

게시판 프로젝트

- Git: <https://github.com/hanabe02/board>
- 주요 역할:
 - 프론트/백엔드 클라이언트 구현
 - JWT 기반 로그인 및 사용자 인증 처리
 - Redux + Redux-Saga 상태 및 비동기 흐름 관리
 - 게시판 CRUD 및 페이지네이션 기능 구현
- 개요:
 - Frontend: React, TypeScript, Redux, Redux-Saga, Axios
 - Backend: Spring Boot, Spring Security, JWT, JPA
 - DB: Oracle
 - 기타: Toast 알림, 유효성 검사, 접근 제어 처리

게시글 목록

■	글 번호	제목	작성자	작성일
<input type="checkbox"/>	1	안녕하세요	hanabe01	2025-07-13
<input type="checkbox"/>	2	테스트 게시글 69	writer69	2025-04-15
<input type="checkbox"/>	3	테스트 게시글 68	writer68	2025-04-15
<input type="checkbox"/>	4	테스트 게시글 67	writer67	2025-04-15
<input type="checkbox"/>	5	테스트 게시글 66	writer66	2025-04-15
<input type="checkbox"/>	6	테스트 게시글 65	writer65	2025-04-15
<input type="checkbox"/>	7	테스트 게시글 64	writer64	2025-04-15
<input type="checkbox"/>	8	테스트 게시글 63	writer63	2025-04-15
<input type="checkbox"/>	9	테스트 게시글 62	writer62	2025-04-15
<input type="checkbox"/>	10	테스트 게시글 61	writer61	2025-04-15

글 쓰기

삭제

로그아웃

로그인 회원가입

- 기능 개요:
- JWT 기반 인증 시스템 구현
 - (로그인 시 Access Token 발급, 인증 필요 요청 시 토큰 검증 보안 처리)
- 회원가입 기능 구현
 - (사용자 정보[아이디, 비밀번호 등) 입력받아 서버 등록)
- 비밀번호 암호화 저장
 - (Spring Security의 BCryptPasswordEncoder 를 이용해 안전하게 저장)
- 로그인 상태 관리/비동기 처리
 - (Redux로 로그인 상태를 전역에서 관리, 페이지 간 유지)
 - (로컬 스토리지에 토큰을 저장 자동 로그인 구현, 로그아웃 시 제거)
- 자동 로그인 및 로그아웃 처리
 - (로컬 스토리지에 토큰을 저장해 자동 로그인 구현, 로그아웃 시 제거)

Hanabe

☐ 로그인 상태 유지

로그인

[회원가입](#)[ID 찾기](#)[비밀번호 찾기](#)

Redux 상태관리

- 기능 개요:

- Redux 구조 분리
 - (authActions.js 로그인/로그아웃 인증 관련 액션 정의)
 - (authReducer.js 인증 상태 업데이트 로그인 성공/실패 로그아웃 처리)
 - (authSaga.js 로그인 API 요청 처리)
- 중앙 상태관리 컴포넌트 간 데이터 공유
 - (사용자 정보 전역으로 관리)
- Redux-Saga를 통한 비동기 흐름 처리
 - (액션 발생 - authSaga API 호출 - 성공/실패 처리 - 상태 업데이트)

```
▼ redux
  > auth
  > board
  > profile
  JS rootReducer.js
  JS rootSaga.js
  JS store.js
```

```
▼ redux
  ▼ auth
    JS authActions.js
    JS authReducer.js
    JS authSaga.js
```

```
export const loginSuccess = (user) => ({
  type: LOGIN_SUCCESS,
  payload: user,
});

export const loginFailure = (error) => ({
  type: LOGIN_FAILURE,
  payload: error,
});

export const logout = () => ({
  type: LOGOUT
});
```

CLUD

기능 구현

- 기능 개요:

- 게시글 목록 조회
 - (전체 게시글을 목록으로 조회 가능)
- 개별 게시글 조회
 - (목록에서 게시글 클릭 시, 상세 페이지로 이동)
- 글 작성
 - (로그인된 사용자만 글 작성 가능)
- 글 수정
 - (본인이 작성한 글에만 수정 버튼 노출)
- 글 삭제
 - (다른 사용자의 글에는 수정/삭제 버튼이 표시 X)

글 작성

hanabe01

제목

내용을 입력하세요

입력완료

안녕하세요

작성자: hanabe01

작성일: 2025-07-13

내용: 환영 부탁드립니다.

환영 부탁드립니다.

수정 저장

로그인 회원가입

- 기능 개요:

- JWT 기반 인증 시스템 구현
 - (로그인 시 Access Token 발급, 인증 필요 요청 시 토큰 검증 보안 처리)
- 회원가입 기능 구현
 - (사용자 정보[아이디, 비밀번호 등) 입력받아 서버 등록)
- 비밀번호 암호화 저장
 - (Spring Security의 BCryptPasswordEncoder 를 이용해 안전하게 저장)
- 로그인 상태 관리/비동기 처리
 - (Redux로 로그인 상태를 전역에서 관리, 페이지 간 유지)
 - (로컬 스토리지에 토큰을 저장 자동 로그인 구현, 로그아웃 시 제거)
- 자동 로그인 및 로그아웃 처리
 - (로컬 스토리지에 토큰을 저장해 자동 로그인 구현, 로그아웃 시 제거)

Hanabe

아이디

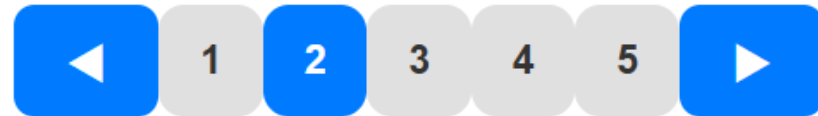
비밀번호

☐ 로그인 상태 유지

로그인

[회원가입](#)[ID 찾기](#)[비밀번호 찾기](#)

페이징 기법



• 기능 개요:

- 대량의 게시글 데이터를 효율적으로 처리하기 위한 UI/UX 기법
- 사용자가 한 번에 많은 데이터를 불러오지 않도록 성능 개선 및 응답 속도 향상
- 현재 페이지, 총 페이지 수 등을 기준으로 클라이언트에서 동적 렌더링

```
@GetMapping("/board")
public List<BoardT0> getBoardInfo(@RequestParam int currentPage,
                                   @RequestParam int size){
    System.out.println("\n" + currentPage);
    System.out.println(size + "\n");
    List<BoardT0> user = boardservice.getBoardInfo(currentPage, size);
    System.out.println(user);
    return user;
}
```

```
@Query(value = "" 1 usage
SELECT * FROM (
    SELECT a.*, ROWNUM rnum FROM (
        SELECT * FROM board ORDER BY BOARD_ID DESC
    ) a WHERE ROWNUM <= :end
) WHERE rnum > :start
"" , nativeQuery = true)
List<BoardT0> findPage(@Param("start") int start, @Param("end") int end);
```

느낀점

- 단순 CRUD 기능에 상태 관리, 인증, 보안, 사용자 경험 등을 고려해보니 좀더 깊은 고민이 필요하다는 필요성을 느꼈습니다.
- 실제로 사용자와 상호작용하는 UI/UX를 구현하고 테스트하면서, 사용자 중심의 개발에 대해 고민할 수 있었습니다.
- 기능 하나하나를 직접 구현하면서, 전체 흐름을 관리하는 경험이 뜻깊었습니다.
- 다양한 기술을 적용해 보면서 좀더 각 장단점을 알아 볼수 있는 경험을 하였습니다.

React + Redux + Redux-Saga 를 활용해 상태 관리와 비동기 흐름 제어의 중요성을 체감하고, 복잡한 로직을 구조화하는 방법을 배웠습니다.
JWT 기반 인증 방식을 직접 구현하여 **보안 개념 및 토큰 처리 흐름**에 대한 감각을 익혔습니다.

백엔드(Spring Boot + JPA)에서 **REST API 설계, DB 연동, 페이징 처리, 시퀀스 설정 등** 실제 서비스에 필요한 핵심 로직 구축하는 경험을 하였습니다.

실제 서비스 게시판 시스템을 구성할 수 있는 실력을 키울 수 있었습니다.

앞으로도 사용자 중심, 유지보수 가능한 코드를 목표로 성장해나가겠습니다.