

PROJECT 01.

내음

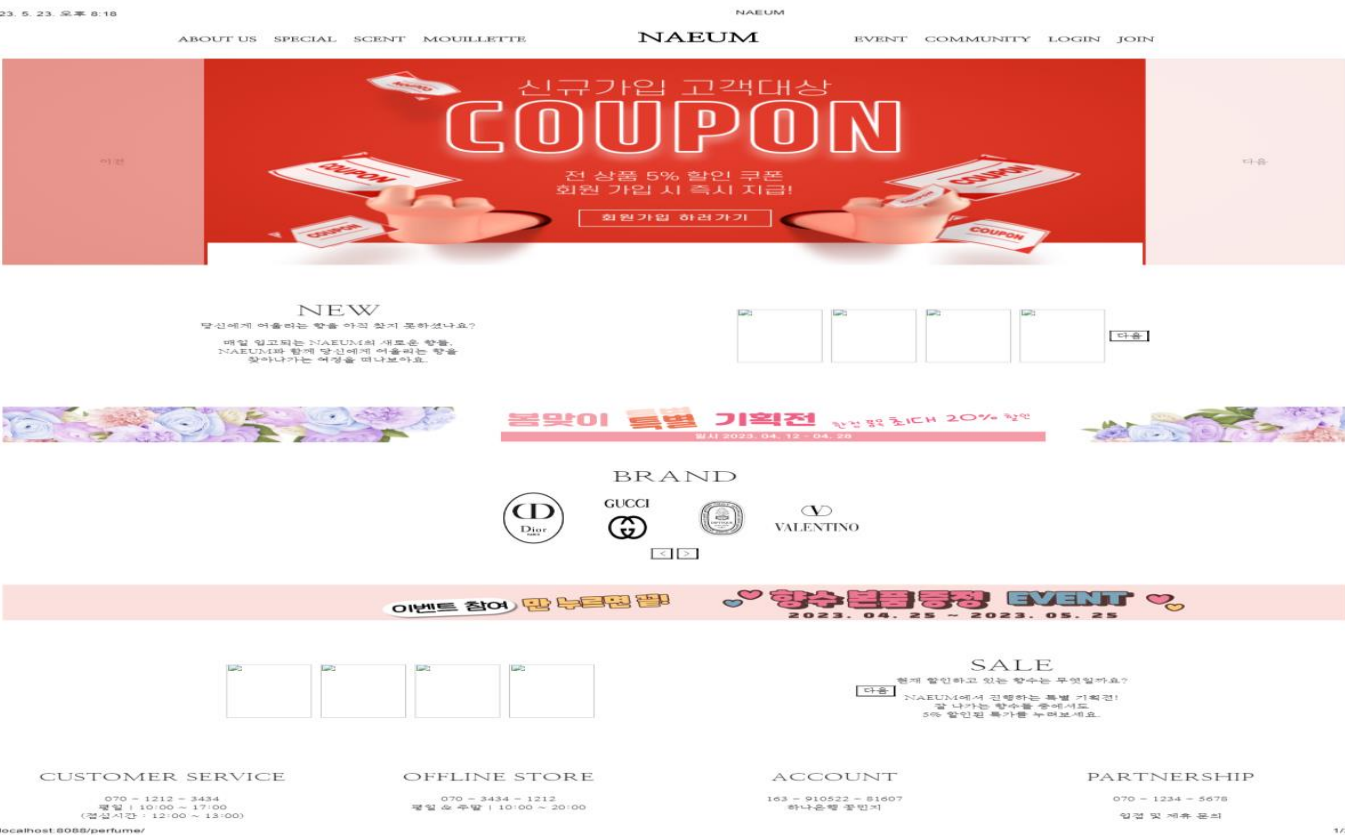
* 포트폴리오는 제가 작업한 내용만 첨부했습니다.

01

ABOUT PROJECT

Spring MVC로
자신에게 맞는 함수를 찾고,
여러 이벤트를 통해 시향 할 수
있는 함수 웹사이트입니다.

내음



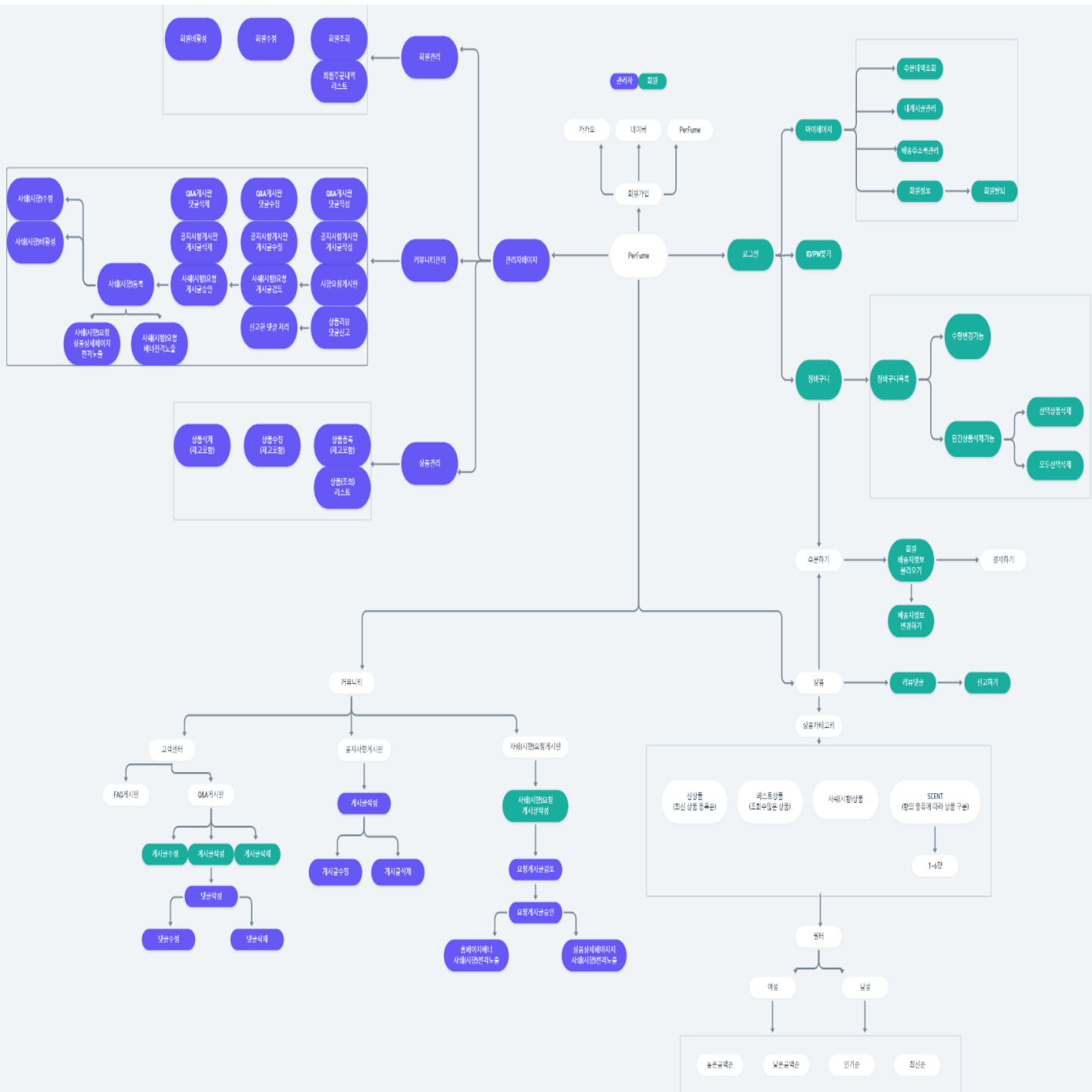
메인페이지

- 프로젝트 주제 : Spring MVC 기반의 사이트 제작
- 프로젝트 기간 : 2023-04-12 ~ 2023-04-28
- 프로젝트 인원 : 6명
- 프로젝트 제작 : JAVA, HTML, CSS, JavaScript, Oracle, BootStrap, Mybatis, Spring, Ajax, jQuery
- 프로젝트 개발 목표
- * Spring Framework와 MVC패턴을 접목하여 쇼핑몰 웹사이트 사이클에 대해 이해하고, 하나의 기능을 완성감 있게 개발하는데 목표를 두었습니다.
- 프로젝트 담당 역할
- * 장바구니 페이지, 주문/결제 페이지, 주문완료 페이지
 - * 기여도 20%
- GitHub 주소
- * <https://github.com/leejh-96/perfume>
 - * 프로젝트 소스 파일 첨부

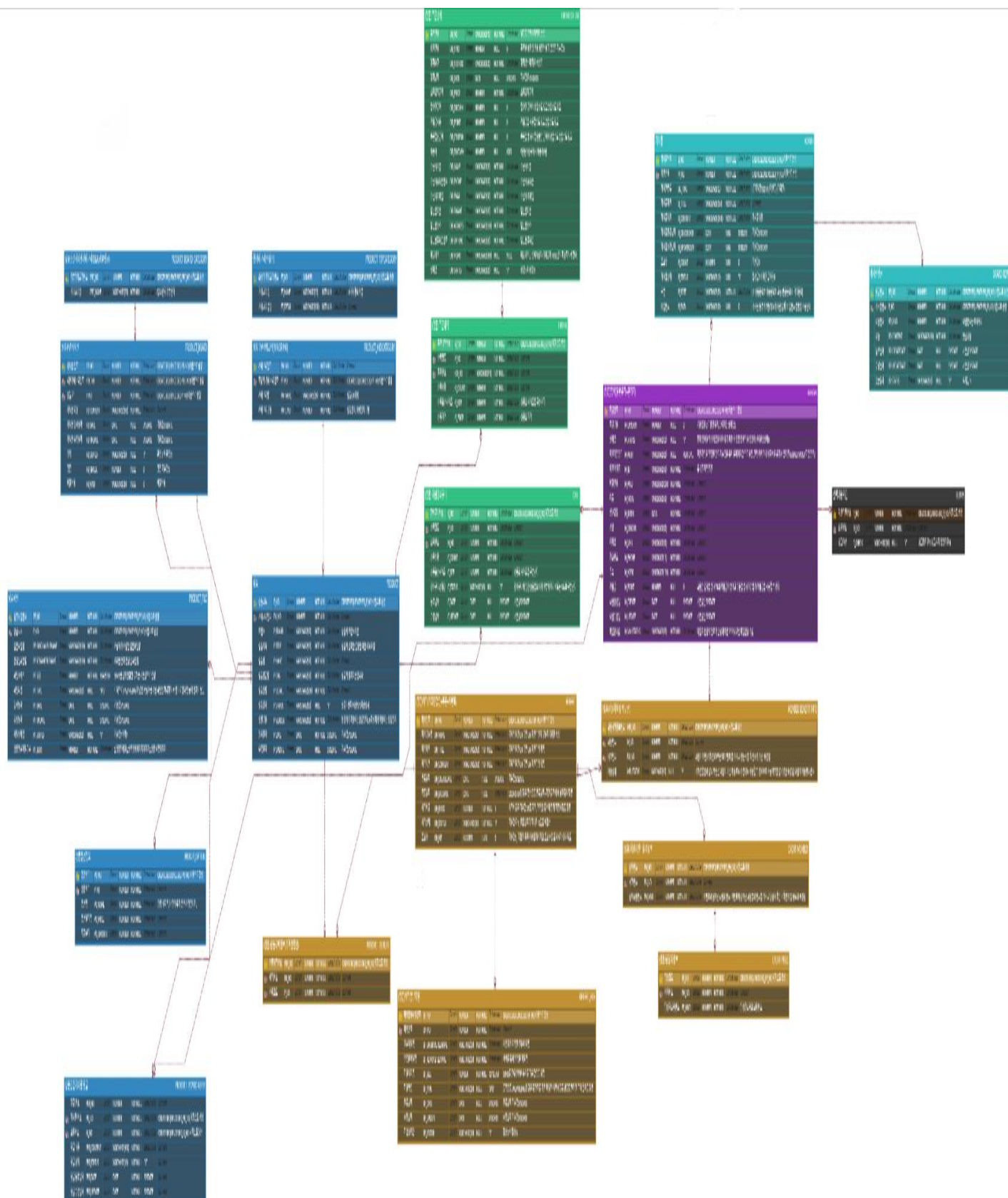
PROJECT 01.

<Flow Chart>

본격적인 UI기획 및 ERD 설계 전,
향수 쇼핑물에 필요한 기능의 작업흐름을 파악하기 위해 플로우 차트를 제작하여 활용했습니다.
(디자인 커뮤니케이션 협업툴인 윈지컬을 통해 제작)



ERD 다이어그램



상세페이지 및 기능구현

<장바구니>



[1] 수량 증가/감소

- 초기에 나타나는 수량은 회원이 장바구니에 처음 저장했을때의 수량이 나타납니다.
- 증가/감소 수량버튼은 현재 상품의 재고수량을 기반으로 작동합니다.
- 증가/감소 버튼을 클릭 할때마다 상품의 할인적용 유무에 따라 각각 상품금액, 최종상품금액이 표시됩니다.
- 체크박스에 체크가 되어있다면 증가/감소버튼은 비활성화 됩니다.



[2] 상품 선택 체크박스/전체 체크박스

- 선택 또는 전체 체크박스를 통해서 상품을 삭제할 수 있습니다.
- Ajax함수를 사용하여 비동기로 서버와 통신하여 수행합니다.

[3] 페이징

- CSS max-height 속성을 사용하여 지정한 사이즈가 넘어가면 스크롤되도록 했습니다.

[4] 합계 계산

- 체크박스를 체크하면 계산기능이 활성화되어 상품금액,배송비,할인금액을 계산하여 합계금액으로 보여줍니다.

[5] 포인트,쿠폰 조회

- 나의포인트-쿠폰란의 [자세히보기]버튼을 통해서 포인트와 쿠폰을 조회할 수 있습니다.
- 포인트와 쿠폰은 장바구니페이지에서는 비활성상태이며, 주문페이지에서 활성화됩니다.



상세페이지 및 기능구현

<장바구니 Service>

- 상품과 회원의 정보를 조회

```
@Override
public CartMember selectCartMemberInfo(int memberNo) {

    CartMember member = null;

    //회원의 쿠폰, 포인트 정보를 가져옴
    member = cartMapper.selectCartMemberInfo(memberNo);

    return member;
}

@Override
public List<Cart> selectCartInfo(int memberNo) {

    List<Cart> clist = null;

    //회원을 식별하는 회원생성번호로 장바구니 목록을 가져옴
    clist = cartMapper.selectCart(memberNo);

    //List에 담긴 상품들의 갯수만큼 반복하는 반복문
    for(int i = 0; i<clist.size(); i++) {
        // n번째 상품의 상품정보, 옵션정보를 가져옴
        clist.get(i).setCartProduct(cartMapper.selectCartProductInfo(clist.get(i).getCartNo(), clist.get(i).getProductNo()));
        // n번째 상품에 적용된 할인정보 목록을 가져옴 (기획전할인, 이벤트적용, 쿠폰적용)
        clist.get(i).setBenefitList(cartMapper.selectProductBenefit(clist.get(i).getProductNo()));
    }
    return clist;
}
```

<장바구니 cart-mapper.xml>

- 장바구니에 담긴 상품의 상세정보, 옵션정보, 이미지를 조회하는 쿼리문

```
<!-- 장바구니에 담긴 상품의 상세정보, 옵션정보, 이미지를 조회하는 쿼리문 -->
<select id="selectCartProductInfo" parameterType="_int" resultMap="selectCartProductInfoResultMap">
SELECT
    P.P_BRAND,
    P.P_TITLE,
    P.P_NAME,
    P.P_ENG,
    PO.PO_NAME,
    PO.PO_PRICE,
    PO.PO_AMOUNT,
    PF.PF_ORIGINALFILENAME,
    PF.PF_RENAMEFILENAME
FROM CART C
JOIN PRODUCT P ON (C.P_NO = P.P_NO)
JOIN PRODUCT_OPTION PO ON (P.P_NO = PO.P_NO)
JOIN PRODUCT_FILE PF ON (P.P_NO = PF.P_NO)
WHERE
    C.C_NO = #{cartNo}
    AND C.C_SIZE = PO.PO_NAME
    AND PF.P_NO = #{productNo}
    AND PF.PF_SORT = 1
    AND P.P_STATUS = 'Y'
    AND PF.PF_STATUS = 'Y'
</select>
```

상세페이지 및 기능구현

01상세구현 > 02주문서작성/결제 > 03주문완료

상품 정보	수량	상품금액	택배	최종상품금액
<div> <div>[이미지]</div> <div> <div>[트라보] Le Labo The N°29 Eau de Parfum</div> <div>[용량사이즈: 100ml]</div> <div>[현재 재고: 100개]</div> </div> </div> <div> <div>-</div> <div>2</div> <div>+</div> </div> <div>100,000 원</div> <div>-</div> <div>200,000 원</div>				

나의 포인트-쿠폰

지재리보기

품질상품을 제외한 가격입니다

총2개의 상품금액200,000원

+

배송비0원

-

기적전 할인금액 0원

-

합계200,000원

< 쇼핑 계속하기

01상세구현 > 02주문서작성/결제 > 03주문완료

상품 정보	수량	상품금액	택배	최종상품금액
<div> <div>[이미지]</div> <div> <div>[트라보] Le Labo The N°29 Eau de Parfum</div> <div>[용량사이즈: 100ml]</div> <div>[현재 재고: 100개]</div> </div> </div> <div> <div>-</div> <div>2</div> <div>+</div> </div> <div>100,000 원</div> <div>-</div> <div>200,000 원</div>				

나의 포인트-쿠폰

지재리보기

이정훈 님의 보유적립금

100000

적립금 사용하기

취소하기

170125 point

이정훈 님의 보유쿠폰

회원가입쿠폰:5%

기간: 2023-04-27~2023-05-27

적용중

취소하기

나에게 맞는 향수 테스트 쿠폰 이벤트:5%

기간: 2023-04-01~2023-06-01

적용하기

총2개의 상품금액200,000원

+

배송비0원

-

기적전 할인금액 0원

-

합계200,000원

주문자 정보

이정훈 님의 보유쿠폰

회원가입쿠폰:5%

기간: 2023-04-27~2023-05-27

적용중

취소하기

나에게 맞는 향수 테스트 쿠폰 이벤트:5%

기간: 2023-04-01~2023-06-01

적용하기

합계200,000원

주문자 정보

* 주문하시는 분 성함

이름을 입력해주세요.

* 휴대폰번호

형식 010-0000-0000

* 이메일

형식 kildoug@kh.com

* 받는곳 주소

도로명 주소란 입력해주세요.

우편번호 찾기

상세 주소란 입력해주세요.

배송 정보

* 배송지 확인

☒ 직접 입려 ☐ 주문지 정보와 동일

* 받으실분 성함

이름을 입력해주세요.

* 받는곳 주소

도로명 주소란 입력해주세요.

우편번호 찾기

상세 주소란 입력해주세요.

* 휴대폰번호

형식 010-0000-0000

결제정보

배송비

-배송비 무료-

할인/적립/쿠폰 사용내역

기적전 할인금액: 0원 적립금 사용: 100000point 쿠폰 사용 할인: 10,000원

최종 상품 결제 금액

190,000원

결제방법

카드결제

키키오페이

페이코

토스페이

☐ * PERFUME의 모든 약관을 확인하고 결제 동의 합니다.

*개인정보 수집 이용 동의

☐ (필수) 개인정보 수집 이용에 대한 내용을 확인 하였습니다 이에 동의 합니다.

자세히보기

*이용약관 동의

☐ (필수) 이용약관에 대한 내용을 확인 하였습니다 이에 동의 합니다.

지재리보기

결제하기

<주문페이지>

- [1] 주문페이지에서는 체크박스,수량버튼이 비활성화됩니다.
- [2] 주문페이지에서는 포인트와 쿠폰이 사용가능합니다.
- [3] 쿠폰의 적용하기 버튼을 누르면 Ajax함수를 사용하여 비동기로 서버와 통신하여 쿠폰을 처리합니다. 또한, 취소하기 버튼을 누르면 쿠폰사용이 취소되며 마찬가지로 비동기통신을 사용하여 쿠폰을 처리합니다.
- [4] 우편번호 찾기를 클릭하면 주소 api를 사용하여 주소를 검색할 수 있습니다.
- [5] 적용된 포인트와 쿠폰의 사용내역은 하단의 결제정보란 사용내역에서 확인이 가능합니다.

상세페이지 및 기능구현

<주문페이지 적립금 사용하기 버튼 코드>

```
//적립금 사용하기를 클릭하면 이벤트가 발생하여 pointCheck함수가 호출
function pointCheck() {

    //입력한 값을 읽어온 후 숫자로 형변환해준다.
    let pointInput = $('#pointInput').val();
    let point2 = parseInt(pointInput);

    //화면에 표시된 회원의 첫 적립금을 읽어온 후 특수문자를 빼고 숫자로 형변환해준다.
    let memberpoint = $('#memberpoint').text();
    let formatmemberpoint= memberpoint.split(',').join("");
    let parseformatmemberpoint = parseInt(formatmemberpoint);

    //입력한 값이 회원의 첫 적립금보다 큰지 체크하는 조건문
    if (point2 > parseformatmemberpoint) {
        alert('적립금 한도를 초과했습니다. 다시 입력해주세요!^^')
        //클릭이벤트를 강제로 발생시켜 취소하기 버튼이 클릭된다.
        $('#reset').trigger('click');
    }else if (point2 < parseformatmemberpoint || point2 == parseformatmemberpoint) {

        //다시 한번 회원에게 사용유무를 확인한다.
        if (confirm(point2+'point 적립금을 사용하시겠습니까?')) {

            if (point2<parseformatmemberpoint || point2 == parseformatmemberpoint) {
                //결제정보란 사용내역에 사용한 적립금을 표시해주기 위해서 포인트 사용내역 text를 읽어온 후 특수문자를 빼고 숫자로 형변환해준다.
                let savesale = $('#save-sale2').text()
                let formatsavesale= savesale.split(',').join("");
                let parseformatsavesale = parseInt(formatsavesale);

                parseformatsavesale = point2
                pointremember = point2
                console.log('pointremember : '+pointremember)

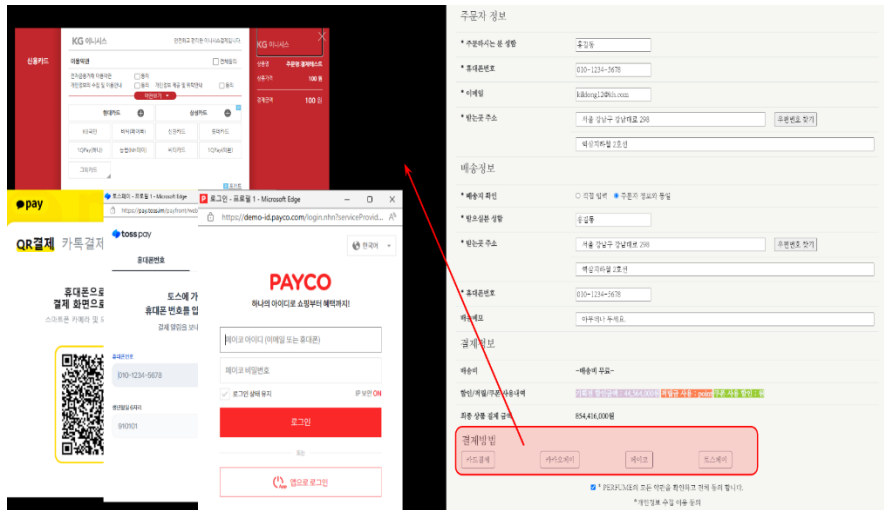
                //결제정보란 사용내역에 사용한 적립금을 표시해준다.
                $('#save-sale2').text(parseformatsavesale);

                //사용된 적립금은 결제정보란 최종결제금액에 반영되어 나타나야하기 때문에 계산해주기 위해 최종결제금액 text를 읽어온 후 특수문자를 빼고 숫자로 형변환해준다.
                let finaltotal = $('#finaltotal').text();
                let formatfinaltotal= finaltotal.split(',').join("");
                let parseformatfinaltotal = parseInt(formatfinaltotal);

                //최종결제금액에서 사용한 적립금을 뺀 후 다시 최종결제금액 text에 (원) 단위로 포맷하여 보여준다.
                $('#finaltotal').text(new Intl.NumberFormat('ko-kr').format(parseformatfinaltotal-pointremember));

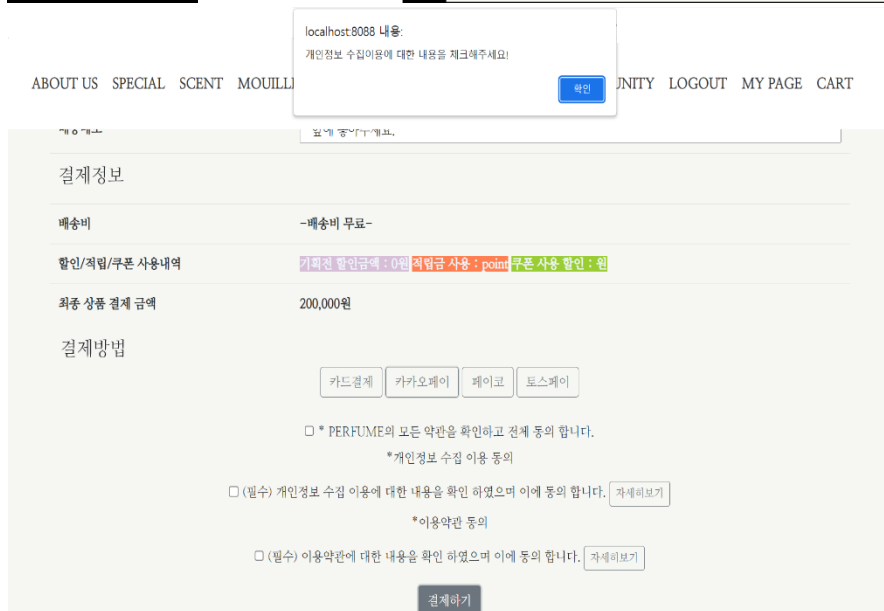
                //마지막으로 적립금사용하기 버튼을 비활성화 시킨다.
                $('#pointtry').attr('disabled', true);
            }
        }
    }else {
        alert('적립금을 입력해주세요!^^')
    }
}
```


상세페이지 및 기능구현



<결제기능>

- [1] 사용자는 4가지 PG사를 통해서 결제를 진행할 수 있습니다.
- [2] 결제방법 선택 후 결제하기 버튼을 통해서 선택한 PG사의 결제창이 호출되어 나타납니다.
- [3] 결제시, (필수)약관 동의사항을 체크하지 않으면 결제가 진행되지 않고 알림창이 나타납니다.
- [4] 결제가 모두 정상적으로 완료되면 주문완료 페이지로 이동합니다.
- [5] 주문완료 페이지의 왼쪽 구성은 주문,배송지,결제정보로 이루어져 있으며, 오른쪽 구성은 주문한 상품들에 대한 정보를 보여줍니다.
- [6] 매 결제 시, 최종결제 금액의 5%가 회원의 적립금으로 적립됩니다.



<결제완료 후 주문완료 페이지 요청 수행 과정>

① Controller

```
@GetMapping("/cart/orderList/{orderNo}/{memberNo}/{plusPoint}")
public String orderList(@PathVariable("orderNo") String orderNo,
                        @PathVariable("memberNo") int memberNo,
                        @PathVariable("plusPoint") int plusPoint,
                        Model model) {
    Order order = null;

    /*
     * 결제 시 생성된 주문번호를 매개변수로 받아서 데이터베이스에 저장된 결제정보, 배송지정보, 주문상품정보를 조회한다.
     */
    order = cartService.orderList(orderNo);

    model.addAttribute("order", order);

    //매 결제시 계산이 완료된 적립금을 주문완료페이지로 렌더링해서 보여주기 위한 작업
    model.addAttribute("plusPoint", plusPoint);

    return "cart/orderList";
}
```

② Service

```
@Override
public Order orderList(String orderNo) {

    Order order = null;

    order = cartMapper.orderList(orderNo);

    return order;
}
```

<결제완료 후 주문완료 페이지 요청 수행 과정>

③cart-mapper.xml

1. DB의 컬럼명 규칙은 스네이크 표기법을 사용하여 설정했고,
Java는 카멜표기법으로 설정을 하여 각각 매핑되는 변수와 컬럼명이 일치하지 않았습니다.
따라서, 복잡한 매핑을 해결하기 위해 Mybatis에서 제공하는 ResultMap을 사용하여 변수와 컬럼명을 매핑시켰습니다.

```
<select id="orderList" parameterType="string" resultMap="orderListResultMap">
    SELECT
        od_no,
        od_mno,
        od_method,
        od_date,
        od_price,
        od_discash,
        od_point,
        od_coupon,
        od_dlvcash,
        od_name,
        od_phone,
        od_email,
        od_dname,
        od_address,
        od_dphone,
        od_memo
    FROM ORDER_DETAIL
    WHERE OD_NO = #{orderNo}
</select>
```

2. 한번의 주문으로 여러 상품들이 주문 가능하기 때문에 테이블 간의 관계는 1:N 관계였습니다.
1:N관계를 resultMap의 collection을 활용하여 서브쿼리 형식으로 데이터를 조회하기 위해 사용했습니다

```
<resultMap id="orderListResultMap" type="com.scent.perfume.cart.model.vo.Order" >
    <id property="orderNo" column="od_no"/>
    <result property="memberNo" column="od_mno"/>
    <result property="orderMethod" column="od_method"/>
    <result property="orderDate" column="od_date"/>
    <result property="finalPrice" column="od_price"/>
    <result property="discount" column="od_discash"/>
    <result property="point" column="od_point"/>
    <result property="coupon" column="od_coupon"/>
    <result property="delivery" column="od_dlvcash"/>
    <result property="orderName" column="od_name"/>
    <result property="orderPhone" column="od_phone"/>
    <result property="orderEmail" column="od_email"/>
    <result property="recipientName" column="od_dname"/>
    <result property="recipientAddr" column="od_address"/>
    <result property="recipientPhone" column="od_dphone"/>
    <result property="memo" column="od_memo"/>
    <collection property="orderList" javaType="arraylist" select="orderDetailList" column="od_no"/>
</resultMap>
```

상세페이지 및 기능구현

3. collection을 활용하여 select에 실행할 쿼리문을 작성해주고, 위와 같이 해당하는 쿼리문이 실행되도록 작성하였습니다.

```
<select id="orderDetailList" parameterType="string" resultMap="orderDetailResultMap">
    SELECT O.OD_NO,
           PF.PF_RENAMEFILENAME,
           P.P_NO,
           P.P_NAME,
           P.P_ENG,
           P.P_BRAND,
           O.O_SIZE,
           O.O_PRICE,
           O.O_COUNT
    FROM
    "ORDER" O
    LEFT JOIN PRODUCT P ON (O.P_NO = P.P_NO)
    LEFT JOIN PRODUCT_FILE PF ON (P.P_NO = PF.P_NO)
    where od_no = #{orderNo} AND PF.PF_SORT = 1
</select>
```

4. 복잡한 매핑을 해결하기 위해 Mybatis에서 제공하는 resultMap을 사용하여 변수와 컬럼명을 매핑시켰습니다.

```
<resultMap id="orderDetailResultMap" type="com.scent.perfume.cart.model.vo.OrderList">
    <id property="orderNo" column="OD_NO"/>
    <result property="orderFileName" column="PF_RENAMEFILENAME"/>
    <result property="productNo" column="P_NO"/>
    <result property="productName" column="P_NAME"/>
    <result property="productEng" column="P_ENG"/>
    <result property="productBrand" column="P_BRAND"/>
    <result property="orderSize" column="O_SIZE"/>
    <result property="orderPrice" column="O_PRICE"/>
    <result property="orderCount" column="O_COUNT"/>
</resultMap>
```

***저의 프로젝트 포트폴리오를 봐주셔서 감사합니다.**