

PORTFOLIO

신입 개발자

이 윤 나

yoonnable@gmail.com

contents

1. 자기소개
2. 프로젝트 1
 - 소개
 - 핵심기능
3. 프로젝트 2
 - 소개
 - 핵심기능

1. 자기소개 신입개발자 이윤나

contact

Tel. 010.7509.5507
Email. yoonnable@gmail.com

license

GTQ 그래픽기술자격 1급

education

Kh정보교육원 강남지원 - 자바프로그래머 과정

skills

- Language



- DataBase



- WAS



- Framework



- Etc



2. 프로젝트 - 소개 1. 온라인 쇼룸이 마련된 가구쇼핑몰

프로젝트 설명

- 사용자가 구매한 상품을 조회할 수 있다.
- 배송 완료된 상품을 교환,반품신청 및 구매확정 할 수 있다.
- 구매확정한 상품은 자동으로 구매 수량만큼 출고되어 재고가 업데이트되며, 사용자는 리뷰 작성이 가능하고 작성한 형태에 따라 포인트가 자동 적립된다.
- 관리자는 취소, 교환, 반품 신청을 확인 후 거절, 승인을 하면 자동으로 회원에게 메일이 발송된다.
- 관리자는 배송상태를 일괄 변경 할 수 있다.
- 관리자는 배송 시 회원이 입력한 모든 정보 수정이 가능하다.
- 상품을 등록 시 입력한 재고만큼 입고 내역이 저장된다.
- 상품 재고를 변경할 시 입고고 내역이 저장된다.
- 모든 화면은 반응형을 적용시켰다.

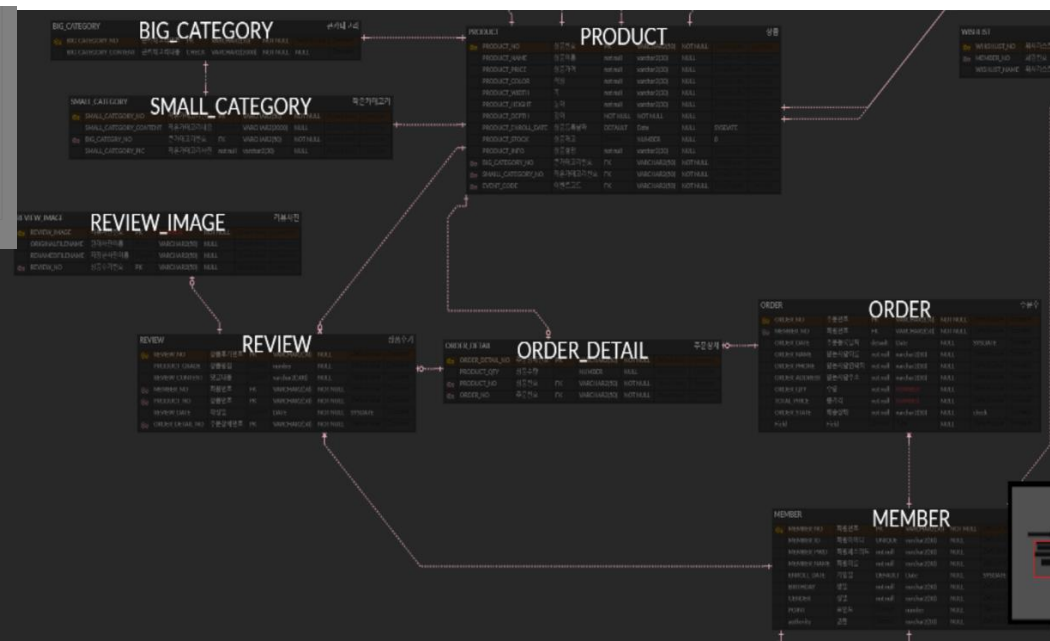
기술스택

- Language : Java, HTML5, CSS3, Javascript, JQuery
- DB : Oracle DB
- Server : Apache Tomcat 8.0
- Tools : STS
- Operating System : Windows OS

프로젝트 기간 : 2020.12.10 ~ 2021.01.11

프로젝트 내용

- 스프링 프레임워크를 교육받으며 수행하였다.
- <https://github.com/ui0426/Orkney>



2. 프로젝트 – 핵심기능 입출고관리-DB

```
--상품 재고 UPDATE -> 입출고 INSERT
--트리거 생성 : 업데이트 재고가 업데이트 전 재고보다 클때 업데이트-업데이트 전, '입고', 업데이트 전 재고가 업데이트 후 재고보다 클때 업데이트-업데이트 후, '출고'
CREATE TRIGGER TR_PRO_STOCK_UPDATE_SHIPPED
AFTER UPDATE OF PRODUCT_STOCK ON PRODUCT
FOR EACH ROW
BEGIN
    IF :NEW.PRODUCT_STOCK > :OLD.PRODUCT_STOCK
    THEN
        INSERT INTO SHIPPED VALUES(SH_SEQ.NEXTVAL, :OLD.PRODUCT_NO, SYSDATE, :NEW.PRODUCT_STOCK - :OLD.PRODUCT_STOCK, '입고');
    END IF;
    IF :NEW.PRODUCT_STOCK < :OLD.PRODUCT_STOCK
    THEN
        INSERT INTO SHIPPED VALUES(SH_SEQ.NEXTVAL, :OLD.PRODUCT_NO, SYSDATE, :OLD.PRODUCT_STOCK - :NEW.PRODUCT_STOCK, '출고');
    END IF;
END;
/
```

```
--상품 등록할 때 수량 입출고 테이블에 입고 INSERT
--트리거생성
CREATE TRIGGER TR_PRO_INSERT_SHIPPE
AFTER INSERT ON PRODUCT
FOR EACH ROW
BEGIN
    INSERT INTO SHIPPED VALUES(SH_SEQ.NEXTVAL, :NEW.PRODUCT_NO, SYSDATE, :NEW.PRODUCT_STOCK, '입고');
END;
/
```

```
--구매확정 하면 PRODUCT에 있는 재고를 구매한 상품 만큼 감소해서 UPDATE
--트리거생성
CREATE TRIGGER TR_OD_SORT_PRO
AFTER UPDATE OF SORT ON ORDER_DETAIL
FOR EACH ROW
WHEN (NEW.SORT = '구매확정')
BEGIN
    UPDATE PRODUCT SET PRODUCT_STOCK = PRODUCT_STOCK - :OLD.PRODUCT_QTY WHERE PRODUCT_NO = :OLD.PRODUCT_NO;
END;
/
```

입출고 내역 저장

- 배송완료 시 사용자가 상품의 구매확정을 하면 자동으로 상품재고가 구매한 수량만큼 줄고, 입출고내역에 출고일자와 수량이 자동으로 저장된다.
- 관리자가 상품을 등록 시 입력한 재고가 입출고 내역에 자동으로 입고일자와 수량이 저장된다.
- 관리자가 상품의 수량을 임의로 변경 시 증가감소에 따라 입출고 내역에 자동으로 일자와 수량이 저장된다.

DB를 상품테이블의 정보가 변경 된 후 입출고테이블이 변경되도록 설계한 관계로,

주문한 상품의 정보가 담긴 order_detail 테이블에 '구매확정'의 데이터로만 update 될 때 작동하는 트리거를 생성해 자동으로 product테이블에 재고를 구매한 수량만큼 감소한 수량으로 update하였고,

Product테이블에 재고 수량이 update되면, 기존 수량보다 갱신된 수량이 클 시 '입고' 그 반대는 '출고' 로 입출고내역에 insert하는 트리거를 생성하여 자동으로 입출고내역이 무결성을 지키며 저장할 수 있도록 하였다.

2. 프로젝트 – 핵심기능 리뷰 등록 (1)-JavaScript

//이미지 미리보기

function handleFiles(file, name){//매개변수로 input태그에 담기는 첨부파일과 파일경로~이름~확장명을 데리고 온다.

console.log("파일경로 이름 확장명 : "+name);

var _lastDot = name.lastIndexOf('.');//뒤에서부터 확장자까지의 자릿수

var ext = name.substring(_lastDot, name.length);//여기까진 확장자 분리하기

ext = name.substring(_lastDot, name.length).toLowerCase();//확장자 무조건 소문자로 변경

//이미지 파일인지 확실하게 확인

if(ext.indexOf('.gif')>-1 || ext.indexOf('.png')>-1 || ext.indexOf('.jpg')>-1 || ext.indexOf('.jpeg')>-1){
file.closest("div").setAttribute("class","none");//input태그가 있는 div는 숨기기

const img = document.createElement("img");

file.parentNode.parentNode.nextSibling.nextSibling.setAttribute("class","review-file");

//이미지 넣을 div에 클래스 설정

file.parentNode.parentNode.nextSibling.nextSibling.appendChild(img);

img.src = URL.createObjectURL(file.files[0]);

img.setAttribute("style","position:absolute; width:100%; height:100%;");

//input태그에 담은 이미지 img태그에 담기 성공

//첨부파일 삭제 버튼보임

//=====여기까지 이미지 미리보기 끝=====

else{
alert("이미지 파일이 아니다");

}

};

첨부파일 미리보기

- 클라이언트가 리뷰를 등록할 때 사진을 3장 이하 첨부할 수 있다.
- 사진을 첨부할 시 업로드 전 미리보기가 가능하다.
- 첨부된 사진을 지우고 다른 사진을 넣을 수도 있다.

타입이 file인 input태그에 파일을 담을 때, 이미지 파일만 담을 수 있도록 파일명의 확장자를 확인하는 분기문을 작성하였다.

확장자 확인 후 이미지파일이면 기존에 input태그가 있던 곳에 img태그를 생성하고, 그 이미지 위로 input태그에 담은 파일을 지울 수 있는 버튼도 생성했다.

2. 프로젝트 – 핵심기능 리뷰 등록 (2)-Jquery

```
//미리보기 이미지 삭제하기
function deletePreview(btn, i){//매개변수로 삭제버튼 자신과 id값에 해당하는 번호(총 3개의 첨부파일을 등록할 수 있도록 해둠)
    var div = btn.parentNode;//삭제버튼과 미리보기 이미지를 담고있는 div
    var img = btn.parentNode.lastChild; //이미지 태그
    img = img.parentNode.removeChild(img);//미리보기 이미지 태그는 완전히 제거
    div.setAttribute("class","none");//미리보기 이미지 담았던 div숨기기

    $(btn).parent().prev().find("input").val("");//근접요소 탐색으로 해당 input태그에 접근해 value값을 지운다
    $(btn).parent().prev().find("input").prop("type","text");//input태그의 타입을 한번 바꾸어 value값을 확실히 지운다
    $(btn).parent().prev().find("input").prop("type","file");//그리고 다시 파일을 첨부할 수 있도록 다시 타입을file로 바꾼다

    var box = "#img-box"+i;
    $(box).removeClass("none").addClass("filebox").addClass("review-file");//input태그의 부모태그 div를 다시 보이게 한다
    //다시 새로운 input 태그 생성
}
```

첨부파일 미리보기 삭제, 값 지우기

- 클라이언트가 리뷰작성 시 이미지를 첨부하여 미리보기로 확인 후 잘못 올렸을 경우 등록 전 첨부한 이미지 파일을 취소할 수 있다.

이미지 미리보기에 있는 삭제버튼을 클릭 시 img태그를 지우고 파일을 첨부하기 전에 있던 input태그를 다시 나타낸다.

Input태그에 담겨진 값을 지워야 하는데, 일반적으로 타입이 text같은 input태그의 값을 지우듯 value=""으로 하면 타입이 file인 input태그의 값은 지워지지 않는다. 그래서 input태그의 타입을 아예 text로 한번 바꿨다가 다시 file로 되돌리는 방식으로 값을 깔끔하게 지워서 미리보기에서 파일을 지운 후 리뷰를 등록 시 서버 단으로 넘어가는 파일이 없도록 하였다.

2. 프로젝트 – 핵심기능 리뷰 등록 (3)-MultipartFile

```
//by온나-리뷰 작성
@RequestMapping(value="/review/reviewInsert.do", method = RequestMethod.POST)
public String insertReview(Review review, Model m, @RequestParam(value="review_img", required=false) MultipartFile[] multi,
    HttpSession session ) throws IOException{

    Map login = (Map)session.getAttribute("login");
    review.setMember_no((String)login.get("MEMBER_NO"));
    Map data = new HashMap(); //포인트 적립 정보 담을 것
    data.put("point", 200);
    data.put("reason", "리뷰작성(텍스트형)");

    //저장경로 지정
    String path=session.getServletContext().getRealPath("/resources/upload/review");
    File dir = new File(path);
    if(!dir.exists()) dir.mkdirs();

    List<ReviewImage> files=new ArrayList();

    for(MultipartFile f : multi) {
        System.out.println("file "+f);
        if(!f.isEmpty()) {
            String originalName = f.getOriginalFilename();
            String ext = originalName.substring(originalName.lastIndexOf(".") + 1);
            System.out.println(ext);

            SimpleDateFormat sdf=new SimpleDateFormat("yyyyMMdd_HHmssSSS");
            int rndValue = (int)(Math.random()*10000);
            String reName=sdf.format(System.currentTimeMillis())+"_"+rndValue+"."+ext;
            try {
                f.transferTo(new File(path+"/"+reName));
            }catch(IOException e) {
                e.printStackTrace();
            }
            ReviewImage ri=ReviewImage.builder().originalFileName(originalName).renamedFileName(reName).build();
            files.add(ri);
            data.put("point", 500);
            data.put("reason", "리뷰작성(사진형)");
        }
    }
    int result=service.insertReview(review, files);
}
```

파일업로드

- 첨부사진을 서버의 알맞은 경로에 업로드 한다.
- 클라이언트가 입력한 별점, 리뷰내용, 첨부 사진을 DB에 REVIEW테이블에 저장한다.
- 텍스트형 리뷰를 작성했을 경우 200포인트를, 사진형 리뷰를 작성했을 경우 500포인트를 각각 자동으로 적립한다.

별점과 리뷰내용은 commend객체로 받아온다. 그리고 포인트 적립을 위해 우선 텍스트형일 때 200포인트 적립할 데이터를 담는다.

첨부파일은 3장 이하로 받아 올 수 있기 때문에 다중파일 업로드를 고려해 MultipartFile[]로 배열 형식으로 파일을 받아온다.

받아온 파일이 있을 경우 파일을 저장할 경로를 지정해준다.

배열로 받아온 파일들을 forEach문을 통해 각각 저장하는데, 저장하기 전 파일에 유일성을 주기 위해 저장할 때는 임의로 겹치지 않는 이름으로 변경해준다.

지정해준 경로와 변경한 이름으로 파일을 저장한 후, 본래의 파일 이름과 변경한 파일 이름도 DB에 저장할 수 있도록 객체에 담는다.

사진형 리뷰는 500포인트를 적립하도록 분기문 내에서 값을 다시 담아준다.

리뷰와 리뷰사진 객체에 담은 데이터를 DB에 저장한다.

2. 프로젝트 – 핵심기능 리뷰 삭제

```
//by윤나-리뷰삭제
@RequestMapping("/admin/deleteReview")
public ModelAndView deleteReview(@RequestParam(value="rNo") int rNo, @RequestParam(value="mNo") String mNo,
    @RequestParam(value="riName",required=false) String[] riName, HttpSession session) {
    ModelAndView mv = new ModelAndView();
    boolean result=false;
    int pResult;
    Map data = new HashMap();
    data.put("no", mNo);
    data.put("type", "자감");
    data.put("reason", "리뷰삭제");
    int point;
    //서버에 저장된 파일 삭제하기
    String path=session.getServletContext().getRealPath("/resources/upload/review");//파일 저장된 경로
    if(riName!=null) {
        point = 500;//사진형은 500포인트 차감
        for(int i=0; i<riName.length; i++) {
            File file = new File(path+"/"+riName[i]);
            if(file.exists()) {
                result=file.delete();
            }else {
                result = true;
            }
        }
    }else {
        point = 200;//텍스트형은 200포인트 차감
        result = true;
    }

    String msg="";
    String loc="";
    //DB에 저장된 리뷰, 리뷰사진 데이터 삭제
    if(result==true) {
        if(service.deleteReview(rNo)>0) {
            data.put("point", point);
            pResult = service.modifyPoint(data);//포인트 차감
            msg="리뷰가 삭제되었습니다. 회원포인트 : "+point+"자동 차감 완료.";
            loc="/admin/review.do";
        }
    }else {
        msg="삭제가 완료되지 않았습니다.";
        loc="/admin/reviewDetail.do?rNo="+rNo;
    }
    mv.addObject("msg", msg);
    mv.addObject("loc", loc);
    mv.setViewName("/common/msg");
    return mv;
}
```

파일삭제

- 관리자가 리뷰를 관리할 수 있다.
- 사진이 첨부된 리뷰 삭제를 할 때 파일도 삭제하여 불필요한 용량사용을 줄인다.
- 리뷰 타입에 따라 작성시 적립된 포인트를 삭제 시 자동으로 회수한다.

첨부된 사진들의 rename을 String[]로 서버단에 받아온다.

첨부파일이 저장되어 있는 경로와 첨부파일 이름을 합친 후 그 경로에 동일한 이름의 파일이 있는지 확인 후 파일이 존재하면 그 파일을 삭제한다.

2. 프로젝트 – 핵심기능 교환, 반품신청-Ajax

```
var data = new FormData();
data.append("order_detail_no",odNo);
data.append("sort",sort);
data.append("refund_qty", rQty);
data.append("refund_reason", rReason);
data.append("refund_content", rContent);
data.append("file",rPic[0] );

$.ajax({
  type: "POST",
  enctype: 'multipart/form-data',
  processData: false,
  contentType: false,
  url : "${path}/order/updateRefund.do",
  data : data,
  success: data => {
    if(data == '교환신청'){
      $("#result-msg").html("교환 신청이 접수되었습니다.");
      $(".r-request-1").css("display","none");
      $(".r-request-2").css("display","block");
    }
    if(data == '반품신청'){
      $("#result-msg").html("반품 신청이 접수되었습니다.");
      $(".r-request-1").css("display","none");
      $(".r-request-2").css("display","block");
    }
    else{
      $("#result-msg").html("에러가 발생하였습니다. 고객센터에 문의바랍니다.");
    }
  }
})
})
```

Ajax 파일업로드

- 클라이언트가 상품하자로 인해 교환, 반품신청을 할 때 상품하자 부분의 사진을 필수로 첨부해야 한다.
- 교환,반품 신청 시 뜨는 모달에서 품을 작성하고 신청하면 새로운 화면으로 전환 하지 않고 동일한 모달에서 신청이 완료되었다는 안내문구가 나온다.

파일을 포함한 데이터들을 form태그 submit이 아닌 ajax로 전송하기 위해 FormData객체를 생성한다.

Append로 file형식의 데이터까지 담아서 form태그로 전송한다.

이때, 속성에 **enctype: 'multipart/form-data'**, **processData: false**, **contentType: false** 이 세가지를 꼭 작성해야 한다.

서버에서 값을 받을 땐 form태그를 submit하는 것과 동일하게 **commend**객체와 **MultipartFile**객체로 받아오면 된다.

3. 프로젝트 - 소개

· 프로젝트 소개

- › 비회원, 회원 모두 상품을 구매할 수 있다.
- › 결제 시 가용포인트를 사용할 수 있다.
- › 사용자가 구매한 상품을 조회할 수 있다.
- › 배송 완료된 상품을 환불신청 및 구매확정을 할 수 있다.

· 기술스택

- › Language : Java, HTML5, CSS3, Javascript, JQuery
- › DB : Oracle DB
- › Server : Apache Tomcat 8.0
- › Tools : Eclipse
- › Operating System : Windows OS

· 프로젝트 기간 : 2020.09.29 ~ 2020.10.16

· 프로젝트 내용

- › 첫 프로젝트이다.
- › <https://github.com/yoonnable/SemiProject>

결제하기

주문자

이름 김민지 연락처 0101010

배송 정보

이름 이준영 입학생번호 2020-12-05 연락처 01010-0000-0000 주소지역 우송대학

2020-12-05 도착예정



삼겹살 돼지고기 먹각구이

가격:22800원

수량: 1 개

총액 :22800

쿠폰 적용

00원 할인쿠폰

포인트적용

현재 가용 포인트 : 9726

0 point

결제 정보

상품금액 22800원

배송비 3000원

할인 금액

쿠폰할인 0원

포인트사용 Point

결제 금액 25800 원

이동약관에 동의합니다.

개인정보취급 동의약관동의 동의(항시)동의不同意

동의

결제하기


주문 / 배송 상세내역 조회	
주문자 정보	
이름 : 임	
연락처 : 1111	
배송정보	
배송지 : 강남 여수시 가곡길 6(여수	
결제완료 : 2020-10-16	
결제정보	
결제수단 : htm5_jinics	총 상품금액 : 57400원
	총 배송비 : 3,000원
	총 결제금액 60400

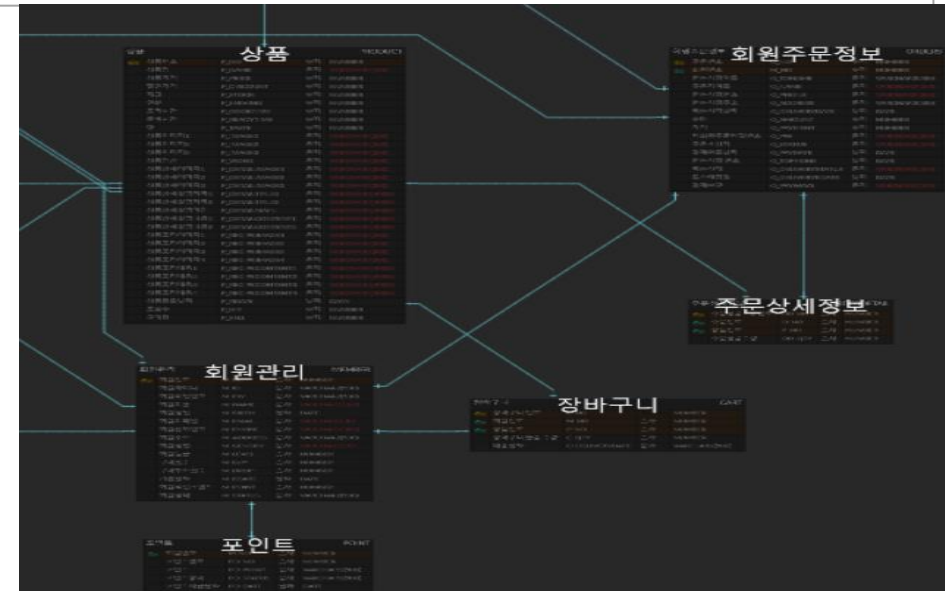
2020-10-16 (10091)

구매확정

2020-10-19 00:00:00.0 도착예정

배송조회

	<p style="color: #e91e63; font-weight: bold;">단호박 치즈오리불고기</p> <p>28800원 / 수량 : 1</p>	<div style="background-color: #e91e63; color: white; padding: 5px 10px; border-radius: 5px;">리뷰하기</div>
	<p style="color: #e91e63; font-weight: bold;">시골치 프리타타</p> <p>15800원 / 수량 : 1</p>	<div style="background-color: #e91e63; color: white; padding: 5px 10px; border-radius: 5px;">리뷰하기</div>
	<p style="color: #e91e63; font-weight: bold;">논골치즈왕갈비</p> <p>24800원 / 수량 : 1</p>	



2. 프로젝트 - 핵심기능 결제-API

```
IMP.request_pay({
  pg: 'html5_inicis', // 위에서 선택한 결제수단이 여기 값으로 들어와야함
  pay_method: 'card',
  merchant_uid: 'merchant_' + new Date().getTime(),
  amount: $("#oPayment").val(),
  buyer_name: $("#oName").val(),
  buyer_tel: $("#oPhone").val(),
}, function(rsp) {
  if ( rsp.success ) {
    //[1] 서버단에서 결제정보 조회를 위해 jQuery ajax로 imp_uid 전달하기
    jQuery.ajax({
      url: "/payments/complete", //cross-domain error가 발생하지 않도록 동일한 도메인으로 전송
      type: 'POST',
      dataType: 'json',
      data: {
        imp_uid : rsp.imp_uid
      }
    }).done(function(data) {
      //[2] 서버에서 REST API로 결제정보확인 및 서비스루틴이 정상적인 경우
      if ( everything_fine ) {
        var msg = '결제가 완료되었습니다.';
        msg += '\n고유ID : ' + rsp.imp_uid;
        msg += '\n상점 거래ID : ' + rsp.merchant_uid;
        msg += '\n결제 금액 : ' + rsp.paid_amount;
        msg += '카드 승인번호 : ' + rsp.apply_num;
        alert(msg);
      } else {
        alert('결제가 취소처리 되었습니다. ');
      }
    });
  } else {
    var msg = '결제에 실패하였습니다.';
    msg += '에러내용 : ' + rsp.error_msg;
    alert(msg);
  }
});
$("#pay").css("display","none");
$("#orderCheak").css("display","block");
```

```
function p_use(){
  const usep = document.getElementById('usePoint').value;
  document.getElementById("checkPoint").innerText = usep;
  const befor = document.getElementById('beforpay').value;
  const pay = Number(befor)-Number(usep);
  document.getElementById("oPayment").value = pay;
}
```

결제

- 회원은 가용포인트 내에서 포인트를 사용할 수 있다.
- 사용할 포인트를 입력 후 적용하면 사용할 포인트와 남은 포인트를 바로 확인 할 수 있다.
- 클라이언트가 실제로 결제할 수 있도록 한다.

회원은 DB에서 가용포인트 데이터까지 가져와서 화면에 출력한다.

사용할 포인트를 입력하면 바로 화면에 사용할 포인트만큼 할인된 금액과 포인트를 사용한 후 남은 포인트를 확인할 수 있다.

아임포트 결제API테스트용을 이용해 실제로 결제가 이루어 지도록 필요한 데이터들을 클라이언트가 입력한 값들로 받아서 API를 적용하였다.

감사합니다

Name : 이윤나

Tel : +82 10 7509 5507

Email : yoonnable@gmail.com

Git : <https://github.com/yoonnable>

Web : <https://yoonnable.github.io/>