

16 검색 및 정렬 기능 구현 (사용자 입력 기반)

App. js 상품검색 품과 사용자 입력을 통한 정렬기능			
순서	작업환경	명령어	
1	App.js	수정할 부분 이미지 참고	<div><div>localhost:3000</div><div>Suplelin</div><div>1</div><div>상품명을 입력하세요</div><div><div>1</div><div>수박</div><div>당도선별 프리미엄 고당도 하우스수박 5~6kg</div><div>29000</div></div><div><div>2</div><div>참외</div><div>산지직송 성주 달콤참외 3kg</div><div>16900</div></div><div><div>3</div><div>사과</div><div>고씨네농장 고당도 청송사과, 햇부사 5kg</div><div>13000</div></div><div><div>4</div><div>바나나</div><div>델몬트 필리핀 바나나 6kg</div><div>15000</div></div><div><div>5</div><div>딸기</div><div>설향 딸기, 킹스베리 1박스</div><div>14500</div></div><div><div>6</div><div>오렌지</div><div>캘리포니아 네이블 오렌지 4kg</div><div>17000</div></div></div>

		return 위 코드 추가	<pre> let [input, setInput] = useState(""); </pre>
		전체코드 참고 녹색 - 주석 빨간색 - 코드 추가	<pre> import './App.css'; import { Button, Navbar, Container, Nav } from 'react-bootstrap'; import data from './db/fruit'; import data2 from './db/veggie'; import { useState } from 'react'; import Products from './components/Products'; import { Routes, Route, Link, useNavigate, Outlet } from 'react-router-dom'; import Detail from './components/Detail'; import Title from './components/Title'; import Title2 from './components/Title2'; import ComVeggie from './components/ComVeggie'; import Footer from './components/Footer'; import axios from 'axios'; import Cart from './components/Cart'; function App() { let [fruit, setFruit] = useState(data); let [veggie, setVeggie] = useState(data2); let [count, setCount] = useState(1); let [input, setInput] = useState(""); // console.log(fruit[0].price); let navigate = useNavigate(); const sortByName = () => { let sortedFruit = [...fruit].sort((a, b) => (a.title > b.title ? 1 : -1)); setFruit(sortedFruit); console.log(sortedFruit); }; const sortByPriceLowToHigh = () => { let sortedFruit = [...fruit].sort((a, b) => a.price - b.price); </pre>

			<pre> setFruit(sortedFruit); console.log(sortedFruit); }; const sortByPriceHighToLow = () => { let sortedFruit = [...fruit].sort((a, b) => b.price - a.price); setFruit(sortedFruit); console.log(sortedFruit); }; return (<div className="App"> <> <Navbar bg="dark" variant="dark"> <Container> <Navbar.Brand onClick={()=>{ navigate('/')}}>과일농장</Navbar.Brand> <Nav className="me-auto"> <Nav.Link onClick={()=>{ navigate('/')}}>홈으로</Nav.Link> <Nav.Link onClick={()=>{ navigate('/detail/1')}}>상세페이지</Nav.Link> <Nav.Link onClick={() => { navigate('/cart') }}>장바구니</Nav.Link> <Nav.Link onClick={() => { navigate('/about') }}>회사소개</Nav.Link> </Nav> </Container> </Navbar> {/* <Link to="/">홈</Link>
 </br> <Link to="/detail">상세페이지</Link> */} <Routes> <Route path="/" element={ <div> <div className="slider"></div> <Title /> {/* <div class="container"> <div class="row"> <div style={{ textAlign: "center" }}> <Button variant="outline-primary" onClick={sortByName}>이름순 정렬</Button>{" "} <Button variant="outline-secondary" onClick={sortByPriceLowToHigh}>낮은가격순 정렬</Button>{" "} */} </div> } </Route> </Routes> </> </div>); </pre>
--	--	--	---

			<pre> <Button variant="outline-success" onClick={sortByPriceHighToLow}>높은가격순 정렬</Button>{" "} </div> </div> </div> */ <div className="container"> <div className="row"> <div className="col-md-6" style={{textAlign:"left"}}> /* 검색 추가 */ <input placeholder="상품명을 입력하세요" // 검색어가 변경될 때마다 input 값을 업데이트 onChange={(e) => setInput(e.target.value)} value={input} style={{ padding: "10px", marginLeft: "10px", borderRadius: "4px", border: "1px solid #ccc", width:"250px", marginRight:"10px" }} /> </div> <div className="col-md-6" style={{textAlign:"right"}}> /* select추가 */ <select onChange={(e) => { if (e.target.value === "low") sortByPriceLowToHigh(); if (e.target.value === "high") sortByPriceHighToLow(); if (e.target.value === "name") sortByName(); }} style={{ padding: "10px", marginLeft: "10px", borderRadius: "4px", border: "1px solid #ccc", width:"150px" }} > <option value="low">낮은가격순 정렬</option> <option value="high">높은가격순 정렬</option> <option value="name">상품명순 정렬</option> </select> </div> </div> </div> </pre>
--	--	--	---

			<pre> }}> <option value="">정렬 선택</option> <option value="low">낮은 가격순</option> <option value="high">높은 가격순</option> <option value="name">이름순</option> </select> </div> </div> </div> {/* <div className="container" style={{marginTop:'30px'}}> <div className="row"> { fruit.map((ele,i)=>{ return(<Products fruit={fruit[i]} i={i} key={fruit[i].id} />) }) } </div> </div> */} <div className="container" style={{ marginTop: "30px" }}> <div className="row"> {/* 'input'에 맞는 제목을 가진 항목들만 필터링 */} {/* .filter(...) 조건에 맞는 데이터만 걸러냄 .map(...) 걸러낸 데이터를 기반으로 컴포넌트를 렌더링 */} {fruit .filter((item) => item.title.toLowerCase().includes(input.toLowerCase()) // 제목 검색) .map(fruit) => (<Products {...fruit} key={fruit.id} /> // 필터링된 항목 출력)} </div> </div> </pre>
--	--	--	---

			<pre> <div class="container"> <div class="row"> <div style={{ textAlign: "center" }}> <Title2 /> <Button variant="outline-success" count={count} onClick={() => { if (count == 1) { axios .get("https://dino-21.github.io/react_data/veggie2.json") .then((result) => { let copy10 = [...veggie, ...result.data]; setVeggie(copy10); setCount(count + 1); }); } else if (count == 2) { axios .get("https://dino-21.github.io/react_data/veggie3.json") .then((result) => { let copy11 = [...veggie, ...result.data]; setVeggie(copy11); setCount(count + 1); }); } if (count === 3) { alert("더이상 상품이 없습니다."); } }} > {" "} + 3개 상품 더 보기{" "} </Button>{" "} </div> </pre>
--	--	--	---

			<pre> </div> </div> <div className="container" style={{ marginTop: "30px" }}> <div className="row"> {veggie.map((ele, i) => { return (<ComVeggie veggie={veggie[i]} key={veggie[i].id} />); })} </div> </div> <Footer /> </div> } /> <Route path="/detail/:paramId" element={<Detail fruit={fruit} />} /> <Route path="/cart" element={<Cart />} /> <Route path="/about" element={<About />}> <Route path="member" element={<Member />} /> <Route path="location" element={<Location />} /> </Route> </Routes> </> </div>); } //app 아래, export default App; 위에 function About() { return (<> <h4>회사정보</h4> <Outlet></Outlet> </>); } function Member() { return (</pre>
--	--	--	---

			<pre><> <h4>Member</h4> </>); } function Location() { return (<> <h4>Location</h4> </>); } export default App;</pre>
--	--	--	---

브라우저에서 확인

왼쪽 input에 수박 상품검색 및 다른 상품도 검색

localhost:3000

햇과일 BEST

농부가 추천하는 제철과일을 만나보세요.

수박

정렬 선택

최대 단맛

수박

당도선별 프리미엄 고당도 하우스수박 5~6kg

29000

localhost:3000

햇과일 BEST

농부가 추천하는 제철과일을 만나보세요.

망고

정렬 선택

최대 단맛

망고

프리미엄 제주애플망고 1박스

18500

오른쪽 select 박스에서 낮은가격순 및 높은 가격순, 이름순 정렬확인

localhost:3000

햇과일 BEST

농부가 추천하는 제철과일을 만나보세요.

수박

정렬 선택

수박

당도선별 프리미엄 고당도 하우스수박 5~6kg

29000

사과

고씨네농장 고당도 청송사과, 햇부사 5kg

13000

딸기

실향 딸기, 킹스베리 1박스

14500

바나나

델몬트 필리핀 바나나 6kg

15000

참외

산지직송 청주 달콤참외 3kg

16900

오렌지

캘리포니아 네이블 오렌지 4kg

17000

망고

프리미엄 제주애플망고 1박스

18500

17 게시판 만들기 - CRUD 기능구현

순서	작업환경	명령어
1	Components 폴더아래 Board.js 만들기	<pre> import React, { useState } from 'react'; import { Button, Table, Form, Alert } from 'react-bootstrap'; const FruitBoard = () => { // 게시물 목록 상태 관리 const [boardList, setBoardList] = useState([{ no: "1", title: '사과는 언제 배송이 되나요?', description: '어제부터 기다렸는데 아직 배송이 안됐어요.', viewCount: 1 }, { no: "2", title: '수박크기가 작아요', description: '수박이 맛있고 달았습니다. 하지만 수박크기는 많이 작았어요. ', viewCount: 2 }, { no: "3", title: '오렌지 당도가 낮아요', description: '당도가 11birx이상은 아닌것같아요.', viewCount: 1 }, { no: "4", title: '딸기향이 이상해요', description: '딸기에서 흙냄새가 납니다.', viewCount: 1 }]); // UI 상태들 const [listOk, setListOk] = useState(true); // 게시물 전체리스트 const [readOk, setReadOk] = useState(false); // 게시물 읽기 const [writeOk, setWriteOk] = useState(false); // 게시물 쓰기 const [editOk, setEditOk] = useState(false); // 게시물 수정 const [boardInfo, setBoardInfo] = useState({}); // 현재 읽고 있는 게시글의 정보 (제목, 내용 등)를 저장하는 상태 // 작성 폼 상태들 const [title, setTitle] = useState(""); const [description, setDescription] = useState(""); // 수정 상태들 추가 const [editNo, setEditNo] = useState(null); // 수정할 게시물 번호 const [editTitle, setEditTitle] = useState(""); const [editDescription, setEditDescription] = useState(""); // 오류 메시지 상태 const [errorMessage, setErrorMessage] = useState(""); </pre>

```

// 게시글 목록 보기
const boardListView = () => {
  setReadOk(false);
  setWriteOk(false);
  setEditOk(false);
  setListOk(true);
};

// 게시글 읽기
const boardRead = (no) => {
  setListOk(false);
  setWriteOk(false);
  setEditOk(false);
  setReadOk(true);

  // 조회수 증가, 클릭한 게시물이면 1증감
  const updatedList = boardList.map(b =>
    b.no === no ? { ...b, viewCount: b.viewCount + 1 } : b
  );
  setBoardList(updatedList); //조회수 1 증가해서 다시 렌더링

  const selectedBoard = boardList.find(b => b.no === no); // 클릭한 게시물 번호를 찾아
  setBoardInfo(selectedBoard); // 현재 읽고 있는 게시글의 정보를 저장해서 화면에 렌더링
};

// 게시글 작성 폼 열기
const boardWrite = () => {
  setListOk(false);
  setWriteOk(true);
};

// 새 글 저장
const boardSave = () => {
  // 제목과 설명이 비어있으면 유효성 검사
  if (title.trim() === '' || description.trim() === '') {
    setErrorMessage('제목과 내용을 모두 입력해주세요!');
    return;
  }
}

```

```

// 새 글 추가
const newBoard= {
  no: (boardList.length + 1).toString(),
  title: title,
  description: description,
  viewCount: 0 // 초기 조회수는 0
};
setBoardList([...boardList, newBoard]);
setTitle("");
setDescription("");
setErrorMessage(""); // 오류 메시지 초기화
boardListView(); // 새글 저장후 게시물 목록 함수 호출
};

// 게시물 삭제
const boardDelete = (no) => {
  // 삭제할 게시글을 `no` 값으로 필터링하여 삭제
  const updatedList = boardList.filter(b => b.no !== no.toString()); // no.toString()으로 비교
  setBoardList(updatedList);
  boardListView(); // 삭제 후 목록 보기로 이동
};

// 게시물 수정 폼 열기
const boardEdit = (no) => {
  setEditOk(true);
  setListOk(false);

  const boardToEdit = boardList.find(b => b.no === no); // 클릭한 번호를 찾아 객체의 데이터(게시글제목, 내용) 가져와서
  setEditNo(boardToEdit.no); // 데이터의 번호
  setEditTitle(boardToEdit.title); // 데이터의 제목
  setEditDescription(boardToEdit.description); // 데이터의 내용을 폼태그에 채워넣는다.
};

```

```

// 수정된 게시글 저장
const updateBoard = () => {
  const updatedBoardList = boardList.map(b => // 게시물에서 수정된 번호를 찾아 게시글제목과 설명을 저장
    b.no === editNo ? { ...b, title: editTitle, description: editDescription } : b
  );
  setBoardList(updatedBoardList); // 게시물 목록을 업데이트
  boardListView(); // 수정 후 목록 보기로 이동
};

return (
  <div className="container" style={{ marginTop: "80px" }}>

    <h3>과일농장 게시판</h3>

    /* 과일 목록 보기 */
    {listOk && (
      <div style={{ marginTop: "30px" }}>
        <Table striped bordered hover>
          <thead>
            <tr>
              <th>번호</th>
              <th>제목</th>
              <th>문의글</th>
              <th>조회수</th>
              <th>문의하기</th>
            </tr>
          </thead>
          <tbody>
            {boardList.slice().reverse().map(board => (
              <tr key={board.no}>
                <td>{board.no}</td>
                <td style={{ cursor: 'pointer', textAlign: 'left' }} onClick={() => boardRead(board.no)>
                  {board.title}
                </td>
                <td style={{ cursor: 'pointer', textAlign: 'left' }} onClick={() => boardRead(board.no)>
                  {board.description}
                </td>
                <td>{board.viewCount}</td> /* 조회수 표시 */
                <td>

```

```

        <Button variant="outline-primary" onClick={() => boardRead(board.no)}>
            게시글 읽기
        </Button>
        <Button variant="outline-success" onClick={() => boardEdit(board.no)} style={{ marginLeft: "10px" }}>수정</Button>
        <Button variant="outline-danger" onClick={() => boardDelete(board.no)} style={{ marginLeft: "10px" }}>
            삭제
        </Button>
    </td>
</tr>
)))
</tbody>
</Table>
<Button variant="primary" onClick={boardWrite} style={{ float: 'right' }}>
    문의글 작성하기
</Button>
</div>
)}

{/* 게시글 읽기 */}
{readOk && (
    <div>
        <h5 style={{textAlign: "left" }}>{boardInfo.title}</h5>
        <hr></hr>
        <p style={{textAlign: "left" }}>{boardInfo.description}</p>
        <br></br>
        <div style={{ textAlign: "right"}}>
            <Button variant="secondary" onClick={boardListView} style={{textAlign: "right" }}>
                목록으로
            </Button>
        </div>
    </div>
)}

```

```

    {/* 새 글 작성 폼 */}
    {writeOk && (
      <div style={{ marginTop: "30px" }}>
        <h5 style={{textAlign: "left" }}>과일농장에게 문의글 남기기</h5>

        {/* 오류 메시지 표시 */}
        {errorMessage && (
          <Alert variant="danger">{errorMessage}</Alert>
        )}

        <Form.Group controlId="formName">
          <Form.Control
            type="text"
            value={title}
            onChange={(e) => setTitle(e.target.value)}
            placeholder="게시글을 입력하세요"
          />
        </Form.Group>
        <Form.Group controlId="formDescription" style={{ marginTop: "30px" }}>
          <Form.Control
            as="textarea"
            rows={3}
            value={description}
            onChange={(e) => setDescription(e.target.value)}
            placeholder="게시물에 작성하세요"
          />
        </Form.Group>
        <br> </br>
        <div style={{ textAlign: "right"}}>
          <Button variant="primary" onClick={boardSave} style={{ marginRight: "10px" }}>저장</Button>
          <Button variant="secondary" onClick={boardListView}>목록으로</Button>
        </div>
      </div>
    )}

```

```

    /* 과일 수정 폼 */
    {editOk && (
      <div style={{ marginTop: "30px" }}>
        <h5 style={{textAlign: "left" }}>게시물 수정</h5>
        <Form.Group controlId="formEditName">
          <Form.Control
            type="text"
            value={editTitle}
            onChange={(e) => setEditTitle(e.target.value)}
            placeholder="수정된 제목"
          />
        </Form.Group>
        <br> </br>
        <Form.Group controlId="formEditDescription">
          <Form.Control
            as="textarea"
            rows={3}
            value={editDescription}
            onChange={(e) => setEditDescription(e.target.value)}
            placeholder="수정된 설명"
          />
        </Form.Group>
        <br> </br>
        <div style={{ textAlign: 'right' }}>
          <Button variant="outline-success" onClick={updateBoard} style={{ marginRight: "10px" }}>수정</Button>
          <Button variant="outline-info" onClick={boardListView}>목록으로</Button>
        </div>
      </div>
    )}

  </div>
);
};

export default FruitBoard;

```


순서	작업환경	명령어	
2	App.js 코드 수정	상단	import Board from "../components/Board";
		return아래 네이게이션에서 Board 추가	<pre> <Navbar bg="dark" variant="dark"> <Container> <Navbar.Brand onClick={() => { navigate("/")}}> 과일농장 </Navbar.Brand> <Nav className="me-auto"> <Nav.Link onClick={() => { navigate("/")}}> 홈으로 </Nav.Link> <Nav.Link onClick={() => { navigate("/detail/1"); }} > 상세페이지 </Nav.Link> <Nav.Link onClick={() => { navigate("/cart"); }}> 장바구니</Nav.Link> <Nav.Link onClick={() => { navigate("/about");}}> 회사소개</Nav.Link> <Nav.Link onClick={() => { navigate("/Board");}}> 게시판</Nav.Link> </Nav> </Container> </Navbar> </pre>
		라우터에서 컴포넌트 Board추가	<pre> <Route path="/detail/:paramId" element={<Detail fruit={fruit} />} /> <Route path="/cart" element={<Cart />} /> <Route path="/about" element={<About />}> <Route path="member" element={<Member />} /> <Route path="location" element={<Location />} /> </Route> <Route path="/board" element={<Board/>} /> </Routes> </pre>

브라우저에서 확인

게시판 목록

localhost:3000/Board

과일농장 홈으로 상세페이지 장바구니 회사소개 게시판

과일농장 게시판

번호	제목	문의글	조회수	문의하기
4	딸기향이 이상해요	딸기에서 써냄새가 납니다.	1	게시글읽기 수정 삭제
3	오렌지 당도가 낮아요	당도가 11brix이상은 아닌것같아요.	1	게시글읽기 수정 삭제
2	수박크기가 작아요	수박이 맛있고 달았습니다. 하지만 수박크기는 많이 작았어요.	2	게시글읽기 수정 삭제
1	사과는 언제 배송이 되나요?	어제부터 기다렸는데 아직 배송이 안왔어요.	1	게시글읽기 수정 삭제

[문의글 작성하기](#)

글작성 - 유효성 체크

localhost:3000/Board

과일농장 홈으로 상세페이지 장바구니 회사소개 게시판

과일농장 게시판

과일농장에게 문의글 남기기

제목과 내용을 모두 입력해주세요!

게시글을 입력하세요

글을 작성하지 않으면
경고메시지
(유효성체크)

[저장](#) [목록으로](#)

글등록

localhost:3000/Board

과일농장 홈으로 상세페이지 장바구니 회사소개 게시판

과일농장 게시판

과일농장에게 문의글 남기기

제목과 내용을 모두 입력해주세요!

수박이 안왔어요.

수박은 언제 배송되나요?
답변주세요.

[저장](#) [목록으로](#)

localhost:3000/Board

과일농장 홈으로 상세페이지 장바구니 회사소개 게시판

게시물이 추가됨

과일농장 게시판

번호	제목	문의글	조회수	문의하기
5	수박이 안왔어요.	수박은 언제 배송되나요? 답변주세요.	0	게시글읽기 수정 삭제
4	딸기향이 이상해요	딸기에서 써냄새가 납니다.	1	게시글읽기 수정 삭제
3	오렌지 당도가 낮아요	당도가 11brix이상은 아닌것같아요.	1	게시글읽기 수정 삭제
2	수박크기가 작아요	수박이 맛있고 달았습니다. 하지만 수박크기는 많이 작았어요.	2	게시글읽기 수정 삭제
1	사과는 언제 배송이 되나요?	어제부터 기다렸는데 아직 배송이 안왔어요.	1	게시글읽기 수정 삭제

[문의글 작성하기](#)

수정폼

localhost:3000/Board

과일농장 홈으로 상세페이지 장바구니 회사소개 게시판

과일농장 게시판

게시물 수정

수박이 안왔어요. 22222

수박은 언제 배송되나요?22222
답변주세요.

수정 목록으로

수정되어서 게시물 목록으로 옴

localhost:3000/Board

과일농장 홈으로 상세페이지 장바구니 회사소개 게시판

글 수정됨 과일농장 게시판 조회수 올라감

번호	제목	문의글	조회수	문의하기
5	수박이 안왔어요. 22222	수박은 언제 배송되나요?22222 답변주세요. 수박이 도착했어요.	1	게시글읽기 수정 삭제
4	딸기향이 이상해요	딸기에서 흠냄새가 납니다.	1	게시글읽기 수정 삭제
3	오렌지 당도가 낮아요	당도가 11brix이상은 아닌것같아요.	1	게시글읽기 수정 삭제
2	수박크기가 작아요	수박이 맛있고 달았습니다. 하지만 수박크기는 많이 작았어요.	2	게시글읽기 수정 삭제
1	사과는 언제 배송이 되나요?	어제부터 기다렸는데 아직 배송이 안됐어요.	1	게시글읽기 수정 삭제

문의글 작성하기

삭제버튼 클릭

localhost:3000/Board

과일농장 홈으로 상세페이지 장바구니 회사소개 게시판

과일농장 게시판

번호	제목	문의글	조회수	문의하기
5	수박이 안왔어요. 22222	수박은 언제 배송되나요?22222 답변주세요. 수박이 도착했어요.	1	게시글읽기 수정 삭제
4	딸기향이 이상해요	딸기에서 흠냄새가 납니다.	1	게시글읽기 수정 삭제
3	오렌지 당도가 낮아요	당도가 11brix이상은 아닌것같아요.	1	게시글읽기 수정 삭제
2	수박크기가 작아요	수박이 맛있고 달았습니다. 하지만 수박크기는 많이 작았어요.	2	게시글읽기 수정 삭제
1	사과는 언제 배송이 되나요?	어제부터 기다렸는데 아직 배송이 안했어요.	1	게시글읽기 수정 삭제

문의글 작성하기

삭제되어서 글목록에 반영됨

localhost:3000/Board

과일농장 홈으로 상세페이지 장바구니 회사소개 게시판

과일농장 게시판

5번 게시물 삭제됨

번호	제목	문의글	조회수	문의하기
4	딸기향이 이상해요	딸기에서 흠냄새가 납니다.	1	게시글읽기 수정 삭제
3	오렌지 당도가 낮아요	당도가 11brix이상은 아닌것같아요.	1	게시글읽기 수정 삭제
2	수박크기가 작아요	수박이 맛있고 달았습니다. 하지만 수박크기는 많이 작았어요.	2	게시글읽기 수정 삭제
1	사과는 언제 배송이 되나요?	어제부터 기다렸는데 아직 배송이 안했어요.	1	게시글읽기 수정 삭제

문의글 작성하기

코드분석

localhost:3000/Board

과일농장 홈으로 상세페이지 장바구니 회사소개 게시판

listOk - 게시글 전체 리스트 t

과일농장 게시판

boardDelete(b.no)

boardRead(b.no)

updatedList

번호	제목	문의글	조회수	문의하기
4	딸기향이 이상해요	딸기에서 흠냄새가 납니다.	1	게시글읽기 수정 삭제
3	오렌지 당도가 낮아요	당도가 11birx이상은 아닌것같아요.	1	게시글읽기 수정 삭제
2	수박크기가 작아요	수박이 맛있고 달았습니다. 하지만 수박크기는 많이 작았어요.	2	게시글읽기 수정 삭제
1	사과는 언제 배송이 되나요?	어제부터 기다렸는데 아직 배송이 안됐어요.	1	게시글읽기 수정 삭제

boardList(데이터)

boardListView

readOk - 읽기 f

과일농장 게시판

오렌지 당도가 낮아요
당도가 11birx이상은 아닌것같아요.

boardListView 목록으로

editOk - 수정 f

과일농장 게시판

게시물 수정

딸기향이 이상해요3333

딸기에서 흠냄새가 납니다.333333

boardInfo

- 현재 읽고 있는 게시글의 정보(제목, 내용)

updateBoard 수정 목록으로 boardListView

과일농장에게 문의글 남기기

게시글을 입력하세요

게시물에 작성하세요

boardSave 저장 목록으로

boardListView

```
import React, { useState } from 'react';
import { Button, Table, Form, Alert } from 'react-bootstrap';
```

```
const FruitBoard = () => {
```

```
// 게시물 목록 상태 관리
```

```
const [boardList, setBoardList] = useState([
  { no: "1", title: '사과는 언제 배송이 되나요?', description: '어제부터 기다렸는데 아직 배',
  { no: "2", title: '수박크기가 작아요', description: '수박이 맛있고 달았습니다. 하지만 수박',
  { no: "3", title: '오렌지 당도가 낮아요', description: '당도가 11birx이상은 아닌것같아요.',
  { no: "4", title: '딸기향이 이상해요', description: '딸기에서 흠냄새가 납니다.', viewCount
]);
```

```
// UI 상태들
```

```
const [listOk, setListOk] = useState(true); // 게시글 전체리스트
const [readOk, setReadOk] = useState(false); // 게시글 읽기
const [writeOk, setWriteOk] = useState(false); // 게시글 쓰기
const [editOk, setEditOk] = useState(false); // 게시글 수정
const [boardInfo, setBoardInfo] = useState({}); // 현재 읽고 있는 게시글의 정보 (제목, 내용)
```

```
// 작성 폼 상태들
```

```
const [title, setTitle] = useState("");
const [description, setDescription] = useState("");
```

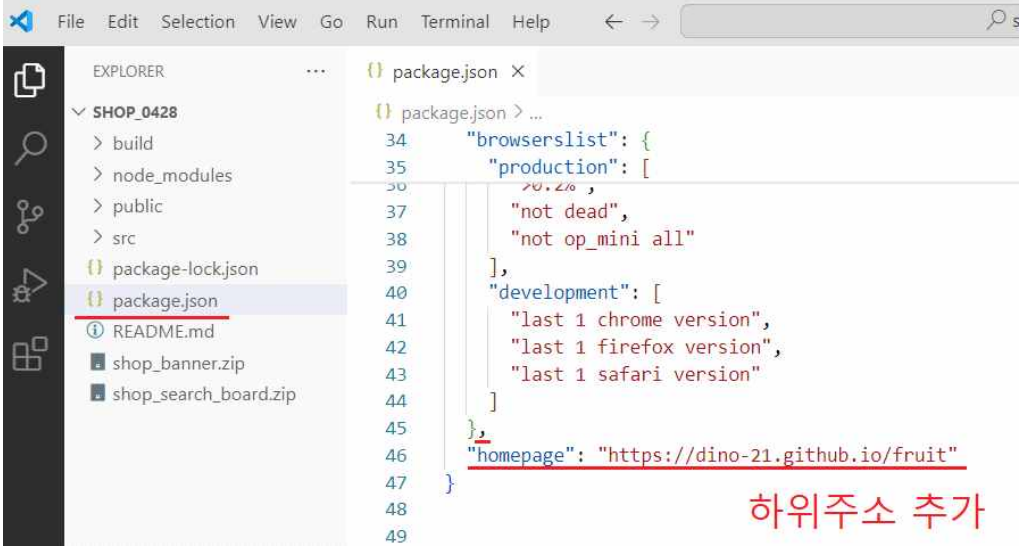
```
// 수정 상태들 추가
```

```
const [editNo, setEditNo] = useState(null); // 수정할 게시물 번호
const [editTitle, setEditTitle] = useState("");
const [editDescription, setEditDescription] = useState("");
```

```
// 오류 메시지 상태
```

```
const [errorMessage, setErrorMessage] = useState("");
```

18 하위주소 배포하기 위해 이미지 경로 수정

하위경로 주소		https://sinaboro.github.io/fruit/
1	package.json 마지막 라인에 추가	 <pre>34 "browserslist": { 35 "production": [36 ">0.2%", 37 "not dead", 38 "not op_mini all" 39], 40 "development": [41 "last 1 chrome version", 42 "last 1 firefox version", 43 "last 1 safari version" 44] 45 }, 46 "homepage": "https://dino-21.github.io/fruit" 47 } 48 49</pre> <p>하위주소 추가</p>
1	package.json 전체 코드(참고)	<pre>{ "name": "shop1", "version": "0.1.0", "private": true, "dependencies": { "@reduxjs/toolkit": "^2.7.0", "@testing-library/dom": "^10.4.0", "@testing-library/jest-dom": "^6.6.3", "@testing-library/react": "^16.3.0", "@testing-library/user-event": "^13.5.0", "axios": "^1.8.4", "bootstrap": "^5.3.5", "react": "^19.1.0", "react-bootstrap": "^2.10.9", "react-dom": "^19.1.0", "react-redux": "^9.2.0",</pre>

```
"react-router-dom": "^6.30.0",
"react-scripts": "5.0.1",
"styled-components": "^6.1.17",
"web-vitals": "^2.1.4"
},
"scripts": {
  "start": "react-scripts start",
  "build": "react-scripts build",
  "test": "react-scripts test",
  "eject": "react-scripts eject"
},
"eslintConfig": {
  "extends": [
    "react-app",
    "react-app/jest"
  ]
},
"browserslist": {
  "production": [
    ">0.2%",
    "not dead",
    "not op_mini all"
  ],
  "development": [
    "last 1 chrome version",
    "last 1 firefox version",
    "last 1 safari version"
  ]
},
"homepage": "https://sinaboro.github.io/fruit"
}
```

2

index.js
basename 추가

```

1  import React from "react";
2  import ReactDOM from "react-dom/client";
3  import "./index.css";
4  import App from "./App";
5  import reportWebVitals from "./reportWebVitals";
6  import { BrowserRouter } from "react-router-dom";
7  import { Provider } from "react-redux";
8  import store from "./store";
9
10 const root = ReactDOM.createRoot(document.getElementById("root"));
11 root.render(
12   // <React.StrictMode>
13   <Provider store={store}>
14     <BrowserRouter basename="/fruit">
15       <App />
16     </BrowserRouter>
17   </Provider>
18   // </React.StrictMode>
19 );
20
21 // If you want to start measuring performance in your app, pass a function
22 // to log results (for example: reportWebVitals(console.log))
23 // or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
24 reportWebVitals();
25

```

```

<BrowserRouter basename="/fruit">
  <App />
</BrowserRouter>

```


2	index.js 전체코드 (참고)	<pre> import React from "react"; import ReactDOM from "react-dom/client"; import "./index.css"; import App from "./App"; import reportWebVitals from "./reportWebVitals"; import { BrowserRouter } from "react-router-dom"; import { Provider } from "react-redux"; import store from "./store"; const root = ReactDOM.createRoot(document.getElementById("root")); root.render(// <React.StrictMode> <Provider store={store}> <BrowserRouter basename="/fruit"> <App /> </BrowserRouter> </Provider> // </React.StrictMode>); // If you want to start measuring performance in your app, pass a function // to log results (for example: reportWebVitals(console.log)) // or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals reportWebVitals(); </pre>
---	-----------------------	--

3	상세페이지 Detail.js	수정 전	
		수정 후 1	
			fruit 하위 폴더가 하드코딩됨 프로젝트 이름이 바뀌거나 폴더 구조가 바뀌면 코드를 일일이 수정해야 함
		수정 후 2 (추천)	
			PUBLIC_URL은 package.json의 "homepage"에 자동으로 맞춰짐 예: "homepage": "https://dino-21.github.io/fruit" 설정 시 process.env.PUBLIC_URL 값은 "https://dino-21.github.io/fruit" 가 됨 배포 환경 경로가 자동으로 적용됨 유지보수에 유리: fruit, shop5 등 이름 바뀌어도 수정할 필요 없음
	process.env.PUBLIC_URL 값은 어디서 나올까?	// package.json "homepage": "https://sinaboro.github.io/fruit" 이 설정 덕분에 React는 fruit 하위에서 실행됨을 알고 PUBLIC_URL에 "https://sinaboro.github.io/fruit"를 자동 대입해준다.	

process.env.PUBLIC_URL 수정하기 - Detail.js

Detail.js
상세페이지

배너 이미지
경로 수정

EXPLORER

SHOP_0428

> build

> node_modules

> public

> img

★ favicon.ico

img.zip

<> index.html

logo192.png

logo512.png

{ } manifest.json

robots.txt

> src

> components

JS Board.js

JS Cart.js

JS ComVeggie.js

JS Detail.js

JS Footer.js

JS Products.js

JS Title.js

JS Title2.js

> db

App.css

...

JS Detail.js

×

src > components > JS Detail.js > ...

1 import React from "react";

2 import { useParams } from "react-router-dom";

3 import { Nav } from "react-bootstrap";

4 import { useState, useEffect } from "react"; // 한 줄로 합친

5 import { addItem } from "../store.js";

6 import { useDispatch } from "react-redux";

7 import { Button } from "react-bootstrap";

8 import { Link } from "react-router-dom";

9 import styled from 'styled-components'

10

11

12 let Banner = styled.div`

13 | padding : 20px;

14

15 `;

16

17 let BannerBtn = styled.button`

18 | color : white;

19 | font-size:30px;

20 | width:100%;

21 | padding : 100px 100px;

22 | border:1px solid #ccc;

23 | background-image: url(\${process.env.PUBLIC_URL + "/img/banner.jpg"});

24 | background-size:cover;

25 | background-position:center;

26 `;

27

28

29 function Detail(props) {

30 | let { productId } = useParams();

상세페이지-배너

```
let BannerBtn = styled.button`
  color : white;
  font-size:30px;
  width:100%;
  padding : 100px 100px;
  border:1px solid #ccc;
  background-image: url(${process.env.PUBLIC_URL + "/img/banner.jpg"});
  background-size:cover;
  background-position:center;
`;
```

Detail.js
상세페이지

상품 이미지
경로 수정

EXPLORER

SHOP_0428

> build

> node_modules

> public

> img

★ favicon.ico

img.zip

<> index.html

logo192.png

logo512.png

{ } manifest.json

robots.txt

> src

> components

JS Board.js

JS Cart.js

JS ComVeggie.js

JS Detail.js

JS Footer.js

JS Products.js

JS Title.js

JS Title2.js

> db

App.css

...

JS Detail.js

×

src > components > JS Detail.js > ...

29 function Detail(props) {

30

31

32 return (

33 | <div className={"container start " + fade2}>

34 |

35 | <Banner>

36 | <BannerBtn>과일농장의 맛과 건강을 선물하세요.</BannerBtn>

37 | </Banner>

38

39 <div className="row">

40 | <div className="col-md-6">

41 |

42 | </div>

43 | <div className="col-md-6">

44 | <h5 className="pt-5">{title}</h5>

45 | <p>{content}</p>

46 | <p>{price}</p>

47 | <Button

48 | | variant="primary"

49 | | onClick={() => {

50 | | // dispatch(addItem({id: 1, imgUrl: 'fruit1.jpg', name: 'Grey Yordan', c

51 | |

52 | | dispatch({

53 | | | addItem({

54 | | | | id: id,

55 | | | | imgUrl: imgUrl.replace("img/", ""),

56 | | | | name: title,

57 | | | | count: 1,

58 | | | })

59 | | })

60 | }

61 |)

상세페이지 - 상품이미지(과일)

리덕스부분은
파일명만 저장 (수정할필요없)

```
<div className="col-md-6">
  <img src={process.env.PUBLIC_URL + "/" + imgUrl} width="100%" alt={title} />
</div>
```

- 26 -

process.env.PUBLIC_URL 수정하기 - Cart.js

```
<Link to={`/${detail}/${id}`}>
    <img src={process.env.PUBLIC_URL + `/img/${imgurl}`} style={smallProductStyle} />
</Link>
```

Cart.js
장바구니
상품이미지
경로 수정

장바구니
- 상품이미지

장바구니
- 상품이미지

process.env.PUBLIC_URL 수정하기 - Products.js

Products.js

상품 컴포넌트
이미지
경로 수정

EXPLORER

... JS Products.js X

SHOP_0428

- build
- node_modules
- public
 - img
 - favicon.ico
 - img.zip
 - index.html
 - logo192.png
 - logo512.png
 - manifest.json
 - robots.txt
- src
 - components
 - Board.js
 - Cart.js
 - ComVeggie.js
 - Detail.js
 - Footer.js
 - Products.js
 - Title.js
 - Title2.js
 - db

src > components > JS Products.js > ...

```
1 import React from "react";
2 import { useNavigate } from "react-router-dom";
3 import { Nav } from "react-bootstrap";
4
5 const Products = (props) => {
6   const { id, title, price, imgUrl, content } = props.fruit;
7   let navigate = useNavigate();
8   return (
9
10     <div className="col-md-4" style={{ marginBottom: "50px" }}>
11       <Nav.Link onClick={() => { navigate("/detail/" + id) }} className="c1">
12         <img src={process.env.PUBLIC_URL + "/" + imgUrl} width="80%" />
13         <h5 style={{ marginTop: "10px" }}>{title}</h5>
14         <p>{content}</p>
15         <span>{price}</span>
16       </Nav.Link>
17     </div>
18   );
19 };
20
21 export default Products;
22
```

상품 컴포넌트에서
이미지 경로수정

process.env.PUBLIC_URL 수정하기 - ComVeggie.js

ComVeggie.js

채소 컴포넌트
이미지
경로 수정

```
EXPLORER  ...  JS ComVeggie.js X
  ▾ SHOP_0428
    > build
    > node_modules
    > public
    ▾ src
      ▾ components
        JS Board.js
        JS Cart.js
        JS ComVeggie.js
        JS Detail.js
        JS Footer.js
        JS Products.js
        JS Title.js
        JS Title2.js
    > db
    # App.css

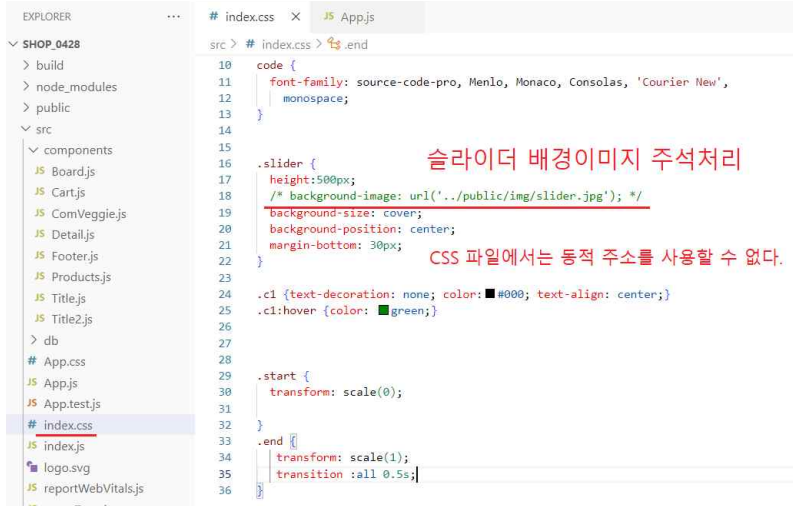
src > components > JS ComVeggie.js > ...
1  import React from "react";
2
3  const ComVeggie = (props) => {
4    const { imgUrl, title, content, price } = props.veggie;
5    console.log(props);
6    return (
7      <div className="col-md-4" style={{marginBottom:"50px"}}>
8        <img src={process.env.PUBLIC_URL + "/" + imgUrl} width="80%" />
9        <h5 style={{marginTop:"10px"}}>{title}</h5>
10       <span>{content}</span>
11       <p>{price}</p>
12     </div>
13   );
14 };
15
16 export default ComVeggie;
17
```

채소 컴포넌트
이미지 수정

process.env.PUBLIC_URL 수정하기 – index.css CSS에서는 동적 주소를 사용할 수 없다.

index.css

배경이미지
주석처리



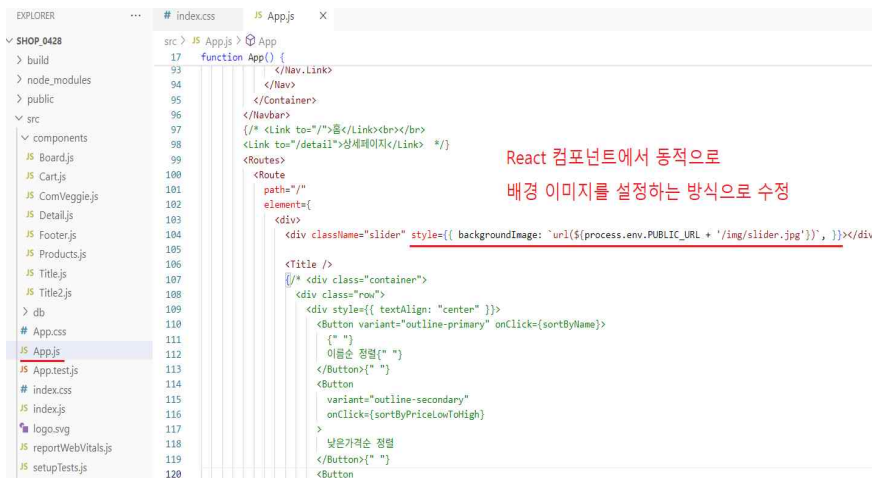
슬라이더 배경이미지 주석처리

CSS 파일에서는 동적 주소를 사용할 수 없다.

```
.slider {
  height:500px;
  /* background-image: url('../public/img/slider.jpg'); */
  background-size: cover;
  background-position: center;
  margin-bottom: 30px;
}
```

React 컴포넌트에서 동적으로 배경 이미지를 설정하는 방식으로 수정해야 함

App.js에서
인라인
style 주기

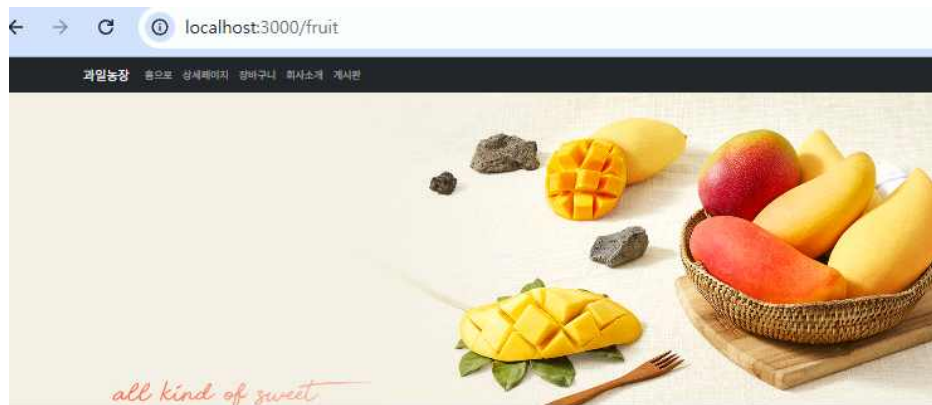


React 컴포넌트에서 동적으로

배경 이미지를 설정하는 방식으로 수정

```
<div className="slider" style={{ backgroundImage:
`url(${process.env.PUBLIC_URL + '/img/slider.jpg'})`, }}></div>
```


터미널에서 프론트서버 실행 - npm start



햇과일 BEST

농부가 추천하는 제철과일을 만나보세요.

상품명만 입력하세요

장면 선택



수박

당도산면 프리미엄 고당도 하우스수박 5~6kg

29000



참외

산지직송 상수 달콤참외 3kg

16900



사과

고씨네농장 고당도 향송사과, 햇부사 5kg

13000



바나나

옌본트 원리핀 바나나 6kg

15000



딸기

실향 딸기, 킹스베리 1박스

14500



오렌지

캘리포니아 네이플 오렌지 4kg

17000

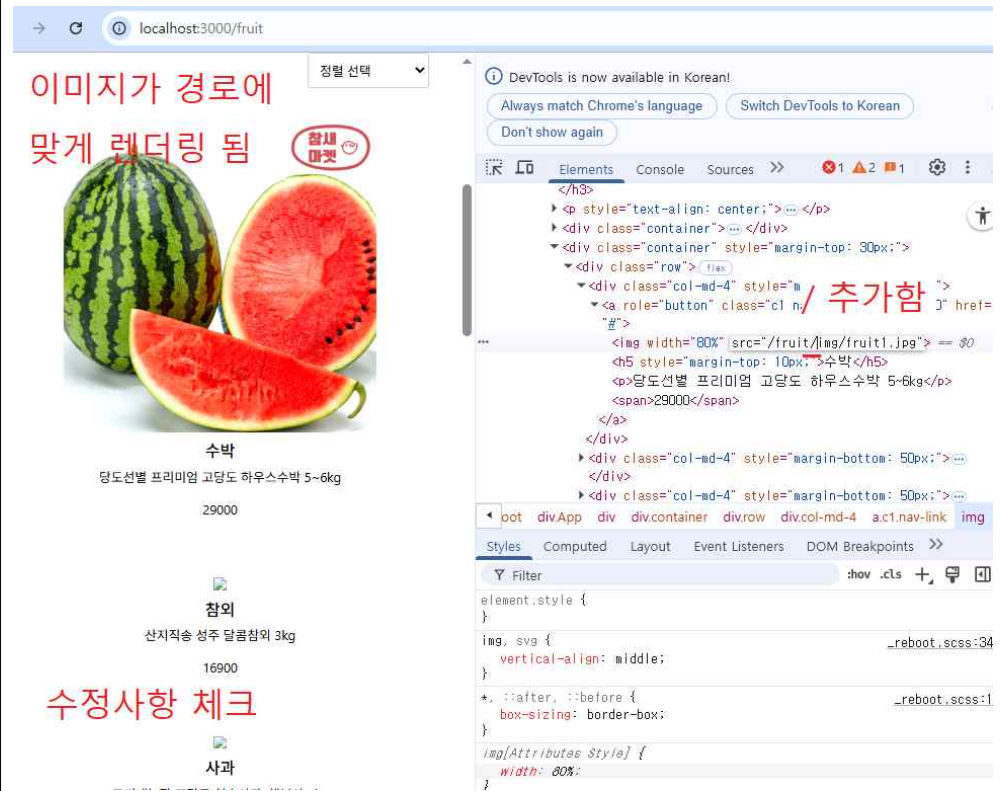
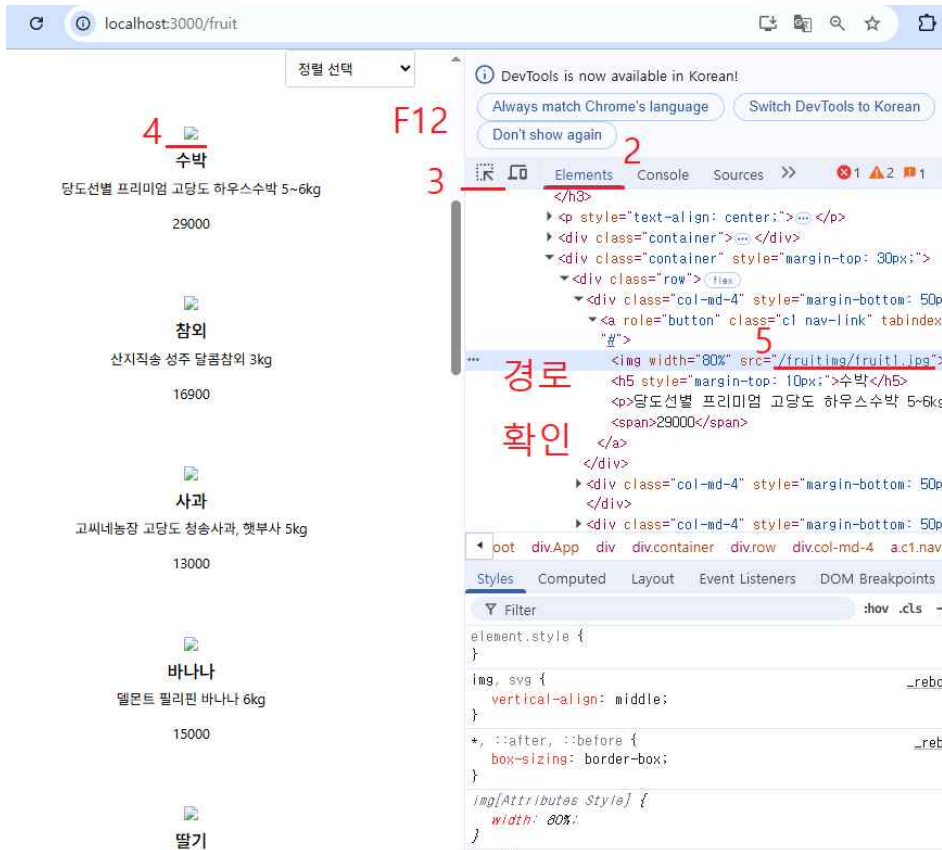
- 1 이미지가 안깨지는지 확인하기
- 2 상세페이지도 클릭하고
- 3 비동기통신으로 데이터도 잘 가져와서 이미지가 안깨지는지 확인
- 4 게시판도 잘되는지 체크
- 5 장바구니에 상품도 추가해서 잘되는지 확인

여기까지 따라왔다면 잘 될테지만, 만약 이미지가 깨진다면

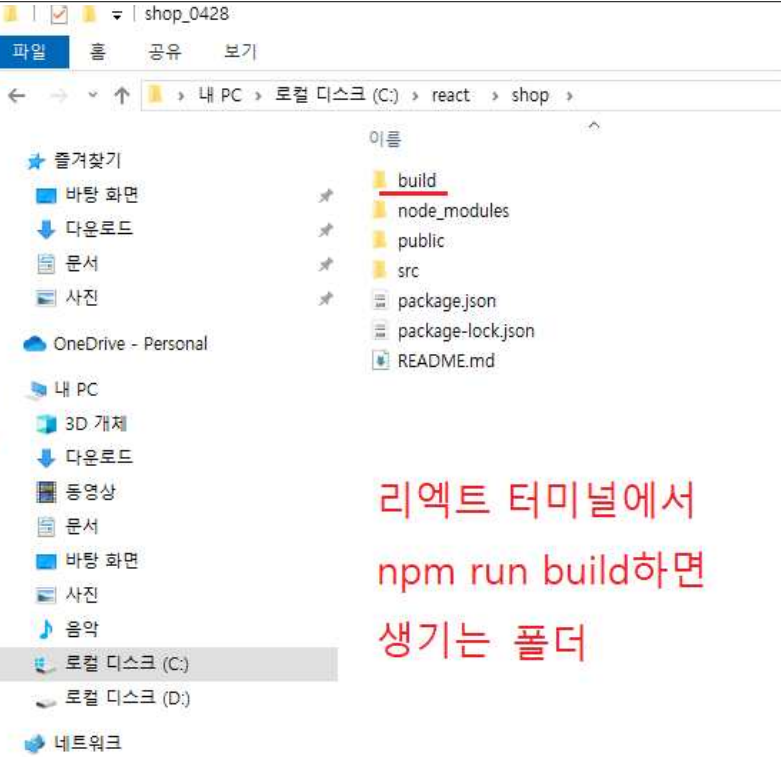
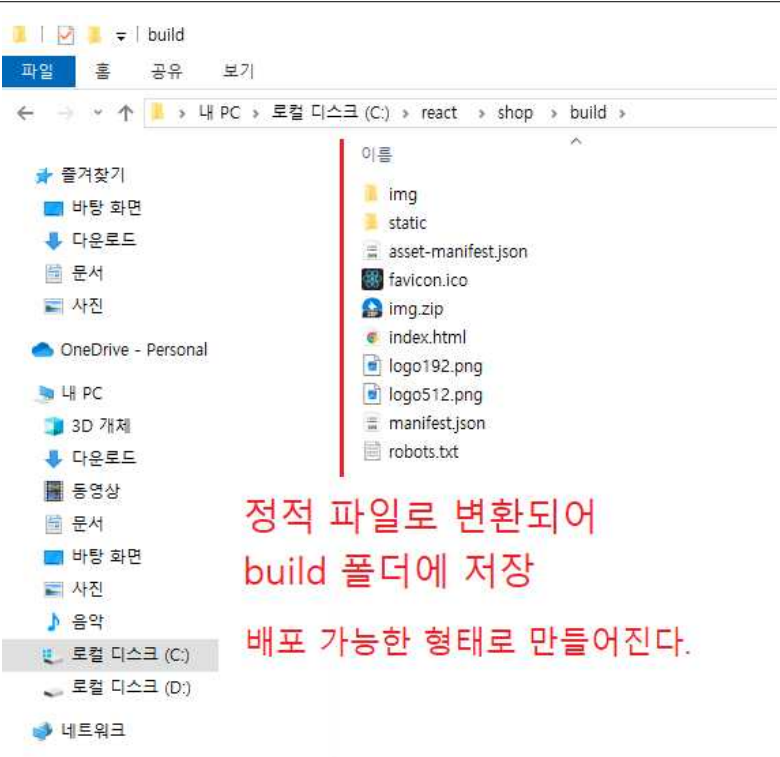
개발자도구(F12)로 가서 확인 ex) src="/fruitimg/fruit1.jpg"

ex) src="/fruitimg/fruit1.jpg"

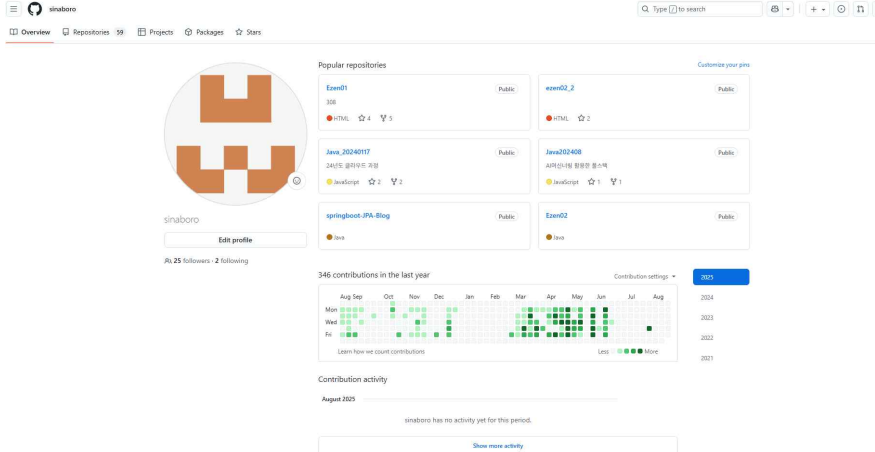
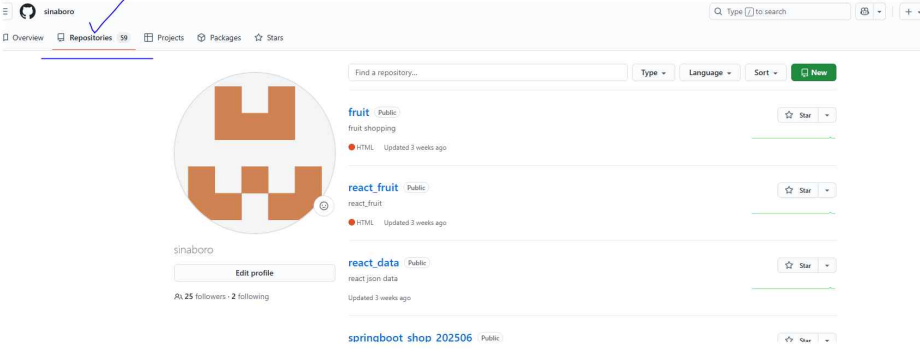
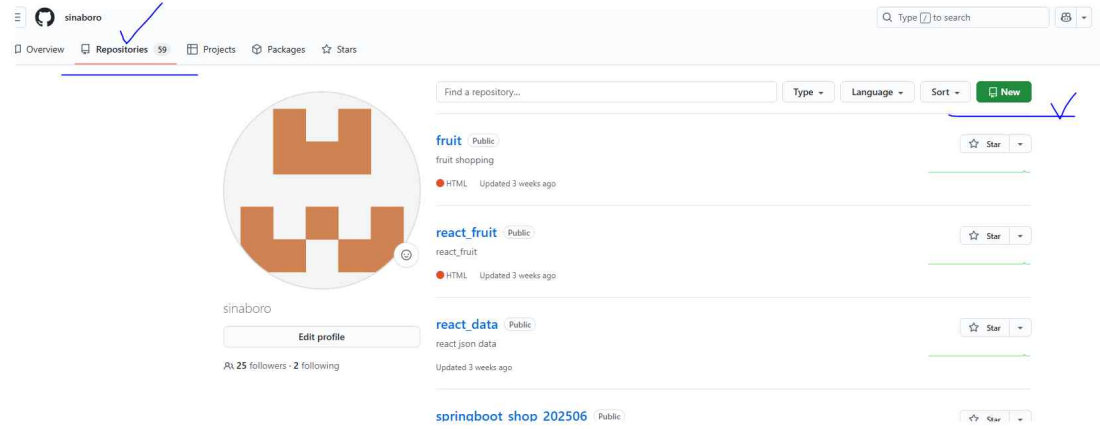
/를 추가해서 된다면 다시 리엑트 코드로 가서 수정해야 함.



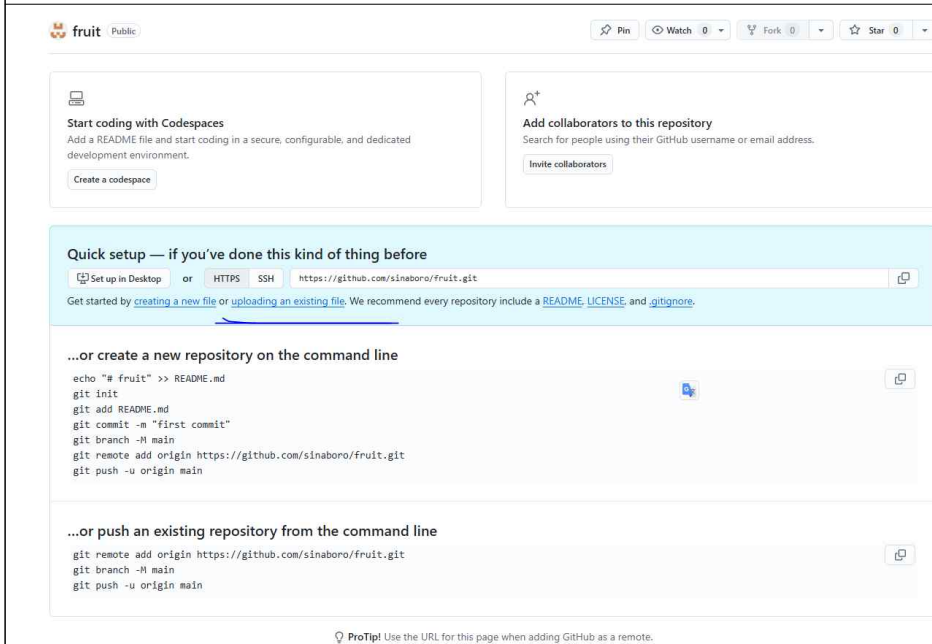
잘 동작되는지 확인 후 - 빌드 파일만들기

터미널에서	npm run build
탐색기로 작업폴더에 가면 build 폴더가 생김	build 파일에 들어가보면 정적 파일로 만들어줌
<div><p>리엑트 터미널에서 npm run build하면 생기는 폴더</p></div>	<div><p>정적 파일로 변환되어 build 폴더에 저장 배포 가능한 형태로 만들어진다.</p></div>
<p>npm run build 명령어를 실행하면, React 앱이 정적 파일로 변환되어 build/ 폴더에 저장된다. 이 파일들에는 HTML, CSS, JavaScript 등이 포함되어 최적화된 상태로 제공된다. 이 정적 파일들은 서버에서 직접 제공되며, 클라이언트가 요청할 때 동적으로 생성되지 않는다. 결국, npm run build 후 생성된 파일들을 웹 서버에 배포하여 사용자에게 제공하는 방식이다.</p>	

19 하위주소 배포

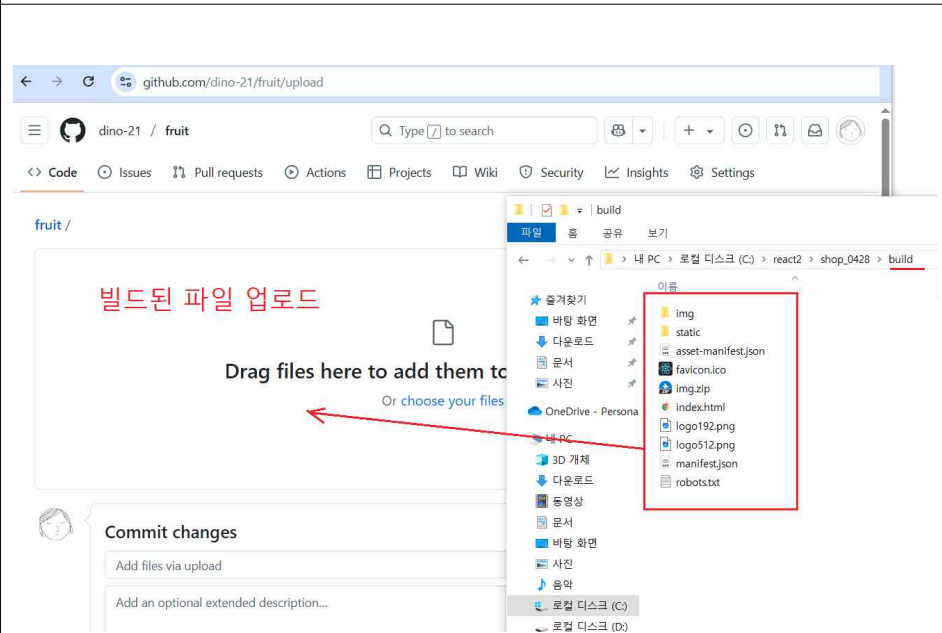
<p>https://github.com/ 로그인하기</p> 	<p>로그인 후 아이콘 클릭 > Your repositories 클릭</p> 
<p>New 클릭</p> 	<p>리액트 작업했던 하위경로와 같게 fruit 입력</p> <p>Create a new repository</p> <p>A repository contains all project files, including the revision history. Already Import a repository.</p> <p>Required fields are marked with an asterisk (*).</p> <p>Owner * <input type="text" value="dino-21"/> / Repository name * <input type="text" value="fruit"/></p> <p>Great repository names are short and memorable. Need inspiration? How a</p> <p>Description (optional) <input type="text"/></p> <p>① You are creating a public repository in your personal account.</p> <p>하단으로 스크롤하고 내려서 Create repository</p>

깃주소 생성 후 파일 업로드 버튼 클릭

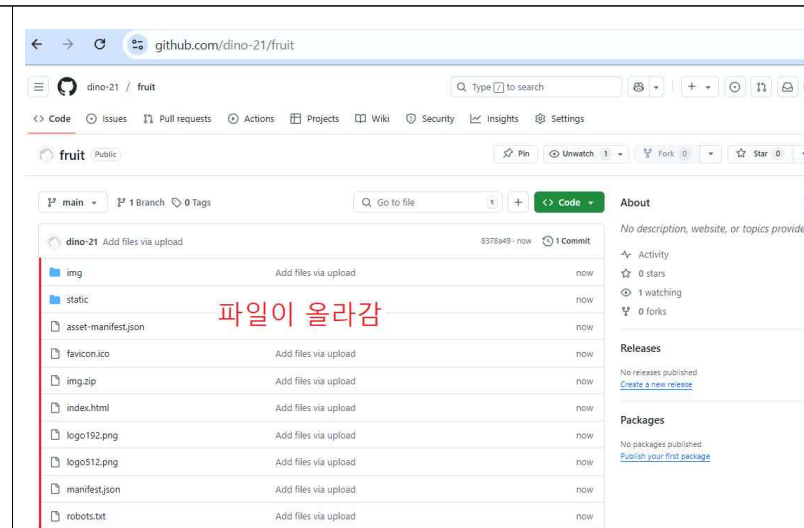
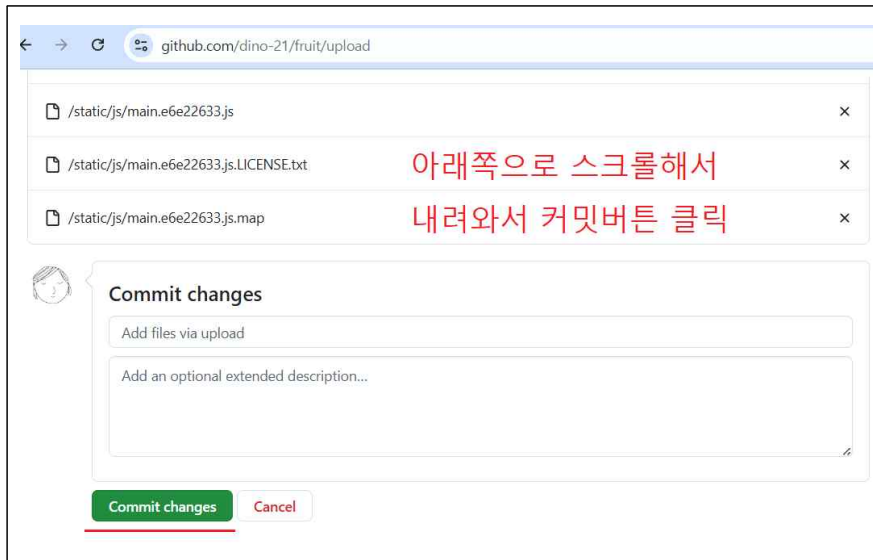


파일이 다 올라가면 아래쪽으로 스크롤> 커밋버튼 클릭

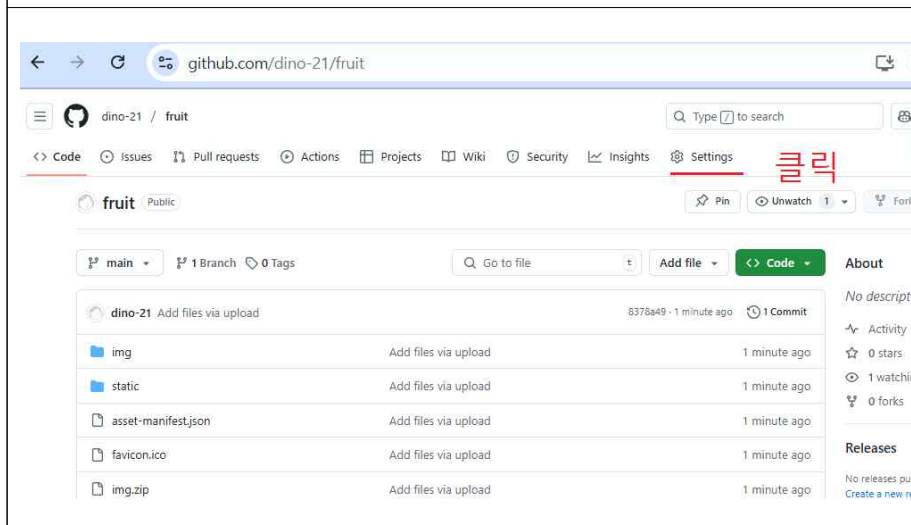
탐색기로 작업폴더이동 > 빌드 된 파일 > 빌드 된 파일 전부 선택 > 업로드



파일 올라갔는지 확인

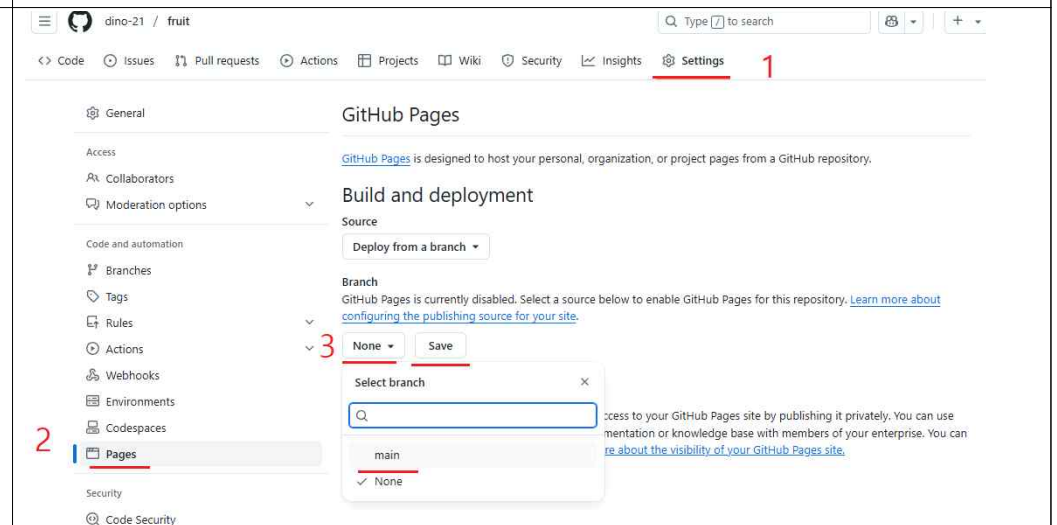


Settings 클릭



깃 배포 주소가 만들어지는데 시간이 조금 걸린다.

Pages > Branch > main을 root로 하고 저장



조금 지난 후 주소가 Pages가 누르면 배포 주소가 생긴다.(클릭해서 확인)

github.com/dino-21/fruit/settings/pages

dino-21 / fruit

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

GitHub Pages source saved.

General

Access

Collaborators

Moderation options

Code and automation

Branches

Tags

Rules

Actions

Webhooks

Environments

Codespaces

Pages

Security

Code Security

GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Build and deployment

Source

Deploy from a branch

Branch

Your GitHub Pages site is currently being built from the main branch. [Learn more about configuring the publishing your site.](#)

1 main / (root) Save

Learn how to [add a Jekyll theme](#) to your site.

Custom domain

Custom domains allow you to serve your site from a domain other than dino-21.github.io. [Learn more about configuring custom domains.](#)

Save Remove

2 몇분 후 클릭

github.com/dino-21/fruit/settings/pages

dino-21 / fruit

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

General

Access

Collaborators

Moderation options

Code and automation

Branches

Tags

Rules

Actions

Webhooks

Environments

Codespaces

Pages

Security

GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Your site is live at <https://dino-21.github.io/fruit/> 2 배포 주소 Visit site

Last deployed by dino-21 1 minute ago

Build and deployment

Source

Deploy from a branch

Branch

Your GitHub Pages site is currently being built from the main branch. [Learn more about configuring the publishing your site.](#)

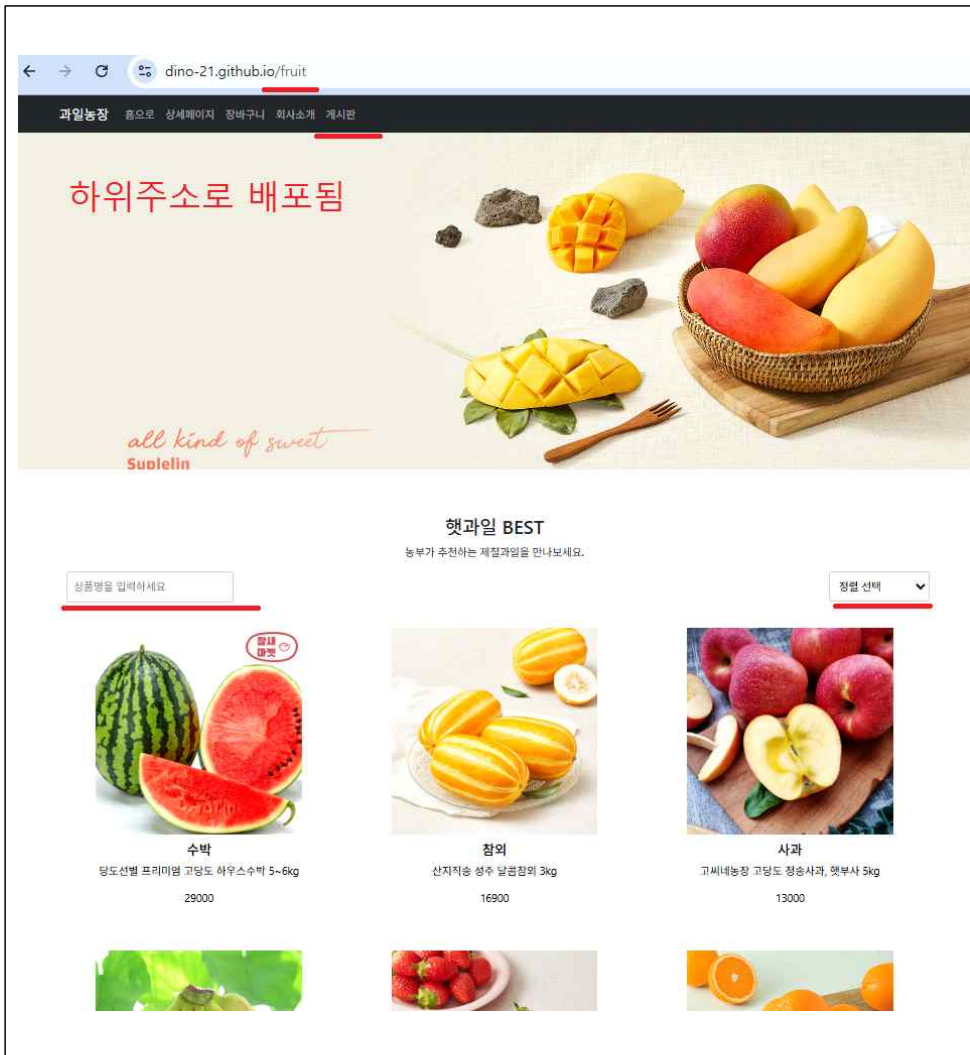
main / (root) Save

Learn how to [add a Jekyll theme](#) to your site.

Your site was last deployed to the [github-pages](#) environment by the [pages build and deployment](#) workflow.

1 2 배포 주소 클릭 생김

배포가 잘 되었는지 확인



<https://dino-21.github.io/fruit/>

- 1 이미지가 안깨지는지 확인하기
- 2 상세페이지도 클릭하고
- 3 비동기통신으로 데이터도 잘 가져와서 이미지가 안깨지는지 확인
- 4 게시판도 잘되는지 체크
- 5 장바구니에 상품도 추가해서 잘되는지 확인

- 끝 -