



Portfolio for React Project

목차

01 - 기획

X1 - 컨셉 선정 이유

X2 - 컬러 가이드

X3 - 페르소나

X4 - 와이어프레임

02 - 구현

X1 - 프로토타입

X2 - 코드 구현

03 - 마무리

X1 - 평가 및 소감

X2 - 마무리

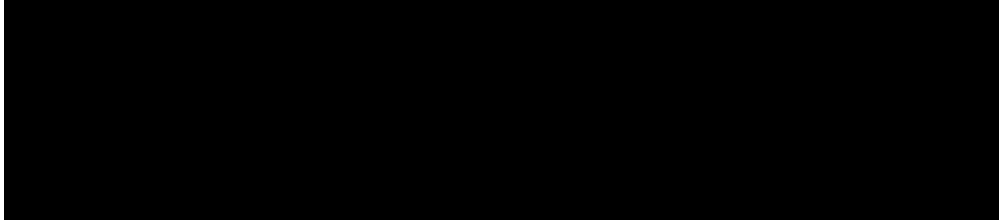
컨셉 선정 이유

리액트를 학습한 지 오래되지 않았기 때문에
처음 프로젝트를 선정할 때는 복잡한 구조를 가진 웹사이트보다는
비교적 직관적인 구조를 갖춘 사이트를 우선 고려했습니다.

그 과정에서 젠틀몬스터(Gentle Monster) 공식 홈페이지를 접하게 되었고
이 사이트는 단순한 구조임에도 불구하고 세련된 디자인과 감각적인 레이아웃을 통해
브랜드의 아이덴티티를 효과적으로 전달하고 있었습니다.

특히, 개발 난이도가 지나치게 높지 않음에도 불구하고
완성도 있는 결과물을 구현할 수 있다는 판단에서,
학습용 클론 프로젝트의 대상으로 선정하게 되었습니다.

#000000

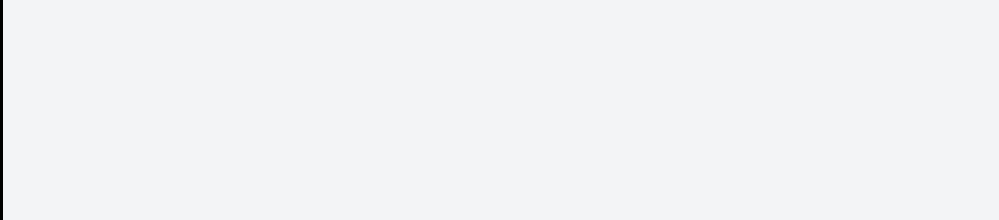


#222222



Color Guide
컬러 가이드

#F3F4F6



#FFFFFF



.PERSONA



이름 : 차이현 나이 : 24 성별 : 여

상황 : 8월 하와이 여행을 준비 중

직업 : 대학생 성격 : 활발하고 트렌드에 민감하며, 사진 찍는 것을 즐김

사용 습관 :

- 주로 스마트폰을 사용하며 , 외출 중이나 잠들기 전 침대에서 쇼핑하는 일이 많다.
- 상세 스펙보다는 이미지 중심의 탐색을 선호 하며, 스타일링 사진이나 룩북을 통해 제품을 판단한다
- 마음에 드는 제품을 구매한 후 SNS에 착용 사진을 공유하는 경우가 잦다.

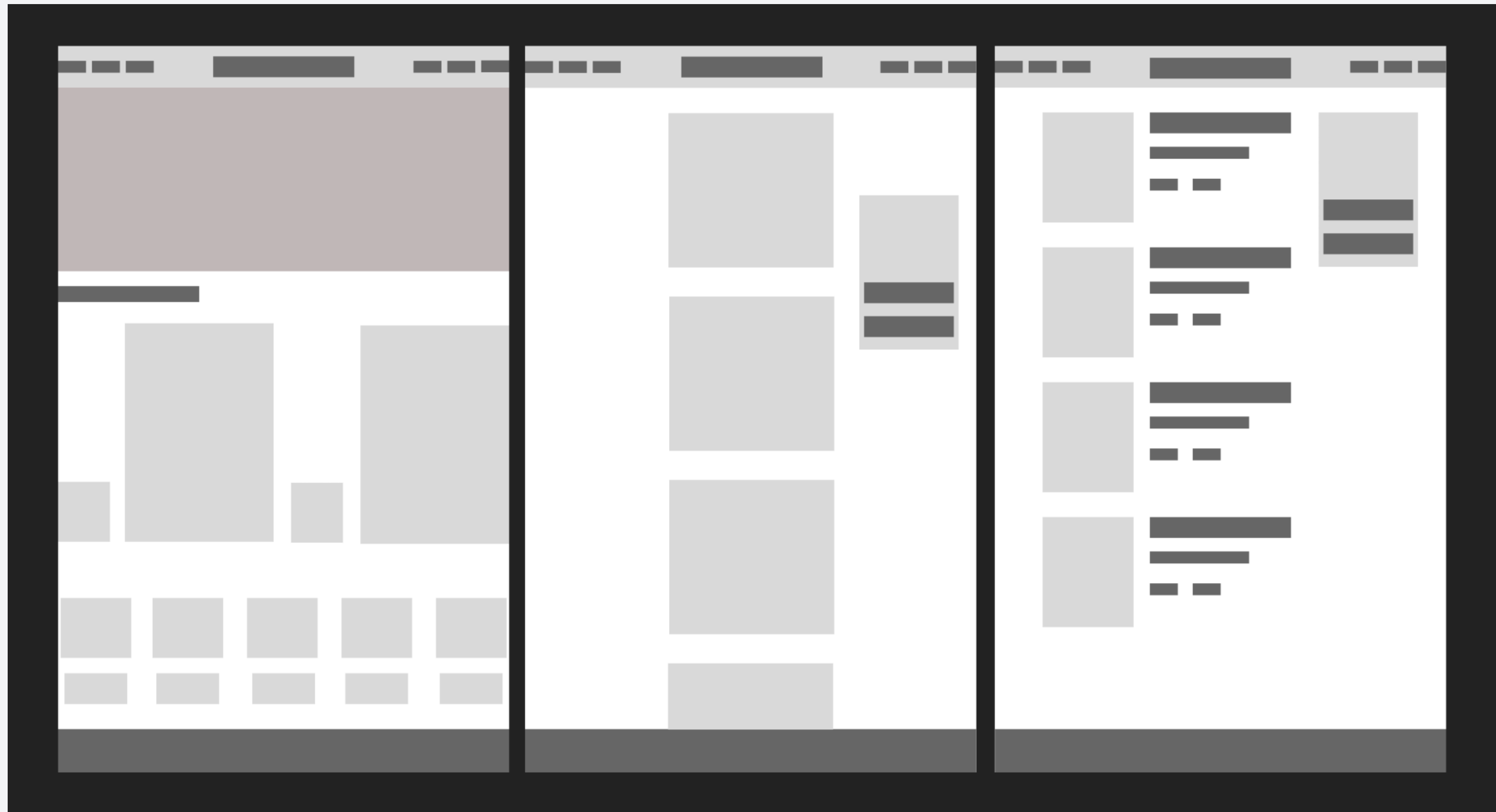
니즈 :

- 여행 중 스타일을 강조할 수 있는 아이웨어
- 감성적인 무드와 어울리는 제품
- 흔하지 않은 유니크한 디자인 선호

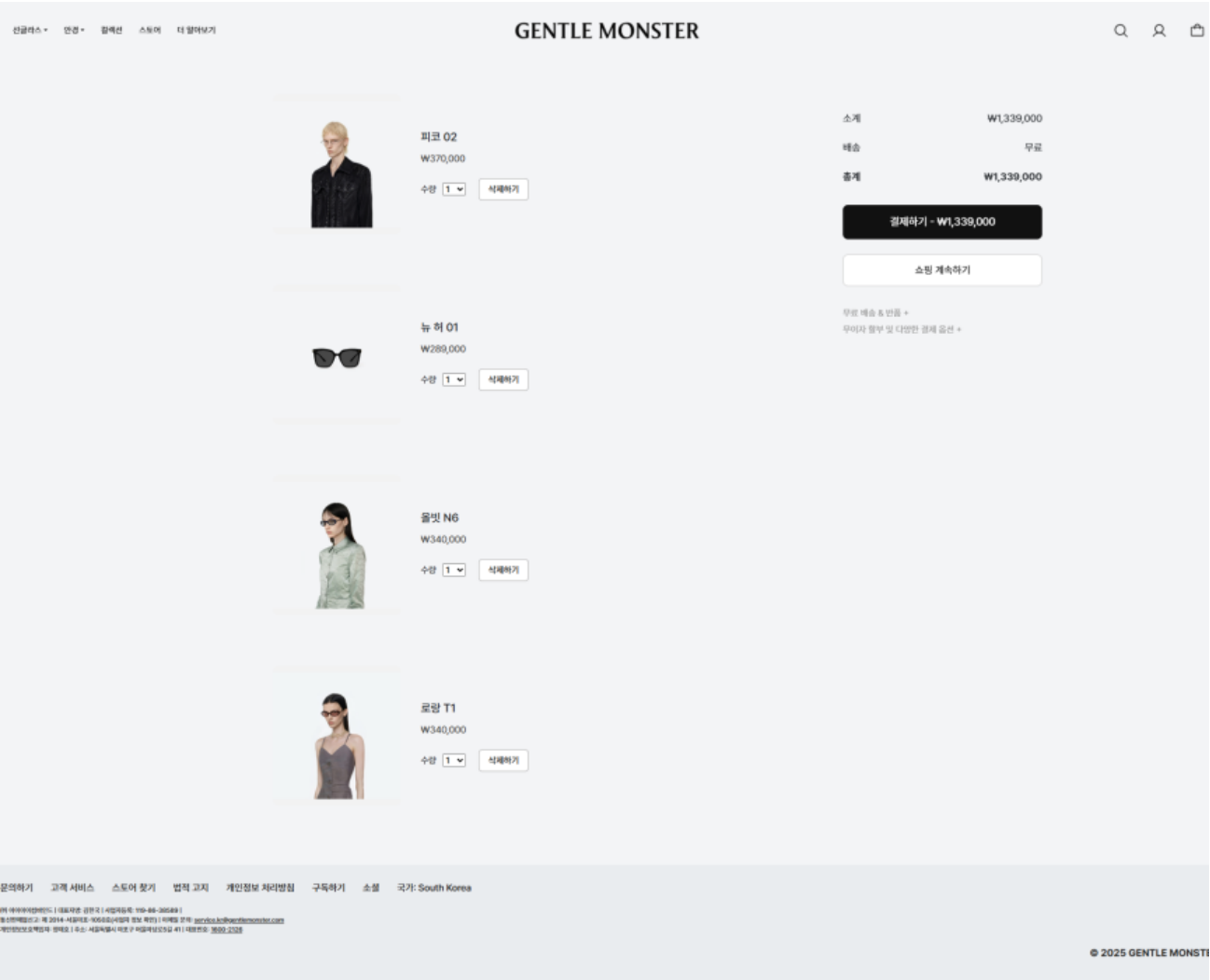
MAIN

DETAIL

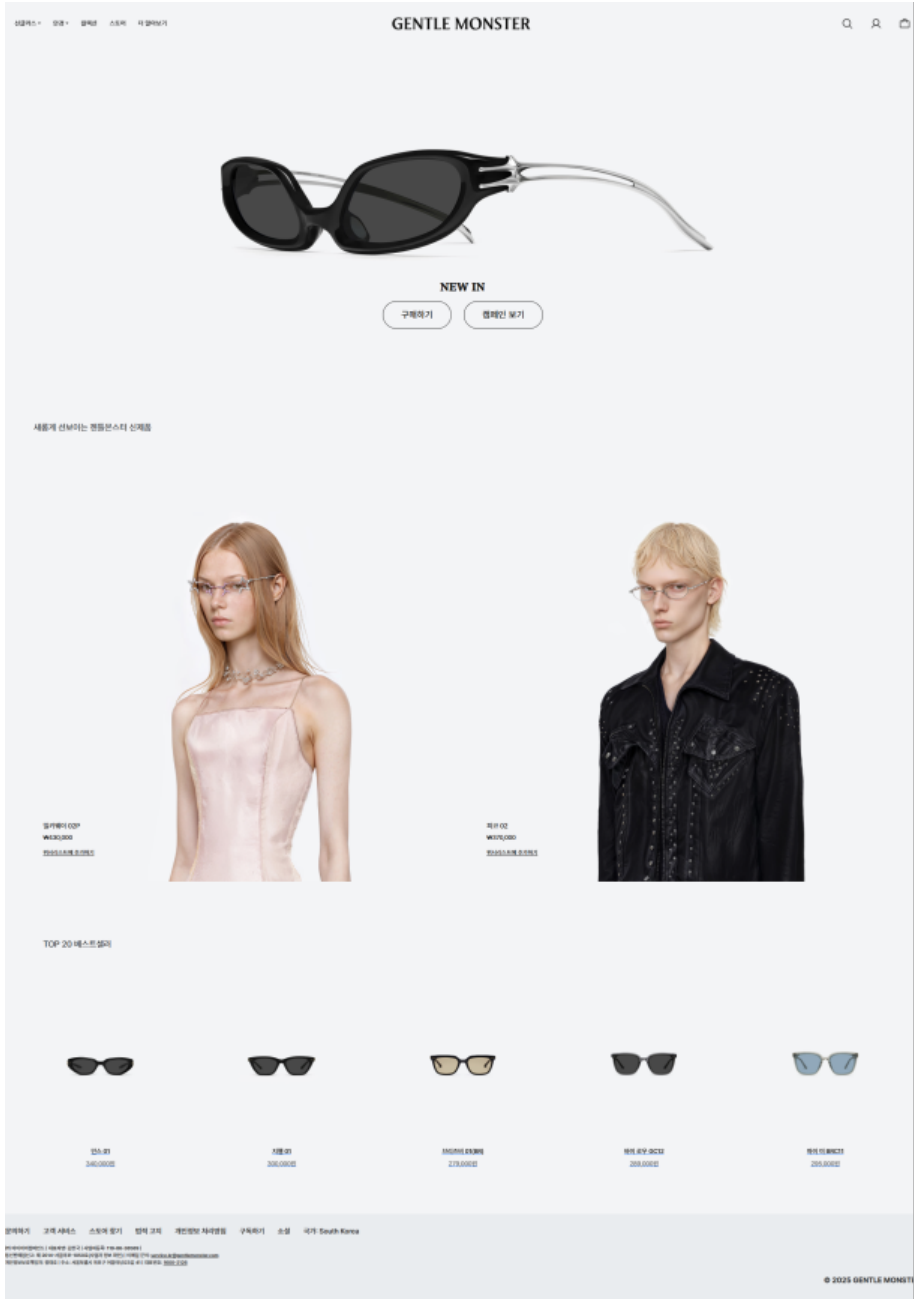
CART



와이어 프레임



프로토타입



PROTOTYPE

코드 구현

메인 페이지의 Mainslider.jsx 컴포넌트는
Swiper 모듈을 활용해
자동 전환 슬라이드 배너를 구현하며,
영상 및 이미지 콘텐츠를
SwiperSlide 단위로 구성하고 있습니다.
첫 번째 슬라이드는 muted 상태의
looping video를 사용하여
시각적 임팩트를 극대화하며, 슬라이드 중앙에는
컬렉션 타이틀과 CTA 버튼('구매하기', '안경 보기')을
배치하여 사용자의 즉각적인 행동 유도를 유도합니다.
autoplay 속성에는 delay: 14000ms 및
disableOnInteraction: false 설정을 적용하여
비인터랙션 환경에서도 지속적인 콘텐츠 노출이
가능하도록 구성되어 있습니다.

37°45' MainPage N, 127°32' Mainslider.jsx E

```
function Mainslider() {  
  return (  
    <div className="main-slider-wrap">  
      <Swiper  
        className="main-swiper"  
        modules={[Autoplay]}  
        loop={true}  
        slidesPerView={1}  
        autoplay={{delay: 14000, disableOnInteraction: false}}  
      >  
        {/* 1번: 영상(흰색) */}  
        <SwiperSlide>  
          <div className="main-slide-bg">  
            <video src={process.env.PUBLIC_URL + "/video/main1.mp4"} autoPlay loop muted playsInline className="main-slide-media" />  
          </div>  
          <div className="main-slide-center slide-white">  
            <h2>POCKET COLLECTION</h2>  
            <div className="main-slide-btns">  
              <button>구매하기</button>  
              <button>안경 보기</button>  
            </div>  
          </div>  
        </SwiperSlide>  
        {/* 2번: 이미지(검은색) */}  
        <SwiperSlide>
```


Detail2.jsx는 URL 파라미터를 통해
특정 상품 정보를 조회하고, 해당 데이터에 기반한
상세 UI를 렌더링하는 컴포넌트입니다.
useParams() 혹은 받아온 id를 기반으로
productData에서 대상 상품을 식별하며,
선택 가능한 컬러 정보는 useState를 통해 관리됩니다.

상품명과 가격은 item.title, item.price를 통해 출력되며,
색상 선택 영역은 item.colors.map()을
통해 선택지 UI를 동적으로 렌더링합니다.
각 색상 박스는 클릭 시 setSelectedColor(idx)를
통해 현재 선택된 색상 상태를 업데이트하며,
선택 여부에 따라 "selected" 클래스를 동적으로 부여합니다.

하단에는 현재 선택된 색상의 라벨 정보를 텍스트로 보여주고,
장바구니 추가 버튼은 handleAddToCart() 함수와 연결되어
선택된 옵션과 함께 제품을 장바구니에 추가하는 인터랙션을
담당합니다.

```
function Detail2() {  
  const { id } = useParams();  
  const item = productData.find(p => p.id === parseInt(id));  
  const [openSections, setOpenSections] = useState([]);  
  const [selectedColor, setSelectedColor] = useState(0);
```

```
  {/* 오른쪽 정보 영역 */}  
  <div className="detail-info">  
    <h2 className="detail-title">{item.title}</h2>  
    <p className="detail-price">₩{item.price.toLocaleString()}</p>  
  
    <div className="color-picker">  
      {item.colors?.map((c, idx) => (  
        <div  
          key={idx}  
          className={`color-box${selectedColor === idx ? " selected" : ""}`}  
          style={{ background: c.code }}  
          onClick={() => setSelectedColor(idx)}  
          title={c.label}  
        />  
      ))}  
    </div>  
    <p className="color-label">  
      {item.colors?.[selectedColor]?.label || "색상명 미정"}  
    </p>  
    <button onClick={handleAddToCart} className="cart-btn">쇼핑백에 추가하기</button>
```

코드 구현

36°18' /gmshop/detail2/1 N, 128°10' Detail2.jsx E

코드 구현

Cart.jsx는 Redux 스토어에 저장된 장바구니 데이터를 기반으로, 상품 목록을 렌더링하고 수량 조절, 삭제, 금액 계산, 결제 요청 등 구매 전 단계의 핵심 로직을 처리하는 컴포넌트입니다.

각 상품은 썸네일, 제목, 가격, 수량 <select>로 구성되어 있으며, 수량 변경 시 dispatch(changeCount())를 통해 전역 상태가 즉시 갱신됩니다. 총액은 .toLocaleString()으로 포맷되어 출력되며, 개별 삭제 및 장바구니 전체 초기화 기능도 포함됩니다.

CartModal.jsx는 동일한 장바구니 데이터를 요약 형태로 모달에 출력하는 컴포넌트로, 상품 리스트와 금액 정보 외에 수량 조절, 장바구니 비우기, 결제 버튼 등을 제공합니다. 외부 클릭 시 모달이 자동으로 닫히도

100

35°50' /gmshop/cart N, 127°55' CartModal.jsx E ,128°02' Cart.jsx E

프로젝트를 마무리 하며....

리액트 수업을 수강하면서 프로젝트 진행 전에는 걱정과 부담감이 많았습니다.

컴포넌트를 일일이 나누어 관리하고, 정리된 구조로 유지해야 하는
리액트의 작업 방식이 처음에는 성격과 잘 맞지 않아 어려움이 있었습니다.

하지만 실제 프로젝트를 진행하면서 리액트가 제공하는
모듈화된 구조와 효율성의 장점을 경험하게 되었고,
결과적으로 프로젝트를 성공적으로 마무리할 수 있었습니다.

이러한 경험 덕분에 앞으로 개발 작업을 진행할 때 리액트를
적극적으로 활용하고자 하는 자신감도 얻었습니다.

다만, 개인 프로젝트 일정이 팀 프로젝트와 겹쳐 사이트 구현 과정
다소 미흡하게 마무리된 점은 아쉬움으로 남습니다.

그럼에도 팀 프로젝트에서 좋은 결과를 얻어 전반적인 성취감은 매우 높았습니다.

이번 경험을 계기로 앞으로 더욱 리액트에 대한 숙련도를 높이고,
기술적 역량을 꾸준히 갈고닦아 제 것으로 만들겠다는 목표가 생겼습니다.

Thank you