

*Sprint 6th Part 3 Team 15*



**Sprint** by codeit;

# 주제 선정 배경



# Taskify

✓ 간단한 일정 관리가 가능한 유용한 애플리케이션

✓ Drag And Drop, 복잡한 Modal 등을 구현해 볼 수 있는 도전적인 경험

✓ 구현할 수 있는 추가 기능들의 아이디어

The image displays two screenshots of the Taskify application interface. The top screenshot shows the main dashboard with three columns: 'To Do' (3 items), 'On Progress' (2 items), and 'Done' (2 items). Each item has a preview card with a title, status, and due date. The bottom screenshot shows a detailed view of a task card titled '새로운 일정 관리 Taskify'. This card includes sections for 'To Do', '프로젝트' (Project), '일정' (Schedule), and '비연드' (Outlook). A modal window is open, showing options like '수정하기' (Edit) and '삭제하기' (Delete). The interface is clean with a white background and a sidebar on the left containing various project and document management links.

## 주요 기능

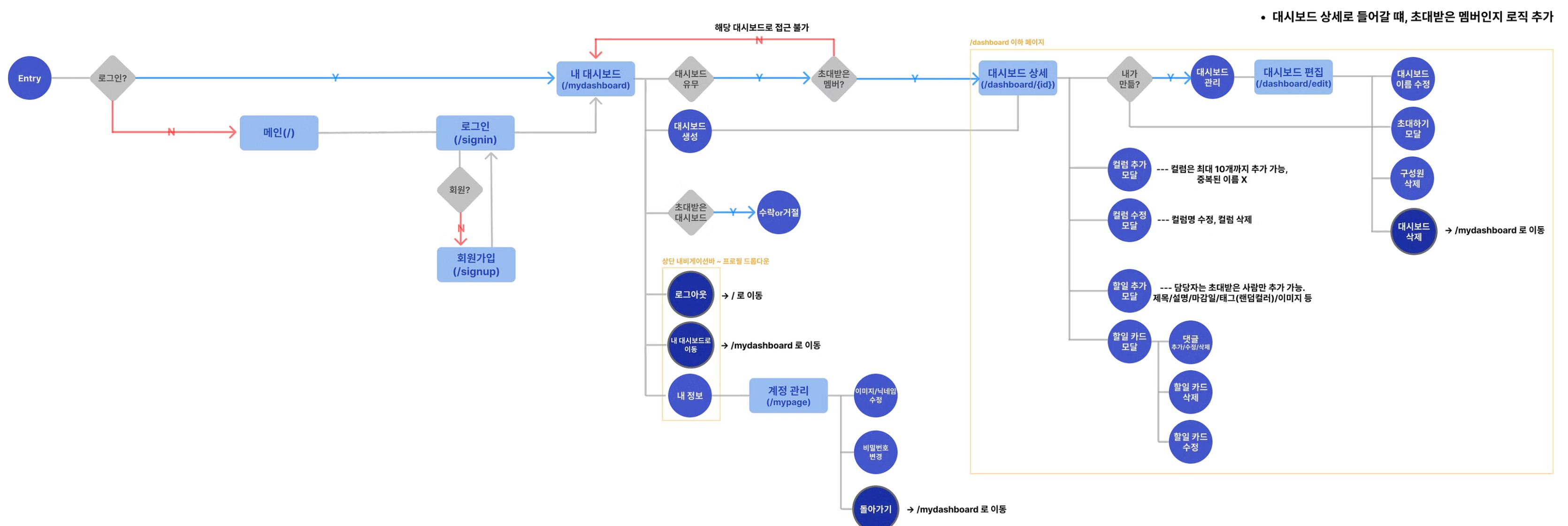
- 대시보드 생성, 수정, 삭제, 구성원 초대 및 관리, 컬럼 관리 및 할 일 카드 관리
- 로그인, 회원가입, 계정 정보 수정, 비밀번호 변경 등의 사용자 인증 및 계정 관리 제공



# 프로젝트 분석

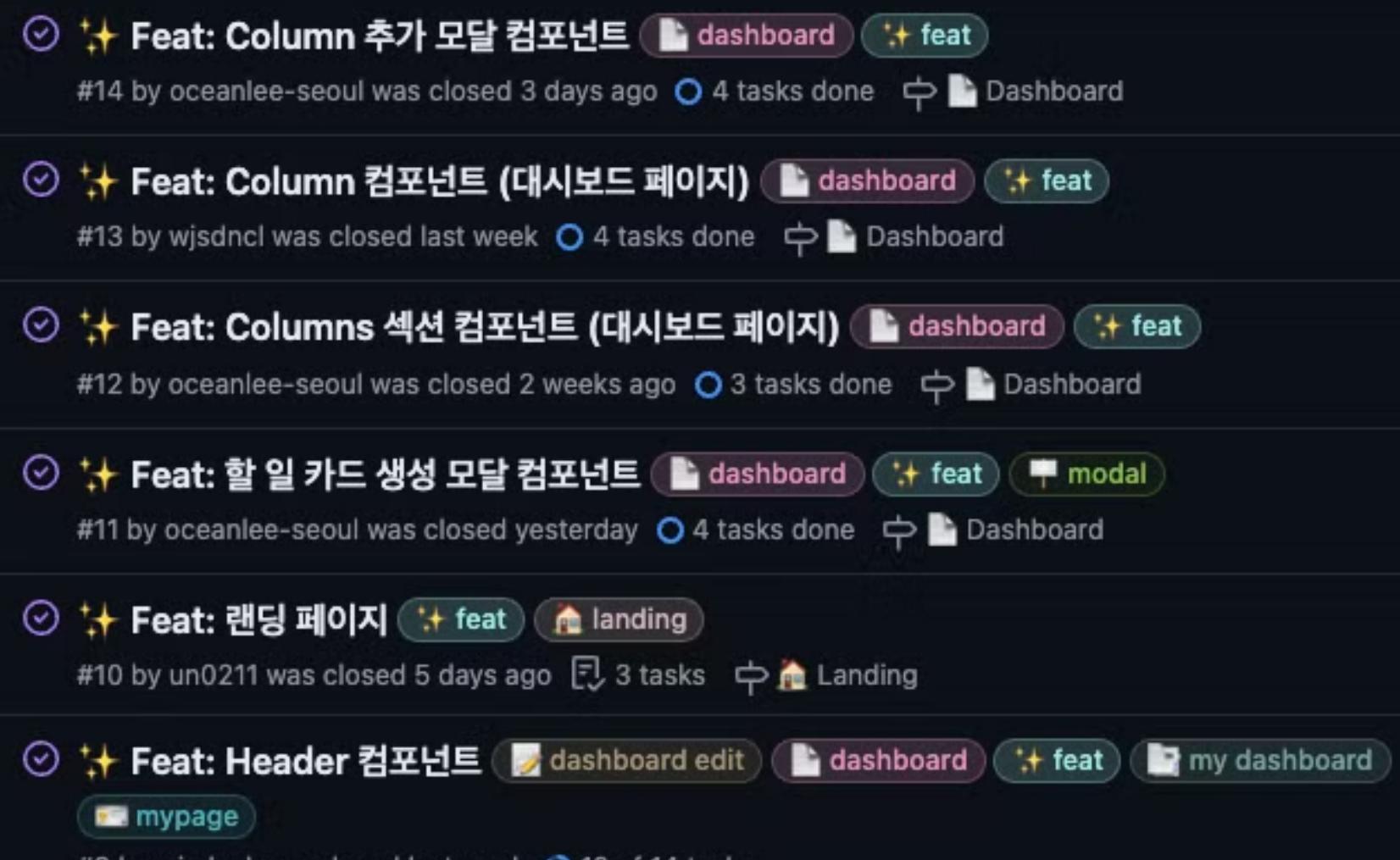
User Flow 작성

## Taskify UserFlow



# 프로젝트 분석

페이지 및 컴포넌트 분석 및 이슈 산정

<b>할 일 생성</b> API : (POST)baseUrl/{teamId}/cards state <ul style="list-style-type: none"><li>담당자 - 해당 대쉬보드의 Member만 가능</li><li>제목, 설명, dueDate, tags, imgFile</li><li>취소, 생성 버튼</li></ul>	<b>대시보드 정보 수정 컴포넌트</b> API : (PUT)baseUrl/{teamId}/dashboards/{dashboardId} state <ul style="list-style-type: none"><li>대시보드 이름 (title)</li><li>컬러</li><li>변경 버튼</li></ul>	 <ul style="list-style-type: none"><li>Feat: Column 추가 모달 컴포넌트</li><li>Feat: Column 컴포넌트 (대시보드 페이지)</li><li>Feat: Columns 섹션 컴포넌트 (대시보드 페이지)</li><li>Feat: 할 일 카드 생성 모달 컴포넌트</li><li>Feat: 랜딩 페이지</li><li>Feat: Header 컴포넌트</li></ul>
<b>할 일 수정</b> API : (PUT)baseUrl/{teamId}/cards state <ul style="list-style-type: none"><li>담당자 - 해당 대쉬보드의 Member만 가능</li><li>제목, 설명, dueDate, tags, imgFile</li><li>취소, 수정 버튼</li></ul>	<b>멤버 컴포넌트</b> API : (GET)baseUrl/{teamId}/members state <ul style="list-style-type: none"><li>멤버 목록 배열</li><li>페이지네이션</li><li>삭제 버튼</li></ul>	
<b>칼럼 추가</b> API : (POST)baseUrl/{teamId}/columns state <ul style="list-style-type: none"><li>title</li><li>취소, 생성 버튼</li></ul>	<b>초대 내역 컴포넌트</b> API : (GET)baseUrl/{teamId}/dashboards/{dashboardId}/invitations state <ul style="list-style-type: none"><li>초대 목록 배열</li><li>페이지네이션</li><li>초대하기 버튼 (초대하기 모달 오픈)</li></ul>	

요구사항과 피그마를 바탕으로 컴포넌트를 분석 (State, API 정리)  
분석된 컴포넌트들을 Issue 등록

# 프로젝트 분석

이슈 산정

1	Rule: 프로젝트 규칙 논의 #32	Rule: 프로젝트 규칙 논의 #32
2	Chore: 프로젝트 초기 설정 #1	Chore: 프로젝트 초기 설정 #1 🎨
3	Analysis: 요구사항 분석 #35	Analysis: 요구사항 분석 #35
4	Feat: 레이아웃 컴포넌트 #31	Feat: 레이아웃 컴포넌트 #31 🎨
5	Feat: 유저 기능 #28	Feat: 유저 기능 #28 🦸
6	Feat: 페이지 라우팅 설정 #30	Feat: 페이지 라우팅 설정 #30 🎨
7	Chore: tailwind 컬러 팔레트 및 반응형 브... #33	Chore: tailwind 컬러 팔레트 및 반응형 브레이크 포인트 세팅 #33 🎨
8	Chore: eslintrc 변경 #37	Chore: eslintrc 변경 #37 🎨
9	Chore: 배포하기 #36	Chore: 배포하기 #36 🎨
10	Feat: Columns 섹션 컴포넌트 (대시보드 ...) #12	Feat: Columns 섹션 컴포넌트 (대시보드 페이지) #12 🦸
11	Feat: SideBar 컴포넌트 #2	Feat: SideBar 컴포넌트 #2 🎨
12	Bug: eslintrc.json 변경에 맞게 패키지 업... #45	Bug: eslintrc.json 변경에 맞게 패키지 업데이트 #45 🎨
13	Feat: 태그 생성시 컬러 부여 기능 util #26	Feat: 태그 생성시 컬러 부여 기능 util #26 🎨
14	Feat: 유저 인증이 필요한 데이터 패치 #54	Feat: 유저 인증이 필요한 데이터 패치 #54 🦸
15	Feat: Column 컴포넌트 (대시보드 페이지) #13	Feat: Column 컴포넌트 (대시보드 페이지) #13 🦸
16	Feat: 대시보드 전역 상태 관리 #52	Feat: 대시보드 전역 상태 관리 #52 🦸
17	Refactor: 서비스 로직 분리 #58	Refactor: 서비스 로직 분리 #58 🦸
18	Feat: Header 컴포넌트 #9	Feat: Header 컴포넌트 #9 🎨
19	Bug: commitlint 규칙때문에 이슈번호 커... #51	Bug: commitlint 규칙때문에 이슈번호 커밋메세지에 추가 거절 #51 🎨

# Tech Stack

Framework & Styling



Next.js



TypeScript



tailwindcss

# Tech Stack

Development Tools and Deployment



# Tech Stack

## Data Management



Redux Toolkit

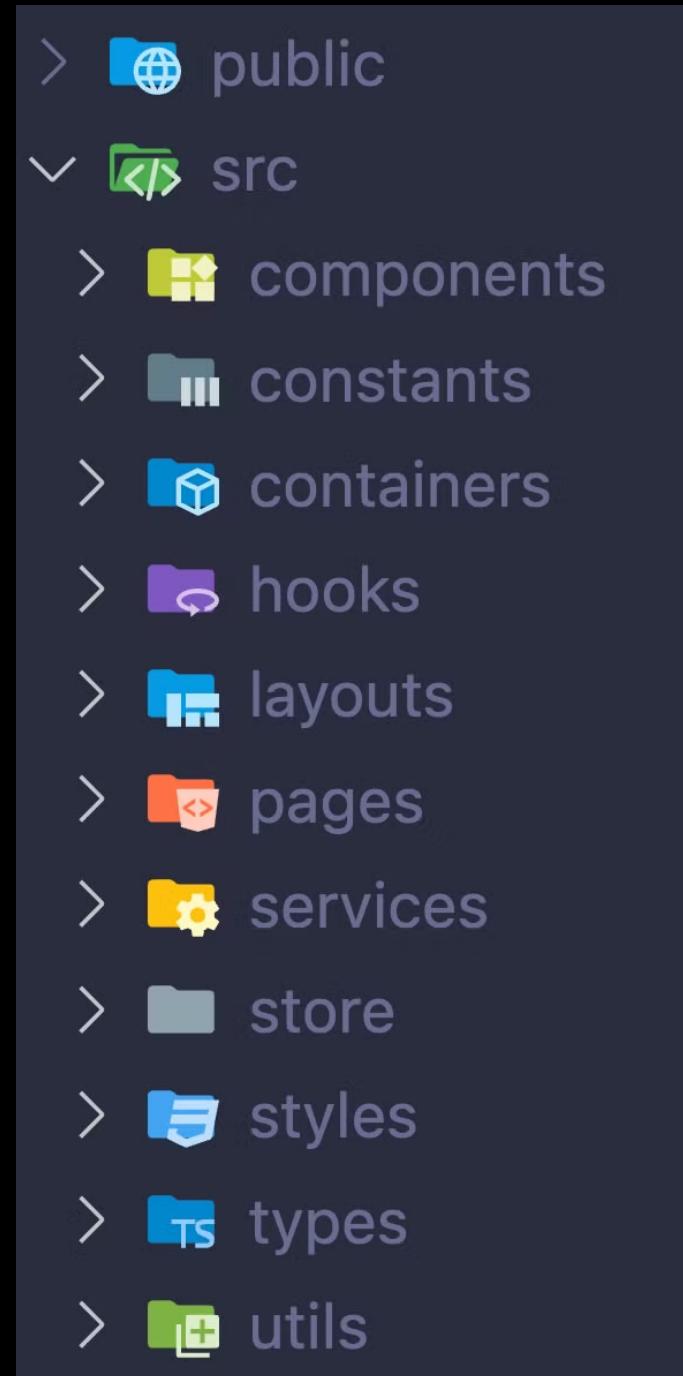


Tanstack Query



axios

# Directory Structure



01

components

프로젝트에서 사용되는 공통 컴포넌트 정의

02

constants

프로젝트에 필요한 상수 정의

03

containers

특정 페이지에 필요한 컴포넌트 정의

04

hooks

커스텀 Hooks 정의

05

layouts

SideBar, Header와 같은 Layout 컴포넌트가 정의

06

pages

Next.js 페이지 라우팅에 필요한 page가 정의

07

services

프로젝트 내에서 사용되는 API들이 정의

08

store

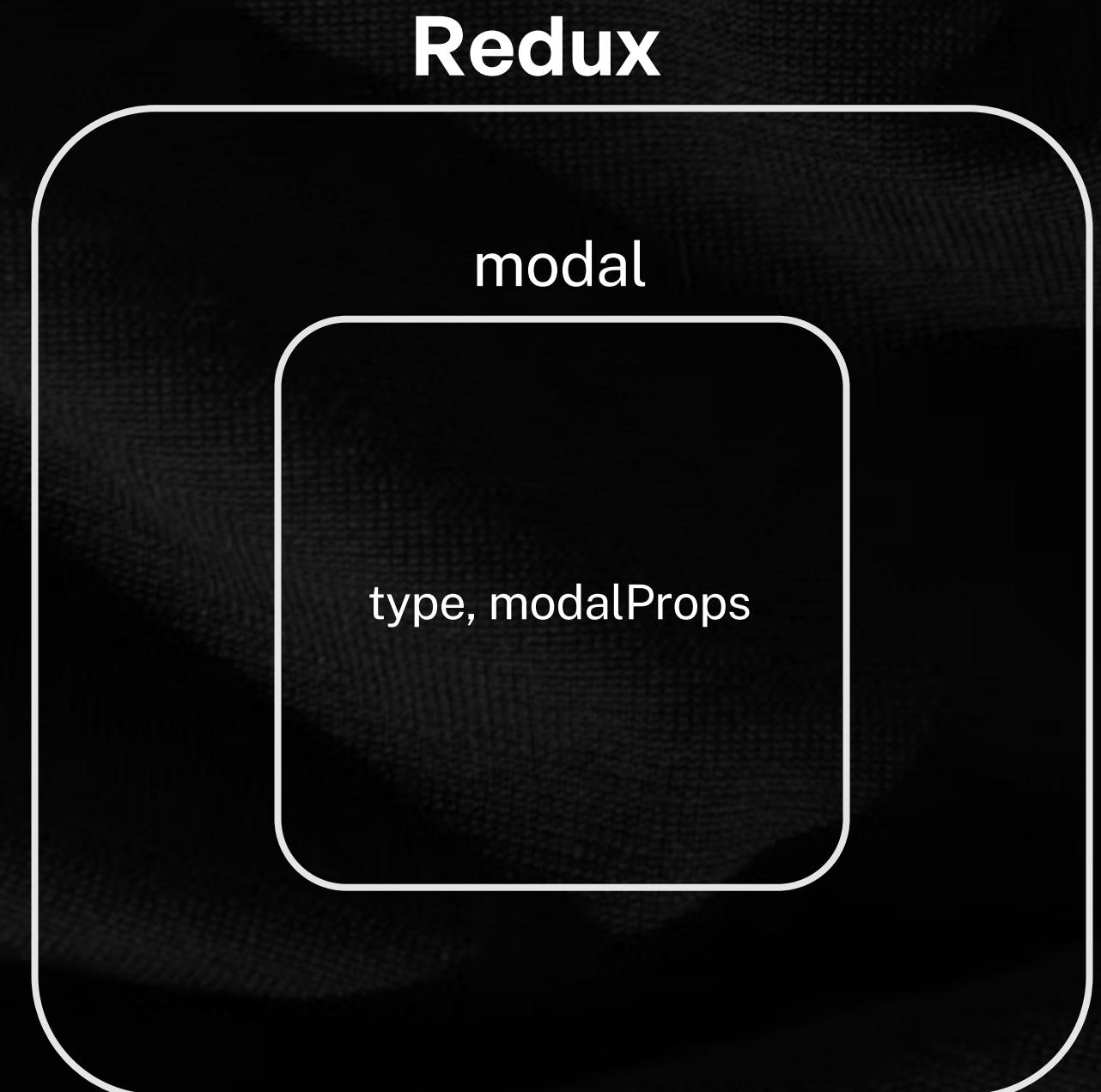
Redux toolkit 사용을 위한 store 정의

# Redux Toolkit의 활용

The image displays several modal dialog boxes from a Taskify application, illustrating the implementation of Redux Toolkit for state management:

- Top Left Modal:** A "새 컬럼 생성" (Create Column) dialog. It contains a title, a text input field with the placeholder "To Do", and a note below stating "중복된 컬럼 이름입니다." (Duplicate column name).
- Top Center Modal:** A "할 일 설정" (Task Configuration) dialog. It includes dropdowns for "상태" (Status) set to "To Do" and "담당자" (Assignee) set to "배우철" (Baeউচেল). Fields for "제목" (Title) containing "스프린트 12" and "설명" (Description) containing "스프린트 12의 아젠다 논의" are also shown.
- Top Right Modal:** A "대시보드 이름" (Dashboard Name) dialog. It shows a text input field with "뉴프로젝트" (New Project), a row of five colored circular icons (green, purple, orange, blue, pink), and two buttons: "취소" (Cancel) and "생성" (Create).
- Middle Left Modal:** A "새로운 일정 관리 Taskify" (New Calendar Management Taskify) dialog. It features a header with tabs like "To Do", "프로젝트", "일반", "백엔드", and "상". Below the tabs is a text area with placeholder text: "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum finibus nibh arcu, quis consequat ante cursus eget. Cras mattis, nulla non laoreet porttitor, diam justo laoreet eros, vel aliquet diam elit at leo." An illustration of two people working at a desk is shown. A message box at the bottom left says "댓글" (Comment) and "댓글 작성하기" (Write comment). A timestamp "정만철 2022.12.27 14:00" and a note "오늘안에 CCC 까지 만들 수 있을까요?" are also present.
- Middle Right Modal:** A confirmation dialog with the text "랜딩 페이지를 제외한 모든 페이지에서 여러가지 모달이 펼쳐집니다!" (Modals will open on all pages except the landing page!).
- Bottom Left Modal:** A confirmation dialog with the text "가입이 완료되었습니다!" (Registration completed!). A "확인" (Confirm) button is at the bottom.

# Redux Toolkit의 활용



# Redux Toolkit의 활용

```
○ ○ ○  
const Modal = () => {  
  const { type, modalProps } = useSelector(modalSelector);  
  const { closeModal } = useModal();  
  
  if (!type) return null;  
  
  return (  
    <div onClick={() => closeModal}>  
      /* 각 모달 타입에 맞는 컴포넌트 */  
      {type === 'NOTIFICATION' && <NotificationModal {...modalProps} />}  
      {type === 'CONFIRM' && <ConfirmModal {...modalProps} />}  
      {/* ...기타 모달 타입들 */}  
    </div>  
  );  
};
```

모달이 필요한 곳에서 useModal을 사용

\_app.tsx

```
○ ○ ○  
= {store}>  
{ ...pageProps } />  
</MainLayout>  
</Provider>;
```

# TanStack Query v5

○ ○ ○

```
const {  
  data: dashboard,  
  isLoading,  
  error,  
} = useFetchData<Dashboard>(['dashboard', id], () => getDashboard(id));
```

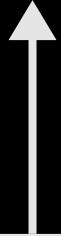


useEffect, async - await 패턴 없이 간단하게 데이터 패칭

# TanStack Query v5

○ ○ ○

```
const {  
  data: dashboard,  
  isLoading,  
  error,  
} = useFetchData<Dashboard>(['dashboard', id], () => getDashboard(id));
```



Data, isLoading, Error와 같은 예외 처리를 간편하게 사용

# TanStack Query v5

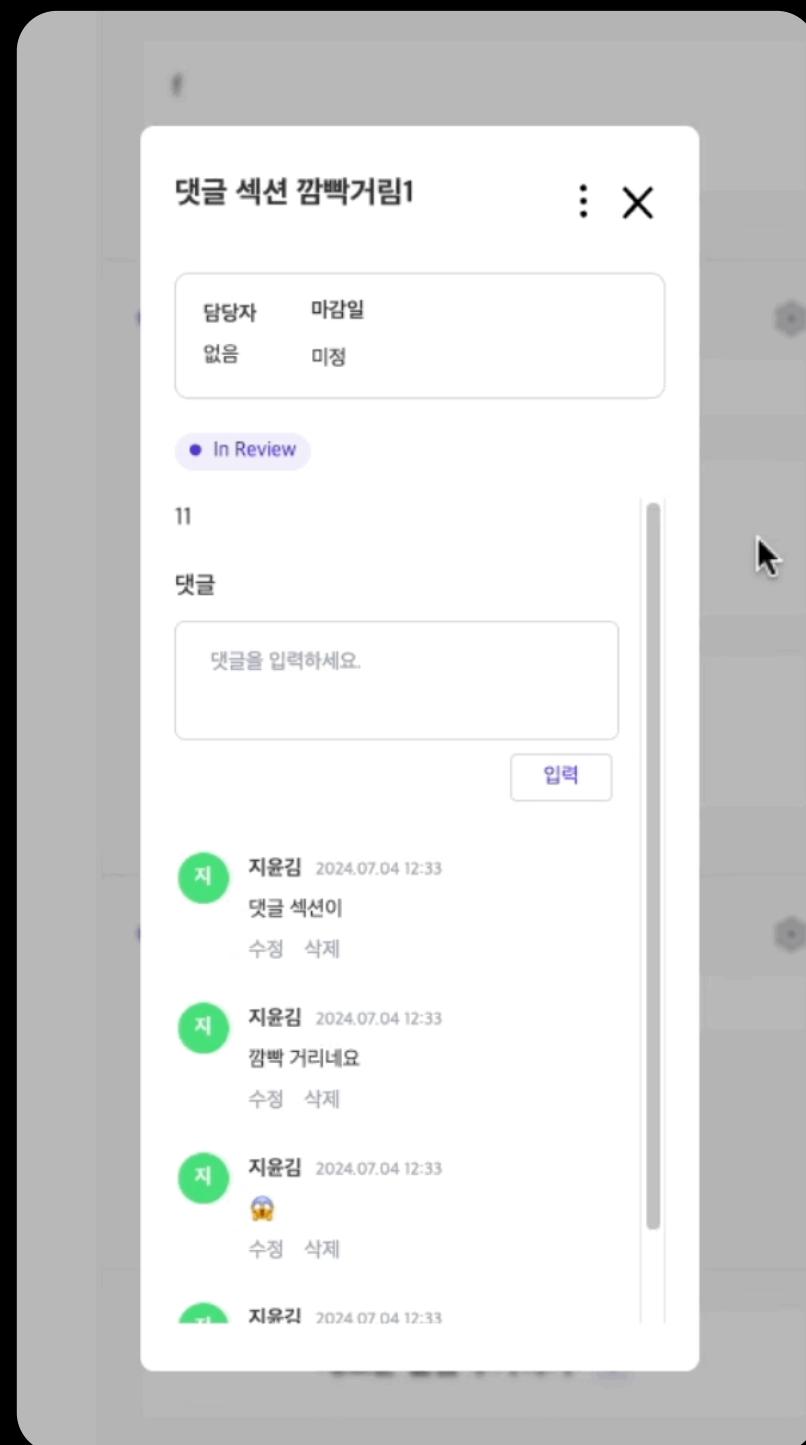
○ ○ ○

```
const {  
  data: dashboard,  
  isLoading,  
  error,  
} = useFetchData<Dashboard>(['dashboard', id], () => getDashboard(id));
```

같은 키를 사용하는 요청은 값을 캐싱하여 빠른  
응답

# TanStack Query v5

# 트러블 슈팅



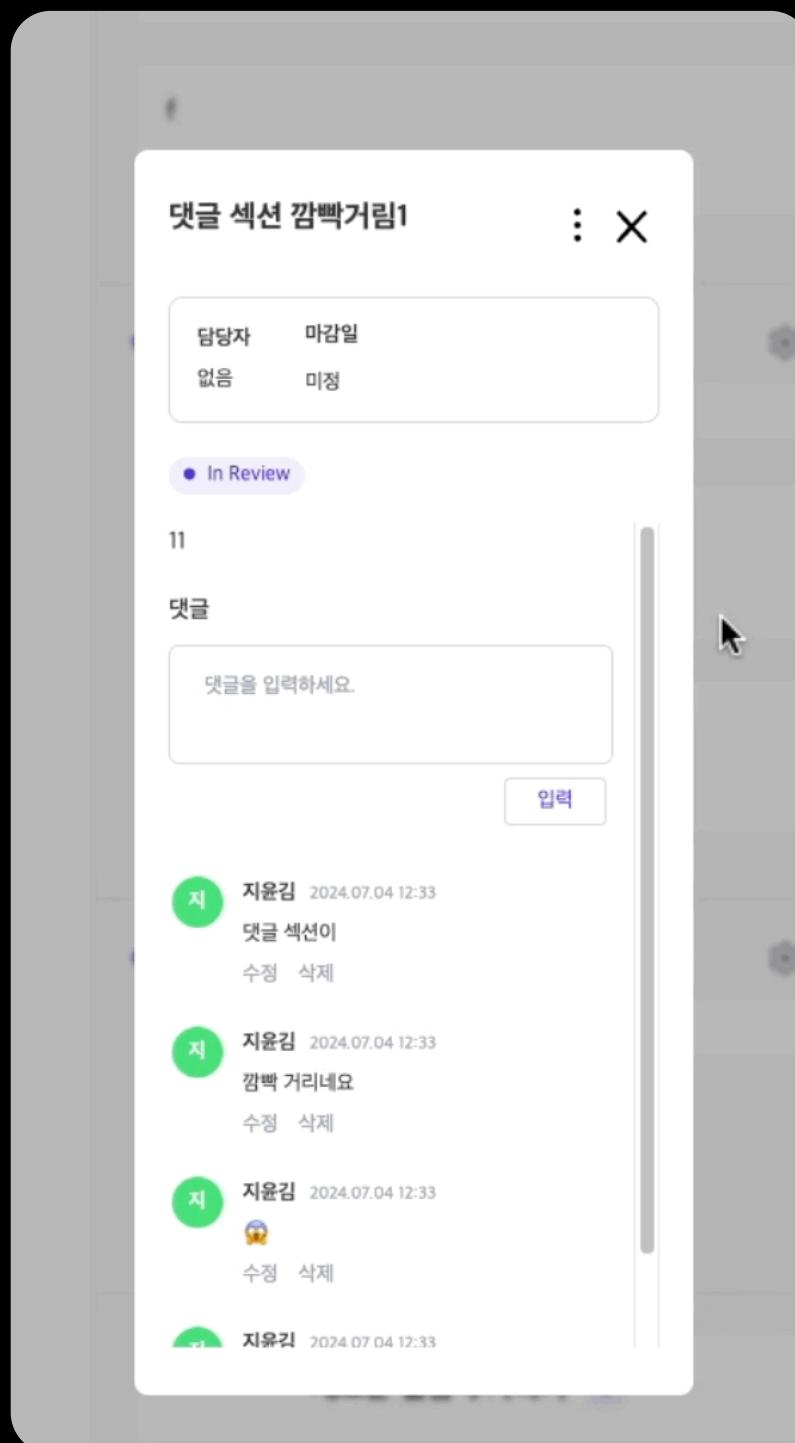
카드 모달의 댓글 섹션이 깜빡 거리는 문제 발생

새로운 데이터를 불러오기 전, 이전에 캐싱되어 있던 데이터가 출력

모달이 그려지기 전에 쿼리가 refetch되도록 순서 보장이 필요

# TanStack Query v5

## 트러블 슈팅



○ ○ ○

// `useFetchData` 커스텀 훅(`useQuery` 반환)에서 `refetch`함수를 받는다.

// 여기서 각 댓글들은 각 카드의 고유한 `id`값을 쿼리키로 등록해 패치한다.

```
const { data, refetch } = useFetchData<CommentsResponse>(['comments', card.id], () => getComments(card.id));
```

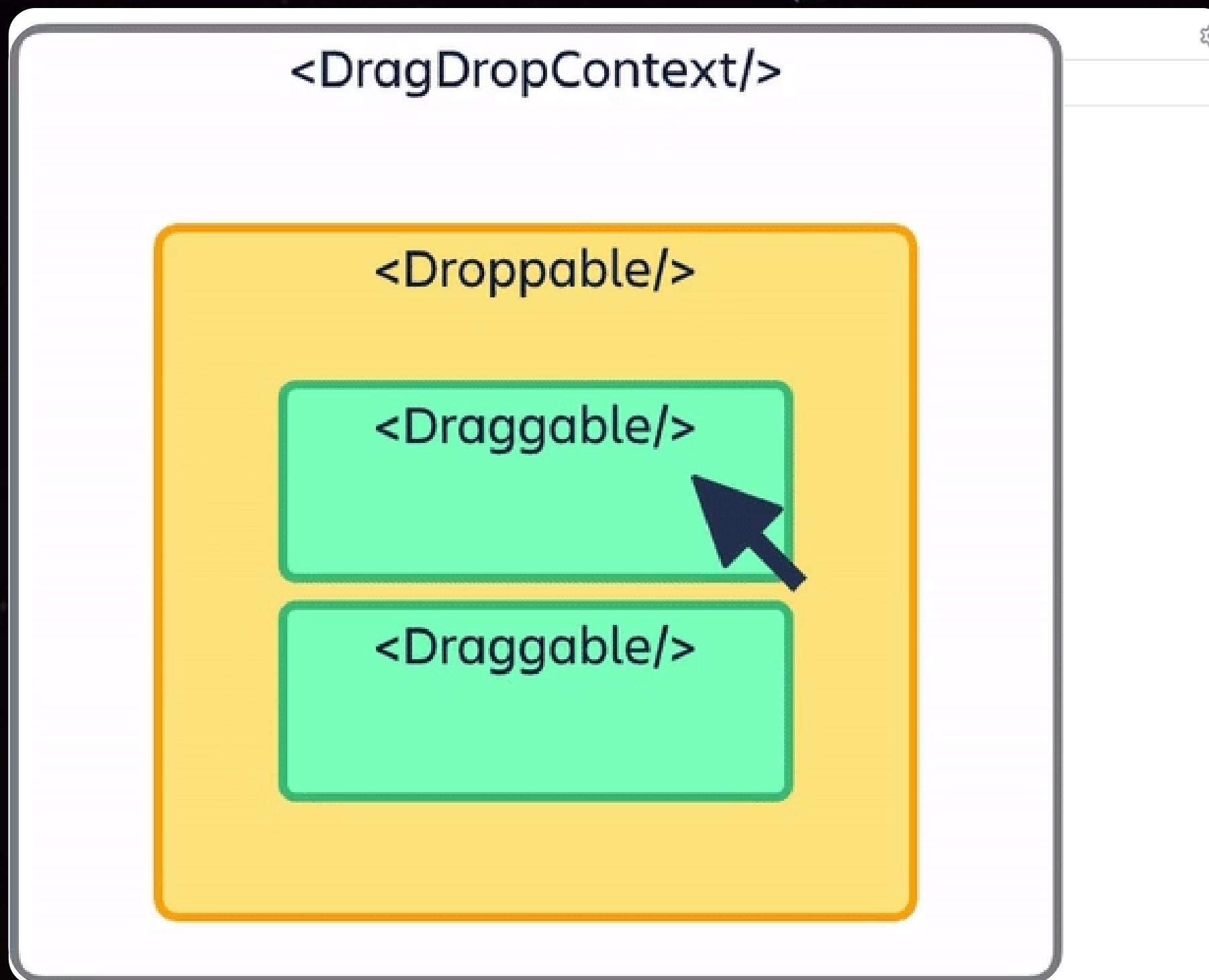
// `useEffect()` 훅으로 모달이 열릴 때마다 `refetch`를 호출해 댓글 데이터를 최신상태로 유지한다.

...

```
useEffect(() => {
  refetch();
}, [refetch]);
```

# Drag & Drop, react-beautiful-dnd

추가 기능 및 개선 사항



<DragDropContext />  
<Droppable />  
<Draggable />  
및  
onDragEnd,  
onDragStart,  
onDragUpdate

# 다크 모드

Taskify

Dashboards +

My Dashboard

My Page

• 습관 만들기 🏆

Theme

습관 만들기 🏆

To do 1 관리 초대하기 imAdmin

On progress 2 관리 초대하기

Done 1 관리 초대하기

운동하기 운동 습관 미정

책 읽기 독서 습관 미정

끝내주게 밥 먹기 밥 습관 미정

리액트 공부하기 습관 공부 미정



Try Pitch

# 다크 모드

○ ○ ○

```
<ThemeProvider attribute='class' defaultTheme='system' enableSystem={true} />
```

# 알림 기능

○ ○ ○

```
import { useQuery, QueryKey, UseQueryResult } from '@tanstack/react-query';

const useFetchData = <T>(
  queryKey: QueryKey,
  getService: () => Promise<{ data: T }>,
  refetchInterval: false | number = false,
  enabled: boolean = true,
  handleSuccess?: () => void,
): UseQueryResult<T, Error> => {
  return useQuery<T, Error>({
    queryKey: queryKey,
    queryFn: async () => {
      try {
        const response = await getService();
        handleSuccess && handleSuccess();
        return response.data;
      } catch (error) {
        throw new Error('데이터를 불러오는 중 에러 발생: ' + error);
      }
    },
    retry: 1,
    refetchInterval: refetchInterval,
    enabled: enabled,
  });
};

export default useFetchData;
```

# 알림 기능

○ ○ ○

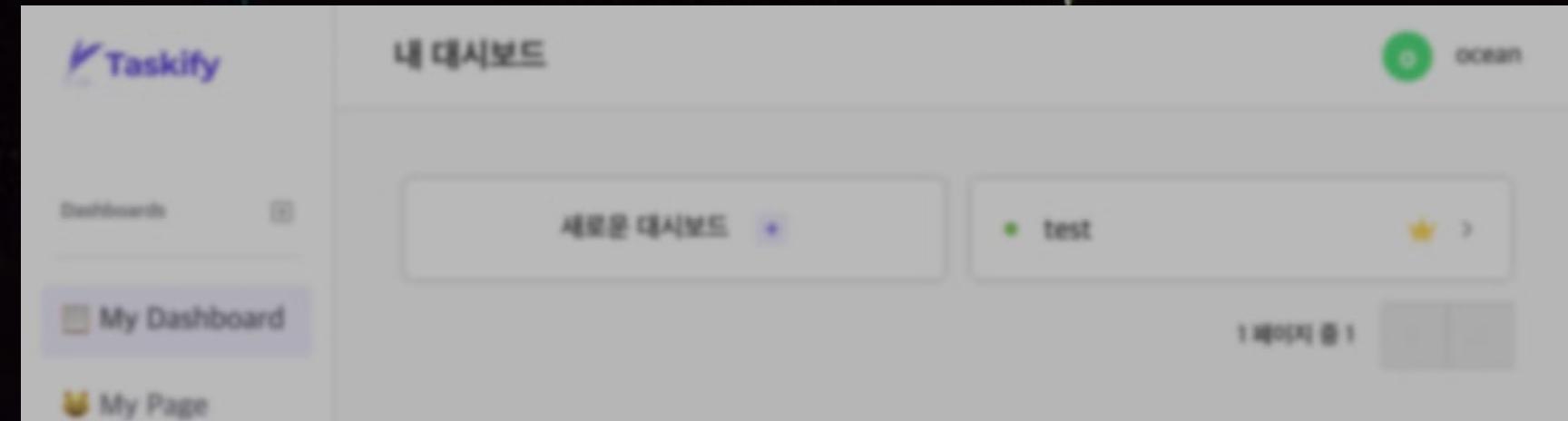
```
const { data } = useFetchData<InvitationsResponse>(['invitations', 'notice'], () => getInvitationsList(), 5000);
```

# 알림 기능

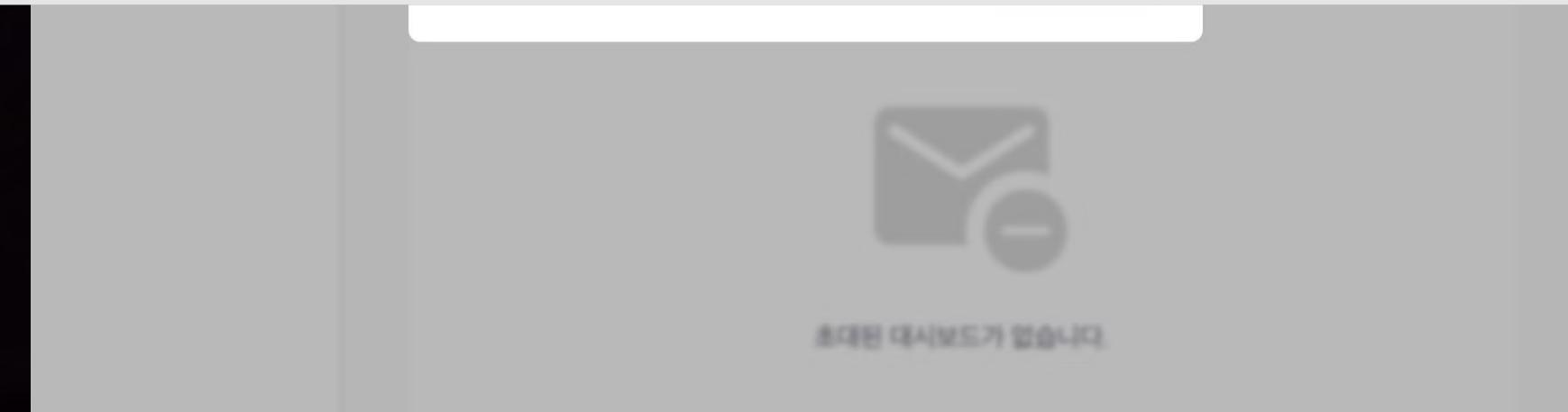
○ ○ ○

```
useEffect(() => {
  if (data && data.invitations[0] && savedInvitations) {
    if (savedInvitations.length === 0) {
      •// NOTE: 기존 초대 없음 -> 모든 새 초대 알림
    } else if (savedInvitations[0] && data.invitations[0].createdAt >= savedInvitations[0].createdAt) {
      // NOTE: 기존 초대 있음 -> 기존 초대 이후에 온 것만 알림
      for (let i = 0; i < data.invitations.length; i += 1) {
        const invitation = data.invitations[i];
        if (invitation.createdAt <= savedInvitations[0].createdAt) break;
        toast.info( 메시지 );
      }
    }
  }
  if (isSuccess) {
    setSavedInvitations(data.invitations);
  }
}, [data]);
```

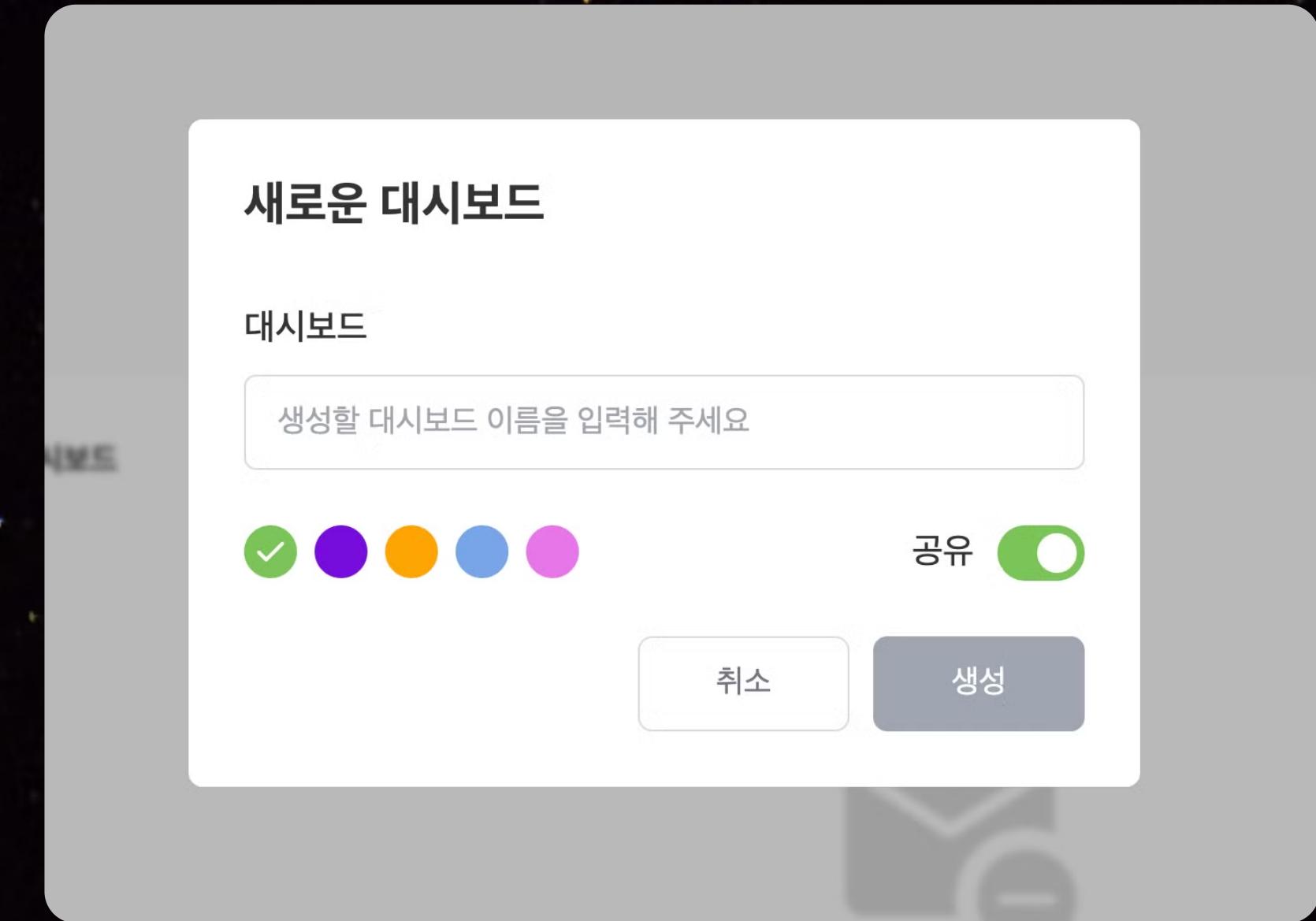
# 공유 기능



초대 받지 못한 멤버도 대시보드를 조회할 수 있는 기능



# 공유 기능



대시보드 생성 시 공유 버튼을 Toggle하여 생성

# 공유 가능

○ ○ ○

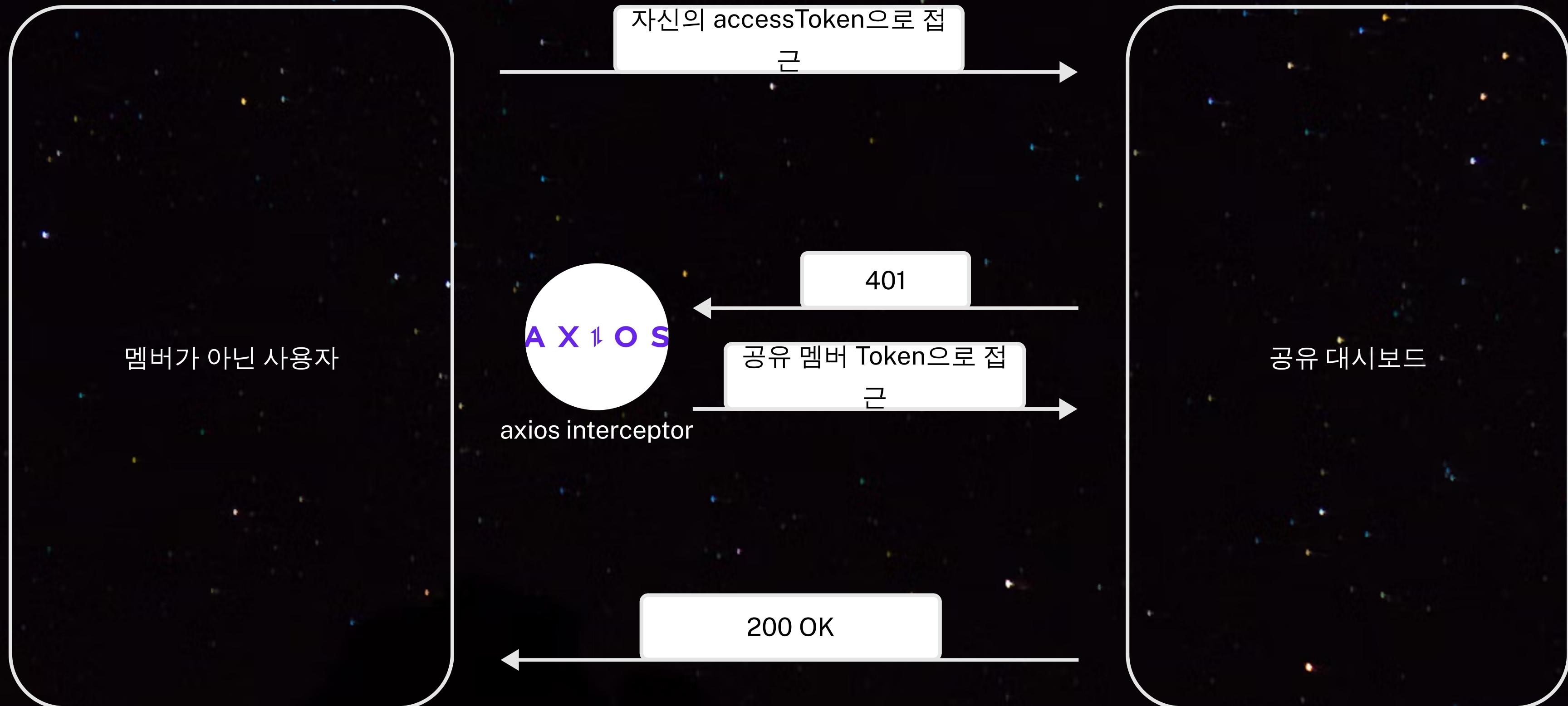
```
const SHARE_EMAIL = String(process.env.NEXT_PUBLIC_SHARE_ACCOUNT_EMAIL);

export const addShareAccount = async (dashboardId: number) => {
  if (!dashboardId) return;

  const { id: invitationId } = await postInviteMember(dashboardId, { email: SHARE_EMAIL });
  await putAcceptInvitation(invitationId, true);
};
```

자동으로 공유 멤버를 해당 대시보드에 초대 및 수락

# 공유 기능



# 공유 기능

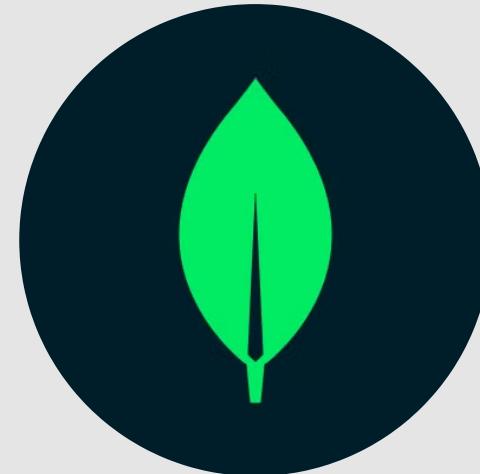
The screenshot shows the Taskify application interface. On the left, there's a sidebar with the Taskify logo at the top. Below it are several navigation items: 'Dashboards' (with a plus icon), 'My Dashboard' (with a clipboard icon), 'My Page' (with a cat icon), '공유 대시보드' (with a crown icon), and '습관 만들기' (with a crown icon). At the bottom of the sidebar is a 'Theme' section with a sun icon. The main area is titled '공유 보드' (Shared Board) in bold black text. It features three columns: 'To do' (0 items), 'On progress' (0 items), and 'Done' (0 items). Each column has a small gear icon for settings and a plus icon to add new tasks. In the top right corner of the main area, there are three user profile icons: a green circle with 'o' and '대', a pink circle with 'i', and a white circle with 'imAdmin'. At the bottom right of the main area is a circular icon containing a tropical scene with palm trees and a sunset.

# 즐겨찾기

- 링크브러리 
- 태스키파이 
- 더줄게 
- 공유 대시보드 
- 습관 만들기 
- 판다마켓
- 팬덤케이
- 롤링
- 공유 보드

기존의 Sidebar는 내가 만든 대시보드 최신, 초대받은 대시보드 최신 순으로 정렬

내가 자주 사용할 것 같은  
대시보드를 고정



**Mongdb Atlas**

# 기능시현

# 회고

원활한 협업을 위해 모두가 부단히 노력

The image shows a GitHub repository interface with several pull requests and commit history. The pull requests include:

- Design: 카드 생성/수정 모달 드롭다운에 다크모드 적용 및 모달 수정 (#217)
- Feat: 할 일 카드 생성 및 수정 모달 수정 사항 (#210)
- Design: 폰트 추가 (#209)
- Feat: 카드 컴포넌트에서 댓글 개수 보여주기 (#207)
- Bug: 이미 초대받은 대시보드의 초대를 또 받을 수 있는 문제 (#200)
- Bug: 초대 알림이 새로운 초대만 오는게 아닌 모든 초대가 옴 (#198)
- Bug: 카드 데이터 퀘리무효화 안됨 (#194)
- Feat: 사이드바 대시보드 즐겨찾기 (#188)
- Bug: 대시보드 관리 페이지에서 대시보드 이름 수정 시 길이 제한 없음 (#185)
- Bug: 대시보드 카드 목록 새로고침이 안됨 (#182)
- Feat: 태그 컴포넌트 추가 기능 (#177)
- Commits on Jul 8, 2024:
  - design(#153): 다크모드 버튼 위치 수정 (#228) - Verified by wjsdnc1 and un0211 committed 1 hour ago - 1/1
  - design(#188): 사이드바 즐겨찾기 글자 수정 (#221) - Verified by wjsdnc1 committed yesterday - 1/1
  - fix(#198): 첫 번째 초대도 알림을 수 있게 변경 (#224) - Verified by un0211 committed yesterday - 1/1
  - fix(#214): 할 일 수정에서 컬럼 변동 시 두 컬럼 모두 새로고침하도록 수정 (#223) - Verified by un0211 committed yesterday - 1/1
- Commits on Jul 7, 2024:
  - fix(#212, #211): 배포 오류 해결 (#222) - Verified by un0211 committed yesterday - 1/1
  - design(#13): 대시보드 컬럼 길이 수정 (#220) - Verified by un0211 committed yesterday - 1/1
  - Fix(#168): 권한에 따른 페이지 접근 제어 (#211) - Verified by un0211 committed yesterday - 1/1
  - feat(#188): 대시보드 즐겨찾기 추가 (#213) - Verified by wjsdnc1 committed yesterday - 1/1
  - refactor(#167): 내 대시보드 페이지 서버 사이드 렌더링 적용 (#199) - Verified by wjsdnc1 committed yesterday - 1/1
  - Fix: 대시보드 페이지 내 버그 수정 (#214) - Verified by wayandway committed yesterday - 1/1
  - design(#217): 다크모드 수정 (#218) - Verified by wjsdnc1 committed yesterday - 1/1

The commit history shows several commits from July 8, 2024, including fixes for deployment errors, sidebar length adjustments, and navigation controls. It also includes a refactor for server-side rendering and a fix for dashboard page bugs.

The pull requests include:

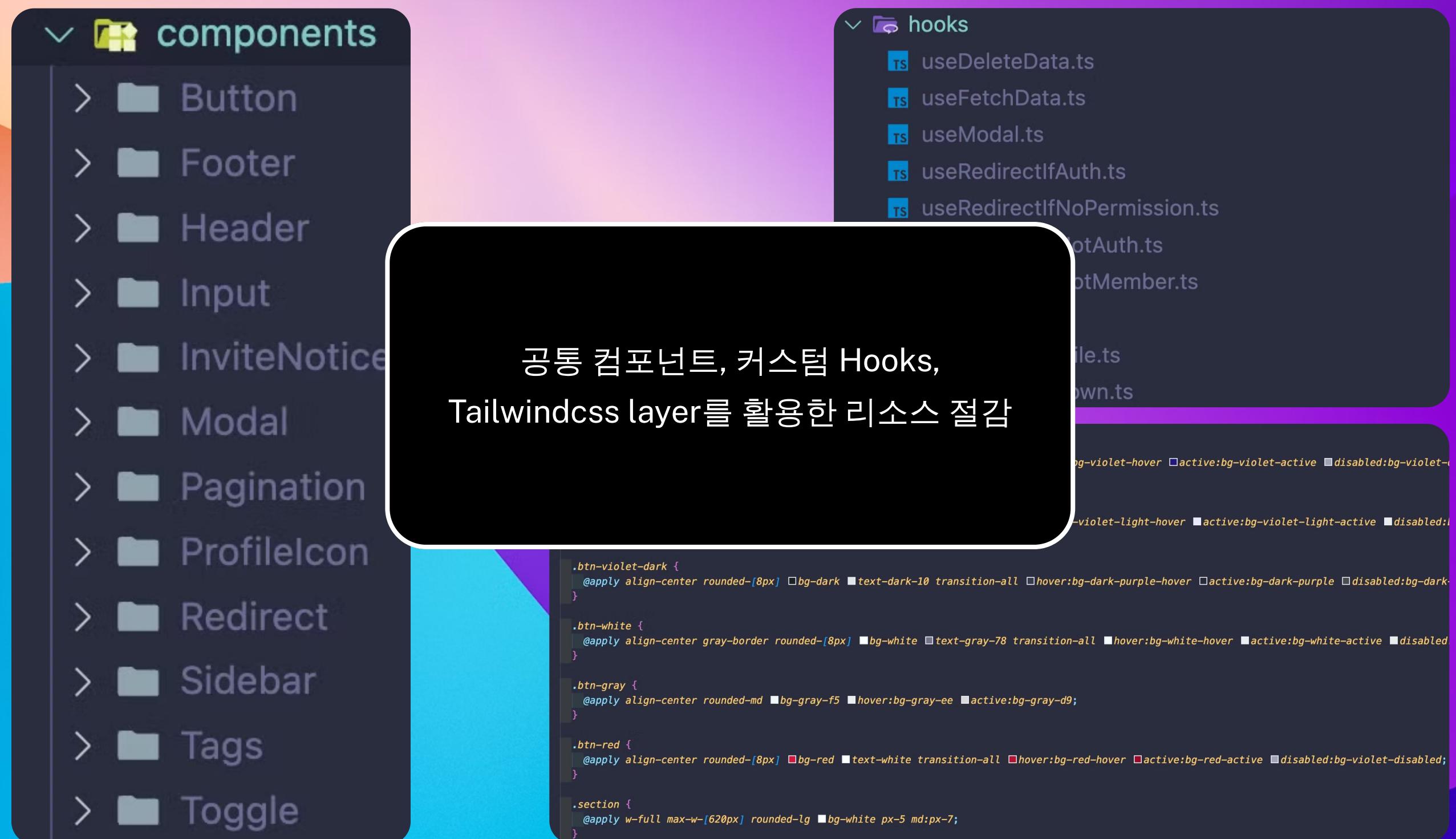
- Feat: 할 일 카드 모달 (#180)
- Feat(#9): Header 컴포넌트 구현 (#78)
- feat(#188): 대시보드 즐겨찾기 추가 (#213)
- refactor(#167): 내 대시보드 페이지 서버 사이드 렌더링 적용 (#199)
- Feat(#11): 할 일 카드 생성 모달 컴포넌트 구현 (#184)
- design(#171): 전체 레이아웃과 대시보드 디자인 문제 수정 (#210)

The commit history shows several commits from July 7, 2024, including fixes for deployment errors, sidebar length adjustments, and navigation controls. It also includes a refactor for server-side rendering and a fix for dashboard page bugs.

The pull requests include:

- Feat: 유저 기능 #28 (#199)
- Feat: 페이지 라우팅 설정 #30 (#200)
- Chore: tailwind 컬러 팔레트 및 반응형 브레이크 포인트 세팅 #33 (#201)
- Chore: eslintrc 변경 #37 (#202)
- Chore: 배포하기 #36 (#203)
- Feat: Columns 섹션 컴포넌트 (대시보드 페이지) #12 (#204)
- Feat: SideBar 컴포넌트 #2 (#205)
- Bug: eslintrc.json 변경에 맞게 패키지 업데이트 #45 (#206)
- Feat: 태그 생성시 컬러 부여 기능 util #26 (#207)
- Feat: 유저 인증이 필요한 데이터 패치 #54 (#208)
- Feat: Column 컴포넌트 (대시보드 페이지) #13 (#209)
- Feat: 대시보드 전역 상태 관리 #52 (#210)
- Refactor: 서비스 로직 분리 #58 (#211)
- Feat: Header 컴포넌트 #9 (#212)
- Bug: commitlint 규칙때문에 이슈번호 커밋메세지에 추가 거절 #51 (#213)
- Feat: MyDashboard 페이지 #15 (#214)
- Feat: 대시보드 목록 컴포넌트 (MyDashboard) #16 (#215)
- Feat: user 관련 저장하는 정보 추가 (이메일, 프로필 이미지) #69 (#216)
- Feat: POST API 로직 분리 #76 (#217)
- Bug: 배포 시 빌드 에러 #79 (#218)
- Feat: 초대받은 대시보드 컴포넌트 (MyDashboard) #20 (#219)
- Feat: 프로필 수정 기능 #8 (#220)

# 회고



# TEAM 15th



류혜원

@un0211

깃 관리, 문서 관리, 헤더, 계정 관리, 대시보드 관리, 초대 알림, 대시보드 공유, 비밀번호 암호화



김지윤

@wayandway

유저 기능, Tanstack query 및 Redux 설정, 대시보드 폐이지, DND



정민재

@wjsdncl

내 대시보드(SSR 적용), 사이드바, 즐겨찾기 기능, 다크모드



이대양

@oceanlee-seoul

전역 모달 관리, 카드 생성 및 수정 모달, 발표





# Want to make a presentation like this one?

Start with a fully customizable template, create a beautiful deck in minutes, then easily share it with anyone.

[Create a presentation \(It's free\)](#)