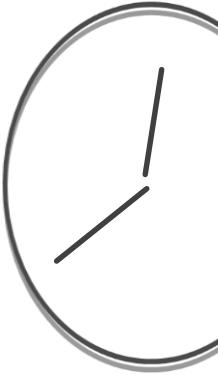


LOL Market PPT

지효남



INDEX

1. 프로젝트 목표
2. 프로젝트 계획
3. 일정표
4. 프로토타입
5. 웹페이지
6. 코드
7. 테스트 및 수정

1. 프로젝트 목표

1. 프로젝트 목표

쇼핑몰에서 보이는 장바구니를 구현해야 합니다.

반드시 10개 사면 할인 or 10개씩 구매 가 있어야 합니다.

옵션 선택이 있어야 하며 이에 따른 가격 변동이 필요합니다.

상품은 최소 3개 이상 있어야 합니다.

옵션이 선택됨을 눈으로 확인이 가능해야 합니다.

사용자 편의를 고려해야 합니다.

2. 프로젝트 계획

2. 프로젝트 계획

상품으로 유명 게임 리그오브레전드 줄임말로
통칭 **LOL(롤)**의 캐릭터 상품을
판매하기로 정했습니다.

사용자 분석 및 니즈 조사 결과 **10~20대가 주 사용자**이며,
다른 시리즈와 다르게 계속 나오고 있는 별 수호자를 선택했습니다.

계속 나온다는 건 다른 시리즈 보다 별 수호자가 잘 팔리는 게 아닐까 생각에서
별 수호자 콘셉트로 나온 4개의 상품을 선택했습니다.

웹에서는 오른손잡이가 평균적으로 높다는 통계에 따라
장바구니 창과 상품의 옵션 등을 오른쪽에 최대한 배치할 생각입니다.

3. 일정표

3. 일정표

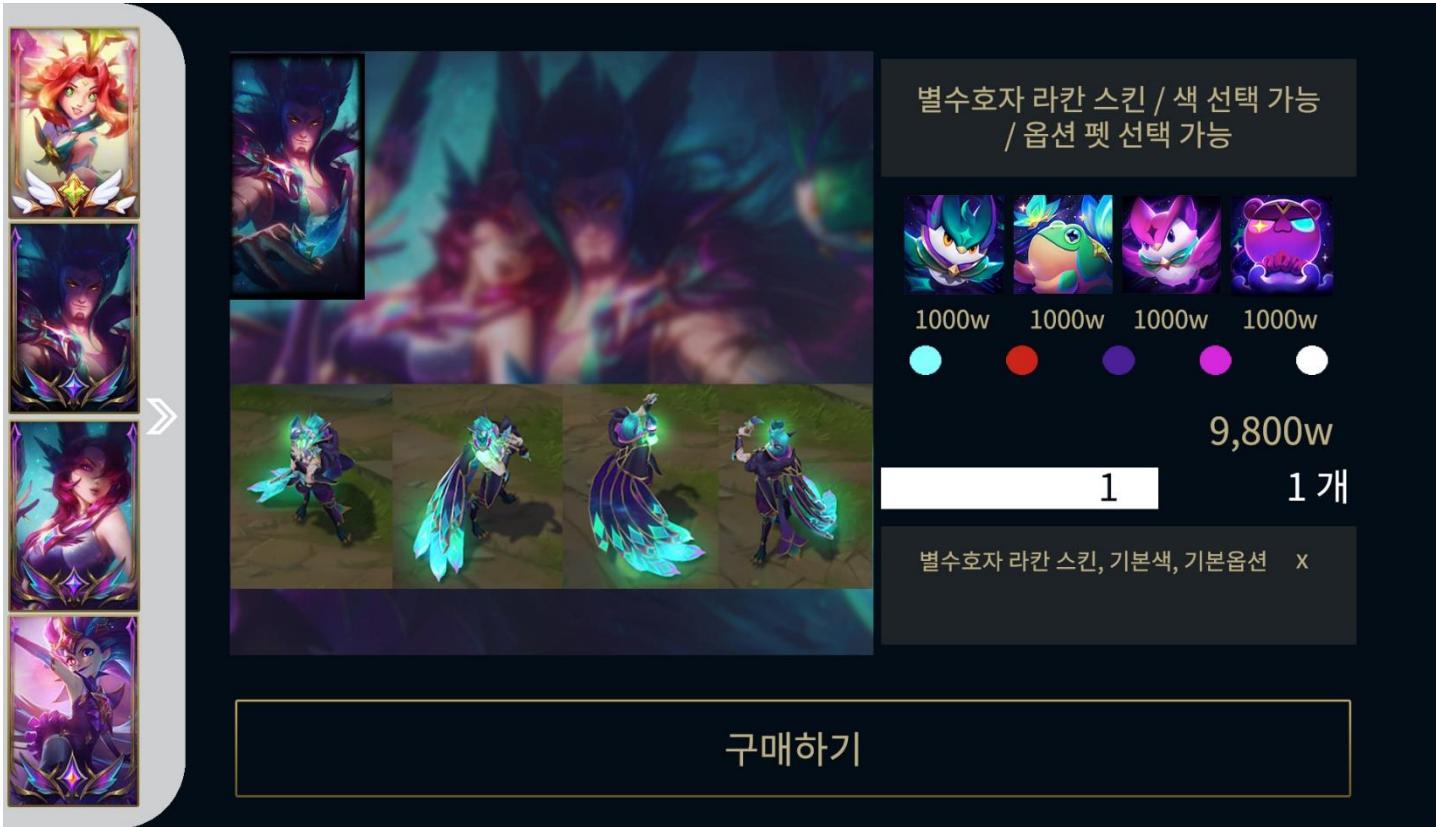
9/23 9/24 9/25 9/26 9/27
 9/30 10/1(마감)

※10/1 마감이었으나 ajax 사용을 위해 연장 요청
10/2(연장) 10/3(연장) 10/4(연장)

9/23 콘셉트 생각 및 스케치 시장조사	9/24 이미지 수집 색상 선택	9/25 프로토타입 제작	9/26 HTML & SCSS 시작 & 완료	9/27 SCRIPT 시작 & 완료
9/30 오류수정	10/1 오류수정 및 문서작업	10/2 HTML & SCSS & SCRIPT 완료	10/3 AJAX 추가 PHP 추가	10/4 오류수정 및 문서작업

4. 프로토타입

4. 프로토타입



유명 게임 LOL Market 콘셉트를 잡았습니다.

옵션 선택과 컬러 선택 수량 기능을 카트 페이지에 맞게 생각해보았습니다.

10개와 옵션 추가 구매 시 9000원 할인 기능을 추가할 생각입니다.

내비게이션은 클릭 시 나오는 형태로 진행할 생각입니다.

5. 웹페이지

5. 웹페이지

별수호자 라칸 / 색 선택 가능 / 아이콘 선택 가능

Normal Red Blue Purple

1,000w 1,000w 1,000w 1,000w

1 1개 9,800w 담기

※ 컬러 선택은 무료입니다.
캐릭터 아이콘은 1회구입당 한개만 가능합니다.
별수호자는 (최소) 1개 ~ 10개 구매가 가능하며
1개당 9,800원입니다.
*아이콘과 *별수호자 *10개를 구매하면 9,000원 할인됩니다.

장바구니 열기

최종 완성본입니다.
우선 내비게이션 부분입니다.
호버 됨을 알려드리기 위해 호버 시 이미지가 오른쪽으로 살짝 이동합니다.

5. 웹페이지

자세히보기

장바구니 열기

별수호자 자야 / 색 선택 가능 / 아이콘 선택 가능

Nomal Red Blue Purple

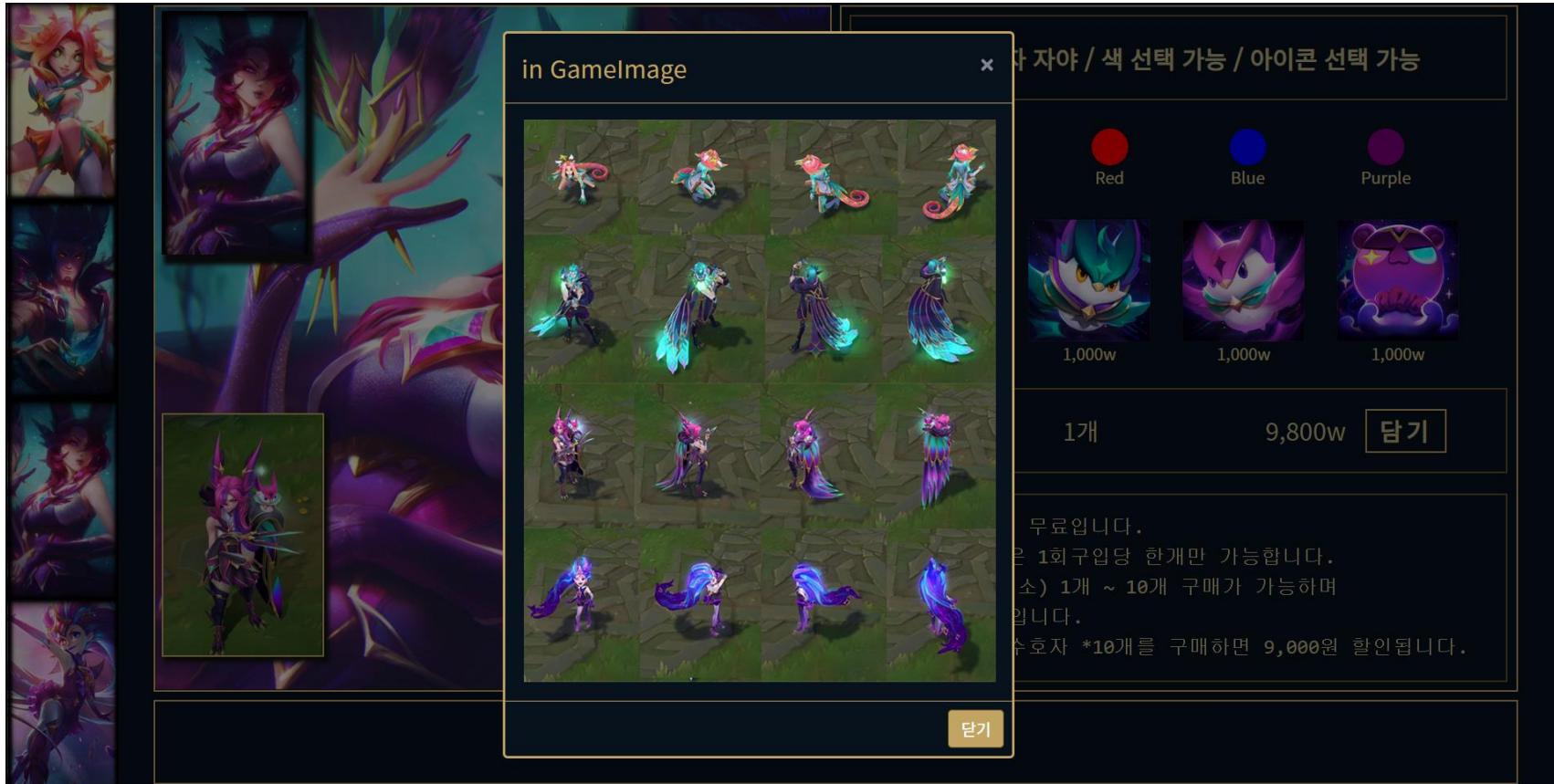
1,000w 1,000w 1,000w 1,000w

1 1개 9,800w 담기

※ 컬러 선택은 무료입니다.
캐릭터 아이콘은 1회구입당 한개만 가능합니다.
별수호자는 (최소) 1개 ~ 10개 구매가 가능하며
1개당 9,800원입니다.
*아이콘과 *별수호자 *10개를 구매하면 9,000원 할인됩니다.

내비게이션 클릭 시
해당 상품으로 이미지 교체와
상품 타이틀이 바뀌게 됩니다.

5. 웹페이지



자세히 보기 클릭 시
상품의 인 게임 사진을
좀 더 다양하게 볼 수 있습니다.

5. 웹페이지

※ 컬러 선택은 무료입니다.
캐릭터 아이콘은 1회구입당 한개만 가능합니다.
별수호자는 (최소) 1개 ~ 10개 구매가 가능하며
1개당 9,800원입니다.
*아이콘과 *별수호자 *10개를 구매하면 9,000원 할인됩니다.

장바구니 열기

컬러 클릭 시
상품의 인 게임 색상이 왼쪽 하단에서 바뀌게 됩니다.
색상 선택은 무료이며, 상품 1개당 9,800원입니다.

5. 웹페이지

자세히보기

장바구니 열기

별수호자 라칸 / 색 선택 가능 / 아이콘 선택 가능

Nomal Red Blue Purple

Click 1,000w 1,000w 1,000w

1 1개 10,800w **담기**

※ 컬러 선택은 무료입니다.
캐릭터 아이콘은 1회구입당 한개만 가능합니다.
별수호자는 (최소) 1개 ~ 10개 구매가 가능하며
1개당 9,800원입니다.
*아이콘과 *별수호자 *10개를 구매하면 9,000원 할인됩니다.

옵션 클릭 시
왼쪽에 옵션 이미지가
추가가 되며 가격이 1,000원 올라갑니다.

5. 웹페이지

자세히보기

장바구니 열기

별수호자 라칸 / 색 선택 가능 / 아이콘 선택 가능

Nomal Red Blue Purple

1,000w 1,000w 1,000w 1,000w

10 10개 90,000w 담기

※ 컬러 선택은 무료입니다.
캐릭터 아이콘은 1회구입당 한개만 가능합니다.
별수호자는 (최소) 1개 ~ 10개 구매가 가능하며
1개당 9,800원입니다.
*아이콘과 *별수호자 *10개를 구매하면 9,000원 할인됩니다.

수량은 1~10개까지 가능하며
10개 + 옵션 선택 시 9,000원 할인됩니다.

5. 웹페이지

장바구니
별수호자 니코 / 색 선택 가능 / 아이콘 선택 가능

상품선택

rakan/nomalColor//9800w
9,800w

nico/nomalColor/jayaPet/10800w
10,800w

20,600w

선택삭제 쇼핑계속하기 상품구매하기

별수호자는 (최소) 1개 ~ 10개 구매가 가능하며 1개당 9,800원입니다.
*아이콘과 *별수호자 *10개를 구매하면 9,000원 할인됩니다.

장바구니 열기

상품 선택 후 담기 버튼을 누르면
장바구니에 담기며
삭제, 계속, 구매, 담기 버튼을 배치했습니다.
장바구니는 닫으면 하단에 열기 버튼으로 다시 열수 있습니다.

6. 코드

6. 코드

```
<!-- 자세히보기 모달 -->


<button class="modalBtn" data-toggle="modal" data-target="#myModal">자세히보기</button>
    <div class="modal fade" id="myModal">
        <div class="modal-dialog modal-dialog-centered">
            <div class="modal-content modalControl">
                <div class="modal-header">
                    <h4 class="modal-title inGameTitle">in GameImage</h4>
                    <button type="button" class="close" data-dismiss="modal">&times;</button>
                </div>
                <div class="modal-body modalBodySize">
                    
                </div>
                <div class="modal-footer">
                    <button type="button" class="btn btn-danger modalCloseBtn" data-dismiss="modal">닫기</button>
                </div>
            </div>
        </div>
    </div>


```

자세히 보기 버튼 클릭 시 나오는 Modal만 부트스트랩을 사용하였습니다.

4. 코드

```
//이미지의 URL 이 계속 변경되므로 함수로 처리 바꿔는 URL을 계속 변수로 가져올수 있음
//백그라운드 이미지로 주었음
function goodsImgChange (ChangeImgUrl_1, ChangeImgUrl_2, ChangeImgUrl_3)
{
    $("#goodsImg").css("background-image", "url('./img/" + ChangeImgUrl_1 + '/' + ChangeImgUrl_2 + '/' + ChangeImgUrl_3 + "')");
}

//src 자체를 바꾸어 변경함 goodsImgChange와 같은 이유로 이미지 경로를 추출해옴
function mainGoodsImgChange (folderUrl, fileUrl)
{
    /* ./img/rakan/rakan_Test.jpg */
    $("#jayaRakanBackImg").attr("src", "./img/" + folderUrl + '/' + fileUrl);
}
```

자주 바뀌는 이미지 박스의 백그라운드 url 추출입니다.
밑에는 자주 바뀌는 이미지 링크의 src 추출입니다.

6. 코드

```
//상품추가
//script로 처리 가능하지만 ajax가 어떤건지 알고 싶어서 사용해봄
//타이틀,컬러,옵션 등 php로 값을 보내서 처리한후 값을 받아옴
function ProductAdd (P_Title, P_Color, P_Option, P_Value)
{
    //장바구니에 이미지를 넣기위해서 이미지 경로를 추출함
    var str = "./img/" + P_Title + "/" + P_Title + "_1.jpg";

    //만약 값이 없다면 공백일때고 공백이라면 - 로 처리
    var optionString = P_Option;
    (optionString != "") || (optionString = " - ");

    $.ajax({
        url:"./php/LOLmarket.php",
        dataType: 'json',
        type: 'POST',
        data: [
            name: P_Title,
            color: P_Color,
            option: optionString,
            Value: P_Value,
            src: str
        ],
        success: function(result)
        {
            //가격을 변수에 담아 append로 다시 끼워넣을 생각
            var productPrice = $("#sumPrice").html();
            // console.log(result['string']);
            //console.log(productPrice);

            //장바구니에 이미지, 선택한 옵션 컬러 가격을 append를 사용해 끼워넣음
            $("#myCartCard").append(
                <div class="myProduct">
                    <label><input type="checkbox" class="deleteCheck" id="deleteCheck" name="deleteCheck" value="deleteCheck">
                    ${result['string']}</p>
                    <p class="cartSelectPrice" id="cartSelectPrice">${productPrice}</p></label>
                </div>
            );
            // console.log("rottin", $(".cartSelectPrice").length);
            // console.log($(".cartSelectPrice")[1]).html();
            let i = 0;
            let cartSumPrice = 0;

            while(i < $(".cartSelectPrice").length)
            {
                cartSumPrice = $(".cartSelectPrice")[i];
                i++;
            }

            setTotalPrice();
        }
    });
}
```

마감 기간이 늘어나서

AJAX를 사용하고 싶었습니다.

장바구니 콘텐츠를 새로 만들고 AJAX를

추가했습니다.

값이 잘 전달되고 받아왔다면

JSON 형태의 값을 통해 장바구니에

append 하여 태그와 값을 같이 넣어

장바구니에 상품이 담기도록 하였습니다.

6. 코드

```
//장바구니에 있는 가격을 변수에 담고  
//가격에 콤마와 원이라는 문자를 공백으로 없앰  
function setTotalPrice()  
{  
    var resultPrice = 0;  
    var resultPriceString = "";  
  
    var i = 0;  
  
    while(i < $(".cartSelectPrice").length)  
    {  
        let tempPrice = $(".cartSelectPrice")[i].html();  
        tempPrice = tempPrice.replace(",","");  
        tempPrice = tempPrice.replace("원","");  
  
        //콤마와 원을 없앤 가격을 넘버로 정확히 찢어주고 이를 리질트에 누적시킴  
        resultPrice += Number(tempPrice);  
        i++;  
    }  
    // 정확히 숫자가 된 resultPrice 를 콤마찍기 정규식으로 콤마를 생성함  
    resultPriceString = numberWithComma(resultPrice);  
    // html에 출력  
    $("#lastPrice").html(resultPriceString + "원");  
}
```

AJAX success 구문 마지막에 있는
function입니다.

태그의 개수만큼 while 문을 반복합니다.
가격이 변수에 담기며 replace로, 와 “원”을
제거하여 갱신합니다.

더한 값을 Number로 숫자로 바꾸어줍니다.
마지막으로 commaF 함수로,를 추가해주고
“원”도 붙여서 장바구니에 출력합니다.

6. 코드

```
//수량카운트
$("#goodsCount").change(function(e)
{
    //우선 수량은 0-9의 한자리 혹은 두자리의 숫자만 받을수 있게 정규식을 사용하여
    //변수에 담아주었고
    var countRegExp = /^[0-9]{1,2}$/;
    //상품개수를 변수에 담았으며
    P_Value = $("#goodsCount").val();
    //한개당 가격이 9800원이며 개수당 증가하는 가격을 변수에 담아주고
    P_Price = P_Value * 9800;

    //정규식을 통과하는지 검사한다
    if(countRegExp.test(P_Value))
    {
        //통과했다면 카운트박스에 몇개를 선택했는지를 알리기 위해 개수 + 문자 "개"를 html에 출력
        $("#CountBox").html(P_Value + "개");
        //개수의 총합에 문자열 "원"을 붙여 html에 출력
        $("#sumPrice").html(numberWithComma(P_Price) + "원");

        //다만 개수가 10개 미하라면
        if(P_Value > 10)
        {
            //10개를 숫자로 확실하게 변경해주고
            P_Value = Number(10);
            //그 숫자당 9800원입을 변수에 담아준다
            P_Price = P_Value * 9800;

            $("#goodsCount").val(P_Value);
            $("#CountBox").html(P_Value + "개");
            $("#sumPrice").html(numberWithComma(P_Price) + "원");

            //다만 옵션이 있을경우
            if( P_option == "nikoPet" || P_option == "rakanPet" || P_option == "jayaPet" || P_option == "joyPet")
            {
                //개수에 9800원이고 옵션값 1000원을 추가함 이를 변수에 담아줌
                P_Price = (P_Value * 9800) + 1000;
                //담은 값에 문자 "원"을 추가해서 html에 출력
                $("#sumPrice").html(numberWithComma(P_Price) + "원");
                //다만 변수에 담긴 값이 99000원이라면
                //즉 10개 구매하고 9800 * 10 = 98000 에 옵션 추가한 1000 = 99000 이라면
                if(P_Price == 99000)
                {
                    //10개에 옵션까지 추가했으니 9000원 할인해줌
                    P_Price = (((P_Value * 9800) + 1000) - 9000);
                    $("#sumPrice").html(numberWithComma(P_Price) + "원");
                }
            }
        }
    }
})
```

수량 카운트입니다.

우선 수량은 0-9의 한자리 혹은 두 자리의 숫자만
받을 수 있게 정규식을 사용했습니다.

한 개당 가격은 9800원이며

개소당 증가하는 가격을 변수에 담아서 사용했습니다.
정규식을 통과한다면 카운트 박스에 몇 개를
선택했는지를 알리기 위해 개수 + 문자 "개"를
html에 출력해줍니다.

개수의 총합에 문자열 "원"을 붙여 html에 출력
다만 개수가 10개 이상이라면

개수를 10개로 고정해줍니다 10개 이상은 구매가 불가능
하도록 하기 위해서입니다.

옵션이 있는지 유무를 확인하고 옵션이 있다면
금액에 1000원이 추가되며

10개를 구매하고 옵션까지 추가한다면
9000원 할인을 해줍니다.

6. 코드

```
//체크박스 전체선택하기에 반응이 온다면
$("#allDeleteCheck").change(function(e)
{
    //isAllCheck가 트루라면
    if(isAllCheck == true)
    {
        var i = 0;
        //체크박스의 개수만큼
        while(i < $(".deleteCheck").length)
        {
            //체크박스를 전부 펄스로 바꿈
            $($(".deleteCheck")[i]).prop("checked", false);
            i++;
        }
        isAllCheck = false;
    }
    //펄스라면
    else
    {
        var i = 0;
        //체크박스의 개수만큼
        while(i < $(".deleteCheck").length)
        {
            //체크박스를 전부 트루로 바꿈
            $($(".deleteCheck")[i]).prop("checked", true);
            i++;
        }
        isAllCheck = true;
    }
});
```

체크박스 전체선택하기에 반응이 온다면
isAllCheck가 트루라면
체크박스의 개수만큼
체크박스를 전부 펄스로 바꿈
만약 펄스상태라면
체크박스의 개수만큼
체크박스를 전부 트루로 바꿈

6. 코드

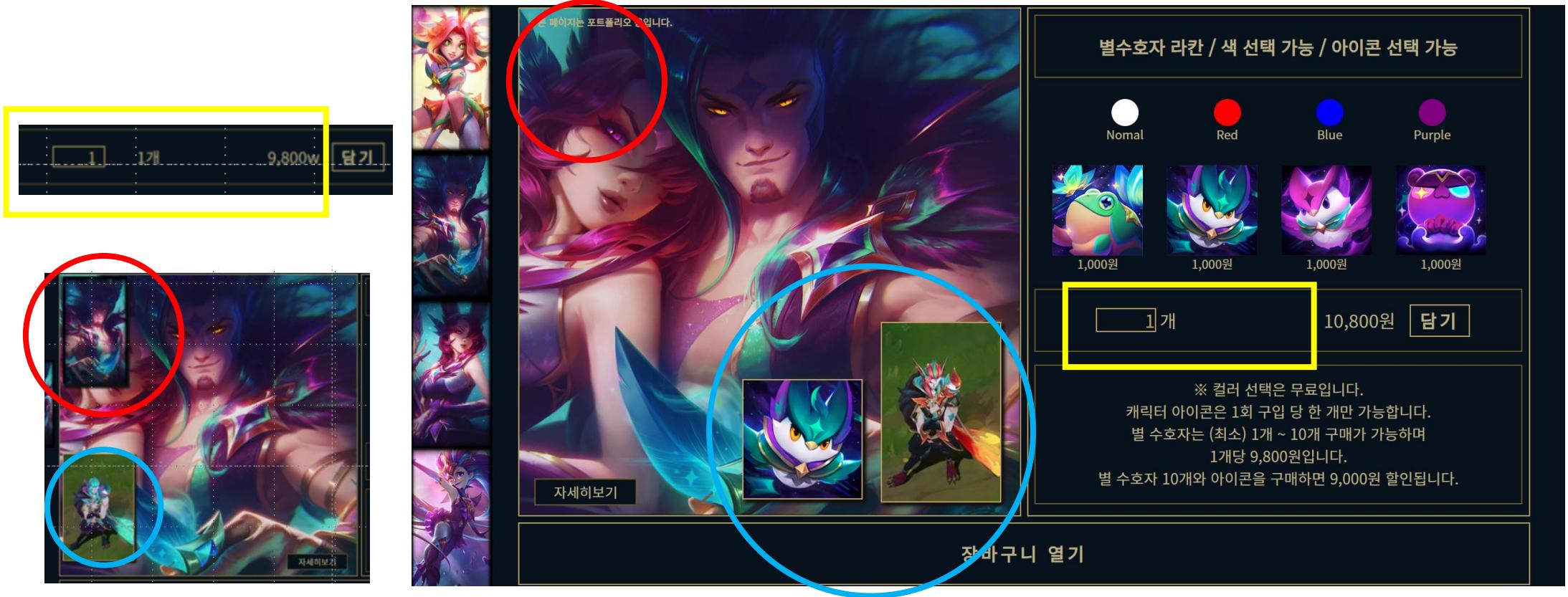
```
//개별 체크박스가 클릭된다면
$("#selectDelete").click(function(e)
{
    var i = 0;
    var deleteArr = $(".deleteCheck");
    //체크박스의 갯수만큼 반복
    while(i < $(".deleteCheck").length)
    {
        if($(".deleteCheck[i]").prop("checked") == true)
        {
            $($(".deleteCheck")[i]).attr("checked", "checked");
        }

        i++;
    }
    //체크박스가 체크라면 태그의 부모의 부모값을 삭제합니다.
    $(".deleteCheck[checked='checked']").parent().parent().remove();
    setTotalPrice();
    resetCheckBox();
});
});
```

선택 삭제 버튼 클릭 시 태그의 개수만큼 반복되지만
체크가 되어있다면 태그의 체크박스를 체크함
체크박스가 체크 라면 태그의 부모의 부모 값을 삭제합니다.

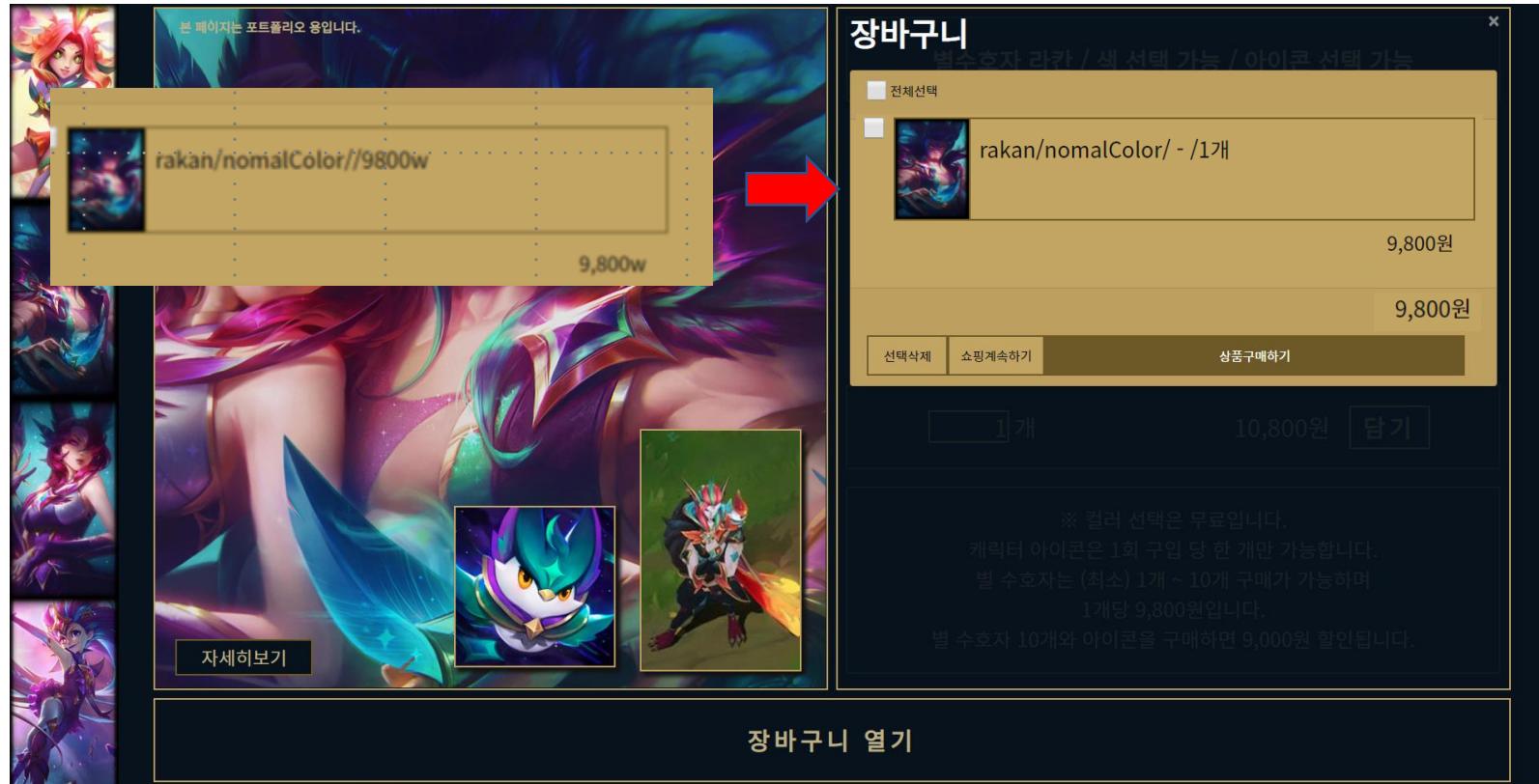
7. 테스트 및 수정

7. 테스트 및 수정

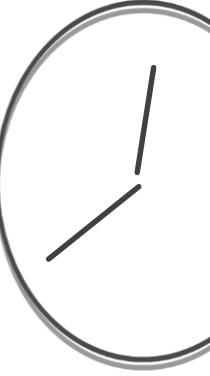


일러스트 상단에 어떤 상품을 선택했는지 이미지를 한 장 더 추가하였으나
너무 많은 이미지로 정보의 질이 떨어지는 효과를 보여 삭제하였습니다.
또한 1개를 고르고 1개가 또 표시되었는데 이 또한 정보의 질이 떨어지는 효과를 보여
삭제하고 인풋칸에 개수 표시를 붙여주었습니다. 특이하게 'w' 단위를 사용했지만
'원' 단위로 바꾸었습니다.

7. 테스트 및 수정



옵션을 선택하지 않으면 // 식으로 공백이었으나 / - / '-' 을 추가했습니다.



감사합니다.

ST23 지효남