



Bose PJ 정윤철



목차

- 1.원본 홈페이지 탐구
- 2.컴포넌트 구성
- 3.주요 기술
- 4.홈페이지 시현
- 5.향후 계획



원본 홈페이지 탐구

Bose는 세계적으로 인정받는 음향기기 전문 기업입니다.

심플한 디자인

간결하고 깔끔한 홈페이지 구성을 통해 사용자 경험을 개선합니다.

단순한 메뉴

주요 카테고리를 메뉴로 구성하여 사용자가 정보를 쉽게 찾을 수 있습니다.

유동적 정보

다양한 미디어 정보를 통해 제품에 대한 흥미를 유발합니다.

색상 변경 기능

상세페이지 외에서도 제품 색상 변경이 가능합니다.

컴포넌트 구성

라우터 시스템을 사용해 페이지마다 메인컴포넌트를 교체한다.

[1] 메인페이지

1 상단 컴포넌트 (메뉴바)

2 메인 컴포넌트

- 동영상배너
- 추천제품
- 서브배너
- vip 커스텀
- 구매이유
- sns미디어

3 하단 컴포넌트

[2] 서브페이지

1 상단 컴포넌트 (메뉴바)

2 메인 컴포넌트

- 이미지 배너
- 제품 리스트
- 제품동영상
- 구매이유
- 추천제품

3 하단 컴포넌트

[3] 상세페이지

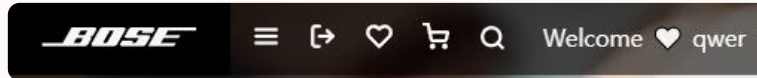
1 상단 컴포넌트 (메뉴바)

2 메인 컴포넌트

- 페이지별 상세내용
- 구매이유
- 추천제품

3 하단 컴포넌트

주요 기술 : 공통



메뉴바

1) 아이콘을 사용해 간단하게 표현

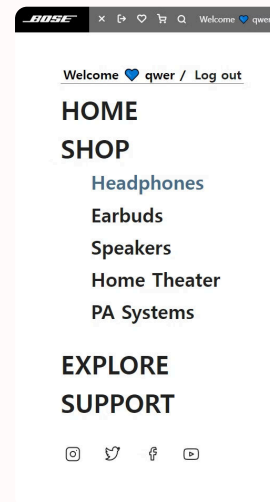
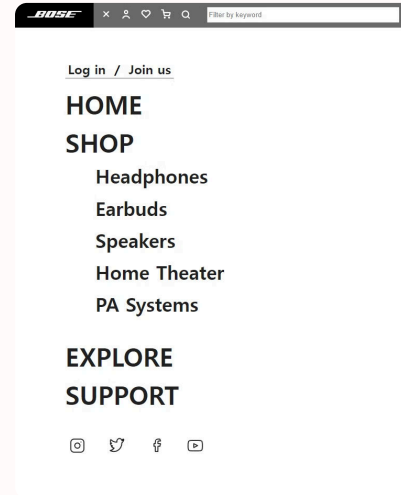
usestate (상태값) 활용

- 햄버거 버튼 클릭시 메뉴 나오기
- X버튼 클릭 or "esc" 시 닫힘
- 열린상태로 다른 버튼 클릭시 상태값 변경

2) 검색바 기능 구현

3) 전역변수 사용해서 로그인 상태 유지

4) 마우스 오버시 해당메뉴에 따른 이미지 변경



메인페이지 1. 상단배너 동적표현 구현

1) 로고배경 동영상 겹치기

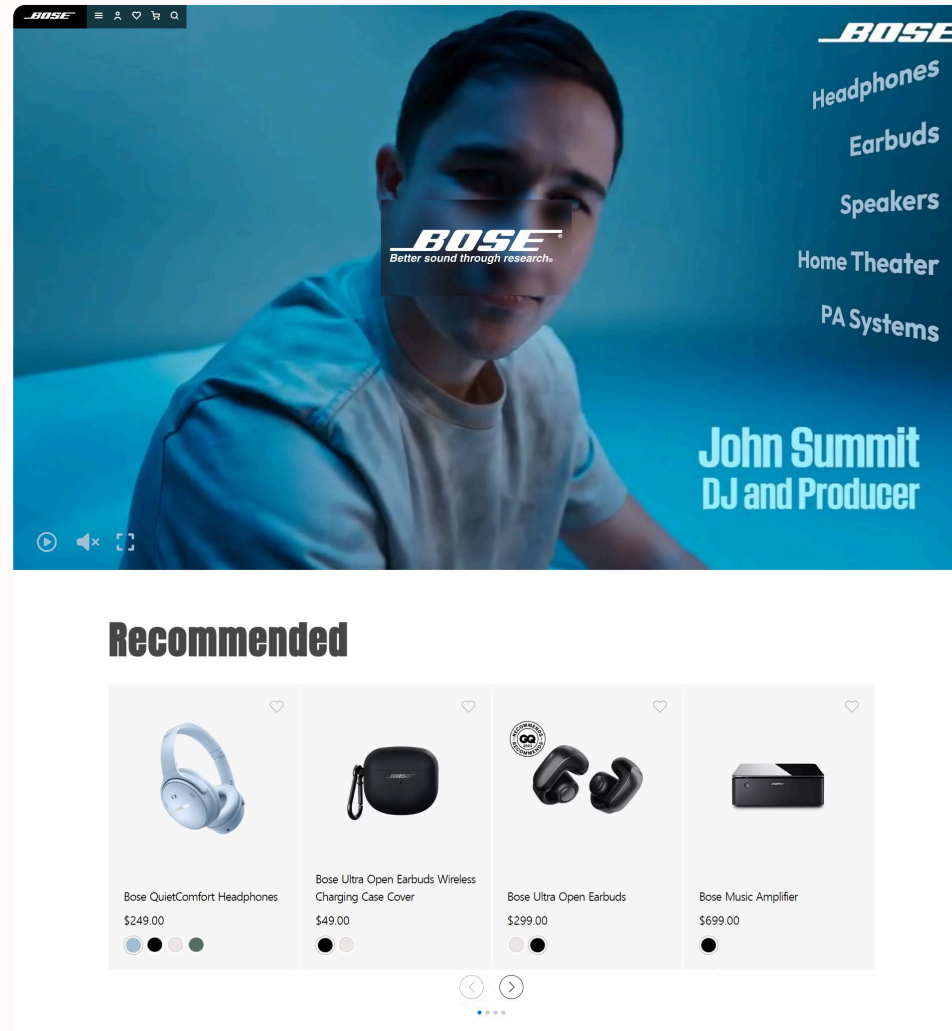
- 로고뒤 동영상 겹침
- 이동효과 **position : sticky**

2) 서브메뉴 동적표현

- 페이지 랜더링 시 측면에서 등장하는 애니 효과
- 원근감 설정 : **perspective+ Y축회전 -20%**

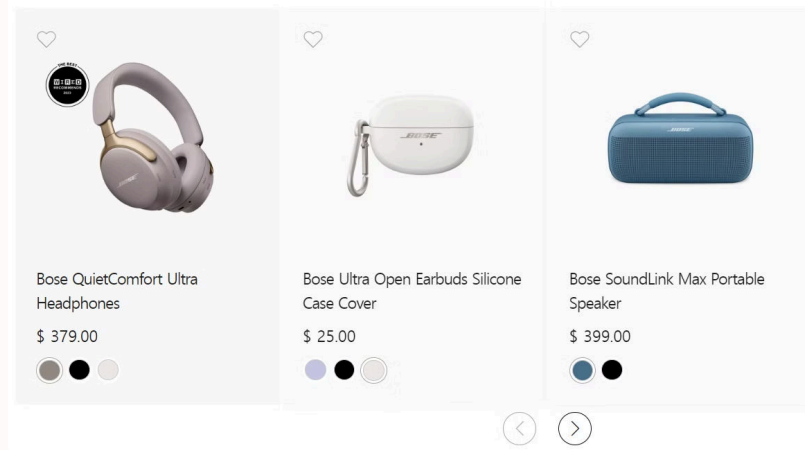
- 마우스 오버시 → 애니 효과(**scale & translate**) + (**filter:invert(1)**)

→ 음악이 퍼져나가는 듯한 효과



메인페이지 2.랜덤함수 사용

Recommended



- 제품 랜덤출력
- 품목별 개수 1~2 랜덤출력
- 품목내의 제품 랜덤
- 제품내에서 색상 랜덤

데이터 출력시 **map()** 메서드 사용해 해당 랜덤함수를 불러와 원하는 데이터를 넣어 사용한다.

랜덤함수 예시

```
// 랜덤 함수 1
const getRdItem = (arr) => {
  const rdIdx = Math.floor(Math.random()
    * arr.length);
  return arr[rdIdx];
}; ///////////////
```

함수사용 예시

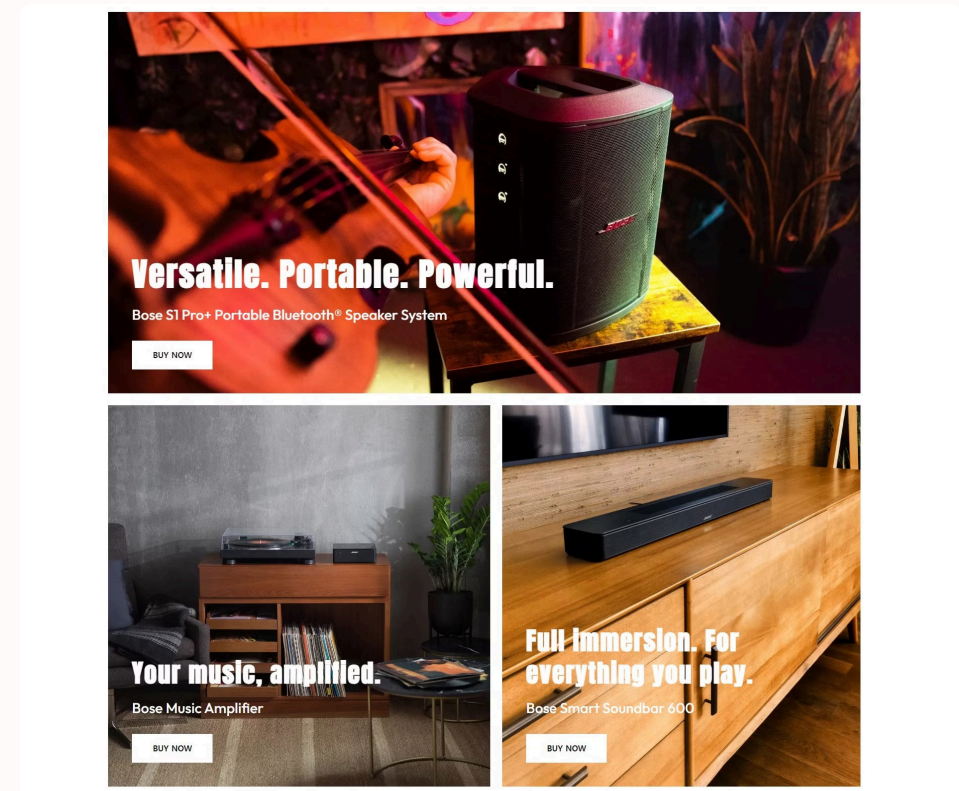
```
products.map((v, i) => {
  // 1.각 카테고리별로 1~2개 제품 랜덤선택
  const rdProduct = getRdItems(v);
  return rdProduct.map((prod, j) => {
    // 2.선택된 제품에서의 가지고있는 색 랜덤선택
    rdColor.current = getRdItem(prod.color);
    return (
      <SwiperSlide key={`-${i}-${j}`}
        className="rd-cat-cont">
```

기타 랜덤함수 사용

상단 동영상 & 오른쪽메뉴버튼 색상



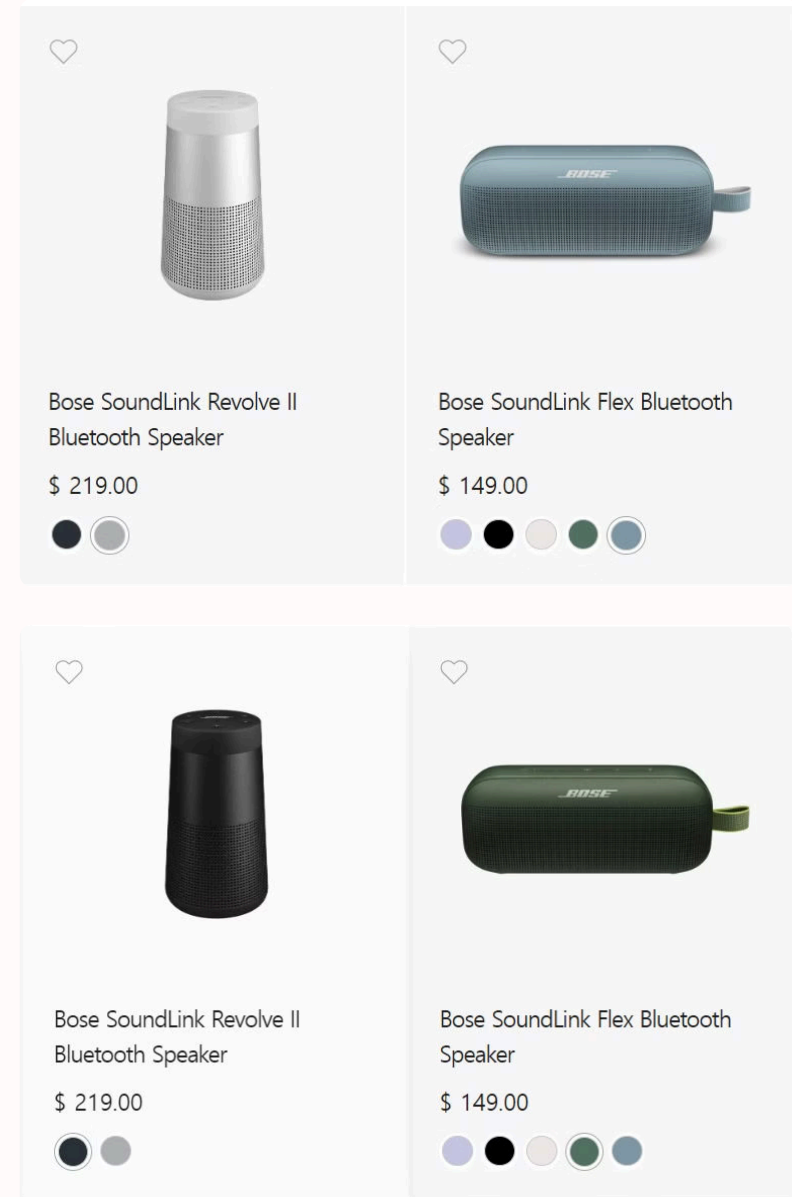
기타 배너 이미지&텍스트



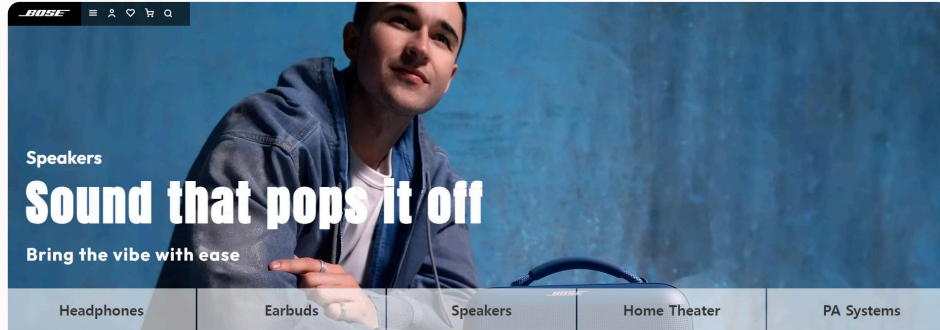
메인페이지 3. 제품 색상 클릭 변경

[색상변경 함수 작동 순서]

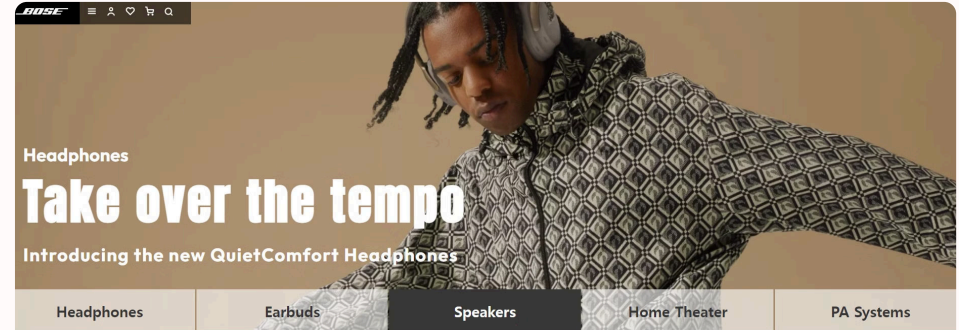
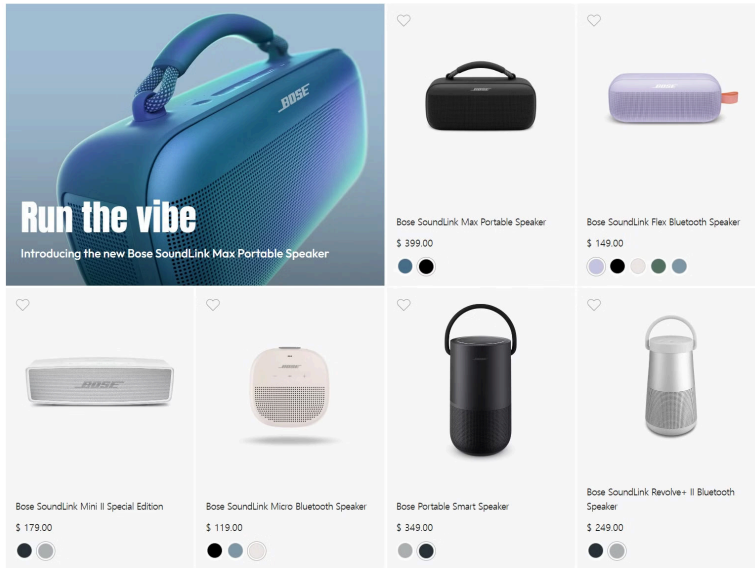
1. 색상버튼 클릭시 colorFn 함수 불러옴
 2. 버튼 자신의 부모parents()까지 올라감
 3. 그아래prev() 이미지를 find() 으로 찾음
 4. 이미지 경로 src 를 불러와 split("/") 으로 분할
 5. 분할된 배열데이터중 색상 부분[9] 번째 를 클릭한 버튼 자신의 색상으로 교체 후 join("/") 으로 합침
 6. 최종 이미지 경로 으로 교체시킴
- 참조 변수에 담긴 색상값 업데이트
 - 변경된 색상으로 체크표시



서브페이지



Shop Speakers



Shop Headphones

1. 배너 메뉴버튼 클릭시

- 해당 메뉴의 서브페이지로 이동
- 메뉴에 해당하는 배너 이미지 변경

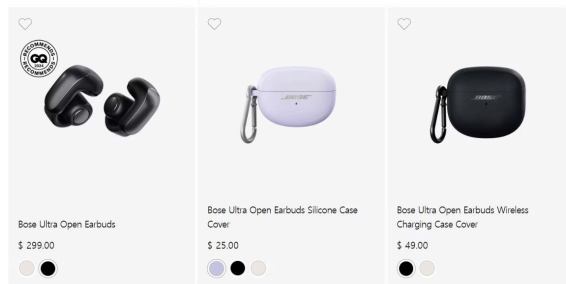
검색페이지

1. 검색 결과 일치 시 데이터 출력



Results found.

Search results for: "open", "3" items



2. 검색결과 불일치시

불일치 안내문구 표시

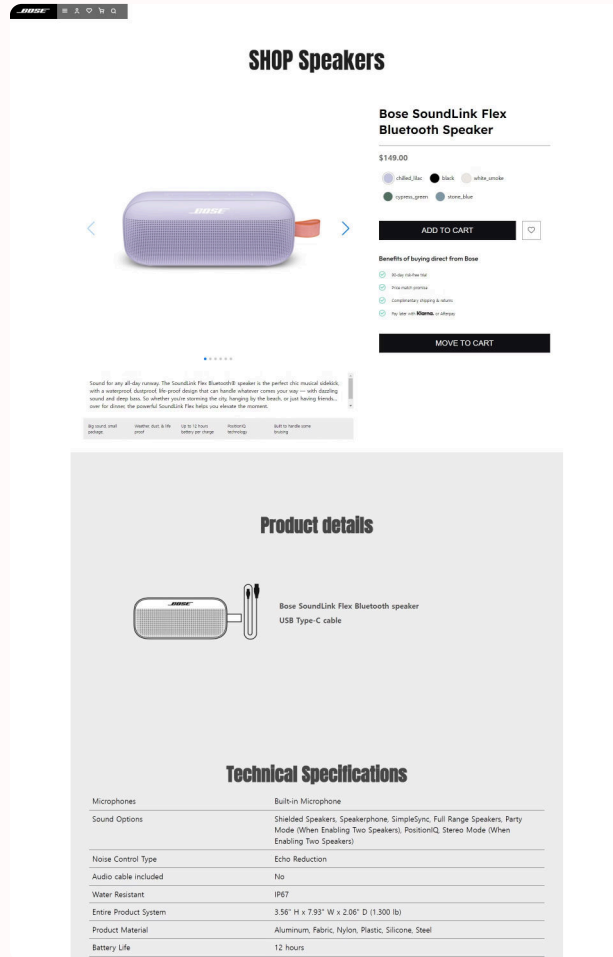
No results found.

Search results for: "○○", "0" items

Please search by product name, type, color, etc.

검색조건 : 제품명 / 타입 / 색상 등

제품 상세 페이지



1. 외부에서 제품 선택시 값전달 후 이동

해당 제품의 **색상 데이터**로 값전달

- 색상데이터는 **전역변수**를 사용

(색상 업데이트를 위해 전역변수로 사용함)

- 나머지 데이터는 **state** 값으로 전달

2. 받아온 데이터 출력

왼쪽박스 : **swiper** 를 사용하여 다양한 이미지 출력

오른쪽박스 : 제품의 기본 내용 출력 + **position sticky** 사용

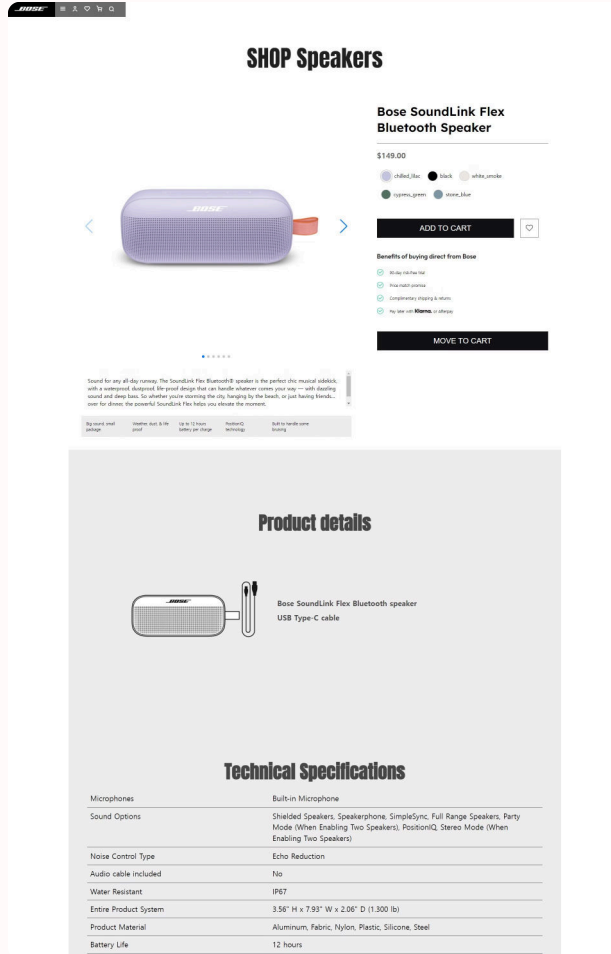
아래 박스 : 상세정보를 출력

3. 색상 클릭시

전역변수에 색상 업데이트

+ 해당 색상의 데이터로 **swiper** 의 이미지 교체

제품 페이지



4. 장바구니 or 위시리스트 버튼 클릭시

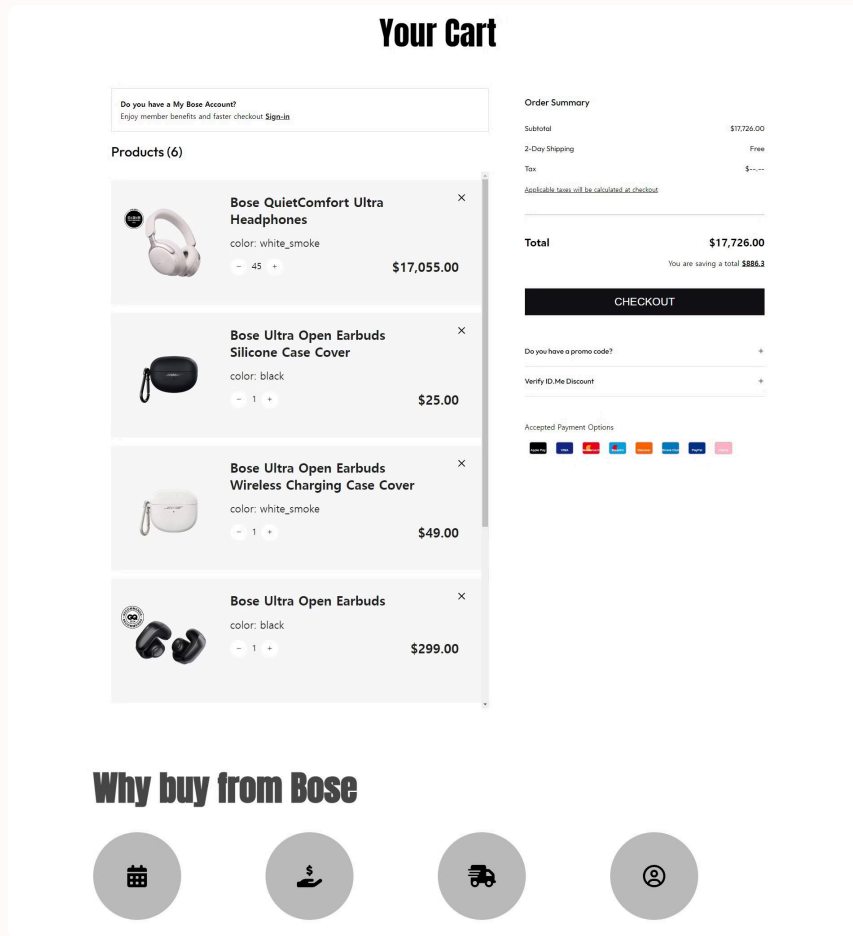
로컬스토리지로 해당 데이터 추가 + 중복검사

1) 중복검사 통과시 → 장바구니 이동 여부 메시지 출력

2) 중복검사 실패시 → 중복메시지 출력

```
if (retSts) {  
    alert("This item is already in your  
    cart!");  
    return;  
}  
  
const userConfirmed = window.confirm("Would  
you like to go to your shopping cart?");  
if (userConfirmed) {  
    initSet();  
} else {  
    console.log("사용자가 취소를 눌렀습니다.");  
}
```

장바구니 페이지



1. 로컬스토리지에서 정보를 불러오기

데이터가 없을경우 빈배열 [] 으로 만들기

2. 데이터 출력

로컬스토리지의 정보를 불러와 **map()** 으로 출력

3. 로그인 상태 여부

비로그인시 안내문구 박스 출력 / 로그인시 삭제

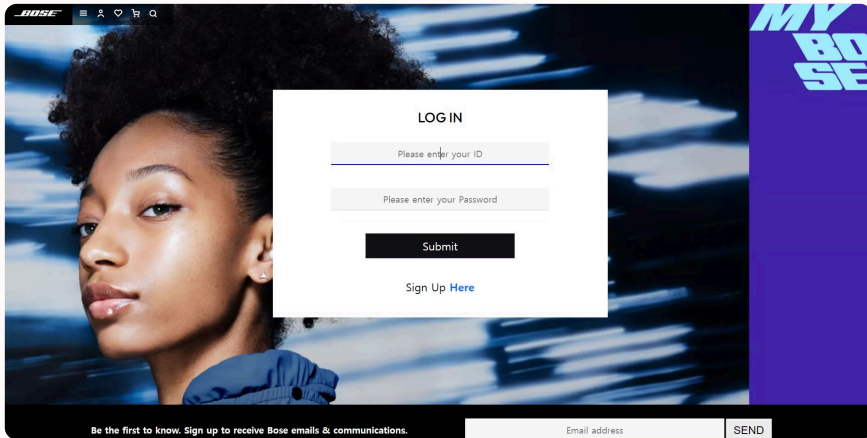
4. 수량 버튼 클릭시

해당 제품의 수량변경 + 총가격 변경

5. 구매하기 버튼 클릭시

로그인 상태에따라 구매페이지 이동/ 메시지출력

로그인 페이지



로그인시 이전 위치에따른 페이지 이동

1.장바구니 에서 로그인시

장바구니로 다시 이동

2.외부에서 로그인시

홈으로 이동

향후 계획

위시리스트 추가항목 업데이트

구매페이지 정보 업데이트

회원가입시 주소 추가

기타 페이지