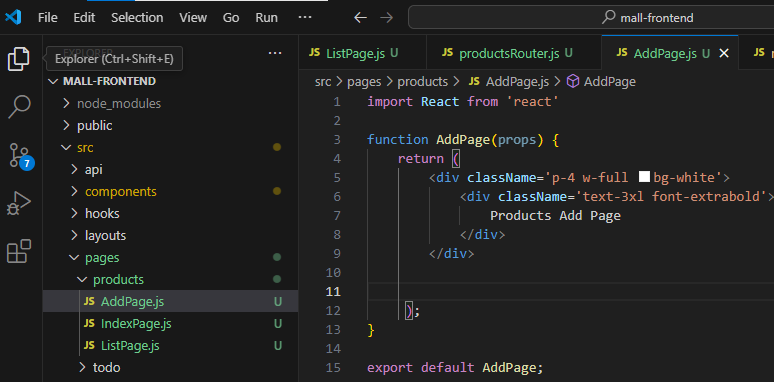
**#상품등록 컴포넌트 처리**

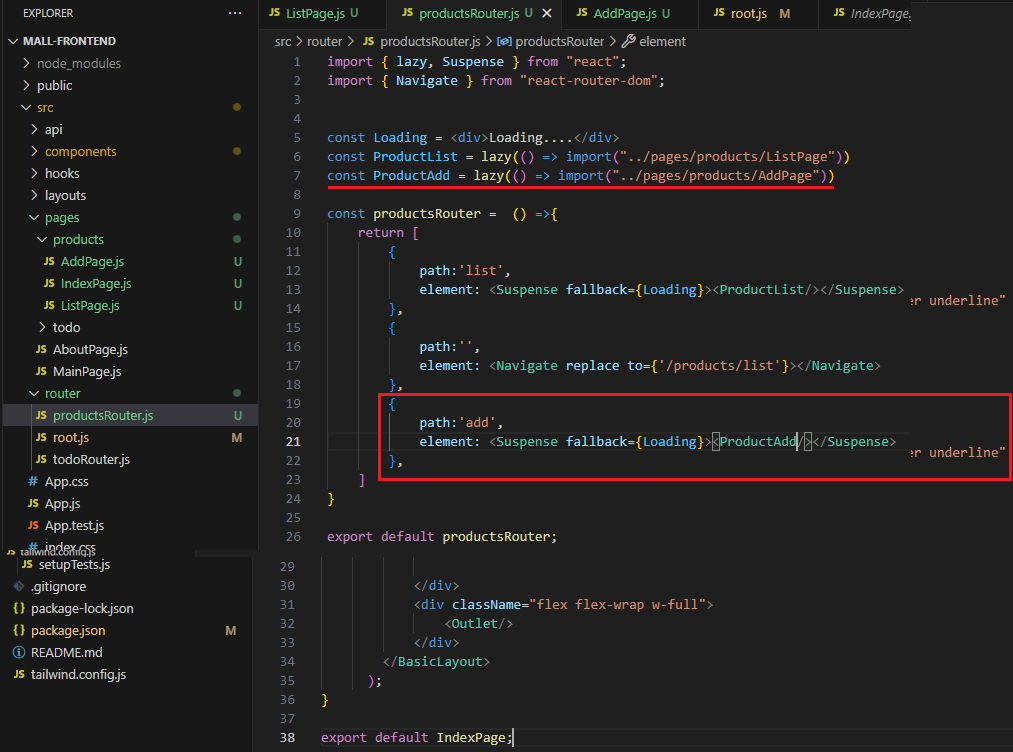
**상품등록 페이지 추가**

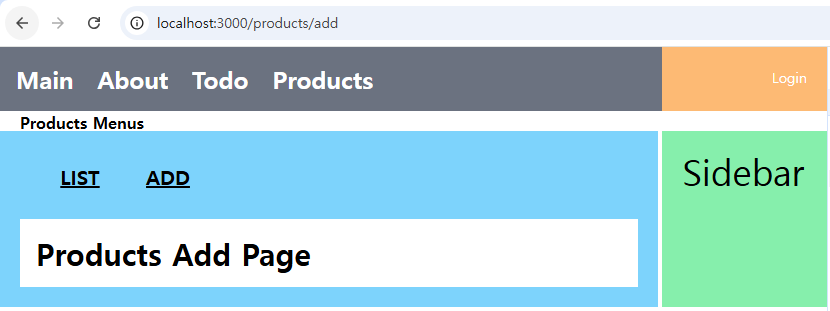
간단하게 작성해주고 라우팅 처리해준다.

**라우팅 처리를 위해 간단하게 작성부터 하는 것이다.**

****

라우팅 처리

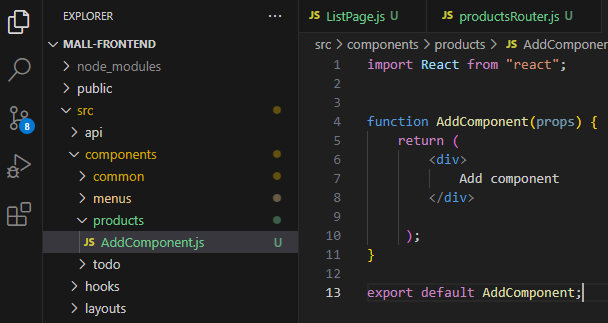




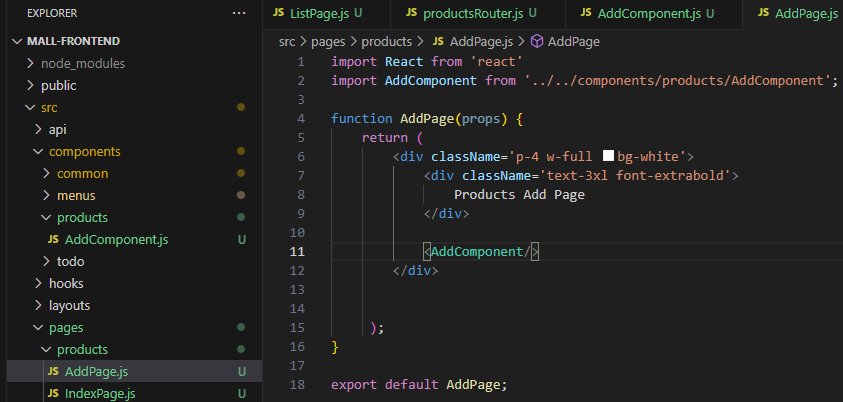
**페이지 추가와 추가 페이지의 라우팅 처리를 마쳤으니 페이지 구성시에 추가할 컴포넌트들을 구성해보자**

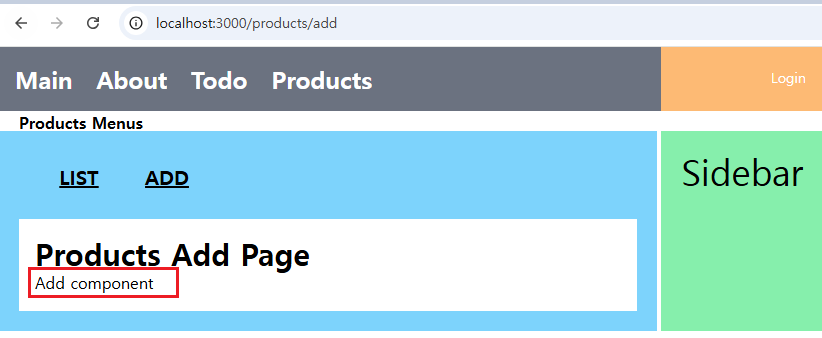
**상품 등록 컴포넌트 만들기**

상품 등록 페이지에 넣을 상품 등록 컴포넌트 추가.



상품 등록 페이지에 상품 등록 컴포넌트 추가시켜놓고 확인해보기.





**상품 등록 컴포넌트**

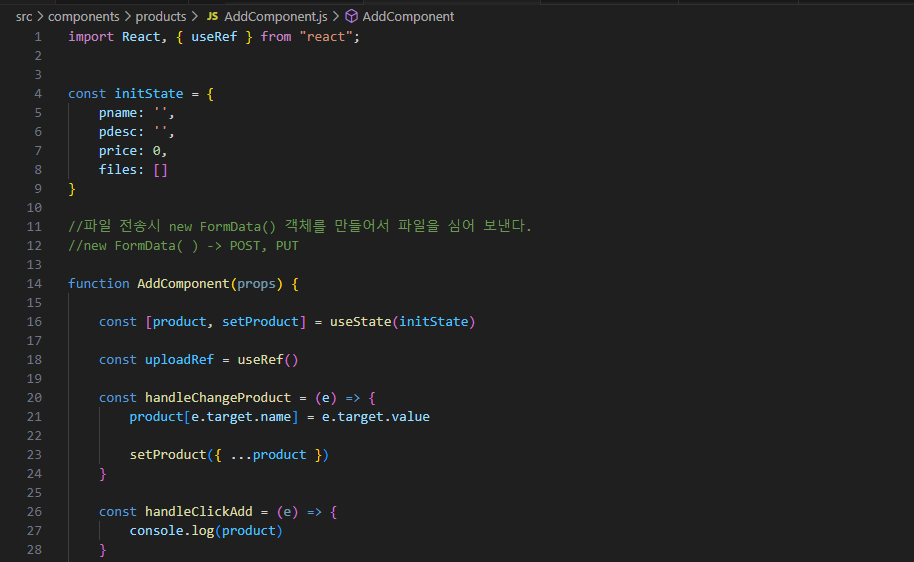
**html코드와 tailwindcss보다는 input태그와 그 속성들 그리고 함수들 위주로 다시보면 좋겠다.**

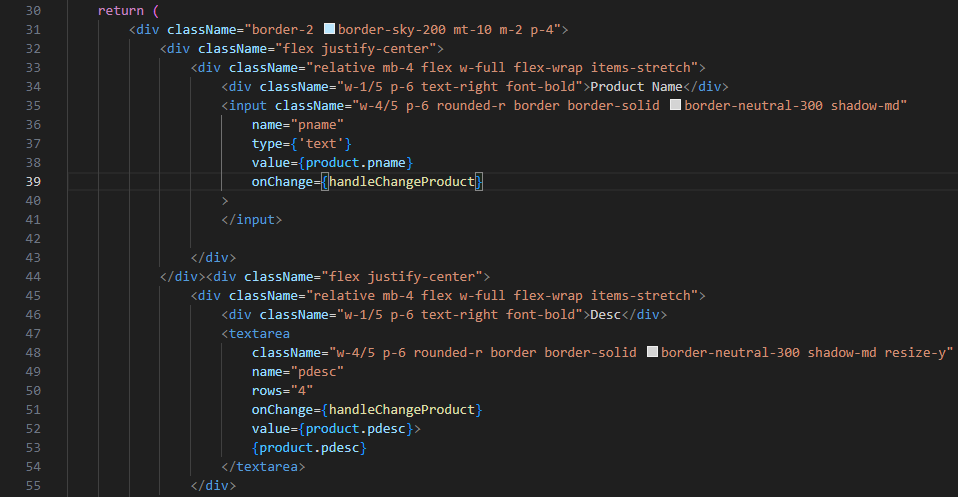
**상품정보에 대해 initState를 구성하고 상태관리를 위해 useState를 사용하였다.**

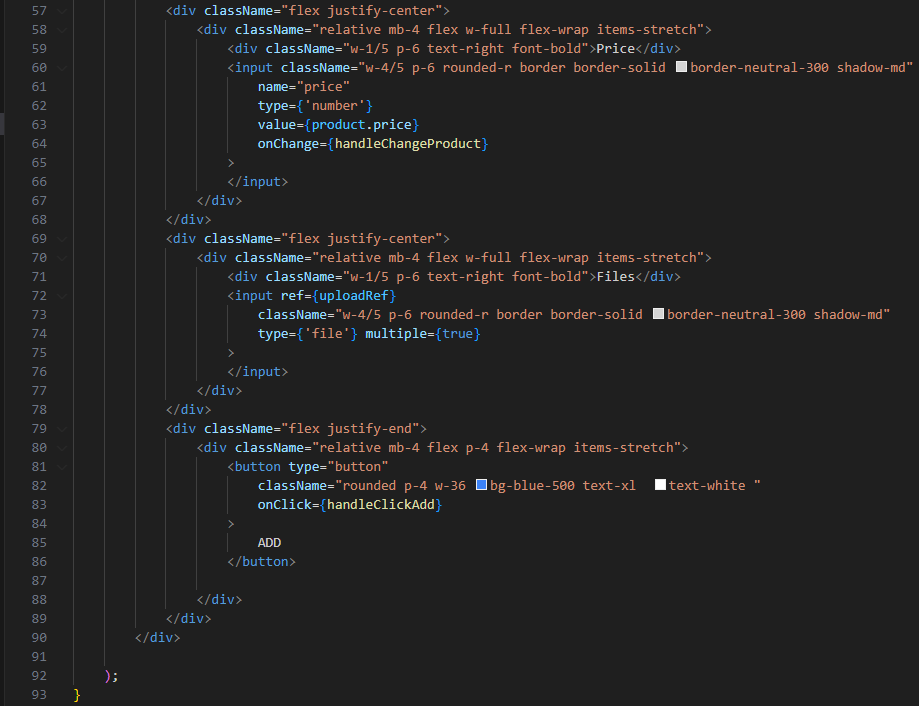
**입력을 받는 input태그의 onChange 속성을 이용하여 입력값이 변경될 때마다 사용자 지정 함수인 handleChangeProduct가 호출하여**

**상태값을 갱신해주고 있다. ...product는 객체 스프레드 문법으로 객체의 모든 속성을 새객체로 복사하여 setProduct하여 갱신하는데 이용한 것이다.**

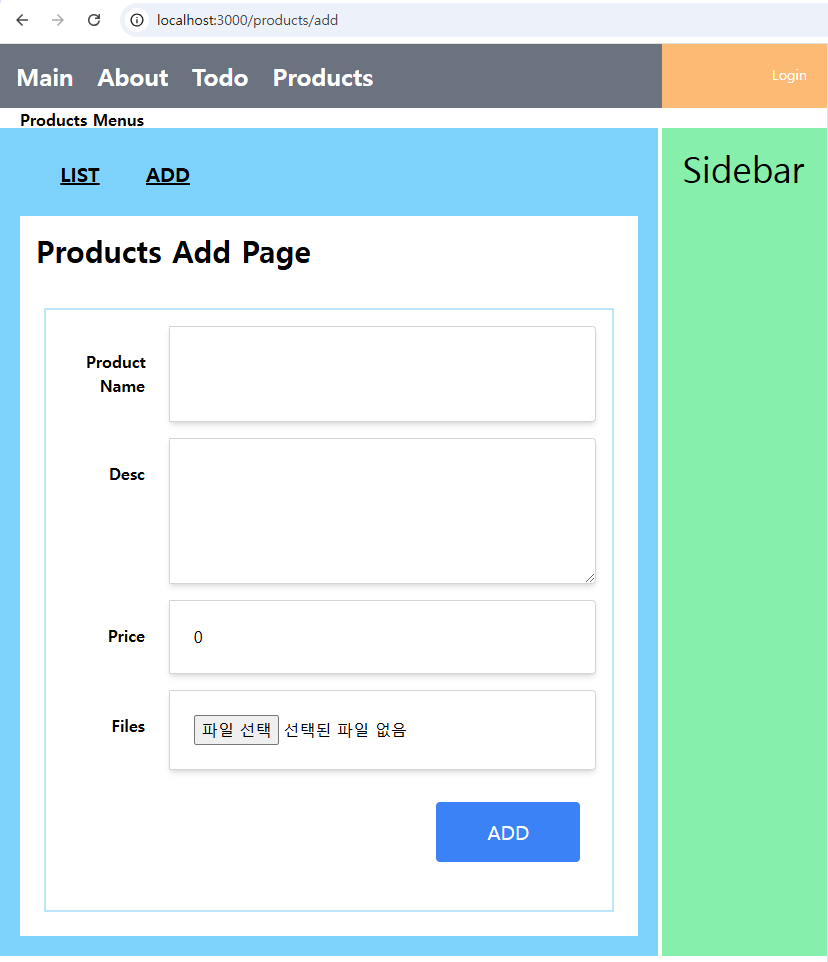
**useRef()는 React에서 파일 업로드 기능을 구현할 때 사용하는 참조 객체이다. 선택한 파일 정보를 상태관리하거나 접근할 수 있다.**





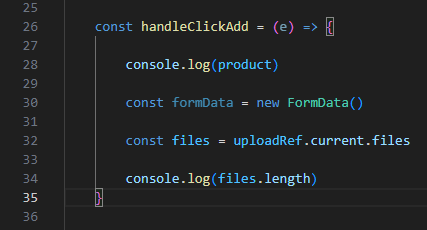


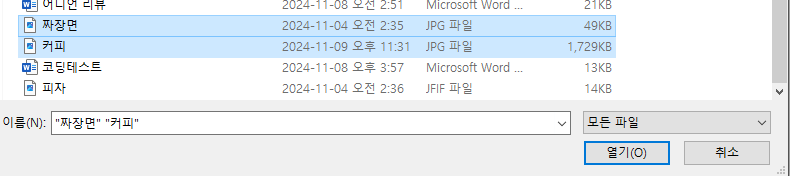


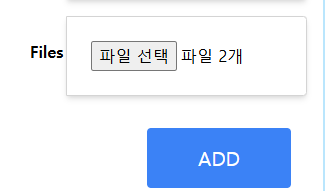


**Add버튼 기능 구현**

useRef 참조 객체를 사용하여 파일의 갯수를 확인할 수 있다. 콘솔로 찍어보면 잘 나오는 모습을 확인할 수 있다.



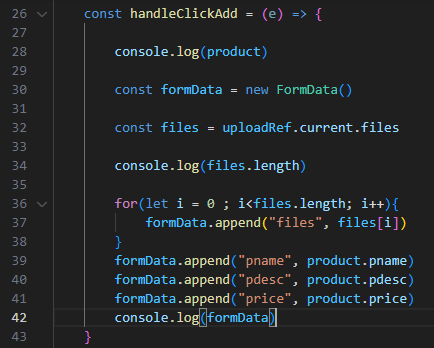






FormData객체를 통해 파일과 product를 초기화 하였다.

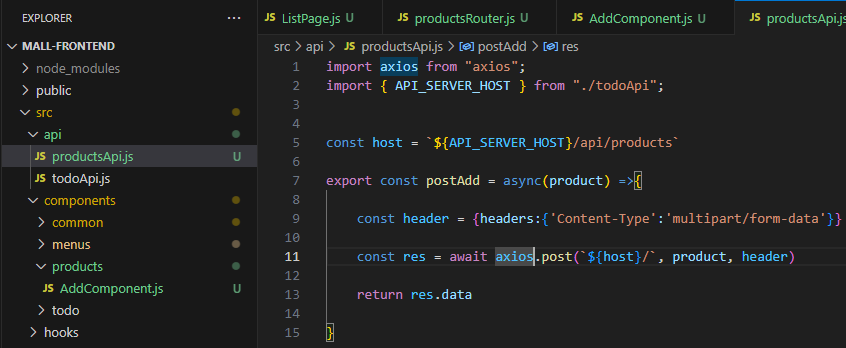
api로 formData를 보내야하는데 아직 api를 개발하지 않았다.



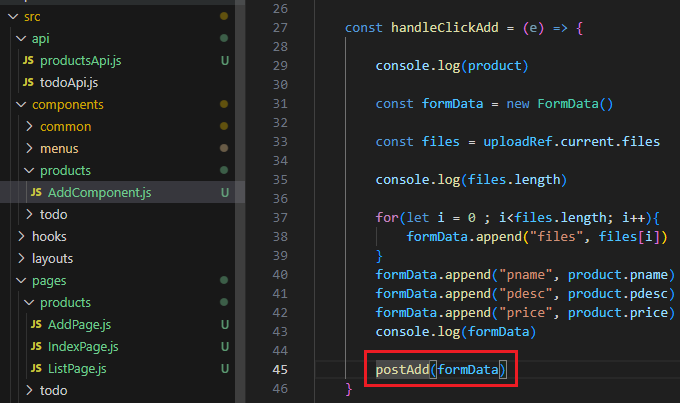
**상품 관련하여 백엔드 서버와 통신할 productsApi.js개발하기**

상품 등록을 위한 postAdd 메서드

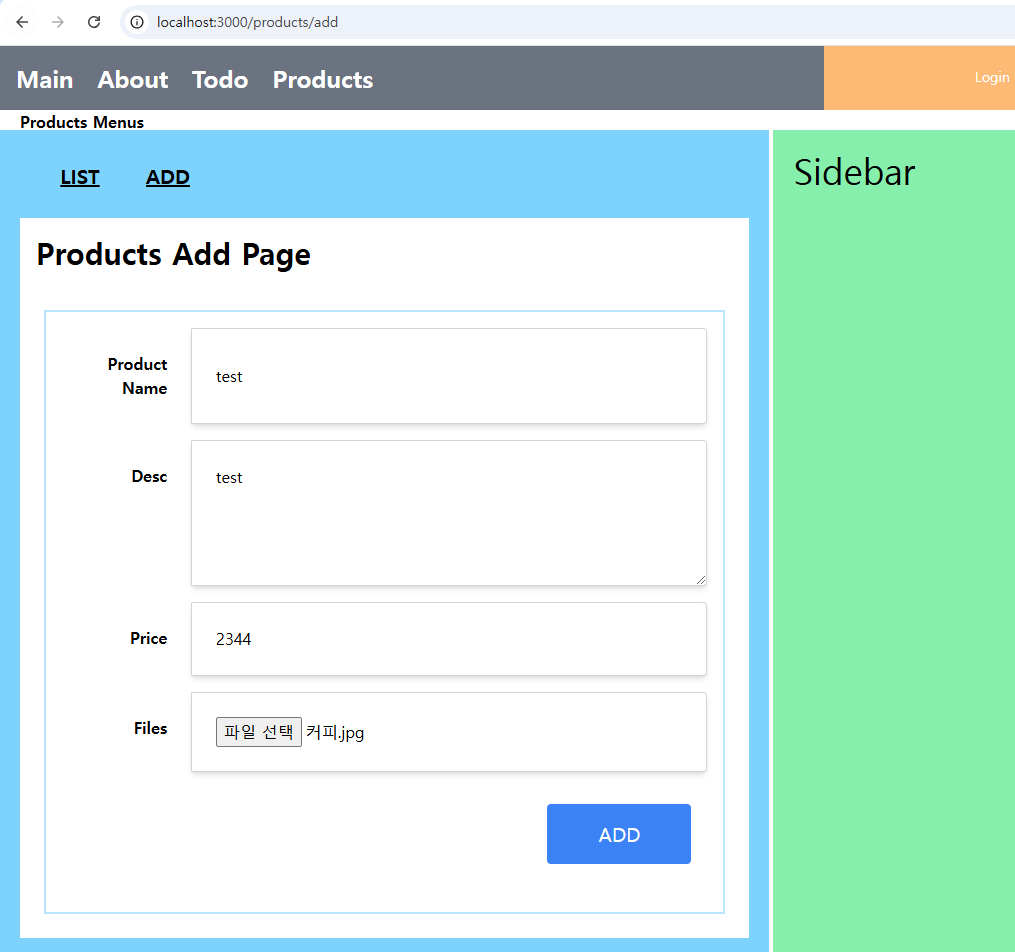
파일 전송을 위해 Content-Type으로 Multipart/form-data를 입력해준다.

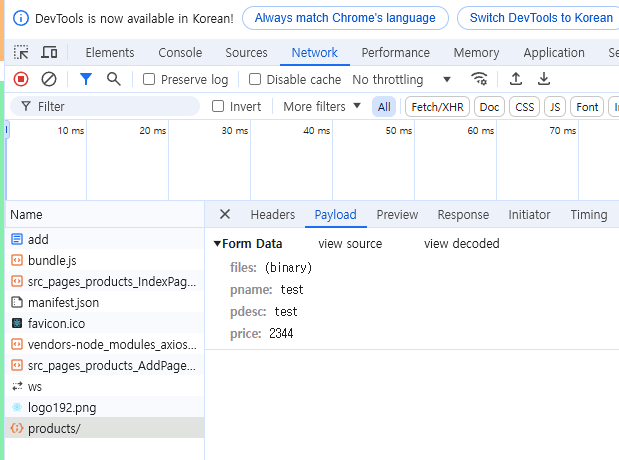


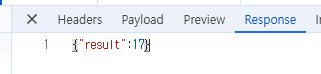
AddComponent의 Add클릭시 호출되는 함수 handleClickAdd의 마지막에 방금 만든 postAdd 메서드를 활용해 마저 작성해준다.



ADD버튼을 누르면 개발자모드>network의 product/ payload에서 formData가 어떻게 구성되어 전송되는지 확인해볼 수 있다.







정상적으로 새로운 상품이 등록되었다.

