

고급소프트웨어실습1

Lecture 7 과제

반 : 4

학번 : 20171669

이름 : 이재영

1. 실습 환경

고급소프트웨어 실습 7주차에서는 웹 프로그래밍을 진행한다. 웹 프로그래밍이란 웹 페이지 상에 보이는 화면은 구성하는 것을 만들어 내는 작업을 말한다. 웹 프로그래밍을 구현하는 많은 방법 중에서 7주차에서는 React를 사용한다. React란 페이스북에서 제공하는 javascript library로 어플리케이션의 동적 기능 구현을 가능케 해주고 프론트엔드와 백엔트 프로그래밍을 구현이 가능하게 해준다.

React의 특징으로는 data flow가 단일 방향으로만 흐르는 one-way data flow의 특징을 가지며 independent한 컴포넌트로 UI가 구성된다. 컴포넌트로는 버튼, 레이아웃 등이 있으며 원하는 컴포넌트만 수정이 가능하며 재사용도 가능하며 관리와 유지보수에 유리하다.

2. 실습 내용 및 코드 설명

Npm start로 시작하면 다음과 같은 화면이 나온다.

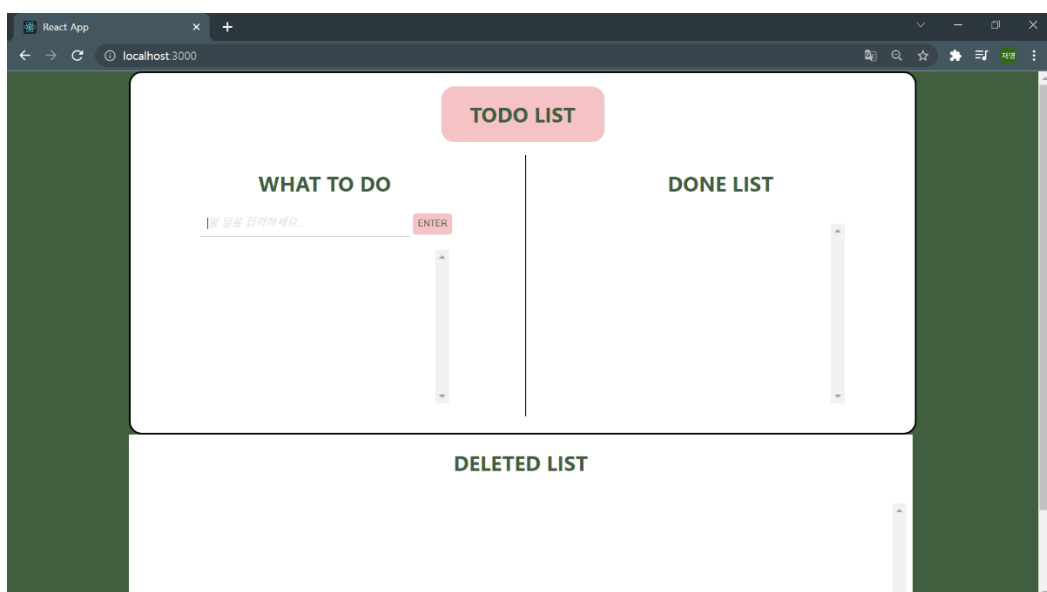
```
root@427371dfec84: /workspace/react-todolist-student
Compiled successfully!

You can now view study-todo in the browser.

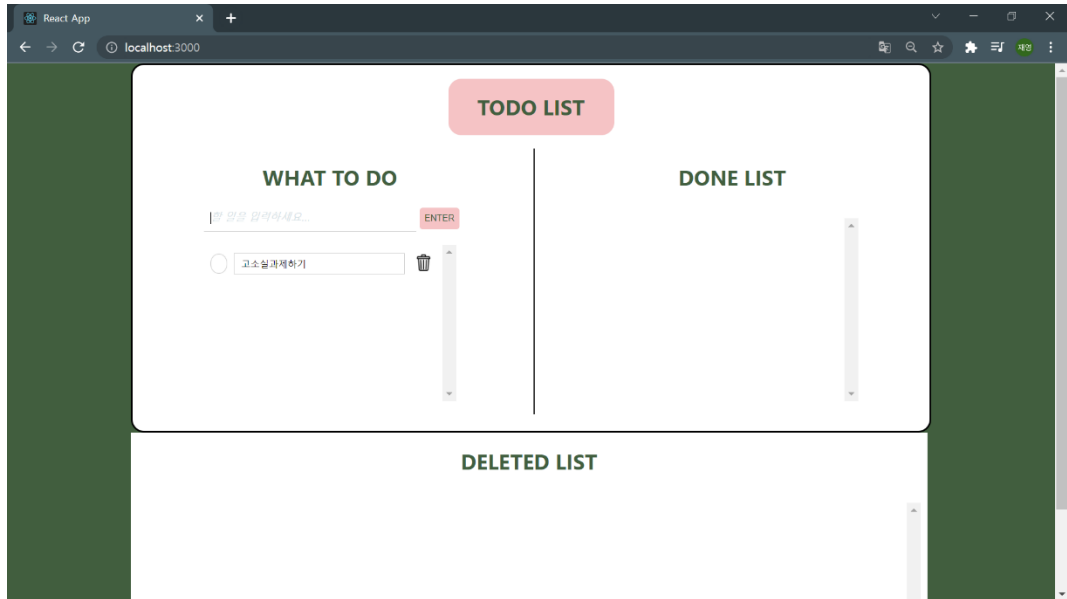
Local:      http://localhost:3000
On Your Network:  http://172.17.0.2:3000

Note that the development build is not optimized.
To create a production build, use npm run build.
```

Compile이 제대로 된 것을 확인할 수 있고 웹에서 localhost:3000로 들어가면 다음과 같은 화면이 나온다.

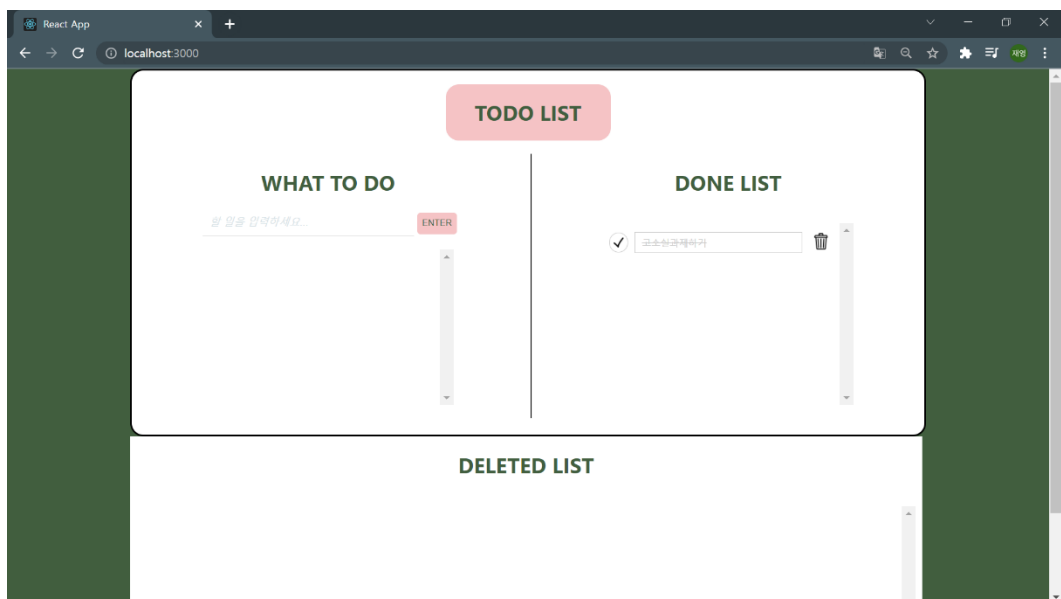


“고소실 과제하기”를 what to do에 추가하면 제대로 추가되는 것을 확인할 수 있고



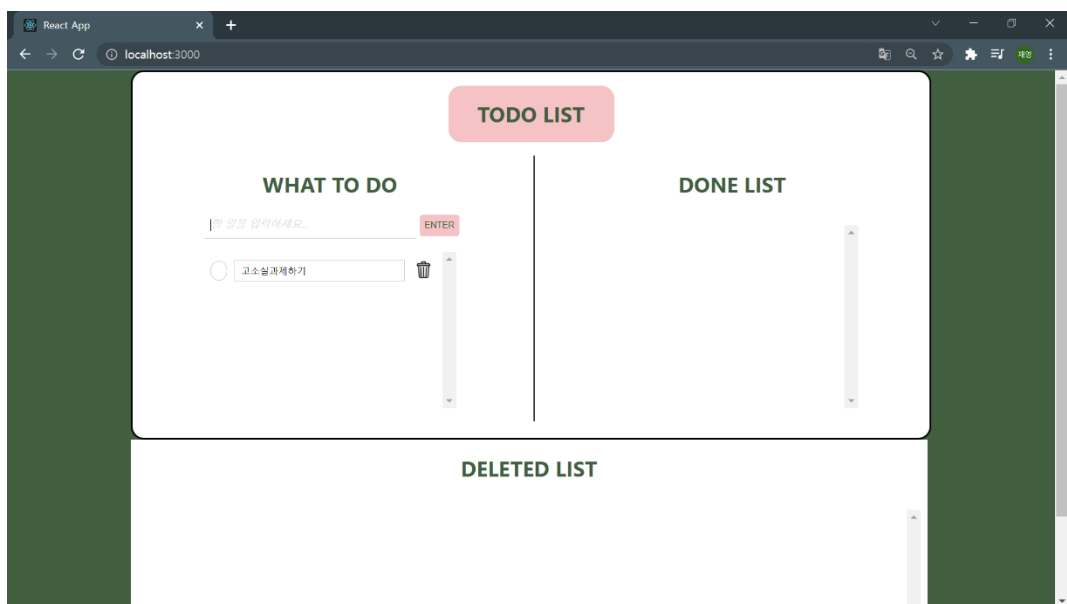
- TodoTemplate.jsx에서 onToggleToDo 함수를 구현해보면 doneTodo라는 const 변수에 todos에서 체크한 것과 같은 것, 즉 id가 같은 것을 찾아 낸 후, Donetodo로 옮겨 옮긴 것을 나타내는 checked를 1로 바꾸면서 구현했다. 이제 WHAT TO DO에서 체크를 하면 Done List로 이동한 것을 확인할 수 있다.

```
// 주어진 What To Do 항목(체크만된 항목)을 찾아 항목 객체의 속성(키)을 사용하여 체크한 후, Done List에 추가하고 What To Do에서 제외
const onToggleToDo = (id) => {
  // 7주차 실습 구현
  const doneTodo = todos.find((todo) => todo.id === id);
  doneTodo.checked = 1;
  setDoneTodos([...doneTodos, doneTodo]);
  setTodos(todos.filter((todo) => todo.id !== id));
};
```



- TodoTemplate.jsx에서 onToggleDoneToDo함수는 DONE LIST에 있는 것을 다시 WHAT TO DO로 옮기는 함수로 onToggleToDo와 비슷하지만 이동 대상이 DONE LIST에서 WHAT TO DO로 바뀐 것만 구현하면 된다. 이 때, DONE이 아니므로 checked는 0으로 바꾼다. 이제 다시 체크를 누르면 DONE LIST에서 WHAT TO DO로 옮겨간 것을 확인할 수 있다.

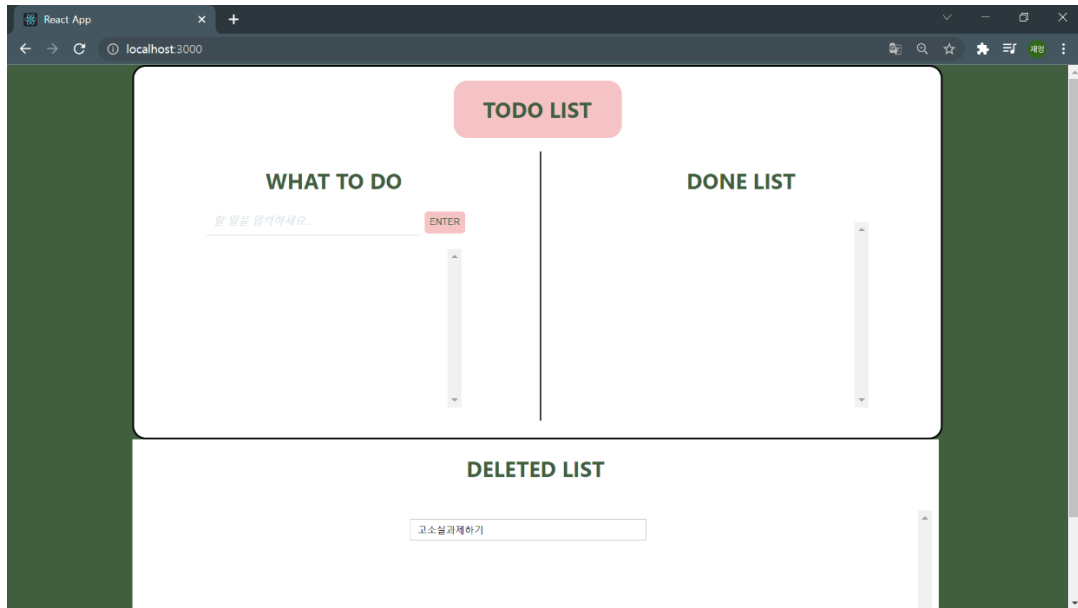
```
// 주어진 Done List 안 항목을 찾아 항목 객체의 속성을 사용하여 체크 해제한 후, What To Do에 추가하고 Done List에서 제외
const onToggleDoneToDo = (id) => {
  // 7주차 실습 구현
  const DoneToDo = doneTodos.find((todo) => todo.id === id);
  DoneToDo.checked = 0;
  setTodos([...todos, DoneToDo]);
  setDoneTodos(doneTodos.filter((doneTodo) => doneTodo.id !== id));
};
```



- TodoTemplate.jsx에서 onRemoveDoneToDo함수는 DONE LIST에서 DELETED LIST로 옮긴 후 기존 DONE LIST에서 삭제해주는 함수이다. doneTodos에서 해당하는 id의 항목을 찾은 후 setDeletedTodos를 이용해서 DELETED LIST로 옮기고 DONE LIST에서 해당하는 것을 지워주는 것을 구현했다. 휴지통버튼을 누르면 DELETED LIST로 이동한 것을 확인할 수 있다.

```
// 주어진 Done List 항목(체크된 항목)의 id로 Done List에서 해당 항목을 찾고 Deleted List에 추가한 후에, 기존 Done List에서 제외
const onRemoveDoneToDo = (id) => {
  // 7주차 실습 구현
  const RemoveDoneToDo = doneTodos.find((doneTodo) => doneTodo.id === id);

  setDeletedTodos([...deletedTodos, RemoveDoneToDo]);
  setDoneTodos(doneTodos.filter((doneTodo) => doneTodo.id !== id));
};
```



3. 실습 설명

컴포넌트들은 각각 jsx파일로 구성되어있고 삭제된 항목을 보여주는 DeletedList.jsx 컴포넌트, 삭제된 항목 하나를 표현하는 DeletedTodoItem.jsx 컴포넌트가 있고 이들은 Delete한 것에 대한 컴포넌트이다. 체크된 항목들을 보여주는 DoneList.jsx 컴포넌트, 체크되지 않은 항목에 대한 WhatToDo.jsx 컴포넌트가 있고 이는 체크하는 행동을 통해 서로 연관이 있다. 항목의 입력창 및 Enter의 TodoInsert.jsx 컴포넌트와 할 일 항목 하나에 대한 컴포넌트인 TodoItem.jsx 는 각각 What To do, DoneList 등에서 연관을 보인다. TodoTemplate.jsx 컴포넌트의 경우에는 각 컴포넌트를 감싸는 전체 컴포넌트로 행동에 대해 각 컴포넌트끼리의 연관을 보여준다.