# 개인 프로젝트 Market

다양한 상품을 판매하는 마켓과 JWT토큰을 이용한 설문조사 페이지

2021.06.01 박정훈

#### 프로젝트 간단 소개

#### 이러한 언어, 프레임워크와 데이터베이스를 사용하였습니다.

사용언어 - JAVA, thymeleaf, JS, CSS 서버 - Apache Tomcat 9.0v

빌드 - Gradle

프레임 워크 - Spring Framework 데이터베이스 - MySQL

#### 기능



#### 회원가입

암호화를 이용하여 유저 정보 저장



#### 검색기능

언제나 원하는 품목 검색가능



#### 설문조사

구매 후, JWT 토큰을 이용한 설문조사



### 목차

#### 1. 모델링

프로젝트의 ERD, 마인드맵을 통한 데이터 흐름을 파악할 수 있습니다.

#### 2.화면구현, 설명

간단한 화면 구현과 설명입니다.

#### 3.코드 설명

어떤식으로 코드를 작성했는지에 대한 설명입니다.

#### 4.일정

프로젝트 일정을 정리했습니다.

#### 프로핔

#### SKILL

JAVA - 클래스 객체 지향

프로그래밍

- 상속, 인터페이스

- GUI 스윙

- JDBC 연동

**웹 HTML** - HTML 마크업

- HTML 4.x 표준태그

- HTML5 신규 태그

- Fill-out Form 태그

- Form 입출력 처리

(파일업로드 FormData)

- CSS

데이터베이스

아파치 Tomcat

- 스키마 정의(테이블, 컬럼, 대응관계

MySQL, 오라클 제약조건-명시-키, 옵션)

- ERD(객체 관계 다이어그램) 표현

- MySql DBMS, SQL

- 설치, 설정, DML(스키마 구조)

- DML(CRUD - insert/select/update/

delete) 처리

- Workbench, phpmyadmin 등

- server.xml 설정

서블릿 컨테이너 - 서블릿 구동

통합개발환경(IDE) - 이클립스

- 안드로이드 스튜디오

- 인텔리제이

- 비주얼 스튜디오

- Java, js 등 개발 가능

언제나 문제를 해결하고 노력하는 개발자가 되겠습니다.



- 박정훈

#### 박정훈 / jeonghoon park

1997.10.30 / 서울특별시

Tel. 010-2672-2963

Email. vvv6737@naver.com

서울특별시 송파구 가락동

#### **GRADUATION**

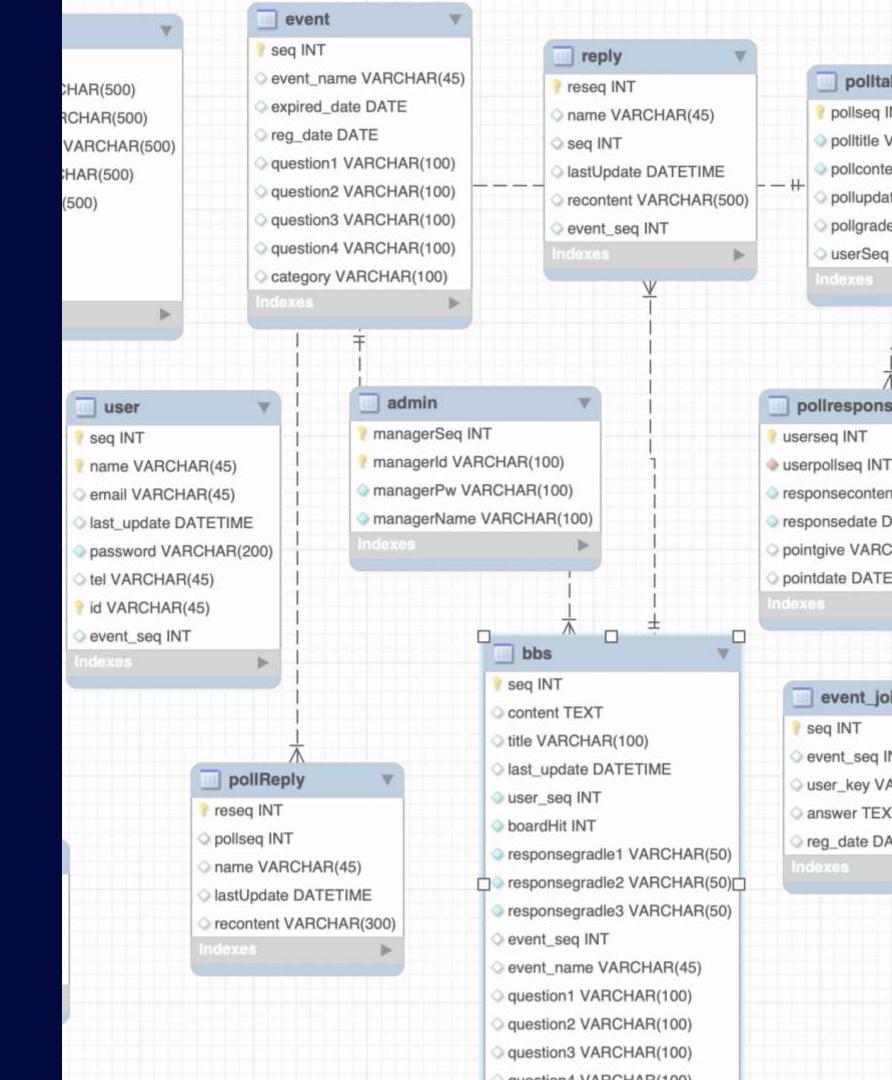
2016 서울고등학교 졸업

2018 아세아 항공 전문대 항공정비학과 졸업

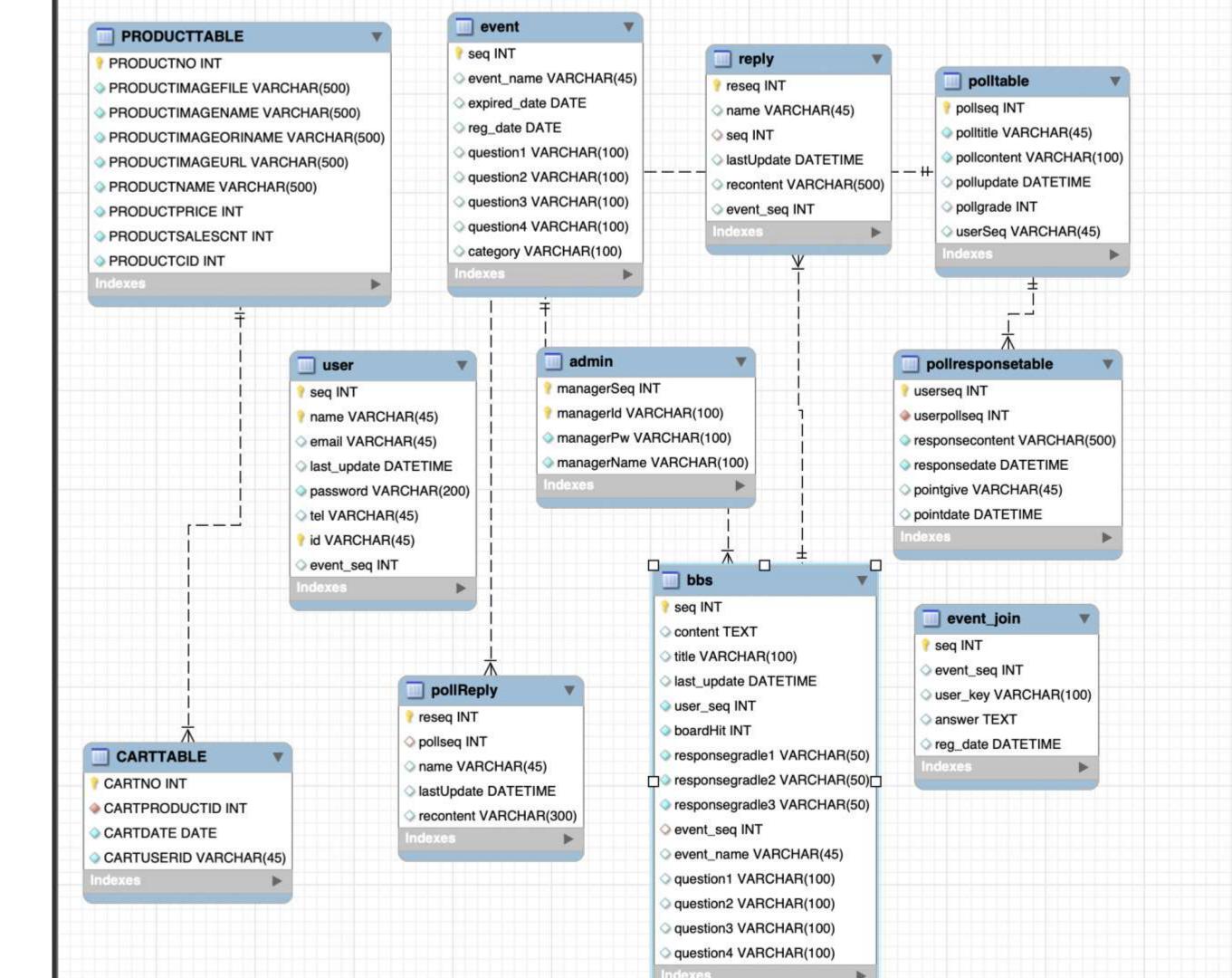
#### 1. 모델링

**ERD** 

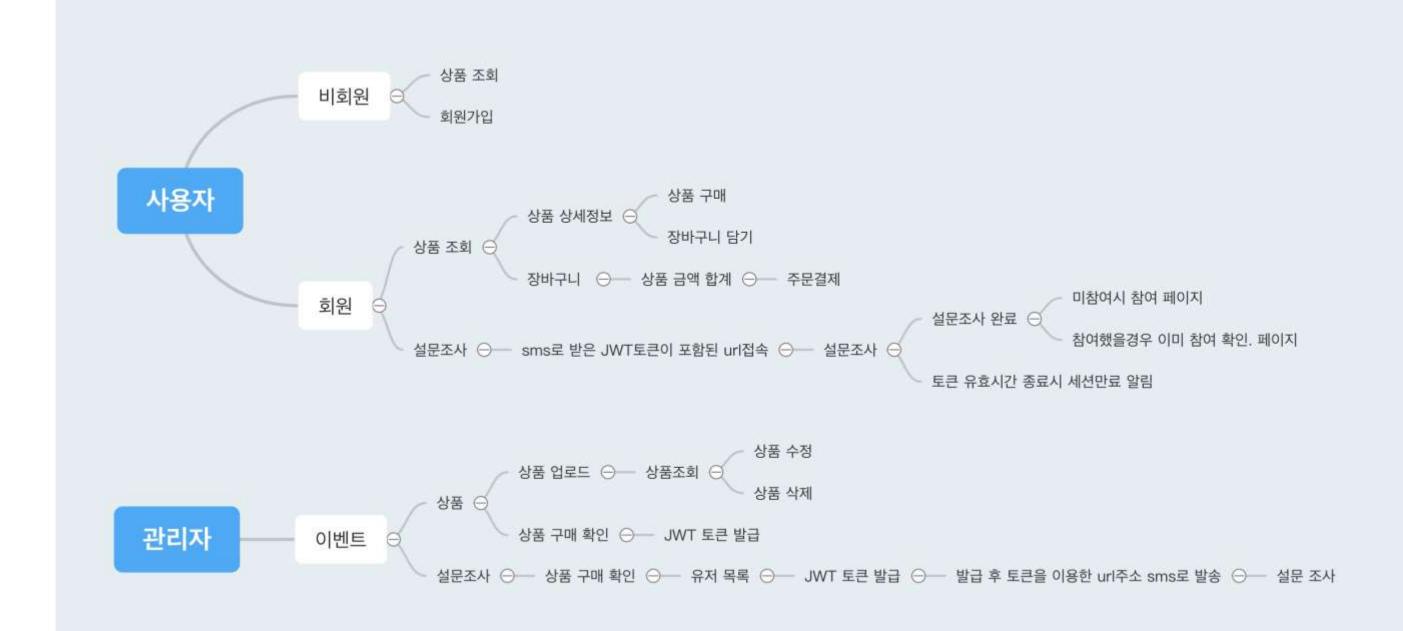
다이어그램



### 1. 모델링 ERD



## 1. 모델링 마인드 맵





PRODUCTS LISTS

RODUCTS LISTS

⁴ 상품목록

장바구니

SURVEY LISTS

표 설문 이벤트 목록



원하는 품목을 검색하세요.

#### 전체 상품

무드등



제품명 : 무드등가격 : 15000₩

〇 상품 상세보기

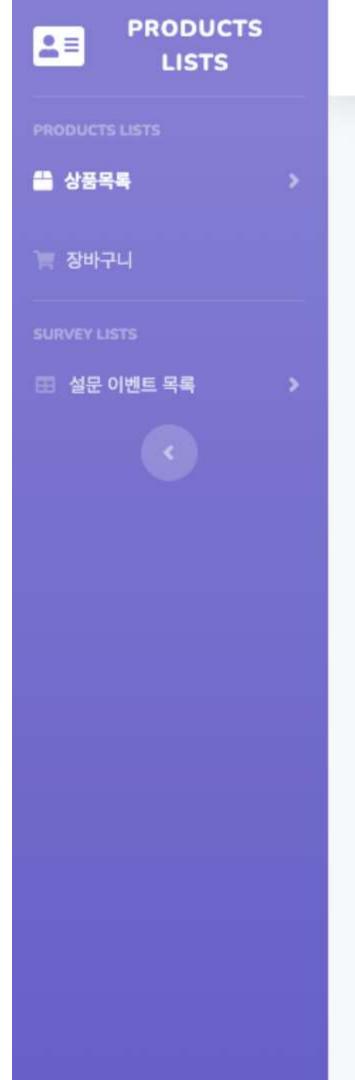
### 2. 화면구현

화면을 구현하여 해당 기능들을 설명합니다.

## 2. 화면구현 메인화면

메인화면 입니다. 아직 추가한 상품이 없으므로 관리자 페이지에서 추가를 해야 합니다.

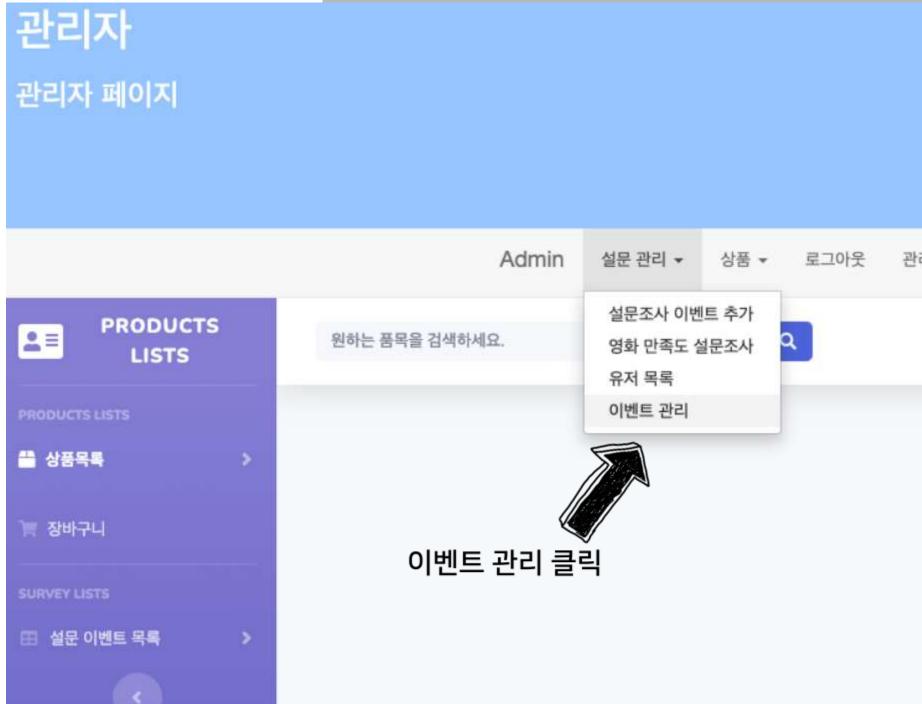
주소 /product/productlist



원하는 품목을 검색하세요.

주소 /adminindex 관리자 페이지로 오면 먼저 로그인을 해야합니다.





Admin 설문 관리 ▼ 상품 ▼ 로그아웃 관리자님 환영합니다.

#### ▲ 이벤트 목록

이벤트명

이벤트 종료일자

이벤트 생성

상세보기

이벤트 생성 클릭

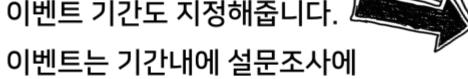
#### ✔ 이벤트 추가

이벤트 작성

카테고리 지정	추가할 카테고리를 입력하세요	
이벤트 제목	제목을 입력하세요	
질문1	질문을 입력하세요	
질문2	질문을 입력하세요	
질문3	질문을 입력하세요	
질문4	질문을 입력하세요	
이벤트 시작일자	연도. 월. 일:	3
이벤트 종료일자	연도. 월. 일:	t

이벤트 시작일자

이곳에서 상품의 카테고리, 이벤트 (설문조사)제목, 설문조사에 사용할 질문, 이벤트 기간도 지정해줍니다.



이벤트는 기간내에 설문조사에 참여할시 포인트를 지급해주는 것으로 가정합니다.

추가

이벤트 두개를 추가해 보겠습니다.

#### ✔ 이벤트 추가

이벤트 작성

	이벤트 종료일자
유용한 물건	
일반적인 설문조사	추가
상품은 만족하시나요?	
가장 만족한 사유를 선택한다	면?
주변에도 추천할 의향 있으신	가요?
기타 의견	
2021. 06. 16. 오후 01:50	
2021. 06. 23. 오후 01:50	
	일반적인 설문조사 상품은 만족하시나요? 가장 만족한 사유를 선택한다 주변에도 추천할 의향 있으신 기타 의견 2021. 06. 16. 오후 01:50

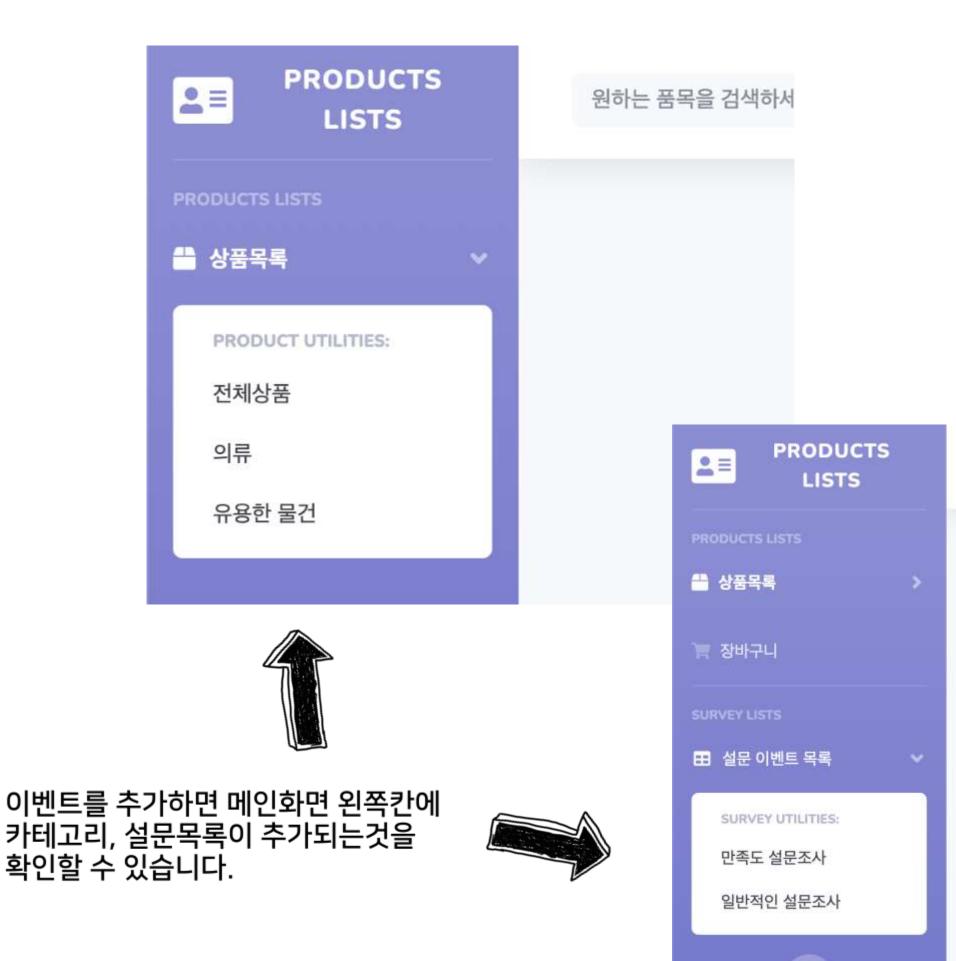


이벤트 작성

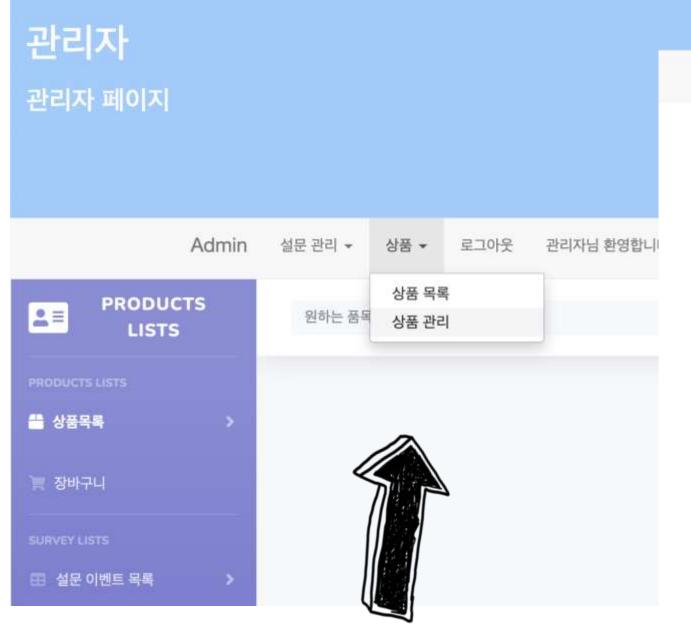
카테고리 지정	의류
이벤트 제목	만족도 설문조사
질문1	상품에 만족하셨나요?
질문2	상품의 가장 만족한 사유를 선택해주세요!
질문3	또 이용하시거나 주변에 추천해 주실 의향이 있으신가요?
질문4	기타 의견
이벤트 시작일자	2021. 06. 16. 오후 01:50
이벤트 종료일자	2021. 06. 23. 오후 01:50

### 2. 화면구현 메인페이지

주소 /product/productlist



원하는 품목을 감



다시 관리자 페이지로 돌아가서 판매할 상품을 추가해줍니다.

> 가격과 재고를 작성하고, 유저가 해당 상품을 구매 후 설문조사 참여 리스트를 받을곳을 지정합니다.



Admin 설문 관리 ▼ 상품 ▼ 로그아웃 관리자님 환영합니다.

#### 상품등록

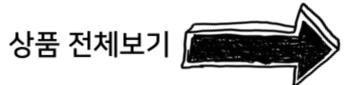
0 0

파일 선택 B8iGuv9Xibc0...OwwJHo.jpeg

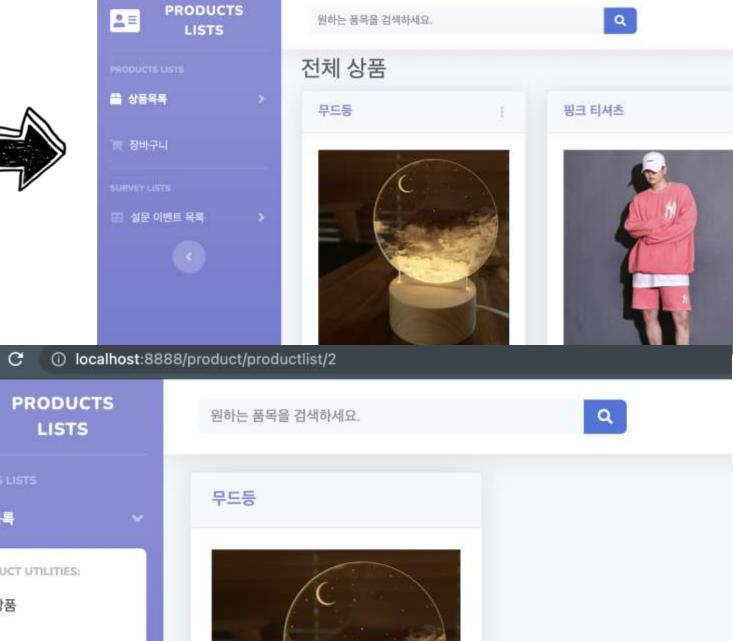


제 품 명		
무드등		
가격		
15000		
판매수량		
30		
종류 선택 의류 사건	선택	
✓ 유용한 물건		
작성 취소		

### 2. 화면구현 메인페이지



<u>.</u>=

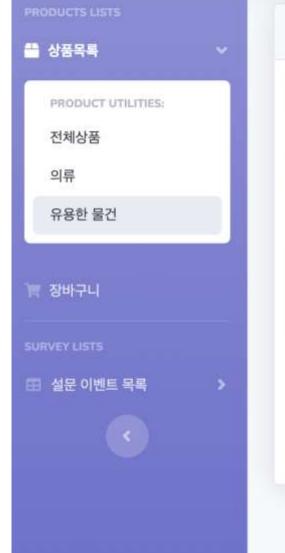


• 제품명:무드등

가격: 15000₩

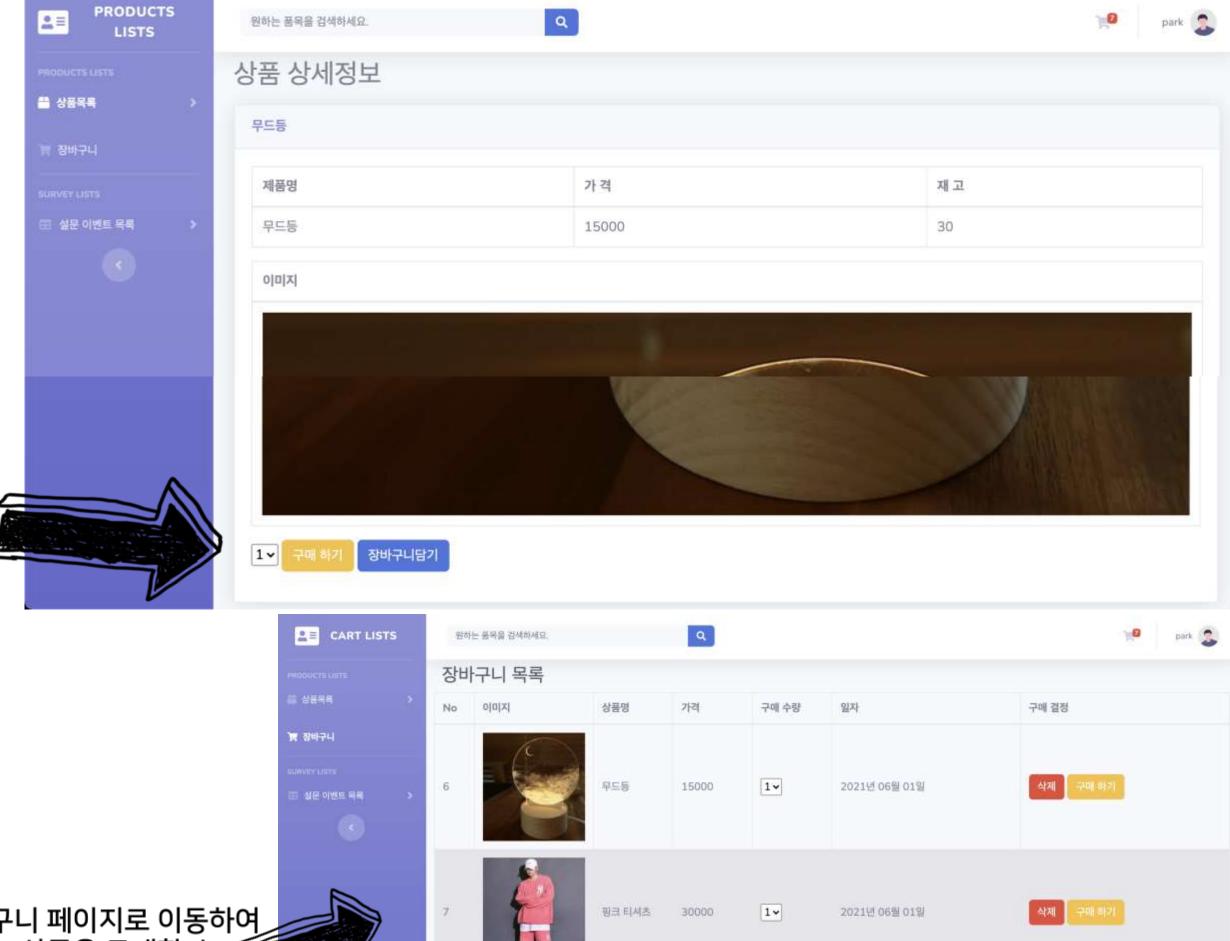
상품 상세보기

이벤트 업로드 때 지정했던 카테고리별 상품 조회



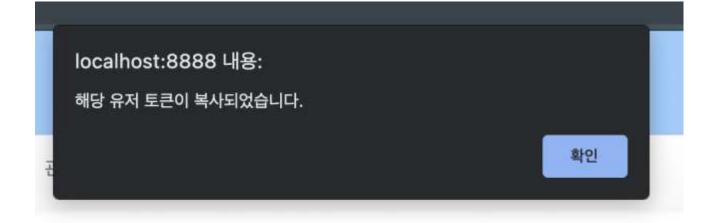
### 2. 화면구현 메인페이지

상품 상세정보를 통해 원하는 상품을 장바구니에 문 담을 수 있습니다.



장바구니 페이지로 이동하여 원하는 상품을 구매할 수 있습니다.

유저가 상품을 구매했다면 설문조사 진행을 위해 관리자 페이지 - 유저목록으로 가서 토큰을 발급합니다.



← → C ① localhost:8888/userlist

관리자 페이지

Admin 설문 관리 ▼ 상품 ▼ 로그아웃 관리자님 환영합니다.

해당유저 정보값을 가져와 토큰으로 발급 후 사용자에게 url주소를 보냅니다.

(sms로 보내는 것으로 가정)

시퀀스	아이디	성명	이메일	전화번호	가입일자	수정	삭제	토큰발급
18	qwe	jeong	qwe@daum.net	01026722963	2021년 04월 26일	수정	삭제	토큰발급
20	zxc	hoon	zxc@nate.net	01071113261	2021년 04월 26일	수정	삭제	토큰발급
22	123	park	123@naver.com	01012345678	2021년 04월 26일	수정	삭제	토큰발급
23	jkl	lee	jkl@gmail.com	01032617111	2021년 05월 06일	수정	삭제	토큰발급
24	456	kim	456@nate.net	01098765432	2021년 05월 06일	수정	삭제	토큰발급
25	park123	박정훈	vvv6737@naver.com	01026722963	2021년 05월 10일	수정	삭제	토콘발급

#### ₫ 계정 추가

♣ 유저목록

1.9	40	Joong	quoedudii.iiot	O I OLO I LLOUD	LULI L UTE LUE	10		
20	ZXC	hoon	zxc@nate.net	01071113261	2021년 04월 26일	수정	삭제	토큰발급
22	123	park	123@naver.com	01012345678	2021년 04월 26일	수정	삭제	토큰발급
23	jkl	lee	jkl@gmail.com	01032617111	2021년 05월 06일	수정	삭제	토큰발급
24	456	kim	456@nate.net	01098765432	2021년 05월 06일	수정	삭제	토큰발급
25	park123	박정훈	vvv6737@naver.com	01026722963	2021년 05월 10일	수정	삭제	토큰발급

#### 토큰이 발급되었습니다.

#### ▲ 계정 추가

아이디를 입력하세요. password 입력하세요. 이름을 입력하세요. email 입력하세요. 전화번호를 입력하세요. 저장 수정 로그인

#### 로 토큰 발급

eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzl1NiJ9.eyJldmVudFNlcSl6MCwibmFtZSl6InBhcmsiLCJ0ZWwiOilwMTAxMjM0NTY3OClsImlkIjoiMTlzliwiZXhwljoxNjlyNTl2OTc4LCJzZXEiOjlyLC

복人

### 2. 화면구현 설문조사 작성 페이지

sms으로 url을 받은 사용자는(가정) 해당 url로 접속하여 처음에 이벤트 생성때 설정 했던 질문을 받으면서 설문조사를 진행합니다.



① localhost:8888/surveylist/1/eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzl1NiJ9.eyJldmVudFNlcSl6MCwibmFtZSl6InBhcmsiLCJ0ZWwiOilwMTAxMjM0NTY3OCIs

#### ✔ 만족도 설문조사

이용해 주셔서 감사드립니다.

설문 조사에 참여해주시면 포인트를 드립니다.

\*포인트 지급기간: 즉시 지급

성명	park
Q1.	상품에 만족하셨나요? ○ 만족 ○ 보통 ○ 불만족
Q2.	상품의 가장 만족한 사유를 선택해주세요! ○ 가격 ○ 서비스 ○ 배송속도
Q3.	또 이용하시거나 주변에 추천해 주실 의향이 있으신가요? ○ 있음 ○ 보통 ○ 없음
기타 의견	전하고 싶은 말이 있으시다면 적어주세요 :)

작성

### 2. 화면구현 설문조사 목록 페이지

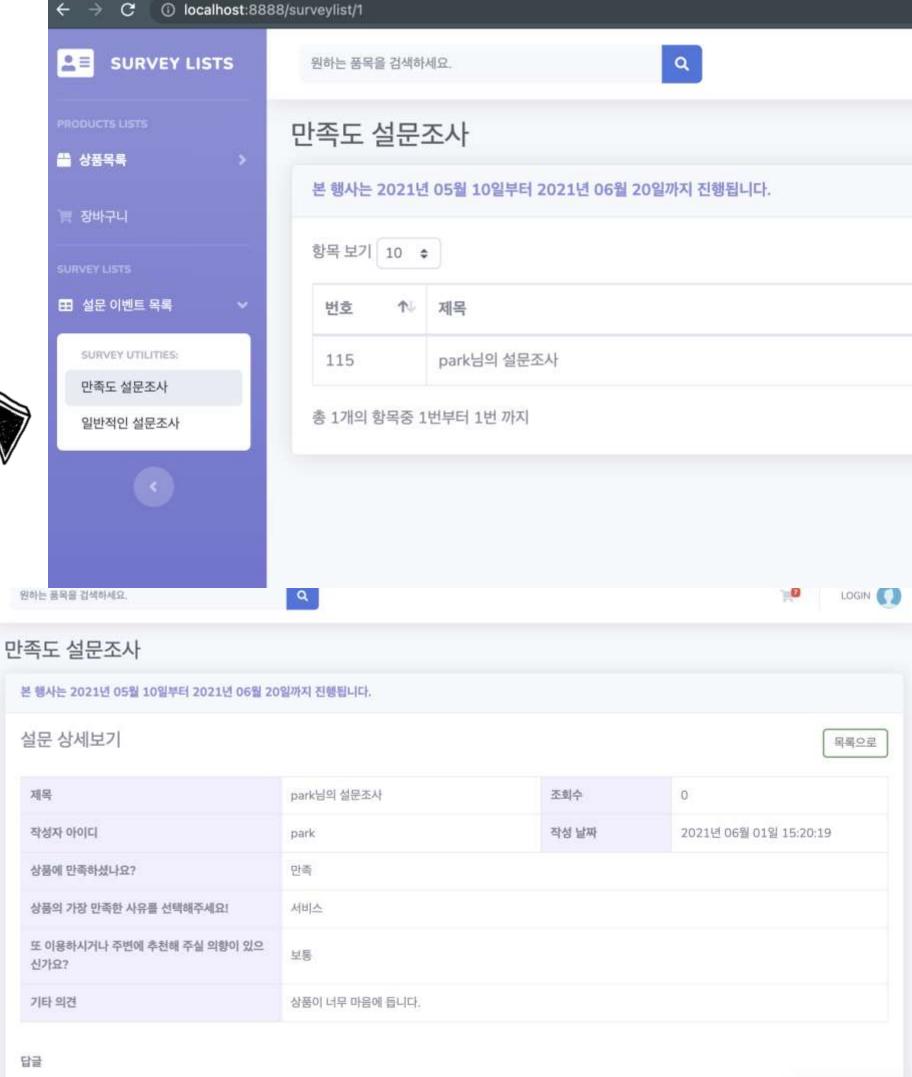
설문에 참여하면 목록이 나타납니다.



주소 /surveylist/\${이벤트 테이블 시퀀스}

> 행사 날싸는 처음 이벤트 생성때 포인트 지급 기간을 설정해 둔것입니다.





iocamost:63342/survey/survey\_main/templates/pages/writeEr

### 2. 화면구현 설문조사 목록 페이지

세션이 만료되었습니다.

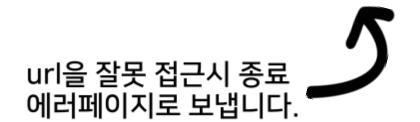
토큰 발급 제한시간이 지날경우, 토큰 만료 페이지로 보냅니다.



종료된 이벤트 입니다.

컨트롤러에서 토큰발급 유효시간을 30분으로 지정해두었습니다.

주소 /surveylist/\${토큰으로 변환된 유저 아이디}



localhost:8888/SurveyCompleted

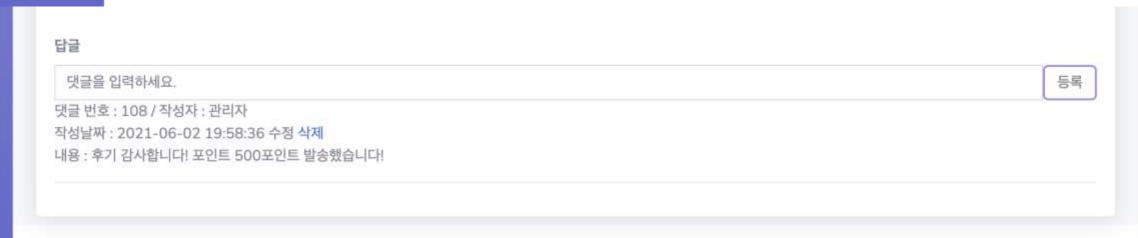
이미 설문작성을 하셨습니다. 감사합니다.

설문조사 작성후 또 다시 발급받은 url로 접속시 설문조사 완료 페이지로 보냅니다.



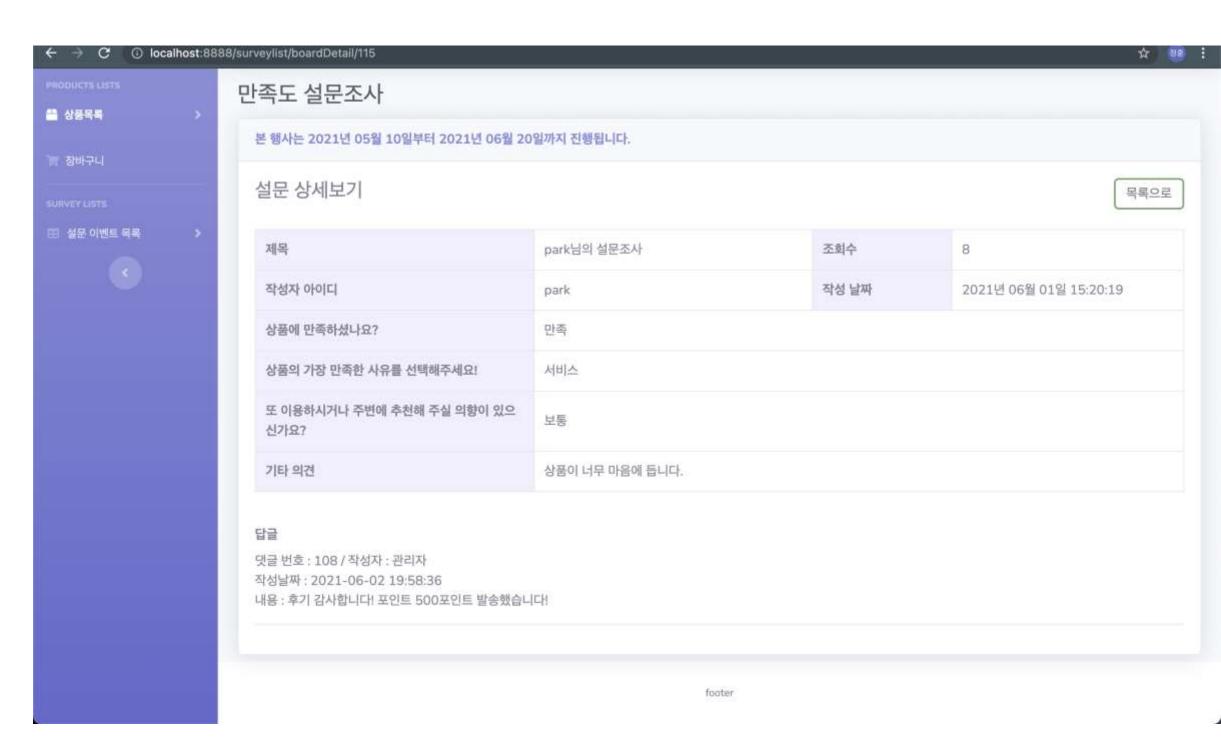
관리자 페이지로가서 유저가 작성한 설문조사에 답글을 달 수 있습니다. (포인트지급, 참여 감사)





## 2. 화면구현 설문조사 페이지

유저가 본 설문조사 상세보기 페이지



#### 3. 코드설명

핵심 코드 설명

```
/토큰 발급
GetMapping("/token/{id}")
oublic String TokenIssuance(@PathVariable String id, Model mod
  try {
       UserV0 userV0 = userService.urlTest(id);
       String jwt = createToken(userV0);
       System.out.println(jwt);
       model.addAttribute( attributeName: "jwt", jwt);
       UserV0 userV01 = SessionUtils.getUser(request);
       model.addAttribute( attributeName: "name", userV01 == nul
       Map<String, Object> claimMap = verifyJWT(jwt);
       System.out.println(claimMap); // 토큰이 만료되었거나 문제가있
       model.addAttribute( attributeName: "claimMap", claimMap)
       if (SessionUtils.setUser(userV0, request)) {
           model.addAttribute( attributeName: "test", userV0);
           attributes.addFlashAttribute( attributeName: "msg",
           attributes.addFlashAttribute( attributeName: "jwt", j
           attributes.addFlashAttribute( attributeName: "copybtn
           return "redirect:/userlist";
       } else {
           return "pages/token/NotInfo";
   } catch (Exception e) {
       e.printStackTrace();
       return "pages/token/NotInfo";
```

### 3. 코드설명 상품 업로드

경로 src/main/java/com/example/survey/ controller/ProductController.java

ProductController.java 상품 업로드 사진파일 위치 지정

```
private String boardInsertProc(HttpServletRequest request, @RequestPart MultipartFile productimagefile) throws Exception {
   // 게시글 등록 화면에서 입력한 값들을 실어나르기 위해 ProductDTO플 생성한다.
   ProductV0 product = new ProductV0();
   product.setProductname(request.getParameter( name: "productname"));
   product.setProductprice(Integer.parseInt(request.getParameter( name: "productprice")));
   product.setProductsalescnt(Integer.parseInt(request.getParameter( name: "productsalescnt")));
   product.setProductid(Integer.parseInt(request.getParameter( name: "productid")));
   ## (productimagefile.isEmpty()) { // 업로드할 파일이 없는 경우
       productService.productInsertService(product); // 게시글만 올린다.
   } else { // 업로드할 파일이 있는 경우
       // FilenameUtils : commons-io defendency를 사용.
        String Productimagefile = productimagefile.getOriginalFilename();
        String ProductimageOriName = productimagefile.getOriginalFilename();
        String fileNameExtension = FilenameUtils.getExtension(ProductimageOriName).toLowerCase();
        File destinationFile;
        String destinationFileName;
       // fileUrl = "uploadFiles 폴더의 위치";
        String productimageUrl = "/Users/mac/Desktop/git/Market-JWT-tokken-survey/Market-JWT-Tokken-Survey/src/main/resources/static/upload/";
       do {
           destinationFileName = RandomStringUtils.randomAlphanumeric( count: 32) + "." + fileNameExtension;
           destinationFile = new File( pathname: productimageUrl + destinationFileName);
        } while (destinationFile.exists());
       // MultipartFile.transferTo() : 요청 시점의 임시 파일을 로컬 파일 시스템에 영구적으로 복사해준다.
        destinationFile.getParentFile().mkdirs();
        productimagefile.transferTo(destinationFile);
       // 파일관련 자료를 product테이블에 등록한다.
        product.setProductno(product.getProductno());
       product.setProductimagefile(Productimagefile);
       product.setProductimageName(destinationFileName);
       product.setProductimageOriName(ProductimageOriName);
        product.setProductimageUrl(productimageUrl);
        System.out.println("Product Controller product : " + product);
        productService.productInsertService(product);
```

### 3. 코드설명 토큰 발급

경로 src/main/java/com/example/survey/ controller/BoardController.java

BoardController.java 토큰 발급 코드 토큰 유효시간은 30분으로 지정해 두었습니다.

```
//토큰 생성
public String createToken(UserV0 userV0) {
    //Header 부분 설정
   Map<String, Object> headers = new HashMap<>();
   headers.put("typ", "JWT");
    headers.put("alg", "HS256");
   //payload 부분 설정
   Map<String, Object> payloads = new HashMap<>();
   //payloads.put("data", "My JWT");
    payloads.put("seq", userVO.getSeq());
    payloads.put("id", userV0.getId());
    payloads.put("name", userV0.getName());
    payloads.put("email", userVO.getEmail());
    payloads.put("tel", userV0.getTel());
    payloads.put("eventSeq", userV0.getEventSeq());
    //Long expiredTime = 1000 * 60L * 60L * 2L; // 토큰 유효 시간 (2시간)
    Long expiredTime = 30 * 60 * 1000L; // 토큰 유효 시간 (30분)
   Date ext = new Date(); // 토큰 만료 시간
   ext.setTime(ext.getTime() + expiredTime);
   // 토큰 Builder
   String jwt = Jwts.builder()
           .setHeader(headers) // Headers 설정
           .setClaims(payloads) // Claims 설정
           //.setSubject(sampleVO.getEmail()) // 토큰 용도(제목)
           .setExpiration(ext) // 토큰 만료 시간 설정
           .signWith(SignatureAlgorithm. HS256, key.getBytes()) // HS256과 Key로 Sign
           .compact(); // 토큰 생성
    return jwt;
```

### 3. 코드설명

### 토큰 발급

경로 src/main/java/com/example/survey/ controller/BoardController.java

BoardController.java 토큰 검증

```
//토큰 검증
public Map<String, Object> verifyJWT(String jwt) throws UnsupportedEncodingException {
   Map<String, Object> claimMap = null;
   try {
       Claims claims = Jwts.parser()
               .setSigningKey(key.getBytes( charsetName: "UTF-8")) // Set Key
               .parseClaimsJws(jwt) // 파싱 및 검증, 실패 시 에러
               .getBody();
       claimMap = claims;
   } catch (ExpiredJwtException e) { // 토큰이 만료되었을 경우
       System.out.println(e);
   } catch (Exception e) { // 그외 에러났을 경우
       System.out.println(e);
    return claimMap;
```

#### 3. 코드설명

### 토큰 발급

경로 src/main/java/com/example/survey/ controller/BoardController.java

BoardController.java id로 토큰을 발급하여 발급한 토큰을 url에 적용하기위한 코드

```
//토큰 발급
@GetMapping("/token/{id}")
public String TokenIssuance(@PathVariable String id, Model model, HttpServletRequest request, RedirectAttributes attributes, HttpServletResponse response) {
        UserV0 userV0 = userService.urlTest(id);
       String jwt = createToken(userV0);
       System.out.println(jwt);
        model.addAttribute( attributeName: "jwt", jwt);
       UserV0 userV01 = SessionUtils.getUser(request);
        model.addAttribute( attributeName: "name", userV01 == null ? "NO NAME" : userV01.getName());
       Map<String, Object> claimMap = verifyJWT(jwt);
        System.out.println(claimMap); // 토큰이 만료되었거나 문제가있으면 null
        model.addAttribute( attributeName: "claimMap", claimMap);
       if (SessionUtils.setUser(userV0, request)) {
            model.addAttribute( attributeName: "test", userV0);
           attributes.addFlashAttribute( attributeName: "msg", attributeValue: true);
           attributes.addFlashAttribute( attributeName: "jwt", jwt);
           attributes.addFlashAttribute( attributeName: "copybtn", attributeValue: true);
            return "redirect:/userlist";
       } else {
            return "pages/token/NotInfo";
    } catch (Exception e) {
       e.printStackTrace();
        return "pages/token/NotInfo";
```



### 4. 일정

프로젝트 일정

### 프로젝트 일정



5/02	5/30		6/1
전체적인 틀 완성	자잘한 에러, 디테일한 기능 구현	전체적인 기능 재 점검	

프로젝트

프로젝트 기간

# 감사합니다!