

MERN Stack TodoList

김동영

CONTENTS

1

기획의도

2

사용기술

3

레이아웃

4

느낀점



1

기획의도

프로젝트명 : MERN TodoList

프로젝트 팀구성: 김동영

일정 (전체, 주간, 일간 등) 총 3일

기획의도: 풀 스택 개발자를 목표로 시작한 React공부를 하면서 쌓은 지식을 활용하여
제목, 작성자, 사용기술, 완료여부로 구성된 TodoList를 구현하기 위하여
시작하게 되었습니다.
새로고침 시 초기화 되지 않도록 NodeJS용 Framework인 ExpressJS, MongoDB와
연결하여 구현하는 것을 목표로 하였습니다.

2

사용기술

MERN은 MongoDB ExpressJS React NodeJS의 약자로서
MEAN(MongoDB ExpressJS AngularJS NodeJS)이후에 나왔으며
JavaScript Front-End Framework로 구성하는 총체적 기술스택입니다.

Mongo DB: 대표적인 No-SQL입니다.

ExpressJS: 웹사이트 개발을 위한 NodeJS용 Framework입니다.

React: 페이스북에서 제공하는 JavaScript UI Library입니다.

NodeJS: JavaScript를 웹 브라우저 밖에서 사용할 수 있게 해주는 프로그램입니다.



2

사용기술

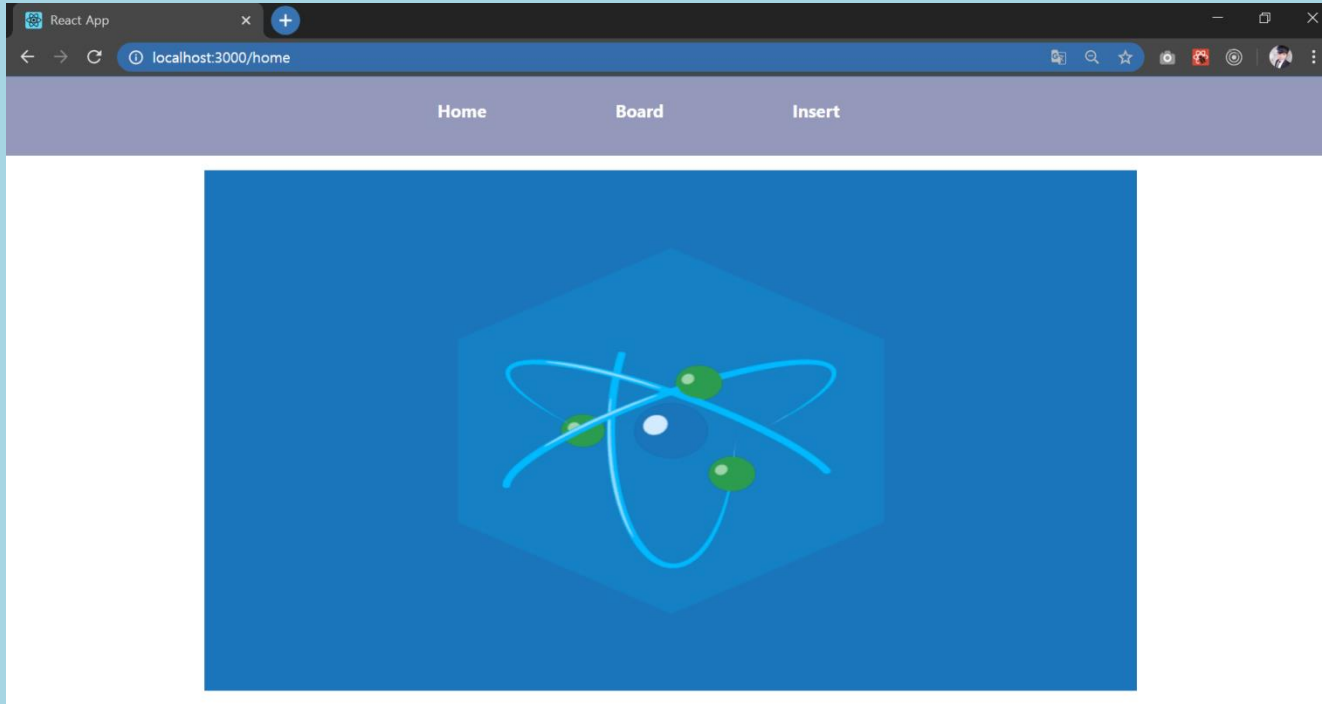


Material-UI는 React 기반 UI 라이브러리 중에 가장 인기 있는 라이브러리 입니다.
Google Material Design 기반으로 제작 되었습니다.

신속하고 편리하게 스타일링을 적용하기 위해서 이번 프로젝트에 Material-UI를 사용하였습니다.

3

레이아웃

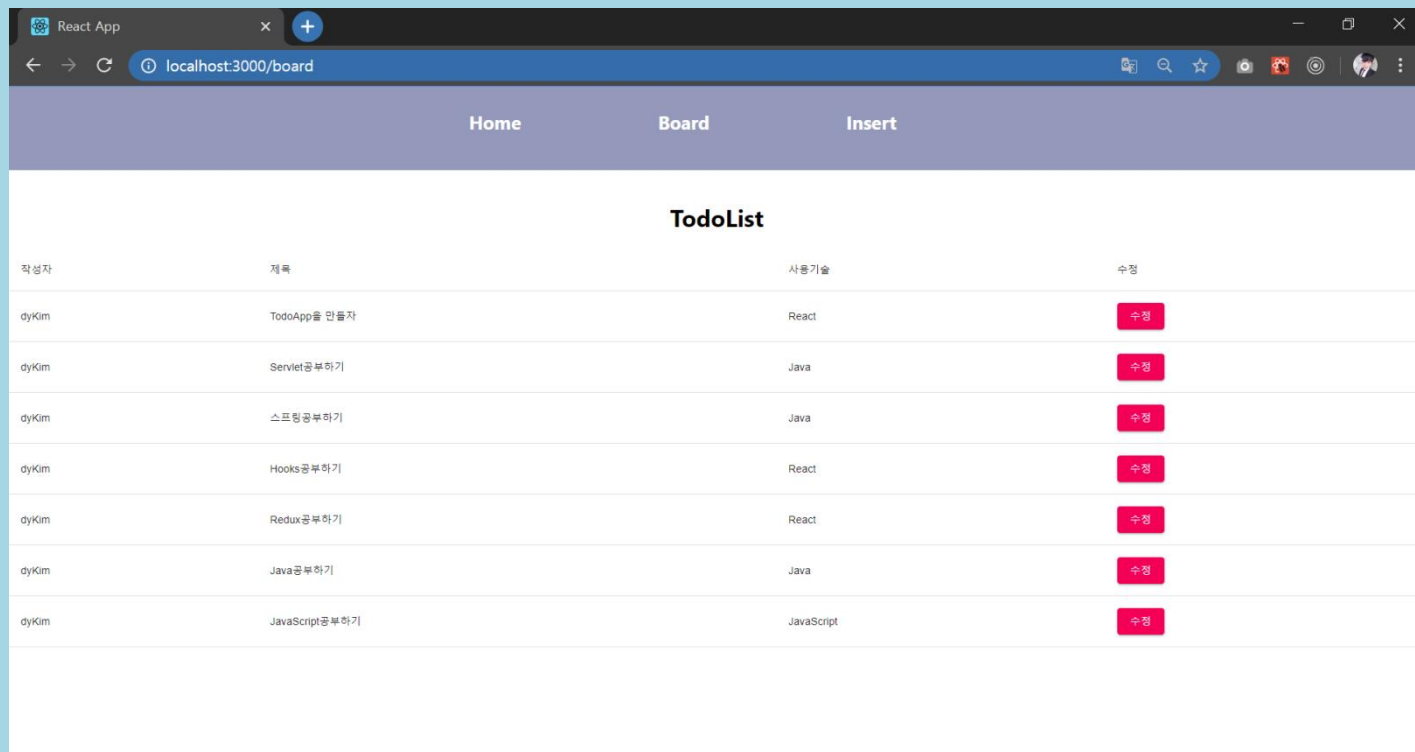


홈 화면

Router, 클래스형 컴포넌트로 Home, Board, Insert로 이동할 수 있도록 구현하였습니다.

3

레이아웃



ToDoList			
작성자	제목	사용기술	수정
dyKim	ToDoApp을 만들자	React	수정
dyKim	Servlet공부하기	Java	수정
dyKim	스프링공부하기	Java	수정
dyKim	Hooks공부하기	React	수정
dyKim	Redux공부하기	React	수정
dyKim	Java공부하기	Java	수정
dyKim	JavaScript공부하기	JavaScript	수정

리스트 화면

axios로 backend Sever의 데이터를 get방식으로 불러와서 Client Server에 React의 map을 사용하여 반복하면서 View에 출력하였습니다.

3

레이아웃

The screenshot shows a web browser window with the address bar at `localhost:3000/create`. The application has a purple header with three navigation links: **Home**, **Board**, and **Insert**. The main content area is titled **Write New Todo** and contains the following elements:

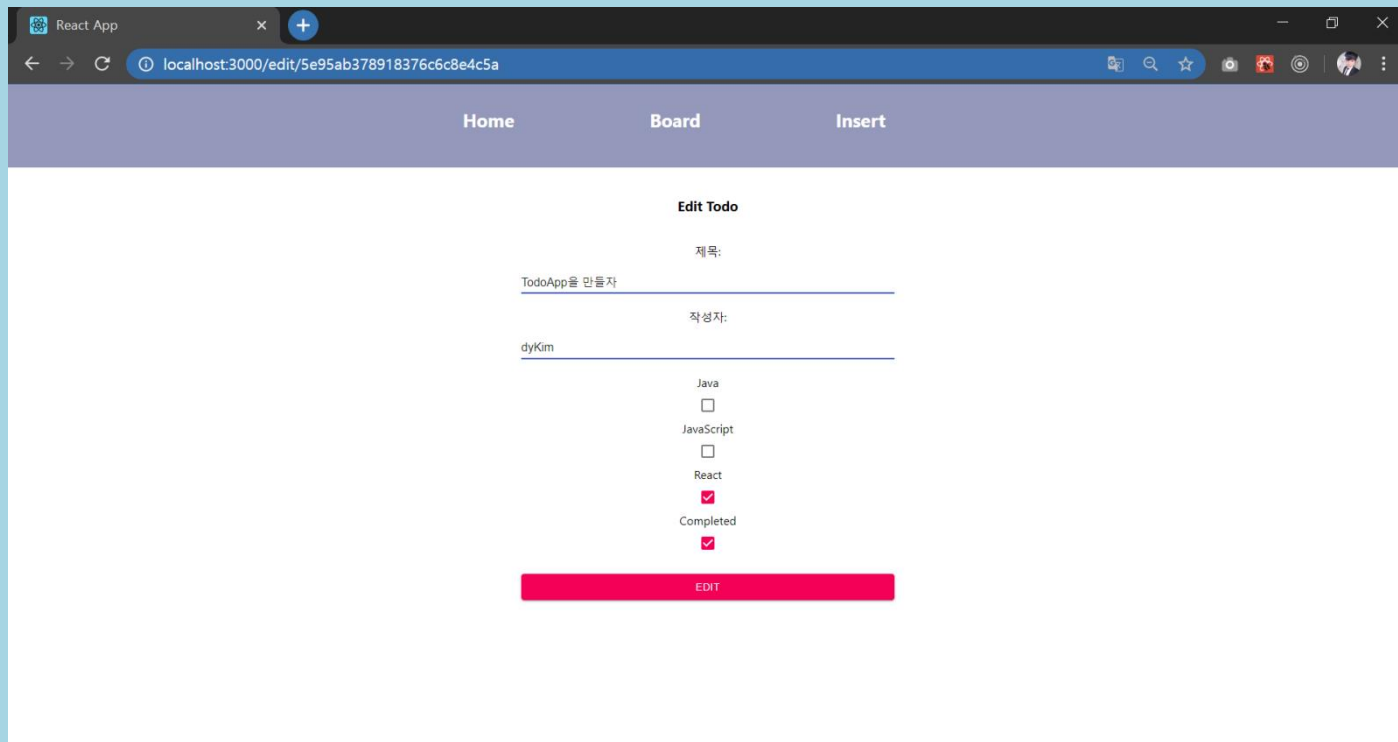
- A label **제목:** followed by a text input field.
- A label **작성자:** followed by a text input field.
- A radio button labeled **Java**, which is currently selected (checked).
- A radio button labeled **JavaScript**, which is currently unselected.
- A radio button labeled **React**, which is currently unselected.
- A red **SAVE** button at the bottom.

글쓰기 화면

입력한 데이터를 axios의 post방식으로 backend Server에 전달하여 MongoDB의 데이터베이스에 저장시키도록 구현하였습니다.

3

레이아웃



글수정 화면

리스트에서 Edit버튼 클릭시 props로 전달받은 Id를 Link 태그와 함께 primary key로 전달하여 axios의 get방식으로 데이터를 불러온 후, post 방식으로 update하도록 구현하였습니다.

React를 공부하고 처음으로 구현한 프로젝트였기 때문에 힘들었지만, 추가로 ExpressJS서버와 MongoDB에 저장시키는 방법까지 경험 할 수 있었던 좋은 경험이었습니다.

현재 공부중인 Hooks와 Redux가 익숙해지면 클래스형 컴포넌트 대신에 함수형 컴포넌트로 다음 프로젝트를 구현할 예정이며, 다양한 언어를 경험하는 것이 해당 언어 이외의 다른 언어에도 도움이 된다고 느꼈습니다.

THANK YOU