Char
Mes

NEW/CHOICE

HTML 기반 HIGHEND BRAND PRODUCTS를
Select shop concept으로 만든 website

BUSSINESS CONTACT
CHARMESOFFICIAL@KOREA.COM

Content

I

THE

I. Project 개요

MAINDIRECTOR / KIMBUMJIN
BACKENDDESIGNER / KWAKHEEWON
FRONTENDDESIGNER / PARKJAEHYUN
KEYDESIGNER / KIMSUIN

SINCE. 2025
@CHARMES_OFFICAL

I. Project 개요

I

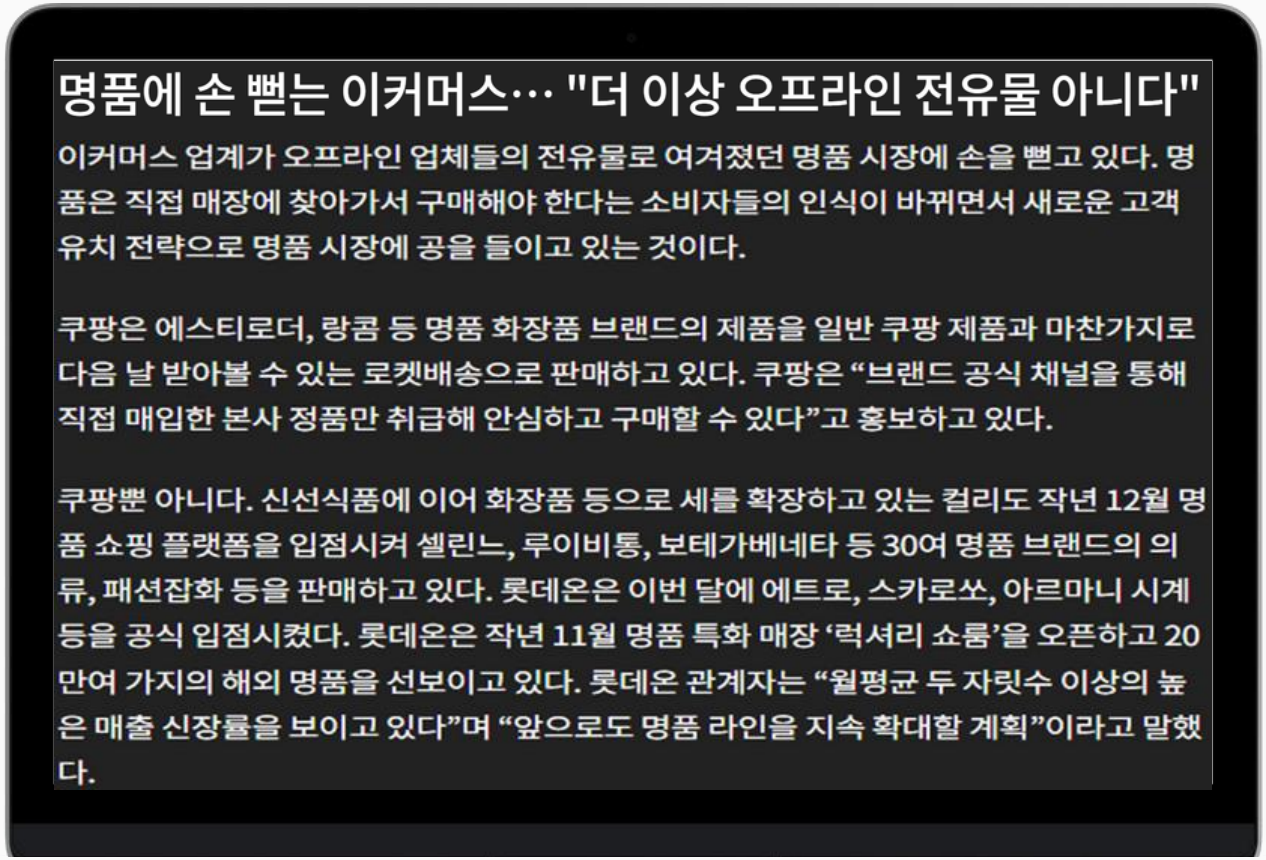
경제활동이라 함은 일반적으로 근로를 통해, 즉 회사를 다니는 근로자로서 근로임금으로 하여금 경제 소득을 창출하는 것이 대표적인 세대였으나, 4차 산업혁명으로 인해 IT 기술의 발전으로 신기술을 이용한 직업 혹은 직무가 생겨나고 있다. 특히 창업을 통해, 경제 소득을 창출하고 싶거나, 이미 하고 있는 새로운 세대가 늘어나고 있고, 청소년을 비롯해 2030세대들이 E커머스 시장에 뛰어들고 있는 추세이다. 그런데 컴퓨터 소프트웨어에 대한 지식이 전무한 일반적인 사람일 경우 웹사이트를 구축하여, 본인만의 웹사이트를 구축하기에는 어려움이 많다. 이를 대신하여 웹사이트를 관리할 수 있는 능력을 갖춘 사람을 고용하여 운영하는 경우가 많은데 본 웹사이트의 핵심적인 가치는 프레임워크를 이용하지 않고, 웹사이트를 구축해내어 웹사이트에 대한 지식이 출중하지 않더라도 누구나 쉽게 생성할 수 있는 웹사이트 형태를 구현해내고자 하는 키 포인트를 가지고 있다.

MZ세대도 눈독들이는 창업 아이템, 쇼핑몰 창업의 장단점

그 중에서도 쇼핑몰 창업은 인기 있는 선택지 중 하나이다. 쇼핑몰 창업은 비교적 낮은 초기 투자 비용과 온라인 시장의 성장으로 인해 확장 가능성이 크다는 장점이 있다. 또한, 개인의 취향과 관심사에 맞는 상품을 판매할 수 있어 자신의 취미나 관심 분야에 대한 사업을 진행할 수 있다는 점도 매력적이다.

I. Project 개요

I



명품에 손 뻗는 이커머스… "더 이상 오프라인 전유물 아니다"

이커머스 업계가 오프라인 업체들의 전유물로 여겨졌던 명품 시장에 손을 뻗고 있다. 명품은 직접 매장에 찾아가서 구매해야 한다는 소비자들의 인식이 바뀌면서 새로운 고객 유치 전략으로 명품 시장에 공을 들이고 있는 것이다.

쿠팡은 에스티로더, 랑콤 등 명품 화장품 브랜드의 제품을 일반 쿠팡 제품과 마찬가지로 다음 날 받아볼 수 있는 로켓배송으로 판매하고 있다. 쿠팡은 "브랜드 공식 채널을 통해 직접 매입한 본사 정품만 취급해 안심하고 구매할 수 있다"고 홍보하고 있다.

쿠팡뿐 아니다. 신선식품에 이어 화장품 등으로 세를 확장하고 있는 컬리도 작년 12월 명품 쇼핑 플랫폼을 입점시켜 셀린느, 루이비통, 보테가베네타 등 30여 명품 브랜드의 의류, 패션잡화 등을 판매하고 있다. 롯데온은 이번 달에 에트로, 스카로쏘, 아르마니 시계 등을 공식 입점시켰다. 롯데온은 작년 11월 명품 특화 매장 '럭셔리 쇼룸'을 오픈하고 20만여 가지의 해외 명품을 선보이고 있다. 롯데온 관계자는 "월평균 두 자릿수 이상의 높은 매출 신장률을 보이고 있다"며 "앞으로도 명품 라인을 지속 확대할 계획"이라고 말했다.

출처 : 명품에 손 뻗는 이커머스… "더 이상 오프라인 전유물 아니다" 석남준 기자, 조선경제
수정 2025.02.10. 16:06
https://www.chosun.com/economy/market_trend/2025/02/10/HWIRCYFE7JABRF3LF672YY5YY4/?utm_source=naver&utm_medium=referral&utm_campaign=naver-news

명품 쇼핑몰 특유의 고급스럽고 접근하기 어려운 분위기를 극복하고자, 우리는 더 친근하고 직관적인 멀티샵 형태를 도입했습니다. 멀티샵 웹사이트는 다양한 상품을 한눈에 확인할 수 있는 구조와 함께, 사용자가 편리하게 탐색할 수 있는 장점을 제공합니다.
이를 통해 명품의 고유한 가치를 유지하면서도, 고객들에게 더 친숙하고 접근 가능한 쇼핑 환경을 제공하고자 했습니다.
결과적으로, 쇼핑몰 이용자들은 원하는 상품을 쉽게 찾을 수 있으며, 직관적인 사용자 경험을 통해 보다 만족스러운 쇼핑을 경험할 수 있도록 계획하였습니다.

I

THE

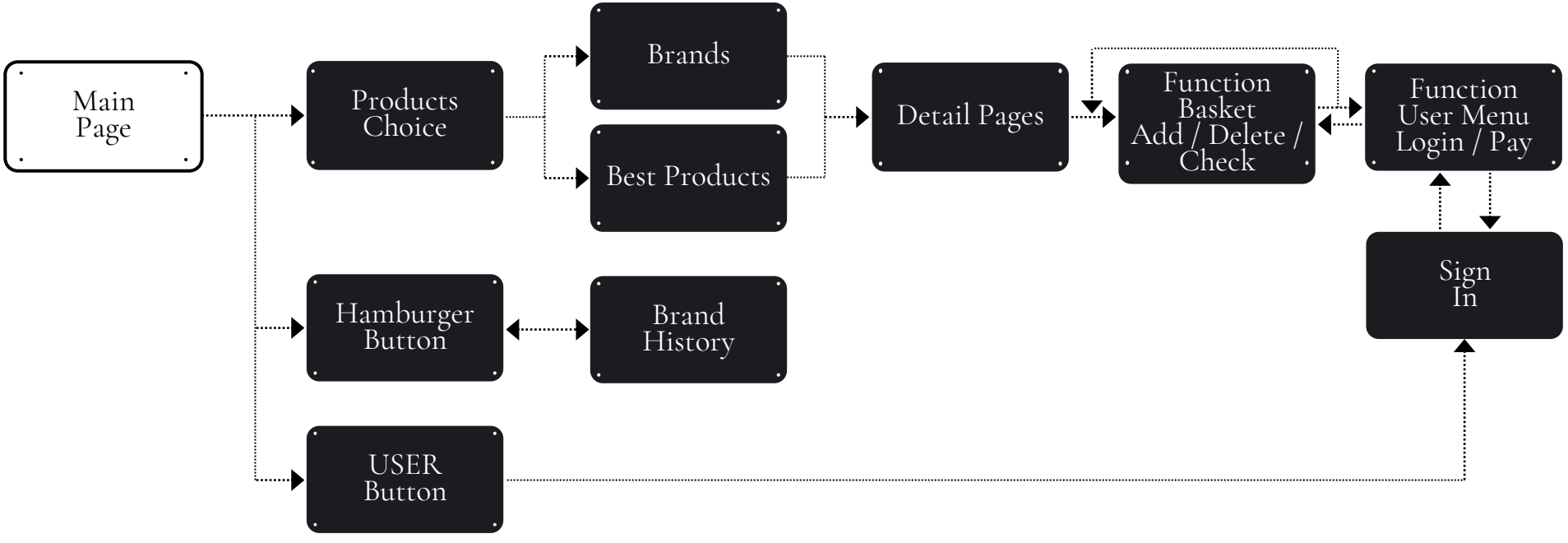
II. UserView Flowchart

MAINDIRECTOR / KIMBUMJIN
BACKENDDESIGNER / KWAKHEEWON
FRONTENDDESIGNER / PARKJAEHYUN
KEYDESIGNER / KIMSUIN

SINCE. 2025
@CHARMES_OFFICAL

II.User View flowchart

II



I

THE

III. Project 역할

MAINDIRECTOR / KIMBUMJIN
BACKENDDESIGNER / KWAKHEEWON
FRONTENDDESIGNER / PARKJAEHYUN
KEYDESIGNER / KIMSUIN

SINCE. 2025
@CHARMES_OFFICAL

UI/UX 디자인

- localStorage에서 불러온 제품 데이터를 JSON 형식으로 변환 후 활용
- iframe을 활용하여 제품 상세 페이지와 장바구니, 로그인 페이지 간 연동 작업 수행
- 메인 홈페이지 및 제품 상세 페이지 간 href를 활용하여 원활한 이동 구현
- 외부 CSS를 import하여 장바구니 스타일링
- table 태그를 활용하여 장바구니 레이아웃을 각각 제품명, 가격, 수량, 합계, 삭제 버튼 등으로 구성
- 결제 버튼, 홈으로 이동 버튼을 시각적으로 강조하는 방향으로 스타일링 후 hover, active 효과 적용하여 UI 및 UX 개선
- increaseButton, decreaseButton의 크기 및 아이콘 활용하여 조작 UI를 직관적으로 구현
- div, form을 활용하여 로그인 UI 레이아웃 구성
- login.css를 활용하여 스타일링 및 화면 중앙 정렬
- text-align. position 속성을 활용하여 로그인 폼, 로그아웃 UI 배치 조정

기능 개발 및 시스템 구현

- localStorage에 저장된 product.json 데이터를 JSON.parse()로 변환하여 cart 변수에 저장
- JSON.stringfy()를 활용하여 상세 페이지에서 추가된 제품 데이터를 cart에 업데이트
- table을 활용하여 장바구니 내 저장된 제품 정보가 테이블에 동적으로 생성되어 삽입되도록 표시 (forEach())
- increaseButton, decreaseButton, onclick 이벤트를 이용하여 장바구니 내 수량 증가/감소 기능 구현
- button을 활용하여 결제 버튼, 홈으로 이동 버튼을 클릭 시 특정 기능 수행하도록 onclick 이벤트 핸들링
- div, form을 활용하여 로그인 폼 구현
- input 필드 및 submit 버튼 추가하여 로그인 기능 구현
- window.location.href를 활용하여 장바구니, 상세 페이지, 메인 홈페이지, 로그인 홈페이지 간의 연동 구현

리서치 및 최적화

- localStorage 활용 시 try-catch 문을 적용하여 JSON 파싱 오류 방지
- JSON.stringfy()로 데이터 저장 시 불필요한 데이터가 포함되지 않도록 필수 데이터만 선택적으로 저장
- window.location.href를 활용한 페이지 이동 시 query string를 활용하여 데이터 유지
- table 데이터 렌더링 시 forEach() 대신 map()과 innerHTML = ...을 최소화하여 DOM 조작 최적화
- localStorage에서 데이터를 불러오고 저장하는 로직을 함수로 분리하여 재사용성 증가
- 장바구니 내 제품 수량 변경 시 setTimeout()을 활용하여 연속 클릭에 대한 이벤트 중복 처리 방지
- 로그인 성공 시 localStorage에 user 정보를 저장하고 페이지 전환 후에도 유지하도록 처리

UI/UX 디자인

- style.display 속성을 활용하여 로그인, 로그아웃, 회원가입 폼을 동적으로 제어하여 불필요한 화면 출력 최소화
- CSS transition 속성을 활용하여 opacity 및 transform을 조정해 자연스러운 효과 적용
- visibility: hidden/visible을 활용하여 상세 페이지 모달창 구현
- 부트스트랩을 이용한 반응형 버튼 구현
- 사용자의 이탈률을 줄이기 위한 애니메이션, 로딩 최적화 기법 조사 및 적용
- 주요 트렌드 조사 및 적용(무한 스크롤 방식 vs 페이지네이션, 이미지 최적화 기법 등)

기능 개발 및 시스템 구현

- fetch() API를 사용하여 외부 JSON 데이터를 비동기적으로 불러와 제품 정보를 상세 페이지에 전달
- slice() 메서드를 활용하여 페이지네이션 방식으로 일정 개수의 제품만 표시하여 성능 최적화
- onclick 이벤트와 increase(), decrease() 함수를 이용하여 quantity 값 증감을 통한 상품 개수 처리
- filter() 메서드를 활용하여 회원 삭제 기능 구현 (관리자 기능)
- localStorage.getItem('currentUser')을 이용하여 현재 로그인한 사용자의 상태 확인
- 로그인 상태에서만 특정 기능(장바구니 추가, 결제 등)을 사용할 수 있도록 활성화/비활성화 처리
- JSON.parse(localStorage.getItem('cart'))을 활용하여 저장된 장바구니 정보 불러오기 및 금액 계산
- localStorage.removeItem('cart')을 이용하여 결제 완료 후 장바구니 데이터 삭제

리서치 및 최적화

- JSON 형식의 제품 데이터 수집 및 정리 (가격, 재고, 설명, 이미지 등)
- fetch() API를 이용한 데이터 교환 방법 연구 및 CORS 정책 이해
- 타사 e-commerce 플랫폼(Amazon, 쿠팡, 11번가 등)의 상품 카테고리 및 UI/UX 구조 분석
- 결제 시스템 및 보안 관련 기술 조사
- div를 활용한 반응형 디자인 패턴 분석
- 데이터 저장 및 관리 방식 연구
- DB를 사용하지 않은 localStorage, sessionStorage, IndexedDB를 활용한 데이터 저장 방식 비교
- JSON 데이터 구조화 및 데이터 직렬화/역직렬화(JSON.stringify(), JSON.parse()) 활용 방법 조사
- 브라우저 기반 스토리지의 보안 및 데이터 유지 기간 조사
- localStorage와 sessionStorage를 활용한 임시 데이터 저장 방식과 차이점 비교

UI/UX 디자인 및 프론트엔드 개발

- iframe을 활용하여 상단 네비게이션 바 분리 및 유지
- 사용자 경험(UX)을 고려한 상단 네비게이션 바 최적화
- 주요 콘텐츠 배치와 상호작용 기획 및 실행
- CSS를 사용한 메인 페이지의 반응형 웹 디자인 구축
- CSS hover 속성을 활용하여 카테고리 드롭다운 기능 구현
- transform 및 transition을 통해 부드러운 애니메이션 효과 적용
- CSS를 활용하여 메인 페이지 슬라이드 바 기능 구축
- UI/UX 트렌드와 접근성을 고려한 디자인 가이드 분석
- 웹사이트 초기 디자인 프로토타입 제작
- 브랜드 정체성을 반영한 디자인 및 구현진행
- CSS와 JavaScript를 결합하여 동적효과 구현

기능 개발 및 시스템 구현

- HTML 링크를 활용하여 사용자가 현재 페이지에서 다른 웹사이트나 페이지로 이동하는 기능 구현
- JavaScript의 try-catch 구문을 활용한 오류 감지 및 예외 처리
- 햄버거 버튼을 활용하여 화면에 숨겨진 div 요소가 나타나고, 이를 X 버튼으로 닫을 수 있는 기능을 구현
- JavaScript를 활용하여 사용자가 페이지를 스크롤할 때, 이미지와 로고가 화면에 자연스럽게 등장하고 사라지는 애니메이션 효과 적용
- JavaScript를 활용하여 사용자가 페이지를 스크롤한 후, 클릭을 통해 페이지의 최상단으로 부드럽게 돌아가는 기능 구현

리서치 및 최적화

- CSS 애니메이션 및 인터랙션 관련 오픈소스 레퍼런스 조사
- CANVA를 활용한 웹사이트 동작 구현 소개를 위한 flowchart제작
- 사이트 주요 참조 래퍼런스 및 자료 조사
- 외부 오픈소스에 불필요한 스타일 제거 및 최적화를 통한 페이지 로딩 속도 향상
- 팀원간의 코드 결합으로 인한 중복된 스타일 속성 통합 및 삭제를 통한 코드의 가독성과 효율성 향상
- 오픈소스 UI 컴포넌트 라이브러리 조사
- 반응형 웹 구현을 위한 CSS 라이브러리 분석
- 레이어 및 텍스트 상호작용 최적화를 통해 일관성 있고 직관적인 사용자 경험 최적화

UI/UX 디자인 및 웹프로젝트 진행 계획

- 스크롤 애니메이션, 트랜지션 효과 적용 가능성 연구
- 웹사이트 속도 최적화를 고려한 css및 javaScript 애니메이션 구현 전략 수립
- 그리드 시스템 및 미디어 쿼리 활용 방안 연구
- 브랜드 아이덴티티와 연관된 디자인 시스템 구축
- 사용자 시각적 집중도를 고려한 레이아웃 및 타이포그래피 설계
- 버튼, 네비게이션 등 UI요소 스타일 가이드 제작
- UX플로우 고려한 주요 콘텐츠 배치 전략
- 스크롤, 호버를 활용한 인터랙션 디자인 기획

기능 개발 및 시스템 구현

- 브랜드 컬러, 폰트, 로고 스타일 가이드 제작
- 브랜드 감성을 반영한 버튼, 아이콘 등 UI요소 디자인
- 브랜드 컨셉 강조 전략 계획 수립 및 영상 스타일 구성
- 웹사이트 스토리보드 기획
- 유사 브랜드 및 경쟁사 콘텐츠 전략 분석
- 브랜드 핵심가치 도출을 위한 디자인 콘텐츠 아이디어 도출
- SNS 및 디지털 마케팅을 위한 비주얼 콘텐츠 기획
- 사용자 페르소나 분석 및 타겟 고객 소비 시각 분석
- 데이터 기반의 UX설계를 위한 레퍼런스 리서치 진행

프로젝트 기획 및 기술구현

- 오픈소스 UI 템플릿 활용 가능성 분석
- Canva 활용한 웹사이트 디자인 및 기능 소개 관련 기획 문서 및 발표 자료 제작
- 디자인 컨셉 및 UI전략을 시각적으로 표현하는 슬라이드 구성
- HTML id와 href=#section을 활용해 HTML 내부 링크 이동 구현

I

THE

IV. Project
작업내용

MAINDIRECTOR / KIMBUMJIN
BACKENDDESIGNER / KWAKHEEWON
FRONTENDDESIGNER / PARKJAEHYUN
KEYDESIGNER / KIMSUIN

SINCE. 2025
@CHARMES_OFFICAL

```
var lastScrollTop = 0;
var delta = 0;
locoScroll.on('scroll', (position) => {

    function productsViewThumb() {
        if (!$('#main').hasClass('products_view_wrap')) return;
        var scrollDistance = position.scroll.y;
        $('.product_gallery_big .img').each(function (i) {
            if ($(this).position().top < (position.scroll.y)) {
                $('.product_gallery_nav ul li').removeClass('active');
                $('.product_gallery_nav ul li').eq(i).addClass('active');
            }
        });
    }

    productsViewThumb();

    if ((position.scroll.y) > delta) {
        document.querySelector('body').classList.add('active');
        document.querySelector('header').classList.add('active');
    } else {
        document.querySelector('body').classList.remove('active');
        document.querySelector('header').classList.remove('active');
    }

    var scrollT = position.scroll.y;
    if (Math.abs(lastScrollTop - scrollT) <= delta)
        return;
    if ((scrollT > lastScrollTop) && (lastScrollTop > 90)) {
        $('header').addClass('hide');

    } else {
        $('header').removeClass('hide');

    }
    lastScrollTop = scrollT;
    if ($('header:not(.fix_style)').hasClass('hide')) {
        $('header nav').fadeOut(200);
    } else {
        $('header nav').fadeIn(300);
    }
});
```

Smooth Viewing
JavaScript

locoScroll.on(‘scroll, (position) -> { … } **scroll의 위치를 받아온 뒤에**

$(‘ .product_gallery_big .img’) .each(function(i)) { … } **상품의 scroll 위치에 도달하면**

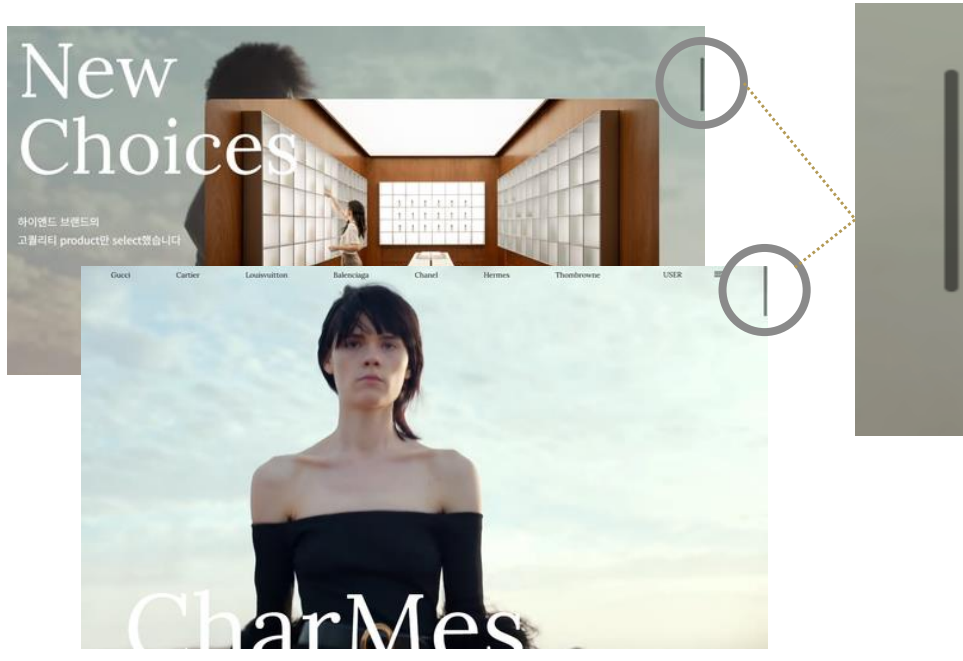
document.querySelector(‘ … ’)classList.add(‘active) **active클래스를 추가하여**

$(‘header nav’).fadeIn(300),

$(‘header nav’).fadeout(200)

fadein, fadeout 효과를 추가하였습니다

```css
/* 스크롤바 주요 서식 */
.c-scrollbar {
  position: absolute;
  right: 0;
  top: 0;
  width: 11px;
  height: 100%;
  transform-origin: center right;
  transition: transform 0.3s, opacity 0.3s;
  opacity: 0; }
  .c-scrollbar:hover {
    transform: scaleX(1.45); }
  .c-scrollbar:hover, .has-scroll-scrolling .c-scrollbar, .has-scroll-dragging .c-scrollbar {
    opacity: 1; }
  [data-scroll-direction="horizontal"] .c-scrollbar {
    width: 100%;
    height: 10px;
    top: auto;
    bottom: 0;
    transform: scaleY(1); }
    [data-scroll-direction="horizontal"] .c-scrollbar:hover {
      transform: scaleY(1.3); }

.c-scrollbar_thumb {
  position: absolute;
  top: 0;
  right: 0;
  background-color: black;
  opacity: 0.5;
  width: 6px;
  border-radius: 10px;
  margin: 2px;
  cursor: -webkit-grab;
  cursor: grab;
  }
  .has-scroll-dragging .c-scrollbar_thumb {
    cursor: -webkit-grabbing;
    cursor: grabbing; }
  [data-scroll-direction="horizontal"] .c-scrollbar_thumb {
    right: auto;
    bottom: 0; }
    /* /스크롤바 주요 서식 */
```



Custom Scroll Bar

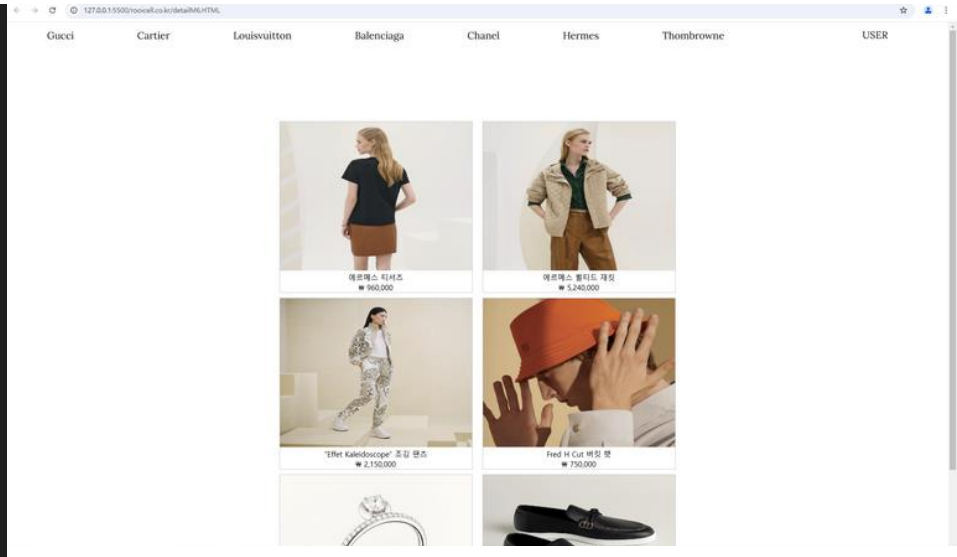
CSS를 이용하여 **.c-scrollbar 스타일을 생성**

ScrollBar : hover시 **Scroll 확대 효과 적용**

마우스 스크롤이 아닌 스크롤을 직접 잡아 당겼을 때

transform을 사용해 부드러운 이동 효과 적용

```
<script>
    try {
        cart = JSON.parse(localStorage.getItem('cart') || '[]');
    } catch (e) {
        console.error('Parsing error:', e);
        cart = [];
    }

    const maxProducts = 6; // 한 페이지에 최대 6개 표시
    const desiredBrand = "Hermes"

    fetch('products.json')
        .then(response => response.json())
        .then(products => {
            const productList = document.getElementById('product-list');
            const modal = document.getElementById('modal');
            const closeModal = document.getElementById('close');

            // 기존 데이터 제거 후 6개까지만 표시
            productList.innerHTML = '';

            // 특정 브랜드의 제품만 필터링
            const filteredProducts = products.filter(product => product.brand === desiredBrand);
            // 필터링된 제품 중 최대 maxProducts 개수만 표시
            filteredProducts.slice(0, maxProducts).forEach(product => {
                const productDiv = document.createElement('div');
                productDiv.className = 'product';
                productDiv.innerHTML = `
            <img src="${product.imageUrl}" alt="${product.name}"
            style="width: 100%; height: 300px; object-fit: fill;">
            <p style="font-size: 15px; margin:0px">
            ${product.name}
            </p>
            <p style="font-size: 15px; margin:0px">
            ${product.price}
            </p>
            `;
```



Detail Page

localStorage.getItem('cart') -> **저장된 장바구니 데이터를 불러옴**

const maxProducts = 6; -> **한 페이지에 최대 6개의 상품 출력**

const desiredBrand = "Hermes" -> **상품 브랜드 변수 저장**

fetch('products.json')

products.json에서 상품 정보 불러옴

.then(response => response.json())

.then(products => {

데이터를 json형식으로 변환하고 products 배열에 저장

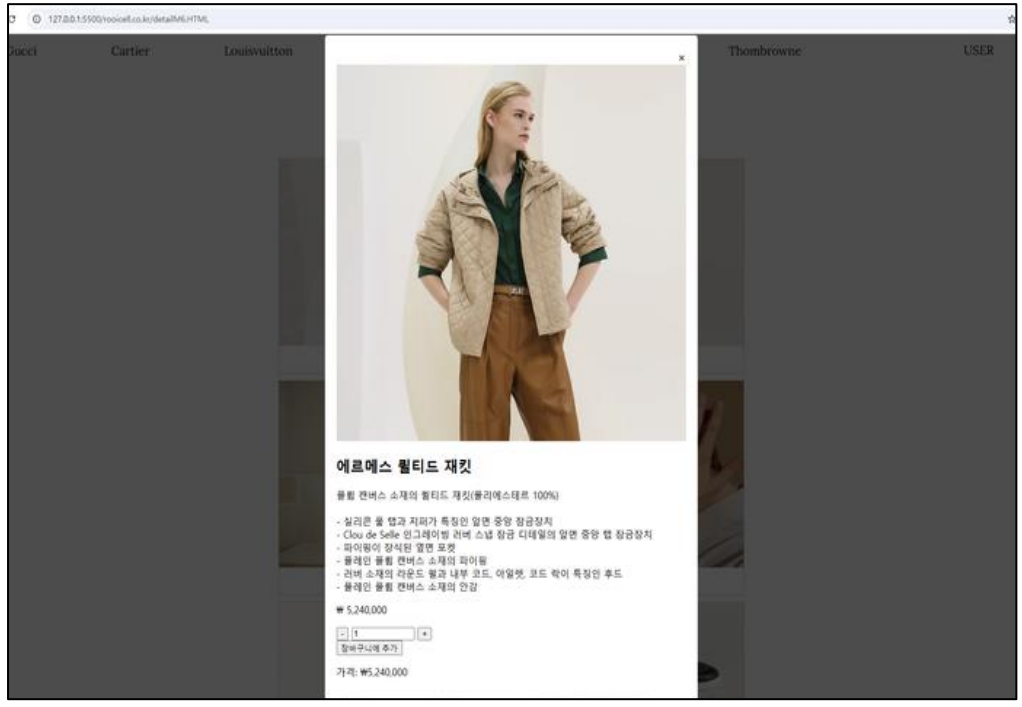
const filteredProducts -> product.brand === desiredBrand

filter() 함수를 사용하여 지정한 브랜드 제품만 필터링

forEach(product)

필터링 후 상품을 각각 처리

```
// 제품 클릭 시 모달창 열기
productDiv.onclick = () => {
    document.getElementById('modal-product-name').innerText = product.name;
    document.getElementById('modal-product-description').innerText = product.description;
    document.getElementById('modal-product-price').innerText = product.price;
    document.getElementById('modal-product-image').src = product.imageUrl;
    document.getElementById('modal-quantity-input').value = 0;
    modal.classList.add('show');
};
productList.appendChild(productDiv);
    });
})
    .catch(error => console.error('문제가 발생했습니다:', error));

// 모달창 닫기 기능
    document.getElementById('close').onclick = () => {
    document.getElementById('modal').classList.remove('show');
    };
    window.onclick = (event) => {
        if (event.target === document.getElementById('modal')) {
            document.getElementById('modal').classList.remove('show');
        }
    };
    function calculateTotal() {
    const quantity = parseInt(document.getElementById('modal-quantity-input').value, 10);
    const priceText = document.getElementById('modal-product-price').innerText;

// ₩와 쉼표 제거하고 숫자로 변환
    const productPrice = parseFloat(priceText.replace(/[^0-9.-]+/g, ""));

    if (isNaN(productPrice)) {
        document.getElementById('modal-total-amount').innerText = '가격: 0원';
        return;
    }

    const total = productPrice * quantity;
    const formattedTotal = new Intl.NumberFormat('ko-KR', { style: 'currency', currency: 'KRW' }).format(total);
    document.getElementById('modal-total-amount').innerText = `가격: ${formattedTotal}`;
    }
```




```
    // 수량 증가 버튼 클릭 이벤트
    document.getElementById('modal-increase-button').onclick = function () {
    const quantityInput = document.getElementById('modal-quantity-input');
    quantityInput.value = parseInt(quantityInput.value) + 1;
    calculateTotal();
    };

    // 수량 감소 버튼 클릭 이벤트
    document.getElementById('modal-decrease-button').onclick = function () {
    const quantityInput = document.getElementById('modal-quantity-input');
    if (parseInt(quantityInput.value) > 0) {
    quantityInput.value = parseInt(quantityInput.value) - 1;
    calculateTotal();
        }
    };

    // 주문하기 버튼 클릭 이벤트
    document.getElementById('modal-order-button').onclick = function () {
    const quantity = parseInt(document.getElementById('modal-quantity-input').value);
    const productName = document.getElementById('modal-product-name').innerText;
    const productPrice = parseFloat(document.getElementById('modal-product-price').innerText.replace(/[^0-9.-]+/g, "")); // 상품 가격 가져오기

    // 수량이 0이면 장바구니로 추가되지 않도록 처리
    if (quantity === 0) {
        alert('수량을 선택해주세요.');
        return;
    }
    // 동일 상품 존재 확인 후 업데이트
    const existingItem = cart.find(item => item.name === productName);
    if (existingItem) {
        existingItem.quantity += quantity; // 수량 누적
    } else {
        cart.push({ id: Date.now(), name: productName, price: productPrice, quantity: quantity });
    }

    // 로컬 스토리지 업데이트
    localStorage.setItem('cart', JSON.stringify(cart));
    window.location.href = `basket.html?cart=${JSON.stringify(cart)}`
    alert('장바구니에 추가되었습니다.');
    modal.classList.remove('show'); // 모달 닫기
    };
</script>
```

Detail Page(Modal)

productDiv.onclick = ()

사용자가 제품을 클릭하면 onclick 이벤트 발생 -> 모달창 열림

document.getElementByID('').innereText = ''

모달창을 **선택한 상품에 대한 정보로 설정**

quantityInput.value = parseInt(quantityInput.value)

수량 증감 버튼으로 제품 수량 정의

if (quantity === 0) -> return;

수량이 0이면 장바구니로 추가되지 않음

if (existingItem) -> existingItem.quantity += quantity;

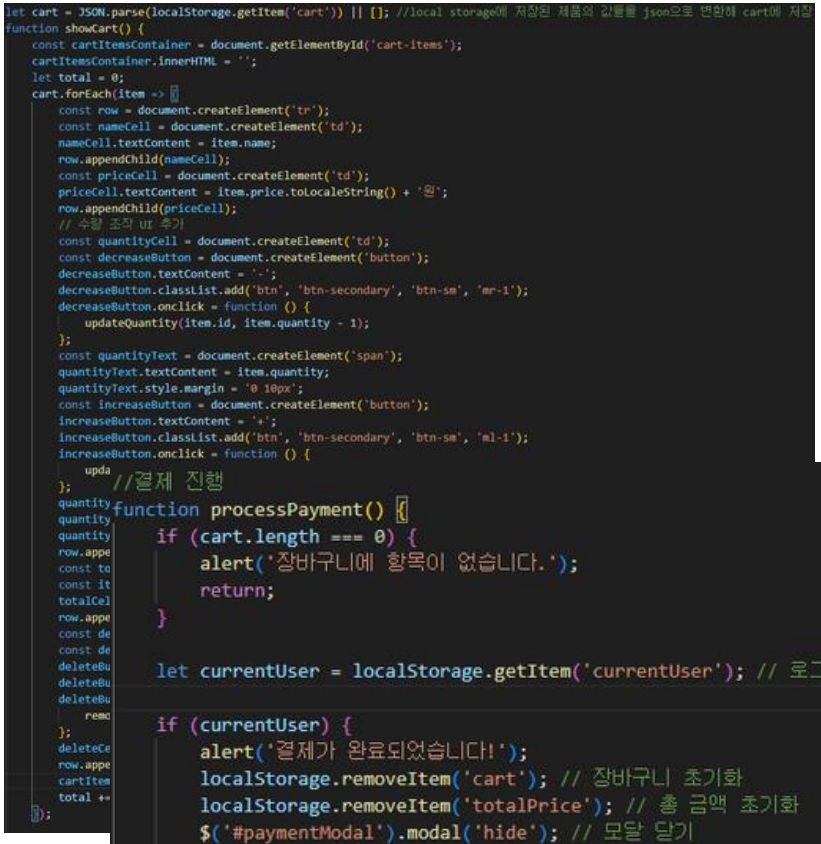
같은 제품이 이미 있으면 수량 누적

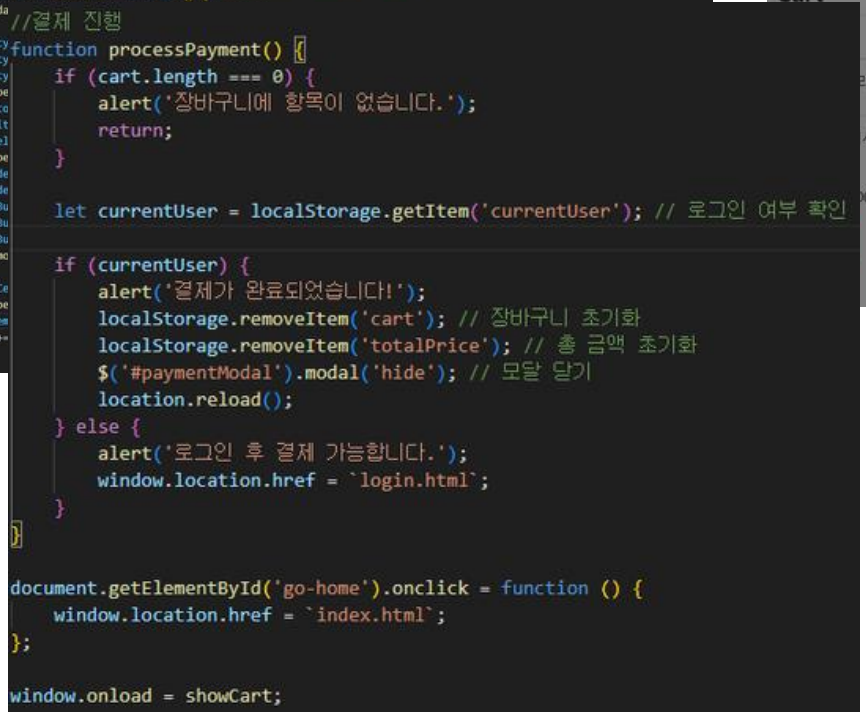
else -> cart.push

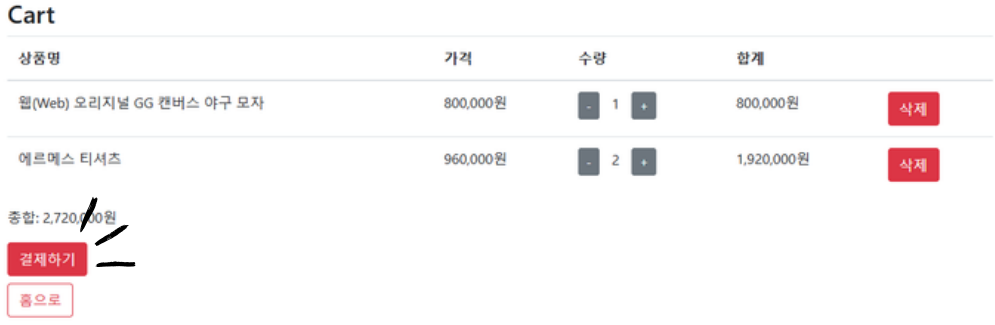
장바구니로 상품 정보를 push

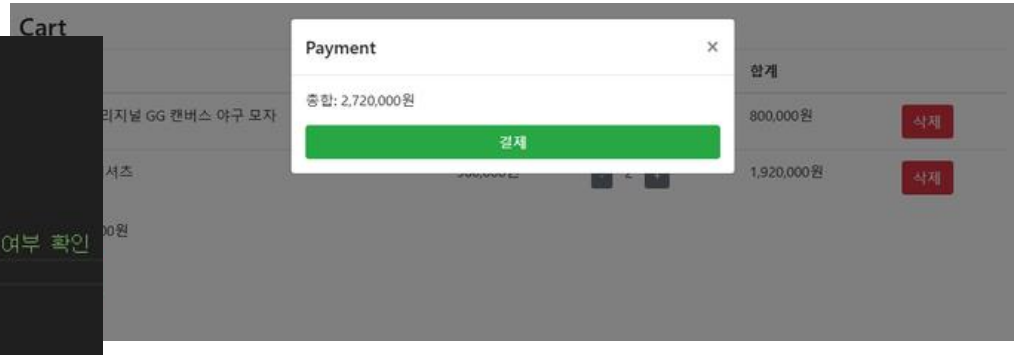
localStorage.setItem('cart', JSON.stringify(cart));

장바구니 데이터를 localStorage에 전달









Cart

cart.forEach(item =>

cart 배열의 각 제품(item)을 반복하면서 테이블 생성

let currentUser = localStorage.getItem('currentUser');

로그인 여부 확인

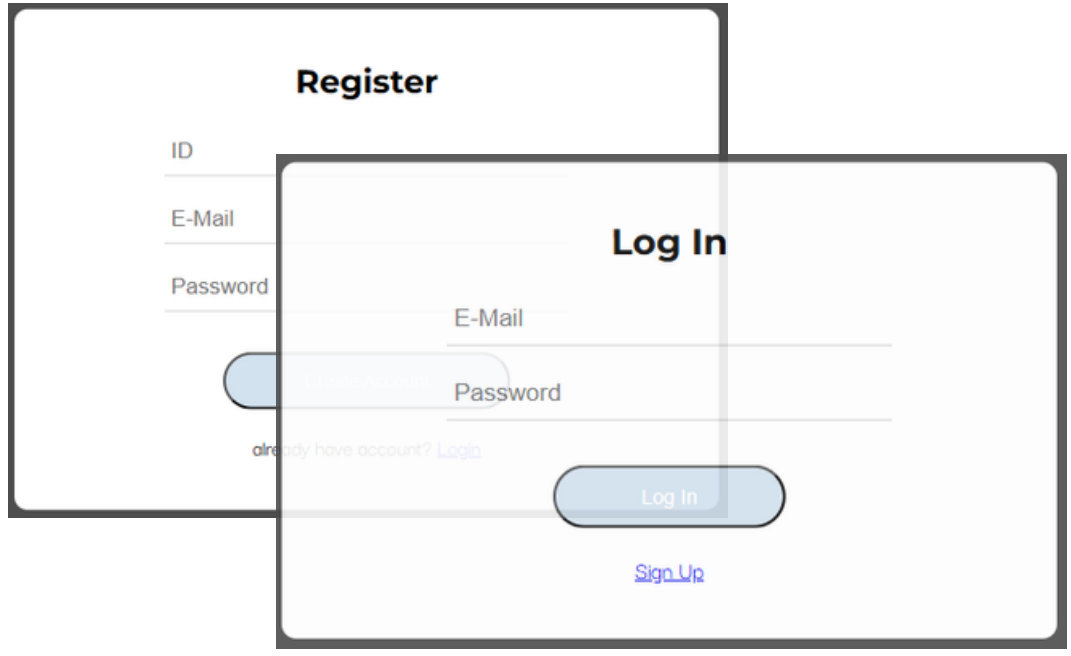
if (currentUser) => localStorage.removeItem();

확인 후 결제 완료 메세지 출력, 장바구니 및 금액 초기화

else => window.location.href = `login.html`;

로그인 상태가 아니라면 'login.html' 페이지로 보냄

```js
// 회원가입 폼 제출
document.getElementById('signup-form').addEventListener('submit', function (e) {
    e.preventDefault();
    const username = document.getElementById('signup-username').value.trim();
    const email = document.getElementById('signup-email').value.trim();
    const password = document.getElementById('signup-password').value;

    let users = JSON.parse(localStorage.getItem('users')) || [];

    // 중복 이메일 체크
    if (users.some(user => user.email === email)) {
        alert('이미 등록된 이메일입니다.');
        return;
    }

    // 회원 추가
    users.push({ username, email, password });
    localStorage.setItem('users', JSON.stringify(users));
    alert('회원가입 성공!');
    showLoginSection();
});

// 로그인 폼 제출
document.getElementById('login-form').addEventListener('submit', function (e) {
    e.preventDefault();
    const email = document.getElementById('login-email').value.trim();
    const password = document.getElementById('login-password').value;

    const users = JSON.parse(localStorage.getItem('users')) || [];
    const user = users.find(user => user.email === email && user.password === password);

    if (user) {
        alert(`환영합니다, ${user.username}!`);
        localStorage.setItem('currentUser', JSON.stringify(user));
        // 로그인 성공 시 detailM1.html로 이동
        window.location.href = 'index.html';
    } else {
        alert('이메일 또는 비밀번호가 올바르지 않습니다.');
    }
});
```



Register & Login

if (users.some(user => user.email === email)

중복 이메일 확인

users.push({ username, email, password });

새로운 회원 정보 users 배열에 추가

localStorage.setItem('users', JSON.stringify(users));

users 배열을 문자열로 변환하여 localStorage에 저장

const user = users.find(user => user.email === email && user.password === password);

find 매서드로 email과 password가 일치하는 회원 검색 후 있다면 로그인

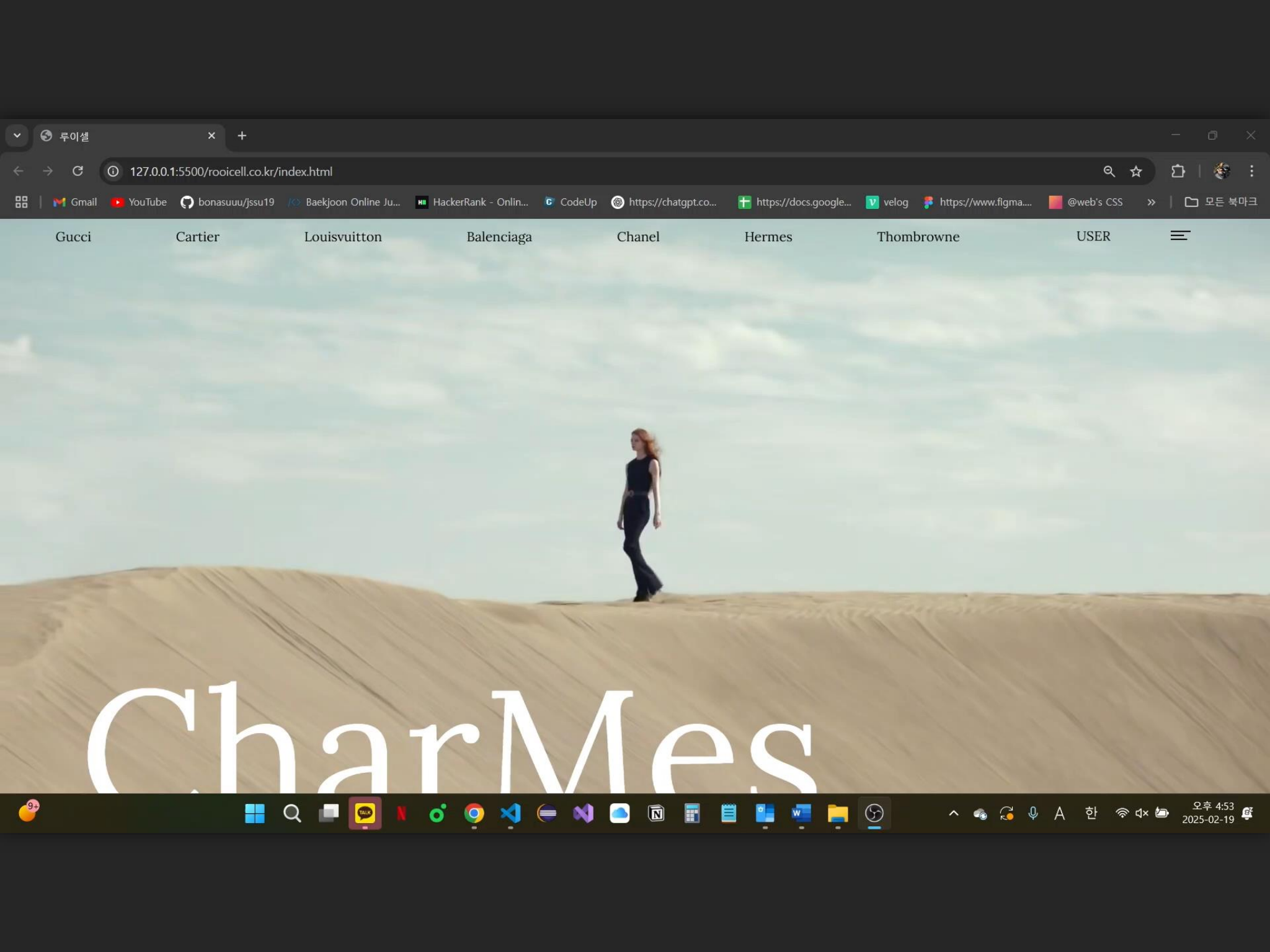
I

THE

V. Project구동영상

MAINDIRECTOR / KIMBUMJIN
BACKENDDESIGNER / KWAKHEEWON
FRONTENDDESIGNER / PARKJAEHYUN
KEYDESIGNER / KIMSUIN

SINCE. 2025
@CHARMES_OFFICAL


루이셀
127.0.0.1:5500/rooicell.co.kr/index.html
Gmail
YouTube
bonasuuu/jssu19
Baekjoon Online Ju...
HackerRank - Onlin...
CodeUp
https://chatgpt.co...
https://docs.google...
velog
https://www.figma....
@web's CSS
모든 북마크
Gucci
Cartier
Louisvuitton
Balenciaga
Chanel
Hermes
Thombrowne
USER
CharMes
오후 4:53
2025-02-19

I

THE

VI. Project자체평가

MAINDIRECTOR / KIMBUMJIN
BACKENDDESIGNER / KWAKHEEWON
FRONTENDDESIGNER / PARKJAEHYUN
KEYDESIGNER / KIMSUIN

SINCE. 2025
@CHARMES_OFFICAL

자체평가 - 김범진

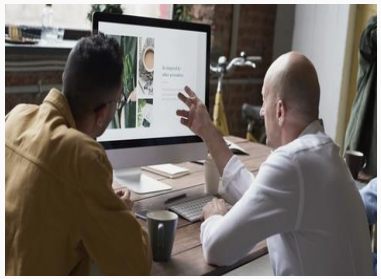
본 프로젝트는 프레임워크의 사용없이 정적인 HTML 페이지 기반으로 개발을 진행하였습니다.
프레임워크의 사용이 가능했다면 컴포넌트 기반의 동적 랜더링을 활용하여 사용자의
상호작용에 맞게 페이지 내부에 특정 요소만 변경하는 형식으로 UI를 구현할 수 있었으나,

이러한 기능의 사용이 불가능한 상태에서 프로젝트를 진행하게 되었으며 반면에 사전에 정의된
개별 HTML 페이지를 load하여 전환하는 방식으로 홈페이지 개발을 진행하게 되었습니다.
이러한 방식을 활용하기 위해 본 프로젝트에서는 제품의 상세 정보를 저장하는 상세 페이지와
장바구니 페이지 등의 사전 구현이 필수적으로 이루어졌습니다.

데이터 관리 부분은 DB 대신 JSON을 기반한 데이터 관리 방식을 채택하였기에
실시간 데이터 변경 및 저장 용량의 제한 등과 같은 문제점에 노출되었습니다.
DB를 활용할 시 index 기능을 이용한 검색 최적화 및 고급 Query 기능 구현이 가능했던 부분이나,
DB 대신 JSON을 활용하면서 효율적인 검색 기능을 구현하는데 한계가 있었습니다.

또한, 브라우저 종료 후에도 데이터 저장 값의 보존을 위해 local Storage 기능을 활용하였으나,
chrome 자체의 보안 정책으로 인해 storage의 키 값 초기화 작업이 정상적으로 이루어지지 않아
데이터 동기화 및 저장값 관리에 어려움을 겪었습니다

위와 같은 문제점들에서 보았듯이 프레임워크의 활용 없이 HTML 페이지 기반으로 프로젝트를
진행하는 상황으로 인해 상기한 문제점들과 봉착하면서 본래 구현하고자 했던 UI 및 기능 구현과
데이터 저장값의 유지가 이루어지지 않았습니다. 추후 프레임워크에 대한 추가적인 학습을 통해
이번 프로젝트에서 겪은 데이터 관리 및 서버 정책 기반의 부분 동적 랜더링과 같은 기능들을
활용하면서 이번 프로젝트 진행 중에 겪은 문제점의 보완을 시도해보고 싶습니다.



자체평가 - 곽희원

본 프로젝트는 프레임워크 없이 정적인 HTML 페이지를 기반으로 개발되었습니다.
이로 인해 동적 렌더링 및 반응형 UI 구현에 제약이 있었으며, 특히 컴포넌트 기반의 동적 랜더링을
활용할 수 없었습니다. 이를 보안하기 위해 정의된 개별 HTML 페이지를 전환하는 방식으로 구현하였고,
상세 페이지 이동 시 기존 페이지가 새롭게 로드되는 방식으로 처리하였습니다.

다양한 디바이스에서 원활한 작동을 보장하기 위한 최적화 작업에서 성능과 기능 간의 균형을 맞추는 것에 있어
어려움이 있었으며, 그로 인해 일부 기능을 구현하지 못하거나 성능 저하를 우려해 기능을
제한하는 상황이 발생했습니다.

프레임워크를 사용하지 않는 프로젝트였기에 가볍고, 처리하기 쉬운 데이터 형식을 필요로 하여
JSON을 선택하였습니다. 이는 프로그래밍 언어 및 소프트웨어 시스템을 광범위하게 지원하는
유연한 데이터 교환 형식으로서, 구조화된 데이터를 직관적으로 처리할 수 있고, 별도의 복잡한 설정 없이
local storage같은 환경에서 데이터를 쉽게 저장하고, 불러올 수 있으므로 개발에서 유용하게 활용하였습니다.

로그인과 장바구니 기능에서 브라우저 종료 후에도 데이터를 유지하기 위해 세션 대신 local storage를
사용하였으나, chrome의 보안 정책으로 인해 local storage에서 지정한 키 값을 명시적으로 초기화하는 기능이
적용되지 않아 아쉬움이 남았습니다.
프레임워크를 활용하는 부분이 가능하였다면 local storage를 직접 조작할 필요 없이 상태 관리 기능을 통해
데이터 관리 및 페이지 연동을 훨씬 효율적으로 처리할 수 있었을 것입니다. 또한 chrome의 보안 정책에 의한
제한 사항이 일부 해결 되었을 것입니다.

마지막으로 local storage의 경우, 저장 용량 제한이 있어 대량의 데이터를 요구하는 서비스,
예를 들어 쇼핑몰 서비스에서는 적용에 한계가 있었습니다.
향후 더 많은 용량의 상품 정보가 추가된 데이터를 처리할 경우 데이터베이스 연동 및 클라우드 스토리지
활용 방안을 고려할 필요가 있습니다.

자체평가 - 김수인

본 프로젝트는 프레임워크 없이 정적인 HTML 페이지를 기반으로 개발되었으며, 이로 인해 동적 랜더링과 반응형 UI 구현에 어려움이 있었습니다. 특히, 컴포넌트 기반의 개발이 불가능하여 UI 요소가 중복되고, 현재 제작한 웹사이트를 기반으로 실제 웹사이트를 제작한다고 가정했을 때 유지보수 비용이 증가할 것이며, 변경 사항이 발생할 때마다 여러 개의 HTML 파일을 일일이 수정해야 하는 비효율적인 구조를 가질 수밖에 없었습니다.

또한, JSON은 정적 파일이므로, 서버에서 데이터를 실시간으로 요청하여 가져오는 것이 불가능하여 로그인 세션을 유지하거나 사용자 맞춤 데이터를 제공하는 기능을 구현하기 어려웠고, 데이터베이스와 연동하여 실시간으로 데이터를 갱신하는 작업도 불가능하여 사용자 요청에 따라 변하는 데이터를 효과적으로 관리할 수 없었습니다.

이러한 한계를 보완하기 위해 개별 HTML 페이지를 생성하고 페이지 이동 시마다 새로 로드하는 방식으로 구현하였으나, 이 과정에서 페이지가 전환될 때마다 새로고침이 발생하여 사용자 경험이 저하되었고, 동일한 UI 요소가 여러 HTML 파일에 중복되어 있어 유지보수가 어렵고 비효율적인 문제가 발생했으며, JavaScript를 활용하여 일부 동적인 기능을 추가하려 했으나, 기존의 HTML과 CSS로 이미 구현된 부분을 JavaScript로 다시 처리해야 하는 상황이 발생했습니다.

또한, CSS 적용 과정에서도 오픈소스 코드에서 body나 *와 같은 전역 스타일을 정의하고 있는 경우 기존 CSS와 충돌이 발생하여 특정 컴포넌트에서만 적용하려던 스타일이 전체 페이지에 영향을 미치는 문제가 발생했으며, 다양한 해상도와 디바이스 환경에서 최적화된 사용자 경험을 제공하는 과정에서도 사용된 기술 스택과 외부 라이브러리의 제약으로 인해 원활한 대응이 어려웠고, 성능과 기능 간의 균형을 맞추는 과정에서 일부 기능을 구현하지 못하거나 성능 저하를 우려해 기능을 제한해야 하는 상황이 발생하면서 결국 시스템의 성능, 확장성, 그리고 크로스 플랫폼 호환성 측면에서 부정적인 영향을 미쳤습니다.

자체평가 - 박재현

본 프로젝트는 프레임워크 없이 정적인 HTML 페이지를 기반으로 개발되었습니다.
그로 인해 동적 렌더링 및 반응형 UI 구현에 제약이 있었으며, 특히 컴포넌트 기반의 동적 렌더링을 활용할 수 없었습니다. 이를 보안하기 위해 정의된 개별 HTML 페이지를 전환하는 방식으로 구현하였고, 상세 페이지 이동 시 기존 페이지가 새롭게 로드되는 방식으로 처리하였습니다.

하지만 이는 내부에서 리디렉션을 통한 화면 이동을 처리함에 있어, 크롬 보안 정책 위반이라는 제약이 발생하였습니다. 이를 해결하기 위해 iframe을 활용하려 했으나, 이 방식은 SEO(검색 엔진 최적화)나, 사이트 간 연계성 측면에서 불편함을 초래할 수 있어 아쉬움이 남았습니다.

CSS를 적용하는 과정에서 발생한 어려움도 있었습니다.
예를 들어, body나 *와 같은 범위가 겹치는 부분이 오픈소스 기능 과 팀원 간의 코드를 합치는 과정에 있어서 스타일을 적용 시키는데 문제가 발생하였습니다. 새로운 오픈소스 기반 기능들의 추가에 어려움이 있었으나, JavaScript와 CSS를 적절히 섞어 스타일을 표현하는 방식으로 제한되지만 자연스러운 기능을 추가하였습니다.

다양한 환경의 해상도에 맞춘 해상도 최적화 과정에서 기술 스택과 외부 라이브러리의 제약으로 인해 다양한 디바이스와 화면 크기의 적응력이 낮아져 최적화된 사용자 경험을 제공하지 못한 것에 대하여 아쉬움이 남았습니다.

또한, 다양한 디바이스에서 원활한 작동을 보장하기 위한 최적화 작업에서 성능과 기능 간의 균형을 맞추는 데 어려움이 있었으며, 그로 인해 해상도 변경으로 인한 동적인 이미지, 애니메이션 처리와 같은 기능과 상세 페이지로 이동함에 있어 성능 저하를 우려해 기능을 제한하여 일부 기능을 구현하지 못하는 상황이 발생했습니다. 이로 인해 시스템의 성능, 확장성, 그리고 크로스 플랫폼 호환성에서 부정적인 영향을 끼쳐 아쉬움이 남았습니다.

Thank You

We're always here for you

CHARMESOFFICIAL@KOREA.com

NEW CHOICES | NEW POSSIBILITIES