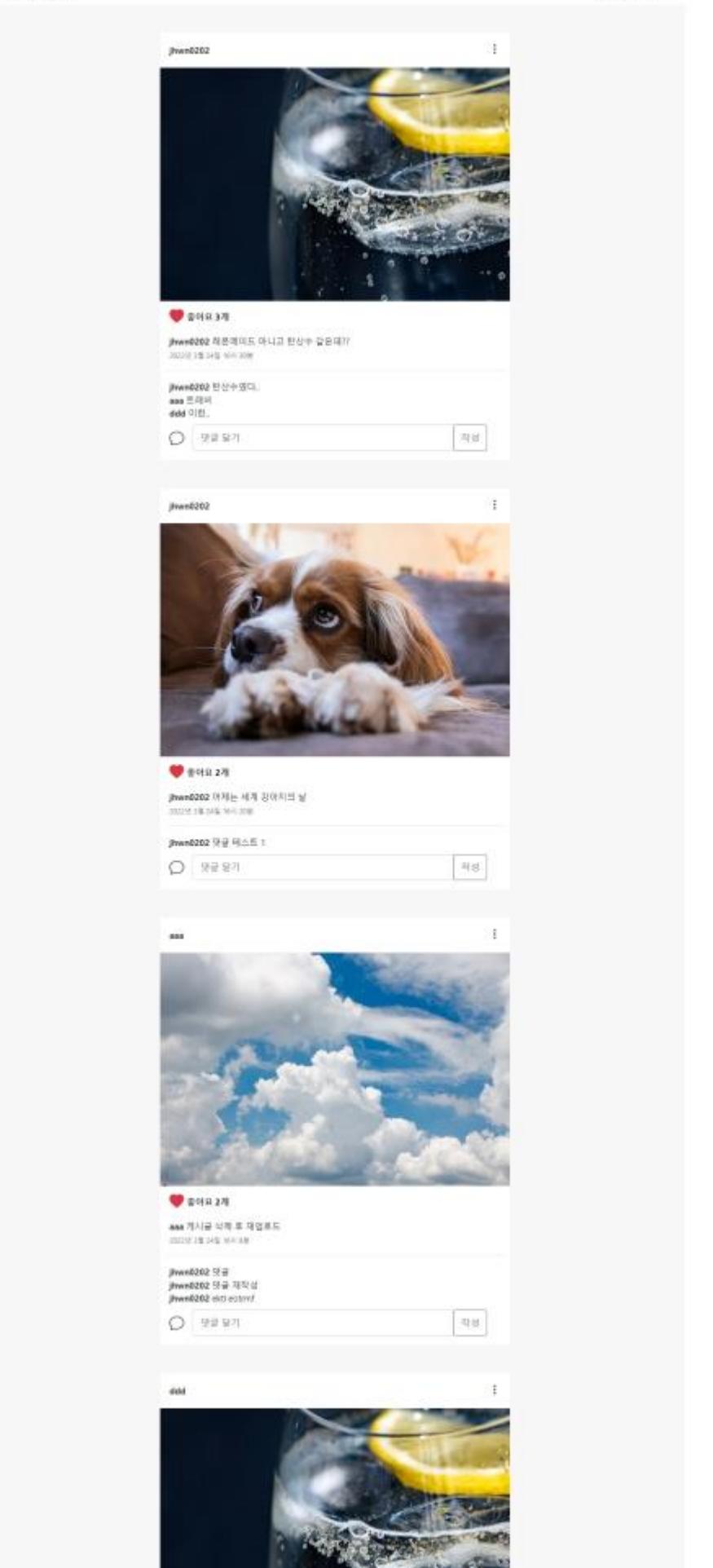
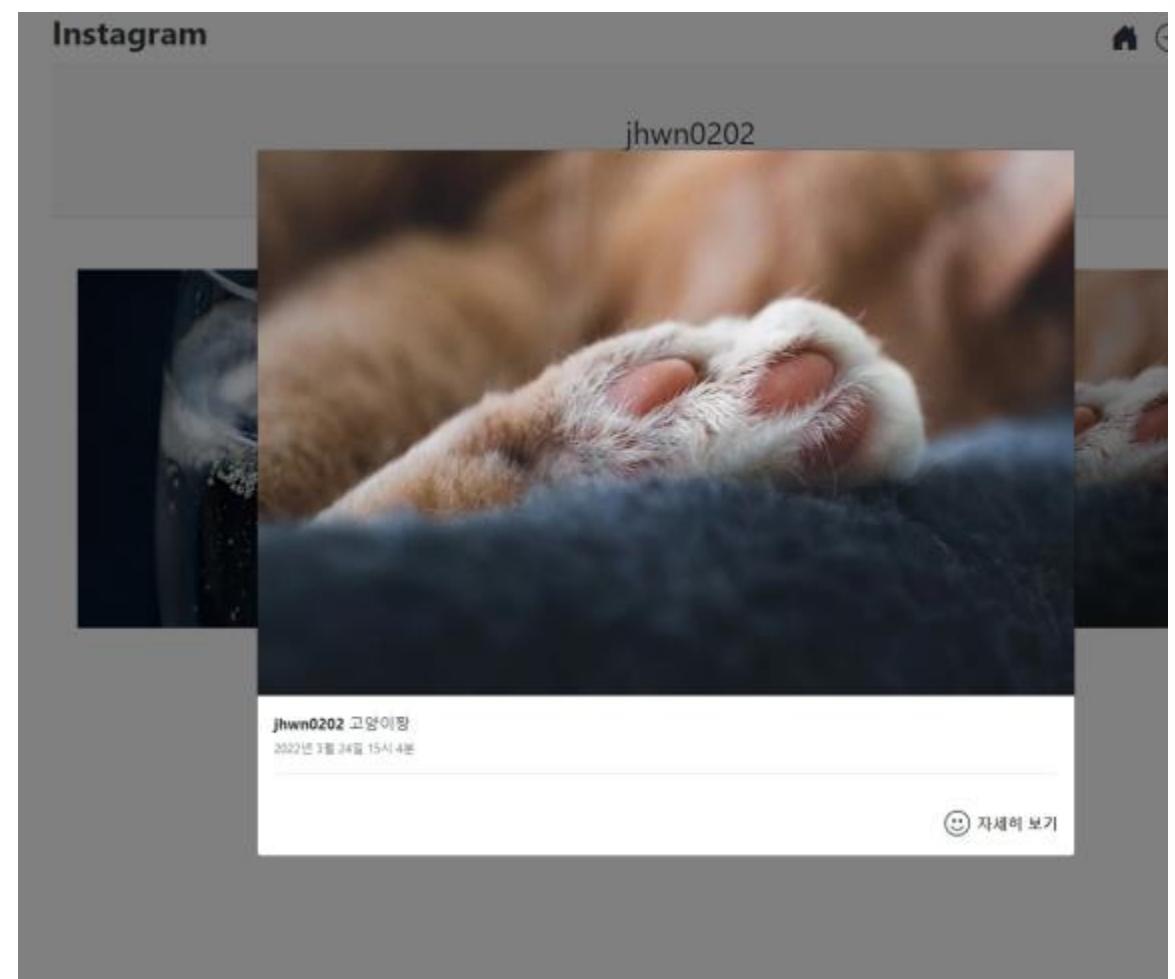
프로젝트 소개

SNS 프로젝트

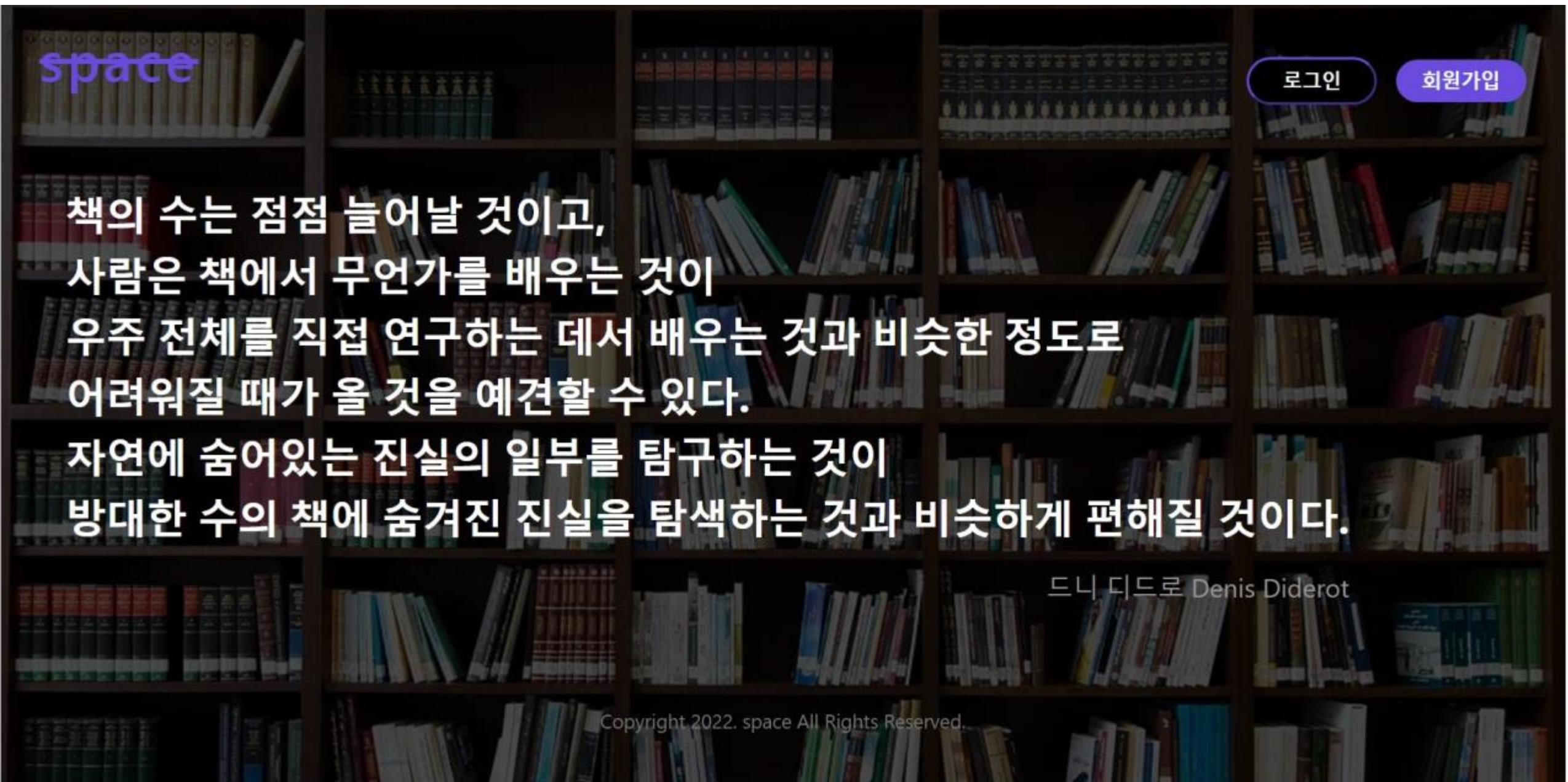
인스타그램과 유사하게 만든 SNS 웹페이지입니다.
로그인한 사용자는 게시글 작성, 다른 사용자의 게시글
에 좋아요, 댓글 작성하기 기능을 사용할 수 있습니다.
로그인하지 않은 사용자는 타임라인에서 SNS 사용자들
의 게시글을 보는 것만 가능합니다.



<https://github.com/ffaikrw/Instagram>



프로젝트 소개



소설 구독 플랫폼

—space—



<https://github.com/ffaikrw/space>



<http://52.79.141.125:8080/user/start>

테스트 계정(저장된 데이터 X)

: test@gmail.com / test

테스트 계정(저장된 데이터 O)

: content@gmail.com / content

* 한 줄 평은 구병모 <아가미> 상세정보에 들어가면 보실 수 있습니다.

홈 화면 접속 시 도서 오픈 API에 한 번에 많은 요청을 넣고 결과를 받아오기 때문에 홈 화면을 불러오는데 시간이 소요됩니다. 접속 시간을 줄이기 위해 기존에 한 번에 많은 도서 정보를 요청하고 응답 리스트의 인덱스 0번째 정보만 홈 화면에 가져오던 코드를 홈 화면 구성에 필요한 최소한의 도서 정보만을 요청하도록 수정했으며, 더 나은 방법을 찾기 위해 계속해서 고민 중에 있습니다.

프로젝트 소개

소설 구독 플랫폼 space



space는 넓은 우주를 유영하며 새로운 별을 찾아내듯, 사용자로 하여금 수많은 소설 속을 둘러보며 마음에 드는 소설을 찾아낼 수 있도록 돋는 소설 구독 플랫폼입니다.

화면 구성은 OTT 플랫폼 Watcha를, 기능은 도서 구독 플랫폼 밀리의 서재를 참고하였습니다. 도서 정보 및 검색은 온라인 도서 사이트 알라딘의 오픈 API를 사용하여 가져왔고, 알라딘 API에서 제공하는 신작 도서, 베스트셀러, 편집자 추천, 주목할만한 신작 리스트를 홈 화면 및 검색 화면에 배치해 사용자가 보다 손쉽게 다양한 소설을 탐색할 수 있도록 하였습니다.

또한 가져온 정보 중 ISBN을 통해 도서를 구분하여 사용자가 읽고 싶은 소설을 저장하는 '읽어볼까?' 기능과 다 읽은 소설을 저장하는 '내 서재' 페이지를 구현하였습니다. 특히 '내 서재'에 소설을 담으면 읽은 소설에 대한 감상이나 메모를 적을 수 있는 '독서 노트' 기능과 완독 일을 입력하면 달력에서 입력한 날짜에 어떤 도서를 읽었는지 볼 수 있는 '독서 캘린더' 기능을 사용할 수 있습니다. 독서 노트는 웹 에디터 summernote로 구현하였고 독서 캘린더는 캘린더 라이브러리 FullCalendar를 사용하였습니다.

읽어볼까 페이지와 내 서재 페이지에 소설을 저장하면 홈 화면에서도 접근할 수 있으며, 내 서재에 담은 소설의 작가들이 쓴 또 다른 소설을 확인할 수 있도록 '서재 속 작가들' 기능도 홈 화면에 구현하였습니다.

그 외에도 내 서재에 소설을 담으면 해당 도서를 추천할 수 있는 기능을 만들었고, 이 기능을 사용해 space 사용자들의 추천을 많이 받은 상위 10개 소설을 보여주는 'space TOP 10' 기능을 구현했습니다. 또한 소설을 선택하면 도서 상세 페이지로 이동하는데, 그곳에서는 다른 사용자가 작성한 한 줄 평을 볼 수도 있고 작성할 수도 있습니다.

프로젝트를 진행하며 느낀 점

**웹 개발에 있어 어떤 것을 알고 어떤 것을 모르는지,
현재 할 수 있는 것은 어디까지이고 어떻게 이를 발전시킬 수 있는지 알게 되었습니다.**

개발을 시작하기 전, 프로젝트 기획 과정에서는 이걸 할 수 있을지 없을지는 생각하지 않고 만들고 싶은 것, 만들어보고 싶은 기능을 이것저것 넣어 기획했습니다. 그런 기획문서를 가지고 개발을 진행하다 보니 모든 게 다 기획한대로 술술 풀리지는 않았습니다. 생각보다 웹 개발에 있어 개념이 허술한 부분이 있었고, Controller와 BO의 역할도 명확한 기준 없이 모호하게 사용하고 있었습니다. 이렇게 두면 안되겠다 싶어서 웹 개발의 흐름에 대해 다시 생각해보고 Controller는 클라이언트의 요청을 BO에 전달만 해주는 곳, BO는 Controller에서 받은 요청을 처리하는 곳이란 점을 항상 염두에 두고 개발을 진행했습니다. 또한, 머릿속에서는 분명 될 것 같았는데 실제로 해 보니 쉽게 구현할 수 없었던 것들이 있었고 이런 것들을 해결하기 위해 검색을 많이 이용했습니다. 그 과정에서 같은 결론을 내기 위해 각기 다른 과정을 거치는 것을 보면서 많은 공부가 되었습니다. 그리고 프로젝트 진행 초반에는 사용하려고 하는 라이브러리, 프레임워크, 오픈 API에 대한 이해도가 낮은 상태에서 검색으로 나온 내용을 아무거나 적용해보고 결과가 원하는대로 나오지 않으면 그제서야 원인을 찾았는데, 이러한 과정을 몇 번 거쳐가며 검색 결과의 내용과 내가 적용한 결과가 어떻게 다른지, 왜 다른지에 대해서도 알게 되었고 공식 문서도 여러 번 읽으면서 현재 사용하고 있는 라이브러리, 프레임워크, 오픈 API 등의 사용법에 대한 이해가 높아졌습니다.

또한 프로젝트를 진행하면서 처음 보는 오류를 많이 마주했는데, 그 중 가장 인상 깊었던 오류은 예외처리가 빠져서 발생한 오류였습니다. 지금껏 예외처리는 저에게 있어 존재감이 크지 않았는데, 예외처리를 중요하게 여기지 않고 빠트리면 웹페이지를 서비스할 때 예외가 발생하여 페이지가 작동하지 않는 경우가 분명히 생길 것이고, 이는 결국 사용자의 입장에서 생각하지 않는 개발이란 점을 깨달았습니다.

기획한대로 구현하고 싶은 기능이 몇 있었으나 시간이나 제 개발 실력과 타협하게 되어 분명 아쉬운 점도 있습니다. 하지만 이는 남은 기능을 구현하는데 더 집중할 수 있는 동력이 되었고, 동시에 개발 공부를 더 많이 해야겠다는 생각을 들게 했습니다. 계속 노력하여 더 괜찮은 개발자가 되고 싶습니다.

기술 정보



GitHub

<https://github.com/ffaikrw/Space>



Tech Stack

JAVA

MySQL CSS

HTML JSTL AWS

JavaScript SpringBoot



Licence

알라딘 OPEN API

Copyright 1999 – 2022 Aladdin com
munications Inc.
[https://blog.aladin.co.kr/openapi/gr
oup/18264538](https://blog.aladin.co.kr/openapi/group/18264538)

Kakao Login API

[https://developers.kakao.com/docs/
latest/ko/kakaologin/rest-api](https://developers.kakao.com/docs/latest/ko/kakaologin/rest-api)
© Kakao Corp.

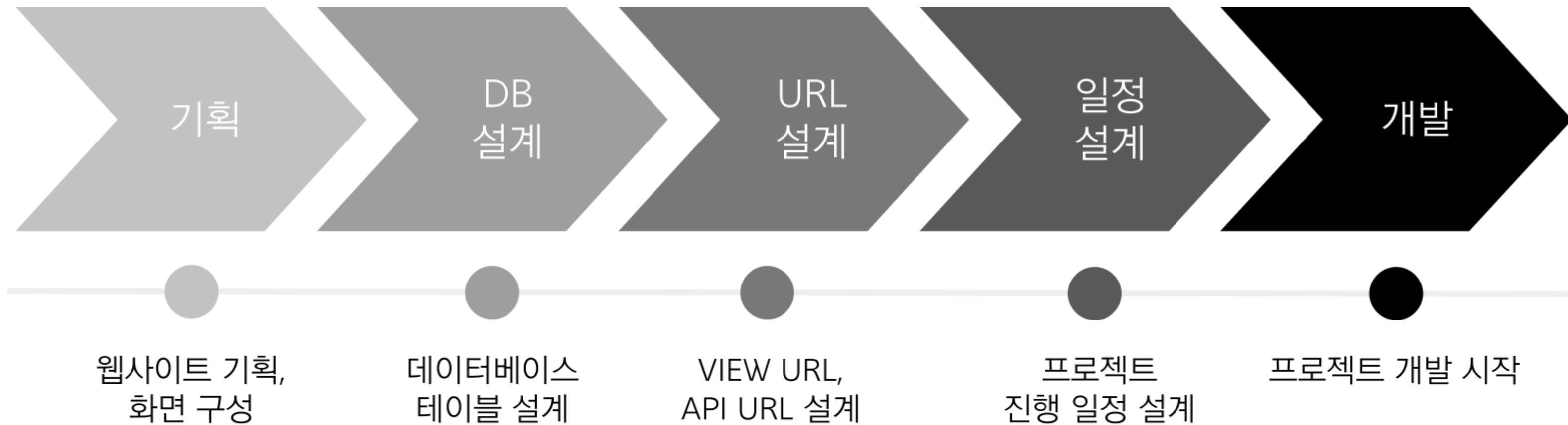
FullCalendar

FullCalendar Standard
MIT license
<https://fullcalendar.io/license>

summernote

The MIT License (MIT)
Copyright (c) 2015~ Summernote T
eam ([https://github.com/orgs/sum
mernote/people](https://github.com/orgs/sum
mernote/people))

프로젝트 진행 단계



웹사이트 기획,
화면 구성

데이터베이스
테이블 설계

VIEW URL,
API URL 설계

프로젝트
진행 일정 설계

프로젝트 개발 시작

프로젝트 진행 단계

The screenshot shows the homepage of the 'space' website. It features a dark background with colorful cards for different sections:

- Best Sellers:** A green card.
- Recommended Books:** A purple card with a small thumbnail and text: "일상에서 찾아낸 윤을 한 조각" and "이서수 <헬프 미 시스터>".
- Today's Top 10 Books:** A yellow card with a list of 6 books, each with a thumbnail and the title "책 제목".
- Books to Read:** A blue card with a list of 6 books, each with a thumbnail and the title "책 제목".
- Authors in the Library:** A row of 6 circular profile pictures of authors.
- Drama/Movie/Animation Original Books:** An orange card with a list of 6 books, each with a thumbnail and the title "책 제목".
- Books to Read Next?** A section at the bottom with the text "아직 고민 중이신가요?" and "읽어볼까 한 도서를 랜덤으로 감상해보세요!".

Navigation and search bars are visible on the left, and a footer with copyright information is at the bottom.

기획

기획 단계에서는 어떤 웹사이트를 제작할 것인지, 어떤 기능을 넣고 어떻게 페이지를 구성할 것인지를 고민하고 ovenapp을 이용하여 기획문서를 작성했습니다.

UI는 OTT 플랫폼 Watcha를, 세부 기능은 도서 구독 플랫폼 밀리의 서재를 참고하였습니다.

또한 초록색 글씨로 해당 기능을 어떤 방식으로 구현할지도 간략하게 작성했습니다.

The screenshot shows the search results for '하가시노 게이고' on the 'space' website. The results are displayed in a grid format:

작가명	작가명	작가명	작가명	작가명	작가명
방황하는 칼날	용의자 X의 혼신	몽환화	가면산장 살인사건	라플라스의 마녀	편지
작가명 1441	작가명 1441	작가명 1441	작가명 1441	작가명 1441	작가명 1441

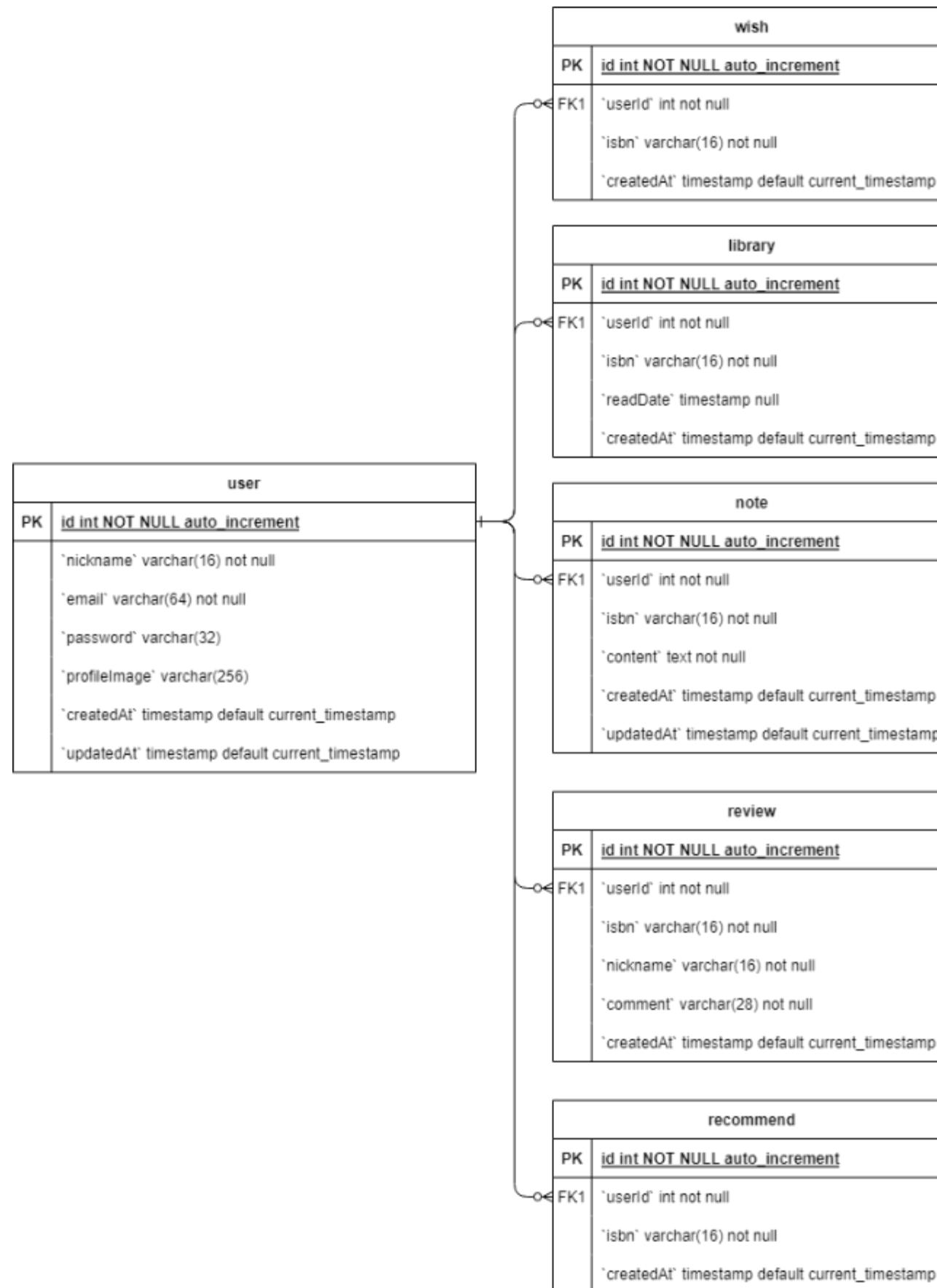
Below the main grid, there are two more rows of results:

작가명	작가명	작가명	작가명	작가명	작가명
연예의 행방	비밀	나미야 잡화점의 기적	눈보라 체이스	언어가 잠든 집	악의
작가명 1441	작가명 1441	작가명 1441	작가명 1441	작가명 1441	작가명 1441

A link "[전체 기획 문서 보기]" is located at the bottom right of the search results.



<https://ovenapp.io/view/PtfXvmHtrrSzcQVmzholFvYxcwRbjci/>



프로젝트 진행 단계

DB 설계

테이블명: library				
설명: 내 서재 테이블				
컬럼명	타입	NULL 여부	AUTO_INCREMENT	설명
id	int	N	Y	primary key, 서재에 담긴 책의 id
userId	int	N	N	사용자 아이디(userId)
isbn	int	N	N	도서 isbn
readDate	timestamp	N	N	완독 날짜
createdAt	timestamp	N	N	서재에 추가된 날짜

오픈 API를 통해 isbn을 받아서 DB에 저장

데이터베이스 설계는 구글 스프레드 시트를 사용하여 작성하였습니다.

사용자의 입력이 발생하는 기능을 기준으로 잡고 사용자 정보, 위시리스트, 내 서재, 독서노트, 추천, 한 줄 평으로 테이블을 나누어 설계했습니다. 또한, 서로 관계가 있는 테이블은 어떤 컬럼이 연결되어 있는지 행에 배경색을 주어 눈에 띄게 표시하였고, 도서 오픈 API를 사용하여 정보를 저장할 컬럼은 첨부한 이미지와 같이 하늘색으로 구분하였습니다.

[전체 설계 문서 보기]



<https://docs.google.com/spreadsheets/d/1WfN0UFY1LEcxX2-XBPeL47Vu4KMUNo1xD7KuGLP4xU/edit?usp=sharing>

프로젝트 진행 단계

URL 설계

View URL

제목	URL	parameter	설명
시작	/user/start		시작 화면
로그인	/user/sign_in		로그인 화면
회원가입	/user/sign_up		회원가입 화면
홈	/browse/home		전체 게시글 화면 로그인한 상태에서만 접근 가능
이번 주 신작	/browse/weekly_new		이번 주 신작 리스트 화면
베스트셀러	/browse/bestseller		베스트셀러 리스트 화면
편집자 추천 소설	/browse/editor_recommend		오늘의 추천 소설 리스트 화면
작가의 저서	/author	작가명	특정 작가의 저서 리스트
검색	/browse/search		검색 및 장르 화면 로그인한 상태에서만 접근 가능
검색 결과	/browse/search_result		검색 결과 화면 로그인한 상태에서만 접근 가능
내 서재	/library	userId	내 서재 화면
독서 노트 작성	/library/create_note	isbn	로그인한 상태에서만 접근 가능 독서 노트 작성 화면
독서 노트 보기	/library/note_view	isbn	특정 도서의 독서 노트 보기 화면
읽어볼까?	/wishlist		읽어볼까에 담은 도서 리스트 화면
도서 상세 정보	/book_info	isbn	특정 도서의 상세 정보 화면
힌 줄 평	/short_review	isbn	특정 도서의 힌 줄 평 리스트 화면
프로필 관리	/user/profile	userId	프로필 관리 화면 로그인한 상태에서만 접근 가능

API URL

1) 로그인

URL: /user/signin_api

method: post

parameter

parameter 이름

데이터 타입

NULL 가능 여부

설명

email

String

N

로그인 이메일

password

String

N

로그인 비밀번호

응답값(json)

{"result": "success"}

성공시

{"result": "fail"}

[전체 설계 문서 보기]



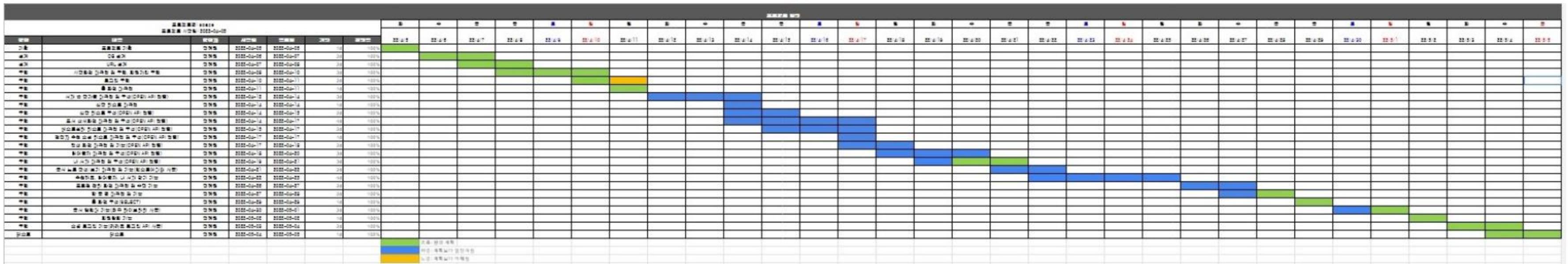
<https://docs.google.com/spreadsheets/d/1WfN0UFY1LEcxX2-XBPeL47Vu4KMUNo1dxD7KuGLP4xU/edit?usp=sharing>

URL 설계 시 view와 API에 요청되는 parameter와 요청 방식 등이 다르기 때문에 view URL과 API URL을 나누어 설계하였습니다.

view URL의 경우 받아야 하는 parameter도 함께 작성하였고, API URL은 URL과 더불어 요청 방식이 GET인지 POST인지, parameter명과 데이터 타입 및 NULL 여부, json에 저장하여 클라이언트에 전달할 응답값의 내용도 작성하였습니다.

프로젝트 진행 단계

일정 설계



기획 단계부터 프로젝트 테스트 과정까지의 전체 일정을 관리하기 위해 일정 설계를 진행하였습니다.

화면 마크업, 기능 구현 등으로 나누어 시작일 및 종료일, 기간, 진행률과 함께 작성하였고,

이에 맞춰 녹색으로 기간을 설정해두었습니다.

하나의 일정이 마무리 될 때마다 실제 시작한 날짜 및 종료된 날짜, 기간으로 수정하였고,

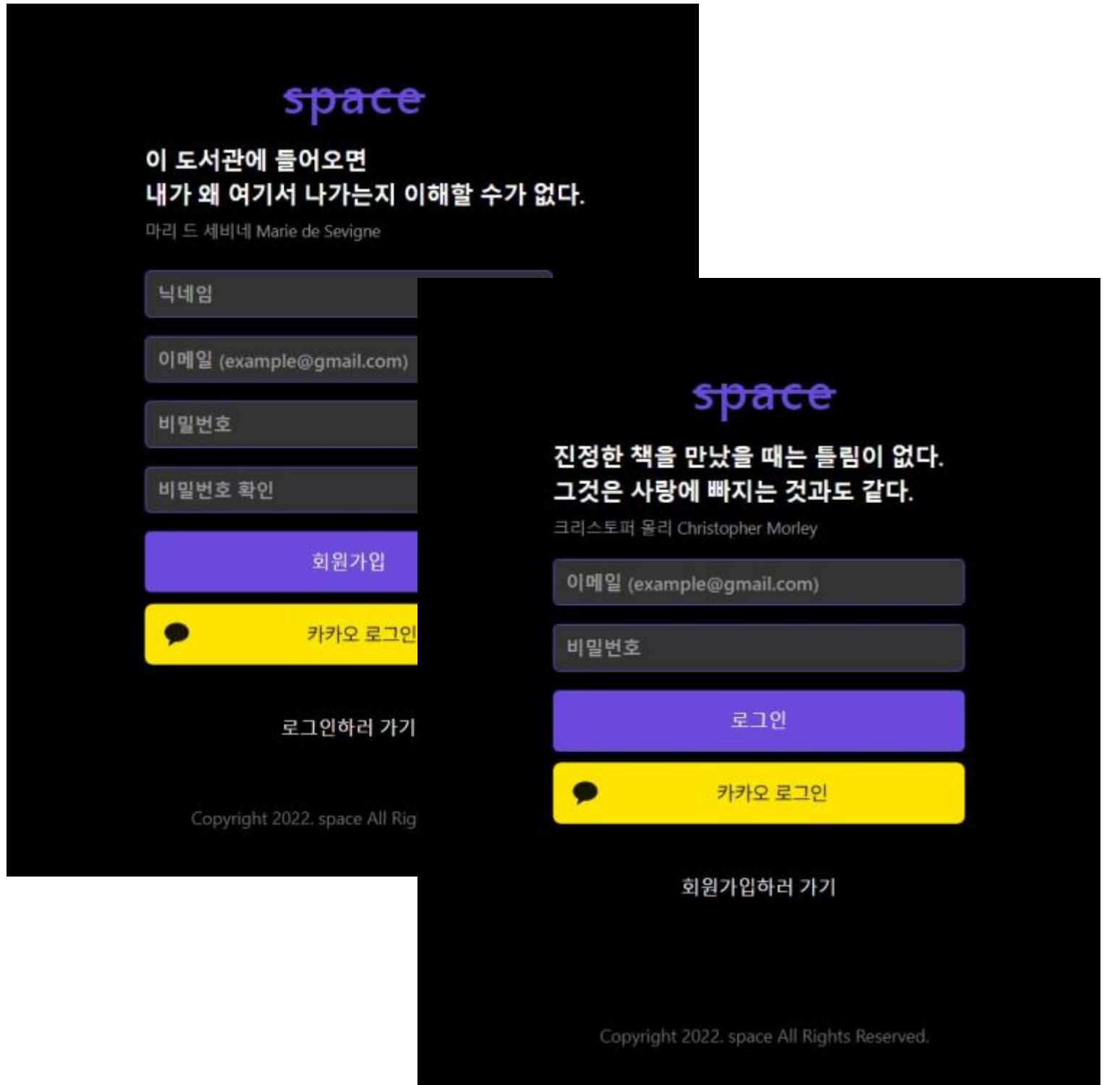
종료일이 예상보다 미뤄졌을 경우 노란색으로, 앞당겨졌을 경우 파란색으로 수정했습니다.

[전체 설계 문서 보기]



<https://docs.google.com/spreadsheets/d/1WfN0UFY1EcX2-XBPeL47Vu4KMUNo1dxD7KuGLP4xU/edit?usp=sharing>

기능 소개 회원가입/로그인



기능

회원가입

- 닉네임, 이메일은 입력하면 데이터베이스를 조회하여 중복되는 정보가 있는지 확인한 후 중복되면 회원가입을 막고, 중복되지 않으면 회원가입 가능하도록 구현했습니다. 또한 사용자가 입력한 비밀번호는 암호화메소드를 사용하여 그 결과를 DB에 저장하였습니다.

로그인

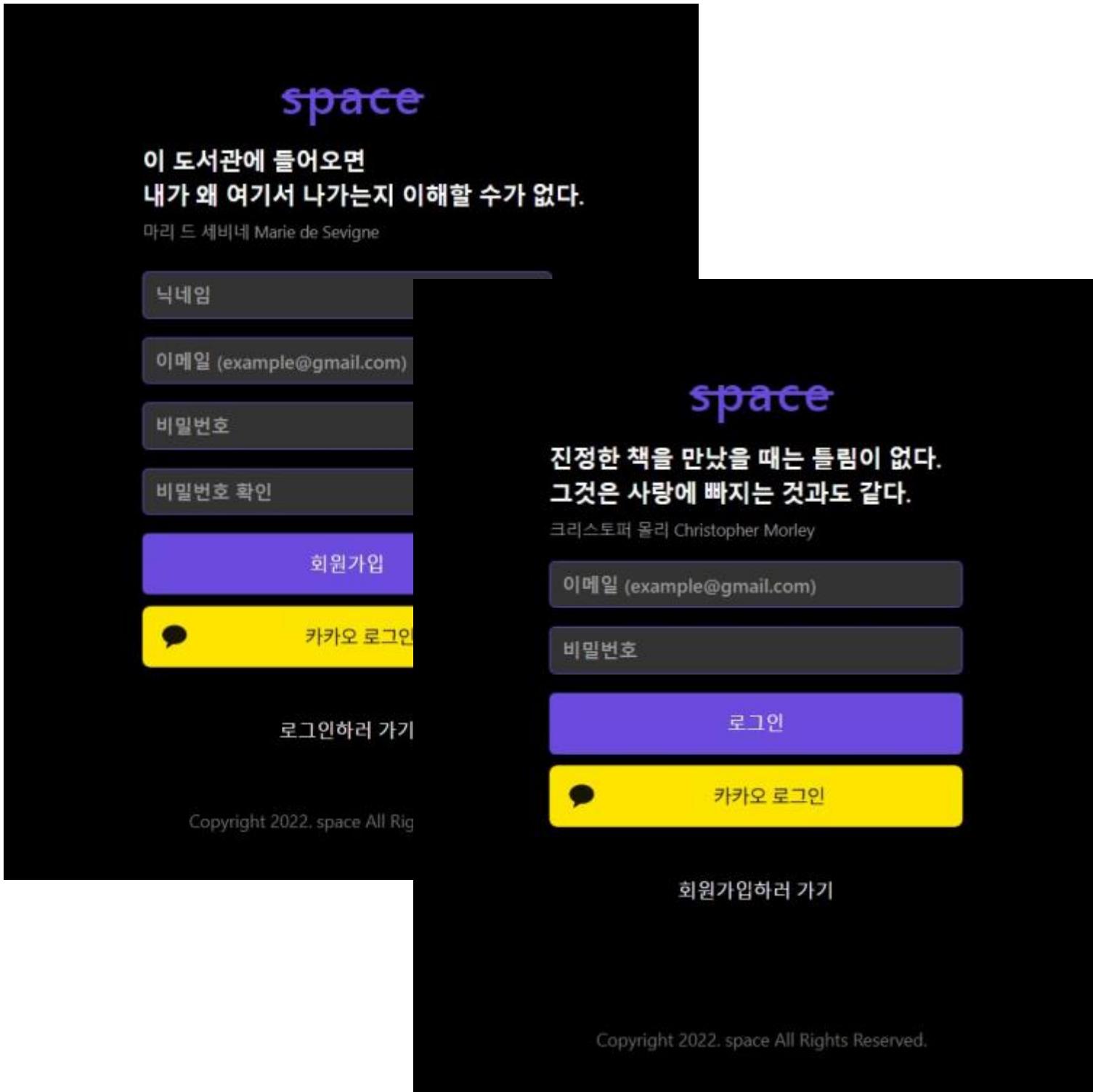
- 사용자가 이메일과 비밀번호를 입력하면 데이터베이스를 조회하여 일치하는 회원 정보가 있는지 확인한 후 있으면 로그인 성공, 없으면 로그인을 막도록 설정했습니다. 로그인에 성공하면 세션에 user 테이블의 id, nickname, profileImage 컬럼을 가져와 저장합니다.

카카오 로그인

- 카카오 로그인 API를 사용하여 따로 회원가입을 하지 않아도 카카오 아이디로 로그인할 수 있도록 구현했습니다.
- 사용자가 카카오 로그인 버튼을 누르면 인가코드를 발급하는 URI로 넘어가고, Service단에서 RestTemplate을 사용하여 발급 받은 인가코드를 카카오에 전달하여 토큰 발급을 요청합니다. 토큰 응답은 json 형태이기 때문에 보다 쉽게 활용하기 위해 ObjectMapper를 활용해 java 형태로 변환해 DTO에 담았습니다. 사용자 정보를 요청하는 메소드에서 발급 받은 토큰을 parameter로 주어 사용자 정보 응답을 받았고, 이 역시 json 형태이기 때문에 마찬가지로 ObjectMapper를 통해 새로운 DTO에 담았습니다. 이를 로그인, 회원가입을 관리하는 user 패키지에서 활용하여 로그인 및 회원가입을 처리했습니다.

다음 페이지: 오류 해결 및 고려사항

기능 소개 회원가입/로그인

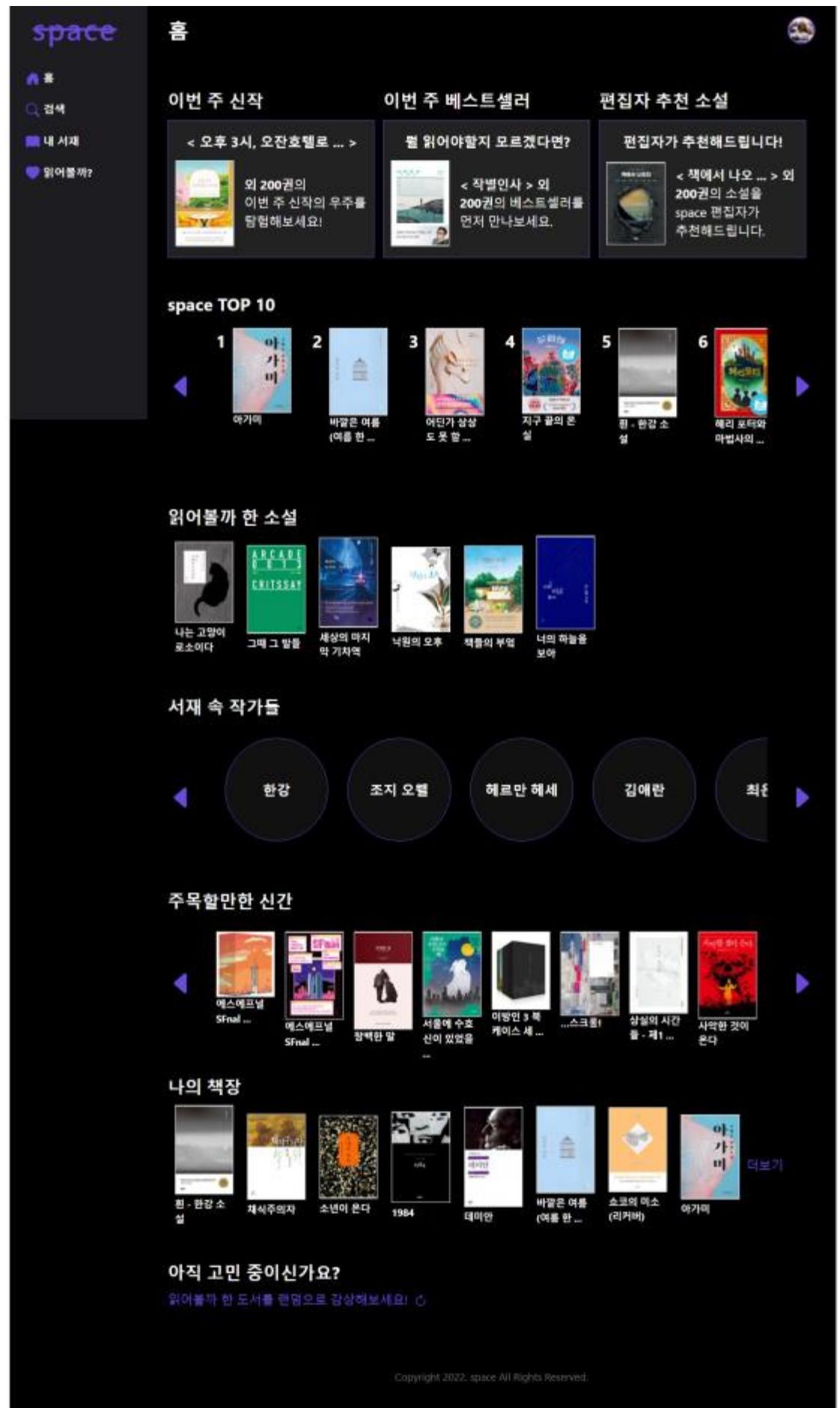


오류 해결

토큰을 이용하여 사용자 정보를 받은 뒤 사용자 정보를 담은 DTO를 UserController에서 활용하려 할 때 해당 DTO가 NULL이라는 NullPointerException 발생했습니다. 처음에는 사용자 정보 자체를 받아오지 못한 것이라 생각하고 오류 원인을 A. 토큰 발급 오류, B. 사용자 정보 요청 URI 오류 두 가지 경우로 나누어 코드를 확인해보았으나 문제가 없었습니다. 그래서 사용자 정보 자체는 받아지는지 확인하기 위해 return 형을 DTO가 아닌 ResponseEntity<String>으로 바꾸어 실험해보았고 그 결과 사용자 정보는 제대로 받아와지는 것을 확인하였습니다. 결국 ObjectMapper를 사용하는 과정에서 문제가 발생하였다는 결론이 나왔고 ObjectMapper의 사용법을 여러가지 찾아보았습니다. boolean형의 경우 DTO에서 generate Getter/Setter를 사용하여 getter를 생성하면 메소드 이름이 get으로 시작하지 않고 is로 시작하게 되는데, 이를 ObjectMapper에서 getter로 인식하지 못하고 값을 mapping할 수 없는 것이 원인이었습니다. 이를 해결하기 위해 boolean형의 변수 위에 모두 @JsonProperty()을 붙여 generate Getter/Setter를 이용하더라도 제대로 값을 mapping할 수 있게끔 수정했습니다.

고려 사항

카카오 로그인이 아닌 일반 회원가입은 단순히 비밀번호만 암호화하여 데이터 베이스에 저장하는 형태로 보안이 취약합니다. 일반 회원가입을 SpringSecurity를 활용하면 보안을 더 강화할 수 있을 것이라 생각됩니다. 또한 카카오 로그인 뿐만 아니라 네이버, 구글 로그인 API도 적용하면 사용자가 더 편리하게 사용할 수 있어 추가하면 좋을 것 같습니다.



기능 소개

홈

이번 주 신작, 베스트셀러, 편집자 추천 소설, 주목할만한 신간

- 알라딘 도서 API에서 제공하는 도서 리스트를 활용하여 구성하였습니다.
- RestTemplate을 사용하여 알라딘에 도서 리스트 API 요청에 필요한 parameter들을 전달하고 응답을 바로 AladinResponse DTO에 저장. 이를 browseBO에서 이번 주 신작, 베스트셀러, 편집자 추천 소설, 주목할만한 신간 별로 나눈 메소드에 활용하여 홈 화면에 가져왔습니다.

space TOP 10

- space의 사용자들이 소설을 내 서재에 담으면 소설 추천하기를 누를 수 있습니다. 이렇게 데이터베이스에 저장된 추천들을 도서 ISBN으로 group by하여 카운트하고, 추천 수 상위 10개의 도서만을 select 해 홈 화면에 보여주었습니다.

서재 속 작가들

- 알라딘 API의 경우 도서 정보 응답값의 작가명이 List 형태가 아닌 ','와 '(작가명)', 외국소설일 경우 '(옮긴이)'로 이어진 String 형태로 주어집니다. (아래 예시)

ex1) 한강 (저자명)

ex2) 베르나르 베르베르 (저자명), 이세욱 (옮긴이)

ex3) 임솔아, 김멜라, 김병운, 김지연, 김혜진, 서수진, 서이제 (저자명)

이를 그대로 사용하면 해당 작가의 다른 저서를 보기 위해서는 검색 화면에 들어가 일일이 작가명을 검색해야 하는 어려움이 있기 때문에 Service단에서 작가명을 받아 DTO에 세팅할 때 String으로 주어진 작가명을 ','를 기준으로 잘라 배열에 담은 후, 반복문을 돌려 하나씩 List에 담는 과정을 수행하였습니다.

- 서재 속 작가들은 사용자가 내 서재에 저장한 소설의 작가명을 위의 과정을 통해 하나씩 잘라내고 그 중 첫 번째 작가명만을 가져와 홈 화면에서 볼 수 있도록 구현하였습니다. 해당 작가명을 클릭하면 그 작가의 저서 목록 화면으로 이동합니다.

기능 소개

페이지

space 이번 주 베스트셀러

50

Copyright 2022. space All Rights Reserved.

The screenshot shows a dark-themed interface for the 'Best Seller' section. It features a sidebar with navigation links: '홈', '검색', '내 서재', and '읽어볼까?'. The main content area displays a book cover for '김중혁 소설집' (Kim Jungsik's Novel Collection) with a brief description: '김중혁 작가의 다섯 번째 소설집. 예외적인 존재들의 삶과 더불어 죽음이라는 본질적 문제를 진중하게 다룬다. 작가 특유의 경쾌하고 유머러스한 문제를 잊지 않으면서도 거부할 수 없는 숙명으로서의 죽음과 거기서 비롯되는 새로운 가능성은 포착한다.' Below this is another book entry for '멋진 신세계' by Aldous Huxley, with a description: '올더스 헉슬리는 명문 집안 출신의 영국 작가로서 광범위한 지식뿐 아니라 예리한 지성과 우아한 문체, 그리고 때로는 낭소적인 유머 감각으로 유명하다. 그가 1932년에 발표한 작품 <멋진 신세계>는 금세기에 미래를 가장 깊이 있고 날카롭게 파헤친 작품 중의 하나로 평가 받는다.' At the bottom of the page are navigation links: ① ② ③ ④.

The screenshot shows a search results page for books. It displays five book entries in a grid format:

- 김동인 작품 모음집 3권 세트 - ... (Author: 김동인, Publisher: 구병모 (추천))
- 젊은 그들 2 (Author: 김동인, Publisher: 구병모 (추천))
- 소설 제주 (Author: 김동인, Publisher: 전석준 김경희 SOOJA ...)
- 오늘의 장르문학 소설 제주 (Author: 김탁환 이영도 이영수(듀니), Publisher: 구병모)
- 위저드 베이커리 (크리스마스 에디 ... (Author: 구병모))

Each book entry includes a small thumbnail image, the title, author, publisher, and a rating section with a blue thumbs-up icon and the number '0'.

기능

알라딘 도서 API에 도서 리스트나 도서 검색 결과를 요청할 때, 어떤 정보를 요청할지와 더불어 응답 받을 최대 도서 수와 결과의 시작 페이지 숫자를 함께 전달합니다. 알라딘 도서 API는 한 페이지에 최대 50개, 총 결과 수는 200개 이상이라도 사용할 수 있는 도서 정보는 200개까지만 받을 수 있기 때문에 space는 모든 화면의 페이지 당 최대 도서 수를 50개로 설정해두었고, 이를 기준으로 페이지 코드를 작성하였습니다.

BookResultInfo DTO를 만들어 페이지 계산을 마치고 나온 페이지 수(EndIndex)를 세팅할 변수를 담았습니다. 페이지 계산은 다음 페이지에서 설명하겠습니다.

기능 소개

페이지 계산

The screenshot shows a dark-themed website interface. On the left, there's a sidebar with icons for 'Home', 'Search', 'My Collection', and 'Read Later'. The main content area has a header '이번 주 베스트셀러' (This Week's Best Seller). It lists two books: '김중혁 소설집' (Kim Jungsuk Novel Collection) and '멋진 신세계' (The Beautiful New World) by Aldous Huxley. Each book entry includes a thumbnail, a brief description, and a 'Read Later?' button. Below the books are page navigation buttons (1, 2, 3, 4) and a copyright notice: 'Copyright 2022. space All Rights Reserved.'

This screenshot shows a grid of five book recommendations. From left to right: '김동인 작품 모음집 3권 세트' (A set of 3 volumes of Kim Dong-in's works), '젊은 그들 2' (Young People 2), '소설 제주' (Novel Jeju), '오늘의 장르문학' (Today's Genre Literature), and '위저드 베이커리 (크리스마스 에디션)' (Wizard Baker (Christmas Edition)). Each item has a thumbnail, a title, a brief description, and a 'Read Later?' button. At the bottom, there are page navigation buttons (1, 2).

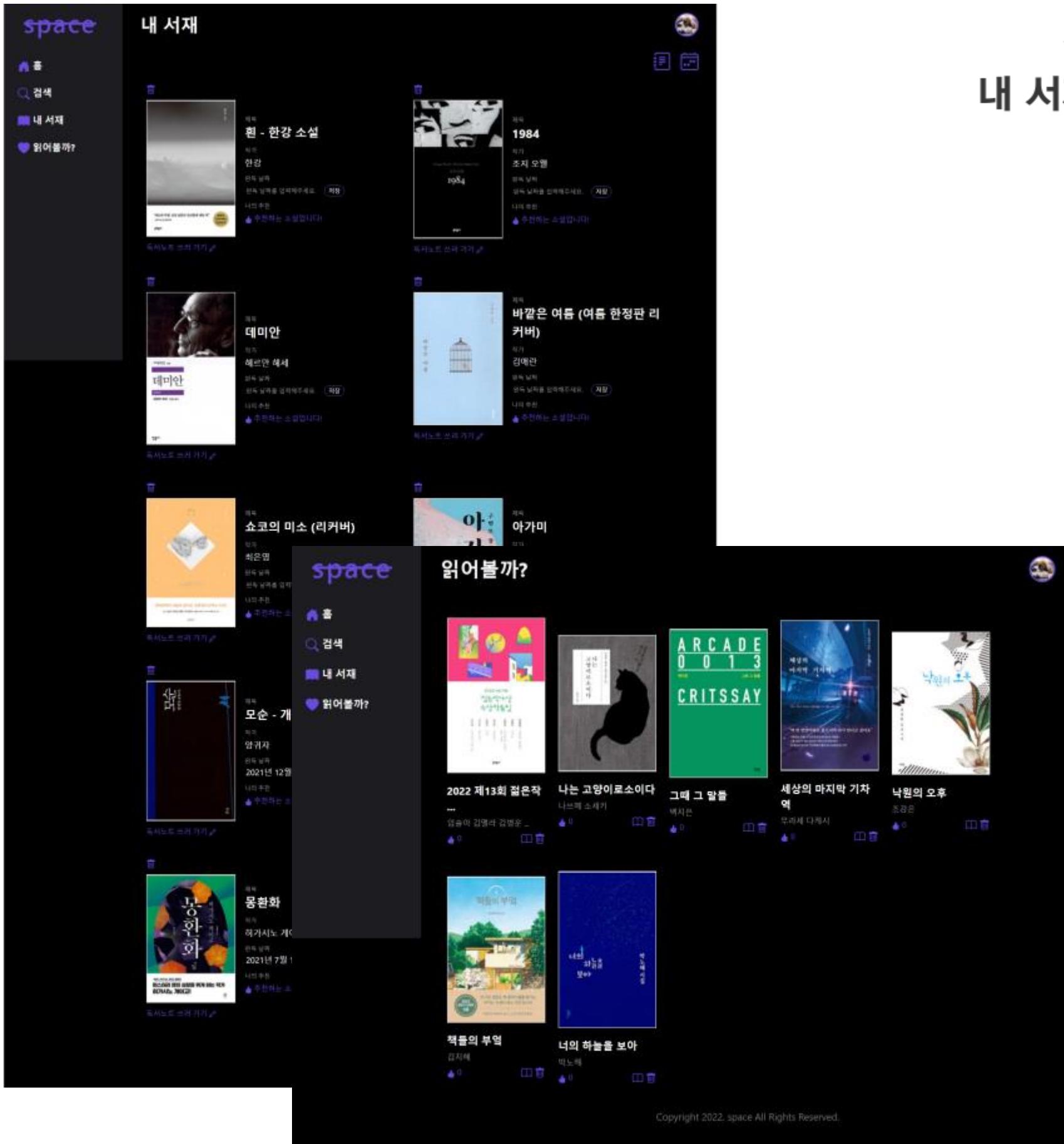
`aladinResponse.getTotalResults` : 알라딘 API에서 받아온 총 결과 수
`bookResultInfo.setEndIndex` : 페이징 계산 후 나온 페이지 수

200 : 알라딘에서 최대로 제공하는 도서 정보의 수

50 : 기본으로 설정해 둔 페이지 당 최대 도서 수

```
if(aladinResponse.getTotalResults() > 200) {  
    bookResultInfo.setEndIndex(4);  
} else if((aladinResponse.getTotalResults() % 50) > 0) && ((aladinResponse.getTotalResults() % 50) < 50)) {  
    bookResultInfo.setEndIndex((aladinResponse.getTotalResults() / 50) + 1);  
} else if((aladinResponse.getTotalResults() % 50) == 0) {  
    bookResultInfo.setEndIndex(aladinResponse.getTotalResults() / 50);  
} else {  
    bookResultInfo.setEndIndex(1);  
}
```

페이징 기능이 들어갈 jsp에서 JSTL c:forEach로 페이지 아이콘을 반복할 때 `begin="1"`, `end="페이지 계산으로 나온 페이지 수"`로 설정하여 페이지 아이콘을 페이지 수만큼 보여지게 하고, 아이콘에 걸어둔 a태그의 href 파라미터에 `page=varStatus.count`를 심어 페이지 아이콘을 누를 때마다 알라딘 API에 요청할 시작페이지 수를 전달하도록 했습니다.



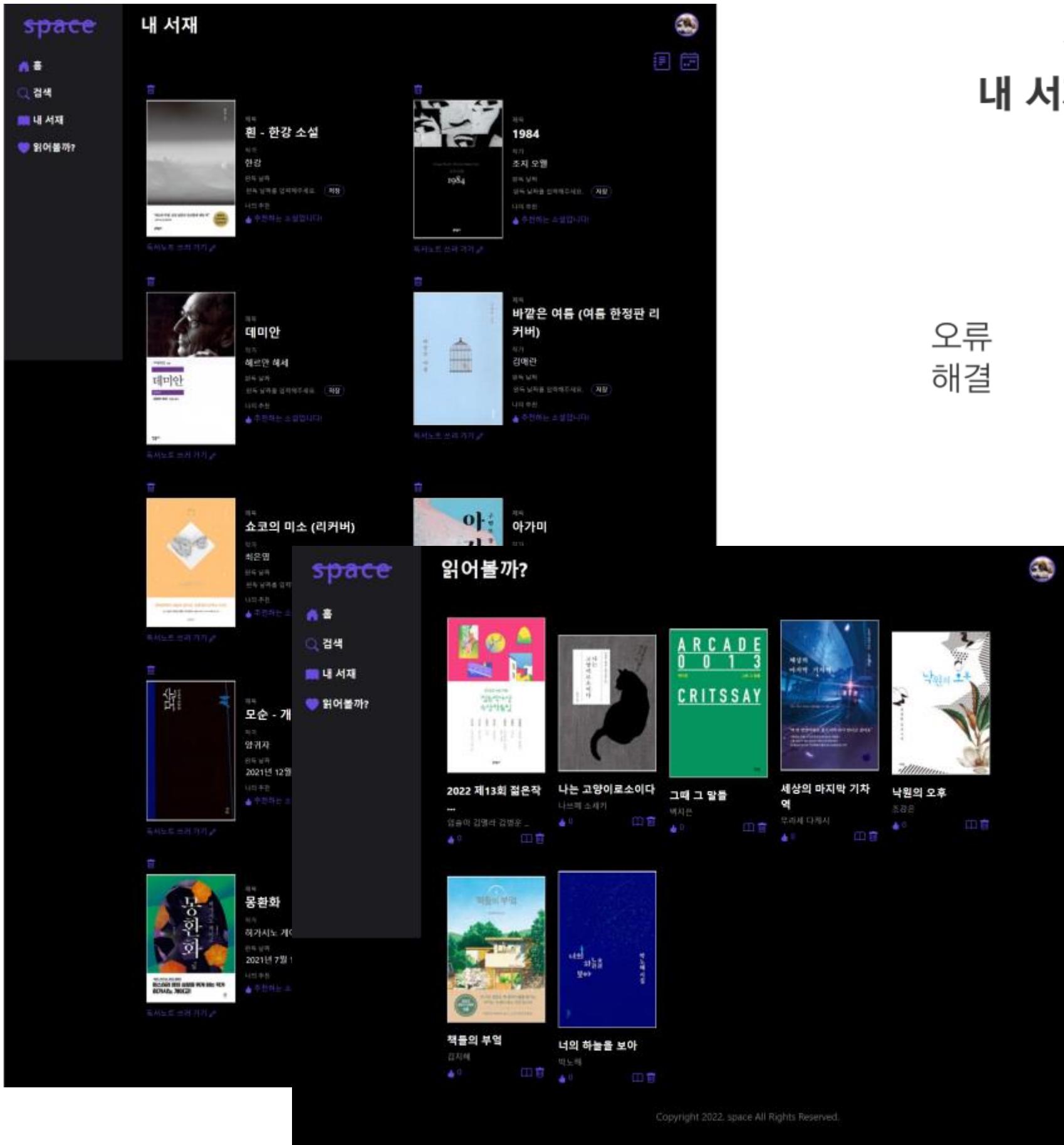
기능 소개

내 서재 / 읽어볼까?

'내 서재'는 사용자가 이미 읽은 소설일 경우 도서 상세 페이지에서 내 서재에 담기를 클릭하면 내 서재에 해당 소설이 담기는 기능이며, 내 서재 화면에서는 도서마다 완독일 입력, 추천하기, 독서노트 쓰러 가기/보러 가기 기능을 사용할 수 있습니다. 상단 우측의 노트와 캘린더 아이콘을 클릭하면 각각 독서노트 모아보기와 독서 캘린더로 이동할 수 있습니다. '읽어볼까'는 위시리스트 기능입니다. 내 서재에 담기 기능과 같은 방식으로 구현했습니다. 읽어볼까 화면은 내 서재보다 단순히 담은 책을 나열하는 방식이며 개별 도서마다 붙어있는 책 아이콘(내 서재로 이동)을 클릭하면 읽어볼까 DB의 해당 도서 데이터가 삭제되고 내 서재 DB에 데이터가 추가됩니다.

기능

다음 페이지: 오류 해결



오류 해결

기능 소개 내 서재 / 읽어볼까?

'읽어볼까' HTTP 500 ERROR

읽어볼까 BO에서는 읽어볼까 테이블에 저장된 사용자의 ISBN을 모두 select하여 DTO에 List 형태로 저장하고 이를 가져옵니다. 그리고 반복문으로 이 리스트에서 ISBN을 하나씩 꺼내 알라딘 API에 도서 정보 요청을 넣고, 받은 응답을 List 형태로 만들어 반복문으로 도서의 제목, 작가, 표지 등 상세 정보를 담는 DTO에 값을 세팅합니다.

여기서 알라딘 API를 통해 받은 응답 리스트가 NULL이므로 리스트의 인덱스를 순회하는 반복문을 수행할 수 없다는 오류가 발생했습니다. 이전까지는 코드가 제대로 작동하다가 어느 날 갑자기 오류가 발생한 것이고, 같은 코드를 사용하는 내 서재 화면은 정상적으로 작동했기 때문에 코드 상의 오류는 아니라고 판단했습니다.

원인을 찾기 위해 오류가 발생한 부분인 알라딘 API를 통해 받은 응답 리스트를 반복문으로 돌려 도서 상세 정보 DTO에 값을 세팅하는 부분에서 디버깅을 진행했고, 그 결과 읽어볼까 테이블에 저장된 id 3의 ISBN과 일치하는 도서 정보 자체에 문제가 생긴 걸 알게 되었습니다. 테이블에는 ISBN만 저장되고 제목 등 다른 도서 정보는 저장되지 않기 때문에 ISBN만으로 그 도서가 어떤 것이었는지는 찾을 수 없었지만 해당 도서 자체가 알라딘 서버에서 사라진 것으로 추정하고 있습니다.

이처럼 도서 정보 자체가 사라질 일은 앞으로도 일어날 가능성이 있기 때문에 ISBN을 통해 하나의 도서 정보를 불러올 때, 받아온 도서 정보가 NULL일 경우 해당 ISBN을 읽어볼까 테이블에서 삭제하고 continue를 사용해 밑의 도서 상세 정보 DTO에 세팅하는 코드를 건너뛰고 다음 반복문을 수행해 다음 ISBN으로 도서 정보를 불러오도록 예외처리를 해주었습니다.

또한, 앞서 언급했듯 읽어볼까와 같은 코드를 사용하는 내 서재에서도 동일한 오류가 발생할 가능성이 높기 때문에 내 서재 BO에서도 같은 예외처리를 적용하였습니다.

기능 소개

독서노트

The screenshot displays two views of the 'space' application's 'Reading Note' feature. The top view, titled '독서노트 기록하기', shows a rich text editor with a toolbar for font style, size, and alignment. The text area contains a quote from Ken Liu: 'Ken Liu <어딘가 상상도 못 할 곳에, 수많은 순록 떼가>'. Below the text is a summary of the book's content. The bottom view, titled '독서노트 보기', shows the same note content in a simplified list format, including the author, creation date (May 11, 2022), and update date (May 13, 2022). Both views include a sidebar with navigation links: 'home', '검색' (Search), '내 서재' (My Library), and '읽어볼까?' (What to Read).

독서노트 기록하기 / 수정하기

- 독서노트 작성 및 수정은 웹에디터 summernote를 사용하여 구현하였습니다. summernote는 부트스트랩과 호환이 잘 되기 때문에 선택하였고, 여기에 가장 기본적인 한글 폰트를 추가하였습니다.

독서노트 보기

- summernote 에디터로 작성한 글은 글에 적용된 html과 css 태그가 함께 DB의 독서노트 테이블에 저장되기 때문에 이를 css에서 white-space: pre-wrap을 적용하여 화면에 사용자가 작성한 그대로 보이도록 했습니다.

- 독서노트 박스의 상단에는 사용자가 독서노트를 작성한 날짜를 알 수 있도록 독서노트 테이블의 createdAt 컬럼을 가져와 보여주었고, 사용자가 독서노트를 수정하게 되면 updatedAt 컬럼을 가져와 수정일도 함께 볼 수 있게끔 구현하였습니다.

기능

기능 소개 독서캘린더

독서 캘린더

2021년 7월

일	월	화	수	목	금	토
27일	28일	29일	30일	1일	2일	3일
4일	5일	6일	7일	8일	9일	10일
11일	12일	13일	14일 <small>동학회</small>	15일	16일	17일
18일	19일	20일	21일	22일	23일	24일

2021년 12월

일	월	화	수	목	금	토
28일 <small>여전히 창작도 못 함</small>	29일	30일	1일	2일	3일	4일
5일	6일	7일	8일	9일	10일	11일
12일	13일	14일	15일	16일	17일	18일
19일	20일	21일	22일	23일	24일	25일
26일 <small>모순 - 개정판</small>	27일	28일	29일	30일	31일	1일
2일	3일	4일	5일	6일	7일	8일

독서캘린더는 사용자가 내 서재에 담은 소설 중, 완독일을 입력한 소설의 경우 캘린더에서 해당 날짜에 완독한 소설의 이름을 보여주는 기능입니다.

캘린더 라이브러리는 FullCalendar를 사용하였으며 캘린더 jsp의 자바스크립트에서 캘린더 이벤트를 작성하는 곳에 ajax를 이용하여 내 서재에 정보를 요청하였습니다.

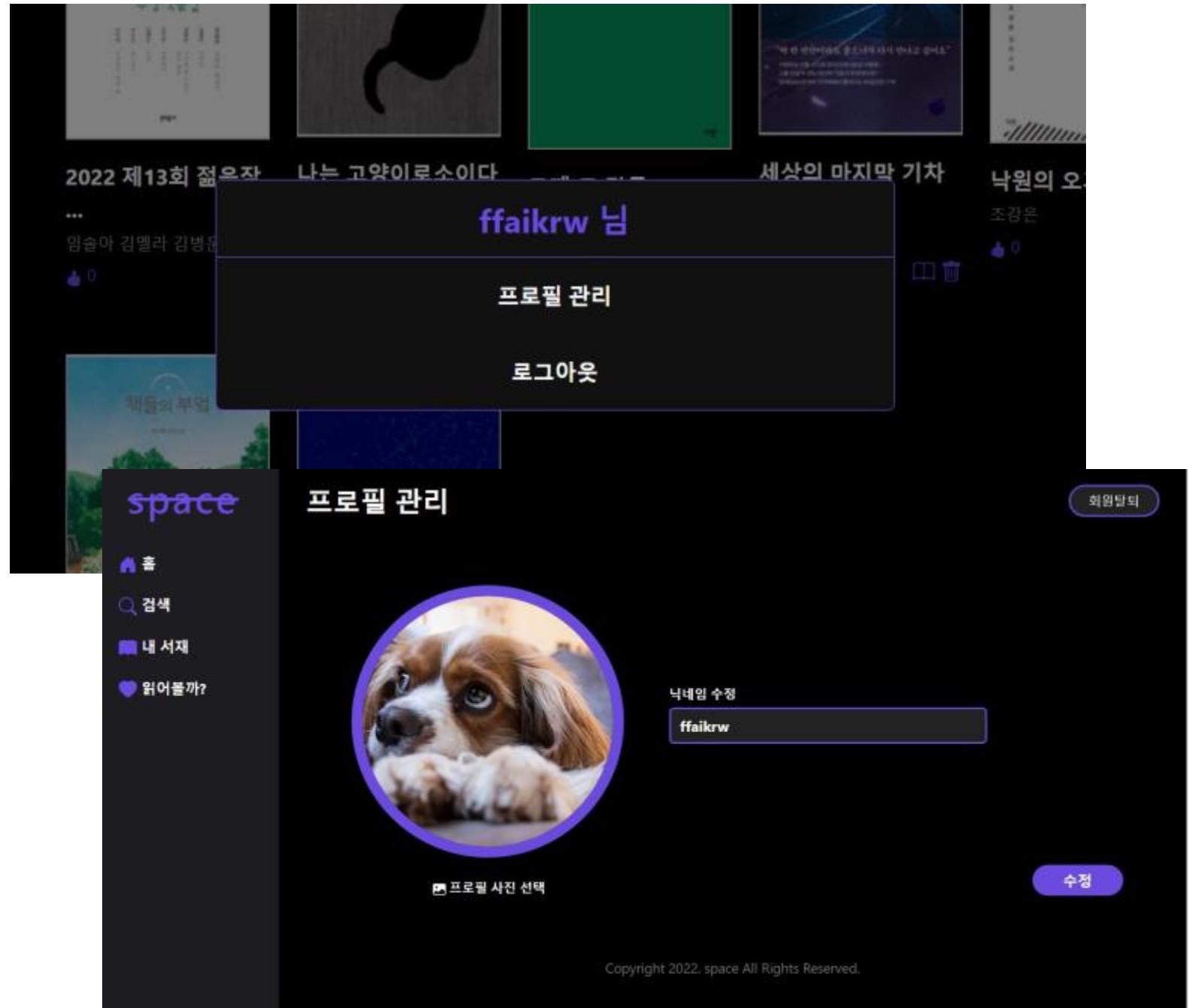
요청한 데이터는 내 서재의 Controller에서 내 서재 BO에 전달해주고, 내 서재 BO에서는 그 요청을 처리하도록 했습니다. 캘린더 이벤트에 넣을 정보를 쉽게 관리하기 위해 캘린더 DTO를 만들어 두고, 내 서재 BO에서 내 서재 테이블에 저장된 ISBN 리스트를 가져와 반복문으로 ISBN을 리스트에서 하나씩 꺼내 알라딘에 개별 도서 정보를 요청하였습니다. 요청한 정보 중에 캘린더에 필요한 도서명만 가져와 캘린더 DTO에 세팅하고, 내 서재 테이블에 저장된 완독일을 가져와 이 역시 캘린더 DTO에 세팅하였습니다. 완독일은 TIMESTAMP 형식으로 저장되어 있기 때문에 SimpleDateFormat을 사용해 완독일의 날짜 형식을 FullCalendar 사용법에 맞는 yyyy-MM-dd 형태로 변경하여 세팅했습니다. 또한 이벤트의 배경색 및 글자색을 바꿔주기 위해 캘린더 DTO에 색도 String으로 세팅해주었습니다.

이렇게 세팅한 DTO를 리스트에 담아 반환해주었고, 내 서재 Controller에서 반환된 리스트를 @ResponseBody 어노테이션을 사용해 json으로 리턴될 수 있도록 했고, 캘린더 jsp의 ajax에서 이 데이터를 받아 캘린더에 보여질 수 있게 구현하였습니다.

기능

기능 소개

프로필 관리 / 회원탈퇴



기능

프로필 관리 모달

- space의 모든 화면 우측 상단에 사용자의 프로필 사진이 보이도록 설정했습니다. 이 사진에 부트스트랩 모달을 적용해 클릭 시 모달창으로 사용자의 닉네임이 보이고 **프로필 관리**, **로그아웃**으로 이동할 수 있도록 했습니다.

프로필 관리 화면

- 모달을 거쳐 프로필 관리 화면으로 이동하면 사용자 닉네임과 프로필 사진을 수정할 수 있습니다. 닉네임은 중복이 불가능하도록 설정했기 때문에 닉네임을 입력했을 때 자바스크립트 ajax를 사용해 DB의 user 테이블을 조회하여 **중복되는 닉네임이 있는지 확인**한 후 닉네임 중복 여부 변수에 중복되지 않으면 false를, 그렇지 않으면 true를 저장하고 닉네임 중복 여부가 true인지 확인하는 if문을 통해 참일 경우 수정이 불가능하도록 return했습니다. 단, **사용자의 원래 닉네임으로 수정했을 경우(수정사항이 없을 경우)** 해당 닉네임이 세션에 저장된 닉네임과 같은지 확인한 후 닉네임 중복 여부 변수에 false를 저장하여 닉네임 중복 여부를 확인하는 if문에서 걸리지 않도록 설정했습니다. **닉네임 입력란이 공백일 경우에도 기존 닉네임으로 저장되도록 했습니다.**

- 프로필 이미지 수정은 자바스크립트를 통해 파일을 선택할 경우 바로 화면에 보이도록 설정했고, 사용자가 업로드한 파일을 저장하고 관리하는 **FileManagerService** 클래스를 이용하여 MultipartFile형으로 전달된 파일을 String 변경해 DB의 user 테이블에 저장했습니다.

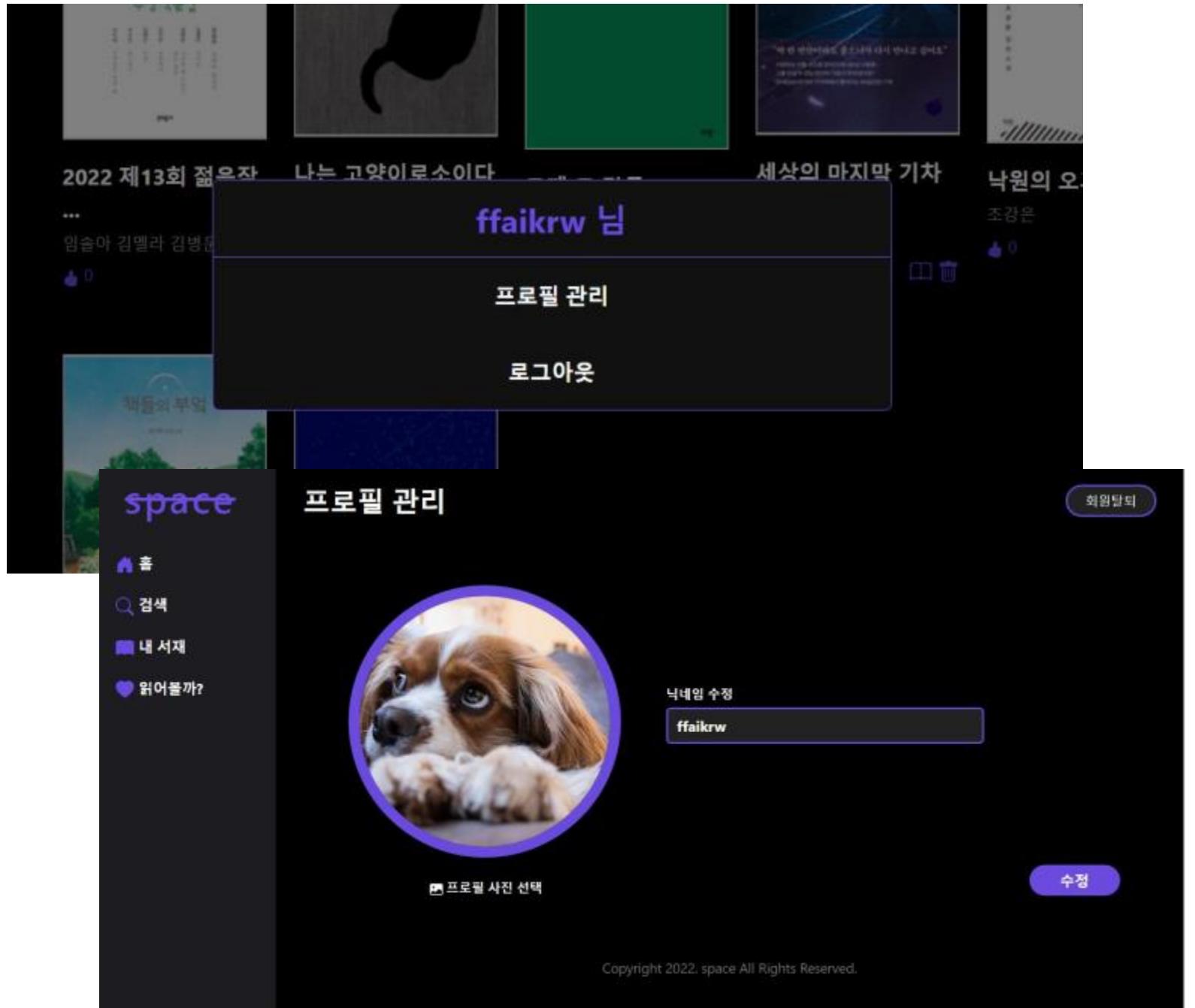
회원탈퇴

- 회원탈퇴 버튼을 누르면 DB의 user, 읽어볼까, 내 서재, 독서노트, 한 줄 평에서 사용자의 데이터가 모두 **삭제되고** 세션에 저장된 사용자 정보 역시 삭제되도록 구현하였습니다.

다음 페이지: 오류 해결

기능 소개

프로필 관리 / 회원탈퇴



오류
해결

이미지 설정

- 처음에는 프로필 관리 화면의 수정 버튼은 프로필 이미지와 닉네임을 한 번에 수정하도록 구현했습니다. 그러나 그렇게 하면 닉네임만 수정했을 경우 파일이 선택되지 않아 DB의 user 테이블에 NULL이 저장되어 기존의 프로필 사진이 보이지 않게 되었습니다. 이를 해결하기 위해 사용자 BO에서 파일이 NULL일 경우와 그렇지 않을 경우로 나누어 NULL이면 닉네임만 수정하는 DAO로 리턴하고, NULL이 아니면 둘 다 수정하는 DAO로 리턴하도록 코드를 수정했습니다.

회원탈퇴

- 회원 탈퇴 기능은 프로필 관리 jsp의 자바스크립트에서 ajax를 통해 회원 정보 삭제를 요청하면, UserBO의 회원탈퇴 메소드에서 user, 내 서재, 읽어볼까, 독서노트, 한 줄 평 테이블에 사용자 정보를 삭제하도록 하고 모두 삭제되면 boolean isAllDeleted 변수에 true를, 하나라도 삭제되지 않으면 false를 저장하여 이를 리턴합니다. UserController에서는 BO의 리턴값이 true면 {"result": "success"}, false면 {"result": "fail"} 형식으로 응답값을 작성하여 ajax에 다시 전달하고, 응답값이 success일 경우 alert창으로 "회원탈퇴가 정상적으로 완료되었습니다.", fail일 경우 "회원탈퇴를 다시 시도해주세요." 가 뜨도록 구현했습니다. 처음 회원탈퇴를 눌렀을 때 "회원탈퇴를 다시 시도해주세요" 가 떠서 다른 아이디로 가입하여 계속 시도해봤더니 결과는 같았습니다. 사용자 BO에서 회원탈퇴 코드를 확인해보니 내 서재, 읽어볼까, 독서노트, 한 줄 평 중에서 하나라도 데이터가 저장되어 있지 않을 경우 delete가 이루어지지 않기 때문에 삭제 실패가 되어 isAllDeleted 변수에 false가 저장되어 반환되었습니다. 오류 해결을 위해 BO에서 각 테이블의 데이터를 삭제하는 코드를 수행하기 전에 그 테이블에 사용자의 데이터가 저장되어 있는지 확인하는 과정을 추가하여 정보가 있으면 삭제를 진행한 후 삭제 성공 시 isAllDeleted에 true를, 실패 시 false를 저장하고, 정보가 없으면 바로 isAllDelete d에 true를 저장하도록 수정했습니다.

감사합니다!
