

라인컴퓨터아트학원

CART 장바구니

장보라

웹표준기반 스마트UI/UX 디자인 콘텐츠 제작_A

INDEX

1 장바구니 (2 - 3)

2 사용 색상 / 폰트 (4 - 5)

3 스케치 (6 - 7)

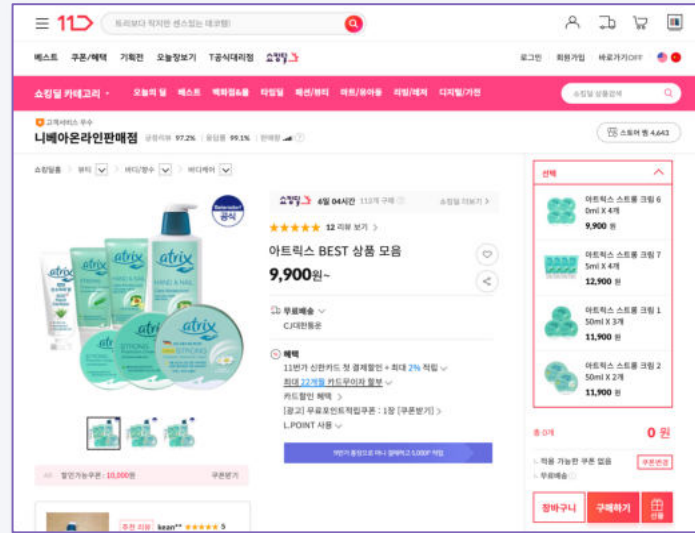
4 화면 구성 (8 - 10)

5 코드 (11 - 18)

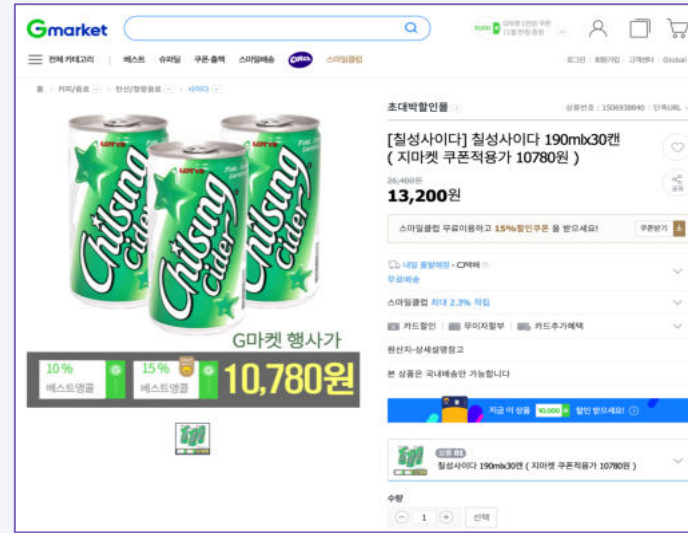
6 총평 (19 - 20)

1 장바구니

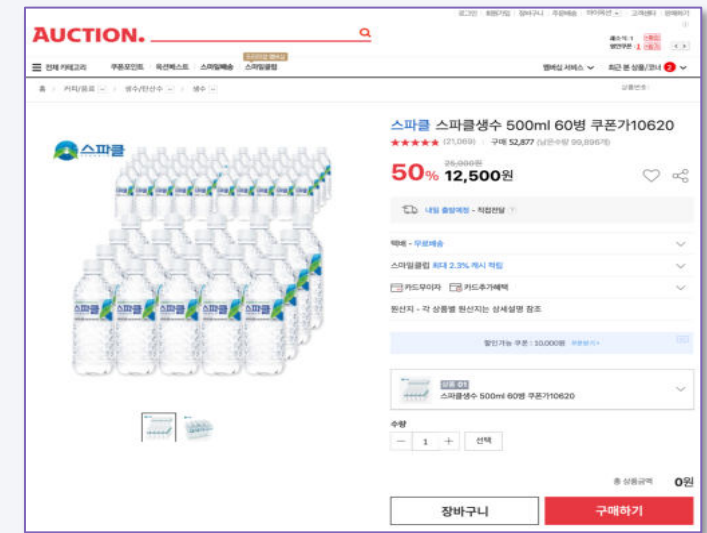
장바구니



< 11번가 >



< Gmarket >



< AUCTION >

위 세 곳의 사이트에서 판매 목록에 있는 상품을 클릭하면 나오는 홈페이지의 화면 구성을 참고하여 장바구니 홈페이지를 구상하였다.

상품의 종류는 평소에 관심이 많은 화장품으로 정하였고, 홈페이지의 기능으로는 상품의 옵션을 고르고, 수량을 지정한 후 장바구니에 담을 수 있게 하였다.

2 사용 색상 / 폰트

사용 색상 / 폰트

사용 색상



#323232



#C7C7C7



#FFFFFF

제품의 이미지와 어울리는 색이면서 고급스러운 느낌을 주기 위하여 무채색으로 정하였다.

텍스트에는 가독성을 높이기 위해 탁한 검정색인 #323232를 사용하였다.

화면의 영역을 나누는 테두리는 텍스트를 읽을 때 방해되지 않도록 밝은 회색인 #C7C7C7을 사용하였다.

그리고 홈페이지의 배경색으로는 심플한 느낌을 주기 위해 기본 값인 #FFFFFF를 사용하였다.

폰트

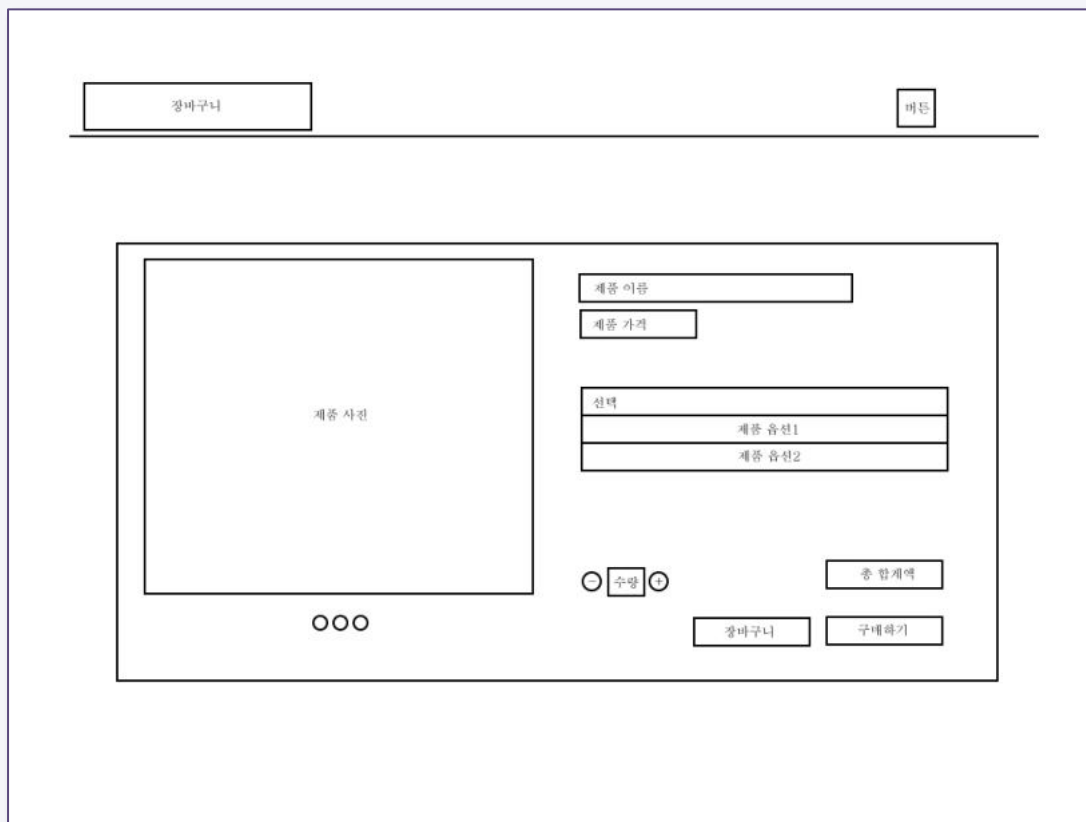
리디바탕체

화장품 이미지에 맞추어 깔끔하면서 제품의 대한 정보가 한 눈에 또렷하게 읽힐 수 있도록 홈페이지 전체에 리디바탕체를 사용하였다.

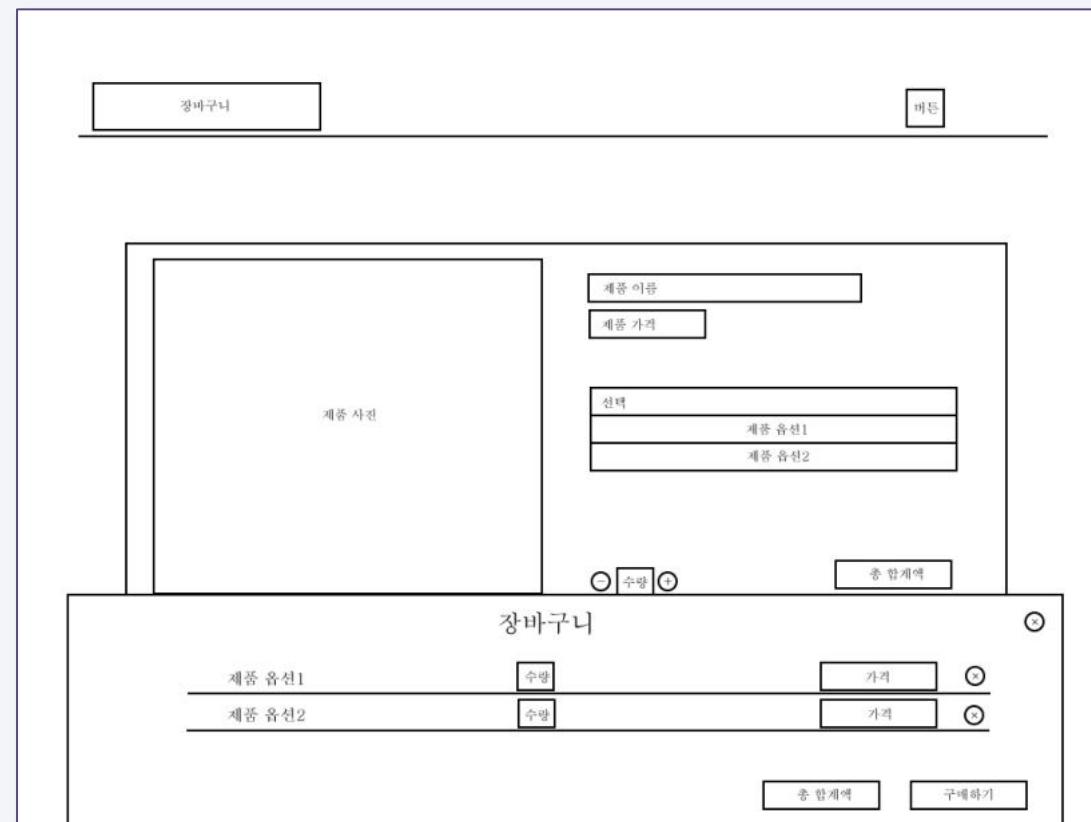
3

스케치

스케치



< 기본 상품 구매 화면 >



< 장바구니 버튼 클릭 모달 >

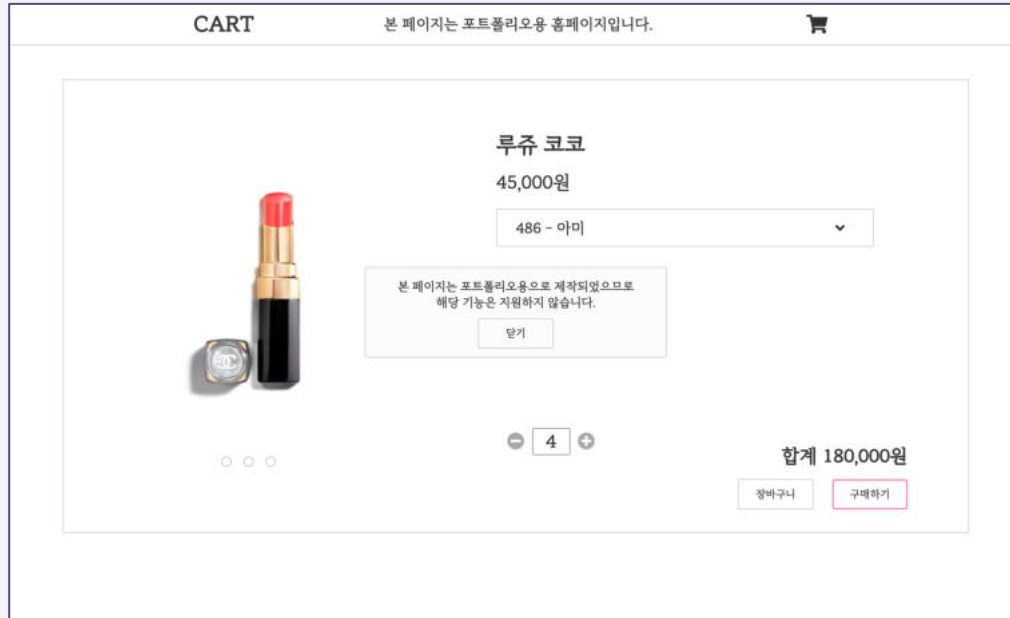
4 화면구성

화면 구성

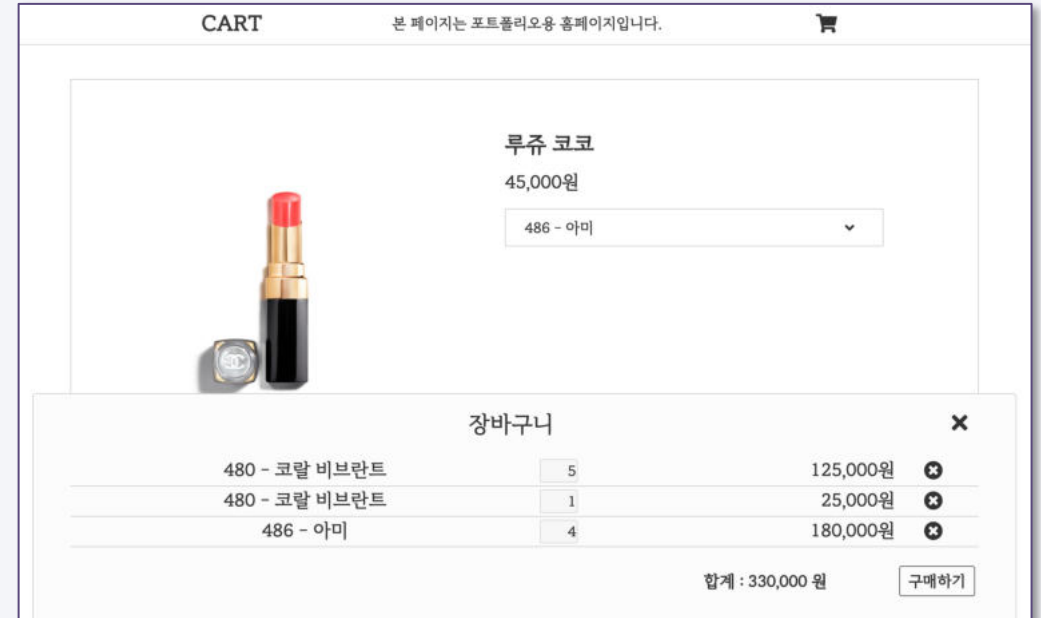


<기본 화면>

화면구성



<구매하기 클릭 시 화면>



<장바구니 클릭 시 화면>

구매하기 버튼을 누를 경우 포트폴리오용 홈페이지라는 것을 알리는 모달이 나타나고, 장바구니 버튼을 누를 경우 선택하고 있던 제품의 종류와 개수가 추가된 장바구니 모달이 나타난다. 장바구니 모달에서는 장바구니에 추가한 제품을 우측 버튼을 통해 삭제할 수 있다. 장바구니 모달의 구매하기 버튼도 마찬가지로 포트폴리오 알림 모달이 나타난다. 우측 상단의 X버튼을 누를 경우 모달이 사라진다.

5 코드

코드 - SCSS

```
.modalPofol{
  position: absolute;
  top:50%;
  left:50%;
  margin-top:-60px;
  margin-left:-200px;
  width:400px;
  height:120px;
  border:1px solid $grey;
  background-color: #fcfcfc;
  border-radius: 3px;
  font-size: 16px;
  text-align: center;
  padding-top:15px;
  button{
    width:100px;
    height:40px;
    @include btnNone();
    border:1px solid $grey;
    border-radius: 3px;
    margin-top:10px;
  }
  button:hover{
    width:100px;
    height:40px;
    @include btnNone();
    border:1px solid $grey;
    border-radius: 3px;
    background-color: #rgb(231, 231, 231);
    margin-top:10px;
  }
}
```

```
$black : ■ #323232;
$grey: ■ #c7c7c7;
@mixin btnNone{
  background: inherit ;
  border:none; box-shadow:none;
  border-radius:0; padding:0;
  overflow:visible; cursor:pointer;
}
```

자주 사용되는 색상을 변수로 지정하여 사용하였고,
버튼의 기본 스타일 속성값을 지우기 위한
특정 속성값을 @mixin을 통해 지정하였다.

코드 - JQuery

```
class PdMake {  
    constructor(name, price) {  
        this.name = name;  
        //제품 이미지를 담을 배열  
        this.pdImg = [];  
        //동그라미 버튼을 누르면 이미지 바뀌게 할 때 쓰는 버튼 아이디를 모은 배열  
        this.imgCircle = ["cirOne", "cirTwo", "cirThr"];  
        //가격 원데이터  
        this.price = price;  
        this.commaPrice = price;  
    }  
  
    pdPriceSetting() {--  
    }  
  
    firstSettingPrice() {--  
    }  
  
    //000 , 정규식  
    commaSetting() {--  
    }  
  
    listClickEvent() {--  
    }  
  
    //제품 옵션에서 선택한 항목이 제일 위칸에 적용되게 한다.  
    selectTop() {--  
    }  
  
    defaultTop(){--  
    }  
  
    //제품 이미지 배열 생성  
    imgMake() {--  
    }  
  
    //제품 이미지 하단 버튼 메소드?  
    cirImg() {--  
    }  
  
    totalMakePd() {--  
    }  
}  
  
const lipAmy = new PdMake("amy", 45000);  
const lipCoral = new PdMake("coral", 25000);
```

Class를 사용하여 각 상품 옵션의 값이 유지되도록 new로 인스턴스를 개별 생성하였다. (lipAmy, lipCoral)

이미지 관련 메소드와 제품 옵션을 바꿀 때 마다 발생해야하는 이벤트를 Class의 메소드로 만들어 활용하였다.

코드 - JQuery

```
//장바구니 추가 클로저
function basketFnc(timesArg , number){
    k=0;
    let innerBasketFnc =() => {
        basName = $("#pdSelect").text();
        basCount = $("#productCount").val();
        (number)&&(k=number);
        ($("#pdSelect").text()) == ($("#firstSelect").text())&&(basPrice=lipAmy.price)&&lipAmy.listClickEvent();
        ($("#pdSelect").text()) == ($("#secondSelect").text())&&(basPrice=lipCoral.price)&&lipCoral.listClickEvent();
        let onePara = '<ul class="pdBasket delRemove'+k+'"><li class="basketName">'+basName+'</li><li class="basketCou
        let twoPara = (basCount * basPrice)+'<span>원</span></li><li><a><i class="fas fa-times-circle inTimesBtn'+k+' d
        $(".modalForm").append(onePara+twoPara);
        k++;
    }
    return innerBasketFnc;
}
let basketHandler = basketFnc();
```

선택한 제품을 장바구니에 개별적으로 담기게 하기 위해서 클로저를 별도로 만들어 사용하였다.

클로저가 사용될 때의 값을 화면에서 가져와서 장바구니 모달안에 넣어준다.

코드 - JQuery

```
//장바구니 삭제 버튼
$(document).on("click", '.delBtn', ()=>{
    let delArr = (event.target.classList);
    delArr = $.isNumeric(delArr[2].slice(-2));
    !(delArr)&&(delArr = (event.target.classList[2].slice(-1))) || (delArr = (event.target.classList[2].slice(-2)));
    let delMinusWon = ($.basketPrice+delArr).text().slice(0,-1);
    delMinusWon = delMinusWon.replace(/,/g, '');
    let delTotal = parseInt($.basketTotal).text().replace(/,/g, '');
    let replDelTotal = String(delTotal-delMinusWon).replace(/B(?:=\d{3})+(?!d))/g, "");
    ($.basketTotal).text(replDelTotal);
    $(".delRemove"+delArr).remove();
});
```

장바구니에 추가된 제품 목록에서 x버튼을 누를 경우 장바구니에서 해당 제품이 삭제되게 한다.

목록에 있는 태그의 클래스 리스트를 배열에 담아준 후 slice를 사용하여 클릭된 제품이 몇 번째 줄의 제품인지 확인한 후, remove메소드를 사용하여 해당 줄을 삭제한다.

코드 - JQuery

```
//장바구니 모달
$("#inputBasket").on("click",()=>{
    event.preventDefault();
    $("#pdSelect").nextAll().css({display:"none"});
    $(".modalBottom").show();
    basketHandler();
    ($("#pdSelect").text() == ($("#firstSelect").text())&&(lipAmy.firstSettingPrice()));
    ($("#pdSelect").text() == ($("#secondSelect").text())&&(lipCoral.firstSettingPrice()));
    let noCommaPrice = $('.basketPrice'+j).text().replace(/,/g, '');
    ($('.basketTotal').text() == 0 )&& (tmpTotalV = 0);
    tmpTotalV = parseInt(noCommaPrice.slice(0,-1)) + parseInt(tmpTotalV);
    let commaBasketPrice = ($('.basketPrice'+j).text().replace(/\B(?=(\d{3})+(?!\d))/g,","));
    $('.basketPrice'+j).text(commaBasketPrice);
    replTotalV = String(tmpTotalV).replace(/\B(?=(\d{3})+(?!\d))/g,"");
    $('.basketTotal').text(replTotalV);
    j++;
    return false;
})
```

장바구니 버튼을 클릭하면 장바구니 모달이 나타나고, 그 값을 유지하기 위한 클로저를 사용한 후, 어떤 제품이 선택되었는지 확인한다.

그리고 선택된 제품의 가격을 생성한 인스턴스의 메소드로 제품 화면의 합계 금액을 초기값으로 바꾼다.

마지막으로 정규식을 사용하여 ,를 추가한 후 장바구니에 들어간다.

코드 - JQuery

```
//제품 옵션 선택 목록 클릭시
$("li").on("click", function () {
    switch (event.target.id) {
        //제품 종류선택시
        case "firstSelect":
            lipAmy.totalMakePd();
            $("#pdSelect").nextAll().css({
                display: "none"
            });
            break
        case "secondSelect":
            lipCoral.totalMakePd();
            $("#pdSelect").nextAll().css({
                display: "none"
            });
            break
    }
});
```

제품의 옵션에서 해당 옵션을 클릭할 경우,
해당 인스턴스와 메소드를 사용하여
화면에 나타나는 제품의 이미지와 가격 등을 다르게 하고,
제일 위의 박스를 제외하고 나머지는 사라지게 하기 위해서
nextAll().css()를 사용하여 display를 none으로 변경한다.

코드 - JQuery

```
//이미지 밑 원 버튼 호버시 클래스 변경 이벤트
function circleOver(targetOver) {
    $(targetOver).removeClass();
    $(targetOver).addClass("fas fa-circle");
}

function circleLeave(targetLeave) {
    $(targetLeave).removeClass();
    $(targetLeave).addClass("far fa-circle");
}

$(".circleBox i").hover(() => {
    switch (event.target.id) {
        case "cirOne":
            circleOver("#cirOne")
            break
        case "cirTwo":
            circleOver("#cirTwo")
            break
        case "cirThr":
            circleOver("#cirThr")
            break
    }
}, () => {
    switch (event.target.id) {
        case "cirOne":
            circleLeave("#cirOne")
            break
        case "cirTwo":
            circleLeave("#cirTwo")
            break
        case "cirThr":
            circleLeave("#cirThr")
            break
    }
});
```

제품의 이미지 밑에 있는 fontawsome을 사용하여 넣은 기호를 호버할 경우, 색이 바뀌는 것을 addClass() 메소드와 removeClass() 메소드로 클래스를 지우고 추가하여 불러오는 기호가 달라지게 한다.

해당 이벤트에 적용되는 내용은 circleOver함수와 circleLeave함수로 만들어 사용한다.

6 총평

처음 장바구니 페이지를 구상할 때에는 제품 목록 페이지부터 시작해서 객체를 이용하여 제품 추가 버튼까지 만들고 싶었지만 시간이 부족하여 한 제품의 페이지밖에 만들 수 없었다. 그래서 제품페이지에서 제품 추가를 할 수 있게 하고 싶었다. 하지만 그렇게 하기 위해서는 코드를 거의 새로 작성해야 했기 때문에 포기하였다. 맨 처음 계획과는 다르게 간단하게 수정하면서 체계적인 홈페이지 만들기가 이루어지지 않았다. 이번에도 역시 일정 계획 수립의 중요성을 느꼈다. 시간이 부족해서 몇가지 기능을 뺀 만큼 아쉬움이 있다. 그리고 제품을 장바구니에 담을 때 담긴 제품이 각각 들어가는 것만 생각해서 동일한 제품이 합쳐져야 한다는 것을 생각하지 못했다. 장바구니 페이지를 제시 시간에 구상하고 작성했다면 선생님의 조언을 듣고 수정을 할 수 있었을 텐데 하는 생각이 들었다. 앞으로는 주어진 시간 내에 홈페이지를 완성할 수 있도록 노력해야겠다.

감사합니다.