
UI 구현

정기 수행 평가 13



Contents

1 UI 설계
- 스토리보드

2 UI 구현 화면
- 구현 화면

3 구현 코드
- 코드

logo
CATE1 CATE2 CATE3 CATE4 CATE5
CART / MY PAGE / LOGIN

CART ②

<input type="checkbox"/> 상품 정보	선택	판매가	수량	적립금	주문금액	배송정보
<input type="checkbox"/> 상품명 <input type="checkbox"/> 옵션[색상] <input type="checkbox"/> 변경	<input type="checkbox"/> 주문하기 <input type="checkbox"/> 카탈로그 <input type="checkbox"/> 삭제	29,800원	1	<input type="checkbox"/> 596원 <input type="checkbox"/> 298원 <input type="checkbox"/> 298원 <input type="checkbox"/> 298원	26,900원	2 무료
<input type="checkbox"/> 상품명 <input type="checkbox"/> 옵션[색상] <input type="checkbox"/> 변경	<input type="checkbox"/> 주문하기 <input type="checkbox"/> 카탈로그 <input type="checkbox"/> 삭제	29,800원	1	<input type="checkbox"/> 596원 <input type="checkbox"/> 298원 <input type="checkbox"/> 298원 <input type="checkbox"/> 298원	29,800원	무료

☐ 선택 상품 삭제
☐ 장바구니 비우기

공 주문 상품 2개

56,700원
상품금액
+
무료
배송비
=
56,700원
총 주문금액

WISH LIST

상품이미지

상품명
29,800원
26,900원

상품이미지

상품명
29,800원

1. 등급별 아이콘, 결제수단 별 (2%) 볼을지 별지 정할 것
2. 합계 5만 > 배송비 무료 컬럼병합
3. 수량버튼, 선택버튼 이미지 제작

1 UI 설계

| 스토리 보드



위시리스트 레이아웃

2 UI 구현 화면

Beauty Shop

김동근 로그아웃 장바구니(1) 마이페이지 고객센터

NEW

BEST

OUTER

[TOP](#)

BOTTOM

DRESS

ETC

1 정바구니 2 주문/결제 3 주문완료

장바구니 ①

<input type="checkbox"/>	상품 정보	판매가	수량	합계	배송비	선택
<input type="checkbox"/>	 <p>여성 롱기디언 플루분트 니트 셔츠 간갈이 신상 봄가을 겨울 카디건 신상품 대매리 (원산지: 네이비, M) 출신매장</p>	24,000원	5	24,000원	무료 ①	<div>주정차기</div> <div>원산지등록</div> <div>착제</div>

선택상품 삭제

총 주문 상품

$$120,000\text{원} + 0\text{원} - 0\text{원} = 120,000\text{원}$$
[illegible]

배승비

상품 할인금액

● 中国

선택상품주문

계속 소독하기

위시리스트

467



꾸안꾸 부트인 프롬세스엔
더로 표이로를 꾸안이드팅

김동근님

Lv.1 실버

누적 구매 금액
250,000원

현재 포인트
5,235원

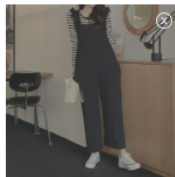
보유 쿠폰
3개

등급 점수
2,500점

- 누적 구매금액은 실시간 반영이 아닌 출고요청 상태 주문건을 기준으로 매주 월요일 새벽에 집계됩니다.
- 누적 구매금액은 최근 3년 치가 반영됩니다.

위시리스트 1

비우기 X



꾸안꾸 루드의 프릴서스펜
더트포인트를 준 와이드...
25,000

장바구니, 위시리스트 구현 화면

3 구현 코드

| 구현 코드

```
<!-- 메인 -->
<main>
  <div class="mainBackgorund">
    <div class="mainContent">
      <!-- 콘텐츠 -->
      <section class="myContent">
        <!-- 마이페이지 메뉴-->
        <th:block th:replace="~{myshop/_mymenu :: mymenuFragment}"></th:block>

        <!-- 회원정보 -->
        <th:block th:replace="~{myshop/_mymenu :: myheadFragment}"></th:block>

        <!-- 위시리스트 -->
        <div class="wishlistFrame">
          <div class="wishlistHead">
            위시리스트
            <span class="headCount" id="headCount" th:text="#{#lists.size(wishlist)}"></span>
            <a class="deleteAllWish" href="#">비우기 X</a>
          </div>
          <article class="wishlist">
            <div class="emptyWish" th:if="#{#lists.size(wishlist) == 0}">원한 상품이 없습니다.</div>
            <!-- 아이템 반복 -->
            <th:block th:each="wish:${wishlist}">
              <div class="item">
                <div class="itemThumb">
                  <a th:href="@{/shop/view(pno=${wish.prodNo})}">
                    <!--
                      
                    -->
                    
                  </a>
                  <a class="delete" th:href="@{/myshop/deleteSelectedWish(wishNo=${wish.wishNo})}">x</a>
                </div>
                <div class="itemInfo">
                  <a th:href="@{/shop/view(pno=${wish.prodNo})}" th:text="#{wish.prodName}"></a>
                  <span class="price" th:text="#{#numbers.formatInteger(wish.getSalePrice, 0, 'COMMA')}"></span>
                </div>
              </div>
            </th:block>
            <!-- 아이템 반복 종료 -->
          </article>
        </div>
      </section>
    </div>
  </div>
</main>
```

```
<delete id="deleteSelectedWish">
  DELETE FROM `wish` WHERE `wishNo`=#{wishNo};
</delete>

<delete id="deleteAllWish">
  DELETE FROM `wish` WHERE `uid`=#{uid};
</delete>

$(function(){
  /* 위시리스트 전체삭제 */
  $(' .wishlistFrame > div.wishlistHead > a.deleteAllWish').click(function(e){
    e.preventDefault();
    if(confirm('위시리스트를 비우시겠습니까?')){
      $.ajax({
        url      : '/Beauty/myshop/deleteAllWish',
        type     : 'POST',
        success  : function(data){
          if(data == 1){
            $(' .wishlistFrame > article.wishlist').empty();
            $('#headCount').text("0");
            $(' .wishlistFrame > article.wishlist').append('<div clas');
          }
        }
      });
    }
    else{
      return;
    }
  });
});
});
```

Wishlist.html

Wishlist.js
동적 삭제 구현

3 구현 코드

| 구현 코드

```
<th:block th:each="cart:${cartList}">
<tr>
<td><input name="chkCart" type="checkbox" th:value="${cart.cartNo}"></td>
<td>
<a th:href="@{/shop/view(pno=${cart.prodNo})}">  </a>
</td>
<td>
<strong><a th:href="@{/shop/view(pno=${cart.prodNo})}" th:text="${cart.profileName}"></a><br/></strong>
<span>
<span class="color" th:text="${cart.color}"></span>,
<span class="size" th:text="${cart.size}"></span>
<br/>
</span>
<span>
<button type="button" class="btn openOption" data-bs-toggle="modal" data-bs-target="#changeOptionModal">
옵션변경
</button>
</span>
</td>
<td class="tdTotal">
<!-- 할인 상품 구분 -->
<th:block th:if="${cart.discount == 0}">
<span class="price" th:text="${#numbers.formatInteger(cart.getSalePrice, 0, 'COMMA')}"></span>원
</th:block>
<th:block th:unless="${cart.discount == 0}">
<span class="oriPrice" th:text="${#numbers.formatInteger(cart.price, 0, 'COMMA')}"></span>원 <br>
<span class="price" th:text="${#numbers.formatInteger(cart.getSalePrice, 0, 'COMMA')}"></span>원 <span class="discount" th:text="${cart.discount}">
</th:block>
<input type="hidden" id="totalCount" name="count" th:value="${cart.count}" readonly="readonly">
</td>
<td class="tdCount">
<span class="countFrame">
<input id="count" type="text" name="count" th:value="${cart.count}" readonly="readonly">
<a class="btnIncrease" href="#"></a>
<a class="btnDecrease" href="#"></a>
</span>
</td>
<td><span th:text="${#numbers.formatInteger(cart.getTotalPrice, 0, 'COMMA')}"></span>원</td>
<!-- 유행예안 나열 및 클릭처리를 위해 CSS적용 -->
<td class="deliveryInfo" rowspan="100">
<!-- 배송비 정보 -->
<span th:if="${#aggregates.sum(cartList.[getTotalPrice]) >= 50000}">무료</span>
<span th:unless="${#aggregates.sum(cartList.[getTotalPrice]) >= 50000}">2500원</span>
<label data-bs-toggle="popover" data-bs-placement="top" data-bs-html="true" data-bs-trigger="focus" tabindex="0">
data-bs-content="기름 배송비 2,500원 <br/> 50,000원 이상 구매 시 무료배송">
<span id="deliIcon" class="material-symbols-outlined">help</span>
</label>
</td>
<td>
<a class="btn btnTableOrder" href="#">주문하기</a>
<a class="btn btnTableAddwish" href="#">관심상품등록</a>
<a class="btn btnTableDelete" href="#">삭제</a>
<input type="hidden" name="prodNo" th:value="${cart.prodNo}">
</td>
</tr>
</th:block>
<!-- 아이템 반복 끝-->
```

Cart.html
아이템 부분

```
<!-- 삭제 버튼 -->
<div class="cartBtns">
<a id="btnDeleteSelectedCart" class="btn" href="#">선택상품 삭제</a>
<a id="btnDeleteALLCart" class="btn" href="#">비우기</a>
</div>
</article>
<!-- Total 테이블 -->
<table class="totalTable" th:if="${#lists.size(cartList) != 0}">
<tr>
<th>총 주문 상품 <span id="totalCount"></span></th>
</tr>
<tr>
<td>
<div class="priceInfoContainer">
<div class="cartPrice">
<div class="priceWrap">
<p id="totalPrice"></p>
<p>상품금액</p>
</div>
</div>
<div class="deliPrice">
<div style="line-height: 35px;">
<i class="im-icon im-ico-plus-plain"></i>
</div>
<div class="priceWrap">
<p id="totalDelivery"></p>
<p>배송비</p>
</div>
</div>
<div class="disPrice">
<div style="line-height: 35px;">
<i class="im-icon im-ico-minus-plain"></i>
</div>
<div class="priceWrap">
<p id="totalDisPrice"></p>
<p>상품 할인금액</p>
</div>
</div>
<div class="totalPrice">
<div style="line-height: 35px;">
<i class="im-icon im-ico-equals-plain"></i>
</div>
<div class="priceWrap">
<p id="totalTotalPrice"></p>
<p>총 주문금액</p>
</div>
</div>
</td></tr>
</table>
</div>
<!-- 주문 버튼 -->
<div class="cartBtns">
<a id="btnOrdersSelect" class="btnOrder select" href="#">선택상품주문</a>
<a id="btnOrderALL" class="btnOrder all" href="#">전체상품주문</a>
<a class="keepShopping" th:href="@{/index}">계속 쇼핑하기</a>
</div>
<!-- 회원인증 위시리스트 -->
<th:block sec:authorize="isAuthenticated()">
```

Cart.html
버튼 및 가격 테이블

3 구현 코드

| 구현 코드

```
/* 체크박스 처리 */
/* 전체선택시 */
$('#input[name=chkAll]').click(function(){
    let checkList = $('#input[name=chkCart]');
    if($(this).is(":checked")){
        checkList.prop("checked", true);
    }else{
        checkList.prop("checked", false);
    }
    check();
});
/* 개별선택 -> 모두 체크한 경우 전체선택 자동해제 */
$('#input[name=chkCart]').click(function(){
    let isAll = true;
    $('#input[name=chkCart]').each(function(){
        isAll = isAll && $(this).is(":checked");
    });
    $('#input[name=chkAll]').prop("checked", isAll);
    check();
});
/* 전부 체크 해제인 경우 */
function check() {
    if ($('#input:checkbox[name=chkCart]').is(":checked")==false) {
        /* 전부 선택 해제 */
        sumAll();
    }else {
        /* 개별 선택 해제 */
        sumChecked();
    }
}

/* 수량증가 */
$('#wrap table a.btnIncrease').click(function(e){
    e.preventDefault();
    let amount = $(this).prev().val();
    if(amount==10){
        alert('10개 초과로 주문할 수 없습니다.');
```

Cart.js
체크박스, 수량 증감

```
/* 카트 선택 삭제 */
$('#btnDeleteSelectedCart').click(function(e){
    e.preventDefault();
    if ($('#input[name=chkCart]').length == 0) {
        alert('장바구니에 상품이 없습니다.');
```

Cart.js
카트 삭제 동적 구현

3 구현 코드

| 구현 코드

```
/* tableBtns */
/* 주문하기 */
$('div.cartFrame > article.cartList > table > tbody td > a.btnTableOrder').click(function(e){
    e.preventDefault();
    let chkList = [];
    chkList.push(parseInt($(this).parent().parent().children('td:first-child').children('input').val()));
    redirectGet('/Beauty/order/orderform', chkList);
});

/* 삭제 */
$('div.cartFrame > article.cartList > table > tbody td > a.btnTableDelete').click(function(e){
    e.preventDefault();
    let tr = $(this).parent().parent();
    let cartNo = parseInt($(this).parent().parent().children('td:first-child').children('input').val());
    $.ajax({
        url: '/Beauty/order/deleteSelectedCart',
        type: 'GET',
        data: {'cartNo': cartNo},
        dataType: 'json',
        success: function(data){
            console.log(data);
            if(data == 1){
                //장바구니 개수 변경, 가격 반영, 행 삭제
                tr.remove();
                $('#headCount').text($('article.cartList > table > tbody').children().length);
                $("input:checkbox[name='chkAll']").prop("checked", false);
                totalPrice();
                //테이블이 비었으면
                if($('article.cartList > table > tbody').children().length == 0){
                    emptyTable();
                }
            }else{
            }
        }
    });
});
```

```
/* 빈 카트 */
function emptyTable(){
    //카트 비었으면
    //테이블, 토달테이블 삭제, div.emptyCart 삽입
    let tag = "<tr><td colspan='8'>장바구니에 담긴 상품이 없습니다.</td></tr>";
    $('tbody').append(tag);
    $('totalTable').remove();
}

/* 선택 구분 함수 */
function sumChecked(){
    initVal();
    $('input[name=chkCart]').each(function(){
        if($(this).is(":checked")){
            sum($(this));
        }
    });
    checkDelivery();
    inputTotal();
}

function sumAll(){
    initVal();
    $('input[name=chkCart]').each(function(){
        sum($(this));
    });
    checkDelivery();
    inputTotal();
}

/* 실제 기능 구현 함수 */
function makeNum(nanValue){
    return parseInt(nanValue.split(',').join(""));
}

function initVal(){
    headCnt = 0;
    count = 0;
    price = 0;
    delivery = '2,500';
    disPrice = 0;
    total = 0;
}

function sum(arg){
    let tds = arg.parent().parent().children();
    headCnt += 1;
    count = makeNum(tds[4].children[0].children[0].value);
    price += (makeNum(tds[3].children[0].innerText) * count);
    if(tds[3].children[1] != null){
        disPrice += (makeNum(tds[3].children[0].innerText) - makeNum(tds[3].children[2].innerText)) * count;
    }
    total += makeNum(tds[6].children[0].innerText);
}

function checkDelivery(){
    if(total >= 50000){
        delivery = '무료';
    }else{
        delivery = '2,500원';
        total += 2500;
    }
}
```

Cart.js
개별 주문, 삭제 (동적)

Cart.js
함수들

3 구현 코드

| 구현 코드

```
<!-- 김동근 -->
<!-- 카트 -->
<select id="selectCartList" resultType="CartVO">
    SELECT a.`cartNo`, a.`prodNo`, a.`count`, b.`prodName`, b.`price`, b.`discount`,
    FROM `cart` AS a
    JOIN `product` AS b
    ON a.`prodNo` = b.`prodNo`
    <if test="arg0 != null and arg1 == null">
        WHERE `uid` = #{arg0}
    </if>
    <if test="arg1 == null and arg1 != null">
        WHERE `uid` = #{arg1}
    </if>
    <if test="arg1 != null and arg1 != null">
        WHERE `uid` = #{arg0} OR `uid` = #{arg1}
    </if>
    ORDER BY a.`cartNo` DESC;
</select>

<delete id="deleteSelectedCart">
    DELETE FROM `cart` WHERE `cartNo`=#{cartNo};
</delete>

<delete id="deleteAllCart">
    DELETE FROM `cart` WHERE `uid`=#{uid};
</delete>

<select id="checkCountForUpdate" resultType="int">
    SELECT `count` FROM `cart`
    WHERE `cartNo` = #{cartNo};
</select>

<update id="cartIncrease">
    UPDATE `cart` SET
    count = count + 1
    WHERE `cartNo` = #{cartNo};
</update>

<update id="cartDecrease">
    UPDATE `cart` SET
    count = count - 1
    WHERE `cartNo` = #{cartNo};
</update>
```

```
<!-- 카트 모듈 -->
<select id="openOption" resultType="String">
    <!-- 서버쪽과 값을 하나의 행으로 병합 -->
    SELECT GROUP_CONCAT(A.colorName) AS colorName
    FROM (
        <!-- 중복제거 처리한 cartNo가 같은 colorName 행을 반환하는 서버쪽 -->
        SELECT DISTINCT po.prodNo, po.colorName
        FROM `cart` AS c JOIN `product_option` AS po
        ON c.`prodNo` = po.`prodNo`
        WHERE c.`cartNo` = #{cartNo}
    ) A
    GROUP BY `prodNo`;
</select>
<select id="selectOption" resultType="String">
    SELECT GROUP_CONCAT(size) AS sizes
    FROM `product_option`
    WHERE `prodNo`=#{prodNo} AND `colorName`=#{color} AND `sold` > 0
    GROUP BY `prodNo`;
</select>
<update id="saveOption">
    UPDATE `cart` SET
    `color` =#{color},
    `size` =#{size}
    WHERE `cartNo`=#{cartNo};
</update>
```

옵션변경

옵션변경

상품명					상품명							
COLOR	다크그레이	아이보리	블랙	네이비	COLOR	다크그레이	아이보리	블랙	네이비			
SIZE	XS	S	M	L	XL	FREE	XS	S	M	L	XL	FREE
취소 저장					취소 저장							

정기 수행 평가

13. UI 구현 Beauty



Thank you!

Kim Dong Geun
From GreenAcademy
2023.03.24