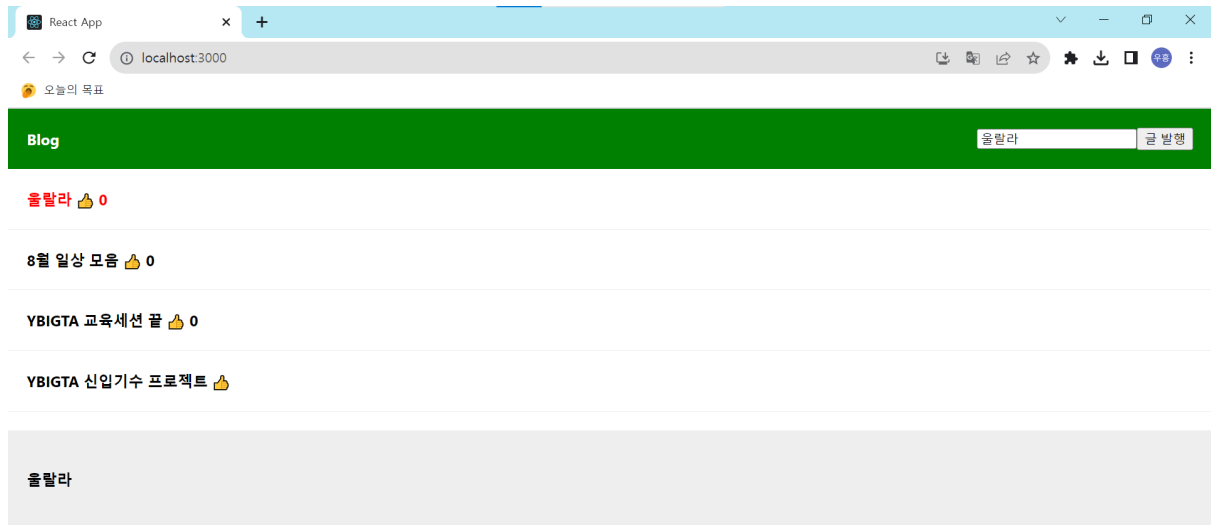


React과제



```
// App.css 파일 불러오기
// ----- 이 주석을 지우고 여기에 코드 작성 -----
import './App.css'
// -----

// React, useState 불러오기
// ----- 이 주석을 지우고 여기에 코드 작성 -----
import React, { useState } from 'react';
// -----

function App() {

  const [title, setTitle] = useState([
    "8월 일상 모음",
    "YBIGTA 교육세션 끝",
    "YBIGTA 신입기수 프로젝트",
  ]);
  const [good, setGood] = useState([0, 0, 0]);
  const [modal, setModal] = useState(false);
  const [changeTitle, setChangeTitle] = useState(0);

  // 작성내용을 저장하는 상태관리 변수 생성
  // ----- 이 주석을 지우고 여기에 코드 작성 -----
  const [inputValue, setInputValue] = useState("");
  // -----

  // 내용 클릭하면 색상 변하는 상태관리 변수 생성
  // ----- 이 주석을 지우고 여기에 코드 작성 -----
  const [activeColor, setActiveColor] = useState(Array(title.length).fill(false));
  // -----

  return (
    <div className="App">
      <div className="black-nav" style={{backgroundColor: 'green'}}> {/*배경색 다른색으로 수정*/}
        <div>
          <h4 style={{ color: "white", fontSize: "16px" }}>Blog</h4>
        </div>
        <div className="submit">
          {/*글 발행 버튼 누르면 작성한 내용이 작성내용 상태관리 변수에 업데이트 되는 코드 작성*/}
          <input
            type="text"
            onChange={(e) => setInputValue(e.target.value)}
            value={inputValue}
          />
        </div>
      </div>
    </div>
  );
}
```

```

    />
    <button
      onClick={() => {
        let copy = [...title]; // 현재 타이틀 배열을 복사
        copy.unshift(inputValue); // 작성된 내용(inputValue)을 복사한 배열의 맨 앞에 추가
        setTitle(copy); // 타이틀 배열 상태를 업데이트하여 새로운 글이 목록에 추가되도록 함
      }}
    >
      글 발행
    </button>

  </div>
</div>
{title.map(function (a, i) {
  return (
    <div className="list">
      <h4
        onClick={() => {
          // title 클릭 시 상태 변경 코드 작성
          // ----- 이 주석을 지우고 여기에 코드 작성 -----
          setModal(!modal);
          setChangeTitle(i);
          //
          const newActiveColor = [...activeColor];
          newActiveColor[i] = !newActiveColor[i];
          setActiveColor(newActiveColor);
          //
        }}
        // 클릭되면 글자색 다른색으로 변경되는 코드 작성
        // ----- 이 주석을 지우고 여기에 코드 작성 -----
        style={activeColor[i] ? { color: "red" } : null}
      >
        {title[i]} {/*현재 인덱스의 타이틀을 표시*/}
        <span
          onClick={(e) => {
            e.stopPropagation(); // 상위 요소에 이벤트가 전파되지 않도록 함
            let copy = [...good]; // 현재 좋아요 수 배열을 복사
            copy[i] = copy[i] + 1; // 현재 인덱스의 좋아요 수를 1 증가
            setGood(copy); // 좋아요 수 배열 상태를 업데이트
          }}
        >
          {" "}
          🍌{" "} {/*좋아요 아이콘 표시*/}
        </span>
        {good[i]} {/*현재 인덱스의 좋아요 수를 표시*/}
      </h4>
    </div>
  );
});
}
{modal === true ? ( // modal 상태가 true인 경우
  <Modal
    title={title} // title 배열을 Modal 컴포넌트에 전달
    changeTitle={changeTitle} // 현재 선택된 타이틀의 인덱스를 Modal 컴포넌트에 전달
  ></Modal>
  ) : null} {/*modal 상태가 false인 경우, 아무것도 렌더링하지 않음*/}
</div>
);
}

function Modal(props) {
  return (
    <div className="modal"> {/*modal 클래스를 가진 div로 묶음*/}
      <h4>{props.title[props.changeTitle]}</h4> {/*props로 전달받은 title 배열에서 changeTitle 인덱스에 해당하는 값을 h4 태그로 표시*/}
    </div>
  );
}

export default App;

```