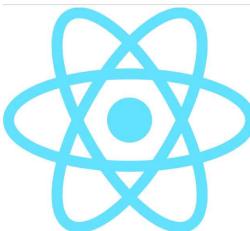


# 미니 블로그 프로젝트(Mini Blog)

## REACT



과 목 명 웹 프로그래밍 기초

학 과 인공지능소프트웨어

학 번 2501110205

성 명 박 경 구

담 당 교 수 민 경 주

제 출 일 2025.12.09

## ■ 프로젝트 개요

- 과제명 : 기말고사 대체 과제 (Term Project)
- 주제 : 홈페이지 제작 및 2페이지 이상을 React로 작성 (미니 블로그 구현)
- GitHub : [https://github.com/Goldstargreat/Mini\\_Blog](https://github.com/Goldstargreat/Mini_Blog)
- 커밋 메시지: ★ 기말고사과제 ★

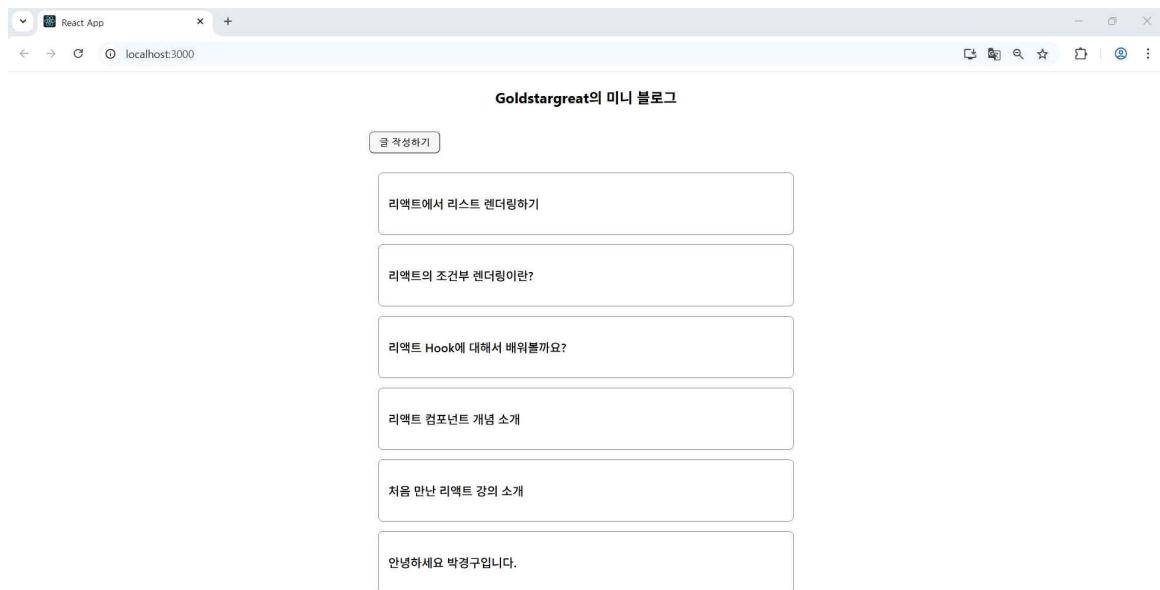
## ■ 주제 선정 이유

- 본 과제는 기존에 HTML, CSS, JAVASCRIPT로 제작했던 홈페이지의 대표 페이지를 React 함수형 컴포넌트를 사용하여 재구현하는 것을 목표로 한다. 단 본 과제는 기존의 홈페이지와는 별개로 작성되었으며, 기술적으로는 리액트에 초점을 맞추고 있기에 백엔드에 해당하는 부분은 더미 데이터로 대체한다. 더미 데이터를 사용하는 이유는 본 과제에서 데이터베이스와 서버를 구축하는 부분까지는 다룰 수 없기 때문이다. 더미 데이터는 JSON 형태로 되어있는데 현재 서버와의 통신에서 가장 많이 사용하는 데이터 유형이다. 본 과제의 웹 애플리케이션은 데이터베이스와 API가 연동되어 있지 않기 때문에 실제로 글, 댓글이 작성되지는 않는다. 미니 블로그를 주제로 선정한 사유는 다음과 같다.

- \* 선정 페이지: 네이버의 블로그를 모방하여 게시판 및 방명록 기능을 통합한 미니 블로그 형태를 선정하여 메인페이지, 글 보기 페이지, 글 작성 페이지 등을 구현했다.
- \* 강의 시간에 학습한 함수형 컴포넌트 및 Hooks (특히 useState, useEffect) 사용 능력을 숙달한다.
- \* React Router를 이용해 SPA (Single Page Application) 환경에서 자연스러운 페이지 이동(라우팅)을 구현한다.

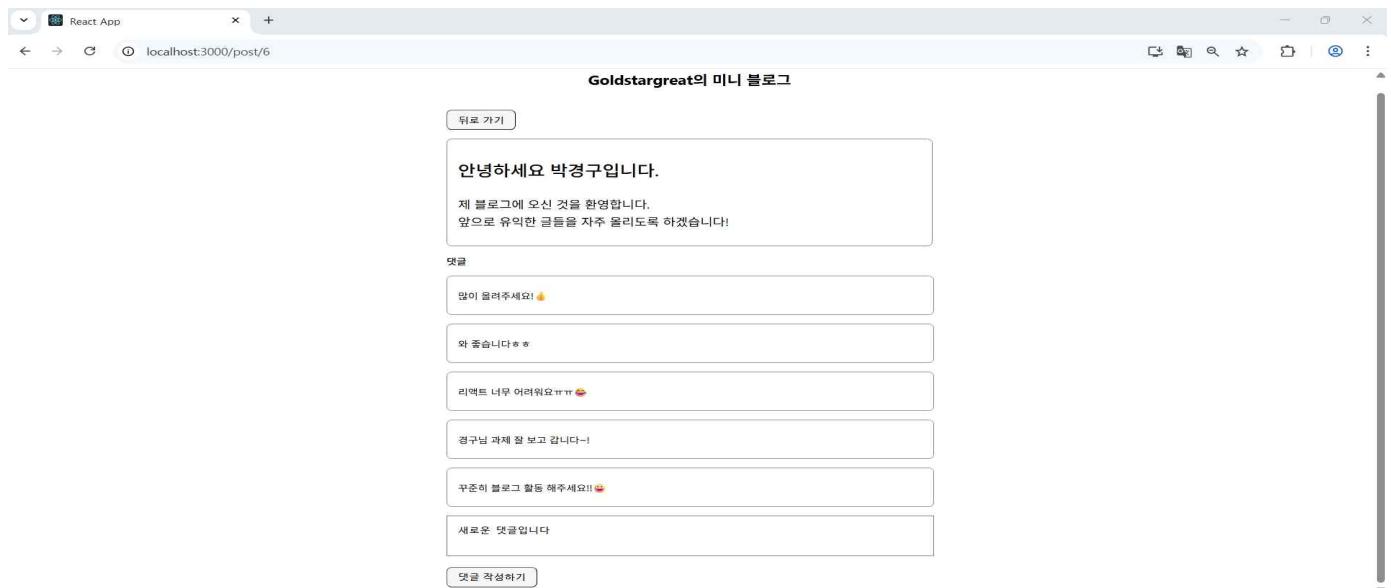
## ■ 구현 페이지 분석 및 설명

### ○ 메인 홈페이지 (Main Page)



- ◆ 기술: React-Router-Dom을 통해 라우팅을 구성했다. 라우팅은 사용자가 원하는 경로로 보내는 과정이다. JSON 파일로 게시물 목록을 동적으로 생성했다. React Router를 사용하여 각 게시물 항목 클릭 시 상세 페이지로 이동하도록 라우팅을 설정했다.
- ◆ UI/UX 편의성: 블로그의 주요 포스트를 한눈에 확인할 수 있도록 깔끔한 카드 형태로 디자인한다. '글 작성하기' 버튼을 상단에 배치하여 사용자로 하여금 새로운 콘텐츠 생성에 쉽게 접근할 수 있도록 한다.

## ○ 포스트 상세 페이지(Post Detail & Comment)



- ◆ 기술: 동적 라우팅 (e.g., /post/:postId)을 구현하여 URL 파라미터에 따라 해당 ID의 포스트 내용을 불러오도록 한다. `useState` Hook을 사용하여 댓글 목록과 새로운 댓글 입력 상태를 관리한다. 댓글 역시 배열 렌더링을 통해 출력된다.
- ◆ UI/UX 편의성: 본문 내용과 댓글 영역을 구분하여 가독성을 높인다. '뒤로 가기' 버튼을 제거하여 사용자가 쉽게 이전 목록으로 복귀할 수 있도록 구현한다. 댓글 작성 후 즉시 목록에 반영되도록 했다.

## ○ 포스트 작성 페이지 (Post Write)



- ◆ 기술: `useState` Hook을 사용하여 제목과 본문의 입력 값 상태를 각각 관리하는 품 핸들링을 구현한다. 이벤트 핸들러 (`onChange`, `onClick`)를 통해 입력값 변경 및 '글 작성하기' 버튼 클릭 이벤트를 처리하며, 작성 완료 후 메인 페이지로 라우팅되도록 했다.
- ◆ UI/UX 편의성: 제목과 내용 입력 영역을 구분하여 사용자가 작성할 내용을 인지할 수 있도록 했다. '글 작성하기' 버튼을 하단에 배치하여 작성 완료 후 바로 제출할 수 있게 했다.

## ■ 결론

이번 과제를 통해 웹프로그래밍 기초 수업에서 배운 React의 핵심 개념을 실제 애플리케이션으로 구현해 보는 흥미로운 경험을 했다.

함수형 컴포넌트와 Hooks 활용: 과제 요구사항에 맞춰 모든 UI를 함수형 컴포넌트로 작성하고, useState를 이용한 데이터 상태 관리, 라우팅 처리를 통해 페이지 전환을 구현했다. 특히, 댓글 기능과 포스트 작성 시 입력 데이터의 상태 변화를 실시간으로 추적하는 과정에서 React의 단방향 데이터 흐름에 대한 이해도를 높일 수 있었다.

컴포넌트 기반 개발의 이해: 포스트 목록, 상세 내용, 댓글 목록, 댓글 작성 폼 등을 각각 독립적인 컴포넌트로 분리하고 조합함으로써, 코드의 재사용성과 유지 및 보수성이 얼마나 중요한 것인지 직접 체감할 수 있었다.

CSS를 작성 및 수정하는 게 나름 어려웠는데 jsx 파일로 애플리케이션의 골격을 먼저 만들어 놓고 css는 별도의 파일을 만드는 것으로 가독성을 높였다.

현재는 프론트엔드 내에서 정의된 가짜 데이터(data.json)를 사용하고 있다. 추후에는 백엔드 서버와 API 통신 (예: fetch 또는 axios)을 통해 데이터를 주고받는 기능을 구현하여 CRUD (Create, Read, Update, Delete) 로직을 완성하고 싶다.