[프론트 과제수행 가이드]

|  |
| --- |
| 다음은 easyboard-front-blank 프로젝트 코드 중에서 작성해야 할 공백이 있는 파일들입니다.  해당 파일의 TODO 주석에는 구현할 내용에 대한 간단한 설명도 포함하고 있습니다. |

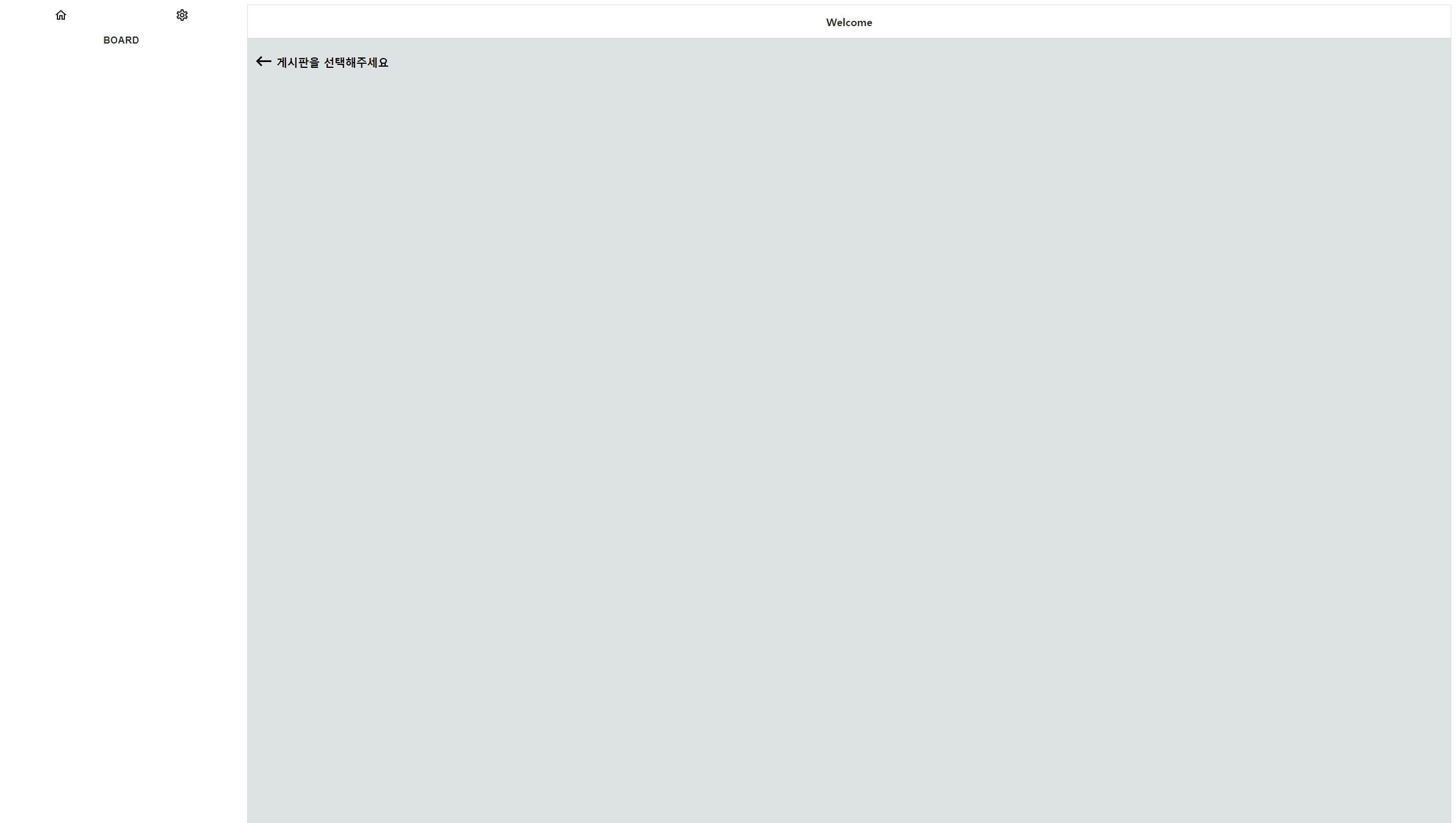
**[화면이동(라우터) 구현]**

번호마다 위치한 이미지는 해당 라우터 구현 후 url을 전환했을 때 보여질 화면입니다.

각 파일은 라우터 외에도 추가 기능 구현 과제가 있습니다.

1. src/pages/index.page.tsx

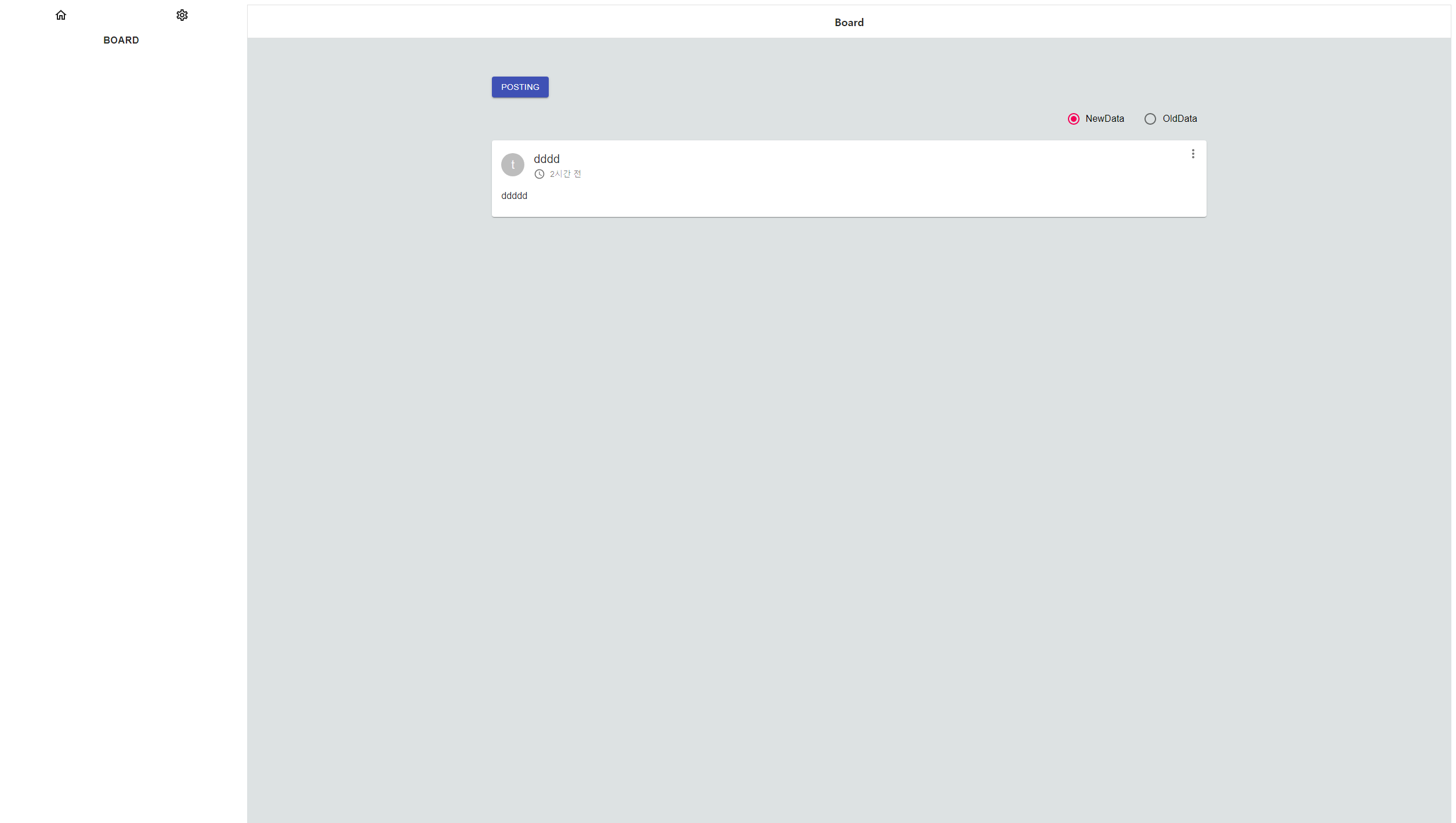
- 생애주기에 따라, 첫 화면(Home)으로 이동하는 라우터를 구현하세요.



2. src/pages/postings/index.page.tsx

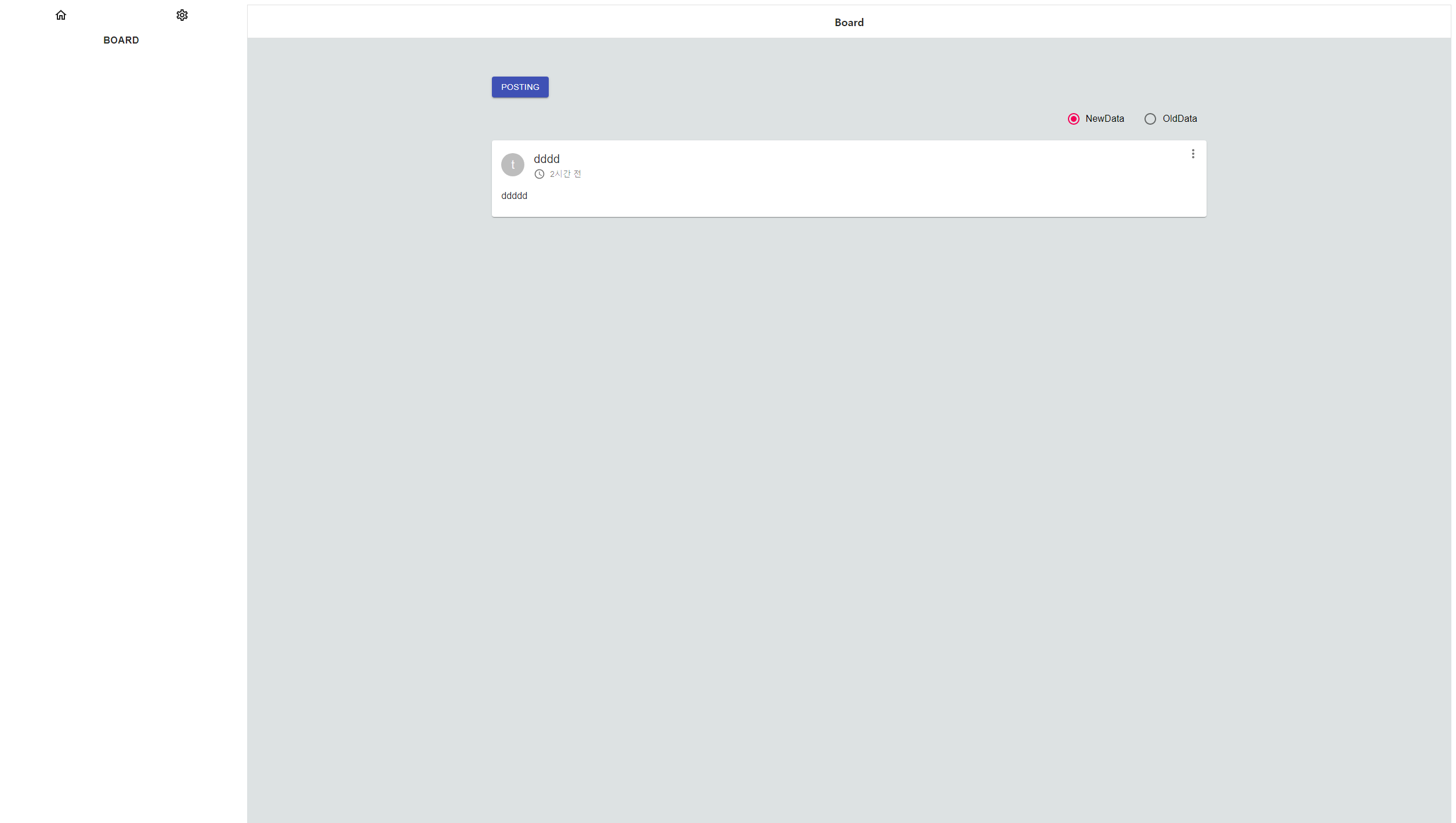
- 게시글 수정과 등록 페이지로 이동하는 라우터를 구현하세요.

- 게시글 등록 페이지로 이동하는 버튼을 추가로 구현하세요.



3. src/comp/view/shared/LayoutSideBar/LayoutSideBarView.tsx

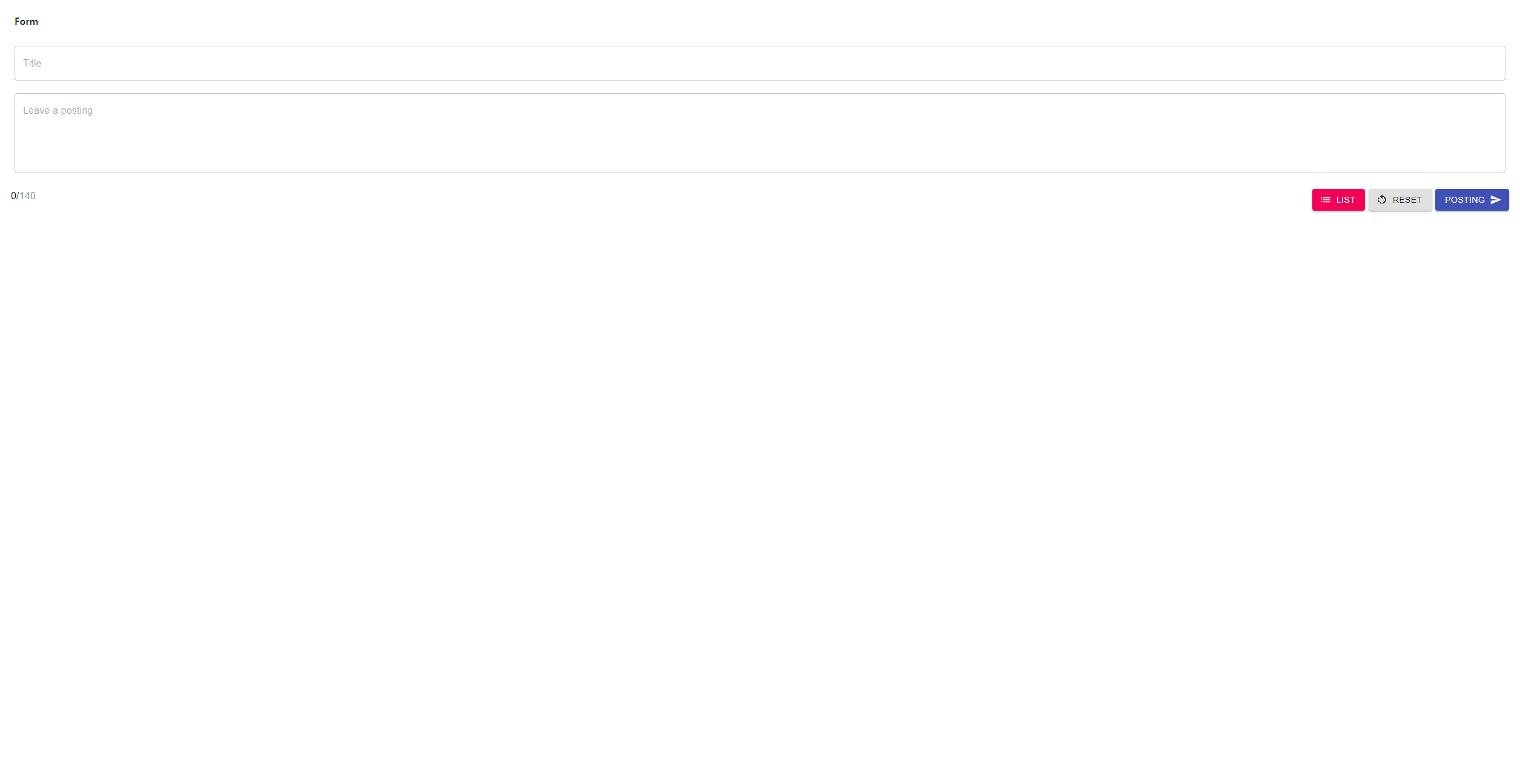
- 사이드바에서 사용할 Home과 게시판 페이지로 이동할 라우터를 각각 구현하세요.



4.

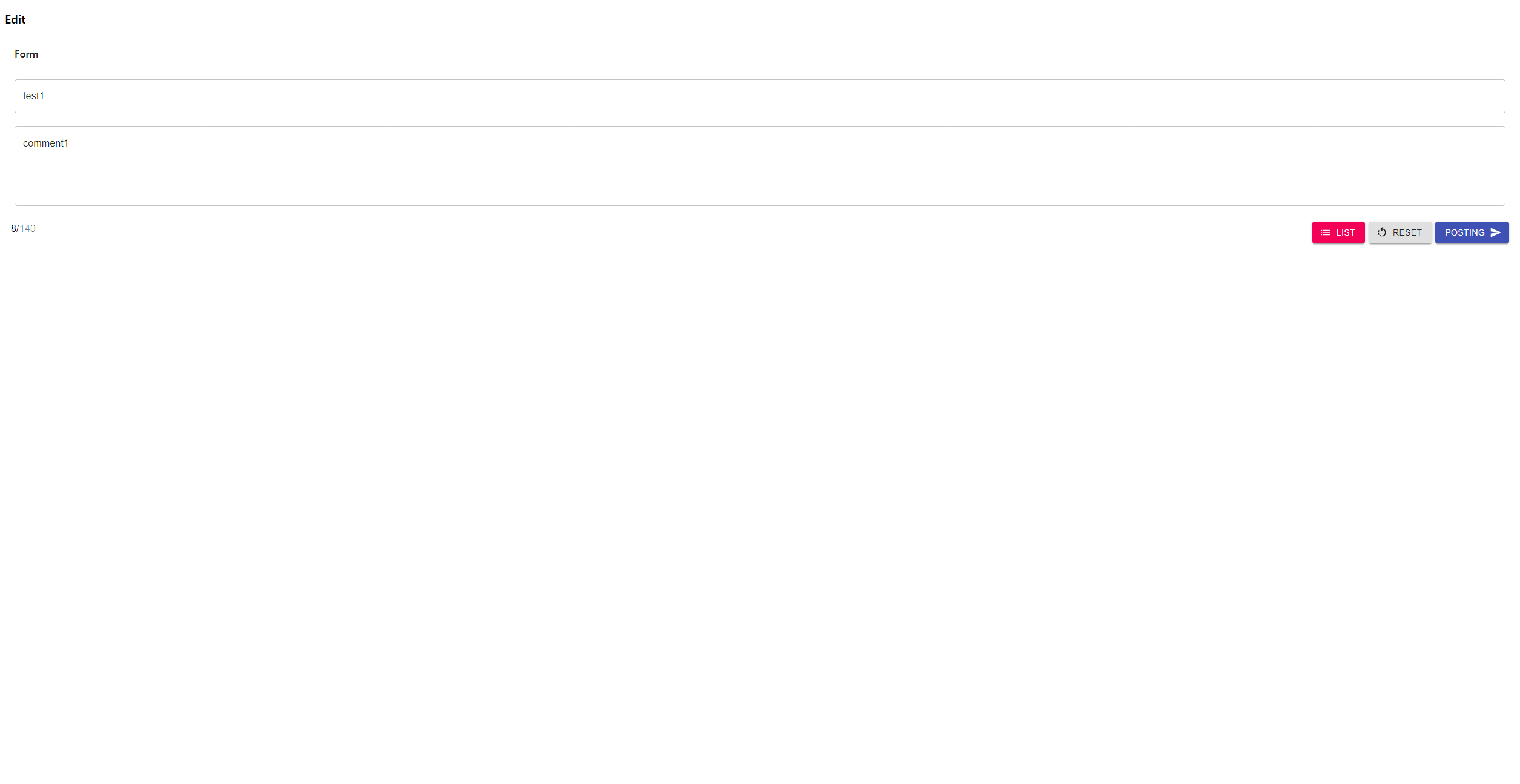
4-1. src/pages/postings/new/index.page.tsx

* 새 게시글을 작성하는 페이지입니다.
* 게시글 목록으로 이동할 라우터를 구현하세요.



4-2. src/pages/postings/edit/[postingId]/index.page.tsx

* EDIT 버튼을 클릭했을 때 전환될 페이지로, 9번문제와도 연관됩니다.
* 게시글 목록으로 이동할 라우터를 구현하세요.
* 화면은 3-1과 동일하지만, 선택한 게시글 데이터가 input 폼 안에 입력돼 있습니다.



**[기능 구현]**

5. src/comp/state/posting/keeper/PostingStateKeeper.ts

6. src/comp/state/posting/keeper/PostingsStateKeeper.ts

- postingId로 게시글 단건의 property 수정

7. src/comp/view/posting/PostingForm/PostingFormContainer.tsx

- initPosting()에서 아래와 같이 postingStateKeeper의 posting 값에 따른 로직을 구현하세요.

1. props의 postingId가 있을 경우, postingId에 대한 posting 검색 후 설정

2. props의 postingId가 undefined일 경우, boardId, userId, writerName으로 posting 설정

onChange()에서 게시글 property를 수정하는 로직을 구현하세요.

onSubmit()에서 Posting 저장 후 결과에 따라 onSuccess/onFail 실행되는 로직을 구현하세요.

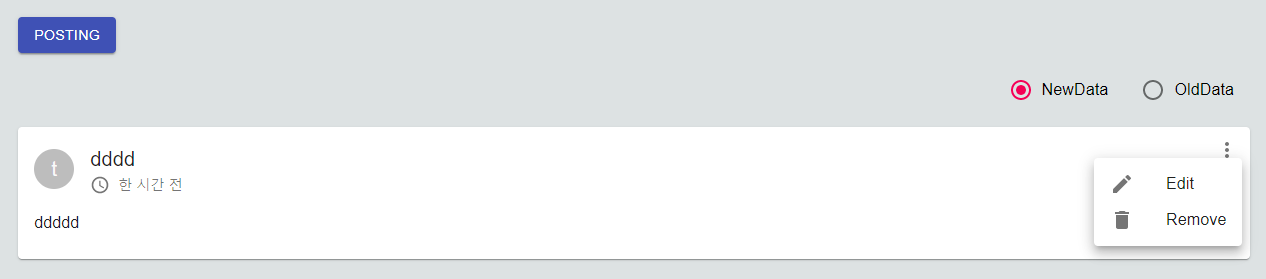
8. src/comp/view/posting/PostingList/PostingListContainer.tsx

* 특정 조건을 만족할 경우, 게시글 목록에서 삭제된 글까지 볼 수 있도록 허용하는 기능입니다.   
  적절한 조건식을 만드세요.
* 이 기능을 실행하는 버튼은 화면 상에 존재하지 않습니다.

9. src/comp/view/posting/PostingList/sub-comp/PostingListContent/PostingListContentContainer.tsx

- 게시글 수정버튼과 삭제버튼 이벤트 함수를 직접 구현하세요.

* 최종적으로, 아래 버튼을 각각 클릭했을 때 실행될 두 가지 이벤트 함수(수정, 삭제)를 구현합니다.



* 구현한 이벤트 함수는 props로 **버튼이 위치한 컴포넌트까지** 전달돼 기능이 동작하도록 하세요.

모두 9문항이며 각 파일의 TODO 부분을 구현한 후 제출해주시면 됩니다.